

# ULTIMA Computing

## Jurnal Sistem Komputer

**FREDY SUSANTO, SYAFNIDAWATI**

Fast Tracking of Detection Offenders Smoking Zone Menggunakan Sensor MQ-2 Berbasis Internet of Things (1-4)

**FARICA PERDANA PUTRI, ADHI KUSNADI**

Pengenalan Tulisan Tangan Offline Dengan Algoritma Generalized Hough Transform dan Backpropagation (5-12)

**MITHA ANGGREANI RUPANG, ADHI KUSNADI**

Implementasi Metode Entropy dan Topsis Dalam Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik (Studi Kasus : Jakarta Smart City) (13-18)

**MUHAMMAD IQBAL PERKASA, EKO BUDI SETIAWAN**

Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token (19-26)

**ANDRE UTOMO MARTLIONG, NI MADE SATVIKA ISWARI**

Rancang Bangun Sistem Rekomendasi Restoran Menggunakan Metode AHP dan VIKOR pada Platform LINE (27-33)

**NUNIK AFRILIANA, ROSALINA, REGINA VALERIA**

Pendeteksian Ruang Kosong Parkir di dalam Ruangan (34-40)



**UMN**

## SUSUNAN REDAKSI

### Pelindung

Dr. Ninok Leksono

### Penanggungjawab

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

### Pemimpin Umum

Hargyo Tri Nugroho Ignatius, S.Kom., M.Sc.

### Mitra Bestari

(UMN) Hira Meidia, B.Eng., Ph.D.

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

(UMN) Dr. Hugeng, S.T., M.T.

(Universitas Indonesia) Filbert Hilman Juwono,

S.T., M.T.

(Tanri Abeng University) Nur Afny Catur

Andryani, M.Sc.

(UMN) Kanisius Karyono, S.T., M.T.

(UMN) Felix Lokananta, S.Kom., M.Eng.Sc.

(UMN) Samuel, M.T.I.

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

(UMN) Wolfgang Xaverius D. J. N., S.T., M.Sc.

(UMN) Nabila Husna Shabrina, S.T., M.T.

### Ketua Dewan Redaksi

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

### Dewan Redaksi

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

Samuel, M.T.I.

### Desainer & Layouter

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

### Sirkulasi dan Distribusi

Sularmin

### Keuangan

I Made Gede Suteja, S.E.

## ALAMAT REDAKSI

Universitas Multimedia Nusantara (UMN)

Jl. Scientia Boulevard, Gading Serpong

Tangerang, Banten, 15811

Tlp. (021) 5422 0808

Faks. (021) 5422 0800

Email: ultimacomputing@umn.ac.id



**Jurnal ULTIMA Computing** merupakan Jurnal Program Studi Sistem Komputer Universitas Multimedia Nusantara yang menyajikan artikel-artikel penelitian ilmiah dalam bidang Sistem Komputer serta isu-isu teoritis dan praktis yang terkini, mencakup komputasi, organisasi dan arsitektur komputer, *programming*, *embedded system*, sistem operasi, jaringan dan internet, integrasi sistem, serta topik lainnya di bidang Sistem Komputer. Jurnal ULTIMA Computing terbit secara berkala dua kali dalam setahun (Juni dan Desember) dan dikelola oleh Program Studi Sistem Komputer Universitas Multimedia Nusantara bekerjasama dengan UMN Press.

# Call for Paper



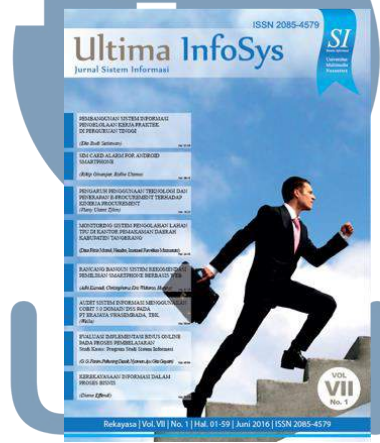
**International Journal of New Media Technology (IJNMT)** is a scholarly open access, peer-reviewed, and interdisciplinary journal focusing on theories, methods and implementations of new media technology. IJNMT is published annually by Faculty of Engineering and Informatics, Universitas Multimedia Nusantara 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 31<sup>st</sup>, 2018**  
Deadline for submission of papers
- November 30<sup>th</sup>, 2018**  
Announcement for Acceptance
- December 14<sup>th</sup>, 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

<b>Fast Tracking of Detection Offenders Smoking Zone Menggunakan Sensor MQ-2 Berbasis Internet of Things</b>	
Fredy Susanto, Syafnidawati	1-4
<b>Pengenalan Tulisan Tangan Offline Dengan Algoritma Generalized Hough Transform dan Backpropagation</b>	
Farica Perdana Putri, Adhi Kusnadi	5-12
<b>Implementasi Metode Entropy dan Topsis Dalam Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik (Studi Kasus : Jakarta Smart City)</b>	
Mitha Anggreani Rupang, Adhi Kusnadi	13-18
<b>Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token</b>	
Muhammad Iqbal Perkasa, Eko Budi Setiawan	19-26
<b>Rancang Bangun Sistem Rekomendasi Restoran Menggunakan Metode AHP dan VIKOR pada Platform LINE</b>	
Andre Utomo Martliong, Ni Made Satvika Iswari	27-33
<b>Pendeteksian Ruang Kosong Parkir di dalam Ruangan</b>	
Nunik Afriliana, Rosalina, Regina Valeria	34-40

The logo of Universitas Muhammadiyah Negeri (UMN) is a large, semi-transparent watermark in the center of the page. It consists of a blue circle containing a white grid pattern, with the letters 'UMN' in a bold, blue, sans-serif font positioned below the circle.

## KATA PENGANTAR

Salam ULTIMA!

ULTIMA Computing – Jurnal Sistem Komputer 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 komputasi, organisasi dan arsitektur komputer, programming, embedded system, sistem operasi, jaringan dan internet, integrasi sistem, serta topik lainnya di bidang Sistem Komputer.

Pada ULTIMA Computing Edisi Juni 2018 ini, terdapat delapan artikel ilmiah yang berasal dari para peneliti, akademisi, dan praktisi di bidang Sistem Komputer, yang mengangkat beragam topik, antara lain: Fast Tracking of Detection Offenders Smoking Zone Menggunakan Sensor MQ-2 Berbasis Internet of Things; Pengenalan Tulisan Tangan Offline Dengan Algoritma Generalized Hough Transform dan Backpropagation; Implementasi Metode Entropy dan Topsis Dalam Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik (Studi Kasus : Jakarta Smart City); Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token; Rancang Bangun Sistem Rekomendasi Restoran Menggunakan Metode AHP dan VIKOR pada Platform LINE; Pendeteksian Ruang Kosong Parkir di dalam Ruangan.

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

Akhir kata, kami mengucapkan terima kasih kepada seluruh kontributor dalam ULTIMA Computing 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

# Fast Tracking of Detection Offenders Smoking Zone Menggunakan Sensor MQ-2 Berbasis Internet of Things

Fredy Susanto<sup>1</sup>, Syafnidawati<sup>2</sup>

<sup>1</sup> Sistem Komputer, STMIK Raharja, Jl. Jenderal Sudirman No.40 Cikokol- Tangerang 15117

<sup>2</sup> Teknik Informatika, STMIK Raharja, Jl. Jenderal Sudirman No.40 Cikokol- Tangerang 15117

fredy@raharja.info

syafnidawati@raharja.info

Diterima 22 Maret 2018

Disetujui 8 Juni 2018

**Abstract**—With the increasing violations by smokers in free zone cigarettes, very affecting for human health cigarettes or not smokers. So that designed a supervision system to reduce the number of violations by smokers. The system is based on the aimed to detect and provide a warning to smokers that the room cannot be cigarettes if breaking will be imposed sanctions. This system is made up of hardware, consists of smoke sensor base their MQ-2, Camera C270 and Raspberry pi. This system created by the language of Python programming technique. How this appliance if violations by smokers, when smoke is detected then the webcam will take pictures of the offenders and the results of the photo will be sent automatically to the email, if the results of the photo has been sent to email and alert sound alarm will be active and a notification that in the area cannot be cigarettes. The results of the design of this system is not only to detect and provide a notification only, but it also has been able to give a warning and evidence of who is a breach in the zone.

**Index Terms**— Smoke sensor, MQ-2, raspberry pi.

## I. PENDAHULUAN

Pada kehidupan sehari-hari banyak sekali dampak asap rokok pada perokok di sekitar kita, banyak sekali kerugian yang ditimbulkan akibat asap rokok bagi kesehatan manusia karena asap rokok tersebut sangat berbahaya bagi kesehatan. Asap rokok mengandung berbagai zat kimia yang sangat berbahaya yang dapat menyebabkan bermacam-macam penyakit seperti batuk kronis kanker paru-paru dan gangguan kesehatan lainnya [3].

Walaupun asap rokok ini sangat berbahaya bagi kesehatan tapi kebutuhan akan rokok ini masih terbilang cukup besar dikarenakan banyak orang yang tidak peduli dengan kesehatan dan efek negative pada asap rokok ini. Faktor buruk yang dibawa oleh asap rokok maka pada daerah tertentu diadakan larangan yang tidak memperbolehkan seorang untuk rokok misalnya pada rumah sakit, gedung, sekolah, bioskop dan ruangan perkantoran yang ber AC dan lain lain.

Hal tersebut dilakukan agar asap pada perokok tidak mengenai orang yang tidak rokok dan asap yang ditimbulkan tidak mengganggu orang lain yang berada disekitarnya. Perkembangan teknologi sangat maju dengan pesat perkembangan teknologi ini merupakan hasil kerja keras dari rasa ingin tahunya manusia terhadap suatu hal yang pada akhirnya diharapkan akan mempermudah manusia dengan adanya teknologi tersebut akan bermunculan alat alat yang sangat canggih yang dapat bekerja secara otomatis yang dapat memudahkan pekerjaan manusia.

Salah satu sistem cerdas yang dibutuhkan manusia adalah alat yang dapat mendeteksi asap rokok dan memberi peringatan dan dapat memberikan kepada perokok untuk tidak rokok di area tersebut dengan demikian dengan adanya alat ini dapat diharapkan pengawasan terhadap rokok tidak lagi dibutuhkan karena sudah dilakukan secara otomatis dan diharapkan dengan adanya alat ini juga dapat meningkatkan tingkat kedisiplinan perokok untuk tidak rokok pada area tertentu [2].

## II. PERMASALAHAN

Sistem yang berjalan saat ini masih menggunakan cara manual jadi pengawas memeriksa area di sekitar apakah ada yang melakukan pelanggaran jika tercium asap rokok di sekitar maka pengawas akan melakukan tindakan seperti mencari bukti bahwa ada yang melakukan pelanggaran jika terbukti maka akan diproses atau di tegur, berdasarkan hal tersebut maka dirancanglah suatu alat yang dapat mendeteksi asap rokok pada suatu zona dan memberi peringatan kepada perokok bahwa di zona tersebut tidak diperbolehkan untuk rokok.[7] Berikutnya adalah kenapa menggunakan system tidak menggunakan sistem mikrokontroller yang lain.

## III. METODE PENELITIAN

Dalam metode ini memanfaatkan sistem kecerdasan buatan yang diterapkan pada mikrokontroller untuk memberi informasi, yang

dijelaskan pada gambar 1, yaitu terdiri dari tiga layer : layer akuisisi data , layer transport data, dan layer pengolahan data. Data akuisisi diperoleh dari sinyal asap yang dihasilkan, layer transport diberikan adalah sinyal asap diteruskan ke sinyal data TCP/IP sehingga dapat dikenali oleh media web.



Gambar 1. Metode penitian Arsitektur IOT.

IV. PEMECAHAN MASALAH

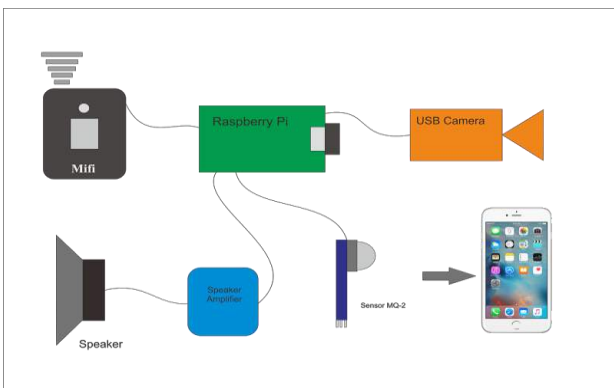
cangan ini dibuat dengan menggunakan hardware yang minim dengan Raspberry Pi sebagai otak atau pusat pengendali semua program yang ada. Untuk visualisasi video akan digunakan logitech c270 sebagai kamera dan untuk koneksi ke jaringan internet. Rangka yang digunakan disertai dengan satu alarm dan sensor MQ-2 yang akan bertugas sebagai pendeteksi objek yang akan disesuaikan dengan kebutuhan pergerakan alat ini.

A. Literature Review

Penelitian yang dilakukan oleh Brawner Brian L. Heyasa dan Van Ryan Kristoper dengan judul Initial Development and testing of Microcontroller MQ-2 Gas Sensor for University Air Quality Monitoring, dalam jurnal ini menggunakan alat microcontroller arduino uno sebagai pengganti alat sensor yang cukup mahal. IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEEE) e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 12, Issue 3 Ver. II (May – June 2017), PP 47-53 www.iosrjournals.org [12]

B. Diagram Blok

Pada gambar 2, Dalam perancangan pemantauan dengan Raspberry Pi secara umum perangkat keras atau hardware minimal dibutuhkan beberapa komponen elektronika sebagai berikut, yaitu: SD Card untuk Operating System (OS). Camera Webcam untuk menangkap atau capture gambar, serta perlengkapan atau device penunjang lainnya. Agar sistem dapat bekerja dan berjalan dengan baik sesuai dengan fungsi. Untuk mudah dipahami penulis membuat diagram blok dan alur kerjanya:

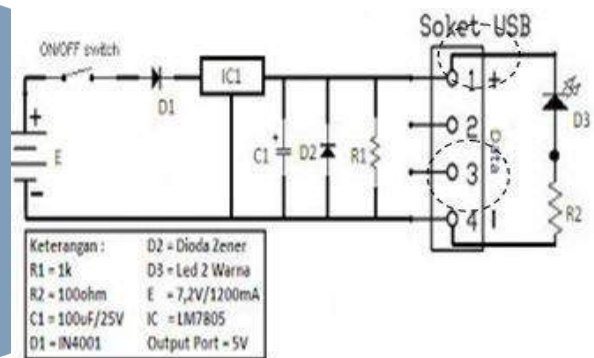


Gambar 2. Diagram Blok

C. Pengujian rangkaian catu daya

Catu daya sebagai sumber tegangan pergerakan alat merupakan bagian yang sangat penting. Dalam merealisasi sistem alat ini dibutuhkan catu daya. untuk Raspberry Pi membutuhkan tegangan dan arus yang cukup besar untuk melakukan pergerakan. Raspberry Pi hanya membutuhkan tegangan sebesar 5v untuk dapat bekerja, sedangkan untuk sensor MQ-2 minimal 3.5v dan bisa menerima tegangan sampai dengan 5v.[5]

Pengujian Catu Daya untuk Raspberry Pi dilakukan dengan cara menggunakan multimeter. Ujung multimeter berwarna merah dihubungkan ke pada pin positif pada soket USB dan ujung multimeter berwarna hitam dihubungkan ke pin negatif pada soket USB [8].



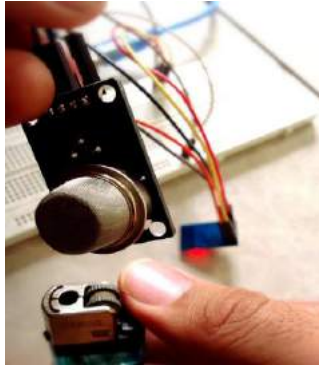
Gambar 3. Pengujian Catu Daya untuk Raspberry Pi

Setelah dilakukan pengujian sesuai gambar 9 didapatkan hasil tegangan yang keluar dari Catu Daya sebesar 5v dengan arus 3 Ampere. Hasil ini bisa dikatakan cukup untuk menghidupkan Raspberry Pi dan sensor MQ-2.[11]

D. Pengujian Sensor MQ-2

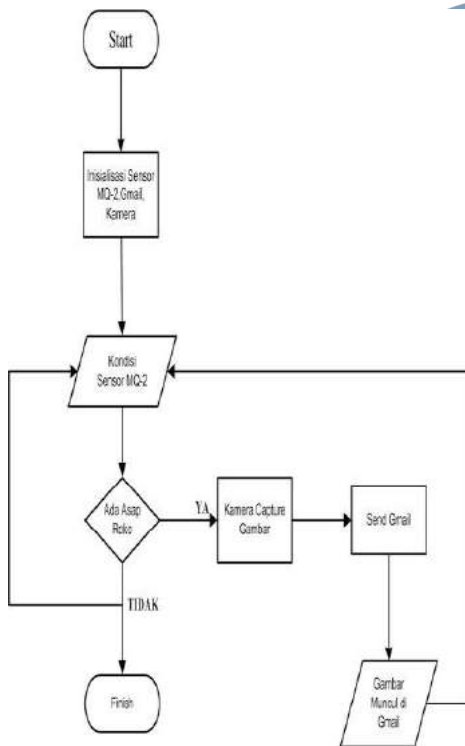
Dilihatkan pada gambar 4, Pengujian sensor MQ-2 sensor yang biasa digunakan sebagai pendeteksi asap rokok di ruangan perkantoran atau industri, sangat cocok untuk mendeteksi asap rokok, LPG, iso-butane, propane, & LNG. Dapat menghindari gangguan dari alkohol, asap masakan [6].





Gambar 4. Sensor Asap Rokok MQ-2

Pada gambar 5, menerangkan flowchart dari alur program yang di buat. Bermula dari enditeksian sensor MQ-2, kondisi sensor MQ-2 dilihat apakah ada asap atau tidak , jika ada asap capture camera dan kirim via email.



Gambar 5. Flowchart Sistem Alat

Adapun listing program yang digunakan pada pengujian sensor MQ-2 ini adalah :

```

1 import RPi.GPIO as GPIO
2 import time
3 import os
4
5 #adjust for where your switch is connected
6 #buttonPin = 17
7 GPIO.setmode(GPIO.BCM)
8 GPIO.setup(17,GPIO.IN)
9
10 while True:
11     #assuming the script to call is long enough we can ignore bouncing
12     i=GPIO.input(17)
13     if i==0:
14         #this is the script that will be called (as root)
15         os.system("fswebcam -r 960x720 -d /dev/video0 /home/pi/emailgas/webcam.jpg")
16         os.system("python /home/pi/emailgas/sendnotify.py")
17         os.system("mpg123 /home/pi/emailgas/alarm.mp3")
    
```

Gambar 6. Listing program pengujian sensor asap rokok

E. Pengujian Email Terkirim

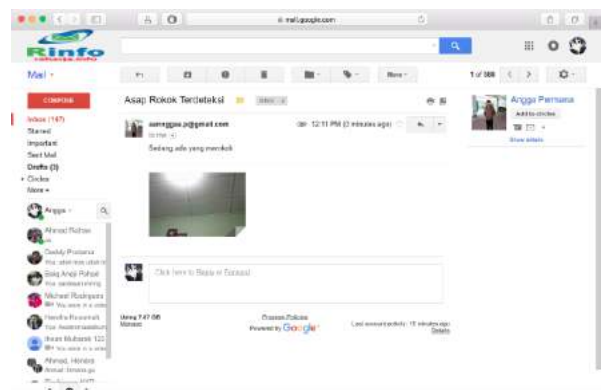
Pada pengujian email ini dimana photo akan di kirimkan dari raspberry yang akan di kirimkan otomatis ke email langsung ke server gmail. [4]

Adapun listing program yang digunakan pada pengujian Email ini adalah [9]:

```

#!/usr/bin/env python
1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.base import MIMEBase
4 from email.mime.text import MIMEText
5 from email.utils import COMMASPACE, formatdate
6 import os
7
8 USERNAME = "aanggga@gmail.com"
9 PASSWORD = "yourpassword"
10
11 def main():
12     msg = MIMEMultipart()
13     msg['from'] = USERNAME
14     msg['to'] = COMMASPACE.join(to)
15     msg['date'] = formatdate(localtime=True)
16     msg['subject'] = subject
17     msg.attach(MIMEText(text))
18
19     #file in file:
20     part = MIMEBase('application', 'octet-stream')
21     with open('doorbell.jpg', 'rb') as f:
22         data = f.read()
23         part.set_payload(data)
24         encoders.encode_base64(part)
25         part.add_header('Content-Disposition', 'attachment; filename="doorbell.jpg"')
26     msg.attach(part)
27
28     server = smtplib.SMTP('smtp.gmail.com:587')
29     server.starttls()
30     server.login(USERNAME, PASSWORD)
31     server.sendmail(USERNAME, to, msg.as_string())
32     server.quit()
33
34 sendMail(["aangga.permana@raharjo.info",
35         "doorbell.jpg",
36         "doorbell is ringing the doorbell, picture attached",
37         "/home/pi/emailgas/webcam.jpg"])
    
```

Gambar 7. Koding Email [1]



Gambar 8. Hasil Email Terkirim

## V. SIMPULAN

Prototipe ini dibuat menggunakan Raspberry pi yang dihubungkan dengan sensor MQ-2 sebagai media input nya. Sensor MQ-2 ini berfungsi mendeteksi Asap rokok, Gas dan sejenisnya. Dengan menggunakan sensor ini, maka Raspberry pidapat mendeteksi Asap rokok secara efektif. Pada Uji coba yang telah dilakukan pendeteksian optimal sensor MQ-2 ini adalah pada jarak 5 cm. Karena tingkat kepekaan asap pada jarak ini paling optimal.

Tabel 1. Hasil Pengujian Sensor Asap Rokok

Jarak (cm)	Nilai kepekatan Asap Rokok			Setatus sensor
	5(s)	10(s)	15(s)	
0 - 2 cm	300	330	360	Aktif
0 - 4 cm	338	365	372	Aktif
0 - 8 cm	232	260	300	Aktif
0 - 16 cm	157	242	286	Aktif
0 - 18 cm	-	-	-	Tidak Aktif

Keterangan :

1. Jarak (cm) : Jarak sensor dari sumber gas, dinyatakan dalam satuan sentimeter (cm).
2. Nilai Kepekatan Asap Rokok : Nilai kepekatan asap rokok dinyatakan dalam satuan ppm (part per milion), yang setiap ppm nya dihitung dalam kurun waktu 5 sekon, 10 sekon dan 15 sekon.

Perancangan alat ini menggunakan Raspberry pi, pc berukuran mini yang di dalamnya sudah tersedia wifi dan bluetooth berfungsi sebagai, koneksi ke internet sehingga alat ini dapat terhubung dengan internet mengirimkan informasi melalui server gmail. Kendala pada alat ini lebih kepada koneksi internet, karena data yang terdeteksi dari sensor MQ-2 dikirimkan melalui raspberry dan raspberry mengupload lagi ke server gmail. Raspberry pi yang telah diprogram yang didalamnya telah disisipkan

library perintah yang akan dikirim sebagai peringatan jika terdeteksi Asap rokok. Penggunaan dipilih karena sudah merupakan system mendekati ke system komputer, yang mempunyai internet protocol sehingga dapat dan mudah dikendalikan dari mana saja. Pemilihan sensor menggunakan sensor MQ-2, karena sensor ini dapat langsung menghasilkan sinyal analog yang dapat di proses pada pin.

## DAFTAR PUSTAKA

- [1] Adelia, dan Jimmy Setiawan. 2011. Implementasi Customer Relationship Management (CRM) pada Sistem Reservasi Hotel berbasis Website dan Desktop. Bandung: Universitas Kristen Maranatha. Vol. 6, No. 2, September 2011:113-126.
- [2] Agung, Fajri Septia and Farhan, M. 2013. Sistem Deteksi Asap Rokok Pada Ruangan Bebas Asap Rokok Dengan Keluaran Suara. STMIK MDP
- [3] Albahari, Fahad. 2013. Pendeteksi Asap Rokok Untuk Lingkungan Bebas Asap Rokok Berbasis Mikrokontroler Atmega32U4. Tangerang : Bina Am Ma'mur
- [4] Jarot s, Darma. 2009. Buku Pintar Menguasai Internet. Jakarta : Mediakita.
- [5] Malik, Ibnu dan Mohammad Unggul Juwana. 2010. ANEKA PROYEK Mikrokontroler PIC16F84/A. Jakarta: PT Elex Media Komputindo.
- [6] Mandagi, Albert dan Immanuel, Stehen. 2014. Penggunaan Sensor Gas Mq-2 Sebagai Pendeteksi Asap Rokok. Jurnal UKRIDA PRESS No. 9, Vol.3, January 2014
- [7] Nurdin, N (2011) Detektor Asap Rokok Dilengkapi Dengan Blower Otomatis Dan Suara Peringatan Berbasis Mikrokontroler Atmega8535. UNY.
- [8] Oscar. 2012. Elektronika Dasar: Pengenalan Praktis. Jakarta: Elek Media Komputindo.
- [9] Rizky. 2011. Konsep Dasar Rekayasa Perangkat Lunak. Jakarta: PT Prestasi Pustakaraya
- [10] Santoso, Ari Beni, Martinus dan Sugiyanto. 2013. Pembuatan Otomasi Pengaturan Ketera Api, Pengereman, dan Palang Pintu Pada Rel Kereta Api Mainan Berbasis Mikrokontroler. Jurnal FEMA Vol. 1, No. 1, Januari 2013
- [11] Fredy Susanto, Internet of Things Pada sistem keamanan ruangan ,studi kasus ruang server Perguruan Tinggi Raharja, SEMNASTEKNOMEDIA Vol.5 ,No.1,2017 Universitas AMIKOM Jogja
- [12] IOSR Journal of Electrical and Electronics Engineering (IOSRJEEE) e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 12, Issue 3 Ver. II (May – June 2017), PP 47-53 www.iosrjournals.org.

