

**KLASIFIKASI PENYAKIT SKIZOFRENIA DAN EPISODE DEPRESI
PADA GANGGUAN KEJIWAAN DENGAN MENGGUNAKAN
METODE *SUPPORT VECTOR MACHINE* (SVM)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Silvia Aprilla

NIM: 145150201111071



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



PENGESAHAN

KLASIFIKASI PENYAKIT SKIZOFRENIA DAN EPISODE DEPRESI PADA GANGGUAN KEJIWAAN DENGAN MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE* (SVM)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :
Silvia Aprilla
NIM: 145150201111071

Skripsi ini telah diuji dan dinyatakan lulus pada
10 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II


M. Tanzil Furgon, S.Kom, M.CompSc
NIP: 19820930 200801 1 004


M. Ali Fauzi, S.Kom, M.Kom
NIK: 201502 890101 1 001

Mengetahui

Ketua Jurusan Teknik Informatika




M. Astoro 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,



Silvia Aprilla
NIM: 145150201111071



KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah melimpahkan rahmat-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Klasifikasi Penyakit Skizofrenia Dan Episode Depresi Pada Gangguan Kejiwaan Dengan Menggunakan Metode *Support Vector Machine (SVM)*". Dalam proses penyusunan dan penyelesaian skripsi ini penulis mendapat bantuan maupun dorongan yang baik dalam berbagai bentuk dari berbagai pihak. Oleh karena itu, penulis mengucapkan banyak terimakasih kepada:

1. Bapak Muhammad Tanzil Furqon, S.Kom, M.CompSc., selaku dosen pembimbing I yang telah membimbing, memberi arahan, dan ilmu kepada penulis selama proses pengerjaan skripsi penulis.
2. Bapak Mochammad Ali Fauzi, S.Kom, M.Kom., selaku dosen pembimbing II yang telah membimbing, memberi arahan, dan ilmu kepada penulis selama proses pengerjaan skripsi penulis.
3. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Tri Astoto Kurninawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Agus Wahyu Widodo, S.T., M.Cs selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
6. Dokter Nur Aida, dr. SpKJ yang telah bersedia membantu penulis dalam melancarkan pekerjaan skripsi penulis.
7. Orangtua dan adik penulis yang telah memberikan dukungan serta doa untuk penulis dalam kelancaran pekerjaan skripsi penulis.
8. Seluruh bapak dan ibu dosen yang telah memberikan ilmu serta mendidik penulis selama menempuh Pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
9. Teman baik penulis Articha, Patrisia, Windy, dan Sisca yang telah mendukung penulis serta memberikan doa untuk penulis selama pengerjaan skripsi penulis.
10. Teman F-Zoo yaitu Imada, Wanda, Ratna, Rizky, Egi, Rayza, Rahmat, Chandra, dan Harits yang telah memberikan semangat serta membantu penulis selama masa perkuliahan.
11. Sara dan Tryse yang telah memberi dukungan, memberi masukan, serta menemani penulis selama proses pengerjaan skripsi.
12. Saudari Eva, Tania, dan Ingrid yang telah menghibur, mendukung penulis dalam doa, serta membantu pertumbuhan iman penulis selama perkuliahan.
13. Josua Fernando yang telah memberikan masukan, menemani, mendoakan, dan menjadi motivasi penulis dalam menyelesaikan skripsi penulis.
14. Teman-teman PMK Daniel dan BIOS yang telah memberikan dukungan dengan berbagai bentuk kepada penulis.



15. Teman-teman Program Studi Teknik Informatika yang telah memberikan doa serta dukungan selama penulis menempuh perkuliahan di Fakultas Ilmu Komputer Universitas Brawijaya.
16. Seluruh pihak yang tidak dapat disebutkan oleh penulis satu per satu yang telah memberi dukungan serta doa kepada penulis dalam penyelesaian skripsi penulis.

Semoga seluruh doa dan dukungan yang telah diberikan kepada penulis dapat dibalas oleh Tuhan Yang Maha Esa. Penulis menyadari bahwa skripsi penulis tidak lepas dari kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun agar dapat memperbaiki kekurangan yang dimiliki oleh skripsi ini. Harapan penulis, skripsi ini dapat menjadi manfaat untuk pembaca dan segala pihak.

Malang, Juni 2018

Penulis
silvia.lubis1@gmail.com



ABSTRAK

Gangguan kejiwaan adalah suatu gangguan pada otak manusia yang tidak normal atau berbeda dari orang pada umumnya. Pada gangguan kejiwaan terdapat berbagai macam jenis penyakit di dalamnya. Skizofrenia dan Depresi merupakan jenis gangguan kejiwaan yang banyak diderita oleh masyarakat. Terdapat juga jenis-jenis penyakit Skizofrenia dan Depresi, salah satu jenis pada masing-masing penyakit adalah Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik. Menurut data yang ada di Rumah Sakit Jiwa Dr. Radjiman Wediodiningrat Lawang, kedua penyakit ini masuk dalam 10 besar diagnosis penyakit pasien rawat jalan dan pasien rawat jalan IGD periode tahun 2017. Dikarenakan banyaknya pasien yang terserang penyakit tersebut, maka dibutuhkan suatu sistem yang dapat mengklasifikasikan penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik. Klasifikasi merupakan pembuatan suatu model yang digunakan untuk mengelompokkan suatu objek yang memiliki ciri-ciri yang sama ke dalam suatu kelas yang telah ditentukan. Dalam klasifikasi penyakit ini, digunakan algoritme klasifikasi yaitu *support vector machine* (SVM) dengan kernel *polynomial of degree 2*. Data yang digunakan sebanyak 200 data yang diambil dari RSJ. Dr. Radjiman Wediodiningrat Lawang, data ini terdiri dari 80% data latih dan 20% data uji. Metode pengujian yang digunakan adalah dengan *K-Fold Cross Validation*. Berdasarkan hasil pengujian parameter SVM didapat nilai rata-rata akurasi terbaik sebesar 79% dengan nilai $\gamma = 0,00001$, $\lambda = 0,1$, $C = 0,01$, $itermax = 150$, dan $\epsilon = 1.10^{-10}$.

Kata kunci: *Gangguan Kejiwaan, Skizofrenia Hebefrenik, Episode Depresi Berat dengan Gejala Psikotik, Klasifikasi, Support Vector Machine*

ABSTRACT

Psychiatric disorders are a disorders of the human brain that is not normal or different from people in general. There are many types of psychiatric disorders. Schizophrenia and Depression are a type of psychiatric disorders suffered by many people. There are also types of Schizophrenia and Depression, one type of disease in each is Schizophrenia Hebephrenic and Psychotic Depression. According to data in soul hospital of Dr. Radjiman Wediodiningrat Lawang, both of these diseases are include in the top 10 diagnoses of outpatient and outpatient illnesses in 2017. Due to the large number of patients affected by the didease, soul hospital needed a system that can classify Schizophrenia Hebephrenic and Psychotic Depression Disease. Classification is the manufacture of a model that used to make a group for an object with the same characteristics into a determined class. To classify the disease used supprot vector machine (SVM) algorithm with polynomial of degree 2 kernel. The data used are 200 data taken from soul hospital of Dr. Radjiman Wediodiningrat Lawang. This data consists of 80% data training and 20% data testing. The test method used is K-fold cross validation. Based on the results of testing SVM parameters obtained the highest average accuracy is 79% with the value of $\gamma = 0,00001$, $\lambda = 0,1$, $C = 0,01$, $itermax = 150$, and $\epsilon = 1.10^{-10}$.

Keywords: Psychiatric Disorders, Schizophrenia Hebephrenic, Psychotic Depression, Classification, Supprot Vector Machine

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR KODE PROGRAM	xi
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
BAB 1 PENDAHULUAN.....	1
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
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka	6
2.2 Gangguan Kejiwaan.....	10
2.2.1 Skizofrenia.....	10
2.2.2 Episode Depresif	12
2.3 <i>Data Mining</i>	13
2.4 Klasifikasi	14
2.5 <i>Support Vector Machine (SVM)</i>	14
2.5.1 SVM Non-Linear	15
2.5.2 <i>Sequential Training</i>	16
2.5.3 <i>Testing SVM</i>	17
2.6 Evaluasi Hasil	18
2.7 <i>K-Fold Cross Validation</i>	19
BAB 3 METODOLOGI	20
3.1 Studi Literatur	20



3.2 Pengumpulan Data.....	20
3.3 Perancangan Sistem.....	21
3.4 Implementasi Sistem.....	21
3.5 Pengujian dan Analisis.....	21
3.6 Kesimpulan.....	21
BAB 4 PERANCANGAN.....	22
4.1 Formulasi Permasalahan.....	22
4.2 Alur Proses Algoritme <i>Support Vector Machine</i> (SVM).....	23
4.2.2 Alur Proses Perhitungan Kernel.....	24
4.2.3 Alur Proses Perhitungan <i>Sequential Training SVM</i>	25
4.2.4 Alur Proses Perhitungan Matriks <i>Hessian</i>	26
4.2.5 Alur Proses Perhitungan E_i	27
4.2.6 Alur Proses Perhitungan δ_{ai}	28
4.2.7 Alur Proses Perhitungan α_i baru.....	29
4.2.8 Alur Proses Perhitungan Nilai wx^+	30
4.2.9 Alur Proses Perhitungan Nilai wx^-	31
4.2.10 Alur Proses Perhitungan Nilai Bias.....	32
4.2.11 Alur Proses <i>Testing SVM</i>	32
4.3 Proses Perhitungan Manualisasi.....	34
4.3.1 Perhitungan Manual Kernel.....	35
4.3.2 Perhitungan Manual <i>Sequential Training</i>	36
4.3.3 Perhitungan Manual Nilai w dan b	39
4.3.4 Perhitungan Manual <i>Testing SVM</i>	41
4.3.5 Perhitungan Manual Tingkat Akurasi.....	44
4.4 Perancangan Pengujian Sistem.....	45
4.4.1 Perancangan Pengujian Terhadap Variabel λ (<i>Lambda</i>).....	45
4.4.2 Perancangan Pengujian Terhadap Variabel γ (<i>Gamma</i>).....	45
4.4.3 Perancangan Pengujian Terhadap Variabel C (<i>Complexity</i>).....	46
4.4.4 Perancangan Pengujian Terhadap Jumlah <i>Itermax</i>	46
4.5 Evaluasi Hasil.....	47
BAB 5 IMPLEMENTASI.....	48
5.1 Spesifikasi Sistem.....	48
5.1.1 Spesifikasi Perangkat Keras.....	48
5.1.2 Spesifikasi Perangkat Lunak.....	48



5.2 Implementasi Algoritme.....	49
5.2.1 Implementasi Algoritme Perhitungan Kernel SVM.....	49
5.2.2 Implementasi Algoritme Perhitungan Matriks <i>Hessian</i>	50
5.2.3 Implementasi Algoritme Perhitungan Nilai E_i	50
5.2.4 Implementasi Algoritme Perhitungan Nilai δ_{ai}	51
5.2.5 Implementasi Algoritme Perhitungan Nilai α_i	52
5.2.6 Implementasi Algoritme Perhitungan Nilai wx^+ dan wx^-	53
5.2.7 Implementasi Algoritme Perhitungan Nilai Bias.....	54
5.2.8 Implementasi Algoritme Perhitungan Nilai Kernel <i>Test</i>	54
5.2.9 Implementasi Algoritme Perhitungan Nilai $f(x)$	55
5.3 Implementasi Hasil.....	56
5.3.1 Implementasi Hasil Data.....	56
5.3.2 Implementasi Hasil <i>Sequential Training</i>	56
5.3.3 Implementasi Hasil <i>Testing SVM</i>	58
5.3.4 Implementasi Hasil Klasifikasi.....	58
BAB 6 PENGUJIAN DAN ANALISIS.....	59
6.1 Pengujian Terhadap Variabel λ (<i>Lambda</i>).....	59
6.1.1 Skenario Pengujian Terhadap Variabel λ (<i>Lambda</i>).....	59
6.1.2 Analisis Hasil Pengujian Terhadap Variabel λ (<i>Lambda</i>).....	61
6.2 Pengujian Terhadap Variabel γ (<i>Gamma</i>).....	61
6.2.1 Skenario Pengujian Terhadap Variabel γ (<i>Gamma</i>).....	61
6.2.2 Analisis Hasil Pengujian Terhadap Variabel γ (<i>Gamma</i>).....	63
6.3 Pengujian Terhadap Variabel C (<i>Complexity</i>).....	64
6.3.1 Skenario Pengujian Terhadap Variabel C (<i>Complexity</i>).....	64
6.3.2 Analisis Hasil Pengujian Terhadap Variabel C (<i>Complexity</i>).....	66
6.4 Pengujian Terhadap <i>Itermax</i>	67
6.4.1 Skenario Pengujian Terhadap Jumlah <i>Itermax</i>	67
6.4.2 Analisis Hasil Pengujian Terhadap Jumlah <i>Itermax</i>	68
BAB 7 PENUTUP.....	70
7.1 Kesimpulan.....	70
7.2 Saran.....	71
DAFTAR PUSTAKA.....	72
DAFTAR LAMPIRAN.....	74

DAFTAR KODE PROGRAM

Kode Program 5.1 Implementasi Kernel Polynomial of Degree 2	49
Kode Program 5.2 Implementasi Matriks Hessian.....	50
Kode Program 5.3 Implementasi Nilai E_i	51
Kode Program 5.4 Implementasi Nilai $\delta\alpha_i$	52
Kode Program 5.5 Implementasi Nilai α_i	52
Kode Program 5.6 Implementasi Nilai w_{x+} dan w_{x-}	53
Kode Program 5.7 Implementasi Nilai Bias.....	54
Kode Program 5.8 Implementasi Nilai Kernel Test	55
Kode Program 5.9 Implementasi Nilai $f(x)$	55



DAFTAR GAMBAR

Gambar 2.1 Gambaran Support Vector Machine dengan Hyperplane Terbaik ... 14

Gambar 2.2 Suport Vector Machine Non-Linear 16

Gambar 3.1 Tahapan Penelitian..... 20

Gambar 4.1 Diagram Alir Algoritme Support Vector Machine 23

Gambar 4.2 Diagram Alir Proses Perhitungan Kernel SVM 24

Gambar 4.3 Diagram Alir Proses Perhitungan Training SVM..... 25

Gambar 4.4 Diagram Alir Proses Perhitungan Matriks Hessian..... 26

Gambar 4.5 Diagram Alir Proses Perhitungan E_i 27

Gambar 4.6 Diagram Alir Proses Perhitungan Nilai $\delta\alpha_i$ 28

Gambar 4.7 Diagram Alir Proses Perhitugan α_i 29

Gambar 4.8 Diagram Alir Proses Perhitungan Nilai wx^+ 30

Gambar 4.9 Diagram Alir Proses Perhitungan Nilai wx^- 31

Gambar 4.10 Diagram Alir Proses Perhitungan Nilai Bias..... 32

Gambar 4.11 Diagram Alir Proses Testing SVM 33

Gambar 5.1 Implementasi Hasil Data 56

Gambar 5.2 Implementasi Hasil Kernel dan Matriks Hessian..... 57

Gambar 5.3 Implementasi Hasil Sequential Training Iterasi 1..... 57

Gambar 5.4 Implementasi Hasil Sequential Training Iterasi 2..... 57

Gambar 5.5 Implementasi Hasil Nilai wx^+ , wx^- , dan Bias 58

Gambar 5.6 Implementasi Hasil Testing SVM..... 58

Gambar 5.7 Implementasi Hasil Hasil Klasifikasi 58

Gambar 6.1 Grafik Hasil Pengujian Terhadap Variabel λ (Lambda)..... 61

Gambar 6.2 Grafik Hasil Pengujian Terhadap Variabel γ (Gamma)..... 63

Gambar 6.3 Grafik Pengujian Terhadap Variabel C (Complexity)..... 66

Gambar 6.4 Pengujian Terhadap Jumlah Itermax 69



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka dari Penelitian Sebelumnya	8
Tabel 2.2 Kernel yang umum digunakan pada SVM	16
Tabel 2.3 Model Confusion Matrix.....	18
Tabel 4.1 Tabel Data Latih dan Data Uji.....	22
Tabel 4.2 Data Latih	34
Tabel 4.3 Data Uji	35
Tabel 4.4 Hasil Matriks Kernel Polynomial of Degree 2.....	35
Tabel 4.5 Hasil Perhitungan Manual Matriks Hessian	36
Tabel 4.6 Hasil Perhitungan Manual E_i Iterasi 1	37
Tabel 4.7 Hasil Perhitungan Manual $\delta\alpha_i$ Iterasi 1.....	38
Tabel 4.8 Hasil Perhitungan Manual α_i Iterasi 1	38
Tabel 4.9 Hasil Perhitungan Manual E_i Iterasi 2	38
Tabel 4.10 Hasil Perhitungan Manual $\delta\alpha_i$ Iterasi 2	39
Tabel 4.11 Hasil Perhitungan Manual α_i Iterasi 2	39
Tabel 4.12 Hasil Kernel $K(x_i, x_+)$ dan $K(x_i, x_-)$	40
Tabel 4.13 Hasil Perhitungan Nilai w_{x+} dan w_{x-}	41
Tabel 4.14 Hasil Perhitungan Kernel Testing	42
Tabel 4.15 Hasil Perhitungan $\alpha_i y_i K(x_i, x)$	43
Tabel 4.16 Tabel Confusion Matrix	44
Tabel 4.17 Perancangan Pengujian Terhadap Variabel λ (Lambda)	45
Tabel 4.18 Perancangan Pengujian Terhadap Variabel γ (Gamma).....	45
Tabel 4.19 Perancangan Pengujian Terhadap Variabel C (Complexity).....	46
Tabel 4.20 Perancangan Pengujian Terhadap Itermax	46
Tabel 5.1 Spesifikasi Perangkat Keras	48
Tabel 5.2 Spesifikasi Perangkat Lunak	48
Tabel 6.1 Hasil Pengujian Terhadap Variabel λ (Lambda).....	59
Tabel 6.2 Hasil Pengujian Terhadap Variabel γ (Gamma).....	62
Tabel 6.3 Pengujian Terhadap Variabel C (Complexity).....	64
Tabel 6.4 Hasil Pengujian Terhadap Jumlah Itermax	67



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Gangguan kejiwaan merupakan suatu gangguan yang terjadi di dalam otak manusia yang tidak normal atau tidak seperti pada umumnya. Gangguan ini bisa disebabkan oleh faktor keturunan ataupun faktor eksternal lainnya. Orang yang mengalami penyakit psikologis tampak sehat dari luar namun pada dalam dirinya terdapat berbagai macam tekanan yang membuat orang tersebut tidak menjalani harinya dengan normal baik secara individual maupun secara social (Suhaimi, 2015). Hasil Survei Kesehatan Rumah Tangga (SKRT) dalam yang dilakukan oleh Badan Penelitian dan Pengembangan Departemen Kesehatan pada tahun 1995 menunjukkan terdapat 140 per 1.000 anggota rumah tangga dengan gangguan mental remaja dan dewasa serta 104 per 1.000 anggota rumah tangga dengan gangguan mental anak usia sekolah (Suhaimi, 2015). Gangguan kejiwaan ini memiliki banyak jenis atau macamnya, beberapa jenis dari gangguan kejiwaan ini adalah penyakit Skizofrenia dan Episode Depresi.

Menurut Kementerian Kesehatan Republik Indonesia dengan data Riskesdas 2013 menunjukkan bahwa gangguan mental dengan gejala depresi dan kecemasan pada manusia dengan usia 15 tahun ke atas mencapai kurang lebih 14 juta orang dan gangguan jiwa berat seperti skizofrenia mencapai kurang lebih 400 ribu orang (Kementerian Kesehatan Republik Indonesia, 2016). Skizofrenia adalah sebuah kelainan yang ditandai dengan banyak gejala yaitu gejala positif (delusi dan halusinasi), gejala negatif (apati, penarikan, kekurangan pemikiran, dan pembatasan efek), gejala disorganisasi (pikiran dan tindakannya tidak terorganisir) dan gejala kognitif (memori, perhatian, memori kerja, pemecahan masalah, kecepatan memproses dan kognisi sosial) (Lu, et al., 2016). Sedangkan depresi adalah suatu gangguan kesehatan mental yang memperngaruhi pola pikir, perasaan, suasana hati dan cara menghadapi aktivitas sehari-hari yang terjadi selama dua minggu atau lebih. Orang yang mengalami depresi akan merasa sedih yang berkepanjangan, putus asa, tidak memiliki motivasi dalam diri, dan selalu menyalahkan diri sendiri. Gejala umum yang biasa dialami oleh penderita penyakit depresi adalah afek depresif, tidak ada minat dan kegembiraan, energi yang berkurang sehingga menyebabkan pasien mudah lelah dan membuat aktifitasnya menurun (Maslim, 2013). Skizofrenia dan Episode Depresi memiliki banyak jenis-jenisnya tersendiri. Skizofrenia memiliki jenis-jenis seperti skizofrenia paranoid, hebefrenik, katatonik, tak terinci, pasca-skizofrenia, residual, simpleks, skizofrenia lainnya, dan skizofrenia YTT. Sedangkan untuk episode depresi terbagi atas episode depresi ringan, sedang, berat tanpa gejala psikotik, berat dengan gejala psikotik, episode depresi lainnya, dan episode depresi YTT.

Pada penelitian ini akan berfokus pada klasifikasi dua jenis gangguan kejiwaan yaitu Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik. Menurut data yang ada di Rumah Sakit Jiwa Dr. Radjiman Wediodiningrat Lawang, kedua penyakit ini masuk dalam 10 besar diagnosis penyakit pasien rawat jalan

dan pasien rawat jalan IGD pada periode tahun 2017. Skizofrenia hebefrenik merupakan penyakit dengan peringkat pertama pada diagnosis penyakit pasien rawat jalan, rawat jalan IGD, dan rawat inap dengan total pasien masing-masing sebanyak 18.048, 1.857, dan 2231 orang dalam periode tahun 2017. Kemudian untuk penyakit episode depresi berat dengan gejala psikotik berada pada peringkat ke-8 untuk pasien rawat jalan dan peringkat ke-10 untuk pasien rawat jalan IGD dengan total pasien masing-masing sebanyak 449 dan 59 orang dalam periode tahun 2017. Salah satu dokter yang ada di Rumah Sakit Jiwa Dr. Radjiman Wediodiningrat Lawang yaitu dokter Nur Aida, dr.SpKJ mengatakan bahwa untuk penyakit depresi memang sangat jarang ditemukan masuk dalam pasien rawat inap dikarenakan orang dengan penyakit ini masih sadar akan dirinya sendiri. Dikarenakan jumlah pasien yang sangat besar inilah akan sulit jika mengklasifikasikan data pasien secara manual. Maka dari itu dibuatlah sistem yang dapat memudahkan dunia kedokteran jiwa dalam mengklasifikasi penyakit ini.

Penelitian yang berjudul "*Classification of Schizophrenia and Depression by EEG with ANNs**" dilakukan oleh Ying-jie Li dan Fei-yan Fan (2005). Dalam penelitian ini penulis mengklasifikasikan pasien yang menderita Skizofrenia, Depresi dan orang yang sehat dengan menggunakan Jaringan Syaraf Tiruan. Terdapat dua pendekatan Jaringan Syaraf Tiruan yang digunakan pada penelitian ini yaitu pendekatan BP ANN dan Self-organizing competitive Network ANN, kemudian menggunakan *Electroencephalography (EEG)* sebagai vektor fitur. Output target yang digunakan pada penelitian ini adalah -1 untuk orang sehat, 0 untuk penderita Skizofrenia, dan 1 untuk penderita Depresi. Setiap nilai dari output antara 0.4 dan 0.6 (atau antara -0.4 dan -0.6) digunakan untuk pola uji dengan kasus sementara nilai output antara -0.4 dan 0,4 untuk penderita Skizofrenia, nilai output di atas 0.6 untuk penderita Depresi, dan nilai output di bawah -0.6 untuk orang yang sehat. Hasil dari penelitian ini menunjukkan data yang dilatih menggunakan BP ANN dapat mengidentifikasi pasien dengan penyakit Skizofrenia, Depresi, dan orang sehat dengan benar yang memiliki akurasi keseluruhan 60%~80%. Sedangkan jika menggunakan Self-organizing competitive Network didapatkan hasil akurasi keseluruhan 40%~60%.

Penelitian yang dilakukan oleh Ana Mariyam Puspitasari, Dian Eka Ratnawati, dan Agus Wahyu Widodo (2018), bertujuan untuk mengklasifikasikan penyakit gigi dan mulut menggunakan metode *Support Vector Machine (SVM)*. Terdapat empat kelas hasil dari klasifikasi ini yaitu kelas pulpitis, gingivitis, nekrosis pulpa dan periodontitis dengan jumlah parameter sebanyak 16. Pada penelitian ini menggunakan 122 dataset yang terdiri dari data training dan data testing. Dataset tersebut akan dibagi sesuai dengan perbandingan rasio 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%, 40%:60%, 30%:70%, 20%:80%, 10%:90% yang digunakan untuk melihat hasil akurasi yang terbaik. Pengujian yang dilakukan menggunakan *kernel RBF* dan nilai parameter *sequential training*. Hasil dari penelitian ini menunjukkan bahwa penggunaan metode *Support Vector Machine (SVM)* dalam pengklasifikasian penyakit gigi dan mulut mendapatkan hasil terbaik dengan akurasi rata-rata 94.442%. Kemudian pada pengujian kernel RBF yang paling optimal pada nilai akurasi rata-rata 93.329%.

Pada penelitian ini digunakan metode *Support Vector Machine* (SVM). Metode *Support Vector Machine* (SVM) merupakan metode dengan kelas *supervised learning* yang dapat digunakan untuk pengklasifikasian maupun regresi. Metode SVM ini digunakan untuk mencari pembatas (*hyperplane*) yang paling optimal antara dua set data dari dua kelas yang berbeda dalam ruang vektor. Pada SVM juga terdapat proses training data yang digunakan untuk mencari posisi paling optimal dari *hyperplane* di ruang vektor. Pada dasarnya SVM merupakan *linear classifier*, namun SVM juga dapat digunakan dalam permasalahan non-linear dengan memasukkan fungsi kernel di dalamnya. Pada SVM non-linear, data diproyeksikan ke ruang vektor baru berdimensi tinggi hingga data tersebut dapat dipisahkan secara linear.

Metode *Support Vector Machine* (SVM) digunakan dalam penelitian ini karena metode SVM ini merupakan metode klasifikasi dengan waktu komputasi yang cepat. Beberapa penelitian menunjukkan bahwa metode SVM unggul digunakan dalam sistem klasifikasi. Menurut penelitian Rustam, et al (2003) dalam Octaviani, et al (2014) menyatakan bahwa metode SVM lebih unggul dibandingkan dengan metode *K-Nearest Neighbor* (KNN) dalam klasifikasi data aroma yaitu SVM memiliki tingkat akurasi 100%. Kemudian pada penelitian yang dilakukan oleh Rachman dan Purnami (2012) dalam klasifikasi tingkat keganasan *breast cancer* menggunakan metode Regresi Logistik Ordinal dan SVM didapatkan hasil ketepatan klasifikasi pada kedua metode ini adalah masing-masing sebesar 56,60% dan 98,11% dengan kernel SVM yang digunakan adalah RBF dan *polynomial*.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan sistem klasifikasikan penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik menggunakan algoritme *Support Vector Machine* (SVM) ?
2. Bagaimana tingkat akurasi yang didapat pada sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik menggunakan algoritme *Support Vector Machine* (SVM) ?

1.3 Tujuan

Berdasarkan pada rumusan masalah yang telah dikemukakan, maka dapat dirumuskan tujuan penelitian sebagai berikut:

1. Mengimplementasikan sistem klasifikasikan penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik menggunakan algoritme *Support Vector Machine* (SVM).
2. Menguji tingkat akurasi yang didapat pada sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik menggunakan algoritme *Support Vector Machine* (SVM).

1.4 Manfaat

Adapun manfaat dari penelitian ini adalah agar dapat membantu tenaga medis dalam melakukan klasifikasi penyakit kejiwaan Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik.

