

# ULTIMATICS

Jurnal Teknik Informatika

AGUSTINUS RENDI WALEWOWAN, WILLY SUDIARTO RAHARJO,  
GLORIA VIRGINIA

Pembangunan Dashboard Beasiswa dan Pinjaman Program Studi  
Informatika UKDW

NATANAEL BENEDIKTUS, RAYMOND SUNARDI OETAMA

Algoritma Klasifikasi Decision Tree C5.0 untuk Memprediksi Performa  
Akademik Siswa

FELIX INDRA KURNIADI, VINNIA KEMALA PUTRI

Gray Level Co-ocurrence Matrix untuk Pengekstrasian  
Ciri Topeng Cirebon

ALETHEA SURYADIBRATA, JULIO CHRISTIAN YOUNG

Visualisasi Algoritma sebagai Sarana Pembelajaran K-Means Clustering

IS MARDIANTO

Pengenalan Wajah Bebas Ekspresi Menggunakan Metode Nearest  
Feature Line dengan Representasi Ciri dalam Ruang Eigen

STEVEN SEN, DEDY SUGIARTO, ABDUL ROCHMAN

Komparasi Metode Multilayer Perceptron (MLP) dan Long Short  
Term Memory (LSTM) dalam Peramalan Harga Beras

MUHAMMAD TAUFIQ, DEDY SUGIARTO, ABDUL ROCHMAN

Peramalan Utilisasi Perangkat Jaringan dan Bandwidth dengan Metode  
Holt-Winters dan Multilayer Perceptron

FENINAADLINE TWINCE TOBING, JAMES RONALD TAMBUNAN

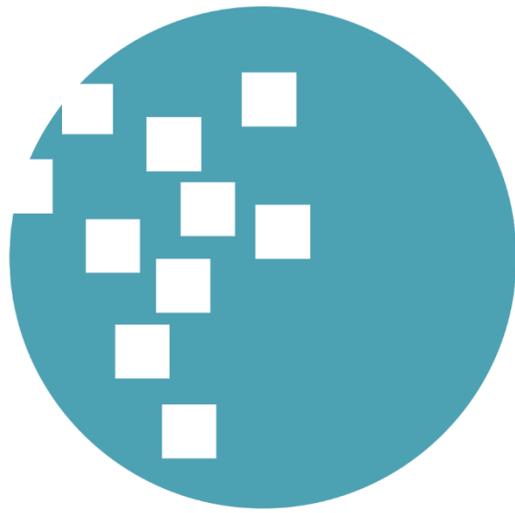
Analisis Perbandingan Efisiensi Algoritma Brute Force dan  
Divide and Conquer dalam Proses Pengurutan Angka

RINA PRAMITASARI, IMAM ROFIKI

Kombinasi Backpropagation dan Hopfield Modifikasi  
untuk Persamaan Polynomial

ANDRE SETIAWAN, FARICA PERDANA PUTRI

Implementasi Algoritma Apriori untuk  
Rekomendasi Kombinasi Produk Penjualan



**UMN**

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## SUSUNAN REDAKSI

### Pelindung

Dr. Ninok Leksono

### Penanggungjawab

Dr. Ir. P.M. Winarno, M.Kom.

### Pemimpin Umum

Nunik Afriliana, S.Kom., M.M.S.I.

### Mitra Bestari

(UMN) Dr. Ranga Winantyo, Ph.D.

(Universitas Indonesia) Filbert Hilman Juwono,  
S.T., M.T.

(Tanri Abeng University) Nur Afny Catur  
Andryani, M.Sc.

(UMN) Ir. Andrey Andoko, M.Sc.

(UMN) Adhi Kusnadi, S.T., M.Si.

(UMN) Alethea Suryadibrata, S.Kom, M.Eng.

(UMN) Arya Wicaksana, S.Kom., M.Eng.Sc.,  
OCA, CEH

(UMN) Dennis Gunawan S.Kom., M.Sc., CEH,  
CEI, CND

(UMN) Farica Perdana Putri, S.Kom., M.Sc.

(UMN) Marcel Bonar Kristanda, S.Kom., M.Sc.

(UMN) Seng Hansun, S.Si., M.Cs.

(UMN) Yustinus Widya Wiratama, S.Kom., M.Sc.,  
OCA

(UMN) Julio Christian Young, S.Kom.

### Ketua Dewan Redaksi

Ni Made Satvika Iswari, S.T., M.T.

### Dewan Redaksi

Wella, S.Kom., M.S.I., COBIT5

M.B.Nugraha, S.T., M.T.

Andre Rusli, S.Kom., M.Sc.

### Desainer dan Layouter

M.B.Nugraha, S.T., M.T.

Dyah Ayu Anggreini Tuasikal, S.T., M.T.

### Sirkulasi dan Distribusi

Sularmin

### Keuangan

I Made Gede Suteja, S.E.

## ALAMAT REDAKSI

Universitas Multimedia Nusantara (UMN)

Jl. Scientia Boulevard

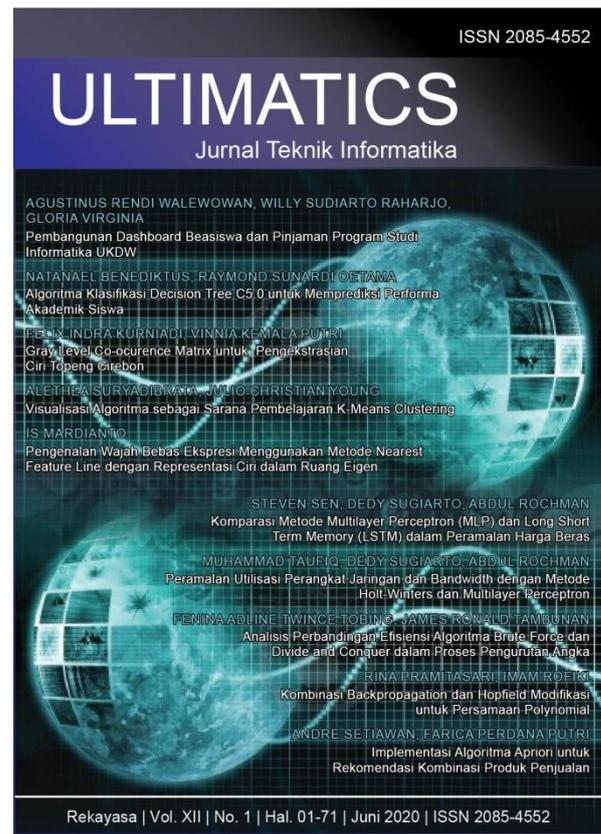
Gading Serpong

Tangerang, Banten - 15811

Telp. (021) 5422 0808

Faks. (021) 5422 0800

Surel. 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 Papers



**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. 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, and robotics. IJNMT is published annually by Information and Communication Technology Faculty of Universitas Multimedia Nusantara in cooperation with UMN Press.



**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 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.



**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.

# DAFTAR ISI

<b>Pembangunan Dashboard Beasiswa dan Pinjaman Program Studi Informatika UKDW</b> Agustinus Rendi Walewowan, Willy Sudiarto Raharjo, Gloria Virginia	01-13
<b>Algoritma Klasifikasi <i>Decision Tree</i> C5.0 untuk Memprediksi Performa Akademik Siswa</b> Natanael Benediktus, Raymond Sunardi Oetama	14-19
<b><i>Gray Level Co-ocurrence Matrix</i> untuk Pengekstrasian Ciri Topeng Cirebon</b> Felix Indra Kurniadi, Vinnia Kemala Putri	20-24
<b>Visualisasi Algoritma sebagai Sarana Pembelajaran <i>K-Means Clustering</i></b> Alethea Suryadibrata, Julio Christian Young	25-29
<b>Pengenalan Wajah Bebas Ekspresi Menggunakan Metode <i>Nearest Feature Line</i> dengan Representasi Ciri dalam Ruang Eigen</b> Is Mardianto	30-34
<b>Komparasi Metode <i>Multilayer Perceptron</i> (MLP) dan <i>Long Short Term Memory</i> (LSTM) dalam Peramalan Harga Beras</b> Steven Sen, Dedy Sugiarto, Abdul Rochman	35-41
<b>Peramalan Utilisasi Perangkat Jaringan dan <i>Bandwidth</i> dengan Metode <i>Holt-Winters</i> dan <i>Multilayer Perceptron</i></b> Muhammad Taufiq, Dedy Sugiarto, Abdul Rochman	42-51
<b>Analisis Perbandingan Efisiensi Algoritma <i>Brute Force</i> dan <i>Divide and Conquer</i> dalam Proses Pengurutan Angka</b> Fenina Adline Twince Tobing, James Ronald Tambunan	52-58
<b>Kombinasi <i>Backpropagation</i> dan Hopfield Modifikasi untuk Persamaan <i>Polynomial</i></b> Rina Pramitasari, Imam Rofiki	59-65
<b>Implementasi Algoritma Apriori untuk Rekomendasi Kombinasi Produk Penjualan</b> Andre Setiawan, Farica Perdana Putri	66-71

## KATA PENGANTAR

Salam ULTIMA!

ULTIMATICS – Jurnal Teknik Informatika UMN kembali menjumpai para pembaca dalam terbitan saat ini Edisi Juni 2020, Volume XII, 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 2020 ini, terdapat sepuluh artikel ilmiah yang berasal dari para peneliti, akademisi, dan praktisi di bidang Teknik Informatika, yang mengangkat beragam topik, antara lain: Implementasi Algoritma Apriori untuk Rekomendasi Kombinasi Produk Penjualan, Kombinasi *Backpropagation* dan Hopfield Modifikasi untuk Persamaan *Polynomial*, Analisis Perbandingan Efisiensi Algoritma *Brute Force* dan *Divide and Conquer* dalam Proses Pengurutan Angka, Peramalan Utilisasi Perangkat Jaringan dan *Bandwidth* dengan Metode *Holt-Winters* dan *Multilayer Perceptron*, Komparasi Metode *Multilayer Perceptron* (MLP) dan *Long Short Term Memory* (LSTM) dalam Peramalan Harga Beras, Pengenalan Wajah Bebas Ekspresi Menggunakan Metode *Nearest Feature Line* dengan Representasi Ciri dalam Ruang Eigen, Visualisasi Algoritma sebagai Sarana Pembelajaran *K-Means Clustering*, *Gray Level Co-ocurrence Matrix* untuk Pengekstrasian Ciri Topeng Cirebon, Algoritma Klasifikasi *Decision Tree* C5.0 untuk Memprediksi Performa Akademik Siswa, dan Pembangunan Dashboard Beasiswa dan Pinjaman Program Studi Informatika UKDW.

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](mailto:ultimatics@umn.ac.id).

Akhir kata, kami mengucapkan terima kasih kepada seluruh kontributor dalam ULTIMATICS Edisi Juni 2020 ini. Kami berharap artikel-artikel ilmiah hasil penelitian dalam jurnal ini dapat bermanfaat dan memberikan sumbangsih terhadap perkembangan penelitian dan keilmuan di Indonesia.

Juni 2020,

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

# Pembangunan *Dashboard* Beasiswa dan Pinjaman Program Studi Informatika UKDW

Agustinus Rendi Walewowan<sup>1</sup>, Willy Sudiarto Raharjo<sup>2</sup>, Gloria Virginia<sup>3</sup>  
<sup>1,2,3</sup>Departemen Informatika, Universitas Kristen Duta Wacana, Yogyakarta, Indonesia  
 agustinus.rendi@ti.ukdw.ac.id  
 willysr@ti.ukdw.ac.id  
 virginia@staff.ukdw.ac.id

Diterima 10 Januari 2020

Disetujui 17 Juni 2020

**Abstract**—The head of the study program (kaprodi) has many tasks and responsibilities that must be done in academic activities. One of the tasks of the head of the study program is a report on the entire implementation of activities in a study program. For that, we need a system that can be used for daily operational activities and provide reports. A *dashboard* is an information panel display that is used in an organization to consider a problem making it easier for someone to make a decision. This study made a *dashboard* scholarship and loan design using the prototyping method. Prototyping is a software development method, which consists of a physical work system model and an initial version of the system. The results of the average task success testing in both iterations, namely 96.66 so that the system built can be said to be quite effective in displaying scholarship and loan information and easy to learn. Interface design evaluations were carried out using the System Usability Scale (SUS) for 5 respondents in each iteration and resulted in a SUS score of 75.2 in iteration I and 76.6 in iteration II. Based on these results, the average SUS score for both iterations is 75.9. Thus, the system interface is stated to be good with a grade scale of C value, adjective rating of Good value, and acceptability ranges are acceptable.

**Index Terms**—Dashboard, Prototyping, System Usability Scale, Task Success

## I. PENDAHULUAN

Fakultas Teknologi Informasi Universitas Kristen Duta Wacana (UKDW) Yogyakarta merupakan salah satu fakultas favorit sehingga memiliki jumlah mahasiswa yang banyak. Dengan jumlah mahasiswa yang banyak, data yang disimpan juga semakin banyak sehingga menghasilkan informasi yang beragam, diantaranya adalah informasi mengenai peminjaman dan beasiswa.

Ketua program studi (kaprodi) memiliki peranan penting di sebuah program studi (prodi). Dalam kegiatan akademik kaprodi memiliki banyak tugas dan tanggung jawab yang harus dilaksanakan. Salah satu tugas dari seorang kaprodi adalah melaporkan seluruh pelaksanaan kegiatan di suatu prodi. Untuk itu, diperlukan suatu sistem yang dapat digunakan untuk memantau kegiatan operasional sehari-hari dan

memberikan laporan, khususnya kegiatan mahasiswa mengenai pinjaman dan penerimaan beasiswa dalam lingkungan program studi (prodi) informatika, sekaligus membantu proses pengambilan keputusan strategis.

*Dashboard* adalah sebuah tampilan panel informasi yang digunakan dalam suatu organisasi untuk mengevaluasi suatu masalah sehingga memudahkan seseorang untuk mengambil keputusan. *Tactical dashboard* adalah sebuah tampilan *dashboard* yang menggunakan data pada kondisi sebelumnya untuk mengidentifikasi tren sehingga dapat membantu seseorang untuk menganalisis dan membuat keputusan kedepannya.

Penelitian ini bertujuan melakukan perancangan sebuah *dashboard* beasiswa dan pinjaman dengan menggunakan metode *prototyping*. *Prototyping* adalah metode pengembangan perangkat lunak, yang berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem. [1]. Dengan adanya *dashboard* tersebut, diharapkan dapat membantu kaprodi dalam menganalisis aktivitas peminjaman dan beasiswa mahasiswa prodi informatika.

## II. LANDASAN TEORI

### A. Dashboard

Menurut Few dalam [2], *Dashboard* dapat diartikan sebagai *information dashboard* yaitu tampilan visual dari informasi penting yang diperlukan untuk mencapai satu atau beberapa tujuan dengan mengkonsolidasikan dan mengatur informasi dalam satu layar (*single screen*), sehingga kinerja organisasi dapat dimonitor secara sekilas.

Ada beberapa macam tipe *dashboard*, menurut Rasmussen, Bansal, dan Chen dalam [3], yaitu:

1. *Strategic dashboard* digunakan untuk mendukung manajemen level strategis memberikan informasi dalam membuat keputusan bisnis, memprediksi peluang, dan memberikan arahan pencapaian tujuan strategis.

2. *Analytical dashboard* berfokus pada proses analisis untuk menentukan penyebab dari suatu kondisi atau kejadian tertentu.
3. *Operational dashboard* yang berfungsi sebagai pendukung monitoring dari aktifitas proses bisnis yang spesifik. Fokus pada monitoring aktifitas dan kejadian yang tidak berubah secara konstan.

#### B. Tactical Dashboard

*Tactical dashboard* merupakan *dashboard* yang disiapkan untuk tujuan yang lebih rinci, biasanya digunakan untuk melacak tren dan dalam kaitannya dengan tujuan dan inisiatif perusahaan [4].

#### C. Prototyping

*Prototyping* merupakan metode pengembangan perangkat lunak, yang berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem. Metode *prototyping* akan menghasilkan *prototype* sistem sebagai perantara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan pengembangan sistem informasi [1].

##### 1. Requirement gathering

Tahap ini *customer* dan *developer* bersama-sama menentukan tujuan seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibangun.

##### 2. Quick design

Tahap ini berfokus pada representasi semua aspek perangkat lunak yang terlihat oleh user seperti perancangan antarmuka dalam membentuk *mock up* atau desain tampilan sistem disesuaikan dengan kebutuhan user.

##### 3. Building of the prototype

Tahap ini dibuat *prototype* yang mewakili sistem yang akan dibangun.

##### 4. Evaluation and refinement requirement

Tahap ini evaluasi ini dilakukan oleh *customer* apakah *prototype* yang dibangun sudah sesuai dengan keinginan *customer*.

#### D. The Advantages of Prototyping

*Prototyping method* memiliki beberapa kelebihan [5], yaitu :

1. Komunikasi yang baik antara *developer* dan *customer*.
2. *Developer* dapat bekerja lebih baik dalam menentukan kebutuhan *customer*.
3. *Customer* berperan aktif dalam pengembangan sistem.

4. Aplikasi menjadi lebih mudah karena *user* tahu apa yang diharapkan.

#### E. Usability Testing

*Usability* adalah atribut kualitas yang menilai seberapa mudah antarmuka pengguna digunakan. Kata "*usability*" juga merujuk pada metode untuk meningkatkan kemudahan penggunaan selama proses desain [6]. *Usability* didefinisikan menjadi 5 komponen, yaitu:

- *Learnability*: seberapa mudah bagi pengguna untuk menyelesaikan tugas-tugas dasar saat pertama kali mereka menghadapi desain?
- *Efficiency*: seberapa cepat mereka dapat melakukan tugas setelah pengguna mempelajari desain?
- *Memorability*: Ketika pengguna kembali ke desain setelah periode tidak menggunakannya, seberapa mudah mereka dapat membangun kembali kemahiran?
- *Errors*: Berapa banyak kesalahan yang dilakukan pengguna, seberapa parah kesalahan ini, dan seberapa mudah mereka pulih dari kesalahan?
- *Satisfaction*: Seberapa menyenangkan menggunakan desain?

#### F. Hukum Gestalt

Prinsip atau hukum Gestalt adalah aturan yang menggambarkan bagaimana mata manusia memandang elemen visual. Prinsip-prinsip ini bertujuan untuk menunjukkan bagaimana adegan kompleks dapat direduksi menjadi bentuk yang lebih sederhana. Hal tersebut bertujuan untuk menjelaskan bagaimana mata memandang bentuk sebagai satu kesatuan, bentuk daripada elemen-elemen sederhana yang terlibat [7].

Kecenderungan pengorganisasian tata letak secara keseluruhan ada pada prinsip-prinsip gestalt tentang pengelompokan (*Gestalt Principles of Grouping*) yang berpijak pada kecenderungan manusia untuk mengorganisasi stimuli yang terpisah menjadi pengelompokan yang berdasar pada *proximity* (kedekatan), *similarity* (kemiripan), *closure* (ketertutupan), *continuity* (kesinambungan), dan *symmetry* (simetris) [8].

1. *Proximity* (kedekatan), bentuk-bentuk sederhana yang disusun bersama dapat menciptakan gambar yang lebih kompleks.
2. *Similarity* (kemiripan), mata manusia cenderung membangun hubungan antara elemen-elemen serupa dalam suatu desain. Kesamaan dapat dicapai dengan

menggunakan elemen-elemen dasar seperti bentuk, warna, dan ukuran.

3. *Closure* (ketertutupan) mata manusia cenderung melihat suatu bentuk secara utuh atau melihatnya secara keseluruhan.
4. *Continuity* (kesinambungan) mata manusia mengikuti jalur, garis, dan kurva desain, dan lebih suka melihat aliran elemen visual yang berkelanjutan daripada objek yang terpisah.
5. *Symmetry* (simetris) desain yang dibangun harus seimbang dan lengkap. Jika tidak, pengguna akan menghabiskan waktu dan upaya mencoba memahami gambar keseluruhan.

### G. Reporting

*Reporting* adalah salah satu fungsi manajemen berupa penyampaian perkembangan atau hasil kegiatan mengenai segala hal yang berkaitan dengan tugas dan fungsi-fungsi kepada pejabat yang lebih tinggi. [9].

*Reporting* sering digunakan oleh suatu organisasi untuk membantu dalam menganalisis tren dan perkembangan sehingga dapat menentukan keputusan yang strategis.

### H. Success Rate

*Success rate* adalah persentase dari task yang berhasil dikerjakan oleh pengguna. *Success rate* digunakan untuk mengetahui tingkat kemudahan pengguna dalam menyelesaikan tugas yang diberikan [10].

Berikut rumus yang digunakan untuk menghitung *success rate*:

$$\left[ \frac{\text{Success Task} + (\text{Partial Success} \times 0,5)}{\text{Total Task}} \times 100\% \right] \quad (1)$$

Penilaian yang diberikan berdasarkan aturan yaitu, setiap *task* yang berhasil akan diberi nilai 1 dan berhasil sebagian diberi nilai 0,5, sedangkan jika *task* tersebut gagal akan diberi nilai 0.

### I. System Usability Scale

*System Usability Scale* (SUS) adalah metode pengujian yang menyediakan alat ukur yang “*quick and dirty*”. SUS terdiri dari 10 kuesioner dengan lima pilihan respons untuk responden, dari “sangat tidak setuju” hingga “sangat setuju” [11].

### J. Key Performance Indicator (KPI)

Menurut Banerjee dan Buoti, [12]. *Key Performance Indicator* merupakan ukuran berskala dan kuantitatif yang digunakan untuk mengevaluasi kinerja organisasi dalam tujuan mencapai target organisasi. KPI juga digunakan untuk menentukan objektif yang terukur, melihat tren, dan mendukung pengambilan keputusan

Dalam lingkungan perguruan tinggi, KPI sangat berhubungan erat dengan proses bisnis organisasi, dan standar yang ditetapkan Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT). Salah satu contoh KPI adalah profil mahasiswa, dengan menguraikan KPI menjadi beberapa jenis informasi, yaitu jumlah calon mahasiswa baru (ikut seleksi), jumlah mahasiswa baru, dan total mahasiswa [13].

### K. Tabel Fakta (Fact Tables)

*Fact Table* merupakan tabel yang mengacu pada setiap performa yang dapat diukur (*measure*). Tabel fakta merupakan *primary key* dari masing-masing tabel dimensi yang di kumpul menjadi *foreign key* dalam satu tabel. Hampir setiap tabel fakta biasanya berisi nilai *numeric*. Namun, tidak semua nilai *numeric* termasuk dalam tabel fakta. Misalnya, ukuran luas fasilitas. Ukuran luas fasilitas tidak termasuk tabel fakta karena merupakan sebuah deskripsi statis yang harus disimpan dalam dimensi fasilitas [14].

### L. Tabel Dimensi (Dimension Tables)

*Dimension Table* adalah tabel dalam dimensional model dengan sebuah *primary key* dan kolom atribut-atribut yang deskriptif. Atribut pada tabel dimensi memiliki dua tujuan utama yaitu *filtering* dan *labelling*. Tabel dimensi berisi konteks tekstual yang terkait dengan kegiatan pengukuran proses bisnis, menjelaskan "siapa, apa, dimana, kapan, bagaimana, dan mengapa" berkaitan dengan kegiatan tersebut [14].

## III. PERANCANGAN SISTEM

### A. Key Performance Indicator (KPI)

Tabel 1. *Key Performance Indicator* (KPI)

No.	Nama	Fungsi	Satuan
1.	Persentase mahasiswa yang melunasi / membayar pinjaman Biro 3.	Melihat persentase mahasiswa yang telah melunasi pinjaman di biro 3 setiap semester	Jumlah mahasiswa
2.	Persentase mahasiswa yang pinjam ke biro 3 di setiap angkatan.	Melihat persentase mahasiswa yang melakukan peminjaman ke biro 3	Jumlah mahasiswa
3.	Persentase mahasiswa prodi informatika yang mendapatkan beasiswa (Internal UKDW / Diluar UKDW).	Melihat persentase mahasiswa yang mendapatkan beasiswa internal maupun diluar UKDW.	Jumlah mahasiswa
4.	Rata-rata jumlah mahasiswa di setiap angkatan yang mendapatkan beasiswa.	Menghitung rata - rata jumlah mahasiswa penerima beasiswa setiap angkatan.	Jumlah mahasiswa
5.	Rata-rata jumlah dana beasiswa di setiap angkatan.	Menghitung rata - rata jumlah dana beasiswa	Rupiah

	setiap angkatan.	
--	------------------	--

Tabel 1 merupakan daftar *Key Performance Indicator* (KPI) beasiswa dan pinjaman yang akan digunakan dalam sistem. KPI diperlukan agar pengguna dapat mengetahui porforma mengenai data pinjaman dan beasiswa berada dalam keadaan buruk, normal, atau baik.

### B. Perancangan Unit

Perancangan sistem ini memiliki beberapa fungsi utama sebagai berikut:

#### 1. Pencarian Data

User dapat melakukan pencarian terhadap suatu data. Pencarian data yang dimaksud adalah pencarian mengenai data mahasiswa seperti nama, angkatan, dan nim.

#### 2. Detail Informasi

User dapat melihat detail informasi mengenai data yang ditampilkan dalam bentuk chart.

#### 3. Reporting

Fungsi ini menampilkan data secara keseluruhan dalam bentuk tabel sehingga mudah diolah oleh user.

### C. Data Schema

Schema data yang digunakan dalam penelitian ini menagacu pada *data warehouse* prodi informatika.

Tabel 2. Struktur tabel *dim\_basiswa*

Nama Kolom	Tipe Data
id_basiswa	<i>big integer</i>
id_wilayah	<i>big integer</i>
nama_basiswa	<i>char</i>
jumlah_basiswa	<i>decimal</i>

Tabel 3. Struktur tabel *dim\_wilayah*

Nama Kolom	Tipe Data
id_wilayah	<i>big integer</i>
level_wilayah	<i>char</i>

Tabel 2 merupakan tabel *dim\_basiswa* yang akan menampung data penerima beasiswa. Sedangkan tabel 3 merupakan tabel *dim\_wilayah* yang akan menampung data asal beasiswa berdasarkan level wilayah. Level wilayah dibagi menjadi 4 yaitu internal, regional, nasional, dan internasional.

Tabel 4. Struktur tabel *dim\_mahasiswa*

Nama Kolom	Tipe Data
id_mahasiswa	<i>big integer</i>
id_prodi	<i>big integer</i>
nim	<i>char</i>

tahun_angkatan	<i>char</i>
nama	<i>char</i>
jenis_kelamin	<i>char</i>
tanggal_lahir	<i>date</i>
tempat_lahir	<i>char</i>
agama	<i>char</i>
id_kodepos_mahasiswa	<i>char</i>
alamat	<i>char</i>
slta_smk	<i>char</i>
jurusan_slta_smk	<i>char</i>
id_kodepos_slta_smk	<i>char</i>
id_tanggal_do	<i>int</i>
id_tanggal_undur_diri	<i>int</i>
id_tanggal_registrasi	<i>int</i>
status	<i>char</i>
ipk_terakhir	<i>float</i>

Tabel 4 merupakan tabel *dim\_mahasiswa* yang akan menampung semua data tentang mahasiswa.

Tabel 5. Struktur tabel *dim\_jenis\_pembayaran*

Nama Kolom	Tipe Data
id_jenis_pembayaran	<i>big integer</i>
id_basiswa	<i>big integer</i>
jumlah_pembayaran_mandiri	<i>decimal</i>
jumlah_pembayaran_basiswa	<i>decimal</i>
jumlah_pembayaran_pinjaman	<i>decimal</i>

Tabel 5 merupakan tabel *dim\_jenis\_pembayaran* yang akan menampung data tentang pembayaran. Data pembayaran yang dimaksud adalah data pembayaran mandiri, beasiswa, dan pinjaman.

Tabel 6. Struktur tabel *dim\_date*

Nama Kolom	Tipe Data
id_date	<i>big integer</i>
tanggal	<i>date</i>
hari_dalam_tahun	<i>int</i>
hari_dalam_bulan	<i>int</i>
hari_dalam_minggu	<i>int</i>
nama_hari	<i>char</i>
bulan	<i>int</i>
nama_bulan	<i>char</i>
kuartal	<i>int</i>
tahun	<i>char</i>
tahun_ajaran	<i>char</i>
semester	<i>int</i>
semester_yudisium	<i>int</i>

Tabel 6 merupakan tabel *dim\_date* yang akan menampung semua yang berhubungan dengan tanggal, tahun, bulan, dan hari.

Tabel 7. Struktur tabel *fact\_pembayaran* registrasi

Nama Kolom	Tipe Data
id_pembayaran_registrasi	<i>big integer</i>
id_mahasiswa	<i>big integer</i>
id_date	<i>big integer</i>
id_jenis_pembayaran	<i>big integer</i>
jumlah	<i>decimal</i>

Tabel 7 merupakan tabel *fact\_pembayaran\_registrasi* yang akan menampung semua *primary key* dari masing-masing tabel dimensi sehingga menjadi *foreign key*.

Tabel 8. Struktur tabel *dim\_prodi*

Nama Kolom	Tipe Data
id_prodi	<i>big integer</i>
kode_prodi	<i>varchar</i>
kode_fakultas	<i>varchar</i>
nama_fakultas	<i>varchar</i>
nama_prodi	<i>varchar</i>
nama_universitas	<i>varchar</i>

Tabel 8 merupakan tabel *dim\_prodi* yang menampung semua data mengenai program studi dan fakultas.

Tabel 9. Struktur tabel *dim\_kodepos*

Nama Kolom	Tipe Data
id_kodepos	<i>big integer</i>
kodepos	<i>varchar</i>
kabupaten_kota	<i>varchar</i>
provinsi	<i>varchar</i>
provinsi_en	<i>varchar</i>

Tabel 9 merupakan tabel *dim\_kodepos* yang menampung data mengenai kode pos, kabupaten dan provinsi. kedua tabel tersebut akan membantu menampung data yang diperlukan oleh tabel mahasiswa.

Tabel 10. Struktur tabel *fact\_transaksi\_lunas*

Nama Kolom	Tipe Data
id_transaksi_lunas	<i>big integer</i>
id_pembayaran_registrasi	<i>big integer</i>
id_date	<i>big integer</i>
jumlah_pembayaran_cicilan	<i>decimal</i>

Tabel 10 merupakan tabel *fact\_transaksi\_lunas* yang akan menampung *primary key* dari tabel *fact\_pembayaran\_registrasi* dan *dim\_date*. *ID\_pembayaran\_registrasi* pada tabel ini digunakan sebagai penampung ID yang sudah melakukan pembayaran pinjaman, sedangkan *id\_date* digunakan untuk menampung tanggal ID tersebut membayar pinjaman.

## IV. IMPLEMENTASI DAN ANALISIS

### A. Implementasi Dashboard

#### A.1 Halaman Login

Gambar 1. Halaman *login*

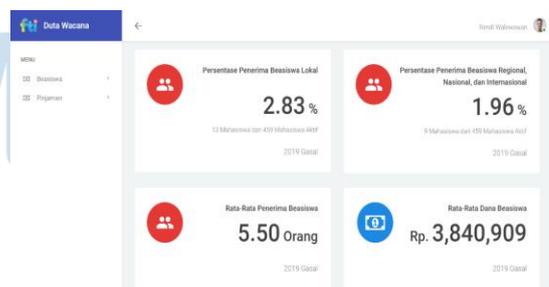
Gambar 1 merupakan tampilan halaman *login* dari *dashboard* yang telah dibangun. Halaman *login* adalah halaman awal yang ditampilkan pada saat membuka sistem. Agar dapat masuk ke dalam sistem user harus mengisi email dan *password* yang telah terdaftar sebagai bagian dari proses autentikasi.

#### A.2 Halaman Utama *Dashboard* Mahasiswa



Gambar 2. Indikator dan filter tahun periode

Gambar 2 merupakan tampilan indikator dan tahun periode pada halaman utama *dashboard*.

Gambar 3. Tampilan utama *dashboard* beasiswa

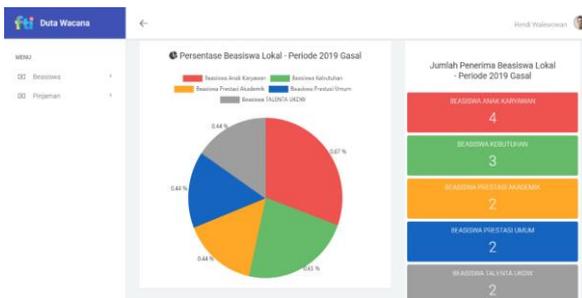
Gambar 3 merupakan tampilan utama dari *dashboard*. Setelah *user* melakukan autentikasi, *user* akan diarahkan ke halaman ini. Halaman utama *dashboard* akan menampilkan informasi sesuai dengan periode waktu yang dipilih. Selain itu, halaman ini terdapat *icon* berwarna yang menggambarkan kondisi sesuai dengan periode waktu yang dipilih. Warna - warna tersebut sesuai dengan warna indikator yang dapat dilihat pada Gambar 2.

A.3 Halaman Beasiswa Lokal



Gambar 4. KPI persentase penerima beasiswa lokal

Gambar 4 merupakan tampilan dari KPI persentase penerima beasiswa lokal. Ketika user melakukan click pada KPI tersebut, maka user akan diarahkan ke halaman beasiswa lokal seperti pada Gambar 5.



Gambar 5. Halaman beasiswa lokal

Halaman beasiswa lokal menampilkan persentase beasiswa dalam bentuk *pie chart* serta jumlah penerima beasiswa yang dikategorikan berdasarkan nama beasiswa.

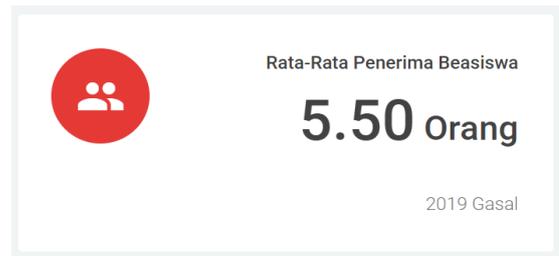
A.4 Halaman Beasiswa Regional, Nasional, Internasional



Gambar 6. KPI persentase beasiswa regional, nasional, internasional

Gambar 6 merupakan tampilan dari KPI persentase penerima beasiswa regional, nasional dan internasional. Ketika *user* melakukan *click* pada KPI tersebut, maka *user* akan diarahkan ke halaman beasiswa regional, nasional dan internasional seperti pada Gambar 7.

A.5 Halaman Rata-rata Penerima Beasiswa

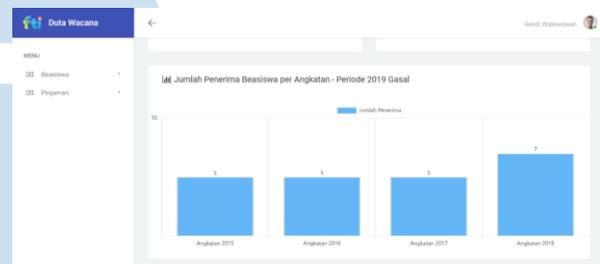


Gambar 7. KPI rata-rata penerima beasiswa

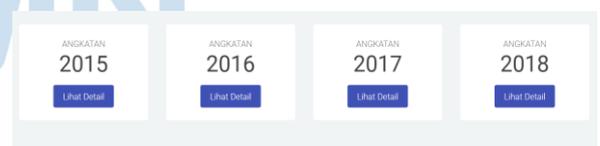
Gambar 8 merupakan tampilan dari KPI rata-rata penerima beasiswa pada periode waktu tertentu.



Gambar 8. Total penerima dan total dana beasiswa



Gambar 9. *Bar chart* jumlah penerima beasiswa



Gambar 10. *Card* tahun angkatan

NIM	Nama	Angkatan	Nama Beasiswa	Tanggal	Jumlah
2118090	TUHANELL A LAMAM	2015	Beasiswa Bidik Misi	20/08/2019	Rp 7.200.000
2118114	FENNY LEO	2015	Beasiswa Anak Karyawan	30/08/2019	Rp 3.800.000
2118103	TITUS RANGGA WIS PRABUKAWA	2015	Beasiswa Prestasi Umum	20/08/2019	Rp 500.000
2118123	KUSHETRA ALERY PRANATA	2015	Beasiswa Prestasi Akademik	20/08/2019	Rp 1.000.000
2118125	NOVA PERAYONG DAVY TOONG	2015	Beasiswa Kibulatihan	30/08/2019	Rp 1.000.000

Gambar 11. Halaman daftar penerima beasiswa per angkatan

Gambar 9 sampai 12 merupakan tampilan halaman rata-rata penerima beasiswa. Halaman ini menampilkan total penerima dan total dana beasiswa pada periode waktu tertentu seperti pada Gambar 9. Gambar 10 menampilkan jumlah penerima beasiswa per angkatan dalam bentuk *bar chart*.

Gambar 11 merupakan *card* yang menampilkan tahun angkatan sesuai dengan yang ditampilkan pada *bar chart*. Pada *card* tersebut terdapat tombol “Lihat Detail” yang akan mengarah ke halaman daftar penerima beasiswa per angkatan seperti pada Gambar 12.

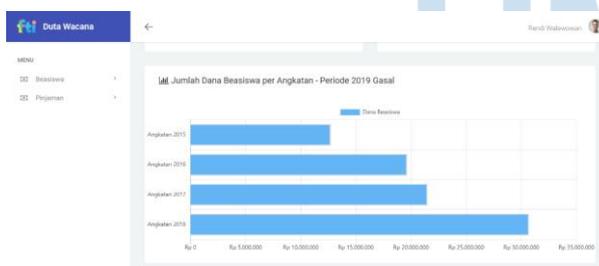
#### A.6 Halaman Rata-rata Dana Beasiswa



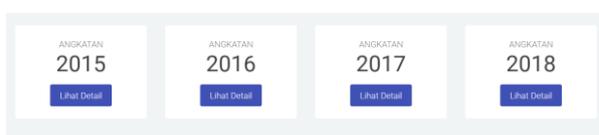
Gambar 12. KPI rata-rata dana beasiswa



Gambar 13. Total penerima dan total dana beasiswa



Gambar 14. *Horizontal bar chart* jumlah dana beasiswa



Gambar 15. *Card* tahun angkatan

NIM	Nama	Angkatan	Nama Beasiswa	Tanggal	Jumlah
71150001	THEMA L LAMAR	2015	Beasiswa Bidik Misi	30-08-2015	Rp 7.300.000
71150114	FERRY LEO	2015	Beasiswa Anak Karawana	30-08-2015	Rp 2.800.000
71150103	TITUS BANGSA WIDYARANDHAKO	2015	Beasiswa Prestasi Umum	30-08-2015	Rp 900.000
71150109	RIKMETIYA ALYAH PANGAKA	2015	Beasiswa Prestasi Akademik	30-08-2015	Rp 1.000.000
71150105	WINDA WIDYANINGRAT	2015	Beasiswa Prestasi Akademik	30-08-2015	Rp 1.000.000
71150105	NINA BERNARDI SANTI TOSKI	2015	Beasiswa Kehidupan	30-08-2015	Rp 1.000.000

Gambar 16. Halaman daftar penerima beasiswa per angkatan

Gambar 13 sampai 17 merupakan tampilan halaman rata-rata dana beasiswa. Halaman ini tidak jauh berbeda dengan halaman rata-rata penerima beasiswa yang tetap menampilkan total penerima dan total dana beasiswa pada periode waktu tertentu seperti pada Gambar 14. Perbedaannya terletak hanya pada pemilihan chart seperti pada Gambar 15 yang menampilkan jumlah dana beasiswa per angkatan dalam bentuk *horizontal bar chart*. Selain itu, Gambar 16 merupakan *card* yang menampilkan tahun angkatan sesuai dengan yang ditampilkan pada *horizontal bar chart* yang akan mengarah ke halaman daftar penerima beasiswa per angkatan seperti pada Gambar 17.