# Pengenalan Tulisan Tangan Offline Dengan Algoritma Generalized Hough Transform dan Backpropagation

Farica Perdana Putri<sup>1</sup>, Adhi Kusnadi<sup>2</sup>

Program Studi Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia  
farica@umn.ac.id  
adhi.kusnadi@umn.ac.id

Diterima 30 Maret 2018

Disetujui 8 Juni 2018

**Abstract**—Offline handwriting recognition is a technique used to recognize handwriting in paper document which converting it to digital form. Each handwriting has a unique style and shape that can be used to identify the owner. This research aims to develop a method to recognize the digital data handwriting. The method combines two algorithms; the first is Generalized Hough Transform in feature extraction process to detect arbitrary objects on the image; the second algorithm is Backpropagation to train the neural network based on feature values from feature extraction process. Artificial Neural Network (ANN) is used to improve the accuracy of the recognition system. The experiments are performed by using 100 handwriting images of 10 different people. The number of hidden units is defined through experiment to obtain optimal neural network. The experiment result shows that the recognition accuracy is up to 80%.

**Index Terms**—Artificial Neural Network, Backpropagation, Generalized Hough Transform, Offline handwriting recognition

## I. PENDAHULUAN

Tulisan tangan merupakan salah satu hal unik yang dapat dihasilkan oleh manusia selain tanda tangan. Seperti halnya tanda tangan, tulisan tangan juga dapat digunakan untuk mengidentifikasi pemiliknya. Beberapa penelitian pun telah memfokuskan pada teknik-teknik baru dan metode yang akan mengurangi waktu pemrosesan dan tingkat keakuratan yang lebih tinggi [1].

Pengenalan tulisan merupakan bagian dari pengenalan oleh komputer (*computer recognition*). Pengenalan oleh komputer merupakan proses otomatisasi yang menggunakan teknik pengenalan pola dan pembelajaran mesin untuk mengenal karakter atau kata yang diberikan [2]. Berdasarkan teknik yang digunakan untuk memperoleh input data, terdapat dua macam pengenalan tulisan tangan, yaitu *offline* dan *online*. Pada pengenalan tulisan *online*, data dikumpulkan ketika dibuat pada permukaan digital dengan menggunakan alat tulis seperti *stylus* [3]. Tetapi, pada pengenalan tulisan *offline*, data berupa

dokumen tulisan tangan yang dipindai terlebih dahulu [3]. Secara umum, terdapat lima tahapan dalam sistem pengenalan tulisan tangan *offline*, yaitu *image scanning*, *image preprocessing*, segmentasi, ekstraksi ciri, dan *neural network* [4].

Ada banyak algoritma yang dapat digunakan pada tahap ekstraksi ciri yang telah dikembangkan, salah satu diantaranya adalah *Generalized Hough Transform* (GHT). GHT merupakan pengembangan dari algoritma *Hough Transform* (HT). Berdasarkan rujukan [5] dan [6], GHT terbukti lebih efektif digunakan untuk mendeteksi bentuk yang berubah-ubah, seperti tulisan tangan, dibandingkan dengan HT. GHT adalah sebuah algoritma berbasis kontur yang hanya menggunakan informasi kontur untuk proses pengenalan pola, sehingga secara efektif dapat mengurangi pengaruh *noise* [7].

Setelah melalui ekstraksi ciri, ciri-ciri penting yang diperoleh digunakan pada tahap *neural network*. Jaringan syaraf tiruan (JST) atau *artificial neural network* digunakan untuk meningkatkan akurasi dalam mengenali tulisan tangan seseorang. Salah satu algoritma yang dapat digunakan untuk melatih jaringan syaraf adalah *Backpropagation*. Algoritma ini dapat meningkatkan kecepatan dari proses pembelajaran jaringan [4]. Penggunaan algoritma ini sesuai untuk aplikasi pengenalan pola dan merupakan salah satu algoritma pelatihan dengan supervisi yang dapat meningkatkan kinerja jaringan karena telah ditetapkan tujuan atau target dari pelatihan [8]. Penelitian ini bertujuan untuk mengimplementasikan algoritma ekstraksi ciri *Generalized Hough Transform* dan *Backpropagation* pada pengenalan tulisan tangan *offline*.

Pembahasan penelitian ini dibagi menjadi beberapa bagian. Analisis masalah dan penelitian dijelaskan pada bagian PENDAHULUAN. Landasan teori yang digunakan diuraikan pada bagian TELAAH LITERATUR. Perancangan dari arsitektur jaringan syaraf tiruan yang digunakan akan dijelaskan pada bagian PERANCANGAN ARSITEKTUR

JARINGAN SYARAF TIRUAN. Perancangan sistem pengenalan tulisan tangan dibahas pada bagian PERANCANGAN SISTEM. Uji coba dan hasil pengujian dibahas pada bagian IMPLEMENTASI APLIKASI dan UJI COBA. Adapun kesimpulan dan saran yang dapat digunakan untuk penelitian selanjutnya dijabarkan pada bagian SIMPULAN. Kemudian, saran yang dapat digunakan untuk penelitian selanjutnya diuraikan pada bagian SARAN.

## II. TELAHAH LITERATUR

### A. Pengenalan Tulisan Tangan Offline

Pengenalan tulisan tangan mengarah pada identifikasi karakter (huruf ataupun angka) yang tertulis [9]. Berdasarkan bagaimana input didapatkan, pengenalan tulisan tangan dibagi menjadi dua, yaitu pengenalan tulisan tangan *offline* dan pengenalan tulisan tangan *online* [4]. Pengenalan tulisan tangan *offline* merupakan pengenalan tulisan yang didasarkan pada citra dari sebuah teks tertulis [10].

Pengenalan tulisan tangan *offline* ini melibatkan konversi otomatis teks dalam citra ke dalam kode-kode huruf yang digunakan dalam komputer dan aplikasi pengolahan teks [10]. Data yang diperoleh dianggap sebagai representasi statis tulisan tangan. Tahapan dalam pengenalan tulisan tangan ini adalah [4]:

1. *Image Scanning*. Pada tahap ini, dokumen tertulis akan dipindai dengan menggunakan *scanner* dan kemudian citra tersebut akan menjadi input ke dalam aplikasi, khususnya tahap *image preprocessing*. Citra ini memiliki format tertentu seperti JPEG, BMP, dll.
2. *Image Preprocessing*. Keluaran dari tahap sebelumnya merupakan masukan bagi tahap *image preprocessing* [4]. Data mentah dari tulisan tangan akan diberikan sejumlah tahapan *preprocessing* untuk membuat data tersebut dapat berguna untuk tahapan selanjutnya [4]. Tahap *image preprocessing* bertujuan untuk mengekstraksi bagian-bagian tekstual, bagian yang terdapat tulisan atau karakter, yang relevan dan mempersiapkannya untuk tahap segmentasi dan ekstraksi ciri [4]. *Image preprocessing* pada sebuah citra diartikan sebagai penerapan beberapa prosedur seperti *thresholding*, penghalusan, penyaringan, pengubahan skala, dan normalisasi sehingga algoritma untuk klasifikasi akhir dapat lebih sederhana dan akurat.
3. Segmentasi. Sebuah citra ditampilkan dalam bentuk deretan karakter-karakter yang diuraikan ke dalam bagian-bagian citra dari individual karakter pada tahapan segmentasi [4].
4. Ekstraksi ciri. Ekstraksi ciri berkonsentrasi pada menemukan bentuk pada citra [11]. Setiap karakter memiliki ciri yang berbeda-beda, yang memainkan peranan penting dalam pengenalan

pola [4]. Tujuan utama dari ekstraksi ciri adalah untuk menyediakan informasi yang paling relevan dari data asli dan merepresentasikan informasi tersebut pada ruang dimensional yang lebih rendah [4].

5. *Neural Network*. Jaringan syaraf belajar dari citra pelatihan awal. Jaringan akan dilatih, kemudian dapat membuat identifikasi dari karakter. Setiap jaringan syaraf, secara unik mempelajari *properties* dan perbedaan dari citra-citra tersebut [10]. Selanjutnya, akan mencari *properties* yang sama pada citra untuk diidentifikasi [10]. *Backpropagation* merupakan algoritma yang digunakan untuk melatih jaringan syaraf. Jaringan dapat digunakan untuk mempelajari karakter pada pola dan melatih jaringan disimulasikan untuk pengenalan karakter yang ditampilkan dalam bentuk citra [4].

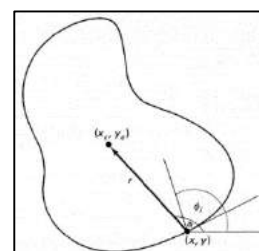
### B. Algoritma Generalized Hough Transform

*Hough Transform* merupakan salah satu algoritma yang dapat digunakan untuk ekstraksi ciri. Ide dasar dari *Hough Transform* adalah untuk memetakan titik tepi citra ke parameter ruang, yang mewakili semua instance dari ciri yang terlihat pada citra [6]. *Hough Transform* dapat mengekstrak bentuk – bentuk yang sederhana seperti garis dan bentuk dua dimensi lainnya, tetapi tidak untuk bentuk – bentuk yang lebih kompleks dan berubah-ubah [7]. Oleh karena itu, digunakanlah algoritma *Generalized Hough Transform*.

*Generalized Hough Transform* merupakan penemuan dari DH. Ballard. *Generalized Hough Transform* adalah perkembangan dari *Hough Transform* untuk mendeteksi obyek yang berubah-ubah pada citra. Semua bentuk tidak dapat secara mudah dinyatakan dengan sekelompok kecil parameter [6]. Solusi yang diusulkan kemudian untuk membuat tabel, dinamakan *R-table*, untuk menyimpan semua piksel sisi dari target bentuk yang diberikan [6].

Implementasi *Generalized Hough Transform* menggunakan informasi sisi untuk mendefinisikan pemetaan dari orientasi titik tepi ke titik referensi dari bentuk tersebut. Dalam menerapkan algoritma *Generalized Hough Transform* terdapat beberapa tahapan [12], yaitu:

1. Menetapkan titik acuan  $(x_c, y_c)$  yang dapat dihitung menggunakan rumus berikut dengan contoh obyek citra seperti pada Gambar 1 di bawah ini.





Gambar 1. Contoh Obyek Citra Untuk Implementasi GHT [12]

Rumus yang digunakan untuk menghitung titik acuan adalah berikut.

$$x_c = x_i + r_i \cos(\alpha_i) \quad (1)$$

$$y_c = y_i + r_i \sin(\alpha_i) \quad (2)$$

2. Menggambar garis dari titik acuan ke titik pada tepi obyek (P<sub>i</sub>) dan menghitung panjang dari r<sub>i</sub> dan sudut α<sub>i</sub>.
3. Menuliskan koordinat dari titik acuan (x<sub>c</sub>, y<sub>c</sub>) sebagai fungsi dari r<sub>i</sub> dan α<sub>i</sub>.
4. Menyimpan orientasi gradien (φ<sub>i</sub>) pada P<sub>i</sub>.
5. Membuat R-table dengan data-data di atas, dan sebagai indeks φ<sub>i</sub>.

Angle measured from figure boundary to reference point	Set of radii {r <sup>k</sup> } where r = (r, α)
φ <sub>1</sub>	r <sub>1</sub> <sup>1</sup> , r <sub>2</sub> <sup>1</sup> , ..., r <sub>n<sub>1</sub></sub> <sup>1</sup>
φ <sub>2</sub>	r <sub>1</sub> <sup>2</sup> , r <sub>2</sub> <sup>2</sup> , ..., r <sub>n<sub>2</sub></sub> <sup>2</sup>
⋮	⋮
⋮	⋮
φ <sub>m</sub>	r <sub>1</sub> <sup>m</sup> , r <sub>2</sub> <sup>m</sup> , ..., r <sub>n<sub>m</sub></sub> <sup>m</sup>

Gambar 2. Contoh R-table pada Generalized Hough Transform [12]

6. Menemukan versi kurva obyek setelah diputar, diperbesar/perkecil, dan ditranslasikan dengan membuat accumulator array dari titik acuan, diperbesar/perkecil dengan faktor S dan diputar dengan sudut θ. Untuk setiap tepi (x, y) pada citra, menghitung φ(x, y).

Kemudian untuk setiap (r, α) yang sesuai dengan φ(x, y) lakukan untuk setiap S dan θ:

$$x_c = x_i + r(f) S \cos[a(f) + q] \quad (3)$$

$$y_c = y_i + r(f) S \sin[a(f) + q] \quad (4)$$

$$A(x_c, y_c, S, q) = A(x_c, y_c, S, q) + 1 \quad (5)$$

Dan mencari maxima dari A.

### C. Algoritma Backpropagation

Algoritma Backpropagation merupakan salah satu algoritma pembelajaran yang bekerja pada jaringan multilayer feed-forward, menggunakan gradien turunan pada bobot untuk meminimalkan kesalahan keluaran [13]. Terdapat dua fase pada algoritma Backpropagation [14]:

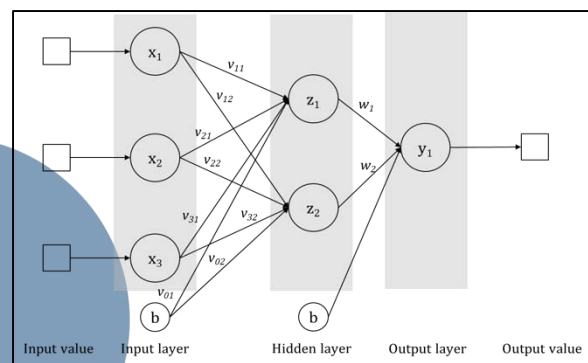
#### 1. Fase Forward

Fase ini merupakan tempat dimana fungsi aktivasi merambat dari input layer ke output layer [14]. Fungsi aktivasi yang digunakan adalah sigmoid biner dengan rumus sebagai berikut.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

#### 2. Fase Backward

Fase ini merupakan tempat dimana nilai aktual yang diamati dan nilai nominal yang diminta pada output layer dirambatkan mundur ke input layer sehingga bobot dapat dimodifikasi sampai nilai keluaran yang dihasilkan mendekati atau bahkan sama dengan diinginkan [14].



Gambar 3. Contoh Arsitektur Jaringan Syaraf Tiruan

Gambar 3. merupakan contoh dari arsitektur jaringan syaraf tiruan untuk Backpropagation. Tahapan dari Backpropagation adalah sebagai berikut.

1. Menginisialisasi nilai bobot dan bias dengan nilai secara acak yang bernilai antara [0,1] atau [-0,5, 0,5].
2. Menghitung keluaran dari hidden layer.

$$z_j = \frac{1}{1 + e^{v_{0j} + \sum_{i=1}^n v_{ij}x_i}} \quad (7)$$

z<sub>j</sub> = keluaran dari unit ke-j pada hidden layer

v<sub>0j</sub> = bobot bias pada input layer ke hidden unit ke-j

x<sub>i</sub> = masukkan pada unit ke-i

v<sub>ij</sub> = bobot dari input unit ke-i ke hidden layer ke-j

3. Menghitung keluaran dari output layer.

$$y_k = \frac{1}{1 + e^{w_{0k} + \sum_{j=1}^m w_{jk}z_j}} \quad (8)$$

y<sub>k</sub> = keluaran dari output unit ke-k

w<sub>jk</sub> = bobot dari hidden unit ke-j ke output unit ke-k

w<sub>0k</sub> = bobot bias pada hidden layer ke output unit ke-k

4. Menghitung kesalahan untuk unit pada output layer.

$$\delta_k = (t_k - y_k) y_k (1 - y_k) \quad (9)$$

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (10)$$

$$\Delta w_{0k} = \alpha \delta_k \quad (11)$$

$\delta_k$  = sinyal *error* untuk *output* unit ke- $k$  pada *output layer*

$\alpha$  = laju pembelajaran (*learning rate*)

5. Menghitung kesalahan untuk unit pada *hidden layer*.

$$\delta_j = \sum_{k=1}^m w_{jk} \delta_k z_j (1 - z_j) \quad (12)$$

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (13)$$

$$\Delta v_{0j} = \alpha \delta_j \quad (14)$$

$\delta_j$  = sinyal *error* untuk *output* unit ke- $j$  pada *hidden layer*

6. Memperbaharui bobot pada *output layer*.

$$w_{jk}^{new} = w_{jk}^{old} + \Delta w_{jk} \quad (15)$$

7. Memperbaharui bobot pada *hidden layer*.

$$v_{ij}^{new} = v_{ij}^{old} + \Delta v_{ij} \quad (16)$$

8. Tes kondisi penghentian

### III. PERANCANGAN ARSITEKTUR JARINGAN SYARAF TIRUAN

Perancangan arsitektur JST yang akan dilakukan terdiri dari dua tahap, yaitu pada tahap pelatihan dan tahap pengenalan. Pada tahap pelatihan JST, algoritma *Backpropagation* digunakan untuk mendapatkan bobot yang nantinya disimpan ke dalam *database*. Jaringan akan terus mengubah nilai bobot sampai kriteria tertentu terpenuhi, dalam aplikasi ini adalah nilai kesalahan yang lebih kecil dari toleransi kesalahan.

Tahap selanjutnya adalah tahap pengenalan, aliran pemrosesan dari jaringan hanya dapat bergerak maju, berbeda dengan tahap pelatihan. Bobot yang disimpan dalam *database* akan diambil dan kemudian diberikan kepada JST sebagai bobot untuk pengenalan sehingga didapatkan hasil keluaran.

Kedua tahap di atas, membutuhkan masukan yang didapatkan dari hasil ekstraksi ciri menggunakan algoritma GHT. Jika ingin menggunakan fungsi aktivasi sigmoid biner, data harus ditransformasikan dulu karena rentang keluaran fungsi aktivasi sigmoid adalah  $[0,1]$  [8]. Normalisasi dilakukan dengan menggunakan rumus normalisasi sebagai berikut [8].

$$x' = \frac{0.8(x - \text{nilai terkecil})}{\text{nilai terbesar} - \text{nilai terkecil}} + 0.1 \quad (17)$$

Adapun arsitektur JST yang digunakan diuraikan sebagai berikut.

1. Jumlah masukan pada *input layer* ditetapkan adalah 40 buah masukan. GHT menghasilkan 10 titik ciri dari tiap karakter. Banyak masukan

pada jaringan syaraf tiruan didapatkan dari proses ekstraksi ciri dengan GHT (10 titik ciri) dikalikan dengan 4 huruf yang terdapat pada *input* citra.

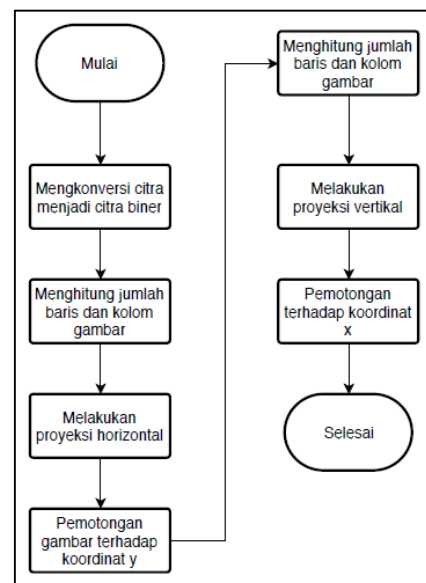
2. *Hidden layer* berjumlah 1 dan jumlah unit pada *hidden layer* akan ditentukan setelah proses pencarian jaringan optimum dengan menghitung nilai *root-mean-square error* (RMSE).
3. Antar *layer* terdapat koneksi satu arah yang memiliki bobot dengan nilai awal dipilih secara acak antara  $-0,5 - 0,5$ .
4. Jumlah keluaran pada *output layer* adalah 1 dengan nilai target yaitu 1.

### IV. PERANCANGAN SISTEM

Sistem pengenalan tulisan tangan *offline* menggunakan algoritma GHT pada tahap ekstraksi ciri dan *Backpropagation* pada tahap *neural network classification* agar dapat mengenali tulisan tangan seseorang pada citra digital. Ciri-ciri yang didapatkan dari tahap ekstraksi ciri menggunakan GHT menjadi *input* untuk jaringan syaraf tiruan pada tahap *neural network classification*.

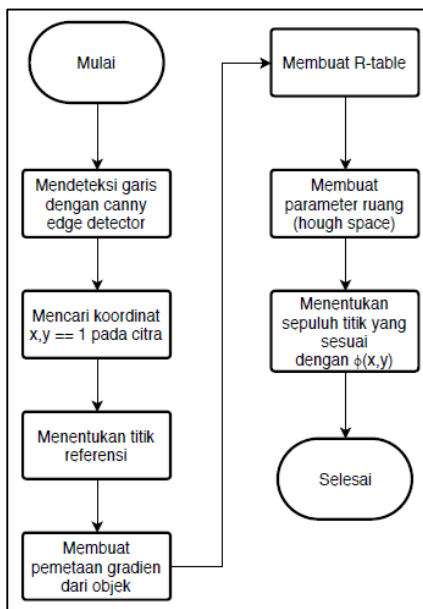
Pada perancangan sistem, digunakan *flowchart* untuk menggambarkan alur proses dan fungsi-fungsi pada sistem. *Flowchart* yang diberikan menggambarkan fungsi segmentasi tulisan tangan, ekstraksi ciri menggunakan GHT, pelatihan JST menggunakan *Backpropagation*, dan pengenalan tulisan tangan.

Gambar 4 merupakan alur dari proses segmentasi tulisan tangan yang dilakukan saat melakukan proses pelatihan ataupun pengenalan JST. Masukkan untuk proses segmentasi merupakan hasil keluaran dari proses *binarization* dari citra yang diberikan oleh *user*. Proses segmentasi adalah sebuah proses untuk memecah karakter atau huruf-huruf dari kata pada citra.

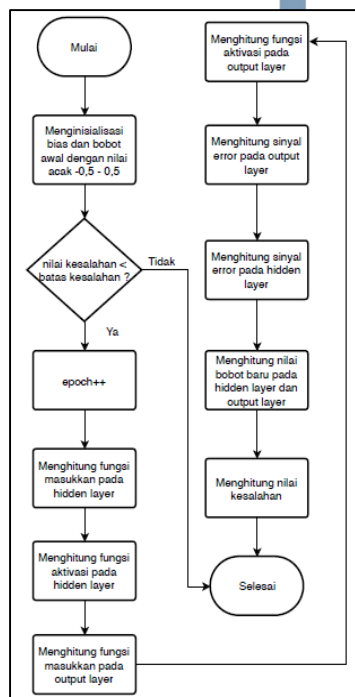


Gambar 4. Flowchart Proses Segmentasi Tulisan Tangan

Setelah proses segmentasi selesai, citra yang telah disegmentasi akan melalui tahap ekstraksi ciri menggunakan GHT. Proses ekstraksi ciri digambarkan pada Gambar 5. Algoritma GHT melakukan pendeteksian garis menggunakan algoritma *canny edge detector* sebelum pemrosesan GHT dilakukan.



Gambar 5. Flowchart Algoritma Generalized Hough Transform



Gambar 6. Flowchart

Hasil keluaran dari proses ekstraksi ciri digunakan sebagai masukan bagi JST pada tahap pelatihan ataupun pengenalan. Gambar 6 menggambarkan alur dari proses pelatihan JST menggunakan algoritma *Backpropagation*.

Selama proses pelatihan JST, bobot pada jaringan akan terus diperbaharui sampai nilai kesalahan lebih kecil dibandingkan dengan batas kesalahan, yaitu 0,00001. Nilai kesalahan dihitung menggunakan rumus sebagai berikut.

$$E = \frac{1}{2} \sum_i (t_i - o_i)^2 \quad (18)$$

$E$  = kesalahan jaringan

$t_i$  = target

$o_i$  = keluaran / output

Proses pengenalan tulisan tangan yang dilakukan hanya proses *feed-forward* atau maju, tidak ada perubahan bobot pada proses ini. Penghitungan yang dilakukan adalah fungsi masukan dan aktivasi dari masing-masing unit pada *hidden layer* dan *output layer*.

## V. UJI COBA

### A. Persiapan Pengujian

Sebelum tahap pengujian tulisan tangan, data yang digunakan untuk pelatihan dan pengujian dikumpulkan terlebih dahulu. Tulisan tangan yang digunakan merupakan tulisan cetak dan terdiri dari huruf latin. Data tersebut merupakan tulisan tangan dari 10 orang yang berbeda dan dipindai menjadi citra untuk diproses menggunakan teknik *image processing*. Setiap orang akan menulis sebanyak 10 kali kata “Saya” seperti contoh pada Gambar 7. Data citra dikelompokkan berdasarkan kepemilikan seperti diuraikan pada Tabel 1.



