

ULTIMATICS

Jurnal Teknik Informatika

MARVIN APRIYADI, SENG HANSUN

Sistem Pendukung Keputusan Penerima Beasiswa UMN dengan Profile Matching (1-6)

FELIX INDRA KURNIADI, VINNIA KEMALA PUTRI

Perbandingan Regresi Linear dengan Heaviside Activation Function dengan Logistic Regression untuk Klasifikasi Diabetes (7-10)

VINNIA KEMALA PUTRI, FELIX INDRA KURNIADI

Klasifikasi Diabetes Menggunakan Model Pembelajaran Ensemble Blending (11-15)

W. S. RAHARJO, GANI INDRIYANTA, AMSAL MAESTRO

Analisis Kesiapan Kebutuhan Infrastruktur Replikasi Basis Data pada Sekolah Musik Indonesia Solo (16-25)

RUDI SUTOMO, JOHNY HIZKIA SIRINGO RINGO

Sistem Penunjang Keputusan Penilaian Kinerja Karyawan Berprestasi Menggunakan Acuan MBO dan Metode AHP Menggunakan Aplikasi Expert Choice (Studi Kasus Universitas Tanri Abeng) (26-33)

ANDRE RUSLI

Ekstraksi Kebutuhan Aplikasi Berdasarkan Feedback Pengguna Menggunakan Naïve Bayes dan Gamifikasi (34-40)

RICKY SURYA, DENNIS GUNAWAN

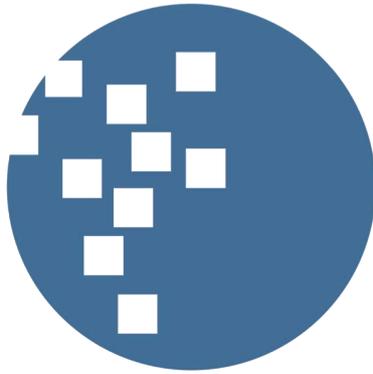
Situsparu: Sistem Pakar Untuk Deteksi Penyakit Tuberkulosis Paru (41-47)

BETTI NOVIYANI, EKO RUDI SETIAWAN

Aplikasi Survei Ubinan Berbasis Android (48-56)

R. R. BINTANA, C. FATICHAH, D. PURWITASARI

Pencarian Question-Answer Menggunakan Convolutional Neural Network Pada Topik Agama Berbahasa Indonesia (57-64)



UMN

SUSUNAN REDAKSI

Pelindung

Dr. Ninok Leksono

Penanggungjawab

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

Pemimpin Umum

Maria Irmina Prasetyowati, S.Kom., M.T.

Mitra Bestari

(UMN) Dr. Rangga 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) Gamaliel Cahya Kristanto, S.Kom., M.M.S.I.
(UMN) Marcel Bonar Kristanda, S.Kom., M.Sc.
(UMN) Nunik Afriliana, S.Kom., M.M.S.I.
(UMN) Ranny, S.Kom., M.Kom.
(UMN) Seng Hansun, S.Si., M.Cs.
(UMN) Yustinus Widya Wiratama, S.Kom., M.Sc., OCA
(UMN) Felix Lokananta, S.Kom., M.Eng.Sc.
(UMN) Wella, S.Kom., M.MSI., COBIT5
(UMN) Andre Rusli, S.Kom., M.Sc.
(UMN) Julio Christian Young, S.Kom.

Ketua Dewan Redaksi

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

Dewan Redaksi

Andre Rusli, S.Kom., M.Sc.
Wella, S.Kom., M.MSI., COBIT5

Desainer & Layouter

Wella, S.Kom., M.MSI., COBIT5

Sirkulasi dan Distribusi

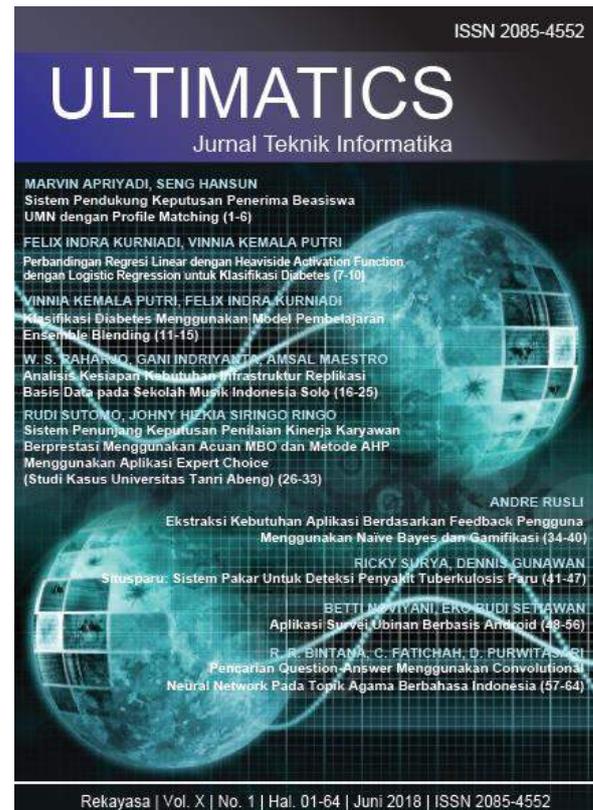
Sularmin

Kuangan

I Made Gede Suteja, S.E.

ALAMAT REDAKSI

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



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

Call for Paper



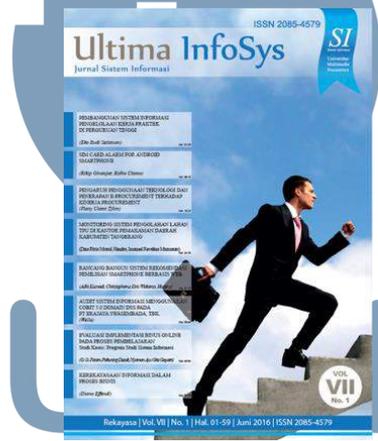
International Journal of New Media Technology (IJNMT) is a scholarly open access, peer-reviewed, and interdisciplinary journal focusing on theories, methods and implementations of new media technology. IJNMT is published annually by Faculty of Engineering and Informatics, Universitas Multimedia Nusantara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

Important Dates

- October 31st, 2018**
Deadline for submission of papers
- November 30th, 2018**
Announcement for Acceptance
- December 14th, 2018**
Deadline for submission of final papers



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



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



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

DAFTAR ISI

Sistem Pendukung Keputusan Penerima Beasiswa UMN dengan Profile Matching	
Marvin Apriyadi, Seng Hansun	1-6
Perbandingan Regresi Linear dengan Heaviside Activation Function dengan Logistic Regression untuk Klasifikasi Diabetes	
Felix Indra Kurniadi, Vinnia Kemala Putri	7-10
Klasifikasi Diabetes Menggunakan Model Pembelajaran Ensemble Blending	
Vinnia Kemala Putri, Felix Indra Kurniadi	11-15
Analisis Kesiapan Kebutuhan Infrastruktur Replikasi Basis Data pada Sekolah Musik Indonesia Solo	
Willy Sudiarto Raharjo, Gani Indriyanta, Amsal Maestro	16-25
Sistem Penunjang Keputusan Penilaian Kinerja Karyawan Berprestasi Menggunakan Acuan MBO dan Metode AHP Menggunakan Aplikasi Expert Choice (Studi Kasus Universitas Tanri Abeng)	
Rudi Sutomo, Johny Hizkia Siringo Ringo	26-33
Ekstraksi Kebutuhan Aplikasi Berdasarkan Feedback Pengguna Menggunakan Naïve Bayes dan Gamifikasi	
Andre Rusli	34-40
Situsparu: Sistem Pakar Untuk Deteksi Penyakit Tuberkulosis Paru	
Ricky Surya, Dennis Gunawan	41-47
Aplikasi Survei Ubinan Berbasis Android	
Betti Noviyani, Eko Budi Setiawan	48-56
Pencarian Question-Answer Menggunakan Convolutional Neural Network Pada Topik Agama Berbahasa Indonesia	
Rizqa Raaiqa Bintana, Chastine Fatichah, Diana Purwitasari	57-64

KATA PENGANTAR

Salam ULTIMA!

ULTIMATICS – Jurnal Teknik Informatika UMN kembali menjumpai para pembaca dalam terbitan saat ini Edisi Juni 2018, Volume X, 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 2018 ini, terdapat sepuluh artikel ilmiah yang berasal dari para peneliti, akademisi, dan praktisi di bidang Teknik Informatika, yang mengangkat beragam topik, antara lain: Sistem Pendukung Keputusan Penerima Beasiswa UMN dengan Profile Matching; Perbandingan Regresi Linear dengan Heaviside Activation Function dengan Logistic Regression untuk Klasifikasi Diabetes; Klasifikasi Diabetes Menggunakan Model Pembelajaran Ensemble Blending; Analisis Kesiapan Kebutuhan Infrastruktur Replikasi Basis Data pada Sekolah Musik Indonesia Solo; Sistem Penunjang Keputusan Penilaian Kinerja Karyawan Berprestasi Menggunakan Acuan MBO dan Metode AHP Menggunakan Aplikasi Expert Choice (Studi Kasus Universitas Tanri Abeng); Ekstraksi Kebutuhan Aplikasi Berdasarkan Feedback Pengguna Menggunakan Naïve Bayes dan Gamifikasi; Situsparu: Sistem Pakar Untuk Deteksi Penyakit Tuberkulosis Paru; Aplikasi Survei Ubinan Berbasis Android; Pencarian Question-Answer Menggunakan Convolutional Neural Network Pada Topik Agama Berbahasa Indonesia.

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

Akhir kata, kami mengucapkan terima kasih kepada seluruh kontributor dalam ULTIMATICS Edisi Juni 2018 ini. Kami berharap artikel-artikel ilmiah hasil penelitian dalam jurnal ini dapat bermanfaat dan memberikan sumbangsih terhadap perkembangan penelitian dan keilmuan di Indonesia.

Juni 2018,

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

Sistem Pendukung Keputusan Penerima Beasiswa UMN dengan Profile Matching

Marvin Apriyadi¹, Seng Hansun²

Program Studi Teknik Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
m.a_vin@hotmail.com
hansun@umn.ac.id

Diterima 2 Januari 2018

Disetujui 8 Juni 2018

Abstract—This paper describes about the design and development of UMN scholarship decision support system using Profile Matching method. At Universitas Multimedia Nusantara (UMN), there are scholarships for students in order to ease the burden of education costs. There are achievement scholarships, scholarships for students whose parents died, and non-academic scholarships. Decision support systems play an important role in the termination of the final decision, and therefore this application is built by implementing the method of Profile Matching to aid in the selection of the scholarship decisions for students whose parent were died and non-academic scholarship. Profile Matching is a method that aims at taking decisions by assuming that there is an ideal level of predictor variables that must be met by the subjects studied, instead of the minimum rate that must be met or passed. From the results, it can be concluded that this method successfully implemented into applications that can help in decision making process. Result of user satisfaction level for this application is 72%, the result of manual calculation with the application calculation results is not much different so that this application can assist in supporting the UMN scholarship decision.

Index Terms— decision support system, non-academic scholarship, Profile Matching, scholarships for student whose parents died, UMN

I. INTRODUCTION

Universitas Multimedia Nusantara (UMN) adalah sebuah perguruan tinggi swasta yang berdiri pada tahun 2006 dan berada di daerah Summarecon Serpong, Tangerang. Pada perguruan tinggi swasta ini, terdapat beasiswa untuk mahasiswa agar dapat meringankan beban biaya pendidikan. Di UMN terdapat beasiswa prestasi, beasiswa untuk orang tua yang meninggal, dan beasiswa non-akademik.

Perguruan tinggi akan memberikan beasiswa kepada mahasiswa yang kurang mampu ataupun berprestasi. Kriteria yang dipakai untuk mendapatkan beasiswa adalah sebagai berikut: indeks prestasi kumulatif (IPK), penghasilan orang tua, tanggungan orang tua, dan semester. Pemilihan beasiswa bisa dilakukan secara terkomputerisasi maupun tidak. Pada beasiswa prestasi di UMN, sudah dilakukan secara terkomputerisasi. Pada sistem saat ini, data mahasiswa yang mengajukan dan data mahasiswa

yang mendapatkan beasiswa masih disimpan dalam bentuk kertas. Hal ini menyebabkan kekurangan-kekurangan seperti keterbatasan dalam mengakses data, dan juga sulit untuk mencari data salah satu mahasiswa jika data yang tersimpan sudah sangat banyak.

Pada pemilihan secara terkomputerisasi, dapat menggunakan metode *Profile Matching* untuk membantu pemilihan keputusan beasiswa di UMN. Walaupun pemilihan tetap ditentukan sepenuhnya oleh pihak UMN, namun sistem pendukung keputusan ini akan menampilkan prioritas-prioritas tertinggi hingga terendah, sehingga akan memudahkan dan membantu pihak UMN dalam mengambil keputusan.

Setelah mewawancarai Ibu Lindi sebagai *Student Support officer* yang bertanggung jawab dalam mendapatkan data mahasiswa untuk kepentingan beasiswa orang tua meninggal, diketahui bahwa saat ini data masih dalam bentuk *form* yang diisi saat konsultasi dengan mahasiswa tersebut. Maka dari itu, dibutuhkan suatu sistem yang memudahkan dalam merekap data mahasiswa. Selain itu, peneliti mewawancarai Ibu Devi sebagai *Student Development officer* yang bertanggung jawab dalam mendapatkan data untuk kepentingan beasiswa non-akademik, dan diketahui bahwa di tahun 2016 semua data baru disimpan dalam bentuk digital.

Pada penelitian sebelumnya dengan judul "Penerapan Profile Matching Untuk Pencarian Siswa SMP Penerima Beasiswa Miskin dan Berprestasi" oleh Irawan dan Kriestanto [1], diperoleh kesimpulan bahwa tidak ada perbedaan signifikan antara hasil perhitungan secara manual maupun dengan menggunakan aplikasi yang sudah dibuat pada penelitian tersebut. Metode tersebut berhasil diterapkan pada beasiswa di sekolah SMP Negeri 1 Toili. Pada penelitian lainnya dengan judul "Penerapan Metode Profile Matching dalam Sistem Pendukung Keputusan Penilaian Kinerja Karyawan (Studi Kasus: PT. Perkebunan Nusantara III Medan)" oleh Puspitasari [2], ditarik kesimpulan bahwa proses penilaian kinerja karyawan pada perusahaan tersebut belum dilakukan secara efektif dan efisien, sehingga

perlu adanya sistem yang baru untuk mempermudah dalam pengambilan keputusan. Metode tersebut juga berhasil diterapkan pada penilaian kinerja karyawan di suatu perusahaan.

Berdasarkan latar belakang permasalahan akan kebutuhan atas suatu sistem pendukung keputusan dalam menentukan mahasiswa penerima beasiswa UMN, khususnya beasiswa bagi mahasiswa yang orang tuanya meninggal dan beasiswa non-akademik, yang dapat mempermudah pihak kemahasiswaan UMN, maka penelitian tentang rancang bangun sistem pendukung keputusan beasiswa UMN dengan metode Profile Matching ini dilakukan.

II. SISTEM PENDUKUNG KEPUTUSAN

Menurut Alter dalam Kusrini [3], sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan manipulasi data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. Menurut Moore dan Chang [4], sistem pendukung keputusan dapat digambarkan sebagai sistem yang berkemampuan mendukung analisis data, dan pemodelan keputusan, berorientasi keputusan, orientasi perencanaan masa depan, dan digunakan saat-saat yang tidak biasa.

Dengan pengertian Moore dan Chang [4], diambil suatu kesimpulan bahwa sistem pendukung keputusan bukan merupakan alat pengambilan keputusan, melainkan sistem yang membantu pengambilan keputusan untuk melengkapi informasi dari data yang telah diolah secara relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat. Fungsi sistem pendukung keputusan adalah untuk meningkatkan kemampuan para pengambil keputusan dengan memberikan alternatif-alternatif keputusan yang lebih banyak atau lebih baik, sehingga dapat membantu untuk merumuskan masalah dan keadaan yang dihadapi. Sistem pendukung keputusan dapat dikatakan secara singkat bertujuan untuk meningkatkan efektivitas dan efisiensi dalam pengambilan keputusan.

III. PROFILE MATCHING

Profile Matching adalah sebuah metode yang bertujuan mengambil keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dipenuhi oleh subyek yang diteliti, bukannya tingkat minimal yang harus dipenuhi atau dilewati [5]. Contoh yang dapat diimplementasikan seperti evaluasi kinerja karyawan untuk promosi jabatan, manajemen pemain bola, dan penerimaan beasiswa yang layak.

Berdasarkan penjelasan yang dibuat oleh Setiyaningstih [5], terdapat beberapa langkah perhitungan untuk mendapatkan hasil keputusan, yaitu:

1. Menetapkan aspek-aspek yang akan digunakan. Aspek-aspek akan didapatkan dari perusahaan atau atasan yang bertanggung jawab dalam memilih keputusan (contoh: Aspek Kecerdasan, Aspek Perilaku, dan Aspek Sikap Kerja).
 2. Menetapkan penilaian skala ordinal. Skala ordinal adalah skala yang membedakan kategori berdasarkan tingkat atau urutan (contoh: tinggi, sedang, dan pendek).
 3. Menetapkan nilai target pada setiap sub-aspek yang sudah ditentukan (contoh dalam Aspek Kecerdasan, 1. Common Sense 3, 2. Sistematika Berpikir 4, 3. Konsentrasi 3).
 4. Melakukan pemetaan *Gap* (perbedaan) kompetensi setelah memiliki data kandidat-kandidat. Dalam proses ini akan menentukan kandidat mana yang paling cocok menduduki tempat sebagai orang yang memiliki kinerja baik untuk mendapatkan beasiswa. Perhitungan pemetaan *gap* kompetensi ditunjukkan pada rumus di bawah ini:
- $$Gap = Value Atribut - Value Target \quad (1)$$
5. Setelah mendapatkan nilai *gap* masing-masing kandidat, setiap kandidat akan diberi bobot nilai sesuai ketentuan pada tabel bobot nilai *gap*.
 6. Menghitung hasil dari tabel bobot pada setiap kandidat.
 7. Mengelompokkan *Core* dan *Secondary Factor*. Setelah menentukan bobot nilai semua aspek dari setiap kandidat, nilai aspek kemudian dibagi menjadi dua ke kelompok *Core Factor* dan *Secondary Factor*. Untuk perhitungan *Core Factor* dapat ditunjukkan pada rumus di bawah ini:

$$NCF = \frac{\sum NC(I,s,p)}{\sum IC} \quad (2)$$

Keterangan:

NCF – Nilai rata-rata *core factor*

NC(I, s, p) – Jumlah total nilai *core factor*

IC – Jumlah *item core factor*

Sedangkan untuk perhitungan *Secondary Factor* dapat ditunjukkan pada rumus di bawah ini:

$$NSF = \frac{\sum NS(I,s,p)}{\sum IS} \quad (3)$$

Keterangan:

NSF – Nilai rata-rata *secondary factor*

$NS(I, s, p)$ – Jumlah total nilai *secondary factor*

IS – Jumlah *item secondary factor*

8. Melakukan Perhitungan Nilai Total dari setiap aspek, lalu dihitung nilai total berdasarkan persentase dari *core factor* dan *secondary factor* yang diperkirakan berpengaruh terhadap kinerja tiap-tiap profil. Rumus perhitungan nilai total dapat ditunjukkan pada rumus di bawah ini:

$$(x)\% \cdot NCF(I, s, p) + (x)\% \cdot NSF(I, s, p) = N(I, s, p) \quad (4)$$

Keterangan:

$NCF(I, s, p)$ – Nilai rata-rata *core factor*

$NSF(I, s, p)$ – Nilai rata-rata *secondary factor*

$N(I, s, p)$ – Nilai total dari aspek

$(x)\%$ – Nilai persen yang diinputkan

9. Hasil akhir dari proses *Profile Matching* adalah ranking dari kandidat yang diajukan. Penentuan ranking mengacu pada hasil perhitungan tertentu. Semakin besar nilai hasil akhir, semakin besar pula kesempatan untuk menduduki posisi yang ada. Rumus perhitungan ranking dapat ditunjukkan pada rumus di bawah ini:

$$Ranking = (x)\% \cdot Ni + (x)\% \cdot Ns + (x)\% \cdot Np \quad (5)$$

Keterangan:

Ni – Nilai kecerdasan

Ns – Nilai sikap kerja

Np – Nilai perilaku

$(x)\%$ – Nilai persen yang diinputkan

IV. RANCANGAN APLIKASI DAN METODOLOGI PENELITIAN

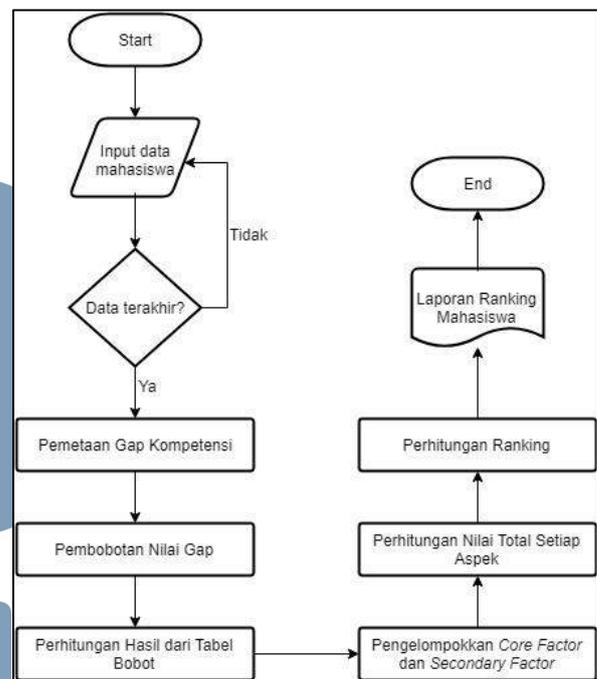
Penjelasan dalam bab ini dimulai dengan rancangan alur aplikasi dan dilanjutkan dengan metodologi penelitian yang diterapkan.

A. Rancangan Alur Aplikasi

Dalam penelitian ini, aplikasi Sistem Pendukung Keputusan dibuat dengan bahasa C# dan digunakan untuk mendukung keputusan beasiswa di UMN dengan mengimplementasikan metode *Profile Matching*. Gambar 1 menampilkan garis besar alur aplikasi yang telah dibuat. Dimulai dengan input data mahasiswa, lalu mengecek data terakhir atau bukan.

Jika sudah selesai menambahkan data, sistem akan melakukan pemetaan *gap* kompetensi. Lalu dilakukan proses pembobotan nilai *gap* berdasarkan pemetaan *gap* kompetensi sebelumnya. Sistem akan menghitung tabel bobot berdasarkan hasil nilai *gap*.

Pengelompokkan *core factor* dan *secondary factor* pada setiap sub-aspek seperti IPK, tanggungan orang tua, dan lainnya. Setelah dikelompokkan, akan dihitung nilai total untuk setiap aspeknya. Berdasarkan hasil yang diperoleh, akan digunakan untuk menghitung ranking. Ranking semakin besar, maka semakin besar mahasiswa tersebut mendapat kesempatan untuk memperoleh beasiswa.



Gambar 1. Diagram Alur Aplikasi

B. Metodologi Penelitian

Tahapan penelitian yang dilakukan meliputi langkah-langkah berikut:

1. Studi Literatur

Studi literatur merupakan tahapan dimana peneliti melakukan pembelajaran dari buku, jurnal, artikel, maupun referensi lain, yang tersedia secara *online* maupun *offline*, yang berhubungan dengan penelitian yang dilakukan.

2. Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan sistem, meliputi fitur-fitur yang perlu disediakan, seperti *software* dan data-data mahasiswa yang sudah dikonsultasikan dengan pihak terkait.

3. Desain Sistem

Pada tahap desain sistem ini dilakukan perancangan *user interface*, desain modul-modul aplikasi, dan desain konten.

4. Pemrograman Sistem

Pada tahap ini, dilakukan pemrograman sistem yang dibutuhkan sesuai rancangan, dan berjalan sesuai rancangan yang telah dibuat.

5. Testing

Pada tahap ini, dilakukan *testing* atau percobaan terhadap aplikasi yang telah dirancang untuk memeriksa apakah ada *error*, kekurangan, atau ada yang tidak sesuai dengan rancangan.

6. Implementasi

Pada tahap ini, dilakukan implementasi sistem yang sudah dibuat peneliti kepada pengguna.

7. Evaluasi

Pada tahap ini, dilakukan evaluasi terhadap pengguna sistem. Berdasarkan hasil evaluasi dilakukan analisa untuk ditarik kesimpulan dari penelitian.

8. Penulisan Laporan dan Dokumentasi

Pada tahap ini, dilakukan penulisan laporan untuk menjabarkan penelitian yang dilakukan dan sebagai dokumentasi hasil penelitian.

V. HASIL DAN PEMBAHASAN

Setelah tahap desain dilakukan, maka pemrograman aplikasi dilaksanakan. Gambar 2 dan 3 memperlihatkan halaman input data untuk beasiswa orang tua meninggal dan beasiswa non-akademik. Proses ubah dan hapus data juga dapat dilakukan pada submenu lainnya.

Gambar 2. Halaman Input Data Beasiswa Orang Tua Meninggal

Gambar 3. Halaman Input Data Beasiswa Non-Akademik

Gambar 4 adalah gambar halaman untuk menampilkan data yang sudah diproses menggunakan metode *Profile Matching*. Terdapat tiga tombol, yaitu tombol Beasiswa Non-Akademik untuk menampilkan semua data mahasiswa beserta ranking (hasil dari metode *Profile Matching*) yang mengikuti beasiswa non-akademik di *listview*, tombol Beasiswa Orang tua Meninggal untuk menampilkan semua data mahasiswa beserta ranking yang mengikuti beasiswa orang tua meninggal di *listview*, dan tombol Print untuk mencetak data yang terdapat di *listview* saat itu. Terdapat juga fitur *search* atau pencarian dengan satu *field* dan dua *radio button*. *Field* tersebut digunakan untuk mencocokkan dan mencari nama atau NIM (berdasarkan pilihan *radio button*) yang terdapat di *listview*.

No.	Nama	NIM	Prodi	Semester	IPK	Ranking	Tanggal
1	MICHELLE LAURENCEA	1312210384	Desain Kom. Vis.	3	1,63	2	11/06/2016
2	MARACUMI DWI	1312210347	Desain Kom. Vis.	3	3,84	2.115	11/06/2016
3	DEVANA CLARETIA OCTAVIA	1314211043	Inf. Komunikasi	3	1,83	2	11/06/2016
4	IRESTI ARIANA PERANG	1314211042	Inf. Komunikasi	3	3,66	1.78	11/06/2016
5	ZARANA LILANDA	1312210314	Desain Kom. Vis.	5	1,82	2.25	11/06/2016
6	BERGANDHANGSARAHANUSYATI	1314211036	Inf. Komunikasi	5	1,67	3	11/06/2016
7	CHENY ROSTANDEY	1314211036	Inf. Komunikasi	5	3,74	2	11/06/2016
8	SHENTYASINDERA	1414211040	Inf. Komunikasi	3	3,66	2	11/06/2016
9	WANGSA WIDYAPRANSA	1314211034	Inf. Komunikasi	5	2,71	2	11/06/2016
10	REFFA RAHMANUSUS	1314211076	Inf. Komunikasi	5	3,47	2.25	11/06/2016
11	ADINA RAHMANUSUS	1312210251	Desain Kom. Vis.	5	2,38	2	11/06/2016
12	LAWANGY GEBELLA	1414211043	Inf. Komunikasi	3	2,72	1,75	11/06/2016
13	ADYTA DINALIA	1414211040	Inf. Komunikasi	3	3,88	2	11/06/2016

Gambar 4. Halaman Tampilkan Data

Ranking adalah hasil akhir dari metode *Profile Matching* yang ditampilkan pada aplikasi bagian Tampilkan Data. Nilai maksimal ranking adalah 3 yang artinya paling diprioritaskan, dan nilai minimum ranking adalah 0, yang paling tidak diprioritaskan. Hasil tersebut diolah saat pengguna mengisi atau mengubah data, lalu disimpan ke dalam *database*. Diterima beasiswa yang dimaksud adalah mahasiswa yang mendapatkan beasiswa. Berikut adalah hasil yang diproses oleh aplikasi.

$$\begin{aligned} \text{NCF_E} &= (\text{E1} + \text{E3}) / 2; \\ \text{NSF_E} &= (\text{E2}) / 1; \\ \text{NCF_A} &= (\text{A1}) / 1; \\ \text{NSF_A} &= (\text{A2}) / 1; \end{aligned}$$

Gambar 5. Potongan Kode Perhitungan *Core* dan *Secondary Factor*

Gambar 5 memperlihatkan sistem menghitung variabel NCF_E yaitu nilai total dari *core factor* aspek ekonomi, NSF_E yaitu nilai total dari *secondary factor* aspek ekonomi, NCF_A yaitu nilai total dari *core factor* aspek akademik, dan NSF_A yaitu nilai total dari *secondary factor* aspek akademik. Variabel yang digunakan adalah variabel hasil dari perhitungan bobot nilai *gap* E1, E2, E3, A1, dan A2 (yakni nilai kriteria aspek).

$$\begin{aligned} \text{total_A} &= (\text{NCF_A} * \text{persen_core_A} / 100) + (\text{NSF_A} * \text{persen_secondary_A} / 100); \\ \text{total_E} &= (\text{NCF_E} * \text{persen_core_E} / 100) + (\text{NSF_E} * \text{persen_secondary_E} / 100); \end{aligned}$$

Gambar 6. Potongan Kode Perhitungan Total Setiap Aspek

Pada Gambar 6, sistem menghitung total variabel total_A (aspek akademik) dan total_E (aspek ekonomi) dari hasil perhitungan *core factor* dan *secondary factor* masing-masing aspek, yaitu NCF_A, NSF_A, NCF_E, dan NSF_E. Variabel persen_core_A, persen_core_E, persen_secondary_A, dan persen_secondary_E ditetapkan sesuai dengan permintaan pengguna, yaitu semuanya 50 persen, yang berarti semua aspek dianggap memiliki kontribusi yang sama besar.

$$\text{rank_beasiswa_orangtua_meninggal} = (\text{total_A} * \text{persen_A} / 100) + (\text{total_E} * \text{persen_E} / 100);$$

Gambar 7. Halaman Tampilkan Data

Pada Gambar 7, sistem menghitung ranking beasiswa menggunakan variabel total_A dan total_E, yang dikalikan dengan persentase variabel persen_A dan persen_E yang sudah ditetapkan yaitu 50 persen. Hasil perhitungan ini akan disimpan ke dalam *database*.

Tabel 1. Hasil Ranking Beasiswa Non-Akademik

Nomor Data	Ranking	Diterima Beasiswa
1	2	Ya
2	2.125	Ya
3	2	Ya
4	1.75	Ya
5	2.25	Ya
6	2	Ya
7	2	Ya
8	2	Ya
9	2	Ya
10	2.25	Ya
11	2	Ya
12	1.75	Ya
13	2	Ya
14	0.375	Tidak
15	0.375	Tidak
16	1.375	Tidak
17	1.625	Tidak

Hasil ranking beasiswa Non-Akademik pada Tabel 1 menunjukkan 4 dari 17 data tidak mendapatkan beasiswa dan nilai ranking di bawah 1.625, artinya mahasiswa yang mendapatkan nilai ranking di bawah 1.625 lebih cenderung tidak diterima beasiswa.

Tabel 2. Hasil Ranking Beasiswa Orang Tua Meninggal

Nomor Data	Ranking	Diterima Beasiswa
1	2	Ya
2	1.75	Ya
3	1.75	Ya
4	2	Ya
5	2.375	Ya
6	2.125	Ya
7	1.625	Tidak
8	2.5	Ya
9	1.75	Ya
10	2.25	Ya
11	2.125	Tidak
12	1.875	Ya
13	2	Ya
14	2.125	Ya
15	2.625	Ya
16	2.25	Ya
17	2.125	Ya
18	2.375	Ya
19	2.125	Ya
20	1.75	Ya
21	1.5	Tidak
22	1.875	Ya
23	2.125	Ya
24	2.125	Ya

Nomor Data	Ranking	Diterima Beasiswa
25	2	Ya
26	1.75	Ya
27	2.25	Ya
28	2.125	Ya
29	1.625	Tidak

Hasil ranking beasiswa orang tua meninggal pada Tabel 2 menunjukkan 4 dari 29 data tidak mendapatkan beasiswa dan tiga diantaranya karena memiliki nilai ranking di bawah 1.625, namun terdapat satu data mahasiswa yang memiliki nilai ranking 2.125 dan tidak diterima beasiswa. Hal ini disebabkan karena hasil pertimbangan oleh pengambil keputusan tidak sesuai dengan prioritas yang diberikan oleh aplikasi. Pada data nomor 11, orang tua mahasiswa memiliki penghasilan yang sangat tinggi, memiliki dua tanggungan orang tua, usia orang tua yang masih hidup kurang dari 55 tahun, mengikuti sembilan kegiatan akademik, dan status ditolak dari psikolog. Hasil kedua bagian beasiswa dipengaruhi oleh beberapa faktor, yaitu penetapan aspek, penetapan nilai target aspek, pembagian *core factor* dan *secondary factor*, penetapan tabel kriteria aspek, persentase untuk nilai total dan nilai ranking yang masih belum optimal.

VI. KESIMPULAN DAN SARAN

Aplikasi Sistem Pendukung Keputusan Beasiswa UMN yang dapat mendukung pengguna untuk memutuskan mahasiswa yang berhak mendapatkan beasiswa dengan menggunakan metode *Profile Matching* telah berhasil dibangun oleh peneliti. Berdasarkan hasil penelitian, tingkat kepuasan pengguna terhadap aplikasi adalah 72%. Hasil ini menunjukkan aplikasi sistem pendukung keputusan beasiswa UMN yang dibangun telah cukup baik dan pengguna merasa puas dengan hasil yang diberikan. Pada aplikasi bagian beasiswa Non-Akademik dan

beasiswa orang tua meninggal, hasil perhitungan manual dengan hasil perhitungan aplikasi tidak jauh berbeda, sehingga dapat disimpulkan bahwa hasil rekomendasi aplikasi sesuai dengan hasil penerimaan beasiswa selama ini dan dengan demikian, aplikasi bisa digunakan untuk membantu dalam mendukung pengambilan keputusan beasiswa UMN.

Ada beberapa saran yang dapat dijabarkan berdasarkan hasil penelitian yang diperoleh, yaitu:

1. Kembangkan dan bandingkan metode *Profile Matching* dengan metode lain seperti *simple additive weighting* pada beasiswa UMN agar mengetahui metode yang lebih efektif.
2. Optimalisasi penetapan aspek, penetapan nilai target aspek, pembagian *core factor* dan *secondary factor*, penetapan tabel kriteria aspek, persentase untuk nilai total dan nilai ranking agar lebih efektif.
3. Kembangkan aplikasi ke dalam bentuk situs atau *website* yang diminta oleh pengguna untuk memudahkan dalam mengakses.

DAFTAR PUSTAKA

- [1] M. T. Irawan dan D. Kriestanto, "Penerapan Profile Matching untuk Pencarian Siswa SMP Penerima Beasiswa Miskin dan Berprestasi", *Jurnal Informatika dan Komputer (JIKO)* - Vol.1, No.1, pp.24-29, 2016..
- [2] L. Puspitasari, "Penerapan Metode Profile Matching dalam Sistem Pendukung Keputusan Penilaian Kinerja Karyawan (Studi Kasus: PT. Perkebunan Nusantara III Medan)", *Pelita Informatika Budi Darma*, Volume:V, Nomor:3, 2013.
- [3] Kusriani, *Konsep dan Aplikasi Sistem Pendukung Keputusan*, Yogyakarta: Andi Offset, 2007.
- [4] J. H. Moore dan M. G. Chang, "Design of Decision Support Systems", *ACM SIGMIS Database - Selected papers on decision support systems from the 13th Hawaii International Conference on System Sciences*. Volume 12 Issue 1-2, pp.8-14, 1980.
- [5] W. Setyaningsih. *Decision Support System Menggunakan Metode Profile Matching*. 2013. URL: <http://staff.budiluhur.ac.id/deni.mahdiana/files/2013/05/Metode-Profile-Matching.pdf>, diakses tanggal 3 Maret 2016.

Perbandingan Regresi Linear dengan Heaviside Activation Function dengan Logistic Regression untuk Klasifikasi Diabetes

Felix Indra Kurniadi¹, Vinnia Kemala Putri²

¹ School of Engineering and Technology: Informatics Engineering, Tanri Abeng University, Jakarta, Indonesia

² Fakultas Ilmu Komputer, Universitas Indonesia, Depok, Indonesia

felixindra@tau.ac.id

vinnia.kemala51@ui.ac.id

Diterima 15 Februari 2018

Disetujui 8 Juni 2018

Abstract —Diabetes is one of the diseases that rapidly increase in the world. One of the most used dataset for diabetes is Pima indian dataset. Pima indian have 8 features such as pregnancies, glucose, blood pressure, insulin, BMI, diabetes pedigree function and age. In this research we are comparing between Linear Regression using Heaviside Activation Function and Logistic Regression. Logistic regression gives better result compare linear regression using Heaviside Activation Function.

Index Terms—Diabetes, Regresi, Heaviside Activation Function, Logistic Regression

I. PENDAHULUAN

Diabetes adalah salah satu penyakit yang disebabkan dikarenakan ketidak mampuan pankreas untuk menghasilkan insulin yang mengontrol gula dalam darah. Hal ini jika dibiarkan akan berbahaya dikarenakan dapat terjadi komplikasi seperti jantung, stroke, glaukoma ataupun kanker[1].

Berdasarkan data yang didapatkan oleh WHO (World Health Organization) bahwa[2]:

- Terdapat 346 juta penduduk yang menderita penyakit diabetes mellitus
- Pada tahun 2004, estimasi 3.4 juta penduduk meninggal dikarenakan konsekuensi dari tingginya gula darah.
- Lebih dari 80% penderita diabetes mellitus yang meninggal merupakan warga dengan pendapatan menengah kebawah.

Di Indonesia, berdasarkan data yang dikeluarkan oleh WHO pada tahun 2016, kematian yang disebabkan oleh diabetes sebanyak 99.400 jiwa, dimana 48.300 kasus kematian terjadi pada usia produktif [3]. Sedangkan berdasarkan data yang diambil oleh Riset Kesehatan Dasar (Riskesdas) pada tahun 2013 terdapat 2.650.340 jiwa yang diketahui menderita penyakit diabetes [4]. Dari angka yang diberikan oleh WHO dan Riskesdas dapat disimpulkan bahwa diabetes merupakan salah satu penyakit yang mematikan.

Penggunaan sistem cerdas dapat membantu tenaga medis untuk melakukan klasifikasi terhadap penyakit diabetes berdasarkan data kesehatan pasien. Proses pengklasifikasian diharapkan dapat memberikan pilihan terhadap tenaga medis sehingga tidak terjadi kesalahan dalam pengobatan yang dapat mengakibatkan kematian. Hal inilah yang menyebabkan perkembangan penelitian untuk meningkatkan kemampuan komputer untuk melakukan klasifikasi dengan data pasien[5].

Tujuan dari penelitian ini adalah untuk melakukan proses klasifikasi diabetes menggunakan metode linear regression dengan *activation function* menggunakan *Heaviside methods* kemudian metode akan dibandingkan dengan metode *Logistic Regression*.

Jurnal ini terdiri atas 5 bagian. Pada Bagian I akan menjelaskan mengenai pendahuluan, pada Bagian II akan menjelaskan mengenai penelitian sebelumnya, Bagian III akan menjelaskan mengenai eksperimen yang akan dilakukan, Bagian IV akan memberikan hasil dan diskusi dari penelitian dan Bagian V akan menjelaskan mengenai kesimpulan yang didapatkan dari penelitian ini.