#### A.7 Halaman Reporting Beasiswa



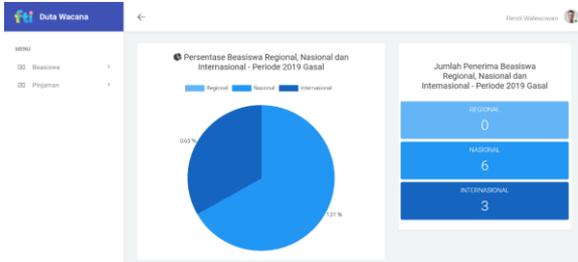
Gambar 17. Filter tahun periode dan rekap data

NIM	Nama	Angkatan	Beasiswa	Wilayah	Tanggal	Semester	Tahun Ajaran	Jumlah
71150003	NATHANIEL CLARENCE HARVANTO	2015	Beasiswa Scranton	Internasional	27-08-2015	GASAL	2015/2016	Rp 3.420.000
71150004	REXY DWY BRUNDY	2015	Beasiswa Prestasi Akademik	Lokal	27-08-2015	GASAL	2015/2016	Rp 1.000.000
71150005	MICHELL BERNARDI SUSANTO	2015	Beasiswa Bidik Misi	Nasional	27-08-2015	GASAL	2015/2016	Rp 7.360.000

Gambar 18. Tabel *reporting* beasiswa

Halaman *reporting* menyediakan dua *filter* tahun periode yang memungkinkan *user* untuk melihat data dalam *range* waktu tertentu sesuai dengan keinginan *user* dan rekap data secara keseluruhan seperti pada gambar 18. Selain itu, terdapat tabel *reporting* yang menampilkan data penerima beasiswa yang terdiri dari

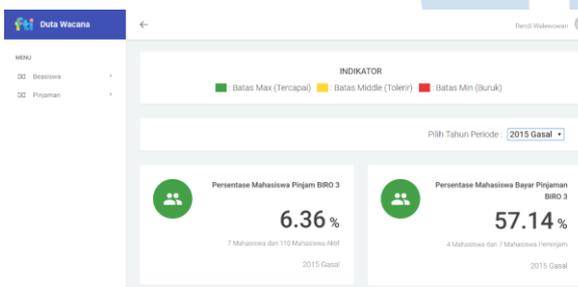
nim, nama, angkatan, nama beasiswa, wilayah, tanggal terima beasiswa, semester, tahun ajaran dan jumlah beasiswa yang diterima seperti yang terlihat pada Gambar 19. Pada halaman *reporting* terdapat beberapa fitur yang dapat membantu *user* dalam mengolah data seperti *search*, *export* data ke dalam bentuk *excel*, *pdf*, *print* dan *copy*.



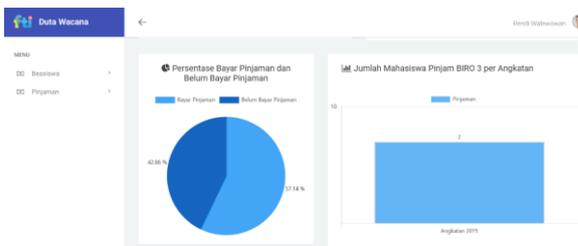
Gambar 19. Halaman beasiswa regional, nasional, internasional

Halaman beasiswa regional, nasional dan internasional menampilkan persentase beasiswa dalam bentuk *pie chart* serta jumlah penerima beasiswa yang dikategorikan berdasarkan wilayah beasiswa.

A.8 Halaman Utama *Dashboard* Pinjaman



Gambar 20. Tampilan utama *dashboard* pinjaman



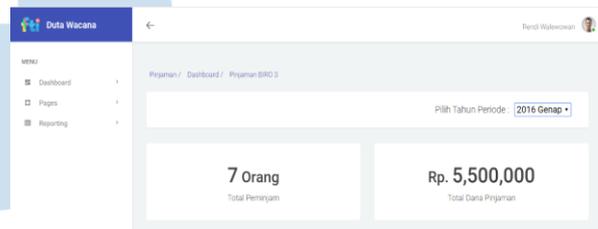
Gambar 21. *Pie chart* dan *bar chart* halaman utama *dashboard* pinjaman



Gambar 22. *Line chart* halaman utama *dashboard* pinjaman

Gambar 20 sampai Gambar 22 merupakan tampilan utama dari *dashboard* pinjaman. Halaman utama *dashboard* pinjaman akan menampilkan informasi sesuai dengan periode waktu yang dipilih dan tren pinjaman per tahun dalam bentuk *line chart*. Selain itu, halaman ini terdapat *icon* berwarna yang menggambarkan kondisi sesuai dengan periode waktu yang dipilih. Gambar 21 menampilkan persentase bayar pinjaman dan belum bayar pinjaman dalam bentuk *pie chart* serta jumlah mahasiswa pinjam per angkatan dalam bentuk *bar chart*.

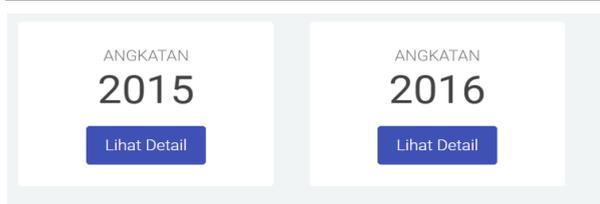
A.9 Halaman Pinjaman BIRO 3



Gambar 23. Filter tahun dan rekap data peminjam



Gambar 24. *Bar chart* halaman peminjam biro 3 per angkatan



Gambar 25. Card tahun angkatan

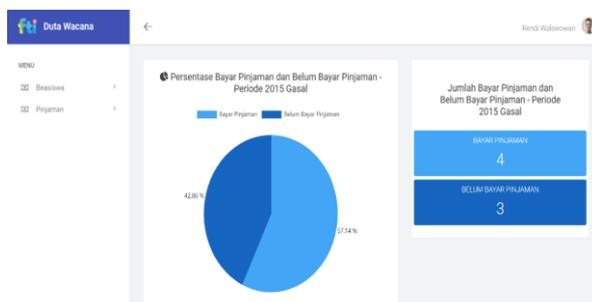
Gambar 23 sampai 25 merupakan tampilan halaman pinjaman biro 3. Halaman ini menampilkan informasi total peminjam dan total dana pinjaman pada periode waktu tertentu seperti pada Gambar 22. Halaman ini menampilkan jumlah peminjam per angkatan dalam bentuk *bar chart*. Gambar 23 sampai 25 merupakan tampilan halaman pinjaman biro 3. Halaman ini menampilkan informasi total peminjam dan total dana pinjaman pada periode waktu tertentu seperti pada Gambar 22. Halaman ini menampilkan jumlah peminjam per angkatan dalam bentuk *bar chart*.

NIM	Nama	Angkatan	Program Studi	Tanggal	Jumlah
7115000	MICHEL BERNARDO SUSANTO	2015	Informatika	30-08-2014	Rp 500.000
7115019	ANDRIS ALWANTO	2015	Informatika	29-08-2014	Rp 1.000.000

Gambar 26. Halaman daftar pinjaman biro 3 per angkatan

Gambar 25 merupakan *card* yang menampilkan tahun angkatan sesuai dengan yang ditampilkan pada *chart* yang akan mengarah ke halaman daftar peminjam per angkatan seperti pada Gambar 26.

#### A.10 Halaman Pembayaran Pinjaman



Gambar 27. Halaman pembayaran pinjaman

Gambar 27 merupakan tampilan halaman pembayaran pinjaman. Halaman ini menampilkan informasi jumlah mahasiswa yang telah membayar dan belum membayar pinjaman pada periode waktu tertentu.

User akan diarahkan ke halaman tabel mahasiswa yang belum membayar pinjaman dan yang sudah membayar pinjaman seperti pada Gambar 28 dan Gambar 29 jika melakukan *click* pada *card* jumlah bayar pinjaman dan belum bayar pinjaman pada Gambar 27.

NIM	Nama	Angkatan	Program Studi	Tanggal Bayar	Jumlah Cicilan
7115015	WOLFHARDUS ALEXANDER RABU	2015	Informatika	31-08-2015	Rp 200.000
7115017	PUTU PRIDANTA RAKA PUTRA P	2015	Informatika	31-08-2015	Rp 500.000

Gambar 28. Tabel bayar pinjaman

NIM	Nama	Angkatan	Program Studi	Tanggal Pinjam	Jumlah Pinjaman
7115019	ANDREAS ALYANTO	2015	Informatika	30-08-2015	Rp 500.000
7115020	TITUS RANGGA WEDI PAMUNGKAS	2015	Informatika	30-08-2015	Rp 1.000.000
7115021	KARENSIUS VICTOR	2015	Informatika	31-08-2015	Rp 1.000.000

Gambar 29. Tabel belum bayar pinjaman

#### A.11 Halaman Reporting Pinjaman



Gambar 30. Filter tahun periode dan rekap data pinjaman

NIM	Nama	Angkatan	Program Studi	Tanggal	Semester	Tahun Ajaran	Jumlah
71150115	WOLFHARDUS ALEXANDER RABU	2015	Informatika	28-08-2015	GASAL	2015/2016	Rp. 500.000
71150116	EMANUELI NIRMANI PUTERA WIRCHANA	2015	Informatika	28-08-2015	GASAL	2015/2016	Rp. 1.000.000
71150117	PUTU PRIDAVITA RAKA PUTIYA P	2015	Informatika	29-08-2015	GASAL	2015/2016	Rp. 1.000.000
71150118	ARISTARHUS HARISAN SIKOTANG	2015	Informatika	30-08-2015	GASAL	2015/2016	Rp. 500.000

Gambar 31. Tabel reporting pinjaman

Fitur pada halaman reporting pinjaman tidak berbeda dengan halaman reporting beasiswa. Perbedaan pada halaman ini hanya terletak pada isi row tabel yang terdiri dari nim, nama, angkatan, program studi, tanggal pinjam, semester, tahun ajaran dan jumlah pinjaman seperti yang terlihat pada Gambar 31.

B. Evaluasi Sistem Iterasi I

Tabel 11. Hasil pengujian sistem iterasi I

	Ta sk 1	Ta sk 2	Ta sk 3	Ta sk 4	Ta sk 5	Ta sk 6	Ta sk 7	Ta sk 8	Ta sk 9	Task Success
R1	B	B	B	B	B	B	B	B	B	100 %
R2	B	B	B	B	B	G	B	B	B	89 %
R3	B	B	B	B	B	B	B	B	B	100 %
R4	B	B	B	B	B	B	B	B	B	100 %
R5	B	B	B	B	B	G	B	B	B	89 %
Tas k Succ ess	10 0 %	10 0 %	10 0 %	10 0 %	10 0 %	60 0 %	10 0 %	10 0 %	10 0 %	95,5 5 %

Tabel 11 merupakan hasil pengujian sistem kepada beberapa responden. Pengujian sistem ini dinilai dari berhasil atau tidaknya responden dalam melakukan task yang diberikan. Setiap task yang berhasil akan diberi nilai 1 dan berhasil sebagian diberi nilai 0,5, sedangkan jika task tersebut gagal akan diberi nilai 0.

Berdasarkan Tabel 11 terdapat 9 task untuk masing-masing responden, dari hasil yang diperoleh terdapat 1 task yang gagal dikerjakan oleh responden ke-2 dan 5, yaitu pada task 6 dengan nilai 89% untuk masing-masing responden dan nilai rata-rata task 6, yaitu 60%, hal ini dikarenakan responden sulit menemukan halaman dashboard pinjaman. Dengan demikian diperoleh hasil untuk success rate pada iterasi I adalah 95,55%.

Tabel 12. Hasil pengujian SUS iterasi I

	Skor Hasil Perhitungan										Jumlah Total 1	Nilai (Jumlah Total x 2,5)
	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10		
R 1	4	4	4	4	4	3	4	4	4	1	36	90
R 2	3	3	3	3	3	3	2	3	3	0	26	65
R 3	3	3	3	4	4	2	3	3	3	4	32	80
R 4											29	73
R 5	3	3	3	4	2	3	3	3	3	2	27	68
Rata - rata Nilai SUS												75,2

Skor dari masing-masing pertanyaan tersebut didapatkan dari perhitungan dimana hasil dari pertanyaan bernomor genap akan dikurangi dengan 5 dan hasil dari pertanyaan bernomor ganjil akan dikurangi 1.

$$\text{Pertanyaan Ganjil} = [\text{Hasil Pertanyaan Ganjil} - 1] \quad (2)$$

$$\text{Pertanyaan Genap} = [5 - \text{Hasil Pertanyaan Genap}] \quad (3)$$

Skor dari masing-masing pertanyaan setiap responden akan dijumlahkan secara keseluruhan lalu dikalikan dengan 2,5 yang merupakan nilai kali yang berasal dari aturan SUS.

$$\text{Nilai SUS} = \sum_{x=1}^{10} Q_x \times 2,5 \quad (4)$$

Berdasarkan hasil perhitungan, rata-rata nilai SUS pada tabel 12 yaitu 75,2.

C. Evaluasi Sistem Iterasi II

Tabel 13. Hasil pengujian sistem iterasi II

	Ta sk 1	Ta sk 2	Ta sk 3	Ta sk 4	Ta sk 5	Ta sk 6	Ta sk 7	Ta sk 8	Ta sk 9	Tas k Succ ess
R1	B	B	B	B	B	G	B	B	B	89 %
R2	B	B	B	B	B	B	B	B	B	100 %
R3	B	B	B	B	B	B	B	B	B	100 %
R4	B	B	B	B	B	B	B	B	B	100 %
R5	B	B	B	B	B	B	B	B	B	100 %
Tas k Succ ess	10 0 %	10 0 %	10 0 %	10 0 %	10 0 %	80 0 %	10 0 %	10 0 %	10 0 %	97,7 7 %

Berdasarkan Tabel 13 terdapat 9 task untuk masing-masing responden, dari hasil yang diperoleh terdapat 1 task yang gagal dikerjakan oleh responden ke-1, yaitu pada task 6 dengan nilai 89% dan nilai rata-rata task 6, yaitu 80%, hal ini dikarenakan responden ke-1 juga masih sulit menemukan halaman dashboard pinjaman. Namun, jika dilihat secara

keseluruhan ada peningkatan pada *task 6* sebesar 20% dari iterasi I dikarekan adanya perubahan pada tampilan *menu*. Dengan demikian diperoleh hasil untuk *success rate* pada iterasi II adalah 97,77%.

Tabel 14. Hasil pengujian SUS iterasi II

	Skor Hasil Perhitungan										Jumlah Total 1	Nilai (Jumlah Total x 2,5)
	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10		
R 1	3	3	3	1	3	3	3	3	3	1	26	65
R 2	4	3	3	4	2	3	2	4	1	4	30	75
R 3	4	3	3	4	3	3	3	3	4	3	33	83
R 4	4	4	4	3	3	3	4	4	3	4	36	90
R 5	3	3	3	3	3	3	3	3	3	1	28	70
Rata – rata Nilai SUS												76,6

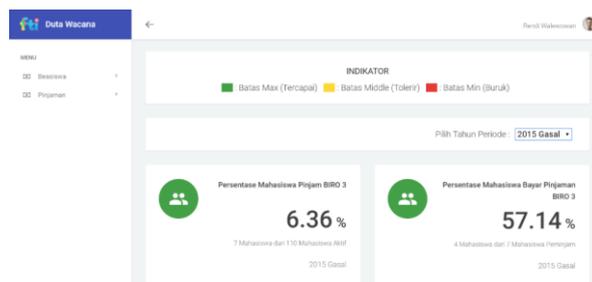
Berdasarkan hasil perhitungan rata-rata nilai SUS pada tabel 14 yaitu 76,6.

#### D. Analisis Antarmuka

Desain *dashboard* beasiswa dan pinjaman menerapkan beberapa prinsip desain dalam perancangannya. Berikut beberapa contoh penerapan prinsip desain dalam pembangunan *dashboard* beasiswa dan pinjaman.

##### D.1 Proximity

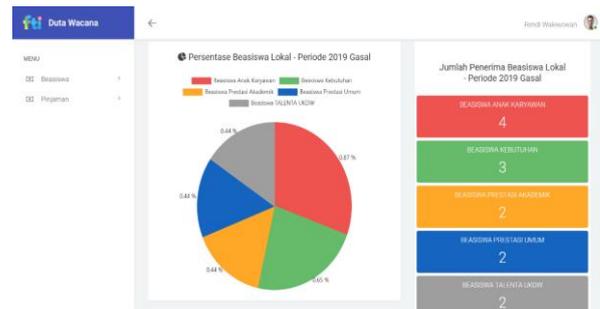
Dalam perancangan desain *dashboard* beasiswa maupun pinjaman prinsip *proximity* selalu digunakan dalam mengatur tata letak konten setiap halaman, baik halaman utama maupun halaman detail dari masing-masing KPI. Berdasarkan tampilan utama *dashboard* ini dapat dikelompokkan menjadi 5 elemen dasar, yaitu 1 tampilan menu pada sisi kiri, 1 tampilan konten indikator, 1 tampilan untuk filter berdasarkan tahun, 1 tampilan untuk masing-masing KPI, dan 1 tampilan untuk masing-masing *chart*. Berikut contoh penggunaannya:



Gambar 32. *Proximity* pada halaman utama *dashboard* pinjaman

Sedangkan untuk masing-masing halaman detail, dikelompokkan menjadi 4 elemen, yaitu 1 tampilan menu pada sisi kiri, 1 link navigasi pada sisi kiri atas,

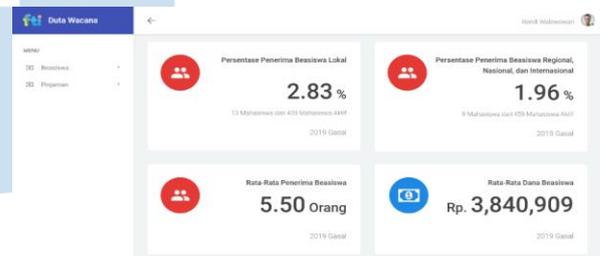
1 tampilan untuk filter berdasarkan tahun, 1 tampilan untuk masing-masing konten, dan 1 tampilan untuk masing-masing *chart*. Berikut contoh penggunaannya:



Gambar 33. *Proximity* pada halaman beasiswa lokal

##### D.2 Similarity

Tampilan utama *dashboard* beasiswa dan pinjaman menggunakan prinsip *similarity* pada konten KPI *dashboard* dengan kesamaan bentuk kotak (*card*) untuk setiap konten KPI-nya. Semua kotak (*card*) juga menampilkan perilaku yang sama ketika meletakkan mouse (*hover*) di atas *card*, hal ini menciptakan pengalaman yang memperkuat prinsip *similarity*. Berikut beberapa contoh penggunaannya:

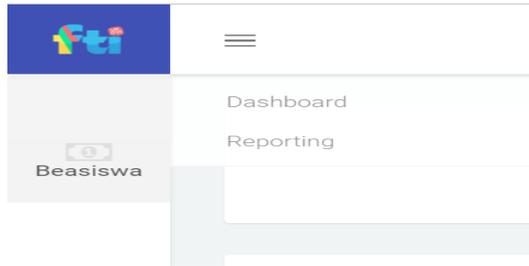


Gambar 34. *Similarity* halaman utama *dashboard* pinjaman

##### D.3 Common Fate

Prinsip *common fate* menyatakan bahwa mata manusia cenderung melihat unsur-unsur yang bergerak bersama-sama sebagai suatu kelompok yang lebih mirip satu sama lain daripada unsur yang diam atau bergerak ke arah yang berbeda [15].

Pada *dashboard* beasiswa dan pinjaman prinsip *common fate* berlaku pada tampilan menu pada sisi kiri halaman *dashboard*. Prinsip ini digunakan untuk menghubungkan antara *menu* dan *sub-menu*.



Gambar 35. Common fate pada menu beasiswa

D.4 Focal Point

Prinsip focal point menyatakan bahwa obyek dengan point of interest akan lebih cepat ditangkap mata dan lebih menarik perhatian karena adanya penekanan atau perbedaan dari unsur obyek lainnya [15].



Gambar 36. Focal point pada KPI beasiswa lokal

E. Analisis Pengujian SUS

Hasil rata-rata nilai SUS pada tabel 12 dan tabel 14 akan diukur menggunakan skala perhitungan SUS, hasil pada pengujian iterasi I menunjukkan SUS skor yang diperoleh adalah 75,2 dan pada iterasi II diperoleh hasil 76,6. Berdasarkan hasil tersebut maka diperoleh rata-rata skor pada kedua iterasi, yaitu 75,9. Dengan demikian, desain iterasi I dan II untuk penilaian acceptability ranges dapat diterima (acceptable). Sedangkan untuk penilaian grade scale hasil pengujian iterasi I dan II mendapat nilai C dan penilaian adjective ratings pengujian mendapatkan nilai Good. Dengan hasil tersebut, maka usability pada sistem ini dapat dikatakan cukup baik.

F. Analisis Pengujian Task Rate



Gambar 37. Perbandingan iterasi I dan II

Gambar 37 menunjukkan adanya perbandingan antara hasil iterasi I dan II. Hasil pada iterasi I menunjukkan pada angka 95,55% dan iterasi II mengalami peningkatan sebesar 2.22% sehingga diperoleh hasil 97,77%. Dengan demikian rata-rata task success dari kedua iterasi, yaitu 96, 66%.

Perbandingan antara iterasi I dan II terletak pada task skenario nomor 6, yaitu “Melihat persentase mahasiswa yang pinjam ke BIRO 3 pada periode 2016 Gasal”. Pada task tersebut responden gagal mengerjakan dikarenakan default halaman utama dari sistem yang dibangun ketika responden melakukan login, yaitu halaman dashboard beasiswa. Untuk menampilkan persentase mahasiswa pinjam, responden harus masuk ke halaman dashboard pinjaman terlebih dahulu.

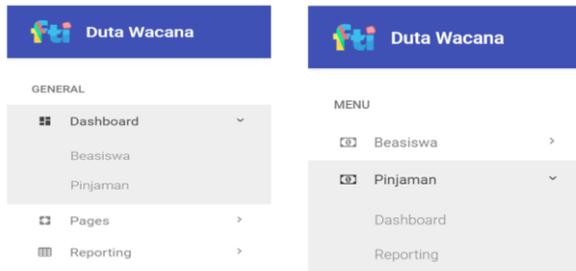


Gambar 38. Task success iterasi I



Gambar 39. Task success iterasi II

Desain *menu* pada iterasi I mengharuskan responden masuk melalui *menu dashboard* dan *sub-menu* pinjaman seperti pada gambar 40 (kiri). Sedangkan pada iterasi II tampilan *menu* dan *sub-menu* memisahkan antara beasiswa dan pinjaman seperti pada gambar 45 (kanan).



Gambar 40. Menu iterasi I (kiri) dan menu iterasi II (kanan)

Perubahan yang dilakukan pada desain menu tersebut memberikan dampak yang terlihat dari grafik *task success* iterasi II pada gambar 39. Pada grafik tersebut, *task* nomor 6 mengalami peningkatan sebesar 20% pada iterasi II yang sebelumnya pada iterasi I, pada gambar 4.50 bernilai 60% sedangkan pada iterasi kedua, *task* nomor 6 menunjukkan nilai 80%. Oleh karena itu pada implementasinya tampilan menu menggunakan tampilan pada iterasi II seperti pada gambar 40 (kanan).

## V. SIMPULAN

Berdasarkan implementasi, analisis dan pembahasan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

- Dengan menerapkan metode prototyping maka sistem berhasil dibangun dan memvisualisasikan informasi mengenai beasiswa dan pinjaman ke dalam bentuk *dashboard* sesuai dengan keinginan *user*.
- Hasil rata-rata pengujian *task success* pada kedua iterasi, yaitu 96,66% sehingga sistem yang dibangun dapat dikatakan cukup efektif dalam menampilkan informasi beasiswa dan pinjaman serta mudah untuk dipelajari.
- Evaluasi desain antarmuka yang dilakukan menggunakan *System Usability Scale* (SUS) menunjukkan skor SUS 75,2 pada iterasi I dan 76,6 pada iterasi II. Berdasarkan hasil tersebut diperoleh rata-rata skor SUS untuk kedua

iterasi, yaitu 75,9. Dengan demikian, antarmuka sistem dinyatakan cukup baik dengan *grade scale* bernilai C, *adjective rating* bernilai Good, dan *acceptability ranges* dapat diterima (*acceptable*).

## DAFTAR PUSTAKA

- P. M. Ogedebe and B. . P. Jacob, "Software Prototyping: A Strategy to Use When User Lacks Data Processing Experience," ARPN Journal of Systems and Software , pp. 219-224, 2012.
- R. K. Putra, "Apa itu Dashboard ?," 7 Januari 2018. [Online]. Available: <https://timur.ilearning.me/2018/01/07/apa-itu-dashboard/>.
- F. C. Saputro, W. Anggraeni and A. Mukhlason, "Pembuatan Dashboard Berbasis Web Sebagai Sarana Evaluasi Diri Berkala untuk Persiapan Penilaian Akreditasi Berdasarkan Standar Badan Akreditasi Nasional Perguruan Tinggi," JURNAL TEKNIK ITS, pp. 2301 - 9271, 2012.
- B. Dashboard, "Business Intelligence, Data visualization, KPI," BI Dashboards, [Online]. Available: <https://www.bidashboard.org/types/tactical.html>. [Accessed 30 May 2020].
- A. Susanto and M. , "System Development Method with The Prototype Method," INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH, vol. 8, no. 7, pp. 141-144, 2019.
- J. Nielsen, "Nielsen Norman Group," 3 Januari 2012. [Online]. Available: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
- M. Soegaard, "Interaction Design Foundation," 13 September 2019. [Online]. Available: <https://www.interaction-design.org/>.
- R. W. Putra and B. Muhdaliha, "Analisis Visual Game Arena Of Valor Sebagai Dasar Pengembangan Konsep Visual Game," Jurnal Avant Garde V, pp. 98-113, 2017.
- "Temukan Pengertian," 2014. [Online]. Available: <https://www.temukanpengertian.com/2014/03/pengertian-reporting.html>.
- M. Y. Alfique, I. Aknuranda and N. H. Wardani, "Evaluasi Usability Pada Aplikasi UBER Menggunakan Pengujian Usability," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, pp. 2599-2606, 2018.
- J. Sauro, "MEASURING USABILITY WITH THE SYSTEM USABILITY SCALE (SUS)," 2 February 2011. [Online]. Available: <https://measuringu.com/sus/>.
- B. Kho, "Pengertian KPI (Key Performance Indicator) atau Indikator Kinerja Utama," 24 Mei 2018. [Online]. Available: <https://ilmumanajemenindustri.com/pengertian-kpi-key-performance-indicators-indikator-kinerja-utama/>.
- H. and E. Winarko, "Rancangan Sistem Data Warehouse Evaluasi Kinerja," Seminar Nasional Teknologi Informasi dan Multimedia, pp. 13-18, 2015.
- R. Kimball, M. Ross, W. Thornthwaite, J. Mundy and B. Becker, The Data Warehouse Lifecycle Toolkit, Indianapolis: Wiley Publishing, Inc, 2008.
- R. W. Anggoro, 23 September 2016. [Online]. Available: <https://www.niagahoster.co.id/blog/website-profesional-gestalt/>.

# Algoritma Klasifikasi *Decision Tree* C5.0 untuk Memprediksi Performa Akademik Siswa

Natanael Benediktus<sup>1</sup>, Raymond Sunardi Oetama<sup>2</sup>

<sup>1,2</sup> Fakultas Teknik Informatika, Jurusan Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia

[natanael.benediktus@student.umn.ac.id](mailto:natanael.benediktus@student.umn.ac.id)

[raymond@umn.ac.id](mailto:raymond@umn.ac.id)

Diterima 5 Maret 2020

Disetujui 17 Juni 2020

**Abstract**—Student’s performance is often used as a benchmark and a student’s activeness is frequently used as a criteria of how well a student academically perform at school. The goal of this study is to find out whether the activeness of a student can predict their academic performance. The data used is an educational dataset is collected using a learning management system (LMS), which is a learner activity tracker tool that is connected by the internet. This data has numerical and categorical variables, so it is needed to have the right algorithm to classify data accurately and ensure data validity. In this study, the C.50 algorithm is used to test the data, where the data is divided into training data by 75% and testing data by 25%. And the result from the tested data, an accuracy of 71.667% is obtained.

**Index Terms**—C5.0 Algorithm, Data Classification, Decision Tree, Student’s Performance

## I. PENDAHULUAN

Performa seorang siswa seringkali disimpulkan dari keaktifannya di dalam ataupun di luar kelas. Siswa yang sering mengangkat tangan dalam kelas atau sering berpartisipasi dalam diskusi, atau yang inisiatif menambah ilmu diluar sekolah, atau bahkan yang jarang absen dalam kelas tentunya tidak jauh dengan label “anak pintar” [1]. Hal seperti ini biasanya hanya ditarik dari sebagian siswa saja yang mendapatkan nilai baik, namun tentunya terdapat sebagian siswa yang memiliki kriteria yang sama, namun memiliki nilai yang kurang memuaskan [2]. Maka dalam penelitian ini ingin dicari tahu apakah benar kriteria yang seringkali digunakan tersebut dapat memprediksi nilai seseorang atau mungkin terdapat kriteria lain yang lebih akurat, bahkan mungkin sebenarnya tidak ada kriteria yang dapat memprediksi nilai seseorang. Perhatian utama dalam penelitian ini adalah untuk mengekstraksi yang selama ini tersembunyi dari data siswa yang dapat meningkatkan performa akademik siswa, karena jika ditinjau banyak sekali aspek dalam dunia ini yang melihat nilai sebagai tolak ukur. Sebagai contoh adalah dalam melamar pekerjaan, sebuah perusahaan tentunya akan melihat nilai dari pelamar kerja sebagai salah satu tolak ukurnya.

Dari beberapa penelitian terdahulu, *Decision Tree* sering dipergunakan untuk prediksi prestasi siswa dengan performa yang sangat baik. Algoritma *Decision Tree* untuk memprediksi nilai akhir dari siswa dengan akurasi 85% [3]. Optimasi *Decision Tree* dapat diterapkan untuk prediksi siswa berpotensi bermasalah dengan tingkat akurasi 99,08% [4]. Untuk memprediksi prestasi siswa kelas XII dengan *Decision Tree*, diperoleh hasil akurasi yang diperoleh sangat baik yaitu 97,22% [5]. Sehingga dalam penelitian ini, akan menggunakan metode klasifikasi dengan algoritma *Decision Tree*. Sedangkan beberapa kriteria yang dipergunakan dalam prediksi prestasi siswa juga diambil dari beberapa penelitian terdahulu meliputi: siswa mengangkat tangan [6], berpartisipasi dalam diskusi [7], inisiatif siswa untuk belajar di luar sekolah [8], dan absensi siswa [9].

Berdasarkan latar belakang di atas, dapat ditarik beberapa permasalahan, yang pertama adalah seberapa akuratkah algoritma *Decision Tree* ini dengan faktor keaktifan siswa di dalam kelas maupun di luar kelas yang mencakup keseringan dalam mengangkat tangan, keseringan berpartisipasi dalam diskusi, insiatif siswa untuk belajar di luar sekolah, dan absensi dari siswa itu sendiri untuk memprediksi performa akademik siswa. Yang kedua adalah faktor apa yang paling berpengaruh terhadap performa akademik siswa. Dan yang terakhir adalah kriteria apa yang memperlihatkan performa akademik siswa yang paling baik. Manfaat dari penelitian ini diharapkan dapat memperlihatkan kriteria tertentu yang dapat meningkatkan performa akademik siswa.

## II. LANDASAN TEORI

### A. Algoritma C5.0

Algoritma yang bisa digunakan untuk pembuatan pohon keputusan diantaranya adalah *Decision Tree* [10]. Algoritma C5.0 atau yang lebih dikenal dengan nama *Decision Tree* merupakan salah satu algoritma dari klasifikasi. Algoritma C5.0 ini mengklasifikasikan datanya menjadi pohon bercabang yang memiliki beberapa *node* seperti *root node* yakni *node* di bagian *decision tree* yang paling atas dimana tidak memiliki

panah atau *edge* yang menunjuknya, internal *node* yakni *node* yang memiliki panah atau *edge* yang menunjuk dan juga keluar dari *node* tersebut, dan terminal *node* (*leaf*) yakni *node* di bagian ujung bawah *decision tree* dimana hanya ada panah atau *edge* yang menunjuknya tanpa panah atau *edge* yang keluar dari *node* tersebut [11].

### B. Sejarah Algoritma C5.0

Algoritma C5.0 ini merupakan perkembangan dari algoritma C4.5 yang juga merupakan perkembangan dari algoritma ID.3. Algoritma ID.3 (*Iterative Dichotomiser 3*) ini ditemukan oleh John Ross Quinlan sejak tahun 1986. Algoritma ini merupakan metode untuk membangun sebuah *decision tree* yang paling dasar yang melakukan pencarian secara menyeluruh pada semua kemungkinan *decision tree*. Algoritma ID.3 ini lalu dikembangkan menjadi algoritma C4.5 dimana algoritma ini mampu menangani atribut dengan tipe diskrit dan kontinu dimana algoritma ini menggunakan ukuran *entropy* untuk memilih atribut yang tepat. Algoritma C4.5 juga mampu menangani *missing value* dimana dengan memberikan nilai berdasarkan nilai yang paling dominan. Dan algoritma ini juga mampu mengatasi *decision tree* dimana digunakan untuk mengatasi masalah *over-fitting* yang terjadi karena adanya data yang tidak relevan di data training, hal ini disebabkan oleh banyaknya cabang di *decision tree* yang lalu dipangkas atau yang lebih sering dikenal dengan istilah *pruning* untuk memperkecil pohon yang akhirnya lebih mudah untuk dipahami. Algoritma C4.5 ini pun dikembangkan menjadi algoritma C5.0 dikarenakan masih terdapat berbagai kelemahan di algoritma C4.5 seperti terjadinya *overlapping* terutama saat data yang dikelola sangat banyak, yang juga menyebabkan meningkatnya waktu pengambilan keputusan. Dengan munculnya algoritma C5.0 ini, tingkat akurasi yang dimiliki lebih tinggi, lebih cepat untuk mengambil keputusan, dan penggunaan memori yang jauh lebih rendah dari algoritma sebelumnya [12].

### C. Kegunaan Algoritma C5.0

Algoritma C5.0 digunakan untuk mengklasifikasikan data yang menghasilkan model *decision* keputusan dan aturan-aturan keputusan. Aturan-aturan keputusan ini akan digunakan untuk membantu dalam pengambilan keputusan yang tepat. Algoritma C5.0 juga digunakan untuk mengekstraksi data dimana dapat ditemukan hubungan variabel tertentu dengan variabel target [13].

### D. Kelebihan dan Kelemahan Algoritma C5.0

Kelebihan dari algoritma C5.0 adalah kemampuannya untuk menangani masalah seperti *missing value* dan data dengan jumlah yang besar. Algoritma ini juga dapat melakukan *training* data dalam waktu yang cepat untuk digunakan dalam

*testing* data. Algoritma C5.0 ini juga menawarkan metode *boosting* yang dapat meningkatkan tingkat akurasi [14].

Kelemahan dari algoritma C5.0 ini sama seperti *decision tree* lainnya yakni model prediksi akan menjadi tidak stabil apabila varians data sangat kecil, dan ketergantungan akan informasi dimana perubahan kecil dalam inputan data dapat menyebabkan perubahan yang besar pada pohon [15]. Sedangkan *Pseudocode* C5.0 [16] dapat dilihat pada Gambar 1.

#### Pseudo Code:

1. Check for base cases.
2. For each attribute calculate the Normalized information gain for splitting an attribute.
3. Out of this select the best attribute which has the highest information gain.
4. Find a decision node that splits the best, as root node.
5. Recurs on the sub lists obtained by splitting on best of a and add those nodes as children node.

Gambar 1. *Pseudocode* C5.0

## III. METODOLOGI

### A. Objek Penelitian

Dataset yang digunakan dalam penelitian adalah data dari siswa yang mencakup 3 macam kategori dengan nilai akhir dari siswa. Kategori pertama yaitu fitur demografis seperti jenis kelamin dan kewarganegaraan. Kategori kedua yaitu fitur latar belakang akademik seperti tingkatan pendidikan, dan tingkat kelas dari siswa. Dan kategori ketiga yaitu fitur perilaku siswa seperti berapa kali seorang siswa mengangkat tangan dalam kelas, berapa kali siswa mengunjungi situs tertentu untuk belajar mandiri, berapa kali siswa berpartisipasi dalam diskusi, dan absensi dari siswa tersebut. Dalam penelitian ini, data yang akan digunakan hanyalah fitur perilaku siswa sebagai faktor untuk memprediksi nilai akhir, dimana faktor tersebutlah yang seringkali digunakan sebagai tolak ukur tingginya nilai seorang siswa.

### B. Pengumpulan Data

Bentuk dari data yang digunakan dalam penelitian ini berbentuk numerik untuk faktor banyaknya angkat tangan, banyaknya kunjungan siswa dalam situs tertentu untuk belajar mandiri, dan banyaknya partisipasi siswa untuk berdiskusi. Sedangkan, faktor absensi dalam bentuk kategorikal yang dibagi menjadi 2 kategori yaitu lebih dari 7 kali dan kurang dari 7 kali. Dan untuk data nilai akhir siswa dalam bentuk kategorikal juga dimana dibagi menjadi 3 kategori yaitu *high-level* dengan interval nilai 90 sampai dengan 100, *mid-level* dengan interval nilai 70 sampai dengan 89, dan *low-level* dengan interval nilai 0 sampai dengan 69.

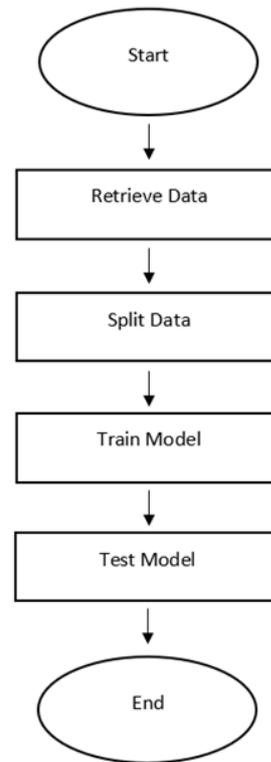
Teknik pengumpulan data yang digunakan adalah dengan teknik observasi dimana data dikumpulkan melalui *Learning Management System* (LMS) yang berfungsi sebagai alat pelacak aktivitas pelajar melalui koneksi internet. Data ini dikumpulkan pada tahun 2016 yang lalu oleh seorang profesor dari sebuah universitas yang bernama “*The University of Jordan*”, dimana data ini merupakan data dari siswa sebaran Jordan

(<https://www.kaggle.com/aljarah/xAPI-Edu-Data>).

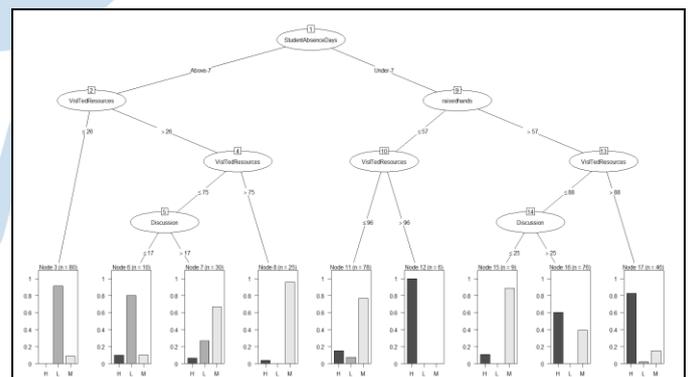
Data ini diyakini sebagai sebuah data yang valid dikarenakan penggunaan *Learning Management System* tersebut dimana observasi tidak hanya sebatas observasi seorang manusia, dimana hal ini tidak mencakup *human error* yang merupakan salah satu faktor akan data yang tidak valid.

