

## Jurnal Ilmiah

# Informatika Komputer

- ANALISIS PERAMALAN TINGKAT KEMISKINAN DI INDONESIA DENGAN MODEL ARIMA 95  
**Rinaldo Isnawan Prasetyono, Dyah Anggraini**
- PENGELOMPOKAN WILAYAH PENYEBARAN COVID-19 DI KABUPATEN KARAWANG MENGGUNAKAN ALGORITMA K-MEANS 111  
**Zulia Imami Alfianti**
- PEMBUATAN PROTOTYPE SMART HOME MENGGUNAKAN NODEMCU ESP8266 V3 DAN CHAT BOT PADA SMARTPHONE ANDROID 123  
**Renaldi Dewangga Sindhu, Ilmiyati Sari, Dewi Putrie Lestari**
- IMPLEMENTASI FIESTEL BLOCK CIPHER DALAM ENKRIPSI FILE BERBENTUK TEKS 136  
**Ety Sutanty, Meilani B. Siregar, Esti Setyaningsih**
- IMPLEMENTASI ALGORITMA KLASIFIKASI SUPPORT VECTOR MACHINE UNTUK ANALISA SENTIMEN PENGGUNA TWITTER TERHADAP KEBIJAKAN PSBB 149  
**Heart Parasian PR Zuriel, Achmad Fahrurozi**
- EKSTRAKSI FITUR PEMBULUH DARAH CITRA FUNDUS RETINA MENGGUNAKAN FUZZY LOGIC 163  
**Asyaroh Ramadona Nilawati, Taufik Hidayat**
- KLASIFIKASI TUMOR JINAK DAN TUMOR GANAS PADA CITRA MAMMOGRAM MENGGUNAKAN GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) DAN SUPPORT VECTOR MACHINE (SVM) 176  
**Lulu Mawaddah Wisudawati**

## **DEWAN REDAKSI JURNAL ILMIAH INFORMATIKA KOMPUTER**

### **Penanggung Jawab**

Prof. Dr. E.S. Margianti, S.E., M.M.  
Prof. Suryadi Harmanto, SSI., M.M.S.I.  
Drs. Agus Sumin, M.M.S.I.

### **Dewan Editor**

Dr. Fitriarningsih, Universitas Gunadarma  
Dr. Rodiah, Universitas Gunadarma  
Dr. Ias Sri Wahyuni, S.Si, M.Si, Universitas Gunadarma

### **Mitra Bebestari**

Prof. Dr. Rer. Nat. A. Benny Mutiara, Universitas Gunadarma  
Prof. Dr. Achmad Nizar Hidayanto, Universitas Indonesia  
Prof. Dr.-Ing. Adang Suhendra, S.Si, S.Kom, M.Sc, Universitas Gunadarma  
Prof. Dr. Sarifuddin Madenda, Universitas Gunadarma  
Surya Sumpeno, PhD, Institut Teknologi Sepuluh Nopember (ITS)  
Noor Akhmad Setiawan, Ph.D, Universitas Gadjah Mada  
Dr. Paulus Insap Santosa, Universitas Gadjah Mada  
Dr. Agus Harjoko, M.Sc, Ph.D, Universitas Gadjah Mada  
Dr. Ernastuti, Universitas Gunadarma  
Dewi Agushinta R., Universitas Gunadarma  
Dr. Eri Prasetyo, Universitas Gunadarma  
Dr. Sunny Arief Sudiro, STMIK Jakarta STI&K  
Dr. Tubagus Maulana Kusuma, S.Kom.,Mengsc, Universitas Gunadarma  
Dr. Lussiana ETP, S.si., M.T, STMIK Jakarta STI&K  
Harry Budi Santoso, Ph.D, Universitas Indonesia

### **Sekretariat Redaksi**

Universitas Gunadarma  
Jalan Margonda Raya No. 100 Depok 16424  
Phone : (021) 78881112 ext 516.

# JURNAL ILMIAH INFORMATIKA KOMPUTER

NOMOR 2, VOLUME 26, AGUSTUS 2021

## DAFTAR ISI

ANALISIS PERAMALAN TINGKAT KEMISKINAN DI INDONESIA DENGAN MODEL ARIMA Rinaldo Isnawan Prasetyono, Dyah Anggraini	95
PENGELOMPOKAN WILAYAH PENYEBARAN COVID-19 DI KABUPATEN KARAWANG MENGGUNAKAN ALGORITMA K-MEANS Zulia Imami Alfianti	111
PEMBUATAN PROTOTYPE SMART HOME MENGGUNAKAN NODEMCU ESP8266 V3 DAN CHAT BOT PADA SMARTPHONE ANDROID Renaldi Dewangga Sindhu, Ilmiyati Sari, Dewi Putrie Lestari	123
IMPLEMENTASI FEISTEL BLOCK CIPHER DALAM ENKRIPSI FILE BERBENTUK TEKS Ety Sutanty, Meilani B. Siregar, Esti Setyaningsih	136
IMPLEMENTASI ALGORITMA KLASIFIKASI SUPPORT VECTOR MACHINE UNTUK ANALISA SENTIMEN PENGGUNA TWITTER TERHADAP KEBIJAKAN PSBB Heart Parasian PR Zuriel, Achmad Fahrurozi	149
EKSTRAKSI FITUR PEMBULUH DARAH CITRA FUNDUS RETINA MENGGUNAKAN FUZZY LOGIC Asyaroh Ramadona Nilawati, Taufik Hidayat	163
KLASIFIKASI TUMOR JINAK DAN TUMOR GANAS PADA CITRA MAMMOGRAM MENGGUNAKAN GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) DAN SUPPORT VECTOR MACHINE (SVM) Lulu Mawaddah Wisudawati	176

# ANALISIS PERAMALAN TINGKAT KEMISKINAN DI INDONESIA DENGAN MODEL ARIMA

<sup>1</sup>Rinaldo Isnawan Prasetyono, <sup>2</sup>Dyah Anggraini  
<sup>1,2</sup>Program Pasca Sarjana Universitas Gunadarma,  
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat  
<sup>1</sup>aldoisnawan18@gmail.com, <sup>2</sup>d\_anggraini@staff.gunadarma.ac.id

## Abstrak

Kemiskinan di Indonesia merupakan masalah yang kompleks dan multidimensi, karena tingkat kemiskinan di suatu negara akan mempengaruhi indikator keberhasilan baik dari segi pembangunan maupun perekonomian negara tersebut. Berdasarkan permasalahan tersebut diperlukan sebuah cara untuk mengetahui tingkat kemiskinan di Indonesia baik wilayah Perkotaan, Pedesaan maupun secara Nasional, salah satunya yaitu dengan menggunakan metode peramalan. Pada penelitian kali ini, peneliti menggunakan sebuah model dari Box Jenkins yaitu Auto Regressive Moving Average (ARIMA) untuk meramalkan tingkat kemiskinan di Indonesia pada masa yang akan datang. Dataset kemiskinan yang digunakan bersumber dari Badan Pusat Statistik (BPS) dengan data pengujian dari tahun 2011 hingga tahun 2020. Peneliti akan menggunakan 3 parameter error untuk mengevaluasi hasil tingkat kemiskinan di Perkotaan, Pedesaan maupun secara Nasional yaitu RMSE, MAE dan MAPE. Berdasarkan pengujian yang dilakukan bahwa dataset perkotaan menghasilkan model ARIMA(2,2,5) sebagai model ARIMA terbaik dengan RMSE=1.246582, MAE=0.923255 dan MAPE=12%, untuk dataset pedesaan menghasilkan model ARIMA(1,2,1) sebagai yang terbaik dengan RMSE=0.392650, MAE=0.311529 dan MAPE=2%. Sementara untuk dataset secara nasional menghasilkan model ARIMA(0,2,5) sebagai yang terbaik dengan RMSE=2.533166, MAE=2.090505 dan MAPE=20%. Dari 3 pengujian tersebut disimpulkan bahwa model ARIMA berhasil menghasilkan nilai peramalan tingkat kemiskinan di Indonesia baik wilayah Perkotaan, Pedesaan maupun secara Nasional dengan hasil yang baik

**Kata Kunci:** ARIMA, Kemiskinan, MAE, MAPE, RMSE

## Abstract

Poverty in Indonesia is a complex and multidimensional problem, because the level of poverty in a country will affect indicators of success both in terms of development and the country's economy. Based on these problems, a way is needed to find out the level of poverty in Indonesia, both in urban, rural and national areas, one of which is by using the forecasting method. In this study, the researcher uses a model from Box Jenkins, namely the Auto Regressive Moving Average (ARIMA) to predict the level of poverty in Indonesia in the future. The poverty dataset used is sourced from the Central Statistics Agency (BPS) with test data from 2011 to 2020. Researchers will use 3 error parameters to evaluate the results of poverty rates in urban, rural and national levels, namely RMSE, MAE and MAPE. Based on the tests conducted, the urban dataset produces the ARIMA(2,2,5) model as the best ARIMA model with RMSE=1.246582, MAE=0.923255 and MAPE=12%, for the rural dataset, the ARIMA(1,2,1) model as the the best with RMSE=0.392650, MAE=0.311529 and MAPE=2%. Meanwhile, for the national dataset, the ARIMA(0,2,5) model is the best with RMSE=2.533166, MAE=2.090505 and MAPE=20%. From these 3 tests, it is concluded that the ARIMA model has succeeded in producing forecasting values for poverty levels in Indonesia, both in urban, rural and national areas with good results.

**Keywords:** ARIMA, MAE, MAPE, Poverty, RMSE

## PENDAHULUAN

Indonesia termasuk salah satu negara berkembang dimana permasalahan pokok yang biasa dialami yaitu tentang kemiskinan. Dalam kehidupan bernegara masalah sosial seperti kemiskinan dapat mempengaruhi nilai suatu bangsa dalam mensejahterakan masyarakatnya. Kemiskinan menjadi suatu masalah yang kompleks dan multidimensi karena indikator keberhasilan pembangunan suatu negara tergantung pada tingkat kesejahteraan masyarakat [1]. Berdasarkan data dari Badan Pusat Statistik bahwa sejak tahun 2011 hingga tahun 2019 persentase tingkat kemiskinan di Indonesia mengalami penurunan, hanya pada tahun 2015 semester I bulan Maret ada kenaikan tingkat kemiskinan. Pada tahun 2020 tingkat kemiskinan di Indonesia naik secara signifikan [2]. Terlepas dari faktor eksternal dimana terjadi pandemi covid-19 yang melanda dunia tak terkecuali Indonesia, sudah seharusnya pemerintah mulai sadar terhadap permasalahan kemiskinan yang terjadi. Salah satu bentuk yang bisa dilakukan adalah dengan melakukan peramalan tingkat kemiskinan di Indonesia pada masa yang akan datang. Peramalan diperlukan untuk membantu *user* dalam proses perencanaan maupun pengambilan keputusan pada masa yang akan datang [3]. Peramalan berguna untuk perencanaan dan pengambilan keputusan, namun hal ini tidak selalu akurat, karena keakuratan proses peramalan ini tergantung dari data yang diperoleh maupun metode yang

digunakan. Penelitian kali ini menggunakan model *Auto Regressive Moving Average* (ARIMA). Model ARIMA merupakan kombinasi dari 2 model yaitu *Auto Regressive* (AR) dan *Moving Average* (MA). Model AR merupakan model persamaan regresi yang menghubungkan nilai variabel sebelumnya yang bersifat dependen, biasanya model ini ditulis dengan ordo  $p$  [4] dan model MA merupakan model yang menetapkan bahwa variabel *output* bergantung secara linear pada nilai saat ini dan berbagai nilai dimasa lalu dan ditulis dengan ordo  $q$  [5]. Beberapa penelitian yang pernah dilakukan untuk memprediksi kemiskinan maupun penggunaan model ARIMA telah dilakukan diantaranya Prediksi Tingkat Kemiskinan di Kalimantan Timur dengan Menggunakan *Single* dan *Double Exponential Smoothing*. Penelitian tersebut menghasilkan nilai MAPE pada model *Single Exponential Smoothing* sebesar 10.94% dan *Double Exponential Smoothing* sebesar 14.90%, sehingga dari perbandingan tersebut menunjukkan model *Single Exponential Smoothing* memiliki tingkat keakuratan paling baik dalam memprediksi kemiskinan di Kalimantan Timur [6]. Prediksi tingkat kemiskinan di Provinsi Jawa Barat menggunakan *Seasonal ARIMA* (SARIMA). Penelitian ini menghasilkan nilai MAPE sebesar 2.81% yang artinya model tersebut masuk kedalam kategori sangat baik dalam melakukan peramalan [7]. Penelitian mengenai prediksi tingkat kemiskinan di Provinsi NTB dengan menggunakan metode ARDL

(*Auto Regressive Distributed Lag*) dimana pada penelitian tersebut memiliki nilai MAPE sebesar 3% sehingga peramalan tersebut bisa dikategorikan sangat baik dan penggunaan metode ARDL cocok dalam memprediksi tingkat kemiskinan di wilayah tersebut [8]. Penelitian tentang memprediksi jumlah penduduk miskin di wilayah Yogyakarta dengan menggunakan model *Triple Exponential Smoothing*, dimana penelitian tersebut memiliki nilai MAPE sebesar 3% sehingga peramalan dikategorikan sangat baik [9]. Berdasarkan permasalahan diatas, maka untuk penelitian mengenai peramalan tingkat kemiskinan di Indonesia menggunakan model ARIMA. Metode ARIMA dipilih karena memiliki akurasi terbaik dalam memprediksi untuk penelitian jangka pendek [10]. Diharapkan dengan penggunaan model ARIMA ini mampu menghasilkan tingkat akurasi terbaik dalam memprediksi tingkat kemiskinan di Indonesia. Pada penelitian ini menggunakan model ARIMA dalam melakukan peramalan data kemiskinan di Indonesia berdasarkan wilayah Perkotaan, Pedesaan dan secara Nasional. Penelitian ini menggunakan data kemiskinan yang bersumber dari Badan Pusat Statistik tahun 2011 hingga 2020 dengan jumlah 60 *record* data dengan harapan mampu untuk meramalkan tingkat kemiskinan di Indonesia pada masa 4 tahun yang akan datang dari tahun 2021 hingga 2024. Hasil yang diharapkan dari penelitian ini yaitu mampu memberikan nilai akurasi peramalan yang baik dan nilai

kesalahan yang rendah sehingga mampu membantu pemerintah dalam mengidentifikasi tingkat kemiskinan yang terdapat di Indonesia berdasarkan wilayah baik itu Perkotaan, Pedesaan maupun dalam skala Nasional pada masa yang akan datang.

## METODE PENELITIAN

Penelitian yang dilakukan terdiri dari beberapa tahapan proses, hal ini dilakukan untuk mendapatkan akurasi terbaik dalam meramalkan tingkat kemiskinan di Indonesia dengan model *Auto Regressive Moving Average* (ARIMA). Gambar 1 merupakan gambaran umum dari tahapan penelitian yang dilakukan. Berdasarkan Gambar 1, tahapan awal penelitian dilakukan dengan menginput data kemiskinan di Indonesia, data yang digunakan bersumber dari Badan Pusat Statistik ([www.bps.go.id](http://www.bps.go.id)). Setelah menginput data, langkah berikutnya melakukan identifikasi model dengan menggunakan uji stasioneritas dan diferensiasi data untuk mendapatkan pendugaan model ARIMA ( $p,d,q$ ).

Tahapan berikutnya setelah didapatkan model ARIMA yang diduga yaitu melakukan estimasi parameter model dengan menggunakan ACF (*Autocorrelation Function*) dan PACF (*Partial Autocorrelation Function*).

Tahapan berikutnya dilanjutkan dengan pemeriksaan diagnostik dengan parameter AIC (*Akaike Information Criterion*) dan SBC (*Schwarz Bayes Criterion*) untuk melihat parameter model yang terbaik. Langkah

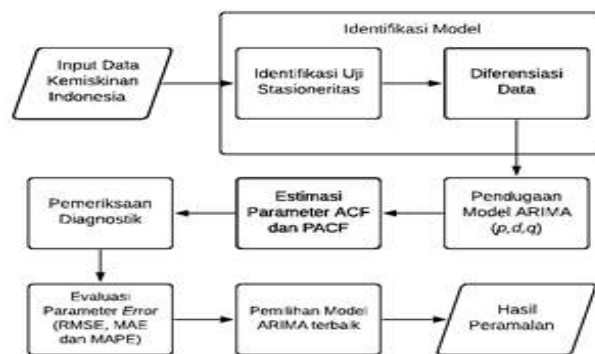
selanjutnya melakukan uji parameter *error* dengan 3 kategori yaitu menggunakan RMSE (*Root Mean Square Error*), MAE (*Mean Absolute Error*) dan MAPE (*Mean Absolute Percentage Error*) untuk dilihat model ARIMA yang paling kecil nilai *error* dan paling baik akurasi. Setelah didapatkan model ARIMA yang memiliki nilai akurasi terbaik dengan *error* terkecil barulah data tersebut bisa digunakan untuk proses peramalan.

### Input Data Kemiskinan di Indonesia

Pada tahapan ini data yang diinput berjumlah 102 data dari periode 1996 hingga 2020. Pengambilan data kemiskinan di Indonesia pada tahun 1996 hingga 2010 dilakukan setahun sekali, sedangkan pada tahun 2011 hingga 2020 pengambilan data

dilakukan per-6 bulan yaitu Maret dan September. Maka untuk memudahkan observasi dalam penelitian, digunakan data dari tahun 2011 hingga 2020 untuk peramalan dengan jumlah 60 *dataset*. Tabel 1 contoh sampel data pengujian tingkat kemiskinan di Indonesia.

Pada bulan Maret 2015 persentase tingkat kemiskinan di Indonesia wilayah perkotaan sebesar 8.29 %, sedangkan untuk wilayah pedesaan lebih besar yaitu sebesar 14.21 % dan dalam skala Nasional sebesar 11,22 %. Persentase tingkat kemiskinan dilakukan dalam satuan persen, dihitung berdasarkan sensus tiap tahunnya dari banyaknya jumlah penduduk yang berada dibawah garis kemiskinan dan rata-rata pengeluaran per kapita penduduk yang berada dibawah garis kemiskinan.



Gambar 1. Tahapan Model Penelitian

Tabel 1. Sampel Data Pengujian Tingkat Kemiskinan di Indonesia

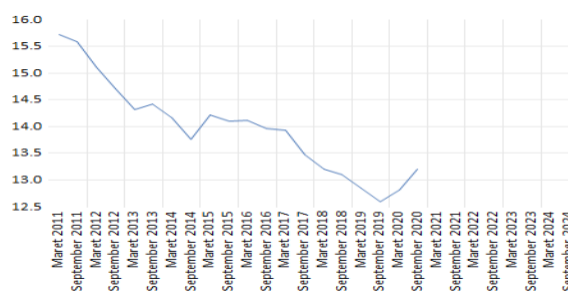
Bulan	Kota	Desa	Nasional
Maret 2011	9,23	15,72	12,49
September 2011	9,09	15,59	12,36
Maret 2012	8,78	15,12	11,96
September 2012	8,6	14,7	11,66
Maret 2013	8,39	14,32	11,37
September 2013	8,52	14,42	11,47
Maret 2014	8,34	14,17	11,25
September 2014	8,16	13,76	10,96
Maret 2015	8,29	14,21	11,22
September 2015	8,22	14,09	11,13

[Sumber : [www.bps.go.id](http://www.bps.go.id), 2021]

## Identifikasi Model

Pada tahap ini data yang diinput akan dilakukan pengecekan stasioner melalui pengujian unit *root* dengan menggunakan *Augmented Dickey-Fuller* (ADF) serta melakukan pengujian dengan level diferensiasinya. Pengujian stasioner dilakukan untuk menghasilkan nilai rata-rata dan varian yang konstan. Gambar 2 ditampilkan grafik tingkat kemiskinan di Indonesia. Berdasarkan Gambar 2 variabel X ditunjukkan dengan nama bulan dan variabel Y nilai rata-rata tingkat kemiskinan. Dari grafik tersebut terlihat data belum stasioner hal ini ditunjukkan dengan keadaan naik dan penurunan grafik masih belum konstan atau tidak stabil. Sebagai contoh pada rata-rata tingkat kemiskinan tahun 2014 dimana terlihat terjadi kenaikan grafik lalu kemudian terjadi sedikit naik turun grafik dari rentang 2015 hingga 2016 yang kemudian terjadi penurunan sangat signifikan pada periode 2017 hingga 2019. Maka dari itu dikatakan bahwa grafik pengujian pada

tingkat kemiskinan belum stasioner. Langkah selanjutnya untuk menjadikan data stasioner yaitu dengan pengujian diferensiasi dimana pengujian bisa dilakukan dengan kriteria *Akaike Info Criterion* (AIC) atau *Schwarz Info Criterion* (SBC). Pada pengujian stasioner, nilai probabilitas (*p-value*) harus lebih kecil dari nilai signifikan ( $\alpha$ ) 0.05 dan nilai *t-statistic* lebih kecil dari nilai kritis pengujian. Nilai signifikan ( $\alpha$ ) sebesar 0.05 merupakan nilai batas maksimal dari kesalahan yang dilakukan dalam penelitian, nilai probabilitas (*p-value*) merupakan besarnya peluang dalam statistik pengujian yang dilakukan dan nilai *t-statistic* merupakan sebuah nilai yang digunakan untuk menguji kebenaran dalam sample yang digunakan pada penelitian [11]. Dalam pengujian diferensiasi dilakukan 3 kriteria yaitu Level (Level 0), 1<sup>st</sup> Difference (Level 1) dan 2<sup>nd</sup> Difference (Level 2). Apabila data sudah stasioner, maka pengujian diferensiasi dihentikan pada level tersebut.



**Gambar 2. Sampel Grafik Tingkat Kemiskinan di Indonesia**

**Tabel 2. Aturan Pola ACF dan PACF**

ACF	PACF	Model
<i>Tails off</i>	<i>Cut off</i> setelah lag $q$	AR ( $p$ )
<i>Cut off</i> setelah lag $p$	<i>Tails off</i>	MA ( $q$ )
<i>Cut off</i> setelah lag $p$	<i>Cut off</i> setelah lag $p$	AR( $p$ ), MA( $q$ )

Keterangan : *tails off* adalah kondisi dimana parameter model berubah menjadi nol secara asimptotik dan *cut off* adalah kondisi dimana parameter model menjadi nol secara tiba-tiba [12].



## Pendugaan Model dan Estimasi Parameter ACF-PACF

Data *time series* yang sudah stasioner dapat dilakukan pengujian estimasi parameter model menggunakan *Autocorelation Function* (ACF) dan *Partial Autocorelation Function* (PACF). Model pembentukan ARIMA harus memperhatikan aturan pola seperti pada Tabel 2. Setelah pendugaan model ARIMA ditemukan berdasarkan plot ACF dan PACF dari data *time series* yang sudah stasioner, maka selanjutnya dilakukan tahapan analisis regresi melalui metode *Least Square*. Pengujian menggunakan metode *Least Square* untuk menemukan nilai penduga dalam pemodelan regresi yang meminimumkan jumlah kuadrat galat. Pengujian ini untuk menghasilkan nilai *Akaike Info Criterion* (AIC) dan *Schwarz Info Criterion* (SBC).

## Pemeriksaan Diagnostik

Pemeriksaan Diagnostik dilakukan untuk mengetahui apakah dari pemodelan ARIMA sementara menghasilkan nilai yang signifikan atau tidak. Model dikatakan signifikan apabila probabilitas (*p-values*) < 0.05. Pengujian yang dilakukan pada tahap ini yaitu dengan uji *white noise*. Pengujian dengan *white noise* dikatakan baik dan bisa dijadikan pemodelan ARIMA apabila plot ACF dan PACF memiliki probabilitas (*p-value*) > 0.05.

## Evaluasi Parameter Error

Pengujian dilakukan untuk melihat model ARIMA yang terbaik dengan melihat

persentase *error* dari hasil peramalan. Parameter untuk menguji *error* menggunakan 3 kriteria yaitu *Root Mean Square Error* (RMSE), *Mean Square Error* (MSE) dan *Mean Absolute Percentage Error* (MAPE). Kriteria RMSE digunakan untuk memberikan gambaran lengkap tentang distribusi kesalahan menggunakan Persamaan 1.

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^T (Y_t - \hat{Y}_t)^2} \quad (1)$$

Kriteria MSE untuk mengukur tingkat kesalahan dari prediksi menggunakan Persamaan 2. Kriteria MAPE merupakan presentase kesalahan hasil peramalan terhadap nilai aktual pada periode tertentu menggunakan Persamaan 3.

$$MSE = \frac{1}{T} \sum_{t=1}^T (Y_t - \hat{Y}_t)^2 \quad (2)$$

$$MAPE = \frac{1}{T} \sum_{t=1}^T \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right| \quad (3)$$

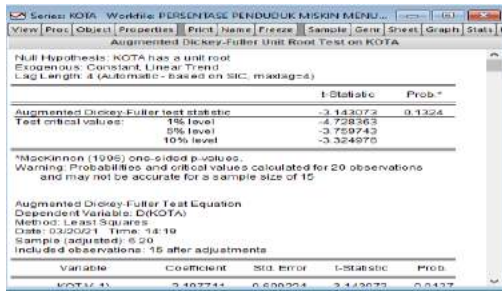
## HASIL DAN PEMBAHASAN

Dalam penelitian kali ini yang akan diramalkan adalah tingkat kemiskinan di Indonesia berdasarkan wilayah perkotaan, pedesaan dan nasional pada masa 4 tahun kedepan, dari tahun 2021 hingga 2024. Pengujian dengan model ARIMA diharapkan mampu meramalkan tingkat kemiskinan di Indonesia berdasarkan model terbaik dengan nilai RMSE, MAE dan MAPE.

**Hasil Identifikasi Model**

Proses pengidentifikasian model yang akan digunakan dalam peramalan, langkah yang dilakukan yaitu dengan melakukan identifikasi stasioneritas data yang diuji dan

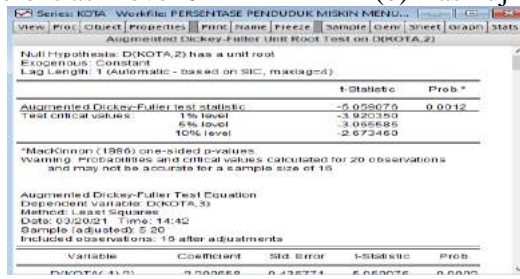
pengujian diferensiasi. Diperlukan uji *unit root* menggunakan *Augmented Dickey-Fuller* (ADF) untuk mengidentifikasi stasioneritas data dan pengujian diferensiasi.



(a) Hasil uji Diferensiasi Level 0



(b) Hasil uji Diferensiasi Level 1

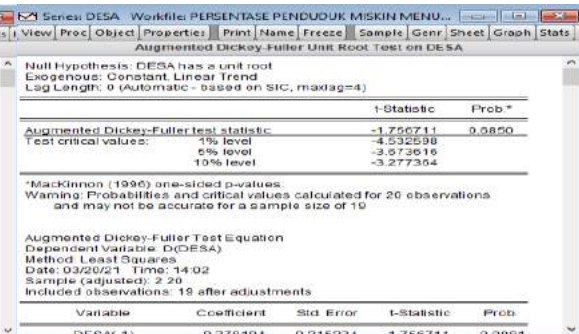


(c) Hasil uji Level 2

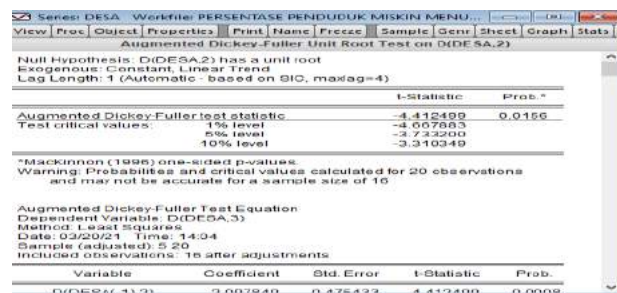
**Gambar 3. Hasil Uji data Perkotaan dengan ADF**



(a) Hasil uji Diferensiasi Level 0

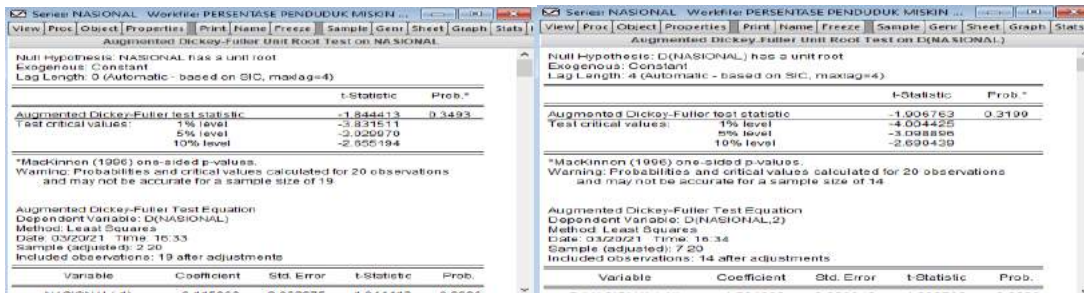


(b) Hasil uji Diferensiasi Level 1



(c) Hasil Uji Diferensiasi Level 2

**Gambar 4. Hasil Uji data Pedesaan dengan ADF**



(a) Hasil uji Diferensiasi Level 0

(b) Hasil uji Diferensiasi Level 1



(c) Hasil Uji Diferensiasi Level 2

**Gambar 5. Hasil Uji data Nasional dengan ADF**

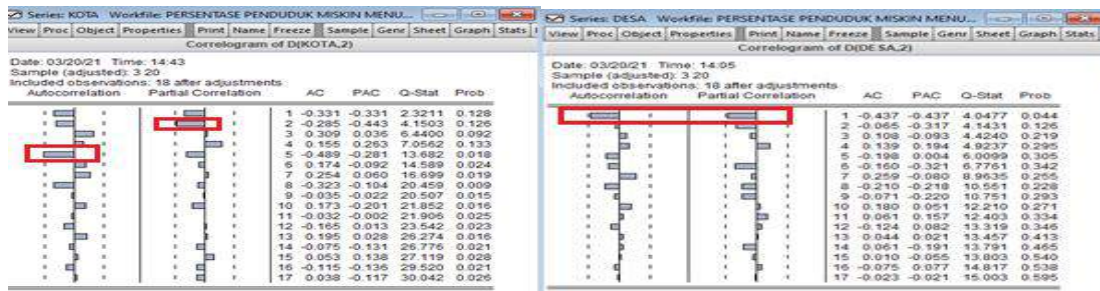
Pada Gambar 4(a) dapat dilihat bahwa pada pengujian awal (level 0) didapatkan nilai  $p$  value  $0.6850 > 0.05$  dan  $t$  statistic  $-1.756711 > -3.673616$  sehingga bisa dikatakan  $H_0$  ditolak dan data belum stasioner. Pada gambar 4(b) bahwa diferensiasi level 1 ( $d=1$ ) mendapatkan  $p$  value  $0.0586 > 0.05$  dan  $t$  statistic  $-3.600253 > -3.690814$  sehingga bisa dikatakan  $H_1$  ditolak dan data belum stasioner. Gambar 4(c) bahwa diferensiasi level 2 ( $d=2$ ) mendapatkan  $p$  value  $0.0156 < 0.05$  dan  $t$  statistic  $-4.412499 < -3.733200$  sehingga bisa dikatakan  $H_2$  dan data sudah stasioner. Hasil uji diferensiasi untuk data pedesaan adalah 2<sup>nd</sup> Difference (Level 2).

Pada pengujian Gambar 5(a) dapat dilihat bahwa pada pengujian awal (level 0) didapatkan nilai  $p$  value  $0.3493 > 0.05$  dan  $t$  statistic  $-1.844413 > -3.029970$  sehingga bisa dikatakan  $H_0$  dan data belum stasioner. Gambar 5(b) bahwa diferensiasi level 1

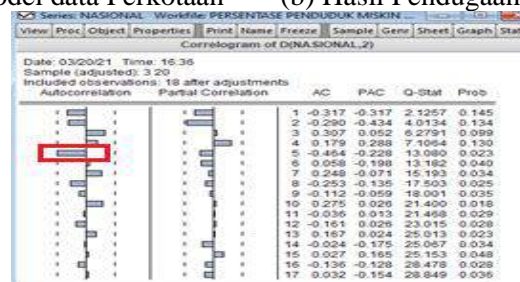
( $d=1$ ) mendapatkan  $p$  value  $0.3199 > 0.05$  dan  $t$  statistic  $-1.906763 > -3.098896$  sehingga bisa dikatakan  $H_1$  dan data belum stasioner. Pada gambar 5(c) bahwa diferensiasi level 2 ( $d=2$ ) mendapatkan  $p$  value  $0.0013 < 0.05$  dan  $t$  statistic  $-4.977644 < -3.065585$  sehingga bisa dikatakan  $H_2$  dan data sudah stasioner. Hasil uji diferensiasi untuk data nasional adalah 2<sup>nd</sup> Difference (Level 2).

### Hasil Estimasi Parameter Model

Setelah data stasioner maka selanjutnya pengujian dengan parameter model ACF dan PACF. Pada pengujian sebelumnya masing-masing data memiliki level diferensiasi 2. Selanjutnya untuk mendapatkan model ARIMA bisa dilihat pada Gambar 6. Gambar 6(a) menampilkan pendugaan model plot ACF dan plot PACF dengan data kemiskinan di Perkotaan diferensiasi level 2. Plot ACF mengalami signifikan pada lag ke 5 dan plot PACF mengalami signifikan di lag ke 2.