II. PENELITIAN SEBELUMNYA

Beberapa penelitian yang mencoba menyelesaikan permasalahan mengenai klasifikasi diabetes. Heydari mencoba membandingkan diabetes *type 2* di Iran dengan menggunakan *Support Vector Machine*, *Artificial Neural Network*, *Decision Tree*, *Bayesian Network* dan *Nearest Neighbour*. Data yang digunakan untuk penelitian ini adalah dataset dari *Iranian Ministry Health, Food and Drug Administration* untuk diabetes *type 2*, hasil terbaik didapatkan dengan menggunakan *Artificial Neural Network* dengan tingkat akurasi 0.9744[5].

Jaafar mencoba menangani permasalahan diabetes melitus dengan menggunakan *Back-propagation Neural Network*. Data yang digunakan adalah data Pima Indian dimana fitur diambil adalah sebanyak delapan fitur. Penelitian ini menyatakan bahwa penggunaan neural network sangat bergantung pada data apabila data yang

digunakan lebih banyak dapat memberikan hasil yang lebih baik dibandingkan dengan saat percobaan dilakukan[6].

Nai-arun mencoba melakukan klasifikasi diabetes dengan menggunakan *Decision Tree*, *Artificial Neural Network*, *Random Forest* dan *Naïve Bayes* dan setiap metode akan dilakukan proses *ensemble* dengan menggunakan metode *Bagging* dan *Boosting*. Data didapatkan dari *Sawanpracharak Regional Hospital* antara tahun 2012-2013, kemudian diambil sebanyak 11 fitur yang digunakan. Hasil akurasi terbaik didapatkan metode *Random Forest* dengan akurasi sebesar 85.558%. Hasil penelitian ini kemudian diaplikasikan ke web yang telah dibuat[7].

Penelitian lainnya dilakukan oleh Chandgude, metode yang digunakan adalah *Fuzzy Inference System*. Hasil yang didapatkan bahwa hasil menggunakan FIS didapatkan *precision* sebesar 0.98 dengan 900 *training dataset*. Data yang digunakan adalah *dataset Pima Indian*.

Kumari menggunakan *Back Propagation Neural Network* untuk mencoba mengatasi klasifikasi terhadap orang penderita diabetes mellitus. Data yang digunakan adalah data yang diambil dengan melakukan proses survei. Fitur yang digunakan adalah jenis kelamin, umur, berat, tinggi badan, kekurangan berat badan, keinginan untuk minum, keinginan untuk makan, nafsu makan, pusing, muntah, infeksi dan pandangan berkabut. Hasil yang didapatkan menggunakan *Back Propagation Neural Network* adalah 92.8%[2].

III. EKSPERIMEN

A. Data

Data yang digunakan berasal dari data *Pima Indians Diabetes dataset* yang berasal dari *University of California Irvine Machine Learning Repository* (<https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes>). Data pima indian sendiri berasal dari *National Institute of Diabetes and Digestive and Kidney Disease* dengan mengambil sampel terhadap 768 orang perempuan keturunan pima indian antara umur 21 tahun ke atas[8].

Data pima indian memiliki 9 attribute yang digunakan:

1. *Pregnancies*: berapa kali sampel mengandung
2. *Glucose*: Konsentrasi *plasma glucose*, 2 jam setelah melakukan *oral glucose tolerance test*
3. *Blood Pressure*: tekanan darah diastolik (mm Hg)
4. *Skin Thickness*: ketebalan kulit trisep
5. Insulin: 2 jam serum insulin
6. BMI: *body mass index*

7. *Diabetes Pedigree Function*: Silsilah diabetes
8. Umur
9. Hasil: diabetes atau tidak diabetes (0 atau 1)

Data pima indian sendiri memiliki permasalahan utama dimana terdapat *missing value* pada beberapa fitur data. Pada penelitian ini nilai *missing value* hanya diubah menjadi 0.

B. Pengaturan Eksperimen

Data yang ada dibagi menjadi dua bagian yaitu untuk proses pelatihan dan proses pengujian. Metode yang digunakan adalah pembagian acak terhadap data utama. Pembagian dilakukan dengan aturan 75% untuk data pelatihan dan 25% untuk data pengujian.

C. Proses Eksperimen

Pada tahapan penelitian data yang sudah dibagi akan dilakukan pencarian model untuk mendapatkan model regresi yang diinginkan. Dari data tersebut dipisah fitur yang digunakan dengan target yang ingin dicapai. Pada penelitian ini fitur yang digunakan adalah *pregnancies*, *glucose*, *skin thickness*, *insulin*, *BMI*, *diabetes pedigree function* dan umur. Sedangkan atribut hasil adalah target yang diinginkan.

Pada penelitian mencoba membandingkan metode regresi linear dengan *Heaviside Activation function* dengan metode *Logistic Regression*. Regresi linear merupakan salah satu metode yang digunakan untuk melakukan prediksi nilai sehingga untuk melakukan suatu proses klasifikasi dibutuhkan sebuah *activation function* untuk memberikan klasifikasi terhadap hasil prediksi nilai yang dibuat oleh regresi linear. Fungsi Regresi linear dapat dilihat pada persamaan (1)[9]:

$$\hat{y} = b_0 + b_1X_1 + b_2X_2 + \dots + b_nX_n \quad (1)$$

Dimana \hat{y} adalah hasil prediksi, $b_0, b_1, b_2, \dots, b_n$ adalah koefisien regresi dan X_1, X_2, \dots, X_n adalah Fitur yang digunakan untuk melakukan klasifikasi.

Proses mencari koefisien regresi yang digunakan dalam penelitian ini dapat dilihat dari persamaan (2):

$$b = (X_i^T \cdot X_i)^{-1} X_i^T Y \quad (2)$$

Dimana Y adalah target yang diinginkan X_i adalah fitur yang digunakan. Hasil yang telah didapatkan dari mencari koefisien regresi akan dimasukkan ke dalam persamaan (1) menjadi sebuah persamaan untuk proses klasifikasi menggunakan metode *Heaviside activation function*. *Heaviside activation function* merupakan metode sederhana dengan menggunakan *threshold* untuk membagi antara nilai 0 dan 1 atau -1 dan 1. Persamaan *Heaviside activation function* adalah[10]:

$$f(x) = \begin{cases} 0 & \text{if } x < \theta \\ 1 & \text{if } x \geq \theta \end{cases} \quad (3)$$

Dimana θ adalah *threshold*, dan $f(x)$ adalah nilai hipotesis target yang akan didapatkan.

Selain itu hasil dari proses klasifikasi akan dibandingkan dengan *Logistic Regression*. *Logistic Regression* merupakan salah satu metode untuk melakukan klasifikasi yang baik. *Logistic Regression* itu sendiri adalah sebuah metode untuk pembuatan model yang outputnya berbentuk *crisp* Persamaan untuk *Logistic Regression*[7]:

$$\hat{y} = \frac{e^{(b_0+b_1X_1+b_2X_2+\dots+b_nX_n)}}{1 + e^{(b_0+b_1X_1+b_2X_2+\dots+b_nX_n)}} \quad (4)$$

Dimana e adalah *Euler's equations*

D. Evaluasi

Pengevaluasian hasil terhadap penelitian ini akan menggunakan *accuracy*, *F1-score*, *precision* dan *recall*. *Precision* digunakan untuk mengukur probabilitas terhadap *classifier exactness*, sedangkan *recall* digunakan untuk mengukur probabilitas terhadap *classifier completeness*. Berbeda dengan *precision* dan *recall*, *F1-score* mencoba membandingkan keseimbangan antara *precision* dan *recall*[11][12].

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} * 100\% \quad (5)$$

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

$$recall = \frac{tp}{tp + fn} * 100\% \quad (7)$$

$$F1 - Score = 2 * \left(\frac{recall * precision}{recall + precision} \right) * 100\% \quad (8)$$

dimana tp dan tn merupakan jumlah prediksi yang benar.

IV. HASIL EKSPERIMEN

Pada penelitian ini akan dilakukan pencarian nilai menggunakan regresi linear yang dibantu dengan metode *Heaviside activation function*. Pada penelitian ini *threshold* (θ) yang digunakan adalah 1 sehingga apabila θ lebih kecil dari 1 akan diberi nilai 0 dan sebaliknya maka akan diberi nilai 1. Selain itu penelitian ini akan membandingkan dengan menggunakan *logistic regression*. Hasil untuk *accuracy*, *precision*, *recall* dan *F-Score* dapat dilihat pada Tabel 1 dan Tabel 2. Sedangkan pada Tabel 3 dan Tabel 4 akan diberikan data *confusion matrix*.

Tabel 1. Hasil Akurasi, *Precision*, *Recall* dan *F1-score* dari Metode *Linear Regression* menggunakan *Heaviside activation Function*

	Result(%)
Accuracy	74.03
Precision	72.5
Recall	50
F1-Score	59.18

Tabel 2. Hasil Akurasi, *Precision*, *Recall* dan *F1-score* dari Metode *Logistic Regression*

	Result(%)
Accuracy	75.97
Precision	76.92
Recall	51.72
F1-Score	61.86

Tabel 3. *Confusion Matrix* dari Metode *Linear Regression* menggunakan *Heaviside activation Function*

	Diabetes	Tidak Diabetes
Diabetes	85	11
Tidak Diabetes	29	29

Tabel 4. *Confusion Matrix* dari Metode *Logistic Regression*

	Diabetes	Tidak Diabetes
Diabetes	87	9
Tidak Diabetes	28	30

Berdasarkan tabel 1, dan tabel 2 dapat dikatakan bahwa *logistic regression* memberikan hasil yang lebih baik dibandingkan dengan metode *linear regression* dengan menggunakan *Heaviside activation function*. Pada tabel *confusion matrix* dari kedua metode dapat dilihat adanya ketimpangan data pada data diabetes. Hal ini dapat membuat hasil klasifikasi lebih condong untuk memilih seseorang diabetes.

V. KESIMPULAN DAN PENELITIAN SELANJUTNYA

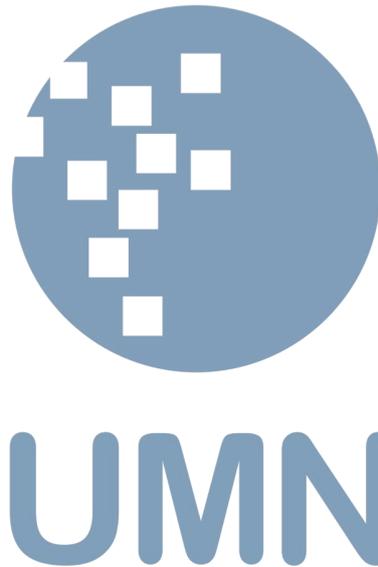
Pada penelitian ini dapat dilihat bahwa berdasarkan nilai akurasi, *precision*, *recall* dan *F-Score*, metode *Logistic Regression* memberikan hasil yang lebih baik dibandingkan dengan metode *linear regression* dengan *Heaviside activation function*. Tetapi perbedaan akurasi kedua metode tersebut tidak terlalu signifikan sehingga kedua metode ini layak untuk digunakan pada penelitian lainnya.

Beberapa kendala yang belum ditangani pada data pima indian adalah ketidakseimbangan antara kelas diabetes dengan tidak diabetes. Kendala pada data ini perlu dikembangkan seperti penambahan data seperti pengambil data lebih banyak untuk menyetarakan ketimpangan pada data. Selain itu kendala pada *missing value* dapat dilakukan proses pencarian nilai *missing value* pada penelitian berikutnya.

DAFTAR PUSTAKA

- [1] WebMD, "WebMD Diabetes Center: Types, Causes, Symptoms, Tests, and Treatments." [Online]. Available: <https://www.webmd.com/diabetes/default.htm>. [Accessed: 02-Feb-2018].
- [2] S. Kumari and A. Singh, "A Data Mining Approach for the Diagnosis of Diabetes Mellitus," in *International Conference on Intelligent Systems and Control*, 2013, pp. 373–375.

- [3] World Health Organization, "Diabetes country profile 2016," 2016.
- [4] Pusat Data dan Informasi Kementerian Kesehatan Republik Indonesia, "Situasi dan Analisis Diabetes," Jakarta, 2014.
- [5] M. Heydari, M. Teimouri, and Z. Heshmati, "Comparison of various classification algorithms in the diagnosis of type 2 diabetes in Iran," *Int. J. Diabetes Dev. Ctries.*, vol. 36, no. 2, pp. 167–173, 2015.
- [6] S. F. B. Jaafar and D. M. Ali, "Diabetes Mellitus Forecast Using Artificial Neural Network (ANN)," in *2005 Asian Conference on Sensor and The International Conference on New Techniques in Pharmaceutical and Biomedical Research Proceedings*, 2005, pp. 135–139.
- [7] N. Nai-arun and R. Moungrai, "Comparison of Classifiers for the Risk of Diabetes Prediction," in *7th International Conference on Advances in Information Technology*, 2015, pp. 132–142.
- [8] R. S. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," *Johns Hopkins APL Tech. Dig.*, vol. 10, pp. 262–266, 1988.
- [9] I. Naseem, R. Togneri, S. Member, and M. Bennamoun, "Linear Regression for Face Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106–2112, 2010.
- [10] J. Wang, "Analysis and Design of a k -Winners-Take-All Model With a Single State Variable and the Heaviside Step Activation Function," *IEEE Trans. Neural Networks*, vol. 21, no. 9, pp. 1496–1506, 2010.
- [11] C. Goutte and E. Gaussier, "A Probabilistic Interpretation of Precision , Recall and F -Score , with Implication for Evaluation," in *European Conference on Infromation Retrieval*, 2005, pp. 345–359.
- [12] J. Brownlee, "Classification Accuracy is not Enough Measures You Can See," 2014. [Online]. Available: <https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/>. [Accessed: 06-Apr-2018].



Klasifikasi Diabetes Menggunakan Model Pembelajaran Ensemble Blending

Vinnia Kemala Putri¹, Felix Indra Kurniadi²

¹ Fakultas Ilmu Komputer, Universitas Indonesia, Depok, Indonesia

² School of Engineering and Technology: Informatics Engineering, Tanri Abeng University, Jakarta, Indonesia
vinnia.kemala51@ui.ac.id
felixindra@tau.ac.id

Diterima 15 Februari 2018

Disetujui 8 Juni 2018

Abstract—Diabetes mellitus is one of the deadliest disease and it is increasing in occurrence through the world. This can be prevented by conducting early diagnosis and treatment. However, in developing countries, less than half of people with diabetes are diagnosed correctly which lead to lose of human lives. In this Big Data era, medical databases have enormous quantities of data about their patients. But this medical data may contain noise and a lot of useless information which may mislead the expert in making a decision for medical diagnosis. Data mining is a technique to that is very effective for medical applications for identifying patterns and extracting useful information for databases. This paper proposed a data mining approach using an ensemble blending method to tackle a diabetes prediction problem in Pima Indian Diabetes Dataset. We proposed a blending ensemble classifier approach using a combination of Decision Tree and Logistic Regression as base classifiers, and Support Vector Machine as a top blender classifier. Our approach reached accuracy of 81% and F1-score of 0.81 proves to be higher when compared with basic classifier without combination.

Index Terms—diabetes, ensemble, data mining

I. PENDAHULUAN

Diabetes merupakan suatu penyakit yang terjadi akibat kurangnya insulin atau ketidakmampuan pancreas dalam tubuh untuk memproduksi hormon insulin. Insulin berguna untuk mengubah glukosa dalam tubuh menjadi glikogen dan trigliserida. Kedua zat ini merupakan sumber energi yang disimpan dalam tubuh. Kekurangan insulin dapat mengakibatkan menumpuknya kadar glukosa dalam darah. Penumpukkan tersebut dapat menyebabkan komplikasi pada ginjal dan beberapa masalah lainnya [1].

Diagnosis medis merupakan hal yang penting dan krusial. Kesalahan diagnosis yang dilakukan oleh dokter dapat mengakibatkan malpraktek dan berujung pada kematian pasien. Hasil analisis data yang diambil dari pasien dan kemampuan dokter untuk mengambil keputusan sangatlah mempengaruhi hasil diagnosis medis. Tahap-tahap yang dilakukan sangatlah panjang

dan membutuhkan waktu dan tenaga yang sangat banyak. Oleh karena itu tindak medis tersebut harus dilakukan secara tepat dan efisien.

Sayangnya, di beberapa negara berkembang ketersediaan dokter ahli dalam diabetes sangatlah terbatas dan dapat mengakibatkan kesalahan diagnosis. Hal ini dapat membuang-buang waktu, tenaga, dan bahkan nyawa pasien. Maka dibutuhkanlah suatu metode yang dapat membantu para ahli kesehatan agar dapat memberikan diagnosis yang lebih tepat dan meningkatkan kesejahteraan masyarakat.

Untuk meningkatkan keakuratan dan ketepatan dalam memberikan diagnosis, penting untuk menganalisis pola data medis pasien. Dengan munculnya era *Big Data* dan adanya algoritma yang cepat dan efisien untuk menganalisis data, memahami suatu pola data medis adalah suatu hal yang mungkin untuk dilakukan. Namun, dengan ketersediaan data yang sangat besar seringkali data yang didapat dari basis data rumah sakit memiliki banyak derau dan informasi yang tidak dibutuhkan. Hal ini dapat menyesatkan para ahli dalam menganalisis data dan memberikan keputusan.

Penambangan data merupakan suatu teknik pengolahan dan penarikan informasi penting dari sejumlah data yang besar. Teknik ini sering digunakan untuk membantu penelitian terkait masalah kesehatan, yang mana digunakan untuk memprediksi, mendiagnosis dan memberikan solusi perawatan terhadap suatu penyakit [2]. Penambangan data juga telah banyak digunakan untuk mengatasi masalah klasifikasi pada diabetes seperti yang dilakukan oleh M. Panwar et al. menggunakan pendekatan penambangan data dan *K-Nearest Neighbor* (KNN) untuk melakukan klasifikasi *binary* pada *Pima Indian diabetes* [3]. T. Jayalakshmi and A. Santhakumaran melakukan praproses pada *missing value* dan klasifikasi diabetes menggunakan Neural Network (NN) [4]. L. Chang-Shing dan W. Mei-Hui mengusulkan *semantic decision support* untuk membantu pengambilan keputusan diagnosis diabetes dengan menggunakan *Fuzzy Diabetes Ontology* (FDO)

[5]. R. Vaishali, et al. mengusulkan algoritma genetika (GA) untuk fitur seleksi dan *Multi Objective Evolutionary* (MOE) *Fuzzy* untuk klasifikasi diabetes [6]. M. Aakanksha, et al. menggunakan Principal Component Analysis (PCA) untuk mereduksi dimensi dan GA untuk meningkatkan kemampuan NN [7]. E.K. Hashi mengusulkan penggunaan *Decision Tree* C4.5 dan KNN sebagai *supervised* klasifier [8].

Salah satu metode pembelajaran mesin yang kuat adalah model *ensemble*. *Ensemble* mengkombinasikan beberapa model pembelajaran mesin yang sederhana seperti *Naïve Bayes*, *Decision Tree*, *Logistic Regression* dan lainnya, menjadi satu kesatuan model yang memiliki performa setingkat model pembelajaran mesin yang kuat. Beberapa teknik *ensemble* yang populer adalah *bagging* [9] dan *boosting* [10].

Ensemble sering digunakan dalam penambahan data dan pembelajaran mesin dikarenakan model tersebut sederhana, mudah diaplikasikan dan tahan terhadap bias dan variansi. D.A. Davis et al. [11] menggunakan *collaborative filtering* untuk melakukan prediksi jenis penyakit. *Ensemble* sering sekali digunakan dalam pemodelan sistem rekomendasi. M. Jahrer et al. [12] menggabungkan beberapa algoritma *collaborative filtering* seperti SVD, *Neighborhood based approaches*, *Restricted Boltzmann Machine* (RBM), *Asymmetric Factor Model* dan *Global Effect* untuk membuat sistem rekomendasi pada dataset Netflix. R. M. Bell et al. [13] menggunakan teknik *ensemble blending* untuk menggabungkan KNN, *factorization*, RBM, dan *Asymmetric Factor Model*. J. Sill et al. [14] mengusulkan teknik *stacking* dengan pembobotan. *Stacking* adalah salah satu teknik *ensemble* yang mana hasil prediksi pada kumpulan model tingkat pertama, diberikan ke model tingkat kedua sebagai input. Namun, penelitian diabetes menggunakan model *ensemble* masih sangatlah jarang.

Penelitian ini mengusulkan metode *ensemble blending* dengan mengkombinasikan *Decision Tree* (DT), *Logistic Regression* (LogReg), dan *Support Vector Machine* (SVM) untuk melakukan klasifikasi binari pada data diabetes mellitus. Penelitian ini hanya fokus pada pemodelan menggunakan metode *ensemble blending* untuk meningkatkan akurasi pada proses pengklasifikasian, tidak pada prapengolahan data seperti penghilangan derau atau data tidak berhubungan.

II. EKSPLORASI DATASET

Dataset yang dipakai pada penelitian ini adalah dataset yang berasal dari *National Institute of Diabetes and Digestive and Kidney Diseases* dan dapat diakses secara publik di UCI Machine Learning Repository: Pima Indians Diabetes Dataset [15]. Dataset ini memiliki informasi tentang 768 pasien wanita dengan 8 diagnosis kondisi medis yang berbeda-beda. Tujuan

dari penelitian ini adalah untuk memprediksi apakah seorang pasien terkena diabetes atau tidak.

A. Ringkasan Dataset

Dataset terdiri dari 9 kolom sebagai atribut yaitu:

- Pregnancies: Jumlah berapa kali pasien pernah mengandung.
- Glucose: Konsentrasi plasma glukosa selama 2 jam pada tes oral konsentrasi glukosa.
- BloodPressure: Tekanan darah diastole (mm Hg).
- SkinThickness: Tebal lapisan kulit pada bagian triceps (mm).
- Insulin: Kadar 2 jam serum insulin (mu U/ml).
- BMI: Indeks massa tubuh (kg/m²)
- DiabetesPedigreeFunction: Hubungan genetik pasien dengan keluarga yang menderita diabetes.
- Age: Umur dari pasien (tahun).
- Outcome: Terdiri dari 2 kelas, 0 (negatif diabetes) dan 1 (positif diabetes). Atribut yang akan diprediksi.

III. METODOLOGI

Bagian ini memaparkan tentang langkah-langkah penelitian yang dilakukan. Metode yang dilakukan adalah prapengolahan data, pelatihan model dan evaluasi model.

A. Prapengolahan Data

Dikarenakan banyaknya derau pada dataset, prapengolahan data perlu dilakukan untuk meningkatkan akurasi dari penambahan data sebelum mengimplementasikan algoritma pembelajaran mesin pada dataset yang digunakan. Beberapa langkah yang dilakukan yaitu:

- Mengolah *missing values*. Pada dataset ini, selain pada atribut *Outcome*, banyak terdapat angka 0 yang menandakan data tidak tertangkap dengan baik. Angka-angka 0 tersebut terdapat pada atribut Glucose, BloodPressure, SkinThickness, Insulin dan BMI. Angka-angka 0 tersebut diganti menjadi NA dan kemudian diisi dengan menggunakan KNN.
- Melakukan *oversampling* untuk mengatasi ketidakseimbangan jumlah data pada tiap kelas.
- Melakukan normalisasi *center* dan *scale* pada data untuk memperkecil bias dan variansi pada

hasil eksperimen dan mempercepat waktu komputasi.

B. Metode Ensemble - Blending

Dalam mengatasi masalah klasifikasi, pendekatan yang paling sering digunakan adalah melatih beberapa model klasifier dan memilih satu yang memberikan hasil terbaik dari *cross validation*. Namun, tiap-tiap memiliki variansi dan bias yang berbeda-beda. Tiap-tiap akan konvergen dan gagal di tempat yang berbeda.

Salah satu metode pembelajaran mesin yang kuat adalah model *ensemble*. *Ensemble* mengombinasikan beberapa model klasifier dengan tujuan dapat menggabungkan kelebihan dari tiap-tiap model.

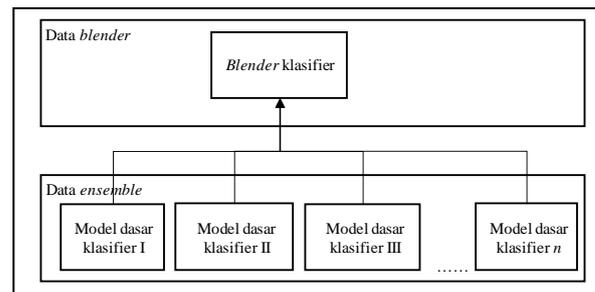
Syarat utama dalam pengaplikasian *ensemble* adalah model-model dasar klasifier harus saling independen dengan tujuan *error rate* dari tiap-tiap model tidak saling berkorelasi. Selain itu, performa tiap-tiap model dasar harus lebih baik dari sebuah klasifier dengan *random guessing* ($error < 0,5$). Model-model dasar dipilih bukan dari tingginya akurasi tetapi dari kesederhanaannya.

Blending adalah salah satu dari teknik *ensemble*. *Blending* pertama kali diperkenalkan oleh pemenang kompetisi Netflix 2007 [13]. Secara umum, cara kerja *blending* mirip dengan *stacked generalization* [16]. Ide dasar dari *blending* adalah menggunakan beberapa klasifier sebagai model dasar kemudian menggunakan sebuah *blender* klasifier untuk mengombinasikan prediksi-prediksi dari model-model dasar.

Beberapa langkah dari *blending* yaitu:

- Membagi data pelatihan menjadi 2 bagian: data *ensemble* dan data *blender*.
- Melakukan pelatihan model-model dasar dengan menggunakan data *ensemble*.
- Melakukan prediksi untuk data *blender* dan data pengujian dengan menggunakan model-model dasar yang telah dilatih sebelumnya.
- Hasil prediksi oleh model-model dasar untuk data *blender*, dijadikan atribut baru untuk data *blender*. Demikian juga hasil prediksi oleh model-model dasar untuk data pengujian, dijadikan atribut baru untuk data pengujian.
- Melatih model akhir *blender* klasifier dengan menggunakan data *blender* yang telah ditambah atributnya.
- Melakukan prediksi dengan menggunakan model akhir *blender* klasifier untuk data pengujian yang telah ditambah atributnya.

Gambar 1 mengilustrasikan ide dasar metode *blending*.



Gambar 1. Ide dasar metode *ensemble blending*

C. Evaluasi Model

Hasil penelitian dievaluasi dengan menggunakan akurasi dan F1-score.

Metode lain untuk mengevaluasi hasil klasifikasi dari model yang berbeda-beda dapat menggunakan *confusion matrix*. Dalam klasifikasi biner, *confusion matrix* memberikan beberapa detail sebagai berikut:

- *True Positive* (TP): Data pada kelas positif diklasifikasikan pada kelas positif.
- *True Negative* (TN): Data pada kelas negatif diklasifikasikan pada kelas negatif.
- *False Positive* (FP): Data pada kelas negatif diklasifikasikan pada kelas positif.
- *False Negative* (FN): Data pada kelas positif diklasifikasikan pada kelas negatif.

Akurasi mengindikasikan rasio antara data teridentifikasi dengan tepat dan keseluruhan data pengujian. Persamaan untuk mengukur akurasi adalah sebagai berikut:

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

Sensitifitas mengindikasikan rasio antara TP dengan keseluruhan data aktual kelas positif. Persamaan untuk mengukur sensitifitas adalah sebagai berikut:

$$sensitifitas = \frac{TP}{FN + TP} \quad (2)$$

Presisi mengindikasikan rasio antara TP dengan keseluruhan data prediksi kelas positif. Persamaan untuk mengukur sensitifitas adalah sebagai berikut:

$$presisi = \frac{TP}{FP + TP} \quad (3)$$

F1-score mengindikasikan ratio rata-rata antara sensitifitas dan presisi. Persamaan untuk mengukur F1-score adalah sebagai berikut:

$$F1 - score = \frac{2 \times presisi \times sensitifitas}{presisi + sensitifitas} \quad (4)$$

IV. EKSPERIMEN DAN HASIL

Bagian ini menjelaskan tentang eksperimen yang dilakukan untuk memprediksi kondisi pasien terkena diabetes atau tidak. Eksperimen dilakukan dengan menggunakan bantuan *tool* R studio.

Dataset awal berjumlah 768 pasien dengan 500 pasien tidak terkena diabetes (kelas 0) dan 268 pasien terkena diabetes (kelas 1). Dikarenakan terdapat ketidakseimbangan jumlah data pada tiap kelas, maka dilakukan *oversampling* data pada kelas dengan jumlah data lebih kecil, yaitu kelas 1. Setelah dilakukan *oversampling*, jumlah data pada kelas 1 menjadi 500. Dataset akhir yang digunakan berjumlah 1000 pasien.

Proporsi data pelatihan dan pengujian adalah 70% (700 pasien) dan 30% (300 pasien) secara berurutan. Metode validasi yang digunakan adalah *k-fold cross validation* dengan nilai $k = 10$.

A. Eksperimen I – Pemilihan Model

Pada eksperimen I melakukan beberapa klasifikasi dengan menggunakan algoritma pembelajaran mesin yang berbeda-beda guna menentukan model dasar untuk model *ensemble* nantinya. Algoritma yang digunakan adalah *decision tree*, *k-nearest neighbor*, *logistic regression*, dan SVM. Eksperimen menggunakan keseluruhan data pelatihan dan data pengujian. Tabel 1 memaparkan hasil evaluasi dari klasifikasi data diabetes Pima Indian pada data pengujian.

Tabel 1. Perbandingan hasil evaluasi

Algoritma	Akurasi	F1-Score
<i>Decision tree</i>	74,33%	0,74
<i>k-nearest neighbor</i>	73,33%	0,73
<i>Logistic regression</i>	75,33%	0,76
SVM (kernel linear)	73,67%	0,73
SVM (kernel radial)	74,00%	0,74

Pada tabel I dapat terlihat bahwa model dengan nilai akurasi dan F1-score terbaik adalah *logistic regression* dengan akurasi sebesar 75,33% dan F1 score 0,76.

B. Eksperimen II – Model Ensemble

Pada tahap eksperimen ini, pemilihan model dasar pembelajaran mesin didasari pada hasil eskperimen I. Model yang dipilih adalah *decision tree*, *logistic regression* dan SVM dengan kernel radial. Pembagian

data *ensemble* dan data *blender* masing-masing adalah 50% dari data pelatihan.

Tahap ini melatih tiga kombinasi model yang berbeda, yaitu:

- Model I; dengan model dasar adalah *decision tree* dan *logistic regression*, dan *blender* adalah SVM dengan kernel radial.
- Model II; dengan model dasar adalah *decision tree* dan SVM dengan kernel radial, dan *blender* adalah *logistic regression*.
- Model III; dengan model dasar adalah *logistic regression* dan SVM dengan kernel radial, dan *blender* adalah *decision tree*.

Tabel 2. Perbandingan hasil evaluasi

Model	Akurasi	F1-Score
I. Model dasar: DT+LogReg Blender: SVM (kernel radial)	81,00%	0,81
II. Model dasar: DT+ SVM (kernel radial) Blender: LogReg	80,33%	0,81
III. Model dasar: LogReg+ SVM (kernel radial) Blender: DT	79,67%	0,79

Pada tabel 2 dapat dilihat bahwa model *ensemble* memberikan peningkatan yang signifikan daripada model tanpa dilakukan *ensemble* seperti yang tertera pada tabel 1. Tabel 3-5 memaparkan *confusion matrix* untuk masing-masing model.

Tabel 3. *Confusion Matrix* Model I Blender SVM

n=300		Aktual	
		0	1
Prediksi	0	TN=124	FN=18
	1	FP=39	TP=119

Tabel 4. *Confusion Matrix* Model II Blender LogReg

n=300		Aktual	
		0	1
Prediksi	0	TN=107	FN=26
	1	FP=33	TP=134

Tabel 5. *Confusion Matrix* Model III Blender DT

n=300		Aktual	
		0	1
Prediksi	0	TN=121	FN=31
	1	FP=30	TP=118

V. SIMPULAN

Penelitian ini mengusulkan penggunaan model *ensemble blending* untuk mengatasi masalah klasifikasi diagnosis diabetes. Dari hasil evaluasi dapat terlihat bahwa metode yang diusulkan memberikan nilai akurasi dan *F1-score* yang lebih tinggi daripada model pembelajaran mesin biasa.

Banyak cara yang dapat dilakukan untuk meningkatkan performa dari model yang diusulkan ini. Salah satunya dengan melakukan seleksi fitur dan reduksi dimensi pada praproses dataset. Diharapkan dengan melakukan praproses data yang lebih baik, dapat mengurangi derau pada data dan otomatis meningkatkan performa dari model.

Dengan adanya penelitian ini diharapkan dapat membantu para ahli medis dalam mengambil keputusan ketika sedang melakukan diagnosis medis kepada pasien. Sehingga, nilai kematian pasien akibat kesalahan diagnosis penyakit diabetes dapat dikurangi, memberikan informasi mengenai kecenderungan.

DAFTAR PUSTAKA

- [1] N. Yilmaz, O. Inan, dan MS. Uzer, "New data preparation method based on clustering algorithm for diagnosis systems of heart and diabetes diseases," *J. Med Syst* vol. 38, 2014 hal 48-59.
- [2] Richards, Graeme, et al, "Data mining for indicators of early mortality in a diabetes of clinical records." *Artificial intelligence in medicine* 22.3, 2001, hal 215-231.
- [3] M. Panwar, et al, "K-Nearest Neighbor Based Methodology for Accurate Diagnosis of Diabetes Melitus." *Sixth International Symposium on Embedded Computing and System Design (ISED)*, 2016.
- [4] T. Jayalakshmi dan A. Santhakumaran, "A novel classification method for diagnosis of diabetes melitus using artificial neural networks." *Data storage and Data Engineering (DSDE)*, 2010 International Conference on. IEEE, 2010.
- [5] L. Chang-Shing dan W. Mei-hui. "A fuzzy expert system for diabetes decision support application." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41.1, 2011, hal 139-153.
- [6] R. Vaishali, et al, "Genetic algorithm based feature selection and MOE fuzzy classification algorithm on Pima Diabetes dataset." *IEEE International Conference on Computing Networking and Informatics (ICCN)*, 2017.
- [7] M. Aakanksha, et al, "Diagnosis of Diabetes Mellitus using PCA and Genetically Optimized Neural Network." *IEEE International Conference on Computing, Communication and Automation (ICCCA)*, 2017.
- [8] E.K. Hashi, et al, "An expert clinical decision support system to predict disease using classification techniques." *IEEE International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2017.
- [9] L. Breiman, "Bagging Predictor." *Technical Report 421*, Departement of Statistics, University of California, 1994.
- [10] Y. Freund dan R. Schapire, "Experiments with a new Boosting Algorithm." In *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996, hal 148-156.
- [11] D. A. Davis, N. V. Chawla, N. A. Christakis, dan A.-L. Barabasi. "Time to CARE: a collaborative engine for practical disease prediction." *Springer*, 2009.
- [12] M. Jahrer, et al, "Combining Predictions for Accurate Recommender Systems." 2010.
- [13] R. M. Bell, Y. Koren, dan C. Volinsky, "The BellKor Solution to the Netflix Prize." 2007.
- [14] J. Sill, G. Takacs, L. Mackey, dan D. Lim, "Feature-Weighted Linear Stacking." *Arxiv:0911.0460v2*, 2009
- [15] J.W. Smith, et al. "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus." *IEEE Computer Society Press. In Proceedings of the Symposium on Computer Applications and Medical Care*, 1988, hal 261-265.
- [16] D.H. Wolpert, "Stacked Generalization." *Neural Network* 5(2), 1992, hal 241-259.

Analisis Kesiapan Kebutuhan Infrastruktur Replikasi Basis Data pada Sekolah Musik Indonesia Solo

Willy Sudiarto Raharjo¹, Gani Indriyanta², Amsal Maestro³

Prodi Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana Yogyakarta

willysr@ti.ukdw.ac.id

ganind@staff.ukdw.ac.id

amsalmaestro@ti.ukdw.ac.id

Diterima 20 Februari 2018

Disetujui 8 Juni 2018

Abstract— Sekolah Musik Indonesia (SMI) Solo is the center of SMI which has internal data using web-based application at appssmi.com site with SQL Server database and has not been backed up regularly. Database replication is a technique for copying and distributing data and database objects from one database to another and implementing synchronization so data consistency can be guaranteed. Replication can be implemented to the cloud by requiring Internet access. The main concern in SMI Solo was the quality access of the Internet connection and also infrastructure used in SMI Solo. The purpose of this research is to analyze the readiness of data replication infrastructure needs at SMI Solo. The results of the analysis are then used as the basis for making recommendations and design of information technology architecture in the implementation of SMI database replication. We concluded that the infrastructure owned by SMI Solo is sufficient to be used for database replication. This is demonstrated by the very satisfactory performance of the SMI Solo server network with 3.85 Mbps download throughput, 3.49 Mbps upload throughput, 0% packet loss, 25.88 ms delay, and 0.09 ms jitter. On database replication performance thorough test scenarios, average performance on snapshot replication is using for CPU 4.78%, DTU 5.94%, I/O 0.06% and log data I/O 5.25%. The average performance on transactional replication is CPU 0.09%, DTU 0.09%, data I/O 0%, and log I/O 0.04%. Some of the challenges in developing database replication infrastructure to be implemented in all SMI's can run efficiently if each SMI has a local server and Internet network albeit with unstable throughput.

Index Terms—Network Performance, Replication, Snapshot, Transactional

I. PENDAHULUAN

Data merupakan aset yang paling berharga dalam dunia digital seperti saat ini. Ketersediaan data menjadi salah satu faktor penting dalam proses bisnis sebuah institusi atau perusahaan karena tanpa adanya data, maka tidak akan ada informasi yang bisa didapatkan untuk pengambilan keputusan atau analisa performa kinerja sebuah perusahaan. Dalam mendukung ketersediaan data di *server*, terdapat

sebuah teknik yang disebut dengan replikasi basis data yang melakukan penyalinan dan pendistribusian data seperti objek-objek basis data dari satu basis data ke basis data lain atau dari media penyimpanan satu ke media penyimpanan yang lain dan melaksanakan sinkronisasi antara basis data sehingga ketersediaan data dapat terjamin.

Dalam melakukan replikasi basis data, tidak mungkin tanpa biaya. Replikasi basis data memerlukan penyimpanan tambahan dan memperbaharui data memerlukan waktu tersendiri [1]. Penyimpanan tambahan yang dimaksud di sisi *server*. Oleh karena itu, *server* memerlukan kesiapan kebutuhannya seperti jaringan Internet yang digunakan untuk mentransfer data. Performa jaringan Internet yang memiliki *bandwidth up to* dapat mempengaruhi keberlangsungan proses replikasi karena *throughput (bandwidth aktual)* selalu berubah-ubah setiap saat.

Sekolah Musik Indonesia (SMI) adalah bidang usaha PT. Sarana Menjangkau Indonesia yang bergerak di dunia pendidikan musik. SMI mempunyai banyak cabang dan karyawan di Indonesia. Sejauh ini, untuk mendapatkan data/informasi mengenai data penting (informasi terkait keuangan, pembelajaran, unit dan investor) yang ada di suatu cabang menggunakan aplikasi berbasis web di situs appssmi.com. Situs appssmi.com menggunakan *database SQL Server* yang di-hosting menggunakan Windows Azure dan belum ada *backup* secara berkala di server milik SMI sendiri. SMI Solo sebagai pusat SMI di Indonesia memiliki Windows Server 2008 R2 Standard yang memiliki 1 IP publik statis dan *bandwidth up to* 4 Mbps.

II. TINJAUAN PUSTAKA

Penelitian tentang replikasi basis data bukanlah hal yang baru, namun pada setiap kasus yang dijumpai, permasalahan yang dicoba untuk diselesaikan memiliki kekhasan sendiri-sendiri.

Silitonga melakukan implementasi replikasi basis

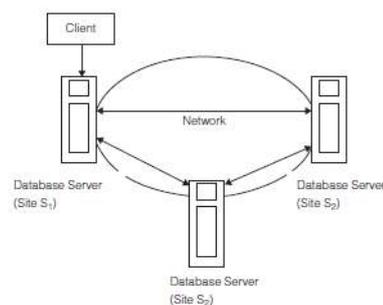
data sistem informasi akademik pada Universitas Katolik Santo Thomas dengan memanfaatkan replikasi multi-master [2], sedangkan Zani dkk mencoba melakukan implementasi metode *asynchronous* pada Replikasi *Database* Inventaris Universitas Bina Darma [3]. Metode *asynchronous* digunakan karena proses replikasi terjadi setelah transaksi di *master* ditahan dulu oleh *buffer* sampai selesai dan dapat diterapkan pada *database open source* seperti MySQL. Mattias juga mencoba melakukan replikasi database multi-master pada layanan E-Learning di Tanzania namun hasil yang di dapatkan ternyata kurang bisa diimplementasikan secara praktis[4].