1.5 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian ini meliputi:

1. Kelas klasifikasi yang digunakan adalah 2 kelas yaitu kelas penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik.
2. Data yang digunakan adalah data rekam medik pasien yang terserang penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik, data tersebut didapatkan dari Rumah Sakit Jiwa Dr. Radjiman Wediodiningrat Lawang.
3. Data yang digunakan sebanyak 200 data pasien dengan 100 data pasien Skizofrenia Hebefrenik dan 100 data pasien Episode Depresi Berat dengan Gejala Psikotik.
4. Fitur yang digunakan adalah gejala dari penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik dengan fitur sebanyak 70 gejala.
5. Keluaran yang dihasilkan dari sistem berupa penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik sesuai dengan gejala yang dimiliki oleh pasien.
6. Kernel SVM yang digunakan adalah fungsi kernel *Polynomial of degree 2*.
7. Bahasa pemrograman yang digunakan adalah Java dengan basis data menggunakan Microsoft Excel.

1.6 Sistematika Pembahasan

Sistematika penulisan skripsi ini secara garis besar akan diuraikan sebagai berikut:

BAB 1: PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang masalah yang diangkat, solusi yang ditawarkan, tujuan dan manfaat, metode yang digunakan, dan sistematika pembahasan.

BAB 2: LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang teori-teori yang relevan dengan variabel yang akan diteliti sebagai dasar untuk menjawab sementara rumusan masalah pada penelitian.

BAB 3: METODOLOGI

Bab ini menjelaskan tentang metodologi yang akan digunakan dalam penelitian ini, seperti langkah-langkah yang dilakukan dalam proses penelitian.

BAB 4: PERANCANGAN

Bab ini menjelaskan perancangan sistem yang akan dibuat seperti alur sistem, manualisasi sistem, perancangan hasil, perancangan pengujian.

BAB 5: IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem yang telah dirancang agar dapat menjadi sebuah sistem seperti yang diinginkan. Bab ini berisi spesifikasi sistem, implementasi algoritme serta implementasi hasil.

BAB 6: PENGUJIAN DAN ANALISIS

Bab ini menjelaskan tentang pengujian sistem yang telah dibuat dan menganalisis hasil pengujian yang telah dilakukan.

BAB 7: PENUTUP

Bab ini menjelaskan tentang kesimpulan dari penelitian yang telah dilakukan serta saran untuk penelitian yang selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada penelitian ini terdapat penelitian-penelitian sebelumnya yang telah dilakukan dan menjadi acuan atau referensi peneliti dalam penelitian ini. Penelitian pertama yang menjadi referensi dalam penulisan penelitian ini berjudul "*Classification of Schizophrenia and Depression by EEG with ANNs**" yang ditulis oleh Ying-jie Li dan Fei-yan Fan (2005). Dalam penelitian ini membahas tentang pengklasifikasian seseorang yang menderita penyakit Skizofrenia, Depresi dan orang yang sehat dengan menggunakan pendekatan Jaringan Syaraf Tiruan (JST). Dalam penelitian ini juga membahas tentang perbandingan antara dua pendekatan Jaringan Syaraf Tiruan (JST) dalam menyelesaikan permasalahan tiga jenis subjek tersebut. Pendekatan yang digunakan adalah pendekatan BP ANN dan Self-organizing competitive Network ANN, kemudian menggunakan *Electroencephalography (EEG)* sebagai vektor fitur. Output target yang digunakan pada penelitian ini adalah -1 untuk orang sehat, 0 untuk penderita Skizofrenia, dan 1 untuk penderita Depresi. Setiap nilai dari output antara 0.4 dan 0.6 (atau antara -0.4 dan -0.6) digunakan untuk pola uji dengan kasus sementara nilai output antara -0.4 dan 0,4 untuk penderita Skizofrenia, nilai output di atas 0.6 untuk penderita Depresi, dan nilai output di bawah -0.6 untuk orang yang sehat. Data yang digunakan adalah 30 subyek data yang akan dibagi menjadi 15 subyek data untuk data training dan 15 subyek data untuk data testing. Hasil dari penelitian ini menunjukkan data yang dilatih menggunakan BP ANN dapat mengidentifikasi pasien dengan penyakit Skizofrenia, Depresi, dan orang sehat dengan benar yang memiliki akurasi keseluruhan 60% ~ 80%. Sedangkan jika menggunakan Self-organizing competitive Network didapatkan hasil akurasi keseluruhan 40%~60%.

Penelitian kedua yang menjadi referensi dalam penelitian ini berjudul "*Clasification of Schizophrenia using Genetic Algorithm-Support Vector Machine (GA-SVM)*" yang ditulis oleh Ming-Hsien Hiesh, et al. (2013). Pada penelitian ini penulis melakukan klasifikasi penyakit Skizofrenia dengan membuat sistem yang dibangun menggunakan *Wavelet Transform (WT)*, *Genetic Algorithm (GA)*, *Support Vector Machine (SVM)* untuk menganalisa uji sinkronisasi band Gamma data. Terdapat empat bagian dalam mengklasifikasikan penyakit Skizofrenia yang digunakan pada penelitian ini yaitu *pre-processing data* dengan menggunakan *low pass filter*, ekstraksi fitur dengan entropi dan nilai statistik, seleksi fitur menggunakan *Genetic Algorithm (GA)*, dan klasifikasi penyakit menggunakan *Support Vector Machine (SVM)*. Peneliti mengumpulkan 5 kontrol dan 5 pasien penyakit Skizofrenia dari Rumah Sakit National Taiwan University yang menjalani sinkronisasi band Gamma dengan memberi 20Hz, 30Hz, dan 40Hz stimulasi pendengaran selama 10 menit. Hasil dari penelitian ini menunjukkan sistem melakukan 89.48% sensitivitas, 87% spesifisitas dan akurasi pada kelompok kontrol serta kelompok masing-masing pasien sebesar 87% dan 89.48%. Didapatkanlah hasil akurasi dari klasifikasi sebesar 88.24% dengan semua fitur yang memiliki berat lebih dari 50%.

Penelitian ketiga yang menjadi referensi dalam penulisan penelitian ini ditulis oleh Ana Mariyam Puspitasari, Dian Eka Ratnawati, dan Agus Wahyu Widodo (2018) dengan judul “Klasifikasi Penyakit Gigi dan Mulut Menggunakan Metode *Support Vector Machine*”. Dalam penelitian ini membahas tentang pengklasifikasian penyakit gigi dan mulut sejak dini dengan menggunakan metode *Support Vector Machine* sehingga dapat diketahui jenis penyakit gigi dan mulut yang diderita oleh pasien secara cepat dan mudah. Terdapat empat kelas hasil dari klasifikasi ini yaitu kelas pulpitis, gingivitis, nekrosis pulpa dan periodontitis dengan jumlah parameter sebanyak 16. Pada penelitian ini menggunakan 122 dataset yang terdiri dari data training dan data testing. Dataset tersebut akan dibagi sesuai dengan perbandingan rasio 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%, 40%:60%, 30%:70%, 20%:80%, 10%:90% yang digunakan untuk melihat hasil akurasi yang terbaik. Pengujian yang dilakukan menggunakan *kernel* RBF dan nilai parameter *sequential training*. Hasil dari penelitian ini menunjukkan bahwa penggunaan metode *Support Vector Machine* (SVM) dalam pengklasifikasian penyakit gigi dan mulut mendapatkan hasil terbaik menggunakan *sequential training* dengan akurasi rata-rata 94.442% yang memiliki nilai λ (Lambda) = 0.1, γ (gamma) = 0.1, C (Complexity) = 1, ϵ (epsilon) = 1.10^{-10} , *itermax* = 50, dan rasio data 80%:20%. Kemudian pada pengujian kernel RBF yang paling optimal pada nilai akurasi rata-rata 93.329%. Sehingga dapat disimpulkan bahwa pengujian dengan kernel RBF lebih cocok dalam permasalahan ini dibandingkan dengan kernel linear atau polynomial karena kernel RBF mampu memrediksi kelas pada data tersebut.

Penelitian keempat yang mejadi referensi dalam penelitian ini berjudul “Klasifikasi Gangguan Jiwa Skizofrenia Menggunakan Algoritme *Support Vector Machine* (SVM)” yang ditulis oleh Daisy Kurniawaty, Imam Cholissodin, dan Putra Pandu Adikara. Penelitian ini merupakan penelitian yang menjadi acuan utama dalam pembuatan sistem Klasifikasi Penyakit Skizofrenia Hebefreni dan Episode Depresi Berat dengan Gejala Psikotik. Pada penelitian ini membahas tentang pengklasifikasian gangguan kejiwaan Skizofrenia menggunakan metode *Support Vector Machine* (SVM) dengan dua kelas yang digunakan yaitu kelas Skizofrenia *Paranoid* dan *Simplex*. Data yang digunakan merupakan data rekam medik pasien pada RSUD Ngawi, dengan pemilihan data latih dan uji secara acak sesuai dengan rasio perbandingan yang ditentukan. Pengujian yang digunakan adalah pengujian terhadap parameter-parameter SVM seperti rasio perbandingan data, fungsi kernel, nilai *gamma*, nilai *lambda*, nilai *complexity*, dan jumlah *itermax*. Berdasarkan hasil pengujian yang dilakukan pada penelitian ini, didapatkan hasil akurasi terbaik adalah sebesar 100%. Dengan akurasi 100% tersebut digunakan rasio perbandingan data sebesar 90%:10%, jenis kernel *polynomial of degree 2*, nilai *gamma* = 0,00001, nilai *lambda* = 3, nilai C = 0,01, dan jumlah *itermax* = 1000. Sehingga dapat disimpulkan bahwa metode *support vector machine* dengan menggunakan kernel *polynomial of degree 2* sangat cocok digunakan pada sistem klasifikasi penyakit gangguan jiwa Skizofrenia *Paranoid* dan *Simplex*.

Penelitian kelima yang menjadi referensi penulis berjudul “Analisis Sentimen *Cyberbullying* pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine” yang ditulis oleh Wanda Athira Luqyana, Imam Cholissodin, dan

Rizal Setya Perdana. Penelitian ini merupakan referensi penulis dalam memilih *degree* yang pada penelitian ini yaitu $d = 2$. Pada penelitian ini membahas tentang klasifikasi komentar *cyberbullying* pada Instagram. Digunakan fitur *Term Frequency-Inverse Document Frequency* (TF-IDF) untuk mengetahui setiap sentimen pada komentar dan menggunakan metode *Support Vector Machine* sebagai metode klasifikasi. Kelas yang digunakan adalah kelas sentimen positif *cyberbullying* dan sentimen negatif *cyberbullying*. Data yang digunakan adalah data yang diambil secara luring (*offline*) dengan banyak data sebesar 400 data dan 1799 fitur. Berdasarkan pengujian yang dilakukan, didapatkan parameter SVM terbaik yang digunakan adalah nilai *degree* pada kernel *polynomial* sebesar 2, nilai *learning rate* sebesar 0,0001, dan jumlah iterasi maksimum sebesar 200 kali. Perbandingan data yang digunakan adalah sebanyak 50%:50%, yaitu 50% data latih dan 50% data uji tanpa mengimplementasikan algoritme *Lexion Based Features*. Nilai akurasi tertinggi yang didapatkan sistem adalah sebesar 90% dengan nilai *precision* sebesar 94,44%, *recall* sebesar 85%, dan *f-measure* sebesar 89,47%.

Kajian pustaka dari penelitian sebelumnya yang menjadi referensi penulis dapat dilihat pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka dari Penelitian Sebelumnya

No.	Peneliti	Objek	Motode	Hasil
1.	(Li & Fan, 2005)	- Objek data penyakit Skizofrenia dan Depresi. - Masukan berupa gejala penyakit.	BP ANN dan Self-organizing competitive Network ANN	BP ANN dapat mengidentifikasi pasien dengan penyakit Skizofrenia, Depresi, dan orang sehat dengan benar yang memiliki akurasi keseluruhan 60% ~ 80% dan menggunakan Self-organizing competitive Network didapatkan hasil akurasi keseluruhan 40%~60%.
2.	(Hiesh, et al., 2013)	- Objek data pasien yang menderita penyakit Skizofrenia yang menjalani	<i>Genetic Algorithm</i> (GA) dan <i>Support Vector Machine</i> (SVM)	Hasil dari penelitian ini menunjukkan sistem melakukan 89.48% sesitivitas, 87% spesifisitas dan akurasi pada kelompok kontrol serta



		<p>sinkronisasi band Gamma.</p> <ul style="list-style-type: none"> - Setiap pasien diberikan 20Hz, 30Hz, dan 40Hz stimulasi pendengaran selama 10 menit. 		<p>kelompok masing-masing pasien sebesar 87% dan 89.48%. Didapatkanlah hasil akurasi dari klasifikasi sebesar 88.24% dengan semua fitur yang memiliki berat lebih dari 50%.</p>
3.	(Puspitasari, Ratnawati, & Widodo, 2018)	<ul style="list-style-type: none"> - Objek data penyakit gigi dan mulut. 	Support Vector Machine (SVM)	Support Vector Machine (SVM) dalam pengklasifikasian penyakit gigi dan mulut mendapatkan hasil terbaik menggunakan <i>sequential training</i> dengan akurasi rata-rata 94.442% dan pada pengujian kernel RBF yang paling optimal pada nilai akurasi rata-rata 93.329%.
4.	(Kurniawaty, Cholissodin, & Adikara, 2018)	<ul style="list-style-type: none"> - Objek data penyakit Skizofrenia <i>Paranoid</i> & <i>Simplex</i> 	Support Vector Machine (SVM)	Didapatkan akurasi terbaik adalah sebesar 100% dengan rasio perbandingan data sebesar 90%:10%, jenis kernel <i>polynomial of degree 2</i> , nilai <i>gamma</i> = 0,00001, nilai <i>lambda</i> = 3, nilai <i>C</i> = 0,01, dan jumlah <i>itermax</i> = 1000.
5.	(Luqyana, Cholissodin, & Perdana)	<ul style="list-style-type: none"> - Objek data merupakan komentar yang ada pada Instagram - Kelas yang digunakan adalah kelas 	Term Frequency-Inverse Document Frequency (TF-IDF) dan Support Vector	Akurasi terbaik sebesar 90% dengan nilai <i>precision</i> sebesar 94,44%, <i>recall</i> sebesar 85%, dan <i>f-measure</i> sebesar 89,47%. Parameter SVM terbaik yang digunakan adalah nilai



		sentimen positif <i>cyberbullying</i> dan sentimen negatif <i>cyberbullying</i>	Machine (SVM)	<i>degree</i> pada kernel <i>polynomial</i> sebesar 2, nilai <i>learning rate</i> sebesar 0,0001, dan jumlah iterasi maksimum sebesar 200 kali. Komposisi data yang digunakan adalah 50% data latih dan 50% data uji tanpa mengimplementasikan algoritme <i>Lexion Based Features</i> .
--	--	---	---------------	---

2.2 Gangguan Kejiwaan

Menurut *the Diagnostic and Statistical Manual of Mental Disorder (DSM)-IV*, gangguan jiwa merupakan sindrom psikologis yang ada pada seseorang dan diasosiasikan memiliki resiko mengalami kematian, penderitaan, disabilitas, atau kehilangan kebebasan yang meningkat secara signifikan (Suhaimi, 2015). Gangguan jiwa ini dapat menimpa siapapun yang dikarenakan oleh berbagai faktor. Dari berbagai macam pengertian tentang gangguan jiwa, didapatkan konsep gangguan jiwa berupa adanya gejala klinis yang bermakna. Gejala klinis tersebut berupa pola perilaku dan pola psikologik yang menimbulkan *distress* dan *disability*. *Distress* yang dialami berupa rasa tidak nyaman, nyeri, terganggu, organ tubuh yang disfungsi, dan sebagainya. Sedangkan *disability* dilakukan pada kegiatan sehari-hari seperti biasa untuk merawat diri dan kelangsungan hidup (Maslim, 2013). Menurut hasil Riskesdas pada tahun 2013 menyatakan 1,7 per mil dari penduduk Indonesia mengalami gangguan jiwa berat. Gangguan jiwa berat ini ditandai dengan gejala berupa halusinasi, ilusi, waham, anehnya tingkah laku, proses berpikir dan kemampuan berpikir yang terganggu (Badan Penelitian dan Pengembangan Kesehatan, 2013). Terdapat berbagai jenis gangguan jiwa berat seperti Skizofrenia dan Episode Depresif Berat dengan Gejala Psikotik. Berikut merupakan penjelasan mengenai penyakit tersebut:

2.2.1 Skizofrenia

Skizofrenia adalah gangguan mental serius dan melumpuhkan, mempengaruhi memori, perhatian dan fungsi eksklusif manusia yang menimpa sekitar 1% dari populasi umum (Hiesh, et al., 2013). Pada umumnya, Skizofrenia ditandai dengan penyimpangan yang fundamental serta karakteristik dari pikiran dan persepsi. Afek yang tidak wajar juga merupakan suatu tanda penyakit ini. Orang dengan penyakit Skizofrenia memiliki kesadaran yang jernih dan kemampuan intelektualnya yang terpelihara, meskipun kemampuan kognitif tertentu menurun dan berkembang kemudian (Maslim, 2013). Terdapat banyak penderita Skizofrenia mengalami keterbatasan dan keberadaan terisolasi bahkan



memiliki tingkat pengangguran yang tinggi, pendapatan rendah, dan kesehatan fisik yang buruk. Secara teori Skizofrenia merupakan gangguan mental kronis yang ditandai dengan delusi, halusinasi, pikiran kacau, dan mengalami perubahan perilaku. Kondisi tersebut dapat dikatakan sebagai gangguan jiwa dikarenakan penderita tidak mampu membedakan kenyataan dengan pikirannya sendiri.

Gejala seseorang terserang Skizofrenia berdasarkan data rekam medik yang didapatkan adalah penderita mudah marah, sering seyum-senyum sendiri, tertawa sendiri, berbicara melantur, sering berbicara kotor, memakan makanan sampah, sering curiga terhadap orang lain, korban pasung, sering keluyuran, gluyur, gelisah, membenturkan kepala, menelantarkan diri, bingung, tidak bisa berjalan, berbicara sendiri, mengomel, berperilaku aneh, tidak mau makan, pasif, BAB (buang air besar) di sembarang tempat, sering membuka celana dan menunjukkannya kepada orang lain, tidak merawat diri, mencoba bunuh diri, dan tidak mau minum obat.

Terdapat banyak sekali jenis-jenis dari penyakit Skizofrenia sendiri menurut Maslim (2013), seperti:

1. Skizofrenia Paranoid (F20.0): merupakan penyakit Skizofrenia dengan gejala halusinasi mendengar suara yang mengancam pasien atau memberi perintah, mengalami halusinasi pembauan atau pengecap rasa, delusi yang dikendalikan, dipengaruhi, atau *passivity*, dan keyakinan bahwa dirinya dikejar-kejar.
2. Skizofrenia Hebefrenik (F20.1): penyakit ini pertama kali ditegakkan pada hanya pada usia remaja pada kisaran usia 15-25 tahun. Namun, untuk mendiagnosa penyakit ini dibutuhkan waktu pengamatan sekitar dua atau tiga bulan. Gejala umumnya adalah perilaku tidak bertanggung jawab dan tidak bisa diramalkan, ada kecenderungan untuk selalu menyendiri, perilaku yang hampa tujuan dan hampa perasaan, afeknya yang dangkal serta tidak wajar, serta mengalami disorganisasi dan pembicaraan tak menentu secara bertahan.
3. Skizofrenia Katatonik (F20.2): gejala yang ada pada pasien dengan penyakit ini adalah berkurangnya reaktivitas pada lingkungan, gerakan, serta aktifitas spontan (*stupor*). Kemudian gaduh-gelisah, memperlihatkan posisi tubuh tertentu yang tidak wajar, negativisme terhadap semua perintah, rigiditas, fleksibilitas cerea, tidak komunikatif, perilaku yang patuh pada perintah, dan melakukan pengulangan terhadap kata serta kalimat. Namun, untuk mendiagnosa pasien dengan gejala seperti ini harus diperoleh bukti untuk gejala lain yang memadai.
4. Skizofrenia Tak Terinci (*Undifferentiated*) (F20.3): untuk mendiagnosa pasien dengan penyakit ini adalah dengan ciri umum untuk mendiagnosa Skizofrenia, bukan termasuk Skizofrenia Paranoid, Hebefrenik, atau katatonik serta bukan termasuk skizofrenia residual atau depresi pasca-skizofrenia.
5. Depresi pasca-Skizofrenia (F20.4): pasien dengan penyakit ini adalah pasien yang telah menderita penyakit Skizofrenia selama paling tidak 12 bulan terakhir, namun gejala Skizofrenia masih ada dalam diri pasien walaupun tidak

mendominasi. Gejala depresif yang menonjol dan mengganggu pasien dalam kurun waktu paling sedikit dua minggu lamanya.

6. Skizofrenia Residual (F20.5): untuk meyakinkan diagnosis penyakit ini adalah dengan gejala Skizofrenia “negatif” yang menonjol, setidaknya ada riwayat pada masa lalu yang menunjukkan kriteria Skizofrenia, sedikitnya telah melebihi satu tahun gejala seperti halusinasi sangat berkurang, dan tidak mengalami gangguan otak organik lain.
7. Skizofrenia Simpleks (F20.6): untuk mendiagnosa penyakit ini sedikit lebih sulit dibandingkan jenis lainnya, dikarenakan tergantung pada perkembangan dari pasien tersebut dari gejala Skizofrenia “negatif” tanpa riwayat halusinasi, waham dan lainnya.
8. Skizofrenia Lainnya (F20.8)
9. Skizofrenia YTT (F20.9): Skizofrenia yang tidak tergolongkan (*unspecified*).

2.2.2 Episode Depresif

Depresi merupakan gangguan mental yang menyebabkan penderitanya memiliki mood dalam keadaan rendah terus menerus, kehilangan gairah hidup, dan selalu berpikir buruk. Penyebab seseorang terserang depresi ada berbagai macam seperti faktor ekonomi, kehidupan sehari-hari, akibat kecelakaan, dan banyak hal lainnya yang dapat menjadi penyebab depresi. Orang yang mengalami depresi akan merasa tidak ada motivasi dalam menjalani kesehariannya dan selalu menyalahkan dirinya sendiri. Penyakit jiwa Depresi ini dapat berdampak pada pola pikir, pola hidup, suasana hati, perasaan, dan cara menghadapi hidup penderita. Gejala umum pada pasien dengan depresi ringan adalah afek depresi, pada depresi sedang adalah kehilangan minat dan kegembiraan, serta pada depresi berat adalah mudah lelah walau hanya bekerja sedikit yang berpengaruh pada menurunnya aktifitas pasien. Kemudian gejala lain pada depresi adalah berkurangnya konsentrasi, perhatian, harga diri, dan kepercayaan. Tak hanya itu, penderita depresi ini memiliki pikiran tentang masa depan yang suram (pesimis), merasa bersalah, berpikiran yang tidak berguna, memiliki rasa ingin bunuh diri, melakukan hal yang membahayakan diri, nafsu makan yang berkurang dan sulit tidur (Maslim, 2013).

Gejala yang digunakan berdasarkan data rekam medik yang didapatkan adalah sering melamun, pusing, tiba-tiba diam, tidak konsentrasi, kecemasan, malas, kurang semangat, seing bepikir negatif, menangis, guling-guling, sulit diarahkan, memukul diri, tegang, lemas, gemetar, melukai diri sendiri, sedih, mendengar bisikan, malu (menutupi wajah), bangun pagi mondar-mandir, rasa bersalah, berteriak, mengingat masa lalu, tertekan, kaku, sulit tidur, merasa orang lain membicarakannya, dan minder. Untuk mendiagnosa depresi dari tingkat ringan hingga berat dibutuhkan waktu sekurang-kurangnya dua minggu, namun proses diagnosa ini dapat dipercepat jika pasien mengalami gejala yang sangat berat dan berlangsung cepat (Maslim, 2013).

Seperti yang diketahui bahwa depresi memiliki 3 tingkatan, berikut penjelasan depresi berdasarkan tingkatan tersebut menurut Maslim(2013):

1. Episode Depresif Ringan (F32.0): pada depresif ringan ini paling tidak dua dari tiga gejala umum terpenuhi dan paling tidak dua dari gejala lainnya. Lama episodenya berlangsung adalah paling tidak sekitar dua minggu serta sedikit kesulitan dalam melakukan pekerjaan dan kegiatan sosialnya.
2. Episode Depresif Sedang (F32.1): gejala pada depresif ringan paling tidak dua dari tiga gejala umum dan tiga atau empat lebih baik dari gejala lainnya terpenuhi. Episodenya berlangsung minimum sekitar dua minggu lamanya. Kemudian pasien akan mengalami kesulitan yang nyata dalam melakukan pekerjaan, kegiatan sosial maupun rumah tangga.
3. Episode Depresif Berat tanpa Gejala Psikotik (F32.2): pada depresif sedang harus memenuhi ketiga gejala yang ada pada gejala umum dan paling tidak empat dari gejala lainnya dan beberapa diantaranya betingkat berat. Episodenya biasa berlangsung seharusnya dua minggu, namun bila gejala terlalu berat dan cepat berkembang maka akan baik jika mendiagnosis dalam kurang dari dua minggu. Pasien tidak mampu melanjutkan pekerjaan, kegiatan sosial, serta urusan rumah tangga kecuali dalam taraf yang sangat terbatas.
4. Episode Depresif Berat dengan Gejala Psikotik (F32.3): gejala yang sama dengan kriteria pada F32.2, namun disertai dengan waham tentang ide dosa, malapetaka yang akan mengancam (kemiskinan), dan merasa bertanggung jawab akan hal tersebut. Kemudian disertai halusinasi auditorik atau olfaktorik yang berupa suara menghina dan menuduh, atau mencium bau kotoran atau daging busuk.
5. Episode Depresif Lainnya (F32.8)
6. Episode Depresif YTT (F32.9): Episode Depresif yang tidak tergolongkan (*unspecified*).

2.3 Data Mining

Data mining merupakan penemuan informasi baru dari data yang sangat besar dengan mencari suatu pola-pola tertentu di dalam data besar tersebut (Arga, Lestari, & Sutanta, 2017). Proses penemuan informasi atau mengekstrak pengetahuan pada *data mining* dapat dilakukan proses seleksi dan eksplorasi dengan menggunakan perhitungan tertentu. Informasi atau pengetahuan yang didapat harus baru, bermanfaat, dan mudah dimengerti (Pramadhani & Setiadi, 2014). *Data mining* terbagi menjadi *supervised learning* dan *unsupervised learning*. *Supervised learning* merupakan sebuah pendekatan dengan menggunakan data latih dengan tujuan untuk mengelompokkan suatu data ke dalam data yang ada. Namun, *unsupervised learning* merupakan pendekatan yang tidak menggunakan data latih di dalamnya yang bertujuan untuk mengelompokkan suatu data menjadi n kelompok. Salah satu cara penemuan informasi atau mengekstrak pengetahuan dengan *data mining* adalah klasifikasi.

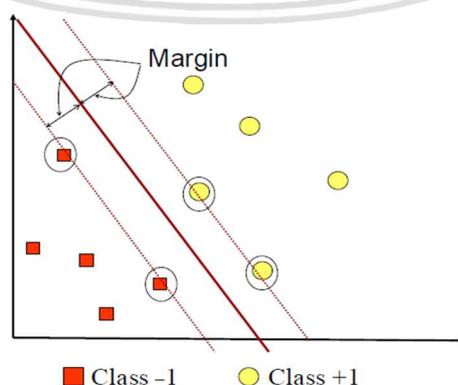
2.4 Klasifikasi

Menurut Kamus Besar Bahasa Indonesia (KBBI), klasifikasi merupakan penyusunan bersistem dalam kelompok atau golongan menurut kaidah atau standar yang ditetapkan. Klasifikasi adalah sebuah kegiatan pembangunan suatu model yang mengelompokkan objek yang memiliki atribut-atribut yang sama ke dalam suatu kelas yang telah ditentukan (Susilowati, Sabariah, & Gozali, 2015). Tujuan dari klasifikasi adalah untuk mengelompokkan benda yang memiliki ciri-ciri sama agar mudah dikenal. Dalam mengklasifikasikan objek dapat dilakukan berdasarkan persamaan, perbedaan, manfaat, ciri morfologi dan anatomi, dan berdasarkan ciri biokimia.