(a) Hasil Pendugaan Model data Perkotaan (b) Hasil Pendugaan Model data Pedesaan



(c) Hasil Pendugaan Model data Nasional  
Gambar 6. Hasil Estimasi Model Data Uji

Berdasarkan hasil tersebut pendugaan model ARIMA data kemiskinan wilayah Perkotaan adalah *autoregressive* dengan nilai AR(2) dan *moving average* dengan nilai MA(5). Tabel 3 terdapat 3 model pendugaan yang sesuai untuk data kemiskinan perkotaan yaitu ARIMA(2,2,5), ARIMA(0,2,5) dan ARIMA(2,2,0). Gambar 6 (b) menampilkan pendugaan model plot ACF dan plot PACF dengan data kemiskinan di Pedesaan diferensiasi level 2. Pada plot ACF mengalami signifikan pada lag ke 1 dan plot PACF juga

mengalami signifikan di lag ke 1. Berdasarkan hasil tersebut pendugaan model ARIMA data kemiskinan wilayah Perkotaan adalah *autoregressive* dengan nilai AR(1) dan *moving average* dengan nilai MA(1) dapat dilihat pada Tabel 4. Maka terdapat 3 model yang sesuai untuk data kemiskinan perkotaan yaitu ARIMA(1,2,1), ARIMA(0,2,1) dan ARIMA(1,2,0). Gambar 6(c) menampilkan pendugaan model plot ACF dan plot PACF dengan data kemiskinan secara Nasional diferensiasi level 2.

**Tabel 3. Pendugaan Variabel ACF dan PACF Perkotaan**

	variabel p (ACF)	variabel d (level diferensiasi)	Variabel q (PACF)
ARIMA (2,2,5)	2	2	5
ARIMA (0,2,5)	0	2	5
ARIMA (2,2,0)	2	2	0

**Tabel 4. Pendugaan Variabel ACF dan PACF Pedesaan**

	variabel p (ACF)	variabel d (level diferensiasi)	Variabel q (PACF)
ARIMA (1,2,1)	1	2	1
ARIMA (0,2,1)	0	2	1
ARIMA (1,2,0)	1	2	0

**Tabel 5. Pendugaan Variabel ACF dan PACF Nasional**

	variabel p (ACF)	variabel d (level diferensiasi)	Variabel q (PACF)
ARIMA (0,2,5)	0	2	5

**Tabel 6. Perbandingan Model ARIMA Data Perkotaan**

Model ARIMA	AIC	SBC
ARIMA (2,2,5)	0.646196	0.844056
ARIMA (0,2,5)	0.579118	0.727513
ARIMA (2,2,0)	0.779273	0.927669

Dependent Variable: D(D(KOTA))  
 Method: ARIMA Maximum Likelihood (OPG - BHHH)  
 Date: 03/26/21 Time: 10:22  
 Sample: 3 20  
 Included observations: 18  
 Convergence achieved after 21 iterations  
 Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.029397	0.047031	0.625070	0.5413
MA(5)	-0.723190	1.189205	-0.608129	0.5522
SIGMASQ	0.061888	0.047003	1.316697	0.2077
R-squared	0.419326	Mean dependent var		0.025556
Adjusted R-squared	0.341903	S.D. dependent var		0.335931
S.E. of regression	0.272518	Akaike info criterion		0.579118
Sum squared resid	1.113991	Schwarz criterion		0.727513
Log likelihood	-2.212050	Hannan-Quinn criter.		0.599590
F-statistic	5.416027	Durbin-Watson stat		2.587611
Prob(F-statistic)	0.016963			
Inverted MA Roots	.94	.29-.99i	.29+.99i	-.76+.55i
	-.76-.55i			

**Gambar 7. Pendugaan Parameter Model ARIMA (0,2,5) data Perkotaan**

**Tabel 7. Perbandingan Model ARIMA Data Pedesaan**

Model ARIMA	AIC	SBC
ARIMA (1,2,1)	0.689214	0.887075
ARIMA (0,2,1)	0.581924	0.730319
ARIMA (1,2,0)	0.819069	0.967465

Dependent Variable: D(DEGA)  
 Method: ARIMA Maximum Likelihood (OPG - BHHH)  
 Date: 03/26/21 Time: 11:02  
 Sample: 2 20  
 Included observations: 19  
 Convergence achieved after 6 iterations  
 Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.129445	0.093943	-1.542048	0.1426
MA(1)	0.120484	0.258315	0.466425	0.6472
SIGMASQ	0.069578	0.029275	2.342510	0.0324
R-squared	0.011373	Mean dependent var		-0.132632
Adjusted R-squared	-0.112206	S.D. dependent var		0.270593
S.E. of regression	0.265370	Akaike info criterion		0.474691
Sum squared resid	1.302979	Schwarz criterion		0.623773
Log likelihood	-1.509192	Hannan-Quinn criter.		0.499869
F-statistic	0.092029	Durbin-Watson stat		1.837323
Prob(F-statistic)	0.912558			
Inverted MA Roots		-.12		

**Gambar 8. Pendugaan Parameter Model ARIMA (0,2,1) Data Pedesaan**

Pada plot ACF mengalami signifikan pada lag ke 5 dan plot PACF tidak mengalami lag signifikan. Berdasarkan hasil tersebut pendugaan model ARIMA data kemiskinan secara Nasional adalah *moving average* dengan nilai MA(5) dapat dilihat pada Tabel 5. Maka terdapat 1 model yang sesuai untuk data kemiskinan perkotaan yaitu ARIMA (0,2,5). Setelah diperoleh pendugaan model ARIMA maka langkah selanjutnya menguji

model tersebut berdasarkan kriteria *Akaike Info Criterion* (AIC) dan *Schwarz Info Criterion* (SBC). Berdasarkan hasil pengujian terhadap masing-masing data maka diperoleh pendugaan model terbaik dapat dilihat pada Tabel 6.

Penentuan nilai estimasi parameter model terbaik dapat dilihat dari hasil AIC dan SBC terkecil. Dari perbandingan 3 model ARIMA yang terdapat pada Tabel 6, model

ARIMA(0,2,5) memiliki estimasi parameter model terbaik dibandingkan model ARIMA (2,2,5) dan ARIMA(2,2,0) dengan nilai AIC sebesar 0.579118 dan SBC sebesar 0.727513. Nilai ARIMA(0,2,5) dapat dilihat pada Gambar 7. Penentuan nilai estimasi parameter model terbaik dapat dilihat dari hasil AIC dan SBC terkecil. Dari perbandingan 3 model ARIMA yang terdapat pada Tabel 7, model ARIMA (0,2,1) memiliki estimasi parameter model terbaik dibandingkan model ARIMA (1,2,1) dan ARIMA(1,2,0) dengan nilai AIC sebesar 0.581924 dan SBC sebesar 0.730319.

Penentuan nilai estimasi parameter model untuk data secara Nasional hanya memiliki 1 model jadi tidak diperlukan adanya perbandingan. Hasil pengujian model ARIMA (0,2,5) data kemiskinan secara nasional memiliki nilai AIC sebesar 0.574757 dan nilai SBC sebesar 0.723153. Hasil dapat dilihat pada Tabel 8 dan Gambar 9.

**Tabel 8. Perbandingan Model ARIMA Data Nasional**

Model ARIMA	AIC	SBC
ARIMA (0,2,5)	0.574757	0.723153

Dependent Variable: D(D(NASIONAL))  
Method: ARMA Maximum Likelihood (OPG - BHHH)  
Date: 03/20/21 Time: 10:42  
Sample: 3 20  
Included observations: 18  
Convergence achieved after 12 iterations  
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std Error	t-Statistic	Prob.
C	0.025262	0.063292	0.399142	0.6954
MA(5)	-0.479577	0.588105	-0.815461	0.4276
SIGMASQ	0.069347	0.032932	2.105786	0.0525

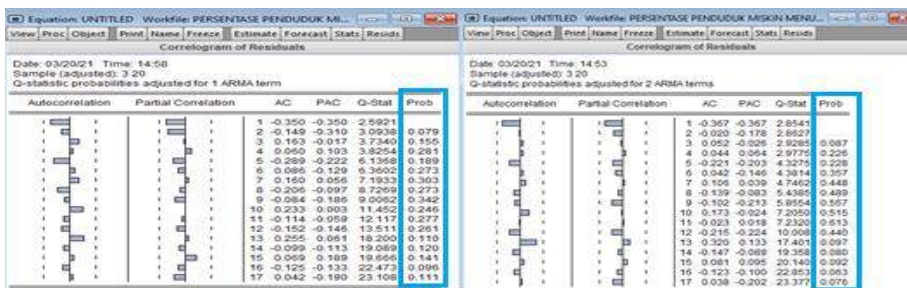
  

R-squared	0.257881	Mean dependent var	0.030000
Adjusted R-squared	0.158931	S.D. dependent var	0.314549
S.E. of regression	0.288472	Akaike info criterion	0.574757
Sum squared resid	1.248245	Schwarz criterion	0.723153
Log likelihood	-2.172817	Hannan-Quinn criter.	0.595210
F-statistic	2.606192	Durbin-Watson stat	2.639835
Prob(F-statistic)	0.106796		

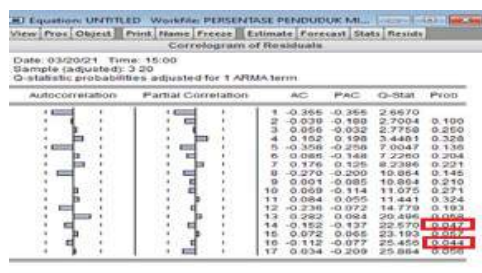
  

Inverted MA Roots				
86	.27-.82i	27+.82i	-70+.51i	
	-70-.51i			

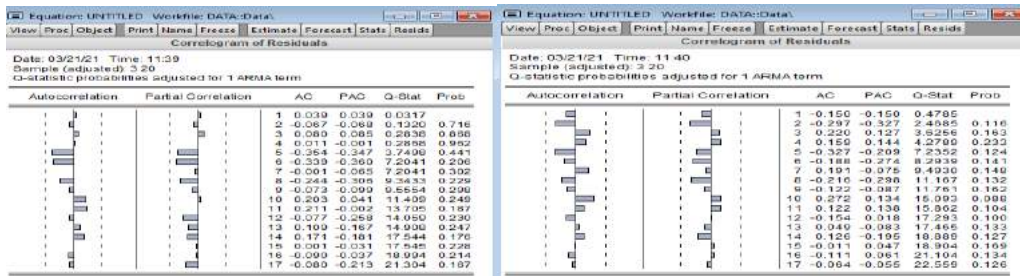
**Gambar 9. Pendugaan Parameter Model ARIMA (0,2,5) Data Nasional**



(a) Hasil Diagnostik ARIMA (2,2,5) (b) Hasil Diagnostik ARIMA (0,2,5)

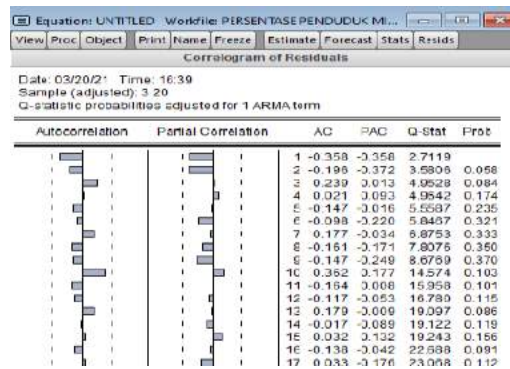
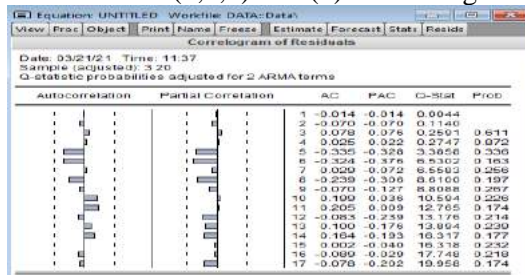


(c) Hasil Diagnostik ARIMA (2,2,0)  
**Gambar 10. Hasil Diagnostik Model ARIMA Perkotaan**



(a) Hasil Diagnostik ARIMA (1,2,1)      (b) Hasil Diagnostik ARIMA (0,2,1)

(c) Hasil Diagnostik ARIMA (1,2,0)  
**Gambar 11. Hasil Diagnostik Model ARIMA Pedesaan**



**Gambar 12. Hasil Diagnostik Model ARIMA (0,2,5) Nasional**

**Pemeriksaan Diagnostik**

Pada tahap ini untuk memeriksa apakah model yang digunakan sudah baik dengan melihat residual. Analisis residual dapat dikatakan baik jika memiliki *white noise* dengan melihat nilai probabilitas pada ACF dan PACF yang tidak signifikan ( $p\text{-value} > \alpha = 0.05$ ).

Berdasarkan Gambar 11. (a), (b) dan (c) *lag* ke-1 sampai dengan *lag* ke-17 tidak ada *lag* yang signifikan dengan ditunjukkan bahwa nilai probabilitas dari masing-masing pengujian lebih besar dari pada nilai signifikan 5% ( $p\text{-value} > 0.05$ ). Hal ini menunjukkan bahwa residual model ARIMA (1,2,1), ARIMA(0,2,1) dan ARIMA(1,2,0) sudah *white noise* dan terdistribusi dengan

baik sehingga 3 model tersebut bisa digunakan untuk proses peramalan. Berdasarkan Gambar 12 *lag* ke-1 sampai dengan *lag* ke-17 tidak ada *lag* yang signifikan ditunjukkan bahwa nilai probabilitas pada pengujian tersebut lebih besar dari nilai signifikan 5% ( $p\text{-value} > 0.05$ ). Hal ini menunjukkan bahwa residual model ARIMA (0,2,5) sudah *white noise* dan terdistribusi dengan baik sehingga model tersebut bisa digunakan untuk proses peramalan.

### Pemeriksaan Error

Pada tahap ini masing-masing model ARIMA yang terdapat pada data akan di evaluasi akurasi dan *error* untuk mendapatkan hasil peramalan terbaik. Untuk evaluasi *error* digunakan 3 parameter yaitu *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE) dan *Mean Absolute Percentage*

*Error* (MAPE). Dari Tabel 9. bahwa model terbaik untuk peramalan data kemiskinan perkotaan yaitu ARIMA (2,2,5) karena memiliki nilai *error* lebih kecil dari ARIMA (0,2,5). Nilai ARIMA(2,2,5) memiliki RMSE=1.246582, MAE=0.923255 dan MAPE=12%. Dari Tabel 10. bahwa model terbaik untuk peramalan data kemiskinan pedesaan yaitu ARIMA(1,2,1) karena memiliki nilai *error* lebih kecil dari ARIMA(0,2,1) dan ARIMA(1,2,0). Nilai ARIMA(1,2,1) memiliki RMSE=0.392650, MAE=0.311529 dan MAPE=2%, sementara untuk peramalan data kemiskinan secara nasional hanya menggunakan 1 model saja yaitu ARIMA (0,2,5) dan tidak ada perbandingan. Berdasarkan hasil pengujian untuk ARIMA(0,2,5) nilai akurasi RMSE sebesar 2.533166, MAE sebesar 2.090505 dan MAPE sebesar 20%.

**Tabel 9. Perbandingan Error Model ARIMA Perkotaan**

Model	RMSE	MAE	MAPE
ARIMA(2,2,5)	1.246582	0.923255	12.95332
ARIMA(0,2,5)	2.319173	1.831840	25.16329

**Tabel 10. Perbandingan Error Model ARIMA Pedesaan**

Model	RMSE	MAE	MAPE
ARIMA(1,2,1)	0.392650	0.311529	2.255216
ARIMA(0,2,1)	1.904272	1.671111	12.47963
ARIMA(1,2,0)	0.574083	0.469415	3.505666

**Tabel 11. Model ARIMA Terbaik**

Perkotaan	ARIMA (2,2,5)
Pedesaan	ARIMA (1,2,1)
Nasional	ARIMA (0,2,5)

**Tabel 12. Hasil Peramalan**

	Perkotaan	Pedesaan	Nasional
Maret 2021	9.731618	12.86529	14.68985
September 2021	10.00904	12.90742	15.06510



Maret 2022	10.30984	12.96710	15.46560
September 2022	10.63402	13.04433	15.89138
Maret 2022	10.98156	13.13910	16.34241
September 2023	11.35246	13.25141	16.81871
Maret 2024	11.74672	12.38128	17.32027
September 2024	12.16436	13.52868	17.84709

### Hasil Peramalan

Berdasarkan pengujian terhadap masing-masing data maka diperoleh model ARIMA terbaik dapat lihat pada Tabel 11. Setelah menemukan model ARIMA terbaik dari pengujian, tahap selanjutnya proses peramalan. Tabel 12 merupakan hasil peramalan dari pengujian data kemiskinan di Indonesia berdasarkan wilayah Perkotaan, Pedesaan dan secara Nasional. Pada Tabel 12. diperoleh hasil peramalan yang dilakukan terhadap data kemiskinan dengan pemodelan ARIMA. Peramalan dilakukan untuk periode Maret 2021 hingga September 2024. Berdasarkan pengujian yang telah dilakukan dalam penelitian ini dapat dikatakan bahwa hasil pengujian dengan model ARIMA terhadap data kemiskinan di Indonesia termasuk kategori pengujian yang baik. Hal ini dibuktikan terhadap nilai akurasi maupun perhitungan *error* dengan menggunakan parameter *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE) dan *Root Mean Square Error* (RMSE) menghasilkan nilai yang akurat dan persentase *error* rendah. Keunggulan dari penelitian ini dibanding penelitian terdahulu yaitu terdapat pada tahap pengujian akurasi dan nilai *error* dimana pada penelitian ini dilakukan pengujian dengan 3 parameter yaitu MAE untuk menguji tingkat keakuratan model [13], MAPE untuk menguji persentase

kesalahan dalam penelitian [14] dan RMSE untuk mengukur tingkat kesalahan dari perbandingan hasil prediksi yang dilakukan [15], sementara pada penelitian lain hanya menggunakan 1 parameter saja. Oleh karena itu penelitian yang dilakukan untuk meramalkan angka kemiskinan di Indonesia dengan model ARIMA sangat baik namun perlu juga dilakukan dengan metode lainnya untuk mengetahui nilai akurasi dan persentase yang berbeda.

### KESIMPULAN DAN SARAN

Disimpulkan bahwa peramalan dengan model ARIMA terhadap data kemiskinan wilayah perkotaan, model ARIMA yang digunakan adalah model ARIMA(2,2,5). Model ini memiliki nilai akurasi dan *error* lebih baik dari pada model ARIMA lainnya dengan nilai RMSE sebesar 1.246582, nilai MAE sebesar 0.923255 dan nilai MAPE sebesar 12%, untuk data kemiskinan wilayah pedesaan menggunakan model ARIMA(1,2,1) karena memiliki nilai akurasi dan *error* lebih baik dari pada model ARIMA lainnya dengan nilai RMSE sebesar 0.392650, nilai MAE sebesar 0.311529 dan nilai MAPE sebesar 2% dan data kemiskinan secara Nasional hanya diperoleh 1 model ARIMA yaitu model ARIMA(0,2,5) dengan nilai akurasi dan *error* berdasarkan parameter RMSE sebesar

2.533166, MAE sebesar 2.090505 dan MAPE sebesar 20%.

Beberapa saran yang dapat diberikan agar penelitian berikutnya menjadi lebih baik antara lain.

1. Penggunaan *dataset* disarankan lebih banyak daripada yang digunakan pada penelitian saat ini. Karena semakin banyak *dataset* yang digunakan, semakin besar pula nilai akurasi yang dihasilkan
2. Metode peramalan bisa menggunakan selain dari ARIMA, hal ini diharapkan dengan metode yang lain bisa menghasilkan tingkat akurasi yang lebih baik dalam peramalan tingkat kemiskinan di Indonesia
3. Metode untuk menguji parameter kesalahan juga bisa menggunakan selain dari 3 metode yang digunakan pada penelitian saat ini, harapannya yaitu bisa menghasilkan nilai *error* yang lebih baik dari metode yang digunakan sebelumnya.

## DAFTAR PUSTAKA

- [1] B. P. Statistik, Data dan Informasi Kemiskinan Kabupaten dan Kota Tahun 2019, Jakarta: CV Nario Sari, 2020.
- [2] B. P. Statistik, "bps.go.id," Persentase Penduduk Miskin Menurut Wilayah (Persen), 1996-2020, 2021. [Online]. Available: [https://www.bps.go.id/indicator/23/184/1/persentase-penduduk-miskin-](https://www.bps.go.id/indicator/23/184/1/persentase-penduduk-miskin-menurut-wilayah.html)
- [3] Heizer, B. and Render, B., Operations Management, Jakarta: Salemba Empat, 2011.
- [4] M. Ekananda, Analisis Data Time Series Untuk Penelitian Ekonomi, Manajemen dan Akuntansi, Bogor: Mitra Wacana, 2014.
- [5] N. Bakar, and S. Rosbi, "Autoregressive Integrated Moving Average (ARIMA) Model for Forecasting Cryptocurrency Exchange Rate in High Volatility Environment: A New Insight of Bitcoin Transaction," International Journal of Advanced Engineering Research and Science (IJAERS), vol. 4, no. 1, 2017.
- [6] P. Herman, B. Yuniarta, and H. Rahmawati, "Prediksi Jumlah Penduduk Miskin Kalimantan Timur Menggunakan Single dan Double Exponential Smoothing," Jurnal Ilmiah Ilmu Komputer, vol. 15, no. 1, pp. 47-51, 2020.
- [7] S. Desi, "Forecasting of Poverty Data Using Seasonal ARIMA Modeling in West Java Province," JTAM (Jurnal Teori dan Aplikasi Matematika), vol. 4, no. 1, pp. 76-86, 2020.
- [8] A. Aulia, S. Eka, J. Miftahul, L.K. Pathullaili, and S. Ahmad, "ARDL Method: Forecasting Data Kemiskinan di NTB," JTAM (Jurnal Teori dan

- Aplikasi Matematika), vol. 3, no. 1, pp. 52-57, 2019.
- [9] P. Jana, "Aplikasi Triple Exponential Smoothing untuk Forecasting Jumlah Penduduk Miskin," *Jurnal Derivat*, vol. 3, no. 2, pp. 75-81, 2016.
- [10] B. P. Statistik, "bps.go.id," ARIMA, 2021. [Online]. Available: [https://daps.bps.go.id/file\\_artikel/77/arima.pdf](https://daps.bps.go.id/file_artikel/77/arima.pdf). [Accessed 18 Maret 2021].
- [11] F. Hutapea, "Analisis Strategi Diferensiasi Untuk Meningkatkan Keunggulan Bersaing Pada Produk Flexy Classy di PT.Telkom Representative Office Medan Area Commerce I Sumatera," Universitas Sumatera Utara, Medan, 2012.
- [12] Q. Stack, "qastack.id," Istilah cut-off dan tail off tentang ACF, fungsi PACF, 2021. [Online]. Available: <https://qastack.id/stats/241914/terms-cut-off-and-tail-off-about-acf-pacf-functions>. [Accessed 21 May 2021].
- [13] A. Suryanto, and A. Muqtadir, "Penerapan Metode Mean Absolute Error (MAE) Dalam Algoritma Regresi Linear Untuk Prediksi Produksi Padi," *SAINTEKBU*, vol. 11, pp. 78-83, 2019.
- [14] Khoiri, "Khoiri.com," Cara Menghitung Mean Absolute Percentage Error, 2020. [Online]. Available: <https://www.khoiri.com/2020/12/pengertian-dan-cara-menghitung-mean-absolute-percentage-error-mape.html>. [Accessed 21 May 2021].
- [15] S.W. Hasniah, and Y. Desi, "Penerapan Metode ARIMA Ensemble pada Peramalan (Studi Kasus : Inflasi di Indonesia)," *Jurnal Eksponensial*, vol. 7, no. 1, 2016.

# **PENGELOMPOKAN WILAYAH PENYEBARAN COVID-19 DI KABUPATEN KARAWANG MENGGUNAKAN ALGORITMA K-MEANS**

*Zulia Imami Alfianti  
Universitas Bina Sarana Informatika Kampus Kabupaten Karawang  
Jl. Banten No. 1, Karangpawitan, Karawang 41371, Jawa Barat  
zulia.zim@bsi.ac.id*

## **Abstrak**

*Virus Covid-19 merupakan keluarga besar coronavirus yang dapat menyerang hewan. Covid-19 merupakan penyakit menular yang ditandai oleh gejala pada bagian pernapasan akut (Coronavirus 2 (Severe Acute respiratory Syndrome Coronavirus 2) atau SARS-CoV-2). Virus ini menyerang penduduk di Wuhan China sejak Desember 2019. Penularan virus ini terjadi jika adanya kontak langsung antar manusia. Sejak Februari 2020 penyebaran virus covid-19 mulai terjadi di berbagai wilayah di Indonesia dan semakin meningkat setiap harinya. Pada penelitian ini akan dilakukan pengklasteran wilayah penyebaran Covid-19 di Kabupaten Karawang. Pengklasteran dilakukan bertujuan untuk mengelompokkan himpunan data yang besar untuk memudahkan analisis data atau pengolahan data lebih lanjut menggunakan metode k-means dengan membagi data kecamatan di kabupaten Karawang menjadi tiga cluster yaitu cluster wilayah penyebaran rendah, cluster wilayah penyebaran sedang, dan wilayah penyebaran tinggi. Hasil dari penelitian ini yaitu terdapat 50% persen wilayah yang tingkat penyebarannya rendah, 33,3% persen wilayah yang tingkat penyebarannya sedang, dan 16,7% persen wilayah yang tingkat penyebarannya tinggi.*

**Kata Kunci:** *Clustering, Covid-19, Data Mining, K-Means*

## **Abstract**

*The Covid-19 virus is a big family of coronaviruses that can infect animals. Covid-19 is an infectious disease characterized by symptoms in the acute respiratory tract (Coronavirus 2 (Severe Acute Respiratory Syndrome Coronavirus 2) or SARS-CoV-2). This virus has attacked residents in Wuhan, China since December 2019 The transmission of this virus occurs if there is direct contact between humans. Since February 2020 the spread of the Covid-19 virus has begun to occur in various regions in Indonesia and is increasing every day. In this study, clustering of the Covid-19 distribution area in Karawang Regency will be carried out. Clustering is carried out with the aim of grouping big data sets to facilitate data analysis or further data processing using the k-means method by dividing sub-district data in Karawang district into three clusters, namely clusters of low distribution areas, clusters of distribution areas, medium, and high distribution areas. The results of this study are that there are 50% percent of areas with low distribution, 33.3% of areas with moderate distribution, and 16.7% percent of areas with high distribution.*

**Keywords:** *Clustering, Covid-19, Data Mining, K-Means*

## PENDAHULUAN

Virus Covid-19 merupakan penyakit menular yang ditandai oleh gejala pada bagian pernapasan akut (*Coronavirus 2 (Severe Acute Respiratory Syndrome Coronavirus 2)*) atau SARS-CoV-2) [1]. Infeksi Virus Corona atau Covid-19 disebabkan oleh Corona virus, yaitu kelompok virus yang menginfeksi sistem pernapasan, pada sebagian besar kasus corona virus hanya menyebabkan infeksi pernapasan ringan sampai sedang, seperti flu, akan tetapi, virus ini juga bisa menyebabkan infeksi pernapasan berat, seperti Pneumonia, *MiddleEast Respiratory Syndrome* (MERS) dan *Severe Acute Respiratory Syndrome* (SARS) [2]. Virus ini dapat menyerang hewan dan juga manusia dan pada manusia gejalanya berupa infeksi yang serupa dengan penyakit SARS dan MERS, hanya saja Covid-19 bersifat lebih masif perkembangannya. Saat ini, penyebaran SARS-CoV-2 dari manusia ke manusia menjadi sumber transmisi utama sehingga penyebaran menjadi lebih agresif. Transmisi SARS-CoV-2 dari pasien simptomatik terjadi melalui droplet yang keluar saat batuk atau bersin [3]. Indonesia juga merupakan salah satu negara yang terdampak wabah yang satu ini [4]. Keberadaan virus Covid-19 ini tentunya melumpuhkan semua kegiatan baik kegiatan sosial dan maupun kegiatan perekonomian [5]. Penyebaran virus Covid-19 di seluruh

provinsi di Indonesia menyebabkan berbagai dampak pada semua bidang yang tentunya menghambat aktivitas manusia. Beberapa wilayah ditetapkan sebagai zona merah yang artinya wilayah yang memiliki tingkat penderita dan penyebarannya sangat tinggi. Sempat beberapa kali dikategorikan sebagai wilayah zona merah yang artinya wilayah yang tingkat penderitanya tinggi, membuat wilayah Kabupaten Karawang harus mendapat penanganan lebih cepat oleh pemerintah daerah Kabupaten Karawang [6]

Penelitian ini akan mengelompokkan wilayah penyebaran virus Covid-19 di Kabupaten Karawang. *Data mining* yang biasa disebut sebagai *Knowledge Discovery in Database* (KDD) merupakan kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola hubungan dalam himpunan data yang berukuran besar [7]. Hasil perhitungan dari *data mining* ini dapat digunakan untuk pengambilan keputusan. Teknik dalam *data mining* yang digunakan untuk melakukan pengelompokkan sejumlah data atau objek ke dalam *cluster* (grup) disebut dengan *clustering*. *Clustering* ini memungkinkan melakukan pengelompokkan data wilayah penyebaran Covid-19 menjadi menjadi beberapa *cluster* (grup).

Penelitian serupa mengenai metode *clustering* telah dilakukan oleh beberapa peneliti diantaranya Nayuni Diwitri dkk. [8] mengenai penerapan algoritma *K-Means*

dalam menentukan tingkat penyebaran Covid-19 di Indonesia. Hasil penelitian tersebut yaitu dari 34 provinsi di Indonesia terbagi menjadi pusat *cluster* yang diperoleh yaitu untuk *cluster* pertama atau penyebaran tingkat tinggi berada pada Provinsi DKI Jakarta, pusat *cluster* kedua atau penyebaran tingkat sedang berada pada provinsi Jawa Barat, dan pusat klaster ketiga atau penyebaran tingkat sedang berada pada Provinsi Banten. Penelitian selanjutnya oleh M. Hasnil Aditya dkk. [9] mengenai Penggunaan algoritma *K-Means* untuk *clustering* obat-obatan di RSUD Pekanbaru. Dari hasil klasterisasi pada obat-obatan tersebut kelompok obat yang termasuk pemakaian sedikit rata-rata permintaan obat setiap tahunnya kurang dari 18000 buah, dan obat yang termasuk pemakaian sedang rata-rata permintaan obat setiap tahunnya diantara 18000–70000 buah, sedangkan obat yang masuk kedalam kelompok obat yang pemakaian tinggi rata-rata permintaan obat setiap tahunnya diatas 70000 buah. Penelitian yang ketiga oleh Muhammad Rafi Muttaqin dan Meriska Defriani [10] mengenai penggunaan algoritma *K-Means* untuk pengelompokan topik skripsi mahasiswa hasil pengelompokan dengan nilai *cluster* yang paling tinggi dapat menunjukkan kemampuan mahasiswa pada tiap kelompok bidang keahlian. Hasil pengelompokan dianalisis dengan melihat jumlah matakuliah pada setiap kelompok bidang keahlian. Jumlah matakuliah yang paling banyak pada suatu kelompok bidang keahlian menandakan

bahwa mahasiswa memiliki kemampuan lebih baik pada bidang tersebut sehingga direkomendasikan topik skripsi yang sesuai dengan kelompok bidang keahlian tersebut.

Penelitian ini bertujuan untuk menentukan pengelompokan wilayah berdasarkan tingkat penyebaran Covid-19 di wilayah Kabupaten Karawang menggunakan metode *clustering* algoritma *K-Means*.

## METODE PENELITIAN

Penelitian dilakukan dengan menggunakan metode klasifikasi *K-Means*. Metode klasifikasi *K-Means* merupakan algoritma yang melakukan pengelompokan data berdasarkan data yang terdekat dengan titik pusat *cluster*.

Data yang digunakan dalam penelitian ini adalah data penyebaran virus Covid-19 di kabupaten Karawang yang diperoleh dari laman covid19.karawangkab.go.id. data yang digunakan adalah data penyebaran yang direkap pada tanggal 20 Mei 2021.

*Data mining* disebut juga *Knowledge Discovery in Database* (KDD) melakukan ekstraksi informasi dari tumpukan data. Proses pencarian informasi tersebut akan menemukan pola kecenderungan dari data yang kemudian hasil dari penambangan (*mining*) dapat menjadi informasi yang mudah dipahami.

Analisis *cluster* merupakan salah satu teknik *data mining* yang bertujuan untuk mengidentifikasi sekelompok objek yang

mempunyai kemiripan karakteristik tertentu yang dapat dipisahkan dengan kelompok objek lainnya, sehingga objek yang berada dalam kelompok yang sama relatif lebih homogen daripada objek yang berada pada kelompok yang berbeda [11].

