ISSN 2085-4552

ULTIMATICS

Jurnal Teknik Informatika

Rekayasa | Vol. IX | No. 1 | Hal. 01-66 | Juni 2017 | ISSN 2085-4552

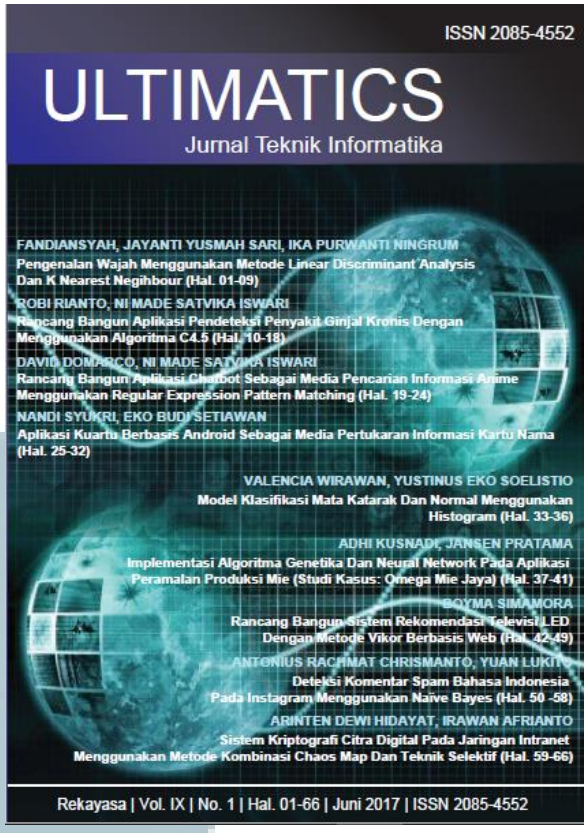

UMN

ISSN 2085-4552

SUSUNAN REDAKSI

ALAMAT REDAKSI

Universitas Multimedia Nusantara (UMN)
Jl. Scientia Boulevard, Gading Serpong
Tangerang, Banten, 15811
Tlp. (021) 5422 0808
Faks. (021) 5422 0800
Email: ultimatics@umn.ac.id



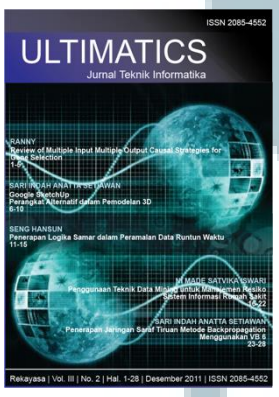
Jurnal ULTIMATICS merupakan Jurnal Program Studi Teknik Informatika Universitas Multimedia Nusantara yang menyajikan artikel-artikel penelitian ilmiah dalam bidang analisis dan desain sistem, *programming*, algoritma, rekayasa perangkat lunak, serta isu-isu teoritis dan praktis yang terkini, mencakup komputasi, kecerdasan buatan, pemrograman sistem *mobile*, serta topik lainnya di bidang Teknik Informatika. Jurnal ULTIMATICS terbit secara berkala dua kali dalam setahun (Juni dan Desember) dan dikelola oleh Program Studi Teknik Informatika Universitas Multimedia Nusantara bekerjasama dengan UMN Press.

Call for Paper

International Journal of New Media Technology (IJNMT) is a scholarly open access, peer-reviewed, and interdisciplinary journal focusing on theories, methods and implementations of new media technology. IJNMT is published annually by Faculty of Engineering and Informatics, Universitas Multimedia Nusantara in cooperation with UMN Press. Topics include, but not limited to digital technology for creative industry, infrastructure technology, computing communication and networking, signal and image processing, intelligent system, control and embedded system, mobile and web based system, robotics

Important Dates

October 31st, 2017
Deadline for submission of papers
November 30th, 2017
Announcement for Acceptance
December 15th, 2017
Deadline for submission of final papers



Jurnal ULTIMATICS merupakan Jurnal Program Studi Teknik Informatika Universitas Multimedia Nusantara yang menyajikan artikel-artikel penelitian ilmiah dalam bidang analisis dan desain sistem, *programming*, algoritma, rekayasa perangkat lunak, serta isu-isu teoritis dan praktis yang terkini, mencakup komputasi, kecerdasan buatan, pemrograman sistem *mobile*, serta topik lainnya di bidang Teknik Informatika.

Jurnal ULTIMA InfoSys merupakan Jurnal Program Studi Sistem Informasi Universitas Multimedia Nusantara yang menyajikan artikel-artikel penelitian ilmiah dalam bidang Sistem Informasi, serta isu-isu teoritis dan praktis yang terkini, mencakup sistem basis data, sistem informasi manajemen, analisis dan pengembangan sistem, manajemen proyek sistem informasi, *programming*, *mobile information system*, dan topik lainnya terkait Sistem Informasi.

Jurnal ULTIMA Computing merupakan Jurnal Program Studi Sistem Komputer Universitas Multimedia Nusantara yang menyajikan artikel-artikel penelitian ilmiah dalam bidang Sistem Komputer serta isu-isu teoritis dan praktis yang terkini, mencakup komputasi, organisasi dan arsitektur komputer, *programming*, *embedded system*, sistem operasi, jaringan dan internet, integrasi sistem, serta topik lainnya di bidang Sistem Komputer.

ISSN 2085-4552

DAFTAR ISI

ISSN 2085-4552

KATA PENGANTAR

Salam ULTIMA!

ULTIMATICS – Jurnal Teknik Informatika UMN kembali menjumpai para pembaca dalam terbitan saat ini Edisi Juni 2017, Volume IX, No. 1. Jurnal ini menyajikan artikel-artikel ilmiah hasil penelitian mengenai analisis dan desain *system*, pemrograman, analisis algoritma, rekayasa perangkat lunak, serta isu-isu teoritis dan praktis terkini.

Pada ULTIMATICS Edisi Juni 2017 ini, terdapat sembilan artikel ilmiah yang berasal dari para peneliti, akademisi, dan praktisi di bidang Teknik Informatika, yang mengangkat beragam topik, antara lain: Pengenalan Wajah Menggunakan Metode *Linear Discriminant Analysis* dan k Nearest Neighbour; Rancang Bangun Aplikasi Pendeteksi Penyakit Ginjal Kronis dengan Menggunakan Algoritma C4.5; Rancang Bangun Aplikasi *Chatbot* Sebagai Media Pencarian Informasi *Anime* Menggunakan *Regular Expression Pattern Matching*; Aplikasi Kuartu Berbasis Android Sebagai Media Pertukaran Informasi Kartu Nama; Model Klasifikasi Mata Katarak dan Normal Menggunakan *Histogram*; Implementasi Algoritma Genetika dan *Neural Network* Pada Aplikasi Peramalan Produksi Mie (Studi Kasus: Omega Mie Jaya); Rancang Bangun Sistem Rekomendasi Televisi LED dengan Metode Vikor Berbasis *Web*; Deteksi Komentar *Spam* Bahasa Indonesia Pada Instagram Menggunakan *Naïve Bayes*; Sistem Kriptografi *Citra Digital* pada Jaringan *Intranet* Menggunakan Metode Kombinasi *Chaos Map* dan Teknik Selektif.

Pada kesempatan kali ini juga kami ingin mengundang partisipasi para pembaca yang budiman, para peneliti, akademisi, maupun praktisi, di bidang Teknik dan Informatika, untuk mengirimkan karya ilmiah yang berkualitas pada: *International Journal of New Media Technology* (IJNMT), ULTIMATICS, ULTIMA InfoSys, ULTIMA *Computing*. Informasi mengenai pedoman dan *template* penulisan, serta informasi terkait lainnya dapat diperoleh melalui alamat surel ultimatics@umn.ac.id.

Akhir kata, kami mengucapkan terima kasih kepada seluruh kontributor dalam ULTIMATICS Edisi Juni 2017 ini. Kami berharap artikel-artikel ilmiah hasil penelitian dalam jurnal ini dapat bermanfaat dan memberikan sumbangsih terhadap perkembangan penelitian dan keilmuan di Indonesia.

Juni 2017,

Ni Made Satvika Iswari, S.T., M.T.
Ketua Dewan Redaksi

ISSN 2085-4552

Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan *k* Nearest Neighbor

Fandiansyah[1], Jayanti Yusmah Sari[2], Ika Purwanti Ningrum[3],
Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari, 93232, Indonesia
lastfandiansyah@gmail.com[1], jayanti.yusmah.sari@gmail.com[2], ika.purwanti.n@gmail.com[3]

Diterima 12 April 2017
Disetujui 2 Juni 2017

***Abstract*—Face recognition is one of the biometric system that mostly used for individual recognition in the absent machine or access control. This is because the face is the most visible part of human anatomy and serves as the first distinguishing factor of a human being. Feature extraction and classification are the key to face recognition, as they are to any pattern classification task.**
In this paper, we describe a face recognition method based on Linear Discriminant Analysis (LDA) and k-Nearest Neighbor classifier. LDA used for feature extraction, which directly extracts the proper features from image matrices with the objective of maximizing between-class variations and minimizing within-class variations. The features of a testing image will be compared to the features of database image using K-Nearest Neighbor classifier. The experiments in this paper are performed by using using 66 face images of 22 different people. The experimental result shows that the recognition accuracy is up to 98.33%.

***Index Terms*—face recognition, k nearest neighbor, linear discriminant analysis.**

I. PENDAHULUAN

Sistem biometrika adalah teknologi pengenalan diri menggunakan bagian tubuh manusia seperti sidik jari, telinga, wajah, geometri tangan, telapak tangan, retina, gigi dan bibir [1]. Pengenalan wajah menjadi salah satu sistem biomterika yang paling banyak digunakan untuk identifikasi personal, misalnya pada penggunaan mesin absensi atau akses *control*. Hal ini karena wajah merupakan salah satu biometrika yang paling umum digunakan untuk mengenali seseorang [2]. Selain itu, pengenalan wajah juga tidak mengganggu kenyamanan seseorang saat akuisisi citra. Pada sistem pengenalan wajah, citra wajah dari seorang individu akan diakuisisi sebagai citra masukan yang kemudian dicocokkan dengan sekumpulan citra wajah yang sebelumnya telah disimpan di dalam *database* untuk mengenali citra wajah dari individu tersebut.

Penelitian tentang pengenalan wajah telah dimulai sejak tahun 1970-an [3]. Secara umum, ada 4 tahapan dalam sistem pengenalan wajah yaitu akusisi citra wajah, *preprocessing*, ekstraksi fitur dan klasifikasi citra wajah tersebut. Dari keempat tahapan tersebut, tahapan ekstraksi fitur dan klasifikasi merupakan tahapan yang paling penting dalam sistem pengenalan wajah [4].

Ada banyak metode ekstraksi fitur yang telah dikembangkan untuk membangun sistem pengenalan wajah. Dari semua metode tersebut, masing-masing memiliki kelebihan dan kelemahan. Ada metode yang membutuhkan waktu komputasi yang cepat namun mengorbankan tingkat akurasi pengenalan, dan ada pula metode dengan tingkat akurasi pengenalan yang tinggi namun membutuhkan waktu komputasi yang lama. Baik waktu komputasi maupun tingkat akurasi pada sistem pengenalan wajah, keduanya dipengaruhi oleh metode yang digunakan pada tahap ekstraksi fitur [4] dan metode yang digunakan pada tahap klasifikasi [5]. Oleh karena itu, untuk mengoptimalkan waktu komputasi tanpa harus mengorbankan tingkat akurasi pengenalan, penelitian ini membangun sistem pengenalan wajah dengan memfokuskan pada kedua tahapan yaitu tahapan ekstraksi fitur dan tahapan klasifikasi.

Salah satu metode ekstraksi fitur yang telah digunakan untuk sistem pengenalan wajah yaitu *Principal Component Analysis* (PCA). Metode PCA bertujuan untuk mereduksi dimensi dengan melakukan transformasi linear dari suatu ruang berdimensi tinggi ke dalam ruang berdimensi rendah [6]. Namun, kelemahan dari metode PCA adalah kurang optimal dalam pemisahan antar kelas sehingga dapat mempengaruhi tingkat akurasi pengenalan wajah [7]. Untuk mengatasi masalah tersebut, Belhumeur dkk [7] memperkenalkan *Linear Discriminant Analysis* (LDA) untuk pengenalan wajah. Metode ini bekerja dengan menemukan subruang linear yang memaksimalkan perpisahan dua kelas pola menurut *Fisher Criterion* (bobot *fisher* kriteria). Hal ini dapat diperoleh dengan

meminimalkan jarak matriks sebaran *within-class* S_w dan memaksimalkan jarak matriks sebaran *between-class* S_b secara simultan sehingga menghasilkan *fisher criterion* yang maksimal. LDA akan menemukan subruang linear di mana kelas-kelas saling terpisah dengan memaksimalkan *fisher criterion* [2]. Metode LDA digunakan dalam penelitian ini sebagai metode ekstraksi fitur untuk mengoptimalkan pemisahan antar kelas citra wajah sehingga hasil pengenalannya lebih akurat. Dengan demikian, metode LDA akan diuji berdasarkan tingkat akurasi pengenalan.

Setelah melalui tahapan ekstraksi fitur, fitur-fitur penting dari citra wajah nantinya akan digunakan untuk proses klasifikasi atau pengenalan. Metode klasifikasi yang akan digunakan dalam penelitian ini adalah metode klasifikasi *k nearest neighbor*. Metode klasifikasi *k nearest neighbor* memiliki komputasi yang lebih sederhana namun kemampuan klasifikasinya lebih baik dibandingkan dengan metode lain seperti Hidden Markov Model dan metode Kernel method [5]. Metode *k nearest neighbor* adalah suatu metode yang menggunakan algoritma *supervised* di mana hasil dari *query instance* masukan akan diklasifikasikan berdasarkan mayoritas dari kategori pada *k nearest neighbor* [5]. Penelitian ini bertujuan untuk mengimplementasikan metode ekstraksi fitur *Linear Discriminant Analysis* (LDA) dan metode klasifikasi *k nearest neighbor* untuk membangun sistem pengenalan wajah dengan tingkat akurasi pengenalan yang tinggi dan waktu komputasi yang optimal.

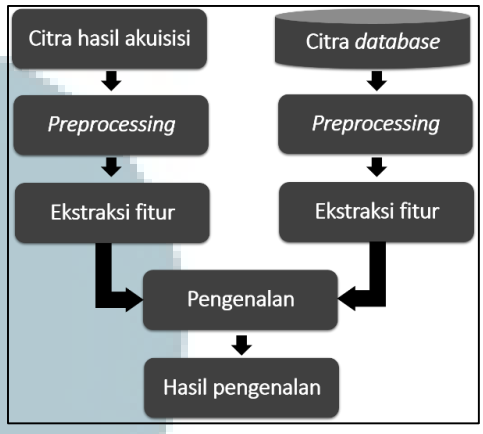
Pembahasan penelitian ini dibagi dalam beberapa bagian. Analisis masalah dan penelitian terkait diuraikan pada bagian "PENDAHUULUAN". Metodologi penelitian yang diusulkan dibahas pada bagian "METODE PENELITIAN". Selanjutnya, hasil pengujian dibahas pada bagian "HASIL dan PEMBAHASAN". Adapun kesimpulan dan saran untuk penelitian selanjutnya dibahas pada bagian "KESIMPULAN dan SARAN",

II. METODE PENELITIAN

A. *Perancangan Sistem*

Sistem pengenalan wajah menggunakan metode *Linear Discriminant Analisys* (LDA) dan *k nearest neighbor* dirancang agar dapat mendeteksi wajah seseorang pada citra digital kemudian mengenali citra wajah tersebut dengan cara mencocokkan hasil ekstraksi fiturnya dengan fitur citra wajah yang sudah disimpan di dalam *database*.

Dalam penenelitian ini, digunakan 66 citra wajah yang berformat **.png* dan berasal dari 22 Mahasiswa Jurusan Teknik Informatika Universitas Halu Oleo. Setiap individu masing-masing mempunyai 3 citra wajah dengan ekspresi yang berbeda-beda. Jumlah data citra wajah yang akan diidentifikasi sebanyak 56 citra meliput: 20 citra dari 20 individu yang datanya ada di *database*, 20 citra dari 20 individu yang datanya ada di *database* namun diberikan gangguan *ocean distortion*, 12 citra dari 12 individu yang datanya ada di dalam *database* namun diakuisisi dengan pose wajah yang berbeda, serta 4 citra dari 4 individu yang berbeda namun diakuisisi dengan pencahayaan yang berbeda. Gambar. 1 menunjukkan gambaran umum sistem yang diusulkan.



Gambar 1. Gambaran Umum Sistem

B. *Perancangan Proses*

Perancangan proses dibagi menjadi dua yaitu perancangan tahap *preprocessing*, dan perancangan tahap *processing* yang meliputi ekstraksi fitur dan pengenalan (*recognize*). Gambar 1 menggambarkan proses yang dilakukan oleh sistem. Proses yang akan dilakukan pertama adalah tahap akuisis citra *testing* maupun citra *database* berfungsi untuk pengambilan citra wajah, tahap *preprocessing* yang berfungsi untuk menyeragamkan ukuran citra wajah dan memperbaiki kualitasnya sebelum melalui tahap ekstraksi fitur. Selanjutnya akan dimulai pembentukan fitur citra *testing* yang berfungsi mengekstraksi ciri fitur citra untuk dipergunakan sebagai perhitungan mencari bobot.

Tahap pembentukan fitur citra *database* berfungsi sama seperti ekstraksi citra *testing*. Proses pengenalan menggunakan metode klasifikasi *k nearest neighbor* yang berfungsi untuk mengelompokkan informasi fitur citra dari proses pembentukan fitur citra *database*, kemudian dilakukan klasifikasi pada sistem dengan data yang sudah diberikan *label*.

C. *Preprocessing citra*

Citra wajah yang disimpan di *database* dan citra yang akan diidentifikasi harus melalui tahap

preprocessing terlebih dahulu yang meliputi akuisisi citra, konversi citra RBG-*grayscale*, dan ekualisasi *histogram*. Berikut tahapan *preprocessing* yang harus dilakukan secara berurutan.

1) Akuisisi citra wajah menggunakan *webcam* laptop. Citra hasil akuisisi merupakan citra RGB 24 bit berformat PNG dengan ukuran 92 x 112 piksel.
2) Konversi citra wajah hasil akuisisi dari RGB ke *grayscale*.
3) Ekualisasi histogram (*histogram equalization*), hasilnya akan disimpan untuk proses pengenalan selanjutnya yang terdiri dari 66 citra yang ada di dalam *database* wajah dan 22 citra sebagai citra *testing*. Tahapan ini merupakan tahapan terakhir dari *preprocessing* citra wajah.

D. Processing Citra

Pada tahapan *processing*, akan diterapkan metode LDA untuk menghasilkan vektor fitur dari citra wajah dan melakukan pencocokan vektor fitur citra di *database* dengan vektor fitur citra *testing* menggunakan perhitungan *k nearest neighbor*.

D.1. Proses Pembentukan Vektor Fitur Citra

Setelah melakukan tahapan *preprocessing*, maka selanjutnya adalah proses ekstrasi fitur citra. Ekstraksi fitur dilakukan pada citra *testing* maupun citra yang ada di dalam *database*. Pada tahap awal proses ekstraksi fitur pada penelitian ini digunakan Algoritma *Eigenface* sebagai salah satu algoritma berbasis metode PCA (*Principal Component Analysis*) untuk memperoleh bobot. Bobot tersebut selanjutnya akan digunakan pada metode LDA (*Linear Discriminant Analysis*) untuk membentuk vektor fitur citra wajah.

Perhitungan PCA [6]

Eigenface merupakan salah satu algoritma pengenalan wajah manusia yang berbasis metode PCA. Untuk menghasilkan *eigenface*, sekumpulan citra digital dari wajah manusia diambil pada kondisi pencahayaan yang sama kemudian dinormalisasikan dan diproses pada resolusi yang sama (misal *m* x *n*), kemudian citra tersebut diperlakukan sebagai vektor dimensi *m* x *n* di mana komponennya diambil dari nilai piksel citra.

Berikut langkah-langkah Algoritma *Eigenface* untuk memperoleh bobot dari citra *database*.

a. Mengubah *M* jumlah citra di *database* dengan ukuran *B* x *K* (92 x 112) piksel ke dalam bentuk vektor *Y* dengan panjang 1 x 10304.

$$X = \begin{bmatrix} U_{1,1} & \dots & U_{1,92} \\ \vdots & \ddots & \vdots \\ U_{112,1} & \dots & U_{112,92} \end{bmatrix} \quad (1)$$

$$Y = [\Gamma_1, \Gamma_2, \dots, \dots, \Gamma_{10304}] \quad (2)$$

b. Simpan vektor dari 66 citra di *database* ke dalam *list image* (*S*).

$$S = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \dots \\ \Gamma_{66} \end{bmatrix} = \begin{bmatrix} U_{1,1} & U_{1,2} & \dots & U_{1,10304} \\ U_{2,1} & U_{2,2} & \dots & U_{2,10304} \\ \dots & \dots & \dots & \dots \\ U_{66,1} & U_{66,2} & \dots & U_{66,10304} \end{bmatrix} \quad (3)$$

c. Mencari matriks rata-rata (Ψ) dari semua piksel citra di *database* P_m dengan *m* merupakan jumlah piksel citra dan *M* adalah jumlah citra di *database*, dengan menggunakan persamaan (4) sehingga dihasilkan matriks vektor rata-rata dengan dimensi 1 x 10304.

$$\Psi = \frac{1}{M} \sum_{m=1}^{M} P_m \; ; \; m = 1,2,..M \quad (4)$$

d. Menghitung selisih (Φ) antara citra di *database* Γ_m dengan nilai tengah (Ψ) menggunakan persamaan (5) sehingga diperoleh matriks selisih Φ berukuran 66 x 10304.

$$\Phi = \Gamma_M - \Psi \quad (5)$$

e. Kemudian tentukan matriks kovarian *C* dengan mengalikan selisih Φ dan matriks *transpose* Φ^T menggunakan persaamaan (6). Matriks *C* yang diperoleh nanti berukuran 66 x 66.

$$C = \Phi \, \Phi^T \quad (6)$$

f. Tentukan nilai *eigenvalue* (λ) dan *eigenvector* (υ) dari matriks kovarian *C* menggunakan persamaan (7). (λ) yang diperoleh berukuran 1 x 66 dan sedangkan υ berukuran 66 x 66.

$$\begin{gathered} C\,V = \lambda\, V \\ (\mathrm{C} - \lambda \mathrm{I})V = 0 \\ (\Phi\, \Phi^T - \lambda \mathrm{I})V = 0 \end{gathered} \quad (7)$$

Dengan *I* adalah matriks identitas, λ adalah *eigenvalue* dari *C* dan *V* adalah *eigenvector* yang bersesuian dengan λ. Untuk mencari nilai λ, maka *determinant* (*C* - λ*I*) harus sama dengan 0. Setiap citra akan memiliki *n* buah λ, ambil λ dengan nilai tertinggi dan cari *eigenvector* yang bersesuian dengan λ tersebut. Setiap *eigenvector* akan memiliki panjang *M* elemen.

g. Masukkan nilai *eigenvector* (*v*) yang dipilih ke dalam matriks *eigenface* (*Ek*) dengan mengalikan *v* yang berukuran *M*x*M* (*M*=66) dengan matriks selisih Φ yang berukuran *M* x *BK* (*BK*=10304) dan simpan hasilnya dalam matriks yang berukuran *M* x *BK*.

$$E_k = \sum_{k=1}^{M} v_k \Phi_k \; ; \; k = 1,2,\dots,M \quad (8)$$

h. Normalisasi *eigenface* dengan menggunakan persamaan (9), dan simpan *eigenface* hasil

normalisasi ini ke dalam matriks *En* berukuran 45 x 10304 pada *feature extraction*.

$$E_{nk} = \frac{E_k}{E_k, E_k^T} \quad (9)$$

i. Hitung bobot dengan mengalikan matriks selisi Φ_k berukuran *M* x *BK* dengan matriks *transpose* dari E_{nk} yang berukuran *BK* x *M*, menggunakan persamaan (10).

$$\omega = \Phi_k \text{x } E_{nk}^T \quad (10)$$

j. Bobot yang diperoleh merupakan nilai ciri yang yang digunakan dalam proses pengenalan selanjutnya. Simpan bobot ini ke dalam matriks **Ω** berukuran *M* x *N* dengan *N* < *M*.

Perhitungan LDA

LDA bekerja berdasarkan analisis matriks penyebaran yang bertujuan menemukan suatu proyeksi optimal sehingga dapat memproyeksikan data *input* pada ruang dengan dimensi yang lebih kecil dimana semua pola dapat dipisahkan semaksimal mungkin. Karenanya untuk tujuan pemisahan tersebut maka LDA akan mencoba untuk memaksimalkan penyebaran data-data *input* di antara kelas-kelas yang berbeda dan sekaligus juga meminimalkan penyebaran *input* pada kelas yang sama. Perbedaan antar kelas direpresentasikan oleh matriks *Sb* dan perbedaan dalam kelas direpresentasikan oleh matriks *Sw*. Berikut langkah-langkah pembentukan vektor fitur citra wajah menggunakan metode LDA [8][9].

a. Hasil dari bobot *eigenfaces* (E_k) dijadikan sebagai input yang akan ditranformasikan ke dalam vektor kolom.

b. Menghitung rata-rata dalam kelas (m_i) dan rata-rata keseluruhan kelas (m) dari seluruh citra di *database*.

c. Menghitung matriks sebaran antar kelas (*between class scatter matrix*, S_b).

$$S_b = \sum_{i=1}^{k} n_i (m_i - m_0)(m_i - m_0)^T \quad (11)$$

d. Menghitung matriks sebaran dalam kelas (*within class scatter matrix* S_w).

$$S_w = \sum_{i=i}^{k} \sum_{j=1}^{n_i} \left(x_i^{(j)} - m_i\right)\left(x_i^{(j)} - m_i\right)^T \quad (12)$$

e. Memproyeksikan matriks sebaran dalam kelas. Matriks sebaran dalam kelas (S_w) adalah jarak matriks dalam kelas yang sama, menggunakan persamaan (13).

$$J_2(W) = maxtrace\ ((W^T S_w W)^{-1}(W^T S_b W)) \quad (13)$$

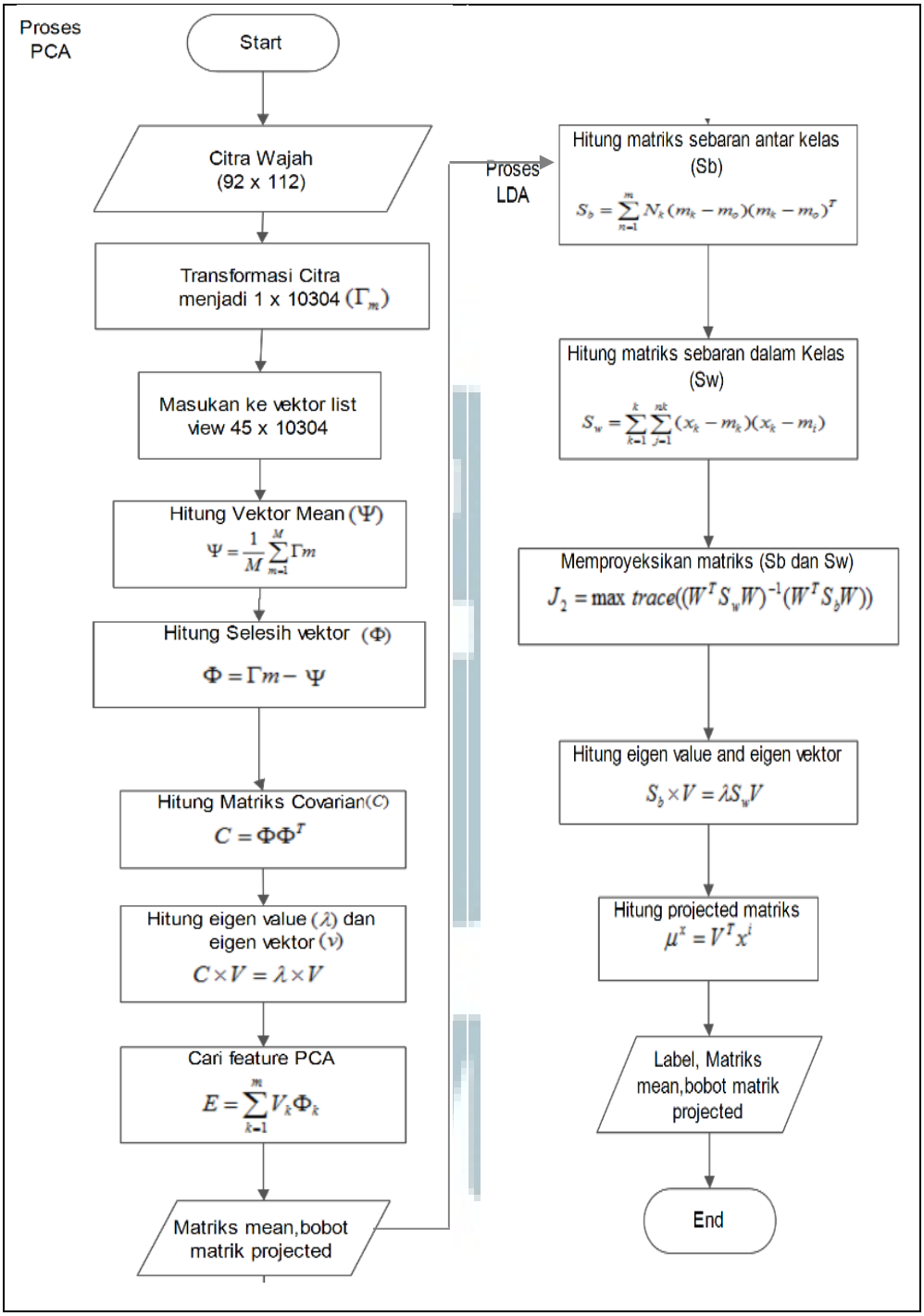
f. Mencari *eigen value* (λ) dan nilai *eigenvector* (υ) menggunakan persamaan (14).

$$S_b \upsilon = \lambda S_w \upsilon \quad (14)$$

g. Mengurutkan *eigen value* (λ) sesui dengan urutan nilai yang ada pada nilai eigen dari besar ke kecil. Selanjutnya proyeksi menggunakan *k*-1 *eigenvector* (υ) (di mana *k* adalah jumlah kelas).

h. Memproyeksikan seluruh citra asal (bukan *centerend image*) ke *fisher basis* vektor dengan menghitung *dot product* dari citra asal V^T ke tiap-tiap *fisher basis vector* x^i menggunakan persamaan (15).

$$\bar{u}^x = V^T x^i \quad (15)$$



Gambar 2. Proses Pembentukan Vektor Fitur Citra *Database*

Pembentukan vektor fitur untuk pengenalan wajah memiliki beberapa tahapan seperti yang telah ditunjukkan pada Gambar 2. Untuk ekstraksi fitur citra *testing* yang telah melalui tahap *preprocessing* dilakukan langkah-langkah yang sama dengan ekstraksi fitur citra pada citra *database*. Perbedaanya hanya terletak pada jumlah citra (*M*), jika pada ekstraksi fitur citra *database* nilai *M*=66, maka pada ekstraksi fitur citra *testing*, nilai *M*=1.

D.2. Proses Pencocokan Fitur

Pencocokan bertujuan untuk menentukan kelas dari suat citra *testing* berdasarkan ciri-ciri yang telah diekstraksi. Proses pencocokan fitur hasil ekstraksi menggunakan metode klasifikasi *k nearest neighbor (kNN)*. Metode klasifikasi *k nearest neighbor* merupakan pengembangan dari metode klasifikasi *nearest neighbor (NN)* [5]. Pada metode klasifikasi *nearest neighbor*, citra masukan akan diuji berdasarkan jarak fiturnya dengan fitur citra lain di dalam *database* untuk memperoleh satu citra dengan nilai jarak fitur paling minimum. Citra ini disebut dengan ketetanggaan terdekat.

Pada metode klasifikasi *nearest neighbor*, diperoleh satu citra yang memiliki jarak minimum dengan citra uji. Sedangkan pada metode klasifikasi *k nearest neighbor* dihasilkan citra sample sejumlah *k* yang memiliki jarak minimum dengan citra uji. Setiap citra sampel tersebut masing-masing memiliki label kelas. Citra uji akan diklasifikasikan ke dalam suatu kelas dengan jumlah citra sampel yang paling banyak dari citra sample sejumlah *k*. Nilai *k* pada *kNN* ini mempengaruhi performansi dari metode klasifikasi *kNN* [5].

Metode klasifikasi *k nearest neighbor* melakukan proses pencocokan/pengenalan berdasarkan jumlah tetangga terdekat untuk penentukan kelasnya. Untuk mencari jarak kelas menggunakan perhitungan jarak *euclidean distance*. Tahapan dalam metode klasifikasi *k nearest neighbor* yaitu [10]:

a. Menentukkan nilia *k*.
b. Menghitung jarak antara citra *testing* dengan seluruh citra pada *database* menggunakan persamaan *euclidean distance*, persamaan (16) dan menentukan citra terdekat dengan citra *testing* berdasarkan nilai *k*.

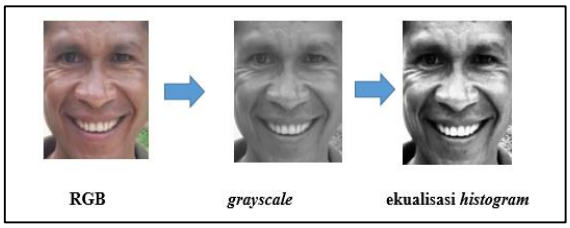
$$d_i = \sqrt{\sum_{i=1}^{p} (x_{2i} - x_{1i})^2} \qquad (16)$$

c. Menentukan hasil klasifikasi berdasarkan kelas yang memiliki anggota terbanyak.
d. Jika terjadi konflik atau keadaan seimbang pada kelas dengan jumlah anggota yang sama maka digunakan pemecahan konflik.

III. Hasil dan Pembahasan

A. Preprocessing

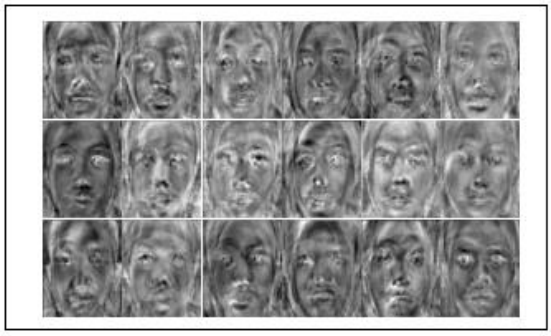
Tahap *preprocessing* berfungsi untuk menyiapkan citra sebelum proses ekstraksi fitur. *Preprocessing* dalam penelitian ini meliputi akuisisi citra, konversi citra RGB-*grayscale* dan ekualisasi *histogram*. Dari Gambar 3, dapat dilihat bahwa *output* dari tahap *preprocessing* berupa citra *grayscale* yang telah mengalami ekualisasi histogram.



Gambar 3. Contoh Hasil *Preprocessing*

B. Ektraksi fitur

Selanjutnya citra akan diproses melalui tahapan ekstraksi fitur untuk membuat suatu *set fisherface* dari citra *database* menggunakan perhitungan *Principal Component Analysis* dan *Linear Discriminant Analysis*. *Fisherface* ini sebenarnya merupakan seatur vektor eigen dengan nilai eigen tertentu yang ditampilkan ke dalam gambar dua dimensi dengan *mode grayscale*. Suatu set *fisherface* setiap gambar wajah dapat diekstraksi kembali dengan bobot yang berbeda beda dari tiap *fisherface*. Bobot *fisherface* tersebut dibandingkan untuk melakukan pencocokan dari citra *testing*.



Gambar 4. Hasil Ektraksi Fitur Citra *Database*

Gambar 4 merupakan hasil dari ektraksi fitur citra *database* menggunakan metode LDA, walaupun hasil ekstraksi fitur tidak menyerupai gambar aslinya, akan tetapi setiap gambar yang merupakan satu kelas memiliki gambar yang hampir sama yang diproyeksikan ke dalam kelas yang sama perbedaan gambarnya dapat dihilangkan atau diekstraksi dengan menjadi satu gambar.

C. Hasil Pengujian

Pengujian sistem pengenalan wajah menggunakan metode LDA dan *k*-NN pada penelitian ini dilakukan dengan dua skenario uji coba. Pengujian dilakukan dengan menggunakan 20 citra wajah dari individu yang citra wajahnya sudah

disimpan terlebih dahulu di dalam *database*. Setiap individu memiliki 3 sampel citra wajah pada *database*.

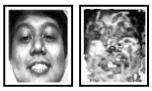
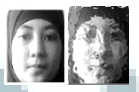
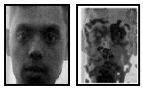
Pengujian I menggunakan citra uji normal, yaitu citra wajah hasil akuisisi dengan *pose* dan ekspresi berbeda-beda. Hasil dari pengujian ini ditunjukkan pada Tabel 1. Adapun pengujian II dilakukan menggunakan citra uji dengan gangguan *ocean distortion*. Pengujian ini bertujuan untuk mengukur performansi sistem pengenalan wajah yang dibuat dalam penelitian ini. Hasil dari pengujian kedua ditunjukkan dalam Tabel 2. Hasil pengujian pada Tabel 1 dan Tabel 2 berupa pengenalan yang kurang tepat yaitu citra uji dikenali sebagai wajah dari individu lain, diberi tanda kotak merah.

Tabel 1 menunjukkan bahwa dari hasil uji coba pertama, yaitu pengujian dengan citra wajah normal menggunakan nilai k=3, 5 dan 7, diperoleh satu pengenalan wajah yang kurang tepat yaitu pada individu ke-12. Sedangkan pada Tabel 2, dapat dilihat bahwa terdapat 8 kesalahan pengenalan saat uji coba kedua menggunakan citra wajah yang diberi efek gangguan *ocean distortion* yaitu pada individu ke-6 dan ke-12 untuk k=3, pada individu ke-11, 12 dan 13 untuk k=5 dan pada individu ke-9, 12 dan 13 untuk k=7.

Tabel 1 Hasil Pengujian I Menggunakan Citra Uji Normal

No	Citra *Testing*	Hasil pengenalan menggunakan *kNN* disertai jarak (*d*)			No	Citra *Testing*	Hasil pengenalan menggunakan *kNN*		
		k=3	*k*=5	*k*=7			*k*=3	*k*=5	*k*=7
1		 d = 0.1798	 d = 0.1485	 d = 0.1485	11		 d = 0.1054	 d = 0.10548	 d = 0.1305
2		 d = 0.2705	 d = 0.2529	 d = 0.2323	12		 d = 0.1942	 d = 0.1942	 d = 0.2983
3		 d = 0.0638	 d = 0.1014	 d = 0.2577	13		 d = 0.2035	 d = 0.3035	 d = 0.3894
4		 d = 0.0930	 d = 0.1184	 d = 0.1474	14		 d =0.1899	 d = 0.1903	 d = 0.4398
5		 d = 0.3567	 d = 0.3238	 d = 0.3238	15		 d = 0.2866	 d = 0.2886	 d = 0.3429
6		d = 0.2853	d = 0.3233	d = 0.3238	16		d = 0.1987	d = 0.2289	d = 0.2389
7		d = 0.0972	d = 0.0659	d = 0.0955	17		d = 0.1142	d = 0.1873	d = 0.1873
8		d = 0.1939	d = 0.2199	d = 0.2199	18		d =0.1639	d = 0.2254	d = 0.2254
9		d = 0.2388	d = 0.2493	d = 0.1762	19		d = 0.1375	d = 0.1459	d = 0.7834
10		d = 0.2707	d = 0.3060	d = 0.0760	20		d = 0.2108	d = 0.2108	d = 0.2108

Tabel 2 Hasil Pengujian II Menggunakan Citra Uji Dengan Penambahan Gangguan *Ocean Distortion*

No	Citra *Testing*	Hasil pengenalan menggunakan *kNN* disertai jarak (*d*)			No	Citra *Testing*	Hasil pengenalan menggunakan *kNN*		
		k=3	*k*=5	*k*=7			*k*=3	*k*=5	*k*=7
1		*d* = 0.0881	*d* = 0.1177	*d* = 0.1177	11		*d* = 0.1320	*d* = 0.1320	*d* = 0.1320
2		*d* = 0.1267	*d* = 0.1267	*d* = 0.1267	12		*d* = 0.0319	*d* = 0.0627	*d* = 0.0627
3		*d* = 0.0760	*d* = 0.1091	*d* = 0.1091	13		*d* = 0.0940	*d* = 0.1203	*d* = 0.1203
4		*d* = 0.1626	*d* = 0.1626	*d* = 0.1626	14		*d* = 0.1824	*d* = 0.1824	*d* = 0.1824
5		*d* = 0.1672	*d* = 0.1672	*d* = 0.1672	15		*d* = 0.1732	*d* = 0.1752	*d* = 0.2939
6		*d* = 0.1427	*d* = 0.1784	*d* = 0.1784	16		*d* = 0.1767	*d* = 0.1767	*d* = 0.1912
7		*d* = 0.1677	*d* = 0.1677	*d* = 0.1677	17		*d* = 0.1253	*d* = 0.1284	*d* = 0.1252
8		*d* = 0.2297	*d* = 0.1097	*d* = 0.2297	18		*d* = 0.1808	*d* = 0.1808	*d* = 0.1802
9		*d* = 0.2218	*d* = 0.2218	*d* = 0.2218	19		*d* = 0.1101	*d* = 0.1459	*d* = 0.1101
10		*d* = 0.1380	*d* = 0.1380	*d* = 0.1380	20		*d* = 0.1398	*d* = 0.1402	*d* = 0.1398

Tabel 3. Hasil Pengujian I

Nilai *k*	Benar	Salah	Citra *Testing*	Akurasi
3	20	0	20	100%
5	20	0	20	100%
7	19	1	20	95%
Rata-rata akurasi pengenalan				**98.33%**

Tabel 4. Hasil Pengujian II

Nilai *k*	Benar	Salah	Citra *Testing*	Akurasi
3	18	2	20	90%
5	17	3	20	85%
7	17	3	20	85%
Rata-rata akurasi pengenalan				**86.66%**

Tabel 3 dan Tabel 4 menunjukkan akurasi pengelanan terhadap citra uji yang normal dan citra uji dengan gangguan *ocean distortion*. Akurasi pengenalan dihitung berdasarkan hasil pengujian

pada Tabel 1 dan 2, dengan membandingan jumlah citra uji dengan hasil pengenalan benar terhadap jumlah seluruh citra yang diuji dengan menggunakan persamaan (17).

$$Akurasi = \frac{Total\ Pengenalan\ benar}{Total\ Citra\ Uji} x100\ \% \qquad (17)$$

Dari Tabel 3 dapat dilihat bahwa nilai akurasi tertinggi yaitu 100% untuk pengujian pertama, yaitu pengenalan citra uji normal diperoleh dengan *k*=3 dan *k*=5. Sedangkan untuk nilai akurasi tertinggi yaitu 90% pada pengujian kedua seperti yang ditunjukkan pada Tabel 4, diperoleh pada nilai *k*=3. Hal ini tidak berarti bahwa nilai *k*=3 lebih baik dibandingkan dengan *k*=5 atau 7. Hasil uji coba ini hanya menunjukkan bahwa nilai *k* yang sesuai dengan data pada penelitian ini adalah *k*=3. Adapun akurasi rata-rata yang diperoleh pada pengujian pertama dan kedua secara berturut-turut yaitu sebesar 98.33% dan 86.66%. Hasil pengujian ini membuktikan bahwa metode *Linear Discriminant Analisys* dan *k nearest neighbor* mampu mengoptimalkan hasil pengenalan pada sistem pengenalan wajah. Hal ini terlihat dari tingkat akurasi pengenalan yang cukup tinggi baik pada pengujian I yang menggunakan citra uji normal maupun pada pengujian II yang menggunakan citra uji dengan gangguan *ocean distortion*.

IV. KESIMPULAN

Dalam penelitian ini, dibangun sistem pengenalan wajah menggunakan metode *Linear Discriminant Analisys* dan *k nearest neighbor*. Metode *Linear Discriminant Analisys* digunakan untuk mengekstraksi fitur citra wajah sedangkan metode *k nearest neighbor* digunakan untuk mencocokkan fitur citra wajah pada tahap pegenalan. Sistem telah diuji menggunakan 66 citra wajah dari 22 individu. Pengujian menggunakan citra wajah dalam keadaan normal menghasilkan rata-rata akurasi pengenalan sebesar 98.33% sedangkan pengujian menggunakan citra wajah dengan gangguan *noise* menghasilkan rata-rata akurasi pengenalan sebesar 86,66%. Berdasarkan hasil pengujian tersebut, maka dapat disimpulkan bahwa sistem pengenalan wajah yang dibangun menggunakan metode *Linear Discriminant Analisys* dan *k nearest neighbor* mampu melakukan pengenalan wajah dengan akurasi pengenalan yang baik, yaitu mencapai 98.33%.

