

IDENTIFIKASI PENYAKIT MATA MENGGUNAKAN METODE *LEARNING VECTOR QUANTIZATION (LVQ)*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Entra Betlin Ladauw
NIM: 135150201111124



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

PENGESAHAN

IDENTIFIKASI PENYAKIT MATA MENGGUNAKAN METODE *LEARNING VECTOR QUANTIZATION* (LVQ)

SKRIPSI

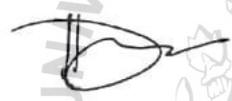
Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

**Disusun Oleh :
Entra Betlin Ladauw
NIM: 135150201111124**

**Skripsi ini telah diuji dan dinyatakan lulus pada
2 Agustus 2018
Telah diperiksa dan disetujui oleh:**

Dosen Pembimbing I

Dosen Pembimbing II



**Dian Eka Ratnawati, S.Si., M.Kom
NIP: 19730619 200212 2 001**

**Ahmad Afif Supianto, Dr.Eng., S.Si.,
M.Kom
NIK: 2012018206231001**

**Mengetahui
Ketua Jurusan Teknik Informatika**



**Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 197410518 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, 18 Juli 2018



Entra Betlin Ladauw

NIM: 13515020111124



KATA PENGANTAR

Segala puji dan syukur hanya bagi Tuhan Yesus Kristus, oleh karena penyertaan dan anugerah-Nya yang melimpah, kemurahan dan kasih setia-Nya yang besar sehingga penulis mampu menyelesaikan skripsi yang berjudul “Identifikasi Penyakit Mata Menggunakan Metode *Learning Vector Quantization* (LVQ)”.

Penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari kesempurnaan dan tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh sebab itu, penulis ingin menyampaikan rasa terima kasih atas segala pemikiran, motivasi, dukungan serta doa kepada pihak-pihak tersebut, diantaranya:

1. Ibu Dian Eka Ratnawati, S.Si., M.Kom selaku Dosen Pembimbing I dan Bapak Ahmad Afif Supianto, Dr.Eng., S.Si., M.Kom selaku Dosen Pembimbing II yang telah meluangkan waktu serta dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika.
3. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer.
4. Seluruh dosen dan civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah mendidik, membina karakter serta membantu penulis selama menempuh pendidikan sarjana hingga dalam pelaksanaan penyusunan skripsi.
5. Ibu Cynthia Aquansih dan Bapak Achmad Ladauw, selaku orang tua penulis yang dengan kesabaran dan cintanya senantiasa mengayomi, mendidik, dan memotivasi penulis dalam menyelesaikan pendidikan sarjana.
6. Erik Piterson Ladauw, Izaak Rayner Ladauw, Theresia Budi Utami Bandi, selaku kakak penulis yang memberikan semangat, masukan serta arahan selama masa pengerjaan skripsi.
7. Para sahabat penulis antara lain Meidy H.D. Pinangsirih, Putri R. Hardein, Annisa S. Sani, Fida D. Febriani, Alawiyah Nurmala, Donny Kurniawan, Irfan Aprison, Muhammad Ridha Anshari, dan Linda Pratiwi. Terima kasih atas dukungan, doa serta kebersamaannya.

Malang, 18 Juli 2018

Penulis

entrabethlin@gmail.com

ABSTRAK

Menjaga dan merawat kesehatan mata sangatlah penting bagi manusia sebab mata merupakan salah satu indera yang membantu manusia dalam melakukan kegiatan sehari-hari. Mata yang memberikan informasi visual kepada manusia, tidak terlepas dari ancaman akan penyakit yang dapat menyerangnya. Penyakit tersebut bisa menyerang dalam skala yang ringan ataupun skala yang berat. Sayangnya, kesehatan mata dianggap tidak memiliki potensi untuk membahayakan manusia, sehingga kesehatan mata seringkali diabaikan oleh masyarakat pada umumnya. Oleh karena itu, pada permasalahan ini kemudian dikembangkan sistem identifikasi penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ). Metode ini mampu memberikan klasifikasi terhadap pola yang akan mewakili kelas tertentu. Pola tersebut akan berpindah ke posisi lebih dekat pada kelas yang sesuai jika titik data klasifikasi benar. Pada penelitian ini, terdapat 21 gejala dan 9 penyakit mata yang kemudian akan diproses dalam pelatihan dan pengujian dengan data-datanya akan dibagi menjadi data latih dan data uji. Pada proses pelatihan, metode LVQ akan melakukan beberapa tahapan untuk mendapatkan bobot akhir. Bobot ini nantinya akan digunakan pada proses pengujian. Dengan menggunakan metode LVQ, nilai parameter yang didapatkan antara lain adalah $\alpha = 0,4$, $\text{Dec } \alpha = 0,8$, $\text{Min } \alpha = 0,00001$, $\text{Max Epoch} = 25$, data latih = 100 data (80%) dan data uji = 25 data (20%). Pada pengujian akurasi sistem ini, hasil rata-rata akurasi yang diperoleh sebesar 82,80% dan hasil akurasi tertinggi sebesar 92% yang menunjukkan bahwa sistem dapat berfungsi dengan baik. Sehingga metode LVQ mampu diterapkan pada identifikasi penyakit mata.

Kata kunci: identifikasi, penyakit mata, *Learning Vector Quantization*

ABSTRACT

Taking care and maintaining healthy eyes are very important for human, because eyes are one of the senses that help human to do daily activities. Eyes that give visual information to human, cannot be separated from the threat of many eyes diseases. The diseases can attack from small to big scale. Unfortunately, eyes diseases are usually considered not to have such potential to harm human, so eyes health often to be ignored by people in general. Therefore, in this paper a system to identify eyes diseases has been developed using Learning Vector Quantization (LVQ) method. This method can give classification to a pattern that represent spesific class, which will move to a nearer position to corresponding class when the classification data point is true. In this research, there are 21 symptoms and 9 eyes diseases that processed in training and testing processes, where the data were divided into training data and testing data. In training process, LVQ method did some stages to get final weight. The weight will be used in testing process. Using LVQ method, obtained parameter values are $\alpha = 0.4$, Dec $\alpha = 0.8$, Min $\alpha = 0.00001$, Max Epoch = 25, training data = 100 data (80%) and data test = 25 data (20%). From accuracy testing for this system, the result show 82.80% average accuracy and 92% highest accuracy, that means this system works fine. So, it can be concluded that LVQ method can be used for eyes diseases identification.

Keyword: *identification, eyes diseases, Learning Vector Quantization*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR KODE SUMBER.....	xv
DAFTAR LAMPIRAN	xvi
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 Penyakit mata	12
2.2.1 Glaukoma	12
2.2.2 Katarak	14
2.2.3 Konjungtivitis.....	15
2.2.4 Keratitis	16
2.2.5 Pterigium	17
2.2.6 Uveitis.....	17
2.2.7 Dry Eyes.....	18
2.2.8 Bleparitis	19
2.2.9 Kalazion	20
2.3 Jarak <i>Euclidean</i>	20



2.4 Jaringan saraf tiruan	20
2.5 <i>Learning Vector Quantization</i> (LVQ)	22
2.6 Pengujian	23
BAB 3 METODOLOGI	26
3.1 Studi literatur	26
3.2 Pengumpulan data	27
3.3 Analisis kebutuhan	27
3.3.1 Kebutuhan Fungsional	27
3.3.2 Kebutuhan Non Fungsional	27
3.4 Perancangan	28
3.5 Implementasi	28
3.6 Pengujian	29
3.7 Kesimpulan dan saran	29
BAB 4 PERANCANGAN	30
4.1 Perancangan sistem	30
4.2 Proses pelatihan	31
4.3 Proses perhitungan jarak	33
4.4 Proses pengujian	35
4.5 Data yang digunakan pada penelitian	36
4.5.1 Proses konversi biner	36
4.6 Proses manualisasi	40
4.6.1 Proses pelatihan	40
4.6.2 Proses pengujian	56
4.6.3 Proses perhitungan akurasi	59
4.7 Perancangan desain antarmuka	60
4.7.1 Desain antarmuka data latih	60
4.7.2 Desain antarmuka proses training	61
4.7.3 Desain antarmuka data uji	62
4.7.4 Desain antarmuka proses pengujian	62
4.7.5 Desain antarmuka hasil identifikasi	63
4.7.6 Desain antarmuka identifikasi dengan <i>inputan user</i>	64
4.8 Perancangan pengujian	65



4.8.1 Perancangan pengujian pengaruh nilai <i>learning rate</i> (α) terhadap hasil akurasi.....	65
4.8.2 Perancangan pengujian pengaruh pengurangan nilai rate (dec α) terhadap hasil akurasi	65
4.8.3 Perancangan pengujian pengaruh minimal <i>learning rate</i> (min α) terhadap hasil akurasi	66
4.8.4 Perancangan pengujian pengaruh iterasi maksimum (Max Epoch) terhadap hasil akurasi	66
4.8.5 Perancangan pengujian pengaruh perbandingan jumlah data latih dan data uji terhadap hasil akurasi.....	67
BAB 5 IMPLEMENTASI	68
5.1 Spesifikasi sistem	68
5.1.1 Spesifikasi perangkat keras	68
5.1.2 Spesifikasi perangkat lunak.....	68
5.2 Batasan implementasi	69
5.3 Implementasi algoritme.....	69
5.3.1 Metode <i>Learning Vector Quantization</i> (LVQ)	70
5.4 Implementasi antarmuka.....	79
5.4.1 Implementasi halaman data latih	79
5.4.2 Implementasi halaman LVQ-training	79
5.4.3 Implementasi halaman data uji	80
5.4.4 Implementasi halaman LVQ-testing.....	80
5.4.5 Implementasi halaman hasil identifikasi.....	81
5.4.6 Implementasi halaman identifikasi dengan <i>inputan user</i>	82
BAB 6 PENGUJIAN DAN ANALISIS.....	83
6.1 Pengujian <i>learning rate</i> (α).....	83
6.2 Pengujian pengurangan <i>learning rate</i> (Dec α).....	84
6.3 Pengujian <i>minimal learning rate</i> (Min α).....	86
6.4 Pengujian jumlah iterasi maksimum (Max Epoch)	87
6.5 Pengujian antara perbandingan data latih dan data uji	89
BAB 7 PENUTUP	92
7.1 Kesimpulan.....	92
7.2 Saran	92



DAFTAR PUSTAKA..... 93
LAMPIRAN A Tabel data 96



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	9
Tabel 3.1 Spesifikasi Perangkat Keras	28
Tabel 3.2 Spesifikasi Perangkat Lunak	28
Tabel 4.1 Data Latih	37
Tabel 4.2 Konversi Biner pada Data Latih	39
Tabel 4.3 Bobot Awal Variabel	40
Tabel 4.4 Hasil Proses Perhitungan Jarak Terdekat dengan Menggunakan <i>Euclidean Distance</i> untuk Iterasi Pertama	43
Tabel 4.5 Perubahan Bobot w_1 pada Iterasi 1.....	45
Tabel 4.6 Perubahan Bobot w_2 pada Iterasi 1.....	46
Tabel 4.7 Perubahan Bobot w_3 pada Iterasi 1.....	47
Tabel 4.8 Perubahan Bobot w_4 pada Iterasi 1.....	48
Tabel 4.9 Perubahan Bobot w_5 pada Iterasi 1.....	49
Tabel 4.10 Perubahan Bobot w_6 pada Iterasi 1.....	50
Tabel 4.11 Perubahan Bobot w_7 pada Iterasi 1.....	51
Tabel 4.12 Perubahan Bobot w_8 pada Iterasi 1.....	52
Tabel 4.13 Perubahan Bobot w_9 pada Iterasi 1.....	53
Tabel 4.14 Hasil Bobot Akhir yang Didapatkan pada Iterasi Kelima	55
Tabel 4.15 Data Uji	56
Tabel 4.16 Hasil dari Proses Perhitungan Jarak Terdekat dengan Menggunakan <i>Euclidean Distance</i> pada Data Uji	59
Tabel 4.17 Perbandingan Kelas Target dan Prediksi pada Proses Pengujian	59
Tabel 4.18 Perancangan Pengujian Pengaruh Nilai <i>Learning Rate</i> (α) terhadap Hasil Akurasi	65
Tabel 4.19 Perancangan Pengujian Pengaruh Pengurangan Nilai <i>Rate</i> (Dec α) terhadap Hasil Akurasi	66
Tabel 4.20 Perancangan Pengujian Pengaruh Nilai <i>Minimal Learning Rate</i> (Min α) terhadap Hasil Akurasi	66
Tabel 4.21 Perancangan Pengujian Pengaruh Nilai Iterasi Maksimum (Max Epoch) terhadap Hasil Akurasi	66
Tabel 4.22 Perancangan Pengujian Pengaruh Perbandingan Jumlah Data Latih dan Data Uji terhadap Hasil Akurasi	67

Tabel 5.1 Spesifikasi Perangkat Keras 68

Tabel 5.2 Spesifikasi Perangkat Lunak 69

Tabel 6.1 Tabel Pengujian *Learning Rate* (α) 83

Tabel 6.2 Tabel Pengujian Pengurangan *Learning Rate* (Dec α)..... 85

Tabel 6.3 Tabel Pengujian *Minimal Learning Rate* (Min α) 86

Tabel 6.4 Tabel Pengujian Jumlah Iterasi Maksimum (Max Epoch) 88

Tabel 6.5 Tabel Pengujian Perbandingan Data Latih dan Data Uji 89

Tabel 6.6 Tabel Pengujian Akurasi 91



DAFTAR GAMBAR

Gambar 2.1 Glaukoma	13
Gambar 2.2 Katarak	14
Gambar 2.3 Konjungtivitis.....	16
Gambar 2.4 Keratitis	16
Gambar 2.5 Pterigium	17
Gambar 2.6 Uveitis.....	18
Gambar 2.7 Dry Eyes	19
Gambar 2.8 Bleparitis.....	19
Gambar 2.9 Kalazion	20
Gambar 2.10 Jaringan Saraf pada <i>Learning Vector Quantization (LVQ)</i>	22
Gambar 3.1 Metode Penelitian.....	26
Gambar 4.1 Perancangan Secara Umum	30
Gambar 4.2 <i>Flowchart</i> (Diagram Alir) Perancangan Sistem	31
Gambar 4.3 <i>Flowchart</i> (Diagram Alir) Proses Pelatihan	32
Gambar 4.4 <i>Flowchart</i> (Diagram Alir) Proses Perhitungan Jarak.....	34
Gambar 4.5 <i>Flowchart</i> (Diagram Alir) Proses Pengujian.....	35
Gambar 4.6 Antarmuka Data Latih	60
Gambar 4.7 Antarmuka Proses Training	61
Gambar 4.8 Antarmuka Data Uji.....	62
Gambar 4.9 Antarmuka Proses Pengujian	63
Gambar 4.10 Antarmuka Hasil Identifikasi	63
Gambar 4.11 Antarmuka Identifikasi dengan <i>Inputan User</i>	64
Gambar 5.1 Implementasi Halaman Data Latih.....	79
Gambar 5.2 Implementasi Halaman LVQ-Training	80
Gambar 5.3 Implementasi Halaman Data Uji	80
Gambar 5.4 Implementasi Halaman LVQ-Testing.....	81
Gambar 5.5 Implementasi Halaman Hasil Identifikasi.....	82
Gambar 5.6 Implementasi Halaman Identifikasi dengan <i>Inputan User</i>	82
Gambar 6.1 Pengujian <i>Learning Rate</i> (α).....	84
Gambar 6.2 Pengujian Pengurangan <i>Learning Rate</i> ($\text{Dec } \alpha$)	85



Gambar 6.3 Pengujian *Minimal Learning Rate* ($\text{Min } \alpha$) 87

Gambar 6.4 Pengujian Jumlah Iterasi Maksimum (Max Epoch) 88

Gambar 6.5 Pengujian Perbandingan Data Latih dan Data Uji 90



DAFTAR KODE SUMBER

Kode Sumber 5.1 Menentukan Bobot Awal..... 71

Kode Sumber 5.2 Perhitungan Jarak Minimum (*Euclidean Distance*) 78

Kode Sumber 5.3 Perhitungan Jarak Minimum (Cari Nilai Minimum)..... 78

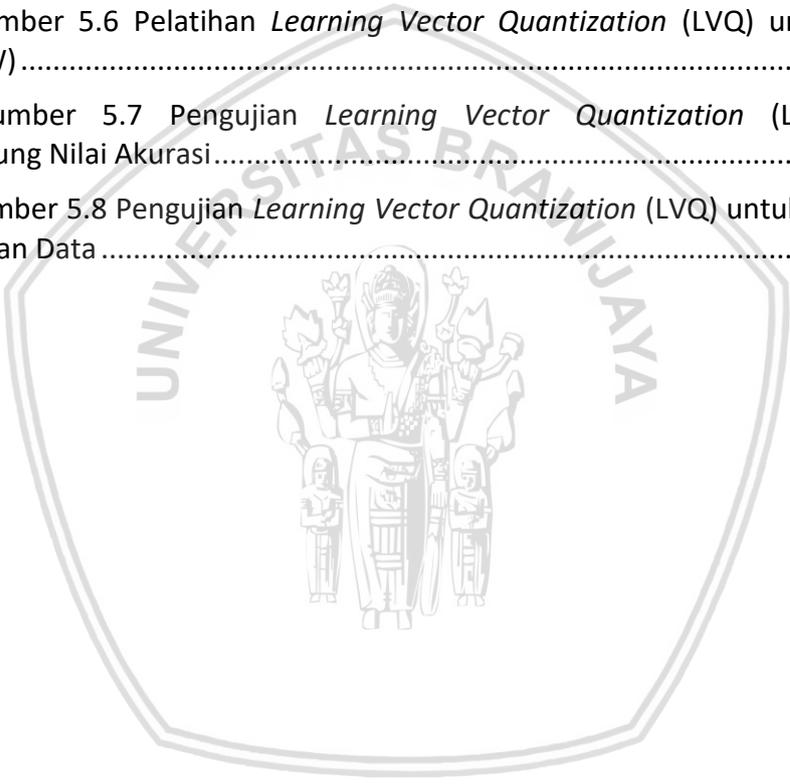
Kode Sumber 5.4 Pencarian Nilai Class/Category Berdasarkan Perhitungan Jarak Minimum..... 72

Kode Sumber 5.5 Pelatihan *Learning Vector Quantization* (LVQ) untuk Hitung Delta Bobot (ΔW) 73

Kode Sumber 5.6 Pelatihan *Learning Vector Quantization* (LVQ) untuk Hitung Bobot (W)..... 76

Kode Sumber 5.7 Pengujian *Learning Vector Quantization* (LVQ) untuk Menghitung Nilai Akurasi..... 77

Kode Sumber 5.8 Pengujian *Learning Vector Quantization* (LVQ) untuk Mengecek Kesesuaian Data 77



DAFTAR LAMPIRAN

LAMPIRAN A Tabel data 96



BAB 1 PENDAHULUAN

1.1 Latar belakang

Mata merupakan salah satu indera pada manusia yang sangat berperan penting dalam kehidupan. Manusia dapat menyerap informasi visual yang digunakan untuk melakukan aktivitas sehari-hari dengan indera mata. Peran dari mata yang begitu penting membuat manusia harus mampu menjaga kesehatan mata agar terhindar dari ancaman penyakit mata yang menyerang dalam skala yang ringan ataupun skala yang berat. Mata memberikan informasi yang dapat diproses oleh manusia sebagai salah satu indera yang digunakan untuk melihat, sehingga dengan memiliki mata yang baik dan sehat dapat memberikan kebaikan pada manusia dalam melakukan kegiatan sehari-hari (Perdami, 2013). Namun, pada kenyataannya kesehatan pada mata sering diabaikan karena masyarakat pada umumnya sering menganggap bahwa penyakit yang menyerang mata bukanlah penyakit yang mematikan dan tidak memberikan dampak yang berbahaya bagi hidup. Pandangan masyarakat tersebut memberikan makna bahwa penyakit mata dianggap tidak memiliki potensi untuk membahayakan manusia, padahal penyakit pada mata dapat mengakibatkan gangguan terhadap penglihatan yang akan memengaruhi aktivitas manusia (Perdami, 2013).

Di Indonesia, terdapat dua penyebab penyakit mata yang sering terjadi pada pasien penyakit mata yaitu gangguan refraksi dan katarak (Perdami, 2013). Jumlah orang yang mengalami gangguan penglihatan diseluruh dunia pada tahun 2010 adalah sebanyak 285 juta orang atau sekitar 4,24% populasi, sedangkan jumlah orang yang menderita kebutaan sebanyak 0,58% atau sekitar 39 juta dan jumlah orang yang mengalami *low vision* sebanyak 3,65% atau setara dengan 246 juta orang. Sisanya 65% adalah merupakan orang-orang yang mengalami gangguan penglihatan dari penyandang kebutaan untuk usia 50 tahun atau lebih (Perdami, 2013). Badan kesehatan dunia seperti *World Health Organization* (WHO) dan PERDAMI (Persatuan Dokter Spesialis Mata Indonesia) telah berupaya dalam memberikan banyak program untuk membantu masyarakat dalam melakukan pencegahan terhadap penyakit mata. Salah satu program yang dibuat untuk penanggulangan kebutaan dan gangguan penglihatan adalah program *Vision 2020* yang direkomendasikan WHO. Program *Vision 2020* membutuhkan ketersediaan data mengenai keadaan kebutaan dan gangguan penglihatan disuatu negara atau wilayah dengan menggunakan *Rapid Assessment of Avoidable Blindness* (RAAB) sebagai standar untuk melakukan pengumpulan data. Di Indonesia, RAAB telah dilakukan di tiga provinsi yaitu Sulawesi Selatan, Jawa Barat, dan Nusa Tenggara Barat (NTB), namun yang menjadi hambatan adalah penyebaran RAAB yang belum menyeluruh kepada daerah-daerah di Indonesia secara merata (Perdami, 2013).

Berdasarkan pada uraian di atas, beberapa cara dilakukan untuk mengatasi permasalahan tersebut agar dapat menanggulangi kebutaan dan gangguan penglihatan diantaranya adalah melakukan identifikasi terhadap penyakit mata.

Identifikasi terhadap penyakit mata sudah dilakukan beberapa kali oleh penelitian sebelumnya, diantaranya adalah penelitian terkait dengan tema yang sama namun dengan menggunakan metode yang berbeda yakni *Naïve Bayes-Certainty Factor*. Pada penelitian tersebut, peneliti menggunakan data latih sebanyak 80 data dan data uji sebanyak 125 data dengan pengujian yang digunakan adalah *K-Fold Cross Validation* yang dibagi sebanyak lima *fold* untuk masing-masing data uji yang telah ada. Selain itu terdapat juga penyakit mata serta gejala yang digunakan peneliti yakni 9 penyakit mata dan 21 gejala pada penyakit mata yang menunjang dalam pengerjaan penelitian tersebut. Pengujian akurasi yang dilakukan dengan menggunakan metode *Naïve Bayes-Certainty Factor* mendapatkan hasil akurasi rata-rata sebesar 78,4% (Lestari, Hidayat, & Putri, 2016). Kekurangan dari penelitian sebelumnya adalah komposisi data yang digunakan belum seimbang dengan menggunakan metode *Naïve Bayes-Certainty Factor* sehingga peluang untuk memberikan hasil akurasi yang lebih baik masih dirasa kurang. Hasil akurasi yang didapatkan pada penelitian sebelumnya dipengaruhi oleh pengujian yang digunakan peneliti dengan menggunakan *K-Fold Cross Validation* dengan nilai k pada pengujian akan berpengaruh, baik nilainya kecil maupun besar terhadap nilai akurasi yang dihasilkan.

Penelitian ini kemudian mengangkat permasalahan yang sama namun menggunakan metode yang berbeda yakni metode *Learning Vector Quantization* (LVQ). Metode ini diharapkan dapat memberikan proses perhitungan serta hasil akurasi rata-rata yang jauh lebih baik dari metode sebelumnya. Pemilihan metode klasifikasi dengan menggunakan LVQ pada permasalahan ini selain karena proses perhitungan yang dilakukan jauh lebih baik pada saat *learning* baik terhadap data latih dan data uji, metode ini juga tergantung pada beberapa parameter yang menunjang dalam proses identifikasi penyakit mata antara lain adalah *max epoch*, *alpha (learning rate)*, *min alpha (minimum learning rate)* dan *dec alpha (decrease/learning rate reduction)*. Dasarnya, struktur jaringan metode LVQ terdiri atas tiga *layer* yakni *input*, *hidden* dan *output* dan dalam metode LVQ digunakan *vector* untuk mengklasifikasikan *output* sebagai kategori atau kelas yang diwakili oleh pendekatan *Euclidean distance* (Fausett, 1994).

Berbagai penelitian terkait dengan metode klasifikasi *Learning Vector Quantization* (LVQ) telah dilakukan oleh beberapa peneliti sebelumnya. Salah satunya adalah penelitian yang dilakukan oleh Nurul Islamiah terkait dengan penyakit gigi dan mulut yang mendapatkan hasil akurasi rata-rata terbaik yaitu 100% dengan masing-masing perbandingan data latih dan data uji sebesar 90:10 (Islamiah, Cholissodin, & Dewi, 2016). Penelitian selanjutnya adalah mengenai perawatan kulit yang dilakukan oleh Wijayanti dengan menggunakan metode LVQ dengan hasil akurasi terbaik yang diperoleh adalah sebesar 98,25% pada citra kulit seseorang (Wijayanti, Cholissodin, & Murlistyarini, 2016). Kemudian metode LVQ juga digunakan pada penelitian yang dilakukan oleh Prabowo terkait dengan masalah pengenalan menggunakan bahasa isyarat dengan mendapat hasil akurasi rata-rata terbaik sebesar 88% menggunakan citra digital (Prabowo, Dewi, & Suprpto, 2015). Penelitian yang dilakukan dengan metode LVQ juga tidak hanya berpusat kepada manusia namun juga diterapkan kedalam pemilihan

kemintaan yang dilakukan oleh peneliti Martinuva dalam penelitiannya yang mengangkat studi kasus di dalam program Teknik Informatika UB. Hasil akurasi yang didapatkan peneliti tersebut adalah sebesar 84,5% untuk rata-rata yang terbaik (Martinuva, Ratnawati, & Indriati, 2015).

Selain penelitian mengenai diagnosis penyakit pada manusia, metode LVQ juga dapat diterapkan pada tanaman dan hewan. Penelitian yang dilakukan oleh Umam adalah dengan menggunakan citra pada daun dari tanaman kedelai yang kemudian setelah diproses mendapatkan hasil rata-rata akurasi terbaik sebesar 93% guna mengidentifikasi penyakit dari tanaman tersebut (Umam, Dewi, & Cholissodin, 2015). Lalu, terdapat pula penelitian yang terkait dengan tanaman yang dilakukan oleh Krisnanti terkait dengan kualitas dari daun nilam. Peneliti melakukan klasifikasi dengan menggunakan metode LVQ yang kemudian memperoleh hasil rata-rata akurasi terbaik sebesar 98% menggunakan citra digital (Krisnanti, Dewi, & Cholissodin, 2015). Penelitian selanjutnya terkait dengan mutu kualitas pada susu sapi yang dilakukan oleh Ganidar dengan menggunakan metode LVQ. Dalam melakukan klasifikasi pada mutu susu sapi tersebut, peneliti mendapatkan hasil akurasi yang berbeda sebab peneliti membuat dua dataset yang diurut dan diacak. Untuk dataset yang diurut, peneliti mendapatkan hasil akurasi terbaik sebesar 89,51% sedangkan untuk dataset yang di acak peneliti mendapatkan hasil akurasi terbaik sebesar 91,79% (Ganidar, Putri, & Dewi, 2015).

Berdasarkan pada penjelasan latar belakang yang telah diuraikan beserta dengan penelitian-penelitian terdahulu maka penulis akan membuat penelitian yang berjudul "Identifikasi Penyakit Mata Menggunakan Metode *Learning Vector Quantization* (LVQ)". Pada penelitian ini, peneliti akan menggunakan metode *Learning Vector Quantization* (LVQ) untuk mengidentifikasi penyakit mata. Harapannya dengan melakukan identifikasi terhadap penyakit mata berdasarkan pada gejala-gejala serta jenis dari penyakit mata, dapat membantu memberikan hasil terbaik untuk melakukan penanganan serta pengobatan pada penyakit mata secara baik dan benar.

1.2 Rumusan masalah

Setelah penjabaran terkait latar belakang masalah yang akan diangkat pada penelitian ini, berikut ini adalah rumusan masalah yang didapatkan:

1. Apakah metode *Learning Vector Quantization* (LVQ) dapat diterapkan kepada penyakit mata?
2. Seberapa akurat metode *Learning Vector Quantization* (LVQ) terhadap penyakit mata?

1.3 Tujuan

Setelah didapatkan rumusan masalah, selanjutnya adalah menentukan tujuan yang ingin dicapai pada penelitian ini. Berikut adalah tujuan yang terkait dengan masalah penelitian dan hasil yang ingin dicapai.

1. Menerapkan metode *Learning Vector Quantization* (LVQ) pada penyakit mata.
2. Melakukan pengujian terhadap akurasi untuk implementasi metode *Learning Vector Quantization* (LVQ) pada penyakit mata.

1.4 Manfaat

Manfaat yang didapatkan pada penelitian terkait pada penyakit mata dengan menggunakan implementasi metode *Learning Vector Quantization* (LVQ) adalah sebagai berikut.

1. Memudahkan para pengguna nantinya dalam melakukan identifikasi terhadap penyakit mata.
2. Membantu para pengguna dalam mengenali jenis gejala dan jenis penyakit pada mata.
3. Menjadi referensi bagi para pengguna dalam melakukan identifikasi terhadap penyakit mata.

1.5 Batasan masalah

Setelah diuraikan serta dijelaskan secara garis besar mengenai masalah yang dihadapi seperti pada latar belakang, maka pada batasan masalah ini dilakukan agar lingkup masalah tidak melebar dan keluar dari masalah utama. Berikut ini adalah batasan masalah pada penelitian ini.

1. Sistem yang akan dibangun menggunakan data gejala serta penyakit yang diperoleh dari skripsi milik Putri Lestari yang diambil dari poli mata RSUD Kabupaten Sidoarjo tahun 2015-2016.
2. Penyakit mata yang digunakan untuk melakukan identifikasi dengan menggunakan metode LVQ adalah 9 jenis penyakit dan 21 jenis gejala.
3. Pengujian yang akan dilakukan adalah pengujian yang terkait dengan parameter *Learning Vector Quantization* (LVQ) yakni parameter *Learning Rate* (α), Pengurangan *Learning Rate* (Dec α), *Minimal Learning Rate* (Min α) dan Iterasi Maksimum (Max Epoch). Selain itu juga akan dilakukan pengujian terhadap akurasi dengan menggunakan perbandingan antara jumlah data latih dan jumlah data uji.

1.6 Sistematika pembahasan

Pada bagian sistematika pembahasan akan diuraikan menjadi beberapa bab yang pada tiap bab akan menjelaskan masing-masing isi dari bab tersebut. Berikut ini adalah penjelasan lebih lanjut mengenai sistematika pembahasan.

BAB 1 PENDAHULUAN

Pada bab ini akan diuraikan mengenai latar belakang masalah mengenai penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ), rumusan masalah yang didasarkan atas latar belakang yang telah

dijabarkan, kemudian tujuan penelitian dibuat, manfaat, batasan masalah serta sistematika pembahasan yang membahas tentang penulisan dari tiap-tiap bab yang terdapat pada dokumen penelitian.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan diuraikan tentang kajian pustaka yang diperoleh dari penelitian-penelitian sebelumnya yang telah dilakukan baik itu terkait dengan masalah serta metode yang digunakan. Penelitian-penelitian tersebut meliputi penyakit mata yang menjadi masalah atau objek yang akan dibahas beserta dengan metode yang digunakan yakni *Learning Vector Quantization* (LVQ) untuk pengembangan penelitian.

BAB 3 METODOLOGI

Pada bab ini akan diuraikan tentang metodologi yang akan dibahas dalam pembangunan sistem baik dari tahapan analisis, perancangan sampai kepada pengimplementasian yang dilakukan untuk permasalahan penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ). Tahapan-tahapan dalam uraian bab ini meliputi studi literatur, pengumpulan data terkait dengan penyakit mata, analisis kebutuhan yang menjadi lingkungan pengujian bagi proses yang akan dikerjakan, perancangan proses, implementasi, pengujian serta kesimpulan dan saran.

BAB 4 PERANCANGAN

Pada bab ini akan diuraikan tentang perancangan proses yang akan digunakan nantinya untuk melakukan identifikasi pada penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ). Perancangan proses yang dibuat pada bab ini untuk kedepannya dapat digunakan sebagai acuan dalam mengimplementasikan proses terkait penyakit mata yang dibahas dengan menggunakan metode *Learning Vector Quantization* (LVQ).

BAB 5 IMPLEMENTASI

Pada bab ini akan diuraikan tentang implementasi sistem berdasarkan kepada perancangan yang telah dibuat pada bab sebelumnya. Implementasi sistem dilakukan untuk mengidentifikasi penyakit pada mata dengan menggunakan metode *Learning Vector Quantization* (LVQ).

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan diuraikan tentang pengujian serta analisis yang dilakukan pada sistem yang telah dibangun yakni identifikasi pada penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ).