Dalam proses *Split Data*, input yang diperlukan berupa dataset dalam RStudio yang merupakan output dari proses “*retrieve data*”. Data tersebut akan dibagi menjadi data *training* dan data *testing*, dimana data *training* akan digunakan untuk membangun model, dan data *testing* akan digunakan untuk tes validasi model. Namun, sebelum data dibagi menjadi data *training* dan data *testing*, syntax “*set.seed*” diperlukan agar nilai akan selalu tetap setiap membagi data.

Total data yang dipergunakan adalah 480. Dalam penelitian ini, data akan dibagi sebesar 75% yaitu 360 untuk data training dan sisanya 25% untuk data testing. Output yang dihasilkan dari proses ini berupa data training dan data testing. Data *training* akan menampung hasil dari temp diatas. Data *testing* akan menampung kebalikannya hasil temp diatas. Dalam proses *Train Model*, input yang diperlukan berupa data yang telah dibagi menjadi data training dan data testing yang merupakan output dari proses “*split data*”, dan juga *package C50* yang akan digunakan untuk membangun model *Decision Tree C5.0*. Output dari proses ini sendiri merupakan *decision tree* dan *decision rule* dari model data training.

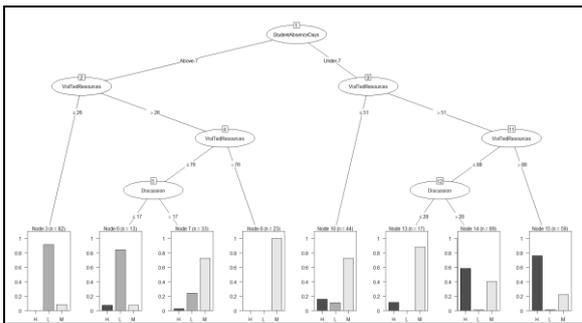


Gambar 2. Framework

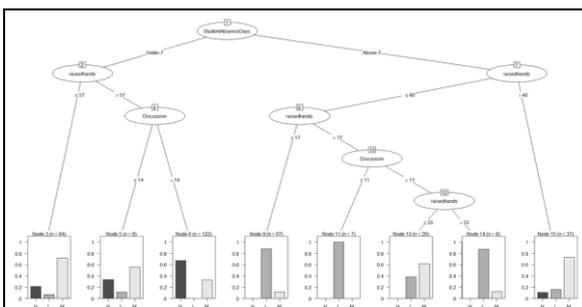


Gambar 3. Model C5.0 seluruh faktor

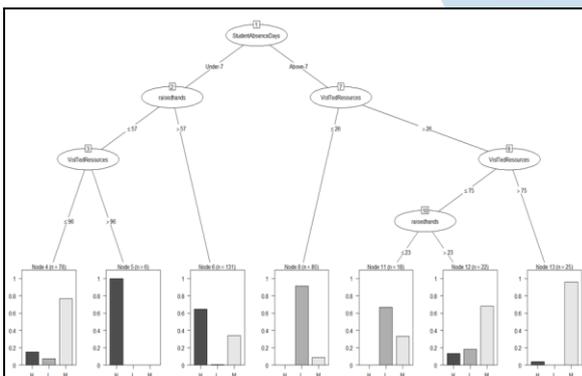
Untuk memvalidasi hasil, akan digunakan tes model dari data *training* yang merupakan output dari proses *train model*, dan data *testing* yang merupakan 25% dari keseluruhan data yang telah dibagi di proses *split* data sebelumnya. Output yang diterima berupa prediksi akan data testing, dimana hasil dari prediksi ini sendiri akan menghasilkan akurasi dari model algoritma C5.0 ini terhadap dataset tersebut. Metode validasi yang digunakan pada penelitian merupakan akurasi dari prediksi model terhadap data tes. Akurasi sendiri merupakan nilai yang mengacu pada tingkat ketepatan prediksi dengan keseluruhan data (Resika Arthana, 2019).



Gambar 4. Model C5.0 tanpa faktor mengangkat tangan



Gambar 5. Model C5.0 tanpa faktor belajar mandiri



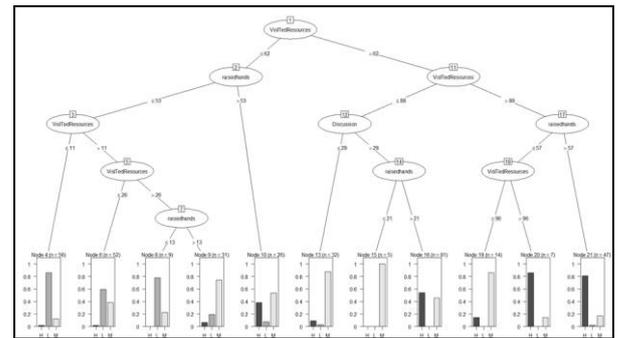
Gambar 6. Model C5.0 tanpa faktor diskusi

#### IV. HASIL DAN PEMBAHASAN

##### A. Model Decision Tree pada Semua Faktor

Pada Gambar 3, plot *decision tree* tersebut merupakan plot dari model dengan seluruh faktor. Absensi menjadi faktor yang paling berpengaruh dimana nilai seorang siswa cenderung lebih buruk saat absensinya kurang dari 7 kali, dan nilai seorang siswa cenderung lebih baik saat absensinya lebih dari 7 kali. Faktor selanjutnya yang berpengaruh adalah tingkat keseringan dalam belajar mandiri dan mengangkat tangan

##### B. Model Decision Tree Tanpa Faktor Angkat Tangan



Gambar 7. Model C5.0 tanpa faktor absensi

Seperti yang dapat dilihat pada gambar 4, plot *decision tree* tersebut merupakan plot dari model tanpa faktor angkat tangan. Absensi menjadi faktor yang paling berpengaruh dimana nilai seorang siswa cenderung lebih buruk saat absensinya kurang dari 7 kali, dan nilai seorang siswa cenderung lebih baik saat absensinya lebih dari 7 kali. Faktor selanjutnya yang menjadi faktor yang berpengaruh adalah tingkat keseringan dalam belajar mandiri.

##### C. Model Decision Tree Tanpa Faktor Belajar Mandiri

Seperti yang dapat dilihat pada Gambar 5, plot *decision tree* tersebut merupakan plot dari model tanpa faktor belajar mandiri. Absensi menjadi faktor yang paling berpengaruh dimana nilai seorang siswa cenderung lebih buruk saat absensinya kurang dari 7 kali, dan nilai seorang siswa cenderung lebih baik saat absensinya lebih dari 7 kali. Faktor selanjutnya yang menjadi faktor berpengaruh adalah tingkat keseringan dalam mengangkat tangan.

##### D. Model Decision Tree Tanpa Faktor Diskusi

Seperti yang dapat dilihat pada gambar 6, plot *decision tree* tersebut merupakan plot dari model tanpa faktor diskusi. Absensi menjadi faktor yang paling berpengaruh dimana nilai seorang siswa cenderung lebih buruk saat absensinya kurang dari 7 kali, dan nilai seorang siswa cenderung lebih baik saat absensinya lebih dari 7 kali. Faktor selanjutnya yang menjadi faktor yang berpengaruh adalah tingkat keseringan dalam belajar mandiri dan mengangkat tangan.

##### E. Model Decision Tree Tanpa Faktor Absensi

Seperti yang dapat dilihat pada gambar 7, plot *decision tree* tersebut merupakan plot dari model tanpa faktor absensi. Faktor yang paling berpengaruh adalah tingkat keseringan dalam belajar mandiri dimana nilai seorang siswa cenderung lebih buruk saat keseringannya dalam belajar mandiri kurang dari

sama dengan 62 kali, dan nilai seorang siswa cenderung lebih baik saat keseringannya dalam belajar mandiri lebih dari 62 kali.

Tabel 1. Validasi hasil

Evaluation Measures	C5.0				
	All Factor	Without Raised Hands	Without Visited Resource	Without Discussion	Without Absence
Accuracy	71.67%	74.17%	66.67%	66.67%	57.5%

#### F. Validasi Hasil

Sebagaimana terlihat pada Tabel 1, akurasi yang terdata merupakan hasil dari *confusion matrix* yang telah dibuat sebelumnya yakni hasil dari proses tes data atas 5 kali percobaan dengan campuran faktor yang berbeda-beda pada algoritma C5.0.

#### G. Diskusi

Menurut hasil tes dari model yang telah dibuat dari data *training*, akurasi yang dicapai oleh algoritma C5.0 ini mencapai 71.667% untuk prediksi performa akademik siswa dari faktor keaktifan siswa di dalam kelas maupun di luar kelas yang mencakup tingkat keseringan dalam mengangkat tangan, keseringan berpartisipasi dalam diskusi, inisiatif siswa untuk belajar di luar sekolah, dan absensi dari siswa itu sendiri pada data testing.

Hasil *modelling* data *training* ini juga memperlihatkan faktor-faktor yang berpengaruh terhadap performa akademik siswa, sebenarnya seluruh faktor sangatlah berpengaruh karena apabila dilihat dari *decision tree* yang telah dilampirkan, tiap kriteria memiliki dampak yang sangat besar terhadap hasilnya. Namun, apabila dilihat secara keseluruhan absensi seorang siswa dan inisiatif siswa untuk belajar mandiri adalah yang paling berpengaruh terhadap performa akademik mereka, sedangkan partisipasi untuk mengikuti diskusi ataupun tingkat keseringan dalam mengangkat tangan tidak seberpengaruh kedua faktor tersebut.

Dari hasil ini, diperlihatkan juga kriteria terbaik untuk memperoleh performa akademik yang sangat memuaskan, dimana seorang siswa perlu untuk selalu datang ke sekolah dan jarang absen, dimana menurut hasil penelitian ini absensi harus kurang dari 7 kali. Lalu siswa tersebut pun juga harus sering mengangkat tangan dimana mungkin siswa yang jarang untuk mengangkat tangan tentunya memendam rasa keingintahuannya yang malah menjadi tidak mengerti akan pelajarannya, dari hasil penelitian ini menganjurkan untuk mengangkat tangan setidaknya lebih dari 57 kali dalam setahunnya. Lalu, setelah jarang absen dan sering mengangkat tangan, siswa

tersebut pun harus sering belajar mandiri, mungkin di situs tertentu, dimana dianjurkan untuk belajar mandiri setidaknya lebih dari 88 kali dalam setahunnya.

## V. SIMPULAN

### A. Kesimpulan

Kesimpulan yang dapat ditarik dari penelitian ini adalah penggunaan algoritma *Decision Tree* C5.0 ini cukuplah efektif untuk memprediksi performa akademik siswa dimana tercapai akurasi sebesar 71.667%.

Kesimpulan lainnya yang dapat ditarik dari *decision tree* yang telah ditunjukkan adalah seorang siswa memang diperlukan untuk aktif dan selalu datang ke sekolah agar tidak tertinggal pelajaran untuk mendapatkan nilai yang memuaskan. Akan tetapi, memang ada saja siswa yang tidak aktif namun mempunyai nilai yang memuaskan ataupun sebaliknya yang sangat aktif namun mempunyai nilai yang kurang memuaskan. Namun, penelitian ini telah membuktikan bahwa sebagian besar yang aktif memang dapat mendapatkan nilai yang memuaskan.

### B. Saran

Untuk peneliti selanjutnya yang ingin atau sedang mengklasifikasikan data bisa menggunakan algoritma *Decision Tree* C5.0 ini, karena hasil yang didapatkan cukup mudah untuk dimengerti terutama untuk para kaum awam.

### C. Limitasi

Limitasi dari penelitian ini adalah tidak adanya perbandingan dengan algoritma lain, dimana tentunya akan menghasilkan informasi yang lebih memuaskan dengan algoritma yang berbeda-beda, dimana dapat memberikan kriteria yang berbeda pula.

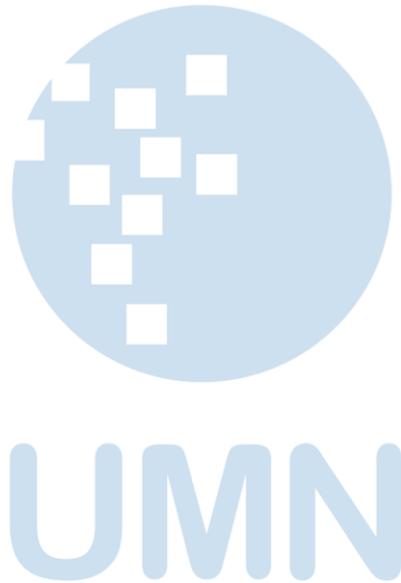
## UCAPAN TERIMA KASIH

Penelitian ini mendapatkan pendanaan dari Universitas Multimedia Nusantara, Tangerang, Indonesia.

## DAFTAR PUSTAKA

- [1] E. D. Putra, dan I. R. Panglipur, "Analisis Level Kinerja Practitioner Melalui Aktivitas Belajar Siswa", *Jurnal Pendidikan Matematika (JUDIKA EDUCATION)*, vol. 2, no. 1, hal. 25-35.
- [2] D. A. Nurhidayah, D., "Analisis Faktor Kesulitan Belajar Matematika Siswa Sma Pada Implementasi Kurikulum 2013". In *Seminar Nasional Pendidikan*, hal. 804-811, 2015.
- [3] P. Guleria, N. Thakur, and M. Sood, "Predicting student performance using decision tree classifiers and information gain," *2014 International Conference on Parallel, Distributed and Grid Computing*, 2014.
- [4] H. Noor, "Optimasi Model Klasifikasi C4. 5 Dan Particle Swarm Optimization Untuk Prediksi Siswa Bermasalah", *Technologia: Jurnal Ilmiah*, vol. 9, no. 4, hal.228-237, 2018.

- [5] M. H. Nasution dan M. I. Sofyan, "Penerapan Metode Decision Tree Untuk Memprediksi Prestasi Siswa Kelas XII Dilihat dari Nilai Akhir Semester di SMK Negeri 1 Selong Tahun Pelajaran 2017/2018". *Infotek: Jurnal Informatika dan Teknologi*, vol. 3, no. 1, hal. 58-65, 2020.
- [6] R. J. Sari, dan A. P. Utomo, "Peningkatan Keaktifan Siswa Dan Hasil Belajar Dengan Menggunakan Metode Pembelajaran Problem Based Learning Pada Siswa Smpn 1 Mayang Kelas Ix". *ScienceEdu*, hal. 80-85, 2019.
- [7] M. I. Ibrahim, A. Dassa, dan M. Dinar, "Pengaruh Model Pembelajaran Kooperatif Tipe Think-Talk-Write (TTW) Terhadap Partisipasi Siswa dan Hasil Belajar Siswa dalam Pelajaran Matematika", *Issues in Mathematics Education (IMED)*, vol.1, no.1, hal.26-32, 2019.
- [8] R. Kartikasari, S. Suwarjo, dan S. Siswanto, "Hubungan Bimbingan Belajar di Luar Sekolah dengan Hasil Belajar Matematika", *Jurnal Pedagogi*, vol.1, no.6, 2019.
- [9] K. Malik dan M. Faid, "Prediksi Prestasi Siswa Smp Nurul Jadid Menggunakan Algoritma C4. 5", *Nusantara Journal of Computers and Its Applications (NJCA)*, vol.1, no.2, 2017.
- [10] K. Nurkhasanah, "Perbandingan Hasil Penggunaan Metode Decision Tree Dan Random Tree Pada Data Training Aplikasi Pencarian Tukang", *ULTIMA INFOSYS*, vol.10, no.2, 2019.
- [11] R. S. Octama, "Enhancing Decision Tree Performance in Credit Risk Classification and Prediction", *Ultimatics*, June 2015.
- [12] J. C. Pal, M. A. Hall, E. Frank, I. H. Witten, "Data Mining: Practical Machine Learning Tools and Techniques", Fourth Edition. New Zealand: Morgan Kaufmann, 2016.
- [13] I. Kurniawan, dan R. A., "Penerapan Algoritma C5. 0 Pada Sistem Pendukung Keputusan Kelayakan Penerimaan Beras Masyarakat Miskin", *Jurnal Informatika*, vol.4, no.2, 2017.
- [14] N. Sagala dan N. Tampubolon, "Komparasi Kinerja Algoritma Data Mining pada Dataset Konsumsi Alkohol Siswa. *Khazanah Informatika*", *Jurnal Ilmu Komputer dan Informatika*, vol.4, no.2, hal.98-103, 2018.
- [15] A. P. Wibawa, M. G. A. Purnama, M. F. Akbar, dan F. A. Dwiyanto, "Metode-metode Klasifikasi". In *Prosiding Seminar Ilmu Komputer dan Teknologi Informasi (SAKTI)*, vol.3, no.1, hal. 134-138, 2018.
- [16] S. Hardikar, A. Shrivastava and V. Choudhary, "Comparison between ID3 and C4.5 in Contrast to IDS", *VSRDIJCSIT*, vol. 2, no. 7, 2012.



# Gray Level Co-ocurrence Matrix untuk Pengekstrasian Ciri Topeng Cirebon

Felix Indra Kurniadi<sup>1</sup>, Vinnia Kemala Putri<sup>2</sup>

<sup>1</sup> Fakultas Teknik Informatika, Jurusan Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia

felixindra@tau.ac.id

<sup>2</sup> Fakultas Ilmu Komputer, Universitas Indonesia, Depok, Indonesia  
vinnia.kemala51@ui.ac.id

Diterima 5 Maret 2020

Disetujui 17 Juni 2020

**Abstract**—Cirebon mask is one of the intangible cultural heritage in Indonesia. It is one of the prominent cultural assets from Cirebon and becoming one of the identity Cirebon culture. However, the current condition people tend to forget the cultural asset and lack of help from the government makes the Cirebon mask become the third-rate assets. Our concern lays on the extinction of this Mask. We want to implement digitation and automatic identification using image processing techniques. In this paper, we applied the Gray Level Co-occurrence Matrix for extracting the features. K-Nearest Neighbour as the classifier. The best result of this research is done by first order statistical features with the accuracy 40,67%.

**Index Terms**—Cirebon Mask, GLCM, Indonesian Culture, K-Nearest Neighbour

## I. PENDAHULUAN

Topeng Cirebon merupakan salah satu kekayaan budaya Indonesia. Topeng ini merupakan identitas budaya Jawa Barat terutama Cirebon. Topeng Cirebon sendiripun merupakan salah satu bentuk akulturisasi antara adat Jawa Barat dengan Agama. Saat ini topeng Cirebon sering digunakan untuk upacara keadatan maupun pertunjukan hiburan. Sayangnya kebudayaan topeng Cirebon tidak menjadi perhatian dari Pemerintahan maupun kaum generasi muda bahkan berdasarkan data di UNESCO yang menjadi Warisan budaya takbenda dari Indonesia adalah pencak silat, pinisi, tiga jenis tarian bali, Noken, Tari Saman, Angklung, Batik, Kris dan Teater Wayang [1]. Hal ini ditakutkan akan membuat kesenian topeng Cirebon makin menghilang dan dilupakan oleh orang-orang.

Berdasarkan problem ini kami melakukan proses digitalisasi untuk menyimpan kebudayaan topeng Cirebon dalam bentuk gambar. Sayangnya proses mengklasifikasikan topeng-topeng tersebut sangat susah dan memakan waktu yang lama dikarenakan beberapa hal seperti keunikan setiap jenis topeng Cirebon baik secara ornament maupun ekspresinya. Hal inilah yang membuat penulis ingin melakukan proses klasifikasi secara otomatis topeng-topeng Cirebon.

Pengklasifikasian topeng Cirebon pernah dilakukan oleh Kurniadi. Kurniadi mengusulkan penyelesaian klasifikasi topeng Cirebon menggunakan metode robust statistic menggunakan z-score untuk mencari data yang dianggap sebagai outlier. Metode yang diusulkan oleh Kurniadi memberikan hasil penambahan akurasi akan tetapi kelemahan terbesar dari penelitian ini adalah penggunaan metode ekstraksi fitur yang sederhana dalam menyelesaikan data yang kompleks [2].

Saat ini penelitian terhadap topeng Cirebon masih sedikit, oleh karena hal ini maka untuk mendapatkan sebuah keterbaruan yang layak digunakan dalam proses klasifikasi dari topeng Cirebon maka perlu dilakukan proses pencarian state of the art dari berbagai penelitian yang mencoba menyelesaikan permasalahan klasifikasi dan identifikasi terhadap budaya Indonesia lainnya.

Penelitian pernah dilakukan untuk menyelesaikan permasalahan klasifikasi terhadap warisan budaya Indonesia. Harjoko et.al. mencoba mengatasi permasalahan terhadap klasifikasi topeng Bali menggunakan *Convolutional Neural Network* (CNN). Hasil yang diberikan dengan menggunakan metode CNN sudah baik walaupun dalam beberapa kasus tidak mampu membedakan topeng yang kompleks [3]. Aditya et.al. mengusulkan penggunaan *Gray Level Co-occurrence Matrix* (GLCM) dan *statistical color* RGB menggunakan *neural network* sebagai *classifier*-nya hasil yang didapatkan pada penelitian ini adalah kombinasi dari kedua fitur ini menghasilkan nilai akurasi yang baik [4]. Sulistianingsih et.al. mengusulkan metode GLCM untuk ekstraksi fitur dengan *Correlation Based Feature Selection* (CFS) untuk seleksi fitur dan klasifikasi menggunakan metode *K-Nearest Neighbour* (KNN) dan *Backpropagation Neural Network* sebagai klasifier. Hasil terbaik didapatkan menggunakan *Backpropagation Neural Network* [5]. Irawan et.al. menggunakan GLCM untuk klasifikasi Batik Lasem dengan klasifier menggunakan KNN dimana hasil yang didapatkan didapatkan bahwa penggunaan data train yang banyak akan memberikan hasil yang lebih baik dimana akurasi terbaik adalah 73.33% [6].

Berdasarkan penelitian sebelumnya dapat disimpulkan bahwa penggunaan GLCM adalah salah satu metode yang banyak digunakan untuk melakukan proses ekstraksi ciri. Sehingga pada penelitian ini akan menggunakan ekstraksi ciri menggunakan metode GLCM dan *classifier* menggunakan *K-Nearest Neighbour*. Secara garis besar yang dilakukan mirip dengan yang dilakukan oleh Kurniadi akan tetapi perbedaan antara penelitian ini dengan yang dilakukan pada penelitian sebelumnya adalah tidak adanya penggunaan outlier detection, hal ini dilakukan untuk membuktikan bahwa metode GLCM lebih baik dibandingkan metode statistical feature sederhana

Kontribusi penulis dalam penelitian ini adalah:

1. Proses digitalisasi dan identifikasi otomatis berbasis pendekatan *artificial intelligence*.
2. Pengimplementasian metode GLCM pada dataset topeng Cirebon. Dimana GLCM bekerja sebagai ekstraksi fitur.
3. Pengimplementasian metode *classifier* seperti *K-Nearest Neighbour*.
4. Penggunaan augmentasi data untuk mengukur keakuratan model.

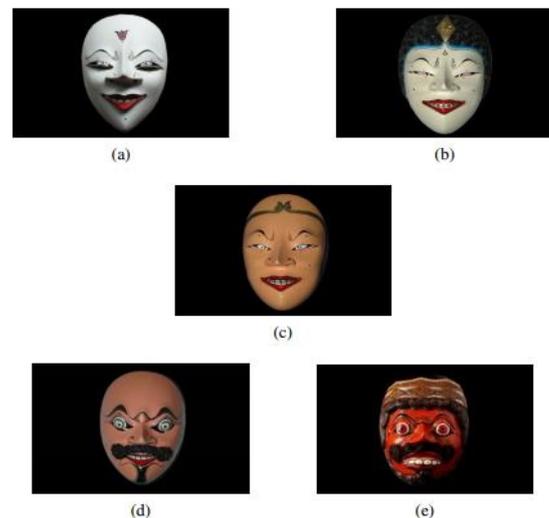
## II. LANDASAN TEORI

Pada bagian ini akan menjelaskan beberapa teori dasar yang mendukung penelitian ini. Secara berurutan akan dijelaskan mengenai Topeng Cirebon, *Gray Level Co-occurrence Matrix*, dan *K-Nearest Neighbour*.

### A. Topeng Cirebon

Topeng Cirebon diciptakan oleh Sunan Kalijaha yang bekerja sama dengan Sunan Gunung Jati, Topeng Cirebon awalnya merupakan salah satu sarana untuk media dakwah agama Islam di pesisir utara pada abad ke-15. Topeng Cirebon sendiri merupakan adaptasi dari kisah Panji yang berasal dari kerajaan Jenggala Jawa Timur [7].

Topeng Cirebon sendiri memiliki lima jenis topeng yang paling terkenal dan sering digunakan dalam pertunjukan yaitu Panji, Samba, Rummyang, Kelana dan Tumenggung. Gambar 1 memberikan gambaran kelima jenis topeng Cirebon.



Gambar 1. Jenis-jenis topeng Cirebon: a) Panji; b) Samba; c) Rummyang; d) Tumenggung; e) Kelana

Setiap topeng Cirebon memiliki karakteristik penokohan yang unik dan terkadang kompleks seperti sifat manusia dimana Panji yang bertindak sebagai gambaran bayi yang baru lahir dan kewibawaan dari seorang raja dimana Panji memiliki ketenangan dan kewibawaan. Samba merepresentasikan sikap suka merendahkan serta kawan dan penuh kaingin tahun. Rummyang merepresentasikan seseorang yang penuh dengan rasa ingin tahu terhadap lingkungan sekitarnya dan refleksi diri dalam keragu-raguan. Tumenggung merepresentasikan kedewasaan dan kemauan keras serta keberanian dan ambisius. Kelana merepresentasikan Hasrat duniawi yang penuh nafsu dan kekerasan [8].

### B. *Gray Level Co-occurrence Matrix*

*Gray Level Co-occurrence Matrix* (GLCM) adalah sebuah fitur ekstraksi yang mengambil konsep dari seon-order statistical features yang merupakan pecahan dari Harallick features. GLCM diketahui baik dalam mengekstraksi ciri dari tekstur sebuah gambar. Secara general, GLCM membagi gambar dalam 4 angular direction yaitu  $\theta = 00, 45, 90, \text{ and } 135$ . GLCM menggunakan beberapa konsep yaitu *Contrast*, *Homogeneity*, *Energy* dan *Correlation*. Sehingga dari hasil ini setiap gambar akan menghasilkan 16 fitur vektor. *Contrast* merupakan pengukuran perbedaan antara *gray level* dalam sebuah area dalam gambar. *Contrast* sangat bergantung pada proses pengambilan gambar yang diambil [6].

$$\text{contrast} = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} (i-j)^2 \cdot P_{ij} \quad (1)$$

*Homogeneity* merujuk pada *Inverse Difference Moment* yang merupakan kemiripan terhadap intensitas yang didapatkan dalam sebuah gambar. Secara garis besar gambar yang homogenus akan

memiliki nilai *gray level* yang *uniform* dan memiliki nilai *inverse value* yang tinggi [6].

$$\text{homogeneity} = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} \frac{1}{1 + (i - j)^2} P_{ij} \quad (2)$$

*Correlation* adalah sebuah teknik untuk menghitung secara akurat perubahan pada gambar. Biasanya *correlation* digunakan untuk mengukur ketergantungan linear terhadap intensitas [6].

$$\text{correlation} = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} \frac{(i - \mu_i)(j - \mu_j)P_{ij}}{\sigma_i \sigma_j} \quad (3)$$

Energi menghitung keseragaman antar *pixel* yang memiliki pasangan berulang dalam citra [6].

$$\text{energy} = \sum_{i=1}^M \sum_{j=1}^M (P_{i,j})^2 \quad (4)$$

Dimana  $P(i,j)$  adalah matriks GLCM,  $\mu$  adalah mean dan  $\sigma$  adalah standar deviasi dari matriks GLCM dan  $i$  serta  $j$  merepresentasikan baris dan kolom dari matriks GLCM.

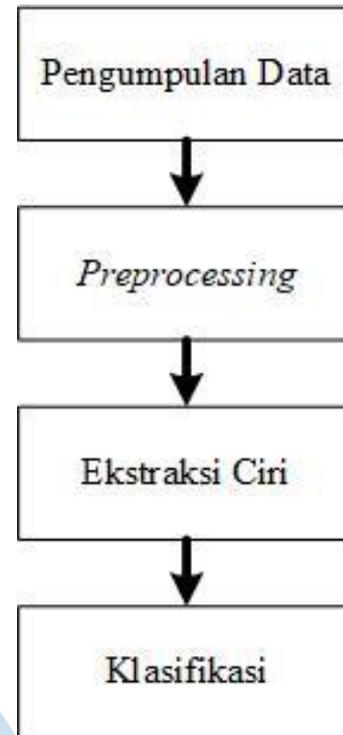
### C. K-Nearest Neighbour

*K-Nearest Neighbour* (KNN) merupakan salah satu algoritma sederhana yang sering digunakan dikarenakan banyak data yang secara linear dapat dipisahkan. Hal inilah yang membuat KNN merupakan salah satu algoritma yang tangguh [9]. Secara garis besar KNN melihat kedekatan antar  $k$ -data. Dimana data test yang dimasukkan akan melihat persebaran tetangga terdekat dan mengambil kelas berdasarkan kelas terbanyak dari tetangga terdekat [9].

Metode pengukuran jarak yang digunakan adalah *Euclidean distance* dan berdasarkan penelitian yang pernah dilakukan oleh Kurniadi mengenai perbandingan *distance metric* [10] bahwa penggunaan *Euclidean* dan *Minkowski distance* memberikan hasil terbaik dibandingkan dengan metode *distance metric* lainnya. Persamaan dibawah merupakan persamaan untuk *Euclidean distance*.

$$d(a, b) = \sqrt{\sum_{j=1}^n (a_j - b_j)^2} \quad (5)$$

Dimana  $a, b$  adalah dua data dan  $d$  merepresentasikan jarak dan  $j$  adalah banyaknya data.



Gambar 2. Metodologi penelitian klasifikasi topeng Cirebon

## III. METODOLOGI

Pada Bagian ini akan dibahas mengenai proses pengumpulan data dan bagaimana data tersebut diolah. Pra-proses terhadap data yang digunakan, proses ekstraksi ciri dan proses klasifikasi. Gambar 2 menggambarkan secara runtut proses yang dilakukan dalam penelitian yang dilakukan saat ini.

### A. Pengumpulan Data

Data dikumpulkan menggunakan search engine dan pengambilan data primer ke lapangan. Penggunaan data search engine yang tersedia untuk publik dilakukan sebagai benchmark apakah data yang bersifat publik cukup relevan dengan data yang didapatkan dari lapangan. Hasil dari pengumpulan data menggunakan search engine dan pengambilan data primer menghasilkan total 150 gambar yang dibagi menjadi 30 untuk setiap kelas. Untuk membuat representasi yang baik maka pembagian data untuk menjadi data latih dan data uji menggunakan perbandingan 80:20. Dimana data latih menjadi sebesar 24 dan data uji sebesar 6.

Data yang didapatkan secara manual ini dikembangkan dengan menggunakan proses augmentasi. Proses augmentasi dilakukan untuk membuat dataset menjadi lebih banyak dengan melakukan proses pemberian derau, rotasi dan peningkatan intensitas. Hasil dari augmentasi citra ini mendapatkan total 4500 data dengan 900 citra untuk setiap kelas. Hasil ini kemudian dibagi menjadi data

latih dan data uji. Tabel 1 sampai Tabel 2 menggambarkan pembagian data dan jumlah data.

Tabel 1. Total dataset dari *original* dataset dengan augmentasi dataset

Original	150
Augmentasi	4500

Tabel 2. Dataset per-kelas dari original dataset dengan augmentasi dataset dan pembagian antara data latih dan data uji

	train	test	Total
Original	24	6	30
Augmentasi	792	198	900

### B. Pra Proses

Pengontrolan dataset dilakukan dengan menghapus *background* pada citra dan merubah gambar menjadi keabuan. Selain kedua proses tersebut *window size* dari gambar juga diubah menjadi 100 x 100 *pixel*. Proses ini dilakukan dengan bantuan perangkat lunak untuk memanipulasi gambar.

### C. Fitur Ekstraksi

Pada proses ekstraksi fitur dilakukan pengekstrasian ciri citra dengan menggunakan metode GLCM yang sudah dijelaskan pada Bagian II. Proses ini dilakukan dengan menggunakan semua 9 sehingga menghasilkan 16 fitur vector yang merupakan gabungan antara 4 9 dan 4 GLCM algoritma yaitu *homogeneity, contrast, correlation* dan *energy*.

Hasil dari proses ekstraksi fitur ini kemudian akan dibuatkan model dengan menggunakan metode yang diusulkan yaitu *K-Nearest Neighbour*.

### D. Klasifikasi

Pada proses klasifikasi dilakukan pemasangan model secara garis besar proses klasifikasi berbeda antara proses latih dan uji, dimana pada proses latih fitur yang didapatkan pada proses ekstraksi fitur dimasukkan kedalam metode KNN dan metode KNN akan melakukan persebaran data yang nantinya akan menjadi model utama yang akan digunakan dalam proses pengujian.

Pada proses pengujian fitur yang dimasukkan ke dalam model dan model akan menghasilkan nilai kelas dan jika nilai kelas yang dihasilkan sesuai dengan nilai kelas yang diinginkan maka model dapat mengenali data tersebut dengan baik.

Proses evaluasi pada penelitian ini akan menggunakan akurasi, presisi dan *recall*.

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} * 100\% \quad (6)$$

$$precision = \frac{tp}{tp + fp} * 100\% \quad (7)$$

$$recall = \frac{tp}{tp + fn} * 100\% \quad (8)$$

Dimana *tp, tn, fp* dan *fn* merupakan *true positive, false positive, true negative* dan *false negative*. Konsep ini diambil dari konsep *confusion matrix*.

## IV. EKSPERIMEN DAN HASIL

Pada bagian ini akan dijelaskan mengenai skenario eksperimen yang dilakukan serta hasil yang didapatkan dari penelitian ini. Selain itu akan dijelaskan mengenai hasil yang didapatkan.

### A. Skenario Eksperimen dan Konfigurasi Eksperimen

Pada penelitian ini akan digunakan KNN dengan membandingkan 3 jenis *K-Neighbour* yaitu bernilai 3, 5 dan 7. Selain itu data yang akan digunakan adalah data *original* dan *augmentasi*.

Eksperimen ini dilakukan menggunakan CPU dengan prosesor Intel i7 dan RAM sebesar 4GB. Proses pembagian *train* dan *test* dilakukan secara manual.

Tabel 3. Hasil eksperimen pada data original

Metode	Neighbour	akurasi	presisi	recall
First order statistical[2]	k=3	40	0.47	0.4
	k=5	40	0.42	0.44
	k=7	36.67	0.43	0.37
GLCM	k=3	16.67	0.11	0.16
	k=5	26.67	0.2	0.27
	k=7	26.67	0.23	0.27

Tabel 4. Hasil eksperimen pada data augmentasi

Metode	Neighbour	akurasi (%)	presisi	recall
First order statistical[2]	k=3	40.69	0.4	0.41
	k=5	40.69	0.4	0.41
	k=7	40.49	0.39	0.41
GLCM	k=3	33.14	0.34	0.33
	k=5	33.17	0.34	0.33
	k=7	31.07	0.31	0.31

Berdasarkan hasil di atas bahwa metode GLCM yang diusulkan masih tidak mampu mengalahkan metode first order statistic yang sudah dilakukan oleh Kurniadi [2] sebelumnya. Secara keseluruhan akurasi yang didapatkan baik dari data *augmentasi* dan bukan *augmentasi* mengecewakan dikarenakan hasil tidak dapat menghasilkan nilai lebih dari 50%. Hal ini merupakan problem yang dikemukakan oleh [2] pada penelitian sebelumnya dimana dataset public yang tersedia tidak merepresentasikan data yang dilapangan sehingga tidak cocok digunakan.

Proses augmentasi datapun tidak merubah hasil untuk *first order statistical feature* akan tetapi pada metode GLCM memberikan peningkatan hasil. Sayangnya penggunaan data augmentasi tidak terlalu berperan banyak dalam bahkan peningkatan signifikan hanya terjadi pada  $k=3$  dengan metode GLCM.

#### V. SIMPULAN

Penelitian pengklasifikasian topeng Cirebon menggunakan metode GLCM memberikan beberapa hasil baru salah satunya adalah data publik yang digunakan tidak dapat digunakan menjadi acuan yang tepat sehingga harus dilakukan observasi lapangan yang lebih detail. Metode GLCM yang diusulkan tidak memberikan hasil yang memuaskan bahkan metode yang diusulkan tidak dapat menandingi metode yang pernah diusulkan sebelumnya, yang dimana hasil akurasi terbaik dari metode GLCM adalah 33.14% pada data augmentasi. Selain itu penggunaan data augmentasi pada penelitian ini tidak terlalu berpengaruh.

Penelitian ini bisa menjadikan dasar untuk penelitian lain yang ingin berfokus pada klasifikasi topeng Cirebon.

#### UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih kepada Fendy Hendriyanto sebagai pengumpul data dan penulis juga berterima kasih kepada Kementerian Riset dan Teknologi Republik Indonesia.

#### DAFTAR PUSTAKA

- [1] United Nations Educational, Scientific and Cultural Organization, "Browse the Lists of Intangible Cultural Heritage and the Register of good safeguarding practices," Intangible Cultural Heritage. [Online]. Available: [https://ich.unesco.org/en/lists?text=&country\[\]=00104&multi-national=3&display=inscriptionID#tabs](https://ich.unesco.org/en/lists?text=&country[]=00104&multi-national=3&display=inscriptionID#tabs). [Accessed: 21-Feb-2020].
- [2] F. I. Kurniadi and F. Hendriyanto, "Cirebon Mask Classification using Robust K-Nearest Neighbour," in 2019 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2019, pp. 143–146, doi: 10.1109/ICOIACT46704.2019.8938499.
- [3] B. P. Ida Bagus Gede and A. Harjoko, "Pengenalan Topeng Bali Menggunakan Algoritma Convolutional Neural Network," Master's Thesis, Universitas Gadjah Mada, 2017.
- [4] C. S. K. Aditya, M. Hani'ah, R. R. Bintana, and N. Suciati, "Batik classification using neural network with gray level co-occurrence matrix and statistical color feature extraction," in 2015 International Conference on Information Communication Technology and Systems (ICTS), 2015, pp. 163–168, doi: 10.1109/ICTS.2015.7379892.
- [5] N. Sulistianingsih, I. Soesanti, and R. Hartanto, "Classification of Batik Image using Grey Level Co-occurrence Matrix Feature Extraction and Correlation Based Feature Selection," in 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 2018, pp. 492–497, doi: 10.1109/ISRITI.2018.8864237.
- [6] C. Irawan, E. N. Ardyastiti, D. R. I. M. Setiadi, E. H. Rachmawanto, and C. A. Sari, "A Survey: Effect of the Number of GLCM Features on Classification Accuracy of Lasem Batik Images using K-Nearest Neighbor," in 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 2018, pp. 33–38, doi: 10.1109/ISRITI.2018.8864443.
- [7] Lasmiyati, "Sejarah Pertumbuhan dan Pengembangan Tari Topeng Cirebon Abad XV-XX," Patanjala, vol. 3, no. 3, pp. 472–487, Sep. 2011.
- [8] Pusat Studi Sunda, Kujang, bedog, dan topeng dan kajian lainnya mengenai budaya Sunda, Seri Sunda. Pusat Studi Sunda, 2008.
- [9] F. I. Kurniadi and V. K. Putri, "A Comparison of Human Crafted Features and Machine Crafted Features on White Blood Cells Classification," J. Phys.: Conf. Ser., vol. 1201, p. 012045, May 2019, doi: 10.1088/1742-6596/1201/1/012045.
- [10] F. I. Kurniadi and V. K. Putri, "Perbandingan Distance Metric pada Nearest Neighbour untuk Klasifikasi Sel Darah Putih," Ultimatics, vol. XI, no. 1, p. 4.

