

16.215/4/02

PERANCANGAN DAN PEMBUATAN
PERANGKAT LUNAK
SEGMENTASI CITRA MULTI TEKSTUR
DENGAN MENGGUNAKAN
JARINGAN SARAF TIRUAN
TUGAS AKHIR



FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



RSIF
005.1
Kus
P-1
2001

Oleh :

LESTARI KUSUMAHATI

NRP. 2693 100 040

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2001

PERPUSTAKAAN ITS	
Tgl. Terima	00/01/02
Terima Dari	14
No. Agenda Prp.	21.4425

**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK
SEGMENTASI CITRA MULTI TEKSTUR
DENGAN MENGGUNAKAN JARINGAN
SARAF TIRUAN**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya**

Mengetahui / Menyetujui,

Dosen Pembimbing I



Ir. Esther Hanaya, M.Sc.
NIP. 130 816 212

Dosen Pembimbing II



Rully Soelaiman, S.Kom.
NIP. 132 085 802

**SURABAYA
Agustus, 2001**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

اللَّهُمَّ مَا أَصْبَحَ بِي مِنْ نِعْمَةٍ أَوْ بِأَحَدٍ

مِنْ خَلْقِكَ فَمَنْكَ وَحْدَكَ لَا شَرِيكَ لَكَ،

فَلَكَ الْحَمْدُ، وَ لَكَ الشُّكْرُ

*“Ya Allah, ni'mat apapun yang Engkau berikan kepadaku
atau kepada setiap dari makhluk ciptaan-Mu,
tiada lain hanya dari-Mu saja, tidak ada sekutu bagi-Mu.*

Maka bagi-Mu segala puji dan rasa syukur.”

*[HR. Abu Dawud, An Nasa'i dan Ibnu Hibban
dari Abdullah bin Ghanam al Bayadli ra.]*

*Untuk Ibunda dan Ayahanda tercinta
yang mencurahkan seluruh kasih sayang dan tenaga
yang selalu membimbing dan mengarahkan
yang menempati tempat khusus di hati Ananda
serta mengiringi hidup Nanda dengan do'a dan air mata
karya sederhana ini kupersembahkan*



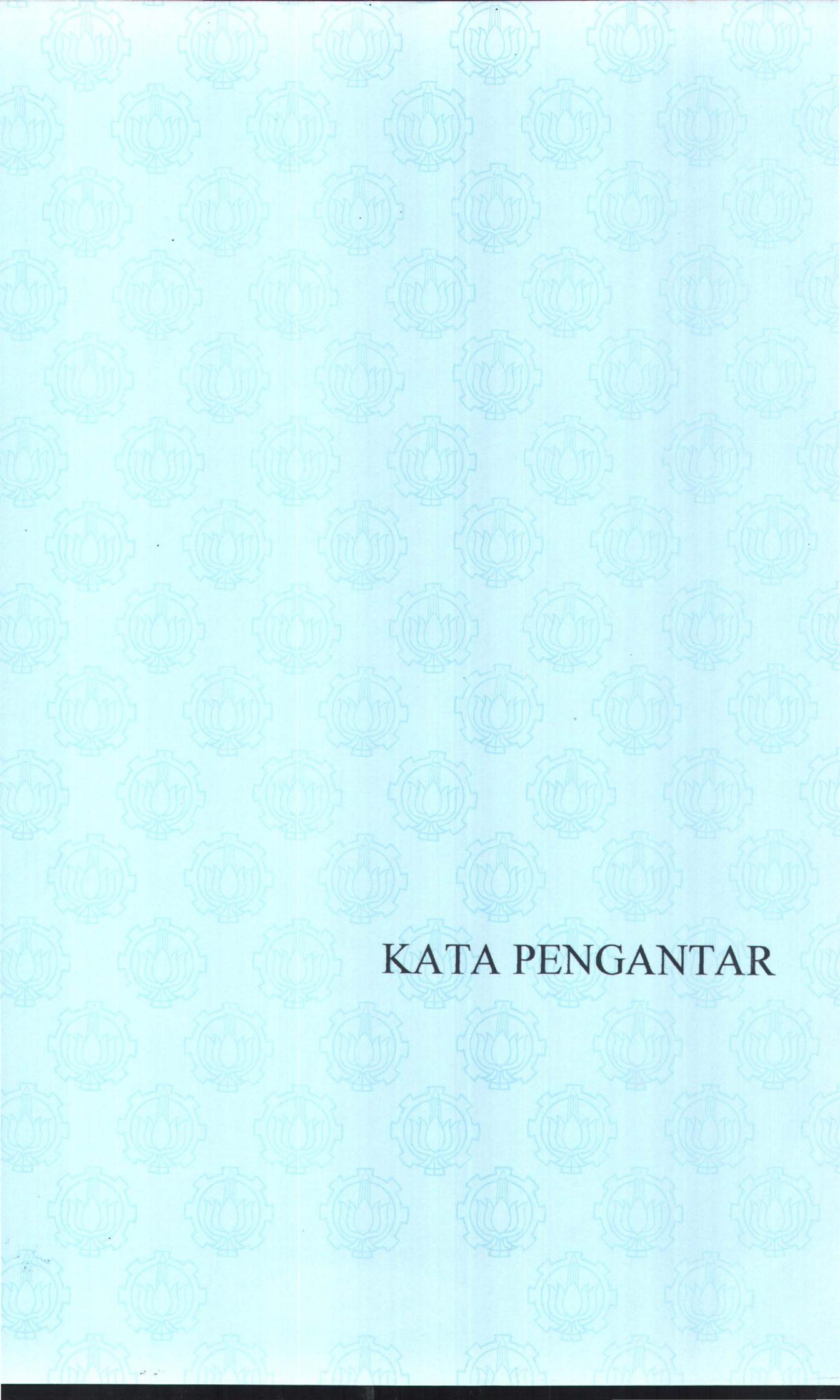
ABSTRAK

ABSTRAK

Segmentasi citra bertujuan untuk memisahkan komponen-komponen dari suatu citra berdasarkan kriteria tertentu, seperti intensitas atau tingkat keabuan, warna, tekstur dan lain-lain. Hasil segmentasi akan digunakan dalam proses pengolahan citra lebih lanjut. Segmentasi memegang peranan yang penting, karena proses ini akan menentukan keberhasilan atau kegagalan dari sebuah analisis. Penggunaan filter pada sebuah citra merupakan mekanisme yang umum digunakan untuk mengurangi bagian yang tidak diperlukan dan memperkuat bagian yang diperlukan dari spektrum. Pada aplikasi segmentasi citra digunakan metoda yang memanfaatkan filter, baik dengan filter tunggal (*single filter*) maupun dengan banyak filter (*multi filter*). Metoda ini mampu meminimalkan kesalahan dalam segmentasi, tetapi lambat dalam proses, hal ini bisa dipahami karena citra masukan diuji oleh setiap filter yang tersedia satu per satu, padahal tidak semua filter dibutuhkan (hanya filter yang sesuai dengan citra masukan).

Jaringan Saraf memiliki kemampuan untuk mempelajari karakteristik tekstur, mengenali tekstur berdasarkan karakteristiknya, serta mengelompokkan citra masukan berdasarkan teksturnya. Jaringan Saraf minimal terdiri dari dua *layer*, yaitu *layer* masukan (*input layer*) dan *layer* keluaran (*output layer*). Di antara *layer* masukan dan *layer* keluaran bisa terdapat satu atau beberapa *layer* tambahan yang disebut *layer* tersembunyi (*hidden layer*). Setiap *layer* berisi beberapa node. Setiap node mempunyai fungsi aktivasi dan hubungan dengan semua node pada *layer* di atasnya. Pada masing-masing hubungan mempunyai nilai bobot yang berbeda.

Dalam Tugas Akhir ini digunakan metoda Jaringan Saraf dengan *backpropagation* untuk memecahkan permasalahan segmentasi. Jaringan Saraf dilatih untuk mendapatkan serangkaian nilai bobot yang berfungsi sebagaimana filter pada metoda *single filter* atau *multi filter*. Nilai bobot ini digunakan untuk mengenali karakteristik citra masukan. Citra masukan dikelompokkan berdasarkan keluaran yang mempunyai nilai aktivasi tertinggi. Dengan demikian citra masukan diuji secara serentak dalam waktu yang bersamaan. Hasil yang memuaskan akan diperoleh apabila nilai bobot yang dihasilkan proses pelatihan memiliki ketelitian dalam menangani kesalahan pengelompokan.



KATA PENGANTAR

KATA PENGANTAR

AlhamdulillahLaahi Robbil 'Aalamiin. Tiada terputus-putusnya rasa syukur yang penulis panjatkan kepada Allah SWT, Pemilik segala kekuatan dan keindahan, yang telah memberikan penghidupan, nikmat iman, serta kemampuan bagi penulis untuk menyelesaikan tugas akhir ini.

Shalawat dan salam semoga senantiasa tercurah kepada Rasulullah Muhammad SAW, imam para rasul dan penutup para nabi, yang telah menyampaikan petunjuk dan agama yang benar bagi umat manusia, serta kepada siapa saja yang mencurahkan hidupnya untuk mengikuti jejak langkah beliau dalam menegakkan dienul Islam yang mulia ini hingga akhir jaman.

Atas selesainya tugas akhir dengan judul :

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

SEGMENTASI CITRA MULTI TEKSTUR

DENGAN MENGGUNAKAN JARINGAN SARAF TIRUAN

penulis ingin menyampaikan rasa terima kasih dan penghargaan yang sedalam-dalamnya kepada :

- ❖ Bapak Dr. Ir. Arif Djunaidy, MSc. selaku Ketua Jurusan Teknik Informatika ITS sekaligus sebagai Dosen Wali, yang menyertai penulis sejak awal perkuliahan.
- ❖ Ibu Ir. Esther Hanaya, MSc. selaku Dosen Pembimbing I yang dengan sabar membimbing, membantu dan mengiringi penyelesaian tugas akhir ini.
- ❖ Bapak Rully Soelaiman, S. Kom. selaku Dosen Pembimbing II yang memberikan ide-ide, saran, bimbingan dan bahkan motivasi bagi penulis.

- ❖ Segenap Dosen Jurusan Teknik Informatika ITS yang telah memberikan ilmunya bagi penulis.
- ❖ Staf Tata Usaha dan Karyawan Jurusan Teknik Informatika ITS yang telah membantu kelancaran studi penulis selama ini.
- ❖ Ibunda dan Ayahanda yang telah memberikan *andil* terbesar dalam hidup penulis. Yang membuat penulis tetap bertahan untuk menyelesaikan amanah ini. Semoga penulis bisa mewujudkan harapan menjadi anak yang shalihah. Dan semoga ini menjadi salah satu perwujudan bakti penulis kepada Ibunda dan Ayahanda tercinta.
- ❖ Dik Sara tersayang. Semoga Engkau menjadi generasi masa depan harapan umat.
- ❖ Mbak Niswah yang pertama kali mengenalkan penulis bahwa ada tujuan yang haqiqi dalam hidup ini.
- ❖ Mbak Uswah atas segala perdebatan yang justru membuat kita dekat. Yang tidak pernah menyerah untuk *memperjuangkan* penulis. Sungguh merupakan anugerah saat-saat bersama ukhti. Semoga Allah mengganti semua kebaikan ukhti dengan yang lebih baik.
- ❖ Mbak Nurul, teman, shahabat dan saudara yang membuat penulis merasa *berarti*. Semoga Allah selalu melindungi dan memberikan yang terbaik, serta menguatkan setiap langkah ukhti.
- ❖ Mbak Rina atas *jeweran-jewerannya* yang *mbikin kangen* dan atas perhatian serta kasih sayangnya selama ini. Maaf kalau penulis terlambat menyadarinya. Semoga ukhti selalu menjadi salah seorang yang dicintai Allah karena kecintaan ukhti kepada orang lain.

- ❖ Mbak Aida yang memberi semangat baru di akhir pembuatan Tugas Akhir ini. Semoga Allah selalu memberi kekuatan dan kesehatan, serta semoga penulis bisa menjadi adik dan shahabat yang baik bagi ukhti.
- ❖ Fakhrah, yang tanpa disadari telah menjadi shahabat dan saudara terbaik bagi penulis. Semoga kita dapat selalu *bersama* dalam keridhoan-Nya.
- ❖ Dik Zaki, atas *tsaqofah* bahasa Arab-nya, yang sudah menjadi shahabat dan saudara bagi penulis. Semoga Allah mempermudah segala sesuatunya bagi ukhti.
- ❖ Dik Yasmin, atas bantuannya yang sungguh berarti dalam menyelesaikan amanah ini. Atas perhatian dan kepercayaannya. Semoga Allah memberi kelancaran dalam menyelesaikan segala amanah ukhti.
- ❖ Mbak Fauziah dan Dik Shobrin, yang tanpa kalian sadari sudah memberi semangat dan *cambukan* bagi penulis. Semoga kita semua senantiasa istiqomah di jalan-Nya.
- ❖ Dik Asma', Dik Nada dan Dik Izza, atas bantuan komputer dan info-nya.
- ❖ Dik Daniyah, Dik Yuni dan Dik Afiyah, atas segala perhatian, bantuan dan do'anya bagi penulis.
- ❖ Dik Afidah, Dik Syahidah dan Fadhilah, atas kekompakkan kita yang secara tidak langsung amat membantu penulis.
- ❖ Mbak Ni'mah tersayang atas waktu dan diskusinya.
- ❖ Mbak Syarifah atas bantuannya dan Dik Jihad atas pinjaman bukunya.
- ❖ Dik Kamil, Dik Ulfa dan Arif atas komputernya.
- ❖ Ela, Lailil, Maully dan Ana yang telah menjadi shahabat dan menyertai penulis selama masa perkuliahan.

