

**KLASIFIKASI KUALITAS DAUN NILAM (*POGOSTEMON
CABLIN BENTH*) MENGGUNAKAN CITRA DIGITAL DAN
ALGORITMA *LEARNING VECTOR QUANTIZATION***

SKRIPSI

Untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh:

GREGORIA WILLA KRISNANTI

115090601111015

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER

MALANG

2015

LEMBAR PERSETUJUAN

KLASIFIKASI KUALITAS DAUN NILAM (*POGOSTEMON CABLIN BENTH*) MENGGUNAKAN CITRA DIGITAL DAN ALGORITMA *LEARNING VECTOR QUANTIZATION*

SKRIPSI



Disusun oleh :

GREGORIA WILLA KRISNANTI

NIM. 115090601111015

Skripsi ini telah disetujui oleh:

Dosen Pembimbing I,

Candra Dewi, S.Kom., MSc.

NIP. 19771114 200312 2 001

Dosen Pembimbing II,

Imam Cholissodin, S.Si., M.Kom

NIK. 850719 16 1 1 0422

LEMBAR PENGESAHAN

KLASIFIKASI KUALITAS DAUN NILAM (*POGOSTEMON CABLIN BENTH*) MENGGUNAKAN CITRA DIGITAL DAN ALGORITMA *LEARNING VECTOR QUANTIZATION*

SKRIPSI

Disusun oleh :

GREGORIA WILLA KRISNANTI

NIM. 115090601111015

Skripsi ini telah diuji dan dinyatakan lulus

26 Juni 2015

Dosen Penguji I,

Dosen Penguji II,

Wayan F. Mahmudy, S.Si, M.T, Ph.D Rekyan Regasari M. P., S.T, M.T

NIP. 19720919 199702 1 001

NIK. 77041406120253

Dosen Penguji III,

M. Ali Fauzi, S.Kom, M.Kom

NIK. -

Mengetahui,
Ketua Program Studi Teknik Informatika / Ilmu Komputer

Drs. Marji, MT

NIP. 19670801 199203 1001

LEMBAR PERNYATAAN

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata dalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan perturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 26 Juni 2015

Gregoria Willa Krisnanti

115090601111015

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Tuhan yang Maha Esa, yang telah melimpahkan berkat dan anugerah-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi ini yang berjudul “KLASIFIKASI KUALITAS DAUN NILAM (*POGOSTEMON CABLIN BENTH*) MENGGUNAKAN CITRA DIGITAL DAN ALGORITMA *LEARNING VECTOR QUANTIZATION*”. Pada penyusunan Skripsi ini tidak semata-mata hasil kerja penulis sendiri, melainkan juga berkat dan bimbingan dan dorongan dari pihak-pihak yang telah membantu, baik secara materi maupun non materi. Maka dari itu penulis ingin mengucapkan banyak terima kasih yang tak terhingga serta penghargaan yang setinggi-tingginya kepada orang-orang yang telah membantu penulis secara langsung maupun tidak langsung kepada yang terhormat:

1. Candra Dewi, S.Kom., MSc. dan Imam Cholissodin, S.Si.,M.Kom selaku dosen pembimbing skripsi yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Drs. Marji, M.T selaku Ketua Prodi Informatika / Ilmu Komputer yang telah memberikan kesempatan kepada penulis untuk menyelesaikan skripsi ini.
3. Segenap bapak dan ibu dosen program studi Informatika / Ilmu Komputer beserta seluruh staff administrasi yang telah membantu selama perkuliahan.
4. Kedua orang tua penulis Cornelius Widiarto dan Lily Margaretha, serta saudara penulis yang telah memberikan dukungan, doa, nasehat dan kasih sayang yang tak terhingga.
5. Teman – teman Ilmu Komputer angkatan 2011 yang telah memberikan masukan dan inspirasi kepada penulis selama menempuh studi dan menyelesaikan skripsi ini.
6. Yulius Christanto A. W. yang selalu memberikan semangat dalam pengerjaan Skripsi ini.
7. Dan semua pihak yang telah membantu dalam penyelesaian skripsi ini.

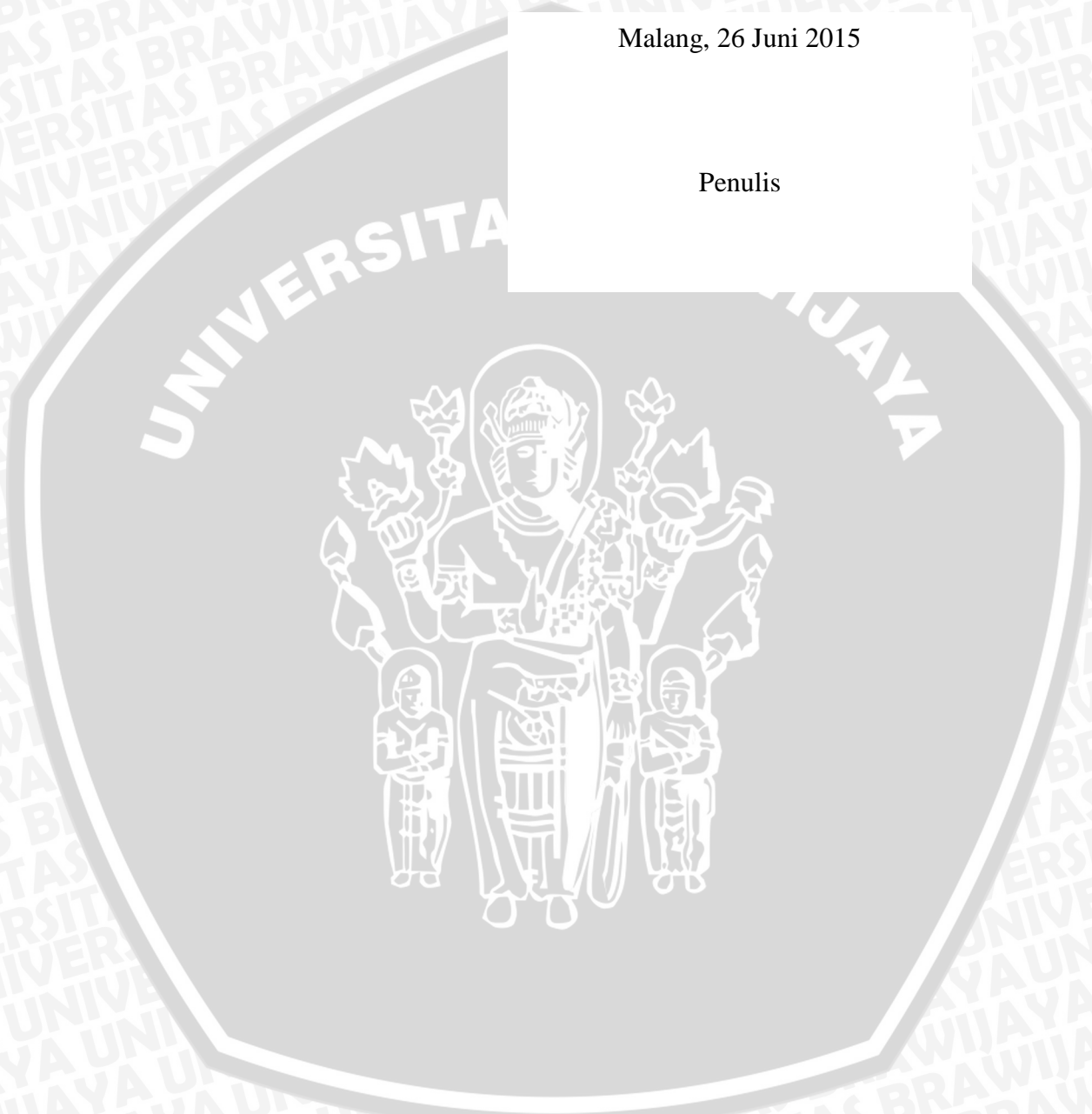
Penulis menyadari bahwa penulisan skripsi ini jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan masukan berupa saran dan kritik dari semua

repository.ub.ac.id

pihak demi tercapainya kesempurnaan dalam skripsi ini. Akhir kata semoga penulisan skripsi ini dapat memberikan manfaat bagi semua pihak.

Malang, 26 Juni 2015

Penulis



ABSTRAK

Daun nilam merupakan bagian tanaman nilam yang sangat bermanfaat bagi kehidupan karena minyak yang dapat dihasilkannya. Pengetahuan akan kualitas daun nilam sangatlah penting bagi petani nilam karena dari kualitas daun nilam inilah petani bisa mengetahui kualitas minyak yang dapat dihasilkan. Untuk mempermudah proses identifikasi kualitas daun nilam, maka diperlukan suatu sistem yang otomatis dapat melakukan klasifikasi kualitas daun nilam. Sistem yang dibangun menggunakan teknologi Jaringan Syaraf Tiruan (JST) *Learning Vector Quantization* (LVQ) dengan ekstraksi fitur warna dan luas daun. Fitur warna yang digunakan yaitu rata-rata dari setiap warna merah, hijau, dan biru. Sistem bekerja dengan melakukan beberapa tahapan meliputi input citra daun nilam, ekstraksi fitur warna, binerisasi, ekstraksi fitur luas, pelatihan dengan algoritma LVQ, pengujian dengan algoritma LVQ yang akan menghasilkan keluaran berupa kelas “kualitas baik” atau “kualitas kurang baik”. Hasil rata-rata akurasi terbaik yang didapatkan melalui pengujian sistem adalah 98% dari 10 data uji dengan menggunakan data latih sebanyak 90, laju pembelajaran=0.1, pengurang laju pembelajaran=0.3, laju pembelajaran minimum= 10^{-3} , dan jumlah iterasi=4.

Kata kunci: Kualitas Daun Nilam, Citra Digital, *Learning Vector Quantization*

ABSTRACT

Patchouli leaf is one part of patchouli plant that is very useful for living because of its oil. Information about the quality of patchouli leaf is important for farmers because from the quality of patchouli leaf, the farmers will know the quality of oil that can be produced from leaf. To simplify the process of patchouli leaf's quality identification, then we need an automatic system that can do the quality clasification of patchouli leaf. The system will be build with Learning Vector Quantization (LVQ) of Artificial Neural Network (ANN) technology with extraction of colour and leaf's surface area feature. Colour feature that will be used is the average of each colour, red, green and blue. The system works by performing several steps, including: input image of patchouli leaf, colour feature extraction, surface area feature extraction, training with LVQ algorithm, testing with LVQ algorithm and the result is a class with "good quality" or "poor quality". The result with best average of accuration obtained by system testing is 98% of 10 testing data using 90 training data, learning rate = 0.1, deduction of learning rate = 0.3, minimum of learning rate = 10^{-3} and the number of iteration = 4.

Keyword: *Quality Of Patchouli Leaf, Digital Image, Learning Vector Quantization*

DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
KATA PENGANTAR.....	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiv
DAFTAR SOURCE CODE	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	4
BAB II KAJIAN PUSTAKA DAN DASAR TEORI.....	6
2.1 Kajian Pustaka	6
2.2 Tanaman Nilam	11
2.3 Citra Digital	12
2.3.1 Citra Biner.....	13
2.3.2 Citra <i>Grayscale</i>	13
2.3.3 Citra Warna.....	14
2.4 Ekstraksi Fitur.....	15
2.4.1 Fitur Warna	15
2.4.2 Fitur Luas	15
2.5 Jarak <i>Euclidean</i>	16
2.6 Jaringan Syaraf Tiruan (JST).....	16
2.6.1 Karakteristik Jaringan Syaraf Tiruan	17

2.6.2	Arsitektur Jaringan Syaraf Tiruan.....	18
2.6.3	Jaringan dengan lapisan tunggal (<i>single layer net</i>).....	18
2.6.4	Jaringan dengan lapisan kompetitif (<i>competitive layer net</i>)	19
2.6.5	Prinsip Dasar Pelatihan/Pembelajaran Jaringan Syaraf Tiruan	20
2.6.6	Proses Algoritma Pelatihan/Pembelajaran Jaringan Syaraf Tiruan.....	20
2.6.7	Belajar dengan Pengawasan (<i>Supervised Learning</i>).....	20
2.6.8	Algoritma Jaringan Syaraf Tiruan.....	21
2.7	<i>Learning Vector Quantization (LVQ)</i>	21
2.7.1	Arsitektur Jaringan <i>Learning Vector Quantization</i>	22
2.7.2	Algoritma <i>Learning Vector Quantization</i>	22
2.8	Evaluasi.....	23
BAB III METODE PENELITIAN DAN PERANCANGAN		24
3.1	Metode Penelitian	24
3.1.1	Studi Literatur	24
3.1.2	Pengumpulan Data	25
3.1.3	Analisis Kebutuhan	26
3.1.4	Perancangan Sistem	26
3.1.4.1	Deskripsi Umum Sistem	27
3.1.4.2	Diagram Blok Sistem	27
3.1.4.3	Arsitektur Sistem.....	28
3.1.5	Implementasi Sistem.....	28
3.1.6	Pengujian dan Analisis.....	28
3.1.7	Kesimpulan dan Saran	29
3.2	Perancangan.....	29
3.2.1	Proses Kerja Sistem Secara Umum.....	29
3.2.2	Proses Ekstraksi Fitur Warna.....	30
3.2.3	Proses Binerisasi	32
3.2.4	Proses Ekstraksi Fitur Luas.....	33
3.2.5	Proses Pelatihan dengan Algoritma LVQ.....	34
3.2.6	Proses Pengujian dengan LVQ.....	35

3.3 Perhitungan Manual.....	36
3.3.1 Ekstraksi Nilai RGB.....	37
3.3.2 Ekstraksi Nilai Rata-Rata Red, Green, dan Blue	38
3.3.3 Binerisasi Citra Daun Nilam	39
3.3.4 Ekstraksi Nilai Luas Permukaan	40
3.3.5 Pelatihan dengan Algoritma LVQ.....	40
3.3.6 Pengujian dengan Algoritma LVQ	48
3.3.7 Perhitungan Akurasi.....	51
3.4 Perancangan antarmuka.....	52
3.4.1 Antarmuka Halaman Beranda	52
3.4.2 Antarmuka Halaman Pelatihan LVQ	52
3.4.3 Antarmuka Halaman Pengujian LVQ	54
3.4.5 Antarmuka Halaman Uji Coba 1.....	55
3.4.6 Antarmuka Halaman Uji Coba 2.....	56
3.4.7 Antarmuka Halaman Uji Coba 3.....	57
3.4.6 Antarmuka Halaman Uji Coba 4.....	58
3.4.7 Antarmuka Halaman Uji Coba 5.....	58
3.4.8 Antarmuka Halaman Uji Coba 6.....	59
BAB IV IMPLEMENTASI	61
4.1 Lingkungan Implementasi	61
4.1.1 Lingkungan Perangkat Keras	61
4.1.2 Lingkungan Perangkat Lunak	61
4.2 Implementasi Program.....	62
4.2.1 Ekstraksi Fitur Warna	62
4.2.2 Binerisasi.....	63
4.2.3 Ekstraksi Fitur Luas	64
4.2.4 Algoritma <i>Learning Vector Quantization</i>	65
4.2.4.1 Pelatihan dengan algoritma LVQ.....	71
4.2.4.2 Pengujian dengan algoritma LVQ.....	74
4.3 Implementasi Antarmuka.....	75
4.3.1 Halaman Beranda.....	75

4.3.2 Halaman Pelatihan LVQ	75
4.3.3 Halaman Pengujian LVQ	76
4.3.4 Halaman Uji Coba 1	77
4.3.5 Halaman Uji Coba 2	77
4.3.6 Halaman Uji Coba 3	78
4.3.7 Halaman Uji Coba 4	79
4.3.8 Halaman Uji Coba 5	79
4.3.9 Halaman Uji Coba 6	80
BAB V PENGUJIAN DAN ANALISIS.....	81
5.1 Hasil dan Analisis Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran terhadap Hasil Akurasi	81
5.2 Hasil dan Analisis Pengujian untuk Mengetahui Pengaruh Nilai Konstanta Pengurang Laju Pembelajaran terhadap Hasil Akurasi	83
5.3 Hasil dan Analisis Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran Minimum terhadap Hasil Akurasi	85
5.4 Hasil dan Analisis Pengujian untuk Mengetahui Pengaruh Jumlah Iterasi Pembelajaran terhadap Hasil Akurasi	86
5.5 Hasil dan Analisis Pengujian Untuk Mengetahui Pengaruh Perbandingan Jumlah Data Latih dengan Data Uji terhadap Hasil Akurasi	88
5.6 Hasil dan Analisis Pengujian untuk Mengetahui Hasil Akurasi dari Proses Klasifikasi menggunakan <i>Minimum Distance</i>	90
BAB VI PENUTUP	92
6.1 Kesimpulan	92
6.2 Saran	92
DAFTAR PUSTAKA	DP-1
LAMPIRAN	L-1

DAFTAR GAMBAR

Gambar 2.1	Daun Nilam (a) Kualitas Baik (b) Kualitas Kurang Baik.....	12
Gambar 2.2	Koordinat Citra Digital	12
Gambar 2.3	Citra biner	13
Gambar 2.4	Citra grayscale	14
Gambar 2.5	Ilustrasi proses ekstraksi fitur luas daun.....	16
Gambar 2.6	Arsitektur Jaringan Saraf Tiruan	18
Gambar 2.7	Jaringan Dengan Lapisan Tunggal	19
Gambar 2.8	Jaringan dengan lapisan kompetitif	19
Gambar 2.9	Arsitektur jaringan <i>Learning Vector Quatization</i> (LVQ).....	22
Gambar 3.1	Diagram Alur Metodologi Penelitian	24
Gambar 3.2	Diagram Blok Sistem Klasifikasi Kualitas Daun Nilam menggunakan Algoritma LVQ	27
Gambar 3.3	Arsitektur Jaringan LVQ untuk Klasifikasi Kualitas Daun Nilam	28
Gambar 3.4	Diagram Alir Proses Kerja Sistem Secara Umum.....	30
Gambar 3.5	Diagram Alir Proses Ekstraksi Fitur Warna	31
Gambar 3.6	Diagram Alir Proses Binerisasi	32
Gambar 3.7	Diagram Alir Proses Ekstraksi Fitur Luas	33
Gambar 3.8	Diagram Alir Proses Pelatihan dengan Algoritma LVQ	35
Gambar 3.9	Diagram Alir Proses Pengujian dengan Algoritma LVQ	36
Gambar 3.10	Contoh Citra dengan ukuran 400x500 <i>Pixels</i>	37
Gambar 3.11	Contoh Citra dengan Ukuran 8x10 <i>Pixels</i>	37
Gambar 3.12	Antarmuka Halaman Beranda	52
Gambar 3.13	Rancangan Antarmuka Halaman Pelatihan LVQ	53
Gambar 3.14	Rancangan Antarmuka Halaman Pengujian LVQ.....	54
Gambar 3.15	Rancangan Antarmuka Halaman Uji Coba 1.....	55
Gambar 3.16	Rancangan Antarmuka Halaman Uji Coba 2.....	56
Gambar 3.17	Rancangan Antarmuka Halaman Uji Coba 3.....	57

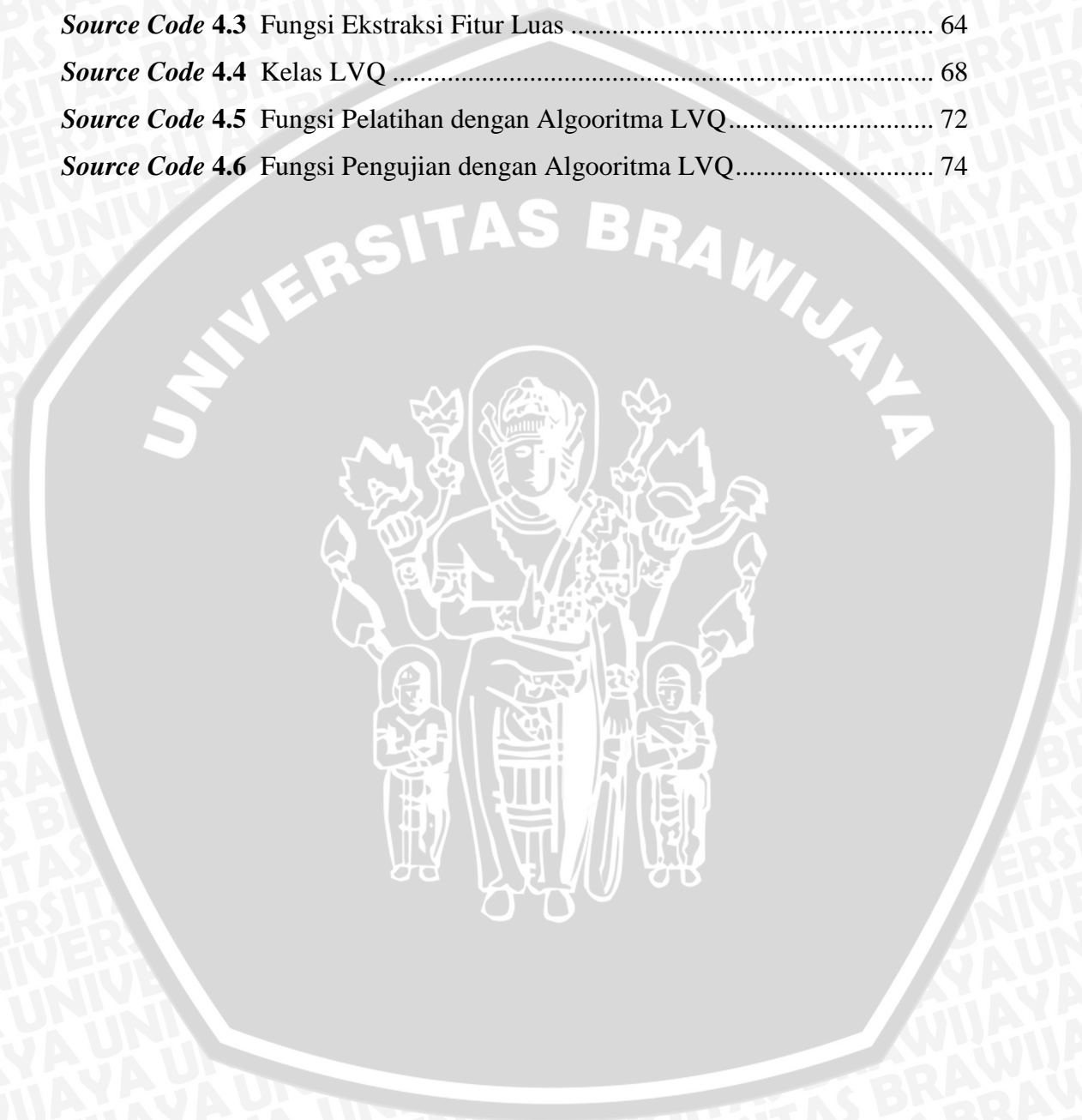
Gambar 3.18	Rancangan Antarmuka Halaman Uji Coba 4.....	58
Gambar 3.19	Rancangan Antarmuka Halaman Uji Coba 5.....	59
Gambar 3.20	Rancangan Antarmuka Halaman Uji Coba 6.....	60
Gambar 4.1	Implementasi Antarmuka Halaman Beranda.....	75
Gambar 4.2	Implementasi Antarmuka Halaman Pelatihan LVQ.....	76
Gambar 4.3	Implementasi Antarmuka Halaman Pengujian LVQ.....	76
Gambar 4.4	Implementasi Antarmuka Halaman Uji Coba 1.....	77
Gambar 4.5	Implementasi Antarmuka Halaman Uji Coba 2.....	78
Gambar 4.6	Implementasi Antarmuka Halaman Uji Coba 3.....	78
Gambar 4.7	Implementasi Antarmuka Halaman Uji Coba 4.....	79
Gambar 4.8	Implementasi Antarmuka Halaman Uji Coba 5.....	79
Gambar 4.9	Implementasi Antarmuka Halaman Uji Coba 6.....	80
Gambar 5.1	Grafik Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran terhadap Hasil Akurasi.....	82
Gambar 5.2	Grafik Pengujian untuk Mengetahui Pengaruh Nilai Konstanta Pengurang Laju Pembelajaran terhadap Hasil Akurasi.....	84
Gambar 5.3	Grafik Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran Minimum terhadap Hasil Akurasi.....	86
Gambar 5.4	Grafik Pengujian untuk Mengetahui Pengaruh Jumlah Iterasi Maksimum terhadap Hasil Akurasi.....	87
Gambar 5.5	Grafik Pengujian untuk Mengetahui Pengaruh Perbandingan Jumlah Data Latih dengan Data Uji terhadap Hasil Akurasi.....	89
Gambar 5.6	Grafik Pengujian untuk Mengetahui Hasil Akurasi dari Proses Klasifikasi menggunakan <i>Minimum Distance</i>	91

DAFTAR TABEL

Tabel 2.1 Kajian Pustaka.....	7
Tabel 3.1 Hasil Ekstraksi Nilai RGB	37
Tabel 3.2 Hasil Ekstraksi Nilai Rata-Rata <i>Red, Green, dan Blue</i>	39
Tabel 3.3 Hasil Konversi Citra RGB ke Citra <i>Grayscale</i>	39
Tabel 3.4 Hasil Konversi Citra <i>Grayscale</i> ke citra Biner.....	40
Tabel 3.5 Data Latih untuk Perhitungan Manual	40
Tabel 3.6 Data Uji untuk Perhitungan Manual.....	48
Tabel 3.7 Data Hasil Klasifikasi Perhitungan Manual	51
Tabel 4.1 Lingkungan Perangkat Keras	61
Tabel 4.2 Lingkungan Perangkat Lunak	61
Tabel 5.1 Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran terhadap Hasil Akurasi.....	81
Tabel 5.2 Pengujian untuk Mengetahui Pengaruh Nilai Konstanta Pengurang Laju Pembelajaran terhadap Hasil Akurasi	83
Tabel 5.3 Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran Minimum terhadap Hasil Akurasi.....	85
Tabel 5.4 Pengujian untuk Mengetahui Pengaruh Jumlah Iterasi Pembelajaran terhadap Hasil Akurasi	87
Tabel 5.5 Pengujian untuk Mengetahui Pengaruh Perbandingan Jumlah Data Latih dengan Data Uji terhadap Hasil Akurasi.....	89
Tabel 5.6 Pengujian untuk Mengetahui Hasil Akurasi dari Proses Klasifikasi menggunakan <i>Minimum Distance</i>	90

DAFTAR SOURCE CODE

Source Code 4.1 Fungsi Ekstraksi Fitur Warna.....	62
Source Code 4.2 Fungsi Binerisasi.....	63
Source Code 4.3 Fungsi Ekstraksi Fitur Luas	64
Source Code 4.4 Kelas LVQ	68
Source Code 4.5 Fungsi Pelatihan dengan Algoritma LVQ.....	72
Source Code 4.6 Fungsi Pengujian dengan Algoritma LVQ.....	74



BAB I PENDAHULUAN

1.1 Latar Belakang

Nilam (*Pogostemon cablin Benth*) merupakan salah satu tanaman yang telah dikenal sebagai tanaman yang sering dimanfaatkan dalam menghasilkan minyak atsiri. Diantara minyak atsiri yang diekspor oleh Indonesia, minyak nilam merupakan minyak yang diekspor dalam jumlah yang paling besar [ALA-07]. Volume ekspor minyak nilam Indonesia mencapai 800-1.500 ton pada periode 1995-1998, dengan nilai devisa US\$ 18-53 juta. Indonesia memiliki kontribusi sebesar 90% dalam memasok minyak nilam di dunia, sehingga minyak nilam tersebut berperan penting sebagai sumber devisa negara [PRA-06].

Baik harga maupun volume ekspor minyak nilam Indonesia di pasaran dunia sangat tidak stabil dalam beberapa tahun terakhir. Sejak tahun 2001 hingga tahun 2005 harga dan volume ekspor minyak nilam mengalami kemerosotan dari 1.189.000 kg seharga US\$ 17,30 per kg di tahun 2001 menjadi 1.102.982 kg seharga US\$ 7,16 per kg di tahun 2005 [SIL-06]. Kemerosotan harga dan volume ini salah satunya disebabkan oleh kualitas minyak nilam yang kurang baik [ALA-07].

Pada umumnya, untuk mengidentifikasi kualitas minyak nilam dapat dilihat dari komponen *Patchouli alcohol* [ALA-07], terutama pada daun karena daun merupakan bagian tanaman nilam yang memiliki kadar minyak tertinggi, dimana jumlah *Patchouli alcohol* pada daun nilam memiliki kandungan berkisar antara 30-50% [ARN-12]. Daun nilam yang dikatakan memiliki kualitas baik adalah daun nilam yang berukuran sempit, tebal, dan berwarna merah kekuningan [PRA-06].

Proses identifikasi kualitas daun nilam yang selama ini dilakukan oleh para petani nilam di daerah Indonesia adalah dengan cara melihat ciri fisik daun nilam secara langsung. Setelah itu, petani melakukan perkiraan bahwa daun nilam tersebut berkualitas baik atau tidak. Dengan kemampuan penglihatan manusia yang terbatas dan berbeda pada setiap orang, perkiraan terhadap kualitas daun yang diidentifikasi tersebut tidak dapat dipastikan kebenarannya. Untuk memastikan perkiraan tersebut, maka daun nilam yang akan diidentifikasi harus dibawa ke laboratorium untuk diuji dan didapatkan hasil identifikasi yang lebih tepat. Dengan

adanya laboratorium tersebut memang dapat membantu para petani nilam dalam mengidentifikasi kualitas daun nilam lebih tepat, namun waktu yang diperlukan cukup lama dan biaya yang dikeluarkan cukup banyak. Maka dari itu, seiring perkembangan teknologi saat ini, pembuatan sistem cerdas sangat diperlukan untuk mengotomatisasi proses identifikasi kualitas daun nilam. Sistem cerdas yang dibuat merupakan sistem yang mampu melakukan pengenalan dengan teknik pengolahan citra digital sehingga dapat mengenali ciri fisik daun lebih tepat dibandingkan dengan penglihatan manusia. Dengan adanya sistem cerdas yang dibuat ini, maka waktu yang diperlukan menjadi lebih singkat dan biaya yang dikeluarkan pun menjadi lebih sedikit dalam mengidentifikasi kualitas daun nilam.

Penelitian sebelumnya yang berkaitan dengan identifikasi yaitu penelitian dalam pengenalan penyakit daun gandum [ZHA-13]. Penelitian tersebut menggunakan algoritma *Learning Vector Quantization* dalam mengklasifikasikan penyakit daun gandum ke dalam tiga kelas berdasarkan fitur bentuk, warna dan tekstur penyakit daun gandum yang diekstraksi untuk menyusun 32 parameter. Tiga kelas penyakit daun gandum tersebut diantaranya adalah *Wheat stripe rust*, *Wheat leaf rust*, dan *Wheat Powdery Mildew* dengan tingkat akurasi yang dihasilkan dari klasifikasi pada masing-masing kelas sebesar 92,5%, 90%, dan 92,5% [ZHA-13].

Penelitian selanjutnya yang juga berkaitan dengan identifikasi yaitu penelitian dalam melakukan identifikasi daun jati [HAR-13]. Penelitian tersebut menggunakan algoritma *Backpropagation* dalam mengklasifikasikan daun jati ke dalam enam kelas berdasarkan fitur area, perimeter, diameter, panjang dan lebar daun yang kemudian diekstraksi menjadi 17 parameter. Enam kelas daun jati tersebut diantaranya adalah biotrop, emas, jobika, muna, prima dan super dengan tingkat akurasi yang dihasilkan sebesar 87,50% [HAR-13].

Selanjutnya, perbandingan antara algoritma *Learning Vector Quantization* dengan *Backpropagation* telah dilakukan pada penelitian klasifikasi penyakit diabetes mellitus [NUR-11]. Jika dibandingkan dengan algoritma *Backpropagation*, Algoritma LVQ menghasilkan akurasi yang lebih tinggi dan waktu yang digunakan lebih singkat dalam melakukan klasifikasi. Dalam penelitian tersebut didapatkan akurasi sebesar 82.56% pada LVQ dan 73.26% pada

Backpropagation, serta waktu selama 1,00 detik pada LVQ dan 1,53 detik pada *Backpropagation* [NUR-11].

Berdasarkan uraian di atas, maka dalam skripsi ini akan diimplemetasikan algoritma *Learning Vector Quantization* dalam menyelesaikan permasalahan klasifikasi kualitas daun nilam. Sistem yang dibuat memiliki masukan berupa citra daun nilam yang kemudian digunakan dalam pengolahan citra untuk mendapatkan parameter yang akan diolah menggunakan algoritma *Learning Vector Quantization*. Algoritma *Learning Vector Quantization* bekerja dengan melakukan pembelajaran pada lapisan kompetitif yang terawasi [QUR-10]. Dalam melakukan klasifikasi kualitas daun nilam, pembelajaran terawasi lebih sesuai karena adanya target keluaran yang ditentukan. Keunggulan dan penerapan algoritma tersebut diharapkan mampu membantu dalam melakukan klasifikasi yang lebih baik.

1.2 Rumusan Masalah

Dari latar belakang yang telah di sebutkan di atas, maka dapat ditarik rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasikan algoritma *Learning Vector Quantization* untuk klasifikasi kualitas daun nilam.
2. Berapa tingkat akurasi dari hasil implementasi algoritma *Learning Vector Quantization* untuk memberikan hasil klasifikasi kualitas daun nilam.

1.3 Batasan Masalah

Batasan masalah dalam penulisan tugas akhir ini adalah:

1. Citra yang digunakan diperoleh dari hasil pemindaian daun nilam tampak depan dengan resolusi 600 dpi (dots per inch) dengan format file gambar Joint Photographic Experts Group (JPEG) berukuran 400x500 pixels.
2. Daun nilam yang digunakan merupakan daun nilam yang diambil dari perkebunan nilam yang berada di Kesamben, kabupaten Blitar.
3. Daun nilam yang digunakan merupakan daun jenis nilam Aceh.
4. Daun nilam yang digunakan merupakan daun yang diambil pada bagian pucuk tanaman nilam yang berusia 6 bulan.
5. Daun nilam untuk kelas kualitas baik dan kualitas kurang baik masing-masing diambil sebanyak 50 daun .

6. Dalam tugas akhir ini tidak membahas budidaya tanaman nilam yaitu perlakuan dari proses pembibitan hingga sebelum proses panen, melainkan membahas klasifikasi kualitas daun nilam pasca panen.
7. Output dari sistem yang dibuat berupa kelas kualitas baik dan kualitas kurang baik.

1.4 Tujuan

Tujuan yang ingin dicapai dari pembuatan tugas akhir ini adalah:

1. Mengimplementasikan algoritma Learning Vector Quantization untuk klasifikasi kualitas daun nilam.
2. Mengetahui tingkat akurasi dari hasil implementasi algoritma Learning Vector Quantization untuk memberikan hasil klasifikasi kualitas daun nilam.

1.5 Manfaat

Manfaat yang diperoleh dari penelitian ini adalah:

1. Memberikan kemudahan dalam mengidentifikasi kualitas daun nilam berdasarkan ciri fisik daun nilam.
2. Mempercepat proses identifikasi kualitas daun nilam.
3. Tersedianya teknologi informasi untuk identifikasi kualitas daun nilam.

1.6 Sistematika Penulisan

Sistematika pembahasan dalam penelitian ini terbagi kedalam enam bab, sebagai berikut :

BAB I : Pendahuluan, terdiri atas Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan, Manfaat, dan Sistematika Penulisan.

BAB II : Kajian Pustaka dan Dasar Teori, terdiri atas Kajian Pustaka dan dasar teori yang berkaitan dengan topik yang dibahas oleh penulis.

BAB III : Metode Penelitian dan Perancangan, terdiri atas Metode Penelitian dan Perancangan.

BAB IV : Implementasi, membahas penerapan algoritma *Learning Vector Quantization* ((LVQ) dalam klasifikasi kualitas daun nilam.

BAB V : Pengujian dan Analisis, memuat hasil dan analisis pengujian terhadap system yang telah direalisasikan.:

BAB VI : Penutup, memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang telah dibuat dalam skripsi ini serta memuat saran-saran untuk pengembangan selanjutnya.



BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini merupakan bab yang berisi kajian pustaka dan dasar teori. Kajian pustaka dan dasar teori yang ditulis dipergunakan untuk menunjang penelitian dalam skripsi ini. Kajian pustaka memberikan informasi mengenai beberapa penelitian yang sudah ada. Dasar teori memberikan informasi mengenai beberapa teori yang dibutuhkan untuk menyusun skripsi ini. Beberapa teori yang sekiranya dibutuhkan adalah teori yang berkaitan dengan tanaman nilam, citra digital, Jaringan Syaraf Tiruan, dan algoritma *Learning Vector Quantization*.

2.1 Kajian Pustaka

Dalam pembuatan tugas akhir ini diperlukan beberapa pustaka untuk dibandingkan. Setiap pustaka tersebut menjelaskan judul, obyek / *input* dan parameter, metode / proses, dan *output* / hasil dari sebuah penelitian. Pada tabel 2.1 menjelaskan perbandingan dari pustaka yang berkaitan dengan tugas akhir ini. Bagian akhir dalam tabel 2.1 merupakan topik dan judul yang diusulkan oleh penulis.

Terdapat lima penelitian yang digunakan oleh penulis sebagai referensi dalam penelitian yang akan dilakukan. Penelitian pertama dilakukan oleh Siti Munawaroh dan Felix Andreas Sutanto. Objek pada penelitian ini adalah uang kertas yang nantinya akan diidentifikasi nominalnya. Penelitian kedua dilakukan oleh Puji Lestari, Dr. Ir. Bambang Hidayat, dan Eko Susatio, ST. Objek pada penelitian ini adalah daun teh yang nantinya akan diidentifikasi jenis cacatnya.

Penelitian ketiga yang dilakukan oleh Asep Haryono dan Aziz Kustiyo. Objek pada penelitian ini adalah daun tanaman jati yang nantinya akan diidentifikasi jenis varietasnya. Penelitian keempat dilakukan oleh Muh Ismail dan Mustikasari. Objek pada penelitian ini adalah daun teh yang nantinya akan diidentifikasi jenis penyakitnya. Penelitian kelima dilakukan oleh Feiyun Zhang. Objek pada penelitian ini adalah daun gandum yang nantinya akan diidentifikasi jenis penyakitnya. Uraian dan penjelasan terhadap kelima penelitian tersebut ditunjukkan pada tabel 2.1.

Tabel 2.1 Kajian Pustaka

No	Judul	Obyek	Metode	Output
		<i>Input dan Parameter</i>	<i>Proses</i>	<i>Hasil Penelitian</i>
1.	“Pengolahan Citra Digital untuk Identifikasi Uang Kertas” [MUN-10]	Uang Kertas	Pengolahan Citra Digital (perbaikan citra, segmentasi dan pencocokan)	Output dari penelitian ini yaitu pengenalan terhadap uang kertas dengan nominal 1000, 5000, 10000, 20000, 50000, dan 100000.
		rata-rata warna merah, rata-rata warna hijau, rata-rata warna biru	<ol style="list-style-type: none"> 1. <i>Input</i> citra uang kertas 2. Perbaikan citra (kontras dan <i>brightness</i>) 3. Segmentasi melalui <i>cropping</i> dengan ukuran tertentu 4. Ekstraksi ciri (rata-rata RGB) 5. Pencocokan data antara <i>query</i> dengan pola yang terdapat pada <i>database</i> 	Akurasi yang didapatkan dari penelitian ini adalah 86%.
2	“Deteksi Cacat Daun Teh <i>Camellia sinensis</i> dengan Pengolahan Citra Digital dan JST <i>Learning Vector Quantization</i> ” [LES-12]	Cacat daun teh <i>Camellia sinensis</i>	<i>Learning Vector Quantization</i> (LVQ).	Sistem dengan klasifikasi 4 kelas, keluarannya yaitu daun teh normal tanpa cacat, daun teh yang cacat blister, cacat karena hama mite, cacat karena hama thrips. Sedangkan Sistem dengan klasifikasi 2 kelas,

				keluarannya yaitu daun normal tanpa cacat dan daun cacat karena hama.
		intensitas warna merah, intensitas warna hijau, intensitas warna biru, RGB rata-rata, indeks warna merah, indeks warna hijau, indeks warna biru, intensitas warna hue, intensitas warna <i>saturation</i> , intensitas warna <i>intensity</i> , HSI rata-rata, indeks warna <i>hue</i> , indeks warna <i>saturation</i> , dan indeks warna <i>intensity</i> , serta <i>Principal Component Analysis</i> (PCA).	<p>Metode Pertama</p> <ol style="list-style-type: none"> 1. <i>Input</i> citra daun teh 2. <i>Preprocessing</i> (<i>grayscale</i>) 3. PCA 4. Pelatihan dengan LVQ 5. Pengujian dengan LVQ <p>Metode Kedua</p> <ol style="list-style-type: none"> 1. <i>Input</i> citra daun teh 2. <i>Preprocessing</i> (<i>grayscale</i>, deteksi tepi, dilasi, <i>filling</i>, <i>labelling</i> dan <i>thresholding</i>, <i>cropping</i>) 3. Ekstraksi ciri 4. PCA 5. Pelatihan dengan LVQ 6. Pengujian dengan LVQ 	Tingkat keberhasilan sistem terbaik dihasilkan dari sistem dengan klasifikasi dua kelas (daun normal dan cacat) dan metode kedua dengan tingkat keakuratan seluruh data sebesar 85,38 %.
3	“Identifikasi Daun Tanaman Jati Menggunakan Jaringan Syaraf Tiruan <i>Backpropagation</i> dengan Ekstraksi Fitur Ciri Morfologi Daun” [HAR-13]	<p>Daun tanaman jati</p> <p><i>area</i>, <i>perimeter</i>, <i>diameter</i>, <i>physiological length</i>, <i>physiological width</i>, <i>smooth</i></p>	<p><i>Backpropagation</i></p> <ol style="list-style-type: none"> 1. <i>Input</i> citra daun teh 2. <i>Preprocessing</i> (<i>grayscale</i>, binerisasi) 3. Ekstraksi ciri 	Enam kelas daun jati, yaitu biotrop, emas, jobika, muna, prima dan super tingkat akurasi yang dihasilkan sebesar 87,50%

		<p><i>factor, aspect ratio, form factor, rectangularity, narrow factor, perimeter</i></p> <p><i>ratio of diameter, perimeter ratio of physiological length and physiological width</i> serta lima ciri urat daun (<i>vein features</i>)</p>	<p>4. Pelatihan dengan <i>Backpropagation</i></p> <p>5. Pengujian dengan <i>Backpropagation</i></p>	
4	<p>“<i>Intelligent System for Tea Leaf Disease Detection</i>”</p> <p>[ISM-13]</p>	<p>Penyakit daun teh</p>	<p><i>Learning Vector Quantization (LVQ)</i>.</p>	<p>4 kelas penyakit daun teh, yaitu <i>Healthy</i> (tidak terdapat penyakit), <i>Meliola sp</i>, <i>Exobasidium Vexam</i>, dan <i>Chepaleuros sp</i>.</p>
		<p>Intensitas warna grayscale</p>	<p>1. <i>Input</i> citra daun teh</p> <p>2. <i>Preprocessing</i> (<i>high-pass-filter, sharpening, cropping</i>)</p> <p>3. Ekstraksi ciri</p> <p>4. Pelatihan dengan LVQ</p> <p>5. Pengujian dengan LVQ</p>	<p>Tingkat keakuratan pada data uji diperoleh sebesar 100% untuk kelas <i>Healthy</i>, 80% untuk kelas <i>Meliola sp</i>, 60% untuk kelas <i>Exobasidium vexan</i>, 60% untuk kelas <i>Cephaleuros sp</i>.</p>
5	<p>“<i>Image Recognition of Wheat Leaf Disease based on Learning Vector</i></p>	<p>Penyakit daun gandum</p>	<p><i>Learning Vector Quantization (LVQ)</i></p>	<p>3 kelas penyakit daun gandum, yaitu <i>Wheat stripe rust</i>, <i>Wheat leaf rust</i>, dan <i>Wheat Powdery Mildew</i>.</p>

	<i>Quantization Neural Network and Combination Feature Parameters</i> [ZHA-13]	parameter fitur bentuk, warna dan tekstur penyakit gandum diekstraksi untuk menyusun 32 parameter.	<ol style="list-style-type: none"> 1. <i>Input</i> citra daun gandum 2. <i>Preprocessing</i> 3. Ekstraksi fitur 4. Pelatihan dengan LVQ 5. Pengujian dengan LVQ 	Tingkat keakuratan pada data uji diperoleh sebesar 92,5% untuk kelas <i>Wheat stripe rust</i> , 90% untuk kelas <i>Wheat leaf rust</i> , dan 92,5% untuk kelas <i>Wheat Powdery Mildew</i> .
6	“Klasifikasi Kualitas Daun Nilam Menggunakan Citra Digital dan Algoritma <i>Learning Vector Quantization</i> ” [Usulan]	Kualitas daun nilam rata-rata warna merah, rata-rata warna hijau, rata-rata warna biru, luas permukaan daun	<i>Learning Vector Quantization (LVQ)</i> <ol style="list-style-type: none"> 1. <i>Input</i> citra daun nilam 2. Ekstraksi fitur warna 3. Binerisasi 4. Ekstraksi fitur luas 5. Pelatihan dengan LVQ 6. Pengujian dengan LVQ 	Dua kelas kualitas daun nilam, yaitu kelas kualitas baik dan kualitas kurang baik.

Sumber: [MUN-10] [LES-12] [HAR-13] [ISM-13] [ZHA-13]

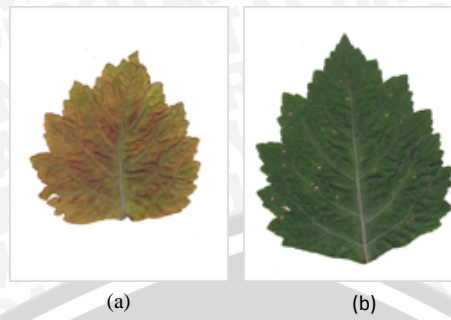
2.2 Tanaman Nilam

Nilam termasuk famili *Labaiatae*, ordo *Lamiales*, kelas *Angiospermae* dan divisi *Spermatophyta*. Di Indonesia terdapat tiga jenis nilam yang dapat dibedakan menurut karakter morfologinya, kandungan PA dan kualitas minyak serta ketahanan terhadap cekaman biotik dan abiotik. Ketiga jenis nilam tersebut adalah 1). *P. cablin Benth* disebut nilam aceh, 2). *P. Heyneanus Benth* disebut nilam jawa, dan 3). *P. hortensis Becker* disebut nilam sabun [SUP-11]

Nilam merupakan salah satu tanaman yang telah dikenal sebagai tanaman yang banyak ditanam untuk diambil minyaknya. Minyak Nilam tersebut dimanfaatkan sebagai bahan kosmetik, parfum, antiseptik, dan lain-lain. Minyak nilam merupakan salah satu jenis dari minyak atsiri. Minyak atsiri dapat ditemukan pada daun, bunga, buah, biji, batang, kulit, dan akar. Untuk tanaman nilam, minyak atsirinya banyak tersimpan di dalam sel-sel kelenjar minyak pada daun [SUP-11].

Pada umumnya, *Patchouli alcohol* merupakan komponen utama yang menentukan kualitas minyak nilam [ALA-07]. Daun merupakan bagian tanaman nilam yang memiliki kadar minyak tertinggi. Khususnya pada tiga pasang daun termuda memiliki jumlah *Patchouli alcohol* sebagai kandungan utama berkisar antara 30-50% [ARN-12].

Kualitas dari daun akan menentukan kualitas minyak yang dihasilkan. Daun dengan kualitas yang baik akan menghasilkan minyak dengan kualitas yang baik. Sebaliknya, daun dengan kualitas kurang baik akan menghasilkan minyak dengan kualitas yang kurang baik. Daun nilam dengan kualitas baik adalah daun yang memiliki kandungan *Patchouli alcohol* yang tinggi. Daun dengan kualitas baik tersebut dapat diidentifikasi melalui ciri fisiknya yaitu memiliki ukuran daun yang sempit, tebal, dan berwarna merah kekuningan. Sedangkan daun dengan kualitas kurang baik memiliki ukuran daun yang luas, tipis, dan berwarna hijau. Daun nilam dengan kualitas baik dan kualitas kurang baik ditunjukkan pada Gambar 2.1.

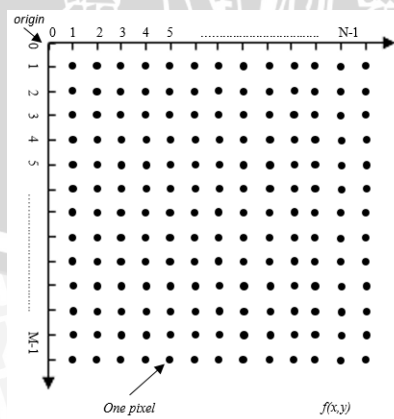


Gambar 2.1 Daun Nilam (a) Kualitas Baik (b) Kualitas Kurang Baik
Sumber: [Observasi]

2.3 Citra Digital

Representasi atau gambaran yang mirip dan sebagai imitasi dari sebuah objek disebut dengan citra. Citra dapat memiliki beberapa sifat sebagai keluaran suatu sistem perekam data, yaitu citra bersifat optik, analog, atau digital. Foto merupakan citra yang bersifat optik. Sinyal-sinyal video seperti gambar pada monitor televisi merupakan citra yang bersifat analog. Sedangkan citra yang bersifat digital adalah citra yang dapat langsung disimpan pada media penyimpanan [PUT-10].

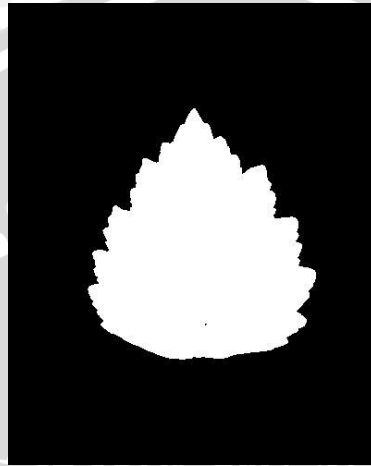
Representasi deretan bit dari nilai-nilai real maupun kompleks dalam sebuah larik atau *array* disebut dengan citra digital. Sebuah citra dapat dinyatakan dalam fungsi $f(x,y)$. (x,y) menyatakan koordinat spasial, sedangkan nilai dari f pada titik (x,y) menyatakan tingkat kecerahan citra pada titik tersebut. Fungsi $f(x,y)$ tersebut disusun dalam bentuk matriks berukuran M baris dan N kolom [PUT-10]. Koordinat pada sebuah citra digital dapat dilihat pada Gambar 2.2.



Gambar 2.2 Koordinat Citra Digital
Sumber: [PUT-10]

2.3.1 Citra Biner

Citra digital yang disebut dengan citra biner adalah citra yang memiliki dua nilai *pixel* yaitu hitam dan putih. Sebutan lain dari citra biner adalah citra B&W (*black and white*) atau citra monokrom. Citra biner hanya membutuhkan 1 bit untuk mewakili nilai setiap *pixel*-nya [PUT-10]. Citra biner ditunjukkan pada Gambar 2.3.



Gambar 2.3 Citra biner

Citra biner dihasilkan dari proses *thresholding*. Terdapat dua macam teknik *thresholding*, yaitu: *Uniform Thresholding* dan *Adaptive Thresholding*. Metode yang digunakan dalam *uniform thresholding* adalah dengan menentukan suatu batas level, kemudian batas level tersebut dipergunakan untuk menentukan warna *pixel*. Warna akan dirubah menjadi putih apabila nilai *pixel* lebih dari batas level (*threshold*), dan sebaliknya warna akan dirubah menjadi hitam apabila nilai *pixel* kurang dari batas level (*threshold*). [SAR-11].

2.3.2 Citra Grayscale

Citra digital yang disebut sebagai citra *grayscale* adalah citra yang hanya memiliki satu nilai kanal pada setiap *pixel*-nya. Tingkat intensitas pada citra ditunjukkan oleh nilai tersebut. Citra grayscale hanya memiliki warna dari hitam, keabuan, dan putih. Tingkatan keabuan pada citra *grayscale* merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Gambar 2.4 menunjukkan citra *grayscale* dengan kedalaman warna 8 bit (256 kombinasi warna keabuan) [PUT-10].



Gambar 2.4 Citra grayscale

Citra *grayscale* juga dapat dikatakan sebagai citra hasil pemrosesan citra warna menjadi citra dengan gradasi warna hitam dan putih. Untuk menghasilkan citra tersebut dapat dilakukan dengan cara menjumlahkan ketiga nilai unsur warna dasar, kemudian dirata-rata seperti pada persamaan (2-1). Namun perubahan citra warna ke *grayscale* lebih tepat dengan memberikan bobot yang berbeda pada setiap unsur warna *red*, *green*, dan *blue* seperti pada persamaan (2-2) [SAR-11].

$$X = \frac{R+G+B}{3} \dots\dots\dots(2-1)$$

$$X = (0,299 * R) + (0,114 * G) + (0,587 * B) \dots\dots\dots(2-2)$$

dimana :

X : Warna *grayscale* (*pixel*)

R : Warna merah (*pixel*)

G : Warna hijau (*pixel*)

B : Warna biru (*pixel*)

2.3.3 Citra Warna

Jumlah warna dalam citra warna bisa beragam mulai dari 16, 256, 65536 atau 16 juta warna yang masing-masing direpresentasikan oleh 4,8,16 atau 24 bit data untuk setiap *pixel*-nya. Warna yang ada terdiri dari 3 komponen utama yaitu nilai merah (*red*), nilai hijau (*green*) dan nilai biru (*blue*). Paduan ketiga komponen utama pembentuk warna tersebut dikenal sebagai RGB *color* yang nantinya akan membentuk citra warna [ANN-09].



2.4 Ekstraksi Fitur

Ekstraksi fitur merupakan bagian penting dalam analisis suatu citra. fitur dapat dikatakan sebagai keunikan dari sebuah objek. Fitur yang baik adalah fitur yang dapat membedakan sebuah objek dengan objek yang lainnya dan jumlahnya sedikit. Fitur yang jumlahnya sedikit dapat menghemat waktu komputasi dan ruang penyimpanan yang digunakan [PUT-10].

2.4.1 Fitur Warna

Proses ekstraksi fitur warna pada citra warna sebuah objek dapat dilakukan dengan menghitung rata-rata warna merah, hijau dan biru pada keseluruhan pixel gambar. Perhitungan rata-rata warna merah, hijau, dan biru masing-masing ditunjukkan pada persamaan (2-3), (2-4), dan (2-5) [MUN-10].

$$Grratr = \frac{\sum Grtotr}{N} \dots\dots\dots(2-3)$$

$$Grratg = \frac{\sum Grtotg}{N} \dots\dots\dots(2-4)$$

$$Grratb = \frac{\sum Grtotb}{N} \dots\dots\dots(2-5)$$

dimana:

Grratr : Nilai rata-rata warna merah

Grtotr : Total warna merah

Grratg : Nilai rata-rata warna hijau

Grtotg : Total warna hijau

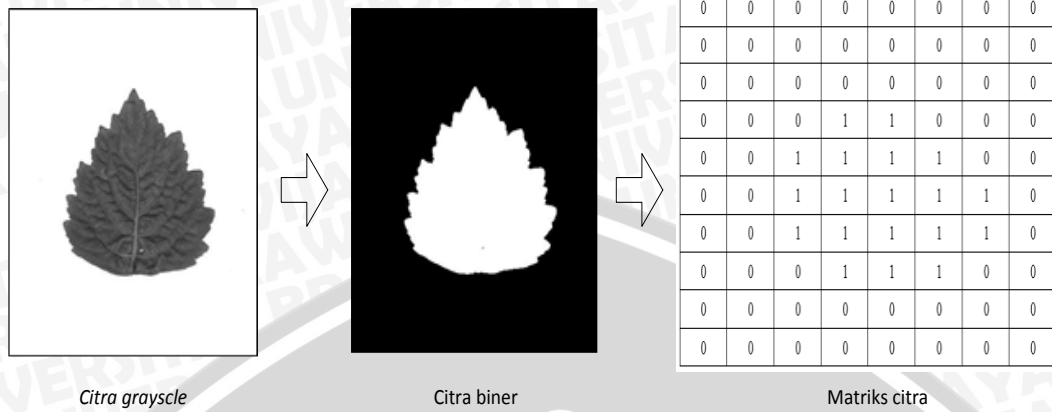
Grratb : Nilai rata-rata warna biru

Grtotb : Total warna biru

N : Total *pixel*

2.4.2 Fitur Luas

Proses ekstraksi fitur luas pada sebuah objek dapat dilakukan dengan sebuah masukan berupa citra biner dengan *threshold* tertentu. Nilai *threshold* tersebut dapat memisahkan *pixel* yang merupakan objek dan *pixel* yang merupakan latar belakang citra. Luas sebuah objek didapatkan dari perhitungan jumlah *pixel* bernilai 1 yang berada di dalam tepi objek [HAR-13]. Gambar 2.5 mengilustrasikan proses ekstraksi fitur luas dari citra daun.



Gambar 2.5 Ilustrasi proses ekstraksi fitur luas daun

2.5 Jarak Euclidean

Jarak *Euclidean* adalah jarak yang paling umum digunakan pada data numerik. Jarak *Euclidean* menghitung akar dari kuadrat selisih antara koordinat sepasang objek [SIG-13]. Jarak *Euclidean* ditentukan dengan persamaan (2-6).

$$d(\vec{u}, \vec{v}) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \dots + (u_n - v_n)^2} \dots\dots\dots(2-6)$$

dengan:

- $d(\vec{u}, \vec{v})$ = jarak *Euclidean*
- \vec{u} = vektor ciri masukan
- \vec{v} = vektor ciri basis data
- n = jumlah vektor ciri

2.6 Jaringan Syaraf Tiruan (JST)

Jaringan Syaraf Tiruan (JST) merupakan sistem pemroses informasi yang bekerja dengan karakteristik tertentu menyerupai jaringan syaraf biologis. JST telah dikembangkan sebagai model umum matematika dari aspek syaraf biologis atau kognitif manusia, yaitu dinyatakan pada beberapa asumsi seperti berikut [FAU-94]:

- a. Pada elemen-elemen yang disebut sebagai neuron terjadi pemrosesan informasi.
- b. Perambatan sinyal terjadi di antara neuron melalui koneksi yang saling terhubung.

- c. Setiap koneksi yang saling terhubung mempunyai bobot yang bersesuaian sehingga berfungsi untuk mengalikan sinyal yang dikirim
- d. Setiap neuron memiliki fungsi aktivasi yang pada umumnya tidak linear pada masukan jaringan sehingga dapat menentukan sinyal keluarannya.

Terdapat beberapa definisi JST yang dapat dideskripsikan, diantaranya adalah sebagai berikut [SIA-11]:

- a. JST merupakan suatu metode pemrosesan informasi yang memodelkan dan mensimulasikan sistem syaraf biologis serta berbasis komputer.
- b. JST merupakan suatu model matematika yang memiliki elemen-elemen proses yang tersusun dalam lapisan-lapisan.
- c. JST merupakan sistem berbasis komputer yang dibangun dari elemen-elemen proses yang saling terhubung sehingga dapat memproses informasi yang masuk dan mampu memberi respon.
- d. JST merupakan teknologi berbasis komputer yang dimodelkan berdasarkan syaraf yang mampu memproses berbagai macam masukan.

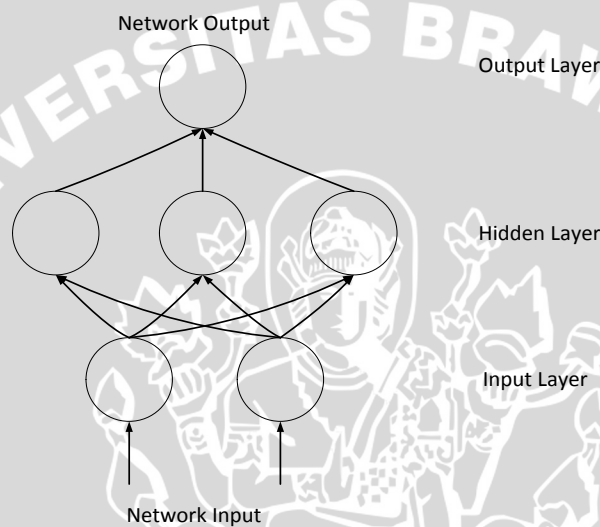
2.6.1 Karakteristik Jaringan Syaraf Tiruan

Jaringan syaraf tiruan memiliki tiga karakteristik utama sebagai tiruan dari sistem jaringan biologis manusia, diantaranya adalah sebagai berikut [SIA-11]:

- a. Arsitektur Jaringan
Merupakan pola dari neuron-neuron yang berhubungan. Neuron-neuron yang berhubungan tersebut membentuk suatu jaringan.
- b. Algoritma Jaringan
Merupakan metode yang digunakan untuk menentukan nilai bobot. Terdapat dua metode dalam algoritma jaringan syaraf tiruan, yaitu metode dimana JST tersebut melakukan pembelajaran (pelatihan) dan metode dimana JST tersebut melakukan pengenalan (pengujian).
- c. Fungsi Aktivasi
Merupakan fungsi untuk menentukan nilai keluaran. Nilai keluaran tersebut juga ditentukan oleh nilai total masukan pada neuron. Fungsi aktivasi pada suatu algoritma jaringan dengan algoritma jaringan yang lainnya dapat berbeda.

2.6.2 Arsitektur Jaringan Syaraf Tiruan

Pada arsitektur JST terdapat lapisan neuron (*neuron layer*) yang merupakan lapisan-lapisan yang terdiri dari kumpulan neuron. Setiap neuron pada satu lapisan berhubungan dengan lapisan-lapisan sebelum dan sesudahnya. Informasi yang masuk ke dalam jaringan syaraf akan mulai merambat dari lapisan masukan hingga ke lapisan keluaran melalui lapisan tersembunyi (*hidden layer*) [SIN-14]. Pada Gambar 2.6 digambarkan arsitektur dari jaringan saraf tiruan secara umum beserta keterhubungan antar neuron di dalamnya.



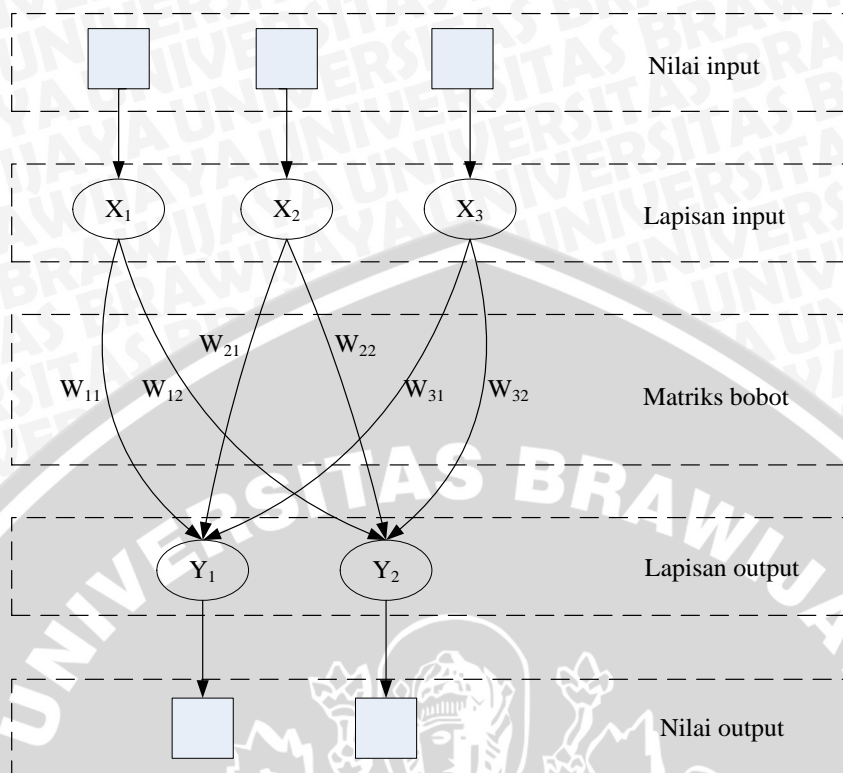
Gambar 2.6 Arsitektur Jaringan Saraf Tiruan
Sumber: [SIA-11]

Terdapat 3 macam arsitektur JST, yaitu [SIN-14]:

- Jaringan dengan lapisan tunggal (*single layer net*)
- Jaringan dengan banyak lapisan (*multilayer net*)
- Jaringan dengan lapisan kompetitif (*competitive layer net*)

2.6.3 Jaringan dengan lapisan tunggal (*single layer net*)

Jaringan dengan lapisan tunggal hanya memiliki 2 lapisan yaitu lapisan *input* dan *output* dengan bobot-bobot yang terhubung. Informasi yang masuk dalam jaringan ini akan diolah menjadi keluaran tanpa melalui lapisan tersembunyi. Pada gambar 2.7 dinyatakan bahwa setiap neuron pada kedua lapisan (lapisan masukan dan lapisan keluaran) saling terhubung dan diantaranya terdapat bobot yang bersesuaian untuk menentukan besarnya hubungan antara dua neuron [SIN-14].

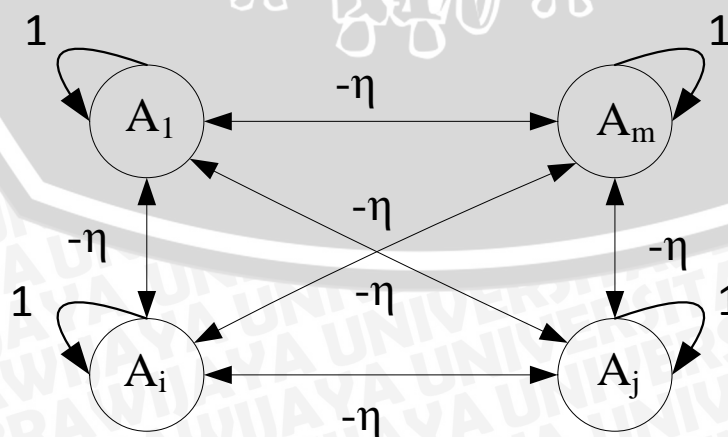


Gambar 2.7 Jaringan Dengan Lapisan Tunggal

Sumber: [SIN-14]

2.6.4 Jaringan dengan lapisan kompetitif (*competitive layer net*)

Jaringan dengan lapisan kompetitif memiliki sekumpulan neuron yang bersaing untuk mendapatkan hak untuk menjadi aktif. Neuron-neuron yang berhubungan dalam lapisan kompetitif ini biasanya tidak diperlihatkan pada diagram arsitektur. Salah satu contoh arsitektur jaringan dengan lapisan kompetitif digambarkan dalam gambar 2.8 dengan bobot $-\eta$ [SIN-14].



Gambar 2.8 Jaringan dengan lapisan kompetitif

Sumber: [SIN-14]

2.6.5 Prinsip Dasar Pelatihan/Pembelajaran Jaringan Syaraf Tiruan

Dalam pelatihan untuk mengasosiasikan pola dapat diambil contoh paling sempurna adalah kasus pada manusia. Jika seorang bayi dihadapkan dengan buah jeruk berwarna kuning kemudian seseorang yang memegang buah jeruk tersebut mengatakan “kuning” secara berulang kali maka terjadi peningkatan kekuatan koneksi *synaptic*. Peningkatan tersebut terjadi jika sel syaraf yang sesuai dari pusat optik secara simultan teraktivitasi bersama dengan yang berasal dari pusat auditori suara untuk kata jeruk yang diucapkan [SIA-11].

Prinsip tersebut dinyatakan oleh Hebb (1949). Hebb hanya menyatakan jika neuron A dan neuron B tereksitasi secara simultan maka kekuatan koneksi *synaptic* diantara keduanya akan meningkat. Prinsip inilah yang biasanya menjiwai semua model matematis dari Jaringan Syaraf Tiruan yang ada sekarang [SIA-11].

2.6.6 Proses Algoritma Pelatihan/Pembelajaran Jaringan Syaraf Tiruan

Sebelum diaplikasikan, pelatihan JST dilakukan untuk mengenal fungsi pemetaan. Pelatihan merupakan proses dimana JST melakukan pembelajaran dengan menyesuaikan bobot terkoneksi jaringan. Suatu Jaringan Syaraf Tiruan belajar dari pengalaman. Pada umumnya, terdapat 3 tugas dalam proses pembelajaran pada JST, diantaranya adalah sebagai berikut [SIA-11]:

- a. Perhitungan *output*
- b. Membandingkan *output* dengan target yang ditentukan
- c. Menyesuaikan bobot dan mengulang prosesnya.

Proses pelatihan atau pembelajaran tersebut merupakan proses penentuan bobot antar neuron. Penentuan bobot tersebut dapat menyelesaikan sebuah masalah.

Terdapat dua proses pembelajaran pada JST, yaitu [SIA-11]:

- a. Belajar dengan pengawasan (*Supervised learning*)
- b. Belajar tanpa pengawasan (*Unsupervised learning*)

2.6.7 Belajar dengan Pengawasan (*Supervised Learning*)

Belajar dengan Pengawasan (*Supervised Learning*) merupakan konsep pelatihan untuk menghasilkan suatu target keluaran yang ditentukan. Dengan adanya target keluaran, perubahan masukan akan disesuaikan oleh keluaran dengan mengubah bobot interkoneksinya sesuai algoritma belajar yang ditentukan.

Pasangan pola-pola masukan dan keluaran digunakan dalam pelatihan tersebut [SIA-11].

2.6.8 Algoritma Jaringan Syaraf Tiruan

Suatu permasalahan yang akan diselesaikan mempengaruhi penggunaan Jaringan Syaraf Tiruan. Beberapa permasalahan yang dapat diselesaikan dengan Jaringan Syaraf Tiruan diantaranya adalah pengenalan pola dan optimasi. Memilih algoritma yang terbaik sangat diperlukan untuk dapat menyelesaikan masalah.

Beberapa algoritma dalam jaringan syaraf tiruan yaitu:

- a. Algoritma Jaringan *Kohonen*
- b. Algoritma Jaringan *Fractal*
- c. Algoritma Jaringan *Learning Vector Quantization*
- d. Algoritma Jaringan *Cyclic*
- e. Algoritma Jaringan *Alternating Projection*
- f. Algoritma Jaringan *Hammimg*
- g. Algoritma Jaringan *Feedforwad* Banyak Lapis

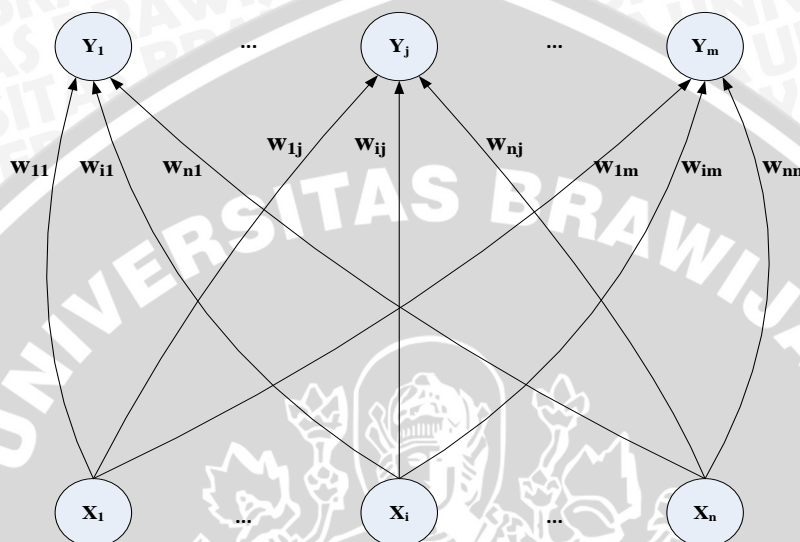
Dari beberapa algoritma jaringan syaraf tiruan tersebut, maka salah satu algoritma jaringan yang cocok dan dapat menyelesaikan masalah sesuai permasalahan harus ditentukan. Peneliti juga dapat menciptakan algoritma jaringan sendiri jika diperlukan [SIA-11].