Truica, Boicea, dan Radulescu telah mencoba melakukan penelitian untuk membandingkan performa dari proses replikasi *asynchronous* basis data Microsoft SQL Server dan PostgreSQL. SQL Server memiliki kelebihan untuk mereplikasikan hanya beberapa bagian pada *database*, kemampuan ini tidak dimiliki oleh PostgreSQL dan MySQL. Untuk performa pengujian replikasi, basis data yang memperoleh waktu yang baik adalah PostgreSQL [5]. Penelitian lain dilakukan oleh Shahapure & Jayarekha untuk membandingkan *performance metric, type*, dan *scalability* pada 11 algoritma replikasi yang dapat diterapkan pada arsitektur *cloud computing*. Hasil penelitiannya adalah setiap algoritma mempengaruhi performa *cloud* ketika melakukan replikasi. Saran dari penelitian ini adalah layanan dengan ketersediaan tinggi memerlukan setidaknya dua server yang identik. Satu *host* utama yang menjalankan layanan, *host* lainnya menjadi tempat cadangan. Dengan menggunakan replikasi, *service* suatu aplikasi dapat ditingkatkan keamanan dan ketersediaan datanya [6].

III. LANDASAN TEORI

A. Replikasi Basis Data

Replikasi merupakan proses menyalin data dari berbagai komputer. Ini merupakan kunci keefektifan dari sistem terdistribusi yang dapat menambah *performance, high availability, dan fault tolerance* [7]. Sebagai contoh, sumber daya dari suatu *web server* di *browser* dan *server web proxy* adalah bentuk replikasi, karena data yang ada di *cache browser* dan pada *server* ada replika dari satu sama lain. Gambaran replikasi ditunjukkan pada Gambar 1.



Gambar 1. Gambaran proses replikasi basis data

Replikasi mempunyai beberapa istilah sebagai berikut:

- *Publisher* : Sebuah *Database Management System (DBMS)* yang membuat data tersedia ke lokasi lain lewat replikasi. *Publisher* bisa satu atau lebih, didefinisikan sebagai set dari objek dan data yang direplikasikan
- *Distributor* : Sebuah DBMS yang menyimpan data replikasi dan metadata tentang publikasi, dalam beberapa kasus bertindak sebagai antrian untuk data bergerak dari *publisher* ke *subscriber*. Sebuah DBMS dapat bertindak sebagai *publisher* dan distributor.
- *Subscriber* : Sebuah DBMS yang menyimpan data replikasi. Subscriber dapat menerima data dari berbagai *publisher* tergantung kepada tipe replikasi yang dipilih, subscriber dapat melewatkan data perubahan ke penerbit atau menerbitkan data ke *subscriber* lain.

Ada beberapa jenis replikasi yaitu :

- *Snapshot Replication*, suatu *copy* berkala diambil dari *database* master, baik operasi menambah (*insert*) maupun proses memperbaharui (*update*) *database slave*.
- *Synchronous Replication*, data dimasukkan ke dalam kedua *database* master dan *slave* pada waktu yang sama. Jika *database* tidak bisa berkomunikasi, maka kemudian tidak ada data yang dapat dimasukkan, namun bagaimanapun juga *database* selalu disamakan.
- *Asynchronous Replication*, perubahan di dalam master *database* dimasukkan ke suatu antrian dan mentransfer ke *database* lain pada suatu basis periode tertentu. Jika komunikasi atau suatu server *down*, sistem akan menunda perpindahan data.

B. Cloud Computing

Cloud computing adalah sebuah model yang memungkinkan untuk *ubiquitous* (dimanapun dan kapanpun), nyaman, *on-demand* akses jaringan ke sumber daya komputasi yang dapat dengan cepat dirilis atau ditambahkan. *Cloud computing* dapat digunakan dengan berbasis jaringan/Internet [8].

Secara umum arsitektur *cloud computing* terbagi menjadi 3, yaitu [9]:

- *Infrastructure as a Service (IaaS)*. IaaS menyediakan sumber daya pemroses, *storage*, kapasitas jaringan, dan sumber daya komputasi lainnya. Contoh penyedia layanan ini adalah Amazon EC2 dan TelkomCloud.
- *Platform as a Service (PaaS)*. PaaS menyediakan *platform* (bahasa pemrograman, *tools*, web server, basis data) yang berguna untuk pengembangan aplikasi yang berjalan pada infrastruktur *cloud* dan hasilnya dapat dimanfaatkan untuk konsumen. Contoh penyedia layanan ini adalah Azure SQL Database.
- *Software as a Service (SaaS)*. SaaS menyediakan layanan berupa aplikasi yang dapat digunakan oleh konsumen yang berjalan pada infrastruktur *cloud*. Contoh penyedia layanan ini adalah gmail, google docs, Office 365, dan SalesForce.

C. Database Transaction Unit (DTU)

Database Transaction Unit (DTU) adalah satuan/ukuran yang dikembangkan Microsoft untuk membantu pengguna memilih jumlah sumber daya yang dibutuhkan ketika menggunakan Azure SQL Database. Microsoft menggunakan operasi yang unik dalam permintaan *online transaction processing (OLTP)*, lalu mengukur banyaknya transaksi yang dapat diselesaikan per detik di dalam suatu basis data dalam setiap tingkat performa [10].

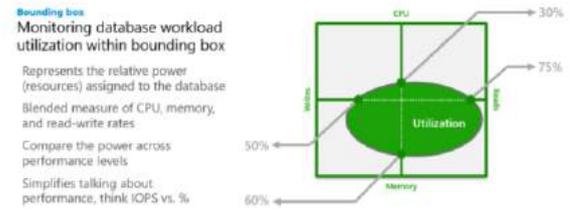
Tabel 1. Pilihan performa DTU

	Basic			Standard				Premium				
	5	10	20	50	100	125	250	500	1,000	1,750		
DTUs	5	10	20	50	100	125	250	500	1,000	1,750		
Max storage (GB)	2	250			1,000							
Max in-memory OLTP storage (GB)	N/A	N/A	N/A	N/A	N/A	1	2	4	8	14		
Max concurrent workers	30	60	90	120	200	200	400	800	1,600	2,400		
Max concurrent logins	30	60	90	120	200	200	400	800	1,600	2,400		
Max concurrent sessions	300	600	900	1,200	2,400	2,400	4,800	9,600	19,200	32,000		
Point-in-time restore	Any point last 7 days			Any point last 14 days				Any point last 35 days				
Disaster recovery	Active Geo-Replication, up to 4 offline or online (readable) secondary backups											

Pada Tabel 1, dijelaskan 3 level performa yang bisa dipilih pada Azure SQL Database. Ada level *service* dari DTU yang digunakan. Semakin besar transaksi basis data suatu aplikasi, maka DTU yang diperlukan semakin besar.

Gambar 2 menjelaskan isi dari DTU yaitu *Central Processing Unit (CPU)*, *memory*, transaksi *writes*, dan transaksi *reads*. DTU dapat dipilih menyesuaikan kebutuhan transaksi aplikasi yang menggunakan Azure SQL Database.

Database Transaction Unit – DTU



Gambar 2. Komponen di dalam DTU

IV. METODOLOGI PENELITIAN

A. Pengumpulan Data Survey Lapangan

Proses pengumpulan data dilakukan peneliti melalui kunjungan lingkungan kerja SMI Solo dan melihat secara langsung infrastruktur jaringan yang sudah ada, serta melakukan wawancara kepada tim *software developer* SMI. Kemudian peneliti mengambil data yang diperlukan untuk menganalisis. Data kuantitatif yang akan diambil adalah *throughput*, *packet loss*, *delay*, dan *jitter*. Selain itu peneliti juga mengamati *hardware* yang digunakan sebagai *server* dan mengambil data terkait basis data appssmi.com.

Setelah peneliti mendapatkan data awal, selanjutnya dilakukan analisis terhadap *server*, basis data SQL Server, dan tempat menampung basis data cadangan menggunakan teknologi *cloud computing*. Analisis dilakukan untuk mengetahui apakah kinerja *server*, basis data SQL Server, dan *cloud computing* saling mendukung untuk diterapkan atau tidak.

B. Pengambilan Data Jaringan dan Basis Data

Peneliti menggunakan beberapa alat bantu untuk memperjelas data. Alat bantu yang digunakan untuk mengambil data performa jaringan menggunakan *command prompt* dan *biznet speedtest*. Untuk memetakan struktur basis data peneliti menggunakan aplikasi DBDesigner 4.

C. Analisis Kebutuhan Infrastruktur Replikasi Basis Data

Langkah selanjutnya adalah melakukan analisis kebutuhan infrastruktur replikasi basis data yaitu data jaringan berdasarkan standarisasi TIPHON [11]. Pengolahan data pengamatan disajikan dalam bentuk grafik dan tabel sehingga mudah untuk menarik kesimpulan dan mengidentifikasi masalah. Hasil pengujian dilihat dari data yang dikumpulkan berupa data *bandwidth test (throughput)*, *delay*, *packet loss*, dan *jitter* yang disajikan dalam bentuk tabel dan grafik. Hasil evaluasi tersebut menjadi dasar untuk penerapan replikasi basis data yang akan diimplementasikan. Tahapan ini diharapkan sudah dapat menjadi jaringan yang siap digunakan pada lingkungan kerja SMI Solo menerapkan replikasi basis data.

D. Perancangan dan Implementasi Replikasi Basis Data

Peneliti merancang dan mengimplementasikan replikasi basis data sesuai dengan pertimbangan kepada *developer* SMI Solo. Pada tahap ini, peneliti mengimplementasikan replikasi menggunakan teknologi Windows Azure yaitu Azure SQL Database. Ada beberapa batasan yang dilakukan ketika melakukan replikasi dari *server* lokal ke Azure SQL Database sehingga menjadi bahan pertimbangan penerapan replikasi.

E. Monitoring

Setelah replikasi sudah diuji coba dan berhasil, tahap berikutnya adalah menganalisis beberapa tantangan yang ada sesuai dengan kesiapan infrastrukturnya SMI Solo kedepannya. Terakhir, menarik kesimpulan dari segala proses yang telah dikerjakan.

V. HASIL DAN PEMBAHASAN

A. Pembuatan Alat Uji Jaringan

Alat ukur yang digunakan untuk mengukur kualitas dari jaringan Internet *server* adalah *ping test* dan *Biznet Speedtest*. *Ping* merupakan *program tool* dari sistem operasi Windows melalui *command prompt* yang berfungsi untuk mengetahui jumlah *delay* dan *packet loss*. *Biznet Speedtest* adalah aplikasi *online* yang digunakan untuk mengetahui berapa kecepatan koneksi (*throughput*) yang sedang dipakai oleh *user*. Hasil dari pemantauan menggunakan *ping* dan *speedtest* diolah menjadi grafik dan dihitung rata-ratanya. Pengujian ini dilakukan pada *server* SMI Solo.

B. Hasil Pengujian Performa Jaringan

Peneliti ingin mengetahui kesiapan kebutuhan replikasi basis data pada SMI Solo dapat dilakukan atau tidak dengan kondisi saat ini. *Server* yang digunakan pada saat ini tidak yakin digunakan oleh *developer* karena bukan Internet dengan *bandwidth* yang *dedicated* sehingga dengan pengujian jaringan ini dapat meyakinkan *developer* kedepannya.

Pengambilan data Internet *server* SMI Solo dilakukan pada jam kerja selama 12 sesi (setiap interval 30 menit) pada hari Senin sampai Sabtu pukul 12.00 – 18.00 WIB. Data yang diambil yaitu *throughput*, *packet loss*, *delay*, dan *jitter* sesuai dengan standarisasi QoS [12]. Data kemudian diolah menjadi data statistik.

Untuk standarisasi hasil pengujian *throughput*, *delay*, *jitter*, dan *packet loss* digunakan acuan dari TIPHON seperti pada tabel 2, 3, 4.

Tabel 2. Standarisasi *throughput* TIPHON

Category	Throughput
Very Good	100%

Good	75%
Average	50%
Bad	< 25%

Tabel 3. Standarisasi *delay* TIPHON

Category	Delay
Very Good	< 150ms
Good	150 – 300 ms
Average	300 – 450 ms
Bad	> 450 ms

Tabel 4. Standarisasi *jitter* TIPHON

Category	Delay
Very Good	0 ms
Good	0 – 75 ms
Average	75 – 125 ms
Bad	125 – 225 ms

Tabel 5. Standarisasi *packet loss* TIPHON

Category	Packet Loss
Very Good	0%
Good	3%
Average	15%
Bad	25%

C. Rekapitulasi dan Analisis Data Performa Jaringan

1) Analisis *Throughput* Internet

Setelah data *throughput* didapatkan, kemudian diolah dengan menggunakan Microsoft Excel dan ditampilkan dengan perbandingan rata-rata dalam bentuk Tabel 6. Secara penilaian TIPHON, rata-rata nilai menunjukkan bahwa pada *throughput download* dan *upload* termasuk dalam kategori sangat memuaskan.

Tabel 6. Rekapitulasi pengukuran *throughput server* SMI Solo

Hari	Throughput Download (Mbps)	Throughput Upload (Mbps)	Limitasi (Mbps)	Rata-rata Pengguna Harian	Standarisasi TIPHON
Senin	3,78	3,36	4	1-10	Sangat Bagus
Selasa	3,87	3,48			Sangat Bagus
Rabu	3,94	3,47			Sangat Bagus
Kamis	3,55	3,49			Sangat Bagus
Jumat	3,97	3,6			Sangat Bagus
Sabtu	4	3,54			Sangat Bagus
Rata-rata	3,85	3,49	4	1-10	Sangat Bagus

Rata-rata *throughput download* sebesar 3,85 Mbps dan *throughput upload* sebesar 3,49 Mbps dengan pengguna harian 1-10 pengguna. Dapat dikatakan pula pemanfaatan *throughput* Internet *server* jika akan jadi *server* lokal untuk domain *appssmi.com* menunjukkan performa jaringan yang baik.

2) Analisis *Packet Loss*

Pengujian *packet loss* dengan menghitung manual paket yang hilang/loss dari *command prompt* saat tes *ping* menuju www.google.com. Pengujian dilakukan sebanyak 72 kali dengan setiap pengujiannya melakukan permintaan *echo* sebanyak 10 *byte*.

Berdasarkan data yang diambil peneliti, dapat disimpulkan bahwa nilai *packet loss* Internet *server* SMI Solo sangat rendah karena tidak ada paket yang hilang dengan nilai 0% (kategori penilaian TIPHON sangat bagus).

3) Analisis *Delay*

Tabel 7 menjelaskan *delay server* SMI Solo. *Delay* tertinggi terdapat pada hari Selasa dengan nilai 29,19 ms. Sedangkan *delay* terendah berada di hari Sabtu dengan nilai 22,45 ms. Secara keseluruhan nilai rata-rata *delay* untuk Internet *server* SMI Solo memiliki nilai 25,88 ms dimana dalam standarisasi TIPHON termasuk kategori sangat bagus.

Tabel 7. Rekapitulasi hasil pengukuran *delay server* SMI Solo

Hari	Delay (ms)	Standarisasi TIPHON
Senin	23,31	Sangat Bagus
Selasa	29,19	Sangat Bagus
Rabu	26,59	Sangat Bagus
Kamis	28,65	Sangat Bagus
Jumat	25,42	Sangat Bagus
Sabtu	22,45	Sangat Bagus
Rata-rata	25,88	Sangat Bagus

4) Analisis *Jitter*

Tabel 8 menjelaskan performa *jitter* di *server* SMI Solo. *Jitter* tertinggi terdapat di hari Rabu dengan nilai 0,19 ms dan terendah di hari Kamis dan Jumat dengan nilai 0,05 ms. Secara keseluruhan nilai rata-rata *jitter* sebesar 0,09 dan termasuk ke dalam kategori bagus berdasarkan penilaian TIPHON.

Tabel 8. Rekapitulasi hasil pengukuran *jitter server* SMI Solo

Hari	Jitter (ms)	Standarisasi TIPHON
Senin	0,07	Bagus
Selasa	0,1	Bagus
Rabu	0,19	Bagus
Kamis	0,05	Bagus
Jumat	0,05	Bagus
Sabtu	0,06	Bagus
Rata-rata	0,09	Bagus

D. Perancangan dan Implementasi Replikasi Basis Data

1) Membuat domain *aplikasismi.com*

Pada Gambar 3, peneliti membuat domain lain dengan basis data *appssmi.com* untuk kepentingan replikasi sehingga mencegah hal-hal yang tidak diinginkan. Domain ini datanya di-hosting pada *server* lokal SMI Solo.



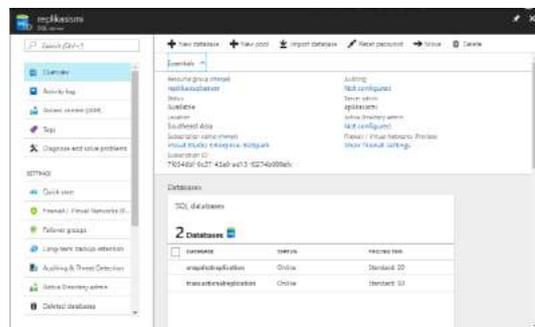
Gambar 3. Tampilan situs *aplikasismi.com*

2) *Server* lokal SMI Solo sebagai tempat hosting

Basis data dan data *website* *aplikasismi.com* ditempatkan pada *server* lokal SMI Solo. *Server* lokal dipilih untuk menjadi tempat utama *website* dikarenakan spesifikasi *server* dan performa jaringan yang sangat memuaskan.

Dengan perancangan seperti ini, peneliti akan melakukan replikasi ke *Azure SQL Database*. *Azure* dipilih karena teknologi *cloud* yang aman, selalu *online* 24 jam, rekomendasi developer *website* SMI, dan memiliki *interface* yang sederhana sehingga mudah dimengerti.

3) Membuat *instance Server* dan *Azure SQL Database* di Microsoft *Azure*



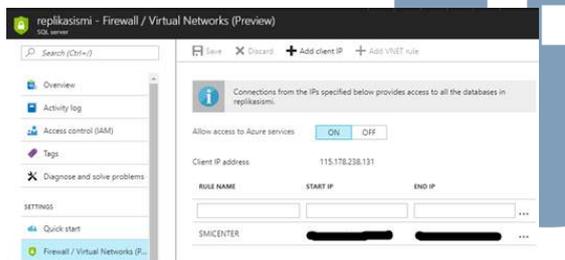
Gambar 4. Membuat 1 *instance server* dan 2 *Azure SQL Databases* di Microsoft *Azure*

Gambar 4 menjelaskan ketika peneliti ingin membuat 1 atau lebih *Azure SQL Database*, memerlukan setidaknya 1 *server* untuk menampung *database* tersebut. Peneliti menentukan lokasi di Southeast Asia dengan pertimbangan dekat, murah, dan memiliki *latency* yang paling rendah dari Indonesia.

Setiap Azure SQL Database memiliki *performance* yang berbeda-beda disesuaikan dengan kebutuhan pengguna. Peneliti memilih *performance* dengan tipe *standard* dengan minimal 10 DTU dan *storage* sebesar 5GB dijelaskan pada Gambar 5. Peneliti menggunakan 10 DTU dikarenakan sudah pernah mencoba menggunakan tipe *basic* dan hasilnya ada beberapa proses DTU akan *overload* hingga 100% sehingga akan mengurangi proses kecepatan replikasi basis data. Untuk ukuran *database* bisa memilih antara 1 – 250 GB tanpa dikenakan biaya, peneliti hanya memilih muatan *database* sebesar 5GB.



Gambar 5. Pilihan *performance* tipe *Standard* untuk Azure SQL Database



Gambar 6. Menambahkan pengaturan *firewall*

Gambar 6 menjelaskan ketika *server* akan diakses suatu *client*, maka IP *client* tersebut harus dikonfigurasi pada pengaturan *firewall*. Peneliti mengkonfigurasi *rule name*, *start IP*, dan *end IP* yang merupakan identitas *server* SMI Solo. *Start IP* dan *end IP* sengaja dirahasiakan karena IP publik statis *server* SMI Solo.

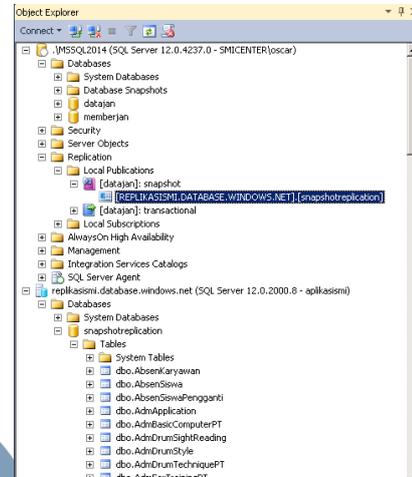
4) Proses Replikasi menggunakan SQL Server Management Studio 2014

Gambar 7 menjelaskan ketika konfigurasi replikasi sudah berhasil dilakukan, maka akan ada keterangan tambahan yang muncul pada saat selesai konfigurasi *Publisher* dan *Subscriber*. Peneliti membuat jadwal replikasi *snapshot* pada waktu terpendek yaitu selalu *update* selama 1 jam sekali (pengaturan waktu *backup* dapat disesuaikan dengan kebutuhan, peneliti melampirkan konfigurasi di lampiran B). Jadi, setiap 1 jam akan ada proses besar untuk replikasi jenis ini.

Pada Gambar 8, digambarkan beberapa grafik terhadap proses terjadinya replikasi jenis *snapshot* pertama kali. Terlihat DTU dan Log I/O yang paling tinggi prosesnya sebanyak 69,21%, CPU bekerja

sebanyak 37,39% dan *database size* terisi 1,82% dari total 5GB.

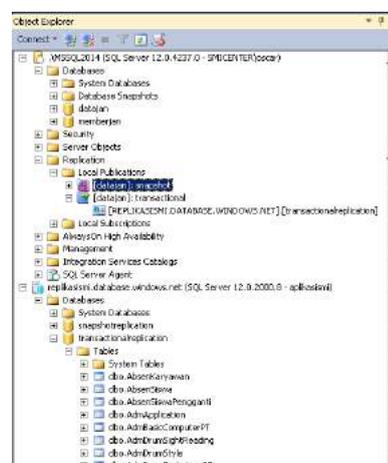
Gambar 9 menjelaskan ketika konfigurasi replikasi sudah berhasil dilakukan, maka akan ada keterangan tambahan yang muncul pada saat selesai konfigurasi *Publisher* dan *Subscriber*.



Gambar 7. Tampilan *snapshot replication* di SSMS 2014

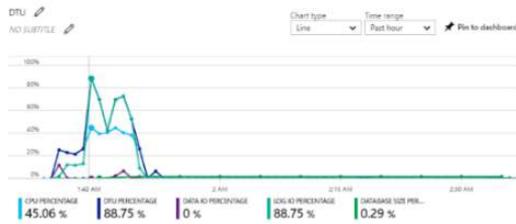


Gambar 8. Proses replikasi pertama kali pada jenis *snapshot replication*



Gambar 9. Tampilan *transactional replication* di SSMS 2014

Pada Gambar 10, digambarkan beberapa grafik terhadap proses terjadinya replikasi jenis *transactional* pertama kali. Terlihat DTU dan Log I/O yang paling tinggi prosesnya sebanyak 88,75%, CPU bekerja sebanyak 45,06% dan *database size* baru terisi 0,29% dari total 5GB.



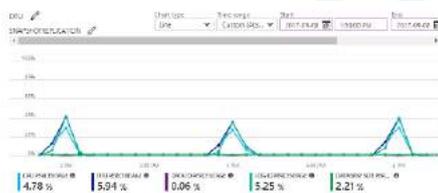
Gambar 10. Proses replikasi pertama kali pada jenis *transactional replication*

E. Hasil Pengujian Replikasi Basis Data

Pengujian replikasi basis data menggunakan 4 skenario yang di-submit pada waktu yang sama dengan diulangi selama 5 kali dengan 13 klien yang berperan sebagai *customer service*. Data yang dihasilkan kemudian akan diolah menjadi data statistik pada Microsoft Excel. Setelah nilai hasil replikasi telah didapat, lalu dibandingkan dengan yang lainnya.

Besarnya DTU yang terpakai dipengaruhi dari Log Input/Output dan Central Processing Unit yang terpakai. Berikut perbedaannya melalui DTU yang digunakan (lebih jelas jika dilihat lewat *transactional replication*) :

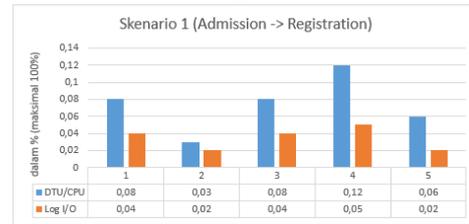
1) Snapshot Replication



Gambar 11. Rata-rata proses *snapshot replication* selama pengujian

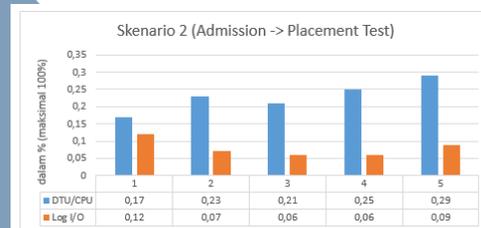
Gambar 11 menjelaskan rata-rata *performance* dari CPU, DTU, data I/O, log I/O, dan *database size* pada *snapshot replication* selama pengujian. Rata-rata *performancenya* 4,78% pada CPU, 5,94% pada DTU, 0,06% pada data I/O, 5,25% pada log I/O, dan ukuran *database* sebesar 2,21% dari 5GB. *Database* yang dikonfigurasi replikasi jenis *snapshot* akan besar penggunaan DTUnya setiap 1 jam sekali (sistem otomatis melakukan replikasi selama 20 menit). Penggunaan DTU tergolong cukup besar setiap 1 jam sekali karena sistem akan meng-overwrite data yang ada.

2) Transactional Replication



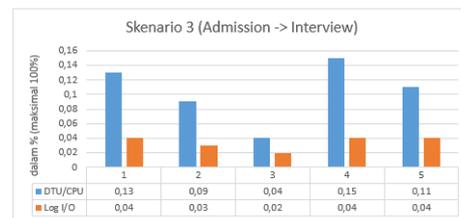
Gambar 12. Grafik *performance* hasil replikasi ke Azure untuk Skenario 1

Pada Gambar 12 menjelaskan nilai *performance* dari DTU, CPU, dan Log I/O pada skenario 1. Pada proses replikasi jenis transaksional, DTU dan CPU bernilai sama. Untuk proses DTU yang paling banyak ada di percobaan keempat sebanyak 0,12% dan paling sedikit di percobaan kedua sebanyak 0,03%. Input paling banyak terjadi pada percobaan keempat sebanyak 0,05% dan paling sedikit di percobaan kedua dan terakhir sebanyak 0,02%.



Gambar 12. Grafik *performance* hasil replikasi ke Azure untuk Skenario 2

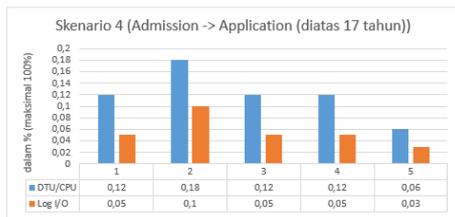
Pada Gambar 13 menjelaskan nilai *performance* dari DTU, CPU, dan Log I/O skenario 2. Untuk proses DTU yang paling banyak ada di percobaan kelima sebanyak 0,29% dan paling sedikit di percobaan pertama sebanyak 0,17%. Input paling banyak terjadi pada percobaan pertama sebanyak 0,12% dan paling sedikit di percobaan ketiga dan keempat sebanyak 0,06%.



Gambar 14. Grafik *performance* hasil replikasi ke Azure untuk Skenario 3

Pada Gambar 14 menjelaskan nilai *performance* dari DTU, CPU, dan Log I/O skenario 3. Untuk proses DTU yang paling banyak ada di percobaan keempat sebanyak 0,15% dan paling sedikit di percobaan ketiga sebanyak 0,04%. Input paling banyak terjadi pada percobaan pertama, keempat, dan

kelima sebanyak 0,04% dan paling sedikit di percobaan ketiga sebanyak 0,02%.



Gambar 15. Grafik *performance* hasil replikasi ke Azure untuk Skenario 4

Pada Gambar 15 menjelaskan nilai *performance* dari DTU, CPU, dan Log I/O skenario 4. Untuk proses DTU yang paling banyak ada di percobaan kedua sebanyak 0,18% dan paling sedikit di percobaan kelima sebanyak 0,06%. Input paling banyak terjadi pada percobaan kedua sebanyak 0,1% dan paling sedikit di percobaan kelima sebanyak 0,03%.



Gambar 16. Rata-rata proses *transactional replication*

Gambar 16 menjelaskan rata-rata *performance* dari CPU, DTU, data I/O, log I/O dan *database size* dari *transactional replication* selama pengujian. Rata-rata *performancenya* 0,09% pada CPU, 0,09% pada DTU, 0% pada data I/O, 0,04% pada log I/O, dan ukuran *database* sebesar 1,82% dari 5GB. Penggunaan DTU tergolong kecil karena setiap waktu sistem akan menyalin data setiap kali berubah.

F. Analisis Replikasi Basis Data

1) Snapshot Replication

Pengujian *snapshot replication* dapat dilihat secara detail hanya beberapa saat ada proses-proses ringan yang terjadi tetapi tidak mengubah isi *database* jika belum saatnya sinkronisasi otomatis yang sudah dikonfigurasi. Pengujian selama proses itu berlangsung, data tidak akan langsung ter-*update* sehingga harus menunggu sinkronisasi berikutnya.

Pada replikasi jenis *snapshot*, rata-rata dari 10 DTU yang dialokasikan, dipakai sebanyak 40%-an selama pengujian. Proses ini terbilang memakan banyak *resources* dan tidak efisien karena *database* akan selalu di-*overwrite* datanya selama waktu yang ditentukan.

2) Transactional Replication

Tabel 9. Rekapitulasi hasil pengukuran *performance transactional replication*

Skenario	DTU/CPU (%)	Log I/O (%)	Utilisasi (%)
1	0,074	0,034	100
2	0,23	0,08	
3	0,104	0,034	
4	0,12	0,056	
Rata-rata	0,132	0,064	

Setelah data *performance* didapatkan, kemudian diolah dengan menggunakan Microsoft Excel dan ditampilkan dengan perbandingan rata-rata dalam pada Tabel 9. Melalui rata-rata tersebut terlihat penggunaan DTU/CPU besar pada skenario 2 dan kecil pada skenario 1. Sedangkan untuk log I/O besar pada skenario 2 dan kecil pada skenario 1 dan 3.

Secara pengamatan, didapati bahwa dengan 4 skenario yang tergolong berat (berdasarkan pengamatan peneliti dan developer appssmi) dengan 13 klien *submit* bersamaan, replikasi jenis transaksional prosesnya sangat ringan dengan 10 DTU. Oleh karena itu, jika nanti Azure SQL Database di-*downgrade* performanya menjadi 5 DTU dapat menghemat biaya dan performa tetap terjamin berjalan dengan baik.

G. Analisis Tantangan Pengembangan Infrastruktur Replikasi Basis Data SMI

1) Analisis Perkembangan Basis Data Appssmi.com

Data internal SMI dengan domain appssmi.com per 22 Oktober 2017 memiliki basis data yang berukuran 97 MB di Azure (Azure memiliki algoritma kompresi data, ukuran data sebenarnya bila di-*download* dari Azure bermuatan 310 MB). Data ini digunakan oleh 12 cabang SMI di Indonesia dan cabang selalu bertambah setiap tahunnya.

Tabel 10. *Traffic* basis data appssmi.com selama 12 hari

Tanggal	Jumlah Data Masuk (KB)	Jumlah Transaksi
9 Oktober 2017	380,69	404
10 Oktober 2017	498,63	531
11 Oktober 2017	355,58	378
12 Oktober 2017	387,18	413
13 Oktober 2017	258,09	277
14 Oktober 2017	220,09	237
16 Oktober 2017	362,25	388
17 Oktober 2017	283,36	295
18 Oktober 2017	391,6	418
19 Oktober 2017	222,41	247
20 Oktober 2017	300,34	321
21 Oktober 2017	258,75	278
Rata-rata per hari	326,58	348,92

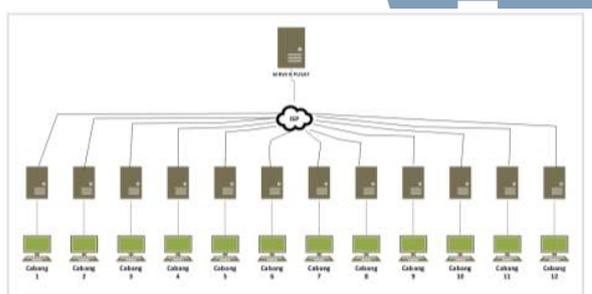
Dilihat dari perkembangan data selama 12 hari pada Tabel 10, maka peneliti memiliki estimasi jika 1

tahun appssmi.com (sementara dengan 12 cabang) yang buka selama 317 hari (sudah dikurangi libur hari minggu setahun ada 48, tanpa libur nasional) dikali rata-rata data masuk perhari yaitu 326,58 KB akan menambah data sebanyak 103525,86 KB (103,53 MB dalam desimal) secara linier. Muatan ini tergolong kecil untuk 12 cabang karena isinya teks.

2) Analisis Kesiapan Kebutuhan Infrastruktur Replikasi Basis Data di Setiap Cabang SMI

Dengan data yang akan selalu bertambah, kesiapan kebutuhan infrastruktur dan performa jaringan Internet di setiap cabang SMI perlu ditingkatkan dan diefisiensikan untuk menambah ketersediaan data yang terdistribusi. Di masa yang akan datang, peneliti menganalisis terkait infrastruktur secara khusus dalam pengelolaan jaringan Internet yang baik untuk mengimplementasikan replikasi basis data.

Jaringan Internet di setiap cabang SMI merupakan kebutuhan primer. Hal ini didasari oleh setiap proses belajar mengajar dan pengelolaan data internal memerlukan Internet. Jaringan Internet dengan *bandwidth dedicated* harganya lebih mahal bila dibandingkan dengan jaringan Internet dengan *bandwidth up to* (karena tingkat kestabilan dan kehandalan).



Gambar 17. Pengembangan arsitektur teknologi SMI di masa depan

Dari Gambar 17, dijelaskan setiap cabang memiliki *server* lokal sendiri yang menampung data internal perusahaan. Dengan adanya dukungan *server* lokal, koneksi untuk mengakses data internal SMI akan cepat dan stabil (tidak terlalu tergantung Internet) sehingga penggunaan Internet akan terfokus terhadap proses belajar mengajar selama jam operasional SMI. Selanjutnya, bila jam SMI tutup, masing-masing *server* sudah diberikan jadwal (setiap *server* cabang untuk melakukan replikasi tidak boleh sama jadwalnya 1 dengan yang *server* cabang lainnya) untuk melakukan *merge replication* agar setiap cabang melakukan proses *backup* basis datanya ke *server* pusat terlebih dahulu sampai semua cabang selesai. Akhirnya, *server* pusat memiliki jadwal untuk melakukan *merge replication* sekali lagi ke setiap *server* cabang agar semua mendapat basis data yang sama.

VI. KESIMPULAN

Setelah melakukan serangkaian penelitian, pengujian, dan analisa, maka kesimpulan dari penelitian ini adalah sebagai berikut :

- Kesiapan kebutuhan infrastruktur replikasi basis data SMI Solo sudah baik dengan *bandwidth* Internet server SMI Solo dengan *up to* 4 Mbps dan basis data internal yang kecil (hanya teks).
- Kinerja dari jaringan *server* SMI Solo memiliki kualitas yang sangat memuaskan dilihat dari rata-rata *bandwidth download* 3,85 Mbps, *bandwidth upload* 3,49 Mbps, *packet loss* 0%, *delay* 25,88 ms, dan *jitter* 0,09 ms.
- Kinerja dari replikasi basis data dalam hal penggunaan *resource* CPU, DTU, data I/O, log I/O, dan *database size* dapat dilihat dari :
 - Rata-rata penggunaan CPU dari *snapshot replication* yaitu 4,78% dalam skenario ketika dijadwalkan 1 jam sekali melakukan sinkronisasi dan *transactional replication* yaitu 0,09%.
 - Rata-rata penggunaan DTU dari *snapshot replication* yaitu 5,94% dalam skenario ketika dijadwalkan 1 jam sekali melakukan sinkronisasi dan *transactional replication* yaitu 0,09%.
 - Rata-rata data I/O dari *snapshot replication* yaitu 0,06% dalam skenario ketika dijadwalkan 1 jam sekali melakukan sinkronisasi dan *transactional replication* yaitu 0%.
 - Rata-rata log I/O dari *snapshot replication* yaitu 5,25% dalam skenario ketika dijadwalkan 1 jam sekali melakukan sinkronisasi dan *transactional replication* yaitu 0,04%.
 - Rata-rata *database size* dari total 5GB pada *snapshot replication* yaitu 2,21% dan *transactional replication* yaitu 1,82%. Perbedaan ini didasarkan pada data yang direplikasikan pada proses *transactional replication* semua tabel yang direplikasikan harus memiliki *primary key*, sedangkan tabel "trx" pada appssmi.com tidak memiliki *primary key* sehingga tidak dapat direplikasikan.

Saran untuk penelitian selanjutnya adalah mencoba membandingkan dengan layanan cloud computing lain sebagai perbandingan dengan apa yang dimiliki oleh Azure, seperti halnya Google Cloud Computing dan Alibaba Cloud Computing. Amazon Web Service tidak disertakan karena sampai dengan saat tulisan ini dibuat, AWS belum mendukung fitur replikasi yang dibutuhkan. Selain itu juga bisa melakukan perbandingan algoritma replikasi data seperti pada [13] dan [14].

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada Sekolah Musik Indonesia Solo beserta tim pengembang appssmi.com yang telah mendukung penelitian ini dengan menyediakan infrastruktur serta akses pada arsitektur sistem yang digunakan.

DAFTAR PUSTAKA

- [1] M. Christian Mazilu, "Database Replication", Database Systems Journal Vol 1, No 2, hal 33-38, 2010.
- [2] P. Silitonga, "Replikasi Basis Data Pada Sistem Pengolahan Data Akademik Universitas Katolik Santo Thomas". Jurnal TIME Vol III No 1. hal. 32-36, 2014.
- [3] F. Zaini, L. Atika, & U. Ependi, "Implementasi Metode Asynchronous pada Replikasi Database Inventaris Universitas Bina Darma", Jurnal Informatika Universitas Bina Darma Palembang, Oktober 2015.
- [4] M. Holmgren, "Multi-Master Database Replication and e-Learning – Theoretical and Practical Evaluation" (Disertasi), 2015.
- [5] C-O. Truica, A. Boicea, & F. Radulescu, "Asynchronous Replication in Microsoft SQL Server, PostgreSQL, and MySQL". International Conference on Cyber Science and Engineering Hal. 50-55, 2013.
- [6] N.H. Shahapure & P. Jayarekha, "Replication: A Technique for Scalability in Cloud Computing", International Journal of Computer Applications Vol 122, No 5, hal 13-18, July 2015.
- [7] G. Coulouris, J. Dollimore, T. Kindberg, & G. Blair, "Disributed Systems: Concepts and Design, 5th Ed", Pearson, 2011.
- [8] P. Mell & T. Grance, "The NIST Definition of Cloud Computing", NIST Special Publication 800-145, September 2011.
- [9] A. Syaikh, "Komputasi Awan (Cloud Computing) Perpustakaan Pertanian", Jurnal Pustakawan Indonesia Vol 10, No 1, hal 1-12, 2010.
- [10] C. Rabeler, "Migrating SQL Server Databases to Azure, 1st Ed", 2016.
- [11] The European Telecommunications Standards Institute, "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON): General aspects of Quality of Service (QoS)", ETSI TR 101 329, France, Vol 2.1.1, June 1999.
- [12] M. Flanagan, R. Froom, & K. Turek, "Cisco Catalyst QoS: Quality of Service in Campus Networks 1st Ed", 2003.
- [13] MC. Lee, FY. Leu, YP. Chen, "PFRF: An adaptive data replication algorithm based on star-topology data grids", Future Generations Computer Systems, Vol 28 Issue 7, July 2012.
- [14] JW. Lin, CH. Chen, J.M. Chang, "QoS-Aware Data Replication for Data-Intensive Applications in Cloud Computing Systems", IEEE Transactions on Cloud Computing, Vol 1 Issue 1, June 2013.