BAB 7 PENUTUP

Pada bab ini akan diuraikan tentang kesimpulan dan saran dari penulis terkait dengan penelitian yang telah dikerjakan berupa sistem yang dibangun untuk identifikasi pada penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ).

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisikan mengenai kajian pustaka yang membahas tentang metode sekaligus teori yang digunakan sebagai penunjang dalam penelitian yang dilakukan. Kajian pustaka yang terdapat pada bab ini akan membahas mengenai penelitian-penelitian sebelumnya, baik yang memiliki kesamaan pada metode yang digunakan maupun teori yang terkait. Selain itu juga ada beberapa teori yang disajikan dalam bab ini sebagai landasan kepustakaan atas penelitian yang dibuat antara lain penyakit mata, jarak Euclidean, jaringan saraf tiruan dan *Learning Vector Quantization* (LVQ).

2.1 Kajian pustaka

Pada penelitian yang akan dikerjakan, diperlukan beberapa penelitian sebelumnya yang membantu, mendukung dan menunjang terkait dengan permasalahan penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ). Penjelasan mengenai masing-masing penelitian yang dilakukan sebelumnya nantinya akan dibandingkan satu dengan yang lainnya.

Pada penelitian pertama terkait dengan penyakit mata menggunakan metode *Naïve Bayes – Certainty Factor* menggunakan data sebanyak 80 data. Untuk jenis penyakit mata yang digunakan sebanyak 9 jenis yakni Kalazion, Glaukoma, Bleparitis, Katarak, *Dry Eyes*, Konjungtivitis, Uveitis, Keratitis, Uveitis dan Pterigium sedangkan untuk gejala pada penyakit mata yang digunakan sebanyak 21 gejala. Dengan menggunakan pengujian akurasi *K-Fold Cross Validation*, penulis mendapatkan hasil nilai akurasi sebanyak pengujian yang dilakukan sebanyak 5 kali. Untuk pengujian pertama dan ketiga didapatkan 84%, pengujian kedua didapatkan 72% dan pengujian keempat dan kelima didapatkan 76%. Sehingga rata-rata dari keseluruhan pengujian yang dilakukan oleh penulis dengan menggunakan metode *Naïve Bayes – Certainty Factor* adalah sebesar 78,4% (Lestari et al., 2016).

Penelitian kedua terkait dengan penyakit gigi dan mulut dengan menggunakan 16 gejala dengan 4 kelas penyakit. Pada masukan gejala yang sebanyak 16 yaitu gigi berlubang, gigi sakit/nyeri ketika menguyah/menggigit, gusi berdarah, gusi kemerahan, adanya karang gigi, gigi ngilu ketika mendapat rangsangan (makan manis/asam atau minum panas/dingin), gusi bengkak, gusi gatal, gigi sisa akar, gigi terasa sakit terus menerus, gigi goyang, gusi terasa sakit, bau mulut, sakit sampai ke telinga, nyeri spontan pada gigi tanpa rangsangan, dan gigi berubah warna abu-abu kehitaman/coklat. Sedangkan untuk 4 kelas penyakit adalah berupa pulpitis, gingivitis, nekrosis pulpa dan periodontitis. Peneliti disini menggunakan metode LVQ sebagai metode yang digunakan untuk melakukan klasifikasi pada penyakit gigi dan mulut sehingga pada data latih digunakan data sebanyak 84 data untuk data latih masing-masing kelas sebanyak 21 data dan untuk data uji digunakan data sebanyak 36 data untuk data uji masing-masing kelas sebanyak 9 data. Pada metode LVQ, tahapan yang dilakukan untuk melakukan klasifikasi pada penyakit gigi dan mulut adalah

dengan memberikan pembelajaran beserta dengan parameter yakni laju pembelajaran (α), konstanta pengurang laju pembelajaran (c), laju pembelajaran minimum dan iterasi maksimum. Pada pengujian yang dilakukan oleh peneliti didapatkan hasil rata-rata akurasi terbaik adalah 100% dengan melakukan perbandingan antara data latih dan data uji sebesar 90:10 dengan parameter yang ada adalah untuk laju pembelajaran sebesar 0,1, konstanta laju pembelajaran sebesar 0,2, laju pembelajaran minimum sebesar 10^{-5} dan jumlah iterasi maksimum sebanyak 10 (Islamiah et al., 2016).

Pada penelitian ketiga dilakukan oleh Wijayanti terkait dengan perawatan kulit, dengan data yang digunakan merupakan data dari 4 pasien yang pada 44 titik citra yang telah ditentukan adalah citra dari data pasien yang mendapatkan perawatan pada kulit saat sebelum dan sesudah perawatan. Setelah menentukan 44 titik yang telah dibantu oleh pakar, kemudian peneliti melakukan identifikasi perubahan pada dataset yang ada dengan menggunakan fitur RGB (*Red, Green, Blue*) pada bagian kulit yang sebelum dan sesudah mendapatkan perawatan. Sesudah identifikasi dilakukan, penelitian kemudian melanjutkan kedalam tahapan proses LVQ untuk pelatihan data sampai mendapatkan bobot terakhir, dengan parameter yang digunakan dalam metode LVQ adalah maksimum iterasi, nilai *learning rate* dan nilai pengurang *learning rate*. Pada pengujian, peneliti mendapatkan hasil untuk akurasi terbaik adalah sebesar 98,25% untuk 300 iterasi. Sedangkan penambahan parameter baik pada nilai *learning rate*=0,5, nilai pengurang *learning rate*=0,2 serta perbandingan jumlah data latih dan data uji sebesar 18:7 menghasilkan akurasi sebesar 92,31% (Wijayanti et al., 2016).

Penelitian keempat terkait dengan identifikasi penyakit pada tanaman kedelai, peneliti menggunakan citra pada daun dari tanaman kedelai untuk mengidentifikasi adanya penyakit Karat dan penyakit *Downy Mildew* pada tanaman kedelai. Dataset yang didapatkan oleh peneliti sebanyak 96 data dengan masing-masing data dibagi menjadi 2 bagian sejumlah 48 data yang menjadi bagian dari penyakit Karat dan penyakit *Downy Mildew*. Data kemudian diproses dengan cara 30 data digunakan sebagai data latih, 15 data digunakan sebagai data uji dan 3 data digunakan sebagai bobot awal. Peneliti selanjutnya memproses data yang telah didapatkan tersebut untuk kemudian dilakukan tahap ekstraksi pada fitur untuk citra daun tanaman kedelai. Sesudah tahapan tersebut, maka proses selanjutnya adalah masuk pada metode LVQ dengan parameter yang digunakan adalah nilai *learning rate*, minimum *learning rate* serta pengurang *learning rate*. Pada tahap pengujian, peneliti mendapatkan hasil untuk akurasi terbaik sebesar 93% dengan parameter yang digunakan untuk nilai *learning rate*=0,1 dan nilai pengurang *learning rate*=0,1 (Umam et al., 2015).

Penelitian kelima terkait dengan klasifikasi kualitas daun nilam berdasarkan citra digital guna membantu para petani nilam dalam memilih kualitas daun nilam yang baik. Citra digital pada daun nilam akan diproses melalui tahapan pelatihan serta pengujian dengan menggunakan metode LVQ yang mana pada *input* akan diproses citra dari daun nilam serta hasil *output* yang didapatkan dari sistem yang dibangun adalah berupa hasil klasifikasi dari daun nilam. Setelah

proses pengujian sisem dilakukan, didapatkan hasil akurasi untuk klasifikasi daun nilam adalah sebesar 98% dari banyaknya data uji sebanyak 10 dengan jumlah data latih sebanyak 90. Untuk parameter yang digunakan dalam menggunakan metode LVQ sendiri, jumlah iterasi adalah 4, laju pembelajaran 0,1 pengurangan laju pembelajaran 0,3 serta laju pembelajaran minimum adalah 10^{-3} (Krisnanti et al., 2015).

Pada penelitian keenam terkait dengan masalah mutu susu sapi, peneliti menggunakan metode LVQ untuk melakukan klasifikasi guna memberikan hasil yang baik terkait dengan mutu susu sapi. Peneliti mengambil dataset sebanyak 269 data, nantinya data tersebut akan diklasifikasi berdasarkan pada 8 ketentuan atribut yang telah ditentukan oleh pakar dan juga peneliti yakni lemak, SNF (*Solid Non-Fat*), kekentalan susu, protein, laktosa, air dan suhu. Sedangkan 1 atribut kelas lagi akan menentukan kualitas dari susu sapi berdasarkan kualitas yaitu kualitas rendah, kualitas standar dan kualitas tinggi. Pada penelitian, metode LVQ nantinya akan dibantu oleh beberapa parameter yakni iterasi maksimum (*Max Epoch*), minimum *learning rate* (*min α*), *learning rate* (α) dan penurunan *learning rate* (*dec α*). Tahap pengujian pada penelitian dilakukan peneliti dengan mengurutkan dataset serta melakukannya secara *random* untuk dataset yang. Hasil terbaik yang didapatkan untuk masing-masing dataset yang telah diurut dan *dirandom* diperoleh 89,51% dan 91,79% untuk tiap-tiap parameter yang disarankan oleh peneliti berupa *learning rate*=0,1 dan pengali *learning rate*=0,5 sedangkan untuk data yang terurut peneliti menggunakan 70% data latih, iterasi maksimum=12 dan nilai *minimum α* =0,000000001. Untuk data yang *dirandom* peneliti menggunakan 90% data latih, iterasi maksimum=20 dan nilai *minimum α* =0,0000001 (Ganidar et al., 2015).

Pada penelitian ketujuh terkait dengan pengenalan akan bahasa isyarat. Pengenalan bahasa isyarat tersebut dilakukan dengan menggunakan citra digital untuk mendapatkan performa yang baik dengan data yang digunakan adalah berupa data *input* yang akan dimasukan oleh pengguna dan juga terdapat data *output* yang akan menghasilkan teks alphabet atau suara yang sesuai dengan masukan yang diterima oleh sistem yang dibangun. Penulis dalam membangun sistem yang akan digunakan sebagai pengenalan bahasa isyarat memberikan data masukan yang dapat *diinputkan* berupa alphabet (kecuali J dan Z), dengan hasil yang didapatkan oleh penulis untuk hasil akurasi adalah sebesar 88% dengan jumlah data latih yang digunakan sebanyak 288 (12 set) (Prabowo et al., 2015).

Pada penelitian kedelapan terkait dengan pemilihan akan keminatan pada program studi informatika Universitas Brawijaya. Penulis menggunakan metode LVQ untuk melakukan pemilihan keminatan berdasarkan pada data yang didapatkan sebanyak 123 data yang mana data tersebut merupakan hasil kuisisioner. Dair hasil kuisisioner yang didapatkan, penulis kemudian mengubah hasil kuisisioner tersebut menjadi angka yang akan diproses dengan menggunakan metode LVQ dalam proses pelatihan dan juga pengujian. Setelah pelatihan dan pengujian yang dilakukan, hasil akurasi terbaik yang diperoleh adalah sebesar

84,5% dengan jumlah data latih yang digunakan sebanyak 70 (Martinuva et al., 2015). Berikut ini adalah kajian pustaka yang dirangkum pada Tabel 2.1 untuk penelitian-penelitian yang telah diuraikan sebelumnya terkait dengan permasalahan yang dibahas yakni penyakit mata dan metode yang digunakan yaitu *Learning Vector Quantization* (LVQ).

Tabel 2.1 Kajian Pustaka

No.	Judul	Objek	Metode	Hasil
1.	Pemodelan Sistem Pakar Diagnosa Penyakit Mata Menggunakan Metode <i>Naïve Bayes – Certainty Factor</i> - (Lestari et al., 2016)	Objek: Penyakit Mata <i>Input</i> : Gejala penyakit mata yang diderita oleh pengguna	<i>Naïve Bayes – Certainty Factor</i>	Dengan menggunakan metode <i>Naïve Bayes – Certainty Factor</i> , diperoleh hasil akurasi terbaik adalah sebesar 78,4%
2.	Klasifikasi Kualitas Daun Nilam (<i>Pogostemon Cablin Benth</i>) Menggunakan Citra Digital dan Algoritme Learning Vector Quantization - (Krisnanti et al., 2015)	Objek: Daun Nilam <i>Input</i> : Citra Daun Nilam, Ekstraksi Fitur Warna, Binerisasi, serta Ekstraksi Fitur Luas	<i>Learning Vector Quantization</i> (LVQ)	Dengan menggunakan metode LVQ, peneliti mendapatkan hasil akurasi terbaik sebesar 98%
3.	Pengklasifikasian Mutu Susu Sapi Menggunakan Metode <i>Learning Vector Quantization</i> (LVQ) (Studi Kasus: UPT Laboratorium Kesehatan Hewan Malang) - (Ganidar et al., 2015)	Objek: Mutu Kualitas Susu Sapi <i>Input</i> : No, Lemak (<i>Fat</i>), SNF (<i>Solid Non-Fat</i>), Kekentalan (<i>Density</i>), Protein, Laktosa (<i>Lactose</i>), Air (<i>Water</i>), Suhu (<i>Temperature</i>) dan Target (Mutu/ <i>Class</i>)	<i>Learning Vector Quantization</i> (LVQ)	Dengan menggunakan metode LVQ, diperoleh hasil akurasi terbaik untuk data yang diurutkan sebesar 89,51% dan data yang <i>dirandom</i> sebesar 91,79%

Tabel 2.1 Kajian Pustaka (Lanjutan)

No.	Judul	Objek	Metode	Hasil
4.	Pengenalan Bahasa Isyarat Menggunakan Citra Digital dengan Algoritme <i>Learning Vector Quantization</i> - (Prabowo et al., 2015)	Objek: Pengenalan Bahasa Isyarat <i>Input</i> : Citra dalam bentuk tangan yang terdiri atas alphabet sebanyak 24 kecuali huruf J dan huruf Z	<i>Learning Vector Quantization</i> (LVQ)	Dengan menggunakan metode LVQ, diperoleh hasil akurasi terbaik untuk sistem yang dibangun sebesar 88%
5.	Implementasi <i>Learning Vector Quantization</i> (LVQ) dalam Pemilihan Keminatan (Studi Kasus: Program Studi Informatika Universitas Brawijaya) - (Martinuva et al., 2015)	Objek: Pemilihan Keminatan <i>Input</i> : Kuisisioner seputaran dengan nilai dan beberapa pertanyaan yang terkait dengan keminatan yang ada kemudian nantinya akan diisi oleh pengguna	<i>Learning Vector Quantization</i> (LVQ)	Dengan menggunakan metode LVQ, penulis memperoleh hasil akurasi terbaik dari sistem yang telah dibangun sebesar 84,5%
6.	Implementasi Metode <i>Learning Vector Quantization</i> (LVQ) untuk Identifikasi Penyakit pada Citra Daun Tanaman Kedelai - (Umam et al., 2015)	Objek: Penyakit pada Tanaman Kedelai Menggunakan Citra Daun <i>Input</i> : 96 data dengan masing-masing data dibagi menjadi 2 bagian sejumlah 48 data yang menjadi bagian dari penyakit Karat dan penyakit <i>Downy Mildew</i> . Data ini sebelumnya telah diproses dalam ekstraksi fitur untuk mendapatkan citra dalam bentuk angka	<i>Learning Vector Quantization</i> (LVQ)	Dengan menggunakan metode LVQ, didapatkan perolehan hasil akurasi terbaik untuk identifikasi penyakit pada tanaman kedelai melalui citra daun adalah sebesar 93%



Tabel 2.1 Kajian Pustaka (Lanjutan)

No.	Judul	Objek	Metode	Hasil
7.	Klasifikasi Penyakit Gigi dan Mulut Menggunakan Metode <i>Learning Vector Quantization</i> (LVQ) - (Islamiah et al., 2016)	Objek: Penyakit Gigi dan Mulut <i>Input:</i> dataset gejala dengan gejala terdiri atas gigi berlubang, gigi sakit/nyeri ketika menguyah/menggigit, gusi berdarah, gusi kemerahan, adanya karang gigi, gigi ngilu ketika mendapat rangsangan (makan manis/asam atau minum panas/dingin), gusi bengkak, gusi gatal, gigi sisa akar, gigi terasa sakit terus-menerus, gigi goyang, gusi terasa sakit, bau mulut, sakit sampai ke telinga, nyeri spontan pada gigi tanpa rangsangan dan gigi berubah warna abu-abu kehitaman/coklat	<i>Learning Vector Quantization</i> (LVQ)	Dengan menggunakan metode LVQ dapat melakukan klasifikasi pada penyakit Gigi dan Mulut dengan akurasi sebesar 100%
8.	Identifikasi Diagnosi Perubahan Hasil Perawatan Kulit Menggunakan Metode <i>Learning Vector Quantization</i> (LVQ) - (Wijayanti et al., 2016)	Objek: Perawatan Kulit <i>Input:</i> dataset berupa 44 titik dari citra perawatan kulit sebelum dan sesudah perawatan dengan citra. Hasil citra tersebut berupa angka yang akan diproses pada tahapan metode LVQ	<i>Learning Vector Quantization</i> (LVQ)	Dengan menggunakan metode LVQ, hasil akurasi terbaik yang didapatkan untuk melakukan identifikasi terhadap hasil perawatan kulit adalah sebesar 98,25%

Tabel 2.1 Kajian Pustaka

No.	Judul	Objek	Metode	Hasil
9.	Usulan: Identifikasi Penyakit Mata Menggunakan Metode <i>Learning Vector Quantization</i> (LVQ)	Objek: Penyakit Mata <i>Input</i> : Gejala penyakit mata yang diderita oleh pengguna	<i>Learning Vector Quantization</i> (LVQ)	Hasil: Identifikasi berupa penyakit mata yang didapatkan melalui perhitungan dengan menggunakan metode LVQ dan tingkat akurasi sistem

2.2 Penyakit mata

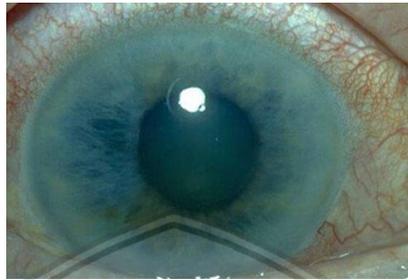
Mata merupakan salah satu indera yang penting bagi manusia, melalui mata manusia menyerap informasi visual yang digunakan untuk melakukan kegiatan sehari-hari (Perdami, 2013). Dengan informasi visual yang diterima, manusia dapat berinteraksi dengan sekitarnya. Namun gangguan terhadap mata yang memengaruhi penglihatan banyak terjadi, mulai dari gangguan dalam tingkatan ringan sampai kepada gangguan tingkat yang berat sehingga mengakibatkan kebutaan (Perdami, 2013). Berikut ini adalah beberapa penjelasan mengenai jenis-jenis penyakit mata yakni:

2.2.1 Glaukoma

Glaukoma merupakan salah satu penyakit yang menyerang mata. Glaukoma adalah suatu keadaan dengan tekanan bola mata yang merusak saraf optik (Ilyas, 2003). Penyebab gangguan penglihatan terbanyak dan penyebab kebutaan terbanyak diseluruh dunia salah satunya adalah glaukoma (Perdami, 2013). Glaukoma yang menunjukkan suatu keadaan dengan tekanan bola mata yang lebih tinggi dari normal sehingga mengakibatkan kerusakan saraf penglihatan dan bisa mengakibatkan kebutaan. Normalnya, tekanan bola mata berkisar antara 15-20 mmHg. Terjadinya glaukoma bisa disebabkan oleh beberapa sebab antara lain adalah cairan mata (*Humor aqueus*) selalu mengalir dan terjadi gangguan pengeluaran cairan menyebabkan *Humor aqueus* menjadi banyak dan menyebabkan tekanan bola mata menjadi tidak normal (Reztaputra, 2012).

Pada kasus glaukoma, terjadinya penyempitan pada pandangan bisa terjadi karena adanya tekanan pada bola mata yang meningkat, kemudian berlanjut kepada pembuluh darah yang seharusnya menyuplai nutrisi yang cukup namun tidak mendapatkan secara cukup. Kerusakan pada serabut saraf juga terjadi secara bertahap sampai pada titik keseluruhan penglihatan akhirnya menghilang.

Faktanya di Indonesia, 700.000 penderita penyakit mata adalah kasus glaukoma yang mana sepertiganya buta dan dua per tiganya terancam buta. Glaukoma jika dibiarkan terus-menerus tanpa mendapatkan penanganan yang baik dan tepat akan berakhir kepada kebutaan sehingga diagnosis dan pengobatan yang lebih dini dapat membantu dalam mengontrol penyakit dan mengurangi ancaman kebutaan (Reztaputra, 2012). Glaukoma yang menjadi gangguan pada mata ditunjukkan pada Gambar 2.1.



Gambar 2.1 Glaukoma

Sumber: (Dahl, 2009d)

Glaukoma sendiri memiliki beberapa jenis yang dapat dikategorikan yakni adalah sebagai berikut (Reztaputra, 2012):

a. Glaukoma Kongenital

Jenis glaukoma ini terjadi sejak lahir dengan terdapatnya ketidaksempurnaan pada perkembangan saluran *Humor aqueus* pada saat didalam kandungan (pada masa janin). Gejala yang paling menonjol ditunjukkan oleh jenis glaukoma kongenital adalah biasanya penderita sangat peka terhadap rangsangan cahaya, mata berwarna merah dan kornea pada bola mata membesar dari ukuran normal.

b. Glaukoma Akut

Jenis glaukoma ini terjadi dikarenakan sudut sempit yang terjadi diantara iris pada bola mata dan juga kornea pada bola mata. Selain itu, iris pada bola akan bergeser dari tempatnya sehingga menutup pangkal saluran *Humor aqueus*. Gejala yang paling menonjol ditunjukkan oleh jenis glaukoma akut adalah biasanya penderita merasakan nyeri pada mata, mata berwarna merah, mual serta pandangan menjadi kabur dalam waktu yang relative cepat.

c. Glaukoma Kronis

Jenis glaukoma ini terjadi dikarenakan penyempitan pada saluran keluar dari *Humor aqueus*. Tipe glaukoma kronis merupakan salah satu jenis penyakit glaukoma yang paling sering dijumpai dan parahnya gejala-gejala yang ditunjukkan oleh glaukoma kronis tidak terlalu terlihat/nampak sehingga penderita baru akan menyadari hal tersebut jika penglihatannya sudah mulai kabur dan tidak jelas.

d. Glaukoma Sekunder

Jenis glaukoma ini terjadi dikarenakan adanya pemicu dari penyakit lain yang menyebabkan jenis glaukoma sekunder ini muncul. Pemicu dari penyakit lain ini seperti misalnya penyakit uveitis, diabetes melitus dan juga pengaruh terhadap pemakaian obat-obatan yang dikonsumsi oleh penderita.

2.2.2 Katarak

Katarak berasal dari bahasa Yunani yaitu *Katarrhakies*, dalam bahasa Inggris yakni *Cataract* serta dalam bahasa Latin yaitu *Cataracta* yang artinya adalah air terjun (Ilyas, 1998). Pengertian air terjun sendiri dikarenakan perubahan pada lensa mata, yang awalnya semuanya jernih dan tembus akan rangsangan cahaya namun perlahan-lahan berubah menjadi keruh dan berkabut seolah-olah penderita melihat dari balik air terjun. Katarak bisa menjadi berbahaya yang dapat mengakibatkan kebutaan dikarenakan katarak membuat perubahan kimia yang terjadi sehingga lensa pada bola mata yang awalnya bening menjadi keruh. Lensa yang keruh akan membuat penderita sulit untuk merasakan rangsangan cahaya, sehingga retina akan menangkap informasi visual secara kabur (Reztaputra, 2012).

Jika sudah dalam tahap yang parah, informasi visual yang awalnya diterima kabur lambat laun akan berubah menjadi parah dan berujung kepada kebutaan pada penderita. Gejala yang terjadi pada katarak terbagi menjadi dua bagian yakni gejala yang bersifat subyektif dan gejala yang bersifat obyektif. Untuk gejala secara subyektif, biasanya gejala awal untuk tanda-tanda munculnya katarak ditandai dengan penglihatan ganda yang dialami oleh penderita, kemudian disusul rangsangan pada cahaya yang semakin peka sehingga membuat penderita melihat pada saat malam hari terasa lebih nyaman. Adapun gejala lain yang ditunjukkan adalah seperti penglihatan yang kemudian mengalami penurunan secara bertahap. Sedangkan untuk gejala secara obyektif ditandai dengan kekeruhan pada lensa di bola mata dalam berbagai bentuk serta tingkatan yang dapat dilihat (Reztaputra, 2012). Katarak yang menjadi gangguan pada mata ditunjukkan pada Gambar 2.2.



Gambar 2.2 Katarak

Sumber: (Benelli, 2009)

Katarak sendiri memiliki beberapa jenis yang dapat dikategorikan sebagai berikut (Reztaputra, 2012):

a. Katarak Kongenital

Jenis katarak ini terjadi dikarenakan bawaan sejak lahir. Biasanya gejala-gejala yang ditunjukkan oleh jenis katarak kongenital adalah disebabkan oleh infeksi yang berasal dari virus maupun parasit saat ibu hamil muda (pada masa janin) sehingga mengakibatkan penderita mengalami katarak kongenital.

b. Katarak Senilis

Jenis katarak ini terjadi dikarenakan usia lanjut yang dialami oleh manusia yang mengalami penuaan. Biasanya katarak senilis muncul pada penderita yang usianya sudah menginjak umur paling cepat 40 tahun.

c. Katarak Traumatik

Jenis katarak ini terjadi dikarenakan trauma yang didapatkan oleh mata. Bisa saja mata mendapatkan cedera atau bentuk gangguan yang mengakibatkan mata mengalami rasa trauma yang menyebabkan katarak ini muncul pada penderita.

d. Katarak Komplikata

Jenis katarak ini terjadi dikarenakan infeksi berupa parasit atau virus namun bisa juga dikarenakan penyakit tertentu yang diderita pasien mengakibatkan katarak komplikata muncul. Biasanya jenis katarak ini muncul pada beberapa jenis penyakit yang membuat mata ikut mengalami gangguan misalnya penyakit diabetes melitus.

2.2.3 Konjungtivitis

Konjungtivitis terjadi pada bagian bola mata yang bernama konjungtiva. Konjungtiva sendiri merupakan selaput lendir yang melapisi pada bagian kelopak mata dan melipat kedalam bola mata yang berguna untuk melapisi bola mata pada bagian depan. Konjungtivitis terjadi dikarenakan peradangan yang terjadi pada bagian konjungtiva akibat infeksi berupa parasit atau virus. Selain itu juga terjadinya peradangan pada konjungtiva bisa diakibatkan dari beberapa faktor seperti penyakit lain, asap atau ransangan sinar yang diterima terlalu kuat sehingga mata tidak mampu menerimanya secara normal. Pada umumnya, gejala yang paling terlihat pada konjungtivitis adalah (Reztaputra, 2012):

- a. Mata merah dikarenakan injeksi konjungtiva (biasanya gejala-gejala kecil mengikuti seperti bola mata terasa panas, gatal dan terasa sakit).
- b. Mata berair, dikarenakan peradangan yang terjadi pada konjungtiva yang menyebabkan mata terasa panas membuat mata menjadi sering berair.
- c. Eksudasi (belekan).
- d. Pseudoptosis (mata yang susah dibuka dikarenakan otot Muller yang mengalami gangguan karena peradangan yang terjadi pada konjungtiva).

Biasanya konjungtivitis terjadi pada kedua mata penderita. Hal ini disebabkan pada saat mata satu mengalami peradangan konjungtiva maka mata sebelahnya akan terkena juga. Namun ada beberapa kasus yang hanya terjadi pada satu mata dan hal tersebut bukanlah konjungtivitis melainkan mata mengalami alergi (Reztaputra, 2012). Konjungtivitis yang menjadi gangguan pada mata ditunjukkan pada Gambar 2.3.



Gambar 2.3 Konjungtivitis

Sumber: (Dahl, 2009a)

2.2.4 Keratitis

Keratitis atau ulkus kornea merupakan infeksi yang terjadi pada kornea dibola mata yang disebabkan oleh beberapa faktor yakni bakteri, parasit ataupun virus (Perdami, 2013). Pada umumnya, keratitis akan mengganggu penglihatan yang terlatak dipusat bola mata. Tipe dari keratitis sendiri ada dua yakni infeksi dan non-infektif.

Penderita yang bisa berisiko terkena penyakit keratitis adalah orang-orang yang memiliki riwayat trauma, pengguna lensa kontak (baru pakai tidak tahu caranya; sudah lama pakai tapi tidak tahu cara merawatnya; dipakai dalam waktu yang cukup lama), menggunakan tetes mata steroid dalam jangka waktu yang cukup lama, terdapat riwayat mata merah dan mata yang menjadi buram saat melihat secara timbul tenggelam serta mata kadang menangkap rangsangan cahaya yang silau serta mata biasanya menunjukkan tanda-tanda berair, belean dan sering kelilipan (Reztaputra, 2012). Keratitis yang menjadi gangguan pada mata ditunjukkan pada Gambar 2.4.



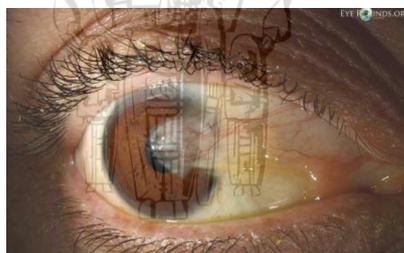
Gambar 2.4 Keratitis

Sumber: (Upadhyay, Srinivasan, & Whitcher, 2015)

2.2.5 Pterigium

Pterigium merupakan jenis penyakit yang menyerang pada mata dengan kondisi mata akan ditutupi oleh selaput pada bagian putih dibola mata dan mengalami pembesaran yang menuju kearah kornea mata yang mengakibatkan penderita mengalami gangguan dalam penglihatan. Umumnya, pterigium terdapat pada dua mata atau hanya salah satu mata yang terserang. Secara garis besar, belum dapat dipastikan penyebab munculnya pterigium dikarenakan sebab apa, namun penderita yang terserang penyakit mata jenis tersebut yang sering melakukan kegiatan diluar (kegiatan *outdoor*). Risiko munculnya pterigium bisa terjadi dikarenakan mata yang mengalami banyak gangguan atau terserang oleh beberapa bakteri, virus maupun parasit pada saat penderita berada diluar ruangan dan melakukan banyak aktivitas (Reztaputra, 2012).

Gejala yang paling utama terlihat pada pterigium adalah selaput yang muncul dan tumbuh dan kemudian merambah kebola mata. Gejala-gejala lainnya mengikuti seperti gejala pada penyakit mata lainnya, seperti mata yang berwarna merah, bisa dikarenakan terkena asap atau berbagai macam gangguan yang membuat mata berwarna merah, kemudian diikuti oleh gejala iritasi serta terasa panas sehingga mata akan berair. Selain gejala tersebut, gejala-gejala ringan yang biasanya terjadi pada penyakit mata seperti rasa gatal, perih, mata yang mengalami keganjalan karena ada sesuatu dimata (dirasakan) serta pandangan mata yang tidak mampu melihat secara tajam (Reztaputra, 2012). Pterigium yang menjadi gangguan pada mata ditunjukkan pada Gambar 2.5.



Gambar 2.5 Pterigium

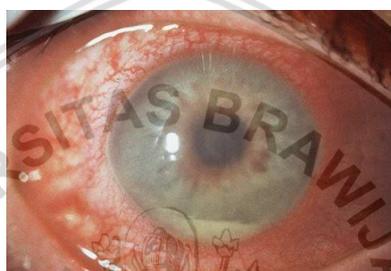
Sumber: (Vislisel, 2015)

2.2.6 Uveitis

Uveitis merupakan jenis penyakit pada mata yang muncul akibat peradangan yang terjadi pada bagian bola mata yakni peradangan pada bagian uvea. Namun, pengaruh atas peradangan yang terjadi pada uvea dapat berakibat kepada bagian koroid dan juga iris pada waktu yang bersamaan. Perbedaan dari penyakit mata lainnya adalah peradangan yang terjadi pada bagian uvea bukanlah peradangan yang menyebar dengan cara menular, melainkan dimulai semenjak peradangan uvea timbul dan berkelanjutan sehingga membawa perubahan terhadap sistem kekebalan tubuh penderita. Uveitis sendiri terbagi atas dua jenis yakni uveitis anterior dan uveitis posterior. Pada uveitis anterior, gejala yang timbul disebabkan oleh infeksi yang muncul karena ada penyakit lain yang diderita oleh penderita seperti penyakit sifilis, tuberkolosis dan lain-lain.

Sedangkan gejala-gejala pada umumnya seperti mata yang berwarna merah, nyeri pada mata serta pandangan yang mulai kabur (Reztaputra, 2012).

Pada uveitis posterior, jenis ini ada dikarenakan jenis-jenis penyakit yang menyerang penderita untuk jenis penyakit jaringan ikat/*autoimun* (salah satunya adalah penyakit HIV/AIDS). Sedangkan untuk gejala-gejala secara umum yang timbul akibat uveitis posterior adalah penderita yang merasakan mata terasa tenang, penderita akan mengeluhkan rasa mengganjal yang dirasakan didalam bola mata, penglihatan yang mulai tidak jelas/buram, sensitive terhadap rangsangan cahaya serta mata yang berwarna merah (Reztaputra, 2012). Namun, secara umum 30-40% pasien penderita uveitis tidak dapat didiagnosis. Oleh sebab itu diusulkan untuk melakukan berbagai jenis pemeriksaan terhadap mata untuk menemukan secara tepa tapa penyebab sebenarnya uveitis muncul. Uveitis yang menjadi gangguan pada mata ditunjukkan pada Gambar 2.6.