Klasifikasi menggunakan pendekatan *supervised learning* sehingga metode-metode klasifikasi menggunakan data latih di dalamnya. Dalam mengklasifikasikan suatu objek diawali dengan membuat aturan dalam pengklasifikasian menggunakan algoritme klasifikasi tertentu dengan tahap pembelajaran menggunakan data *training* dan tahap pengujian menggunakan data *testing* (Raharjo & Winarko, 2014). Terdapat banyak algoritme klasifikasi yang dapat digunakan untuk mengklasifikasikan suatu objek. Beberapa algoritma yang sering kali digunakan adalah *Naive Bayes*, *K-Nearest Neighbor*, *Decision Tree*, *Support Vector Machine*, dan *Neural Network*.

2.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah sebuah metode klasifikasi yang menggunakan parameter-parameter untuk mengklasifikasikannya dan kemudian dapat digunakan untuk mengklasifikasikan data baru. Metode SVM ini merupakan metode untuk mencari *hyperplane* terbaik dalam memisahkan dua kelas berbeda pada ruang vektor. Untuk menemukan *hyperplane* terbaik pada dua kelas dapat dilakukan dengan mengukur margin *hyperplane* tersebut dan mencari titik maksimalnya (Nugroho, Witarto, & Handoko, 2003). Pada dasarnya, metode SVM ini digunakan untuk penyelesaian masalah linear. Gambaran *support vector machine* terlihat seperti pada Gambar 2.1.



Gambar 2.1 Gambaran *Support Vector Machine* dengan *Hyperplane* Terbaik

Sumber: (Nugroho, Witarto, & Handoko, 2003)

Sesuai dengan konsep metode *support vector machine* yaitu mencari *hyperplane* yang terbaik untuk memisahkan dua kelas, Gambar 2.1 menyatakan gambaran *support vector machine* dengan *hyperplane* terbaik. Terlihat bahwa kelas -1 dan kelas +1 dipisahkan dengan satu garis lurus yang disebut *hyperplane* dan terdapat *margin* diantaranya. *Margin* merupakan jarak antar *hyperplane* dengan *pattern* terdekat dari setiap kelas dan *pattern* terdekat dapat disebut dengan *support vector*. Untuk objek merah dan kuning yang berada pada lingkaran hitam merupakan *support vector*.

Dinotasikan label masing-masing data dengan $y_i \in \{-1, +1\}$ untuk $i = 1, 2, 3, \dots, n$. Diasumsikan kedua kelas terpisah secara sempurna oleh *hyperplane* digunakan Persamaan 2.1, yang termasuk kelas -1 (sampel negatif) digunakan Persamaan 2.2, dan yang termasuk kelas +1 (sampel positif) digunakan Persamaan 2.3 dapat didefinisikan seperti fungsi berikut.

$$\vec{w} \cdot \vec{x} + b = 0 \tag{2.1}$$

$$\vec{w} \cdot \vec{x} + b \leq -1 \tag{2.2}$$

$$\vec{w} \cdot \vec{x} + b \geq +1 \tag{2.3}$$

Margin terbesar dapat ditemukan dengan fungsi $1/\|\vec{w}\|$ dan dapat dirumuskan sebagai *Quadratic Programming (QP) Problem*. Dalam *QP Problem* akan dicari titik minimal (Persamaan 2.4) dengan memperhatikan constraint (Persamaan 2.5) sebagai berikut.

$$\min_{\vec{w}} \tau(w) = \frac{1}{2} \|\vec{w}\|^2 \tag{2.4}$$

$$y_i(\vec{x}_i \cdot \vec{w} + b) - 1 \geq 0, \forall_i \tag{2.5}$$

Dengan memodifikasi persamaan *Lagrange Multiplier* sebagai maksimalisasi *problem* yang hanya mengandung α_i dengan Persamaan 2.6 sebagai berikut.

Maximize:

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j \tag{2.6}$$

Keterangan:

$$\alpha_i \geq 0 \quad \sum_{i=1}^l \alpha_i y_i = 0$$

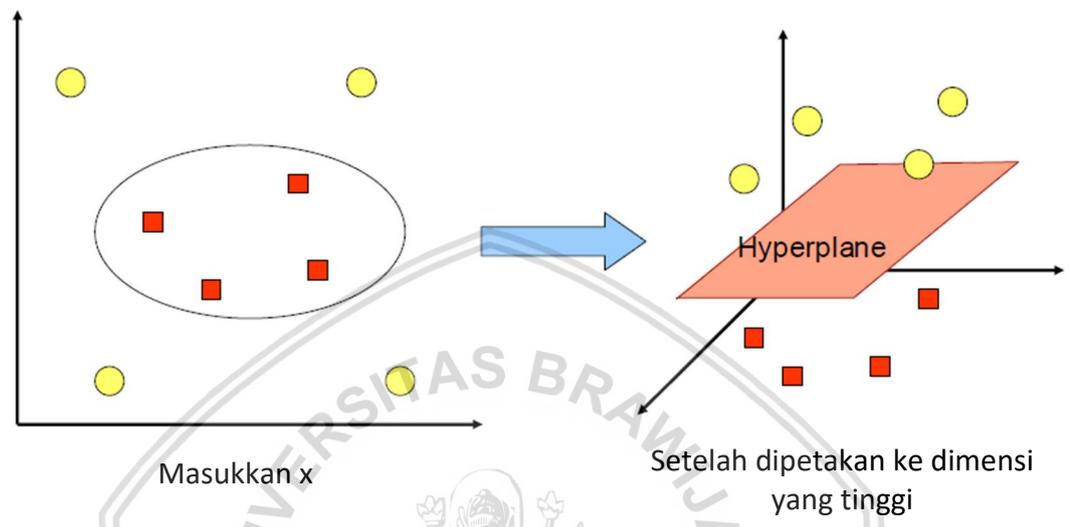
Hasil perhitungan yang diperoleh dari persamaan ini terdapat α_i yang kebanyakan bernilai positif dan data yang berkorelasi dengan α_i yang positif inilah *support vector*.

2.5.1 SVM Non-Linear

Dalam permasalahan di dunia nyata akan sangat jarang sekali menggunakan SVM yang bersifat linear, SVM yang bersifat non-linear akan lebih banyak digunakan. SVM non-linear digunakan untuk menyelesaikan masalah non-linear. Untuk dapat menyelesaikan masalah non-linear maka SVM dimodifikasi dengan



adanya masukan fungsi kernel di dalamnya. Pada SVM non-linear, proses diawali dengan data \vec{x} dipetakan oleh $\Phi(\vec{x})$ ke dalam dimensi yang lebih tinggi dikarenakan garis pemisah antar kedua kelasnya tidak membentuk garis linear. Pada dimensi inilah garis pemisah kedua kelas tersebut atau *hyperplane* dapat dikonstruksikan atau dapat berbentuk linear (Nugroho, Witarto, & Handoko, 2003). Seperti pada Gambar 2.2.



Gambar 2.2 Suport Vector Machine Non-Linear

Menurut teori Mercer fungsi kernel yang awalnya merupakan *dot product* antara $\Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$ dapat digantikan dengan fungsi kernel $K(\vec{x}_i, \vec{x}_j)$ agar mempermudah pemahaman. Kernel yang mendefinisikan transformasi Φ secara implisit disebut *kernel trick* yang dirumuskan seperti:

$$K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j) \tag{2.7}$$

Pada SVM, fungsi kernel sangat mempermudah dalam penentuan *support vector*. Kernel-kernel yang umum digunakan adalah seperti pada Tabel 2.2.

Tabel 2.2 Kernel yang umum digunakan pada SVM

Jenis Kernel	Fungsi Kernel
<i>Polynomial</i>	$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^p$
<i>Gaussian</i>	$K(\vec{x}_i, \vec{x}_j) = \exp\left(-\frac{\ \vec{x}_i - \vec{x}_j\ ^2}{2\sigma^2}\right)$
<i>Sigmoid</i>	$K(\vec{x}_i, \vec{x}_j) = \tanh(\alpha \vec{x}_i \cdot \vec{x}_j + \beta)$

Sumber: (Nugroho, Witarto, & Handoko, 2003)

2.5.2 Sequential Training

Sequential training merupakan proses yang sederhana yang dikembangkan oleh Vijayakumar untuk menemukan *hyperplane* yang terbaik dalam SVM. Pada

dasarnya, untuk menemukan *hyperplane* terbaik dalam SVM adalah dengan cara merumuskannya ke dalam *Quadratic Programming (QP) problem*. Kemudian akan diselesaikan menggunakan *library* yang mayoritas tersedia dalam bahasa numerik atau angka. Berikut merupakan metode sekuensial yang dikembangkan oleh Vijayakumar (Vijayakumar & Wu, 1999):

1. Menginisialisasi nilai alpha awal $\alpha_i = 0$. Kemudian menghitung nilai Matriks *Hessian* dengan fungsi seperti berikut:

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2), \text{ dengan } i, j = 1, 2, 3 \dots, n. \quad (2.8)$$

2. Menghitung fungsi a, b, dan c hingga $i = n$.

- a. Menghitung nilai E_i dengan fungsi persamaan:

$$E_i = \sum_{j=1}^n \alpha_j D_{ij} \quad (2.9)$$

- b. Menghitung nilai $\delta\alpha_i$ dengan fungsi persamaan:

$$\delta\alpha_i = \min\{\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\} \quad (2.10)$$

- c. Menghitung nilai α_i dengan fungsi persamaan:

$$\alpha_i = \alpha_i + \delta\alpha_i \quad (2.11)$$

Keterangan:

D_{ij} = matriks *Hessian*

E_i = *error rate*

α_i = nilai alpha ke-i

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

C = nilai konstanta

3. Mengulangi langkah 2 hingga nilai α_i mencapai konvergen dengan syarat $\max(|\delta\alpha_i|) < \epsilon$ (*epsilon*) atau hingga mencapai iterasi maksimum (*itermax*).

2.5.3 Testing SVM

Proses *testing* pada SVM ini digunakan untuk mengklasifikasikan objek yang akan diuji untuk masuk ke dalam kelompok yang telah tersedia. Berikut merupakan proses *testing* pada SVM (Vijayakumar & Wu, 1999):

1. Langkah pertama adalah mengitung nilai kernel *testing* dengan persamaan:

$$K(x_{test}, x_i) = (x_{test} \cdot x_i + 1)^d \quad (2.12)$$

Keterangan:

x_{test} = data *testing*

x_i = data *training* ke-i

d = *degree*

2. Menghitung nilai $f(x)$ dengan persamaan:

$$f(x) = \sum_{i=1}^n \alpha_i y_i K(x_{test}, x_i) + b \tag{2.13}$$

$$b = -\frac{1}{2} [\sum_{i=1}^n \alpha_i y_i K(x_i, x^+) + \sum_{i=1}^n \alpha_i y_i K(x_i, x^-)] \tag{2.14}$$

Keterangan:

α_i = nilai alpha ke-i

y_i = kelas data ke-i

y_j = kelas data ke-j

b = nilai bias

n = banyak data

$K(x_i, x^+)$ = kernel untuk kelas positif

$K(x_i, x^-)$ = kernel untuk kelas negatif

3. Mengklasifikasi data dengan persamaan:

$$Fungsi\ klasifikasi = sign\ f(x) \tag{2.15}$$

2.6 Evaluasi Hasil

Evaluasi hasil dibutuhkan untuk dapat mengetahui sistem yang dibuat dapat berjalan dengan baik sesuai dengan tujuannya. Nilai akurasi suatu sistem dapat menjadi acuan keberhasilan sistem tersebut. Terdapat beberapa cara untuk mengukur kinerja dari model yang digunakan dalam sistem klasifikasi. Pada penelitian ini digunakan fungsi *confusion matrix* yang merupakan tabel untuk menunjukkan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan (Indriani, 2014). Tabel *confusion matrix* dapat dilihat seperti pada Tabel 2.3.

Tabel 2.3 Model Confusion Matrix

Kelas Aktual	Kelas Klasifikasi	
	Kelas Positif	Kelas Negatif
Kelas Positif	TP	FN
Kelas Negatif	FP	TN

Keterangan untuk Tabel 2.3. adalah:

True Possitive (TP), menyatakan jumlah data pada kelas positif yang benar diklasifikasikan sebagai kelas positif.

False Possitive (FP), menyatakan jumlah data pada kelas negatif yang salah diklasifikasikan sebagai kelas positif.

False Negative (FN), menyatakan jumlah data pada kelas positif yang salah diklasifikasikan sebagai kelas negatif.



True Negative (TN), merupakan jumlah data pada kelas negatif yang benar diklasifikasikan sebagai kelas negatif.

Untuk mengukur akurasi yang dimiliki sistem dapat digunakan fungsi *accuracy*, *precision*, *recall*, dan *f-measure* seperti berikut (Rosandy, 2016):

$$\text{Akurasi} = \frac{TP+TN}{TP+FN+FP+TN} \times 100\% \quad (2.16)$$

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% \quad (2.17)$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \quad (2.18)$$

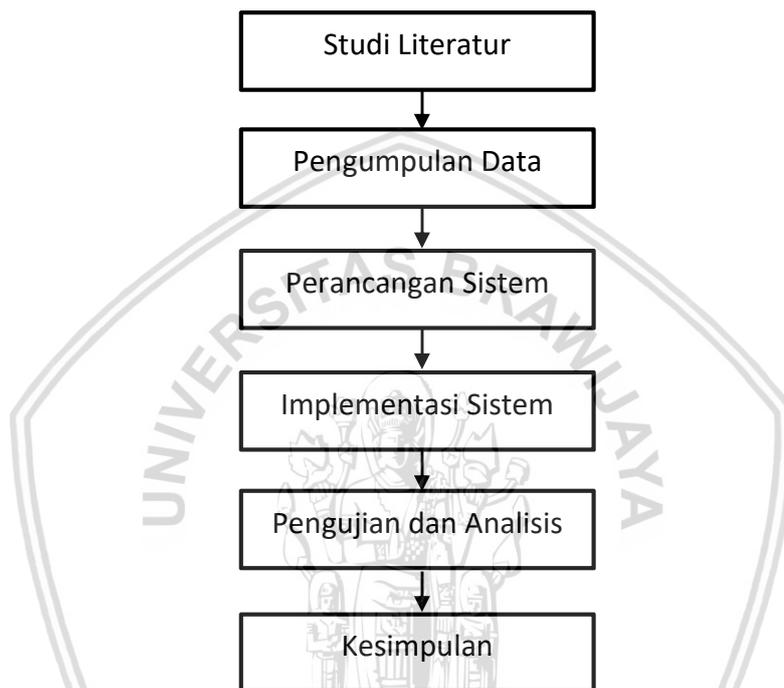
$$F - \text{Measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{rec}} \times 100\% \quad (2.19)$$

2.7 K-Fold Cross Validation

K-fold cross validation merupakan suatu metode pengujian untuk mengetahui rata-rata keberhasilan suatu sistem dengan mengacak data latih dan data uji sebanyak K yang digunakan. *K-fold cross validation* dilakukan dengan membagi data menjadi K bagian yang telah ditentukan dan dengan ukuran/banyak yang sama setiap bagiannya. Pada proses, dipilih satu bagian data yang akan menjadi data uji dan bagian lainnya akan menjadi data latih (Putri, Suparti, & Rahmawati, 2014). Sehingga mendapatkan hasil akurasi pada setiap percobaan dan hasil dari setiap percobaan akan dirata-ratakan menjadi hasil akhir dari proses. Menurut Hastie et al (2008) dalam Banjarsari, Budiman, & Farmadi (2015), untuk mengukur tingkat kesalahan yang terjadi dapat digunakan K sebanyak 5 dan 10. Karena dengan K tersebut sudah cukup untuk membedakan data latih yang dimiliki.

BAB 3 METODOLOGI

Metodologi penelitian merupakan langkah-langkah yang dilakukan dalam menyelesaikan permasalahan pada penelitian. Dalam penelitian ini, peneliti melakukan 6 tahapan penelitian yaitu studi literatur, pengumpulan data, perancangan sistem, implementasi sistem, pengujian dan analisis sistem, dan pengambilan kesimpulan. Tahapan-tahapan tersebut dapat ditunjukkan dengan Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.1 Studi Literatur

Studi literatur merupakan pengumpulan informasi-informasi dari berbagai sumber terpercaya dan memiliki kaitan dengan penelitian yang dilakukan. Informasi-informasi tersebut akan dijadikan landasan dalam penyelesaian masalah pada penelitian. Studi literatur dapat berupa jurnal, artikel, buku, artikel web, e-book, dan lainnya. Studi literatur yang digunakan penelitian ini adalah penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik, Klasifikasi, dan *Support Vector Machine* (SVM).

3.2 Pengumpulan Data

Pengumpulan data merupakan tahap dimana peneliti mengumpulkan data yang akan digunakan dalam penelitian. Pada penelitian ini, peneliti mengumpulkan data rekam medik pasien penderita Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik. Metode yang digunakan dalam

pengumpulan data ini adalah dengan metode pengumpulan data Sekunder, yaitu mengumpulkan data yang sudah ada pada Rumah Sakit Jiwa Dr. Radjiman Wediodiningrat Lawang.

3.3 Perancangan Sistem

Perancangan sistem merupakan tahap untuk merancang sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik metode *Support Vector Machine* (SVM). Dimana data gejala pasien yang terserang penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik akan menjadi data fitur pada sistem yang akan dibuat. Langkah awal yang dilakukan adalah dengan proses *sequential training* pada data latih untuk menemukan garis *hyperplane* yang optimal. Kemudian nilai tersebut dimasukkan ke dalam proses *testing* pada data uji untuk menemukan hasil klasifikasi dari data uji.

3.4 Implementasi Sistem

Pada tahap ini, sistem yang telah dirancang sebelumnya dengan perhitungan *Support Vector Machine* (SVM) diimplementasikan ke sebuah sistem dengan menggunakan bahasa pemrograman Java dan menggunakan *microsoft excel* sebagai media penyimpanan basis data.

3.5 Pengujian dan Analisis

Pada tahap pengujian ini, sistem klasifikasi diuji keberhasilannya dengan pengujian akurasi terhadap sistem yang telah diimplementasikan sebelumnya. Pengujian yang dilakukan dalam penelitian ini adalah meliputi pengujian terhadap variabel λ , variabel γ , pengujian terhadap nilai C (*Complexity*), serta pengujian terhadap iterasi maksimal.

3.6 Kesimpulan

Tahap terakhir pada penelitian ini adalah pengambilan kesimpulan. Setelah melalui proses-proses serta hasil dan analisis sistem, maka akan diambil kesimpulan dari penelitian yang telah dibuat. Dengan kesimpulan ini akan didapatkan kekurangan sistem yang bisa menjadi saran untuk penelitian selanjutnya.

BAB 4 PERANCANGAN

Bab perancangan ini akan membahas mengenai formulasi permasalahan pada sistem klasifikasi penyakit kejiwaan Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik, alur proses algoritme klasifikasi menggunakan metode *Support Vector Machine*, perhitungan manual, perancangan hasil serta perancangan pengujian dari sistem klasifikasi yang akan dibuat.

4.1 Formulasi Permasalahan

Sistem yang akan dibangun merupakan sistem klasifikasi penyakit kejiwaan yaitu Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik. Pada sistem ini terdapat 2 kelas penyakit yang akan diklasifikasikan dengan menggunakan metode *Support Vector Machine (SVM)*. Data yang digunakan merupakan data rekam medis dari RSJ. Radjiman Wediodiningrat Lawang yang terdiri dari 100 data dengan penyakit Skizofrenia Hebefrenik dan 100 data dengan penyakit Episode Depresi Berat dengan Gejala Psikotik. Data tersebut akan digabungkan menjadi 200 data yang akan digunakan sebagai data latih serta data uji dari sistem yang akan dibuat. Klasifikasi dengan menggunakan metode SVM ini, menggunakan kernel *Polynomial of degree 2*. Data latih akan diproses melalui tahapan *sequential training*, kemudian data uji akan diproses dengan tahapan *testing* yang dilakukan setelah tahapan *sequential training*.

Pada proses manualisasi, digunakan data latih sebanyak 10 data latih dan data uji sebanyak 2 data. Setiap data latih dan data uji terbagi atas dua kelas data yang seimbang. Data yang digunakan pada manualisasi dapat dilihat pada Tabel 4.1.

Tabel 4.1 Tabel Data Latih dan Data Uji

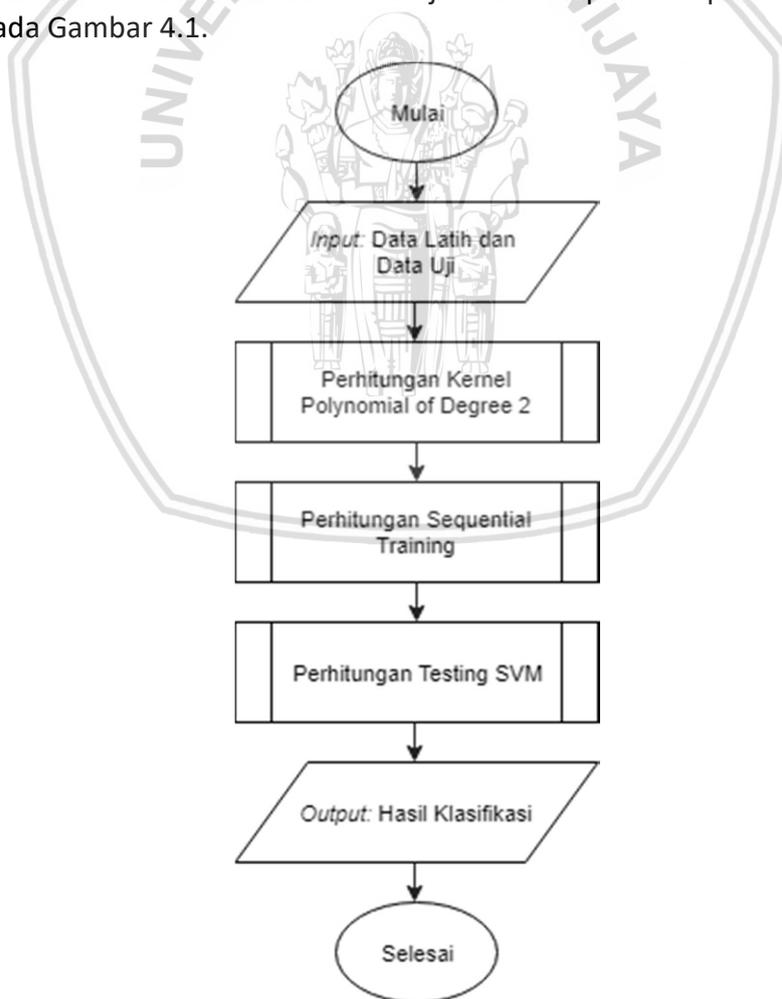
No.	G1	G2	G3	G4	G5	G6	G7	G8	...	G69	G70	Y
1	1	0	0	0	0	0	0	0	...	0	0	1
2	0	0	1	1	0	0	0	0	...	0	0	1
3	0	0	0	1	0	1	0	0	...	0	0	1
4	1	0	1	0	1	0	0	0	...	0	0	1
5	1	0	1	1	1	0	0	0	...	0	0	1
6	0	0	0	0	0	0	0	0	...	0	0	-1
7	0	0	0	0	0	0	0	0	...	0	0	-1
8	0	0	0	1	0	0	0	0	...	0	0	-1
9	0	0	0	0	0	0	0	0	...	0	0	-1
10	0	0	0	1	0	0	0	0	...	0	0	-1

11	1	0	1	1	1	0	0	0	...	0	0	1
12	1	0	0	1	0	0	0	0	...	0	1	-1

Pada tabel tersebut terlihat bahwa setiap parameter G1 sampai dengan G70 yang bernilai 0 menyatakan pasien tidak mengalami gejala tersebut, sedangkan yang bernilai 1 menyatakan bahwa pasien mengalami gejala tersebut. Parameter Y merupakan kelas klasifikasi dengan nilai -1 menunjukkan penyakit Episode Depresi Berat dengan Gejala Psikotik dan nilai 1 menunjukkan penyakit Skizofrenia Hebefrenik.

4.2 Alur Proses Algoritme Support Vector Machine (SVM)

Dalam proses perhitungan *Support Vector Machine*, langkah awal yang dilakukan adalah mengambil dataset yang dibutuhkan dalam sistem klasifikasi. Kemudian dilakukan proses perhitungan kernel SVM, pada penelitian ini digunakan kernel *Polynomial of degree 2*. Setelah itu dilakukan perhitungan *training SVM* dengan *sequential training* dan perhitungan *testing SVM* yang akan menghasilkan hasil klasifikasi dari data uji. Alur tahapan dari proses SVM dapat dilihat pada Gambar 4.1.



Gambar 4.1 Diagram Alir Algoritme Support Vector Machine

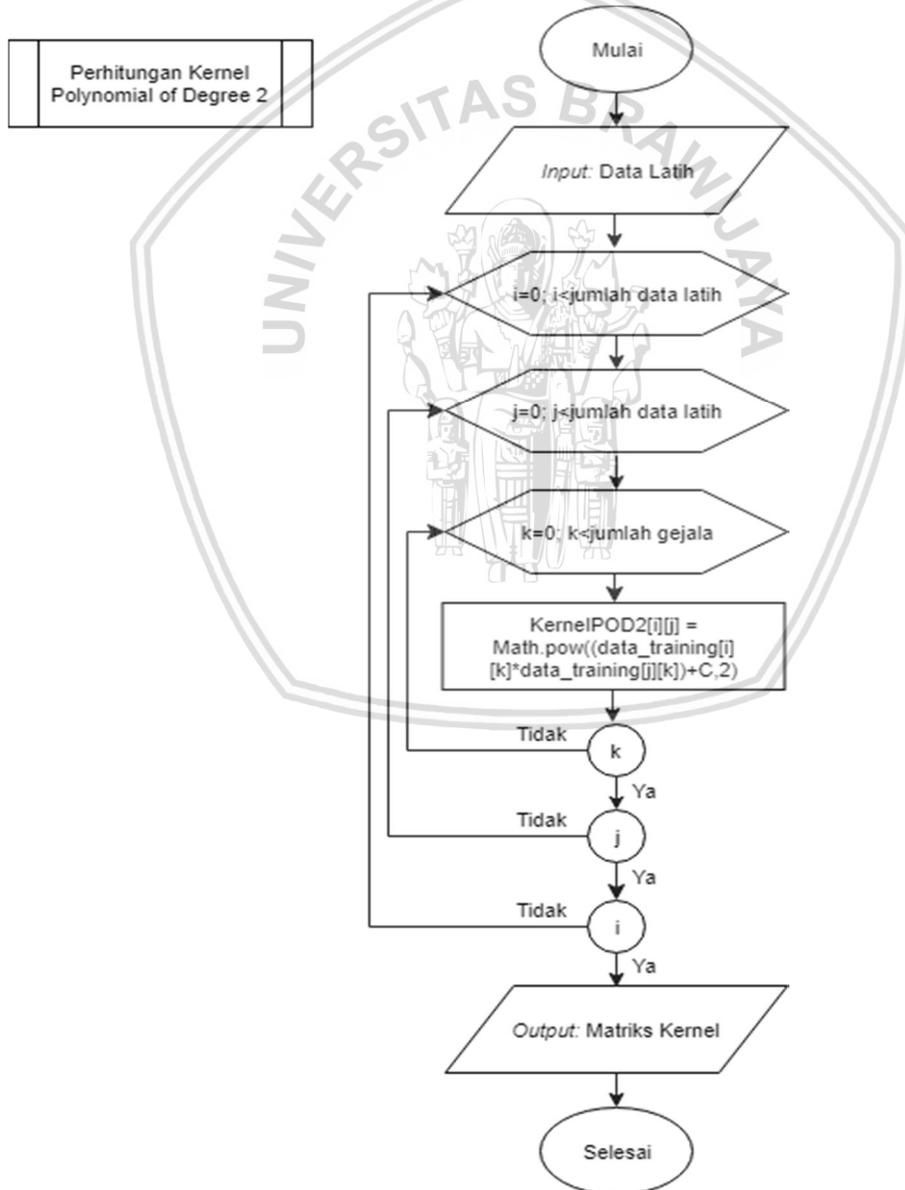


Penjelasan untuk Diagram Alir Algoritme *Support Vector Machine* adalah sebagai berikut:

1. Memasukkan data pasien Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik.
2. Melakukan perhitungan kernel dengan kernel *Polynomial of degree 2*.
3. Melakukan perhitungan *training SVM* dengan *Sequential Training*.
4. Melakukan perhitungan *testing SVM*.
5. Menampilkan hasil klasifikasi.

