

**KLASIFIKASI POLA SIDIK BIBIR UNTUK MENENTUKAN JENIS  
KELAMIN MANUSIA DENGAN METODE *GRAY LEVEL CO-  
OCCURRENCE MATRIX* DAN *SUPPORT VECTOR MACHINE***

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Eka Novita Shandra

NIM: 145150207111028



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

### PENGESAHAN

KLASIFIKASI POLA SIDIK BIBIR UNTUK MENENTUKAN JENIS KELAMIN MANUSIA  
DENGAN METODE *GRAY LEVEL CO-OCCURRENCE MATRIX* DAN *SUPPORT VECTOR  
MACHINE*

SKRIPSI

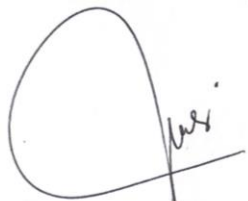
Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Eka Novita Shandra  
NIM: 145150207111028

Skripsi ini telah diuji dan dinyatakan lulus pada  
28 Desember 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Budi Darma Setiawan, S.Kom., M.Cs.

Yuita Arum Sari, S.Kom., M.Kom.

NIP: 198410152014041002

NIK: 2016098807152001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D.

NIP: 19710518 200312 1 001 



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Desember 2018



Eka Novita Shandra

NIM: 145150207111028



## PRAKATA

Alhamdulillah, puji syukur penulis panjatkan kepada Allah SWT karena atas segala rahmat dan karuniaNya penulis dapat menyelesaikan laporan skripsi berjudul “Klasifikasi Pola Sidik Bibir Untuk Menentukan Jenis Kelamin Manusia Dengan Metode Gray Co-occurrence Matris dan Support Vector Machine”. Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Bapak Budi Darma Setiawan S.Kom, M.Cs selaku pembimbing pertama penulis dan Bu Yuita Arum Sari S.Kom., M.Kom selaku dosen pembimbing kedua penulis.
2. Seluruh dosen Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya atas ilmu yang diberikan kepada penulis selama bangku perkuliahan.
3. Keluarga besar terutama kedua orang tua penulis, Ibu Siti Hadijah dan Bapak Ruslan yang tidak pernah berhenti mendoakan, menyemangati, dan memotivasi penulis dalam menyelesaikan skripsi ini.
4. Teman-teman yang tidak bisa disebutkan satu persatu yang bersedia difoto bibirnya untuk dataset skripsi penulis.
5. Mas Vendy dan Kak Azza sebagai penasehat dan juga membantu penulis dikala kebingungan dalam pengerjaan skripsi ini.
6. Nisa, Ninda, Sarah, Adit, Danang, Elha, Alfin, Rifqi, Alpia dan Mayang selaku teman dekat penulis.
7. Teman-teman dari Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya dan lainnya.

Penulis menyadari bahwa dalam penyusunan laporan skripsi ini masih banyak kekurangan, baik dari segi susunan maupun isi. Oleh karena itu, dengan kerendahan hati, penulis sangat mengharapkan kritik dan saran yang membangun agar skripsi ini dapat bermanfaat dan dapat digunakan untuk penelitian selanjutnya.

Malang, 17 Desember 2018

Penulis

ekanovitashandra@gmail.com

## ABSTRAK

**Eka Novita Shandra, Klasifikasi Pola Sidik Bibir Untuk Menentukan Jenis Kelamin Manusia Dengan Metode *Gray Level Co-Occurrence Matrix* Dan *Support Vector Machine***

**Pembimbing: Budi Darma Setiawan, S.Kom., M.Cs. dan Yuita Arum Sari, S.Kom., M.Kom.**

Identifikasi merupakan satu cara yang dapat dilakukan untuk mengenali ciri-ciri individu. Identifikasi diperlukan untuk mencari tahu kejelasan identitas personal pada orang yang sudah meninggal maupun orang yang masih hidup. Didalam dunia kedokteran forensik, peran identifikasi ini sangat penting. Seperti sidik jari, sidik bibir juga memiliki karakteristik unik pada setiap individu. Sidik bibir dapat digunakan sebagai sarana untuk identifikasi kasus forensik maupun non forensik. Untuk kasus nonforensik, sidik bibir dapat menentukan jenis kelamin pada individu. Untuk membantu pada proses identifikasi jenis kelamin berdasarkan sidik bibir, dibutuhkan suatu sistem klasifikasi yang dapat mengklasifikasikan jenis kelamin wanita dan pria. Proses diawali dengan mengumpulkan citra sidik bibir yang kemudian dilakukan preprocessing dan melakukan ekstraksi fitur tekstur dengan metode *Gray Leveled Co-ocurrence* (GLCM). Fitur yang digunakan ada 4 yaitu ASM, Contrast, Correlation dan IDM dengan sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ . Kemudian nilai fitur digunakan data untuk proses *training* dan *testing* menggunakan metode *Support Vector Machine* (SVM). Data latih yang digunakan pada pengujian yaitu 60 data. Hasil pada penelitian ini belum memberikan tingkat akurasi yang baik karena sistem hanya mampu memberikan akurasi sebesar 51,4% dengan pengujian parameter GLCM yaitu jarak =1 dan parameter SVM  $\lambda$  (*lambda*) = 0.5, *C* (*complexity*) = 1, konstanta (*gamma*) = 0.01, dan *itermax* = 100.

**Kata kunci:** Identifikasi, Jenis Kelamin, Sidik Bibir, *Support Vector Machine*, *Gray Leveled Co-ocurrence*

## ABSTRACT

**Eka Novita Shandra, Klasifikasi Pola Sidik Bibir Untuk Menentukan Jenis Kelamin Manusia Dengan Metode *Gray Level Co-Occurrence Matrix* Dan *Support Vector Machine***

**Pembimbing: Budi Darma Setiawan, S.Kom., M.Cs. dan Yuita Arum Sari, S.Kom., M.Kom.**

*Identification is one way that can be done to recognize individual characteristics. Identification is needed to find out the clarity of personal identity, for both deceased and living people. In the world of forensic medicine, the role of identification is very important. Like fingerprints, lip prints also have unique characteristics for each individual. Lip prints can be used as a means to identify forensic and non-forensic cases. For nonforensic cases, lip prints can determine the sex of an individual. To help in the process of identifying gender based on lip prints, a classification system is needed that can classify the sex of women and men. The process begins with collecting lip print images which are then preprocessed and extracted texture features using the Gray Leveled Co-occurrence (GLCM) method. There are 4 features that are used namely ASM, Contrast, Correlation and IDM with angles of  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$ . Then the feature value is used by data for the training and testing process using the Support Vector Machine (SVM) method. The training data used in the test is 60 data. The results of this study have not provided a good level of accuracy because the system is only able to provide an accuracy of 51.4% by testing the GLCM parameter, namely distance = 1 and SVM parameters  $\lambda$  (lambda) = 0.5, C (complexity) = 1, constant (gamma) = 0.01, and itermax = 100.*

**Keywords:** *Identification, Gender, Lip Prints, Support Vector Machine, Gray Leveled Co-occurrence*

## DAFTAR ISI

<b>PENGESAHAN .....</b>	<b>ii</b>
<b>PERNYATAAN ORISINALITAS .....</b>	<b>iii</b>
<b>PRAKATA .....</b>	<b>iv</b>
<b>ABSTRAK.....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vi</b>
<b>DAFTAR ISI .....</b>	<b>vii</b>
<b>DAFTAR TABEL .....</b>	<b>x</b>
<b>DAFTAR GAMBAR .....</b>	<b>xii</b>
<b>DAFTAR KODE PROGRAM .....</b>	<b>xiv</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan .....	3
1.4 Manfaat.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Pembahasan .....	4
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>6</b>
2.1 Kajian Pustaka .....	6
2.2 Bencana.....	8
2.3 Odontologi Forensik.....	9
2.3.1 Sidik Bibir .....	9
2.3.2 Klasifikasi Sidik Bibir .....	10
2.4 Pengolahan Citra.....	11
2.4.1 Jenis - Jenis Citra .....	12
2.5 <i>Gray Level Co-Occurrence Matrix (GLCM)</i> .....	12
2.6 <i>Support Vector Machine</i> .....	16
2.6.1 <i>Support Vector Machine Non-Linear</i> .....	17
2.6.2 <i>Sequential Training SVM</i> .....	17
2.6.3 <i>One Against All</i> .....	18
<b>BAB 3 METODOLOGI .....</b>	<b>20</b>

3.1 Studi Pustaka.....	20
3.2 Pengumpulan Data .....	21
3.3 Pengolahan Data.....	21
3.4 Perancangan Algoritme .....	23
3.5 Implementasi Sistem .....	24
3.6 Pengujian dan Analisis Sistem.....	24
3.7 Kesimpulan.....	24
<b>BAB 4 PERANCANGAN .....</b>	<b>25</b>
4.1 Formulasi Permasalahan.....	25
4.2 Perancangan Algoritme .....	25
4.2.1 Perancangan Proses Algoritme GLCM .....	27
4.2.2 Perancangan Proses Algoritme <i>Support Vector Machine</i> .....	43
4.3 Manualisasi .....	52
4.3.1 Perhitungan Manual Proses GLCM .....	52
4.3.2 Perhitungan Manual Proses SVM.....	66
4.4 Perancangan Pengujian .....	81
4.4.1 Perancangan Pengujian Pada Parameter <i>Lamda</i> .....	81
4.4.2 Perancangan Pengujian Pada Parameter <i>Gamma</i> .....	81
4.4.3 Perancangan Pengujian Pada Nilai <i>Complexity (C)</i> .....	82
4.4.4 Perancangan Pengujian Pada Nilai Iterasi Maksimal .....	83
4.4.5 Perancangan Pengujian Pada Nilai Jarak ( <i>Distance</i> ) .....	83
<b>BAB 5 IMPLEMENTASI .....</b>	<b>84</b>
5.1 Spesifikasi Sistem .....	84
5.2 Batasan Implementasi .....	84
5.3 Implementasi Algoritme .....	85
5.3.1 Implementasi <i>Gray Level Co-Occurence Matrix</i> .....	85
5.3.2 Implementasi <i>Support Vector Machine (SVM)</i> .....	89
5.4 Hasil Implementasi .....	94
<b>BAB 6 PENGUJIAN .....</b>	<b>97</b>
6.1 Pengujian Parameter $\lambda$ ( <i>Lambda</i> ).....	97
6.1.1 Skenario Pengujian Nilai <i>Lambda</i> .....	97
6.1.2 Analisis Pengujian Nilai <i>Lambda</i> .....	98



6.2 Pengujian Parameter <i>Gamma</i> .....	99
6.2.1 Skenario Pengujian Nilai <i>Gamma</i> .....	99
6.2.2 Analisis Pengujian Nilai <i>Gamma</i> .....	100
6.3 Pengujian Parameter <i>Complexity (C)</i> .....	100
6.3.1 Skenario Pengujian Nilai <i>Complexity (C)</i> .....	100
6.3.2 Analisis Pengujian Nilai <i>Complexity (C)</i> .....	101
6.4 Pengujian Parameter Iterasi Maksimal.....	102
6.4.1 Skenario Pengujian Nilai Iterasi Maksimal .....	102
6.4.2 Analisis Pengujian Nilai Iterasi Maksimal .....	103
6.5 Pengujian Nilai Jarak ( <i>d</i> ).....	104
6.5.1 Skenario Pengujian Nilai Jarak ( <i>d</i> ).....	104
6.5.2 Analisis Pengujian Nilai Jarak ( <i>d</i> ).....	105
<b>BAB 7 PENUTUP</b> .....	<b>112</b>
7.1 Kesimpulan.....	112
7.2 Saran .....	112
<b>DAFTAR REFERENSI</b> .....	<b>114</b>

## DAFTAR TABEL

Tabel 2.1 Keterangan Tipe Sidik Bibir.....	10
Tabel 2.2 Klasifikasi Sidik Bibir Berdasarkan Jenis Kelamin .....	11
Tabel 2.3 Area kerja matriks .....	13
Tabel 2.4 Contoh SVM dengan <i>One Against All</i> .....	19
Tabel 3.1 Tahapan Pengolahan Data .....	22
Tabel 3.2 Tahapan Pengolahan Data (lanjutan).....	23
Tabel 4.1 <i>Dataset</i> Nilai Citra Sidik Bibir.....	25
Tabel 4.2 Nilai <i>red</i> tiap-tiap <i>pixel</i> Gambar 4.27 .....	53
Tabel 4.3 Nilai <i>green</i> tiap-tiap <i>pixel</i> Gambar 4.27 .....	53
Tabel 4.4 Nilai <i>blue</i> tiap-tiap <i>pixel</i> Gambar 4.27 .....	54
Tabel 4.5 Hasil perhitungan <i>grayscale</i> .....	54
Tabel 4.6 Hasil nilai normalisasi nilai <i>pixel grayscale</i> .....	55
Tabel 4.7 Matriks <i>co-occurrence</i> GLCM sudut $0^{\circ}$ dari Gambar 4.28.....	56
Tabel 4.8 Matriks <i>co-occurrence</i> GLCM sudut $45^{\circ}$ dari Gambar 4.29.....	56
Tabel 4.9 Matriks <i>co-occurrence</i> GLCM sudut $90^{\circ}$ dari Gambar 4.30.....	57
Tabel 4.10 Tabel <i>co-occurrence</i> GLCM sudut $135^{\circ}$ dari Gambar 4.31 .....	58
Tabel 4.11 Data Latih Level 1 .....	67
Tabel 4.12 Data Latih ke-1 .....	67
Tabel 4.13 Data Latih ke-2 .....	68
Tabel 4.14 Data Latih ke-3 .....	69
Tabel 4.15 Hasil Perhitungan Data Latih Level 1 dengan <i>Kernel Polynomial</i> .....	70
Tabel 4.16 Hasil Perhitungan Matriks <i>Hessian</i> Level 1 .....	71
Tabel 4.17 Hasil Perhitungan $E_i$ Level 1 .....	72
Tabel 4.18 Hasil Perhitungan $\delta\alpha_i$ pada Iterasi ke 1 Level 1.....	73
Tabel 4.19 Hasil Perhitungan $\alpha_i$ pada Iterasi ke 1 Level 1.....	73
Tabel 4.20 Hasil Perhitungan Nilai $E_i$ Level 1 pada iterasi ke 2 .....	74
Tabel 4.21 Hasil Perhitungan $\delta\alpha_i$ pada Iterasi ke 2 Level 1.....	74
Tabel 4.22 Hasil Perhitungan $\alpha_i$ pada Iterasi ke 2 Level 1.....	75
Tabel 4.23 Hasil Perhitungan $K(x_i, x^+)$ dan nilai $K(x_i, x^-)$ Level 1 .....	76

Tabel 4.24 Hasil Perhitungan $w.x^+$ dan nilai $w.x^-$ Level 1.....	77
Tabel 4.25 Data Uji .....	78
Tabel 4.26 Hasil Perhitungan <i>Kernel</i> dan $\alpha_i y_i K(x_i, x)$ data ke-13 .....	80
Tabel 4.27 Data Hasil Perhitungan $f(x)$ dan Klasifikasi.....	81
Tabel 4.28 Skenario Pengujian Pada Nilai <i>Lambda</i> .....	81
Tabel 4.29 Skenario Pengujian Pada Nilai <i>Gamma</i> .....	82
Tabel 4.30 Skenario Pengujian Pada Nilai <i>C</i> .....	82
Tabel 4.31 Skenario Pengujian pada Iterasi Maksimal .....	83
Tabel 4.32 Skenario Pengujian Pada Nilai Jarak ( <i>Distance</i> ) .....	83
Tabel 6.1 Hasil Pengujian Nilai <i>Lambda</i> .....	97
Tabel 6.2 Hasil Pengujian Nilai <i>Gamma</i> .....	99
Tabel 6.3 Hasil Pengujian Nilai <i>Complexity(C)</i> .....	101
Tabel 6.4 Hasil Pengujian Nilai Iterasi Maksimal .....	103
Tabel 6.5 Hasil Pengujian Nilai Jarak ( <i>d</i> ) .....	105
Tabel 6.6 Hasil Nilai Parameter Terbaik Dari Setiap Pengujian .....	106
Tabel 6.7 Hasil Klasifikasi untuk data uji yang belum diketahui kelasnya .....	106
Tabel 6.8 Hasil Klasifikasi.....	107
Tabel 6.9 Perbandingan Hasil Klasifikasi Menggunakan Semua Fitur terhadap Fitur berdasarkan Uji Korelasi.....	109
Tabel 6.10 Nilai Rata-Rata Ekstraksi Ciri GLCM.....	111

## DAFTAR GAMBAR

Gambar 2.1 Klasifikasi Tipe Sidik Bibir Suzuki dan Tsuchihashi .....	10
Gambar 2.2 Pembagian Kuadran Bibir .....	10
Gambar 2.3 Ketetapan <i>pixel</i> sebagai fungsi orientasi sudut dan jarak.....	13
Gambar 2.4 Ilustrasi penentuan nilai awal matriks GLCM.....	14
Gambar 2.5 Hasil pembentukan GLCM simetris .....	14
Gambar 2.6 Normalisasi matriks dengan probabilitas .....	14
Gambar 2.7 <i>Hyperplane</i> SVM.....	16
Gambar 2.8 Contoh SVM dengan <i>One Against All</i> .....	19
Gambar 3.1 Diagram Blok Tahapan Penelitian .....	20
Gambar 3.2 Citra bibir sebelum di lakukan <i>cropping</i> .....	21
Gambar 3.3 Citra bibir setelah dilakukan <i>cropping</i> .....	21
Gambar 3.4 Tahapan-tahapan dari sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia .....	24
Gambar 4.1 Diagram Alir Proses Sistem .....	26
Gambar 4.2 Diagram Alir Proses GLCM .....	27
Gambar 4.3 Diagram Alir <i>Pre-processing</i> .....	28
Gambar 4.4 Diagram Alir Perhitungan GLCM $0^\circ$ , $45^\circ$ , $90^\circ$ dan $135^\circ$ .....	29
Gambar 4.5 Diagram Alir Perhitungan <i>Grayscale</i> .....	30
Gambar 4.6 Diagram Alir Min-Max Normalization .....	31
Gambar 4.7 Diagram Alir Matriks Ternormalisasi Sudut $0^\circ$ , $45^\circ$ , $90^\circ$ , dan $135^\circ$ ..	32
Gambar 4.8 Diagram Alir Perhitungan Matriks Berpasangan.....	34
Gambar 4.9 Diagram Alir Proses Matriks Transpose .....	35
Gambar 4.10 Diagram Alir Perhitungan Matriks Simetris Kookurensi .....	36
Gambar 4.11 Diagram Alir Perhitungan Matriks Normalisasi.....	37
Gambar 4.12 Diagram Alir Perhitungan Ekstraksi Fitur .....	38
Gambar 4.13 Diagram Alir Perhitungan Fitur ASM .....	39
Gambar 4.14 Diagram Alir Perhitungan Fitur Contrast.....	40
Gambar 4.15 Diagram Alir Perhitungan Fitur Correlation .....	41
Gambar 4.16 Diagram Alir Perhitungan Fitur IDM.....	42
Gambar 4.17 Diagram Alir Proses <i>Support Vector Machine</i> .....	43

Gambar 4.18 Diagram Alir Perhitungan <i>Kernel Polynomial</i> .....	44
Gambar 4.19 Diagram Alir Perhitungan <i>Sequential training SVM</i> .....	45
Gambar 4.20 Diagram Alir Perhitungan Matriks <i>Hessian</i> .....	46
Gambar 4.21 Diagram Alir Perhitungan Nilai $E_i$ .....	47
Gambar 4.22 Diagram Alir Perhitungan Nilai $\delta\alpha_i$ .....	48
Gambar 4.23 Diagram Alir Perhitungan Nilai $\alpha_i$ baru.....	49
Gambar 4.24 Diagram Alir Perhitungan <i>Testing SVM</i> .....	50
Gambar 4.25 Diagram Alir Perhitungan Nilai $f(x)$ .....	51
Gambar 4.26 Diagram Alir Perhitungan Nilai $b$ .....	52
Gambar 4.27 Potongan citra bibir dengan ukuran 10x10 piksel .....	53
Gambar 4.28 Pembentukan <i>co-occurrence pixel (3,4)</i> sudut $0^\circ$ dan $d=1$ .....	55
Gambar 4.29 Pembentukan <i>co-occurrence pixel (3,4)</i> sudut $45^\circ$ dan $d=1$ .....	56
Gambar 4.30 Pembentukan <i>co-occurrence pixel (3,4)</i> sudut $90^\circ$ dan $d=1$ .....	57
Gambar 4.31 Pembentukan <i>co-occurrence pixel (3,4)</i> sudut $135^\circ$ dan $d=1$ .....	57
Gambar 5.1 Tampilan Antarmuka Hasil Implementasi .....	95
Gambar 5.2 Nilai Proses <i>Training</i> .....	96
Gambar 5.3 Nilai Hasil Klasifikasi .....	96
Gambar 5.4 Hasil Klasifikasi Proses Pengujian Akurasi .....	96
Gambar 6.1 Grafik Hasil Pengujian Parameter <i>Lambda</i> .....	98
Gambar 6.2 Grafik Hasil Pengujian Parameter <i>Gamma</i> .....	100
Gambar 6.3 Grafik Hasil Pengujian Parameter Nilai $C$ .....	102
Gambar 6.4 Grafik Hasil Pengujian Parameter Iterasi Maksimal.....	104
Gambar 6.5 Grafik Hasil Pengujian Parameter Nilai Jarak ( $d$ ).....	105

## DAFTAR KODE PROGRAM

Kode Program 5.1 Perhitungan <i>Grayscale</i> .....	85
Kode Program 5.2 Perhitungan Min-Max Normalization .....	86
Kode Program 5.3 Perhitungan Matrix Coocurance .....	87
Kode Program 5.4 Perhitungan Fitur ASM .....	87
Kode Program 5.5 Perhitungan Fitur Contrast .....	88
Kode Program 5.6 Perhitungan Fitur Correlation .....	88
Kode Program 5.7 Perhitungan Fitur IDM .....	89
Kode Program 5.8 Implementasi Perhitungan Kernel SVM .....	90
Kode Program 5.9 Perhitungan Matriks Hessian .....	90
Kode Program 5.10 Perhitungan Nilai $E_j$ .....	91
Kode Program 5.11 Perhitungan Nilai $\delta\alpha_i$ .....	92
Kode Program 5.12 Perhitungan Nilai $\alpha_i$ .....	92
Kode Program 5.13 Perhitungan Nilai $w, x +$ dan $w, x -$ .....	93
Kode Program 5.14 Perhitungan Nilai $b$ .....	94
Kode Program 5.15 Perhitungan Nilai $f(x)$ .....	94

## BAB 1 PENDAHULUAN

Pada bab ini berisi tentang penjelasan mengenai latar belakang pemilihan topik skripsi, rumusan masalah yang akan di selesaikan, tujuan penelitian, manfaat yang akan di dapat setelah penelitian selesai, batasan masalah yang membatasi penelitian, dan kerangka sistematika pembahasan.

### 1.1 Latar Belakang

Manusia memiliki ciri-ciri yang unik yang melekat pada setiap bagian tubuhnya. Bagian tubuh itulah yang dapat membedakan satu individu dengan individu lain. Beberapa diantaranya yaitu sidik jari, iris dan retina mata, bentuk geometri tangan, kunci frekuensi suara, dan rincian ciri wajah. Terdapat satu bagian tubuh yang tidak banyak kita tahu bahwa dapat membedakan individu satu dengan yang lain, bagian tubuh itu adalah bibir. Bibir memiliki guratan garis yang menyebar di seluruh permukaannya, guratan garis tersebut adalah sidik bibir. Studi yang mempelajari tentang metode identifikasi berdasarkan sidik bibir disebut *cheiloscropy* (Randhawa, Narang, & Arora, 2011). Sidik bibir dapat menjadi sarana identifikasi untuk kasus forensik maupun non forensik. Untuk kasus forensik, sidik bibir dapat digunakan untuk memecahkan kasus pembunuhan, pemerkosaan ataupun pencurian. Sedangkan pada kasus yang non forensik, sidik bibir dapat digunakan untuk mengidentifikasi jenis kelamin, usia, dan juga ras (Septadina, 2015). Pemanfaatan sidik bibir pernah dilakukan di Hongaria pada tahun 1962. Peristiwa yang terjadi yaitu pada kasus pembunuhan seseorang. Sidik bibir ditemukan pada pintu kaca tempat kejadian, dikarenakan tangan pembunuh bersimbah darah, pembunuh mendorong pintu menggunakan kepalanya dan tidak sengaja meninggalkan bekas cap bibir pada pintu kaca. Sidik bibir kemudian dibandingkan dengan salah satu terduga kriminal dan mendapatkan kecocokan hasil pada identifikasinya (Sekharan, 2013).

Selain pada kasus kriminalitas, sidik bibir juga dapat membantu dalam proses identifikasi jenazah korban bencana. Dari website Data Informasi Bencana Indonesia (DIBI), pada 10 tahun terakhir Indonesia mengalami bencana dengan jumlah kejadian sebanyak 20.733 kali dan tercatat memakan korban jiwa sebanyak 8.197 korban diantaranya ada yang meninggal dan hilang. Dari beberapa jenis bencana, kecelakaan transportasi paling banyak memakan korban jiwa, yaitu sebanyak 2.377 korban. Indonesia juga termasuk salah satu negara yang masih memiliki tingkat kecelakaan lalu lintas tertinggi di ASEAN setelah Thailand, Vietnam, dan Malaysia (Dewi, 2017). Namun kecelakaan yang terjadi tidak hanya kecelakaan lalu lintas yang disebabkan oleh *Man Made Disaster*, tetapi dapat terjadi *Natural Disaster* yang disebabkan oleh alam (Prawestiningtyas & Algozi, 2009). Berdasarkan data yang didapatkan, salah satu kecelakaan terjadi di akhir tahun 2006. Tenggelamnya KM.Senopati Nusantara merupakan kecelakaan yang menewaskan ratusan korban jiwa. Tetapi dari seluruh korban jiwa yang ditemukan, tidak semua dapat diidentifikasi dengan baik. Dari 36 jenazah yang dikirimkan ke Instalasi Ilmu Kedokteran Forensik

hanya 13 jenazah yang dapat diidentifikasi dengan pemeriksaan DNA. Selain kecelakaan yang terjadi di laut, terjadi pula kecelakaan di udara dengan terbakarnya pesawat Garuda GA 200 PK-GZC Boeing 737-400 di tahun 2007. Kecelakaan udara yang membawa 133 penumpang dan 7 awak pesawat, telah menewaskan 21 penumpang. Karena korban terbakar, maka tidak semua jenazah dapat diidentifikasi dengan mudah. Sehingga digunakan berbagai cara kombinasi dari identifikasi pada korban jiwa (Prawestiningtyas & Algozi, 2009).

Berdasarkan contoh kasus-kasus tersebut, identifikasi perlu dikembangkan. Pemeriksaan forensik dengan DNA tidaklah cukup untuk membantu penanganan secara medis pada korban jiwa bencana. Tidak berbeda dari sidik jari, sidik bibir bisa dikembangkan pada proses forensik karena sidik bibir memiliki pola yang unik dan bersifat individual setiap orangnya. Sidik bibir sudah dapat diamati saat bayi masih berusia empat bulan. Beberapa penelitian melaporkan bahwa sidik bibir bersifat permanen dan tidak berubah selama hidup seseorang (Septadina, 2015). Sehingga sidik bibir dapat dijadikan salah satu saran dalam mengidentifikasi korban jiwa secara langsung ditempat kejadian perkara. Identifikasi sidik bibir belum banyak dilakukan, tetapi sudah mulai dikembangkan seperti kasus non forensik untuk menentukan jenis kelamin seseorang. Proses identifikasi masih dilakukan dengan metode manual. Metode ini dirasa sangat memakan waktu dalam prosesnya, seperti penelitian yang dilakukan oleh Kumar (2016). Peneliti melakukan proses yang lumayan banyak, seperti mempersiapkan pewarna bibir, isolasi bening, gunting, buku catatan, dan kaca pembesar. Lalu melakukan pewarnaan bibir yang kemudian akan mendapatkan cap bibir dengan ditempelkan isolasi bening, lalu kemudian mengamati tipe bibir dengan kaca pembesar. Peneliti mendapatkan beberapa sampel tipe sidik bibir dari objek penelitiannya, yaitu untuk bibir wanita kuadran yang dominan dengan tipe I dan I', sedangkan pria kuadran yang dominan dengan tipe III dan IV. Maka dari itu cara membedakan jenis kelamin pria dan wanita yang lebih akurat adalah melalui tekstur sidik bibir. Dari analisis tersebut Penulis memilih tekstur sebagai ciri pembeda antara bibir untuk jenis kelamin pria dan bibir untuk jenis kelamin wanita. Ada beberapa metode untuk membedakan tekstur salah satunya adalah metode ekstraksi ciri GLCM.

GLCM adalah salah satu metode untuk memperoleh ciri-ciri citra tekstur dengan menghitung probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan arah tertentu. Haralick (1973) mengusulkan berbagai jenis ciri tekstur yang dapat diekstraksi dari matriks kookurensi. Adapun parameter atau ciri-ciri tekstur yang biasa digunakan dalam penelitian sebelumnya dari metode GLCM diantaranya adalah *Angular Second Moment*, *Contrast*, *Correlation*, dan *Inverse Difference Moment*. GLCM merupakan metode yang sudah banyak digunakan oleh para peneliti untuk mendapatkan nilai tekstur dari sebuah citra. Salah satu penelitian tentang ekstraksi fitur GLCM diantaranya yaitu penelitian oleh Husdi (2016) tentang pengenalan ekspresi wajah berbasis citra dengan menggunakan GLCM sebagai ekstraksi fitur tekstur dengan 4 fitur yaitu ASM, *Contrast*, *Correlation*, dan IDM. Akurasi yang didapatkan dari penelitian tersebut bernilai 88,89%. Penelitian selanjutnya dilakukan oleh Kurniawan (2016) yaitu identifikasi



penyakit daun tembakau dengan metode GLCM sebagai ekstraksi fitur tekstur dan menggunakan metode SVM sebagai pengklasifikasi jenis penyakit daun tembakau. Rata-rata hasil tingkat keberhasilan pada sistem ini adalah 74% dengan persentase keberhasilan tertinggi 80% pada kernel *Polynomial* dengan jarak piksel 1, 2, 3, 5 dan 6. Sedangkan persentase keberhasilan terkecil bernilai 63% pada kernel *gaussian* (rbf) dengan jarak piksel 1. Pada penelitian tersebut menggunakan 4 fitur GLCM yaitu *ASM*, *Contrast*, *Correlation*, dan *IDM*, yang kemudian hasil dari masing-masing ekstraksi fitur diklasifikasikan menggunakan metode *Support Vector Machine*. SVM merupakan salah satu metode pembelajaran yang menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang fitur (*feature space*) berdimensi tinggi, dilatih dengan algoritme pembelajaran yang didasarkan pada teori optimasi dengan mengimplementasikan *learning bias* yang berasal dari teori pembelajaran statistik. Dengan melihat dari konsep metode *Support Vector Machine* ini, muncul pemikiran apakah metode ini dapat digunakan untuk mengidentifikasi jenis kelamin pria dan wanita menggunakan sidik bibir. SVM dipilih karena merupakan salah satu metode klasifikasi yang *robust* dalam mengklasifikasikan dua kelas data (Lathief, 2016).

Dari hasil pemaparan mengenai fakta-fakta terkait dengan sidik bibir, metode GLCM dan metode SVM tersebut menjadikan alasan utama Penulis dalam membuat suatu sistem yang dapat berguna dan membantu penentuan jenis kelamin individu menggunakan sidik bibir. Penentuan jenis kelamin didasarkan pada klasifikasi pola sidik bibir manusia. Penanganan yang ingin dilakukan oleh Penulis adalah membuat suatu rancangan program yang memanfaatkan citra digital bibir dalam mengidentifikasi jenis kelamin manusia, maka dibuatlah sebuah sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan ekstraksi fitur tekstur menggunakan *Gray Level Co-occurrence Matrix* dan dalam melakukan klasifikasi tipe bibir menggunakan *Support Vector Machine*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana tingkat akurasi dari implementasi metode *Gray Level Co-occurrence matrix* dan *support vector machine* dalam mengklasifikasikan pola sidik bibir?
2. Bagaimana pengaruh variasi nilai parameter *lambda*, *gamma*, *C*, iterasi maksimal *dan d* untuk akurasi dari implementasi metode *Gray Level Co-occurrence Matrix* dan *Support Vector Machine* dalam mengklasifikasi pola sidik bibir?

## 1.3 Tujuan

Dari rumusan masalah yang telah diberikan, penelitian ini dilakukan dengan tujuan:

1. Menguji tingkat akurasi dari implementasi metode *Gray Level Co-occurrence Matrix* dan *Support Vector Machine* dalam mengklasifikasi pola sidik bibir.
2. Mengetahui pengaruh variasi nilai parameter SVM untuk implementasi metode *Gray Level Co-occurrence Matrix* dan *Support Vector Machine* dalam mengklasifikasi pola sidik bibir.

#### 1.4 Manfaat

Manfaat bagi Penulis, yaitu:

1. Dapat lebih memahami tentang implementasi metode *Gray Level Co-occurrence Matrix* dan *Support Vector Machine* dalam mengklasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia.
2. Dapat menerapkan ilmu yang telah didapat terutama tentang metode *Gray Level Co-occurrence Matrix* dan *Support Vector Machine* dalam mengklasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia.

Manfaat bagi pengguna, yaitu:

1. Mempermudah pengguna dalam penentuan jenis kelamin berdasarkan klasifikasi pola sidik bibir dengan metode *Gray Level Co-occurrence Matrix* dan *Support Vector Machine*.
2. Membantu dalam dunia kedokteran untuk mengidentifikasi jenazah melalui sidik bibir.

#### 1.5 Batasan Masalah

Untuk melakukan penelitian secara spesifik dan jelas, diperlukan batasan yang diterapkan pada penelitian ini, batasan-batasan tersebut meliputi:

1. *Dataset* yang digunakan yaitu citra bibir wanita berjumlah 30 dan pria berjumlah 30.
2. *Cropping dataset* dilakukan menggunakan aplikasi Adobe Photoshop Express.
3. *Dataset* diperoleh dengan menggunakan kamera iPhone 5s dengan jarak 20cm dari objek bibir.
4. Pengambilan citra dilakukan pada tempat *indoor* dan *outdoor*.

#### 1.6 Sistematika Pembahasan

##### Bab 1 Pendahuluan

Bab ini membahas mengenai alasan utama dari suatu topik yang diangkat sebagai objek penelitian pada klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine*. Alasan utama tersebut dituangkan dalam latar belakang masalah, rumusan masalah pada penelitian, tujuan penelitian, manfaat pada hasil penelitian, batasan masalah yang akan dibahas pada penelitian, dan sistematika pembahasan.

## **Bab 2 Landasan Kepustakaan**

Bab ini membahas mengenai teori-teori dasar sebagai referensi ataupun pernyataan pendukung dilakukannya penelitian terhadap klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine*. Teori yang dibahas pada landasan kepustakaan yaitu mengenai bencana, odontologi forensik, citra digital, klasifikasi, dan metode yang digunakan dalam penelitian.

## **Bab 3 Metodologi Penelitian**

Bab ini membahas mengenai alur penelitian yang dilakukan seperti studi kepustakaan, pengumpulan data, perancangan sistem, implementasi sistem serta pengujian sistem pada klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine*.

## **Bab 4 Perancangan Sistem**

Bab ini membahas mengenai analisis kebutuhan serta perancangan sistem yang akan diterapkan pada penelitian dalam pembuatan sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine*.

## **Bab 5 Implementasi Sistem**

Bab ini membahas mengenai penjelasan implementasi klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine*.

## **Bab 6 Pengujian dan Analisis**

Bab ini membahas mengenai pengujian terhadap sistem sehingga dapat mengetahui tingkat keberhasilan sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine* dalam memberikan suatu solusi.

## **Bab 7 Kesimpulan dan Saran**

Bab ini membahas mengenai kesimpulan dari setiap proses dan hasil yang didapatkan selama penelitian berlangsung dan dibahas secara ringkas. Saran yang diperoleh dari hasil penelitian akan dicantumkan dengan harapan jika terdapat penelitian serupa akan memberikan hasil yang jauh lebih baik dari penelitian sebelumnya.

## BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisi pembahasan kajian pustaka, dasar teori, dan metode yang digunakan dalam penelitian ini. Kajian pustaka pada penelitian berasal dari penelitian terdahulu yang relevan dengan penelitian yang dilakukan sekarang.

### 2.1 Kajian Pustaka

Terdapat beberapa penelitian tentang metode *Gray Level Co-occurrence Matrix* dan *Support Vector Machine* dilakukan oleh (Neneng, Adi & Isnanto, 2016), (Dewi & Ginardi, 2014), (Kurniawan, Rasmana & Triwidyastuti, 2016), dan (Patil, et al., 2016). Permasalahannya yaitu klasifikasi jenis daging, indentifikasi penyakit pada daun tebu, indentifikasi jenis penyakit daun tembakau, dan pengenalan iris. Dimana hasil yang didapatkan terbukti optimal dan efektif dengan menggunakan metode *Gray Level Co-occurrence Matrix* dan *Support Vector Machine*. Metode *Gray Level Co-occurrence Matrix* digunakan sebagai ekstraksi fitur citra, sedangkan metode *Support Vector Machine* untuk klasifikasi, untuk lebih jelasnya sapat dilihat pada Tabel 2.1.

Tabel 2.1 Tabel Penelitian Sebelumnya

No	Peneliti	Input	Metode & Proses	Hasil & Akurasi
1	(Neneng, Adi & Isnanto., 2016)	Citra digital daging kerbau, kuda dan sapi.	Metode: GLCM-SVM Langkah-langkah: 1. <i>Pre-Processing</i> citra 2. Ekstraksi ciri fitur menggunakan GLCM 3. Menghitung ciri atau rata-rata ciri statistik. 4. Klasifikasi menggunakan SVM	- Hasil: pengenalan klasifikasi daging, yaitu daging kambing, kerbau, kuda, dan sapi. - Akurasi: 87,5%
2	(Dewi & Ginardi., 2014)	Citra digital daun tebu yang telah di berikan label kelas: <ul style="list-style-type: none"> <li>• kelas 1 untuk tebu dengan penyakit karat.</li> <li>• kelas 2 untuk tebu dengan penyakit</li> </ul>	Metode: GLCM-SVM Langkah-langkah: 1. <i>Pre-Processing</i> citra. 2. Ekstraksi fitur tekstur dengan <i>Gray Level Co-occurrence Matrix</i> (GLCM) dan ekstraksi fitur warna dengan <i>color moments</i> . 3. Klasifikasi menggunakan SVM	- Hasil: identifikasi penyakit pada daun tebu. - Akurasi: 97%

		<p>mosaik.</p> <ul style="list-style-type: none"> <li>• kelas 3 untuk tebu normal.</li> </ul>		
3	(Kurniawan, Rasmana & Triwidyastuti., 2016)	<p>citra <i>cropping</i> penyakit daun tembakau yang berukuran 96 x 96 piksel</p>	<p>Metode: GLCM-SVM Langkah-langkah: 1. <i>Gray-Scalling</i> citra. 2. Membuat matrix kookurensi. 3. mencari nilai fitur-fitur dari GLCM, yaitu <i>energy</i>, <i>Contrast</i>, <i>Correlation</i> dan <i>homogeneity</i>. 4. Pelatihan data latih dengan SVM.</p>	<ul style="list-style-type: none"> <li>- Hasil: hasil keluaran yang didapat dari penelitian ini berupa penyakit tembakau.</li> <li>- Akurasi : 74%</li> </ul>
4	(Patil, et al., 2016)	<p>Citra mata yang sudah di segmentasi</p>	<p>Metode: GLCM-SVM Langkah-langkah: 1. Segmentasi citra. Tepi-tepi batas dalam dan luar iris di deteksi menggunakan deteksi tepi <i>Canny</i>. 2. Pola iris yang kompleks dideteksi menggunakan momen <i>Gaussian Hermite</i>. 3. Melakukan Hough transform yaitu digunakan untuk mencari tahu lingkaran dalam Gambar iris dan mengurangi piksel pada batas lingkaran. 4. Menghitung parameter Hough transform dengan GLCM, yaitu <i>energy</i>, <i>Contrast</i>, <i>entropy</i>, <i>solidity</i>,</p>	<ul style="list-style-type: none"> <li>- Hasil: Menghasilkan pengenalan individu.</li> <li>- Akurasi : 97,5%.</li> </ul>

			<p><i>Correlation</i> dan <i>homogeneity</i>.</p> <p>5. Ekstraksi ciri iris dilakukan dengan biometrik GLCM (Gray Scale Co-occurrence Matrix) dan HD (Hausdorff Dimension).</p> <p>6. Klasifikasi data menggunakan SVM.</p>	
5	(Dongarwar, Bhowate, & Degwekar, 2013).	20 sampel bibir pria dan 20 sampel bibir wanita	<p>1. Memberi lipstik secara merata pada bibir.</p> <p>2. Menempelkan <i>tape</i> pada bibir kemudian menempelkan hasil cetakan pada kertas.</p> <p>3. Memperhatikan pola sidik bibir menggunakan kaca pembesar.</p>	<p>- Menghasilkan jenis kelamin dari pemilik sample bibir.</p> <p>- 18 bibir dari 20 cetakan sidik bibir wanita benar dan 16 dari 20 cetakan bibir pria benar.</p>

## 2.2 Bencana

Bencana merupakan hal yang sering terjadi, khususnya di Indonesia. Bencana adalah peristiwa maupun rangkaian dari peristiwa yang dapat mengancam dan mengganggu kehidupan masyarakat. Terdapat dua faktor dari bencana yaitu, bencana yang disebabkan oleh alam dan non-alam. Faktor non-alam terjadi ketika bencana disebabkan oleh manusia, sehingga terdapat korban jiwa, kerusakan lingkungan, kerugian harta benda, maupun dampak psikologis (BNPB, 2017).

Berdasarkan Undang-Undang nomor 24 Tahun 2007 yang telah ditetapkan, telah didefinisikan dari setiap bencana yang terjadi. Baik dari bencana alam, bencana non-alam, dan bencana sosial. Jenis-jenis bencana tersebut adalah sebagai berikut (BNPB, 2017):

1. Bencana alam, merupakan bencana yang disebabkan oleh alam pada umumnya bencana alam disebabkan oleh perubahan alam, baik secara perlahan ataupun secara ekstrim. Bencana alam yang terjadi diantaranya gempa bumi, gunung meletus, banjir, tsunami, kekeringan, angin topan, dan tanah longsor.

2. Bencana non-alam, peristiwa bencana yang terjadi ketika penyebabnya adalah manusia. Contoh dari bencana non-alam yaitu gagal teknologi, gagal modernisasi, epidemia, wabah penyakit.
3. Bencana sosial, bencana yang terjadi karena peristiwa yang diakibatkan oleh konflik sosial antarkelompok atau antar komunitas masyarakat, dan teror. Bencana sosial yang biasanya disebabkan oleh konflik atau perbedaan, contohnya adalah perbedaan kepercayaan, tingkat kesejahteraan, perbedaan warna kulit dan adanya provokator dari pihak-pihak tertentu yang dapat menyebabkan konflik ataupun kerusuhan.

## 2.3 Odontologi Forensik

Cabang ilmu dari kedokteran gigi salah satunya ialah *odontologi forensik*. Penerapan ilmu dari *odontologi forensik* berguna dalam membantu permasalahan hukum dalam kepentingan peradilan, baik hukum pidana maupun hukum perdata. Ilmu kedokteran gigi forensik juga dapat membantu mempermudah penyidik dalam mengidentifikasi korban baik melalui gigi, *rugae palatina*, dan sidik bibir (*lip print*). Peran utama yang diberikan dari ilmu ini adalah membantu forensik terhadap korban peristiwa bencana. Menurut Eckert (1997) forensik dapat dilakukan pada kematian yang tidak wajar dan tidak terduga, kerusakan fisik yang direncanakan, dan keterlambatan ketika menemukan jenazah yang menghambat proses identifikasi tubuh yang tidak dapat dikenali.

Menurut Saxena, Sharma dan Gupta (2010) proses odontologi forensik yang diterapkan saat ini adalah :

1. Pemeriksaan dan evaluasi pada luka yang terdapat pada jaringan lunak mulut, rahang, dan gigi.
2. Identifikasi pada individu dalam tindak kriminal maupun bencana massal.
3. Identifikasi, evaluasi, dan pemeriksaan bekas gigitan yang terjadi pada beberapa kasus kejahatan seksual, kekerasan pada anak, dan pertahanan diri terhadap seseorang.
4. Penentuan usia seseorang.