2.7 Learning Vector Quantization (LVQ)

Learning Vector Quantization (LVQ) merupakan salah satu algoritma dalam jaringan syaraf tiruan (JST) dengan tipe arsitektur jaringan lapis-tunggal umpan-maju (*Single Layer Feedforward*) yang terdiri atas unit masukan dan unit keluaran. LVQ melakukan pembelajaran pada lapisan kompetitif yang terawasi. Algoritma LVQ bekerja dengan mengklasifikasikan vektor-vektor *input* ke dalam kelas-kelas sesuai dengan jarak antara vektor-vektor *input* tersebut. 2 buah vektor *input* akan diletakkan oleh lapisan kompetitif ke dalam kelas yang sama jika kedua vektor *input* tersebut mendekati sama [KUS-03].

2.7.1 Arsitektur Jaringan *Learning Vector Quantization*

Arsitektur jaringan *Learning Vector Quantization* (LVQ) dapat dilihat pada gambar 2.9. Gambar 2.9 memperlihatkan bahwa struktur jaringan saraf LVQ terdiri dari tiga lapisan. Lapisan pertama adalah masukan, lapisan kedua adalah bobot dan lapisan ketiga adalah keluaran.



Gambar 2.9 Arsitektur jaringan *Learning Vector Quantization* (LVQ)
Sumber: [FAU-94]

2.7.2 Algoritma *Learning Vector Quantization*

Algoritma pada jaringan LVQ memiliki tujuan untuk menemukan unit *output* yang paling dekat dengan vektor *input*. Untuk mencapai tujuan itu, jika x dan w_c berada pada kelas yang sama, maka vektor bobot akan disesuaikan agar lebih dekat dengan vektor masukan. Jika x dan w_c berada pada kelas yang berbeda, maka vektor bobot akan disesuaikan agar menjauhi vektor masukan. Tata nama yang digunakan dalam algoritma ini adalah sebagai berikut [FAU-94]:

- x Vektor pelatihan ($x_1, \dots, x_i, \dots, x_n$).
- T Kategori atau kelas untuk vektor pelatihan.
- w_j Vektor bobot terhadap kelas ke- j .
- C_j Kategori atau kelas ke- j
- $\|x-w_j\|$ Jarak *Euclidean* antara vektor *input* dan (vektor bobot untuk) *output* ke- j .

Langkah-langkah algoritma pembelajaran metode LVQ dapat dijelaskan sebagai berikut [FAU-94]:

Langkah 0 Menginisialisasi vektor referensi dan menginisialisasi laju pembelajaran (α).

Langkah 1 Selama kondisi berhenti adalah salah, lakukan Langkah 2 sampai dengan Langkah 6.

Langkah 2 Untuk setiap vektor pelatihan x , lakukan Langkah 3 sampai dengan Langkah 4

Langkah 3 Tentukan j (sebut dengan C_j), dimana $\|x - w_j\|$ adalah minimum.

Langkah 4 Update w_j sebagai berikut:

Jika $T = C_j$, maka

$$w_j(\text{baru}) = w_j(\text{lama}) + \alpha[x - w_j(\text{lama})]$$

Jika $T \neq C_j$, maka

$$w_j(\text{baru}) = w_j(\text{lama}) - \alpha[x - w_j(\text{lama})]$$

Langkah 5 Kurangi laju pembelajaran.

$$\alpha = \alpha * \text{pengurang}$$

Langkah 6 Uji kondisi berhenti:

Kondisi berhenti dapat ditentukan dengan menetapkan jumlah iterasi atau laju pembelajaran mencapai nilai yang cukup kecil yaitu tidak kurang dari laju pembelajaran minimum.

2.8 Evaluasi

Evaluasi dilakukan agar dapat mengetahui tingkat keberhasilan sistem dalam melakukan klasifikasi. Tingkat keberhasilan tentunya dikaitkan dengan jumlah data hasil klasifikasi yang relevan (sesuai dengan target) maupun yang tidak relevan (tidak sesuai dengan target). Konsep evaluasi yang digunakan dalam penelitian ini adalah akurasi.

Akurasi merupakan nilai dari perbandingan jumlah data yang relevan atau sesuai dengan target dengan jumlah seluruh data yang ada. umumnya, nilai akurasi direpresentasikan ke dalam persentase. Persamaan akurasi ditunjukkan oleh persamaan (2-7) [NUG-14].

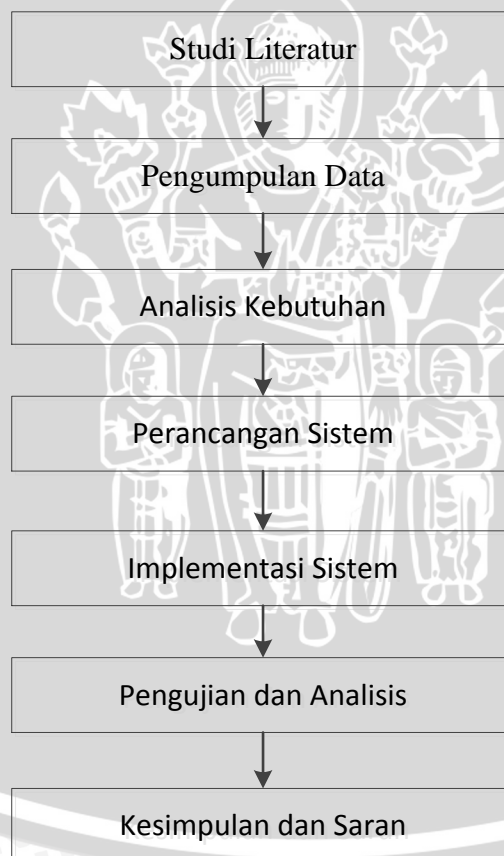
$$\text{Akurasi} = \frac{\text{jumlah data yang relevan}}{\text{jumlah seluruh data yang ada}} \times 100\% \dots \dots \dots (2-7)$$

BAB III METODE PENELITIAN DAN PERANCANGAN

Bab ini menjelaskan metode penelitian dan perancangan yang dilakukan dalam penelitian klasifikasi kualitas daun nilam menggunakan algoritma *Learning Vector Quantization* (LVQ).

3.1 Metode Penelitian

Bagian ini menjelaskan langkah-langkah yang dilakukan dalam penelitian klasifikasi kualitas daun nilam menggunakan algoritma *Learning Vector Quantization*. Gambar 3.1 menunjukkan diagram alur metode penelitian yang dilakukan dalam penelitian ini.



Gambar 3.1 Diagram Alur Metodologi Penelitian

3.1.1 Studi Literatur

Tahap ini bertujuan untuk mendapatkan studi literatur sebagai sumber acuan dalam penelitian ini. Acuan tersebut diperoleh dari buku, jurnal, situs internet serta

literatur lain yang berkaitan. Teori pustaka yang berkaitan dengan penelitian dalam skripsi ini meliputi :

- Tanaman Nilam
- Citra Digital meliputi citra biner, citra *grayscale*, dan citra warna.
- Ekstraksi Fitur meliputi fitur warna dan fitur luas.
- Jarak Euclidian yang akan digunakan dalam algoritma *Learning Vector Quantization* (LVQ).
- Jaringan Syaraf Tiruan (JST) meliputi karakteristik JST, arsitektur JST, jaringan dengan lapisan tunggal, jaringan dengan lapisan kompetitif, prinsip dasar pelatihan JST, proses algoritma pelatihan JST, belajar dengan pengawasan, dan algoritma JST.
- Learning Vector Quantization (LVQ) meliputi arsitektur LVQ dan algoritma LVQ.
- Evaluasi meliputi perhitungan akurasi.

3.1.2 Pengumpulan Data

Pengumpulan data pada penelitian ini menggunakan data primer. Data primer merupakan data yang didapatkan secara langsung dari sumber penelitian. Data yang digunakan berupa citra daun nilam, dimana daun nilam tersebut diambil dari perkebunan nilam di Kesamben, kabupaten Blitar.

Daun nilam yang digunakan merupakan daun dari jenis nilam Aceh. Daun diambil di bagian pucuk tanaman dengan usia 6 bulan. Daun nilam dengan kualitas baik maupun kurang baik masing-masing diambil sebanyak 50 daun. Setelah didapatkan, kemudian seluruh daun nilam tersebut disimpan kedalam bentuk citra digital dengan format file gambar JPG.

Citra digital yang digunakan diperoleh dari hasil pemindaian daun nilam tampak depan dengan resolusi 600 dpi (dots per inch) kemudian disimpan dengan ukuran 400x500 *pixels* dan 8x10 *pixels*. Ukuran citra yang lebih besar digunakan kedalam proses kerja sistem, sedangkan ukuran citra yang lebih kecil digunakan untuk perhitungan manual. Citra tersebut dihasilkan dari proses *cropping* dari citra asli terlebih dahulu, kemudian dilakukan proses *resize* serta dilakukan perubahan warna pada *background*. *Background* yang sebelumnya memiliki warna putih tidak

sempurna dan memiliki *noise*, kemudian dihilangkan sehingga warnanya menjadi putih sempurna. Beberapa proses tersebut dilakukan dengan menggunakan bantuan aplikasi *Photoshop CS6 Portable*.

3.1.3 Analisis Kebutuhan

Dalam tahap ini dilakukan analisa kebutuhan sesuai dengan kebutuhan sistem yang dibangun untuk dapat melakukan klasifikasi kualitas daun nilam sehingga sistem yang dibangun dapat berjalan dengan optimal. Secara keseluruhan, kebutuhan yang digunakan dalam pembuatan sistem ini meliputi:

1. Spesifikasi kebutuhan *hardware*, meliputi:
 - Laptop dengan *processor Intel(R) Core(TM) i3-2350M CPU @ 2.30GHz*
2.30 GHz
 - *Memory 2GB*
2. Spesifikasi kebutuhan *software*, meliputi:
 - *Windows 7 Ultimate 64-bit*
 - *NetBeans IDE 7.4*
 - *Java Development Kit (JDK) 7*
 - *Microsoft office 2013*
 - *Visual studio 2010*
 - *Photoshop CS6 Portable*

3.1.4 Perancangan Sistem

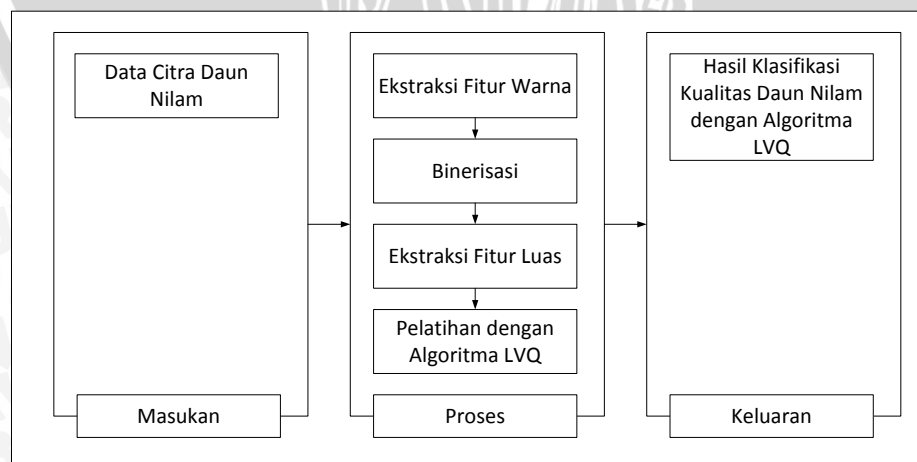
Perancangan sistem adalah tahap dimana dirancang suatu sistem yang mampu memenuhi semua kebutuhan fungsional aplikasi dalam penelitian ini. Teori-teori dari pustaka dan data yang digunakan digabungkan dengan ilmu yang didapat kemudian diimplementasikan untuk merancang sistem klasifikasi kualitas daun nilam. Hasil dari perancangan sistem ini akan menjadi dasar dari proses implementasi. Tahap-tahap yang ada di dalam perancangan antara lain adalah perancangan diagram alir sistem, perhitungan manual, serta perancangan antar muka.

3.1.4.1 Deskripsi Umum Sistem

Dalam penelitian ini, sistem yang dibangun merupakan sistem yang mengimplementasikan algoritma *Learning Vector Quantization* (LVQ) untuk dapat mengklasifikasi kualitas daun nilam berdasarkan ciri fisik daun nilam. Ciri fisik daun nilam yang digunakan adalah warna dan luas daun, sehingga dibutuhkan proses ekstraksi fitur warna dan luas dari citra daun nilam. Fitur yang telah didapat kemudian digunakan sebagai masukan pada pelatihan maupun pengujian dengan algoritma LVQ. Algoritma LVQ bekerja dengan mengklasifikasikan vektor-vektor input ke dalam kelas-kelas sesuai dengan jarak antara vektor-vektor input tersebut. 2 buah vektor input akan diletakkan oleh lapisan kompetitif ke dalam kelas yang sama jika kedua vektor input tersebut mendekati sama. Keluaran yang dihasilkan sistem berupa kelas “kualitas baik” atau “kualitas kurang baik”.

3.1.4.2 Diagram Blok Sistem

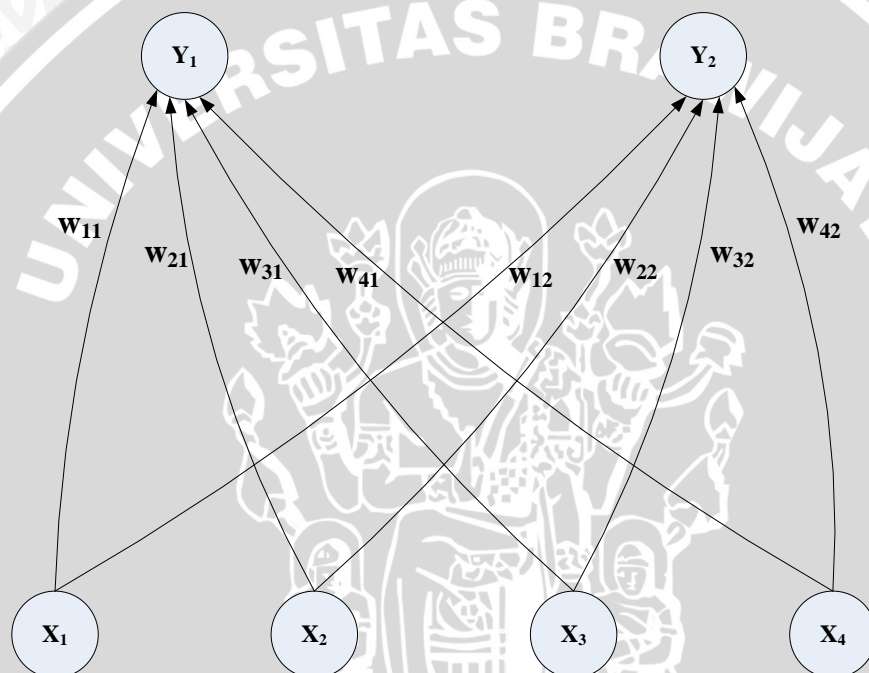
Diagram blok sistem menjelaskan cara kerja sistem yang akan dibuat mulai dari masukan, proses, sampai keluaran yang dihasilkan oleh sistem. Secara umum sistem bekerja dengan melakukan beberapa tahapan meliputi *input* citra daun nilam, ekstraksi fitur warna (rata-rata dari setiap warna merah, hijau, dan biru), binerisasi, ekstraksi fitur luas, pelatihan dengan algoritma LVQ, pengujian dengan algoritma LVQ, dan menghasilkan keluaran berupa hasil klasifikasi kualitas daun nilam. Gambar 3.2 merupakan diagram blok yang menggambarkan secara umum dari sistem yang akan dibangun.



Gambar 3.2 Diagram Blok Sistem Klasifikasi Kualitas Daun Nilam menggunakan Algoritma LVQ

3.1.4.3 Arsitektur Sistem

Arsitektur jaringan LVQ yang digunakan memiliki 3 layer yaitu layer *input*, layer bobot, dan layer *output*. Jumlah node yang terdapat pada layer *input* sebanyak 4 node sesuai dengan jumlah fitur yang digunakan yaitu rata-rata *red*, *green*, *blue*, dan luas. Sedangkan jumlah node yang terdapat pada layer *output* sebanyak 2 node sesuai dengan jumlah kelas yang ditentukan yaitu kelas “kualitas baik” dan “kualitas kurang baik”. Arsitektur jaringan LVQ untuk klasifikasi kualitas daun nilam ditunjukkan pada Gambar 3.3.



Gambar 3.3 Arsitektur Jaringan LVQ untuk Klasifikasi Kualitas Daun Nilam

3.1.5 Implementasi Sistem

Pada tahap ini dijelaskan bagaimana implementasi algoritma *Learning Vector Quantization* (LVQ) untuk klasifikasi kualitas daun nilam dengan citra digital yang mengacu pada perancangan sistem yang telah dilakukan sebelumnya. Implementasi sistem ini menggunakan bahasa pemrograman *Java*, *Netbeans IDE 7.4*, dan *tools* pendukung lainnya.

3.1.6 Pengujian dan Analisis

Pengujian sistem ini dilakukan agar dapat memastikan bahwa aplikasi yang telah dibangun dapat bekerja sesuai dengan spesifikasi dari kebutuhan yang telah ditentukan. Pengujian yang dilakukan adalah pengujian tingkat akurasi data dengan

membandingkan keluaran dari sistem dengan label yang diberikan oleh pakar menggunakan algoritma *Learning Vector Quantization* (LVQ). Terdapat beberapa skenario pengujian yang dilakukan meliputi:

1. Pengujian untuk mengetahui pengaruh nilai laju pembelajaran terhadap hasil akurasi.
2. Pengujian untuk mengetahui pengaruh nilai konstanta pengurang laju pembelajaran terhadap hasil akurasi.
3. Pengujian untuk mengetahui pengaruh nilai laju pembelajaran minimum terhadap hasil akurasi.
4. Pengujian untuk mengetahui pengaruh jumlah iterasi pembelajaran terhadap hasil akurasi.
5. Pengujian untuk mengetahui pengaruh perbandingan jumlah data latih dengan data uji terhadap hasil akurasi.
6. Pengujian untuk mengetahui hasil akurasi dari proses klasifikasi menggunakan *minimum distance*.

3.1.7 Kesimpulan dan Saran

Pengambilan kesimpulan dan saran dilakukan setelah semua tahapan, perancangan, implementasi algoritma *Learning Vector Quantization*, dan pengujian sistem terhadap algoritma yang digunakan telah selesai dilakukan. Kesimpulan dapat diperoleh dari hasil pengujian. Penulisan saran berguna untuk memberikan pertimbangan atas pengembangan metode dan aplikasi selanjutnya.

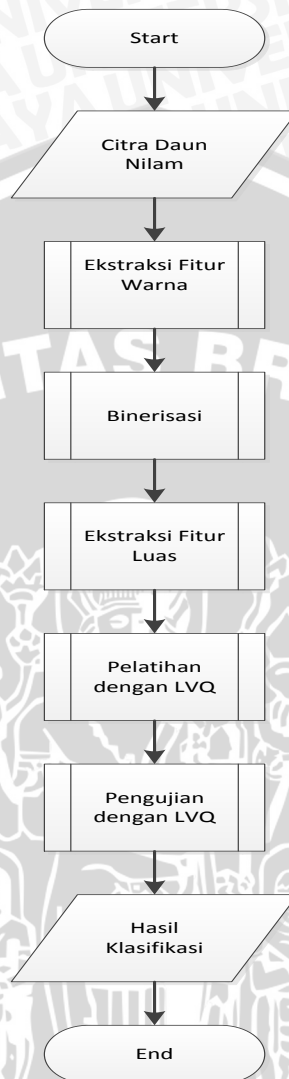
3.2 Perancangan

Bagian ini menjelaskan bagaimana merancang proses kerja sistem yang dibangun. Diagram alir digunakan untuk mempresentasikan beberapa proses kerja sistem.

3.2.1 Proses Kerja Sistem Secara Umum

Secara umum, proses kerja sistem ini terdiri dari beberapa tahapan meliputi *input* citra daun nilam, ekstraksi fitur warna, binerisasi, ekstraksi fitur luas, pelatihan dengan algoritma LVQ, pengujian dengan algoritma LVQ, dan keluaran

hasil klasifikasi kualitas daun nilam. Proses kerja yang dilakukan sistem secara umum ditunjukkan pada Gambar 3.4.



Gambar 3.4 Diagram Alir Proses Kerja Sistem Secara Umum

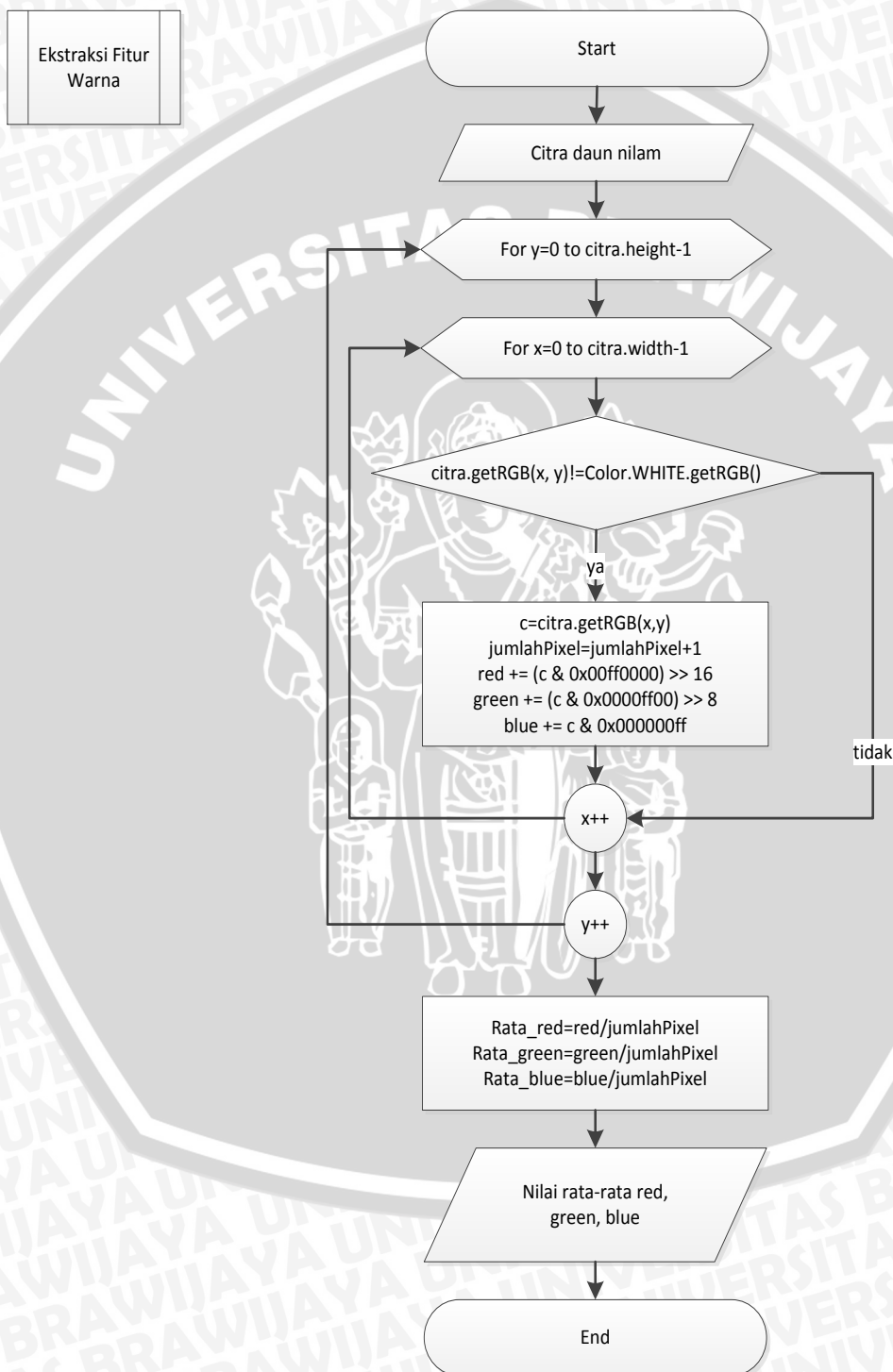
3.2.2 Proses Ekstraksi Fitur Warna

Proses ekstraksi fitur warna dalam sistem ini terdiri dari beberapa tahapan sebagai berikut:

1. *input* citra daun nilam berupa citra warna
2. Menghitung jumlah *pixel* serta menghitung masing-masing jumlah total nilai *red*, *green*, dan *blue* dengan ketentuan nilai *pixel* pada koordinat $f(x,y)$ tidak sama dengan nilai warna putih. Warna putih mempresentasikan *background*, sedangkan warna lain mempresentasikan objek daun pada citra.

3. Menghitung masing-masing nilai rata-rata *red*, *green*, *blue* dengan membagi masing-masing jumlah total nilai *red*, *green*, *blue* dengan jumlah *pixel*.
4. Mengembalikan nilai rata-rata *red*, *green*, *blue*.

Gambar 3.5 menunjukkan diagram alir proses ekstraksi fitur warna.



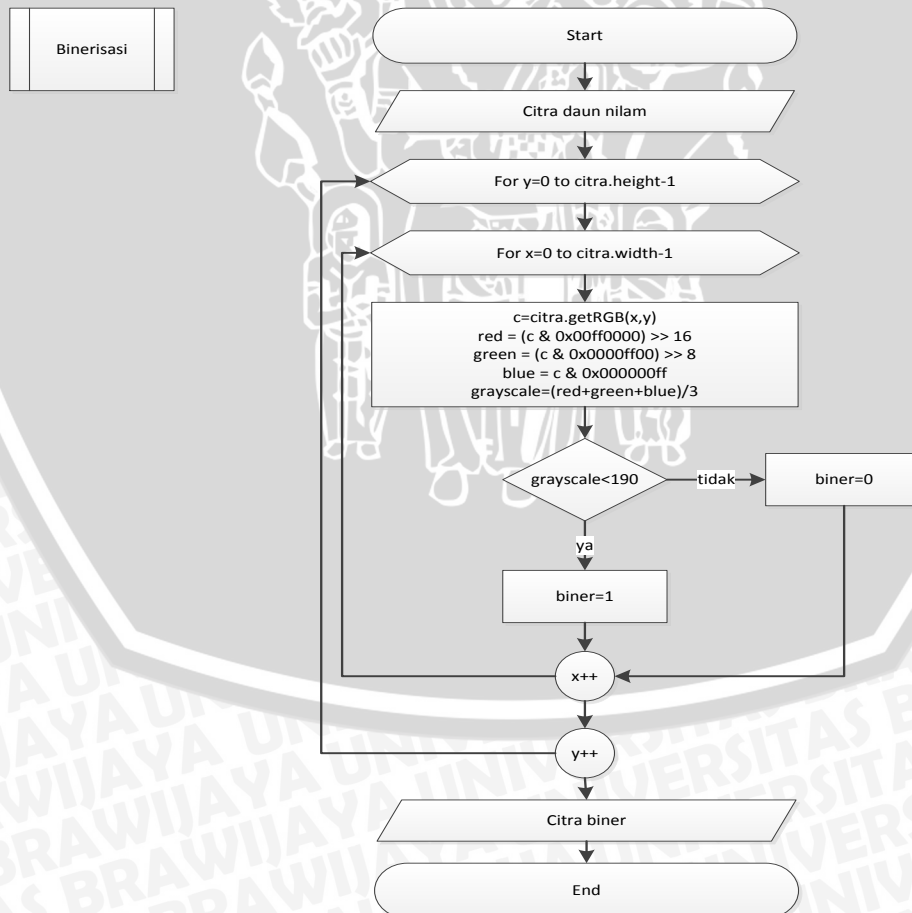
Gambar 3.5 Diagram Alir Proses Ekstraksi Fitur Warna

3.2.3 Proses Binerisasi

Proses binerisasi dilakukan dengan menetapkan nilai *threshold*=190. Angka 190 digunakan karena dapat memisahkan objek daun nilam atau *foreground* dengan *background* lebih sempurna jika dibandingkan dengan menggunakan angka 140, 150, 160, 170, 180, 200, 210, dan 220 setelah dilakukan beberapa kali percobaan. Proses binerisasi dalam sistem ini terdiri dari beberapa tahapan sebagai berikut:

1. *input* citra daun nilam berupa citra warna
2. Mengubah nilai *pixel* pada koordinat $f(x,y)$ menjadi sama dengan nilai total penjumlahan nilai *red*, *green*, *blue* pada koordinat $f(x,y)$ dibagi tiga kemudian disimpan dalam variabel *grayscale*. Setelah itu, melakukan proses *thresholding* dengan batas nilai 190 sehingga diperoleh nilai biner=1 jika nilai *grayscale* kurang dari 190 dan nilai biner=0 jika nilai *grayscale* lebih dari 190.
3. Mengembalikan citra biner.

Gambar 3.6 menunjukkan diagram alir proses binerisasi.



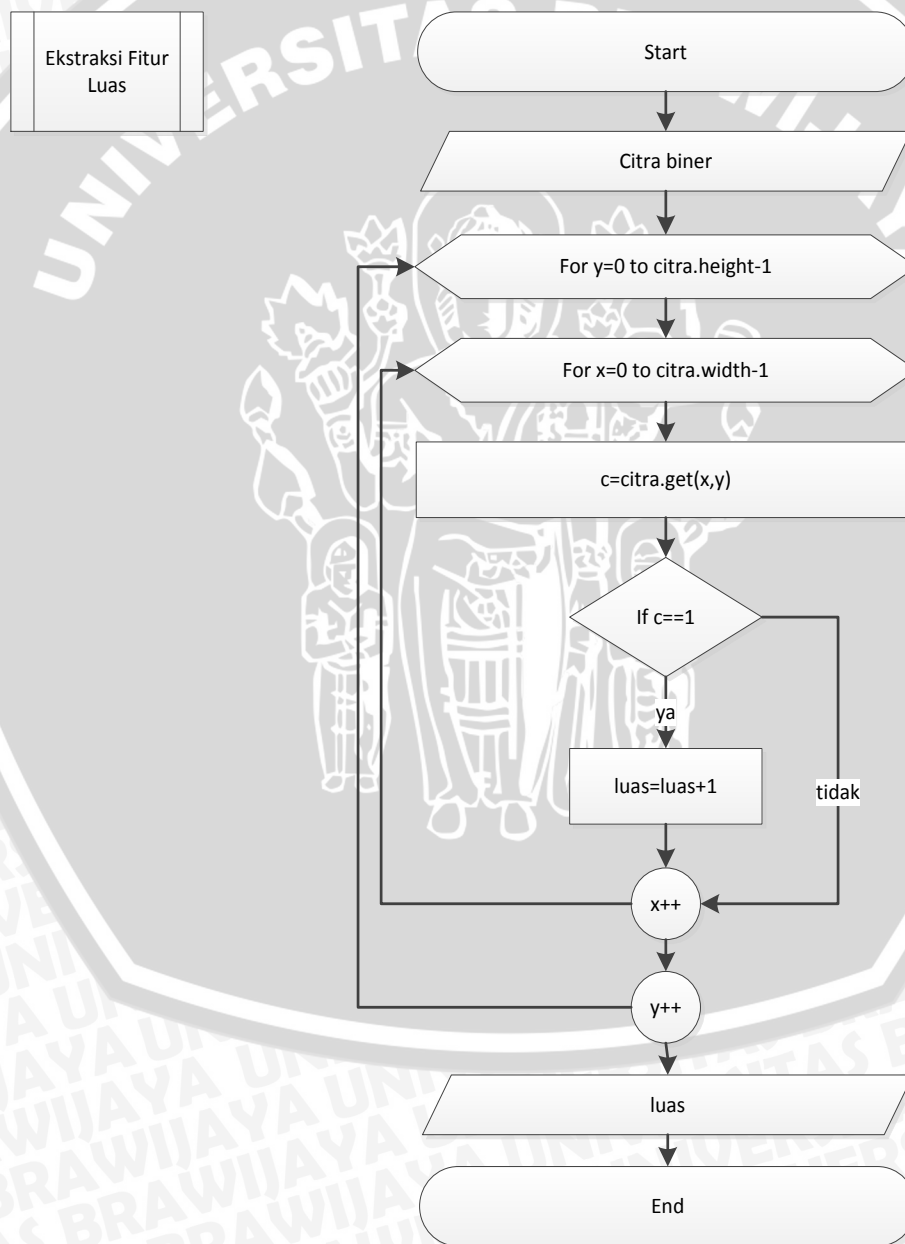
Gambar 3.6 Diagram Alir Proses Binerisasi

3.2.4 Proses Ekstraksi Fitur Luas

Proses ekstraksi fitur luas dalam sistem ini terdiri dari beberapa tahapan sebagai berikut:

1. *input* citra daun nilam berupa citra biner.
2. Menghitung luas permukaan daun dengan menambahkan nilai 1 pada variabel luas jika setiap nilai *pixel* pada koordinat $f(x,y)$ sama dengan 1.
3. Mengembalikan nilai luas.

Gambar 3.7 menunjukkan diagram alir proses ekstraksi fitur luas.



Gambar 3.7 Diagram Alir Proses Ekstraksi Fitur Luas



3.2.5 Proses Pelatihan dengan Algoritma LVQ

Proses pelatihan dengan algoritma LVQ merupakan proses dimana pada setiap data latih dilakukan proses inialisasi vektor pelatihan hingga proses perubahan bobot. Proses ini dilakukan setelah fitur warna dan fitur luas dari seluruh data citra diperoleh. Proses pelatihan dengan algoritma LVQ dalam sistem ini terdiri dari beberapa tahapan sebagai berikut:

1. *Input* data latih. Data latih yang digunakan berisi data id citra, rata-rata *red*, *green*, *blue*, luas, dan target.
2. Inialisasi vektor bobot dengan nilai random, inialisasi laju pembelajaran, laju pembelajaran minimum, dan iterasi maksimum.
3. Melakukan langkah 4 sampai dengan langkah 7 dengan ketentuan ketika laju pembelajaran belum mencapai laju pembelajaran minimum atau iterasi belum mencapai iterasi maksimum. Kondisi pemberhentian ini diharapkan telah mencapai konvergensi pada perubahan bobot yang dihasilkan, dimana selisih bobot yang dihasilkan dengan bobot sebelumnya adalah mendekati nol.
4. Untuk setiap vektor pelatihan x , lakukan langkah 5 sampai dengan langkah 6
5. Menentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.
6. *Update* w_j dengan ketentuan sebagai berikut:

Jika $T=C_j$, maka

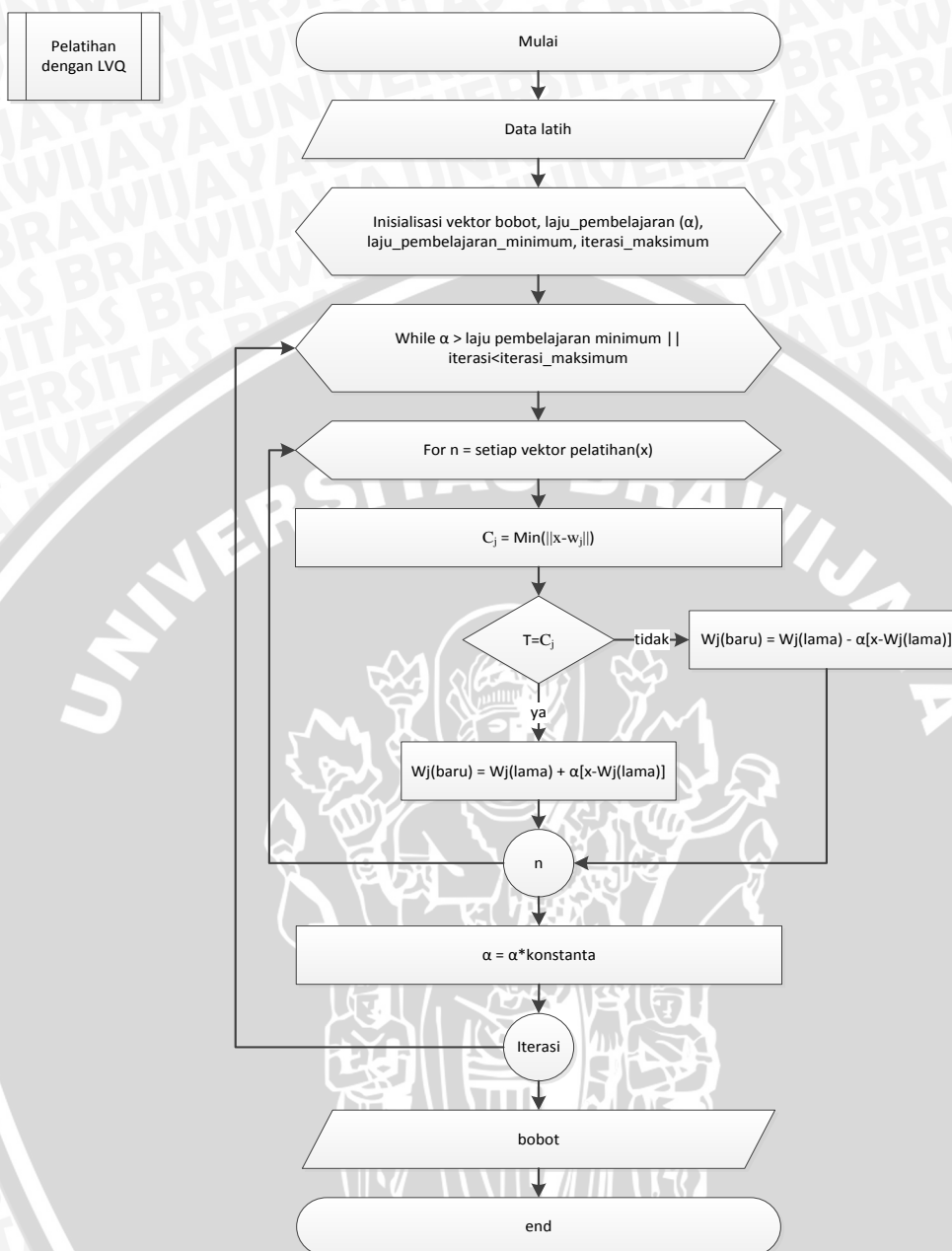
$$w_j(\text{baru}) = w_j(\text{lama}) + \alpha[x - w_j(\text{lama})]$$

Jika $T \neq C_j$, maka

$$w_j(\text{baru}) = w_j(\text{lama}) - \alpha[x - w_j(\text{lama})]$$

7. Kurangi laju pembelajaran dengan mengalikan laju pembelajaran dengan suatu konstanta.
8. Mengembalikan vektor bobot yang dihasilkan.