Gambar 7. Contoh Data Citra Tulisan Tangan

Tabel 1. Penamaan dan Pengelompokkan Data Citra Tulisan Tangan

Tulisan Tangan Orang ke -	Penamaan Data Citra
1	TT_1 sampai TT_10
2	TT_11 sampai TT_20
3	TT_21 sampai TT_30
4	TT_31 sampai TT_40
5	TT_41 sampai TT_50
6	TT_51 sampai TT_60
7	TT_61 sampai TT_70
8	TT_71 sampai TT_80
9	TT_81 sampai TT_90
10	TT_91 sampai TT_100

Kemudian, melakukan pemilihan arsitektur jaringan syaraf optimum yang akan digunakan pada tahap pelatihan dan pengenalan tulisan tangan. Pada tahap ini, arsitektur jaringan syaraf optimum, ditentukan menggunakan 40 masukan, 1 *hidden layer*, laju pembelajaran (*learning rate*) 0,5, batas kesalahan 0,00001, fungsi aktivasi yang digunakan adalah fungsi sigmoid biner dengan satu keluaran, dan target keluaran adalah 1. Sedangkan, jumlah unit pada *hidden layer* yang akan diuji coba adalah 2 sampai 10. Perubahan bobot akan terus berlangsung sampai nilai kesalahan lebih kecil dari batas kesalahan yang ditetapkan.

Tabel 2. Hasil Pengujian terhadap Jumlah Unit pada *Hidden Layer (Hidden Unit)*

Parameter	Pengujian				
	1	2	3	4	5
Laju Pembelajaran	0,5	0,5	0,5	0,5	0,5
Hidden Unit	2	3	4	5	6
Batas Kesalahan	$10^{-5}$	$10^{-5}$	$10^{-5}$	$10^{-5}$	$10^{-5}$
RMSE	7,24	7,23	7,19	<b>7,18</b>	7,27

Parameter	Pengujian			
	6	7	8	9
Laju Pembelajaran	0,5	0,5	0,5	0,5
Hidden Unit	7	8	9	10
Batas Kesalahan	$10^{-5}$	$10^{-5}$	$10^{-5}$	$10^{-5}$
RMSE	7,38	7,40	7,35	7,28

Tabel 2 merupakan hasil uji coba untuk menentukan jumlah unit pada *hidden layer* dengan menghitung nilai RMSE. Berdasarkan data pada Tabel 2, nilai RMSE

terkecil didapatkan saat jumlah unit pada *hidden layer* sebanyak 4 buah, yaitu 7,18. Ini dapat diartikan jumlah unit pada *hidden layer* untuk arsitektur JST yang optimum dengan laju pembelajaran 0,5, batas kesalahan 0,00001, 40 unit masukan, 1 *hidden layer*, dan 1 keluaran adalah 4 buah.

Selanjutnya, tahap pelatihan jaringan syaraf tiruan menggunakan 20 tulisan tangan dari 10 orang yang berbeda. Arsitektur jaringan yang digunakan sesuai dengan arsitektur jaringan yang telah diuji sebelumnya.

Tabel 3. Hasil Pelatihan Jaringan Syaraf Tiruan

Data Tulisan Tangan Orang ke-	Keluaran	E
1	0,99552831	9,99801E-06
2	0,99631147	6,80263E-06
3	0,99615998	7,37288E-06
4	0,99577645	8,91919E-06
5	0,99580301	8,80736E-06
6	0,99576086	8,98515E-06
7	0,99562138	9,58616E-06
8	0,99561736	9,60377E-06
9	0,99599469	8,02125E-06
10	0,99618233	7,2873E-06

Seperti yang ditunjukkan pada Tabel 3, nilai kesalahan (E) yang dihasilkan dari data pelatihan sudah lebih kecil dibandingkan dengan batas kesalahan, sehingga proses pelatihan dihentikan dan dapat dilakukan proses pengujian dengan bobot yang didapatkan dari hasil pelatihan tersebut.

#### B. Hasil Pengujian

Pada pengujian ini, dilihat tingkat akurasi aplikasi dalam melakukan pengenalan tulisan tangan. Pengujian dilakukan dengan:

1. Memasukkan citra tulisan tangan yang akan diuji dan memilih pemilik dari tulisan tangan yang akan diuji keasliannya sesuai yang ada pada data di *database*.
2. Sistem akan melakukan proses pengenalan tulisan tangan dimulai dari image processing, algoritma GHT, dan *feed-forward neural network*.
3. Mencatat hasil uji coba yang dilakukan ke dalam tabel, seperti pada Tabel 4.



$$\text{Akurasi} = \frac{\text{jumlah pengenalan benar}}{\text{total citra uji coba}} \times 100\% \quad (13)$$

Dari persamaan di atas dan data uji coba yang diberikan pada Tabel 3, maka persentase keberhasilannya adalah

$$\text{Akurasi} = \frac{16}{20} \times 100\% = 80\%$$

Tabel 4. Hasil Pengujian Pengenalan Tulisan Tangan

Uji Coba ke -	Tulisan Tangan 1	Tulisan Tangan 2	Keluaran	Hasil
1	TT_3	TT_1	0.99597653	Cocok
2	TT_12	TT_11	0.99602671	Cocok
3	TT_27	TT_21	0.49510268	Tidak cocok
4	TT_33	TT_31	0.99594956	Cocok
5	TT_42	TT_41	0.99589848	Cocok
6	TT_56	TT_51	0.99584048	Cocok
7	TT_64	TT_61	0.99576848	Cocok
8	TT_78	TT_71	0.49497389	Tidak cocok
9	TT_89	TT_81	0.99587663	Cocok
10	TT_100	TT_91	0.99591680	Cocok
11	TT_6	TT_70	0.99563015	Cocok
12	TT_12	TT_65	0.39472854	Tidak cocok
13	TT_13	TT_8	0.35531605	Tidak cocok
14	TT_14	TT_8	0.48539911	Tidak cocok
15	TT_15	TT_4	0.35516635	Tidak cocok
16	TT_16	TT_6	0.69602320	Cocok
17	TT_11	TT_5	0.46524995	Tidak cocok
18	TT_13	TT_3	0.48493201	Tidak cocok
19	TT_14	TT_7	0.79563189	Cocok
20	TT_15	TT_10	0.29539982	Tidak cocok

Pada Tabel 4 di atas, Tulisan Tangan 1 merupakan tulisan tangan yang akan diuji, Tulisan Tangan 2 merupakan tulisan tangan yang ada pada *database* dan sudah dilatih. Keluaran merupakan nilai keluaran dari jaringan syaraf pada tahap pengujian, dan hasil merupakan hasil apakah Tulisan Tangan 1 sesuai atau cocok dengan Tulisan Tangan 2 atau tidak.

### C. Pembahasan

Berdasarkan pada pengujian yang telah dilakukan, dengan hasil yang telah diberikan sebelumnya, dapat ditentukan persentase akurasi dengan penghitungan seperti di bawah ini.

Persentase keberhasilan dari aplikasi pengenalan tulisan tangan *offline* menggunakan algoritma *Generalized Hough Transform* dan *Backpropagation* adalah 80%.

Persentase akurasi yang dihasilkan sudah cukup baik. Menurut hasil pelatihan jaringan syaraf tiruan pada Tabel 2, dapat disimpulkan bahwa proses pelatihan dengan arsitektur JST yang ada sudah baik dikarenakan nilai kesalahan sudah sangat kecil. Namun, proses pelatihan JST dapat ditingkatkan lagi dengan melakukan uji coba jumlah unit pada *hidden layer* sebesar 11 sampai dengan jumlah unit masukkan dikurangi 1.

## VI. SIMPULAN DAN SARAN

Sistem pengenalan tulisan tangan dengan menggabungkan algoritma *Generalized Hough Transform* dan *Backpropagation* telah berhasil dilakukan. Algoritma *Generalized Hough Transform* digunakan untuk memperoleh ciri dari citra tulisan tangan pada tahap ekstraksi ciri. *Backpropagation* digunakan untuk melatih jaringan syaraf tiruan dalam mengenali citra tulisan tangan.

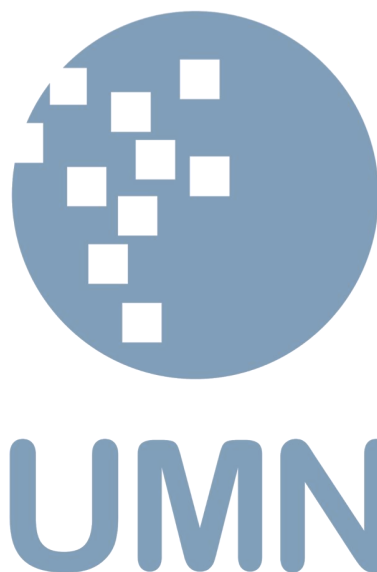
Sistem telah diuji dengan menggunakan model arsitektur jaringan berupa 40 buah masukkan, 1 *hidden layer* (dengan 4 buah unit), 1 keluaran, laju pembelajaran 0.5 dan batas kesalahan 0.00001. Fungsi aktivasi yang digunakan pada layar tersembunyi dan layar keluaran adalah fungsi sigmoid biner. Data yang digunakan berjumlah 100 dari 10 orang yang berbeda. Dari hasil uji coba yang dilakukan, didapatkan bahwa penggabungan algoritma *Generalized Hough Transform* dan *Backpropagation* dalam pengenalan tulisan tangan *offline* menghasilkan nilai persentase akurasi sebesar 80%.

Adapun saran untuk penelitian selanjutnya yaitu perlu dilakukan penelitian dengan menggunakan algoritma atau metode lain untuk proses pelatihan jaringan syaraf maupun ekstraksi ciri. Sistem yang dikembangkan pada kesempatan kali ini belum dapat menganalisis citra tulisan tangan yang berbentuk sambung. Oleh karena itu, penelitian selanjutnya diharapkan dapat menganalisis tulisan tangan yang berbentuk latin ataupun huruf lain seperti aksara Arab, Mandarin, Jepang, dan lain sebagainya.

### DAFTAR PUSTAKA

- [1] Althaf, M. K. M. dan M. B. Begum, "Handwritten Characters Pattern Recognition Using Neural Networks," International Conference on Computing and Control Engineering, vol. 4, no. 7, 2012.

- [2] Liu, Zhi-Qiang, Jin-Hai Cai, dan Richard Buse, "Introduction" di dalam *Handwriting Recognition: Soft Computing and Probabilistic Approach*, edisi 1, Berlin: Springer, 2003, bab I.
- [3] Suryani, Dewi, Patrick Doesch, dan Hermann Ney, "On the Benefits of Convolutional Neural Network Combinations in Offline Handwriting Recognition," International Conference on Frontiers in Handwriting Recognition (ICHFR), Shenzhen, hal. 193-198, 2016.
- [4] Kumar, Ashok dan P. K. Bhatia, "Offline Handwritten Character Recognition Using Improved Back-Propagation Algorithm," International Journal of Advances in Engineering Sciences, vol. 3, no. 3, 2013.
- [5] Ballard, D. H., "Generalizing The Hough Transform to Detect Arbitrary Shapes," Pattern Recognition, vol. 13, no. 2, hal. 111-122, 1981.
- [6] Touj, S., Amara, N. B., & Amiri, H., "Generalized Hough Transform for Arabic Printed Optical Character Recognition," The International Arab Journal of Information Technology, vol. 2, no. 4, hal. 326-333, 2005.
- [7] Qian, Lisi dan Bin Fang, "Optimized Generalized Hough Transform for Road Making Recognition Application," SPAC, Shenzhen, hal. 361-367, 2017.
- [8] Siang, J. J., "Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab," Yogyakarta: Penerbit Andi, 2009.
- [9] Kala, R., H. Vazirani, A. Shukla, dan R. Tiwari, "Offline Handwriting Recognition using Genetic Algorithm," IJCSI, vol. 7, no. 1, hal. 16-25, March 2010.
- [10] Chakravarthy, A. S. N., P. V. K. Raja, dan P. S. Avadhani, "Handwritten Text Image Authentication Using Back Propagation," IJNSA, vol. 3, no. 5, hal 121-130, Sep 2011.
- [11] Nixon, Mark dan A. Aguado, "Introduction" di dalam *Feature Extraction and Image Processing*, edisi 2, United Kingdom: Elsevier, 2008, bab I.
- [12] Ballard, D. H. dan C. M. Brown, "The Hough Method for Curve Detection" di dalam Computer Vision, edisi 1, New Jersey: Prentice-Hall, hal. 128-131, 1982, bab IV.
- [13] Russel, Stuart J. dan Peter Norvig, "Artificial Neural Network" di dalam *Artificial Intelligence: A Modern Approach*, edisi 3, New Jersey: Prentice-Hall, hal. 733, 2010, bab XVIII.
- [14] Lagudu, S. dan CH. V. Sarma, "Hand Writing Recognition Using Hybrid Particle Swarm Optimization & Back Propagation Algorithm," IJAIEM, vol. 2, no. 1, hal. 75-81, 2013.



# Implementasi Metode Entropy dan Topsis Dalam Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik

(Studi Kasus : Jakarta Smart City)

Mitha Anggreani Rupang<sup>1</sup>, Adhi Kusnadi<sup>2</sup>

Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia

mitha.anggreani@student.umn.ac.id

adhi.kusnadi@umn.ac.id

Diterima 20 April 2018

Disetujui 8 Juni 2018

**Abstract**—Employee is a part of the company's most important asset in its efforts to maintain survival, growth, ability to compete and profit. At this time the process of assessment of employees in Jakarta Smart City is still in the form of manual and the decision only from one party only, so the process is still not accurate. So it takes the methods that must be able to replace that system. For that reason, a Decision Support System (SPK) was created to determine the best employees in Jakarta Smart City. In the system implemented the method of Entropy and TOPSIS. Entropy method can be trusted in determining the weight of the criteria to be used. And TOPSIS method can quickly perform the ranking process. Criteria to be used are quality and quantity of work, obedience, cooperation, morale, and work discipline. The index of satisfaction level of respondents to the decision support system ranges from 70% -80%, meaning that the assessment of the system created gives results at a fairly good level.

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

## I. PENDAHULUAN

Karyawan merupakan aset terpenting yang dimiliki oleh perusahaan, dalam usahanya mempertahankan kelangsungan hidup, berkembang, bersaing serta mendapatkan laba. Kualitas karyawan yang baik, dapat meningkatkan produktivitas perusahaan. Agar kualitas para karyawan terjaga dan meningkat dengan baik, perusahaan perlu melakukan suatu penilaian kinerja pegawai [1].

Hal tersebut belaku juga untuk Jakarta *Smart City* (JSC). Sebuah perusahaan di bawah pemerintah DKI Jakarta, yang menggunakan konsep *smart city* untuk mengoptimalkan penggunaan teknologi Informasi dan Komunikasi (TIK). Bermanfaat untuk mengetahui, memahami, dan mengontrol sumber daya di suatu kota dengan lebih efektif dan efisien, sehingga dapat memaksimalkan pelayanan publik, menyediakan solusi untuk masalah, dan mendukung pembangunan yang berkelanjutan.

Dengan bertambahnya karyawan maka keanekaragaman karyawan juga semakin kompleks sehingga lebih sulit memilih karyawan terbaik. Pada saat ini proses penilaian kinerja karyawan masih dalam bentuk manual dan keputusan hanya dari satu pihak saja, yaitu proyek manager. Sehingga proses yang dilakukan masih belum akurat, bias saja diterperensi oleh pihak lain. Oleh karena itu perlu dibuat aplikasi sistem yang dapat memberikan hasil yang akurat dan independent. Penilaian tidak hanya oleh satu pihak, tapi oleh beberapa pihak yang berkepentingan terhadap karyawan tersebut seperti beberapa kepala unit.

Pengambilan keputusan terkait dengan pemilihan produk dengan banyak kriteria dapat menggunakan metode- metode yang banyak digunakan, seperti AHP, ELECTRE, TOPSIS, ENTROPY, SAW, dan FUZZY [2]. Seperti pada penelitian [2], untuk sistem pemilihan handphone menggunakan SAW.

Penelitian yang terkait dengan pengambilan keputusan tersebut diantaranya adalah penelitian yang dilakukan oleh Handoyo dkk. (2014) menunjukkan bahwa SPK (Sistem Pendukung Keputusan) dengan metode *entropy* dapat melakukan proses seleksi dengan cepat dan cukup *powerful* untuk menghitung suatu bobot kriteri karena metode ini bisa digunakan untuk berbagai jenis data, baik kuantitatif (angka) maupun kualitatif (kata) [3].

Sementara penelitian yang dilakukan oleh Ramirez dkk. (2010) menunjukkan bahwa TOPSIS dapat dijadikan metode SPK (Sistem Pendukung Keputusan) yang dapat memberikan alternatif pilihan secara cepat dan menghasilkan nilai keputusan yang sangat tinggi sehingga memudahkan *decision maker* dalam mengambil keputusan [4]. Pada penelitian [5], perbandingan hasil analisis metode SAW dan TOPSIS tersebut menunjukkan TOPSIS memberikan hasil yang lebih baik untuk narasumber tertentu.

Berdasarkan penelitian di atas, maka dibuatlah sistem pendukung keputusan yang diharapkan berfungsi untuk membantu pihak JSC (Jakarta Smart City) untuk melakukan penilaian kinerja setiap karyawan berdasarkan *ranking* yang dihitung menggunakan metode *entropy* dan TOPSIS. Metode Entropy digunakan untuk menghitung bobot kriteria, dan TOPSIS digunakan untuk menentukan ranking dari karyawan.

## II. TINJAUAN PUSTAKA

### A. Metode Entropi

Dalam metode pembobotan, *entropy* dapat diaplikasikan untuk pembobotan atribut-atribut, hal ini dilakukan oleh Hwang dan Yoon (1981). Menggunakan metode *entropy*, kriteria dengan variasi nilai tertinggi akan mendapatkan bobot tertinggi. Vivi Triyanti dan M. T. Gadis menyimpulkan, metode *entropy* cukup powerful untuk menghitung bobot suatu kriteria. [6].

Metode penilaian untuk rating kriteria menggunakan bilangan bulat ganjil antara 1-10 dengan *range scope* yaitu 1, 3, 5, 7, dan 9. Angka tersebut menunjukkan tingkat kepentingan tiap kriteria, nilai 1 menunjukkan sangat tidak penting sampai angka 9 menunjukkan sangat penting. Skala penilaian tersebut merupakan skala perbandingan yang umumnya sering digunakan dalam penelitian atribut kualitatif yang selalu subjektif [3].

Berikut adalah langkah-langkah dalam metode *entropy* [6].

#### 1. Membuat tabel data rating kriteria

Tabel data rating kinerja adalah nilai alternatif pada setiap kriteria dimana setiap kriteria tidak saling bergantung satu dengan lainnya

#### 2. Normalisasi tabel data kriteria

Rumus Normalisasi adalah sebagai berikut.

$$d_i^k = \frac{x_i^k}{x_{i_{maks}}^k} \quad (1)$$

$$D_k = \sum_{k=1}^m d_i^k \quad k = 1, 2, \dots, m$$

Dimana :

$d_i^k$  = nilai data yang telah dinormalisasi

$x_i^k$  = nilai data yang belum dinormalisasi

$x_{i_{maks}}^k$  = nilai data yang belum dinormalisasi yang mempunyai nilai paling tinggi

$D_k$  = jumlah nilai data yang telah dinormalisasi

#### 3. Perhitungan Entropy

Langkah selanjutnya adalah pengukuran *entropy* untuk setiap atribut ke-k dengan terlebih dahulu mencari  $e_{max}$  pada rumus 2 dan K pada rumus 3, Rumusnya adalah :

$$e_{max} = \ln m \quad (2)$$

Dimana :  $m$  = jumlah alternatif

$$K = \frac{1}{e_{max}} \quad (3)$$

Perhitungan *entropy* untuk setiap kriteria ke-k ditunjukkan pada rumus 2.5.

$$e(d_k) = -K \sum_{k=1}^m \frac{d_i^k}{D_k} \ln \frac{d_i^k}{D_k}, \quad K > 0 \quad (4)$$

Dimana :

$e(d_k)$  = nilai *entropy* pada setiap kriteria ( $k=1, 2, \dots, m$ )

$d_i^k$  = nilai data yang telah dinormalisasi

$D_k$  = jumlah nilai data yang telah dinormalisasi

$m$  = jumlah alternatif

Setelah mendapat  $e(d_i)$  untuk masing-masing atribut, maka dapat ditentukan total *entropy* untuk masing-masing atribut, rumusnya adalah :

$$E = \sum_{k=1}^n e(d_k) \quad (5)$$

Dimana :

$e(d_k)$  = nilai *entropy* pada setiap kriteria ( $k=1,2,\dots,n$ )

$E$  = total *entropy*

#### 4. Perhitungan Bobot Entropy

Langkah berikutnya adalah menghitung bobot dengan menggunakan rumus sebagai berikut :

$$\bar{\lambda}_k = \frac{1}{n - E} [1 - e(d_k)], \quad 0 \leq \lambda \leq 1 \quad (6)$$

$$\sum_{k=1}^n \lambda = \pm 1$$

Dimana :

$e(d_k)$  = nilai *entropy* pada setiap kriteria ( $k=1,2,\dots,n$ )

$E$  = total *entropy*

$n$  = jumlah kriteria

$\bar{\lambda}_k$  = bobot *entropy*

Setelah mendapatkan bobot *entropy* untuk masing-masing kriteria jika sebelumnya telah ada bobot awal atau bobot yang telah ditentukan sebelumnya, maka hasil bobot *entropy* yang sebenarnya untuk tiap kriteria akan didapat dengan perhitungan berikut ini:

$$\lambda_k = \frac{\bar{\lambda}_k \times w_k}{\sum_{i=1}^n \bar{\lambda}_k \times w_k} \quad k = 1, \dots, n \quad (7)$$

Dimana :

$e(d_k)$  = nilai *entropy* pada setiap kriteria ( $k=1,2,\dots,n$ )

$E$  = total *entropy*

$n$  = jumlah kriteria

$\bar{\lambda}_k$  = bobot *entropy*

$\lambda_k$  = bobot *entropy* akhir

### C. Metode TOPSIS

TOPSIS adalah salah satu metode pengambilan keputusan multikriteria atau alternatif pilihan yang merupakan alternatif yang mempunyai jarak terkecil dari solusi ideal positif dan jarak terbesar dari solusi



ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean. Namun, alternatif yang mempunyai jarak terkecil dari solusi ideal positif, tidak harus mempunyai jarak terbesar dari solusi ideal negatif. Maka dari itu, TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif secara bersamaan. Solusi optimal dalam metode TOPSIS didapat dengan menentukan kedekatan relatif suatu alternatif terhadap solusi ideal positif. TOPSIS akan meranking alternatif berdasarkan prioritas nilai kedekatan relatif suatu alternatif terhadap solusi ideal positif. Alternatif-alternatif yang telah diranking kemudian dijadikan sebagai referensi bagi pengambil keputusan untuk memilih solusi terbaik yang diinginkan.

Berikut adalah langkah-langkah dalam metode TOPSIS [6].

1. Menentukan matriks rating kinerja

Matrik rating kinerja adalah nilai alternatif  $A_i$  ( $i=1,2,\dots,m$ ) pada setiap kriteria  $C_j$  ( $j=1,2,\dots,n$ ) dimana setiap kriteria tidak saling bergantung satu dengan yang lainnya

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (8)$$

Dimana :

$i = 1,2,\dots,m$

$j = 1,2,\dots,n$

$r_{ij}$  adalah elemen dari matriks keputusan yang ternormalisasi  $r$

$x_{ij}$  adalah elemen dari matriks keputusan  $x$

3. Menghitung matriks ternormalisasi terbobot

$$y_{ij} = w_j \cdot r_{ij} \quad (9)$$

Dengan  $i = 1,2,3,\dots,m$ , dan  $j=1,2,3,\dots,n$ ,

Dimana :

$y_{ij}$  = elemen dari matriks keputusan yang ternormalisasi terbobot  $y$

$w_j$  = bobot dari kriteria ke- $j$

$r_{ij}$  = elemen dari matriks keputusan yang ternormalisasi  $r_{ij}$

4. Menghitung matriks solusi ideal positif dan matriks solusi ideal negatif

$$y_j^+ = \begin{cases} \max\{y_{ij}\} \\ i \\ \min\{y_{ij}\} \\ i \end{cases} \quad (10)$$

$$y_j^- = \begin{cases} \min\{y_{ij}\} \\ i \\ \max\{y_{ij}\} \\ i \end{cases}$$

Dimana :

$y_j^+$  = max, dimana  $j$  adalah kriteria keuntungan (*benefit*)

$y_j^+$  = min, dimana  $j$  adalah kriteria biaya (*cost*)

$y_j^-$  = min, dimana  $j$  adalah kriteria keuntungan (*benefit*)

$y_j^-$  = max, dimana  $j$  adalah kriteria biaya (*cost*)

Berdasar persamaan 12 dan 13, selanjutnya dicari nilai solusi ideal positif ( $A^+$ ) dan solusi ideal negatif ( $A^-$ ) dengan persamaan 14 dan 15.

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+)$$

$$A^- = (y_1^-, y_2^-, \dots, y_n^-) \quad (11)$$

Dimana  $j = 1,2,\dots,n$  ( $n$  adalah indeks kriteria)

5. Menghitung jarak antara nilai setiap alternatif dengan solusi ideal positif ( $D^+$ ) dan jarak antara nilai setiap alternatif dengan solusi ideal negatif ( $D^-$ )

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_j^-)^2} \quad i=1,2,\dots,m$$

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_j^+ - y_{ij})^2} \quad i=1,2,\dots,m$$

Dimana :

$D_i^+$  = jarak alternatif ke- $i$  dari solusi ideal positif.

$D_i^-$  = jarak alternatif ke- $i$  dari solusi ideal negatif.

$y_{ij}$  = elemen dari matriks keputusan yang ternormalisasi terbobot  $y$ .

$y_j^+$  = elemen matriks solusi ideal positif.

$y_j^-$  = elemen matriks solusi ideal negatif.

6. Menghitung nilai prefensi untuk setiap karyawan terbaik ( $V_i$ ).

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \quad i=1,2,\dots,m$$

Dimana :

$v_i$  = kedekatan relatif dari alternatif ke- $i$  terhadap solusi ideal positif.