Sistem Penunjang Keputusan Penilaian Kinerja Karyawan Berprestasi Menggunakan Acuan MBO dan Metode AHP Menggunakan Aplikasi Expert Choice

Studi Kasus Universitas Tanri Abeng

Rudi Sutomo¹, Johny Hizkia Siringo Ringo²

Fakultas Teknik & Teknologi, Program Studi Teknik Informatika, Universitas Tanri Abeng, Jl Swadarma Raya No.58 Ulujami Pesanggrahan, Jakarta Selatan 12250 Indonesia

rudi.sutomo@tau.ac.id

johny.hizkia@tau.ac.id

Diterima 23 Maret 2018

Disetujui 8 Juni 2018

Abstract—In the determination of employees who have the achievement required assessment of the assessment. There are several policy setting criteria to be able to prevent subjective decision-makers such as the influence of "likes and dislikes", so it is often wrong to judge employees. This paper is intended to provide a solution to the problem of choosing qualified and qualified employees using the AHP method and using decision support systems application Expert Choice to assist decision making in determining outstanding employees based on MBO method references. From the result of comparison of criteria weight that has been inputted and has been adjusted with comparison matrix Sub Criteria of Achievement then got that occupy the highest priority data that is consumer satisfaction with point 0,434, discipline with point 0,285, operational performance with point 0,071 and achievement with point 0,058 with inconsistency 0, 03 with 0 missing judgments. The results of AHP calculations will be applied to produce the highest intensity of employee priority outputs so that employees with the highest score are eligible for rewards or rewards.

Index Terms—AHP, Aplikasi Expert Choice, MBO, SPK.

I. PENDAHULUAN

Institusi memiliki jumlah karyawan cukup banyak dalam perencanaan jenjang karir tiap karyawan dan pergantian jabatan dalam perusahaan sering terjadi dan juga menghabiskan banyak waktu untuk memberikan penilaian yang akurat karena divisi Human Resource Departement (HRD) mungkin tidak terlalu memiliki data prestasi atau karakter setiap karyawan yang berada dalam lingkungan Institusi Perguruan Tinggi. Hal ini menjadi kendala yang cukup signifikan dalam rangka menyusun jenjang karir dari tiap karyawan dan kaderisasi atau pergantian jabatan dalam struktur organisasi dengan menggunakan data yang ada.

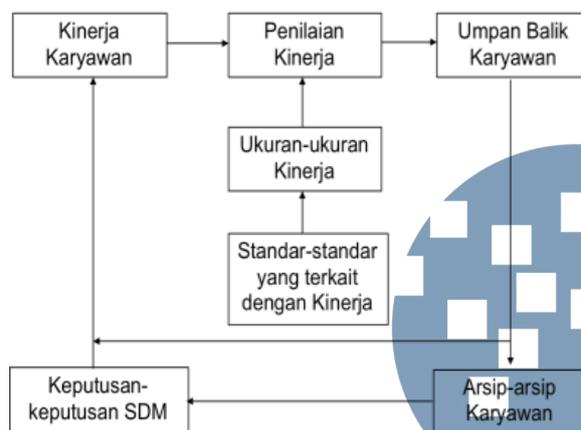
Pelaksanaan kerja oleh karyawan berbeda-beda disesuaikan dengan divisi dalam organisasi dan menjadi ukuran kinerja di dalam ukuran standar kerja.

Penulis mencoba menggunakan metode MBO secara intinya dan dimplementasi ke dalam sistem pendukung keputusan metode AHP sebagai pengukuran kinerja karyawan di perguruan tinggi yang diharapkan nantinya dapat membantu para pembuat keputusan di suatu perguruan tinggi dalam memutuskan alternatif-alternatif terbaik dalam pengukuran kinerja karyawan dan akhirnya penentuan karyawan berprestasi. Dengan tujuan membangun sebuah sistem penunjang keputusan untuk melakukan analisa pemilihan karyawan berprestasi menggunakan metode Analytic Hierarchy Process (AHP), tentunya menggunakan beberapa kriteria atau faktor- faktor penilaian. Dalam hal ini karyawan akan dibandingkan satu dengan yang lainnya sehingga memberikan output nilai prioritas yang memiliki point nilai karyawan terhadap setiap kinerja karyawan berprestasi. Hal yang penting dalam penelitian ini yaitu Sistem penunjang keputusan ini membantu melakukan penilaian karyawan dengan melakukan penilaian angka kriteria, dan angka nilai bobot.[4] Hasil dari penilaian ini memberikan data untuk pengambil keputusan yang terkait dengan masalah pemilihan karyawan berprestasi, sehingga karyawan yang berprestasi layak diberi penghargaan atau peningkatan karir. Maka perlu dikaji lebih lanjut proses pelaksanaan sistem penunjang keputusan penilaian karyawan berprestasi menggunakan acuan MBO dan metode AHP dengan menggunakan aplikasi Expert Choice di kampus Universitas Tanri Abeng.

II. LANDASAN TEORI

A. Elemen-elemen Kunci Penilaian Kinerja

Dalam Pengertian dari manfaat penilaian kinerja diantaranya sebagai berikut harus adanya Perbaikan kinerja, penyesuaian kompensasi, penempatan karyawan, Kebutuhan pelatihan dan pengembangan, Perencanaan dan pengembangan karir, Mendeteksi kelemahan proses staffing, Mendeteksi ketidaktepatan informasi, Mendeteksi kesalahan desain jabatan, Menjamin kesempatan kerja yang setara, Mendeteksi faktor eksternal yang mempengaruhi kinerja, Memberikan umpan balik bagi divisi SDM.[1] Elemen-elemen Kunci Penilaian Kinerja dapat dilihat pada skema pada Gambar 1.



Gambar 1. Elemen-Elemen Kunci Penilaian Kinerja

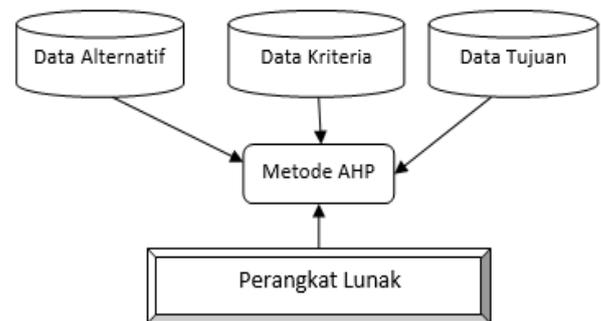
B. Metode Analytic Hierarchy Process (AHP)

AHP merupakan suatu model pendukung keputusan yang dikembangkan oleh Thomas L. Saaty. Model pendukung keputusan ini akan menguraikan masalah multi faktor atau multi kriteria yang kompleks menjadi suatu hirarki, Hirarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang kompleks dalam suatu struktur multi level dimana level pertama adalah tujuan, yang diikuti level faktor, kriteria, subkriteria, dan seterusnya ke bawah hingga level terakhir dari alternatif. Dengan hierarki, suatu masalah yang kompleks dapat diuraikan ke dalam kelompok-kelompoknya yang kemudian diatur menjadi suatu bentuk hierarki sehingga permasalahan akan tampak lebih terstruktur dan sistematis.[7]

AHP sering digunakan sebagai metode pemecahan masalah dibanding dengan metode yang lain karena alasan-alasan sebagai berikut :

1. Struktur yang berhirarki, sebagai konsekuensi dari kriteria yang dipilih, sampai pada subkriteria yang paling dalam.
2. Memperhitungkan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh pengambil keputusan.

3. Memperhitungkan daya tahan output analisis sensitivitas pengambilan keputusan.
4. Model Sistem Penunjang Keputusan penilaian kinerja karyawan dilihat pada gambar.



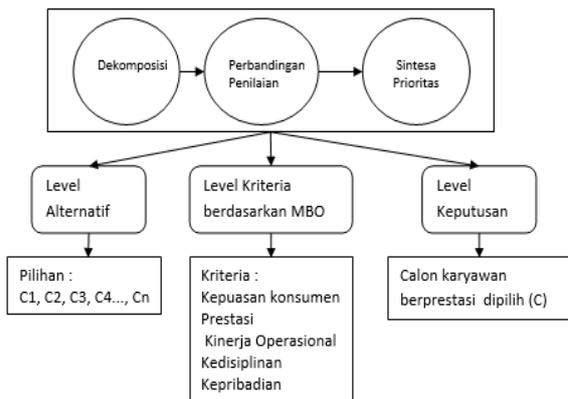
Gambar 2. Model SPK

C. Management by Objectives (MBO)

Penilaian kinerja karyawan dengan metode Management by Objectives mewajibkan karyawan untuk menyusun konsep target jangka pendek dan kemudian mendiskusikannya dengan manajer. Jika diterima oleh manajernya, target tersebut menjadi tolok ukur penilaian kinerja karyawan tersebut.[2] Penilaian kinerja karyawan dengan metode Management by Objectives dapat dilaksanakan pada pekerjaan yang output-nya dapat diukur secara kuantitatif. Manajemen Berbasis Obyek (Management by Objectives), mengharuskan para manajer untuk menetapkan tujuan-tujuan terukur yang spesifik untuk setiap karyawan berdasarkan hasil diskusi dengan karyawan tersebut, lalu secara periodik membahas kemajuan tujuan-tujuan tersebut. Penerapan metode Management by Objectives terdiri dari enam langkah. Berikut ini langkah-langkah penerapan metode Management by Objectives dalam menilai kinerja karyawan [5].

III. METODOLOGI PENELITIAN

Berdasarkan kerangka teoritis dan penelitian terdahulu maka dapat disusun suatu kerangka pikir mengenai penilaian kinerja karyawan berdasarkan acuan metode Management By Objective (MBO) dan sistem penilaian dengan menggunakan metode Analytical Hierarchy Process (AHP).



Gambar 3. Kerangka Pikir SPK Penilaian Kinerja Karyawan Berdasarkan Acuan Metode MBO dan Perhitungan Hasil dengan Metode AHP

Dalam kerangka pikir di dalamnya dibagi tiga level yaitu level keputusan yaitu Keputusan Karyawan berprestasi. Level Kriteria yaitu Kepuasan Konsumen, Kinerja Operasional, Prestasi, Kedisiplinan, kepribadian berdasarkan metode Management By Objective (MBO). Dan terakhir adalah level Alternatif yaitu pada Calon (C) akan dipilih berdasarkan kriteria telah ditentukan sesuai Divisinya.

IV. HASIL DAN PEMBAHASAN

Dalam penelitian ini bermaksud memahami teoritis dan melakukan eksplorasi dengan mengungkapkan logika-logika pemikiran. Proses penelitian dimulai dengan menyusun pengumpulan kebutuhan, melakukan analisa, desain dan diimplementasikan ke dalam penggunaan aplikasi.

A. Alat Penelitian

Menurut Jogiyanto, terdapat dua jenis penelitian, yaitu riset eksploratori (exploratory research) dan riset pengujian hipotesis (hypothesis testing). Penelitian ini menggunakan riset eksploratori, yaitu jenis penelitian yang tujuannya untuk mengenai konsep atau pola yang digunakan dalam penelitian. Dalam penelitian ini, peneliti belum memiliki gambaran akan definisi atau konsep penelitian. Peneliti akan mengajukan what untuk menggali informasi lebih jauh. Sifat dari penelitian ini adalah kreatif, fleksibel, terbuka, dan semua sumber dianggap penting sebagai sumber informasi. Tujuan dari penelitian ini adalah untuk menjadikan topik baru lebih dikenal oleh masyarakat luas, memberikan gambaran dasar mengenai topik bahasan, menggeneralisasi gagasan dan mengembangkan teori yang bersifat tentatif, membuka kemungkinan akan diadakannya penelitian lanjutan terhadap topik yang dibahas. Data diperoleh dengan mempelajari literatur-literatur yang berkaitan dengan penelitian [4].

B. Bahan Penelitian

Dari faktor-faktor yang menentukan pemilihan karyawan berprestasi yaitu menggunakan metode Management by Objective (MBO) sebagai acuan dasar perhitungan kinerja karyawan dengan menggunakan metode Analytical Hierarchy Process (AHP) dan digunakan pada divisi Staff pada Divisi BAAK, Staff pada Divisi UPT.

Pada bagian penggunaan aplikasi dan implementasi dengan menggunakan Aplikasi Expert Choice sebagai Sistem Penunjang Keputusan Penilaian Kinerja Karyawan di Universitas Tanri Abeng.

C. Metode Pengumpulan Data

Pengumpulan data dengan wawancara ini dilakukan untuk mencari data dan informasi tentang hal-hal yang dibutuhkan dalam penelitian. Wawancara dilakukan dengan lembaga atau institusi dijadikan objek penelitian. Wawancara yang dilakukan lebih menitikberatkan bagaimana standar prosedur melakukan penilaian kinerja karyawan, khususnya dalam pembobotan pada prioritas kriteria dan alternatif.

D. Metode Penarikan Sampel

Adapun metode penarikan sampel yang akan digunakan dalam penelitian ini adalah menentukan dengan cara sengaja (purposive sampling) yaitu pada Divisi BAAK, Divisi UPT diambil dua Divisi dari sebelas Divisi pada Struktur Organisasi Universitas Tanri Abeng dengan alasan karena berkaitan langsung dengan operasional di lapangan dan dengan pemilihan dari segi operasional akan menjadi kandidat SDM di segi strategis ke depannya [6].

E. Definisi Level Kriteria

Berikut ini adalah kriteria dari metode Management By Objective (MBO) disesuaikan dengan di lapangan dengan ukuran pada Sasaran Kinerja Individu (SKI) yang menjadi dasar dalam penentuan level kriteria dari Model AHP yang ditunjukkan pada Tabel 1.

Tabel 1. Definisi Level Kriteria

Definisi Level Kriteria	Keterangan
Kepuasan Konsumen	Memurut Kotler (2000), mengatakan bahwa kepuasan konsumen merupakan tingkat perasaan seseorang setelah membandingkan antara kinerja produk yang ia rasakan dengan harapannya.
Kinerja Operasional	Memurut Drucker (2002, p.134) adalah tingkat prestasi atau hasil nyata dicapai, dipergunakan untuk diperoleh suatu hasil positif.
Prestasi	Memurut Bernard dan Russel (1993:378), memberikan definisi tentang prestasi kerja sebagai berikut : "performance is defined as the record of outcome produced on a specified job function or activity during a specified time period". (Prestasi kerja didefinisikan sebagai catatan dari hasil-hasil yang diperoleh melalui fungsi-fungsi pekerjaan tertentu atau kegiatan selama tempo waktu tertentu).
Kedisiplinan	Memurut Keith Davis (1985 : 366) mengemukakan bahwa: "discipline is management action to enforce organizational standards". Pengertian disiplin tersebut diinterpretasikan sebagai pelaksanaan manajemen untuk memperteguh pedoman-pedoman organisasi dalam suatu organisasi atau lembaga. Pengertian ini pada dasarnya merupakan pelajaran, patuh, taat, kesetiaan, hormat kepada ketentuan peraturan/norma yang berlaku dalam hubungannya dengan disiplin pegawai, disiplin merupakan unsur pengikat, unsur integrasi dan merupakan unsur yang dapat mengarahkan kerja pegawai, bahkan dapat pula sebaliknya.
Kepribadian	Memurut Gordon W.Allport. Kepribadian adalah suatu organisasi yang dinamis dari sistem psiko-fisik individu yang menentukan tingkah laku dan pemikiran individu secara khas. Terjadinya interaksi psiko-fisik mengarahkan tingkah laku manusia. Maksud dinamis pada pengertian tersebut adalah perilaku mungkin saja berubah-ubah melalui proses pembelajaran atau melalui pengalaman-pengalaman, reward, punishment, pendidikan, dan sebagainya.

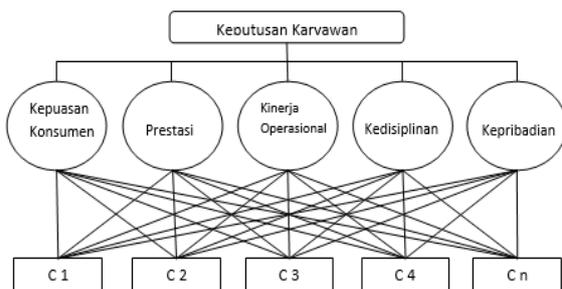
F. Identifikasi Level Kriteria

Identifikasi sumber data yang akan dianalisa untuk menentukan rumusan kriteria pengukuran karyawan dengan data aturan klarifikasi tujuan, kriteria dan alternatif pengambilan keputusan dapat dilihat pada Tabel 2.

Tabel 2. Data Klasifikasi Level Keputusan

Level Tujuan	Level Kriteria	Level Alternatif				
Pemilihan Karyawan	Kepuasan Konsumen	C1	C2	C3	C4	Cn
	Prestasi	C1	C2	C3	C4	Cn
	Kinerja operasional	C1	C2	C3	C4	Cn
	Kedisiplinan	C1	C2	C3	C4	Cn
	Kepribadian	C1	C2	C3	C4	Cn

Tujuan, kriteria dan alternatif keputusan dalam sistem penunjang keputusan dalam pengukuran kinerja karyawan dijelaskan pada Gambar 4.



Gambar 4. Struktur Hierarki Keputusan AHP pada SPK Penilaian Kinerja Karyawan

G. Pengukuran dan Pengamatan Variabel di Lapangan

Dalam tahap pengumpulan kebutuhan diperlukan data untuk masuk ke dalam pengukuran maka itu harus adanya indikator pengukuran dan perhitungan rumus dasar AHP sehingga nantinya akan mempermudah data diperhitungkan ke dalam aplikasi.

H. Perbandingan Berpasangan untuk Kriteria

Perbandingan berpasangan dilakukan berdasarkan aturan penilaian bobot untuk kriteria terlihat pada Tabel 3 di bawah ini:

Tabel 3. Kriteria dan Subkriteria Penilaian

Kriteria Penilaian	Subkriteria	Skala
Kepuasan Konsumen	Baik	3
	Cukup	2
	Kurang	1
Prestasi	Baik	3
	Cukup	2
	Kurang	1
Kinerja Operasional	Baik	3
	Cukup	2
	Kurang	1
Kedisiplinan	Baik	3
	Cukup	2
	Kurang	1
Kepribadian	Baik	3
	Cukup	2
	Kurang	1

Dimulai dengan melihat pada kriteria dan melakukan perbandingan antara kepuasan konsumen, kinerja operasional, prestasi, kedisiplinan, kepribadian dengan menggunakan paramater dan skala pada tabel bobot kriteria sesuai dengan tabel di atas.

Kriteria kepuasan konsumen dibandingkan dengan kriteria kinerja operasional. Kemudian kriteria kepuasan konsumen dibandingkan dengan prestasi, kriteria kepuasan konsumen dibandingkan dengan kriteria kedisiplinan, kriteria kepuasan konsumen dibandingkan dengan kriteria kepribadian.

Perbandingan antar kriteria menggunakan matriks berpasangan.

I. Penentuan Prioritas Kriteria

Langkah yang harus dilakukan dalam menentukan prioritas kriteria adalah sebagai berikut:

1. Membuat matriks perbandingan berpasangan. Pada tahap ini dilakukan penilaian perbandingan antara satu kriteria dengan kriteria yang lain. Hasil penilaian dapat dilihat dalam tabel 4 berikut ini:

Tabel 4. Matriks Perbandingan Berpasangan untuk Kriteria

Kriteria	Kepuasan Konsumen	Prestasi	Kinerja Operasional	Kedisiplinan	Kepribadian
Kepuasan Konsumen	1	2	3	5	7
Prestasi	0,5	1	3	3	5
Kinerja Operasional	0,33	0,67	1	3	3
Kedisiplinan	0,2	0,4	0,6	1	1
Kepribadian	0,14	0,29	0,43	0,72	1
Jumlah	2,17	4,36	8,03	12,72	17

- Membuat matriks nilai kriteria. Dari hasil matrik kriteria berpasangan yang akan tampil setelah melakukan input matrik kriteria dan hasil menu matrik berpasangan diperoleh dari perbandingan antara nilai-nilai kriteria yang dimasukkan dari proses menu matrik kriteria sebelumnya. Rancangannya dapat dilihat pada gambar di bawah ini. Penentuan prioritas untuk kriteria-kriteria tersebut dilakukan dalam bentuk matriks, sebagai berikut:

Tabel 5. Matriks Nilai Kriteria

Kriteria	Kepuasan Konsumen	Prestasi	Kinerja Operasional	Kedisiplinan	Kepribadian	jumlah	prioritas
Kepuasan Konsumen	0,460820493	0,458715596	0,42674253	0,42662116	0,38888889	2,16179767	0,43236
Prestasi	0,230414747	0,228357798	0,28446502	0,2559727	0,27777778	1,27801804	0,255604
Kinerja Operasional	0,152073733	0,153869725	0,14224751	0,17064846	0,16696967	0,7853061	0,157061
Kedisiplinan	0,062165899	0,091743119	0,08534851	0,08532423	0,11111111	0,46586287	0,093136
Kepribadian	0,084518129	0,086513761	0,08116943	0,06143345	0,05555556	0,30918532	0,081637

- Membuat matriks penjumlahan setiap baris. Jumlah tiap baris kriteria ini menjelaskan serta menampilkan hasil kriteria. dijelaskan bahwa, hasil penjumlahan diperoleh dari nilai prioritas dikalikan dengan nilai matrik pada tabel perbandingan berpasangan. Matrik penjumlahan setiap baris, dapat dilihat pada tabel sebagai berikut:

Tabel 6. Matrik Penjumlahan setiap Baris

Kriteria	Kepuasan Konsumen	Prestasi	Kinerja Operasional	Kedisiplinan	Kepribadian	Jumlah
Kepuasan Konsumen	0,432359534	0,511207216	0,47118366	0,46569287	0,43285945	2,31330273
Prestasi	0,216179767	0,255603608	0,31412244	0,27941572	0,30918532	1,37450866
Kinerja Operasional	0,142678646	0,171254417	0,15706122	0,18627715	0,18551119	0,84278262
Kedisiplinan	0,086471907	0,102241443	0,09423673	0,09313857	0,12367413	0,49976278
Kepribadian	0,060530335	0,074125046	0,06753632	0,06705977	0,06183706	0,33108854

- Penghitungan rasio konsistensi. Dari proses pencarian nilai CR (Rasio Konsistensi) untuk memastikan bahwa nilai CR (Rasio Konsistensi), perhitungan rasio konsistensi dapat dilihat pada tabel 7.

Tabel 7. Perhitungan Rasio Konsistensi

Kriteria	Jumlah per Baris	Prioritas	Hasil
Kepuasan Konsumen	2,3133027	0,43236	2,7456627
Prestasi	1,3745069	0,2556	1,6301069
Kinerja Operasional	0,8427826	0,15706	0,9998426
Kedisiplinan	0,4997628	0,09314	0,5929028
Kepribadian	0,3310885	0,06184	0,3929285

Dari tabel sebelumnya, diperoleh nilai-nilai sebagai berikut :

Jumlah :	6,3614435
n (jumlah kriteria)	5
Imaks (jumlah/n)	1,2722887
CI ((Imaks-n/n))	-0,74554226
CR(CI/IR(lihat tabel))	0,665662732

Oleh karena CR < 0,1 maka rasio konsistensi dari perhitungan tersebut bisa diterima dan dijadikan dasar perhitungan AHP pengukuran kinerja karyawan. Prioritas hasil perhitungan pada langkah 1 dan langkah 2 kemudian dituangkan ke dalam matrik hasil dapat terlihat pada tabel 8.

Tabel 8. Penghitungan Rasio Konsistensi Perhitungan Hasil

Kriteria	Jumlah per Baris	Prioritas	Hasil
Baik	1,737679	0,60325	2,340929
Cukup	0,823108	0,283107	1,106215
Kurang	0,327718	0,113643	0,441361

Dari tabel sebelumnya, diperoleh nilai-nilai sebagai berikut :

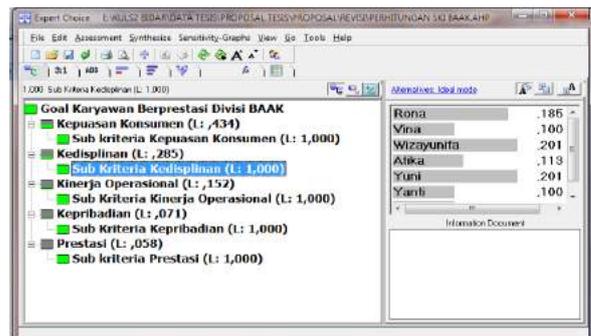
Jumlah :	3,888506
n (jumlah kriteria)	3
I maks (jumlah/n)	1,296169
CI ((Imaks-n/n-1))	-0,85192
CR(CI/IR(lihat tabel))	-1,46882

Oleh karena CR < 0,1 maka rasio konsistensi dari perhitungan tersebut bisa diterima.

Tabel 9. Matrik Hasil

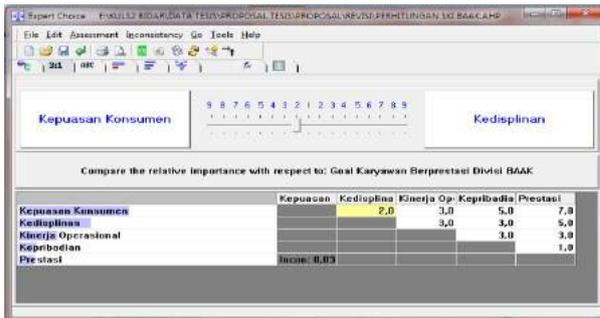
Kepuasan Konsumen	Prestasi	Kinerja Operasional	Kedisiplinan	Kepribadian
0,432359534	0,25560361	0,15706122	0,09313857	0,061837065
Baik	Baik	Baik	Baik	Baik
1	1	1	1	1
Cukup	Cukup	Cukup	Cukup	Cukup
0,410078965	0,44007583	0,44498343	0,44498343	0,469302099
Kurang	Kurang	Kurang	Kurang	Kurang
0,167215587	0,17787665	0,17976877	0,17976877	0,188384578

J. Perhitungan AHP Karyawan Berprestasi pada Divisi BAAK dengan menggunakan Aplikasi Expert Choice:



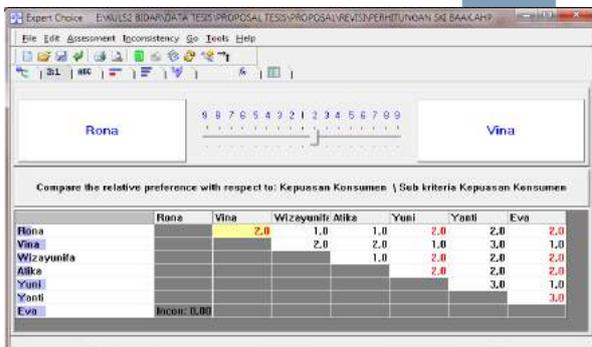
Gambar 5. Goal karyawan Berprestasi Divisi BAAK

Pada Gambar 5 di atas menjelaskan tampilan awal pengisian Goal,Kriteria dan sub kriteria dan pada kolom kanan diisikan data karyawan yang akan dinilai dengan penggunaan metode AHP.



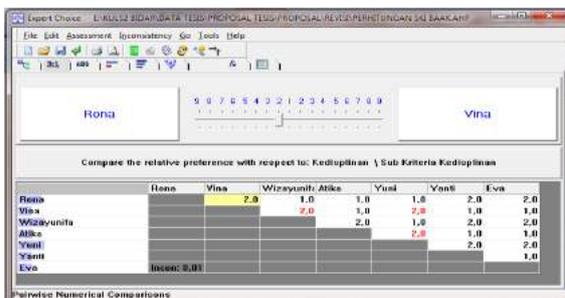
Gambar 6 Perbandingan Kriteria Penilaian Karyawan Berprestasi

Pada gambar 6 di atas menjelaskan perbandingan antara kriteria berdasarkan prioritas yang disesuaikan dengan penilaian SKI.



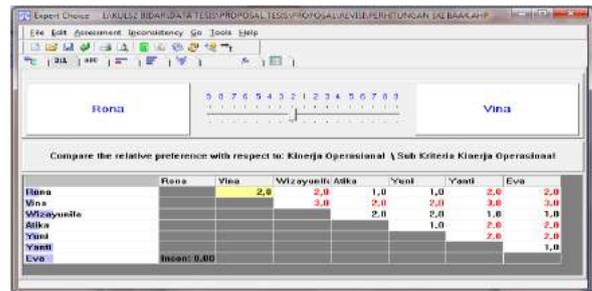
Gambar 7 Perbandingan Sub Kriteria Kepuasan Konsumen pada Pemilihan Karyawan Berprestasi.

Pada gambar 7 di atas menjelaskan data karyawan sesuai dengan bobot kriteria yang telah diisikan pada form penilaian SKI dan disesuaikan dengan matrik perbandingan Sub Kriteria Kepuasan Konsumen.



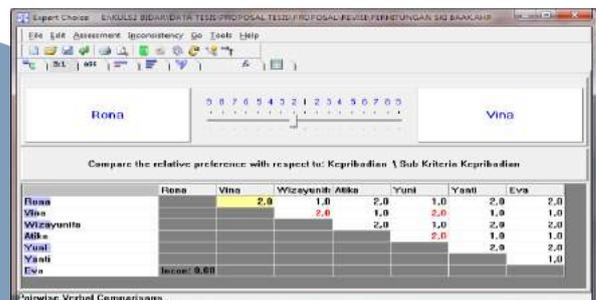
Gambar 8 Perbandingan Sub Kriteria Kedisiplinan pada Pemilihan Karyawan Berprestasi

Pada gambar 8 di atas menjelaskan data karyawan sesuai dengan bobot kriteria yang telah diisikan pada form penilaian SKI dan disesuaikan dengan matrik perbandingan Sub Kriteria Kedisiplinan.



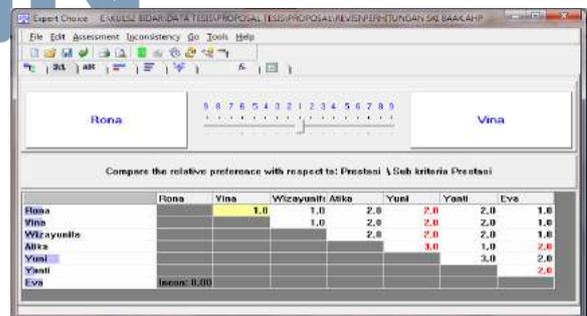
Gambar 9 Perbandingan Sub Kriteria Kinerja Operasional Pada Pemilihan Karyawan Berprestasi.

Pada gambar 9 di atas menjelaskan data karyawan sesuai dengan bobot kriteria yang telah diisikan pada form penilaian SKI dan disesuaikan dengan matrik perbandingan Sub Kriteria Kinerja Operasional.



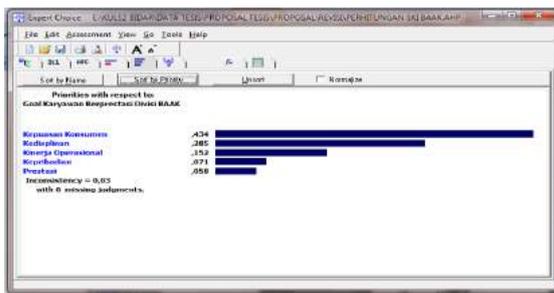
Gambar 10 Perbandingan Sub Kriteria Keprilaku pada Pemilihan Karyawan Berprestasi.

Pada gambar 10 di atas menjelaskan data karyawan sesuai dengan bobot kriteria yang telah diisikan pada form penilaian SKI dan disesuaikan dengan matrik perbandingan Sub Kriteria Keprilaku.



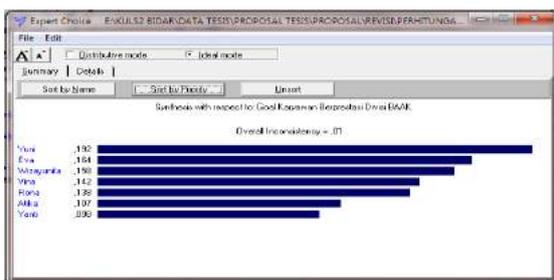
Gambar 11 Perbandingan Sub Kriteria Prestasi pada Pemilihan Karyawan Berprestasi.

Pada gambar 11 di atas menjelaskan data karyawan sesuai dengan bobot kriteria yang telah diisikan pada form penilaian SKI dan disesuaikan dengan matrik perbandingan Sub Kriteria Prestasi.



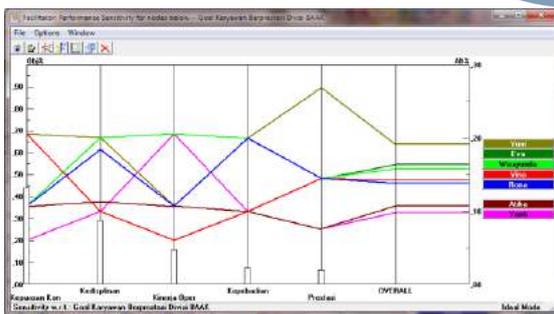
Gambar 12 Urutan Prioritas Sub Kriteria Dari Matrik Perbandingan

Gambar 12 di atas menjelaskan tingkatan prioritas dari sub kriteria yang ada dengan inkonsistensi 0,03.



Gambar 13 Hasil Penilaian Graphical Comparison

Gambar 13 di atas menjelaskan urutan penyeleksian karyawan berprestasi dengan menggunakan graphical comparison setelah data diinput.



Gambar 14 Tampilan grafik dengan menggunakan Sensitivity-Graphs

Gambar 14 di atas menjelaskan tampilan Sensitivity-Graphs dalam bentuk performance.

K. Pelaksanaan Pengujian Sistem Penilaian AHP

Hasil dari pengujian yaitu menggunakan aplikasi Expert Choice versi 11 sebagai penentuan indikator dan percobaan pelaksanaan penginputan data dan pencapaian hasil data karyawan yang diinput ke dalamnya dan digunakan pada dua Divisi yaitu Divisi UPT dan Divisi BAAK.

Sebagai pelaksanaan yang akan digunakan oleh Pimpinan Divisi untuk mempermudah penilaian karyawan yang diasuhnya dan untuk memberikan

reward atau punishment atau pun promosi kenaikan jabatan untuk perkembangan ke depannya.

V. SIMPULAN DAN SARAN

Berdasarkan hasil pembahasan dan analisis yang telah dilakukan serta sesuai dengan maksud dan tujuan penelitian, maka dapat diambil kesimpulan sebagai berikut:

Dari hasil perbandingan bobot kriteria yang telah diinput dan telah disesuaikan dengan matrik perbandingan Sub Kriteria Prestasi maka didapatkan yang menempati prioritas data tertinggi yaitu kepuasan konsumen dengan point 0,434, kedisiplinan dengan point 0,285, kinerja operasional dengan point 0,071 dan prestasi dengan point 0,058 dengan inkonsistensi 0,03 with 0 missing judgements.

Hasil perhitungan AHP di atas akan diterapkan untuk menghasilkan keluaran nilai intensitas prioritas karyawan tertinggi sehingga karyawan yang memiliki nilai tertinggi layak untuk mendapatkan reward atau penghargaan.

Adanya persiapan model sistem yang dapat dikembangkan dengan kesesuaian pada aplikasi yang telah dibuat sehingga dapat membantu perhitungan cepat dan akurat pada tingkat manajemen perguruan tinggi.

Berdasarkan hasil pembahasan, analisis serta kesimpulan, maka penulis menyampaikan saran-saran sebagai berikut:

Untuk membangun suatu sistem pemilihan karyawan berprestasi yang ideal membutuhkan banyak hal yang dianggap penting untuk mendukung kinerja sistem itu sendiri. Batasan-batasan yang diberikan terhadap sistem harus bisa bersifat fleksibel. Hal ini dimaksudkan untuk memperoleh suatu sistem yang benar-benar dapat dimanfaatkan, baik dalam content maupun fitur yang ada.

Adapun metode AHP yang telah dipaparkan oleh peneliti merupakan salah satu metode yang dalam melakukan suatu proses pemilihan dengan multikriteria, seperti pemilihan karyawan berprestasi pada tingkat perguruan tinggi.

UCAPAN TERIMA KASIH

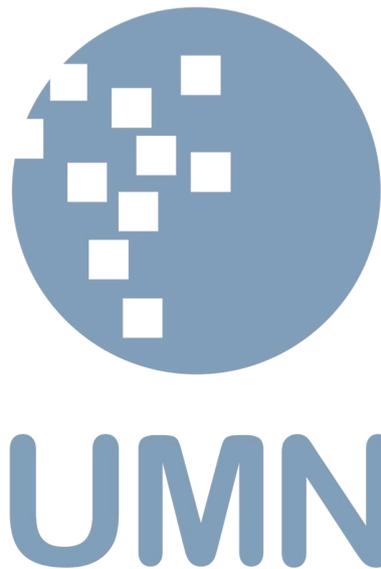
Terima kasih kepada Kampus Universitas Tanri Abeng yang telah banyak membantu dalam upaya proses penulisan penelitian dapat berjalan dengan lancar.

DAFTAR PUSTAKA

[1] Arki Rifazka dkk,2009, Penilaian performansi kerja account manager dengan menggunakan metode MBO (Management By Objective) dan rich (rank inclusion ini criteria hierarchies),SNATI 2009

[2] Drucker, Peter F, 2002. "Chapter 8:Management by Objectives and Self-Control". Martin Hinterseer, Zusammenfassung Kapitel 8.

- [3] Kusri, Awaluddin M, 2007, Sistem pendukung keputusan evaluasi kinerja karyawan untuk promosi jabatan.
- [4] Jogiyanto, HM, 2008, "Metodologi Penelitian Sistem Informasi: Pedoman dan Contoh Melakukan Penelitian di Bidang Sistem Teknologi Informasi", Yogyakarta: Andi.
- [5] Rakhmat Nugroho, 2006, Analisis faktor-faktor yang mempengaruhi kinerja karyawan (Studi Empiris pada PT. Bank tabungan negara (Persero), cabang Bandung)
- [6] Saaty, T.L., 1993, "Decision Making for Leader". The Analytical Hierarchy Process for Decision in Complex World, Prentice Hall Coy: Ltd, Pittsburgh.
- [7] Suryadi, K. dan Ramdhani, MA.1998. Sistem Pendukung Keputusan. PT Remaja Rosdakarya, Bandung.



Ekstraksi Kebutuhan Aplikasi Berdasarkan Feedback Pengguna Menggunakan Naïve Bayes dan Gamifikasi

Andre Rusli

Program Studi Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
andre.rusli@umn.ac.id

Diterima 30 Maret 2018

Disetujui 8 Juni 2018

Abstrak—Requirements engineering is a series of activities which aims to elicit, analyze, evaluate, and document the requirements of a system that is being developed. The activities do not stop after the product is deployed but continues as the users use the product and provide feedbacks to the system and matter how decent the functionalities of a product are, if it cannot address the correct problem and/or opportunities of the stakeholders or users, the product cannot be considered useful. That being said, not all stakeholders are willing to participate in providing useful feedbacks to improve the product after deployment, for many reasons. Gamification is considered as an opportunity that can be utilized to improve the motivation of user to use a product by implementing game design elements into an existing software product, thus increasing user participation to contribute in providing useful feedbacks and evolving requirements of a software product. This research proposes a model to support engineers in motivating users to provide feedbacks using gamification and also Naïve Bayes Classifier to classify user feedbacks into categories needed by the developer to extract the requirements stated in the feedback, such as bug reports, feature request, user experiences, etc.