Gambar 3.8 menunjukkan diagram alir proses pelatihan dengan Algoritma LVQ.



Gambar 3.8 Diagram Alir Proses Pelatihan dengan Algoritma LVQ

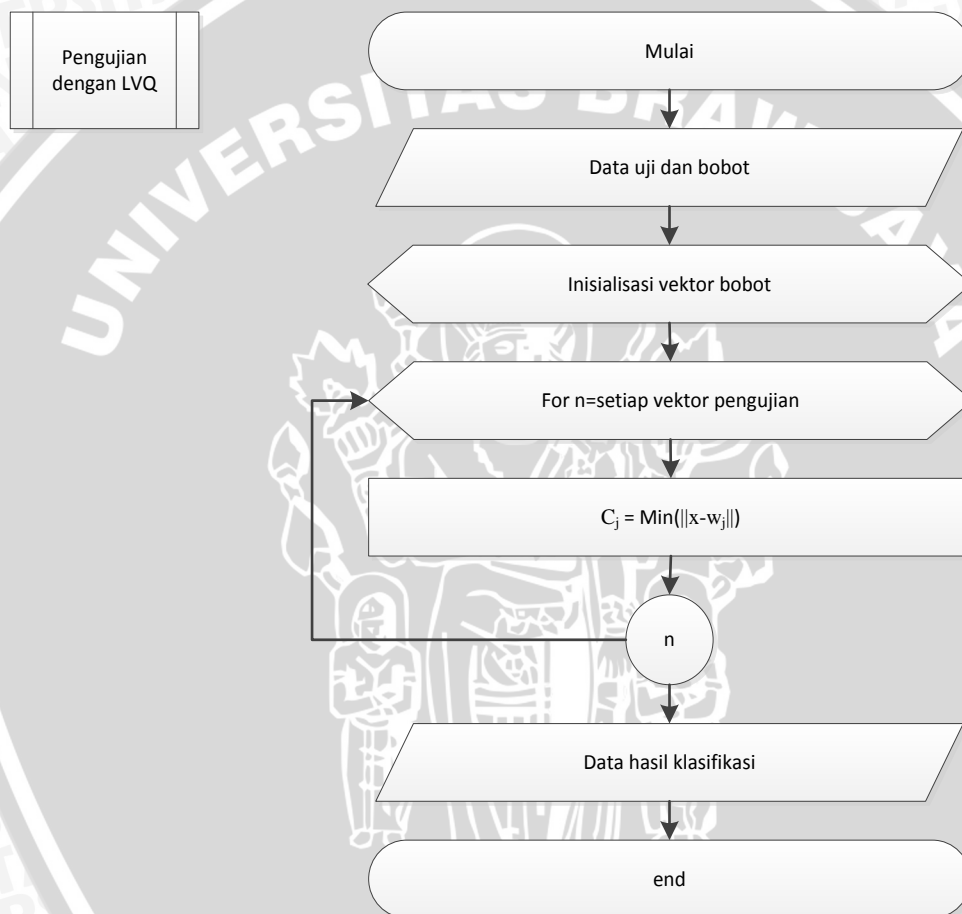
3.2.6 Proses Pengujian dengan LVQ

Proses pengujian dengan algoritma LVQ merupakan proses dimana pada setiap data uji dilakukan proses inisialisasi vektor pengujian hingga proses penentuan kelas. Proses pengujian dengan algoritma LVQ dalam sistem ini terdiri dari beberapa tahapan sebagai berikut:

1. *Input* data uji dan bobot dari hasil pelatihan. Data uji yang digunakan berisi data id citra, rata-rata red, green, blue, luas, dan target.

2. Inisialisasi vektor bobot dengan bobot yang diinputkan.
3. Untuk setiap vektor pengujian x , tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.
4. Mengembalikan data hasil klasifikasi yang berisi id citra, target, dan kelas hasil klasifikasi.

Gambar 3.9 menunjukkan diagram alir proses pengujian dengan Algoritma LVQ.



Gambar 3.9 Diagram Alir Proses Pengujian dengan Algoritma LVQ

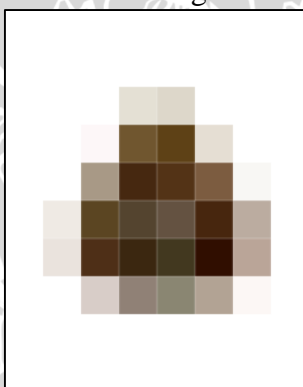
3.3 Perhitungan Manual

Pada bagian ini dijelaskan proses manual dari ekstraksi nilai RGB, ekstraksi nilai rata-rata *red*, *green*, *blue*, binerisasi, dan pengambilan nilai luas permukaan dari citra daun nilam. Citra daun nilam yang digunakan sebagai contoh dalam perhitungan manual ini adalah citra daun nilam yang dibuat dengan ukuran 8×10 *pixels*. Gambar 3.10 menunjukkan contoh citra daun nilam dengan ukuran asli

sebesar 1600x2000 *pixels*. Sedangkan Gambar 3.11 merupakan contoh citra daun nilam dengan ukuran 8x10 *pixels* yang khusus digunakan dalam perhitungan manual. Kedua citra tersebut merupakan citra dengan id yang sama yaitu citra dengan id “1-1”.



Gambar 3.10 Contoh Citra dengan ukuran 400x500 *Pixels*



Gambar 3.11 Contoh Citra dengan Ukuran 8x10 *Pixels*

3.3.1 Ekstraksi Nilai RGB

Ekstraksi nilai RGB bertujuan untuk menampilkan nilai-nilai RGB setiap *pixel* pada citra. Tabel 3.1 mempresentasikan hasil ekstraksi nilai RGB dari contoh citra.

Tabel 3.1 Hasil Ekstraksi Nilai RGB

R	255	255	255	255	255	255	255	255
G	255	255	255	255	255	255	255	255
B	255	255	255	255	255	255	255	255
R	255	255	255	255	255	255	255	255
G	255	255	255	255	255	255	255	255
B	255	255	255	255	255	255	255	255
R	255	255	255	228	221	255	255	255
G	255	255	255	224	215	255	255	255

B	255	255	255	212	202	255	255	255
R	255	255	253	112	94	229	255	255
G	255	255	247	86	65	222	255	255
B	255	255	248	47	22	211	255	255
R	255	255	168	70	83	124	248	255
G	255	255	153	40	51	92	247	255
B	255	255	134	16	22	64	244	255
R	255	239	91	84	100	71	187	255
G	255	234	69	68	82	38	172	255
B	255	228	34	47	65	14	160	255
R	255	234	78	59	66	48	186	255
G	255	227	47	39	56	14	165	255
B	255	221	23	16	31	0	152	255
R	255	255	216	144	138	178	252	255
G	255	255	205	129	134	163	247	255
B	255	255	200	118	114	148	245	255
R	255	255	255	255	255	255	255	255
G	255	255	255	255	255	255	255	255
B	255	255	255	255	255	255	255	255
R	255	255	255	255	255	255	255	255
G	255	255	255	255	255	255	255	255
B	255	255	255	255	255	255	255	255

3.3.2 Ekstraksi Nilai Rata-Rata Red, Green, dan Blue

Dalam ekstraksi nilai rata-rata *red*, *green*, dan *blue* perlu diketahui jumlah nilai *red*, *green*, *blue*, dan jumlah *pixel* terlebih dahulu. Jumlah dari masing-masing fitur tersebut ditentukan dari kondisi *pixel* yang tidak berwarna putih ($R=255$, $G=255$, $B=255$). Berdasarkan Tabel 3.1, *pixel* dengan kondisi tidak berwarna putih meliputi *pixel* dengan koordinat $f(3,2)$, $f(4,2)$, $f(2,3)$, $f(3,3)$, $f(4,3)$, $f(5,3)$, $f(2,4)$, $f(3,4)$, $f(4,4)$, $f(5,4)$, $f(6,4)$, $f(1,5)$, $f(2,5)$, $f(3,5)$, $f(4,5)$, $f(5,5)$, $f(6,5)$, $f(1,6)$, $f(2,6)$, $f(3,6)$, $f(4,6)$, $f(5,6)$, $f(6,6)$, $f(2,7)$, $f(3,7)$, $f(4,7)$, $f(5,7)$, dan $f(6,7)$. Tabel 3.2 mempresentasikan hasil ekstraksi nilai rata-rata *red*, *green*, dan *blue*. Secara rinci, proses perhitungan rata-rata *red*, *green*, dan *blue* masing-masing dapat dijabarkan sesuai persamaan (2-3), (2-4), dan (2-5) sebagai berikut:

$$\text{RataRata}_R = \frac{\text{Jumlah}_R}{\text{Jumlah pixel}} = \frac{4201}{28} = 150,04$$

$$\text{RataRata}_G = \frac{\text{Jumlah}_G}{\text{Jumlah pixel}} = \frac{3731}{28} = 133,25$$

$$\text{RataRata}_B = \frac{\text{Jumlah}_B}{\text{Jumlah pixel}} = \frac{3238}{28} = 115,64$$

Tabel 3.2 Hasil Ekstraksi Nilai Rata-Rata *Red*, *Green*, dan *Blue*

Jumlah			Jumlah <i>pixel</i>	Rata-rata		
R	G	B		R	G	B
4201.0	3731.0	3238.0	28.0	150.04	133.25	115.64

3.3.3 Binerisasi Citra Daun Nilam

Binerisasi citra dapat dilakukan dengan melakukan konversi citra warna (RGB) menjadi citra *grayscale* terlebih dahulu. Untuk menghasilkan citra *grayscale* tersebut dapat dilakukan dengan cara menjumlahkan ketiga nilai unsur warna dasar, kemudian dirata-rata sesuai dengan persamaan (2-1). Tabel 3.3 mempresentasikan hasil konversi citra RGB menjadi citra *grayscale*. Contoh perhitungan konversi citra RGB menjadi citra *grayscale* terhadap *pixel* $f(3,2)$ adalah sebagai berikut:

$$f(3,2) = \frac{(R + G + B)}{3} = \frac{(228 + 224 + 212)}{3} = 221$$

Tabel 3.3 Hasil Konversi Citra RGB ke Citra *Grayscale*

255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	221	212	255	255	255
255	255	249	81	60	220	255	255
255	255	151	42	52	93	246	255
255	233	64	66	82	41	173	255
255	227	49	38	51	20	167	255
255	255	207	130	128	163	248	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Langkah selanjutnya adalah melakukan konversi citra *grayscale* ke citra biner dengan proses *thresholding*. Proses *thresholding* dilakukan dengan menentukan suatu batas level sebesar 190. Nilai *pixel* yang kurang dari 190 diubah

menjadi 1, sedangkan yang lebih dari 190 diubah menjadi 0 (nol). Tabel 3.4 mempresentasikan hasil konversi citra *grayscale* ke citra biner.

Tabel 3.4 Hasil Konversi Citra *Grayscale* ke citra Biner

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	1	0
0	0	1	1	1	1	1	0
0	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

3.3.4 Ekstraksi Nilai Luas Permukaan

Ekstraksi nilai luas permukaan dapat dilakukan dengan menjumlahkan *pixel* yang bernilai 1 yang berada di dalam tepi objek dari citra biner yang telah diperoleh. Berdasarkan Tabel 3.4, terdapat *pixel* yang bernilai 1 sebanyak 19 *pixel*, sehingga luas permukaan daun yang didapat pun sebesar 19.

3.3.5 Pelatihan dengan Algoritma LVQ

Pelatihan dengan algoritma LVQ dalam perhitungan manual ini memerlukan data latih sebanyak 8 data. 8 data tersebut terdiri dari 4 data dari kelas 1 (kualitas baik) dan 4 data dari kelas 2 (kualitas kurang baik). Tabel 3.5 mempresentasikan data latih untuk perhitungan manual dalam pelatihan dengan Algoritma LVQ.

Tabel 3.5 Data Latih untuk Perhitungan Manual

Id_citra	Rata_R	Rata_G	Rata_B	Luas	Target
1.1	150,04	133,25	115,64	19	1
1.2	163,48	145,93	125,93	16	1
1.3	163,67	145,71	118	15	1
1.4	150,16	124,79	101,26	28	1
2.2	102,8	118,06	95,07	46	2
2.5	126,22	141,15	101,37	31	2

2.24	129,33	142,19	108,44	26	2
2.35	142,21	153,76	135,14	18	2

Pelatihan dengan algoritma LVQ dilakukan sesuai dengan teori algoritma jaringan LVQ yang telah dibahas di bab sebelumnya. Dari data latih diatas, maka dapat dilakukan pelatihan sebagai berikut:

- Menginisialisasi vektor bobot dengan nilai random

w_1	0,169542709	0,897071019	0,439710711	0,396042895
w_2	0,350376452	0,569058306	0,43665082	0,772582723

Menginisialisasi laju pembelajaran $\alpha = 0,1$

Menginisialisasi laju pembelajaran minimum = 0,0001

Menginisialisasi iterasi=0

Menginisialisasi iterasi maksimum = 10

- Iterasi < iterasi maksimum atau laju pembelajaran > laju pembelajaran minimum, maka lakukan langkah berikutnya.
- Untuk vektor pelatihan x pertama, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$D_1 = \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2}$$

$$= \sqrt{(150,04 - 0,16)^2 + (133,25 - 0,89)^2 + (115,64 - 0,43)^2 + (19 - 0,39)^2}$$

$$= 231,5074$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$D_2 = \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2}$$

$$= \sqrt{(150,04 - 0,35)^2 + (133,25 - 0,56)^2 + (115,64 - 0,43)^2 + (19 - 0,77)^2}$$

$$= 231,5497$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- $T=1$ dan $C_1=1$, maka perbarui bobot w_1 sebagai berikut:

$$w_1(\text{baru}) = w_1(\text{lama}) + \alpha[x - w_1(\text{lama})]$$

$$= \begin{pmatrix} 0,16 \\ 0,89 \\ 0,43 \\ 0,39 \end{pmatrix} + 0,1 \left[\begin{pmatrix} 150,04 \\ 133,25 \\ 115,64 \\ 19 \end{pmatrix} - \begin{pmatrix} 0,16 \\ 0,89 \\ 0,43 \\ 0,39 \end{pmatrix} \right]$$

$$= \begin{pmatrix} 15,15658844 \\ 14,13236392 \\ 11,95973964 \\ 2,256438605 \end{pmatrix}$$

Sehingga diperoleh bobot baru sebagai berikut

w_1	15,15658844	14,13236392	11,95973964	2,256438605
w_2	0,350376452	0,569058306	0,43665082	0,772582723

- Untuk vektor pelatihan x kedua, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$D_1 = \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2}$$

$$= \sqrt{(163,48 - 15,15)^2 + (145,93 - 14,13)^2 + (125,93 - 11,95)^2 + (16 - 2,25)^2}$$

$$= 229,2347204$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$D_2 = \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2}$$

$$= \sqrt{(163,48 - 0,35)^2 + (145,93 - 0,56)^2 + (125,93 - 0,43)^2 + (16 - 0,77)^2}$$

$$= 252,4312429$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- $T=1$ dan $C_1=1$, maka perbarui bobot w_1 sebagai berikut:

$$w_1(\text{baru}) = w_1(\text{lama}) + \alpha[x - w_1(\text{lama})]$$

$$= \begin{pmatrix} 15,15 \\ 14,13 \\ 11,95 \\ 2,25 \end{pmatrix} + 0,1 \left[\begin{pmatrix} 163,48 \\ 145,93 \\ 125,93 \\ 16 \end{pmatrix} - \begin{pmatrix} 15,15 \\ 14,13 \\ 11,95 \\ 2,25 \end{pmatrix} \right]$$

$$= \begin{pmatrix} 29,98892959 \\ 27,31212753 \\ 23,35676568 \\ 3,630794745 \end{pmatrix}$$

Sehingga diperoleh bobot baru sebagai berikut

w_1	29,98892959	27,31212753	23,35676568	3,630794745
w_2	0,350376452	0,569058306	0,43665082	0,772582723

- Untuk vektor pelatihan x ketiga, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak euclidean adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$D_1 = \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2}$$

$$= \sqrt{(163,67 - 29,98)^2 + (145,71 - 27,31)^2 + (118 - 23,35)^2 + (15 - 3,63)^2}$$

$$= 202,4235298$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$D_2 = \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2}$$

$$= \sqrt{(163,67 - 0,35)^2 + (145,71 - 0,56)^2 + (118 - 0,43)^2 + (15 - 0,77)^2}$$

$$= 248,5211316$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- $T=1$ dan $C_1=1$, maka perbarui bobot w_1 sebagai berikut:

$$w_1(\text{baru}) = w_1(\text{lama}) + \alpha[x - w_1(\text{lama})]$$

$$= \begin{pmatrix} 29,98 \\ 27,31 \\ 23,35 \\ 3,63 \end{pmatrix} + 0,1 \left[\begin{pmatrix} 163,67 \\ 145,71 \\ 118 \\ 15 \end{pmatrix} - \begin{pmatrix} 29,98 \\ 27,31 \\ 23,35 \\ 3,63 \end{pmatrix} \right]$$

$$= \begin{pmatrix} 43,35703663 \\ 39,15191477 \\ 32,82108911 \\ 4,76771527 \end{pmatrix}$$

Sehingga diperoleh bobot baru sebagai berikut

w_1	43,35703663	39,15191477	32,82108911	4,76771527
w_2	0,350376452	0,569058306	0,43665082	0,772582723

- Untuk vektor pelatihan x keempat, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$D_1 = \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2}$$

$$= \sqrt{(150,16 - 43,35)^2 + (124,79 - 39,15)^2 + (101,26 - 32,82)^2 + (28 - 4,76)^2}$$

$$= 154,8043223$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$D_2 = \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2}$$

$$= \sqrt{(150,16 - 0,35)^2 + (124,79 - 0,56)^2 + (101,26 - 0,43)^2 + (28 - 0,77)^2}$$

$$= 220,8629567$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- $T=1$ dan $C_1=1$, maka perbarui bobot w_1 sebagai berikut:

$$w_1(\text{baru}) = w_1(\text{lama}) + \alpha[x - w_1(\text{lama})]$$

$$= \begin{pmatrix} 43,35 \\ 39,15 \\ 32,82 \\ 4,76 \end{pmatrix} + 0,1 \left[\begin{pmatrix} 150,16 \\ 124,79 \\ 101,26 \\ 28 \end{pmatrix} - \begin{pmatrix} 43,35 \\ 39,15 \\ 32,82 \\ 4,76 \end{pmatrix} \right]$$

$$= \begin{pmatrix} 54,03733297 \\ 47,7157233 \\ 39,6649802 \\ 7,090943743 \end{pmatrix}$$

Sehingga diperoleh bobot baru sebagai berikut

w_1	54,03733297	47,7157233	39,6649802	7,090943743
w_2	0,350376452	0,569058306	0,43665082	0,772582723

- Untuk vektor pelatihan x kelima, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$D_1 = \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2}$$

$$= \sqrt{(102,8 - 54,03)^2 + (118,06 - 47,71)^2 + (95,07 - 39,66)^2 + (46 - 7,09)^2}$$

$$= 109,131782$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$D_2 = \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2}$$

$$= \sqrt{(102,8 - 0,35)^2 + (118,06 - 0,56)^2 + (95,07 - 0,43)^2 + (46 - 0,77)^2}$$

$$= 187,8857014$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- $T=2$ dan $C_1=1$, maka perbarui bobot w_1 sebagai berikut:

$$w_1(\text{baru}) = w_1(\text{lama}) - \alpha[x - w_1(\text{lama})]$$

$$= \begin{pmatrix} 54,03 \\ 47,71 \\ 39,66 \\ 7,09 \end{pmatrix} - 0,1 \left[\begin{pmatrix} 102,8 \\ 118,06 \\ 95,07 \\ 46 \end{pmatrix} - \begin{pmatrix} 54,03 \\ 47,71 \\ 39,66 \\ 7,09 \end{pmatrix} \right]$$

$$= \begin{pmatrix} 49,16106627 \\ 40,68129563 \\ 34,12447822 \\ 3,200038118 \end{pmatrix}$$

Sehingga diperoleh bobot baru sebagai berikut

w_1	49,16106627	40,68129563	34,12447822	3,200038118
w_2	0,350376452	0,569058306	0,43665082	0,772582723

- Untuk vektor pelatihan x keenam, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$D_1 = \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2}$$

$$= \sqrt{(126,22 - 49,16)^2 + (141,15 - 40,68)^2 + (101,37 - 34,12)^2 + (31 - 3,20)^2}$$

$$= 146,0371114$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$D_2 = \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2}$$

$$= \sqrt{(126,22 - 0,35)^2 + (141,15 - 0,56)^2 + (101,37 - 0,43)^2 + (31 - 0,77)^2}$$

$$= 216,1189511$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- $T=2$ dan $C_1=1$, maka perbarui bobot w_1 sebagai berikut:

$$\begin{aligned} w_1(\text{baru}) &= w_1(\text{lama}) - \alpha[x - w_1(\text{lama})] \\ &= \begin{pmatrix} 49,16 \\ 40,68 \\ 34,12 \\ 3,20 \end{pmatrix} - 0,1 \left[\begin{pmatrix} 126,22 \\ 141,15 \\ 101,37 \\ 31 \end{pmatrix} - \begin{pmatrix} 49,16 \\ 40,68 \\ 34,12 \\ 3,20 \end{pmatrix} \right] \\ &= \begin{pmatrix} 41,4551729 \\ 30,63442519 \\ 27,39992604 \\ 0,420041929 \end{pmatrix} \end{aligned}$$

Sehingga diperoleh bobot baru sebagai berikut

w_1	41,4551729	30,63442519	27,39992604	0,420041929
w_2	0,350376452	0,569058306	0,43665082	0,772582723

- Untuk vektor pelatihan x ketujuh, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$\begin{aligned} D_1 &= \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2} \\ &= \sqrt{(129,33 - 41,45)^2 + (142,19 - 30,63)^2 + (108,44 - 27,39)^2 + (26 - 0,42)^2} \\ &= 165,4945901 \end{aligned}$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$\begin{aligned} D_2 &= \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2} \\ &= \sqrt{(129,33 - 0,35)^2 + (142,19 - 0,56)^2 + (108,44 - 0,43)^2 + (26 - 0,77)^2} \\ &= 216,1189511 \end{aligned}$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- $T=2$ dan $C_1=1$, maka perbarui bobot w_1 sebagai berikut:

$$w_1(\text{baru}) = w_1(\text{lama}) - \alpha[x - w_1(\text{lama})]$$

$$\begin{aligned}
 &= \begin{pmatrix} 41,45 \\ 30,63 \\ 27,39 \\ 0,42 \end{pmatrix} - 0,1 \left[\begin{pmatrix} 129,33 \\ 142,19 \\ 108,44 \\ 26 \end{pmatrix} - \begin{pmatrix} 41,45 \\ 30,63 \\ 27,39 \\ 0,42 \end{pmatrix} \right] \\
 &= \begin{pmatrix} 32,66769018 \\ 19,47886771 \\ 19,29591864 \\ -2,137953878 \end{pmatrix}
 \end{aligned}$$

Sehingga diperoleh bobot baru sebagai berikut

w_1	32,66769018	19,47886771	19,29591864	-2,137953878
w_2	0,350376452	0,569058306	0,43665082	0,772582723

- Untuk vektor pelatihan x kedelapan, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$\begin{aligned}
 D_1 &= \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2} \\
 &= \sqrt{(142,21 - 32,66)^2 + (153,76 - 19,47)^2 + (135,14 - 19,29)^2 + (18 - -2,13)^2} \\
 &= 209,4190261
 \end{aligned}$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$\begin{aligned}
 D_2 &= \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2} \\
 &= \sqrt{(142,21 - 0,35)^2 + (153,76 - 0,56)^2 + (135,14 - 0,43)^2 + (18 - 0,77)^2} \\
 &= 249,0650389
 \end{aligned}$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- $T=2$ dan $C_1=1$, maka perbarui bobot w_1 sebagai berikut:

$$w_1(\text{baru}) = w_1(\text{lama}) - \alpha[x - w_1(\text{lama})]$$

$$\begin{aligned}
 &= \begin{pmatrix} 32,66 \\ 19,47 \\ 19,29 \\ -2,13 \end{pmatrix} - 0,1 \left[\begin{pmatrix} 142,21 \\ 153,76 \\ 135,14 \\ 18 \end{pmatrix} - \begin{pmatrix} 32,66 \\ 19,47 \\ 19,29 \\ -2,13 \end{pmatrix} \right] \\
 &= \begin{pmatrix} 21,7134592 \\ 16,050754477 \\ 7,711510507 \\ -4,151749265 \end{pmatrix}
 \end{aligned}$$

Sehingga diperoleh bobot baru sebagai berikut

w_1	21,7134592	6,050754477	7,711510507	-4,151749265
w_2	0,350376452	0,569058306	0,43665082	0,772582723

- Kurangi laju pembelajaran dengan cara

$$\alpha = \alpha * 0,3 = 0,1 * 0,3 = 0,03$$
- Iterasi < iterasi maksimum atau laju pembelajaran > laju pembelajaran minimum, maka lakukan kembali langkah seperti diatas untuk setiap vektor pelatihan x .
- Setelah pelatihan berhenti pada iterasi ke-6, bobot terakhir yang dihasilkan adalah sebagai berikut

w_1	27,91972588	6,368009234	9,093588042	-5,992343183
w_2	0,350376452	0,569058306	0,43665082	0,772582723

3.3.6 Pengujian dengan Algoritma LVQ

Pengujian dengan algoritma LVQ dalam perhitungan manual ini memerlukan data uji sebanyak 5 data. 5 data tersebut terdiri dari 3 data dari kelas 1 (kualitas baik) dan 2 data dari kelas 2 (kualitas kurang baik). Tabel 3.6 mempresentasikan data latih untuk perhitungan manual dalam pelatihan dengan Algoritma LVQ.

Tabel 3.6 Data Uji untuk Perhitungan Manual

Id_citra	Rata_R	Rata_G	Rata_B	Luas	Target
1.1	160,29	140,29	124,62	15	1
1.2	157,5	143,58	124,23	17	1
1.3	148,82	134,86	118,75	21	1
1.4	126,31	141,86	112,69	26	2
2.2	125,98	141,16	113,19	29	2

Dari bobot yang telah dihasilkan di proses pelatihan dan berdasarkan data uji diatas, maka dapat dilakukan pengujian sebagai berikut:

- Untuk vektor pengujian x pertama, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.
Jarak ke w_1 dihitung dengan cara

$$D_1 = \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2}$$

$$= \sqrt{(160,29 - 27,91)^2 + (140,29 - 6,36)^2 + (124,62 - 9,09)^2 + (15 - -5,99)^2}$$

$$= 221,9099354$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$D_2 = \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2}$$

$$= \sqrt{(160,29 - 0,35)^2 + (140,29 - 0,56)^2 + (124,62 - 0,43)^2 + (15 - 0,77)^2}$$

$$= 246,4275722$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- Untuk vektor pengujian x kedua, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$D_1 = \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2}$$

$$= \sqrt{(157,5 - 27,91)^2 + (143,58 - 6,36)^2 + (124,23 - 9,09)^2 + (17 - -5,99)^2}$$

$$= 222,2683492$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$D_2 = \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2}$$

$$= \sqrt{(157,5 - 0,35)^2 + (143,58 - 0,56)^2 + (124,23 - 0,43)^2 + (17 - 0,77)^2}$$

$$= 246,4472682$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- Untuk vektor pengujian x ketiga, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak euclidean adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$D_1 = \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2}$$

$$= \sqrt{(148,82 - 27,91)^2 + (134,86 - 6,36)^2 + (118,75 - 9,09)^2 + (21 - -5,99)^2}$$

$$= 209,475973$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$D_2 = \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2}$$

$$= \sqrt{(148,82 - 0,35)^2 + (134,86 - 0,56)^2 + (118,75 - 0,43)^2 + (21 - 0,77)^2}$$

$$= 233,4191148$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- Untuk vektor pengujian x keempat, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$D_1 = \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2}$$

$$= \sqrt{(126,31 - 27,91)^2 + (141,86 - 6,36)^2 + (112,69 - 9,09)^2 + (26 - -5,99)^2}$$

$$= 199,4854686$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$D_2 = \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2}$$

$$= \sqrt{(126,31 - 0,35)^2 + (141,86 - 0,56)^2 + (112,69 - 0,43)^2 + (26 - 0,77)^2}$$

$$= 221,5089027$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_1 = 1$.

- Untuk vektor pengujian x kelima, tentukan kelas C_j dimana $\|x-w_j\|$ dengan persamaan jarak *Euclidean* adalah minimum.

Jarak ke w_1 dihitung dengan cara

$$D_1 = \sqrt{\sum_{i=1}^4 (x_i - w_{i1})^2}$$

$$= \sqrt{(125,98 - 27,91)^2 + (141,16 - 6,36)^2 + (113,19 - 9,09)^2 + (29 - -5,99)^2}$$

$$= 199,612688$$

Sedangkan jarak ke w_2 dihitung dengan cara

$$D_2 = \sqrt{\sum_{i=1}^4 (x_i - w_{i2})^2}$$

$$= \sqrt{(125,98 - 0,35)^2 + (141,16 - 0,56)^2 + (113,19 - 0,43)^2 + (29 - 0,77)^2}$$

$$= 221,4920315$$

Maka dipilih jarak terdekat yaitu jarak ke w_1 , sehingga $C_i = 1$.

- Berdasarkan proses pengujian yang telah dilakukan, maka diperoleh data hasil klasifikasi perhitungan manual yang ditunjukkan pada Tabel 3.7.

Tabel 3.7 Data Hasil Klasifikasi Perhitungan Manual

Id_citra	Rata_R	Rata_G	Rata_B	Luas	Target	C (Kelas)
1.1	160,29	140,29	124,62	15	1	1
1.2	157,5	143,58	124,23	17	1	1
1.3	148,82	134,86	118,75	21	1	1
1.4	126,31	141,86	112,69	26	2	1
2.2	125,98	141,16	113,19	29	2	1

3.3.7 Perhitungan Akurasi

Setelah pengujian dengan algoritma LVQ dilakukan, maka perhitungan akurasi dapat dilakukan. Akurasi dihitung sesuai dengan persamaan (2-7).

Perhitungan tersebut dapat dijabarkan sebagai berikut:

$$\text{Akurasi} = \frac{\text{jumlah data yang relevan}}{\text{jumlah seluruh data yang ada}} \times 100 = \frac{3}{5} \times 100 = 60$$

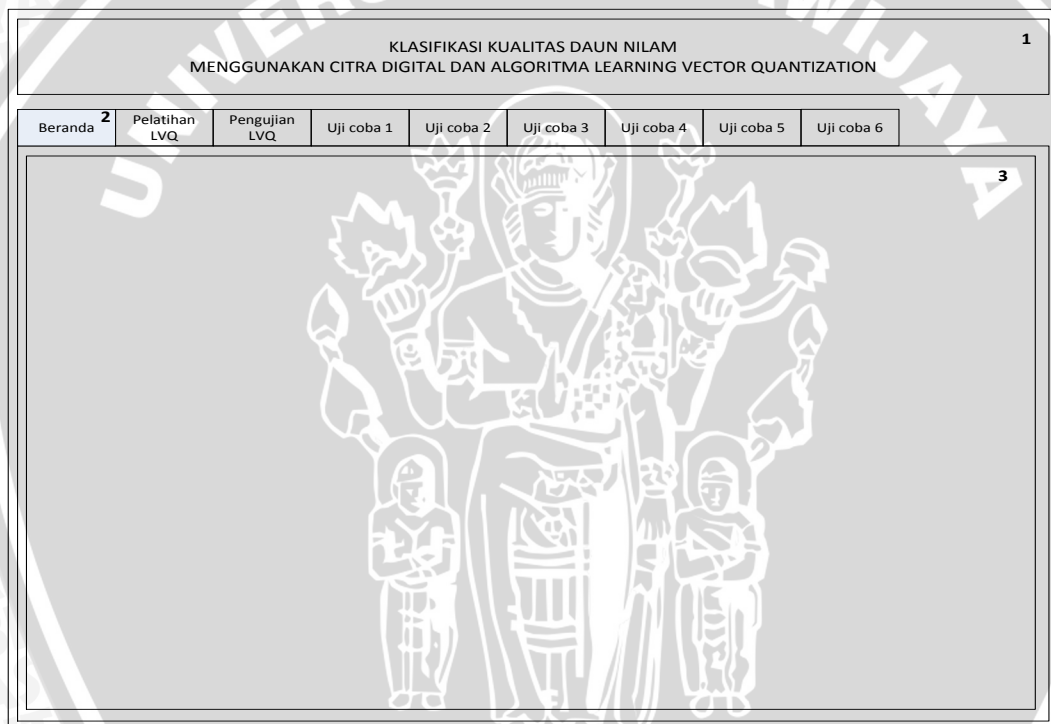
Akurasi yang diperoleh dinyatakan dalam persentase sehingga hasil akurasi yang diperoleh ditulis menjadi 60%.

3.4 Perancangan antarmuka

Rancangan antarmuka dari sistem yang dibuat terdiri dari 9 halaman utama. Halaman-halaman tersebut meliputi halaman beranda, pelatihan LVQ, pengujian LVQ, uji coba 1, uji coba 2, uji coba 3, uji coba 4, uji coba 5, dan uji coba 6.

34.1 Antarmuka Halaman Beranda

Halaman beranda merupakan halaman yang akan muncul pertama kali saat aplikasi dijalankan. Secara keseluruhan, dari halaman utama ini langsung dapat dilihat bahwa terdapat 8 *tabbed pane* untuk mengakses 8 halaman yang berbeda. Rancangan antarmuka halaman beranda ditunjukkan pada gambar 3.12.