V. SARAN

Adapun saran untuk penelitian tetang pengenalan wajah menggunakan metode *Linear Discriminant Analisys* dan *k nearest neighbor* yang selanjutnya yaitu perlu dilakukan penelitian untuk perbandingan hasil klasifikasi dengan menggunakan metode klasifikasi yang lain, seperti *Fuzzy k-Nearest Neighbor* dan SVM (*Support Vector Machine*).

DAFTAR PUSTAKA

[1] S.Z. Li, dan A. Jain, Encyclopedia of biometrics. Springer Publishing Company, Incorporated, 2015.

[2] F. Mahmud, M. T. Khatun, S. T. Zuhori, S. Afroge, M. Aktar, dan B. Pal, "Face recognition using Principle Component Analysis and Linear Discriminant Analysis," Electrical Engineering and Information Communication Technology (ICEEICT), 2015 International Conference on, pp. 1-4. IEEE, 2015.

[3] P.S. Hiremath dan M. Hiremath, "Linear Discriminant Analysis for 3D Face Recognition Using Radon Transform," in Swamy P., Guru D. (eds) Multimedia Processing, Communication and Computing Applications, Lecture Notes in Electrical Engineering, vol 213. Springer, New Delhi, 2013.

[4] M. Li, dan B. Yuan, "2D-LDA: A statistical linear discriminant analysis for image matrix," *Pattern Recognition Letters* 26.5 (2005): 527-532.

[5] H. Ebrahimpour, dan A. Kouzani, "Face Recognition Using Bagging KNN," International Conference on Signal Processing and Communication Systems (ICSPCS'2007) Australia, Gold Coast, 2007.

[6] J.Y. Sari, *Pengenalan wajah pada citra digital menggunakan algoritma Eigenface dan Euclidean distance*, Skripsi, Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari, 2012.

[7] P.N. Belhumeur, J.P. Hespanha, dan D.J. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1997, *19*(7), 711-720.

[8] R.A. Saragih, Pengenalan Wajah Menggunakan Metode Fisherface, *Jurnal Teknik Elektro*, *7*(1), No.1, Maret 2007. Universitas Kristen Maranatha, Bandung.

[9] N.A. Singh, M.B. Kumar, dan M.C. Bala, Face Recognition System based on SURF and LDA Technique. *International Journal of Intelligent Systems and Applications*, 8(2), 13, 2016.

[10] M.C. Thomas dan E.H. Peter, Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 1967, IT-13:21-27.

ISSN 2085-4552

Rancang Bangun Aplikasi Pendeteksi Penyakit Ginjal Kronis dengan Menggunakan Algoritma C4.5

Robi Rianto[1], Ni Made Satvika Iswari[2]
Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
robyryanto@yahoo.com[1], satvika@umn.ac.id2

Diterima 12 April 2017
Disetujui 22 Mei 2017

Abstract*— Kidneys are two bean-shaped organs, each about the size of a fist. They are located just below the rib cage, one on each side of the spine. Kidneys are vital organs contained in the human body and serve to filter the blood of metabolic waste product and throw it in the form of urine. Given the changing conditions in the body can affect the kidneys, causing a decrease in the function of these organ and lead to chronic kidney disease. In an article on the website of the National Institute of Diabetes and Digestive and Kidney Diseases (2014) argued that chronic kidney disease is a silent disease, in which patients appear normal and show no symptomp but the test results stating the patient's kidney function had decreased. Based on the above information, the research on how to detect chronic kidney disease will be conducted. Applications built on the basis of desktop and using Decision Tree algorithm C4.5. The trial was conducted to determine how much the level of accuracy that can be generated by the application. The testing process is done by using cross-validation and based on the results already calculated this application has an accuracy of 91.50% at the time of the decision tree is made without using preprocess menu.

***Index Terms*— C4.5, *chronic kidney disease, decision tree, detection system*.**

I. PENDAHULUAN

Ginjal merupakan salah satu organ penting yang terdapat dalam tubuh manusia dan berfungsi untuk menyaring sampah hasil metabolisme pada darah lalu membuangnya dalam bentuk urin[1]. Namun adanya perubahan kondisi tertentu dalam tubuh dapat mempengaruhi kondisi ginjal sehingga menyebabkan penurunan fungsi organ tersebut dan memicu penyakit ginjal kronis.

Dalam artikel yang terdapat pada *website* milik *National Institute of Diabetes and Digestive and Kidney Diseases* pada tahun 2014 [2] dikatakan bahwa penyakit ginjal kronis merupakan *silent disease*, dimana pasien terlihat normal dan tidak menunjukkan gejala tetapi hasil tes menyatakan ginjal pasien tersebut sudah mengalami penurunan fungsi.

Peran ginjal yang penting bagi tubuh manusia dan sulitnya melakukan deteksi terhadap penyakit ginjal kronis menjadi latar belakang untuk membangun sebuah aplikasi yang dapat melakukan prediksi terhadap penyakit tersebut. Aplikasi dibangun agar penyakit ginjal kronis dapat dideteksi sejak dini dan mencegah penyakit tersebut berkembang sehingga menyebabkan gagal ginjal pada penderita.

Penelitian sebelumnya dilakukan oleh Abdul Rohman[3], yang bertujuan untuk membantu dalam proses pendeteksian penyakit jantung dan menggunakan algoritma C4.5. Algoritma berhasil melakukan prediksi terhadap penyakit jantung, dan menghasilkan akurasi 86.59%.

Penelitian lainnya telah dilakukan Nita Apriliani Puteri, Warih Maharani, dan Mahmud Dwi Suliiyo [4], yang juga dilakukan untuk mendeteksi penyakit jantung, tetapi algoritma yang digunakan adalah *classification and regression tree*. Algoritma tersebut berhasil melakukan prediksi dan menghasilkan nilai akurasi sebesar 77%.

Pada prinsipnya, *decision tree* digunakan untuk memprediksi keanggotaan objek untuk kategori yang berbeda (*class*), dengan mempertimbangkan nilai-nilai yang sesuai dengan atributnya[5]. Algoritma C4.5 adalah algoritma yang berfungsi untuk membuat *Decision Tree* berdasarkan *dataset* yang telah disediakan [6].

Berdasarkan informasi di atas maka penelitian mengenai cara mendeteksi penyakit ginjal kronis akan dilakukan. Aplikasi akan dibangun dengan basis *desktop* dan menggunakan metode *Decision Tree* dengan algoritma C4.5. Pemilihan algoritma tersebut dikarenakan C4.5 dapat menyimpulkan suatu keputusan berdasarkan sebuah *dataset* yang ada dan perbandingan antara penelitian yang sudah ada sebelumnya, hasil yang didapatkan oleh penelitian menggunakan algoritma C4.5 lebih besar dibandingkan menggunakan algoritma *classification and regression tree*. Perbedaan dengan penelitian sebelumnya adalah penelitian ini memprediksi penyakit ginjal kronis menggunakan data yang didapat dari *dataset* dengan metode *Decision Tree* dan algoritma C4.5.

II. DASAR TEORI

A. Ginjal Kronis

Ginjal kronis merupakan akibat akhir dari kehilangan fungsi ginjal lanjut secara bertahap. Kegagalan ginjal kronis terjadi bila ginjal sudah tidak mampu mempertahankan lingkungan internal yang konsisten dengan kehidupan dan pemulihan fungsi tidak dimulai [7].

Seiring bertambahnya usia juga akan diikuti oleh penurunan fungsi ginjal. Hal tersebut terjadi terutama karena pada usia lebih dari 40 tahun akan terjadi proses hilangnya beberapa nefron. Dengan semakin meningkatnya usia dan ditambah dengan penyakit kronis seperti tekanan darah tinggi atau diabetes, ginjal cenderung akan menjadi rusak dan tidak dapat dipulihkan kembali. Selain dari faktor usia, penurunan fungsi ginjal juga dapat disebabkan oleh gaya hidup, gaya hidup tidak banyak bergerak ditambah dengan pola makan buruk yang tinggi lemak dan karbohidrat serta tidak diimbangi serat menyebabkan menumpuknya lemak dengan gejala kelebihan berat badan. Dalam jangka panjang, hal tersebut akan menyebabkan penumpukan lemak dalam lapisan pembuluh darah. Ginjal bergantung pada sirkulasi darah dalam menjalankan fungsinya sebagai pembersih darah dari sampah metabolisme tubuh [8].

B. Data mining

Data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan dan machine learning untuk mengambil dan mengenal informasi yang terdapat pada berbagai *database*. Data mining sering disebut juga sebagai *Knowledge Discovery in Database* (KDD)[9].

C. Decision Tree

Pada prinsipnya, *decision tree* digunakan untuk memprediksi keanggotaan objek untuk kategori yang berbeda (*class*), dengan mempertimbangkan nilai-nilai yang sesuai dengan atributnya. Meskipun *decision tree* tidak begitu luas di bidang *pattern recognition* dari sudut pandang statistik probabilistik pandang, algoritma ini banyak digunakan di domain lainnya seperti, misalnya, obat-obatan (diagnosis), ilmu komputer (struktur data), botani (klasifikasi), psikologi (teori keputusan perilaku), dan sebagainya[5].

Berikut merupakan tahap-tahap dalam pembuatan *Decision Tree* menurut Dr. Saed Sayad[13]:

1. Hitung *entropy* dari target yang ingin dicari dengan rumus ini jika menghitung *entropy* untuk satu *attribute*.

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i \qquad (1)$$

E(S) : Entropy target
Pi : Peluang terjadinya kejadian i

Jika menghitung *entropy* untuk dua *attribute* maka gunakan rumus berikut.

$$E(T,X) = \sum_{c \in X} P(c)E(c) \qquad (2)$$

E(T,X) : Entropy target
P(c) : Peluang terjadinya kejadian c
E(c) : Nilai Entropy c

2. Hitung *Information Gain* dengan rumus berikut :

$$Gain(T.X) = Entropy(T) - Entropy(T,X) \qquad (3)$$

3. Lalu pilih *attribute* dengan nilai *information gain* terbesar untuk dijadikan sebagai *decision node.*
4. Cabang yang mimiliki nilai *entropy* 0 merupakan *leaf node*.
5. Cabang yang memiliki nilai *entropy* lebih dari 0 berarti harus dipecah kembali.

D. Algoritma C4.5

Algoritma C4.5 adalah algoritma yang berfungsi untuk membuat Decision Tree berdasarkan *dataset* yang telah disiapkan. Secara umum algoritma ini digunakan untuk membuat Decision Tree dengan cara sebagai berikut [6]:

1. Pilih *attribute* sebagai akar.
2. Buat cabang untuk masing-masing nilai.
3. Bagi kasus dalam cabang.
4. Ulangi proses untuk masing - masing nilai.
5. Ulangi proses untuk masing - masing cabang sampai semua kasus selesai.

E. Data Preprocessing

Data Preprocessing digunakan untuk meningkatkan kualitas data yang akan dipakai dalam data mining, beberapa faktor yang menentukan tingkat kualitas data yaitu *accuracy*, *completeness*, *consistency*, *timeliness*, *believability*, dan *interpretability*. Beberapa tugas utama yang dilakukan untuk melakukan *data preprocessing* adalah sebagai berikut [10]:

1. **Data Cleaning**
   Dalam proses *data cleaning* pekerjaan yang dilakukan adalah mencoba melakukan pengisian *missing value*, melancarkan *noise*, mengidentifikasi *outlier* dan membenarkan *inconsistencies* dalam *dataset*.
2. **Data Integration**
   Data integration dilakukan dengan tujuan untuk menghindari dan mengurangi *redundancies* dan *inconsistencies* yang terdapat pada *dataset*.
3. **Data Reduction**
   Teknik *data reduction* dapat digunakan untuk mengurangi representasi data yang digunakan, tetapi tetap mempertahankan integrity dari data asli.
4. **Data Transformation dan Data Discretization**

Data transformation dilakukan dengan tujuan untuk membuat hasil dari *data mining* menjadi lebih *efficient* dan pola yang dihasilkan menjadi lebih mudah dimengerti.

F. Cross-Validation

Dalam *k-fold cross-validation* data awal secara acak dibagi menjadi k. Pelatihan dan pengujian dilakukan berulang sebanyak k. Pada iterasi i partisi digunakan sebagai *test set*, dan partisi lainnya secara kolektif digunakan untuk melatih model. Berbeda dengan metode *holdout and random subsampling*, *cross-validation* menggunakan setiap sampel dengan jumlah pengulangan yang sama untuk pelatihan dan sekali untuk *testing*, estimasi akurasi dihasilkan dari jumlah *testing* yang berhasil dibagi dengan total data awal. Secara umum *10-fold cross-validation* dianjurkan untuk memperkirakan akurasi karena bias dan variansi yang relatif rendah [10].

G. Weka Library

Weka library merupakan sebuah koleksi dari algoritma *machine learning* yang digunakan untuk pekerjaan *data mining*. Algoritma yang terdapat dalam *weka library* dapat digunakan langsung dengan menggunakan *software* yang dibuat oleh *weka* atau dapat digunakan dalam pembuatan aplikasi yang berbasis *java*. Dalam *library* tersebut terdapat perlengkapan tentang *data preprocessing*, *classification*, *regression*, *clustering*, *association rules*, dan *visualization* yang dapat membantu dalam pembuatan aplikasi. [11].

H. Dataset

Dalam penelitian ini, digunakan *dataset* yang didapatkan dari *website The UCI Machine Learning Repository*. *Website* tersebut menyediakan koleksi *database*, *domain theories*, dan *data generators* yang banyak digunakan oleh *machine learning community* untuk melakukan sebuah analisa terhadap algoritma *machine learning*[14]. Tabel 1 menyajikan *dataset* yang digunakan dalam penelitian ini beserta singkatan yang digunakan dalam artikel ini untuk kemudahan penulisan.

Tabel 1 *Dataset* sebagai Data Input Algoritma

No.	Singkatan	Data Input
1.	age	Age
2.	bp	Blood Pressure
3.	sg	Specific Gravity
4.	al	Albumin
5.	su	Sugar
6.	rbc	Red Blood Cells
7	pc	Pus Cell
8.	pcc	Pus Cell Clumps
9.	ba	Bacteria
10.	bgr	Blood Glucose Random
11.	bu	Blood Urea
12.	sc	Serum Creatinine
13.	sod	Sodium
14.	pot	Potassium
15.	hemo	Hemoglobin
16.	pcv	Packed Cell Volume
17.	wc	White Blood Cell Count
18.	rc	Red Blood Cell Count
19.	htn	Hypertension
20.	dm	Diabetes Mellitus
21.	cad	Coronary Artery Disease
22.	appet	Appetite
23.	pe	Pedal Edema
24.	ane	Anemia

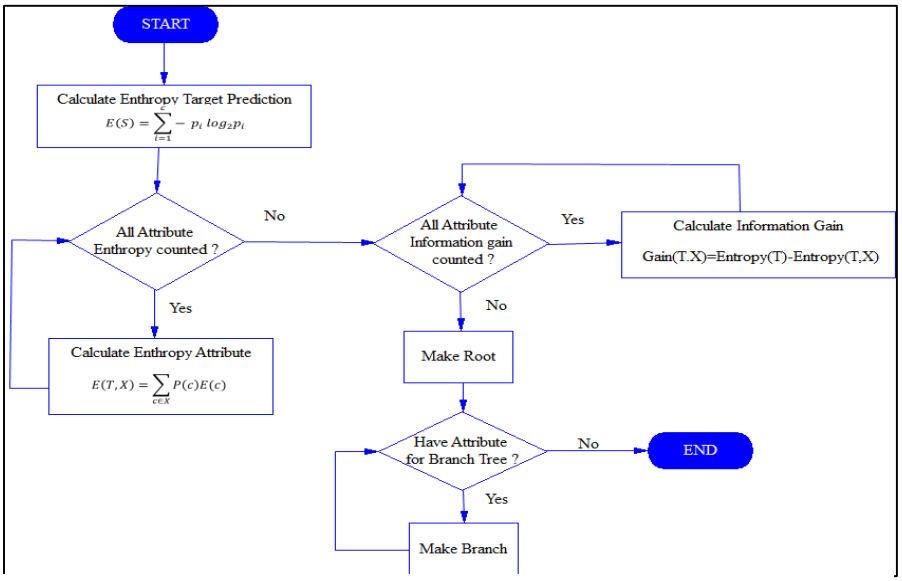
III. METODE PENELITIAN DAN PERANCANGAN APLIKASI

Pada perancangan aplikasi, digunakan *flowchart* untuk merancang fungsi-fungsi dan fitur-fitur pada aplikasi. Rancangan *flowchart* digunakan berdasarkan urutan proses yang dilakukan oleh aplikasi. Secara umum, yang dapat dilakukan pengguna pada aplikasi ini adalah melihat *dataset* yang digunakan untuk pembuatan *decision tree*, melihat *report* dari pembentukan *decision tree* dan melakukan prediksi terhadap penyakit ginjal kronis. Untuk merancang data apa saja yang digunakan pada aplikasi saat dijalankan, digunakan *Data Flow Diagram* pada saat perancangan. *Data Flow Diagram* terdiri dari tiga *level*, yaitu diagram *Context* DFD *level* 1, dan 2.

A. Flowchart

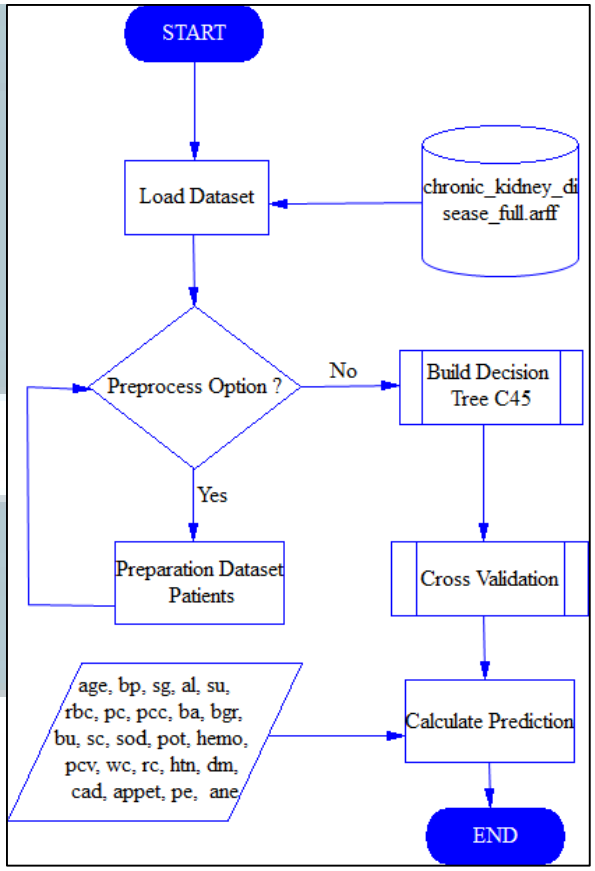
Pada Gambar 1 di bawah aplikasi ini akan digunakan oleh *user* dan pada saat *user* membuka aplikasi maka aplikasi langsung melakukan proses *load data*. Setelah proses tersebut selesai *user* dapat memilih preproses apa saja yang akan digunakan dalam pembuatan *decision tree*¸ saat *dataset* berhasil dibuat oleh aplikasi dengan menggunakan proses pengecekan terhadap *preprocess option*, jika *user* memilih beberapa *preprocess option* maka aplikasi akan menjalakan proses *preparation dataset patients* berdasarkan dari *preprocess option* yang dipilih oleh *user*. Setelah itu proses pengecekan kembali dilakukan, apakah masih ada *preprocess option* yang belum dilakukan, jika masih ada maka proses akan kembali melakukan proses *preparation dataset patients*, jika tidak maka aplikasi siap untuk melakukan proses selanjutnya. Setelah proses *preparation dataset patients* dilakukan maka *user* dapat melihat data-data yang akan digunakan dalam pembuatan *decision tree*.

Proses yang dilakukan berikutnya adalah *build decision tree C4.5* dan *Cross-Validation*. Disaat kedua proses selesai dilakukan, *user* dapat melihat *report decision tree* yang sudah dibuat, *report* tersebut berupa bentuk *decision tree*, *size of tree*, *number of leaf* dan juga akurasi yang dihasilkan oleh proses pengujian *cross-validation*. Proses penghitungan akurasi dilakukan dengan membagi *dataset* menjadi *training set* dan juga *test set*.



Gambar 1 Diagram Alur Proses *Build Decision Tree*

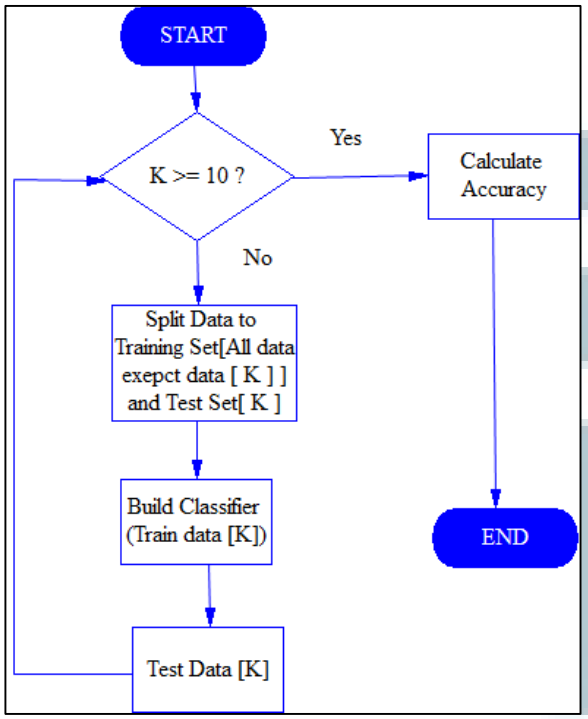
Setelah proses diuji maka aplikasi dapat melakukan penghitungan terhadap *input* yang dimasukan oleh *user*. Hasil yang diberikan oleh proses ini adalah berupa pernyataan apakah data yang dimasukkan tergolong terkena penyakit *chronic kidney disease* atau tidak terkena penyakit tersebut. Pada saat proses ini selesai dilakukan maka *user* dapat tetap melihat *dataset* yang digunakan, *report* dari hasil pembuatan *decision tree*.



Gambar 2 Diagram Alur Aplikasi *Chronic Kidney Disease Detection System*

Pada Gambar 2 menunjukkan proses *build decision tree C4.5* yang dibantu dengan *weka* library. Proses dimulai dari perhitungan *entrophy target prediction* yang merupakan *attribute class* pada *dataset* setelah itu proses dilanjutkan dengan penghitungan *entrophy attribute* yang lain. Setelah sistem mendapatkan seluruh

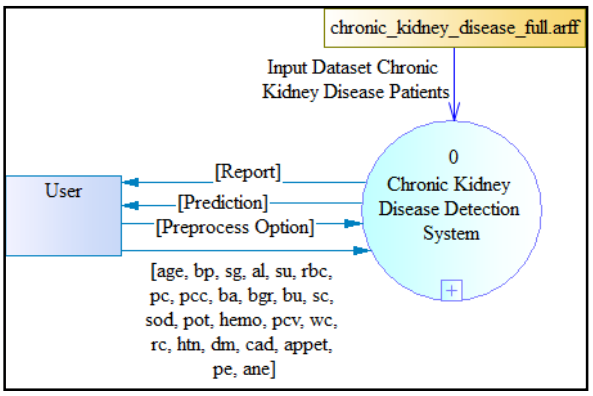
entrophy dari masing-masing *attribute* maka akan dilakukan penghitungan terhadap *information gain* tiap *attribute*. Setelah perhitungan dilakukan maka proses selanjutnya adalah proses pembuatan *root tree* yang diambil berdasarkan *attribute* yang memiliki nilai *information gain* terbesar. Proses selanjutnya adalah pengecekan apakah masih ada *attribute* yang belum menjadi *branch tree*. Jika masih maka proses pembuatan *branch tree* akan dilakukan tapi jika sudah habis maka proses telah selesai membuat *decision tree* dan *rules* dengan algoritma C4.5.



Gambar 3 Diagram Alur Proses *Cross-Validation*

Pada Gambar 3 menunjukkan bahwa proses *cross-validation* dimulai dengan melakukan proses *split data to training set and test set*, dalam proses ini dilakukan pembagian *dataset* menjadi *training set* dan juga *test set*, pembagian ini dilakukan berdasarkan *variable* k. *Dataset* dibagi menjadi 10 bagian lalu *training set* terdiri dari semua bagian *dataset* kecuali *index* k, dan *test set* terdiri dari bagian *dataset* yang memiliki *index* k. Proses selanjutnya adalah membuat *decision tree* berdasarkan dari *training set* yang sudah dibuat pada saat proses sebelumnya. Pembuatan *decision tree* ini dibantu dengan menggunakan *weka library*. Setelah pembuatan itu berhasil selanjutnya diuji dengan menggunakan *test data* yang sudah dibagi sebelumnya. Proses pengujian ini menghasilkan *accuracy* pada setiap putaran K, *accuracy* akhir akan didapatkan saat putaran K mencapai 10 dan hasilnya akan dirata-rata untuk mendapatkan *accuracy* terakhir. Proses perhitungan *accuracy* tersebut dilakukan di proses terakhir yaitu *calculate accuracy*.

B. Data Flow Diagram



Gambar 4 Context Diagram Sistem Pendeteksi Penyakit Ginjal Kronis

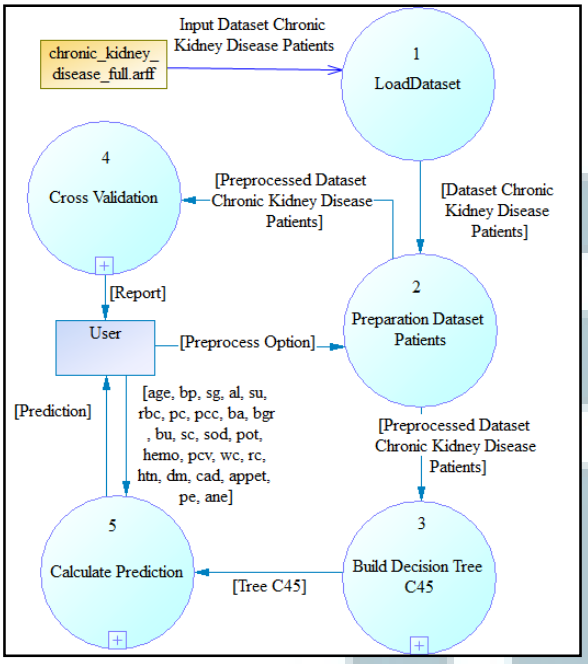
Gambar 4 di atas menunjukkan *Context* Diagram dari Sistem Pendeteksi Penyakit Ginjal Kronis. *Dataset* yang diambil dari arff *file* merupakan *dataset* yang akan digunakan untuk melakukan pembentukan *decision tree* yang dibuat dengan menggunakan *weka library*. *User* memasukkan *input* berupa *preprocess option* yang berfungsi untuk memilih cara-cara preproses yang ingin dilakukan terhadap *dataset*, setelah itu *input* selanjutnya adalah data *user* yang ingin melakukan penghitungan prediksi. Data yang diperlukan age, bp, sg, al, su, rbc, pc, pcc, ba, bgr, bu, sc, sod, pot, hemo, pcv, wc, rc, htn, dm, cad, appet, pe, ane. *Output* yang akan diterima oleh user berupa preproses *report* yang berupa laporan tentang preproses terhadap *dataset* dan juga hasil dari prediksi yang dilakukan oleh *system* dengan *decision tree* yang sudah dibuat dan diuji sebelumnya.

Gambar 5 di bawah menunjukkan *Data Flow Diagram level* 1, pada diagram ini dijelaskan mengenai pemecahan *system* menjadi beberapa proses. Proses awal yang dilakukan oleh *system* adalah membaca data yang terdapat pada arff *file* selanjutnya *system* akan mempesiapkan *dataset* tersebut dengan cara melakukan beberapa preproses yang sudah dipilih oleh *user*. *Input user* tersebut juga terdapat pada gambar di atas yang berupa preproses *option*, preproses yang dapat dipilih oleh *user* ini terbagi menjadi 3 jenis yaitu *remove low information gain*, *replace missing value(mean , mode)*, dan *discretize*.

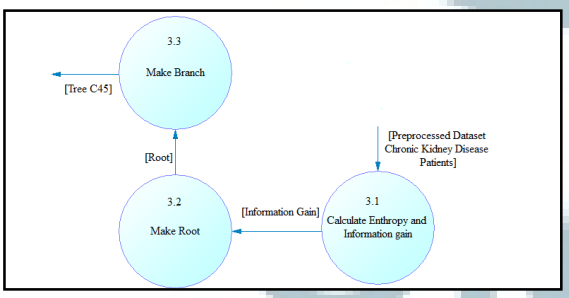
Setelah data siap untuk diproses maka *output* berupa data olahan tersebut dikirimkan ke proses *Cross-validation* agar dapat diuji berapakah akurasi yang dimiliki oleh aturan yang dibuat oleh *decision tree*. *Output* berupa data olahan tadi juga dikirimkan ke proses pembuatan *decision tree* C4.5 untuk dibuat *decision tree* dengan menggunakan bantuan *weka library* dalam perhitungan *information gain* dan juga pembuatan *rules* untuk *decision tree*.

Selanjutnya *tree C4.5* dikirimkan ke proses perhitungan prediksi, disini proses penghitungan

prediksi dilakukan setelah menerima *input* dari user yang berupa data lengkap pasien yaitu age, bp, sg, al, su, rbc, pc, pcc, ba, bgr, bu, sc, sod, pot, hemo, pcv, wc, rc, htn, dm, cad, appet, pe, ane setelah itu *system* akan melakukan penghitungan berdasarkan *rules* yang sudah dibuat sebelumnya dan akhirnya proses tersebut akan memberikan *output* kepada *user* yang berupa hasil dari prediksi penyakit berdasarkan data pasien dan juga *rules* yang dibuat.

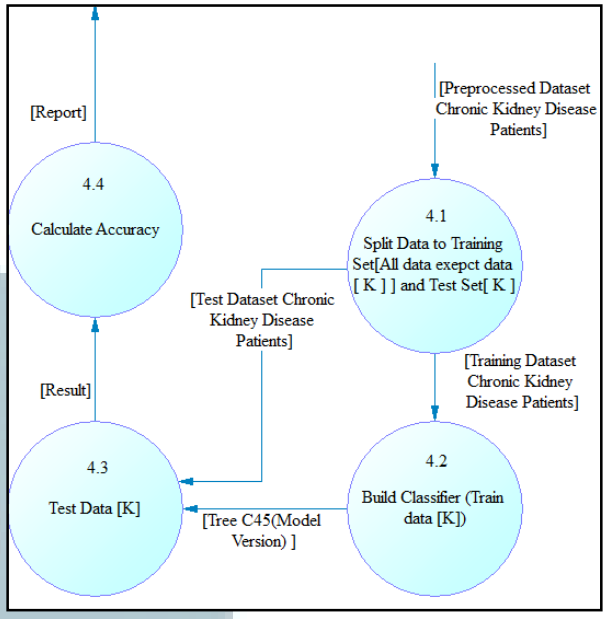


Gambar 5 *Data Flow Diagram Level 1*



Gambar 6 *Data Flow Diagram Build Decision Tree*

Gambar 6 di atas menunjukkan *Data Flow Diagram Build Decision Tree* secara detail, proses pembuatan *decision tree* ini dibuat dengan bantuan *weka library*. Proses ini dimulai dengan penghitungan *enthropy* dan juga *information gain* dari *attribute dataset*, *attribute* ini diambil dari *input* yang berupa *preprocessed dataset chronic kidney disease*. Setelah proses perhitungan berhasil maka nilai *information gain* dikirimkan ke proses selanjutnya, yaitu proses *make root* yang berfungsi untuk memilih *attribute* untuk dijadikan *root* dari *decision tree* yang akan dibuat. Selanjutnya *root* yang telah dibuat dikirimkan ke proses *make branch*, proses ini berfungsi untuk membuat *branch* pada *decision tree* yang dibuat. Proses pembuatan *branch* tersebut terus berulang sampai *attribute dataset* yang terakhir dan tidak memiliki *branch* selanjutnya lagi. Setelah proses ini selesai maka *decision tree* yang terbentuk dikirimkan ke proses selanjutnya.

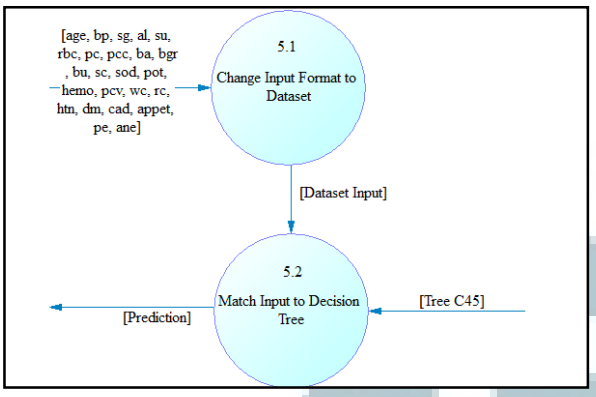


Gambar 7 *Data Flow Diagram Cross-validation*

Gambar 7 di atas menunjukkan *Data Flow Diagram Cross-validation* yang merupakan proses pengujian yang dilakukan terhadap *preprocessed dataset chronic kidney disease*. Proses dimulai dengan membagi *dataset* tersebut menjadi *training data* yang terdiri dari seluruh *dataset* kecuali bagian *dataset* k dan *test data* yang terdiri dari *dataset* bagian k. *Training data* dikirimkan ke proses *build classifier* untuk dijadikan *classifier model* berdasarkan algoritma C4.5. Selanjutnya *test data* akan dikirimkan ke proses *test data [k]* yang berfungsi untuk menguji *decision tree* yang dibuat berdasarkan *train data*, pada proses *test data [k]* ini terdapat *input* data yang dihasilkan oleh proses *build classifier* yang berupa *decision tree*. Setelah proses pengujian telah selesai maka data yang berupa *result* akan dikirimkan ke proses *calculate accuracy* agar proses ini dapat menghitung seberapa besar *accuracy* yang didapatkan dari hasil pengujian tersebut. Setelah itu hasil yang dihasilkan berupa data yang bernama *report* dan dikirimkan ke proses selanjutnya.

Gambar 8 di bawah menunjukkan *Data Flow Diagram Calculate Prediction* yang berawal dengan masuknya *input user* ke proses *change input format to dataset*, proses ini mengubah *input user* yang berupa *string* menjadi *format dataset* yang digunakan oleh *weka library*. Setelah proses itu selesai maka akan dihasilkan *dataset input* yang dikirimkan ke proses *match input to decision tree* yang juga menerima *input*

berupa *decision tree* dari proses sebelumnya. Dalam proses ini penghitungan prediksi dilakukan dengan mencocokkan *input user* kedalam *rules-rules* yang dibuat oleh *decision tree*, setelah proses pencocokan tersebut selesai maka akan dihasilkan sebuah data yang berupa *predicition* dan akan dikirimkan ke *user* dan ditampilkan di layar aplikasi.



Gambar 8 *Data Flow Diagram Calculate Prediction*

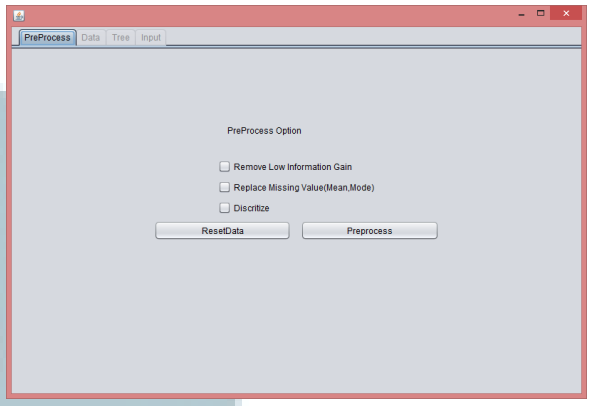
IV. IMPLEMENTASI APLIKASI DAN UJI DETEKSI

A. *Implementasi Algortima dan Metode Preprocess*

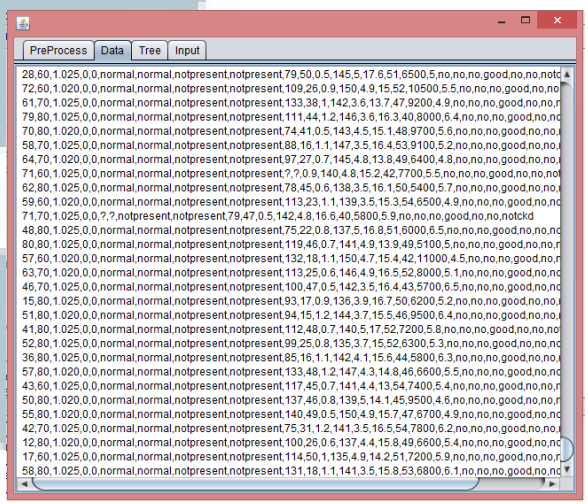
Algoritma C4.5 diimplementasi dengan bantuan dari *weka library*. Yaitu dengan menggunakan *class j48* yang berguna untuk membuat *pruned or unpruned* algoritma *C4.5* [12]. Sebelum proses implementasi dilakukan aplikasi terlebih dahulu melakukan pembacaan terhadap *dataset* yang ingin dijadikan sebagai *training data*.

Untuk dapat meningkatkan kualitas dari *dataset* tersebut maka diperlukan *data preprocessing*. Selain mengisi *missing value* teknik tersebut juga teknik yang akan digunakan dalam penelitian ini adalah *data reduction*, dalam proses ini *data attribute* yang memiliki nilai *information gain* dibawah 0.14 akan dibuang. *Information gain* digunakan sebagai tolak ukur dalam *data reduction* ini adalah dikarenakan setelah melakukan beberapa kali percobaan terhadap aplikasi, penghapusan *data attribute* yang memiliki nilai kecil tidak mempengaruhi *accuracy* yang dihasilkan oleh *decision tree*. Teknik dari *data transformation* yang digunakan pada penelitian ini adalah dengan melakukan *discretize* pada *attribute data* yang memiliki tipe *numeric*. Proses *discretize* dilakukan dengan cara mengubah data yang berbentuk *countinous* menjadi kategori dengan cara mencari *range value* dari data tersebut [10].

Gambar 9 menunjukkan tampilan aplikasi saat *user* membuka aplikasi. Tombol ResetData berguna untuk mengulang proses *data preprocessing* dengan cara membaca ulang *dataset*. Jika pilihan opsi *Remove Low Information Gain* dipilih maka aplikasi akan menseleksi data dengan *information gain* yang memiliki nilai dibawah dari 0.14 akan dibuang dan tidak masuk dalam perhitungan saat pembuatan *decision tree* dimulai. Opsi *replace missing value (mean, mode)* akan berfungsi untuk mengganti seluruh *missing value* yang ada di dalam *dataset* dengan menggunakan *mean* atau *modus* dari *attribute* tersebut. Jika *attribute* memiliki tipe *numeric* maka pengisian *missing value* akan diisi dengan nilai *mean* dari attribute tersebut, jika *attribute* memiliki tipe *nominal* maka pengisian *missing value* akan diisi dengan nilai *modus* dari attribute tersebut. Opsi *discretize* berfungsi untuk mengganti tipe data dari *attribute* yang memiliki tipe *numeric* menjadi *nominal* dengan cara membuat *range* dari nilai *attribute* tersebut.



Gambar 9 Tampilan Awal Aplikasi

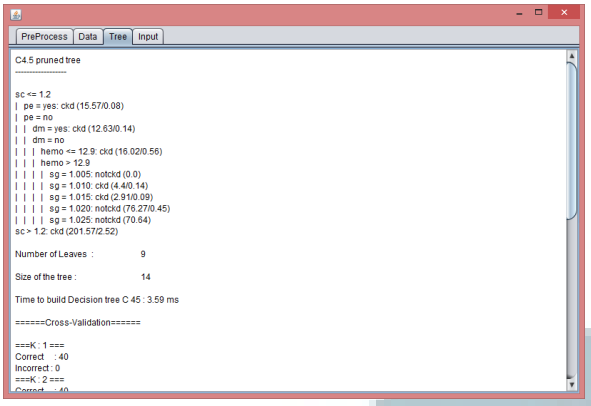


Gambar 10 Tampilan Aplikasi Pada Halaman Data

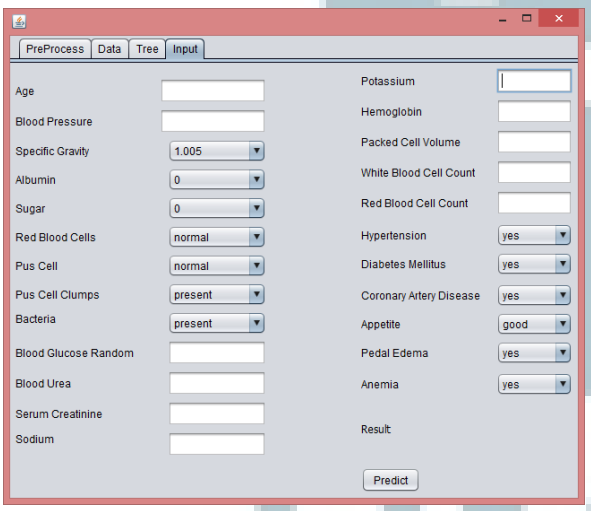
Gambar 10 menunjukkan tampillan aplikasi saat menu *tab* dipilih, pada saat ini aplikasi akan menampilkan seluruh data yang terdapat dalam *dataset* yang sudah dibaca sebelumnya.

Pada Gambar 11, ditampilkan beberapa laporan dari aplikasi terhadap *data preprocessing* yang dilakukan sebelumnya di halaman *preprocessing*, jika *user* tidak memilih opsi apapun saat halaman tersebut maka dalam halaman ini hanya akan ditampilkan struktur *decision tree* yang telah dibuat oleh aplikasi saja. Dalam halaman ini laporan mengenai pengujian *cross-*

validation yang dihitung berdasarkan data prediksi menggunakan *rules* yang dibuat dengan hasil yang sebenarnya yang terdapat di *dataset* ditunjukkan oleh aplikasi.



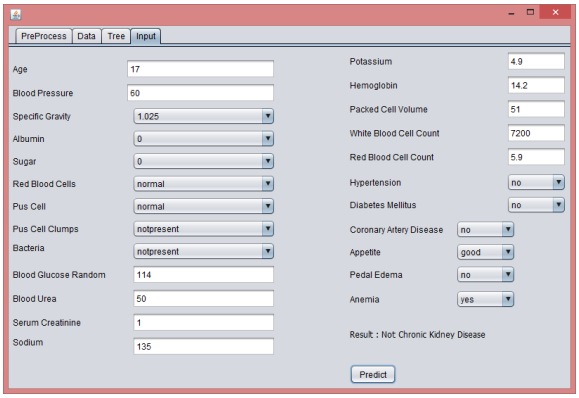
Gambar 11 *Screenshot* Halaman *Tree*



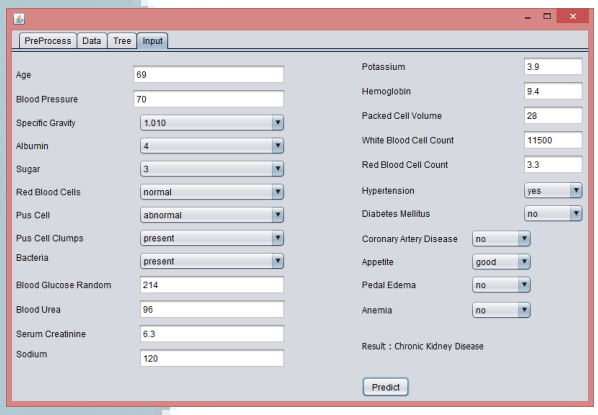
Gambar 12 *Screenshot* Halaman *Input*

Pada Gambar 12 menunjukkan tampilan aplikasi yang digunakan untuk melakukan percobaan deteksi penyakit ginjal kronis dengan menggunakan *decision tree* yang sudah dibuat sebelumnya. Beberapa dari *input* berbentuk *combo box* karena untuk mengisi *attribute* yang bertipe *nominal*, hal ini dikarenakan diperlukan penjagaan agar *user* tidak memasukkan nilai *value* yang tidak seharusnya dimasukkan dalam kotak *input* tersebut.

Pada Gambar 13 menunjukkan tampilan *input* yang sudah menampilkan hasil prediksi. Hasil prediksi pada gambar tersebut menunjukkan bahwa hasil prediksi adalah *Not Chronic Kidney Disease*. Yang berarti data yang dimasukan diprediksikan tidak mengidap penyakit ginjal kronis.