$D_i^+$  = jarak alternatif ke- $i$  dari solusi ideal positif.

$D_i^-$  = jarak alternatif ke- $i$  dari solusi ideal negatif

Nilai  $V_i$  yang lebih besar menunjukkan alternatif yang lebih dipilih

III. IMPLEMENTASI

Penelitian ini diawali dengan melakukan studi pustaka, dengan mempelajari buku, jurnal, artikel, dan referensi lainnya yang berkaitan. Kemudian mengumpulkan dan menganalisa data-data, serta mewawancarai narasumber untuk mengetahui kriteria penilaian karyawan terbaik.

A. Data Karyawan

Proses pembuatan rekomendasi akan dijelaskan menggunakan teknik *simple random sampling* [7].

Sample data yang diambil sebanyak 30 karyawan, sebagian data diperlihatkan pada Tabel 1.

Tabel 1. Sample Data Karyawan

No	Karyawan	Nilai Kriteria				
		Kualitas & Kuantitas	Ketaatan	Kerjasama	Semangat Kerja	Disiplin
1	1	A	A	A	C	A
2	2	A	A	B	B	-
3	3	C	D	C	C	B
...	...	...	...	...	...	...
30	30	A	A	C	B	A

Kemudian dari Tabel 1, dikonversikan menjadi rating kecocokan dari setiap data karyawan pada setiap kriteria. Data karyawan yang dikonversi menjadi rating kecocokan adalah kualitas dan kuantitas kerja, ketaatan, kerja sama, semangat kerja, dan disiplin kerja. Penentuan nilai rating kecocokan kriteria dapat dilihat pada Tabel 2.

Tabel 2. Nilai Rating Kecocokan Kriteria

Kriteria	Ketentuan	Nilai Rating
Kualitas dan kuantitas kerja	A (sangat baik)	9
	B (baik)	7
	C (Cukup)	5
	D (kurang)	3
	E (sangat kurang)	1
Ketaatan	A (sangat baik)	9
	B (baik)	7
	C (Cukup)	5
	D (kurang)	3
	E (sangat kurang)	1
Kerja sama	A (sangat baik)	9
	B (baik)	7
	C (Cukup)	5
	D (kurang)	3
	E (sangat kurang)	1
Semangat kerja	A (sangat baik)	9
	B (baik)	7
	C (Cukup)	5
	D (kurang)	3
	E (sangat kurang)	1
Disiplin kerja	A (sangat baik)	9
	B (baik)	7
	C (Cukup)	5
	D (kurang)	3
	E (sangat kurang)	1

Berdasarkan pada Tabel 1, maka didapat hasil konversi yang bisa dilihat pada Tabel 2.

Tabel 3. Tabel Konversi Nilai Kecocokan

No	Karyawan	Nilai Kriteria				
		Kualitas & Kuantitas	Ketaatan	Kerjasama	Semangat Kerja	Disiplin
1	1	9	9	9	5	9
2	2	9	9	7	7	-
3	3	5	3	5	5	7
...	...	...	...	...	...	...
30	30	9	9	5	7	9

B. Metode Entropy

Dari Tabel 3, kemudian dinormalisasikan sesuai dengan Rumus 1, sehingga mendapatkan nilai kriteria ternormalisasi. Sebagai contoh untuk menghitung nilai kriteria ternormalisasi untuk yang memiliki nilai

kriteria 5, dan kriteria keseluruhan memiliki nilai maksimal 9, maka  $5 \div 9 = 0,56$ . Nilai Data Kriteria Ternormalisasi untuk sampel karyawan dapat dilihat pada tabel 4 berikut:

Tabel 4. Tabel Data Kriteria Ternormalisasi

No	Karyawan	Nilai Kriteria				
		Kualitas & Kuantitas	Ketaatan	Kerjasama	Semangat Kerja	Disiplin
1	1	1	1	1	0,56	1
2	2	1	1	0,78	0,78	-
3	3	0,56	0,33	0,56	0,56	0,78
...	...	...	...	...	...	...
30	30	1	1	0,56	0,78	1

Kemudian nilai data yang telah ternormalisasi dihitung jumlahnya sesuai dengan Rumus 1, dalam jumlah sample data ada sebanyak 30 karyawan. Setelah itu dilakukan perhitungan entropy setiap kriteria ke-i, untuk nilai entropy setiap kriteria maka, diperlukan perhitungan untuk rumus 2 untuk mencari nilai  $e_{max}$ . Nilai  $e_{max} = \ln m$ , dimana m adalah banyak data. Maka nilai  $e_{max} = 3,401$ . Setelah mendapatkan nilai  $e_{max}$ , dilanjutkan dengan rumus 3, yaitu mencari nilai K, dimana nilai  $K = 1 \div 3,401 = 0,294$ . Setelah itu bisa didapat nilai entropy setiap kriteria sesuai dengan rumus 4. Sebagai contoh untuk menghitung nilai entropy kriteria ketaatan, yaitu.

$$e(ketaatan) = -0,3 * (((1 \div 29,3) * \ln (1 \div 29,3)) + ((1 \div 29,3) * \ln (1 \div 29,3)) + \dots + ((dik \div Di) * \ln (dik \div Di))) = 0,997.$$

Nilai entropy setiap kriteria selengkapnya dapat dilihat pada Tabel 5.

Tabel 5. Nilai Entropy Setiap Kriteria

Kriteria	Kualitas & Kuantitas	Ketaatan	Kerjasama	Semangat Kerja	Disiplin
Entropy ke-i	0,998	0,997	0,886	0,991	0,993

Setelah mendapat nilai entropy setiap kriteria sesuai dengan Tabel 5, selanjutnya menentukan total entropy setiap kriteria sesuai dengan rumus 5, yaitu. Total Entropy =  $0,998 + 0,997 + 0,886 + 0,991 + 0,992 = 4,866$ .

kemudian melakukan pencarian bobot entropy sesuai dengan Rumus 6. Sebagai contoh untuk menghitung bobot entropy kriteria semangat kerja.

$$\lambda = \frac{1}{5-4,866} * (1 - 0,997) = 0,0702$$

Bobot Entropy selengkapnya dapat dilihat pada Tabel 6.

Tabel 6. Bobot Entropy

Kriteria	Kualitas & Kuantitas	Ketaatan	Kerjasama	Semangat Kerja	Disiplin
Entropy ke-i	0,009008	0,02320	0,8400	0,0702	0,0574

Jika sebelumnya telah ada nilai bobot yg telah ditentukan sebelumnya maka nilai *entropy* yang sebenarnya akan di dapat sesuai dengan Rumus 7. Nilai bobot awal untuk pengujian dapat dilihat pada Tabel 7.

Tabel 7. Bobot Awal

Kriteria	Kualitas & Kuantitas	Ketaatan	Kerjasama	Semangat Kerja	Disiplin
<b>Bobot Awal</b>	9	9	9	9	9

Berikut adalah contoh untuk menghitung nilai bobot *entropy* akhir dari kriteria ketaatan.

$$\lambda = (9 * 0.0220190) \div (0,08 + 0,2 + 7,7 + 0.6 + 0.5) = 0.0220190$$

Bobot *Entropy* akhir selengkapnya dapat dilihat pada Tabel 8.

Tabel 8 Bobot Entropy Akhir

Kriteria	Kualitas & Kuantitas	Ketaatan	Kerjasama	Semangat Kerja	Disiplin
<b>Bobot Awal</b>	0,009008	0,023209	0,840076	0,070239	0,05747

Dari hasil pengujian dapat disimpulkan bahwa hasil bobot *entropy* dan bobot *entropy* akhir menyatakan kriteria kerja sama yang menjadi kriteria utama dari semua kriteria. Perbedaan ini terjadi karena pada metode *entropy*, bobot dihitung berdasarkan karakteristik data pada kriteria. Semakin tinggi variasi antar data pada kriteria, maka bobot kriteria semakin tinggi atau penting.

### C. Metode TOPSIS

Dari tabel konversi nilai kecocokan pada Tabel 3, kemudian dinormalisasikan menggunakan mentransformasikan setiap elemen sesuai dengan rumus 8 sehingga di dapat bentuk Matriks Keputusan Ternormalisasi. Sebagai contoh untuk keputusan ternormalisasi karyawan 1.

$$R_{ij} = 5 \div 9 \sqrt{5^2 + 9^2 + \dots + x_{ij}^2} = 0,13411$$

Matriks keputusan ternormalisasi selengkapnya dapat dilihat pada Tabel 9.

Tabel 9 Matriks Keputusan Ternormalisasi

No	Karyawan	Nilai Kriteria				
		Kualitas & Kuantitas	Ketaatan	Kerjasama	Semangat Kerja	Disiplin
1	1	0,184	0,185	0,352	0,141	0,191
2	2	0,185	0,185	0,274	0,198	0,0212
3	3	0,102	0,062	0,195	0,141	0,148
...	...	...	...	...	...	...
30	30	0,185	0,185	0,195	0,198	0,190

Kemudian Matriks Keputusan Ternormalisasi akan dikalikan dengan nilai bobot *entropy* akhir berdasarkan bobot awal yang telah ditentukan, sesuai dengan Rumus 8. Nilai bobot *entropy* akhir dapat dilihat pada Tabel 8. Sebagai contoh untuk karyawan no 10 dalam nilai kriteria kerja sama.

$$V_{303} = 0,848273173 * 0.213200716 = 0,1808524478$$

Matriks keputusan ternormalisasi terbobot selengkapnya dapat dilihat pada Tabel 10.

Tabel 10 Matriks Keputusan Ternormalisasi Terbobot

No	Karyawan	Nilai Kriteria				
		Kualitas & Kuantitas	Ketaatan	Kerjasama	Semangat Kerja	Disiplin
1	1	0,00166	0,0043	0,2956	0,0099	0,0109
2	2	0,00166	0,0043	0,2299	0,0139	0,0012
3	3	0,0009	0,0014	0,1642	0,0099	0,0085
...	...	...	...	...	...	...
30	30	0,00166	0,0043	0,1642	0,0139	0,0109

Setelah mendapat nilai Matriks Keputusan Ternormalisasi Terbobot, diteruskan dengan mencari Solusi Ideal Positif dan Solusi Ideal Negatif pada masing-masing kriteria dihitung dengan Rumus 10 dan Rumus 11 dan hasilnya dapat dilihat pada Tabel 11.

Tabel 11 Solusi Ideal Positif dan Solusi Ideal Negatif

Nama Kriteria	Solusi Ideal Positif (A <sup>+</sup> )	Solusi Ideal Negatif (A <sup>-</sup> )
Kualitas & Kuantitas Kerja	0,00166	0,00092
Ketaatan	0,00430	0,00143
Kerjasama	0,2956	0,0328
Semangat Kerja	0,017909	0,00597
Disiplin	0,01097	0,00121

Setelah menetapkan Solusi Ideal Positif dan Solusi Ideal Negatif, dapat dihitung Jarak Alternatif Positif dan Jarak Alternatif Negatifnya dengan Rumus 12 dan Rumus 13 serta dapat dilihat hasilnya pada Tabel 12. Sebagai contoh untuk jarak alternatif positif dengan untuk karyawan no 30.

$$\begin{aligned} D_{30}^+ &= \sqrt{0^2 + 0^2 + (0,017266)^2 + (0,00016)^2 + 0^2} \\ &= \sqrt{0,017281} \\ &= 0,131459 \end{aligned}$$

Sebagai contoh untuk jarak alternatif negatif untuk karyawan no 30.

$$\begin{aligned} D_{30}^- &= \sqrt{0,00000^2 + 0,000008^2} \\ &= \sqrt{(0,017266)^2 + (0,000063)^2 + 0,000095^2} \\ &= \sqrt{0,017433} \\ &= 0,132033 \end{aligned}$$

Tabel 12 Jarak Alternatif Positif dan Negatif

No.	Karyawan	Jarak Alternatif	
		Positif $D_i^+$	Negatif $D_i^-$
1	1	0.007959	0.263024
2	2	0.066538	0.197280
3	3	0.131695	0.131662
...	...	...	...
30	4	0.131459	0.132033

Kemudian menghitung Nilai Preferensi dari masing-masing karyawan tersebut dengan Rumus 13 serta memberikan ranking dari urutan nilai tertinggi dari Nilai Preferensi yang dapat dilihat pada Tabel 13. Sebagai contoh untuk nilai preferensi karyawan dengan nomor 30.

$$V_{30} = \frac{0,132033}{0,132033+0,131459} = 0,501090$$

Tabel 13 Nilai Preferensi dan Urutan Ranking Sistem

No.	Karyawan	Ranking	
		Nilai Preferensi ( $D_i^+$ )	Rangking
1	1	0.970628	2
2	2	0.747788	6
3	3	0.499937	13
...	...	...	...
30	4	0.501090	7

#### D. Evaluasi System

Evaluasi sistem dilakukan dengan membagikan kuesioner ke pengguna sistem, yaitu direksi JSC sebanyak 5 orang. Pertanyaan kuesioner disusun berdasarkan teori Doll dan Torkzadeh (1988) tentang pengukuran kepuasan yang disebut dengan *End-User Computing Satisfaction* (EUCS). Terdapat lima komponen kepuasan yaitu [8] yaitu isi, akurasi, bentuk, kemudahan dan ketepatan waktu. Penilaian dihitung menggunakan skala Likert terhadap pertanyaan dengan angka 1 sampai 5, di mana 5 menunjukkan nilai sangat baik dan merupakan nilai tertinggi untuk pertanyaan *output* yang dapat memenuhi kebutuhan pengguna positif, sedangkan untuk pertanyaan negatif

menunjukkan nilai sangat buruk, untuk nilai 1 menunjukkan nilai sangat kurang untuk pertanyaan positif dan merupakan nilai sangat baik untuk pertanyaan negatif [8].

Didapat indeks tingkat kepuasan responden terhadap sistem pendukung keputusan berkisar antara 70% -80%, artinya penilaian terhadap sistem yang dibuat memberikan hasil pada tingkat yang cukup baik.

#### IV. SIMPULAN

Implementasi metode entropi dan TOPSIS telah berhasil diimplementasikan, karena sudah digunakan. Metode ini memiliki tingkat fleksibilitas yang tinggi dalam pengambilan keputusan. Sistem yang dirancang dan dibangun ini dapat digunakan oleh Project Manager Jakarta Smart City untuk menilai karyawan terbaik. Presentasi hasil nilai preferensi dan peringkat disajikan dalam bentuk tabel sehingga memudahkan kepala unit dalam mengambil keputusan. Pegawai peringkat tertinggi akan direkomendasikan oleh manajer untuk mendapatkan penghargaan dari perusahaan. Indeks tingkat kepuasan responden terhadap sistem pendukung keputusan berkisar antara 70% -80%, artinya penilaian terhadap sistem yang dibuat memberikan hasil pada tingkat yang cukup baik. Untuk meningkatkan tingkat akurasi perlu ditingkatkan isi dan bentuk karena penilaian ini merupakan penilaian yang masih rendah.

#### DAFTAR PUSTAKA

- [1] Herman, I., dkk. (2016). "Sistem Pendukung Keputusan Karyawan Terbaik Menggunakan Metode AHP dan TOPSIS".(pdf).Yogyakarta: Universitas Jenderal Achmad Yani.
- [2] R. K. Gavade, "Multi-Criteria Decision Making: An Overview of Different Selection Problems and Methods"
- [3] Handoyo Eko, Cahyani Andharini Dwi, Yunitarini Rika. (2014). *Sistem Pendukung Keputusan Pemilihan Produk Unggulan Daerah Menggunakan Metode Entropy dan Electre II (Studi Kasus: Dinas Koperasi, Industri dan Perdagangan Kabupaten Lamongan)*.(pdf). Madura: Universitas Trunojoyo.
- [4] M.C.Ramirez,"Ensemble Determination Using TOPSIS Decision Support System in Multi-Objective Evolutionary Neural Network Classifiers", 2010.
- [5] Nurlita Sari Rima , Rukun Santoso , Yasin Hasbi Yasin (2016). "KOMPUTASI METODE SAW DAN TOPSIS MENGGUNAKAN GUI MATLAB UNTUK PEMILIHAN JENIS OBJEK WISATA TERBAIK". JURNAL GAUSSIAN, Volume 5, Nomor 2, Tahun 2016, Halaman 289-298
- [6] Triyanti Vivi, Gadis, M.T. (2008). "Pemilihan Supplier Untuk Industri Makanan Menggunakan Metode Promethee".(pdf). Jakarta: Unika Atma Jaya.
- [7] Dewi, Nurul Kusuma, dkk. (2010). *Populasi dan Sampel Penelitian (Penelitian Pendidikan)*. (pdf). Yogyakarta: Universitas Negeri Yogyakarta.
- [8] Nazir M. (2005). *Metode Penelitian*. Bogor: Ghalia Indonesia



# Pembangunan *Web Service* Data Masyarakat Menggunakan REST API dengan *Access Token*

Muhammad Iqbal Perkasa<sup>1</sup>, Eko Budi Setiawan<sup>2</sup>

Program Studi Teknik Informatika, Universitas Komputer Indonesia, Jalan Dipatiukur 112 Bandung  
 mail@iqbalperkasa.com  
 eko@email.unikom.ac.id

Diterima 23 April 2018

Disetujui 8 Juni 2018

**Abstract**— Data is one of the most important things in this information and information technology era that evolving now. Currently, the government still has not used the public data maximally for administrative purposes. Utilization of this big population data is the creation of a web service application system with REST API where this data will be open and accessible to those who have access. One of the institutions that use this service is the Manpower and Transmigration Service where this system can make the Dinas staff more efficient to create and register job search cards using available community data. This application is able to provide and facilitate many parties, such as data administrators to monitor data usage, registration employee in input data, and people able to register independently.

**Index Terms**—Web service, API, Rest api, People data

## I. PENDAHULUAN

Pada tahun 2010, di salah satu Kabupaten di Jawa Barat tercatat memiliki 1.443.156 jiwa yang berusia di atas 15 tahun. Data tersebut mencakup nama lengkap, jenis kelamin, agama, golongan darah, tanggal dan tempat lahir, status perkawinan, alamat, dan kewarganegaraan, serta informasi detail lainnya. Namun banyaknya data yang tercatat di *server* hingga saat ini belum dimanfaatkan secara maksimal dan belum adanya halaman statistik untuk memonitor penggunaan data kependudukan tersebut.

Pemanfaatan data penduduk yang terolah salah satunya adalah digunakan oleh Dinas Ketenagakerjaan dan Transmigrasi dalam pengisian baris data pencari pekerjaan. Salah satu tugas dari dinas ini adalah pembuatan atau penerbitan Kartu Pencari Kerja untuk masyarakat. Setiap harinya petugas dinas mampu memberikan pelayanan dengan menerbitkan kartu tersebut sebanyak lebih dari 100 kartu.

Setelah dianalisis, Dinas Ketenagakerjaan dan Transmigrasi dapat menggunakan data yang tercatat di *server database* Dinas Kependudukan dan Catatan Sipil yang diharapkan mampu jauh mengurangi beban kerja dari petugas Dinas Ketenagakerjaan dan Transmigrasi. Data yang disediakan bersifat terbuka

dan transparan, meskipun begitu, pihak *developer* harus memiliki *access token* untuk mengakses data di *server*. Berdasarkan pemaparan di atas, maka sebuah *web service* untuk mengambil data masyarakat diharapkan mampu mempercepat dalam penulisan atau pengisian data masyarakat tersebut.

Pemanfaatan *web service* sudah banyak dilakukan pada beberapa sektor. Warsito [1] mengimplementasikan *web service* pada sektor pendidikan di lingkungan perguruan tinggi, Fatoni [2] menggunakannya dalam segi sektor bisnis pembelian tiket pesawat, serta penelitian Kurniawan [3] yang mengimplementasikan *rest web service* untuk proses *sales order* dan *sales tracking*.

Beberapa penelitian yang pernah dilakukan sebelumnya masih belum ada yang menggunakan fitur untuk pengelolaan *access token*. Penelitian ini menggunakan fitur untuk *monitoring* akses ke dalam *database* di *server*, sehingga seorang Administrator dapat dengan mudah memonitor kondisi *traffic* data terutama *request* akses data ke *server*.

## II. TINJAUAN PUSTAKA

### A. Web Service

Menurut W3C yang merupakan lembaga internasional yang bergerak dalam membangun standarisasi *web service* adalah sistem perangkat lunak yang didesain dapat dioperasikan mesin ke mesin melalui jaringan. Arsitektur *web service* menurut Chen, et al [4] memodelkan interaksi antara tiga peran yaitu penyedia layanan, konsumen layanan, dan pendaftar layanan. Penggunaan *web service* juga digunakan oleh Pruter et al [5] untuk mengkoneksikan secara dinamis dari perangkat-perangkat yang sudah diketahui maupun tidak diketahui di dalam satu jaringan komputer. Dengan demikian, *web service* memungkinkan sarana operasi antar perangkat lunak menjadi standar pada berbagai *platform* yang berbeda.

### B. Access Token

*Access token* merupakan sebuah *string unique* dan acak yang berisi angka dan huruf. Seperti yang dibahas pada penelitian Ahmadi [6] bahwa *access token* adalah sebuah kunci yang dibutuhkan oleh pemohon layanan yang harus digunakan untuk mengidentifikasi dan memverifikasi apakah pemohon memiliki hak dalam mengakses *web service*. Masing-masing pemilik akun pada layanan *web service* memiliki sebuah *access token* yang bersifat unik. *Access token* ini berfungsi untuk menambah tingkat keamanan pada *web service*, sehingga *web service* hanya dapat diakses oleh pihak yang berwenang saja.

### C. REST

*REST* menurut Sinha et al [7] dan Zhou et al [8] merupakan gaya arsitektur dalam mendesain sebuah *web service* di mana desain *REST* memiliki *resource* yang dapat diakses melalui sebuah alamat *HTTP URL* yang *unique*. *REST* juga memungkinkan klien dapat melakukan request melalui protokol *HTTP* dengan mudah menggunakan *URI* seperti pada penelitian Kurniawan [9]. Masing-masing alamat *URL* mengacu kepada kumpulan program yang akan dieksekusi dan akan mengembalikan pesan kepada pengirim perintah.

*REST* mengirimkan perintah yang akan dikerjakan oleh *server* menggunakan metode-metode *HTTP request method* yang disebut *verb*. Mengacu pada penelitian Lee [10] dan Rahman [11] terdapat delapan *HTTP request method*, yaitu *GET*, *POST*, *PUT*, *DELETE*, *OPTIONS*, *HEAD*, *TRACE*, dan *CONNECT*. Dalam penggunaan *API REST* hanya menggunakan empat dari metode-metode tersebut, yaitu: *GET*, *POST*, *PUT*, dan *DELETE*.

Pesan yang diterima dari *server* berupa kode *HTTP* berhasil atau gagal di dalam *header* dan isi pesan hasil pengolahan program itu sendiri. Berikut adalah kode-kode *HTTP* yang sering digunakan dalam menggunakan *REST API* [12] :

- a. 200 *OK*  
Perintah yang dikirim ke *server* benar dan berhasil dijalankan.
- b. 400 *Bad Request*  
Perintah yang dikirim ke *server* berisi isian yang salah.
- c. 401 *Unauthorized*  
Pengirim perintah mengirimkan kode kunci yang salah.
- d. 403 *Forbidden*  
Pengirim perintah tidak memiliki hak akses ke dalam *resource* yang dituju.
- e. 404 *Not Found*  
*Resource* yang dituju tidak ditemukan dalam *server*.
- f. 429 *Too Many Requests*

Pengirim perintah mengakses mencapai/melebihi dari limit yang telah ditentukan dari batas waktu tertentu.

### g. 500 *Internal Server Error*

*Server* atau potongan program dalam *resource* mengalami kesalahan.

### D. Apache JMeter

Aplikasi yang dibuat oleh *Apache Software Foundation* ini adalah berfungsi untuk melakukan pengujian terhadap *server*, bagaimana *server* mampu mengimbangi banyaknya permintaan yang datang. Aplikasi ini dapat digunakan untuk pengujian basis data, *FTP*, *LDP*, *Web Service*, *HTTP*, dan koneksi *TCP*.

### E. JSON

*JSON* adalah sebuah format pesan balikan yang berukuran kecil yang mudah dibaca dan ditulis oleh manusia, dan juga mudah ditulis dan diurai oleh mesin. *JSON* merupakan salah satu pesan balikan yang dapat digunakan dalam sebuah *REST API*.

*JSON* terbagi dalam dua struktur, yaitu:

1. Gabungan *name/value*, atau yang biasa dikenal *object* atau *record* dalam bahasa pemrograman.
2. List *value*, atau yang biasa dikenal larik dalam bahasa pemrograman.

### F. PHP

*PHP* adalah sebuah bahasa pemrograman sumber terbuka yang ditujukan untuk pemrograman *web* dan dapat diaplikasikan ke *HTML*. *PHP* merupakan bahasa skrip yang ditanam dalam *HTML*. Ini berarti bahwa kita dapat menggabungkan kode *PHP* dan *HTML* dalam berkas yang sama [13]. *PHP* merupakan bahasa pemrograman *server-side* yang di mana program akan dijalankan di *server* dan hasilnya akan diintegrasikan ke dalam kode sumber *HTML*.

### G. MySQL

*MySQL* adalah sebuah aplikasi *server* basis data yang dijalankan di *server*, *MySQL* menggunakan sintak *SQL* standar dalam penggunaannya. *MySQL* bersifat relasional yang memungkinkan satu tabel dapat berelasi dengan tabel lainnya, sehingga memungkinkan dilakukannya normalisasi untuk mencegah redundansi dalam pemakaian data.