Gambar 2.6 Uveitis

Sumber: (Dahl, 2009e)

2.2.7 Dry Eyes

Dry eyes atau yang biasanya disebut sebagai mata kering terjadi karena terdapat kondisi mata yang tidak mendapatkan cairan maupun pelumas yang cukup untuk mata. Dalam takaran normal, seharusnya mata mendapatkan pelumas dan cairan agar mata tetap lembab dan hal itu membuat mata terasa nyaman. Namun, beberapa faktor membuat kualitas dari pelumas serta cairan mata kurang menjadi kualitas yang cukup baik dan menyebabkan mata tidak memiliki pelumas dan cairan yang efisien sehingga mengarah kepada *dry eyes* (mata kering) (Javadi & Feizi, 2011).

Air mata sendiri diproduksi dengan dua cara, pertama air mata akan terus-menerus diproduksi dengan sendirinya dan kedua air mata yang diproduksi secara berlebihan dikarenakan beberapa alasan seperti menangis, udara yang dingin, mata yang terkena angin serta kemasukan benda asing sehingga menyebabkan mata akan mengeluarkan air mata (Javadi & Feizi, 2011).

Gejala-gejala secara umum yang akan dialami ketika penderita terserang *dry eyes* adalah rasa panas dibola mata, gatal, memiliki lender yang berlebihan pada mata, sekitaran bola mata akan berwarna merah, mata akan terasa berat dan lelah serta munculnya perasaan tidak nyaman pada saat penggunaan lensa kontak. Penyebab lainnya adalah dikarenakan pertambahan usia pada seseorang yang mengakibatkan jumlah air mata pun ikut berkurang. Beberapa faktor

lainnya juga dikarenakan penderita *dry eyes* yang mengonsumsi obat-obatan yang dapat mengurangi jumlah air mata dan menyebabkan mata kering, seperti obat *Diuretical*, *Beta Blockers* serta *Antihistaminics* (Javadi & Feizi, 2011). *Dry eyes* yang menjadi gangguan pada mata ditunjukkan pada Gambar 2.7.



Gambar 2.7 Dry Eyes

Sumber: (Rose, 2017)

2.2.8 Bleparitis

Bleparitis adalah salah satu kondisi yang terjadi pada mata akibat adanya peradangan dibagian kelopak mata dan margo palpebra. Bleparitis memengaruhi kelopak mata dekat pangkal bulu mata. Kondisi peradangan yang terjadi menyebabkan mata penderita akan mengalami beberapa kondisi yakni mata yang berair, mata yang berwarna merah, mata yang terasa terbakar/panas, menyengat, kelopak mata terlihat berminyak serta bulu mata yang mulai berkerak dan sisik mata seperti memiliki kotoran. Bleparitis disebabkan karena adanya bakteri *staphylococcus* (Ilyas, 2003).

Akibat adanya jenis penyakit ini, hilangnya bulu mata bisa terjadi, kehilangan penglihatan pun bisa terjadi dikarenakan semakin parah tanpa penanganan lebih lanjut atau pencegahan awal yang bisa dilakukan. Selain itu juga, bleparitis mengakibatkan pertumbuhan pembuluh darah baru pada bagian kornea, styes dan mengakibatkan mata kering/*dry eyes* (Lindsley, Matsumura, Hatef, & Akpek, 2014). Bleparitis yang menjadi gangguan pada mata seperti yang telah diuraikan sebelumnya ditunjukkan pada Gambar 2.8.



Gambar 2.8 Bleparitis

Sumber: (Dahl, 2009b)

2.2.9 Kalazion

Kalazion merupakan salah satu penyakit yang terjadi dikarenakan terdapat peradangan pada kelenjar Meibom. Peradangan tersebut mengakibatkan kelenjar Meibom tersumbat sehingga infeksi yang seharusnya ringan mampu menjadi peradangan yang berada dalam tingkat kronis jika tidak ditangani dengan baik dan benar. Kalazion seperti sebuah benjolan yang terdapat disekitaran kelopak mata yang berisi kelenjar-kelenjar yang tidak dapat keluar karena tersumbat. Gejala yang paling terlihat adalah pada kelopak mata mengalami pembengkakan, kemudian disusul dengan rasa nyeri pada bagian kelopak mata. Sedangkan gejala-gejala secara umum yang dapat dirasakan oleh penderita adalah rasa iritasi, penglihatan menjadi buram, otot-otot pada mata terasa melemah dan terjadi perubahan bentuk pada bola mata (Ilyas, 1998). Kalazion yang menjadi gangguan pada mata ditunjukkan pada Gambar 2.9.



Gambar 2.9 Kalazion

Sumber: (Dahl, 2009c)

2.3 Jarak *Euclidean*

Jarak *Euclidean* digunakan untuk menghitung jarak pada data numerik yang akan dihitung. Persamaan dari jarak *Euclidean* menjadi perhitungan untuk mencari nilai yang dekat antara jarak dari variabel yang telah ditentukan untuk diproses dalam persamaan *Euclidean* yang adalah sebagai berikut (Greenacre & Primicerio, 2013):

$$d_{x,y} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (2.1)$$

Keterangan:

$d_{x,y}$ = *Euclidean distance* (Jarak *Euclidean*)

x = vektor *input*

y = vektor *database*

n = banyaknya jumlah vector

2.4 Jaringan saraf tiruan

Jaringan saraf tiruan merupakan sistem pengolahan informasi yang memiliki sebuah karakter yang kinerjanya sama dengan jaringan saraf biologis. Jaringan

saraf tiruan sendiri telah dikembangkan sebagai sebuah model cara kerja yang proses memperoleh pengetahuan serta usaha untuk mengenali sesuatu berdasarkan pengalaman yang diambil dari cara kerja manusia atau saraf biologis dari manusia, dengan asumsi bahwa (Fausett, 1994):

1. Pengolahan informasi atau masuknya informasi serta proses yang terjadi pada banyak elemen sederhana disebut sebagai *neuron*.
2. Terdapat penghubung antar satu *neuron* dengan *neuron* lainnya sehingga sinyal yang di alirkan di antara *neuron-neuron* tersebut dapat terkoneksi dengan baik hingga membentuk pola antar *neuron*.
3. Setiap koneksi yang sudah terjalin atau terhubung mempunyai bobot terkait pada masing-masing sambungan yang ada. Bobot ini nantinya akan digunakan untuk mengalikan sinyal yang di kirimkan sebagai penerus kepada *neuron* lainnya untuk menentukan dan mengubah bobot (biasanya disebut sebagai *learning*).
4. Setiap *neuron* yang ada menerapkan sebuah fungsi aktivasi (biasanya non-linier) pada sinyal masukan (*input*) guna menentukan sinyal keluaran (*output*) berdasarkan pada jumlah dari bobot sinyal masukan (*input*) yang telah dihitung pada *neuron* sebelumnya.

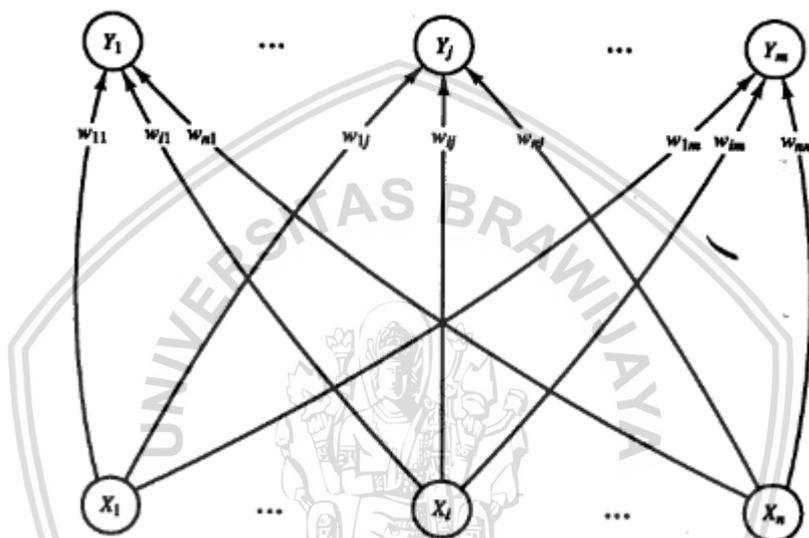
Selain itu, jaringan saraf juga memiliki karakteristik atau sifat yakni (Fausett, 1994):

1. Adanya pola dari koneksi antara *neuron-neuron* (disebut sebagai arsitektur).
2. Adanya metode yang digunakan untuk menentukan bobot dari koneksi (disebut sebagai pelatihan, *learning*, atau algoritme).
3. Adanya fungsi aktivasi

Jaringan saraf sendiri terdiri atas sejumlah besar elemen pemrosesan yang sederhana (disebut sebagai *neuron*, *unit*, *cell* atau *node*). Setiap *neuron* terhubung dengan *neuron* yang lainnya secara terarah, yang kemudian terdapat bobot yang terkait pada tiap-tiap *neuron*. Bobot tersebut merepresentasikan informasi yang akan digunakan untuk menyelesaikan persoalan atau permasalahan yang diproses. Jaringan saraf sebenarnya dapat diterapkan dalam berbagai persoalan seperti misalnya menyimpan dan memanggil kembali data atau pola, mengklasifikasikan pola, melakukan pemetaan umum terhadap pola *input* ke pola *output*, mengelompokkan pola-pola yang sama atau bisa juga menemukan solusi pada masalah pengoptimalan yang terbatas. Setiap *neuron* mempunyai keadaan internal (biasanya disebut sebagai aktivasi atau tingkat aktivasi), yang merupakan sebuah fungsi dari *input* yang diterima. Sederhananya, sebuah *neuron* mengirimkan fungsi aktivasi sebagai sinyal untuk *neuron* yang lainnya. Hal ini penting untuk menjadi sebuah catatan sebab sebuah *neuron* dapat mengirim satu sinyal pada satu waktu, meskipun sinyal tersebut dapat tersebar ke *neuron-neuron* yang lainnya (Fausett, 1994).

2.5 Learning Vector Quantization (LVQ)

Learning Vector Quantization atau LVQ merupakan suatu metode dalam melakukan klasifikasi pada pola untuk tiap-tiap unit yang ada dengan *output* yang akan mewakili kelas tertentu atau kategori yang telah ditentukan yang mana unit *output* dapat digunakan pada tiap-tiap kelas yang ada. Kelas-kelas yang telah diperoleh nantinya akan tergantung kepada jarak antar vektor-vektor *input*. LVQ masuk dalam kategori *single layer*, artinya hanya memiliki *layer input* dan *layer output* yang saling terhubung dengan adanya suatu bobot (*weight*). Berikut ini adalah gambaran struktur jaringan mengenai LVQ yang ditunjukkan oleh Gambar 2.10 (Fausett, 1994).



Gambar 2.10 Jaringan Saraf pada *Learning Vector Quantization* (LVQ)

Sumber: (Fausett, 1994)

Dengan menggunakan metode LVQ, disini representasi dari keluaran/*output* akan menjadi kategori atau kelas. Oleh sebab itu, secara garis besar algoritme LVQ akan diuraikan sebagai berikut (Fausett, 1994):

1. Langkah 0

Menetapkan/menginisialisasi bobot awal variabel *input* ke-*j* menuju kepada kelas atau kluster yang telah didapatkan berdasarkan ke-*i*, dengan W_{ij} menunjukkan $i=1, 2, \dots, m$, kemudian menginisialisasi juga untuk iterasi maksimum, parameter *learning rate* (α), pengurangan *learning rate* (Deca) dan minimal *learning rate* yang diperbolehkan (Min α).

2. Langkah 1

Untuk langkah ke-2 sampai langkah ke-6 akan diulangi sampai kategori kelas diperoleh.

3. Langkah 2

Pada setiap vektor *input* yang ada, untuk data *training*-nya (x) akan melakukan proses pada langkah ke-3 sampai ke-4.

4. Langkah 3

Melakukan perhitungan terhadap j minimum dengan menggunakan rumus jarak yakni *Euclidean distance*.

5. Langkah 4

Melakukan *update* kepada W_j untuk memperbaiki W_j dengan beberapa ketentuan, yakni:

$$\text{- Jika } T = C_j \text{ maka } W_j = W_j + \alpha (X_i - W_j) \quad (2.2)$$

$$\text{- Jika } T \neq C_j \text{ maka } W_j = W_j - \alpha (X_i - W_j) \quad (2.3)$$

6. Langkah 5

Melakukan pengurangan pada nilai α dengan cara sebagai berikut.

$$\text{- } \alpha = \alpha - \text{Dec}\alpha \text{ atau}$$

$$\text{- } \alpha = \alpha - \alpha * \text{Dec}\alpha$$

7. Langkah 6

Pemberhentian proses pada metode LVQ terjadi dikarenakan iterasi yang berhenti pada iterasi tertentu atau kondisi parameter *learning rate* mampu mendekati pada nilai *learning rate minimum*.

Keterangan:

X_i = vektor masukan pada data latih (x_1, x_2, \dots, x_n)

T = kelas pada vektor yang *ditraining*

W_j = bobot pada vektor untuk keluaran/*output* ke- j

C_j = kelas/kategori yang didapatkan oleh hasil keluaran pada *output* ke- j

2.6 Pengujian

Pengujian pada penelitian ini dilakukan untuk memperoleh hasil terbaik pada setiap parameter yang ada pada metode LVQ. Dari setiap parameter yang ada akan dilihat nilai terbaik melalui akurasi yang diperoleh untuk nantinya digunakan nilai tersebut pada pengujian perbandingan jumlah data latih dan data uji untuk mengetahui tingkat akurasi yang didapatkan oleh sistem. Pengujian pada metode LVQ untuk mendapatkan kombinasi yang baik juga akan dilakukan dengan menggunakan cara yakni, *random stratified* (pengambilan data secara acak yang melibatkan populasi ke dalam kelompok yang lebih kecil berdasarkan atribut atau karakteristik data) dan *repeated holdout* (Nugroho, Sulisty, & Saptono, 2016). Adapun pengujian yang dilakukan meliputi:

1. Pengujian Akurasi

Pengujian akurasi dilakukan dengan cara membandingkan jumlah hasil identifikasi yang dilakukan oleh sistem dengan hasil identifikasi yang dilakukan secara manual atau hasil identifikasi yang berasal dari pakar. Nilai akurasi dihitung dengan menggunakan Persamaan 2.4 sebagai berikut:

$$Akurasi = \frac{\text{jumlah data yang benar}}{\text{total semua data yang ada}} \times 100\% \quad (2.4)$$

2. Pengujian *Learning Rate* (α)

Pengujian *learning rate* (α) dilakukan untuk mendapatkan nilai terbaik pada parameter ini dengan cara mengubah nilai *learning rate* (α) dalam cakupan nilai tertentu, kemudian membandingkan hasil akurasi yang didapatkan dari setiap perubahan nilai *learning rate* (α) yang telah diterapkan. Hasil yang diambil berdasarkan pada nilai akurasi tertinggi untuk nilai parameter *learning rate* (α) yang nantinya akan digunakan pada pengujian parameter berikutnya (Dec α , Min α , Max Epoch dan perbandingan jumlah data latih dan data uji).

3. Pengujian Pengurangan *Learning Rate* (Dec α)

Pengujian pengurangan *learning rate* (Dec α) tujuannya adalah untuk mendapatkan nilai terbaik pada parameter tersebut. Pengujian ini dilakukan hampir sama dengan pengujian sebelumnya dengan mengubah nilai pengurangan *learning rate* (Dec α) dalam cakupan tertentu, kemudian hasil yang didapatkan akan dianalisis untuk melihat hasil akurasi yang paling tinggi. Nilai yang paling tinggi nantinya akan diterapkan pada pengujian parameter LVQ berikutnya (Min α , Max Epoch dan perbandingan jumlah data latih dan data uji).

4. Pengujian *Minimal Learning Rate* (Min α)

Pengujian *minimal learning rate* (Min α) dilakukan untuk menguji nilai parameter tersebut agar mendapatkan nilai yang terbaik. Pengujian *minimal learning rate* (Min α) diuji dengan mengubah nilai parameter tersebut dalam cakupan nilai tertentu, kemudian hasilnya akan diambil berdasarkan pada akurasi tertinggi yang diperoleh. Nilai yang sudah didapatkan akan digunakan pada pengujian selanjutnya (Max Epoch dan perbandingan jumlah data latih dan data uji).

5. Pengujian Iterasi Maksimum (Max Epoch)

Pengujian iterasi maksimum (Max Epoch) dilakukan untuk mendapatkan analisis pada parameter iterasi maksimum (Max Epoch). Analisis tersebut dilakukan untuk memperoleh nilai terbaik pada pengujian ini dengan mengubah nilai parameter dalam cakupan tertentu. Hasil yang diperoleh berupa nilai iterasi maksimum (Max Epoch) dengan akurasi terbaik. Nilai ini nantinya akan digunakan pada pengujian berikutnya (perbandingan jumlah data latih dan data uji).

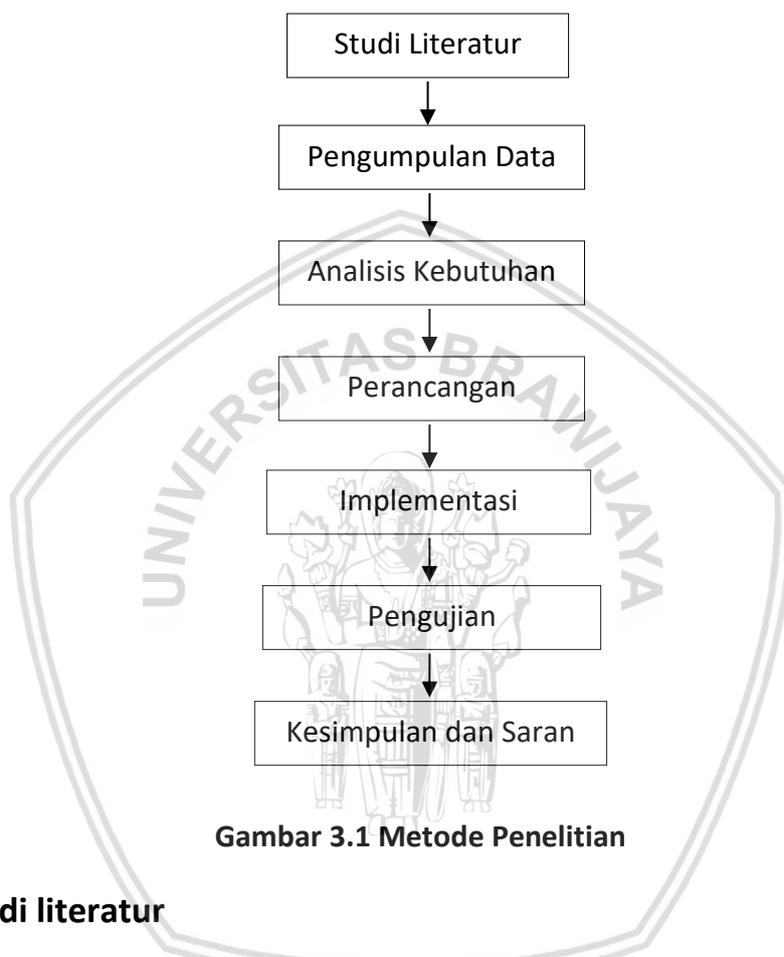
6. Pengujian Perbandingan Jumlah Data Latih dan Jumlah Data Uji

Pengujian ini dilakukan dengan cara mengambil keseluruhan nilai terbaik dari masing-masing parameter yang telah diuji sebelumnya. Pengambilan nilai tersebut, diharapkan memberikan akurasi terbaik pada saat melakukan pengujian terhadap perbandingan jumlah data latih dan jumlah data uji. Hasil terbaik yang didapatkan pada pengujian ini akan digunakan sebagai acuan untuk mendapatkan nilai rata-rata akurasi terbaik dan juga akurasi tertinggi pada sistem.



BAB 3 METODOLOGI

Pada bab ini akan dijelaskan mengenai tahapan alur metode yang akan digunakan dalam melakukan penelitian pada identifikasi penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ). Alur metode yang digunakan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Metode Penelitian

3.1 Studi literatur

Pada tahap studi literatur, proses ini membahas tentang dasar teori yang akan digunakan sebagai sumber pada penelitian yang akan dikerjakan. Dasar teori yang ada memberikan dukungan pada penulisan serta pemahaman pada penelitian yang dikerjakan, dengan studi literatur yang bisa diambil dari beberapa acuan seperti buku, jurnal, paper, internet dan sumber pustaka ilmiah yang terkait dengan penelitian yang sedang dikerjakan. Dasar teori yang bisa digunakan seperti:

1. Mata
2. Penyakit dan gejala pada mata
3. Jaringan saraf tiruan
4. *Learning Vector Quantization* (LVQ)

3.2 Pengumpulan data

Pada tahap pengumpulan data, data yang digunakan merupakan data sekunder. Data sekunder merupakan data yang didapatkan dari penelitian orang lain, namun dapat digunakan untuk melakukan penelitian selanjutnya mengenai data terkait yang bisa digunakan kembali. Data sekunder diperoleh dari poli mata RSUD Kabupaten Sidoarjo tahun 2015-2016 (Lestari et al., 2016).

3.3 Analisis kebutuhan

Pada tahap analisis kebutuhan, proses ini dilakukan untuk mendata dan memahami seluruh kebutuhan yang harus dipenuhi oleh sistem yang nantinya akan dirancang dan diimplementasikan. Adapun kebutuhan yang harus dipenuhi sistem identifikasi penyakit mata dengan menggunakan metode LVQ adalah menentukan kebutuhan fungsional dan non fungsionalnya.

3.3.1 Kebutuhan Fungsional

Adapun pada kebutuhan fungsional, hal yang dilakukan nantinya pada sistem identifikasi penyakit mata dengan menggunakan metode LVQ adalah:

1. Sistem mampu mengambil dan menampilkan data latih yang akan diproses pada pelatihan dengan menggunakan metode LVQ.
2. Sistem mampu menampilkan hasil pemrosesan data pada saat pelatihan dengan menggunakan metode LVQ yaitu dengan menampilkan bobot jaringan yang diperoleh (w), perhitungan jarak minimum (D), dan perubahan bobot jaringan (Δw).
3. Sistem mampu menyimpan nilai bobot terakhir dari proses pelatihan LVQ untuk nantinya akan digunakan pada proses pengujian LVQ.
4. Sistem mampu mengambil dan menampilkan data uji yang akan diproses pada pengujian dengan menggunakan metode LVQ.
5. Sistem mampu menampilkan hasil pemrosesan data pada saat pengujian dengan menggunakan metode LVQ yaitu dengan menampilkan bobot jaringan akhir (w) yang diperoleh pada saat proses pelatihan LVQ, perhitungan jarak minimum (D), dan hasil klasifikasi yang diperoleh untuk melihat berapa data yang sesuai dan data yang tidak sesuai.
6. Sistem mampu menampilkan hasil identifikasi berdasarkan *inputan user* pada saat pengguna memilih gejala serta penyakit. Kemudian sistem nantinya akan menampilkan hasil akhir berupa hasil identifikasi yang didapatkan berdasarkan perhitungan pada jarak minimum (D) untuk melihat *output* jaringan dan target yang diharapkan.

3.3.2 Kebutuhan Non Fungsional

Pada kebutuhan non fungsional, berfokus kepada properti, baik itu perangkat keras maupun perangkat lunak yang harus dipenuhi untuk digunakan pada tahapan implementasi sistem. Berikut ini adalah spesifikasi untuk perangkat

keras yang akan digunakan dalam penelitian ini yang disajikan pada Tabel 3.1 dan spesifikasi perangkat lunak yang disajikan dalam Tabel 3.2.

Tabel 3.1 Spesifikasi Perangkat Keras

No	Nama Komponen	Spesifikasi
1	Processor	Intel Core i3-3217U CPU @ 1.80GHz (4 CPUs), ~1.80GHz
2	Memory (RAM)	10 GB
3	Hardisk	500 GB
4	Graphics Card	Intel HD Graphics 4000
5	Chip Type	Intel HD Graphics Family

Tabel 3.2 Spesifikasi Perangkat Lunak

No	Nama Komponen	Spesifikasi
1	Sistem Operasi	Windows 10 Pro 64-bit
2	Bahasa Pemrograman	Java
3	Media Pemrograman	Netbeans v.8.2

3.4 Perancangan

Pada tahap perancangan, proses yang akan dilakukan adalah:

1. Perancangan manualisasi. Perancangan ini akan berpusat kepada masalah perhitungan pada masalah yang diangkat sebagai objek penelitian. Dataset yang digunakan seperti yang telah dijelaskan pada sub bab pengumpulan data, dengan data penyakit mata yang digunakan pada tahun 2015-2016 dan metode *Learning Vector Quantization*. Pada akhirnya, dari perancangan manualisasi tersebut akan menghasilkan algoritme dan diagram alir sebagai solusi untuk menyelesaikan permasalahan pada objek yang diangkat dalam penelitian.
2. Perancangan pengujian. Perancangan ini akan berpusat kepada pengujian metode *Learning Vector Quantization* dengan cara melakukan pengujian terhadap masing-masing parameter yang mendukung metode tersebut, yakni iterasi maksimum, *learning rate*, pengurangan *learning rate*, *minimum learning rate* dan juga perbandingan antara data latih dan data uji.

3.5 Implementasi

Pada tahap implementasi, tahap ini akan mengacu kepada perancangan yang telah dibuat sebelumnya sesuai dengan ketentuan yang ada. Tahapan ini akan menerapkan implementasi dari metode *Learning Vector Quantization* (LVQ) terhadap permasalahan yang diangkat pada penelitian yakni penyakit mata.

Sistem yang akan dibangun menggunakan bahasa pemrograman Java dengan bantuan yang sudah dijelaskan pada sub bab analisis kebutuhan.

3.6 Pengujian

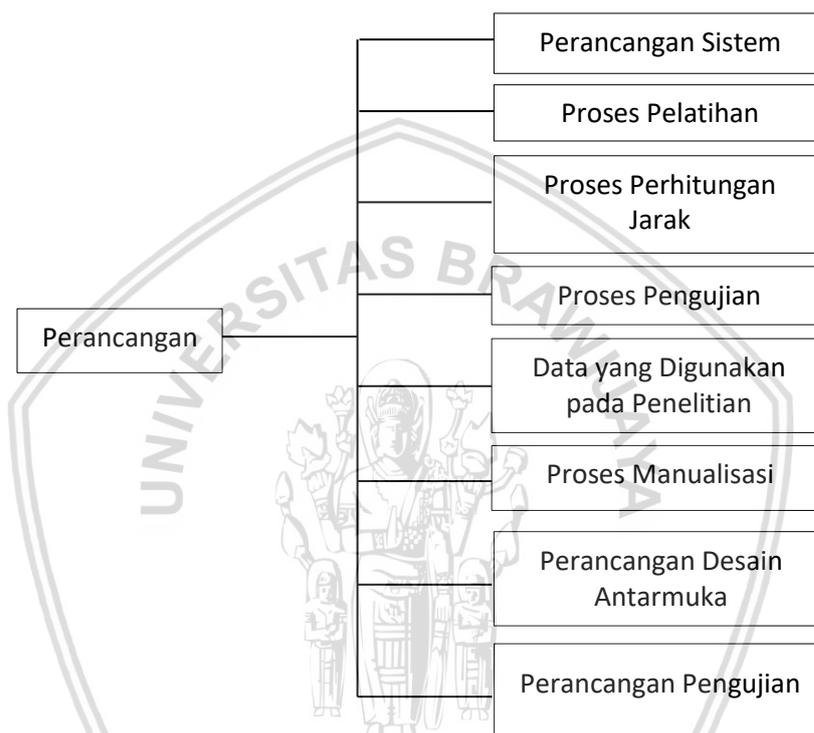
Pada tahapan selanjutnya yaitu pengujian akan dilakukan beberapa pengujian untuk menguji tingkat akurasi yang didapatkan oleh metode *Learning Vector Quantization* (LVQ) pada permasalahan penyakit mata. Pengujian tersebut meliputi beberapa pengujian yang dikaitkan dengan metode LVQ, yakni Pengujian *Learning Rate*, Pengujian Pengurangan *Learning Rate*, Pengujian Jumlah Data Latih yang dibandingkan dengan Jumlah Data Uji, Pengujian Maksimum Iterasi serta Pengujian Minimum Alpha. Kemudian ada juga pengujian validasi yang digunakan untuk mengetahui jalannya sistem yang dibangun sesuai dengan perancangan yang telah dibuat sebelumnya.

3.7 Kesimpulan dan saran

Pada tahap kesimpulan dan saran, peneliti akan menarik kesimpulan dari apa yang sudah dikerjakan pada penelitian terkait dengan identifikasi pada penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ). Kesimpulan didapatkan dari hasil pengujian serta analisis terhadap metode LVQ yang digunakan, selain itu juga peneliti akan memberikan saran kepada pembaca mengenai pertimbangan kedepannya dalam mengembangkan sistem selanjutnya terkait dengan permasalahan utama yang dibahas yakni penyakit mata dengan metode yang digunakan yaitu metode LVQ untuk memperbaiki kesalahan yang dilakukan pada penelitian yang dikerjakan serta kemudian mampu memperbaiki kesalahan yang dilakukan pada penulisan laporan penelitian ini.

BAB 4 PERANCANGAN

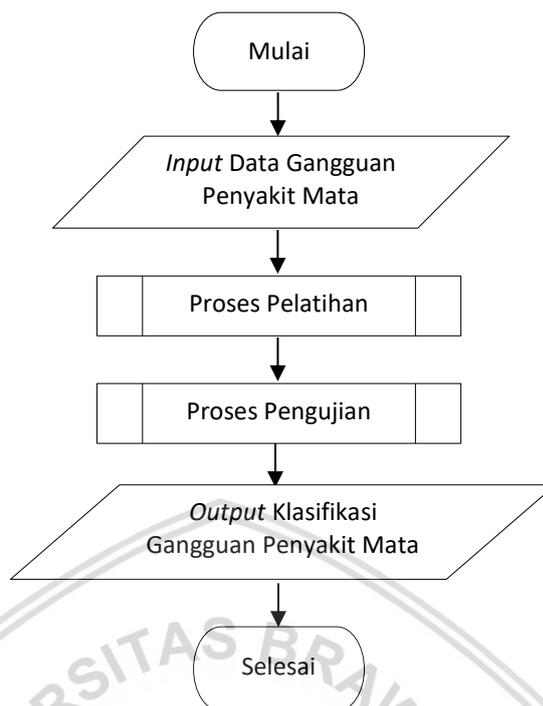
Pada bab ini akan dibahas mengenai proses perancangan yang digunakan untuk melakukan “Identifikasi Penyakit Mata Menggunakan Metode *Learning Vector Quantization* (LVQ)”. Pada bab perancangan ini, terdapat 6 tahapan yang dilakukan untuk membangun rancangan yang akan digunakan untuk mengimplementasikan aplikasi yang ditunjukkan pada Gambar 4.1.



Gambar 4.1 Perancangan Secara Umum

4.1 Perancangan sistem

Pada sub bab ini akan dijelaskan mengenai perancangan sistem menggunakan *flowchart* (diagram alir). Perancangan sistem secara umum terdiri atas beberapa tahapan yang menggambarkan alur dari cara kerja sistem. Tahapan dalam diagram alir meliputi *input* data gejala dari penyakit mata, kemudian masuk ke proses alur yang dikerjakan dalam tahapan tersebut adalah pelatihan dan pengujian yang dilakukan dengan metode *Learning Vector Quantization* (LVQ) dan tahapan terakhir dari algoritme tersebut adalah untuk mendapatkan *output* berupa hasil klasifikasi serta akurasi yang digunakan untuk mengidentifikasi penyakit mata. Tujuan dari adanya *flowchart* adalah untuk membantu dalam memahami alur proses yang terjadi didalam sistem yang akan dibangun. Perancangan sistem yang digambarkan dalam *flowchart* ditunjukkan pada Gambar 4.2.