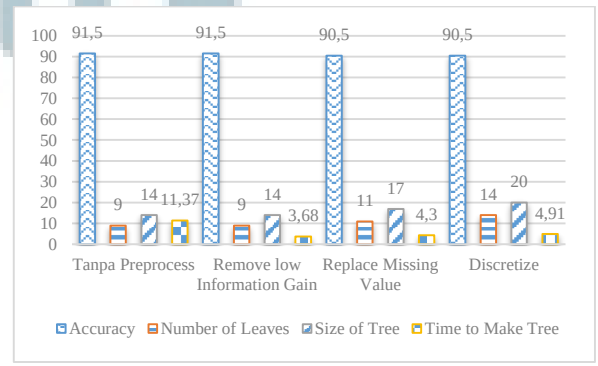
Gambar 13 *Screenshot* Halaman *Input* dengan Hasil Prediksi *Not Chronic Kidney Disease*



Gambar 14 *Screenshot* Halaman *Input* dengan Hasil Prediksi *Chronic Kidney Disease*

Pada Gambar 14 menunjukkan tampilan *input* yang sudah menampilkan hasil prediksi. Hasil prediksi pada gambar tersebut menunjukkan bahwa hasil prediksi adalah *Chronic Kidney Disease*. Yang berarti data yang dimasukan diprediksikan mengidap penyakit ginjal kronis.

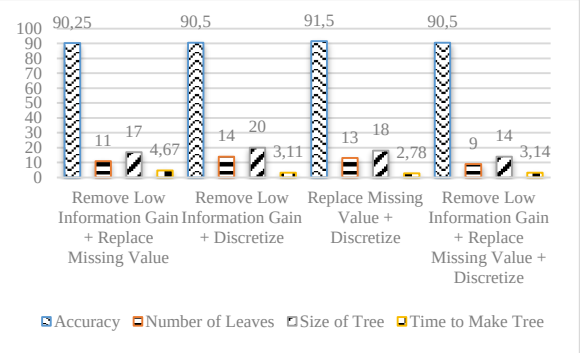
B. Uji Coba Program dan Analisa Preprocess



Gambar 15 *Bar Chart* Pertama Hasil Uji Coba

Gambar 15 menunjukkan tampilan *bar chart* dari hasil uji coba yang telah dilakukan sebelumnya. Pada

tampilan tersebut yang memiliki akurasi terbesar adalah percobaan dengan menggunakan *remove low information gain* dan tanpa menggunakan *preprocess*. Percobaan yang memiliki *number of leaves* dan *size of tree* terkecil adalah tanpa *preprocess* dan *remove low information gain*. Percobaan dengan waktu pembuatan *tree* tercepat adalah *remove low information gain*.



Gambar 16 *Bar Chart* Kedua Hasil Uji Coba

Gambar 16 menunjukkan tampilan *bar chart* dari hasil uji coba menggunakan dua kombinasi metode preproses yang telah dilakukan sebelumnya. Pada tampilan tersebut yang memiliki akurasi terbesar adalah percobaan dengan menggunakan kombinasi *replace missing value* dan *discretize*. Percobaan yang memiliki *number of leaves* dan *size of tree* terkecil adalah kombinasi *remove low information gain*, *replace missing value*, dan *discretize*. Percobaan dengan waktu pembuatan *tree* tercepat adalah kombinasi *replace missing value* dan *discretize*.

V. SIMPULAN

Rancang bangun aplikasi pendeteksi penyakit ginjal kronis dengan menggunakan algoritma C4.5 telah berhasil dibuat, dengan bantuan dari *weka library* dalam pembuatan algoritma *decision tree* yang dihasilkan dapat melakukan prediksi terhadap penyakit ginjal kronis. Uji coba dilakukan untuk mengetahui seberapa besar tingkat akurasi yang dapat dihasilkan oleh aplikasi. Proses pengujian dilakukan dengan menggunakan *cross-validation* dan berdasarkan hasil yang sudah dihitung aplikasi ini memiliki akurasi 91.50% pada saat *decision tree* dibuat tanpa menggunakan *preprocess menu*. Perkembangan yang didapatkan dalam melakukan preproses adalah pengurangan waktu yang diperlukan untuk membuat *decision tree*. Akurasi yang didapatkan setelah preproses tidak ada yang memiliki peningkatan. Ukuran *decision tree* bertambah besar jika menggunakan preproses *discretize* dan *replace missing value*. Metode yang meningkatkan cepatnya pembuatan *decision tree* adalah *remove low information gain*, *replace missing value*, dan *discretize*. Metode yang paling baik digunakan berdasarkan percobaan yang dilakukan adalah *remove low information gain*, yang menghasilkan akurasi 91.50 % dengan ukuran *tree* dan waktu pembuatan *tree* yang minimum.

DAFTAR PUSTAKA

[1] Aswano. 2014. *MAKALAH GAGAL GINJAL KRONIK*. Tersedia dalam: http://www.slideshare.net/septianraha/makalah-gagal-ginjal-kronik-37197034. Diakses 2 Maret 2016.

[2] Kidney Disease Statistics for the United States. 2016. National Institute of Diabetes and Digestive and Kidney Diseases. [Online]. Avalaible: https://www.niddk.nih.gov/health-information/health-statistics/kidney-disease

[3] Rohman, A. 2013. *Penerapan Algoritma C4.5 Berbasis Adaboost untuk Prediksi Penyakit Jantung*. Majalah Ilmiah Universitas Pandanaran Vol 11. No. 26.

[4] Puteri, N. A., Maharani, W., Suliiyo, M. D. 2013. *Prediksi Penyakit Jantung dengan Algoritma Classification and Regression Tree*. Telkom University

[5] Gorunescu, F. 2011. Data Mining: Concepts, Models and Techniques.

[6] Kusrini. 2007. Konsep dan Aplikasi Sistem Pendukung Keputusan. Yogyakarta: Penerbit Andi.

[7] Sangadah, N. dan Tri, S. P. R. 2012. *MAKALAH GAGAL GINJAL AKUT DAN KRONIS*. Tersedia dalam: http://www.scribd.com/doc/93094328/MAKALAH-GAGAL-GINJAL-KRONIS#scribd. Diakses 2 Maret 2016.

[8] Aisyah, J. 2011. *Karakteristik Penderita Gagal Ginjal Rawat Inap Di RS Haji Medan Tahun 2009* Tersedia dalam: http://repository.usu.ac.id/handle/123456789/24681. Diakses 2 Maret 2016.

[9] Ridwan, M., Suryono, H., Sarosa, M. 2013. Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naïve Bayes Classifier.

[10] Han J., Kamber M., Pei J. 2012. Data Mining: Concepts and Techniques 3rd Edition.

[11] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[12] Dr. Bhargava, N., Dr. Bhargava, R., Mathuria, M. 2013. *International Journal of Advanced Research in Computer Science and Software Engineering*..

[13] Sayad, S. 2010. *Decision Trees: Classification & Regression*. University of Toronto.

[14] Lichman, M. 2013. *{UCI} Machine Learning Repository*. [Online]. Available : http://archive.ics.uci.edu/ml

ISSN 2085-4552

Rancang Bangun Aplikasi Chatbot Sebagai Media Pencarian Informasi Anime Menggunakan Regular Expression Pattern Matching

David Domarco[1], Ni Made Satvika Iswari[2]
Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
david.domarco@student.umn.ac.id[1], satvika@umn.ac.id[2]

Diterima 12 April 2017
Disetujui 16 Mei 2017

***Abstract*— Technology development has affected many areas of life, especially the entertainment field. One of the fastest growing entertainment industry is anime. Anime has evolved as a trend and a hobby, especially for the population in the regions of Asia. The number of anime fans grow every year and trying to dig up as much information about their favorite anime. Therefore, a chatbot application was developed in this study as anime information retrieval media using regular expression pattern matching method. This application is intended to facilitate the anime fans in searching for information about the anime they like. By using this application, user can gain a convenience and interactive anime data retrieval that can't be found when searching for information via search engines. Chatbot application has successfully met the standards of information retrieval engine with a very good results, the value of 72% precision and 100% recall showing the harmonic mean of 83.7%. As the application of hedonic, chatbot already influencing Behavioral Intention to Use by 83% and Immersion by 82%.**

***Index Terms*—anime, chatbot, information retrieval, Natural Language Processing (NLP), Regular Expression Pattern Matching**

I. PENDAHULUAN

Memasuki awal abad ke-20, seniman grafis Jepang mulai merasakan pengaruh dari perkembangan teknologi Barat, yaitu film digital. Perubahan tersebut merupakan pemicu lahirnya sebuah seni modern Jepang, yang dikenal dunia dengan nama *anime* atau film animasi Jepang. *Anime* sendiri merupakan film animasi dengan teknik penggambaran yang melibatkan emosi dari setiap karakter dengan alur yang kompleks. Berdasarkan rekap data distribusi *anime* yang dikeluarkan oleh AJA[1], distribusi *anime* di internet berkembang secara pesat setiap tahunnya. Hal ini dipicu oleh bertambah besarnya jumlah penggemar *anime* yang mengekpresikan ketertarikan mereka dengan mencari segala informasi yang berhubungan dengan kegemaran mereka.

Berdasarkan survey yang dilakukan Shawar, Atwell, dan Roberts[2] *chatbot* telah sukses dikembangkan dan diterapkan pada bidang edukasi, pencarian informasi, bisnis, dan *e-commerce*. Banyak *chatbot anime* yang telah beredar di internet baik chatbot yang dibuat oleh profesional maupun kaum awam. Namun, belum ada aplikasi *chatbot anime* yang dibangun untuk dapat menangani permintaan pengguna dalam mencari informasi spesifik dari seluruh film *anime* yang ada. Pada penelitian ini dibangun sebuah aplikasi *chatbot* yang dikhususkan sebagai sebuah media pencarian informasi seputar film *anime* yang diberi nama Reikobot. Reikobot dibangun sebagai sebuah fasilitas untuk memudahkan penggemar *anime* dalam berinterkasi dan mencari informasi seputar film *anime* selayaknya bercakap dengan manusia secara natural.

Saat ini, sudah banyak dikembangkan aplikasi *chatter-boot*, atau dikenal sebagai *chatbot* untuk berbagai tujuan tertentu, atau hanya untuk media hiburan[3]. Aplikasi *chatbot* akan mencocokan kalimat input dari pengguna dengan *pattern* yang ada pada *knowledge* yang sudah disimpan sebelumnya. *Knowledge* didapatkan dari berbagai sumber dan menjadi bahan dari pola *chat* antara pengguna dengan *chatbot*[4]. Dalam penelitian ini dapat dilihat seberapa besar pengaruh *chatbot* dalam memberikan informasi yang relevan bagi pengguna serta rasa nyaman selama dalam menggunakan aplikasi.

II. PENELITIAN TERKAIT

Sebuah *chatbot* pencarian informasi yang telah berhasil dibangun adalah FAQchat[2], yang digunakan sebagai sebuah aplikasi *question answering* seputar Frequently Ask Question (FAQ) di sebuah *School of Computing* (SoC), University of Leeds. FAQchat adalah sebuah *chatbot* pencarian informasi yang dikembangkan dengan menggunakan AIML (*Artificial Intelligence Markup Language*) *pattern matching* yang mengambil informasi dari *databae* FAQ (*Frequently Ask Question*). FAQchat

memperoleh informasi menggunakan kata yang signifikan sebagai kata kunci, kemudian mencoba mencari *pattern* terpanjang yang sesuai tanpa menggunakan *linguistic tools* atau menganalisa makna dari *query* pengguna. FAQchat tidak membutuhkan modul pengetahuan *linguistic* dan menerapkan prinsip *language independent*. Proses kerja FAQchat adalah sebagai berikut [2]:

1. Semua pertanyaan dan jawaban diekstrak dari *database* setelah melakukan process *filtering* untuk menghilangkan *tag* yang tidak dibutuhkan.
2. *Database* FAQ tersusun atas pertanyaan dan jawaban. Dengan menggunakan pola tersebut, sebuah daftar *link* dibangun, yang berisi *link* dari FAQ ke halaman web yang berisi jawaban.
3. Disusunlah sebuah kamus yang berisikan semua kata dalam pertanyaan dengan frekuensi kejadian. Kemudian, kata pertama atau kedua yang paling signifikan diambil dari setiap pertanyaan yang telah diajukan.
4. AIML *patern matching rules*, atau dikenal sebagai "kategori" dibuat. Terdapat dua tipe *pattern matching* pada FAQchat, yaitu seluruh pertanyaan, atau hanya kata pertama atau kedua yang paling signifikan yang mengalami kecocokan dengan data pertanyaan pada *database* FAQ. Terdapat dua kemungkinan respon yang diberikan FAQchat, yaitu sebuah jawaban jika terdapat satu pertanyaan yang cocok atau beberapa jawaban jika kata pertama atau kedua yang paling signifikan ditemukan dalam beberapa pertanyaan.

III. KOMPONEN CHATBOT

A. *Part of Speech Tagging*

Part of Speech Tagging (POS *Tagging)* merupakan sebuah proses yang bekerja untuk mengidentifikasi tata bahasa pada suatu kalimat [5]. POS *Tagging* digunakan untuk mengelompokkan kata-kata yang terdapat dalam kalimat natural guna memahami kategori kata di dalam suatu kalimat. Sebagai contoh berdasarkan tata bahasa natural, kalimat "*What is rezero anime*?" tersusun atas empat kata yaitu "*what*", "*is*", "*rezero*" dan "*anime*". Pada proses *tagging* ketiga kata tersebut akan diidentifikasi menjadi "*what"* sebagai WH *Word*, "*is"* sebagai *verb*, dan "*rezero anime*" sebagai *noun*. WH *Word* merupakan kata pertanyaan untuk menanyakan pertanyaan, seperti *what, where, when, who, whom, why*, dan *how*. Disebut sebagai WH *Word* karena dalam Bahasa Inggris, kebanyakan kata tersebut dimulai dengan huruf *wh*. *Noun* merupakan objek dari suatu perbincangan di dalam kalimat sehingga berdasarkan pengelompokkan di atas, sistem dapat mengenali bahwa topik yang terdapat di dalam bahasa natural tersebut adalah "*rezero anime"*. Topik yang didapat dapat digunakan sebagai *keywords* dalam pencarian judul film *anime*.

Aplikasi *chatbot* pencarian informasi *anime* (Reikobot) menggunakan Brill Tagger, yaitu sebuah algoritma POS *Tagging* yang menggunakan metode induksi berbasis transformasi untuk memahami tata bahasa natural[6]. Teknik pemberian *tag* ini menggunakan seperangkat aturan yang telah ditetapkan sebelumnnya dalam bentuk sebuah kumpulan *vocabulary*. Jika kata diketahui, maka kata akan diberikan *tag* sesuai dengan sebuah kamus *vocabulary* yang telah disiapkan. Jika kata tersebut tidak diketahui, maka secara naif kata tersebut akan diberikan *tag noun* sedangkan jika kata tersebut diketahui, maka kata tersebut akan diberikan *tag* sesuai *record* pada kamus *vocabulary*. Kemudian, kata tersebut akan memasuki seperangkat *if rules* yang secara bertahap akan mengubah *tag* awal menjadi *tag* lain yang lebih akurat bila kata tersebut memenuhi kondisi yang ditentukan pada suatu *rules*.

Gambar 1 menampilkan contoh *pseudocode* dari penerapan metode Bril *tagger*.

```
FOR each word as word
    IF the word IS SET IN dictionary THEN
        tag word WITH dictionary tag
    ELSE
        tag word AS noun
    END IF.

    IF word tag AS verb AND the word before IS 'the' THEN
        change word tag TO noun
    END IF.

    IF the word before IS 'would'
        chage word tag TO verb
    END IF.

    ..... // apply any if rules

    IF the word END WITH 'ly'
        chage word tag TO adverb
    END IF.

    // and so on
END FOR.
```

Gambar 1. Pseudocode Brill Tagger

B. *Regular Expression Pattern Matching*

Menurut Navarro dan Raffinot [7] *regular expression* memberikan solusi yang sangat kuat dalam mengekpresikan sederet pencarian pattern. *Regular Expression Pattern Matching* menggunakan sekumpulan *regular expression* yang disusun menjadi sebuah *pattern*. Bila kalimat masukan cocok dengan salah satu *pattern* yang ada, maka sistem akan melakukan proses sesuai perintah yang ditentukan pada *pattern* tersebut. Gambar 2 menampilkan contoh *pattern* menggunakan *regular expression* dalam pencarian informasi *anime*.

```
$patterns = array(
    'desc' => "/((description)|(summar)|(info)|(story)|(synop))/", // desc
    'type' => "/( ) *((type)|(series)|(movie)|(ova)|(music))/", // type
    'eps' => "/((episode)|(chapter)|(long)|(day))/", // episode
    'stat' => "/((status))/", // status
    'date' => "/((date)|(month))/", // aired
    'season' => "/((season)|(start)|(aired)|(end)|(period)|(year))/", // season
    'studio' => "/((studio)|(make)|(produce))/", // studio
    'source' => "/(source)|(adapt)/", // source
    'genre' => "/((genre))/", // genre
    'rank' => "/((rank))/", // ranked
    'pop' => "/((popular))/", // popularity
    'url' => "/((link).*((anime)|()))|((url).*((anime)|()))/", // link .... anim
    //
    );
```

Gambar 2. *Regular Expression Pattern*

C. *Precision, Recall, dan Harmonic Mean*

Menurut Manning dkk. [5] dalam bukunya yang berjudul "*Introduction to Information Retrieval*" unsur utama yang diukur dalam sistem pencarian informasi adalah *precision* dan *recall*. *Precision* merupakan seberapa banyak jumlah dokumen relevan yang didapatkan dari seluruh dokumen yang berhasil diambil. *Recall* mendefnisikan seberapa banyak sistem mengembalikan dokumen yang relevan dari seluruh dokumen relevan yang tersedia.

Tabel 1 Tabel Kontingensi

	Relevant	***Non Relevant***
Retrieved	*true positive (tp)*	*false positive (fp)*
Non Retrieved	*false negative (fn)*	*true negative (tn)*

Tabel 1 menunjukan beberapa kondisi yang perlu diperhitungkan dalam *information retrieval*. Kondisi *true positive* (*tp*) merupakan jumlah dokumen relevan yang terambil. Kondisi *false positive* (*fp*) merupakan jumlah dokumen tidak relevan yang terambil. Kondisi *false negative* (*fn*) merupakan jumlah dokumen relevan yang tidak ditemukan. Sementara kondisi *true negative* (*tn*) merupakan jumlah dokumen tidak relevan yang tidak ditemukan.

Berdasarkan Tabel Kontingensi (Tabel 1) gagasan *precision* dan *recall* dapat ditafsirkan menjadi sebuah persamaan berikut [5]:

$$\text{Precision} = \frac{\#(relevant\ items\ retrieved)}{\#(retrieved\ items)} = \left(\frac{tp}{tp+fp}\right) x\ 100\% \quad (1)$$

$$\text{Recall} = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} = \left(\frac{tp}{tp+fn}\right) x\ 100\% \quad (2)$$

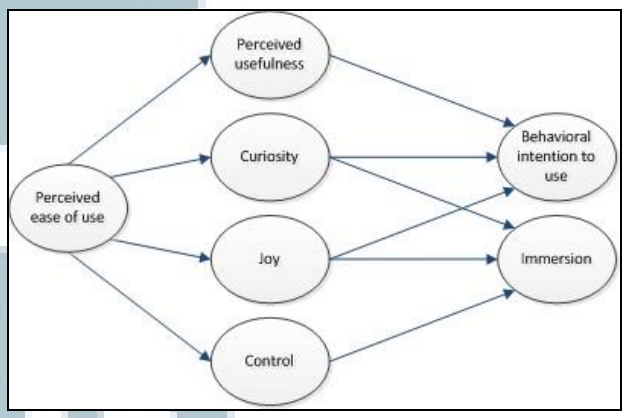
Oleh karena *precision* dan *recall* merupakan unsur yang saling berlawanan (*trade-off*), dalam mengevaluasi toleransi *trade-off* tersebut digunakanlah penghitungan F-Measure yang merupakan *weighted harmonic mean* dari *precision* dan *recall*. Nilai minimum *harmonic mean* yang harus dicapai adalah sebesar 70% dikarenakan pada titik tersebut nilai dari presisi dan *recall* mencapai keseimbangan *trade-off* [11].

$$\text{F\_Measure} = \frac{\#(2PR)}{P+R} \quad (3)$$

IV. HEDONIC MOTIVATION SYSTEM ADOPTION MODEL (HMSAM)

HMSAM [8] adalah sebuah model untuk mengukur sebuah sistem yang mengadaptasi motivasi hedonis. Ada lima faktor yang menjadi fokus pengukuran pada HMSAM, yaitu *perceived usefulness*, *perceived ease of use*, *curiosity*, *control*, dan *joy* yang akan mempengaruhi *behavioral intentional to use* dan *immersion* suatu aplikasi. Aspek-aspek tersebut adalah sebagai berikut:

- *Perceived usefulness*, mengukur peningkatan kinerja ketika menggunakan suatu sistem.
- *Perceived ease of use*, mengukur kemudahan penggunaan suatu sistem.
- *Curiosity*, sejauh mana suatu sistem dapat meningkatkan rasa ingin tahu dalam aspek kognitif.
- *Control*, persepsi pengguna bahwa dirinya yang diajak berinteraksi oleh sistem.
- *Joy*, aspek kesenangan yang didapatkan dari interaksi dengan sistem.
- *Behavioral intentional to use*, keinginan pengguna untuk menggunakan aplikasi.
- *Focussed Immersion*, total keterlibatan yang dilakukan antara pengguna dengan aplikasi yang mengukur seberapa dalam pengguna terfokus dalam menggunakan sistem.



Gambar 3. *Overview of* HMSAM [3]

V. PERANCANGAN SISTEM

Penelitian ini membuat sebuah rancang bangun sistem chatbot sebagai media pencarian informasi *anime* yang ditulis dalam bahasa PHP. Sistem chatbot ini diberi nama Reikobot.

A. *Peancangan Sistem*

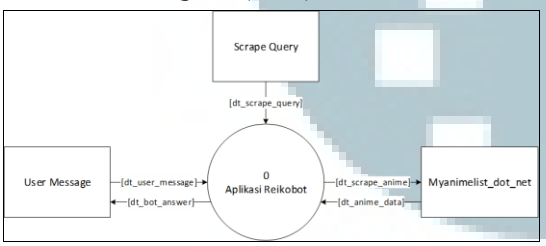
Desain rancangan sistem Reikobot adalah sebagai berikut.

1. Kalimat natural atau pertanyaan yang dimasukkan pengguna di pecah menjadi sekumpulan *array* kata, kemudian dilakukan transformasi (normalisasi, *stop word* dan *strip punctuation*)

terhadap setiap kata sehingga didapatkan kata dalam bentuk kata dasar. Sesudah itu, dibentuklah kalimat baru (kalimat sederhana) dengan unsur kata yang telah disederhanakan.

2. Dilakukan teknik *tagging* pada kalimat sederhana sehingga setiap kata mempunyai sebuah *tag grammar*. Lalu, dilakukan penyeleksian berdasarkan *tag* agar tercipta sebuah kalimat baru (kalimat *tag*) yang didalamnnya hanya berisi kata dengan *tag* yang bermakna.
3. Sistem memiliki beberapa topik yang dikelompokkan ke dalam *pattern* tertentu yaitu *pattern* untuk pertanyaan terbuka dan *pattern* untuk pertanyaan khusus atau spesifik. Kalimat *tag* akan dicocokkan dengan *pattern* sehingga sistem dapat mengkategorikan jenis pertanyaan yang terkandung di dalam kalimat *tag*.
4. Jenis pertanyaan yang telah dikategorikan akan diproses sesuai perintah yang diprogram di dalam sistem dan akan mengebalikkan keluaran berupa jawaban yang memenuhi kriteria. Jika kriteria tidak ditemukan, maka sistem akan memberikan jawaban seperti "*Sorry dear, Nee-san can't found the data you are looking for*".

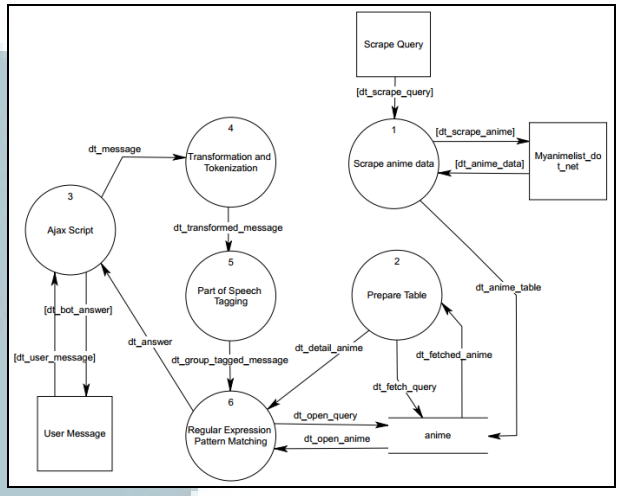
B. *Data Flow Diagram (DFD)*



Gambar 4. Diagram Konteks Aplikasi Reikobot

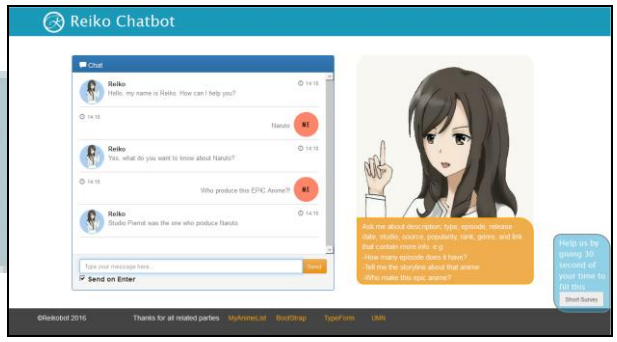
Gambar 4 menunjukan Diagram Konteks dari Aplikasi Reikobot yang dibangun. Sementara Gambar 5 menampilkan alur data tingkat satu dari sistem *chatbot* pencarian informasi *anime*. Terdapat lima buah proses dan dua buah entitas di dalam diagram alur data. Alur data `dt_scrape_query` dan `dt_anime_data` berisi *query* dengah hasil berupa seluruh data dan informasi mengenai film *anime* yang berasal dari entitas `Myanimelist_dot_net` dan didapatkan melalui proses *scrape anime data*. Data dan informasi *anime* yang berhasil diambil disimpan ke dalam tabel *anime*. Data `dt_message` dari entitas user diterima oleh proses *ajax script*, kemudian message tersebut diolah menjadi data `dt_ajax_post` yang akan melewati tiga buah proses *natural language processing*, yaitu *transformation and tokenization*, *part of speech tagging*, dan *regular expression pattern matching* yang masing-masing menghasilkan data berupa `dt_transformed_message`,`dt_group_tagged_message`, dan `dt_json_encode`.

Proses *regular expression pattern matching* membutuhkan data `dt_detail_anime` dan `dt_open_anime` yang akan diolah menjadi sebuah keluaran dalam bentuk JSON (JavaScript Object Notation). Data `dt_detail_anime` merupakan data yang berisi informasi anime terkait pertanyaan tertutup sedangkan `dt_open_anime` merupakan data hasil *query* yang berisi informasi *anime* terkait pertanyaan terbuka. Setelah melewati ke tiga buah proses tersebut, `dt_json_encode` yang dikirimkan kembali ke proses *ajax script* akan diolah menjadi sebuah jawaban bernama `dt_answer`. Lalu, data `dt_answer` diterima oleh entitas *user*.



Gambar 5. DFD Level 1

VI. IMPLEMENTASI



Gambar 6. Hasil Implementasi Reikobot

Setelah berhasil diimplementasikan aplikasi *chatbot* pencarian infromasi *anime* (Reikobot) memilki tampilan seperti pada Gambar 6. Tampilan antar muka Reikobot memiliki dua buah elemen utama yaitu *chat box* yang terletak di sisi kiri dan *instruction box* yang terletak di sisi kanan. *Chat box* adalah tempat pengguna beriteraksi dengan aplikasi Reikobot. *Instruction box* berisi *hint* yang membantu pengguna untuk berinteraksi dengan aplikasi.

Reikobot. Selain kedua elemen utama tersebut terdapat elemen *online survey* yang terletak di pojok kanan bawah dari tampilan.

VII. Evaluasi Sistem

Setelah aplikasi *chatbot* pencarian informasi *anime* (Reikobot) telah selesai diimplementasikan tahap selanjutnya adalah pengujian terhadap sistem. Terdapat dua buah pengujian yang dilakukan yaitu evaluasi *chatbot* sebagai mesin pencarian informasi dan evaluasi *chatbot* sebagai sebuah aplikasi hedonis.

A. *Evaluasi Chatbot sebagai Mesin Pencarian Informasi*

Berdasarkan *rule of thumb* dari pencarian informasi, jumlah minimum pencarian data dalam suatu tes adalah 50 buah [5]. Relevansi dari dokumen ditentukan oleh satu orang *judge* sebagai titik pengukuran dengan mengikuti standar pengukuran *unraked retrieval sets*. Pengukuran menggunakan nilai biner, yaitu nilai "0" untuk menandakan dokumen yang tidak relevan dan nilai "1" untuk menandakan dokumen yang relevan. Berikut merupakan penghitungan *Recall* dan *Precision* sesuai hasil tes relevansi yang dinotasikan pada Tabel 1.

Tabel 2. Notasi Hasil Tes Relevansi

	Relevant	***Non Relevant***
Retrieved	36	14
Non Retrieved	0	0

Berdasarkan Tabel 2, dilakukan penghitungan *precision*, *recall*, dan *F-Measure*. Dari hasil penghitungan didapatkan *precision* sebesar 72%, *recall* sebesar 100%. Niali *harmonic mean* yang tercapai adalah sebesar 83.7%.

B. *Evaluasi Chatbot dengan Metode HMSAM*

Chatbot merupakan aplikasi *entertainment* atau hiburan yang mengarah pada aktivitas dengan unsur intrinsik di dalamnya sehingga *chatbot* dikategorikan sebagai sebuah sistem hedonis. Menurut Lowry dkk. [5] teknik evaluasi yang digunakan untuk menguji sistem hedonis adalah dengan menggunakan aspek-aspek dalam *Hedonic Motivation System Adoption Model* yang disingkat sebagai HMSAM.

Menurut Gay and Diehl [9] jumlah minimum *sample* yang diperlukan dalam melakukan suatu penelitian adalah sebanyak 30 buah. Melalui kuisoner *online* yang disebarkan didapatkan 33 orang responden. Pertanyaan yang diajukan kepada 33 responden ini adalah lima buah pertanyaan mengenai *perceived ease of use, perceived usefulness, curiosity, control*, dan *joy* yang akan mempengaruhi *behavioral intention to use* dan *immersion* suatu aplikasi hedonis. Respon yang diberikan oleh ke tiga puluh tiga responden dapat dilihat pada Tabel 3.

Tabel 3 Respon Kuisoner HMSAM

#	***Strongly Disagree***	***Dis-agree***	***Nei-ther***	***Agree***	***Strongly Angree***	***Aspek HMSAM***
1	0	1	1	19	12	*Perceived Ease of Use*
2	0	1	1	22	9	*Perceived Usefulness*
3	1	1	7	12	12	*Curiosity*
4	1	1	5	8	18	*Control*
5	0	2	7	15	9	*Joy*

Berdasarkan Tabel 3, dilakukan penghitungan Likert scale [10] yang mewakili aspek-aspek HMSAM. Hasil dari penghitungan disajikan pada Tabel 4.

Aspek *Behavioral Intention to Use* dan *Immersion* dihitung menggunakan rata-rata aritmatika dan didapatkan Behavioral Intention to Use sebesar 83% serta *Immersion* sebesar 82%.

Tabel 4 Hasil Penghitungan Likert Scale HMSAM

Aspek HMSAM	**Penghitungan Likert**
Perceived Ease of Use	0.854545 (*Strongly Agree*)
Perceived Usefulness	0.836364 (*Strongly Agree*)
Curiosity	0.8 (*Strongly Agree*)
Joy	0.848485 (*Strongly Agree*)
Control	0.787879 (*Agree*)
Behavioral Intention to Use	0.834848 (*Strongly Agree*)
Immersion	0.81812 (*Strongly Agree*)

VIII. Simpulan

Aplikasi chatbot sebagai media interaktif dalam mendapatkan informasi seputar *anime* berbasis teks menggunakan *regular expression pattern matching* telah berhasil dirancang dan dibangun dengan nama Reikobot. Aplikasi Reikobot menghasilkan nilai *precision* sebesar 72% dan *recall* sebesar 100% yang menghasilkan nilai *harmonic mean* sebesar 83,7%. Adapun nilai *harmonic mean* tersebut bermakna bahwa aplikasi dapat menyajikan informasi dengan *precision* dan *recall* yang harmonis atau seimbang dengan bobot yang tidak terlalu jauh berbeda.

Berdasarkan penghitungan Likert, aplikasi Reikobot menghasilkan tingkat Behavioral intention to use (BIU) sebesar 83% yang berarti pengguna sangat setuju bahwa aplikasi meningkatkan minat pengguna dalam pencarian informasi anime dan menghasilkan tingkat Immersion sebesar 82% yang berarti pengguna sangat terfokus ketika menggunakan aplikasi.

Daftar Pustaka

[1] The Association of Japanese Animation. March 2015. *Anime Industry 2014 Summary*. The Association of Japanese Animation, Database Working Group, Akihabara.

[2] Shawar, B. A., Atwell, E; dan Roberts, A. 2005. FAQchat as in Information Retrieval system. Web: eprints.whiterose.ac.uk/4663, diakses pada 4 April 2016.

[3] Augello, A., Pilato, G., Machi, A., dan Gaglio, S. 2012. *An Approach to Enhance Chatbot Semantic Power and Maintainability: Experiences Within The FRASI Project*.

Proc. Of 2012 IEEE Sixth International Conference on Semantic Computing.

[4] Setiaji, B. dan Wibowo, F.W. 2016. *Chatbot Using A Knowledge in Database*. Proc. Of 2016 7th International Conference on Intelligent System, Modelling and Simulation.

[5] Manning, C. D., Raghavan, P., dan Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

[6] Brill, Eric. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. Computational Linguistics, December 1995.

[7] Navarro, G., dan Raffinot M. 2002. Flexible Pattern Matching in String: Practical On-Line Search Algorithms for Texts and Biological Sequence. Cambridge University Pers.

[8] Lowry, P. B. dkk. 2013. *Taking 'fun and games' seriously: Proposing the hedonic-motivation system adoption model (HMSAM)*. Journal of the Association for Information Systems (JAIS), vol. 14(11), 617–671.

[9] Gay L.R. dan Diehl P.L. 1992. Research Method for Business and Management.

[10] Likert, Rensis. 1932. *A Technique for the Measurement of Attitudes*. Archives of Psychology 140.

[11] van Rijsbergen, C. J. 1979. *Information Retrieval*. London: Butterworths.

ISSN 2085-4552

Aplikasi Kuartu Berbasis Android Sebagai Media Pertukaran Informasi Kartu Nama

Nandi Syukri [1], Eko Budi Setiawan[2]
Program Studi Teknik Informatika, Universitas Komputer Indonesia, Bandung, Indonesia
nandisyukri12@gmail.com[1], eko@email.unikom.ac.id[2]

Diterima 21 April 2017
Disetujui 30 Mei 2017

Abstract*— Business Card is the most efficient, effective and appropriate tool for every business men no matter they are owners, employees, more over marketers to provide information about their businesses. Unfortunately, it is very difficult to bring and manage business card in large numbers also to remember the face of the business card owner. A Business Card application need to be built to solve all those issues mentioned above. The Application or software must be run in media which can be accessed anywhere and anytime such as smart phone. Kuartu is as business card application run in mobile devices. Kuartu is developed using object base modeling for mobile sub system. The platform of the mobile sub system is android, as it is the most widely used platform in the world. The Kuartu application utilizing NFC and QR Code technology to support the business card information exchange and the Chatting feature for communication. Based on the experiment and test using black box methodology, it can be concluded that Kuartu application makes business card owner to communicate each other easily, business card always carried, easy to manage the cards and information of the business card owner can be easily obtained.

Index Terms*— Business Card, Android, Kuartu, NFC, QrCode, Chatting.

I. PENDAHULUAN

Kartu nama adalah sebuah kartu yang menyampaikan informasi tentang sebuah perusahaan ataupun individu yang disampaikan hanya sebagai pengingat dalam sebuah perkenalan formal. Sebuah Kartu nama biasanya berisi tentang nama perusahaan termasuk logo perusahaan, alamat pos, nomer telepon, nomor fax dan *email*.

Pada saat ini, pelaku usaha selalu memakai kartu nama sebagai media promosi dan media informasi pribadi untuk orang-orang yang ingin memakai jasanya. Penelitian awal dilakukan dengan survey yang dilakukan pada bulan September-Oktober 2016 kepada beberapa para pemilik usaha di Kota Bandung. Berdasarkan hasil survey kuisioner, sebanyak 83% dari 30 koresponden yang terdata, banyak orang mengalami kesulitan dalam berkenalan dengan pemilik kartu nama jika mempunyai ratusan kartu nama. Kesulitan yang terjadi adalah apabila seseorang tersebut harus menghubungi satu per satu jika ingin berkenalan dengan pemilik kartu. Hal seperti ini tidak efektif dalam berkenalan dengan pemilik kartu nama.

Pada dasarnya sebagai manusia mempunyai sifat buruk alami yaitu sifat khilaf atau lupa dalam mengingat sesuatu. Situasi seperti ini terjadi bagi pemilik usaha yang mempunyai kartu nama. Berdasarkan hasil survei kuisioner, 63% pemilik kartu terkadang lupa membawa kartu namanya sendiri. Masalah seperti ini menjadi kendala jika saling bertukar kartu nama dengan pemilik usaha lainnya.

Pada permasalahan lainnya berdasarkan survei data kuisioner 19 orang dari 30 koresponden pemilik usaha sering mengalami kewalahan dalam mengatur kartu nama milik seseorang dengan jumlah yang banyak. Kesulitan tersebut dapat mengakibatkan kehilangan kartu nama dan lupa menaruh kartu nama tersebut. Hal ini menjadi tidak efisien, jika suatu saat nanti akan membutuhkan atau menghubungi orang tersebut untuk membuat sebuah acara atau bekerja sama dalam menjalankan usaha.

Informasi didalam kartu nama biasanya berisi informasi pribadi dan tidak mencantumkan wajah pemilik kartu nama tersebut. Berdasarkan survey kuisioner 76% dari 30 koresponden banyak orang tidak mengingat wajah pemilik kartu seseorang. Hal ini menyebabkan sulitnya mengenal wajah orang tersebut jika ingin mengadakan pertemuan pertama kali.

Pemilihan perangkat *smartphone* android dalam penelitian ini karena banyaknya pengguna *smartphone* android di Indonesia seperti yang dikutip disitus berita detik.com yang di lansir dari analis Horace H bahwa di tahun 2014 android menempati posisi pertama sebagai *platform* yang banyak digunakan dengan jumlah pengguna sebanyak 1 miliar dibandingkan IOS sebanyak 700 juta [1]. Tahun 2015 android memiliki *market share* sebesar 53,2% dari OS *smartphone* yang lainnya [2]. Maka dilihat dari presentase pengguna dan market *share* android aplikasi ini cocok dibuat dengan berbasis android agar bisa dipakai orang banyak dan bermanfaat untuk semua orang.

Saat ini banyak teknologi yang dapat digunakan untuk mengatasi masalah tersebut, salah satunya dengan memanfaatkan teknologi yang memiliki kemampuan berinteraksi dua arah antar perangkat elektronik yang lebih aman dan sederhana yaitu dengan menerapkan teknologi *Near Field Communication* (NFC) sebagai media pertukaran kartu nama [3]. Berbagai penelitian yang sudah ada dalam penggunaan teknologi NFC, diharapkan teknologi NFC mampu mengatasi masalah yang ada pada pemilik kartu nama.

Berdasarkan masalah-masalah yang telah diuraikan, terdapat permasalahan seperti kesulitan berkomunikasi untuk berkenalan dengan orang banyak, maka dibutuhkan sistem aplikasi komunikasi untuk mengatasi masalah tersebut. Pemanfaatan teknologi NFC dan QrCode sebagai media pertukaran kartu nama diharapkan mampu mengatasi permasalahan pada pemilik kartu nama jika pemilik kartu nama lupa membawa kartu nama. Aplikasi ini dapat mengatur dan mengelola kontak kartu nama dengan jumlah banyak untuk mengatasi kesulitan pemilik kartu nama dalam mengelola kartu nama dan permasalahan lainya adalah banyak orang mengalami kendala untuk mengingat wajah pemilik kartu nama, dengan sistem yang akan di buat diharapkan mampu mengatasi permasalahan tersebut.

II. LANDASAN TEORI

A. Kartu Nama

Kartu nama adalah sebuah kartu yang menyampaikan informasi tentang sebuah perusahaan ataupun individu yang disampaikan hanya sebagai pengingat dalam sebuah perkenalan formal. Sebuah Kartu nama biasanya berisi tentang nama perusahaan termasuk logo perusahaan, alamat pos, nomor telepon, nomor fax dan *email*. Desain kartu nama harus sesuai dengan karakter usaha, dalam hal ini perorangan maupun perusahaan sehingga bisa meningkatkan nilai kualitas pada diri seseorang. Kartu nama harus menunjukkan siapa pemiliknya dan memberikan informasi mengenai segala hal yang bisa dilakukan oleh pemilik dan juga layanan yang bisa diberikan di masa mendatang.

B. Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar (*smartphone*) dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti begerak. Ponsel Android pertama mulai dijual pada bulan Oktober 2008. Android memiliki OS yang sangat baik, cepat dan kuat serta memiliki antarmuka pengguna intuitif yang dikemas dengan pilihan dan fleksibilitas. Android SDK (*Software Development Kit*) menyediakan *tools* dan API yang diperlukan untuk mengembangkan aplikasi pada *platform* android dengan menggunakan bahasa pemrograman Java [4].

C. Near Field Communication

Near Field Communication atau NFC merupakan teknologi komunikasi baru dengan menggunakan induksi magnet berbasis teknologi *Radio Frequency Identification* (RFID). NFC beroperasi pada frekuensi 13,56 MHz dengan kecepatan transmisi pengiriman mencapai 424 kbit/s. Jarak transmisi NFC sekitar 4-10 cm. Perbedaan antara NFC dan teknologi komunikasi *contactless* lainnya yaitu perangkat NFC dapat bersifat aktif – aktif (*peer to peer*) dan aktif – pasif. Oleh karena itu NFC. selalu melibatkan inisiator *(reader*) dan target. Inisiator aktif menghasilkan medan RF (*Radio Frequency*) yang dapat memberikan kekuatan ke target yang pasif sehingga tidak memiliki sumber daya. Hal ini memungkinkan target NFC untuk memiliki bentuk yang sangat sederhana seperti stiker, gantungan kunci, atau kartu yang tidak memerlukan energi khusus [5].

D. QR Code

QR Code adalah *image* berupa matriks dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya. *QR Code* merupakan evolusi dari kode batang. *Barcode* merupakan sebuah simbol penandaan objek nyata pada komputer. Contoh gambar *QR Code* dapat dilihat pada Gambar 1 berikut :

Gambar 1. *QR Code*

Seiring berkembangnya *QR Code*, semakin banyak penelitian yang dilakukan mengenai kode simbol ini. Beberapa penelitian yang telah dilakukan diantaranya adalah [6] :

1. Pembuatan aplikasi pembacaan *QR Code* menggunakan perangkat mobile berbasis J2ME.
2. *QR Code* untuk tandatangan digital.
3. *QR Code* untuk autentikasi novel user.
4. *QR Code* untuk edukasi.

III. HASIL DAN PEMBAHASAN

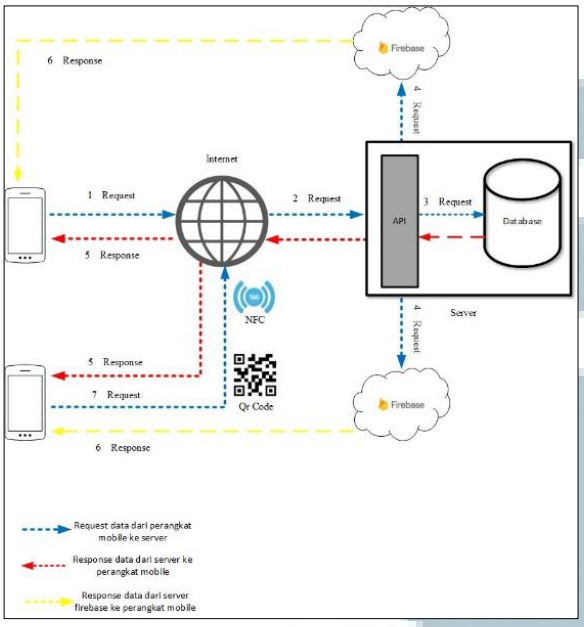
Hasil dan pembahasan yang akan dibahas pada penelitian ini terdiri dari analisis dan perancangan sistem, implementasi sistem serta yang terakhir yaitu pengujian sistem.