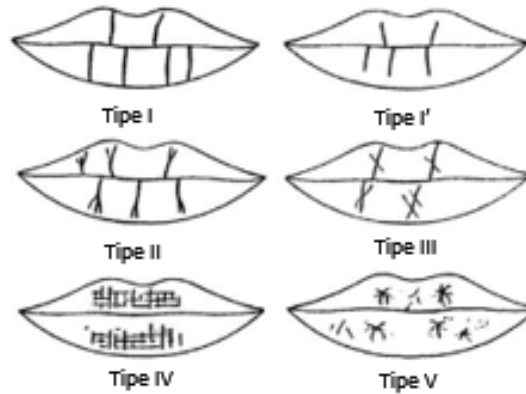
### 2.3.1 Sidik Bibir

Bibir adalah salah satu bagian tubuh, merupakan suatu jaringan lunak yang mengelilingi mulut. Anatomi pada bibir dibagi menjadi dua bagian, yaitu bibir bagian atas dan bibir bagian bawah. Secara histologi kedua bagian bibir terusun dari epidermis, jaringan subkutan, serat otot orbikularis oris, dan membran mukosa (Syafri, Bambang & Fahmi, 2017). Secara tak kasat mata akan terlihat alur pada bibir. Alur tersebut memiliki berbagai macam variasi, bentuk, maupun arah pada setiap individu. Arah yang terbentuk dapat berupa vertikal, horizontal, dan miring. Alur yang terbentuk akan berbeda pada setiap individu, maka dari itu didapatkan sidik bibir dari gambaran alur-alur pada bibir. Alur pada bibir sehingga ditemukan sidik bibir dinamakan sebagai *figura linearum labiorum rubrorum* oleh Suzuki dan Tsuchihashi. Karena keunikan dari sidik bibir maka dari

setiap sidik bibir manusia dapat dilakukan identifikasi identitas seperti sidik jari (Kasprzak, 1990).

### 2.3.2 Klasifikasi Sidik Bibir

Identifikasi pada sidik bibir dilakukan dengan mengklasifikasikan jenis-jenis sidik bibir yang ada disetiap individunya. Namun, dalam melakukan identifikasi diperlukan beberapa pola sidik bibir. Hal ini telah dikemukakan oleh Suzuki dan Tsuchihashi (1970) yang membagi pola sidik bibir menjadi enam, diantaranya:

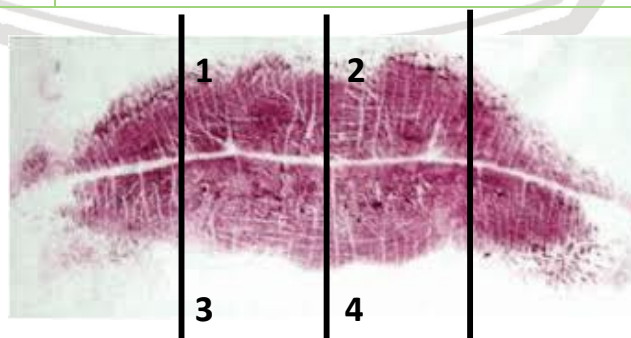


**Gambar 2.1 Klasifikasi Tipe Sidik Bibir Suzuki dan Tsuchihashi**

Sumber : Suzuki & Tsuchihashi (1970)

**Tabel 2.1 Keterangan Tipe Sidik Bibir**

No.	Tipe Bibir	Keterangan
1	Tipe I	alur vertikal disepanjang permukaan bibir.
2	Tipe I'	alur vertikal seperti tipe I, tapi tidak disepanjang permukaan bibir.
3	Tipe II	alur yang bercabang.
4	Tipe III	alur yng saling memotong.
5	Tipe IV	alur yang menyerupai jala.
6	Tipe V	alur yang memiliki tipe yang berbeda dari tipe I-IV.



**Gambar 2.2 Pembagian Kuadran Bibir**



Untuk mengidentifikasi pola sidik bibir berdasarkan jenis kelamin yaitu menggunakan kuadran 1, 2, 3, dan 4. Untuk kuadran 1 yaitu bibir atas bagian kanan, kuadran 2 yaitu bibir atas bagian kiri, kuadran 3 yaitu bibir bawah bagian kanan, kuadran 4 yaitu bibir bawah bagian kiri. Sudut bibir tidak digunakan karena permukaannya semakin mengecil sehingga pembacaan tipe sidik bibir pun akan sulit dilakukan (Dongarwar, Bhowate, & Degwekar, 2013). Penelitian yang dilakukan oleh Vahanwala-Parekh disarankan agar pola sidik bibir diklasifikasikan berdasarkan jenis kelamin dengan ketentuan pada Tabel 2.2.

**Tabel 2.2 Klasifikasi Sidik Bibir Berdasarkan Jenis Kelamin**

Dominan	Kuadran	Tipe
Wanita	1	Tipe I & Tipe I'
Pria	2	Tipe II
Wanita	3	Tipe I
Wanita	4	Tipe I
Pria	1 dan 2	Tipe III
Varied Patterns	Semua Kuadran	Pria
Same (alike) Patterns	Semua Kuadran	Wanita

## 2.4 Pengolahan Citra

Citra adalah gambar berbentuk dua dimensi (dwimatra). Citra berupa gambar yang dihasilkan dari gambar analog dua dimensi dan secara kontinu menjadi gambar diskrit. Proses pengolahan gambar dari analog hingga menjadi diskrit dilakukan dari proses sampling gambar analog yang dibagi menjadi baris dan kolom (Purba, 2010). Secara matematis citra adalah fungsi secara kontinu yang dipengaruhi oleh intensitas cahaya, dimana sebagian dari berkas cahaya yang mengenai objek akan dipantulkan kembali oleh objek. Bayangan objek yang terekam adalah pantulan dari cahaya yang ditangkap oleh alat-alat optik, seperti mata, kamera, *scanner*, dan lainnya (Munawaroh & Susanto., 2010).

Dalam memproses suatu citra yang telah ditangkap, diperlukan pengolahan citra yang membantu dalam menginterpretasikan citra yang mengalami gangguan. Pengolahan citra dapat dilakukan dengan cara memanipulasi citra menjadi wujud citra yang lebih baik. Menurut Jain (1989) operasi yang dapat dilakukan dalam pengolahan citra adalah sebagai berikut :

1. Memodifikasi atau melakukan perbaikan pada citra agar meningkatkan kualitas citra dengan menonjolkan beberapa aspek informasi terkandung,
2. Elemen-elemen pada citra yang perlu dikelompokkan, dicocokkan, ataupun diukur.
3. Penggabungan sebagian dari citra dengan citra lainnya.

Pengolahan citra yang memerlukan bantuan mesin disebut juga pengolahan citra digital. Citra yang diolah secara digital perlu direpresentasikan nilainya menjadi *numeric* atau nilai-nilai diskrit. Hasil yang diberikan dengan bantuan

mesin disebut dengan citra digital, dimana bentuk dari citra digital dinyatakan sebagai tinggi dan lebar (Munir, 2004).

### 2.4.1 Jenis - Jenis Citra

Pada pengolahan citra *digital* pada umumnya, citra *digital* dapat dibagi menjadi tiga, yaitu *color image*, *black and white*, dan *binary image*.

#### 1. RGB (*Red, Green, Blue*)

Dalam model ini, tiap warna ditunjukkan dengan kombinasi tiga warna primer yang membentuk sistem koordinat cartesian tiga dimensi. Jika masing-masing warna memiliki range 0-255, maka totalnya adalah  $255^3 = 16.581.375$  variasi warna yang berbeda pada gambar. Seperti pada Gambar 2.3 sub ruang pada diagram tersebut menunjukkan posisi tiap warna. Nilai RGB terletak pada satu sudut dengan *cyan*, *magenta*, dan *yellow* berada di sudut lainnya.

#### 2. Grayscale

Citra *digital black and white* atau *grayscale* disimpan sebagai integer 8 bit sehingga memberikan  $2^8 = 256$  tingkat keabuan dari warna putih sampai hitam. Untuk memperoleh citra *grayscale*, perlu dilakukan pengubahan nilai representasi RGB menjadi nilai representasi *grayscale* menggunakan persamaan 2.1.

$$grayscale = \frac{R+G+B}{3} \quad (2.1)$$

Keterangan:

R = nilai representasi dari *Red*

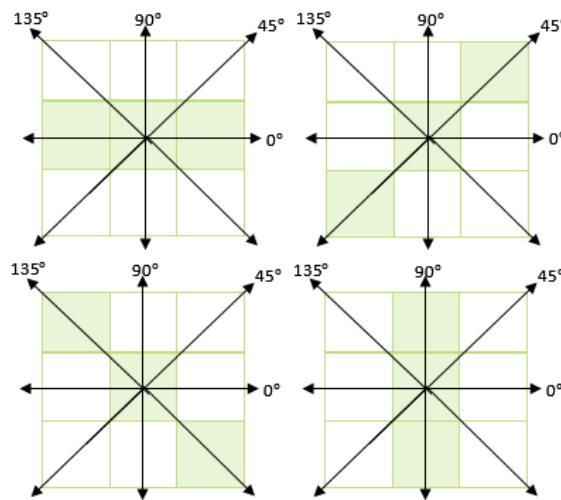
G = nilai representasi dari *Green*

B = nilai representasi dari *Blue*

### 2.5 Gray Level Co-Occurrence Matrix (GLCM)

GLCM merupakan salah satu metode dalam ekstraksi fitur. GLCM didapatkan dari perhitungan tekstur dengan dua orde. Orde pertama berasal dari hasil perhitungan statistik berdasarkan nilai piksel citra asli dan tidak memperhatikan piksel ketetanggaan, sedangkan pada orde kedua adalah hasil perhitungan dari pasangan dua piksel citra asli (Kadir & Susanto, 2012).

Bentuk dari GLCM adalah matriks yang didapatkan dari histogram tingkat dua. matriks yang didapatkan merupakan gambaran dari frekuensi munculnya pasangan piksel pada jarak  $d$  dan orientasi arah dengan sudut  $\theta$  dalam citra yang berfungsi dalam menghitung fitur-fitur GLCM. Penggunaan jarak  $d$  adalah berbentuk piksel sedangkan orientasi sudut digambarkan dalam bentuk derajat (Zulpe & Pawar., 2012). Elemen-elemen *Gray Level Co-occurrence Matrix* mempunyai jumlah pasangan piksel yang memiliki tingkat kecerahan tertentu, di mana pasangan piksel itu terpisah dengan jarak  $d$ , dan dengan suatu sudut inklinasi  $\vartheta$ . Dengan kata lain, matriks kookurensi adalah probabilitas munculnya gray level  $i$  dan  $j$  dari dua piksel yang terpisah pada jarak  $d$  dan sudut  $\vartheta$ . Suatu piksel yang bertetanggaan memiliki  $d$  diantara keduanya, dapat terletak didelapan arah yang berlainan, hal ini ditunjukkan Gambar 2.3.



**Gambar 2.3** Ketetanggan *pixel* sebagai fungsi orientasi sudut dan jarak

Langkah-langkah metode GLCM

1. Normalisasi, yaitu proses mengubah rentang nilai intensitas *pixel*. Dengan rumus *min-max normalization*:

$$\text{Min-max normalization} = (I - \text{Min}) \frac{\text{newMax} - \text{newMin}}{\text{Max} - \text{Min}} + \text{newMin} \quad (2.2)$$

Keterangan:

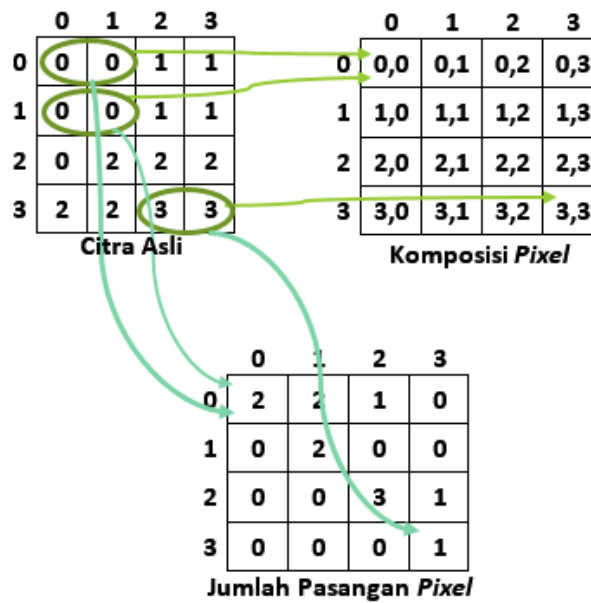
- I* : nilai piksel.
- Max* : nilai maksimal variabel *I*.
- Min* : nilai minimum variabel *I*.
- newMax* : nilai baru maksimal variabel *I*.
- newMin* : nilai baru minimum variabel *I*.

2. Membuat area kerja matriks, misalkan sebuah matriks memiliki empat arah keabuan, maka jumlah nilai piksel tetangga dan nilai piksel referensi pada area kerja matriks berjumlah empat. Berikut ditunjukkan pada Tabel 2.4 contoh area kerja matriks.

**Tabel 2.3** Area kerja matriks

Nilai piksel Tetangga / Nilai piksel referensi	0	1	2	3
0	0,0	0,1	0,2	0,3
1	1,0	1,1	1,2	1,3
2	2,0	2,1	2,2	2,3
3	3,0	3,1	3,2	3,3

3. Menentukan hubungan spasial antara piksel referensi dengan piksel tetangga, berapa nilai sudut  $\theta$  dan jarak  $d$ . Kemudian menghitung jumlah kookurensi dan mengisikannya pada area kerja matriks. Misalnya sudut 0 dan jarak 1:



Gambar 2.4 Ilustrasi penentuan nilai awal matriks GLCM

4. *Symmetric*, diartikan sebagai kemunculan posisi piksel yang sama. Misalkan terdapat piksel (2,3). Maka secara orientasi horizontal piksel (2,3) sama dengan piksel (3,2). Oleh karena itu, *symmetric* adalah hasil penjumlahan matriks kookurensi dengan matriks *transpose* nya sendiri. Pembentukan matriks GLCM yang simetris dapat dilihat pada Gambar 2.5.

$$\begin{bmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

Gambar 2.5 Hasil pembentukan GLCM simetris

5. Normalisasi matriks untuk mengubahnya ke bentuk probabilitas. Nilai GLCM dinormalisasikan agar tidak menyebabkan ketergantungan terhadap ukuran citra, sehingga apabila dijumlahkan maka nilainya adalah 1.

$$\begin{bmatrix} \frac{4}{24} & \frac{2}{24} & \frac{1}{24} & \frac{0}{24} \\ \frac{2}{24} & \frac{4}{24} & \frac{0}{24} & \frac{0}{24} \\ \frac{1}{24} & \frac{0}{24} & \frac{6}{24} & \frac{1}{24} \\ \frac{0}{24} & \frac{0}{24} & \frac{1}{24} & \frac{2}{24} \end{bmatrix} \rightarrow \begin{bmatrix} 0.1667 & 0.0833 & 0.0416 & 0 \\ 0.0833 & 0.1667 & 0 & 0 \\ 0.0416 & 0 & 0.25 & 0.0416 \\ 0 & 0 & 0.0416 & 0.0833 \end{bmatrix}$$

Gambar 2.6 Normalisasi matriks dengan probabilitas



Selanjutnya pengukuran nilai tekstur didasarkan pada persamaan Haralick (1973) yang didefenisikan sebagai berikut:

1. Momen Angular Kedua (*Angular Second Moment*)

ASM digunakan untuk mengukur tentang keseragaman atau sering disebut angular second moment. ASM merupakan ukuran homogenitas citra yang dapat dihitung seperti pada Persamaan 2.3.

$$ASM = \sum_{i,j=0}^{N_g} Pd(i,j)^2 \quad (2.3)$$

Keterangan:

$N_g$  : nilai level keabuan.

$i$  dan  $j$  : nilai sifat keabuan dari resolusi 2 piksel yang berdekatan.

$Pd(i,j)$  : nilai pada baris  $i$  dan kolom  $j$  pada matriks kookurensi yang sudah ternormalisasi.

2. Kontras (*Contrast*)

Kontras digunakan untuk menunjukkan ukuran penyebaran (*MomentInertia*) elemen-elemen matriks citra yang dapat dihitung seperti pada Persamaan 2.4.

$$Contrast = \sum_{i,j=0}^{N_g} (i - j)^2 Pd(i,j) \quad (2.4)$$

Keterangan:

$N_g$  : nilai level keabuan.

$i$  dan  $j$  : nilai sifat keabuan dari resolusi 2 piksel yang berdekatan.

$Pd(i,j)$  : nilai pada baris  $i$  dan kolom  $j$  pada matriks kookurensi yang sudah ternormalisasi.

3. Korelasi (*Correlation*)

Korelasi digunakan untuk menunjukkan ukuran ketergantungan linear derajat keabuan citra sehingga dapat memberikan petunjuk adanya struktur linear dalam citra yang dapat dihitung seperti pada Persamaan 2.3.

$$Correlation = \sum_{i,j=0}^{N_g} \frac{(i - \mu_i)(j - \mu_j) \cdot Pd(i,j)}{\sigma_i \cdot \sigma_j} \quad (2.5)$$

Keterangan:

$N_g$  : nilai level keabuan.

$i$  dan  $j$  : nilai sifat keabuan dari resolusi 2 piksel yang berdekatan.

$Pd(i,j)$  : nilai pada baris  $i$  dan kolom  $j$  pada matriks kookurensi yang sudah ternormalisasi.

$\mu_i$  : nilai rata-rata elemen kolom pada matriks  $Pd(i,j)$ .

$\mu_j$  : nilai rata-rata elemen baris pada matriks  $Pd(i,j)$ .

$\sigma_i$  : nilai simpangan baku elemen kolom pada matriks.

$\sigma_j$  : nilai simpangan baku elemen baris pada matriks.

4. Momen Selisih Terbalik (*Inverse Difference*)

IDM digunakan untuk mengukur homogenitas yang dapat dihitung seperti pada Persamaan 2.3.

$$IDM = \sum_{i,j=0}^{N_g} \frac{1}{1+(i-j)^2} Pd(i,j) \tag{2.6}$$

Keterangan:

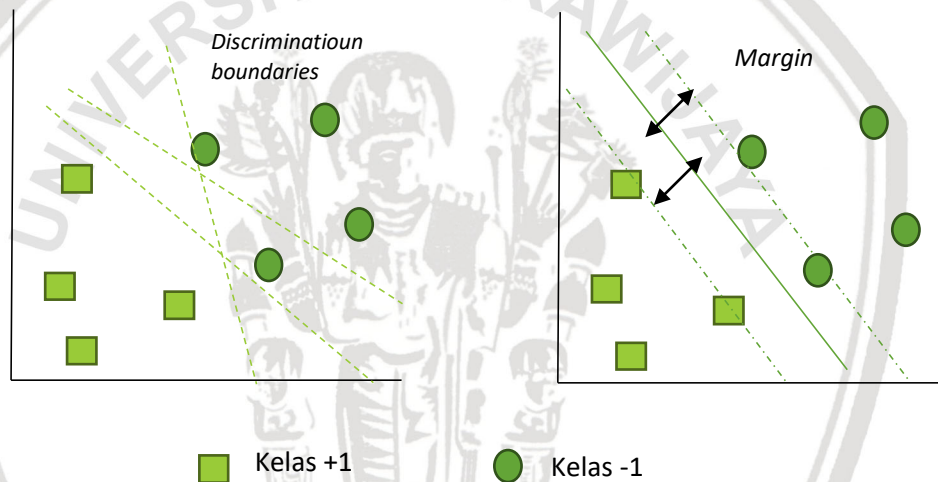
$N_g$  : Nilai Level Keabuan.

$i$  dan  $j$  : Nilai sifat keabuan dari resolusi 2 piksel yang berdekatan.

$Pd(i,j)$  : Nilai pada baris  $i$  dan kolom  $j$  pada matriks kookurensi yang sudah ternormalisasi.

## 2.6 Support Vector Machine

*Support Vector Machine* merupakan salah satu metode dari klasifikasi, metode pembelajaran yang digunakan adalah terbimbing. Algoritme yang diciptakan oleh Vladimir Vapnik ini berfungsi dalam menganalisis data dan mengenali pola agar mendapatkan hasil klasifikasi (Lidya, Sitompul & Effendi.,2015). Konsep sederhana dari Algoritme SVM adalah mencari *hyperplane* atau garis pembatas terbaik yang berfungsi untuk memisahkan dua kelas pada *input space*. Contoh pemisahan kelas pada SVM ditunjukkan pada Gambar 2.7.



**Gambar 2.7 Hyperplane SVM**

Sumber : Nugroho, Witarto, Handoko (2003)

Pada Gambar 2.7 telah ditunjukkan dua buah kelas yaitu +1, yang menunjukkan kelas positif dan -1, menunjukkan kelas negatif. Untuk mendapatkan garis *hyperplane* terbaik digunakan perhitungan *margin hyperplane* dan mencari titik maksimal. Dalam membagi dua buah kelas dengan *hyperplane* terbaik, dapat didefinisikan seperti pada Persamaan 2.7:

$$(w \cdot x_i) + b = 0 \tag{2.7}$$

Di dalam pattern  $x_i$  yang termasuk pada *class* -1 dapat dirumuskan seperti pada Pertidaksamaan 2.8.

$$(w \cdot x_i + b) \leq -1, y_i = -1 \tag{2.8}$$

Sedangkan pattern  $x_i$  yang termasuk pada *class* +1 dapat dirumuskan seperti pada Pertidaksamaan 2.9.

$$w \cdot x_i + b \geq 1, y_i = 1 \tag{2.9}$$

Keterangan:

$x_i$  : Data ke- $i$ .

$y_i$  : Kelas data ke- $i$ .

$w$  : Vektor tegak lurus terhadap *hyperplane*.

$b$  : Nilai bias.

Parameter support vektor merupakan sebuah garis vektor yang tegak lurus dengan titik pusatnya dan garis *hyperplane*. Nilai untuk parameter bias adalah nilai dari koordinat garis yang relatif terhadap titik koordinatnya. Dalam persamaan 2.10 dan persamaan 2.11 akan dijelaskan persamaan untuk mendapatkan nilai parameter  $w$  dan  $b$ .

$$b = -\frac{1}{2}(w \cdot x^+ + w \cdot x^-) \tag{2.10}$$

$$w = \sum_{i=1}^n \alpha_i y_i K(x_i, x) \tag{2.11}$$

Keterangan:

$b$  = Nilai untuk parameter bias.

$w \cdot x^+$  = Nilai bobot yang ada dalam kelas data yang bernilai positif.

$w \cdot x^-$  = Nilai bobot yang ada dalam kelas data yang bernilai negatif.

$w$  = Nilai untuk parameter bobot support vector.

$\alpha_i$  = Nilai parameter alpha.

$y_i$  = Sebuah kelas yang ada pada data ke- $i$ .

$K(x_i, x)$  = Hasil dari nilai *kernel*.

### 2.6.1 Support Vector Machine Non-Linear

Dalam masalah klasifikasi kebanyakan sampel data tidak terpisah secara linier sehingga jika digunakan SVM linier maka hasil yang diperoleh tidak optimal dan mengakibatkan hasil klasifikasi yang buruk. SVM linier dapat diubah menjadi SVM non-linier dengan menggunakan menambahkan fungsi *kernel*. Metode ini bekerja dengan cara memetakan data input ke ruang *feature* yang dimensinya lebih tinggi. Diharapkan data input hasil pemetaan ke ruang *feature* akan terpisah secara linier sehingga dapat dicari *hyperplane* yang optimal. Berikut beberapa fungsi *kernel* yang sering digunakan (Prasetyo, 2012).

1. Kernel Linier

$$K(x, y) = x \cdot y \tag{2.12}$$

2. Kernel Polinomial of degree  $d$

$$K(x, y) = x \cdot y^d \tag{2.13}$$

3. Kernel Polinomial of degree up to  $d$

$$K(x, y) = (x \cdot y + c)^d \tag{2.14}$$

### 2.6.2 Sequential Training SVM

Menurut Vijayakumar & Wu (1999) metode ini dikembangkan untuk klasifikasi, langkah-langkah *sequential learning* adalah sebagai berikut :



- a. Melakukan inialisasi terhadap  $\alpha_i=0$  dan parameter lainnya, seperti nilai  $\lambda, \gamma, C$  dan iterasi maksimal.

Keterangan:

$\alpha_i$  = alfa ke-i, digunakan untuk mencari support vector.

$\gamma$  = konstanta *gamma* untuk mengontrol kecepatan.

$C$  = konstanta *C* untuk membatasi nilai alfa.

- b. Menghitung matriks Hessian menggunakan persamaan 2.16.

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (2.16)$$

Dengan  $i, j = 1, \dots, n$ .

Keterangan:

$x_i$  = data ke-i

$x_j$  = data ke-j

$y_i$  = kelas data ke-i

$y_j$  = kelas data ke-j

$n$  = jumlah data

$\lambda$  = nilai *Lambda*

$K(x_i, x_j)$  = fungsi *Kernel* yang digunakan

- c. Melakukan iterasi untuk setiap iterasi yang telah di inialisasi di awal, lalu menghitung nilai  $E_i$  yang dihitung menggunakan persamaan 2.17.

$$E_i = \sum_{j=1}^n \alpha_j D_{ij} \quad (2.17)$$

Keterangan:

$\alpha_i$  = alfa ke-j

$E_i$  = *Error rate*

$D_{ij}$  = *Matriks Hessian*

- d. Menghitung nilai  $\delta\alpha_i$  yang dihitung menggunakan persamaan 2.18.

$$\delta\alpha_i = \min(\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i) \quad (2.18)$$

Keterangan:

$\alpha_i$  = alfa ke-i

$E_i$  = *Error rate*

$C$  = Konstanta *C*

$\gamma$  = konstanta *gamma*

Memperbarui nilai  $\alpha_i$  yang dihitung menggunakan persamaan 2.19.

$$\alpha_i = \alpha_i + \delta\alpha_i \quad (2.19)$$

Keterangan:

$\alpha_i$  = alfa ke-i

$\delta\alpha_i$  = delta alfa ke-i

- e. Kembali ke langkah ketiga dan lakukan secara berulang-ulang sampai mendapatkan iterasi maksimum atau  $\max(|\delta\alpha_i|) < \epsilon$ .

### 2.6.3 One Against All

Dengan menggunakan metode ini, dibangun k buah model SVM biner ( $k$  adalah jumlah kelas). Setiap model klasifikasi ke- $i$  dilatih dengan menggunakan keseluruhan data, untuk mencari solusi dalam permasalahan. Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Untuk pelatihan

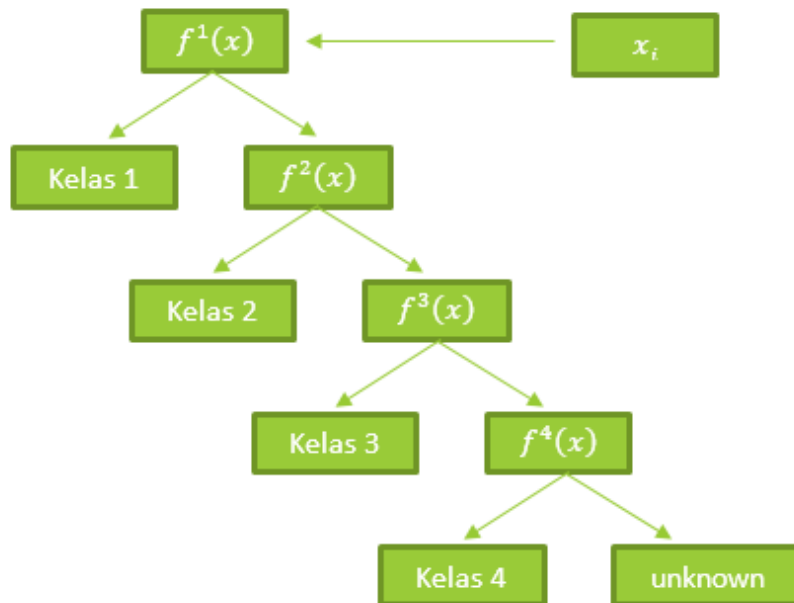


digunakan 4 buah SVM biner dan penggunaannya dalam mengklasifikasi data baru dapat dilihat pada Tabel 2.7.

**Tabel 2.4 Contoh SVM dengan *One Against All***

$y_i = 1$	$y_j = -1$	Hipotesis
Kelas 1	Bukan kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan kelas 2	$f^2(x) = (w^2)x + b^2$
Kelas 3	Bukan kelas 3	$f^3(x) = (w^3)x + b^3$
Kelas 4	Bukan kelas 4	$f^4(x) = (w^4)x + b^4$

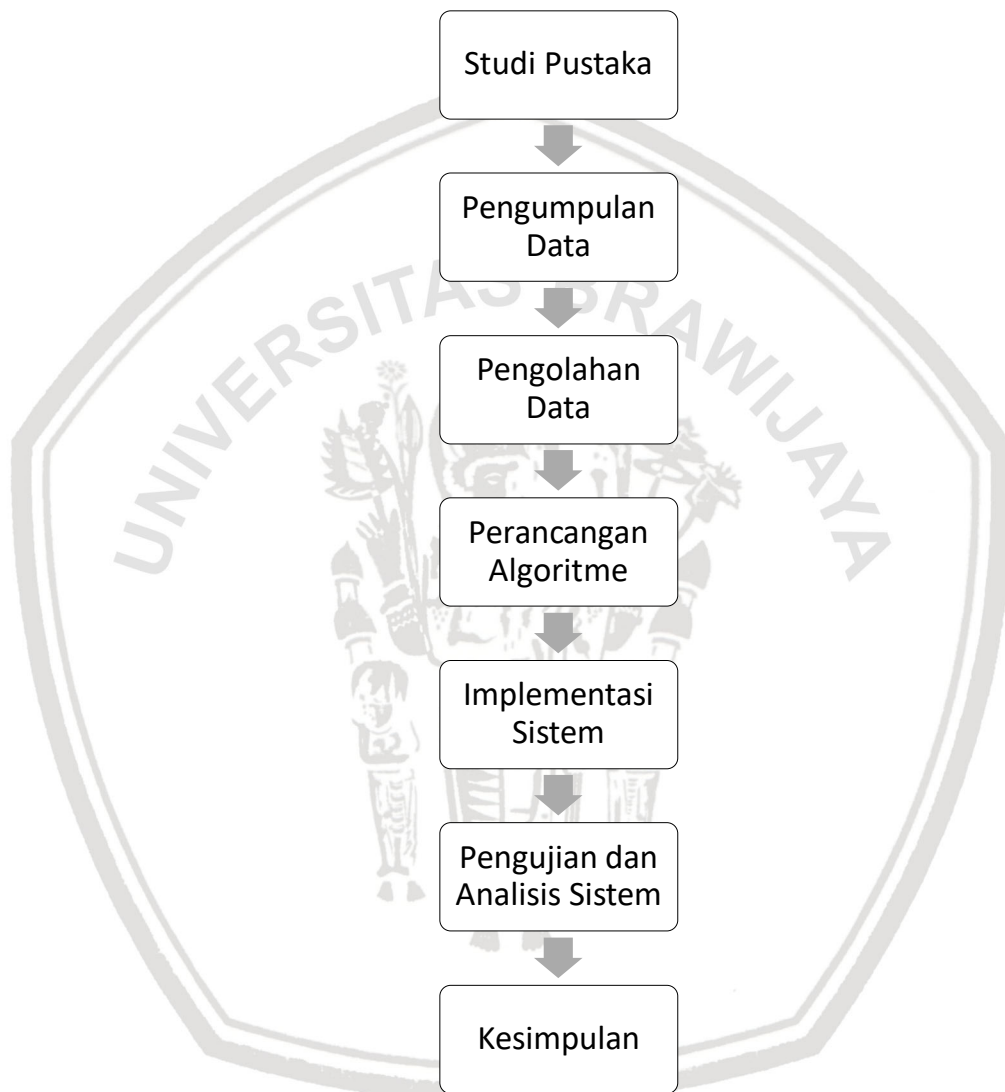
Pada Gambar 2.8 dapat dilihat bahwa ketika proses *training* dilakukan terhadap  $f^1(x)$ , semua data dalam kelas 1 diberi label +1 dan data yang lainnya selain dari kelas 1 yaitu kelas 2 diberi label - 1. Begitu pula ketika proses latih dilakukan terhadap  $f^2(x)$ , semua data dalam kelas tersebut diberi label +1 dan data selain yang berada di kelas tersebut diberi label -1. Begitu terus selanjutnya untuk semua  $i=1, 2, 3, 4$ .



**Gambar 2.8 Contoh SVM dengan *One Against All***

## BAB 3 METODOLOGI

Pada bab ini berisi urutan langkah-langkah yang harus dilakukan dalam menyelesaikan penelitian pembuatan sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine*. Adapun diagram blok tahapan metode dalam penelitian ini, ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Blok Tahapan Penelitian

### 3.1 Studi Pustaka

Dalam tahapan penelitian ini pertama yaitu studi pustaka yang dilakukan dengan mencari, mendapatkan, dan mengumpulkan dasar-dasar teori dan sumber acuan mengenai bencana, odontologi forensik, metode *Gray Level Co-occurrence Matrix* dan metode *Support Vector Machine*. Penulis melakukan pencarian referensi di buku, skripsi yang sebelumnya, *paper* dan internet.

### 3.2 Pengumpulan Data

Pada penelitian ini proses pengumpulan data yang dilakukan yaitu dengan mengambil beberapa foto bibir dari mahasiswa dan mahasiswi Fakultas Ilmu Komputer, Universitas Brawijaya. Data yang digunakan dalam penelitian ini adalah hasil foto bibir mahasiswa dan mahasiswi Fakultas Ilmu Komputer Universitas Brawijaya baik pria maupun wanita, dengan ketentuan bibir dalam kondisi normal, dalam posisi istirahat dan tidak menggunakan pewarna bibir yang mengkilap. Lokasi pengambilan data bertempat di ruang kelas dan diluar kelas Fakultas Ilmu Komputer, Universitas Brawijaya. Perangkat yang digunakan dalam pengumpulan data citra bibir yaitu kamera dari Iphone 5s dengan spesifikasi kamera 8MP (belakang), Dual LED, F/2.2. Format yang digunakan pada citra bibir yaitu *Joint Photographic Group* (.jpg). Jarak pengambilan foto kurang lebih 20cm dari objek. Jumlah citra bibir yang diambil sebanyak 60, untuk citra bibir pria diambil sebanyak 30 dan wanita sebanyak 30. Setiap citra mewakili 1 orang yang berarti semua citra bibir didapatkan dari orang yang berbeda (1 foto bibir mewakili 1 orang). Gambar atau citra mentah kemudian dilakukan *cropping* menggunakan aplikasi Adobe Photoshop Express, yaitu agar citra yang diolah dipersempit objeknya, yaitu fokus dibagian bibir saja. Pada Gambar 3.2 dan Gambar 3.3 ditunjukkan citra bibir sebelum dan setelah dilakukan *cropping*.



**Gambar 3.2** Citra bibir sebelum di lakukan *cropping*






**Gambar 3.3** Citra bibir setelah dilakukan *cropping*

### 3.3 Pengolahan Data

Tahap pengolahan data meliputi data latih dan data uji. Dataset yang telah didapatkan yaitu citra bibir akan dilakukan *cropping* dengan ukuran 400x200 pixel. *Cropping* citra menggunakan aplikasi Adobe Photoshop Express. Setelah itu

citra bibir dipotong lagi menjadi 8 bagian, yaitu dengan masing-masing ukuran 100x100 pixel. Tujuan pemotongan menjadi 8 bagian adalah agar citra bibir terbagi menjadi kuadran-kuadran yang kemudian setiap kuadran tersebut akan dilabeli sesuai dengan kelas klasifikasinya. Kuadran citra yang akan digunakan sebagai data skripsi ini hanya 4 kuadran saja, kuadran yang berada di bagian ujung bibir tidak digunakan karena alur garis bibir (sulci) kurang terlihat garis-garisnya. Setelah itu pelabelan kelas dilakukan secara manual dengan memperhatikan sulci setiap kuadran berdasarkan pola sidik bibir dari Suzuki dan Tsuchihashi. Jika pada kuadran terdapat beberapa pola, maka akan diambil pola yang paling dominan dan ditetapkan sebagai kelas. Teknik yang dilakukan penulis dalam menentukan kelas yaitu citra bibir dicetak, kemudian menetapkan alur dari pola sidik bibir dengan cara mencoret garis-garis sulci menggunakan spidol merah.

**Tabel 3.1 Tahapan Pengolahan Data**

Tahapan Pengolahan Data	Keterangan
<p>Langkah 1 :</p> 	<p>Citra yang telah dilakukan <i>cropping</i> dengan ukuran 400x200 pixel.</p>
<p>Langkah 2 :</p> 	<p>Gambar yang telah dilakukan <i>cropping</i> menjadi 8 bagian dengan ukuran 100x100 pixel. Urutan kuadran dari kiri atas sampai kanan bawah.</p>
<p>Langkah 3 :</p> 	<p>Mengambil gambar citra pada bagian tengah saja, yaitu kuadran 2,3,5,dan 6.</p>

Tabel 3.2 Tahapan Pengolahan Data (lanjutan)

<p>Langkah 4 :</p> 	<p>Mencetak citra yang dijadikan sebagai dataset dan melakukan pelabelan dengan memperhatikan alur pola yang terdapat pada citra bibir.</p>
<p>Langkah 5 :</p> 	<p>Melakukan pelabelan kelas dengan memberikan nama disetiap file citra. Untuk tipe I' adalah kelas 1, tipe I yaitu kelas 2, tipe II untuk kelas 3, sampai tipe V untuk kelas 6. Nama file dari citra akan diberi nama dengan urutan tipe-nomerbibir dan kuadran bibir.</p>

### 3.4 Perancangan Algoritme

Tahap perancangan sistem bertujuan untuk menganalisis kebutuhan apa saja yang dibutuhkan dalam pembuatan sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine* untuk mempermudah Penulis dalam melakukan tahap selanjutnya dan agar menghasilkan sistem yang optimal. Perancangan yang dibuat yaitu berupa perancangan Algoritme. Perancangan Algoritme terdiri dari beberapa tahap yakni *Pre-Processing*, *GLCM*, dan *SVM*. Data inputan yang digunakan adalah citra bibir, kemudian dilakukan preprocessing citra bibir dan dilakukan proses ekstraksi ciri menggunakan *GLCM*. Sedangkan hasil keluaran (*output*) dari sistem ini adalah data hasil klasifikasi jenis kelamin pria dan wanita berdasarkan sidik bibir.



Gambar 3.4 Tahapan-tahapan dari sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia

### 3.5 Implementasi Sistem

Tahap implementasi sistem ini yaitu dengan menerapkan metode GLCM dan *Support Vector Machine* dalam permasalahan klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan perancangan sistem yang telah dirancang. Tahapan dari implementasi meliputi:

- Implementasi algoritme untuk melakukan perhitungan menggunakan metode Gray Level Co-occurrence dan Support Vector Machine.
- Keluaran dari sistem yaitu hasil klasifikasi jenis kelamin pada pria dan wanita.

### 3.6 Pengujian dan Analisis Sistem

Pada tahap pengujian sistem ini Penulis menguji sistem sesuai dengan parameter Algoritme *Support Vector Machine* dan parameter Algoritme GLCM, yaitu antara lain pengujian terhadap parameter SVM dan parameter jarak (*distance*) pada GLCM. Pengujian dilakukan agar dapat membuktikan bahwa hasil sistem tersebut telah mampu bekerja dengan baik sesuai dengan kebutuhan dan mengetahui seberapa baik akurasi yang didapatkan dalam penerapan Algoritme GLCM dan SVM dalam menyelesaikan permasalahan klasifikasi jenis kelamin pria dan wanita berdasarkan sidik bibir.

### 3.7 Kesimpulan

Kesimpulan diambil dari hasil pengujian terhadap sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *Gray Level Co-occurrence Matrix* dan *Support Vector Machine* yang telah diterapkan. Tahapan terakhir yang dilakukan adalah menuliskan saran. Penulisan saran ini dimaksudkan agar penelitian berikutnya bisa memperbaiki kekurangan-kekurangan yang ada serta memberikan pertimbangan atas pengembangan metode selanjutnya.

## BAB 4 PERANCANGAN

Pada bab ini akan dibahas mengenai formulasi permasalahan, perancangan algoritme GLCM dan SVM, perhitungan manual algoritme GLCM dan SVM dan perancangan antarmuka *user* serta perancangan pengujian sistem.

### 4.1 Formulasi Permasalahan

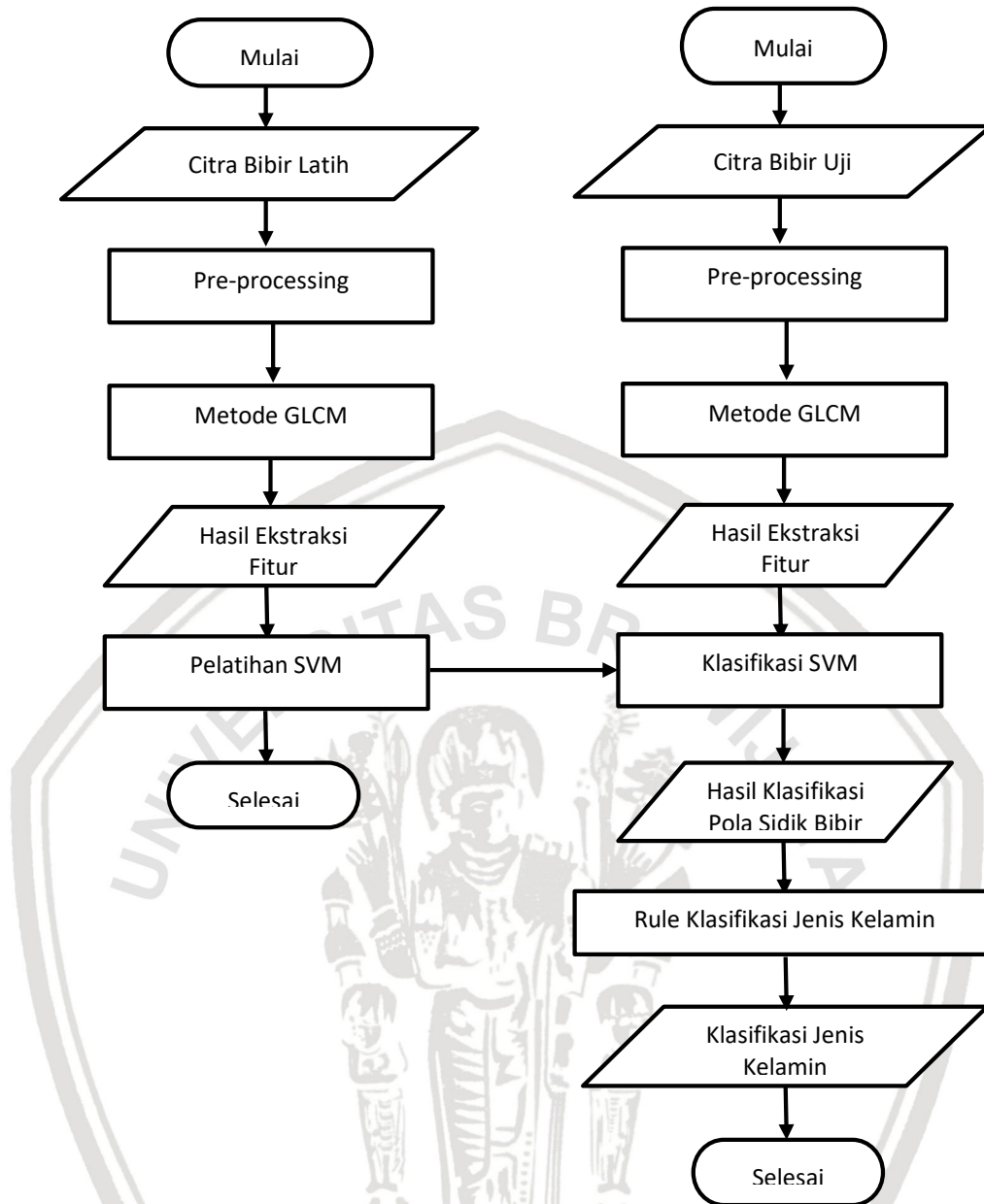
Penelitian ini menyelesaikan permasalahan mengenai klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan 6 kelas yaitu tipe I, tipe I', tipe II, tipe III, tipe IV, tipe V. Untuk tipe I' dianggap sebagai kelas ke 1, tipe I dianggap sebagai kelas ke 2, tipe II dianggap sebagai kelas ke 3, tipe III dianggap sebagai kelas ke 4, tipe IV dianggap sebagai kelas ke 5, dan tipe V dianggap sebagai kelas ke 6. Proses identifikasi dilakukan dengan menggunakan metode SVM yang sebelumnya data masukan dilakukan ekstraksi fitur terlebih dahulu menggunakan metode GLCM. Jumlah atribut untuk setiap data adalah 16 atribut. Nilai-nilai fitur pada SVM didapatkan dari ekstraksi fitur citra dengan 4 fitur yaitu, *Angular Second Moment (ASM)*, *Contrast*, *Correlation*, *Inverse Difference Moment (IDM)* dengan masing-masing sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ .

**Tabel 4.1 Dataset Nilai Citra Sidik Bibir**

Data	Nama	ASM $0^\circ$	CON $0^\circ$	COR $0^\circ$	IDM $0^\circ$	...	COR $135^\circ$	IDM $135^\circ$	Tipe Bibir	Kelas
1	TI'B13K3W	0,849	0,016	0,321	0,992	...	0,286	0,988	I'	1
2	TI'B1K3W	0,615	0,018	0,381	0,991	...	0,334	0,980	I'	1
3	TIB2K7W	0,624	0,068	0,403	0,966	...	0,369	0,958	I	-1
4	TIB3K3L	0,601	0,026	0,372	0,987	...	0,358	0,978	I	-1
5	TIIB11K2L	0,845	0,061	0,157	0,969	...	0,125	0,965	II	-1
6	TIIB2K2W	0,422	0,075	0,301	0,963	...	0,292	0,950	II	-1
7	TIIB4K2L	0,375	0,142	0,279	0,929	...	0,282	0,927	III	-1
8	TIIB4K6L	0,423	0,097	0,285	0,951	...	0,281	0,947	III	-1
9	TIVB5K6L	0,677	0,051	0,407	0,975	...	0,370	0,966	IV	-1
10	TIVB9K6L	0,483	0,084	0,410	0,958	...	0,370	0,941	IV	-1
11	TVB11K6L	0,433	0,089	0,271	0,956	...	0,267	0,948	V	-1
12	TVB13K3L	0,618	0,029	0,485	0,985	...	0,467	0,981	V	-1

### 4.2 Perancangan Algoritme

Perancangan Algoritme digambarkan dengan *flowchart* (diagram alir) yang menunjukkan alur proses dari sistem. Fungsi dari pembuatan *flowchart* adalah untuk memudahkan dalam memahami alur proses yang akan dibuat. Perancangan Algoritme yang dibuat yaitu proses algoritme GLCM dan proses algoritme SVM. Untuk perancangan secara keseluruhan disusun dalam diagram alir pada Gambar 4.1.