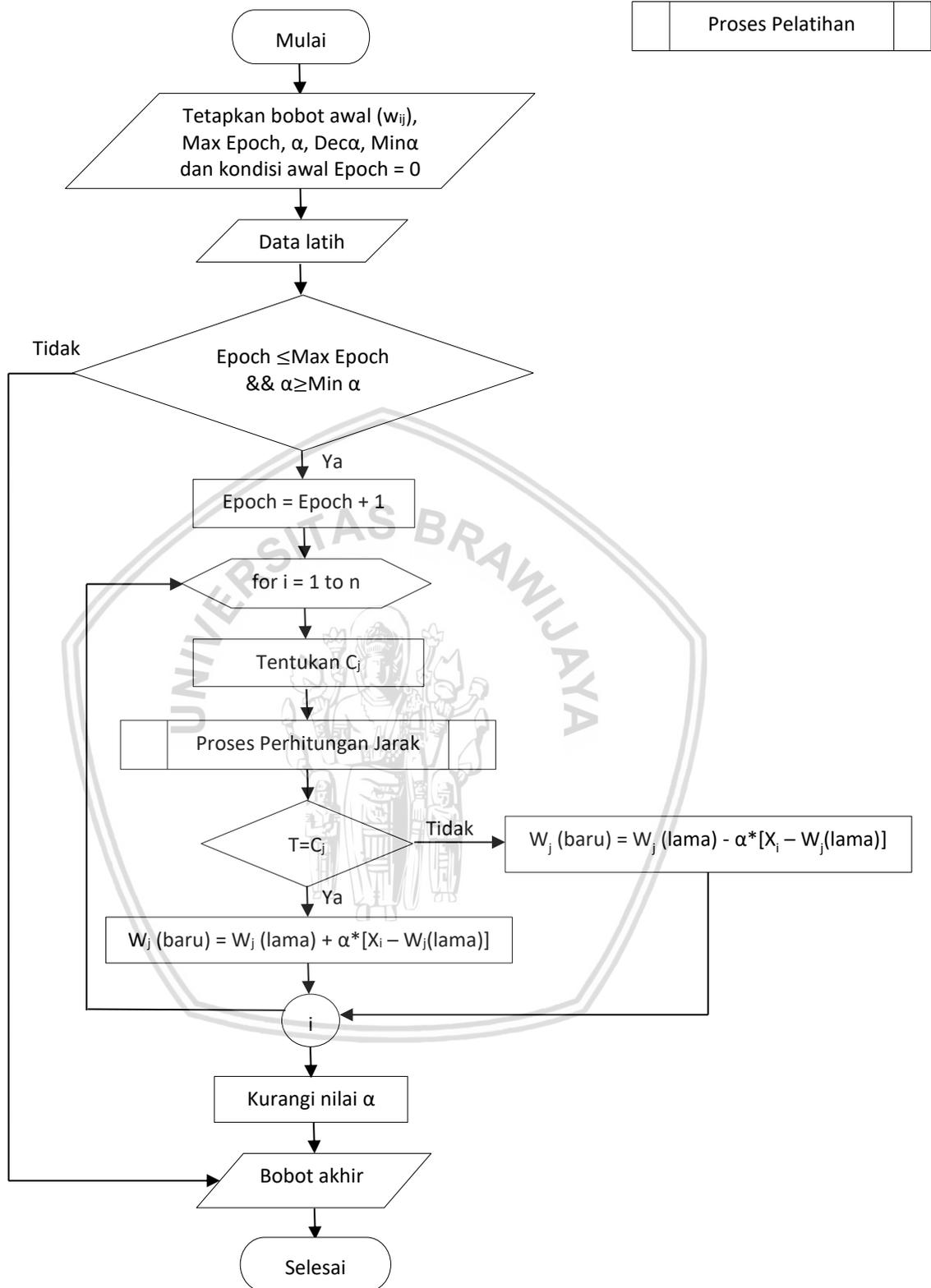


Gambar 4.2 Flowchart (Diagram Alir) Perancangan Sistem

Pada Gambar 4.2, terlihat alur dari diagram alir sistem yang akan dibangun. Dimulai dari *input* data gangguan penyakit mata dengan data yang akan diproses berupa data gejala penyakit mata. Setelah proses *input* berhasil dikerjakan, maka selanjutnya masuk kedalam proses pelatihan. Pada proses pelatihan, data gejala penyakit pada penyakit mata yang telah dimasukkan sebelumnya akan diproses. Proses pelatihan juga melibatkan proses perhitungan jarak dengan menggunakan *Euclidean Distance* dengan hasil dari proses ini akan mendapatkan bobot akhir yang akan digunakan pada proses pengujian. Didalam proses pengujian, *input* yang akan diproses adalah data uji serta bobot akhir yang didapatkan dari proses pelatihan guna mendapatkan hasil klasifikasi berupa penyakit mata dengan menggunakan metode *Learning Vector Quantization (LVQ)*.

4.2 Proses pelatihan

Pada sub bab ini akan diuraikan mengenai proses pelatihan pada perancangan sistem yang akan dibangun. Proses pelatihan dilakukan pertama kali yakni inialisasi awal untuk menetapkan bobot awal dan parameter-parameter LVQ yang mendukung. Kemudian data tersebut juga akan di proses dengan menggunakan jarak Euclidean untuk mendapatkan bobot dan bias akhir yang akan terus diproses dalam iterasi dengan hasil akhirnya berupa bobot akhir yang akan digunakan pada proses pengujian. Berikut ini adalah Gambar 4.3 yang menunjukkan diagram alir dari proses pelatihan LVQ pada identifikasi penyakit mata.



Gambar 4.3 Flowchart (Diagram Alir) Proses Pelatihan

Dari Gambar 4.3, tahapan diagram alir untuk proses pelatihan dengan menggunakan metode *Learning Vector Quantization* (LVQ) dapat dijelaskan sebagai berikut.

1. Menetapkan bobot awal variabel (w_{ij}) dengan $i = 1,2,\dots,k$ dan $j=1,2,\dots,m$; iterasi maksimum (Max Epoch), parameter *Learning Rate* (α), pengurangan *Learning Rate* (Dec α), minimal *Learning Rate* yang diperbolehkan (Min α) dan kondisi awal pada iterasi (Epoch) dengan Epoch = 0.
2. *Input* berupa data latih yang sudah didapatkan. Data latih terdiri atas beberapa data yang masuk dalam kategori gejala penyakit mata, jenis penyakit mata dan target yang berupa kelas untuk banyaknya data latih yang diperoleh.
3. Selanjutnya adalah menjalankan syarat yakni Epoch \leq Max Epoch dan $\alpha \geq$ Min α . Penentuan proses ini akan berakhir atau tidaknya tergantung kepada jumlah iterasi yang sudah dikerjakan dan α yang didapatkan dari hasil pengurangannya tidak melebihi minimal *Learning Rate* yang telah ditentukan.
 - a. Iterasi yang bertambah yakni Epoch = Epoch + 1 untuk menjalankan proses berikutnya.
 - b. Proses selanjutnya ialah melakukan perulangan dengan proses yang akan dikerjakan didalam perulangan tersebut adalah sebagai berikut.
 - 1) Menentukan C_j untuk mendapatkan nilai minimal pada proses perhitungan jarak yang digunakan agar dapat memengaruhi perubahan bobot pada proses perhitungan pelatihan data.
 - 2) Setelah selesai dengan tahapan proses perhitungan jarak, selanjutnya adalah untuk memperbaiki bobot lama untuk mendapatkan bobot yang baru. Bobot baru didapatkan dengan kondisi sebagai berikut.
 - a) Apabila $T=C_j$ maka proses yang dijalankan adalah:

$$W_j(\text{baru}) = W_j(\text{lama}) + \alpha (x_i - W_j(\text{lama}))$$
 - b) Apabila $T \neq C_j$ maka proses yang dijalankan adalah:

$$W_j(\text{baru}) = W_j(\text{lama}) - \alpha (x_i - W_j(\text{lama}))$$

Tahapan ini akan berakhir sampai dengan batas n yang telah ditentukan sebelumnya didalam proses perulangan.

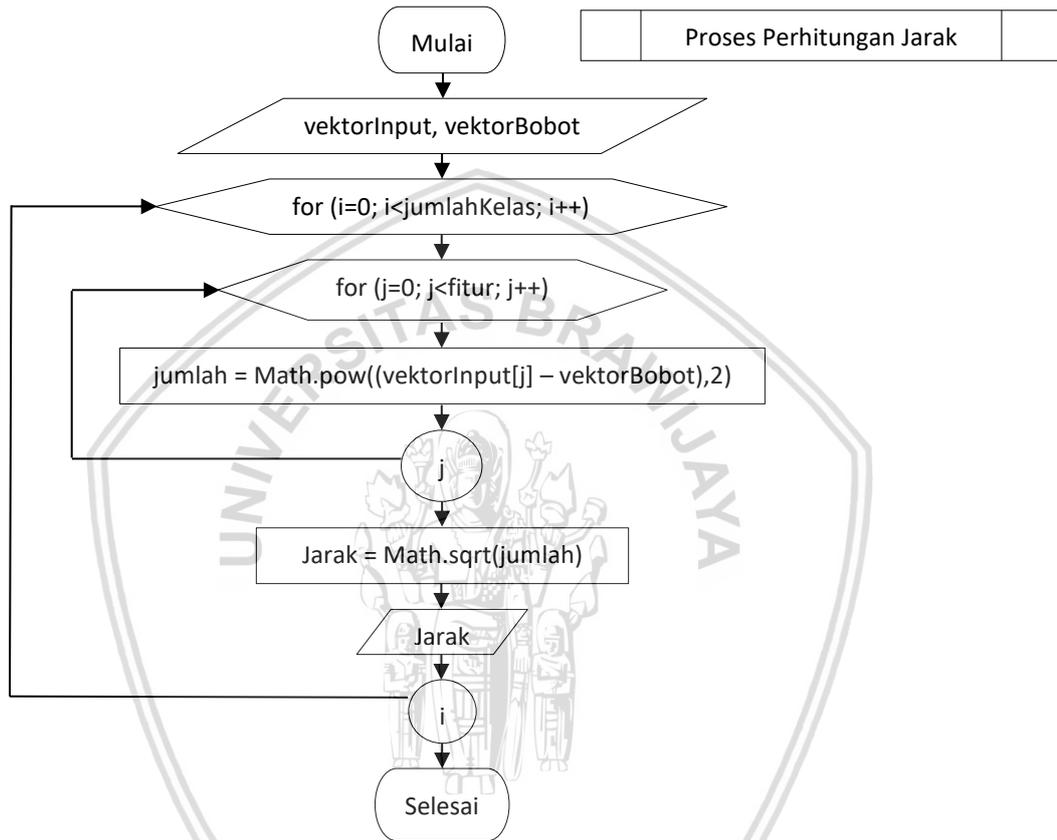
 - c. Selanjutnya adalah mengurangi nilai α . Pengurangan nilai α bisa dilakukan dengan dua cara yaitu:
 - 1) $\alpha = \alpha - \text{Dec } \alpha$
 - 2) $\alpha = \alpha - \alpha * \text{Dec } \alpha$
4. Hasil yang didapatkan dari proses pelatihan ini adalah bobot akhir yang akan digunakan pada proses pengujian.

4.3 Proses perhitungan jarak

Pada sub bab proses perhitungan jarak akan dilakukan dengan proses perhitungan dengan menggunakan *Euclidean distance*. Proses perhitungan jarak



pada metode *Learning Vector Quantization* (LVQ) dilakukan untuk mencari jarak minimum. Proses perhitungan jarak yang dilakukan dengan *Euclidean distance* memberikan pendekatan terhadap LVQ yang mengklasifikasi *output* yang diwakili oleh satu kelas atau kategori tertentu. Hal tersebut dilakukan karena metode LVQ memberlakukan vektor acuan terhadap vektor bobot yang akan dikelompokkan berdasarkan kedekatan antara jarak vektor *input* terhadap bobot yang diperoleh. Diagram alir dari proses perhitungan jarak ditunjukkan pada Gambar 4.4.



Gambar 4.4 Flowchart (Diagram Alir) Proses Perhitungan Jarak

Dari Gambar 4.4, tahapan diagram alir untuk proses perhitungan jarak dengan menggunakan *Euclidean distance* untuk mendapatkan pendekatan terhadap pengelompokan atas jarak kedekatan dari vektor *input* dan bobot adalah sebagai berikut.

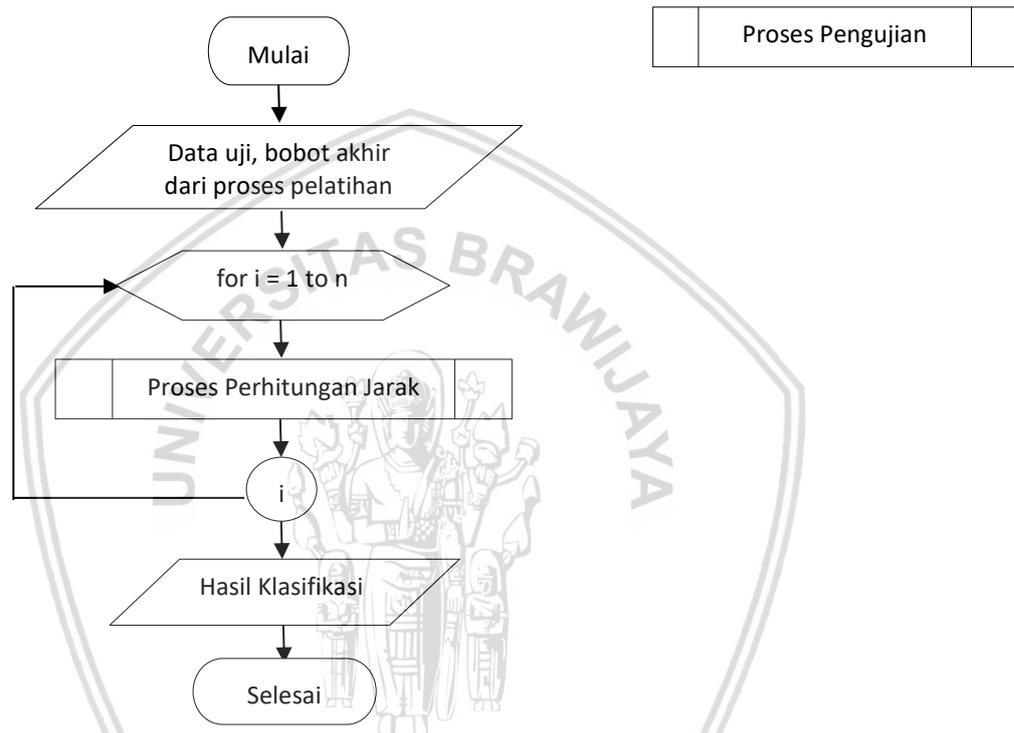
1. *Input* untuk perhitungan jarak adalah vektorInput dan vektorBobot. vektorInput merupakan inisialisasi masukan data, baik itu untuk data latih maupun data uji nantinya. Sedangkan vectorBobot adalah inisialisasi masukan data untuk nilai bobot.
2. Proses selanjutnya ialah melakukan perulangan dengan proses yang akan dikerjakan didalam perulangan tersebut adalah sebagai berikut.
 - a. Menghitung nilai Jarak untuk mendapatkan nilai jarak minimum/jarak terdekat dengan menggunakan rumus *Euclidean distance*.



- b. Hasil yang diperoleh adalah berupa nilai jarak terdekat/minimum. Nilai ini nantinya dapat digunakan baik didalam proses pelatihan maupun proses pengujian.

4.4 Proses pengujian

Pada sub bab ini akan diuraikan mengenai proses pengujian pada perancangan sistem yang akan dibangun. Proses pengujian dilakukan agar proses klasifikasi yang dilakukan terhadap penyakit mata dapat bekerja secara optimal. Tahapan dari proses pengujian ditunjukkan pada Gambar 4.5.



Gambar 4.5 Flowchart (Diagram Alir) Proses Pengujian

Dari Gambar 4.5, tahapan diagram alir untuk proses pengujian dengan menggunakan metode *Learning Vector Quantization* (LVQ) dapat dijelaskan sebagai berikut.

1. *Input* berupa data uji yang sudah didapatkan. Data uji terdiri atas beberapa data yang masuk dalam kategori gejala penyakit mata, jenis penyakit mata dan target yang berupa kelas untuk banyaknya data uji yang diperoleh. Data uji digunakan untuk memastikan metode *Learning Vector Quantization* (LVQ) berjalan dengan optimal dalam melakukan proses klasifikasi terhadap penyakit mata. Selain data uji, terdapat juga bobot akhir yang diambil dari proses pelatihan sebelumnya.
2. Selanjutnya adalah melakukan proses perulangan yaitu proses perhitungan jarak antara data uji dan bobot akhir hasil perhitungan dari proses pelatihan. Langkah-langkah yang dilakukan sama dengan proses yang ditunjukkan pada

Gambar 4.4. Tahapan ini akan berakhir sampai dengan batas n yang telah ditentukan.

3. Hasil akhir yang didapatkan dari proses pengujian ini adalah berupa klasifikasi kelas pada penyakit mata.

4.5 Data yang digunakan pada penelitian

Pada tahapan perancangan untuk mencapai perhitungan manualisasi, sebelumnya akan dijelaskan mengenai data yang akan digunakan pada penelitian ini. Proses ini dilakukan untuk mendapatkan data yang akan dikonversikan ke biner agar mempermudah perhitungan dengan metode *Learning Vector Quantization* (LVQ) pada penyakit mata. Berikut ini adalah penjelasan mengenai proses perubahan data menjadi angka biner.

4.5.1 Proses konversi biner

Pada tahapan ini, proses yang dilakukan adalah data yang telah didapatkan akan dikonversikan ke biner untuk memudahkan dalam perhitungan dengan menggunakan metode LVQ. Pada penelitian ini, penyakit mata yang akan digunakan sebanyak 9 penyakit yang menyesuaikan dengan jumlah dari data penyakit mata yang didapatkan dari sumber (Lestari et al., 2016). Berikut ini akan dijelaskan mengenai keterangan penyakit mata yang akan digunakan pada penelitian ini yakni:

- 1 = Glaukoma
- 2 = Katarak
- 3 = Konjungtivitis
- 4 = Keratitis
- 5 = Pterigium
- 6 = Uveitis
- 7 = Dry Eyes
- 8 = Bleparitis
- 9 = Kalazion

Selanjutnya adalah keterangan pada gejala penyakit mata yang berjumlah 21 gejala yang menyesuaikan dengan jumlah data yang diperoleh dari sumber. Berikut ini adalah keterangan gejala penyakit mata yaitu:

- x1 = Sakit/nyeri kepala hebat
- x2 = Penglihatan kabur perlahan
- x3 = Silau
- x4 = Merah
- x5 = Nyeri

- x6 = Perut mual
 x7 = Penglihatan berkabut (berasap)
 x8 = Lensa mata keruh
 x9 = Gatal
 x10 = Berair
 x11 = Belekan
 x12 = Kelopak bengkak
 x13 = Panas
 x14 = Mengganjal
 x15 = Lengket
 x16 = Merah jika terkena sinar matahari
 x17 = Tumbuh selaput pada mata
 x18 = Timbul bayangan
 x19 = Usia > 50 tahun
 x20 = Kelopak mata timbul benjolan
 x21 = Perih

Untuk data yang akan dikonversi tersebut adalah data gejala yang terkait dengan penyakit mata. Misalkan pada Tabel 4.1 terdapat contoh data latih yang akan digunakan untuk proses konversi data ke biner.

Tabel 4.1 Data Latih

No	Gejala Penyakit Mata	Penyakit Mata
1	x1, x2, x3, x4, x5, x6, x14, x19	Glaukoma
2	x1, x3, x5, x19	Glaukoma
3	x2, x5, x7, x8, x19	Katarak
4	x2, x7, x10, x18, x19	Katarak
5	x4, x10, x11, x15, x21	Konjungtivitis
6	x3, x4, x10, x11, x14, x15, x19	Konjungtivitis
7	x5, x9, x10, x14, x19	Keratitis
8	x4, x5, x10, x11, x14, x19	Keratitis
9	x2, x14, x16, x17, x19	Pterigium
10	x2, x17	Pterigium
11	x1, x2, x3, x5, x18	Uveitis

Tabel 4.1 Data Latih (Lanjutan)

No	Gejala Penyakit Mata	Penyakit Mata
12	x1, x2, x3, x4, x18	Uveitis
13	x9, x10, x19	Dry Eyes
14	x9, x10, x19, x21	Dry Eyes
15	x4, x9, x10, x11, x12, x19	Bleparitis
16	x5, x9, x10, x11, x12, x19	Bleparitis
17	x19, x20	Kalazion
18	x5, x20	Kalazion

Pada Tabel 4.1, dapat dilihat bahwa data yang akan digunakan untuk proses data latih kemudian akan dikonversi ke angka biner. Untuk 21 gejala yang ada, akan diberikan nilai 1 untuk gejala yang bernilai ya dan akan diberikan nilai 0 untuk gejala yang bernilai tidak. Sedangkan untuk target penyakit yang ada, akan diberikan nilai 1 untuk Glaukoma, nilai 2 untuk Katarak, nilai 3 untuk Konjungtivitis, nilai 4 untuk Keratitis, nilai 5 untuk Pterigium, nilai 6 untuk Uveitis, nilai 7 untuk Dry Eyes, nilai 8 untuk Bleparitis dan nilai 9 untuk Kalazion.

Misalkan untuk menjelaskan proses konversi biner ini akan diambil contoh untuk data ke-1, dengan data ke-1 yang terdiri atas gejala penyakit mata x1, x2, x3, x4, x5, x6, x14 dan x19 dengan penyakit mata yaitu Glaukoma. Maka hasil yang akan diperoleh dari konversi biner ini adalah $x_1 = 1$, $x_2 = 1$, $x_3 = 1$, $x_4 = 1$, $x_5 = 1$, $x_6 = 1$, $x_{14} = 1$ dan $x_{19} = 1$ sedangkan sisanya bernilai 0. Untuk penyakit mata yang diperoleh karena berupa Glaukoma maka nilai target yang akan diberikan adalah 1. Untuk proses konversi biner pada data latih yang selengkapnya bisa dilihat pada Tabel 4.2.

Tabel 4.2 Konversi Biner pada Data Latih

No.	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21	Target
1.	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
2.	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
3.	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	2
4.	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	2
5.	0	0	0	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	3
6.	0	0	1	1	0	0	0	0	0	1	1	0	0	1	1	0	0	0	1	0	0	3
7.	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	4
8.	0	0	0	1	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1	0	0	4
9.	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	5
10.	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	5
11.	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	6
12.	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	6
13.	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	7
14.	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	7
15.	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	8
16.	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	8
17.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	9
18.	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	9

4.6 Proses manualisasi

Pada tahapan perancangan untuk perhitungan manualisasi akan dijelaskan mengenai proses perhitungan manual untuk klasifikasi pada penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ). Proses perhitungan manual terdiri atas beberapa tahapan yakni proses pelatihan, proses pengujian dan proses perhitungan akurasi.

4.6.1 Proses pelatihan

Tahapan pada proses pelatihan adalah dengan menggunakan data yang telah diuraikan pada Tabel 4.2, data tersebut akan dilatih terlebih dahulu dengan $\alpha = 0,05$, Dec $\alpha = 0,1$ dan iterasi maksimal = 5. Berikut ini akan diuraikan tahapan proses pelatihan yang dilakukan untuk menghitung data latih dengan menggunakan metode LVQ.

1. Inisialisasi bobot awal variabel (w_{ij}). Banyaknya bobot awal sejumlah 9 data didapatkan dari jumlah kelas/target pada penyakit mata. Bobot awal dipilih secara acak dari banyaknya data akan diambil masing-masing sejumlah 9 sebagai acuan dari banyaknya kelas yang ada seperti ditunjukkan pada Tabel 4.3.

Tabel 4.3 Bobot Awal Variabel

Bobot Awal (w_{ij})																					
w	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
5	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
6	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
7	0	0	0	0	1	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0
8	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
9	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

2. Inisialisasi parameter yang mendukung dalam algoritma LVQ yakni iterasi maksimum (Max Epoch) = 5, *learning rate* (α) = 0,05, pengurangan *learning rate* (Dec α) = 0,1 dan minimal *learning rate* yang diperbolehkan (Min α) = 0,0000001.
3. *Input* data latih. Data latih diperoleh dari Tabel 4.2.
4. Menetapkan kondisi awal pada itarasi (Epoch) dengan Epoch = 0.
5. Kondisi selanjutnya adalah akan dikerjakan apabila memenuhi syarat Epoch \leq Max Epoch dan $\alpha \geq$ Min α .
6. Apabila syarat tersebut memenuhi, maka proses yang dikerjakan selanjutnya adalah untuk menghitung *Euclidean distance* agar mendapatkan jarak

minimum atau jarak terdekat dari jumlah data yang telah diinisialisasi pada Tabel 4.2 dengan menggunakan bobot yang ada pada Tabel 4.3. Berikut adalah hasil perhitungannya untuk iterasi pertama:

- Untuk data ke-1 ($x_{1,1} - x_{1,21}$) untuk jarak bobot ke-1 ($w_{1,1} - w_{1,21}$):

$$D1(x_{ij}-w_{ij})$$

$$= \sqrt{\begin{matrix} (1-0)^2 + (1-1)^2 + (1-1)^2 + (1-1)^2 + (1-1)^2 + (1-0)^2 + \\ (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + \\ (0-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (1-1)^2 + (0-0)^2 + (0-0)^2 \end{matrix}}$$

$$= 2$$

- Untuk data ke-1 ($x_{1,1} - x_{1,21}$) untuk jarak bobot ke-2 ($w_{2,1} - w_{2,21}$):

$$D2(x_{ij}-w_{ij})$$

$$= \sqrt{\begin{matrix} (1-0)^2 + (1-1)^2 + (1-1)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + \\ (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + \\ (0-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (1-1)^2 + (0-0)^2 + (0-0)^2 \end{matrix}}$$

$$= 2,449489743$$

- Untuk data ke-1 ($x_{1,1} - x_{1,21}$) untuk jarak bobot ke-3 ($w_{3,1} - w_{3,21}$):

$$D3(x_{ij}-w_{ij})$$

$$= \sqrt{\begin{matrix} (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-1)^2 + (1-0)^2 + (1-0)^2 + \\ (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 + (0-0)^2 + \\ (0-0)^2 + (1-0)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (1-0)^2 + (0-0)^2 + (0-0)^2 \end{matrix}}$$

$$= 3,31662479$$

- Untuk data ke-1 ($x_{1,1} - x_{1,21}$) untuk jarak bobot ke-4 ($w_{4,1} - w_{4,21}$):

$$D4(x_{ij}-w_{ij})$$

$$= \sqrt{\begin{matrix} (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-1)^2 + (1-0)^2 + \\ (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + \\ (0-0)^2 + (1-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (1-0)^2 + (0-0)^2 + (0-0)^2 \end{matrix}}$$

$$= 2,828427125$$

- Untuk data ke-1 ($x_{1,1} - x_{1,21}$) untuk jarak bobot ke-5 ($w_{5,1} - w_{5,21}$):

$$D5(x_{ij}-w_{ij})$$



$$= \sqrt{\begin{matrix} (1-1)^2 + (1-1)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + \\ (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + \\ (0-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 \\ + (1-1)^2 + (0-0)^2 + (0-0)^2 \end{matrix}}$$

= 2,645751311

- Untuk data ke-1 ($x_{1,1} - x_{1,21}$) untuk jarak bobot ke-6 ($w_{6,1} - w_{6,21}$):

D6($x_{ij} - w_{ij}$)

$$= \sqrt{\begin{matrix} (1-0)^2 + (1-1)^2 + (1-0)^2 + (1-1)^2 + (1-1)^2 + (1-0)^2 + \\ (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + \\ (0-0)^2 + (1-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 \\ + (1-1)^2 + (0-0)^2 + (0-0)^2 \end{matrix}}$$

= 2

- Untuk data ke-1 ($x_{1,1} - x_{1,21}$) untuk jarak bobot ke-7 ($w_{7,1} - w_{7,21}$):

D7($x_{ij} - w_{ij}$)

$$= \sqrt{\begin{matrix} (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-1)^2 + (1-0)^2 + \\ (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + \\ (0-1)^2 + (1-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (1-1)^2 + (0-0)^2 + (0-1)^2 \end{matrix}}$$

= 3

- Untuk data ke-1 ($x_{1,1} - x_{1,21}$) untuk jarak bobot ke-8 ($w_{8,1} - w_{8,21}$):

D8($x_{ij} - w_{ij}$)

$$= \sqrt{\begin{matrix} (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-1)^2 + (1-0)^2 + (1-0)^2 + \\ (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 + \\ (0-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (1-1)^2 + (0-0)^2 + (0-0)^2 \end{matrix}}$$

= 3

- Untuk data ke-1 ($x_{1,1} - x_{1,21}$) untuk jarak bobot ke-9 ($w_{9,1} - w_{9,21}$):

D9($x_{ij} - w_{ij}$)

$$= \sqrt{\begin{matrix} (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-1)^2 + (1-0)^2 + \\ (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + \\ (0-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ + (1-1)^2 + (0-1)^2 + (0-0)^2 \end{matrix}}$$

= 2,645751311



Setelah dilakukan perhitungan untuk data ke-1 pada keseluruhan bobot mulai dari w_1 sampai dengan w_9 didapatkan jarak terkecil dengan menggunakan jarak Euclidean adalah pada bobot w_1 dan bobot w_6 . Jika terdapat 2 bobot atau lebih yang menghasilkan nilai bobot yang sama, maka diambil salah satu bobot yang akan di *update*. Misalkan, pada kasus ini bobot yang terpilih adalah bobot w_1 . Berikut ini adalah Tabel 4.4 untuk menunjukkan keseluruhan data latih sejumlah 18 data pada iterasi 1 dalam proses perhitungan jarak minimum atau jarak terdekat dengan menggunakan jarak *Euclidean*.

Tabel 4.4 Hasil Proses Perhitungan Jarak Terdekat dengan Menggunakan *Euclidean Distance* untuk Iterasi Pertama

No	D1	D2	D3	D4	D5	D6	D7	D8	D9	Min	Cj
1	2	2,449	3,317	2,828	2,646	2	3	3	2,646	2	1
2	1,952	2	3	2,449	2,236	2,449	2,646	2,646	1,732	1,732	9
3	1,735	1,732	3,162	2,646	2,449	2,236	2,828	2,828	2,027	1,732	2
4	2,238	1,705	2,828	2,646	2	2,236	2,828	2,828	2,471	1,705	2
5	2,985	2,954	1,414	2,646	2,828	3	2,449	2,449	2,847	1,414	3
6	2,610	2,630	1,976	2,646	2,828	2,646	2,449	2,449	2,865	1,976	3
7	2,610	2,575	2,412	1	2,449	2,236	2	2,449	2,027	1	4
8	2,410	2,762	2,150	1,975	2,646	2	2,236	2,236	2,260	1,975	4
9	2,610	2,196	3,102	2,593	2	2,236	2,828	2,828	2,471	2	5
10	2,431	1,956	2,612	2,432	1,676	2,449	3	2,646	2,260	1,676	5
11	2,193	2,196	3,118	2,630	2,394	2,236	3,162	3,162	2,511	2,193	1
12	2,281	2,218	2,779	2,969	2,394	2,236	3,464	2,828	2,882	2,218	2
13	2,666	2,195	1,980	1,680	1,957	2,646	2,449	2	2,027	1,680	4
14	2,847	2,412	2,196	2,028	2,198	2,828	2,236	2,236	2,260	2,028	4
15	2,830	2,814	1,709	2,492	2,614	2,828	3	1	2,666	1	8
16	2,847	2,778	2,218	2,029	2,614	2,828	2,646	1,704	2,260	1,704	8
17	2,471	1,981	2,593	2,514	2,153	2,449	2,646	2,217	1,004	1,004	9
18	2,492	2,434	2,612	2,032	2,557	2,449	2,646	2,610	1,005	1,005	9

- Setelah berhasil melakukan perhitungan nilai jarak dan mendapatkan nilai dari jarak terdekat, maka selanjutnya adalah menentukan apakah syarat untuk melakukan *update* bobot bisa dilakukan dengan melihat target (T) dan C_j . Apabila $T=C_j$ maka *update* bobot dilakukan dengan menggunakan rumus $w_{j(\text{baru})} = w_{j(\text{lama})} + \alpha (x_i - w_{j(\text{lama})})$ sedangkan $T \neq C_j$ maka *update* bobot dilakukan dengan menggunakan rumus $w_{j(\text{baru})} = w_{j(\text{lama})} - \alpha (x_i - w_{j(\text{lama})})$. C_j didapatkan dari nilai minimal yang sudah dihitung pada proses perhitungan jarak dengan menggunakan *Euclidean distance*.