Kata Kunci—requirements engineering, gamification, Naïve Bayes, user feedback

I. PENDAHULUAN

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 [1]. Tahapan analisis kebutuhan atau yang disebut juga dengan *requirements engineering* (RE) menjadi salah satu tahapan yang paling penting dalam *software development life cycle* (SDLC), karena hasil dari proses analisis kebutuhan tersebutlah yang akan menjadi pondasi dan landasan dari proyek yang akan dijalankan.

Setelah kebutuhan didefinisikan di awal, kebutuhan terhadap sistem dapat terus berubah dan berkembang seiring berjalannya waktu dan perkembangan teknologi. Setelah *system-to-be* selesai dibangun, muncul lagi kebutuhan untuk merancang dan membangun *system-to-be-next*. Objektif suatu sistem, struktur konseptual, kebutuhan dan asumsi yang telah didapatkan, dievaluasi, dispesifikasi, dan dianalisis sebelumnya mungkin perlu diubah dan/atau dikembangkan lebih lanjut karena berbagai alasan. Beberapa alasan tersebut mencakup *error* atau *bug* yang ditemukan pada sistem, perubahan prioritas dan *constraints* pada sistem, keunggulan dan keterbatasan sistem, dan banyak hal lainnya. Hal tersebut yang dinamakan dengan *requirements evolution* [1] atau evolusi kebutuhan.

Perubahan-perubahan tersebut dapat terjadi baik saat analisis kebutuhan dilakukan di awal, saat kebutuhan dievaluasi, saat perancangan maupun pembangunan *software*, atau bahkan setelah sistem dirilis ke publik dalam bentuk *user feedback*. *Feedback* pengguna, seperti yang dapat ditemukan pada pasar aplikasi seperti Google Play Store, dapat memiliki banyak informasi terkait *requirements* seperti *bugs* pada aplikasi [2], permintaan pengembangan fitur [3], dan juga ide-ide baru terkait fitur tertentu pada aplikasi atau bahkan fitur baru [4].

Partisipasi pengguna atau yang sering disebut dengan *stakeholder* dalam suatu proyek pengembangan piranti lunak sangatlah penting. Keterlibatan pengguna dalam menganalisis kebutuhan terbukti dapat meningkatkan penerimaan pengguna [5], mengurangi resiko kegagalan proyek [6], dan juga meningkatkan pemahaman pengguna terhadap sistem [7]. Adapula survei yang menunjukkan bahwa partisipasi pengguna dalam suatu proyek pengembangan *software* merupakan faktor paling penting yang dapat menentukan keberhasilan maupun kegagalan suatu proyek teknologi informasi [8]. Namun kenyataannya, tidak semua penggunanya melibatkan diri secara aktif dalam proses analisis kebutuhan suatu proyek karena berbagai alasan, termasuk dalam memberikan *feedback* terkait pengalaman yang dirasakan pengguna setelah

menggunakan suatu aplikasi atau *software*. Hal ini terutama terjadi pada proyek yang melibatkan banyak pengguna/*stakeholder* yang bervariasi, seperti proyek-proyek yang biasa dilakukan oleh *software producing organization* (SPO) dengan *multiple clients*, mencocokkan kebutuhan para pengguna dengan visi dan *road map* suatu produk semakin menambah kesulitan yang ada [9].

Konsep gamifikasi tumbuh dan berkembang sebagai suatu tren yang menjanjikan dalam berbagai area. Penelitian yang dilakukan M2 Research pada tahun 2011 memprediksi bahwa pada pasar gamifikasi akan mencapai 2.8 miliar dolar Amerika pada tahun 2016 [10]. Penggunaan elemen-elemen *game design* pada aplikasi-aplikasi maupun proses dalam cakupan *non-game* dianggap dapat meningkatkan partisipasi pengguna dan juga pada akhirnya meningkatkan kualitas suatu produk [11]. Kontribusi pertama dari penelitian ini yaitu adalah untuk mengajukan ide bahwa implementasi konsep gamifikasi dapat meningkatkan partisipasi pengguna dalam memberikan *feedback* terhadap suatu aplikasi yang dapat menjadi sumber informasi terkait munculnya *requirements* baru atau kesalahan-kesalahan yang terjadi pada analisis kebutuhan yang telah dilakukan sebelumnya.

Namun sekedar mendapatkan *feedback* pengguna saja tidak langsung dapat menyelesaikan semua masalah, karena walaupun motivasinya masih harus ditingkatkan, tapi jumlah *feedback* dari pengguna ini bisa datang dalam jumlah yang cukup besar dan untuk mengklasifikasi lalu mengekstrak *requirements* dari sana secara manual dapat memakan waktu. Selain itu, tidak semua *feedback* dari pengguna dapat dinilai berguna dan berkualitas bagi pengembangan proyek seperti *spam* atau informasi yang sama sekali tidak ada kaitannya dengan aplikasi. Salah satu algoritma *machine learning* yang dapat digunakan untuk membantu proses klasifikasi *user feedback* adalah Naïve Bayes.

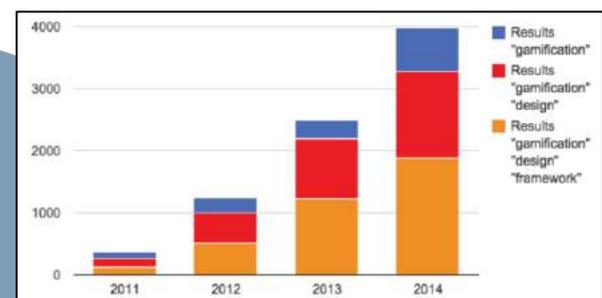
Dalam penelitian sebelumnya [12], telah dilakukan klasifikasi *review* aplikasi yang terdapat di Google Play dan Apple Stores, namun belum ada kajian terkait penggunaan gamifikasi untuk meningkatkan motivasi pengguna untuk memberikan *feedback* yang berkualitas. Berdasarkan hasil dari penelitian tersebut pula, Naïve Bayes merupakan algoritma klasifikasi yang tepat karena dapat mencapai tingkat akurasi yang tinggi dengan *training set* yang relatif kecil serta memiliki kecepatan yang tinggi dibandingkan algoritma lain yang dikaji pada penelitiannya yaitu Maximum Entropy dan Decision Tree Learning [12]. Adapun kontribusi berikutnya dari penelitian ini adalah untuk mengintegrasikan *automatic app reviews classifier* menggunakan Naïve Bayes ke dalam model yang diajukan.

Selanjutnya, karya ilmiah ini terstruktur sebagai berikut. Bab II akan menjelaskan tentang gamifikasi

beserta beberapa literatur terkait. Bab III menjelaskan mengenai klasifikasi *feedback* pengguna menggunakan Naïve Bayes, bab IV menjabarkan model yang diajukan pada penelitian ini, dan bab V menyajikan kesimpulan dari penelitian yang sedang dijalankan serta tahapan-tahapan berikutnya yang dapat dilakukan untuk mengembangkan penelitian ini.

II. GAMIFIKASI

Gamifikasi merupakan metode yang belakangan ini mendapatkan banyak sorotan di dunia teknologi. Gambar 1 di bawah menunjukkan hasil pencarian terhadap penelitian akademis seputar gamifikasi beserta perkembangannya dari tahun 2011 sampai 2014. Dapat dilihat bahwa jumlah penelitian akademis mengenai “gamifikasi” beserta “design” dan “framework” terkait gamifikasi meningkat dengan cukup pesat dari tahun ke tahun.



Gambar 1. Hasil Pencarian Tentang Gamifikasi pada Penelitian Akademis [13]

Gamifikasi didefinisikan sebagai penggunaan elemen-elemen pada *game design* pada konteks non-game [14]. Walau begitu, tujuan penggunaan elemen-elemen *design* dalam gamifikasi agak berbeda dengan *game* pada umumnya. Pada gamifikasi, penggunaan elemen-elemen *game design* lebih ditujukan untuk meningkatkan *user engagement* pada konteks sistem yang sudah ada. Sedangkan, pada *game*, penggunaan elemen-elemen umumnya ditujukan murni untuk *entertainment*. Oleh karena itu, perancangan sistem yang digamifikasi tidak bisa disamakan dengan proyek pembuatan *game*.

Dalam merancang dan membangun suatu sistem yang tergamifikasi, terdapat beberapa *framework* yang dapat digunakan sebagai acuan dalam membuat suatu sistem yang digamifikasi. Pada penelitian yang dilakukan Mora dkk. [13], setelah mengkaji 18 *framework* yang ada, ditemukan bahwa ada 10 *game design items* pada gamifikasi yang dianggap “*the ten most meaningful (in terms of results and heterogeneity)*”. Adapun sepuluh hal tersebut adalah sebagai berikut.

1. *Viability*: studi pendahuluan, evaluasi dan analisis potensi untuk mengaplikasikan gamifikasi pada suatu sistem.
2. *Stakeholders*: teknik yang digunakan untuk mengidentifikasi dan berfokus pada pihak-

- pihak yang berinteraksi pada proses perancangan.
3. *Loop*: mekanika game yang digabung dengan *reinforcement* dan *feedback* yang digunakan untuk menarik pengguna ke dalam *key system actions*.
 4. *Endgame*: akhir permainan yang sudah ditentukan di awal, biasanya menarik pengguna untuk menggunakan seluruh kemampuannya.
 5. *On-boarding*: cara memulai pemakaian sistem bagi partisipan baru.
 6. *Rules*: aturan main yang ditentukan oleh *system designer*.
 7. *Metrics*: standar pengukuran efisiensi, performa, progres, proses, ataupun kualitas.
 8. *Ethics*: cabang filosofi yang melibatkan mensistematisasi, mempertahankan, dan merekomendasikan konsep benar dan salah.
 9. *UI/UX*: segala hal yang dirancang ke dalam sistem yang digamifikasi, termasuk interaksi, perilaku, sikap, dan emosi pemain.
 10. *Technology*: komponen *software* yang diperlukan dan digunakan dalam pembangunan sistem.

Dari 18 *framework* gamifikasi yang dikaji, terdapat dua *framework* yang di dalamnya membahas kesepuluh *item* yang disebutkan diatas, baik secara implisit maupun eksplisit, sedangkan pada 16 *framework* lainnya tidak semua *item* tersebut dibahas. Salah satunya adalah *framework* yang diajukan oleh Werbach dan Hunter, yang dinamakan Six Steps to Gamification [15]. Adapun keenam langkah tersebut adalah sebagai berikut.

1. *DEFINE business objectives*. Salah satu tujuan dari gamifikasi adalah untuk menciptakan sesuatu yang disukai orang namun itu bukanlah tujuan satu-satunya dari gamifikasi. *Business objective* yang ingin dikaji harus didefinisikan dengan jelas. Bila pengguna menyukai sistem yang dikembangkan namun tidak meningkatkan produktivitas, efisiensi, maupun kesejahteraan di lingkungan kerja, maka hasil implementasi gamifikasi dapat melenceng dari tujuan awalnya.
2. *DELINEATE target behavior*. Langkah berikutnya adalah jabarkan secara konkrit perilaku atau *behavior* apa yang diinginkan dari target pengguna sistem untuk mencapai tujuan awal atau *business objective* yang dikaji.
3. *DESCRIBE your players*. Werbach pada *interview* dengan pihak dari Engaging Leader [16], menyebutkan bahwa penting untuk menganggap pengguna sebagai *players/pemain*, bukan sekedar pengguna atau

karyawan. Pemain ini yang menggunakan sistem yang dibuat secara sukarela, dan perlu merasakan kesenangan dan hasrat untuk berkembang dalam sistem yang dibuat. Ada banyak tipe pemain yang dapat dimotivasi oleh berbagai hal, seperti kompetisi atau tantangan, dan ada juga yang termotivasi karena sosialisasi yang dapat dilakukan pada sistem. Penting untuk mengetahui tipe pemain untuk merancang sistem dan menggunakan elemen-elemen yang tepat untuk meningkatkan partisipasi dan motivasi pemain pada sistem yang digamifikasi.

4. *DEVISE activity cycles*. Merancang bagaimana sistem yang digamifikasi bekerja. Dibagi menjadi dua level yaitu mikro dan makro. Pada level mikro, harus dipikirkan bagaimana hal-hal sederhana seperti *feedback* dapat memotivasi pengguna untuk terus bermain. Sedangkan pada level makro, lebih difokuskan pada yang disebut dengan "Player Journey", yang adalah perjalanan pemain dari awal sebagai pemula yang lalu terus berkembang dan tetap termotivasi untuk bermain sampai ke tahap dimana pemain dapat dianggap sukses.
5. *DON'T forget the fun!* Langkah selanjutnya adalah dimana *developer* harus berhenti memikirkan hal teknis sejenak, agar tidak melupakan *fun* dalam permainan. *System designer* harus memikirkan pertanyaan-pertanyaan seperti "All right, would people actually find this fun and engaging? If not, what can I do to make it feel more fun". Namun tetap harus dilakukan tanpa melupakan struktur, hal teknis, dan tujuan awal sistem digamifikasikan.
6. *DEPLOY the appropriate tools*. Pada tahap ini, cari dan tentukan tools yang dibutuhkan untuk merancang dan membangun sistem yang digamifikasi. Baik itu software tools maupun mekanisme-mekanisme sederhana dalam game. Tentukan elemen-elemen game (*dynamics, mechanics, dan components*) yang ingin digunakan, lalu bangun sistemnya.

Dalam membangun sistem yang digamifikasi, keenam langkah tersebut tidak hanya cukup dilakukan dalam satu iterasi dan sistem selesai dibangun. *Developer* harus memperhatikan *feedback* yang didapat dari pengguna dan memperbaiki sistem yang ada sesuai kebutuhan pengguna atau menambahkan fitur-fitur yang dirasa perlu.

III. KLASIFIKASI FEEDBACK PENGGUNA DENGAN NAÏVE BAYES

Pada penelitiannya, Maalej dan Nabil [12] mengimplementasikan tiga algoritma klasifikasi *machine learning* yaitu Naïve Bayes, Maximum Entropy, dan Decision Tree untuk mengklasifikasi *user review* dari suatu aplikasi yang dapat ditemukan di

Google Play Store atau Apple App Store. *User reviews* yang didapat kemudian diklasifikasikan ke dalam empat kategori yaitu *bug reports*, *feature request*, *user experiences*, dan *ratings* [12].

Bug reports menjelaskan masalah dengan aplikasi terkait yang harus diperbaiki, seperti *crash*, perilaku sistem yang tidak diinginkan, atau isu-isu terkait performa aplikasi. Pada kategori *feature request*, pengguna menanyakan fungsionalitas atau konten yang dianggap kurang atau belum ada atau yang dianggap baik dan patut ditambahkan pada *update* berikutnya. *User experiences* menggambarkan pengalaman yang dirasakan pengguna ketika menggunakan aplikasi dan fitur-fiturnya pada kondisi-kondisi tertentu. Dan yang terakhir adalah *ratings*, yang merupakan representasi dari jumlah bintang yang diterima aplikasi.

Dari hasil penelitian yang dilakukan, disimpulkan untuk mengklasifikasi *user review* ke dalam empat kategori di atas, dapat dikatakan bahwa Naïve Bayes merupakan algoritma yang lebih tepat dibanding algoritma *classifier* lainnya karena dapat mencapai tingkat akurasi yang tinggi dengan ukuran *training set* yang kecil, dengan waktu dua kali lebih cepat dibanding algoritma lainnya [12]. Adapun klasifikasi dengan Naïve Bayes dilakukan setelah data *user review* diperoleh, dikategorikan berdasarkan beberapa *keyword* yang sudah dipilih sebelumnya, dan kalimat-kalimatnya diproses dengan menggunakan teknik *natural language processing (NLP)* seperti *stopword removal*, *stemming*, *lemmatization*, *tense detection*, dan *bigrams*, yang berada di luar cakupan penelitian ini.

Setelah kalimat diproses menjadi *bag of words*, dengan algoritma Naïve Bayes Classifier, setiap fitur dipertimbangkan dalam menentukan kelas/*label* apa saja yang dapat diberikan terhadap suatu input. Untuk menentukan *label* yang diberikan, Naïve Bayes menghitung probabilitas awal dari setiap *label*, yang bisa didapat dari pengecekan frekuensi kemunculan *label* tersebut pada *training set* [17].

Naïve Bayes merupakan algoritma yang sangat populer sebagai *binary classifier* [17], yang dapat menentukan apakah suatu input termasuk ke dalam satu kategori atau tidak. Dalam klasifikasi ke dalam lima kategori sebagai contoh, suatu *user feedback* dapat tergolong ke dalam lebih dari satu kategori, dan pada penelitian ini, suatu input yang berupa *feedback* dapat diklasifikasi secara biner (termasuk dalam satu kategori atau tidak) sebanyak lima kali. Model yang digunakan pada penelitian ini adalah *multinomial document model*, dimana tidak seperti *Bernoulli document model*, frekuensi kemunculan suatu kata dalam suatu dokumen dianggap penting [18]. Selain itu, pada dokumen dengan ukuran *vocabulary* yang besar tingkat kesalahan dapat dikurangi hingga 27% dengan menggunakan *multinomial model* jika dibandingkan dengan *multi-variate Bernoulli model* [19].

Hal pertama yang harus dilakukan untuk mengklasifikasikan suatu dokumen dengan Naïve Bayes Classifier adalah membuat suatu *training set* yang akan dijadikan acuan dalam menentukan probabilitas dokumen baru yang masuk nantinya. Setelah *training set* tersedia, maka dapat dilakukan perhitungan *probability* dengan *multinomial text classification model* sebagai berikut [18].

- Tentukan dan kumpulkan daftar seluruh kata yang ada dan dipertimbangkan (V)
- Hitung hal-hal dibawah ini:
 - N : jumlah total seluruh dokumen (*feedback* dari pengguna) yang ada
 - N_k : jumlah dokumen yang diklasifikasikan ke dalam suatu kategori k , untuk setiap kategori yang ada
 - x_{it} : frekuensi kemunculan suatu kata w_i dalam suatu dokumen D_i , untuk setiap kata yang ada dalam V – dengan kata lain, hitung $n_k(w_i)$: frekuensi kemunculan kata w_i dalam dokumen dengan kategori k
- Hitung probabilitas kemunculan suatu kata (w_i) dalam semua dokumen yang termasuk dalam suatu kategori (C_k) dengan juga menerapkan *Laplace's law of succession* untuk menghindari *zero probability problem* [18], menggunakan rumus (1) atau (2) di bawah ini, dimana z_{ik} merupakan variabel indikator yang bernilai 1 ketika suatu dokumen termasuk dalam satu kategori, dan bernilai 0 ketika tidak.

$$P(w_t|C_k) = \frac{1 + \sum_{i=1}^N x_{it} z_{ik}}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^N x_{is} z_{ik}} \quad (1)$$

$$P(w_t|C_k) = \frac{1 + n_k(w_t)}{|V| + \sum_{s=1}^{|V|} n_k(w_s)} \quad (2)$$

- Hitung probabilitas terklasifikasinya suatu dokumen ke dalam suatu kategori $P(C_k)$ dengan menggunakan rumus (3) di bawah ini.

$$P(C_k) = \frac{N_k}{N} \quad (3)$$

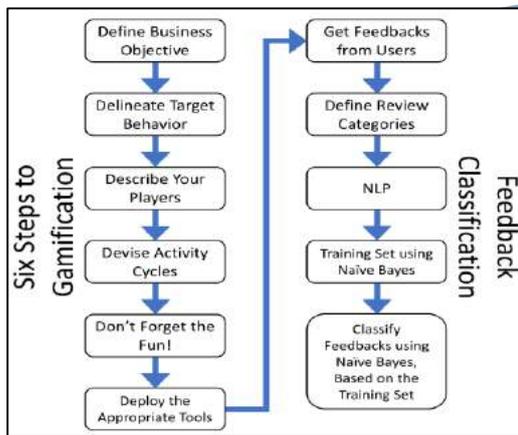
Langkah berikutnya adalah untuk menerima dokumen baru yang belum diklasifikasikan (D), lalu akan diklasifikasikan ke masing-masing kategori yang telah ditentukan berikutnya. Berikut rumus (4) yang menampilkan formula untuk menghitung probabilitas suatu dokumen/*feedback* untuk diklasifikasikan ke dalam suatu kategori. Dimana x adalah frekuensi kemunculan suatu kata dalam dokumen D .

$$P(C_k|D) = \frac{P(C_k|x)}{\alpha P(x|C_k)P(C_k)} \quad (4)$$

$$\alpha P(C_k) \prod_{j=1}^{|V|} P(w_t|C_k)^{x_t}$$

IV. MODEL UNTUK EKSTRAKSI KEBUTUHAN APLIKASI DENGAN GAMIFIKASI DAN NAÏVE BAYES

Berdasarkan penelitian-penelitian yang telah disebutkan sebelumnya, dapat dilihat bahwa *requirements evolution* merupakan suatu hal yang nyata dan terjadi pada suatu proyek pengembangan sistem, dan salah satu masukan yang berguna bagi *developer* adalah *feedback* yang didapat dari penggunaannya. Penelitian ini memperkenalkan suatu model yang dianggap mampu meningkatkan motivasi pengguna untuk memberikan *feedback* yang berguna bagi *developer* dan juga cara untuk mengklasifikasikan *feedback* tersebut sehingga dapat mempermudah *developer* untuk menyimpulkan fitur-fitur mana saja yang harus dikembangkan, *bugs and errors* yang harus diperbaiki, dan lain sebagainya.



Gambar 2. Model Ekstraksi Kebutuhan Aplikasi

Model yang diajukan dibagi menjadi dua bagian secara garis besar seperti yang dapat dilihat pada Gambar 2. Bagian pertama adalah implementasi gamifikasi untuk meningkatkan partisipasi pengguna untuk memberikan *feedback* yang berguna untuk pengembangan produk ke depannya, yang menerapkan Six Steps to Gamification, *framework* gamifikasi yang diajukan oleh Werbach dan Hunter [15]. Lalu pada bagian selanjutnya, setelah *feedback* didapatkan, maka *developer* harus menentukan kategori-kategori *feedback* sesuai dengan kebutuhannya, lalu *feedback* yang didapat diproses dengan NLP, yang lalu bila belum ada *training set* yang dapat dijadikan acuan bagi Naïve Bayes untuk menentukan kategori *feedback*, maka *training set* tersebut harus dibuat terlebih dahulu, dan akhirnya *developer* dapat menggunakan Naïve Bayes untuk mengklasifikasikan *feedback* yang datang berikutnya.

Dalam mengimplementasikan gamifikasi sesuai dengan model diatas, berikut penjelasan keenam panduan berdasarkan Six Steps to Gamification [15].

1. Define Business Objective

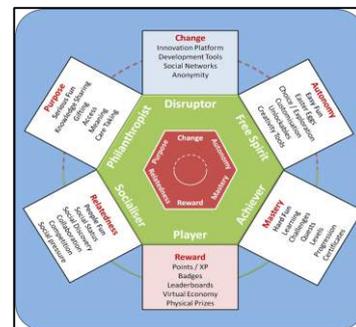
Pada tahap ini, pihak pengembang aplikasi harus mendefinisikan secara jelas apa objektif yang ingin dicapai dengan model ini. Secara umum, objektif dari gamifikasi pada model yang diajukan adalah untuk mendapatkan informasi terkait evolusi kebutuhan berdasarkan masukan dari pengguna yang didapat dari *feedback*. Maka elemen-elemen gamifikasi yang digunakan pun harus searah agar dapat memotivasi pengguna untuk turut serta berpartisipasi dalam memberikan *feedback* yang berguna demi mendukung agar tujuan tersebut dapat tercapai.

2. Delineate Target Behavior

Untuk mencapai objektif yang telah disebutkan di atas yaitu untuk mendapatkan *feedback* yang berguna sebagai masukan terkait kebutuhan aplikasi yang harus dikembangkan, pengguna diharapkan untuk dapat secara aktif berpartisipasi memberikan *feedback* yang berarti, sesuai kebutuhan *developer*. Panduan terkait syarat-syarat agar suatu *feedback* menjadi berarti dapat diinformasikan kepada pengguna, penggunaan elemen-elemen gamifikasi yang adapun harus dijelaskan sehingga perilaku pengguna dapat disesuaikan dengan tujuan diimplementasikannya gamifikasi untuk ekstraksi kebutuhan.

3. Describe Your Players

Tidak semua pengguna memiliki karakteristik yang sama. Sesuai dengan tipenya, elemen-elemen gamifikasi yang baik digunakan pada sistem bisa berbeda. Gambar 3 menampilkan tipe-tipe pengguna yang harus dipertimbangkan dalam membuat suatu sistem yang digamifikasi (*gamified system*). Menurut Marczewski [20], secara umum pengguna dapat dikategorikan ke dalam 6 tipe yaitu Philanthropist, Socialiser, Player, Achiever, Free Spirit, dan Disruptor.



Gambar 3. Tipe-Tipe Pengguna [20]

Philanthropist termotivasi oleh tujuan dan arti dari aktivitas yang dijalankannya. Pengguna dengan tipe ini ingin membantu orang lain, tidak egois, dan ingin dapat menjadi bermanfaat untuk sesuatu tanpa mengharapkan imbal balik. **Socialiser** termotivasi oleh keterhubungan pengguna dengan pengguna lainnya, suka berinteraksi dan membuka koneksi social yang baru. **Player** dapat dimotivasi dengan *reward* atau

hadiah. Mereka secara sukarela turut berpartisipasi dalam permainan karena tertarik dengan *reward* yang bisa didapatkan dari suatu aktivitas.

Achiever dapat termotivasi dengan keahlian dari suatu aktivitas. Tipe ini sangat tertarik dengan hal baru dan aktivitas yang dapat dilakukan untuk mengembangkan diri mereka dan bahkan menjadi ahli di suatu bidang. **Free Spirit** suka mencoba sesuatu yang baru dan bereksplorasi, tipe ini dapat dimotivasi dengan adanya otonomi dan kebebasan mengekspresikan diri sendiri. Yang terakhir adalah **Disruptor**, tipe ini dapat dimotivasi dengan perubahan dan kemampuan mereka untuk menjatuhkan sistem yang ada baik secara positif maupun negatif [20].

Setiap tipe memiliki karakteristiknya masing-masing yang bisa disesuaikan dengan kebutuhan sistem dan elemen-elemen gamifikasi yang digunakan untuk meningkatkan motivasi pengguna untuk menggunakan sistem. Seperti contohnya bagi pemain dengan tipe Player, elemen gamifikasi seperti *points* dan *leaderboard* dapat digunakan untuk menarik pemain untuk berpartisipasi dalam sistem karena sifat Player yang menginginkan *reward* atas usaha yang telah dilakukannya. Berbeda dengan Socialiser, contohnya, yang mungkin dapat tertarik dengan fitur yang memungkinkan mereka untuk dapat bersosialisasi dalam sistem dan berkolaborasi dengan pengguna lainnya.

Marczewski menyajikan hasil dari *user type test* yang telah dilakukan terhadap lebih dari 10.000 orang menunjukkan bahwa Philanthropist dan Free Spirit merupakan 2 tipe yang paling populer dengan 20% dan 19% dari total pengguna yang telah melakukan *test* tersebut [21]. Namun, hasil yang disajikan tersebut tidak dapat digunakan langsung pada proyek yang ingin dilakukan karena pada akhirnya, setiap proyek bisa memiliki tipe-tipe pengguna yang berbeda dan *developer* harus mengetahui tipe-tipe calon pengguna dari sistem yang dibangun.

4. Devise Activity Cycles

Pada suatu permainan, biasanya selalu memiliki awal dan kemungkinan juga memiliki akhir. Namun, di antara kedua hal tersebut, terdapat banyak kondisi yang berulang dan bercabang. Ada dua jenis *cycles* atau *loops* yang harus dipertimbangkan dalam gamifikasi, yaitu *engagement loops (micro level)* dan *progression stairs (macro level)* [15].

Pada level mikro, aksi yang dilakukan pengguna berangkat dari motivasi awal untuk berpartisipasi dalam sistem, lalu harus dilanjutkan dengan *feedback* yang berupa respon dari sistem. *Feedback* (dari sistem) merupakan cara efektif untuk memotivasi pengguna untuk terus menggunakan sistem, yang dalam hal ini adalah untuk memberikan masukan terkait kebutuhan-kebutuhan yang berubah atau bertambah. *Feedback* sistem dapat bermacam-macam bentuknya, bahkan *rewards* seperti poin dan *achievements* pun merupakan *feedback* dari sistem atas aksi yang dilakukan pengguna.

Sedangkan pada level makro, sistem harus dirancang dan dibangun dengan memikirkan progres yang dapat dirasakan pengguna ketika sistem digunakan. Salah satu contoh elemen yang dapat digunakan untuk menunjukkan progres pengguna adalah *level* dan *difficulty*. Semakin lama pengguna berpartisipasi, maka *levelnya* semakin tinggi dan semakin sulit dan menantang pula hal yang harus dilakukan.

5. Don't Forget the Fun

Hal yang tidak boleh dilupakan ketika merancang suatu sistem dengan gamifikasi adalah untuk berhenti sejenak dan berpikir, apakah sistem yang dirancang itu menyenangkan? Banyak pertanyaan-pertanyaan yang dapat ditanyakan oleh *developer* saat ada di tahap ini. Salah satu pertanyaan yang dapat dipikirkan adalah, bila tidak ada *rewards* secara ekstrinsik yang didapatkan pengguna, bila pengguna tidak diminta untuk menggunakan sistem, apakah pengguna akan ingin menggunakan sistem hanya karena sistem tersebut menarik dan dirasa menyenangkan? Bila tidak, maka ada hal-hal yang harus diperbaiki untuk meningkatkan rasa *fun* tersebut.

6. Deploy the Appropriate Tools

Tahap terakhir dari bagian pertama model yang diajukan pada penelitian ini adalah untuk menentukan mekanika dan komponen yang akan digunakan pada sistem yang digamifikasi, dan implementasi sesuai dengan perancangan yang telah dibuat sebelumnya.

Setelah keenam langkah di atas dilakukan, sistem yang telah digamifikasi selesai dirancang dan dibangun, dan elemen-elemen yang digunakan pun telah terimplementasi ke dalam sistem. Setelah sistem yang telah digamifikasi ini dirilis ke publik dan pengguna secara aktif berpartisipasi untuk memberikan *feedback* terkait produk yang dikaji untuk mengetahui kebutuhan-kebutuhan yang berevolusi atau yang masih kurang pada produk yang dibuat sebelumnya, saatnya beranjak ke bagian kedua dari model yang diajukan, yaitu mengklasifikasikan *feedback* tersebut.

7. Get Feedback from Users

Feedback yang didapat dari pengguna lalu disimpan dan diproses – bisa menggunakan NLP – untuk mengekstrak kata-kata yang memang relevan dan dapat digunakan. Output dari tahap ini adalah *bag of words*, atau kumpulan kata-kata dari setiap *feedback* dari pengguna yang nantinya akan digunakan untuk membuat training set dan masukan yang akan diklasifikasi.

8. Define Review Categories

Untuk mengklasifikasikan *feedback* ke dalam beberapa kategori, *developer* harus menentukan terlebih dahulu kategori tersebut. Pada penelitian yang dilakukan sebelumnya, *review* pengguna yang didapatkan dari pasar aplikasi diklasifikasikan ke dalam empat kategori yaitu *bug reports*, *feature request*, *user experiences*, dan *ratings* [12]. Namun

tentu saja kategori-kategori tersebut dapat berbeda sesuai kebutuhan *developer*.

9. Define the Training Sets

Pada tahap ini, *feedback* dari pengguna sudah didapatkan dan diproses dan kategori yang diinginkan pun sudah ditentukan. Untuk membentuk *training set*, *developer* secara manual memilah-milah suatu set dokumen dan lalu mengklasifikasikannya ke dalam kategori-kategori yang ada.

10. Classify the Feedbacks

Setelah *training set* dibentuk dan probabilitas-probabilitas yang dibutuhkan sudah dihitung, maka langkah selanjutnya adalah untuk melakukan perhitungan probabilitas suatu dokumen baru termasuk ke dalam satu atau lebih kategori yang telah ditentukan sebelumnya, dengan menggunakan formula yang tercatat di rumus (4).

V. SIMPULAN

Pada penelitian ini, diajukan sebuah model yang dapat digunakan untuk meningkatkan motivasi pengguna agar turut berpartisipasi secara aktif dalam memberikan masukan berupa *user feedback* yang berguna sebagai *requirements* yang dapat digunakan *developer* untuk mengembangkan suatu produk lebih lanjut ke depannya dengan menggunakan *framework* gamifikasi yang dinamakan Six Steps to Gamification yang dipublikasi oleh Werbach dan Hunter [15]. Selain itu, model pada penelitian ini juga mengimplementasikan Naïve Bayes dengan *multinomial document model* sebagai *text classifier* yang dapat membantu *developer* dalam mengklasifikasikan *feedback* dari pengguna yang banyak jumlahnya.

Dengan klasifikasi yang dibantu menggunakan Naïve Bayes Classifier, *developer* diharapkan dapat menghemat waktu dan bekerja lebih efisien dalam mengkaji kebutuhan-kebutuhan suatu produk yang berubah, berkembang, atau bertambah, dan juga dapat membantu proses *requirements prioritization* berdasarkan jumlah dan urgensi yang dapat diketahui berdasarkan *feedback* pengguna.

Langkah selanjutnya yang dapat dilakukan adalah untuk mengimplementasikan model yang telah diajukan dan merancang bangun suatu *feedback system* untuk memotivasi pengguna dan mengklasifikasikan *feedback* yang didapat sesuai dengan kategori yang ditentukan sebelumnya oleh *developer*. Setelah itu, model dapat diuji dan dievaluasi keefektifannya. Selain itu, penelitian dari segi natural language processing pun dapat digali lebih dalam untuk menentukan proses-proses yang lebih baik dalam mengolah *feedback* dari pengguna.

VI. DAFTAR PUSTAKA

- [1] A. v. Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specification*, Chichester: John Wiley & Sons, 2009.
- [2] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," Rio de Janeiro, 2013.
- [3] C. Jacob and R. Harrison, "Retrieving and analyzing mobile apps feature requests from online reviews," San Francisco, 2013.
- [4] L. V. G. Carreno and K. Winbladh, "Analysis of user comments: An approach for software requirements evolution," San Francisco, 2013.
- [5] S. Kujala, "User Involvement: A Review of the Benefits and Challenges," 2003.
- [6] K. E. Emam, S. Quintin and N. H. Madhavji, "User participation in the requirements engineering process: An empirical study," 1996.
- [7] L. Damodaran, "User involvement in the systems design process-a practical guide for users," 1996.
- [8] The Standish Group, "CHAOS Summary 2009: The 10 Laws of CHAOS," The Standish Group International, Incorporated, 2009.
- [9] G. Lucassen, F. Dalpiaz, J. M. van der Werf and S. Brikkemper, "Bridging the Twin Peaks: the Case of the Software Industry," Florence, 2015.
- [10] S. Peterson, "gamesindustry.biz," 2012. [Media perantara]. Alamat situs: <https://www.gamesindustry.biz/articles/2012-05-21-gamification-market-to-reach-usd2-8-billion-in-2016>. [Diakses 23 March 2018].
- [11] C. Eickhoff, C. G. Harris, A. P. de Vries and P. Srinivasan, "Quality through flow and immersion: gamifying crowdsourced relevance assessments," Portland, 2012.
- [12] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? On automatically classifying app reviews," Ottawa, 2015.
- [13] A. Mora, D. Riera, C. Gonzales and J. Arnedo-Moreno, "A Literature Review of Gamification Design Frameworks," Skovde, 2015.
- [14] S. Deterding, D. Dixon, R. Khaled and L. E. Nacke, "Gamification: Toward a Definition," Vancouver, 2011.
- [15] K. Werbach and D. Hunter, *For the Win: How Game Thinking Can Revolutionize Your Business*, Wharton Digital Press, 2012.
- [16] K. Werbach, "Engaging Leader," 2013. [Media perantara]. Alamat situs: <http://www.engagingleader.com/6-steps-to-effective-gamification-transcript/>. [Diakses 26 March 2018].
- [17] S. Bird, E. Klein and E. Loper, *Natural Language Processing with Python*, First Edition ed., Sebastopol: O'Reilly Media Inc., 2009.
- [18] H. Shimodaira, "Text Classification using Naive Bayes," Edinburgh, 2018.
- [19] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI-98 workshop on learning for text categorization*, hal. 41-48, 1998.
- [20] A. Marczewski, "Gamified UK," 2015. [Media perantara]. Alamat situs: <https://www.gamified.uk/user-types/>. [Diakses 26 March 2018].
- [21] A. Marczewski, "Gamified UK," 2016. [Media perantara]. Alamat situs: <https://gamified.uk/UserTypeTest2016/user-type-test-results.php?lid=#.Wrs2kWaB3Bl>. [Diakses 26 March 2018]

Situsparu: Sistem Pakar Untuk Deteksi Penyakit Tuberkulosis Paru

Ricky Surya¹, Dennis Gunawan²

Jurusan Informatika Fakultas Teknik dan Informatika Universitas Multimedia Nusantara, Tangerang, Indonesia
ricky010196@gmail.com
dennis.gunawan@umn.ac.id

Diterima 1 April 2018

Disetujui 8 Juni 2018

Abstract—Tuberculosis is an infectious disease caused by mycobacterium tuberculosis. It can affect some parts of the body: lungs, lymph nodes, intestines, kidneys, endometrium, bones, and brain. According to the survey of tuberculosis prevalence conducted by Republic of Indonesia Ministry of Health in 2013-2014, Indonesia was the second country in the world with the most case of tuberculosis. It makes Indonesia become a country with emergency in lungs tuberculosis. An expert system for lungs tuberculosis detection is built to help people detecting the possibility of suffering from lungs tuberculosis. Therefore, it is hoped that the lungs tuberculosis patient can have early treatment. Certainty factor is used to solve the uncertainty problem delivered by the doctor when examining the patient. Thus, certainty factor is an appropriate method to be used in the expert system for detecting certain disease. This method has been correctly implemented, proved by comparing system detection result to manual calculation result. The expert system has 81.25% accuracy, 83.49% success using DeLone and McLean model, and a cronbach alpha of 0.82 which indicates a good reliability based on the indicators used in the questionnaire.

Index Terms— Certainty Factor, Disease Detection, Expert System, Pulmonary Tuberculosis, Situsparu.

I. PENDAHULUAN

Tuberkulosis merupakan penyakit yang diakibatkan oleh infeksi kuman Mikobakterium tuberkulosis yang dapat menyerang organ-organ tubuh seperti paru-paru, kelenjar getah bening, usus, ginjal, kandungan, tulang sampai otak. Sebagian besar penyakit ini mengenai organ paru-paru dibandingkan organ lainnya. Tuberkulosis paru atau yang biasa dikenal dengan sebutan TBC sangat mudah menular melalui udara, yaitu ketika seseorang penderita TBC batuk atau bersin dan dihirup oleh orang-orang di sekitarnya [1].

Seseorang yang memiliki daya tahan tubuh tinggi dan gizi yang baik, akan susah untuk tertular penyakit TBC. Sedangkan, pada orang yang mengalami kekurangan gizi dan daya tahan tubuh buruk atau sering menghirup udara yang mengandung kuman tuberkulosis akibat lingkungan yang buruk, akan lebih mudah terinfeksi TBC [1].

Berdasarkan WHO Global Tuberculosis Control tahun 2010 menyatakan bahwa dari data Badan Kesehatan Dunia (WHO) diketahui jumlah penderita Tuberkulosis tahun 2007 di Indonesia mencapai 528.000 dan menempati urutan ke-3 sedunia dalam hal jumlah penderita Tuberkulosis (TB). Baru pada tahun 2009 menurun menjadi posisi kelima dunia dengan jumlah penderita TB sebesar 429.000 orang [2].

Berdasarkan Survei Pravelensi TB oleh Badan Litbangkes Kemenkes RI Tahun 2013-2014, dengan perkiraan jumlah penduduk Indonesia 250.000.000, setiap tahun terdapat 1.000.000 kasus TB paru baru dengan angka kematian sebesar 100.000 orang per tahun atau 273 orang per harinya. Dengan hasil survei tersebut, Indonesia berada di peringkat kedua dengan kasus TB terbanyak di dunia setelah India dan menjadikan Indonesia sebagai negara dengan kondisi darurat TB paru [3].