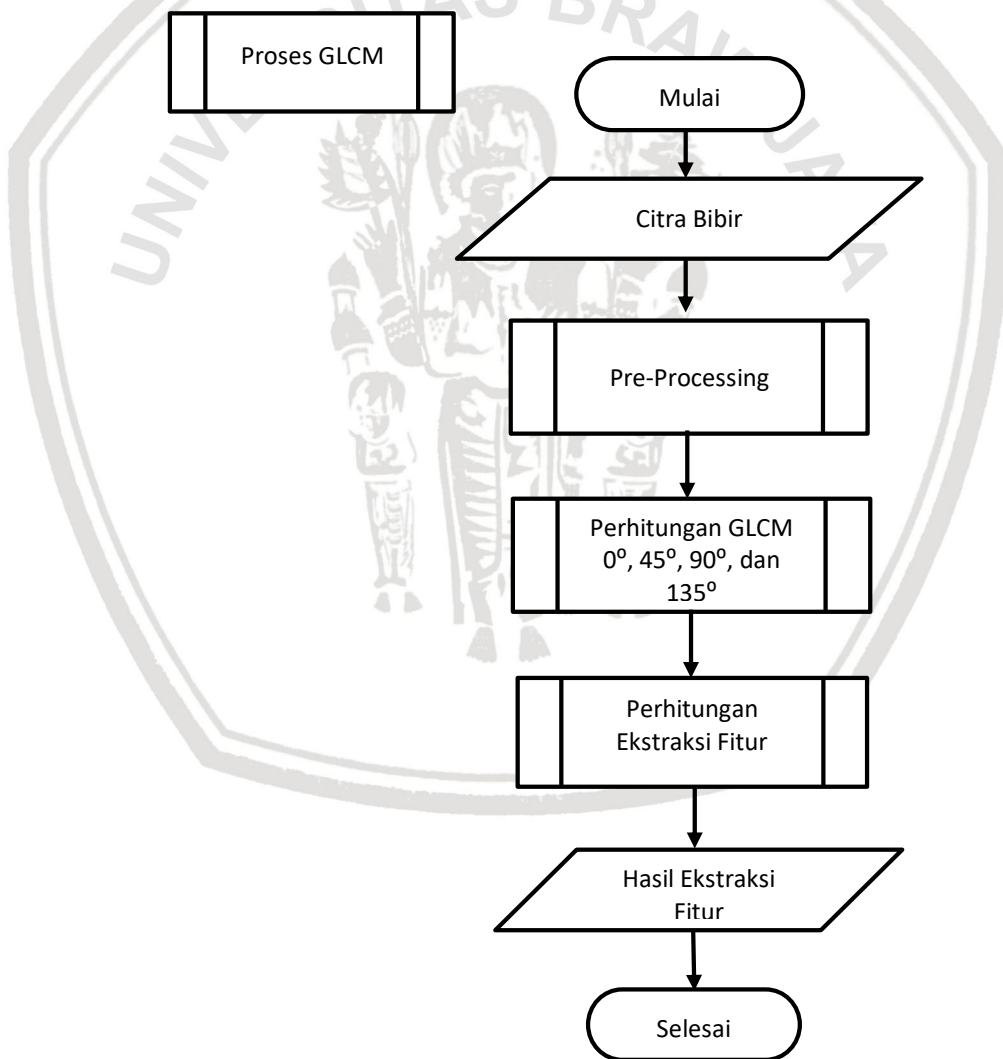
**Gambar 4.1 Diagram Alir Proses Sistem**

Proses diawali dengan pelatihan citra bibir latih yang untuk proses awalnya yaitu memasukkan citra bibir, kemudian melakukan grayscale dan mencari nilai ekstraksi fitur tekstur dengan metode GLCM. Setelah mendapatkan nilai fitur, melakukan pelatihan data dengan metode sequential learning SVM. Untuk citra bibir uji juga dilakukan proses awal yang sama dengan citra latih, yaitu melakukan grayscale dan mencari nilai ekstraksi fitur tekstur dengan metode GLCM. Setelah didapat nilai fitur, melakukan proses klasifikasi SVM yang untuk data *trainingnya* menggunakan data yang sudah dilakukan pelatihan SVM sebelumnya. Setelah proses klasifikasi dilakukan, didapatkan hasil klasifikasi pola sidik bibir dan kemudian untuk menentukan jenis kelamin dengan berdasarkan rule pada Tabel 2.2.



### 4.2.1 Perancangan Proses Algoritme GLCM

Pada proses ini akan dijelaskan mengenai alur dari metode GLCM yang dimulai dari memasukan data berupa citra bibir. Setelah input citra, melakukan perhitungan fitur tekstur GLCM dari yang sebelumnya dilakukan pre-processing terlebih dahulu yaitu mengubah citra dari RGB ke dalam greyscale. Setelah mendapatkan nilai pixel dari citra yang sudah dirubah menjadi greyscale, melakukan normalisasi terhadap nilai matrix pixel citra greyscale. Setelah itu menghitung matriks awal yaitu hubungan spasial nilai matrix citra grayscale dengan sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$  dan jarak=1. Setelah itu melakukan penjumlahan antara matriks awal dengan transposenya untuk mendapatkan hasil matriks *co-occurrence* yang simetris. Hasil matriks simetris kemudian dinormalisasi dengan membagi tiap-tiap nilai pada matriks dengan total keseluruhan nilai pada matriks. Dari hasil matriks yang telah dinormalisasi, dilakukan perhitungan ekstraksi fitur sebanyak 4 fitur yaitu ASM, Contrast, Correlation dan IDM . Diagram alir proses GLCM dapat dilihat pada Gambar 4.2.



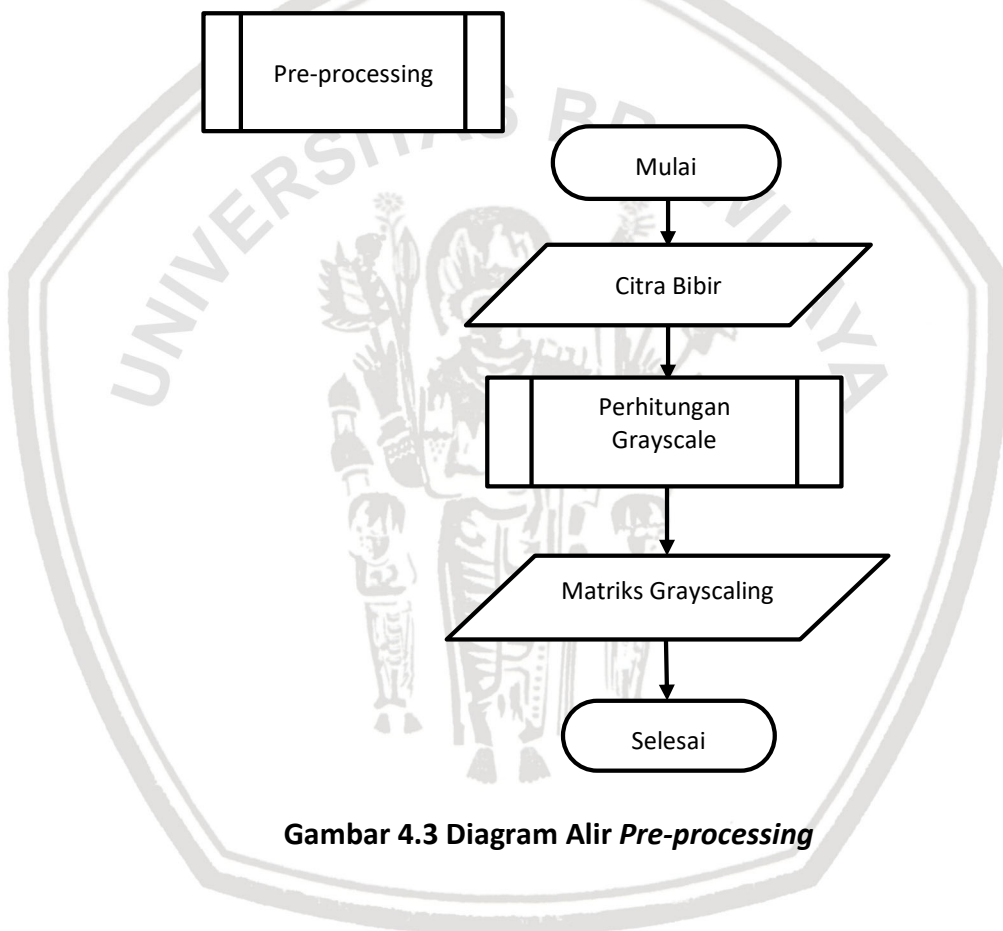
Gambar 4.2 Diagram Alir Proses GLCM

Berikut penjelasan Proses GLCM.

1. Mengambil masukan data berupa citra bibir.
2. Melakukan *Pre-processing*.
3. Melakukan perhitungan GLCM  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$
4. Melakukan perhitungan Ekstraksi Fitur
5. Keluaran yang dihasilkan berupa Hasil Ekstraksi Fitur

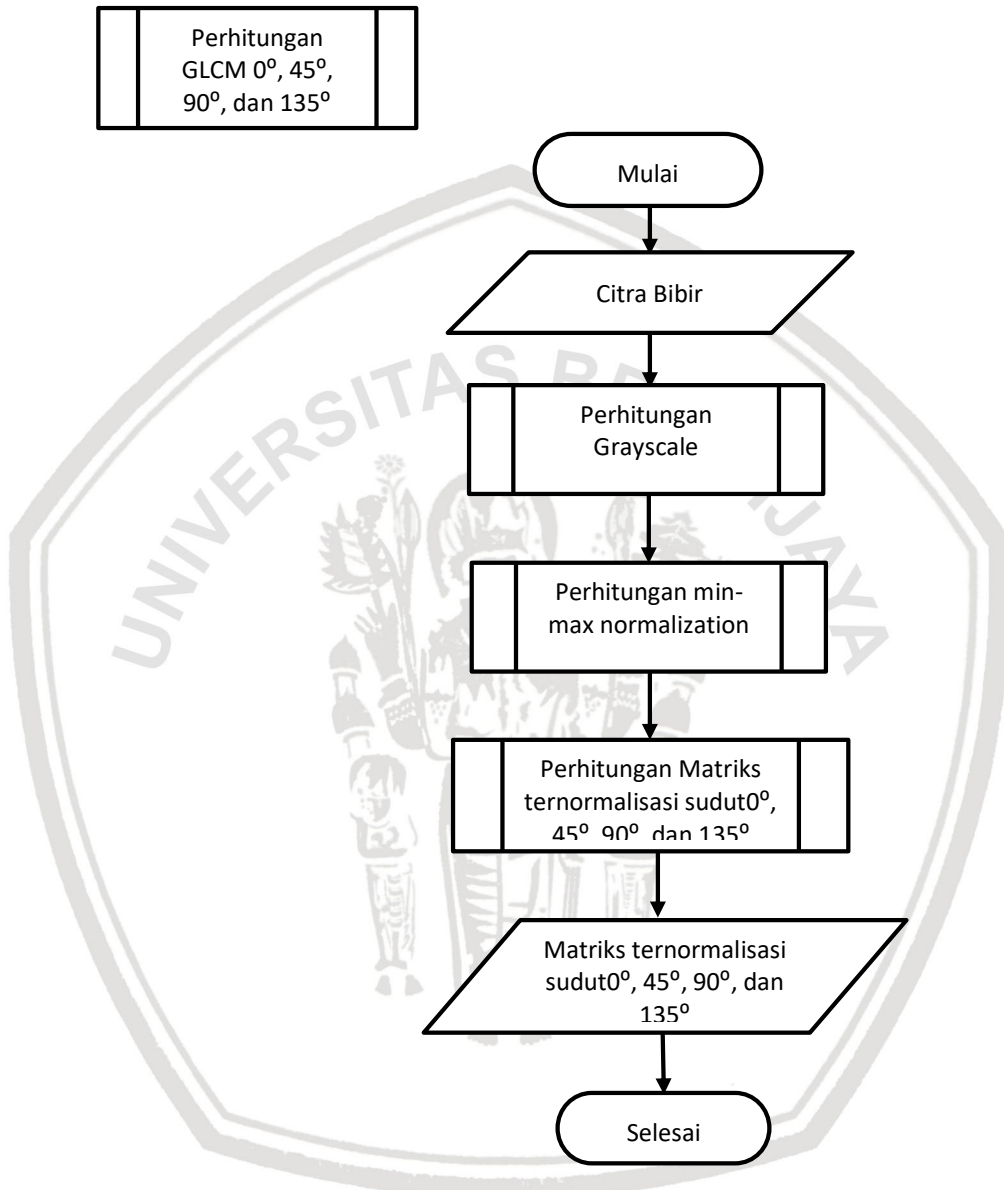
#### 4.2.1.1 *Pre-processing*

*Pre-processing* merupakan langkah awal dalam proses sistem ini. Citra bibir akan dilakukan proses Grayscale agar mendapatkan nilai keabuan dari citra yang kemudian digunakan untuk mencari nilai matriks korelasi dari citra bibir. Untuk tahapan proses *pre-processing* dapat dilihat pada Gambar 4.3 .



#### 4.2.1.2 Perhitungan GLCM 0°, 45°, 90° dan 135°

Perhitungan GLCM 0°, 45°, 90° dan 135° dilakukan untuk mendapatkan nilai matriks ternormalisasi dari keseluruhan data yang nantinya digunakan untuk menghitung nilai fitur GLCM. Untuk tahapan proses perhitungan GLCM 0°, 45°, 90° dan 135° dapat dilihat pada Gambar 4.4.



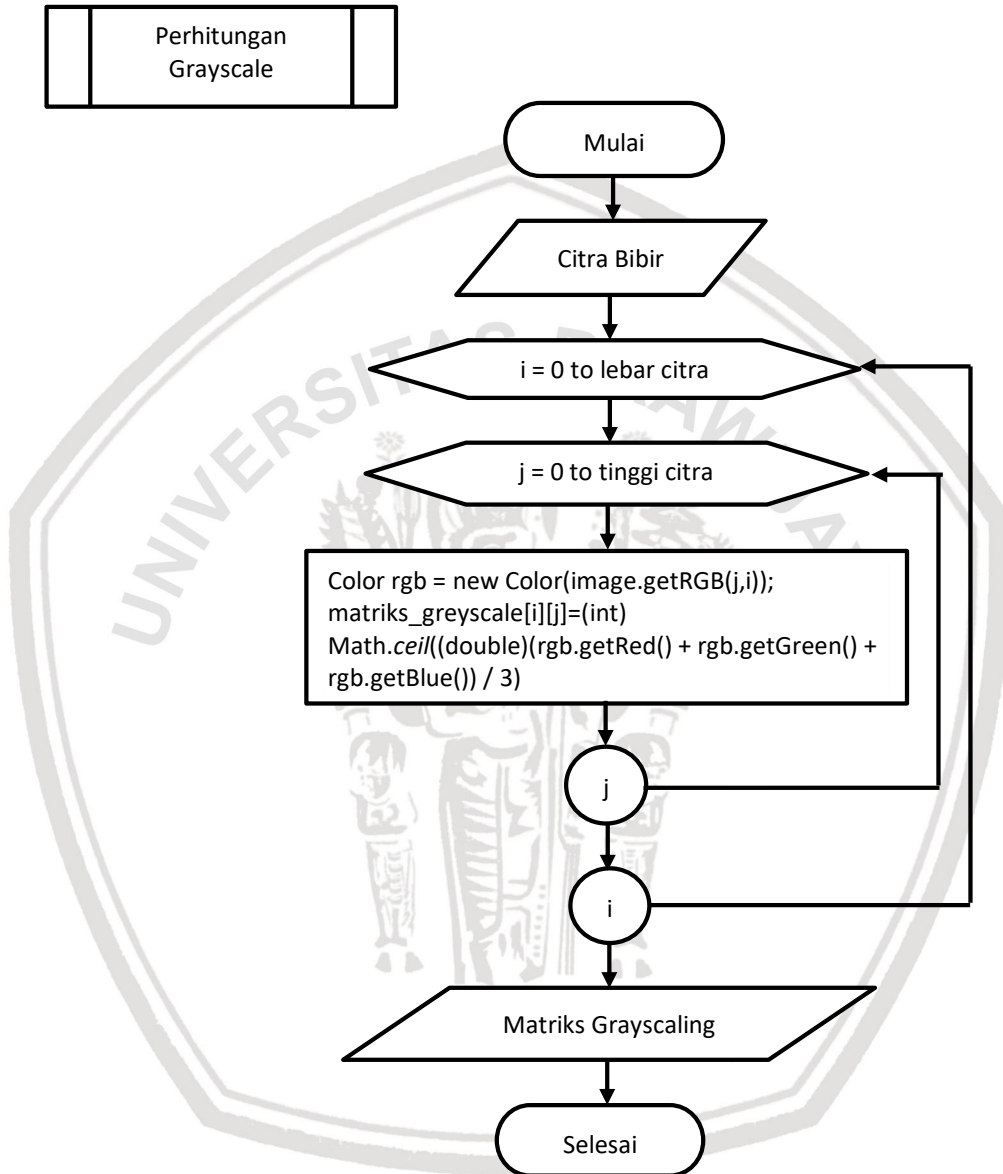
**Gambar 4.4 Diagram Alir Perhitungan GLCM 0°, 45°, 90° dan 135°**

Berikut penjelasan alir perhitungan GLCM 0°, 45°, 90° dan 135°.

1. Mengambil masukan data berupa citra bibir.
2. Melakukan perhitungan Grayscale.
3. Melakukan perhitungan Min-max Normalization.
4. Melakukan perhitungan ternormalisasi sudut 0°, 45°, 90° dan 135°.
5. Keluaran yang dihasilkan berupa nilai Matriks ternormalisasi sudut 0°, 45°, 90° dan 135°.

### 4.2.1.3 Perhitungan Grayscale

Perhitungan grayscale dilakukan untuk mendapatkan nilai keabuan citra dari keseluruhan data yang nantinya digunakan untuk menghitung nilai fitur GLCM. Cara menghitung nilai grayscale yaitu dengan menjumlahkan nilai *red*, *green* dan *blue* kemudian dibagi dengan 3. Untuk tahapan proses perhitungan grayscale dapat dilihat pada Gambar 4.5.



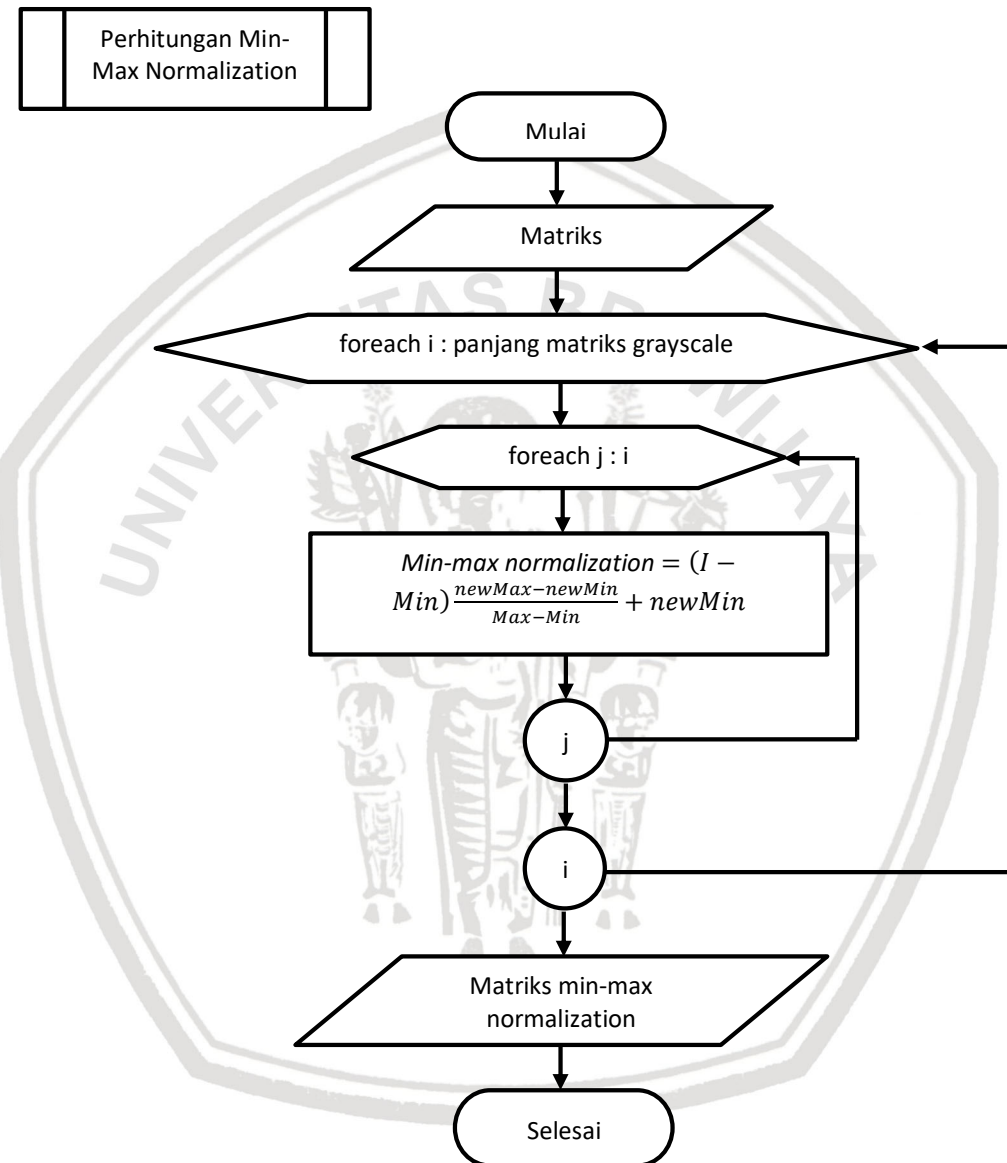
**Gambar 4.5 Diagram Alir Perhitungan Grayscale**

Berikut penjelasan alir perhitungan grayscale.

1. Mengambil masukan Citra Bibir.
2. Melakukan perulangan sebanyak jumlah lebar citra.
3. Melakukan perulangan sebanyak jumlah tinggi citra.
4. Melakukan proses perhitungan Grayscale dengan Persamaan 2.1.
5. Keluaran yang dihasilkan berupa nilai Matriks Grayscale.

#### 4.2.1.4 Perhitungan Min-Max Normalization

Perhitungan min-max normalization dilakukan untuk mendapatkan nilai keabuan citra dengan cara menormalisasi nilai minimal dan maksimal nilai pixel grayscale sesuai dengan nilai range nilai minimal dan nilai maksimal baru. Nilai matriks min-max normalization ini nantinya digunakan untuk menghitung nilai matriks awal. Untuk tahapan proses perhitungan grayscale dapat dilihat pada Gambar 4.6.



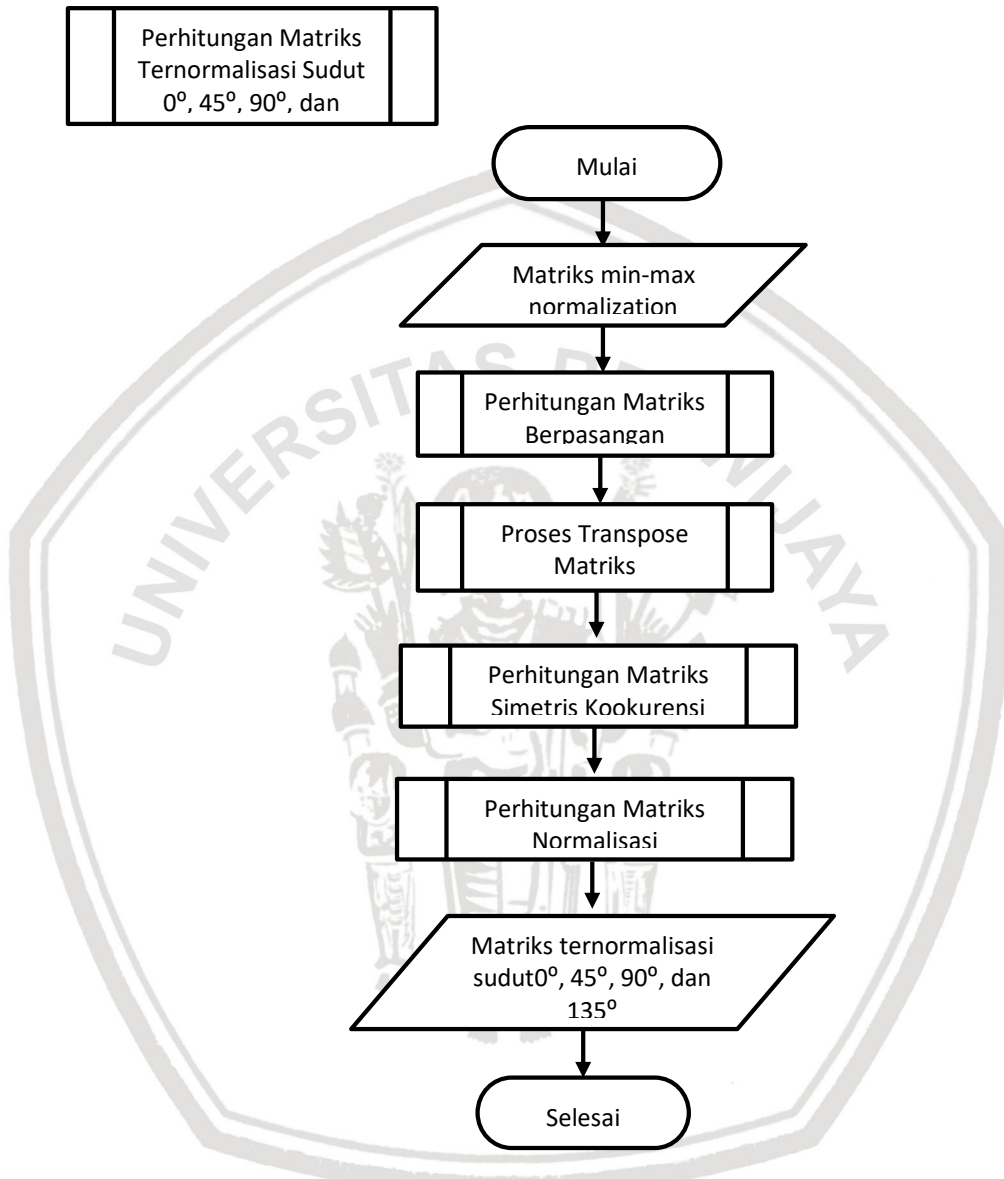
**Gambar 4.6 Diagram Alir Min-Max Normalization**

Berikut penjelasan alir perhitungan min-max normalization.

1. Mengambil masukan berupa hasil nilai matriks grayscale.
2. Melakukan perulangan sebanyak jumlah panjang matriks grayscale .
3. Melakukan perulangan sebanyak jumlah panjang i.
4. Melakukan proses perhitungan GrayScale dengan Persamaan 2.2.
5. Keluaran yang dihasilkan berupa nilai Matriks Grayscale.

#### 4.2.1.5 Perhitungan Matriks Ternormalisasi Sudut $0^\circ$ , $45^\circ$ , $90^\circ$ , dan $135^\circ$

Perhitungan matriks ternormalisasi sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$  dilakukan untuk mendapatkan nilai matriks ternormalisasi. Untuk tahapan proses perhitungan Matriks Ternormalisasi  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$  dapat dilihat pada Gambar 4.7.



**Gambar 4.7 Diagram Alir Matriks Ternormalisasi Sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$**

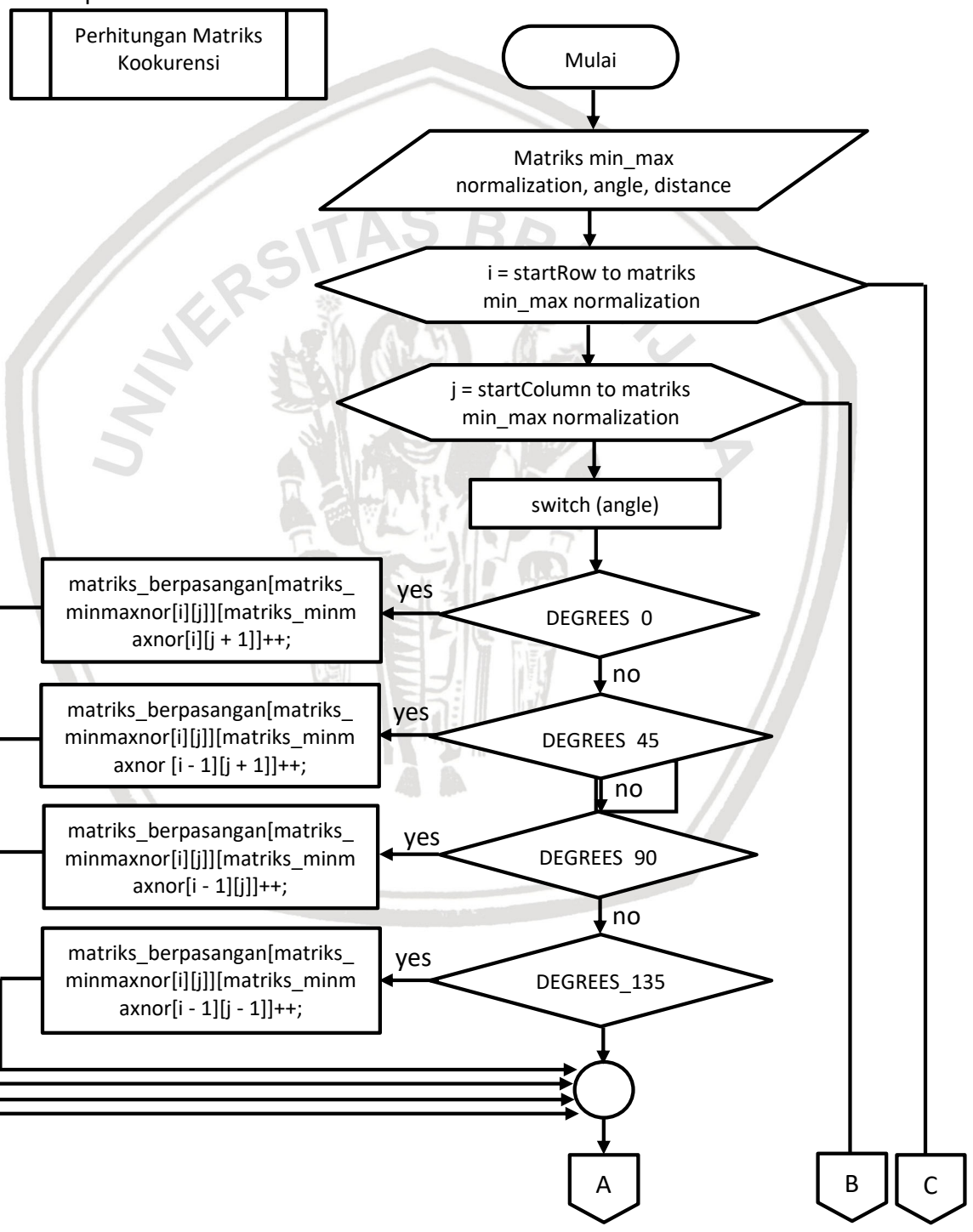
Berikut penjelasan perhitungan matriks ternormalisasi Sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , &  $135^\circ$ .

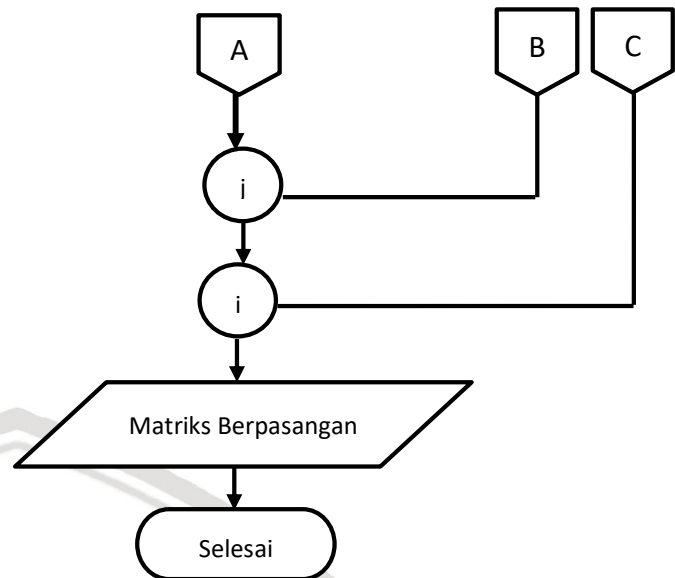
1. Mengambil masukan *matrix min-max normalization*
2. Melakukan Perhitungan Matriks Berpasangan
3. Melakukan Proses Transpose Matriks
4. Melakukan Perhitungan Matriks Simetris Kookurensi
5. Melakukan Perhitungan Matriks Normalisasi

- 6. Keluaran yang dihasilkan adalah matriks ternormalisasi sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ .

#### 4.2.1.6 Perhitungan Matriks Kookurensi

Perhitungan matriks kookurensi dilakukan untuk mendapatkan nilai jumlah pixel berpasangan dengan arah sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$  dan jarak=1. Nilai matriks ini digunakan untuk menghitung nilai matriks simetris kookurensi sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ . Untuk tahapan proses perhitungan kookurensi dapat dilihat pada Gambar 4.8.





**Gambar 4.8 Diagram Alir Perhitungan Matriks Berpasangan**

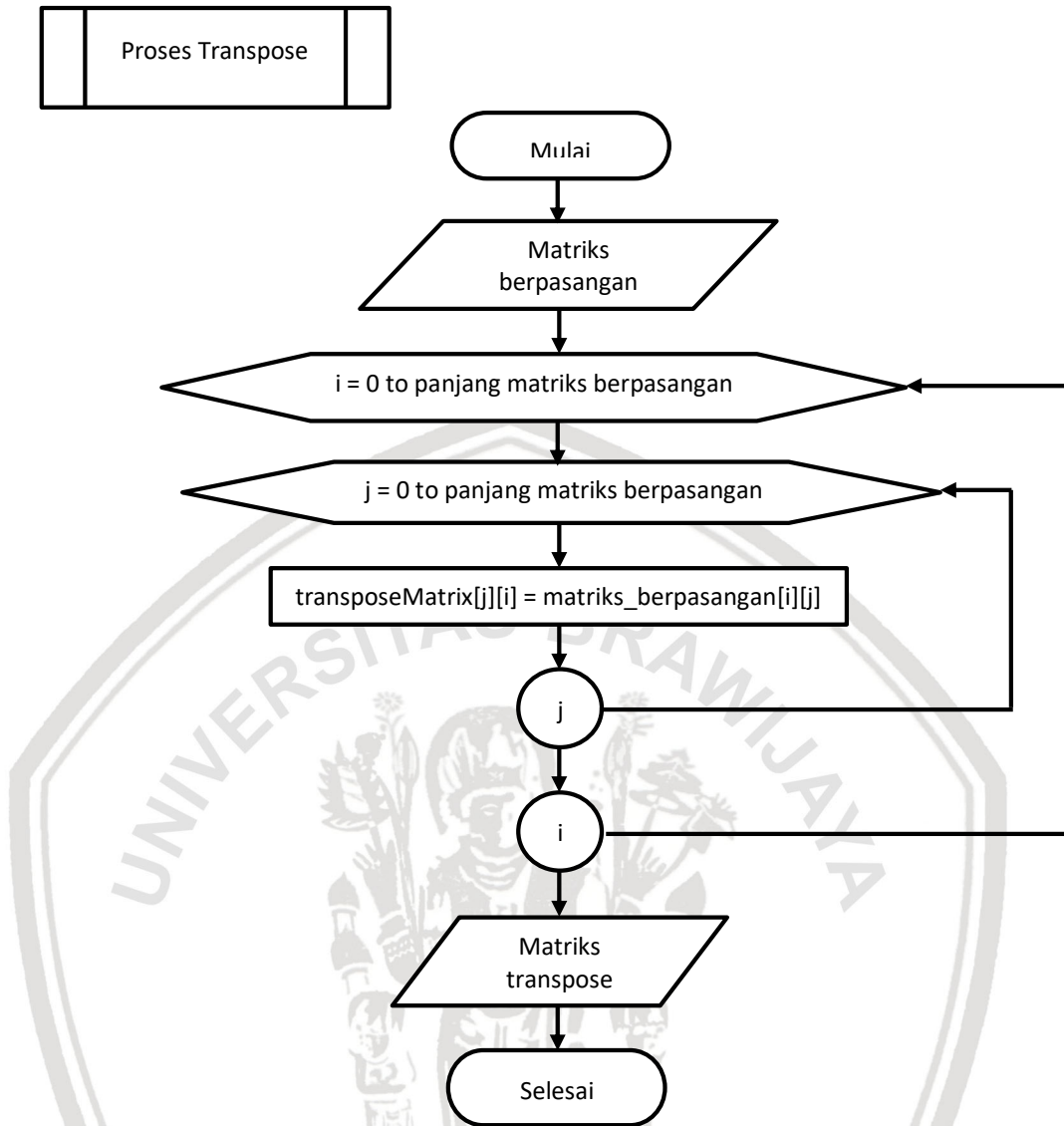
Berikut penjelasan perhitungan matriks berpasangan.

1. Mengambil masukan *matrix min-max normalization*, *angle*, dan *distance*.
2. Melakukan perulangan sebanyak jumlah panjang matriks *min-max normalization*.
3. Melakukan pemilihan kondisi sudut.
4. Melakukan proses perhitungan ketika sudut bernilai  $0^\circ$ , jika benar maka proses melakukan perhitungan banyaknya kemunculan pixel berpasangan dari sisi kiri ke sisi kanan, jika salah maka lanjut ke kondisi berikutnya.
5. Melakukan proses perhitungan ketika sudut bernilai  $45^\circ$ , jika benar maka proses melakukan perhitungan banyaknya kemunculan pixel berpasangan dari sisi diagonal kiri bawah ke sisi diagonal kanan atas, jika salah maka lanjut ke kondisi berikutnya.
6. Melakukan proses perhitungan ketika sudut bernilai  $90^\circ$ , jika benar maka proses melakukan perhitungan banyaknya kemunculan pixel berpasangan dari sisi bawah ke sisi atas, jika salah maka lanjut ke kondisi berikutnya.
7. Melakukan proses perhitungan ketika sudut bernilai  $135^\circ$ , jika benar maka proses melakukan perhitungan banyaknya kemunculan pixel berpasangan dari sisi diagonal kanan bawah ke sisi diagonal kiri atas.
8. Keluaran yang dihasilkan adalah matriks berpasangan sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ .

#### 4.2.1.7 Proses Transpose

Proses transpose dilakukan untuk mendapatkan matriks transpose dari matriks berpasangan. Proses transpose yaitu mengubah elemen baris menjadi elemen kolom, dan sebaliknya mengubah elemen kolom menjadi elemen baris. Nilai matriks ini digunakan untuk menghitung nilai matriks simetris sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ . Untuk tahapan proses matriks transpose dapat dilihat pada Gambar 4.9.





**Gambar 4.9 Diagram Alir Proses Matriks Transpose**

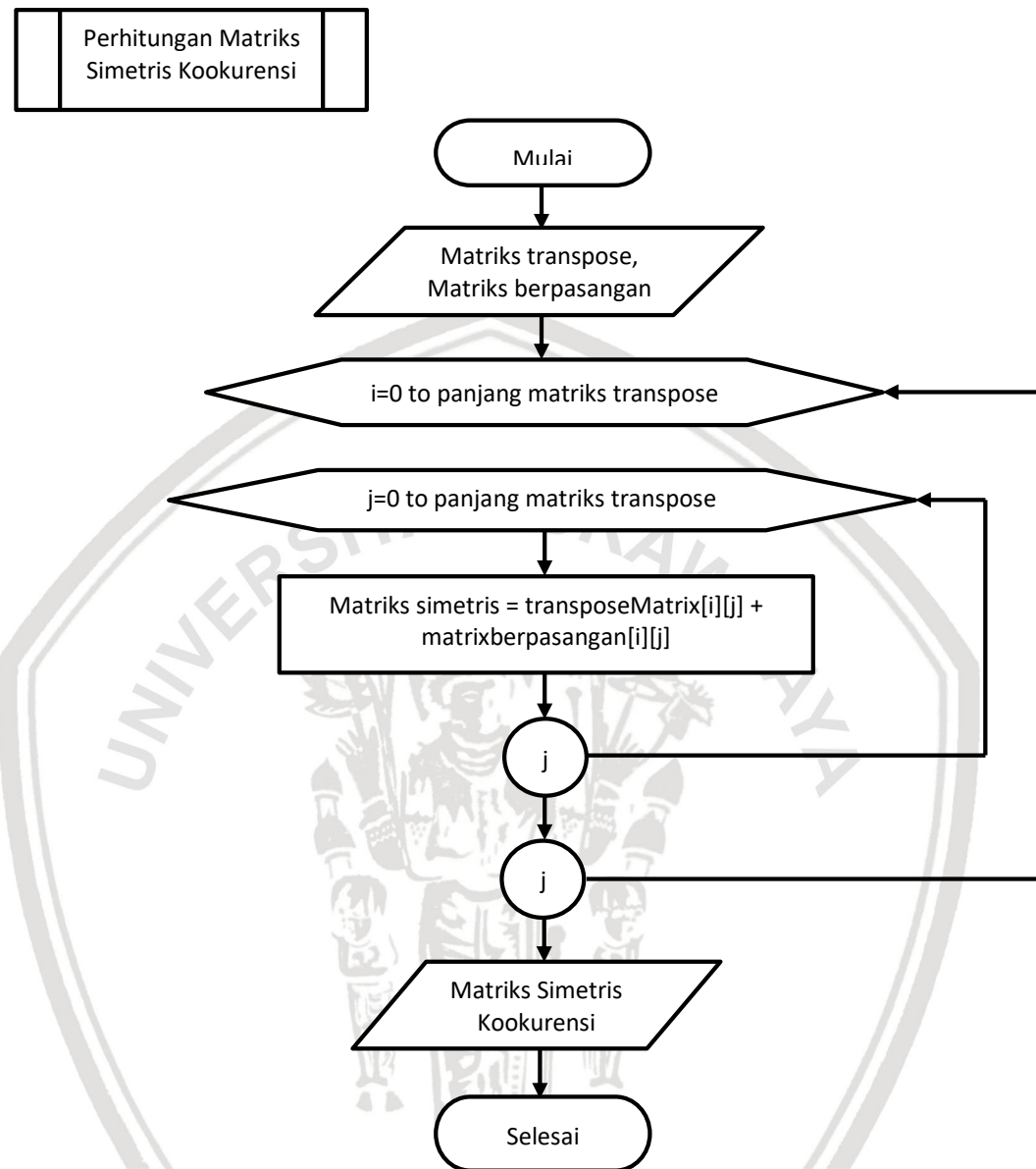
Berikut penjelasan proses matriks transpose Mengambil masukan berpasangan.

1. Melakukan perulangan sebanyak jumlah panjang baris matriks berpasangan.
2. Melakukan perulangan sebanyak jumlah panjang baris matriks berpasangan.
3. Melakukan proses transpose matriks.
4. Keluaran yang dihasilkan berupa nilai Matriks Grayscale.

#### 4.2.1.8 Perhitungan Matriks Simetris Kookurensi

Perhitungan matriks simetris dilakukan untuk mendapatkan matriks simetris dari matriks berpasangan. Cara mendapatkan nilai matriks simetris dengan menjumlahkan matriks berpasangan dengan matriks transposenya. Nilai matriks ini digunakan untuk menghitung nilai matriks simetris sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$

dan 135°. Untuk tahapan proses matriks transpose dapat dilihat pada Gambar 4.10.