- ❖ Teman-teman angkatan '93 dan teman-teman *seperjuangan* dalam masa kuliah.
- ❖ Anib dan Andi Sucipto atas bantuan waktu dan pikirannya sehingga penulis bisa menyelesaikan tugas akhir ini.
- ❖ Adik-Adik Akhwat Jurusan Teknik Informatika ITS.
- ❖ Semua pihak yang tidak bisa penulis sebutkan satu-persatu, yang telah banyak membantu penulis selama ini.

Penulis menyadari akan kekurangan dan keterbatasan diri serta pada pembuatan tugas akhir ini. Namun, semoga karya sederhana ini dapat bermanfaat bagi kita semua.

Surabaya, Agustus 2001

Lestari Kusumahati



DAFTAR ISI

DAFTAR ISI

Halaman

LEMBAR JUDUL	
LEMBAR PENGESAHAN	
LEMBAR DEDIKASI	
ABSTRAK	
KATA PENGANTAR	i
DAFTAR ISI	v
DAFTAR GAMBAR	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan dan Manfaat	2
1.3 Perumusan Masalah	2
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	4
BAB II PENGOLAHAN CITRA DIGITAL	6
2.1 Model Citra Sederhana	6
2.2 Model Hubungan Antar Piksel	8
2.2.1 Tetangga (<i>neighbors</i>) Suatu Piksel	8
2.2.2 Hubungan Antar Piksel (<i>connectivity</i>)	9
2.3 Metoda Pengambilan Data	11
2.3.1 Domain Spasial	11
2.3.2 Domain Frekuensi	13

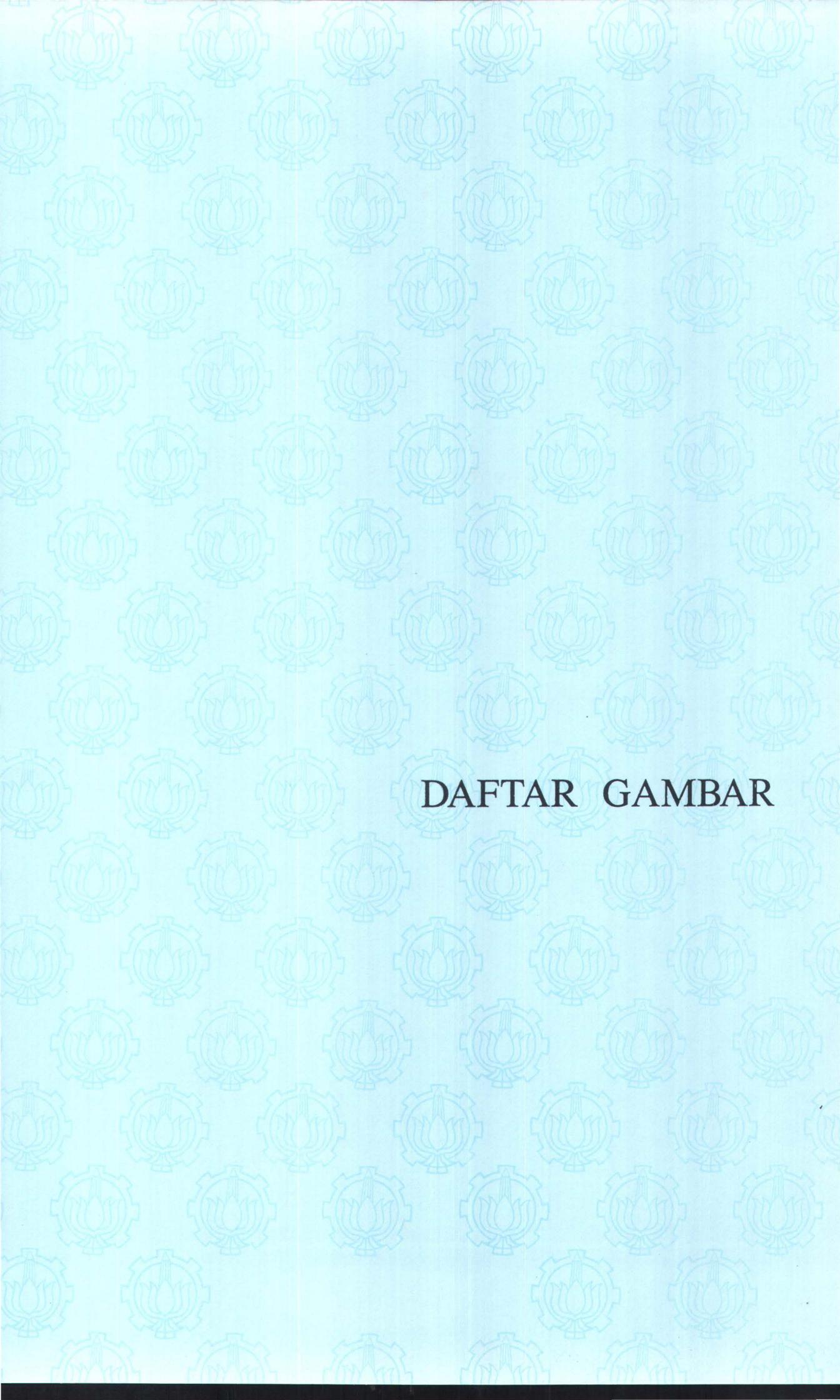
2.4	Segmentasi Citra.....	14
2.5	Tekstur.....	15
2.5.1	Analisis Tekstur.....	16
2.5.2	Metoda Penggambaran Tekstur.....	17
2.5.2.1	Metoda statistik.....	18
2.5.2.2	Metoda struktural.....	18
2.5.2.3	Metoda spektral.....	19
2.6	Teori Dasar Statistika.....	20
2.6.1	Distribusi Normal.....	20
2.6.2	Distribusi Normal Baku.....	21
BAB III JARINGAN SARAF TIRUAN.....		23
3.1	Neuron.....	23
3.3.1	Neuron pada Manusia.....	24
3.3.2	Neuron pada Jaringan Saraf Tiruan.....	25
3.2	Arsitektur Jaringan Saraf.....	26
3.3	Jaringan Saraf <i>Backpropagation</i>	30
3.3.1	Fungsi Aktivasi.....	30
3.3.2	Nilai Bobot.....	32
3.3.3	Metoda Belajar.....	33
3.3.4	Algoritma <i>Backpropagation</i>	34
BAB IV PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK.....		37
4.1	Deskripsi Perangkat Lunak.....	37
4.1.1	Masukan.....	37
4.1.2	Proses dalam Jaringan Saraf.....	38
4.1.3	Pengklasifikasian.....	38

4.1.4	Penampilan Hasil	38
4.2	Perancangan Data	39
4.2.1	Data Masukan	39
4.2.2	Data Saat Pemrosesan	39
4.2.3	Data Keluaran.....	39
4.3	Perancangan Proses.....	40
4.3.1	Diagram Alir Data (<i>Data Flow Diagram</i>)	40
4.3.2	Hirarki Modul	44
4.4	Implementasi Perangkat Lunak	45
4.4.1	Implementasi Struktur Data.....	45
4.4.2	Implementasi Proses.....	47
4.4.2.1	Proses <i>training</i> (pelatihan).....	47
4.4.2.2	Proses segmentasi (<i>testing</i>)	49
BAB V	HASIL UJI COBA DAN EVALUASI PERANGKAT LUNAK	51
5.1	Proses Pelatihan	52
5.2	Proses Segmentasi	54
BAB VI	PENUTUP	65
6.1	Kesimpulan	65
6.2	Saran	66

DAFTAR PUSTAKA

LAMPIRAN

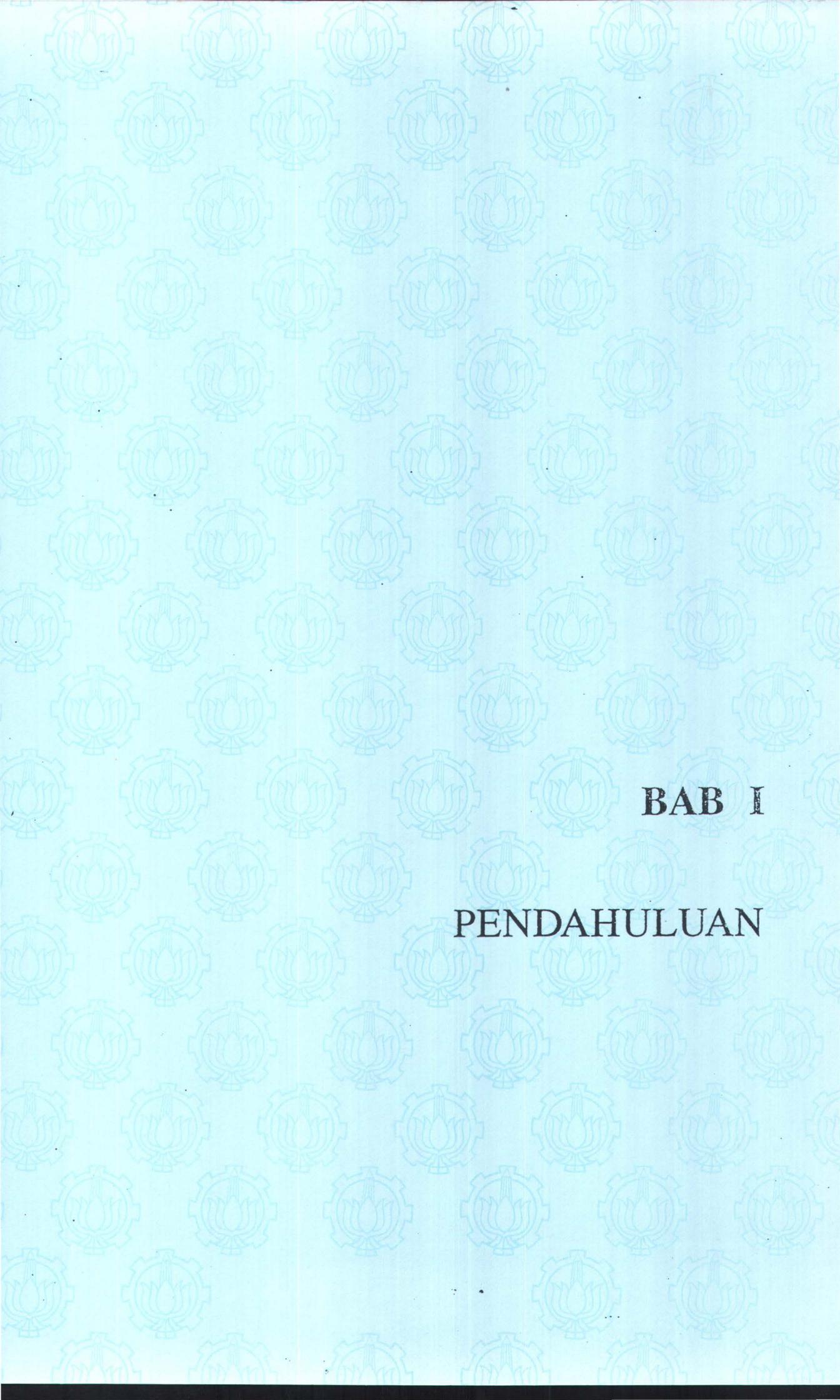
- A. Petunjuk Pemakaian (*User Manual*)
- B. Nilai Kesalahan dari Metoda Pelatihan dengan 2 *Layer*
- C. Nilai Kesalahan dari Metoda Pelatihan dengan 3 *Layer*
- D. Perhitungan Kesalahan untuk Proses Segmentasi