Gambar 3.12 Antarmuka Halaman Beranda

Keterangan Gambar 3.12:

1. Judul aplikasi
2. *Tabbed pane* untuk menampilkan halaman beranda
3. Panel untuk menampilkan informasi tentang aplikasi yang dibuat

3.4.2 Antarmuka Halaman Pelatihan LVQ

Halaman pelatihan merupakan halaman yang akan menampilkan informasi yang berkaitan dalam proses pelatihan menggunakan algoritma LVQ. Informasi

yang ditampilkan tersebut meliputi data latih dan parameter yang digunakan dalam pelatihan. Data latih ditampilkan dalam bentuk citra dan data ekstraksi fitur. Rancangan antarmuka halaman pelatihan LVQ ditunjukkan pada gambar 3.13.

The screenshot shows a software interface for LVQ training. At the top, a title bar reads 'KLASIFIKASI KUALITAS DAUN NILAM MENGGUNAKAN CITRA DIGITAL DAN ALGORITMA LEARNING VECTOR QUANTIZATION'. Below this is a tabbed menu with options: 'Beranda', 'Pelatihan LVQ', 'Pengujian LVQ', and six 'Uji coba' (test) options. The main area is divided into several panels:

- Panel 3:** A control panel with 'Jumlah data latih' (training data count) and 'Tampilkan data' (display data) with a dropdown menu (4).
- Panel 5 & 6:** Two image display areas labeled 'Citra RGB' and 'Citra Biner'.
- Panel 7:** A table area labeled 'Fitur' for displaying extracted features.
- Panel 8:** A parameter settings panel with input fields for 'Laju pembelajaran' (learning rate), 'Pengurang aju pembelajaran' (learning rate decay), 'Laju pembelajaran minimum' (minimum learning rate), and 'Iterasi maksimum' (maximum iterations). It also includes a radio button for 'Data hasil uji coba' (test results data) and 'Proses' (process) and 'Default' buttons.
- Panel 12 & 13:** 'Bobot awal' (initial weights) section with input fields for 'W1' and 'W2'.
- Panel 14 & 15:** 'Bobot yang dihasilkan' (generated weights) section with input fields for 'W1' and 'W2'.

Gambar 3.13 Rancangan Antarmuka Halaman Pelatihan LVQ

Keterangan Gambar 3.13:

1. Judul aplikasi
2. *Tabbed pane* untuk menampilkan halaman pelatihan LVQ
3. Panel untuk menampilkan informasi data latih
4. *Combo box* untuk memilih citra data latih yang ingin ditampilkan berdasarkan id citranya.
5. Panel untuk menampilkan citra warna daun nilam
6. Panel unuk menampilkan citra biner daun nilam
7. Tabel untuk menampilkan fitur hasil ekstraksi
8. Panel untuk menampilkan parameter yang digunakan dalam proses pelatihan. Di dalamnya terdapat *text field* yang memungkinkan pengguna dalam memberi masukan pada beberapa parameter tersebut.
9. *Radio button* untuk memilih data hasil uji coba sebagai data latih yang akan diproses.

10. *Button* untuk memulai proses pelatihan
11. *Button* untuk memasukkan parameter dengan nilai default atau nilai yang telah ditetapkan.
12. Tabel untuk menampilkan bobot awal kelas 1 (kualitas baik)
13. Tabel untuk menampilkan bobot awal kelas 2 (kualitas kurang baik)
14. Tabel untuk menampilkan bobot yang dihasilkan dari proses pelatihan pada kelas 1 (kualitas baik)
15. Tabel untuk menampilkan bobot yang dihasilkan dari proses pelatihan pada kelas 2 (kualitas kurang baik)

3.4.3 Antarmuka Halaman Pengujian LVQ

Halaman pengujian merupakan halaman yang menampilkan sebuah proses klasifikasi dari sebuah daun nilam dengan parameter pelatihan yang telah ditentukan. Rancangan antarmuka halaman pengujian LVQ ditunjukkan pada gambar 3.14.

The screenshot shows the LVQ testing interface with the following components:

- 1**: Title bar containing the text "KLASIFIKASI KUALITAS DAUN NILAM MENGGUNAKAN CITRA DIGITAL DAN ALGORITMA LEARNING VECTOR QUANTIZATION".
- 2**: A tabbed pane with tabs for "Beranda", "Pelatihan LVQ", "Pengujian LVQ", and "Uji coba 1" through "Uji coba 6".
- 3**: A form area for data input, including "Jumlah data latih" (with a numeric input field **4**), "Tampilkan data" (with a numeric input field **5**), and a "Browse" button **6**.
- 7**: A display area for the "Citra RGB" (input image).
- 8**: A display area for the "Citra Biner" (processed image).
- 9**: A "Fitur" (feature) display area.
- 10**: A "Target" (target) display area.
- 11**: A "Masuk kelas" (class membership) display area.
- 12**: An "Akurasi" (accuracy) display area.

Gambar 3.14 Rancangan Antarmuka Halaman Pengujian LVQ

Keterangan Gambar 3.14:

1. Judul aplikasi
2. *Tabbed pane* untuk menampilkan halaman pengujian LVQ

3. Panel untuk menampilkan informasi data uji
4. *Combo box* untuk memilih citra data uji yang ingin ditampilkan sekaligus diuji berdasarkan id citranya.
5. *Text field* untuk menampilkan id citra yang telah dipilih melalui pencarian.
6. *Button* untuk melakukan pencarian citra yang ingin ditampilkan sekaligus diuji
7. Panel untuk menampilkan citra warna daun nilam
8. Panel unuk menampilkan citra biner daun nilam
9. Tabel untuk menampilkan fitur hasil ekstraksi
10. Panel untuk menampilkan target citra daun nilam yang diuji
11. Panel untuk menampilkan hasil klasifikasi
12. Panel untuk menampilkan akurasi

3.4.5 Antarmuka Halaman Uji Coba 1

Halaman uji coba 1 merupakan halaman yang menampilkan kasus pengujian untuk mengetahui pengaruh nilai laju pembelajaran terhadap hasil akurasi. Rancangan antarmuka halaman uji coba 1 ditunjukkan pada Gambar 3.15.

KLASIFIKASI KUALITAS DAUN NILAM								
MENGUNAKAN CITRA DIGITAL DAN ALGORITMA LEARNING VECTOR QUANTIZATION								
Beranda	Pelatihan LVQ	Pengujian LVQ	Uji coba 1 ²	Uji coba 2	Uji coba 3	Uji coba 4	Uji coba 5	Uji coba 6
								1
								3
								4
								5

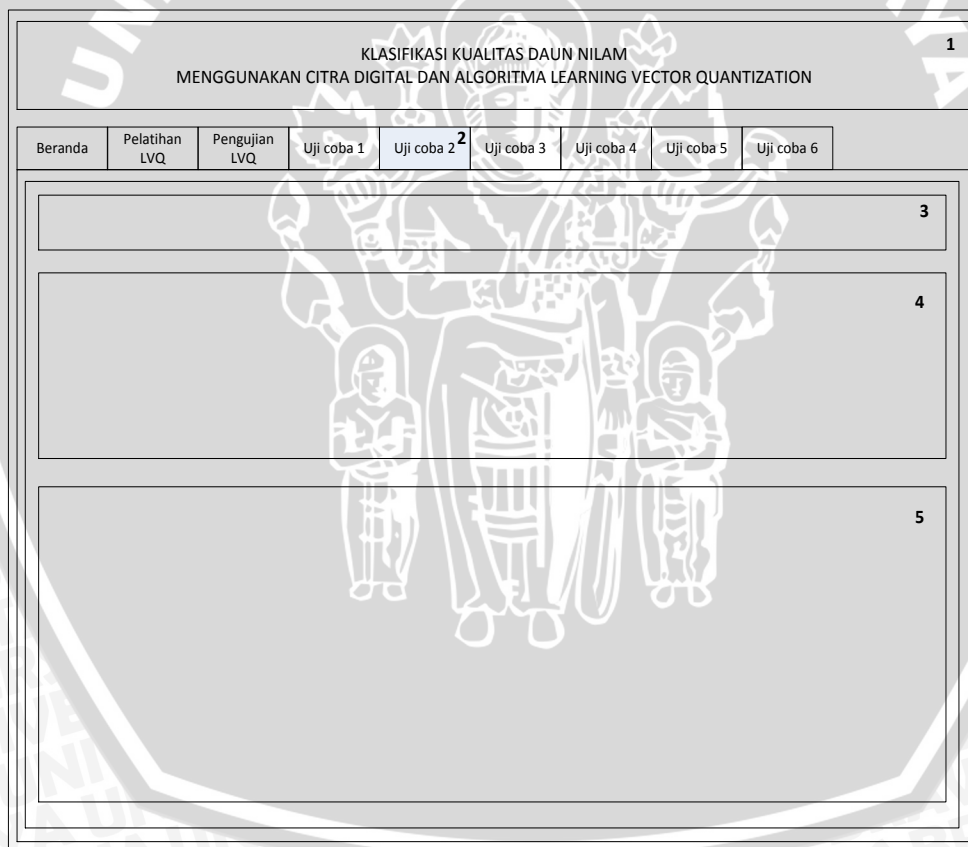
Gambar 3.15 Rancangan Antarmuka Halaman Uji Coba 1

Keterangan Gambar 3.15:

1. Judul sistem
2. *Tabbed pane* untuk menampilkan halaman uji coba 1
3. Judul pengujian
4. Panel untuk menampilkan parameter pelatihan yang telah ditentukan
5. Tabel untuk menampilkan data hasil pengujian

3.4.6 Antarmuka Halaman Uji Coba 2

Halaman uji coba 2 merupakan halaman yang menampilkan kasus pengujian untuk mengetahui pengaruh nilai konstanta pengurang laju pembelajaran terhadap hasil akurasi. Rancangan antarmuka halaman uji coba 2 ditunjukkan pada Gambar 3.16.



Gambar 3.16 Rancangan Antarmuka Halaman Uji Coba 2

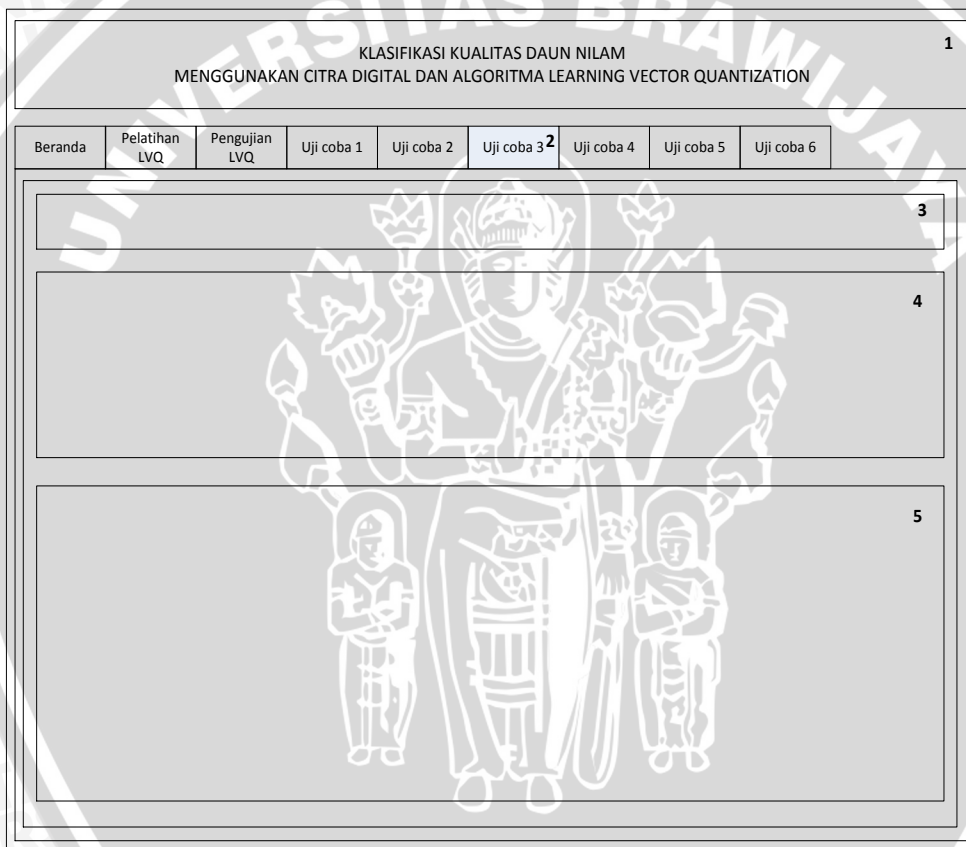
Keterangan Gambar 3.16:

1. Judul sistem
2. *Tabbed pane* untuk menampilkan halaman uji coba 2

3. Judul pengujian
4. Panel untuk menampilkan parameter pelatihan yang telah ditentukan
5. Tabel untuk menampilkan data hasil pengujian

3.4.7 Antarmuka Halaman Uji Coba 3

Halaman uji coba 3 merupakan halaman yang menampilkan kasus pengujian untuk mengetahui pengaruh nilai laju pembelajaran minimum terhadap hasil akurasi. Rancangan antarmuka halaman uji coba 3 ditunjukkan pada Gambar 3.17.



Gambar 3.17 Rancangan Antarmuka Halaman Uji Coba 3

Keterangan Gambar 3.17:

1. Judul sistem
2. *Tabbed pane* untuk menampilkan halaman uji coba 3
3. Judul pengujian
4. Panel untuk menampilkan parameter pelatihan yang telah ditentukan
5. Tabel untuk menampilkan data hasil pengujian

3.4.6 Antarmuka Halaman Uji Coba 4

Halaman uji coba 4 merupakan halaman yang menampilkan pengujian untuk mengetahui pengaruh jumlah iterasi pembelajaran terhadap hasil akurasi. Rancangan antarmuka halaman uji coba 4 ditunjukkan pada Gambar 3.18.



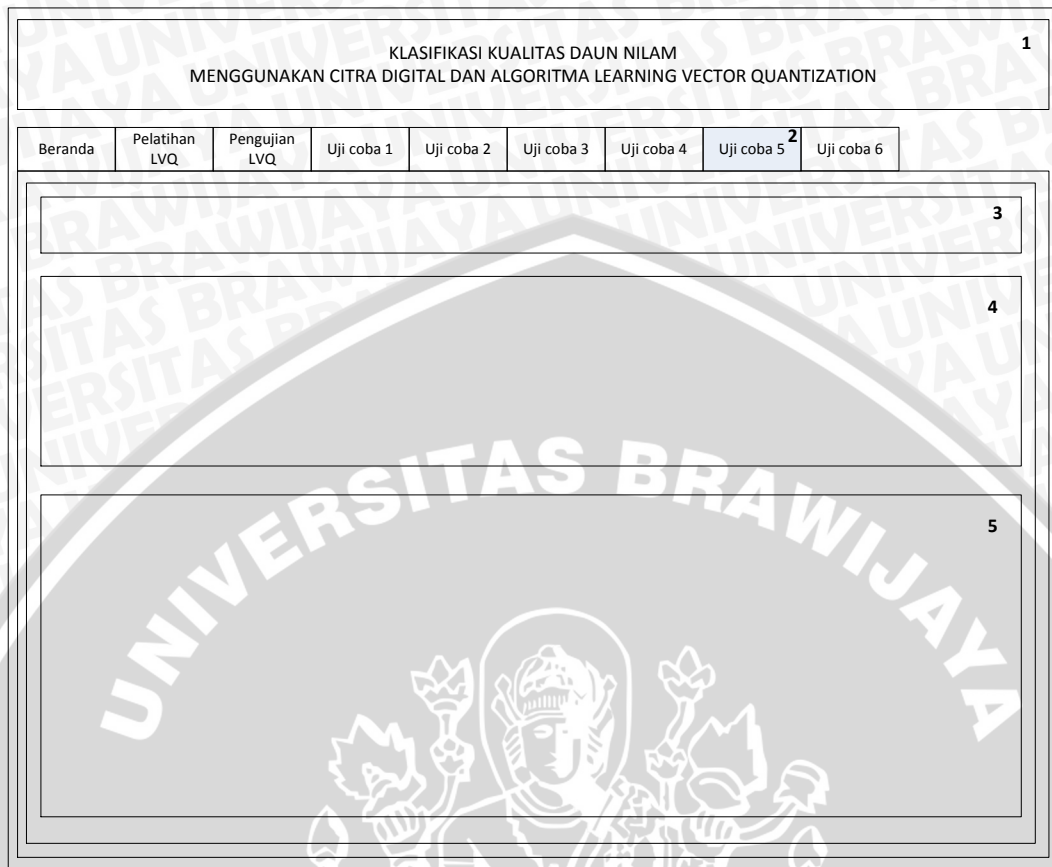
Gambar 3.18 Rancangan Antarmuka Halaman Uji Coba 4

Keterangan Gambar 3.18:

1. Judul sistem
2. *Tabbed pane* untuk menampilkan halaman uji coba 4
3. Judul pengujian
4. Panel untuk menampilkan parameter pelatihan yang telah ditentukan
5. Tabel untuk menampilkan data hasil pengujian

3.4.7 Antarmuka Halaman Uji Coba 5

Halaman uji coba 5 merupakan halaman yang menampilkan kasus pengujian untuk mengetahui pengaruh perbandingan jumlah data latih dengan data uji terhadap hasil akurasi. Rancangan antarmuka halaman uji coba 5 ditunjukkan pada Gambar 3.19.



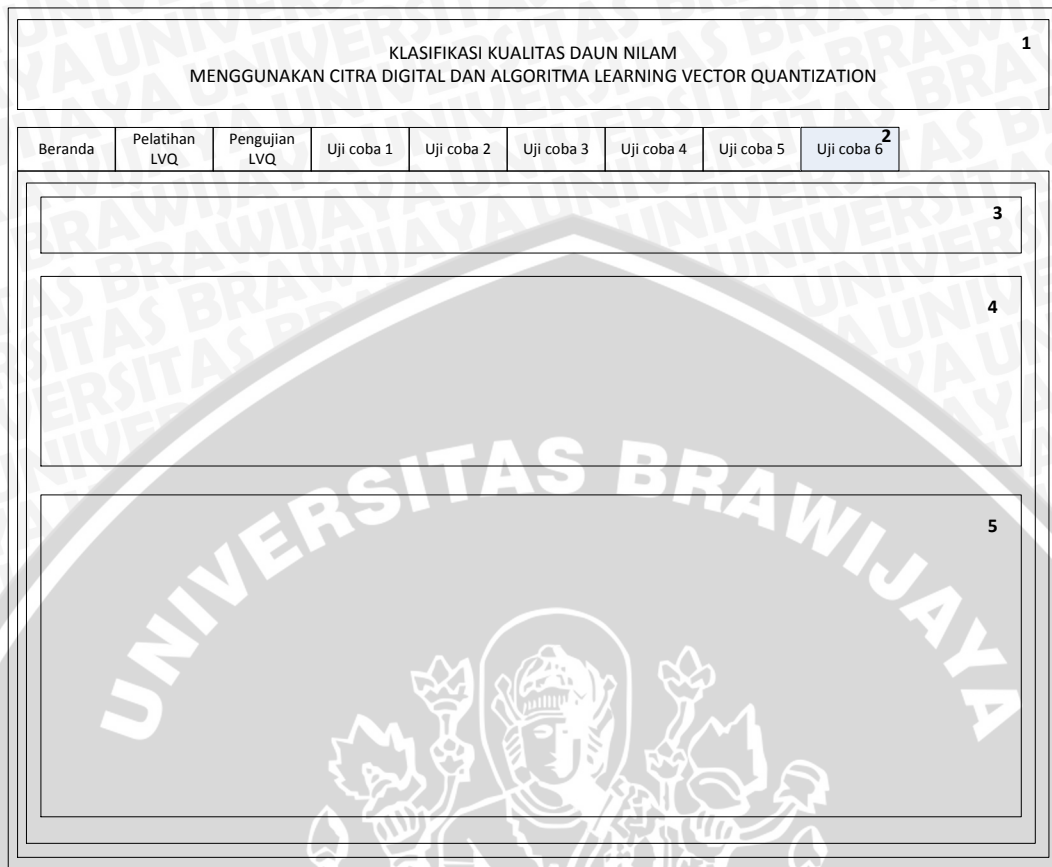
Gambar 3.19 Rancangan Antarmuka Halaman Uji Coba 5

Keterangan Gambar 3.19:

1. Judul sistem
2. *Tabbed pane* untuk menampilkan halaman uji coba 5
3. Judul pengujian
4. Panel untuk menampilkan parameter pelatihan yang telah ditentukan
5. Tabel untuk menampilkan data hasil pengujian

3.4.8 Antarmuka Halaman Uji Coba 6

Halaman uji coba 6 merupakan halaman yang menampilkan kasus pengujian untuk mengetahui hasil akurasi dari proses klasifikasi menggunakan *minimum distance*. Rancangan antarmuka halaman uji coba 6 ditunjukkan pada Gambar 3.20.



Gambar 3.20 Rancangan Antarmuka Halaman Uji Coba 6

Keterangan Gambar 3.20:

1. Judul sistem
2. *Tabbed pane* untuk menampilkan halaman uji coba 5
3. Judul pengujuan
4. Panel untuk menampilkan parameter pelatihan yang telah ditentukan
5. Tabel untuk menampilkan data hasil pengujuan

BAB IV IMPLEMENTASI

Bab ini menjelaskan mengenai lingkungan implementasi, implementasi algoritma dan implementasi antarmuka dalam sistem klasifikasi kualitas daun nilam menggunakan citra digital dan algoritma Learning Vector Quantization.

4.1 Lingkungan Implementasi

Pada sub bab ini akan membahas lingkungan perangkat keras dan perangkat lunak yang digunakan dalam proses pengembangan sistem klasifikasi kualitas daun nilam menggunakan citra digital dan algoritma Learning Vector Quantization.

4.1.1 Lingkungan Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pengembangan sistem yang dibangun dalam penelitian ini ditunjukkan pada Tabel 4.1.

Tabel 4.1 Lingkungan Perangkat Keras

Perangkat Keras	Keterangan
<i>Processor</i>	<i>Intel(R) Core(TM) i3-2350M CPU @ 2.30GHz 2.30 GHz</i>
<i>Memory RAM</i>	2 GB
<i>System type</i>	64-bit <i>Operating System</i>

4.1.2 Lingkungan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem yang dibangun dalam penelitian ini ditunjukkan pada Tabel 4.2.

Tabel 4.2 Lingkungan Perangkat Lunak

Perangkat Lunak	Keterangan
Sistem Operasi	<i>Windows 7 Ultimate 64-bit</i>
Bahasa Pemrograman	Java
<i>Tools</i>	<i>NetBeans IDE 7.4, Photoshop CS6 Portable, Microsoft office 2013, Visual studio 2010</i>

4.2 Implementasi Program

Sub bab ini akan menjelaskan tentang implementasi program yang digunakan dalam pengembangan sistem klasifikasi kualitas daun nilam menggunakan citra digital dan algoritma Learning Vector Quantization. Beberapa implementasi program yang akan dibahas dalam sub bab ini meliputi implementasi fungsi ekstraksi fitur warna, binerisasi, ekstraksi fitur luas, dan algoritma *Learning Vector Quantization*.

4.2.1 Ekstraksi Fitur Warna

Ekstraksi fitur warna digunakan untuk mendapatkan fitur rata-rata nilai warna merah, rata-rata nilai warna hijau, dan rata-rata nilai warna biru dari citra daun nilam. Implementasi fungsi ekstraksi fitur warna dapat dilihat pada *Source Code 4.1*.

```
1 public void setFiturWarna() {
2     int red=0, green=0, blue=0;
3     double jumlahpixel=0;
4     int c = citra.getRGB(0,0);
5     for(int y=0; y<citra.getHeight(); y++){
6         for(int x=0; x<citra.getWidth(); x++){
7             if(citra.getRGB(x, y)!=Color.WHITE.getRGB()){
8                 jumlahpixel++;
9                 c = citra.getRGB(x,y);
10                red+= (c & 0x00ff0000) >> 16;
11                green+= (c & 0x0000ff00) >> 8;
12                blue+= c & 0x000000ff;
13            }
14        }
15    }
16    rata_r=red/jumlahpixel;
17    rata_g=green/jumlahpixel;
18    rata_b=blue/jumlahpixel;
19 }
```

Source Code 4.1 Fungsi Ekstraksi Fitur Warna

Keterangan *Source Code 4.1*:

Baris 1 : Sebuah fungsi bernama “setFiturWarna”.

Baris 2-4 : Deklarasi variabel yang digunakan untuk algoritma ekstraksi fitur warna.

Baris 5 : Perulangan dengan kondisi variabel y sama dengan 0 hingga kurang dari tinggi citra.

Baris 6 : Perulangan dengan kondisi variabel x sama dengan 0 hingga kurang dari lebar citra

- Baris 7 : Pengecekan dengan kondisi dimana nilai RGB *pixel* citra pada koordinat $f(x,y)$ tidak sama dengan nilai warna putih.
- Baris 8 : Penambahan jumlah *pixel*.
- Baris 9 : Pengambilan nilai RGB *pixel* citra pada koordinat $f(x,y)$ yang disimpan kedalam variabel *c*.
- Baris 10 : Pengambilan sekaligus penambahan nilai warna merah yang disimpan kedalam variabel *red*.
- Baris 11 : Pengambilan sekaligus penambahan nilai warna hijau yang disimpan kedalam variabel *green*.
- Baris 12 : Pengambilan sekaligus penambahan nilai warna biru yang disimpan kedalam variabel *biru*.
- Baris 16 : Perhitungan rata-rata nilai warna merah yang disimpan kedalam variabel *rata_r*.
- Baris 17 : Perhitungan rata-rata nilai warna hijau yang disimpan kedalam variabel *rata_g*.
- Baris 18 : Perhitungan rata-rata nilai warna biru yang disimpan kedalam variabel *rata_b*.

4.2.2 Binerisasi

Binerisasi digunakan untuk mendapatkan citra biner, yaitu citra yang berwarna hitam dan putih. Implementasi fungsi binerisasi dapat dilihat pada *Source Code 4.2*.

```

1 public void setCitraBiner() throws IOException{
2     int red=0, green=0, blue=0, gray=0, biner=0;
3     for(int y=0; y<citra.getHeight(); y++){
4         for(int x=0; x<citra.getWidth(); x++){
5             int c = citra.getRGB(x,y);
6             red = (c & 0x00ff0000) >> 16;
7             green = (c & 0x0000ff00) >> 8;
8             blue = c & 0x000000ff;
9             gray = (int) Math.round((red+green+blue)/3);
10            if(gray<190) biner=255;
11            else biner=0;
12            Color cl= new Color(biner,biner,biner);
13            citra.setRGB(x, y, cl.getRGB());
14        }
15    }
16 }
17 }

```

Source Code 4.2 Fungsi Binerisasi

Keterangan *Source Code* 4.2:

- Baris 1 : Sebuah fungsi bernama “setCitraBiner” .
- Baris 2 : Deklarasi variabel yang digunakan untuk algoritma binerisasi.
- Baris 3 : Perulangan dengan kondisi variabel y sama dengan 0 hingga kurang dari tinggi citra.
- Baris 4 : Perulangan dengan kondisi variabel x sama dengan 0 hingga kurang dari lebar citra
- Baris 5 : Pengambilan nilai RGB *pixel* citra pada koordinat $f(x,y)$ yang disimpan kedalam variabel c .
- Baris 6 : Pengambilan sekaligus penambahan nilai warna merah yang disimpan kedalam variabel red .
- Baris 7 : Pengambilan sekaligus penambahan nilai warna hijau yang disimpan kedalam variabel $green$.
- Baris 8 : Pengambilan sekaligus penambahan nilai warna biru yang disimpan kedalam variabel $biru$.
- Baris 9 : Perhitungan nilai *grayscale* yang disimpan kedalam variabel $gray$.
- Baris 10-11: Pengecekan dimana jika nilai $gray$ kurang dari 190 maka variabel $biner$ bernilai 255 yaitu nilai warna putih, jika selain itu maka nilai variabel $biner$ bernilai 0 yaitu nilai warna hitam.
- Baris 13 : Menyimpan warna biner kedalam *pixel* citra dengan koordinat $f(x,y)$.

4.2.3 Ekstraksi Fitur Luas

Ekstraksi fitur luas digunakan untuk mendapatkan luas daun nilam pada citra. Implementasi fungsi ekstraksi fitur luas dapat dilihat pada *Source Code* 4.3.

```

1 public void setFiturLuas(){
2     for(int x=0; x<citra.getWidth(); x++){
3         for(int y=0; y<citra.getHeight(); y++){
4             if (citra.getRGB(x, y)==Color.WHITE.getRGB()){
5                 luas++;
6             }
7         }
8     }
9 }
```

Source Code 4.3 Fungsi Ekstraksi Fitur Luas

Keterangan *Source Code* 4.3:

- Baris 1 : Sebuah fungsi bernama “setFiturLuas” .

- Baris 2 : Perulangan dengan kondisi variabel x sama dengan 0 hingga kurang dari lebar citra.
- Baris 3 : Perulangan dengan kondisi variabel y sama dengan 0 hingga kurang dari tinggi citra.
- Baris 4 : Pengecekan dengan kondisi dimana nilai RGB *pixel* citra pada koordinat $f(x,y)$ tidak sama dengan nilai warna putih.
- Baris 5 : Penambahan nilai variabel *luas*.

4.2.4 Algoritma *Learning Vector Quantization*

Algoritma *Learning Vector Quantization* (LVQ) digunakan untuk proses klasifikasi. Terdapat dua buah proses dalam klasifikasi kualitas daun nilam menggunakan algoritma LVQ, yaitu proses pelatihan dan pengujian. Sub bab ini akan membahas implementasi kelas bernama “LVQ” yang menrepresentasikan beberapa fungsi yang nantinya dapat digunakan dalam proses pelatihan maupun pengujian dengan algoritma LVQ. Implementasi keelas LVQ dapat dilihat pada *Source Code 4.4*.

```

1 public class LVQ {
2     private double x[][];
3     private double t[];
4     private double d1[];
5     private double d2[];
6     private double C[];
7     private double dw1[][];
8     private double dw2[][];
9     private double w1_awal[];
10    private double w2_awal[];
11    private double w1[][];
12    private double w2[][];
13    private double learningRate, akurasi;
14    private int iterasi=0;
15    private String proses;
16    private LinkedHashMap<String, LinkedHashMap<String, Double>>
17    dataLVQ = new LinkedHashMap<String, LinkedHashMap<String,
18    Double>>();
19
20    public LVQ( String proses,int jumlahData, int jumlahFitur){
21        x= new double[jumlahData][jumlahFitur];
22        t= new double[jumlahData];
23        d1= new double[jumlahData]; d2= new double[jumlahData];
24        C= new double[jumlahData];
25        dw1= new double[jumlahData][jumlahFitur]; dw2= new
26        double[jumlahData][jumlahFitur];
27        w1_awal= new double[jumlahData]; w2_awal= new
28        double[jumlahData];
29        w1= new double[jumlahData][jumlahFitur]; w2= new
30        double[jumlahData][jumlahFitur];
31        this.proses=proses;
32    }

```

```
33
34     public void setX(int index, LinkedHashMap<String, Double>
35     data){
36         for(int j=0; j<x[index].length; j++)
37             x[index][j]=data.get("x"+(j+1)+"");
38             setTarget(index, data.get("target"));
39     }
40
41     public double getX(int i, int j){return x[i][j];}
42     public double[] getX(int i){ return x[i];}
43     public double[][] getX(){ return x;}
44
45     public void setTarget(int index, double t){ this.t[index]=t; }
46     public double getTarget(int index){return this.t[index];}
47
48     public void setLearningRate(double learningRate){
49     this.learningRate= learningRate;}
50     public double getLearningRate(){ return this.learningRate;}
51
52     public void setW_awal(double w1_awal[], double w2_awal[]){
53     this.w1_awal=w1_awal;
54     this.w2_awal=w2_awal;
55     }
56     public double getW1_awal(int i){return w1_awal[i];}
57     public double getW2_awal(int i){ return w2_awal[i];}
58     public double[] getW1_awal(){return w1_awal;}
59     public double[] getW2_awal(){ return w2_awal;}
60
61     public void setD1(int index){
62         if(this.proses.equalsIgnoreCase("training")){
63             for(int j=0; j<x[index].length; j++){
64                 if(index==0){
65                     d1[index]+=Math.pow(x[index][j]-w1_awal[j],2);
66                 }
67                 else{
68                     d1[index]+=Math.pow(x[index][j]-w1[index-
69                     1][j],2);
70                 }
71             }
72         }
73         if(this.proses.equalsIgnoreCase("testing")){
74             for(int j=0; j<x[index].length; j++){
75                 d1[index]+=Math.pow(x[index][j]-w1_awal[j],2);
76             }
77         }
78         d1[index]=Math.sqrt(d1[index]);
79     }
80     public double getD1(int index){return d1[index];}
81
82     public void setD2(int index){
83         if(this.proses.equalsIgnoreCase("training")){
84             for(int j=0; j<x[index].length; j++){
85                 if(index==0){
86                     d2[index]+=Math.pow(x[index][j]-w2_awal[j],2);
87                 }
88                 else{
89                     d2[index]+=Math.pow(x[index][j]-w2[index-
90                     1][j],2);
91                 }
92             }
93         }
94         else if(this.proses.equalsIgnoreCase("testing")){
95             for(int j=0; j<x[index].length; j++){
96                 d2[index]+=Math.pow(x[index][j]-w2_awal[j],2);
```

```
97     }
98     }
99     d2[index]=Math.sqrt(d2[index]);
100 }
101 public double getD2(int index){return d2[index];}
102
103 public void setC(int index){
104     if(d1[index]<d2[index]) C[index]=1;
105     else if (d2[index]<d1[index]) C[index]=2;
106 }
107 public double getC(int index){return C[index];}
108
109 public void setW(int index, double c, String s){
110     if(c==1){
111         setw1(index,s);
112         setw2(index, "tetap");
113     }
114     else{
115         setw2(index,s);
116         setw1(index, "tetap");
117     }
118 }
119
120 public void setw1(int index, String s){
121     for(int j=0; j<x[index].length; j++){
122         if(s.equalsIgnoreCase("didekatkan")){
123             if (index==0){
124                 w1[index][j]=w1_awal[j]+(learningRate*(x[index]
125                 [j]-w1_awal[j]));
126             }
127             else{
128                 w1[index][j]=w1[index-
129                 1][j]+(learningRate*(x[index][j]-w1[index-
130                 1][j]));
131             }
132         }
133         else if(s.equalsIgnoreCase("dijauhkan")){
134             if (index==0){
135                 w1[index][j]=w1_awal[j]-
136                 (learningRate*(x[index][j]-w1_awal[j]));
137             }
138             else{
139                 w1[index][j]=w1[index-1][j]-
140                 (learningRate*(x[index][j]-w1[index-1][j]));
141             }
142         }
143         else if(s.equalsIgnoreCase("tetap")){
144             if (index==0){
145                 w1[index][j]=w1_awal[j];
146             }
147             else{
148                 w1[index][j]=w1[index-1][j];
149             }
150         }
151     }
152 }
153
154 public void setw2(int index, String s){
155     for(int j=0; j<x[index].length; j++){
156         if(s.equalsIgnoreCase("didekatkan")){
157             if (index==0){
158                 w2[index][j]=w2_awal[j]+(learningRate*(x[index]
159                 [j]-w2_awal[j]));
160             }
```



```

161         else{
162             w2[index][j]=w2[index-
163                 1][j]+(learningRate*(x[index][j]-w2[index-
164                     1][j]));
165         }
166     }
167     else if(s.equalsIgnoreCase("dijauhkan")){
168         if (index==0){
169             w2[index][j]=w2_awal[j]-
170                 (learningRate*(x[index][j]-w2_awal[j]));
171         }
172         else{
173             w2[index][j]=w2[index-1][j]-
174                 (learningRate*(x[index][j]-w2[index-1][j]));
175         }
176     }
177     else if(s.equalsIgnoreCase("tetap")){
178         if (index==0){
179             w2[index][j]=w2_awal[j];
180         }
181         else{
182             w2[index][j]=w2[index-1][j];
183         }
184     }
185 }
186 }
187 public double getw1(int i, int j){return w1[i][j];}
188 public double[] getw1(int i){return w1[i];}
189 public double[][] getw1(){return w1;}
190 public double getw2(int i, int j){return w2[i][j];}
191 public double[] getw2(int i){return w2[i];}
192 public double[][] getw2(){return w2;}
193
194 public void setAkurasi(){
195     double benar=0;
196     for(int i=0; i<x.length; i++){
197         System.out.println(t[i]+"\\t"+C[i]);
198         if(t[i]==C[i]) benar+=1;
199     }
200     akurasi=(benar/x.length)*100
201 }
202 public double getAkurasi(){return akurasi;}
203
204 public void setIterasi(){ iterasi+=1;}
205 public int getIterasi(){return iterasi;}
206
207 public void setDataLVQ(String id, double target, double C){
208     LinkedHashMap<String, Double> data = new
209     LinkedHashMap<String, Double>();
210     data.put("target", target);
211     data.put("masuk kelas", C);
212     dataLVQ.put(id, data);
213 }
214 public LinkedHashMap<String, LinkedHashMap<String, Double>>
215 getDataLVQ(){
216     return dataLVQ;
217 }
218 }

```

Source Code 4.4 Kelas LVQ

Keterangan Source Code 4.4:

Baris 1 : Sebuah kelas bernama “LVQ”.