A. *Analisis dan Perancangan Sistem*

Pada bahasan analisis dan perancangan sistem, penelitian ini terdiri dari analisis arsitektur sistem, analisis NFC, analisis QRCode, analisis fitur *chatting*, analisis kebutuhan fungsional.

A.1. *Analisis Arsitektur Sistem*

Analisis Arsitektur Sistem merupakan sistem yang akan dibangun, aplikasi Kartu nama yang berbasis internet ini akan berkomunikasi oleh web *service* dengan menggunakan pertukaran bahasa JSON, adapun rancanganan sistem yang akan dibangun dapat dilihat di Gambar 2 berikut.



Gambar 2. Arsitektur Sistem Perangkat Lunak

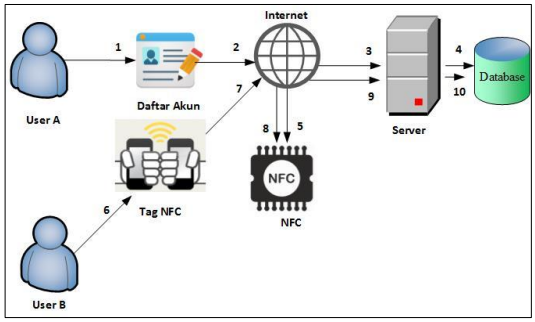
Berikut adalah deskripsi dari Gambar 2 Arsitektur Sistem Perangkat lunak:

1. Perangkat *mobile* pengguna melakukan *request* data menggunakan jaringan internet ke *server* melalui API. Pengguna bisa melakukan pendaftaran akun dan menyimpan data pribadi, gambar kartu nama, menambah kontak pertemanan serta berinteraksi secara langsung di dalam aplikasi.
2. *Server* menerima *request* data dari *server* dan menentukan jenis *request* yang diminta. *Server* akan mengecek jenis *request* yang disampaikan. Jenis *request* berupa data *text*, data gambar dan data kirim pesan.
3. Jika *server* menerima permintaan data *text* dan data gambar maka *server* akan langsung menyimpan atau mengambil data yang ada di dalam *database*. *Database* akan mengecek ketersedian data *tex*t dan data gambar milik pengguna. Bila tidak tersedia, *database* akan memberikan notifikasi ke *server* dan diteruskan ke pengguna.
4. Perangkat *mobile* pengguna melakukan *request* data pesan menggunakan jaringan internet ke *server firebase* melalui API. Pengguna melakukan kirim pesan kepada pengguna lain dimana pesan tersebut akan dikirim ke *server firebase*. *Firebase* adalah layanan DbaaS (*Database as a Service*) dengan konsep *realtime*. *Firebase* merupakan penyedia layanan *cloud* dengan *backend* sebagai *service*.
5. Setelah *server* menerima data yang diminta berupa data *text* dan data gambar, maka data tersebut akan dikembalikan dalam bentuk JSON untuk diproses perangkat *mobile* pengguna. Pengguna dapat melakukan modifikasi data pribadi dan dapat melakukan pergantian data gambar.
6. Setelah *server firebase* menerima data pesan yang diminta, maka data tersebut akan dikembalikan dalam bentuk JSON untuk diproses dan menampilkan notifikasi terima pesan ke perangkat *mobile* pengguna.
7. Perangkat *mobile* melakukan pertukaran kartu nama dengan menggunakan fitur NFC atau *QR Code*. Pengguna dapat melakukan 2 cara untuk dapat melakukan pertukaran kartu nama yaitu dengan NFC atau *QR Code*. Jika 2 pengguna mempunyai fitur NFC pada *smartphone*-nya maka pengguna bisa menggunakan fitur NFC pada aplikasi dan jika tidak mempunyai fitur NFC, pengguna dapat menggunakan fitur *QR Code* didalam aplikasi..

A.2. *Analisis NFC*

Sistem yang akan dibangun adalah aplikasi untuk pertukaran kartu nama dengan menggunakan teknologi NFC pada fitur *Smartphone* yang di miliki oleh *user*. Sistem pertukaran kartu nama yang dilakukan yaitu dengan menghidupkan fitur NFC pada *smartphone* kedua *user* dan mendekatkan kedua *smartphone* tersebut.

Sistem *write* dan *reader* digunakan oleh untuk pengguna aplikasi, sistem *write* untuk menginputkan data *user* ke dalam *tag* NFC pada aplikasi. Sistem *reader* digunakan untuk melakukan pertukaran kartu nama atau menambah kontak pertemanan didalam aplikasi. Deskripsi dari analisis cara kerja NFC pada sistem yang akan dibangun ditunjukkan pada Gambar 3.



Gambar 3. Proses Baca dan Tulis Tag *NFC*

Berikut adalah deskripsi dari Gambar 3 proses baca dan tulis *tag* NFC:

1. Pengguna A melakukan pendaftaran akun pada aplikasi. Pendaftaran berisi *email*, nama depan , nama belakang dan *password*.
2. Data yang sudah didaftarkan akan diteruskan ke *server* dan disimpan didalam *database* melalui jaringan internet.
3. *Server* menerima data yang akan diteruskan ke dalam *database*.
4. Data telah tersimpan didalam *database*
5. Data yang sudah tersimpan akan dikonversi ke NDEF *Record* yang berisi Id *User* A.
6. *User* B bersiap melakukan mendekatkan *smartphone*nya didalam aplikasi.
7. *User* B dan *user* A saling mendekatkan *smartphone*nya menggunakan jaringan internet.
8. Proses berhasil dan hasilnya akan di konversi menjadi id kontak pada tabel kontak.
9. *Server* menerima data yang akan diteruskan ke dalam *database*.
10. Data telah tersimpan didalam *database*.

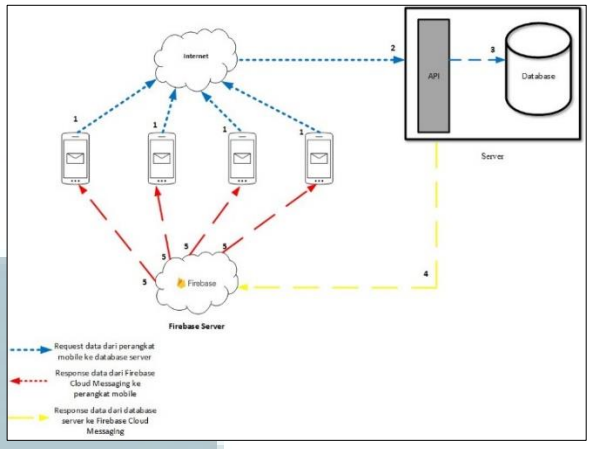
A.3. *Analisis QrCode*

Aplikasi yang dibangun merupakan aplikasi yang melakukan *encoding* dan *decoding* kartu nama menjadi *QR Code*. Pada proses *encoding*, terjadi perubahan data dari id *user* dalam database aplikasi yang bertipe integer dengan panjang 4 digit menjadi sebuah *QR Code* yang berupa citra. Dalam mengkonversi inputan berupa id *user* menjadi output berupa citra *QR Code* tersebut dilakukan dalam beberapa proses yang merupakan tahapan dari proses *encoding*.

Sedangkan untuk proses *decoding* dibutuhkan citra *QR Code* yang akan diambil secara *realtime* menggunakan kamera perangkat Android untuk kemudian dari input citra tersebut diolah lebih lanjut oleh sistem untuk mendapatkan id kontak, kemudian id kontak dikirim ke *server* guna masuk ke dalam *database* aplikasi dan berhasil menambah kontak pada akun.

A.4. *Analisis Fitur Chatting*

Platform mobile adalah salah satu subsistem yang dipilih untuk pembangunan *chatting* dari perangkat lunak ini. Arsitektur *chatting* perangkat lunak pada *platform mobile* mengambarkan bagaimana perangkat lunak saling berinteraksi seperti diilustrasikan pada Gambar 4.



Gambar 4. Arsitektur Sistem *Chatting*

Berikut adalah deskripsi dari gambar 4 Arsitektur Ssitem *Chatting* pada perangkat lunak :

1. Perangkat *mobile* pengguna melakukan *request* data kirim pesan menggunakan jaringan internet ke database *server* dan firebase *server* melalui API. Pengguna melakukan kirim pesan kepada pengguna lain dimana pesan tersebut akan dikirim ke *server* firebase.
2. *Server* menerima data kirim pesan yang diminta, data kirim pesan berupa teks dan data tersebut diteruskan ke dalam database.
3. Database menerima data pesan dari *server*. Data pesan tersimpan didalam database ke tabel *message*s
4. API me*response* data kirim pesan tersebut dan diteruskan ke *server firebase cloud messaging*.
5. *Server Firebase Cloud Messaging* meneruskan pesan ke pengguna perangkat *mobile* lainnya dan Perangkat *mobile* mendapatkan notifikasi terima pesan.

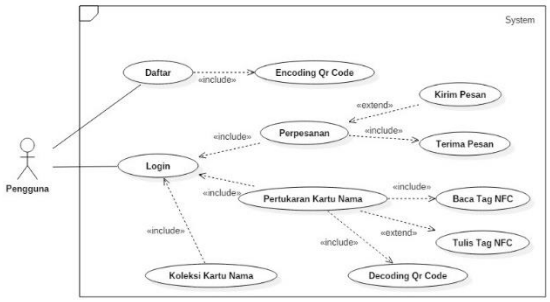
Firebase adalah layanan *DbaaS* (*Database as a Service*) dengan konsep *realtime*. *Firebase* merupakan penyedia layanan *cloud* dengan *backend* sebagai *service* yang berbasis di San Fransisco, California yang sekarang dimiliki oleh *google*. *Firebase* terdiri dari fitur pelengkap yang bisa dipadukan sesuai dengan kebutuhan.

A.5. *Analisis Kebutuhan Fungsional*

Bagian ini akan membahas mengenai *use case diagram* aplikasi *mobile*, *class diagram* dan skema relasi dari struktur tabel pada *backend web*.

A.5.1. *Use Case Diagram Aplikasi Mobile*

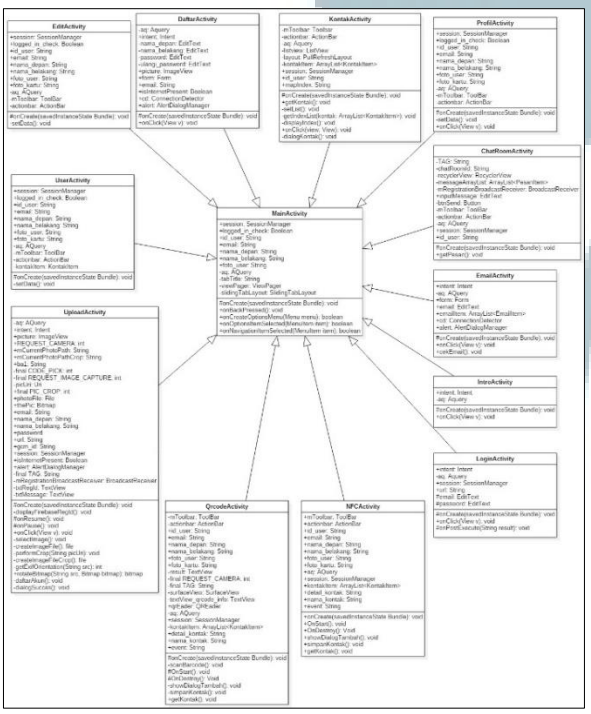
UseCase diagram adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat pengguna. Dari identifikasi aktor yang terlibat di atas maka *UseCase Diagram mobile* dapat dilihat pada Gambar 5.



Gambar 5. *Use Case Diagram Aplikasi Mobile*

A.5.2. *Class Diagram Aplikasi Mobile*

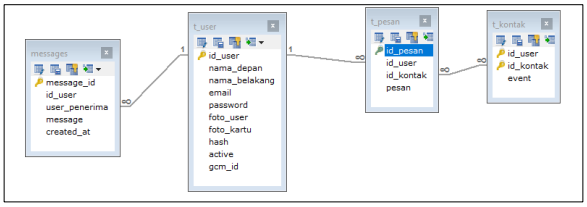
Diagram *Class* digunakan untuk menggambarkan secara abstrak struktur dari aplikasi yang akan dibangun, *class-class* yang terlibat, serta hubungan antar *class* untuk saling berkomunikasi satu sama lain. Adapun gambar dari *class diagram* aplikasi *mobile* yang dibangun, dapat dilihat pada Gambar 6 berikut.



Gambar 6. *Class Diagram Aplikasi Mobile*

A.5.3 *Diagram Relasi Aplikasi Web Backend*

Berikut merupakan skema atau diagram relasi dari keterhubungan antar tabel dalam *database* pada sub sistem web *backend*. Diagram relasi dapat dilihat pada Gambar 7.



Gambar 7. Diagram Relasi Tabel *Web Backend*

B. *Implementasi Sistem*

Tahap Implementasi dan pengujian sistem merupakan tahap penterjemahan perancangan berdasarkan hasil analisis ke dalam suatu bahasa pemrograman tertentu serta penerapan perangkat lunak yang dibangun pada lingkungan yang sesungguhnya.

B.1. *Lingkungan Implementasi*

Lingkungan implementasi yang akan dibahas dalam penelitian ini merupakan spesifikasi *hardware* dan *software* dimana sistem ini akan diimplementasikan dan diakses. Spesifikasi perangkat keras untuk membangun dan implementasi sistem dapat dilihat pada Tabel 1 berikut.

Tabel 1. Spesifikasi Perangkat Keras PC

Perangkat Keras	Spesifikasi
Processor	*Intel(R) i5-3230M CPU @ 2.60GHz*
Monitor	*14 inches WXGA, 1,366 x 768*
RAM	*4 GB*
Graphics	*Intel(R) HD Graphics 4000 2 GB*
Harddisk	*750 GB*

Sedangkan untuk spesifikasi perangkat keras untuk mengakses sistem dari sisi *mobile* android yaitu :

Tabel 2. Spesifikasi Perangkat *Mobile* Android

Perangkat Keras	Spesifikasi
Processor	*Qualcomm MSM8992 Snapdragon 808*
Display	*IPS LCD capacitive touchscreen, 16M colors, 5.2 inches*
Memory	*2 GB*
Storage	*32 GB*
Connectivity	*Wi-Fi 802.11 a/b/g/n/ac, dual-band, Wi-Fi Direct, DLNA, hotspot*

Berikut adalah spesifikasi kebutuhan *software* yang digunakan dalam pembangunan perangkat lunak ini dapat dilihat pada Tabel 3.

Tabel 1 Lingkungan *Software* Pembangun Perangkat Lunak

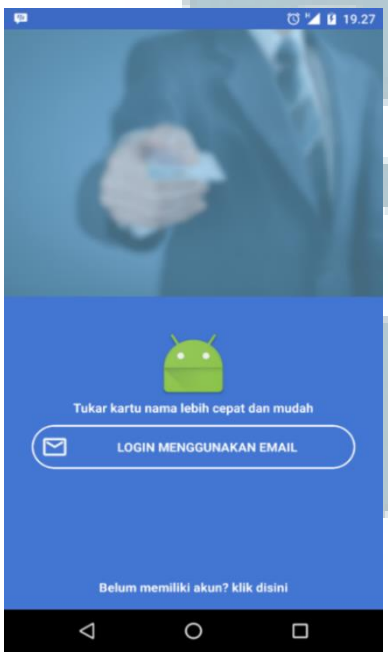
Perangkat Lunak	Spesifikasi
Sistem Operasi	*Windows 10 Pro 64-bit*
Sistem Operasi Android	*Android versi 7.1.1 Nougat*
Bahasa Pemrograman	*Java, JSON, PHP*
Tools Pembangun	*Android Studio, Sublime Text 3*

B.2. *Implementasi Antarmuka Mobile*

Berikut adalah implementasi dari beberapa antarmuka dari sistem *platfom mobile android* yang dapat dibangun.

B.2.1. *Implementasi Antarmuka Halaman Login*

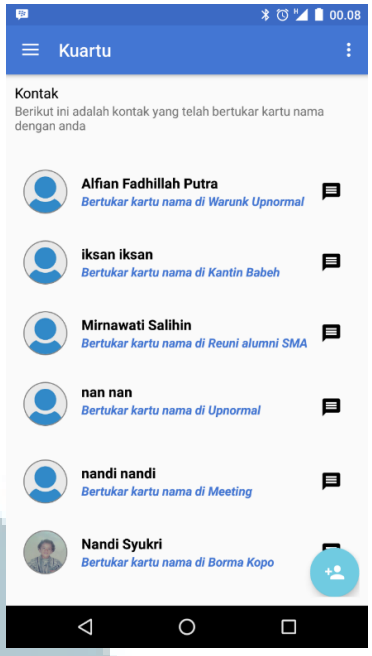
Halaman ini berfungsi untuk *Login* kedalam sistem. Halaman ini harus menginputkan *email* dan *password* yang sudah terdaftar didalam sistem. Tampilannya dapat dilihat pada Gambar 8.



Gambar 8. Antarmuka Halaman *Login*

B.2.2. *Implementasi Antarmuka Halaman Kontak*

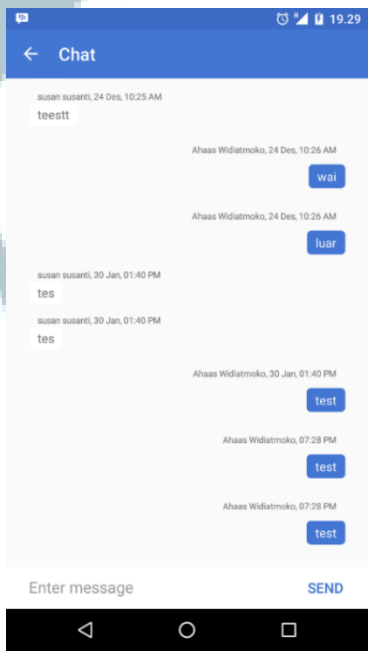
Halaman ini memunculkan informasi kontak pertemanan yang sudah bertukar kartu nama melalui aplikasi. Halaman kontak dapat dilihat pada Gambar 9.



Gambar 9. Halaman Kontak

B.2.3. *Implementasi Antarmuka Halaman Chatting*

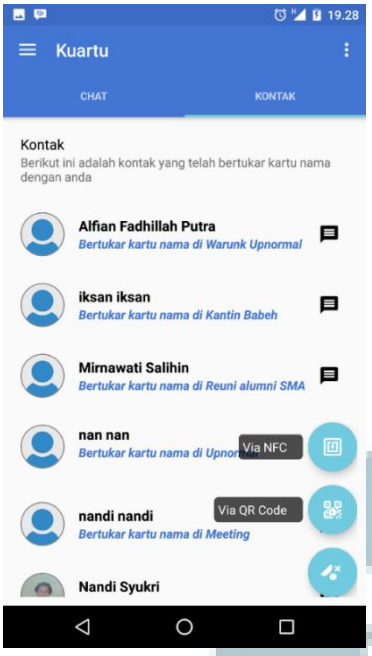
Halaman ini berfungsi untuk melakukan komunikasi dengan kontak lainnya yang sudah bertukar kartu nama. Tampilannya dapat dilihat pada Gambar 10.



Gambar 10. Halaman *Chatting*

B.2.4. *Implementasi Antarmuka Pertukaran Kartu*

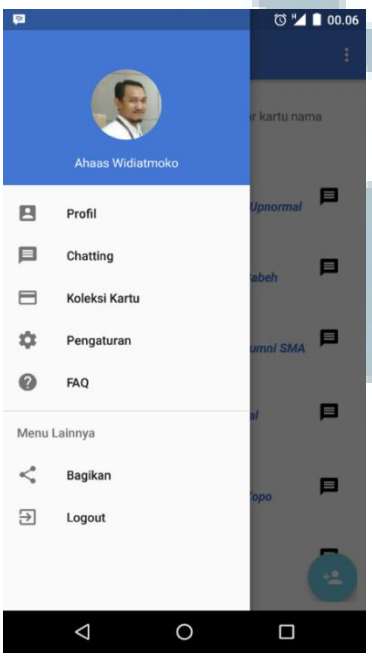
Halaman ini berfungsi untuk melakukan pertukaran kartu nama. Pertukaran ini bisa menggunakan melalui NFC dan *QRCode*. Tampilannya dapat dilihat pada Gambar 11.



Gambar 11. Halaman Antarmuka Pertukaran Kartu

F.2.4. Implementasi Antarmuka *Informasi Akun*

Halaman ini menampilkan informasi akun didalam aplikasi. Halaman ini menampilkan beberapa menu didalam aplikasi. Tampilannya dapat dilihat pada Gambar 12.



Gambar 12. Halaman Antarmuka Informasi Akun

C. *Pengujian Sistem*

Pengujian sistem dalam penelitian ini dilakukan dengan dua tahapan, yaitu pengujian fungsional dengan menggunakan metode *black box*, serta pengujian kualitas perangkat lunak menggunakan metriks kualitas eksternal aplikasi berdasarkan ISO-9126. Berikut adalah hasil pengujian dari Pembangunan Aplikasi Kuartu yang telah dilakukan.

1. Evaluasi Pengujian Fungsional

Pengujian fungsional dalam penelitian ini menggunakan metode *black-box*. Ada 11 item fungsional yang diuji yaitu proses *login*, daftar, tulis tag NFC, *encoding* QrCode, *chatting*, kirim pesan, terima pesan, koleksi kartu nama, pertukaran kartu nama, baca tag NFC, dan yang terakhir yaitu proses *decoding* QrCode. Berdasarkan hasil pengujian yang telah dilakukan dapat disimpulkan bahwa secara fungsional proses pada sistem Aplikasi Kuartu Sebagai Media Pertukaran Informasi Kartu Nama Berbasis Android telah berjalan dengan baik dan sesuai dengan yang diharapkan.

2. Evaluasi Pengujian Kualitas Eksternal Aplikasi Kuartu

Model ISO-9126 memiliki beberapa metriks utama. Pengujian kualitas pada penelitian ini dikhususkan menggunakan tiga metriks yaitu metriks *functionality*, *usability* dan metriks *efficiency*. Pengujian kualitas eksternal ini dilakukan kepada 20 responden *early adopter* pengguna aplikasi yang dilakukan pada bulan Februari 2017. Berdasarkan hasil pengujian kuesioner metriks kualitas eksternal aplikasi menggunakan ISO-9126 yang telah dilakukan, didapatkan hasil penilaian seperti berikut :

a. Dari segi faktor *functionality* didapatkan hasil penilaian dengan nilai 0,41 sehingga Aplikasi Kuartu ini termasuk kedalam kategori "*Medium*" atau cukup dari segi *functionality*.
b. Dari segi faktor *usability* mendapatkan hasil penilaian 0,47 sehingga Aplikasi Kuartu ini termasuk dalam kategori "*Medium*" atau cukup dilihat dari segi *usability*.
c. Dari segi faktor *effectiveness* mendapatkan hasil penilaian 0,49 sehingga Aplikasi Kuartu ini termasuk kedalam kategori "*Medium*" atau cukup dilihat dari segi *effectiveness*.
d. Dari segi faktor *productivity* mendapatkan hasil penilaian 0,80 sehingga Aplikasi Kuartu ini termasuk kedalam kategori produktivitas tinggi atau *"High*".
e. Dari segi faktor *safety* mendapatkan hasil penilaian 0,97 sehingga Aplikasi Kuartu ini termasuk kedalam kategori keamanan yang tinggi atau *"High*".
f. Dari segi faktor *satisfaction* mendapatkan hasil penilaian 0,62 sehingga Aplikasi Kuartu ini termasuk kedalam kategori "*Medium*" atau cukup dilihat dari segi *factor satisfaction*.

IV. SIMPULAN

Adapun kesimpulan dan saran yang didapatkan dari hasil penelitian ini yaitu :

A. *Kesimpulan*

Berdasarkan hasil yang didapat dari penelitian yang dilakukan, maka dapat disimpulkan beberapa kesimpulan berikut ini.

1. Pengguna dapat berkomunikasi secara langsung melalui aplikasi. Hal ini dapat memudahkan dalam bekomunikasi dengan pemilik kartu nama lainnya dengan lancar dan efektif.
2. Pengguna dapat selalu membawa kartu namanya menggunakan aplikasi android hasil dari penelitian. Hal ini dapat mengurangi kecerobohan pemilik kartu nama dalam membawa kartu namanya dan memudahkan bertukar kartu nama dengan pemilik kartu nama lainnya.
3. Pengguna dapat mengatur dan mengolah kartu nama milik seseorang dengan jumlah banyak. Hal ini memudahkan pengguna menyimpan milik kartu nama seseorang dan pengguna dapat menghubungi seseorang dari kartu nama yang disimpan.
4. Pengguna dapat mengetahui informasi pribadi lengkap pemilik kartu nama lainnya. Hal ini memudahkan pengguna untuk berkenalan dengan pemilik kartu nama lainnya saat bertemu atau berkenalan disuatu tempat.

B. Saran

Beberapa saran yang dapat digunakan sebagai paduan pengembangan perangkat lunak kearah yang lebih baik guna mendukung peningkatan pengguna pada aplikasi kuartu ini. Adapun saran-saran terhadap pengembangan aplikasi kuartu adalah sebagai berikut:

1. Menyempurnakan kualitas dalam penggunaan media komunikasi, mengingat masyarakat luas selalu menggunakan media komunikasi. Fitur komunikasi didalam aplikasi kuartu terkadang mengalami kendala. Kendala yang terjadi terkadang tidak *real timenya* pesan yang masuk saat melakukan *chat*.
2. Mengembangkan perangkat lunak dari segi performansi. Keterbatasan *server* dan *hosting* menjadi kekurangan. Kekurangan yang terjadi server terkadang mengalami *down* sehingga aplikasi tidak bisa diakses dan penggunaan *bandwith* kuota *hosting* perlu ditingkatkan.
3. Mengembangkan *platform* yang dapat didukung oleh semua perangkat lunak, mengingat saat ini hanya mendukung *platform* android saja. Hal ini berkaca terhadap masyarakat luas yang tidak hanya menggunakan *smartphone* android saja.

DAFTAR PUSTAKA

[1] Heriyanto Trisno.,2014, Indonesia Masuk 5 Besar Negara Pengguna Smartphone [Online],http://inet.detik.com/read/2014/02/03/171002/2485920/317/indonesiamasuk-5-besar-negara-pengguna-smartphonen, (diakses 21 Agustus 2016).

[2] Lella Adam., 2015, comScore Reports January 2015 U.S. Smartphone Subscriber MarketShare[Online],http://www.comscore.com/Insights/Market\Rankings/comScore-Reports-January-2015-US-Smartphone-Subscriber-Market-Share

[3] Chandra, Sadewa; Nurochmah, Tri Yanti., 2014. "Pengenalan Near Field Communication (NFC)".

[4] N. Safat, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, Bandung: Informatika Bandung, 2012.

[5] T, Igoe; D, Coleman; D, Jepson. 2014. "Begining NFC," dalam Near Field Communication With Ardunio, Android & Phonegap.

[6] Rahayu and Y. D. , "Pembacaan Quick Response Code Menggunakan Perangkat," in Politeknik Elektronika Negeri Surabaya Institut, Surabaya, 2006.

ISSN 2085-4552

Model Klasifikasi Mata Katarak dan Normal Menggunakan Histogram

Valencia Wirawan[1], Yustinus Eko Soelistio[2]
Information System Department, Universitas Multimedia Nusantara
valencia.wirawan@student.umn.ac.id[1], yustinus.eko@umn.ac.id[2]

Diterima 6 April 2017
Disetujui 14 Juni 2017

***Abstract*—Telah banyak penelitian pada citra medis telah diadopsi oleh sebagian besar ilmuwan dan dokter yang dapat membantu dalam mendeteksi gangguan pada mata terutama katarak. Namun, umumnya penelitian tersebut menggunakan citra medis atau digital yang relatif mahal dan sulit didapatkan oleh sebagian orang, dan metode yang rentan akan translasi (pergeseran), serta perubahan ukuran gambar dan bentuk objek. Penelitian ini mengembangkan sebuah metode menggunakan model *histogram* untuk mengklasifikasi mata katarak dari citra digital dengan (1) format yang lebih umum seperti JPEG dan (2) lebih toleranterhadap translasi dan perubahan ukuran. Metode ini juga mampu bekerja dengan baik menggunakan citra digital dalam citra mata yang tidak tegak lurus terhadap kamera. Metode ini mencapai akurasi 79,03% dalam kondisi bebas dan 88.47% dalam kondisi mata tegak lurus terhadap kamera. Metode ini mempunyai kompleksitas yang rendah sehingga dapat digunakan pada komputer dengan spesifikasi rendah dan sistem yang membutuhkan kecepatan mendekati *real-time*.**

Index Terms*—*Image processing, cataract, classification, histogram

I. PENDAHULUAN

Gangguan mata dan kebutaan merupakan kondisi yang dapat mempengaruhi jangka waktu hidup seseorang, baik melalui biaya medis untuk pengobatan maupun melalui biaya ekonomi dan sosial yang tinggi untuk mengurangi kehilangan penglihatan [1]. Berdasarkan data dari Kementrian Kesehatan RI, 50% kasus kebutaan di Indonesia disebabkan oleh katarak. Setiap tahunnya, penderita katarak terus bertambah hingga diperkirakan mencapai 1000 pasien. Secara global, Indonesia menempati urutan kedua sebagai negara dengan jumlah penderita katarak tertinggi setelah Etiopia [2].

Terkait dengan permasalahan tersebut, kini telah banyak penelitian citra medis yang dilakukan dalam bidang pendeteksian gangguan pada mata [3-11]. Penelitian pada citra medis telah diadopsi oleh sebagian besar ilmuwan dan dokter yang dapat membantu dalam mendeteksi kelainan pada mata seperti pada katarak [3-6]. Penelitian-penelitian tersebut menghasilkan berbagai metode pendeteksian katarak yang memanfaatkan beragam jenis citra digital. [3-5].

Penelitian yang pertama [3] mendeteksi adanya indikasi katarak dan mengklasifikasikan citra digital katarak ke dalam dua kategori yaitu *nuclear* dan *cortical*. Deteksi katarak dilakukan dengan membandingkan tingkat intensitas keabuan citra digital mata normal dan katarak. Pengelompokkan kedua kategori katarak dilakukan dengan pengukuran bentuk bundar (*circularity*). Metode pengukuran bentuk bundar ini dapat rentan akan ukuran dan bentuk objek misalnya ketika citra digital mata berada dalam kondisi yang tidak utuh seperti tertutup objek lain atau pengambilan gambar yang kurang tepat oleh kamera. Penelitian ini berhasil mencapai tingkat akurasi 94.96%. Digunakan data citra digital dengan bola mata yang mengarah lurus ke depan atau ke arah kamera dan iris yang utuh sehingga tidak diketahui bagaimana performa sistem apabila kondisi mata pada citra digital tidak dalam keadaan utuh.

Penelitian yang kedua [4] berfokus pada pembuatan sistem deteksi katarak otomatis jarak jauh dengan memanfaatkan teknologi cloud. Set data yang digunakan berupa citra fundus retina. Dibutuhkan sebuah fundus camera untuk memperoleh citra fundus sehingga relatif lebih mahal dan sulit untuk diperoleh.

Penelitian yang ketiga [5] bertujuan untuk mengklasifikasikan mata normal, katarak, dan pascakatarak. Terdapat empat fitur yang dideteksi antara lain *Small Ring Area* (SRA), *Big Ring Area* (BRA), dan *Edge Pixel Count* (EPC) yang menggunakan 2 metode ambang (*threshold*) untuk mendeteksi tepian yang kuat dan lemah dan kemudian menghitung jumlah pixel putih pada keluaran dari deteksi tepian, dan object perimeter. Penentuan SRA dan BRA serta metode ekstraksi yang menggunakan threshold performanya dapat dipengaruhi oleh translasi (pergeseran) dan ukuran gambar.

Walaupun ketiga metode yang dijelaskan diatas telah berhasil mendeteksi mata katarak dengan baik, metode-metode tersebut membutuhkan citra digital yang telah dikondisikan dalam kondisi tertentu yang apabila kondisi tersebut tidak tercapai maka dapat menurunkan performa dari mereka. Penelitian ini mengembangkan metode pendeteksian katarak dengan menggunakan citra digital yang (1) diambil dalam kondisi bebas (tidak mengharuskan mata menghadap lurus kearah kamera), dan (2) berdasarkan *visible light* dan dapat diambil dengan kamera umum

(contoh: *webcam*). Penelitian ini berfokus pada pembuatan model klasifikasi mata katarak dan mata normal dengan pengolahan citra digital dalam format JPG/JPEG yang lebih umum, relatif lebih murah dan mudah diperoleh menggunakan metode yang lebih tahan terhadap translasi dan perubahan ukuran, serta mampu bekerja dengan baik menggunakan citra digital dalam kondisi bebas.

II. METODE PENELITIAN

A. *Pengumpulan Data*

Pengumpulan citra digital mata katarak dan mata normal dilakukan melalui Google Image (images.google.com) yang merujuk pada beberapa situs: Science Photo Library (www.sciencephoto.com), Flickr (www.flickr.com), dan Eye Rounds (eyerounds.org). Setiap citra digital memuat gambar mata dan daerah sekitar mata (contoh: alis, bulu mata). Untuk kategori mata katarak, citra digital yang dikumpulkan adalah citra digital mata katarak dengan kekeruhan lensa yang terlihat jelas dan tidak terbatas pada jenis katarak tertentu.

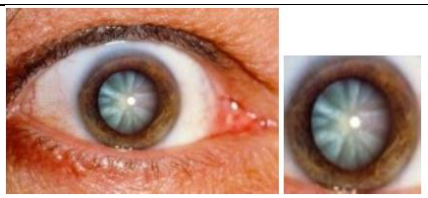
Data citra digital tersebut dapat diperoleh dengan mengirimkan email kepada penulis. Data tersebut dapat digunakan oleh umum untuk kebutuhan penelitian.

B. *Data Preprocessing*

Citra bagian sklera atau tepian luar bola mata yang berwarna putih dibuang karena memiliki warna yang sama antar kedua kategori katarak dan normal sehingga pada metode ini hanya digunakan fitur warna dari bagian iris dan pupil saja. Citra digital yang digunakan sebagai *input* dalam metode ini dibagi menjadi 2 jenis yaitu citra digital dalam kondisi bebas dan citra digital dalam kondisi ideal.

Pada citra digital dalam kondisi bebas, setiap gambar mata dipotong dengan skala 1:1 dan diubah ukurannya menjadi 200×200 pixel. Dilakukan pemotongan dari titik tengah citra digital dengan ukuran 50 pixel ke arah atas, bawah, kiri, dan kanan dengan tujuan untuk mengambil bagian iris dan pupil (Gambar 1.a). Sedangkan pada citra digital dalam kondisi ideal, setiap gambar dipotong tepat pada bagian iris mata (Gambar 1.b).

(a) (i) citra asli, (ii) citra asli dengan ukuran 200×200 pixel, (iii) citra setelah pemotongan dari titik tengah.



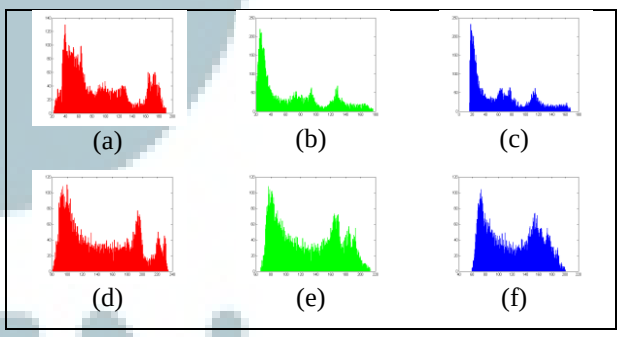
(b) (i) citra asli, (ii) citra setelah pemotongan pada bagian iris mata.

Gambar 1: Proses pemotongan dan pengubahan ukuran citra digital (a) bebas dan (b) ideal.

C. *Pembangunan Model Histogram*

Model dibuat dengan menjumlahkan nilai seluruh pixel dari seluruh citra digital pada set data model per saluran warna yaitu merah, hijau, dan biru untuk masing-masing kategori normal dan katarak. Setiap saluran warna memiliki ukuran 3 dimensi yaitu panjang (kanan-kiri), tinggi (atas-bawah), dan kedalaman (terang-gelap). Kemudian setiap nilai pixel hasil penjumlahan dibagi dengan jumlah citra digital pada set data model untuk mendapatnya nilai rata-rata dari seluruh citra digital (Gambar 2).

Model direpresentasikan dalam bentuk *histogram* dengan jumlah $bin = 256$ dimana setiap *bin* merupakan representasi intensitas warna dari 0 (gelap total) sampai 255 (terang total).



Gambar 2: Model *histogram* dari mata (a-c) normal dan (d-f) katarak dalam saluran warna (a, d) merah, (b, e) hijau, dan (c, f) biru.

Nilai pixel pada masing-masing saluran warna dirubah menjadi satu dimensi dengan cara menggabungkan nilai pixel pada setiap saluran warna. Setelah masing-masing saluran warna memiliki nilai pixel dengan ukuran 1 dimensi. Masing-masing saluran warna pada kedua model (model mata katarak dan model mata normal) membentuk model histogram yang akan digunakan untuk membandingkan citra digital yang akan di klasifikasi. Perhitungan *histogram* dilakukan dengan menggunakan MATLAB versi 2012b.

D. *Klasifikasi*

Citra digital yang akan di klasifikasi melalui proses yang sama seperti pada proses pembangunan model (Bab 2.C). Klasifikasi dilakukan dengan cara membandingkan perbedaan (jarak *Euclidean*) antara *histogram* citra digital baru dengan *histogram* kedua

model. Citra digital baru akan di klasifikasikan sebagai mata katarak / mata normal berdasarkan nilai terkecil dari jarak *Euclidean* antara citra digital tersebut dengan kedua model.

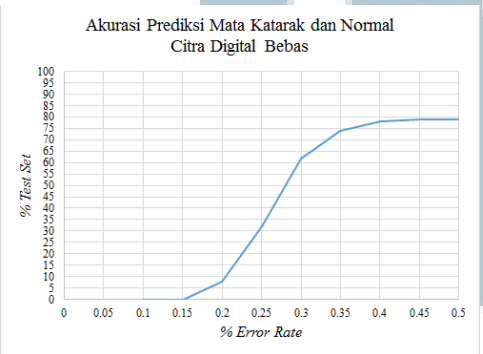
E. Pengukuran Tingkat Akurasi

Hasil klasifikasi dihitung tingkat akurasinya dengan *k-fold cross validation* dengan nilai $k = 10$. Akurasi diukur berdasarkan *error rate*, sebagai persentase selisih antara jarak *Euclidean* setiap citra digital yang diuji dan model.

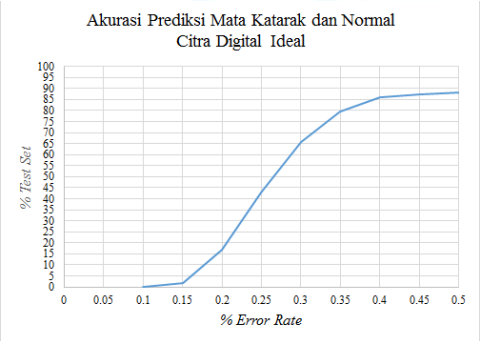
III. EVALUASI MODEL

A. Evaluasi Validitas Data

Proses pengumpulan data citra digital melalui Google Images (images.google .com) yang merujuk pada beberapa situs seperti Science Photo Library (www.science photo.com), Eye Rounds (www.eyerounds.org), dan Flickr (www.flickr.com) menghasilkan 86 buah citra digital yang diperoleh untuk masing-masing kategori mata katarak dan mata normal dengan total 172 citra. Data citra digital telah divalidasi melalui pemeriksaan oleh dokter spesialis mata RS Hermina Tangerang, dr. Suminto, Sp.M.



(a)



(b)

Gambar 3: Grafik *e* dari klasifikasi menggunakan citra digital dalam kondisi (a) bebas, dan (b) ideal. Perhatikan bahwa akurasi model telah relatif stabil ketika $e \geq 0.4$ dalam kedua kondisi.

B. Klasifikasi

Histogram setiap citra digital yang akan diuji diperoleh dengan menjumlahkan nilai seluruh pixel dan dijadikan satu dimensi per saluran warna (Bab 2.C). Hasil evaluasi menggunakan *10-fold validation* memperlihatkan akurasi sebesar 79,03% pada citra digital bebas (*true positive* (TP) = 94,17%, *true negative* (TN) = 63,89%), dan 88,47% pada citra digital ideal (TP = 94,31%, TN = 81,53%) menggunakan *error rate (e)* = 0,4% (Gambar 3, dimana akurasi model telah relatif stabil pada $e \geq 0.4$).

IV. DISKUSI

Hasil pengujian model menunjukkan bahwa metode ini memperoleh akurasi sebesar 79.03% dan 88.47%, dimana akurasi dalam kondisi ideal lebih baik dari kondisi bebas. Hal ini seperti yang diharapkap, dimana posisi iris mata pada kondisi ideal terlihat secara utuh dan perada dalam posisi yang relatif sama (tengah) sedangkan sebaliknya terjadi pada kondisi bebas yakni posisi iris bisa berada dimana saja di dalam citra digital dan tidak dapat dipastikan untuk terlihat secara utuh. Posisi iris mata yang berubah pada kondisi bebas dapat mengakibatkan perbedaan hasil *histogram* karena *histogram* hanya memanfaatkan fitur warna sehingga penggunaan metode ini membuat model kehilangan informasi spasial. Kesalahan dalam klasifikasi yang terjadi dikarenakan ketidakmampuan *histogram* dalam mengenali warna pada lokasi atau posisi tertentu pada citra digital.

Bila digunakan dalam implementasi sehari-hari, performa metode ini diukur berdasarkan hasil evaluasi pada citra digital bebas. Hasil yang didapat pada citra digital bebas mempunyai akurasi yang lebih baik pada pendeteksian mata katarak (TP), yang berarti metode ini akan lebih banyak melakukan kesalahan dalam mendeteksi mata normal (*i.e.* mata normal akan berpotensi untuk terdeteksi salah sebagai mata katarak).

Kompleksitas dari metode ini terdapat pada proses perbandingan histogram dengan $O(n)$, dimana n adalah jumlah *bin* dalam histogram. Dalam ujicoba, metode ini hanya membutuhkan 82 milidetik ($\sim 12\ fps$) untuk melakukan klasifikasi sebuah citra digital sehingga dapat pada sistem dengan spesifikasi rendah atau digunakan mendekati *real-time*.

Dibandingkan dengan penelitian terdahulu metode ini berhasil diimplementasikan pada citra digital umum (RGB) tanpa mengharuskan posisi mata menghasap lurus ke kamera (*cf.* [3]). Metode ini mempunyai kompleksitas yang lebih rendah dibandingkan metode [4]. Metode ini hanya menggunakan fitur warna sehingga tidak dipengaruhi oleh perubahan translasi (pergeseran) dan ukuran gambar (*cf.* [5]).

V. SIMPULAN DAN SARAN

Metode *histogram* berhasil diimplementasikan untuk melakukan klasifikasi mata katarak dengan normal dengan tingkat akurasi cukup baik pada kondisi ideal. Metode ini juga terbukti untuk dapat digunakan pada kondisi mata bebas sehingga penggunaannya tidak dibatasi oleh gambar mata tegak

lurus terhadap kamera. Metode ini juga mempunyai kompleksitas yang rendah sehingga dapat digunakan pada sistem dengan spesifikasi rendah atau mendekati secara *real-time*.

UCAPAN TERIMA KASIH

Terima kasih kepada dr. Suminto, Sp.M. dari RS Hermina Tangerang untuk ketersediaannya melakukan evaluasi data pada penelitian ini. Hasil penelitian ini merupakan bagian dari penelitian skripsi yang dilakukan oleh Valencia Wirawan.

DAFTAR PUSTAKA

[1] Prevent Blindness. The Economic Burden of Vision Loss and Eye Disorders in the United States. Dalam , diakses pada 8 Januari 2017.

[2] ”BCA in Support of Cataract-free Indonesia 2020”. The Jakarta Post 27 Agustus 2016.

[3] Patwari, Anayet U., Muammer D. Arif, N.A. Chowdhury, A. Arefin, & I. Imam. (2011). Detection, Categorization, and Assessment of Eye Cataracts Using Digital Image Processing. The First International Conference on Interdisciplinary Research and Development, Thailand.

[4] Kolhe, S., & Shanthi K. Guru. (2016). Remote Automated Cataract Detection System Based on Fundus Images. International Journal of Innovative Research in Science, Engineering and Technology, Vol. 5, pp. 10334-10341.

[5] Nayak, Jagadish. (2013). Automated Classification of Normal, Cataract and Post Cataract Optical Eye Image using SVM Classifier. Annalen der Physik, 322(10):891921.