*K-Means* yaitu salah satu dari metode pengelompokan data nonhierarki (sekatan) yang dapat mempartisi data kedalam bentuk dua kelompok ataupun lebih [9]. *K-Means clustering* merupakan sebuah konstanta dari sejumlah *cluster* yang diinginkan, sedangkan *Means* atau dapat didefinisikan sebagai *cluster* adalah suatu nilai rata-rata dari sekumpulan populasi data[12]. Metode *K-Means* digunakan untuk mengelompokkan data ke dalam beberapa kelompok, data dalam satu kelompok mempunyai karakteristik yang sama dan mempunyai karakteristik yang berbeda dengan data dari kelompok yang lain [13]. Metode ini berusaha meminimalkan variasi antar data yang ada di dalam suatu *cluster* dan memaksimalkan variasi data yang ada di *cluster* lainnya.

Langkah-langkah *clustering* menggunakan algoritma *K-Means* [14]:

1. Tentukan Jumlah *cluster*
2. Alokasikan data ke dalam *cluster* secara acak
3. Menghitung centroid/rata-rata dari data yang ada di masing-masing *cluster*
4. Alokasikan masing-masing data ke *centroid/cluster* terdekat
5. Jika masih ada data yang berpindah *cluster* adanya selisih antara nilai *centroid* lama

dengan *centroid* baru maka kembali lakukan perhitungan *centroid* kembali di masing-masing *cluster*. Jika tidak ada perubahan terhadap nilai *centroid* maka perhitungan dihentikan.

Proses mengolah data yang telah didapatkan dengan menyusun pola serta mengelompokkan pola tersebut menjadi beberapa *cluster* dengan metode *K-means*. Pengelompokkan pola menjadi beberapa *cluster* menggunakan rumus *euclidian distance*. Dengan menghitung jarak antara data dengan *centroid* sama dengan akar dari hasil penjumlahan antara data pada atribut ke-*n* dikurangi objek data pada *centroid* ke-*k*. Hasil jarak ini yang akan digunakan untuk mencari data mana yang memiliki tingkat kemiripan berdasarkan jarak yang paling kecil/terpendek. Perhitungan menggunakan rumus *euclidian distance* diformulasikan dengan menggunakan persamaan 1.

$$d(a, b) = \sqrt{\sum_{k=1}^n (a_k - b_k)^2} \quad (1)$$

Dimana:

d = jarak

n = jumlah dimensi (atribut)

$a_k$  dan  $b_k$  = atribut ke- *k* dari objek data

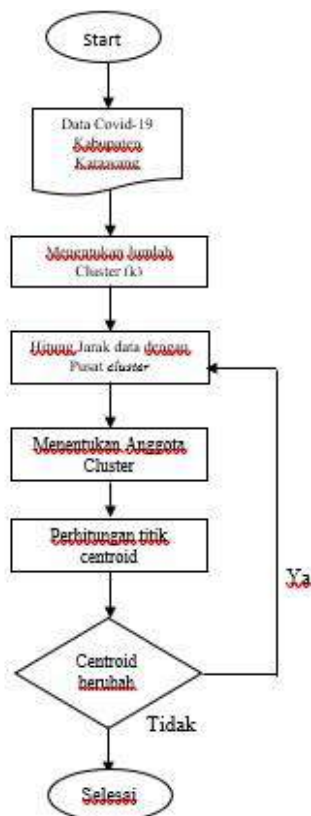
Untuk melengkapi pengetahuan dasar dalam melakukan penelitian ini digunakan studi literatur yang bersumber dari jurnal maupun penelitian terkait.

## HASIL DAN PEMBAHASAN

Proses pengelompokan daerah terdampak Covid-19 di Kabupaten Karawang menggunakan algoritma *K-Means*. Langkah pertama yang dilakukan adalah menentukan jumlah *cluster* yang ditentukan yaitu jumlah *cluster* harus lebih sedikit dari jumlah data yang akan diteliti. Selanjutnya, pilih titik *centroid* secara acak yang nantinya titik ini akan menjadi *centroid* dari masing-masing *cluster*. Kemudian hitung jarak dan alokasikan masing-masing data ke *centroid* terdekat. Setelah itu, tentukan *centroid* baru dari data yang ada di masing-masing *cluster*. Apabila masih ada data yang berpindah *cluster* atau ada perubahan nilai *centroid*, maka kembali hitung jarak dan alokasikan kembali data ke masing-masing *centroid*. Jika tidak ada perpindahan *cluster* atau perubahan

nilai *centroid* maka hentikan proses *clustering*. Langkah-langkah dalam metode *K-Means* dari data yang akan diteliti digambarkan dalam *flowchart* yang dapat dilihat pada Gambar 1.

Langkah pertama yang dilakukan pada proses *clustering* menggunakan metode *K-Means* adalah menentukan jumlah *cluster* awal secara bebas yaitu 3 *cluster* dan *centroid* dari data 30 Kecamatan di Kabupaten Karawang yang ditandai dengan K1 sampai dengan K30. Ditentukan 3 titik *centroid* awal yaitu K17, K18, K27 diambil dari jumlah kasus tinggi, sedang/menengah, dan rendah. Data kecamatan terdampak Covid-19 di Kabupaten Karawang dapat dilihat pada Tabel 1.



**Gambar 1. Langkah-Langkah Pengklusteran Menggunakan Metode *K-Means***



**Tabel 1. Data Kecamatan Terdampak Covid-19 Kabupaten Karawang pada Tanggal 20 Mei 2021**

Kode	Kecamatan	Perawatan	Isolasi	Sembuh	Meninggal
K1	Banyusari	1	0	165	7
K2	Batujaya	1	0	112	8
K3	Ciampel	2	0	307	6
K4	Cibuaya	1	0	154	9
K5	Cikampek	13	0	1016	41
K6	Cilamaya Kulon	0	0	137	7
K7	Cilamaya Wetan	3	0	380	18
K8	Cilebar	1	0	112	2
K9	Jatisari	3	0	403	18
K10	Jayakarta	0	0	178	13
K11	Karawang Barat	12	4	2378	62
K12	Karawang Timur	8	0	2100	54
K13	Klari	8	7	1910	68
K14	Kota Baru	6	0	1390	37
K15	Kutawaluya	1	0	144	10
K16	Lemahaban g	1	0	217	23
K17	Majalaya	3	0	990	20
K18	Pakisjaya	0	0	19	6
K19	Pangkalan	0	0	82	2
K20	Pedes	0	0	172	14
K21	Purwasari	6	0	750	21
K22	Rawamerta	1	0	314	18
K23	Rengasdeng klok	2	4	462	28
K24	Tegalwaru	0	0	134	3
K25	Telagasari	1	0	415	32
K26	Telukjambe Barat	5	5	345	9
K27	Telukjambe Timur	8	1	3246	73
K28	Tempuran	2	0	169	9
K29	Tirtajaya	0	0	147	6
K30	Tirtamulya	0	0	102	4

Sumber : covid19.karawangkab.go.id

Dari data penyebaran Covid-19 30 Kecamatan di Kabupaten Karawang tersebut ditentukan tiga *cluster* sebagai titik *centroid* awal yaitu data ke-27, 17, dan 18 dapat dilihat pada Tabel 1. dipilih berdasarkan jumlah kasus keseluruhan dengan menjumlahkan atribut dari masing-masing kecamatan

sehingga didapatkan satu kecamatan yang memiliki kasus tertinggi, satu kecamatan yang memiliki kasus sedang/menengah, dan satu kecamatan yang memiliki kasus terendah dari 30 kecamatan di Kabupaten Karawang yang ditetapkan sebagai *centroid* awal. *Centroid* awal yang telah ditentukan dapat dilihat pada

Tabel 2. Setelah dilakukan penentuan *centroid* awal, maka langkah selanjutnya adalah menghitung jarak antara *centroid* awal dengan titik *centroid* menggunakan rumus sebagai berikut. Berdasarkan persamaan (1). Hasil yang didapatkan untuk menentukan jarak antara *centroid* awal dengan titik *centroid* menggunakan rumus *euclidian*

*distance* yang hasilnya terdapat pada Tabel 3 ini, kemudian dipilih *centroid* mana dari masing-masing kode yang memiliki jarak terpendek.

$$d_{(a,b)} = \sqrt{(1-5)^2 + (0-5)^2 + (165-345)^2 + (7-9)^2}$$

$$d_{(a,b)} = \sqrt{(1-5)^2 + (0-5)^2 + (165-345)^2 + (7-9)^2}$$

$$d_{(a,b)} = \sqrt{(1-0)^2 + (0-0)^2 + (165-19)^2 + (7-6)^2}$$

**Tabel 2. Hasil Pemilihan Centorid Awal**

Centorid Awal	Atribut 1	Atribut 2	Atribut 3	Atribut 4
C1	5	5	345	9
C2	3	0	990	20
C3	0	0	19	6

**Tabel 3. Hasil Perhitungan Jarak Cluster dengan Titik Centroid**

Kode	C1	C2	C3	Kode	C1	C2	C3
K1	180	825	146	K16	128.9	773	198.7
K2	233	878	93	K17	645	0	971
K3	38.5	683	288	K18	326	971	0
K4	191	836	135	K19	263	908	63
K5	671.8	34.9	997.7	K20	173	818	153
K6	208	853	118	K21	405	240	731
K7	36.5	610	361	K22	32.9	676	295
K8	233	878	93	K23	118.5	528	443.5
K9	58.9	587	384	K24	211	856	115
K10	167	812	159	K25	73.9	575	396.8
K11	2033.7	1388.6	2359.6	K26	0	645	326
K12	1755.5	1110.5	2081.5	K27	2901.7	2256.6	3227.7
K13	1566	921	1892	K28	176	821	150
K14	1045	400	1371	K29	198	843	128
K15	201	846	125	K30	243	888	83

**Tabel 4. Keanggotaan Cluster Hasil Perhitungan Centroid pada Iterasi 1**

Centorid	Keanggotaan Cluster													
C1	K3	K7	K9	K16	K22	K23	K25	K26						
C2	K5	K11	K12	K13	K14	K17	K21	K27						
C3	K1	K2	K4	K6	K8	K10	K15	K18	K19	K20	K24	K28	K29	K30

**Tabel 5. Nilai Centroid dari Hasil Perhitungan Iterasi 1**

Centorid Awal	Atribut 1	Atribut 2	Atribut 3	Atribut 4
C1	2	1.125	355.375	19
C2	8	1.75	1722.5	47
C3	0.5	0	131	7.1

Dari hasil perhitungan untuk mencari jarak terpendek pada Tabel 3 didapatkan jarak terpendek antara data dengan *centroid* sehingga didapatkan keanggotaan *cluster* yaitu C1 sebanyak 8, C2 sebanyak 8, dan C3 sebanyak 14, dapat dilihat pada Tabel 4. Langkah selanjutnya yaitu dengan kembali menghitung nilai rata-rata anggota *cluster* sehingga didapatkan nilai *centroid* iterasi 1 yang dapat dilihat pada Tabel 5: Hasil yang didapatkan di iterasi 1 ternyata ada perubahan nilai *centroid*, maka diharuskan melakukan perhitungan kembali menggunakan rumus *eucledian distance* untuk menentukan jarak antara *centroid* dengan titik *centroid* yang

hasilnya terdapat pada Tabel 6, kemudian dipilih kembali *centroid* mana dari masing-masing kode yang memiliki jarak terpendek, hasil perhitungan tersebut dapat dilihat pada Tabel 6.

Berdasarkan hasil perhitungan nilai *centroid* pada iterasi 2 terdapat perbedaan antara nilai *centroid* yang dihasilkan pada iterasi 1. Sehingga diperlukan iterasi berikutnya dengan melakukan perhitungan kembali jarak antara data dengan nilai *centroid* yang terakhir untuk kemudian ditentukan kembali keanggotaan *cluster* melalui jarak terkecil antara data dengan *cluster* serta hitung nilai *centroid* berikutnya.

**Tabel 6. Hasil Perhitungan Jarak Cluster dengan Titik Centroid Sebelumnya Untuk Menentukan Centroid pada Iterasi 2**

Kode	C1	C2	C3	Kode	C1	C2	C3
K1	190.7	1558	34.5	K16	138.4	1505.7	87.9
K2	243	1610.9	18.5	K17	644.6	733	859.5
K3	50	1416	176.5	K18	336.6	1704	111.5
K4	201.6	1568.9	23.5	K19	273.9	1641	48.7
K5	661	706.5	886	K20	183.4	1550.8	42
K6	218.7	1586	6.5	K21	394.6	972.8	619.6
K7	24.6	1342.8	249.7	K22	41	1408.8	183.8
K8	243.9	1611	19	K23	107	1260.6	332
K9	47.6	1342.8	272.7	K24	221	1589	5
K10	177	1544.8	47.8	K25	61	1307.6	285.5
K11	2023	655.6	2248	K26	15	1378	214.6
K12	1744.9	377.5	1970	K27	2891	1523.7	3116
K13	1555	188.7	1780	K28	186.6	1553.9	38.5
K14	1034.7	332.6	1259.8	K29	208.7	1576	16.5
K15	211.5	1578.9	13.8	K30	253.8	1621	28.6

**Tabel 7. Nilai Centroid yang Dihasilkan pada Iterasi 2**

Centorid Awal	Atribut 1	Atribut 2	Atribut 3	Atribut 4
C1	4	1.1	538.2	21.4
C2	8.4	2.4	2204.8	58.8
C3	0.53	0	8	8.2

Berikut ini nilai *centroid* dari iterasi 2 yang dapat dilihat pada Tabel 7. Hasil yang didapatkan untuk menentukan jarak antara *centroid* dengan titik *centroid* menggunakan rumus *euclidian distance* pada iterasi 3 yang hasilnya dijabarkan pada Tabel 8, kemudian dipilih *centroid* mana dari masing-masing kode yang memiliki jarak terpendek, hasil perhitungan tersebut yang dapat dilihat pada Tabel 8. Setelah dilakukan perhitungan jarak data dengan titik *centroid*, didapatkan hasil perhitungan *centroid* pada iterasi 3. Berikut ini nilai *centroid* dari iterasi 3 yang dapat

dilihat pada Tabel 9. Setelah dilakukan perhitungan, proses iterasi terhenti di iterasi 3 karena nilai *centroid* iterasi 3 sama dengan nilai *centroid* iterasi 2. Perbandingan nilai *centroid* yang berasal dari iterasi awal sampai dengan iterasi 3 tersebut dapat dilihat pada Tabel 10.

Setelah dilakukan perbandingan nilai *centroid* pada Tabel 10, proses iterasi berhenti pada iterasi 3 karena nilai *centroid* pada iterasi 2 sama dengan nilai *centroid* pada iterasi 3. Sehingga didapatkan keanggotaan *cluster* seperti ditampilkan pada Tabel 11.

**Tabel 8. Hasil Perhitungan Jarak Cluster dengan Titik Centroid Sebelumnya Untuk Menentukan Centroid pada Iterasi 3**

Kode	C1	C2	C3	Kode	C1	C2	C3
K1	373	2040	156.8	K16	321	1988	209
K2	426	2093	103.8	K17	451.8	1215	981.8
K3	231.7	1898.5	298.8	K18	519	2186	11
K4	384	2051	145.8	K19	456.6	2123	74
K5	478	1188.9	1008	K20	366	2033	163.9
K6	401	2068	128.8	K21	211.8	1455	741.9
K7	158.2	1825	371.9	K22	224	1891	305.9
K8	426.6	2093.5	103.9	K23	76.5	1743	454
K9	135	1802	394.9	K24	404.6	2071.5	125.9
K10	360	2027	169.8	K25	123.6	1790	407
K11	1840	173	2370	K26	193.6	1860	336.8
K12	1562	104.9	2092	K27	2708	1041	3238
K13	1372.6	294.9	1902.7	K28	369	2036	160.8
K14	851.9	851	1382	K29	391.5	2058	138.8
K15	394	2061	135.8	K30	436.5	2103.5	93.8

**Tabel 9. Nilai Centroid yang Dihasilkan pada Iterasi 3**

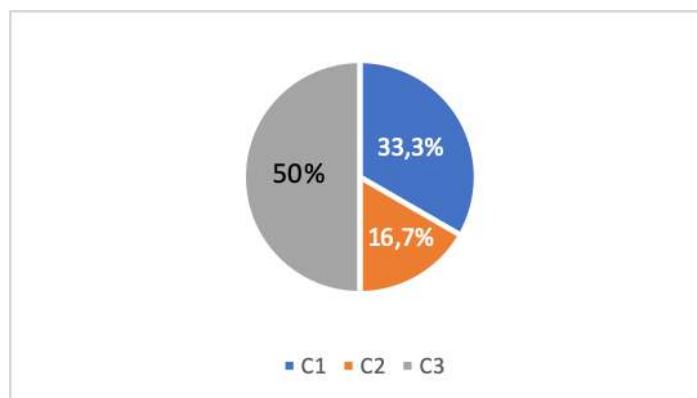
Centorid Awal	Atribut 1	Atribut 2	Atribut 3	Atribut 4
C1	4	1.1	538.2	21.4
C2	8.4	2.4	2204.8	58.8
C3	0.53	0	8	8.2

**Tabel 10. Perbandingan Nilai *Centroid* pada Iterasi Awal sampai Iterasi 3**

Centorid	Keanggotaan Cluster			
	Awal	Atribut 1	Atribut 2	Atribut 3
Iterasi 0				
C1	5	5	345	9
C2	5	0	990	20
C3	0	0	19	6
Iterasi 1				
C1	2	1.125	355.375	19
C2	8	1.75	1722.5	47
C3	0.5	0	131	71
Iterasi 2				
C1	4	1.1	538.2	21.4
C2	8.4	2.4	2204.8	58.8
C3	0.53	0	8	8.2
Iterasi 3				
C1	4	1.1	538.2	21.4
C2	8.4	2.4	2204.8	58.8
C3	0.53	0	8	8.2

**Tabel 11. Keanggotaan *Cluster* Hasil Perhitungan *Centroid* pada Iterasi 3**

Centorid	Keanggotaan Cluster														
C1	K3	K5	K7	K9	K17	K21	K22	K23	K25	K26					
C2	K11	K12	K13	K14	K27										
C3	K1	K2	K4	K6	K8	K10	K15	K16	K18	K19	K20	K24	K28	K29	K30



**Gambar 2. Diagram Hasil Pengklasteran Wilayah Terdampak Covid-19 Kabupaten Karawang**

Dari hasil *clustering* wilayah penyebaran Covid-19 diatas, maka untuk lebih jelasnya penulis gambarkan pada diagram lingkaran yang ditampilkan pada Gambar 2. Hasil perhitungan menggunakan metode *K-Means* pada diagram lingkaran yang terdapat pada Gambar 2, dari 30

kecamatan di wilayah Kabupaten Karawang yang telah melewati proses *clustering* menggunakan metode *K-Means* dibagi menjadi tiga *cluster* penyebaran yang dikategorikan menjadi tiga wilayah tingkat penyebaran tinggi yang memiliki tingkat kasus tertinggi, wilayah yang memiliki

tingkat penyebaran sedang/menengah yang memiliki jumlah kasus diantara tinggi dan rendah, dan wilayah yang memiliki tingkat wilayah penyebaran rendah memiliki tingkat kasus terendah.

Pada diagram lingkaran diatas yang dihasilkan dari proses *clustering* terdapat 16,7% yaitu 5 kelurahan yang memiliki tingkat penyebaran tinggi, 33,3% terdiri dari 10 kelurahan memiliki tingkat penyebaran sedang/menengah karena berada diantara persentasi tingkat tinggi dan rendah, dan 50% terdiri dari 15 kelurahan memiliki tingkat penyebaran rendah.

## KESIMPULAN DAN SARAN

Setelah dilakukan proses pengelompokan wilayah terdampak Covid-19 di Kabupaten Karawang dengan menggunakan metode *K-Means* dapat disimpulkan bahwa dari 30 kecamatan yang ada di Kabupaten Karawang 50% wilayah termasuk kedalam kategori terdampak rendah, 33,3% wilayah termasuk kedalam kategori terdampak sedang, dan 16,7% wilayah termasuk kedalam kategori terdampak tinggi.

Mengingat pandemi Covid-19 yang masih berlangsung maka diperlukan pengkajian lanjutan mengenai wilayah terdampak Covid-19 di Kabupaten Karawang dengan menggunakan data-data yang telah diperbaharui setiap harinya berdasarkan kasus yang terjadi di masing-masing kecamatan Kabupaten Karawang.

Untuk mengembangkan penelitian ini, dibutuhkan penelitian lanjutan metode *clustering* terhadap wilayah penyebaran Covid-19 yang lebih luas yaitu pada 34 provinsi di Indonesia yang dilakukan secara lebih *valid* dengan ditetapkan nilai *centroid* terbaik untuk mengetahui pengelompokan wilayah berdasarkan tingkat penyebaran Covid-19 di Indonesia.

## DAFTAR PUSTAKA

- [1] C. C. Lai, T. P. Shih, W. C. Ko, H. J. Tang, and P. R. Hsueh, "Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and coronavirus disease-2019 (COVID-19): The epidemic and the challenges," *Int. J. Antimicrob. Agents*, vol. 55, no. 3, p. 105924, 2020, doi: 10.1016/j.ijantimicag.2020.105924.
- [2] D. Indriyani and C. Virus, "Penanganan dan Pencegahan Pandemi Wabah Virus Corona ( Covid-19 )," vol. 2, 2020.
- [3] A. Susilo *et al.*, "Coronavirus Disease 2019: Tinjauan Literatur Terkini," *J. Penyakit Dalam Indones.*, vol. 7, no. 1, p. 45, 2020, doi: 10.7454/jpdi.v7i1.415.
- [4] I. Wahidah, M. A. Septiadi, M. C. A. Rafqie, N. Fitria, S. Hartono, and R. Athallah, "Pandemik Covid-19 : Analisis Perencanaan Pemerintah dan Masyarakat dalam Berbagai Upaya Pencegahan COVID-19 Pandemic :

- Analysis of Government and Community Planning in Various Prevention Measures,” vol. 11, no. 3, pp. 179–188, 2020.
- [5] C. Fikri, “Tiga Dampak Pandemi Covid-19 Bagi Perekonomian Nasional,” *Beritasatu.com*, p. 2021, 2021, [Online]. Available: <https://www.beritasatu.com/ekonomi/728997/tiga-dampak-pandemi-covid19-bagi-perekonomian-nasional>.
- [6] Y. F. Silitonga, “Seluruh Kecamatan di Karawang Zona Merah,” p. 2021, 2021.
- [7] S. Handoko, F. Fauziah, and E. T. E. Handayani, “Implementasi Data Mining Untuk Menentukan Tingkat Penjualan Paket Data Telkomsel Menggunakan Metode K-Means Clustering,” *J. Ilm. Teknol. dan Rekayasa*, vol. 25, no. 1, pp. 76–88, 2020, doi: 10.35760/tr.2020.v25i1.2677.
- [8] N. Dwitri, J. A. Tampubolon, S. Prayoga, F. Ilmi Zer, and D. Hartama, “Penerapan Algoritma K-Means Dalam Menentukan Tingkat Kepuasan Pembelajaran Online Pada Masa Pandemi Covid-19 di Indonesia,” *Jti (Jurnal Teknol. Informasi)*, vol. 4, no. 1, pp. 101–105, 2020.
- [9] M. H. Adiya and Y. Desnelita, “Jurnal Nasional Teknologi dan Sistem Informasi Penerapan Algoritma K-Means Untuk Clustering Data Obat-Obatan Pada RSUD Pekanbaru,” vol. 01, pp. 17–24, 2019.
- [10] M. R. Muttaqin and M. Defriani, “Algoritma K-Means untuk Pengelompokan Topik Skripsi Mahasiswa,” *Ilk. J. Ilm.*, vol. 12, no. 2, pp. 121–129, 2020, doi: 10.33096/ilkom.v12i2.542.121-129.
- [11] T. Suprawoto, “Klasifikasi data mahasiswa menggunakan metode k-means untuk menunjang pemilihan strategi pemasaran,” vol. 1, no. 1, pp. 12–18, 2016.
- [12] dkk Sugiono, “Pengelompokan Perilaku Mahasiswa Pada Perkuliahan E-Learning dengan K-Means Clustering,” *J. Kaji. Ilm.*, vol. 19, no. 2, pp. 126–133, 2019.
- [13] Z. I. Alfianti, M. A. Azis, A. Fauzi, F. T. Informasi, and U. B. Saranainformatika, “JurnalMantik,” vol. 4, no. 4, pp. 2336–2341, 2021.
- [14] S. Ramadani, I. Ambarita, and A. M. H. Pardede, “Metode K-Means Untuk Pengelompokan Masyarakat Miskin Dengan Menggunakan Jarak Kedekatan Manhattan City Dan Euclidean ( Studi Kasus Kota Binjai ),” *Inf. Syst. Dev.*, vol. 04, no. 2, pp. 15–29, 2019.
- [15] Satgas Covid-19 Karawang. “Pantauan kasus aktif Covid-19 untuk sebaran kecamatan” Internet: <http://covid19.karawangkab.go.id/data/>, tanggal di-update [20 Mei 2021].

# PEMBUATAN PROTOTYPE SMART HOME MENGGUNAKAN NODEMCU ESP8266 V3 DAN CHAT BOT PADA SMARTPHONE ANDROID

<sup>1</sup>Renaldi Dewangga Sindhu, <sup>2</sup>Ilmiyati Sari, <sup>3</sup>Dewi Putrie Lestari

<sup>1</sup>Fakultas Teknologi Industri Universitas Gunadarma, <sup>2,3</sup>Pusat Studi Komputasi Matematika Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok, 16424, Jawa Barat

<sup>1</sup>renaldidewanggasindhu@student.gunadarma.ac.id, <sup>2</sup>ilmiyati@staff.gunadarma.ac.id,

<sup>3</sup>dewi\_putrie@staff.gunadarma.ac.id

## Abstrak

*Internet of things adalah sebuah gagasan untuk membuat semua benda di dunia nyata dapat berkomunikasi satu sama lain sebagai bagian dari satu kesatuan sistem terpadu menggunakan jaringan internet sebagai penghubung. Pemanfaatan internet of things dapat digunakan untuk mengendalikan alat elektronik seperti pada pembuatan smart home. Smart home memadukan teknologi informasi dan teknologi komputasi yang diterapkan di dalam rumah ataupun bangunan yang dihuni oleh manusia dengan mengandalkan efisiensi, otomatisasi perangkat, kenyamanan, keamanan, dan penghematan perangkat elektronik rumah. Smart home yang dibuat dalam penelitian ini hanya dapat mengontrol lampu dan kipas angin dari jarak jauh serta dapat mengetahui status ketinggian air di rumah melalui smartphone. Tahap pembuatan aplikasi dimulai dari skema diagram rancangan aplikasi, flowchart, use case diagram smart home, rancangan rangkaian smart home, pembuatan chat bot, dan pengujian. Pembuatan smart home menggunakan NodeMCU ESP8266 V3 dan aplikasi chat bot pada smartphone Android. Setelah dilakukan uji coba, pengontrolan lampu dan kipas angin dari jarak jauh berhasil dilakukan dengan menggunakan smartphone berbasis Android yang terhubung ke internet. Selain itu, status ketinggian air dalam rumahpun dapat diketahui penghuni dimanapun berada.*

**Kata Kunci:** aplikasi chat bot, internet of things, NodeMCU ESP8266, smart home

## Abstract

*Internet of things is an idea to make all objects in the real world able to communicate with each other as part of an integrated system using the internet network as a connector. The utilization of the internet of things can be used to control electronic devices such as in construction of smart homes. Smart home combines information and computing technology that is applied in homes or buildings inhabited by humans by relying on efficiency, device automation, convenience, security, and savings in home electronic devices. The smart home made in this study can only control lights and fans remotely and can find out the status of the water level at home via smartphone. The application development stage starts from the application design schematic diagram, flowchart, use case diagram of smart home, circuit design of smart home, chat bot creation, and testing. Making a smart home using NodeMCU ESP8266 V3 and chat bot application on Android. After testing, controlling the lights and fans remotely was successfully carried out using an Android-based smartphone connected to the internet. In addition, the status of the water level in the house can be known wherever residents are.*

**Keywords:** chat bot application, internet of things, NodeMCU ESP8266, smart home



## PENDAHULUAN

Perkembangan teknologi yang sangat berkembang pesat saat ini telah mampu menjadikan internet sebagai media komunikasi dan kontrol terhadap perangkat dari jarak jauh selama masih terkoneksi dan saling terhubung. *Internet of things* merupakan sebuah pengembangan komunikasi jaringan dari benda yang saling terkait, terhubung satu dengan yang lain lewat komunikasi internet serta dapat saling bertukar data yang kemudian mengubahnya menjadi informasi [1]. *Internet of things* atau dikenal juga dengan singkatan IoT merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus [2-4]. Pada dasarnya perangkat IoT terdiri dari sensor sebagai media pengumpul data, sambungan internet sebagai media komunikasi dan *server* sebagai pengumpul informasi yang diterima sensor dan untuk analisa [5].

Revolusi industri 4.0 telah melahirkan teknologi *internet of things* yang memungkinkan adanya berbagai usaha untuk memberikan kemudahan dan kenyamanan bagi masyarakat dalam kegiatan sehari-hari. Pemanfaatan *internet of things* dapat digunakan untuk mengendalikan alat elektronik seperti pada pembuatan *smart home* [5]. Perangkat *smart home* adalah sebuah perangkat yang memiliki sistem otomatisasi sangat canggih untuk

mengendalikan lampu dan suhu, perangkat multi media untuk memantau dan menghidupkan sistem keamanan yang terhubung dengan pintu atau jendela dan beberapa fungsi yang lainnya [6]. Rumah pintar (*smart home*) merupakan sebuah aplikasi yang dirancang dengan bantuan komputer yang akan memberikan kenyamanan, keamanan, dan penghematan energi yang berlangsung secara otomatis sesuai dengan kendali pengguna dan terprogram melalui komputer pada gedung atau tempat tinggal pengguna [7-10]. Pengguna dapat memantau dan mengendalikan perangkat-perangkat listrik di dalam rumah dari jarak jauh melalui suatu saluran komunikasi seperti melalui jaringan internet, Wi-Fi atau Bluetooth [11].

Sistem *smart home* biasanya terdiri dari perangkat kontrol, monitoring, dan otomatisasi beberapa perangkat atau peralatan rumah yang dapat diakses melalui sebuah komputer [12]. Salah satu perangkat yang terpenting dalam *smart home* adalah perangkat kontrol yang berfungsi pada sistem pengendalian jarak jauh. Sistem kendali dan pemantauan perangkat ruangan pada *smart home* merupakan sebuah bentuk kendali dan dipantau secara otomatis terhadap alat-alat listrik rumah tangga, sistem penerangan atau sistem keamanan rumah yang semuanya mampu dikendalikan dan dipantau secara langsung sesuai keinginan oleh pemilik [13]. Saat ini pengendalian pada *smart home* dapat dilakukan dari jarak jauh dengan

menggunakan *smartphone*. Namun, perangkat *smartphone* harus terhubung dengan internet yang berfungsi sebagai jembatan penghubung antara alat dan sistem kontrol yang digunakan. Beberapa penelitian telah membuat sistem *smart home*. Orpwood, Gibbs, Adlam, Faulkner, dan Meegahawatte tahun 2005 yang memberikan analisis kebutuhan pengguna, rancangan dan evaluasi *smart home* untuk penderita demensia [14]. Sistem ini menggunakan infra red sebagai sensor yang efektifitas hasil luaran sebanding dengan jumlah radiasi infra red yang diterima. Robles dan Kim tahun 2010 melakukan penelitian untuk membuat *smart home* dengan teknologi PCS (*Power Carries System*) [15]. Kekurangan sistem ini yaitu sistem kontrol harus tersambung ke PC berbasis *windows* selama penggunaannya sehingga kurang efisien. Arifiyanto, Syafei, dan Somantri tahun 2013 telah menghasilkan pemantauan dan pengendalian perangkat listrik yang dapat dilakukan menggunakan *smartphone* melalui jaringan internet [16]. Komputer *server* sebagai pengendali pusat yang menghubungkan komunikasi data antara *smartphone* dengan perangkat mikrokontroler. Peran pengguna atau manusia masih terlihat pada model *smart home* ini.

Penelitian selanjutnya dilakukan oleh Dinata dan Sunanda pada tahun 2015 yang menghasilkan sistem *smart home* dengan *wireless* seperti internet *shield* untuk mengirimkan data energi listrik dari Arduino ke *web server* secara *real time* [17].

Parameter listrik yang dikirimkan antara lain Vrms, Irms, daya nyata, dan faktor daya. Kelebihan dari sistem ini adalah penggunaan energi listrik dapat dipantau melalui suatu *web* secara *real time*. Selain menggunakan komunikasi *wireless* dan jaringan internet untuk mengendalikan perangkat listrik jarak jauh, ada beberapa penelitian yang menggunakan jaringan TCP/IP dan komunikasi Bluetooth pada *smartphone* [18]. Kelemahan model komunikasi ini adalah hanya bersifat lokal saja atau terbatas jaraknya. Selain itu, Ramesh, Kumar, Vamsi dan Akarsh tahun 2017 telah menghasilkan sistem *smart home* untuk mengontrol perangkat berbasis IR melalui Wi-Fi [19].