## III. HASIL DAN PEMBAHASAN

Pada tahap ini, akan dijelaskan mengenai tahapan-tahapan dari penelitian yang dilakukan, yaitu adalah analisis masalah yang menjelaskan mengenai masalah yang ditemukan; analisis pembentukan *access token*, bagaimana proses *access token* dibentuk; analisis alur validasi, menjelaskan validasi permintaan masing-masing request sebelum data diberikan; analisis request dan response dari permintaan *web service*;

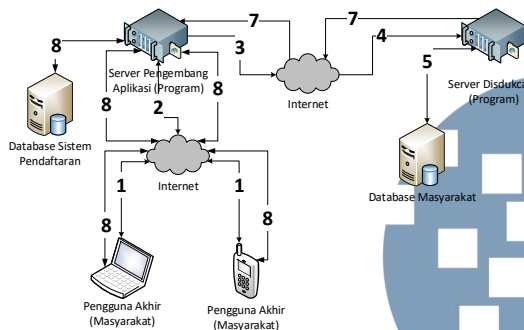
analisis infrastruktur; *use case diagram*; implementasi sistem; pengujian; dan terakhir adalah wawancara ke administrator data penduduk.

#### A. Analisis Masalah

Analisis masalah yang ditemukan setelah proses identifikasi di Dinas Kependudukan dan Catatan Sipil adalah data masyarakat yang dimiliki oleh Dinas Kependudukan dan Catatan Sipil belum dimanfaatkan dengan baik dan administrator data di Dinas Kependudukan dan Catatan Sipil belum memiliki aplikasi untuk memonitor penggunaan data masyarakat.

#### B. Gambaran Umum

Berikut adalah gambaran umum sistem API masyarakat yang dapat dilihat pada Gambar 1:



Gambar 1. Gambaran umum sistem API

Penjelasan dari Gambar 1 adalah sebagai berikut, yang diterangkan dengan bentuk poin per *poin* bagaimana tahapan dari pengambilan data dari *server*:

1. Pengguna akhir yaitu masyarakat mengakses web untuk mendaftarkan kartu pencari kerja pada aplikasi yang telah dikembangkan oleh pengembang aplikasi.
2. Permintaan akan masuk ke dalam *server* pengembang aplikasi. Program akan membaca permintaan dari pengguna akhir, yaitu meminta data masyarakat berdasarkan NIK.
3. *Server* pengembang aplikasi harus terkoneksi ke internet untuk dapat mengakses *server* Dinas Kependudukan dan Catatan Sipil pada alamat domain tertentu.
4. *Server* pengembang aplikasi mengakses *server* Dinas Kependudukan dan Catatan Sipil pada alamat endpoint URL API yang telah dibuat.
5. Setelah berhasil melakukan validasi permintaan yang *valid* dari pengembang aplikasi, program akan terhubung ke basis data masyarakat pada *server database* dan mengambil data masyarakat berdasarkan NIK yang dimaksud.
6. Setelah berhasil mengambil data yang dimaksud, program akan menuliskan catatan *history* di *server database* aplikasi API dan

mengurangi limit pemakaian *access token* pengembang yang bersangkutan.

7. *Server* Dinas Kependudukan dan Catatan Sipil mengembalikan balikan berupa data masyarakat berformat *JSON* ke *server* pengembang aplikasi.
8. Data yang telah didapat akan dikembalikan ke aplikasi yang digunakan oleh pengguna akhir, yaitu masyarakat.

#### C. Analisis Pembentukan Access Token

Pembentukan *access token* merupakan asumsi dari peneliti. Token ini didapat dari kombinasi alamat *email* penanggung jawab pemilik *access token* dan waktu *timestamp* pembuatan *access token* yang dienkripsi menggunakan metode SHA256. Contoh dari pembentukan *access token* yang dapat dilihat pada Tabel 1.

Tabel 1. Contoh pembentukan *access token*

Alamat Email	email@domain.com
Timestamp Pembuatan Token	1492035660 (12 April 2017 22:20)
Fungsi SHA256	sha256('email@domain.com1492035660')
Access Token yang Didapat	91b37dee294eb66ce67c9d8bbbe6f71f0d9baf8f9a697c37f35e4108bbccf9be

Berdasarkan data pada tabel 1 maka diperoleh dari pembangkitan *access token* tersebut adalah berupa string `91b37dee294eb66ce67c9d8bbbe6f71f0d9baf8f9a697c37f35e4108bbccf9be`. String inilah yang kemudian harus disisipkan di setiap permintaan data API oleh akun `email@domain.com`.

#### D. Analisis Alur Validasi

Untuk memastikan permintaan berasal dari sumber yang jelas dan untuk memastikan keamanan data, maka perlu dibuatnya tahapan-tahapan validasi dari sumber permintaan. Masing-masing tahap harus dipenuhi untuk mendapatkan data yang dibutuhkan. Terdapat delapan tahap validasi yang harus dipenuhi, yaitu:

1. Validasi *access token* kosong. Sistem akan memeriksa apakah sumber permintaan membawa *access token* atau tidak. Apabila parameter *access token* kosong, maka sistem akan mengembalikan kode *error* 403 *Forbidden* dan pesan *error* "Harap masukkan *access token*".
2. Validasi format *access token*. Sistem akan memeriksa apakah sumber permintaan membawa *access token* dengan format yang valid atau tidak. *Access token* yang valid adalah berupa string dengan panjang string adalah enam puluh empat (64) karakter. Apabila format yang dikirimkan salah, maka sistem akan mengembalikan kode *error* 400

- Bad Request* dan pesan error “Format *access token* salah”.
- Validasi kebenaran *access token*. Sistem akan memeriksa apakah sumber permintaan membawa *access token* yang valid atau tidak. Sistem akan membandingkan *access token* yang dibawa oleh sumber dengan daftar *access token* di basis data. Apabila *access token* tidak ditemukan, maka sistem akan mengembalikan kode error 401 *Unauthorized* dan pesan error “*Access token* salah”.
  - Validasi *access token* dengan domain terdaftar. Sistem akan memeriksa apakah *domain* sumber permintaan berasal dari *domain* yang terdaftar pada *access token* yang bersangkutan. Sistem akan membandingkan *domain* dari sumber permintaan dengan daftar *domain access token* yang bersangkutan di basis data. Apabila *domain* tidak ditemukan, maka sistem akan mengembalikan kode error 403 *Forbidden* dan pesan error “Sumber *domain* tidak terdaftar pada *access token* ini”. Validasi ini penting untuk memblokir permintaan AJAX dari *domain* yang tidak terdaftar.
  - Validasi batas *limit*. Sistem akan memeriksa apakah *access token* yang bersangkutan memiliki limit akses data atau tidak. Sistem akan memeriksa jumlah permintaan dari *access token* yang bersangkutan dari batas waktu tertentu di basis data. Apabila limit *access token* telah dicapai, maka sistem akan mengembalikan kode error 429 *Too Many* dan pesan error “*Access token* salah”.
  - Validasi NIK kosong. Sistem akan memeriksa apakah sumber permintaan membawa NIK atau tidak. Apabila parameter NIK kosong, maka sistem akan mengembalikan kode error 404 *Not Found* dan pesan error “Harap masukkan NIK”.
  - Validasi format NIK. Sistem akan memeriksa apakah sumber permintaan membawa NIK dengan format yang valid atau tidak. NIK yang valid adalah berupa kumpulan angka dengan panjang numerik adalah enam belas (16) karakter. Apabila format yang dikirimkan salah, maka sistem akan mengembalikan kode error 400 *Bad Request* dan pesan error “Format NIK salah”.
  - Validasi kebenaran NIK. Sistem akan memeriksa apakah sumber permintaan membawa NIK yang valid atau tidak. Sistem akan membandingkan NIK yang dibawa oleh sumber dengan daftar NIK di basis data Dinas Kependudukan dan Catatan Sipil. Apabila NIK tidak ditemukan, maka sistem akan mengembalikan kode error 404 *Unauthorized* dan pesan error “NIK salah”.

### E. Analisis Request-Response

Setelah permintaan yang masuk ke dalam *server* divalidasi dan diterima dengan benar, maka *server* akan memproses permintaan menghubungkan dengan basis data.

#### 1. URL Endpoint.

Semua permintaan diakses melalui sebuah endpoint URL yang beralamat di `http://11x.97.187.1xx:8080/api/get_data_by_nik`.

#### 2. Parameter

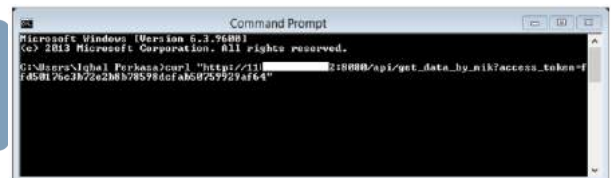
Terdapat dua parameter yang dibutuhkan dalam setiap proses. Kedua parameter harus valid yang dalam setiap permintaan *server* akan memvalidasi dan akan mengembalikan balikan *error* apabila terjadi kesalahan. Parameter-parameter tersebut adalah *access token*, kunci *access token* yang dimiliki oleh masing-masing akun yang valid (terverifikasi, berstatus aktif, dan memiliki sisa limit); dan *nik*, NIK masyarakat yang akan diambil datanya.

#### 3. HTTP Verb

HTTP *Verb* yang digunakan untuk sistem API masyarakat ini adalah *GET*.

#### 4. Otentikasi

Setiap permintaan, sistem akan memvalidasi apakah permintaan membawa *access token* yang valid atau tidak. *Access token* dikirim melalui parameter pada URL.



Gambar 2. Otentikasi *access token* dikirim melalui parameter url

#### 5. Limiting

Masing-masing *access token* memiliki batas limit permintaan data sebanyak 60 permintaan setiap menit. Tiap menit limit ini akan kembali di-*reset* dan mengulang hitungan. Administrator memiliki hak untuk mengubah jumlah batas limit.

#### 6. Response

Apabila sistem berhasil memvalidasi permintaan yang masuk, maka sistem akan mengembalikan permintaan yang dibutuhkan. Balikan yang diberikan berupa format *JSON* dan informasi-informasi *header*. Contoh balikan dari permintaan yang valid dapat dilihat pada gambar 3.



x-limit-limit → 999999  
 x-limit-remaining → 999998

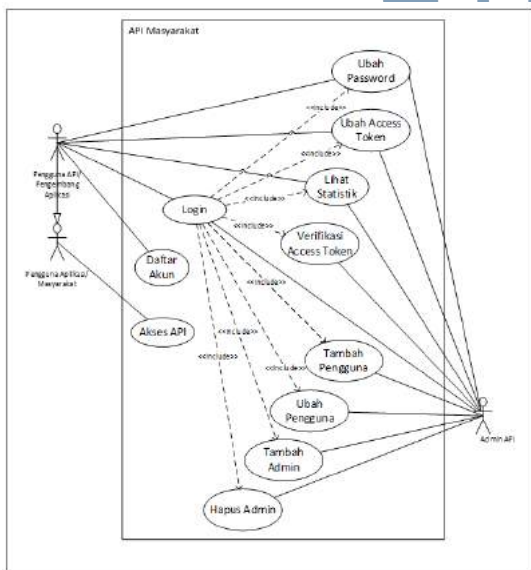
```

1 * {
2   "SUCCESS": true,
3   "MESSAGE": [],
4   "DATA": {
5     "NIK": "0018",
6     "NAMA_LOKP": "CIANJUR",
7     "TMPT_LHR": "JAWA BARAT",
8     "NAMA_PROP": "CIANJUR",
9     "NAMA_KAB": "SELAJAMBE",
10    "NAMA_KEC": "32",
11    "NO_PROP": "3",
12    "NO_KAB": "9",
13    "NO_KEC": "2001",
14    "AGAMA": "ISLAM",
15    "JENIS_KLMIN": "PEREMPUAN",
16    "GOL_DRH": "TIDAK TAHU",
17    "STAT_KMN": "BELUM KAMIN",
18  }
19 }
    
```

Gambar 3. Hasil balikan data masyarakat berupa JSON

F. Use Case Diagram

Use Case Diagram adalah diagram yang menggambarkan kegiatan dan cakupan yang dikerjakan oleh masing-masing aktor. Use case diagram pada aplikasi yang akan dibangun yang dapat dilihat pada gambar 4.



Gambar 4. Use case diagram

G. Implementasi Sistem

Tahap implementasi menjelaskan mengenai implementasi yang dilakukan dari pembahasan dan analisis pada tahapan-tahapan sebelumnya.

G.1. Implementasi Perangkat Lunak

Pada tahap ini akan dijelaskan mengenai implementasi perangkat lunak yang terdapat di lapangan yang dapat dilihat pada tabel 2.

Tabel 2. Implementasi perangkat lunak

No.	Komponen	Spesifikasi
1.	Arsitektur sistem operasi	32 bit
2.	Apache Server	versi 2.4.4
3.	PHP	versi 5.5.3
4.	MySQL/MariaDB	versi 5.6.11
5.	Browser	Tersedia

G.2. Implementasi Server

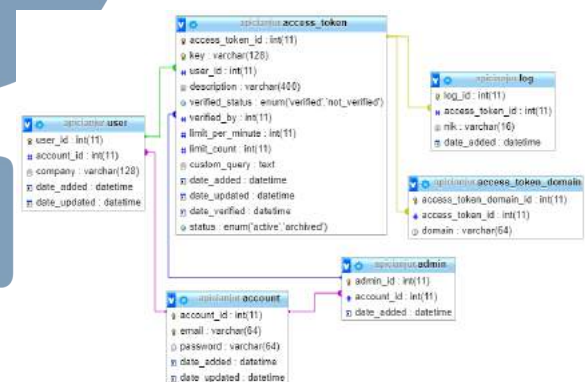
Pada tahap ini akan dijelaskan mengenai implementasi server yang terdapat di lapangan yang dapat dilihat pada Tabel 3.

Tabel 3. Implementasi server

No.	Komponen	Spesifikasi
1.	CPU	Intel Xeon X5650 @ 2.67 GHz
2.	Media penyimpanan (HDD/SSD)	1 TB
3.	Memori (RAM)	12 GB
4.	Keyboard	Tersedia
5.	Mouse	Tersedia
6.	Koneksi Internet	Tersedia

G.3. Implementasi Basis Data

Implementasi basis data merupakan implementasi tabel-tabel basis data yang digunakan untuk menyimpan seluruh data sistem yang dapat dilihat pada gambar 5.



Gambar 5. Skema implementasi basis data

H. Implementasi Antarmuka

Tahap ini akan memaparkan implementasi antarmuka dari halaman web untuk memonitor penggunaan data yang telah digunakan dari pengakses. Antarmuka meliputi tampilan untuk melihat penggunaan data dalam bentuk grafik dan tabel dan form pengelolaan data.

1. Login

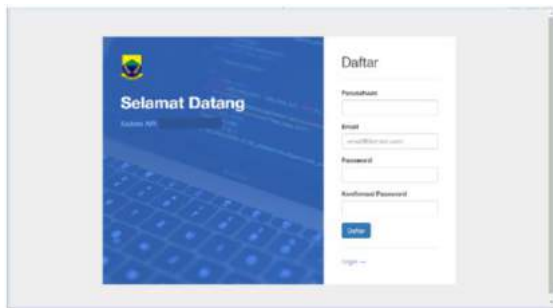
Antarmuka untuk menampilkan form login yang digunakan untuk masuk ke dalam sistem. Dibutuhkan dua masukan, yaitu email dan password dari pengguna yang bersangkutan.



Gambar 6. Antarmuka login

## 2. Daftar Akun

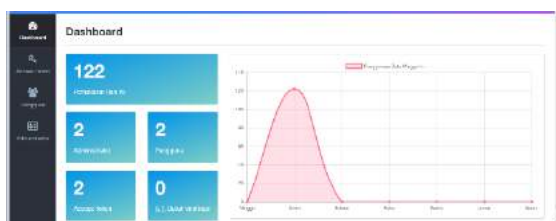
Antarmuka untuk menampilkan form pendaftaran akun yang digunakan untuk mendaftar ke sistem. Setiap pendaftar harus memasukkan detail data instansi atau perusahaan mereka. Data yang harus dimasukkan adalah nama perusahaan, email, dan *password*.



Gambar 7. Antarmuka daftar akun

## 3. Dashboard

Antarmuka untuk menampilkan statistik penggunaan data masyarakat dari keseluruhan pengguna di sistem. Grafik menampilkan penggunaan data pada minggu tersebut. Di sebelah kiri dashboard terdapat jumlah dari data yang dimiliki di sistem, yaitu data jumlah access token, data jumlah pengguna, dan data jumlah administrator. Tampilan antarmuka dashboard dapat dilihat pada gambar 8.



Gambar 8. Antarmuka dashboard

## 4. List Access Token

Antarmuka untuk menampilkan list *access token* yang terdaftar di sistem. Terdapat tiga tombol untuk

melakukan aksi dari masing-masing baris, yaitu lihat/ubah *access token*, melihat deskripsi dari penggunaan *access token*, dan lihat statistik. Tampilan list access token dapat dilihat pada gambar 9.

Gambar 9. Antarmuka list access token

## 5. Form Ubah Access Token

Antarmuka untuk menampilkan form ubah *access token* yang digunakan untuk mengubah data dari masing-masing *access token*. Terdapat juga tombol untuk meregenerasi *access token* yang baru. Tampilan form ubah *access token* dapat dilihat pada gambar 10.

Gambar 10. Antarmuka form ubah access token

## 6. Statistik Access Token

Antarmuka untuk menampilkan statistik penggunaan data dari *access token* yang bersangkutan. Statistik ditampilkan dalam bentuk grafik dan tabel. Terdapat juga log dari 50 penggunaan data terakhir serta jumlah penggunaan dari *access token* di menit tersebut. Tampilan statistik *access token* dapat dilihat pada gambar 11.



Gambar 11. Antarmuka statistik *access token*

### I. Pengujian Web Service

Pengujian pertama yang dilakukan adalah dengan menghitung jumlah kartu pencari kerja sebelum dan sesudah menggunakan *web service*. Didapat dari perhitungan menggunakan *query count SQL*, jumlah kartu pencari kerja sebelum adalah berkisar 300 kartu dan setelah menggunakan *web service*, didapat jumlah kartu pencari kerja adalah berkisar 900 kartu.

Pengujian selanjutnya adalah pengujian menggunakan aplikasi *Apache JMeter*. Dengan aplikasi ini kita mampu melihat bagaimana ketahanan *server* akan permintaan *request* yang dilakukan. Pengujian akan dilakukan dengan *sample* permintaan 100 kali dalam 1 detik. Hasil dari pengujian yang dilakukan yang akan disampaikan dalam bentuk tabel pada tabel 4 dan grafik yang disajikan pada gambar 12.

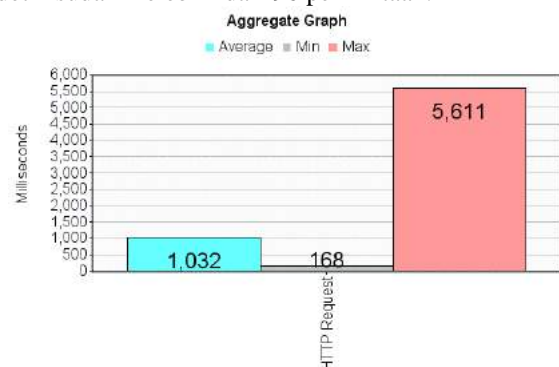
Tabel 4. Hasil pengujian 100 *sample* permintaan

Sample #	Sample Time (ms)	Sample #	Sample Time (ms)
1	232	51	1198
2	212	52	1169
3	265	53	1198
4	293	54	1226
5	241	55	1196
6	291	56	1449
7	203	57	1469
8	215	58	1245
9	204	59	1256
10	198	60	1243
11	241	61	1524
12	231	62	1503
13	210	63	1261
14	216	64	1248
15	246	65	1322
16	235	66	1623
17	237	67	1281
18	279	68	1524
19	168	69	1235
20	214	70	1215
21	236	71	1240
22	207	72	1225
23	248	73	1237
24	195	74	1234
25	266	75	1508
26	246	76	1300
27	186	77	1293
28	195	78	1284
29	194	79	1256
30	215	80	1554
31	217	81	1341
32	231	82	1603
33	227	83	1259
34	227	84	1642

Sample #	Sample Time (ms)	Sample #	Sample Time (ms)
35	212	85	1360
36	212	86	1330
37	177	87	1292
38	212	88	1356
39	223	89	1326
40	214	90	1361
41	173	91	1428
42	230	92	1409
43	219	93	1360
44	1201	94	1362
45	1233	95	1403
46	1219	96	2288
47	1267	97	5579
48	1259	98	5588
49	1298	99	5605
50	1196	100	5611

Kita bisa lihat bahwa pada *sample* percobaan ke 1 sampai 43, *server* mampu menangani permintaan dengan cepat, yaitu dalam rentang waktu 168 – 293 ms dengan rata-rata 223.093 ms, kemudian *sample* 44 – 96 menunjukkan kenaikan waktu namun tidak berarti yaitu dalam rentang waktu 1169 – 2288 ms dengan rata-rata 1345.453 ms. Kenaikan yang cukup berarti adalah ketika *server* menangani lebih dari 96 permintaan, waktu yang dibutuhkan adalah lebih dari 5500 ms.

Dari hasil 100 *sample* percobaan yang dilakukan dalam 1 detik, empat *sample* terakhir baru menunjukkan indikasi *server* mengalami beban, maka dapat diambil kesimpulan bahwa *server* mampu menangani permintaan data masyarakat ini adalah ~96 permintaan dalam 1 detik. Hasil ini dapat menjadi masukan bagi pihak Dinas untuk menambah *resource* pada *server* apabila rata-rata pengaksesan dalam satu detik sudah melebihi dari 96 permintaan.

Gambar 12. Hasil percobaan 100 *sample* permintaan

Gambar 12 menunjukkan grafik waktu permintaan dari 100 permintaan dalam 1 detik. Gambar tersebut menunjukkan rata-rata waktu adalah 1.032 ms, minimal waktu adalah 168 ms, dan maksimal waktu adalah 5.611 ms.

#### J. Wawancara Administrator

Wawancara dilakukan untuk mengetahui penilaian dari administrator basis data selaku administrator dari sistem API Masyarakat. Setelah dilakukan wawancara dengan narasumber pada hari Rabu tanggal 26 Juli 2017, didapatkan kesimpulan wawancara sebagai berikut:

1. Data-data yang ditampilkan dalam aplikasi ini sudah baik yang di mana ditampilkan berupa grafik dan tabel.
2. Kantor-kantor sebagai pengembang dan masyarakat dapat dimudahkan dengan adanya *web service* ini untuk memudahkan dalam pengambilan data masyarakat.
3. Tampilan dan menu dari aplikasi ini sudah baik dan mudah digunakan.
4. Aplikasi sudah bagus dan memudahkan untuk melihat data-datanya, pun dengan tampilan sudah baik dan tepat.

#### IV. SIMPULAN

Adapun kesimpulan dan saran yang didapat dari penelitian ini adalah sebagai berikut:

##### A. Kesimpulan

Dari hasil yang didapat dari tahap-tahap yang telah dikerjakan melalui proses perancangan, implementasi, pengujian, dan wawancara dengan administrator, didapat kesimpulan sebagai berikut:

1. *Web service* ini mampu dan dapat mempercepat pendaftaran dengan keandalan *server* yang baik.
2. Administrator data penduduk dapat dimudahkan dalam memonitor penggunaan data penduduk dan mengatur hak akses masing-masing *access token*.

##### B. Saran

Adapun beberapa saran yang dapat disampaikan adalah sebagai berikut:

1. Memberikan statistik data penggunaan data masyarakat secara *real-time*, untuk memberikan *user experience* yang lebih baik untuk administrator data.
2. Memberikan fitur *cache* pada *web service* untuk penggunaan data yang lebih efisien.

#### DAFTAR PUSTAKA

- [1] Warsito, A.B., Ananda, A. and Triyanjaya, D. Penerapan Data JSON Untuk Mendukung Pengembangan Aplikasi Pada Perguruan Tinggi Dengan Teknik Restfull Dan Web Service. *Technomedia Journal*, 2(1), pp.26-35, 2017.
- [2] Fatoni, A., Rancang Bangun Aplikasi PembelianTiket Pesawat Online Berbasis Mobile Pada platform IOS dengan Bahasa Pemrograman Swift memanfaatkan API Tiket. com. *Jurnal Manajemen Informatika*, 5(2), 2016.
- [3] Kurniawan, E. Implementasi Rest Web Service Untuk Sales Order dan Sales Tracking Berbasis Mobile. *Jurnal Eksplorasi Karya Sistem Informasi dan Sains*, 7(1), 2015.
- [4] Chen, M., Zhang, D. and Zhou, L, Providing web services to mobile users: the architecture design of an m-service portal. *International Journal of Mobile Communications*, 3(1), pp.1-18, 2005.
- [5] Prüter, S., Moritz, G., Zeeb, E., Salomon, R., Golatowski, F. and Timmermann, D., Applicability of web service technologies to reach real time capabilities. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on* (pp. 229-233). IEEE, 2008.
- [6] Ahmadi, R., Heidari, E. and Zand, M, Security Enhancementfor Restful Web Services. *Journal of Fundamental and Applied Sciences*, 8(2S), pp.2804-2817, 2016.
- [7] Sinha, R., Khatkar, M. and Gupta, S.C., Design & Development of a REST based Web Service Platform for Applications Integration on Cloud. *International Journal of Innovative Science, Engineering and Technology*, 1(7), pp.385-389, 2014.
- [8] Zhou, W., Li, L., Luo, M. and Chou, W., REST API design patterns for SDN northbound API. In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on* (pp. 358-365). IEEE, 2014.
- [9] Kurniawan, Y.K., Oslan, Y. and Kristanto, H., Implementasi Rest-API Untuk Portal Akademik Ukdw Berbasis Android. *Jurnal Eksplorasi Karya Sistem Informasi dan Sains*, 6(2), 2015.
- [10] Lee, H.M. and Mehta, M.R., Defense against REST-based web service attacks for enterprise systems. *Communications of the IIMA*, 13(1), p.5, 2013.
- [11] Rahman, M.A., Kuswardayan, I. and Hariadi, R.R., Perancangan dan Implementasi RESTful Web Service untuk Game Sosial Food Merchant Saga pada Perangkat Android. *Teknik Informatika ITS*, 1(2), 2013
- [12] T. Fredrich, "RESTful Service Best Practices," 2012.
- [13] R. Sianipar, *Membangun Web PHP dan MySQL*, Informatika : Bandung, 2015.