# Visualisasi Algoritma sebagai Sarana Pembelajaran *K-Means Clustering*

Alethea Suryadibrata<sup>1</sup>, Julio Christian Young<sup>2</sup>

<sup>1,2</sup>Program Studi Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia  
alethea@umn.ac.id  
julio.christian@umn.ac.id

Diterima 30 Maret 2020

Disetujui 17 Juni 2020

**Abstract**—Algorithm Visualization (AV) is often used in computer science to represent how an algorithm works. Educators believe that visualization can help students to learn difficult algorithms. In this paper, we put our interest in visualizing one of Machine Learning (ML) algorithms. ML algorithms are used in various fields. Some of the algorithms are used to classify, predict, or cluster data. Unfortunately, many students find that ML algorithms are hard to learn since some of these algorithms include complicated mathematical equations. We hope this research can help computer science students to understand K-Means Clustering in an easier way.

**Index Terms**—Algorithm Visualization, K-Means Clustering, Machine Learning

## I. PENDAHULUAN

Memberikan penjelasan yang dapat dimengerti pelajar merupakan tantangan dari seorang pengajar. Berbagai studi dilakukan untuk membantu pengajar menyelesaikan masalah tersebut salah satunya melalui proses visualisasi materi ajar. Proses visualisasi yang dimaksud adalah serangkaian kegiatan untuk merepresentasikan informasi dalam berbagai bentuk seperti animasi, teks, dan diagram statis [1]. Pada bidang Ilmu Komputer, proses visualisasi banyak digunakan untuk menjelaskan bidang ilmu yang melibatkan konsep pemrograman dan algoritma.

Pada penelitian terdahulu, Cetin dan Andrews-Larson telah mencoba memvisualisasikan tahapan-tahapan dalam suatu algoritma *sorting* [2]. Berdasarkan proses visualisasi, peneliti menyimpulkan bahwa materi algoritma yang tadinya bersifat *cryptic* menjadi lebih mudah dimengerti oleh para pelajar. Kemudian, pada penelitian lainnya, Nathasya dkk. telah mencoba mengintegrasikan proses visualisasi untuk proses pembelajaran materi struktur data [3]. Melalui penelitian yang dilakukan, proses visualisasi bahkan terbukti meningkatkan motivasi, performa dan capaian pembelajaran para pelajar (para pelajar mendapatkan nilai ujian yang lebih baik dan lebih bersemangat dalam mengerjakan tugas). Selain digunakan untuk materi-materi pembelajaran yang sederhana, visualisasi juga dapat digunakan untuk membantu proses pembelajaran algoritma yang bersifat lebih kompleks. Hal ini dibuktikan melalui

penelitian yang dilakukan oleh Sriadhi dkk., yang memvisualisasikan proses dari salah satu algoritma pada bidang kriptografi, RC4 [4].

Selain bidang kriptografi, pembelajaran mesin juga merupakan salah satu bidang yang bersifat kompleks dan dirasa sulit oleh para pelajar. Hal ini dikarenakan algoritma pembelajaran mesin banyak melibatkan rumus-rumus matematika yang bersifat abstrak dan *cryptic*. Padahal, algoritma pembelajaran mesin cukup banyak digunakan untuk menyelesaikan berbagai permasalahan di dunia nyata seperti memprediksi pergerakan data runtun waktu [5], pengenalan karakter dalam dokumen cetak [6], ataupun pada berbagai bidang lainnya.

Berdasarkan alasan di atas, penelitian ini mencoba untuk memvisualisasikan salah satu algoritma pembelajaran mesin, yaitu algoritma *K-Means Clustering*. Algoritma ini merupakan salah satu algoritma pembelajaran mesin yang sering digunakan untuk mengelompokkan sekumpulan data berdasarkan informasi yang dimilikinya.

## II. TELAAH LITERATUR

### A. Visualisasi Algoritma

Visualisasi algoritma merupakan salah satu alternatif untuk mempelajari suatu algoritma dengan menggunakan sekumpulan elemen grafis yang berisikan informasi-informasi penting dalam suatu algoritma. Elemen grafis digunakan untuk visualisasi seperti titik, garis, dan sebagainya [7].

Terdapat 2 variasi pada visualisasi algoritma yaitu:

1. Visualisasi algoritma statis, di mana penjelasan algoritma melibatkan elemen-elemen grafis yang tidak bergerak.
2. Visualisasi algoritma dinamis (sering juga disebut animasi algoritma), di mana proses visualisasi melibatkan elemen-elemen grafis yang saling berinteraksi. Teknik visualisasi algoritma dinamis cenderung lebih canggih dan sulit untuk diimplementasikan.

### B. Pembelajaran Mesin

Pemelajaran mesin (*machine learning*) adalah pemrograman komputer untuk mengoptimalkan performa menggunakan sampel data atau pengetahuan sebelumnya [8]. Pemelajaran mesin merupakan bagian dari kecerdasan buatan (*Artificial Intelligence / AI*). Untuk menjadi cerdas, sebuah sistem di dalam lingkungan yang berubah-ubah harus memiliki kemampuan untuk belajar. Jika sistem dapat belajar untuk beradaptasi dalam perubahan yang ada, maka desainer sistem tidak perlu meramalkan serta menyediakan solusi untuk semua situasi yang dapat terjadi.

Pemelajaran mesin juga membantu kita untuk mencari solusi dari masalah-masalah dari data visual, pengenalan suara, dan robotik [8]. Pemelajaran mesin dapat menghasilkan prediksi untuk masa depan, atau deskriptif untuk mendapatkan pengetahuan dari data, atau bahkan keduanya. Pemelajaran mesin menggunakan teori statistik dalam membangun suatu model matematika. Peran ilmu komputer pada pemelajaran mesin dapat dibagi menjadi dua. Pertama, pada pelatihan, kita membutuhkan algoritma yang efisien untuk menyelesaikan masalah optimasi serta menyimpan dan memroses data yang kita miliki. Kedua, setelah model dipelajari, representasi serta solusi yang diberikan juga harus efisien.

### C. K-Means Clustering

Pada bidang pemelajaran mesin, istilah *clustering* merujuk pada sebuah tugas untuk memisahkan/mengelompokkan sekumpulan data ke dalam sejumlah cluster berdasarkan hasil pengukuran suatu metrik yang menyatakan kesamaan/ jarak dari data [9]. *K-Means* merupakan salah satu algoritma untuk proses *clustering* yang bersifat sederhana secara matematis dan efektif, sehingga dalam lima dekade terakhir, algoritma ini masih sangat sering digunakan untuk menyelesaikan berbagai macam permasalahan.

Langkah-langkah algoritma *K-Means Clustering*:

1. Tentukan suatu nilai  $k$  yang menyatakan jumlah kelompok (*cluster*).
2. Inisialisasi posisi sejumlah  $k$  vektor secara acak, di mana setiap vektor merepresentasikan pusat (*center*) dari sebuah kelompok.
3. Berdasarkan setiap vektor pusat yang ada, hitung jarak dari setiap vektor pusat ke setiap data dalam representasi vektor. Berdasarkan perhitungan jarak yang dilakukan, sebuah titik akan dijadikan anggota dari sebuah kelompok berdasarkan jarak titik pusat kelompok terdekat dengannya.
4. Hitung kembali vektor pusat dari setiap kelompok dengan mencari rata-rata dari semua vektor data dalam kelompok.

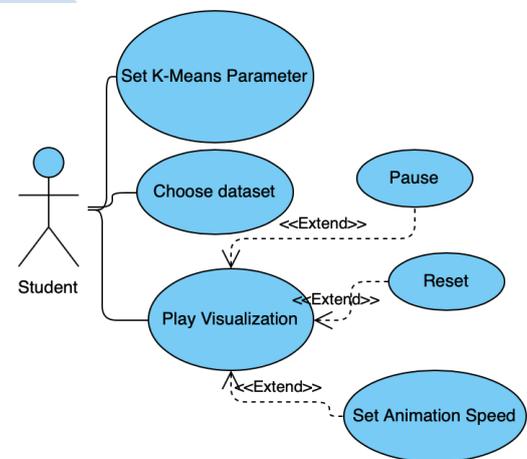
5. Jika nilai vektor pusat tidak mengalami perubahan, maka setiap titik pusat telah merepresentasikan kelompok dengan benar dan proses pengelompokan selesai. Jika nilai vektor pusat masih berubah, kembali ke tahapan nomor 3.

## III. METODE PENELITIAN

Bagian ini menjelaskan bagaimana perancangan pembuatan aplikasi dilakukan serta fitur-fitur yang dianggap penting untuk meningkatkan interaktifitas aplikasi dengan pengguna. Fokus pada penelitian ini adalah untuk membuat visualisasi yang dapat digunakan oleh pelajar yang ingin memahami cara kerja algoritma *K-Means Clustering*.

### A. Use Case Diagram

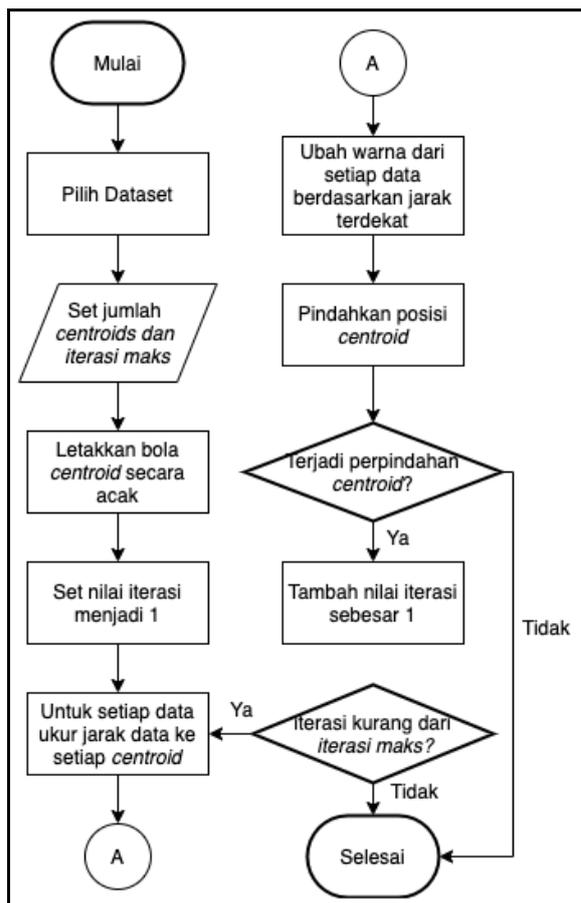
Pada perancangan aplikasi, dibuatlah sebuah *Use Case Diagram* untuk merepresentasikan interaksi *user* dengan aplikasi. Berdasarkan Gambar 1, pengguna dapat memasukan jumlah cluster dan jumlah iterasi ke dalam aplikasi sebelum visualisasi dimulai.



Gambar 1. *Use Case Diagram* visualisasi algoritma *K-Means Clustering*

### B. Flowchart Algoritma K-Means Clustering

Pada penelitian ini, proses visualisasi dilakukan dengan merepresentasikan data sebagai sekumpulan bola-bola yang tersebar dalam sebuah ruang 3D. Di dalam aplikasi, setiap kelompok data akan direpresentasikan sebagai sebuah warna. Untuk setiap bola yang merepresentasikan sebuah data, warna dari bola akan berubah-ubah menyesuaikan kelompok di mana ia berada. Gambar 2 menunjukkan *flowchart* algoritma *K-Means Clustering* dan bagaimana proses visualisasi dilakukan dengan menggunakan algoritma tersebut.



Gambar 2. Flowchart visualisasi algoritma *K-Means Clustering*

### C. Fitur-fitur Interaktif Aplikasi

Dalam sebuah proses visualisasi, fitur-fitur interaktif digunakan agar aplikasi menjadi lebih menarik sehingga dapat meningkatkan motivasi belajar. Selain itu, fitur-fitur interaktif juga dapat digunakan untuk mengikuti kecepatan belajar masing-masing pelajar. Dalam aplikasi yang dikembangkan, salah satu fitur interaktif yang diimplementasikan adalah animasi dari proses perhitungan jarak antara setiap vektor data dan setiap vektor pusat yang ada menggunakan sebuah garis pembantu. Dalam proses visualisasi, garis pembantu berperan untuk menunjukkan proses pencarian jarak terdekat dari setiap vektor data.

Selain itu, fitur interaktif kedua yang terdapat dalam aplikasi adalah animasi perubahan warna dari setiap vektor data berdasarkan warna dari titik pusat *cluster* yang paling dekat dengannya. Proses ini digunakan untuk menunjukkan kepada para peserta ajar bagaimana titik pusat dari setiap *cluster* dapat terbentuk dan berubah-ubah pada setiap iterasi dalam algoritma. Kemudian, untuk mempermudah pengguna dalam proses belajar, aplikasi juga menyediakan fitur untuk mengatur kecepatan animasi dan fitur jeda (*pause*) yang dapat digunakan penggunaannya sesuai dengan kebutuhan.

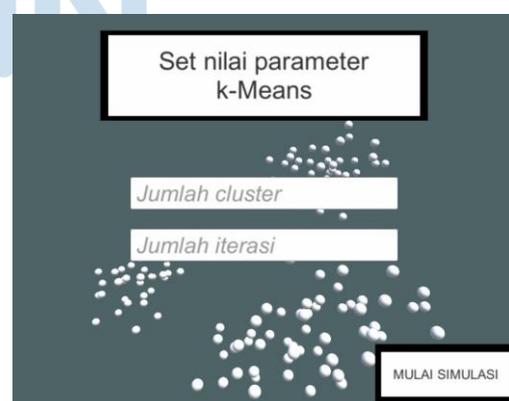
## IV. IMPLEMENTASI

Aplikasi yang digunakan untuk membuat visualisasi algoritma *K-Means Clustering* dibuat menggunakan sebuah *game engine* bernama *Unity* dan merupakan aplikasi 3D. Sebelum menjalankan visualisasi, pengguna tentunya harus menentukan dataset yang ingin digunakan; dan beberapa parameter seperti nilai  $k$ , *tolerance*, jumlah iterasi maksimum dari algoritma. Gambar 3 dan Gambar 4 menunjukkan proses pemilihan dataset dan pengaturan parameter dalam aplikasi.

Melalui Gambar 3, dapat terlihat bahwa pengguna dapat memilih dataset yang ingin digunakan menggunakan tanda panah yang berada di kiri dan kanan layar dan memulai simulasi dengan menekan tombol “Mulai Simulasi”. Kemudian, setelah memilih suatu dataset, sebelum simulasi dimulai, pengguna akan diminta untuk mengisi nilai parameter *K-Means* yang ingin digunakan seperti pada Gambar 4.

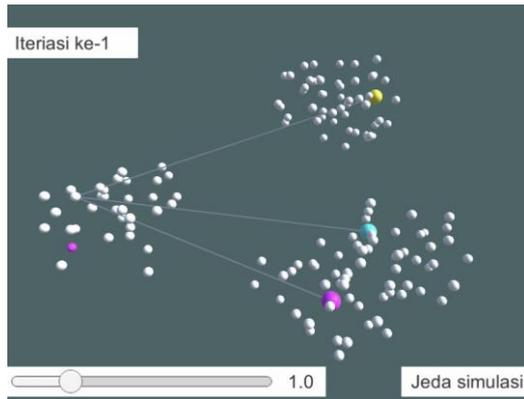


Gambar 3. Proses pemilihan dataset dalam aplikasi

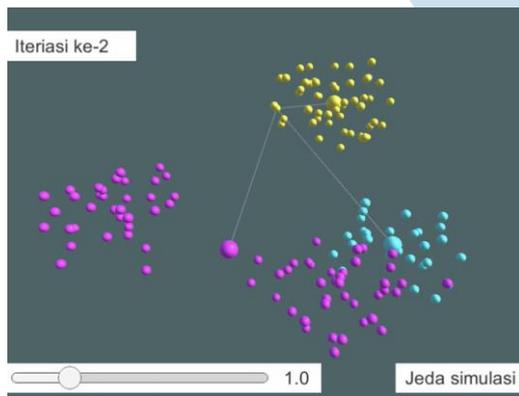


Gambar 4. Proses parameterisasi algoritma *K-Means*

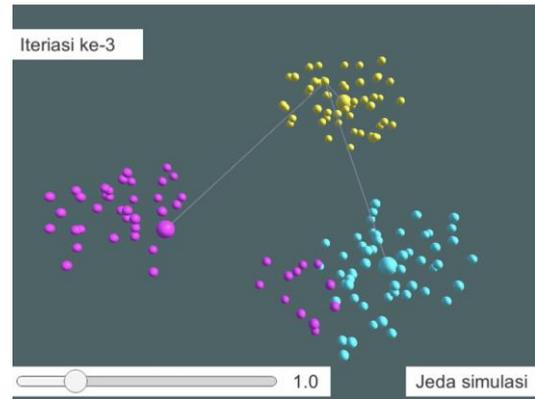
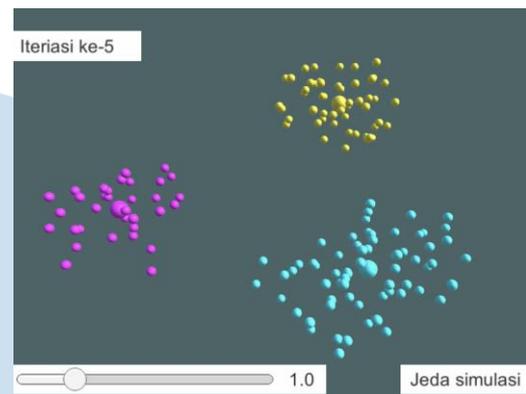
Saat visualisasi dijalankan, mula-mula *centroid* akan diletakkan secara acak dan direpresentasikan sebagai suatu warna. Kemudian, animasi yang menunjukkan perbandingan jarak untuk setiap data ke setiap *centroid* dilakukan (proses ini ditunjukkan pada Gambar 5).

Gambar 5. Proses perbandingan setiap *centroid*

Pada Gambar 5, tiga garis yang mengarah dari setiap bola yang berwarna ke sebuah bola putih yang berada di bagian kiri layar menunjukkan proses perbandingan jarak dari setiap *centroid* ke sebuah data. Berdasarkan proses ini warna dari bola putih akan berubah berdasarkan jarak (garis) terpendek yang terhubung ke dirinya. Animasi dari proses ini akan dilakukan untuk setiap data dan kemudian dilanjutkan dengan proses update lokasi *centroid* sebelum algoritma berjalan ke iterasi selanjutnya. Gambar 6 menunjukkan lokasi *centroid* yang telah di-update berdasarkan iterasi pertama pada iterasi ke-2.

Gambar 6. Proses perpindahan lokasi *centroid* dan perhitungan jarak kembali pada iterasi ke-2

Proses simulasi akan terus dijalankan mengikuti alur dari algoritma *K-Means Clustering* yang telah dijelaskan pada subbab sebelumnya. Gambar 7 dan Gambar 8 menunjukkan proses lanjutan yang terjadi di dalam aplikasi sampai simulasi berakhir.

Gambar 7. Perpindahan *centroid* berdasarkan perhitungan pada iterasi ke-2Gambar 8. *Centroid* telah mencapai tidak lagi mengalami perpindahan

## V. EVALUASI

Berdasarkan aplikasi visualisasi algoritma *K-Means Clustering* yang telah dikembangkan dilakukan *Focus Group Discussion* (FGD) untuk menggali ketepatan, kelebihan serta kekurangan dari aplikasi visualisasi yang telah dikembangkan. FGD dilakukan bersama 3 orang pengajar di salah satu perguruan tinggi swasta yang mengampu mata kuliah pembelajaran mesin.

Berdasarkan FGD yang dilakukan para pengajar merasa aplikasi dapat membantu mempermudah proses pengajaran ataupun proses pembelajaran algoritma secara mandiri. Melalui elemen-elemen visual di dalam aplikasi konsep-konsep abstrak seperti pencarian jarak terdekat, perpindahan *centroid*, perhitungan ulang vektor *centroid* yang bersifat *cryptic* jika disajikan sebagai teks dirasa akan mempermudah proses pemahaman para peserta ajar. Selain itu, berdasarkan diskusi yang dilakukan, dengan menyediakan beberapa variasi dataset dan mengizinkan pengguna melakukan parameterisasi dari algoritma, aplikasi dirasa dapat membantu para pengajar dalam menjelaskan kelebihan dan kekurangan dari algoritma (seperti bagaimana

algoritma *K-Means Clustering* dalam menghadapi kondisi *density-connected region*).

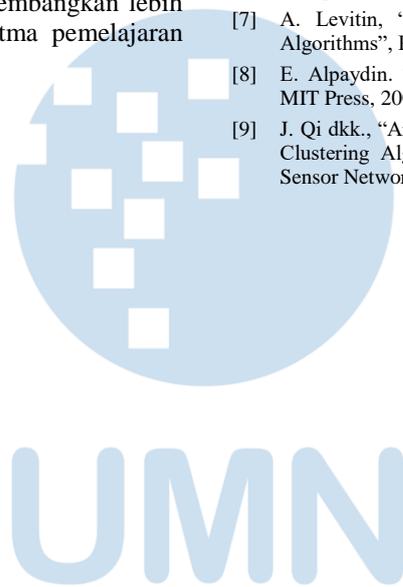
Akan tetapi, terlepas dari kelebihan di atas, aplikasi dirasa masih memiliki beberapa kelemahan berkaitan dengan hal-hal definitif yang berkaitan dengan algoritma *K-Means* seperti bagaimana proses pengukuran jarak dilakukan di dalam aplikasi dan proses perpindahan *centroid*.

## VI. SIMPULAN

Visualisasi algoritma *K-Means Clustering* telah dibuat dalam bentuk 3D menggunakan *Unity* dan bahasa pemrograman *C#*. Visualisasi ini diharapkan dapat membantu pelajar dalam memahami algoritma *K-Means Clustering*. Evaluasi telah dilakukan dalam bentuk FGD terhadap beberapa pengajar mata kuliah Pemelajaran Mesin. Untuk penelitian selanjutnya, dapat dilakukan evaluasi terhadap para pelajar berupa kuesioner yang mengacu pada model evaluasi tertentu. Aplikasi ini juga dapat dikembangkan lebih lanjut dengan menambahkan algoritma pemelajaran mesin lainnya.

## DAFTAR PUSTAKA

- [1] C. D. Hundhausen, S.A. Douglas, dan J.T. Stasko, "A Meta-Study of Algorithm Visualization Effectiveness," *Journal of Visual Languages & Computing*, vol. 13(3), Juni 2002, hal. 259-290.
- [2] I. Cetin dan C. Andrews-Larson, "Learning Sorting Algorithms through Visualization Construction", *Journal Computer Science Education*, vol. 26, 2016, hal 27-43.
- [3] R.A. Nathasya, O. Karnalim, dan M. Ayub, "Integrating Program and Algorithm Visualization for Learning Data Structure Implementation", *Egyptian Informatics Journal*, Vol. 20(3), November 2019, hal. 193-204.
- [4] S. Sriadhi, R. Rahim, dan A.S. Ahmar, "RC4 Algorithm Visualization for Cryptography Education", *Journal of Physics*, [2nd International Conference on Statistics, Mathematics, Teaching, and Research, 2017].
- [5] K. Johan, J.C. Young, dan S. Hansun, "LSTM-RNN Automotive Stock Price Prediction", *International Journal of Scientific & Technology Research*, vol. 9 (8), 2019, hal. 173-176.
- [6] D.P. Chandra dan A. Suryadibrata, "Implementasi Jaringan Saraf Tiruan Backpropagation untuk Pengenalan Karakter pada Dokumen Tercetak", *Ultima Computing: Jurnal Sistem Komputer*, vol. 11, 2019, hal 46-51.
- [7] A. Levitin, "Introduction to the Design & Analysis of Algorithms", Pearson-Addison Wesley, 2007.
- [8] E. Alpaydin. "Introduction to Machine Learning" London : MIT Press, 2009.
- [9] J. Qi dkk., "An Effective and Efficient Hierarchical K-Means Clustering Algorithm", *International Journal of Distributed Sensor Networks*. Vol.13(8), 2017.



UMN

# Pengenalan Wajah Bebas Ekspresi Menggunakan Metode *Nearest Feature Line* dengan Representasi Ciri dalam Ruang Eigen

Is Mardianto<sup>1</sup>

<sup>1</sup> Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Trisakti, Jakarta, Indonesia  
mardianto@trisakti.ac.id

Diterima 17 April 2020

Disetujui 17 Juni 2020

**Abstract**—Facial recognition with different expressions is one part of the pattern recognition problem which is quite complex when compared to pattern recognition on a normal profile. The expression-free face recognition method using the Nearest Feature Line (NFL) technique works by finding the closest projection distance between feature vectors, assuming that the closer the projection distance of a feature vector (face) to another feature vector (face), the more similar the properties will be physical feature vector (face) which are close together. The NFL distance calculation is performed on the eigen dimensional space with the aim that the calculated feature vector (face) dimension has a much smaller dimension in order to increase the level of recognition accuracy and speed up computational time. The test results obtained indicate the NFL method provides a fairly good level of recognition accuracy in the average value of 76.7% with the advantage of low computational time needed when compared with other intelligent methods such as artificial neural network systems.

**Index Terms**—Eigen Value, Expression, Face Recognition, Nearest Feature Line, Principal Component Analysis

## I. PENDAHULUAN

Saat ini perkembangan teknologi pengukuran dan analisa statistik yang berkaitan dengan data biologis berkembang dengan pesat. Penyebabnya karena ciri biologis manusia memberi informasi yang unik berkaitan dengan identifikasi seseorang. Dalam ranah ilmu komputer, teknologi biometrik merujuk kepada teknologi pengukuran dan analisis karakteristik bagian tubuh manusia suara, mata, wajah dan sidik jari.

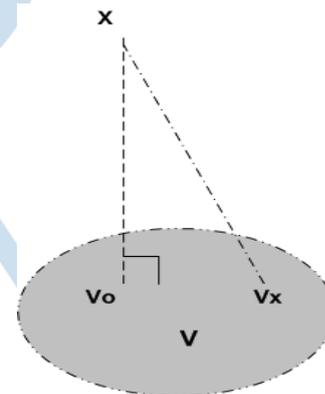
Dalam hal membaca sifat-sifat bagian tubuh manusia seperti wajah dibutuhkan peralatan penyimpanan dan penganalisis pola. Saat "user" mencoba mengakses suatu sistem, maka sistem akan membandingkan data biometrik yang tersimpan dengan data biometrik yang akan mencoba masuk. Algoritme biometrik yang baik akan mampu memutuskan dengan benar apakah "user" yang sedang mencoba mengakses suatu sistem diperkenankan atau tidak memperoleh akses kedalam sistem tersebut. Pada kasus ini, obyek yang sama dengan perbedaan asesoris wajah harus dapat dikenali sebagai satu obyek yang

sama. Untuk memenuhi kebutuhan tersebut dikembangkanlah sistem pengenalan wajah bebas ekspresi dengan menggunakan metode *Nearest Feature Line*.

## II. LANDASAN TEORI

### A. Teorema Proyeksi

Metode *Nearest Feature Line* (NFL) [1] bekerja dengan memanfaatkan teorema proyeksi yaitu "jarak sebuah titik ke sebuah bidang adalah jarak tegak lurus dari titik ke bidang tersebut" seperti terlihat pada Gambar 1.

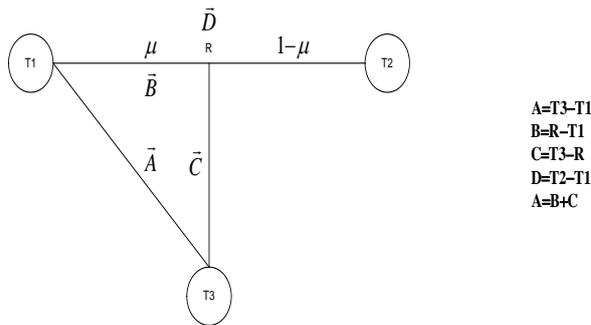


Gambar 1. Proyeksi titik pada sebuah bidang

Dari teorema proyeksi [2] dapat diturunkan sebuah teknik NFL untuk mencari jarak terpendek antara sebuah vektor ciri dengan himpunan vektor ciri lainnya [3].

Dari ilustrasi Gambar 2, dapat diturunkan solusi dari pencarian jarak citra uji T3 terhadap garis ciri yang menghubungkan citra acuan T1 dan T2.

Berdasarkan prinsip interpolasi linear bila diberikan 2 buah titik dalam ruang (T1 dan T2) maka dapat dibentuk sebuah garis lurus yang melewati kedua buah titik tersebut yang memenuhi persamaan 1.



Gambar 2. Penurunan rumus teknik NFL

$$\mathbf{R} = (1-\mu)\mathbf{T1} + \mu\mathbf{T2} = \mathbf{T1} + (\mathbf{T2}-\mathbf{T1})\mu \quad (1)$$

$$\text{sehingga } \mu = \frac{(\mathbf{R}-\mathbf{T1})}{(\mathbf{T2}-\mathbf{T1})}$$

dengan

$$\mathbf{A} = \mathbf{T3}-\mathbf{T1}, \mathbf{B} = \mathbf{R}-\mathbf{T1}, \mathbf{C} = \mathbf{T3}-\mathbf{R}, \mathbf{D} = \mathbf{T2}-\mathbf{T1}, \mathbf{A} = \mathbf{B} + \mathbf{C}$$

maka  $\mu \mathbf{D} = \mathbf{B}$ ,

$$\mu (\mathbf{D} \cdot \mathbf{D}) = \mathbf{B} \cdot \mathbf{D} = (\mathbf{A} - \mathbf{C}) \cdot \mathbf{D} = \mathbf{A} \cdot \mathbf{D} - \mathbf{C} \cdot \mathbf{D} \quad (2)$$

karena jarak terpendek yang dapat dibentuk sebuah titik terhadap sebuah garis adalah hasil perpotongan antara garis yang ditarik dari titik terhadap garis acuan yang membentuk sudut sebesar  $90^\circ$  maka sudut yang dibentuk antara vektor C dengan D adalah  $90^\circ$ , sehingga persamaan 2 menjadi:

$$\mu (\mathbf{D} \cdot \mathbf{D}) = \mathbf{A} \cdot \mathbf{D}$$

sehingga didapatkan solusi:

$$\mu = \frac{(\mathbf{A} \cdot \mathbf{D})}{(\mathbf{D} \cdot \mathbf{D})} = \frac{(\mathbf{T3}-\mathbf{T1}) \cdot (\mathbf{T2}-\mathbf{T1})}{(\mathbf{T2}-\mathbf{T1}) \cdot (\mathbf{T2}-\mathbf{T1})} \quad (3)$$

sedangkan untuk menghitung jarak terdekat antara  $\mathbf{T3}$  dengan  $\mathbf{R}$ :

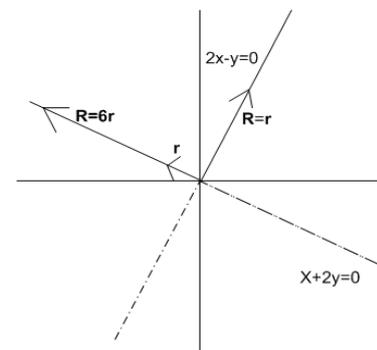
$$\text{Jarak}[\mathbf{T3}, (\mathbf{T2}-\mathbf{T1})] = \|\mathbf{T3}-\mathbf{R}\| \\ \left\| \mathbf{T3} - \left[ \mathbf{T1} + (\mathbf{T2}-\mathbf{T1}) \left[ \frac{(\mathbf{T3}-\mathbf{T1}) \cdot (\mathbf{T2}-\mathbf{T1})}{(\mathbf{T2}-\mathbf{T1}) \cdot (\mathbf{T2}-\mathbf{T1})} \right] \right] \right\| \quad (4)$$

Solusi persamaan (4) digunakan secara praktis untuk menghitung jarak terpendek kemiripan antara sebuah citra dengan citra lainnya dengan memandang sebuah citra sebagai sebuah vektor citri.

### B. Eigen Value dan Eigen Vector

Pemahaman tentang konsep “eigen value” dan “eigen vector” dijelaskan dalam bentuk persoalan fisis [4]. Terdapat matriks M yang menggambarkan bentuk deformasi membran akibat mendapat perlakuan fisik

yakni ditarik, dirotasi dan ditekan terhadap titik asal. Maka setelah deformasi, titik awal pada bidang membran (x,y) berubah lokasi menjadi (X,Y).



Gambar 3. Interpretasi fisis eigen value dan eigen vector

Interpretasi dari bentuk tersebut seperti terlihat pada Gambar 3 adalah “eigen value” yang terbentuk dari matriks transformasi memberi informasi tentang besar nilai deformasi bidang membran yang terjadi, sedangkan “eigen vector” matriks transformasi memberi informasi tentang arah pergerakan yang terjadi akibat perubahan deformasi bidang membran tersebut [4].

## III. TEKNIK PENGENALAN

### A. Representasi Citra Wajah

Wajah sebagai ukuran biometrik merupakan kerangka fisis yang dapat direpresentasikan sebagai sekumpulan informasi dalam hal ini berupa nilai intensitas citra wajah. Representasi dari n buah vektor/citra dapat dinyatakan sebagai sehimpunan n buah sampel berdimensi d berupa  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ .

Permasalahan bisa timbul jika representasi citra wajah tersebut cukup besar untuk diolah dalam sistem komputasi, untuk mengatasi masalah tersebut salah satunya adalah dengan menggunakan metode *Principal Component Analysis* (PCA). Metode PCA bekerja dengan prinsip bahwa jumlah kuadrat jarak antara  $\mathbf{x}_0$  (vektor representasi) dengan vektor yang akan direpresentasikan  $\mathbf{x}_k$  sekecil mungkin [5].

### B. Teknik PCA

Teknik PCA dimulai ketika Harold Hotelling [5] mengajukan teknik baru yang bertujuan untuk mengurangi jumlah dimensi sebuah ruang yang biasanya direpresentasikan oleh variabel statistik  $x_1, x_2, \dots, x_n$ , dimana variabel-variabel tersebut biasanya saling berkorelasi satu dengan yang lain. Akibat konsekuensi sifat tersebut, lalu muncul pertanyaan apakah ada himpunan variabel baru yang sifatnya relatif sama dengan variabel yang diwakili sebelumnya, dan diharapkan himpunan variabel baru tersebut memiliki jumlah dimensi (variabel) yang lebih sedikit dari dimensi sebelumnya. Dikemudian

hari transformasi dimensi ini dikenal dengan nama Transformasi Hotelling atau Transformasi Karhunen Loeve dan tekniknya dikenal dengan nama *Principal Component Analysis (PCA)* yang “basis” kerjanya berdasarkan prinsip statistika dari citra [6].

Teknik *Principal Component Analysis (PCA)* terdiri atas beberapa langkah yaitu :

1. Jika diasumsikan terdapat himpunan p buah citra pelatihan yang direpresentasikan kedalam matriks  $X = [X_1, X_2, \dots, X_p]$  dengan dimensi tiap citra adalah q (baris x kolom), maka dapat dibentuk matriks data pelatihan Y dengan orientasi vektor kolom sebagai berikut:

$$Y = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1q} \\ x_{21} & x_{22} & \dots & x_{2q} \\ \dots & \dots & \dots & \dots \\ x_{p1} & x_{p2} & \dots & x_{pq} \end{bmatrix}$$

dengan  $X_{ij}$  menunjukkan nilai intensitas dari piksel citra pelatihan ke-i variabel (dimensi) ke-j.

2. Matriks Y kemudian di-normalisasi menjadi:

$$Z = \begin{bmatrix} \frac{x_{11}}{\sqrt{x_{11}^2 + x_{12}^2 + \dots + x_{1q}^2}} & \frac{x_{12}}{\sqrt{x_{11}^2 + x_{12}^2 + \dots + x_{1q}^2}} & \dots & \frac{x_{1q}}{\sqrt{x_{11}^2 + x_{12}^2 + \dots + x_{1q}^2}} \\ \frac{x_{21}}{\sqrt{x_{21}^2 + x_{22}^2 + \dots + x_{2q}^2}} & \frac{x_{22}}{\sqrt{x_{21}^2 + x_{22}^2 + \dots + x_{2q}^2}} & \dots & \frac{x_{2q}}{\sqrt{x_{21}^2 + x_{22}^2 + \dots + x_{2q}^2}} \\ \dots & \dots & \dots & \dots \\ \frac{x_{p1}}{\sqrt{x_{p1}^2 + x_{p2}^2 + \dots + x_{pq}^2}} & \frac{x_{p2}}{\sqrt{x_{p1}^2 + x_{p2}^2 + \dots + x_{pq}^2}} & \dots & \frac{x_{pq}}{\sqrt{x_{p1}^2 + x_{p2}^2 + \dots + x_{pq}^2}} \end{bmatrix}$$

3. Hitung rata-rata vektor citra  $\mu_j = \frac{1}{p} \sum_{i=1}^p Z_{pj}$ .

Lalu bentuk matriks rata-rata  $\mu = [\mu_1 \ \mu_2 \ \dots \ \mu_q]$ .

4. Hitung selisih vektor citra dengan rata-rata vektor citra  $A = Z - \mu$ .
5. Dari matriks A langkah ke-4, hitung matriks total scatter berupa matriks kovarian yang bersifat *real* dan simetrik  $S = AA^T$  (berordo q x q).
6. Tentukan “*eigen vector*” V dan “*eigen value*” D dari matriks S pada langkah ke-6, dan urutkan *eigen vector* secara “*ascending order*”.
7. Tentukan dimensi ruang ciri yang akan digunakan berdasarkan aturan:

$$m = \min q \left\{ \begin{array}{l} \sum_{i=1}^q D_i \\ \sum_{i=1}^p D_i \end{array} \right\} > \theta$$

8. Reduksi “*vector eigen*” V menjadi “*vector eigen*” sesuai dengan nilai m yang akan dikenal dengan istilah “*eigenfaces*”.
9. Transformasi vektor citra menjadi vektor ciri  $AV^T$ .

#### IV. HASIL DAN PEMBAHASAN

Berbeda dengan penelitian sebelumnya yang menggunakan metode *Nearest Feature Line* untuk menguji klasifikasi pola pada “*half face*” [7], penelitian ini menggunakan metode *Nearest Feature Line* untuk menguji tingkat akurasi pengenalan wajah dengan ekspresi yang berbeda-beda, dengan pose wajah dibatasi diambil dari tampak depan dan kualitas foto dibatasi pada dimensi citra 35x42 pixel. Data sampel yang digunakan sebanyak 44 buah yang terbagi kedalam 4 kelompok orang, dengan berbagai macam ekspresi dan hambatan visual diantaranya: berkacamata, berbayang pada satu sisi, normal, tersenyum, merengut, memejamkan mata, tertawa dan mengedipkan satu mata. Komputasi dijalankan pada bermacam data wajah yang telah direduksi dimensinya menggunakan metode *Principal Component Analysis* dengan tujuan untuk mencari ciri-ciri terpenting dari wajah. Dengan harapan dapat mengurangi beban lamanya waktu komputasi akibat besarnya dimensi ciri wajah yang diolah.



Gambar 4. Data objek 1 (A1...A11)



Gambar 5. Data objek 2 (B1...B11)



Gambar 6. Data objek 1 (C1...C11)

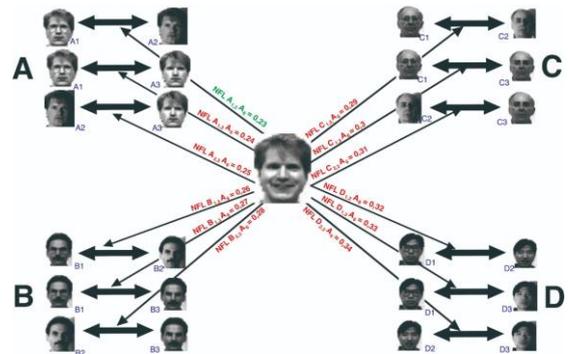


Gambar 7. Data objek 1 (D1...D11)

A. Teknik Klasifikasi Menggunakan Metode NFL

Teknik klasifikasi dilakukan dengan terlebih dulu membentuk garis ciri Eigen pada setiap vektor (gambar) acuan yang terdapat pada basis data seperti yang ditunjukkan pada Gambar 4, 5, 6 dan 7. Jika dimisalkan menggunakan masing-masing 3 vektor (gambar) acuan untuk setiap sampel/kelas A, B, C dan D, maka akan terdapat 3 garis ciri Eigen untuk setiap sampel/kelas uji A, B, C dan D. Selanjutnya vector (gambar) uji yang akan diklasifikasi akan dihitung jarak NFL-nya terhadap setiap garis ciri Eigen yang terbentuk sebelumnya yakni 12 garis ciri Eigen seperti ditunjukkan pada Gambar 8. Penentuan klasifikasi suatu vector (gambar) uji masuk kedalam kelas A, B, C atau D ditentukan berdasarkan jarak terpendek yang diperolehnya, dalam kasus Gambar 8, Gambar uji A6 terklasifikasi sebagai kelompok/kelas wajah A dikarenakan jarak NFL-nya paling pendek terbentuk dengan garis ciri A1 dengan A2 dan hasil ini dianggap benar karena vektor(gambar) ciri A6 memang berada pada kelas kelompok wajah A.