Berdasarkan penelitian-penelitian tersebut maka dalam penelitian ini dibuat *smart home* yang dapat mengontrol kipas angin dan lampu dari jarak jauh melalui *smartphone* dengan *platform* Android. Selain itu, *smart home* yang dibuat juga memungkinkan penghuni rumah untuk memantau status ketinggian air dalam rumah walaupun penghuni sedang tidak berada di rumah dengan menggunakan sensor *water level*. *Water level* merupakan sensor yang berfungsi untuk mendeteksi ketinggian air dengan *output* analog kemudian diolah menggunakan mikrokontroler.

## **METODE PENELITIAN**

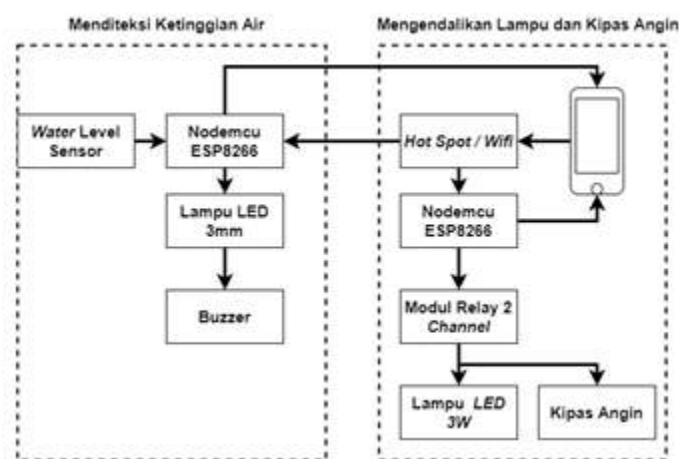
Alat yang diperlukan dalam pembuatan *smart home* pada penelitian ini antara lain

mikrokontroler, modul NodeMCU ESP8266 V3, *project board*, modul *relay 2 channel*, kabel *jumper*, lampu, *water level sensor*, *power bank*, dan *buzzer*. Pengontrolan *smart home* melalui *smartphone* dengan *platform* Android dilengkapi dengan aplikasi Telegram untuk membuat *chat bot*. *Chat bot* berfungsi untuk menuliskan perintah pengguna yang diberikan pada *smart home*. Skema diagram rancangan *smart home* yang dibuat dalam penelitian ini dapat dilihat pada Gambar 1.

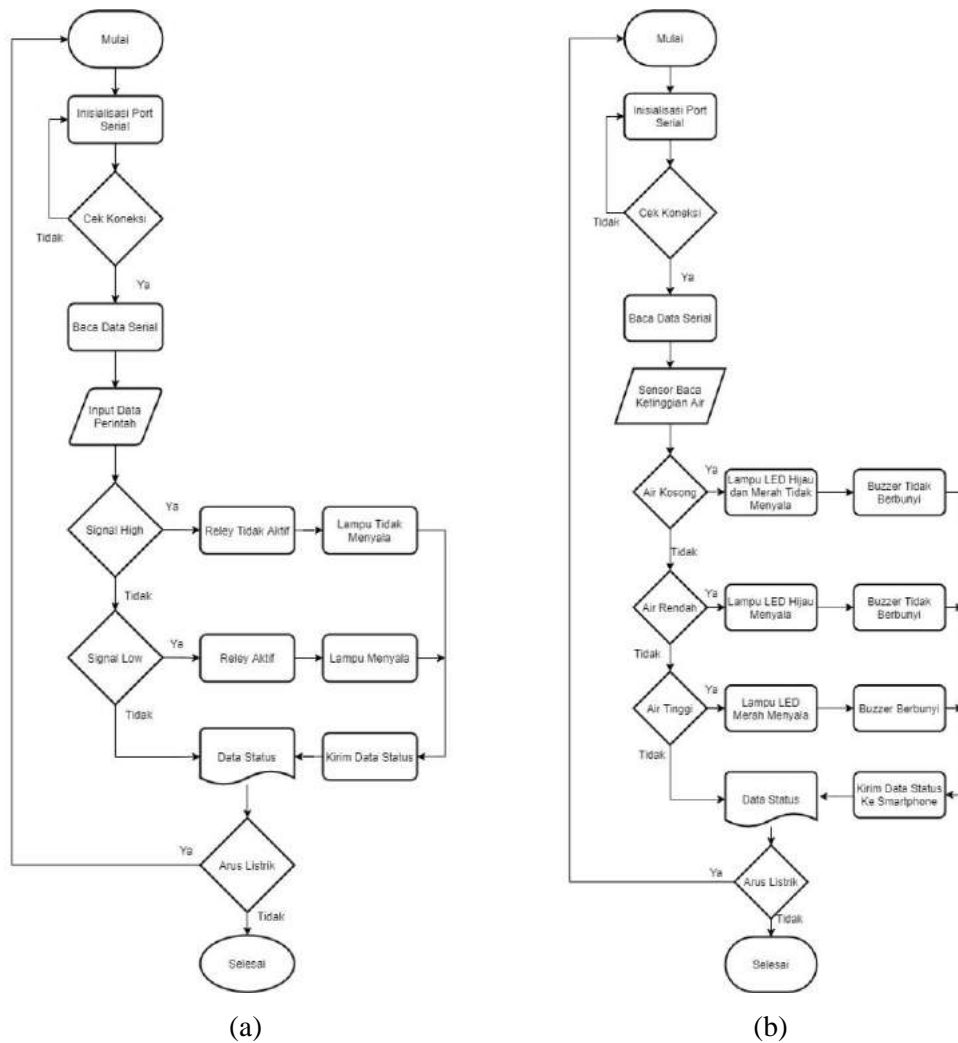
Langkah pertama pada Gambar 1 adalah memasang aplikasi Telegram pada *smartphone* dan membuat *chat bot* dalam aplikasi tersebut. *Chat bot* berfungsi untuk mendapatkan IP *address* yang digunakan pada rangkaian *smarthome*, sehingga *chat bot* dapat terhubung dengan rangkaian *smarthome*. Kemudian merangkai NodeMCU ESP8266 V3 dengan modul *relay 2 channel* dan *water level sensor*. Modul *relay 2 channel* selanjutnya dirangkai dengan

lampu LED 3W. Selain sebagai pemisah antara tegangan AC dan DC, *relay* juga berfungsi sebagai komponen utama untuk mematikan dan menghidupkan lampu serta kipas angin. *Water level sensor* berfungsi untuk mendeteksi ketinggian air di rumah. Jika air menyentuh *water level sensor* maka *buzzer* akan berbunyi, kemudian notifikasi status ketinggian air akan dikirim ke *smartphone*.

*Flowchart* pengendalian lampu dan kipas angin diberikan pada Gambar 2(a) dan *flowchart* deteksi status ketinggian air di rumah diberikan pada Gambar 2(b). Langkah pertama pada Gambar 2(a) dan 2(b) adalah melakukan inisialisasi *port* serial. Selanjutnya periksa koneksi. Jika tidak berhasil terkoneksi, maka lakukan inisialisasi *port* serial kembali. Jika *port* serial berhasil terkoneksi, maka akan dilakukan proses baca data serial. Kemudian pengguna dapat meng-*input* data perintah.



**Gambar 1. Skema Diagram Rancangan Smart Home**



**Gambar 2. Alur Kerja Pada Smart Home (a) Flowchart Pengendalian Lampu dan Kipas Angin (b) Flowchart Deteksi Ketinggian Air**

Pada Gambar 2(a), setelah perintah diinput oleh pengguna maka akan ada 4 kondisi: (1) jika perintah nyalakan lampu maka proses selanjutnya *relay* aktif sehingga lampu menyala setelah itu mengirim data status kemudian menampilkan data status ke *smartphone* bahwa lampu menyala, (2) jika perintah matikan lampu maka proses selanjutnya *relay* tidak aktif sehingga lampu tidak menyala setelah itu mengirim data status kemudian menampilkan data status pada *smartphone* bahwa lampu sudah mati

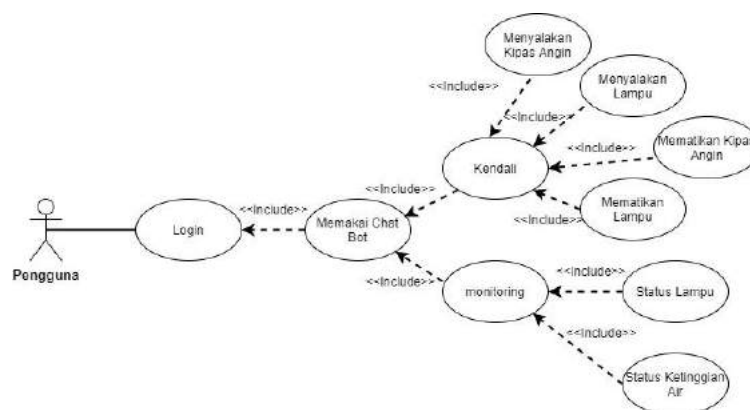
menyala, (3) jika perintah nyalakan kipas maka proses selanjutnya *relay* aktif sehingga kipas menyala setelah itu mengirim data status kemudian menampilkan data status ke *smartphone* bahwa kipas sudah menyala, dan (4) jika perintah matikan kipas maka proses selanjutnya *relay* tidak aktif sehingga kipas tidak menyala setelah itu mengirim data status kemudian menampilkan data status pada *smartphone* bahwa kipas sudah mati. Jika perintah tidak berada pada 4 kondisi yang telah disebutkan, maka akan menampilkan

data status di *smartphone* bahwa perintah yang dimasukan salah. Pada Gambar 2(b), setelah proses baca data serial berhasil dilakukan, langkah selanjutnya adalah melakukan proses sensor baca ketinggian air. Pada proses ini terdapat 3 kondisi yang dimungkinkan sebagai hasil baca *water level sensor*: (1) jika ketinggian air berada di keadaan air kosong maka lampu led berwarna hijau dan merah tidak akan menyala dan *buzzer* tidak berbunyi setelah itu status ketinggian air akan dikirim ke *smartphone*, (2) jika ketinggian air berada di keadaan air aman maka lampu led berwarna hijau akan menyala dan *buzzer* tidak berbunyi setelah itu status ketinggian air akan dikirim ke *smartphone*, dan (3) jika ketinggian air berada di keadaan air berbahaya maka lampu led berwarna merah akan menyala dan *buzzer* akan berbunyi setelah itu status ketinggian air akan dikirim ke *smartphone*. Gambar 3 memperlihatkan interaksi antara pengguna dan *smart home* yang dibuat. Pengguna harus *login* terlebih dahulu pada aplikasi Telegram

di *smartphone* untuk menggunakan *smart home*. Pengguna dapat menggunakan *chat bot* pada aplikasi Telegram setelah melakukan *login*. Ketika pengguna memakai *chat bot*, pengguna dapat mengendalikan lampu dan kipas angin yaitu menyalakan dan mematikan. Selain itu, pengguna dapat melakukan pemantauan terhadap kondisi lampu menyala atau tidak dan kondisi ketinggian air dengan cara mendapatkan pesan balasan dari *chat bot*.

## HASIL DAN PEMBAHASAN

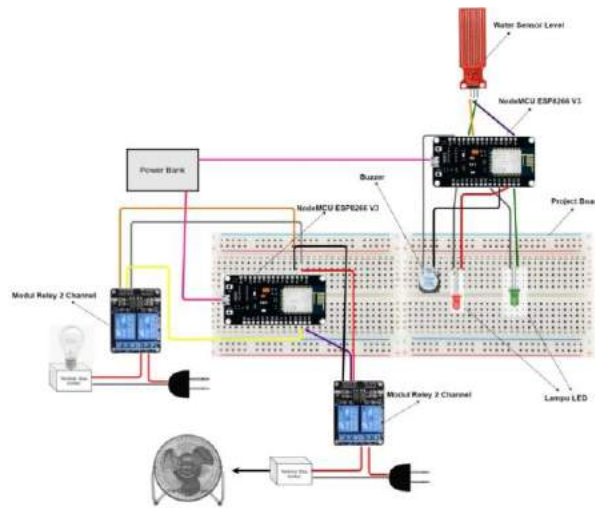
Penelitian ini menghasilkan sebuah sistem *smart home* yang dapat menghidupkan dan mematikan lampu dan kipas angin serta mendeteksi status ketinggian air melalui *smartphone*. Secara garis besar terdapat tiga langkah dalam pembuatan *smart home* pada penelitian ini, yaitu pembuatan rangkaian perangkat keras *smart home*, pembuatan *chat bot*, dan pengujian.



Gambar 3. Use Case Diagram Smart Home

Rancangan rangkaian *smart home* diberikan oleh Gambar 4(a) dan hasil rancangan rangkaian *smart home* ini dapat dilihat pada Gambar 4(b). Ada 10 tahap yang dibutuhkan untuk menghasilkan rangkaian *smart home* pada Gambar 4(b), antara lain:

1. Hubungkan pin VIN pada NodeMCU ESP8266 ke pin VCC modul *relay* menggunakan kabel *jumper*.
2. Hubungkan pin G pada NodeMCU ESP8266 ke pin GND modul *relay* menggunakan kabel *jumper*.
3. Hubungkan pin D2 pada NodeMCU ESP8266 ke pin IN 1 modul *relay* menggunakan kabel *jumper* yang berfungsi untuk menyalakan lampu.
4. Hubungkan pin D1 pada NodeMCU ESP8266 ke pin IN 2 modul *relay* menggunakan kabel *jumper* yang berfungsi untuk menyalakan kipas.
5. Hubungkan pin NodeMCU ESP8266 ke pin modul *relay*, kemudian masukan kabel lampu ke *port* NO 1 dan COM 1 untuk menyalakan lampu dan masukan kabel ke *port* NO 2 dan COM 2.
6. Hubungkan (kutub +) ke pin D1 pada NodeMCU ESP8266 dan (kutub -) ke pin G pada NodeMCU ESP8266 untuk menyalakan lampu LED berwarna hijau yang digunakan sebagai indikator deteksi ketinggian air aman.
7. Hubungkan (kutub +) ke pin D2 pada NodeMCU ESP8266 dan (kutub -) ke pin G pada NodeMCU ESP8266 untuk menyalakan lampu LED berwarna kuning yang digunakan sebagai indikator deteksi ketinggian air rendah.
8. Hubungkan (kutub +) ke pin D3 pada NodeMCU ESP8266 dan (kutub -) ke pin G pada NodeMCU ESP8266 untuk menyalakan lampu LED berwarna merah yang digunakan sebagai indikator deteksi ketinggian air tinggi.
9. Hubungkan *buzzer* (kutub +) ke pin D4 NodeMCU ESP8266, lalu hubungkan (kutub -) ke pin G pada NodeMCU ESP8266.
10. Hubungkan (kutub +) *water level sensor* ke pin tegangan 3V pada NodeMCU ESP8266 untuk menggunakan *water level sensor*, kemudian sambungkan (kutub -) dengan G lalu hubungkan pin S dengan sinyal analog yaitu A0 pada NodeMCU ESP8266.



(a)



(b)

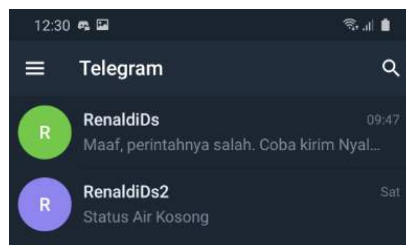
**Gambar 4. Sistem Smart Home (a) Rancangan Rangkaian Smart Home (b) Hasil Rangkaian Smart Home**

Langkah selanjutnya adalah pembuatan *chat bot* pada aplikasi Telegram. Langkah-langkah pembuatan *chat bot* ini sebagai berikut:

1. Buka perangkat lunak Telegram.
2. Ketik *botfather* pada kontak pencarian sehingga muncul banyak kontak bertuliskan *botfather*.
3. Pilih kontak paling atas sehingga muncul halaman baru.
4. Pilih *start* pada halaman tersebut.
5. Ketik perintah */newbot* sehingga *botfather* mengirim pesan balasan untuk memulai pembuatan *bot*.
6. Ketik nama *bot* yang akan dibuat dengan format penulisan nama *bot* yaitu *namabot\_bot*.
7. Proses pembuatan *chat bot* berhasil dilakukan jika mendapatkan API token yang digunakan untuk mengakses API *bot* dan untuk mengontrol NodeMCU ESP8266.

*Chat bot* pada aplikasi Telegram dapat digunakan jika *smartphone* terhubung ke jaringan internet. *Chat bot* berfungsi untuk menuliskan perintah dari pengguna kepada *smart home*. Pada Gambar 5 terdapat *chat bot* RenaldiDs yang digunakan untuk mengirim pesan atau perintah menyalakan atau mematikan lampu maupun kipas, sedangkan *chat bot* RenaldiDs2 digunakan untuk mendeteksi status ketinggian air. Pada langkah terakhir dilakukan proses pengujian. Pengujian dilakukan pada *smartphone* dan *chat bot*. Hasil pengujian ini dapat dilihat pada Gambar 6, Gambar 7, dan Gambar 8. Pada Gambar 6, ketika pengguna menuliskan perintah “Nyalakan Lampu”, maka perintah akan diterima oleh NodeMCU ESP8266 V3. Kemudian *relay* akan aktif dan lampu akan

menyala. Selanjutnya akan muncul pesan balasan pada *chat bot* bahwa “Lampu Menyala”. Jika pengguna menuliskan perintah “Nyalakan Kipas”, maka perintah akan diterima oleh NodeMCU ESP8266 V3. Kemudian *relay* akan aktif dan kipas akan menyala. Selanjutnya akan muncul pesan balasan pada *chat bot* bahwa “Kipas Sudah Menyala”. Gambar 7 adalah hasil yang terjadi pada *smart home* jika pengguna memberikan perintah menyalakan lampu dan kipas angin. Hal yang sama juga dapat dilakukan untuk mematikan lampu atau kipas angin. Ketika perintah yang dikirim salah maka *chat bot* akan mengirim pesan bahwa “Maaf, perintahnya salah. Coba kirim Nyalakan Lampu/Nyalakan Kipas atau Matikan Lampu/Matikan Kipas”.



**Gambar 5. Tampilan Awal *Chat Bot***

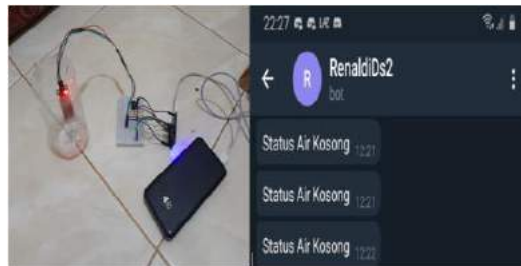


**Gambar 6. Tampilan Perintah Pada *Chat Bot***





**Gambar 7. Rangkaian *Smart Home* Jika Diberikan Perintah Hidupkan Lampu dan Kipas Angin**



**Gambar 8. Keadaan *Smart Home* dan Tampilan *Chat Bot* Saat Status Air Kosong**



**Gambar 9. Keadaan *Smart Home* dan Tampilan *Chat Bot* Saat Status Air Berbahaya**

Gambar 8 memperlihatkan keadaan *smart home* dan tampilan *chat bot* saat status air kosong. Ketika ketinggian air belum menyentuh *water level sensor*, maka sensor mendeteksi bahwa status air kosong yang ditampilkan pada serial monitor pada Arduino IDE. Setelah 30 detik akan terkirim pesan pada *chat bot* bahwa status air kosong dan di indikator lampu LED tidak menyala serta

*buzzer* tidak berbunyi. Pada Gambar 9, ketika ketinggian air sudah menyentuh *water level sensor* dengan kondisi ketinggian air tinggi, maka sensor mendeteksi bahwa status air tinggi yang ditampilkan pada serial monitor pada Arduino IDE. Setelah 30 detik akan terkirim pesan pada *chat bot* bahwa status air berbahaya dan indikator lampu LED menyala dengan warna merah serta *buzzer* berbunyi.

```

unsigned long waktusekarang kirim=millis();
if ( waktusekarang kirim-
waktusebelum kirim>=30000){
    digitalWrite(Buzzer,LOW);
robot.sendMessage(id, "Status Air
Berbahaya");
    Serial.println("Pesan Terkirim");
    waktusebelum kirim=millis();
}
}
delay(30000);
}

```

**Gambar 10. Pengaturan Pengiriman Pesan Status Ketinggian Air**

Gambar 10 memperlihatkan blok program yang berfungsi untuk mengirim pesan status ketinggian air ke *smartphone* berdasarkan hasil deteksi dari *water level sensor*. Pada bagian waktu kirim pesan diberi *input 30000 millisecond*, dimana jika diubah ke dalam *second* akan menjadi 30 *second* atau 30 detik. Hal ini menandakan bahwa pesan status ketinggian air baru masuk ke *smartphone* setelah 30 detik sensor mendeteksi adanya air atau dapat dikatakan bahwa proses pengiriman pesan status ketinggian air ke *smartphone* mengalami *delay* 30 detik.

## KESIMPULAN DAN SARAN

Berdasarkan hasil dari perancangan dan pengujian penerapan *internet of things* dalam pembuatan *smart home* menggunakan NodeMCU ESP8266 V3 dan *chat bot* pada *smartphone* Android dapat diambil

kesimpulan yaitu *smarthome* yang sudah dibuat dapat digunakan untuk mengendalikan lampu, kipas angin, dan mendeteksi status ketinggian air di rumah sesuai dengan yang diharapkan. Perangkat NodeMCU ESP8266 V3 merupakan sebuah mikrokontroler yang terjangkau tetapi benar-benar baik digunakan secara *standalone* (berdiri sendiri).

Penggunaan mikrokontroler NodeMCU ESP8266 V3 ini sangat bergantung dengan keadaan sinyal jaringan internet yang diterima, sehingga menyebabkan cepat atau lambatnya proses pengiriman perintah dari *smartphone* yang menggunakan *chat bot* dalam mematikan atau menghidupkan lampu dan kipas angin. Pada pendeteksian status ketinggian air, *water level sensor* hanya dapat membaca ketinggian air berdasarkan ketinggian alat tersebut. *Water level sensor* ini hanya bekerja ketika air sudah mengenai lempengan yang bergaris-garis pada sensor. Ketika *water level sensor* membaca status

ketinggian air seperti status air kosong, status air aman atau status air berbahaya, maka status ketinggian air tersebut langsung dikirim ke *smartphone* dengan proses pengiriman pesan mengalami *delay* 30 detik.

Sistem *smart home* dan *chat bot* pada penelitian ini selanjutnya dapat dikembangkan dengan memberikan status ketinggian air yang lebih lengkap serta dapat digunakan untuk alat elektronik yang memiliki daya listrik yang lebih besar. Tampilan rangkaian *smart home* yang dibuat pada penelitian ini juga masih sangat sederhana sehingga masih perlu dikembangkan.

#### DAFTAR PUSTAKA

- [1] B. Artono and F. Susanto, "Wireless smart home system menggunakan internet of things," *Jurnal Teknologi Informatika dan Terapan*, vol. 05, no. 01, pp. 17-24, 2017.
- [2] A. Arafat, "Sistem pengamanan pintu rumah berbasis Internet Of Things (IoT) dengan ESP8266," *Jurnal Ilmiah Technologia*, vol. 7, no. 4, pp. 262-268, 2016.
- [3] Y. Efendi, "Internet of Things (IOT) sistem pengendalian lampu menggunakan Raspberry PI berbasis mobile," *Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer Universitas Al Asyariah Mandar*, vol. 4, no. 1, pp. 19-26, 2018.
- [4] A. Kusumaningrum, A. Pujiastuti, and M. Zeny, "Pemanfaatan internet of things pada kendali lampu," *Compiler*, vol. 6, no. 1, pp. 53-59, 2017.
- [5] R. Y. Endra, A. Cucus, and M. B. Syahputra, *Smart room menggunakan internet of things untuk efisiensi biaya dan keamanan ruangan*. Bandar Lampung: Anugrah Utama Raharja, 2019.
- [6] D. Bregman, R. Blvd, and R. Lezion, "Smart home intelligence -the ehome that learns," *Int. J. Smart Home Smart*, vol. 4, no. 4, pp. 35-46, 2010.
- [7] F. Masykur and F. Prasetyowati, "Aplikasi rumah pintar (smart home) pengendali peralatan elektronik rumah tangga berbasis web," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 3, no. 1, pp. 51-58, 2016.
- [8] H. Setiadi and Munadi, "Desain model smarthome system berbasis mikrokontroler ATmega8535," *Jurnal Teknik Mesin*, vol. 3, no. 2, pp. 138-142, 2015.
- [9] F. G. Aditya, Hafidudin, and A. G. Permana, "Analisis dan perancangan prototype smart home dengan sistem client server berbasis platform Android melalui komunikasi wireless," *eProceedings of Engineering*, vol. 2, no. 2, pp. 3070-3077, 2015.
- [10] S. Kumar, "Ubiquitous smart home system using android application,"

- IJCNC, vol. 6, no. 1, pp. 33-43, 2014.
- [11] D. Kurnianto, A. M. Hadi, and E. Wahyudi, "Perancangan sistem kendali otomatis pada *smart home* menggunakan modul Arduino Uno," *Jurnal Nasional Teknik Elektro*, vol. 5, no. 2, pp. 260-270, 2016.
- [12] T. F. Yurmama and N. Azman, "Perancangan *software* aplikasi *pervasive smart home*," *Seminar Nasional Aplikasi Teknologi Informasi*, 2009.
- [13] F. Z. Rachman, "Smart home berbasis IOT," *Seminar Nasional Inovasi Teknologi Terapan*, pp. 369-374, 2017.
- [14] R. Orpwood, C. Gibbs, T. Adlam, R. Faulkner, and D. Meegahawatte, "The design of smart homes for people with dementia—user-interface aspects," *Universal Access in The Information Society*, vol. 4, no. 2, pp. 156-164, 2005.
- [15] J. R. Rosslin and K. Tai-hoon, "Applications, systems and methods in smart home technology: a review," *International Journal of Advanced Science and Technology*, vol. 15, pp. 37-47, 2010.
- [16] F. Arifiyanto, W. A. Syafei, and M. Somantri, "Perancangan *prototype web-based online smart home controlled by smartphone*," *TRANSIENT*, vol. 2, no. 4, 2013.
- [17] I. Dinata and W. Sunanda, "Implementasi *wireless* monitoring energi listrik berbasis *web database*," *JNTE*, vol. 4, no. 1, pp. 83-88, 2015.
- [18] A. S. Rafika, M. S. H. Putra, and W. Larasati, "Smart home automatic menggunakan media Bluetooth berbasis mikrokontroler ATmega328," *CCIT*, vol. 8, no. 3, pp. 215-222, 2015.
- [19] N. V. K. Ramesh, S. T. Kumar, V. Vamsi, and S. Akarsh, "Wi-Fi controlled universal remote using ESP8266," *ARPJN Journal of Engineering and Applied Sciences*, vol. 12, pp. 7233-7238, 2017.

# IMPLEMENTASI FEISTEL *BLOCK CIPHER* DALAM ENKRIPSI FILE BERBENTUK TEKS

<sup>1</sup> Ety Sutanty, <sup>2</sup> Meilani B. Siregar, <sup>3</sup>Esti Setyaningsih  
<sup>1,2,3</sup>Fakultas Teknologi Industri Universitas Gunadarma  
Jl. Margonda Raya 100, Depok 16424, Jawa Barat  
<sup>1,2,3</sup>{ety\_s, meilani, esti}@staff.gunadarma.ac.id

## Abstrak

Salah satu cara mengamankan file digital yang dikirimkan melalui jaringan komputer adalah kriptografi. Kriptografi melakukan pengamanan terhadap file digital yang dikirimkan dengan melakukan pengacakan informasi asli (plaintext) menggunakan kunci (key). Pada penelitian ini akan diimplementasikan salah satu metode dalam melakukan enkripsi terhadap file berbentuk Teks (txt, pdf, docx dan rtf) dengan maksimal input file digital berukuran 80 MB menggunakan Feistel Block Cipher dengan metode 3DES. Proses penyandian (enkripsi) algoritma DES diproses sebanyak 3 kali menggunakan 3 kunci dengan total kunci yang dibangkitkan menjadi 168 bit. Ketiga kunci yang digunakan dapat bersifat saling bebas, atau hanya 2 kunci saja yang saling bebas tergantung penggunaan yang dibutuhkan. Hasil pengujian menunjukkan ukuran file berpengaruh pada waktu yang dibutuhkan dalam proses enkripsi maupun dekripsi, selain itu proses yang berjalan pada sistem operasi juga mempengaruhi waktu pengenkripsian suatu file. Tabel hasil uji coba menunjukkan bahwa waktu yang dibutuhkan dalam proses enkripsi tidak jauh berbeda dengan waktu yang dibutuhkan dalam proses dekripsi pada file yang sama, yaitu memiliki selisih sebesar 0.03% dimana proses dekripsi lebih cepat daripada proses enkripsi.

**Kata Kunci:** Dekripsi, DES, Enkripsi, Feistel, Kunci

## Abstract

One way of sending digital files that are sent over a computer network is cryptography. Cryptography protects digital files sent by scrambling the original information (plaintext) using a key. In this study, one method will be implemented to encrypt files in the form of Text (txt, pdf, docx dan rtf) with a maximum input of 80 MB digital files using Feistel Block Cipher with the 3DES method. The DES algorithm's encryption (encryption) process becomes 3 times using 3 keys with a total key generated of 168 bits. The third key used can be independent, or only 2 keys are used depending on the required use. The test results show that the file size affects the time needed in the encryption and decryption process, besides that the process running on the operating system also affects the encryption time of a file. The table of test results shows that the time required for the encryption process is not much different from the time required for the decryption process on the same file, which has a difference of 0.03% where the decryption process is faster than the encryption process.

**Keywords:** Decrypt, DES, Encrypt, Feistel, Key, Video

## PENDAHULUAN

Salah satu metode dalam melakukan pengamanan terhadap file digital yang dapat

dikirimkan melalui jalur komunikasi [1] adalah metode kriptografi. Metode ini tidak mencegah terjadinya pencurian data [2] melainkan membuat data yang dicuri menjadi

data yang tidak dapat dimengerti oleh pencurinya [3]. Tingkat keamanan kriptografi dalam mengamankan sebuah data bergantung pada kerahasiaan kuncinya [4]. Dalam kriptografi terdapat berbagai algoritma dengan tingkat komputasi dan aturan yang berbeda-beda yang dapat diimplementasikan pada file digital berbentuk teks, salah satunya dengan menggunakan Feistel Block Cipher. Jaringan Feistel membutuhkan inputan 64-bit hasil permutasi awal dan 16 sub-kunci yang masing-masing berukuran 48-bit yang biasanya terdapat pada algoritma 3DES (*Data Encryption Standart*). Algoritma 3DES menggunakan 3 kunci dengan total kunci yang dibangkitkan menjadi 168 bit. Ketiga kunci yang digunakan dapat bersifat saling bebas, atau hanya 2 kunci saja yang saling bebas tergantung penggunaan yang dibutuhkan. Penelitian terkait implementasi Jaringan Feistel dalam melakukan enkripsi terhadap file digital dilakukan peneliti terdahulu. Penelitian [5] menggunakan pemrograman VHDL (*VHSIC Hardware Description Language*) untuk mengimplementasikan Feistel Block Cipher dengan algoritma 3DES dalam chip FPGA. Implementasi 3DES dalam sebuah chip FPGA membutuhkan banyak gerbang-gerbang digital, terutama karena operasinya yang 64 bit. Fungsi utama dari FPGA sendiri adalah dalam pengkodean dan pendekodean pesan yang dikirim atau diterima oleh mikrokontroler. Algoritma 3DES diimplementasikan ke dalam FPGA dengan cara

dibagi-bagi menjadi modul-modul yang dibuat menggunakan VHDL. Penelitian [5] menggunakan tiga modul utama yang membutuhkan *resource* FPGA cukup besar antara lain *rounding* untuk implementasi *F-function*, penjadwalan kunci (*key scheduling*) pada tahap *preprocessing* dan control (termasuk *multiplexer*) [6]. Penelitian dengan mengimplementasikan penggunaan jaringan Feistel pada algoritma 3DES digunakan peneliti [7] dengan menggunakan verilog yang merupakan salah satu bahasa HDL (*Hardware Description Language*). Implementasi jaringan Feistel dengan penggunaan algoritma 3DES dilakukan peneliti [8] dengan mengimplementasikan pada suatu aplikasi yang dapat mengenkripsi sekaligus mendekripsikan pesan atau file. Aplikasi dibuat dengan menggunakan alat bantu Matlab, Aplikasi tersebut dioperasikan untuk mengetahui waktu yang dibutuhkan untuk melakukan proses enkripsi dan dekripsi. Penelitian ini juga melakukan uji keamanan algoritma 3DES dari serangan *brute force*. Peneliti lain mengkombinasikan 3DES dengan Base64 sebagai algoritma pengamanan pada proses *handshaking* suatu aplikasi Delphi yang diimplementasikannya pada suatu CV tertentu [9]. Selain itu, jaringan Feistel yang diimplementasikan pada algoritma 3DES tidak hanya diimplementasikan namun juga diuji oleh beberapa peneliti dengan cara membandingkan algoritma 3DES dengan algoritma simetri lainnya [10].

Pada penelitian ini, diimplementasikan penggunaan Feistel Block Cipher untuk melakukan enkripsi pada File Teks. Proses enkripsi dilakukan setelah proses inisialisasi permutasi selesai. Plaintext berupa file digital berbentuk Teks dibagi menjadi dua bagian yaitu L0 dan R0 dimana kedua bagian berukuran 32-bit akan diproses sebanyak 16 kali/putaran. Implementasi jaringan feistel pada enkripsi file teks ini dapat dikembalikan seperti semula (dekripsi) walaupun diproses dengan banyak iterasi sehingga pihak penerima pesan dapat memahami pesan asli.