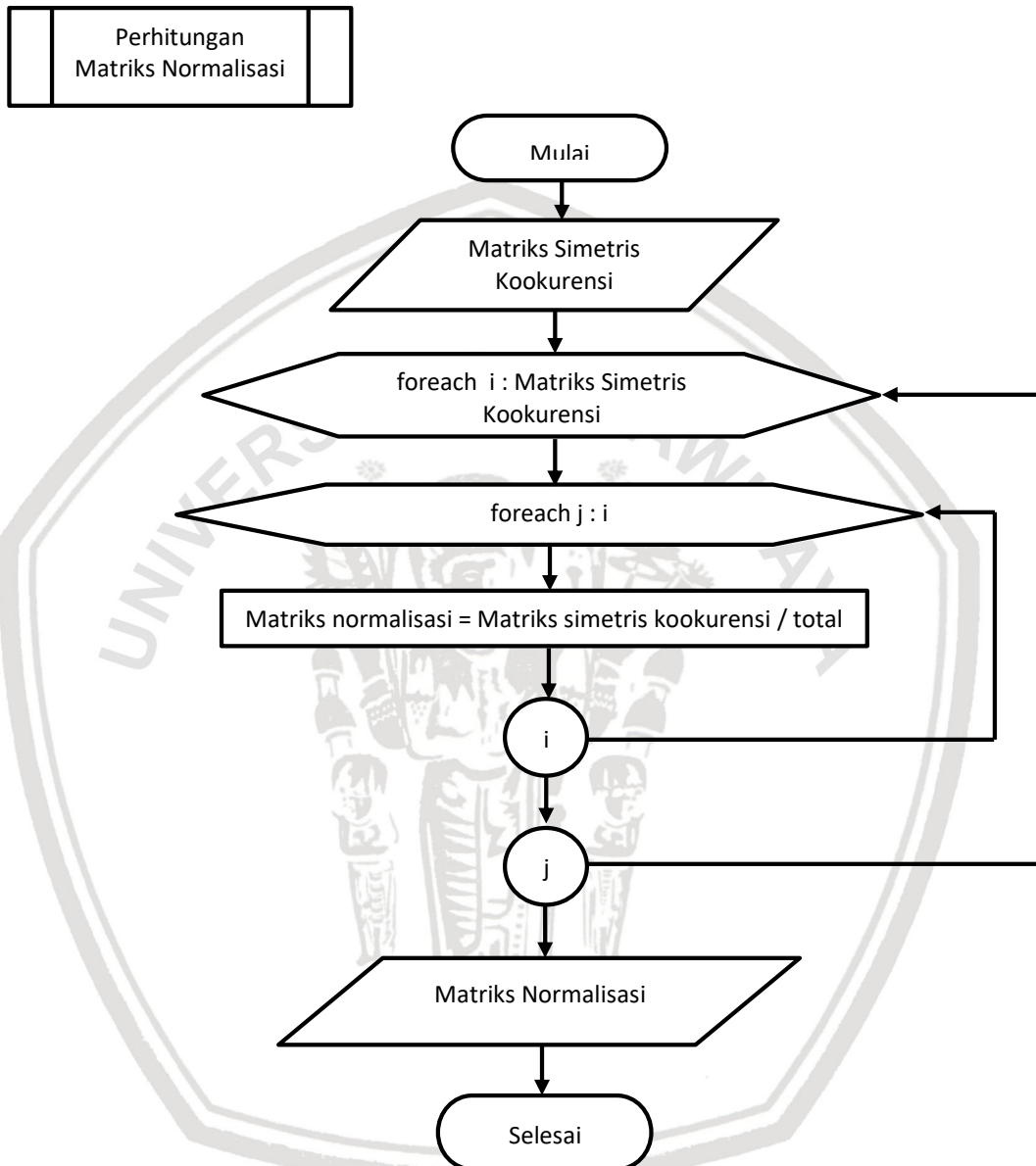
**Gambar 4.10 Diagram Alir Perhitungan Matrik Simetris Kookurensi**

Berikut penjelasan perhitungan matriks simetris kookurensi.

1. Mengambil masukan matrix berpasangan dan matriks transpose.
2. Melakukan perulangan sebanyak jumlah jumlah panjang baris matriks transpose.
3. Melakukan perulangan sebanyak jumlah jumlah panjang baris matriks transpose.
4. Melakukan perhitungan matriks simetris dengan menjumlahkan matriks transpose dengan matriks berpasangan.
5. Keluaran yang dihasilkan berupa nilai matriks simetris.

#### 4.2.1.9 Perhitungan Matriks Normalisasi

Perhitungan matriks normalisasi dilakukan untuk mendapatkan matriks normalisasi dari matriks simetris. Nilai matriks ini digunakan untuk menghitung nilai matriks normalisasi sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ . Untuk tahapan proses matriks transpose dapat dilihat pada Gambar 4.11.



**Gambar 4.11 Diagram Alir Perhitungan Matriks Normalisasi**

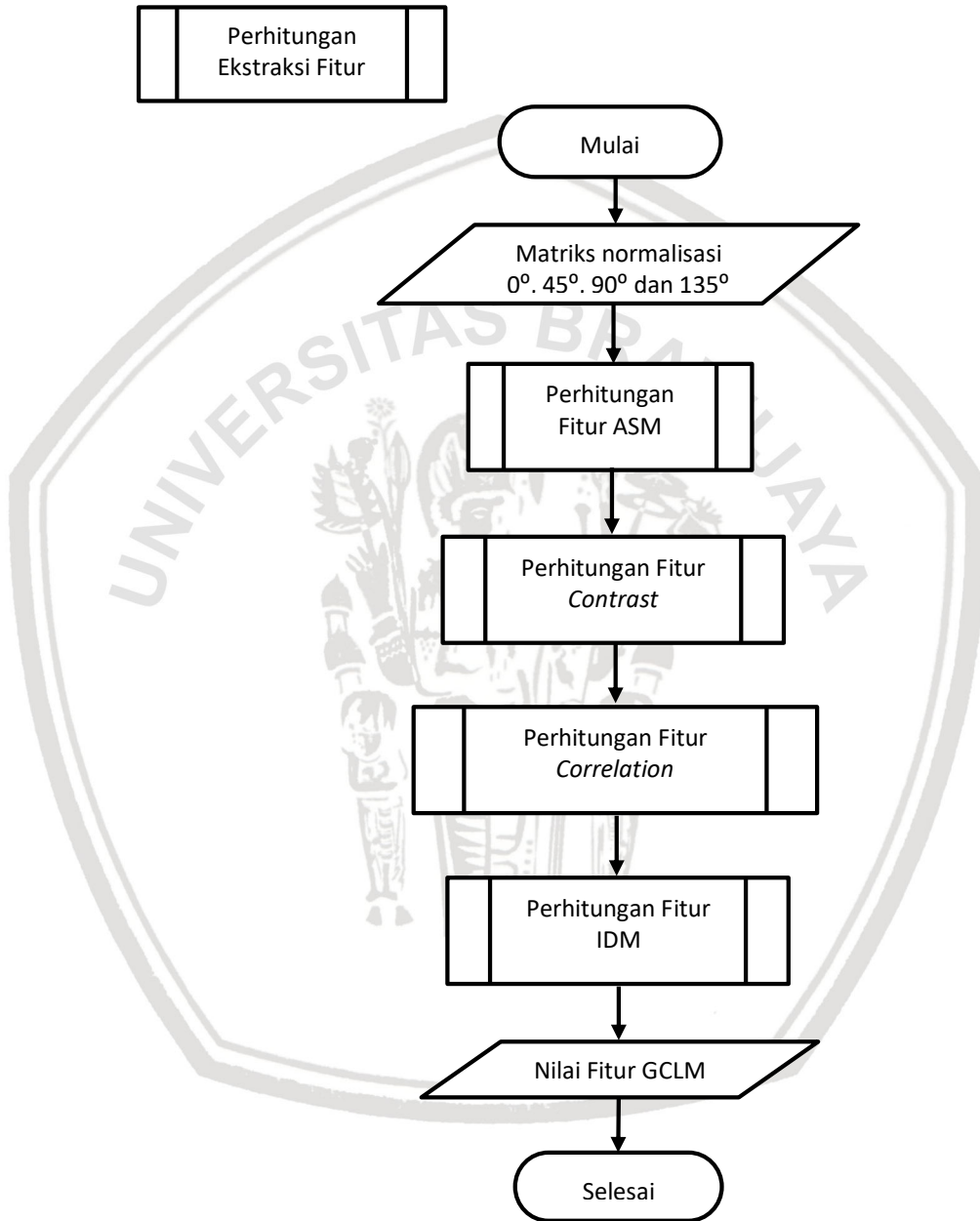
Berikut penjelasan perhitungan normalisasi.

1. Mengambil masukan matriks simetris.
2. Melakukan perulangan sebanyak jumlah panjang baris matriks simetris kookurensi.
3. Melakukan perulangan sebanyak jumlah panjang i.
4. Melakukan perhitungan matriks normalisasi.
5. Keluaran yang dihasilkan berupa nilai matriks simetris.



#### 4.2.1.10 Perhitungan Ekstraksi Fitur

Perhitungan ekstraksi fitur dilakukan untuk mendapatkan nilai-nilai fitur dari citra bibir yang digunakan sebagai data untuk perhitungan pada proses algoritme SVM. Fitur GLCM yang dihitung berdasarkan sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$  yang meliputi 4 fitur yaitu ASM, Contrast, Correlation dan IDM. Untuk tahapan proses matriks transpose dapat dilihat pada Gambar 4.12.



**Gambar 4.12 Diagram Alir Perhitungan Ekstraksi Fitur**

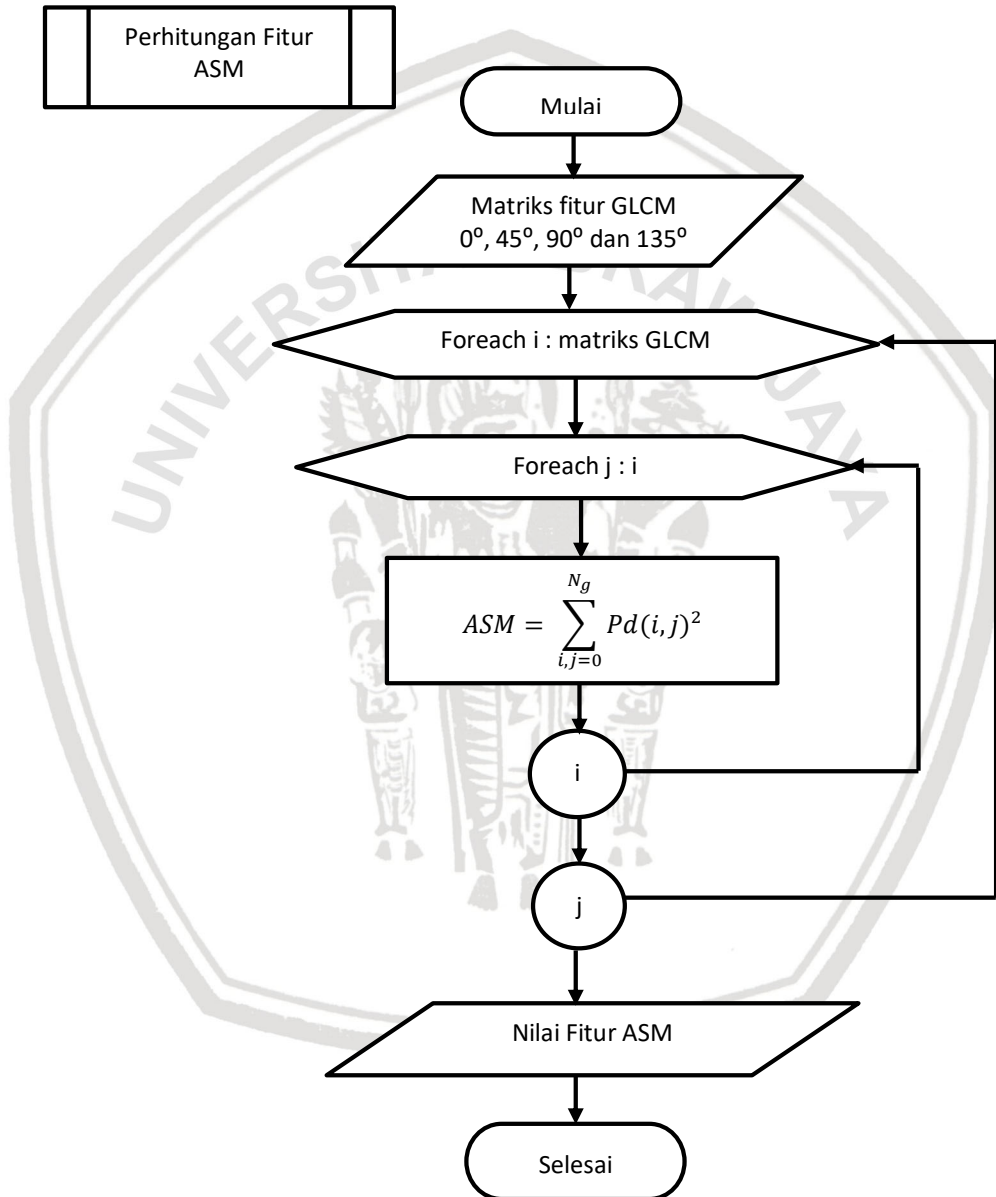
Berikut penjelasan perhitungan Ekstraksi Fitur.

1. Mengambil masukan nilai matriks kookurensi sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ .
2. Melakukan perhitungan fitur ASM.

3. Melakukan perhitungan fitur Contrast.
4. Melakukan perhitungan fitur Correlation.
5. Melakukan perhitungan fitur IDM.
6. Keluaran yang dihasilkan berupa nilai fitur GLCM.

#### 4.2.1.11 Perhitungan Fitur ASM

Perhitungan fitur ASM dilakukan untuk mendapatkan nilai fitur ASM. Nilai matriks ini akan digunakan sebagai data latih pada perhitungan algoritme SVM. Untuk tahapan perhitungan fitur ASM dapat dilihat pada Gambar 4.13.



**Gambar 4.13 Diagram Alir Perhitungan Fitur ASM**

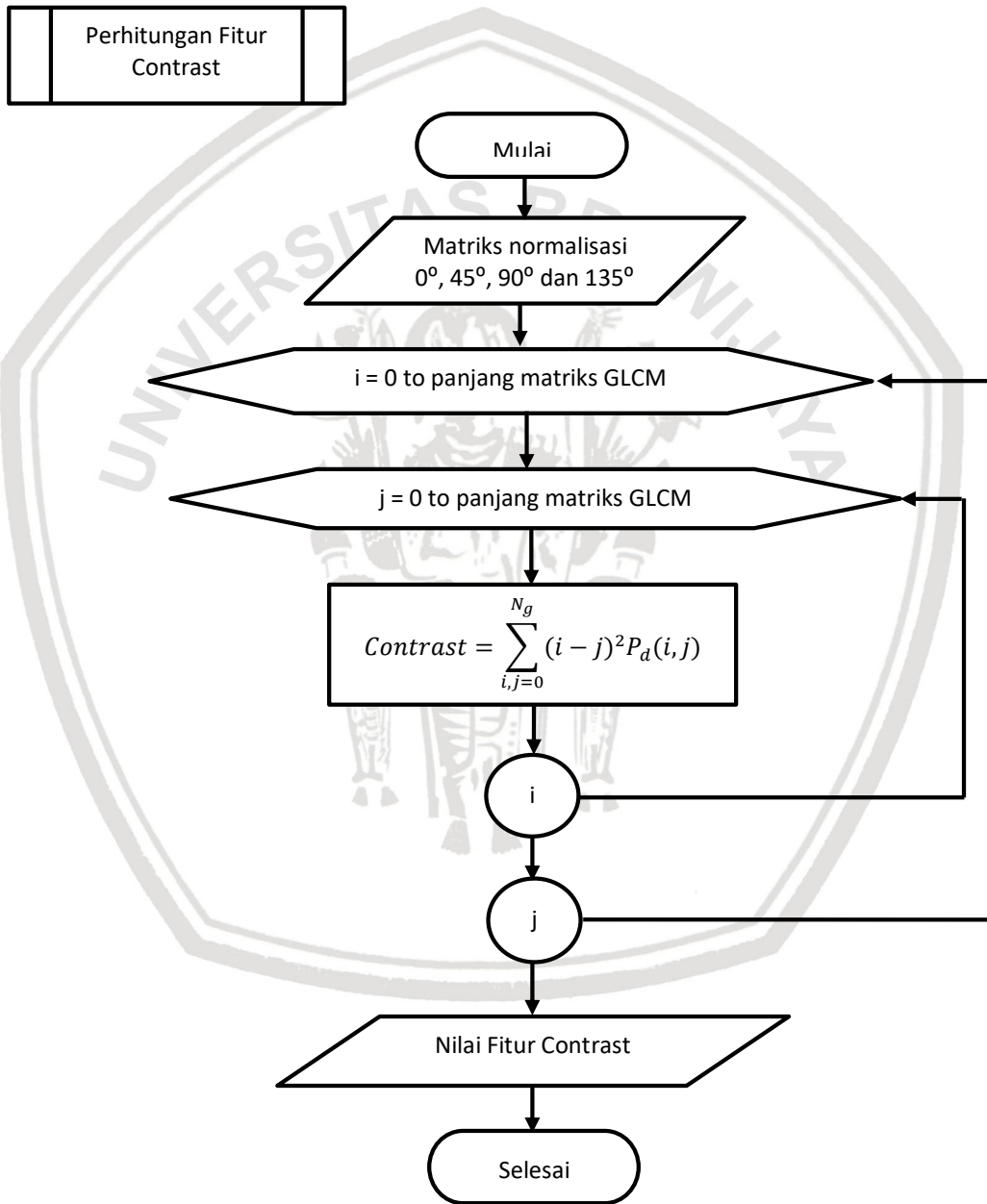
Berikut penjelasan perhitungan Fitur ASM.

1. Mengambil masukan nilai matriks normalisasi sudut 0°, 45°, 90° dan 135°.

2. Melakukan perulangan sebanyak jumlah panjang baris matriks GLCM.
3. Melakukan perulangan sebanyak jumlah panjang baris matriks GLCM.
4. Melakukan perhitungan ASM dengan Persamaan 2.3.
5. Keluaran yang dihasilkan berupa nilai fitur ASM.

**4.2.1.12 Perhitungan Fitur Contrast**

Perhitungan fitur contrast dilakukan untuk mendapatkan nilai fitur contrast. Nilai matriks ini akan digunakan sebagai data latih pada perhitungan algoritme SVM. Untuk tahapan perhitungan fitur contrast dapat dilihat pada Gambar 4.14.



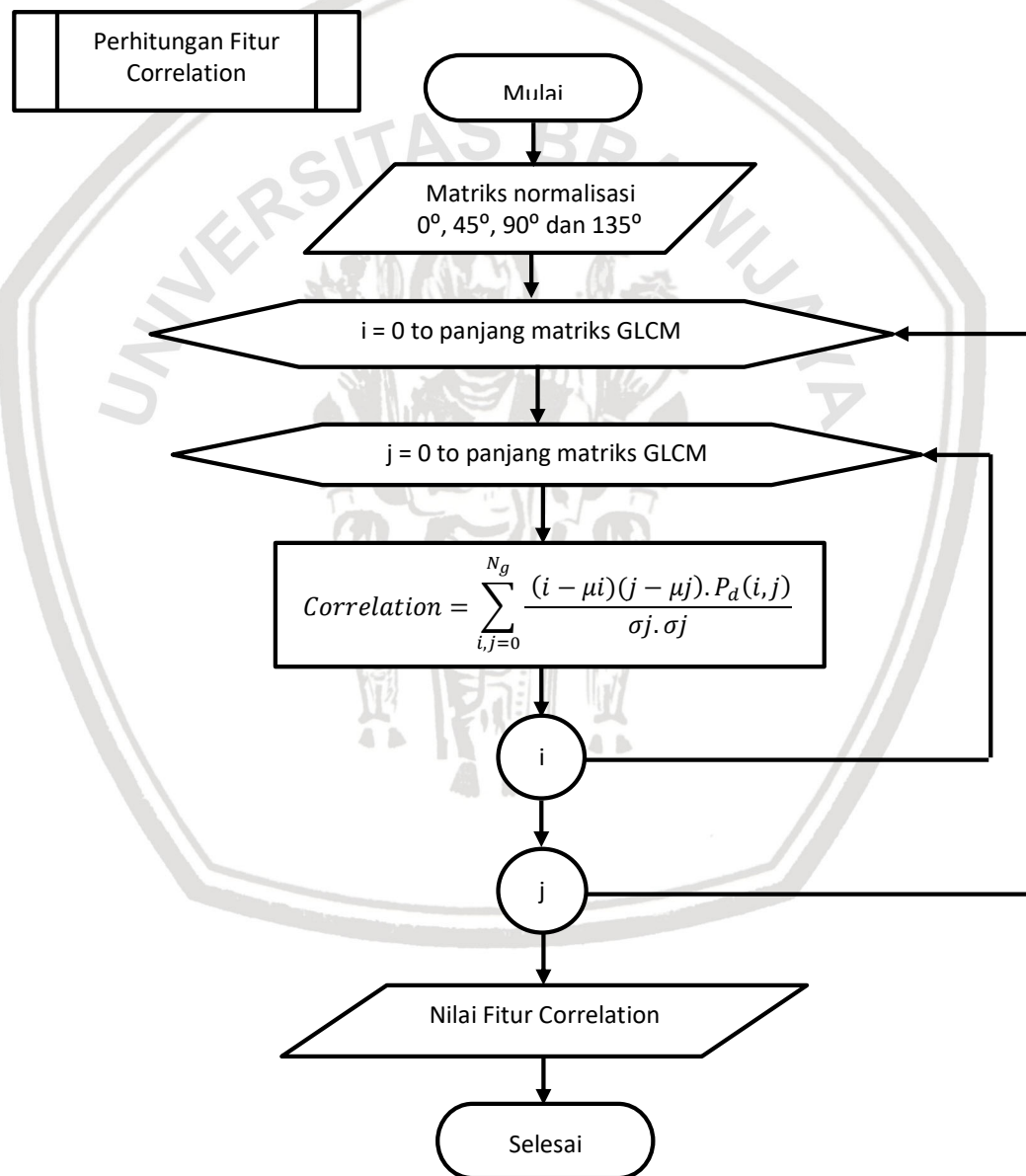
**Gambar 4.14 Diagram Alir Perhitungan Fitur Contrast**

Berikut penjelasan perhitungan Fitur Contrast.

1. Mengambil masukan nilai matriks normalisasi sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ .
2. Melakukan perulangan sebanyak jumlah panjang baris matriks GLCM.
3. Melakukan perulangan sebanyak jumlah panjang baris matriks GLCM.
4. Melakukan perhitungan contrast dengan Persamaan 2.4.
5. Keluaran yang dihasilkan berupa nilai fitur contrast.

#### 4.2.1.13 Perhitungan Fitur *Correlation*

Perhitungan fitur contrast dilakukan untuk mendapatkan nilai fitur correlation. Nilai matriks ini akan digunakan sebagai data latih pada perhitungan algoritme SVM. Untuk tahapan perhitungan fitur correlation dapat dilihat pada Gambar 4.15.



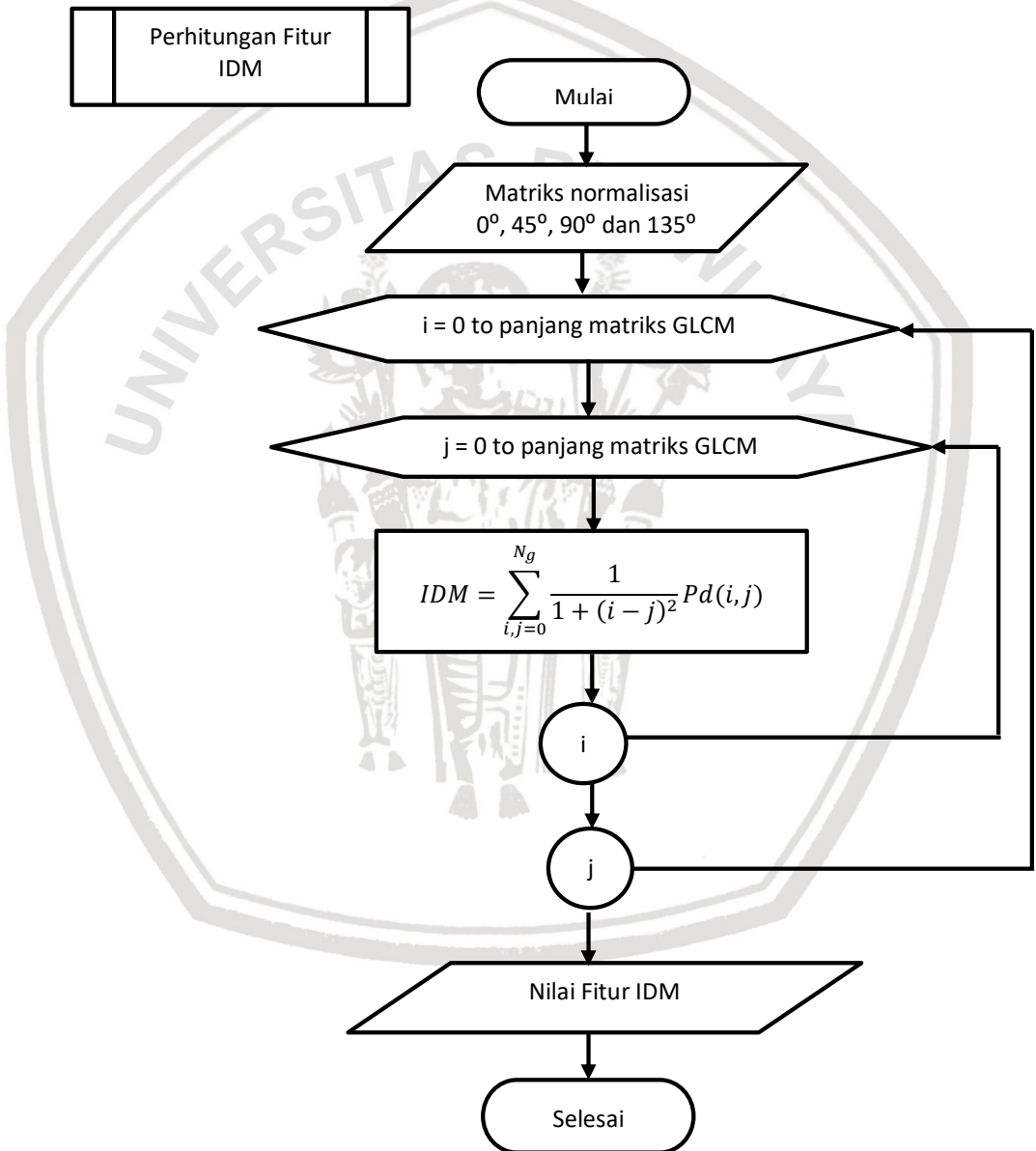
**Gambar 4.15 Diagram Alir Perhitungan Fitur Correlation**

Berikut penjelasan perhitungan Fitur Correlation.

1. Mengambil masukan nilai matriks normalisasi sudut 0°, 45°, 90° dan 135°.
2. Melakukan perulangan sebanyak jumlah panjang baris matriks GLCM.
3. Melakukan perulangan sebanyak jumlah panjang baris matriks GLCM.
4. Melakukan perhitungan correlation dengan Persamaan 2.5.
5. Keluaran yang dihasilkan berupa nilai fitur correlation.

**4.2.1.14 Perhitungan Fitur IDM**

Perhitungan fitur contrast dilakukan untuk mendapatkan nilai fitur IDM. Nilai matriks ini akan digunakan sebagai data latih pada perhitungan algoritme SVM. Untuk tahapan perhitungan fitur IDM dapat dilihat pada Gambar 4.16.



**Gambar 4.16 Diagram Alir Perhitungan Fitur IDM**

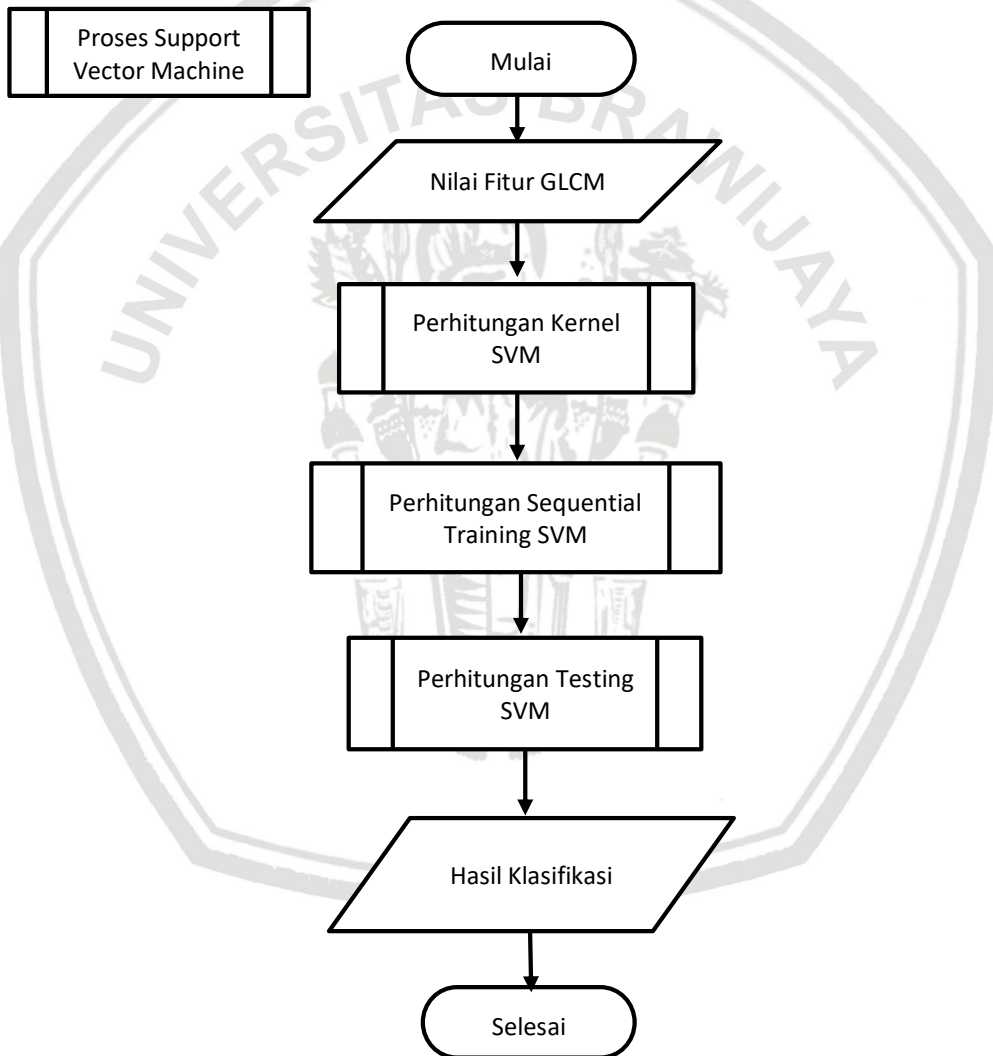
Berikut penjelasan perhitungan Fitur IDM.



1. Mengambil masukan nilai matriks normalisasi sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ .
2. Melakukan perulangan sebanyak jumlah panjang baris matriks GLCM.
3. Melakukan perulangan sebanyak jumlah panjang baris matriks GLCM.
4. Melakukan perhitungan IDM dengan Persamaan 2.6.
5. Keluaran yang dihasilkan berupa nilai fitur IDM.

#### 4.2.2 Perancangan Proses Algoritme *Support Vector Machine*

Pada proses SVM, langkah pertama yang dilakukan adalah mengambil *dataset*. Kemudian dilakukan proses pembuatan *kernel* dengan menentukan *kernel* yang digunakan. Pada penelitian ini menggunakan *Kernel Polynomial* yang akan menghasilkan matriks *kernel*. Diagram alir proses SVM dapat dilihat pada Gambar 4.17.



**Gambar 4.17 Diagram Alir Proses *Support Vector Machine***

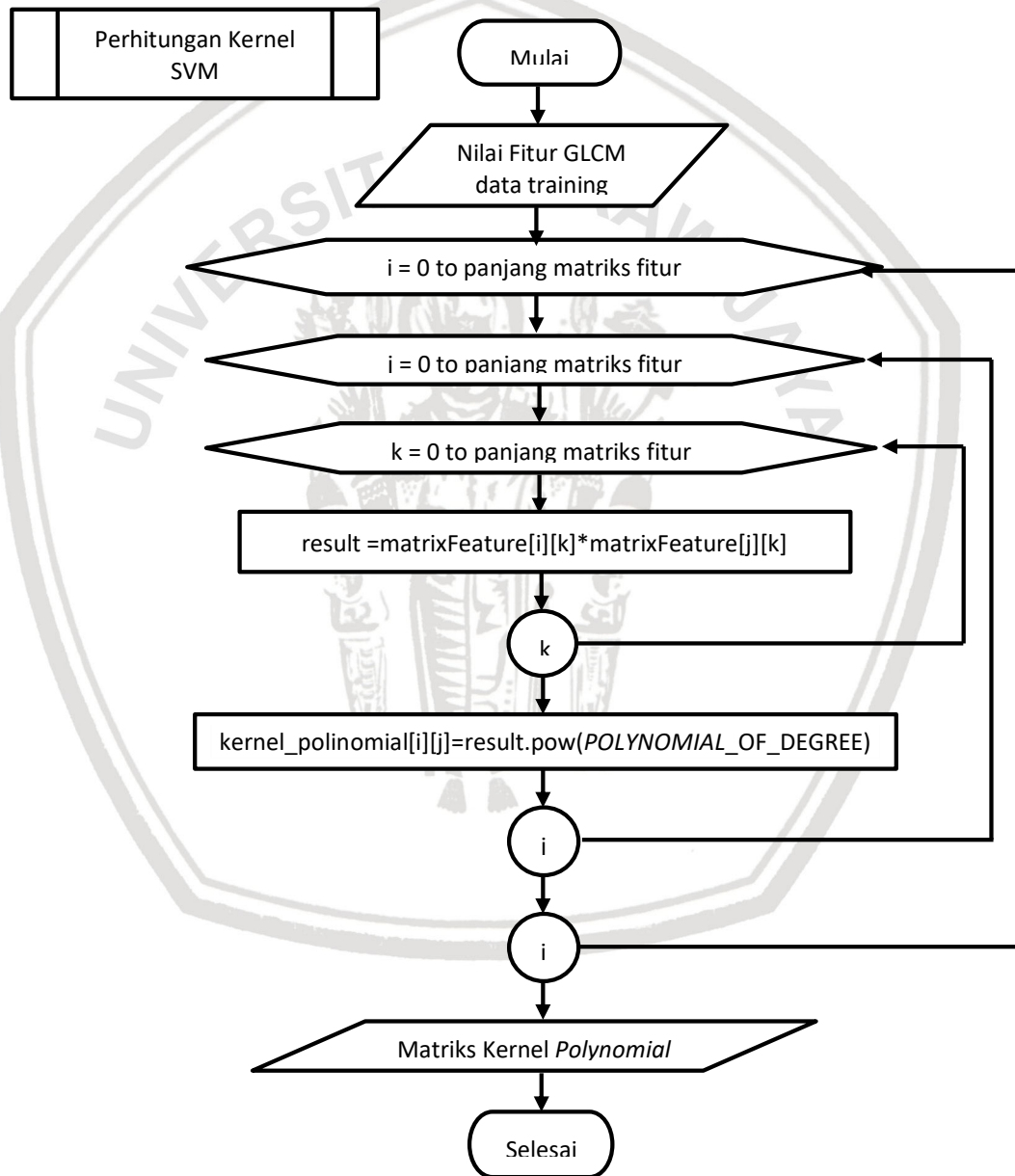
Berikut penjelasan Proses Support Vector Machine.

1. Mengambil masukan *data fitur* yaitu berupa nilai fitur GLCM.

2. Melakukan perhitungan kernel SVM.
3. Melakukan perhitungan data latih SVM dengan menggunakan metode *sequential training SVM*.
4. Melakukan perhitungan *testing SVM* dengan menerapkan strategi *One-Against-All* untuk menentukan kelas yang sesuai dengan klasifikasi.
5. Keluaran yang dihasilkan adalah hasil klasifikasi.

#### 4.2.2.1 Perhitungan Kernel SVM

Pada tahap perhitungan *kernel SVM* langkah yang dilakukan adalah mengambil data latih, kemudian data latih tersebut dilakukan perhitungan *kernel Polynomial*. Gambar 4.18 menjelaskan tentang proses perhitungan *kernel SVM*.



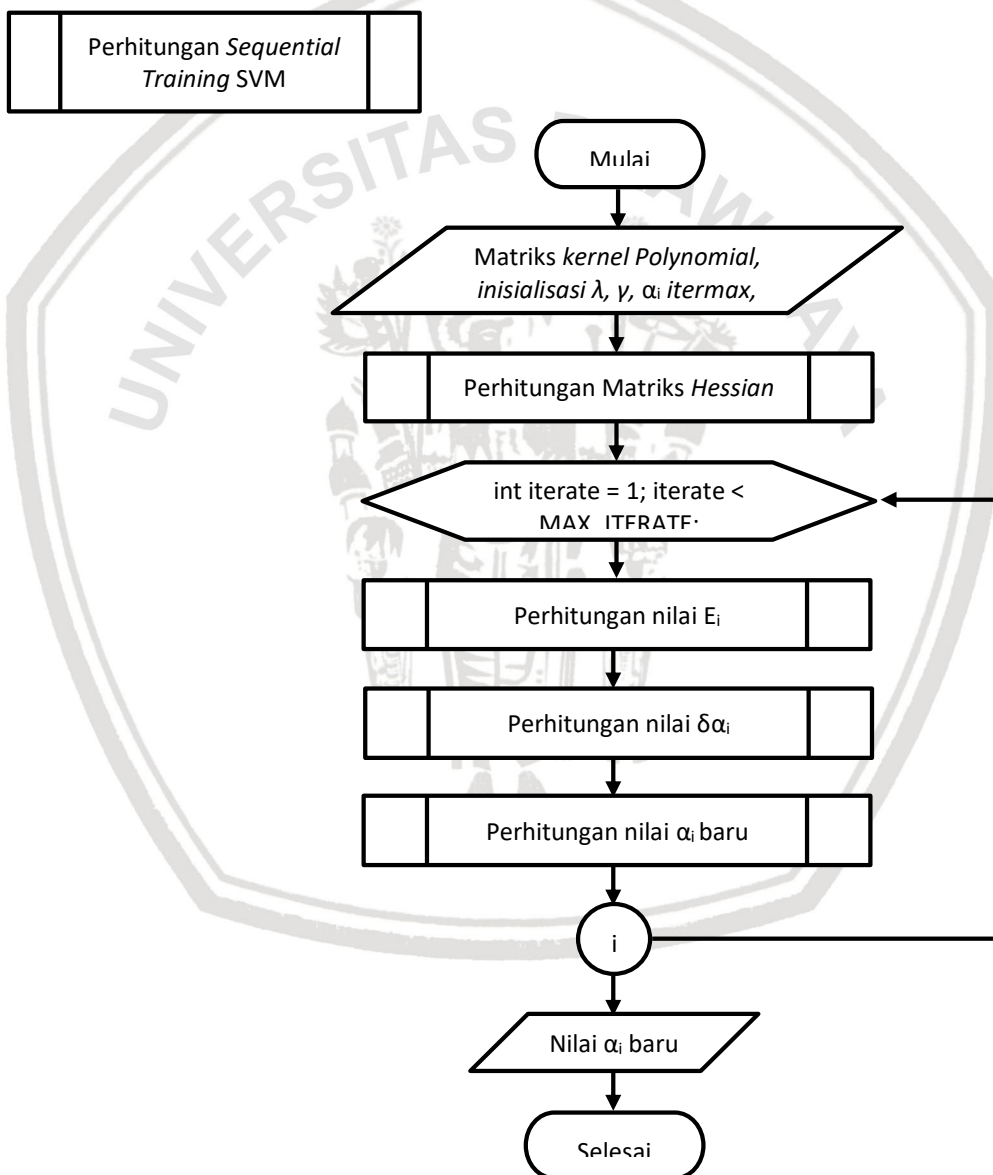
Gambar 4.18 Diagram Alir Perhitungan *Kernel Polynomial*

Berikut penjelasan perhitungan Kernel *Polynomial*.

1. Mengambil masukkan data berupa hasil nilai fitur GLCM.
2. Melakukan perulangan sebanyak jumlah panjang baris matriks fitur.
3. Melakukan perulangan sebanyak jumlah panjang baris matriks fitur.
4. Melakukan perulangan sebanyak jumlah panjang baris matriks fitur.
5. Perhitungan kernel *polynomial* seperti persamaan 2.13.
6. Keluaran berupa nilai kernel *polynomial* dalam bentuk matriks.

#### 4.2.2.2 Perhitungan *Sequential Training SVM*

Perhitungan *Sequential training SVM* dilakukan untuk mendapatkan *hyperplane* yang optimal. Langkah perhitungan *Sequential training SVM* dapat dilihat pada Gambar 4.19.



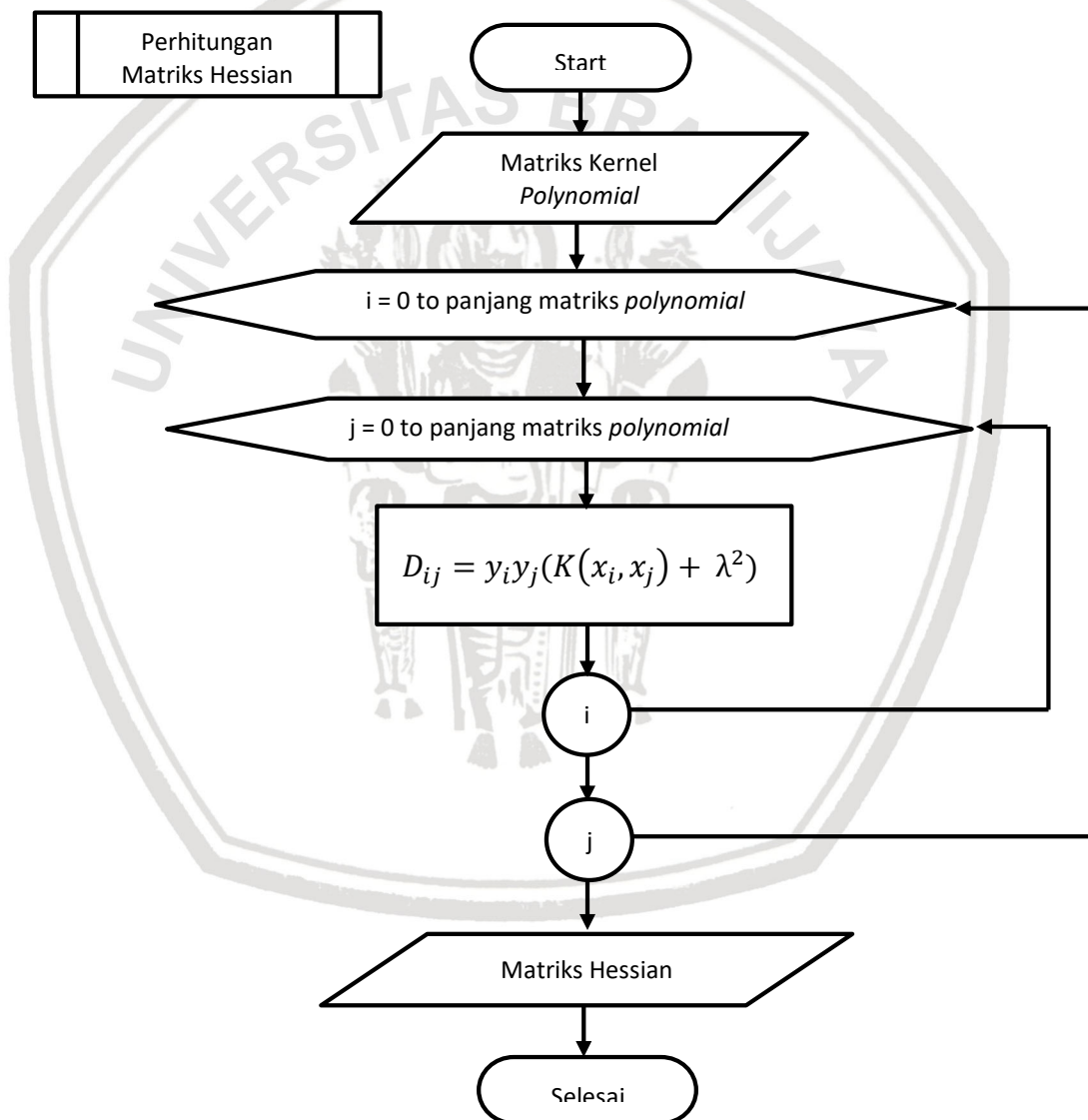
**Gambar 4.19** Diagram Alir Perhitungan *Sequential training SVM*

Berikut penjelasan perhitungan *Sequential training SVM*.

1. Mengambil masukkan Matriks kernel *Polynomial* dan melakukan inialisasi  $\lambda$ ,  $\gamma$ ,  $\alpha_i$  itermax, C.
2. Melakukan perhitungan Matriks Hessian.
3. Melakukan perulangan sebanyak nilai itermax yang di inputkan.
4. Melakukan perhitungan nilai  $E_i$ .
5. Melakukan perhitungan nilai  $\delta\alpha_i$ .
6. Melakukan perhitungan nilai  $\alpha_i$  baru.
7. Keluaran berupa nilai nilai  $\alpha_i$  baru.

#### 4.2.2.3 Perhitungan Matriks Hessian

Perhitungan Matriks Hessian dilakukan setelah mendapatkan nilai matriks kernel *polynomial*. Proses perhitungan matriks hessian dapat dilihat dalam diagram alir perhitungan matriks hessian pada Gambar 4.20.