Penelitian ini menguji 7 jenis percobaan seperti yang ditunjukkan pada tabel 3, dimana hasil optimal didapatkan pada kasus 6 buah vektor acuan dengan 5 buah vektor uji untuk setiap masing-masing kelas wajah A, B, C dan D.



Gambar 8. Teknik klasifikasi NFL

B. Sampel Implementasi Model 5 Vektor Acuan, 6 Vektor Uji

Tabel 1. Vektor acuan (5) dan vektor uji (6)

Vektor Acuan (File)	Vektor Uji (File)
SA1...SA5	SA6...SA11
SB1...SB5	SB6...SB11
SC1...SC5	SC6...SC11
SD1...SD5	SD6...SD11

Hasil perhitungan jarak terdekat dan ketepatan klasifikasi dari sampel; 5 vektor acuan dan 6 vektor uji diilustrasikan pada Tabel 2.

Tabel 2. Hasil pengujian jarak sampel Tabel 1

File Uji	Kategori	Jarak
SA\SA006.txt	termasuk SA 0 4	0.234973
SA\SA007.txt	termasuk SC 1 4	0.159983
SA\SA008.txt	termasuk SA 2 4	0.414572
SA\SA009.txt	termasuk SA 0 4	0.343121
SA\SA010.txt	termasuk SA 0 4	0.129782
SA\SA011.txt	termasuk SA 0 4	0.224511
SB\SB006.txt	termasuk SB 0 2	0.196233
SB\SB007.txt	termasuk SB 0 2	0.309947
SB\SB008.txt	termasuk SB 0 2	0.267249
SB\SB009.txt	termasuk SB 1 2	0.231585
SB\SB010.txt	termasuk SD 0 4	0.479127
SB\SB011.txt	termasuk SB 0 2	0.20002
SC\SC006.txt	termasuk SC 1 2	0.319528
SC\SC007.txt	termasuk SC 2 3	0.122509
SC\SC008.txt	termasuk SC 1 2	0.343282
SC\SC009.txt	termasuk SC 1 2	0.366861
SC\SC010.txt	termasuk SC 2 3	0.161312
SC\SC011.txt	termasuk SC 2 3	0.134812
SD\SD006.txt	termasuk SA 0 3	0.571568
SD\SD007.txt	termasuk SA 0 3	0.609441

SD\SD008.txt	termasuk SA 0 3	0.609441
SD\SD009.txt	termasuk SD 0 4	0.491857
SD\SD010.txt	termasuk SA 1 3	0.555666
SD\SD011.txt	termasuk SA 1 3	0.572898

Ket: warna merah menandakan salah klasifikasi

Tingkat akurasi: 70,83 %

### C. Hasil Eksperimen Lengkap

Dari Tabel 3, Terlihat metode NFL menunjukkan hubungan tingkat keakuratan dengan jumlah vektor acuan yang digunakan, dimana semakin banyak jumlah vektor acuan meningkatkan akurasi pengenalan. Hal ini disebabkan semakin banyak vektor acuan maka jumlah garis vektor uji dengan vektor acuan juga semakin banyak, yang pada akhirnya berkontribusi pada pilihan garis vektor terpendek yang bisa dipilih.

Tabel 3. Tingkat akurasi pengenalan wajah (bebas ekspresi) dengan metode NFL

<b>Jumlah Vektor Acuan</b>	3	4	5	6	7	8	9
<b>Jumlah Vektor Uji</b>	8	7	6	5	4	3	2
<b>Akurasi</b>	50 %	67.9 %	70.8 %	90 %	87.5 %	83.3 %	87.5 %

## V. SIMPULAN

Kesimpulan yang dapat ditarik dari pengenalan wajah dengan ekspresi yang berbeda-beda menggunakan metoda NFL adalah:

- Semakin banyak vektor acuan, semakin tinggi tingkat akurasi pengenalan wajah.
- Secara rata rata tingkat akurasi metode NFL mendekati tingkat akurasi metode *Back Propagation* [8] pada penelitian lain [3] dimana secara rata-rata diatas 85%.
- Untuk model dan karakter penelitian yang sama, waktu komputasi metode NFL jauh

lebih singkat dibandingkan metode *Back Propagation*, dimana untuk pelatihan *Back Propagation* dengan 50.000 *epoch* membutuhkan waktu rata-rata 446 detik, sedangkan metode NFL proses perhitungan jarak hingga klasifikasi membutuhkan waktu rata-rata 15 detik, hal ini dikarenakan metode NFL tidak mengalami proses pembelajaran berulang seperti halnya *Back Propagation* [8], melainkan cukup menghitung jarak terpendek [1] antara *vector eigen* acuan dengan vektor uji. Sehingga jika ukuran waktu menjadi pertimbangan maka metode NFL sangat layak dijadikan metode pengenalan pola yang handal.

Kedepannya penelitian ini dapat dikembangkan dengan meningkatkan tingkat akurasi pengenalan, dengan jalan memperbaiki metode pencarian proyeksi jarak terdekat dengan model matematika yang baru atau hasil modifikasi metode NFL.

## UCAPAN TERIMA KASIH

Penulis mengucapkan banyak terimakasih kepada Fakultas Teknologi Industri Universitas Trisakti atas bantuan moril dan materil hingga selesainya penelitian yang penulis lakukan.

## DAFTAR PUSTAKA

- [1] L. Zhao, W. Qi, S. Z. Li, S.-Q. Yang, and H. J. Zhang, "Key-frame extraction and shot retrieval using nearest feature line (NFL)," in Proceedings of the 2000 ACM workshops on Multimedia - MULTIMEDIA '00, 2000, pp. 217–220, doi: 10.1145/357744.357942.
- [2] W. C. Moon, Todd K., Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Prentice-Hall, Inc, 2000.
- [3] I. Mardianto, "Sistem Pengenalan Wajah 3Dimensi Dengan Automatisasi Penentuan Konstanta Bobot Sudut Pandang Berdasar Pada Penerapan Metode Proyeksi Jarak Terdekat dan Representasi Ciri Pada Ruang Eigen," 2004.
- [4] Mary L. Boas, *Mathematical Methods in the Physical Sciences*, 3rd ed. Wiley, 2005.
- [5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. 2001.
- [6] Yu-Jin ZHANG, *Image Engineering. Processing, Analysis and Understanding*. Cengage Learning Asia Pte Ltd, 2009.
- [7] Q. Feng, L. Yan, T. S. Pan, and J. S. Pan, "Nearest feature line and extended nearest feature line with half face," in Advances in Intelligent Systems and Computing, 2014, doi: 10.1007/978-3-319-01796-9\_8.
- [8] L. Fausett, *Fundamentals of Neural Network, Architectures Algorithms and Applications*. Prentice Hall, 1994.

# Komparasi Metode *Multilayer Perceptron* (MLP) dan *Long Short Term Memory* (LSTM) dalam Peramalan Harga Beras

Steven Sen<sup>1</sup>, Dedy Sugiarto<sup>2</sup>, Abdul Rochman<sup>3</sup>

<sup>1,3</sup>Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Trisakti, Jakarta, Indonesia  
stev.strife@gmail.com

<sup>2</sup>Program Studi Sistem Informasi, Fakultas Teknologi Industri, Universitas Trisakti, Jakarta, Indonesia

Diterima 21 April 2020

Disetujui 17 Juni 2020

**Abstract**—Rice is one of the main commodities in Indonesian society. The main problem with rice nationally is inflation of rice prices. Therefore, this research predicts the price of rice using Multilayer Perceptron (MLP) artificial neural network architecture and deep learning: Long Short Term Memory (LSTM) to anticipate these problems. The data used in this study are real data on rice prices during 2016 - 2019 obtained from PT. Food Station. The total dataset is 1307 with the distribution of 1123 as data train and 184 as test data. The final results obtained in this study are LSTM superior to MLP, with the value of Root Mean Square Error (RMSE) training data: 0.49 RMSE loss value of test data is 0.27. The most optimal LSTM model from 3 tests conducted, is a model with a number of hidden layers = 16 and epochs = 150 times.

**Index Terms**—Artificial Neural Network, Deep Learning, Forecasting Time Series, Long Short Term Memory (LSTM), Multilayer Perceptron (MLP)

## I. PENDAHULUAN

Beras merupakan komoditi pokok bagi masyarakat Indonesia sampai saat ini. Prediksi terhadap harga beras sangat diperlukan untuk mewujudkan stabilisasi harga bahan pokok guna tercapainya ketahanan pangan nasional, terutama pada masa pandemi Covid-19 seperti sekarang. Tentunya pada masa sulit ini, diharapkan harga-harga pangan tetap stabil untuk menunjang kehidupan masyarakat, terutama beras. Data harga beras yang dihimpun pemerintah tiap tahunnya, termasuk dalam tipe data *time series*.

Banyak metode *machine learning* tersedia untuk memprediksi data *time series*, baik yang memiliki arsitektur *shallow* seperti : *Principal Component Analysis* (PCA), *Isomap*, dan *Support Vector Machine* (SVM), maupun arsitektur *deep* seperti: *Convolutional Neural Network* (CNN), *Multilayer Perceptron* (MLP), dan *Long Short Term Memory* (LSTM).

*Multilayer Perceptrons* (MLP) adalah arsitektur *perceptron* yang paling banyak digunakan untuk jaringan saraf. Lapisan *perceptron* digabungkan dan membentuk arsitektur *multilayer*, hal ini memberikan kompleksitas yang diperlukan dari pemrosesan

jaringan saraf. Dengan menambahkan lebih banyak lapisan dan lebih banyak neuron per lapisan, akan meningkatkan spesialisasi model untuk melatih data [1].

LSTM merupakan sebuah arsitektur RNN [2] yang memiliki *memory cell*. Dengan *memory cell*, arsitektur LSTM dapat bekerja lebih baik dibanding jaringan saraf rekuren biasa, karena memiliki kemampuan untuk mengingat informasi untuk periode waktu yang lebih lama, sehingga menjadikannya algoritma yang lebih baik untuk prediksi data berjenis *time series* [3].

Penelitian ini bertujuan untuk melakukan prediksi dan komparasi model terhadap data *time series* harga beras jenis IR-64 kualitas III selama 24 bulan pada tingkat grosir yang didapatkan dari *website* PT.Food Station. Metode yang akan digunakan adalah komparasi antara jaringan saraf tiruan yaitu *Multilayer Perceptron* (MLP) dan *deep learning* dengan arsitektur *Long Short Term Memory* (LSTM) yang akan dibangun dengan environment berbasis bahasa *python*. Hal ini termasuk sebagai inovasi baru dalam perbandingan kedua model jaringan syaraf tiruan dalam bidang *deep learning* peramalan data *time series* terhadap permasalahan yang dibahas, mengingat semakin besarnya data dan kompleksnya kebutuhan akan pengolahan data.

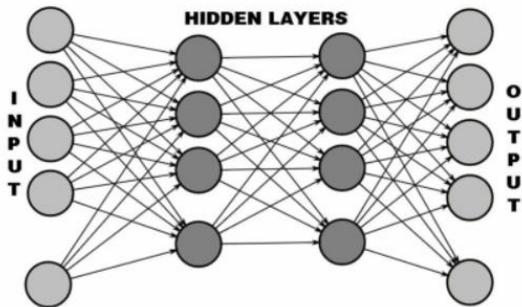
Proses dalam penelitian ini adalah mencari nilai prediksi dengan akurasi terbaik antara model MLP dan LSTM, serta membuat plot perbandingan data test dengan data hasil prediksi. Tingkat keakuratan dari tiap model, akan diukur menggunakan algoritma *Root Mean Squared Error* (RMSE) sesuai skenario yang penulis buat.

## II. TINJAUAN PUSTAKA

### A. *Multilayer Perceptron* (MLP)

*Multilayer* merupakan bentuk lapisan *perceptron* yang digabungkan, dengan menambahkan lebih banyak layer dan neuron tiap layer. *Multilayer*

Perceptron (MLP) sendiri merupakan arsitektur yang paling banyak digunakan untuk jaringan saraf.

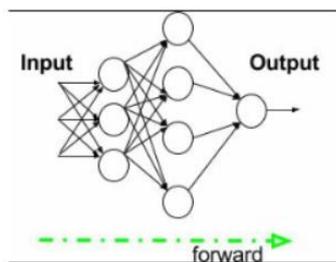


Gambar 1. Deep neural network MLP [1]

Pemrosesan dari *layer input* ke *hidden layer* dan kemudian ke lapisan *output* adalah disebut *forward propagation*. Jumlah (*input \* bobot*) + bias diterapkan pada setiap *layer* dan kemudian nilai fungsi aktivasi disebarkan ke *layer* berikutnya. Lapisan selanjutnya bisa berupa *hidden layer* lain atau *output layer* [1].

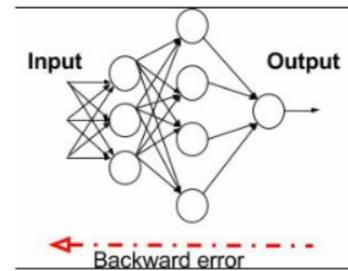
MLP dilatih dari data pelatihan melalui proses yang disebut *backpropagation*. Proses ini dapat digambarkan sebagai cara untuk memperbaiki kesalahan secara progresif segera setelah terdeteksi.

Pada awalnya, semua bobot ditetapkan secara acak. Kemudian jaringan diaktifkan untuk setiap input dalam set pelatihan: nilai disebarkan ke depan (*forward propagation*) dari tahap input melalui tahap tersembunyi ke tahap *output* di mana prediksi dibuat.



Gambar 2. Feed forward

Karena nilai *real* yang diamati dalam set pelatihan diketahui, maka memungkinkan untuk menghitung kesalahan yang dibuat dalam prediksi. Proses kerja utama dalam *backtracking* adalah melakukan alur kembali dari *output* menuju *input* dengan menggunakan algoritma pengoptimal yang tepat, seperti *gradient descent*, untuk menyesuaikan bobot (*weight*) jaringan saraf dengan tujuan mengurangi kesalahan.



Gambar 3. Backpropagation

Proses *feed propagation* dari *input* ke *output* dan *backpropagation* diulang beberapa kali sampai kesalahan mencapai nilai di bawah ambang batas yang telah ditentukan [4].

### B. Recurrent Neural Network (RNN)

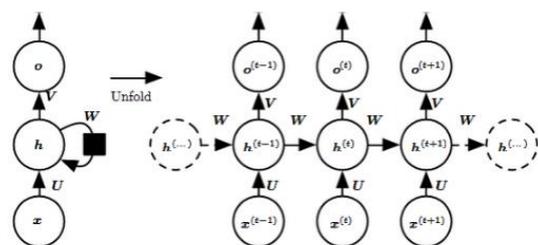
*Recurrent Neural Network* (RNN) adalah jenis arsitektur *Neural Network* (jaringan saraf tiruan) dimana *output* dari *hidden layer* akan menjadi input bagi pemrosesan berikutnya. Ciri RNN adalah melakukan prediksi terhadap data tidak hanya menggunakan input pada satu waktu saja, akan tetapi juga membutuhkan masukan dari input sebelumnya, sehingga antar *input* saling berhubungan dan menggunakan hubungan tersebut untuk memberikan informasi ke seluruh *hidden layer* yang ada dalam RNN. RNN memiliki memori yang berisikan hasil rekaman informasi yang dihasilkan sebelumnya. Persamaan yang terbentuk adalah *forward propagation* dalam RNN (1), *hidden layer* (2), *output gate* (3), dan target (4).

$$a^{(t)} = b + Wh^{(t-1)} + Ux^{(t)} \tag{1}$$

$$h^{(t)} = \tanh(a^{(t)}) \tag{2}$$

$$o^{(t)} = c + Vh^{(t)} \tag{3}$$

$$\hat{y}^{(t)} = \text{softmax}(o^{(t)}) \tag{4}$$



Gambar 4. Recurrent Neural Network dan formulasinya [5]

Satu masalah pada arsitektur RNN adalah masalah menghilangnya gradien (*vanishing problem*) pada proses *backward pass* [4]. Cara yang cukup populer

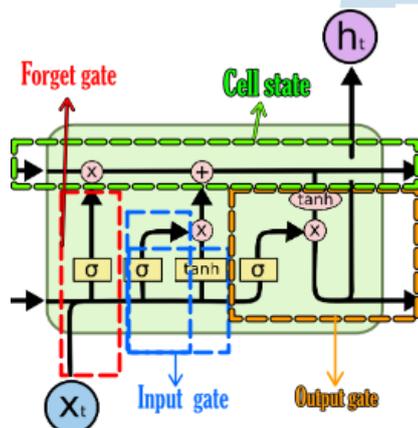
untuk mengatasi *vanishing problem* adalah dengan menggunakan unit RNN yaitu *Long Short Term Memory* (LSTM) [6].

### C. Long Short Term Memory (LSTM)

LSTM adalah jenis RNN khusus yang bekerja lebih baik dalam praktiknya, dikarenakan adanya pembaruan dari persamaan dan adanya dinamika *backpropagation* didalamnya [7]. Arsitektur *Long Short Term Memory network* (LSTM), seperti terlihat pada gambar 3, memiliki gate yang berfungsi untuk menghapus maupun menambah informasi yaitu *forget gate*, *output gate*, dan *input gate*. [8].

Persamaan yang terbentuk dari *forget gate* (1), *input gate* (2), *output gate* (3), lapisan tanh menciptakan vektor nilai kandidat baru (4), *cell state* (5), dan *hidden layer* (6) adalah:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) & (1) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) & (2) \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) & (3) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) & (4) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t & (5) \\ h_t &= o_t * \tanh(C_t) & (6) \end{aligned}$$



Gambar 5. Unit dari LSTM [6]

- *Forget Gate*: secara kondisional memutuskan apakah suatu informasi harus dibuang atau tidak dari pemrosesan.
- *Input Gate*: secara kondisional untuk menentukan sebuah masukan akan ditambahkan ke dalam memori *cell state* saat itu atau tidak [9].
- *Cell state* ini berfungsi sebagai memori atau ingatan untuk sebuah layer. Nilai suatu *cell state* dimanipulasi menggunakan sebuah sistem gerbang.
- *Output Gate*: secara kondisional memutuskan apa yang akan dihasilkan berdasarkan input dan memori blok [6].

### D. Deep Learning with Keras

Keras adalah *framework deep learning* untuk *Python* yang menyediakan cara mudah untuk mendefinisikan dan melatih hampir semua jenis model *deep-learning*. Keras awalnya dikembangkan untuk para peneliti, dengan tujuan memungkinkan eksperimen berjalan lebih cepat. Sekarang, Keras adalah *framework* terpopuler di Kaggle dan berbagai kompetisi *deep learning* dunia [10]. *Framework* ini mendukung 2 bahasa *data science* yang paling populer yaitu: *R* and *Python* [11]. Kelebihan yang ditawarkan adalah dengan munculnya fitur-fitur : membuat *code* berjalan mulus pada CPU atau GPU, *user-friendly API* yang membuatnya mudah untuk membuat prototipe sebuah *deep-learning*, dan mendukung model RNN [10].

### III. METODE PENELITIAN

Dalam membangun model LSTM ini, membutuhkan beberapa tahapan proses. Proses-proses tersebut terdiri dari proses *preprocessing* data, inialisasi parameter, *training* jaringan LSTM, dan melakukan uji terhadap data *testing*. Dalam pembangunan sistem, dataset yang didapatkan diolah terlebih dahulu dengan menggunakan teknik normalisasi *min max scaling*. Dilakukan inialisasi pada setiap parameter, setelah itu dilakukan *training* pada arsitektur jaringan saraf yang dibuat sesuai dengan parameter yang telah ditetapkan. Selanjutnya dilakukan uji pada model LSTM, hasil dari proses *training* terhadap data *testing*. Proses tersebut terus diulang hingga mendapatkan model dengan akurasi yang baik.

#### A. Tahap Preprocessing Data

Dataset yang digunakan adalah data *real* harga beras dari tanggal 1 Januari 2016 s.d 31 Januari 2020. Untuk meminimalkan *error*, dilakukan normalisasi pada dataset dengan mengubah data aktual menjadi nilai dengan *range* interval [0,1]. Teknik normalisasi yang digunakan adalah dengan *min-max scaling*.

Dataset kemudian dibagi dalam data *train* dan tes. Data *train* adalah data dari tanggal 1 Januari 2016 sampai 31 Juli 2019. Sedangkan data tes adalah sisanya, yaitu data dari tanggal 1 Agustus 2019 s.d 31 Januari 2020.

Tabel 1. Pembagian data

Data Latih	Data Tes
1245 Data	184 Data
01-01-2016 s.d 31-07-2019	01-08-2019 s.d 31-01-2020
Total data : 1429	

Data *train* dan tes di-*reshape* menjadi 3 dimensi.

## B. Inisialisasi Parameter

### B.1 Model MLP

Setelah *dataset* dilakukan *preprocessing*, dan selanjutnya menentukan inisialisasi parameter dasar. Pada MLP yang dibutuhkan antara lain:

Tabel 2. Parameter model MLP

Paramater	Jumlah
Total layer	5
Input layer	1 layer (64 neuron)
Hidden layer	3 layer (128,128,8 neuron)
Output layer	1 neuron
Jumlah epochs	100

### B.2 Model LSTM

Inisialisasi parameter pada LSTM yang akan dibangun yaitu:

Tabel 3. Parameter model LSTM

Paramater	Jumlah
Total layer	4
Hidden layer	1 layer, 64 neuron
Jumlah LSTM cell (input layer)	128 neuron
Output layer	1 neuron
Jumlah epochs	100

## C. Tahap Analisis dan Perancangan Model

### C.1 Model MLP

Tahap analisis dan perancangan model dilakukan setelah data melewati tahapan *preprocessing*. Model yang digunakan di penelitian ini, yaitu MLP.

- Digunakan *dataset train* untuk melatih model MLP. Pada tahap awal, model dibangun dengan memanggil fungsi *sequential*.
- Kemudian ditambahkan 1 buah *layer input* MLP dengan *dense* 64 neuron, *input dimension* = 1, dan *activation* = 'relu'.
- Ditambahkan 3 *hidden layer* dengan jumlah neuron masing-masing 128, 128, dan 8.
- Pada *layer* terakhir, yaitu *output layer* dengan 1 buah neuron.
- Digunakan *loss function* yaitu RMSE dan *optimizer* Adam.
- Terakhir model dilatih dengan menggunakan fungsi *fit*, yang menggunakan parameter data *train*.

### C.2 Model LSTM

Tahap analisis dan perancangan model dilakukan setelah data melewati tahapan *preprocessing*. Model

yang digunakan kali ini yaitu LSTM dengan detail sebagai berikut:

- Digunakan *dataset train* untuk melatih model LSTM. Pada tahap awal, model dibangun dengan menggunakan fungsi *sequential*.
- Kemudian dibuat model dengan 1 buah *layer* LSTM sebagai *input layer*, dengan dimensi (1,1), 1 *hidden layer* dengan 64 neuron, dan *output* 1 neuron.
- Digunakan *loss function* yaitu RMSE dan *optimizer* Adam.
- Terakhir model dilatih dengan menggunakan fungsi *fit*, yang menggunakan parameter data *train*.

Tahap analisis dan perancangan model dilakukan setelah data melewati tahapan *preprocessing*. Model yang digunakan di penelitian ini, yaitu MLP.

## D. Testing

Model yang telah dilatih pada proses *training*, akan diuji dengan menggunakan data *test* dari proses *preprocessing* data. Model akan menghasilkan nilai prediksi harga beras. Dilakukan juga proses perhitungan nilai RMSE pada hasil data *train* dan data *test*.

### D.1 Skenario Pengujian MLP

Skenario pengujian dalam penelitian ini adalah menganalisa dampak tiap parameter terhadap akurasi yang didapatkan. Parameter yang diuji berupa jumlah neuron pada *input layer* dan besarnya *epoch* maksimum untuk menghasilkan bobot MLP yang optimal. Berikut adalah nilai parameter yang akan diuji:

- Jumlah neuron *input* : 64,128, 256
- Jumlah *Epoch* : 100, 150, 200

Dilakukan observasi sebanyak 3 kali. Hal ini dilakukan karena tidak cukup dilakukan observasi hanya sekali saja, karena hasil yang didapatkan bisa saja kebetulan baik ataupun buruk.

### D.2 Skenario Pengujian LSTM

Skenario pengujian dalam penelitian ini adalah menganalisa dampak tiap parameter terhadap akurasi yang didapatkan. Parameter yang diuji berupa jumlah neuron pada *hidden layer* dan besarnya *epoch* maksimum untuk menghasilkan bobot LSTM yang optimal. Berikut adalah nilai parameter yang akan diuji:

- Jumlah neuron *hidden* : 64,128, 256
- Jumlah *Epoch* : 100, 150, 200

Dilakukan observasi sebanyak 3 kali. Hal ini dilakukan karena tidak cukup dilakukan observasi hanya sekali saja, karena hasil yang didapatkan bisa saja kebetulan baik ataupun buruk.

#### E. Membandingkan Data Hasil Prediksi dan Plotting Data

Tahap akhir yaitu membandingkan data hasil prediksi model LSTM dengan data aktual dalam sebuah tabel. Nilai train dan prediksi juga divisualisasikan dalam bentuk grafik plot.

### IV. HASIL DAN PEMBAHASAN

Berdasarkan data *real* harga beras yang terdapat pada FoodStation, dapat dilihat bahwa terdapat kenaikan harga setiap awal tahun dibanding bulan lainnya terutama pada tahun 2018. Lonjakan tertinggi terdapat pada awal tahun 2018 yang mencapai harga lebih dari 12.500, diikuti lonjakan awal tahun 2019 yang juga cukup tinggi.



Gambar 6. Visualisasi data beras tanggal 1 Januari 2016 s.d 31 Januari 2020

Sedangkan harga beras cenderung menurun pada pertengahan tahun. Harga beras pada pertengahan tahun 2019 sampai awal tahun 2020 cenderung stabil. Dapat ditarik informasi bahwa pada awal tahun dan akhir tahun harga beras cenderung meningkat.

#### A. Model MLP

Tabel 4. Kombinasi uji coba MLP

Uji Coba ke -	Jumlah neuron hidden layer	epochs	Nilai RMSE Train	Nilai RMSE Tes
1	64	100	52.52	21.14
2	128	150	95.38	39.32
3	256	200	88.98	50.86

Berdasarkan lampiran Tabel 4, menunjukkan hasil bahwa model MLP dengan jumlah neuron *input layer*= 64 dan *epochs*= 100, menghasilkan nilai RMSE yang lebih kecil dibandingkan dengan percobaan lainnya. Bisa dikatakan bahwa percobaan pertama lebih optimal dibanding dengan kedua percobaan lainnya. Jumlah *input layer* mengolah data *input* dan menghubungkannya dengan neuron *output*. Tidak ada aturan pasti mengenai jumlah *hidden neuron* yang paling optimal dalam memprediksi data

*time series* contohnya harga beras ini, maka dilakukanlah percobaan untuk mengetahui model paling optimal.

#### B. Model LSTM

Tabel 5. Kombinasi uji coba LSTM

Uji Coba ke -	Jumlah neuron hidden layer	epochs	Nilai RMSE Train	Nilai RMSE Tes
1	64	100	0.52	0,35
2	128	150	0.49	0.27
3	256	200	0.59	0.37

Berdasarkan lampiran Tabel 5, menunjukkan hasil bahwa model LSTM dengan jumlah neuron *hidden layer*= 128 dan *epochs*= 150, menghasilkan nilai RMSE yang lebih kecil dibandingkan dengan percobaan lainnya. Bisa dikatakan bahwa percobaan kedua lebih optimal dibanding dengan kedua percobaan lainnya. *Hidden layer* mengolah neuron *input* dan menghubungkannya dengan neuron *output*, sehingga jumlah *hidden neuron* akan menentukan nilai *output* yang dihasilkan oleh unit LSTM.

Tabel 6. Perbandingan data aktual dan prediksi MLP

Tanggal	Harga Aktual	Nilai Prediksi	Nilai akurasi RMSE
		MLP	MLP
01 Jan 2016	10000	9968,594	0,937
02 Jan 2016	10000	9968,594	0,937
03 Jan 2016	10000	9968,594	0,937
04 Jan 2016	10100	10034,286	4,945
05 Jan 2016	10200	10253,814	4,362
=====	=====	=====	=====
27 Jan 2020	11225	11194,379	0,914
28 Jan 2020	11225	11194,379	0,914
29 Jan 2020	11250	11212,434	1,121
30 Jan 2020	11250	11212,434	1,121
31 Jan 2020	11250	11212,434	1,121

Dari tabel perbandingan data aktual dan prediksi MLP, menunjukkan bahwa MLP memprediksi nilai harga dengan rentang yang cukup besar terhadap harga aktual. Nilai *error* RMSE cukup besar menandakan bahwa model ini kurang efektif dalam mengolah data *time series* harga beras.

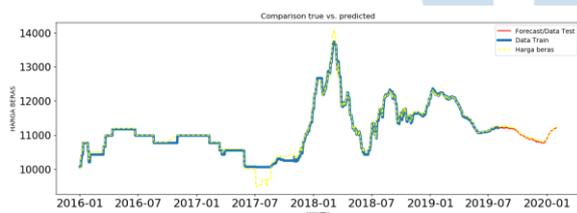
Hasil *training* model LSTM, maka akan dihasilkan nilai prediksi harga dan *loss* pada bulan selanjutnya yaitu sebagai berikut:

Tabel 7. Perbandingan data aktual dan prediksi LSTM

Tanggal	Harga Aktual	Nilai Prediksi	Nilai akurasi RMSE
		LSTM	LSTM
01 Jan 2016	10000	9999,547	0,014
02 Jan 2016	10000	9999,547	0,014
03 Jan 2016	10000	9999,547	0,014
04 Jan 2016	10100	10099,568	0,013
05 Jan 2016	10200	10199,624	0,011
===== 27 Jan 2020	===== 11225	===== 11227,099	===== 0,063
28 Jan 2020	11225	11227,099	0,063
29 Jan 2020	11250	11252,168	0,065
30 Jan 2020	11250	11252,168	0,065
31 Jan 2020	11250	11252,168	0,065

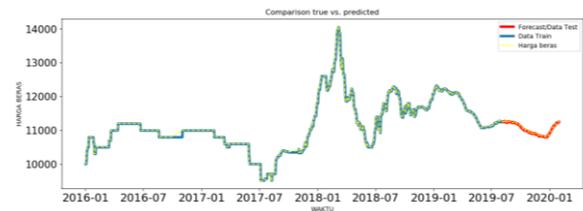
Tabel perbandingan data aktual dan prediksi LSTM, menunjukkan bahwa LSTM berhasil memprediksi nilai harga dengan optimal dan mendekati harga aktual. Nilai *error* RMSE yang sangat kecil menandakan bahwa model ini cukup efektif dalam mengolah data *time series* harga beras. Terkait nilai MSE yang optimal, adalah tergantung dari data dan model yang dibangun.

Hasil *plotting data train* dan tes pada model MLP yang dibangun.

Gambar 7. Visualisasi data *train* dan tes MLP

Visualisasi data *train* (*biru*), *tes* (*merah*), dan data harga aktual (*kuning*) menunjukkan bahwa harga beras cenderung berubah secara signifikan terutama pada tahun 2018. Menurut *training* dan *testing* MLP, harga 6 bulan kedepan cenderung fluktuatif yaitu dari 1 Agustus 2019 s.d 31 Januari 2020 akan berkisar antara 10.600 – 11.100. Garis harga prediksi dan aktual tidak nampak ada perbedaan, karena perbedaan nilai yang sangat kecil.

Hasil *plotting data train* dan *tes* pada model LSTM yang dibangun.

Gambar 8. Visualisasi data *train* dan tes LSTM

Visualisasi data *train* (*biru*), *tes* (*merah*), dan data harga aktual (*kuning*) menunjukkan bahwa harga beras cenderung tidak berubah secara signifikan seperti pada tahun 2018. Menurut prediksi LSTM, harga 6 bulan kedepan yaitu dari 1 Agustus 2019 s.d 31 Januari 2020 akan berkisar antara 10.300 – 10.700. Garis harga prediksi dan aktual tidak nampak ada perbedaan, karena perbedaan nilai yang sangat kecil.

## V. SIMPULAN

Prediksi terhadap data *time series* harga beras pada PT. FoodStation dirancang dengan menggunakan perbandingan antara dua model *Multilayer Perceptron* (MLP) dan arsitektur *Long Short Term Memory* (LSTM). Sebagai data percobaan, digunakan data harga beras tanggal 1 Januari 2019 s.d 31 Juli 2019. Model MLP dan LSTM dibangun dengan *library* Keras dan menggunakan *loss function* yaitu RMSE.

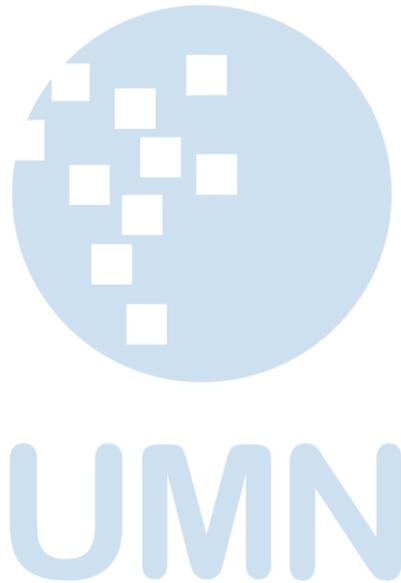
Pada tahap *forecasting*, digunakan data 6 bulan terakhir harga beras sebagai data *testing* yang menghasilkan output prediksi harga dari kedua model. Perbandingan hasil *output*, menghasilkan kesimpulan bahwa model LSTM lebih akurat jika dilihat dari rentang dengan harga aktual dan nilai RMSE-nya yang lebih kecil dibandingkan MLP. Pada akhirnya, didapatkan nilai *loss* RMSE pada LSTM, yaitu *training data score*: 0.49 RMSE, *test data score*: 0.27 RMSE, dan hasil grafik plot yang menunjukkan bahwa tidak ada perbedaan signifikan antara hasil *train* dan *testing* dengan data aktual.

Penelitian ini, diharapkan dapat menjadi referensi pilihan metode untuk pengembangan penelitian lainnya dan dikembangkan lagi agar menghasilkan metode yang lebih akurat terutama dalam bidang *forecasting deep learning*.

## DAFTAR PUSTAKA

- [1] G. Ciaburro and B. Venkateswaran, *Neural network with R*, Packt Publishing, 2017.
- [2] W. Ahmed and M. B. , "The Accuracy of the LSTM Model for Predicting the S&P 500 index and the Difference Between Prediction and Backtesting," Degree Project in Technology, 2018.
- [3] H. Prasetyanwar and J. , "Peramalan Nilai Tukar IDR-USD Menggunakan Long Short Term Memory," e-Proceeding of Engineering : Vol.5, No.2 Agustus 2018 , vol. 5, p. 3820, 2018.
- [4] A. Gulli and S. P. , *Deep Learning with Keras*, Birmingham: Packt Publishing Ltd, 2017.

- [5] J. Nabi, "Recurrent Neural Networks (RNNs)," [Online]. Available: <https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85>.
- [6] C. Olah, "Understanding LSTM Networks," 27 August 2015. [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [7] A. Karpathy, "The Unreasonable Effectiveness of Recurrent Neural Networks," 21 May 2015.
- [8] J. Brownlee, "Time Series Prediction with LSTM Recurrent Neural Network in Python with Keras," 2016.
- [9] L. Zaman, S. S. and M. H. , "Analisis Kinerja LSTM dan GRU sebagai Model Generatif," vol. 8, p. 143, 2019.
- [10] F. Chollet, Deep Learning with Python, New York: Manning, 2018.
- [11] F. Chollet and J. J. Allaire, Deep Learning with R, vol. 1, Manning Publications, 2017.



# Peramalan Utilisasi Perangkat Jaringan dan *Bandwidth* dengan Metode *Holt-Winters* dan *Multilayer Perceptron*

Muhammad Taufiq<sup>1</sup>, Dedy Sugiarto<sup>2</sup>, Abdul Rochman<sup>3</sup>

<sup>1,3</sup> Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Trisakti, Jakarta, Indonesia  
m.taufiq064001800502@std.trisakti.ac.id

rochman@trisakti.ac.id

<sup>2</sup> Jurusan Sistem Informasi, Fakultas Teknologi Industri, Universitas Trisakti, Jakarta, Indonesia  
dedy@trisakti.ac.id

Diterima 22 April 2020

Disetujui 17 Juni 2020

**Abstract**—Network devices become an important medium for transferring data from one node to another node in the form of switches, routers or network security devices. The reliability of network devices must be maintained both in terms of device resources and bandwidth. The study was conducted by applying the Holt-Winters and Multilayer Perceptron (MLP) method to network device and bandwidth data utilization. The two methods are compared to assess which accuracy is better when applied to network device and bandwidth utilization data by calculating Root Mean Squared Error (RMSE) and Mean Absolute Percentage (MAPE). The results of the measurement of accuracy in the network device testing data, MLP produces a value of RMSE of 5,67 and MAPE of 2.34, and Holt-Winters produces a value of RMSE of 14.56 and MAPE of 2.95. For the results of the measurement of accuracy in the bandwidth testing data with MLP produces a value of RMSE of 0.13 and MAPE of 7.27, and Holt-Winters produces RMSE values of 2.59 and MAPE of 134.31. Based on the results of these measurements it is concluded that the MLP method has a smaller error value compared to the Holt-Winters method applied to network device and bandwidth utilization data with a span of 3 years historical data. Build web application design using Mysql as a Database Management System, Laravel as a web application framework, Pentaho for data integration and R programming for data analysis.

**Index Terms**—Framework Laravel, Holt-Winters Method, Multi Layer Perceptron, Network Devices and Bandwidth, Pentaho Data Integration, R

## I. PENDAHULUAN

Perangkat – perangkat jaringan seperti *switch*, *router* atau perangkat keamanan jaringan harus dikelola dengan baik dan benar. Dalam melakukan tugasnya perangkat jaringan harus memiliki kehandalan dan ketersediaan sumber daya untuk mengirim dan menerima data. Pemanfaatan terhadap data tersebut tentunya akan memudahkan perusahaan untuk membuat perencanaan dan pengalokasian dana untuk kebutuhan teknologi *Information Technology*

(IT) seperti perangkat jaringan dan *bandwidth*. Dibutuhkan analisis data terhadap data-data historis utilisasi perangkat dan *bandwidth* dengan menerapkan metode-metode tertentu, sehingga perencanaan menjadi tepat. Pada perusahaan untuk menentukan kebutuhan kedepannya, belum banyak memaksimalkan penggunaan data *mining* untuk menganalisa dan meramalkan data-data utilisasi perangkat jaringan dan *bandwidth*. Untuk itu perlu dilakukan analisis data-data historis pada tahun-tahun sebelumnya lalu meramalkan data yang akan muncul ke depannya.