## METODE PENELITIAN

Pada penelitian ini digunakan File teks, seperti : .txt, .pdf, .docx dan .rtf. Penelitian ini menggunakan data (pesan) dengan ukuran minimal 64-bit dan sebuah kunci simetri berukuran 56-bit. Ukuran data minimal tersebut merupakan satu blok data [11] yang nantinya akan diproses dengan menggunakan Feistel Block

### A. Pembangkitan Kunci (*Key Generation*)

Data *plaintext* diinput bersama dengan kunci enkripsi. Kunci enkripsi yang terdiri 8 karakter akan dikonversi ke dalam satuan

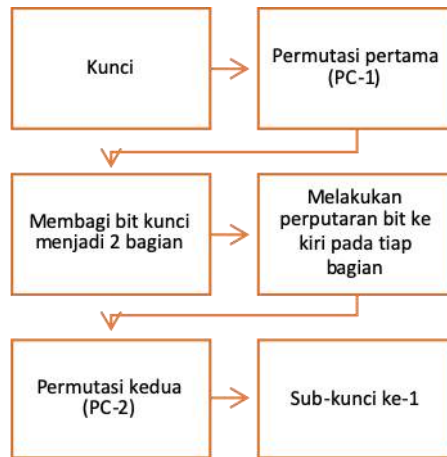
biner. Delapan karakter jika dikonversikan ke dalam byte menjadi 8 byte dan dalam bit menjadi 64-bit (yang dibutuhkan sebesar 56-bit). Ukuran minimal *plaintext* yang dapat dienkripsi sebesar 64-bit atau sebanyak 8 karakter. Ukuran minimal ini merupakan satu blok data yang dapat dienkripsi dimana jika ukuran *plaintext* lebih besar maka *plaintext* akan dibagi menjadi beberapa blok data [12], sedangkan jika ukuran *plaintext* lebih kecil maka *plaintext* akan ditambah dengan bit kosong hingga berukuran 64-bit. Penambahan karakter “spasi” dilakukan untuk menyesuaikan ukuran *plaintext*. Jumlah karakter “spasi” yang ditambahkan sesuai dengan jumlah kekurangan karakter (*jdk*) dari kelipatan 8, jika dinotasikan seperti persamaan 1.

$$jkk = 8 - lenP \text{ mod } 8 \quad (1)$$

Penambahan karakter dilakukan dengan menggunakan karakter “spasi” karena karakter “spasi” dianggap tidak menimbulkan banyak perubahan pada file hasil dekripsi.

### B. Pembentukan Sub-Kunci

Kunci enkripsi akan dibagi menjadi 16 sub-kunci dengan metode key schedule yang dapat dilihat pada Gambar 1.



**Gambar 1. Bagan Pembuatan Sub-kunci**

Kunci 64-bit akan diubah menjadi 16 sub-kunci yang masing-masing berukuran 48-bit. Penjelasan pembuatan sub-kunci sebagai berikut:

1. Pada setiap bit ke-8 dan kelipatannya pada kunci yang merupakan *odd parity* bit tidak diproses oleh algoritma, sehingga kunci yang digunakan sebesar 56-bit.
2. Selanjutnya sejumlah 56-bit dimasukkan pada tabel permutasi pertama(PC-1). Permutasi dilakukan untuk menukar bit dengan bit lain sesuai dengan tabel permutasi.

$pc1 = [ 56, 48, 40, 32, 24, 16, 8,$   
 $0, 57, 49, 41, 33, 25, 17,$   
 $9, 1, 58, 50, 42, 34, 26,$   
 $18, 10, 2, 59, 51, 43, 35,$   
 $62, 54, 46, 38, 30, 22, 14,$   
 $6, 61, 53, 45, 37, 29, 21,$   
 $13, 5, 60, 52, 44, 36, 28,$   
 $20, 12, 4, 27, 19, 11, 3]$

Tabel permutasi menunjukkan bahwa bit pertama (ke-0 jika index dimulai dari 0)

pada hasil PC-1 adalah bit ke-56 pada kunci sebelum dilakukan permutasi pertama, begitu juga seterusnya.

3. Hasil PC-1 dibagi menjadi 2 bagian, bagian kiri dan kanan yang masing-masing berukuran 28 bit.
4. Melakukan proses perputaran bit ke kiri sejumlah 1 atau 2 bit pada tiap bagian sesuai dengan putaran sub-kuncinya [13]. Aturan perputaran bit ke kiri sebagai berikut:

a. Pada putaran ke- $i = 1, 2, 9, 16$  ; kedua bagian tersebut melakukan perpindahan bit ke kiri sebanyak satu bit.

b. Pada putaran lainnya (selain  $i = 1, 2, 9, 16$ ) kedua bagian tersebut melakukan perpindahan bit ke kiri sebanyak dua bit.

Atau dapat dituliskan sesuai dengan urutannya:

$left\_rotations = [1, 1, 2, 2, 2, 2, 2, 2, 2, 1,$   
 $2, 2, 2, 2, 2, 2, 1]$

5. Melakukan permutasi kedua (PC-2) pada hasil perputaran bit dengan tabel permutasi



kedua untuk mendapatkan nilai sub-kunci yang berukuran 48-bit.

pc2 = [13, 16, 10, 23, 0, 4,  
2, 27, 14, 5, 20, 9,  
22, 18, 11, 3, 25, 7,  
15, 6, 26, 19, 12, 1,  
40, 51, 30, 36, 46, 54,  
29, 39, 50, 44, 32, 47,  
43, 48, 38, 55, 33, 52,  
45, 41, 49, 35, 28, 31 ]

- Proses diatas terus dilakukan sebanyak 16 kali, sehingga menghasilkan 16 sub-kunci yang nantinya akan digunakan pada proses enkripsi pada iterasi masing-masing sub-kunci.

### C. Permutasi Awal

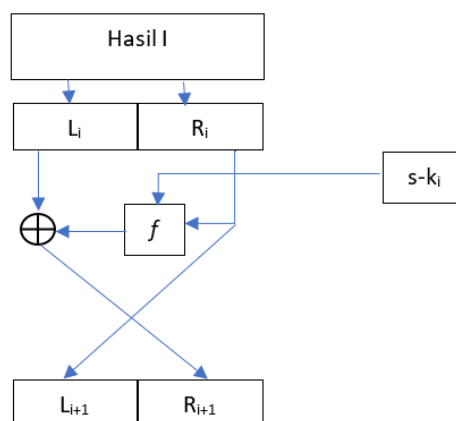
Proses permutasi kembali dibutuhkan untuk menyelesaikan algoritma DES. Perbedaan proses permutasi awal atau biasa disebut initial permutation (IP) berada pada tabel pertukaran bit [14] yang digunakan dan

IP tidak melakukan permutasi pada kunci melainkan pada plaintext yang ingin dienkripsi. Inputan pada proses IP sebesar 64-bit dengan keluaran yang sama yaitu 64-bit.

ip = [ 57, 49, 41, 33, 25, 17, 9, 1,  
59, 51, 43, 35, 27, 19, 11, 3,  
61, 53, 45, 37, 29, 21, 13, 5,  
63, 55, 47, 39, 31, 23, 15, 7,  
56, 48, 40, 32, 24, 16, 8, 0,  
58, 50, 42, 34, 26, 18, 10, 2,  
60, 52, 44, 36, 28, 20, 12, 4,  
62, 54, 46, 38, 30, 22, 14, 6 ]

### D. Enkripsi dengan Jaringan Feistel

Enkripsi merupakan proses merubah *plaintext* menjadi *ciphertext*. Penelitian ini mengimplementasikan penggunaan jaringan feistel agar file yang telah dienkripsi dapat dikembalikan seperti semula [15] walaupun diproses dengan banyak iterasi seperti dapat dilihat pada Gambar 2.



Gambar 2. Bagan Jaringan Feistel

Jaringan Feistel membutuhkan inputan 64-bit hasil permutasi awal dan 16 sub-kunci yang masing-masing berukuran 48-bit. Berikut langkah penyelesaian dengan menggunakan jaringan feistel:

1. Membagi hasil permutasi awal menjadi 2 bagian, bagian kiri ( $L_i$ ) dan kanan ( $R_i$ ) yang masing-masing berukuran 32-bit.
2. Menyalin bagian kanan untuk menjadi bagian kiri iterasi berikutnya ( $L_{i+1}$ ).
3. Menyisipkan bagian kanan bersamaan dengan sub-kunci kedalam fungsi  $f$  pada DES.
4. Melakukan operasi XOR pada hasil fungsi  $f$  dengan bagian kiri ( $L_i$ ) untuk menjadi bagian kanan pada iterasi berikutnya ( $R_{i+1}$ ).
5. Mengulangi langkah ke-2 sampai ke-4 hingga iterasi ke-16.

Tahap penyelesaian fungsi  $f$  pada DES sebagai berikut:

1. Melakukan *expansion* pada plaintext bagian kanan ( $R_i$ ) 32 bit menjadi 48 bit. Expansion dilakukan dengan melakukan permutasi sesuai dengan tabel permutasi E.

```
expansion_table = [ 31, 0, 1, 2, 3, 4,
                    3, 4, 5, 6, 7, 8,
                    7, 8, 9, 10, 11, 12,
                    11, 12, 13, 14, 15, 16,
                    15, 16, 17, 18, 19, 20,
                    19, 20, 21, 22, 23, 24,
                    23, 24, 25, 26, 27, 28,
                    27, 28, 29, 30, 31, 0 ]
```

2. Hasil permutasi E dioperasikan dengan sub-kunci dengan menggunakan operasi XOR.
3. Hasil operasi XOR akan dibagi menjadi 8 bagian yang masing-masing berukuran 6-bit.
4. Tiap bagian akan dilakukan substitusi sesuai dengan tabel S-box-nya masing-masing sehingga setiap bagian menghasilkan 4-bit data.

```
sbox =
[# S1
[14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0,
7,
0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3,
8,
4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5,
0,
15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6,
13],
# S2
[15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5,
10,
3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11,
5,
0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 2,
15,
13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14,
9],
..
..
..
# S8
[13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12,
7,
1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9,
2,
```

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

5. Kedelapan bagian yang ukurannya telah menjadi 4-bit digabungkan kembali menjadi 32 bit lalu dilakukan permutasi P.

$p = [$  15, 6, 19, 20, 28, 11,  
27, 16, 0, 14, 22, 25,  
4, 17, 30, 9, 1, 7,  
23, 13, 31, 26, 2, 8,  
18, 12, 29, 5, 21, 10,  
3, 24  $]$

#### E. Permutasi akhir

Permutasi akhir ( $IP^{-1}$ ) dilakukan setelah jaringan Feistel diproses sebanyak 16 iterasi. Permutasi dilakukan sesuai dengan tabel permutasi akhir (*Final Permutation*).

$fp = [$  39, 7, 47, 15, 55, 23, 63, 31,  
38, 6, 46, 14, 54, 22, 62, 30,  
37, 5, 45, 13, 53, 21, 61, 29,  
36, 4, 44, 12, 52, 20, 60, 28,  
35, 3, 43, 11, 51, 19, 59, 27,  
34, 2, 42, 10, 50, 18, 58, 26,  
33, 1, 41, 9, 49, 17, 57, 25,  
32, 0, 40, 8, 48, 16, 56, 24  $]$

Hasil permutasi akhir berupa 64-bit data yang telah diacak dalam bentuk biner, sehingga ketika data biner tersebut diubah menjadi bentuk karakter maka karakter yang didapat merupakan karakter yang berbeda.

## HASIL DAN PEMBAHASAN

Studi kasus dilakukan sebagai bahan uji penelusuran alur proses menggunakan Feistel Block Cipher. Sebagai contoh aplikasi kriptografi akan mengenkripsi plaintext “pandemi corona” dengan 3 kunci simetri “enkripsi”, “dekripsi”, “shenozar”.

1. Aplikasi akan menghitung jumlah karakter plaintext. Jumlah karakter pada plaintext “pandemicorona” adalah 13 karakter. Plaintext ini tidak berukuran kelipatan 8 karakter/64-bit. Algoritma tidak dapat memproses plaintext ini, maka aplikasi menambahkan karakter spasi sesuai dengan jumlah karakter yang kurang. Jumlah kekurangan karakter (jkk) didapat dari notasi dibawah ini.

$$\begin{aligned} jkk &= 8 - \text{jumlah\_karakter} \text{ modulus } 8 \\ &= 8 - 13 \text{ mod } 8 = 8 - 5 = 3 \text{ karakter} \end{aligned}$$

Sehingga plaintext saat ini menjadi “pandemicorona ”

2. Aplikasi akan mengkonversi inputan baik plaintext maupun kunci ke dalam satuan biner.

Plaintext “pandemicorona” = 01110011  
01101000 01100001 01110011  
01100001 00100000 01110010  
01100001 01100111 01100001  
01111010 01111010 01101001  
00100000 00100000 00100000  
Kunci-1 “enkripsi” = 01100101  
01101110 01101011 01110010  
01101001 01110000 01110011  
01101001

Kunci-2 "dekripsi" = 01100100  
 01100101 01101011 01110010  
 01101001 01110000 01110011  
 01101001  
 Kunci-3 "shenozar" = 01110011  
 01101000 01100101 01101110  
 01101111 01111010 01100001  
 01110010

3. Aplikasi melakukan menentukan sub-kunci pertama hingga ke-16 untuk kunci-1 "enkripsi"

a. Permutasi pertama (PC-1)

Permutasi awal membuat 64-bit kunci menjadi 56-bit.

Hasil PC-1 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0]

b. Membagi bit menjadi 2 bagian

Bagian L 28-bit = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0]

Bagian R 28-bit = [0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0]

c. Perputaran bit

Perputaran bit ke kiri untuk sub-kunci pertama berjumlah 1 putaran.

Bagian L 28-bit = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0]

Bagian R 28-bit = [1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0]

d. Permutasi kedua (PC-2)

Kedua bagian disatukan kembali lalu dilakukan permutasi ke-2.

Hasil PC-2 = [1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1] = sub-kunci pertama.

e. Perputaran bit kembali dilakukan untuk mendapatkan sub-kunci lainnya. Berikut sub-kunci ke-2 hingga ke-16.

Sub-kunci ke-2 = [1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1]

Sub-kunci ke-3 = [1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0]

....

.....

Sub-kunci ke-16 = [1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0]

4. Setelah sub-kunci didapat akan mulai mengenkripsi plaintext. sebelum plaintext dipermutasi dengan IP, plaintext akan dibagi menjadi per-blok yang tiap bloknnya berjumlah 8 karakter = "pandemic"  
 "pandemic" = 01110011 01101000  
 01100001 01110011 01100001  
 00100000 01110010 01100001

- a. Plaintext dipermutasi dengan IP  
 IP = [1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1]
- b. Hasil IP dibagi menjadi 2 bagian  
 Bagian L 32-bit = [1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1]  
 Bagian R 32-bit = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1]
- c. Bagian  $R_i$  menjadi  $L_{i+1}$ .  
 Bagian  $L_2$  32-bit = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1]
- d. Bagian  $R_i$  bersama sub-kunci ke- $i$  dimasukkan ke dalam fungsi  $f$ .
- i. Bagian R dipermutasi dengan tabel e-box (expansion table)  
 Hasil e-box 48-bit = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0]
- ii. Bagian R XOR sub-kunci ke-1  
 Hasil XOR 48-bit = [0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1]
- iii. Hasil XOR dibagi 8 bagian, yang setiap bagian dimasukkan ke dalam tiap 8 s-box  
 Hasil XOR dibagi 8 bagian = [0, 1, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 1, 0, 0, 0], [1, 0, 0, 1, 0, 0, 0, 0], [0, 1, 1, 0, 0, 0, 0], [1, 1, 0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 1, 0, 0, 0], [1, 1, 1, 0, 1, 0, 0, 1], [0, 0, 1, 0, 0, 0, 1, 0, 0, 1]
- iv. Hasil S-box disatukan kembali lalu dilakukan permutasi P menjadi hasil fungsi  $f$   
 Hasil S-box = [5] [11] [4] [11] [9] [15] [2] [10]  
 Dalam biner = [0101], [1011], [0100], [1011], [1001], [1111], [0010], [1010]  
 Hasil permutasi P 32-bit = [1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0]
- e. Hasil fungsi  $f$  dioperasikan dengan bagian  $L_i$  menggunakan operasi XOR menjadi  $R_{i+1}$ .  
 Bagian  $R_2$  32-bit = [0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1]
- f.  $L_2$  dan  $R_2$  telah didapatkan. Proses terus diulang hingga iterasi ke-16 menggunakan sub-kunci sesuai iterasinya.
- g.  $L_{16}$  dan  $R_{16}$  disatukan untuk dilakukan permutasi akhir ( $IP^{-1}$ )  
 Hasil  $IP^{-1}$  = [1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0]

5. Ciphertext untuk plaintext “pandemic” telah ditemukan (Hasil IP<sup>-1</sup>).

6. Blok plaintext lainnya “orona” lalu diproses dengan menggunakan langkah-langkah *enciphering* yang sama.

Ciphertext blok-2 = [0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0]

7. Proses enkripsi ke-1 telah selesai diproses. Kini aplikasi akan mengenkripsi hasil enkripsi ke-1 ke dalam proses dekripsi ke-2 dengan menggunakan kunci ke-2 “dekripsi”(proses pembentukan sub-kunci juga kembali dilakukan).

Ciphertext ke-2 blok ke-1 = [0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1]

Ciphertext ke-2 blok ke-2 = [1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1]

8. Hasil enkripsi ke-2 lalu dienkrpsi kembali menggunakan kunci ke-3 “shenozar”.

Ciphertext ke-3 blok ke-1 = [1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0]

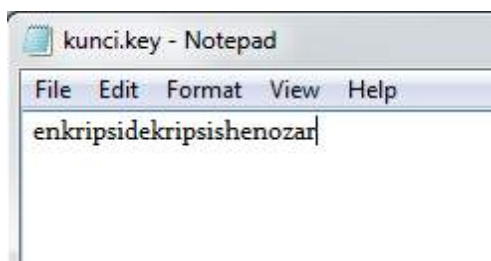
Ciphertext ke-3 blok ke-2 = [0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1]

9. Ciphertext implementasi Feistel Block Cipher telah didapatkan. Mode operasi yang digunakan adalah mode ECB (*Electronic Codebook Mode*) maka tiap blok hanya perlu disatukan tanpa melalui proses tertentu.

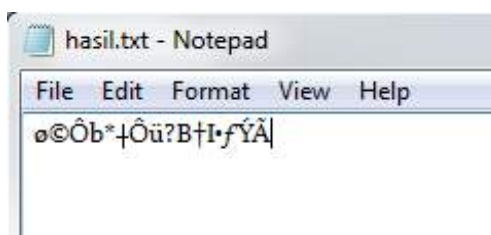
Plaintext = “pandemicorona ”

Ciphertext dalam biner = [1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1]

Keluaran *ciphertext* dikonversi ke dalam biner agar bisa dicocokkan dengan hasil perhitungan dapat dilihat pada Gambar 3 dan Gambar 4. Gambar 4 menunjukkan hasil ciphertext pada studi kasus dengan Teks: pandemicorona. Gambar 5 menunjukkan hasil konversi ciphertexts kedalam notasi biner. Implementasi terhadap Feistel Block Cipher juga diuji dengan cara menginput beberapa macam ekstensi file teks yang berbeda, mulai dari dari file *plaintext.txt*, file *format-text(.docx, .pdf, .rtf)* seperti dapat dilihat pada Tabel 1.



Gambar 3. Kunci Feistel Block pada Studi Kasus



Gambar 4. *Ciphertext* studi kasus

```

IDLE 2.6.6      ==== No Subprocess ====
>>> a='ø@Ôb*+Ôü?B+I•fYÃ'
>>> ' '.join(format(ord(i), 'b').zfill(8) for i in a)
'11111000 10101001 11010100 01100010 00101010 00010000 110101
00 11111100 00111111 01000010 10000110 01001001 10010101 1000
0011 11011101 11000011'
>>> |

```

Gambar 5. Konversi *Ciphertext* ke dalam Biner

Tabel 1. Hasil Uji Coba Aplikasi pada Berbagai File Digital

No	Nama file	Ukuran file	Waktu Proses Enkripsi (s)	Waktu Proses Dekripsi (s)
1.	nama.txt	1 kb (13 karakter)	0.00058	0.00048
2.	teks2.txt	1 kb (455 karakter)	0.00077	0.00075
3.	teks3.txt	3 kb (2716 karakter)	0.00212	0.00213
4.	doc1.docx	13 kb	0.00834	0.00830
5.	jarkom1.docx	494 kb	0.29117	0.28510
6.	pweb7.docx	1.301 kb	0.81959	0.74977
7.	ebook4.pdf	625 kb	0.35708	0.36676
8.	ebook5.pdf	2.503 kb	1.43324	1.41381
9.	ebook9.pdf	4.213 kb	2.44992	2.39780
10.	rtf1.rtf	45 kb	0.02525	0.02729

Seperti dapat dilihat pada Tabel 1 hasil ujicoba enkripsi terhadap beberapa input file digital, terdapat perbedaan waktu enkripsi pada file yang ukurannya berbeda. Semakin besar file, maka waktu proses enkripsi berlangsung akan semakin lama. Terdapat selisih waktu 0.03% antara proses dekripsi

menjadi pesan asli dengan proses enkripsi pesan asli.

## KESIMPULAN DAN SARAN

Implementasi Feistel Block Cipher berhasil melakukan enkripsi terhadap file plaintext berbentuk teks. Ukuran file berpengaruh pada waktu yang dibutuhkan dalam proses enkripsi maupun dekripsi, selain itu proses yang berjalan pada sistem operasi juga mempengaruhi waktu pengenkripsian suatu file. Tabel hasil uji coba menunjukkan bahwa waktu yang dibutuhkan dalam proses enkripsi tidak jauh berbeda dengan waktu yang dibutuhkan dalam proses dekripsi pada file yang sama, yaitu memiliki selisih sebesar 0.03% dimana proses dekripsi lebih cepat daripada proses enkripsi.

Pengembangan dapat dilakukan dengan menambahkan fitur/fungsi baru seperti pemberitahuan ketika file input tidak sesuai (ukuran file melebihi 70MB) serta penggabungan antara metode simetris dan asimetris untuk mendapatkan ciphertext yang benar-benar aman.

## DAFTAR PUSTAKA

- [1] L.Y. Zhang, Y. Liu, F. Pareschi, Y. Zhang, K.W. Wong, R. Rovatti, and G. Setti, "On the security of a class of diffusion mechanisms for image encryption," *IEEE Transactions on Cybernetics*, vol. 48, no. 4, pp. 1-13, 2017
- [2] X. Zhang, Y. Niu, C. Shen, and G. Cui, "Fluorescence resonance energy transfer-based photonic circuits using single-stranded tile self-assembly and DNA strand displacement," *Journal of Nanoscience and Nanotechnology*, vol. 17, no. 2, pp. 1053–1060, 2017
- [3] J. P. L. Cox, "Long-term data storage in DNA," *Trends in Biotechnology*, vol. 19, no. 7, pp. 247-250, 2001
- [4] X. Zhang, Y. Wang, G. Cui, Y. Niu, and J. Xu, "Application of a novel IWO to the design of encoding sequences for DNA computing," *Computers and Mathematics with audioApplications*, vol. 57, no. 11-12, pp. 2001–2008, 2009
- [5] Y. Erlich, and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950-954, 2016
- [6] S.P. Venigalla, M N. Babu, S. Boddu, and G.S.S. Vemana, "Implementation Of The Triple-Desblock Cipher Using Vhdl", *International Journal Of Advances In Engineering & Technology (IJAET)*, India. ISSN: 22311963117. Vol. 3, Issue 1, Pp. 117-128, 2012
- [7] M. S.Narula dan S. Singh, "Implementation Of Triple Data Encryption Standard Using Verilog", *International Journal Of Advanced Research In Computer Science And*



- Software Engineering (IJARCSSE)*, Page 667 Volume 4 , Issue 1. ISSN: 2277 128X, 2014.
- [8] A. Hidayat, *Enkripsi dan Dekripsi Data dengan Algoritma 3DES*, Bandung: Universitas Padjadjaran, 2008.
- [9] Rahim, M. Abdul, *Implementasi Kombinasi Algoritma 3DES dan Algoritma Base64 pada Sistem Keamanan Handshaking Animation Store di CV*. Edukreasi, Semarang: Universitas Dian Nuswantoro, 2013
- [10] H.O. Alanazi, B.B.Zaidan, A.A.Zaidan, H.A. Jalab, M.S., and Y.Al-Nabhani. “New Comparative Study Between DES, 3DES, and AES within Nine Factors”, *Journal Of Computing*. Page 152 Volume 2, Issue 3. ISSN: 2151-9617, 2010
- [11] W. Stallings, *Cryptography and Network Security*, Principles and Practice, Third Edition, Pearson, 2003.
- [12] H. Feistel, “Cryptography and Computer Privacy”, *Scientific American*, Vol. 228, No.5, pp. 15-23, 1973.
- [13] V.U.K. Sastry and K. Anup Kumar, “A Modified Feistel Cipher involving a key as a multiplicand on both the sides of the Plaintext matrix and supplemented with Mixing, Permutation and XOR Operation”, *International Journal of Computer Technology and Applications*, ISSN 2229-6093, Vol 3 (1), pp, 23-31 , 2012.
- [14] V.U.K. Sastry and K. Anup Kumar, “A Modified Feistel Cipher involving a key as a multiplicand on both the sides of the Plaintext matrix and supplemented with Mixing, Permutation and Modular Arithmetic Addition”, *International Journal of Computer Technology and Applications*, ISSN 2229-6093, Vol 3 (1), pp, 32-39 , 2012.
- [15] V.U.K. Sastry and K. Anup Kumar, “A Modified Feistel Cipher Involving a Key as a Multiplicand on Both the Sides of the Plaintext Matrix and Supplemented with Mixing, Permutation, and Modular Arithmetic Addition”, *International Journal of Computer Science and Information Technologies* , ISSN 0975 – 9646. Vol. 3 (1) , 2012, pp, 3133 – 3141,2012.

# IMPLEMENTASI ALGORITMA KLASIFIKASI SUPPORT VECTOR MACHINE UNTUK ANALISA SENTIMEN PENGGUNA TWITTER TERHADAP KEBIJAKAN PSBB

<sup>1</sup>Heart Parasian PR Zuriel, <sup>2</sup>Achmad Fahrurozi  
<sup>1,2</sup>Fakultas Teknologi Industri, Universitas Gunadarma  
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat  
<sup>1</sup>simanungkalitparasian@gmail.com, <sup>2</sup>achmad.fahrurozi12@gmail.com

## Abstrak

Pada saat ini penggunaan Twitter semakin luas. Pengguna Twitter dapat dengan bebas untuk berpendapat dan membagikan sudut pandang mereka mengenai isu tren dunia. Hal ini membuat konten Twitter menjadi beragam dan menarik untuk dianalisa, termasuk dengan tren ini membuat pemerintah mengeluarkan kebijakan yang bertujuan untuk menekan laju pertambahan orang yang terinfeksi virus. Kebijakan ini diberi nama Pembatasan Sosial Berskala Besar atau yang dikenal PSBB. Kebijakan ini pun hangat diperbincangkan di berbagai sosial media, tak terkecuali Twitter. Analisa sentimen dilakukan dengan menggunakan Support Vector Machine (SVM) sebagai algoritma klasifikasi pada data tweet yang berjumlah 22.335 data. Pelabelan data dalam penelitian ini dilakukan menggunakan metode Lexicon Based. Pada penelitian ini terdapat 4 model SVM yang dibangun menggunakan 4 fungsi kernel yaitu kernel Linear, RBF, Polinomial dan Sigmoid. Hasil klasifikasi dari masing-masing model diukur performanya menggunakan k-fold cross validation. Berdasarkan perhitungan, diperoleh bahwa performa model klasifikasi SVM dengan kernel RBF merupakan yang terbaik dibanding kernel lainnya dalam kasus penelitian analisa sentimen ini. Nilai accuracy, precision, recall, dan f1-score-nya yang diperoleh model klasifikasi dengan kernel RBF secara berturut-turut adalah 95.94%, 94.41 %, 97.8%, dan 96.08%. Model klasifikasi dengan kernel RBF ini memberikan mengklasifikasikan 11.764 (52.7%) data tweet ke dalam kelas positif dan 10.571 (47.3%) data tweet ke dalam kelas negative. Hasil tersebut memberikan kesimpulan bahwa pengguna Twitter cenderung bersentimen positif terhadap kebijakan PSBB.

**Kata Kunci:** Analisa Sentimen, Fungsi Kernel, PSBB, Support Vector Machine, Twitter

## Abstract

Nowadays, the use of Twitter is increasingly widespread. Twitter users can freely express their opinions and share their point of view on world trending issues. This makes Twitter content diverse and interesting to analyze, including the trend of government policies that are widely discussed in Indonesia. The emergence of the Covid-19 pandemic has prompted the government to issue a policy that aims to reduce the rate of increase in people infected with the virus. This policy is called Large-Scale Social Restrictions or what is known as PSBB. This policy is also hotly discussed on various social media, including Twitter. Sentiment analysis is carried out using the Support Vector Machine (SVM) as a classification algorithm for tweet data, totaling 22,335 data. Data labeling in this study done by using Lexicon Based method. In this study, there are 4 SVM models built using 4 kernel functions, namely the Linear kernel, RBF, Polinomial and Sigmoid. In this study, there are 4 SVM models built using 4 kernel functions, namely the Linear kernel, RBF, Polinomial and Sigmoid. The results of the classification of each model are measured for their performance using k-fold cross validation. Based on the calculations, it is found that the performance of the SVM classification model with the RBF kernel is the best compared to other kernels in the case of this sentiment analysis research. The accuracy, precision, recall, and f1-score values obtained by the classification

model with the RBF kernel were 95,94%, 94,41 %, 97,8%, and 96,08%, respectively. This classification model with the RBF kernel provides classifying 11.764 (52,7%) tweet data into a positive class and 10.571 (47,3%) tweet data into a negative class. These results conclude that Twitter users tend to have a positive sentiment towards the PSBB policy.

**Keywords:** Kernel Function, PSBB, Sentiment analysis, Support Verctor Machine, Twitter

## PENDAHULUAN

Internet saat ini sudah menjadi kebutuhan bagi masyarakat terutama di Indonesia dimana penggunaan internet setiap tahunnya terus bertambah. Sebuah *survey* yang dilakukan oleh Asosiasi Penyelenggaraan Jasa Internet Indonesia (APJII) menunjukkan ada peningkatan 10,12% penggunaan internet pada tahun 2019 dibandingkan dengan tahun 2018. Pengguna internet yang sudah masif ini turut bertambah dengan penggunaan sosial media, dimana sosial media ini menjadi wadah masyarakat untuk saling berinteraksi satu sama lain tanpa batas waktu maupun tempat. Sosial media yang populer dikalangan masyarakat salah satunya adalah Twitter. Menurut data yang dimiliki PT. Bakrie Telkom, Twitter yang merupakan situs *microblogging* populer ini memiliki pengguna di Indonesia pada tahun 2013, terdapat 19,5 juta dari 500 juta pengguna secara global.

Twitter menjadi populer karena dirasa lebih simpel dalam hal menyampaikan opini atau pendapat terhadap produk, isu sosial, hingga kebijakan pemerintah yang menjadi topik hangat dalam bentuk teks singkat yang disebut dengan *tweet*. Kebijakan yang sempat ramai diperbincangan di Twitter salah satunya

adalah kebijakan PSBB. Kebijakan PSBB ini merupakan kebijakan pemerintah yang bertujuan untuk menekan angka penyebaran kasus Covid-19 di tanah air. Hal ini disebabkan karena pertumbuhan angka kasus infeksi virus semakin hari semakin meningkat.

Data yang diperoleh dari website gugus tugas percepatan penanggulangan Covid-19, per tanggal 19 Mei 2020 jumlah kasus terinfeksi covid-19 sebesar 43,803 kasus dengan penambahan kasus perharinya 1.041 kasus [1]. Kebijakan PSBB menuai pro kontra di media sosial Twitter dan menjadi topik yang hangat untuk di bahas dan komentar-komentar dari masyarakat ini pun memiliki sentimen positif maupun negatif terhadap kebijakan ini yang menjadi menarik untuk dilakukan analisa sentimen pengguna terhadap kebijakan PSBB ini.

Analisa sentimen dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seseorang [2]. Analisa sentimen diimplementasikan pada berbagai situs *review* pengguna layanan jasa hotel [3], *review* pengunjung lokasi wisata [4], maupun pada situs *microblogging* Twitter [5], bertujuan untuk mengetahui tingkat kepuasan konsumen, *tweet* yang mengarah ke *cyber bullying* [6], sentimen masyarakat terhadap

calon Gubernur [7], maupun sentimen terhadap penerapan *social distancing* dalam masa pandemi [8]. Secara umum, penelitian terkait analisa sentimen dapat dibedakan berdasarkan metode pelabelan *dataset*, metode klasifikasi yang digunakan, serta jumlah kelas yang didefinisikan.