Dalam mendiagnosis suatu penyakit, pakar sering kali mengeluarkan ungkapan seperti “mungkin”, “kemungkinan besar”, “hampir pasti”. Metode certainty factor digunakan untuk mengakomodasi ketidakpastian tersebut sehingga metode ini cocok digunakan pada sistem pakar yang mengukur sesuatu yang pasti atau tidak pasti seperti mendiagnosis penyakit [4]. Metode certainty factor pernah digunakan untuk membuat sistem pakar pendeteksi resiko osteoporosis dan osteoarthritis, dimana tingkat keakuratannya mencapai 80% [5] dan pembuatan sistem pakar untuk menentukan klasifikasi american society of anesthesiologists physical status dengan tingkat akurasi sebesar 71,9% [6]. Terdapat penelitian sistem pakar mengenai penyakit tuberkulosis dengan judul Sistem Pakar Pendiagnosa Penyakit Tuberkulosis, dimana sistem pakar tersebut menggunakan metode inferensi forward chaining dan teori certainty factor yang digunakan untuk melakukan deteksi terhadap tuberkulosis paru, kelenjar getah bening, payudara, dan tulang belakang [7]. Selain hasil deteksi penyakit tuberkulosis, sistem pakar tersebut menyediakan informasi penanganan terhadap penyakit yang diderita.

Perkembangan teknologi yang berkembang pesat telah membuat orang lebih banyak menggunakan internet dalam mengumpulkan atau mencari informasi,

termasuk dalam mencari informasi mengenai penyakit. Hal tersebut terbukti dari survei yang dilakukan sepanjang 2016 oleh Asosiasi Penyelenggara Jaringan Internet Indonesia (APJII), 132.700.000 dari 256.200.000 orang Indonesia telah terhubung dengan Internet [8]. Hal ini menunjukkan bahwa sekitar 51,8 persen penduduk Indonesia telah menggunakan internet dalam aktivitas sehari-harinya. Berdasarkan data tersebut, sistem pakar ini akan dirancang berbasis website dengan harapan masyarakat dapat dengan mudah untuk melakukan deteksi penyakit tuberkulosis paru.

Oleh karena itu, dibutuhkan sebuah sistem pakar untuk mendeteksi penyakit tuberkulosis paru menggunakan metode certainty factor berbasis web. Pada penelitian ini, akan dilakukan uji coba kelayakan sistem dengan menyebarkan kuesioner yang dibuat berdasarkan model Delone dan Mclean. Kemudian, akan dilakukan uji reliabilitas kuesioner menggunakan cronbach alpha.

II. LANDASAN TEORI

A. Certainty Factor

Untuk mendapatkan tingkat keyakinan dari sebuah rule, digunakan Rumus (1) [4].

$$CF(h,e) = MB(h,e) - MD(h,e) \quad (1)$$

Keterangan:

CF(h,e) : faktor kepastian

MB(h,e) : tingkat keyakinan terhadap hipotesis (h), jika diberikan evidence (e) antara 0 dan 1

MD(h,e) : tingkat ketidak yakinan terhadap hipotesis (h), jika diberikan evidence (e) antara 0 dan 1

Menurut Grosan dan Abraham, terdapat kombinasi CF ketika lebih dari satu evidence menghasilkan CF untuk fakta yang sama, yaitu sebagai berikut [6].

1. Jika $CF(e1)$ dan $CF(e2) > 0$

$$CF(H, e1 \wedge e2) = CF(e1) + CF(e2) * (1 - CF(e1)) \quad (2)$$

2. Jika $CF(e1)$ dan $CF(e2) < 0$

$$CF(H, e1 \wedge e2) = CF(e1) + CF(e2) * (1 + CF(e1)) \quad (3)$$

3. Jika tanda $CF(e1) \neq$ tanda $CF(e2)$

$$CF(H, e1 \wedge e2) = (CF(e1) + CF(e2)) / (1 - \min(|CF(e1)|, |CF(e2)|)) \quad (4)$$

Tabel 1. Representasi Nilai CF [9]

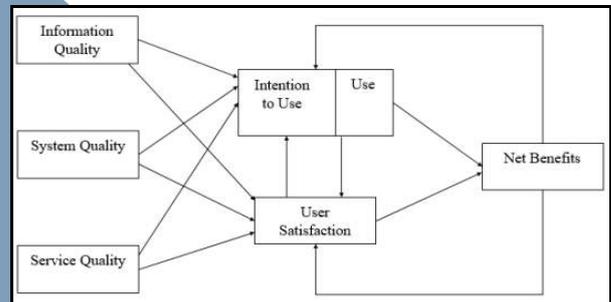
Uncertain Term	CF
Pasti tidak	-1.0
Hampir pasti tidak	-0.8
Kemungkinan besar tidak	-0.6
Mungkin tidak	-0.4
Tidak tahu	-0.2 sampai 0.2

Uncertain Term	CF
Mungkin	0.4
Kemungkinan besar	0.6
Hampir pasti	0.8
Pasti	1.0

Tabel representasi nilai CF merupakan tabel acuan untuk mendapatkan nilai CF suatu gejala sesuai dengan apa yang pakar katakan terhadap gejala tersebut.

B. Model DeLone dan McLean

Model DeLone dan McLean dapat digunakan untuk melakukan uji coba kelayakan sistem yang dibangun. Model DeLone dan McLean menyebutkan bahwa information quality, system quality, dan service quality akan berpengaruh positif pada use dan user satisfaction dan selanjutnya akan berpengaruh pada net benefit [11]. Gambar 1 menunjukkan model DeLone dan McLean.



Gambar 1. Model DeLone dan McLean [12]

Model DeLone dan McLean mendefinisikan enam variabel berikut untuk menentukan kesuksesan sistem [12].

1. System Quality. System Quality mendefinisikan karakteristik yang diinginkan dari sistem, seperti kemudahan menggunakan sistem, kehandalan sistem, kemudahan pembelajaran, dan waktu respons.
2. Information Quality. Information Quality mendefinisikan kualitas keluaran dari sistem, seperti kelengkapan, mudah dimengerti, akurasi, dan konsistensi.
3. Service Quality. Service Quality mendefinisikan dukungan yang diberikan oleh sistem, seperti responsivitas dan keandalan dari sistem.
4. Use. Use mendefinisikan kegunaan dari sistem.
5. User Satisfaction. User Satisfaction mendefinisikan kepuasan pengguna terhadap sistem.

6. Net Benefits. Net Benefits mendefinisikan dampak sistem bagi individu, kelompok, organisasi, dan industri.

C. Cronbach Alpha

Cronbach alpha digunakan untuk mengukur keandalan indikator-indikator yang digunakan dalam kuesioner penelitian [13]. Cronbach alpha merupakan sebuah ukuran keandalan yang memiliki nilai berkisar nol sampai satu [14]. Nilai tingkat keandalan cronbach alpha dapat dilihat pada Tabel 2.

Tabel 2. Tingkat Keandalan Cronbach Alpha

Nilai Cronbach Alpha	Tingkat Keandalan
$r \geq 0.8$	Sangat andal
$0.8 > r \geq 0.6$	Andal
$0.6 > r \geq 0.4$	Cukup andal
$0.4 > r \geq 0.2$	Agak andal
$r < 0.2$	Kurang andal

Untuk menghitung koefisien reliabilitas instrumen dengan menggunakan cronbach alpha, digunakan Rumus (5).

$$r = \left(\frac{k}{k-1} \right) \left(1 - \frac{\sum \sigma_b^2}{\sigma^2} \right) \quad (5)$$

Keterangan:

- r : koefisien reliabilitas instrumen (cronbach alpha)
 k : jumlah butir pertanyaan
 $\sum \sigma_b^2$: total butir varians
 σ^2 : total varians

III. PERANCANGAN SISTEM

A. Pengetahuan Pakar

Dalam mendiagnosis apakah seseorang terkena TBC atau tidak, dokter akan melakukan pengecekan fisik maupun medis agar mendapatkan hasil yang akurat terkait penyakit TBC [1]. Terdapat beberapa gejala yang mengindikasikan seseorang terkena penyakit TBC menurut dr. Bambang Irawan, Sp.P., yaitu sebagai berikut.

1. Periode batuk
2. Batuk berdahak dan mengeluarkan darah
3. Dada terasa nyeri ketika bernafas atau batuk
4. Berkeringat pada malam hari
5. Berat badan turun
6. Nafsu makan menurun

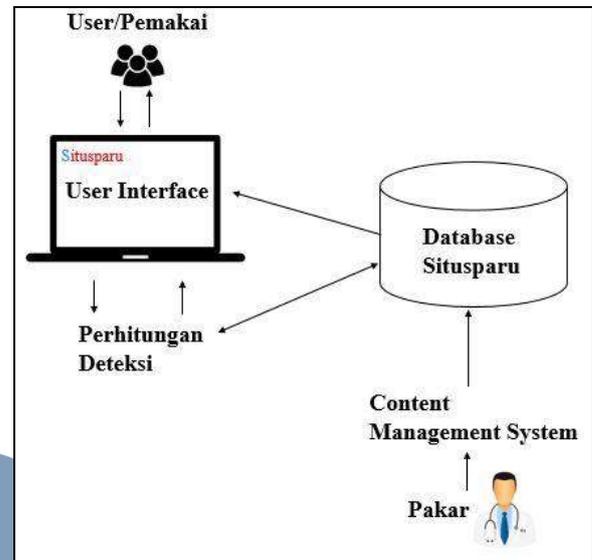
Terdapat beberapa faktor yang meningkatkan resiko seseorang terkena penyakit TBC, yaitu sebagai berikut.

1. Tinggal di lingkungan yang kumuh
2. Merokok atau Minum alkohol

3. Terkena HIV atau Diabetes.

B. Model Sistem Pakar Situsparu

Berikut merupakan model atau komponen sistem pakar pada Situsparu.



Gambar 2. Model Sistem Pakar Situsparu

1. Database Situsparu. Pada Situsparu, pertanyaan-pertanyaan gejala TB paru yang digunakan untuk melakukan deteksi, disimpan di dalam database.
2. Perhitungan Deteksi. Dalam proses deteksi, dilakukan perhitungan kemungkinan seseorang terkena TB paru menggunakan metode certainty factor berdasarkan pertanyaan gejala yang telah dijawab.
3. Antarmuka Pengguna (User Interface). Situsparu menampilkan pertanyaan gejala satu persatu, sehingga seseorang yang ingin melakukan deteksi harus menjawab pertanyaan pertama untuk menjawab pertanyaan selanjutnya, hingga pertanyaan terakhir agar mendapatkan hasil deteksi sesuai dengan jawaban yang telah dipilih.
4. Content Management System. Selain dapat digunakan user untuk melakukan deteksi, Situsparu juga memiliki admin untuk melakukan pengelolaan data dari sistem pakar ini sehingga user mendapatkan data yang terbaru.
5. Pakar. Pertanyaan-pertanyaan yang digunakan untuk melakukan deteksi TB paru didapatkan dari dr. Bambang Irawan, Sp.P.

C. Bobot Gejala

Bobot gejala merupakan bobot dari setiap gejala yang dijadikan pertanyaan untuk melakukan deteksi tuberkulosis paru. Setelah seseorang menjawab

pertanyaan-pertanyaan yang merupakan gejala tuberkulosis paru, sistem akan menghitung bobot gejala dari jawaban yang dipilih menggunakan metode certainty factor dan hasil akhirnya sistem akan menampilkan kemungkinan seseorang terkena tuberkulosis paru dari skala 0-100%. Gejala-gejala tuberkulosis paru dan bobot gejala didapat dari dr. Bambang Irawan, Sp.P, dokter spesialis paru di Rumah Sakit Pluit. Bobot setiap gejala dapat dilihat pada Tabel 3.

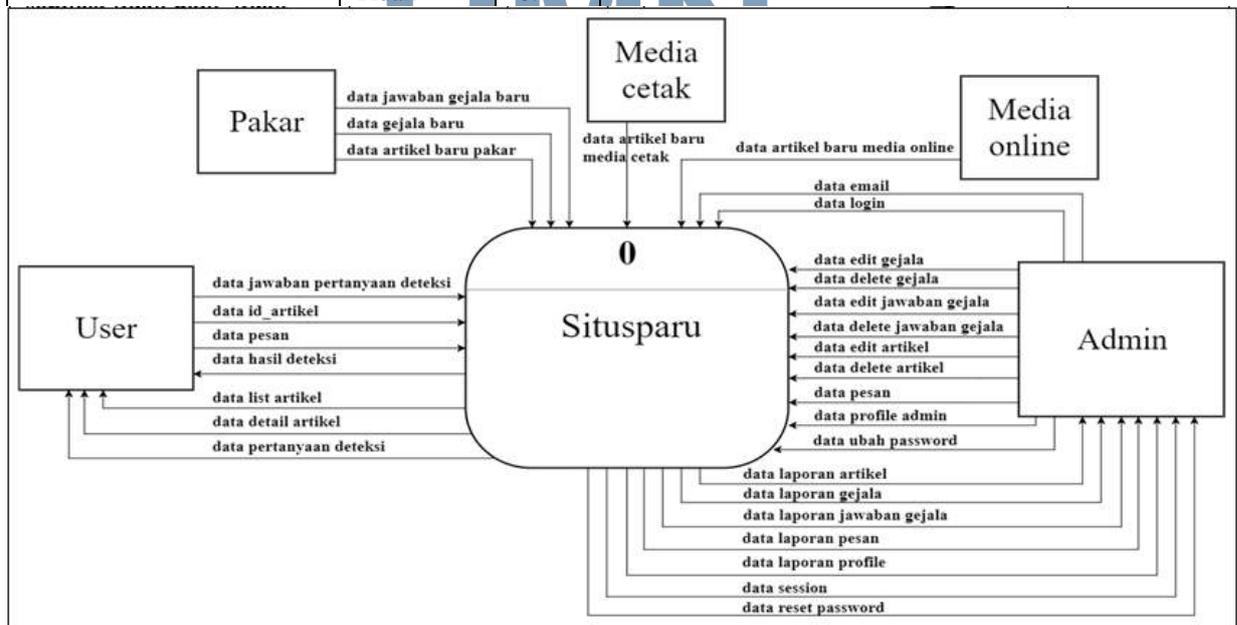
Tabel 3. Bobot Gejala Tuberkulosis Paru

Gejala	Jawaban Gejala	Bobot Gejala
Sudah berapa lama Anda mengalami batuk?	> 2 minggu	0.6
	<= 2 minggu	-0.2
	Tidak batuk	-0.8
Apakah Anda batuk berdarah (berwarna kuning/hijau) dan berdarah?	Ya	0.6
	Hanya berdarah	0.4
	Hanya berdarah	0.4
	Tidak keduanya	-0.4
Apakah dada Anda terasa nyeri ketika bernafas atau batuk?	Ya	0.2
	Tidak	-0.4
Apakah Anda mengalami keringat di punggung pada malam hari tanpa ada aktivitas?	Ya	0.4
	Tidak	-0.6
Apakah nafsu makan Anda menurun?	Ya	0.4
	Tidak	-0.4

Gejala	Jawaban Gejala	Bobot Gejala
Apakah berat badan Anda menurun secara signifikan?	Ya	0.4
	Tidak	-0.4
Apakah sekeliling tempat tinggal Anda kumuh (kotor, sanitasi tidak baik, tidak terkena sinar matahari)?	Ya	0.4
	Tidak	-0.6
Apakah Anda merokok atau meminum minuman beralkohol?	Ya	0.2
	Tidak	-0.2
Apakah Anda menderita diabetes atau HIV?	Ya	0.4
	Tidak tahu	0
	Tidak	-0.4

D. Data Flow Diagram

Data Flow Diagram digunakan untuk mengetahui aliran data dari suatu sistem yang dibuat. Berikut Data Flow Diagram dari sistem pakar deteksi tuberkulosis paru. Gambar 3 merupakan context diagram yang menggambarkan aliran data yang terjadi di sistem pakar deteksi tuberkulosis paru. Terdapat lima entitas, yaitu User yang ingin menggunakan sistem pakar ini, Pakar yang memberikan data mengenai gejala dan jawaban gejala tuberkulosis paru serta data artikel tentang tuberkulosis paru, Media cetak dan Media online yang memberikan data artikel tentang tuberkulosis paru, serta Admin yang mengelola data-data terkait dengan sistem pakar deteksi tuberkulosis paru ini. Proses Situsparu menerima tiga data dari user, tiga data dari pakar, satu data dari media cetak, satu data dari media online, dan sebelas data dari admin. User menerima empat data dari proses Situsparu, sedangkan admin menerima tujuh data dari proses Situsparu.

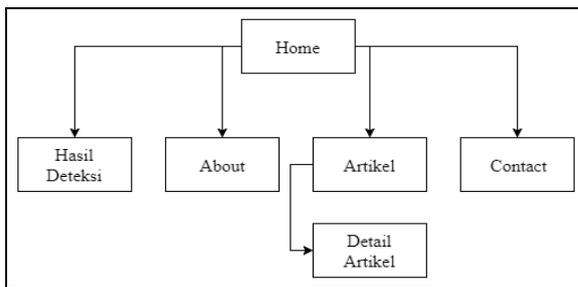


Gambar 3. Context Diagram

E. Sitemap

Gambar 4 merupakan sitemap dari pengguna Situsparu. Dari halaman Home, yang berguna untuk menampilkan form deteksi tuberkulosis paru, terdapat satu halaman yaitu halaman Hasil Deteksi yang berguna untuk menampilkan rangkuman jawaban pengguna serta hasil perhitungan dari jawaban gejala yang telah dijawab oleh pengguna. Pada halaman About, terdapat penjelasan mengenai sistem pakar deteksi tuberkulosis paru.

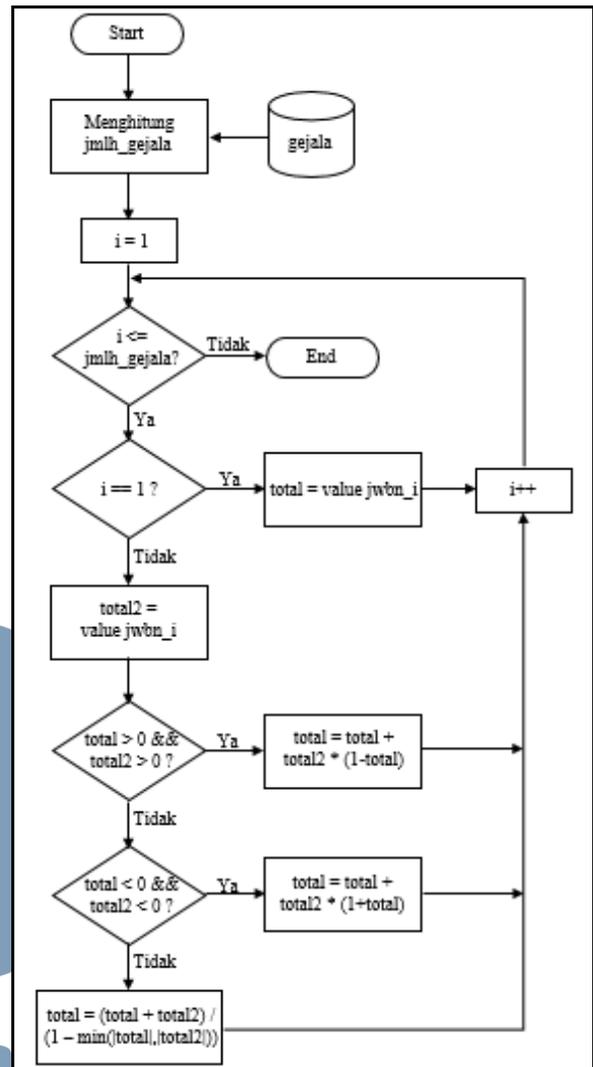
Dari halaman Artikel, yang berguna untuk menampilkan daftar artikel tuberkulosis paru, terdapat satu halaman yaitu halaman Detail Artikel yang berguna untuk menampilkan data detail suatu artikel yang dipilih. Pada halaman Contact, terdapat form yang berguna untuk menyampaikan keluhan atau pertanyaan.



Gambar 4. Sitemap User Situsparu

F. Flowchart

Gambar 5 merupakan alur proses Menghitung Bobot Gejala Menggunakan Metode Certainty Factor. Sebelum melakukan proses perhitungan, dilakukan pengambilan jumlah data gejala di database untuk dilakukan looping sebanyak jumlah gejala. Ketika looping pertama, value dari bobot gejala yang dipilih oleh user ditampung di variabel total. Lalu pada looping berikutnya, value dari bobot gejala yang dipilih oleh user ditampung di variabel total2. Kemudian dilakukan pengecekan terhadap variabel total dan total2 untuk menentukan rumus yang akan digunakan.



Gambar 5. Flowchart Menghitung Bobot Gejala

IV. IMPLEMENTASI DAN UJI COBA

A. Implementasi

Gambar 6 merupakan tampilan halaman Home bagi user. Di halaman ini, terdapat form untuk melakukan deteksi tuberkulosis paru. Ketika user telah selesai menjawab pertanyaan pertama, maka user harus menekan tombol Next sehingga akan berganti menjadi pertanyaan selanjutnya dan sampai pertanyaan terakhir untuk menekan tombol Finish.



Gambar 6. Tampilan Halaman Home

Gambar 7 merupakan tampilan halaman Hasil Deteksi ketika user selesai menjawab semua pertanyaan deteksi. User dapat melihat rangkuman jawaban yang telah dipilih sebelumnya, dan terdapat hasil deteksi tuberkulosis paru. Jika user ingin kembali melakukan deteksi, dapat menekan tombol Back to Home.



Gambar 7. Tampilan Halaman Hasil Deteksi.

B. Uji Coba Sistem Pakar

Uji coba sistem pakar dilakukan dengan membandingkan hasil deteksi sistem pakar yang telah dibangun dengan hasil deteksi pakar untuk mendapatkan tingkat akurasi dari sistem pakar yang telah dibangun. Setelah dilakukan perbandingan terhadap 32 kasus yang pernah ditangani oleh pakar, didapatkan 26 kasus sesuai dengan analisis pakar, dan 6 kasus tidak sesuai atau salah. Hasil analisis pakar berdasarkan gejala dan faktor resiko yang terdapat pada kasus tersebut tanpa menggunakan hasil uji laboratorium dan rontgen dada. Setelah melakukan uji coba, didapatkan tingkat akurasi Situsparu sebesar 81,25%.

C. Uji Coba Kelayakan Sistem

Uji coba kelayakan sistem dilakukan dengan menyebarkan kuesioner kepada 32 orang. Kuesioner dibuat berdasarkan model DeLone dan McLean, dan menghasilkan 9 pertanyaan yang mengacu pada 6 variabel model DeLone dan McLean serta 1 pertanyaan kritik atau saran. Pertanyaan tersebut juga dibuat berdasarkan referensi dari kuesioner yang dibuat oleh [10]. Susunan kuesioner yang dibuat dapat dilihat pada Tabel 4.

Tabel 4. Butir Pertanyaan Kuesioner.

<i>System Quality</i>	1. Situsparu mudah untuk digunakan dan <i>user friendly</i>
	2. Situsparu memenuhi kebutuhan untuk melakukan deteksi Tuberkulosis Paru
<i>Information Quality</i>	3. Informasi mengenai Tuberkulosis Paru yang terdapat pada Situsparu lengkap dan akurat
	4. Hasil deteksi dari Situsparu mudah untuk dimengerti
<i>Service Quality</i>	5. Situsparu menyediakan panduan yang jelas untuk menggunakan sistem
	6. Situsparu mampu menangani kesalahan pengguna ketika menggunakan sistem
<i>User Satisfaction</i>	7. Situsparu menyediakan hasil deteksi yang memuaskan
<i>Use</i>	8. Situsparu menghemat waktu dalam mendeteksi atau mencari artikel mengenai Tuberkulosis Paru
<i>Net Benefits</i>	9. Situsparu membantu dalam melakukan deteksi Tuberkulosis Paru
	10. Kritik dan saran bagi Situsparu

Tabel 5 Menunjukkan hasil perhitungan persentase skor dari variabel yang digunakan pada kuesioner

Variabel Pertanyaan Kuesioner	Persentase Skor
<i>System Quality</i>	85%
<i>Information Quality</i>	83.125%
<i>Service Quality</i>	82.1875%
<i>User Satisfaction</i>	80%
<i>Use</i>	87.5%
<i>Net Benefits</i>	83.125%

Setelah dilakukan perhitungan persentase skor dari setiap variabel yang digunakan pada kuesioner, dilakukan perhitungan persentase skor akhir dengan menghitung rata-rata persentase skor dari setiap variabel untuk mendapatkan persentase kesuksesan sistem.

Dengan persentase skor akhir sebesar 83.49%, dapat disimpulkan bahwa para responden sangat setuju bahwa Situsparu merupakan sistem pakar untuk deteksi Tuberkulosis Paru yang sukses berdasarkan enam variabel yang telah ditentukan.

D. Uji Reliabilitas Kuesioner

Uji Reliabilitas atau keandalan dilakukan untuk mengukur apakah hasil kuesioner dapat dipercaya berdasarkan indikator-indikator yang digunakan dalam kuesioner penelitian. Uji reliabilitas dilakukan dengan menggunakan cronbach alpha.

Dengan hasil perhitungan nilai cronbach alpha sebesar 0.82, dapat disimpulkan bahwa kuesioner sangat andal atau dipercaya berdasarkan indikator-indikator yang digunakan dalam kuesioner penelitian.

V. SIMPULAN

Sistem pakar untuk deteksi penyakit tuberkulosis paru menggunakan metode certainty factor berbasis web telah berhasil dirancang dan dibangun menggunakan framework CodeIgniter. Deteksi dilakukan berdasarkan gejala dan faktor resiko penyebab tuberkulosis paru yang dijawab oleh user. Uji coba perhitungan metode certainty factor yang dilakukan dengan membandingkan hasil deteksi sistem pakar dengan hasil perhitungan manual menghasilkan hasil yang sama. Sistem pakar yang dibangun mempunyai tingkat akurasi sebesar 81.25% berdasarkan hasil validasi yang dilakukan oleh dr. Bambang Irawan, Sp.P. Selain itu, dilakukan uji coba kelayakan sistem dengan menyebarkan kuesioner ke 32 orang dan didapatkan hasil sebesar 83.49% yang menunjukkan tingkat kelayakan sistem yang telah dibangun. Kemudian, dilakukan perhitungan uji reliabilitas menggunakan cronbach alpha dan mendapatkan hasil sebesar 0.82 yang menunjukkan kuesioner sangat andal berdasarkan indikator-indikator yang digunakan dalam kuesioner penelitian.

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan masukan untuk pengembangan sistem dan penelitian selanjutnya, yaitu sebagai berikut.

1. Sistem pakar ini dapat dibangun dengan berbasis mobile baik android maupun iOS.
2. Dapat ditambahkan daftar rumah sakit atau dokter spesialis paru yang dekat dengan lokasi pengguna.

DAFTAR PUSTAKA

- [1] Anies. 2016. Ensiklopedia Penyakit. PT Kanisius, Yogyakarta.
- [2] Wibowo, A. 2014. Kesehatan Masyarakat Di Indonesia. PT RajaGrafindo Persada, Jakarta.
- [3] Reksoprodjo, M. 2016. Indonesia Darurat Tuberkulosis [online]. Tersedia dalam: <http://www.stoptbindonesia.org/2016/04/darurat-tuberkulosis.html> [diakses 1 Mei 2017].
- [4] Turban, E., & Aronson, J.E. 2001. Decision Support System and Intelligent System, 6th Edition. Prentice Hall International Edition, New Jersey.
- [5] Halim, S. 2014. Rancang Bangun Sistem Pakar Pendeteksi Resiko Osteoporosis dan Osteoarthritis dengan Metode Certainty Factor Berbasis Android.
- [6] Ramadhan, F. 2016. Rancang Bangun Sistem Pakar Untuk Menentukan Klasifikasi American Society Of Anesthesiologists Physical Status Dengan Metode Certainty Factor Berbasis Android.
- [7] Aini, N., Ramadiani, dan Hatta, H.R. 2017. Sistem Pakar Pendiagnosa Penyakit Tuberkulosis. Jurnal Informatika Mulawarman, 12(1), hal. 56-63.
- [8] Widiartanto, Y. H. 2016. 2016, Pengguna Internet di Indonesia Capai 132 Juta [online]. Tersedia dalam: <http://teknokompas.com/read/2016/10/24/15064727/2016.pengguna.internet.di.indonesia.capai.132.juta> [diakses 10 Maret 2017].
- [9] Puspitasari, D. 2012. Sistem Pakar Diagnosa Diabetes Nefropathy Dengan Metode Certainty Factor Berbasis Web Dan Mobile. Politeknik Elektronika Negeri Surabaya ITS, hal. 3.
- [10] Sirsat, S.S. dan Sirsat, M.S. 2016. A Validation of The Delone And Mclean Model On The Educational Information System of The Maharashtra State (India). International Journal of Education and Learning Systems, 1, hal. 9-18.
- [11] Saputro, P.H. 2015. Model Delone and Mclean untuk Mengukur Kesuksesan E-government Kota Pekalongan. Scientific Journal of Informatics, 2(1), hal. 2-5.
- [12] DeLone, W.H. dan McLean, E.R. 2003. The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. Journal of Management Information Systems, 19(4), hal. 9-30.
- [13] McDaniel, C. dan Gates, R. 2007. Marketing Research: Seventh Edition. John Wiley & Sons, Inc., USA.
- [14] Hair, dkk. 2010. Multivariate Data Analysis, Seventh Edition. Pearson Prentice Hall.

Aplikasi Survei Ubinan Berbasis Android

Betti Noviyani¹, Eko Budi Setiawan²

Program Studi Teknik Informatika, Universitas Komputer Indonesia

betti.noviyani@gmail.com

eko@email.unikom.ac.id

Diterima 23 April 2018

Disetujui 8 Juni 2018

Abstract— Central Bureau of Statistics is one of the institutions that collect data in all fields and one of them is data of food crops. The Central Bureau of Statistics conducted the ubinan survey to obtain food crop data. Found some obstacles in the field in conducting ubinan survey. This can reduce the quality of food crop data if not quickly addressed. Given food crop data is very important for the life of the community, it takes a tool that can help the surveyor so that the data produced can be more quickly and accurately. In this research will be made an application with a given feature that is to store the coordinate point of survey implementation, to provide location information on harvest measurement in the field, to detect harvest income writing errors, display compass as a tool for determining the direction of the wind. So that the work of ubinan survey can be further improved the quality of food crop data in the Central Bureau of Statistics.

Index Terms— Android, Survei, Ubinan, Tanaman Pangan.

I. PENDAHULUAN

Sektor pertanian merupakan salah satu sektor yang mempunyai peranan cukup penting dalam perekonomian Indonesia. Salah satu subsektor pertanian yang mempunyai peranan yang sangat strategis adalah subsektor tanaman pangan, yang meliputi tanaman padi dan palawija. Peranan penting subsektor tanaman pangan adalah dalam hal pemenuhan kebutuhan pangan masyarakat Indonesia.

Badan Pusat Statistik (disingkat BPS) bekerja sama dengan Direktorat Jenderal Tanaman Pangan, Kementerian Pertanian melakukan pengumpulan data produksi tanaman pangan setiap tahun. Sebagai pengumpul data ubinan, BPS juga menemukan beberapa kendala dalam pelaksanaannya. Salah satunya yaitu pelaksana ubinan di lapangan yang merasa kesulitan pada saat menentukan lokasi pengukuran hasil panen pada bidang terpilih. Karena prosesnya yang dianggap rumit oleh beberapa pelaksana di lapangan, tidak jarang pelaksana di lapangan perlu meminta bantuan staff atau pun rekan lainnya yang sudah menguasai betul tata cara penentuan lokasi ubinan dengan menggunakan cara manual yaitu menggunakan tabel angka random. Selain mudah dan cepat, alat bantu ubinan juga harus dapat mendeteksi pada saat lokasi ubinan melebihi pembatas sawah. Karena setelah lokasi didapatkan, seringkali lokasi pengukuran ubinan melewati

pembatas sawah. Sehingga penentuan lokasi pengukuran harus dilakukan kembali terus menerus sampai tidak melewati pembatas sawah.

Dikarenakan banyaknya sampel survei ubinan, tidak jarang pelaku survei ubinan harus melakukan survei di lebih dari 1 kecamatan. Hal ini menyebabkan pelaku survei ubinan kesulitan dalam menentukan arah mata angin yang diperlukan dalam penggambaran bentuk bidang yang harus sesuai arah mata anginnya.

Berdasarkan wawancara pada bulan Agustus 2017 dengan Ibu Ima Primasari selaku Kepala Seksi Statistik Produksi, beliau mengungkapkan bahwa diperlukan suatu teknologi untuk mencatatkan posisi KSK (Koordinator Statistik Kecamatan) dan staff pada saat melakukan survei ubinan. Hal ini dirasa perlu agar posisi survei ubinan dapat ditelusuri kembali pada saat ada pemeriksaan dari BPS Provinsi atau Dinas Pertanian. Dikarenakan banyaknya plot sampel ubinan di Kabupaten Sumedang, tidak jarang pelaksana survei ubinan kesulitan untuk menelusuri kembali lokasi pelaksanaan survei ubinan.

Selain kendala di lapangan, human error dirasa turut menjadi kendala dalam mengolah data hasil survei ubinan. Berdasarkan wawancara dengan Ibu Ima Primasari, tidak jarang pelaku survei ubinan salah menuliskan hasil survei ubinan pada lembar kuesioner. Terutama jika kesalahan terjadi pada penulisan penggunaan bibit dan pupuk serta pada penulisan hasil pengukuran berat hasil ubinan. Hal tersebut akan sangat mempengaruhi penghitungan produktivitas tanaman padi dan palawija di Kabupaten Sumedang khususnya. Maka diperlukan range acuan kewajaran dalam isian yang ditanyakan untuk menghindari salah pengisian.

Teknologi saat ini telah berkembang dengan cepat dan memberikan kemudahan dan manfaat dalam aktivitas kehidupan manusia [1]. Menurut pengamatan langsung oleh peneliti di BPS Kabupaten Sumedang pada bulan Agustus 2017, 96% pelaksana survei ubinan yang sudah menggunakan smartphone berbasis Android. Android adalah sebuah sistem operasi perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka [2]. Android memiliki OS yang sangat baik, cepat dan kuat serta memiliki antarmuka pengguna intuitif yang dikemas

dengan pilihan dan fleksibilitas [3]. Hampir setiap ponsel sekarang ini sudah dilengkapi dengan fitur Global Positioning System atau yang biasa disingkat dengan GPS. Salah satu fungsi GPS yaitu untuk menentukan letak atau lokasi dimana seseorang berada [4]. Dalam hal ini fitur yang digunakan yaitu A-GPS. Dalam penentuan arah mata angin, peneliti menggunakan fitur kompas digital yang ada pada ponsel android, kompas digital yang merepresentasikan kemana arah pengguna alat menghadap [5].

II. TINJAUAN PUSTAKA

A. Survei Ubinan

Pengumpulan data produktivitas tanaman pangan dilakukan secara sampel melalui Survei Ubinan dengan pendekatan rumah tangga. Tanaman padi meliputi padi sawah dan padi ladang, sedangkan tanaman palawija meliputi jagung, kedelai, kacang tanah, kacang hijau, ubi kayu dan ubi jalar. Pengumpulan data produktivitas tanaman pangan menerapkan metode pengukuran langsung pada plot ubinan terpilih dan metode wawancara dengan petani sampel untuk mengumpulkan data beberapa karakteristik yang berkaitan dengan produktivitas seperti penggunaan pupuk, benih, pengairan, pestisida, cara penanaman, dan sebagainya [6].

Berikut ini garis besar proses pendataan Survei Ubinan di lapangan:

1. Gambarkan bentuk lokasi survei ubinan pada Blok III dengan menyesuaikan arah mata anginnya yaitu utara berada di bagian atas.
2. Menghitung jumlah petak yang dikuasai oleh petani dan ditanami tanaman pangan terpilih. Kemudian memberikan nomor urut pada jumlah petak di seluruh bidang yang dikuasai dan ditanami tanaman pangan terpilih.
3. Jumlah petak dikalikan dengan angka random yang telah ditentukan oleh BPS Pusat, dan didapatkan nomor urut petak terpilih yang akan dilakukan pengukuran langsung.
4. Menggambarkan petak di seluruh bidang yang dikuasai dan ditanami tanaman pangan terpilih pada tempat yang sudah disediakan untuk mengetahui arah mata anginnya.
5. Menghitung panjang sisi petak dengan menggunakan langkah kaki biasa dari arah barat ke timur dan dari utara ke selatan.
6. Melakukan picingan mata pada tabel angka random untuk mendapatkan lima angka. Angka pertama untuk menentukan halaman tabel angka random yang akan digunakan (halaman 1 atau halaman 2), angka kedua dan ketiga untuk menentukan baris pada tabel angka random (baris 1 sampai baris 35), angka keempat dan kelima untuk menentukan kolom

pada tabel angka random (kolom 1 sampai kolom 25).

7. Dari halaman, baris dan kolom yang telah didapat, mulailah dicari empat angka untuk menentukan posisi panen yang akan diukur hasilnya. Angka pertama dan kedua untuk menentukan jarak dari barat ke timur nya. Angka ketiga dan keempat untuk menentukan jarak dari utara ke selatan nya. Ilustrasinya dapat dilihat pada gambar 1.



Gambar 1 Menentukan posisi pengukuran hasil panen pada survei ubinan

8. Dari titik yang sudah didapat maka KSK melakukan pengukuran hasil panen pada titik tersebut dengan ukuran bidang yang dipanen yaitu panjang 2,5 meter dan lebar 2,5 meter.
9. Setelah plot selesai dipanen, lalu hasil gabah kering panen (GKP) yang didapat ditimbang dan dihitung jumlah rumpun nya. Tuliskan berat dan banyaknya rumpun pada kuesioner.
10. Lakukan wawancara kepada pengelola lahan untuk beberapa pertanyaan yang ada di kuesioner (luas bidang, teknik penanaman, penggunaan pupuk, bibit, alat perontokan padi dan perbandingan hasil panen dengan tahun sebelumnya).

B. A-GPS

Assisted GPS (A-GPS) adalah salah satu kontributor utama meluasnya penggunaan GPS, terutama untuk ponsel dan unit genggam lainnya. A-GPS menggabungkan GPS dan komunikasi, terutama wireless dan memanfaatkan chip GPS dengan menambahkan daya pemrosesan murah dan ribuan correlator. Satelit GPS terbatas dalam jumlah daya yang dapat mereka berikan kepada pengguna di tanah ribuan mil jauhnya. A-GPS menyediakan informasi penting, melalui saluran komunikasi nirkabel terpisah, untuk secara substansial memperbaiki kekuatan pemrosesan receiver GPS, sehingga dapat beroperasi dengan sukses di lokasi dan situasi yang kurang

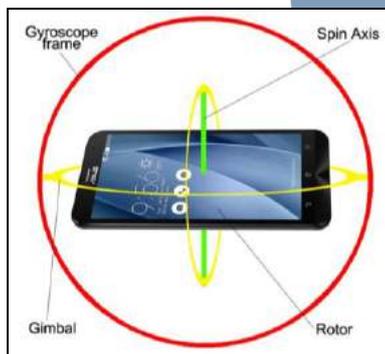
beruntung dimana bangunan, pohon, bukit sebagian dapat menurunkan sinyal GPS [7].

C. Kompas

Penentuan arah pada perangkat android memanfaatkan `Sensor.Type_Accelerometer` Dan `Sensor.Type_Magnetic_Field` untuk menentukan arah utara kompas, sehingga apabila device android tersebut diputar - putar kekiri ataupun kekanan, maka arah utara kompas pada device android tersebut tetap pada posisi utara yang benar [8].

Sensor yang digunakan adalah accelerometer, kompas dan giroskop, yang sebenarnya adalah tiga sensor dalam satu kesatuan. Selain accelerometer, kompas juga dikenal sebagai magnetometer, yang merasakan medan magnet yang paling kuat. Jika tidak ada magnet lain di dekat situ, medan magnet terkuat berasal dari kutub utara kutub.