**Gambar 4.20 Diagram Alir Perhitungan Matriks Hessian**

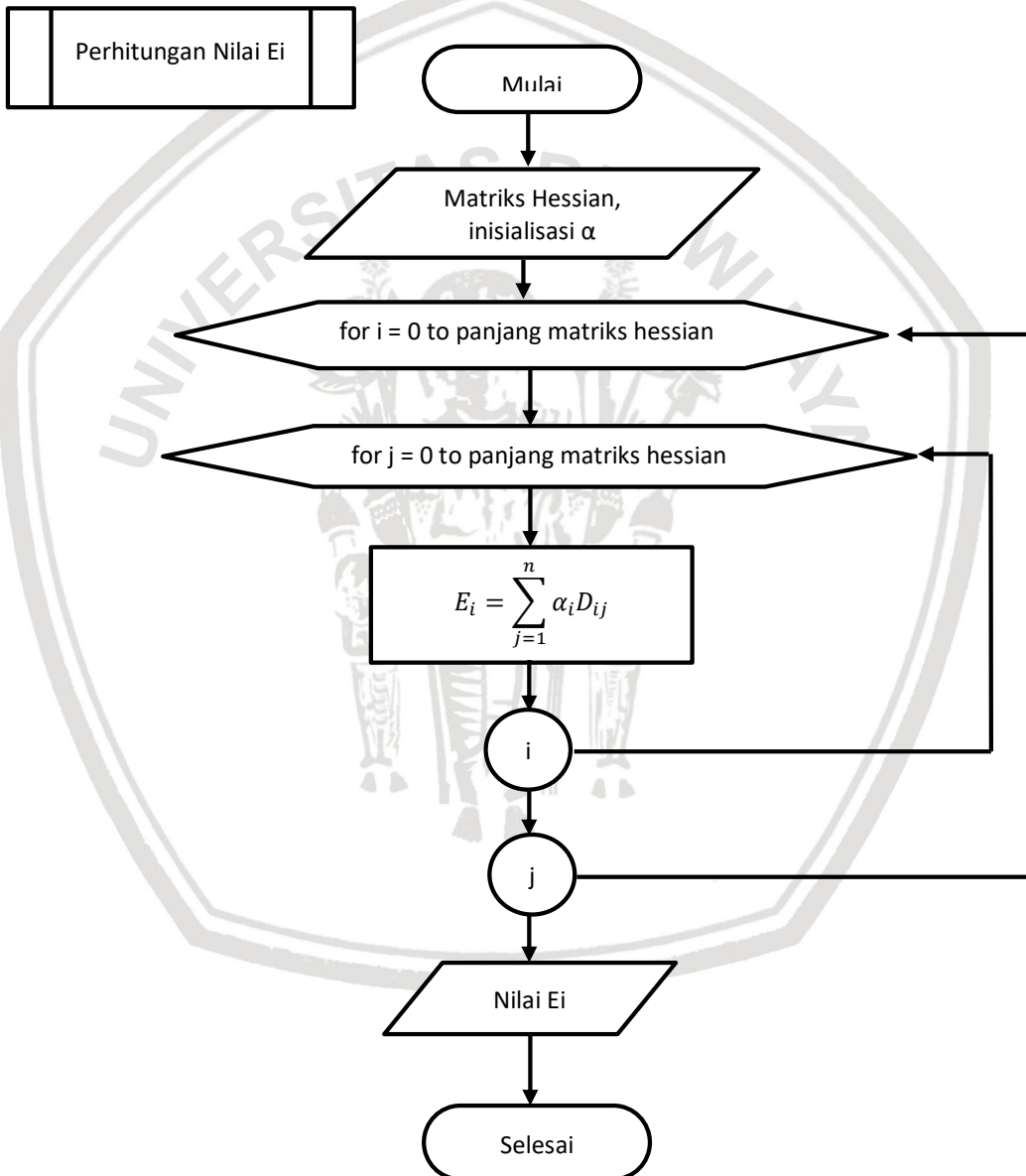
Berikut penjelasan perhitungan Matriks Hessian.

1. Mengambil masukkan matriks kernel *Polynomial*.

2. Melakukan perulangan sebanyak jumlah panjang baris matriks *polynomial*.
3. Melakukan perulangan sebanyak jumlah panjang baris matriks *polynomial*.
4. Perhitungan matriks *hessian* menggunakan Persamaan 2.16.
5. Keluaran yang dihasilkan adalah nilai matriks *hessian*.

#### 4.2.2.4 Perhitungan Nilai $E_i$

Perhitungan Nilai  $E_i$  dilakukan setelah mendapatkan nilai matriks *hessian*. Proses perhitungan Nilai  $E_i$  dapat dilihat dalam diagram alir perhitungan matriks *hessian* pada Gambar 4.21.



**Gambar 4.21 Diagram Alir Perhitungan Nilai  $E_i$**

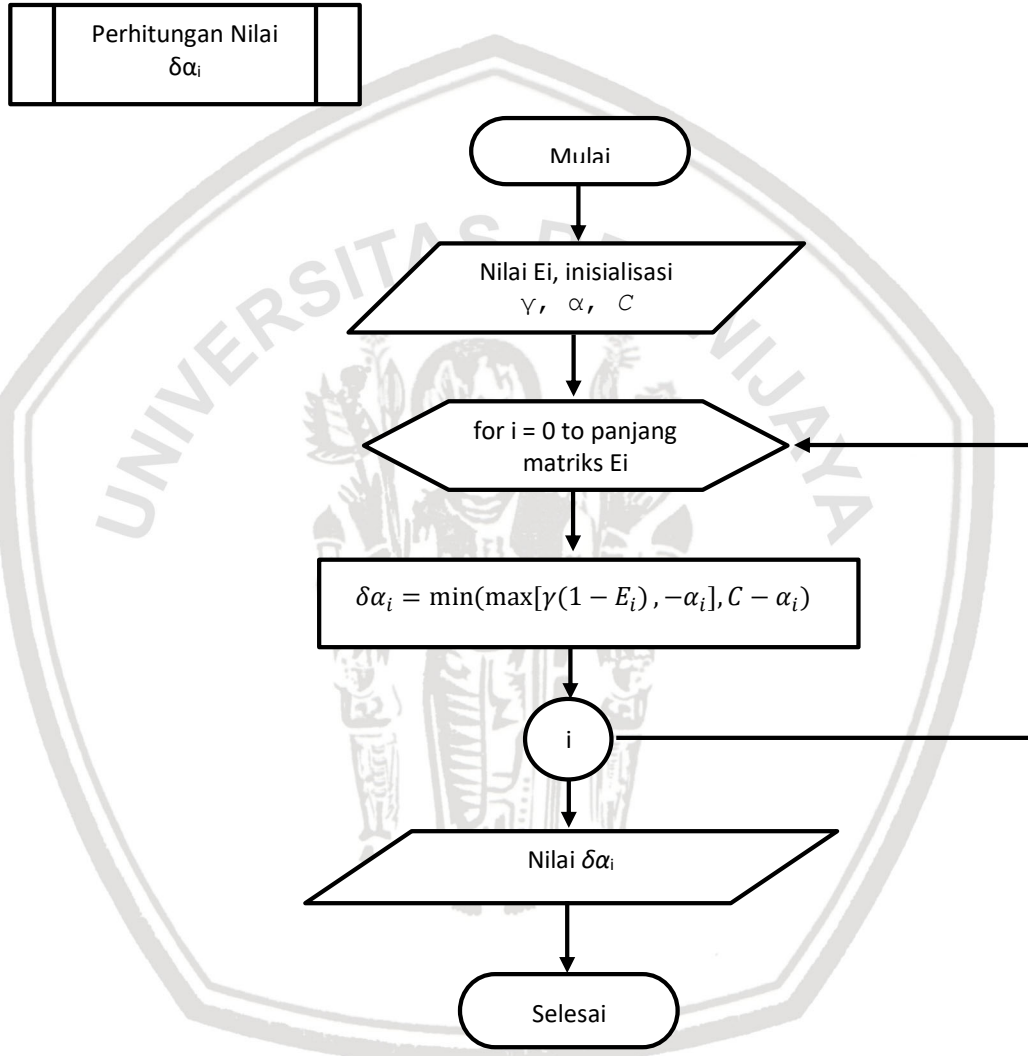
Berikut penjelasan perhitungan nilai  $E_i$ .

1. Mengambil masukkan matriks *hessian* dan inialisasi dari  $\alpha$ .

2. Melakukan perulangan sebanyak jumlah panjang baris matriks hessian.
3. Melakukan perulangan sebanyak jumlah panjang baris matriks hessian.
4. Perhitungan Nilai  $E_i$  menggunakan Persamaan 2.17.
5. Keluaran yang dihasilkan adalah Nilai  $E_i$ .

#### 4.4.1.1 Perhitungan Nilai $\delta\alpha_i$

Perhitungan Nilai  $\delta\alpha_i$  dilakukan setelah mendapatkan Nilai  $E_i$ . Proses perhitungan Nilai  $\delta\alpha_i$  dapat dilihat dalam diagram alir perhitungan matriks hessian pada Gambar 4.22.



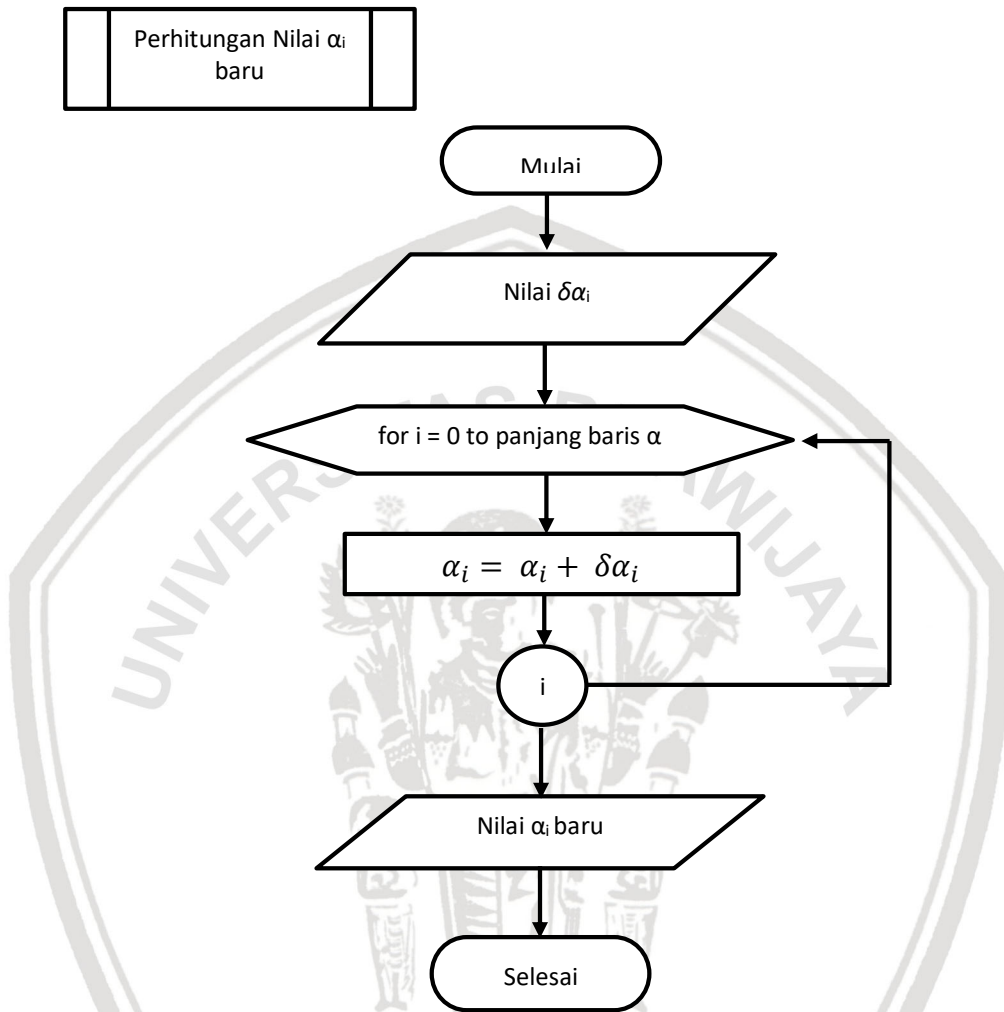
Gambar 4.22 Diagram Alir Perhitungan Nilai  $\delta\alpha_i$

Berikut penjelasan perhitungan nilai  $\delta\alpha_i$ .

1. Mengambil masukkan nilai  $E_i$  dan inialisasi dari  $C$ , learning rate dan  $\alpha_i$ .
2. Melakukan perulangan sebanyak jumlah panjang baris  $E_i$ .
3. Perhitungan  $\delta\alpha_i$  menggunakan Persamaan 2.18.
4. Keluaran yang dihasilkan adalah Nilai  $\delta\alpha_i$ .

#### 4.2.2.5 Perhitungan Nilai $\alpha_i$ baru

Perhitungan Nilai  $\alpha_i$  baru dilakukan setelah mendapatkan Nilai  $\delta\alpha_i$ . Proses perhitungan Nilai  $\alpha_i$  baru dapat dilihat dalam diagram alir perhitungan matriks hessian pada Gambar 4.23.



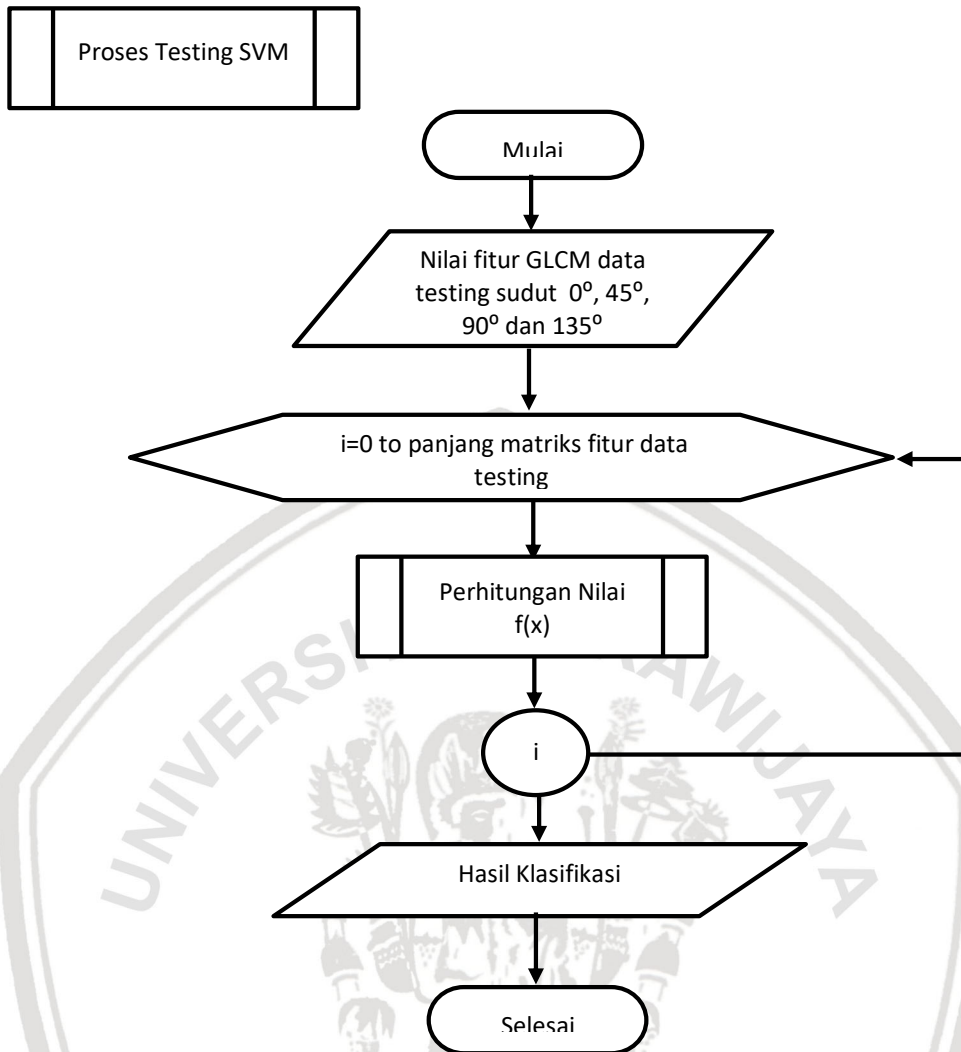
Gambar 4.23 Diagram Alir Perhitungan Nilai  $\alpha_i$  baru

Berikut penjelasan alir perhitungan nilai  $\alpha_i$  baru.

1. Mengambil masukkan nilai  $\delta\alpha_i$ .
2. Melakukan perulangan sebanyak jumlah panjang baris  $\alpha$ .
3. Perhitungan  $\alpha_i$  menggunakan Persamaan 2.19.
4. Keluaran yang dihasilkan adalah Nilai  $\alpha_i$ .

#### 4.2.2.6 Perhitungan *Testing SVM*

Perhitungan *testing SVM* dilakukan untuk melakukan pengujian data uji terhadap data latih yang telah dilakukan pada proses *sequential training SVM*. Langkah perhitungan *testing SVM* dapat dilihat pada Gambar 4.24.



**Gambar 4.24 Diagram Alir Perhitungan *Testing* SVM**

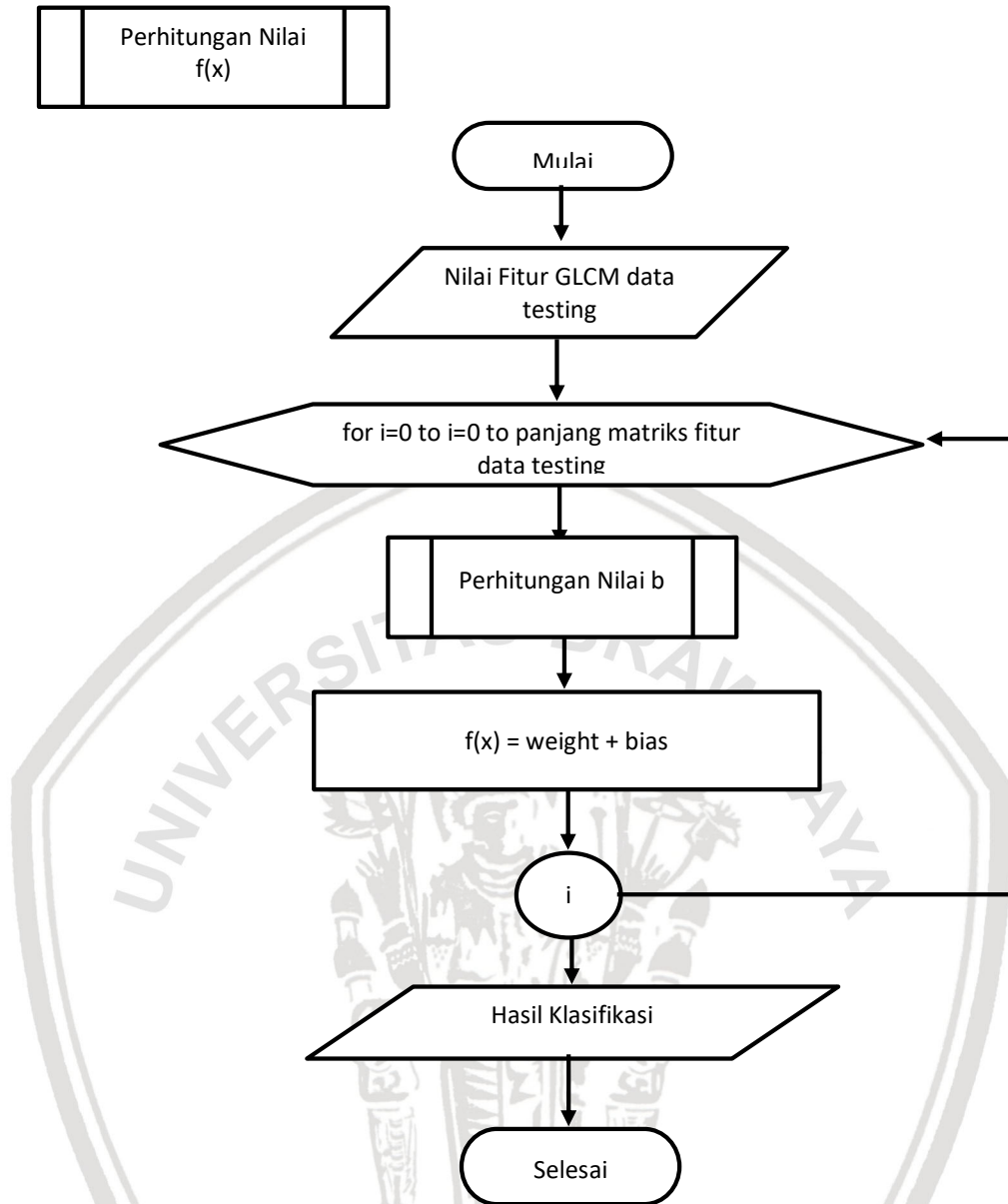
Berikut penjelasan perhitungan Fitur ASM.

1. Mengambil masukan nilai fitur GLCM data *testing* sudut 0°, 45°, 90° dan 135°.
2. Melakukan perulangan sebanyak jumlah panjang baris matriks fitur data *testing*.
3. Melakukan perhitungan nilai  $f(x)$  dengan Persamaan .
4. Keluaran yang dihasilkan berupa hasil klasifikasi.

#### 4.2.2.7 Perhitungan Nilai $f(x)$

Perhitungan Nilai  $f(x)$  dilakukan untuk mendapatkan hasil klasifikasi. Proses perhitungan Nilai  $f(x)$  dapat dilihat dalam diagram alir perhitungan matriks hessian pada Gambar 4.25.





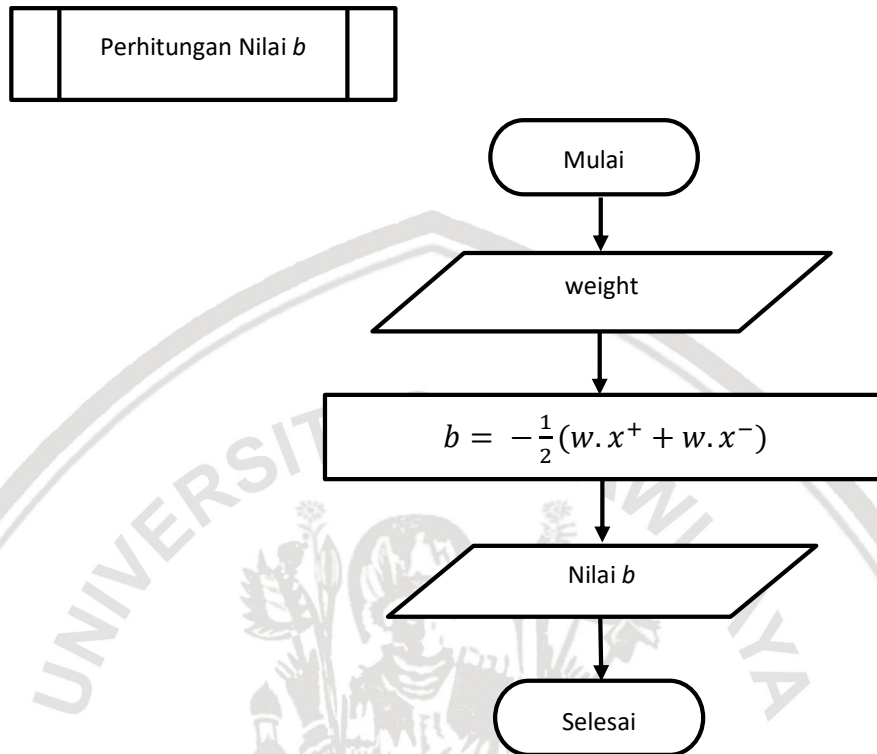
**Gambar 4.25 Diagram Alir Perhitungan Nilai  $f(x)$**

Berikut penjelasan perhitungan nilai  $f(x)$ .

1. Mengambil masukan nilai fitur GLCM data *testing* sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ .
2. Melakukan perulangan sebanyak jumlah panjang baris matriks fitur data *testing*.
3. Melakukan perhitungan nilai b.
4. Perhitungan Nilai  $f(x)$  dengan menjumlahkan nilai weight dengan bias.
5. Keluaran yang dihasilkan adalah hasil klasifikasi.

#### 4.2.2.8 Perhitungan Nilai $b$

Perhitungan nilai  $b$  dibutuhkan pada proses perhitungan nilai  $f(x)$ . Proses perhitungan nilai  $b$  dapat dilihat dalam diagram alir perhitungan matriks hessian pada Gambar 4.26.



Gambar 4.26 Diagram Alir Perhitungan Nilai  $b$

Berikut penjelasan perhitungan nilai  $b$ .

1. Mengambil masukan nilai weight.
2. Melakukan perhitungan nilai  $b$  dengan Persamaan 2.10.
3. Keluaran yang dihasilkan adalah nilai  $b$  pada tiap level.

### 4.3 Manualisasi

#### 4.3.1 Perhitungan Manual Proses GLCM

##### 4.3.1.1 Pre-Processing

Pada proses manualiasi untuk perhitungan nilai ekstraksi fitur GLCM menggunakan citra bibir yang telah dipotong pada titik yang dipilih dengan ukuran 10x10 piksel. Hasil potongan gambar yang digunakan untuk perhitungan manualisasi dapat dilihat pada Gambar 4.15.



**Gambar 4.27 Potongan citra bibir dengan ukuran 10x10 piksel**

Kemudian melakukan pre-processing yaitu merubah citra RGB menjadi citra *grayscale* atau disebut *grayscale*. Untuk citra RGB seperti Gambar 4.27, akan dihasilkan nilai *Red*, *Green* dan *Blue* pada setiap titik *pixel* seperti pada Tabel 4.2 hingga Tabel 4.4.

**Tabel 4.2 Nilai *red* tiap-tiap *pixel* Gambar 4.27**

x,y	0	1	2	3	4	5	6	7	8	9
0	225	222	223	231	240	242	232	222	229	230
1	234	232	231	232	232	229	223	217	231	233
2	235	235	235	233	230	228	228	228	236	236
3	236	236	237	237	237	237	238	238	238	237
4	240	238	237	238	241	241	238	233	235	234
5	231	229	229	233	238	241	238	234	230	230
6	224	224	225	227	229	232	235	237	229	230
7	233	234	232	226	220	219	225	231	231	232
8	239	238	227	226	219	221	238	227	222	236
9	226	231	229	236	228	223	231	213	220	234

**Tabel 4.3 Nilai *green* tiap-tiap *pixel* Gambar 4.27**

x,y	0	1	2	3	4	5	6	7	8	9
0	212	209	210	218	227	229	219	209	216	217
1	221	219	218	219	219	216	210	204	218	220
2	222	222	222	220	217	215	215	215	223	223
3	223	223	224	224	224	224	225	225	225	224
4	227	225	224	225	228	228	225	220	222	221
5	218	216	216	220	225	228	225	221	217	217
6	211	211	212	214	216	219	222	224	216	217
7	220	221	219	213	207	206	212	218	218	219
8	226	225	214	213	206	208	225	214	209	223
9	213	218	216	223	215	210	218	200	207	221

**Tabel 4.4 Nilai *blue* tiap-tiap *pixel* Gambar 4.27**

x,y	0	1	2	3	4	5	6	7	8	9
0	193	190	191	199	208	210	200	190	197	198
1	202	200	199	200	200	197	191	185	199	201
2	203	203	203	201	198	196	196	196	204	204
3	206	206	207	207	207	207	208	208	208	207
4	210	208	207	208	211	211	208	203	205	204
5	201	199	199	203	208	211	208	204	200	200
6	194	194	195	197	199	202	205	207	199	200
7	204	205	203	197	191	190	196	202	202	203
8	210	209	198	197	190	192	209	198	193	207
9	197	202	200	207	199	194	202	184	191	205

Kemudian melakukan *grayscale* dengan menghitung rata-rata nilai *Red*, *Green* dan *Blue* pada tiap *pixel* yang telah didapatkan dengan persamaan 2.1.

$$grayscale(0,0) = \frac{225 + 212 + 193}{3} = \frac{630}{3} = 210$$

ol

Dari contoh perhitungan di atas, untuk nilai *pixel* (0,0) didapat nilai *Red*=225, *Green*=212 dan *Blue*=193. Hasil keseluruhan perhitungan nilai *pixel grayscale* ditunjukkan pada Tabel 4.5.

**Tabel 4.5 Hasil perhitungan *grayscale***

x,y	0	1	2	3	4	5	6	7	8	9
0	210	207	208	216	225	227	217	207	214	215
1	219	217	216	217	217	214	208	202	216	218
2	220	220	220	218	215	213	213	213	221	221
3	222	222	223	223	223	223	224	224	224	223
4	226	224	223	224	227	227	224	219	221	220
5	217	215	215	219	224	227	224	220	216	216
6	210	210	211	213	215	218	221	223	215	216
7	219	220	218	212	206	205	211	217	217	218
8	225	224	213	212	205	207	224	213	208	222
9	212	217	215	222	214	209	217	199	206	220

Lalu melakukan normalisasi nilai *pixel grayscale* dengan persamaan 2.2.

$$Min - max\ normalization = (210 - 0) \frac{4 - 0}{255 - 0} + 0 = 3$$

Dari contoh perhitungan di atas, nilai *pixel grayscale* yang dinormalisasi didapat dari nilai *pixel* ke (0,0). Hasil keseluruhan perhitungan nilai *pixel grayscale* normalisasi ditunjukkan pada Tabel 4.6.

**Tabel 4.6 Hasil nilai normalisasi nilai *pixel grayscale***

x,y	0	1	2	3	4	5	6	7	8	9
0	3	3	3	3	4	4	3	3	3	3
1	3	3	3	3	3	3	3	3	3	3
2	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	4	4	4	3
4	4	4	3	4	4	4	4	3	3	3
5	3	3	3	3	4	4	4	3	3	3
6	3	3	3	3	3	3	3	3	3	3
7	3	3	3	3	3	3	3	3	3	3
8	4	4	3	3	3	3	4	3	3	3
9	3	3	3	3	3	3	3	3	3	3

**4.3.1.2 Ekstraksi Fitur GLCM**

Dari hasil *pre-processing* didapatkan hasil akhir yaitu nilai matriks normalisasi nilai *pixel grayscale* yang kemudian dilakukan perhitungan ekstraksi fitur untuk citra sidik bibir. Berikut perhitungan manualisasi ekstraksi fitur tekstur GLCM.

1. Membuat matriks *co-occurrence* dengan 4 arah sudut yaitu  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$ . Jarak untuk *pixel* asal ke *pixel* tetangga yaitu sebesar 1 *pixel*. Pada Gambar 4.28 sampai Gambar 4.31 merupakan hasil dari pembentukan matriks *co-occurrence* untuk *cooccurrence pixel* (3,4).
  - a.) Sudut  $0^\circ$

3	3	3	3	4	4	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	4	4	4	3
4	4	3	4	4	4	4	3	3	3
3	3	3	3	4	4	4	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
4	4	3	3	3	3	4	3	3	3
3	3	3	3	3	3	3	3	3	3

**Gambar 4.28 Pembentukan *co-occurrence pixel* (3,4) sudut  $0^\circ$  dan  $d=1$**



Hasil pasangan jumlah *co-occurrence pixel* (3,4) pada Gambar 4.28 untuk arah sudut  $0^\circ$  adalah sebanyak 5. Kemudian hasil perhitungan *co-occurrence* dimasukkan kedalam matriks *co-occurrence* GLCM arah sudut  $0^\circ$  seperti pada Tabel 4.7. dengan x adalah *pixel* asal dan y adalah *pixel* tetangga.

**Tabel 4.7 Matriks *co-occurrence* GLCM sudut  $0^\circ$  dari Gambar 4.28**

x,y	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	68	5
4	0	0	0	7	10

b.) Sudut  $45^\circ$

3	3	3	3	4	4	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	4	4	4	3
4	4	3	4	4	4	4	3	3	3
3	3	3	3	4	4	4	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
4	4	3	3	3	3	4	3	3	3
3	3	3	3	3	3	3	3	3	3

**Gambar 4.29 Pembentukan *co-occurrence pixel* (3,4) sudut  $45^\circ$  dan  $d=1$**

Hasil pasangan jumlah *co-occurrence pixel* (3,4) pada Gambar 4.29 untuk arah sudut  $45^\circ$  adalah sebanyak 11. Kemudian hasil perhitungan *co-occurrence* dimasukkan kedalam matriks *co-occurrence* GLCM arah sudut  $45^\circ$  seperti pada Tabel 4.8. dengan x adalah *pixel* asal dan y adalah *pixel* tetangga.

**Tabel 4.8 Matriks *co-occurrence* GLCM sudut  $45^\circ$  dari Gambar 4.29**

x,y	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	55	11
4	0	0	0	11	4

c.) Sudut 90°

3	3	3	3	4	4	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	4	4	4	3
4	4	3	4	4	4	4	3	3	3
3	3	3	3	4	4	4	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
4	4	3	3	3	3	4	3	3	3
3	3	3	3	3	3	3	3	3	3

Gambar 4.30 Pembentukan *co-occurrence pixel* (3,4) sudut 90° dan  $d=1$

Hasil pasangan jumlah *co-occurrence pixel* (3,4) pada Gambar 4.30 untuk arah sudut 90° adalah sebanyak 13. Kemudian hasil perhitungan *co-occurrence* dimasukkan kedalam matriks *co-occurrence GLCM* arah sudut 90° seperti pada Tabel 4.9. dengan  $x$  adalah *pixel* asal dan  $y$  adalah *pixel* tetangga.

Tabel 4.9 Matriks *co-occurrence GLCM* sudut 90° dari Gambar 4.30

x,y	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	62	13
4	0	0	0	11	4

d.) Sudut 135°

3	3	3	3	4	4	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	4	4	4	3
4	4	3	4	4	4	4	3	3	3
3	3	3	3	4	4	4	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
4	4	3	3	3	3	4	3	3	3
3	3	3	3	3	3	3	3	3	3

Gambar 4.31 Pembentukan *co-occurrence pixel* (3,4) sudut 135° dan  $d=1$

Hasil pasangan jumlah *co-occurrence pixel* (3,4) pada Gambar 4.31 untuk arah sudut 135° adalah sebanyak 13. Kemudian hasil perhitungan *co-occurrence* dimasukkan kedalam matriks *co-occurrence* GLCM arah sudut 135° seperti pada Tabel 4.10. dengan x adalah *pixel* asal dan y adalah *pixel* tetangga.

**Tabel 4.10** Tabel *co-occurrence* GLCM sudut 135° dari Gambar 4.31

x,y	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	54	14
4	0	0	0	10	3

2. Membentuk matriks *co-occurrence* simetris dengan menjumlahkan antara matriks *co-occurrence* GLCM dan matriks tranposenya.

a.) Sudut 0°

$$\begin{aligned}
 GLCM &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 68 & 5 \\ 0 & 0 & 0 & 7 & 10 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 68 & 7 \\ 0 & 0 & 0 & 5 & 10 \end{pmatrix} \\
 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 136 & 12 \\ 0 & 0 & 0 & 12 & 20 \end{pmatrix}
 \end{aligned}$$

b.) Sudut 45°

$$\begin{aligned}
 GLCM &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 55 & 11 \\ 0 & 0 & 0 & 11 & 4 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 55 & 11 \\ 0 & 0 & 0 & 11 & 4 \end{pmatrix} \\
 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 110 & 22 \\ 0 & 0 & 0 & 22 & 8 \end{pmatrix}
 \end{aligned}$$



c.) Sudut 90°

$$\begin{aligned}
 GLCM &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 62 & 13 \\ 0 & 0 & 0 & 11 & 4 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 62 & 11 \\ 0 & 0 & 0 & 13 & 4 \end{pmatrix} \\
 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 124 & 24 \\ 0 & 0 & 0 & 24 & 8 \end{pmatrix}
 \end{aligned}$$

d.) Sudut 135°

$$\begin{aligned}
 GLCM &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 54 & 14 \\ 0 & 0 & 0 & 10 & 3 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 54 & 10 \\ 0 & 0 & 0 & 14 & 3 \end{pmatrix} \\
 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 108 & 24 \\ 0 & 0 & 0 & 24 & 6 \end{pmatrix}
 \end{aligned}$$

3. Normalisasi terhadap hasil matriks *co-occurrence* simetris untuk memperoleh nilai probabilitas dari tiap-tiap pasangan *pixel*. Perhitungan manualisasi yaitu membagi nilai dari matriks *co-occurrence* simetris dengan jumlah total elemen nilai matriks *co-occurrence* simetris.

a.) Sudut 0°

Untuk jumlah nilai total elemen nilai matriks *co-occurrence* simetris sudut 0° adalah 180. Perhitungan matriks normalisasi dengan sudut 0° adalah sebagai berikut.

$$\begin{aligned}
 GLCM &= \frac{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 136 & 12 \\ 0 & 0 & 0 & 12 & 20 \end{pmatrix}}{180} \\
 &= \begin{pmatrix} 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,756 & 0,067 \\ 0,000 & 0,000 & 0,000 & 0,067 & 0,111 \end{pmatrix}
 \end{aligned}$$

b.) Sudut 45°

Untuk jumlah nilai total elemen nilai matriks *co-occurrence* simetris sudut 45° adalah 160. Perhitungan matriks normalisasi dengan sudut 45° adalah sebagai berikut

$$GLCM = \frac{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 110 & 22 \\ 0 & 0 & 0 & 22 & 8 \end{pmatrix}}{162}$$

$$= \begin{pmatrix} 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,679 & 0,136 \\ 0,000 & 0,000 & 0,000 & 0,136 & 0,049 \end{pmatrix}$$

c.) Sudut 90°

Untuk jumlah nilai total elemen nilai matriks *co-occurrence* simetris sudut 90° adalah 180. Perhitungan matriks normalisasi dengan sudut 90° adalah sebagai berikut

$$GLCM = \frac{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 124 & 24 \\ 0 & 0 & 0 & 24 & 8 \end{pmatrix}}{180}$$

$$= \begin{pmatrix} 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,689 & 0,133 \\ 0,000 & 0,000 & 0,000 & 0,133 & 0,044 \end{pmatrix}$$

d.) Sudut 135°

Untuk jumlah nilai total elemen nilai matriks *co-occurrence* simetris sudut 135° adalah 162. Perhitungan matriks normalisasi dengan sudut 135° adalah sebagai berikut

$$GLCM = \frac{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 108 & 24 \\ 0 & 0 & 0 & 24 & 6 \end{pmatrix}}{162}$$

$$= \begin{pmatrix} 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,000 & 0,000 \\ 0,000 & 0,000 & 0,000 & 0,667 & 0,148 \\ 0,000 & 0,000 & 0,000 & 0,148 & 0,037 \end{pmatrix}$$

4. Melakukan perhitungan untuk ekstraksi fitur tekstur dengan menggunakan hasil matriks normalisasi. Fitur yang dipakai yaitu *Angular Second Moment (ASM)*, *Contrast*, *Correlation*, *Inverse Difference Moment (IDM)*. Perhitungan ekstraksi fitur tekstur untuk sudut  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  dan  $135^\circ$  menggunakan persamaan 2.3 sampai 2.6.

a.) Sudut  $0^\circ$

$$\begin{aligned} ASM &= 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 \\ &\quad + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0,756^2 + 0,067^2 \\ &\quad + 0^2 + 0^2 + 0^2 + 0,067^2 + 0,111^2 \\ &= 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \\ &\quad + 0 + 0 + 0 + 0,571 + 0,004 + 0 + 0 + 0 + 0 + 0,004 \\ &\quad + 0,012 \\ &= 0,592 \end{aligned}$$

$$\begin{aligned} Contrast &= ((0 - 0)^2 * 0) + ((0 - 1)^2 * 0) + ((0 - 2)^2 * 0) \\ &\quad + ((0 - 3)^2 * 0) + ((0 - 4)^2 * 0) + \dots + ((4 - 3)^2 \\ &\quad * 0,067) + ((4 - 4)^2 * 0,111) \\ &= 0 + 0 + 0 + 0 + 0 + \dots + 0,067 + 0 \\ &= 0,133 \end{aligned}$$

Pada perhitungan fitur *Correlation*, sebelumnya akan dilakukan perhitungan rata-rata dari  $Pd_i$ ,  $Pd_j$  dan standar deviasi dari  $Pd_i$ ,  $Pd_j$ .

$$\begin{aligned} \mu_i &= (0 * 0) + (0 * 0) + \dots + (3 * 0,756) + (3 * 0,067) + (4 * 0) \\ &\quad + (4 * 0) + (4 * 0) + (4 * 0,067) + (4 * 0,111) \\ &= 0 + 0 + \dots + 2,268 + 0,201 + 0 + 0 + 0 + 0,268 + 0,444 \\ &= 3,178 \end{aligned}$$

$$\begin{aligned} \mu_j &= (0 * 0) + (0 * 0) + \dots + (3 * 0,756) + (3 * 0,067) + (4 * 0) \\ &\quad + (4 * 0) + (4 * 0) + (4 * 0,067) + (4 * 0,111) \\ &= 0 + 0 + \dots + 2,268 + 0,201 + 0 + 0 + 0 + 0,268 + 0,444 \\ &= 3,178 \end{aligned}$$

$$\begin{aligned} \sigma_i^2 &= ((0 - 3,178)^2 * 0) + ((1 - 3,178)^2 * 0) + ((2 - 3,178)^2 * 0) \\ &\quad + \dots + ((3 - 3,178)^2 * 0,067) + ((4 - 3,178)^2 * 0,111) \\ &= 0 + 0 + 0 + \dots + 0,002 + 0,075 \\ &= 0,292 \end{aligned}$$

$$\sigma_i = \sqrt{0,292} = 0,540$$

$$\begin{aligned}\sigma_j^2 &= ((0 - 3,178)^2 * 0) + ((1 - 3,178)^2 * 0) + ((2 - 3,178)^2 * 0) \\ &\quad + \dots + ((3 - 3,178)^2 * 0,067) + ((4 - 3,178)^2 * 0,111) \\ &= 0 + 0 + 0 + \dots + 0,002 + 0,075 \\ &= 0,292\end{aligned}$$

$$\sigma_j = \sqrt{0,292} = 0,540$$

$$\begin{aligned}\text{Correlation} &= \left( \frac{(0 - 3,178) * (0 - 3,178) * 0}{0,540 * 0,540} \right) + \dots \\ &\quad + \left( \frac{(4 - 3,178) * (3 - 3,178) * 0,067}{0,540 * 0,540} \right) \\ &\quad + \left( \frac{(4 - 3,178) * (4 - 3,178) * 0,111}{0,540 * 0,540} \right) \\ &= 0 + \dots + 0,224 + 0,257 \\ &= 0,272\end{aligned}$$

$$\begin{aligned}\text{IDM} &= \left( \left( \frac{1}{1 + ((0 - 0)^2)} \right) * 0 \right) + \dots + \left( \left( \frac{1}{1 + ((4 - 2)^2)} \right) * 0 \right) \\ &\quad + \left( \left( \frac{1}{1 + ((4 - 3)^2)} \right) * 0,067 \right) \\ &\quad + \left( \left( \frac{1}{1 + ((4 - 4)^2)} \right) * 0,111 \right) \\ &= 0 + \dots + 0 + 0,133 + 0,111 \\ &= 0,933\end{aligned}$$

b.) Sudut 45°

$$\begin{aligned}\text{ASM} &= 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 \\ &\quad + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0,679^2 + 0,136^2 \\ &\quad + 0^2 + 0^2 + 0^2 + 0,136^2 + 0,049^2 \\ &= 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \\ &\quad + 0 + 0 + 0 + 0,461 + 0,018 + 0 + 0 + 0 + 0 + 0,018 \\ &\quad + 0,002 \\ &= 0,500\end{aligned}$$

$$\begin{aligned}\text{Contrast} &= ((0 - 0)^2 * 0) + ((0 - 1)^2 * 0) + ((0 - 2)^2 * 0) \\ &\quad + ((0 - 3)^2 * 0) + ((0 - 4)^2 * 0) + \dots + ((4 - 3)^2 \\ &\quad * 0,136) + ((4 - 4)^2 * 0,049) \\ &= 0 + 0 + 0 + 0 + 0 + \dots + 0,136 + 0 \\ &= 0,271\end{aligned}$$

Pada perhitungan fitur *Correlation*, sebelumnya akan dilakukan perhitungan rata-rata dari  $Pd_i$ ,  $Pd_j$  dan standar deviasi dari  $Pd_i$ ,  $Pd_j$ .