Berdasarkan metode klasifikasinya, terdapat tiga pendekatan utama dalam analisa sentimen, yaitu pendekatan *machine learning*, pendekatan *lexicon-based*, dan *hybrid* antar keduanya. Pendekatan *lexicon-based* sendiri dibagi ke dalam 2 kelompok besar, yaitu pendekatan *dictionary-based* (didasarkan pada kata-kata dalam kamus, misalnya WordNet, SenticNet, dan lain-lain) dan pendekatan *corpus-based* (menggunakan data *corpus* data yang dibangun, yang dibagi lagi ke dalam pendekatan *statistical* dan *semantic*) [9]. Algoritma *machine learning* yang umumnya digunakan dalam analisa sentimen dengan pendekatan *machine learning* adalah *Naive-Bayes classifier* [10], SVM [8], LSTM [11], maupun *ensemble classifier* [4].

T. Ahn Le, dkk [11] menggunakan 4 ribu *tweet* yang dilabeli manual sebagai *dataset*, terdiri dari *tweet* yang mengandung banyak kata-kata singkatan atau istilah tidak formal dalam bahasa Indonesia. Pada penelitian tersebut didefinisikan 3 kelas (positif, negatif, dan netral), menggunakan algoritma *Long Short Term Memory* (LSTM) sebagai *classifier* dan memperoleh tingkat akurasi sebesar 73,2%. U. Khaira, dkk [6] juga melakukan pelabelan secara manual dan

mendefinisikan 3 kelas, sementara klasifikasi dilakukan dengan menggunakan *lexicon-based* dan SentiStrength sebagai pendekatannya. Dari *dataset* berupa 454 *tweet*, diperoleh bahwa 161 *tweet* tidak terindikasi *bullying* (35.4%), 87 *tweet* netral (19.1%), dan 206 *tweet* terindikasi *bullying* (45.4%). Model klasifikasi dalam penelitian ini memiliki tingkat akurasi sebesar 60.5%. Tingkat akurasi yang rendah kemungkinan diperoleh karena menggunakan *dictionary* yang tidak sesuai dengan karakteristik *dataset* yang dimiliki.

Sementara itu C. Shofiya dan S. Abidi [8] menggunakan pendekatan lainnya dalam melakukan analisa sentimen, yaitu pendekatan *machine learning*. Penelitian tersebut menggunakan total 629 *tweet* sebagai *dataset*, dengan persentase *tweet* netral, negatif, dan positif berturut-turut adalah 40%, 35%, dan 25%.

Proporsi data latih dan data uji yang diterapkan adalah 80% berbanding 20%, sementara tingkat akurasi dari klasifikasi dengan 3 kelas tersebut adalah 71% menggunakan kernel RBF. C. Shofiya dan S. Abidi mengubah jumlah kelas menjadi 2 dan memperoleh peningkatan akurasi menjadi 81% jika menggunakan kernel Linier, namun tidak meningkat jika menggunakan kernel RBF. Di sisi lain, Murni, dkk [4] menggunakan pendekatan *hybrid* dalam melakukan analisa sentiment terhadap review pengunjung tempat wisata. Secara singkat, pendekatan *hybrid* tersebut menggunakan metode *lexicon-based* untuk melabelkan

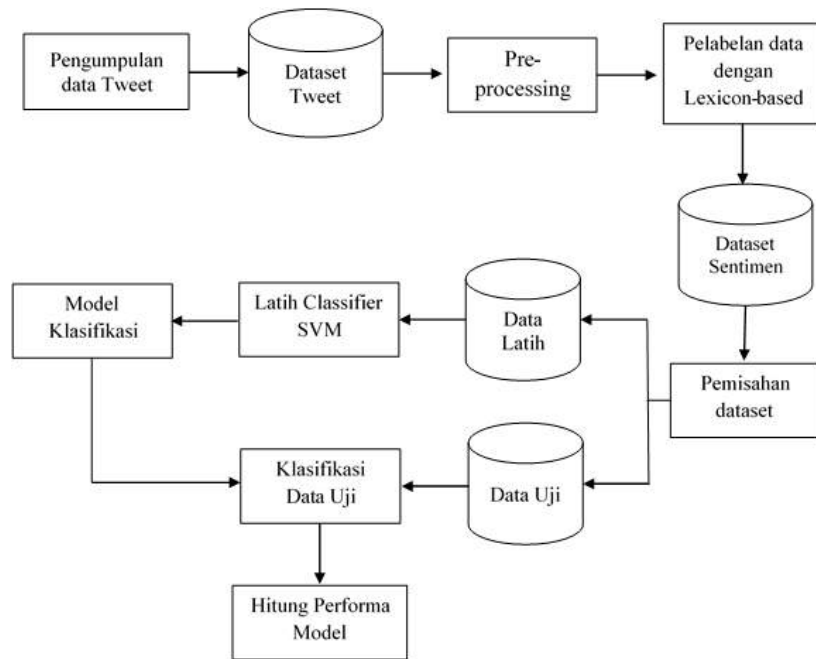
*dataset*, sehingga tidak perlu dilakukan secara manual, sementara metode *machine learning* digunakan sebagai *classifier*.

Analisa sentimen dinilai dapat alat bantu untuk mengetahui bagaimana sentimen masyarakat dalam opininya terhadap sebuah layanan, kebijakan, dan lain sebagainya. Oleh karena itu diperlukan proses otomatis untuk melakukan proses klasifikasi dan sentimen analisis setiap opini atau *dataset* yang diambil. Berdasarkan penelitian terdahulu, SVM merupakan algoritma *machine learning* yang sering digunakan dan memberikan hasil yang cukup mumpuni dalam riset analisa sentimen. SVM juga menyediakan pilihan yang banyak dalam pembentukan model klasifikasinya, terkait dengan fungsi kernel yang dapat digunakan untuk membangun modelnya.

Pemilihan fungsi kernel berbeda dapat meningkatkan akurasi [8] pada karakteristik data tertentu. Penelitian ini bertujuan untuk menerapkan algoritma SVM untuk menganalisa sentimen pengguna Twitter terhadap kebijakan PSBB dengan mempertimbangkan fungsi kernel yang digunakan dalam membangun model klasifikasinya.

## METODE PENELITIAN

Analisa sentimen dilakukan menggunakan *dataset* berupa *tweet* yang didapat dari sosial media sosial Twitter. *tweet* diambil berdasarkan kata kunci 'psbb' yang diambil pada tanggal 28 Mei, 16 September dan 27 September 2020. Data *tweet* mentah diperoleh menggunakan akses Twitter *API* dengan cara *crawling*. Setelah itu data mentah melalui tahap *pre-processing* seperti pemebersihan *tweet* dari simbol, angka, username, RT, hashtag, tanda baca, url dan lainnya yang dirasa mengganggu proses ekstraksi fitur dan klasifikasi *tweet*. Setelah proses itu, data akan dilabeli dengan positif dan negatif menggunakan *Lexicon Based*, kemudian data yang telah dilabeli tersebut akan menjadi *dataset* untuk diolah pada tahap *processing*. Pada tahap *processing* data diklasifikasikan menggunakan model algoritma *Support Vector Machine*, diberikan juga hasil dari pelatihan dan uji model algoritma. Pada tahap akhir merupakan penyajian hasil model algoritma klasifikasi yang divisualisasikan dalam bentuk tabel maupun diagram. Tahapan dalam melakukan analisa sentimen ini dapat dilihat pada Gambar 1.



**Gambar 1. Alur Proses Analisa Sentimen**

### **Pengumpulan Data**

Data diperoleh dengan menggunakan teknik *crawling data* pada situs [www.twitter.com](http://www.twitter.com). Penulis menggunakan fasilitas dari Twitter berupa Twitter API untuk melakukan request data. Sebelum melakukan pengumpulan data dengan *crawling* pada situs Twitter langkah pertama yang harus dilakukan adalah mendaftarkan akun menjadi pengembang pada situs <https://developer.twitter.com/en/apps> dan melakukan pendaftaran *Apps*. Adapun data diambil pada tanggal 28 Mei, 16 September, dan 27 September 2020.

Setelah melalui semua proses dan diterima oleh pihak Twitter selanjutnya dilakukan pembuatan *Apps* untuk mendapatkan *consumer API key*, *consumer API secret key*, *access token*, dan *access token secret*, dimana 4 hal itu dibutuhkan untuk

menjadi *credential* untuk melakukan *request* pada Twitter *API*. *Crawling* data dilakukan dengan menggunakan kata kunci ‘psbb’ pada tanggal 28 Mei, 16 September dan 27 September 2020. Setelah menjalankan naskah kode diatas diperoleh data sebanyak 24,704 dan kemudian disimpan kedalam file berformat csv.

### **Preprocessing Data**

*Preprocessing* data merupakan tahap untuk mengubah bentuk data menjadi lebih terstruktur sehingga dapat sesuai dengan kebutuhan untuk proses lebih lanjut. Secara garis besar tahapan *preprocessing* disajikan pada Gambar 2, dengan input berupa data *tweet*. Terdapat 4 proses secara terurut yang dilakukan pada tahap ini, yaitu *Case Folding*, *Cleaning*, *Tokenizing*, dan *Stopword Removal*. Berikut *Case folding* merupakan proses

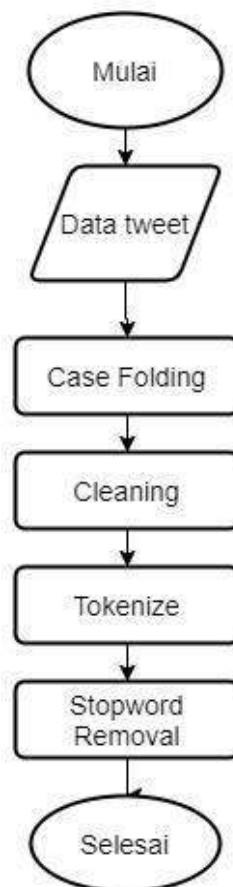
merubah huruf besar alfabet atau *uppercase* menjadi huruf kecil (*lowercase*), sementara *Cleaning* merupakan proses membersihkan data dari komponen- komponen yang tidak diperlukan dalam proses analisa sentimen. Hal yang dibersihkan pada proses ini adalah tautan URL, simbol RT, simbol @, digit, *whitespace*, *tags*, dan *hashtags*. Selanjutnya masuk ke proses *Tokenize*, merupakan penguraian kalimat menjadi token-token atau kata tunggal. Proses terakhir yaitu *Stopword Removal*, merupakan salah satu proses penting dalam melakukan analisa sentimen.

### Pembobotan Kata dan Pelabelan Data

Pembobotan kata adalah proses pemberian nilai terhadap setiap kata yang ada pada setiap opini yang sudah melewati proses data *preprocessing*. Pada penelitian ini digunakan metode TF-IDF dengan menggunakan persamaan (1) dan persamaan (2).

$$IDF(w) = \log\left(\frac{N}{DF(w)}\right) \quad (1)$$

$$TFIDF(w,d) = TF(w,d) \times IDF(w) \quad (2)$$



**Gambar 2.** Alur Tahapan *Preprocessing* Data

Proses pelabelan data pada penelitian ini dilakukan dengan menggunakan metode *Lexicon-based*. Prosedur untuk menentukan kelas sentimen dengan menggunakan *Lexicon-based* adalah dengan menghitung skor jumlah bobot kata positif ditambah skor jumlah bobot kata negatif dalam setiap kalimat.

Kalimat yang memiliki skor lebih besar dari nol diklasifikasikan ke dalam kelas positif, sedangkan kalimat yang memiliki skor sama dengan nol akan diklasifikasikan ke dalam kelas netral, sedangkan kalimat yang memiliki skor lebih kecil dari diklasifikasikan ke dalam kelas negatif. Bobot positif dan negatif diperoleh dari kamus *lexicon* yang diunduh dari situs <https://github.com/fajri91/InSet> [12].

### Tahap Pelatihan dan Pengujian Model Klasifikasi

Pada Model klasifikasi sentimen terlebih dahulu dilatih dengan data latih yang telah diberikan bobot dengan TF-IDF atau yang disebut tahap pelatihan, tahap pelatihan bertujuan untuk menemukan nilai  $a$ , nilai  $w$  dan nilai konstanta  $b$  sebagai parameter untuk menemukan *hyperplane* terbaik. Data latih yang telah diberikan bobot diubah menjadi bentuk vector dengan  $y \in (-1, 1)$ . Untuk mencari margin terbesar digunakan persamaan (3), dengan kendala yang dinyatakan oleh pertidaksamaan (4).

$$\min r(w) = \frac{1}{2} \|w\|^2 \quad (3)$$

$$y_i(w \cdot x_i + b) - 1 \geq 0, \forall i \quad (4)$$

Model matematis dari permasalahan di atas dapat diselesaikan dengan lebih mudah dengan mengubahnya menjadi fungsi Lagrangian dan disederhanakan menjadi persamaan (5), dengan variabel  $\bar{a}_i \geq 0$  sebagai Lagrange *multiplier*.

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i y_i (w^T \cdot x_i + b) + \sum_{i=1}^n a_i \quad (5)$$

Nilai optimum dari persamaan (5) dapat diperoleh dengan meminimumkan  $L$  terhadap  $w$  sekaligus terhadap  $b$  dan  $a$ , sehingga masalah Lagrange dapat dinyatakan dengan persamaan (6) berikut:

$$\min L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i y_i (w^T \cdot x_i + b) + \sum_{i=1}^n a_i \quad (6)$$

Selanjutnya akan diperoleh nilai  $\bar{a}_i$  yang digunakan untuk menemukan nilai bobot ( $w$ ). Terdapat nilai  $\bar{a}_i$  untuk setiap data latih dalam masalah klasifikasi. Nilai  $\bar{a}_i$  yang lebih besar dari merupakan *support vector* dan dapat digunakan untuk mencari *hyperplane* ( $b$ ). Untuk memisahkan data yang sulit dipisahkan secara linear maka digunakan fungsi kernel untuk melakukan transformasi data. Persamaan kernel yang digunakan contohnya adalah kernel linier, seperti pada persamaan (7) [13].



$$K(x_i, x_j) = (x_i^T x_j) \quad (7)$$

Pada penelitian ini digunakan 4 macam kernel, yaitu kernel Linear, RBF, Polinomial dan Sigmoid. Setelah pembentukan model klasifikasi, kemudian model diuji dengan data uji yang sudah diperoleh. Pengujian ini dilakukan untuk menghitung performa model klasifikasi yang dibuat dan melihat prediksi yang dilakukan oleh model klasifikasi terhadap data uji.

## HASIL DAN PEMBAHASAN

### Hasil Pengumpulan Data

Berdasarkan penjelasan sebelumnya, setelah data berhasil dikumpulkan dengan menggunakan teknik *crawling* data, kemudian data disimpan dalam bentuk csv. Dari proses pengumpulan data berhasil dikumpulkan

24,704 data *tweet*. Tabel 1 menyajikan beberapa contoh dari hasil pengumpulan data.

### Hasil *Preprocessing*

Setelah data dikumpulkan, data masuk ke dalam tahap *preprocessing* sebelum data tersebut dapat digunakan untuk melatih dan menguji model klasifikasi.

Pada tahap *preprocessing* ini, data melewati proses *casefolding*, *cleaning*, *tokenizing*, dan *Stopword Removal*. Tabel 2 menyajikan beberapa hasil dari tahap *preprocessing* data.

### Hasil Pelabelan Data

Setelah dilakukan *preprocessing* dan pembobotan kata pada setiap data *tweet* yang diperoleh, dilakukan pemberian label terhadap tiap data tersebut dengan menggunakan metode *Lexicon Based*.

**Tabel 1. Contoh Hasil Pengumpulan Data**

No.	<i>Tweet</i>
1	@Lyonskylight Sabar woee lagi psbb wkww
2	RT @indo35meme: wah psbb diperpanjang <a href="https://t.co/1hxzevkOtW">https://t.co/1hxzevkOtW</a>
3	Udah mau 3x psbb tidak ada perubahan. <a href="https://t.co/KwFAbOYQuH">https://t.co/KwFAbOYQuH</a>

**Tabel 2. Hasil Data *Preprocessing***

Data Sebelum Tahap <i>Preprocessing</i>	Data Sesudah Tahap <i>Preprocessing</i>
@Lyonskylight Sabar woee lagi psbbwkww	sabar woee psbb wkww
RT @indo35meme: wah psbb diperpanjang <a href="https://t.co/1hxzevkOtW">https://t.co/1hxzevkOtW</a>	psbb diperpanjang
Udah mau 3x psbb tidak ada perubahan. <a href="https://t.co/KwFAbOYQuH">https://t.co/KwFAbOYQuH</a>	udah psbb tidak perubahan

Tabel 3 menyajikan beberapa contoh hasil pelabelan data. Dari 24,704 data yang telah dilabeli dengan *Lexicon Based*, terdapat 11.357 data berlabel positif, 10.978 data berlabel negatif, dan 1.739 data berlabel netral. Pada penelitian ini, analisa sentimen dilakukan hanya pada kelas positif dan negatif, sehingga data berlabel netral dihapus dari *dataset*. Dengan demikian, jumlah data berlabel yang digunakan dalam tahap pelatihan dan pengujian berjumlah total 22.335 data.

### Hasil Analisa Sentimen

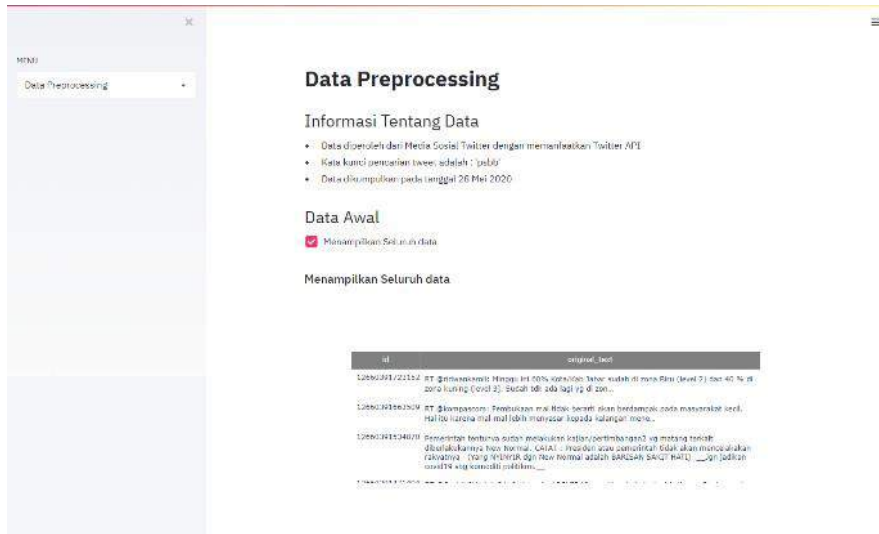
Hasil analisa sentimen pada penelitian ini disajikan ke dalam web interaktif yang memungkinkan *user* untuk memilih fungsi kernel yang digunakan dalam membangun model klasifikasi SVM. Secara garis besar, *website* yang dibuat memiliki 4 buah halaman, yaitu halaman Tentang, halaman Data *Preprocessing*, halaman

Klasifikasi Sentimen, dan halaman Hasil Klasifikasi. Halaman Tentang berfungsi untuk menampilkan judul dari penelitian, abstrak, dan data diri pembuat web ini. Pada halaman tentang ini hampir keseluruhan isinya berupa teks, namun terdapat juga checkbox untuk menampilkan data diri pembuat web.

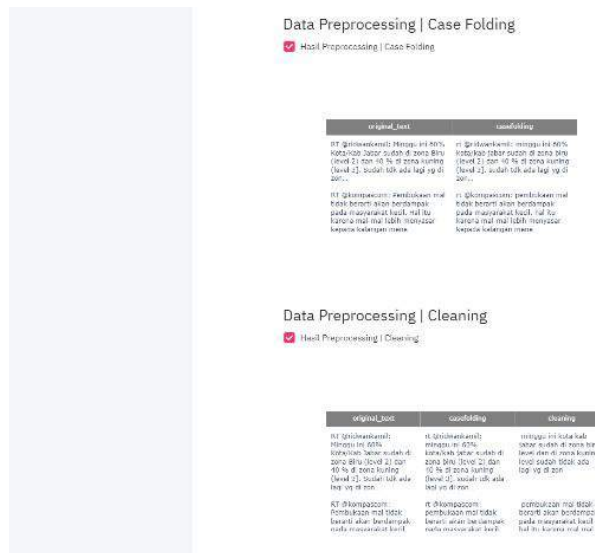
Halaman Data *Preprocessing* dirancang untuk menampilkan tahapan yang terjadi pada tahap *preprocessing* data. Halaman ini menampilkan data awal yang disajikan dalam bentuk tabel, berisi hasil dari pengumpulan data yang telah dilakukan. Pada halaman Data *Preprocessing* juga ditampilkan tabel data hasil dari proses *case-folding*, *cleaning*, *tokenizing*, maupun *Stopword Removal*. Tampilan halaman Data *Preprocessing* dapat dilihat pada Gambar 3, Gambar 4, dan Gambar 5 yang berturut-turut menampilkan data awal, data hasil proses *case-folding* dan *cleaning*, serta data hasil proses *tokenizing* dan *Stopword Removal*.

**Tabel 3. Contoh Hasil Pelabelan pada Data Penelitian**

Kode	Data	Sentimen
D1	sabar woe psbb wkwkw	1
D2	psbb diperpanjang	1
D3	udah psbb tidak perubahan	-1



Gambar 3. Tampilan Data Awal Pada Halaman Data *Preprocessing*



Gambar 4. Tampilan Data Hasil Proses *Case-Folding* dan *Cleaning* Pada Halaman Data *Preprocessing*

Pada halaman Klasifikasi Sentimen disajikan hasil evaluasi dari model klasifikasi yang dibangun. Selain itu ditampilkan tabel data yang digunakan untuk pelatihan dan pengujian model klasifikasi. Data yang ditampilkan merupakan data yang sudah dilabeli dengan metode *Lexicon Based*. Pada

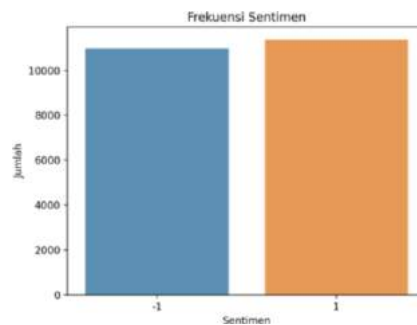
halaman ini terdapat juga histogram frekuensi sentimen dari basis data penelitian dan selectbox untuk memilih fungsi kernel untuk membangun model klasifikasi SVM yang diinginkan. Tampilan bagian awal dan contoh isi dari halaman Klasifikasi Sentimen dapat dilihat pada Gambar 6 dan Gambar 7.

The image shows two screenshots from a data preprocessing application. The top screenshot is titled "Data Preprocessing | Tokenizer" and displays a table with columns for "Original text", "Tokenized", "Lowercase", and "Cleaned". The bottom screenshot is titled "Data Preprocessing | Stopword Removal" and shows a similar table with an additional "Stopword Removed" column.

**Gambar 5. Tampilan Data Hasil Proses *Tokenizing* dan *Stopword Removal* pada Halaman *Data Preprocessing***

The image shows the initial view of a sentiment classification application. On the left is a sidebar with a search bar and a "Klasifikasi Sentimen" dropdown menu. The main area features a header "Klasifikasi Sentimen" with a sub-header "2019-nCoV" and an illustration of a person wearing a face mask. Below the header, there is a brief description of the algorithm used (K-Nearest Neighbor) and a "Data" section containing a table of input data.

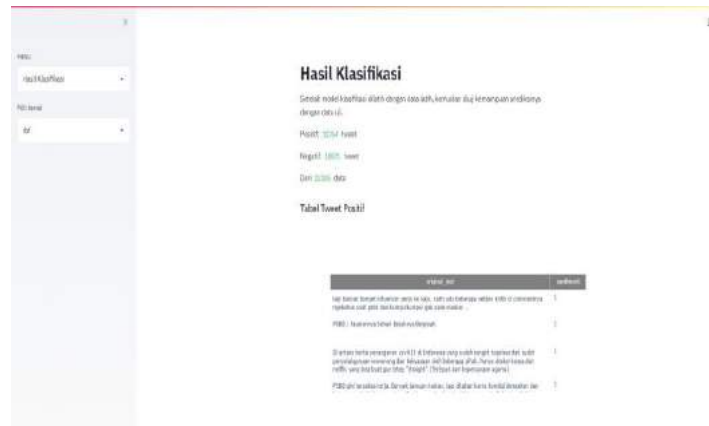
**Gambar 6. Tampilan Awal Halaman *Klasifikasi Sentimen***



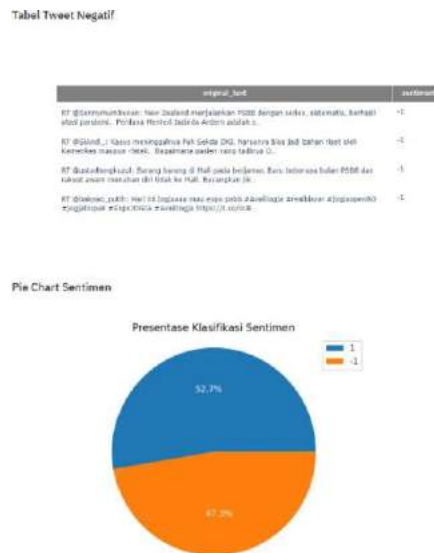
**Gambar 7. Histogram *Frekuensi Sentimen Basis Data* pada Halaman *Klasifikasi Sentimen***

Sementara itu, halaman Hasil Klasifikasi menampilkan hasil dari prediksi yang dilakukan oleh model klasifikasi SVM yang dibangun sesuai dengan kernel yang dipilih oleh *user*. Hasil klasifikasi ini disajikan dalam bentuk tabel dan *pie chart*. Tabel

menyajikan data uji yang termasuk ke dalam kelas positif maupun kelas negatif, sementara *pie chart* menyajikan persentase data uji yang masuk ke kelas positif dan negatif. Tampilan halaman Hasil Klasifikasi dapat dilihat pada Gambar 8 dan Gambar 9.



Gambar 8. Tampilan Bagian Awal dari Halaman Hasil Klasifikasi



Gambar 9. Bagian Akhir dari Halaman Hasil Klasifikasi

Tabel 4. Hasil Pengukuran Performa Model Klasifikasi Berdasarkan Fungsi Kernel

Parameter Performa	Linear	RBF	Polinomial	Sigmoid
Akurasi	94.93%	<b>95.94%</b>	94.55%	<b>93.01%</b>
Precision	<b>95.0%</b>	94.41%	90.59%	<b>93.06%</b>
Recall	95.03%	97.8%	<b>99.62%</b>	<b>93.20%</b>
F1-score	<b>95.01%</b>	<b>96.08%</b>	<b>94.89%</b>	<b>93.13%</b>

### Performa Model Klasifikasi

Performa model klasifikasi meliputi empat buah parameter, yaitu akurasi, precision, recall dan f1-score. Pengukuran performa model didasarkan pada confusion

matrix dan dengan menggunakan data berjumlah 22.335. Pengukuran performa dilakukan terhadap 4 buah model SVM yang dibangun dengan 4 fungsi kernel berbeda. Tabel 4 menyajikan hasil pengukuran

performa model SVM berdasarkan fungsi kernelnya. Performa yang disajikan merupakan rata-rata dari hasil perhitungan tiap iterasi dengan metode *k-fold cross validation*, dengan nilai *k* yang dipilih adalah 5.

Setelah dilakukan pengukuran performa sebanyak 5 kali dengan *5-cross validation* diperoleh bahwa model klasifikasi SVM dengan kernel RBF merupakan model klasifikasi yang memberikan performa keseluruhan terbaik dengan *dataset* pada penelitian ini, dengan nilai akurasi, *precision*, *recall* dan *f1-score* secara berturut-turut adalah 95.94%, 94.41 %, 97.8%, dan 96.08%. Model klasifikasi ini memprediksi terdapat sentimen positif sebanyak 11.764 *tweet* dan sentimen negatif sebanyak 10.571 *tweet*, seperti dapat dilihat pada *pie chart* dalam Gambar 11. Hasil tersebut menunjukkan bahwa pengguna Twitter cenderung memiliki sentimen positif terhadap kebijakan PSBB menurut model klasifikasi sentimen yang dibuat pada penelitian ini.

## KESIMPULAN DAN SARAN

Berdasarkan hasil pengukuran performa model klasifikasi SVM dengan kernel RBF mendapatkan hasil keseluruhan yang paling baik dibanding model dengan kernel lainnya. Nilai akurasi, *precision*, *recall*, dan *f1-score* yang dimiliki oleh model tersebut secara berturut-turut adalah 95.94%, 94.41 %, 97.8%, dan 96.08%. Model

klasifikasi memprediksi terdapat sentimen positif sebanyak 11.764 *tweet* (52.7%) dan sentimen negatif sebanyak 10.571 *tweet* (47.3%). Dari hasil ini disimpulkan bahwa dari 22.335 *tweet* pengguna Twitter yang diambil pada tanggal 28 Mei, 16 September dan 27 September 2020, cenderung bersentimen positif terhadap kebijakan PSBB di Indonesia. Adapun saran untuk penelitian selanjutnya diantaranya adalah melakukan tahap *preprocessing* data yang lebih banyak dan menggunakan data selain dari sosial media Twitter. Penggunaan algoritma klasifikasi lain juga disarankan untuk pengembangan penelitian selanjutnya.

## DAFTAR PUSTAKA

- [1] Gugus Tugas Percepatan Penanganan Covid 19, "Peta Sebaran", 2020, <https://covid19.go.id/peta-sebaran>, diakses 19 Juni 2020.
- [2] B. Pang dan L. Lee, "Opinion Mining and Sentiment Analysis", *Foundation and Trends Information Retrieval*, Vol. 2 (No. 1-2), hal. 1-135, 2008.
- [3] G. Markopoulos, G. Mikros, A. Iliadi, dan M. Lontos, "Sentiment Analysis of Hotel Reviews in Greek: A Comparison of Unigram Features", *Springer Proceedings in Business and Economics*, 2015, hal. 373-382.
- [4] Murni, T. Handhika, A. Fahrurozi, I. Sari, D.P. Lestari, "Hybrid Method for Sentiment Analysis Using

- Homogeneous Ensemble *Classifier*”, 2nd International Conference of Computer and Informatics Engineering (IC2IE), 2019, hal. 232-236.
- [5] O.Y. Adwan, M. Al-Tawil, A.M. Huneiti, R.A. Shahin, A. Zayed, R.H. Al-Dibsi, "Twitter Sentiment Analysis Approaches: A Survey", *iJET*, Vol. 15, No. 15, hal. 79-92, 2020.
- [6] U. Khaira, R. Johanda, P. Utomo, dan T. Suratno, “Sentiment Analysis of Cyberbullying on Twitter Using SentiStrength”, *Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)*, Vol. 3, No.1, hal. 21-27, 2020.
- [7] G.A. Buntoro, “Sentiment Analysis to Prediction DKI Jakarta Governor 2017 on Indonesian Twitter”, *International Journal Of Science, Engineering, And Information Technology*, Vol. 2, No. 1, hal. 36-40, 2017.
- [8] C. Shofiya dan S. Abidi, “Sentiment Analysis on COVID-19-Related Social Distancing in Canada Using Twitter Data”, *Int. J. Environ. Res. Public Health*, 18, 5993, 2021.
- [9] A. Sadia, F. Khan, dan F. Bashir, “An Overview of *Lexicon-Based* Approach For Sentiment Analysis”, 2018 3rd International Electrical Engineering Conference (IEEC 2018), 2018, hal. 1-6.
- [10] Y. Cahyono dan Saprudin, “Analisis Sentiment Tweets Berbahasa Sunda Menggunakan Naive Bayes *Classifier* dengan Seleksi Feature Chi Squared Statistic”, *Jurnal Informatika Universitas Pamulang*, Vol. 4, No. 3, hal. 87-94, 2019.
- [11] T. Anh Le, D. Moeljadi, Y. Miura, dan T. Ohkuma, “Sentiment Analysis for Low Resource Languages: A Study on Informal Indonesian Tweets”, *Proceedings of the 12th Workshop on Asian Language Resources*, 2016, hal. 123–131.
- [12] F. Koto dan G. Y. Rahmanytyas, “InSet *Lexicon*: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs”, 2017 International Conference on Asian Language Processing (IALP), hal. 391-394, 2017.
- [13] S. S. Chaeikar, “PFW: Polygonal Fuzzy Weighted—An SVM Kernel for the Classification of Overlapping Data Groups,” *Electronics*, hal. 1-14, 2020.