Sensor ketiga adalah giroskop, yang merupakan alat yang bisa merasakan laju orientasi pada sumbu spin melalui torsi luar. Ini pada dasarnya berarti bahwa giroskop bisa merasakan ke arah mana kita berputar, dibandingkan dengan accelerometer yang merasakan arah mana kita memiringkan sensor [9]. Cara kerja kompas pada smartphone android dapat dilihat pada gambar 2.



Gambar 2 Cara kerja kompas pada smartphone android.

III. HASIL DAN PEMBAHASAN

Dalam bahasan ini terdiri dari analisis dan perancangan sistem, serta implementasi dan pengujian sistem.

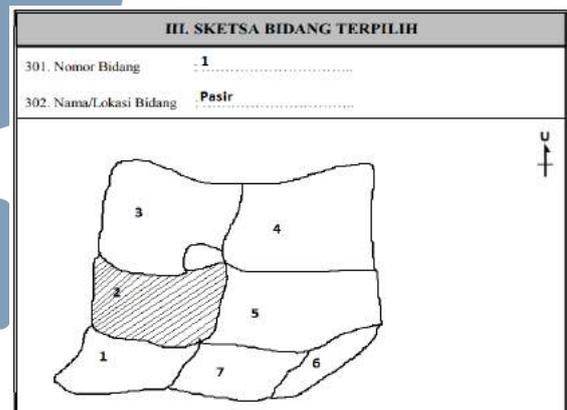
A. Analisis dan Perancangan Sistem

Aplikasi yang dibangun terdiri dari dua platform, yaitu berbasis website dan berbasis mobile android. Perancangan yang digunakan pada platform website dilakukan secara terstruktur, sehingga pemodelannya menggunakan konsep data flow diagram, sedangkan pada platform android dibuat secara object oriented programming (OOP) sehingga pemodelannya menggunakan konsep UML.

A.1. Analisis SOP Pelaksanaan Survei Ubinan

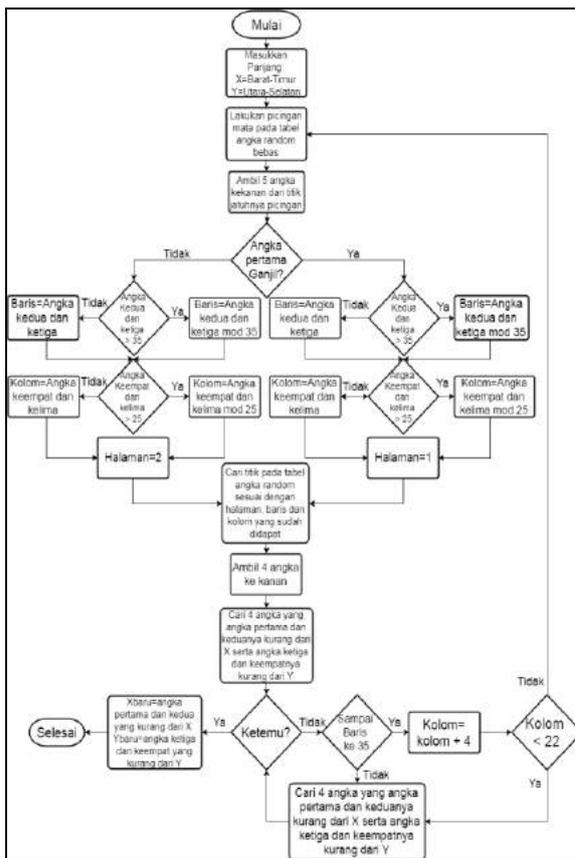
Survei ubinan adalah survei yang dilakukan oleh BPS setiap tahun dalam 3 periode. Kuesioner survei ubinan terdiri dari 4 halaman 9 blok. Dalam setiap blok, terdapat rincian-rincian pertanyaan yang penomorannya terdiri dari 3 angka. Angka pertama yaitu keterangan blok dan angka kedua dan ketiga adalah nomor urut rincian pertanyaan, misalnya rincian 402 berarti rincian kedua di blok 4. Dan berikut ini penjelasan mengenai setiap blok:

1. Blok I Keterangan Tempat. Blok I berisi keterangan tentang tempat, responden terpilih dan angka random yang telah ditentukan oleh pusat.
2. Blok II Keterangan Pengambilan Sampel Petak. Blok II berisi tentang jumlah bidang, jumlah petak yang terdapat tanaman terpilih serta penentuan petak terpilih dengan cara mengalikan jumlah seluruh petak pada seluruh bidang dengan nomor angka random yang terdapat pada blok I.
3. Blok III Sketsa Bidang Terpilih. Pada Blok III ini, pelaksana survei ubinan diminta menggambarkan bentuk bidang dimana terdapat petak terpilih dengan memperhatikan arah mata anginnya yaitu utara berada di bagian atas. Contoh penggambaran sketsa bidang terpilih dapat dilihat pada gambar 3.



Gambar 3 Penggambaran sketsa bidang terpilih

4. Blok IV Keterangan Pengambilan Sampel Plot. Blok IV ini adalah salah satu kendala bagi pelaksana survei ubinan di lapangan. Karena prosesnya yang dianggap rumit. Prosesnya dapat dilihat pada gambar 4.



Gambar 4 diagram alur proses pengerjaan blok IV

Contoh hasil picingan pada tabel angka random dapat dilihat pada gambar 5.

Contoh Hasil Picingan pada Tabel Angka Random

	.	14	15	16	17	18	19	20	21	22	23	24	25
.	.	9	9	1	0	8	2	5	3	7	5	9	
27	9	1	7	4	6	6	0	2	9	4	3		
28	2	5	1	2	6	3	8	7	9	7	8	4	
29	8	9	7	0	1	5	0	8	7	7	5	6	
30	4	3	3	4	9	1	3	3	4	8	6	8	

Lima angka di sebelah kanan tanda titik (.) adalah **26387**

Gambar 5 Contoh hasil picingan pada tabel angka random

Angka picingan mata sebanyak lima digit yang diperoleh dengan picingan mata diisikan pada tempat yang telah disediakan.

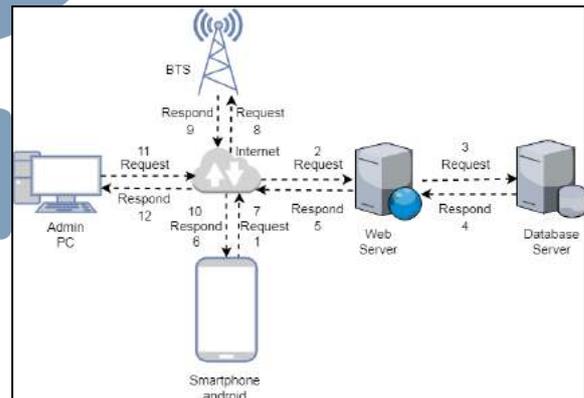
5. Blok V Keterangan Petugas. Blok V berisi keterangan petugas survei ubinan. Isinya yaitu berisikan nama petugas pencacah, nama pengawas/pemeriksa, tanggal pencacahan, tanggal pengawasan/ pemeriksaan.
6. Blok VI Keterangan Umum Tanaman Pangan. Blok VI berisi keterangan umum tanaman

pangan terpilih seperti jenis lahan, ukuran ubinan, luas tanaman, cara penanaman, sistem penanaman, pola penanaman, upaya peningkatan produksi, kelompok varietas, jumlah bibit dan pupuk serta cara penanggulangan hama yang dilakukan petani.

7. Blok VII Hasil Ubinan. Blok VII berisi tentang hasil ubinan. dengan cara melakukan penimbangan hasil panen pada posisi yang didapatkan pada blok IV tadi. Dengan cara memanen tanaman pada titik yang didapat dengan luas lahan 2,5 m X 2,5 m lalu hasilnya ditimbang dan diisikan pada Blok VII.
8. Blok VIII Keterangan Pendukung. Blok VIII berisi keterangan pendukung seperti sumber benih dan pupuk, keikutsertaan dalam kelompok tani, keadaan lahan dan perbandingan hasil dengan tahun lalu.
9. Blok IX Catatan. Blok IX digunakan untuk mencatatkan seluruh informasi tambahan terkait survei ubinan yang tidak wajar atau informasi lainnya yang dianggap diperlukan.

A.2. Analisis Arsitektur Sistem

Aplikasi akan dibuatkan dalam dua versi. Yang pertama yaitu aplikasi frontend yang berjalan pada smartphone android dan aplikasi backend yang berjalan pada personal computer (PC). Gambaran arsitektur sistem frontend dan backend dapat dilihat pada gambar 6.



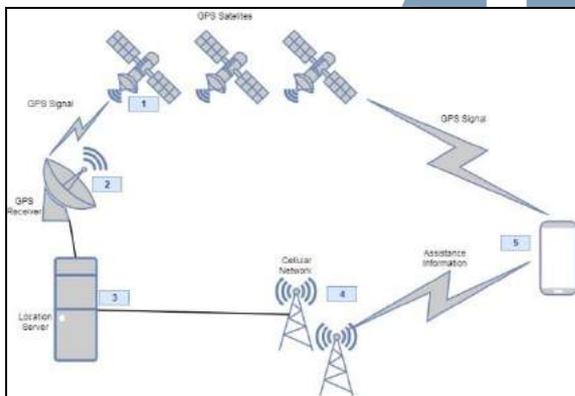
Gambar 6 Arsitektur sistem yang akan dibangun

1. User yang mengakses aplikasi pada smartphone me-request sebuah perintah yang harus melewati internet terlebih dahulu.
2. Web server menerima request dari smartphone dan menentukan jenis request.
3. Perangkat smartphone pengguna melakukan request data ke database server.
4. Request akan dikembalikan sebagai respond dari database server ke web server.
5. Respond diteruskan ke smartphone melewati internet terlebih dahulu.

6. Melalui internet, smartphone menerima data yang diambil dari database server.
7. Smartphone akan kembali me-request sebuah perintah melewati internet untuk mendapatkan lokasi user.
8. Request diteruskan ke BTS terdekat.
9. Setelah lokasi user didapatkan, BTS akan mengembalikan request berupa respond ke user melalui internet.
10. User menerima informasi berupa titik koordinat user dari BTS melalui internet.
11. User yang mengakses aplikasi pada PC sebagai aplikasi backend me-request sebuah perintah yang harus melewati internet terlebih dahulu.
12. Melalui internet, PC menerima data yang diambil dari database server.

A.3. Analisis cara kerja A-GPS

Cara kerja A-GPS pada smartphone android dapat dilihat pada gambar 7.



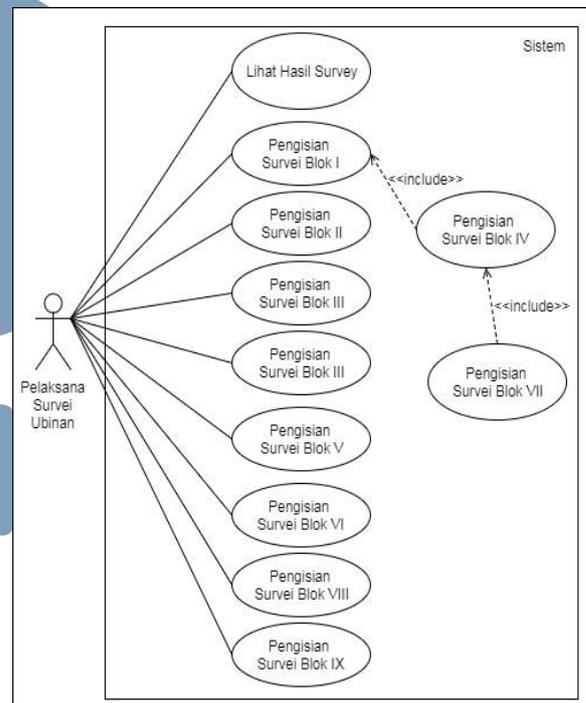
Gambar 7 Cara kerja A-GPS

1. GPS adalah sebuah sistem yang berguna untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (Synchronization) dengan sinyal satelit yang mengirimkan sinyal mikro ke bumi.
2. Kemudian sinyal mikro tersebut ditangkap dan diterima oleh perangkat GPS yang ada di bumi dan data yang ditangkap tersebut diolah untuk mendapatkan informasi lokasi, kecepatan, arah dan waktu.
3. Sinyal lalu diolah oleh setiap location server dari setiap provider yang lalu diteruskan melalui sinyal radio ke banyak menara BTS.

4. Menara BTS yang mengeluarkan sinyal memiliki fitur GPS juga yang dapat menentukan lokasi dengan tepat. Dan dikarenakan ruang lingkup suatu BTS tidak terlalu jauh sehingga operator telekomunikasi memiliki sinyal yang baik di semua tempat, maka operator telekomunikasi tersebut harus memiliki banyak menara BTS.
5. BTS yaitu menara tinggi yang dapat menghasilkan sinyal radio yang kemudian ditangkap oleh ponsel.

A.3. Use Case Aplikasi Mobile Android

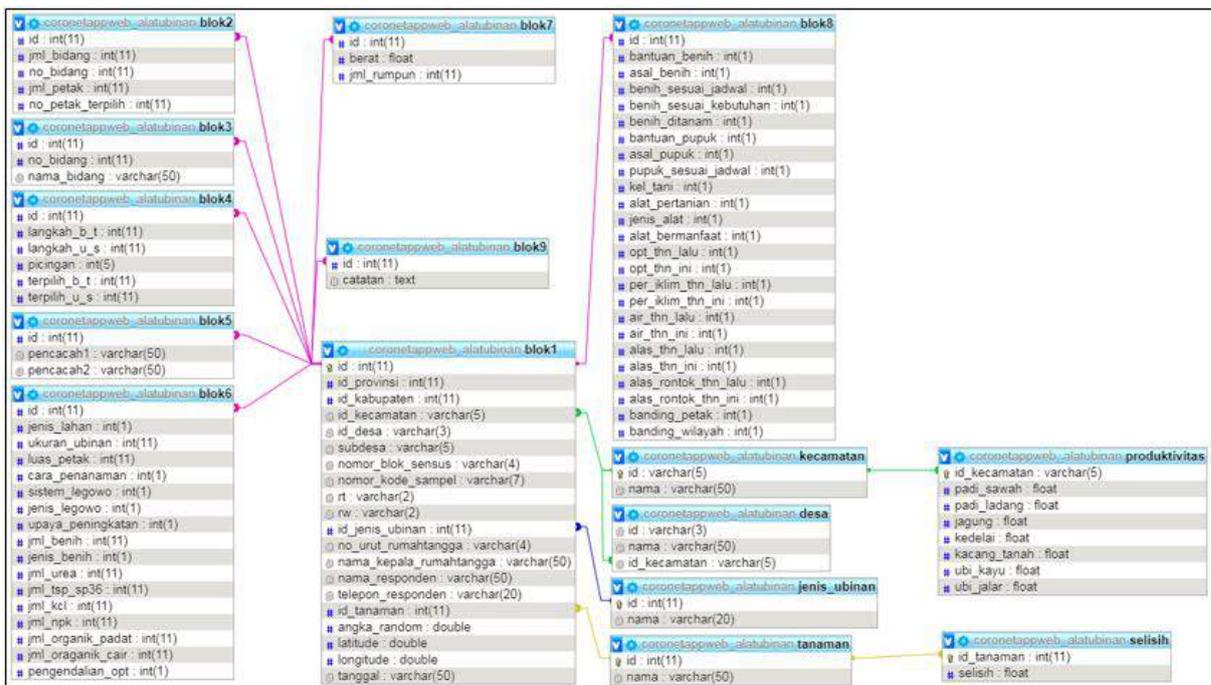
Sebuah use case merepresentasikan interaksi antara aktor dengan sistem. Use case sangat menentukan karakteristik sistem yang sedang dibuat. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [10]. Usecase diagram dari aplikasi yang dibangun dapat dilihat pada gambar 8.



Gambar 8 Use case diagram dari aplikasi android yang dibangun

A.4. Skema relasi tabel

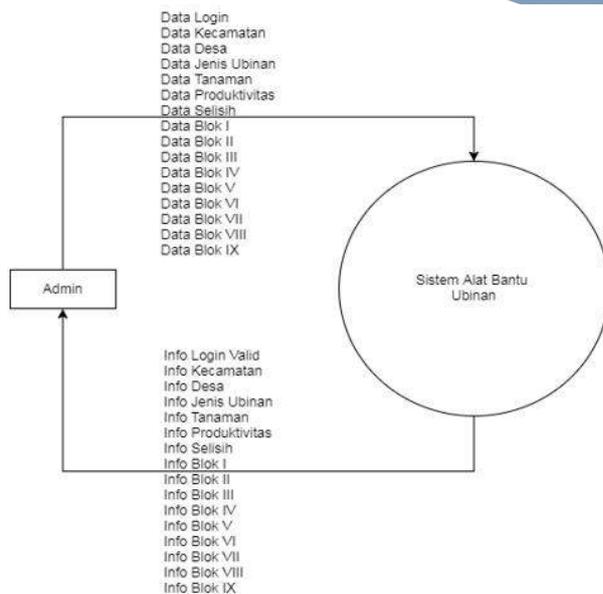
Skema relasi digunakan untuk menggambarkan keterhubungan antara tabel yang ada pada database. Skema relasi pada aplikasi ini dapat dilihat pada gambar 9.



Gambar 9 Skema relasi tabel

A.5. Diagram Konteks Aplikasi Backend

Diagram konteks menjelaskan bagaimana data digunakan untuk proses dalam bentuk aliran data yang masuk dan keluar dari sistem alat bantu ubinan. Berikut merupakan diagram konteks pada backend sistem alat bantu ubinan. Diagram konteks dalam aplikasi ini dapat dilihat pada gambar 10.



Gambar 10 Diagram konteks aplikasi backend.

B. Implementasi Sistem

Implementasi sistem merupakan tahapan untuk menerapkan perancangan yang telah dilakukan terhadap sistem sehingga siap untuk dioperasikan.

Implementasi sistem meliputi implementasi perangkat keras, implementasi perangkat lunak, implementasi data dan implementasi antar muka.

B.1. Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk membangun sistem frontend dan backend dapat dilihat pada tabel 1.

Tabel 1 Implementasi perangkat keras yang direkomendasikan

Nama Perangkat Keras	Spesifikasi
RAM Smartphone	1 GB
Memori Internal Smartphone	8 GB
RAM PC	2 GB
Koneksi Internet	5 mbps

B.2. Implementasi Perangkat Lunak

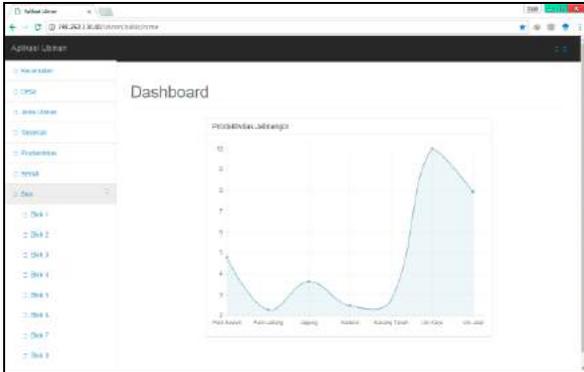
Aplikasi berbasis android ini memerlukan proses instalasi pada perangkat android. Pelaksana Survei harus memiliki aplikasi yang telah terinstall pada perangkat smartphone android yang dimilikinya, dan harus terhubung ke internet sebagai penghubung ke database. Perangkat lunak yang digunakan untuk menjalankan sistem frontend dan backend dapat dilihat pada tabel 2.

Tabel 2 Implementasi kebutuhan perangkat lunak

Perangkat Lunak	Spesifikasi
Sistem Operasi Mobile	Android
Peramban PC	Google Chrome/Mozilla Firefox

B.3. Implementasi Antarmuka

Implementasi antarmuka dilakukan terhadap setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk file aplikasi. Adapun tampilan dari implementasi antarmuka halaman backend dapat dilihat pada gambar 11.



Gambar 11 Implementasi antarmuka backend

Backend digunakan untuk mengelola setiap data pada database. Terdiri dari 15 menu untuk mengelola 15 tabel pada database yaitu Kecamatan, Desa, Jenis Ubinan, Tanaman, Produktivitas, Selisih, Blok 1 sampai dengan Blok 9.

C. Pengujian Sistem

Pengujian sistem dimaksudkan untuk mengetahui apakah sistem yang dibuat sudah memenuhi kriteria, sesuai dengan tujuan yang akan dicapai. Pengujian dilakukan dengan dua tahapan yaitu tahap pengujian alpha dan pengujian beta.

C.1. Rencana Pengujian

Pengujian sistem yang akan dilakukan terbagi menjadi dua tahapan. Tahap pertama yaitu pengujian alpha dan Tahap kedua dilakukan penelitian terhadap responden atau pengguna sistem dengan melakukan pengumpulan data menggunakan angket. Metode yang digunakan dalam penelitian ini adalah metode skala likert.

C.2. Pengujian Alpha

Tahap pertama pengujian alpha yaitu pengujian program berdasarkan fungsional dari program dengan menggunakan metode pengujian blackbox. Detail pengujian fungsional dapat dilihat pada tabel 3.

Tabel 3 Detail Pengujian Fungsional

Implementasi Antarmuka	Detail Pengujian	Hasil Pengamatan
 Entry Blok I	Menampilkan halaman entry Blok I	[√] Sukses Sesuai dengan output yang diharapkan
	Membuat sebuah data baru pada sistem	
	Verifikasi data Blok I	

Implementasi Antarmuka	Detail Pengujian	Hasil Pengamatan
 Entry Blok II	Menampilkan menu Blok 2 sampai dengan Blok 9	[√] Sukses Sesuai dengan output yang diharapkan
	Menampilkan halaman entry Blok 2	
	Verifikasi data blok II	
 Entry Blok III	Memilih kuesioner yang telah terisi blok I nya	[√] Sukses Sesuai dengan output yang diharapkan
	Memilih menu blok 3	
	Mengisi isian kuesioner blok III	
	Menekan tombol simpan data	
 Entry Blok IV	Memilih kuesioner yang telah terisi blok I nya	[√] Sukses Sesuai dengan output yang diharapkan
	Memilih menu blok 4	
	Mengisi isian kuesioner blok IV	
	Menekan tombol simpan data	
 Entry Blok V	Memilih kuesioner yang telah terisi blok I nya	[√] Sukses Sesuai dengan output yang diharapkan
	Memilih menu blok 5	
	Mengisi isian kuesioner blok V	
	Menekan tombol simpan data	
 Entry Blok VI	Memilih kuesioner yang telah terisi blok I nya	[√] Sukses Sesuai dengan output yang diharapkan
	Memilih menu blok 6	
	Mengisi isian kuesioner blok VI	
	Menekan tombol simpan data	
 Entry Blok VII	Memilih kuesioner yang telah terisi blok I nya	[√] Sukses Sesuai dengan output yang diharapkan
	Memilih menu blok 7	
	Mengisi isian kuesioner blok VII	
	Menekan tombol simpan data	
 Entry Blok VIII	Memilih kuesioner yang telah terisi blok I nya	[√] Sukses Sesuai dengan output yang diharapkan
	Memilih menu blok 8	
	Mengisi isian kuesioner blok VIII	
	Menekan tombol simpan data	
 Entry Blok IX	Memilih kuesioner yang telah terisi blok I nya	[√] Sukses Sesuai dengan output yang diharapkan
	Memilih menu blok 9	
	Mengisi isian kuesioner blok IX	
	Menekan tombol simpan data	

Berdasarkan pengujian yang telah dijelaskan di tabel 3, maka kesimpulan hasil pengujian alpha yang didapatkan yaitu poses-proses pada aplikasi telah melalui tahap perbaikan dan sudah dimaksimalkan. Dengan demikian secara fungsional aplikasi yang dibangun sudah menghasilkan output yang diharapkan sehingga sistem sudah layak untuk digunakan.

C.3. Hasil Pengujian Beta

Adapun pernyataan pada angket yang diberikan pada tanggal 4 Februari 2018 kepada 23 orang karyawan Badan Pusat Statistik Kabupaten Sumedang sebagai pengguna dari sistem yang dibangun dan hasil pengujianya serta perhitungannya dapat dilihat pada tabel 4.

Tabel 4 Hasil pengujian beta

Pernyataan pertama: Aplikasi ini membantu Anda untuk mendapatkan posisi yang tepat dalam pengukuran hasil panen di lapangan				
SS	S	R	TS	STS
7	16	0	0	0
Rata-rata = $(35 + 64 + 0 + 0 + 0) / 23 \times 100\% = 86\%$				
Pernyataan kedua: Aplikasi ini membantu Anda terhindar dari posisi ubinan melewati pembatas sawah atau kebun				
SS	S	R	TS	STS
5	16	2	0	0
Rata-rata = $(25 + 64 + 6 + 0 + 0) / 23 \times 100\% = 82\%$				
Pernyataan ketiga: Aplikasi ini memudahkan Anda dalam penentuan arah mata angin di lapangan				
SS	S	R	TS	STS
7	14	2	0	0
Rata-rata = $(35 + 56 + 6 + 0 + 0) / 23 \times 100\% = 84\%$				
Pernyataan keempat: Aplikasi ini membantu Anda mengurangi kesalahan penulisan isian kuesioner Survei Ubinan				
SS	S	R	TS	STS
6	15	2	0	0
Rata-rata = $(30 + 60 + 6 + 0 + 0) / 23 \times 100\% = 83\%$				
Pernyataan kelima: Tampilan aplikasi ini mudah dimengerti				
SS	S	R	TS	STS
11	12	0	0	0
Rata-rata = $(55 + 48 + 0 + 0 + 0) / 23 \times 100\% = 89\%$				
Pernyataan keenam: Aplikasi ini mudah digunakan				
SS	S	R	TS	STS
11	11	1	0	0
Rata-rata = $(55 + 44 + 3 + 0 + 0) / 23 \times 100\% = 88\%$				
Pernyataan ketujuh: Aplikasi ini membantu Anda dalam mengerjakan survei ubinan di lapangan				
SS	S	R	TS	STS
8	14	1	0	0
Rata-rata = $(40 + 56 + 3 + 0 + 0) / 23 \times 100\% = 86\%$				
Rata-rata akhir = $(86+82+84+83+89+88+86)/7 = 85\%$				

Berdasarkan hasil perhitungan pengujian beta yang telah disajikan pada tabel 4, maka dapat diambil beberapa kesimpulan hasil pengujian, yaitu responden merasakan bahwa aplikasi ini sudah membantu dalam pengerjaan survei ubinan di lapangan. Dapat dilihat dari hasil rata-rata akhir angket yaitu 85%. Hasil dari angket yang diberikan, dituangkan dalam bentuk interval pada gambar 12.



Gambar 12 Interval hasil pengujian beta

Garis merah adalah hasil rata-rata dari setiap pernyataan dan garis biru merupakan hasil rata-rata akhir. Dapat dilihat bahwa semua garis berada di range SS (sangat setuju), maka dapat disimpulkan bahwa aplikasi ini sudah dapat digunakan dan dimanfaatkan.

IV. SIMPULAN DAN SARAN

Berdasarkan pembahasan analisis, perancangan, implementasi sampai pada tahapan pengujian, maka dari itu dapat ditarik kesimpulan serta saran untuk aplikasi ini agar dapat mencapai pemanfaatan yang lebih baik lagi.

A. Simpulan

Kesimpulan dari penelitian ini yaitu :

1. Pelaksana survei ubinan dapat menggunakan aplikasi alat bantu ubinan untuk melakukan perhitungan dengan menggunakan tabel angka random secara otomatis untuk memudahkan pelaksana survei ubinan untuk mendapatkan posisi yang tepat dalam pengukuran hasil panen di lapangan.
2. Pelaksana survei ubinan akan mendapatkan posisi pengukuran hasil panen yang tidak melewati pembatas sawah atau kemungkinan lokasi ubinan yang melewati pembatas sawah diperkecil.
3. Pelaksana survei ubinan dapat melihat kompas langsung melalui aplikasi. Hal ini dapat mempermudah pelaksana survei ubinan dalam penentuan arah mata angin.
4. Pelaksana survei ubinan akan mendapatkan peringatan pada saat isian hasil ubinan melebihi data produktivitas ditambah atau dikurangi dengan selisih wajar setiap komoditas. Hal ini dapat mendeteksi kesalahan penulisan hasil survei ubinan oleh pelaksana survei ubinan akibat human error pada saat pengisian hasil ubinan.

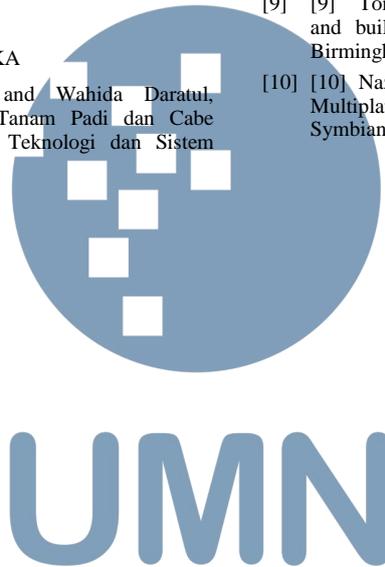
B. Saran

Adapun saran yang dapat dikembangkan untuk penelitian selanjutnya adalah :

1. Menyempurnakan fitur alat bantu ubinan agar data masukan dapat langsung dibuatkan laporan sehingga dapat langsung diolah oleh Badan Pusat Statistik.
2. Mengembangkan perangkat lunak dari segi performansi. Keterbatasan server dan hosting menjadi kekurangan. Kekurangan yang terjadi server terkadang mengalami down sehingga aplikasi tidak bisa diakses dan penggunaan bandwidth kuota hosting perlu ditingkatkan.
3. Mengembangkan platform yang dapat didukung oleh perangkat lunak, mengingat saat ini hanya mendukung platform android. Karena hal ini berkaca terhadap masyarakat luas yang tidak hanya menggunakan smartphone android saja.

DAFTAR PUSTAKA

- [1] [1] Harison, Mandarani Putri, and Wahida Daratul, "Perancangan Aplikasi Bercocok Tanam Padi dan Cabe Kriting Berbasis Android," *Jurnal Teknologi dan Sistem Informasi*, vol. 03, p. 1, 2017.
- [2] [2] Andi Juansyah, "Pembangunan Aplikasi Child Tracker Berbasis Assisted-Global Positioning System (A-Gps) Dengan Platform Android," *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA) Edisi 1*, vol. 1, p. 2, 2015.
- [3] [3] Surawijaya Surahman and Eko Budi Setiawan, "Aplikasi Mobile Driver Online Berbasis Android Untuk Perusahaan Rental Kendaraan," *Jurnal Ultima Infosys*, Vol.8 no.1, 2017.
- [4] [4] Latifa Khairani, R. A. Eva Soraya, Johannes Petrus, and Rachmansyah, "Rancangan Bangun aplikasi Pemantauan Posisi Anggota Keluarga Berbasis Android," p. 1, 2013.
- [5] [5] Luqman Hakim, Rifqi Budi Raharjo, and Didik Dwi Waluyo, "Prototype Robot Untuk Menentukan Arah Kiblat Dengan Tanda Shaf Sholat," p. 2, 2013.
- [6] [6] Badan Pusat Statistik and Kementerian Pertanian, *Pedoman Pengumpulan Data Survei Ubinan Tanaman Pangan 2015*. Jakarta: Badan Pusat statistik, 2015.
- [7] [7] Frank Van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, 2009.
- [8] [8] M. Didik R. Wahyudi, "Rancang Bangun Perangkat Lunak Penentu Arah Kiblat, Penghitung Waktu Shalat dan Konversi Kalender Hijriyah Berbasis Smartphone Android," *Jurnal Teknik*, vol. 5, no. 1, p. 3, April 2015.
- [9] [9] Tony Olsson, *Arduino Wearable Projects design, code and build exciting wearable project using Aduino tools*. Birmingham, United Kingdom: Packt Publishing Ltd., 2015.
- [10] [10] Nazruddin Safaat H, *Rancang Bangun Aplikasi Multiplatform Android | IOS | Windows Phone | Blackberry | Symbian*. Bandung: Informatika, 2015.



Pencarian *Question-Answer* Menggunakan *Convolutional Neural Network* Pada Topik Agama Berbahasa Indonesia

Rizqa Raaiqa Bintana¹, Chastine Fatchah², Diana Purwitasari³

Departemen Informatika, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

rizqa.raaiqa.bintana@gmail.com

chastine@cs.its.ac.id

diana.purwitasari@gmail.com

Diterima 6 Mei 2018

Disetujui 8 Juni 2018

Abstract—Community-based question answering (CQA) is formed to help people who search information that they need through a community. One condition that may occur in CQA is when people cannot obtain the information that they need, thus they will post a new question. This condition can cause CQA archive increased because of duplicated questions. Therefore, it becomes important problems to find semantically similar questions from CQA archive towards a new question. In this study, we use convolutional neural network methods for semantic modeling of sentence to obtain words that they represent the content of documents and new question. The result for the process of finding the same question semantically to a new question (query) from the question-answer documents archive using the convolutional neural network method, obtained the mean average precision value is 0,422. Whereas by using vector space model, as a comparison, obtained mean average precision value is 0,282.

Index Terms—community-based question answering, convolutional neural network, question retrieval

I. PENDAHULUAN

Komunitas tanya-jawab dibentuk untuk mempermudah seseorang dalam memperoleh informasi yang dibutuhkannya melalui suatu komunitas, contohnya Yahoo! Answers. Salah satu kondisi yang bisa terjadi dalam komunitas tanya-jawab adalah ketika pencari informasi tidak mampu menemukan informasi yang dibutuhkan, sehingga mereka akan menginputkan pertanyaan baru ke dalam sistem. Hal ini dapat mengakibatkan jumlah arsip dokumen pertanyaan dan jawaban menjadi ganda. Sehingga menjadi permasalahan yang penting untuk bisa menemukan pertanyaan yang sama secara semantik antara pertanyaan baru terhadap pertanyaan yang ada di arsip.

Classical retrieval models, seperti TF-IDF dan Okapi BM25, menggunakan representasi *bag-of-words* dan tidak mampu secara efektif menangkap informasi kontekstual dari sebuah kata. Model ini bekerja dengan mempertimbangkan informasi kemunculan kata dalam sebuah dokumen. Sebagian besar tugas

temu kembali (*retrieval*) menggunakan metode-metode yang berdasarkan semantik dengan pencocokan leksikal untuk pengambilan informasi. Hal ini sebagian disebabkan bahwa konteks yang sama sering dinyatakan dengan menggunakan kosa kata dan gaya bahasa yang berbeda dalam dokumen dan *query*. Beberapa penelitian menggunakan pengetahuan dari *WordNet* untuk menemukan kata yang sama secara semantik dan membantu pengukuran nilai kemiripan semantik diantara dua kata [1]. *WordNet* merupakan *database* leksikal yang menyimpan sinonim suatu kata, dan digunakan secara luas dalam analisa teks, namun terbatas hanya untuk kata dalam bahasa Inggris.

Berbagai metode diusulkan untuk pembelajaran representasi kata terdistribusi (*word embeddings*) dalam ruang vektor berdimensi rendah. Representasi kata terdistribusi membantu algoritma pembelajaran (*learning algorithm*) untuk mencapai kinerja yang lebih baik dengan cara mengelompokkan kata-kata yang mirip, dan telah diterapkan secara luas pada bidang bahasan pemrosesan bahasa alami (*natural language processing*) [2] [3]. Selain menggunakan representasi kata terdistribusi, beberapa metode lain untuk memodelkan kalimat (*neural sentence models*, model yang dikombinasikan terhadap struktur *neural network*), seperti *Neural Bag-of-Words* (NBOW), *recurrent neural network* (RNN), *recursive neural network* (RecNN), dan *convolutional neural network* (CNN) [4].

NBOW merupakan metode yang sederhana dan intuitif, namun mempunyai kekurangan dimana susunan kata menjadi hilang. Pemodelan kalimat berdasarkan RNN sensitif terhadap susunan kata, tetapi memiliki bias terhadap kata-kata terbaru yang dibutuhkan sebagai inputan. Hal ini memberikan RNN kinerja yang sangat baik dalam memodelkan bahasa, tetapi kurang optimal untuk memodelkan keseluruhan kalimat. RecNN mengadopsi struktur yang lebih umum untuk mengkodekan kalimat. Di setiap *node* dalam *tree*, konteks pada anak *node* kiri dan kanan

digabungkan oleh *classical layer*. Bobot dari lapisan dibagi di semua *node* dalam *tree*. Lapisan yang dihitung pada *node* atas memberikan sebuah representasi untuk kalimat. Namun, RecNN bergantung pada *external constituency parse tree* yang disediakan oleh *external parse tree*. CNN mempunyai kelebihan yaitu, dapat mempertahankan informasi susunan kata yang sangat penting untuk kalimat pendek, serta aktivasi nonlinier dalam CNN dapat mempelajari karakteristik yang lebih abstrak [4].

Pemodelan kalimat adalah cara untuk menganalisa dan merepresentasikan isi semantik yang ada dalam sebuah kalimat, yang melibatkan pemahaman bahasa alami. *Neural sentence models* digunakan untuk menghasilkan kata demi kata dari suatu kalimat [5] [6]. *Neural network* digunakan untuk mengekstrak struktur semantik yang tersimpan dalam sebuah kalimat ataupun dokumen. *Neural sentence models* bekerja dengan cara memetakan kata melalui inputan *query* dan representasi kata terdistribusi dari koleksi dokumen. Dari pemetaan kata tersebut akan diperoleh ekstrak kata yang dianggap sama secara semantik terhadap *query* melalui lapisan (*layer*) proyeksi.

Dalam penelitian ini, diterapkan model CNN untuk pemodelan semantik kalimat pertanyaan. Penelitian ini bertujuan untuk mengetahui performansi metode CNN dalam menemukan pertanyaan dari arsip komunitas tanya-jawab yang sesuai dan sama secara semantik dengan pertanyaan baru (*query*) yang diinputkan oleh penanya. Pada bagian berikutnya dalam tulisan ini berisi tinjauan literatur yang terkait. Dilanjutkan dengan bagian penjelasan metodologi penelitian, penjelasan hasil dan pembahasan. Dan bagian terakhir diberikan kesimpulan.

II. LITERATUR TERKAIT

A. Questions Retrieval

Dalam komunitas tanya-jawab, berbagai cara telah dipelajari untuk menyelesaikan permasalahan *lexical gap* dalam temu kembali pertanyaan (*questions retrieval*). Meskipun sebagian besar model *retrieval* yang sederhana mengasumsikan bahwa kemunculan kata benar-benar independen, namun informasi kontekstual sangat penting untuk mendeteksi maksud pencarian tertentu dari sebuah *query*. Pendekatan berbasis model terjemahan (*translation model-based*) diusulkan [7] yang mencoba untuk mengekstrak hubungan frase-ke-frase berdasarkan *click through data*. Hubungan tersebut diharapkan menjadi lebih efektif dalam menjembatani kesenjangan antara *query* dan dokumen. Pendekatan lainnya yang diterapkan dalam temu kembali pertanyaan adalah dengan pencocokan leksikal untuk pengambilan informasi. Pendekatan ini berdasarkan sifat semantik antar kata dalam *query* dan koleksi dokumen, sehingga konteks yang sama namun dinyatakan dengan kosa kata yang berbeda juga akan dapat di-*retrieve*.

B. Word Embeddings

Word embeddings (representasi kata terdistribusi) merupakan cara yang merepresentasikan kata-kata bahasa alami dengan cara mempertahankan kemiripan semantik dan sintaksis di antara kata-kata tersebut. Hal ini didapat melalui representasi kata-kata sebagai vektor berdimensi tinggi, yaitu hubungan spasial di antara *embeddings* merepresentasikan hubungan di antara kata-kata. Sebagai contoh, representasi dari kata “fisika” dan “kimia” akan dekat secara bersama, dan kata “mobil” akan dekat dengan kata “balap” dan “supir”. Ada dua teknik yang dikembangkan untuk memperoleh *word embeddings* yaitu, word2vec dan GloVe. Teknik tersebut dilakukan dengan mengolah bentuk bebas teks sehingga menghasilkan vektor berkualitas tinggi yang merepresentasikan kata-kata. Teknik yang digunakan dalam penelitian ini adalah teknik word2vec.