# Rancang Bangun Sistem Rekomendasi Restoran Menggunakan Metode AHP dan VIKOR pada Platform LINE

Andre Utomo Martliong<sup>1</sup>, Ni Made Satvika Iswari<sup>2</sup>

Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia

<sup>1</sup>[andreutomo@outlook.com](mailto:andreutomo@outlook.com), <sup>2</sup>[satvika@umn.ac.id](mailto:satvika@umn.ac.id)

Diterima 26 April 2018

Disetujui 8 Juni 2018

**Abstract** — Since 2012 until 2017, culinary business in Indonesia has increased by an average 7% to 14% per year. Nowadays, application technology such as Zomato, helps consumers to find information about restaurant's promos, advantages and disadvantages of a restaurant, and variations of the menu their provided. People who owned the culinary business can interact directly with consumers through social media such as LINE. LINE has 90 million users that communicate with each other by using features that LINE's provided, such as stickers, news portals, voice and video calls, and LINE Message API service. With 90 million LINE's users in Indonesia and growing culinary business, developing a restaurant recommendation system in LINE will provide solutions to many people, which inspired the author to develop this system. Data of restaurant it's taken using Zomato API. Developing this system needs a method of AHP (Analytic Hierarchy Process) and VIKOR (Visekriterijumsko KOMPromisno Rangiranje), where the application will be built using PHP language. The AHP method is used to assign a weighting value to criteria, and the VIKOR method is used to sort the best alternative option. Based on research through questionnaires, this system's design and implementation has reached 71,13 value of usability, which mean that system is useful to help people to find restaurant their looking for.

**Index Terms** - Culinary, Zomato API, LINE Message API, AHP, VIKOR

## I. PENDAHULUAN

Seiring dengan berkembangnya zaman fungsi dari kebutuhan pangan atau makanan saat ini tidak hanya sebagai kebutuhan pokok, melainkan sudah menjadi bagian dari gaya hidup masyarakat Indonesia [1]. Saat ini bisnis kuliner di Indonesia mengalami peningkatan dengan rata-rata 7% sampai 14% per tahun dalam 5 tahun terakhir [2]. Dengan berkembangnya teknologi internet, pelaku bisnis di bidang kuliner dapat memberikan informasi mengenai produk yang dijual melalui *website* dan berinteraksi langsung dengan

konsumen melalui sosial media.

*Website* yang dimaksud dalam memberikan informasi tentang promo, kelebihan dan kekurangan suatu restoran, serta variasi menu yang disediakan adalah Zomato, Pergikuliner, dan Qraved [3]. Sosial media seperti LINE dan Whatsapp dapat digunakan untuk berinteraksi langsung dengan konsumen. Saat ini LINE menjadi populer di Indonesia berkat fitur-fitur yang ditawarkannya, seperti stiker untuk mengekspresikan emosi pengguna, portal berita, *voice* dan *video* call, serta diberikannya layanan message API untuk *developer*. Berbeda dengan Whatsapp yang hanya memiliki fitur standar seperti berkirim pesan, *video* dan *voice* call. Berkat fitur yang ditawarkan oleh LINE membuat pengguna aktif LINE di Indonesia meningkat dengan jumlah lebih dari 90 juta [4].

Dengan meningkatnya pengguna LINE di Indonesia serta banyaknya *website* tentang informasi restoran maka dibuatlah sebuah aplikasi pada platform LINE yang dapat memberikan rekomendasi restoran. Dalam pembuatannya dibutuhkan sebuah metode MCDM (*Multi Criteria Decision Making*), yaitu *Analytic Hierarchy Process* (AHP) dan *Visekriterijumsko KOMPromisno Rangiranje* (VIKOR).

Metode *Analytic Hierarchy Process* (AHP) dapat digunakan untuk menentukan suatu masalah yang kompleks. Penggunaan metode AHP sudah pernah diteliti oleh Purnomo dengan judul "*Analisis Perbandingan Menggunakan Metode AHP, TOPSIS, dan AHP-TOPSIS dalam Studi Kasus Sistem Pendukung Keputusan Penerimaan Siswa Program Akselerasi*" mendapatkan kesimpulan bahwa metode AHP baik digunakan untuk melakukan pembobotan suatu kriteria, tetapi kurang dalam mengurutkan pilihan alternatif terbaik [5].

Metode VIKOR (*Visekriterijumsko KOMPromisno Rangiranje*) dalam bahasa Serbia yang memiliki arti *Multicriteria Compromise Ranking*) merupakan metode MCDM yang digunakan untuk melakukan perbandingan dari multikriteria. Penelitian yang dilakukan Lengkong

dengan judul “Implementasi Metode VIKOR untuk Seleksi Penerima Beasiswa” menyimpulkan dalam mengurutkan suatu pilihan alternatif terbaik menggunakan metode VIKOR memberikan hasil yang sesuai, namun dalam pemberian pembobotan kriteria masih kurang [6]. Maka dari itu metode AHP dan metode VIKOR sangat cocok untuk diimplementasikan ke dalam penelitian ini karena metode AHP dan metode VIKOR masing-masing memiliki kelebihan dan kekurangan yang saling melengkapi.

Penelitian yang menggunakan metode AHP dan metode VIKOR yang berjudul “Hybrid Multi Attribute Decision Making untuk Seleksi Penerima Bantuan Studi”, dapat disimpulkan dengan menggabungkan metode AHP dan metode VIKOR menghasilkan perbandingan yang lebih baik dibandingkan dengan satu metode saja, karena proses pembobotan kriteria menggunakan metode AHP dan metode VIKOR digunakan dalam perbandingan alternatif [7].

## II. TELAHAH LITERATUR

### A. LINE

LINE merupakan sebuah *platform* komunikasi berbasis internet yang menawarkan berbagai layanan, seperti *chatting*, *voice call*, *video call*, dan *smart portal* yang berkaitan dengan konten, hiburan, dan bisnis. LINE@ merupakan salah satu layanan dari LINE yang ditujukan bagi pelaku bisnis untuk meningkatkan serta memberikan informasi kepada konsumen. Fitur yang ditawarkan oleh LINE@ antara lain *broadcast message*, *1:1 chat*, *keyword reply*, dan *auto reply*, dimana fitur-fitur tersebut didukung dengan *Bot API*. Tahun 2016 LINE meluncurkan *messaging API* yang merupakan sebagai pengganti dari *bot API*, dimana *messaging API* memberikan kemudahan bagi pengembang untuk memberikan respon yang interaktif dalam *room chat* atau *group* dari *server* atau disebut *webhook* [8].



Gambar 1 Cara Kerja LINE Messaging API

Gambar 1 menjelaskan bahwa dengan *Messaging API*, aplikasi yang dibuat dapat berkirim informasi antara *server* dengan aplikasi LINE pengguna melalui *platform* LINE. Ketika pengguna LINE menambahkan akun bisnis sebagai teman atau pengguna mengirimkan pesan, *platform* LINE akan mengirimkan notifikasi ke *server* dengan mengirimkan informasi dalam bentuk

JSON melalui HTTP ke URL yang sudah didaftarkan sebagai *webhook*. Selanjutnya *server* akan mengirimkan respon ke pengguna berdasarkan *request* yang dikirim sebelumnya.

### B. Metode Analytic Hierarchy Process

Metode *Analytic Hierarchy Process* (AHP) dikembangkan oleh Prof. Thomas Lorie Satty. Metode AHP digunakan untuk menyelesaikan suatu masalah yang kompleks [7]. Dalam penelitian yang dilakukan, metode AHP hanya sampai pada tahap pembobotan kriteria dan selanjutnya menggunakan metode VIKOR dalam hal perbandingan. Langkah-langkah dalam metode AHP adalah sebagai berikut:

1. Menyusun kepentingan relatif antara dua buah kriteria ke dalam bentuk matriks perbandingan. Kriteria tersebut diukur berdasarkan skala numerik dengan nilai kepentingan 1 (sama penting) sampai 9 (mutlak lebih penting).
2. Bentuk matriks perbandingan kriteria dalam menggunakan Persamaan (2.1) dimana nilai  $C_1$ ,  $C_2$ , dan  $C_m$  merupakan kriteria yang disusun berpasangan menjadi matriks A.

$$A = \begin{matrix} C_1 & \begin{bmatrix} C_1 & C_2 & \dots & C_m \\ C_2 & 1 & C_{12} & \dots & C_{1m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ C_m & C_{m1} & C_{m2} & \dots & 1 \end{bmatrix} \\ \dots(2.1) \end{matrix}$$

3. Menormalisasikan hasil matriks yang didapat pada Persamaan (2.2) dan menghitung bobot setiap kriteria ( $w$ ) pada Persamaan (2.3).

$$\sum_i a_{ij} = 1 \quad \dots(2.2)$$

$$w_i = \frac{1}{n} \sum_j a_{ij}^1$$

... (2.3)

4. Menghitung perkalian matriks antara matriks (A) dengan  $w_i$ .
5. Menghitung nilai  $t$  menggunakan Persamaan (2.4).

$$t = \frac{1}{n} \sum_{i=1}^n \left( \frac{\text{elemen ke-}i \text{ pada } (A)(w^T)}{\text{elemen ke-}i \text{ pada } (w^T)} \right) \quad \dots(2.4)$$

6. Menghitung nilai konsistensi CI, menggunakan Persamaan (2.5).

$$CI = \frac{t-n}{n-1}$$

... (2.5)

Jika nilai CI=0 maka matriks A konsisten, jika CI ≤ 0,1 maka matriks A cukup konsisten, dan jika nilai CI > 0,1 maka matriks A sangat tidak konsisten.

C. Metode VIKOR

Metode VIKOR atau *Visekriterijumsko KOMPromisno Rangiranje* merupakan bahasa Serbia yang memiliki arti *Multicriteria Compromise Ranking*. Metode VIKOR merupakan bagian dari metode MCDM yang berfungsi dalam melakukan perankingan dari multikriteria untuk melihat solusi atau alternatif terdekat sebagai pendekatan kepada solusi ideal dalam perankingan [8]. Menurut Opricovic dan Tzeng seperti yang dilansir dalam jurnal yang berjudul “*Penentuan Kelayakan Lokasi Usaha Franchise Menggunakan Metode AHP dan VIKOR*”, tujuan utama dari metode ini adalah untuk menghasilkan solusi kompromi dengan melakukan perankingan terhadap hasil nilai alternatif dan kriteria yang bertolak belakang. Solusi kompromi memiliki arti bahwa sebuah solusi yang layak atau yang mendekati solusi ideal, dimana kompromi berarti persetujuan yang dibuat dengan mengizinkan satu sama lain. Berikut prosedur perhitungan yang digunakan pada metode VIKOR seperti yang dilansir [9]:

1. Membuat matriks F, dimana  $A_i$  adalah alternatif dari  $i=1$  sampai  $n$  dan  $C_{xn}$  adalah kriteria dari  $j=1$  sampai  $m$ . Penyusunan matriks A dapat dilihat pada Persamaan (2.6).

$$F = \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{matrix} \begin{bmatrix} C_{x1} & C_{x2} & \dots & C_{xn} \\ X_{11} & X_{12} & \dots & X_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{m2} \end{bmatrix} \dots(2.6)$$

2. Menghitung normalisasi matriks dengan mengikuti Persamaan (2.7).

$$R_{ij} = \frac{(f_i^*) - (f_{ij})}{(f_i^*) - (f_i^-)} \dots(2.7)$$

Keterangan:

- $R_{ij}$  = nilai normalisasi matriks
  - $f_{ij}$  = nilai data sampel  $i$  kriteria  $j$
  - $f_i^*$  = nilai terbaik dalam suatu kriteria
  - $f_i^-$  = nilai terburuk dalam suatu kriteria
3. Mengkalikan nilai matriks yang sudah dinormalisasi dengan nilai bobot kriteria yang didapat dari metode AHP. Normalisasi bobot dapat dilihat pada Pesamaan (2.8).

$$\text{normalisasi bobot} = (W_i \times R_{ij}) \dots(2.8)$$

4. Menghitung *utility measures* dari setiap alternatif dengan menggunakan Persamaan (2.9) dan Persamaan (2.10).

$$S_i = \sum_{j=1}^n w_j \frac{(f_j^* - f_{ij})}{(f_j^* - f_j^-)}$$

...(2.9)

$$R_i = \text{MAX}_j \left[ w_j \frac{(f_j^* - f_{ij})}{(f_j^* - f_j^-)} \right]$$

...(2.10)

Dimana  $S_i$  (*maximum group utility*) dan  $R_i$  (*minimum individual regret of the opponent*) adalah *utility measures* yang diukur dari titik terjauh dan titik terdekat dari solusi ideal. Nilai  $w_i$  adalah bobot untuk setiap kriteria yang diperoleh dari perhitungan AHP.

5. Menghitung indeks VIKOR dari setiap alternatif  $i$  dengan menggunakan Persamaan (2.11).

$$Q_i = v \left[ \frac{S_i - S^*}{S^- - S^*} \right] + 1 (1 - v) \left[ \frac{R_i - R^*}{R^- - R^*} \right] \dots \dots(2.11)$$

Dimana nilai  $S^*$  adalah nilai terkecil dari  $S_i$ . Nilai  $S^-$  merupakan nilai terbesar dari  $S_i$ . Nilai  $R^*$  adalah nilai terkecil dari  $R_i$ , dan  $R^-$  nilai terbesar dari  $R_i$ .  $v$  adalah bobot yang berkisar antara 0 sampai 1 (umumnya bernilai 0,5). Semakin kecil nilai  $Q_i$  (VIKOR) maka semakin baik solusi alternatif yang ada.

D. Euclidean Distance

*Euclidean distance* diperkenalkan oleh Euclid yang merupakan seorang matematikawan asal Yunani. *Euclidean distance* digunakan untuk menghitung jarak antar 2 buah titik yang saling berhubungan antara sudut dan jarak, dengan menggunakan Persamaan (2.12) [10].

$$\text{Jarak}_{km} = \sqrt{\left( (Lat_A - Lat_B) \times 111,319 \right)^2 + \left( (Long_A - Long_B) \times 111,319 \right)^2} \dots \dots(2.12)$$

Keterangan:

- $Lat_A$  = Titik koordinat *latitude* awal
- $Lat_B$  = Titik koordinat *latitude* akhir
- $Long_A$  = Titik koordinat *longitude* awal
- $Long_B$  = Titik koordinat *longitude* akhir
- 111,319 = Hasil konversi 1 derajat bumi dalam km

E. System Usability Scale

*System Usability Scale* atau SUS merupakan metode yang digunakan untuk menilai kegunaan dari suatu sistem yang terdiri dari 10 buah pertanyaan dengan 5 pilihan jawaban antara sangat setuju sampai sangat tidak

setuju. Metode ini memiliki keuntungan dimana sangat mudah untuk menentukan jumlah partisipan, dapat digunakan dalam sampel yang sedikit, dan hasil yang diberikan valid dengan menyimpulkan dapat digunakan atau tidaknya suatu sistem [11]. Tabel 1 merupakan daftar pertanyaan dalam Bahasa Indonesia yang digunakan dalam metode SUS [12].

Tabel 1 Daftar Pertanyaan SUS

#	Pertanyaan	Skala
1	Saya pikir bahwa saya akan ingin lebih sering menggunakan aplikasi ini	1 s/d 5
2	Saya menemukan bahwa aplikasi ini, tidak harus dibuat serumit ini	1 s/d 5
3	Saya pikir aplikasi ini mudah untuk digunakan	1 s/d 5
4	Saya pikir bahwa saya akan membutuhkan bantuan orang teknis untuk dapat menggunakan aplikasi ini	1 s/d 5
5	Saya menemukan berbagai fungsi di aplikasi ini diintegrasikan dengan baik	1 s/d 5
6	Saya pikir ada terlalu banyak ketidaksesuaian dalam aplikasi ini	1 s/d 5
7	Saya bayangkan bahwa kebanyakan orang akan mudah untuk mempelajari aplikasi ini dengan sangat cepat	1 s/d 5
8	Saya menemukan, aplikasi ini sangat rumit untuk digunakan	1 s/d 5
9	Saya merasa sangat percaya diri untuk menggunakan aplikasi ini	1 s/d 5
10	Saya perlu belajar banyak hal sebelum saya bisa memulai menggunakan aplikasi ini	1 s/d 5

Untuk menghitung nilai SUS dapat dilakukan dengan memberikan nilai bobot dari setiap pertanyaan dimana untuk pertanyaan ganjil (1,3,5,7,9) nilai yang didapat adalah skala yang dipilih dikurangi 1, untuk pertanyaan genap (2,4,6,8,10) nilai yang didapat adalah 5 dikurangi skala. Selanjutnya semua nilai dari pertanyaan dijumlahkan dan dikalikan dengan 2,5 untuk mendapatkan nilai SUS dari masing-masing responden. Nilai SUS yang didapat memiliki rentang nilai 0-100. Jika nilai SUS berkisar 0-50,9 termasuk dalam kategori *not acceptable*, jika nilai SUS 51-70,9 maka termasuk dalam *marginal*, dan jika nilai SUS 71-100 maka termasuk dalam kategori *acceptable*

### III. METODOLOGI DAN PERANCANGAN

Pada bagian ini akan dijelaskan metode dan perancangan untuk sistem rekomendasi restoran menggunakan metode AHP dan VIKOR pada platform LINE.

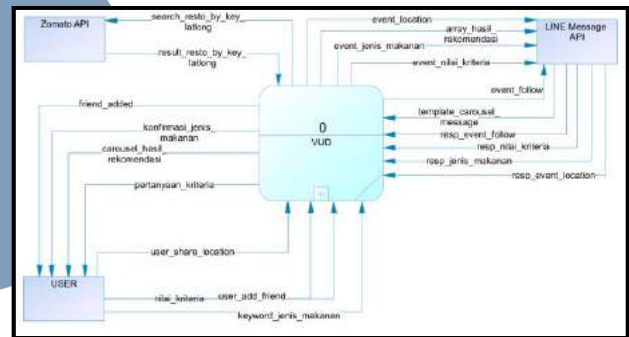
#### A. Teknik Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data restoran yang terdapat di *website* Zomato. Data tersebut akan diambil menggunakan Zomato API yang

disediakan oleh Zomato kepada pengembang dengan format JSON (*JavaScript Object Notation*). Data restoran yang digunakan adalah nama restoran, harga rata-rata untuk 2 orang, rata-rata *rating*, titik koordinat *latitude longitude*, nama daerah restoran, *link* url gambar utama restoran, dan *link* url restoran.

#### B. Perancangan dan Pembangunan

Tahap dalam perancangan dan pembangunan sistem dilakukan dengan merancang dan membangun sistem rekomendasi restoran menggunakan metode AHP dalam menghitung nilai bobot suatu kriteria dan metode VIKOR digunakan untuk memilih solusi terbaik dari pilihan alternatif. Dalam pembuatannya digunakan bahasa pemrograman PHP dan data restoran yang dimaksud akan diambil dari *website* Zomato menggunakan Zomato API. Selanjutnya akan diimplementasikan ke dalam platform LINE menggunakan *class-class* yang ada pada LINE Message API.



Gambar 2 Context Diagram VUD

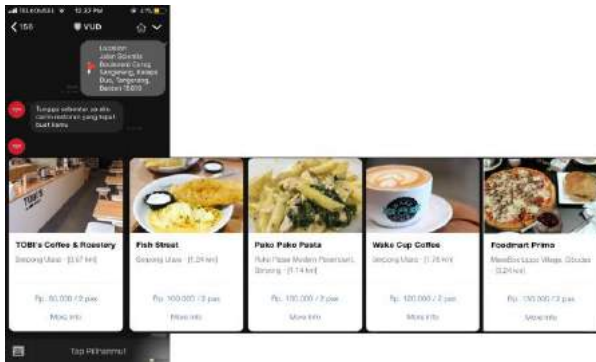
Gambar 2 merupakan Context Diagram dari sistem rekomendasi restoran, yang diberi nama VUD. Secara umum, sistem terhubung dengan Zomato API untuk mendapatkan informasi restoran. Sementara itu, sistem terhubung dengan LINE Messaging API untuk menyajikan informasi dalam platform LINE.

## IV. IMPLEMENTASI DAN UJI COBA SISTEM

### A. Hasil Implementasi

Sistem rekomendasi restoran dapat digunakan pada smartphone dengan OS Android dan iOS yang sudah memiliki aplikasi LINE. Sistem ini dibuat dengan bantuan Zomato API sebagai sumber data restoran dan LINE Message API untuk menampilkan data restoran di platform LINE, yang dimana sistem ini dibangun menggunakan bahasa pemrograman PHP dan rekomendasi yang dihasilkan menggunakan metode AHP dan VIKOR.





Gambar 3 Hasil Implementasi Sistem pada Platform LINE

Gambar 3 merupakan tampilan akhir dalam merekomendasikan restoran kepada user dalam platform LINE. Daftar restoran yang ditampilkan berupa carousel message, yaitu pesan yang berisikan berbagai macam konten seperti button, gambar, teks, atau tombol yang dapat melakukan aksi, yang dimana semua konten tersebut diletakkan pada format scrolling horizontal.

B. Simulasi Perhitungan

Simulasi perhitungan yang dilakukan dengan pemilihan jenis makanan, nilai kriteria dan lokasi sebagai berikut:

- Jenis Makanan : Pasta
- Titik Koordinat Latitude User : -6.2571366
- Titik Koordinat Longitude User : 106.6166282

Sedangkan untuk nilai kriteria yang dipilih user adalah sebagai berikut:

- Harga : 4 (Sangat Penting)
- Lokasi : 3 (Cukup Penting)
- Rating : 3 (Cukup Penting)

Langkah pertama yang dilakukan adalah membuat matriks perbandingan berdasarkan nilai kriteria yang sudah ditentukan. Hasil matriks perbandingan dapat dilihat pada Tabel 2.

Tabel 2 Matriks Perbandingan Nilai Kriteria

	Harga	Lokasi	Rating
Harga	4/4	4/3	4/3
Lokasi	3/4	3/3	3/3
Rating	3/4	3/3	3/3

Selanjutnya adalah menormalisasikan matriks tersebut menggunakan Perasamaan 2.2 dan menghitung nilai bobot dengan Persamaan 2.3, dapat dilihat pada Tabel 3.

Tabel 3 Hasil Normalisasi Matriks dan Nilai Bobot

	Harga	Lokasi	Rating	Bobot
Harga	0.4	0.4	0.4	0.4
Lokasi	0.3	0.3	0.3	0.3
Rating	0.3	0.3	0.3	0.4

Kemudian mengkalikan nilai matriks dengan nilai bobot dan selanjutnya menghitung nilai t untuk mengetahui konsistensi nilai kriteria.

$$t = \frac{1}{3} \times \left( \frac{1,2}{0,4} + \frac{0,9}{0,3} + \frac{0,9}{0,3} \right) = 3$$

$$CI = \frac{3 - 3}{3 - 1} = 0$$

Nilai CI dikatakan konsisten jika hasil dari nilai CI=0 atau CI ≤ 0,1. Berdasarkan perhitungan yang sudah dilakukan, nilai CI yang di dapat adalah 0 sehingga nilai kriteria yang di pilih user konsisten dan dapat dilanjutkan dengan menggunakan metode VIKOR.

Tabel 4 Nilai Terbaik dan Terburuk Kriteria

Kriteria	Nilai Terburuk	Nilai Terbaik
Harga	350000	60000
Rating	0	4.7
Jarak	6.070157582808800	0.649156756948108

Tabel 4 digunakan untuk menghitung normalisasi matriks alternatif F. Setelah itu hasil dari normalisasi dikalikan dengan nilai bobot yang didapat pada metode AHP, sehingga mendapatkan hasil seperti yang ditunjukkan pada Tabel 5.