# EKSTRAKSI FITUR PEMBULUH DARAH CITRA FUNDUS RETINA MENGGUNAKAN FUZZY LOGIC

<sup>1</sup>Asyaroh Ramadona Nilawati, <sup>2</sup>Taufik Hidayat

<sup>1,2</sup>Fakultas Ilmu Komputer dan Sistem Informasi Universitas Gunadarma

<sup>1</sup>Jl. Margonda Raya 100, Depok 16424, Jawa Barat

<sup>1,2</sup>{rama,taufikhidayat}@staff.gunadarma.ac.id

## Abstrak

Ekstraksi pola pembuluh darah retina dapat dimanfaatkan dalam sistem biometrik sebagai otentikasi keamanan. Citra hasil ekstraksi pola pembuluh darah retina dapat dimasukkan ke dalam fitur untuk identifikasi sistem biometrik. Salah satu metode yang dapat dilakukan untuk melakukan segmentasi pembuluh darah retina adalah metode fuzzy logic. Pada penelitian ini, dilakukan ekstraksi pembuluh darah citra fundus retina menggunakan implementasi fuzzy logic. Peneliti menggunakan sejumlah 120 citra fundus yang diperoleh dari dataset DRIVE berformat .tif. Proses segmentasi dimulai dengan tahap preprocessing yang berisikan konversi citra menjadi grayscale, median filtering, perataan histogram CLAHE, dan eliminasi optic disc, kemudian dilanjutkan dengan pembuatan fuzzy inference system. Tahapan preprocessing yang digunakan merupakan hasil dari rangkaian uji coba peneliti dengan melihat hasil dari setiap uji coba yang dilakukan, sehingga mendapatkan citra yang menonjolkan fitur pembuluh darah dan menghilangkan noise atau fitur retina yang tidak diperlukan seperti optic disc. Uji coba segmentasi dilakukan pada Polyspace R2020a sebagai media untuk menjalankan program mulai dari preprocessing hingga segmentasi menggunakan fuzzy logic. Keluaran dari segmentasi ini berupa citra segmentasi hasil dari metode fuzzy logic dan crisp value. Hasil uji coba penelitian implementasi metode fuzzy logic berhasil melakukan ekstraksi pembuluh darah retina dan menghasilkan crisp value. Nilai crisp value ini dapat digunakan sebagai salah satu fitur dalam membentuk sistem identifikasi biometrik retina.

**Kata Kunci:** Crisp Value, Ekstraksi, Fuzzy Logic, Pembuluh Darah, Retina

## Abstract

Retinal vein pattern extraction can be utilized in biometric systems as security authentication. The image extracted from the retinal blood vessel pattern can be inserted into the feature to determine the biometric system. One method that can be used to segment retinal blood vessels is the fuzzy logic method. In this study, the retinal fundus image was extracted using fuzzy logic implementation. Researchers used a number of 120 fundus images obtained from the .tif format DRIVE dataset. The segmentation process begins with the preprocessing stage which contains grayscale image conversion, median filtering, CLAHE histogram alignment, and optical disc elimination, then proceeds with the creation of a fuzzy inference system. The preprocessing stage used is the result of a series of experiments by researchers by looking at the results of each trial carried out, so as to obtain images that highlight blood vessel features and eliminate noise or unnecessary retinal features such as optical discs. Segmentation trials were carried out on Polyspace R2020a as a medium for running programs ranging from preprocessing to segmentation using fuzzy logic. The output of this segmentation is a segmented image resulting from fuzzy logic and crisp value methods. The trial results of the implementation of the fuzzy logic method have succeeded in extracting retinal blood vessels and producing sharp values. This crisp value can be used as a feature in forming a retinal biometric system.

**Keywords:** Blood Vessels, Crisp Value, Extraction, Fuzzy Logic, Retina



## PENDAHULUAN

Pada saat ini, identifikasi individu adalah hal yang penting semenjak banyak sistem yang memerlukan pengguna yang sah untuk mengakses kontrol [1], khususnya untuk sistem yang menyimpan *data* dan dokumen berharga. Salah satu metode teknologi identifikasi yang saat ini sedang dikembangkan adalah *biometric feature-based identification technology* [2]. Sistem identifikasi biometrik adalah sistem yang melakukan identifikasi dan pengenalan menggunakan pola karakteristik biometrik [3] yang dimiliki sendiri. Penelitian sebelumnya pada biometrik menyatakan bahwa biometrik bisa digunakan untuk meningkatkan keamanan, kenyamanan, dan akuntabilitas sebuah sistem selama proses deteksi dan mencegah penipuan [4]. Menentukan fitur yang tepat menjadi sangat penting untuk menghasilkan sistem biometrik yang bisa mengidentifikasi individu secara benar.

Proses identifikasi biometrik salah satunya dapat dilakukan dengan menggunakan salah satu organ pada manusia yaitu retina. Retina pada setiap individu memiliki keunikan tersendiri yang bisa dipakai sebagai fitur karakteristik, salah satunya adalah melihat pembuluh darah retina [5] yang memiliki bentuk yang membedakan dari individu yang satu dengan yang lain. Salah satu metode yang dapat digunakan untuk melakukan ekstraksi terhadap fitur pembuluh darah [6] adalah *fuzzy logic*. *Fuzzy logic*

adalah sebuah wujud logika banyak nilai yang dimana nilai benar suatu variabel bisa dalam bentuk bilangan *real* antara nol dan satu keduanya inklusif [7], digunakan untuk menangani konsep kebenaran parsial, dimana nilai kebenarannya mungkin berada pada rentang sepenuhnya benar atau sepenuhnya salah [8].

Penelitian ekstraksi pembuluh darah retina menggunakan *fuzzy logic* sudah dilakukan pada peneliti sebelumnya. Penelitian [9] menggunakan *fuzzy c-means* dan pendekatan neutrosodik, penelitian ini dilakukan terhadap *dataset* DRIVE dan DIARETDB1. Penelitian [10] melakukan segmentasi dengan menggunakan *fuzzy logic* pada *dataset* DRIVE, STARE, DIARETDB1, dan MESSIDOR. Penelitian [11] mengimplementasikan *Possibilistic Fuzzy c-means* (PCFM) dengan metode *cuckoo search*, penelitian ini dilakukan terhadap *dataset* DRIVE dan STARE. Penelitian [12] menggunakan metode *Fuzzy C Means* untuk melakukan segmentasinya, dalam penelitian ini dilakukan beberapa tahap *preprocessing* sebelum dilakukan segmentasi pada *dataset* DRIVE dan STARE. Penelitian [13] menerapkan metode *Fuzzy Mathematical Morphology (FMM) operator* untuk pemrosesan *pipeline* yang dilakukan pada *datasets* STARE dan DRIVE menggunakan morfologi *opening*.

Penelitian ini menggunakan citra fundus yang digunakan berasal dari *dataset* DRIVE sejumlah 120 citra dengan format

.tif. Citra yang sudah melalui tahapan *preprocessing* kemudian dilakukan konvolusi untuk mendapat gradien  $x$  dan  $y$  sebagai *input* dari *fuzzy inference system*. Pada pembentukan *Fuzzy Inferense System*, peneliti melakukan operasi *zero mean Gaussian membership function* pada *input*. Penelitian ini juga melakukan pembentukan *membership function output* dari *fuzzy inference system*, dan pembentukan *rules* untuk menentukan hasil citra *output*. Evaluasi *fuzzy inference system* dilakukan untuk menghasilkan *crisp value*. Hasil penelitian ini diharapkan dapat menghasilkan fitur ciri yang dapat digunakan sebagai bagian dari *input biometric feature-based identification technology*.

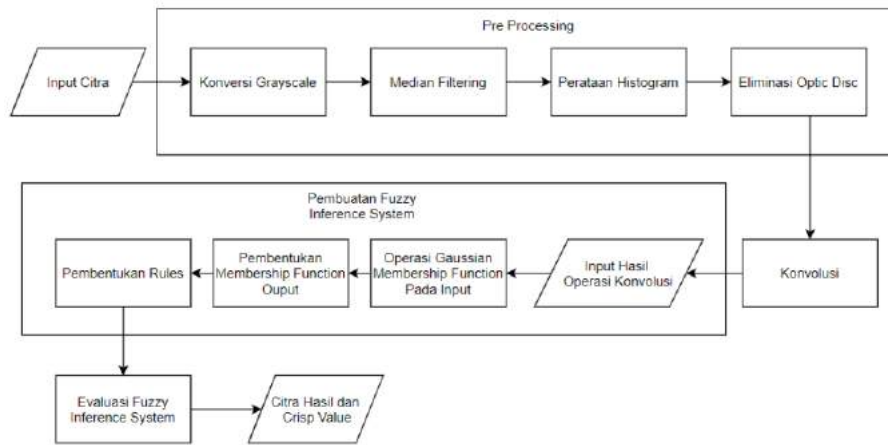
## METODE PENELITIAN

Pada penelitian ini dilakukan beberapa tahapan-tahapan untuk segmentasi pembuluh darah citra fundus retina seperti dapat dilihat pada Gambar 1. Tahapan pertama dilakukan dengan mengambil *input* citra yang berasal dari *dataset DRIVE* sejumlah 120 citra dengan ukuran 565 x 584 piksel dan mempunyai format .tif. Tahapan kedua yaitu melakukan *pre-processing* terhadap citra *input* dari *dataset DRIVE*, pada tahap ini ada beberapa sub proses yang dilakukan yaitu, pertama, lakukan konversi citra ke dalam citra

*grayscale*, kedua, operasikan median *filtering* ke citra *grayscale*, ketiga, lakukan perataan histogram menggunakan metode CLAHE, keempat, aplikasikan morfologi matematika untuk menghilangkan *optic disc* pada citra pembuluh darah retina. Tahapan ketiga yaitu lakukan konvolusi terhadap citra yang telah melalui tahap *preprocessing* untuk mendapat gradien  $x$  dan  $y$  sebagai *input* dari *fuzzy inference system*. Tahapan keempat yaitu pada pembuatan *Fuzzy Inferense System*, dalam tahap ini ada beberapa tahapan kecil yaitu memasukan *input* hasil operasi konvolusi, kemudian melakukan operasi *zero mean Gaussian membership function* pada *input*, selanjutnya pada pembentukan *membership function output* dari *fuzzy inference system*, dilakukan pembentukan *rules* untuk menentukan hasil gambar *output*. Tahap keenam yaitu lakukan evaluasi *fuzzy inference system* akan menghasilkan *crisp value*.

### A. Citra Masukan

*Input* citra yang digunakan pada penelitian adalah citra fundus pembuluh darah retina berwarna dari *dataset* yang bernama *DRIVE* [14]. Citra pembuluh darah retina merupakan citra berwarna dengan model warna RGB dengan ukuran piksel 565 x 584 seperti dapat dilihat pada Gambar 2.



**Gambar 1. Gambaran Umum Metode Penelitian**



**Gambar 2. Contoh Citra Fundus yang Digunakan [14]**

### B. Preprocessing Citra Fundus

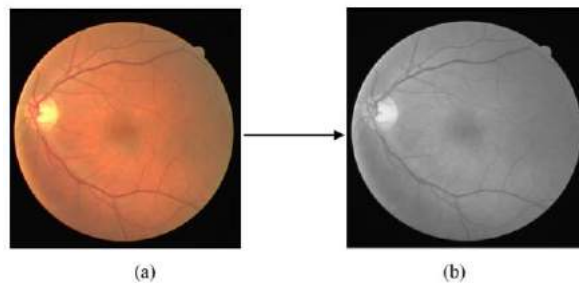
*Preprocessing* adalah tahapan awal pengolahan *Data* yang dilakukan terhadap citra pembuluh darah retina yang bertujuan untuk memperbaiki citra, melakukan eliminasi terhadap informasi yang tidak dibutuhkan pada tahap selanjutnya dan meningkatkan fitur citra yang akan diekstraksi. Preprocessing pada penelitian ini terdiri atas tahapan :

1. Melakukan konversi *grayscale* menggunakan persamaan 1.

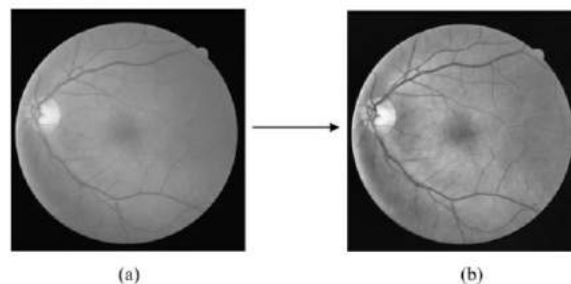
$$I = 0.33 * R + 0.5 * G + 0.166 * B \quad (1)$$

Dimana *output* dari citra *grayscale* adalah *I*, dan komponen merah, hijau, dan biru adalah *R*, *G*, dan *B* berurutan. Seperti yang dapat dilihat pada Gambar 3.

2. Melakukan operasi *filtering* yang digunakan untuk memperhalus citra retina menggunakan operasi median filter [15] yang dilakukan dengan mengganti piksel pada citra dengan nilai median dari tetangganya. Hasil dari operasi median filter ini digunakan sebagai input pada perataan histogram menggunakan *Contrast Limited Adaptive Histogram Equalization (CLAHE)* untuk menonjolkan citra pembuluh darah retina yang akan diekstraksi, sedemikian sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama seperti dapat dilihat pada Gambar 4.



Gambar 3. (a) Citra Fundus DRIVE (b) Citra *Grayscale*



Gambar 4. Citra Hasil CLAHE (a) Citra *Median Filter* (b) Citra CLAHE

Perataan histogram menggunakan CLAHE dilakukan dengan tujuan untuk mempertajam objek pembuluh darah sehingga dapat terlihat dengan jelas dibanding dengan citra median *filter* yang objek pembuluh darahnya belum terlihat jelas menggunakan persamaan 2.

$$\beta = \frac{M}{N} \left( 1 + \frac{\alpha}{100} (S_{max} - 1) \right) \quad (2)$$

Variabel M menyatakan luas *region size*, N menyatakan nilai grayscale (256), dan  $\alpha$  merupakan *clip factor* menyatakan penambahan batas limit suatu histogram yang bernilai antara 0 sampai dengan 100[16].

### C. Operasi *Filtering* dan *Morfologi Citra*

Operasi *filtering* dilakukan digunakan untuk memperhalus citra retina menggunakan operasi median filter [15] yang dilakukan

dengan mengganti piksel pada citra dengan nilai median dari tetangganya. Hasil dari operasi median filter ini digunakan sebagai input pada perataan histogram menggunakan *Contrast Limited Adaptive Histogram Equalization* (CLAHE) untuk menonjolkan citra pembuluh darah retina yang akan diekstraksi, sedemikian sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama seperti dapat dilihat pada Gambar 4.

Perataan histogram menggunakan CLAHE dilakukan dengan tujuan untuk mempertajam objek pembuluh darah sehingga dapat terlihat dengan jelas dibanding dengan citra median *filter* yang objek pembuluh darahnya belum terlihat jelas menggunakan persamaan 3.

$$\beta = \frac{M}{N} \left( 1 + \frac{\alpha}{100} (S_{max} - 1) \right) \quad (3)$$

Variabel  $M$  menyatakan luas *region size*,  $N$  menyatakan nilai grayscale (256), dan  $\alpha$  merupakan *clip factor* menyatakan penambahan batas limit suatu histogram yang bernilai antara 0 sampai dengan 100[16].

#### D. Eliminasi Optic Disc

Tahap ini dilakukan untuk menghilangkan *optic disc* pada citra karena yang akan kita pakai hanyalah pembuluh darahnya saja, maka dari itu pada tahap eliminasi *optic disc* dilakukan beberapa proses antara lain :

1. Melakukan operasi komplement citra grayscale menggunakan persamaan 4.

$$\overline{A_{i,j}^c} = 255 - A_{i,j} \quad (4)$$

2. Melakukan perataan histogram menggunakan *Adaptive Histogram Equalization* (AHE) untuk memperbaiki kontras citra.
3. Melakukan operasi morfologi matematika *opening* pada citra menggunakan *structuring element*nya 'disk' mengikuti bentuk dari citra fundus yang berbentuk bulat dengan ukuran 10x10 menggunakan persamaan 5.

$$\overline{f(x,y) \circ SE} = (f(x,y) \ominus SE) \oplus SE \quad (5)$$

Persamaan 6 merupakan persamaan *structuring element*.

$$\overline{SE = strel(p,q)} \quad (6)$$

Dimana  $p$  adalah *Structuring element's* type dan  $q$  merupakan parameter

4. Eliminasi *optic disc* dilakukan dengan cara mengurangi setiap nilai di dalam piksel citra hasil AHE dengan citra hasil morfologi matematika. Hasil eliminasi *optic disc* dapat dilihat pada Gambar 5.

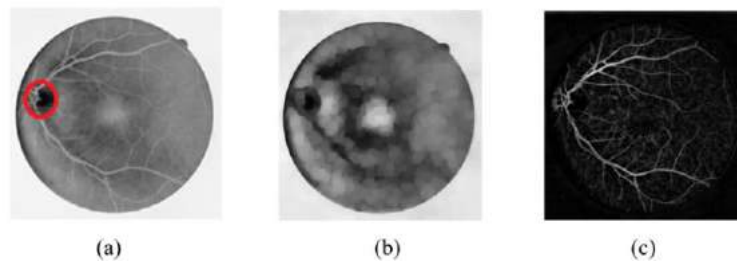
Seperti yang dapat dilihat pada gambar 5 (a) bagian citra yang memiliki fitur *optic disc* dilingkari merah, eliminasi *optic disc* dilakukan karena yang diperlukan hanyalah pembuluh darah untuk proses selanjutnya. Eliminasi *optic disc* dilakukan dengan cara mengurangi nilai pada citra piksel (a) dengan citra piksel (b) sehingga menghasilkan citra (c) yang tidak mempunyai *optic disc*.

#### E. Konvolusi Citra

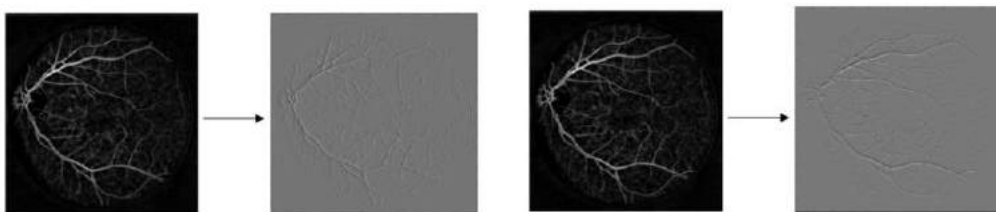
Konvolusi citra digunakan untuk mendeteksi tepi dengan menggunakan operator gradien pertama, konvolusi citra dilakukan untuk mendapatkan gradient  $x$  dan gradien  $y$ . Konvolusi citra dengan operator gradien pertama dilakukan dengan menggunakan *mask* konvolusi menggunakan persamaan 7 dan persamaan 8.

$$\overline{G_1(x) = [-1 \ 1]} \quad (7)$$

$$\overline{G_1(y) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}} \quad (8)$$



**Gambar 5. Citra Eliminasi *Optic Disc***  
**(a) Citra Komplemen (b) Citra Hasil Opening (c) Citra Tanpa *Optic disc***



**Gambar 6. Citra Hasil Deteksi Tepi. (a) Gradien *x* (b). Gradien *y***

Hasil dari deteksi tepi menggunakan operator gradien pertama pada gradien *x* dan *y* dapat dilihat pada Gambar 6. Konvolusi citra untuk mendapatkan gradien *x* dan *y* dilakukan agar kedua gradien tersebut menjadi *input* dari metode proses selanjutnya.

#### ***F. Pembuatan Fuzzy Inference System***

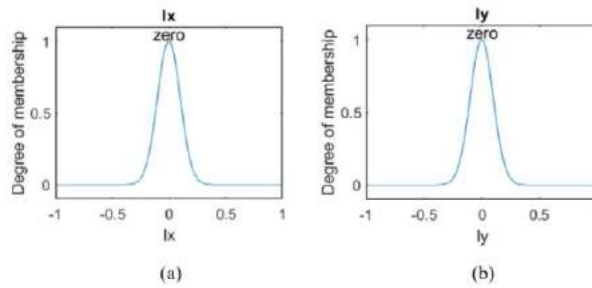
Pembuatan *fuzzy inference system* (FIS) dilakukan dengan menggunakan metode Mamdani dimana *rules* dibuat secara linguistik. Berikut adalah *pseudocode* pembuatan *fuzzy inference system mamdani*:

```
edgeFIS = mamfis('Name','edgeDetection');
```

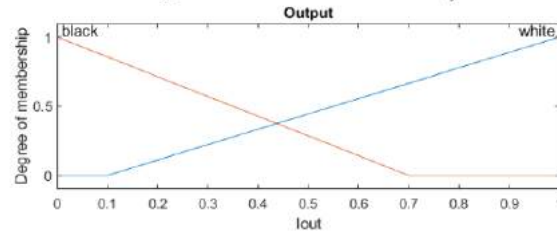
fungsi ‘*mamfis*’ adalah fungsi pada matlab yang berfungsi untuk membuat Mamdani *fuzzy inference system* dengan parameter

nama yaitu ‘*edgeDetection*’. Dalam pembuatan FIS ada beberapa tahap antara lain:

1. *Input* Hasil Operasi Konvolusi. Hasil operasi konvolusi gradien *x* dan *y* dijadikan sebagai *input* dari *Fuzzy Inference System* dengan normalisasi angka rentang dari minus satu (-1) sampai dengan satu (1)
2. *Membentuk Membership Function Input* dengan melakukan operasi zero-mean Gaussian *membership function* untuk setiap *input* dengan standar deviasi 0,1. Jika nilai gradien sebuah piksel adalah nol, maka dia termasuk ke dalam *zero membership function* dengan derajat satu. Grafik *membership function* pada *input* dapat dilihat pada Gambar 7



**Gambar 7. Grafik Membership Function Input (a). Grafik Membership Function Gradien X (b). Grafik Membership Function Gradien Y**



**Gambar 8. Grafik Membership Function Output**

3. Membentuk *Membership Function Output* menggunakan dua buah *output* yaitu white dan black, jenis *membership function* yang digunakan adalah *triangular membership function* dengan tiga nilai parameter A, B, dan C yang masing-masing adalah awal, puncak, dan akhir dengan ketentuan  $A > B > C$ . Parameter untuk setiap *output* yaitu :

- White      • Black
- A = 0,1      A = 0
- B = 1        B = 0
- C = 1        C = 0,7

Grafik *triangular membership function output* dapat dilihat pada Gambar 8. Seperti yang terlihat pada gambar 3.13. Parameter white dan black merepresentasikan berapa derajat keanggotaan *background* dan *foreground* dari masing-masing piksel *output* citra dari pembentukan *rules* yang akan

dibuat, grafik tersebut menunjukkan *Iout* (*fuzzy set*) ketika satu maka derajat keanggotaan ‘white’ satu dan ‘black’ nol yang nantinya pembentukan *membership function output* berguna untuk mengubah nilai membership function menjadi *fuzzy sets* sesuai dengan derajat keanggotaannya.

### G. Pembentukan Rules

Pembentukan *rules* dilakukan untuk membuat piksel menjadi putih jika termasuk ke dalam wilayah seragam dan hitam jika sebaliknya. Sebuah piksel adalah *background* jika gradien citranya nol di kedua arah. Jika salah satu gradien bukan nol, maka piksel tersebut adalah sebuah *foreground*. Rule ditentukan persamaan 9 dan persamaan 10.

$$I(x,y) = \text{white for } I_x = 0 \text{ and } I_y = 0 \text{ (indicating the background)}$$





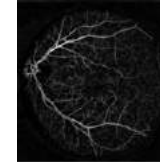

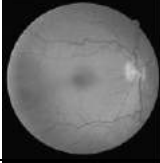
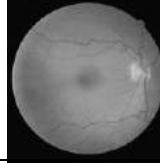

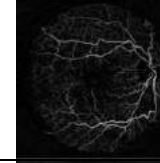

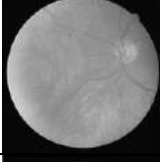
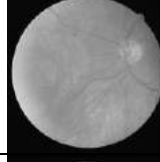
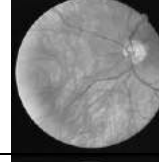
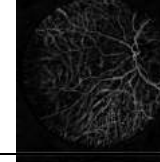




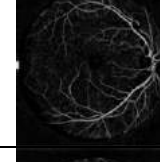

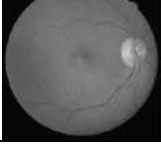
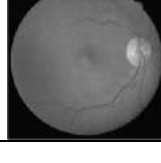


$I(x,y) = \text{black for } I_x \neq 0 \text{ or } I_y \neq 0; \text{ and } I(x,y) = \text{white for } I_x = 0 \text{ and } I_y = 0$

0)

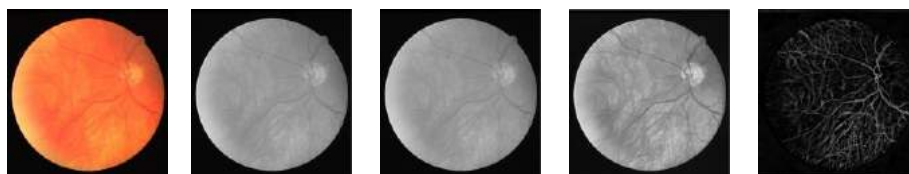
Pembentukan *rules* ini bertujuan untuk memberikan *output* citra dengan piksel  $\overline{I(x,y)}$  adalah white maka piksel tersebut akan berwarna putih dan merepresentasikan *background*, sedangkan jika  $\overline{I(x,y)}$  adalah black maka piksel tersebut akan berwarna hitam dan merepresentasikan *foreground* (objek pembuluh darah).

Uji coba dilakukan pada 120 citra fundus yang diperoleh dari *dataset* DRIVE. Pada penelitian ini, tahap *preprocessing* terdiri dari empat proses yaitu konversi ke citra *grayscale*, *median filtering*, perataan histogram (CLAHE), dan eliminasi *optic disc*. Keluaran dari proses *preprocessing* ini menghasilkan citra fundus tanpa *optic disc* yang sudah menunjukkan pola pembuluh darah.

**Tabel 1. Hasil *Preprocessing* Citra Fundus**

Nama File	Citra Asli	Citra Grayscale	Median Filter	CLAHE	Eliminasi Optic Disc
21_training.tif					
22_training.tif					
23_training.tif					
24_training.tif					
25_training.tif					





Gambar 9. Hasil Citra *Preprocessing* (a) Citra Fundus Asli (b) Citra *Grayscale* (c) Citra *Median Filter* (d) Citra *CLAHE* (e) Citra *Eliminasi Optic Disc*

Tabel 2. Citra Hasil Ekstraksi Fitur Ciri

Nama File	Citra Konvolusi Gradient X	Citra Konvolusi Gradient Y	Citra Fundus Tersegmentasi
21_training.tif			
22_training.tif			
23_training.tif			
24_training.tif			
25_training.tif			

Sebagai contoh diambil 5 hasil ujicoba dari total 120 citra pada dataset seperti dapat dilihat pada Tabel 1. Tabel 1 menunjukkan tahapan citra fundus dalam *preprocessing*

dimulai dari citra asli hingga citra yang sudah dilakukan eliminasi *optic disc*. Citra asli hingga tahap akhir memiliki ukuran dimensi yang sama yaitu 565 x 585 piksel. Dapat

diambil contoh citra hasil pada baris Tabel 1 gambar 23\_training.tif dapat dilihat pada Gambar 9.

Gambar 9 menunjukkan hasil dari metode *preprocessing* yang dilakukan. Tahap-tahap yang dilakukan adalah konversi citra warna menjadi grayscale, operasi median *filter* untuk mereduksi *noise* dan mempersiapkan citra untuk dilakukan operasi CLAHE, setelah itu menggunakan operasi pembukaan morfologi matematika untuk mengeliminasi *optic disc*.

**Hasil Pembentukan *Rules* dan Evaluasi *Fuzzy Inference System***

Hasil pembentukan *rules* dan evaluasi *fuzzy inference system* berupa matriks dua

dimensi yang merepresentasikan setiap piksel dengan *output crisp value* dan hasil segmentasi citra fundus.

Visualisasi *rules*, citra hasil ekstraksi seperti dapat dilihat pada Tabel 2. Tabel 3 menunjukkan nilai *crisp value* dari salah satu citra pada beberapa piksel. Nilai *crisp value* 0.7033 dan 0.6965 pada

Tabel 3 masing-masing menunjukkan bahwa dia adalah tepi objek atau *foreground* objek dan *background*.

Pada baris Tabel 3 yang diberi warna merah merupakan *output crisp value* yang menghasilkan citra terekstraksi hasil FIS Gambar 9, sebagai contoh nilai *crisp value* 0.7033 dan 0.6965 seperti terlihat pada Tabel 3.

**Tabel 3. Visualisasi *Output Crisp Value***

0.6965	0.6965	0.6965	0.6965	0.7033	0.6965	0.6965	0.7033	0.7033
0.7033	0.7033	0.7033	0.6965	0.6965	0.6965	0.6965	0.6965	0.6965
0.6965	0.6965	0.7033	0.7033	0.7033	0.6965	0.6965	0.6965	0.7033
0.7033	0.6965	0.6965	0.7033	0.7033	0.7033	0.7033	0.6965	0.6965
0.7033	0.7033	0.6965	0.7033	0.7033	0.7033	0.7033	0.7033	0.6965



**Gambar 9. Citra Hasil *Fuzzy Inference System***

## KESIMPULAN DAN SARAN

Salah satu cara melakukan pembentukan sistem identifikasi biometrik dengan menggunakan retina, dimana setiap individu memiliki keunikan retina tersendiri yang dapat digunakan fitur (ciri) unik dalam melakukan identifikasi. Pembuluh darah terekstraksi merupakan salah satu ciri yang dapat dimasukkan kedalam Sistem Biometrik Retina. Hasil ujicoba menunjukkan implementasi *fuzzy logic* pada citra fundus retina berhasil melakukan ekstraksi sehingga fitur pembuluh darah citra terekstraksi berhasil dilakukan. *Membership function input* juga berhasil dibuat menggunakan *zero-mean Gaussian membership function* yang menghasilkan grafik *gaussian membership function* yang digunakan untuk mendefuzifikasi *fuzzy set* menjadi nilai crisp (*Crisp Value*). *Crisp Value* hasil pembuluh darah retina terekstraksi ini merupakan salah satu fitur yang dapat digunakan dalam membentuk sistem biometrik retina.

Pengembangan penelitian lanjutan dapat dilakukan dengan menggunakan metode *preprocessing* yang lain agar fitur objek pembuluh darah lebih terlihat dengan jelas, penggunaan fitur unik lain pada objek citra retina untuk membentuk sistem biometrik yang handal.

## DAFTAR PUSTAKA

- [1] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," *IEEE security & privacy*, vol. 99, no. 2, pp. 33–42, 2003
- [2] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 4–20, 2004
- [3] S. K. Sahoo, T. Choubisa, and S. M. Prasanna, "Multimodal biometric person authentication: A review," *IETE Technical Review*, vol. 29, no. 1, pp. 54–75, 2012
- [4] K. A. Nixon, V. Aimale, and R. K. Rowe, "Spoof detection schemes," in *Handbook of biometrics*, pp. 403–423, Springer, 2008
- [5] Susetianingtias, D. T., Madenda, S., Fitriyaningsih, Adlina, D., Rodiah, & Arianty, R, "Retinal blood vessel extraction using wavelet decomposition", *International Journal of Advanced Computer Science and Applications*, 11(4), 351–355, <https://doi.org/10.14569/IJACSA.2020.0110448>, 2020
- [6] A. Budai, R. Bock, A. Maier, J. Hornegger, G. Michelson, "Robust Vessel Segmentation in Fundus Images", *International Journal of Biomedical Imaging*, 2013
- [7] V. Novák, I. Perfilieva, J. Močkoř, *Mathematical principles of fuzzy logic*,

- Dordrecht: Kluwer Academic. ISBN 978-0-7923-8595-0,1999.
- [8] F. Liu, “An efficient centroid type-reduction strategy for general type-2 fuzzy logic system”, *Information Sciences*,178.9, pp. 2224-2236, 2008.
- [9] I. Kaur, and L.M. Singh, “A Method of Disease Detection and Segmentation of Retinal Blood Vessels using Fuzzy C-Means and Neutrosophic Approach”, *Imperial journal of interdisciplinary research* 6, 551–557, 2016
- [10] C. Bhardwaj, S. Jain, and M. Sood, “Automatic Blood Vessel Extraction of Fundus Images Employing Fuzzy Approach”, *Indonesian Journal of Electrical Engineering and Informatics*, vol 7, No. 4, pp. 757–771, Dec 2019.
- [11] E. Emary, H.M. Zawbaa, A.E. Hassanien, G. Schaefer, and A.T. Azar, “Retinal Vessel Segmentation based on Possibilistic Fuzzy c-means Clustering Optimised with Cuckoo Search”, IEEE 2014 International Joint Conference on Neural Networks, Beijing, China, 2014, Volume: 1792 – 1796.
- [12] A.T. Franklin, and M. June, “Biological and Chemical Sciences GUI For Retinal Blood Vessel Articulation for Detection Of Diabetic”, *Research Journal of Pharmaceutical*, 8(2372), 2372–2385, 2017.
- [13] D.O. Dantas, D.D.S. Oliveira, H. Danilo, and P. Leal, “Blood vessels extraction using Fuzzy Mathematical Morphology”, IEEE International Conference on Acoustics, Speech and Signal Processing, 2017.
- [14] High-Resolution Fundus Image Databaset. Tersedia pada: <https://www5.cs.fau.de/research/data/fundus-images>. Tanggal akses: 1 Agustus 2020
- [15] E. Sutanty, S. Madenda, D.A. Rahayu, Rodiah, D.T. Susetianingtias, "Retinal blood vessel segmentation and bifurcation detection using combined filters", 3rd International Conference on Science in Information Technology (ICSITech), October 2017..
- [16] F. Kanditami, D. Saepudin, and A. Rizal, “Analisis Contrast Limited Adaptive Histogram Equalization ( Clahe ) Dan Region Growing Dalam Deteksi Gejala Kanker Payudara Pada Citra Mammogram”, *Jurnal Elektro Unika Atma Jaya*, 7 No.1, 15–28, 2014.
- [17] URL: High-Resolution Fundus Image Databaset. Tersedia pada: <https://www5.cs.fau.de/research/data/fundus-images>. Tanggal akses: 1 Agustus 2020.