Word2vec, diperkenalkan oleh Mikolov dkk (2013), menggunakan teknik yang disebut “*skip-gram with negative sampling*”. Teknik ini tidak memprediksi kata berdasarkan pada konteks, tapi mencoba untuk memaksimalkan klasifikasi sebuah kata berdasarkan kata lain dalam kalimat yang sama. Lebih tepatnya, kita menggunakan setiap kata (*current word*) sebagai inputan untuk *log-linear classifier* dengan lapisan proyeksi yang kontinyu, dan memprediksi kata-kata dalam jarak tertentu sebelum dan setelah kata inputan tersebut (*current word*). Dari penelitian Mikolov dkk (2013) ditemukan bahwa peningkatan jarak memperbaiki kualitas vektor kata yang dihasilkan, tetapi juga meningkatkan kompleksitas komputasi. Karena kata-kata yang lebih jauh jaraknya biasanya kurang terkait dengan kata inputan daripada yang berjarak dekat dengan kata inputan. Berikut gambaran umum tentang cara kerja teknik word2vec:

- Mengambil kata di dalam koleksi dokumen (*corpus*) latih, dan sejumlah kata-kata yang terletak dekat dengan konteks.
- Merepresentasikan setiap kata-kata tersebut melalui sebuah vektor (sejumlah daftar kata).

Karena teknik word2vec dan GloVe menangkap hubungan semantik dan sintaksis, kedua teknik ini bisa digunakan untuk pencarian (sinonim, *query expansion*) serta rekomendasi. *Word embeddings* terlihat tidak memberikan diskriminatif antara konsep terkait namun konsep yang berbeda.

C. Convolutional Neural Network

Ada beberapa metode untuk memodelkan kalimat yang disebut *neural sentence model*. Peranan penting dari *neural sentence model* adalah untuk merepresentasikan *variable-length sentence* sebagai *fixed-length vector*. *Convolutional neural network* (CNN) merupakan salah satu *neural sentence model* yang digunakan untuk memodelkan kalimat [8]. Hal pertama yang dilakukan dalam tahap model CNN

adalah mentransformasi semua kata tunggal (*token*) dalam kalimat pertanyaan menjadi vektor melalui *lookup layer* dan menggunakan *word embedding* dalam kalimat secara berurutan. CNN merangkum makna sebuah kalimat melalui *convolutional layer* dan *pooling layer*, hingga mencapai sebuah representasi *fixed-length vector* pada lapisan (*layer*) akhir.

CNN mempunyai kelebihan, yaitu dapat mempertahankan informasi susunan kata dimana hal tersebut menjadi sangat penting untuk kalimat pendek. *Convolutional layer* menerapkan matriks filter satu dimensi yang melewati tiap baris fitur dalam matriks kalimat. Pembelitan (*convolving*) filter yang sama dengan *n*-gram di setiap posisi dalam kalimat memungkinkan fitur-fitur untuk diekstrak secara bebas dari posisi mereka dalam kalimat. *Convolutional layer* diikuti oleh *dynamic pooling layer* dan non-linearitas dari pemetaan fitur [8].

Arsitektur *convolutional* untuk kalimat pemodelan, seperti digambarkan pada Gambar 1, dibutuhkan sebagai inputannya berupa *word embedding* (yang dilatih terlebih dahulu dengan metode *unsupervised*) dalam kalimat yang selaras secara berurutan, dan meringkas makna kalimat melalui lapisan *convolutional* dan *pooling*, sampai mencapai representasi kata dalam *fixed length vector* pada lapisan terakhir.

Embeddings untuk seluruh kata dalam kalimat s membangun matriks inputan $\mathbf{s} \in \mathbb{R}^{n_w \times l_s}$, dimana l_s menyatakan panjang s . Sebuah *convolutional layer* diperoleh melalui *convolving* sebuah matriks dari bobot (filter) $\mathbf{m} \in \mathbb{R}^{n_m}$ dengan matriks aktivasi pada layer berikutnya, dimana m adalah lebar filter. Lapisan (*layer*) pertama diperoleh dengan menggunakan *convolutional filter* untuk matriks kalimat s dalam layer inputan. Dimensi n_w dan lebar filter m adalah *hyper-parameters* pada *network*.

Jaringan menangani rentetan inputan dari berbagai variasi panjang kata. Lapisan dalam jaringan interleave *convolutional layers* dan *dynamic k-maxpooling layers* satu dimensi. *Dynamic k-maxpooling* adalah generalisasi dari operator *max pooling*. Operator *max pooling* merupakan fungsi subsampling non-linear yang mengembalikan nilai-nilai maksimum [8].

III. METODE PENELITIAN

Untuk mencapai tujuan penelitian, maka detail rancangan keseluruhan proses yang dilakukan dalam penelitian ini seperti yang terlihat pada Gambar 2. Berdasarkan rancangan tersebut akan dibangun sistem yang mencakup keseluruhan proses untuk membantu proses *training* dan *testing* dalam penelitian ini.

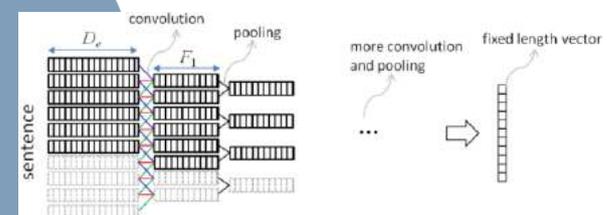
A. Dataset

Tahapan penelitian dimulai dari pengumpulan data penelitian. Dalam penelitian ini, *dataset* yang digunakan adalah dokumen pertanyaan-jawaban yang diambil dari komunitas tanya-jawab online www.piss-ktb.com. Pertanyaan baru yang diinputkan penanya,

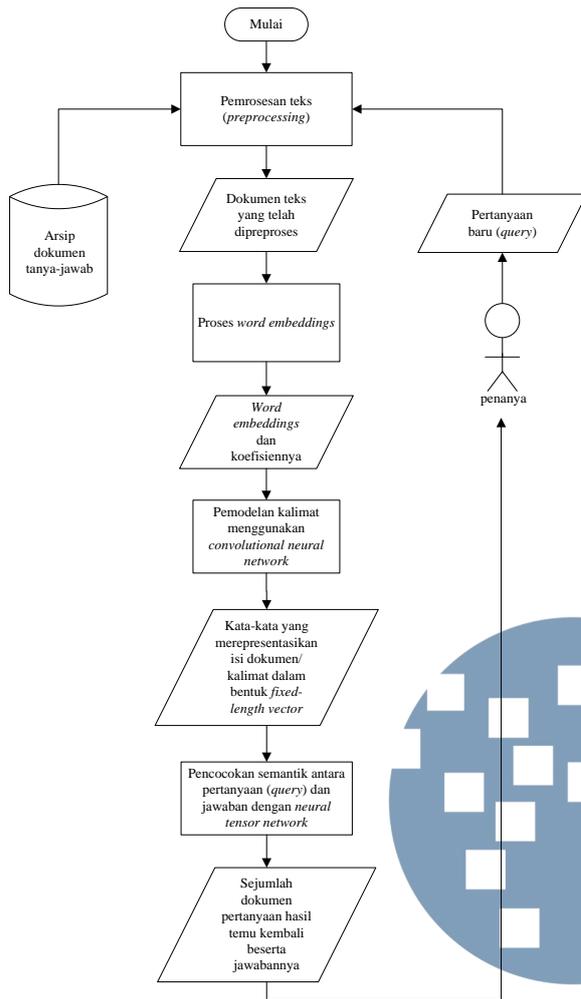
yang selanjutnya akan disebut *query*, akan dibandingkan dengan pertanyaan yang ada di arsip (koleksi dokumen) komunitas tanya-jawab dengan tujuan untuk menemukan pertanyaan dari arsip komunitas tanya-jawab yang sama secara semantik dengan *query*.

B. Pemrosesan Teks

Seperti yang dapat dilihat pada Gambar 2, beberapa proses yang akan dilakukan dalam penelitian ini setelah pengumpulan data yaitu, melakukan preproses terhadap arsip dokumen tanya-jawab dan *query*, melakukan proses *word embeddings* terhadap keseluruhan kata dalam dokumen tanya-jawab yang telah dipreproses, proses pemodelan kalimat untuk dokumen pertanyaan dan jawaban di sisi *training* dan *query* di sisi *testing* yang dimodelkan menggunakan *convolutional neural network*, dan mengukur kecocokan antara pertanyaan dan jawaban yang ada di arsip dokumen untuk proses *training* serta antara *query* dan dokumen jawaban pada proses *testing* menggunakan model *neural tensor network*.



Gambar 1. Arsitektur *convolutional* secara keseluruhan dalam memodelkan kalimat [9]



Gambar 2. Rancangan keseluruhan proses dalam penelitian

Pemrosesan teks dalam penelitian ini dilakukan sebagai tahapan preproses terhadap arsip dokumen tanya-jawab dan *query*. Tujuan dilakukannya pemrosesan teks adalah untuk membersihkan data sebagai langkah awal untuk analisis data. Proses yang dilakukan dalam tahapan ini, yaitu:

1. Tahap penghapusan *tag markup* dan format khusus dari dalam dokumen pertanyaan-jawaban. Sebelum dilakukan tokenisasi, semua *tag markup* dan format khusus akan dihapus dari dalam dokumen. Karena koleksi dokumen pertanyaan-jawaban yang digunakan dalam penelitian ini adalah *file* dengan ekstensi *.html*, maka seluruh *tag* maupun *javascript* serta *style* akan dihapus.
2. Tahap tokenisasi, merupakan proses pemisahan rangkaian kata. Dalam tahap ini, seluruh kata di dalam dokumen atau kalimat dipisahkan menjadi potongan kata tunggal (*term*). Dalam proses ini juga dilakukan penghapusan karakter-karakter tertentu, yaitu tanda baca serta mengubah semua kata ke bentuk huruf kecil (*lowercase*).

3. Tahap penghapusan *stop-words* (*linguistic preprocessing*). Setelah tahap tokenisasi dilakukan, maka dilanjutkan dengan tahap penghapusan *stop-words* dari dalam dokumen. Dalam tahap ini, ada dua operasi utama yang dilakukan yaitu penghapusan *stop-words* (*stop-words removal*) dan *stemming* (pemotongan imbuhan). *Stop-words* adalah kata yang sering muncul dalam dokumen, namun kata tersebut tidak dapat mendeskripsikan topik atau sub-topik dari dokumen tersebut, sehingga tidak dapat membedakan dokumen satu dengan dokumen lainnya di dalam koleksi (*corpus*). Karena itu, kata tersebut dihapus dari dalam dokumen. Contoh data inputan penelitian dan output dalam tahapan *preprocessing* dapat dilihat pada Tabel 1.

Tabel 1. Dokumen sebelum dan setelah tahapan *preprocessing*

Dokumen Sebelum <i>Preprocessing</i>
3729. puasa : cara mengqadha puasa yang tidak diketahui jumlahnya pertanyaan >> agus suryo komputro assalamu alaikum wa rahmatullahi wa barakatuh saya mau tanya nih ... jika saya (laki-laki) dulunya lalai dalam beribadah khususnya berpuasa bulan ramadhan sering bolong bolong. saya berniat membayar puasa ramadhan saya yang bolong tapi sudah lupa berapa banyak yang bolong apa yang harus saya lakukan berdasar al-quran & hadits yang ada ? syukron jawaban >> ghuftron bkl wa'alaikumussalaam warohmatullah wabarokaatuh wajib mengqadha puasa sampai yakin sudah dikerjakan semua. referensi: hawasyi asy-syarwani iii / 396 وَلَوْ عَلِمَ أَنَّهُ صَامَ بَعْضَ اللَّيَالِي وَبَعْضَ الْأَيَّامِ وَلَمْ يَعْلَمْ مِقْدَارَ الْأَيَّامِ الَّتِي صَامَهَا فَظَاهِرٌ أَنَّهُ يَأْخُذُ بِالْيَقِينِ فَمَا تَبَيَّنَتْهُ مِنْ صَوْمِ الْأَيَّامِ أَجْزَاءَ وَقَضَى مَا زَادَ عَلَيْهِ سِوَاهِ حَوَاشِي الشَّرْوَانِي ج ٣ ص ٣٩٦ مكتبة دار إحياء التراث العربي "apabila ada seseorang mengetahui bahwa dirinya berpuasa sebagian jatuh pada malam hari (karena tinggal di daerah yang tidak diketahui batas siang dan malamnya), dan sebagian jatuh pada siang hari, sedangkan dia tidak mengetahui jumlah puasa yang dikerjakan pada siang harinya, maka menurut qoul yang jelas orang tersebut wajib mengambil hitungan yang diyakininya, maka hitungan puasa siang hari yang diyakininya itu cukup baginya (untuk dijadikan jumlah puasa siang orang wajib hitung hitung puasa siang jumlah puasa siang wajib mengqadha sisa puasa malam wallahu a'lam
Dokumen Setelah <i>Preprocessing</i>
puasa mengqadha puasa jumlah laki lalai ibadah puasa bulan ramadhan bolong bolong niat bayar puasa ramadhan bolong lupa bolong laku dasar al-quran hadits wajib mengqadha puasa yakin hawasyi asy-syarwani puasa jatuh malam tinggal daerah batas siang malam jatuh siang jumlah puasa siang orang wajib ambil hitung hitung puasa siang jumlah puasa siang wajib mengqadha sisa puasa malam

C. Proses Pemodelan Kalimat Menggunakan *Convolutional Neural Network*

Selanjutnya akan dilakukan pemodelan semantik terhadap masing-masing kalimat pertanyaan dan jawaban di dalam dokumen. Dalam proses pemodelan semantik, digunakan model CNN. Metode CNN digunakan hanya untuk proses pemodelan kalimat,

bukan untuk proses klasifikasi seperti pada umumnya. *Hyperparameters* yang dibutuhkan dalam menerapkan model CNN, yaitu inputan kata representasi (*word embeddings*), jumlah *convolution filters*, *pooling strategies* (*max-pooling*), dan fungsi aktivasi. Dalam proses pemodelan semantik dengan CNN, terkait *natural language processing*, maka inputan yang digunakan berupa koefisien (nilai) dari masing-masing *word embeddings* terhadap kosa kata yang ada dalam kalimat pertanyaan serta jawaban dimana direpresentasikan sebagai matriks dua dimensi. *Word embeddings* diperoleh melalui cara seperti yang telah dijelaskan pada bab II. Data inputan untuk proses dalam memperoleh *word embeddings* adalah berupa dokumen teks yang telah dipreproses dan output dari proses ini berupa beberapa *word embeddings* yang merepresentasikan suatu kata beserta koefisien *word embeddings*-nya (yang menunjukkan nilai kemiripan atau kedekatan makna antara kata dan *word embeddings*-nya yang diperoleh dari proses *word embeddings*), sehingga satu kata memungkinkan akan memiliki beberapa kata lain yang mungkin memiliki kedekatan makna. Dalam penelitian ini, dari hasil akhir proses *word embeddings* untuk setiap kosa kata, diambil sebanyak 15 kata yang merepresentasikan sebuah kosa kata. Contoh output dari proses *word embeddings* seperti terlihat pada Tabel 2.

Hal selanjutnya yang dilakukan dalam tahap model CNN adalah mentransformasi semua kata tunggal (*token*) dalam kalimat (dalam hal ini direpresentasikan dalam bentuk koefisien dari masing-masing *word embeddings* yang dimiliki setiap kata tunggal dalam kalimat) menjadi vektor oleh *lookup layer*, kemudian mengubahnya (*encode*) menjadi *fixed-length vector* melalui *convolutional layer* dan *pooling layer* dengan kedalaman layer 3. Dalam penelitian ini, *filters slide* melewati *full rows* sebuah matriks sehingga lebar *filters slide* akan sama dengan lebar matriks inputan (jumlah *word embeddings* yang akan digunakan untuk per kata). Sedangkan untuk tinggi (*region-size*) *filters slide* atau *sliding windows* melewati 3 kata. *Convolutional layer* menerapkan matriks filter satu dimensi yang melewati tiap baris fitur dalam matriks kalimat. Pembelitan (*convolving*) filter yang sama dengan *n-gram* di setiap posisi dalam kalimat memungkinkan fitur-fitur untuk diekstrak secara bebas dari posisi mereka dalam kalimat. Gambar 3 menunjukkan gambaran dari proses pemodelan kalimat menggunakan CNN dengan inputan berupa koefisien dari masing-masing *word embeddings* (*d*) terhadap kata tunggal (*t*) dalam kalimat yang akan dimodelkan hingga diperoleh output dari proses ini dalam bentuk *fixed-length vector*.

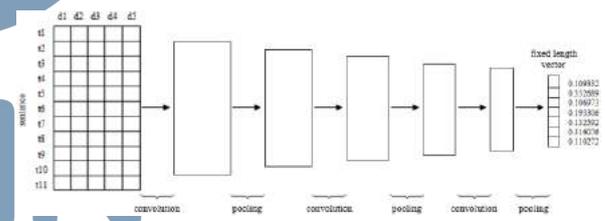
D. Pencocokan Semantik Antara Kalimat Pertanyaan-Jawaban Menggunakan *Neural Tensor Network*

Fixed-length vector dari masing-masing kalimat dokumen pertanyaan dan jawaban yang diperoleh dari proses pemodelan kalimat dengan menggunakan CNN akan diukur kecocokannya. Dalam penelitian ini, pencocokan tersebut dimodelkan dengan *non-linear tensor layer*, dimana sebelumnya sudah pernah

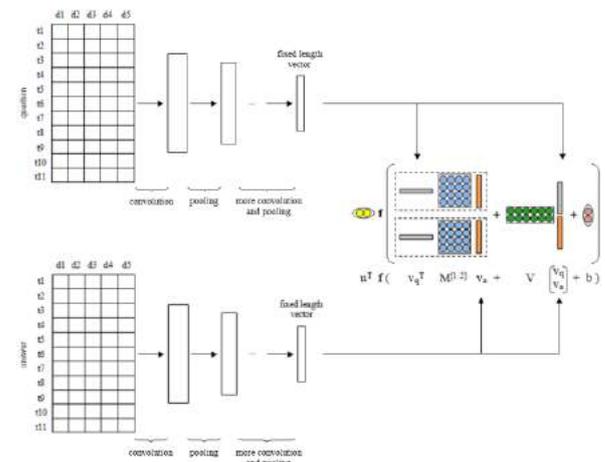
digunakan untuk pemodelan interaksi relasional data secara eksplisit [10]. Sebuah pertanyaan *q* dan jawabannya *a*, akan digunakan dua CNN untuk memodelkan keduanya menjadi *fixed vectors* *q* dan *fixed vectors* *a*. Berikutnya *Neural Tensor Network*, sebuah *tensor layer* yang diterapkan pada akhir dari CNN untuk memodelkan relasi antara pertanyaan dan jawabannya. Proses tersebut digambarkan seperti pada Gambar 4. *Tensor layer* menghitung kecocokan pasangan pertanyaan-jawaban melalui *score function* seperti pada (1).

Tabel 2. Beberapa kosa kata dan *word embeddings*-nya

Kosa Kata	Word Embeddings
puasa	mati; capai; tinggal; bawa; bani; ijthad; kalang; warga; bulan; alami; salah; saleh; pimpin; anggap; uang
mengqadha	harun; saleh; ahli; dosa; majlis; hasan; ijma; bani; puasa; sedekah; nyata; cari; takwil; anak; makkah
ibadah	uang; nadzar; jasa; capai; witiir; santri; negara; qadha; mudah; izin; pesantren; nikah; akibat; ajak; kencing
ramadhan	qobliyah; amal; hadir; qunut; jumat; subuh; tahiyat; maghrib; tidal; majlis; harap; shalatnya; neraka; pahala; fardhu
niat	arah; mutlak; jatuh; taukid; tarik; qobul; batin; talak; hajar; panitia; jil; ijthad; zain; syar; yakin
bayar	lunas; negara; pajak; berat; kena; angsur; riba; ganti; pecah; izin; qadha; palsu; makkah; akibat; wajib
wajib	rugi; qadha; santri; nadzar; jasa; zakat; negara; fidyah; nishab; akibat; masyhur; daerah; pindah; pajak; berat



Gambar 3. Deskripsi pemodelan kalimat menggunakan CNN



Gambar 4. Pencocokan kalimat antara pertanyaan dan jawaban [4]

$$s(q, a) = u^T f \left(v_q^T M^{[1:r]} v_a + V \begin{bmatrix} v_q \\ v_a \end{bmatrix} + b \right) \quad (1)$$

dimana $f = \tanh$ adalah *standard nonlinearity*, $M^{[1:r]} \in \mathbb{R}^{n_s \times n_s \times r}$ adalah sebuah *tensor*, r adalah jumlah *tensor slice*, dan parameter yang lainnya adalah bentuk standar dari *neural network*, $V \in \mathbb{R}^{r \times 2n_s}$, $b \in \mathbb{R}^r$ dan $u \in \mathbb{R}^r$.

Parameter dalam penelitian ini adalah L , W_{CNN}^q , W_{CNN}^a , u , $M^{[1:r]}$, V , dan b . Dimana L adalah *word embeddings*, W_{CNN}^q dan W_{CNN}^a adalah parameter dari CNN untuk pertanyaan dan jawaban, dan parameter lainnya dari *tensor layer*. *Objective function*-nya adalah (2).

$$L = \sum_{(q,a) \in C} \sum_{(q,a') \in C'} [\gamma - s(q, a) + s(q, a')] + \lambda \|\theta\|_2^2 \quad (2)$$

dimana $\gamma > 0$ (dalam penelitian ini nilainya 1) adalah *maximum margin*, dan (q, a') adalah pasangan pertanyaan-jawaban yang salah (random). C adalah kumpulan data latih dari pasangan pertanyaan-jawaban dalam dokumen dan C' menunjukkan kumpulan dari seluruh pasangan pertanyaan-jawaban yang salah. Dan untuk optimasinya, digunakan L-BFGS. Selanjutnya, *fixed-length vector* dari kalimat *query* dan dokumen jawaban akan diukur kecocokannya pada sisi *testing*. Output dari tahapan ini, yaitu berupa nilai kecocokan antara *query* dan jawaban dari arsip dokumen penelitian.

IV. HASIL DAN PEMBAHASAN

Untuk uji coba, jumlah pasangan dokumen yang digunakan terdiri dari 200 dokumen tanya-jawab dan akan diujikan untuk 5 pertanyaan baru (*query*) yang diinputkan penanya. Masing-masing dari *query* tersebut akan dibandingkan terhadap 200 dokumen tanya-jawab dengan tujuan untuk mendapatkan pertanyaan yang relevan terhadap *query* dengan menggunakan metode CNN. Output dari penelitian berupa nilai kecocokan antara *query* dan dokumen jawaban yang diperoleh menggunakan *neural tensor network* dengan menerapkan (1) dan (2) sebagaimana yang telah dijelaskan pada bab III. Output tersebut akan dikembalikan kepada penanya dalam bentuk tampilan dokumen pertanyaan-jawaban dari arsip berdasarkan urutan nilai kecocokan tersebut.

Pertanyaan-jawaban yang dikembalikan dinilai secara objektif apakah relevan atau tidak terhadap *query*. Selanjutnya dilakukan evaluasi dan validasi terhadap dokumen tanya-jawab hasil temu kembali dengan cara mengukur kualitas hasil temu kembali pertanyaan. Parameter yang digunakan untuk mengukur kualitas hasil temu kembali tersebut adalah *mean average precision* (MAP) dimana secara luas digunakan dalam *question retrieval*. *Mean average precision* untuk satu set *query* adalah rata-rata dari nilai presisi rata-rata (*average precision*) untuk setiap *query* yang dihitung menggunakan (3).

$$MAP = \frac{\sum_{q=1}^Q AveP(q)}{Q} \quad (3)$$

dimana Q adalah jumlah *query* dan *AveP* diperoleh dari (4) untuk masing-masing *query*. *Average precision* (*AveP*) adalah rata-rata nilai presisi yang diperoleh untuk kumpulan k dokumen teratas yang ada setelah setiap dokumen yang relevan di-*retrieve*, dan nilai ini kemudian dirata-ratakan berdasarkan kebutuhan informasi.

$$AveP = \frac{\sum_{k=1}^n (P(k) \times rel(k))}{\text{number of relevant documents}} \quad (4)$$

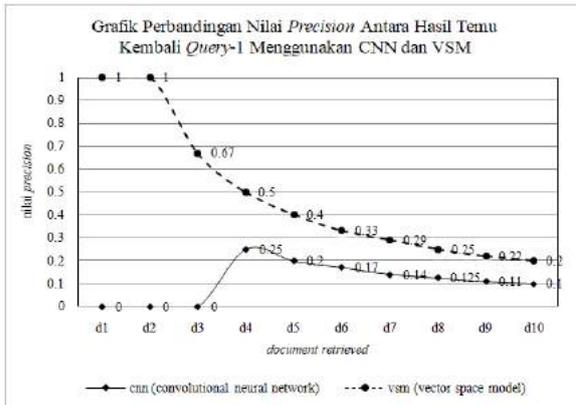
n adalah jumlah dokumen yang di-*retrieve*, $P(k)$ adalah nilai presisi dokumen di peringkat k yang dihitung menggunakan (5), dan $rel(k)$ merupakan fungsi indikator yang bernilai 1 jika dokumen di peringkat k adalah dokumen yang relevan, jika tidak maka bernilai 0. Jika dokumen yang relevan tidak di-*retrieve* sama sekali, maka nilai pada (4) dianggap 0 [11].

$$P(k) = \frac{\text{Jumlah dokumen retrieve dan relevant hingga } K \text{ teratas}}{K} \quad (5)$$

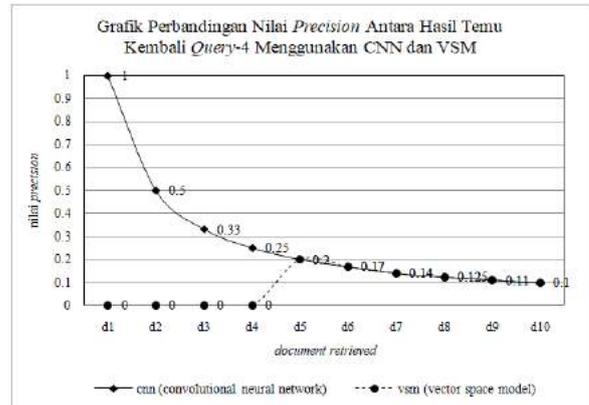
Dalam penelitian ini, nilai *precision* yang dihitung merupakan nilai *precision* pada masing-masing 10 dokumen teratas dari hasil temu kembali (*top-10 retrieved documents*), atau disebut juga sebagai *precision@10*. Kelima *query*, selain di uji cobakan terhadap metode CNN juga diuji cobakan terhadap metode VSM (*vector space model*) dalam tahapan proses mencari dan menemukan dokumen pertanyaan yang sama secara semantik dari dalam koleksi dokumen tanya-jawab terhadap *query*.

Gambar 5 memperlihatkan grafik perbandingan nilai *precision* masing-masing dokumen pertanyaan-jawaban hasil temu kembali (*retrieve*) terhadap *query*-1 yang berada di urutan 10 dokumen teratas hasil pencarian dengan menggunakan metode CNN dan VSM. Nilai *precision* tersebut diperoleh menggunakan (5). 10 dokumen pertanyaan-jawaban yang ditemukan dengan menggunakan metode CNN berbeda dengan yang ditemukan dengan menggunakan metode VSM.

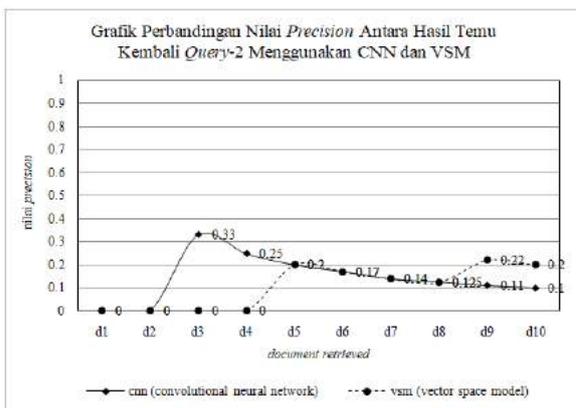
Untuk nilai *precision* dari hasil temu kembali dokumen pertanyaan terhadap masing-masing *query*-2, *query*-3, *query*-4, dan *query*-5 yang berada di urutan 10 dokumen teratas hasil pencarian menggunakan metode CNN dan VSM, dapat dilihat grafik perbandingannya pada Gambar 6, Gambar 7, Gambar 8, dan Gambar 9. Sepuluh dokumen pertanyaan-jawaban yang ditemukan oleh kedua metode tersebut berbeda-beda. Kelima *query* inputan tersebut masing-masing mengembalikan beberapa dokumen pertanyaan-jawaban dari dalam koleksi dokumen tanya-jawab sebagai hasil temu kembali dari proses pencarian.



Gambar 5. Grafik perbandingan nilai *precision* antara dokumen pertanyaan hasil temu kembali *query-1* menggunakan metode CNN dan VSM

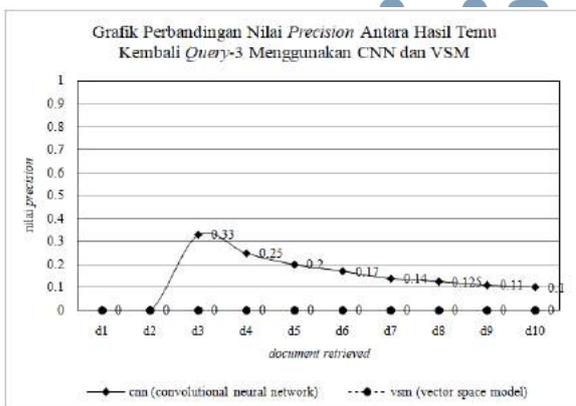


Gambar 8. Grafik perbandingan nilai *precision* antara dokumen pertanyaan hasil temu kembali *query-4* menggunakan metode CNN dan VSM

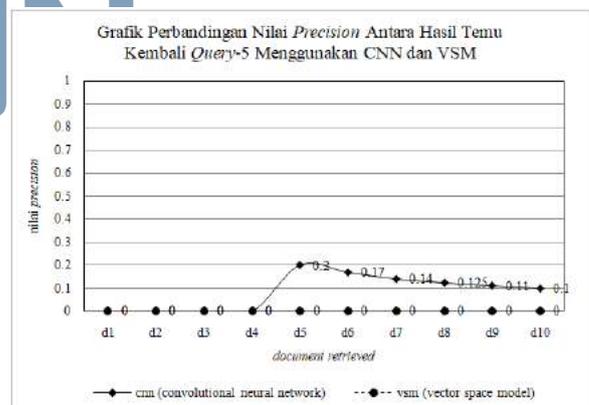


Gambar 6. Grafik perbandingan nilai *precision* antara dokumen pertanyaan hasil temu kembali *query-2* menggunakan metode CNN dan VSM

Dokumen pertanyaan-jawaban yang dikembalikan tersebut, oleh sistem dianggap memiliki korelevanan dengan *query*. Namun hanya diambil 10 dokumen pertanyaan dalam peringkat teratas yang dikembalikan oleh sistem untuk dijadikan acuan penilaian evaluasi terhadap hasil temu kembali pertanyaan oleh sistem. Dari hasil tersebut, dapat dihitung nilai *average precision*-nya untuk masing-masing *query* inputan menggunakan (4). Sehingga diperoleh nilai *average precision* terhadap hasil temu kembali dokumen pertanyaan menggunakan metode CNN untuk *query* inputan pertama hingga kelima berturut-turut yaitu, 0,25, 0,33, 0,33, 1, dan 0,2. Sedangkan nilai *average precision* terhadap hasil temu kembali dokumen pertanyaan menggunakan metode VSM untuk *query* inputan pertama hingga kelima berturut-turut yaitu, 1, 0,21, 0, 0,2, dan 0.



Gambar 7. Grafik perbandingan nilai *precision* antara dokumen pertanyaan hasil temu kembali *query-3* menggunakan metode CNN dan VSM



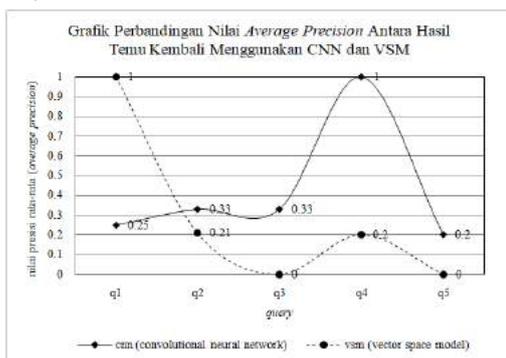
Gambar 9. Grafik perbandingan nilai *precision* antara dokumen pertanyaan hasil temu kembali *query-5* menggunakan metode CNN dan VSM

Tabel 3. *Average precision (AveP)* untuk hasil temu kembali pertanyaan menggunakan CNN dan VSM

Query Inputan Ke-	Kalimat Query	AveP (CNN)	AveP (VSM)
1	assalamualaikum... apakah wajib mencabut sesuatu yang palsu dari tubuh	0,25	1

Query Inputan Ke-	Kalimat Query	AveP (CNN)	AveP (VSM)
	jenazah sebelum dikubur?		
2	perintah membaca ta'awudz dalam al-quran. assalamualaikum... saya pernah dengar seorang khatib ketika akan mengucapkan ayat al-quran begini "qoolallahhu ta'ala fil qur anil adzim.. a u'dzubillahi minasysyaithonirrojim" setelah itu baru membaca ayat al-quran. pertanyaan saya : 1. apa khatib itu termasuk berbohong? karena di dalam al-quran tidak ada bacaan ta'awudz. 2. solusinya yang tepat gimana?	0,33	0,21
3	assalamu'alaikum. kepada para kyai dan member. mohon ditakwilkan mengenai mimpi meninggal dunia. sekian dan terimakasih sebelumnya	0,33	0
4	assalamualaikum... bagaimana hukum muamalahnya bensin oplosan / campuran ? apakah termasuk tindak kriminal ?	1	0,2
5	assalaamu'alaikum. afwan, mohon pencerahannya... hal-hal apa sajakah yang di senangi oleh allah swt ? sehingga allah swt memberi ridho dan rahmat-nya kepada orang tersebut ? syukron katsir	0,2	0

Nilai MAP untuk sekumpulan uji coba adalah rata-rata dari nilai *average precision* untuk setiap *query*. MAP dipengaruhi oleh bobot dari setiap *query* yang dilaporkan dalam bentuk penilaian *average precision* tiap *query*, baik itu diperoleh banyak dokumen pertanyaan yang relevan dengan *query* maupun yang sangat sedikit yang relevan dengan *query*. Nilai MAP untuk keseluruhan uji coba terhadap lima *query* yang berbeda tersebut dengan metode CNN dihitung menggunakan (3), sehingga diperoleh nilai MAP-nya yaitu 0,422. Sedangkan nilai MAP untuk keseluruhan uji coba dengan metode VSM yaitu 0,282.



Gambar 10. Grafik perbandingan nilai *average precision* antara hasil temu kembali menggunakan metode CNN dan VSM

V. SIMPULAN

Kemampuan metode *convolutional neural network* dalam menemukan pertanyaan yang sama secara semantik dengan *query* dari dalam arsip dokumen tanya-jawab bernilai 0,422 berdasarkan hasil hitungan *mean average precision*-nya (MAP). Sedangkan pencarian dengan menggunakan *vector space model*, MAP-nya bernilai 0,282. Untuk memperoleh nilai MAP yang lebih besar lagi (mendekati 1), dapat dilakukan penelitian lebih lanjut dengan kemungkinan menggabungkan penggunaan metode pengelompokan dokumen untuk proses pencarian yang lebih efisien atau penggunaan metode pencarian yang lainnya.

DAFTAR PUSTAKA

- [1] K. Wang, Z. Ming, and Tat-Seng Chua, "A Syntactic Tree Matching Approach to Finding Similar Question in Community-based QA Services," in *Proceedings of SIGIR*, 2009, pp. 187-194.
- [2] J. Turian, L. Ratinov, and Y. Bengio, "Word Representations: A Simple and General Method for Semi-supervised Learning," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Sweden, 2010, pp. 384-394.
- [3] R. Collobert et al., "Natural Language Processing (Almost) from Scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493-2537, 2011.
- [4] Xipeng Qiu and Xuanjing Huang, "Convolutional Neural Network Architecture for Community-based Question Answering," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2015, pp. 1305-1311.
- [5] T. Mikolov and G. Zweig, "Context Dependent Recurrent Neural Network Language Model," in *Spoken Language Technology (SLT)*, 2012, pp. 234-239.
- [6] N. Kalchbrenner and P. Blunsom, "Recurrent Continuous Translation Models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, 2013, pp. 1700-1709.
- [7] J. Gao, X. He, and Jian-Yun Nie, "Clickthrough-Based Translation Models for Web Search: from Word Models to Phrase Models," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, Canada, 2010, pp. 1139-1148.
- [8] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," in *Proceedings of ACL*, 2014.
- [9] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen, "Convolutional Neural Network Architectures for Matching Natural Language Sentences," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Canada, 2014, pp. 2042-2050.
- [10] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng, "Reasoning With Neural Tensor Networks for Knowledge Base Completion," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, Nevada, 2013, pp. 926-934.
- [11] C. D. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*. England: Cambridge University Press, 2009.

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 dengan subjek sesuai Jurnal yang dipilih.
- Seluruh isi naskah yang dikirimkan harus memenuhi syarat dan ketentuan yang ditentukan.
- Kami akan menjaga segala kerahasiaan dan Hak Cipta karya Anda.
- Sertakan biodata penulis pertama yang lengkap, meliputi nama, alamat kantor, alamat penulis, telpon kantor/ rumah dan hp, serta No NPWP (bagi yang memiliki NPWP).

6. Penilaian Naskah

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

7. Pengiriman Naskah Final

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

8. Copyright dan Honorarium

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

9. Biaya Tambahan

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

10. Alamat Redaksi

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

Judul Paper

Sub Judul (jika diperlukan)

Nama Penulis A¹, Nama Penulis B², Nama Penulis C²

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

² Baris pertama (dari afiliasi): nama departemen organisasi, nama organisasi, kota, negara
Baris kedua: alamat surel jika diinginkan

Diterima dd mmmmm yyyy

Disetujui dd mmmmm yyyy

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

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

I. PENDAHULUAN

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

Isi Pendahuluan mengandung latar belakang, tujuan, 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/m²” atau “webers per meter persegi,” bukan “webers/m².”
- Gunakan angka nol di depan suatu bilangan desimal, seperti “0,25” bukan “.25.”

C. Persamaan

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

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

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

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".

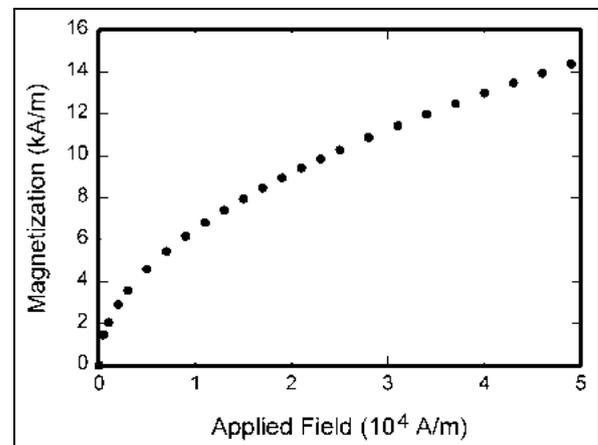
C. Gambar dan Tabel

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

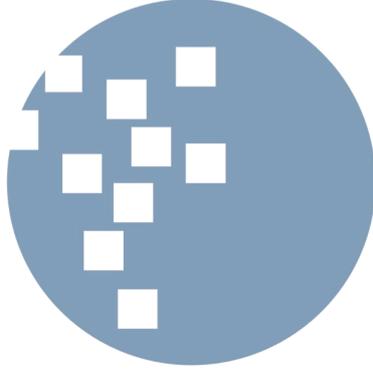
bawah kolom dimana catatan kaki tersebut dirujuk. Jangan letakkan catatan kaki di dalam daftar pustaka.

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

The logo for Universitas Muhammadiyah Negeri (UMN) is displayed in a large, blue, sans-serif font. The letters 'U', 'M', and 'N' are connected and have a slightly rounded, modern appearance. The logo is centered at the bottom of the page.



UMN

ISSN 2085-4552



9 772085 455006



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