DAFTAR GAMBAR

DAFTAR GAMBAR

	Halaman
Gambar 2.1	Keterhubungan antar piksel 10
Gambar 2.2	Sebuah 3x3 tetangga pada titik (x,y) dalam sebuah citra 12
Gambar 2.3	Operasi dari sistem linier 14
Gambar 2.4	Tekstur dengan menggunakan metoda struktural 19
Gambar 2.5	Kurva normal 21
Gambar 2.6	Transformasi distribusi normal ke distribusi normal baku 22
Gambar 3.1	Neuron pada manusia 24
Gambar 3.2	Neuron pada jaringan saraf tiruan 25
Gambar 3.3	Arsitektur jaringan saraf dengan satu <i>layer</i> tersembunyi 27
Gambar 3.4	Skema <i>multichannel filter</i> 28
Gambar 3.5	Jaringan saraf untuk klasifikasi tekstur 29
Gambar 4.1	Diagram alir data level 0 41
Gambar 4.2	Diagram alir data level 1, detil dari level 0 41
Gambar 4.3	Diagram alir data level 2, detil dari pengambilan nilai bobot hasil <i>training</i> 42
Gambar 4.4	Diagram alir data level 2, detil dari proses <i>feed forward</i> 43
Gambar 4.5	Diagram alir data level 2, detil dari proses klasifikasi tekstur 43
Gambar 4.6	Hirarki modul dari segmentasi tekstur 44
Gambar 5.1	Tekstur-tekstur untuk proses pelatihan 52
Gambar 5.2	Grafik konvergenitas dari metoda pelatihan dengan 2 <i>layer</i> 53
Gambar 5.3	Grafik konvergenitas dari metoda pelatihan dengan 3 <i>layer</i> 54
Gambar 5.4	Citra masukan untuk uji coba pertama 55
Gambar 5.5	Hasil uji coba 1 56
Gambar 5.6	Hasil uji coba 1 dengan menggunakan filter median 57
Gambar 5.7	Citra masukan untuk uji coba kedua 58
Gambar 5.8	Hasil uji coba 2 59
Gambar 5.9	Hasil uji coba 2 dengan menggunakan filter median 60
Gambar 5.10	Citra masukan untuk uji coba ketiga 61
Gambar 5.11	Hasil uji coba 3 62
Gambar 5.12	Hasil uji coba 3 dengan menggunakan filter median 63



BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

Dalam bab ini dibahas mengenai latar belakang, tujuan dan manfaat, permasalahan, batasan masalah, sistematika pembahasan, serta metodologi yang digunakan untuk membuat Tugas Akhir ini.

1.1 LATAR BELAKANG

Langkah awal dalam menganalisis suatu citra adalah proses segmentasi dari citra yang dimaksud. Segmentasi merupakan proses pengelompokan citra berdasarkan bagian-bagian (objek) tertentu. Objek-objek hasil segmentasi tadi digunakan lebih lanjut dalam pengolahan citra. Segmentasi ini memegang peranan yang penting dalam pemrosesan gambar, karena proses ini akan menentukan keberhasilan atau kegagalan dari sebuah analisis.

Penggunaan filter pada sebuah citra merupakan mekanisme yang umum digunakan untuk mengurangi bagian yang tidak diperlukan dan memperkuat bagian yang diperlukan dari spektrum. Pada aplikasi segmentasi citra digunakan metoda yang memanfaatkan filter, baik dengan filter tunggal (*single filter*) maupun dengan banyak filter (*multi filter*).

Metoda ini mampu meminimalkan kesalahan dalam segmentasi, tetapi lambat dalam proses, hal ini bisa dipahami karena citra masukan diuji oleh setiap filter yang tersedia satu per satu, padahal tidak semua filter dibutuhkan (hanya filter yang sesuai dengan citra masukan). Bahkan bisa jadi sebuah citra hanya

terdiri dari dua atau tiga tekstur saja. Tetapi citra ini tetap harus difilter dengan semua filter yang ada satu per satu sebelum bisa disegmentasikan.

Salah satu metoda segmentasi adalah dengan menggunakan jaringan saraf. Jaringan saraf memiliki kemampuan untuk mempelajari karakteristik tekstur, mengenali tekstur berdasarkan karakteristiknya, serta mengelompokkan citra masukan berdasarkan teksturnya.

1.2 TUJUAN DAN MANFAAT

Tujuan dari Tugas Akhir ini adalah merancang dan membuat suatu perangkat lunak untuk mengelompokkan citra berdasarkan tekstur dengan menggunakan jaringan saraf. Adapun manfaat dari Tugas Akhir ini memudahkan dalam analisis citra yang kemudian digunakan untuk proses lebih lanjut, seperti interpretasi dan pengenalan pola.

1.3 PERUMUSAN MASALAH

Beberapa permasalahan yang harus dipecahkan dalam Tugas Akhir ini adalah sebagai berikut :

- ◆ Bagaimana memetakan proses penyusunan arsitektur jaringan saraf dengan menggunakan prinsip filter *multichannel* yang sesuai dengan tekstur yang dipergunakan.
- ◆ Bagaimana mendapatkan serangkaian nilai bobot yang berfungsi sebagai filter set, sehingga akhirnya didapatkan hasil citra yang telah disegmentasi dengan tingkat ketepatan yang optimal.

1.4 BATASAN MASALAH

Permasalahan yang berhubungan dengan bidang segmentasi tekstur menggunakan jaringan saraf sangat rumit dan kompleks, serta membutuhkan waktu yang tidak sedikit, sehingga dalam tugas akhir ini diberikan beberapa batasan masalah, yaitu :

- ◆ Citra yang digunakan berupa file bitmap (.bmp) berukuran 128 x 128 piksel.
- ◆ Tipe tekstur dari citra masukan maksimum lima macam.
- ◆ Citra tekstur yang akan diproses sifatnya periodik (berulang) dan terstruktur.
- ◆ Menggunakan fasilitas filter median yang terdapat dalam Adobe Photoshop 5.5.

1.5 METODOLOGI

Langkah-langkah yang dilakukan untuk pembuatan tugas akhir ini dibagi menjadi beberapa tahapan sebagai berikut :

- ◆ Studi Literatur

Pada tahap ini dilakukan studi literatur terhadap teori pengolahan citra digital, khususnya pada materi segmentasi citra, tekstur, statistik citra dan studi literatur mengenai jaringan saraf.

- ◆ Menetapkan Definisi Kebutuhan

Pada tahap ini dilakukan pendefinisian terhadap kemampuan perangkat lunak yang akan dirancang dan batasan-batasannya.

- ◆ Perancangan Perangkat Lunak

Pada tahap ini dilakukan perancangan struktur data, algoritma dan diagram alur yang akan digunakan dalam program.

- ◆ Pembuatan Perangkat Lunak

Pada tahap ini dilakukan pengimplementasian struktur data dan algoritma yang telah dirancang, ke dalam bahasa pemrograman.

- ◆ Evaluasi dan Revisi

Pada tahap ini dilakukan pengevaluasian dan perbaikan program yang telah dibuat, sampai didapatkan hasil yang sesuai atau tidak menyimpang dari definisi kebutuhan.

- ◆ Penyusunan Naskah Tugas Akhir

Pada tahap ini dilakukan penulisan naskah untuk menjelaskan dasar teori yang dipergunakan serta penyusunan laporan.

1.6 SISTEMATIKA PEMBAHASAN

Pembahasan mengenai perangkat lunak yang disusun dalam tugas akhir ini dibagi dalam beberapa bab, yang dijelaskan di bawah sebagai berikut :

 **Bab I** berisi pendahuluan yang memuat latar belakang pembuatan perangkat lunak, tujuan dan manfaat, perumusan masalah, batasan masalah, langkah-langkah pembuatan tugas akhir ini dan sistematika penulisan buku tugas akhir.

 **Bab II** berisi landasan teori yang berkaitan dengan dasar-dasar pengolahan citra digital dan beberapa teori statistika yang mendukung permasalahan secara umum.

BAB II

PENGOLAHAN CITRA DIGITAL

Pada bab ini dijelaskan mengenai beberapa konsep dari pengolahan citra digital yang berkaitan dengan tugas akhir. Selain itu juga diberikan penjelasan mengenai teori-teori dasar statistik sebagai penunjang tugas akhir ini.

2.1 MODEL CITRA SEDERHANA

Citra monokrom atau citra sederhana, dirumuskan sebagai suatu fungsi intensitas cahaya dua dimensi $f(x,y)$, dimana x dan y menyatakan koordinat spasial (koordinat ruang 2 dimensi) dan nilai dari f pada setiap titik (x,y) menyatakan tingkat kecerahan (*gray level*) dari citra pada titik tersebut.¹

Agar dapat diproses dengan komputer, fungsi $f(x,y)$ harus didiskritkan baik koordinat spasialnya maupun tingkat kecerahannya. Sehingga dapat digambarkan citra digital sebagai matriks yang posisi baris dan kolomnya (nilai x dan y), menunjukkan pada sebuah titik dalam suatu citra dan nilai elemen matriksnya menunjukkan tingkat kecerahan dari titik yang bersangkutan.

¹ Rafael C. Gonzales, Richard E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company Inc., 1993, halaman 6

Proses pengolahan citra menjadi bentuk digital dari koordinat spasial (x,y) disebut *image sampling*, sedangkan pengolahan citra ke dalam tingkat kecerahannya atau amplitudo disebut sebagai *gray-level quantization*.

Penggambaran citra kontinyu dengan fungsi $f(x,y)$ sebagai matrik dengan ukuran $N \times M$ seperti ditunjukkan pada persamaan (2.1-1), di mana setiap elemen dari matrik mempunyai bentuk nilai diskrit.

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \dots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (2.1-1)$$

Sisi kanan dari persamaan di atas disebut sebagai *digital image*, sedangkan setiap elemen dari array disebut sebagai piksel (*image element*).

Pada proses pengolahan citra digital, diperlukan informasi tentang besarnya ukuran suatu citra serta nilai maksimum dari *gray level*. Hal ini dapat ditunjukkan dalam bentuk :

$$N = 2^n \text{ dan } G = 2^m$$

Dimana N adalah ukuran dimensi citra yang berbentuk bujursangkar ($N \times N$). Ukuran ini yang sering disebut sebagai resolusi piksel. Sedangkan G menunjukkan nilai maksimum *gray level* dan m adalah panjang bit yang digunakan untuk menyimpan nilai intensitas setiap piksel. Besar kecilnya nilai *gray level* yang digunakan, akan menentukan kualitas gambar yang dihasilkan.

2.2 MODEL HUBUNGAN ANTAR PIKSEL²

Model hubungan antar piksel pada pengolahan citra digital merupakan salah satu hal yang sangat penting. Seperti yang sudah dijelaskan sebelumnya, sebuah gambar akan dinotasikan dengan $f(x,y)$. Untuk piksel tertentu dinotasikan dengan huruf kecil, seperti p dan q , sedangkan subset dari himpunan piksel akan dinotasikan dengan huruf besar, seperti S dan V .

2.2.1 Tetangga (*neighbors*) Suatu Piksel

Sebuah piksel p dengan koordinat (x,y) mempunyai 4 tetangga pada arah horisontal dan vertikal dari piksel yang bersangkutan, dengan koordinat-koordinat sebagai berikut :

$$(x+1,y), (x-1,y), (x,y+1), (x,y-1)$$

himpunan dari piksel-piksel di atas disebut 4-tetangga dari p , yang dinotasikan dengan $N_4(p)$.

Selain itu, p juga mempunyai 4 diagonal tetangga yang mempunyai koordinat :

$$(x+1,y+1), (x+1,y-1), (x-1,y+1), (x-1,y-1)$$

yang dinotasikan dengan $N_D(p)$. Gabungan dari piksel 4 tetangga dan diagonal tetangga disebut dengan 8 tetangga dari p , dinotasikan dengan $N_8(p)$.

Beberapa piksel dari $N_4(p)$, $N_D(p)$ dan $N_8(p)$ akan berada di luar gambar jika $p(x,y)$ berada pada batas dari gambar.

² ibid, halaman 40 – 42

2.2.2 Hubungan Antar Piksel (*connectivity*)

Hubungan antar piksel adalah hal yang penting. Karena hal ini digunakan untuk menentukan batas dari objek-objek atau komponen-komponen pada gambar. Untuk menentukan apakah dua piksel saling berhubungan atau tidak, harus dilihat apakah keduanya mempunyai hubungan yang sesuai (misalnya : keduanya berhubungan 4-tetangga) dan jika *gray-level* dari piksel-piksel itu memenuhi kriteria tertentu yang sama. Jadi ada kemungkinan dalam suatu gambar nanti ada piksel yang berdekatan dengan 4 tetangga, tetapi mereka dikatakan tidak terhubung, karena tidak mempunyai nilai *gray level* yang sama.

Misalkan V adalah himpunan dari nilai *gray level* yang digunakan untuk menentukan keterhubungan antar piksel, dan hanya piksel-piksel dengan nilai intensitas 7, 8, 9 dan 10 yang diperhatikan, maka $V = \{7,8,9,10\}$.

Ada tiga jenis keterhubungan, yaitu :

(a). *4-keterhubungan*.

Dua piksel p dan q dengan nilai intensitas ada di V adalah termasuk jenis keterhubungan ini jika q anggota dari $N_4(p)$.