4.2.2 Alur Proses Perhitungan Kernel

Proses perhitungan kernel ini dimulai dengan mengambil data *training* yang akan diolah menggunakan kernel *Polynomial of degree 2*. Alur tahapan proses perhitungan kernel ini dapat dilihat pada Gambar 4.2.



Gambar 4.2 Diagram Alir Proses Perhitungan Kernel SVM

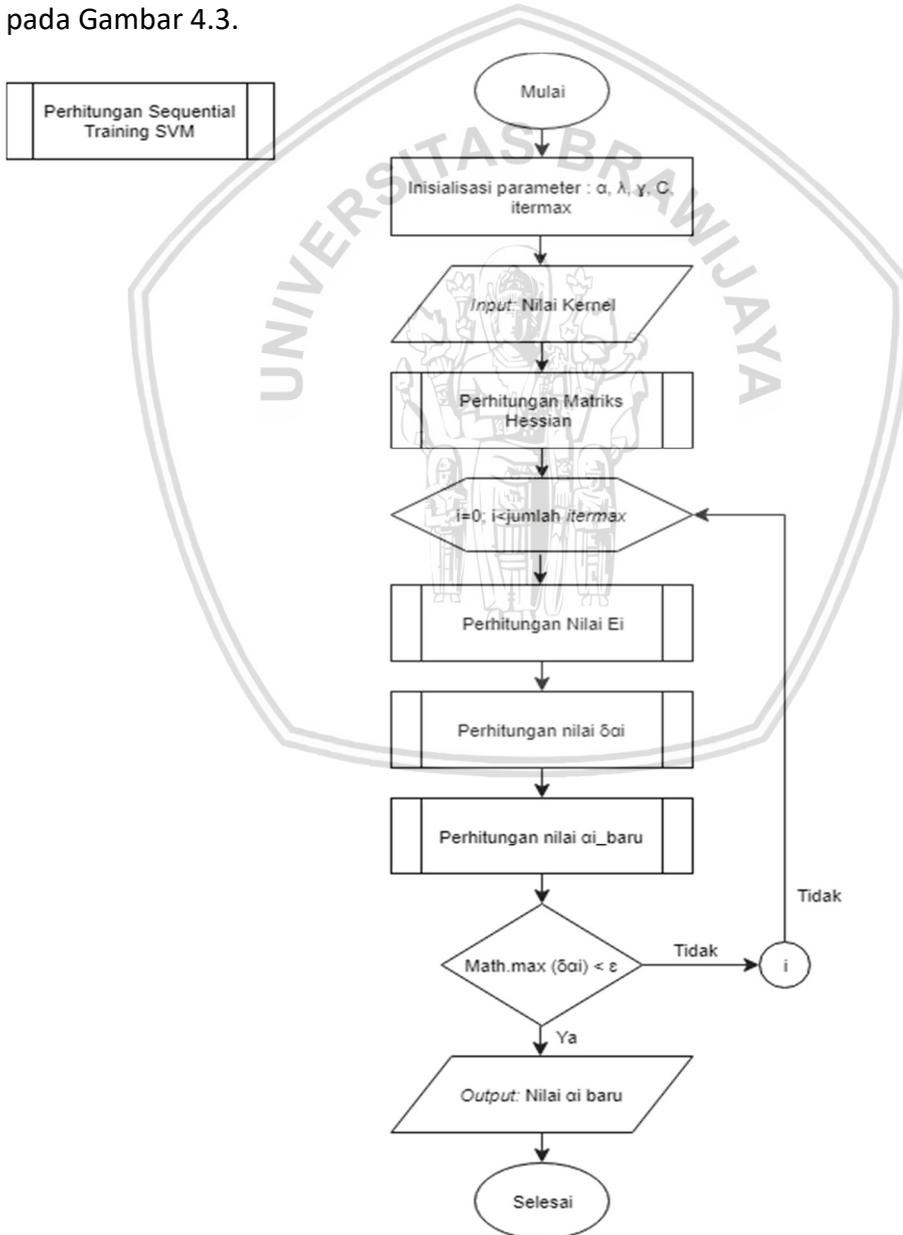


Penjelasan untuk Diagram Alir Proses Perhitungan Kernel SVM adalah sebagai berikut:

1. Memasukkan data *training*.
2. Melakukan perulangan terhadap banyaknya data *training* dan gejala.
3. Menghitung kernel *Polynomial of degree 2*.
4. Menampilkan matriks hasil perhitungan kernel.

4.2.3 Alur Proses Perhitungan *Sequential Training SVM*

Proses perhitungan *sequential training SVM* dilakukan dengan menginisialisasikan parameter dan mengambil nilai matriks kernel. Proses *sequential training* ini dilakukan untuk mendapatkan garis *hyperplane* yang optimal pada sistem klasifikasi. Tahapan pada *sequential training* ini dapat dilihat pada Gambar 4.3.



Gambar 4.3 Diagram Alir Proses Perhitungan *Training SVM*

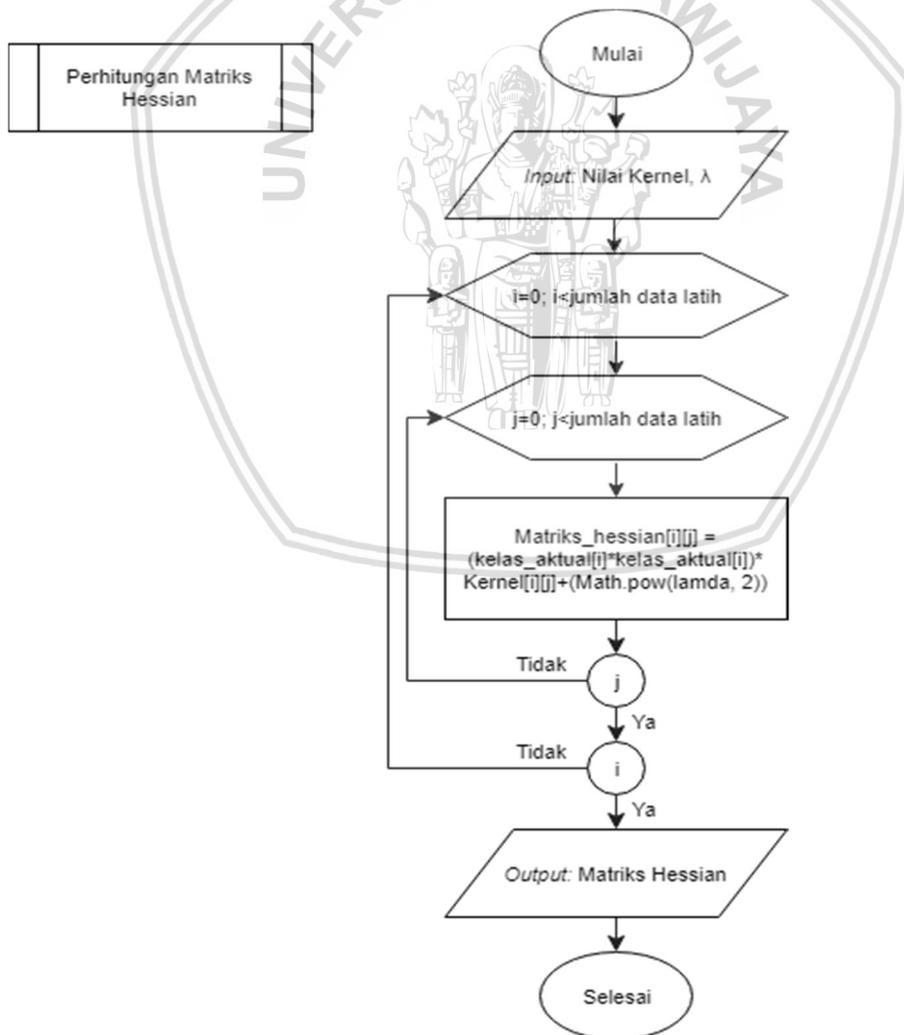


Penjelasan untuk Diagram Alir Proses Perhitungan *Training* SVM adalah sebagai berikut:

1. Menginisialisasi parameter.
2. Memasukkan nilai matriks kernel.
3. Menghitung matriks *Hessian* dengan menggunakan Persamaan 2.8.
4. Menghitung nilai E_i dengan menggunakan Persamaan 2.9.
5. Menghitung nilai $\delta\alpha_i$ dengan menggunakan Persamaan 2.10.
6. Menghitung nilai α_i dengan menggunakan Persamaan 2.11.
7. Iterasi akan dilakukan hingga nilai α_i mencapai konvergen ($|\delta\alpha_i| < \epsilon$) atau iterasi telah mencapai *itermax*.

4.2.4 Alur Proses Perhitungan Matriks *Hessian*

Proses perhitungan Matriks *Hessian* merupakan tahap pertama dalam proses *sequential training* pada SVM. Hasil yang didapatkan akan membentuk matriks yang banyak baris dan kolomnya sama dengan banyak data yang digunakan. Proses perhitungan matriks *Hessian* dapat dilihat seperti pada Gambar 4.4.



Gambar 4.4 Diagram Alir Proses Perhitungan Matriks *Hessian*

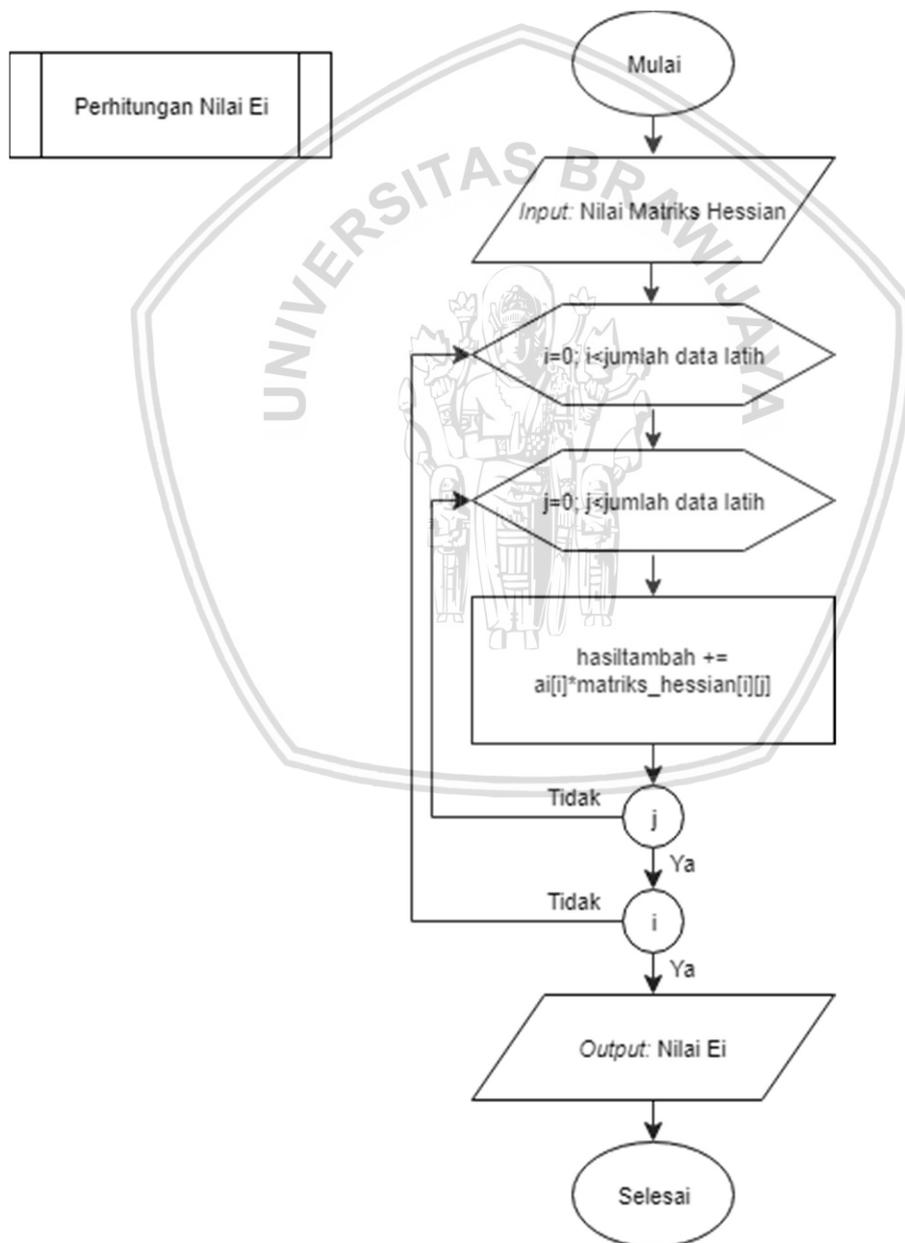


Penjelasan untuk Diagram Alir Proses Perhitungan Matriks *Hessian* adalah sebagai berikut:

1. Memasukkan nilai kernel dan λ .
2. Melakukan perulangan terhadap banyaknya data *training* dan gejala.
3. Menghitung matriks *Hessian* dengan Persamaan 2.8.
4. Menampilkan hasil matriks *Hessian*.

4.2.5 Alur Proses Perhitungan E_i

Proses perhitungan nilai E_i ini dilakukan dengan cara mengalikan antara nilai a_i dengan matriks *Hessian*. Proses ini dapat dilihat seperti pada Gambar 4.5.



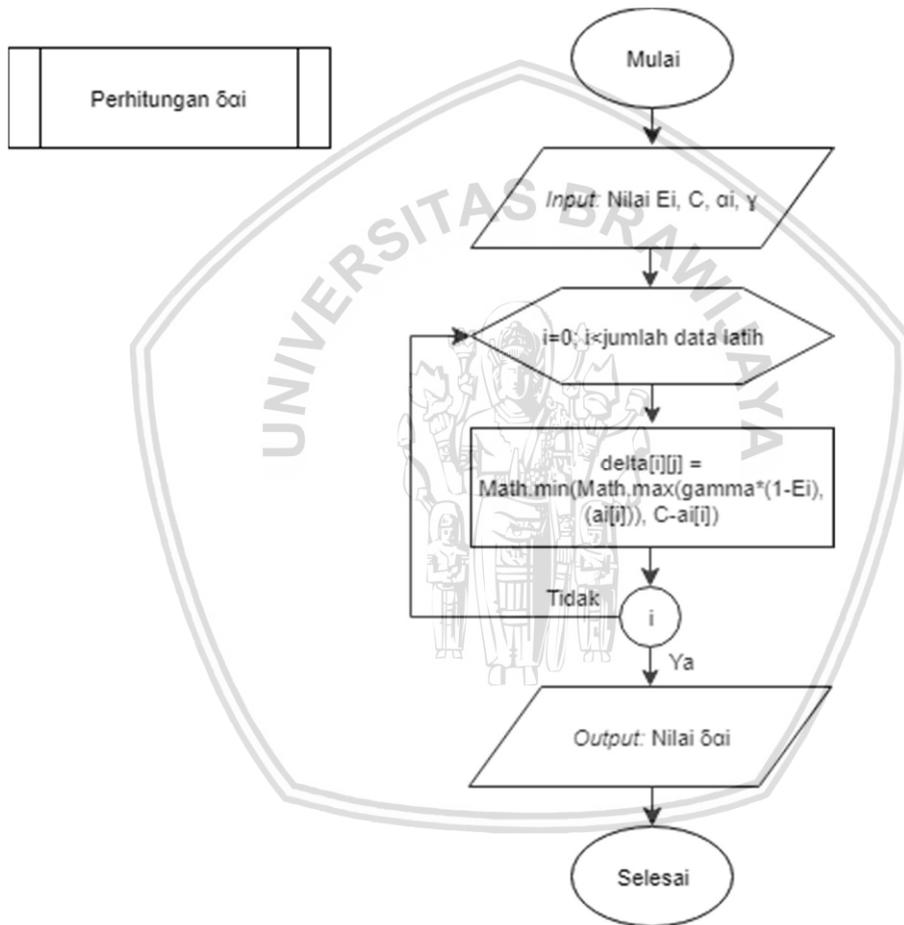
Gambar 4.5 Diagram Alir Proses Perhitungan E_i

Penjelasan untuk Diagram Alir Proses Perhitungan E_i adalah sebagai berikut:

1. Memasukkan nilai matriks *Hessian*.
2. Melakukan perulangan terhadap banyaknya data *training* dan gejala.
3. Menghitung nilai E_i sesuai dengan Persamaan 2.9.
4. Menampilkan nilai E_i .

4.2.6 Alur Proses Perhitungan δ_{ai}

Tahapan dalam proses perhitungan δ_{ai} dapat dilihat seperti pada Gambar 4.6.



Gambar 4.6 Diagram Alir Proses Perhitungan Nilai δ_{ai}

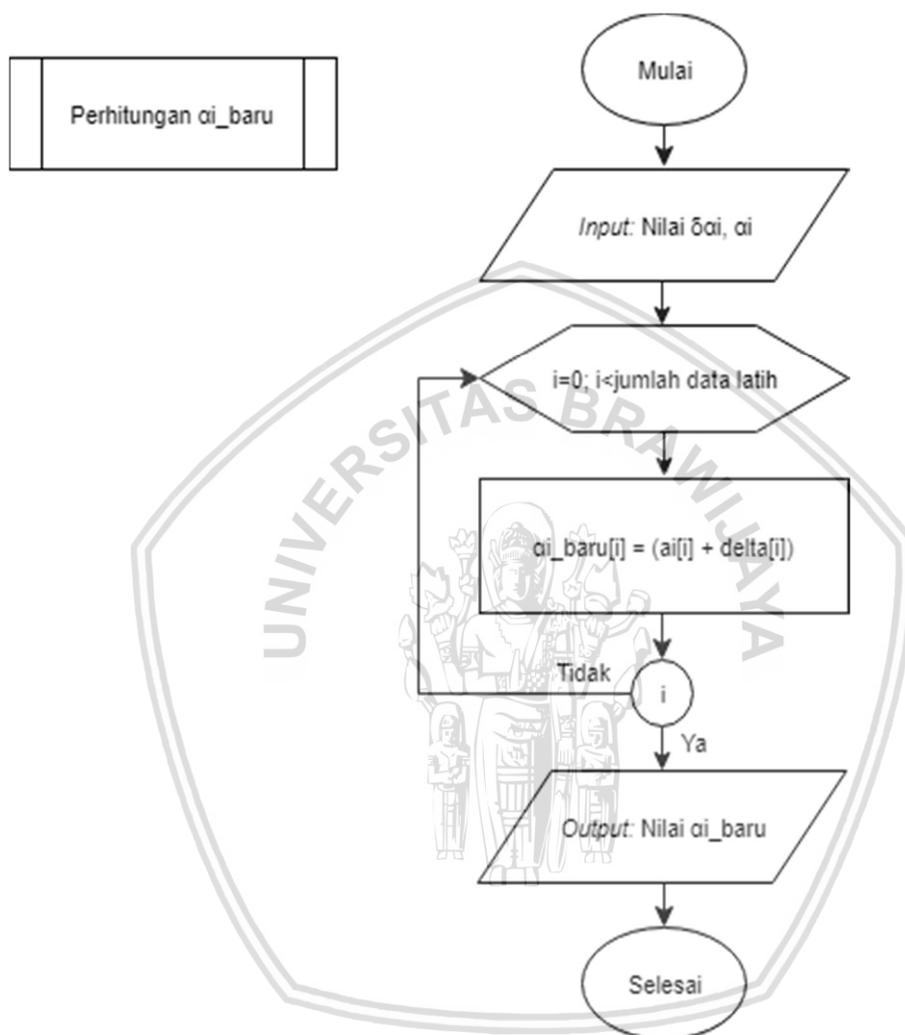
Penjelasan untuk Diagram Alir Proses Perhitungan Nilai δ_{ai} adalah sebagai berikut:

1. Memasukkan nilai E_i , C , a_i , γ .
2. Melakukan perulangan terhadap banyaknya data *training*.
3. Menghitung nilai δ_{ai} sesuai dengan Persamaan 2.10.
4. Menampilkan nilai δ_{ai} .



4.2.7 Alur Proses Perhitungan α_i baru

Proses perhitungan α_i merupakan proses untuk mencari nilai α_i baru dengan menambahkan nilai α_i lama dengan nilai $\delta\alpha_i$. Proses ini dapat dilihat pada Gambar 4.7.



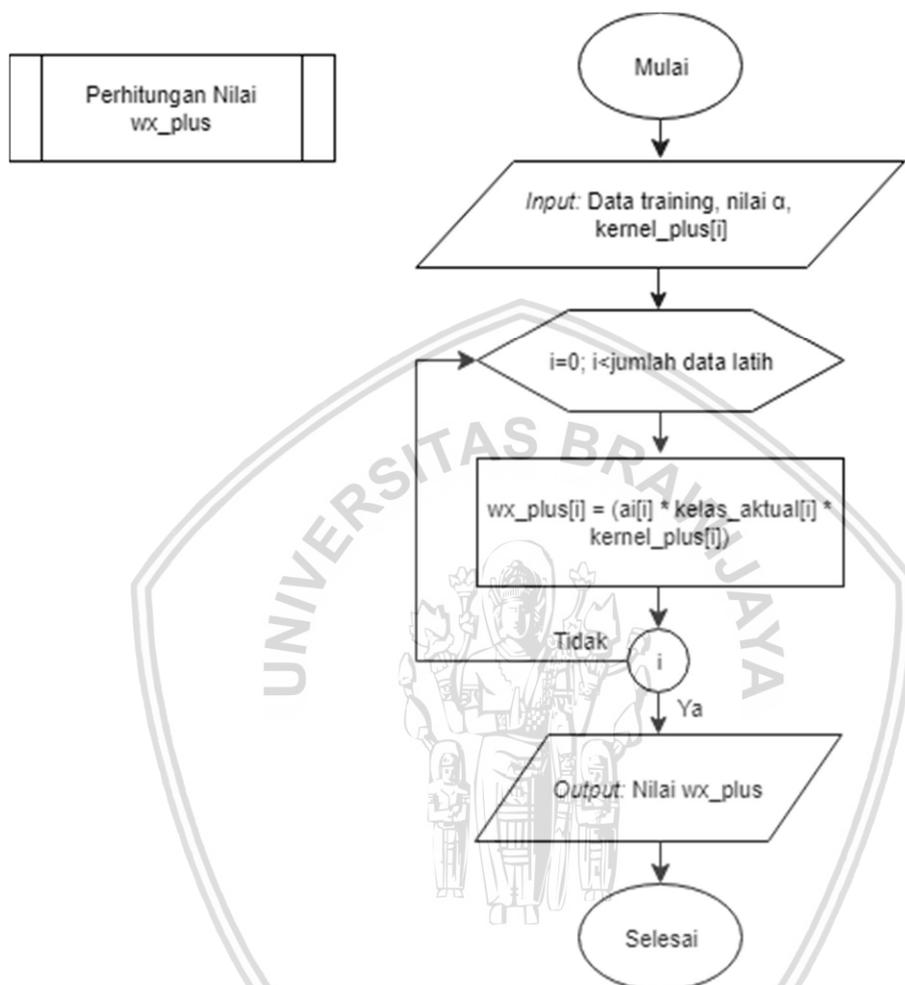
Gambar 4.7 Diagram Alir Proses Perhitungan α_i

Penjelasan untuk Diagram Alir Proses Perhitungan α_i adalah sebagai berikut:

1. Memasukkan nilai $\delta\alpha_i$ dan α_i .
2. Melakukan perulangan terhadap banyaknya data *training*.
3. Menghitung nilai α_i baru sesuai dengan Persamaan 2.11.
4. Menampilkan nilai α_i baru.

4.2.8 Alur Proses Perhitungan Nilai wx^+

Tahapan dalam Proses menghitung nilai wx^+ dapat dilihat seperti Gambar 4.8.



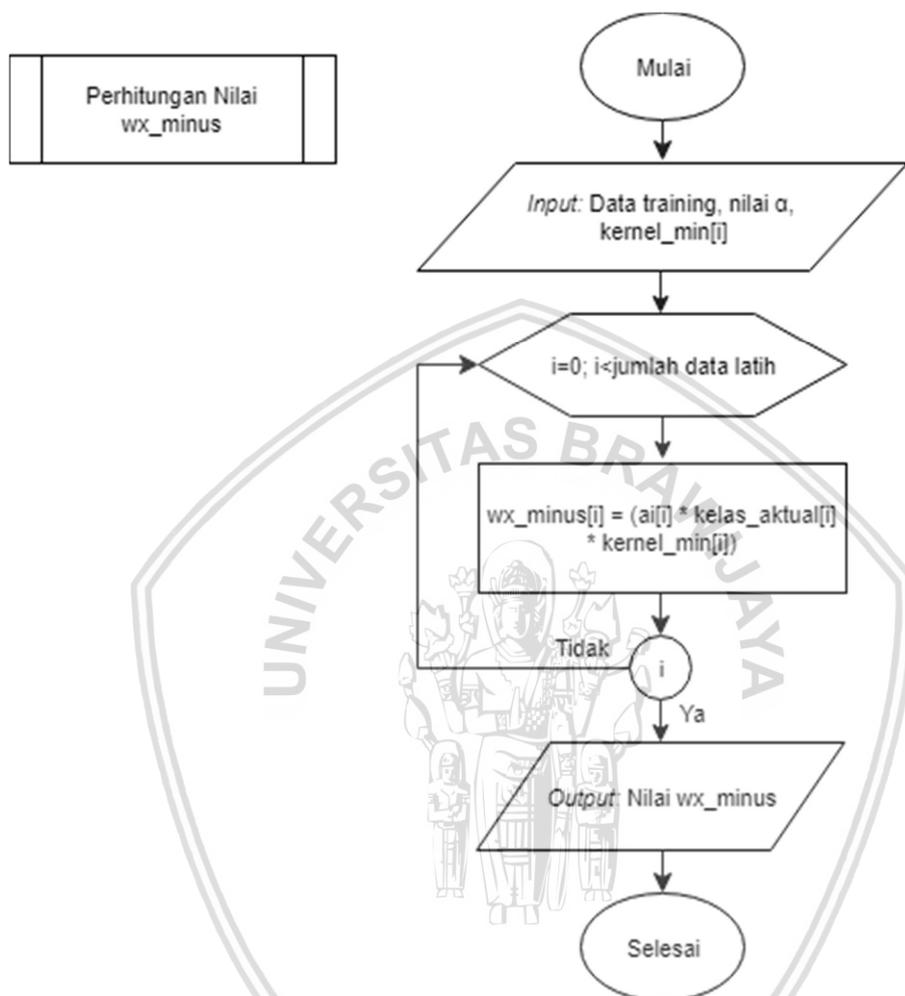
Gambar 4.8 Diagram Alir Proses Perhitungan Nilai wx^+

Penjelasan untuk Diagram Alir Proses Perhitungan Nilai wx^+ adalah sebagai berikut:

1. Memasukkan data *training*, nilai α , kernel plus.
2. Melakukan perulangan terhadap banyaknya data *training*.
3. Menghitung nilai wx^+ .
4. Menampilkan nilai wx^+ .

4.2.9 Alur Proses Perhitungan Nilai w_x^-

Tahapan dalam Proses menghitung nilai w_x^- dapat dilihat seperti Gambar 4.9.