$$\begin{aligned} \mu_i &= (0 * 0) + (0 * 0) + \dots + (3 * 0,756) + (3 * 0,067) + (4 * 0) \\ &\quad + (4 * 0) + (4 * 0) + (4 * 0,067) + (4 * 0,111) \\ &= 0 + 0 + \dots + 2,268 + 0,201 + 0 + 0 + 0 + 0,268 + 0,444 \\ &= 3,185 \end{aligned}$$

$$\begin{aligned} \mu_j &= (0 * 0) + (0 * 0) + \dots + (3 * 0,756) + (3 * 0,067) + (4 * 0) \\ &\quad + (4 * 0) + (4 * 0) + (4 * 0,067) + (4 * 0,111) \\ &= 0 + 0 + \dots + 2,268 + 0,201 + 0 + 0 + 0 + 0,268 + 0,444 \\ &= 3,185 \end{aligned}$$

$$\begin{aligned} \sigma_i^2 &= ((0 - 3,185)^2 * 0) + ((1 - 3,185)^2 * 0) + ((2 - 3,185)^2 * 0) \\ &\quad + \dots + ((3 - 3,185)^2 * 0,067) + ((4 - 3,185)^2 * 0,111) \\ &= 0 + 0 + 0 + \dots + 0,002 + 0,075 \\ &= 0,281 \end{aligned}$$

$$\sigma_i = \sqrt{0,281} = 0,530$$

$$\begin{aligned} \sigma_j^2 &= ((0 - 3,185)^2 * 0) + ((1 - 3,185)^2 * 0) + ((2 - 3,185)^2 * 0) \\ &\quad + \dots + ((3 - 3,185)^2 * 0,067) + ((4 - 3,185)^2 * 0,111) \\ &= 0 + 0 + 0 + \dots + 0,002 + 0,075 \\ &= 0,281 \end{aligned}$$

$$\sigma_j = \sqrt{0,281} = 0,530$$

$$\begin{aligned} \text{Correlation} &= \left( \frac{(0 - 3,185) * (0 - 3,185) * 0}{0,530 * 0,530} \right) + \dots \\ &\quad + \left( \frac{(4 - 3,185) * (3 - 3,185) * 0,679}{0,530 * 0,530} \right) \\ &\quad + \left( \frac{(4 - 3,178) * (4 - 3,178) * 0,049}{0,530 * 0,530} \right) \\ &= 0 + \dots + 0,224 + 0,257 \\ &= 0,053 \end{aligned}$$

$$\begin{aligned} IDM &= \left( \left( \frac{1}{1 + ((0 - 0)^2)} \right) * 0 \right) + \dots + \left( \left( \frac{1}{1 + ((4 - 2)^2)} \right) * 0 \right) \\ &\quad + \left( \left( \frac{1}{1 + ((4 - 3)^2)} \right) * 0,136 \right) \\ &\quad + \left( \left( \frac{1}{1 + ((4 - 4)^2)} \right) * 0,049 \right) \\ &= 0 + \dots + 0 + 0,068 + 0,049 \\ &= 0,864 \end{aligned}$$

c.) Sudut 90°

$$\begin{aligned}
 ASM &= 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 \\
 &\quad + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0,689^2 + 0,133^2 \\
 &\quad + 0^2 + 0^2 + 0^2 + 0,133^2 + 0,044^2 \\
 &= 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \\
 &\quad + 0 + 0 + 0 + 0,475 + 0,018 + 0 + 0 + 0 + 0 + 0,018 \\
 &\quad + 0,002 \\
 &= 0,512
 \end{aligned}$$

$$\begin{aligned}
 Contrast &= ((0 - 0)^2 * 0) + ((0 - 1)^2 * 0) + ((0 - 2)^2 * 0) \\
 &\quad + ((0 - 3)^2 * 0) + ((0 - 4)^2 * 0) + \dots + ((4 - 3)^2 \\
 &\quad * 0,133) + ((4 - 4)^2 * 0,044) \\
 &= 0 + 0 + 0 + 0 + 0 + \dots + 0,133 + 0 \\
 &= 0,267
 \end{aligned}$$

Pada perhitungan fitur *Correlation*, sebelumnya akan dilakukan perhitungan rata-rata dari  $Pd_i$ ,  $Pd_j$  dan standar deviasi dari  $Pd_i$ ,  $Pd_j$ .

$$\begin{aligned}
 \mu_i &= (0 * 0) + (0 * 0) + \dots + (3 * 0,689) + (3 * 0,133) + (4 * 0) \\
 &\quad + (4 * 0) + (4 * 0) + (4 * 0,133) + (4 * 0,044) \\
 &= 0 + 0 + \dots + 2,067 + 0,399 + 0 + 0 + 0 + 0,533 + 0,176 \\
 &= 3,178
 \end{aligned}$$

$$\begin{aligned}
 \mu_j &= (0 * 0) + (0 * 0) + \dots + (3 * 0,689) + (3 * 0,133) + (4 * 0) \\
 &\quad + (4 * 0) + (4 * 0) + (4 * 0,689) + (4 * 0,044) \\
 &= 0 + 0 + \dots + 2,067 + 0,399 + 0 + 0 + 0 + 2,756 + 0,176 \\
 &= 3,178
 \end{aligned}$$

$$\begin{aligned}
 \sigma_i^2 &= ((0 - 3,178)^2 * 0) + ((1 - 3,178)^2 * 0) + ((2 - 3,178)^2 * 0) \\
 &\quad + \dots + ((3 - 3,178)^2 * 0,131) + ((4 - 3,178)^2 * 0,050) \\
 &= 0 + 0 + 0 + \dots + 0,065 + 0,173 \\
 &= 0,270
 \end{aligned}$$

$$\sigma_i = \sqrt{0,270} = 0,520$$

$$\begin{aligned}
 \sigma_j^2 &= ((0 - 3,178)^2 * 0) + ((1 - 3,178)^2 * 0) + ((2 - 3,178)^2 * 0) \\
 &\quad + \dots + ((3 - 3,178)^2 * 0,131) + ((4 - 3,178)^2 * 0,050) \\
 &= 0 + 0 + 0 + \dots + 0,065 + 0,173 \\
 &= 0,270
 \end{aligned}$$

$$\sigma_j = \sqrt{0,270} = 0,520$$

$$\begin{aligned}
 \text{Correlation} &= \left( \frac{(0 - 3,178) * (0 - 3,178) * 0}{0,520 * 0,520} \right) + \dots \\
 &+ \left( \frac{(4 - 3,178) * (3 - 3,178) * 0,133}{0,520 * 0,520} \right) \\
 &+ \left( \frac{(4 - 3,178) * (4 - 3,178) * 0,044}{0,520 * 0,520} \right) \\
 &= 0 + \dots + 0,039 + 0,111 \\
 &= 0,047
 \end{aligned}$$

$$\begin{aligned}
 \text{IDM} &= \left( \left( \frac{1}{1 + ((0 - 0)^2)} \right) * 0 \right) + \dots + \left( \left( \frac{1}{1 + ((4 - 2)^2)} \right) * 0 \right) \\
 &+ \left( \left( \frac{1}{1 + ((4 - 3)^2)} \right) * 0,133 \right) \\
 &+ \left( \left( \frac{1}{1 + ((4 - 4)^2)} \right) * 0,044 \right) \\
 &= 0 + \dots + 0 + 0,267 + 0,044 \\
 &= 0,866
 \end{aligned}$$

d.) Sudut 135°

$$\begin{aligned}
 \text{ASM} &= 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 \\
 &+ 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0,667^2 + 0,148^2 \\
 &+ 0^2 + 0^2 + 0^2 + 0,148^2 + 0,037^2 \\
 &= 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \\
 &+ 0 + 0 + 0 + 0,445 + 0,022 + 0 + 0 + 0 + 0 + 0,022 \\
 &+ 0,001 \\
 &= 0,489
 \end{aligned}$$

$$\begin{aligned}
 \text{Contrast} &= ((0 - 0)^2 * 0) + ((0 - 1)^2 * 0) + ((0 - 2)^2 * 0) \\
 &+ ((0 - 3)^2 * 0) + ((0 - 4)^2 * 0) + \dots + ((4 - 3)^2 \\
 &* 0,148) + ((4 - 4)^2 * 0,037) \\
 &= 0 + 0 + 0 + 0 + 0 + \dots + 0,148 + 0 \\
 &= 0,296
 \end{aligned}$$

Pada perhitungan fitur *Correlation*, sebelumnya akan dilakukan perhitungan rata-rata dari  $Pd_i$ ,  $Pd_j$  dan standar deviasi dari  $Pd_i$ ,  $Pd_j$ .

$$\begin{aligned}
 \mu_i &= (0 * 0) + (0 * 0) + \dots + (3 * 0,667) + (3 * 0,148) + (4 * 0) \\
 &+ (4 * 0) + (4 * 0) + (4 * 0,148) + (4 * 0,037) \\
 &= 0 + 0 + \dots + 2,001 + 0,444 + 0 + 0 + 0 + 0,592 + 0,148 \\
 &= 3,185
 \end{aligned}$$

$$\begin{aligned}
 \mu_j &= (0 * 0) + (0 * 0) + \dots + (3 * 0,667) + (3 * 0,148) + (4 * 0) \\
 &+ (4 * 0) + (4 * 0) + (4 * 0,148) + (4 * 0,037) \\
 &= 0 + 0 + \dots + 2,001 + 0,444 + 0 + 0 + 0 + 0,592 + 0,148
 \end{aligned}$$

$$\begin{aligned}
 &= 3,185 \\
 \sigma_i^2 &= ((0 - 3,185)^2 * 0) + ((1 - 3,185)^2 * 0) + ((2 - 3,185)^2 * 0) \\
 &\quad + \dots + ((3 - 3,185)^2 * 0,148) + ((4 - 3,185)^2 * 0,037) \\
 &= 0 + 0 + 0 + \dots + 0,172 + 0,157 \\
 &= 0,255 \\
 \sigma_i &= \sqrt{0,255} = 0,504 \\
 \sigma_j^2 &= ((0 - 3,185)^2 * 0) + ((1 - 3,185)^2 * 0) + ((2 - 3,185)^2 * 0) \\
 &\quad + \dots + ((3 - 3,185)^2 * 0,148) + ((4 - 3,185)^2 * 0,037) \\
 &= 0 + 0 + 0 + \dots + 0,172 + 0,157 \\
 &= 0,255 \\
 \sigma_j &= \sqrt{0,255} = 0,504 \\
 \text{Correlation} &= \left( \frac{(0 - 3,185) * (0 - 3,185) * 0}{0,504 * 0,504} \right) + \dots \\
 &\quad + \left( \frac{(4 - 3,185) * (3 - 3,185) * 0,148}{0,504 * 0,504} \right) \\
 &\quad + \left( \frac{(4 - 3,185) * (4 - 3,185) * 0,037}{0,504 * 0,504} \right) \\
 &= 0 + \dots + 0,008 + 0,091 \\
 &= 0,10 \\
 \text{IDM} &= \left( \left( \frac{1}{1 + ((0 - 0)^2)} \right) * 0 \right) + \dots \left( \left( \frac{1}{1 + ((4 - 2)^2)} \right) * 0 \right) \\
 &\quad + \left( \left( \frac{1}{1 + ((4 - 3)^2)} \right) * 0,148 \right) \\
 &\quad + \left( \left( \frac{1}{1 + ((4 - 4)^2)} \right) * 0,037 \right) \\
 &= 0 + \dots + 0 + 0,296 + 0,037 \\
 &= 0,851
 \end{aligned}$$

### 4.3.2 Perhitungan Manual Proses SVM

#### 4.3.2.1 Perhitungan Manual Proses *Sequential Training* SVM

Pada proses perhitungan manual proses *training* SVM ini menggunakan metode *sequential training* yang akan mengklasifikasikan data dari sekumpulan objek. Diketahui *hyperplane* yang merupakan fungsi pemisah dengan nilai *margin* yang maksimal. *Hyperplane* merupakan bidang pemisah yang terletak di tengah antara dua set objek dari dua kelas. Metode *sequential* digunakan untuk mendapatkan nilai *margin* yang maksimal.

Data latih yang digunakan dapat dilihat pada Tabel 4.11 Pada data ke-1 sampai dengan data ke-12. Data latih tersebut kemudian digunakan untuk menghitung nilai *kernel*. Data baris satu sampai dua menunjukkan bahwa data tersebut masuk ke dalam kelas Tipe I, baris tiga sampai empat masuk ke dalam kelas Tipe I', baris lima sampai enam masuk ke dalam kelas Tipe II, baris tujuh sampai delapan masuk ke dalam kelas Tipe III, baris sembilan sampai sepuluh





masuk ke dalam kelas Tipe IV dan baris sebelas sampai duabelas masuk ke dalam kelas Tipe V. Karena data latih yang dimiliki memiliki enam kelas, maka diperlukan adanya strategi *One-Against-All*, sehingga untuk level 1 kelas pertama diindikasikan sebagai 1 dan kelas lainnya adalah -1.

**Tabel 4.11 Data Latih Level 1**

Data	Nama	ASM 0°	CON 0°	COR 0°	IDM 0°	...	COR 135°	IDM 135°	Tipe Bibir	Kelas
1	TI'B13K3W	0,849	0,016	0,321	0,992	...	0,286	0,988	I'	1
2	TI'B1K3W	0,615	0,018	0,381	0,991	...	0,334	0,980	I'	1
3	TIB2K7W	0,624	0,068	0,403	0,966	...	0,369	0,958	I	-1
4	TIB3K3L	0,601	0,026	0,372	0,987	...	0,358	0,978	I	-1
5	TIIB11K2L	0,845	0,061	0,157	0,969	...	0,125	0,965	II	-1
6	TIIB2K2W	0,422	0,075	0,301	0,963	...	0,292	0,950	II	-1
7	TIIB4K2L	0,375	0,142	0,279	0,929	...	0,282	0,927	III	-1
8	TIIB4K6L	0,423	0,097	0,285	0,951	...	0,281	0,947	III	-1
9	TIVB5K6L	0,677	0,051	0,407	0,975	...	0,370	0,966	IV	-1
10	TIVB9K6L	0,483	0,084	0,410	0,958	...	0,370	0,941	IV	-1
11	TVB11K6L	0,433	0,089	0,271	0,956	...	0,267	0,948	V	-1
12	TVB13K3L	0,618	0,029	0,485	0,985	...	0,467	0,981	V	-1

*Kernel* yang digunakan dalam penelitian ini adalah *Kernel Polynomial*. Hasil perhitungan keseluruhan *kernel Polynomial* dapat dilihat pada Tabel 4.15. Berikut adalah contoh perhitungan *kernel Polynomial*.

Tabel 4.12 adalah Tabel data latih dengan data ke 1 untuk perhitungan dengan *kernel Polynomial*.

**Tabel 4.12 Data Latih ke-1**

Data	Nama	ASM 0°	CON 0°	COR 0°	HOM 0°	ASM 45°	...	COR 135°	HOM 135°
1	TI'B13K3W	0,849	0,016	0,321	0,992	0,853	...	0,286	0,988

$k(x, y)$  misal,  $x = x_1, y = x_1$ , maka  $k(x, y) =$

$$k(x_1, x_1) = (x_1, x_1)^2 = (7,198)^2 = 51,815$$



$$(x_1, x_1) = \begin{pmatrix} 0,849 \\ 0,016 \\ 0,321 \\ 0,992 \\ 0,853 \\ 0,021 \\ 0,299 \\ 0,990 \\ 0,855 \\ 0,017 \\ 0,313 \\ 0,991 \\ 0,850 \\ 0,024 \\ 0,286 \\ 0,988 \end{pmatrix} \cdot \begin{pmatrix} 0,849 \\ 0,016 \\ 0,321 \\ 0,992 \\ 0,853 \\ 0,021 \\ 0,299 \\ 0,990 \\ 0,855 \\ 0,017 \\ 0,313 \\ 0,991 \\ 0,850 \\ 0,024 \\ 0,286 \\ 0,988 \end{pmatrix}$$

$$\begin{aligned} (x_1, x_1) &= (0,849 * 0,849) + (0,016 * 0,016) + (0,321 * 0,321) \\ &\quad + (0,992 * 0,992) + (0,853 * 0,853) + (0,021 * 0,021) \\ &\quad + (0,299 * 0,299) + (0,990 * 0,990) + (0,855 * 0,855) \\ &\quad + (0,017 * 0,017) + (0,313 * 0,313) + (0,991 * 0,991) \\ &\quad + (0,850 * 0,850) + (0,024 * 0,024) + (0,286 * 0,286) \\ &\quad + (0,988 * 0,988) \end{aligned}$$

$$(x_1, x_1) = 7,198$$

Tabel 4.13 adalah Tabel data latih dengan data ke 2 untuk perhitungan dengan *kernel Polynomial*.

**Tabel 4.13 Data Latih ke-2**

Data	Nama	ASM 0°	CON 0°	COR 0°	HOM 0°	ASM 45°	...	COR 135°	HOM 135°
2	TI'B1K3W	0,615	0,018	0,381	0,991	0,607	...	0,334	0,980

$k(x, y)$  misal,  $x = x_1, y = x_2$ , maka  $k(x, y) =$

$$k(x_1, x_2) = (x_1, x_2)^2 = (6,406)^2 = 41,042$$

$$(x_1, x_2) = \begin{pmatrix} 0,849 \\ 0,016 \\ 0,321 \\ 0,992 \\ 0,853 \\ 0,021 \\ 0,299 \\ 0,990 \\ 0,855 \\ 0,017 \\ 0,313 \\ 0,991 \\ 0,850 \\ 0,024 \\ 0,286 \\ 0,988 \end{pmatrix} \cdot \begin{pmatrix} 0,615 \\ 0,018 \\ 0,381 \\ 0,991 \\ 0,607 \\ 0,033 \\ 0,346 \\ 0,983 \\ 0,609 \\ 0,031 \\ 0,350 \\ 0,985 \\ 0,602 \\ 0,039 \\ 0,334 \\ 0,980 \end{pmatrix}$$

$$\begin{aligned} (x_1, x_2) &= (0,849 * 0,615) + (0,016 * 0,018) + (0,321 * 0,381) \\ &\quad + (0,992 * 0,991) + (0,853 * 0,607) + (0,021 * 0,033) \\ &\quad + (0,299 * 0,346) + (0,990 * 0,983) + (0,855 * 0,609) \\ &\quad + (0,017 * 0,031) + (0,313 * 0,350) + (0,991 * 0,985) \\ &\quad + (0,850 * 0,602) + (0,024 * 0,039) + (0,286 * 0,334) \\ &\quad + (0,988 * 0,980) \end{aligned}$$

$$(x_1, x_2) = 6,406$$

Tabel 4.14 adalah Tabel data latih dengan data ke 3 untuk perhitungan dengan *kernel Polynomial*.

**Tabel 4.14 Data Latih ke-3**

Data	Nama	ASM 0°	CON 0°	COR 0°	HOM 0°	ASM 45°	...	COR 135°	HOM 135°
3	TIB2K7W	0,624	0,068	0,403	0,966	0,614	...	0,369	0,958

$k(x, y)$  misal,  $x = x_1, y = x_3$ , maka  $k(x, y) =$

$$k(x_1, x_3) = (x_1, x_3)^2 = (6,430)^2 = 41,350$$

$$(x_1, x_3) = \begin{pmatrix} 0,849 \\ 0,016 \\ 0,321 \\ 0,992 \\ 0,853 \\ 0,021 \\ 0,299 \\ 0,990 \\ 0,855 \\ 0,017 \\ 0,313 \\ 0,991 \\ 0,850 \\ 0,024 \\ 0,286 \\ 0,988 \end{pmatrix} \cdot \begin{pmatrix} 0,624 \\ 0,068 \\ 0,403 \\ 0,966 \\ 0,614 \\ 0,086 \\ 0,367 \\ 0,957 \\ 0,644 \\ 0,052 \\ 0,435 \\ 0,974 \\ 0,615 \\ 0,085 \\ 0,369 \\ 0,958 \end{pmatrix}$$

$$(x_1, x_3) = (0,849 * 0,624) + (0,016 * 0,068) + (0,321 * 0,403) + (0,992 * 0,966) + (0,853 * 0,614) + (0,021 * 0,086) + (0,299 * 0,367) + (0,990 * 0,975) + (0,855 * 0,644) + (0,017 * 0,052) + (0,313 * 0,435) + (0,991 * 0,974) + (0,850 * 0,615) + (0,024 * 0,085) + (0,286 * 0,369) + (0,988 * 0,958)$$

$$(x_1, x_3) = 6,430$$

Tabel 4.15 merupakan hasil perhitungan data latih menggunakan *kernel polynomial*.

**Tabel 4.15 Hasil Perhitungan Data Latih Level 1 dengan *Kernel Polynomial***

K(x,y)	1	2	3	4	5	6	...	11	12
1	51,815	41,042	41,350	40,285	47,540	30,984	...	31,206	43,262
2	41,042	34,364	34,571	33,893	37,005	27,071	...	27,071	36,617
3	41,350	34,571	35,024	34,123	37,120	27,172	...	27,162	37,111
4	40,285	33,893	34,123	33,453	36,233	26,795	...	26,777	36,195
5	47,540	37,005	37,120	36,233	44,921	28,075	...	28,447	38,134
6	30,984	27,071	27,172	26,795	28,075	22,327	...	22,274	28,733
7	28,460	25,061	25,260	24,835	25,888	20,929	...	20,882	26,614
8	31,128	27,080	27,206	26,796	28,293	22,302	...	22,271	28,706
9	43,518	35,980	36,377	35,472	39,200	28,020	...	28,042	38,481
10	34,250	29,751	30,101	29,460	30,528	24,155	...	24,049	32,056
11	31,206	27,071	27,162	26,777	28,447	22,274	...	22,255	28,619
12	43,262	36,617	37,111	36,195	38,134	28,733	...	28,619	39,765

Setelah nilai perhitungan *kernel* didapatkan, selanjutnya melakukan perhitungan metode *sequential training SVM* untuk mendapatkan parameter yang lainnya. Tahapan perhitungan *sequential training SVM* seperti berikut.



1. Inialisasi  $a_i = 0$ .
2. Menghitung matriks *hessian* seperti pada Persamaan 2.16, yaitu  $D_{ij} = y_i y_j (k(x_i, x_j) + \lambda^2)$ , dimana  $y_i = 1$  adalah kelas pada atribut pertama dan  $y_j = -1$  pada atribut kedua.  $(x_i, x_j)$  adalah *kernel* yang telah dihitung pada tahap sebelumnya. Inialisasi nilai *lambda*  $\lambda = 0.5$ . contoh perhitungan data ke-1 data ke-2 dan data ke-3 adalah sebagai berikut:

$$D_{11} = y_1 y_1 (k(x_1, x_1) + \lambda^2)$$

$$= 1 * 1 (51,815 + 0,5^2)$$

$$= 1 * (51,815 + 0,25)$$

$$= 52,06542871$$

$$D_{12} = y_1 y_2 (k(x_1, x_2) + \lambda^2)$$

$$= 1 * 1 (41,042 + 0,5^2)$$

$$= 1 * (41,042 + 0,25)$$

$$= 41,29199647$$

$$D_{13} = y_1 y_3 (k(x_1, x_3) + \lambda^2)$$

$$= 1 * -1 (41,350 + 0,5^2)$$

$$= 1 * (41,350 + 0,25)$$

$$= -41,59959724$$

Tabel 4.16 adalah hasil perhitungan matriks *hessian* untuk data latih perhitungan manual.

**Tabel 4.16 Hasil Perhitungan Matriks *Hessian* Level 1**

Dij	1	2	3	...	11	12
1	52,06542871	41,29199647	-41,59959724	...	-31,45608155	-43,5123025
2	41,29199647	34,61358818	-34,82139643	...	-27,32137017	-36,86673822
3	-41,59959724	-34,82139643	35,27400842	...	27,41157149	37,36109628
4	-40,53473633	-34,14288066	34,37309616	...	27,02669487	36,44494509
5	-47,7903613	-37,25533093	37,370133	...	28,69707116	38,38406731
6	-31,23357946	-27,32060585	27,42192524	...	22,52379646	28,98251946
7	-28,70975557	-25,31115829	25,50993274	...	21,13243621	26,86412325
8	-31,37844925	-27,32987915	27,4561494	...	22,52111645	28,95573655
9	-43,7681285	-36,23008858	36,62696544	...	28,29187767	38,731072
10	-34,50016962	-30,00061129	30,35124773	...	24,29947216	32,30630453
11	-31,45608155	-27,32137017	27,41157149	...	22,50478016	28,86867924
12	-43,5123025	-36,86673822	37,36109628	...	28,86867924	40,0149867

Setelah menghitung nilai matriks *hessian*, langkah selanjutnya adalah menghitung nilai *error* menggunakan persamaan 2.17.

3. Dilakukan iterasi untuk setiap  $i = 1, 2, \dots, n$ . pada proses perhitungan ini dilakukan iterasi (*itermax*) = 100,  $C = 1$ , dan  $y$  adalah hasil konstanta. Dalam



penelitian ini nilai  $y = 0.01$ . untuk mendapatkan nilai  $y$  digunakan rumus konstanta dibagi dengan nilai max diagonal matriks *hessian*.

Iterasi ke 1 :

- a.) Menghitung persamaan 2.17. Inialisasi awal  $\alpha_1=0$ . Contoh perhitungan untuk data ke-1 adalah sebagai berikut

$$\begin{aligned}
 E_1 &= ((0 * 52,06542871) + (0 * 41,29199647) + (0 * (-41,59959724)) \\
 &\quad + (0 * (-40,53473633)) + (0 * (-47,7903613)) \\
 &\quad + (0 * (-31,23357946)) + (0 * (-28,70975557)) \\
 &\quad + (0 * (-31,37844925)) + (0 * (-43,7681285)) \\
 &\quad + (0 * (-34,50016962)) + (0 * (-31,45608155)) \\
 &\quad + (0 * (-43,5123025))) \\
 &= 0
 \end{aligned}$$

Perhitungan dilakukan untuk seluruh data latih, hasilnya dapat dilihat pada Tabel 4.17.

**Tabel 4.17 Hasil Perhitungan Ei Level 1**

$\alpha_i$ Dij	1	2	3	4	5	6	7	8	9	10	11	12	Ei
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0

- b.) Menghitung nilai  $\delta\alpha_i$  menggunakan persamaan 2.18 . contoh perhitungan untuk data ke-1 adalah sebagai berikut :

$$\begin{aligned}
 \delta\alpha_1 &= \min\{\text{maks}[\gamma(1 - E_1), -\alpha_1], C - \alpha_1\} \\
 &= \min\{\text{maks}[0.01(1 - 0), -0], 1 - 0\} \\
 &= \min\{\text{maks}[0.01(1), -0], 1\} \\
 &= \min\{\text{maks}[0.01, -0], 1\} \\
 &= \min\{\text{maks}[0.01], 1\} \\
 &= 0.01
 \end{aligned}$$

Perhitungan dilakukan untuk seluruh data latih, hasilnya dapat dilihat pada Tabel 4.18.



Tabel 4.18 Hasil Perhitungan  $\delta\alpha_i$  pada Iterasi ke 1 Level 1

Data	$\delta\alpha_i$
1	0.01
2	0.01
3	0.01
4	0.01
5	0.01
6	0.01
7	0.01
8	0.01
9	0.01
10	0.01
11	0.01
12	0.01

c.) Memperbarui nilai  $\alpha_1$  dengan persamaan 2.19. Contoh perhitungan untuk data ke-1 adalah sebagai berikut :

$$\alpha_1 = \alpha_1 + \delta\alpha_1$$

$$\alpha_1 = 0 + 0.01 = 0.01$$

perhitungan dilakukan untuk seluruh data latih, hasilnya dapat dilihat pada Tabel 4.19.

Tabel 4.19 Hasil Perhitungan  $\alpha_i$  pada Iterasi ke 1 Level 1

Data	$\alpha_i$
1	0.01
2	0.01
3	0.01
4	0.01
5	0.01
6	0.01
7	0.01
8	0.01
9	0.01
10	0.01
11	0.01
12	0.01

d.) Selanjutnya melakukan perhitungan  $E_i$  iterasi ke 2. Langkahnya sama dengan langkah pada iterasi ke 1, tetapi  $\alpha_1$  yang digunakan adalah  $\alpha_1$  yang ada pada Tabel 4.19. Hasil perhitungan keseluruhan untuk iterasi ke 2 ada pada Tabel 4.20.

**Tabel 4.20 Hasil Perhitungan Nilai  $E_i$  Level 1 pada iterasi ke 2**

$\alpha_i$ Dij	1	2	3	...	12	$E_i$
1	0,520654287	0,412919965	-0,415995972	...	-0,435123025	-2,811257361
2	0,412919965	0,346135882	-0,348213964	...	-0,368667382	-2,406944749
3	-0,415995972	-0,348213964	0,352740084	...	0,373610963	2,427351322
4	-0,405347363	-0,341428807	0,343730962	...	0,364449451	2,379599851
5	-0,477903613	-0,372553309	0,37370133	...	0,383840673	2,542923683
6	-0,312335795	-0,273206058	0,274219252	...	0,289825195	1,94726701
7	-0,287097556	-0,253111583	0,255099327	...	0,268641232	1,821608972
8	-0,313784493	-0,273298791	0,274561494	...	0,289557365	1,94777553
9	-0,437681285	-0,362300886	0,366269654	...	0,38731072	2,510653491
10	-0,345001696	-0,300006113	0,303512477	...	0,323063045	2,126935503
11	-0,314560815	-0,273213702	0,274115715	...	0,288686792	1,945000442
12	-0,435123025	-0,368667382	0,373610963	...	0,400149867	2,565344897

e.) Menghitung nilai  $\delta\alpha_1$  untuk iterasi ke 2 menggunakan persamaan 2.18. Perhitungan dilakukan untuk seluruh data latih, hasilnya dapat dilihat pada Tabel 4.21.

**Tabel 4.21 Hasil Perhitungan  $\delta\alpha_i$  pada Iterasi ke 2 Level 1**

Data	$\delta\alpha_i$
1	0,038112574
2	0,034069447
3	-0,01
4	-0,01
5	-0,01
6	-0,00947267
7	-0,00821609
8	-0,009477755
9	-0,01
10	-0,01
11	-0,009450004
12	-0,01



- f.) Memperbarui nilai  $\alpha_1$  dengan persamaan 2.19. Perhitungan dilakukan untuk seluruh data latih, hasilnya dapat dilihat pada Tabel 4.22 di bawah.
- g.) Setiap iterasi yang dilakukan akan dilakukan pengecekan. Apabila nilai *support vector* sudah mencapai iterasi maksimum, maka proses perhitungan dihentikan.

**Tabel 4.22 Hasil Perhitungan  $\alpha_i$  pada Iterasi ke 2 Level 1**

Data	$\alpha_i$
1	0,048112574
2	0,044069447
3	0
4	0
5	0
6	0,00052733
7	0,00178391
8	0,000522245
9	0
10	0
11	0,000549996
12	0

Setelah perhitungan data latih menggunakan *sequential training*, tahap selanjutnya adalah mencari nilai  $w$  dengan menggunakan proses *sequential training*. Nilai  $w \cdot x^+$  memiliki nilai bobot dan nilai awal yang lebih besar. Sedangkan nilai  $w \cdot x^-$  adalah nilai dari salah satu *support vector* dari kelas negatif.  $x^-$  memiliki nilai bobot dan nilai awal yang lebih kecil. Pada data latih penelitian ini, untuk level pertama pada kelas positif adalah data dari kelas 1 yaitu kelas Tipe I, sedangkan untuk kelas negative adalah kelas 2,3,4,5,6 yaitu Tipe I', Tipe II, Tipe III, Tipe IV, Tipe V.

Contoh perhitungan untuk  $K(x_i, x^+)$  dan untuk perhitungan  $K(x_i, x^-)$  dilakukan dengan contoh yang sama. Nilai  $x_i$  untuk kelas  $x^+$  didapatkan dari nilai  $\alpha_i$  terbesar dari kelas positif yaitu 0,048112574 dan nilai  $x_i$  untuk kelas  $x^-$  didapatkan nilai  $\alpha_i$  terbesar dari kelas negative yaitu 0,00178391 . Untuk hasil perhitungan secara keseluruhan untuk mendapatkan nilai  $K(x_i, x^+)$  dan nilai  $K(x_i, x^-)$  dapat dilihat pada Tabel 4.24.



Tabel 4.23 Hasil Perhitungan  $K(x_i, x^+)$  dan nilai  $K(x_i, x^-)$  Level 1

Data ke-	$K(x_i, x^+)$	$K(x_i, x^-)$
1	51,815	28,460
2	41,042	25,061
3	41,350	25,260
4	40,285	24,835
5	47,540	25,888
6	30,984	20,929
7	28,460	19,775
8	31,128	20,929
9	43,518	25,958
10	34,250	22,613
11	31,206	20,882
12	43,262	26,614

Setelah didapatkan nilai dari masing-masing kelas, langkah selanjutnya adalah menghitung nilai  $w$ . berikut akan dijelaskan untuk salah satu data menggunakan Persamaan 2.11, dengan hasil perhitungan untuk kelas positif dan kelas negative adalah sebagai berikut.

$$w.x^+ = \sum_{t=1}^n \alpha_i y_{ik}(x_i, x^+)$$

$$\begin{aligned} w.x^+ &= (0,048112574 * 1 * 51,815) + (0,044069447 * 1 * 41,042) \\ &\quad + (0 * 1 * 41,350) + (0 * -1 * 40,285) + (0 * -1 * 47,540) \\ &\quad + (0,00052733 * -1 * 30,984) + (0,00178391 * -1 * 28,460) \\ &\quad + (0,000522245 * -1 * 31,128) + (0 * -1 * 43,518) \\ &\quad + (0 * -1 * 34,250) + (0,000549996 * -1 * 31,206) \\ &\quad + (0 * -1 * 43,262) \end{aligned}$$

$$w.x^+ = 4,201143643$$

$$\begin{aligned} w.x^- &= (0,048112574 * 1 * 28,460) + (0,044069447 * 1 * 25,061) \\ &\quad + (0 * 1 * 25,260) + (0 * -1 * 24,835) + (0 * -1 * 25,888) \\ &\quad + (0,00052733 * -1 * 20,929) + (0,00178391 * -1 * 19,775) \\ &\quad + (0,000522245 * -1 * 20,929) + (0 * -1 * 25,958) \\ &\quad + (0 * -1 * 22,613) + (0,000549996 * -1 * 20,882) \\ &\quad + (0 * -1 * 26,614) \end{aligned}$$



$$w.x^- = 2,404974725$$

Untuk hasil perhitungan secara keseluruhan mendapatkan nilai  $w.x^+$  dan  $w.x^-$  dapat dilihat pada Tabel 4.25.

**Tabel 4.24 Hasil Perhitungan  $w.x^+$  dan nilai  $w.x^-$  Level 1**

Data ke-	$w.x^+$	$w.x^-$
1	2,492973628	1,369272085
2	1,808698108	1,104431399
3	0	0
4	0	0
5	0	0
6	-0,016338568	-0,011036275
7	-0,050769651	-0,03527735
8	-0,016256668	-0,010929886
9	0	0
10	0	0
11	-0,017163207	-0,011485248
12	0	0
$\Sigma$	4,201143643	2,404974725

Langkah selanjutnya adalah mencari nilai  $b$  untuk Level 1 dengan menggunakan Persamaan 2.10.

$$\begin{aligned}
 b &= -\frac{1}{2}(w.x^+ + w.x^-) \\
 &= -\frac{1}{2}(4,201143643 + 2,404974725) = -3,303059184
 \end{aligned}$$

Setelah didapatkan nilai  $b$  untuk level 1, proses selanjutnya adalah menghitung level 2. Untuk level 2 data latih yang digunakan adalah data untuk Tipe I' bernilai 1 dan Tipe II, Tipe III, Tipe IV, Tipe V yang bernilai -1. Langkah perhitungan yang dilakukan sama seperti pada level 1.

#### 4.3.2.2 Perhitungan Manual Proses *Testing* SVM

Setelah dilakukan proses *training*, langkah selanjutnya adalah melakukan proses *testing* menggunakan data uji. Tahap proses *testing* SVM dilakukan untuk mendapatkan nilai  $f(x)$ . Data uji yang digunakan pada perhitungan manual dapat dilihat pada Tabel 4.26.



Tabel 4.25 Data Uji

Data	Nama	ASM 0°	CON 0°	COR 0°	IDM 0°	ASM 45°	...	ASM 135°	CON 135°	COR 135°	IDM 135°
13	B12K2L	0,619	0,097	0,264	0,952	0,609	...	0,023	215,788	0,797	0,134
14	B12K3L	0,935	0,020	0,281	0,990	0,936	...	0,044	21,039	0,934	0,280
15	B12K6L	0,600	0,128	0,207	0,936	0,597	...	0,026	115,353	0,883	0,169
16	B12K7L	0,469	0,079	0,384	0,960	0,453	...	0,030	72,378	0,903	0,207

Untuk contoh proses *testing SVM* berikut langkah hasil perhitungannya menggunakan data ke-13. Pertama adalah menghitung *kernel Polynomial* untuk data uji 13 dengan semua data latih. Berikut adalah perhitungan *kernel Polynomial* untuk data ke-1 dengan data ke-13.

$k(x_i, x)$  misal,  $x_i = x_1, x = x_{13}$ , maka

$$k(x_1, x_{13}) = (x_1, x_{13})^2 = (6,216943)^2 = 38,65039045$$

$$(x_1, x_{13}) = \begin{pmatrix} 0,849 \\ 0,016 \\ 0,321 \\ 0,992 \\ 0,853 \\ 0,021 \\ 0,299 \\ 0,990 \\ 0,855 \\ 0,017 \\ 0,313 \\ 0,991 \\ 0,850 \\ 0,024 \\ 0,286 \\ 0,988 \end{pmatrix} \cdot \begin{pmatrix} 0,619 \\ 0,097 \\ 0,264 \\ 0,952 \\ 0,609 \\ 0,112 \\ 0,251 \\ 0,944 \\ 0,643 \\ 0,066 \\ 0,289 \\ 0,967 \\ 0,612 \\ 0,107 \\ 0,249 \\ 0,947 \end{pmatrix}$$

$$\begin{aligned} (x_1, x_{13}) &= (0,849 * 0,619) + (0,016 * 0,097) + (0,321 * 0,264) \\ &+ (0,992 * 0,952) + (0,853 * 0,609) + (0,021 * 0,112) \\ &+ (0,299 * 0,251) + (0,990 * 0,944) + (0,855 * 0,643) \\ &+ (0,017 * 0,066) + (0,313 * 0,289) + (0,991 * 0,967) \\ &+ (0,850 * 0,612) + (0,024 * 0,107) + (0,286 * 0,249) \\ &+ (0,988 * 0,947) \end{aligned}$$

$$(x_1, x_{13}) = 6,216943$$

Berikut adalah perhitungan *kernel Polynomial* untuk data ke-2 dengan data ke-13.

$K(x_i, x)$  misal,  $x_i = x_2, x = x_{13}$ , maka  
 $K(x_2, x_{13}) = (x_2, x_{13})^2 = (5,645831)^2 = 31,87541155$

$$(x_2, x_{13}) = \begin{pmatrix} 0,615 \\ 0,018 \\ 0,381 \\ 0,991 \\ 0,607 \\ 0,033 \\ 0,346 \\ 0,983 \\ 0,609 \\ 0,031 \\ 0,350 \\ 0,985 \\ 0,602 \\ 0,039 \\ 0,334 \\ 0,980 \end{pmatrix} \cdot \begin{pmatrix} 0,619 \\ 0,097 \\ 0,264 \\ 0,952 \\ 0,609 \\ 0,112 \\ 0,251 \\ 0,944 \\ 0,643 \\ 0,066 \\ 0,289 \\ 0,967 \\ 0,612 \\ 0,107 \\ 0,249 \\ 0,947 \end{pmatrix}$$

$$\begin{aligned} (x_2, x_{13}) &= (0,615 * 0,619) + (0,018 * 0,097) + (0,381 * 0,264) \\ &\quad + (0,991 * 0,952) + (0,607 * 0,609) + (0,033 * 0,112) \\ &\quad + (0,346 * 0,251) + (0,983 * 0,944) + (0,609 * 0,643) \\ &\quad + (0,031 * 0,066) + (0,350 * 0,289) + (0,985 * 0,967) \\ &\quad + (0,602 * 0,612) + (0,039 * 0,107) + (0,334 * 0,249) \\ &\quad + (0,980 * 0,947) \end{aligned}$$

$$(x_2, x_{13}) = 5,645831$$

Perhitungan *kernel Polynomial* data uji dilakukan dengan seluruh data latih. Tahap selanjutnya adalah mencari nilai  $\alpha_i \cdot y_i \cdot K(x_i, x)$ .

- Perhitungan untuk data ke-1  
 $= 0,048112574 \times 1 \times 38,65039045 = 1,859569756$
- Perhitungan untuk data ke-2  
 $= 0,044069447 \times 1 \times 31,87541155 = 1,404731775$
- Perhitungan untuk data ke-3  
 $= 0 \times -1 \times 32,11516092 = 0$

Hasil perhitungan data uji 13 untuk mendapatkan nilai *kernel* dan  $\alpha_i \cdot y_i \cdot K(x_i, x)$  dapat dilihat pada Tabel 4.27.



Tabel 4.26 Hasil Perhitungan *Kernel* dan  $\alpha_i y_i K(x_i, x)$  data ke-13

Data ke-	$K(x_1, x_{13})$	$\alpha_i \cdot y_i \cdot K(x_1, x_{13})$
1	38,65039045	1,859569756
2	31,87541155	1,404731775
3	32,11516092	0
4	31,40079328	0
5	35,5888221	0
6	25,17810123	-0,013277166
7	23,45711851	-0,041845395
8	25,26570631	-0,013194881
9	33,47502108	0
10	27,47063961	0
11	25,29062447	-0,013909732
12	33,54889966	0
$\Sigma$		3,182074357

Selanjutnya adalah menghitung nilai  $f(x)$ . Hasil klasifikasi akan ditentukan dengan fungsi klasifikasi dengan rumus :

$$f^1(x) = \sum_{i=1}^m \alpha_i y_i K(x_1, x_{13}) + b$$

$$f^1(x) = 3,182074357 + (-3,303059184) = -0,120984827$$

$$\text{Fungsi Klasifikasi} = \text{sign}(-0,120984827) = -1$$

$$f^2(x) = 3,182074357 + (-1,491320596) = 1,690753761$$

$$\text{Fungsi Klasifikasi} = \text{sign}(1,690753761) = 1$$

Nilai  $b$  yang digunakan adalah nilai  $b$  pada level pertama dan level kedua. Karna fungsi klasifikasi bernilai -1 pada level pertama, maka dilanjutkan melakukan perhitungan fungsi  $f(x)$  sampai bernilai positif. Selanjutnya melakukan perhitungan untuk  $f(x)$  menggunakan nilai  $b$  pada level kedua, didapatkan nilai klasifikasi bernilai 1, maka hasil klasifikasi untuk data ke-13 adalah Tipe I. Tabel 4.28 menjelaskan tentang kelas sebenarnya dan hasil klasifikasi pada perhitungan data dan uji.

Tabel 4.27 Data Hasil Perhitungan  $f(x)$  dan Klasifikasi

Data	$f^1(x)$	$f^2(x)$	$f^3(x)$	$f^4(x)$	$f^5(x)$	Hasil Klasifikasi	Kelas Sebenarnya
13	-0,120984827	1,690753761	-	-	-	1 dilevel 1 = Tipe I	TipeI
14	1,193119869	-	-	-	-	1 dilevel 1 = Tipe I'	TipeI
15	-0,274890485	1,536848103	-	-	-	1 dilevel 1 = Tipe I	TipeI
16	-0,39903325	1,412705337	-	-	-	1 dilevel 1 = Tipe I	TipeI

#### 4.4 Perancangan Pengujian

##### 4.4.1 Perancangan Pengujian Pada Parameter *Lambda*

Pengujian ini dilakukan untuk melihat pengaruh nilai *lambda* pada *sequential training* terhadap nilai akurasi sistem. Pengujian ini dilakukan untuk mendapatkan nilai *lambda* yang dapat memberikan nilai akurasi yang paling baik. Pada Tabel 4.28 disajikan skenario pengujian pada nilai *lambda*.

Tabel 4.28 Skenario Pengujian Pada Nilai *Lambda*

<i>Lambda</i>	Akurasi					Rata - rata Akurasi (%)
	Percobaan ke-					
	1	2	3	4	5	
0,00001						
0,0001						
0,001						
0,01						
0,1						
0,5						
5						
10						
50						
100						

##### 4.4.2 Perancangan Pengujian Pada Parameter *Gamma*

Pengujian ini dilakukan untuk melihat pengaruh nilai *gamma* pada *sequential training* terhadap nilai akurasi sistem. Pengujian ini dilakukan untuk mendapatkan nilai *gamma* yang dapat memberikan nilai akurasi yang paling baik. Pada Tabel 4.29 disajikan skenario pengujian pada nilai *gamma*.

**Tabel 4.29 Skenario Pengujian Pada Nilai *Gamma***

<i>Gamma</i>	Akurasi					Rata - rata Akurasi (%)
	Percobaan ke-					
	1	2	3	4	5	
0,001						
0,01						
0,1						
1						
1,5						
2						
2,5						
3						
3,5						
4						

**4.4.3 Perancangan Pengujian Pada Nilai *Complexity (C)***

Pengujian ini dilakukan untuk melihat pengaruh nilai *C* pada *sequential training* terhadap nilai akurasi sistem. Pengujian ini dilakukan untuk mendapatkan nilai *C* yang dapat memberikan nilai akurasi yang paling baik. Pada Tabel 4.30 disajikan skenario pengujian pada nilai *C*.

**Tabel 4.30 Skenario Pengujian Pada Nilai *C***

Nilai <i>C</i>	Akurasi					Rata - rata Akurasi (%)
	Percobaan ke-					
	1	2	3	4	5	
1						
5						
10						
20						
30						
40						
50						
100						
150						
200						





#### 4.4.4 Perancangan Pengujian Pada Nilai Iterasi Maksimal

Pengujian ini dilakukan untuk melihat pengaruh nilai iterasi maksimal pada *sequential training* terhadap nilai akurasi sistem. Pengujian ini dilakukan untuk mendapatkan nilai iterasi maksimal yang dapat memberikan nilai akurasi yang paling baik. Pada Tabel 4.31 disajikan skenario pengujian pada nilai iterasi maksimal.

**Tabel 4.31 Skenario Pengujian pada Iterasi Maksimal**

Iterasi Maksimum	Akurasi					Rata - rata Akurasi (%)
	Percobaan ke-					
	1	2	3	4	5	
10						
20						
30						
40						
50						
100						
200						
300						
400						
500						

#### 4.4.5 Perancangan Pengujian Pada Nilai Jarak (*Distance*)

Pengujian ini dilakukan untuk melihat pengaruh nilai jarak (*distance*) pada GLCM terhadap nilai akurasi sistem. Pengujian ini dilakukan untuk mendapatkan nilai jarak (*distance*) yang dapat memberikan nilai akurasi yang paling baik. Pada Tabel 4.32 disajikan skenario pengujian pada nilai jarak.