Penelitian terkait jaringan komputer sebelumnya “Peramalan Kebutuhan *Bandwidth* Iub Jaringan UMTS dan HSDPA Menggunakan *Fuzzy Inference System* dan *Time Series*” dengan hasil *Mean Absolute Percentage Error* (MAPE) sebesar 20,95 % sedangkan metode FTS memiliki rata-rata MAPE sebesar 11,14% [1]. Penerapan algoritma *Holt-Winters* sudah diteliti dengan judul penelitian “Prediksi Kekeringan Menggunakan Metode *Holt-Winters* Studi Kasus di Wilayah Kabupaten Boyolali” dengan hasil ramalan dengan nilai kesalahan ramalan terkecil bernilai 4,27 [2]. Pendekatan selanjutnya dengan menggunakan *Neural Network*, *Learning* merupakan kemampuan utama yang mendasar dari *Neural Network*. Salah satu jenis *Neural Network* adalah *Multilayer Perceptron* (MLP), Metode *Multilayer Perceptron* pernah diteliti dengan judul penelitian “Peramalan harga beras IR64 kualitas III menggunakan metode *Multilayer Perceptron*, *Holt-Winters* dan *Auto Regressive Integrated Moving Average*” yang menghasilkan nilai akurasi RMSE 5.67 pada metode MLP [3].

Kedua metode tersebut nantinya akan digunakan untuk meramalkan data-data terkait utilisasi jaringan, dari hasil ramalan kedua metode tersebut, Peneliti membandingkan penggunaan metode *Holt-Winters* dan *Multilayer Perceptron*, dengan menilai tingkat

kesalahan pada kedua metode tersebut, kemudian data-data utilisasi jaringan dan *bandwidth* dilakukan integrasi data dengan *Pentaho Data Integration* dari format *spreadsheet* ke bentuk ekstensi *sql* pada tabel *Database Management System MySQL*. Untuk memudahkan dalam pengguna dalam meramalkan data, pengoptimalan model dan melihat hasil ramalan, penulis membangun aplikasi berbasis web dengan menerapkan konsep *Model View Controller* dengan menggunakan *Framework Laravel*.

## II. METODOLOGI PENELITIAN

### A. Teknik Pengumpulan dan Analisis Data

Analisis data dapat digambarkan sebagai suatu proses yang terdiri dari beberapa langkah di mana data mentah ditransformasikan dan diproses untuk menghasilkan visualisasi data dan dapat membuat prediksi dengan model yang dipilih berdasarkan data yang dikumpulkan [4]. Langkah-langkah yang dilakukan dalam analisis data perangkat jaringan dan *bandwidth*:

#### 1. Data Set dan Ekstraksi Data

Penulis memperoleh data histori utilisasi perangkat jaringan dan *bandwidth* perusahaan yang ekstraksi dengan bantuan *network monitoring server* baik yang diekstrak melalui aplikasi maupun diekstrak dengan manual. Data perangkat jaringan yang telah diekstrak merupakan data rata-rata *memory* setiap bulannya dari beberapa titik perangkat jaringan di Kantor Pusat Jakarta, mulai dari Januari 2017 hingga Desember 2019. Untuk data *bandwidth* yang telah diekstrak merupakan data penggunaan rata-rata *bandwidth* yang diterima setiap bulannya pada titik-titik perangkat *router* pada kantor di daerah, mulai dari Januari 2017 hingga November 2019.

#### 2. Data Preparation

Data-data utilisasi perangkat jaringan dan *bandwidth* memiliki variabel yang bervariasi. Pada Perangkat Jaringan terdapat variabel *Average Unreachable*, *Average CPU usage*, *Average Response time*, *Average Memory usage* sedangkan pada *bandwidth* terdapat variabel *Average Transmit*, *Average Receive*, *Max Transmit*, *Max Received*. Penulis memilih *Average Memory usage* dan *Average Receive* untuk diteliti. Dilakukan penormalan terhadap data yang menjadi objek penelitian seperti satuan dan rentang bilangan. Pengoptimalan data sesuai format data *time series* yang dapat dieksekusi oleh metode peramalan. Setiap data utilisasi perangkat jaringan dan *bandwidth* lalu dibagi menjadi dua bagian yaitu data *training* dan data uji, dimana persentase data *training* adalah 85% dan data uji 15%. Data *training* akan digunakan

untuk pembelajaran pola dengan menggunakan *Holt-Winters* dan *Multilayer Perceptron* dan data uji akan dilakukan pengukuran akurasi. Berikut data set utilisasi perangkat jaringan pada Tabel 1 dan data set *bandwidth* pada Tabel 2, dengan penjelasan warna kolom sebagai berikut:

	Data Training
	Data Uji

Tabel 1. Data set utilisasi perangkat jaringan

No.	Bulan	Tahun		
		2017	2018	2019
1	Januari	401.79	413.10	414.81
2	Februari	400.29	413.10	410.75
3	Maret	406.06	416.31	421.43
4	April	410.97	417.37	417.80
5	Mei	411.40	419.29	417.37
6	Juni	404.56	416.52	418.65
7	Juli	410.97	416.95	423.35
8	Agustus	411.61	418.87	431.25
9	September	410.97	418.01	436.59
10	Oktober	415.02	417.16	444.70
11	November	413.96	417.16	447.26
12	Desember	414.38	418.23	

Tabel 2. Data set bandwidth

No.	Bulan	Tahun		
		2017	2018	2019
1	Januari	0.45	1.02	1.04
2	Februari	0.54	0.76	0.92
3	Maret	0.41	0.51	1.32
4	April	0.29	0.69	1.19
5	Mei	0.17	0.51	1.07
6	Juni	0.15	0.37	1.16
7	Juli	0.15	0.64	1.27
8	Agustus	0.33	0.64	1.31
9	September	0.65	0.91	1.50
10	Oktober	0.73	0.93	1.61
11	November	0.76	1.05	
12	Desember	0.84	1.16	

#### 3. Metode Peramalan Holt-Winters

Metode *Holt-Winters* menggunakan tiga parameter pemulusan yaitu konstanta pemulusan, parameter untuk tren, dan parameter untuk musiman. Metode *Holt-Winters* memiliki keunggulan pada *Exponential smoothing* bersifat sederhana, intuitif dan mudah dipahami. Sederhana tetapi sangat bermanfaat untuk peramalan pendek dari data *time series* yang panjang.

Dilihat dari fluktuasi musiman pada data utilisasi perangkat jaringan dan *bandwidth* yang bersifat variasi penulis menggunakan model *Multiplicative* jika besarnya variasi musiman meningkat dengan peningkatan tingkat rata-rata deret waktu[5].

Persamaan dasar untuk model *Holt-Winters Multiplicative*:

Nilai Pemulusan keseluruhan

$$L_t = \alpha \frac{Y_t}{S_{t-s}} + (1 - \alpha)(L_{t-1} + b_{t-1})$$

Nilai Pemulusan komponen *trend*

$$b_t = \beta(L_t - L_{t-1}) + (1 - \beta)b_{t-1}$$

Nilai Pemulusan komponen musiman

$$S_t = \gamma \frac{Y_t}{L_t} + (1 - \gamma)S_{t-s}$$

Nilai Ramalan

$$F_{t+m} = (L_t + b_t m)S_{t-s+m}$$

Data utilisasi perangkat jaringan dan *bandwidth* setiap bulannya akan diproses pada persamaan model *Holt-Winters Multiplicative* dengan 3 parameter yang bisa disesuaikan yaitu ( $\alpha$ ) konstanta pemulusan, ( $\beta$ ) parameter *trend*, dan ( $\gamma$ ) parameter musiman. Untuk mengetahui pola pada data *training* baik utilisasi perangkat jaringan dan *bandwidth*. Model yang terbentuk dari penyesuaian parameter konstanta pemulusan, parameter *trend* dan parameter musiman. Menghasilkan nilai parameter pemulusan  $\alpha = 0.471$ ,  $\beta = 0$  dan  $\gamma = 1$  pada utilisasi perangkat jaringan dan dihasilkan nilai parameter pemulusan  $\alpha = 0.881$ ,  $\beta = 0$  dan  $\gamma = 1$  pada *bandwidth*.

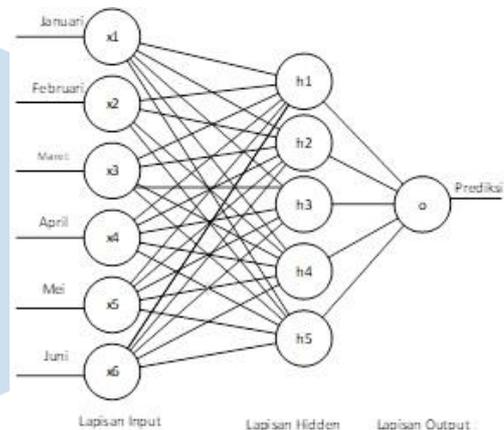
Selanjutnya dengan nilai parameter yang sudah ditetapkan, penulis melakukan peramalan 5 bulan kedepannya yaitu bulan juli, agustus, september, oktober, november pada data utilisasi perangkat jaringan dan bulan juli, agustus, september, oktober pada data *bandwidth*.

#### 4. Metode Peramalan *Multilayer Perceptron*

Cara kerja MLP ini mengadopsi jaringan saraf mahluk hidup, sehingga terkenal bagus karena proses pembelajaran yang mampu dilakukan secara terarah. MLP dikembangkan dari *Perceptron* dimana untuk memecahkan

permasalahan yang tidak bisa dipecahkan oleh *perceptron*. Arsitektur MLP:

- Lapisan *Input* merupakan lapisan dimana setiap unit nya menyimpan data, pada lapisan *input* ini akan disimpan data utilisasi perangkat jaringan dan data *bandwidth*.
- Lapisan *Hidden* merupakan lapisan diantara lapisan *input* dan lapisan *output*, lapisan *hidden* dapat terdiri dari 1 lapisan atau lebih dari 1, dilakukan pembobotan dari sejumlah *input* dan bias dimasukkan ke fungsi aktivasi melalui fungsi *transfer* untuk menghasilkan *output*.
- Lapisan *Output* merupakan hasil kalkulasi atau pembelajaran dari lapisan sebelumnya.



Gambar 1. Arsitektur MLP

*Backpropagation* merupakan suatu algoritma pembelajaran yang dapat memperkecil tingkat sebuah *error* dengan bobotnya yang disesuaikan berdasarkan perbedaan dari *output* dan atau target. Di dalam *backpropagation* terdapat fungsi aktivasi [6].

Fungsi aktivasi digunakan agar nilai *threshold* membatasi nilai keluaran agar selalu dalam batas nilai yg ditetapkan. Fungsi yang digunakan adalah *sigmoid biner* memiliki nilai pada *range* 0 sampai 1, didefinisikan:

$$y = f(x) = \frac{1}{1 + e^{-x}}$$

#### 5. Algoritma *Backpropagation*

Inisialisasi semua bobot, lapisan *input* akan diisi bobot nilai data sesuai bulan secara sekuensial, lapisan tersembunyi akan diisi dengan nilai acak.

a. Tahap Propagasi Maju

Tiap unit *input* menerima sinyal dan meneruskannya ke unit *hidden* setelahnya. Tiap unit *hidden* mendapatkan nilai dengan mengalikan nilai dari tiap unit *input* dengan bobot menuju pada unit *hidden* lalu menghitung keluaran di unit *hidden* dengan fungsi aktivasi. Selanjutnya unit *output* akan mendapatkan nilai dengan mengalikan nilai dari tiap unit *hidden* dengan bobot menuju unit *output*, lalu menghitung keluaran unit di unit *output* dengan fungsi aktivasi.

b. Tahap Propagasi Mundur

Mengitung faktor  $\delta$  unit keluaran berdasarkan *error* di setiap unit keluaran. Hitung *factor*  $\delta$  unit tersembunyi berdasarkan eror di setiap lapisan tersembunyi.

c. Tahap Perubahan bobot

Menghitung semua perubahan bobot. Perubahan bobot garis yang menuju ke lapisan keluaran dan perubahan bobot garis yang menuju ke lapisan *hidden*.

B. Pengujian dan Pengukuran Akurasi Peramalan

Pada tahapan pengujian dilakukan pengukuran akurasi terhadap ramalan dengan:

- *Root Mean Squared Error (RMSE)*

Pengukuran dengan menggunakan RMSE dengan menghitung nilai akar rata-rata kuadrat kesalahan dari suatu peramalan, dimana  $X_i$  adalah nilai target aktual,  $F_i$  adalah nilai keluaran prediksi, dan  $n$  adalah jumlah baris data.

$$RMSE = \sqrt{\left(\frac{\sum_{i=1}^n (X_i - F_i)^2}{n}\right)}$$

- *Mean Absolute Percentage Error (MAPE)*

Metode ini dihitung dengan menggunakan kesalahan yang absolut pada setiap periode lalu dibagi dengan nilai penelitian atau observasi yang nyata untuk periode itu. Kemudian, merata-ratakan kesalahan persentase absolut tersebut.

$$MAPE = \frac{\sum \frac{|e_i|}{X_i} \times 100\%}{n} = \frac{\sum \frac{|X_i - F_i|}{X_i} \times 100\%}{n}$$

C. Pengkodean Program

C.1 Pengkodean pada R

Pada tahapan ini dilakukan pengkodean program untuk implementasi dari metode algoritma peramalan dan mendapat akurasi dari tingkat peramalan. Untuk metode peramalan dengan *Holt-Winters* dapat dilihat pada *Listing 1* dan untuk metode peramalan dengan *Multilayer Perceptron* dapat dilihat pada *Listing 2*. Data set akan dimasukan ke dalam program (1)berisi tentang data-data utilisasi perangkat jaringan dan *bandwidth*. Kemudian akan dibuat pemisahan data untuk data *training* sebanyak 85% dari data set(3). Pendefinisian data *training* yaitu dimulai dari data awal hingga data batas *training* (4). Didefinisikan juga data uji yang dimulai dari data batas *training* hingga jumlah data yang diinput(5). Pendefinisian variabel ramal dengan nilai kosong dan juga variabel *history* yang diisi dengan data *training* (6,7). Dilakukan peramalan dengan fungsi pengulangan dengan batas banyaknya data pada data *testing* (8). Masing-masing data *training* disesuaikan dengan model peramalannya, penyesuaian parameter *alfa* ( $\alpha$ ), *beta* ( $\beta$ ) dan *gamma* ( $\gamma$ ) pada *Holt-Winters*, dan penyesuaian lapisan *input*, *hidden*, dan *output* pada MLP (9). Dilakukan pemodelan untuk menemukan model yang paling optimal pada data *training* dengan menyesuaikan parameter yang ada pada model (10). Dilakukan peramalan ke  $x$  dengan menggunakan model yang paling optimal (11), kemudian nilai ramalan ke  $x$  diappend ke dalam variabel ramal (12). Data *testing* ke  $x$  lalu dimasukkan kedalam variabel observasi(13). kemudian nilai observasi di masukan kedalam variabel *history* untuk (14). Nilai yang ada pada variabel observasi di *append* kedalam variabel *history*, sebagai penanda batas terakhir permalan pada pengulangan *for* selanjutnya. Dilakukan pengulangan (9...14) sampai nilai  $x$  sama dengan nilai banyaknya data pada data *testing*. Setelah data-data ramalan didapat, lalu dilakukan pengukuran akurasi peramalan dengan menggunakan *Mean Squared Error* dengan menggunakan data *testing* dan data ramal (16), setelah itu dilakukan akar untuk menghasilkan nilai RMSE (17). Dilakukan perhitungan akurasi juga menggunakan *Mean Absolute Percent Error* (MAPE) (18).

Listing 1. Pseudo code peramalan *Holt-Winters*

```
#Proses Data
(1)Inputs: data
(2)Output: RMSE dan MAPE dari peramalan data

#Data Preparation
(3)batas <- length(data) * 0.85
(4)training <- series[0...batas]
(5)testing <- series[batas...length(data)]
(6)ramal<- empty
(7)history <- training
#Model dan Peramalan
(8) For each x in range(length(testing)) do
(9)  model <- Holt-Winters(training, alfa, beta,
    gamma, Multiplicative)
(10) model_fit<- model.fit()
(11) nramal <- model_fit.forecast()
```

```
(12) ramal.append(nramal)
(13) observasi <- testing[x]
(14) history.append(observasi)
(15) end for
      #Pengukuran Akurasi
(16) MSE = mean_squared_error(testing,ramal)
(17) RMSE = sqrt(MSE)
(18)MAPE=mean_absolute_percent_error(testing,
ramal)
```

Listing 2. Pseudo code peramalan MLP

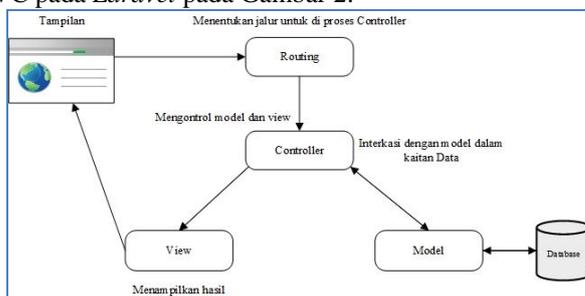
```
#Proses Data
(1)Inputs: data
(2)Output: RMSE dan MAPE dari peramalan
data
      #Data Preperation
(3)batas <- length(data) * 0.85
(4)training <- series[0..size]
(5)testing <- series[batas...length(data)
(6)ramal<- empty
(7)history <- training

      #Model dan Peramalan
(8)For each x in range(length(testing)) do
(9)model <- mlp(training)
(10)model fit<- model.fit()
(11) nramal <- model_fit.forecast()
(12) ramal.append(nramal)
(13) observasi <- test[t]
(14) history.append(observasi)
(15) end for

      #Pengukuran Akurasi
(16)MSE = mean_squared_error(test,ramal)
(17)RMSE = sqrt(MSE)
(18)MAPE = mean_absolute_percent_error
```

C.2 Laravel dan Model View Controller (MVC)

Framework Laravel menggunakan kerangka kerja pengembangan web MVC yang ditulis dalam bahasa pemrograman PHP. Dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya perawatan yang berkelanjutan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi kita dengan menyediakan sintaksis mudah yang jelas dan serangkaian fungsionalitas inti yang akan menghemat waktu dalam implementasi [7]. Berikut konsep arsitektur MVC pada Laravel pada Gambar 2.



Gambar 2. Konsep arsitektur MVC Laravel [8]

C.3.1 Model

Model biasanya berhubungan dengan data dan interaksi ke database atau *web service*. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk *file* teks, file XML maupun *web service*. Dengan adanya model, dengan mudah melakukan *create, read, update* dan *delete* pada tabel *cdevice*, namun pada penelitian ini hanya digunakan untuk menampilkan/*Read* pada tabel *cdevice* yang digunakan untuk menampilkan tabel perangkat yang di hubungkan nantinya ke dalam *ip address* tabel utilisasi harian, kemudian dilakukan visualisasi.

- Pada jendela *command* dilakukan pembuatan model pada *Laravel* yaitu:

```
php artisan make:model cdevice -m
```

- Kemudian dilakukan pengeditan pada direktori `htdocs\NetworksControllerv5\database\migration\s\2019__131221_create_cdevices_table.php` seperti yang terlihat pada Gambar 3.

```
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateCdevicesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('cdevices', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('address');
            $table->string('nama');
            $table->string('tipe');
            $table->string('manufacturer');
            $table->string('mac');
            $table->string('status');
            $table->string('kategori');
            $table->integer('jumlahport');
            $table->integer('rateport');
            $table->integer('portup');
            $table->integer('portuppenasional');
            $table->integer('adminportup');
            $table->integer('jumlahdevice');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('cdevices');
    }
}
```

Gambar 3. Kode program *create\_cdevices\_table.php*

- Selanjutnya digunakan fitur *migration* pada *Laravel* sehingga dimungkinkan untuk membuat dan memodifikasi tabel yang ada pada basisdata sesuai dengan kerangka model yang telah dibuat pada proses sebelumnya.

```
php artisan migrate
```

C.3.2 View

Bagian *presentation logic* merupakan bagian penting dalam aplikasi. Pada suatu aplikasi web,

bagian ini biasanya berupa file template *HyperText Markup Language*, yang diatur oleh *controller.view* bertugas untuk menerima dan merepresentasikan data hasil dari model dan *controller* kepada pengguna aplikasi. Pada aplikasi web disediakan halaman untuk melakukan peramalan dan menampilkan hasil ramalan data.

### C.3.3 Controller

Ketika pengguna aplikasi melakukan *request* data dengan mengklik *link*, maka *controller* akan melakukan pemanggilan ke model untuk mendapatkan data yang sesuai, dan kemudian mempersiapkan data tersebut untuk ditampilkan. *File controller* di letakkan pada folder `/app/Http/Backend/AnalisisController.php` dengan kode program yang dapat dilihat pada Gambar 4.

```
public function forecastingview($address)
{
    return view("backend.analisis.forecastingview", [$address]);
}
public function forecastingviewmlp($address)
{
    return view("backend.analisis.forecastingviewmlp", [$address]);
}
public function forecastingviewmlpht($address)
{
    return view("backend.analisis.forecastingviewmlpht", [$address]);
}
```

Gambar 4. Kode program pada AnalisisController

### C.3.4 Routing

*Routing* merupakan suatu proses untuk mengarahkan suatu permintaan atau item agar sampai pada tujuan. Fungsi *routing* pada *Laravel* terletak pada direktori *web.php* dengan kode program pada aplikasi *web* yang dapat dilihat pada Gambar 5.

```
Route::get('/backend/analisis/forecastingview/{id}', 'Backend\AnalisisController@forecastingview');
Route::get('/backend/analisis/forecastingviewmlp/{id}', 'Backend\AnalisisController@forecastingviewmlp');
Route::get('/backend/analisis/forecastingviewmlpht/{id}', 'Backend\AnalisisController@forecastingviewmlpht');
```

Gambar 5. Kode pada web.php

## C.3 Integrasi Data

Setelah data di-*cleansing*, tahap selanjutnya melakukan transformasi format data *spreadsheet*, penyesuaian terhadap struktur nama kolom dan pengisian data kosong pada *data set* serta bagaimana mengubahnya dalam format *structured query language* sehingga data dapat di load kedalam MySQL bersama dengan *records*nya. Berikut skenario *transform* data yang terlihat pada Gambar 6.



Gambar 6. Skenario *transform* data

### C.3.1 Microsoft Excel Input

Pada tahap ini dilakukan penarikan data utilisasi jaringan dan bandwidth ke dalam Lembar kerja *Pentho Data Integration*.

### C.3.2 Select Values

Pada tahap ini dilakukan perubahan *header* kolom pada tabel dengan nama sesuai dengan *field* nantinya pada MySQL, sehingga diperlukan transformasi terhadap *header* datanya. Penyesuaian dilakukan agar membuat penyeragaman struktur *field*, sehingga nantinya memudahkan dalam pengkodean dengan *Framework Laravel*.

### C.3.3 Replace In String

Pada tahap ini dilakukan perubahan nilai *record* yang tidak memiliki nilai atau memiliki string “-“ dengan mengganti nilai tersebut dengan “0”. Hal ini dilakukan agar tidak ada data yang kosong, untuk selanjutnya diberikan nilai 0.

### C.3.4 Table Output

Pada tahap ini dilakukan pembuatan *field* sesuai struktur tabel yang telah di proses sebelumnya dan ekstraksi *records* ke dalam bentuk sql. Berikut hasil integrasi data pada Gambar 7 dan *records* pada MySQL pada Gambar 8.

#	Stepname	Copied	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (1/s)	input/output	
1	Microsoft Excel Input	0	0	781	781	0	0	0	0	0	Finished	0.5s	1.616	-
2	Select values	0	781	781	0	0	0	0	0	0	Finished	0.5s	1.658	-
3	Replace in string	0	781	781	0	0	0	0	0	0	Finished	0.5s	1.637	-
4	Table output	0	781	781	0	781	0	0	0	0	Finished	0.6s	1.276	-

Gambar 7. Log hasil *transformation*

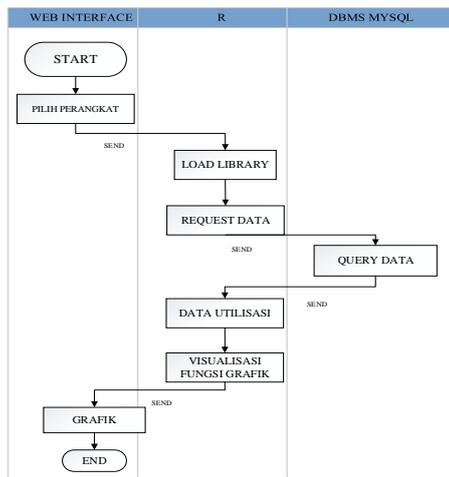
Year	Month	Bandwidth	Name	avgthroughput	maxthroughput	avgcpu	avgmemory	maxmemory	avgconnection	maxconnection	total	
2017	1	10.242.228.108	Interconnection Switch	0	0	10.0	11	24	11	0.343	43	0
2017	1	172.20.1.115	Git Squashed (Desktop)	0	0	10.0	10.00	24	10.0	0	0	0

Gambar 7. Records pada MySQL

## C.4 Visualisasi Data pada Antarmuka Web

Dalam melakukan visualisasi data diperlukan fungsi untuk membuat grafik, fungsi tersebut ada pada program *r* yang biasa digunakan adalah *plot*.

Untuk melakukan hal tersebut diperlukan skenario pada Gambar 9 untuk dapat menampilkan grafik data pada antarmuka *web*.



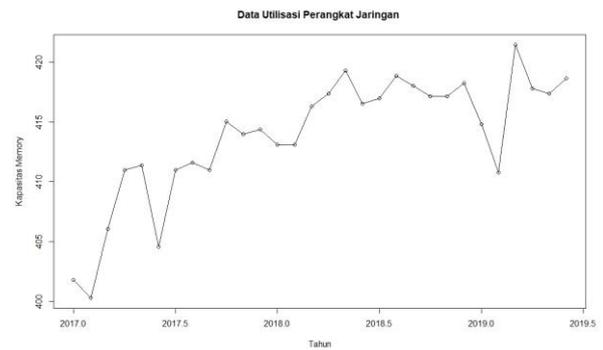
Gambar 9. Skenario visualisasi *plot*

Untuk dapat menampilkan output pada R pada antarmuka web dilakukan *editing* kode program seperti pada sintaks dibawah ini dengan bantuan fungsi *sink()*:

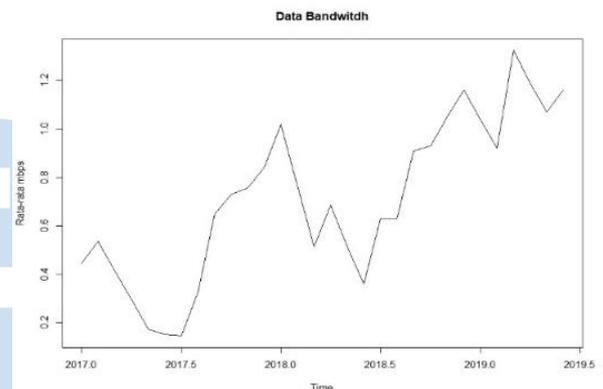
```
sink("htdocs/NetworksControllerv5/public/backend/ht
data.txt")
show(timeseries.all)
show(timeseries.training)
show(timeseries.uji)
fihht.timeseries
fihht.timeseries$fitted
htcast
sink() #stop sinking, =sink(NULL)
```

### III. HASIL DAN ANALISIS

Pada plot data *Average Memory* dan data *bandwidth* seperti yang terlihat pada Gambar 10 dan Gambar 11. Utilisasi perangkat jaringan dan *bandwidth* cenderung naik di bulan Juni-Desember. Hal tersebut dipengaruhi dengan pengaksesan konten-konten yang semakin banyak pada bulan tersebut. Konten-konten yang dimaksud adalah konten terkait kegiatan dalam implementasi Indikator Kinerja Utama (IKU) pada unit, yang semakin banyak menggunakan aplikasi-aplikasi elektronik *office* dan aplikasi proses bisnis. Hal yang paling rawan adalah interval oktober hingga desember, karena pada bulan tersebut penutupan anggaran tahunan.



Gambar 10. Data *training* utilisasi perangkat jaringan



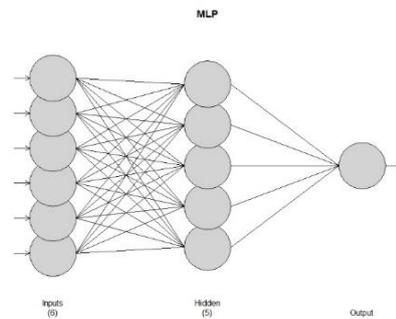
Gambar 11. Data *training* utilisasi perangkat *bandwidth*

Peramalan dengan menggunakan metode *Holt-Winters* pada data utilisasi perangkat jaringan menghasilkan nilai parameter pemulusan  $\alpha = 0,47$   $\beta = 0$ , dan  $\gamma = 1$ . Pada Gambar 12 terlihat model yang dihasilkan menyesuaikan pola pada data *training* utilisasi perangkat jaringan.

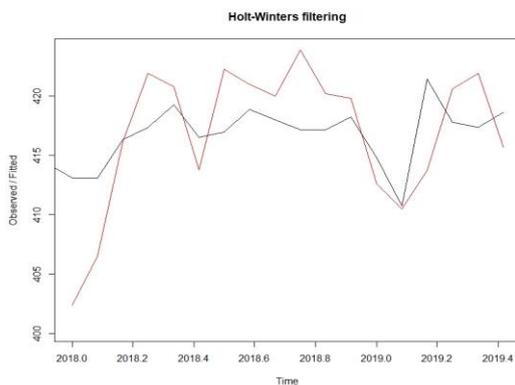
Peramalan dengan menggunakan metode *Holt-Winters* pada data *bandwidth* menghasilkan nilai parameter pemulusan  $\alpha = 0,88$ ,  $\beta = 0$ , dan  $\gamma = 1$ . Pada Gambar 13 terlihat model yang dihasilkan menyesuaikan pola pada data *training bandwidth*.



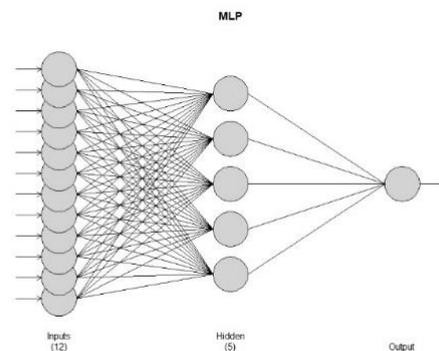
Gambar 12. Penyesuaian parameter data utilisasi perangkat jaringan dengan *Holt-Winters*



Gambar 14. Struktur model MLP pada data utilisasi perangkat jaringan



Gambar 13. Penyesuaian parameter data *bandwith* dengan *Holt-Winters*



Gambar 15. Struktur model MLP pada data *bandwidth*

Peramalan dengan menggunakan metode MLP pada data utilisasi perangkat jaringan menghasilkan model lapisan 6-5-1 pada Gambar 13 dimana 6 unit pada lapisan *input*, 5 unit lapisan *hidden*, dan 1 unit pada lapisan *output*. Data training utilisasi perangkat jaringan akan diformulasikan dengan model tersebut untuk mendapat nilai peramalan. Peramalan dengan menggunakan metode MLP pada data utilisasi perangkat jaringan menghasilkan model lapisan 12-5-1 pada Gambar 15 dimana 12 unit pada lapisan *input*, 5 unit lapisan *hidden*, dan 1 unit pada lapisan *output*. Data *training* utilisasi perangkat jaringan akan diformulasikan dengan model tersebut untuk mendapat nilai peramalan.

Hasil peramalan dengan menggunakan *Holt-Winters* dan MLP pada data utilisasi perangkat jaringan dan data *bandwidth* pada Tabel 3.

Tabel 3. Tabel hasil peramalan

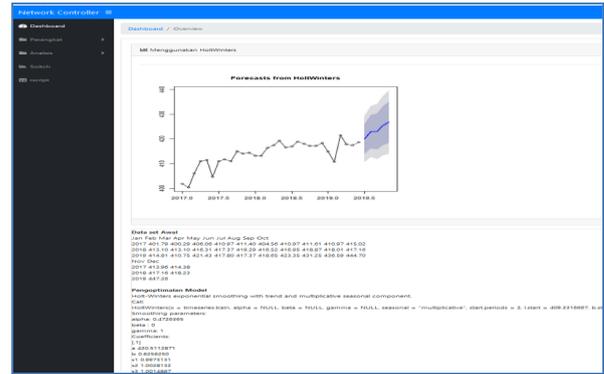
Data	Bulan	Data Aktual	Ramalan MLP	Ramalan Holt-Winters
Data Utilisasi Perangkat Jaringan	July	423.35	426.27	420.00
	Agust	431.25	420.96	422.95
	Sep.	436.59	421.85	423.01
	Okt	444.7	429.92	425.32
	Nov	447.26	438.76	426.77
Data Bandwidth	Juli	1.27	1.30	1.18
	Agust	1.31	1.25	2.28
	Sept.	1.50	1.35	4.71
	Okt	1.61	1.40	5.58

Dilakukan pengukuran akurasi peramalan pada data uji dengan menggunakan RMSE dan MAPE pada Tabel 4 akurasi peramalan.

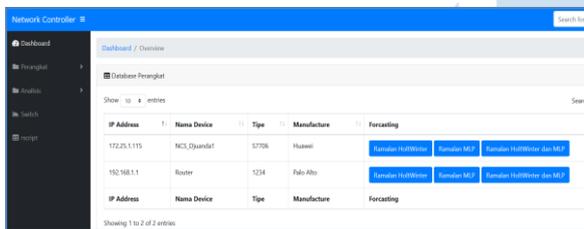
Tabel 4. Akurasi ramalan

Data	Metode Ramalan	RMSE	MAPE
Data Utilisasi Perangkat Jaringan	MLP	11.16	2.34
	HW	14.56	2.95
Data Bandwidth	MLP	0.13	7.27
	HW	2.59	134.31

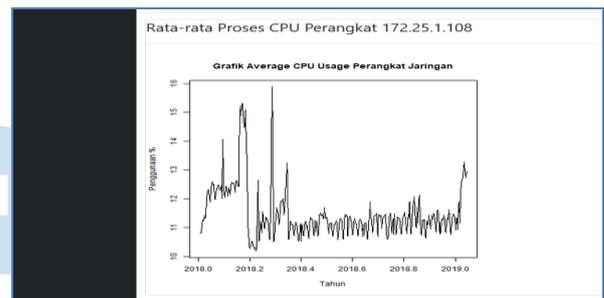
Hasil aplikasi berbasis web dengan *Framework Laravel* untuk peramalan dengan mengkoneksikan *Framework Laravel*, R dan MySQL menjadikan aplikasi dapat digunakan oleh user pengguna tanpa melakukan *editing* kode program pada R. Pada Gambar 16 dapat dilihat tampilan halaman pemilihan metode peramalan dan juga data yang ingin diramalkan, pada Gambar 17 dan 18 halaman hasil ramalan *Holt-Winters* dan MLP, sedangkan pada Gambar 19 visualisasi grafik data histori pada perangkat jaringan.



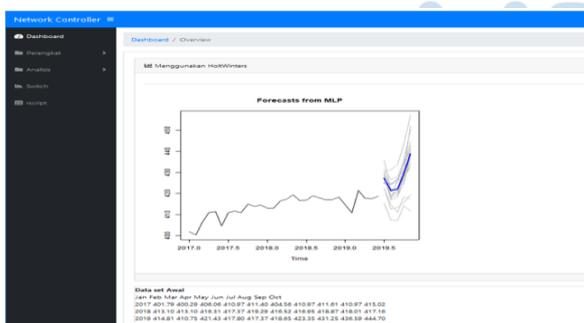
Gambar 18. Halaman hasil ramalan *Holt-Winters*



Gambar 16. Halaman pemilihan metode peramalan



Gambar 19. Halaman grafik utilisasi



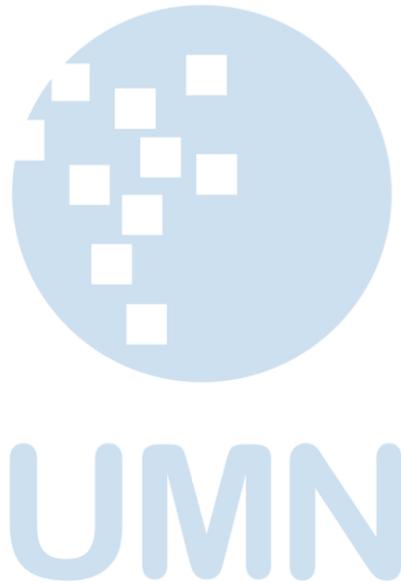
Gambar 17. Halaman hasil ramalan MLP

#### IV. SIMPULAN

Data utilisasi perangkat jaringan dan *bandwidth* mengalami peningkatan bulan-bulan pada akhir tahun. Berdasarkan pengukuran nilai akurasi menggunakan RMSE dan MAPE terhadap utilisasi perangkat jaringan dan *bandwidth* didapatkan bahwa metode MLP memiliki nilai kesalahan (*error*) lebih kecil dibandingkan dengan metode *Holt-Winters*. Perlu dilihat nilai ambang batas *memory* pada perangkat jaringan, jika ramalan data melebihi nilai ambang batas, maka menjadi perhatian untuk dievaluasi. Untuk *bandwidth* perlu dilihat kapasitas *bandwidth* yang disewa, jikalau hasil ramalan data pada *bandwidth* melebihi perlu dilakukan evaluasi penambahan kapasitas *bandwidth* jaringan. Aplikasi berbasis web dengan konsep MVC pada *Framework Laravel* dapat di visualisasikan hasilnya dalam bentuk grafik dengan dikoneksikan pada program R, hal tersebut dapat mempermudah dalam menggambarkan keadaan dari pemanfaatan sumber daya perangkat jaringan dan peramalan data. Penggunaan *Pentaho Data Integration* untuk melakukan ekstraksi data yang *records*-nya banyak dengan format *spreadsheet* ditransformasikan dan di-load ke dalam MySQL dapat mengefesienkan waktu pekerjaan.

## DAFTAR PUSTAKA

- [1] A. Wahyu, E. Yudaningsy, and S. H. Pramono, "Peramalan Kebutuhan Bandwidth Iub Jaringan UMTS dan HSDPA Menggunakan Fuzzy Inference System dan Time Series," *J. EECCIS*, 2014.
- [2] D. A. W. S. Putra, K. D. Hartomo, and R. Tanone, "Model Prediksi Kekeringan Menggunakan Metode Holt-Winters (Studi Kasus: Wilayah Kabupaten Boyolali)," *Indones. J. Comput. Model.*, 2018, doi: 10.24246/j.icm.2018.v1.i1.p36-41.
- [3] A. B. Ariwibowo, D. Sugiarto, I. A. Marie, and J. F. Agatha, *Peramalan harga beras IR64 kualitas III menggunakan metode Multilayer Perceptron, Holt-Winters dan Auto Regressive Integrated Moving Average*, vol. XI. 2019.
- [4] F. Nelli, *Python Data Analytics*. New York: Apress, 2015.
- [5] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction Time Series Analysis and Forecasting*, Second. New Jersey: John Wiley & Sons, Inc., 2016.
- [6] Y. A. Lesnussa, S. Latuconsina, and E. R. Persulesy, "Aplikasi Jaringan Saraf Tiruan Backpropagation untuk Memprediksi Prestasi Siswa SMA (Studi kasus: Prediksi Prestasi Siswa SMAN 4 Ambon)," *J. Mat. Integr.*, 2015, doi: 10.24198/jmi.v11.n2.9427.149-160.
- [7] S. McCool, *Laravel Starter*. Birmingham: Packt Publishing, 2012.
- [8] Chandra Shekhar, "General architecture of Laravel based application Chandra Shekhar Pandey." [Online]. Available: <http://www.cutehits.com/2017/06/general-architecture-of-laravel-based-application/>. [Accessed: 21-Apr-2020].