[6] Aquino, A., Gegúndez-Arias, M. E., & Marín, D. (2010). Detecting the optic disc boundary in digital fundus images using morphological, edge detection, and feature extraction techniques. IEEE transactions on medical imaging, 29(11), 1860-1869.

[7] Osareh, A., Mirmehdi, M., Thomas, B., & Markham, R. (2006). Classification and localisation of diabetic-related eye disease. Computer Vision—ECCV 2002, 325-329.

[8] Philips, R., Forrester, J., Sharp, P.: Automated Detection and Quantification of Retinal Exudates. Graefes Archive for Clinical and Experimental Ophthalmology 231 (1993) 90-94.

[9] Gardner, G.G., Keating, D., Williamson, T.H., Elliott, A.T.: Automatic Detection of Diabetic Retinopathy Using an Artificial Neural Network: A Screening Tool. British Journal of Ophthalmology 80 (1996) 940-944.

[10] Li, H., Chutatape, O.: Automatic Location of Optic Disk in Retinal Images. IEEE International Conference on Image Processing (2001) 837-840.

[11] M. Foracchia, E. Grisan, and A. Ruggeri, “Detection of optic disc in retinal images by means of a geometrical model of vessel structure,” IEEE Trans. Med. Imag., vol. 23, no. 10, pp. 1189–1195, Oct. 2004.

ISSN 2085-4552

Implementasi Algoritma Genetika dan Neural Network Pada Aplikasi Peramalan Produksi Mie (Studi Kasus : Omega Mie Jaya)

Adhi Kusnadi[1], Jansen Pratama[2]
Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Gading Serpong, Indonesia
adhi.kusnadi@umn.ac.id[1], jansen_pratama@yahoo.co.id[2]

Diterima 15 Mei 2017
Disetujui 16 Juni 2017

Abstract*— Companies that produce products must be able to regulate the amount of production so that it have plan production. Therefore, it is necessary to be able to predict the amount of production. This research aims to create an application that is useful in determining the amount of production. These applications using genetic algorithms and neural network. Genetic algorithm is used to optimize the weights in the neural network. From the test results, this application uses network with 12 inputs, 5 neuron in first hidden layer, 3 neurons in the second hidden layer, and 3 neurons in the last hidden layer. Then for the genetic algorithm parameters used were 10 individuals, 50 generations, crossover probability 0.8 and mutations probability 0.1. Based on the test results, this application has the forecasting's accuracy rate reaches 86%.

Keyword : forecasting, production forecasting, genetic algorithm neural network, optimization.

I. PENDAHULUAN

Kegiatan ekonomi sangat penting bagi kehidupan manusia untuk memenuhi kebutuhannya, terdiri dari produksi, distribusi dan konsumsi. Produksi mencakup setiap usaha manusia yang menghasilkan barang/jasa yang langsung atau tidak berguna untuk memenuhi kebutuhan manusia [1]. Tentunya produksi dilakukan atas dasar permintaan dari konsumen. Perusahaan yang menghasilkan produk harus bisa mengatur jumlah produksi agar produksi menjadi terencana dengan baik. Oleh karena itu diperlukan suatu kemampuan untuk dapat meramalkan jumlah produksi [2]. Dengan demikian, maka produksi akan menjadi lebih efisien dan efektif serta dapat menghemat biaya. Untuk itu, dibutuhkan suatu sistem yang dapat membantu perusahaan dalam menentukan jumlah produksi. Menurut Gaspersz, peramalan adalah dugaan yang dibuat secara sederhana tentang apa yang akan terjadi di masa depan berdasarkan informasi yang tersedia saat ini. Jadi perusahaan dapat melakukan peramalan menggunakan data produksi yang ada. Kemudian dapat ditemukan pola-pola produksi dari data tersebut. Dengan data pola pola dapat dilakukan peramalan untuk jumlah produksi.

Ada beberapa algoritma yang dapat digunakan untuk menentukan jumlah produksi, salah satunya dengan *Artificial Neural Network* (ANN) dengan metode *backpropagation*. Metode *Backpropagation Neural Network* telah digunakan untuk oleh Rini Oktavian Maru'ao dalam jurnalnya dengan judul *Neural Network Implementation in Foreign Exchange Kurs Prediction*, untuk membuat suatu aplikasi untuk memprediksi kurs valuta asing [3]. Selain itu ANN juga dapat dikombinasikan dengan beberapa metode lainnya untuk meningkatkan keakuratan hasil peramalan seperti *Particle Swarm Optimalization* (PSO). *Particle Swarm Optimalization Neural Network* dapat digunakan untuk memprediksi laju inflasi [4]. Selain PSO ada lagi algoritma yang dapat dikombinasikan,yaitu algoritma genetika. Algoritma ini dipakai untuk mendapatkan solusi optimal yang tepat untuk masalah dari satu variabel atau multi variabel. Terdapat penelitian sebelumnya mengenai Algoritma Genetika oleh Devi Rahmayanti untuk menentukan nilai optimal dalam *routing* [5] dan Algoritma Genetika juga bisa diterapkan dalam ANN[6].

Perusahaan Omega Mie Jaya adalah salah satu produsen mie yang berada di Pemangkat, Kalimantan Barat. Omega Mie Jaya belum memiliki suatu sistem yang dapat membantu untuk menetapkan jumlah produksi mie. Hal ini dikarenakan kurangnya sumber daya manusia yang mengerti tentang sistem untuk menentukan jumlah produksi. Maka perlu dibuat suatu system yang dapat meramalkan jumlah produksi mi. Dengan dibuatnya sistem ini, maka dapat perusahaan memiliki proses produksi yang semakin baik dan terencana. Dari hasil uraian tersebut, maka dibuatlah suatu aplikasi peramalan produksi mie yang akan diimplementasikan ke Perusahaan Omega Mie Jaya dengan menggunakan Algoritma Genetika Neural Network.

II. TINJAUAN PUSTAKA

A. Peramalan

Metode peramalan digunakan untuk mengukur keadaan di masa datang. Peramalan dapat dilakukan secara kualitatif dan kuantitatif. Pengukuran kuantitatif menggunakan metode statistik, sedangkan pengukuran secara kualitatif berdasarkan pendapat (*judgement*) dari yang melakukan peramalan [7]. Peramalan biasanya diklasifikasikan berdasarkan horizon waktu di masa depan yang terbagi atas beberapa kategori [8]:

1. Peramalan jangka pendek, peramalan ini mencakup jangka waktu hingga satu tahun, tetapi umumnya kurang dari tiga bulan. Peramalan ini digunakan untuk merencanakan pembelian, penjadwalan kerja, jumlah tenaga kerja, penugasan kerja, dan tingkat produksi.
2. Peramalan jangka menengahm, peramalan ini umumnya mencakup hitungan bulanan hingga tiga tahun. Peramalan ini digunakan untuk menrencanakan penjualan, perencanaan dan anggaran produksi, anggaran kas, dan menganalisis bermacam-macam rencana operasi.
3. Peramalan jangka panjang, peramalan ini umumnya untuk perencanaaan masa tiga tahun atau lebih. Peramalan ini digunakan untuk merencanakan produk baru, pembelanjaan modal, lokasi atau pengembangan fasilitas, serta penelitian dan pengembangan.

B. Produksi

Produksi adalah usaha manusia untuk mengubah serta mengolah sumber daya ekonomi menjadi bentuk serta kegunaan baru. Dengan kata lain, kegiatan produksi adalah proses mengolah produk yang dapat berupa barang dan jasa [9]. Kegiatan produksi adalah kegiatan menciptakan serta menambah nilai guna suatu barang atau jasa dengan menggunakan faktor – faktor produksi [10]. Faktor – faktor produksi antara lain sumber daya alam, sumber daya manusia, sumber daya modal, sumber daya pengusaha.

C. Neural Network

Neural Network merupakan salah satu representasi dari otak manusia yang mencoba mensimulasikan proses pembelajaran pada otak manusia [11]. *Neural Network* ini pertama kali dipresentasikan oleh Warren McCulloch dan Walter Pitt pada tahun 1943 [12]. Neural Network memiliki beberapa fitur atraktif antara lain:

- Ketahanan dan toleransi kesalahan. Rusaknya salah satu sel tidak mempengaruhi *performance* secara signifikan.
- Fleksibel. Jaringan dapat menyesuaikan diri dengan lingkungan baru tanpa perlu menggunakan program terinstruksi. - Memiliki kemampuan komputasi yang kolektif.
- Kemampuan dalam berhubungan dengan variasi data.

D. Backpropagation

Dalam *neural network* ada dua proses pembelajaran yaitu terawasi dan tidak terawasi. Pelatihan yang terawasi dilakukan dengan cara memberi *neural network* data beserta dengan output yang telah diantisipasi oleh data tersebut. *Neural network* akan melakukan proses iterasi hingga output sesuai dengan diharapkan dengan rate-error yang kecil. Ada beberapa algoritma dalam pembelajaran terawasi, salah satunya adalah backpropagation [13]. Algoritma B*ackpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*) [11]. Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi sigmoid.

Pelatihan *backpropagation* meliputi tiga fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari lapisan masukan hingga lapisan keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi.

Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di lapisan keluaran. Fase ketiga adalah fase modifikasi bobot untuk menurunkan kesalahan yang terjadi. Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan [14].

E. Algoritma Genetika

Pada tahun 1975, Holland mengembangkan gagasan ini dalam bukunya " *Adaption in natural and artificial systems*". Ia menggambarkan bagaimana menerapkan prinsip evolusi alami untuk masalah optimasi dan dibangun algoritma genetika pertama. Teori Holland telah dikembangkan lebih lanjut dan sekarang algoritma genetika berdiri sebagai alat yang ampuh untuk memecahkan masalah pencarian dan optimasi. Algoritma genetika didasarkan pada prinsip genetik dan evolusi [15]. Algoritma genetika adalah merupakan metode adaptive yang biasa digunakan untuk pencarian nilai dalam suatu masalah optimasi. Algoritma ini adalah salah satu evolusi yang paling populer algoritma di mana populasi individu berkembang sesuai dengan set aturan seperti seleksi, crossover dan mutasi [16].

F. Algoritma Genetika dan Neural Network

Algoritma genetika dapat digunakan untuk meningkatkan kinerja ANN dengan banyak cara yang

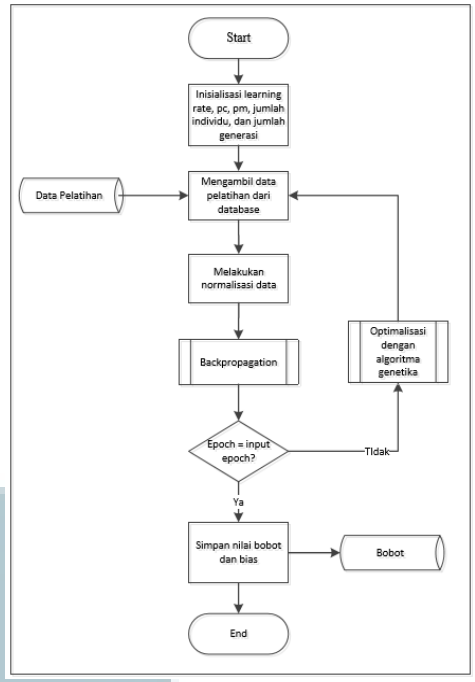
berbeda. Algoritma genetika adalah metode pencarian umum stochastic, yang dapat digunakan dengan *backpropagation neural network* untuk menentukan jumlah node tersembunyi dan lapisan tersembunyi, pilih subset fitur yang relevan, tingkat pembelajaran, momentum, dan menginisialisasi serta mengoptimalkan bobot koneksi jaringan dari backpropagation neural network. Algoritma genetika telah digunakan untuk secara optimal merancang parameter neural network termasuk ANN arsitektur, bobot, pilihan input, fungsi aktivasi, jenis neural network, algoritma pelatihan, jumlah iterasi, dan rasio partisi data [17] [18].

Proses algoritma GA - NN untuk proses peramalan ini adalah sebagai berikut:

1. Inisialisasi count = 0 , fitness = 0 , jumlah siklus
2. Generasi populasi awal . Kromosom individu dirumuskan sebagai urutan gen berturut-turut, masing-masing pengkodean input.
3. Desain jaringan yang cocok
4. Menetapkan bobot
5. Melakukan training dengan backpropagation
6. Cari kesalahan kumulatif dan nilai fitness. Kemudian dievaluasi berdasarkan nilai fitness.
7. Jika fitness sebelumnya < nilai fitness saat ini, simpan nilai saat ini
8. Count = count +1
9. Seleksi : Dua induk dipilih dengan menggunakan mekanisme wheel roulette
10. Operasi Genetik : crossover, mutasi dan reproduksi untuk menghasilkan fitur baru set
11. Jika (jumlah siklus < = count) kembali ke nomor empat
12. Pelatihan jaringan dengan fitur yang dipilih
13. Studi kinerja dengan data uji.

III. PERANCANGAN DAN IMPLEMENTASI

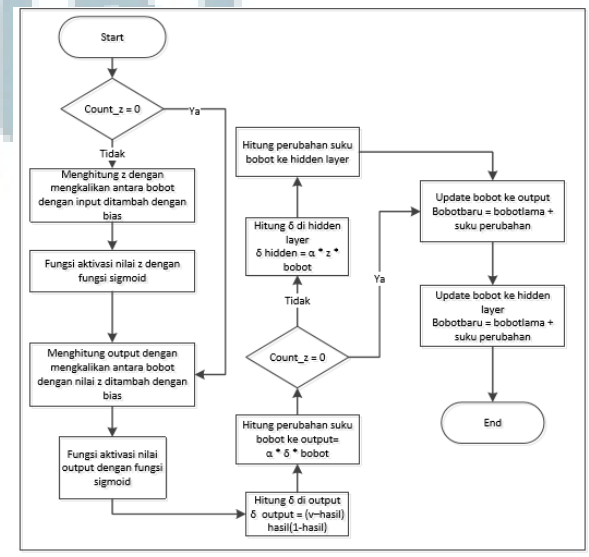
Berikut ini akan digambarkan flowchart sistem peramalan produksi :



Gambar 1. Flowchart Pelatihan

Sistem terdiri dari dua proses utama yaitu pelatihan dan peramalan. Untuk dapat melakukan suatu peramalan menggunakan *neural network*, diperlukan suatu proses pelatihan. Proses pelatihan digabung dengan algoritma genetika untuk mendapatkan bobot yang lebih optimal.

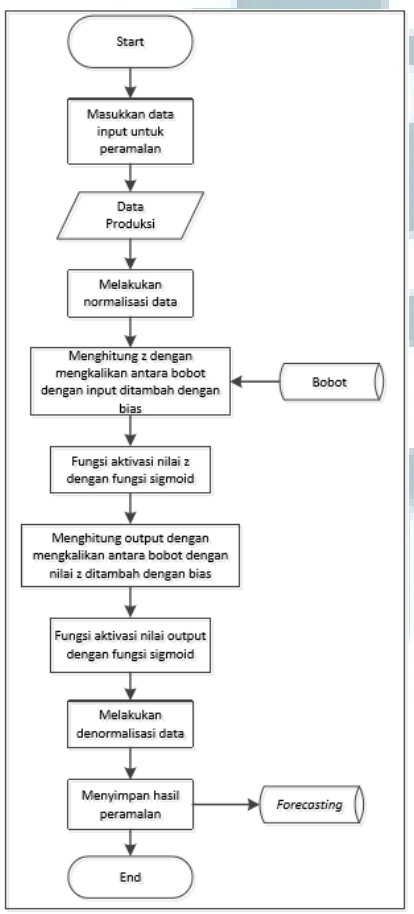
Langkah pertama tentunya kita menginisialisasi semua nilai yang akan digunakan selama proses pelatihan. Kemudian kita mengambil data-data yang digunakan untuk proses pelatihan yang diambil dari database. Setelah itu menormalisasi data tersebut untuk menyesuaikan antara range input-output dengan fungsi aktivasinya. Langkah selanjutnya adalah masuk ke backpropagation sebagai berikut :



Gambar 2. *Proses Backpropagation*

Pada subproses ini, dilakukan penghitungan nilai neuron di *hidden layer* dan ditambah dengan biasnya. Kemudian hasil yang didapat diberi fungsi aktivasi sigmoid. Hal ini juga dilakukan di *layer output*. Proses tersebut merupakan fase pertama dari *backpropagation*. Fase kedua adalah mencari δ di *layer output* dan menghitung suku perubahan ke *layer output*. Setelah itu, juga mencari δ di *hidden layer* dan menghitung suku perubahan ke *hidden layer*. Lalu meng-update ke *hidden layer* dan ke *layer output*.

Proses peramalan dimulai ketika user memasukkan data input produksi perbulan selama 12 bulan. Lalu sistem akan melakukan normalisasi data tersebut. Kemudian akan dilakukan *feedforward* dengan menggunakan bobot terbaik yang didapat dari proses pelatihan sebelumnya. Setelah nilai output berhasil dihitung, maka akan didenormalisasikan lagi sehingga akan menampilkan dan menyimpan data peramalan tersebut. Beikut ini adalah gambarnya dalam bentuk flowchart.



Gambar 3. Proses Peramalan Dengan *Backpropagation*

IV. UJI COBA

Uji coba dilakukan untuk menentukan parameter-parameter yang tepat untuk digunakan dalam aplikasi ini. Caranya adalah dengan melakukan uji coba terhadap parameter-parameter tersebut dengan menggunakan data – data produksi Omega Mie Jaya. Parameter-parameter yang tepat adalah parameter-parameter yang saat dilakukan pelatihan dan testing memiliki nilai error yang terkecil dengan data validasi.

Tabel 1. *Learning Data*

Bulan1	Bulan2	Bulan3	Bulan4	Bulan5	Bulan6	Bulan7	Bulan8	Bulan9	Bulan10	Bulan11	Bulan12	Target
2816	2992	3366	3960	4048	4576	5324	5258	4774	4554	3690	3168	2508
2992	3366	3960	4048	4576	5324	5258	4774	4554	3690	3168	2508	3916
3366	3960	4048	4576	5324	5258	4774	4554	3690	3168	2508	3916	4026
3960	4048	4576	5324	5258	4774	4554	3690	3168	2508	3916	4026	4334
4048	4576	5324	5258	4774	4554	3690	3168	2508	3916	4026	4334	5478
4576	5324	5258	4774	4554	3690	3168	2508	3916	4026	4334	5478	5852
5324	5258	4774	4554	3690	3168	2508	3916	4026	4334	5478	5852	6600
5258	4774	4554	3690	3168	2508	3916	4026	4334	5478	5852	6600	5874
4774	4554	3690	3168	2508	3916	4026	4334	5478	5852	6600	5874	5082
4554	3690	3168	2508	3916	4026	4334	5478	5852	6600	5874	5082	4598
3690	3168	2508	3916	4026	4334	5478	5852	6600	5874	5082	4598	3872
3168	2508	3916	4026	4334	5478	5852	6600	5874	5082	4598	3872	3520
2508	3916	4026	4334	5478	5852	6600	5874	5082	4598	3872	3520	2684
3916	4026	4334	5478	5852	6600	5874	5082	4598	3872	3520	2684	4094
4026	4334	5478	5852	6600	5874	5082	4598	3872	3520	2684	4094	4378
4334	5478	5852	6600	5874	5082	4598	3872	3520	2684	4094	4378	4752
5478	5852	6600	5874	5082	4598	3872	3520	2684	4094	4378	4752	5682
5852	6600	5874	5082	4598	3872	3520	2684	4094	4378	4752	5682	5984
6600	5874	5082	4598	3872	3520	2684	4094	4378	4752	5682	5984	6798
5654	5082	4598	3872	3520	2684	4094	4378	4752	5682	5984	6798	6116

Berdasarkan data – data yang ada, terdapat 36 data yang membuat 24 pola data. 24 pola data dibagi menjadi dua bagian yaitu 20 pola data (tabel 1) untuk data pelatihan dan 4 data digunakan untuk testing (tabel 2).

Tabel 2. *Testing Data*

Bulan1	Bulan2	Bulan3	Bulan4	Bulan5	Bulan6	Bulan7	Bulan8	Bulan9	Bulan10	Bulan11	Bulan12	Target
5082	4598	3872	3520	2684	4094	4378	4752	5684	5984	6798	6116	5214
4598	3872	3520	2684	4094	4378	4752	5684	5984	6798	6116	5214	4686
3872	3520	2684	4094	4378	4752	5684	5984	6798	6116	5214	4686	4202
3520	2684	4094	4378	4752	5684	5984	6798	6116	5214	4686	4202	3612

Uji coba dimulai dari menentukan arsitektur neural network. Jaringan memiliki tiga bagian, yaitu input, *hidden layer*, dan juga output. Jadi kita menentukan jumlah neuron terbaik yang nantinya akan digunakan untuk aplikasi ini. Tahap kedua adalah menetapkan parameter-parameter genetika. Parameter algoritma genetika akan diuji coba pada tahapan berikutnya sehingga pada tahap ini nilainya belum diubah–ubah. Ketika sudah mendapatkan arsitektur jaringan terbaik, maka dilakukan proses uji coba untuk parameter algoritma genetika. Proses uji coba tahap dilakukan dengan menentukan jumlah neuron kemudian melakukan proses pelatihan dengan cara inisialisasi bobot dengan angka random.

Kemudian dihitung nilai neuron dan juga bobotnya dengan menggunakan data pelatihan yang telah dinormalisasi terlebih dahulu. Proses tersebut merupakan proses yang dilakukan pada tahap fungsi evaluasi dalam algoritma genetika. Kemudian masuk ke tahap seleksi, crossover, dan mutasi. Setelah proses pelatihan, maka dilakukan testing feedforward dan

dihitung nilai error antara data yang dihasilkan program dengan data validasi. Variabel awal alogritma genetika yang digunakan adalah enam inidividu, sepuluh generasi, 0,6 peluang crossover, 0,1 peluang mutasi.

Tabel 1. Percobaan Mencari Parameter

Learning Rate	Arstitektur	Error Peramalan
0,05	12 – 3 – 1	24,1 %
0,1	12 – 3 – 1	26,9 %
0,2	12 – 3 – 1	22,5 %
0,05	12 – 5 – 1	19,3 %
0,1	12 – 5 – 1	21,4 %
0,2	12 – 5 – 1	17,5 %
0,05	12 – 7 – 1	20,5 %
0,1	12 – 7 – 1	28,7 %
0,2	12 – 7 – 1	29,3 %
0,1	12 – 5 – 2 – 1	18,6 %
0,2	12 – 5 – 2 – 1	18,4 %
0,05	12 – 5 – 2 – 1	19,5 %
0,2	12 – 5 – 3 – 1	19,2 %
0,1	12 – 5 – 3 – 2 – 1	18,1 %
0,2	12 – 5 – 3 – 2 – 1	17,8 %
0,2	12 – 5 – 3 – 3 – 1	17,3 %
0, 1	12 – 5 – 3 – 3 – 1	19, 4 %

Berdasarkan tabel diatas dapat disimpulkan bahwa arsitektur yang memiliki error yang terkecil adalah arsitektur dengan konfigurasi jaringan 12 – 5 – 3 – 3 – 1 yang memiliki tingkat akurasi mencapai 17,3 %.

Setelah melakukan tahap pertama, masuk ke tahap berikutnya yaitu ujicoba variabel algoritma genetika untuk mendapatkan nilai error yang terkecil. Pada tahap ini kita berusaha mendapatkan jumlah individu, jumlah Generasi peluang crossover, dan peluang mutasi yang terbaik dengan cara menghitung error dari bobot yang dihasilkan dalam proses genetika dengan data validasi. Dari hasil uji coba berikutnya, parameter untuk algoritma genetika yang terbaik adalah 10 individu, 50 generasi, 0.8 peluang crossover, dan 0.1 peluang mutasi dapat mencapai error sampai dengan 14%. Hal ini menunjukkan bahwa perubahan parameter algoritma genetika juga mempengaruhi penurunan tingkat error dari hasil peramalan produksi.

V. SIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa aplikasi peramalan produksi mie untuk Omega Mie Jaya dengan menggunakan algoritma genetika dan *neural network* berhasil dilakukan. Aplikasi peramalan ini memiliki arsitektur jaringan berupa 12 input, tiga jaringan *hidden layer* dimana layer pertama terdiri dari 5 neuron, layer kedua memiliki tiga neuron dan lapisan terakhir mempunyai tiga neuron, dan satu buah output dengan laju pembelajaran yang digunakan adalah 0,2. Parameter yang digunakan algoritma genetika adalah sepuluh individu, 0,8 peluang crossover, 0,1 peluang mutasi, dan 50 generasi. Dari uji coba yang dilakukan, aplikasi peramalan ini memiliki tingkat akurasi peramalan mencapai 86%.

DAFTAR PUSTAKA

[1] Gilarso, T. 2004. "Pengantar Ilmu Ekonomi Makro". Yogyakarta: Kanisius.

[2] Gasperz, Vincent. 2002. "Total Quality Management". Jakarta: PT Gramedia Pustaka Utama .

[3] Maru'ao, Dini Oktaviani. 2010. "Neural Network Implementation in Foreign Exchange Kurs Prediction". Universitas Gunadharma.

[4] Raharjo, Joko S. Dwi. 2013. "Model Artificial Neural Network Berbasis Particle Swarm Optimization untuk Prediksi Laju Inflasi". Dalam Jurnal Sistem Komputer Volume 3 hlm. 11-21.

[5] Rahmayanti, Devi. 2010. "Optimasi Routing Berbasis Algoritma Genetika pada Sistem Komunikasi Bergerak". Dalam Jurnal EEECIS Volume IV hlm. 18-23.

[6] Dhanwani, Deepak dan Avinash Wadhe. 2013. "Study of Hybrid Genetic Algorithm Using Artificial Neural Network In Data Mining For The Diagnoses of Stroke Disease" Dalam International Journal of Computational Engineering Research, hlm 95-100.

[7] Herjanto, Eddy. 2008. "Manajemen Operasi (Edisi Ketiga)". Jakarta: Grasindo.

[8] Prasetya, Heri dan Fitri Lukiastuti. 2009. "Manajemen Operasi". Yogyakarta: MedPress.

[9] Ahman, Eeng dan Epi Indriani. 2007. "Membina Kompetensi Ekonomi". Bandung: Grafindo Media Pratama.

[10] Nafari, M. 2007. "Penganggaran Peusahaan". Jakarta: Salemba Empat.

[11] Kusumadewi, S. 2004. "Membangun Jaringan Syaraf Tiruan Menggunakan Matlab dan Excel Link". Yogyakarta: Graha Ilmu.

[12] Rojas, Raul. 1996. "Neural Networks". New York: Springer-Verlaag Berlin Heidelberg.

[13] Heaton, Jeff. 2008. "Introduction to Neural Network with Java (Second Edition)". Heaton Research.

[14] Siang, J.J. 2005. "Jaringan Syaraf Tiruan dan Pemogramannya Dengan Matlab". Yogyakarta: Penerbit Andi.

[15] Sivanandam, S.N dan S.N. Deepa. 2008. "Introduction to Genetic Algortihm". New York: Springer-Verlaag Berlin Heidelberg.

[16] Suyanto. 2005. "Algoritma Genetika Dalam Matlab". Yogyakarta: Andi Offset.

[17] Lanaraga, P. dkk. 1997. "Learning Bayesian networks by genetic algorithms: a case study in the rediction of survival in malignant skin melanoma".

[18] Dhanwani, Deepak dan Avinash Wadhe. 2013. "Study of Hybrid Genetic Algorithm Using Artificial Neural Network In Data Mining For The Diagnoses of Stroke Disease" Dalam International Journal of Computational Engineering Research, hlm 95-100.

ISSN 2085-4552

Rancang Bangun Sistem Rekomendasi Televisi LED Dengan Metode Vikor Berbasis Web

Boyma Simamora
Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
boyma.simamora@student.umn.ac.id

Diterima 19 Mei 2017
Disetujui 19 Juni 2017

***Abstract*-Beragam macam pilihan televisi LED membuat pembeli mengalami kebingungan saat menentukan tipe televisi LED yang akan dibeli. Beberapa merk televisi terkenal menawarkan macam-macam tipe televisi LED dengan keunggulannya masing-masing. Perbedaan dapat dilihat dari segi harga, ukuran layar, resolusi layar, berat dan fasilitas pendukung lain yang diberikan. Sehingga dibuatlah sistem rekomendasi pembelian televisi LED untuk memberikan masukan dan bantuan kepada calon pembeli sebagai bahan pertimbangan sesuai dengan keinginannya. Penggunaan metode VIKOR diterapkan karena termasuk jenis dari *Multi Criteria Decision Making* untuk mengambil keputusan bersifat diskrit berdasarkan beberapa kriteria dan membantu pengambil keputusan sebagai hasil terbaik dari rekomendasi sistem. Kriteria yang digunakan yaitu rentang harga, merk, resolusi layar, ukuran layar, berat dan fasilitas sebagai deskripsi televisi. Pembuatan sistem dibangun dengan menggunakan bahasa pemrograman PHP dan *database* MySQL. Uji Kepuasan Pengguna Sistem dilakukan dengan melakukan survei kepada 30 responden dengan menggunakan metode *Likert* dan *Cronbach Alpha*. Hasil yang diperoleh sebesar 0,72 dan dapat disimpulkan dalam kategori cukup baik.**

Index Terms*-Televisi LED, Sistem Rekomendasi, VIKOR, Skala *Likert, Cronbach Alpha.

I. PENDAHULUAN

Kebutuhan akan adanya informasi, hiburan dan pendidikan diperlukan untuk menambah pengetahuan seseorang. Hal itu didapat dari salah satu alat elektronik yaitu televisi. Televisi adalah sistem elektronik yang mengirimkan gambar diam dan gambar hidup bersama suara melalui kabel dan ruang [1]. Televisi dapat dimanfaatkan untuk keperluan pendidikan, yang sangat mudah dijangkau melalui siaran udara.

Media televisi dinilai sebagai media yang paling sering diakses oleh masyarakat Indonesia. Penelitian dilakukan oleh MarkPlus mengenai media yang paling sering diakses oleh kaum muda pada 6 bulan terakhir pada 2014 di 10 Kota di Indonesia [2]. Hasil menunjukkan bahwa media yang paling sering diakses adalah media televisi dengan perolehan sebesar 100%, ini menunjukkan bahwa seluruh responden mengakses televisi.

Menurut lembaga survei AC Nielsen [3], tingkat konsumsi media di lima kota besar di luar Pulau Jawa lebih tinggi dibandingkan kota besar di Pulau Jawa. Secara keseluruhan menunjukkan bahwa televisi masih menjadi medium utama yang dikonsumsi masyarakat Indonesia di perkotaan khususnya pada Pulau Jawa maupun luar Jawa, yaitu sebesar 95%, disusul oleh internet dengan 33%, radio 20%, surat kabar 12%, tabloid 6% dan majalah 5%.

Kemudian, pada 2014 KOMINFO mengadakan survei akses dan pengguna indikator TIK sektor rumah tangga. Sampel dalam survei ini berjumlah 9.680 rumah tangga dengan tingkat keyakinan 95% dan *margin of error estimation* sekitar 1%. Berdasarkan hasil survei akses TIK di rumah tangga, persentase kepemilikan radio di rumah tangga Indonesia hanya sebesar 27,2%, telepon kabel 5,8% dan mayoritas rumah tangga Indonesia memiliki televisi sebesar 87,20% [4]. Hal tersebut menunjukkan bahwa televisi merupakan perangkat TIK paling banyak dimiliki dan penting untuk masyarakat Indonesia.

Televisi sebagai salah satu media elektronik yang digunakan masyarakat Indonesia berkembang seiring dengan perkembangan teknologi dan kebutuhan akan televisi. Pada tahun 2006 diproduksi televisi LED berbasis DLP *(Digital Light Processing)* HDTV pertama. televisi LED *(Light Emitting Diodes)* merupakan pengembangan dari LCD *(Liquid Crystal Display)* TV yang menggunakan LED *Backlight* sebagai pengganti cahaya *fluorescent* pada LCD TV [5].

Secara umum televisi LED menawarkan kualitas gambar yang baik khususnya, *contrast*, konsumsi listrik yang lebih sedikit dibandingkan LCD dan ramah lingkungan. Televisi LED memungkinkan produsen untuk memproduksi televisi dengan layar yang berukuran tipis. Sehingga televisi LED menjadi salah satu produk pilihan perangkat elektronik yang paling digemari oleh banyak orang.

Namun sering sekali calon konsumen mengalami kebingungan saat memilih tipe televisi LED yang akan dibeli [6]. Adapun faktor yang menyebabkan

kebingungan dalam memilih televisi LED adalah adanya beberapa merk yang menawarkan macam-macam tipe televisi dengan berbagai macam spesifikasi televisi (*inches), resolusi, USB, HDMI, output audio, VGA output,* TV *System*, harga, dan daya (*watt)* [6]. Oleh karena itu, dibutuhkan sebuah sistem rekomendasi pembelian televisi LED untuk membantu calon konsumen membeli televisi LED sesuai dengan kebutuhan dari kriteria-kriteria yang diinginkan. Kriteria-kriteria yang digunakan yaitu harga, merk, resolusi layar, ukuran layar, berat dan fasilitas berdasarkan hasil kuesioner dari Sistem Pendukung Keputusan untuk Merekomendasikan TV Layar Datar Menggunakan Metode *Weighted Product* [7].

Metode VIKOR merupakan metode *Multi-Criteria Decision Making* (MCDM) yang dapat digunakan untuk menyeleksi lebih dari satu kriteria, kemudian akan dilakukan proses menyeleksi hasil yang terbaik. Sehingga diharapkan dapat membantu proses pembelian televisi LED yang tepat sesuai kriteria-kriteria yang telah ditentukan.

Terdapat beberapa penelitian sebelumnya yang telah menggunakan metode VIKOR dalam pemilihan robot industri untuk penggunaan secara spesifik pada aplikasi teknik [8]. Pemilihan robot industri merupakan salah satu masalah yang paling menantang pada saat proses menentukan keputusan robot industri yang paling tepat sesuai tujuan dan biaya seminimal mungkin dan kemampuannya secara spesifik. Dari hasil penelitian tersebut menunjukkan metode VIKOR mampu memilih robot yang tepat berdasarkan hasil seleksi. Lalu, penelitian implementasi metode VIKOR untuk seleksi penerimaan beasiswa yang menunjukkan bahwa metode VIKOR dapat membantu proses seleksi dan menentukan penerimaan beasiswa yang tepat, serta membuat hasil terbaik kompromi alternatif dari sejumlah alternatif [9].

Penggunaan metode VIKOR untuk pembangkit listrik tenaga air (PLTA) berkelanjutan dengan 56 kriteria disederhanakan menjadi 23 kriteria kemudian menggunakan metode VIKOR untuk membuat perankingan terhadap alternatif yang ada. Hasil penelitian menunjukkan bahwa metode VIKOR dapat membantu pembuatan keputusan dalam proses desain, menyediakan dukungan untuk pemilihan tempat dan parameter operasi [10]. Pada penelitian yang membandingkan antara metode TOPSIS dan VIKOR menyimpulkan bahwa metode VIKOR lebih mendekati titik solusi ideal dengan menggunakan normalisasi linear dibandingkan dengan metode TOPSIS yang menggunakan normalisasi vektor [11].

Berdasarkan pemaparan-pemaparan tersebut maka, dipilih metode VIKOR untuk digunakan dalam penelitian pada sistem rekomendasi pembelian televisi LED. Penggunaan metode VIKOR diharapkan dapat membantu calon konsumen dapat dalam memilih televisi LED yang sesuai dengan kriteria dan kebutuhannya.

II. KAJIAN PUSTAKA

A. *Sistem Rekomendasi*

Sistem rekomendasi adalah sebuah sistem yang dapat memberikan rekomendasi kepada para pengguna sistem yang akan dibuat [12]. Rekomendasi yang diberikan dapat berdasarkan karakteristik dari data pengguna tersebut. Sedangkan dalam mengumpulkan data untuk pembuatan sistem rekomendasi dapat dilakukan secara langsung dan tidak langsung [13].

B. *Multi Criteria Decision Making (MCDM)*

Multi-Criteria Decision Making (MCDM) adalah suatu metode pengambilan keputusan untuk menetapkan alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu. Kriteria biasanya berupa ukuran-ukuran atau aturan-aturan atau standar yang digunakan dalam pengambilan keputusan. Secara umum, tujuan MCDM menyeleksi alternatif terbaik dari sejumlah alternatif [14]. Berdasarkan tujuannya metode MCDM memiliki dua model yaitu *Multi Attribute Decision Making* (MADM) untuk menyeleksi alternatif terbaik dari sejumlah alternatif dan *Multi Objective Decesion Making* (MODM) untuk merancang alternatif terbaik [15].

C. *Metode VIKOR*

VIKOR (*VlseKriterijumska Optimizacija I Kompromisno Resenje* dalam bahasa Serbia, yang artinya *Multicriteria Optimization dan Compromise Solution)* adalah metode perankingan dengan menggunakan indeks peringkat multikriteria berdasarkan ukuran tertentu dari kedekatan dengan solusi yang ideal. Metode VIKOR merupakan salah satu metode yang dapat dikategorisasikan dalam *Multicriteria decision analysis* [17]. Metode VIKOR dikembangkan sebagai metode *multicriteria decision making* untuk menyelesaikan pengambilan keputusan bersifat diskrit pada kriteria yang bertentangan dan *non-commensurable* (tidak ada cara yang tepat untuk menentukan mana yang lebih akurat) [18].

Metode VIKOR fokus pada perankingan dan memilih dari satu set sampel dengan kriteria yang saling bertentangan, yang dapat membantu para pengambil keputusan untuk mendapatkan keputusan akhir [18]. Metode ini sangat berguna pada situasi dimana pengambil keputusan tidak memiliki kemampuan untuk menentukan pilihan pada saat desain sebuah sistem dimulai [19].

Metode VIKOR adalah sebuah metode untuk optimisasi/optimalisasi kriteria majemuk dalam suatu sistem yang kompleks [20]. Konsep dasar VIKOR adalah menentukan ranking dari sampel-sampel yang ada dengan melihat hasil dari nilai-nilai sesalan atau *regrets* (R) dari setiap sampel. Metode VIKOR telah digunakan oleh beberapa peneliti dalam MCDM, seperti dalam pemilihan vendor [21], perbandingan metode-metode *outranking* [18], pemilihan bahan dalam industry [22]. Masih banyak penelitian-penelitian yang menggunakan metode VIKOR ini.

Langkah-langkah yang digunakan dalam Metode VIKOR adalah sebagai berikut [23]:

1. Tabel pengamatan

Dari data yang didapat pada *database* digunakan menjadi data untuk tabel pengamatan, lalu menentukan nilai data terbaik (f_i^*) dan terburuk (f_i^-)atau dengan istilah *Cost* dan *Benefit* dalam satu variabel penelitian. Penentuan data terbaik dan terburuk ditentukan oleh jenis data variabel penelitian *higher-the-better (HB)* atau *lower-the-better (LB)* [23].

2. Bobot kriteria

Menentukan bobot kriteria yang diperoleh dari pengguna sistem sesuai dengan kebutuhan atau kriteria yang diinginkan.

3. Normalisasi matriks [23].

$$R_{ij} = \frac{(f_i^*) - (f_{ij})}{(f_i^*) - (f_i^-)} \quad \text{…(Rumus 2.1)}$$

dimana :
R_{ij} = nilai normalisasi sampel *i* kriteria *j*
f_{ij} = nilai data sampel *i* kriteria *j*
f_i^* = nilai terbaik dalam satu kriteria
f_i^- = nilai terjelek dalam satu kriteria

4. Normalisasi bobot $(W_j \text{ x } R_{ij})$

Melakukan perkalian antara nilai data yang telah dinormalisasi dengan nilai bobot kriteria yang telah ditentukan.

5. Menghitung nilai *Utility Measure* (*S*) dan *Regret Measure* (*R*) [11]

$$S_j = \sum_{j=1}^{n} W_i \left(\frac{(f_i^*) - (f_{ij})}{(f_i^*) - (f_i^-)}\right) \quad \text{…(Rumus 2.2)}$$

$$R_j = \text{Max}_j \left[W_i \left(\frac{(f_i^*) - (f_{ij})}{(f_i^*) - (f_i^-)}\right)\right] \quad \text{…(Rumus 2.3)}$$

dimana :
W_j = bobot kriteria

6. Menghitung indeks VIKOR [11]

$$Q_j = \left[\frac{S_j - S^*}{S^- - S^*}\right] \text{ x } v + \left[\frac{R_j - R^*}{R^- - R^*}\right] \text{ x } (1\text{-}v) \quad \text{…(Rumus 2.4)}$$

dimana :
S^* = nilai S terkecil
S^- = nilai S terbesar
R^* = nilai R terkecil
R^- = nilai R terbesar

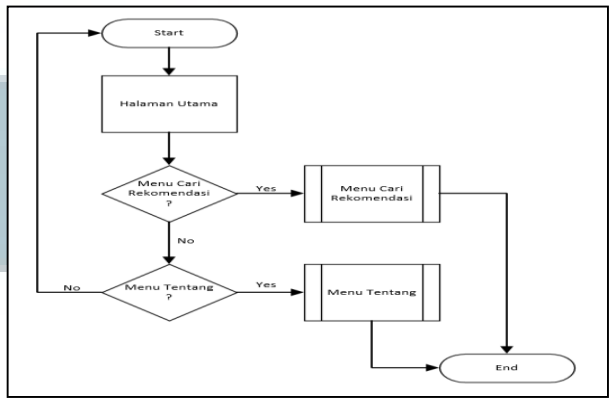
7. Perankingan alternatif [11]

Setelah Q_j dihitung, maka Pengurutan perankingan ditentukan dari nilai yang paling rendah dengan solusi kompromi sebagai solusi ideal dilihat dari perankingan Q_j dengan nilai terendah. Karena nilai S_j merupakan solusi yang diukur dari titik terjauh solusi ideal, sedangkan nilai R_j merupakan solusi yang diukur dari titik terdekat solusi ideal.

III. METODE DAN PERANCANGAN SISTEM

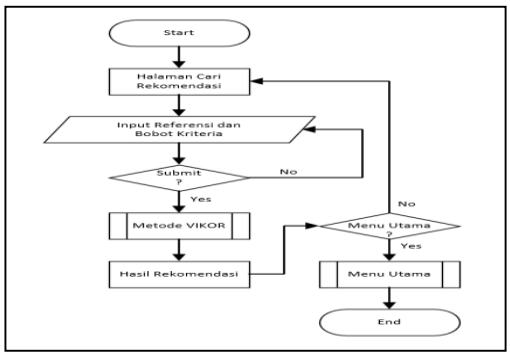
A. *Flowchart*

Flowchart atau diagram alur adalah bagan-bagan yang memiliki arus untuk menggambarkan langkah-langkah dan proses pada suatu sistem. *Flowchart* pada sistem ini terdiri dari dua bagian yaitu *flowchart* untuk admin (*backend*) dan *flowchart* untuk *user* (*frontend*). Pada Gambar 3.1 menunjukkan alur proses pertama pada sistem rekomendasi ketika pertama kali membuka sistem ini. Pada Halaman Utama terdapat dua menu yang dapat digunakan oleh *user* yaitu, Menu Cari Rekomendasi dan Menu Tentang.



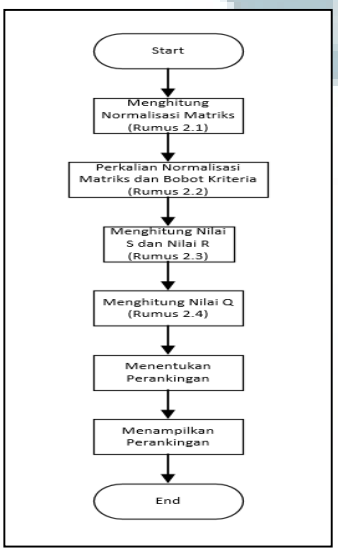
Gambar 3.1 *Flowchart* Menu Utama

Pada Gambar 3.2 menunjukkan alur proses yang dilakukan sistem ketika *user* melakukan proses cari rekomendasi. Dimulai dengan memilih referensi dan memasukkan *input* bobot kriteria kemudian nilai bobot akan masuk ke *database* dan dilakukan perhitungan dengan menggunakan metode VIKOR yang menghasilkan hasil rekomendasi berupa daftar televisi LED yang direkomendasikan oleh sistem.