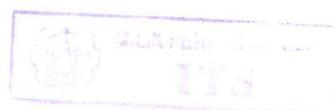
(b). *8-keterhubungan*.

Dua piksel p dan q dengan nilai intensitas ada di V adalah termasuk jenis keterhubungan ini jika q anggota dari $N_8(p)$.

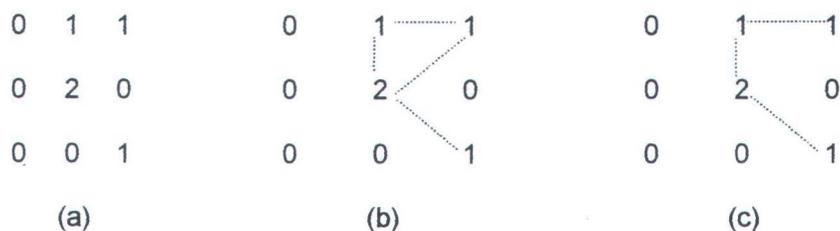
(c). *c-keterhubungan (campuran)*.

Dua piksel p dan q dengan nilai intensitas ada di V adalah termasuk jenis keterhubungan ini jika,

- (i) q ada di $N_4(p)$ atau
- (ii) q ada di $N_D(p)$ dan hasil dari $N_4(p) \cap N_4(q)$ adalah himpunan kosong.



Keterhubungan campuran adalah hasil modifikasi dari 8-keterhubungan dan ini diperkenalkan untuk *multiple path connection* yang sering terjadi jika menggunakan 8-keterhubungan. Sebagai contoh, suatu piksel tersusun dengan bentuk seperti gambar 2.1(a). Diasumsikan $V=\{1,2\}$ dan jalur 8-tetangga dari piksel dengan nilai intensitas 2 adalah gambar 2.1(b), dimana terlihat kerancuan jalur yang dihasilkan dengan menggunakan 8-keterhubungan. Kerancuan ini dapat dihilangkan dengan menggunakan keterhubungan-campuran, seperti ditunjukkan pada gambar 2.1(c).



Gambar 2.1 Keterhubungan antar piksel

- a. Susunan piksel
- b. 8-keterhubungan dari piksel pusat
- c. Keterhubungan campuran dari piksel pusat

Piksel p disebut berdekatan atau bersebelahan (*adjacent*) dengan piksel q jika mereka saling terhubung. Dapat didefinisikan 4-, 8-, atau c-kedekatan dari piksel, tergantung pada jenis keterhubungan yang ditentukan. Dua sub himpunan S_1 dan S_2 dari suatu gambar adalah berdekatan, jika ada beberapa piksel pada S_1 yang berdekatan dengan beberapa piksel dari S_2 .

Sebuah jalur (*path*) dari piksel p dengan koordinat (x,y) ke piksel q dengan koordinat (s,t) adalah sebuah urutan dari piksel-piksel yang berbeda dengan koordinat :

$$(x_0, y_0), (x_1, y_1), \dots (x_n, y_n)$$

dimana $(x_0, y_0) = (x, y)$ dan $(x_n, y_n) = (s, t)$, (x_i, y_i) bersebelahan dengan (x_{i-1}, y_{i-1}) , $1 \leq i \leq n$ dan n adalah panjang dari jalur. Bisa didefinisikan 4-, 8-, atau c-jalur, tergantung dari jenis kedekatan yang dibutuhkan.

Jika p dan q adalah piksel pada gambar dengan sub himpunan S , maka p terhubung ke dalam S jika ada jalur dari p ke q yang jalurnya terdiri dari semua piksel yang ada di S . Untuk setiap piksel p di S , himpunan piksel di S yang terhubung ke p disebut komponen terhubung (*connected component*) dari S . Karena itu tiap dua piksel dari sebuah komponen terhubung adalah terhubung satu sama lainnya dan komponen terhubung yang berlainan adalah saling lepas (*disjoint*).

2.3 METODA PENGAMBILAN DATA³

Dalam sistem pengolahan citra digital, metoda pengambilan data yang akan diolah dapat dikategorikan dalam dua macam, yaitu metoda domain spasial dan domain frekuensi.

2.3.1 Domain Spasial

Domain spasial menunjuk pada sekumpulan piksel yang ditempatkan secara bersama dalam suatu citra dan metoda domain spasial merupakan suatu

³ ibid, halaman 162-165

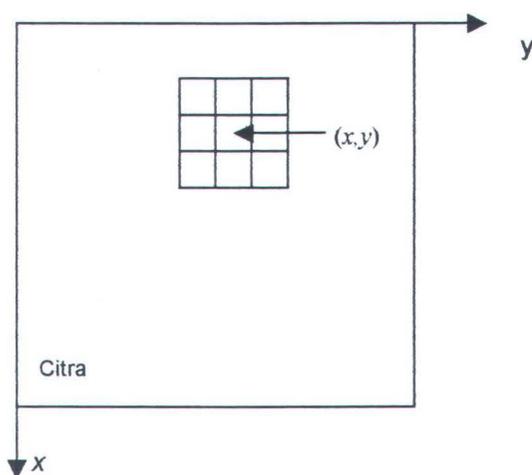
prosedur yang dioperasikan secara langsung pada piksel. Fungsi pengolahan citra pada domain spasial adalah sebagai berikut :

$$g(x,y) = T[f(x,y)] \quad (2.3.1-1)$$

dimana $f(x,y)$ adalah citra input, $g(x,y)$ adalah citra hasil olahan, serta T adalah operator pada fungsi f , yang didefinisikan melalui nilai tetangga dari (x,y) .

Pendekatan yang digunakan untuk mendefinisikan tetangga dari (x,y) dapat berbentuk daerah bujursangkar atau persegi panjang yang berpusat di koordinat (x,y) seperti yang telah dijelaskan pada bagian sebelumnya.

Daerah ini disebut dengan jendela. Jendela ini digerakkan sepanjang daerah citra mulai dari posisi kiri atas sampai dengan kanan bawah untuk menghasilkan nilai g pada tiap lokasi piksel, seperti diperlihatkan pada gambar 2.2 di bawah.



Gambar 2.2 Sebuah 3x3 tetangga pada titik (x,y) dalam sebuah citra

2.3.2 Domain Frekuensi

Landasan awal pada metoda domain frekuensi adalah teorema konvolusi. Misalkan $g(x,y)$ merupakan citra yang dibentuk dengan proses konvolusi dari sebuah citra $f(x,y)$ dan fungsi linier, yakni :

$$g(x,y) = h(x,y)*f(x,y) \quad (2.3.2-1)$$

Kemudian dari teorema konvolusi relasi dengan domain frekuensi menjadi :

$$G(u,v) = H(u,v)F(u,v) \quad (2.3.2-2)$$

dimana G , H dan F merupakan transformasi Fourier dari g , h dan f . Di dalam istilah teori sistem linier, transformasi $H(u,v)$ dinamakan dengan fungsi pindah (*transfer function*) dari proses.

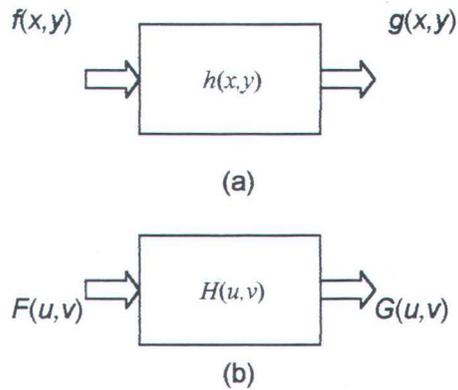
Berbagai permasalahan perbaikan citra dapat dinotasikan dalam bentuk persamaan (2.3.2-2). Di dalam penerapannya, $f(x,y)$ diberikan dan tujuan sesudah perhitungan dari $F(u,v)$ adalah memilih $H(u,v)$ sehingga citra yang diinginkan adalah :

$$g(x,y) = \mathcal{F}^{-1}[H(u,v)F(u,v)] \quad (2.3.2-3)$$

memperlihatkan beberapa karakteristik atau ciri pokok dari $f(x,y)$. Misalnya, tepi-tepi dalam $f(x,y)$ dapat ditonjolkan dengan menggunakan fungsi $H(u,v)$ yang menekankan bagian-bagian frekuensi tinggi dari $F(u,v)$.

Gambar 2.3 mengilustrasikan sebuah sistem yang berfungsi untuk menghasilkan citra keluaran $g(x,y)$ dari citra masukan $f(x,y)$. Sistem tersebut menjalankan konvolusi dari $h(x,y)$ dengan citra input dan menghasilkan citra keluaran. Sesuai dengan teorema konvolusi, melalui sudut pandang yang berbeda, hasil yang sama akan diperoleh dengan mengalikan $F(u,v)$ dan $H(u,v)$

untuk menghasilkan $G(u,v)$. Berikutnya mencari nilai invers transformasi Fourier dari keluaran tersebut akan menghasilkan citra yang diinginkan.



Gambar 2.3 Operasi dari sistem linier

- Keluaran sistem merupakan konvolusi $h(x,y)$ dengan masukannya.
- Keluaran merupakan perkalian dari $H(u,v)$ dengan masukannya.

2.4 SEGMENTASI CITRA⁴

Tujuan dari segmentasi permukaan sebuah gambar adalah memisahkan komponen-komponen dari sebuah citra yang berhubungan dengan objek fisik dalam gambar. Komponen yang telah dipisahkan kemudian digunakan untuk proses pada level yang lebih tinggi seperti interpretasi dan pengenalan. Kesulitan utama dalam segmentasi gambar adalah kesalahan dalam mendefinisikan fakta, bahwa biasanya kita tertarik untuk menemukan batas-batas atau boundari dari objek fisik dalam gambar tiga dimensi, akan tetapi kita hanya memiliki informasi gambar dua dimensi.

⁴ Tzay Y. Young, King-Sun Fu, *Handbook of Pattern Recognition and Image Processing*, Academic Press, Inc., 1986, halaman 215-216

Pada kebanyakan metoda, segmentasi gambar diasumsikan bahwa objek memiliki permukaan halus dan rata yang berhubungan dengan region secara konstan atau memiliki intensitas yang berubah secara tajam pada bagian boundari atau batas. Asumsi ini sesuai, akan tetapi tidak selalu benar, misalnya pada objek yang memiliki permukaan bertekstur dengan intensitas yang berbeda, juga keberadaan noise akan menyebabkan permasalahan tambahan.

2.5 TEKSTUR⁵

Dari beberapa karakteristik citra, isi tekstur merupakan salah satu karakteristik citra yang mempunyai bentuk yang khas. Karakteristik dari isi tekstur inilah yang digunakan untuk menggambarkan suatu daerah pada citra.

Meskipun secara formal tidak ada definisi khusus tentang tekstur, tetapi karakteristik tekstur dapat digambarkan suatu ukuran tentang sifat-sifat kehalusan (*smoothness*), kekasaran (*coarseness*) dan keteraturan (*regularity*).

Tekstur dalam hubungannya dengan pengolahan citra (*image processing*) adalah penggambaran penyebaran piksel secara spasial pada suatu bagian (*region*) dalam suatu citra.

Tekstur sangat berguna untuk beberapa aplikasi pengolahan citra. Perubahan dalam suatu tekstur dapat digunakan untuk segmentasi citra secara spasial. Ukuran parameter dalam suatu tekstur dapat digunakan untuk mengelompokkan area-area dalam gambar, seperti mengelompokkan daerah pertambangan, pertanian, jalan, ataupun rumah. Dan pada akhirnya tekstur

⁵ Rafael C. Gonzales, Richard E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company Inc., 1993, halaman 506

dapat dibuat untuk mengisi bagian-bagian dalam gambar untuk keperluan pengkodean citra.

2.5.1 Analisis Tekstur

Analisis tekstur merupakan sebuah proses untuk menemukan, mengenali dan memahami pola-pola yang berhubungan dengan *performance* dari proses citra. Salah satu tujuan yang mendasar dari analisis tekstur dengan menggunakan komputer adalah memberi bantuan kepada mesin, sehingga memiliki kemampuan yang serupa dengan kemampuan manusia dalam mengenali sesuatu. Sebagai contoh, sebuah sistem yang dirancang untuk membaca citra dari tipe dokumen, pola yang dijadikan objek adalah karakter baik yang berupa angka maupun huruf dan tujuan yang hendak dicapai adalah mengenali karakter dengan tepat, sebagaimana kemampuan yang ditunjukkan oleh manusia dalam menyelesaikan masalah tersebut.

Ada beberapa pembahasan utama dalam analisis tekstur yang diuraikan secara berurutan oleh Ehrich dan Foith, sebagai berikut :⁶

- a. Diberikan region bertekstur, dilakukan proses untuk menentukan nilai batas dari kelas yang dimiliki oleh region.
- b. Diberikan region bertekstur, dilakukan proses untuk menentukan penggambaran atau model dari tekstur.
- c. Diberikan citra yang memiliki banyak area bertekstur, dilakukan proses untuk menentukan batas antara region-region yang memiliki tekstur berlainan.