**Tabel 4.32 Skenario Pengujian Pada Nilai Jarak (*Distance*)**

Jarak	Akurasi					Akurasi (%)
	Percobaan ke-					
	1	2	3	4	5	
1						
2						
3						
4						
5						

## BAB 5 IMPLEMENTASI

Pada bab ini berisi mengenai spesifikasi kebutuhan sistem, implementasi dari sistem dengan algoritme GLCM dan SVM, dan implementasi antarmuka pengguna.

### 5.1 Spesifikasi Sistem

Dalam implementasi sistem ini, spesifikasi kebutuhan perangkat dalam pembuatan sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *Gray Level Co-Occurrence Matrix* dan *Support Vector Machine* agar dapat berjalan dengan baik dibutuhkan suatu sistem yaitu sebagai berikut :

- a. Spesifikasi Kebutuhan Perangkat Keras (*Hardware*) yaitu:
  - Prosesor Intel(R) Core(TM) i5-6200U CPU @2.80GHz.
  - Memori 4.00 GB.
  - Hardisk dengan kapasitas 1 TB.
- b. Spesifikasi Kebutuhan Perangkat Lunak (*Software*) yaitu:
  - Sistem Operasi Microsoft Windows 10.
  - Bahasa Pemrograman Java.
  - Aplikasi IntelliJ IDEA.
  - Adobe Photoshop Express.

### 5.2 Batasan Implementasi

Berikut merupakan batasan dalam implementasi sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *Gray Level Co-Occurrence Matrix* dan *Support Vector Machine* :

- Implementasi untuk *pre-processing* citra yaitu *grayscale*.
- Pembacaan *dataset* untuk perhitungan pada program dilakukan dengan mencari lokasi folder dari dataset yang akan digunakan.
- Jumlah *dataset* yang digunakan yaitu sebanyak 60 data yang terbagi menjadi 6 kelas yaitu 10 data untuk kelas tipe I', 10 data untuk kelas tipe I, 10 data untuk kelas tipe II, 10 data untuk kelas tipe III, 10 data untuk kelas tipe IV, dan 10 data untuk kelas tipe V.
- Ukuran citra yang digunakan untuk data latih adalah 100x100.
- Ukuran citra yang digunakan untuk data uji adalah 400x200.
- Pengujian dilakukan dengan menggunakan data yang pembagiannya dilakukan secara acak untuk data uji dan data latih.
- Metode ekstraksi fitur tekstur yang digunakan yaitu *Gray Level Co-Occurrence Matrix*. Jumlah ekstraksi fitur yang digunakan ada 4 yaitu ASM, *contrast*, *correlation*, dan IDM.
- Metode klasifikasi yang digunakan yaitu *Support Vector Machine* dengan strategi *One Against All* dan menggunakan kernel *polynomial*.

- Input parameter dan hasil klasifikasi ditampilkan pada GUI.
- Nilai perhitungan sistem ditampilkan pada *console*.

### 5.3 Implementasi Algoritme

Pada sub bab Implementasi Algoritme akan dijelaskan mengenai implementasi kode program berdasarkan perancangan pada BAB 4. Pada sub bab ini terbagi menjadi beberapa sub bab yaitu implementasi *Gray Level Co-Occurence Matrix* dan Implementasi *Support Vector Machine*.

#### 5.3.1 Implementasi *Gray Level Co-Occurence Matrix*

##### 5.3.1.1 Implementasi Perhitungan *Grayscale*

Implementasi perhitungan *grayscale* merupakan proses untuk mendapatkan nilai keabuan pada setiap *pixel* pada data gambar yang dimasukkan. Pada tahap ini proses untuk mendapatkan nilai keabuan untuk setiap *pixel* yaitu dengan menjumlahkan nilai *red*, *green*, dan *blue* yang kemudian dibagi dengan 3. Potongan kode program dari algoritme *grayscale* ditunjukkan pada Kode Program 5.1.

```

1 public int[][] calcGreyLeveledMatrix(BufferedImage image) {
2     int[][] matriks_greyscale = new
3     int[image.getHeight()][image.getWidth()];
4     for (int i = 0; i < image.getHeight(); i++) {
5         for (int j = 0; j < image.getWidth(); j++) {
6             Color rgb = new Color(image.getRGB(j,i));
7             matriks_greyscale [i][j]=(int) Math.ceil((double) (rgb.getRed()
8 + rgb.getGreen() + rgb.getBlue()) / 3);
9         }
10    }
11    return matriks_greyscale;
12 }

```

##### Kode Program 5.1 Perhitungan *Grayscale*

Berikut penjelasan kode program pada Kode Program 5.1 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2 merupakan inialisasi terhadap variabel yang digunakan di dalam *method calcGreyLeveledMatrix*.
- Baris 3-6 merupakan proses perhitungan matriks *grayscale*.
- Baris 9 merupakan proses mengembalikan nilai *matriks\_greyscale*.

##### 5.3.1.2 Implementasi Perhitungan *Min-Max Normalization*

Implementasi perhitungan *min-max normalization* merupakan proses untuk mengubah rentang nilai intensitas *pixel* keabuan yang dinormalisasi sesuai dengan *range* minimal dan maksimal baru yang sudah ditentukan. Proses perhitungan ini dapat berjalan apabila telah mendapatkan hasil perhitungan dari *method calcGreyLeveledMatrix()*. Potongan kode program dari algoritme *min-max normalization* ditunjukkan pada Kode Program 5.2.

```

1 public int[][] calcMinMaxNormalizationMatrix(int[][]
  greyLeveledMatrix) {
2     int[][] matriks_minmaxnor = Arrays.stream(greyLeveledMatrix)
3         .map(i -> Arrays.stream(i)
4             .map(j -> (int) Math.round((j - MIN) * (double) (NEW_MAX -
  NEW_MIN) / (MAX - MIN) + NEW_MIN))
5             .toArray())
6         .toArray(int[][]::new);
7     return matriks_minmaxnor;
  }

```

### Kode Program 5.2 Perhitungan *Min-Max Normalization*

Berikut penjelasan kode program pada Kode Program 5.2 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2 merupakan inisialisasi terhadap variabel yang digunakan di dalam *method calcMinMaxNormalizationMatrix*.
- Baris 3-6 proses perulangan untuk data ke-i dan data ke-j terhadap matriks *grayscale* kemudian melakukan proses perhitungan *min-max normalization*.
- Baris 7 merupakan proses mengembalikan nilai *matriks\_minmaxnor*.

#### 5.3.1.3 Implementasi Perhitungan *Matrix Co-Occurrence*

Implementasi perhitungan *matrix co-ocurance* merupakan proses untuk mendapatkan jumlah nilai *pixel* ketetanggaan dari sudut 0°, 45°, 90°, dan 135°. Pada tahap ini proses mendapatkan nilai ketetanggaan yaitu setiap koordinat *pixel* (i,j) dihitung jumlah total *pixel* berpasangannya sesuai dengan jarak *pixel* asal dan *pixel* tetangganya yang sudah ditentukan. Proses perhitungan ini dapat berjalan apabila telah mendapatkan hasil perhitungan dari *method calcRangeIntensityMatrix()*. Potongan kode program dari algoritme *matrix co-occurrence* ditunjukkan pada Kode Program 5.3.

```

1 public int[][] calcCoOccurrenceMatrix(int[][] matriks_minmaxnor,
  DegreesEnum angle, int distance) {
2     int[][] matriks_berpasangan = new int[NEW_MAX + 1][NEW_MAX + 1];
3
4     boolean valid = angle != null &&
5         (DEGREES_0 == angle || DEGREES_45 == angle ||
6          DEGREES_90 == angle || DEGREES_135 == angle);
7     if (!valid) {
8         return matriks_berpasangan;
9     }
10    int startRow = this.startRow(angle, distance);
11    int startColumn = this.startColumn(angle, distance);
12    int endColumn = this.endColumn(angle, matriks_minmaxnor,
13    distance);
14    Arrays.stream(matriks_berpasangan).forEach(i -> Arrays.setAll(i,
15    index -> 0));
16
17    for (int i = startRow; i < matriks_minmaxnor.length; i++) {
18        for (int j = startColumn; j < endColumn; j++) {
19            switch (angle) {
20                case DEGREES_0:
21                    matriks_berpasangan[matriks_minmaxnor[i][j]]
22                    [matriks_minmaxnor[i][j + 1]]++;
23                    break;

```

```

20     case DEGREES_45:
21         matriks_berpasangan[matriks_minmaxnor[i][j]]
           [matriks_minmaxnor[i - 1][j + 1]]++;
22         break;
23     case DEGREES_90:
24         matriks_berpasangan[matriks_minmaxnor[i][j]]
           [matriks_minmaxnor[i - 1][j]]++;
           break;
25     case DEGREES_135:
26         matriks_berpasangan[matriks_minmaxnor[i][j]]
           [matriks_minmaxnor[i - 1][j - 1]]++;
27         break;
28     }
29 }
30 }
31 return matriks_berpasangan;}

```

**Kode Program 5.3 Perhitungan Matrix Co-Occurance**

Berikut penjelasan kode program pada Kode Program 5.3 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2-14 merupakan inialisasi terhadap variabel-variabel yang digunakan di dalam *method calcCoOccurrenceMatrix*.
- Baris 14-28 merupakan proses perhitungan untuk mendapatkan nilai matriks *coocurrence* dengan arah/sudut 0°, 45°, 90°, dan 135°.
- Baris 32 merupakan proses mengembalikan nilai *matriks\_berpasangan*.

#### 5.3.1.4 Implementasi Perhitungan Fitur ASM

Implementasi perhitungan fitur ASM merupakan proses untuk mendapatkan nilai konsentrasi pasangan dengan derajat keabuan dari suatu citra. Nilai fitur ASM ini akan disimpan dan digunakan sebagai data *training* untuk proses klasifikasi. Potongan kode program dari algoritme fitur ASM ditunjukkan pada Kode Program 5.4.

```

1 public double calcAngularSecondMoment(double[][] matrix) {
2     Double asm = IntStream.range(0, matrix.length)
3         .mapToObj(i -> IntStream.range(0, matrix[i].length)
4             .mapToDouble(j -> matrix[i][j] * (1 / (1 + Math.pow(i -
5 j, 2))))
6             .reduce(0d, Double::sum))
7         .reduce(0d, Double::sum);
8     return asm;}

```

**Kode Program 5.4 Perhitungan Fitur ASM**

Berikut penjelasan kode program pada Kode Program 5.4 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2 merupakan inialisasi terhadap variabel yang digunakan di dalam *method calcAngularSecondMoment*.
- Baris 3-6 merupakan proses perhitungan untuk mencari nilai ASM.
- Baris 7 merupakan proses mengembalikan nilai ASM.



### 5.3.1.5 Implementasi Perhitungan Fitur *Contrast*

Implementasi perhitungan fitur *contrast* merupakan proses untuk mendapatkan nilai jumlah keberagaman derajat keabuan dari suatu citra. Nilai fitur *contrast* ini akan disimpan dan digunakan sebagai data *training* untuk proses klasifikasi. Potongan kode program dari algoritme fitur *contrast* ditunjukkan pada Kode Program 5.5.

```

1 public double calcContrast(double[][] matrix) {
2     Double contrast = IntStream.range(0, matrix.length)
3         .mapToObj(i -> IntStream.range(0, matrix[i].length)
4             .mapToDouble(j -> Math.pow(Math.abs(i - j), 2) *
5                 matrix[i][j])
6                 .reduce(0d, Double::sum))
7         .reduce(0d, Double::sum);
8     return contrast;
9 }

```

#### Kode Program 5.5 Perhitungan Fitur *Contrast*

Berikut penjelasan kode program pada Kode Program 5.5 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2 merupakan inialisasi terhadap variabel yang digunakan di dalam *method calcContrast*.
- Baris 3-6 merupakan proses perhitungan untuk mencari nilai *contrast*.
- Baris 7 merupakan proses mengembalikan nilai *contrast*.

### 5.3.1.6 Implementasi Perhitungan Fitur *Correlation*

Implementasi perhitungan fitur *correlation* merupakan proses untuk mendapatkan nilai jumlah ketergantungan linear derajat keabuan dari suatu citra. Nilai fitur *correlation* ini akan disimpan dan digunakan sebagai data *training* untuk proses klasifikasi. Potongan kode program dari algoritme fitur *correlation* ditunjukkan pada Kode Program 5.6.

```

1 public double calcCorrelation(double[][] matrix) {
2     double mui = helperService.calcMUX(matrix);
3     double muj = helperService.calcMUY(matrix);
4     double sdi = helperService.calcStandardDeviationX(matrix);
5     double sdj = helperService.calcStandardDeviationY(matrix);
6     double corelation = IntStream.range(0, matrix.length)
7         .mapToObj(i -> IntStream.range(0, matrix[i].length)
8             .mapToDouble(j -> ((i - mui) * (j - muj) * matrix[i][j])
9                 / (sdi * sdj))
10                .sum())
11         .mapToDouble(Double::doubleValue)
12         .sum();
13     return corelation;

```

#### Kode Program 5.6 Perhitungan Fitur *Correlation*

Berikut penjelasan kode program pada Kode Program 5.6 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.

- Baris 2-6 merupakan inialisasi terhadap variabel yang digunakan di dalam *method calcCorrelation*.
- Baris 7-12 merupakan proses perhitungan untuk mencari nilai *correlation*.
- Baris 7 merupakan proses mengembalikan nilai *correlation*.

### 5.3.1.7 Implementasi Perhitungan Fitur IDM (*Inverse Different Moments*)

Implementasi perhitungan fitur IDM (*Inverse Different Moments*) merupakan proses untuk mendapatkan nilai kehomogenan derajat keabuan dari suatu citra. Nilai fitur IDM ini akan disimpan dan digunakan sebagai data *training* untuk proses klasifikasi. Potongan kode program dari algoritme fitur IDM ditunjukkan pada Kode Program 5.7.

```

1 public double calcInverseDifference(double[][] matrix) {
2     Double idm = IntStream.range(0, matrix.length)
3         .mapToObj(i -> IntStream.range(0, matrix[i].length)
4             .mapToDouble(j -> matrix[i][j] * (1 / (1 + Math.pow(i -
5                 j, 2))))
6             .reduce(0d, Double::sum))
7         .reduce(0d, Double::sum);
8     return idm;
9 }

```

#### Kode Program 5.7 Perhitungan Fitur IDM

Berikut penjelasan kode program pada Kode Program 5.7 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2 merupakan inialisasi terhadap variabel yang digunakan di dalam *method calcInverseDifference*.
- Baris 3-6 merupakan proses perhitungan untuk mencari nilai *Inverse Difference*.
- Baris 7 merupakan proses mengembalikan nilai IDM.

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

#### 5.3.2.1 Implementasi Perhitungan Kernel SVM

Implementasi untuk perhitungan kernel SVM merupakan langkah pertama dari algoritme SVM yang dilakukan setelah mendapatkan nilai-nilai fitur dari citra bibir. Dalam perhitungan kernel SVM ini menggunakan kernel *polynomial*. Perhitungan ini dilakukan untuk mendapatkan nilai matriks kernel *polynomial* yang selanjutnya akan digunakan untuk mencari nilai matriks *hessian*. Potongan kode program dari algoritme kernel *polynomial* ditunjukkan pada Kode Program 5.8.

```

1 public BigDecimal[][] calcPolynomialKernel(List<DataModel>
2     dataList) {
3     double[][] matrixFeature =
4         helperService.convertDataModelToMatrix(dataList);
5     BigDecimal[][] kernel_polynomial = IntStream.range(0,
6         matrixFeature.length)
7         .mapToObj(i -> IntStream.range(0,
8             matrixFeature.length)

```

```

5      .mapToObj(j -> IntStream.range(0,
6          matrixFeature[j].length)
7          .mapToObj(k ->
8              BigDecimal.valueOf(matrixFeature[i][k])
9                  .multiply(BigDecimal.valueOf(
10                     matrixFeature[j][k])))
11             .reduce(BigDecimal.ZERO, BigDecimal::add))
12     .map(result -> result.pow(POLYNOMIAL_OF_DEGREE))
13     .toArray(BigDecimal[]::new)
14     .toArray(BigDecimal[][]::new);
return kernel_polinomial;
}

```

**Kode Program 5.8 Implementasi Perhitungan Kernel SVM**

Berikut penjelasan kode program pada Kode Program 5.8 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2-3 merupakan inialisasi terhadap variabel yang digunakan di dalam *method calcPolynomialKernel*.
- Baris 4-10 merupakan proses perhitungan untuk mencari nilai kernel *polynomial*.
- Baris 11-12 merupakan proses menyimpan nilai *array* yang dihasilkan dari perhitungan kernel *polynomial*.
- Baris 13 merupakan proses mengembalikan nilai *kernel\_polinomial*.

### 5.3.2.2 Implementasi Perhitungan Matriks *Hessian*

Implementasi untuk perhitungan matriks *Hessian* merupakan proses perhitungan yang dilakukan setelah mendapatkan hasil nilai kernel *polynomial*. Proses untuk mendapatkan nilai matriks *hessian* yaitu dengan melakukan perkalian antara nilai label kelas data *training* dengan nilai kernel *polynomial*, setelah itu dijumlahkan dengan nilai *lambda* kuadrat. Potongan kode program dari algoritme matriks *hessian* ditunjukkan pada Kode Program 5.9.

```

1 public BigDecimal[][] calcMatrixHessian(BigDecimal[][]
2     kernel_polinomial, int[] classification, double λ) {
3     BigDecimal[][] matriks_hessian = IntStream.range(0,
4         kernel_polinomial.length)
5         .mapToObj(i -> IntStream.range(0,
6             kernel_polinomial[i].length)
7             .mapToObj(j -> kernel_polinomial[i][j]
8                 .add(BigDecimal.valueOf(Math.pow(λ, 2)))
9                 .multiply(BigDecimal.valueOf(classification[i] *
10                    classification[j])))
11             .toArray(BigDecimal[]::new))
12     .toArray(BigDecimal[][]::new);
13     return matriks_hessian; }

```

**Kode Program 5.9 Perhitungan Matriks *Hessian***

Berikut penjelasan kode program pada Kode Program 5.9 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2 merupakan inialisasi terhadap variabel yang digunakan di dalam *method calcMatrixHessian*.





- Baris 3-6 merupakan proses perhitungan untuk mencari nilai matriks *hessian*.
- Baris 7-8 merupakan proses menyimpan nilai *array* yang dihasilkan dari perhitungan matriks *hessian*.
- Baris 9 merupakan proses mengembalikan nilai *matriks\_hessian*.

### 5.3.2.3 Implementasi Perhitungan Nilai $E_i$

Implementasi untuk perhitungan nilai  $E_i$  merupakan proses perhitungan yang dilakukan setelah mendapatkan hasil nilai matriks *hessian*. Proses untuk mendapatkan nilai  $E_i$  yaitu dengan melakukan perkalian antara nilai alfa dengan nilai matriks *hessian*, kemudian hasil dari perkalian tersebut dijumlahkan seluruhnya untuk setiap baris data. Potongan kode program dari algoritme  $E_i$  ditunjukkan pada Kode Program 5.10.

```

1 public BigDecimal[] calcMatrixEi(BigDecimal[][] matriks_hessian,
2   List<AlphaClassification>  $\alpha$ ) {
3     BigDecimal[] nilai_Ei = IntStream.range(0,
4       matriks_hessian.length)
5       .mapToObj(i -> IntStream.range(0, matriks_hessian[i].length)
6         .mapToObj(j ->
7           BigDecimal.valueOf( $\alpha$ .get(j).getAlpha()).multiply(matriks_hessian[i]
8             [j]))
9           .reduce(BigDecimal.ZERO, BigDecimal::add))
10      .toArray(BigDecimal[]::new);
11     return nilai_Ei;
12 }

```

#### Kode Program 5.10 Perhitungan Nilai $E_i$

Berikut penjelasan kode program pada Kode Program 5.10 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2 merupakan inialisasi terhadap variabel yang digunakan di dalam *method calcMatrixEi*.
- Baris 3-5 merupakan proses perhitungan untuk mencari nilai  $E_i$ .
- Baris 6 merupakan proses menyimpan nilai *array* yang dihasilkan dari perhitungan matriks *hessian*.
- Baris 7 merupakan proses mengembalikan nilai *nilai\_Ei*.

### 5.3.2.4 Implementasi Perhitungan Nilai $\delta\alpha_i$

Implementasi untuk perhitungan nilai  $\delta\alpha_i$  merupakan proses perhitungan yang dilakukan setelah mendapatkan hasil nilai  $E_i$ . Proses untuk mendapatkan nilai  $\delta\alpha_i$  yaitu mencari nilai maksimum dari 1 dikurang nilai  $E_i$  kemudian dikalikan dengan nilai *gamma*, lalu dibandingkan dengan nilai  $-\alpha$ . Setelah didapat nilai maksimum selanjutnya mencari nilai minimum dari nilai maksimum dibandingkan dengan nilai  $C$  dikurangi dengan nilai alfa. Potongan kode program dari algoritme  $\delta\alpha_i$  ditunjukkan pada Kode Program 5.11.

```

1 public double[] calcDeltaAlpha(BigDecimal[] Ei, double γ, double
  c, List<AlphaClassification> α) {
2     return IntStream.range(0, Ei.length)
3         .mapToDouble(i -> {
4             double x = γ * (1 - Ei[i].doubleValue());
5             double y = -1 * α.get(i).getAlpha();
6             double z = c - α.get(i).getAlpha();
7             double max = x > y ? x : y;
8             return max < z ? max : z; })
9         .toArray();
    }

```

**Kode Program 5.11 Perhitungan Nilai  $\delta\alpha_i$**

Berikut penjelasan kode program pada Kode Program 5.11 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2-8 merupakan proses perhitungan untuk mencari nilai  $\delta\alpha_i$ .
- Baris 9 merupakan proses menyimpan nilai *array* yang dihasilkan dari perhitungan nilai  $\delta\alpha_i$ .

### 5.3.2.5 Implementasi Perhitungan Nilai $\alpha_i$

Implementasi untuk perhitungan nilai  $\alpha_i$  merupakan proses perhitungan yang dilakukan setelah mendapatkan hasil nilai  $\delta\alpha_i$ . Proses untuk mendapatkan nilai  $\alpha_i$  yaitu dengan menjumlahkan nilai alfa baru dengan nilai delta alfa. Potongan kode program dari algoritme  $\alpha_i$  ditunjukkan pada Kode Program 5.12.

```

1 public List<AlphaClassification>
  calcAlpha(List<AlphaClassification> α, double[] δαi, int level,
  int iterate) {
2     List<AlphaClassification> alpha = IntStream.range(0, α.size())
3         .mapToObj(i -> new AlphaClassification(level, iterate, i,
  α.get(i).getClassification(), α.get(i).getAlpha() + δαi[i]))
4         .collect(Collectors.toList());
5     return alpha;
}

```

**Kode Program 5.12 Perhitungan Nilai  $\alpha_i$**

Berikut penjelasan kode program pada Kode Program 5.12 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2 merupakan inisialisasi terhadap variabel yang digunakan di dalam *method calcAlpha*.
- Baris 3-4 merupakan proses perhitungan untuk mencari nilai  $\alpha_i$ .
- Baris 5 merupakan proses mengembalikan nilai *nilai\_Ei*.

### 5.3.2.6 Implementasi Perhitungan Nilai $w, x^+$ dan $w, x^-$

Implementasi untuk perhitungan nilai  $w, x^+$  dan  $w, x^-$  merupakan proses perhitungan untuk mencari bobot dari kelas positif dan kelas negatif. Proses untuk mendapatkan nilai bobot untuk kelas positif dan kelas negatif diawali dengan mencari nilai kernel pada kelas positif dan kelas negatif, keduanya ditentukan dari data yang memiliki nilai maksimal alfa iterasi paling terakhir, lalu mengambil nilai kernel pada data tersebut. Setelah didapatkan nilai kernel pada kelas positif dan negatif, kemudian menghitung nilai  $w, x^+$  yaitu dengan



mengalikan nilai kernel pada kelas positif dengan nilai alfa iterasi terakhir yang dikali dengan nilai kelas data, sedangkan nilai  $w, x^-$  didapat dengan mengalikan nilai kernel pada kelas negatif dengan nilai alfa iterasi terakhir yang dikali dengan nilai kelas data. Potongan kode program dari algoritme  $w, x^+$  dan  $w, x^-$  ditunjukkan pada Kode Program 5.13.

```

1 public WeightModel calcWeight(BigDecimal[][] matriks_polinomial,
2   List<AlphaClassification> α) {
3     AlphaClassification alphaPositive = this.calcMaxAlfa(α,
4       POSITIVE);
5     AlphaClassification alphaNegative = this.calcMaxAlfa(α,
6       NEGATIVE);
7
8     List<BigDecimal> polynominalPositive = helperService
9       .getColumnMatrixByIndex(polynomialMatrix,
10        alphaPositive.getIndex());
11    List<BigDecimal> polynominalNegative = helperService
12      .getColumnMatrixByIndex(polynomialMatrix,
13        alphaNegative.getIndex());
14
15    BigDecimal weightPositive = α.stream()
16      .map(c -> BigDecimal.valueOf(c.getClassification()))
17      .multiply(polynominalPositive.get(c.getIndex()))
18      .multiply(BigDecimal.valueOf(c.getAlpha()))
19      .reduce(BigDecimal.ZERO, BigDecimal::add);
20
21    BigDecimal weightNegative = α.stream()
22      .map(c -> BigDecimal.valueOf(c.getClassification()))
23      .multiply(polynominalNegative.get(c.getIndex()))
24      .multiply(BigDecimal.valueOf(c.getAlpha()))
25      .reduce(BigDecimal.ZERO, BigDecimal::add);
26
27    return new WeightModel(weightPositive, weightNegative);}

```

#### Kode Program 5.13 Perhitungan Nilai $w, x^+$ dan $w, x^-$

Berikut penjelasan sourcecode pada Kode Program 5.13 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 3-7 merupakan inisialisasi terhadap variabel-variabel yang digunakan di dalam *method calcWeight*.
- Baris 9-13 merupakan proses perhitungan untuk mencari nilai  $w, x^+$ .
- Baris 15-19 merupakan proses perhitungan untuk mencari nilai  $w, x^-$ .
- Baris 21 merupakan proses mengembalikan nilai *weight*.

#### 5.3.2.7 Implementasi Perhitungan Nilai $b$

Implementasi untuk perhitungan nilai  $b$  merupakan proses perhitungan yang dilakukan setelah mendapatkan hasil nilai  $w, x^+$  dan  $w, x^-$ . Proses untuk mendapatkan nilai  $b$  yaitu dengan menjumlahkan nilai  $w, x^+$  dengan nilai  $w, x^-$  dan setelah itu dikali dengan -0,5. Potongan kode program dari algoritme  $b$  ditunjukkan pada Kode Program 5.14.

```

1 public BigDecimal calcBiasValue(WeightModel weight) {
2     return (weight.getPositive().add(weight.getNegative()))
3         .multiply(BigDecimal.valueOf(-0.5));}

```

#### Kode Program 5.14 Perhitungan Nilai $b$

Berikut penjelasan kode program pada Kode Program 5.14 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 2-3 merupakan proses perhitungan nilai  $b$ .

#### 5.3.2.8 Implementasi Perhitungan Nilai $f(x)$

Implementasi untuk perhitungan nilai  $f(x)$  merupakan proses perhitungan yang dilakukan setelah mendapatkan hasil nilai  $b$ . Proses untuk mendapatkan nilai  $f(x)$  yaitu dengan menjumlahkan nilai bias dengan nilai bobot untuk data *testing*. Potongan kode program dari algoritme  $b$  ditunjukkan pada Kode Program 5.15.

```

1 public List<ClassificationDataModel>
2   calcFxCClassification(List<BigDecimal> bias,
3     List<WeightUndefineClass> weightUndifineClasses) {
4     List<ClassificationDataModel> FxCClassification = new
5     ArrayList<>();
6
7     for (WeightUndefineClass weight : weightUndifineClasses) {
8         List<BigDecimal> values = new ArrayList<>();
9         int index = 0;
10        BigDecimal value;
11        do {
12            value = weight.getValue().add(bias.get(index));
13            values.add(value);
14            index++;
15        } while (value.signum() < 0 && index < MAX_LEVEL); //
16
17        int ordinal = values.size() > MAX_LEVEL ? (MAX_LEVEL + 1) :
18        values.size() >= MAX_LEVEL ? values.size() + 1 : values.size(); }
19    return FxCClassification;

```

#### Kode Program 5.15 Perhitungan Nilai $f(x)$

Berikut penjelasan kode program pada Kode Program 5.15 :

- Baris 1 merupakan deklarasi *method* beserta parameter yang dibutuhkan.
- Baris 3 merupakan inisialisasi terhadap variabel yang digunakan di dalam *method calcFxCClassification*.
- Baris 5-15 merupakan proses perhitungan nilai  $f(x)$ .
- Baris 15 merupakan proses mengembalikan nilai *FxCClassification*.

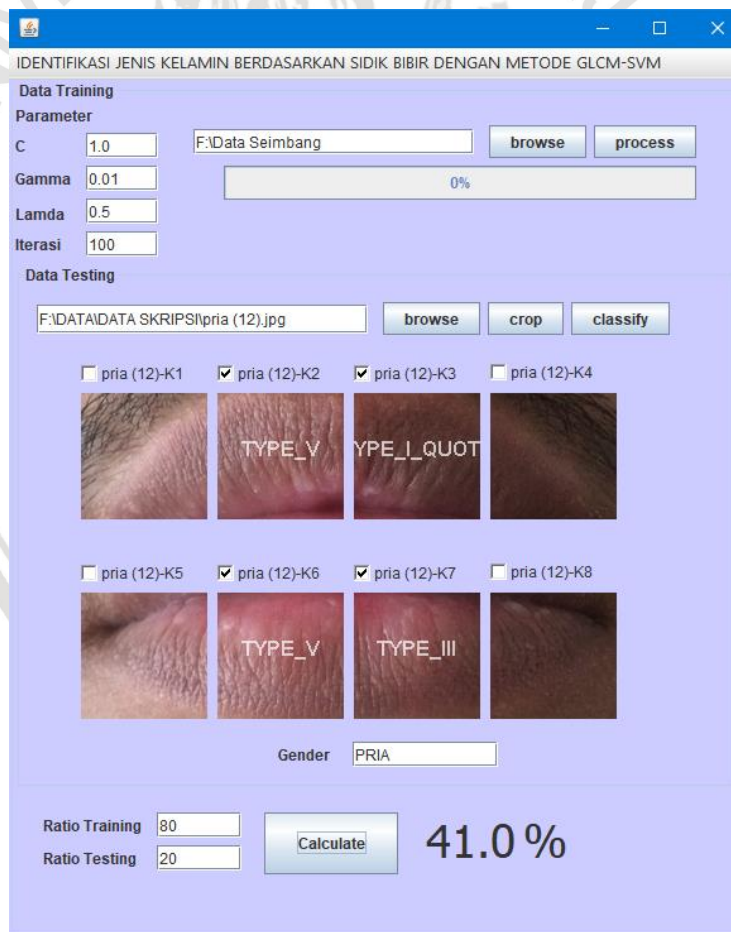
### 5.4 Hasil Implementasi

Pada implementasi sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine* pengguna akan ditampilkan sebuah *form dialog* seperti pada Gambar 5.1 untuk dapat memasukkan *input* parameter SVM dan memilih folder yang berisi *dataset* citra. Folder *dataset* ini berisi citra data latih dengan ukuran 100x100 *pixel* yang untuk label kelasnya terdapat pada nama *file* gambar. Setelah

memasukkan parameter dan memilih folder mana yang akan digunakan sebagai *dataset training*, pengguna lalu melakukan klik *button process*. Kemudian sistem mulai melakukan proses *training* terhadap *dataset* yang dipilih. Hasil perhitungan akan ditampilkan pada *console* seperti pada Gambar 5.2.

Setelah selesai melakukan proses *training*, pengguna dapat mulai melakukan klasifikasi terhadap citra bibir utuh dengan ukuran citra 400x200 *pixel*. Dimulai dengan melakukan klik *button browse* untuk mengambil masukan citra bibir utuh yang akan di klasifikasikan. Kemudian klik *button crop* untuk melakukan pemotongan citra utuh agar terbagi pada beberapa kuadran. Kemudian kuadran yang dipilih hanya yang berada ditengah saja, karena untuk kuadran yang berada diujung bibir kurang terlihat guratan alur dan pola sidik bibir. Setelah melakukan pemilihan dengan cara *checkboxlist* pada kuadran yang ditengah saja, pengguna dapat memulai klasifikasi dengan klik *button classify*. Hasil perhitungan terhadap data uji bibir utuh juga akan ditampilkan pada *console* seperti Gambar 5.3.

Kemudian melakukan pengujian dengan memasukkan nilai *ratio training* dan *ratio testing* yang sudah ditentukan, kemudian pengguna klik *button calculate* untuk dapat mulai melakukan proses perhitungan nilai akurasi. Hasil proses pengujian akurasi ditampilkan pada *console* seperti Gambar 5.4.



Gambar 5.1 Tampilan Antarmuka Hasil Implementasi

```

Run: Main x
Kernel Polinomial Training Level 1
68.31241218166815 , 20.553194150699294 , 84.34135539953205 , 51.087526880626534 , 273.2324376829171 , 51.612793479783775
20.553194150699294 , 11.803989980522651 , 23.32621759421501 , 17.26533534343034 , 50.69750135473965 , 16.881832151382827
84.34135539953205 , 23.32621759421501 , 118.13251203892996 , 65.4557139370377 , 375.65929427476027 , 65.29181803992957 ,
51.087526880626534 , 17.26533534343034 , 65.4557139370377 , 39.748999918680866 , 196.25657250906923 , 39.70427652255897
273.2324376829171 , 50.69750135473965 , 375.65929427476027 , 196.25657250906923 , 1435.3635716094473 , 203.7262149858704
51.612793479783775 , 16.881832151382827 , 65.29181803992957 , 39.70427652255897 , 203.7262149858704 , 40.508097273539754
401.69863106791496 , 67.88125067446666 , 592.8087153378424 , 293.70573725021666 , 2271.4578324386453 , 305.8981964456363
108.22689890702313 , 26.29862443334346 , 151.28793605479206 , 82.00259754273944 , 515.6254459461042 , 83.4411101711744 ,
119.97637702908352 , 28.4514133953157 , 159.3395396841084 , 88.48126696556989 , 559.806303955848 , 90.33352887353755 , 8
31.065805376166768 , 13.90841864091939 , 37.09232403349241 , 24.99907724023104 , 96.67918370950733 , 24.868384214089286
133.08946749621396 , 30.64306475236789 , 176.81845334612143 , 97.673523957333 , 630.1295932784191 , 100.10765577845692 ,
24.906287980097876 , 12.702706833942287 , 30.224514997884047 , 20.81093923738431 , 70.66049489898967 , 20.36784450237374

Matriks Hessian Training Level 1
68.56241218166815 , 20.803194150699294 , -84.59135539953205 , -51.337526880626534 , -273.4824376829171 , -51.86279347978
20.803194150699294 , 12.053989980522651 , -23.57621759421501 , -17.51533534343034 , -50.94750135473965 , -17.13183215138
-84.59135539953205 , -23.57621759421501 , 118.38251203892996 , 65.7057139370377 , 375.90929427476027 , 65.54181803992957
-51.337526880626534 , -17.51533534343034 , 65.7057139370377 , 39.998999918680866 , 196.50657250906923 , 39.9542765225589
-273.4824376829171 , -50.94750135473965 , 375.90929427476027 , 196.50657250906923 , 1435.6135716094473 , 203.97621498587
-51.862793479783775 , -17.131832151382827 , 65.54181803992957 , 39.95427652255897 , 203.9762149858704 , 40.7580972735397
-401.94863106791496 , -68.13125067446666 , 593.0587153378424 , 293.95573725021666 , 2271.7078324386453 , 306.14819644563
-108.47689890702313 , -26.54862443334346 , 151.53793605479206 , 82.25259754273944 , 515.8754459461042 , 83.6911101711744
-120.22637702908352 , -28.7014133953157 , 159.5895396841084 , 88.73126696556989 , 560.056303955848 , 90.58352887353755 ,
-31.315805376166768 , -14.15841864091939 , 37.34232403349241 , 25.24907724023104 , 96.92918370950733 , 25.11838421408928
-133.33946749621396 , -30.89306475236789 , 177.06845334612143 , 97.923523957333 , 630.3795932784191 , 100.35765577845692

```

Gambar 5.2 Nilai Proses Training

```

Run: Main x
ClassificationDataModel{name='pria (12)-K7', type=TYPE_III, functionValues=[-4032.4279295565185595
ClassificationDataModel{name='pria (12)-K6', type=TYPE_V, functionValues=[-17457.17018265315206048
ClassificationDataModel{name='pria (12)-K2', type=TYPE_V, functionValues=[-42231.53130430320546862
ClassificationDataModel{name='pria (12)-K3', type=TYPE_I_QUOTE, functionValues=[32.016910004804716

```

Gambar 5.3 Nilai Hasil Klasifikasi

```

Run: Main x
1.DATA TESTING = 12
2.DATA HASIL KLASIFIKASI[
DataModel{nama data = 'TIII-B5K2L', kelas sebenarnya = TYPE_III, hasil klasifikasi = TYPE_II},
DataModel{nama data = 'TV-B2K3L', kelas sebenarnya = TYPE_V, hasil klasifikasi = TYPE_I_QUOTE},
DataModel{nama data = 'TIII-B7K2L', kelas sebenarnya = TYPE_III, hasil klasifikasi = TYPE_V},
DataModel{nama data = 'TII-B2K7W', kelas sebenarnya = TYPE_II, hasil klasifikasi = TYPE_II},
DataModel{nama data = 'TII-B6K7L', kelas sebenarnya = TYPE_I_QUOTE, hasil klasifikasi = TYPE_I_QUOTE},
DataModel{nama data = 'TI-B1K7W', kelas sebenarnya = TYPE_I, hasil klasifikasi = TYPE_I_QUOTE},
DataModel{nama data = 'TV-B3K3L', kelas sebenarnya = TYPE_V, hasil klasifikasi = TYPE_I_QUOTE},
DataModel{nama data = 'TIV-B8K6L', kelas sebenarnya = TYPE_IV, hasil klasifikasi = TYPE_V},
DataModel{nama data = 'TIII-B18K2W', kelas sebenarnya = TYPE_III, hasil klasifikasi = TYPE_III},
DataModel{nama data = 'TIII-B24K6W', kelas sebenarnya = TYPE_III, hasil klasifikasi = TYPE_III},
DataModel{nama data = 'TI-B1K7L', kelas sebenarnya = TYPE_I_QUOTE, hasil klasifikasi = TYPE_I_QUOTE},
DataModel{nama data = 'TII-B5K7W', kelas sebenarnya = TYPE_II, hasil klasifikasi = TYPE_II}]
3.DATA UJI YANG SESUAI DENGAN HASIL KLASIFIKASI

```

Gambar 5.4 Hasil Klasifikasi Proses Pengujian Akurasi



## BAB 6 PENGUJIAN

Pada bab ini berisi mengenai pengujian yang akan dilakukan pada sistem serta analisis dari percobaan dengan memasukkan variasi nilai jarak dan variasi nilai-nilai parameter dari metode SVM.

### 6.1 Pengujian Parameter $\lambda$ (*Lambda*)

Pada pengujian parameter nilai *lambda* bertujuan untuk mengetahui pada nilai *lambda* berapa yang memberikan nilai akurasi terbaik untuk sistem dan bagaimana hasil analisis pada pengujian *lambda*.

#### 6.1.1 Skenario Pengujian Nilai *Lambda*

Pada pengujian *lambda* akan dilakukan percobaan sebanyak 5 kali dengan beberapa variasi nilai *lambda* yang berbeda. Nilai yang dipakai pada pengujian *lambda* ada 10, dimulai dari 0,00001 sampai 100. Untuk pengujian ini juga menggunakan parameter  $C = 1$ ,  $\gamma = 0,01$  dan  $\text{itermax} = 10$ . Kernel yang digunakan pada pengujian ini adalah kernel *polynomial*. Data yang digunakan pada pengujian *lambda* sebanyak 60 data. Data ini terbagi atas 10 data kelas Tipe I', 10 data kelas Tipe I, 10 data kelas Tipe II, 10 data kelas Tipe III, 10 data kelas Tipe IV, dan 10 data kelas Tipe V. Data kemudian dibagi menjadi data latih dan data uji secara acak. Perbandingan data latih dan data uji pada pengujian ini adalah 80%:20% yaitu meliputi 48 data training dan 12 data uji. Hasil pengujian terhadap parameter *lambda* ditunjukkan pada Tabel 6.1.

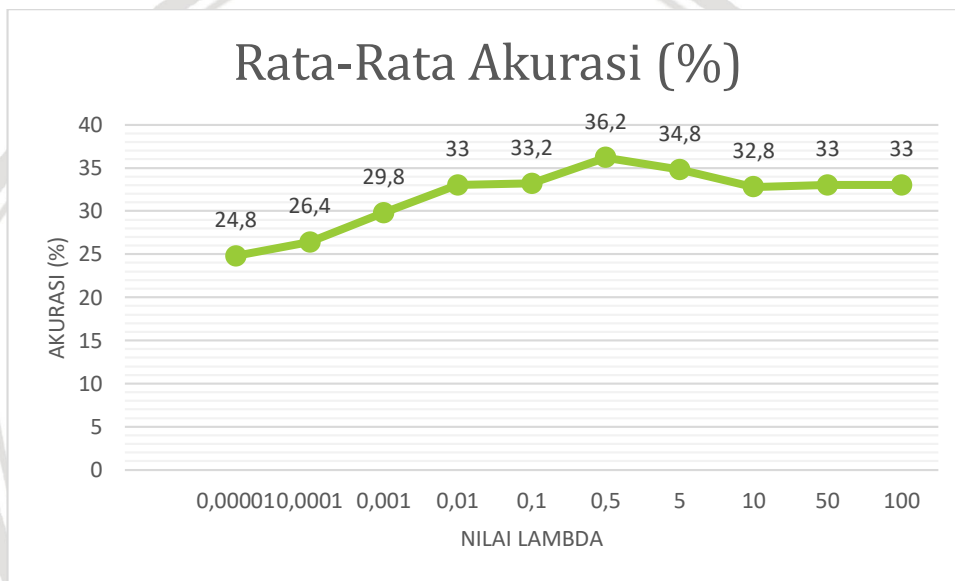
**Tabel 6.1 Hasil Pengujian Nilai *Lambda***

<i>Lambda</i>	Akurasi					Rata - rata Akurasi (%)
	Percobaan ke-					
	1	2	3	4	5	
0,00001	33%	16%	25%	25%	25%	24,8%
0,0001	16%	33%	33%	25%	25%	26,4%
0,001	25%	33%	33%	25%	33%	29,8%
0,01	50%	33%	33%	33%	16%	33%
0,1	25%	33%	50%	33%	25%	33,2%
0,5	41%	33%	33%	33%	41%	36,2%
5	16%	50%	25%	50%	33%	34,8%
10	16%	33%	33%	41%	41%	32,8%
50	41%	33%	25%	33%	33%	33%
100	41%	33%	33%	25%	33%	33%

Pada Tabel 6.1 telah dilakukan 5 percobaan pada setiap nilai  $\lambda$  dengan menggunakan pengujian akurasi. Pada  $\lambda$  dengan nilai 0,00001 dilakukan 5 kali percobaan, percobaan pertama dihasilkan nilai akurasi sebesar 33%. Percobaan kedua dihasilkan nilai akurasi sebesar 16%. Percobaan ketiga, keempat dan kelima dihasilkan nilai akurasi sebesar 25%. Berdasarkan 5 percobaan tersebut didapatkan nilai rata-rata akurasi sebesar 24,8%. Untuk nilai  $\lambda$  yang lainnya juga dilakukan percobaan sebanyak 5 kali.

### 6.1.2 Analisis Pengujian Nilai $\lambda$

Berdasarkan pada Tabel 6.1 dapat dilihat bahwa untuk rata-rata tingkat akurasi terbaik didapatkan pada saat parameter  $\lambda$  diberikan nilai 0,5 yaitu sebesar 36,2%. Pada Gambar 6.1 akan ditunjukkan grafik hasil pengujian parameter  $\lambda$ .



Gambar 6.1 Grafik Hasil Pengujian Parameter  $\lambda$

Pada Gambar 6.1 dapat diketahui bahwa nilai parameter  $\lambda$  berpengaruh pada akurasi sistem. Nilai parameter  $\lambda$  memiliki pengaruh terhadap jarak  $\text{margin}$  yaitu dapat menghasilkan nilai  $\text{hyperplane}$  yang baik (Hasanah, Pratama, & Cholissodin, 2016). Apabila dilihat keseluruhan, pada nilai  $\lambda$  yang bernilai kecil memberikan hasil rata-rata akurasi yang lebih kecil dibandingkan dengan nilai  $\lambda$  yang bernilai besar. Apabila nilai  $\lambda$  terlalu besar dapat mengakibatkan waktu komputasi untuk perhitungan matriks  $\text{Hessian}$  lebih lama. Hal tersebut disebabkan karena  $\text{augmented factor}$  ( $\lambda$ ) dapat mempengaruhi sistem menjadi lebih lambat untuk mencapai konvergensi dan proses pembelajaran menjadi tidak stabil. Jadi dapat disimpulkan bahwa nilai  $\lambda$  yang besar tidak membuat hasil rata-rata akurasi menjadi lebih tinggi, karena pada saat nilai  $\lambda$  bernilai kecil akan membuat lebar  $\text{margin}$  mengecil dan memberikan batas perpotongan atau  $\text{hyperplane}$  yang baik (Souza, 2010). Berdasarkan proses pengujian nilai  $\lambda$  tersebut, maka untuk proses pengujian selanjutnya digunakan  $\lambda = 0,5$ .