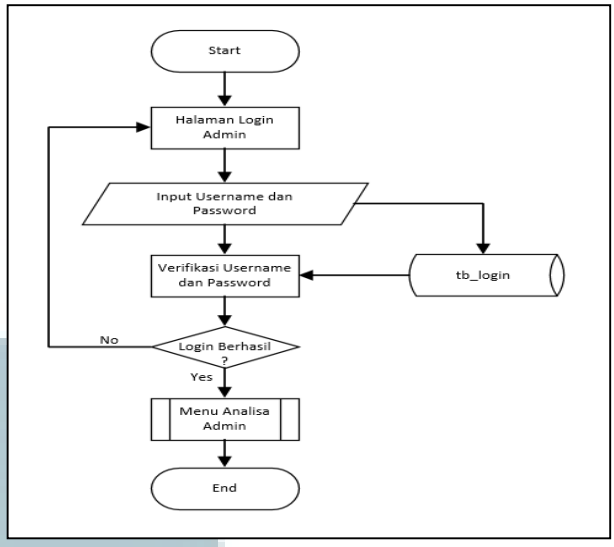
Gambar 3.2 *Flowchart* Menu Cari Rekomendasi

Pada Gambar 3.3 menjelaskan alur proses metode VIKOR. Awal proses dimulai dari inputan bobot kriteria dari *user*, menentukan data alternatif, menentukan nilai terbaik dari data alternatif dan terburuk dari data alternatif televisi LED. Dari data tersebut dilakukan normalisasi matriks pada data televisi dengan rumus, hasil normalisasi dikalikan dengan nilai bobot kriteria yang diinginkan dengan rumus. Dari perhitungan tersebut diperoleh nilai S dan nilai R dengan rumus. Dari nilai S dan nilai R maka kita dapat mencari nilai Q dengan rumus. Hasil nilai Q yang didapat diurutkan berdasarkan dari nilai yang paling terkecil, untuk menentukan peringkat terbaik dari hasil rekomendasi menggunakan metode VIKOR.



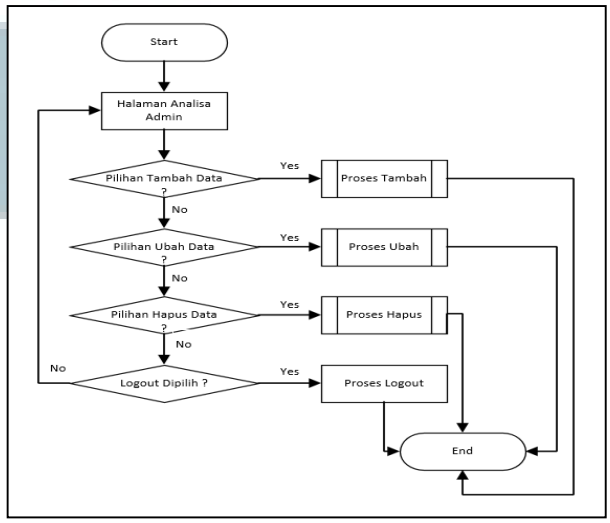
Gambar 3.3 *Flowchart* Metode VIKOR

Pada Gambar 3.4 menunjukkan alur proses saat *admin* memilih Menu *Login* pada sistem. Pada Menu *Login* harus memasukkan *username* dan *password* yang benar, setelah akan diverifikasi dari *database* tabel tb_login. Menu Analisa hanya dapat digunakan oleh *admin* untuk menambah, mengubah dan menghapus data produk televisi yang menampung perubahan ke dalam *database*, serta menampilkan semua data produk televisi.



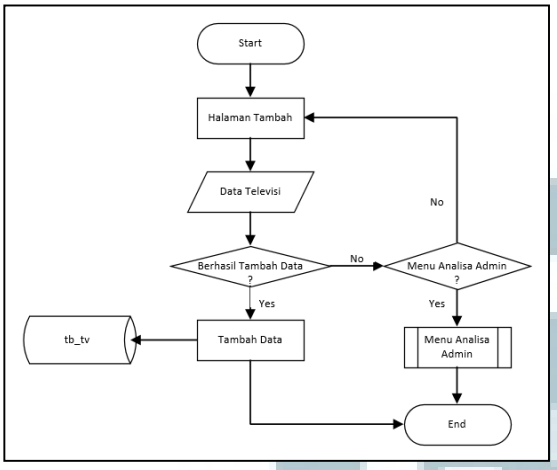
Gambar 3.4 *Flowchart* Menu *Login Admin*

Pada Gambar 3.5 menunjukkan alur proses saat *admin* memilih Menu Analisa pada sistem. Pada halaman Analisa terdapat tiga subproses yang dapat *admin* lakukan yaitu, Proses Tambah, Proses Ubah, dan Proses Hapus. Pada Proses Tambah dapat menambahkan data televisi, pada Proses Ubah dapat merubah data televisi yang sudah ditampilkan di halaman Analisa, dan untuk Proses Hapus dapat digunakan jika *admin* ingin menghapus data televisi yang ada pada halaman Analisa dan perubahan akan disimpan ke dalam *database.*



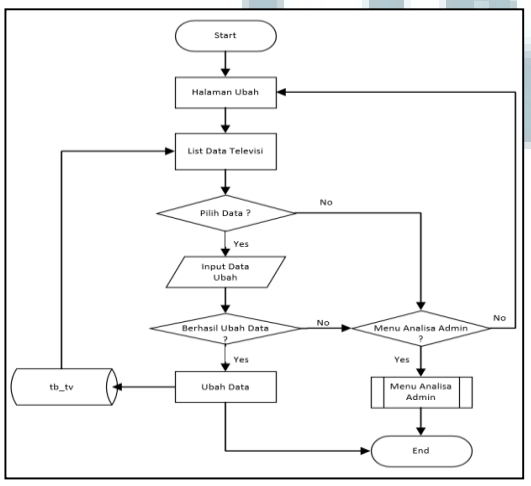
Gambar 3.5 *Flowchart* Menu Analisa *Admin*

Pada Gambar 3.6 menunjukkan alur proses saat *admin* memilih proses Tambah pada sistem. Pertama akan ditampilkan halaman Tambah, kemudian *admin* dapat melakukan tambah data dengan malakukan *input* data televisi yang hanya dapat dilakukan oleh *admin*, jika berhasil melakukan penambahan data maka data yang telah diinput akan disimpan ke *database*.



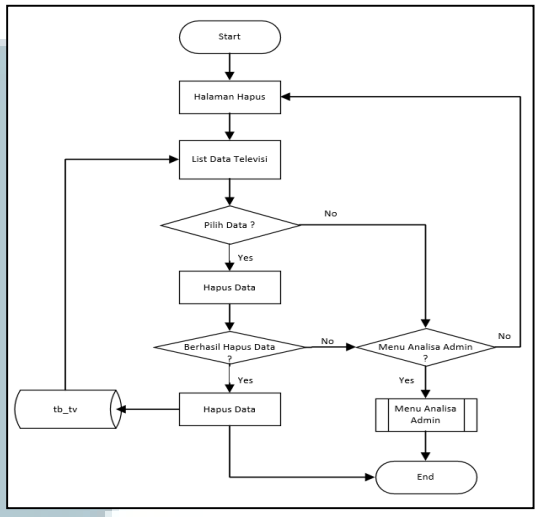
Gambar 3.6 *Flowchart* Proses Tambah

Pada Gambar 3.7 menunjukkan alur proses saat *admin* memilih proses Ubah. Pertama akan ditampilkan halaman Ubah yang berisi data televisi. Kemudian *admin* dapat melakukan Ubah data dengan memilih terlebih dahulu data yang ingin diubah dari tampilan daftar list televisi. Setelah data berhasil diubah maka data tersebut akan disimpan ke *database* pada tabel tb_tv. Selain itu terdapat pilihan untuk kembali ke Menu Analisa jika tidak menggunakan Proses Tambah, sehingga dapat memilih proses yang lain yang terdapat pada halaman Analisa pada sistem ini.



Gambar 3.7 *Flowchart* Proses Ubah

Pada Gambar 3.8 menunjukkan alur proses saat *admin* memilih proses Hapus pada sistem. Pertama akan ditampilkan halaman Hapus yang berisi data televisi. Kemudian *admin* dapat melakukan Hapus data dengan memilih terlebih dahulu data yang ingin dihpaus dari tampilan daftar list televisi. Setelah data berhasil dihapus maka data tersebut akan disimpan ke *database* pada tabel tb_tv. Selain itu terdapat pilihan untuk kembali ke Menu Analisa jika tidak menggunakan Proses Tambah, sehingga dapat memilih proses yang lain yang terdapat pada halaman Analisa pada sistem ini.

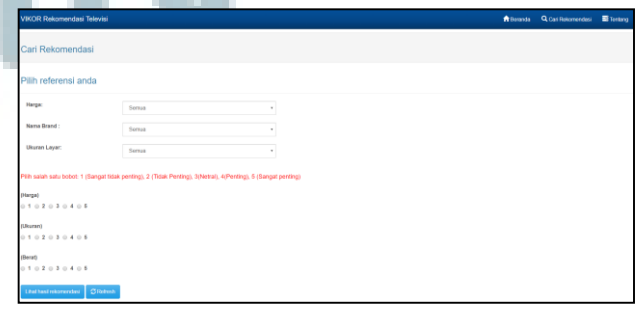


Gambar 3.8 *Flowchart* Proses Hapus

IV. IMPLEMENTASI DAN UJI COBA

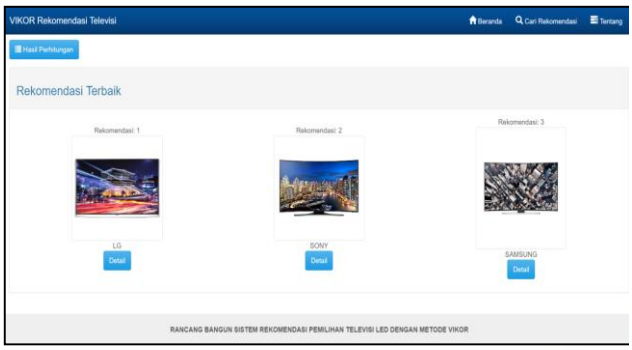
A. *Implementasi Sistem*

Halaman Cari Rekomendasi merupakan halaman utama untuk *user* yang digunakan mencari rekomendasi televisi LED. *User* dapat memilih referensi dan kriteria yang diinginkan.



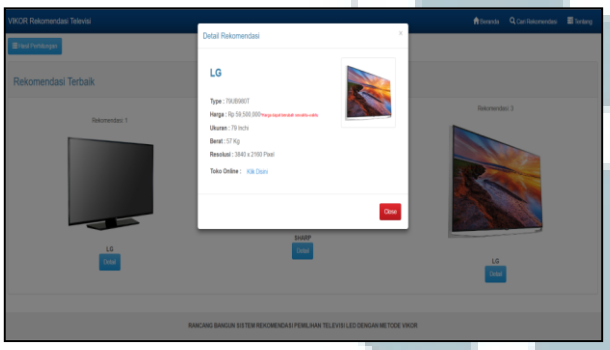
Gambar 4.1 *Interface* Halaman Cari Rekomendasi

Hasil dari perhitungan referensi dan kriteria yang dimasukkan *user*, menampilkan hasil rekomendasi tiga terbaik dari produk televisi LED.



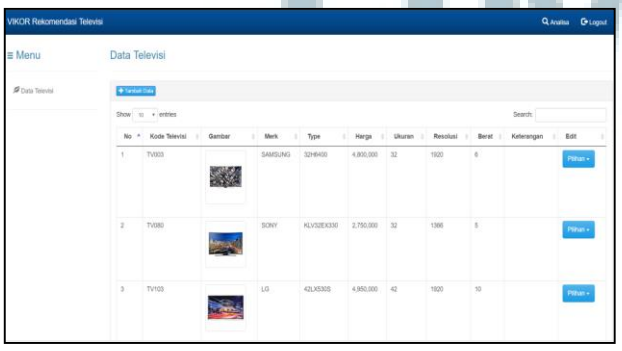
Gambar 4.2 *Interface* Halaman Hasil Rekomendasi

Detail Hasil Rekomendasi dapat dilihat pada halaman ini yang berisi spesifikasi dari data produk televisi yang menjadi hasil rekomendasi VIKOR.



Gambar 4.3 *Interface* Halaman Detail Hasil Rekomendasi

Pada halaman *admin* dapat melihat isi dari data produk televisi pada halaman *admin* data televisi, terdapat tabel yang menampilkan semua data produk televisi serta terdapat gambar produk televisi di dalam tabel tersebut.



Gambar 4.4 *Interface* Halaman *Admin* Data Televisi

B. Uji Coba Skenario

Pengujian dengan responden pertama dilakukan dengan melakukan skenario pemilihan referensi dan kriteria sebagai berikut.

- Harga : 3 s/d 5 Juta
- Nama Merk : SAMSUNG
- Ukuran Layar : 32 s/d 52 Inch

Untuk pemilihan kriteria yang dipilih oleh responden sebagai berikut.

- Harga : 5
- Ukuran Layar : 2
- Berat : 4
- Resolusi : 1

Langkah pertama yang dilakukan adalah dengan melakukan seleksi berdasarkan pemilihan referensi televisi oleh responden. Terdapat tiga televisi dengan data seperti pada Tabel 4.1.

Tabel 4.1 Hasil Data Pemilihan Referensi Skenario Pertama

ID_TV	Merk	Harga	Ukuran	Berat	Resolusi
TV218	SAMSUNG	3.950.000	40	11	1920
TV118	SAMSUNG	4.600.000	40	10	1920
TV123	SAMSUNG	4.599.000	40	10	1920

Selanjutnya, hasil dari pemilihan bobot kriteria oleh *user* dan menentukan nilai Max (Nilai Terbaik) dan nilai Min (Nilai Terburuk) dari keseluruhan *database* televisi seperti pada Tabel 4.2.

Tabel 4.2 Nilai Bobot Kriteria dan Nilai Max Min Skenario Pertama

Kriteria	Bobot	Nilai Terbaik	Nilai Terburuk
Harga	0,42	1.299.000	150.000.000
Ukuran	0,17	90	20
Berat	0,33	74	3
Resolusi	0,08	3.840	1.366

Data televisi yang ada di dalam *database* dinormalisasi dengan Rumus (2.1) dengan menampilkan 3 jenis data televisi sebagai contoh perhitungan manual seperti yang ditampilkan Tabel 4.3.

Tabel 4.3 Normalisasi Data Televisi Skenario Pertama

ID_TV	Harga	Ukuran	Berat	Resolusi
TV218	0,01782771	0,7142882	0,8873242	0,77607125
TV118	0,0221989	0,7142882	0,901409	0,77607125
TV123	0,0221921	0,7142882	0,901409	0,77607125

Hasil normalisasi dikalikan dengan nilai bobot kriteria yang dimasukkan oleh *user*. Perhitungan dilakukan pada semua data yang telah ternomalisasi, akan tetapi sebagai contoh perhitungan dilakukan dengan 3 jenis data televisi yang ditampilkan pada Tabel 4.4.

Tabel 4.4 Normalisasi x Bobot Kriteria Skenario Pertama

ID_TV	Harga	Ukuran	Berat	Resolusi
TV218	0,00748764	0,121429	0,292817	0,0620857
TV118	0,00932354	0,121429	0,297465	0,0620857
TV123	0,00932072	0,121429	0,297465	0,0620857

Untuk mencari *Utility Measure* (*S*) dan *Regret Measure* (*R*) dengan Rumus (2.2) yaitu menjumlahkan nilai yang telah ternomalisasi dan Rumus (2.3) dengan mencari nilai terbesar dari hasil nilai yang telah ternomalisasi.

S_TV218 = 0,00748764 + 0,121429 + 0,292817 + 0,0620857 = 0.483819
S_TV118 = 0,00932354 + 0,121429 + 0,297465 + 0,0620857 = 0.490303
S_TV123 = 0,00932072 + 0,121429 + 0,297465 + 0,0620857 = 0.4903
R_TV218 = 0.292817
R_TV118 = 0.297465
R_TV123 = 0.297465

Tabel 4.5 Nilai Min Max S dan R Skenario Pertama

	Nilai S	Nilai R
Max (S^*)	0.580282	0.42
Min (S')	0.246805	0.0976056

$Q\_TV218 = [\frac{0.483819-0.246805}{0,580282-0.246805}]x0,5+[\frac{0,292817-0.0976056}{0,42-0.0976056}]x(1-0,5) = [\frac{0,237014}{0,333477}] \times 0,5 + [\frac{0.1952114}{0,3223944}] \times 0,5 = 0,355367 + 0,302752 = 0, 658119$

$Q\_TV118=[\frac{0.490303-0.246805}{0,580282-0.246805}]x0,5+[\frac{0,297465-0.0976056}{0,42-0.0976056}]x(1-0,5) = [\frac{0,243498}{0,333477}] \times 0,5 + [\frac{0.1998594}{0,3223944}] \times 0,5 = 0,365089 + 0,309961 = 0,67505$

$Q\_TV123=[\frac{0.4903-0.1998594}{0,580282-0.246805}]x0,5+[\frac{0,297465-0.0976056}{0,42-0.0976056}]x(1-0,5) = [\frac{0,243495}{0,333477}] \times 0,5 + [\frac{0.211972}{0,3223944}] \times 0,5 = 0,365085 + 0,309961 = 0,675046$

Pengurutan perankingan ditentukan dari nilai yang paling rendah dengan solusi kompromi sebagai solusi ideal dilihat dari perankingan Q_j dengan nilai terendah. Karena nilai S_j merupakan solusi yang diukur dari titik terjauh solusi ideal, sedangkan nilai R_j merupakan solusi yang diukur dari titik terdekat solusi ideal. Pada Tabel 4.6 menunjukkan perankingan nilai Q yang diurutkan dari nilai yang terendah.

Setelah mendapat nilai S dan Nilai R maka dapat mencari nilai Q dengan menggunakan Rumus (2.4) dengan menentukan nilai Max (S^*) dan Min (S') terlebih dahulu dari hasil nilai S dan nilai R.

Tabel 4.6 Perankingan VIKOR Skenario Pertama

ID_TV	Merk	Nilai Q
TV218	SAMSUNG	0,658119
TV118	SAMSUNG	0.67505
TV123	SAMSUNG	0.675046

ID TV	Merk	Nilai Q
TV218	SAMSUNG	0.65812
TV118	SAMSUNG	0.67505
TV123	SAMSUNG	0.67505

Gambar 4.5 Hasil Perankingan VIKOR Skenario Pertama dari Sistem

Hasil dari Uji Skenario Pertama dengan membandingkan hasil perhitungan manual pada Tabel 4.6 dengan hasil perhitungan sistem dengan Gambar 4.5 menunjukkan bahwa hasil akhir dari perankingan dengan menggunakan metode VIKOR sesuai dengan urutan yang sama.

C. Uji Survei Kepuasan Pengguna Sistem

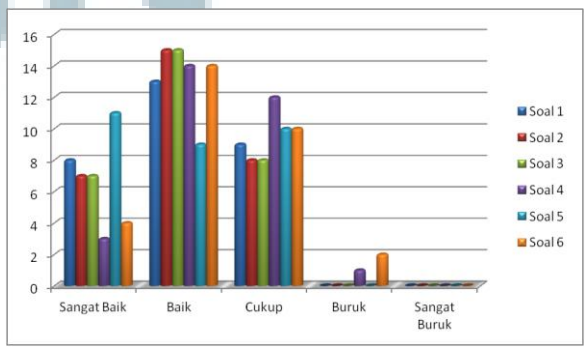
Uji Kepuasan Pengguna dilakukan dengan menggunakan Survei kepuasan kepada 30 orang responden [24]. Kuisioner yang digunakan berisi 6 pertanyaan yang harus dijawab dan pertanyaan itu berkaitan dengan sistem yang telah dibuat berdasarkan lima komponen mengenai kepuasan pemakai sistem informasi menurut Doll dan Torkzadeh [25]. Lalu penskalaan pernyataan sikap pada kuisioner menggunakan Skala Likert [26].

Berikut ini adalah tabel dan diagram dari hasil jawaban responden pada masing-masing soal yang telah diberikan.

Tabel 4.7 Hasil Survei Kepuasan Pengguna

	Soal 1	Soal 2	Soal 3	Soal 4	Soal 5	Soal 6
5	8	7	7	3	11	4
4	13	15	15	14	9	14
3	9	8	8	12	10	10
2	0	0	0	1	0	2
1	0	0	0	0	0	0

Pada gambar 4.6 merupakan grafik dari hasil survei kepuasan pengguna dari hasil jawaban responden pada masing-masing soal yang telah diberikan.



Gambar 4.**6** Grafik Tingkat Kepuasan Pengguna

Hasil dari kuisioner akan dihitung dengan menggunakan rumus Cronbach Alpha untuk menguji reabilitas dari hasil kepuasan pengguna sistem informasi [27]. Pada gambar 4.7 merupakan hasil tabel perhitungan dari hasil survei menggunakan rumus Cronbach Alpha sebesar 0,72 yang menunjukkan bahwa tanggapan responden terhadap sistem dalam kategori cukup baik [28].

Responden	Nomor Kuesioner						Jumlah	Jumlah Kuadrat
	1	2	3	4	5	6		
1	3	4	3	3	3	3	19	361
2	5	3	4	4	4	4	24	576
3	3	5	3	4	5	5	25	625
4	4	3	5	4	5	3	24	576
5	4	4	4	4	4	4	24	576
6	4	4	4	4	3	4	23	529
7	4	4	4	4	3	3	22	484
8	4	4	5	5	5	5	28	784
9	4	4	4	4	3	3	22	484
10	4	3	4	3	5	4	23	529
30	3	4	3	4	5	4	23	529
Jumlah	119	119	119	109	121	110	697	16447
Kuadrat Jumlah	489	487	487	441	509	442		
Varian Tiap Soal	0.565	0.498	0.498	0.498	0.698	0.622		
Total Varian Soal	3,379							
Varian Total Sistem	8,445							
Koefisien Alpha	0,72							

Gambar 4.7 Tabel Perhitungan Cronbach Alpha

V. SIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa rancang bangun sistem rekomendasi pembelian televisi LED dengan metode VIKOR dengan kriteria merk, harga, ukuran layar, resolusi layar, dan berat televisi telah berhasil dilakukan dengan menggunakan bahasa pemrograman PHP dan *database* MySQL. Dari hasil uji coba skenario yang telah dilakukan sudah dapat diverifikasi dengan membandingkan hasil perhitungan sistem dengan perhitungan manual dan survei yang dilakukan menggunakan Skala Likert dan Cronbach Alpha sebesar 0,72 yang dapat disimpulkan bahwa sistem mendapat tanggapan dari responden dalam kategori cukup baik.

REFERENSI

[1] Azhar Arsyad, M.A., (2007). "Media Pembelajaran". Jakarta: PT. Raja Grafindo Persada.

[2] MarkPlus Insight, 2014. Most Frequently Accessed Media in Last Six Month: At ten big Cities of Indonesia. Marketeers, Edisi Februari.

[3] Nielsen, 2014. "Komsumsi Media Lebih Tinggi Di Luar Jawa". Jakarta.

[4] KOMINFO, 2014. Survei Indikator TIK Rumah Tangga: Persentase Kepemilikan Radio, TV, dan Telepon di Rumah Tangga Tahun 2014".

[5] Eko, Stephanus, S.T., MMM, (2010). "LED TV Vs LCD TV Vs Plasma TV, Mana Yang Pantas Dibeli?". http://www.ubaya.ac.id/. Diakses Tanggal Akses 28 April 2016.

[6] Azis, Abdul, Hindayati Mustafidah. 2012. "Rekomendasi Pembelian Televisi Menggunakan Basis Data *Fuzzy* Tahani". Universitas Muhammadiyah, Purwokerto.

[7] Retno, Wahyu, Yessica Nataliana, Ramos Somya. 2012. Sistem Pendukung Keputusan untuk Merekomendasikan TV Layar Datar Menggunakan Metode *Weighted Product* (WP). Universitas Kristen Satya Wacana, Salatiga.

[8] Chatterjee, P., V. M. Athawale, dan S. Chakraborty. 2010. "Selection of industrial robots using compromise ranking and outranking methods". India.

[9] Paulus, Salvius, Adhistya Erna, Silmi Fauziati. 2015. Implementasi Metode VIKOR untuk Seleksi Penerima Beasiswa. Universitas Gadjah Mada, Yogyakarta.

[10] B. Vučijak, T. Kupusović, S. MidŽić-Kurtagić, dan a. Ćerić, "Applicability of multicriteria decision aid to sustainable hydropower," *Appl. Energy*, vol. 101, no. June 2009, pp. 261–267, 2013.

[11] Opricovic, S. dan G. H. Tzeng. "Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS." *Eur. J. Oper. Res.*, vol. 156, no. 2, pp. 445-455, 2004.

[12] Wibowo, Eko Wahyu, dkk. (2013). Penerapan Algoritma Squeezer untuk Memberikan Rekomendasi Pilihan Lagu Berdasarkan Daftar Lagu yang Dimainkan pada Pemutar Mp3 Android. (pdf). Surabaya: Institut Teknologi Sepuluh Nopember (ITS).

[13] Fadlil, Junaidillah dan Mahmudy, Wayan Firdaus. (2007). "Pembuatan Sistem Rekomendasi Menggunakan Decision Tree dan Clustering". (pdf). Malang: Universitas Brawijaya.

[14] Kusumadewi, Sri, Sri Hartati, Agus Harjoko, Retantyo Wardoyo. 2006. Fuzzy MultiAttribute Decision Making (Fuzzy MADM). Graha Ilmu, Yogyakarta.

[15] Zimmermann, H.-J., "Fuzzy Set Theory and Its Applications", Kluwer Academic Publishers, Second Edition, Boston, MA, 1991.

[16] Ardianto, Elvinaro, 2004. "Komunikasi Massa: Suatu Pengantar". Bandung: Simbiosa Rekatama Media.

[17] Opricovic, S. (1998). "Multicriteria optimization of civil engineering systems." Faculty of Civil Engineering, Belgrade **2**(1): 5-21.

[18] Opricovic, S. dan G. H. Tzeng. "Extended VIKOR method in comparison with outranking methods," *Eur. J. Oper. Res.*, vol. 178, pp. 514–529, 2007.

[19] Sayadi, M. K., Heydari, et al. (2009). "Extension of VIKOR method for decision making problem with interval number." Applied Mathematical Modelling 33(5): 2257-2262

[20] Khezrian, M., W. Wan Kadir, et al. (2011). "Service Selection base on VIKOR method." International Journal of Research and Reviews in Computer Science 2(5).

[21] Datta, S., S. S. Mahapatra, et al. (2010). "Comparative study on application of utility concept and VIKOR method for vendor selection."

[22] San Cristobal, J. R., M. V. Biezma, et al. (2009). "Selection of Materials Under Aggressive Environments: The VIKOR method."

[23] Kusdiantoro. (2012). Analisis Usability Website Akademik Di Indonesia Menggunakan Metode Promethee, Vikor, dan Electree. Universitas Negri Yogyakarta.

[24] Roscoe, J.T. (1975) Fundamental Research Statistics for the Behavioural Sciences, 2nd edition. New York: Holt Rinehart & Winston

[25] Doll, W.J., dan G. Torkzadeh. 1988. "The Measurement of End-User Computing Satisfaction". MIS Quarterly. 12 (June). pp. 259-274.

[26] Risnita. (2012). "Pengembangan Skala Model Likert". (pdf). Jambi: Institut Agama Islam Negeri Sulthan Thaha Saifuddin Jambi.

[27] Setyawan, Wahyu. (2013). "Rumus Uji Validasi dan Reliabilitas." (pdf). Jakarta. Universitas Negeri Jakarta.

[28] Gliem, Joseph A. dan Rosenmary R. Gliem. (2003). "Calculating, Interpreting, and Reporting Cronbach's Alpha Reliability Coefficient for Likert Type Scales" Paper yang dipresentasikan pada acara Midwest Research to Practice Conference in Adult Continuing, and Community Education yang diselenggarakan di The Ohio State University, Colombus pada tanggal 8-10 Oktober 2003.

ISSN 2085-4552

Deteksi Komentar *Spam* Bahasa Indonesia Pada Instagram Menggunakan *Naive Bayes*

Antonius Rachmat C[1], Yuan Lukito[2]
Prodi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana Yogyakarta
anton@ti.ukdw.ac.id[1], yuanlukito@ti.ukdw.ac.id[2]

Diterima 19 Mei 2017
Disetujui 16 Juni 2017

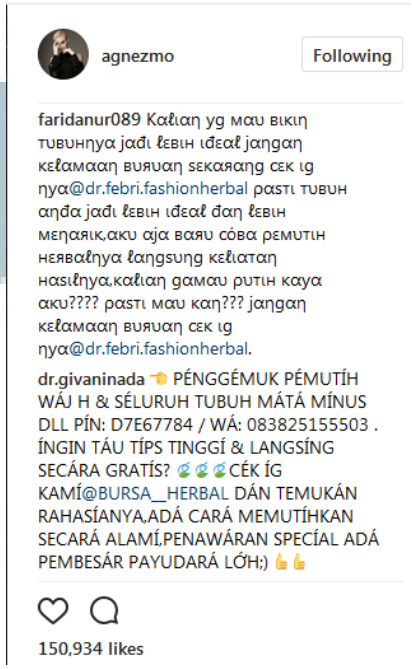
Abstract*— Instagram is the most famous pictures and videos media sharing based on the web & mobile application. Instagram users can have picture posts that can be commented by their followers. Indonesian public figures such as actors, actresses, musicians use Instagram to promote their activities to their followers. Unfortunately, there are a lot of spam comments in Instagram that need special attention and have to be removed. This research grabs Instagram comments and builds the dataset from Indonesian public figures who have more than one million followers. By using preprocessing (tokenization, stop words removal, and stemming), TF-IDF weighting, and supervised learning, Naive Bayes method is used to detect spam comments in Indonesian. Naive Bayes produces 74,31% accuracy rate on unbalanced datasets and 77,25% accuracy rate on balanced datasets. This result shows that Naïve Bayes can be used to build an automatic Indonesian spam comments detector on Instagram with high accuracy rate. The novelty of this research is that Naive Bayes can be used to detect spam comment on our Indonesian Instagram comments dataset.

***Index Terms*—Instagram, Naive Bayes, Indonesian spam comments, spam comments detection.**

I. PENDAHULUAN

Instagram (IG) merupakan salah satu aplikasi media sosial berbasis web dan *mobile* yang khusus digunakan untuk mengunggah gambar / foto. IG merupakan situs media sosial yang semakin banyak digunakan terutama oleh para artis/aktor Indonesia. Para pengguna IG akan mengunggah foto-foto kegiatan mereka dan kemudian untuk masing-masing foto / gambar tersebut akan dapat diberi *caption*, *tagging* akun IG lainnya, lokasi tempat kejadian foto tersebut diunggah, edit foto sesaat sebelum diunggah langsung dari aplikasi *smartphone*, dan *hashtag* tertentu agar foto tersebut makin banyak dilihat orang lain. Aplikasi IG dikembangkan untuk *smartphone*, baik platform iOS, Android, ataupun Windows Phone dan bersifat gratis. IG berbasis web dapat diakses di http://www.instagram.com. Melalui web pengguna hanya dapat melihat foto/gambar dari akun IG lain yang mereka ikuti, melihat dan mengedit profil pengguna aktif, dan melihat aktivitas pengguna IG lainnya, tapi tidak bisa mengunggah foto/gambar dari web, melainkan harus menggunakan aplikasi pada *smartphone*.

Salah satu hal yang menyebabkan IG banyak digunakan adalah kemudahannya untuk mengunggah foto langsung dari *smartphone*, mengingat pengguna media sosial kebanyakan adalah orang muda dan sangat menyukai *selfie*. Namun disamping kelebihan tersebut tentu terdapat kekurangan yang cukup mengganggu yaitu banyaknya komentar yang dapat dikategorikan sebagai komentar spam terhadap suatu *post* foto yang diunggah pada IG. Komentar spam akan semakin banyak terhadap IG artis/orang terkenal karena *follower*-nya juga semakin banyak. Bahkan di film Cek Toko Sebelah [1], yang baru rilis di tanggal 28 Desember 2016 saja diperlihakan bahwa komentar-komentar spam begitu mengganggu dan muncul cukup banyak di IG sampai digunakan sebagai salah satu bahan lelucon pada film itu. Contoh komentar spam pada salah satu foto milik @agnezmo dapat dilihat di Gambar 1.



Gambar 1. Komentar Spam Instagram

Beberapa solusi menghadapi komentar spam sudah ada, namun semuanya dilakukan secara manual. Pada [2] dan [3], pengguna IG dapat menghapus secara manual komentar spam tersebut namun jelas-jelas membutuhkan waktu yang besar dan harus diperiksa satu persatu. Selain dihapus secara manual IG juga menyediakan fitur untuk melaporkan semua komentar sebagai spam secara manual juga, artinya harus dilakukan satu persatu. Hal berikutnya untuk meminimalisasi komentar spam adalah dengan mengubah akun IG menjadi privat. Hal ini tentu sulit dilakukan bagi akun artis / aktor / publik figur, karena jika akun IG dibuat menjadi privat tidak bisa langsung di *follow* oleh akun lain. Hal terakhir yang dapat dilakukan adalah menggunakan pengaturan mengaktifkan fitur IG untuk menghapus komentar yang mengandung kata-kata tertentu yang dimasukkan sendiri oleh pengguna yang dianggap spam. Semua solusi tersebut hanya bisa digunakan dalam bahasa Inggris dan tidak dapat diterapkan dalam bahasa Indonesia.

Berdasarkan latar belakang tersebut pada penelitian ini akan dibangun suatu sistem yang dapat mengklasifikasikan komentar spam berbahasa Indonesia dengan mengambil data *training* komentar-komentar spam pada IG beberapa artis terkenal Indonesia. Terdapat beberapa metode untuk klasifikasi seperti Naive Bayes, K-Nearest Neighbour, Decision Tree, atau Support Vector Machine. Metode klasifikasi yang digunakan dalam penelitian ini adalah Naïve Bayes. Metode Naïve Bayes menggunakan konsep probabilitas setiap kelas dalam pembelajaran klasifikasinya, sehingga jika jarak perbedaan antar kelas tidak besar. Metode ini dipilih karena mudah diimplementasikan dan tidak terlalu membutuhkan sumber daya komputer yang besar.

Penelitian ini akan mengumpulkan data berupa status foto dan komentar dari 10 akun artis / aktor Indonesia yang memiliki *follower* lebih dari 1 juta. Untuk setiap akun artis / aktor akan diambil 50 status terbaru dan semua komentarnya. *Dataset* yang terbentuk akan dilakukan pelabelan secara manual menggunakan tenaga ahli untuk dapat digunakan sebagai data latih sistem *supervised learning* deteksi komentar spam menggunakan Naive Bayes.

II. LANDASAN TEORI

A. *Tinjauan Pustaka*

Spam adalah "*irrelevant or unsolicited messages sent over the Internet, typically to a large number of users, for the purposes of advertising, phishing, spreading malware, etc.*" [4]. Dalam terjemahan bebasnya spam berarti suatu tulisan / pesan yang tidak sesuai / tidak berhubungan dengan topik tertentu sehingga menyebabkan ketidaknyamanan atau bahkan ketidaktepatan informasi yang diperoleh pengguna. Spam dapat berwujud banyak hal, misalnya email spam, iklan spam, *click* spam, link spam, berita spam, dan tulisan (komentar) spam [5]. Istilah web spam (spam *indexing*) pertama kali muncul di tahun 1996 menurut tulisan Convey pada [6] pada harian The Boston Harald. Spam pada komentar merupakan bentuk link spam yang muncul pada web seperti *wikis*, *blogs*, dan *guestbooks*. Beberapa diantaranya sering ditemukan komentar, *trackback*, dan *pingback* spam pada tulisan (*blog*) yang diposting seseorang [7].

Menurut Hines pada [7] beberapa cara manual yang bisa digunakan untuk mendeteksi komentar spam adalah melihat konten teks komentar secara langsung apakah spesifik terhadap tulisan atau tidak, melihat apakah ada *link* pada komentar yang harus diklik atau tidak, pengguna yang berkomentar apakah menggunakan nama asli atau tidak, apakah penulis komentar menggunakan email asli atau tidak, atau menggunakan beberapa email yang berbeda-beda atau tidak. Sedangkan menurut Mishne, Carmel, & Lempel pada tahun 2005 pada [8] beberapa teknik untuk mengurangi komentar spam adalah menggunakan registrasi sebelum posting komentar, menggunakan *captcha* sebelum posting komentar, tidak mengijinkan komentar berbentuk HTML, menggunakan *blacklist* IP a*ddress*, dan memberikan batasan komentar agar tidak berkali-kali memberikan komentar.

Selain cara-cara manual seperti yang telah dijelaskan sebelumnya, deteksi spam pada teks dapat dilakukan secara otomatis menggunakan beberapa metode *content based filtering, link based filtering* dan metode yang tidak terlalu umum seperti menggunakan kebiasaan (*behaviour*) pengguna misalnya klik, *gesture*, *session*, dan lain-lain. Menurut Spirin & Han pada [9] deteksi spam kebanyakan menggunakan metode berbasis isi teks / tulisan yang ditulis. Hal ini dapat dilakukan dengan menggunakn beberapa metode seperti algoritma klasifikasi pada teks seperti algoritma Naive Bayes [10] dan Support Vector Machine [11].

Naive Bayes juga sudah digunakan sebagai metode untuk mengklasifikasikan sentimen pada dataset teks politik (SentiPol *dataset*) yang ternyata mencapai tingkat akurasi sebesar 82,3 %. Dataset SentiPol dikumpulkan dari Facebook Page dan dilabeli secara *crowdsourced labelling* menggunakan metode *Weighted Majority Voting*. Naive Bayes termasuk algoritma yang menghasilkan nilai akurasi yang cukup tinggi dalam klasifikasi data teks [12].

Penelitian yang dilakukan oleh penulis merupakan penelitian yang akan mendeteksi spam dari teks komentar pada web media sosial Instagram berbahasa Indonesia. Penelitian ini dimulai dari pengambilan data pelatihan dari Instagram menggunakan Instagram API (*Grabber*) yang akan dijelaskan pada bagian berikutnya. Setelah data diperoleh maka dilakukan pelabelan data secara manual dan kemudian dilakukan implementasi deteksi spam menggunakan algoritma

yang biasa digunakan untuk klasifikasi teks yaitu Naive Bayes. Keaslian penelitian terdapat pada studi kasus data yang berasal dari akun Instagram dan komentar-komentar bebahasa Indonesia.

B. Instagram dan Instagram API

Instagram merupakan web media sosial yang muncul sejak 2010 (kemudian diakuisisi oleh Facebook tahun 2012) dan dikhususkan untuk menggunggah koleksi foto. Menurut Pew Research Center pengguna Instagram mencapai 26% dari pengguna Internet dewasa, dimana penggunanya berusia 18-29 tahun [13]. Foto-foto yang diunggah dapat dilihat oleh semua followers dalam *timeline*-nya. Instagram memiliki kekhasan yaitu: 1). Proses unggah hanya bisa dilakukan melalui aplikasi berbasis *mobile* dan aplikasinya memiliki fitur manipulasi foto seperti efek-efek foto tertentu. 2). Pengguna IG dapat mengikuti akun pengguna lain (*follow*) atapun dapat diikuti (di *follow*) oleh akun lain. Pengguna IG dapat membuat akun profil IG nya menjadi publik dan privat. 3). Semua foto yang diunggah dapat diberi *caption* dan *hashtag*, termasuk *tagging* terhadap pengguna akun lain. 4). Semua posting foto dapat diberi komentar, di-*like*, bahkan dilaporkan ke IG sebagai *posting* yang tidak baik. 5). Selain foto IG dapat digunakan juga untuk mengunggah video berdurasi pendek (kurang lebih 1 menit) [14].

Instagram API (*Application Programming Interface*) merupakan *tool* pemrograman berbasis layanan yang dapat digunakan untuk mengakses dan memanipulasi pencarian tag, pencarian foto, *trending* foto, print foto, *custom item*, *feeds*, dan komentar-komentar yang terdapat pada Instagram menggunakan bahasa pemrograman tertentu dari sisi *programmer* (*developer*) [15]. Instagram API dapat diakses di https://www.instagram.com/developer/ menggunakan akun Instagram yang telah dibuat sebelumnya. Untuk dapat menggunakan Instagram API dibutuhkan : 1). register API client, 2). *Generate access token*, 3). Lakukan pembuatan aplikasi untuk mengakses hal yang terdapat pada Instagram [16].

C. Text Transformation dan TF-IDF Weighting

Di dalam pengolahan data *mining* berupa teks, proses yang paling penting adalah pengubahan dari data teks tidak terstruktur menjadi data nomimal yang terstruktur. Perubahan yang disebut *text transformation* tersebut, terdiri dari: 1). Tokenisasi, yaitu pengubahan dari string besar ke dalam satu buah kalimat/kata, 2). *Pre-processing*, yaitu proses data *cleaning*, pengubahan ke huruf kecil/besar semua, penghapusan kata-kata yang masuk dalam *stop words*, pengubahan ke bentuk kata dasar (*stemming*), dan konversi *emoticon* & kata-kata khusus, serta 4). Token weighting, yaitu pemberian bobot (w) terhadap token-token tersebut misalnya menggunakan TF-IDF [17] .

TF-IDF merupakan *Term Frequency-Inverse Document Frequency*, yang dapat digunakan untuk memberikan tingkat kepentingan token-token penentu dalam klasifikasi spam seperti pada Persamaan (1) [17]:

$$tfidf(t,d,D) = tf(t,d) * idf(t,D) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Di mana:

- tfidf(t,d,D) adalah bobot kepentingan suatu token yang muncul dalam suatu komentar di seluruh komentar-komentar yang ada, dimana semakin sering muncul suatu token dalam suatu dokumen dan semakin banyak komentar yang memilikinya akan semakin tidak penting karena token tersebut bersifat sangat umum.
- tf(t,d) adalah jumlah token yang terdapat pada satu komentar.
- idf(t,D) adalah *Inverse Document Frequency*, yaitu log dari jumlah seluruh komentar dibanding dengan jumlah seluruh komentar dimana token tersebut muncul.
- Rumus idf(t,D) dapat dilihat pada Persamaan (2):

$$idf(t,D) = \log (N / DF) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Di mana:
N adalah jumlah keseluruhan komentar dan DF adalah jumlah kemunculan token pada semua komentar-komentar.

D. Algoritma Naive Bayes

Algoritma Naive Bayes merupakan algoritma klasifikasi berdasarkan probabilitas dalam statistik yang dikemukakan oleh Thomas Bayes yang memprediksi peluang di masa depan berdasarkan peluang di masa sebelumnya. Metode ini kemudian digabungkan dengan situasi "*naive*" dimana kondisi antar atribut dalam sistem saling bebas dan tidak berhubungan satu sama lain. Dalam data *training*, setiap data traing memiliki atribut-atribut dan 1 label *class*, maka probabilitas suatu data masuk ke dalam suatu label *class* dapat didefinisikan sebagai berikut [18]:

1. Diketahui D adalah data *training* dan label *class*-nya. Setiap data direpresentasikan dalam bantuk n dimensi vektor atribut X = (x1, x2,... , xn)
2. Misalkan terdapat m jumlah class, C1, C2,... ,Cm. Metode Naive Bayes akan memprediksi apakah X masuk dalam *class* yang memiliki nilai *posterior* probabilitas tertinggi. Naive Bayes akan memprediksi X akan masuk ke dalam kelas Ci jika dan hanya jika: $P(Ci|X) > P(Cj|X)$ for $1 \leq j \leq m, j <> i$. Kemudian akan dimaksimumkan P(Ci|X). *Class* terbanyak dari Ci disebut

dengan *maximum* posteriori *hypothesis* yang dihitung menggunakan Persamaan (3):

$$P(C_i|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|C_i)P(C_i)}{P(\boldsymbol{X})}.$$

.....(3)

3. Karena P(X) bersifat tetap untuk semua *class*, maka hanya P(X | Ci) P(Ci) yang harus dimaksimumkan. Jika *prior* probabilitas dari *class* tidak diketahui, maka secara umum diasumsikan semua *class* sama P(C1) = P(C2) = ... = P(Cm). Perlu diingat bahwa *prior* probabilitas dari *class* diestimasi dengan P(Ci) = |Ci,D| / |D|, dimana |Ci,D| adalah jumlah *training* data yang termasuk dalam *class* Ci di *dataset* D.
4. Untuk *dataset* yang memiliki banyak atribut maka kompleksitas komputasi akan sangat tinggi, sehingga perlu direduksi dengan cara mengasumsikan semua kondisi *class* bersifat saling bebas (*independence*). Hal ini menganggap bahwa nilai antar atribut saling tidak mempengaruhi satu sama lain, sehingga dapat didefinisikan :
Dimana probabilitas P(x1| Ci), P(x2| Ci),..., P(xn| Ci) dapat diperoleh dengan mudah dari data *training*. Dimana xk adalah nilai yang ada di atribut Ak untuk data X. Untuk setiap atribut, harus dilihat apakah nilai atribut bersifat kategorikal atau nilai kontinu.
   - Jika Ak bersifat kategorikal, maka P(xk|Ci) adalah jumlah data xk yang memiliki *class* Ci di data *training* D dibagi dengan |Ci,D|, jumlah seluruh data *class* Ci di data *training* D.
   - Jika Ak bersifat kontinu, seperti misalnya data umur, data angka lainnya, yang tidak bisa dikategorikan, maka data tersebut harus dibuat dalam rentang nilai, menggunakan Gaussian Distribution dengan Persamaan (4) dan (5)

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

.....(4)

Sehingga diperoleh P(xk| Ci) = g(xk, μCi, σCi) .....(5)

.