Pada proses manualisasi sebelumnya, diketahui bahwa nilai minimum yang didapatkan adalah pada data 1, sehingga bobot yang akan diperbarui adalah bobot pada w_1 . Kemudian, diketahui juga bahwa $T=1$ dan $C_j=1$ maka masuk dalam syarat $T = C_j$ sehingga hasil yang diperoleh untuk *update* bobot w_1 adalah:



- Untuk $w_{1,1} = w_{1,1} + \alpha (x_{1,1} - w_{1,1}) = 0 + 0,05*(1-0) = 0,05$
- Untuk $w_{1,2} = w_{1,2} + \alpha (x_{1,2} - w_{1,2}) = 1 + 0,05*(1-1) = 1$
- Untuk $w_{1,3} = w_{1,3} + \alpha (x_{1,3} - w_{1,3}) = 1 + 0,05*(1-1) = 1$
- Untuk $w_{1,4} = w_{1,4} + \alpha (x_{1,4} - w_{1,4}) = 1 + 0,05*(1-1) = 1$
- Untuk $w_{1,5} = w_{1,5} + \alpha (x_{1,5} - w_{1,5}) = 1 + 0,05*(1-1) = 1$
- Untuk $w_{1,6} = w_{1,6} + \alpha (x_{1,6} - w_{1,6}) = 0 + 0,05*(1-0) = 0,05$
- Untuk $w_{1,7} = w_{1,7} + \alpha (x_{1,7} - w_{1,7}) = 1 + 0,05*(0-1) = 0,95$
- Untuk $w_{1,8} = w_{1,8} + \alpha (x_{1,8} - w_{1,8}) = 0 + 0,05*(0-0) = 0$
- Untuk $w_{1,9} = w_{1,9} + \alpha (x_{1,9} - w_{1,9}) = 0 + 0,05*(0-0) = 0$
- Untuk $w_{1,10} = w_{1,10} + \alpha (x_{1,10} - w_{1,10}) = 0 + 0,05*(0-0) = 0$
- Untuk $w_{1,11} = w_{1,11} + \alpha (x_{1,11} - w_{1,11}) = 0 + 0,05*(0-0) = 0$
- Untuk $w_{1,12} = w_{1,12} + \alpha (x_{1,12} - w_{1,12}) = 0 + 0,05*(0-0) = 0$
- Untuk $w_{1,13} = w_{1,13} + \alpha (x_{1,13} - w_{1,13}) = 0 + 0,05*(0-0) = 0$
- Untuk $w_{1,14} = w_{1,14} + \alpha (x_{1,14} - w_{1,14}) = 0 + 0,05*(1-0) = 0,05$
- Untuk $w_{1,15} = w_{1,15} + \alpha (x_{1,15} - w_{1,15}) = 0 + 0,05*(0-0) = 0$
- Untuk $w_{1,16} = w_{1,16} + \alpha (x_{1,16} - w_{1,16}) = 0 + 0,05*(0-0) = 0$
- Untuk $w_{1,17} = w_{1,17} + \alpha (x_{1,17} - w_{1,17}) = 0 + 0,05*(0-0) = 0$
- Untuk $w_{1,18} = w_{1,18} + \alpha (x_{1,18} - w_{1,18}) = 0 + 0,05*(0-0) = 0$
- Untuk $w_{1,19} = w_{1,19} + \alpha (x_{1,19} - w_{1,19}) = 1 + 0,05*(1-1) = 1$
- Untuk $w_{1,20} = w_{1,20} + \alpha (x_{1,20} - w_{1,20}) = 0 + 0,05*(0-0) = 0$
- Untuk $w_{1,21} = w_{1,21} + \alpha (x_{1,21} - w_{1,21}) = 0 + 0,05*(0-0) = 0$

Pada Tabel 4.5 – Tabel 4.13 menampilkan perubahan bobot dari w_1 sampai dengan w_9 dalam iterasi 1 untuk keseluruhan data.

Tabel 4.5 Perubahan Bobot w_1 pada Iterasi 1

No	w1																				
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21
	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0.05	1	1	1	1	0.05	0.95	0	0	0	0	0	0	0.05	0	0	0	0	1	0	0
2	0.05	1	1	1	1	0.05	0.95	0	0	0	0	0	0	0.05	0	0	0	0	1	0	0
3	0.05	1	1	1	1	0.05	0.95	0	0	0	0	0	0	0.05	0	0	0	0	1	0	0
4	0.05	1	1	1	1	0.05	0.95	0	0	0	0	0	0	0.05	0	0	0	0	1	0	0
5	0.05	1	1	1	1	0.05	0.95	0	0	0	0	0	0	0.05	0	0	0	0	1	0	0
6	0.05	1	1	1	1	0.05	0.95	0	0	0	0	0	0	0.05	0	0	0	0	1	0	0
7	0.05	1	1	1	1	0.05	0.95	0	0	0	0	0	0	0.05	0	0	0	0	1	0	0
8	0.05	1	1	1	1	0.05	0.95	0	0	0	0	0	0	0.05	0	0	0	0	1	0	0
9	0.05	1	1	1	1	0.05	0.95	0	0	0	0	0	0	0.05	0	0	0	0	1	0	0
10	0.05	1	1	1	1	0.05	0.95	0	0	0	0	0	0	0.05	0	0	0	0	1	0	0
11	0.0025	1	1	1.05	1	0.0525	0.9975	0	0	0	0	0	0	0.0525	0	0	0	-0.05	1.05	0	0
12	0.0025	1	1	1.05	1	0.0525	0.9975	0	0	0	0	0	0	0.0525	0	0	0	-0.05	1.05	0	0
13	0.0025	1	1	1.05	1	0.0525	0.9975	0	0	0	0	0	0	0.0525	0	0	0	-0.05	1.05	0	0
14	0.0025	1	1	1.05	1	0.0525	0.9975	0	0	0	0	0	0	0.0525	0	0	0	-0.05	1.05	0	0
15	0.0025	1	1	1.05	1	0.0525	0.9975	0	0	0	0	0	0	0.0525	0	0	0	-0.05	1.05	0	0
16	0.0025	1	1	1.05	1	0.0525	0.9975	0	0	0	0	0	0	0.0525	0	0	0	-0.05	1.05	0	0
17	0.0025	1	1	1.05	1	0.0525	0.9975	0	0	0	0	0	0	0.0525	0	0	0	-0.05	1.05	0	0
18	0.0025	1	1	1.05	1	0.0525	0.9975	0	0	0	0	0	0	0.0525	0	0	0	-0.05	1.05	0	0

Tabel 4.6 Perubahan Bobot w_2 pada Iterasi 1

No	w2																				
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21
	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3	0	1	0.95	0	0.05	0	1	0.05	0	0	0	0	0	0	0	0	0	0	1	0	0
4	0	1	0.9025	0	0.0475	0	1	0.0475	0	0.05	0	0	0	0	0	0	0	0.05	1	0	0
5	0	1	0.9025	0	0.0475	0	1	0.0475	0	0.05	0	0	0	0	0	0	0	0.05	1	0	0
6	0	1	0.9025	0	0.0475	0	1	0.0475	0	0.05	0	0	0	0	0	0	0	0.05	1	0	0
7	0	1	0.9025	0	0.0475	0	1	0.0475	0	0.05	0	0	0	0	0	0	0	0.05	1	0	0
8	0	1	0.9025	0	0.0475	0	1	0.0475	0	0.05	0	0	0	0	0	0	0	0.05	1	0	0
9	0	1	0.9025	0	0.0475	0	1	0.0475	0	0.05	0	0	0	0	0	0	0	0.05	1	0	0
10	0	1	0.9025	0	0.0475	0	1	0.0475	0	0.05	0	0	0	0	0	0	0	0.05	1	0	0
11	0	1	0.9025	0	0.0475	0	1	0.0475	0	0.05	0	0	0	0	0	0	0	0.05	1	0	0
12	-0.05	1	0.8976	-0.05	0.0499	0	1.05	0.0499	0	0.0525	0	0	0	0	0	0	0	0.0025	1.05	0	0
13	-0.05	1	0.8976	-0.05	0.0499	0	1.05	0.0499	0	0.0525	0	0	0	0	0	0	0	0.0025	1.05	0	0
14	-0.05	1	0.8976	-0.05	0.0499	0	1.05	0.0499	0	0.0525	0	0	0	0	0	0	0	0.0025	1.05	0	0
15	-0.05	1	0.8976	-0.05	0.0499	0	1.05	0.0499	0	0.0525	0	0	0	0	0	0	0	0.0025	1.05	0	0
16	-0.05	1	0.8976	-0.05	0.0499	0	1.05	0.0499	0	0.0525	0	0	0	0	0	0	0	0.0025	1.05	0	0
17	-0.05	1	0.8976	-0.05	0.0499	0	1.05	0.0499	0	0.0525	0	0	0	0	0	0	0	0.0025	1.05	0	0
18	-0.05	1	0.8976	-0.05	0.0499	0	1.05	0.0499	0	0.0525	0	0	0	0	0	0	0	0.0025	1.05	0	0

Tabel 4.7 Perubahan Bobot w_3 pada Iterasi 1

No	w3																				
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21
	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0.95	1	1	0	0	0	1	0	0	0	0	0	0.05
6	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
7	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
8	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
9	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
10	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
11	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
12	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
13	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
14	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
15	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
16	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
17	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475
18	0	0	0.05	1	0	0	0	0	0.9025	1	1	0	0	0.05	1	0	0	0	0.05	0	0.0475

Tabel 4.8 Perubahan Bobot w_4 pada Iterasi 1

No	w_4																				
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21
	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
6	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	0.05	0	0
8	0	0	0	0.05	1	0	0	0	0.95	1	0.05	0	0	1	0	0	0	0	0.0975	0	0
9	0	0	0	0.05	1	0	0	0	0.95	1	0.05	0	0	1	0	0	0	0	0.0975	0	0
10	0	0	0	0.05	1	0	0	0	0.95	1	0.05	0	0	1	0	0	0	0	0.0975	0	0
11	0	0	0	0.05	1	0	0	0	0.95	1	0.05	0	0	1	0	0	0	0	0.0975	0	0
12	0	0	0	0.05	1	0	0	0	0.95	1	0.05	0	0	1	0	0	0	0	0.0975	0	0
13	0	0	0	0.0525	1.05	0	0	0	0.9475	1	0.0525	0	0	1.05	0	0	0	0	0.0524	0	0
14	0	0	0	0.0551	1.1025	0	0	0	0.9449	1	0.0551	0	0	1.1025	0	0	0	0	0.0050	0	-0.05
15	0	0	0	0.0551	1.1025	0	0	0	0.9449	1	0.0551	0	0	1.1025	0	0	0	0	0.0050	0	-0.05
16	0	0	0	0.0551	1.1025	0	0	0	0.9449	1	0.0551	0	0	1.1025	0	0	0	0	0.0050	0	-0.05
17	0	0	0	0.0551	1.1025	0	0	0	0.9449	1	0.0551	0	0	1.1025	0	0	0	0	0.0050	0	-0.05
18	0	0	0	0.0551	1.1025	0	0	0	0.9449	1	0.0551	0	0	1.1025	0	0	0	0	0.0050	0	-0.05

Tabel 4.9 Perubahan Bobot w_5 pada Iterasi 1

No	w5																				
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21
	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
2	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
3	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
4	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
5	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
7	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
8	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0
9	0.95	1	0	0	0	0	0	0	0	0.95	0	0	0	0.05	0	0.05	1	0	1	0	0
10	0.9025	1	0	0	0	0	0	0	0	0.9025	0	0	0	0.0475	0	0.0475	1	0	0.95	0	0
11	0.9025	1	0	0	0	0	0	0	0	0.9025	0	0	0	0.0475	0	0.0475	1	0	0.95	0	0
12	0.9025	1	0	0	0	0	0	0	0	0.9025	0	0	0	0.0475	0	0.0475	1	0	0.95	0	0
13	0.9025	1	0	0	0	0	0	0	0	0.9025	0	0	0	0.0475	0	0.0475	1	0	0.95	0	0
14	0.9025	1	0	0	0	0	0	0	0	0.9025	0	0	0	0.0475	0	0.0475	1	0	0.95	0	0
15	0.9025	1	0	0	0	0	0	0	0	0.9025	0	0	0	0.0475	0	0.0475	1	0	0.95	0	0
16	0.9025	1	0	0	0	0	0	0	0	0.9025	0	0	0	0.0475	0	0.0475	1	0	0.95	0	0
17	0.9025	1	0	0	0	0	0	0	0	0.9025	0	0	0	0.0475	0	0.0475	1	0	0.95	0	0
18	0.9025	1	0	0	0	0	0	0	0	0.9025	0	0	0	0.0475	0	0.0475	1	0	0.95	0	0

Tabel 4.10 Perubahan Bobot w_6 pada Iterasi 1

No	w_6																				
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21
	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
1	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
2	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
3	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
4	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
5	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
6	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
7	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
8	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
9	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
10	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
11	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
12	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
13	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
14	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
15	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
16	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
17	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
18	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0

Tabel 4.11 Perubahan Bobot w_7 pada Iterasi 1

No	w7																				
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21
	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
1	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
2	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
3	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
4	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
5	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
6	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
7	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
8	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
9	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
10	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
11	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
12	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
13	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
14	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
15	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
16	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
17	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
18	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1

Tabel 4.12 Perubahan Bobot w_8 pada Iterasi 1

No	w8																				
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21
	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
2	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
3	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
4	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
5	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
6	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
7	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
8	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
9	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
10	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
11	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
12	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
13	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
14	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0
15	0	0	0	1	0	0	0	0	1	0.05	1	1	0	0	0	0	0	0	1	0	0
16	0	0	0	0.95	0.05	0	0	0	1	0.0975	1	1	0	0	0	0	0	0	1	0	0
17	0	0	0	0.95	0.05	0	0	0	1	0.0975	1	1	0	0	0	0	0	0	1	0	0
18	0	0	0	0.95	0.05	0	0	0	1	0.0975	1	1	0	0	0	0	0	0	1	0	0

Tabel 4.13 Perubahan Bobot w_9 pada Iterasi 1

No	w9																				
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21
	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
2	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
3	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
4	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
5	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
6	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
7	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
8	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
9	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
10	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
11	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
12	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
13	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
14	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
15	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
16	-0.05	0	-0.05	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.05	0
17	-0.0475	0	-0.0475	0	0.95	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.0475	0
18	-0.0451	0	-0.0451	0	0.9525	0	0	0	0	0	0	0	0	0	0	0	0	0	0.95	1.0451	0

8. Setelah proses perhitungan untuk mendapatkan bobot pada keseluruhan data, mulai dari data ke-1 sampai dengan data ke-18 (iterasi 1 telah selesai), maka akan dilakukan pengurangan terhadap nilai *Learning Rate* (α). Pengurangan pada nilai *Learning Rate* (α) dapat menggunakan rumus:

- $\alpha = \alpha - \text{Dec } \alpha$
- $\alpha = \alpha - \alpha * \text{Dec } \alpha$

Misalkan pada perhitungan ini, di gunakan rumus yang kedua yaitu

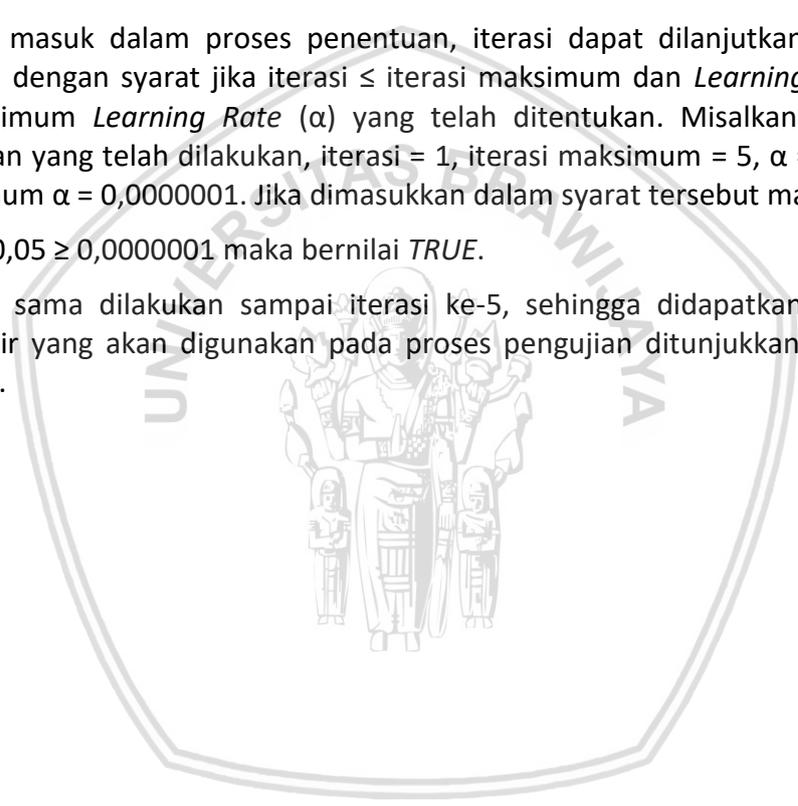
$\alpha = \alpha - \alpha * \text{Dec } \alpha$, maka untuk nilai $\alpha = 0,05$ dan $\text{Dec } \alpha = 0,1$ didapatkan hasil untuk nilai α yang baru adalah

$$\alpha = 0,05 - 0,05 * 0,1 = 0,045$$

9. Kemudian masuk dalam proses penentuan, iterasi dapat dilanjutkan atau dihentikan dengan syarat jika iterasi \leq iterasi maksimum dan *Learning Rate* (α) \geq Minimum *Learning Rate* (α) yang telah ditentukan. Misalkan pada perhitungan yang telah dilakukan, iterasi = 1, iterasi maksimum = 5, $\alpha = 0,05$ dan minimum $\alpha = 0,0000001$. Jika dimasukkan dalam syarat tersebut maka:

$1 \leq 5$ dan $0,05 \geq 0,0000001$ maka bernilai *TRUE*.

10. Cara yang sama dilakukan sampai iterasi ke-5, sehingga didapatkan hasil bobot akhir yang akan digunakan pada proses pengujian ditunjukkan pada Tabel 4.14.



Tabel 4.14 Hasil Bobot Akhir yang Didapatkan pada Iterasi Kelima

w	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21
w1	0.206	0.933	1	0.973	1	0.179	0.794	0	0	0	0	0	0	0.179	0	0	0	- 0.040	1.040	0	0
w2	- 0.036	1	0.655	- 0.036	0.169	0	1.036	0.169	0	0.176	0	0	0	0	0	0	0	0.140	1.036	0	0
w3	0	0	0.179	1.035	0	0	0	0	0.650	1	1.035	0	0	0.179	1.035	0	0	0	0.144	0	0.136
w4	0	0	0	0.191	1.195	0	0	0	0.809	1	0.191	0	0	1.195	0	0	0	0	0.179	0	- 0.079
w5	0.680	1.033	0	0	0	0	0	0	- 0.033	0.647	0	0	0	0.173	0	0.173	1.033	0	0.820	0	0
w6	0.271	1	0.271	0.867	0.862	0	0	0	0	0	0	0	0	0.729	0	0	0	1	0.729	0	0
w7	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1
w8	0	0	0	0.912	0.186	0	0	0	1	0.267	1.098	1.098	0	0	0	0	0	0	1	0	- 0.063
w9	- 0.102	0	- 0.102	0	0.828	0	0	0	0	0	0	0	0	0	0	0	0	0	0.821	1.102	0

4.6.2 Proses pengujian

Tahapan pada proses pengujian dilakukan dengan menggunakan data uji sebanyak 5 data uji. Berikut adalah Tabel 4.15 yang memperlihatkan data uji yang akan diproses pada tahap pengujian.

Tabel 4.15 Data Uji

No	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	x21	Target
1	0	0	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	9
3	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	7
4	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	5
5	0	0	0	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	8

Berikut ini akan diuraikan tahapan proses pengujian yang dilakukan untuk menghitung data uji dengan menggunakan metode LVQ.

1. Proses pengujian dimulai dengan inisialisasi data uji (ditunjukkan pada Tabel 4.15) dan bobot akhir (ditunjukkan pada Tabel 4.14) yang didapatkan dari proses pelatihan.
2. Selanjutnya dilakukan proses perhitungan jarak terdekat dengan menggunakan rumus *Euclidean distance* untuk mendapatkan hasil yang minimal. Berikut ini adalah hasil perhitungan dengan menggunakan rumus tersebut.

- Untuk data ke-1 ($x_{1,1}$ - $x_{1,21}$) untuk jarak bobot ke-1 ($w_{1,1}$ - $w_{1,21}$):

$$D1(x_{ij}-w_{ij})$$

$$= \sqrt{(0 - 0.206151092)^2 + (0 - 0.933097258)^2 + (0 - 1)^2 + (1 - 0.972889183)^2 + (0 - 1)^2 + (0 - 0.179040275)^2 + (0 - 0.793848908)^2 + (0 - 0)^2 + (1 - 0)^2 + (1 - 0)^2 + (1 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0.179040275)^2 + (1 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - (-0.039791925))^2 + (0 - 1.039791925)^2 + (0 - 0)^2 + (0 - 0)^2}$$

$$= 2,948043646$$

- Untuk data ke-1 ($x_{1,1}$ - $x_{1,21}$) untuk jarak bobot ke-2 ($w_{2,1}$ - $w_{2,21}$):

$$D2(x_{ij}-w_{ij})$$

$$= \sqrt{(0 - (-0.036462236))^2 + (0 - 1)^2 + (0 - 0.654588299)^2 + (1 - (-0.036462236))^2 + (0 - 0.169067328)^2 + (0 - 0)^2 + (0 - 1.036462236)^2 + (0 - 0.169067328)^2 + (1 - 0)^2 + (1 - 0.176344373)^2 + (1 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0.139882136)^2 + (0 - 1.036462236)^2 + (0 - 0)^2 + (0 - 0)^2}$$



$$= 2,899606933$$

- Untuk data ke-1 ($x_{1,1}$ - $x_{1,21}$) untuk jarak bobot ke-3 ($w_{3,1}$ - $w_{3,21}$):

$$D3(x_{ij}-w_{ij})$$

$$= \sqrt{(0-0)^2 + (0-0)^2 + (0-0.178879811)^2 + (1-1.035174799)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (1-0.649623375)^2 + (1-1)^2 + (1-1.035174799)^2 + (0-0)^2 + (0-0)^2 + (0-0.178879811)^2 + (1-1.035174799)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0.143705012)^2 + (0-0)^2 + (0-0.136322016)^2}$$

$$= 0,479276929$$

- Untuk data ke-1 ($x_{1,1}$ - $x_{1,21}$) untuk jarak bobot ke-4 ($w_{4,1}$ - $w_{4,21}$):

$$D4(x_{ij}-w_{ij})$$

$$= \sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (1-0.190826526)^2 + (0-1.194963452)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (1-0.809173474)^2 + (1-1)^2 + (1-0.190826526)^2 + (0-0)^2 + (0-0)^2 + (0-1.194963452)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0.178741186)^2 + (0-0)^2 + (0-(-0.078562037))^2}$$

$$= 2,289090189$$

- Untuk data ke-1 ($x_{1,1}$ - $x_{1,21}$) untuk jarak bobot ke-5 ($w_{5,1}$ - $w_{5,21}$):

$$D5(x_{ij}-w_{ij})$$

$$= \sqrt{(0-0.679733376)^2 + (0-1.032805)^2 + (0-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (1-(-0.032805))^2 + (1-0.64692876)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0.172824809)^2 + (1-0)^2 + (0-0.172824809)^2 + (0-1.032805)^2 + (0-0)^2 + (0-0.819753569)^2 + (0-0)^2 + (0-0)^2}$$

$$= 2,741986239$$

- Untuk data ke-1 ($x_{1,1}$ - $x_{1,21}$) untuk jarak bobot ke-6 ($w_{6,1}$ - $w_{6,21}$):

$$D6(x_{ij}-w_{ij})$$



$$= \sqrt{(0 - 0.270755272)^2 + (0 - 1)^2 + (0 - 0.270755272)^2 + (1 - 0.867303753)^2 + (0 - 0.861940976)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 0)^2 + (1 - 0)^2 + (1 - 0)^2 + (0 - 0)^2 + (0 - 0.729244728)^2 + (1 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (0 - 0.729244728)^2 + (0 - 0)^2 + (0 - 0)^2}$$

$$= 2,823253995$$

- Untuk data ke-1 ($x_{1,1}$ - $x_{1,21}$) untuk jarak bobot ke-7 ($w_{7,1}$ - $w_{7,21}$):

D7(x_{ij} - w_{ij})

$$= \sqrt{(0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 0)^2 + (0 - 1)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 0)^2 + (1 - 1)^2 + (1 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (0 - 1)^2 + (1 - 1)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (0 - 0)^2 + (0 - 1)^2}$$

$$= 2,828427125$$

- Untuk data ke-1 ($x_{1,1}$ - $x_{1,21}$) untuk jarak bobot ke-8 ($w_{8,1}$ - $w_{8,21}$):

D8(x_{ij} - w_{ij})

$$= \sqrt{(0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 0.911686257)^2 + (0 - 0.186362313)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 1)^2 + (1 - 0.266957287)^2 + (1 - 1.09804857)^2 + (0 - 1.09804857)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 1)^2 + (0 - 0)^2 + (0 - (-0.063383805))^2}$$

$$= 1,949159701$$

- Untuk data ke-1 ($x_{1,1}$ - $x_{1,21}$) untuk jarak bobot ke-9 ($w_{9,1}$ - $w_{9,21}$):

D9(x_{ij} - w_{ij})

$$= \sqrt{(0 - (-0.102309101))^2 + (0 - 0)^2 + (0 - (-0.102309101))^2 + (1 - 0)^2 + (0 - 0.828248179)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 0)^2 + (1 - 0)^2 + (1 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (1 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0.820837516)^2 + (0 - 1.102309101)^2 + (0 - 0)^2}$$

$$= 2,756045887$$



Setelah dilakukan proses perhitungan jarak untuk data ke-1 pada keseluruhan bobot akhir dengan menggunakan *Euclidean distance* maka didapatkan jarak terkecil adalah pada D3 (untuk menyatakan kelas 3, yakni Konjungtivitis) dengan nilai jarak sebesar 0.479276929. Berikut ini adalah Tabel 4.16 untuk menunjukkan keseluruhan hasil yang didapatkan pada proses perhitungan jarak pada data uji.

Tabel 4.16 Hasil dari Proses Perhitungan Jarak Terdekat dengan Menggunakan *Euclidean Distance* pada Data Uji

Data	D1	D2	D3	D4	D5	D6	D7	D8	D9	Min	Cj
1	2.948044	2.899607	0.479277	2.28909	2.741986	2.823254	2.828427	1.94916	2.756046	0.479277	3
2	2.749028	2.293877	2.270098	2.155402	2.208842	2.692003	2.645751	2.195152	1.65816	1.65816	9
3	2.749028	2.293877	2.209231	2.191548	2.208842	2.692003	2.236068	2.223839	2.225784	2.191548	4
4	2.505725	2.082017	2.729944	2.6952	0.568053	2.388592	2.828427	2.796193	2.481679	0.568053	5
5	2.773258	3.011575	1.816606	1.962143	2.918645	2.692003	3	1.493453	2.634254	1.493453	8

- Setelah didapatkan hasil dari jarak terdekat dengan menghitung data uji dan juga nilai bobot akhir yang diambil dari proses pelatihan, maka selanjutnya adalah menentukan hasil klasifikasi yang dihitung berdasarkan proses perhitungan akurasi yang didapatkan. Berikut adalah Tabel 4.17 untuk membandingkan nilai kelas pada target dan prediksi yang didapatkan.

Tabel 4.17 Perbandingan Kelas Target dan Prediksi pada Proses Pengujian

Data ke-	Kelas Target	Penyakit Mata (Kelas Target)	Prediksi	Penyakit Mata (Prediksi)
1	3	Konjungtivitis	3	Konjungtivitis
2	9	Kalazion	9	Kalazion
3	7	Dry Eyes	4	Keratitis
4	5	Pterigium	5	Pterigium
5	8	Bleparitis	8	Bleparitis

4.6.3 Proses perhitungan akurasi

Pada tahapan perhitungan akurasi, dilakukan pada saat proses pelatihan dan juga proses pengujian. Perhitungan ini menggunakan Persamaan 2.4 yakni:

$$Akurasi = \frac{\text{jumlah data yang benar}}{\text{total semua data yang ada}} \times 100\%$$

Sehingga perhitungan akurasi yang didapatkan dari proses pengujian dengan menggunakan 5 data uji adalah sebagai berikut.

$$Akurasi = \frac{4}{5} \times 100\% = 80\%$$



Dari hasil akurasi tersebut, didapatkan bahwa keseluruhan data yang bernilai benar untuk kelas prediksi terhadap kelas target adalah sebesar 80%, dengan data yang tidak sesuai ada pada data uji ke-3.

4.7 Perancangan desain antarmuka

Tahap perancangan desain antarmuka berguna untuk memudahkan pengguna dalam mengimplementasikan sistem yang terhubung dengan metode yang akan dibangun. Perancangan desain antarmuka terdiri atas antarmuka beranda, antarmuka dataset penyakit mata, antarmuka proses pelatihan (*training*) dengan LVQ, antarmuka proses pengujian (*testing*) dengan LVQ dan antarmuka proses uji coba.

4.7.1 Desain antarmuka data latih

Antarmuka data latih adalah antarmuka yang menampilkan informasi tentang data latih yang akan diambil untuk digunakan pada proses *training* LVQ. Gambar 4.6 menunjukkan antarmuka data latih dengan penjelasan sebagai berikut.

IDENTIFIKASI PENYAKIT MATA MENGGUNAKAN METODE <i>LEARNING VECTOR QUANTIZATION</i> (LVQ)					
1 DATA LATIH	LVQ-TRAINING	DATA UJI	LVQ-TESTING	HASIL IDENTIFIKASI	IDENTIFIKASI ...
2 PILIH DATA					
<input type="text"/>			3 BROWSE	4 AMBIL DATA	
5					

Gambar 4.6 Antarmuka Data Latih

Keterangan Gambar 4.6:

1. *Tab content* untuk menampilkan pengambilan data latih.
2. Menampilkan *input* data latih yang diambil dari file txt.
3. *Button* untuk menjalankan proses *browser* untuk memilih file data latih.
4. *Button* untuk menjalankan proses ambil data dari file txt yang telah dipilih.
5. Menampilkan informasi tabel dari data latih yang telah dipilih.

4.7.2 Desain antarmuka proses training

Antarmuka proses *training* adalah antarmuka yang menampilkan informasi tentang proses pelatihan dengan menggunakan metode LVQ pada data latih. Gambar 4.7 menunjukkan antarmuka proses *training* dengan penjelasan sebagai berikut.

IDENTIFIKASI PENYAKIT MATA MENGGUNAKAN METODE <i>LEARNING VECTOR QUANTIZATION</i> (LVQ)					
DATA LATIH	1 LVQ-TRAINING	DATA UJI	LVQ-TESTING	HASIL IDENTIFIKASI	IDENTIFIKASI ...
2 PARAMETER ALPHA <input type="text"/> 3 DEC ALPHA <input type="text"/> 4 MAX EPOCH <input type="text"/> 5 MIN ALPHA <input type="text"/> 6 <input type="button" value="PROSES TRAINING 7"/> <input type="button" value="RESET 8"/>		BOBOT JARINGAN (W) 9 D(Xij-Wij) 10 DELTA W 11 AKURASI 12			

Gambar 4.7 Antarmuka Proses Training

Keterangan Gambar 4.7:

1. *Tab content* menampilkan proses LVQ-Training.
2. Menampilkan parameter yang terdapat dalam proses LVQ-Training.
3. *Input* parameter alpha (*learning rate*).
4. *Input* parameter dec alpha (pengurangan *learning rate*).
5. *Input* parameter max epoch (iterasi maksimal).
6. *Input* parameter min alpha (minimum *learning rate*).
7. *Button* untuk menjalankan proses LVQ-Training.
8. *Button* untuk menjalankan proses reset untuk mengulang kembali proses LVQ-Training.
9. *Tab content* menampilkan hasil dari bobot jaringan (W).
10. *Tab content* menampilkan hasil dari perhitungan jarak dengan menggunakan *Euclidean Distance*.
11. *Tab content* menampilkan hasil dari perhitungan delta bobot jaringan (ΔW).

12. *Tab content* menampilkan hasil dari akurasi yang didapatkan pada proses LVQ-Training.

4.7.3 Desain antarmuka data uji

Antarmuka data uji adalah antarmuka yang menampilkan informasi tentang data uji yang akan diambil untuk digunakan pada proses *testing* LVQ. Gambar 4.8 menunjukkan antarmuka data uji dengan penjelasan sebagai berikut.



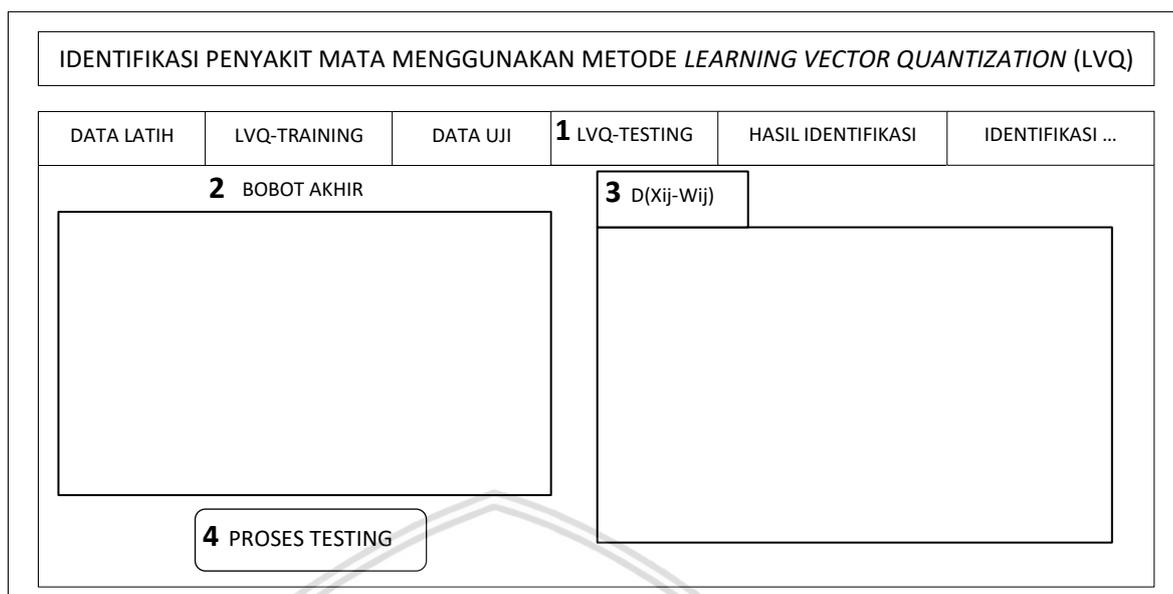
Gambar 4.8 Antarmuka Data Uji

Keterangan Gambar 4.8:

1. *Tab content* untuk menampilkan pengambilan data uji.
2. Menampilkan *input* data uji yang diambil dari file txt.
3. *Button* untuk menjalankan proses *browser* untuk memilih file data uji.
4. *Button* untuk menjalankan proses ambil data dari file txt yang telah dipilih.
5. Menampilkan informasi tabel dari data uji yang telah dipilih.