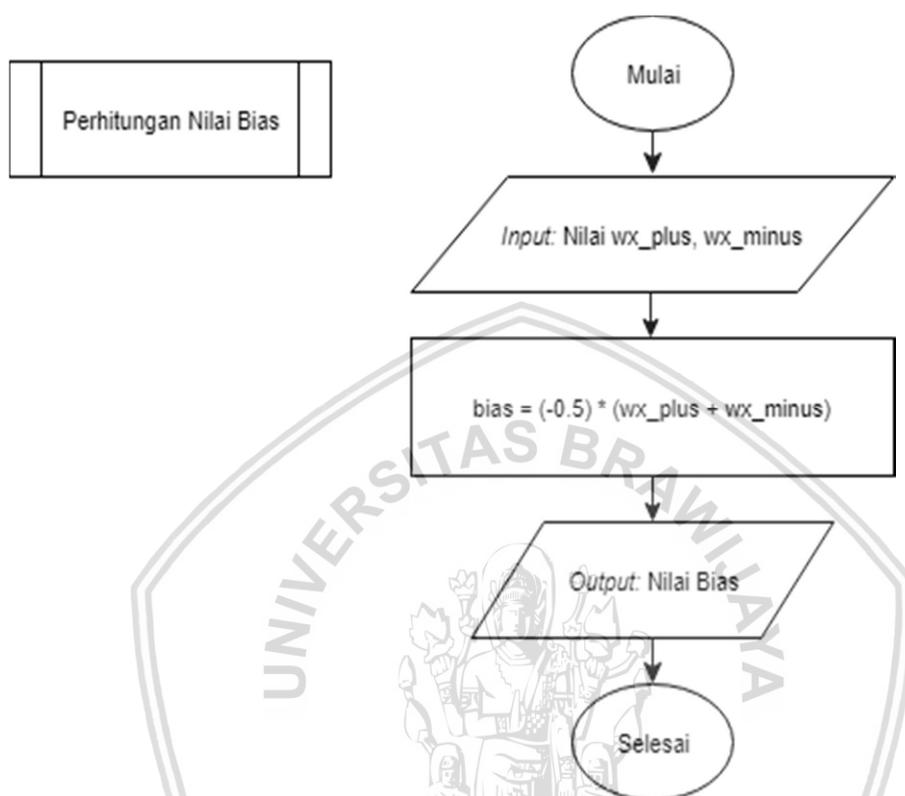
Gambar 4.9 Diagram Alir Proses Perhitungan Nilai w_x^-

Penjelasan untuk Diagram Alir Proses Perhitungan Nilai w_x^- adalah sebagai berikut:

1. Memasukkan data *training*, nilai α , kernel min.
2. Melakukan perulangan terhadap banyaknya data *training*.
3. Menghitung nilai w_x^- .
4. Menampilkan nilai w_x^- .

4.2.10 Alur Proses Perhitungan Nilai Bias

Tahapan dalam Proses menghitung nilai bias dapat dilihat seperti Gambar 4.10.



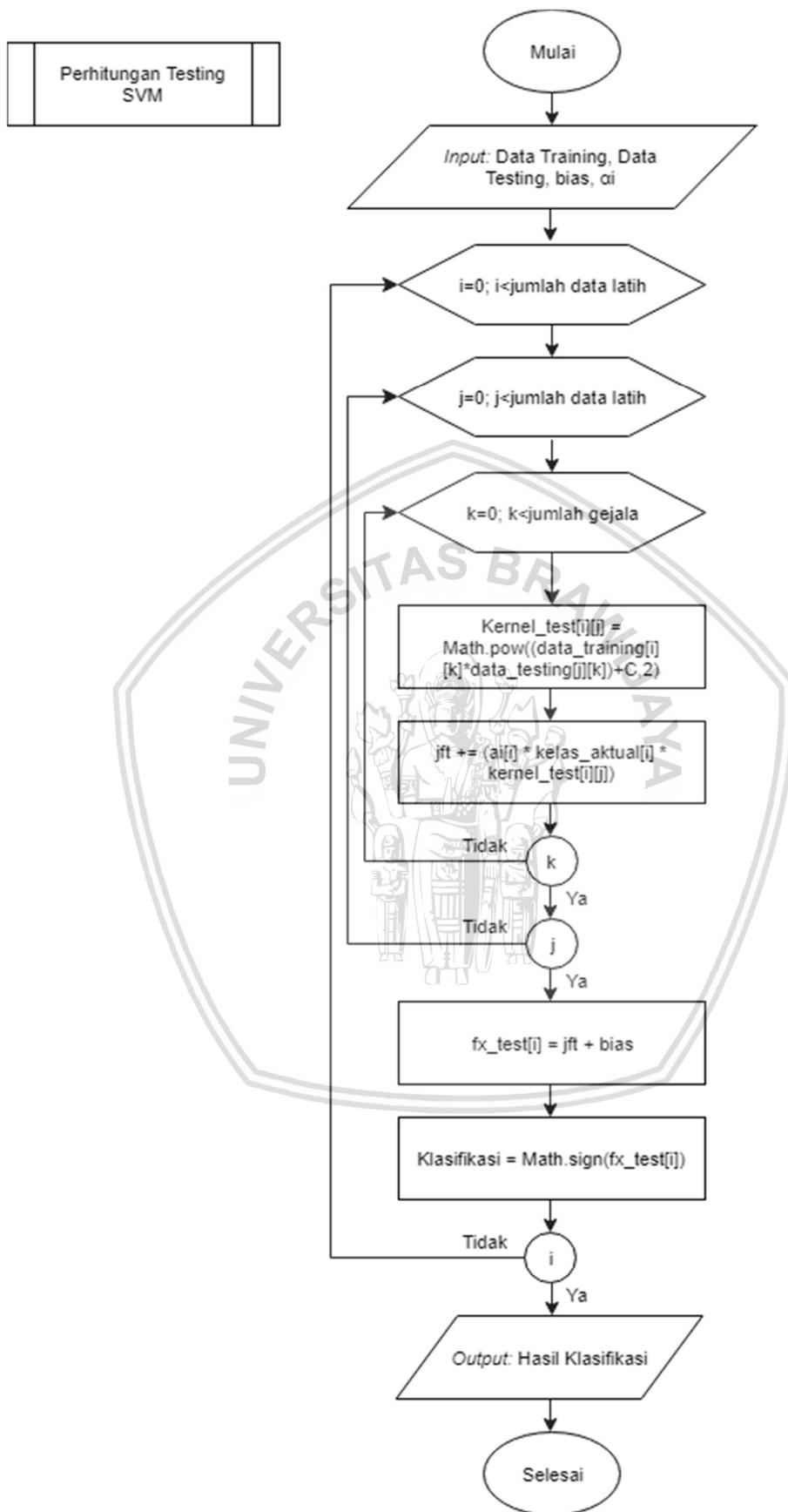
Gambar 4.10 Diagram Alir Proses Perhitungan Nilai Bias

Penjelasan untuk Diagram Alir Proses Perhitungan Nilai Bias adalah sebagai berikut:

1. Memasukkan nilai wx^+ dan wx^- .
2. Menghitung nilai bias dengan menambahkan antara nilai wx^+ dan wx^- .
3. Menampilkan nilai bias.

4.2.11 Alur Proses *Testing* SVM

Setelah melakukan proses *training* pada SVM akan dilakukan juga proses *testing* agar dapat mengklasifikasikan penyakit Skizofrenia Hebefrenik atau Episode Depresi Berat dengan Gejala Psikotik dengan gejala-gejala yang dimiliki oleh data uji. Proses *testing* SVM dapat dilihat seperti Gambar 4.11.



Gambar 4.11 Diagram Alir Proses *Testing* SVM

Penjelasan untuk Diagram Alir Proses Perhitungan *Testing* SVM adalah sebagai berikut:

1. Memasukkan data *training*, data *testing*, nilai bias, dan α .
2. Melakukan perulangan terhadap banyaknya data *training* dan gejala.
3. Menghitung kernel *testing* sesuai dengan Persamaan 2.12.
4. Menghitung nilai jft yang merupakan perkalian antara nilai α , y , dan kernel *testing*.
5. Menghitung nilai f_x sesuai dengan Persamaan 2.13.
6. Melakukan klasifikasi penyakit.
7. Menampilkan hasil klasifikasi penyakit.

4.3 Proses Perhitungan Manualisasi

Proses perhitungan manualisasi pada metode SVM ini terdiri dari proses *sequential training* dan proses *testing* SVM. Proses *sequential training* dilakukan untuk mendapatkan *hyperplane* yang optimal, sedangkan proses *testing* dilakukan untuk mengklasifikasikan data uji berdasarkan gejala yang dimiliki.

Proses manualisasi ini menggunakan 10 data latih yang terdiri dari 5 data penyakit Skizofrenia Hebefrenik dan 5 data dari penyakit Episode Depresi Berat dengan Gejala Psikotik. Data uji yang digunakan sebanyak 2 data yang terdiri dari 1 data penyakit Skizofrenia Hebefrenik dan 1 data penyakit Episode Depresi Berat dengan Gejala Psikotik. Kelas klasifikasi yang digunakan pada data ini adalah 2 kelas klasifikasi yaitu kelas Skizofrenia Hebefrenik yang akan diberi nilai 1 dan kelas Episode Depresi Berat dengan Gejala Psikotik yang akan diberi nilai -1. Representasi dari data latih dan data uji dapat dilihat pada Tabel 4.2 dan Tabel 4.3.

Tabel 4.2 Data Latih

No.	G1	G2	G3	G4	G5	G6	G7	G8	...	G69	G70	Y
1	1	0	0	0	0	0	0	0	...	0	0	1
2	0	0	1	1	0	0	0	0	...	0	0	1
3	0	0	0	1	0	1	0	0	...	0	0	1
4	1	0	1	0	1	0	0	0	...	0	0	1
5	1	0	1	1	1	0	0	0	...	0	0	1
6	0	0	0	0	0	0	0	0	...	0	0	-1
7	0	0	0	0	0	0	0	0	...	0	0	-1
8	0	0	0	1	0	0	0	0	...	0	0	-1
9	0	0	0	0	0	0	0	0	...	0	0	-1
10	0	0	0	1	0	0	0	0	...	0	0	-1

7	1	9	1	9	4	16	64	9	16	25
8	1	4	4	1	16	4	9	64	1	16
9	1	4	1	4	1	9	16	1	36	4
10	1	16	4	25	25	16	25	16	4	121

4.3.2 Perhitungan Manual *Sequential Training*

Proses *sequential training* ini dilakukan untuk memproses data latih yang dimiliki dengan beberapa tahapan agar mendapatkan *hyperplane* yang optimal. Tahapan-tahapan yang ada pada proses *sequential training* dimulai dengan menghitung matriks *Hessian*, kemudian dilanjutkan dengan menghitung nilai E_i , nilai δa_i , dan nilai a_i baru. Berikut merupakan tahapan-tahapan pada proses *sequential training*:

1. Menginisialisasikan nilai-nilai parameter SVM yang akan digunakan pada proses manualisasi dengan nilai a_i awal = 0, C (*Complexity*) = 1, ϵ (*epsilon*) = 0,001, d (*degree*) = 2, λ (*Lambda*) = 0,5, dan γ (*gamma*) = 0,0006.
2. Menghitung Matriks *Hessian*

Tahapan pertama pada proses *sequential training* adalah menghitung matriks *Hessian*. Fungsi yang digunakan dalam menghitung matriks *Hessian* sama seperti Persamaan 2.8. Berikut merupakan contoh untuk perhitungan matriks *Hessian*:

- a. Perhitungan matriks *Hessian* pada baris pertama dan kolom pertama:
 $D_{11} = 1 * 1 (9 + (0,5^2)) = 9,25$
- b. Perhitungan matriks *Hessian* pada baris kedua dan kolom pertama:
 $D_{21} = 1 * 1 (4 + (0,5^2)) = 4,25$

Proses perhitungan ini akan berlanjut hingga baris ke-10 dan kolom ke-10 pada matriks *Hessian*. Sehingga menghasilkan matriks *Hessian* seperti pada Tabel 4.5.

Tabel 4.5 Hasil Perhitungan Manual Matriks *Hessian*

D _{ij}	1	2	3	4	5	6	7	8	9	10
1	9,25	4,25	1,25	4,25	4,25	-1,25	-1,25	-1,25	-1,25	-1,25
2	4,25	49,25	4,25	16,25	25,25	-9,25	-9,25	-4,25	-4,25	-16,25
3	1,25	4,25	9,25	1,25	4,25	-1,25	-1,25	-4,25	-1,25	-4,25
4	4,25	16,25	1,25	144,25	64,25	-25,25	-9,25	-1,25	-4,25	-25,25
5	4,25	25,25	4,25	64,25	169,25	-16,25	-4,25	-16,25	-1,25	-25,25
6	-1,25	-9,25	-1,25	-25,25	-16,25	64,25	16,25	4,25	9,25	16,25
7	-1,25	-9,25	-1,25	-9,25	-4,25	16,25	64,25	9,25	16,25	25,25
8	-1,25	-4,25	-4,25	-1,25	-16,25	4,25	9,25	64,25	1,25	16,25
9	-1,25	-4,25	-1,25	-4,25	-1,25	9,25	16,25	1,25	36,25	4,25
10	-1,25	-16,25	-4,25	-25,25	-25,25	16,25	25,25	16,25	4,25	121,25



3. Menentukan banyaknya iterasi yang akan digunakan sebagai kondisi berhenti dari proses perhitungan E_i , $\delta\alpha$, dan α . Pada perhitungan manualisasi ini digunakan $Itermax = 2$, nilai $\gamma = 0,0006$, dan nilai $C = 1$.

Iterasi 1:

4. Menghitung nilai E_i dengan menggunakan fungsi yang sesuai dengan Persamaan 2.9. Berikut merupakan contoh untuk perhitungan nilai E_i :
 - a. Perhitungan nilai E_i pada baris pertama:

$$E_1 = (0*9,25) + (0*4,25) + (0*1,25) + (0*4,25) + (0*4,25) + (0*-1,25) + (0*-1,25) + (0*-1,25) + (0*-1,25) + (0*-1,25) = 0$$
 - b. Perhitungan nilai E_i pada baris kedua:

$$E_2 = (0*4,25) + (0*49,25) + (0*4,25) + (0*16,25) + (0*25,25) + (0*-9,25) + (0*-9,25) + (0*-4,25) + (0*-4,25) + (0*-16,25) = 0$$

Proses perhitungan nilai E_i ini berlanjut hingga pada baris yang ke-10. Hasil dari perhitungan E_i dapat dilihat pada Tabel 4.6.

Tabel 4.6 Hasil Perhitungan Manual E_i Iterasi 1

i	1	2	3	4	5	6	7	8	9	10	E_i
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0

5. Menghitung $\delta\alpha_i$ sesuai dengan fungsi pada Persamaan 2.10. Berikut merupakan contoh untuk perhitungan $\delta\alpha_i$:

$$\begin{aligned} \delta\alpha_i &= \min \{ \max[0,0006(1-0), -0], 1-0 \} \\ &= \min \{ \max[0,0006(1), -0], 1 \} \\ &= \min \{ \max[0,0006, 1] \} \\ &= \min \{ 0,0006, 1 \} \\ &= 0,0006 \end{aligned}$$

Proses perhitungan ini dilakukan hingga data ke-10 atau data terakhir pada data latih. Sehingga menghasilkan $\delta\alpha_i$ seperti pada Tabel 4.7.



Tabel 4.7 Hasil Perhitungan Manual $\delta\alpha_i$ Iterasi 1

1	2	3	4	5	6	7	8	9	10
0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006

6. Memperbaharui nilai α_i dengan menggunakan fungsi seperti pada Persamaan 2.11. Berikut merupakan contoh perhitungan nilai α_i baru:

$$\alpha_i = 0 + 0,0006 = 0,0006$$

Proses membaharui nilai α_i dilakukan pada seluruh data, sehingga menghasilkan nilai α_i baru seperti pada Tabel 4.8.

Tabel 4.8 Hasil Perhitungan Manual α_i Iterasi 1

1	2	3	4	5	6	7	8	9	10
0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006	0,0006

Iterasi 2:

7. Setelah proses pada iterasi 1 selesai, maka akan dilanjutkan dengan iterasi 2 dimana langkah-langkah pada iterasi 2 sama dengan langkah-langkah pada iterasi 1. Jika pada iterasi 1 nilai α_i awal adalah 0, berbeda dengan nilai α_i pada iterasi 2. Pada iterasi 2 α_i yang digunakan adalah hasil α_i terbaru dari proses iterasi sebelumnya. Hasil E_i pada iterasi kedua dapat dilihat pada Tabel 4.9.

Tabel 4.9 Hasil Perhitungan Manual E_i Iterasi 2

i	1	2	3	4	5	6	7	8	9	10	E_i
1	0,006	0,003	0,001	0,003	0,003	-0,001	-0,001	-0,001	-0,001	-0,001	0,010 2
2	0,003	0,030	0,003	0,010	0,015	-0,006	-0,006	-0,003	-0,003	-0,010	0,033 6
3	0,001	0,003	0,006	0,001	0,003	-0,001	-0,001	-0,003	-0,001	-0,003	0,004 8
4	0,003	0,010	0,001	0,087	0,039	-0,015	-0,006	-0,001	-0,003	-0,015	0,099 0
5	0,003	0,015	0,003	0,039	0,102	-0,010	-0,003	-0,010	-0,001	-0,015	0,122 4
6	-0,001	-0,006	-0,001	-0,015	-0,010	0,039	0,010	0,003	0,006	0,010	0,034 2
7	-0,001	-0,006	-0,001	-0,006	-0,003	0,010	0,039	0,006	0,010	0,015	0,063 6
8	-0,001	-0,003	-0,003	-0,001	-0,010	0,003	0,006	0,039	0,001	0,010	0,040 8



9	-0,001	-0,003	-0,001	-0,003	-0,001	0,006	0,010	0,001	0,022	0,003	0,033
											0
10	-0,001	-0,010	-0,003	-0,015	-0,015	0,010	0,015	0,010	0,003	0,073	0,066
											6

8. Menghitung $\delta\alpha_i$ sesuai dengan fungsi pada Persamaan 2.10. Proses perhitungan ini dilakukan hingga data ke-10 atau data terakhir pada data latih. Sehingga menghasilkan $\delta\alpha_i$ seperti pada Tabel 4.10.

Tabel 4.10 Hasil Perhitungan Manual $\delta\alpha_i$ Iterasi 2

1	2	3	4	5	6	7	8	9	10
0,0005	0,0005	0,0005	0,0005	0,0005	0,0005	0,0005	0,0005	0,0005	0,0005
94	80	97	41	27	79	62	76	80	60

9. Memperbaharui nilai α_i dengan menggunakan fungsi seperti pada Persamaan 2.11. Proses membaharui nilai α_i dilakukan pada seluruh data, sehingga menghasilkan nilai α_i baru seperti pada Tabel 4.11.

Tabel 4.11 Hasil Perhitungan Manual α_i Iterasi 2

1	2	3	4	5	6	7	8	9	10
0,0011	0,0011	0,0011	0,0011	0,0011	0,0011	0,0011	0,001	0,0011	0,001
94	80	97	41	27	79	62	176	80	160

10. Iterasi akan terus berlanjut hingga nilai α_i mencapai konvergen ($|\delta\alpha_i| < \epsilon$) atau iterasi telah mencapai *itermax*. Lebih dari itu maka iterasi akan berhenti.

4.3.3 Perhitungan Manual Nilai w dan b

Proses selanjutnya adalah mencari nilai w (posisi bidang normal) dan nilai b (posisi bidang relatif terhadap koordinat). Untuk nilai w sendiri terbagi menjadi wx^+ dan wx^- . Dimana wx^+ merupakan nilai *support vector machine* untuk kelas positif dan wx^- merupakan nilai *support vector machine* untuk kelas negatif. Sebelum menghitung nilai w dan b, nilai kernel $K(x_i, x^+)$ dan $K(x_i, x^-)$ harus diketahui terlebih dahulu. Berikut merupakan tahapan dari proses menghitung nilai w dan b:

1. Mencari nilai kernel $K(x_i, x^+)$ dan $K(x_i, x^-)$. Nilai kernel $K(x_i, x^+)$ dan $K(x_i, x^-)$ didapat dari matriks kernel *Polynomial of degree 2*. Nilai $K(x_i, x^+)$ didapat dengan mengambil nilai α_i tertinggi dari kelas positif, sedangkan nilai $K(x_i, x^-)$ didapat dengan mengambil nilai α_i tertinggi dari kelas negatif. Dikarenakan nilai α_i tertinggi dari kelas positif adalah 0,001197 yang berada pada indeks ketiga maka nilai $K(x_i, x^+)$ akan diisi dengan nilai kernel *Polynomial of degree 2* pada indeks ketiga. Begitu pula dengan nilai $K(x_i, x^-)$, dikarenakan nilai α_i tertinggi dari kelas negatif adalah 0,001180 yang berada pada indeks sembilan maka



akan diisi dengan nilai kernel *Polynomial of degree 2* pada indeks sembilan. Hasil dari kernel $K(x_i, x^+)$ dan $K(x_i, x^-)$ dapat dilihat seperti pada Tabel 4.12.

Tabel 4.12 Hasil Kernel $K(x_i, x^+)$ dan $K(x_i, x^-)$

i	$K(x_i, x^+)$	$K(x_i, x^-)$
1	1	1
2	4	4
3	9	1
4	1	4
5	4	1
6	1	9
7	1	16
8	4	1
9	1	36
10	4	4

2. Mencari nilai w yaitu wx^+ dan wx^- . Untuk mencari kedua nilai w tersebut dapat menggunakan fungsi yang telah ditentukan sebagai berikut:

$$wx^+ = \sum_{i=1}^i \alpha_i \cdot y_i \cdot K(x_i, x^+) \tag{4.1}$$

$$wx^- = \sum_{i=1}^i \alpha_i \cdot y_i \cdot K(x_i, x^-) \tag{4.2}$$

Berikut merupakan contoh perhitungan nilai wx^+ dan wx^- :

- a. Perhitungan nilai wx^+ adalah sebagai berikut:

$$\begin{aligned} wx^+ &= ((0,001194 \cdot 1 \cdot 1) + (0,001180 \cdot 1 \cdot 4) + (0,001197 \cdot 1 \cdot 9) + \\ &+ (0,001141 \cdot 1 \cdot 1) + (0,001127 \cdot 1 \cdot 4) + (0,001179 \cdot -1 \cdot 1) + (0,001162 \cdot -1 \cdot 1) \\ &+ (0,001176 \cdot -1 \cdot 4) + (0,001180 \cdot -1 \cdot 1) + (0,001160 \cdot -1 \cdot 4)) = 0,00947 \end{aligned}$$

- b. Perhitungan nilai wx^- adalah sebagai berikut:

$$\begin{aligned} wx^- &= ((0,001194 \cdot 1 \cdot 1) + (0,001180 \cdot 1 \cdot 4) + (0,001197 \cdot 1 \cdot 1) + \\ &+ (0,001141 \cdot 1 \cdot 4) + (0,001127 \cdot 1 \cdot 1) + (0,001179 \cdot -1 \cdot 9) + (0,001162 \cdot -1 \cdot 16) \\ &+ (0,001176 \cdot -1 \cdot 1) + (0,001180 \cdot -1 \cdot 36) + (0,001160 \cdot -1 \cdot 4)) = -0,06471 \end{aligned}$$

Hasil dari perhitungan nilai wx^+ dan wx^- dapat dilihat seperti pada Tabel 4.13.



Tabel 4.13 Hasil Perhitungan Nilai wx^+ dan wx^-

i	wx^+	wx^-
1	0,001194	0,001194
2	0,004719	0,004719
3	0,010774	0,001197
4	0,001141	0,004562
5	0,004506	0,001127
6	-0,001179	-0,010615
7	-0,001162	-0,018589
8	-0,004702	-0,001176
9	-0,001180	-0,042487
10	-0,004640	-0,004640
Σ	0,00947	-0,06471

3. Menghitung nilai b dengan menggunakan fungsi seperti yang ada pada Persamaan 2.14. dan berikut merupakan contoh perhitungan nilai b :

$$\begin{aligned} b &= -0,5 (-0,06471 + 0,00947) \\ &= -0,5 (-0,05524) \\ &= 0,027619 \end{aligned}$$

4.3.4 Perhitungan Manual *Testing* SVM

Proses akhir dari perhitungan manualisasi SVM adalah perhitungan *testing* SVM. Data yang digunakan pada proses ini merupakan data *testing* yang akan diklasifikasikan menggunakan metode SVM. Data *testing* yang digunakan pada perhitungan manual sebanyak 2 data yang mewakili setiap kelas yaitu kelas Skizofrenia Hebefrenik yang bernilai 1 dan kelas Episode Depresi Berat dengan Gejala Psikotik yang bernilai -1. Proses *testing* ini diawali dengan tahap menghitung kernel *testing* dengan kernel *polynomial of degree 2*, menghitung nilai $\alpha_i y_i K(x_i, x)$, nilai $f(x)$, dan tahap klasifikasi penyakit. Berikut merupakan tahapan dari perhitungan manual *testing* SVM:

1. Menghitung kernel *testing* dengan rumus *Polynomial of degree 2* yang sama dengan perhitungan kernel *training*. Namun, jika pada perhitungan kernel *training* mengalikan antara data *training* dengan data *training*. Berbeda dengan perhitungan kernel *testing*, pada perhitungan kernel ini mengalikan antara data *testing* dengan data *training*. Berikut merupakan contoh perhitungan kernel *testing*:

- a. Perhitungan kernel untuk data *testing* 1 dengan data *training* 1:

$$K_{\text{test}}(1,1) = (((1*1)+(0*0)+ (1*0)+(1*0)+ (1*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (1*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(1*0)+ (1*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*1)+(0*0)+ (0*0)+(1*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(1*0)+ (0*0)+(0*0)+ (1*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(1*0)+ (0*0)+(0*0)+ (0*0)+(1*0)+ (0*0)+(0*0)+ (0*0)+(0*0)) + 1)^2 = 4$$

- b. Perhitungan kernel untuk data *testing* 2 dengan data *training* 1:

$$K_{\text{test}}(1,2) = (((1*1)+(0*0)+ (0*0)+(1*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(1*0)+ (0*0)+(1*0)+ (1*1)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(1*0)+ (0*0)+(1*0)+ (0*0)+(1*0)+ (1*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (1*0)+(0*0)+ (0*0)+(1*0)+ (0*0)+(0*0)+ (0*0)+(1*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (1*0)+(0*0)+ (0*0)+(1*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (0*0)+(0*0)+ (1*0)+(0*0)+ (0*0)+(1*0)+ (0*0)+(0*0)+ (0*0)+(1*0)) + 1)^2 = 16$$

Proses ini akan berlangsung hingga data *training* ke-10 dan data *testing* ke-2. Sehingga menghasilkan kernel *testing* seperti pada Tabel 4.14.

Tabel 4.14 Hasil Perhitungan Kernel Testing

i	$K(X_{\text{test}1}, X_i)$	$K(X_{\text{test}2}, X_i)$
1	4	9
2	16	16
3	4	4
4	36	25
5	81	36
6	4	25
7	4	25
8	9	16
9	1	9
10	16	36

2. Menghitung nilai $\alpha_i y_i K(x_i, x)$ pada seluruh data *testing*. Berikut merupakan contoh perhitungan nilai $\alpha_i y_i K(x_i, x)$:



- a. Menghitung $\alpha_i y_i K(x_i, x)$ untuk data *testing* 1:

$$\alpha_i y_i K(x_{test1}, x_i) = ((0,001194 * 1 * 4) + (0,001180 * 1 * 16) + (0,001197 * 1 * 4) + (0,001141 * 1 * 36) + (0,001127 * 1 * 81) + (0,001179 * -1 * 4) + (0,001162 * -1 * 4) + (0,001176 * -1 * 9) + (0,001180 * -1 * 1) + (0,001160 * -1 * 16)) = 0,00478$$

- b. Menghitung $\alpha_i y_i K(x_i, x)$ untuk data *testing* 2:

$$\alpha_i y_i K(x_{test2}, x_i) = ((0,001194 * 1 * 9) + (0,001180 * 1 * 16) + (0,001197 * 1 * 4) + (0,001141 * 1 * 25) + (0,001127 * 1 * 36) + (0,001179 * -1 * 25) + (0,001162 * -1 * 25) + (0,001176 * -1 * 16) + (0,001180 * -1 * 9) + (0,001160 * -1 * 36)) = -0,02624$$

Proses menghitung nilai $\alpha_i y_i K(x_i, x)$ ini dapat dilihat seperti pada Tabel 4.15.