5. Untuk memprediksi label class untuk data X, P(X|Ci) P(Ci) maka prediksi dilakukan untuk setiap class Ci. Metode Naive Bayes akan memprediksi class untuk X adalah Ci jika dan hanya jika (Persamaan (6)):

P(X|Ci)P(Ci) > P(X|Cj)P(Cj) for $1 \le j \le m$, j <>i. ..... (6).

Dalam arti prediksi terhadap *class* dengan probabilitas terbesar dihitung dengan Persamaan (7).

$$P(\boldsymbol{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i). \quad (7)$$

E. Confusion Matrix

Untuk melakukan pengujian terhadap sistem, dilakukan evaluasi akurasi sistem dalam mengklasifikasikan sentimen pada dataset dengan menggunakan *confusion matrix* [19] seperti pada Tabel 1 berikut.

Tabel 1. *Confusion Matrix*

		***Class* Hasil Prediksi**	
		Negatif	Positif
***Class* sebenarnya**	Negatif	*True* Negatif (TN)	*False* Negatif (FN)
	Positif	*False* Positif (FP)	*True* Positif (TP)

Di mana:

- *True* negatif = jumlah data negatif yang benar dikategorikan sebagai *class* negatif
- *False* negatif = jumlah data negatif yang dikategorikan sebagai *class* positif
- *False* positif = jumlah data positif yang dikategorikan sebagai *class* negatif
- *True* positif = jumlah data positif yang benar dikategorikan sebagai *class* positif

Dari *confusion matrix* pada Tabel 1 dapat dilakukan perhitungan lebih lanjut untuk mendapatkan tingkat akurasi (*accuracy*), *recall*, *precision* dan *f-measure* dengan Persamaan (8) – Persamaan (13).

Accuracy = (TN + TP) / (TN + FP + FN + TP) .....(8)

Recall / True Positive Rate = TP / (FP + TP)..... (9)

False Positive Rate = FN / (TN + FN) .....(10)

Specificity / True Negative = FP / (FP + TP) .....(11)

Precision = TP / (FN + TP) .....(12)

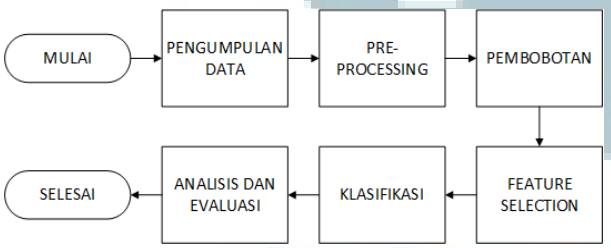
F-Measure = 2 * TP / (2 * TP + FP + FN ..... (13)

III. METODE PENELITIAN

Metode penelitian yang dilakukan dalam penelitian ini terdiri atas lima tahap sebagai berikut:

1. Tahap studi literatur digunakan untuk mempelajari algoritma Naive Bayes yang digunakan, termasuk mempelajari kode program pembuatan Instragram data *collector* menggunakan Instagram API (*tool* Instagram *Grabber*) [20].
2. Tahap pengumpulan data digunakan untuk mengumpulkan data yang berhubungan dengan penelitian yang akan dilakukan. Tahap ini akan menggunakan Instagram API untuk pengambilan data dari Instagram yang berasal dari 10 artis Indonesia yang memiliki *follower* lebih dari 1 juta. Pada tahap ini juga termasuk pre-*processing* data.
3. Tahap implementasi, yaitu tahap untuk melakukan pelabelan secara manual tentang spam / bukan spam dan kemudian dilanjutkan dengan melakukan implementasi kedua metode menggunakan aplikasi dan RapidMiner 7.5. Tahap ini termasuk pembobotan, *features selection*, dan klasifikasi.
4. Tahap analisis dan evaluasi bertujuan untuk menganalisis penggunaan kedua metode yang kemudian dibandingkan keakuratan keduanya menggunakan *confusion matrix*.

Seluruh tahapan penelitian dapat dilihat pada diagram alir Gambar 2 berikut:



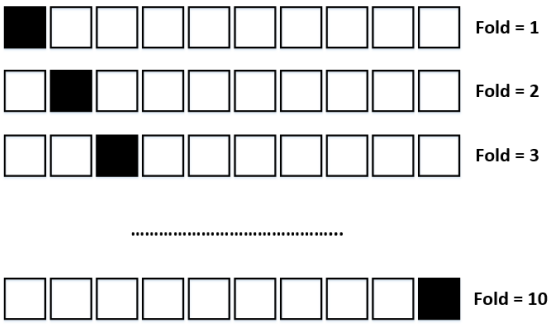
Gambar 2. Tahapan Penelitian

A. *Pengumpulan Data*

Pengujian terhadap implementasi kedua metode akan dilakukan terhadap seluruh data uji yang telah diambil dari 10 akun Instagram artis Indonesia yang memiliki *follower* lebih dari 1 juta akun. Data uji akan disimpan dalam basis data yang berisi data *username* IG artis, *posting* IG, tanggal *posting*, dan semua komentar dari 50 status terbaru, siapa yang berkomentar, dan tanggal komentar.

B. *Metode Pengujian*

Pengujian akan dilakukan menggunakan metode *k-fold validation* dengan nilai k = 10. Metode pengujian ini dilakukan dengan cara membagi dataset menjadi 10 bagian yang sama. Dalam 10 iterasi, setiap bagian dari dataset digunakan sebagai data uji, sedangkan bagian lainnya digunakan sebagai data pelatihan. Ilustrasi *k-fold validation* dapat dilihat pada Gambar 3.



Gambar 3. Ilustrasi metode *k-fold validation* dengan nilai k = 10.

Ringkasan dari pengujian yang akan dilakukan dapat dilihat pada Tabel 2.

Tabel 2. Ringkasan Rancangan Pengujian

Parameter	Nilai
Metode validasi	*k-Fold Validation*
Metrik pengujian	*Confusion Matrix*
Jumlah data	17000 data
Tool	RapidMiner 7.5
Kriteria output yang dihasilkan	*Accuracy* dan *Classification*
Sampling tape	*Shuffled sampling*

IV. HASIL DAN PEMBAHASAN

A. *Implementasi Pengambilan Data*

Data diperoleh diambil langsung dari web Instagram menggunakan teknik *scrapping* menggunakan *tool* PHP Instagram Grabber [19] yang tersedia di GitHub dan dapat diunduh secara gratis. Tool ini dibuat oleh Thomas Bolander dan Kristian Vassard dari Denmark sejak 9 Juni 2016 (versi 1.0.4) di GitHub. *Tool* ini memungkinkan penulis untuk mengambil data-data komentar Instagram dari seorang pemilik akun Instagram yang bersifat publik seperti artis Indonesia tanpa menggunakan *username* dan *password developer*. Teknik yang digunakan oleh tool ini adalah menggunakan teknik *scrapping* halaman-halaman HTML dari Instagram dan kemudian di-*parsing* otomatis berbasis PHP.

Penggunaan *tool* PHP Instagram Grabber adalah sebagai berikut:

1. Pengaturan PHP Instagram *Grabber* pada *server* 'localhost'. Tambahkan sesuai kebutuhan seperti pengaturan: set_time_limit(0) untuk memproses data yang jumlahnya besar dan lama. Pengaturan proxy juga jika perlu menggunakan pada variabel `CURLOPT_PROXY=> <IP address proxy>` dan `CURLOPT_PROXYPORT => <no port>`.
2. Menggunakan *method* `Bolandish\Instagram::getMediaByUserID("<id`

`instagram>",<jumlah post yang hendak diambil>).`

3. Menggunakan perulangan yang dilakukan untuk mengambil semua komentar untuk masing-masing post yang diambil di langkah 2 sebelumnya dengan perintah `Bolandish\Instagram::getCommentsByMediaShortcode(<kode post yg didapat dari langkah 2>, <jumlah komentar yg hendak diambil>).`
4. Data-data komentar tersebut dimasukkan / diekspor ke format lain seperti CSV atau TXT.

Berikut adalah rencana pengambilan data untuk pembentukan *dataset* komentar spam dari Instagram artis Indonesia:

1. Data *post* akan diambil dari 10 akun aktor / artis Indonesia yang memiliki jumlah *follower* terbanyak berdasarkan sumber [21] sebagai berikut:
   - Ayu Tingting @ayutingting92: 522969993 - 18.4 juta *followers*
   - Syahrini @princessyahrini: 24239929 - 16.8 juta *followers*
   - Raffi Ahmad Nagita Slavina @raffinagita1717: 1918078581 - 15.7 juta *followers*
   - Laudya Chinthia Bella @laudyacynthiabella: 2993265 - 14.8juta *followers*
   - Prilli Ratucosina @prillylatuconsina96: 225064794 - 14.8juta *followers*
   - Julia Perezz @juliaperrezz: 30585021 - 12.3 juta *followers*
   - Chelsea Olivia @chelseaoliviaa: 5735890 - 12.6 juta *followers*
   - Raisa @raisa6690: 8115577 - 12.1 juta *followers*
   - Luna Maya @lunamaya: 1948416 - 11.6 juta *followers*
   - Agnes Monica @agnezmo: 4934196 - 11.3 juta *followers*.
2. Dari setiap akun artis / aktor akan diambil 50 *post* terbaru.
3. Dari setiap *post* yang terambil dari langkah 2 akan diambil 50 komentar terbaru, sehingga terdapat data sekitar 10 artis x 50 *post* x 50 komentar = 25000 komentar.
4. Dari 25000 komentar tersebut akan dibuat dalam format CSV per akun artis dan kemudian dilakukan *preprocessing* data pada bagian berikutnya.

Setelah data terambil menggunakan *tool* Grabber, langkah berikutnya adalah melakukan pelabelan data. Pelabelan data dilakukan secara manual menggunakan tenaga pelabel ahli. Proses pelabelan memakan waktu sekitar 1 bulan untuk semua data. Pelabelan dilakukan hanya dengan 2 kelas, yaitu "spam" dan "notspam". Pelabelan relatif mudah karena sangat jelas sekali perbedaan antara komentar spam dan bukan spam.

Tabel 3 merupakan profil hasil klasifikasi pelabelan manual menggunakan tenaga pakar untuk dataset Instagram yang telah dikumpulkan. Dari rencana jumlah data 25.000, pada saat pengambilan tidak semua post memiliki komentar 50 komentar sehingga realisasi pengambilan data terdapat pada Tabel 3.

Tabel 3. Profil Hasil Pelabelan Instagram 10 Artis Indonesia

No.	Artis	Nama Kelas dan Jumlah
1.	Ayu Ting-Ting	Spam (1262), Bukan Span (584)
2.	Julia Perez	Spam (1362), Bukan Spam (739)
3.	Nagita Slavina	Spam (1435), Bukan Spam (610)
4.	Syahrini	Spam (922), Bukan Spam (448)
5.	Laudya Cinthia Bella	Spam (902), Bukan Spam (688)
6.	Prili Ratuconsina	Spam (437), Bukan Spam (1091)
7.	Chelsea Olivia	Spam (1625), Bukan Spam (293)
8.	Luna Maya	Spam (965), Bukan Spam (275)
9.	Raisa	Spam (666), Bukan Spam (621)
10.	Agnes Monica	Spam (1143), Bukan Spam (940)
		JUMLAH SPAM 10.719
		JUMLAH BUKAN SPAM 6.288
		TOTAL KESELURUHAN 17.007

B. Implementasi Preprocessing Data

Setelah data terkumpul dalam format CSV langkah berikutnya adalah tahap *pre-processing (cleaning data)* agar nantinya dapat dimasukkan dalam basis data MySQL dengan mudah. Proses *cleaning* dilakukan secara langsung pada file CSV nya dengan cara:

1. Menggunakan format karakter *unicode* UTF-8 karena semua karakter hasil Instagram menggunakan simbol-simbol dan karakter *unicode*.
2. Konversi karakter khusus seperti *emoticon* menjadi istilah-istilah seperti yang ditampilkan pada Tabel 4 berikut:

Tabel 4. Konversi *Emoticon* dari Instagram

No.	Karakter Emoticon	Konversi
1	:)	Senyum
2	(y)	Suka
3	~_~	Netral
4	(Y)	Suka
5	:*	Cium
6	=))	Senyum
7	:D	Senyum
8	*Like* atau *like*	Suka
9	*YES*, *Yes*, atau *yes*	Suka

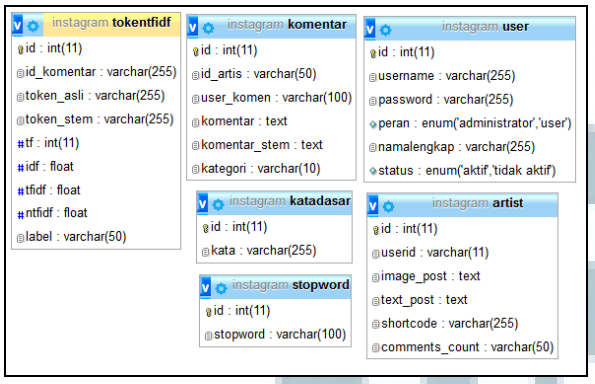
3. *Cleansing*: menghapus karakter-karakter seperti: ~, `, !, $, %, ^, &, *, (,), _, -, +, =, :, ", ', <, >, koma, titik, ?, /, \, dan |.
4. Membuang semua spasi yang berjumlah lebih dari satu dan menggabungkannya menjadi satu spasi saja. Membuang semua spasi di

awal dan akhir kalimat (*trim*), dan menghapus semua baris yang kosong.

5. Membuang semua angka, string dengan format URL, dan email.

C. *Implementasi Basis Data*

Setelah semua data CSV sudah melalui tahapan *cleaning* data, maka langkah selanjutnya dilakukan pembuatan basis data yang terdiri dari 3 tabel: stopwords, komentar, dan artist, sesuai skema pada Gambar 4 berikut.



Gambar 4. Skema Basis Data

Setelah dibuat semua tabel pada basis data MySQL langkah berikutnya adalah mengimpor semua CSV ke dalam tabel artis dan komentar menggunakan *tool import* pada MySQL. Contoh data komentar setelah diimpor dapat dilihat pada Gambar 5 berikut.

widya prasasti2212	butuh followers instagram atau likes instagram sedikit yuk buruan order hanya unicornappsstores
ficwidhi	followers instagram kamu sedikit atau likes instagram kamu sedikit yuk order hanya unicornappsstores
opet awsm	butuh followers instagram atau likes instagram kamu sedikit yuk cek instagram unicornappsstores melayani dengan ramah

Gambar 5. Contoh Data Komentar

Setelah data berada di dalam MysQL, tahap berkutnya adalah proses *cleaning* yang dilakukan dengan langkah sebagai berikut:

1. Mengatur jenis karakter pada MySQL menjadi UTF-8.
2. Menghapus semua karakter ? setelah proses impor dari CSV karena perbedaan karakter, dengan SQL: `update komentar set komentar=replace(komentar,'?',' ')`
3. Menghapus semua komentar yang jumlah hurufnya <= 1 dengan SQL: `delete from komentar where length(komentar)<=1`
4. Mengubah semua spasi yang lebih dari satu menjadi satu spasi dengan memanggil *stored procedure* DELETE_DOUBLE_SPACES, dengan perintah SQL: `update komentar set komentar=DELETE_DOUBLE_SPACES(komentar)`
5. Menghapus semua kata pada komentar yang memiliki jumlah karakter <= 1, dengan memanggil *stored procedure / function* ALPHA, dengan perintah SQL: `update komentar set komentar=ALPHA(komentar)`
6. Menghapus spasi di awal dan akhir komentar dengan SQL: `update komentar set komentar=trim(komentar).`

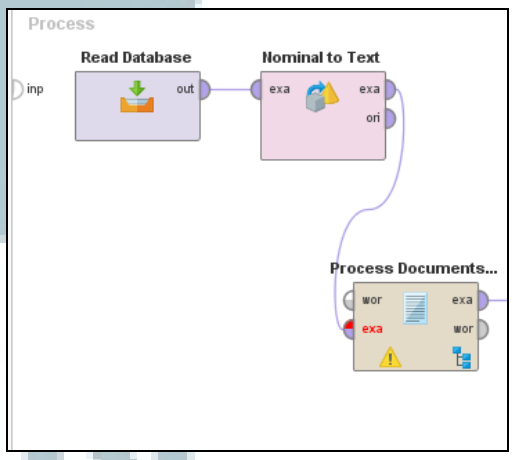
Setelah semua proses *cleaning* data pada MySQL dilakukan maka hasil data terakhir yang siap digunakan untuk proses *learning* klasifikasi berjumlah 16.641 data yang dapat dilihat pada Tabel 5 berikut.

Tabel 5. Hasil Akhir Proses Data *Cleaning*

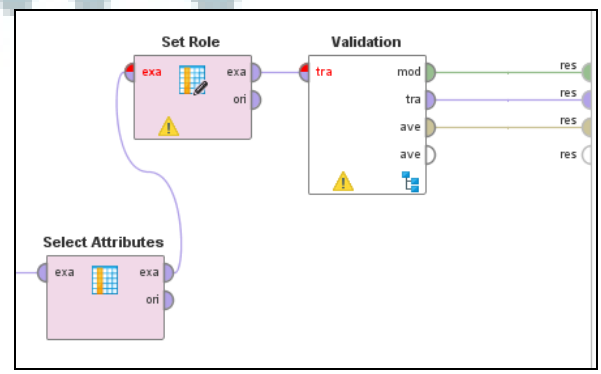
No.	Kelas	Jumlah
1.	Spam	10.399
2.	Not Spam	6.062
	TOTAL	**16.641**

D. *Implementasi Deteksi Spam*

Deteksi spam menggunakan *software* RapidMiner versi 7.5 dengan konfigurasi operator-operator seperti pada Gambar 6 dan 7 berikut.



Gambar 6. Konfigurasi RapidMiner (Bagian 1)

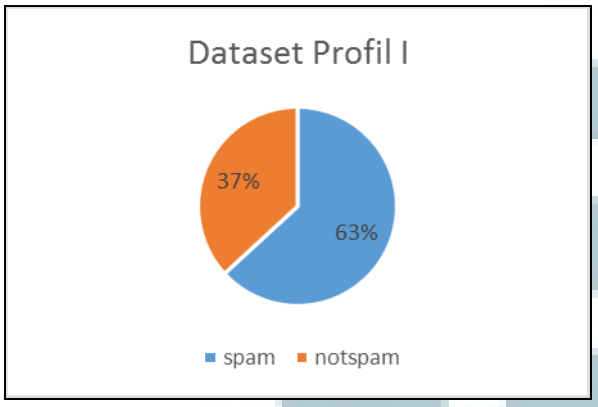


Gambar 7. Konfigurasi RapidMiner (Bagian 2)

E. Pengujian Skenario I (Unbalanced Dataset)

Skenario I adalah pengujian di mana data yang digunakan untuk *training* berjumlah 10.399 untuk data spam dan 6062 untuk data *not* spam. Dari data tersebut dilakukan pengujian menggunakan teknik *k-fold validation* dengan k=10, artinya data uji untuk masing-masing pengujian berjumlah 1646 (10%) dan hasilnya akan dirata-rata.

Dari hasil percobaan menggunakan skenario I (seperti pada Tabel 5 sebelumnya) yang ditampilkan dalam bentuk *pie chart* seperti pada Gambar 8, diperoleh hasil seperti pada Tabel 6.



Gambar 8. Profil Dataset Skenario I

Tabel 6. Hasil *Confusion Matrix* Skenario I menggunakan Naïve Bayes

	True Spam	*True Not Spam*	
Predicted Spam	6388 (TP)	217 (FP)	**96,71% (precision)**
Predicted Not Spam	4011 (FN)	5845 (TN)	**59,30% (fallout)**
	61,43% (recall)	**96,42% (specificity)**	

Waktu proses keseluruhan : 12.33 menit
Proses validasi: 10.24 menit

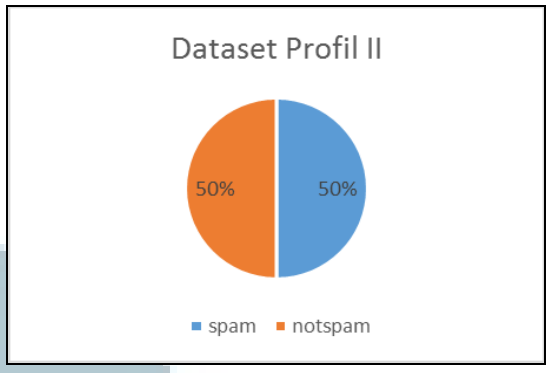
Dari hasil pada Tabel 6 tersebut diperoleh hasil:

- *Accuracy* = 74,31 %
- *Classification error* = 25,69 %
- *Sensitivity (Recall)* = 61,43 %
- *Specificity* = 96,4 %
- *Precision* = 96,71 %
- *F-Measure* = 75, 13 %

Hal ini menujukkan bahwa untuk dataset pelatihan yang tidak seimbang antara spam dan bukan spam akurasi klasifikasinya sudah cukup baik (di atas 70%).

F. Pengujian Skenario II (Balanced Dataset)

Skenario II adalah pengujian di mana data spam dan *not* spam dibuat menjadi seimbang, sehingga yang digunakan untuk *training* berjumlah 6062 untuk data spam dan 6062 untuk data *not spam*. Dari data tersebut dilakukan pengujian menggunakan teknik *k-fold validation* dengan k=10, artinya data uji untuk masing-masing pengujian berjumlah 606 (10%) dan hasilnya akan dirata-rata. Dari hasil percobaan menggunakan skenario II dan ditampilkan seperti dalam bentuk *pie chart* pada Gambar 9 diperoleh hasil pada Tabel 7 sebagai berikut:



Gambar 9. Profil Data Skenario II

Tabel 7. Hasil *Confusion Matrix* Skenario II menggunakan Naïve Bayes

	True Spam	*True Not Spam*	
Predicted Spam	3468 (TP)	164 (FP)	**95.48 % (precision)**
Predicted Not Spam	2594 (FN)	5898 (TN)	**69.45 % (fallout)**
	57.21 % (recall)	**97.29% (specificity)**	

Waktu proses keseluruhan: 6.31 menit
Proses validasi: 4.56 menit

Dari hasil pada Tabel 7 tersebut diperoleh hasil:

- *Accuracy* = 77,25 %
- *Classification error* = 22,75 %
- *Sensitivity (Recall)* = 57,21 %
- *Specificity* = 97, 29 %
- *Precision* = 95,48 %
- *F-Measure* = 71,5 %

Hal ini menujukkan bahwa untuk dataset pelatihan yang seimbang antara spam dan bukan spam akurasi klasifikasinya baik (di atas 75%).

Dilihat dari kedua pengujian menggunakan skenario I dan II diperoleh peningkatan akurasi sebesar 2,94% pada data profil yang seimbang. Dari penelitian ini algoritma Naïve Bayes jelas dapat digunakan untuk sistem detektor spam pada data komentar Instagram berbahasa Indonesia. Berdasarkan hal tersebut, beberapa penelitian yang masih dapat dikembangkan dari penelitian ini adalah membangun aplikasi / sistem dalam bentuk service dan diimplementasikan pada plugin browser baik desktop ataupun mobile untuk mendeteksi dan

menandai komentar spam pada Instagram saat pengguna membuka halaman Instagram.

V. SIMPULAN

Berdasarkan penelitian yang telah dilakukan maka diperoleh beberapa kesimpulan sebagai berikut:

1. Penelitian ini sudah berhasil mengimplementasikan algoritma Naïve Bayes pada studi kasus deteksi komentar spam menggunakan data Instagram berbahasa Indonesia. Tingkat akurasi yang didapatkan sudah cukup baik di atas 75%, yaitu 77,25%.
2. Penelitian ini telah menguji kemampuan algoritma klasifikasi Naive Bayes dan diperoleh akurasi sebesar 74,31 % untuk skenario I (*dataset* tidak seimbang) dan akurasi sebesar 77,25% untuk skenario II (*dataset* seimbang). Terjadi peningkatan keakuratan sebesar 2,94 % untuk *dataset* seimbang.
3. Tahapan *preprocessing* data Instagram bahasa Indonesia yang perlu dilakukan untuk pemrosesan data deteksi komentar spam dari Instagram adalah: *setting encoding* teks ke *encoding* Unicode (UTF-8), tokenisasi, *case folding*, *stop words removal*, *stemming*, dan konversi *emoticon*.

Sedangkan saran-saran yang masih perlu dikerjakan untuk penelitian selanjutnya adalah:

1. Tahap *stemming* masih perlu dilakukan dan diuji apakah memiliki pengaruh terhadap hasil akurasi atau tidak.
2. Melakukan tahapan perbandingan dengan algoritma lain yaitu menggunakan metode Support Vector Machine.
3. Dapat dikembangkan lebih lanjut ke pengembangan *plugin browser* yang secara otomatis mendeteksi komentar spam saat menampilkan halaman Instagram dan menandainya sehingga pengguna mengetahuinya.

UCAPAN TERIMA KASIH

Terima kasih penulis berikan kepada Lembaga Penelitian dan Pengabdian Kepada Masyarakat Universitas Kristen Duta Wacana (UKDW) Yogyakarta yang telah memberikan dana penelitian untuk tahun 2017.

DAFTAR PUSTAKA

[1] E. Prakasa, *Cek Toko Sebelah*. Jakarta: Starvision Plus, 2016.

[2] M. Eskelinen, "How to get rid of spam on Instagram", *Mervi Emilia*, 2015. [Online]. Available: http://merviemilia.com/blog/2015-02-how-to-get-rid-of-spam-on-instagram. [Accessed: 24- Jan- 2017].

[3] D. Tamir, "How To Protect Yourself From Instaspam - ReadWrite", *ReadWrite*, 2015. [Online]. Available: http://readwrite.com/2015/04/15/instagram-spam-instaspam-how-to-avoid/. [Accessed: 24- Jan- 2017].

[4] M. Webster, "Definition of SPAM", *Merriam-webster.com*, 2016. [Online]. Available: https://www.merriam-webster.com/dictionary/spam. [Accessed: 26- Jan- 2016].

[5] Geerthik S., "Survey on Internet Spam: Classification and Analysis", *Int.J.Computer Technology & Applications*, vol. 4, no. 3, pp. 384-391, 2013.

[6] E. Convey, "Porn sneaks way back on web", *The Boston Herald*, 1996.

[7] K. Hines, "How to Identify and Control Blog Comment Spam. Kissmetrics Blog", *Kissmetrics Blog*, 2012.

[8] G. Mishne, D. Carmel and R. Lempel, "Blocking Blog Spam with Language Model Disagreement", in *The First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Chiba, Japan, 2005.

[9] N. Spirin and J. Han, "Survey on web spam detection", *ACM SIGKDD Explorations Newsletter*, vol. 13, no. 2, p. 50, 2012.

[10] S. Raschka, "Naive Bayes and Text Classification {I} - Introduction and Theory", *CoRR*, vol. 14105329, no. 14105329, pp. 1-20, 2014.

[11] D. Sculey and G. Wachman, "Relaxed Online SVMs for Spam Filtering", in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, 2007, pp. 415--422.

[12] A. Rachmat and Y. Lukito, "SENTIPOL: Dataset Sentimen Komentar Pada Kampanye PEMILU Presiden Indonesia 2014 dari Facebook Page", in *Konferensi Nasional Teknologi Informasi dan Komunikasi 2017*, Univerasitas Kristen Duta Wacana, 2016, pp. 218-228.

[13] M. Duggan, N. Ellison, C. Lampe, A. Lenhart and M. Madden, "Social Media Update 2014", *Pew Research Center: Internet, Science & Tech*, 2014. [Online]. Available: http://www.pewinternet.org/2015/01/09/social-media-update-2014/. [Accessed: 30- Jan- 2017].

[14] Instagram, "Instagram Developer Documentation", *Instagram.com*. [Online]. Available: https://www.instagram.com/about/. [Accessed: 28- Jan- 2016].

[15] S. Roncero-Menendez, "8 Ways to Use Instagram's API", *Mashable*, 2017. [Online]. Available: http://mashable.com/2013/09/19/instagram-api-uses/#mSrayzI6Dmq3. [Accessed: 28- Feb- 2016].

[16] E. This, "Everything You Need to Know About Instagram API Integration | CSS-Tricks", *CSS-Tricks*, 2016. [Online]. Available: https://css-tricks.com/everything-need-know-instagram-api-integration/. [Accessed: 28- Jan- 2016].

[17] S. M. Weiss, N. Indurkhya, T. Zhang and F. Damerau , Text mining: Predictive Methods for Analyzing Unstructured Information, New York: Springer, 2005.

[18] J. Han and M. Kamber, *Data mining*. Haryana, India: Elsevier, 2012.

[19] U. DBD, "Confusion Matrix", *Www2.cs.uregina.ca*, 2017. [Online]. Available: http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html. [Accessed: 28- Jan- 2016].

[20] T. Bolander, "Bolandish/PHP-Instagram-Grabber", *GitHub*, 2016. [Online]. Available: https://github.com/Bolandish/PHP-Instagram-Grabber. [Accessed: 01- Feb- 2017].

[21] M. Deoranje, "musdeoranje.net: 10+ Akun Instagram Dengan Followers Terbanyak Di Indonesia", *musdeoranje.net*, 2015. [Online]. Available: http://www.musdeoranje.net/2016/08/akun-instagram-dengan-followers-terbanyak-di-indonesia.html. [Accessed: 8- Feb- 2017].

ISSN 2085-4552

Sistem Kriptografi Citra Digital Pada Jaringan Intranet Menggunakan Metode Kombinasi *Chaos Map* Dan Teknik Selektif

Arinten Dewi Hidayat[1], Irawan Afrianto[2]
Teknik Informatika, Universitas Komputer Indonesia (UNIKOM), Bandung, Indonesia
arinz_jfa@email.unikom.ac.id[1], irawan.afrianto@email.unikom.ac.id[2]

Diterima 29 Mei 2017
Disetujui 16 Juni 2017

Abstract*— Many organizations use design applications to draw the products they create. The drawings are digital files that have various formats of type and size. The images sometimes have to be protected because they are secret designs, such as the design of military vehicles, weapons and other designs. In order to secure digital data, cryptography is one of the solution, including if the data to be secured in the form of digital images. Algorithms that can be used to perform cryptography in digital image one of them is chaos map using Arnold cat map, logistics map and application of selective technique. Chaos was chosen for three reasons: sensitivity to initial conditions, random behavior, and no repetitive periods. While the application of selective technique means only encrypt some elements in the image but the effect of the whole image is encrypted. Cryptography in the image is also implemented because the organization using an intranet network to deliver its design drawings from one division to another. This allows for data tapping or exploitation of digital images while inside the intranet network. So it is necessary to develop a cryptographic system on the intranet network that has the ability to secure digital images that are in the network. The results obtained from black box testing, white box and network security testing show that the built system has been able to secure digital images when sent over the organization's intranet network.

***Index Terms*—Cryptography, Image, Chaos Map Algorithm, Selective Technique, Intranet.**

I. PENDAHULUAN

Saat ini perkembangan perusahaan-perusahaan yang bergerak dibidang desain semakin meningkat. Desain-desain yang dikembangkan pun telah menggunakan aplikasi-aplikasi pengolah gambar/citra guna mempercepat dan mempermudah proses desain gambar. Sehingga gambar-gambar yang dihasilkan berupa citra digital yang memiliki format umum seperti *Bitmap* (BMP), *Joint Photographic Group Experts* (JPEG), *Graphics Interchange Format* (GIF), dan *Portable Network Graphics* (PNG) [10].

Salah satu hal dipertimbangkan oleh perusahaan-perusahaan tersebut adalah bagaimana cara untuk melindungi dan mengamankan file-file gambar desain tersebut hingga tahap produksi. Hal ini dikarenakan banyaknya terjadi penyalahgunaan serta pencurian desain-desain produk terjadi. Apalagi jika desain-desain gambar yang dihasilkan merupakan desain-desain yang bersifat rahasia, seperti desain senjata, teknologi kendaraan militer dan sebagainya. Oleh karena itu dibutuhkan suatu sistem yang dapat melindungi data gambar tersebut sehingga hanya pengguna tertentu saja yang dapat menggunakan gambar tersebut.

Kriprografi pada citra digital menjadi salah satu solusi dalam pengamanan file-file gambar, sehingga lebih terlindungi dan memiliki akses yang terbatas. Kriptografi dapat dikembangkan dengan memanfaatkan algortima-algoritma yang ada guna mengamankan data-data pada citra digital.

Solusi terhadap keamanan citra digital dari penyadapan atau serangan adalah dengan mengekripsinya. Enkripsi citra merupakan teknik untuk melindungi citra dengan cara menyandikan citra (*plain-image*) sehingga tidak dapat dikenali lagi (*chiper-image*) [1]. *Chaos* dipilih karena karakteristik yaitu sensitivitas terhadap kondisi awal, berkelakuan acak, dan tidak memiliki periode berulang. *arnold cat map* digunakan untuk mengacak susunan *pixel-pixel*, sedangkan *logistic map* digunakan sebagai pembangkit *keystream*. Adapun pendekatan teknik selektif yaitu hanya mengenkripsi sebagian elemen di dalam citra namun efeknya keseluruhan citra terenkripsi [1].

Dikarenakan posisi bagian / divisi desain dan divisi produksi letaknya berjauhan, maka organisasi menfasilitasi pengiriman citra desain tersebut menggunakan jaringan intranet. Dikarena sifat jaringan intranet yang banyak pennguna , maka citra desain yang dikirimkan melalui jaringan intranet di organisasi tersebut dilengkapi dengan kritografi pada citra-citra digital yang dikirimkan, dengan harapan data-data citra digital akan menjadi lebih aman dan terlindungi dari kegiatan-kegiatan eksploitasi yang berda di jaringan komputer.

II. LANDASAN TEORI

A. *Kriptografi*

Kriptografi adalah ilmu yang berdasarkan pada teknik matematika yang erat kaitanya dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas. Jadi pengertian kriptografi modern adalah bukan hanya penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi [5].

Proses penyandian *plainteks* menjadi *cipherteks* disebut enkripsi dan proses mengembalikai *cipherteks* menjadi *plainteks* disebut dekripsi [6].

Gambar 1. Konsep kriptogtafi

B. *Citra Digital*

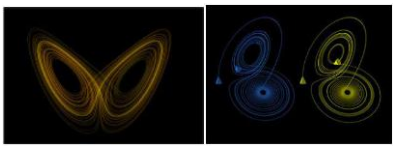
Citra digital adalah citra yang dapat diolah oleh komputer. Dalam konteks lebih luas, pengolahan citra digital mengacu pada pemrosesa setiap data dua dimensi. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real maupun komplek yang direpresentasikan dengan deret bit tertentu. Sebuah citra digital dapat diwakili oleh sebuah matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut *pixel*, yaitu elemen terkecil dari sebuah citra [10]. Salah satu bentuk citra digital adalah citra biner, yaitu citra yang hanya mempunyai dua nilai derajat keabuan: hitam dan putih. *Pixel – pixel* objek bernilai 1 dan *pixel-pixel* latar belakang bernilai 0. Pada waktu menampilkan gambar, 0 adalah putih dan 1 adalah hitam. Jadi, pada citra biner, latar belakang berwarna putih sedangkan objek berwarna hitam [11].

C. *Intranet*

Merupakan suatu konsep *Local Area Network* (LAN) yang mengadopsi teknologi internet, sehingga didalam intranet harus memiliki komponen-komponen yang membangun internet seperti protokol TCP/IP, alamat IP dan protokol lainnya, klien dan juga server [12].

D. *Chaos Map*

Teori chaos menggambarkan kebiasaan dari suatu sistem dinamis, yang keadaannya selalu berubah seiring dengan berubahnya waktu, dan sangat sensitif terhadap kondisi awal dirinya sendiri. Teori chaos ini juga sering disebut dengan sebutan *butterfly effect*.



Gambar 2. *Butterfly Effect*

Dikarenakan oleh sensitivitas yang dimiliki teori chaos terhadap keadaan awal dirinya, teori chaos memiliki sifat untuk muncul secara chaos (kacau). Bahkan perubahan keadaan awal sekecil (10^-100) saja akan membangkitkan bilangan yang benar-benar berbeda. Hal ini sangat berguna dan dapat diterapkan di dalam dunia kriptografi sebagai pembangkit kunci acak yang nantinya akan diolah sebagai sarana dalam melakukan proses enkripsi. Semakin acak bilangan yang dihasilkan, semakin baik pula tingkat keamanan dari suatu chiperteks [3].

E. *Arnold Cat Map (ACM)*

Merupakan fungsi chaos dwimatra dan bersifat reversible. Fungsi chaos ini ditemukan oleh Vladimir Arnold pada tahun 1960, dan kata “cat” muncul karena dia menggunakan citra seekor kucing dalam eksperimennya .ACM mentransformasikan koordinat (x,y) di dalam citra yang berukuran N x N ke koordinat baru (x’ ,y’). Persamaan iterasinya adalah :

$$\begin{bmatrix} X\,\text{i-1} \\ Y\,\text{i-1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix} \begin{bmatrix} X\,\text{i} \\ Y\,\text{i} \end{bmatrix} \bmod N \qquad (1)$$

Dalam hal ini (xi , yi) adalah posisi pixel di dalam citra, (x i-1 , y i-1) posisi pixel yang baru setelah iterasi ke-I; b dan c adalah integer positif sembarang.

Determinan matriks $\begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix}$ harus sama dengan 1 agar hasil transformasinya bersigat area-preserving, yaitu tetap berada di dalam area citra yang sama. ACM termasuk pemetaan yang bersifat satu ke satu karena setiap posisi pixel selalu ditransformasikan ke posisi lain secara unik. ACM diiterasikan sebanyak m kali dan setiap iterasi menghasilkan citra yang acak. Nilai b, c, dan jumlah iterasi m dapat dianggap sebagai kunci rahasia.

Proses yang terjadi di dalam setiap iterasi ACM adalah pergeseran (*shear*) dalam arah y, kemudian dalam arah x, dan semua hasilnya (yang mungkin berada di luar area gambar) dimodulokan dengan N agar tetap berada di dalam area gambar (*area preserving*) [2].

F. *Logistic Map*

Persamaan logistik merupakan contoh pemetaan polinomial derajat dua, dan seringkali digunakan sebagai contoh bagaimana rumitnya sifat chaos (kacau) yang dapat muncul dari suatu persamaan yang sangat sederhana. Persamaan ini dipopulerkan oleh seorang ahli biologi yang bernama Robert May pada

tahun 1976, melanjutkan persamaan logistik yang dikembangkan oleh Pierre Francois Verhulst.

Secara matematis, persamaan logistik dapat dinyatakan dengan persamaan:

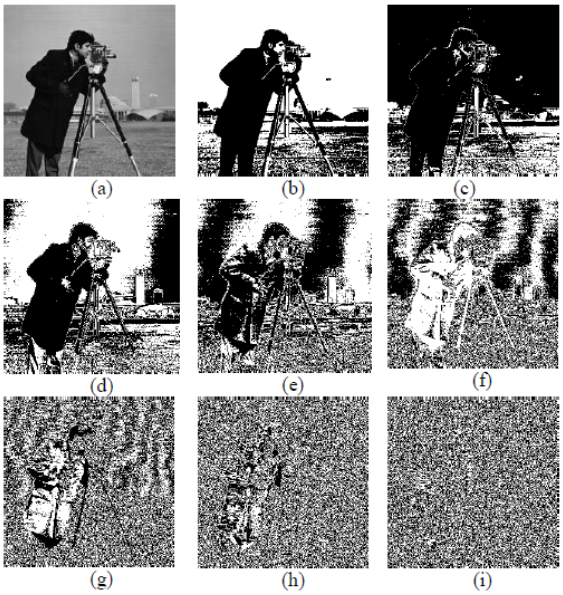
$$X_{i+1} = r_{xi} (1- xi) \quad (2)$$

nilai-nlai xi adalah bilangan riil di dalam selang (0, 1), sedangkan r adalah parameter fungsi yang menyatakan laju pertumbuhan yang nilainya di dalam selang (0, 4). Logistic Map akan bersifat chaos bilamana $0 \leq r \leq 4$. Untuk memulai iterasi Logistic Map diperlukan nilai awal x0. Perubahan sedikit saja pada nilai awal ini (misalnya sebesar 10-10) akan menghasilkan nilai-nilai chaos yang berbeda secara signifikan setelah Logistc Map diiterasi sejumlah kali. Di dalam sistem kriptografi simetri, nilai awal chaos, x0, dan parameter μ berperan sebagai kunci rahasia. Nilai-nilai acak yang dihasilkan dari persamaan (3) tidak pernah berulang kembali sehingga *logistic map* dikatakan tidak mempunyai periode [2].

G. *Enkripsi Selektif*

Pengubahan bit *Most Significant Bit* (MSB) menjadi dasar enkripsi selektif citra digital. Dengan hanya mengenkripsi bit MSB maka proses enkripsi menjadi lebih efisien, sebab tidak semua data citra dienkripsi dengan *stream cipher*.

Setiap pixel direpresentasikan dalam sejumlah byte, susunan bit pada setiap byte adalah b7 b6 b5 b4 b3 b2 b1 b0. Bit – bit paling kiri adalah bit paling berarti (MSB), sedangkan bit – bit paling kanan adalah least significant bits atau LSB. Jika setiap bit ke-1 dari setiap pixel pada citra abu-abu diekstrasi dan diplot ke dalam setiap *bitplane image* maka akan diperoleh citra seperti pada gambar 3 [4].



Gambar 3. Mekanisme Enkripsi Selektif

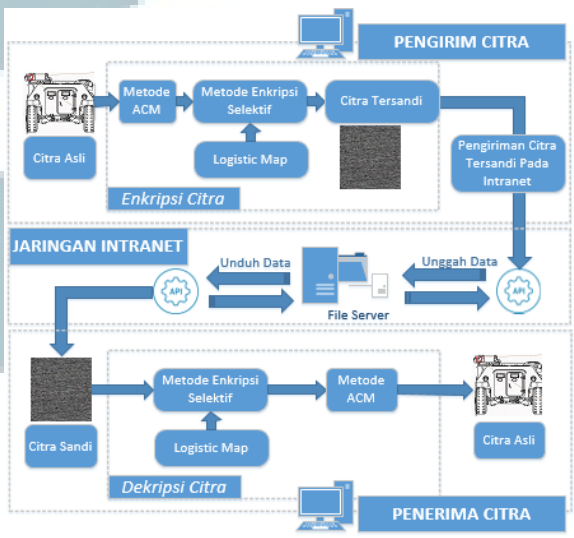
III. ANALISIS DAN PERANCANGAN SISTEM

A. *Gambaran Umum Sistem*

Sistem yang akan dibangun pada penelitian ini adalah sistem kriptografi citra digital , dimana pengirim mengirimkan gambar yang telah di enkripsi sebelumnya menggunakan algoritma *Chaos Map* dan penerapan teknik selektif, selanjutnya citra dalam bentuk *cipher image* dikirim melalui sistem jaringan intranet dan penerima citra bertugas mengembalikan *cipher image* ke bentuk semula (*plain image*).

Sistem yang dikembangkan merupakan sistem yang bekerja di lingkungan jaringan intranet organisisasi, dimana antara divisi desain dan divisi produksi berada pada lokasi yang berjauhan sehingga membutuhkan suatu akses jaringan komputer intranet guna memfasilitasi pengiriman data citra desain yang diperlukan.

Aplikasi pada sisi pengirim citra melakukan enkripsi pada citra yang akan dikirimkan dan meneruskannya (unggah file) ke server jaringan intranet melalui suatu API (*Application Programming Interface*). Sementara pada sisi penerima citra , melakukan proses unduh menggunakan API guna mendapatkan data citra yang dikirm dan melakukan proses dekripsi guna dapat melaihat citra asli yang dikirimkan. Ilustrasi sistem yang dibangun dapat dilihat pada gambar 4.



Gambar 4. Gambaran Umum Sistem Kriptografi

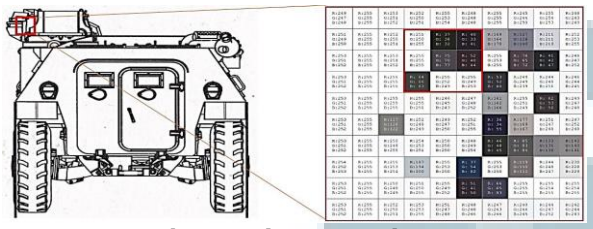
B. *Analisis Algoritma*

Analisis algoritma digunakan untuk mengetahui alur proses dari algoritma yang digunakan untuk dapat diterapkan ke dalam aplikasi yang dibangun. Pembangunan aplikasi ini menggunakan metode

kombinasi *chaos map* yaitu *arnold cat map* dan *logistic map* serta penerapan teknik selektif

C. *Analisis Algoritma Arnold Cat Map*

Pada tahap ini dilakukan operasi permutasi citra dengan ACM yang bertujuan mengacak susunan pixel-pixel di dalam citra. Berikut langkah pengacakan menggunakan citra bimau berukuran 50 x 50. Pengacakan citra dilakukan pada 9 bit tetangga yang diambil dari citra. Gambar merupakan citra dengan nilai RGB.