4.7.4 Desain antarmuka proses pengujian

Antarmuka proses pengujian adalah antarmuka yang menampilkan informasi tentang proses pelatihan dengan menggunakan metode LVQ pada data uji. Gambar 4.9 menunjukkan antarmuka proses pengujian dengan penjelasan sebagai berikut.



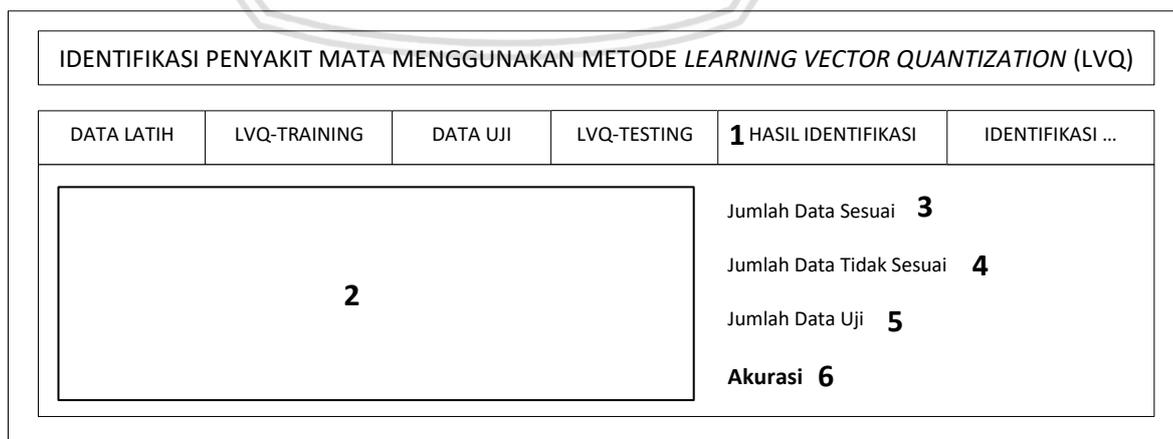
Gambar 4.9 Antarmuka Proses Pengujian

Keterangan Gambar 4.9:

1. *Tab content* untuk menampilkan proses LVQ-Testing.
2. Menampilkan bobot akhir yang diambil dari proses LVQ-Training.
3. *Tab content* untuk menampilkan hasil dari perhitungan jarak dengan menggunakan *Euclidean Distance*.
4. *Button* untuk menjalankan proses LVQ-Testing.

4.7.5 Desain antarmuka hasil identifikasi

Antarmuka hasil identifikasi adalah antarmuka yang menampilkan hasil dari identifikasi pada penyakit mata dengan menggunakan metode LVQ. Gambar 4.10 menunjukkan antarmuka hasil identifikasi dengan penjelasan sebagai berikut.



Gambar 4.10 Antarmuka Hasil Identifikasi

Keterangan Gambar 4.10:

1. *Tab content* untuk menampilkan proses hasil identifikasi.
2. Menampilkan tabel dari hasil proses LVQ-Testing.
3. Menampilkan hasil dari jumlah data yang sesuai pada proses hasil identifikasi LVQ-Testing.
4. Menampilkan hasil dari jumlah data yang tidak sesuai pada proses hasil identifikasi LVQ-Testing.
5. Menampilkan hasil dari jumlah data uji yang digunakan pada proses hasil identifikasi LVQ-Testing.
6. Menampilkan hasil dari akurasi proses identifikasi untuk LVQ-Testing.

4.7.6 Desain antarmuka identifikasi dengan *inputan user*

Antarmuka identifikasi dengan *inputan user* adalah antarmuka yang menampilkan hasil identifikasi dengan memasukkan dari pengguna yang memilih gejala yang ada, kemudian hasilnya akan menampilkan identifikasi berupa penyakit mata yang diperoleh. Gambar 4.11 menunjukkan desain antarmuka identifikasi dengan *inputan user* dengan penjelasan sebagai berikut.

IDENTIFIKASI PENYAKIT MATA MENGGUNAKAN METODE *LEARNING VECTOR QUANTIZATION* (LVQ)

DATA LATIH	LVQ-TRAINING	DATA UJI	LVQ-TESTING	HASIL IDENTIFIKASI	1 IDENTIFIKASI ...
------------	--------------	----------	-------------	--------------------	---------------------------

Pilih Gejala **2**

Sakit/Nyeri Kepala Hebat
 Penglihatan Kabur Perlahan
 Silau
 Merah
 Nyeri
 Perut Mual
 Penglihatan Berkabut (Berasap)
 Lensa Mata Keruh
 Gatal
 Berair

Belekan
 Kelopak Bengkak
 Panas
 Mengganjal
 Lengket
 Merah Jika Terkena Sinar Matahari
 Tumbuh Selaput pada Mata
 Timbul Bayangan
 Usia > 50 Tahun
 Kelopak Mata Timbul Benjolan
 Perih

Penyakit/Target **3**

4

5 Hasil Identifikasi

Gambar 4.11 Antarmuka Identifikasi dengan *Inputan User*

Keterangan Gambar 4.11:

1. *Tab content* untuk menampilkan proses identifikasi dengan *inputan user*.
2. *Check box* untuk memilih jenis gejala penyakit mata.



3. *Combo box* untuk memilih jenis penyakit mata.
4. *Button* untuk melakukan proses identifikasi berdasarkan *inputan user*.
5. Menampilkan hasil dari proses identifikasi berdasarkan *inputan user*.

4.8 Perancangan pengujian

Tahap perancangan pengujian dilakukan untuk mengetahui apakah sistem yang dibuat sudah sesuai dengan metode LVQ yang ditentukan. Selain itu, pada perancangan pengujian akan dilihat seberapa pengaruh nilai pada masing-masing parameter yang menunjang metode LVQ antara lain adalah pengaruh nilai *Learning Rate* (α), Pengurangan Nilai Rate (*Dec* α), *Minimal Learning Rate* (*Min* α), Iterasi Maksimum (*Max Epoch*) dan perbandingan jumlah data latih dan data uji yang berpengaruh terhadap hasil akurasi.

4.8.1 Perancangan pengujian pengaruh nilai *learning rate* (α) terhadap hasil akurasi

Pengujian untuk pengaruh nilai *learning rate* (α) terhadap akurasi akan dilakukan sebanyak 10 kali untuk masing-masing nilai *learning rate* (α) yang berbeda-beda. Hasil dari pengujian ini adalah untuk melihat hasil akurasi yang diperoleh berdasarkan perbedaan yang terletak pada parameter *learning rate* (α) untuk diambil rata-rata akurasi dari jumlah keseluruhan. Tabel 4.18 menampilkan perancangan pengujian atas pengaruh nilai *learning rate* (α) terhadap hasil akurasi.

Tabel 4.18 Perancangan Pengujian Pengaruh Nilai *Learning Rate* (α) terhadap Hasil Akurasi

No	<i>Learning Rate</i> (α)	Pengujian 1	...	Pengujian 10	Akurasi
1					
2					
...					
n					

4.8.2 Perancangan pengujian pengaruh pengurangan nilai rate (*dec* α) terhadap hasil akurasi

Pengujian untuk pengaruh nilai pengurangan nilai *rate* (*Dec* α) terhadap akurasi akan dilakukan sebanyak 10 kali untuk masing-masing pengurangan nilai *rate* (*Dec* α) yang berbeda-beda. Hasil dari pengujian ini adalah untuk melihat hasil akurasi yang diperoleh berdasarkan perbedaan yang terletak pada parameter pengurangan nilai *rate* (*Dec* α) untuk diambil rata-rata akurasi dari jumlah keseluruhan. Tabel 4.19 menampilkan perancangan pengujian atas pengaruh pengurangan nilai *rate* (*Dec* α) terhadap hasil akurasi.

Tabel 4.19 Perancangan Pengujian Pengaruh Pengurangan Nilai *Rate* (Dec α) terhadap Hasil Akurasi

No	Dec α	Pengujian 1	...	Pengujian 10	Akurasi
1					
2					
...					
n					

4.8.3 Perancangan pengujian pengaruh minimal *learning rate* (min α) terhadap hasil akurasi

Pengujian untuk pengaruh nilai *minimal learning rate* (Min α) terhadap akurasi akan dilakukan sebanyak 10 kali untuk masing-masing nilai *minimal learning rate* (Min α) yang berbeda-beda. Hasil dari pengujian ini adalah untuk melihat hasil akurasi yang diperoleh berdasarkan perbedaan yang terletak pada parameter *minimal learning rate* (Min α) untuk diambil rata-rata akurasi dari jumlah keseluruhan. Tabel 4.20 menampilkan perancangan pengujian atas pengaruh nilai *minimal learning rate* (Min α) terhadap hasil akurasi.

Tabel 4.20 Perancangan Pengujian Pengaruh Nilai *Minimal Learning Rate* (Min α) terhadap Hasil Akurasi

No	Min α	Pengujian 1	...	Pengujian 10	Akurasi
1					
2					
...					
n					

4.8.4 Perancangan pengujian pengaruh iterasi maksimum (Max Epoch) terhadap hasil akurasi

Pengujian untuk pengaruh nilai iterasi maksimum (Max Epoch) terhadap akurasi akan dilakukan sebanyak 10 kali untuk masing-masing nilai iterasi maksimum (Max Epoch) yang berbeda-beda. Hasil dari pengujian ini adalah untuk melihat hasil akurasi yang diperoleh berdasarkan perbedaan yang terletak pada nilai iterasi maksimum (Max Epoch) untuk diambil rata-rata akurasi dari jumlah keseluruhan. Tabel 4.21 menampilkan perancangan pengujian atas pengaruh nilai iterasi maksimum (Max Epoch) terhadap hasil akurasi.

Tabel 4.21 Perancangan Pengujian Pengaruh Nilai Iterasi Maksimum (Max Epoch) terhadap Hasil Akurasi

No	Max Epoch	Pengujian 1	...	Pengujian 10	Akurasi
1					

2					
...					
n					

4.8.5 Perancangan pengujian pengaruh perbandingan jumlah data latih dan data uji terhadap hasil akurasi

Pengujian untuk pengaruh perbandingan antara jumlah data latih dan data uji terhadap akurasi akan dilakukan sebanyak 10 kali untuk masing-masing jumlah data latih dan data uji yang berbeda-beda. Hasil dari pengujian ini adalah untuk melihat hasil akurasi yang diperoleh berdasarkan perbedaan yang terletak pada perbandingan jumlah data latih dan data uji untuk diambil rata-rata akurasi dari jumlah keseluruhan. Tabel 4.22 menampilkan perancangan pengujian atas pengaruh perbandingan antara jumlah data latih dan data uji terhadap hasil akurasi.

Tabel 4.22 Perancangan Pengujian Pengaruh Perbandingan Jumlah Data Latih dan Data Uji terhadap Hasil Akurasi

No	Data Latih:Data Uji	Pengujian 1	...	Pengujian 10	Akurasi
1					
2					
...					
n					

BAB 5 IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi sistem yang berguna untuk mendapatkan hasil dari analisis kebutuhan dan perancangan sistem yang telah dibuat sebelumnya. Pada bab implementasi, akan dibahas mengenai beberapa komponen yang menjadi jangkauan dalam implementasi sistem identifikasi penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ) yang sudah dibangun. Cakupan tersebut antara lain spesifikasi dari sistem, batasan pada implementasi, implementasi dari algoritme serta implementasi dari antarmuka sistem yang telah dibangun.

5.1 Spesifikasi sistem

Pada sub-bab ini akan dijelaskan kembali mengenai spesifikasi sistem yang sebelumnya sudah dibahas pada bab 3. Komponen pendukung sistem yang akan dibangun meliputi perangkat keras dan perangkat lunak dengan spesifikasi dari perangkat-perangkat tersebut adalah sebagai berikut.

5.1.1 Spesifikasi perangkat keras

Spesifikasi perangkat keras yang digunakan dalam implementasi sistem identifikasi penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ) dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras

No.	Nama Komponen	Spesifikasi
1	Processor	Intel Core i3-3217U CPU @ 1.80GHz (4 CPUs), ~1.80GHz
2	Memory (RAM)	10 GB
3	Hardisk	500 GB
4	Graphics Card	Intel HD Graphics 4000
5	Chip Type	Intel HD Graphics Family

5.1.2 Spesifikasi perangkat lunak

Spesifikasi perangkat lunak yang digunakan dalam implementasi sistem identifikasi penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ) dijelaskan pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak

No	Nama Komponen	Spesifikasi
1	Sistem Operasi	Windows 10 Pro 64-bit
2	Bahasa Pemrograman	Java
3	Media Pemrograman	Netbeans v.8.2

5.2 Batasan implementasi

Pada sub-bab ini akan dibahas mengenai batasan yang terdapat dalam implementasi sistem identifikasi penyakit mata dengan menggunakan metode *Learning Vector Quantization* (LVQ). Adapun batasan tersebut adalah:

1. Penerapan sistem yang dibangun akan didukung dengan spesifikasi baik untuk perangkat keras maupun perangkat lunak yang telah dijabarkan pada Tabel 5.1 dan Tabel 5.2.
2. Sistem ini dibangun dengan berbasiskan Java Desktop.
3. Metode yang digunakan pada sistem ini adalah *Learning Vector Quantization* (LVQ).
4. Data penyakit mata yang digunakan merupakan data sekunder dari penelitian milik Lestari dan kawan-kawan yang diperoleh dari poli mata RSUD Kabupaten Sidoarjo tahun 2015-2016 (Lestari et al., 2016).
5. Dalam proses implementasi sistem, terdapat 9 penyakit mata yang digunakan sebagai acuan dalam membuat sistem yakni Glaukoma, Katarak, Konjungtivitis, Keratitis, Pterigium, Uveitis, Dry Eyes, Bleparitis, dan Kalazion.
6. Dalam proses implementasi sistem, terdapat 21 gejala pada penyakit mata yang digunakan sebagai acuan dalam membuat sistem yakni Sakit/Nyeri Kepala Hebat, Penglihatan Kabur Perlahan, Silau, Merah, Nyeri, Perut Mual, Penglihatan Berkabut (Berasap), Lensa Mata Keruh, Gatal, Berair, Belekan, Kelopak Bengkak, Panas, Mengganjal, Lengket, Merah Jika Terkena Sinar Matahari, Tumbuh Selaput pada Mata, Timbul Bayangan, Usia < 50 Tahun, Kelopak Mata Timbul Benjolan, dan Perih.
7. *Input* yang diperoleh oleh sistem adalah berupa gejala-gejala serta penyakit mata yang diambil pada data penyakit mata.
8. *Output* yang diperoleh oleh sistem adalah berupa hasil perhitungan berdasarkan metode LVQ, jumlah data yang sesuai dan data yang tidak sesuai serta hasil persentase akurasi dari sistem yang dibangun.

5.3 Implementasi algoritme

Pada sub-bab ini akan dibahas mengenai implementasi algoritme pada sistem yang dibangun. Pembahasan sub-bab ini bertujuan untuk menjelaskan setiap bagian dari kode program yang telah dibuat berdasarkan pada acuan

perancangan yang telah dibahas sebelumnya. Dalam implementasi algoritme yang menggunakan aplikasi Netbeans dengan bahas pemrograman Java, berikut ini adalah penjelasan mengenai metode tahapan yang diimplementasikan yakni metode *Learning Vector Quantization* (LVQ) dalam identifikasi penyakit mata.

5.3.1 Metode *Learning Vector Quantization* (LVQ)

Dalam menerapkan metode *Learning Vector Quantization* (LVQ) untuk mendapatkan klasifikasi penyakit mata yang sesuai dengan masukkan gejala, ada beberapa tahapan yang perlu dikerjakan seperti pada pembahasan sub-bab 2.5. Tahapan tersebut menjalankan beberapa proses yang harus dikerjakan untuk mendapatkan hasil yang diinginkan. Proses tersebut antara lain adalah menentukan bobot awal, perhitungan jarak minimum, pelatihan LVQ dan pengujian LVQ. Dalam implementasi algoritme ini, penulis hanya menjabarkan beberapa algoritme proses terkait dengan metode LVQ yang digunakan untuk memperoleh hasil identifikasi pada penyakit mata.

5.3.1.1 Algoritme menentukan bobot awal

Dalam menerapkan algoritme LVQ, langkah pertama yang harus dilakukan adalah menentukan bobot awal agar dapat memproses algoritme pelatihan LVQ. Proses yang dijalankan dalam menentukan bobot awal adalah mengambil nilai data secara acak pada data *training* yang diwakilkan untuk masing-masing target yang ada disetiap kelasnya. Sehingga setiap bobot awal yang diambil menyatakan target kelas yang telah ditetapkan sama sejumlah dengan penyakit mata yang ada. Berikut ini adalah kode sumber untuk algoritme dalam menentukan bobot awal yang ditunjukkan pada Kode Sumber 5.1.

Algoritme 1: Fungsi menentukan bobot awal

```

1 public void Set_Bobot_Awal(double Data_Latih[][], double Target[])
2 {
3     this.Bobot_Awal = new double[9 * 21];
4     double titik_cek[] = {1.0, 2.0, 3.0, 4.0, 5.0, 6.0,
5     7.0, 8.0, 9.0};
6     int count;
7     int acak;
8
9     for (int i = 0; i < titik_cek.length; i++) {
10        acak = 0 +
11        this.rand_value.nextInt(Data_Latih.length);
12
13        while (Target[acak] != titik_cek[i]) {
14            acak = 0 +
15            this.rand_value.nextInt(Data_Latih.length);
16        }
17
18        if (titik_cek[i] == 1.0) {
19            for (int j = 0; j < 21; j++) {
20                this.Bobot_Awal[j] = Data_Latih[acak][j];
21            }
22        } else if (titik_cek[i] == 2.0) {
23            count = 0;
24            for (int j = 21; j < 42; j++) {
25                this.Bobot_Awal[j] =
26                Data_Latih[acak][count];
27                count++;

```

```

28     }
29     } else if (titik_cek[i] == 3.0) {
30         count = 0;
31         for (int j = 42; j < 63; j++) {
32             this.Bobot_Awal[j] =
33 Data_Latih[acak][count];
34             count++;
35         }
36     } else if (titik_cek[i] == 4.0) {
37         count = 0;
38         for (int j = 63; j < 84; j++) {
39             this.Bobot_Awal[j] =
40 Data_Latih[acak][count];
41             count++;
42         }
43     } else if (titik_cek[i] == 5.0) {
44         count = 0;
45         for (int j = 84; j < 105; j++) {
46             this.Bobot_Awal[j] =
47 Data_Latih[acak][count];
48             count++;
49         }
50     } else if (titik_cek[i] == 6.0) {
51         count = 0;
52         for (int j = 105; j < 126; j++) {
53             this.Bobot_Awal[j] =
54 Data_Latih[acak][count];
55             count++;
56         }
57     } else if (titik_cek[i] == 7.0) {
58         count = 0;
59         for (int j = 126; j < 147; j++) {
60             this.Bobot_Awal[j] =
61 Data_Latih[acak][count];
62             count++;
63         }
64     } else if (titik_cek[i] == 8.0) {
65         count = 0;
66         for (int j = 147; j < 168; j++) {
67             this.Bobot_Awal[j] =
68 Data_Latih[acak][count];
69             count++;
70         }
71     } else if (titik_cek[i] == 9.0) {
72         count = 0;
73         for (int j = 168; j < 189; j++) {
74             this.Bobot_Awal[j] =
75 Data_Latih[acak][count];
76             count++;
77         }
78     }
79 }
80 }

```

Kode Sumber 5.1 Menentukan Bobot Awal

Penjelasan pada Kode Sumber 5.1:

- 1) Baris 3-7 merupakan inisialisasi parameter dan variabel yang akan digunakan untuk proses menentukan bobot awal.
- 2) Baris 9-11 merupakan fungsi pengulangan untuk membuat nilai dari 0 sampai sebanyak data latih.

- 3) Baris 13-16 merupakan fungsi pengulangan untuk mengecek apakah nilai target data (acak) tidak sama dengan titik cek yang telah ditentukan. Hal ini dilakukan agar titik cek dapat digunakan untuk menghindari nilai acak yang muncul berkali-kali.
- 4) Baris 18-78 merupakan fungsi untuk mengecek kondisi titik cek apakah sudah sesuai dengan target data (acak). Apabila nilai pada target data (acak) sudah sesuai dengan titik acak, maka titik cek tersebut akan diperiksa terlebih dahulu kemudian dapat disesuaikan dengan target penyakit yang ada.

5.3.1.2 Algoritme pelatihan Learning Vector Quantization (LVQ)

Algoritme pelatihan *Learning Vector Quantization* (LVQ) adalah proses untuk mendapatkan bobot akhir diakhir proses, bobot akhir tersebut akan digunakan pada proses pengujian LVQ. Didalam algoritme pelatihan LVQ terdapat beberapa tahapan yang harus dikerjakan untuk mendapatkan bobot akhir, yakni menentukan bobot awal, melakukan perhitungan jarak minimum kemudian melakukan update bobot sampai mendapatkan bobot akhir. Sebelumnya, telah diuraikan algoritme untuk menentukan bobot awal (pada sub-bab 5.3.1.1) dan perhitungan jarak minimum (pada sub-bab 5.3.1.2). Sehingga pada tahapan ini akan jelaskan mengenai pencarian nilai *class/category* berdasarkan pada perhitungan jarak minimum yang telah dilakukan sebelumnya dan update bobot. Berikut ini adalah kode sumber untuk algoritme pelatihan LVQ pada tahapan update bobot yang ditunjukkan pada Kode Sumber 5.2, Kode Sumber 5.3 dan Kode Sumber 5.4.

Algoritme 4: Fungsi pencarian nilai <i>class/category</i> berdasarkan perhitungan jarak minimum	
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22	<pre> public void Cari_Ci(double Di_Minimum[], double D_i[], int index) { if (Di_Minimum[index] == D_i[0]) { this.C_i[index] = 1.0; } else if (Di_Minimum[index] == D_i[1]) { this.C_i[index] = 2.0; } else if (Di_Minimum[index] == D_i[2]) { this.C_i[index] = 3.0; } else if (Di_Minimum[index] == D_i[3]) { this.C_i[index] = 4.0; } else if (Di_Minimum[index] == D_i[4]) { this.C_i[index] = 5.0; } else if (Di_Minimum[index] == D_i[5]) { this.C_i[index] = 6.0; } else if (Di_Minimum[index] == D_i[6]) { this.C_i[index] = 7.0; } else if (Di_Minimum[index] == D_i[7]) { this.C_i[index] = 8.0; } else if (Di_Minimum[index] == D_i[8]) { this.C_i[index] = 9.0; } } </pre>

Kode Sumber 5.2 Pencarian Nilai Class/Category Berdasarkan Perhitungan Jarak Minimum



Penjelasan Kode Sumber 5.2:

- 1) Baris 3-21 terdapat beberapa kondisi yang diproses untuk memperoleh *class* berdasarkan jarak minimum yang telah dihitung sebelumnya. Pada proses ini, apabila jarak minimum yang didapatkan berada pada index 1 maka nilai *class/category* yang akan digunakan adalah target 1. Hal tersebut berlaku juga kepada target berikutnya yakni target 2-target 9.

Algoritme 5: Fungsi pelatihan *Learning Vector Quantization* (LVQ) untuk hitung delta bobot (ΔW)

```

1 public void Hitung_Delta_W(double Data_Latih[][[]], double
2 Bobot_Awal[], double Alpha, int index) {
3     this.Delta_W = new double[Bobot_Awal.length];
4     int counter = 0;
5     for (int i = 0; i < 9; i++) {
6         for (int j = 0; j < Data_Latih[0].length; j++) {
7             this.Delta_W[counter] = Alpha *
8 (Data_Latih[index][j] - Bobot_Awal[counter]);
9             counter++;
10        }
11    }
12 }

```

Kode Sumber 5.3 Pelatihan *Learning Vector Quantization* (LVQ) untuk Hitung Delta Bobot (ΔW)

Penjelasan Kode Sumber 5.3:

- 1) Baris 3-4 merupakan inisialisasi variabel dan parameter yang akan digunakan pada proses perhitungan *update* bobot.
- 2) Baris 5-9 merupakan proses pengulangan untuk membuat nilai 0 sampai dengan 9 yang digunakan sebagai penentu banyaknya bobot yang akan dihitung. Didalam kondisi tersebut, terdapat pula kondisi pengulangan untuk membuat nilai dari 0 sampai dengan banyaknya data latih yang akan melakukan proses perhitungan yakni *learning rate* (α) * (data latih – bobot awal).

Algoritme 6: Fungsi pelatihan *Learning Vector Quantization* (LVQ) untuk hitung bobot (W)

```

1 public void Hitung_W(double Target[], double C_i[], double
2 Delta_W[], int index, double Bobot_Awal[]) {
3     this.Bobot_Akhir = new double[Delta_W.length];
4     int Pembatas_1 = (this.Bobot_Akhir.length / 9) * 1;
5     int Pembatas_2 = (this.Bobot_Akhir.length / 9) * 2;
6     int Pembatas_3 = (this.Bobot_Akhir.length / 9) * 3;
7     int Pembatas_4 = (this.Bobot_Akhir.length / 9) * 4;
8     int Pembatas_5 = (this.Bobot_Akhir.length / 9) * 5;
9     int Pembatas_6 = (this.Bobot_Akhir.length / 9) * 6;
10    int Pembatas_7 = (this.Bobot_Akhir.length / 9) * 7;
11    int Pembatas_8 = (this.Bobot_Akhir.length / 9) * 8;
12    int Pembatas_9 = (this.Bobot_Akhir.length / 9) * 9;
13
14    for (int i = 0; i < this.Bobot_Akhir.length; i++) {
15        if (i < Pembatas_1) {
16            if (Target[index] == C_i[index]) {
17                if (C_i[index] == 1.0) {
18                    this.Bobot_Akhir[i] = Bobot_Awal[i] +
19 Delta_W[i];
20                } else {

```



```

21         this.Bobot_Akhir[i] = Bobot_Awal[i];
22     }
23     } else {
24         if (C_i[index] == 1.0) {
25             this.Bobot_Akhir[i] = Bobot_Awal[i] -
26 Delta_W[i];
27         } else {
28             this.Bobot_Akhir[i] = Bobot_Awal[i];
29         }
30     }
31     } else if (i >= Pembatas_1 && i < Pembatas_2) {
32         if (Target[index] == C_i[index]) {
33             if (C_i[index] == 2.0) {
34                 this.Bobot_Akhir[i] = Bobot_Awal[i] +
35 Delta_W[i];
36             } else {
37                 this.Bobot_Akhir[i] = Bobot_Awal[i];
38             }
39         } else {
40             if (C_i[index] == 2.0) {
41                 this.Bobot_Akhir[i] = Bobot_Awal[i] -
42 Delta_W[i];
43             } else {
44                 this.Bobot_Akhir[i] = Bobot_Awal[i];
45             }
46         }
47     }
48     } else if (i >= Pembatas_2 && i < Pembatas_3) {
49         if (Target[index] == C_i[index]) {
50             if (C_i[index] == 3.0) {
51                 this.Bobot_Akhir[i] = Bobot_Awal[i] +
52 Delta_W[i];
53             } else {
54                 this.Bobot_Akhir[i] = Bobot_Awal[i];
55             }
56         } else {
57             if (C_i[index] == 3.0) {
58                 this.Bobot_Akhir[i] = Bobot_Awal[i] -
59 Delta_W[i];
60             } else {
61                 this.Bobot_Akhir[i] = Bobot_Awal[i];
62             }
63         }
64     } else if (i >= Pembatas_3 && i < Pembatas_4) {
65         if (Target[index] == C_i[index]) {
66             if (C_i[index] == 4.0) {
67                 this.Bobot_Akhir[i] = Bobot_Awal[i] +
68 Delta_W[i];
69             } else {
70                 this.Bobot_Akhir[i] = Bobot_Awal[i];
71             }
72         } else {
73             if (C_i[index] == 4.0) {
74                 this.Bobot_Akhir[i] = Bobot_Awal[i] -
75 Delta_W[i];
76             } else {
77                 this.Bobot_Akhir[i] = Bobot_Awal[i];
78             }
79         }
80     } else if (i >= Pembatas_4 && i < Pembatas_5) {
81         if (Target[index] == C_i[index]) {
82             if (C_i[index] == 5.0) {
83                 this.Bobot_Akhir[i] = Bobot_Awal[i] +
84 Delta_W[i];
85             } else {

```

```

86         this.Bobot_Akhir[i] = Bobot_Awal[i];
87     }
88     } else {
89         if (C_i[index] == 5.0) {
90             this.Bobot_Akhir[i] = Bobot_Awal[i] -
91 Delta_W[i];
92         } else {
93             this.Bobot_Akhir[i] = Bobot_Awal[i];
94         }
95     }
96     } else if (i >= Pembatas_5 && i < Pembatas_6) {
97         if (Target[index] == C_i[index]) {
98             if (C_i[index] == 6.0) {
99                 this.Bobot_Akhir[i] = Bobot_Awal[i] +
100 Delta_W[i];
101             } else {
102                 this.Bobot_Akhir[i] = Bobot_Awal[i];
103             }
104         } else {
105             if (C_i[index] == 6.0) {
106                 this.Bobot_Akhir[i] = Bobot_Awal[i] -
107 Delta_W[i];
108             } else {
109                 this.Bobot_Akhir[i] = Bobot_Awal[i];
110             }
111         }
112     } else if (i >= Pembatas_6 && i < Pembatas_7) {
113         if (Target[index] == C_i[index]) {
114             if (C_i[index] == 7.0) {
115                 this.Bobot_Akhir[i] = Bobot_Awal[i] +
116 Delta_W[i];
117             } else {
118                 this.Bobot_Akhir[i] = Bobot_Awal[i];
119             }
120         } else {
121             if (C_i[index] == 7.0) {
122                 this.Bobot_Akhir[i] = Bobot_Awal[i] -
123 Delta_W[i];
124             } else {
125                 this.Bobot_Akhir[i] = Bobot_Awal[i];
126             }
127         }
128     } else if (i >= Pembatas_7 && i < Pembatas_8) {
129         if (Target[index] == C_i[index]) {
130             if (C_i[index] == 8.0) {
131                 this.Bobot_Akhir[i] = Bobot_Awal[i] +
132 Delta_W[i];
133             } else {
134                 this.Bobot_Akhir[i] = Bobot_Awal[i];
135             }
136         } else {
137             if (C_i[index] == 8.0) {
138                 this.Bobot_Akhir[i] = Bobot_Awal[i] -
139 Delta_W[i];
140             } else {
141                 this.Bobot_Akhir[i] = Bobot_Awal[i];
142             }
143         }
144     } else if (i >= Pembatas_8 && i < Pembatas_9) {
145         if (Target[index] == C_i[index]) {
146             if (C_i[index] == 9.0) {
147                 this.Bobot_Akhir[i] = Bobot_Awal[i] +
148 Delta_W[i];
149             } else {
150                 this.Bobot_Akhir[i] = Bobot_Awal[i];

```

```

151     }
152     } else {
153         if (C_i[index] == 9.0) {
154             this.Bobot_Akhir[i] = Bobot_Awal[i] -
Delta_W[i];
155         } else {
156             this.Bobot_Akhir[i] = Bobot_Awal[i];
157         }
158     }
159 }
160 }
161 }
162 }

```

Kode Sumber 5.4 Pelatihan *Learning Vector Quantization* (LVQ) untuk Hitung Bobot (W)

Penjelasan Kode Sumber 5.4:

- 1) Baris 3-12 merupakan inisialisasi variabel dan parameter untuk proses perhitungan *update* bobot pada tahapan perhitungan bobot (W).
- 2) Baris 14-161 merupakan pengulangan dengan kondisi tertentu. Apabila target sama dengan *class/category* (variabel C_i) yang merepresentasikan dari jarak minimal yang telah dihitung sebelumnya, maka proses yang akan dijalankan untuk *update* bobot adalah bobot awal + delta W (ΔW). Sebaliknya, jika target tidak sama dengan *class/category* yang telah didapatkan, maka proses *update* bobot adalah bobot awal – delta W (ΔW). Bobot akan *diupdate* apabila *class* yang didapatkan adalah *class* minimum. Sedangkan bobot lainnya akan mengambil nilai bobot awal/bobot sebelumnya. Hal ini berlaku untuk keseluruhan target penyakit mata, baik dari target 1 sampai dengan target 9.

5.3.1.3 Algoritme pengujian *Learning Vector Quantization* (LVQ)

Pada algoritme ini, proses pengujian LVQ tahapannya sama dengan proses pelatihan LVQ. Alur yang dikerjakan algoritme ini antara lain adalah menghitung jarak minimum dan mencari nilai *class/category* yang disesuaikan dengan perhitungan jarak minimum. Algoritme pengujian LVQ menggunakan bobot akhir yang didapatkan dari proses perhitungan pada pelatihan LVQ. Kemudian setelah didapatkan hasilnya, maka algoritme ini akan menghitung akurasi serta mengecek kesesuaian antara data uji dan hasil klasifikasi yang didapatkan pada metode LVQ. Berikut ini adalah kode sumber untuk algoritme pengujian LVQ untuk menghitung nilai akurasi serta mengecek kesesuaian antara data uji dan hasil data klasifikasi menggunakan metode LVQ yang ditunjukkan pada Kode Sumber 5.5 dan Kode Sumber 5.6.