⁶ Tzay Y. Young, King-Sun Fu, *Handbook of Pattern Recognition and Image Pocessing*, Academic Press, Inc., 1986, halaman 248

Pembahasan (a) berhubungan dengan proses pengenalan pola dari ekstraksi karakteristik tekstur, pembahasan (b) berhubungan dengan model-model generatif dari tekstur, sedangkan pembahasan (c) merupakan penggabungan kedua bahasan sebelumnya dalam sebuah aturan yang digunakan untuk mengerjakan segmentasi tekstur dari sebuah citra.

2.5.2 Metoda Penggambaran Tekstur⁷

Ada tiga pendekatan dalam pengolahan citra untuk menggambarkan suatu tekstur, yaitu pendekatan statistik, struktural dan spektral. Pendekatan statistik menghasilkan karakteristik tekstur seperti kehalusan, kekasaran dan lain-lain. Pendekatan struktural, di lain pihak berhubungan dengan penyusunan dari pola-pola primitif yang disebut *texel (texture element)*, misalnya penggambaran tekstur berdasarkan pada keteraturan garis-garis paralel.

Sedangkan pendekatan secara spektral didasarkan pada sifat-sifat spektrum Fourier dan terutama digunakan untuk mendeteksi periode global dalam suatu gambar dengan mengidentifikasi (mengenali) energi tinggi atau puncak yang sempit dalam spektrum.

⁷ Rafael C. Gonzales, Richard E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company Inc., 1993, halaman 507-511



2.5.2.1 Metoda statistik

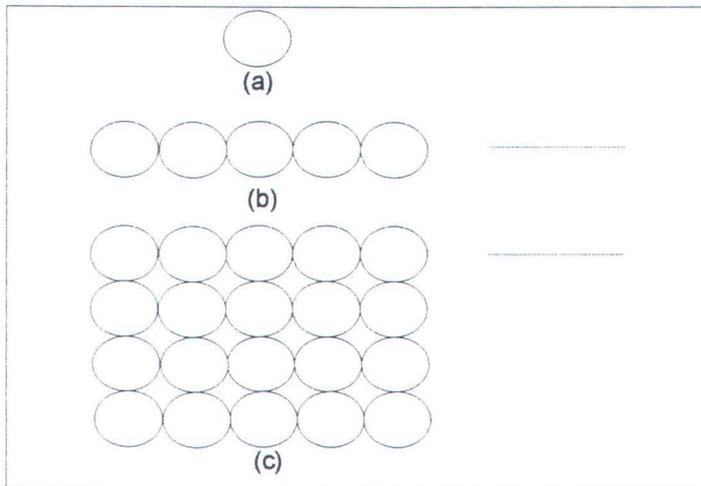
Pada metoda statistik, penggambaran suatu tekstur dapat dilihat dari beberapa karakteristik seperti kehalusan, kekasaran dan butiran-butiran. Metoda ini ditekankan pada tingkat *gray level* piksel dari suatu citra dengan menggunakan analisis tekstur untuk proses pendeteksian citra. Salah satu pendekatan sederhana untuk menggambarkan tekstur adalah dengan menggunakan momen-momen dari *histogram gray level* pada sebuah daerah atau citra.

2.5.2.2 Metoda struktural

Konsep pada metoda ini ditekankan pada karakteristik dari tekstur yang merupakan perulangan pola dari elemen-elemen yang terdapat pada daerah citra. Gambar 2.4 menggambarkan suatu bentuk perulangan pola elemen dari citra yang dinyatakan dalam beberapa aturan. Pada gambar dimisalkan kita mempunyai aturan $S \rightarrow aS$ yang berarti bentuk S dapat dinyatakan sebagai aS . Dan 'a' menyatakan sebuah perulangan dan diartikan sebagai perulangan ke samping kanan (*circles to the right*) ke dalam string 'aaa', dan dimisalkan bentuk aturan $S \rightarrow aS$ dapat diubah ke dalam pola-pola tekstur suatu citra. Maka dengan menambahkan beberapa aturan lagi akan didapatkan bentuk seperti dalam gambar 2.4(c).

Dari beberapa aturan sebelumnya, jika kita tambahkan beberapa aturan $S \rightarrow bS$, $A \rightarrow cA$, $A \rightarrow c$, $S \rightarrow bS$, $S \rightarrow a$, dimana 'b' menyatakan perulangan ke arah bawah (*circle down*), 'c', menyatakan perulangan ke arah sebelah kiri (*circle to the left*), maka dari semua aturan yang ada kita dapat mengubah ke dalam

bentuk yang dapat dinyatakan sebagai matriks perulangan berukuran 3×3 . Gambar 2.4(c) merupakan pola-pola tekstur yang dikembangkan dengan aturan seperti di atas.



Gambar 2.4 Tekstur dengan menggunakan metoda struktural

- a. Bentuk asli dari pola tekstur
- b. Bentuk tekstur yang dihasilkan dari aturan $S \rightarrow aS$
- c. Bentuk tekstur yang dihasilkan dari penggunaan semua aturan seperti yang dijelaskan di atas

2.5.2.3 Metoda spektral

Dalam metoda ini karakteristik tekstur didasarkan pada properti-properti dari spektrum Fourier, yang digunakan untuk mendeteksi bentuk atau pola periodik secara umum pada citra. Caranya yaitu dengan mengidentifikasi *high energy* dalam spektrum Fourier. Ada tiga *feature* dari spektrum Fourier yang

dapat digunakan untuk analisis tekstur yaitu, pertama, puncak-puncak spektrum yang terlihat menonjol yang merupakan penggambaran dari pola-pola tekstur, kedua, lokasi dari puncak-puncak spektrum yang merupakan titik ruang dari pola-pola tadi dan ketiga, dengan mengeliminasi semua komponen yang mempunyai pola secara periodik melalui suatu proses filtering, maka elemen-elemen pada citra yang bersifat non-periodik akan tetap ada dalam citra. Sisa-sisa elemen non-periodik yang masih ada dalam citra ini, dapat menggambarkan bentuk suatu tekstur dengan metoda statistik.

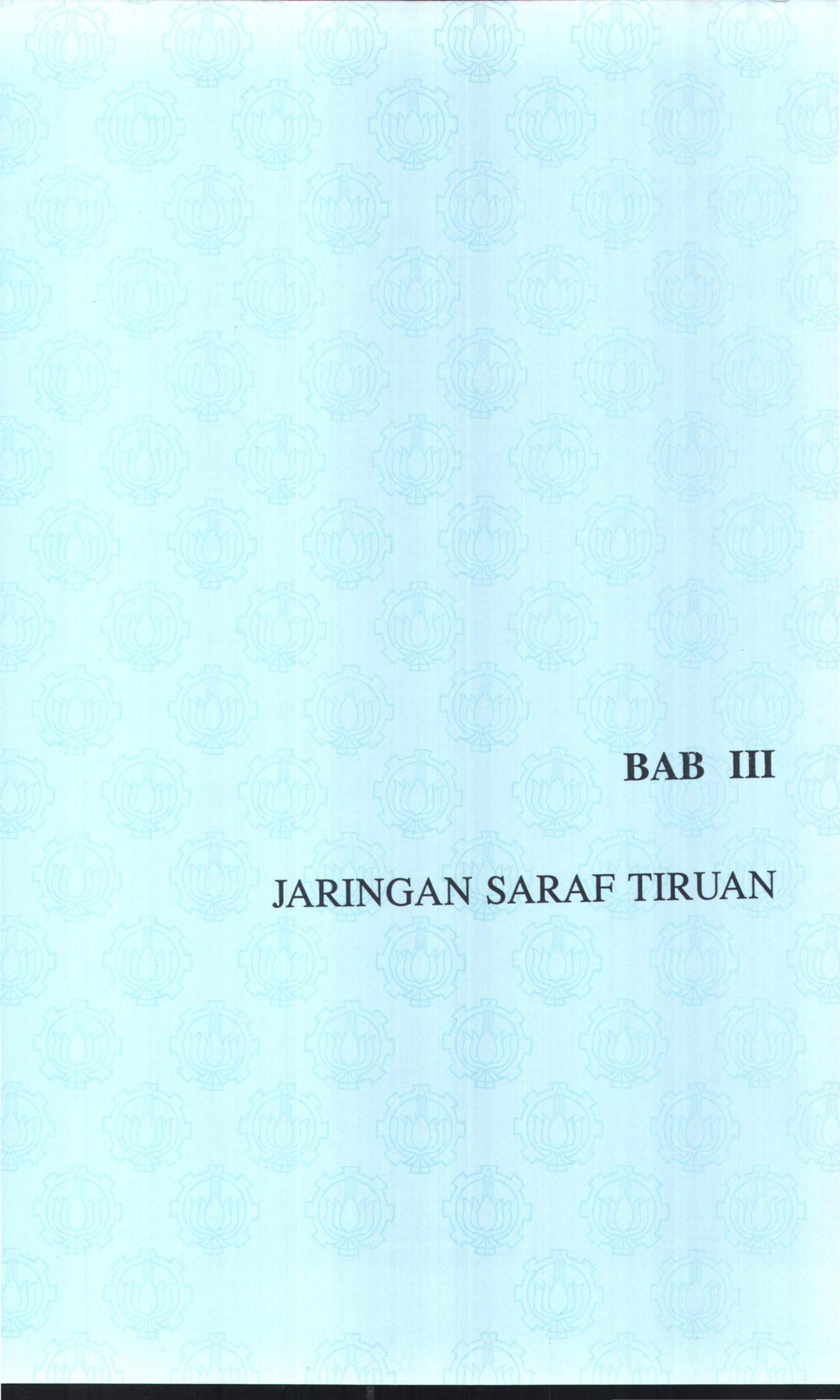
2.6 TEORI DASAR STATISTIKA⁸

Pada bagian berikut ini, akan dijelaskan sedikit mengenai teori-teori dasar statistika yang berhubungan dengan tugas akhir ini.

2.6.1 Distribusi Normal

Distribusi normal dapat menggambarkan dengan cukup baik banyak gejala yang muncul di alam, industri dan penelitian. Pada tahun 1733, Abraham de Moivre menemukan persamaan matematika kurva normal yang kemudian dijadikan dasar bagi banyak teori statistika yang lain. Grafik dari distribusi normal disebut dengan kurva normal, seperti pada gambar 2.5.

⁸ Ronald E. Walpole, Raymond H. Myers, *Ilmu Peluang dan Statistika untuk Insinyur dan Ilmuwan*, Edisi ke-4, diterjemahkan oleh Dr. R.K. Sembiring, Penerbit ITB, Bandung, 1995, halaman 230-238



BAB III

JARINGAN SARAF TIRUAN

BAB III

JARINGAN SARAF TIRUAN

Jaringan saraf tiruan merupakan salah satu cabang dari kecerdasan buatan. Cara kerjanya menirukan cara kerja dari jaringan saraf manusia. Tetapi tidak semua proses dan fungsi pada jaringan saraf manusia dapat diterapkan ke dalam jaringan saraf tiruan. Salah satu fungsi dari jaringan saraf tiruan yang diadopsi dari jaringan saraf manusia adalah kemampuannya untuk mempelajari hal-hal yang baru dikenal, kemudian mengingat hal-hal yang baru dikenal dan dipelajari tadi, serta memberikan respon terhadap suatu rangsangan. Sifat-sifat neuron serta hubungan antar neuron menjadi dasar dari pembentukan jaringan saraf tiruan.

Pada bab ini akan dijelaskan tentang bagaimana cara kerja neuron yang merupakan simulasi dari neuron pada tubuh manusia, serta jaringan saraf *backpropagation* yang menjadi algoritma inti pada tugas akhir ini.

3.1 NEURON⁹

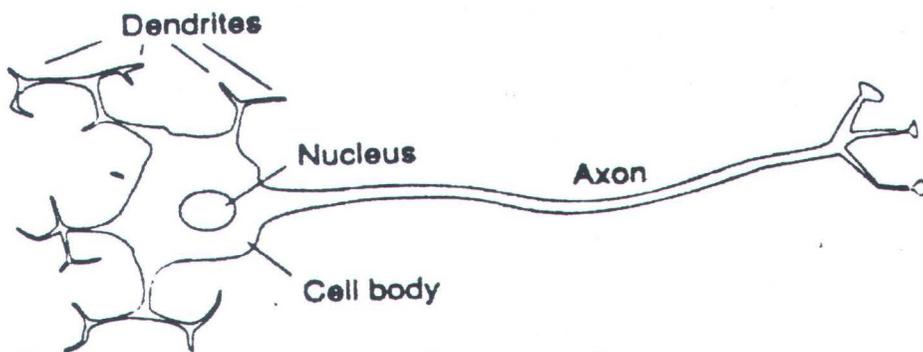
Sebagaimana telah disebutkan di atas bahwa sifat, serta hubungan antar neuron menjadi landasan dari jaringan saraf tiruan, sehingga terlebih dahulu

⁹ Patrick Henry Winston, *Artificial Intelligence*, Third Edition, Addison-Wesley Publishing Company, 1992, halaman 444-446

harus dipelajari tentang bagaimana cara kerja dari neuron pada jaringan saraf manusia, yang nantinya akan disimulasikan ke dalam sebuah algoritma tertentu.