- Baris 2-18 : Deklarasi variabel global yang digunakan dalam kelas LVQ.
- Baris 20 : Sebuah *constructor* bernama “LVQ” dengan beberapa parameter meliputi *proses*, *jumlahData*, dan *jumlahFitur*.
- Baris 21-31 : Deklarasi variabel global yang digunakan dalam kelas LVQ.
- Baris 34 : Sebuah fnksi bernama “setX” yang digunakan untuk inisialisasi vektor pelatihan atau pengujian.
- Baris 36-38 : Inisialisasi vektor pelatihan yang disimpan kedalam variabel *x*.
- Baris 41-43 : 3 fungsi bernama “getX” yang digunakan untuk mengembalikan nilai variabel *x*.
- Baris 45 : Sebuah fungsi bernama “setTarget” yang digunakan untuk inisialisasi target.
- Baris 46 : sebuah fungsi bernama “getTarget” yang digunakan untuk mengembalikan nilai variabel *target*.
- Baris 48 : Sebuah fungsi bernama “setLearningrate” yang digunakan untuk inisialisasi laju pembelajaran.
- Baris 50 : Sebuah fungsi bernama “getLearningrate” yang digunakan untuk mengembalikan nilai variabel *learningRate*.
- Baris 52 : sebuah fungsi bernama “setW_awal” yang digunakan untuk inisialisasi bobot awal.
- Baris 53 : Inisialisasi bobot awal untuk kelas 1 (kualitas baik) yang disimpan kedalam variabel *w1_awal*.
- Baris 54 : Inisialisasi bobot awal untuk kelas 2 (kualitas kurang baik) yang disimpan kedalam variabel *w2_awal*.
- Baris 56-59 : 2 fungsi bernama “getW1_awal” yang digunakan untuk mengembalikan nilai variabel *w1_awal*, serta 2 fungsi bernama “getW2_awal” yang digunakan untuk mengembalikan nilai variabel *w2_awal*.
- Baris 61 : Sebuah fungsi bernama “setD1” yang digunakan untuk menghitung jarak vektor *input* (vektor pelatihan) ke kelas 1.
- Baris 62-78 : Menghitung jarak vektor *input* ke kelas 1 menggunakan jarak *Euclidean* yang hasilnya disimpan kedalam variabel *D1*.

- Baris 80 : Sebuah fungsi bernama “getD1” yang digunakan untuk mengembalikan nilai variabel $D1$.
- Baris 82 : Sebuah fungsi bernama “setD2” yang digunakan untuk menghitung jarak vektor *input* (vektor pelatihan) ke kelas 2
- Baris 83-99 : Menghitung jarak vektor *input* ke kelas 2 menggunakan jarak euclidean yang hasilnya disimpan kedalam variabel $D2$.
- Baris 101 : Sebuah fungsi bernama “getD2” yang digunakan untuk mengembalikan nilai variabel $D2$.
- Baris 103 : Sebuah fungsi bernama “setC” yang digunakan untuk menentukan kelas.
- Baris 104-105 : Penentuan kelas yang hasilnya disimpan kedalam variabel C .
- Baris 107 : Sebuah fungsi bernama “getC” yang digunakan untuk mengembalikan nilai variabel C .
- Baris 109 : Sebuah fungsi bernama “setW” yang digunakan untuk mengubah bobot.
- Baris 110-116 : Mengubah bobot sesuai kondisi yang ditentukan. Jika kelas sama dengan 1 maka ubah bobot kelas 1 dan bobot kelas 2 adalah tetap. Sebaliknya, jika kelas sama dengan 2 maka ubah bobot kelas 2 dan bobot kelas 1 adalah tetap.
- Baris 120 : Sebuah fungsi bernama “setW1” yang digunakan untuk mengubah bobot kelas 1 (kualitas baik).
- Baris 121-152 : Mengubah bobot kelas 1 sesuai kondisi yang ditentukan kemudian disimpan kedalam variabel $w1$.
- Baris 154 : Sebuah fungsi bernama “setW2” yang digunakan untuk mengubah bobot kelas 2 (kualitas kurang baik).
- Baris 155-186 : Mengubah bobot kelas 2 sesuai kondisi yang ditentukan kemudian disimpan kedalam variabel $w2$.
- Baris 187-192 : 3 fungsi bernama “getW1” yang digunakan untuk mengembalikan nilai variabel $w1$, serta 2 fungsi bernama “getW2” yang digunakan untuk mengembalikan nilai variabel $w2$.

- Baris 194 : Sebuah fungsi bernama “setAkurasi” yang digunakan untuk menghitung akurasi yang dihasilkan.
- Baris 195-200 : Menghitung akurasi yang hasilnya disimpan kedalam variabel *akurasi*.
- Baris 202 : Sebuah fungsi bernama “getAkurasi” yang digunakan untuk mengembalikan nilai variabel *akurasi*.
- Baris 204 : Sebuah fungsi bernama “setIterasi” yang digunakan untuk menambah iterasi.
- Baris 205 : Sebuah fungsi bernama “getIterasi” yang digunakan untuk mengembalikan nilai variabel *iterasi*.
- Baris 207 : Sebuah fungsi yang bernama “setDataLVQ” yang digunakan untuk menyimpan data hasil klasifikasi dari proses pengujian.
- Baris 208-212 : Menyimpan data hasil klasifikasi kedalam variabel *dataLVQ*.
- Baris 214 : Sebuah fungsi bernama “getDataLVQ” yang digunakan untuk mengembalikan nilai variabel *dataLVQ*.

4.2.4.1 Pelatihan dengan algoritma LVQ

Pelatihan dengan algoritma LVQ diimplementasikan untuk dapat menghasilkan bobot akhir yang akan digunakan sebagai bobot awal dalam proses pengujian. Implementasi fungsi pelatihan dengan algoritma LVQ ditunjukkan pada *Source Code 4.5*.

```

1 public void prosesTraining(String proses, LinkedHashMap<String,
2 LinkedHashMap<String, Double>> dataTraining, int jumlahData, int
3 jumlahFitur, double lajuPembelajaran, double
4 pengurangLajuPembelajaran, double lajuPembelajaranMinimum, int
5 iterasiMaksimum) {
6
7     training= new LVQ("training", jumlahData, jumlahFitur);
8
9     training.setLearningRate(lajuPembelajaran);
10
11     double w1_awal[] = new double[jumlahFitur], w2_awal[] =
12     new double[jumlahFitur];
13
14     for(int i=0; i<jumlahFitur; i++){
15         w1_awal[i]=Math.random()*1;
16         w2_awal[i]=Math.random()*1;
17     }
18
19     training.setW_awal(w1_awal, w2_awal);
20
21     if(proses.equalsIgnoreCase("Pelatihan
22     LVQ"))tampilTabelBobot("awal");

```

```

23
24     double pengurang=pengurangLajuPembelajaran;
25
26     double learning_rate_minimum= lajuPembelajaranMinimum;
27
28     int iterasi_maksimum=iterasiMaksimum;
29
30     boolean stop=false;
31     while(stop!=true){
32         int index=0;
33         for(String id:dataTraining.keySet()){
34             LinkedHashMap<String, Double> data =
35                 dataTraining.get(id);
36             training.setX(index, data);
37             training.setD1(index);
38             training.setD2(index);
39             training.setC(index);
40             if(training.getTarget(index)==training.getC(index)
41                 ){
42                 training.setW(index,
43                     training.getC(index),"didekatkan");
44             }else{
45                 training.setW(index,
46                     training.getC(index),"dijauhkan");
47             }
48             index++;
49         }
50
51         training.setLearningRate(training.getLearningRate()*pen
52             gurang);
53
54         training.setIterasi();
55
56         training.setW_awal(training.getw1(training.getw1().leng
57             th-1), training.getw2(training.getw2().length-1));
58
59         if(training.getLearningRate()<learning_rate_minimum ||
60             training.getIterasi()>=iterasi_maksimum){
61             stop=true;
62             training.setW_awal(training.getW1_awal(),
63                 training.getW2_awal());
64         }
65     }
66
67     if(proses.equalsIgnoreCase("Pelatihan LVQ")){
68         jLabel_berhentiDiIterasi.setText(training.getIterasi()
69             + "");
70         tampilTabelBobot("akhir");
71     }
72 }
73
74
75
76

```

Source Code 4.5 Fungsi Pelatihan dengan Algoritma LVQ

Keterangan *Source Code 4.5*:

Baris 1 : Sebuah metode bernama “prosesTraining” dengan beberapa parameter meliputi variabel *proses*, *dataTraining*, *jumlahData*,

jumlahFitur, *lajuPembelajaran*, *pengurangLajuPembelajaran*,
lajuPembelajaranMinimum, dan *iterasiMaksimum*.

- Baris 7 : Deklarasi objek training dari kelas LVQ.
- Baris 9 : Inisialisasi laju pembelajaran.
- Baris 11-19: Inisialisasi bobot awal.
- Baris 24 : Inisialisasi pengurang laju pembelajaran.
- Baris 26 : Inisialisasi laju pembelajaran minimum.
- Baris 28 : Inisialisasi iterasi maksimum
- Baris 31 : Perulangan dengan kondisi berhenti adalah salah.
- Baris 32 : Deklarasi variabel *index*.
- Baris 33 : Perulangan terhadap data pelatihan untuk setiap id data latih.
- Baris 34 : Pengambilan data latih.
- Baris 36 : Inisialisasi vektor pelatihan
- Baris 37 : Perhitungan jarak ke kelas 1 (kelas kualitas baik).
- Baris 38 : Perhitungan jarak ke kelas 2 (kelas kualitas kurang baik).
- Baris 39 : Penentuan kelas atau kategori.
- Baris 40-47: Perubahan bobot sesuai dengan kondisi yang ditentukan. Jika target sama dengan kelas maka bobot akan didekatkan dengan cara dijumlahkan. Sebaliknya jika target tidak sama dengan kelas maka bobot akan dijauhkan dengan cara dikurangkan.
- Baris 48 : Penambahan nilai variabel *index*.
- Baris 51 : Reduksi atau pengurangan laju pembelajaran.
- Baris 54 : Penambahan iterasi.
- Baris 56 : Inisialisasi bobot awal untuk iterasi berikutnya.
- Baris 59 : Pengecekan kondisi berhenti dengan syarat nilai laju pembelajaran mencapai nilai laju pembelajaran minimum atau iterasi mencapai iterasi maksimum.
- Baris 61 : Inisialisasi kondisi berhenti adalah benar.
- Baris 62 : Inisialisasi bobot awal yang akan digunakan pada proses pengujian.

4.2.4.2 Pengujian dengan algoritma LVQ

Pengujian dengan algoritma LVQ diimplementasikan untuk mendapatkan hasil klasifikasi kualitas daun nilam yang diuji. Implementasi fungsi pelatihan dengan algoritma LVQ ditunjukkan pada *Source Code 4.6*.

```

1 public void prosesTesting(LinkedHashMap<String,
2 LinkedHashMap<String, Double>> dataTesting, int jumlahData, int
3 jumlahFitur){
4
5     testing= new LVQ("testing",jumlahData, jumlahFitur);
6
7     testing.setW_awal(training.getW1_awal(),training.getW2_awal
8     ());
9
10    int index=0;
11    for(String id:dataTesting.keySet()){
12        LinkedHashMap<String, Double> data =
13        dataTesting.get(id);
14        testing.setX(index, data);
15        testing.setD1(index);
16        testing.setD2(index);
17        testing.setC(index);
18        testing.setDataLVQ(id, testing.getTarget(index),
19        testing.getC(index));
20        index++;
21    }
22
23    testing.setAkurasi();
24 }

```

Source Code 4.6 Fungsi Pengujian dengan Algoritma LVQ

Keterangan *Source Code 4.6*:

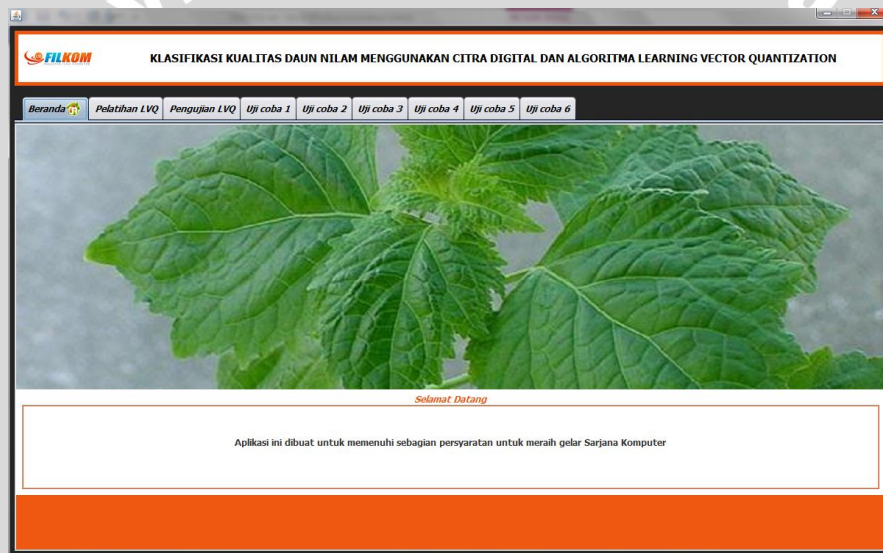
- Baris 1 : Sebuah metode bernama “prosesTesting” dengan beberapa parameter meliputi dataTesting, jumlahData, dan jumlahFitur.
- Baris 5 : Deklarasi objek testing.
- Baris 7 : Inisialisasi bobot awal.
- Baris 10 : Deklarasi variabel *index*.
- Baris 11 : Perulangan terhadap data pengujian untuk setiap id data uji.
- Baris 12 : Pengambilan data uji.
- Baris 14 : Inisialisasi vektor pengujian
- Baris 15 : Perhitungan jarak ke kelas 1 (kelas kualitas baik).
- Baris 16 : Perhitungan jarak ke kelas 2 (kelas kualitas kurang baik).
- Baris 17 : Penentuan kelas atau kategori.
- Baris 18 : Menyimpan data hasil klasifikasi.
- Baris 20 : Penambahan nilai variabel *index*.
- Baris 23 : Menghitung akurasi.

4.3 Implementasi Antarmuka

Implementasi antarmuka bertujuan untuk mengimplementasikan rancangan antarmuka pada sistem yang akan dibangun pada bab sebelumnya. Implementasi antarmuka akan menampilkan halaman beranda, pelatihan LVQ, pengujian LVQ, uji coba 1, uji coba 2, uji coba 3, uji coba 4, uji coba 5, dan uji coba 6.

4.3.1 Halaman Beranda

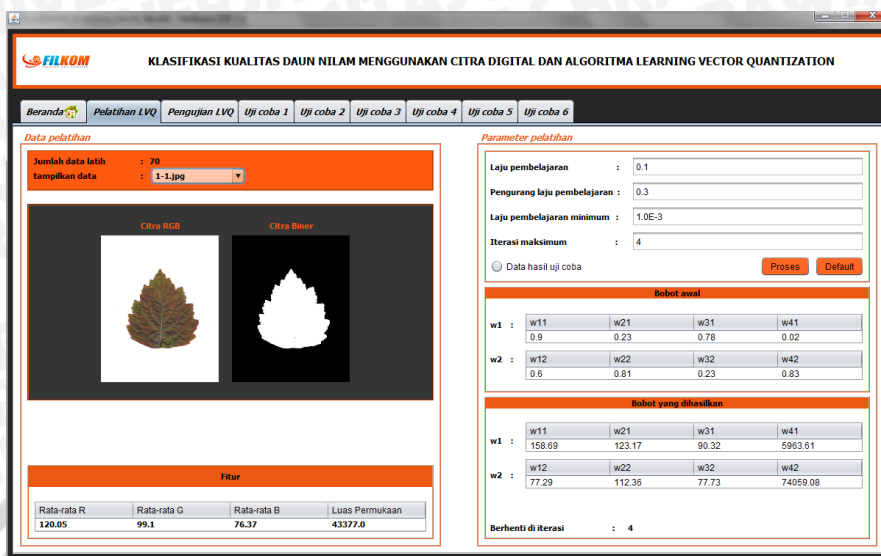
Halaman beranda adalah halaman yang pertamakali muncul saat aplikasi dijalankan. Halaman beranda berisi informasi mengenai beberapa halaman yang termuat dalam sistem serta informasi lainnya. Implementasi antarmuka halaman beranda ditunjukkan pada Gambar 4.1.



Gambar 4.1 Implementasi Antarmuka Halaman Beranda

4.3.2 Halaman Pelatihan LVQ

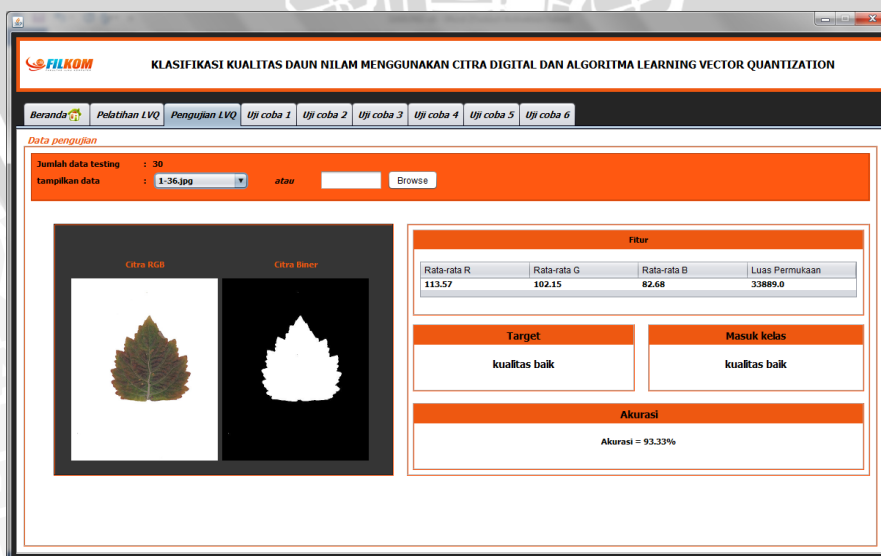
Halaman pelatihan LVQ merupakan halaman yang akan menampilkan proses pelatihan menggunakan algoritma LVQ. Halaman ini menampilkan data latih berupa citra sekaligus fitur yang sudah diekstraksi, serta menampilkan beberapa parameter yang digunakan dalam proses pelatihan dengan algoritma LVQ. Halaman ini juga memungkinkan pengguna untuk dapat memasukkan nilai parameter tersebut. Implementasi antarmuka halaman pelatihan LVQ ditunjukkan pada Gambar 4.2.



Gambar 4.2 Implementasi Antarmuka Halaman Pelatihan LVQ

4.3.3 Halaman Pengujian LVQ

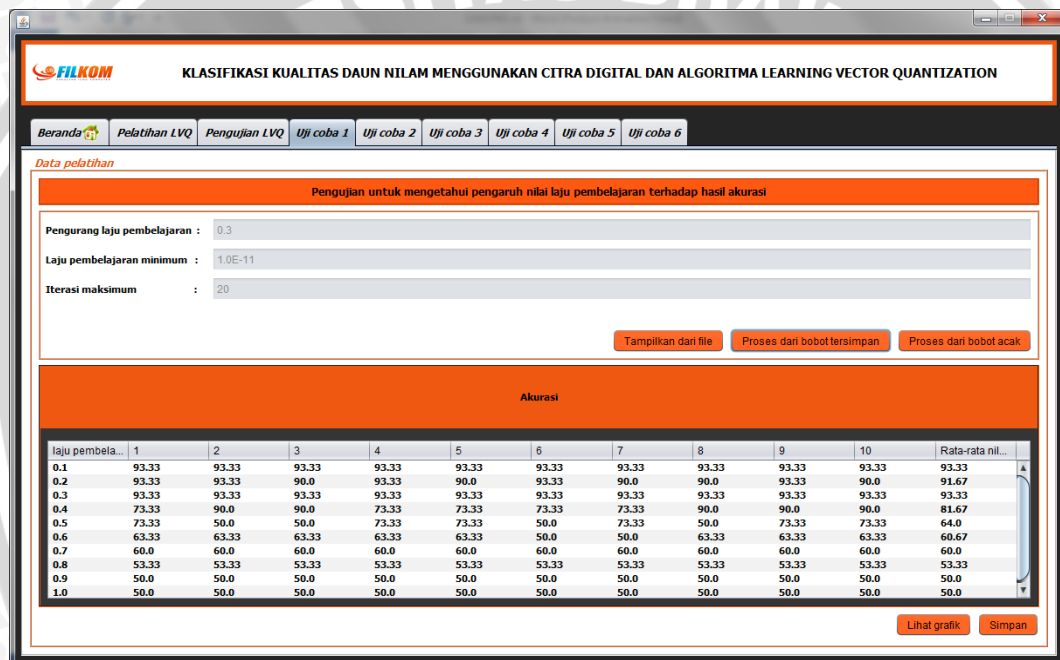
Halaman pengujian LVQ merupakan halaman yang akan menampilkan proses pengujian menggunakan algoritma LVQ. Halaman ini menampilkan data uji berupa citra sekaligus fitur yang sudah diekstraksi, serta menampilkan keluaran berupa hasil klasifikasi dan akurasi yang didapatkan. Halaman ini juga memungkinkan pengguna untuk dapat melakukan pencarian citra daun nilam yang akan diidentifikasi. Implementasi antarmuka halaman pengujian LVQ ditunjukkan pada Gambar 4.3.



Gambar 4.3 Implementasi Antarmuka Halaman Pengujian LVQ

4.3.4 Halaman Uji Coba 1

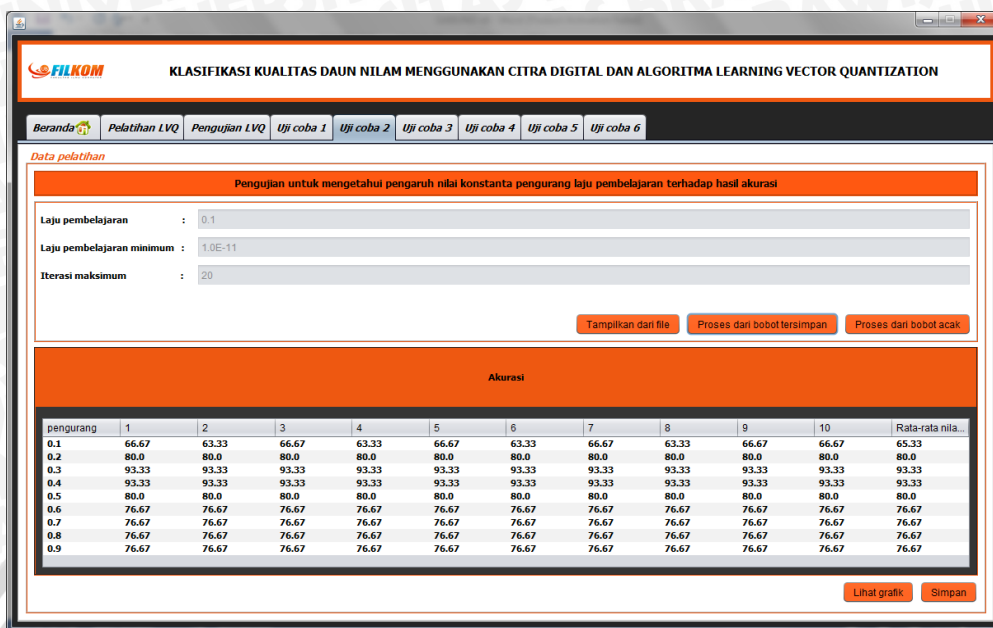
Halaman uji coba 1 merupakan halaman yang menampilkan hasil pengujian untuk mengetahui pengaruh nilai laju pembelajaran terhadap hasil akurasi. Halaman ini menampilkan beberapa parameter yang telah ditentukan serta menampilkan tabel hasil uji coba yang dilakukan. Dalam halaman ini, pengguna dapat memulai proses uji coba baru, menyimpan data hasil uji coba yang ditampilkan pada tabel, melihat grafik, maupun menampilkan kembali data yang telah disimpan ke dalam file. Implementasi antarmuka halaman uji coba 1 ditunjukkan pada Gambar 4.4.



Gambar 4.4 Implementasi Antarmuka Halaman Uji Coba 1

4.3.5 Halaman Uji Coba 2

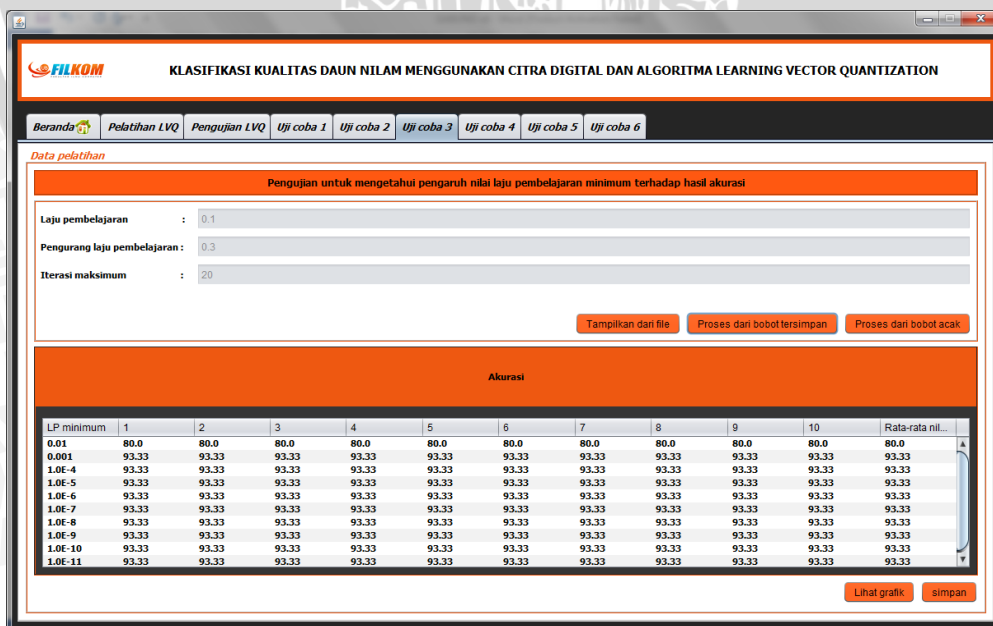
Halaman uji coba 2 merupakan halaman yang menampilkan hasil pengujian untuk mengetahui pengaruh nilai konstanta pengurang laju pembelajaran terhadap hasil akurasi. Halaman ini memiliki fitur yang sama seperti halaman uji coba sebelumnya. Implementasi antarmuka halaman uji coba 2 ditunjukkan pada Gambar 4.5.



Gambar 4.5 Implementasi Antarmuka Halaman Uji Coba 2

4.3.6 Halaman Uji Coba 3

Halaman uji coba 3 merupakan halaman yang menampilkan hasil pengujian untuk mengetahui pengaruh nilai laju pembelajaran minimum terhadap hasil akurasi. Halaman ini memiliki fitur yang sama seperti halaman uji coba sebelumnya. Implementasi antarmuka halaman uji coba 3 ditunjukkan pada Gambar 4.6.



Gambar 4.6 Implementasi Antarmuka Halaman Uji Coba 3



4.3.7 Halaman Uji Coba 4

Halaman uji coba 4 merupakan halaman yang menampilkan hasil pengujian untuk mengetahui pengaruh jumlah iterasi pembelajaran terhadap hasil akurasi. Halaman ini memiliki fitur yang sama seperti halaman uji coba sebelumnya. Implementasi antarmuka halaman uji coba 4 ditunjukkan pada Gambar 4.7.

Iterasi maks	1	2	3	4	5	6	7	8	9	10	Rata-rata nil...
2	80.0	80.0	80.0	80.0	80.0	80.0	80.0	80.0	80.0	80.0	80.0
4	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33
6	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33
8	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33
10	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33
12	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33
14	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33
16	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33
18	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33
20	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33	93.33

Gambar 4.7 Implementasi Antarmuka Halaman Uji Coba 4

4.3.8 Halaman Uji Coba 5

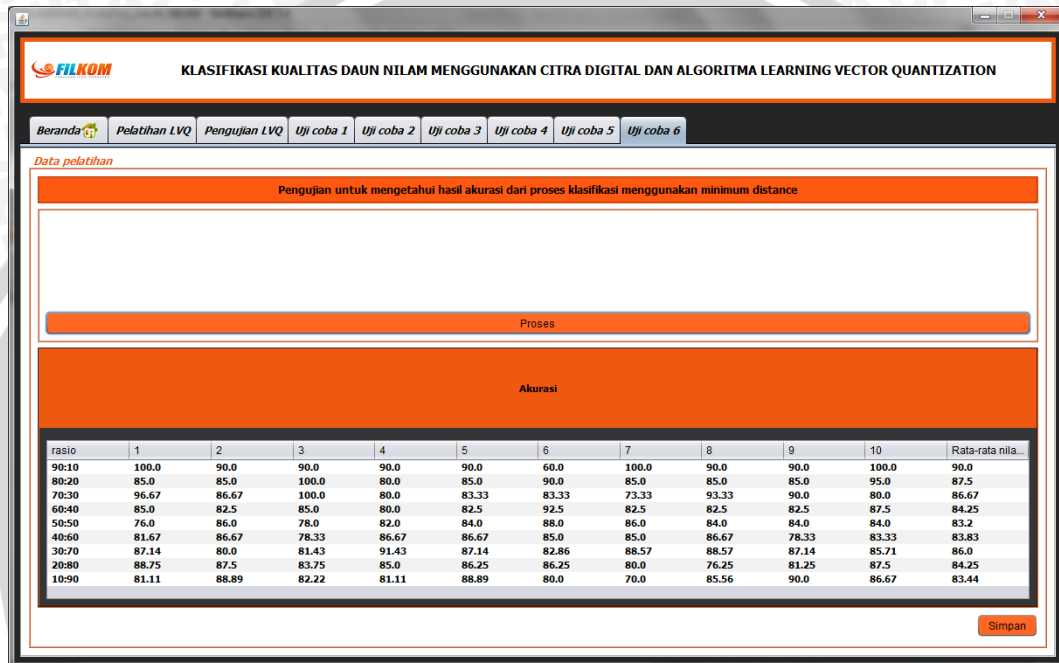
Halaman uji coba 5 merupakan halaman yang menampilkan hasil pengujian untuk mengetahui pengaruh perbandingan jumlah data latih dengan data uji terhadap hasil akurasi. Halaman ini memiliki fitur yang sama seperti halaman uji coba sebelumnya. Implementasi antarmuka halaman uji coba 5 ditunjukkan pada Gambar 4.8.

Ratio	1	2	3	4	5	6	7	8	9	10	Rata-rata nil...
90:10	100.0	90.0	100.0	100.0	100.0	90.0	100.0	100.0	100.0	100.0	98.0
80:20	85.0	95.0	100.0	95.0	90.0	85.0	95.0	100.0	85.0	100.0	93.0
70:30	90.0	86.67	80.0	93.33	90.0	93.33	83.33	86.67	83.33	93.33	88.0
60:40	85.0	87.5	87.5	82.5	92.5	82.5	80.0	80.0	92.5	72.5	82.25
50:50	82.0	78.0	84.0	78.0	82.0	84.0	80.0	76.0	78.0	66.0	78.0
40:60	83.33	60.0	80.0	51.67	58.33	58.33	60.0	65.0	78.33	61.67	63.67
30:70	51.43	57.14	57.14	51.43	52.86	57.14	51.43	51.43	50.0	57.14	53.71
20:80	50.0	50.0	51.25	50.0	50.0	50.0	51.25	52.5	50.0	50.0	50.5
10:90	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0

Gambar 4.8 Implementasi Antarmuka Halaman Uji Coba 5

4.3.9 Halaman Uji Coba 6

Halaman uji coba 6 merupakan halaman yang menampilkan hasil pengujian untuk mengetahui hasil akurasi dari proses klasifikasi menggunakan *minimum distance*. Halaman ini memiliki fitur yang sama seperti halaman uji coba sebelumnya. Implementasi antarmuka halaman uji coba 6 ditunjukkan pada Gambar 4.9.



The screenshot shows a web application window titled 'KLASIFIKASI KUALITAS DAUN NILAM MENGGUNAKAN CITRA DIGITAL DAN ALGORITMA LEARNING VECTOR QUANTIZATION'. The interface includes a navigation menu with options like 'Beranda', 'Pelatihan LVQ', 'Pengujian LVQ', and 'Uji coba 1' through 'Uji coba 6'. The main content area is titled 'Data pelatihan' and contains a sub-header 'Pengujian untuk mengetahui hasil akurasi dari proses klasifikasi menggunakan minimum distance'. Below this, there are sections for 'Proses' and 'Akurasi'. The 'Akurasi' section displays a table with 11 columns: 'Rasio', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', and 'Rata-rata nilai...'. The table contains numerical accuracy values for various ratios.

Rasio	1	2	3	4	5	6	7	8	9	10	Rata-rata nilai...
90:10	100.0	90.0	90.0	90.0	90.0	60.0	100.0	90.0	90.0	100.0	90.0
80:20	85.0	85.0	100.0	80.0	85.0	90.0	85.0	85.0	85.0	95.0	87.5
70:30	96.67	86.67	100.0	80.0	83.33	83.33	73.33	93.33	90.0	80.0	86.67
60:40	85.0	82.5	85.0	80.0	82.5	92.5	82.5	82.5	82.5	87.5	84.25
50:50	76.0	86.0	78.0	82.0	84.0	88.0	86.0	84.0	84.0	84.0	83.2
40:60	81.67	86.67	78.33	86.67	86.67	85.0	85.0	86.67	78.33	83.33	83.83
30:70	87.14	80.0	81.43	91.43	87.14	82.86	88.57	88.57	87.14	85.71	86.0
20:80	88.75	87.5	83.75	85.0	86.25	86.25	80.0	76.25	81.25	87.5	84.25
10:90	81.11	88.89	82.22	81.11	88.89	80.0	70.0	85.56	90.0	86.67	83.44

Gambar 4.9 Implementasi Antarmuka Halaman Uji Coba 6

BAB V PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai pengujian dan analisis sistem klasifikasi kualitas daun nilam menggunakan citra digital dan *algoritma Learning Vector Quantization* (LVQ). Proses pengujian pada penelitian ini dilakukan dengan melakukan skenario yang telah ditentukan sebelumnya terhadap akurasi yang dihasilkan. Analisis hasil pengujian dilakukan untuk menganalisa hasil pengujian yang telah dilakukan.

5.1 Hasil dan Analisis Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran terhadap Hasil Akurasi

Pengujian ini bertujuan untuk mendapatkan rata-rata tingkat akurasi terbaik dari sepuluh kali percobaan pada masing-masing nilai laju pembelajaran (α). Nilai laju pembelajaran yang digunakan dalam pengujian ini meliputi 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, dan 1.