Algoritme 7: Fungsi pengujian *Learning Vector Quantization* (LVQ) untuk menghitung nilai akurasi

```

1 public void Hitung_Akurasi(double Target_Uji[], double C_j[]) {
2     Data_Sesuai = 0;
3
4     for (int i = 0; i < Target_Uji.length; i++) {
5         if (Target_Uji[i] == C_j[i]) {
6             Data_Sesuai += 1;
7         }
8     }

```

```

9
10         this.Akurasi = (Data_Sesuai / (double)
11 Target_Uji.length) * 100;
12     }
    
```

Kode Sumber 5.5 Pengujian *Learning Vector Quantization* (LVQ) untuk Menghitung Nilai Akurasi

Penjelasan Kode Sumber 5.5:

- 1) Baris 2 merupakan inisialisasi variabel `Data_Sesuai` yang bernilai 0. Variabel ini nantinya akan digunakan sebagai acuan untuk menghitung akurasi pada metode LVQ.
- 2) Baris 4-11 merupakan fungsi pengulangan untuk kondisi nilai 0 sampai dengan banyaknya target data uji yang telah ditetapkan. Dalam fungsi tersebut, terdapat kondisi apabila target uji sama dengan *class/category* yang telah didapatkan dari perhitungan jarak minimum maka data yang sesuai akan mengalami penambahan nilai 1. Kemudian setelah proses tersebut diselesaikan, selanjutnya adalah menjalankan fungsi perhitungan akurasi yang digunakan untuk memperoleh persentase akurasi.

Algoritme 8: Fungsi pengujian *Learning Vector Quantization* (LVQ) untuk mengecek kesesuaian data

```

1 public void Cek_Kesesuaian(double Target_Uji[], double C_j[]) {
2     this.Kesesuaian = new String[Target_Uji.length];
3
4     for (int i = 0; i < Kesesuaian.length; i++) {
5         if (Target_Uji[i] == C_j[i]) {
6             Kesesuaian[i] = "Sesuai";
7         } else {
8             Kesesuaian[i] = "Tidak Sesuai";
9         }
10    }
11
12 }
    
```

Kode Sumber 5.6 Pengujian *Learning Vector Quantization* (LVQ) untuk Mengecek Kesesuaian Data

Penjelasan Kode Sumber 5.6:

- 1) Baris 2 merupakan inisialisasi parameter untuk membentuk object String baru pada target data uji yang akan digunakan untuk menghitung kesesuaian data antara data uji dan data hasil klasifikasi menggunakan metode LVQ.
- 2) Baris 4-11 merupakan fungsi pengulangan yang dilakukan untuk membuat nilai 0 sampai dengan banyaknya kesesuaian data yang diperoleh. Didalam fungsi tersebut, terdapat kondisi apabila target untuk data uji sama dengan *class/category* maka akan melakukan *print* "Sesuai". Sebaliknya, apabila target untuk data uji tidak sama dengan *class/category* maka akan melakukan *print* "Tidak Sesuai".

5.3.1.4 Algoritme perhitungan jarak minimum

Perhitungan algoritme jarak minimum menggunakan rumus *Euclidean distance*. Hal ini dilakukan pada proses pelatihan LVQ dan juga pengujian LVQ.



Algoritme perhitungan jarak minimum dilakukan agar hasil dari proses ini digunakan sebagai acuan untuk melakukan perubahan bobot. Hasil yang diperoleh dengan nilai paling sedikit atau paling minimum akan memberikan dampak kepada bobot yang akan diubah, sedangkan bobot yang lain akan bernilai tetap. Berikut ini adalah kode sumber untuk algoritme perhitungan jarak minimum yang ditunjukkan pada Kode Sumber 5.7 dan Kode Sumber 5.8.

Algoritme 2: Fungsi perhitungan jarak minimum (<i>Euclidean distance</i>)	
1	public void Hitung_D_i(int index, double Bobot_Awal[], double
2	Data_Latih[][]) {
3	this.D_i = new double[9];
4	int count = 0;
5	for (int i = 0; i < D_i.length; i++) {
6	double N = 0.0;
7	for (int j = 0; j < Data_Latih[0].length; j++) {
8	N += Math.pow(Data_Latih[index][j] -
9	Bobot_Awal[count], 2);
10	count++;
11	}
12	this.D_i[i] = Math.sqrt(N);
13	}
14	}

Kode Sumber 5.7 Perhitungan Jarak Minimum (*Euclidean Distance*)

Penjelasan Kode Sumber 5.7:

- 1) Baris 3-4 merupakan inisialisasi variabel dan parameter untuk proses perhitungan jarak minimum.
- 2) Baris 5-7 merupakan kondisi pengulangan untuk membuat nilai dari 0 sampai sebanyak data latih.
- 3) Baris 8-12 merupakan fungsi perhitungan jarak dengan menggunakan rumus *Euclidean distance* dengan variabel yang akan dipakai dalam rumus ini adalah data latih dan bobot awal yang sebelumnya telah ditentukan.

Algoritme 3: Fungsi perhitungan jarak minimum (cari nilai minimum)	
1	public void Di_Minimum(double D_i[], int index) {
2	double min = D_i[0];
3	
4	for (int i = 0; i < D_i.length; i++) {
5	if (D_i[i] < min) {
6	min = D_i[i];
7	}
8	}
9	
10	this.Di_Minimum[index] = min;
11	
12	}

Kode Sumber 5.8 Perhitungan Jarak Minimum (Cari Nilai Minimum)

Penjelasan Kode Sumber 5.8:

- 1) Baris 2 merupakan inisialisasi variabel min yang akan digunakan untuk menyimpan nilai minimum yang didapatkan dari proses perhitungan jarak dengan menggunakan *Euclidean distance*.

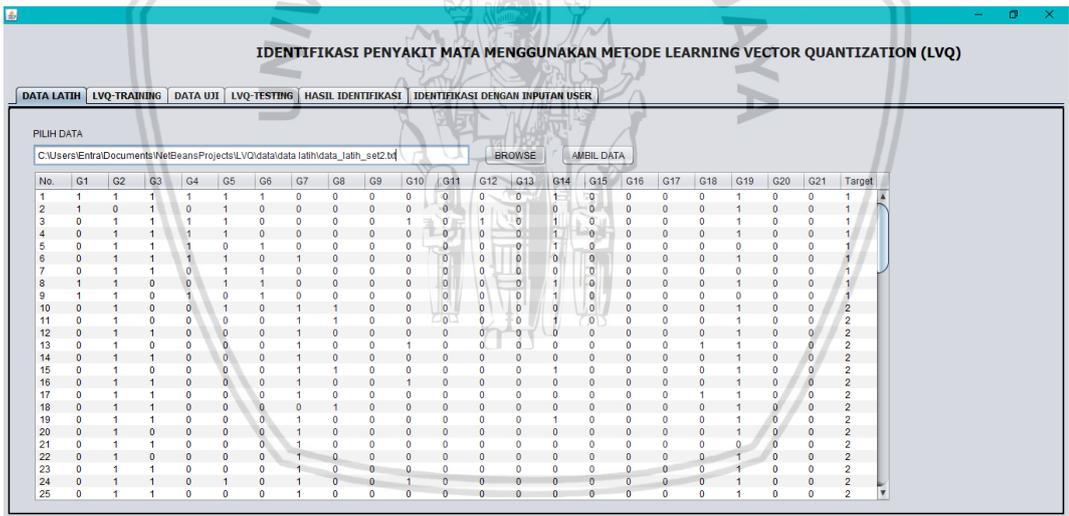
Baris 4-10 merupakan fungsi pengulangan untuk membuat nilai 0 sampai dengan banyaknya nilai variabel D_i . Apabila kondisi variabel D_i kurang dari batas minimum yang telah ditentukan maka nilai D_i tersebut akan diambil sebagai nilai minimum. Jika tidak, maka nilai minimum sebelumnya yang akan dipakai.

5.4 Implementasi antarmuka

Pada sub-bab ini, implementasi antarmuka untuk identifikasi penyakit mata dengan menggunakan metode LVQ berupa halaman data latih, halaman LVQ-*training*, halaman data uji, halaman LVQ-*testing*, dan halaman hasil identifikasi. Berikut adalah uraian untuk implementasi antarmuka pada penelitian ini.

5.4.1 Implementasi halaman data latih

Halaman data latih adalah halaman pertama yang akan tampil saat program sistem identifikasi penyakit mata dijalankan oleh pengguna. Pada halaman ini, tampilan yang tersedia berupa tabel yang akan menampilkan data latih yang dipilih oleh pengguna. Dalam tampilan ini pun, terdapat 2 *button* yakni *button Browse* yang berfungsi untuk mengakses direktori file dan *button Ambil Data* berfungsi untuk mengambil data latih yang telah dipilih oleh pengguna. Halaman data latih ditunjukkan oleh Gambar 5.1.



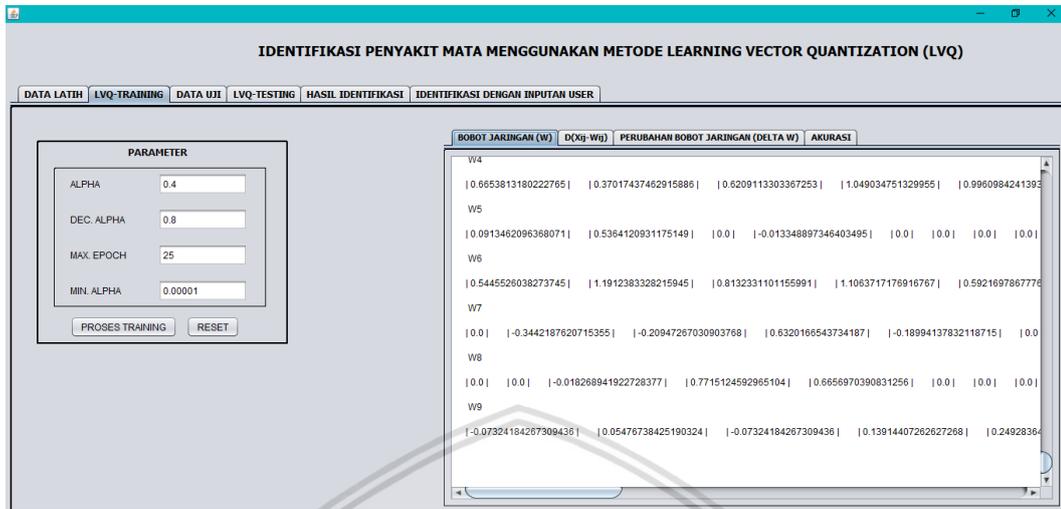
Gambar 5.1 Implementasi Halaman Data Latih

5.4.2 Implementasi halaman LVQ-training

Halaman LVQ-*training* adalah halaman kedua yang akan muncul saat pengguna hendak melakukan proses perhitungan data latih dengan menggunakan metode LVQ. Pada halaman ini, tampilan yang diberikan kepada pengguna berupa parameter dan beberapa tahapan hasil dari proses LVQ-*training*. Pada kolom parameter, pengguna akan mengisi nilai dari *learning rate* (α), pengurangan *learning rate* ($dec \alpha$), iterasi maksimum (*max epoch*) serta *minimal learning rate* ($min \alpha$). Sedangkan pada beberapa tahapan LVQ-*training*, terdapat beberapa area yang akan menunjukkan hasil dari bobot jaringan (W), perhitungan jarak minimum dengan menggunakan *Euclidean distance*,



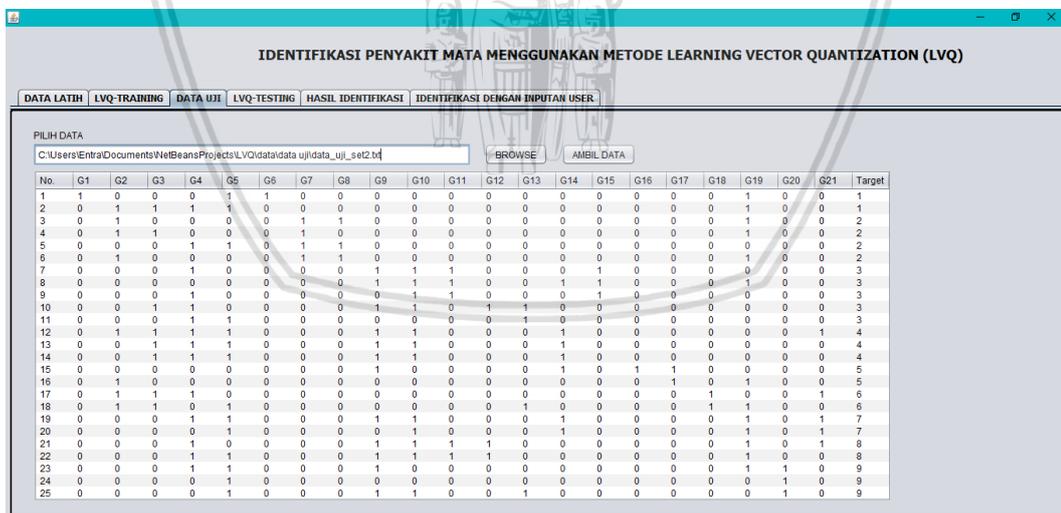
perubahan bobot jaringan (ΔW) serta akurasi. Halaman LVQ-training ditunjukkan pada Gambar 5.2.



Gambar 5.2 Implementasi Halaman LVQ-Training

5.4.3 Implementasi halaman data uji

Halaman data uji adalah halaman ketiga yang akan tampil saat program sistem identifikasi penyakit mata dijalankan oleh pengguna. Pada halaman ini, tampilan yang tersedia berupa tabel yang akan menampilkan data uji yang dipilih oleh pengguna. Dalam tampilan ini pun, terdapat 2 *button* yakni *button Browse* dan *button Ambil Data*. Halaman data latih ditunjukkan oleh Gambar 5.3.

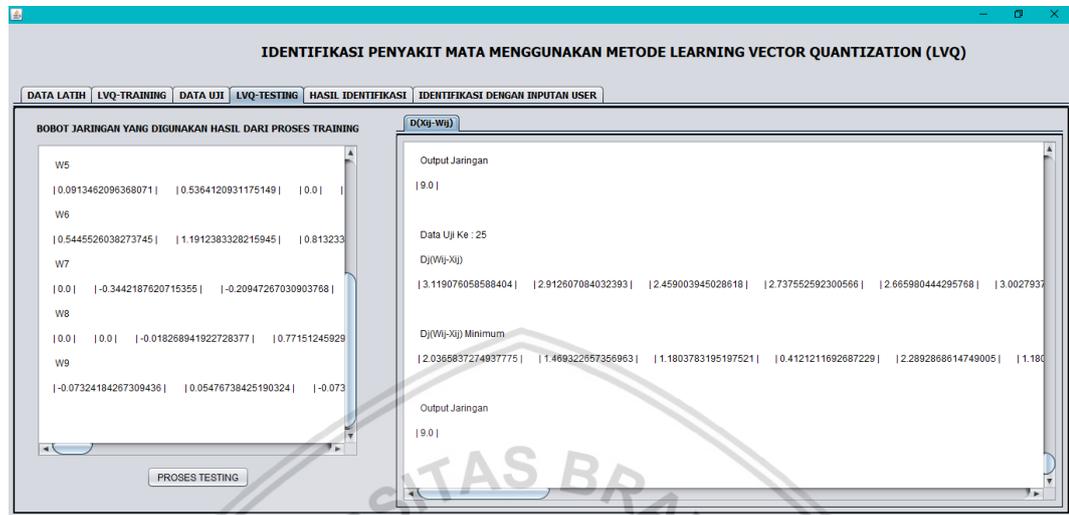


Gambar 5.3 Implementasi Halaman Data Uji

5.4.4 Implementasi halaman LVQ-testing

Halaman LVQ-testing adalah halaman keempat yang akan muncul saat pengguna hendak melakukan proses perhitungan data uji dengan menggunakan metode LVQ. Pada halaman ini, tampilan yang diberikan kepada pengguna berupa area yang menampilkan hasil yang didapatkan dari perhitungan LVQ-

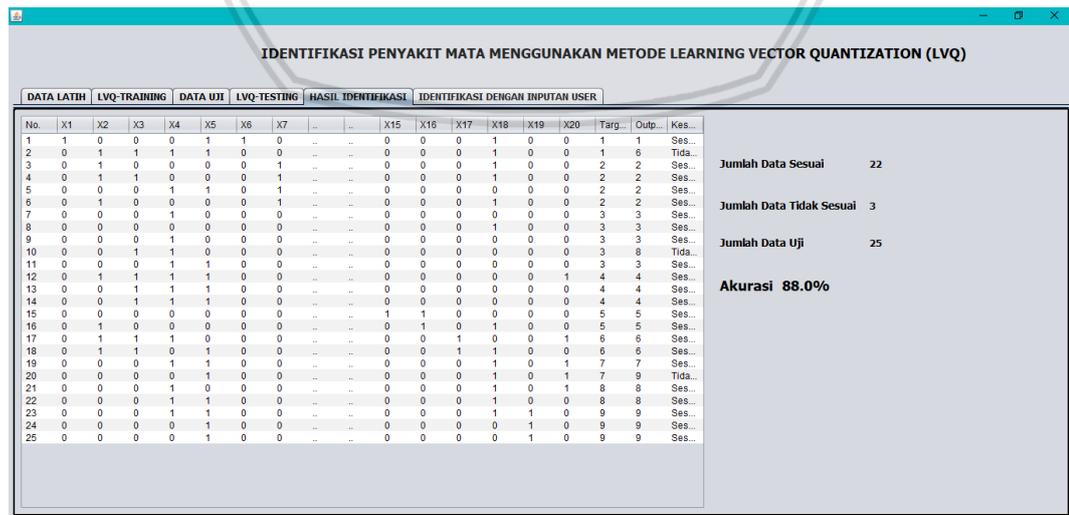
training yakni bobot akhir. Kemudian terdapat *button* Proses Testing yang akan menjalankan proses perhitungan LVQ-testing. Kemudian terdapat pula area hasil perhitungan jarak minimum pada proses pengujian LVQ. Halaman LVQ-testing ditunjukkan pada Gambar 5.4.



Gambar 5.4 Implementasi Halaman LVQ-Testing

5.4.5 Implementasi halaman hasil identifikasi

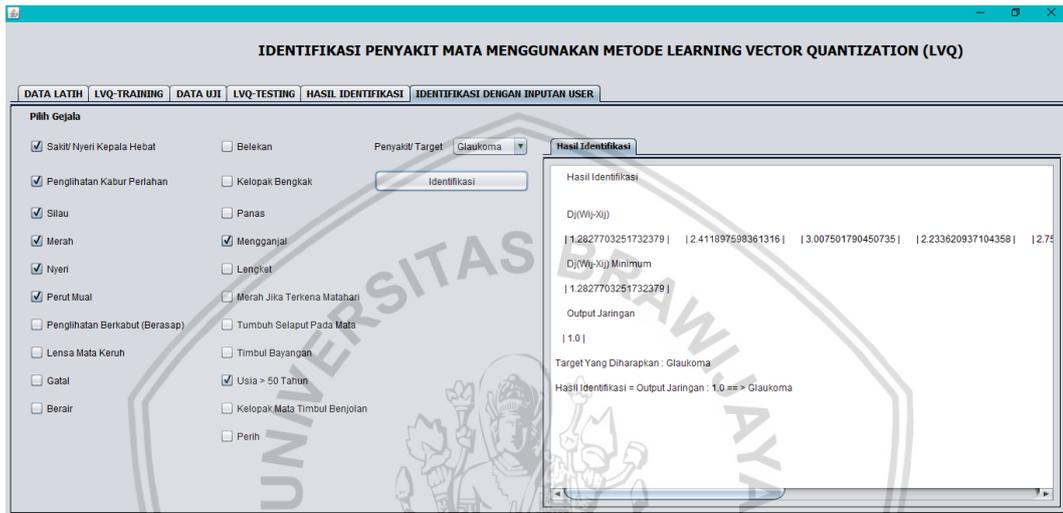
Halaman hasil identifikasi merupakan halaman yang akan tampil saat pengguna menjalankan program penelitian ini. Pada halaman ini, tampilan yang diberikan kepada pengguna berupa tabel yang berisi data uji yang telah dipilih sebelumnya. Didalam tabel tersebut, pengguna akan mengetahui kesesuaian antara data uji dan data sebenarnya. Kemudian ada beberapa tampilan yang menunjukkan kesesuaian data, ketidaksesuaian data, banyaknya jumlah data uji dan akurasi yang didapatkan pada pengujian LVQ yang telah dilakukan. Halaman hasil identifikasi ditunjukkan pada Gambar 5.5.



Gambar 5.5 Implementasi Halaman Hasil Identifikasi

5.4.6 Implementasi halaman identifikasi dengan *inputan user*

Halaman identifikasi dengan *inputan user* merupakan halaman terakhir yang akan ditampilkan saat pengguna menjalankan program penelitian ini. Pada halaman terakhir, tampilan yang diberikan kepada pengguna berupa pilihan *check box* untuk memilih jenis gejala penyakit mata dan *combo box* untuk memilih jenis penyakit mata. Pada halaman ini akan ditampilkan hasil identifikasi berdasarkan pada masukkan pengguna untuk melihat apakah nantinya penyakit yang didapatkan sesuai dengan proses yang akan dikerjakan. Halaman identifikasi dengan *inputan user* ditunjukkan pada Gambar 5.6.



Gambar 5.6 Implementasi Halaman Identifikasi dengan *Inputan User*



BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dibahas mengenai pengujian dan analisis dari implementasi metode LVQ pada identifikasi penyakit mata. Proses pengujian terhadap hasil keluaran dari sistem yang diimplementasikan terdiri atas pengujian *learning rate* (α), pengurangan *learning rate* ($\text{Dec } \alpha$), perbandingan data antara data latih dan data uji, jumlah iterasi maksimum (Max Epoch), minimal *learning rate* (Min α) serta akurasi dan validasi parameter yang optimal. Pengujian *learning rate* (α) sampai dengan pengujian minimal *learning rate* (Min α) merupakan pengujian yang dilakukan untuk menganalisis nilai-nilai parameter yang terbaik dalam metode LVQ. Pengujian akurasi dan validasi parameter optimal merupakan pengujian untuk mencapai tujuan akhir dari analisis yang dilakukan untuk mendapatkan tingkat akurasi yang terbaik dari sistem identifikasi penyakit mata dengan menggunakan metode LVQ pada pengujian parameter sebelumnya.

6.1 Pengujian *learning rate* (α)

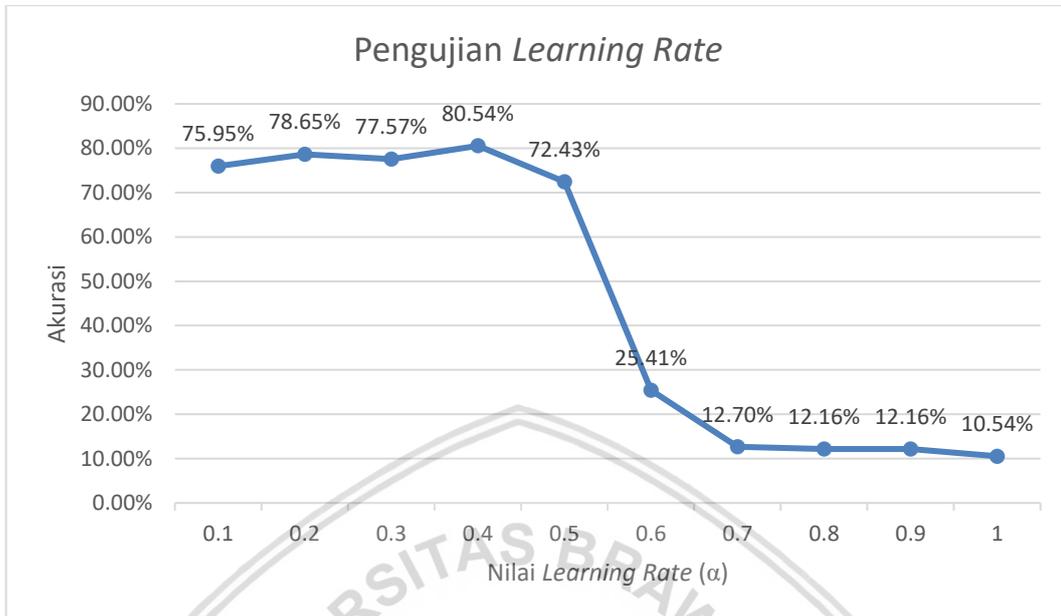
Pengujian *learning rate* (α) dilakukan untuk mendapatkan analisis pada parameter *learning rate* (α) dari hasil perhitungan menggunakan metode *Learning Vector Quantization* (LVQ) untuk identifikasi penyakit mata. Setiap pengujian yang dilakukan untuk menganalisis parameter akan diambil nilainya yang terbaik untuk dipakai pada pengujian parameter berikutnya. Pada pengujian *learning rate* (α), parameter awal yang akan digunakan dalam melakukan analisis ini adalah nilai pengurangan *learning rate* ($\text{Dec } \alpha$) = 0.1, data latih yang digunakan sebesar 70% (88 data latih), jumlah data uji sebesar 30% (37 data uji), iterasi maksimum sebanyak 10 kali dan minimal *learning rate* (Min α) = 0,000000001 (1×10^{-10}). Sedangkan untuk pengujian *learning rate* (α), nilai yang akan digunakan adalah 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 dan 1. Berikut ini adalah Tabel 6.1 yang menunjukkan pengujian pada parameter *learning rate* (α).

Tabel 6.1 Tabel Pengujian *Learning Rate* (α)

α	Skenario Pengujian ke-										Rata-Rata Akurasi
	1	2	3	4	5	6	7	8	9	10	
0.1	83.78%	75.68%	72.97%	75.68%	72.97%	75.68%	75.68%	75.68%	75.68%	75.68%	75.95%
0.2	75.68%	75.68%	78.38%	81.08%	78.38%	83.78%	83.78%	78.38%	75.68%	75.68%	78.65%
0.3	72.97%	78.38%	81.08%	75.68%	78.38%	78.38%	78.38%	75.68%	78.38%	78.38%	77.57%
0.4	81.08%	83.78%	81.08%	81.08%	81.08%	83.78%	78.38%	78.38%	78.38%	78.38%	80.54%
0.5	81.08%	78.38%	78.38%	72.97%	78.38%	83.78%	13.51%	81.08%	72.97%	83.78%	72.43%
0.6	24.32%	78.38%	13.51%	5.41%	13.51%	18.92%	10.81%	13.51%	62.16%	13.51%	25.41%
0.7	8.11%	13.51%	10.81%	13.51%	13.51%	13.51%	13.51%	13.51%	13.51%	13.51%	12.70%
0.8	13.51%	13.51%	5.41%	16.22%	13.51%	8.11%	13.51%	13.51%	13.51%	10.81%	12.16%
0.9	13.51%	13.51%	13.51%	13.51%	13.51%	13.51%	5.41%	13.51%	8.11%	13.51%	12.16%
1	5.41%	8.11%	13.51%	8.11%	24.32%	13.51%	8.11%	13.51%	5.41%	5.41%	10.54%

Berdasarkan pada hasil Tabel 6.1, dari skenario pengujian sebanyak 10 kali didapatkan hasil perolehan rata-rata akurasi terbaik untuk parameter *learning rate* (α) adalah sebesar 80.54% untuk nilai *learning rate* (α) = 0.4. Dari Tabel 6.1

akan dibuat grafik yang akan menggambarkan keseluruhan pengujian pada parameter *learning rate* (α) yang ditampilkan pada Gambar 6.1.



Gambar 6.1 Pengujian *Learning Rate* (α)

Dari Gambar 6.1, dapat disimpulkan bahwa pengujian *learning rate* (α) yang dilakukan menunjukkan grafik yang menurun. Grafik tersebut menggambarkan bahwa semakin besar nilai pengujian *learning rate* (α) yang dikerjakan maka semakin kecil pula nilai akurasi yang didapatkan. Pengujian ini mendapatkan akurasi terbaik sebesar 80.54% untuk nilai *learning rate* 0.4, sedangkan akurasi terendah didapatkan sebesar 10.54% untuk *learning rate* bernilai 1. Hal ini terjadi karena pengujian *learning rate* (α) memberikan dampak kepada *update* bobot yang dilakukan pada metode LVQ sehingga semakin besar nilai *learning rate* (α) yang dikerjakan, maka semakin sedikit proses pembelajaran yang harus dilakukan. Hal ini mengakibatkan akurasi yang didapatkan rendah.

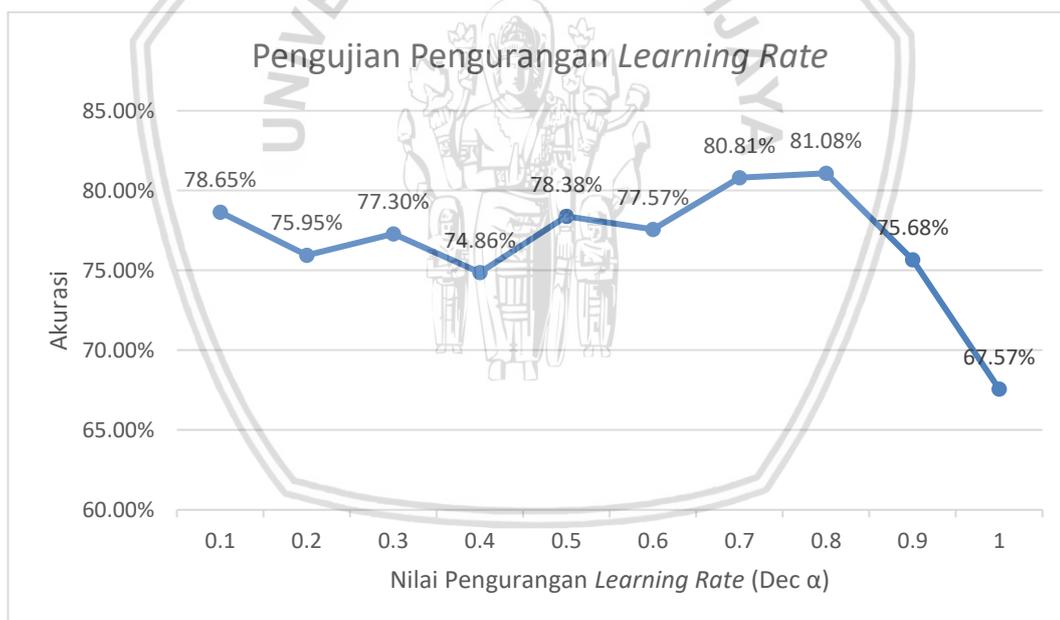
6.2 Pengujian pengurangan *learning rate* (Dec α)

Pengujian pengurangan *learning rate* (Dec α) dilakukan untuk mendapatkan analisis pada parameter pengurangan *learning rate* (Dec α) dari hasil perhitungan menggunakan metode *Learning Vector Quantization* (LVQ) untuk identifikasi penyakit mata. Pada pengujian pengurangan *learning rate* (Dec α), parameter awal yang digunakan sama seperti pada pembahasan di sub-bab 6.1. Sedangkan untuk nilai *learning rate* (α) = 0.4, berdasarkan pada hasil pengujian dengan nilai akurasi paling terbaik. Untuk pengujian ini, parameter pengurangan *learning rate* (Dec α) akan berkisar antara 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, dan 1. Berikut ini adalah Tabel 6.2 yang menunjukkan pengujian pada parameter pengurangan *learning rate* (Dec α).

Tabel 6.2 Tabel Pengujian Pengurangan *Learning Rate* (Dec α)

Dec α	Skenario Pengujian ke-										Rata-Rata Akurasi
	1	2	3	4	5	6	7	8	9	10	
0.1	75.68%	78.38%	81.08%	78.38%	78.38%	78.38%	78.38%	78.38%	81.08%	78.38%	78.65%
0.2	75.68%	75.68%	78.38%	75.68%	72.97%	75.68%	78.38%	75.68%	75.68%	75.68%	75.95%
0.3	78.38%	75.68%	72.97%	78.38%	78.38%	78.38%	78.38%	78.38%	78.38%	75.68%	77.30%
0.4	72.97%	75.68%	75.68%	72.97%	75.68%	72.97%	75.68%	78.38%	75.68%	72.97%	74.86%
0.5	89.19%	72.97%	78.38%	78.38%	78.38%	75.68%	81.08%	75.68%	78.38%	75.68%	78.38%
0.6	83.78%	75.68%	75.68%	78.38%	83.78%	72.97%	70.27%	78.38%	75.68%	81.08%	77.57%
0.7	83.78%	78.38%	75.68%	78.38%	86.49%	81.08%	75.68%	83.78%	78.38%	86.49%	80.81%
0.8	83.78%	75.68%	81.08%	83.78%	81.08%	86.49%	81.08%	78.38%	81.08%	78.38%	81.08%
0.9	81.08%	75.68%	75.68%	78.38%	67.57%	78.38%	67.57%	72.97%	75.68%	83.78%	75.68%
1	78.38%	62.16%	62.16%	62.16%	64.86%	67.57%	64.86%	75.68%	70.27%	67.57%	67.57%

Berdasarkan pada hasil Tabel 6.2, dari skenario pengujian sebanyak 10 kali didapatkan hasil perolehan rata-rata akurasi terbaik untuk parameter pengurangan *learning rate* (Dec α) adalah sebesar 81.08% untuk nilai pengurangan *learning rate* (Dec α) = 0.8. Dari Tabel 6.2 akan dibuat grafik yang akan menggambarkan keseluruhan pengujian yang telah dilakukan pada parameter pengurangan *learning rate* (Dec α) yang digambarkan pada Gambar 6.2.