3.1.1 Neuron pada Manusia

Neuron pada manusia, seperti pada gambar 3.1, terdiri dari sel tubuh, sebuah axon dan banyak dendrit. Axon berfungsi sebagai penghubung yang mengirimkan sinyal keluaran neuron ke neuron-neuron yang lainnya. Sedangkan dendrit berfungsi sebagai fasilitator yang menghubungkan neuron dengan axon dari neuron yang lainnya.



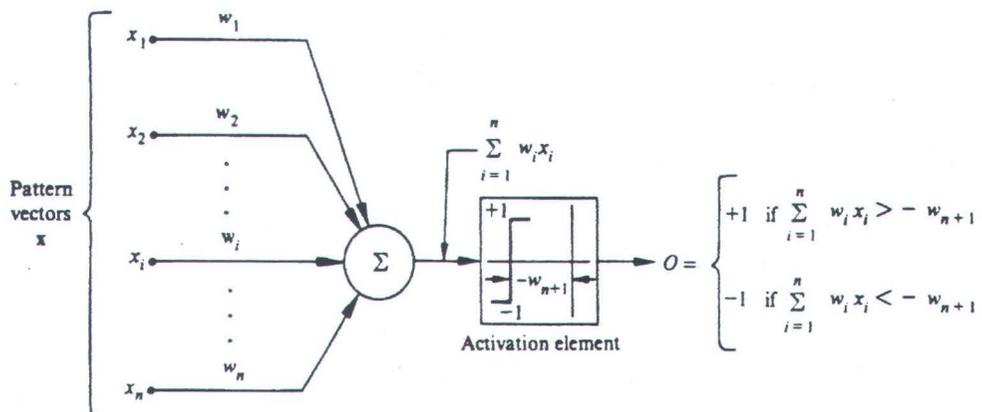
Gambar 3.1 Neuron pada manusia

Cara kerja dari neuron tidak lain adalah menggabungkan pengaruh dari semua masukan apakah pengaruh dari masukan tadi dapat mencapai *threshold* yang telah ditetapkan atau tidak. Jika nilai *threshold* tadi tercapai, maka neuron akan memproduksi keluaran dalam bentuk suatu pulsa tertentu yang diproses dari sel hingga dikeluarkan melalui axon. Jika hal ini terjadi, maka neuron sedang dalam keadaan aktif.

Axon mempengaruhi dendrit melalui jalur yang disebut sinapsis. Stimulasi dari beberapa sinapsis inilah yang membuat neuron dapat aktif ataupun tidak. Di sini terdapat pengaruh yang besar dari proses belajar dan dari masing-masing sinapsis yang berhubungan dengan derajat dari pulsa yang nantinya menentukan apakah sebuah neuron itu aktif atau tidak.

3.1.2 Neuron pada Jaringan Saraf Tiruan

Neuron pada jaringan saraf tiruan (agar lebih mudah selanjutnya kita sebut dengan neuron saja), terdiri dari sebuah node yang dihubungkan dengan neuron-neuron yang lain melalui sebuah penghubung yang cara kerjanya hampir sama dengan cara kerja hubungan dari axon, sinapsis dan dendrit, sebagaimana dapat dilihat pada gambar 3.2.



Gambar 3.2 Neuron pada jaringan saraf tiruan

Tiap penghubung diasosiasikan dengan sebuah nilai bobot (w). Seperti pada sebuah sinapsis, nilai bobot menentukan derajat pengaruh dari sebuah neuron ke neuron yang lainnya. Pengaruh dari sebuah neuron ke neuron yang

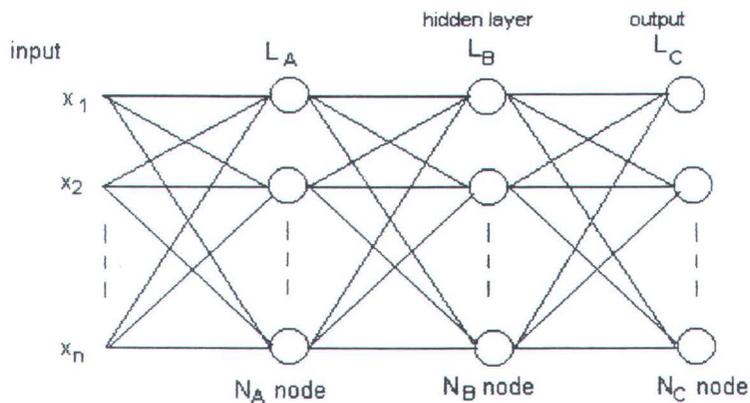
lainnya merupakan hasil kali dari nilai keluaran dari neuron-neuron yang masuk ke neuron (x) dengan nilai bobot (w) yang menghubungkan neuron-neuron tadi.

Tiap neuron dikombinasikan dengan sebuah fungsi aktivasi yang berfungsi sebagai penghubung dari penjumlahan semua nilai masukan dengan nilai keluarannya. Keluaran dari neuron inilah yang nantinya akan menentukan apakah sebuah neuron itu aktif ataukah tidak.

Sekarang telah diketahui dengan jelas mengenai cara kerja dari neuron, kegunaan dari nilai bobot, serta kegunaan dari fungsi aktivasi.

3.2 ARSITEKTUR JARINGAN SARAF

Jaringan saraf minimal terdiri dari dua *layer*, yaitu *layer* masukan (*input layer*) dan *layer* keluaran (*output layer*). Di antara *layer* masukan dan *layer* keluaran bisa juga terdapat satu atau beberapa *layer* tambahan yang biasanya disebut sebagai *layer* tersembunyi (*hidden layer*), seperti pada gambar 3.3. Jaringan saraf yang memiliki susunan seperti ini berarti semua neuron pada *layer* masukan akan terhubung ke semua neuron pada *layer* tersembunyi. Kemudian *layer* tersembunyi ini juga akan terhubung ke *layer* lain yang lebih tinggi dan seterusnya hingga sampai ke *layer* keluaran.



Gambar 3.3 Arsitektur jaringan saraf dengan satu *layer* tersembunyi

Setiap neuron pada masing-masing *layer* mempunyai fungsi aktivasi dan pola hubungan antar neuron pada satu *layer* yang sama. Pola hubungan antar neuron pada satu *layer* bisa terhubung satu dengan yang lain atau tidak terhubung sama sekali. Penyusunan neuron-neuron dalam sebuah *layer* serta pola hubungan antar neuron dan antar *layer* inilah yang disebut sebagai arsitektur jaringan saraf.¹⁰

Salah satu masalah yang timbul dalam penyusunan arsitektur jaringan saraf ini adalah menentukan jumlah neuron pada masing-masing *layer* serta inisialisasi nilai bobot awal pada masing-masing neuron yang nantinya akan digunakan untuk proses belajar dari jaringan saraf itu sendiri.

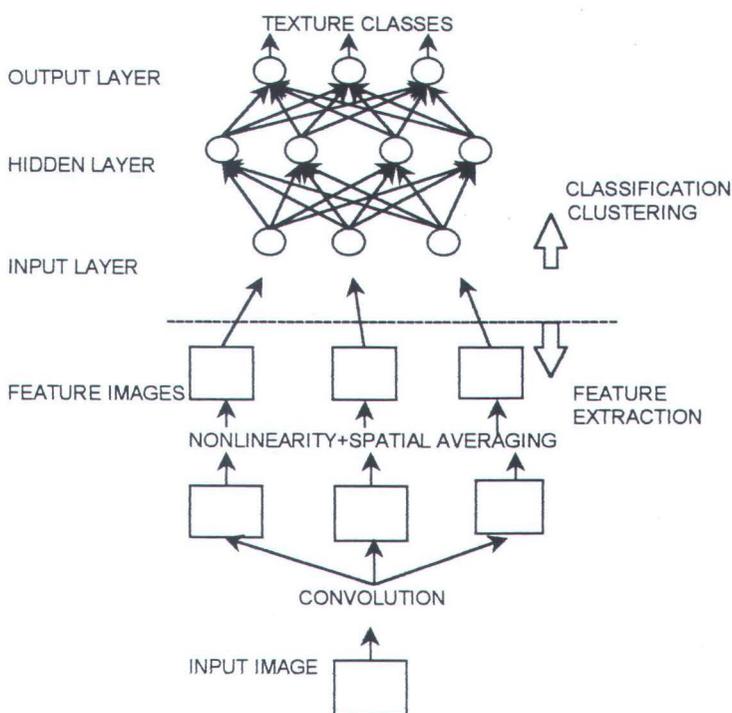
Pada bagian ini akan digambarkan metoda untuk melekatkan susunan *multichannel filter* ke dalam arsitektur jaringan saraf. Masalah untuk

¹⁰ Laurene Fausett, *Fundamental of Neural Network*, Prentice-Hall International, Inc., New Jersey, 1994, halaman 12

mendapatkan set yang minimal dari filter untuk menyelesaikan problem segmentasi tekstur menjadi masalah pelatihan (*training*) pada jaringan saraf.

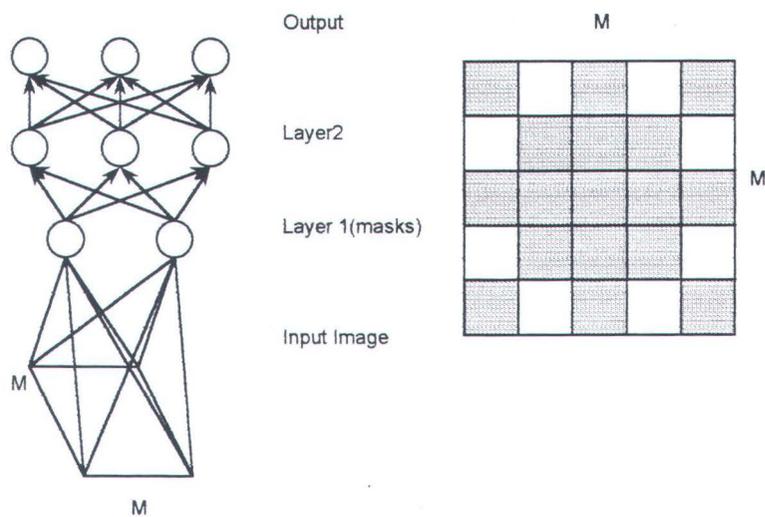
Jaringan dilatih dengan metoda *supervised* untuk meminimalkan kesalahan dalam pengklasifikasian tekstur. Dengan memetakan jaringan ke dalam susunan *multichannel filter*, dapat dilihat bahwa proses belajar yang optimal pada jaringan saraf akan memberikan set yang minimal dari filter untuk ekstraksi karakteristik citra, sebagaimana *classifier* untuk pengklasifikasian tekstur, sehingga akan meningkatkan keakuratan pada proses segmentasi.

Pada bagian ini kita menggabungkan dua fase skema *multichannel filter* yang ditunjukkan dalam gambar 3.4 ke dalam sebuah arsitektur jaringan saraf. Ternyata operasi pada unit dalam jaringan hampir sama dengan ekstraksi fitur dalam satu *channel*. Tiap neuron menghitung perkalian antara masukan dan nilai bobot yang diikuti dengan fungsi energi.



Gambar 3.4 Skema *multichannel filter*

Oleh karena kesamaan ini, tiap *channel* dalam gambar 3.4 dapat diganti dengan neuron seperti gambar 3.5. Masukan ke jaringan langsung dari $M \times M$ *window* (M adalah ukuran *mask*) dan keluarannya digunakan untuk menentukan kelas dari piksel pusat dalam *mask window* yang bersangkutan. Sebutan *mask* digunakan untuk neuron di *layer 1* karena secara esensi mempunyai operasi yang sama dengan *channel* pada *multichannel filter* dan nilai bobot dapat dianggap sebagai koefisien *mask*.



Gambar 3.5 Jaringan saraf untuk klasifikasi tekstur

Memilih ukuran *mask* yang sesuai merupakan suatu permasalahan tersendiri. Secara ideal, *mask* harus cukup besar untuk bisa menangkap karakteristik dari variasi *gray level* pada tekstur dan cukup kecil untuk menentukan batasan antar tekstur yang berlainan. Menambah ukuran *mask* akan menambah jumlah masukan pada jaringan saraf pula, yang akan membuatnya semakin jauh dari piksel pusatnya. Oleh karena itu pada tugas akhir ini, konfigurasi masukan ditentukan sebagaimana pada gambar 3.5. Hanya piksel

yang diarsir saja yang digunakan sebagai masukan untuk mengklasifikasikan piksel pusatnya dalam jendela (*window*) $M \times M$. Sehingga jumlah masukan ke *layer* satu sama dengan jumlah piksel yang diarsir dalam sebuah *mask*.