Pengujian ini menggunakan data latih dan data uji yang telah ditentukan. Data latih yang digunakan berjumlah 70 data, yaitu 35 data dari masing-masing kelas. Sedangkan data uji yang digunakan berjumlah 30 data, yaitu 15 data dari masing-masing kelas. Pengujian ini dilakukan dengan menggunakan nilai konstanta pengurang laju pembelajaran = 0.3, laju pembelajaran minimum = 10^{-11} , dan iterasi maksimum = 20, serta bobot awal yang diinisialisasi secara acak dengan rentang nilai 0-1 pada setiap percobaan. Hasil pengujian untuk mengetahui pengaruh nilai laju pembelajaran terhadap hasil akurasi ditunjukkan pada Tabel 5.1.

Tabel 5.1 Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran terhadap Hasil Akurasi

Laju pembelajaran	Percobaan ke-										Rata-rata akurasi (%)	
	1	2	3	4	5	6	7	8	9	10		
0,1	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
0,2	93,33	93,33	90,00	93,33	90,00	93,33	90,00	90,00	93,33	90,00	90,00	91,67
0,3	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
0,4	73,33	90,00	90,00	73,33	73,33	73,33	73,33	90,00	90,00	90,00	90,00	81,67
0,5	73,33	50,00	50,00	73,33	73,33	50,00	73,33	50,00	73,33	73,33	73,33	64,00

0,6	63,33	63,33	63,33	63,33	63,33	50,00	50,00	63,33	63,33	63,33	60,67
0,7	60,00	60,00	60,00	60,00	60,00	60,00	60,00	60,00	60,00	60,00	60,00
0,8	53,33	53,33	53,33	53,33	53,33	53,33	53,33	53,33	53,33	53,33	53,33
0,9	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00
1	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00

Berdasarkan tabel hasil pengujian diatas, grafik pengujian untuk mengetahui pengaruh nilai laju pembelajaran terhadap hasil akurasi dipresentasikan pada Gambar 5.1.

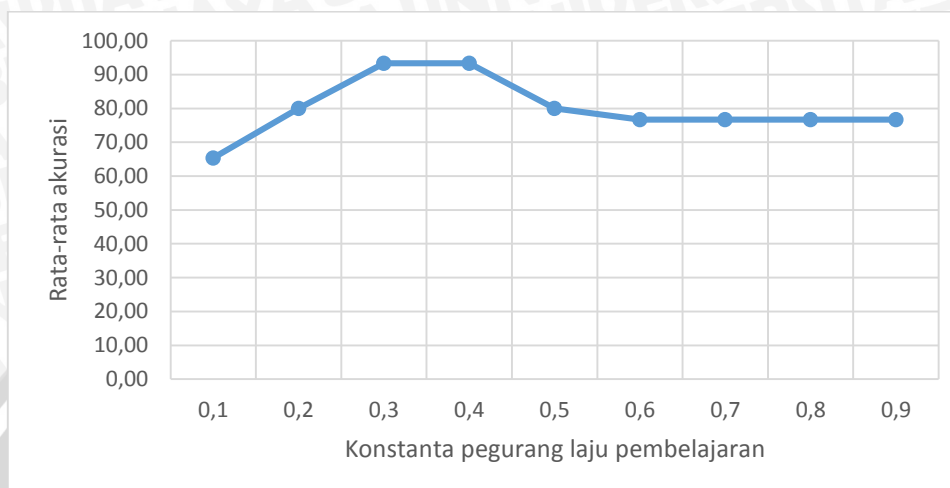


Gambar 5.1 Grafik Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran terhadap Hasil Akurasi

Berdasarkan hasil pengujian terhadap nilai laju pembelajaran didapatkan rata-rata tingkat akurasi tertinggi sebesar 93.33 % dengan nilai laju pembelajaran sama dengan 0.1 dan 0.3. Sedangkan rata-rata tingkat akurasi terendah yang didapatkan adalah sebesar 50% dengan nilai laju pembelajaran sama dengan 0.9 dan 1.

Secara keseluruhan, grafik pada Gambar 5.1 menunjukkan nilai akurasi yang menurun. Dari hasil tersebut diketahui bahwa semakin besar nilai laju pembelajaran mengakibatkan semakin rendah tingkat akurasinya. Pada LVQ, laju pembelajaran akan berpengaruh pada saat pergeseran nilai bobot sesuai dengan

Berdasarkan tabel hasil pengujian diatas, grafik pengujian untuk mengetahui pengaruh nilai konstanta pengurang laju pembelajaran terhadap hasil akurasi dipresentasikan pada Gambar 5.2.



Gambar 5.2 Grafik Pengujian untuk Mengetahui Pengaruh Nilai Konstanta Pengurang Laju Pembelajaran terhadap Hasil Akurasi

Berdasarkan hasil pengujian terhadap nilai konstanta pengurang laju pembelajaran didapatkan rata-rata tingkat akurasi tertinggi sebesar 93.33 % dengan nilai konstanta pengurang laju pembelajaran sama dengan 0.3 dan 0.4. Sedangkan rata-rata tingkat akurasi terendah yang didapatkan adalah sebesar 65.33% dengan nilai konstanta pengurang laju pembelajaran sama dengan 0.1.

Secara keseluruhan, grafik pada Gambar 5.2 menunjukkan nilai akurasi yang mengalami kenaikan dan penurunan. Setelah mengalami kenaikan, grafik menunjukkan nilai akurasi yang konvergen pada nilai konstanta pengurang laju pembelajaran sama dengan 0.3 dan 0.4. begitu pula setelah mengalami penurunan, grafik menunjukkan nilai akurasi yang konvergen pada nilai konstanta pengurang laju pembelajaran sama dengan 0.6 sampai dengan 0.9. Hal tersebut membuat grafik diatas tidak menunjukkan suatu pola untuk diambil suatu kesimpulan. Bila ditinjau dari perannya, konstanta pengurang laju pembelajaran berperan dalam memperkecil laju pembelajaran sehingga dapat mencapai bobot yang ideal. Dari pengujian ini, konstanta pengurang laju pembelajaran yang menghasilkan bobot yang paling ideal adalah 0.3 dan 0.4 karena konstanta tersebut menunjukkan nilai akurasi yang paling besar.

5.3 Hasil dan Analisis Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran Minimum terhadap Hasil Akurasi

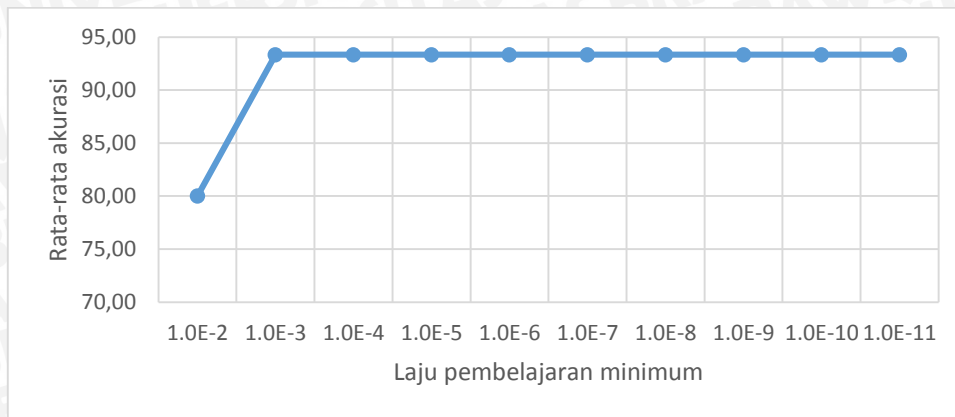
Pengujian ini bertujuan untuk mendapatkan rata-rata tingkat akurasi terbaik dari sepuluh kali percobaan pada masing-masing nilai laju pembelajaran minimum. Nilai laju pembelajaran minimum yang digunakan dalam pengujian ini meliputi 10^{-2} , 10^{-3} , 10^{-4} , 10^{-5} , 10^{-6} , 10^{-7} , 10^{-8} , 10^{-9} , 10^{-10} , dan 10^{-11} .

Pengujian ini menggunakan data latih dan data uji yang telah ditentukan. Data latih yang digunakan berjumlah 70 data, yaitu 35 data dari masing-masing kelas. Sedangkan data uji yang digunakan berjumlah 30 data, yaitu 15 data dari masing-masing kelas. Pengujian ini dilakukan dengan menggunakan nilai laju pembelajaran = 0.1, konstanta pengurang laju pembelajaran = 0.3, dan iterasi maksimum = 20, serta bobot awal yang diinisialisasi secara acak dengan rentang nilai 0-1 pada setiap percobaan. Hasil pengujian untuk mengetahui pengaruh nilai laju pembelajaran minimum terhadap hasil akurasi ditunjukkan pada Tabel 5.3.

Tabel 5.3 Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran Minimum terhadap Hasil Akurasi

Laju pembelajaran minimum	Percobaan ke-										Rata-rata akurasi (%)	
	1	2	3	4	5	6	7	8	9	10		
10^{-2}	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00
10^{-3}	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
10^{-4}	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
10^{-5}	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
10^{-6}	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
10^{-7}	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
10^{-8}	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
10^{-9}	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
10^{-10}	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
10^{-11}	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33

Berdasarkan tabel hasil pengujian diatas, grafik pengujian untuk mengetahui pengaruh nilai laju pembelajaran minimum terhadap hasil akurasi dipresentasikan pada Gambar 5.3.



Gambar 5.3 Grafik Pengujian untuk Mengetahui Pengaruh Nilai Laju Pembelajaran Minimum terhadap Hasil Akurasi

Berdasarkan hasil pengujian terhadap nilai laju pembelajaran minimum didapatkan rata-rata tingkat akurasi tertinggi sebesar 93,33 % dengan nilai laju pembelajaran minimum sama dengan 10^{-3} sampai dengan 10^{-11} . Sedangkan rata-rata tingkat akurasi terendah yang didapatkan adalah sebesar 80% dengan nilai laju pembelajaran minimum sama dengan 10^{-2} .

Grafik pada Gambar 5.3 menunjukkan nilai akurasi mengalami peningkatan mulai dari nilai laju pembelajaran minimum sama dengan 10^{-3} dan tidak mengalami perubahan hingga nilainya sama dengan 10^{-11} . Laju pembelajaran minimum merupakan salah satu kondisi berhenti dalam algoritma pelatihan LVQ. Dari hasil tersebut diketahui bahwa sejak iterasi mencapai kondisi berhenti dengan nilai laju pembelajaran minimum = 10^{-3} , pemberhentian tersebut telah mencapai konvergensi pada perubahan bobot sehingga akurasi yang didapatkan pun lebih baik.

5.4 Hasil dan Analisis Pengujian untuk Mengetahui Pengaruh Jumlah Iterasi Pembelajaran terhadap Hasil Akurasi

Pengujian ini bertujuan untuk mendapatkan rata-rata tingkat akurasi terbaik dari sepuluh kali percobaan pada masing-masing jumlah iterasi. Jumlah iterasi yang digunakan dalam pengujian ini meliputi 2, 4, 6, 8, 10, 12, 14, 16, 18, dan 20.

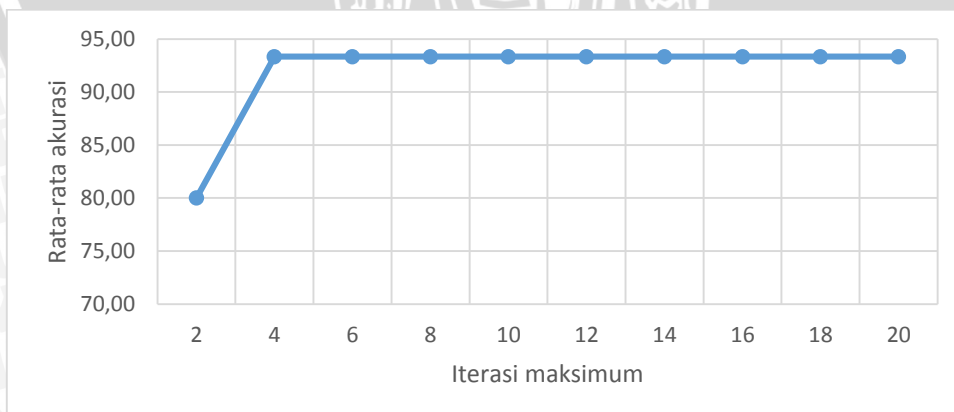
Pengujian ini menggunakan data latih dan data uji yang telah ditentukan. Data latih yang digunakan berjumlah 70 data, yaitu 35 data dari masing-masing kelas. Sedangkan data uji yang digunakan berjumlah 30 data, yaitu 15 data dari masing-masing kelas. Pengujian ini dilakukan dengan menggunakan nilai laju

pembelajaran = 0.1, konstanta pengurang laju pembelajaran= 0.3, dan laju pembelajaran minimum = 10^{-11} , serta bobot awal yang diinisialisasi secara acak dengan rentang nilai 0-1 pada setiap percobaan. Hasil pengujian untuk mengetahui pengaruh jumlah iterasi terhadap hasil akurasi ditunjukkan pada Tabel 5.4.

Tabel 5.4 Pengujian untuk Mengetahui Pengaruh Jumlah Iterasi Pembelajaran terhadap Hasil Akurasi

Iterasi maksimum	Percobaan ke-										Rata-rata akurasi (%)	
	1	2	3	4	5	6	7	8	9	10		
2	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00	80,00
4	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
6	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
8	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
10	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
12	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
14	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
16	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
18	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33
20	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33	93,33

Berdasarkan tabel hasil pengujian diatas, grafik pengujian untuk mengetahui pengaruh jumlah iterasi maksimum terhadap hasil akurasi dipresentasikan pada Gambar 5.4.



Gambar 5.4 Grafik Pengujian untuk Mengetahui Pengaruh Jumlah Iterasi Maksimum terhadap Hasil Akurasi

Berdasarkan hasil pengujian terhadap jumlah iterasi maksimum didapatkan rata-rata tingkat akurasi tertinggi sebesar 93.33 % dengan iterasi maksimum sama dengan 4 sampai dengan 20. Sedangkan rata-rata tingkat akurasi terendah yang didapatkan adalah sebesar 80% dengan iterasi maksimum sama dengan 2.

Grafik pada Gambar 5.4 menunjukkan nilai akurasi mengalami peningkatan mulai dari jumlah iterasi maksimum sama dengan 4 dan tidak mengalami perubahan hingga jumlahnya sama dengan 20. Sama seperti laju pembelajaran minimum, jumlah iterasi maksimum juga merupakan salah satu kondisi berhenti dalam algoritma pelatihan LVQ. Dari hasil tersebut diketahui bahwa sejak iterasi mencapai kondisi berhenti dengan jumlah iterasi maksimum = 4, pemberhentian tersebut telah mencapai konvergensi pada perubahan bobot sehingga akurasi yang didapatkan pun lebih baik. Hal ini pun telah menunjukkan bahwa algoritma LVQ mampu melakukan proses pembelajaran dengan cepat.

5.5 Hasil dan Analisis Pengujian Untuk Mengetahui Pengaruh Perbandingan Jumlah Data Latih dengan Data Uji terhadap Hasil Akurasi

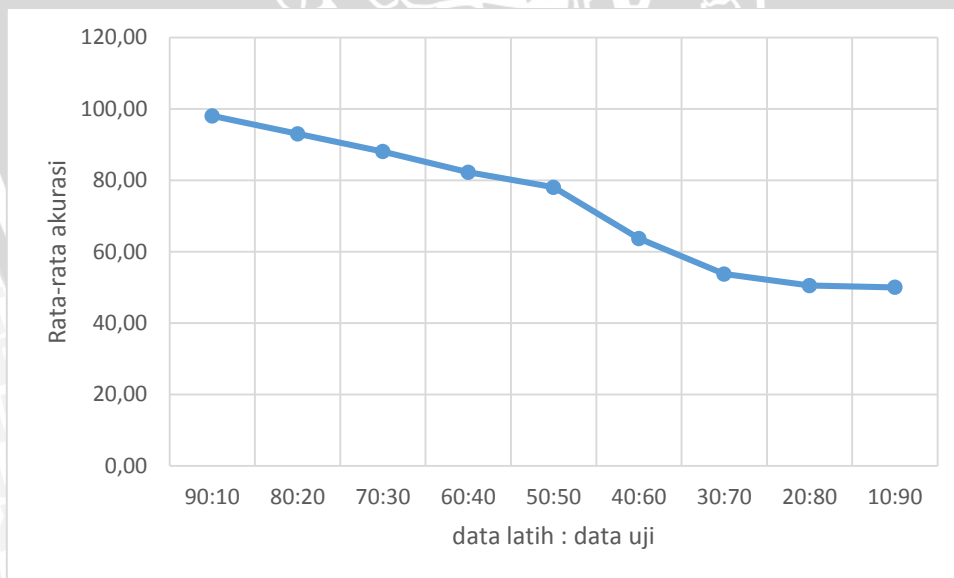
Pengujian ini bertujuan untuk mendapatkan rata-rata tingkat akurasi terbaik dari sepuluh kali percobaan pada masing-masing perbandingan jumlah data latih dengan data uji. Perbandingan jumlah data latih dengan data uji yang digunakan dalam pengujian ini meliputi 90:10, 80:20, 70:30, 60:40, 50:50, 40:60, 30:70, 20:80, dan 10:90. Data latih dan data uji yang digunakan diambil secara acak dari keseluruhan dataset, serta jumlah masing-masing kelas adalah seimbang.

Pengujian ini dilakukan dengan menggunakan parameter dengan akurasi terbaik yang telah didapatkan dari empat skenario pengujian yang telah dilakukan sebelumnya. Parameter yang digunakan meliputi nilai laju pembelajaran = 0.1, konstanta pengurang laju pembelajaran = 0.3, laju pembelajaran minimum = 10^{-3} , dan jumlah iterasi = 4, serta bobot awal yang diinisialisasi secara acak dengan rentang nilai 0-1 pada setiap percobaan. Hasil pengujian untuk mengetahui pengaruh perbandingan jumlah data latih dengan data uji terhadap hasil akurasi ditunjukkan pada Tabel 5.5.

Tabel 5.5 Pengujian untuk Mengetahui Pengaruh Perbandingan Jumlah Data Latih dengan Data Uji terhadap Hasil Akurasi

Data latih : data uji	Percobaan ke-										Rata-rata akurasi (%)
	1	2	3	4	5	6	7	8	9	10	
90:10	100	90,00	100	100	100	90,00	100	100	100	100	98,00
80:20	85,00	95,00	100	95,00	90,00	85,00	95,00	100	85,00	100	93,00
70:30	90,00	86,67	80,00	93,33	90,00	93,33	83,33	86,67	83,33	93,33	88,00
60:40	65,00	87,50	87,50	82,50	92,50	82,50	80,00	80,00	92,50	72,50	82,25
50:50	82,00	70,00	84,00	78,00	82,00	84,00	80,00	76,00	78,00	66,00	78,00
40:60	63,33	60,00	80,00	51,67	58,33	58,33	60,00	65,00	78,33	61,67	63,67
30:70	51,43	57,14	57,14	51,43	52,86	57,14	51,43	51,43	50,00	57,14	53,71
20:80	50,00	50,00	51,25	50,00	50,00	50,00	51,25	52,50	50,00	50,00	50,50
10:90	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00	50,00

Berdasarkan tabel hasil pengujian diatas, grafik pengujian untuk mengetahui pengaruh perbandingan jumlah data latih dengan data uji terhadap hasil akurasi dipresentasikan pada Gambar 5.5.



Gambar 5.5 Grafik Pengujian untuk Mengetahui Pengaruh Perbandingan Jumlah Data Latih dengan Data Uji terhadap Hasil Akurasi

Berdasarkan hasil pengujian terhadap perbandingan jumlah data latih dengan data uji didapatkan rata-rata tingkat akurasi tertinggi sebesar 98% dengan

perbandingan jumlah data latih dengan data uji sama dengan 90:10. Sedangkan rata-rata tingkat akurasi terendah yang didapatkan adalah sebesar 50% dengan perbandingan jumlah data latih dengan data uji sama dengan 10:90.

Secara keseluruhan, grafik pada Gambar 5.5 menunjukkan nilai akurasi yang menurun. Dari hasil tersebut diketahui bahwa semakin besar jumlah data uji dari data latih yang digunakan mengakibatkan tingkat akurasi semakin rendah. Hal ini juga dapat disimpulkan bahwa algoritma LVQ dapat melakukan proses klasifikasi dengan lebih baik jika menggunakan data latih dengan jumlah yang semakin besar karena variasi data latih yang digunakan pun menjadi lebih banyak.

5.6 Hasil dan Analisis Pengujian untuk Mengetahui Hasil Akurasi dari Proses Klasifikasi menggunakan *Minimum Distance*

Pengujian ini bertujuan untuk mendapatkan rata-rata tingkat akurasi dari sepuluh kali percobaan pada masing-masing perbandingan jumlah data latih dengan data uji serta proses klasifikasi yang dilakukan menggunakan *minimum distance*. Perbandingan jumlah data latih dengan data uji yang digunakan dalam pengujian ini meliputi 90:10, 80:20, 70:30, 60:40, 50:50, 40:60, 30:70, 20:80, dan 10:90. Data latih dan data uji yang digunakan merupakan data acak yang telah tersimpan dari hasil pengujian sebelumnya.

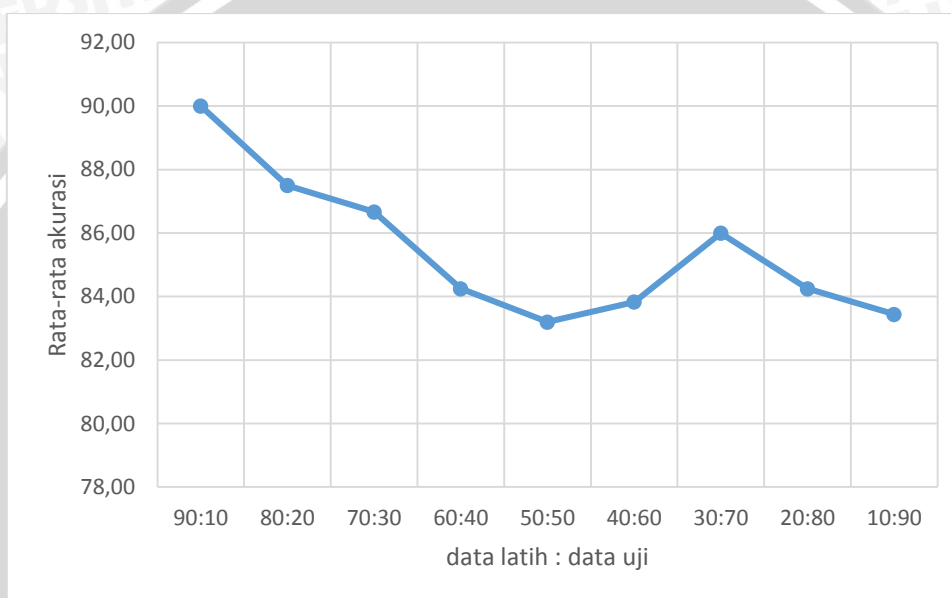
Pengujian ini dilakukan dengan menggunakan vektor bobot yang nilainya diperoleh dari rata-rata setiap fitur pada setiap kelas. Hasil pengujian untuk mengetahui hasil akurasi dari proses klasifikasi menggunakan *minimum distance* ditunjukkan pada Tabel 5.6.

Tabel 5.6 Pengujian untuk Mengetahui Hasil Akurasi dari Proses Klasifikasi menggunakan *Minimum Distance*

Data latih : data uji	Percobaan ke-										Rata-rata akurasi (%)
	1	2	3	4	5	6	7	8	9	10	
90:10	100	90	90	90	90	60	100	90	90	100	90
80:20	85	85	100	80	85	90	85	85	85	95	87,50
70:30	96, 67	86, 67	100	80	83, 33	83, 33	73, 33	93, 33	90	80	86,67
60:40	85	82, 50	85	80	82, 50	92, 50	82, 50	82, 50	82, 50	87, 50	84,25
50:50	76	86	78	82	84	88	86	84	84	84	83,20
40:60	81, 67	86, 67	78, 33	86, 67	86, 67	85	85	86, 67	78, 33	83, 33	83,83

30:70	87,14	80	81,43	91,43	87,14	82,86	88,57	88,57	87,14	85,71	86
20:80	88,75	87,50	83,75	85	86,25	86,25	80	76,25	81,25	87,50	84,25
10:90	81,11	88,89	82,22	81,11	88,89	80	70	85,56	90	86,67	83,44

Berdasarkan tabel hasil pengujian diatas, grafik pengujian untuk mengetahui hasil akurasi dari proses klasifikasi menggunakan *minimum distance* dipresentasikan pada Gambar 5.6.



Gambar 5.6 Grafik Pengujian untuk Mengetahui Hasil Akurasi dari Proses Klasifikasi menggunakan *Minimum Distance*

Secara keseluruhan pada hasil pengujian menunjukkan bahwa klasifikasi kualitas daun nilam dapat dilakukan dengan menggunakan *minimum distance* dengan rata-rata akurasi yang lebih tinggi jika dibandingkan menggunakan algoritma LVQ pada rasio data 60:40, 50:50, 40:60, 30:70, 20:80, dan 10:90. Sedangkan pada rasio data 90:10, 80:20, dan 70:10 dihasilkan rata-rata akurasi terbaik dari proses klasifikasi menggunakan algoritma LVQ. Hal ini menunjukkan bahwa algoritma LVQ dapat melakukan proses klasifikasi dengan lebih baik jika menggunakan rasio jumlah data latih dan data uji minimal sebesar 70:30 karena algoritma LVQ membutuhkan variasi data latih yang banyak untuk mengenali pola yang lebih banyak dalam proses pembelajarannya.

BAB VI PENUTUP

6.1 Kesimpulan

Dari hasil analisis dan pengujian yang telah dilakukan maka kesimpulan yang dapat diambil dari penelitian ini adalah:

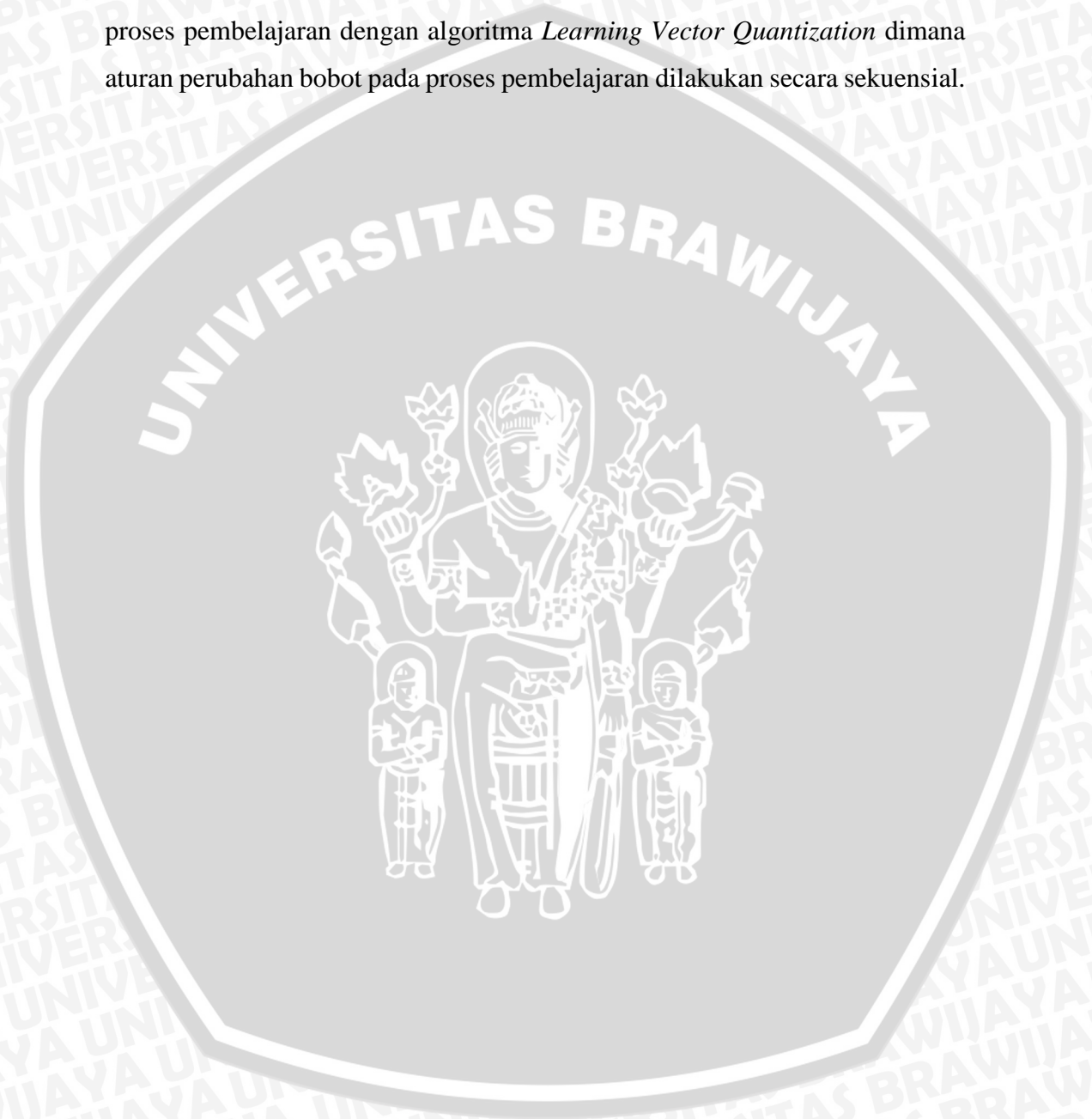
1. Algoritma *Learning Vector Quantization* (LVQ) dapat diterapkan untuk melakukan klasifikasi kualitas daun nilam menggunakan citra digital dengan melakukan tahapan-tahapan proses pengambilan data citra daun nilam dengan pemindaian, ekstraksi fitur warna, binerisasi, dan ekstraksi fitur luas terlebih dahulu, kemudian melakukan proses pelatihan dan pengujian menggunakan algoritma LVQ. Proses pelatihan menghasilkan bobot akhir yang kemudian digunakan kedalam proses pengujian sehingga dapat diperoleh hasil klasifikasi dari daun nilam yang diuji.
2. Hasil rata-rata akurasi terbaik yang didapatkan melalui pengujian sistem adalah sebesar 98% dari 10 data uji dengan menggunakan data latih sebanyak 90, laju pembelajaran=0.1, pengurang laju pembelajaran=0.3, laju pembelajaran minimum= 10^{-3} , dan jumlah iterasi=4.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan penelitian ini selanjutnya antara lain:

1. Pengambilan data citra menggunakan pemindaian belum efisien untuk dilakukan, maka perlu dikembangkan sistem dalam bentuk aplikasi *mobile* agar pengambilan data citra hingga proses klasifikasi dapat diakses oleh pengguna dengan lebih mudah, cepat, dan fleksibel.
2. Menambahkan fitur tekstur karena fitur tekstur sudah cukup banyak digunakan dalam penelitian yang mengangkat topik klasifikasi dengan objek citra daun.
3. Melakukan pengujian *threshold* untuk dapat mengoptimasi *preprocessing* data citra.

4. Melakukan optimalisasi inialisasi bobot awal karena adanya karakteristik dari proses pembelajaran dengan algoritma *Learning Vector Quantization* dimana inialisasi bobot awal pada proses pembelajaran dilakukan secara acak.
5. Melakukan optimalisasi urutan data latih karena adanya karakteristik dari proses pembelajaran dengan algoritma *Learning Vector Quantization* dimana aturan perubahan bobot pada proses pembelajaran dilakukan secara sekuensial.



DAFTAR PUSTAKA

- [ALA-07] Alam, P.N. 2007, “Aplikasi Proses Pengkelatan untuk Peningkatan Mutu Minyak Nilam Aceh”, *Jurnal Rekayasa Kimia dan Lingkungan*, Volume 6.
- [ANN-09] Anna, T. Dan Pakereng, M. A. I. Dan Beeh, Y. R. 2009, “Implementasi Algoritma *Chaos-Based Feedback Stream Cipher* pada Enkripsi-Dekripsi Data Citra Digital”, *Jurnal Informatika*, Volume 5.
- [ARN-12] Arnindya, R.C. 2012, “Peningkatan Kadar *Patchouli Alcohol* Dalam Pemurnian Minyak Nilam Dengan Menggunakan Teknologi Distilasi Vacuum Gelombang Mikro”, Tugas Akhir Program Studi Diploma III Teknik Kimia, Program Diploma Fakultas Teknik, Universitas Diponegoro. Semarang.
- [FAU-94] Fausett, L. 1994, “*Fundamentals of Neural Network: Architectures, Algorithms, and Applications*”, Prentice-Hall, Inc., New Jersey.
- [HAR-13] Haryono, A. dan Kustiyo, A. 2013, “Identifikasi Daun Tanaman Jati Menggunakan Jaringan Syaraf Tiruan *Backpropagation* dengan Ekstraksi Fitur Ciri Morfologi Daun”, Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Bogor.
- [ISM-13] Ismail, M. dan Mustikasari. 2013, “*Intelligent System for Tea Leaf Disease Detection*”, *Information Processing Society of Japan*.
- [KUS-03] Kusumadewi, S. 2003, “*Artificial Intelligence* (Teknik dan Aplikasinya)”, Graha Ilmu, Yogyakarta.
- [LES-12] Lestari, P. dan Hidayat, B. dan Susatio, E. 2012, “Deteksi Cacat Daun Teh *Camellia sinensis* dengan Pengolahan Citra Digital dan *JST Learning Vector Quantization*”, *The Indonesian Mathematical Society* (IndoMS), Volume 1, Sumedang.

- [MUN-10] Munawaroh, S. dan Susanto, F. A. 2010, "Pengolah Citra Digital untuk Identifikasi Uang Kertas", Jurnal Teknologi Informasi DINAMIK, Volume 15.
- [NUG-14] Nugroho, S. 2014, "Deteksi dan Pengenalan Plat Nomor Mobil Menggunakan *Vertical Edge Detection* dan *Backpropagation Neural Network*", Skripsi S-1 Program Studi Informatika/Illmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang.
- [NUR-11] Nurkhozin, A. dan Irawan, M.I. dan Mukhlas, I. 2011, "Komparasi Hasil Klasifikasi Penyakit Diabetes Mellitus menggunakan Jaringan Syaraf Tiruan *Backpropagation* dan *Learning Vector Quantization*", Prosiding Seminar Nasional Penelitian, Pendidikan dan Penerapan MIPA, Fakultas MIPA, Universitas Negeri Yogyakarta, Yogyakarta.
- [PUT-10] Putra, D. 2010, "Pengolahan Citra Digital", ANDI, Yogyakarta.
- [PRA-06] Prawoto, A. A. dan Sholeh, M. 2006, "Produksi Awal dan Kajian Ekonomis Usahatani Nilam Aceh (*Pogostemon cablin Benth.*) Sebagai Tanaman Sela Kakao Muda", Pelita Perkebunan, Volume 22.
- [QUR-10] Qur'ani, D.Y. dan Rosmalinda, S. 2010, "Jaringan Syaraf Tiruan *Learning Vector Quantization* untuk Aplikasi Pengenalan Tanda Tangan", Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010), Yogyakarta.
- [SAR-11] Saraswati, D.A. dan Setiawardhana. 2011, "Sistem Pendeteksian Bakteri dengan Histogram Citra Biner", Jurnal Matematika dan Ilmu Pengetahuan Alam, Volume 14.
- [SIA-11] Siahaan, A. 2011. "Pengenalan Karakter dan Manajemen Database pada Formulir Isian Menggunakan Jaringan Syaraf Tiruan dengan Metode *Learning Vector Quantization (LVQ)*", Skripsi S-1 Jurusan Ilmu Komputer, Departemen Ilmu Komputer, Fakultas Matematika Dan Ilmu Pengetahuan Alam, Universitas Sumatera Utara, Medan.
- [SIG-13] Singh, A. dan Yadav, A. dan Rana, A. 2013, "*K-means with Three different Distance Metrics*", *International Journal of Computer Applications*, Volume 67.