Tabel 5 Hasil Perkalian Matriks dengan Nilai Bobot

#	Nama Restoran	Harga	Jarak KM	Rating
1	TOBI's Coffee & Roastery	0.000000000000000	0.000000000000000	0.057446808510638
2	O! Fish	0.193103448275860	0.031555062033169	0.031914893617021
3	Pako Pako Pasta	0.055172413793103	0.033323747723401	0.102127659574470
4	Fish Sreut	0.055172413793103	0.038756650988738	0.082978723404255
5	Enigma	0.193103448275860	0.056995653808324	0.070212765957447
6	Wake Up Coffee	0.082758620689655	0.067447053175671	0.076595744680851
7	Pancious	0.400000000000000	0.091156676099671	0.038297872340426
8	Port Five Six	0.193103448275860	0.115511176728790	0.038297872340426
9	Pasta Kangen	0.055172413793103	0.127584585327710	0.300000000000000
10	B'Steak Grill & Pancake	0.331034482758620	0.131536201192660	0.031914893617021
11	Porto Bistreau	0.331034482758620	0.131536201192660	0.000000000000000
12	Foodmart Primo	0.096551724137931	0.149836490899330	0.076595744680851
13	Pan & Flip	0.124137931034480	0.160178896213490	0.070212765957447
14	My Warm Day	0.262068965517240	0.160906540668690	0.076595744680851
15	Nanny's Pavillon	0.262068965517240	0.197998405548310	0.063829787234043
16	Steavn Meat Shop	0.262068965517240	0.198971174874550	0.140425531914890
17	Onokabe	0.331034482758620	0.205496484649530	0.076595744680851
18	Canabeans	0.165517241379310	0.237967638620230	0.076595744680851
19	Pizza Marzano	0.262068965517240	0.296572528009190	0.057446808510638
20	Kushiya Monotagari	0.262068965517240	0.300000000000000	0.057446808510638

Selanjutnya, menghitung nilai Si, Ri, dan Qi berdasarkan nilai yang ada pada Tabel 5. Dengan demikian Tabel 6 berikut merupakan nilai akhir dari Qi yang didapat.

Tabel 1 Nilai Qi

#	Nama Restoran	Qi
1	TOBI's Coffee & Roastery	0.00000000000000
2	O! Fish	0.41851646577572
3	Pako Pako Pasta	0.18743586585579
4	Fish Streat	0.17523448689133
5	Enigma	0.47521632649203
6	Wake Up Coffee	0.25410114982006
7	Pancious	0.91988418582682
8	Port Five Six	0.49887946298896
9	Pasta Kangen	0.44730893833632
10	B'Steak Grill & Pancake	0.80256992254865
11	Porto Bistreau	0.77417937301204
12	Foodmart Primo	0.35690368538765
13	Pan & Flip	0.41944851060748
14	My Warm Day	0.72088709120110
15	Nanny's Pavillon	0.74252670210758
16	Steavn Meat Shop	0.81152936794211
17	Onokabe	0.90810958209161
18	Canabeans	0.58285914893036
19	Pizza Marzano	0.82453722874522
20	Kushiya Monotagari	0.82758620689655

Setelah mendapatkan nilai Qi, maka selanjutnya adalah mengurutkan nilai Qi dari yang terkecil hingga terbesar. Dengan demikian nilai Qi yang terkecil merupakan pilihan alternatif terbaik dari kriteria yang ada. Tabel 7 berikut ini adalah daftar restoran yang memiliki nilai Qi terkecil untuk ditampilkan ke *platform* LINE.

Tabel 2 Hasil Rekomendasi Restoran pada LINE

#	Nama Restoran	Kawasan Restoran	Harga	Rating	Jarak KM
1	TOBI's Coffee & Roastery	Serpong Utara	60000	3.8	0.6491684199
2	Fish Streat	Serpong Utara	100000	3.4	1.3495137924
3	Pako Pako Pasta	Ruko Pasar Modern Para..	100000	3.1	1.2513394516
4	Wake Up Coffee	Serpong Utara	120000	3.5	1.8679587533
5	Foodmart Primo	MaxxBBox Lippo Village, ...	130000	3.5	3.3567628677

C. Uji Kepuasan Pengguna

Pengujian kepuasan pengguna menggunakan metode SUS (*System Usable Scale*) dilakukan dengan menyebarkan kuesioner secara *online*. Kuesioner yang disebar secara *online* mendapatkan respon sebanyak 31 responden. Berdasarkan aturan SUS untuk pertanyaan ganjil adalah nilai yang didapat adalah nilai skala dikurangi 1 sedangkan pertanyaan genap nilai yang didapat adalah 5 dikurangi dengan skala. Selanjutnya untuk mengetahui nilai keseluruhan dilakukan dengan menjumlahkan semua nilai pada pertanyaan 1 sampai 10 dan kemudian dikalikan dengan 2,5. Berikut hasil perhitungan SUS dapat dilihat pada Tabel 8.

Tabel 3 Perhitungan Kuesioner SUS

#	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Total	Total * 2.5
R1	3	3	4	4	4	2	3	3	4	3	33	82.5
R2	3	2	3	2	3	2	3	2	3	2	25	62.5
R3	3	0	2	1	4	2	3	2	3	1	21	52.5
R4	3	1	3	1	3	1	3	1	3	1	20	50
R5	4	1	3	3	3	3	3	3	3	3	29	72.5
R6	3	2	3	3	3	3	4	2	4	3	30	75
R7	3	1	3	1	3	1	3	1	3	1	20	50
R8	2	1	2	2	1	2	3	2	1	2	18	45
R9	4	3	4	3	4	3	3	3	4	3	34	85
R10	3	2	3	2	3	2	3	2	3	2	25	62.5
R11	2	2	3	4	3	3	2	3	2	4	28	70
R12	3	3	3	4	3	3	3	4	2	4	32	80
R13	3	3	3	3	3	3	3	3	3	1	28	70
R14	3	2	3	2	4	3	3	3	4	1	28	70
R15	2	2	2	2	2	2	2	2	2	2	20	50
R16	3	3	3	4	4	2	3	4	3	4	33	82.5
R17	2	4	3	4	4	2	3	3	3	4	32	80
R18	2	4	3	3	3	4	3	2	2	3	29	72.5
R19	3	3	3	4	3	4	2	3	3	3	31	77.5
R20	3	4	3	3	3	4	3	4	3	2	32	80
R21	2	1	3	3	3	3	3	3	1	3	25	62.5
R22	3	3	4	3	3	2	3	3	3	4	31	77.5
R23	3	4	4	3	3	2	3	4	2	3	31	77.5
R24	3	4	3	3	3	2	4	4	3	3	32	80
R25	3	3	4	3	3	2	4	4	3	2	31	77.5
R26	4	3	3	3	3	4	3	4	3	4	34	85
R27	4	4	3	3	4	2	3	3	3	4	33	82.5
R28	4	3	3	2	2	2	2	2	2	2	24	60
R29	3	4	3	3	3	4	3	3	3	3	32	80
R30	3	3	4	4	4	2	2	2	3	2	29	72.5
R31	4	4	3	3	2	4	2	4	2	4	32	80

Pada Tabel 8 didapatkan nilai rata-rata SUS sebesar 71.13 sehingga termasuk dalam kategori *acceptable*. Dengan demikian dapat disimpulkan bahwa sistem rekomendasi restoran menggunakan metode AHP dan VIKOR pada *platform* LINE berguna untuk membantu *user* dalam mencari restoran

V. KESIMPULAN DAN SARAN

Simpulan yang didapatkan dalam penelitian yang dilakukan adalah sistem rekomendasi restoran menggunakan metode AHP dan VIKOR pada *platform* LINE telah berhasil dirancang dan dibangun menggunakan bahasa pemrograman PHP. Data restoran yang digunakan diambil menggunakan Zomato API berupa nama restoran, alamat, titik koordinat *latitude* dan *longitude*, harga, dan *rating*. Metode AHP digunakan untuk menghitung nilai bobot pada kriteria dan metode VIKOR digunakan untuk menghitung nilai indeks berdasarkan data yang diambil, sehingga didapatkan hasil rekomendasi restoran berdasarkan nilai

indeks yang sudah terurut dari yang memiliki nilai terkecil sampai yang terbesar. Hasil rekomendasi restoran tersebut akan ditampilkan dalam bentuk *carousel message* di platform LINE menggunakan *LINE Message API*. Berdasarkan hasil kuesioner yang dilakukan dengan menggunakan metode SUS didapatkan hasil dengan rata-rata 71,13, dimana memiliki arti bahwa sistem rekomendasi yang dibuat berguna untuk membantu *user* dalam mencari restoran.

Adapun saran terkait dengan penelitian selanjutnya adalah menambahkan fitur *filtering* terhadap jenis makanan yang ada pada Zomato dengan jenis makanan yang dicari user. Penambahan fitur tersebut bertujuan untuk menghilangkan *tag* jenis makanan yang tidak sesuai dengan restoran yang ada pada Zomato. Selain itu penggunaan algoritma *Natural Language Processing* (NLP) yang bertujuan agar sistem dapat mengenali Bahasa keseharian pengguna agar tidak bergantung pada *keyword* yang ditentukan, sehingga pengguna dapat melakukan interaksi langsung dengan sistem seperti sedang melakukan *chatting* dengan bahasa sehari-hari.

#### REFERENSI

- [1] Kompas.com. "SBY Sebut Gaya Hidup Pengaruhi Ketahanan Pangan". 2016. [online] Tersedia dalam: <http://nasional.kompas.com/read/2016/05/14/15164921/SBY.Sebut.Gaya.Hidup.Pengaruhi.Ketahanan.Pangan> [Diakses 26 September 2017].
- [2] CNN Indonesia. "Gaya Hidup Masyarakat Menjadikan Bisnis Kuliner Menjanjikan". 2017. [online]. Tersedia dalam: <https://www.cnnindonesia.com/gayahidup/2017011812140526287137/gaya-hidup-masyarakat-menjadi-kanbisnis-kuliner-Menjanjikan> [diakses 26 September 2017].
- [3] Tech in Asia Indonesia. "[Update] Kumpulan Aplikasi dan Startup Kuliner Terbaik di Indonesia". 2017. [Online] Tersedia dalam: <https://id.techinasia.com/16-aplikasi-dan-startup-kuliner-di-indonesia> [Diakses 26 September 2017].
- [4] Metrotv News. "LINE: Indonesia Peringkat 4 Pengguna Terbanyak". 2016. [Online] Tersedia dalam: <http://teknologi.metrotvnews.com/news-teknologi/akW4MmLK-line-indonesia-peringkat-4-pengguna-terbanyak> [Diakses 1 Oktober 2017].
- [5] Purnomo, E. N. S., Sihwi, S. W. dan Anggrainingsih, R. "Analisis Perbandingan Menggunakan Metode AHP, TOPSIS, dan AHP-TOPSIS dalam Studi Kasus Sistem Pendukung Keputusan Penerimaan Siswa Program Akselerasi". *Jurnal ITSMART*, 2013, hal. 16-23.
- [6] Lengkong, S. P., Permanasari, A. E. dan Fauziati, S. "Implementasi Metode VIKOR untuk Seleksi Penerima Beasiswa". *Proceedings of The 7th National Conference on Information Technology and Electrical Engineering*, 2015, hal. 107-112.
- [7] Lengkong, S. P., Permanasari, A. E. dan Fauziwati, S. "Hybrid Multi Attribute Decision Making untuk Seleksi Penerima Bantuan Studi". *Seminar Nasional Teknologi Informasi 2015*. 2015
- [8] LINE. "Messaging API". 2017. [Online] Tersedia dalam: <https://developers.line.me/en/services/messaging-api/>
- [9] Venkata, R. R., (2007). "Decision Making in the Manufacturing Environment". United Kingdom: Springer London Ltd.
- [10] Imanuwelita, V., Putri, R. R. M. dan Amalia, F. "Penentuan Kelayakan Lokasi Usaha Franchise Menggunakan Metode AHP dan VIKOR". *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2015, hal. 122-132.
- [11] Prasetyo, A. D. dan Rahmad, C. "Perancangan Aplikasi Schedule Reminders Menggunakan Metode Euclidean Distance Pada Sistem Operasi Android". *Seminar Informatika Aplikatif Polinema*, Vol. III, 2015, hal. 126-128.
- [12] Brooke, J. "SUS: A Retrospective". *Journal of Usability Studies (JUS)*, 2013, hal. 29-40
- [13] Pudjoatmodjo, B. dan Wijaya, R. "Tes Kegunaan (Usability Testing) pada Aplikasi Kepegawain dengan Menggunakan System Usability Scale (Studi Kasus: Dinas Pertanian Kabupaten Bandung)". *Jurnal Seminar Nasional Teknologi Informasi dan Multimedia 2016*, hal. 37-42.

# Pendeteksian Ruang Kosong Parkir di dalam Ruangan

Nunik Afriliana<sup>1</sup>, Rosalina<sup>2</sup>, Regina Valeria<sup>3</sup>

<sup>1</sup> Fakultas Teknik & Informatika: Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia

<sup>2,3</sup> Faculty of Computing, President University, Jababeka Education Park, Cikarang Baru, Bekasi

nunik@umn.ac.id

rosalina@president.ac.id

reginavaleria96@gmail.com

Diterima 6 Mei 2018

Disetujui 25 Juni 2018

**Abstrak**— Menemukan tempat parkir kosong di tempat parkir dalam ruangan seperti pusat perbelanjaan menjadi kesulitan banyak pengemudi, terutama saat jam sibuk di kota-kota besar. Dalam makalah ini, sebuah sistem deteksi kekosongan tempat parkir dalam ruangan diusulkan, dengan menggunakan sistem kamera yang melibatkan OpenCV untuk mempercepat waktu dalam mencari tempat parkir bagi pengemudi kendaraan dengan memberi mereka informasi lokasi dan tempat parkir. Sistem ini menggunakan metode deteksi objek statis, yaitu *Haar-Like Cascade Classifier* yang dikombinasikan dengan *Hough Line Detection* untuk mengidentifikasi area parkir kosong dari gambar parkir yang diambil secara *real time* melalui kamera IP atau kamera USB. Sistem ini dirancang untuk disematkan dengan sistem manajemen parkir sebuah bangunan sebagai alat yang menyediakan tempat parkir untuk membantu pengemudi kendaraan memasuki area parkir.

**Index Terms**—*Haar-Like Cascade Classifier, Hough Line Detection, Sistem Manajemen Parkir*

## I. PENDAHULUAN

Meningkatnya penambahan volume kendaraan berdampak pada kebutuhan akan sarana parkir. Banyak pengemudi menghadapi kesulitan dalam menemukan tempat parkir yang tersedia pada saat berkunjung ke suatu tempat. Bahkan ketika hampir semua pusat perbelanjaan besar dan tempat-tempat umum lainnya memiliki gedung parkir bertingkat tersendiri, semakin sulit untuk menemukan tempat parkir yang tersedia terutama pada jam sibuk dimana bangunan parkir dalam ruangan hampir selalu penuh. Sebagian besar waktu, pengemudi harus mengitari bangunan parkir mobil untuk menemukan tempat kosong untuk memarkirkan mobil mereka.

Informasi mengenai lahan kosong parkir baik di pusat perbelanjaan maupun gedung perkantoran masih sangat minim, dimana umumnya sistem parkir yang digunakan masih manual.

Sistem deteksi kekosongan tempat parkir yang sudah ada menggunakan sensor ultrasonik dengan indikator LED terpasang di atas setiap lot di area

parkir keseluruhan. Jumlah besar perangkat keras yang dibutuhkan cukup mahal dan pengguna masih harus mengelilingi tempat parkir sambil mencoba melihat indikator tempat kosong. Pendekatan alternatif dalam menemukan tempat parkir kosong menggunakan kamera pengawas dengan algoritma penglihatan komputer untuk mendeteksi apakah terdapat mobil di area parkir kosong. Selanjutnya, sistem tersebut memberitahukan tempat mana yang tersedia dan menginformasikannya ke pengendara yang masuk. Fokus utamanya adalah menghemat banyak waktu dan usaha dengan memberikan informasi tentang ketersediaan tempat parkir.

Menemukan ruang kosong untuk memarkir mobil di area parkir dalam ruangan sering menjadi masalah, terdapat beberapa penelitian tentang teknik penentuan posisi mobil seperti yang tercantum dalam referensi [2,3,4,5,6,7,8, 11].

Liu [3] mengusulkan navigasi dalam ruangan berdasarkan *smartphone*, sistem ini dapat menemukan tempat parkir gratis dengan menerima data dari sensor yang terletak di dalam tempat parkir mobil.

Deteksi parkir kosong berbasis video disajikan di [4], proyek ini dievaluasi oleh beberapa kombinasi ekstraktor fitur dan algoritma pembelajaran mesin. Sementara Ichihashi et al [2] mengusulkan pendekatan untuk mengklasifikasikan ruang parkir kosong tunggal berdasarkan logika fuzzy.

Tulisan ini bertujuan untuk membangun dan menerapkan sistem yang akan membantu mengurangi waktu yang dibutuhkan oleh pengemudi dalam mencari tempat parkir kosong dan menghindari perjalanan yang tidak perlu melalui tempat parkir yang penuh.

Sistem ini dirancang untuk memberi informasi tentang lokasi yang tepat pada sebuah tempat parkir kosong. Lokasi harus diperoleh dengan sistem cerdas penglihatan komputer dan informasinya akan diarahkan ke pengemudi. Penelitian ini bertujuan untuk membuat dan menerapkan sistem yang akan membantu mengurangi waktu yang dibutuhkan



pengemudi dalam mencari tempat parkir kosong dan menghindari perjalanan yang tidak perlu melalui tempat parkir yang penuh.

Tujuan dari penelitian ini adalah untuk membuat sebuah aplikasi yang mengimplementasikan algoritma untuk melakukan fitur ekstraksi dan menggunakan *cascade classifier* untuk membangun deteksi objek. Hal ini dilakukan dengan tujuan membuat aplikasi *web* yang bisa menemukan tempat parkir kosong di suatu daerah.

Agar aplikasi ini bisa berfungsi dengan baik, lingkungan harus memenuhi persyaratan area parkir standar yang terdiri dari sederet slot parkir linier dengan garis yang memisahkan masing-masing slot.

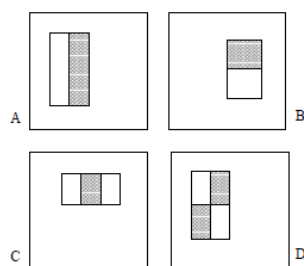
Keterbatasan lainnya adalah:

1. Keakuratan deteksi objek tergantung pada kualitas kamera.
2. Aplikasi ini hanya memproses gambar dari kamera statis sehingga area jangkauannya terbatas.
3. Implementasi deteksi objek hanya akan bekerja untuk mobil.

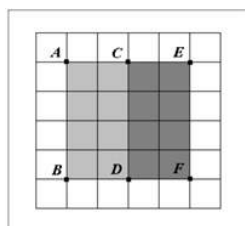
## II. STUDI PUSTAKA

### 2.1 Haar-Like Feature

Pada tahun 2001, Paul Viola dan Michael Jones mempublikasikan pendekatan deteksi wajah yang melibatkan fitur mirip Haar [9]. Fitur mirip-Haar terdiri dari daerah persegi panjang yang berdekatan di lokasi tertentu di jendela deteksi, di mana persegi panjang mewakili subbagian objek. Intinya adalah mengidentifikasi objek berdasarkan nilai fitur dan bukan pada nilai piksel individual dari objek gambar, sehingga menghasilkan eksekusi yang lebih cepat.



Gambar 1. Contoh fitur persegi panjang yang ditampilkan relatif terhadap jendela pendeteksian terlampir [8].



Gambar 2. Perhitungan nilai fitur Haar

Nilai Fitur (ABFE) = jumlah piksel di bawah persegi (CDFE) - jumlah piksel di bawah persegi (ABCD) (1)

Gambar 2. menunjukkan contoh fitur *Haar-like*. Penghitungan nilai fitur dilakukan menggunakan rumus (1), di mana jumlah piksel di bawah area gelap dikurangi dengan jumlah piksel di area terang. Jika nilai fitur ABFE berada di atas ambang batas (disetel selama pembelajaran), maka area di bawahnya dikatakan memenuhi persyaratan fitur *Haar*.

### 2.2 Canny Edge Detection

Deteksi tepi dilakukan umumnya untuk secara signifikan mengurangi jumlah data dalam suatu gambar, sambil tetap menjaga properti struktural untuk pemrosesan gambar lebih lanjut. Deteksi Tepi Canny dikembangkan oleh John F. Canny pada tahun 1986, dan telah menjadi salah satu metode deteksi tepi standar yang digunakan dalam penelitian [11].

Algoritma berjalan dalam 5 langkah terpisah:

- Smoothing: Terapkan filter Gaussian ke gambar untuk menghilangkan noise.
- Mencari gradien: Tepi-tepinya harus ditandai di mana gradien gambar memiliki besaran besar.
- Penekanan yang tidak maksimal: Konversi tepi yang kabur dari gradien menjadi tajam.
- Ambang ganda: Bedakan antara tepi benar yang potensial dan piksel tepi lainnya yang disebabkan oleh suara atau variasi warna.
- Pelacakan tepi dengan histeresis: Tentukan tepi akhir dengan menekan semua sisi yang tidak terhubung ke tepi (kuat) tertentu.

### 2.3 Hough Line Transform

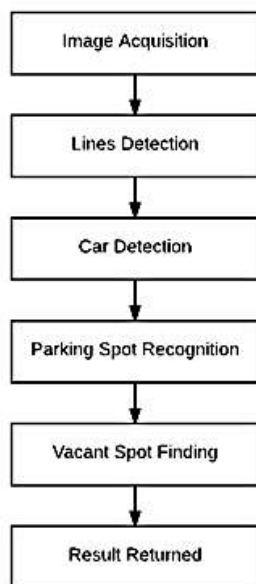
*Hough Line Transform* adalah algoritma untuk menemukan titik-titik yang sejajar dalam gambar yang menciptakan garis lurus. Suatu garis dapat direpresentasikan sebagai  $\rho = x \cos\theta + y \sin\theta$  di mana  $\rho$  adalah jarak dari asal ke garis, dan  $\theta$  adalah sudut yang dibentuk oleh garis jarak dan sumbu horizontal ini. Dalam mencari garis, *Hough Transform* berfungsi dengan menghitung semua kemungkinan kombinasi parameter untuk setiap piksel yang merupakan bagian dari garis. Setiap garis yang mungkin dari *pixel* pertama diwakili dengan kurva, dan titik dengan kurva persimpangan paling mengidentifikasi garis imajiner dalam gambar aslinya.

## III. METODOLOGI PENELITIAN

Proses pencari tempat parkir kosong terdiri dari beberapa urutan langkah seperti yang ditunjukkan pada Gambar 3.

Pertama, program akan mengaktifkan kamera, menangkap gambar dan menyimpannya ke folder sumber daya di server. Gambar kemudian akan dimuat ke dalam program sebagai tipe data Matriks.

Kedua, sistem akan melakukan deteksi objek untuk menemukan lokasi garis-garis pada gambar, yang akan menunjukkan partisi area parkir. Bagian pencarian baris terdiri dari beberapa langkah, yang mengkonversi ke *grayscale*, *blurring*, deteksi tepi, dan operasi *HoughLine*. Konversi warna dari RGB ke abu-abu dilakukan untuk menormalkan nilai warna di setiap piksel. *Gaussian blur* digunakan untuk menghaluskan objek gambar. Citra biner dihasilkan oleh deteksi tepi *Canny*, di mana *HoughLine* di OpenCV dapat diterapkan untuk mendeteksi garis.



Gambar 3. Urutan proses yang dilakukan oleh sistem

Ketiga, sistem akan melakukan deteksi objek pada gambar, untuk menemukan lokasi objek mobil di dalamnya. Untuk deteksi objek, program menggunakan *Haar Cascade Classifier* dengan OpenCV. File pengklasifikasi kaskade yang digunakan dalam makalah ini dibuat oleh Andrews Sobral [1] yang diterbitkan di GitHub. *Dataset* dan program Python untuk pelatihan *Haar-Cascade* termasuk dalam repositori publik. Berdasarkan dokumentasi resminya, *Haar-cascade "cars.xml"* dilatih menggunakan 526 gambar mobil dengan ukuran 360 x 240 piksel. Gambar-gambar itu diambil dari *dataset* Mobil yang diusulkan oleh Brad Philip dan Paul Updike yang diambil dari jalan raya di California selatan [1].



Gambar 4. Dataset Mobil

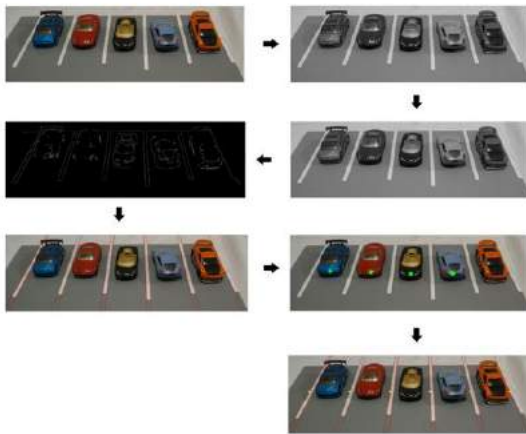
Dalam sistem ini, deteksi garis dilakukan terlebih dahulu, dan deteksi mobil datang setelah itu. Namun, kedua langkah ini dapat dipertukarkan, yang berarti bahwa urutan eksekusi tidak akan mempengaruhi hasilnya, selama hasilnya disimpan dalam variabel masing-masing.

Deteksi garis belum secara sempurna mendeteksi satu garis yang sebenarnya di dalam gambar oleh karena itu hasilnya tidak dapat langsung digunakan dan dihitung sebagai representasi sebenarnya dari garis parkir. Trik untuk menyelesaikan masalah ini adalah dengan memvalidasi garis untuk memperkirakan jarak minimum setiap baris sehingga dapat dianggap sebagai satu garis perwakilan. Dengan demikian, langkah keempat adalah pengenalan tempat parkir, di mana program akan melakukan validasi garis dan mengidentifikasi jumlah slot yang terdeteksi.

Pada langkah kelima, program akan membandingkan lokasi setiap mobil yang terdeteksi dengan slot untuk menemukan semua tempat kosong. Terakhir, sistem akan menampilkan hasilnya kepada pengguna.