# Analisis Perbandingan Efisiensi Algoritma *Brute Force* dan *Divide and Conquer* dalam Proses Pengurutan Angka

Fenina Adline Twince Tobing<sup>1</sup>, James Ronald Tambunan<sup>2</sup>

<sup>1</sup>Program Studi Informatika, Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia  
fenina.tobing@umn.ac.id

<sup>2</sup>Program Studi Manajemen Informatika, AMIK Widya Loka, Medan, Indonesia  
jamesronaldtambunan@gmail.com

Diterima 30 April 2020

Disetujui 17 Juni 2020

**Abstract**—Perbandingan algoritma dibutuhkan untuk mengetahui tingkat efisiensi suatu algoritma. Penelitian ini membandingkan efisiensi dari dua strategi algoritma *sort* yang sudah ada yaitu *Brute Force* dan *Divide and Conquer*. Algoritma *Brute Force* yang akan diuji adalah *Bubble Sort* dan *Selection Sort*. Algoritma *Divide and Conquer* yang akan diuji adalah *Quick Sort* dan *Merge Sort*. Cara yang dilakukan dalam penelitian ini adalah melakukan tes dengan data sebanyak 50 sampai 100.000 untuk setiap algoritma. Tes dilakukan dengan menggunakan bahasa pemrograman *JavaScript*. Hasil dari penelitian ini adalah algoritma *Quick Sort* dengan strategi *Divide and Conquer* memiliki efisiensi yang baik serta *running time* yang cepat dan algoritma *Bubble Sort* dengan strategi *Brute Force* memiliki efisiensi yang buruk serta *running time* yang lama.

**Index Terms**—Algoritma, *Bubble Sort*, *Brute Force*, *Divide and Conquer*, Efisiensi, *Merge Sort*, *Quick Sort*, *Selection Sort*

## I. PENDAHULUAN

Algoritma merupakan hal yang penting dalam menyelesaikan sebuah permasalahan. Dibutuhkan sebuah metode yang tepat karena dapat mempengaruhi hasil yang diinginkan. Sebaik apapun algoritma, jika menghasilkan output yang salah, maka algoritma tersebut bukanlah algoritma yang baik. Sebuah algoritma yang baik adalah algoritma yang efektif, efisien, tepat sasaran dan terstruktur. Hal ini dapat diukur dari waktu eksekusi algoritma (*running time*) dan kebutuhan ruang memori (*memory space*) yang digunakan. Namun, kebutuhan waktu dan ruang dari suatu algoritma bergantung pada jumlah data (*input*) yang ingin diproses dan algoritma yang digunakan.

Algoritma *sorting* adalah salah satu algoritma dasar yang sering digunakan untuk menyelesaikan masalah pengurutan. Pengurutan data atau *sorting* merupakan salah satu jenis operasi penting dalam pengolahan data. Data yang sudah terurut memiliki beberapa

keuntungan. Selain mempercepat waktu pencarian, dari data yang terurut dapat langsung diperoleh nilai maksimum dan nilai minimum.

Dalam penyelesaian algoritma *sorting* atau algoritma pengurutan, ada beberapa metode yang dapat digunakan yaitu: *bubble sort*, *selection sort*, *insertion sort*, *merge sort*, *quick sort*, *heap sort*, *counting sort*, *radix sort*, dan *bucket sort* [1]. Banyaknya metode yang dapat digunakan dalam algoritma pengurutan ini belum tentu merupakan hal yang baik, dikarenakan tidak semua jenis metode memiliki hasil *running time* yang baik.

Pada penelitian sebelumnya [2], mengenai perbandingan efisiensi strategi *Brute Force* dengan algoritma *Bubble Sort* dan strategi *Divide and Conquer* dengan algoritma *Quick Sort*, didapatkan bahwa algoritma *Quick Sort* lebih efektif dan efisien dalam menangani masalah pengurutan. Namun, algoritma yang menggunakan strategi *Brute Force* dan juga *Divide and Conquer* tidak hanya *Bubble Sort* dan *Quick Sort* tetapi ada juga *Selection Sort* yang masuk dalam strategi *Brute Force* dan *Merge Sort* yang masuk dalam strategi *Divide and Conquer*.

Berdasarkan hal tersebut, akan dianalisa strategi algoritma mana yang efektif dan efisien dalam menyelesaikan masalah pengurutan angka dengan menggunakan strategi algoritma *Brute Force* dan *Divide and Conquer*.

## II. LANDASAN TEORI

### A. Algoritma *Sorting*

Algoritma *Sorting* adalah kumpulan langkah-langkah dalam menyelesaikan masalah dengan suatu metode tertentu. *Sorting* didefinisikan sebagai proses pengurutan sejumlah data yang disusun secara acak menjadi terurut dan teratur. Pengurutan ini terbagi menjadi dua yaitu *ascending* dan *descending*.

### B. Kompleksitas Algoritma

Kompleksitas suatu algoritma merupakan ukuran seberapa banyak komputasi yang dibutuhkan algoritma tersebut untuk mendapatkan hasil yang diinginkan. Hal-hal yang mempengaruhi kompleksitas waktu [3]:

1. Jumlah masukan data untuk suatu algoritma ( $n$ ).
2. Waktu yang dibutuhkan untuk menjalankan algoritma tersebut.
3. Ruang memori yang dibutuhkan untuk menjalankan algoritma.

Kompleksitas mempengaruhi performa atau kinerja dari suatu algoritma. Kompleksitas dibagi menjadi 3 jenis [4] yaitu:

- Kompleksitas kasus terburuk (*worst case*) dari algoritma adalah fungsi yang ditentukan oleh jumlah maksimum langkah-langkah yang diambil pada setiap *instance* dari ukuran  $n$ . Hal ini mewakili kurva melewati titik tertinggi dari setiap kolom.
- Kompleksitas kasus terbaik (*best case*) dari algoritma adalah fungsi yang didefinisikan oleh jumlah minimum langkah yang diambil pada setiap *instance* ukuran  $n$ . Hal ini mewakili kurva melewati titik terendah dari setiap kolom.
- Kompleksitas kasus rata-rata (*average case*) dari algoritma adalah fungsi yang didefinisikan oleh jumlah rata-rata langkah yang diambil pada setiap *instance* dari ukuran  $n$ .

Masing-masing jenis kompleksitas ini menunjukkan kecepatan atau waktu yang dibutuhkan algoritma untuk mengeksekusi sejumlah kode.

Ada 2 macam kompleksitas algoritma [5], yaitu kompleksitas waktu dan kompleksitas ruang. Kompleksitas waktu disimbolkan dengan  $T(n)$  dan kompleksitas ruang  $S(n)$ . Kompleksitas waktu,  $T(n)$ , diukur dari jumlah tahapan komputasi yang dibutuhkan untuk menjalankan algoritma sebagai fungsi dari ukuran masukan  $n$ . Kompleksitas ruang,  $S(n)$ , diukur dari memori yang digunakan oleh struktur data yang terdapat di dalam algoritma sebagai fungsi dari ukuran masukan  $n$ .

### C. Kompleksitas Waktu

Kompleksitas waktu [6] adalah konsep dalam ilmu komputer yang berkaitan dengan kuantifikasi jumlah waktu yang diambil oleh seperangkat kode atau algoritma untuk memproses atau menjalankan fungsi dari jumlah input.

Analisis kompleksitas waktu algoritma [7] adalah membandingkan waktu yang dibutuhkan algoritma dalam menyelesaikan perintah. Pada penelitian Estrada, A.H [8] menyebutkan bahwa dengan menganalisis kompleksitas waktu, dapat disimpulkan bahwa banyaknya jumlah data- $n$  berpengaruh terhadap kebutuhan waktu yang diperlukan.

Pada algoritma pengurutan terutama pada pengurutan dengan perbandingan, operasi dasarnya adalah operasi-operasi perbandingan elemen-elemen suatu larik dan operasi pertukaran elemen. Kedua hal itu dihitung terpisah karena jumlah keduanya tidaklah sama. Kompleksitas algoritma dinyatakan secara *asimptotik* dengan notasi *big-O*. Jika kompleksitas waktu untuk menjalankan suatu algoritma dinyatakan dengan  $T(n)$  dan memenuhi  $T(n) \leq C(f(n))$  untuk  $n \geq n_0$ , maka kompleksitas dapat dinyatakan dengan  $T(n) = O(f(n))$  [9].

### D. Brute Force

*Brute Force* adalah teknik paling sederhana untuk menyelesaikan permasalahan komputasi pada umumnya. Secara konseptual, *Brute Force* bekerja sebagai berikut [10]:

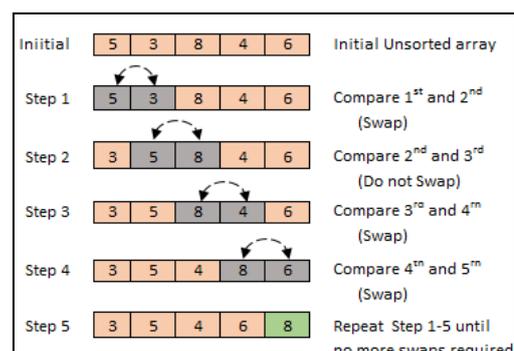
1. Mula-mula *pattern* dicocokkan pada awal teks
2. Dengan bergerak dari kiri ke kanan, bandingkan setiap karakter di dalam *pattern* dengan karakter yang bersesuaian di dalam teks sampai:
  - Semua karakter yang dibandingkan cocok atau sama (pencarian berhasil), atau
  - Dijumpai sebuah ketidakcocokan karakter (pencarian belum berhasil)
3. Bila *pattern* belum ditemukan kecocokannya dan teks belum habis, geser *pattern* satu karakter ke kanan dan ulangi langkah 2.

Algoritma yang menggunakan strategi *Brute Force* adalah algoritma *Bubble Sort* dan *Selection Sort*.

#### D.1 Bubble Sort

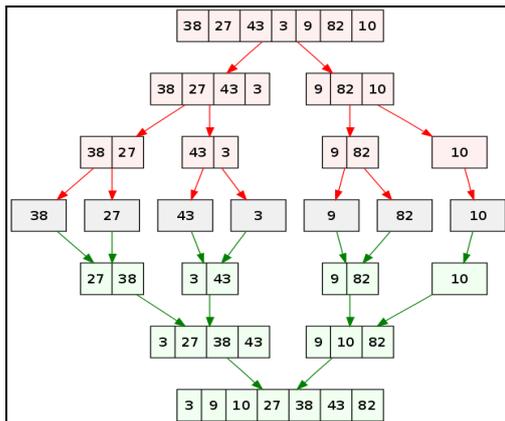
##### D.2.1 Konsep Bubble Sort

*Bubble sort* [11] adalah metode pengurutan yang membandingkan elemen yang sekarang dengan elemen berikutnya, jika elemen sekarang  $>$  elemen berikutnya maka posisinya ditukar, kalau tidak, tidak perlu ditukar, misalnya untuk  $n = 7$  maka akan dilakukan  $(n - 1) = 6$  tahap (mulai dari 0 sampai dengan  $n - 2$ ).



Gambar 1. Contoh simulasi *bubble sort*



Gambar 4. Contoh simulasi *merge sort*

Hasil waktu eksekusi dari algoritma *Bubble Sort* dapat dilihat pada Tabel 1.

Tabel 1. Waktu eksekusi algoritma *Bubble Sort*

Data	Running Time (ms)
50	0.24
100	0.425
500	1.98
1.000	2.825
2.500	9.46
5.000	42.4
10.000	232.175
25.000	1443.36
50.000	5330.995
100.000	19379.27

### E.2.2 Kompleksitas *Merge Sort*

Algoritma ini memiliki kompleksitas waktu rata-rata, *worst case*, dan *best case* yang sama yaitu  $O(n \log n)$  [14].

## III. HASIL DAN PEMBAHASAN

Pada penelitian ini, digunakan beberapa jumlah data yaitu 50, 100, 500, 1.000, 2.500, 5.000, 10.000, 25.000, 50.000, 100.000 dengan *range* 1.000.000 dan diukur menggunakan besaran *millisecond* dengan 3 angka dibelakang koma agar terlihat perbedaannya. Data yang digunakan berlaku untuk semua algoritma *sort* yang dibahas, dan bahasa pemrograman yang digunakan adalah *JavaScript*. Hasil *running time* tiap algoritma bergantung pada kondisi dan jenis laptop atau komputer yang digunakan.

### A. *Bubble Sort*

Gambar 5 merupakan potongan *source code* *JavaScript* dari algoritma *Bubble Sort*:

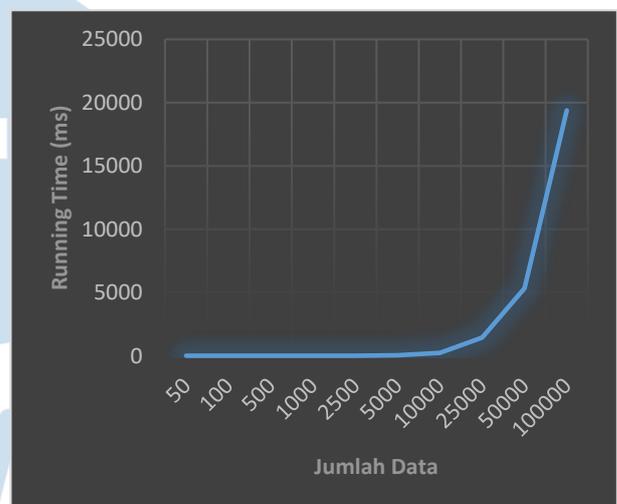
```

let bubbleSort = (data) => {
  let flag;
  let n = data.length - 1;
  let x = data;
  do {
    flag = false;
    for (var i = 0; i < n; i++) {
      if (x[i] > x[i + 1]) {
        var temp = x[i];
        x[i] = x[i + 1];
        x[i + 1] = temp;
        flag = true;
      }
    }
    n--;
  } while (flag);
  return x;
};

```

Gambar 5. Potongan *source code* *Bubble Sort*

Kompleksitas algoritma *Bubble Sort* dapat digambarkan seperti grafik berikut.

Gambar 6. Grafik kompleksitas algoritma *Bubble Sort*

### B. *Selection Sort*

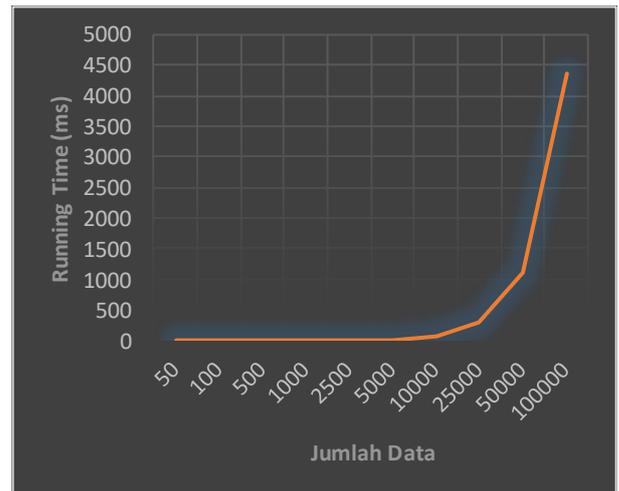
Gambar 7 merupakan potongan *source code* *JavaScript* dari algoritma *Selection Sort*:

```

let selectionSort = (data) => {
  let len = data.length;
  for (let i = 0; i < len; i++) {
    let min = i;
    for (let j = i + 1; j < len; j++) {
      if (data[min] > data[j]) {
        min = j;
      }
    }
    if (min !== i) {
      let tmp = data[i];
      data[i] = data[min];
      data[min] = tmp;
    }
  }
  return data;
};

```

Gambar 7. Potongan source code Selection Sort



Gambar 5. Grafik kompleksitas algoritma Selection Sort

Hasil waktu eksekusi dari algoritma Selection Sort dapat dilihat pada Tabel 2.

Tabel 2. Waktu eksekusi algoritma Selection Sort

Data	Running Time (ms)
50	0.105
100	0.23
500	1.36
1.000	1.19
2.500	2.36
5.000	10.025
10.000	49.48
25.000	284.925
50.000	1109.13
100.000	4354.09

Kompleksitas algoritma Selection Sort dapat digambarkan seperti grafik berikut.

### C. Quick Sort

Gambar 9 merupakan potongan source code JavaScript dari algoritma Quick Sort:

```

let quickSort = (items, left, right) => {
  let index;
  if (items.length > 1) {
    index = partition(items, left, right);
    if (left > index - 1) {
      quickSort(items, left, index - 1);
    }
    if (index > right) {
      quickSort(items, index, right);
    }
  }
  return items;
};

```

Gambar 9. Potongan source code Quick Sort

Hasil waktu eksekusi dari algoritma Quick Sort dapat dilihat pada Tabel 3.

Tabel 1. Waktu eksekusi algoritma Quick Sort

Data	Running Time (ms)
50	0.065
100	0.005
500	0.01
1.000	0.005
2.500	0.005
5.000	0.01
10.000	0.005
25.000	0.005
50.000	0.01
100.000	0.015

Kompleksitas algoritma *Quick Sort* dapat digambarkan seperti grafik berikut.



Gambar 10. Grafik kompleksitas algoritma *Quick Sort*

#### D. Merge Sort

Gambar 11 merupakan potongan *source code* JavaScript dari algoritma *Merge Sort*:

```
let mergeSort = (unsortedArray) => {
  if (unsortedArray.length <= 1) {
    return unsortedArray;
  }
  const middle = Math.floor(unsortedArray.length / 2);
  const left = unsortedArray.slice(0, middle);
  const right = unsortedArray.slice(middle);
  return merge(mergeSort(left), mergeSort(right));
};
```

Gambar 11. Potongan *source code* *Quick Sort*

Hasil waktu eksekusi dari algoritma *Merge Sort* dapat dilihat pada Tabel 4.

Tabel 4. Waktu eksekusi algoritma *Merge Sort*

Data	Running Time (ms)
50	0.15
100	0.18
500	0.81
1.000	2.665
2.500	6.265
5.000	6.3
10.000	9.18
25.000	18.815
50.000	39.265
100.000	78.525

Kompleksitas algoritma *Merge Sort* dapat digambarkan seperti grafik berikut.



Gambar 12. Grafik kompleksitas algoritma *Merge Sort*

#### E. Perbandingan Algoritma *Bubble Sort*, *Selection Sort*, *Quick Sort* dan *Merge Sort*

Grafik dibawah ini merupakan hasil perbandingan efisiensi strategi *Brute Force* dan *Divide and Conquer*.



Gambar 13. Grafik perbandingan kompleksitas algoritma

#### IV. SIMPULAN

Berdasarkan dari hasil algoritma pengurutan yang dilakukan menggunakan strategi *Brute Force* dan *Divide and Conquer*, dapat disimpulkan bahwa strategi *Divide and Conquer* lebih efektif dan efisien dalam menangani masalah pengurutan. Algoritma yang mudah dalam hal implementasi adalah *Bubble Sort* dan *Selection Sort*, keduanya memiliki kompleksitas  $O(n^2)$ . Algoritma yang lebih efisien adalah algoritma *Quick Sort* dan *Merge Sort* dengan

kompleksitasnya adalah  $O(n \log n)$ . Dari keempat algoritma ini, yang memiliki efisiensi paling bagus adalah algoritma *Quick Sort* dengan *running time* rata-rata sebesar 0,0135 ms dari data yang diuji. Sedangkan *bubble sort* merupakan algoritma yang memiliki efisiensi paling buruk dari keempat algoritma yang diuji dengan *running time* rata-rata sebesar 2644,313 ms dari data yang diuji.

#### DAFTAR PUSTAKA

- [1] "Sorting Algorithms," [Online]. Available: <https://guide.freecodecamp.org/algorithms/sorting-algorithms/>. [Accessed 20 Mei 2020].
- [2] Amran. "Perbandingan Strategi Brute Force dan Divide and Conquer Pada Algoritma Pengurutan". [https://www.academia.edu/39080158/Perbandingan\\_Strategi\\_Brute\\_Force\\_and\\_Divide\\_and\\_Conquer\\_Pada\\_Algoritma\\_Pengurutan](https://www.academia.edu/39080158/Perbandingan_Strategi_Brute_Force_and_Divide_and_Conquer_Pada_Algoritma_Pengurutan). [Accessed 20 Mei 2020].
- [3] Wikipedia, [http://en.wikipedia.org/wiki/Big\\_O\\_notation/](http://en.wikipedia.org/wiki/Big_O_notation/). [Accessed 20 Mei 2020].
- [4] "Best, Worst, and Average-Case Complexity," [Online]. Available: <https://www8.cs.umu.se/kurser/TDBA77/VT06/algorithms/BOOK/BOOK/NODE13.HTM>. [Accessed 22 Mei 2020].
- [5] "Algorithm Efficiency," [Online]. Available: [http://www.cs.kent.edu/~durand/CS2/Notes/03\\_Algs/ds\\_alg\\_efficiency.html](http://www.cs.kent.edu/~durand/CS2/Notes/03_Algs/ds_alg_efficiency.html). [Accessed 22 Mei 2020].
- [6] "Time Complexity," [Online]. Available: <https://www.techopedia.com/definition/22573/time-complexity>. [Accessed 22 Mei 2020].
- [7] D. W. Nugraha, "Penerapan Kompleksitas Waktu Algoritma Prim Untuk Menghitung Kemampuan Komputer Dalam Melaksanakan Perintah," vol. 2, 2012.
- [8] A. H. Estrada S, "Telaah Waktu Eksekusi Program Terhadap Kompleksitas Waktu Algoritma Brute Force Dan Divide And Conquer Dalam Penyelesaian Operasi List," vol. 3, 2003.
- [9] S. N. B. Tjaru, "Kompleksitas Algoritma Pengurutan Selection Sort dan Insertion Sort". MAKALAH IF2091 STRATEGI ALGORITMIK, 2009.
- [10] P. Triono, "Analisis Perbandingan Algoritma Sorting dan Searching". <https://www.academia.edu/6526802/Analisis-perbandingan-algoritma>, 2010. [Accessed 22 Mei 2020].
- [11] S. Y. Yahya, "Analisa Perbandingan Algoritma Bubble Sort dan Selection Sort Dengan Metode Perbandingan Eksponensial," *Pelita Informatika Budi Darma*, vol. VI, 2014.
- [12] K. Ali, "A Comparative Study of Well Known Sorting Algorithms," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 1, pp. 277-280, 2017.
- [13] M. Usman, M. Afzal and Z. Bajwa, "Performance Analysis of Sorting Algorithms with C#," *International Journal for Research in Applied Science & Engineering*, vol. 3, no. 1, pp. 201-204, 2015.
- [14] D. Rajagopal and K. Thilakavalli, "Different Sorting Algorithm's Comparison based Upon the," *International Journal of u- and e- Service, Science and Technology*, vol. 9, no. 8, pp. 287-296, 2016.
- [15] N. D. S. L. S. P. Parag Bhalchandra, "A Comprehensive Note on Complexity Issues in Sorting Algorithms," *Advances in Computational Research*, vol. 1, no. 2, pp. 1-9, 2009.
- [16] M. E. Wira Putra, "Perbandingan Algoritma Pengurutan Merge Sort, Quick Sort dan Heap Sort Dilihat dari Kompleksitasnya". <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2009-2010/Makalah0910/MakalahStrukdis0910-023.pdf> [Accessed 22 Mei 2020].



UMN

# Kombinasi *Backpropagation* dan Hopfield Modifikasi untuk Persamaan *Polynomial*

Rina Pramitasari<sup>1</sup>, Imam Rofiki<sup>2</sup>

<sup>1</sup>Teknik Komputer, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia  
rina.pramitasari@amikom.ac.id

<sup>2</sup>Tadris Matematika, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, Indonesia  
imam.rofiki@uin-malang.ac.id

Diterima 23 Mei 2020

Disetujui 23 Juni 2020

**Abstract**—Salah satu sistem kriptografi yang bertahan di era komputer quantum adalah kriptografi kunci publik multivariat. Tingkat keamanan sistem kriptografi kunci publik multivariat adalah sulitnya menyelesaikan sistem persamaan polinomial multivariable. Tujuan artikel adalah mengkriptanalisis dengan mencari akar menyelesaikan persamaan polinomial. Penelitian ini menggunakan metode kombinasi *Backpropagation* dan Hopfield Modifikasi. Hasil penelitian menunjukkan bahwa metode kombinasi *Backpropagation* dan Hopfield Modifikasi lebih baik dari pada metode Hopfield Modifikasi saja. Karena menjamin nilai awal yang diberikan pada Hopfield Modifikasi selalu dekat dengan nilai solusi. Metode kombinasi *Backpropagation* dan Hopfield Modifikasi memberikan solusi yang akurat.

**Index Terms**—*Backpropagation*, Hopfield Network Modifikasi, Persamaan Polinomial

## I. PENDAHULUAN

Sistem kriptografi kunci publik yang paling banyak digunakan adalah RSA. Algoritme RSA berasal dari nama akhir penemunya pada tahun 1977, yaitu Rivest-Shamir-Adleman. Tingkat keamanan RSA terletak pada faktorisasi bilangan bulat. Kunci publik yang aman pada RSA setidaknya 1024 bits atau 2048 bits. Hal ini mengakibatkan proses enkripsi tidak efisien. Ancaman baru juga muncul yang bisa melemahkan RSA yaitu komputer quantum yang sedang berkembang. Ada beberapa kandidat sistem kriptografi yang mampu bertahan dengan komputer quantum yaitu *Lattice-Based Cryptography*, *Code-Based Cryptography*, *Multivariate-Based Cryptography* [4], [7], [8], [11].

Pengujian sistem kriptografi salah satunya adalah dengan menggunakan kriptanalisis. Sistem kriptografi kunci publik atau sistem kriptografi asimetris, menggunakan dua jenis kunci, yaitu kunci publik dan kunci rahasia. Kunci publik digunakan untuk mengenkripsi suatu pesan. Kunci publik bersifat umum, sehingga dapat dilihat oleh semua orang termasuk pihak penyerang. Kunci rahasia digunakan untuk mendekripsi suatu pesan. Kunci rahasia bersifat

dirahasiakan dan hanya orang tertentu yang boleh mengetahuinya [7].

Sistem kriptografi kunci publik multivariat didasarkan pada sistem persamaan *multivariable* nonlinear pada bidang hingga. Tingkat keamanan didasarkan pada sulitnya penyelesaian sebuah sistem persamaan polinomial multivariable kuadrat yang disebut MQ-problem (*multivariate quadratic problem*) [4], [7], [8], [11].

Menyelesaikan sistem persamaan nonlinear bentuk yang sederhana dapat diselesaikan dengan metode analitik. Sedangkan faktanya sering menemukan bentuk yang rumit, tidak semua persamaan dapat dipecahkan dengan metode analitik, maka solusinya dapat dicari dengan metode numerik. Sedangkan metode numerik harus memenuhi syarat dekat dengan solusi untuk bisa konvergen. Maka untuk menyelesaikan sistem persamaan tersebut digunakan jaringan saraf tiruan di mana sebuah prosesor yang terdistribusi parallel dan mempunyai kecenderungan untuk menyimpan pengetahuan yang didapatkannya dari pengalaman dan membuatnya tetap tersedia untuk digunakan [9].

Jaringan *Backpropagation* diambil dari cara kerja yaitu bahwa gradien *error* unit-unit tersembunyi diturunkan dari penyiaran kembali *error* yang diasosiasikan dengan unit *output*. Hal ini karena nilai target untuk unit-unit tersembunyi tidak diberikan [2].

Jaringan Hopfield pertama kali diperkenalkan John Hopfield pada tahun 1982. Jaringan Hopfield adalah pelatihantak terbimbing (*unsupervised learning*). Jaringan Hopfield merupakan jaringan syaraf tiruan yang terhubung penuh (*fully connected*), yaitu bahwa setiap neuron terhubung dengan neuron lainnya [2]. Jaringan Hopfield Modifikasi mampu untuk menyelesaikan sistem persamaan nonlinear di dalam satu variabel. Hasil penelitian menunjukkan bahwa jaringan Hopfield modifikasi akan selalu konvergen terhadap sembarang nilai awal, dimana sebaliknya metode numerik yang harus memenuhi syarat “dekat” dengan solusi [1], [6].

## II. LANDASAN TEORI

### A. Persamaan Polynomial

Persoalan mencari solusi persamaan nonlinear adalah: Tentukan nilai  $x$  yang memenuhi persamaan.

$$f(x) = 0$$

yaitu nilai  $x = s$  sedemikian sehingga  $f(s)$  sama dengan nol [9]. Persamaan nonlinear yang melibatkan fungsi transeden, diantaranya sinus, cosinus, eksponensial, logaritma. Selain itu, persamaan nonlinear yang tidak melibatkan fungsi transenden, yaitu persamaan polynomial. Bentuk umum persamaan polynomial satu variabel  $x$  adalah

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$$

Contohnya:

- a. Satu variabel yaitu  $x$ ,

$$4x^5 + 9x^3 + 2x - 5 = 0$$

- b. Dua variabel yaitu  $x$  dan  $y$ ,

$$3x^2 + 9xy - y = 0$$

Pada penelitian ini, penulis menggunakan satu variabel.

### B. Backpropagation

*Backpropagation* adalah metode yang menurunkan gradien untuk meminimalkan penjumlahan *error* kuadrat *output* jaringan. Nama lain dari *Backpropagation* adalah aturan delta yang digeneralisasi.

Dalam *Backpropagation*, setiap unit yang berbeda di lapisan input terhubung dengan setiap unit yang ada di lapisan tersembunyi. Hal serupa berlaku pula pada lapisan tersembunyi. Setiap unit yang ada di lapisan tersembunyi terhubung dengan setiap unit yang ada di lapisan output [3], [5], [6].

Langkah *Backpropagation* adalah sebagai berikut:

- Inisialisasi bobot dan bias
  1. **While** kondisi berhenti tidak terpenuhi **do** langkah ke-2 sampai langkah ke-9.
  2. Untuk setiap pasang pola training, lakukan langkah ke-3 sampai langkah ke-8.
- Umpan Maju (Feedforward)
  3. Setiap unit *input*, mengirimkan sinyal *input* ke semua unit pada lapisan tersembunyi.
  4. Pada setiap unit di lapisan tersembunyi, sinyal *input* *output* lapisan tersembunyi dihitung dengan dengan fungsi aktivasi terhadap penjumlahan sinyal *input* berbobot.
  5. Pada setiap unit di lapisan *output* unit, dihitung sinyal *output*-nya dengan

menerapkan fungsi aktivasi terhadap penjumlahan sinyal input berbobot.

- Umpan Mundur (*Backpropagation Error*)
  6. Pada setiap unit *output* menerima pola target lalu informasi kesalahan lapisan *output* dihitung. Dikirim ke lapisan di bawahnya dan digunakan untuk menghitung besar koreksi bobot dan bias antara lapisan tersembunyi dengan lapisan *output*.
  7. Pada setiap unit di lapisan tersembunyi dilakukan perhitungan informasi kesalahan lapisan tersembunyi. Kemudian digunakan untuk menghitung besar koreksi bobot dan bias antara lapisan input dan lapisan tersembunyi.
- Update Bobot dan Bias (*Adjustment*)
  8. Pada setiap unit output, dilakukan *update*-an bias dan bobot.
  9. Tes kondisi berhenti. Digunakan kriteria MSE (*Mean Square Error*).

### C. Hopfield Modifikasi

Hopfield mengembangkan rancangan jaringan biner sehingga neuron dapat memperhitungkan nilai real [3]. Pengembangan dari jaringan Hopfield Kontinu adalah menyerupai kerja jaringan diskrit, tetapi jaringan ini mempunyai kemampuan lebih karena arsitekturnya lebih kompleks dan tidak hanya nilai biner 0 dan 1 [1].

Hopfield Modifikasi didasarkan pada prinsip Hopfield kontinu, karena nilai *input* dan *output* yang diinginkan adalah real antara 0 sampai 1, tidak hanya 0 atau 1. Jumlah neuron pada jaringan sama dengan jumlah variabel persamaan. Hubungan antara neuron pada Hopfield Modifikasi untuk menyelesaikan persamaan *polynomial* bergantung pada hubungan antara variabel persamaan dengan koefisien, yang diturunkan sebagai bobot pada jaringan. Hubungan antara variabel pada persamaan *polynomial* adalah hubungan nonlinear, sehingga jaringan Hopfield harus dimodifikasi [1].

Langkah-langkah Hopfield Modifikasi adalah sebagai berikut:

1. Mengubah persamaan *polynomial* menjadi bentuk fungsi energi.

$$g_i(.) = (f_i(x_1, \dots, x_j, \dots, x_n) + (-P_i))$$

$$E = -\frac{1}{2} \sum_{i=1}^m (g_i(.))^2$$

- Menurunkan persamaan fungsi energi.

$$\frac{du_j}{dt} = \frac{\partial E}{\partial x_j}$$

- Kemudian menyesuaikan dengan persamaan Hopfield.

$$\sum_i W_{ji} f_{ji}(x_1, x_2, \dots, x_n) + I_j, (i, j) \in n$$

- Menginisialisasi nilai awal  $x(1)$  dan  $u(1)$ .
- Menghitung dengan metode Euler untuk memperbaiki nilai  $u(t)$ .

$$u(t+1) = u_j(t) + \Delta t \left( \frac{\partial E}{\partial x_j} \right)$$

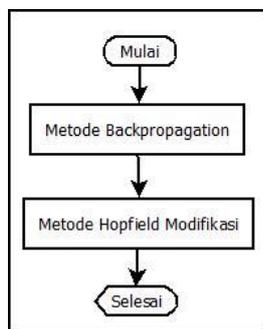
- Memperbarui nilai  $x(t)$ .

$$x(t) = \frac{1}{1 + e^{-u(t)}}$$

- Hingga syarat terpenuhi.

### III. METODE PENELITIAN

Gambar 1 menunjukkan *flowchart* penelitian. Langkah pertama, dengan menggunakan metode *Backpropagation* didapat nilai *output* yang optimal. Langkah kedua, menggunakan nilai *output* untuk memberikan nilai awal pada metode Hopfield Modifikasi untuk menyelesaikan persamaan *polynomial* [10]. Hasil penelitian akan dibandingkan dengan metode Hopfield Modifikasi dan metode konvensional yaitu metode numerik Newton Raphson.



Gambar 1. *Flowchart* penelitian

Misalkan persamaan *polynomial* berbentuk:

$$Ax^2 + Bx + C = 0$$

Proses *Backpropagation* menggunakan *layer input* 3 neuron, *layer hidden* 10 neuron, dan *layer output* 1 neuron. Fungsi aktivasi yang digunakan adalah Sigmoid Biner dan *learning rate* ( $\alpha$ ) = 0,5. *Epoch* = 10000. Toleransi *error* yang digunakan adalah 0,0000000000000001.

Proses Hopfield Modifikasi adalah:

- Mengubah persamaan *polynomial* menjadi bentuk fungsi energi.

$$Ax^2 + Bx + C = 0$$

$$E = -\frac{1}{2} \sum_{i=1}^m (g_i(\cdot))^2$$

$$E = -\frac{1}{2} (Ax^2 + Bx + C)^2$$

- Menurunkan persamaan fungsi energi.

$$\frac{du_j}{dt} = \frac{\partial E}{\partial x_j}$$

$$= -(2A^2x^3 + (AB + 2AB)x^2 + (B^2 + 2AC)x + BC)$$

- Kemudian menyesuaikan dengan persamaan Hopfield.

$$\frac{du}{dt} = W_1x^3 + W_2x^2 + W_3x + I_{bias}$$

$$W_1 = 2A^2, W_2 = AB + 2AB,$$

$$W_3 = B^2 + 2AC, I_{bias} = BC$$

- Menginisialisasi nilai awal  $x(1)$  adalah hasil dari *Backpropagation* dan  $u(1)$  adalah -0.5, 1, atau 0.01.

- Menghitung dengan metode Euler untuk memperbaiki nilai  $u(t)$ .

$$u(t+1) = u(t) + \Delta t (W_1x^3 + W_2x^2 + W_3x + I_{bias})$$

- Memperbarui nilai  $x(t)$

$$x(t) = \frac{1}{1 + e^{-u(t)}}$$

- Hingga syarat terpenuhi.

### IV. PEMBAHASAN

Dalam pembahasan menggunakan pemrograman *Python*. Kemudian menyiapkan data pelatihan dan data pengujian dengan merujuk penelitian [1], [4], [7], [12].

Data pelatihan terdiri dari 2 bentuk persamaan yaitu Pertama berbentuk:

$$Ax^2 + Bx + C = 0$$

Jumlah dari data pelatihan terdiri dari 10 dan 20 data. Data pengujian tersebut terdiri dari 3 persamaan yaitu:

$$x^2 - 0.0324 = 0 \text{ dengan Output } 0.18$$

$$x^2 - 0.3364 = 0 \text{ dengan Output } 0.58$$

$$x^2 - 0.9604 = 0 \text{ dengan Output } 0.98$$

Jaringan *Backpropagation* menggunakan *layer input* 3 neuron, *layer hidden* 10 neuron, dan *layer output* 1 neuron. Fungsi aktivasi yang digunakan adalah Sigmoid Biner dan *learning rate* ( $\alpha$ ) = 0,5. *Epoch* = 10000. Toleransi *error* yang digunakan adalah 0,0000000000000001.

Kedua berbentuk:

$$Ax^4 + Bx^3 + Cx^2 + Dx + E = 0$$

Jumlah dari data pelatihan terdiri dari 10 dan 20 data. Data pengujian tersebut terdiri dari 3 persamaan yaitu:

$$x^4 + x^3 - 0.0324x - 0.00104976 = 0$$

dengan *Output* 0.18

$$x^4 + x^3 - 0.3364x - 0.11316496 = 0$$

dengan *Output* 0.58

$$x^4 + x^3 - 0.9604x - 0.92236816 = 0$$

dengan *Output* 0.98

Jaringan *Backpropagation* menggunakan *layer input* 5 neuron, *layer hidden* 10 neuron, dan *layer output* 1 neuron, Fungsi aktivasi yang digunakan adalah Sigmoid Biner dan *learning rate* ( $\alpha$ ) = 0,5. *Epoch* = 10000. Toleransi *error* yang digunakan adalah 0,0000000000000001.

Kemudian melakukan proses perhitungan jaringan Hopfield Modifikasi berdasarkan perbedaan nilai  $u(t)$ . Nilai  $u(t)$  yang dipilih adalah -0.5, 1, dan 0.01.

Tabel 1. Hasil pengujian dengan persamaan

$$x^2 - 0.0324 = 0$$

Dengan 10 Training Data			
UI	X akhir	Iterasi	Error
-0.5	0.1800000045	840	0.0
1	0.1800000045	845	0.0
0.01	0.1800000046	842	0.0
Dengan 20 Training Data			
UI	X akhir	Iterasi	Error
-0.5	0.1800000045	840	0.0
1	0.1800000045	845	0.0
0.01	0.1800000046	842	0.0

Tabel 2. Hasil pengujian dengan persamaan

$$x^2 - 0.3364 = 0$$

Dengan 10 Training Data			
UI	X akhir	Iterasi	Error
-0.5	0.5799999999	58	0.0
1	0.5800000001	54	0.0
0.01	0.5799999999	54	0.0

Dengan 20 Training Data			
UI	X akhir	Iterasi	Error
-0.5	0.5799999999	58	0.0
1	0.5800000001	54	0.0
0.01	0.5799999999	54	0.0

Tabel 3. Hasil pengujian dengan persamaan  $x^2 - 0.9604 = 0$

Dengan 10 Training Data			
UI	X akhir	Iterasi	Error
-0.5	0.9799999999	218	0.0
1	0.9799999991	217	0.0
0.01	0.9799999989	216	0.0
Dengan 20 Training Data			
UI	X akhir	Iterasi	Error
-0.5	0.9799999999	218	0.0
1	0.9799999999	216	0.0
0.01	0.9799999988	215	0.0

Tabel 4. Hasil pengujian dengan persamaan  $x^4 + x^3 - 0.0324x - 0.00104976 = 0$

Dengan 10 Training Data			
UI	X akhir	Iterasi	Error
-0.5	0.1839713339	8865	0.0
1	0.1839713265	8875	0.0
0.01	0.1839713339	8870	0.0
Dengan 20 Training Data			
UI	X akhir	Iterasi	Error
-0.5	0.1839713339	8865	0.0
1	0.1839713265	8875	0.0
0.01	0.1839713339	8870	0.0

Tabel 5. Hasil pengujian dengan persamaan  $x^4 + x^3 - 0.3364x - 0.11316496 = 0$

Dengan 10 Training Data			
UI	X akhir	Iterasi	Error
-0.5	0.5799851742	41	0.0
1	0.5799851742	33	0.0
0.01	0.5799851742	34	0.0
Dengan 20 Training Data			
UI	X akhir	Iterasi	Error
-0.5	0.5799851742	41	0.0
1	0.5799851742	33	0.0
0.01	0.5799851741	32	0.0

Tabel 6. Hasil pengujian dengan persamaan  
 $x^4 + x^3 - 0.9604x - 0.92236816 = 0$

Dengan 10 Training Data				
U1	X akhir	Iterasi	Error	
-0.5	0.980001311	21	0.0	
1	0.980001311	19	0.0	
0.01	0.980001311	19	0.0	
Dengan 20 Training Data				
U1	X akhir	Iterasi	Error	
-0.5	0.9800013111	20	0.0	
1	0.9800013111	20	0.0	
0.01	0.980001311	21	0.0	

Pada Tabel 1 sampai 6 menunjukkan hasil akar persamaan polinomial dari jumlah data pelatihan 10 dan 20 dan jumlah data pengujian 0.18, 0.58, 0.98.