Tabel 4.15 Hasil Perhitungan $\alpha_i y_i K(x_i, x)$

i	$\alpha_i y_i K(x_{test1}, x_i)$	$\alpha_i y_i K(x_{test2}, x_i)$
1	0,00478	0,01074
2	0,01888	0,01888
3	0,00479	0,00479
4	0,04106	0,02852
5	0,09125	0,04056
6	-0,00472	-0,02949
7	-0,00465	-0,02905
8	-0,01058	-0,01881
9	-0,00118	-0,01062
10	-0,01856	-0,04176
Σ	0,12107	-0,02624

3. Menghitung nilai $f(x)$ dengan menggunakan fungsi seperti Persamaan 2.13. Dan berikut merupakan contoh perhitungan nilai $f(x)$:

$$f(x_1) = 0,12107 + 0,027619 = 0,14869$$

$$f(x_2) = -0,02624 + 0,027619 = 0,00138$$

4. Melakukan klasifikasi data menggunakan fungsi klasifikasi seperti pada Persamaan 2.15. Jika hasil fungsi klasifikasi menghasilkan 1, maka data *testing* tersebut termasuk ke dalam kelas penyakit Skizofrenia Hebefrenik. Sedangkan jika hasil dari fungsi klasifikasi menghasilkan -1, maka data *testing* tersebut termasuk ke dalam kelas penyakit Episode Depresi Berat dengan Gejala Psikotik. Berikut merupakan contoh penggunaan fungsi klasifikasi:



Fungsi klasifikasi₁ = sign $f(0,14869) = 1$

Fungsi klasifikasi₂ = sign $f(0,00138) = 1$

Hasil fungsi klasifikasi menunjukkan bahwa data *testing* 1 dan data *testing* 2 termasuk pada kelas penyakit Skizofrenia Hebefrenik.

4.3.5 Perhitungan Manual Tingkat Akurasi

Sistem yang telah dibuat membutuhkan evaluasi dalam bentuk nilai akurasi yang digunakan untuk mengukur keberhasilan sebuah sistem. Terdapat berbagai macam cara untuk menghitung keberhasilan sebuah sistem. Salah satunya adalah menggunakan *confusion matrix*, dengan mengitung banyak data yang sesuai (benar) diklasifikasi pada kelas Skizofrenia Hebefrenik maupun kelas Depresi Berat dengan Gejala Psikotik yang akan dibandingkan dengan jumlah seluruh data *testing* untuk mendapatkan tingkat akurasinya. Pada penelitian ini menggunakan fungsi akurasi seperti pada Persamaan 2.16. Tabel 4.16 merupakan tabel *confusion matrix* dan contoh dari perhitungan akurasi sistem sebagai berikut:

Tabel 4.16 Tabel Confusion Matrix

Kelas Aktual	Kelas Klasifikasi	
	Skizofrenia Hebefrenik	Depresi Berat Gejala Psikotik
Skizofrenia Hebefrenik	1	0
Depresi Berat Gejala Psikotik	1	0

$$Accuracy = \frac{1}{2} \times 100\% = 50\%$$

$$Precision^+ = \frac{1}{2} \times 100\% = 50\%$$

$$Recall^+ = \frac{1}{1} \times 100\% = 100\%$$

$$Precision^- = \frac{0}{0} \times 100\% = 0\%$$

$$Recall^- = \frac{0}{1} \times 100\% = 0\%$$

$$Rata-rata\ precision = \frac{50\%+0\%}{2} = 25\%$$

$$Rata-rata\ recall = \frac{100\%+0\%}{2} = 50\%$$

$$F-Measure = \frac{2 \times 25\% \times 50\%}{75\%} \times 100\% = 33,33\%$$



4.4 Perancangan Pengujian Sistem

Pengujian dibutuhkan oleh suatu sistem untuk mengukur validasi dari sistem tersebut. Pengujian yang dilakukan akan menghasilkan tingkat rata-rata akurasi pada setiap skenario dan rata-rata nilai akurasi terbaiklah yang akan dicari. Dalam penelitian ini digunakan 4 pengujian yaitu pengujian terhadap nilai λ (*Lambda*), pengujian terhadap nilai γ (*gamma*), pengujian terhadap nilai *C* (*complexity*), dan pengujian terhadap jumlah *itemax*.

4.4.1 Perancangan Pengujian Terhadap Variabel λ (*Lambda*)

Variabel λ (*Lambda*) merupakan salah satu parameter dalam *support vector machine*. Pengujian variabel λ (*Lambda*) dilakukan untuk menemukan rata-rata nilai akurasi terbaik dan melihat pengaruh variabel ini terhadap nilai akurasi sistem. Skenario yang digunakan pada pengujian ini adalah dengan nilai 0,1, 0,3, 0,5, 1, 1,5, 2, 2,5, 3, 3,5 dan 4. Tabel perancangan pengujian terhadap variabel λ (*Lambda*) dapat dilihat seperti pada Tabel 4.17.

Tabel 4.17 Perancangan Pengujian Terhadap Variabel λ (*Lambda*)

Nilai λ (<i>Lambda</i>)	Percobaan ke- <i>i</i>					Rata-rata Akurasi
	1	2	3	4	5	
0,1						
0,3						
0,5						
1						
1,5						
2						
2,5						
3						
3,5						
4						

4.4.2 Perancangan Pengujian Terhadap Variabel γ (*Gamma*)

Pengujian variabel γ (*gamma*) dilakukan untuk menemukan rata-rata nilai akurasi terbaik dan melihat pengaruh variabel ini terhadap nilai akurasi sistem. Skenario yang digunakan pada pengujian ini adalah dengan nilai 0,00001, 0,0001, 0,001, 0,1, 0,5, 1, 1,5, 2,5, 5 dan 10. Tabel perancangan pengujian terhadap variabel γ (*gamma*) dapat dilihat seperti pada Tabel 4.18.

Tabel 4.18 Perancangan Pengujian Terhadap Variabel γ (*Gamma*)

Nilai γ (<i>Gamma</i>)	Percobaan ke- <i>i</i>					Rata-rata Akurasi
	1	2	3	4	5	
0,00001						
0,0001						
0,001						



0,1						
0,5						
1						
1,5						
2,5						
5						
10						

4.4.3 Perancangan Pengujian Terhadap Variabel C (*Complexity*)

Pengujian variabel C (*complexcity*) dilakukan untuk menemukan rata-rata nilai akurasi terbaik dan melihat pengaruh variabel ini terhadap nilai akurasi sistem. Skenario yang digunakan pada pengujian ini adalah dengan nilai 0,000001, 0,00001, 0,0001, 0,01, 0,1, 1, 10, 50, 100 dan 200. Tabel perancangan pengujian terhadap variabel C (*complexcity*) dapat dilihat seperti pada Tabel 4.19.

Tabel 4.19 Perancangan Pengujian Terhadap Variabel C (*Complexity*)

Nilai C (<i>Complexity</i>)	Percobaan ke- <i>i</i>					Rata-rata Akurasi
	1	2	3	4	5	
0,000001						
0,00001						
0,0001						
0,01						
0,1						
1						
10						
50						
100						
200						

4.4.4 Perancangan Pengujian Terhadap Jumlah *Itermax*

Pengujian terhadap jumlah *itermax* ini dilakukan untuk mengetahui *itermax* yang tepat untuk mendapatkan nilai akurasi yang tinggi serta pengaruh jumlah *itermax* terhadap tingkat akurasi sistem. Skenario yang digunakan pada pengujian ini adalah dengan banyak *itermax* 30, 50, 100, 150, 200, 300, 500, 700, 1000 dan 1500. Tabel perancangan pengujian terhadap jumlah *itermax* dapat dilihat seperti pada Tabel 4.20.

Tabel 4.20 Perancangan Pengujian Terhadap *Itermax*

<i>Itermax</i>	Percobaan ke- <i>i</i>					Rata-rata Akurasi
	1	2	3	4	5	
30						
50						
100						



150						
200						
300						
500						
700						
1000						
1500						

4.5 Evaluasi Hasil

Tahapan evaluasi hasil dilakukan setelah semua tahap-tahap penelitian telah dilakukan. Evaluasi hasil ini berisi pembahasan mengenai hasil kinerja sistem yang akan dianalisis. Setelah itu akan diambil kesimpulan yang berisi kelebihan maupun kekurangan dari penelitian yang telah dilakukan, sehingga dapat menjadi saran untuk penelitian-penelitian selanjutnya.



BAB 5 IMPLEMENTASI

Pada bab implementasi akan menjelaskan cara untuk mengimplementasikan sistem yang akan dibuat berdasarkan perancangan sistem. Pada penelitian ini, tahap implementasi berisi spesifikasi sistem, implementasi algoritme serta implementasi hasil sistem.

5.1 Spesifikasi Sistem

Spesifikasi sistem merupakan suatu penjelasan mengenai spesifikasi perangkat yang digunakan dalam mengimplementasikan sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik. Perangkat yang digunakan terdiri dari perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam mengimplementasikan sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik menggunakan SVM ditunjukkan seperti pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>Processor</i>	Intel Core i5-4210U
<i>Hard Disk Drive</i>	500 GB
RAM	4 GB
<i>System Model</i>	HP Pavilion 14 Notebook PC

5.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam mengimplementasikan sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik menggunakan SVM ditunjukkan seperti pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 10 Pro
Bahasa Pemrograman	Java
IDE	Netbeans 8.2

5.2 Implementasi Algoritme

Implementasi algoritme *Support Vector Machine* (SVM) yang digunakan dalam sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik berisi tahapan-tahapan SVM dalam mengklasifikasi penyakit. Berikut merupakan penjelasan dari setiap algoritme yang ada.

5.2.1 Implementasi Algoritme Perhitungan Kernel SVM

Proses pertama pada SVM adalah menghitung nilai kernel yang digunakan. Pada penelitian ini kernel yang digunakan adalah kernel *Polynomial of Degree 2*. Keluaran yang dihasilkan adalah matriks kernel $r \times r$, dimana r merupakan banyaknya data latih yang digunakan. Kode program algoritme perhitungan kernel *Polynomial of Degree 2* dapat dilihat pada Kode Program 5.1.

```

1 public void KernelPOD2() {
2     System.out.println("Kernel Polynomial of Degree 2 :
3     ");
4     for (int i = 0; i < data_training.length; i++) {
5         for (int j = 0; j < data_training.length; j++)
6         {
7             double jmlkernel = 0;
8             for (int k = 0; k < gejala-1; k++) {
9                 jmlkernel += data_training[i][k] *
10                data_training[j][k];
11            }
12            jmlkernel += C;
13            kernel[i][j] = Math.pow(jmlkernel, 2);
14        }
15        System.out.println("Kernel POD");
16        for (int i = 0; i < kernel.length; i++) {
17            for (int j = 0; j < kernel[0].length; j++) {
18                System.out.print(kernel[i][j] + " | ");
19            }
20            System.out.println("");
21        }
22    }
23 }

```

Kode Program 5.1 Implementasi Kernel *Polynomial of Degree 2*

Penjelasan kode program kernel *Polynomial of Degree 2*:

1. Baris 1 merupakan inialisasi method KernelPOD2.
2. Baris 2 untuk menampilkan kalimat pada tanda petik.
3. Baris 3-4 merupakan kode program untuk melakukan perulangan terhadap data latih ke- i dan data latih ke- j sejumlah banyaknya data latih.
4. Baris 5 menginisialisasikan variabel `jmlkernel` sama dengan nol.
5. Baris 6 merupakan kode program untuk melakukan perulangan terhadap gejala ke- k sejumlah banyaknya gejala.
6. Baris 7-10 merupakan kode program untuk menghitung nilai kernel *Polynomial of Degree 2*.
7. Baris 14-15 merupakan baris kode program untuk melakukan perulangan terhadap kernel ke- i dan kernel ke- j sejumlah banyaknya kernel.

8. Baris 16 merupakan baris kode program untuk menampilkan hasil nilai kernel *Polynomial of Degree 2*.

5.2.2 Implementasi Algoritme Perhitungan Matriks *Hessian*

Setelah mendapat nilai kernel *Polynomial of Degree 2*, langkah selanjutnya yang akan dilakukan adalah menghitung matriks *Hessian*. Menghitung matriks *Hessian* dilakukan untuk menemukan nilai *Hessian* yang akan digunakan dalam perhitungan *sequential training*. Kode program dari matriks *Hessian* dapat dilihat seperti pada Kode Program 5.2.

```

1 public void Matriks_Hessian() {
2     System.out.println("Matriks Hessian : ");
3     for (int i = 0; i < data_training.length; i++) {
4         for (int j = 0; j < data_training.length; j++)
5         {
6             matriks_hessian[i][j] =
7             (data_training[i][gejala-1] * data_training[j][gejala-1]) *
8             (kernel[i][j] + Math.pow(lamda, 2));
9         }
10        for (int i = 0; i < matriks_hessian.length; i++) {
11            for (int j = 0; j < matriks_hessian.length; j++)
12            {
13                System.out.print(matriks_hessian[i][j] + "
14                | ");
15            }
16            System.out.println("");
17        }
18    }
19 }

```

Kode Program 5.2 Implementasi Matriks *Hessian*

Penjelasan kode program matriks *Hessian*:

1. Baris 1 merupakan inialisasi method *Matriks_Hessian*.
2. Baris 2 untuk menampilkan kalimat pada tanda petik.
3. Baris 3-4 merupakan kode program untuk melakukan perulangan terhadap data latih ke-*i* dan data latih ke-*j* sejumlah banyaknya data latih.
4. Baris 5 merupakan kode program untuk menghitung nilai matriks *Hessian*.
5. Baris 8-9 merupakan baris kode program untuk melakukan perulangan terhadap matriks *Hessian* ke-*i* dan matriks *Hessian* ke-*j* sejumlah banyaknya kernel.
6. Baris 10 merupakan baris kode program untuk menampilkan hasil nilai matriks *Hessian*.

5.2.3 Implementasi Algoritme Perhitungan Nilai *Ei*

Perhitungan nilai *Ei* merupakan tahapan pertama untuk iterasi. Menghitung nilai *Ei* digunakan untuk mendapatkan nilai *error* dengan mengalikan nilai α ke-*i* dengan nilai matriks *Hessian* ke-*i* dan menjumlahkannya. Kode program untuk nilai *Ei* dapat dilihat seperti Kode Program 5.3.

```

1 public void Ei() {
2     System.out.println("Menghitung Nilai Ei : ");
3     for (int i = 0; i < data_training.length; i++) {
4         for (int j = 0; j < data_training.length; j++)
5             {
6                 error[i][j] = (ai[i] *
7                 matriks_hessian[i][j]);
8             }
9         for (int i = 0; i < error.length; i++) {
10            for (int j = 0; j < error.length; j++) {
11                System.out.print(error[i][j] + " | ");
12            }
13            System.out.println("");
14        }
15        System.out.println("Total Ei : ");
16        for (int i = 0; i < error.length; i++) {
17            double total_ei = 0;
18            for (int j = 0; j < error.length; j++) {
19                total_ei += error[i][j];
20            }
21            ei[i] = total_ei;
22            System.out.print(ei[i] + " | ");
23        }
24        System.out.println("");
25    }

```

Kode Program 5.3 Implementasi Nilai Ei

Penjelasan kode program nilai Ei:

1. Baris 1 merupakan inialisasi method Ei.
2. Baris 2 untuk menampilkan kalimat pada tanda petik.
3. Baris 3-4 merupakan kode program untuk melakukan perulangan terhadap data latih ke-i dan data latih ke-j sejumlah banyaknya data latih.
4. Baris 5 merupakan kode program untuk menghitung nilai Ei.
5. Baris 8-9 merupakan baris kode program untuk melakukan perulangan terhadap *error* ke-i dan *error* ke-j sejumlah banyaknya *error*.
6. Baris 10 merupakan baris kode program untuk menampilkan hasil nilai Ei.
7. Baris 15 dan 17 merupakan baris kode program untuk melakukan perulangan terhadap *error* ke-i dan *error* ke-j sejumlah banyaknya *error*.
8. Baris 18 merupakan baris kode program untuk menjumlahkan nilai Ei per barisnya.
9. Baris 21 merupakan kode program untuk menginisialisasikan $ei[i] = total_ei$.
10. Baris 22 merupakan baris kode program untuk menampilkan nilai Ei yang telah dijumlahkan.

5.2.4 Implementasi Algoritme Perhitungan Nilai δ_i

Proses selanjutnya setelah proses perhitungan nilai Ei pada *sequential training* adalah menghitung nilai δ_i . Kode program untuk perhitungan nilai δ_i dapat dilihat seperti pada Kode Program 5.4.

1	public void delta() {
2	System.out.println("Menghitung $\delta\alpha_i$: ");
3	for (int i = 0; i < data_training.length; i++) {
4	delta[i] = Math.min(Math.max(gamma * (1 - ei[i]), (-1 * ai[i])), C - ai[i]);
5	}
6	for (int i = 0; i < delta.length; i++) {
7	System.out.print(delta[i] + " ");
8	}
9	System.out.println("");
10	}

Kode Program 5.4 Implementasi Nilai $\delta\alpha_i$

Penjelasan kode program $\delta\alpha_i$:

1. Baris 1 merupakan inisialisasi method delta.
2. Baris 2 untuk menampilkan kalimat pada tanda petik.
3. Baris 3 merupakan kode program untuk melakukan perulangan terhadap data latih ke-i sejumlah banyaknya data latih.
4. Baris 4 merupakan kode program untuk menghitung nilai $\delta\alpha_i$.
5. Baris 6 merupakan baris kode program untuk melakukan perulangan terhadap delta ke-i sejumlah banyaknya delta.
6. Baris 7 merupakan baris kode program untuk menampilkan hasil nilai delta.

5.2.5 Implementasi Algoritme Perhitungan Nilai α_i

Proses selanjutnya pada *sequential training* adalah menghitung nilai α_i baru dengan menjumlahkan nilai α_i lama dengan nilai $\delta\alpha_i$. Kode program untuk perhitungan nilai α_i dapat dilihat pada Kode Program 5.5.

1	public void Hasil_Alpha() {
2	System.out.println("Menghitung Hasil Alpha : ");
3	for (int i = 0; i < data_training.length; i++) {
4	hasil_alpha[i] = ai[i] + delta[i];
5	}
6	for (int i = 0; i < hasil_alpha.length; i++) {
7	System.out.print(hasil_alpha[i] + " ");
8	}
9	System.out.println("");
10	ai = hasil_alpha;
11	}

Kode Program 5.5 Implementasi Nilai α_i

Penjelasan kode program nilai α_i :

1. Baris 1 merupakan inisialisasi method Hasil_Alpha.
2. Baris 2 untuk menampilkan kalimat pada tanda petik.
3. Baris 3 merupakan kode program untuk melakukan perulangan terhadap data latih ke-i sejumlah banyaknya data latih.
4. Baris 4 merupakan kode program untuk menghitung nilai α_i baru.
5. Baris 6 merupakan baris kode program untuk melakukan perulangan terhadap α_i ke-i sejumlah banyaknya α_i .

6. Baris 7 merupakan baris kode program untuk menampilkan hasil nilai α_i .
7. Baris 10 merupakan kode program untuk menginisialisasikan nilai Hasil_Alpha menjadi α_i .

5.2.6 Implementasi Algoritme Perhitungan Nilai wx^+ dan wx^-

Perhitungan nilai wx^+ dan wx^- dilakukan untuk mendapatkan nilai kernel pada indeks tertinggi masing-masing kelas. Nilai wx^+ didapatkan dengan mengambil nilai α tertinggi pada kelas positif (+1) dan nilai wx^- didapatkan dengan mengambil nilai α tertinggi pada kelas negatif (-1). Kode program untuk nilai wx^+ dan wx^- dapat dilihat seperti Kode Program 5.6.

```

1 public void wx() {
2     System.out.println("Menghitung wx+ : ");
3     for (int i = 0; i < data_training.length; i++) {
4         wx_plus[i] = (ai[i] * data_training[i][gejala-
5             1] * kernel_plus[i]);
6     }
7     for (int i = 0; i < data_training.length; i++) {
8         System.out.print(wx_plus[i] + " | ");
9     }
10    System.out.println("");
11    System.out.println("Menghitung wx- : ");
12    for (int i = 0; i < data_training.length; i++) {
13        wx_minus[i] = (ai[i] * data_training[i][gejala-
14            1] * kernel_min[i]);
15    }
16    for (int i = 0; i < data_training.length; i++) {
17        System.out.print(wx_minus[i] + " | ");
18    }
19    System.out.println("");
20    for (int i = 0; i < data_training.length; i++) {
21        wx_positif = wx_positif + wx_plus[i];
22        wx_negatif = wx_negatif + wx_minus[i];
23    }
24    System.out.println("Nilai wx+ : " + wx_positif);
25    System.out.println("Nilai wx- : " + wx_negatif);
26 }

```

Kode Program 5.6 Implementasi Nilai wx^+ dan wx^-

Penjelasan kode program nilai wx^+ dan wx^- :

1. Baris 1 merupakan inialisasi method wx .
2. Baris 2 untuk menampilkan kalimat pada tanda petik.
3. Baris 3 merupakan kode program untuk melakukan perulangan terhadap data latih ke-i sejumlah banyaknya data latih.
4. Baris 4 merupakan kode program untuk menghitung nilai wx^+ .
5. Baris 6 merupakan baris kode program untuk melakukan perulangan terhadap data latih ke-i sejumlah banyaknya data latih.
6. Baris 7 merupakan baris kode program untuk menampilkan hasil nilai wx^+ .
7. Baris 10 untuk menampilkan kalimat pada tanda petik.
8. Baris 11 merupakan kode program untuk melakukan perulangan terhadap data latih ke-i sejumlah banyaknya data latih.
9. Baris 12 merupakan kode program untuk menghitung nilai wx^- .

10. Baris 14 merupakan baris kode program untuk melakukan perulangan terhadap data latih ke-i sejumlah banyaknya data latih.
11. Baris 15 merupakan baris kode program untuk menampilkan hasil nilai wx^+ .
12. Baris 18 merupakan baris kode program untuk melakukan perulangan terhadap data latih ke-i sejumlah banyaknya data latih.
13. Baris 19-20 merupakan kode program untuk menjumlahkan keseluruhan nilai wx^+ dan wx^- .
14. Baris 22-23 merupakan kode program untuk menampilkan nilai wx^+ dan wx^- yang telah dijumlahkan.

5.2.7 Implementasi Algoritme Perhitungan Nilai Bias

Nilai bias didapatkan dengan menambahkan antara nilai wx^+ dan wx^- yang telah diperoleh sebelumnya, kemudian mengalikannya dengan $-\frac{1}{2}$. Kode program perhitungan nilai bias dapat dilihat pada Kode Program 5.7.

```

1 public void nilai_b() {
2     System.out.println("Menghitung nilai Bias : ");
3     bias = (-0.5) * (wx_positif + wx_negatif);
4     System.out.println("Nilai Bias = " + bias);
5 }

```

Kode Program 5.7 Implementasi Nilai Bias

Penjelasan kode program nilai bias:

1. Baris 1 merupakan inisialisasi method nilai_b.
2. Baris 2 untuk menampilkan kalimat pada tanda petik.
3. Baris 3 merupakan kode program untuk menghitung nilai bias.
4. Baris 4 merupakan kode program untuk menampilkan nilai bias yang telah dihitung.

5.2.8 Implementasi Algoritme Perhitungan Nilai Kernel Test

Perhitungan nilai kernel *test* dilakukan untuk mendapatkan nilai kernel pada data *testing*. Kernel *test* yang digunakan sama dengan kernel SVM sebelumnya yaitu kernel *Polynomial of Degree 2*. Dalam mencari nilai kernel *test*, dilakukan perkalian antara data latih dan data uji. Kode program untuk perhitungan kernel *test* dapat dilihat pada Kode Program 5.8.

```

1 public void kernel_test() {
2     System.out.println("Kernel Polynomial of Degdree 2
3     Testing : ");
4     for (int i = 0; i < data_testing.length; i++) {
5         for (int j = 0; j < data_training.length; j++)
6         {
7             double jmlkernel = 0;
8             for (int k = 0; k < gejala-1; k++) {
9                 jmlkernel += data_training[j][k] *
10                data_testing[i][k];
11            }
12            jmlkernel += C;
13            kernel_testing[i][j] = Math.pow(jmlkernel,
14            2);
15        }
16    }
17 }

```

```

11         }
12     }
13     for (int i = 0; i < data_testing.length; i++) {
14         for (int j = 0; j < data_training.length; j++)
15         {
16             System.out.print(kernel_testing[i][j] + " |
17 ");
18         }
19     }
20     System.out.println("");
21 }

```

Kode Program 5.8 Implementasi Nilai Kernel Test

Penjelasan kode program kernel test:

1. Baris 1 merupakan inialisasi method kernel_test.
2. Baris 2 untuk menampilkan kalimat pada tanda petik.
3. Baris 3-4 merupakan kode program untuk melakukan perulangan terhadap data uji ke-i sejumlah banyaknya data uji dan data latih ke-j sejumlah banyaknya data latih.
4. Baris 5 menginisialisasikan variabel jmlkernel sama dengan nol.
5. Baris 6 merupakan kode program untuk melakukan perulangan terhadap gejala ke-k sejumlah banyaknya gejala.
6. Baris 7-10 merupakan kode program untuk menghitung nilai kernel test *Polynomial of Degree 2*.
7. Baris 13-14 merupakan baris kode program untuk melakukan perulangan terhadap data uji ke-i sejumlah banyaknya data uji dan data latih ke-j sejumlah banyaknya data latih.
8. Baris 15 merupakan baris kode program untuk menampilkan hasil nilai kernel test *Polynomial of Degree 2*.

5.2.9 Implementasi Algoritme Perhitungan Nilai $f(x)$

Perhitungan nilai $f(x)$ merupakan proses perhitungan dengan mengalikan nilai a_i dengan nilai kelas aktual (y_i) dan nilai kernel test, kemudian menambahkannya dengan nilai bias. Nilai $f(x)$ akan menentukan sebuah data uji termasuk dalam kelas positif atau kelas negatif. Kode program untuk perhitungan nilai $f(x)$ dapat dilihat pada Kode Program 5.9.

```

1 public void fungsi_testing() {
2     System.out.println("Menghitung nilai fx(test) : ");
3     for (int i = 0; i < data_testing.length; i++) {
4         double jft = 0;
5         for (int j = 0; j < data_training.length; j++)
6         {
7             jft += ai[j] * data_training[j][gejala-1] *
8 kernel_testing[i][j];
9         }
10        fx_test[i] = jft + bias;
11        System.out.println(fx_test[i] + " ");
12    }
13 }

```

Kode Program 5.9 Implementasi Nilai $f(x)$

Penjelasan kode program nilai $f(x)$:

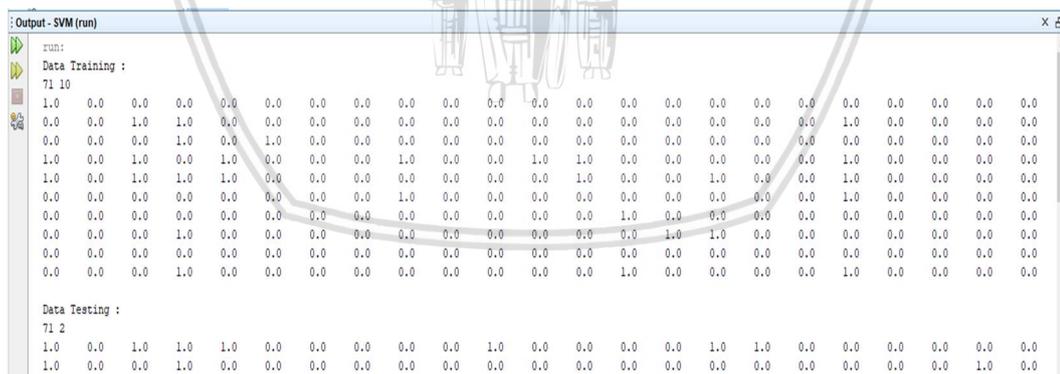
1. Baris 1 merupakan inisialisasi method fungsi_testing.
2. Baris 2 untuk menampilkan kalimat pada tanda petik.
3. Baris 3 merupakan kode program untuk melakukan perulangan terhadap data uji ke-i sejumlah banyaknya data uji.
4. Baris 4 untuk menginisialisasikan variabel jft bertipe double sama dengan nol.
5. Baris 5 merupakan kode program untuk melakukan perulangan terhadap data latih ke-j sejumlah banyaknya data latih.
6. Baris 6-8 merupakan kode program untuk menghitung nilai $f(x)$.
7. Baris 9 merupakan baris kode program untuk menampilkan hasil nilai $f(x)$.