Gambar 6.2 Pengujian Pengurangan *Learning Rate* (Dec α)

Dari Gambar 6.2, dapat disimpulkan bahwa pengujian pengurangan *learning rate* (Dec α) yang dilakukan menunjukkan grafik untuk akurasi terbaik yang didapatkan adalah sebesar 81.08% untuk nilai 0.8, sedangkan akurasi terendah didapatkan sebesar 67.57% untuk nilai 1. Berdasarkan pada grafik yang ditunjukkan pada Gambar 6.2, dapat diambil kesimpulan bahwa pengaruh parameter pengurangan *learning rate* (Dec α) terhadap metode LVQ cenderung naik-turun. Namun, setelah nilai parameter pengurangan *learning rate* (Dec α) lebih dari 0.8 maka pengujian yang dilakukan hasilnya cenderung menurun.



Sehingga parameter pengurangan *learning rate* (Dec α) bernilai lebih dari 0.8 berdampak kepada nilai akurasi yang dihasilkan. Semakin besar nilai tersebut maka semakin sedikit hasil akurasi yang didapatkan. Hal ini disebabkan oleh karena parameter pengurangan *learning rate* (Dec α) berpengaruh terhadap *update* parameter *learning rate* (α) yang dikerjakan pada saat *update* bobot.

6.3 Pengujian *minimal learning rate* (Min α)

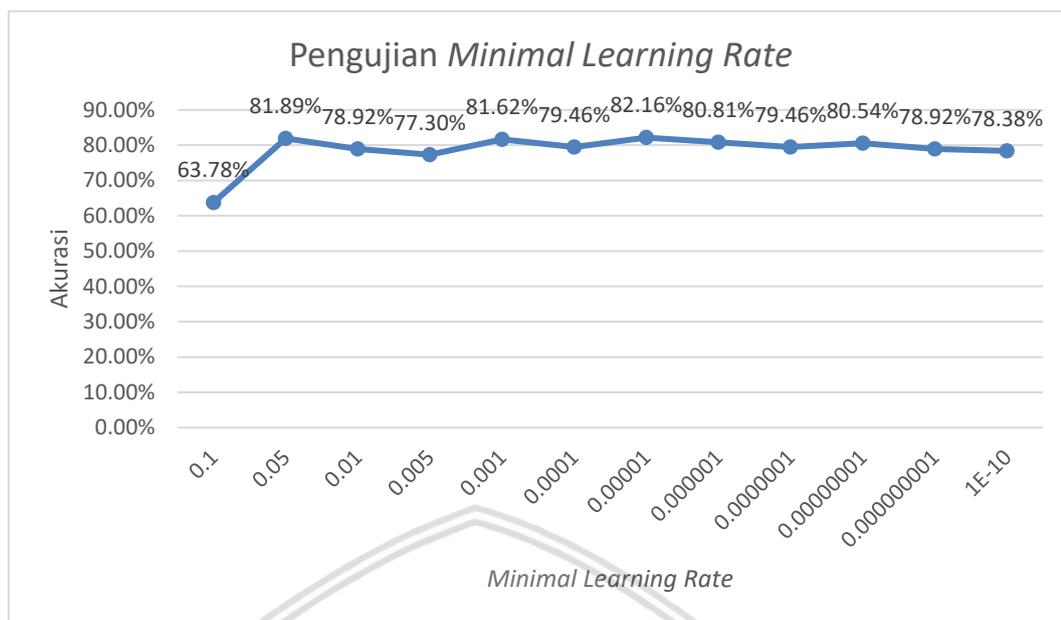
Pengujian *minimal learning rate* (Min α) dilakukan untuk mendapatkan analisis terhadap parameter *minimal learning rate* (Min α) yang memengaruhi akurasi terbaik pada metode LVQ untuk identifikasi penyakit mata. Pada pengujian ini, akan digunakan parameter pendukung untuk mendapatkan akurasi yang optimal dengan parameternya adalah $\alpha = 0.4$ dan Dec $\alpha = 0.8$, sedangkan untuk parameter lainnya sama seperti pada sub-bab 6.1. Untuk pengujian ini, cakupan *minimal learning rate* (Min α) yang akan digunakan adalah 0.1, 0.05, 0.01, 0.005, 0.001, 0.0001, 0.00001, 0.000001, 0.0000001, 0.00000001, dan 0.000000001. Berikut ini adalah hasil pengujian *minimal learning rate* (Min α) yang ditunjukkan ada Tabel 6.3.

Tabel 6.3 Tabel Pengujian *Minimal Learning Rate* (Min α)

Min α	Skenario Pengujian ke-										Rata-Rata Akurasi
	1	2	3	4	5	6	7	8	9	10	
0.1	75.68%	62.16%	51.35%	70.27%	51.35%	67.57%	59.46%	72.97%	64.86%	62.16%	63.78%
0.05	86.49%	75.68%	81.08%	81.08%	67.57%	83.78%	89.19%	83.78%	83.78%	86.49%	81.89%
0.01	81.08%	83.78%	81.08%	72.97%	70.27%	83.78%	83.78%	78.38%	78.38%	75.68%	78.92%
0.005	83.78%	86.49%	62.16%	78.38%	72.97%	81.08%	83.78%	81.08%	62.16%	81.08%	77.30%
0.001	81.08%	81.08%	67.57%	75.68%	81.08%	86.49%	86.49%	86.49%	81.08%	89.19%	81.62%
0.0001	75.68%	83.78%	81.08%	78.38%	81.08%	78.38%	75.68%	78.38%	75.68%	86.49%	79.46%
0.00001	81.08%	83.78%	83.78%	83.78%	83.78%	83.78%	81.08%	83.78%	81.08%	75.68%	82.16%
0.000001	81.08%	78.38%	89.19%	86.49%	81.08%	83.78%	70.27%	81.08%	81.08%	75.68%	80.81%
0.0000001	83.78%	75.68%	78.38%	81.08%	78.38%	83.78%	75.68%	78.38%	81.08%	78.38%	79.46%
0.00000001	75.68%	81.08%	81.08%	83.78%	78.38%	86.49%	72.97%	83.78%	78.38%	83.78%	80.54%
0.000000001	83.78%	86.49%	81.08%	78.38%	81.08%	64.86%	78.38%	75.68%	78.38%	81.08%	78.92%
1E-10	83.78%	72.97%	67.57%	70.27%	78.38%	86.49%	75.68%	83.78%	81.08%	83.78%	78.38%

Berdasarkan pada hasil Tabel 6.3, dari skenario pengujian sebanyak 10 kali untuk parameter *minimal learning rate* (Min α) yang telah dilakukan maka bisa diambil kesimpulan bahwa nilai rata-rata akurasi tertinggi diperoleh sebesar 82.16% untuk nilai *minimal learning rate* (Min α) = 0.00001. Berikut ini akan digambarkan grafik terkait hasil pengujian yang telah dilakukan pada Gambar 6.3 untuk memberikan tampilan secara keseluruhan mengenai pengujian parameter *minimal learning rate* (Min α).





Gambar 6.3 Pengujian *Minimal Learning Rate* (Min α)

Dari Gambar 6.3, dapat disimpulkan bahwa pengujian *minimal learning rate* (Min α) yang dilakukan menunjukkan pada nilai Min $\alpha = 0.00001$ diperoleh rata-rata akurasi terbaik sebesar 82.16% sedangkan untuk nilai Min $\alpha = 0.1$ diperoleh rata-rata akurasi terendah sebesar 63.78%. Dari Gambar 6.3 tersebut, grafik yang terlihat menunjukkan bahwa pengaruh parameter *minimal learning rate* (Min) α cenderung stabil, meskipun naik-turun. Kisaran pengujian yang cenderung stabil didapatkan mulai dari nilai akurasi sebesar 77% sampai dengan nilai akurasi sebesar 81%. Parameter *minimal learning rate* (Min α) sendiri berperan untuk menjalankan kondisi berhenti pada proses pelatihan LVQ, sehingga semakin kecil nilai parameter tersebut maka proses *update* bobot yang terjadi pada pelatihan LVQ akan semakin lama dikerjakan dan mengakibatkan proses pembelajaran semakin banyak.

6.4 Pengujian jumlah iterasi maksimum (Max Epoch)

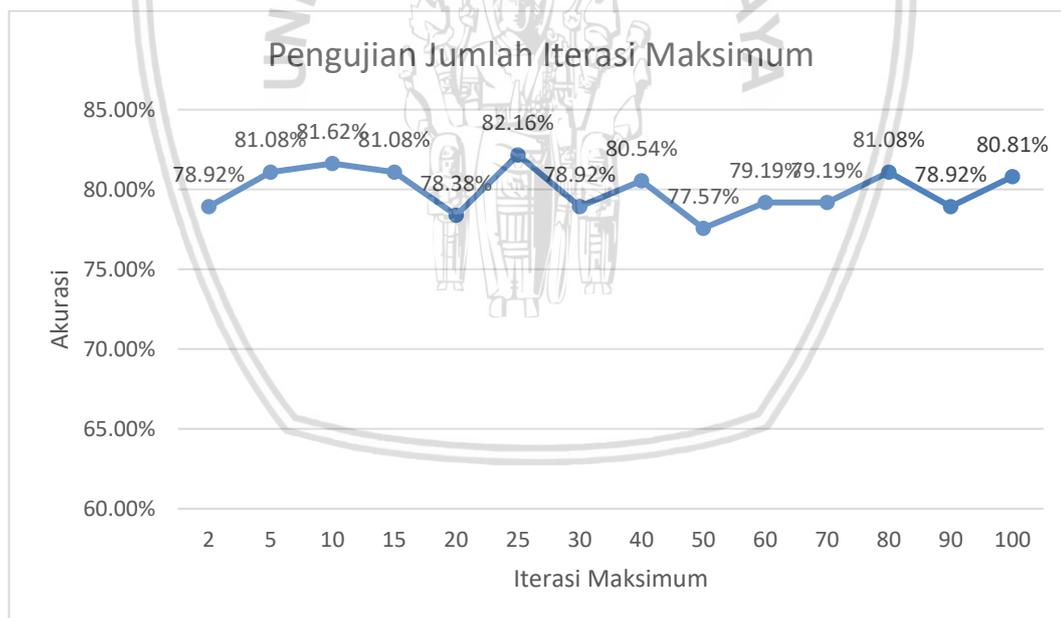
Pengujian jumlah iterasi maksimum dilakukan untuk mendapatkan analisis terhadap parameter jumlah iterasi yang memengaruhi akurasi terbaik pada metode LVQ. Untuk pengujian ini fungsinya adalah mendapatkan akurasi yang paling optimal untuk digunakan pada identifikasi penyakit mata. Pada pengujian ini, parameter awal masih menggunakan parameter yang telah dijelaskan pada sub-bab 6.1, sedangkan untuk parameter yang telah dilakukan pengujian seperti parameter *learning rate* (α), pengurangan *learning rate* (Dec α) dan *minimal learning rate* (Min α) akan diambil nilainya dari akurasi terbaik yang sudah didapatkan. Untuk nilai $\alpha = 0.4$, Dec $\alpha = 0.8$ dan min $\alpha = 0.00001$, sedangkan varian iterasi maksimum yang akan diujikan mencakup 2, 5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, dan 100. Berikut ini adalah hasil pengujian pada iterasi maksimum yang ditunjukkan pada Tabel 6.4.



Tabel 6.4 Tabel Pengujian Jumlah Iterasi Maksimum (Max Epoch)

Max Epoch	Skenario Pengujian ke-										Rata-Rata Akurasi
	1	2	3	4	5	6	7	8	9	10	
2	75.68%	64.86%	78.38%	86.49%	75.68%	89.19%	83.78%	83.78%	72.97%	78.38%	78.92%
5	86.49%	83.78%	83.78%	89.19%	81.08%	86.49%	70.27%	75.68%	70.27%	83.78%	81.08%
10	91.89%	86.49%	81.08%	78.38%	81.08%	78.38%	78.38%	75.68%	86.49%	78.38%	81.62%
15	86.49%	81.08%	78.38%	75.68%	72.97%	78.38%	86.49%	86.49%	81.08%	83.78%	81.08%
20	81.08%	72.97%	86.49%	81.08%	75.68%	78.38%	75.68%	81.08%	72.97%	78.38%	78.38%
25	78.38%	81.08%	83.78%	83.78%	78.38%	81.08%	78.38%	83.78%	81.08%	91.89%	82.16%
30	75.68%	75.68%	81.08%	72.97%	86.49%	86.49%	81.08%	81.08%	75.68%	81.08%	78.92%
40	78.38%	83.78%	86.49%	75.68%	72.97%	81.08%	81.08%	83.78%	83.78%	78.38%	80.54%
50	75.68%	86.49%	78.38%	72.97%	81.08%	75.68%	86.49%	59.46%	78.38%	81.08%	77.57%
60	81.08%	75.68%	78.38%	83.78%	83.78%	83.78%	70.27%	78.38%	72.97%	83.78%	79.19%
70	81.08%	83.78%	72.97%	70.27%	75.68%	81.08%	83.78%	75.68%	89.19%	78.38%	79.19%
80	81.08%	86.49%	78.38%	81.08%	81.08%	70.27%	83.78%	78.38%	86.49%	83.78%	81.08%
90	78.38%	75.68%	83.78%	78.38%	81.08%	70.27%	86.49%	72.97%	81.08%	81.08%	78.92%
100	81.08%	83.78%	83.78%	75.68%	75.68%	78.38%	78.38%	83.78%	83.78%	83.78%	80.81%

Berdasarkan pada hasil Tabel 6.4, dari skenario pengujian sebanyak 10 kali yang telah dilakukan maka diperoleh hasil rata-rata akurasi terbaik untuk iterasi maksimum sebanyak 25 kali iterasi dengan nilai akurasi sebesar 82.16%. Dari Tabel 6.4 tersebut, akan dibuat kedalam bentuk grafik untuk melihat keseluruhan pengujian yang telah dilakukan terhadap iterasi maksimum. Grafik tersebut seperti terlihat pada Gambar 6.4.



Gambar 6.4 Pengujian Jumlah Iterasi Maksimum (Max Epoch)

Dari Gambar 6.4, dapat disimpulkan bahwa pengujian terhadap iterasi maksimum yang dilakukan menunjukkan bahwa pada iterasi ke-25 didapatkan rata-rata akurasi sebesar 82.16% sedangkan untuk iterasi ke-50 didapatkan rata-rata akurasi sebesar 77.57%. Pada Gambar 6.4, menunjukkan bahwa grafik pada pengujian iterasi maksimum tidak cenderung naik ataupun turun (tidak berpola). Hal ini disebabkan oleh ragam nilai yang dipengaruhi oleh bobot awal yang acak sehingga berpengaruh terhadap iterasi. Selain itu, jumlah iterasi maksimum



sendiri berpengaruh terhadap proses pelatihan pada metode LVQ, sama halnya dengan parameter *learning rate* (α) sehingga semakin besar jumlah iterasi yang didapatkan maka proses *update* bobot yang dikerjakan akan semakin lama.

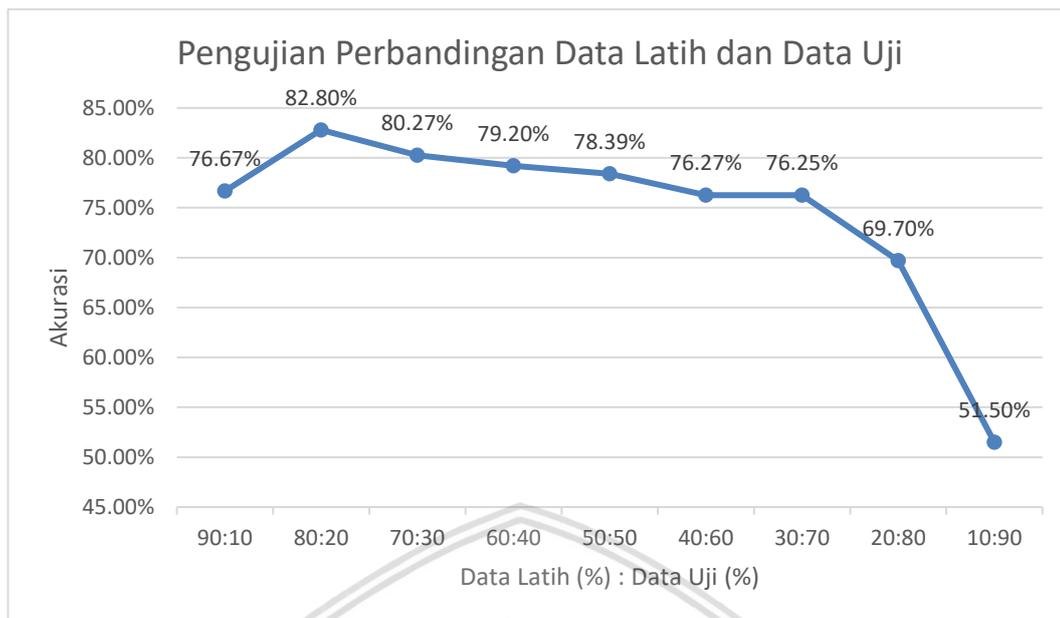
6.5 Pengujian antara perbandingan data latih dan data uji

Pengujian antara perbandingan data latih dan data uji dilakukan untuk mendapatkan analisis terhadap parameter yang memengaruhi akurasi terbaik atau yang paling ideal pada metode LVQ. Pengujian ini berguna untuk membandingkan hasil terbaik yang didapatkan baik pada data latih maupun data uji sehingga pada pengujian akurasi dan validasi parameter optimal nantinya, data latih dan data uji yang digunakan pun bisa memberikan hasil yang optimal. Pada pengujian ini, parameter yang akan digunakan adalah $\alpha = 0.4$, Dec $\alpha = 0.8$, Min $\alpha = 0.00001$ dan Max Epoch = 25 iterasi sedangkan untuk pengujian data latih dan data uji, akan dibagi datanya dengan cakupan 90:10, 80:20, 70:30, 60:40, 50:50, 40:60, 30:70, 20:80 dan 10:90. Berikut ini adalah Tabel 6.5 yang menunjukkan pengujian pada perbandingan data latih dan data uji.

Tabel 6.5 Tabel Pengujian Perbandingan Data Latih dan Data Uji

Data Latih:Data Uji	Skenario Pengujian ke-										Rata-Rata Akurasi
	1	2	3	4	5	6	7	8	9	10	
90:10	75.00%	75.00%	83.33%	83.33%	75.00%	75.00%	75.00%	75.00%	75.00%	75.00%	76.67%
80:20	80.00%	84.00%	80.00%	80.00%	92.00%	88.00%	76.00%	88.00%	76.00%	84.00%	82.80%
70:30	81.08%	78.38%	81.08%	83.78%	86.49%	78.38%	81.08%	81.08%	75.68%	75.68%	80.27%
60:40	80.00%	78.00%	78.00%	80.00%	82.00%	76.00%	78.00%	80.00%	82.00%	78.00%	79.20%
50:50	77.42%	83.87%	82.26%	79.03%	85.48%	75.81%	70.97%	79.03%	80.65%	69.35%	78.39%
40:60	82.67%	70.67%	77.33%	73.33%	78.67%	76.00%	72.00%	82.67%	73.33%	76.00%	76.27%
30:70	79.55%	76.14%	77.27%	79.55%	73.86%	73.86%	72.73%	77.27%	79.55%	72.73%	76.25%
20:80	77.00%	75.00%	66.00%	78.00%	70.00%	58.00%	75.00%	74.00%	62.00%	62.00%	69.70%
10:90	59.29%	53.98%	53.98%	55.75%	45.13%	45.13%	45.13%	45.13%	55.75%	55.75%	51.50%

Berdasarkan pada hasil Tabel 6.5, dari skenario pengujian sebanyak 10 kali didapatkan hasil rata-rata akurasi terbaik untuk perbandingan jumlah data latih dan data uji adalah 82.80% untuk data latih sebesar 80% (100 data latih) dan data uji sebesar 20% (25 data uji). Dari Tabel 6.5 akan dibuat grafik untuk menggambarkan keseluruhan pengujian terhadap perbandingan data latih dan data uji yang ditunjukkan pada Gambar 6.5.



Gambar 6.5 Pengujian Perbandingan Data Latih dan Data Uji

Dari Gambar 6.5, dapat disimpulkan bahwa pengujian terhadap pengaruh perbandingan data latih dan data uji yang dilakukan menunjukkan grafik dengan akurasi terbaik yang didapatkan adalah sebesar 82.80% untuk data latih sebesar 80% (100 data latih) dan data uji sebesar 20% (25 data uji). Sedangkan akurasi terendah didapatkan sebesar 51.50% untuk data latih sebesar 10% (12 data latih) dan data uji sebesar 90% (113 data uji). Berdasarkan pada grafik tersebut, semakin besar jumlah data latih yang digunakan maka hasil akurasi yang didapatkan juga semakin baik. Namun, pada kondisi perbandingan jumlah data 90:10, hasil akurasi yang didapatkan lebih sedikit padahal jumlah data latih yang digunakan lebih banyak dibandingkan jumlah data 80:20. Hal ini disebabkan oleh adanya *overfitting* sehingga memengaruhi kurangnya akurasi yang diperoleh, sebab pada saat *overfitting* akan menghasilkan *output* yang baik untuk data *training* tetapi tidak menghasilkan *output* yang selalu baik untuk data *testing*.

Perbandingan jumlah data latih dan data uji juga berpengaruh terhadap proses pembelajaran bagi metode LVQ. Semakin banyak data latih yang digunakan maka data yang didapatkan pun semakin bervariasi sehingga proses LVQ yang dikerjakan semakin menghasilkan akurasi yang lebih baik. Dari pengujian tersebut, didapatkan kesimpulan bahwa parameter yang optimal adalah $\alpha = 0.4$, $\text{Dec } \alpha = 0.8$, $\text{Min } \alpha = 0.00001$, dan $\text{Max Epoch} = 25$. Dari parameter tersebut, kemudian pengujian jumlah data latih dan data uji dilakukan sehingga mendapatkan 80:20 sebagai data latih dan data uji yang terbaik. Dari data tersebut kemudian diambil akurasi rata-rata, akurasi terbaik dan akurasi terendah yang ditunjukkan pada Tabel 6.6.

Tabel 6.6 Tabel Pengujian Akurasi

No	Pengujian ke-	Skenario Pengujian		
		Sesuai	Tdk Sesuai	Akurasi
1	1	20	5	80.00%
2	2	21	4	84.00%
3	3	20	5	80.00%
4	4	20	5	80.00%
5	5	23	2	92.00%
6	6	22	3	88.00%
7	7	19	6	76.00%
8	8	22	3	88.00%
9	9	19	6	76.00%
10	10	21	4	84.00%
AVERAGE				82.80%
MAX				92.00%
MIN				76.00%

Dari Tabel 6.6 dapat diketahui bahwa rata-rata akurasi yang didapatkan dari skenario pengujian yang dilakukan sebanyak 10 kali adalah sebesar 82.80%. Sedangkan untuk akurasi terbaik adalah 92.00% dan untuk akurasi terendah adalah 76.00%. Perhitungan akurasi tersebut didapatkan atas acuan persamaan (4.1) yang dijabarkan sebagai berikut.

$$Akurasi = \frac{\text{jumlah data yang benar}}{\text{total semua data yang ada}} \times 100\%$$

$$Akurasi = \frac{23}{25} \times 100\%$$

$$Akurasi = 92\%$$

Sehingga dari perhitungan tersebut, dapat diketahui bahwa 23 data didapatkan dari data yang benar (sesuai dengan data yang asli) sedangkan 25 data didapatkan dari keseluruhan data uji yang digunakan. Begitupun dengan hasil akurasi lainnya yang tetap menggunakan persamaan (4.1) sebagai acuan dalam mendapatkan nilai akurasi. Melalui keseluruhan pengujian yang telah dilakukan pada identifikasi penyakit mata dengan menggunakan metode LVQ, maka dapat disimpulkan bahwa sistem dapat berfungsi dengan baik. Sehingga dari keseluruhan pengujian yang telah dilakukan, sistem dapat melakukan identifikasi penyakit mata sesuai dengan gejala dan penyakit yang telah didapatkan.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan pada keseluruhan tahap penelitian mulai dari perancangan, implementasi dan pengujian pada sistem identifikasi penyakit mata dengan menggunakan metode LVQ, maka didapatkan kesimpulan sebagai berikut.

1. Metode LVQ dapat diterapkan ke dalam sistem identifikasi penyakit mata. Dataset penyakit mata yang telah terkumpul akan dipecah keseluruhan data secara acak kedalam bobot awal, data latih dan data uji. Setelah memecah seluruh dataset, dilakukan proses pelatihan LVQ dengan menghitung nilai jarak pada tiap data latih dengan data bobot. Setiap perhitungan tersebut, akan dilakukan *update* data bobot sehingga mendapatkan bobot akhir yang akan digunakan dalam proses pengujian LVQ. Bobot akhir tersebut kemudian menjadi penunjang dalam pengujian dan pengklasifikasian LVQ.
2. Berdasarkan pada skenario pengujian sebanyak 10 kali, dengan parameter optimal yang didapatkan yaitu $\alpha = 0,4$, $\text{Dec } \alpha = 0,8$, $\text{Min } \alpha = 0,00001$, dan $\text{Max Epoch} = 25$ serta perbandingan jumlah data latih dan data uji yang digunakan yaitu 80:20 untuk 100 data latih dan 25 data, maka diperoleh rata-rata akurasi sebesar 82,80% dan akurasi tertinggi sebesar 92%.

7.2 Saran

Saran yang dapat diberikan bagi pengembangan penelitian selanjutnya terkait dengan masalah sistem identifikasi penyakit mata dengan menggunakan metode LVQ antara lain:

1. Untuk penelitian selanjutnya, dapat dilakukan optimasi terhadap metode LVQ dengan menggunakan metode lain. Dengan harapan bahwa metode LVQ dapat dioptimasi dan meningkatkan rata-rata akurasi yang lebih baik lagi untuk objek yang akan diteliti.
2. Penelitian lebih lanjut dapat dilakukan dengan mengembangkan metode LVQ menjadi metode LVQ 2, LVQ 2.1 dan metode LVQ 3. Diharapkan dengan menggunakan tingkatan metode LVQ yang lebih, dapat mengurangi nilai yang tidak sesuai dan mendapatkan hasil rata-rata akurasi yang lebih baik.
3. Melakukan penelitian lebih lanjut terkait dengan masalah penyakit mata. Tidak menutup kemungkinan bahwa semakin banyaknya varian data yang diperoleh, baik untuk gejala dan penyakit mata maka akan memberikan pengaruh terhadap metode yang akan digunakan untuk mengetahui apakah sistem dapat melakukan identifikasi atau tidak.

DAFTAR PUSTAKA

- Benelli, U. (2009). Picture of Cataracts. Retrieved March 15, 2018, from https://www.medicinenet.com/image-collection/cataracts_picture/picture.htm
- Dahl, A. (2009a). Picture of Bacterial Conjunctivitis (Pink Eye). Retrieved March 15, 2018, from https://www.medicinenet.com/image-collection/bacterial_conjunctivitis_pink_eye_picture/picture.htm
- Dahl, A. (2009b). Picture of Blepharitis. Retrieved March 16, 2018, from https://www.medicinenet.com/image-collection/blepharitis_picture/picture.htm
- Dahl, A. (2009c). Picture of Chalazion (Eyelid Cyst). Retrieved March 16, 2018, from https://www.medicinenet.com/image-collection/chalazion_eyelid_cyst_picture/picture.htm
- Dahl, A. (2009d). Picture of Glaucoma. Retrieved March 15, 2018, from https://www.medicinenet.com/image-collection/glaucoma_picture/picture.htm
- Dahl, A. (2009e). Picture of Uveitis. Retrieved March 15, 2018, from https://www.medicinenet.com/image-collection/uveitis_picture/picture.htm
- Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA ©1994.
- Ganidar, F. R., Putri, R. R. M., & Dewi, C. (2015). Pengklasifikasian Mutu Susu Sapi Menggunakan Metode Learning Vector Quantization (LVQ) (Studi Kasus: Upt Laboratorium Kesehatan Hewan Malang). *PTIHK DORO - Repository Jurnal Mahasiswa*, 6(14).
- Greenacre, M., & Primicerio, R. (2013). *Measures of distance between samples: Euclidean. Multivariate Analysis of Ecological Data*. Fundación BBVA.
- Ilyas, S. (1998). *Ilmu Penyakit Mata*. Jakarta: Balai Penerbit FK-UI. Retrieved from <http://digilib.ub.ac.id/inlislite3/opac/detail-opac?id=30321>
- Ilyas, S. (2003). *Penuntun Ilmu Penyakit Mata (2, Rev ed.)*. Jakarta: Balai Penerbit FK-UI. Retrieved from <http://digilib.ub.ac.id/inlislite3/opac/detail-opac?id=2262>
- Islamiah, N., Cholissodin, I., & Dewi, C. (2016). Klasifikasi Penyakit Gigi Dan Mulut Menggunakan Metode Learning Vector Quantization (LVQ). *PTIHK DORO - Repository Jurnal Mahasiswa*, 7(28).
- Javadi, M.-A., & Feizi, S. (2011). Dry Eye Syndrome. *Journal of Ophthalmic and Vision Research*, 6(3), 192–198. Retrieved from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3306104/>

- Krisnanti, G. W., Dewi, C., & Cholissodin, I. (2015). Klasifikasi Kualitas Daun Nilam (Pogostemon Cablin Benth) Menggunakan Citra Digital Dan Algoritma Learning Vector Quantization. *PTIIK DORO - Repository Jurnal Mahasiswa*, 6(8).
- Lestari, P., Hidayat, N., & Putri, R. R. M. (2016). Pemodelan Sistem Pakar Diagnosa Penyakit Mata Menggunakan Metode Naive Bayes - Certainty Factor. *PTIIK DORO - Repository Jurnal Mahasiswa*, 7(7).
- Lindsley, K., Matsumura, S., Hatef, E., & Akpek, E. (2014). Interventions for Chronic Blepharitis. *US National Library of Medicine National Institutes of Health*. Retrieved from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4270370/>
- Martinuva, E. D., Ratnawati, D. E., & Indriati. (2015). Implementasi Learning Vector Quantization (LVQ) Untuk Pemilihan Keminatan (Studi Kasus: Program Studi Informatika Universitas Brawijaya). *PTIIK DORO - Repository Jurnal Mahasiswa*, 6(25).
- Nugroho, P. A., Sulisty, M. E., & Saptono, R. (2016). Perbandingan Metode Probabilistik Naive Bayesian Classifier dan Jaringan Syaraf Tiruan Learning Vector Quantization dalam Kasus Klasifikasi Penyakit Kandungan. *Jurnal Teknologi & Informasi ITSmart*, 2(2), 21. <https://doi.org/10.20961/its.v2i2.628>
- Perdami. (2013). *Gangguan, Situasi Global, Penglihatan*. Departemen Kesehatan RI.
- Prabowo, W. S., Dewi, C., & Suprpto. (2015). Pengenalan Bahasa Isyarat Menggunakan Citra Digital Dengan Algoritma Learning Vector Quantization. *PTIIK DORO - Repository Jurnal Mahasiswa*, 5(13).
- Reztaputra, R. (2012). Anatomi Mata. *Www.Medicinesia.Com*. Retrieved from <http://www.medicinesia.com/kedokteran-dasar/penginderaan-kedokteran-dasar/anatomimata/>
- Rose, J. (2017). Five Natural Remedies for Dry Eyes. Retrieved March 15, 2018, from <https://gazettereview.com/2015/08/five-natural-remedies-for-dry-eyes/>
- Umam, M. S., Dewi, C., & Cholissodin, I. (2015). Implementasi Metode Learning Vector Quantization (LVQ) Untuk Identifikasi Penyakit Pada Citra Daun Tanaman Kedelai. *PTIIK DORO - Repository Jurnal Mahasiswa*, 6(6).
- Upadhyay, M., Srinivasan, M., & Witcher, J. (2015). Diagnosing and Managing Microbial Keratitis. *Community Eye Health Journal*, 28(89), 03–06. Retrieved from <https://www.cehjournal.org/article/diagnosing-and-managing-microbial-keratitis/>
- Vislisel, J. (2015). Pterygium. Retrieved March 15, 2018, from <https://webeye.ophth.uiowa.edu/eyeforum/atlas/pages/Pterygium/index.htm>

Wijayanti, D. M. P., Cholissodin, I., & Murlistyarini, S. (2016). Identifikasi Diagnosis Perubahan Hasil Perawatan Kulit Menggunakan Metode Learning Vector Quantization (LVQ). *PTIIK DORO - Repository Jurnal Mahasiswa*, 8(24).