Untuk representasi keluaran, hanya keluaran yang menghasilkan nilai aktivasi yang tertinggi yang akan memenangkan persaingan atau dengan kata lain, citra masukan akan dihubungkan dengan kelompok yang berkorespondensi dengan keluaran yang mempunyai nilai tertinggi ini.

3.3 JARINGAN SARAF BACKPROPAGATION¹¹

Perancangan arsitektur jaringan saraf pada tugas akhir ini menggunakan metoda *backpropagation*. Adapun pendekatan yang digunakan adalah dari *multichannel filter*. Dibandingkan dengan memakai filter biasa, jaringan saraf dapat dipakai untuk menemukan set yang minimal dari filter, sehingga baik pengekstraksian karakteristik citra dan pengklasifikasian dilakukan oleh jaringan saraf yang sama. Jaringan saraf dipakai pada seleksi karakteristik citra dan ekstraksi untuk meningkatkan kecepatan dan ketepatan dari *classifier*.

3.3.1 Fungsi Aktivasi

Fungsi aktivasi (V_i = nilai aktivasi dari neuron ke- i dalam *layer* l) yang dipergunakan pada tiap-tiap neuron adalah sebagai berikut :

$$V_i^l = \Phi^l \left(\sum_{j=1}^{N^{(l-1)}} \omega_{ij}^l V_j^{l-1} \right); \quad i = 1, \dots, N^l; l = 1, \dots, L$$

¹¹ Anil K. Jain and Kalle Karu, *Learning Texture Discrimination Masks*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 2, February 1996, halaman 195-198

dimana L menggambarkan jumlah *layer* dari jaringan saraf dengan jumlah neuron tiap *layer* sebanyak N -buah. Hubungan antar *layer* adalah terhubung penuh yang mempunyai nilai bobot (ω) pada masing-masing hubungannya.

Sedangkan fungsi energi (Φ^l) yang dipergunakan adalah sebuah fungsi tangen hiperbol sebagaimana pada persamaan (3.3.1-1).

$$\Phi^l(x) = \begin{cases} \tanh(\beta x^2) & \text{untuk } l = 1, \\ \tanh(\beta x) & \text{untuk } l > 1. \end{cases} \quad (3.3.1-1)$$

adapun konstanta yang dipergunakan adalah $\beta = 0.25$.

Fungsi energi ini harus diturunkan, yang nantinya persamaan turunan dari fungsi energi ini akan digunakan untuk mencari nilai bobot yang sesuai bagi proses segmentasi. Untuk menurunkan fungsi energi ini harus diketahui dahulu tentang definisi dari fungsi tangen hiperbol, sebagaimana yang dapat dilihat pada persamaan (3.3.1-2).¹²

$$y = \tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.3.1-2)$$

Turunan dari fungsi tangen hiperbol itu sendiri seperti pada persamaan (3.3.1-3) berikut ini :

$$y^1 = (1 + y)(1 - y) \quad (3.3.1-3)$$

¹² Edwin J. Purcell, University of Arizona, *Calculus with Analytic Geometry*, 3rd Edition, Prentice Hall, Inc., 1978, halaman 425

Dari persamaan (3.3.1-2) dan persamaan (3.3.1-3), dapat ditentukan turunan dari persamaan (3.3.1-1), sebagaimana bisa dilihat pada persamaan berikut ini :

$$\Phi^{l1}(x) = \begin{cases} 2\beta x(1 + \tanh(\beta x^2))(1 - \tanh(\beta x^2)) & \text{untuk } l = 1, \\ \beta(1 + \tanh(\beta x))(1 - \tanh(\beta x)) & \text{untuk } l > 1. \end{cases}$$

3.3.2 Nilai Bobot

Penginisialisasian nilai bobot secara acak (*random*) akan mempengaruhi apakah jaringan saraf dapat mencapai global minimal dari kesalahan atau hanya lokal minimal saja, serta mempengaruhi kecepatan konvergennya juga. Sedangkan perubahan nilai bobot ini tergantung dari fungsi aktivasi dan turunan fungsi aktivasi yang dipergunakan oleh jaringan saraf. Karena alasan inilah, diperlukan suatu upaya agar inisialisasi nilai bobot tidak menyebabkan, baik fungsi aktivasi dan turunan fungsi aktivasi sama dengan nol.¹³

Oleh karena itu, nilai dari inisialisasi bobot tidak boleh terlalu besar. Namun sebaliknya, jika nilai inisialisasi bobot terlalu kecil, maka akan menyebabkan lama dalam mencapai konvergen. Sehingga biasanya -begitu juga yang digunakan pada tugas akhir ini- inisialisasi nilai bobot diset secara acak antara -0,5 dan 0,5.

Proses perubahan nilai bobot merupakan proses yang penting dalam sebuah jaringan saraf. Karena proses ini akan berpengaruh pada kinerja dari jaringan saraf itu sendiri, baik dari segi kecepatan konvergennya maupun dari segi ketepatannya. Oleh karena itu diperlukan suatu metoda yang mampu

¹³ Laurene Fausett, *Fundamental of Neural Network*, Prentice-Hall International, Inc., New Jersey, 1994, halaman 296-297

menghasilkan perubahan nilai bobot yang sesuai dengan permasalahan yang dihadapi.

Pada tugas akhir ini digunakan perubahan nilai bobot berdasarkan persamaan berikut ini :

$$\omega_{ji}(t+1) = \omega_{ji}(t) + \frac{\eta \delta_j O_i + m(\omega_{ji}(t) - \omega_{ji}(t-1))}{\Delta \omega_{ji}}$$

dimana η adalah *learning rate* yang ditetapkan dengan nilai = 0.01, sedangkan m adalah *momentum* yang ditetapkan dengan nilai = 0.5.

3.3.3 Metoda Belajar

Proses belajar (*training*) merupakan proses untuk merubah nilai bobot, agar didapatkan nilai bobot baru yang sesuai dengan tugas dari jaringan saraf. Dalam proses pelatihan terdapat dua metoda belajar, yaitu *supervised* dan *unsupervised*. Namun pada tugas akhir ini hanya akan membahas tentang metoda belajar *supervised*, dimana jaringan harus dilatih terlebih dahulu sebelum dioperasikan.

Metoda *supervised* ini membandingkan nilai keluaran dari jaringan dan nilai keluaran target yang diinginkan. Untuk proses belajar dibutuhkan informasi dari masukan dan informasi dari keluaran yang disebut dengan *training set*. Setiap nilai masukan mempunyai nilai keluaran yang bersesuaian dengannya. Proses belajar akan berhenti bila jaringan saraf sudah menghasilkan nilai keluaran yang diinginkan.

Tujuan dari proses belajar ini adalah untuk meminimalisasi kesalahan (E) pada jaringan saraf, seperti pada persamaan (3.3.3-1), dengan cara mengubah

nilai koneksi (bobot) antar neuron secara terus-menerus hingga didapat nilai bobot akhir yang menghasilkan kesalahan keluaran terkecil.

$$E = \frac{1}{2P} \sum_{j=1}^P \sum_{i=1}^{N^i} (d_j^i - V_j^{i,L})^2 \quad (3.3.3-1)$$

Pada persamaan (3.3.3-1) di atas, P menunjukkan jumlah pola belajar yang digunakan dalam jaringan saraf, d adalah nilai keluaran yang diinginkan, sedangkan V adalah keluaran yang sesungguhnya dari jaringan saraf sebagai respon kepada pola masukan yang ke- i . Nilai bobot yang menghasilkan nilai kesalahan rata-rata terkecil inilah yang akan digunakan pada proses selanjutnya.

3.3.4 Algoritma *Backpropagation*

Algoritma *backpropagation* yang dipergunakan pada tugas akhir kali ini dapat diuraikan sebagaimana berikut ini :

Step 0 Inisialisasi bobot pada semua koneksi (diset dengan nilai acak yang kecil)

Step 1 Apabila belum memenuhi kondisi penghentian, maka dilakukan langkah

1.1 – 1.2

Step 1.1 Pada tiap data proses belajar (*training set*) dilakukan langkah 1.1.1–

1.1.6

Propagasi maju :

Step 1.1.1 Tiap unit masukan (x_i ; $i = 1, \dots, n$) menerima sinyal masukan x_i dan menyebarkannya ke unit-unit selanjutnya

Step 1.1.2 Pada tiap unit tersembunyi (z_i ; $i = 1, \dots, n$) dilakukan penjumlahan terhadap sinyal masuk yang telah dikalikan dengan nilai bobot :

$$net_{pj}^h = \sum_{i=1}^N \omega_{ji}^h x_{pi} + \theta_j^h$$

Dihitung sinyal keluaran dari unit tersebut dengan fungsi aktivasi :

$$i_{pj} = f_j^h(net_{pj}^h)$$

kemudian keluaran dikirim ke unit selanjutnya.

Step 1.1.3 Pada tiap unit keluaran (O_k , $k = 1, \dots, m$) sinyal yang masuk dijumlahkan :

$$net_{pk}^o = \sum_{i=1}^M \omega_{ki}^o i_{pi} + \theta_j^o$$

dan dihitung keluarannya dengan :

$$O_{pk} = f_j^o(net_{pk}^o)$$

Propagasi balik terhadap kesalahan :

Step 1.1.4 Pada tiap unit keluaran (O_k , $k = 1, \dots, m$) dihitung kesalahannya :

$$\delta_{pk}^o = (y_{pk} - o_{pk}) f_k^o'(net_{pk}^o)$$

dan dihitung perubahan bobot :

$$\Delta_{kj}^o = \eta \delta_{kj}^o i_{pj}$$

Step 1.1.5 Pada tiap unit tersembunyi dihitung kesalahan dengan :

$$\delta_{pj}^h = f_j^h'(net_{pj}^h) \sum_k \delta_{pk}^o \omega_{kj}^o$$

dan dihitung perubahan bobot :

$$\Delta_{ij}^h = \eta \delta_{pj}^h x_i$$

Mengubah nilai bobot dan bias yang baru :

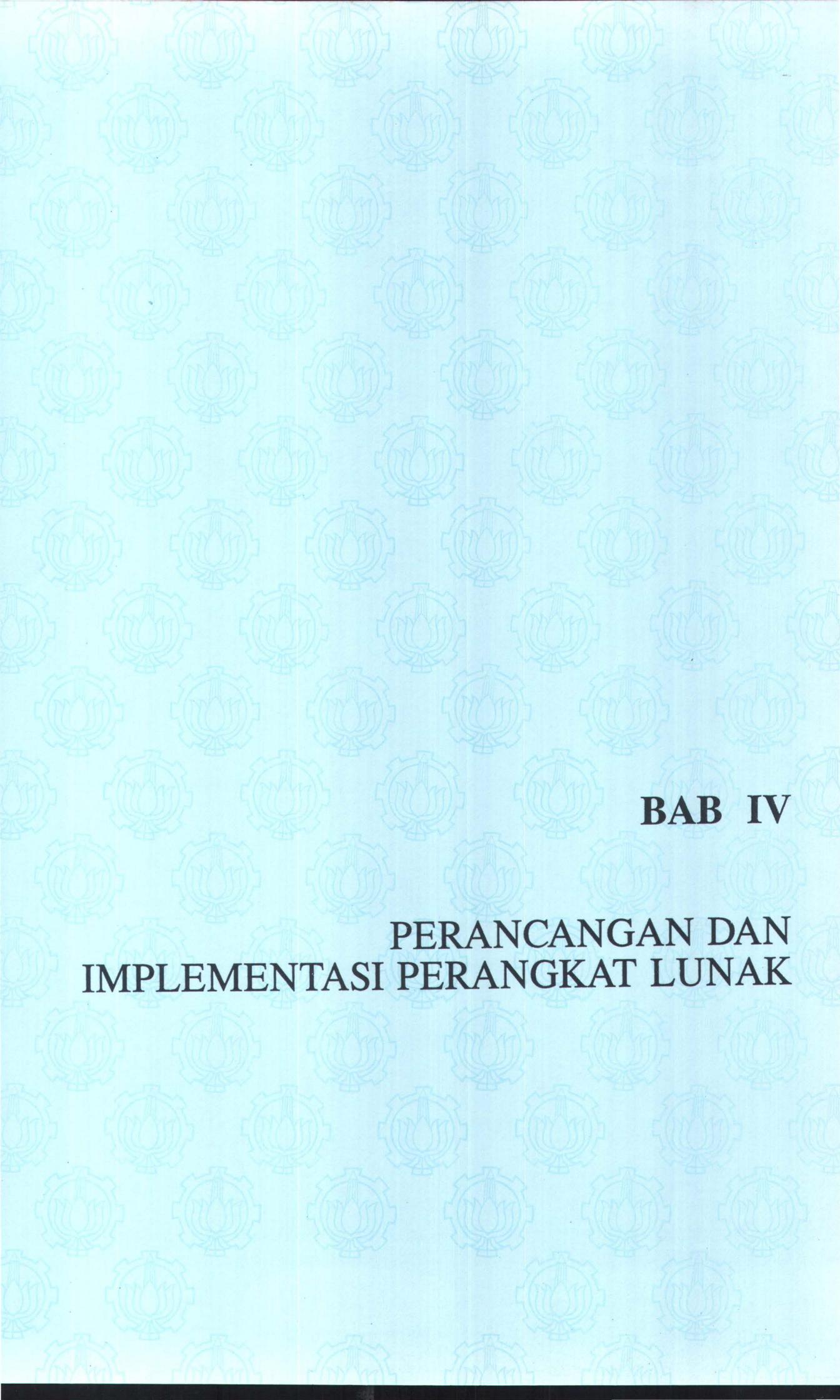
Step 1.1.6 Pada tiap unit keluaran dilakukan perubahan bobot dengan :

$$\omega_{ij}^o(t+1) = \omega_{ij}^o(t) + \Delta_{ij}$$

Pada tiap unit tersembunyi dilakukan perubahan bobot dengan :

$$\omega_{ij}^h(t+1) = \omega_{ij}^h(t) + \Delta_{ij}$$

Step 1.2 Kondisi penghentian



BAB IV

**PERANCANGAN DAN
IMPLEMENTASI PERANGKAT LUNAK**

BAB IV

PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK

Pada bab ini dibahas mengenai perancangan perangkat lunak yang meliputi deskripsi sistem, perancangan data dan perancangan proses, serta implementasi berbagai perancangan tersebut dalam program.