Gambar 5. Nilai RGB pada citra

Parameter *Arnold Cat Map* yang diinputkan m = 2, b = 2, dan c = 3, kemudian dengan transformasi ACM dan menghasilkan iterasi pertama m=1.

$$\begin{bmatrix} Xi+1 \\ Yi+1 \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix} \begin{bmatrix} Xi \\ Yi \end{bmatrix} Mod\ (N)$$

$$\begin{bmatrix} Xi+1 \\ Yi+1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} Mod\ (3) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} Xi+1 \\ Yi+1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 7 \end{bmatrix} Mod\ (3) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} Xi+1 \\ Yi+1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 14 \end{bmatrix} Mod\ (3) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} Xi+1 \\ Yi+1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} Mod\ (3) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} Xi+1 \\ Yi+1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 10 \end{bmatrix} Mod\ (3) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} Xi+1 \\ Yi+1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 17 \end{bmatrix} Mod\ (3) = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} Xi+1 \\ Yi+1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \end{bmatrix} Mod\ (3) = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} Xi+1 \\ Yi+1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 13 \end{bmatrix} Mod\ (3) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

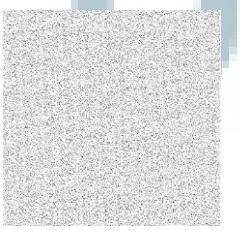
Iterasi 1

	0	1	2
0	1	5	9
1	4	8	3
2	7	2	6

Iterasi 2

	0	1	2
0	1	8	6
1	4	2	9
2	7	5	3

Hasil dari iterasi ACM pada citra dapat dilihat pada gambar 6 berikut:



Gambar 6. Hasil Iterasi ACM Pada Citra

D. *Analisis Pembangkitan Kunci Algoritma Logistic Map*

Untuk mengenkripsi bit-bit MSB dengan stream cipher diperlukan barisan bit kunci rahasia yang menjadi *keystream*. Bit-bit ini dibangkitkan dari fungsi logistic map dengan mengiterasikan persamaan (3) subbab 2.2.11. Nilai awal logistic map (x0) berperan sebagai kunci enkripsi. Namun, keluaran dari logistic map tidak dapat langsung di-XOR-kan dengan bit-bit plainteks karena masih berbentuk bilangan riil antara 0 dan 1.

X1 = 4.0x0(1-x0) = 4.0(0.456)(1 – 0.456) = 0.992256
X2 = 4.0x1(1-x1) = 4.0(0.992256)(1 – 0.992256) = 0.030736
X3 = 4.0x2(1-x2) = 4.0(0.030736)(1 – 0.030736) = 0.119166
...
X100 = 4.0x99(1-x99) = 4.0(0.914379)(1 – 0.914379) = 0.313162

Agar barisan nilai chaotik dapat dipakai untuk enkripsi dan dekripsi dengan stream cipher, maka nilai – nilai chaos tersebut dikonversikan ke nilai integer. Selanjutnya, 1 bit MSB diekstraksi dari nilai integer tersebut. Bit MSB inilah yang menjadi bit kunci untuk enkripsi.

Transformasi yang sederhana adalah dengan mengambil bagian desimal dari bilangan riil, membuang angka nol yang tidak signifikan, lalu mengekstrak t digit integer. Karena nilai-nilai pixel berada di dalam rentang integer [0,255], maka sebelum di-XOR-kan keystream di modulus-kan dengan 256. Jadi pada contoh ini ki = 4358 mod 256 = 6. Pada citra digital yang digunakan, telah mengambil 9 bit tetangga dengan nilai RGB, dan mengiterasikan persamaan ACM pada citra sehingga pixel pada citra telah teracak. Selanjutnya enkstraksi 4 bit MSB dari setiap pixel citra, nyatakan setiap 4 bit tersebut sebagai pi(i = 1, 2, 3,….,n) n = N x N.

Transformasi dengan mengambil bagian desimal dari bilangan riil, membuang angka nol yang tidak signifikan, lalu mengekstrak t digit integer. Ambil nilai desimal dan ekstrak 4 digit yang akan menjadi nilai keystream yang akan di-XOR-kan dengan pi, kemudian moduluskan dengan 256.

X1 = 0.89344 = 8934 mod 256 = 230
230 = 11100110 ; 4 Bit MSB nya adalah k1 = 1110
X2 = 0.361778 = 3617 mod 256 = 33
33 = 00100001 ; 4 Bit MSB nya adalah k2 = 0010
X3 = 0.877399 = 8773 mod 256 = 69
69 = 01000101 ; 4 Bit MSB nya adalah k3 = 0100
X4 = 0.408765 = 4087 mod 256 = 247
247 = 11110101 ; 4 Bit MSB nya adalah k4 = 1111

E. *Analisis Teknik Enkripsi Selektif*

Pixel-pixel citra di dalam struktur citra bitmap berukuran sejumlah byte. Pada citra 8-bit satu pixel berukuran 1 byte (8-bit, sedangkan pada citra 24-bit satu pixel berukuran 3 byte (24 bit). Di dalam susunan byte terdapat bit yang paling tidak berarti (LSB) dan bit yang paling berarti (MSB).

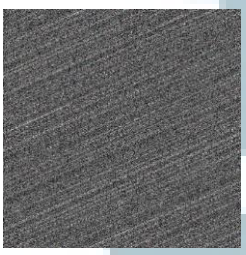
Pengubahan 1 bit LSB tidak mempengaruhi nilai byte secara signifikan, namun pengubahan 1 bit MSB dapat mengubah nilai byte secara signifikan. Secara visual efek perubahan bit-bit MSB pada seluruh pixel

citra menyebabkan citra tersebut menjadi rusak sehingga citra tidak dapat dipersepsi dengan jelas. Inilah yang menjadi dasar enkripsi citra yang bertujuan membuat citra tidak dapat dikenali lagi karena sudah berubah menjadi bentuk yang tidak jelas [4].

Tabel 1. Perubahan bit MSB

4 Bit MSB	Nilai RGB	Nilai RGB (Biner)	CI (4 Bit MSB *Chiper*)	Nilai RGB Setelah nilai 4 bit MSB diganti	Nilai RGB *Chiper*
P_1	R: 37 G: 38 B: 32	= 00100101 = 00100110 = 00100000	1100	= 11000101 = 11000110 = 11000000	R: 197 G: 198 B: 192
P_2	R: 40 G: 33 B: 41	= 00101000 = 00100001 = 00101001	0000	= 00001000 = 00000001 = 00001001	R: 8 G: 1 B: 9
P_3	R: 169 G: 164 B: 170	= 10101001 = 10101000 = 10101010	1110	= 11101001 = 11101000 = 11101010	R: 233 G: 232 B: 234

Hasil citra yang telah dienkripsi dapat dilihat pada gambar 7.

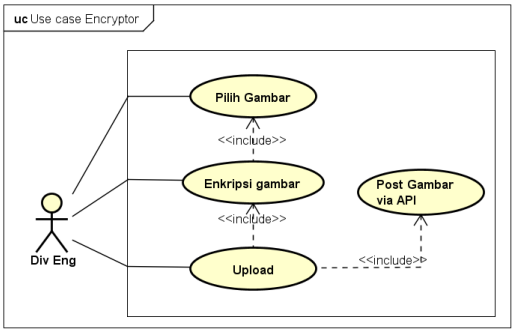


Gambar 7. Hasil Citra Yang Telah Dienkripsi

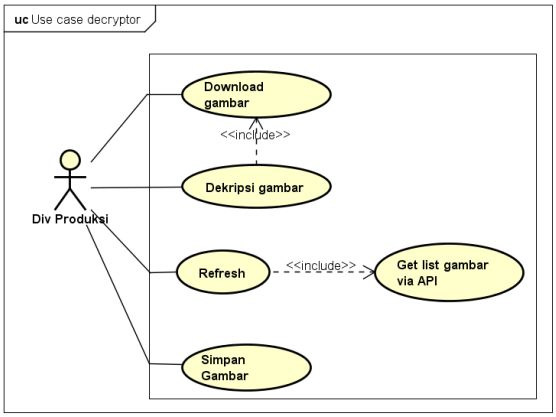
F. *Diagram Use Case*

Diagram *use case* menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Diagram use case merepresentasikan sebuah interaksi antar user sebagai aktor dengan sistem. Seseorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [8].

Diagram use case dari sistem kriptografi citra digital ini terbagi menjadi 2 yaitu, diagram *use case* pada aplikasi *encryptor* (gambar 8) dan diagram *use case* pada aplikasi *decryptor* (gambar 9).



Gambar 8. Diagram Use Case *Encryptor*

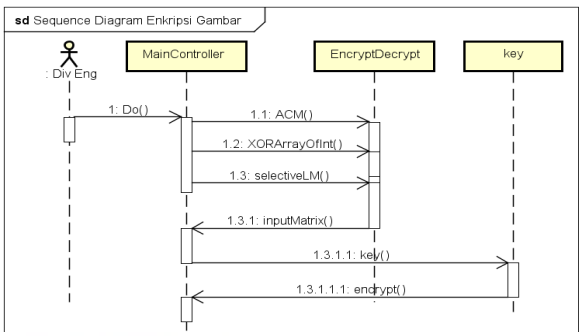


Gambar 9. Diagram Use Case *Decryptor*

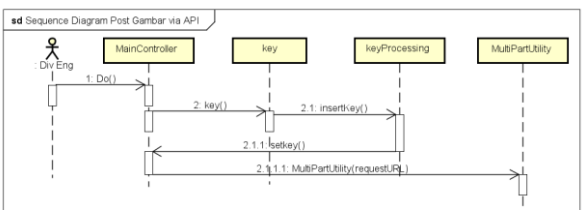
G. *Diagram Sequence*

Sequence diagram menggambarkan interaksi antar objek di dalam dan diluar sistem (termasuk pengguna, display, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence* diagram terdiri dari dimensi vertikal (waktu) dan dimensi horizontal (objek yang terkait) yang biasanya digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu.

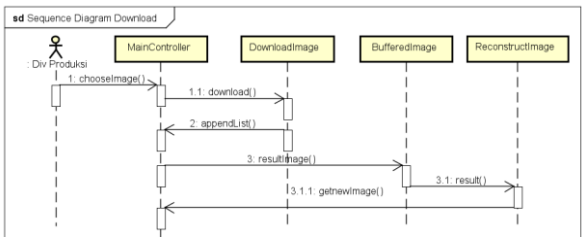
Adapun *sequence* diagram untuk enkripsi dapat dilihat pada gambar 10, *sequence* diagram kirim gambar via API pada gambar 11 dan *sequence* diagram untuk unduh dan dekripsi dapat dilihat pada gambar 12 dan 13.



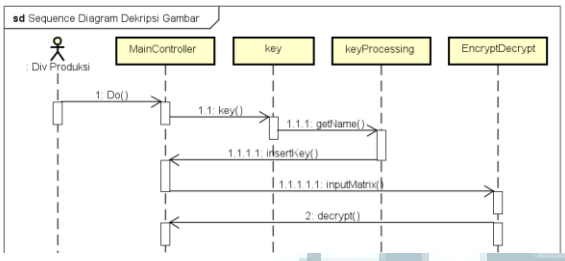
Gambar 10. *Sequence* Diagram Enkripsi Citra



Gambar 11. *Sequence* Diagram Post Citra via API



Gambar 12. *Sequence* Diagram Download Citra via API

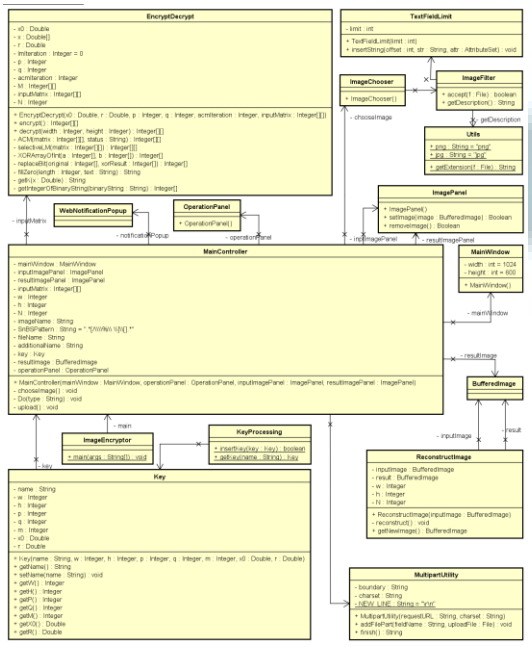


Gambar 13. *Sequence* Diagram Dekripsi Citra

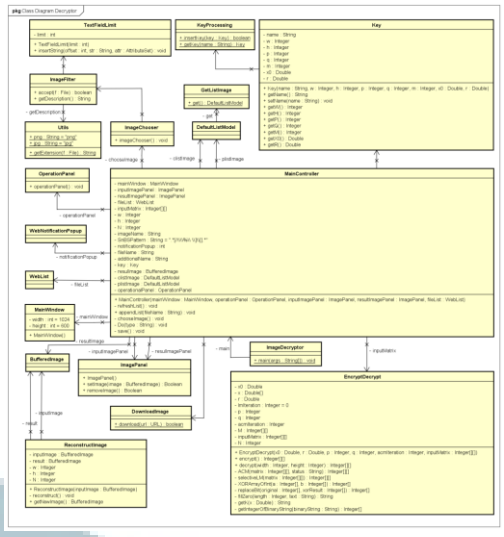
H. Diagram Class

Diagram *class* digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem. Diagram *class* memberikan gambaran sistem secara statis.. Diagram *class* sangat membantu dalam visualisasi struktur kelas dari suatu sistem [9].

Diagram *class* pada aplikasi sistem keamanan enkripsi citra digital terbagi menjadi 2, yaitu diagram *class* untuk *encryptor* (gambar 14) dan diagram *class* untuk *decryptor* (gambar 15).



Gambar 14. Diagram Class *Encryptor*



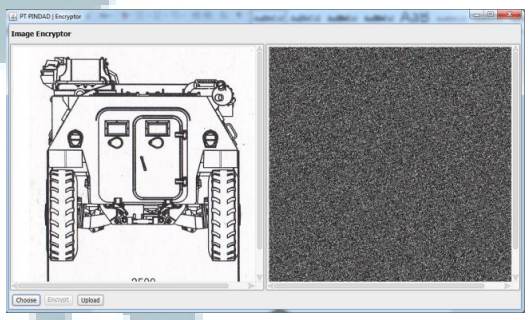
Gambar 15. Diagram Class *Decryptor*

IV. IMPLEMENTASI DAN PENGUJIAN SISTEM

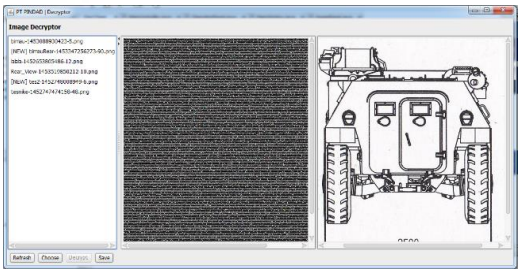
A. Implementasi

Hasil dari penelitian ini adalah sistem kritografi citra yang di*install* pada komputer pengguna, baik sebagai *encryptor* maupun *decryptor*.

Antarmuka sistem kritografi citra dapat dilihat pada gambar 16 untuk mekanisme enkripsi citra, sedangkan gambar 17 menunjukkan mekanisme dekripsi citra.



Gambar 16. Antarmuka Enkripsi Citra



Gambar 17. Antarmuka Dekripsi Citra

B. Pengujian Sistem

Setelah melakukan tahap implementasi, maka tahapan selanjutnya adalah pengujian sistem yang dibuat. Pengujian ini dilakukan untuk mengetahui

apakah rancangan dan implementasi yang telah dilakukan berjalan sesuai dengan prosedur yang diinginkan atau tidak.

B 1. Pengujian Blackbox

Pengujian *blackbox* bertujuan mengukur kinerja dari perangkat lunak apakah fungsinya dapat berjalan dengan baik atau tidak [7]. Berdasarkan rencana pengujian yang disusun, maka dilakukan pengujian seperti yang dicantumkan pada tabel 2.

Tabel 2. Pengujian *blackbox*

Aktivitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Memilih gambar yang akan di enkripsi	Gambar berhasil ditampilakn ada panel image	Berhasil Menampilkan gambar	Diterima
Mengenkripsi gambar yang telah dipilih	Menampilkan hasil enkripsi pada panel image enkripsi	Berhasil menampilkan gambar *cipher image*	Diterima
Mendekripsi gambar yang dipilih	Menampilkan gambar *cipher* pada panel image	Berhasil menampilkan *plain image*	Diterima

B.2. Pengujian Whitebox

Pengujian *whitebox* digunakan untuk mengetahui logika yang dibuat pada sebuah perangkat lunak apakah berjalan dengan baik atau tidak [7].

Pengujian *whitebox* akan digunakan pada algoritma *Chaos Map* dan teknik selektif, untuk mengukur kinerja logika berdasarkan *pseudocode* yang telah dibuat pada tahap analisis.

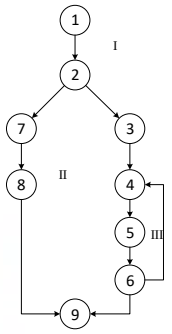
1. Langkah pertama ubah *pseudocode* menjadi *flowchart*
2. Ubah *flowchart* menjadi *flowgraph* ke dalam bentuk yang lebih sederhana.
3. Tahap pengujian, dimana tahap pengujian ini dilakukan dengan 5 cara yaitu, menghitung *region*, menghitung *cyclomatic complexity*, menghitung *independent path*, menggunakan *graph matriks*, dan menghitung *predicate node*.

Pengujian *whitebox* dimulai dengan pengujian *generate* XOR yang dilakukan pada algoritma teknik selectif yang meng-XOR-kan nilai 4-bit MSB dengan *keystream* yang dibangkitkan dari *logistic map*. Tabel 3 merupakan *source code* pengujian *generate* XOR, yang akan menghasilkan *flowgraph* pada gambar 18.

Tabel 3. *Sourcecode generate XOR*

Line	Source

```
1  Integer[] result;
2    if (a.length == b.length) {
3           result = new Integer[a.length];
4           for (int i = 0; i < a.length; i++) {
5                   result[i] = a[i] ^ b[i]; }
6           return result;
7           } else {
8           return null;
9  }
```



Gambar 18. *Flowgraph Generate XOR*

Tahap selanjutnya adalah menghitung *region* hingga *predicate note*.

Hitung Region = 3
Hitung *cyclomatic complexity* yaitu sebagai berikut:

$V_{(G)}$ = Edge – Node + 2
= 10 – 9 + 2
= 3

Berdasarkan pada hasil *cyclomatic complexity* maka didapat dua *independent path* yaitu :

Path 1 = 1-2-3-4-5-6-9
Path 2 = 1-2-7-8-9
Path 3 = 1-2-3-4-5-6-4-5-6-9

Graph matriks generate XOR dapat dilihat pada tabel 4.

Tabel 4. *Graph matriks generate XOR*

Node	1	2	3	4	5	6	7	8	9	Sum
1		1								0
2			1				1			1
3				1						0
4					1					0
5						1				0
6				1					1	1
7								1		0
8									1	0
9										0
Total										2

$V_{(G)}$ = Jumlah *Graph Matriks* + 1
= 2 + 1
= 3

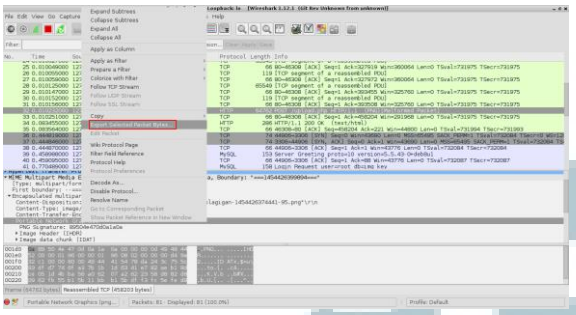
Predicate Node

$V_{(G)}$ = Jumlah node yang memiliki jalur lebih dari 1 jalur
= 2 + 1

= 3

B.3. Pengujian Keamanan Citra

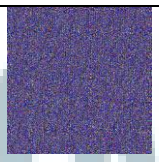
Pengujian keamanan citra terenkripsi mengunakan mekanisme *Man In The Middle Attack* (MITM) yaitu adanya komputer di intranet yang bertindak sebagai penyerang dengan menggunakan aplikasi *wireshark* untuk melakukan tindak penyadapan terhadap data citra yang dikirimkan seperti yang terlihat pada gambar 19.



Gambar 19. *Penyadapan dengan Wireshark*

Hasil yang diperoleh dari penyadapan yang dilakukan oleh aplikasi *wireshark* dapat dilihat pada tabel 5.

Tabel 5. *Pengujian Keamanan Dengan Wireshark*

Citra Uji	**Jenis**	**Pengamatan Pada Wireshark**
 Senjata.jpg	Tidak Terekripsi	
 Senjata.jpg	Terekripsi	

V. SIMPULAN

Berdasarkan hasil analisis, perancangan dan implementasi maka penelitian ini telah dapat menghasilkan hal-hal sebagai berikut.

1. Sistem keamanan citra digital yang menggabungkan algoritma chaos yaitu *arnold cat map* dan *logistik map* dan penerapan teknik selektif mampu memengkripsi citra dengan baik.
2. Pengguna aplikasi ini tidak perlu menyisipkan kunci karena sistem membangkitkan kunci bersama dengan nama yang diinputkan pada citra yang telah dienkripsi.
3. Pengujian *blackbox* menunjukkan bahwa fungsionalitas sistem berjalan dengan baik.
4. Pengujian *Whitebox* pada setiap metode, dihasilkan nilai *cyclomatic complexity* yang sama yaitu 3. Maka dapat disimpulkan bahwa pengujian *whitebox* pada proses *generate* XOR berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.
5. Pengujian pengiriman citra di intranet menunjukkan citra yang telah terenkripsi tidak dapat dilihat aslinya seperti halnya yang belum terenkripsi.

Sementara untuk pengembangan sistem selanjutnya, terdapat beberapa hal yang perlu dilakukan yaitu :

1. Pengembangan tampilan sistem yang lebih interaktif.
2. Citra yang akan dikirim tidak hanya satu , melainkan beberapa citra.
3. Diharapkan pada pengembangan selanjutnya algoritma chaos yang digunakan tidak hanya *arnold cat map* dan *logistic map*, melainkan beberapa algoritma chaos lainnya.

UCAPAN TERIMA KASIH

Terima kasih diucapkan kepada PT.PINDAD Bandung, yang telah bersedia menjadi mitra penelitian yang dilakukan..

DAFTAR PUSTAKA

[1] Munir, R.,. “Analisis Keamanan Algortima Enkripsi Citra Digital Menggunakan Kombinasi *Chaos Map* dan Penerapan Teknik Selektif”, Jurnal Teknik Informatika (JUTI) Vol 10, Nomor 2, 2012.

[2] Munir, R. “Algortima Enkripsi Citra Digital Berbasis Chaos Dengan Penggabungan Teknik Permutasi dan Teknik Subtitusi Menggunakan *Arnold Cat Map* dan Logistic Map”., Jurnal Seminar Nasional Pendidikan Nasinal (SENAPATI 2012), 2012.

[3] Susanto, A. “Penerapan Teori Chaos di Dalam Kriptografi”. http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2008-2009/Makalah2/MakalahIF3058-2009-b031.pdf. Diakses 16 Januari 2016.

[4] Munir, R. “Algoritma Enkripsi Citra Digital Dengan Kombinasi *Chaos Map* Dan Penerapan Teknik Selektif Terhadap Bit-Bit MSB”. Jurnal Seminar Nasional Aplikasi Teknologi Informasi (SNATI 2012), 2012.

[5] Sadikin, R. “Kriptografi Untuk Keamanan Jaringan”, Yogyakarta, Penerbit Andi, 2012.

[6] Wahana Komputer. "Kriptografi Dalam Memahami Model Enkripsi Dan Security Data”, Yogyakarta, Penerbit Andi, 2003.

[7] Irwan, M. “Blackbox Testing and Whitebox Testing”, http://tkjpnup.blogspot.co.id/2013/12/black-box-testing-dan-white-box-testing.html. Diakses 14 Desember 2015.

[8] Sutopo, A.H., “Analsis dan Desain Berorintasi Objek”, Yogtyakarta, J&J Learning, 2002

[9] Irwanto, D., “Perancangan Object Oriented Software dengan UML, Yogyakarta, Penerbit Andi, 2006

[10] Kadir Abdul, Susanto Adhi, Teori dan Aplikasi Pengolahan Citra, Yogyakarta, Penerbit Andi, 2013.

[11] Munir, R., “Pengolahan Citra Digital” , Bandung, Penerbit Informatika, 2004.

[12] Nurkamid, Mukhamad. "Analisa Keefektifan Jaringan Local Area Network (intranet) Universitas Muria Kudus." *Jurnal Sains dan Teknologi* , Vol.4 no. 2 : hal.143-150, 2013.

PEDOMAN PENULISAN JURNAL ULTIMATICS, ULTIMA INFOSYS, DAN ULTIMA COMPUTING

1. Kriteria Naskah

- Naskah belum pernah dipublikasikan atau tidak dalam proses penyuntingan di jurnal berkala lainnya.
- Naskah yang dikirimkan dapat berupa naskah hasil penelitian atau konseptual.

2. Pengetikan Naskah

- Naskah diketik dengan jarak spasi antar baris 1 pada halaman ukuran A4 (21 cm x 29,7 cm), margin kiri-atas 3 cm dan kanan-bawah 2 cm, dengan jenis tulisan Times New Roman.
- Naskah dapat ditulis dalam Bahasa Indonesia atau Bahasa Inggris.
- Jumlah halaman untuk tiap naskah dibatasi dengan jumlah minimal 4 halaman dan maksimal 8 halaman.

3. Format Naskah

- Komposisi naskah terdiri dari Judul, Abstrak, Kata Kunci, Pendahuluan, Metode, Hasil Penelitian dan Pembahasan, Simpulan, Lampiran, Ucapan Terima Kasih, dan Daftar Pustaka.
- Judul memiliki jumlah kata maksimal 15 kata dalam Bahasa Indonesia atau maksimal 12 kata dalam Bahasa Inggris (termasuk subjudul bila ada).
- Abstrak ditulis dengan Bahasa Inggris paling banyak 200 kata, meskipun bahasa yang digunakan dalam penyusunan naskah adalah Bahasa Indonesia. Isi abstrak sebaiknya mengandung argumentasi logis, pendekatan pemecahan masalah, hasil yang dicapai, dan simpulan singkat.
- Kata Kunci ditulis dengan Bahasa Inggris dalam satu baris, dengan jumlah kata antara 4 sampai 6 kata.
- Pendahuluan berisi latar belakang dan tujuan penelitian.
- Metode dapat diuraikan secara terperinci dan dibedakan menjadi beberapa bab maupun subbab yang terpisah.
- Hasil dan Pembahasan disajikan secara sistematis sesuai dengan tujuan penelitian.
- Simpulan menyajikan intisari hasil penelitian yang telah dilaksanakan. Saran pengembangan untuk penelitian selanjutnya juga dapat diberikan di sini.
- Lampiran dan Ucapan Terima Kasih dapat dijabarkan setelah Simpulan secara singkat dan jelas.
- Daftar Pustaka yang dirujuk dalam naskah harus dituliskan di bagian ini secara kronologis berdasarkan urutan kemunculannya. Cara penulisannya mengikuti cara penulisan jurnal dan transaction IEEE.
- Template naskah telah disediakan dan dapat diminta dengan menghubungi surel redaksi.

4. Penulisan Daftar Pustaka

- Artikel Ilmiah:
  N. Penulis, "Judul artikel ilmiah," *Singkatan Nama Jurnal*, vol. x, no. x, hal. xxx-xxx, Sept. 2013.

- Buku
  N. Penulis, "Judul bab di dalam buku," di dalam *Judul dari Buku*, edisi x. Kota atau Negara Penerbit: Singkatan Nama Penerbit, tahun, bab x, subbab x, hal. xxx-xxx.

- Laporan
  N. Penulis, "Judul laporan," Singkatan Nama Perusahaan, Kota Perusahaan, Singkatan Nama Negara, Laporan xxx, tahun.

- Buku Manual/ *handbook*
  Nama dari Buku Manual, edisi x, Singkatan Nama Perusahaan, Kota Perusahaan, Singkatan Nama Negara, tahun, hal. xxx-xxx.

- Prosiding
  N. Penulis, "Judul artikel," di dalam *Nama Konferensi Ilmiah*, Kota Konferensi, Singkatan Nama Negara (jika ada), tahun, hal. xxx-xxx.

- Artikel yang Disajikan dalam Konferensi
  N. Penulis, "Judul artikel," disajikan di Nama Konferensi, Kota Konferensi, Singkatan Nama Negara, tahun.

- Paten
  N. Penulis, "Judul paten," HKI xxxxxx, 01 Januari 2014.

- Tesis dan Disertasi
  N. Penulis, "Judul tesis," M.Sc. thesis, Singkatan Departemen, Singkatan

Universitas, Kota Universitas, Singkatan Nama Negara, tahun.

N. Penulis, "Judul disertasi," Ph.D. dissertation, Singkatan Departemen, Singkatan Universitas, Kota Universitas, Singkatan Nama Negara, tahun.

- Belum Terbit
  N. Penulis, "Judul artikel," belum terbit.

  N. Penulis, "Judul artikel," Singkatan Nama Jurnal, proses cetak.

- Sumber online
  N. Penulis. (tahun, bulan tanggal). Judul (edisi) [Media perantara]. Alamat situs: http://www.(URL)

  N. Penulis. (tahun, bulan). Judul. Jurnal [Media perantara]. *volume(issue)*, halaman jika ada. Alamat situs: http://www.(URL)

  Catatan: media perantara dapat berupa media online, CD-ROM, USB, dan sebagainya.

5. Pengiriman Naskah Awal

- Para penulis dapat mengirimkan naskah hasil penelitiannya dalam bentuk .doc atau .pdf melalui surel ke umnjurnal@gmail.com dengan subjek sesuai Jurnal yang dipilih.
- Seluruh isi naskah yang dikirimkan harus memenuhi syarat dan ketentuan yang ditentukan.
- Kami akan menjaga segala kerahasiaan dan Hak Cipta karya Anda.
- Sertakan biodata penulis pertama yang lengkap, meliputi nama, alamat kantor, alamat penulis, telpon kantor/ rumah dan hp, serta No NPWP (bagi yang memiliki NPWP).

6. Penilaian Naskah

- Seluruh naskah yang diterima akan melalui serangkaian tahap penilaian yang melibatkan mitra bestari.
- Setiap naskah akan direview oleh minimal 2 orang mitra bestari.
- Rekomendasi dari mitra bestari yang akan menentukan apakah sebuah naskah diterima, diterima dengan revisi minor, diterima dengan revisi major, atau ditolak.

7. Pengiriman Naskah Final

- Naskah yang diterima untuk diterbitkan akan diinformasikan melalui surel redaksi.
- Penulis berkewajiban memperbaiki setiap kesalahan yang ditemukan sesuai saran dari mitra bestari.
- Naskah final yang telah direvisi dapat dikirimkan kembali ke surel redaksi beserta hasil scan Copyright Transfer Form yang telah ditandatangani.

8. Copyright dan Honorarium

- Penulis yang naskahnya dimuat harus membaca dan menyetujui isi Copyright Transfer Form kepada redaksi.
- Copyright Transfer Form harus ditandatangani oleh penulis pertama naskah.
- Naskah yang dimuat akan mendapatkan honorarium sebesar Rp 1.000.000,- per naskah, setelah dipotong pajak 2.5% (bila penulis pertama yang memiliki NPWP) dan 3% (tanpa NPWP).
- Honorarium akan ditransfer ke rekening penulis pertama (tidak dapat diwakilkan) paling lambat 2 minggu setelah jurnal naik cetak dan siap didistribusikan.
- Penulis yang naskahnya dimuat akan mendapatkan copy jurnal sebanyak 2 eksemplar.

9. Biaya Tambahan

- Permintaan tambahan copy jurnal harus dibeli seharga Rp 50.000,- per copy.
- Permintaan penambahan jumlah halaman dalam naskah (maksimal 8 halaman) akan dikenai biaya sebesar Rp 25.000,- per halaman.

10. Alamat Redaksi

d.a. Koordinator Riset
Fakultas Teknologi Informasi dan Komunikasi
Universitas Multimedia Nusantara
Gedung Rektorat Lt.6
Scientia Garden, Jl. Boulevard Gading Serpong, Tangerang, Banten -15333
Surel: **umnjurnal@gmail.com**

Judul Paper

Sub Judul (jika diperlukan)

Nama Penulis A[1], Nama Penulis B[2], Nama Penulis C[2]
[1] Baris pertama (dari afiliasi): nama departemen organisasi, nama organisasi, kota, negara
Baris kedua: alamat surel jika diinginkan
[2] Baris pertama (dari afiliasi): nama departemen organisasi, nama organisasi, kota, negara
Baris kedua: alamat surel jika diinginkan

Diterima dd mmmmm yyyy
Disetujui dd mmmmm yyyy

***Abstract*—This electronic document is a "live" template which you can use on preparing your IJNMT paper. Use this document as a template if you are using Microsoft Word 2007 or later. Otherwise, use this document as an instruction set. Do not use symbol, special characters, or Math in Paper Title and Abstract. Do not cite references in the abstract.**

***Index Terms*—enter key words or phrases in alphabetical order, separated by commas**

I. PENDAHULUAN

Dokumen ini, dimodifikasi dalam MS Word 2007 dan disimpan sebagai dokumen Word 97-2003, memberikan panduan yang diperlukan oleh penulis untuk mempersiapkan dokumen elektroniknya. Margin, lebar kolom, jarak antar baris, dan jenis-jenis format lainnya telah disisipkan di sini. Penulis berkewajiban untuk memastikan dokumen yang dipersiapkannya telah memenuhi format yang disediakan.

Isi Pendahuluan mengandung latar belakang, tujuan, idenfikasi masalah dan metode penelitian yang dipaparkan secara tersirat (implisit). Kecuali bab Pendahuluan dan Simpulan, penulisan judul bab sebaiknya eksplisit sesuai dengan isi yang dijelaskan, tidak harus implisit dinyatakan sebagai Dasar Teori, Perancangan, dan sebagainya.

II. PENGGUNAAN YANG TEPAT

A. Memilih Template

Pertama, pastikan Anda memiliki *template* yang tepat untuk artikel Anda. *Template* ini ditujukan untuk Jurnal ULTIMATICS, ULTIMA InfoSys, dan ULTIMA Computing. *Template* ini menggunakan ukuran kertas A4.

B. Mempertahankan Keutuhan Format

Template ini digunakan untuk mem-format artikel dan *style* isi artikel Anda. Seluruh margin, lebar kolom, jarak antar baris, dan jenis tulisan telah diberikan, jangan diubah.

III. PERSIAPKAN ARTIKEL ANDA

Sebelum Anda mulai mem-format artikel Anda, tulislah terlebih dahulu artikel Anda dan simpan sebagai *text file* lainnya. Setelah selesai baru lakukan pencocokkan *style* dokumen. Jangan tambahkan nomor halaman di bagian manapun dari dokumen ini. Perhatikan pula beberapa hal berikut saat melakukan pengecekan tulisan.

A. Singkatan

Definisikan singkatan pada saat pertama kali digunakan di dalam isi tulisan, walaupun singkatan tersebut telah didefinisikan di dalam abstrak. Singkatan seperti IEEE, SI, MKS, CGS, sc, dc, dan rms tidak harus didefinisikan. Singkatan yang menggunakan tanda titik tidak boleh diberi spasi, seperti "C.N.R.S.", bukan "C. N. R. S." Jangan gunakan singkatan di dalam Judul Artikel atau Judul Bab, kecuali tidak dapat dihindari.

B. Unit

- Gunakan baik SI (MKS) atau CGS sebagai unit primer.
- Jangan menggabungkan kepanjangan dan singkatan dari unit, yang tepat seperti "Wb/m2" atau "webers per meter persegi," bukan "webers/m2."
- Gunakan angka nol di depan suatu bilangan desimal, seperti "0,25" bukan ",25."

C. Persamaan

Format persamaan merupakan suatu pengecualian di dalam spesifikasi *template* ini. Anda harus menentukan apakah akan menggunakan jenis tulisan Times New Roman atau Symbol (jangan jenis tulisan yang lain). Bila Anda membuat beberapa persamaan berbeda, akan lebih baik bila Anda mempersiapkan persamaan tersebut sebagai gambar dan menyisipkannya ke dalam artikel Anda setelah diberi *style*.

Beri penomoran untuk persamaan Anda secara berurutan. Nomor persamaan berada dalam tanda kurung seperti (1), dan diletakkan pada bagian kanan dengan menggunakan suatu *right tab stop*.

$$\int_0^{r_2} F(r,\phi)\, dr\, d\phi = [\sigma r_2 / (2\mu_0)] \qquad (1)$$

Perhatikan bahwa persamaan di atas diposisikan di bagian tengah dengan menggunakan suatu *center tab stop*. Pastikan bahwa simbol-simbol yang digunakan dalam persamaan Anda didefinisikan sebelum atau sesudah persamaan. Gunakan "(1)," bukan "Persamaan (1)," kecuali pada awal sebuah kalimat, seperti "Persamaan (1) merupakan …."

D. Beberapa Kesalahan Umum

- Perhatikan tata cara penulisan Bahasa Indonesia yang benar, perhatikan penggunaan kata depan dan kata sambung yang tepat, seperti "di depan" dan "disampaikan".
- Kata-kata asing yang belum diserap ke dalam Bahasa Indonesia dapat dicetak miring, atau diberi garis bawah, atau dicetak tebal (pilih salah satu), seperti "*italic*", "underlined", "**bold**".
- Prefiks seperti "non", "sub", "micro", "multi", dan "ultra" bukan kata yang berdiri sendiri, oleh karenanya harus digabung dengan kata yang mengikutinya, biasanya tanpa tanda hubung, seperti "subsistem".

IV. MENGGUNAKAN TEMPLATE

Setelah naskah artikel Anda selesai di-*edit*, artikel Anda dapat dipersiapkan untuk *template*. Gandakan template ini dengan menggunakan perintah Save As dan simpan dengan penamaan berikut:

- ULTIMATICS_namaPenulis1_judulArtikel.
- ULTIMAInfoSys_namaPenulis1_judulArtikel.
- ULTIMAComputing_namaPenulis1_judulArtikel.

Selanjutnya Anda dapat meng-*import* artikel Anda dan mempersiapkannya sesuai *template* yang diberikan. Perhatikan beberapa hal berikut pada saat melakukan pengecekan.

A. Penulis dan Afiliasi

Template ini didesain untuk tiga penulis dengan dua afiliasi yang berbeda. Penamaan afiliasi yang sama tidak perlu berulang, cukup afiliasi yang berbeda yang ditambahkan. Berikan alamat surel resmi afiliasi atau penulis jika diinginkan.

B. Penamaan Judul Bab dan Subbab

Bab merupakan suatu perangkat organisatorial yang memandu pembaca untuk membaca isi artikel Anda. Terdapat dua jenis bab: bab utama (bab) dan subbab.

Bab utama mengidentifikasikan komponen-komponen yang berbeda dalam artikel Anda dan tidak memiliki hubungan isi yang erat satu sama lainnya. Sebagai contoh PENDAHULUAN, DAFTAR PUSTAKA, dan UCAPAN TERIMA KASIH. Penulisan judul bab utama menggunakan huruf kapital dan penomoran angka Romawi.

Subbab merupakan isi yang dijabarkan lebih terstruktur dan memiliki relasi yang kuat. Penamaan subbab ditulis dengan menggunakan cara penulisan judul kalimat utama (*Capitalize Each Word*) dan penomorannya menggunakan huruf alfabet kapital secara berurutan. Untuk subsubbab, penamaan dan penomoran mengikuti cara penamaan dan penomoran subbab diikuti angka Arab, seperti "A.1 Penulis", "A.1.1 Afiliasi Penulis".

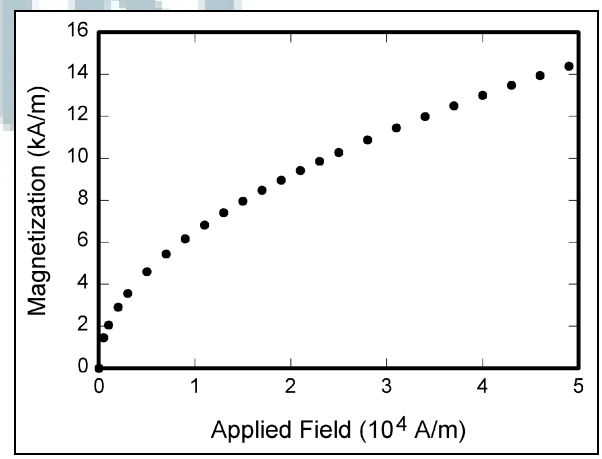
C. Gambar dan Tabel

Letakkan gambar dan tabel di atas atau di bawah kolom. Hindari posisi di tengah kolom. Gambar dan tabel yang besar dapat mengambil area dua kolom menjadi satu kolom. Judul gambar harus diletakkan di bawah gambar, sedangkan judul tabel harus diletakkan di atas tabel. Masukkan gambar dan tabel setelah mereka dirujuk di dalam isi artikel.

Tabel 1. Contoh tabel

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy		

Penamaan judul gambar dan tabel menggunakan cara penulisan kalimat biasa (*Sentence case*). Berikan jarak baris sebelum dan sesudah gambar atau tabel dengan kalimat penyertanya.



Gambar 1. Contoh gambar

V. SIMPULAN

Bagian simpulan bukan merupakan keharusan. Meskipun suatu simpulan dapat memberikan gambaran mengenai intisari artikel Anda, jangan menduplikasi abstrak sebagai simpulan Anda. Sebuah simpulan dapat menekankan pada pentingnya penelitian yang Anda lakukan atau saran pengembangan penelitian selanjutnya yang dapat dikerjakan.

LAMPIRAN

Jika diperlukan, Anda dapat menyisipkan lampiran-lampiran yang digunakan dalam artikel Anda sebelum UCAPAN TERIMA KASIH.

UCAPAN TERIMA KASIH

Di bagian ini Anda dapat memberikan pernyataan atau ungkapan terima kasih pada pihak-pihak yang telah membantu Anda dalam pelaksanaan penelitian yang Anda lakukan.

DAFTAR PUSTAKA

Untuk penamaan daftar pustaka, gunakan tanda kurung siku, seperti [1], secara berurutan dari awal rujukan dilakukan. Untuk merujuknya dalam kalimat, cukup gunakan [2], bukan "Rujukan [3]", kecuali di awal sebuah kalimat, seperti "Rujukan [3] menggambarkan …."

Penomoran catatan kaki dilakukan secara terpisah dengan *superscripts*. Letakkan catatan kaki tersebut di bawah kolom dimana catatan kaki tersebut dirujuk. Jangan letakkan catatan kaki di dalam daftar pustaka.

Kecuali terdapat enam atau lebih penulis, jabarkan nama penulis tersebut satu-satu, jangan gunakan "dkk". Artikel yang belum diterbitkan, meskipun sudah dikirim untuk diterbitkan, harus ditulis "belum terbit" [4]. Artikel yang sudah dikonfirmasi untuk diterbitkan, namun belum terbit, harus ditulis "proses cetak" [5]. Gunakan cara penulisan kalimat (*Sentence case*) untuk penulisan judul artikel.

Untuk artikel yang diterbitkan dalam jurnal terjemahan, tuliskan terlebih dahulu rujukan hasil terjemahannya, diikuti dengan jurnal aslinya [6].

[1] G. Eason, B. Noble, dan I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, hal. 529-551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, hal.68-73.

[3] I.S. Jacobs dan C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, hal. 271-350.

[4] K. Elissa, "Title of paper if known," belum terbit.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., proses cetak.

[6] Y. Yorozu, M. Hirano, K. Oka, dan Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, hal. 740-741, Agustus 1987 [Digests 9th Annual Conf. Magnetics Japan, hal. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.


UMN

ISSN 2085-4552

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA
PRESS

Universitas Multimedia Nusantara
Scientia Garden Jl. Boulevard Gading Serpong, Tangerang
Telp. (021) 5422 0808 | Fax. (021) 5422 0800