Program ini memiliki satu servlet. Java servlet adalah program Java yang memperluas kemampuan server. Servlet bekerja dengan menangani permintaan HTTP dan respons dari browser dan memprosesnya. Dalam metode *doGet*, servlet pada dasarnya menerima permintaan, melakukan pekerjaan yang diperlukan, dan kemudian meneruskan permintaan ke halaman tujuan untuk pengguna.

Perpustakaan OpenCV tersedia di Java, didistribusikan dalam file JAR dan ekstensi .dll nya. Untuk dapat menggunakan perpustakaan, file OpenCV harus tersedia di mesin server dan jalur perlu ditentukan dalam IDE. Visualisasi proses ditunjukkan pada Gambar 5.

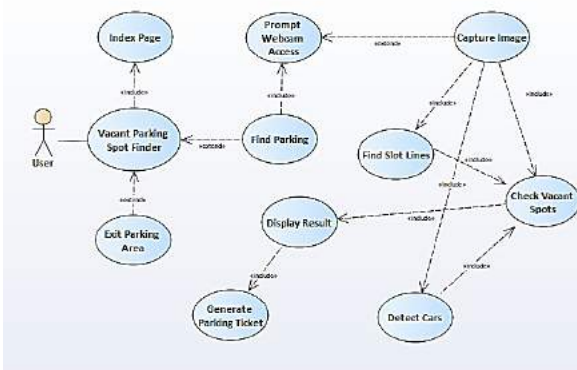


Gambar 5. Alur Proses dari Menemukan Ruang Parkir Kosong

IV. HASIL PENELITIAN

Sistem ini akan melakukan deteksi area dan deteksi objek untuk menemukan contoh objek dari kelas "mobil" dan untuk menemukan di mana ruang parkir kosong dan tersedia.

Ada beberapa fungsi Vacant Parking Spot Finder: (1) pengguna dapat menangkap gambar area parkir, (2) memperoleh data tentang berapa banyak mobil yang menempati area parkir, (3) mendapatkan informasi tempat parkir kosong, dan (4) menghasilkan tiket parkir. Diagram *use-case* dari sistem ini ditunjukkan pada Gambar 6.



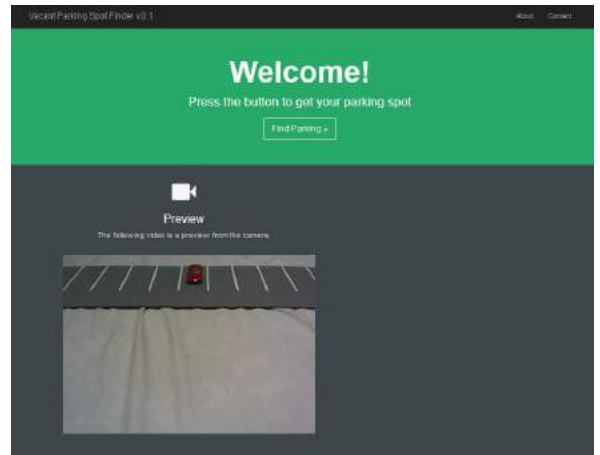
Gambar 6. Diagram Use-Case Vacant Parking Spot Finder

1. Inisiasi Pencarian Lahan Parkir

Ketika pengguna mengklik "Cari Parkir" di tombol menu, sistem akan memeriksa ketersediaan *webcam*. Jika *webcam* tidak tersedia, maka akan ada pemberitahuan kepada pengguna dan tombol untuk menangkap menjadi dinonaktifkan. Jika *webcam* tersedia, *webcam* akan diaktifkan dan ada kotak yang akan menampilkan video langsung dari *webcam*.

Inisialisasi aplikasi juga termasuk pengembangan file pengkodean kaskade Haar seperti yang akan

digunakan untuk deteksi. Fitur seperti Haar [8] adalah filter tepi yang dihitung menggunakan gambar integral.



Gambar 7. Halaman Utama Vacant Parking Spot Finder

2. Persiapan Pencarian Lahan Parkir

Aplikasi ini memiliki beberapa persiapan sebelum sistem mulai melakukan deteksi mobil untuk sistem parkir. Sebelum aplikasi melakukan deteksi mobil, pengguna harus menekan tombol untuk menangkap gambar area parkir. Setelah itu, beberapa pengolahan gambar akan dilakukan untuk mengidentifikasi area parkir, dimana area tersebut dibagi menjadi beberapa area yang ditandai dengan garis putih untuk memisahkan tempat parkir.

Sistem ini mengimplementasikan *Hough Lines Transform* untuk deteksi garis. Untuk menggunakan *Hough Lines Transform*, gambar yang diproses harus biner. Oleh karena itu, gambar warna asli diubah menjadi *grayscale*, sehingga setiap tepinya terdeteksi. Konversi dari gambar RGB ke abu-abu dilakukan dengan fungsi *cvtColor* OpenCV. Sedangkan deteksi tepi yang diimplementasikan dalam sistem ini adalah algoritma deteksi tepi Canny dengan OpenCV.

Untuk membiarkan program mengenali rangkaian garis yang terdeteksi sebagai kolom slot parkir, pertama-tama informasi lokasi mobil yang terdeteksi diproses dan rata-rata *y* dihitung untuk melihat di titik mana sepanjang garis yang terdeteksi dapat digunakan untuk perbandingan dengan lokasi mobil. Rata-rata dilakukan untuk menemukan nilai optimal dari koordinat *y* di mana kolom slot parkir dan mobil berada.

```
public static int findAverageY(List<Point> cars) {
    int sum = 0;
    for (int i = 0; i < cars.size(); i++) {
        sum += cars.get(i).y;
    }
    int avg = sum / cars.size();
    return avg;
}
```

Gambar 8. Fungsi Mencari Rata-Rata Y

Berdasarkan nilai  $y$  yang diperoleh dari `findAverageY`, program akan melintasi vektor yang berisi informasi  $(r, \theta)$  dari garis yang terdeteksi untuk menghitung titik akhir berdasarkan informasi tersebut. Setelah titik akhir dari garis ditemukan, `findXifYKnown` digunakan untuk menentukan satu titik di sepanjang garis di mana koordinat  $y$  sama dengan rata-rata  $y$  yang diperoleh dari `findAverageY`.

```
public static int findXifYKnown(int y, Point start, Point
end) {
    int x = (int) (start.x + ((y - start.y) * (end.x -
start.x) / (end.y - start.y)));
    return x;
}
```

Gambar 9. Fungsi Mencari X jika Y diketahui

Untuk setiap baris memiliki dua set titik akhir Titik A  $(x_A, y_A)$  dan Titik B  $(x_B, y_B)$ , adalah mungkin untuk menentukan Titik C baru di suatu tempat antara A dan B yang memenuhi  $y = y_{Avg}$ .

$$(y_B - y_A) / (x_B - x_A) = (y_C - y_A) / (x_C - x_A) \quad (1.1)$$

sehingga,

$$x_C = x_A + ((y_C - y_A) * (x_B - x_A) / (y_B - y_A)) \quad (1.2)$$

ArrayList of Point pertama diurutkan berdasarkan koordinat poin  $x$ , karena langkah selanjutnya akan membutuhkan daftar poin yang akan disortir mulai dari paling kiri ke kanan.

Dapat diamati bahwa hasil deteksi ini tidak sempurna. Diperlukan untuk menghilangkan kelebihan hasil menggunakan `validateLine`. Validasi baris dilakukan dengan memeriksa jarak dari satu baris yang terdeteksi ke baris berikutnya. Jika celah antara garis dan baris berikutnya ke kiri lebih besar dari minimum yang diperlukan, garis dianggap sah, dan disimpan ke dalam daftar `GarisLini` valid.

### 3. Proses Pencarian Lahan Parkir

Ada beberapa proses menemukan tempat parkir yang kosong saat melakukan deteksi objek. Aplikasi ini akan membaca *file* yang dilatih dengan *cascade*, dan menggunakannya dalam memindai gambar dan menemukan contoh mobil di dalamnya. Untuk setiap tempat parkir, kemunculan mobil akan menunjukkan bahwa tempat tersebut ditempati. Setelah semua tempat parkir diperiksa, aplikasi akan menemukan tempat kosong dan memberikan hasilnya kepada pengguna.

Ada beberapa aturan yang harus diikuti pengguna untuk mendapatkan hasil terbaik dalam proses deteksi mobil:

- Area parkir harus memiliki pencahayaan yang cukup untuk mendapatkan gambar yang layak.
- Mobil-mobil tidak boleh menutupi garis perbatasan yang membagi area parkir.

Untuk mendeteksi mobil, sistem ini menggunakan OpenCV. OpenCV dilengkapi dengan detektor *Haar-*

*cascade* yang dapat digunakan untuk menemukan objek dalam gambar. Fungsi `detectCars` menerima jalur gambar, memuat file pengklasifikasi kaskade yang telah ditentukan, dan memanggil fungsi `detectMultiScale`.

Fungsi `detectMultiScale` dari OpenCV mengembalikan matriks persegi panjang yang dapat diakses sebagai larik untuk mendapatkan koordinat sudut kotak pembatas yang berisi objek yang terdeteksi. Namun, ketika benar-benar ditarik, kotak pembatas biasanya terlihat lebih besar dari ukuran sebenarnya dari mobil, dan itu tidak diinginkan untuk proses selanjutnya untuk memeriksa slot parkir, karena koordinat mobil dan garis harus dibandingkan. Oleh karena itu, `findCenter` digunakan untuk menentukan koordinat pusat mobil. Pusat mobil ditunjukkan dengan warna hijau seperti yang ditunjukkan pada Gambar 9.

Pencari tempat parkir kosong yang diusulkan dalam makalah ini telah diuji. Ada tiga faktor utama untuk mengukur akurasi deteksi parkir saat melakukan pengujian. Pertama adalah kamera *web*, kamera harus mampu menghasilkan gambar yang layak, dengan resolusi setidaknya 640x480 piksel. Kamera *web* harus ditempatkan di lokasi sedemikian rupa sehingga dapat menangkap objek yang harus dimasukkan, yang dapat bervariasi sesuai dengan panjang fokus kamera.

Dalam pengujian ini kamera ditempatkan sekitar 1 meter dari objek. Kedua adalah cahaya, intensitas cahaya sangat mempengaruhi keakuratan pendeteksian, karena intensitas cahaya akan sangat mempengaruhi gambar yang diambil. Evaluasi mencakup beberapa variasi kondisi pencahayaan yang mungkin dari area parkir. Ketiga adalah objek, yaitu area parkir dan mobil, area parkir harus memiliki slot dibagi dengan garis, dan mobil tidak boleh mencakup lebih dari setengah dari garis.

Kasus uji untuk pengujian ketahanan sistem ini didasarkan pada beberapa ruang lingkup dan batasan untuk mengevaluasi kinerja aplikasi. Sistem ini tergantung pada lingkungan sekitarnya, terutama pencahayaan dan kamera. Dengan demikian, sistem ini juga akan dievaluasi oleh kondisi eksternal seperti intensitas cahaya yang berbeda dan jumlah slot parkir yang terlibat, yang akan dijelaskan pada Tabel 1.

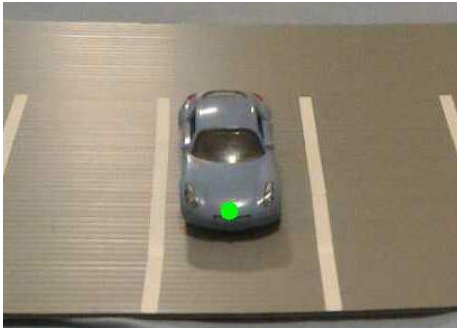
Tabel 1. Batasan Pengujian

No.	Batasan	Keterangan	Hasil
1	Cahaya	Terlalu terang atau terlalu gelap	Garis parkir tidak terdeteksi dengan jelas, mobil tidak terdeteksi
2	Jarak	Jarak mobil terlalu dekat dan terlalu jauh dari webcam	Mobil tidak terdeteksi



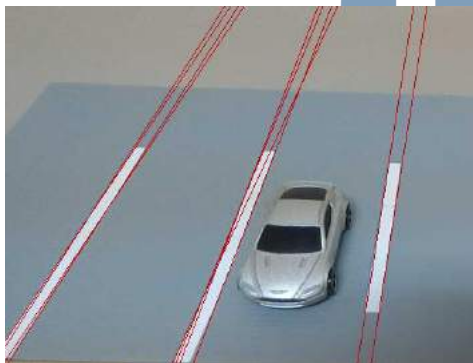
3	Objek selain mobil	Objek lain berada di lahan parkir	Program masih mendeteksi bahwa lahan parkir kosong
---	--------------------	-----------------------------------	--

Hasil simulasi deteksi mobil dapat dilihat pada Gambar 9. Titik hijau menunjukkan pusat area yang terdeteksi sebagai mobil.



Gambar 10. Hasil Deteksi Mobil

Tingkat akurasi deteksi mobil pada foto yang diambil dengan IP Camera (920px x 720px) lebih tinggi daripada tingkat akurasi deteksi mobil pada foto yang diambil dengan kamera web VGA (640px x 360px).



Gambar 11. Hasil Deteksi Garis

Hasil dari simulasi deteksi garis dapat dilihat pada Gambar 10. Garis merah menunjukkan elemen-elemen gambar yang terdeteksi sebagai garis.

Hasil akurasi pengujian sistem terlihat pada Tabel 2, dimana 72,7% sistem dapat mengenali ruang parkir kosong dan 27,3% sistem gagal mengenai ruang parkir kosong. Kegagalan sistem ini salah satunya dikarenakan kualitas kamera, kualitas cahaya dan jarak antar mobil.

Tabel 2. Akurasi Sistem

Jumlah Percobaan	Jumlah Deteksi	
	Benar	Salah
63	62.7%	37.3%

## V. KESIMPULAN

Penelitian ini bertujuan untuk menyediakan sistem untuk mendeteksi mobil di tempat parkir tertentu dan secara otomatis mengenali ruang sebagai deretan slot parkir, oleh karena itu tempat parkir kosong dapat ditemukan dengan tepat. Tujuan utama dari penelitian ini telah berhasil diraih. Pengguna dapat menggunakan proyek ini untuk menemukan tempat parkir kosong tanpa benar-benar memiliki informasi visual dari area parkir.

Metode deteksi mobil yang digunakan dalam makalah ini adalah dengan menggunakan *Haar-Like Cascade Classifier* dengan detektor *built-in* dari OpenCV. Deteksi objek bekerja dengan *file cars.xml* oleh Andrews Sobral dari GitHub. Dapat disimpulkan bahwa *file classifier* cukup terlatih dan oleh karena itu dapat digunakan untuk deteksi mobil.

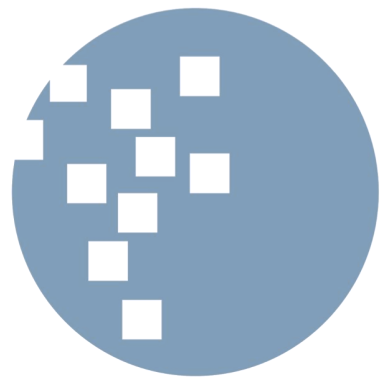
Kualitas gambar sangat mempengaruhi akurasi detektor objek; Oleh karena itu, kualitas kamera, kondisi objek dan lingkungan harus diperhatikan. Untuk tingkat tertentu, aplikasi ini kuat terhadap perubahan ukuran mobil dalam gambar, yang merupakan kasus ketika jarak kamera ke objek berubah.

Di masa depan, Deteksi Tempat Parkir Kosong akan memiliki peningkatan antarmuka pengguna, peningkatan deteksi mobil menggunakan beberapa file pengklasifikasi kaskade, deteksi objek waktu nyata, dan kemampuan untuk memproses input dari beberapa kamera untuk menutupi area yang lebih besar dan / atau menambah kemampuan gerakan ke kamera.

## DAFTAR PUSTAKA

- [1] A. Sobral, "GitHub," 3 December 2016. [Online]. Available: [https://github.com/andrewsobral/vehicle\\_detection\\_haarascades](https://github.com/andrewsobral/vehicle_detection_haarascades). [Accessed 17 September 2017]
- [2] Ichihashi, H., Katada, T., Fujiyoshi, M., Notsu, A. and Honda, K. "Improvement in the performance of camera based vehicle detector for parking lot". Proceedings of the IEEE International Conference on Fuzzy Systems. Pages 1-7, 2010
- [3] Liu J., Chen R., Chen Y., Pei L., Chen L. iParking: An Intelligent Indoor Location-Based Smartphone Parking Service. Sensors. 2012;12:14612–14629.
- [4] Marc tschentscher and marchel neuhaue, "Video Based Parking Space Detection", Institute for Neural Computation, Ruhr Universitat Bochum. 2012.
- [5] R. Yusnita, Fariza Norbaya, and Norazwinawati Basharuddin. "Intelligent Parking Space Detection System Based on Image Processing" International Journal of Innovation, Management and Technology, Vol. 3, No. 3, June 2012.
- [6] Sayanti Banerjee, Pallavi Choudekar and M. K. Muju. "Real time car parking system using image processing," IEEE, pp. 99-103, 2011.
- [7] Shen-En Shih and Wen-Hsiang Tsai. "A Convenient Vision-Based System for Automatic Detection of Parking Spaces in Indoor Parking Lots Using Wide-Angle Cameras". IEEE Transactions on Vehicular Technology, Volume: 63, Issue: 6, July 2014.
- [8] Viola, P. and Jones, M. "Rapid object detection using a boosted cascade of simple features". Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Vol. 1, Pages 511-518, 2001

- 
- [9] W. N. B. W. ISMAIL, "OBJECT DETECTION SYSTEM USING HAAR-CLASSIFIER"
- [10] Y. H. H. Z. a. W. S. F. Mai, "A hierarchical approach for fast and robust ellipse extraction," Pattern Recognition, vol. 41, no. 8, pp. 2512-2524, 2008.
- [11] Z. Bin, J. Dalin, W. Fang, and W. Tingting, A design of parking space detector based on video image, The Ninth International Conference On Electronic Measurement & Instruments ICEMI 2009.



UMN

# PEDOMAN PENULISAN JURNAL ULTIMATICS, ULTIMA INFOSYS, DAN ULTIMA COMPUTING

## 1. Kriteria Naskah

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

## 2. Pengetikan Naskah

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

## 3. Format Naskah

- Komposisi naskah terdiri dari Judul, Abstrak, Kata Kunci, Pendahuluan, Metode, Hasil Penelitian dan Pembahasan, Simpulan, Lampiran, Ucapan Terima Kasih, dan Daftar Pustaka.
- Judul memiliki jumlah kata maksimal 15 kata dalam Bahasa Indonesia atau maksimal 12 kata dalam Bahasa Inggris (termasuk subjudul bila ada).
- Abstrak ditulis dengan Bahasa Inggris paling banyak 200 kata, meskipun bahasa yang digunakan dalam penyusunan naskah adalah Bahasa Indonesia. Isi abstrak sebaiknya mengandung argumentasi logis, pendekatan pemecahan masalah, hasil yang dicapai, dan simpulan singkat.
- Kata Kunci ditulis dengan Bahasa Inggris dalam satu baris, dengan jumlah kata antara 4 sampai 6 kata.
- Pendahuluan berisi latar belakang dan tujuan penelitian.
- Metode dapat diuraikan secara terperinci dan dibedakan menjadi beberapa bab maupun subbab yang terpisah.
- Hasil dan Pembahasan disajikan secara sistematis sesuai dengan tujuan penelitian.
- Simpulan menyajikan intisari hasil penelitian yang telah dilaksanakan. Saran pengembangan untuk penelitian selanjutnya juga dapat diberikan di sini.

- Lampiran dan Ucapan Terima Kasih dapat dijabarkan setelah Simpulan secara singkat dan jelas.
- Daftar Pustaka yang dirujuk dalam naskah harus dituliskan di bagian ini secara kronologis berdasarkan urutan kemunculannya. Cara penulisannya mengikuti cara penulisan jurnal dan transaction IEEE.
- Template naskah telah disediakan dan dapat diminta dengan menghubungi surel redaksi.

## 4. Penulisan Daftar Pustaka

- Artikel Ilmiah:  
N. Penulis, "Judul artikel ilmiah," *Singkatan Nama Jurnal*, vol. x, no. x, hal. xxx-xxx, Sept. 2013.
- Buku  
N. Penulis, "Judul bab di dalam buku," di dalam *Judul dari Buku*, edisi x. Kota atau Negara Penerbit: Singkatan Nama Penerbit, tahun, bab x, subbab x, hal. xxx-xxx.
- Laporan  
N. Penulis, "Judul laporan," *Singkatan Nama Perusahaan, Kota Perusahaan, Singkatan Nama Negara, Laporan xxx, tahun*.
- Buku Manual/ *handbook*  
*Nama dari Buku Manual*, edisi x, *Singkatan Nama Perusahaan, Kota Perusahaan, Singkatan Nama Negara, tahun*, hal. xxx-xxx.
- Prosiding  
N. Penulis, "Judul artikel," di dalam *Nama Konferensi Ilmiah*, Kota Konferensi, *Singkatan Nama Negara (jika ada), tahun*, hal. xxx-xxx.
- Artikel yang Disajikan dalam Konferensi  
N. Penulis, "Judul artikel," disajikan di *Nama Konferensi, Kota Konferensi, Singkatan Nama Negara, tahun*.
- Paten  
N. Penulis, "Judul paten," HKI xxxxxx, 01 Januari 2014.
- Tesis dan Disertasi  
N. Penulis, "Judul tesis," M.Sc. thesis, *Singkatan Departemen, Singkatan*

Universitas, Kota Universitas, Singkatan Nama Negara, tahun.

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

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

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

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

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

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

## 5. Pengiriman Naskah Awal

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

## 6. Penilaian Naskah

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

## 7. Pengiriman Naskah Final

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

## 8. Copyright dan Honorarium

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

## 9. Biaya Tambahan

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

## 10. Alamat Redaksi

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



# Judul Paper

Sub Judul (jika diperlukan)

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

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

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

Diterima dd mmmmm yyyy

Disetujui dd mmmmm yyyy

**Abstract**—This electronic document is a “live” template which you can use on preparing your 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/m<sup>2</sup>” atau “webers per meter persegi,” bukan “webers/m<sup>2</sup>.”
- Gunakan angka nol di depan suatu bilangan desimal, seperti “0,25” bukan “.25.”

### C. Persamaan

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

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

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

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

#### D. Beberapa Kesalahan Umum

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

#### IV. MENGGUNAKAN TEMPLATE

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

- ULTIMATICS\_namaPenulis1\_judulArtikel.
- ULTIMAInfoSys\_namaPenulis1\_judulArtikel.
- ULTIMAComputing\_namaPenulis1\_judulArtikel.

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

##### A. Penulis dan Afiliasi

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

##### B. Penamaan Judul Bab dan Subbab

Bab merupakan suatu perangkat organisatorial yang memandu pembaca untuk membaca isi artikel

Anda. Terdapat dua jenis bab: bab utama (bab) dan subbab.

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

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

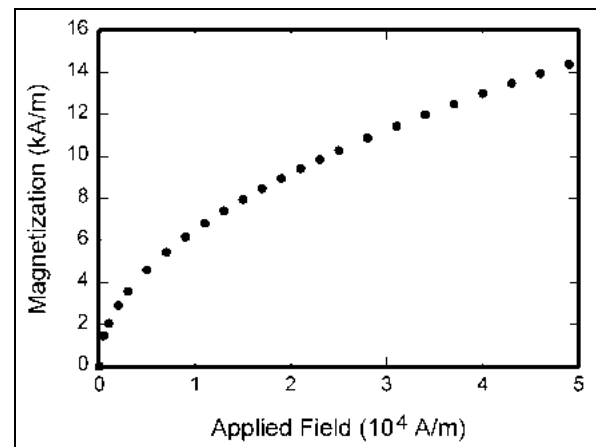
##### C. Gambar dan Tabel

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

Tabel 1. Contoh tabel

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

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



Gambar 1. Contoh gambar

## V. SIMPULAN

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

## LAMPIRAN

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

## UCAPAN TERIMA KASIH

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

## DAFTAR PUSTAKA

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

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

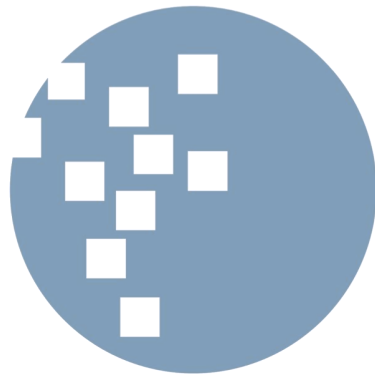
bawah kolom dimana catatan kaki tersebut dirujuk. Jangan letakkan catatan kaki di dalam daftar pustaka.

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

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

- [1] G. Eason, B. Noble, dan I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” *Phil. Trans. Roy. Soc. London*, vol. A247, hal. 529-551, April 1955.
- [2] J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, hal.68-73.
- [3] I.S. Jacobs dan C.P. Bean, “Fine particles, thin films and exchange anisotropy,” in *Magnetism*, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, hal. 271-350.
- [4] K. Elissa, “Title of paper if known,” belum terbit.
- [5] R. Nicole, “Title of paper with only first word capitalized,” *J. Name Stand. Abbrev.*, proses cetak.
- [6] Y. Yorozu, M. Hirano, K. Oka, dan Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” *IEEE Transl. J. Magn. Japan*, vol. 2, hal. 740-741, Agustus 1987 [Digests 9th Annual Conf. Magnetics Japan, hal. 301, 1982].
- [7] M. Young, *The Technical Writer’s Handbook*. Mill Valley, CA: University Science, 1989.

UMN



**UMN**



ISSN 2355-3286



9 772355 328009



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