5.3 Implementasi Hasil

Implementasi hasil dilakukan untuk mengetahui hasil dari sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik. Pada penelitian ini, terdapat empat bagian hasil yang ditampilkan yaitu halaman hasil data, halaman hasil *sequential training*, halaman hasil *testing SVM*, dan halaman hasil klasifikasi.

5.3.1 Implementasi Hasil Data

Implementasi hasil data menampilkan data latih dan data uji yang digunakan pada sistem. Data ini berupa gejala-gejala yang dimiliki seorang pasien dengan penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik. Data latih digunakan untuk pembelajaran sistem dan data uji digunakan untuk menguji sistem yang telah diberi pengetahuan. Implementasi hasil data dapat dilihat pada Gambar 5.1.



Gambar 5.1 Implementasi Hasil Data

5.3.2 Implementasi Hasil *Sequential Training*

Implementasi hasil proses *training* terdiri dari menghitung nilai E_i , nilai δ_{ai} , dan nilai α_i sesuai dengan fungsinya masing-masing. Namun sebelum melakukan ketiga perhitungan tersebut, diperlukan untuk menghitung kernel *Polynomial of Degree 2* dan matriks *Hessian* yang digunakan untuk menghitung proses *training*. Tahap menghitung kernel *Polynomial of Degree 2* dan matriks *Hessian* tersebut dapat dilihat pada Gambar 5.2.



```

: Output - SVM (run)
Masukan nilai Gamma : 0,0006
Masukan nilai Lamda : 0,5
Masukan nilai Complexity : 1
Masukan banyaknya iterasi : 2
Kernel Polynomial of Degree 2 :
Kernel POD
9.0 | 4.0 | 1.0 | 4.0 | 4.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
4.0 | 49.0 | 4.0 | 16.0 | 25.0 | 9.0 | 9.0 | 4.0 | 4.0 | 16.0 |
1.0 | 4.0 | 9.0 | 1.0 | 4.0 | 1.0 | 1.0 | 4.0 | 1.0 | 4.0 |
4.0 | 16.0 | 1.0 | 144.0 | 64.0 | 25.0 | 9.0 | 1.0 | 4.0 | 25.0 |
4.0 | 25.0 | 4.0 | 64.0 | 169.0 | 16.0 | 4.0 | 16.0 | 1.0 | 25.0 |
1.0 | 9.0 | 1.0 | 25.0 | 16.0 | 64.0 | 16.0 | 4.0 | 9.0 | 16.0 |
1.0 | 9.0 | 1.0 | 9.0 | 4.0 | 16.0 | 64.0 | 9.0 | 16.0 | 25.0 |
1.0 | 4.0 | 4.0 | 1.0 | 16.0 | 4.0 | 9.0 | 64.0 | 1.0 | 16.0 |
1.0 | 4.0 | 1.0 | 4.0 | 1.0 | 9.0 | 16.0 | 1.0 | 36.0 | 4.0 |
1.0 | 16.0 | 4.0 | 25.0 | 25.0 | 16.0 | 25.0 | 16.0 | 4.0 | 121.0 |
Matriks Hessian :
9.25 | 4.25 | 1.25 | 4.25 | 4.25 | -1.25 | -1.25 | -1.25 | -1.25 | -1.25 |
4.25 | 49.25 | 4.25 | 16.25 | 25.25 | -9.25 | -9.25 | -4.25 | -4.25 | -16.25 |
1.25 | 4.25 | 9.25 | 1.25 | 4.25 | -1.25 | -1.25 | -4.25 | -1.25 | -4.25 |
4.25 | 16.25 | 1.25 | 144.25 | 64.25 | -25.25 | -9.25 | -1.25 | -4.25 | -25.25 |
4.25 | 25.25 | 4.25 | 64.25 | 169.25 | -16.25 | -4.25 | -16.25 | -1.25 | -25.25 |
-1.25 | -9.25 | -1.25 | -25.25 | -16.25 | 64.25 | 16.25 | 4.25 | 9.25 | 16.25 |
-1.25 | -9.25 | -1.25 | -9.25 | -4.25 | 16.25 | 64.25 | 9.25 | 16.25 | 25.25 |
-1.25 | -4.25 | -4.25 | -1.25 | -16.25 | 4.25 | 9.25 | 64.25 | 1.25 | 16.25 |
-1.25 | -4.25 | -1.25 | -4.25 | -1.25 | 9.25 | 16.25 | 1.25 | 36.25 | 4.25 |
-1.25 | -16.25 | -4.25 | -25.25 | -25.25 | 16.25 | 25.25 | 16.25 | 4.25 | 121.25 |
    
```

Gambar 5.2 Implementasi Hasil Kernel dan Matriks Hessian

Dalam melakukan proses *training* pada penelitian ini menggunakan 2 kali iterasi. Berikut merupakan tahapan menghitung nilai E_i , nilai $\delta\alpha_i$, dan nilai α_i yang ditunjukkan pada Gambar 5.3 yang merupakan iterasi 1 dan Gambar 5.4 yang merupakan iterasi 2.

```

: Output - SVM (run)
iterasi 1
Menghitung Nilai Ei :
0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 |
0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 |
0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 |
0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 |
0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 |
0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 |
-0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
-0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
-0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
-0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
-0.0 | -0.0 | -0.0 | -0.0 | -0.0 | -0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
Total Ei :
0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
Menghitung  $\delta\alpha_i$  :
6.0E-4 | 6.0E-4 |
Menghitung Hasil Alpha :
6.0E-4 | 6.0E-4 |
    
```

Gambar 5.3 Implementasi Hasil Sequential Training Iterasi 1

```

: Output - SVM (run)
iterasi 2
Menghitung Nilai Ei :
0.005549999999999999 | 0.002549999999999999 | 7.4999999999999999E-4 | 0.002549999999999999 | 0.002549999999999999 | -7.4999999999999999E-4 | -7.4999999999999999E-4 | -7.4999999999999999E-4 | -7.4999999999999999E-4 | -7.4999999999999999E-4 |
0.002549999999999999 | 0.025499999999999999 | 0.002549999999999999 | 0.00975 | 0.015149999999999999 | -0.005549999999999999 | -0.005549999999999999 | -0.005549999999999999 | -0.002549999999999999 | -0.002549999999999999 |
7.4999999999999999E-4 | 0.002549999999999999 | 0.005549999999999999 | 7.4999999999999999E-4 | 0.002549999999999999 | -7.4999999999999999E-4 | -7.4999999999999999E-4 | -7.4999999999999999E-4 | -7.4999999999999999E-4 | -7.4999999999999999E-4 |
0.002549999999999999 | 0.00975 | 7.4999999999999999E-4 | 0.086549999999999999 | 0.038549999999999999 | -0.015149999999999999 | -0.005549999999999999 | -7.4999999999999999E-4 | -0.002549999999999999 | 0.002549999999999999 |
0.015149999999999999 | 0.002549999999999999 | 0.038549999999999999 | 0.101549999999999999 | -0.00975 | -0.002549999999999999 | -0.00975 | -7.4999999999999999E-4 | -7.4999999999999999E-4 | -0.005549999999999999 |
-7.4999999999999999E-4 | -0.005549999999999999 | -7.4999999999999999E-4 | -0.015149999999999999 | -0.00975 | 0.038549999999999999 | 0.00975 | 0.002549999999999999 | 0.005549999999999999 | -7.4999999999999999E-4 | -0.002549999999999999 |
-7.4999999999999999E-4 | -0.005549999999999999 | -7.4999999999999999E-4 | -0.005549999999999999 | -0.002549999999999999 | 0.00975 | 0.038549999999999999 | 0.005549999999999999 | 0.005549999999999999 | 0.00975 |
-7.4999999999999999E-4 | -0.002549999999999999 | -0.002549999999999999 | -7.4999999999999999E-4 | -0.00975 | 0.002549999999999999 | 0.005549999999999999 | 0.038549999999999999 | 0.005549999999999999 | 7.4999999999999999E-4 |
-7.4999999999999999E-4 | -0.002549999999999999 | -7.4999999999999999E-4 | -0.002549999999999999 | -7.4999999999999999E-4 | 0.005549999999999999 | 0.00975 | 7.4999999999999999E-4 | 0.025499999999999999 | -0.00975 | -0.002549999999999999 | -0.015149999999999999 | -0.015149999999999999 | 0.00975 | 0.015149999999999999 | 0.00975 | 0.002549999999999999 | 0.07275 |
Total Ei :
0.010199999999999999 | 0.0336 | 0.004799999999999999 | 0.099 | 0.122399999999999999 | 0.034199999999999999 | 0.0636 | 0.040799999999999999 | 0.032999999999999999 | 0.065999999999999999 |
Menghitung  $\delta\alpha_i$  :
5.938E-4 | 5.7984E-4 | 5.9711999999999999E-4 | 5.4059999999999999E-4 | 5.265E-4 | 5.7948E-4 | 5.6184E-4 | 5.7511999999999999E-4 | 5.8019999999999999E-4 | 5.6003999999999999E-4 |
Menghitung Hasil Alpha :
0.00119388 | 0.00117984 | 0.00119711999999999999 | 0.00114059999999999999 | 0.00112656 | 0.00117948 | 0.00116184 | 0.00117551999999999999 | 0.00118019999999999999 | 0.00116003999999999999 |
    
```

Gambar 5.4 Implementasi Hasil Sequential Training Iterasi 2

Setelah melakukan perhitungan *sequential training*, akan dilakukan perhitungan nilai wx^+ , nilai wx^- , dan nilai bias yang digunakan untuk menghitung *testing SVM*. Tahapan implementasi tersebut dapat dilihat pada Gambar 5.5.

```

Output - SVM (run)
Index plus :
0 | 0 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
Kernel Plus :
1.0 | 4.0 | 9.0 | 1.0 | 4.0 | 1.0 | 1.0 | 4.0 | 1.0 | 4.0 |
Index minus :
0 | 0 | 0 | 0 | 0 | 1.5 | 1.5 | 1.8 | 1.8 |
Kernel Minus :
1.0 | 4.0 | 1.0 | 4.0 | 1.0 | 9.0 | 16.0 | 1.0 | 36.0 | 4.0 |
Menghitung wx+ :
0.00119388 | 0.00471936 | 0.010774079999999998 | 0.00114059999999999999 | 0.00450624 | -0.00117948 | -0.00116184 | -0.00470207999999999999 | -0.00118019999999999999 | -0.00464015999999999999 |
Menghitung wx- :
0.00119388 | 0.00471936 | 0.00119711999999999999 | 0.00456239999999999994 | 0.00112656 | -0.01061532 | -0.01859944 | -0.00117551999999999998 | -0.04248719999999999999 | -0.00464015999999999999 |
Nilai wx+ : 0.0094704
Nilai wx- : -0.06470831999999999999
Menghitung nilai Bias :
Nilai Bias = 0.02761895999999999999
    
```

Gambar 5.5 Implementasi Hasil Nilai wx^+ , wx^- , dan Bias

5.3.3 Implementasi Hasil *Testing SVM*

Implementasi hasil *testing SVM* terdiri dari perhitungan kernel *test* dan perhitungan nilai $f(x)$ yang digunakan untuk mengklasifikasikan data uji ke dalam kelas positif atau negatif. Implementasi *testing SVM* dapat dilihat pada Gambar 5.6.

```

Kernel Polynomial of Degdree 2 Testing :
4.0 | 16.0 | 4.0 | 36.0 | 81.0 | 4.0 | 4.0 | 9.0 | 1.0 | 16.0 |
9.0 | 16.0 | 4.0 | 25.0 | 36.0 | 25.0 | 25.0 | 16.0 | 9.0 | 36.0 |
Menghitung nilai fx(test) :
0.148687560000000002
0.0013763999999999999964
    
```

Gambar 5.6 Implementasi Hasil *Testing SVM*

5.3.4 Implementasi Hasil Klasifikasi

Implementasi hasil klasifikasi akan menunjukkan suatu data uji termasuk ke dalam penyakit Skizofrenia Hebefrenik atau Episode Depresif Berat dengan Gejala Psikotik. Implementasi hasil hasil klasifikasi dapat dilihat seperti pada Gambar 5.7.

```

Hasil Klasifikasi :
data uji2
fx_test[0] 0.148687560000000002
1. Hasil Klasifikasi pada data test 1 adalah Skizofrenia Hebefrenik
fx_test[1] 0.0013763999999999999964
2. Hasil Klasifikasi pada data test 2 adalah Skizofrenia Hebefrenik
TP = 1.0 ; TN = 0.0 ; FP = 1.0 ; FN = 0.0 ;
Precision Kelas Positif = 50.0
Precision Kelas Negatif = NaN
Recall Kelas Positif = 100.0
Recall Kelas Negatif = 0.0
Accuracy = 50.0
Precision = NaN
Recall = 50.0
F-Measure = NaN
    
```

Gambar 5.7 Implementasi Hasil Klasifikasi

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini berisi hasil pengujian sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik dengan berbagai skenario yang digunakan. Pada penelitian ini, pengujian yang digunakan untuk menguji sistem adalah pengujian terhadap pengujian terhadap variabel λ (*Lambda*), variabel γ (*gamma*), variabel *C* (*complexity*), serta pengujian terhadap jumlah *itermax*. Pada pengujian ini digunakan *K-fold cross validation*, dimana nilai *K* yang digunakan adalah sebanyak 5 pada setiap pengujian.

6.1 Pengujian Terhadap Variabel λ (*Lambda*)

Pengujian variabel λ (*Lambda*) menjelaskan hasil pengujian dengan skenario pengujian yang digunakan serta analisis hasil pengujian yang dilakukan. Pengujian ini bertujuan untuk mengetahui pengaruh variabel λ (*Lambda*) terhadap sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik menggunakan SVM dengan kernel *polynomial of degree 2*.

6.1.1 Skenario Pengujian Terhadap Variabel λ (*Lambda*)

Untuk mengetahui pengaruh variabel λ (*Lambda*) terhadap sistem adalah dengan melihat rata-rata nilai *accuracy*, *precision*, *recall*, serta *f-measure* dari skenario pengujian variabel λ (*Lambda*). Nilai variabel λ (*Lambda*) yang digunakan adalah 0,1, 0,3, 0,5, 1, 1,5, 2, 2,5, 3, 3,5, dan 4. Parameter yang digunakan pada pengujian variabel λ (*Lambda*) adalah dengan perbandingan data latih dan data uji 80%:20%, *gamma* = 0,00001, *C* = 0,01, *itermax* = 30, serta menggunakan fungsi kernel *polynomial of degree 2* dengan percobaan sebanyak 5 kali. Hasil pengujian terhadap variabel λ (*Lambda*) dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian Terhadap Variabel λ (*Lambda*)

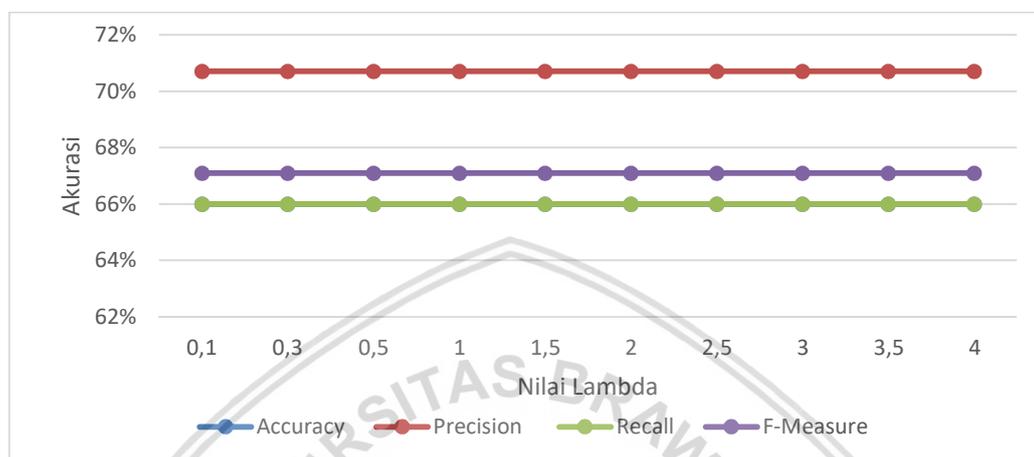
Nilai <i>Lambda</i>		K					Rata-rata
		1	2	3	4	5	
0,1	<i>Accuracy</i>	92,5%	65%	65%	47,5%	60%	66%
	<i>Precision</i>	92,61%	79,41%	79,41%	24,36%	77,78%	70,1%
	<i>Recall</i>	92,5%	65%	65%	47,5%	60%	66%
	<i>F-Measure</i>	92,55%	71,49%	71,49%	32,2%	67,74%	67,1%
0,3	<i>Accuracy</i>	92,5%	65%	65%	47,5%	60%	66%
	<i>Precision</i>	92,61%	79,41%	79,41%	24,36%	77,78%	70,1%
	<i>Recall</i>	92,5%	65%	65%	47,5%	60%	66%
	<i>F-Measure</i>	92,55%	71,49%	71,49%	32,2%	67,74%	67,1%
	<i>Accuracy</i>	92,5%	65%	65%	47,5%	60%	66%

0,5	Precision	92,61%	79,41%	79,41%	24,36%	77,78%	70,1%
	Recall	92,5%	65%	65%	47,5%	60%	66%
	F-Measure	92,55%	71,49%	71,49%	32,2%	67,74%	67,1%
1	Accuracy	92,5%	65%	65%	47,5%	60%	66%
	Precision	92,61%	79,41%	79,41%	24,36%	77,78%	70,1%
	Recall	92,5%	65%	65%	47,5%	60%	66%
1,5	F-Measure	92,55%	71,49%	71,49%	32,2%	67,74%	67,1%
	Accuracy	92,5%	65%	65%	47,5%	60%	66%
	Precision	92,61%	79,41%	79,41%	24,36%	77,78%	70,1%
2	Recall	92,5%	65%	65%	47,5%	60%	66%
	F-Measure	92,55%	71,49%	71,49%	32,2%	67,74%	67,1%
	Accuracy	92,5%	65%	65%	47,5%	60%	66%
2,5	Precision	92,61%	79,41%	79,41%	24,36%	77,78%	70,1%
	Recall	92,5%	65%	65%	47,5%	60%	66%
	F-Measure	92,55%	71,49%	71,49%	32,2%	67,74%	67,1%
3	Accuracy	92,5%	65%	65%	47,5%	60%	66%
	Precision	92,61%	79,41%	79,41%	24,36%	77,78%	70,1%
	Recall	92,5%	65%	65%	47,5%	60%	66%
3,5	F-Measure	92,55%	71,49%	71,49%	32,2%	67,74%	67,1%
	Accuracy	92,5%	65%	65%	47,5%	60%	66%
	Precision	92,61%	79,41%	79,41%	24,36%	77,78%	70,1%
4	Recall	92,5%	65%	65%	47,5%	60%	66%
	F-Measure	92,55%	71,49%	71,49%	32,2%	67,74%	67,1%
	Accuracy	92,5%	65%	65%	47,5%	60%	66%



6.1.2 Analisis Hasil Pengujian Terhadap Variabel λ (*Lambda*)

Berdasarkan pengujian terhadap variabel λ (*Lambda*) yang telah dilakukan, diketahui rata-rata nilai *accuracy* terbaik adalah sebesar 66% yang dimiliki oleh semua skenario pengujian dengan rata-rata nilai *precision* = 70,71%, rata-rata nilai *recall* = 66%, dan rata-rata nilai *f-measure* = 67,1%. Grafik hasil pengujian terhadap variabel λ (*Lambda*) dapat dilihat seperti pada Gambar 6.1.



Gambar 6.1 Grafik Hasil Pengujian Terhadap Variabel λ (*Lambda*)

Berdasarkan pengujian yang telah dilakukan, akan dianalisis hasil dari pengujian tersebut. Gambar 6.1 menunjukkan bahwa rata-rata nilai *accuracy* terbaik adalah 66% yang dimiliki oleh setiap skenario pengujian dengan rata-rata nilai *precision* = 70,71%, rata-rata nilai *recall* = 66%, dan nilai *f-measure* = 67,1%. Setiap nilai variabel λ (*Lambda*) memiliki tingkat *accuracy*, *precision*, *recall*, serta *f-measure* yang sama. Hal ini dikarenakan nilai variabel λ (*Lambda*) hanya digunakan dalam perhitungan matriks *Hessian* saja sehingga membuat nilai variabel λ (*Lambda*) tidak berpengaruh besar pada tingkat *accuracy*, *precision*, *recall*, serta *f-measure* yang dimiliki sistem.

6.2 Pengujian Terhadap Variabel γ (*Gamma*)

Pengujian variabel γ (*gamma*) bertujuan untuk mengetahui pengaruh variabel γ (*gamma*) pada sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik menggunakan SVM dengan kernel *polynomial of degree 2*. Pengujian ini berisi hasil pengujian dengan skenario yang digunakan serta analisis hasil pengujian yang dilakukan.

6.2.1 Skenario Pengujian Terhadap Variabel γ (*Gamma*)

Untuk mengetahui pengaruh variabel γ (*gamma*) terhadap sistem adalah dengan melihat rata-rata nilai *accuracy*, *precision*, *recall*, serta *f-measure* dari setiap skenario yang digunakan. Pada pengujian variabel γ (*gamma*) ini digunakan parameter dengan perbandingan data latih dan data uji 80%:20%, $\lambda = 0,1$, $C = 0,01$, $itermax = 30$, serta menggunakan fungsi kernel *polynomial of degree 2* dengan percobaan sebanyak 5 kali. Nilai variabel γ (*gamma*) yang digunakan



adalah 0,00001, 0,0001, 0,001, 0,1, 0,5, 1, 1,5, 2,5, 5, dan 10. Hasil pengujian terhadap variabel γ (*gamma*) dapat dilihat seperti pada Tabel 6.2.

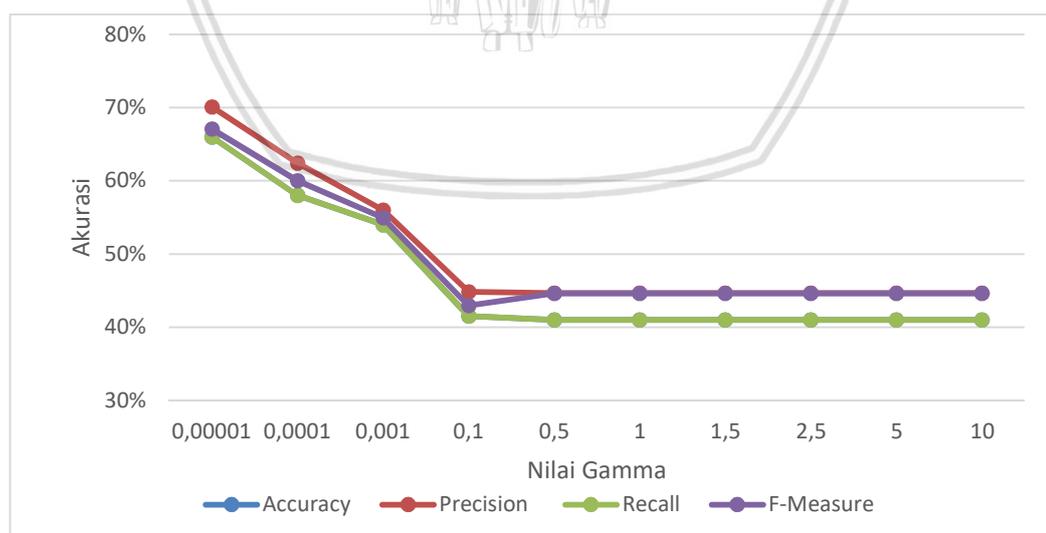
Tabel 6.2 Hasil Pengujian Terhadap Variabel γ (Gamma)

Nilai <i>Gamma</i>		K					Rata-rata
		1	2	3	4	5	
0,0000 1	<i>Accuracy</i>	92,5%	65%	65%	47,5%	60%	66%
	<i>Precision</i>	92,61%	79,41%	79,41%	24,36%	77,78%	70,1%
	<i>Recall</i>	92,5%	65%	65%	47,5%	60%	66%
	<i>F-Measure</i>	92,55%	71,49%	71,49%	32,2%	67,74%	67,1%
0,0001	<i>Accuracy</i>	82,5%	40%	62,5%	55%	50%	58%
	<i>Precision</i>	83,25%	39,58%	62,79%	76,32%	50%	62,4%
	<i>Recall</i>	82,5%	40%	62,5%	55%	50%	58%
	<i>F-Measure</i>	82,87%	39,79%	62,64%	63,93%	50%	60%
0,001	<i>Accuracy</i>	47,5%	35%	60%	75%	52,5%	54%
	<i>Precision</i>	47,33%	34,85%	60,1%	83,33%	54,33%	56%
	<i>Recall</i>	47,5%	35%	60%	75%	52,5%	54%
	<i>F-Measure</i>	47,42%	34,92%	60,05%	78,95%	53,4%	54,95%
0,1	<i>Accuracy</i>	17,5%	27,5%	47,5%	62,5%	52,5%	41,5%
	<i>Precision</i>	17,42%	27,44%	47,49%	78,57%	53,13%	44,83%
	<i>Recall</i>	17,5%	27,5%	47,5%	62,5%	52,5%	41,5%
	<i>F-Measure</i>	17,46%	27,47%	47,5%	69,62%	52,82%	42,97%
0,5	<i>Accuracy</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>Precision</i>	17,42%	27,44%	47,49%	77,78%	53,13%	44,65%
	<i>Recall</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>F-Measure</i>	17,46%	27,47%	47,5%	67,74%	52,82%	44,6%
1	<i>Accuracy</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>Precision</i>	17,42%	27,44%	47,49%	77,78%	53,13%	44,65%
	<i>Recall</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>F-Measure</i>	17,46%	27,47%	47,5%	67,74%	52,82%	44,6%
1,5	<i>Accuracy</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>Precision</i>	17,42%	27,44%	47,49%	77,78%	53,13%	44,65%

	<i>Recall</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>F-Measure</i>	17,46%	27,47%	47,5%	67,74%	52,82%	44,6%
2,5	<i>Accuracy</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>Precision</i>	17,42%	27,44%	47,49%	77,78%	53,13%	44,65%
	<i>Recall</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>F-Measure</i>	17,46%	27,47%	47,5%	67,74%	52,82%	44,6%
5	<i>Accuracy</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>Precision</i>	17,42%	27,44%	47,49%	77,78%	53,13%	44,65%
	<i>Recall</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>F-Measure</i>	17,46%	27,47%	47,5%	67,74%	52,82%	44,6%
10	<i>Accuracy</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>Precision</i>	17,42%	27,44%	47,49%	77,78%	53,13%	44,65%
	<i>Recall</i>	17,5%	27,5%	47,5%	60%	52,5%	41%
	<i>F-Measure</i>	17,46%	27,47%	47,5%	67,74%	52,82%	44,6%

6.2.2 Analisis Hasil Pengujian Terhadap Variabel γ (*Gamma*)

Berdasarkan hasil pengujian terhadap variabel γ (*gamma*) didapatkan rata-rata nilai *accuracy* terbaik dari sistem sebesar 66% yang dimiliki oleh nilai γ (*gamma*) sebesar 0,00001 dengan rata-rata nilai *precision* = 70,71%, rata-rata nilai *recall* = 66%, dan rata-rata nilai *f-measure* = 67,1%. Grafik hasil pengujian terhadap variabel γ (*gamma*) dapat dilihat seperti pada Gambar 6.2.



Gambar 6.2 Grafik Hasil Pengujian Terhadap Variabel γ (*Gamma*)

Analisis pengujian terhadap variabel γ (*gamma*) dilakukan untuk mengetahui pengaruh nilai variabel γ (*gamma*) terhadap tingkat akurasi sistem.