## 6.2 Pengujian Parameter *Gamma*

Pada pengujian parameter nilai *gamma* bertujuan untuk mengetahui pada nilai *gamma* berapa yang memberikan nilai akurasi terbaik untuk sistem dan bagaimana hasil analisis pada pengujian *gamma*.

### 6.2.1 Skenario Pengujian Nilai *Gamma*

Pada pengujian *gamma* akan dilakukan percobaan sebanyak 5 kali dengan beberapa variasi nilai *gamma* yang berbeda. Nilai yang dipakai pada pengujian *gamma* ada 10, dimulai dari 0,001 sampai 4. Untuk pengujian ini juga menggunakan parameter  $C = 1$ ,  $\lambda = 0,5$  dan  $\text{itermax} = 10$ . Kernel yang digunakan pada pengujian ini adalah kernel *polynomial*. Data yang digunakan pada pengujian *gamma* sebanyak 60 data. Data ini terbagi atas 10 data kelas Tipe I', 10 data kelas Tipe I, 10 data kelas Tipe II, 10 data kelas Tipe III, 10 data kelas Tipe IV, dan 10 data kelas Tipe V. Data kemudian dibagi menjadi data latih dan data uji secara acak. Perbandingan data latih dan data uji pada pengujian ini yaitu 80%:20%. Hasil pengujian terhadap parameter *gamma* ditunjukkan pada Tabel 6.2.

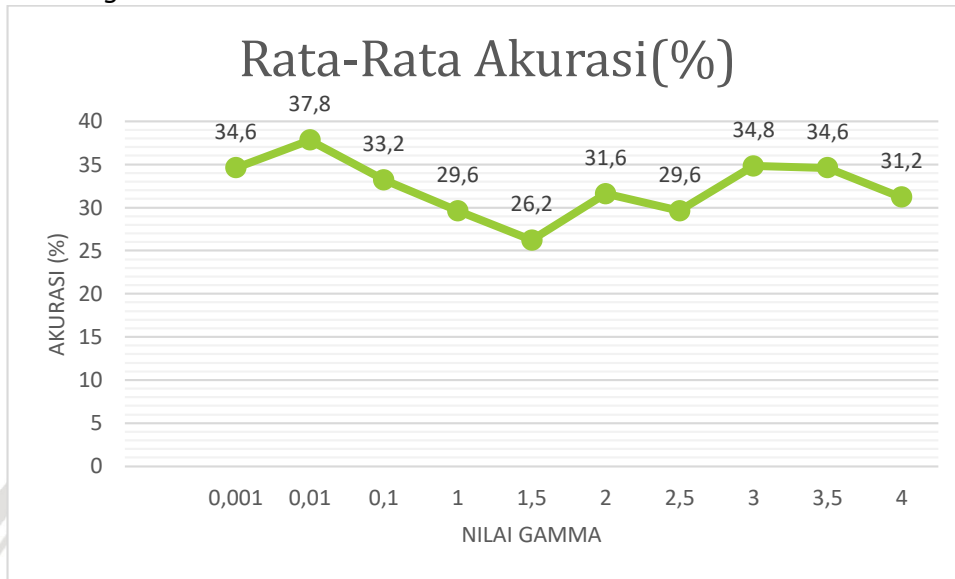
**Tabel 6.2 Hasil Pengujian Nilai *Gamma***

<i>Gamma</i>	Akurasi					Rata - rata Akurasi (%)
	Percobaan ke-					
	1	2	3	4	5	
0,001	41%	25%	41%	33%	33%	34,6%
0,01	33%	41%	41%	41%	33%	37,8%
0,1	33%	25%	25%	50%	33%	33,2%
1	16%	33%	25%	41%	33%	29,6%
1,5	8%	41%	16%	41%	25%	26,2%
2	33%	25%	50%	25%	25%	31,6%
2,5	33%	33%	25%	16%	41%	29,6%
3	25%	50%	41%	25%	33%	34,8%
3,5	16%	25%	50%	41%	41%	34,6%
4	41%	25%	33%	41%	16%	31,2%

Pada Tabel 6.2 telah dilakukan 5 percobaan pada setiap nilai *gamma* dengan menggunakan pengujian akurasi. Pada *gamma* dengan nilai 0,001 dilakukan 5 kali percobaan, percobaan pertama dihasilkan nilai akurasi sebesar 41%. Percobaan kedua dihasilkan nilai akurasi sebesar 25%. Percobaan ketiga dihasilkan nilai akurasi sebesar 41%. Percobaan keempat dan kelima dihasilkan nilai akurasi sebesar 33%. Berdasarkan 5 percobaan tersebut didapatkan nilai rata-rata akurasi sebesar 34,6%. Untuk nilai *gamma* yang lainnya juga dilakukan percobaan sebanyak 5 kali.

### 6.2.2 Analisis Pengujian Nilai *Gamma*

Berdasarkan pada Tabel 6.2 dapat dilihat bahwa untuk rata-rata tingkat akurasi terbaik didapatkan pada saat parameter *gamma* diberikan nilai 0.01 yaitu sebesar 37,8%. Pada Gambar 6.2 akan ditunjukkan grafik hasil pengujian parameter *gamma*.



Gambar 6.2 Grafik Hasil Pengujian Parameter *Gamma*

Pada Gambar 6.2 dapat diketahui bahwa nilai parameter *gamma* berpengaruh pada akurasi sistem. Tetapi pada grafik dapat dilihat jika nilai rata-rata akurasi cenderung tidak stabil pada setiap nilai parameter *gamma* yang diberikan. Hasil pengujian menunjukkan bahwa semakin kecil nilai konstanta *gamma* maka diperlukan iterasi semakin banyak dan akurasi yang dihasilkan dapat lebih baik. Jika semakin besar nilai  $\gamma$  yang digunakan, maka laju pembelajaran juga semakin cepat. Hal tersebut dapat menyebabkan penurunan tingkat akurasi sistem karena ketelitian sistem berkurang (Kurniawaty, Cholissodin, & Adikara, 2018). Parameter *gamma* pada *sequential learning* berfungsi untuk mengontrol kecepatan pada saat pelatihan. Berdasarkan proses pengujian nilai  $\gamma$  tersebut, maka untuk proses pengujian selanjutnya digunakan  $\gamma = 0,01$ .

### 6.3 Pengujian Parameter *Complexity* (C)

Pada pengujian parameter nilai *complexity* bertujuan untuk mengetahui pada nilai C berapa yang memberikan nilai akurasi terbaik untuk sistem.

#### 6.3.1 Skenario Pengujian Nilai *Complexity* (C)

Pada pengujian *complexity* akan dilakukan percobaan sebanyak 5 kali dengan beberapa variasi nilai *complexity* yang berbeda. Nilai yang dipakai pada pengujian *complexity* ada 10, dimulai dari 1 sampai 200. Untuk pengujian ini juga menggunakan parameter  $\lambda = 0,5$ ,  $\gamma = 0,01$  dan  $itermax = 10$ . Kernel yang digunakan pada pengujian ini adalah kernel *polynomial*. Data yang digunakan

pada pengujian *complexity* sebanyak 60 data. Data ini terbagi atas 10 data kelas Tipe I', 10 data kelas Tipe I, 10 data kelas Tipe II, 10 data kelas Tipe III, 10 data kelas Tipe IV, dan 10 data kelas Tipe V. Data kemudian dibagi menjadi data latih dan data uji secara acak. Perbandingan data latih dan data uji pada pengujian ini yaitu 80%:20%. Hasil pengujian terhadap parameter *complexity* ditunjukkan pada Tabel 6.3.

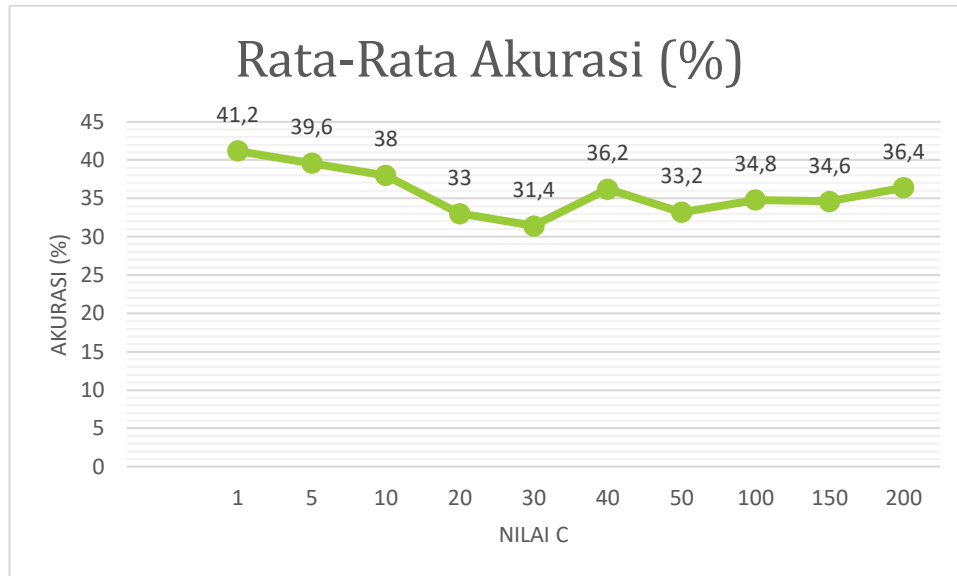
**Tabel 6.3 Hasil Pengujian Nilai *Complexity*(C)**

Nilai C	Akurasi					Rata - rata Akurasi (%)
	Percobaan ke-					
	1	2	3	4	5	
1	41%	33%	58%	41%	33%	41,2%
5	33%	33%	58%	41%	33%	39,6%
10	16%	50%	41%	58%	25%	38%
20	33%	50%	8%	33%	41%	33%
30	8%	41%	25%	33%	50%	31,4%
40	33%	41%	33%	33%	41%	36,2%
50	50%	25%	33%	25%	33%	33,2%
100	25%	33%	50%	33%	33%	34,8%
150	16%	33%	41%	58%	25%	34,6%
200	25%	41%	41%	25%	50%	36,4%

Pada Tabel 6.3 telah dilakukan 5 percobaan pada setiap nilai *C* dengan menggunakan pengujian akurasi. Pada *nilai C* dengan nilai 1 dilakukan 5 kali percobaan, percobaan pertama dihasilkan nilai akurasi sebesar 41%. Percobaan kedua dihasilkan nilai akurasi sebesar 33%. Percobaan ketiga dihasilkan nilai akurasi sebesar 58%. Percobaan keempat dihasilkan nilai akurasi sebesar 41% dan percobaan kelima dihasilkan nilai akurasi sebesar 33%. Berdasarkan 5 percobaan tersebut didapatkan nilai rata-rata akurasi sebesar 41,2%. Untuk nilai *C* yang lainnya juga dilakukan percobaan sebanyak 5 kali.

### 6.3.2 Analisis Pengujian Nilai *Complexity* (C)

Berdasarkan pada Tabel 6.3 dapat dilihat bahwa untuk rata-rata tingkat akurasi terbaik didapatkan pada saat parameter *complexity* diberikan nilai yaitu sebesar 41,2%. Pada Gambar 6.3 akan ditunjukkan grafik hasil pengujian parameter nilai *C*.



**Gambar 6.3 Grafik Hasil Pengujian Parameter Nilai C**

Pada Gambar 6.3 dapat diketahui bahwa nilai parameter  $C$  berpengaruh pada akurasi sistem. Tetapi pada grafik dapat dilihat jika nilai rata-rata akurasi mengalami penurunan pada saat nilai  $C$  bernilai 5 sampai 30. Nilai  $C$  bertujuan untuk meminimalisasikan *error* pada proses *training* bobot dan bias. Sehingga untuk mendapatkan nilai *error* yang kecil, digunakan nilai  $C$  yang kecil pula. Hal ini penting untuk memaksimalkan margin dan meminimalkan jumlah *slack*. Pada saat nilai  $C$  semakin mendekati 0, maka lebar margin pada *hyperplane* akan maksimal dan menyebabkan akurasi sistem yang semakin tinggi (Puspitasari, Ratnawati, & Widodo, 2018). Berdasarkan proses pengujian nilai  $C$  tersebut, maka untuk proses pengujian selanjutnya digunakan  $C = 1$ .

## 6.4 Pengujian Parameter Iterasi Maksimal

Pada pengujian parameter nilai iterasi maksimal bertujuan untuk mengetahui pada nilai iterasi maksimal berapa yang memberikan nilai akurasi terbaik untuk sistem.

### 6.4.1 Skenario Pengujian Nilai Iterasi Maksimal

Pada pengujian nilai iterasi maksimal akan dilakukan percobaan sebanyak 5 kali dengan beberapa variasi nilai nilai iterasi maksimal yang berbeda. Nilai yang dipakai pada pengujian nilai iterasi maksimal ada 10, dimulai dari 1 sampai 200. Untuk pengujian ini juga menggunakan parameter  $\lambda = 0,5$ ,  $\gamma = 0,01$  dan  $C = 1$ . Kernel yang digunakan pada pengujian ini adalah kernel *polynomial*. Data yang digunakan pada pengujian nilai iterasi maksimal sebanyak 60 data. Data ini terbagi atas 10 data kelas Tipe I', 10 data kelas Tipe I, 10 data kelas Tipe II, 10 data kelas Tipe III, 10 data kelas Tipe IV, dan 10 data kelas Tipe V. Data kemudian dibagi menjadi data latih dan data uji secara acak. Perbandingan data latih dan data uji pada pengujian ini yaitu 80%:20%. Hasil pengujian terhadap parameter *complexity* ditunjukkan pada Tabel 6.4.

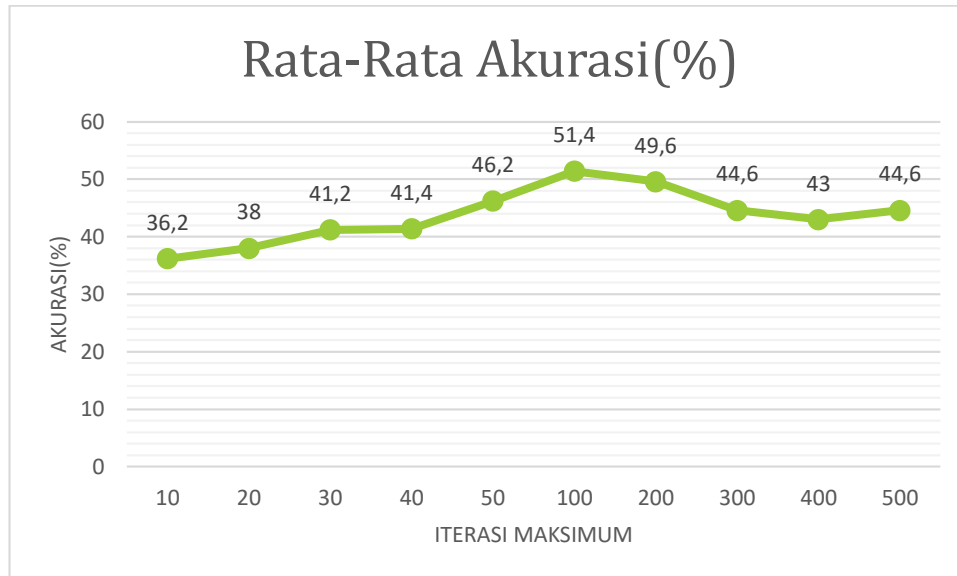
Tabel 6.4 Hasil Pengujian Nilai Iterasi Maksimal

Iterasi Maksimal	Akurasi					Rata - rata Akurasi (%)
	Percobaan ke-					
	1	2	3	4	5	
10	41%	33%	16%	33%	58%	36,2%
20	50%	25%	33%	41%	41%	38%
30	41%	50%	33%	41%	41%	41,2%
40	50%	33%	33%	58%	33%	41,4%
50	50%	41%	58%	41%	41%	46,2%
100	41%	50%	50%	58%	58%	51,4%
200	41%	58%	58%	50%	41%	49,6%
300	50%	33%	58%	41%	41%	44,6%
400	50%	50%	33%	41%	41%	43%
500	58%	33%	41%	41%	50%	44,6%

Pada Tabel 6.4 telah dilakukan 5 percobaan pada setiap nilai iterasi maksimal dengan menggunakan pengujian akurasi. Pada nilai iterasi maksimal dengan nilai 10 dilakukan 5 kali percobaan, percobaan pertama dihasilkan nilai akurasi sebesar 41%. Percobaan kedua dihasilkan nilai akurasi sebesar 33%. Percobaan ketiga dihasilkan nilai akurasi sebesar 16%. Percobaan keempat dihasilkan nilai akurasi sebesar 33% dan percobaan kelima dihasilkan nilai akurasi sebesar 58%. Berdasarkan 5 percobaan tersebut didapatkan nilai rata-rata akurasi sebesar 36,2%. Untuk nilai C yang lainnya juga dilakukan percobaan sebanyak 5 kali.

#### 6.4.2 Analisis Pengujian Nilai Iterasi Maksimal

Berdasarkan pada Tabel 6.4 dapat dilihat bahwa untuk rata-rata tingkat akurasi terbaik didapatkan pada saat parameter nilai iterasi maksimal diberikan nilai 100 yaitu sebesar 51,4%. Pada Gambar 6.4 akan ditunjukkan grafik hasil pengujian parameter nilai iterasi maksimal.



**Gambar 6.4 Grafik Hasil Pengujian Parameter Iterasi Maksimal**

Pada Gambar 6.4 dapat diketahui bahwa nilai iterasi maksimal berpengaruh pada akurasi sistem. Pada grafik dapat dilihat jika nilai rata-rata akurasi mengalami kenaikan pada saat nilai iterasi maksimal bernilai 10 sampai 100, tetapi kemudian disaat iterasi maksimal bernilai 200 sampai 500 mengalami penurunan. Hal tersebut menunjukkan bahwa jumlah iterasi yang besar tidak menjamin tingginya nilai akurasi karena nilai  $\alpha$  belum mencapai nilai konvergen. Pada pengujian iterasi maksimum ini nilai iterasi menghentikan pengaruh pada perubahan nilai  $\alpha$  ( $\alpha$ ). Tidak seimbang rasio *support vector* dan beberapa data mempunyai letak yang jauh dari bidang pemisah (*hyperplane*) yang ideal menjadi penyebab turunnya akurasi disaat iterasi bertambah (Harahap, Muflikhah, & Rahayudi, 2018).

## 6.5 Pengujian Nilai Jarak ( $d$ )

Pada pengujian parameter nilai jarak bertujuan untuk mengetahui pada nilai jarak berapa yang memberikan nilai akurasi terbaik untuk sistem.

### 6.5.1 Skenario Pengujian Nilai Jarak ( $d$ )

Pengujian nilai  $d$  merupakan pengujian parameter ekstraksi fitur GLCM untuk menentukan hasil pembentukan nilai matriks awal berdasarkan jarak *pixel*. Tujuan dari pengujian ini yaitu untuk mengetahui pengaruh nilai parameter  $d$  terhadap hasil akurasi. Pada pengujian nilai  $d$  akan dilakukan percobaan sebanyak 5 kali dengan beberapa variasi nilai  $d$  yang berbeda. Untuk parameter SVM yang digunakan pada pengujian ini yaitu parameter  $\lambda = 0.5$ ,  $\gamma = 0.01$  dan  $C = 1$ . Kernel yang digunakan pada pengujian ini adalah kernel *polynomial*. Data yang digunakan pada pengujian nilai iterasi maksimal sebanyak 60 data. Data ini terbagi atas 10 data kelas Tipe I', 10 data kelas Tipe I, 10 data kelas Tipe II, 10 data kelas Tipe III, 10 data kelas Tipe IV, dan 10 data kelas Tipe V. Data kemudian dibagi menjadi data latih dan data uji secara acak. Perbandingan data latih dan

data uji pada pengujian ini yaitu 80%:20%. Hasil pengujian nilai  $d$  ditunjukkan pada Tabel 6.5.

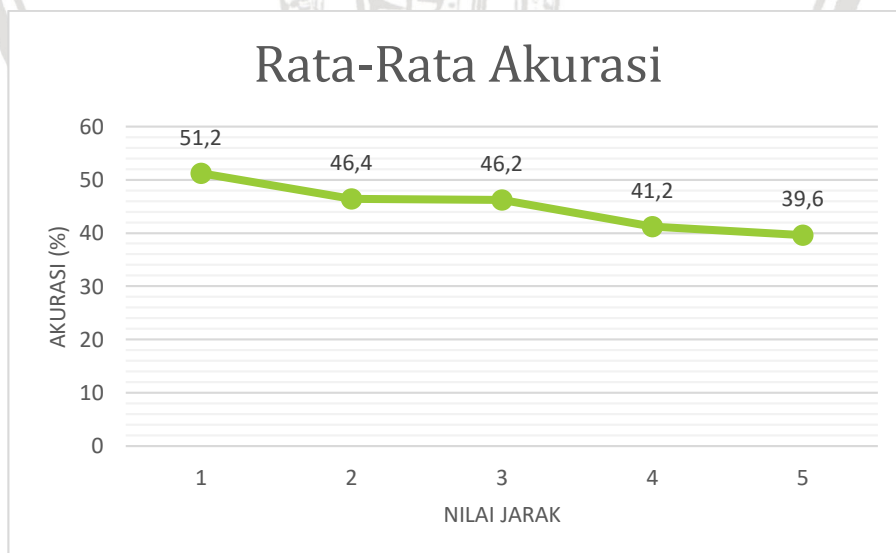
**Tabel 6.5 Hasil Pengujian Nilai Jarak ( $d$ )**

Jarak	Akurasi					Akurasi (%)
	Percobaan ke-					
	1	2	3	4	5	
1	50%	58%	41%	66%	41%	51,2%
2	41%	50%	41%	42%	58%	46,4%
3	41%	41%	41%	58%	50%	46,2%
4	41%	41%	50%	33%	41%	41,2%
5	41%	50%	41%	33%	33%	39,6%

Pada Tabel 6.5 telah dilakukan 5 percobaan pada setiap nilai  $d$  dengan menggunakan pengujian akurasi. Pada nilai  $d$  dengan nilai 1 dilakukan 5 kali percobaan, percobaan pertama dihasilkan nilai akurasi sebesar 50%. Percobaan kedua dihasilkan nilai akurasi sebesar 58%. Percobaan ketiga dihasilkan nilai akurasi sebesar 41%. Percobaan keempat dihasilkan nilai akurasi sebesar 66% dan percobaan kelima dihasilkan nilai akurasi sebesar 41%. Berdasarkan 5 percobaan tersebut didapatkan nilai rata-rata akurasi sebesar 51,2%. Untuk nilai  $d$  yang lainnya juga dilakukan percobaan sebanyak 5 kali.

### 6.5.2 Analisis Pengujian Nilai Jarak ( $d$ )

Berdasarkan pada Tabel 6.5 dapat dilihat bahwa untuk rata-rata tingkat akurasi terbaik didapatkan pada saat parameter nilai jarak diberikan nilai 1 yaitu sebesar 51,2%. Pada Gambar 6.5 akan ditunjukkan grafik hasil pengujian parameter nilai iterasi maksimal.



**Gambar 6.5 Grafik Hasil Pengujian Parameter Nilai Jarak ( $d$ )**

Pada Gambar 6.4 dapat diketahui bahwa nilai jarak berpengaruh pada akurasi sistem. Pada grafik dapat dilihat jika nilai rata-rata akurasi mengalami penurunan terus menerus disaat nilai d bernilai 2 sampai 5. Parameter nilai akurasi menurun karena pengaruh pembentukan matrik kookurensi. Semakin jauh jarak ketetangaan pasangan pixel akan menghasilkan pengenalan nilai yang semakin jauh atau buruk (Hidanti, Zahra & Isnanto, 2018).





Dari 5 hasil pengujian yang dilakukan, didapatkan nilai parameter pengujian terbaik yang menghasilkan rata-rata akurasi terbesar pada tiap-tiap pengujian nya. Nilai- nilai tersebut dapat menjadi rekomendasi untuk proses *training* pada sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine*. Berikut parameter yang menjadi rekomendasi untuk pelatihan ditunjukkan pada Tabel 6.6.

**Tabel 6.6 Hasil Nilai Parameter Terbaik Dari Setiap Pengujian**



Skenario Pengujian	Nilai Parameter Terbaik
<i>Lambda</i>	0.5
<i>Gamma</i>	0.01
<i>Complexity</i>	1
Iterasi Maksimal	100
Nilai Jarak	1

Berikut merupakan hasil percobaan pengujian klasifikasi data uji yang belum diketahui kelasnya berdasarkan nilai rekomendasi parameter terbaik.

**Tabel 6.7 Hasil Klasifikasi untuk data uji yang belum diketahui kelasnya**

Data	Kuadran	Kelas Klasifikasi	Jenis Kelamin	Hasil Klasifikasi
	1	TYPE_V	PRIA	PRIA
	2	TYPE_V		
	3	TYPE_II		
	4	TYPE_II		
	1	TYPE_I'	WANITA	WANITA
	2	TYPE_I'		
	3	TYPE_I'		
	4	TYPE_I'		
	1	TYPE_II	PRIA	WANITA
	2	TYPE_I'		
	3	TYPE_I		
	4	TYPE_I'		
	1	TYPE_V	WANITA	WANITA
	2	TYPE_V		
	3	TYPE_V		
	4	TYPE_V		








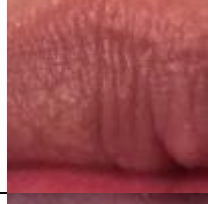





	1	TYPE_I'	WANITA	WANITA
	2	TYPE_I'		
	3	TYPE_I		
	4	TYPE_II		
	1	TYPE_V	PRIA	PRIA
	2	TYPE_I		
	3	TYPE_V		
	4	TYPE_II		

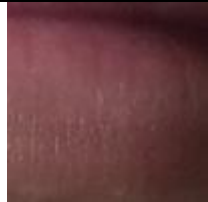
Dari hasil klasifikasi Tabel 6.7 untuk 6 data yang belum diketahui kelas kuadran bibirnya, sistem mampu memberikan hasil klasifikasi jenis kelamin dengan tepat. Penentuan jenis kelamin dilakukan berdasarkan rule klasifikasi jenis kelamin seperti pada Tabel 2.2. Dari Tabel 6.6 juga terdapat 1 data yang belum tepat memberikan hasil klasifikasi terhadap jenis kelamin. Hal ini disebabkan karna nilai tekstur bibir pada data Pria(2) memang masuk kedalam kelas yang merepresentasikan klasifikasi rule untuk jenis kelamin wanita. Untuk kedepannya, dapat dilakukan penambahan data training untuk pelatihan sistem ini agar dapat dengan valid memberikan hasil keluaran klasifikasi. Dikarenakan jumlah dataset yang masih sedikit membuat pembentukan hyperplane pada proses SVM masih terlalu sederhana sehingga belum mampu dengan benar memberikan klasifikasi data yang belum dikenali sebelumnya.

Kemudian Tabel 6.8 merupakan hasil percobaan pengujian klasifikasi untuk data yang telah dilakukan pelabelan kelas berdasarkan nilai rekomendasi parameter terbaik.

**Tabel 6.8 Hasil Klasifikasi**

No	Data	Nama Data	Kelas Sebenarnya	Kelas Klasifikasi	Hasil
1		TIV-B13K2L	TYPE_IV	TYPE_V	Salah
2		TV-B11K6L	TYPE_V	TYPE_V	Benar
3		TII-B4K3L	TYPE_II	TYPE_II	Benar

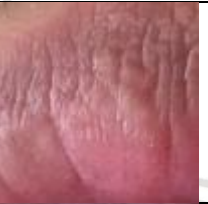

4		TII- B12K7W	TYPE_II	TYPE_II	Benar
5		TI-B5K7W	TYPE_I	TYPE_II	Salah
6		TIII-B5K2L	TYPE_III	TYPE_III	Salah
7		TIII- B23K6W	TYPE_III	TYPE_III	Benar
8		TI-B15K7L	TYPE_I	TYPE_I	Benar
9		TV-B12K3L	TYPE_V	TYPE_I'	Salah
10		TI'-B1K7L	TYPE_I'	TYPE_I'	Benar
11		TIV- B30K6L	TYPE_IV	TYPE_V	Salah



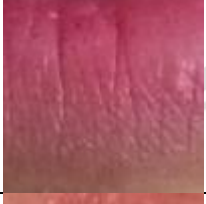
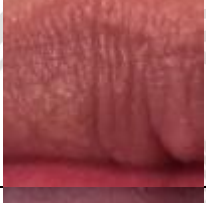




12		TI'-B6K7L	TYPE_I'	TYPE_I'	Benar
----	---	-----------	---------	---------	-------



Pada Tabel 6.8 dapat dilihat untuk hasil klasifikasi yang benar berjumlah 7 data dan data yang tidak sesuai dengan kelas nya ada 5 data. Hal ini disebabkan karena kurangnya data untuk pelatihan pada kelas tersebut. Misal pada kasus data nomer 1 dan 11, nilai support vector pada kelas Tipe IV dan kelas tipe V memiliki margin yang berdekatan diantara hyperplane pemisahannya, sehingga terjadi ketidaktepatan klasifikasi kedalam kelas yang seharusnya. Solusi yang dapat dilakukan yaitu dilakukan yaitu melakukan perbaikan citra agar dapat membedakan pola kelas dengan baik dan memperbanyak pelatihan untuk kelas Tipe IV agar dapat dengan tepat memberikan hyperplane yang optimal untuk dapat memberikan hasil klasifikasi yang benar.

Kemudian juga telah dilakukan uji korelasi untuk mengetahui keterkaitan hubungan antar fitur. Fitur yang memiliki korelasi yang kuat antar nilai-nilai fitur tersebut yaitu dipilih dengan nilai range antara 0.7 sampai dengan 1. Kemudian didapatkan fitur yang memiliki korelasi yang kuat yaitu ASM dengan sudut 0° dan 45°, *contrast* dengan sudut 0°, 45° dan 90°, *correlation* 45° dan 135°, dan IDM dengan sudut 45°, 90° dan 135°. Kemudian dilakukan percobaan pengujian klasifikasi untuk data dengan beberapa fitur yang diambil dari uji korelasi. Hasil pengujian ditunjukkan pada tabel 6.9.

**Tabel 6.9 Perbandingan Hasil Klasifikasi Menggunakan Semua Fitur terhadap Fitur berdasarkan Uji Korelasi**

No	Data	Nama Data	Hasil Klasifikasi		Hasil
			Menggunakan Semua Fitur	Menggunakan Fitur berdasarkan Uji Korelasi	
1		TIV-B13K2L	TYPE_IV	TYPE_IV	Sama
2		TV-B11K6L	TYPE_V	TYPE_V	Sama

3		TII- B4K3L	TYPE_II	TYPE_II	Sama
4		TII- B12K7W	TYPE_II	TYPE_II	Sama
5		TI- B5K7W	TYPE_I	TYPE_I	Sama
6		TIII- B5K2L	TYPE_III	TYPE_III	Sama
7		TIII- B23K6W	TYPE_III	TYPE_III	Sama
8		TI- B15K7L	TYPE_I	TYPE_I	Sama
9		TV- B12K3L	TYPE_V	TYPE_V	Sama
10		TI'- B1K7L	TYPE_I'	TYPE_I'	Sama

11		TIV- B30K6L	TYPE_IV	TYPE_IV	Sama
12		TI'- B6K7L	TYPE_I'	TYPE_I'	Sama

Pada Tabel 6.9 dapat dilihat untuk hasil klasifikasi dengan menggunakan semua fitur dan hasil klasifikasi dengan beberapa fitur memberikan hasil yang sama. Itu artinya pada penelitian ini, ketidaktepatan untuk sistem melakukan klasifikasi bukan karna pengaruh jumlah fitur yang dipakai untuk setiap kelas data.

Kemudian juga telah dilakukan perhitungan rata-rata nilai ekstraksi ciri GLCM untuk mengetahui perbedaan nilai fitur untuk antar kelas. Berikut diambil 12 sampel data latih citra bibir yang telah dihitung nilai rata-ratanya, hasil perhitungan ditunjukkan pada tabel 6.10.

**Tabel 6.10 Nilai Rata-Rata Ekstraksi Ciri GLCM**

Data	Nama	ASM	CON	COR	IDM	Tipe Bibir	Kelas
1	TI'-B1K7L	0,048	1,222	0,047	0,692	Tipe I'	1
2	TI'-B6K7L	0,030	1,001	0,054	0,694	Tipe I'	1
3	TI-B15K7L	0,029	0,756	0,052	0,742	Tipe I	2
4	TI-B5K7W	0,034	0,759	0,064	0,771	Tipe I	2
5	TII-B4K3L	0,018	1,204	0,045	0,656	Tipe II	3
6	TII-B12K7W	0,024	1,029	0,059	0,683	Tipe II	3
7	TIII-B5K2L	0,022	1,859	0,046	0,618	Tipe III	4
8	TIII-B23K6W	0,024	1,269	0,048	0,671	Tipe III	4
9	TIV-B30K6L	0,014	3,631	0,042	0,540	Tipe IV	5
10	TIV-B13K2L	0,013	3,660	0,042	0,550	Tipe IV	5
11	TV-B12K3L	0,064	0,386	0,085	0,822	Tipe V	6
12	TV-B11K6L	0,055	0,550	0,069	0,614	Tipe V	6

Pada tabel 6.10 dapat dianalisis bahwa perbedaan nilai fitur antar kelas memiliki perbedaan nilai ekstraksi ciri yang sudah cukup signifikan, sehingga sudah mampu melakukan proses pengenalan pada saat proses klasifikasi menggunakan SVM. Nilai pada tabel ini didapat dari rata-rata nilai fitur ekstraksi ciri dengan seluruh sudut.

## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan implementasi dan hasil pengujian mengenai sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine* dapat diambil beberapa kesimpulan yakni sebagai berikut.

1. Berdasarkan hasil pengujian yang telah dilakukan pada implementasi permasalahan klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine*, sistem masih kurang dapat memberikan hasil klasifikasi sesuai dengan kelasnya. Hal ini karena tidak adanya seleksi fitur untuk setiap data agar mewakili ciri yang spesifik sebagai pembeda dengan kelas lainnya. Pola antar kelas masih belum bisa dipisahkan secara baik, masih banyak noise dalam lingkup kelas pada data penelitian ini dikarenakan pelabelan kelas yang dilakukan masih manual. Proses klasifikasi jenis kelamin pria dan wanita ini juga masih menggunakan *dataset* yang terbatas yaitu 60 data. *Dataset* ini terbagi atas 10 data kelas Tipe I', 10 data kelas Tipe I, 10 data kelas Tipe II, 10 data kelas Tipe III, 10 data kelas Tipe IV, dan 10 data kelas Tipe V. Nilai rata-rata akurasi hanya mampu mencapai nilai 51,4% dengan skenario pengujian yaitu rasio data 80% data *training* : 20% data *testing*, menggunakan kernel *polynomial* dan nilai parameter yaitu  $\lambda=0,5$ ,  $\gamma=0,01$ ,  $C=1$ , *itermax* = 100 dan jarak=1.
2. Pengaruh variasi nilai parameter *lambda*, *gamma*, *C*, iterasi maksimal dan *d* pada pengujian memberikan hasil yaitu hampir semua nilai akurasi data nilainya tidak optimal. Akurasi yang rendah tersebut disebabkan karena nilai tekstur pada beberapa dataset yang kelasnya berbeda memiliki kemiripan dan perbedaan pola pada setiap kelas tidak signifikan. Walaupun rata-rata akurasi yang didapatkan masih rendah, sistem sudah cukup baik untuk mengklasifikasikan tipe bibir ke dalam 6 kelas yaitu kelas Tipe I', Tipe I, Tipe II, Tipe III, Tipe IV, Tipe V. Kemudian dari hasil klasifikasi tipe bibir, didapatkan hasil keluaran jenis kelamin berdasarkan rule Pahanwala-Parekh untuk pria dan wanita dengan komposisi nilai kuadran yang memberikan hasil yang tepat walaupun ada beberapa yang masih salah.

### 7.2 Saran

Agar dapat meningkatkan hasil yang diperoleh pada sistem klasifikasi pola sidik bibir untuk menentukan jenis kelamin manusia dengan metode *gray level co-occurrence matrix* dan *support vector machine*, diharapkan untuk penelitian selanjutnya untuk bisa melakukan beberapa perbaikan seperti menambahkan beberapa tahapan pre-processing agar membuat kualitas citra menjadi lebih baik dan mencoba menggunakan metode ekstraksi fitur lainnya. Kemudian perlu melakukan pelatihan data yang lebih banyak agar sistem dapat dengan baik

mengenali data baru dan dapat memisahkan dengan baik kelas-kelas yang termasuk didalamnya. Selanjutnya dapat melakukan penelitian lebih lanjut untuk kasus multi-class pada SVM dengan menggunakan strategi *One Against One* ataupun strategi yang lainnya. Kemudian untuk pengujian sistem dapat melakukan pengujian GLCM lebih banyak lagi dengan menguji setiap ciri fitur GLCM terhadap satu sudut dan satu nilai jarak. Hal ini dilakukan agar dapat mengetahui pada ciri tekstur, sudut dan jarak pada nilai berapa yang memberikan pengaruh dan nilai akurasi yang baik untuk pembentukan dataset dengan nilai atribut data yang tepat untuk digunakan pada proses klasifikasi menggunakan SVM. Kemudian kedepannya dapat memanfaatkan sidik bibir untuk mengenali pemiliknya dan menambahkan *dataset* ketika posisi bibir sedang tersenyum, terbuka dan dalam keadaan mengecup agar menghasilkan hasil yang lebih akurat.



## DAFTAR REFERENSI

- BNPB, 2017. *BADAN NASIONAL PENANGGULANGAN BENCANA*. [Online] Available at: <https://www.bnpb.go.id> [Diakses 06 Desember 2017].
- Chang, K. L., Liao, S. K., Tseng, T. W. & Liao, C. Y., 2015. An ANP based TOPSIS approach for Taiwanese service apartment location selection. *Asia Pacific Management Review* 20, pp. 49-55.
- Datta, P., Sood, S. & J.R, S., 2012. Cheiloscopy as a tool for human identification.. *Indian Journal of Forensic Odontology*, Volume 5, pp. 17-23.
- Dewi, A. P., 2017. *Antara News*. [Online] Available at: <https://www.antaraneews.com> [Diakses 06 Desember 2017].
- Dewi, R. K. & Ginardi, R. H., 2014. Identifikasi Penyakit Pada Daun Tebu Dengan Gray Level Co-Occurrence Matrix Dan Color Moments. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, Volume 1, pp. 70-77.
- Dongarwar, G. R., Bhowate, R. R. & Degwekar, S. S., 2013. Cheiloscopy-Method of Person Identification and Sex Determination. *Open Access Scientific Reports*, 2(1).
- Eckert, W. G., 1997. Dalam: *Introduction to Forensic Science 2nd edition*. Boca Raton: CRC Press, pp. 304-306.
- Gencer, C. & Gurpinar, D., 2007. Analytic network process in supplier selection: A case study in an electronic firm. *Applied Mathematical Modelling* 31 *ScientDirect*, pp. 2475-2486.
- Harahap, E. H., Muflikhah, L. & Rahayudi, B., 2018. Implementasi Algoritma Support Vector Machine (SVM) Untuk Penentuan Seleksi Atlet Pencak Silat. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 3843-3848.
- Haralick, R. M., Shanmugam, K. & Dinstein, I., 1973. Textural Features for Image Classification. pp. 610 - 621.
- Hidanti, M., Zahra, A. A. & Isnanto, R. R., 2016. Sistem Identifikasi Jenis Tanaman Obat Menggunakan Matriks Kookurensi Aras Keabuan (GLCM) dan Jarak Canberra. *Proceedings Seminar Nasional Teknik Elektro (FORTEI 2016)*, p. 115.
- Jain, A., 1989. *Fundamentals of Digital Image Processing*. s.l.:Prentice-Hall International.
- Jung, U. & Seo, D., 2010. An ANP approach for R&D project evaluation based on interdependencies between research objective and evaluation criteria. *Decision Support Systems* 49, p. 335-342.



- Kadir, A. & Susanto, A., 2012. *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andi.
- Kasprzak, J., 1990. Possibilities of Cheiloscopy. *Forensic Science International*, pp. 145-151.
- K, R., RS, N. & PC, A., 2011. Study Of The Effect Of Age Changes On Lip Print Pattern And Its Reliability In Sex Determination.. *J Forensic Odontostomatol*, pp. 45-51.
- K, R., RS, N. & PC, A., t.thn. Study Of The Effect Of Age Changes On Lip Print Pattern And Its Reliability In Sex Determination..
- Kunfa, Kunfa, Z., Jianmin, H. & Shouwei, L., 2010. A Measurement Model of Innovative Talents Basing on ANP-TOPSIS. *IEEE*, pp. 1-3.
- Kurniawan, N. Z., Rasmana, S. T. & Triwidyastuti, Y., 2016. Identifikasi Jenis Penyakit Daun Tembakau Menggunakan Metode Gray Level Co-Occurrence Matrix (GlcM) Dan Support Vector Machine (Svm). pp. 158-163.
- Kurniawan, N. Z., Rasmana, S. T. & Triwidyastuti, Y., 2016. Identifikasi Jenis Penyakit Daun Tembakau Menggunakan Metode Gray Level Co-Occurrence Matrix (GLCM) Dan Support Vector Machine (SVM). *JCONES*, Volume 5, pp. 158-163.
- Kurniawaty, D., Cholissodin, I. & Adikara, P. P., 2018. Klasifikasi Gangguan Jiwa Skizofrenia Menggunakan Algoritme Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 1866-1873.
- Lathief, D., 2016. Klasifikasi Jenis Kelamin Manusia menggunakan Citra Wajah berbasis Metode Support Vector Machine (SVM).
- Lidya, S. K., Sitompul, O. S. & Efendi, S., 2015. Sentiment Analysis Pada Teks Bahasa Indonesia Menggunakan Support Vector Machine(SVM) dan K-Nearest Neighbor(K-NN). *Seminar Nasional Teknologi Informasi dan Komunikasi*.
- Munawaroh, S. & Sutanto, F. A., 2010. Pengolah Citra Digital untuk Identifikasi Uang Kertas. *Jurnal Teknologi Informasi DINAMIK*, Volume XV, pp. 34-40.
- Munir, R., 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Infomartika .
- Neneng, Adi, K. & Isnanto, R. R., 2016. Support Vector Machine Untuk Klasifikasi Citra Jenis Daging Berdasarkan Tekstur Menggunakan Ekstraksi Ciri Gray Level Co-Occurrence Matrices (GLCM). *Jurnal Sistem Informasi Bisnis*.
- Patil, A. M., Patil, D. S. & Patil, P. S., 2016. Iris Recognition using Gray Level Co-occurrence Matrix and Hausdorff Dimension. *International Journal of Computer Applications*.
- Poetra, A. R., Mahmudy, W. F. & Indriati, 2015. Implementasi ANP dan TOPSIS Untuk Penentuan Promosi Jabatan Struktural. *Jurnal Skripsi*, pp. 1-10.

- Prawestiningtyas, E. & Algozi, A. M., 2009. Forensic Identification Based on Both Primary and Secondary Examination Priority in. *Jurnal Kedokteran Brawijaya*, Volume XXV.
- Purba, D., 2010. *Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab*. Yogyakarta: Andi.
- Puspitasari, A. M., Ratnawati, D. E. & Widodo, A., 2018. Klasifikasi Penyakit Gigi Dan Mulut Menggunakan Metode Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 802-810.
- Samant, R., Deshpande, S. & Jadhao, A., 2015. Survey on Multi Criteria Decision Making Methods. *International Journal of Innovative Research in Science, Engineering and Technology*, 4(8), pp. 7175-7178.
- Saxena, S., Sharma, P. & Gupta, N., 2010. Experimental studies of forensic odontology to aid identification process. *Journal of Forensic Dental Sciences*, Volume 2, pp. 69-76.
- Sekharan, P., 2013. *Lip Forensics for Criminal Identification*.
- Septadina, I. S., 2015. Identifikasi Individu dan Jenis Kelamin Berdasarkan Pola Sidik Bibir. April, Volume II, pp. 231-236.
- Surya, R. A., Fadlil, A. & Yudhana, A., 2017. Ekstraksi Ciri Metode Gray Level Co-Occurrence Matrix (GLCM) dan Filter Gabor Untuk Klasifikasi Citra Batik Pekalongan. *Jurnal Pengembangan IT*, Volume 02.
- Syafril, N. S., Dr. Ir. Bambang Hidayat, D. & H. Fahmi Oscandar, d. M. S., 2017. IDENTIFIKASI POLA SIDIK BIBIR MENGGUNAKAN METODE CBIR BASED ON GABOR WAVELET DAN KLASIFIKASI K-NN UNTUK APLIKASI BIDANG FORENSIK. *e-Proceeding of Engineering*, Volume 4, p. 527.
- Vijayakumar, S. & Wu, S., 1999. Sequential Support Vector Classifiers and Regression. *International Conference on SoftComputing*, pp. 610-619.
- Wu, L.-Y. & Yang, Y.-Z., 2008. *TOPSIS METHOD FOR GREEN VENDOR SELECTION IN COAL INDUSTRY GROUP*. Henan, China, School of Energy Science & Engineering, Henan Polytechnic University, Jiaozuo, pp. 1721-1725.
- Zulpe, N. & Pawar, V., 2012. GLCM Textural Features for Brain Tumor Classification. *IJCSI International Journal of Computer Science*, Volume 9, p. 6.