- [SIL-06] Silviana dan Purbasari, A. 2006, “Studi Awal Deterpenisasi Minyak Nilam dengan Teknologi Redistilasi Vakum”, Reaktor, Volume 10.
- [SIN-14] Sinaga, D.K. 2014, “Implementasi *Learning Vector Quantization* (LVQ) dalam Mengenal Pertumbuhan Jamur”, Skripsi S-1 Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia.
- [SUP-11] Supriadi dan Rizal, M. dan Wahyuno, D. 2011, “Nilam (*Patchouli*)”, Badan Penelitian Tanaman Obat dan Aromatik, Bogor.
- [ZHA-13] Zhang, F. 2013, “*Image Recognition of Wheat Leaf Disease based on Learning Vector Quantization Neural Network and Combination Feature Parameters*”, *Journal of Convergence Information Technology* (JCIT), Volume 8.








LAMPIRAN






Dataset citra daun nilam dengan kelas kualitas baik yang digunakan dalam penelitian ini adalah sebagai berikut:


				
1-1.jpg	1-2.jpg	1-3.jpg	1-4.jpg	1-5.jpg






				
1-6.jpg	1-7.jpg	1-8.jpg	1-9.jpg	1-10.jpg






				
1-11.jpg	1-12.jpg	1-13.jpg	1-14.jpg	1-15.jpg





				
1-16.jpg	1-17.jpg	1-18.jpg	1-19.jpg	1-20.jpg




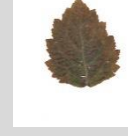

				
1-21.jpg	1-22.jpg	1-23.jpg	1-24.jpg	1-25.jpg

				
1-26.jpg	1-27.jpg	1-28.jpg	1-29.jpg	1-30.jpg

				
1-31.jpg	1-32.jpg	1-33.jpg	1-34.jpg	1-35.jpg




				
1-36.jpg	1-37.jpg	1-38.jpg	1-39.jpg	1-40.jpg






				
1-41.jpg	1-42.jpg	1-43.jpg	1-44.jpg	1-45.jpg



				
1-46.jpg	1-47.jpg	1-48.jpg	1-49.jpg	1-50.jpg






Dataset citra daun nilam dengan kelas kualitas kurang baik yang digunakan dalam penelitian ini adalah sebagai berikut:

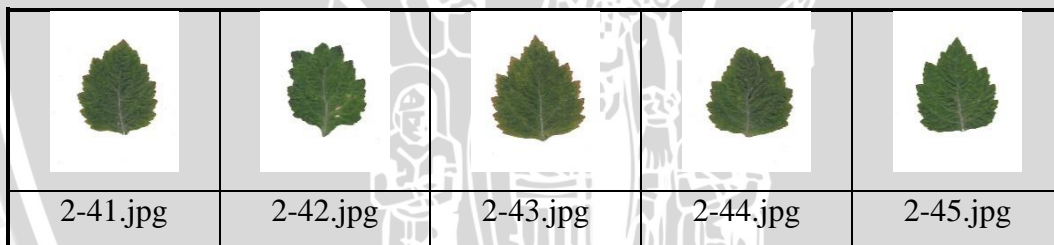
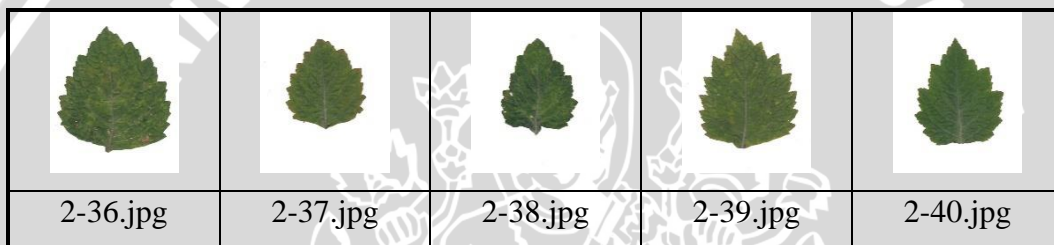
				
2-1.jpg	2-2.jpg	2-3.jpg	2-4.jpg	2-5.jpg

				
2-6.jpg	2-7.jpg	2-8.jpg	2-9.jpg	2-10.jpg

				
2-11.jpg	2-12.jpg	2-13.jpg	2-14.jpg	2-15.jpg

				
2-16.jpg	2-17.jpg	2-18.jpg	2-19.jpg	2-20.jpg

				
2-21.jpg	2-22.jpg	2-23.jpg	2-24.jpg	2-25.jpg



Pada skenario pengujian 1 sampai dengan 4, data latih dan data uji yang digunakan telah ditetapkan dengan jumlah masing-masing adalah sebesar 70 dan 30. Id citra yang digunakan sebagai data latih meliputi “1-1.jpg” urut sampai dengan “1-35.jpg” dan “2-1.jpg” urut sampai dengan “2-35.jpg”, sedangkan sisanya digunakan sebagai data uji.

Nilai bobot awal yang diinisialisasi secara acak pada skenario pengujian 1 dengan nilai laju pembelajaran= 0,1 adalah sebagai berikut:

Percobaan ke-	Bobot awal				
1	w ₁	0.8727058141493215	0.7020201916791098	0.11269848332843369	0.4855970926440738
	w ₂	0.9692370632201275	0.7694961279378557	0.7120297626817829	0.015544619898862688
2	w ₁	0.6305820013765334	0.9677145038976754	0.4345720375025832	0.580643968381088
	w ₂	0.7900886641579039	0.4736025619741647	0.13628276451137222	0.5594538636330411
3	w ₁	0.6976715206116999	0.8387561218678372	0.8523643793575969	0.9564560031465181
	w ₂	0.8237917171711626	0.6840555994670142	0.3626087431536613	0.9886915943712747
4	w ₁	0.21242132748593268	0.5410361288032431	0.6190406497203648	0.2741743006329188
	w ₂	0.7495426462312926	0.7360450205620988	0.3707484681022427	0.4391299501730369
5	w ₁	0.48061704070573286	0.0709979522962186	0.10375207193416791	0.9370828282819981
	w ₂	0.9231989209850573	0.7055289060612878	0.49514200392450236	0.07614234467458914
6	w ₁	0.39769016839013027	0.6314484191376482	0.8751842615744951	0.2875584301937877
	w ₂	0.82547268493652	0.8456625905588544	0.5480794209344365	0.9662436565887665
7	w ₁	0.369865414495398	0.3445294759656957	0.04351146200498879	0.49719694230098266
	w ₂	0.9196091292297909	0.6563454251501728	0.12772710769622375	0.2634887986254735
8	w ₁	0.0661795118946128	0.8863816656142017	0.8179878859825341	0.4196470230824628
	w ₂	0.7818309975982374	0.26266637113139835	0.5993108939729029	0.011410629509404191
9	w ₁	0.583403062206945	0.9380309593961402	0.48720774957714963	0.9256827207688502
	w ₂	0.6178305223641227	0.02802979256064564	0.4877370325470295	0.11127901096182125
10	w ₁	0.855896389359695	0.5970298638295738	0.2790647837349213	0.4900458315205696
	w ₂	0.5285899311851566	0.5777427447706609	0.2699324943968209	0.25746156354097705

Nilai bobot awal yang diinisialisasi secara acak pada skenario pengujian 2 dengan nilai konstanta pengurang laju pembelajaran= 0,3 adalah sebagai berikut:

Percobaan ke-	Bobot awal				
1	w ₁	0.5796673314571192	0.8621468918173055	0.05855499631749095	0.3686994925578244
	w ₂	0.3421110467160504	0.42000410098989815	0.021059494677334367	0.36983113336929374
2	w ₁	0.6616068200771147	0.16584421542577077	0.017869827566357244	0.5745460482584287
	w ₂	0.6516244823068122	0.3908102595004136	0.6928516509101923	0.6994030365743273
3	w ₁	0.031009787695436497	0.32465801769693126	0.9690197744286095	0.21981966489182414
	w ₂	0.03902527192348393	0.4934584632221263	0.030260056326513407	0.8443341709295797
4	w ₁	0.4047863282820572	0.8770444561039773	0.6106933434178776	0.13099264928598697
	w ₂	0.9989560471797858	0.6349667203943712	0.8266341095703997	0.8841311899580805
5	w ₁	0.14813882746492513	0.7837456813205581	0.6867852401630211	0.1279698229660039
	w ₂	0.9436779843530535	0.8332636136218539	0.5687658978519874	0.1146022544986931
6	w ₁	0.5385589760039331	0.07467569272709496	0.997860607677953	0.776351546546591
	w ₂	0.7552835676021321	0.16803670578494245	0.9887671505438944	0.8289363523675728
7	w ₁	0.3421418158001338	0.35206838681206665	0.2659515641022845	0.6932246067521749
	w ₂	0.6624332681594409	0.670362994115155	0.7471108816283998	0.250454891909369
8	w ₁	0.5967477630246074	0.35691920591999204	0.3722176718132547	0.23851219656496725
	w ₂	0.36739726894913893	0.8894021926418091	0.7976186883971605	0.9803170816636861

9	w ₁	0.746544197426603	0.05575127698248106	0.7370013923894125	0.673914341936049
	w ₂	0.5991837552388103	0.12443419318521987	0.23517528847817115	0.5734599266441837
10	w ₁	0.5224575847571021	0.9547376339485906	0.8618096601376125	0.12146961067912421
	w ₂	0.5185348635236995	0.5605221780064084	0.67695110612958	0.10605458442383886

Nilai bobot awal yang diinisialisasi secara acak pada skenario pengujian 3 dengan nilai laju pembelajaran minimum= 10^{-3} adalah sebagai berikut:

Percobaan ke-	Bobot awal				
1	w ₁	0.20677088612386563	0.5624516332161416	0.4338384380523146	0.1540574183800678
	w ₂	0.4586675597854404	0.5271948183615374	0.7798254097683751	0.9863766117479268
2	w ₁	0.1738895258524622	0.555264532821423	0.8759867346393868	0.21421722221225337
	w ₂	0.4911020320033249	0.5440474068255527	0.7836674381985909	0.9897769328448378
3	w ₁	0.042664606391033866	0.5002412178125499	0.7920731309499927	0.019277921490379324
	w ₂	0.4276497485650471	0.8569055575418021	0.8219954562756568	0.9919955555508662
4	w ₁	0.8830395012431077	0.7215368424157457	0.7046655332261801	0.6647998118646211
	w ₂	0.7215502197662006	0.38580752880526614	0.45181520635973915	0.31697986710650894
5	w ₁	0.6197736909346921	0.13763169646453777	0.09545943740160379	0.3438215203675645
	w ₂	0.12984238456656028	0.6457222801783625	0.5239893483786523	0.7872167379302842
6	w ₁	0.25420360514194174	0.0026992953778688378	0.8354432761165019	0.029543034245653854
	w ₂	0.7057629073497852	0.4930524114792886	0.5737375427759251	0.07390372988682603
7	w ₁	0.07019521772446746	0.9261120353293785	0.6402711953558256	0.6334114953278233
	w ₂	0.6625434146388463	0.10779373805814951	0.3891760749277998	0.9956674273376337
8	w ₁	0.7188198011846908	0.3771150312980397	0.07037462439007935	0.4706410649375501
	w ₂	0.034387756515629087	0.9496328488109664	0.3086782715530346	0.12911049260695884
9	w ₁	0.326284176293127	0.1316998770027057	0.845343931917923	0.11519828728915038
	w ₂	0.3104109027590962	0.6762640269092715	0.047836993166989306	0.34446873829621305
10	w ₁	0.019646513222710937	0.36870923498339214	0.4921111968139653	0.9762837869879782
	w ₂	0.46732031258129725	0.262206514616933	0.10681035624631785	0.129659020867251

Nilai bobot awal yang diinisialisasi secara acak pada skenario pengujian 4 dengan jumlah iterasi= 4 adalah sebagai berikut:

Percobaan ke-	Bobot awal				
1	w ₁	0.12916649614117193	0.37282034770178485	0.03871682984909863	0.15795089762561232
	w ₂	0.6563918434483289	0.3995303853662391	0.2592663770861583	0.0748711424797377
2	w ₁	0.430640358819426	0.36852791568511367	0.48277579072718013	0.2536539959999008
	w ₂	0.8743981986622227	0.5793160135532172	0.36573090211006565	0.49103170225373205
3	w ₁	0.9179453840822362	0.8385168095181051	0.797581883471648	0.852776333443445
	w ₂	0.5545357608371094	0.9496752025635721	0.17269143931664777	0.7460860071021236
4	w ₁	0.9884788237784544	0.21215670251969554	0.7183440835115378	0.45573900302335124
	w ₂	0.6897176188874081	0.7020675031568446	0.8005795761405551	0.9929989067774163
5	w ₁	0.7735320215731095	0.6322714103484192	0.18627471822715402	0.8758840828886764
	w ₂	0.31929018785097585	0.10215022048973044	0.5875920426256305	0.03795691666970202
6	w ₁	0.7089280726380688	0.3149154863233277	0.9791942331586194	0.01712838650771964

	w2	0.2785114853773978	0.2941428031490062	0.6513333803887507	0.7874631862459487
7	w1	0.04297644323478533	0.3904128475897837	0.8204684979664696	0.8301448985815862
	w2	0.8975534452623806	0.6689726116600414	0.20662315328543202	0.47210468363703273
8	w1	0.17627568848863073	0.43268900349452166	0.4858097774052762	0.27910348476232605
	w2	0.5758093080732157	0.5344839940662776	0.19024895030107036	0.010273227807317653
9	w1	0.9778956487981824	0.13392789501581204	0.6774728539244398	0.40100173687233953
	w2	0.4072875200657624	0.07599133330164176	0.9934463753123278	0.12512110939720045
10	w1	0.4724701663214137	0.8315846707438879	0.685377211090298	0.7896175939074197
	w2	0.2828732623943019	0.8454578076867636	0.3528874754082131	0.07803302853097493

Nilai bobot awal yang diinisialisasi secara acak pada skenario pengujian 5 dengan perbandingan jumlah data latih dengan data uji= 90:10 adalah sebagai berikut:

Percobaan ke-	Bobot awal				
1	w1	0.45346898866971874	0.5982175997220105	0.6905107131581625	0.4335238943399513
	w2	0.5974788359153355	0.28302532914156	0.5693041723021485	0.8516415882975031
2	w1	0.543521847565361	0.6386438041372342	0.24205276287157873	0.7132196893818208
	w2	0.17241099526980308	0.889455312981038	0.9037698828920273	0.6778909053209781
3	w1	0.6643846050400323	0.35859549322204365	0.25819553606354295	0.5718808979005193
	w2	0.8642190527622653	0.715349379523533	0.295257363926375	0.10631054091582304
4	w1	0.03444189266291131	0.19174507858356216	0.16713476744496136	0.03122032428492849
	w2	0.8158079443585378	0.25177228694974685	0.5647441315781946	0.9506939236442183
5	w1	0.6916652064591416	0.13660904903800752	0.9844223154052697	0.9864661572341452
	w2	0.4311563474090687	0.8048667450968886	0.07095852631951449	0.9834516258241922
6	w1	0.42716403808243997	0.949498521375516	0.7614032891220257	0.6628828460305394
	w2	0.5559531799562119	0.8203239220549894	0.16522046601351592	0.5257728268170772
7	w1	0.9523264152985237	0.30520611087394	0.9590373431399047	0.29005522779380843
	w2	0.27543545739529474	0.1206303589818093	0.933963081252554	0.5548710071196953
8	w1	0.7454260782284251	0.2328796615857217	0.6623788745464194	0.5619029782745034
	w2	0.15045293319353825	0.54137351043143	0.170110580261578	0.7796912402406027
9	w1	0.7794513814731312	0.3909514049317465	0.7297566951514218	0.048549574507007254
	w2	0.830887714582383	0.9039172575391882	0.7508042114513716	0.6089327549985034
10	w1	0.7854302097758068	0.7947863502096373	0.8563129241654691	0.8923844637980591
	w2	0.5600690724684765	0.6267998589862759	0.8922206901366735	0.048748012037850175

Id citra pada data latih dan data uji yang diambil secara acak pada skenario pengujian 5 dengan perbandingan jumlah data latih dengan data uji= 90:10 adalah sebagai berikut:

Percobaan ke-		Kelas							
		Kualitas baik				Kualitas kurang baik			
1	Data latih	1-33.jpg	1-9.jpg	1-46.jpg	1-30.jpg	2-20.jpg	2-15.jpg	2-16.jpg	2-22.jpg
		1-24.jpg	1-35.jpg	1-3.jpg	1-29.jpg	2-28.jpg	2-2.jpg	2-23.jpg	2-33.jpg
		1-20.jpg	1-41.jpg	1-27.jpg	1-22.jpg	2-46.jpg	2-34.jpg	2-5.jpg	2-32.jpg
		1-16.jpg	1-15.jpg	1-23.jpg	1-26.jpg	2-10.jpg	2-17.jpg	2-14.jpg	2-26.jpg
		1-14.jpg	1-19.jpg	1-43.jpg	1-12.jpg	2-8.jpg	2-47.jpg	2-3.jpg	2-40.jpg
		1-8.jpg	1-45.jpg	1-32.jpg	1-36.jpg	2-48.jpg	2-44.jpg	2-6.jpg	2-41.jpg
		1-38.jpg	1-48.jpg	1-47.jpg	1-31.jpg	2-13.jpg	2-29.jpg	2-27.jpg	2-9.jpg
		1-11.jpg	1-44.jpg	1-39.jpg	1-2.jpg	2-19.jpg	2-38.jpg	2-36.jpg	2-7.jpg
		1-49.jpg	1-6.jpg	1-40.jpg	1-42.jpg	2-45.jpg	2-49.jpg	2-37.jpg	2-25.jpg
		1-25.jpg	1-5.jpg	1-4.jpg	1-34.jpg	2-43.jpg	2-11.jpg	2-39.jpg	2-50.jpg
		1-7.jpg	1-17.jpg	1-21.jpg	1-28.jpg	2-42.jpg	2-4.jpg	2-31.jpg	2-18.jpg
		1-18.jpg				2-35.jpg			
	Data uji	1-26.jpg	1-10.jpg	1-50.jpg	1-1.jpg	2-12.jpg	2-30.jpg	2-24.jpg	2-1.jpg
		1-37.jpg				2-21.jpg			
2	Data latih	1-40.jpg	1-19.jpg	1-1.jpg	1-20.jpg	2-2.jpg	2-12.jpg	2-18.jpg	2-46.jpg
		1-11.jpg	1-28.jpg	1-9.jpg	1-17.jpg	2-36.jpg	2-49.jpg	2-44.jpg	2-23.jpg
		1-31.jpg	1-8.jpg	1-14.jpg	1-25.jpg	2-22.jpg	2-8.jpg	2-1.jpg	2-24.jpg
		1-33.jpg	1-49.jpg	1-35.jpg	1-36.jpg	2-3.jpg	2-14.jpg	2-16.jpg	2-37.jpg
		1-41.jpg	1-5.jpg	1-34.jpg	1-3.jpg	2-21.jpg	2-42.jpg	2-34.jpg	2-10.jpg
		1-12.jpg	1-7.jpg	1-10.jpg	1-27.jpg	2-4.jpg	2-47.jpg	2-35.jpg	2-50.jpg
		1-4.jpg	1-24.jpg	1-13.jpg	1-46.jpg	2-17.jpg	2-41.jpg	2-20.jpg	2-40.jpg
		1-37.jpg	1-16.jpg	1-30.jpg	1-43.jpg	2-27.jpg	2-11.jpg	2-45.jpg	2-39.jpg
		1-6.jpg	1-45.jpg	1-29.jpg	1-18.jpg	2-6.jpg	2-19.jpg	2-28.jpg	2-33.jpg
		1-50.jpg	1-48.jpg	1-39.jpg	1-26.jpg	2-38.jpg	2-32.jpg	2-31.jpg	2-48.jpg
		1-21.jpg	1-44.jpg	1-23.jpg	1-22.jpg	2-43.jpg	2-29.jpg	2-26.jpg	2-7.jpg
		1-47.jpg				2-13.jpg			
	Data uji	1-14.jpg	1-32.jpg	1-42.jpg	1-38.jpg	2-9.jpg	2-25.jpg	2-5.jpg	2-15.jpg
		1-2.jpg				2-30.jpg			
3	Data latih	1-26.jpg	1-49.jpg	1-46.jpg	1-32.jpg	2-17.jpg	2-32.jpg	2-16.jpg	2-39.jpg
		1-4.jpg	1-34.jpg	1-21.jpg	1-50.jpg	2-20.jpg	2-47.jpg	2-28.jpg	2-14.jpg
		1-3.jpg	1-43.jpg	1-41.jpg	1-20.jpg	2-44.jpg	2-19.jpg	2-50.jpg	2-35.jpg
		1-5.jpg	1-2.jpg	1-14.jpg	1-1.jpg	2-22.jpg	2-5.jpg	2-27.jpg	2-49.jpg
		1-35.jpg	1-30.jpg	1-27.jpg	1-16.jpg	2-33.jpg	2-21.jpg	2-11.jpg	2-42.jpg
		1-38.jpg	1-29.jpg	1-8.jpg	1-18.jpg	2-13.jpg	2-43.jpg	2-31.jpg	2-24.jpg
		1-31.jpg	1-47.jpg	1-6.jpg	1-12.jpg	2-23.jpg	2-48.jpg	2-46.jpg	2-6.jpg
		1-10.jpg	1-44.jpg	1-24.jpg	1-22.jpg	2-8.jpg	2-9.jpg	2-25.jpg	2-29.jpg
		1-45.jpg	1-19.jpg	1-17.jpg	1-11.jpg	2-45.jpg	2-2.jpg	2-7.jpg	2-38.jpg
		1-48.jpg	1-40.jpg	1-15.jpg	1-28.jpg	2-34.jpg	2-1.jpg	2-26.jpg	2-30.jpg
		1-7.jpg	1-13.jpg	1-25.jpg	1-9.jpg	2-41.jpg	2-12.jpg	2-15.jpg	2-37.jpg
		1-42.jpg				2-3.jpg			
	Data uji	1-15.jpg	1-23.jpg	1-37.jpg	1-36.jpg	2-36.jpg	2-40.jpg	2-18.jpg	2-4.jpg
		1-39.jpg				2-10.jpg			
4	Data latih	1-39.jpg	1-41.jpg	1-45.jpg	1-36.jpg	2-11.jpg	2-41.jpg	2-45.jpg	2-17.jpg
		1-19.jpg	1-15.jpg	1-35.jpg	1-48.jpg	2-4.jpg	2-27.jpg	2-28.jpg	2-35.jpg
		1-23.jpg	1-28.jpg	1-31.jpg	1-6.jpg	2-7.jpg	2-33.jpg	2-26.jpg	2-40.jpg
		1-10.jpg	1-17.jpg	1-7.jpg	1-13.jpg	2-50.jpg	2-8.jpg	2-43.jpg	2-3.jpg
		1-5.jpg	1-3.jpg	1-25.jpg	1-18.jpg	2-37.jpg	2-36.jpg	2-5.jpg	2-39.jpg
		1-42.jpg	1-46.jpg	1-47.jpg	1-44.jpg	2-9.jpg	2-19.jpg	2-22.jpg	2-29.jpg
		1-11.jpg	1-22.jpg	1-33.jpg	1-16.jpg	2-49.jpg	2-42.jpg	2-38.jpg	2-15.jpg
		1-37.jpg	1-27.jpg	1-20.jpg	1-43.jpg	2-23.jpg	2-34.jpg	2-46.jpg	2-48.jpg
		1-38.jpg	1-1.jpg	1-24.jpg	1-40.jpg	2-30.jpg	2-44.jpg	2-2.jpg	2-31.jpg

		1-8.jpg	1-4.jpg	1-50.jpg	1-49.jpg	2-16.jpg	2-24.jpg	2-25.jpg	2-12.jpg
		1-14.jpg	1-34.jpg	1-26.jpg	1-29.jpg	2-6.jpg	2-18.jpg	2-1.jpg	2-13.jpg
		1-32.jpg				2-32.jpg			
	Data uji	1-16.jpg	1-21.jpg	1-9.jpg	1-30.jpg	2-47.jpg	2-21.jpg	2-20.jpg	2-10.jpg
		1-2.jpg				2-14.jpg			
5	Data latih	1-46.jpg	1-30.jpg	1-25.jpg	1-50.jpg	2-22.jpg	2-11.jpg	2-13.jpg	2-23.jpg
		1-13.jpg	1-20.jpg	1-14.jpg	1-5.jpg	2-26.jpg	2-27.jpg	2-4.jpg	2-41.jpg
		1-3.jpg	1-47.jpg	1-11.jpg	1-15.jpg	2-5.jpg	2-46.jpg	2-7.jpg	2-34.jpg
		1-22.jpg	1-29.jpg	1-24.jpg	1-33.jpg	2-43.jpg	2-21.jpg	2-33.jpg	2-18.jpg
		1-10.jpg	1-2.jpg	1-36.jpg	1-12.jpg	2-32.jpg	2-30.jpg	2-9.jpg	2-17.jpg
		1-43.jpg	1-16.jpg	1-31.jpg	1-42.jpg	2-1.jpg	2-40.jpg	2-28.jpg	2-2.jpg
		1-21.jpg	1-18.jpg	1-39.jpg	1-40.jpg	2-29.jpg	2-31.jpg	2-35.jpg	2-16.jpg
		1-23.jpg	1-38.jpg	1-19.jpg	1-6.jpg	2-6.jpg	2-36.jpg	2-20.jpg	2-39.jpg
		1-45.jpg	1-35.jpg	1-26.jpg	1-27.jpg	2-50.jpg	2-48.jpg	2-44.jpg	2-3.jpg
		1-4.jpg	1-48.jpg	1-28.jpg	1-32.jpg	2-42.jpg	2-25.jpg	2-37.jpg	2-19.jpg
		1-49.jpg	1-1.jpg	1-9.jpg	1-41.jpg	2-12.jpg	2-8.jpg	2-45.jpg	2-47.jpg
		1-8.jpg				2-10.jpg			
	Data uji	1-47.jpg	1-7.jpg	1-37.jpg	1-44.jpg	2-15.jpg	2-38.jpg	2-14.jpg	2-24.jpg
		1-17.jpg				2-49.jpg			
6	Data latih	1-2.jpg	1-9.jpg	1-19.jpg	1-11.jpg	2-43.jpg	2-1.jpg	2-10.jpg	2-28.jpg
		1-35.jpg	1-25.jpg	1-31.jpg	1-23.jpg	2-31.jpg	2-41.jpg	2-46.jpg	2-25.jpg
		1-21.jpg	1-3.jpg	1-45.jpg	1-46.jpg	2-20.jpg	2-7.jpg	2-18.jpg	2-8.jpg
		1-13.jpg	1-44.jpg	1-39.jpg	1-4.jpg	2-40.jpg	2-36.jpg	2-50.jpg	2-44.jpg
		1-30.jpg	1-32.jpg	1-14.jpg	1-37.jpg	2-35.jpg	2-6.jpg	2-26.jpg	2-24.jpg
		1-43.jpg	1-15.jpg	1-27.jpg	1-17.jpg	2-16.jpg	2-29.jpg	2-12.jpg	2-4.jpg
		1-48.jpg	1-22.jpg	1-41.jpg	1-8.jpg	2-39.jpg	2-9.jpg	2-22.jpg	2-15.jpg
		1-42.jpg	1-34.jpg	1-49.jpg	1-5.jpg	2-27.jpg	2-32.jpg	2-3.jpg	2-14.jpg
		1-50.jpg	1-26.jpg	1-28.jpg	1-6.jpg	2-48.jpg	2-11.jpg	2-2.jpg	2-21.jpg
		1-7.jpg	1-47.jpg	1-40.jpg	1-33.jpg	2-19.jpg	2-5.jpg	2-49.jpg	2-42.jpg
		1-10.jpg	1-38.jpg	1-1.jpg	1-36.jpg	2-47.jpg	2-30.jpg	2-13.jpg	2-34.jpg
		1-12.jpg				2-17.jpg			
	Data uji	1-5.jpg	1-29.jpg	1-18.jpg	1-24.jpg	2-37.jpg	2-33.jpg	2-23.jpg	2-38.jpg
		1-20.jpg				2-45.jpg			
7	Data latih	1-24.jpg	1-10.jpg	1-14.jpg	1-45.jpg	2-21.jpg	2-46.jpg	2-28.jpg	2-9.jpg
		1-38.jpg	1-9.jpg	1-21.jpg	1-11.jpg	2-33.jpg	2-49.jpg	2-4.jpg	2-36.jpg
		1-1.jpg	1-8.jpg	1-44.jpg	1-46.jpg	2-24.jpg	2-45.jpg	2-50.jpg	2-12.jpg
		1-33.jpg	1-47.jpg	1-23.jpg	1-31.jpg	2-1.jpg	2-19.jpg	2-44.jpg	2-25.jpg
		1-16.jpg	1-2.jpg	1-50.jpg	1-43.jpg	2-40.jpg	2-43.jpg	2-17.jpg	2-31.jpg
		1-35.jpg	1-18.jpg	1-4.jpg	1-22.jpg	2-22.jpg	2-41.jpg	2-18.jpg	2-35.jpg
		1-42.jpg	1-7.jpg	1-26.jpg	1-15.jpg	2-14.jpg	2-47.jpg	2-30.jpg	2-20.jpg
		1-3.jpg	1-13.jpg	1-32.jpg	1-49.jpg	2-37.jpg	2-16.jpg	2-39.jpg	2-3.jpg
		1-25.jpg	1-41.jpg	1-6.jpg	1-28.jpg	2-29.jpg	2-38.jpg	2-23.jpg	2-7.jpg
		1-30.jpg	1-40.jpg	1-48.jpg	1-5.jpg	2-6.jpg	2-34.jpg	2-2.jpg	2-8.jpg
		1-20.jpg	1-19.jpg	1-12.jpg	1-34.jpg	2-27.jpg	2-42.jpg	2-10.jpg	2-11.jpg
		1-29.jpg				2-5.jpg			
	Data uji	1-3.jpg	1-36.jpg	1-39.jpg	1-37.jpg	2-15.jpg	2-48.jpg	2-32.jpg	2-26.jpg
		1-17.jpg				2-13.jpg			
8	Data latih	1-2.jpg	1-8.jpg	1-12.jpg	1-16.jpg	2-34.jpg	2-50.jpg	2-21.jpg	2-8.jpg
		1-35.jpg	1-38.jpg	1-49.jpg	1-44.jpg	2-42.jpg	2-18.jpg	2-33.jpg	2-3.jpg
		1-40.jpg	1-1.jpg	1-18.jpg	1-30.jpg	2-29.jpg	2-20.jpg	2-9.jpg	2-6.jpg
		1-13.jpg	1-28.jpg	1-14.jpg	1-37.jpg	2-27.jpg	2-49.jpg	2-4.jpg	2-11.jpg
		1-31.jpg	1-39.jpg	1-19.jpg	1-23.jpg	2-13.jpg	2-1.jpg	2-24.jpg	2-40.jpg

		1-42.jpg 1-6.jpg 1-4.jpg 1-15.jpg 1-22.jpg 1-32.jpg 1-26.jpg	1-11.jpg 1-5.jpg 1-20.jpg 1-34.jpg 1-24.jpg 1-45.jpg	1-25.jpg 1-41.jpg 1-17.jpg 1-36.jpg 1-21.jpg 1-43.jpg	1-29.jpg 1-50.jpg 1-7.jpg 1-9.jpg 1-48.jpg 1-47.jpg	2-48.jpg 2-7.jpg 2-30.jpg 2-31.jpg 2-43.jpg 2-17.jpg 2-35.jpg	2-22.jpg 2-26.jpg 2-45.jpg 2-39.jpg 2-12.jpg 2-37.jpg	2-16.jpg 2-25.jpg 2-23.jpg 2-44.jpg 2-46.jpg 2-10.jpg	2-15.jpg 2-19.jpg 2-41.jpg 2-36.jpg 2-47.jpg 2-2.jpg
	Data uji	1-31.jpg 1-10.jpg	1-27.jpg	1-46.jpg	1-3.jpg	2-38.jpg 2-28.jpg	2-14.jpg	2-5.jpg	2-32.jpg
9	Data latih	1-44.jpg 1-27.jpg 1-2.jpg 1-29.jpg 1-7.jpg 1-19.jpg 1-43.jpg 1-33.jpg 1-5.jpg 1-4.jpg 1-22.jpg 1-10.jpg	1-18.jpg 1-45.jpg 1-14.jpg 1-49.jpg 1-17.jpg 1-37.jpg 1-3.jpg 1-42.jpg 1-8.jpg 1-15.jpg 1-50.jpg	1-31.jpg 1-11.jpg 1-38.jpg 1-34.jpg 1-28.jpg 1-24.jpg 1-25.jpg 1-6.jpg 1-20.jpg 1-9.jpg 1-35.jpg	1-30.jpg 1-12.jpg 1-40.jpg 1-39.jpg 1-1.jpg 1-26.jpg 1-16.jpg 1-32.jpg 1-41.jpg 1-23.jpg 1-48.jpg	2-42.jpg 2-2.jpg 2-39.jpg 2-28.jpg 2-26.jpg 2-1.jpg 2-36.jpg 2-20.jpg 2-31.jpg 2-50.jpg 2-24.jpg 2-21.jpg	2-37.jpg 2-7.jpg 2-23.jpg 2-45.jpg 2-35.jpg 2-48.jpg 2-5.jpg 2-40.jpg 2-34.jpg 2-15.jpg 2-44.jpg	2-6.jpg 2-17.jpg 2-46.jpg 2-3.jpg 2-33.jpg 2-12.jpg 2-22.jpg 2-32.jpg 2-43.jpg 2-14.jpg 2-49.jpg	2-16.jpg 2-18.jpg 2-10.jpg 2-27.jpg 2-13.jpg 2-29.jpg 2-19.jpg 2-41.jpg 2-47.jpg 2-11.jpg 2-25.jpg
	Data uji	1-2.jpg 1-13.jpg	1-47.jpg	1-46.jpg	1-36.jpg	2-4.jpg 2-9.jpg	2-8.jpg	2-30.jpg	2-38.jpg
10	Data latih	1-44.jpg 1-23.jpg 1-40.jpg 1-19.jpg 1-41.jpg 1-36.jpg 1-47.jpg 1-16.jpg 1-31.jpg 1-8.jpg 1-42.jpg 1-14.jpg	1-12.jpg 1-49.jpg 1-2.jpg 1-39.jpg 1-7.jpg 1-10.jpg 1-3.jpg 1-9.jpg 1-46.jpg 1-29.jpg 1-45.jpg	1-20.jpg 1-13.jpg 1-1.jpg 1-11.jpg 1-4.jpg 1-43.jpg 1-30.jpg 1-38.jpg 1-21.jpg 1-18.jpg 1-6.jpg	1-25.jpg 1-33.jpg 1-5.jpg 1-15.jpg 1-24.jpg 1-35.jpg 1-28.jpg 1-22.jpg 1-26.jpg 1-34.jpg 1-27.jpg	2-32.jpg 2-42.jpg 2-4.jpg 2-26.jpg 2-29.jpg 2-6.jpg 2-36.jpg 2-7.jpg 2-46.jpg 2-20.jpg 2-2.jpg 2-10.jpg	2-22.jpg 2-50.jpg 2-18.jpg 2-44.jpg 2-24.jpg 2-28.jpg 2-8.jpg 2-27.jpg 2-14.jpg 2-45.jpg 2-34.jpg	2-40.jpg 2-25.jpg 2-9.jpg 2-48.jpg 2-12.jpg 2-49.jpg 2-15.jpg 2-39.jpg 2-11.jpg 2-31.jpg 2-38.jpg	2-16.jpg 2-33.jpg 2-41.jpg 2-5.jpg 2-19.jpg 2-17.jpg 2-35.jpg 2-37.jpg 2-30.jpg 2-21.jpg 2-47.jpg
	Data uji	1-29.jpg 1-17.jpg	1-48.jpg	1-32.jpg	1-50.jpg	2-13.jpg 2-43.jpg	2-1.jpg	2-23.jpg	2-3.jpg