Pada Gambar 6.2 terlihat bahwa variabel γ (*gamma*) berpengaruh terhadap tingkat akurasi sistem. Berdasarkan grafik hasil pengujian variabel γ (*gamma*) didapatkan rata-rata nilai *accuracy* terbaik adalah 66% yang dimiliki oleh nilai *gamma* sebesar 0,00001 dengan rata-rata nilai *precision* = 70,71%, rata-rata nilai *recall* = 66%, dan rata-rata nilai *f-measure* = 67,1%. Sedangkan rata-rata nilai *accuracy* terendah adalah 41% yang dimiliki nilai *gamma* sebesar 0,5, 1, 1,5, 2,5, 5, dan 10 dengan rata-rata nilai *precision* = 44,65%, rata-rata nilai *recall* = 41%, dan rata-rata nilai *f-measure* = 42,6%. Sehingga dapat disimpulkan bahwa semakin besar nilai *gamma* yang digunakan pada sistem maka tingkat akurasi sistem akan berkurang, hal ini terlihat pada grafik hasil pengujian dengan nilai variabel γ (*gamma*) antara 0,00001 sampai dengan 10 mengalami penurunan. Semakin tinggi nilai γ (*gamma*) maka nilai *learning rate* atau laju pembelajaran pun akan semakin cepat, sehingga menyebabkan ketelitian sistem berkurang dan akurasi sistem menurun (Kurniawaty, Cholissodin, & Adikara, 2018). Namun sebaliknya, jika nilai γ (*gamma*) semakin rendah akan menyebabkan nilai *learning rate* atau laju pembelajaran semakin lambat dan ketelitian sistem semakin besar.

6.3 Pengujian Terhadap Variabel C (*Complexity*)

Pengujian terhadap variabel C (*complexity*) berisi hasil pengujian dengan skenario yang digunakan serta analisis hasil pengujian yang dilakukan. Pengujian terhadap variabel C (*complexity*) ini bertujuan untuk mengetahui pengaruh variabel C (*complexity*) terhadap sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik menggunakan SVM dengan kernel *polynomial of degree 2*.

6.3.1 Skenario Pengujian Terhadap Variabel C (*Complexity*)

Untuk mengetahui pengaruh variabel C (*complexity*) terhadap sistem adalah dengan melihat rata-rata nilai *accuracy*, *precision*, *recall*, serta *f-measure* dari setiap skenario pengujian yang dilakukan. Parameter yang digunakan pada pengujian variabel C (*complexity*) adalah dengan perbandingan data latih dan data uji sebanyak 80%:20%, *gamma* = 0,00001, *Lambda* = 0,1, *itermax* = 30, serta menggunakan fungsi kernel *polynomial of degree 2* dengan percobaan sebanyak 5 kali. Nilai variabel C (*complexity*) yang digunakan adalah 0,000001, 0,00001, 0,0001, 0,01, 0,1, 1, 10, 50, 100, dan 200. Hasil pengujian terhadap variabel C (*complexity*) dapat dilihat seperti pada Tabel 6.3.

Tabel 6.3 Pengujian Terhadap Variabel C (*Complexity*)

Nilai <i>Complexity</i>		K					Rata-rata
		1	2	3	4	5	
0,000001	<i>Accuracy</i>	92,5%	92,5%	70%	42,5%	77,5%	75%
	<i>Precision</i>	92,61%	93,48%	70,2%	32,86%	78,13%	73,46%
	<i>Recall</i>	92,5%	92,5%	70%	42,5%	77,5%	75%



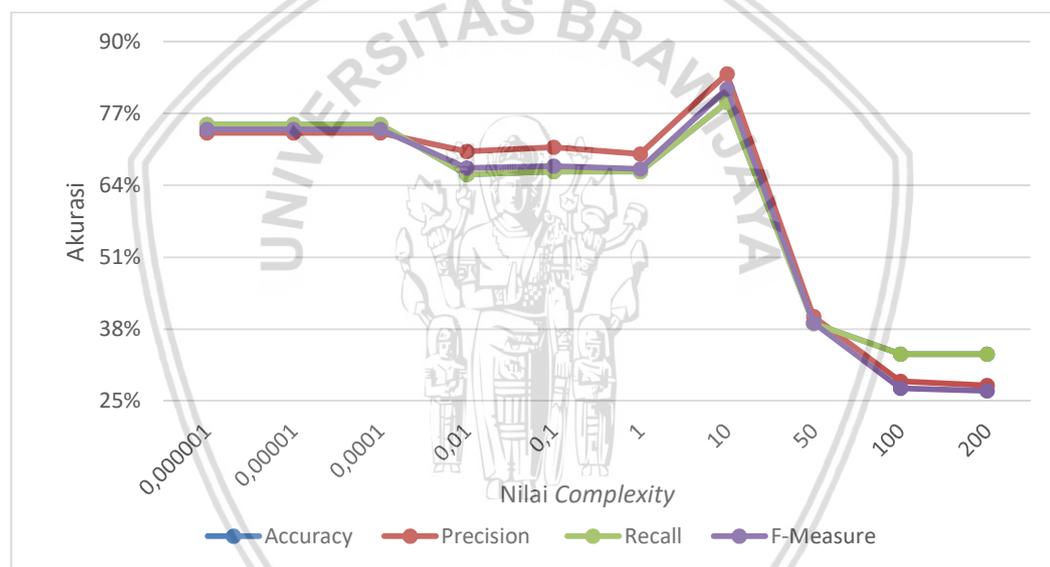
	<i>F-Measure</i>	92,55%	92,99%	70,1%	37,06%	77,82%	74,1%
0,0000 1	<i>Accuracy</i>	92,5%	92,5%	70%	42,5%	77,5%	75%
	<i>Precision</i>	92,61%	93,48%	70,2%	32,86%	78,13%	73,46%
	<i>Recall</i>	92,5%	92,5%	70%	42,5%	77,5%	75%
	<i>F-Measure</i>	92,55%	92,99%	70,1%	37,06%	77,82%	74,1%
0,0001	<i>Accuracy</i>	92,5%	92,5%	70%	42,5%	77,5%	75%
	<i>Precision</i>	92,61%	93,48%	70,2%	32,86%	78,13%	73,46%
	<i>Recall</i>	92,5%	92,5%	70%	42,5%	77,5%	75%
	<i>F-Measure</i>	92,55%	92,99%	70,1%	37,06%	77,82%	74,1%
0,01	<i>Accuracy</i>	92,5%	65%	65%	47,5%	60%	66%
	<i>Precision</i>	92,61%	79,41%	79,41%	24,36%	77,78%	70,1%
	<i>Recall</i>	92,5%	65%	65%	47,5%	60%	66%
	<i>F-Measure</i>	92,55%	71,49%	71,49%	32,2%	67,74%	67,1%
0,1	<i>Accuracy</i>	92,5%	65%	67,5%	47,5%	60%	66,5%
	<i>Precision</i>	92,61%	79,41%	80,3%	24,36%	77,78%	70,9%
	<i>Recall</i>	92,5%	65%	67,5%	47,5%	60%	66,5%
	<i>F-Measure</i>	92,55%	71,49%	73,35%	32,2%	67,74%	67,47%
1	<i>Accuracy</i>	92,5%	65%	65%	47,5%	62,5%	66,5%
	<i>Precision</i>	92,61%	79,41%	73,44%	24,36%	78,57%	69,68%
	<i>Recall</i>	92,5%	65%	65%	47,5%	62,5%	66,5%
	<i>F-Measure</i>	92,55%	71,49%	68,96%	32,2%	69,62%	66,96%
10	<i>Accuracy</i>	92,5%	75%	72,5%	85%	70%	79%
	<i>Precision</i>	92,61%	83,33%	78,21%	85,35%	81,25%	84,15%
	<i>Recall</i>	92,5%	75%	72,5%	85%	70%	79%
	<i>F-Measure</i>	92,55%	78,95%	75,25%	85,18%	75,21%	81,43%
50	<i>Accuracy</i>	27,5%	30%	42,5%	55%	40%	39%
	<i>Precision</i>	26,98%	26,2%	41,45%	76,32%	30,39%	40,27%
	<i>Recall</i>	27,5%	30%	42,5%	55%	40%	39%
	<i>F-Measure</i>	27,24%	27,97%	41,97%	63,93%	34,54%	39,13%
100	<i>Accuracy</i>	12,5%	30%	37,5%	47,5%	40%	33,5%
	<i>Precision</i>	11,64%	18,75%	32,08%	50%	30,39%	28,57%



	<i>Recall</i>	12,5%	30%	37,5%	47,5%	40%	33,5%
	<i>F-Measure</i>	12,05%	23,1%	34,58%	32,2%	34,54%	27,29%
200	<i>Accuracy</i>	15%	30%	35%	47,5%	40%	33,5%
	<i>Precision</i>	13,54%	18,75%	26,56%	50%	30,39%	27,85%
	<i>Recall</i>	15%	30%	35%	47,5%	40%	33,5%
	<i>F-Measure</i>	14,23%	23,1%	30,2%	32,2%	34,54%	26,85%

6.3.2 Analisis Hasil Pengujian Terhadap Variabel C (*Complexity*)

Berdasarkan hasil pengujian terhadap variabel C (*complexity*), didapatkan nilai rata-rata *accuracy* terbaik adalah 79% yang dimiliki oleh nilai C (*complexity*) sebesar 10 dengan rata-rata nilai *precision* = 84,15%, rata-rata nilai *recall* = 70%, dan rata-rata nilai *f-measure* = 81,43%. Garfik hasil pengujian terhadap variabel C (*complexity*) dapat dilihat seperti pada Gambar 6.3.



Gambar 6.3 Grafik Pengujian Terhadap Variabel C (*Complexity*)

Analisis pengujian variabel C (*complexity*) dilakukan untuk mengetahui pengaruh nilai C (*complexity*) terhadap rata-rata nilai *accuracy*, *precision*, *recall*, serta *f-measure*. Gambar 6.3 menunjukkan bahwa variabel C (*complexity*) berpengaruh besar pada sistem. Rata-rata nilai *accuracy* terbaik adalah 79% yang dimiliki oleh nilai C (*complexity*) 10. Setelah itu, terlihat grafik mengalami penurunan yang drastis hingga mencapai rata-rata nilai *accuracy* 33,5% pada nilai C (*complexity*) sebesar 100 dan 200 dengan rata-rata nilai *precision* = 28,57%, rata-rata nilai *recall* = 33,5%, dan rata-rata nilai *f-measure* = 27,29%. Hal ini dikarenakan variabel C (*complexity*) bertujuan untuk meminimalkan nilai *error* pada perhitungan nilai bobot dan bias yang terdapat pada proses *sequential training*. Untuk mendapat nilai *error* yang kecil maka digunakan nilai C (*complexity*) yang kecil pula, karena jika nilai C (*complexity*) mendekati 0 maka lebar margin pada *hyperplane* akan maksimal (Puspitasari, Ratnawati, & Widodo, 2018). Semakin

besar nilai C (*complexity*) juga berpengaruh terhadap waktu komputasi yang lama pada proses *sequential training*.

6.4 Pengujian Terhadap *Itermax*

Pengujian terhadap *itermax* berisi hasil pengujian sistem dengan skenario yang digunakan serta analisis hasil sistem yang digunakan. Pengujian ini bertujuan untuk mengetahui pengaruh *itermax* terhadap rata-rata akurasi sistem klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik menggunakan SVM dengan kernel *polynomial of degree 2*.

6.4.1 Skenario Pengujian Terhadap Jumlah *Itermax*

Untuk mengetahui pengaruh *itermax* terhadap sistem dapat dilihat dari rata-rata nilai *accuracy*, *precision*, *recall*, serta *f-measure* yang dihasilkan pada setiap skenario. Pada pengujian ini, parameter yang digunakan adalah dengan perbandingan data latih dan data uji sebanyak 80%:20%, $\gamma = 0,00001$, $\Lambda = 0,1$, $C = 0,01$, serta menggunakan fungsi kernel *polynomial of degree 2* dengan iterasi sebanyak 5 kali. Jumlah *itermax* yang digunakan pada pengujian ini adalah 30, 50, 100, 150, 200, 300, 500, 700, 1000, dan 1500. Hasil pengujian terhadap jumlah *itermax* dapat dilihat seperti pada Tabel 6.4.

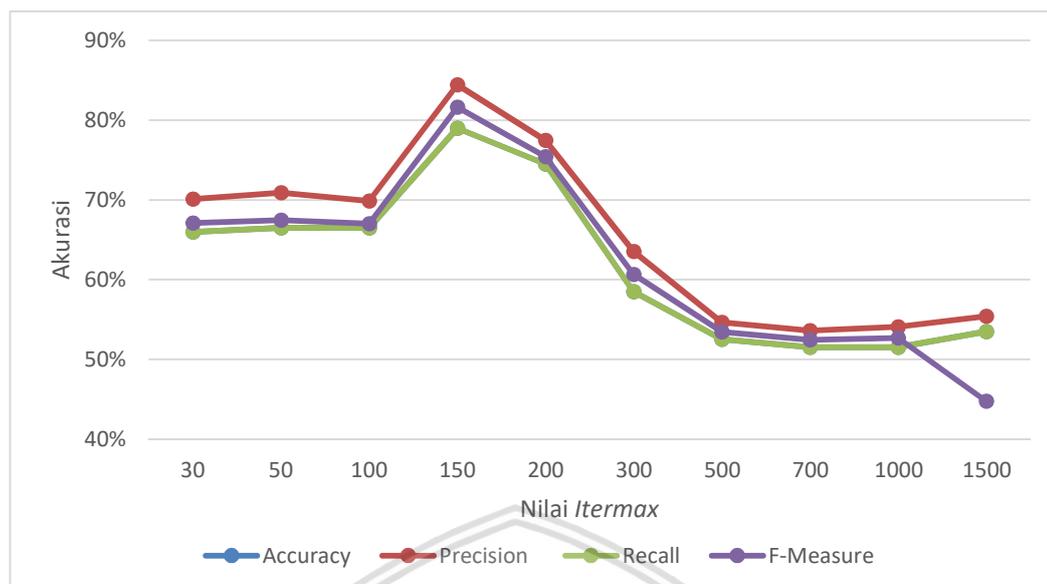
Tabel 6.4 Hasil Pengujian Terhadap Jumlah *Itermax*

<i>Itermax</i>		K					Rata-rata
		1	2	3	4	5	
30	<i>Accuracy</i>	92,5%	65%	65%	47,5%	60%	66%
	<i>Precision</i>	92,61%	79,41%	79,41%	24,36%	77,78%	70,1%
	<i>Recall</i>	92,5%	65%	65%	47,5%	60%	66%
	<i>F-Measure</i>	92,55%	71,49%	71,49%	32,2%	67,74%	67,1%
50	<i>Accuracy</i>	92,5%	65%	67,5%	47,5%	60%	66,5%
	<i>Precision</i>	92,61%	79,41%	80,3%	24,36%	77,78%	70,89%
	<i>Recall</i>	92,5%	65%	67,5%	47,5%	60%	66,5%
	<i>F-Measure</i>	92,55%	71,49%	73,35%	32,2%	67,74%	67,47%
100	<i>Accuracy</i>	92,5%	65%	67,5%	47,5%	60%	66,5%
	<i>Precision</i>	92,61%	79,41%	75,1%	24,36%	77,78%	69,85%
	<i>Recall</i>	92,5%	65%	67,5%	47,5%	60%	66,5%
	<i>F-Measure</i>	92,55%	71,49%	71,1%	32,2%	67,74%	67,02%
150	<i>Accuracy</i>	90%	72,5%	70%	87,5%	75%	79%
	<i>Precision</i>	90,04%	82,26%	76,67%	90%	83,33%	84,46%

	<i>Recall</i>	90%	72,5%	70%	87,5%	75%	79%
	<i>F-Measure</i>	90,2%	77,1%	73,18%	88,73%	78,95%	81,63%
200	<i>Accuracy</i>	92,5%	57,5%	80%	80%	62,5%	74,5%
	<i>Precision</i>	92,61%	57,67%	85,71%	85,71%	65,67%	77,47%
	<i>Recall</i>	92,5%	57,5%	80%	80%	62,5%	74,5%
	<i>F-Measure</i>	92,55%	57,59%	80,15%	82,76%	64,05%	75,42%
300	<i>Accuracy</i>	82,5%	40%	62,5%	55%	52,5%	58,5%
	<i>Precision</i>	83,25%	39,58%	62,79%	76,32%	55,71%	63,53%
	<i>Recall</i>	82,5%	40%	62,5%	55%	52,5%	58,5%
	<i>F-Measure</i>	82,87%	39,79%	62,64%	63,93%	54,06%	60,66%
500	<i>Accuracy</i>	50%	42,5%	55%	67,5%	47,5%	52,5%
	<i>Precision</i>	50%	42%	55,21%	80,3%	45,67%	54,64%
	<i>Recall</i>	50%	42,5%	55%	67,5%	47,5%	52,5%
	<i>F-Measure</i>	50%	42,25%	55,1%	73,35%	46,57%	53,45%
700	<i>Accuracy</i>	40%	42,5%	57,5%	67,5%	50%	51,5%
	<i>Precision</i>	38,09%	42%	57,67%	80,3%	50%	53,61%
	<i>Recall</i>	40%	42,5%	57,5%	67,5%	50%	51,5%
	<i>F-Measure</i>	39,02%	42,25%	57,59%	73,35%	50%	52,44%
1000	<i>Accuracy</i>	45%	35%	57,5%	67,5%	52,5%	51,5%
	<i>Precision</i>	44,5%	34,85%	57,67%	80,3%	53,13%	54,09%
	<i>Recall</i>	45%	35%	57,5%	67,5%	52,5%	51,5%
	<i>F-Measure</i>	44,75%	34,92%	57,59%	73,35%	52,81%	52,68%
1500	<i>Accuracy</i>	45%	35%	60%	75%	52,5%	53,5%
	<i>Precision</i>	44,5%	34,85%	60,1%	83,33%	54,33%	55,42%
	<i>Recall</i>	45%	35%	60%	75%	52,5%	53,5%
	<i>F-Measure</i>	44,75%	34,92%	60,05%	78,95%	53,4%	54,41%

6.4.2 Analisis Hasil Pengujian Terhadap Jumlah *Itermax*

Hasil pengujian terhadap jumlah *itermax* menunjukkan rata-rata nilai *accuracy* terbaik sistem adalah 79% pada *itermax* sebanyak 150 dengan rata-rata nilai *precision* = 84,46%, rata-rata nilai *recall* = 79%, dan rata-rata nilai *f-measure* = 81,63%. Grafik pengujian terhadap jumlah *itermax* dapat dilihat seperti pada Gambar 6.4.



Gambar 6.4 Pengujian Terhadap Jumlah Itermax

Analisis pengujian terhadap jumlah *itermax* dilakukan untuk mengetahui pengaruh jumlah *itermax* terhadap rata-rata nilai *accuracy*, *precision*, *recall*, serta *f-measure*. Hasil pengujian menunjukkan bahwa nilai *accuracy* terbaik sistem adalah 79% pada *itermax* sebanyak 150 dengan rata-rata nilai *precision* = 84,46%, rata-rata nilai *recall* = 79%, dan rata-rata nilai *f-measure* = 81,63%. Sedangkan rata-rata nilai *accuracy* terendah adalah sebesar 51,5% pada *itermax* sebanyak 700 dan 1000 iterasi dengan rata-rata nilai *precision* = 53,61% dan 54,09% , rata-rata nilai *recall* = 51,5% dan 51,5%, dan nilai *f-measure* = 52,44% dan 52,68%. Pada Gambar 6.4 terlihat grafik cenderung menurun saat iterasi semakin banyak. Hal ini disebabkan oleh kondisi *overfitting* yang terjadi pada sistem. Kondisi ini adalah kondisi dimana sistem terlalu bergantung pada data *training* sehingga solusi yang dibutuhkan adalah menambahkan data agar sistem dapat belajar lebih baik dengan variasi data yang semakin tinggi.

BAB 7 PENUTUP

Bab ini memaparkan mengenai kesimpulan yang didapat berdasarkan penelitian yang telah dilakukan dan hasil-hasil yang didapatkan pada penelitian. Kemudian, memaparkan saran-saran yang diambil dari kekurangan penelitian untuk penelitian selanjutnya.

7.1 Kesimpulan

Dari hasil pengujian yang telah dilakukan, dapat diambil kesimpulan dalam klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik menggunakan *Support Vector Machine* dengan kernel *polynomial of degree 2*. Beberapa kesimpulan yang didapat adalah sebagai berikut:

1. Untuk melakukan klasifikasi penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik dapat digunakan metode *Support Vector Machine* dengan menggunakan kernel *polynomial of degree 2*. Hasil klasifikasi yang didapatkan adalah kelas penyakit Skizofrenia Hebefrenik dan kelas penyakit Episode Depresi Berat dengan Gejala Psikotik. Data yang digunakan adalah data rekam medik pasien yang terserang penyakit Skizofrenia Hebefrenik dan Episode Depresi Berat dengan Gejala Psikotik yang didapatkan dari Rumah Sakit Jiwa Dr. Radjiman Wediodiningrat Lawang. Banyak data yang digunakan adalah 200 data yang terdiri dari 100 data pasien dengan penyakit Skizofrenia Hebefrenik dan 100 data pasien dengan penyakit Episode Depresi Berat dengan Gejala Psikotik. Rasio perbandingan data latih dan data uji yang digunakan adalah sebanyak 80%:20% yaitu sebanyak 160 data latih dan 40 data uji. Dari 160 data latih yang digunakan, terdiri atas 80 data pasien dengan penyakit Skizofrenia Hebefrenik dan 80 data pasien dengan penyakit Episode Depresi Berat dengan Gejala Psikotik. Dan dari 40 data uji yang digunakan, terdiri dari 20 data pasien dengan penyakit Skizofrenia Hebefrenik dan 20 data pasien dengan penyakit Episode Depresi Berat dengan Gejala Psikotik. Hasil klasifikasi yang didapatkan berupa kelas penyakit Skizofrenia Hebefrenik dan kelas penyakit Episode Depresi Berat dengan Gejala Psikotik.
2. Berdasarkan hasil pengujian yang telah dilakukan, didapatkan nilai rata-rata *accuracy* terbaik adalah sebesar 79% dengan rata-rata *precision* sebesar 84,46%, rata-rata *recall* sebesar 79%, dan *f-measure* sebesar 81,63%. Perbandingan data yang digunakan adalah 80% data latih dan 20% data uji. Parameter yang digunakan yaitu nilai λ (*Lambda*) terbaik sebesar 0,1, nilai γ (*gamma*) terbaik sebesar 0,00001, nilai C (*complexity*) terbaik sebesar 0,01, dan jumlah *itermax* sebanyak 150 iterasi. Parameter-parameter *support vector machine* sangat berpengaruh terhadap tingkat akurasi yang akan didapatkan.

7.2 Saran

Dari penelitian yang telah dilakukan, terdapat beberapa kekurangan pada penelitian ini. Oleh sebab itu, perlu adanya perbaikan dan pengembangan untuk penelitian selanjutnya dari penelitian ini. Saran untuk penelitian selanjutnya adalah :

1. Data yang digunakan pada penelitian ini adalah 200 data, dimana 200 data merupakan data yang kecil untuk penggunaan metode *support vector machine* (SVM). Oleh karena itu, diharapkan untuk penelitian selanjutnya agar menambah data yang digunakan untuk penelitian agar tidak menyebabkan terjadinya *overfitting*.
2. Kelas klasifikasi yang digunakan pada penelitian berupa kelas penyakit Skizofrenia Hebefrenik dan kelas penyakit Depresi Berat dengan Gejala Psikotik. Diharapkan penelitian selanjutnya untuk menambahkan jenis kelas Skizofrenia maupun Depresi agar menambah variasi data serta kelas yang digunakan.



DAFTAR PUSTAKA

- Arga, D. A., Lestari, U., & Sutanta, E. (2017). Implementasi Data Mining untuk Memprediksi Masa Studi Mahasiswa Menggunakan Algoritme C4.5. *SCRIPT*, 5, 17-24.
- Badan Penelitian dan Pengembangan Kesehatan. (2013). *Riset Kesehatan Dasar*. Jakarta: Kementerian Kesehatan RI. Dipetik April 16, 2018, dari <http://www.depkes.go.id/resources/download/general/Hasil%20Risikedas%202013>
- Banjarsari, M. A., Budiman, H. I., & Farmadi, A. (2015). Penerapan K-Optimal Pada Algoritma Knn untuk Prediksi Kelulusan Tepat Waktu Mahasiswa Program Studi Ilmu Komputer Fmipa Unlam Berdasarkan IP Sampai Dengan Semester 4. *Kumpulan Jurnal Ilmu Komputer (KLIK)*, 2, 50-64.
- Hiesh, M.-H., Andy, Y.-Y. L., Shen, C.-P., Member, S., IEEE, Chen, W., . . . Senior Member, I. (2013). Classification of Schizophrenia using Genetic Algorithm-Support Vector Machine (GA-SVM). *IEEE*, 6047-6050.
- Indriani, A. (2014). Klasifikasi Data Forum dengan menggunakan Metode Naïve Bayes Classifier. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, G5-G10.
- Kementerian Kesehatan Republik Indonesia. (2016). *Peran Keluarga Dukung Kesehatan Jiwa Masyarakat*. Dipetik April 12, 2018, dari <http://www.depkes.go.id/article/print/16100700005/peran-keluarga-dukung-kesehatan-jiwa-masyarakat.html>
- 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*, 2, 1866-1873.
- Lu, X., Yang, Y., Wu, F., Gao, M., Xu, Y., Zhang, Y., . . . Peng, H. (2016). Discriminative Analysis of Schizophrenia Using Support Vector Machine and Recursive Feature Elimination on Structural MRI Images. *Medicine*.
- Luqyana, W. A., Cholissodin, I., & Perdana, R. S. (2018). Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4704-4713.
- Maslim, R. (2013). *Diagnosis Gangguan Jiwa, Rujukan Ringkas PPDGJ-III dan DSM-5*. Jakarta: Bagian Ilmu Kedokteran Jiwa FK-Unika Atmajaya.
- Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). Support Vector Machine.
- Octaviani, P. A., Wilandari, Y., & Ispriyanti, D. (2014). Penerapan Metode Klasifikasi Support Vector Machine (SVM) pada Data Akreditasi Sekolah Dasar (SD) di Kabupaten Magelang. *Jurnal Gaussian*, 811-820.



- Pramadhani, A. E., & Setiadi, T. (2014). Penerapan Data Mining untuk Klasifikasi Prediksi Penyakit ISPA (Infeksi Saluran Pernapasan Akut) dengan Algoritma Decision Tree (ID3). *Sarjana Teknik Informatika*, 2, 831-839.
- Puspitasari, A. M., Ratnawati, D. E., & Widodo, A. W. (2018). Klasifikasi Penyakit Gigi Dan Mulut Menggunakan Metode Support Vector Machine. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2, 802-810.
- Putri, R. E., Suparti, & Rahmawati, R. (2014). Perbandingan Metode Klasifikasi Naive Bayes dan K-Nearest Neighbor pada Analisis Data Status Kerja di Kabupaten Demak Tahun 2012. *Gaussian*, 3, 831-838.
- Rachman, F., & Purnami, S. W. (2012). Perbandingan Klasifikasi Tingkat Keganasan Breast Cancer Dengan Menggunakan Regresi Logistik Ordinal Dan Support Vector Machine (SVM) . *Jurnal Sains dan Seni ITS*, 130-135.
- Raharjo, S., & Winarko, E. (2014). Klasterisasi, Klasifikasi dan Peringkasan Teks Berbahasa Indonesia. *Prosiding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2014)*, 8, 391-401.
- Riyadi, A. S., Retnandi, E., & Deddy, A. (2012). Perancangan Sistem Informasi Berbasis Website Subsistem Guru di Sekolah Pesantren Persatuan Islam 99 Rancabango. *Jurnal Algoritma*, 11.
- Rosandy, T. (2016). Perbandingan Metode Naive Bayes Classifier dengan Metode Decision Tree (C4.5) untuk Menganalisa Kelancaran Pembiayaan. *Jurnal TIM Darmajaya*, 02, 52-62.
- Suhaimi. (2015). Gangguan Jiwa dalam Perspektif Kesehatan Mental Islam . *Risalah*, 26(4), 197-205.
- Susilowati, E., Sabariah, M. K., & Gozali, A. A. (2015). Implementasi Metode Support Vector Machine untuk Melakukan Klasifikasi Kemacetan Lalu Lintas pada Twitter. *e-Proceeding of Engineering*, 2, 1478-1484.
- Vijayakumar, S., & Wu, S. (1999). Sequential Support Vector Classifier and Regression.