Tabel 7. Hasil pengujian Hopfield Modifikasi dengan persamaan  $x^2 - 0.0324 = 0$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	Error
-2.05	-0.5	0.1800000046	852	0.0
0.11	-0.5	0.1800000045	840	0.0
1.65	-0.5	-	-	-
-2.05	1	0.1800000045	854	0.0
0.11	1	0.1800000045	845	0.0
1.65	1	-	-	-
-2.05	0.01	0.1800000042	857	0.0
0.11	0.01	0.1800000046	842	0.0
1.65	0.01	-	-	-

Tabel 8. Hasil pengujian Hopfield Modifikasi dengan persamaan  $x^2 - 0.3364 = 0$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	Error
-2.05	-0.5	0.5800000002	64	0.0
0.11	-0.5	0.5799999999	57	0.0
1.65	-0.5	0.5799999999	6514	0.0
-2.05	1	0.5800000002	65	0.0
0.11	1	0.5800000001	54	0.0
1.65	1	0.5799999999	1507	0.0
-2.05	0.01	0.5800000001	66	0.0
0.11	0.01	0.5799999999	53	0.0
1.65	0.01	0.5799999998	3938	0.0

Tabel 9. Hasil pengujian Hopfield Modifikasi dengan persamaan  $x^2 - 0.9604 = 0$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	Error
-2.05	-0.5	0.9800000012	326	0.0
0.11	-0.5	0.9799999989	217	0.0
1.65	-0.5	0.9799999989	510	0.0
-2.05	1	0.9800000013	344	0.0
0.11	1	0.9799999999	216	0.0
1.65	1	0.9799999989	285	0.0
-2.05	0.01	0.9800000001	335	0.0
0.11	0.01	0.9799999988	215	0.0
1.65	0.01	0.9799999991	2397	0.0

Tabel 10. Hasil pengujian Hopfield Modifikasi dengan persamaan

$$x^4 + x^3 - 0.0324x - 0.00104976 = 0$$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	Error
-2.05	-0.5	-	-	-
0.11	-0.5	0.1839713339	8865	0.0
1.65	-0.5	-	-	-
-2.05	1	-	-	-
0.11	1	0.1839713265	8875	0.0
1.65	1	-	-	-
-2.05	0.01	-	-	-
0.11	0.01	0.1839713339	8870	0.0
1.65	0.01	-	-	-

Tabel 11. Hasil pengujian Hopfield Modifikasi dengan persamaan

$$x^4 + x^3 - 0.3364x - 0.11316496 = 0$$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	Error
-2.05	-0.5	-	-	-
0.11	-0.5	0.5799851742	41	0.0
1.65	-0.5	-	-	-
-2.05	1	-	-	-
0.11	1	0.5799851741	31	0.0
1.65	1	-	-	-
-2.05	0.01	-	-	-
0.11	0.01	0.5799851742	34	0.0
1.65	0.01	-	-	-

Tabel 12. Hasil pengujian Hopfield Modifikasi dengan persamaan

$$x^4 + x^3 - 0.9604x - 0.92236816 = 0$$

Metode Hopfield Modifikasi				
X1	U1	X akhir	Iterasi	Error
-2.05	-0.5	-	-	-
0.11	-0.5	-	-	-
1.65	-0.5	-	-	-
-2.05	1	-	-	-
0.11	1	0.9800013111	22	0.0
1.65	1	-	-	-
-2.05	0.01	-	-	-
0.11	0.01	-	-	-
1.65	0.01	-	-	-

Pada Tabel 7 sampai 12 menunjukkan hasil pengujian dengan metode Hopfield Modifikasi. Hasil akar persamaan polinomial dari data pengujian 0.18, 0.58, 0.98.

Tabel 13. Hasil pengujian Newton Raphson dengan persamaan  $x^2 - 0.0324 = 0$ 

Metode Newton Raphson			
X1	X akhir	Iterasi	Error
-2.05	-	-	-
0.11	0.18	5	0.0
1.65	0.18	8	0.0

Tabel 14. Hasil pengujian Newton Raphson dengan persamaan  $x^2 - 0.3364 = 0$ 

Metode Newton Raphson			
X1	X akhir	Iterasi	Error
-2.05	-	-	-
0.11	0.58	7	0.0
1.65	0.58	6	0.0

Tabel 15. Hasil pengujian Newton Raphson dengan persamaan  $x^2 - 0.9604 = 0$ 

Metode Newton Raphson			
X1	X akhir	Iterasi	Error
-2.05	-	-	-
0.11	0.98	8	5.66440318e-17
1.65	0.98	5	5.66440318e-17

Tabel 16. Hasil pengujian Newton Raphson dengan persamaan  $x^4 + x^3 - 0.0324x - 0.00104976 = 0$ 

Metode Newton Raphson			
X1	X akhir	Iterasi	Error
-2.05	-	-	-
0.11	0.18	8	-6.62617204e-15
1.65	0.18	11	-1.99055818e-15

Tabel 17. Newton Raphson Hasil pengujian dengan persamaan  $x^4 + x^3 - 0.3364x - 0.11316496 = 0$ 

Metode Newton Raphson			
X1	X akhir	Iterasi	Error
-2.05	-	-	-
0.11	-	-	-
1.65	0.58	8	2.86484918e-17

Tabel 18. Newton Raphson Hasil pengujian dengan persamaan  $x^4 + x^3 - 0.9604x - 0.92236816 = 0$ 

Metode Newton Raphson			
X1	X akhir	Iterasi	Error
-2.05	-	-	-
0.11	-	-	-
1.65	0.98	6	-1.34736562e-15

Pada Tabel 13 sampai 18 menunjukkan hasil pengujian dengan metode Newton Raphson. Hasil akar persamaan polinomial dari data pengujian 0.18, 0.58, 0.98.

## V. SIMPULAN

Berdasarkan hasil penelitian yang dilakukan metode kombinasi *Backpropagation* dan Hopfield Modifikasi telah berhasil dilakukan. Tingkat akurasi tersebut dipengaruhi oleh beberapa faktor yaitu pemilihan data pelatihan, nilai awal yang diberikan pada Hopfield Modifikasi terjamin dekat dengan nilai akar data pengujian. Konvergensi metode *Backpropagation* dapat terjebak di dalam minimum lokal. Sehingga dikombinasikan dengan metode Hopfield Modifikasi dapat keluar dari konvergensi yang terjebak di dalam minimum lokal.

Kemudian dibandingkan dengan metode Hopfield Modifikasi saja. Hasil pengujian didapat jika diberikan nilai awal jauh dari nilai akar dan kompleksnya persamaan polinomial, tidak mendapatkan hasil yang akurat. Kemudian dibandingkan dengan metode Newton Raphson. Hasil pengujian didapat nilai akar akan konvergensi jika dekat dengan nilai solusinya.

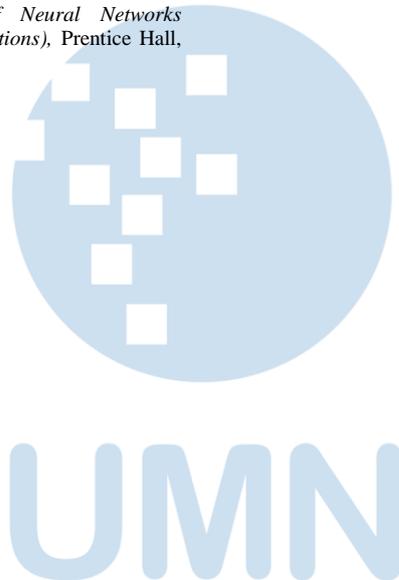
Karena nilai awal yang diberikan terlalu jauh, menyebabkan terjadinya iterasi divergen.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Amikom Yogyakarta yang telah memberikan dana penelitian.

#### DAFTAR PUSTAKA

- [1] D. Mishra, P. K. Kalra, *Modified Hopfield Neural Network Approach for Solving Nonlinear Algebraic Equations*, *Engineering Letters*, 14:1, EL\_14\_1\_23 (Advance online publication: 12 February 2007).
- [2] D. Puspitaningrum, *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta: Andi, 2006.
- [3] D. Ratnawati, *Penerapan JST dengan Metode Backpropagation Studi Kasus : Prakiraan Cuaca dan Kualitas Udara Wilayah DKI Jakarta*, *Tesis*, Program Pascasarjana, Universitas Gadjah Mada, Yogyakarta. 2008.
- [4] J. Ding, J. E. Gower, dan D. S. Schmidt, 2006, *Multivariate Public Key Cryptosystem*, Springer, USA. [13]
- [5] L. Fausett, 1994, *Fundamentals of Neural Networks (Architectures, Algorithms, and Applications)*, Prentice Hall, Englewood Cliffs, New Jersey.
- [6] M. M. Gupta, L. Jin, N. Homma, 2003, *Static and Dynamic Neural Networks, From Fundamentals to Advanced Theory*, John Wiley & Sons, Inc., Hoboken, New Jersey.
- [7] M. Z. Riyanto, "Sistem Kriptografi Kunci Publik Multivariat", Seminar Nasional Matematika dan Pendidikan Matematika, 27 November 2010 di Jurusan Pendidikan Matematika FMIPA UNY.
- [8] O. Billet, and J. Ding, "Overview of Cryptanalysis Techniques in Multivariate Public Key Cryptography", DOI 10.1007/978-3-540-93806-4\_15, Springer-Verlag Berlin Heidelberg, 2009.
- [9] R. Munir, *Metode Numerik*. Bandung: Informatika, 2008.
- [10] U. M. Kaczmar, and T. Switek, "Combined Unsupervised-Supervised Classification Method", J.D. Vel'asquez et al. (Eds.): KES 2009, Part II, LNAI 5712, pp. 861–868, 2009.
- [11] X. Wang, B. Yang, "An Improved Signature Model of Multivariate Polynomial Public Key cryptosystem Against Key Recovery Attack", DOI: 10.3934/mbe.2019388, <http://www.aimspress.com/journal/MBE>, 2019.
- [12] Y. Shin, J. Ghosh, 1991, *Approximation of Multivariate Functions Using Ridge Polynomial Networks*, Department of Electrical and Computer Engineering, The University of Texas at Austin, 1991.



# Implementasi Algoritma Apriori untuk Rekomendasi Kombinasi Produk Penjualan

Andre Setiawan<sup>1</sup>, Farica Perdana Putri<sup>2</sup>

<sup>1,2</sup>Program Studi Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia  
andre.setiawan@student.umn.ac.id  
farica@umn.ac.id

Diterima 8 Juni 2020

Disetujui 17 Juni 2020

**Abstract**—Analyzing and systematically extracting essential information from recording transactions is important for a business, including online stores. Sometimes, some online stores offer a product package that is not suitable for the customer. It happens because they did not process the data transaction to observe the association between products on a package. A web-based recommendation system was built using the CodeIgniter framework with PHP programming language. The system developed using Market Basket analysis that can determine the combination of products. Apriori algorithm used as a technique to analyze the relationship between products based on the data transaction. The lift ratio value generated from the rule is 1.18, which means that the rule has the power of relationships between items. We evaluate the system using USE questionnaire with usefulness results is 90.83%, ease of use 89.09%, ease of learning 95%, and satisfaction 90.94%, which strongly agree in every aspect.

**Index Terms**—Apriori, CodeIgniter, Lift Ratio, Market Basket Analysis, Recommendation

## I. PENDAHULUAN

PT Grup Gaga Indonesia atau biasa disebut sebagai *Party Gaga* adalah sebuah toko daring di Indonesia yang menawarkan berbagai pilihan kebutuhan pesta yang membuat jauh lebih mudah, terjangkau dan menyenangkan untuk mendekorasi, serta meningkatkan pengalaman acara agar lebih mengesankan [1].

Dalam proses penjualan jasa ataupun barang, seringkali toko daring menggunakan kombinasi produk yang kurang tepat sasaran dan diminati oleh para pembeli [2]. Hal ini tentunya dapat menurunkan tingkat penjualan karena barang-barang tersebut yang jarang dibeli pembeli secara bersamaan. Lalu, pemahaman dan definisi yang tepat akan suatu masalah merupakan hal yang penting dalam menciptakan suatu solusi *software* yang tepat [3]. Maka dari itu, melakukan kombinasi produk penjualan dari data transaksi yang ada, bisa menguntungkan toko daring karena barang yang dikombinasikan berdasarkan barang yang paling sering dibeli oleh pembeli. Rekomendasi kombinasi

produk (*itemset*) tersebut dapat menggunakan metode *Market Basket Analysis* karena mendukung sistem rekomendasi melalui penemuan pola antar *item* dalam transaksi-transaksi yang terjadi [4].

*Market Basket Analysis* ini dianggap mampu memberikan jalan keluar dalam merekomendasikan kombinasi kategori produk yang terkait dengan penggunaan algoritma Apriori melalui metode *Association Rules* [5]. Algoritma Apriori adalah suatu metode di mana akan mencari pola hubungan antar satu atau 3 lebih *item* dalam sebuah *dataset* [6]. Kelebihan algoritma Apriori adalah jumlah kandidat yang harus dihitung *support*-nya dapat dikurangi dengan cara pemangkasan sehingga memiliki performa yang baik [7].

Pemahaman dan definisi yang tepat akan suatu masalah merupakan hal yang penting dalam menciptakan suatu solusi *software* yang tepat. Untuk itu, diperlukan proses-proses untuk menemukan, memahami, memformulasi, menganalisis, dan menemui persetujuan terkait masalah apa yang harus diselesaikan, kenapa masalah tersebut harus diselesaikan, dan siapa yang harus terlibat dalam penyelesaian masalah tersebut. Secara umum, proses analisis kebutuhan meliputi hal-hal yang telah disebutkan di atas

Penelitian terkait yang pernah dilakukan, dihasilkan sejumlah kombinasi *item set* yang cocok untuk pemasaran ini dan kekuatan hubungan antar *item* dapat dipercaya dan dijadikan acuan karena ketepatan kombinasi *item set* yang dihasilkan valid karena semua nilai *lift ratio*-nya  $> 1$  [8]. *Lift ratio* digunakan untuk mengukur seberapa penting aturan atau yang telah terbentuk agar dapat dipercaya sepenuhnya [9].

## II. LANDASAN TEORI

### A. *Market Basket Analysis*

*Market Basket Analysis* digunakan untuk mengetahui produk yang dibeli secara bersamaan oleh pelanggan agar dapat meningkatkan taktik pemasaran dan penjualan di tingkat toko [10]. Istilah *Market*

*Basket Analysis* masuk dalam kategori *Association Rules*. *Association Rule* adalah sebuah ekspresi implikasi dari bentuk  $A \rightarrow B$ , di mana  $A$  dan  $B$  adalah itemset yang saling terpisah (*disjoint*), dengan kata lain  $A \cap B = \emptyset$ . Dalam menentukan *Association Rule*, terdapat suatu ukuran ketertarikan yang didapatkan dari hasil pengolahan data dengan perhitungan tertentu, yaitu sebagai berikut [11].

1. *Support*: bagian transaksi yang mengandung kedua  $A$  dan  $B$ .

$$\text{Support}(A) = \frac{\text{Jumlah transaksi mengandung } A}{\text{Total transaksi}} \quad (1)$$

Atau jika terdapat dua buah item dalam  $A$ , nilai *support* diperoleh dari rumus sebagai berikut.

$$\text{support}(A) = \frac{\text{Jumlah transaksi mengandung } A}{\text{Total transaksi}} \quad (2)$$

2. *Confidence*: seberapa sering item dalam  $B$  muncul di transaksi yang mengandung  $A$ .

$$\text{Confidence } P(A \setminus B) = \frac{\text{Jumlah transaksi mengandung } A \text{ dan } B}{\text{Total transaksi mengandung } A} \quad (3)$$

#### B. Apriori

Apriori adalah salah satu pendekatan yang sering digunakan pada *Frequent Itemset Mining*. Prinsip algoritma Apriori adalah jika sebuah *itemset infrequent*, maka *itemset* yang *infrequent* tidak perlu lagi dieksplor *superset* sehingga jumlah kandidat yang harus diperiksa menjadi berkurang, berikut adalah ilustrasinya [11].

Langkah-langkah Apriori adalah sebagai berikut [12]:

1. Tentukan minimum *support* (nilai minimum dari kombinasi item).
2. Pembentukan kandidat *itemset* ( $C_k$ ) dibangun dengan menggabungkan  $L_{k-1}$ , dengan dirinya sendiri.
3. Pemangkasan kandidat *itemset* dilakukan setiap  $C_k$ -*itemset* yang tidak sering muncul (*not frequent*) tidak dapat menjadi *subset* dari *frequent k-itemset*.
4. Menentukan *support* dari kombinasi *k-itemset* sebelumnya. *Itemset* yang memenuhi *minimum support* akan dipilih sebagai pola *frequent* tinggi ( $L_k$ ) dari kandidat.

5. Tetapkan nilai *k-itemset* dari *support* yang telah memenuhi *minimum support* dari *k-itemset*.
6. Jika sudah tidak ada lagi *k-itemset* yang memenuhi *minimum support* maka proses dihentikan.

#### C. Lift Ratio

*Lift ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar item  $A$  dibeli bersamaan item  $B$ , dengan rumus sebagai berikut [9].

$$\text{Nilai lift} = \frac{\text{Support}(A \cap B)}{\text{Support}(A) \times \text{Support}(B)} \quad (4)$$

Sebuah transaksi dikatakan valid, jika mempunyai nilai *lift ratio* lebih dari 1, yang berarti bahwa dalam transaksi tersebut item  $A$  dan item  $B$  benar-benar dibeli secara bersamaan [9].

#### D. White Box

*White Box* digunakan untuk menguji suatu aplikasi di mana struktur internal telah diketahui oleh pengujian [13]. Ada beberapa cara dalam melakukan uji coba *white box*, salah satunya adalah *Unit Testing*. *Unit Testing* merupakan pengujian dilakukan pada setiap modul atau blok kode selama pengembangan. Pengujian ini biasa dilakukan oleh *developer* yang menulis kode [13]. Teknik pada *Unit Testing* ada beberapa, salah satunya adalah *Path Coverage*. *Path Coverage* merupakan uji coba untuk memastikan bahwa setiap alur program dapat dilalui setidaknya satu kali [14].

#### E. USE Questionnaire

*USE questionnaire* adalah salah satu paket kuesioner yang dapat digunakan untuk mengevaluasi *website* yang dibangun dari segi *usability*. Secara umum, *usability* mengacu pada sejauh mana kepuasan *user* dan sejauh mana *user* dapat belajar dan menggunakan suatu produk untuk mencapai tujuannya [15]. *USE* memiliki beberapa aspek yaitu, *usefulness*, *ease of use*, *ease of learning*, *satisfaction*. Meskipun ditemukan beberapa dimensi lain, tetapi aspek ini merupakan yang paling mudah diamati dan dibandingkan hasilnya [15].

Lalu paket kuesioner yang akan digunakan selengkapnya dapat dilihat pada tabel di bawah ini [16].

Tabel 1. Kriteria pengukuran *USE questionnaire*

No	Kriteria
	<i>Usefulness</i>
U1	<i>It helps me be more effective.</i>
U2	<i>It helps me be more productive.</i>

No	Kriteria
	<b>Usefulness</b>
U3	<i>It is useful.</i>
U4	<i>It gives me more control over the activities in my life.</i>
U5	<i>It make the things I want to accomplish easier to get done.</i>
U6	<i>It saves me time when I use it.</i>
U7	<i>It meets me needs.</i>
U8	<i>It does everything I would expect it to do.</i>
	<b>Ease of Use</b>
EU1	<i>It is easy to use.</i>
EU2	<i>It is simple to use.</i>
EU3	<i>It is user friendly.</i>
EU4	<i>It requires the fewest steps possible to accomplish what I want to do with it.</i>
EU5	<i>It is flexible.</i>
EU6	<i>Using it is effortless.</i>
EU7	<i>I can use it without written instructions.</i>
EU8	<i>I don't notice any inconsistencies as I use it.</i>
EU9	<i>Both occasional and regular users would like it.</i>
EU10	<i>I can recover from mistakes quickly and easily.</i>
EU11	<i>I can use it successfully every time.</i>
	<b>Ease of Learning</b>
EL1	<i>I learned to use it quickly.</i>
EL1	<i>I easily remember how to use it.</i>
EL1	<i>It is easy to learn to use it.</i>
EL1	<i>It is quickly became skillful with it.</i>
	<b>Satisfaction</b>
S1	<i>I am satisfied with it.</i>
S2	<i>I would recommend it to a friend.</i>
S3	<i>It is fun to use.</i>
S4	<i>It works the way I want it to work.</i>
S5	<i>It is wonderful.</i>
S6	<i>I feel I need to have it.</i>
S7	<i>It is pleasant to use.</i>

#### F. Skala Likert

Evaluasi *website* dalam penelitian ini menggunakan *USE questionnaire*. Kuesioner ini menggunakan skala Likert sebagai pengukuran aspeknya. Skala Likert digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang kejadian atau fenomena sosial [17]. Skala Likert menggunakan beberapa butir pertanyaan untuk mengukur perilaku individu dengan merespon lima titik pilihan pada setiap butir

pertanyaan, sangat setuju (SS), setuju (S), netral (N), tidak setuju (TS) dan sangat tidak setuju (STS) [18].

Tabel 2. Skor skala Likert

Angka (%)	Jawaban
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Persentase skor pada pertanyaan kuesioner dapat dihitung dengan menggunakan Rumus 5 [17].

$$\frac{(SS * 5) + (S * 4) + (N * 3) + (TS * 2) + (ST * 1)}{S * (\text{Jumlah responden})} * 100\% \quad (5)$$

Tabel 3. Kategori kelayakan

Angka (%)	Jawaban
<21	Sangat Tidak Layak
21-40	Tida Layak
41-60	Cukup
61-80	Layak
81-100	Sangat Layak

Persentase skor yang diperoleh kemudian akan disesuaikan nilai yang didapat berdasarkan Tabel 3 [16].

### III. METODOLOGI PENELITIAN

#### A. Telaah Literatur

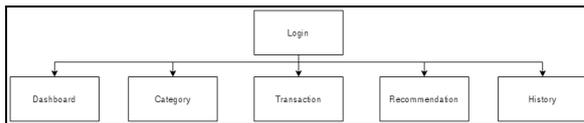
Bentuk kegiatan dalam pengumpulan berbagai informasi pada tahap ini mulai dari jurnal-jurnal, buku ataupun referensi penelitian-penelitian yang berkaitan dengan penelitian yang akan dilakukan. Tahapan ini adalah proses awal penelitian yang akan dilakukan.

#### B. Pengumpulan Data

Data yang dikumpulkan dalam penelitian ini berasal dari data transaksi yang tercatat pada PT Grup Gaga Indonesia. Data yang diperoleh adalah transaksi dalam jangka waktu 2 tahun dimulai dari 1 Januari 2016 sampai 31 Desember 2018 dengan total jumlah data adalah 529 transaksi.

#### C. Perancangan

Gambar 1 merupakan *sitemap* dari *website* yang akan digunakan oleh pemilik toko daring.



Gambar 1. Sitemap

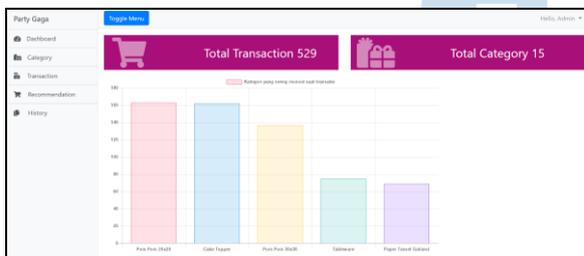
Perancangan *website* dilakukan dengan membuat struktur halaman *website*. Halaman *login* adalah halaman pertama saat *admin* membuka *website*. Setelah *login*, *admin* dapat bernavigasi ke halaman *dashboard*, *category*, *transaction*, *recommendation*, dan *history*.

D. Uji Coba dan Evaluasi

Pada tahap ini, dilakukan pembentukan *association rules* dengan algoritma Apriori. Adapun langkah-langkah yang dilakukan sesuai dengan yang dijelaskan pada Sub-bab II pada Landasan Teori. Perhitungan *lift ratio* dilakukan untuk mengetahui tingkat korelasi antar produk.

IV. IMPLEMENTASI DAN UJI COBA

A. Implementasi



Gambar 2. Halaman *dashboard*

Gambar 2 merupakan halaman *dashboard*. Terdapat *sidebar* pada bagian sebelah kiri dan menampilkan total transaksi, total kategori dan grafik pada bagian tengah.

Name	Code
Balloon	BAL
Banner	BAN
Cake Topper	BKD
Candle	BND
Centrepiece	BRC
Confetti	BRC
Glass	BOD
Paper Fan 20	BRC
Paper Fan 30	BRC
Paper Garland	BRC

Gambar 3. Halaman *category*

Gambar 3 merupakan halaman *category*. Halaman ini menampilkan tabel data kategori.

Transaksi ID	Date	Name
2000001	2016-07-29	Candle
2000002	2016-08-04	Candle
2000003	2016-08-05	Pom Pom 20x30
2000004	2016-08-10	Candle
2000005	2016-08-10	Paper Tassel Garland Pom Pom 20x30
2000006	2016-08-11	Candle
2000007	2016-08-12	Pom Pom 20x30
2000008	2016-08-16	Cake Topper
2000009	2016-08-16	Pom Pom 30x30
2000010	2016-08-19	Candle

Gambar 4. Halaman *transaction*

Gambar 4 merupakan halaman *transaction*. Halaman ini menampilkan tabel data transaksi.

The recommendation page includes input fields for 'Category', 'Database transaction until', 'Minimum Support Value (1-100)', and 'Minimum Confidence Value (1-100)'. It also contains an 'Information' section explaining the recommendation process.

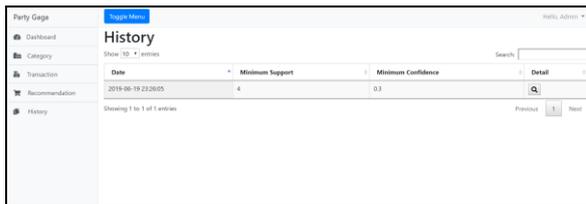
Gambar 5. Halaman *recommendation*

Gambar 5 merupakan halaman *recommendation*. Halaman ini menampilkan *field input* tanggal transaksi dari, tanggal transaksi sampai, nilai *minimum support*, nilai *minimum confidence*, serta terdapat keterangan yang menjelaskan istilah-istilah.

Item	Support Count	Support
Pom Pom 20x30	163	0.31
Cake Topper	162	0.31
Pom Pom 30x30	137	0.26
Candle	75	0.14
Paper Tassel Garland	69	0.13
Spokecap	49	0.09
Balloon	36	0.07

Gambar 6. Halaman hasil *recommendation*

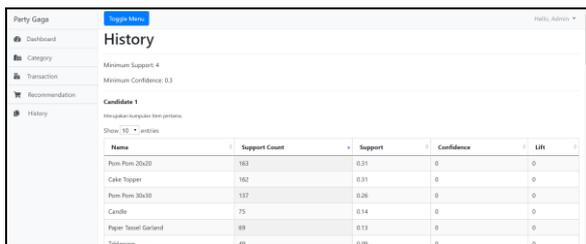
Gambar 6 merupakan halaman hasil *recommendation*. Halaman ini menampilkan hasil dari proses yang dilakukan.



Gambar 7. Halaman *history*

Gambar 7 merupakan halaman *history*. Halaman ini menampilkan tabel dari setiap proses yang dilakukan. Ikon detail akan menavigasi halaman ke *history detail*.

Jika	Support A	Maka	Support C	Support Count	Support (A,C)	Conf	Lift
Pom Pom 20x20	0,31	Pom Pom 30x30	0,26	100	0,19	0,61	2,37
Cake Topper	0,31	Pom Pom 30x30	0,26	40	0,08	0,29	0,95
Pom Pom 20x20 Pom Pom 30x30	0,19	Cake Topper	0,31	36	0,07	0,36	1,18



Gambar 8. Halaman *history detail*

Gambar 8 merupakan halaman *history detail*. Halaman ini menampilkan data *history* dari setiap *history* yang dipilih.

**B. Hasil Rekomendasi**

*Dataset* menggunakan data sebesar 529 data transaksi dengan *minimum support* sebesar 4 dan *minimum confidence* 0.3. Lalu *item* yang *support count* tidak memenuhi *minimum support* akan dihapus.

Tabel 5 menampilkan rekap nilai hasil perhitungan terhadap algoritma Apriori. Pengambilan rekomendasi kombinasi produk merupakan perhitungan dari nilai *support*, *confidence* dan *lift ratio*. Kolom *support count* menunjukkan jumlah kemunculan suatu aturan (*association rules*). Kolom *support* menunjukkan nilai dari aturan yang dihasilkan. Kolom *confidence* menunjukkan suatu nilai kepastian atau kehandalan dari aturan yang dibuat. Kemudian, nilai *lift ratio* menunjukkan nilai evaluasi kuat tidaknya sebuah aturan. Kombinasi atau aturan yang muncul telah memenuhi nilai *minimum support* 4, *minimum confidence* 0.3, dan memenuhi nilai *lift ratio* > 1.

Tabel 5. Hasil perhitungan

Jika	Support A	Maka	Support C	Support Count	Support (A,C)	Conf	Lift
Pom Pom 20x20	0,31	Cake Topper	0,31	42	0,08	0,26	0,84

**C. Uji Coba White Box**

Uji coba dilakukan pada halaman *Recommendation* dengan *Unit Testing*. Teknik *Unit Testing* yang akan digunakan adalah *Path Testing*. Setiap uji coba yang dilakukan berhasil dijalankan sehingga kode pada halaman *Recommendation* memiliki 100 persen *path coverage*.

**D. Evaluasi Sistem**

Dari keempat aspek yang telah diperoleh hasil nilainya, aspek *ease of use* merupakan skor paling rendah dengan 89.09% yang dikarenakan setengah dari responden pada pertanyaan nomor enam memilih skor netral. Kemudian aspek *usefulness*, *ease of learning*, *satisfaction* memperoleh persentase skor lebih dari 90% yang bisa disimpulkan bahwa hampir semua responden menyatakan sangat layak berdasarkan Tabel 3 atau sangat setuju pada setiap aspek.

Tabel 6. Hasil evaluasi system

Aspek	Perhitungan Skor
<i>Usefulness</i> (Kegunaan)	90,83%
<i>Ease of Use</i> (Kemudahan Penggunaan)	89,09%
<i>Ease of Learning</i> (Kemudahan Belajar)	95%
<i>Satisfaction</i> (Kepuasan)	90,94%

**V. SIMPULAN**

Sistem rekomendasi kombinasi produk yang sering dibeli secara bersamaan yaitu Pom Pom 20x20, Pom Pom 30x30 dan *Cake Topper*. *Website* ini dilakukan pengujian menggunakan *white box* untuk memastikan alur program dapat dilalui setidaknya satu kali dan dievaluasi menggunakan *USE Questionnaire* dengan hasil pada *usefulness* 90.83%, *ease of use* 89.09%, *ease of learning* 95% dan *satisfaction* 90.94% yang menyatakan sangat layak atau setuju pada setiap aspek. Nilai *lift ratio* yang dihasilkan dari aturan menunjukkan 1.18 dari Pom

Pom 20x20, Pom Pom 30x30 dan *Cake Topper* yang artinya bahwa aturan tersebut memiliki kekuatan hubungan antar *item* sehingga dapat dipercaya untuk dijadikan kombinasi *item*.

#### DAFTAR PUSTAKA

- [1] PT. Grup Gaga Indonesia, 2018. Katalog online.
- [2] A. Kusnadi, J. Pratama, "Implementasi Algoritma Genetika dan Neural Network Pada Aplikasi Peramal Produksi Mie (Studi Kasus: Omega Mie Jaya)", *ULTIMATICS*, Vol. IX, No 1, 2017.
- [3] A. Rusli, "Ekstrasi Kebutuhan Aplikasi Berdasarkan Feedback Pengguna Menggunakan Naïve Bayes dan Gamifikasi", *ULTIMATICS*, Vol X, No 1, 2018.
- [4] A. Masnur, "Analisa Data Mining Menggunakan Market Basket Analysis untuk Mengetahui Pola Beli Konsumen" Vol. 1, No. 2, 2015.
- [5] M. Riadi. *kajianpustaka.com*. 2017. [Online] Tersedia dalam: <https://www.kajianpustaka.com/2017/09/data-mining.html> [Diakses 21 Juni 2018].
- [6] H.D. Anggraeni, "Aplikasi Data Mining Analisis Data Transaksi Penjualan Obat Menggunakan Algoritma Apriori (Studi Kasus di Apotek Setya Sehat Semarang)" Volume 4, 2013.
- [7] O. S. A. Destiyati, E. Aribowo, "Analisis Perbandingan Algoritma Apriori dan Algoritma Hash Based pada Market Basket Analysis di Apotek UAD", *Jurnal Sarjana Teknik Informatika*, Vol 3 No 1, 2015.
- [8] D. Fitriati, "Implementasi Data Mining untuk Menentukan Kombinasi Media Promosi Barang Berdasarkan Perilaku Pembelian Pelanggan Menggunakan Algoritma Apriori", *Annual Research Seminar*, 2016.
- [9] E. Widiati, "Implementasi Association Rule Terhadap Penyusunan Layout Makanan dan Penentuan Paket Makanan Hemat di RM Roso Echo Dengan Algoritma Apriori" *Jurnal Ilmiah Komputer dan Informatika*, Vol 3, No 2, 2013.
- [10] M. Kaur, S. Kang, "Market Basket Analysis: Identify the changing trends of market data using association rule mining", *International Coference on Computational Modeling and Security*, 2016.
- [11] Christie, D. A., Baskoro, D.A., Ambarwati, L. & Wicaksana I.W.S. (2013). *Belajar Data Mining dengan Rapid Miner*, Jakarta.
- [12] Larose, D.T. (2014). *Discovering Knowledge in Data: An Introduction to Data Mining*.
- [13] P. Kurniawati. *medium.com*. 2018. [Online] Tersedia dalam : <https://medium.com/skyshidigital/pengujian-sistem-52940ee98c77> [Diakses 3 Januari 2019].
- [14] *Softwaretestinghelp*. 2019. *softwaretestinghelp.com* [Online] Tersedia dalam: <https://www.softwaretestinghelp.com/white-box-testing-techniques-with-example/>.
- [15] K. Aelani, Falahah, "Pengukuran Usability Sistem Menggunakan USE Questionnaire (Studi Kasus Perwalian Online STMIK "AMIKBANDUNG")", *Seminar Nasional Aplikasi Teknologi Informasi*, 2012.
- [16] W. Kusuma, dan G. I. Marthasari, "Analisis Usability dalam User Experience pada Sistem KRS Online UMM menggunakan USE Questionnaire", *ResearchGate*, 2016.
- [17] Sugiyono, "Metodologi Penelitian Kuantitatif, Kualitatif dan R&D", Bandung: Alfabeta, 2012.
- [18] Budiaji, W. (2013). Skala Pengukuran dan Jumlah Respon Skala Likert, *Jurnal Ilmu Pertanian dan Perikanan*, Volume 2.
- [19] Sharma, M., Choudhary, J., Sharma, G, "Evaluating The Performance of Apriori and Predictive Apriori Algorithm To Find New Association Rules Based On The Statiscal Measures of Datasets.", *International Jounrak of engineering Research & Technology*, Vol 1 Issue 6, 2012.



UMN

# 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.\(URL\)](http://www.(URL))

N. Penulis. (tahun, bulan). Judul. Jurnal [Media perantara]. *volume(issue)*, halaman jika ada. Alamat situs: [http://www.\(URL\)](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](mailto: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: [ftijurnal@umn.ac.id](mailto:ftijurnal@umn.ac.id)

# Judul Paper

Sub Judul (jika diperlukan)

Nama Penulis A<sup>1</sup>, Nama Penulis B<sup>2</sup>, Nama Penulis C<sup>2</sup>

<sup>1</sup> Baris pertama (dari afiliasi): nama departemen organisasi, nama organisasi, kota, negara  
Baris kedua: alamat surel jika diinginkan

<sup>2</sup> 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 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, identifikasi 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/m<sup>2</sup>” atau “webers per meter persegi,” bukan “webers/m<sup>2</sup>.”
- 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*.

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

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 mengidentifikasi 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 penomorannya mengikuti cara penamaan dan penomoran subbab diikuti angka Arab, seperti "A.1 Penulis", "A.1.1 Afiliasi Penulis".

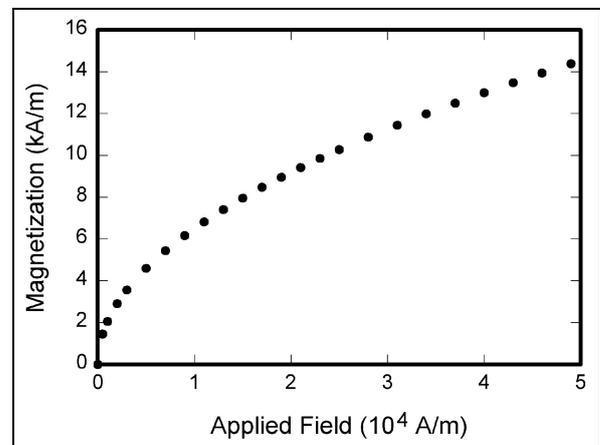
##### C. Gambar dan Tabel

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 ....”

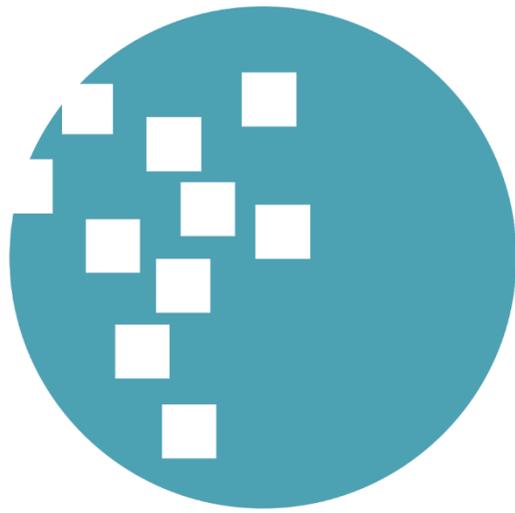
Penomoran catatan kaki dilakukan secara terpisah dengan *superscripts*. Letakkan catatan kaki tersebut di

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. Magnetism Japan, hal. 301, 1982].
- [7] M. Young, *The Technical Writer’s Handbook*. Mill Valley, CA: University Science, 1989.



**UMN**

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

ISSN 2085-4552



9 772085 455006



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