# KLASIFIKASI TUMOR JINAK DAN TUMOR GANAS PADA CITRA MAMMOGRAM MENGGUNAKAN GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) DAN SUPPORT VECTOR MACHINE (SVM)

Lulu Mawaddah Wisudawati  
Fakultas Teknologi Industri Universitas Gunadarma  
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat  
lulu\_mawadah@staff.gunadarma.ac.id

## Abstrak

Kanker payudara merupakan penyebab utama kematian pada wanita. Data Global Cancer Observatory 2018 dari World Health Organization (WHO) menunjukkan kasus kanker yang paling banyak terjadi di Indonesia adalah kanker payudara, yakni 58.256 kasus atau 16.7% dari total 348.809 kasus kanker. Mamografi merupakan teknik yang paling umum digunakan dalam mendeteksi tumor payudara menggunakan sistem sinar-X dosis rendah. Ada beberapa tipe abnormalitas dalam citra mammogram, yaitu mikrokalsifikasi dan massa. Penelitian ini bertujuan untuk meningkatkan performa sistem Computer-Aided Diagnosis (CAD) dalam mengklasifikasi tumor jinak dan tumor ganas dengan mengembangkan metode ekstraksi fitur menggunakan Gray Level Co-Occurrence Matrix (GLCM) dan metode klasifikasi menggunakan Support Vector Machine (SVM). Uji coba dilakukan dengan menggunakan database DDSM dengan 256 citra abnormal (95 tumor jinak dan 161 tumor ganas) menghasilkan nilai akurasi sebesar 83.59% dengan nilai sensitivitas dan spesifisitas 87.58% dan 76.84%. Selain itu, didapatkan nilai AUC sebesar 0.98%. Metode tersebut menunjukkan bahwa sistem memberikan hasil performa yang baik dalam mengklasifikasi tumor jinak dan tumor ganas.

**Kata Kunci:** Tumor Payudara, Sistem CAD, GLCM, SVM, Mammogram

## Abstract

Breast cancer is a leading cause of death in women. Data Global Cancer Observatory 2018 from World Health Organization (WHO) shows that the most common cancer cases in Indonesia are breast cancer, which is 58,256 cases or 16.7% of the total 348,809 cancer cases. Mammography is the most commonly used technique in detecting breast tumors using a low-dose X-ray system. There are several types of abnormalities in mammogram images, namely microcalcifications and masses. This study aims to improve the performance of CAD systems in classifying benign and malignant tumors by developing a feature extraction method using Gray Level Co-Occurrence Matrix (GLCM) and a classification method using Support Vector Machine (SVM). The experiment was conducted using DDSM database with 256 abnormal images (95 benign and 161 malignant) obtained an accuracy value of 83.59% with sensitivity and specificity values of 87.58% and 76.84%, respectively. In addition, an AUC value of 0.98% was obtained. The method shows that the system gives better performance results in classifying benign and malignant breast tumors.

**Keywords:** Breast Tumors, CAD system, GLCM, SVM, Mammogram

## PENDAHULUAN

Kanker payudara merupakan suatu pertumbuhan jaringan payudara abnormal dengan pertumbuhan berlebihan dan tidak ada koordinasi dengan pertumbuhan jaringan normal, tumbuh *infiltratif*, dan destruktif serta dapat bermetastase dan tetap akan tumbuh dengan cara yang berlebihan. Kanker payudara merupakan penyebab utama kematian pada wanita. Setiap tahun lebih dari 250.000 kasus baru kanker payudara terdiagnosa di Eropa dan kurang lebih 175.000 di Amerika Serikat. Data Global Cancer Observatory 2018 dari World Health Organization (WHO) [1] menunjukkan kasus kanker yang paling banyak terjadi di Indonesia adalah kanker payudara, yakni 58.256 kasus atau 16.7% dari total 348.809 kasus kanker. Kementerian Kesehatan Republik Indonesia [2] menyatakan, angka kanker payudara di Indonesia mencapai 42,1 orang per 100 ribu penduduk. Upaya pencegahan dan pengendalian kanker payudara dapat dilakukan dengan cara deteksi dini. Teknologi pencitraan medis untuk pemeriksaan tumor pada payudara yaitu: *Mammography*, *MRI*, *Ultrasound (USG)* [3]. Mamografi merupakan teknik yang paling umum digunakan dalam mendeteksi tumor payudara menggunakan sistem sinar-X dosis rendah. Ada beberapa tipe abnormalitas dalam citra mammogram, yaitu keberadaan mikro-kalsifikasi (berbentuk seperti noda berukuran kecil dan terkadang berupa titik-titik) dan keberadaan massa. Massa adalah lesi dan

biasanya muncul pada mammogram sebagai daerah yang relatif padat. Karakteristik massa seperti bentuk, batas tepi, dan densitas pada citra mammogram dapat digunakan sebagai acuan untuk mengelompokkan ke dalam tumor jinak ataupun tumor ganas [4].

*Computer-Aided Diagnosis (CAD)* telah dikembangkan untuk meningkatkan hasil performa dalam deteksi keberadaan massa dan diagnosis tumor payudara. Dalam citra *grayscale*, tekstur mencerminkan variasi lokal dari nilai *gray-level* berupa kombinasi pada *smoothness*, kekasaran, dan keteraturan objek [3]. Fitur tekstur telah terbukti berguna dalam memberikan informasi tentang karakteristik citra. Ekstraksi fitur menggunakan *Gray Level Co-Occurrence Matrix (GLCM)* dilakukan oleh peneliti (Biswas, Nath, dan Roy) [5] dan fitur yang digunakan terdiri 4 fitur yaitu kontras, energi, korelasi, dan homogenitas. Hasil akurasi menggunakan metode *3NN (Neural Network)* dan *ANN (Artificial Neural Network)* sebesar 95% dan 75% (klasifikasi normal dan abnormal). Pada penelitian (Wisudawati et al.) [6] proses ekstraksi fitur menggunakan kombinasi metode *2D-Discrete Wavelet Transform* dan *GLCM* dengan 4 fitur yaitu kontras, korelasi, energi dan homogenitas. Hasil yang akurasi yang didapatkan dalam mengklasifikasi keberadaan massa pada citra mammogram adalah 100% dan mendapatkan akurasi 93.8% dalam mengklasifikasi normal, tumor jinak, dan tumor ganas. Peneliti (Wisudawati et al.) [7] kemudian juga mengembangkan metode

tersebut dalam mengklasifikasi tumor jinak dan tumor ganas menggunakan analisis tekstur menggunakan *GLCM* dan *Backpropagation Neural Network*. Hasil akurasi yang didapatkan yaitu 95.83% dengan sensitivitas 95.23% dan spesifisitas 96.49%. Peneliti (S. M. Salve) [8] pada penelitiannya yang berjudul “*Mammographic Image Classification using Gabor wavelet*” melakukan klasifikasi tumor jinak dan tumor ganas pada citra mammogram menggunakan *SVM (Support Vector Machine)*. Hasil dari *preprocessing* digunakan untuk ekstraksi fitur dengan 114 citra abnormal (63 tumor jinak dan 51 tumor ganas) menggunakan metode *Gabor Wavelet* serta *SVM* untuk klasifikasi. Hasil akurasi didapatkan sebesar 86% dengan menggunakan *Gabor Wavelet*. Pada penelitian tersebut hanya menggunakan sedikit dataset. Selain itu (Sharifah et al.) [9] dalam penelitiannya yang berjudul “*Cancer Detection Using Artificial Neural Network dan Support Vector Machine*” membandingkan *SVM* dan *ANN* menggunakan dataset yang berbeda. Hasil penelitian menunjukkan bahwa kedua metode menghasilkan performa yang baik tapi metode *SVM* masih lebih baik dibandingkan dengan *ANN*. Pada penelitian (Ankit Verma, Ankit Kumar dan Sanjeev Kumar) [10] juga memprediksi kanker payudara menggunakan

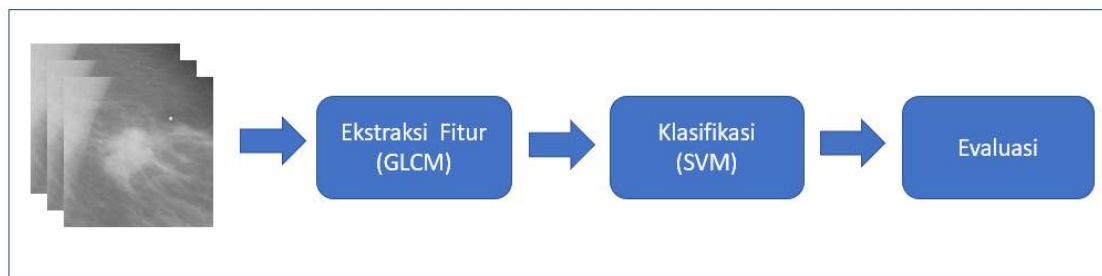
*Support Vector Machine* menggunakan database *Wisconsin Diagnosis Breast Cancer (WDBC)*. Hasil klasifikasi tumor jinak dan tumor ganas menggunakan *SVM* sebesar 96.07%. Hal tersebut menunjukkan bahwa *SVM* memberikan performa yang baik dalam proses klasifikasi tersebut.

Dengan melihat kelemahan dan kelebihan metode dari peneliti sebelumnya, maka pada penelitian ini mengusulkan metode ekstraksi fitur dengan metode *GLCM* menggunakan 4 fitur statistik yaitu kontras, korelasi, entropi dan homogenitas serta metode klasifikasi menggunakan *SVM* untuk meningkatkan performa dalam klasifikasi tumor jinak dan tumor ganas.

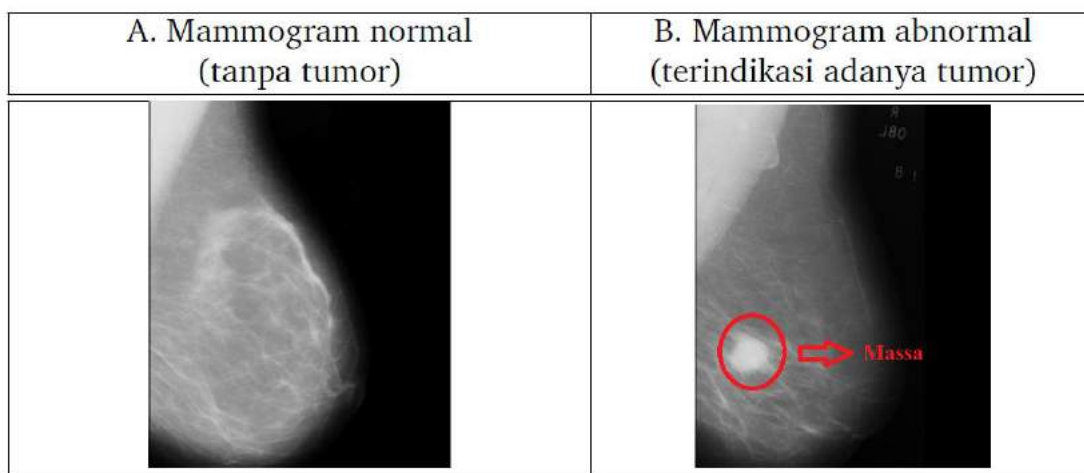
## **METODE PENELITIAN**

### **Gambaran Umum Sistem**

Gambaran umum sistem yang diajukan dapat dilihat pada Gambar 1 yang menunjukkan tahapan pengembangan sistem *CAD* meliputi input citra mammogram yang berupa daerah massa tumor jinak dan tumor ganas, ekstraksi fitur dilakukan dengan metode *GLCM*, klasifikasi (pelatihan dan pengujian) menggunakan *SVM* dan proses evaluasi untuk melihat performa sistem



**Gambar 1. Gambaran Umum Sistem**



**Gambar 2. Citra mammogram normal (A) dan abnormal (B)**  
(Sumber: publik database *Mammographic Image Analysis Society (MIAS)*)

### Citra Mammogram

Citra mammogram merupakan hasil dari mesin mamografi yang dapat digunakan untuk deteksi tumor payudara secara dini dan radiasi yang dipancarkan oleh mesin mamografi aman bagi tubuh, yaitu 0,7 mV. Citra mammogram menampilkan struktur jaringan payudara yang cukup kompleks. Keberadaan tumor pada payudara dapat dilihat dari pola tekstur citra mammogram. Tumor biasanya terdapat pada daerah dengan nilai intensitas yang lebih besar dari daerah sekitarnya dan juga dapat dilihat dari bentuk massa itu sendiri sehingga pada citra mammogram normal dan abnormal

memiliki ciri karakteristik tekstur berbeda seperti pada Gambar 2.

Pada citra mammogram normal (A) dan citra mammogram abnormal terdapat massa (terindikasi adanya tumor). Massa adalah lesi dan biasanya muncul pada mammogram sebagai daerah yang relatif padat. Gambar 3. menunjukkan area massa pada citra mammogram tumor jinak dan tumor ganas. Massa digambarkan oleh tiga fitur, yaitu bentuk atau kontur, batas tepi, dan densitas [11]. Pengelompokan tumor jinak dan tumor ganas pada karakteristik massa berdasarkan Breast Imaging-Reporting and Data System



(BI-RADS) descriptor [12] dapat dilihat pada Tabel 1.

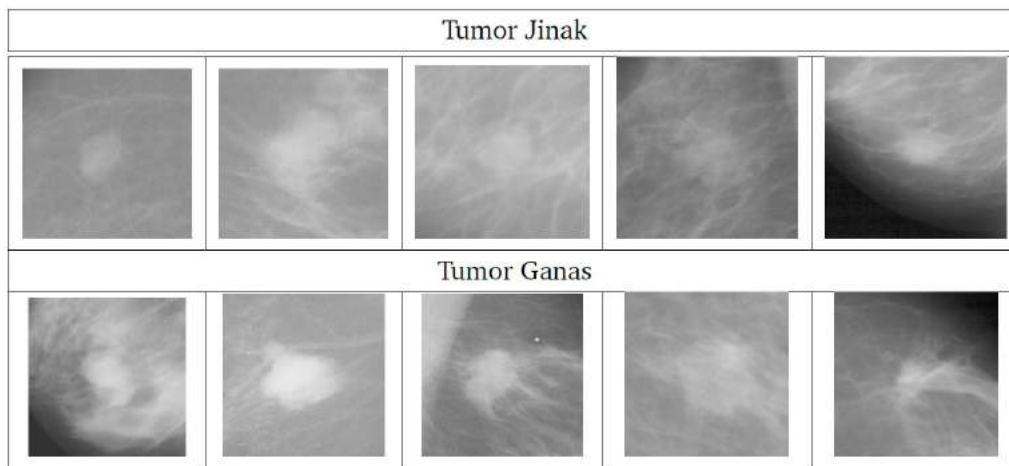
**Dataset**

Database yang digunakan dalam penelitian ini adalah citra mammogram dari publik database *DDSM (Digital Database for Screening Mammography)* dengan format *TIFF (Temporary Instruction File Format)* berjumlah 256 citra abnormal yang terdiri dari 95 tumor jinak dan 161 tumor ganas.

**GLCM (Gray Level Co-Occurrence Matrix)**

*GLCM* merupakan perhitungan tekstur pada orde kedua [13] dan merupakan matrik yang menggambarkan frekuensi munculnya pasangan piksel pada jarak  $d$  dan orientasi arah dengan sudut  $\theta$  dalam citra yang digunakan untuk menghitung fitur-fitur *GLCM*.

Jarak  $d$  yang digunakan adalah 1 yang dinyatakan dalam piksel, sementara untuk orientasi sudut dinyatakan dalam derajat dengan sudut  $0^\circ, 45^\circ, 90^\circ, \text{ dan } 135^\circ$ .



**Gambar 3. Massa Tumor jinak dan Tumor Ganas pada Citra Mammogram**  
[Sumber: publik database DDSM]

**Tabel 1. Pengelompokan Tumor Jinak dan Tumor Ganas pada Karakteristik Massa Berdasarkan BI-RADS Descriptor [12]**

Karakteristik	BI-RADS Descriptor	Fitur Memihak Jinak	Fitur Memihak Ganas
Massa	Bentuk	<i>Oval, round</i>	<i>Irregular, lobulated, architectural distortion</i>
	Batas Tepi	<i>Circumscribed</i>	<i>Microlobulated, spiculated, obscured, ill-defined</i>
	Densitas	<i>Fat, low, equal</i>	<i>High</i>

[Sumber: ACR American College of Radiology]

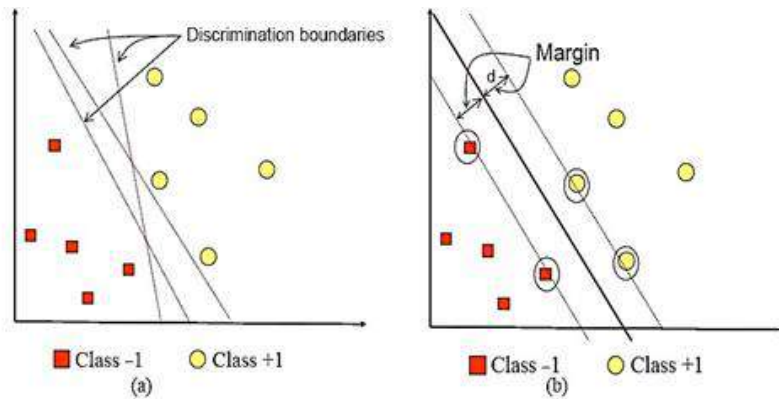
Pembentukan *GLCM* dilakukan dengan menentukan hubungan spasial antara piksel referensi dengan piksel tetangga, dengan sudut  $\theta$  dan jarak  $d$ . Dalam penelitian ini menggunakan sudut  $45^0$  dengan jarak 1 (offset, [-1 1]) dan Numlevel 32, dimana numlevel adalah kedalaman bit citra yang akan dianalisis. Penentuan awal matriks *GLCM* dilakukan dengan menghitung matriks kookurensi dan menjumlahkan matrik kookurensi dengan transposenya untuk menjadikannya simetris. Untuk menghilangkan ketergantungan pada ukuran citra, nilai-nilai elemen *GLCM* dinormalisasi. Perhitungan pengukuran statistika fitur-fitur *GLCM* yang digunakan dalam penelitian ini, antara lain [14]:

- **Kontras** merupakan ukuran keberadaan variasi aras keabuan piksel citra
- **Energi** merupakan ukuran homogenitas dari suatu citra
- **Korelasi** menyatakan ukuran ketergantungan linear derajat keabuan citra
- **Homogenitas** merupakan keseragaman intensitas keabuan pada citra.

#### ***SVM (Support Vector Machine)***

*SVM* adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan

menemukan *hyperplane* terbaik yang memisahkan dua buah class pada *input space*. Pada dasarnya *SVM* memiliki prinsip linear, akan tetapi pada saat ini *SVM* telah berkembang sehingga dapat menyelesaikan masalah *non-linear*. Cara kerja *SVM* pada masalah *non-linear* adalah dengan memasukkan konsep kernel pada ruang berdimensi tinggi. Pada ruang yang berdimensi ini, nantinya akan dicari pemisah atau yang sering disebut *hyperplane*. *Hyperplane* dapat memaksimalkan jarak atau *margin* antara kelas data. *Hyperplane* terbaik antara kedua kelas dapat ditemukan dengan mengukur *margin* dan kemudian mencari titik maksimalnya. Gambar 4a menunjukkan beberapa *pattern* yang merupakan anggota dari dua buah *class* yaitu positif (dinotasikan dengan +1) dan negatif (dinotasikan dengan -1). *Pattern* yang tergabung pada *class* negatif disimbolkan dengan kotak, sedangkan *pattern* pada *class* positif disimbolkan dengan lingkaran. Proses pembelajaran dalam problem klasifikasi dilakukan dengan menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Garis solid pada Gambar 4b menunjukkan *hyperplane* yang terbaik yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah support vector.



Gambar 4. SVM (*Support Vector Machine*)

Tabel 2. Hasil Ekstraksi Fitur Tumor Jinak dan Tumor Ganas Menggunakan Database DDSM

	Kontras	Korelasi	Energi	Homogenitas	Target
<b>Tumor Jinak</b>	0,147871	0,934014	0,26298	0,926064	0
	0,149028	0,997239	0,088731	0,926277	0
	0,14596	0,978987	0,116624	0,92702	0
	0,135729	0,973172	0,160773	0,932203	0
	0,18201	0,993342	0,072134	0,910461	0
<b>Tumor Ganas</b>	0,242303	0,96368	0,113848	0,879752	1
	0,258754	0,977753	0,08104	0,88562	1
	0,257806	0,979546	0,105777	0,885429	1
	0,279349	0,989364	0,057117	0,863161	1
	0,25209	0,825455	0,210785	0,874393	1

## HASIL DAN PEMBAHASAN

Pengembangan sistem *CAD* dalam klasifikasi tumor jinak dan tumor ganas dilakukan dengan menggunakan perangkat lunak MATLAB.

### Ekstraksi fitur menggunakan GLCM

Hasil dari ekstraksi fitur menggunakan GLCM untuk tumor jinak dan tumor ganas dapat dilihat pada Tabel 2. Fitur kontras pada tumor jinak mempunyai nilai yang lebih kecil dibandingkan dengan tumor ganas. Sedangkan fitur homogenitas pada tumor jinak lebih besar dibandingkan dengan tumor ganas. Target 0

menyatakan tumor jinak dan Target 1 menyatakan tumor ganas. Target tersebut akan digunakan dalam proses klasifikasi menggunakan *SVM*.

Hasil rata-rata dari nilai ekstraksi fitur dapat dilihat pada Tabel 3 dan dapat disimpulkan bahwa nilai rata-rata kontras pada tumor ganas lebih besar dibandingkan dengan tumor jinak. Hal tersebut menunjukkan bahwa ukuran keberadaan variasi atas keabuan piksel citra tumor jinak kecil. Pada fitur korelasi tumor jinak mempunyai nilai rata-rata lebih besar dibandingkan dengan tumor ganas. Hal tersebut menunjukkan bahwa pada citra tumor jinak mempunyai nilai yang kecil pada ukuran

konsentrasi pasangan dengan intensitas keabuan tertentu pada matriks. Pada fitur energi dan homogenitas tumor jinak mempunyai nilai rata-rata lebih besar dibandingkan dengan tumor ganas. Hal tersebut menunjukkan bahwa pada citra tumor jinak mempunyai nilai piksel yang mirip dengan piksel lainnya dan mempunyai keseragaman intensitas keabuan yang tinggi pada citra

### Klasifikasi Tumor Jinak dan Tumor Ganas menggunakan SVM

Proses klasifikasi dibagi menjadi dua yaitu (lihat Gambar 5):

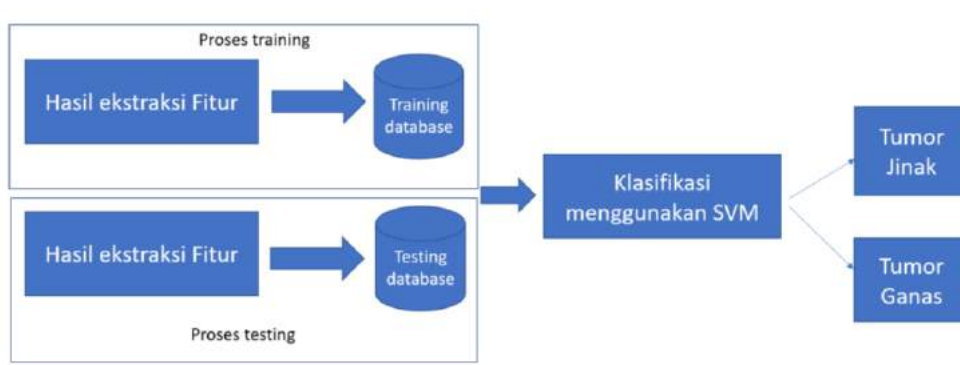
- Proses *training*: pada proses training digunakan training set yang telah diketahui label-labelnya untuk membangun model atau fungsi.

- Proses *testing*: dilakukan untuk mengetahui keakuratan model atau fungsi yang akan dibangun pada proses training. Data testing digunakan untuk memprediksi label-labelnya.

Hasil fitur-fitur yang telah didapatkan dalam proses ekstraksi fitur kemudian dimasukkan dalam proses klasifikasi tumor jinak dan tumor ganas menggunakan metode SVM. Metode, pelatihan, pengujian dan evaluasi menggunakan *10-fold cross validation* yang merupakan pilihan terbaik untuk mendapatkan hasil validasi yang akurat. Metode tersebut membagi dataset menjadi 10-buah partisi secara acak. Kemudian dilakukan 10 kali eksperimen, dimana masing-masing eksperimen menggunakan data partisi ke-10 sebagai *data testing* dan memanfaatkan sisa partisi lainnya sebagai *data training*.

**Tabel 3. Nilai rata-rata Fitur Kontras, Korelasi, Energi, Homogenitas pada Tumor Jinak dan Tumor Ganas**

	Nilai Rata-rata			
	Kontras	Korelasi	Energi	Homogenitas
<b>Tumor Jinak</b>	0,177	0,980	0,106	0,915
<b>Tumor Ganas</b>	0,259	0,969	0,098	0,883



**Gambar 5. Bagan Klasifikasi Tumor Jinak dan Tumor Ganas**

Hasil *Confusion Matrix* dapat dilihat pada Tabel 4. Performa sistem diukur berdasarkan akurasi, sensitivitas, spesifisitas dan *AUC (Area Under Curve)*. Nilai *True Positive* = 141, *False Positive* = 22, *False Negative* = 20 dan *True Negative* = 73. Hasil uji diagnostik dapat dilihat pada Tabel 5. Akurasi adalah ukuran seberapa dekat suatu hasil pengukuran dengan nilai yang benar atau diterima dari kuantitas besaran yang diukur. Hasil akurasi yang didapatkan sebesar 83.59 % yang menunjukkan sistem dapat mengklasifikasi tumor jinak dan tumor ganas sebesar 83.59%.

$$\text{Akurasi} = \frac{141+73}{141+73+20+22} \times 100\% = 83.59\%$$

Sensitivitas adalah ukuran keakuratan tes yaitu seberapa besar kemungkinan tes untuk mendeteksi positif orang-orang yang memiliki penyakit. Hasil sensitivitas menunjukkan sebesar 87.58% sistem dapat mendeteksi positif orang-orang yang memiliki penyakit.

$$\text{Sensitivitas} = \frac{141}{141+20} \times 100\% = 87.58\%$$

Spesifisitas adalah proporsi orang yang benar-benar tidak sakit dan tidak sakit pula saat diidentifikasi dengan tes skrining atau ukuran statistik mengenai akurasi tes, yaitu seberapa baik tes mengidentifikasi negatif orang-orang yang tidak memiliki penyakit. Hasil spesifisitas menunjukkan sebesar 76.84% sistem dapat mengidentifikasi negatif orang-orang yang tidak memiliki penyakit

$$\text{Spesifisitas} = \frac{73}{73+22} \times 100\% = 76.84\%$$

*Area under the curve (AUC)* memberikan gambaran tentang keseluruhan pengukuran atas kesesuaian dari model yang digunakan. Semakin besar *AUC* maka semakin baik variabel yang diteliti dalam memprediksi kejadian. Nilai *AUC* yang didapatkan sebesar 0.908. Mengacu pada klasifikasi akurasi nilai tersebut menunjukkan sistem dapat mendiagnosis tumor jinak dan tumor ganas dengan baik (*excellent classification*) [15].

**Tabel 4. Hasil *Confusion Matrix***

		Nilai Sebenarnya	
		Positives	Negatives
Jumlah data= 256	Positives	141	22
	Negatives	20	73
Nilai Prediksi	Positives	True Positive	False Positive
	Negatives	False Negative	True Negative

**Tabel 5. Hasil Uji Diagnostik**

Uji Diagnostik	Nilai Diagnostik
Akurasi	83.59
Sensitivitas	87.58%
Spesifisitas	76.84%
<i>Area under the curve (AUC)</i>	0.908

## KESIMPULAN DAN SARAN

Pendekatan metode yang digunakan dalam penelitian ini untuk meningkatkan performa sistem CAD berhasil dilakukan. Pendekatan ekstraksi fitur menggunakan metode GLCM dengan sudut  $45^0$  dan metode klasifikasi menggunakan SVM. Uji coba yang dilakukan dengan menggunakan database DDSM dengan 256 citra abnormal (95 tumor jinak dan 161 tumor ganas) menghasilkan nilai akurasi sebesar 83.59% dengan nilai sensitivitas dan spesifisitas 87.58% dan 76.84%. Selain itu, didapatkan nilai AUC sebesar 0.98%. Hal tersebut menunjukkan bahwa metode klasifikasi menggunakan GLCM dan SVM memberikan hasil performa yang baik dalam mengklasifikasi tumor jinak dan tumor ganas.

Tahapan selanjutnya, ujicoba dilakukan dengan menambah dataset pada tahapan pelatihan dan pengujian. Selain itu juga akan dilakukan proses *pre-processing* untuk menghilangkan derau pada citra mammogram. Pengembangan metode klasifikasi juga akan dilakukan dengan metode lainnya seperti *LibSVM*, *deep learning*, *K-Nearest Neighbor*.

## DAFTAR PUSTAKA

- [1] World Health Organization, "Cancer Today: Data and Methods". Global Cancer Observatory (GCO), 2018. [Online]. Available: <http://gco.iarc.fr/>. [Accessed: February 20, 2020]
- [2] Kementerian Kesehatan Republik Indonesia, "Hari Kanker Sedunia 2019", Jan 31, 2019. [Online]. Available: <https://www.kemkes.go.id/article/view/19020100003/hari-kanker-sedunia-2019.html>. [Accessed: February 20, 2020]
- [3] S. Madenda, *Pengolahan Citra dan Video Digital*. Penerbit Erlangga, 2015
- [4] L.M. Wisudawati, "Metode Identifikasi Tumor Jinak dan Tumor Ganas Pada Citra Mammogram Berdasarkan Pada Fitur Tekstur Massa Menggunakan Wavelet-GLCM dan Jaringan Saraf Tiruan", Disertasi, Universitas Gunadarma, 2020.
- [5] R. Biswas, A. Nath, and S. Roy, "Mammogram classification using gray-level co-occurrence matrix for diagnosis of breast cancer", International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), pages 161–166. 2016. DOI: 10.1109/ICMETE.2016.85
- [6] L. M. Wisudawati, S. Madenda, E. P. Wibowo, A. A. Abdullah. "Feature extraction optimization with combination 2D discrete wavelet transform and gray level co-occurrence matrix for classifying normal and abnormal breast tumors", Modern Applied Science 2020; vol. 14 no. 5, pp. 51-62. DOI:10.5539/mas.v14n5p51

- [7] L. M. Wisudawati, S. Madenda, E. P. Wibowo, A. A. Abdullah. "Benign and malignant breast tumors classification based on texture analysis and backpropagation neural network". *Computer Optics* 2020, vol. 45 no. 2, pp. 227- 234. DOI: 10.18287/2412-6179-CO-769
- [8] S. M. Salve, "Mammographic image classification using gabor wavelet". *International Research Journal of Engineering and Technology (IRJET)*, vol. 03 no. 03 Mar-2016, pp. 202-207.
- [9] H. S. Sharifah, R. Sallehuddin, and N. Azizah, "Cancer Detection Using Artificial Neural Network and Support Vector Machine: A Comparative Study," *Jurnal Teknologi*, vol. 65 no. 1, pp. 73–81, 2018.
- [10] DOI: 10.11113/jt.v65.1788
- [11] A. Verma, A. Kumar., S., Kumar, "Breast Cancer Prediction Using Support Vector Machine". *International Research Journal of Engineering and Technology (IRJET)*, vol. 06 no. 04 April-2019, pp. 2640-2643.
- [12] S. B., Halls, S. B. "Potentially malignant microcalcification by texture". Oct 24, 2019. [Online]. Available: <https://breast-cancer.ca/malcalctex/>
- [13] Radiopaedia, "Birads: mammography. in breast imaging reporting and data system: Bi-rads atlas". *ACR American College of Radiology*. Sept 29, 2021. [Online]. Available: [https://radiopaedia.org/articles/breast-imaging-reporting-and-data-system-bi-rads#:~:text=BI%2DRADS%20\(Breast%20Imaging%2D,mammography%2C%20ultrasound%2C%20and%20MRI.](https://radiopaedia.org/articles/breast-imaging-reporting-and-data-system-bi-rads#:~:text=BI%2DRADS%20(Breast%20Imaging%2D,mammography%2C%20ultrasound%2C%20and%20MRI.)
- [14] M. Hall-Beyer, "GlcM texture: A tutorial v. 3.0", March 2017. DOI : 10.13140/RG.2.2.12424.21767.
- [15] R. Haralick, "Statistical and structural approaches to texture". *Published in IEEE, Proceedings*, vol 67 no. 5 pp.786–804. 2015.
- [16] Gorunescu, Florin. "Data Mining: Concepts, Models, and Techniques". *Verlag Berlin Heidelberg: Springer*. Vol 12, 2011, DOI: 10.1007/978-3-642-19721-5.