4.1 DESKRIPSI PERANGKAT LUNAK

Segmentasi merupakan proses pengelompokan citra berdasarkan bagian-bagian atau tekstur tertentu. Untuk melakukan proses tersebut, dipergunakan metoda jaringan saraf. Tahapan-tahapan prosesnya secara keseluruhan adalah penerimaan masukan dari pengguna (*user*), proses pelatihan dalam jaringan saraf dengan menggunakan metoda *backpropagation*, proses pengelompokan keluaran jaringan saraf ke dalam kelas-kelas yang bersesuaian (klasifikasi) dan penampilan citra keluaran (citra hasil segmentasi).

4.1.1 Masukan

Jaringan saraf yang dipergunakan pada tugas akhir ini adalah jaringan saraf *backpropagation* dengan metode *supervised*, sehingga masukan yang diberikan berupa citra yang terdiri dari tekstur-tekstur yang sudah dikenali oleh sistem atau dengan kata lain dipergunakan proses pelatihan terlebih dahulu agar jaringan saraf dapat mengenali tekstur-tekstur yang akan disegmentasikan.

Untuk keperluan ini, telah disediakan citra dengan tipe tekstur yang periodik dan terstruktur. Citra yang diinputkan kemudian dikonversikan ke dalam nilai *gray level* dengan nilai antara 0 – 255 dan kemudian citra dinormalisasikan dengan nilai *mean* = 0 dan nilai *varian* = 1. Ukuran citra yang digunakan adalah 128 x 128 piksel dengan jumlah tekstur maksimal tiap citra adalah 5.

4.1.2 Proses dalam Jaringan Saraf

Proses dalam jaringan saraf sendiri terdiri dari dua bagian, yaitu proses pelatihan jaringan saraf dan proses *feed forward* (proses *test* jaringan saraf). Proses pelatihan jaringan saraf berfungsi untuk melatih agar jaringan saraf dapat mengenali tekstur-tekstur yang akan disegmentasikan. Dari proses ini dihasilkan serangkaian nilai bobot yang digunakan pada proses *feed forward*. Hasil keluaran dari jaringan saraf inilah yang akan menentukan jenis tekstur yang terdapat pada suatu citra.

4.1.3 Pengklasifikasian

Proses pengklasifikasian merupakan proses untuk mengelompokkan citra hasil jaringan saraf pada kelas-kelas tekstur yang bersesuaian, sesuai dengan nilai node keluaran yang terbesar.

4.1.4 Penampilan Hasil

Hasil dari proses pengklasifikasian ditampilkan berupa citra dimana tekstur yang berbeda akan diberi warna yang berlainan.

4.2 PERANCANGAN DATA

Pada bagian ini diuraikan perancangan data yang meliputi data masukan, data saat pemrosesan dan data keluaran.

4.2.1 Data Masukan

Data masukan dari perangkat lunak ini berasal dari citra masukan yang bertipe bitmap, dengan ukuran 128 x 128 piksel. Selain itu, untuk mendapatkan nilai bobot hasil proses belajar, pengguna harus memberikan masukan mengenai ukuran *mask* dan jumlah *layer* yang akan dipergunakan pada proses segmentasi.

4.2.2 Data Saat Pemrosesan

Pada saat perangkat lunak menerima data masukan, ada beberapa data yang akan digunakan dalam proses yang bersesuaian dengan data masukan, yaitu :

- ✍ Ukuran dari *mask* yang dipergunakan untuk memfilter citra masukan. Ukuran *mask* ini mempengaruhi jumlah masukan ke *layer* satu, sebagaimana dapat dilihat dalam tabel berikut ini.

Ukuran <i>mask</i>	11x11	15x15	21x21
Jumlah masukan ke <i>layer</i> 1	41	57	81

- ✍ Jumlah *layer* yang akan mempengaruhi arsitektur dari jaringan saraf, yaitu dua dan tiga.
- ✍ Nilai bobot yang didapatkan dari proses belajar, sesuai dengan ukuran *mask* dan jumlah *layer*-nya, yang berfungsi sebagai koefisien *mask*. Nilai bobot ini disimpan dalam sebuah file dan dipanggil pada saat proses.

4.2.3 Data Keluaran

Pembuatan perangkat lunak pada tugas akhir ini bertujuan untuk mengelompokkan citra masukan ke dalam kelas tekstur tertentu. Sehingga keluaran yang dihasilkan berupa citra dengan ukuran sesuai ukuran citra masukan, yakni 128 x 128 piksel, dimana untuk kelas tekstur yang berlainan akan diberi warna berbeda.

4.3 PERANCANGAN PROSES

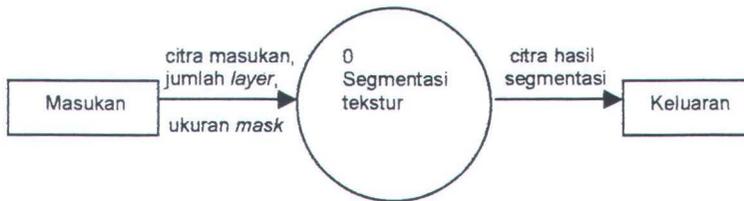
Dalam perangkat lunak ini terdapat tiga buah proses utama. Proses yang pertama adalah ekstraksi karakteristik dari citra masukan dengan menggunakan proses belajar, untuk mendapatkan serangkaian nilai bobot. Setelah itu dengan menggunakan nilai bobot tersebut, dilakukan proses yang kedua yaitu proses dalam *feed forward* itu sendiri. Selanjutnya proses klasifikasi untuk mengelompokkan citra berdasarkan tekstur yang bersesuaian, maka akan diperoleh citra hasil segmentasi.

Bagian ini akan menjelaskan lebih lanjut tentang diagram alir data serta hirarki modul dari perangkat lunak.

4.3.1 Diagram Alir Data (*Data Flow Diagram*)

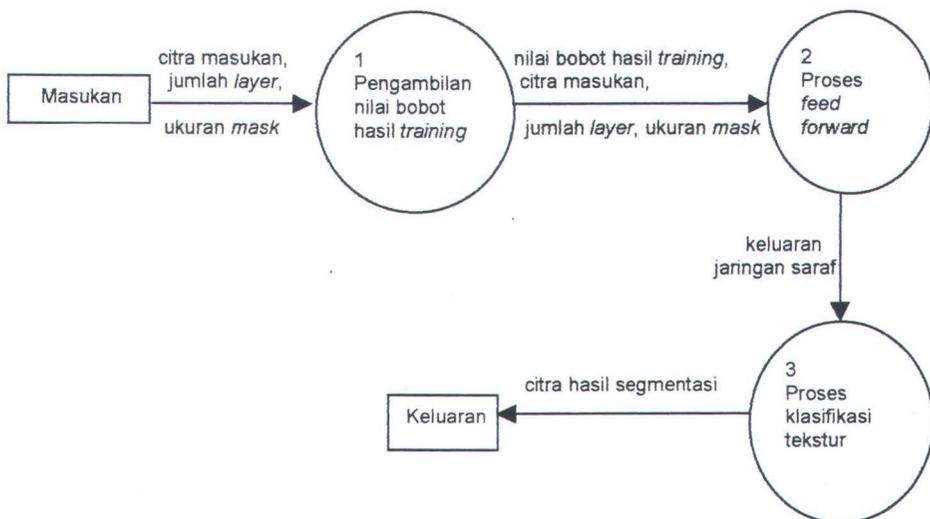
Diagram alir data (DFD) merupakan teknis grafis yang menggambarkan aliran informasi data dari masukan sampai keluaran. Diagram ini digunakan untuk menjelaskan berbagai proses yang akan dilakukan serta data-data yang diperlukan dalam tahap implementasi.

Susunan diagram alir data untuk perangkat lunak yang dibuat sebagaimana digambarkan pada gambar 4.1.



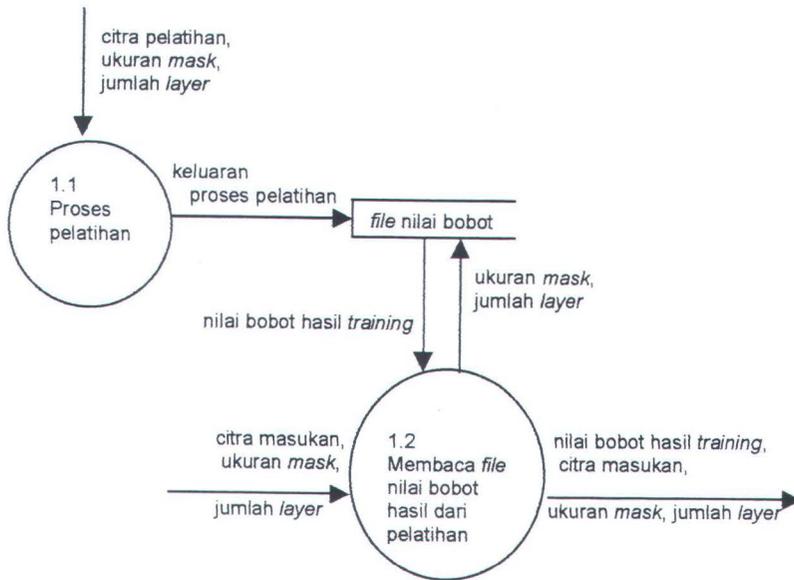
Gambar 4.1 Diagram alir data level 0

Data yang digunakan dalam level 0 ini berasal dari *file* masukan bertipe bitmap dengan ukuran 128 x 128 piksel, berupa citra yang terdiri dari maksimal 5 tekstur yang berlainan. Data bitmap yang diberikan kemudian dikonversikan ke dalam nilai *gray level*-nya, setelah itu dinormalisasi untuk diproses lebih lanjut.

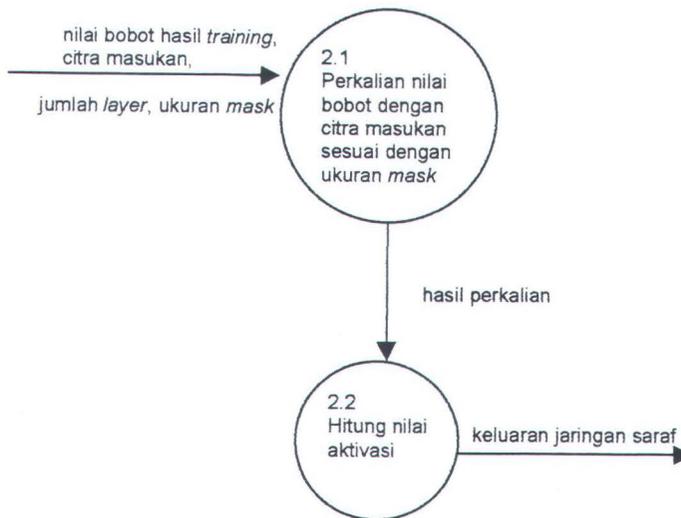


Gambar 4.2 Diagram alir data level 1, detail dari level 0

Diagram 1, 2 dan 3 pada gambar 4.2 masih terdiri dari beberapa proses yang dapat dilihat secara berturut-turut pada gambar 4.3 sampai dengan gambar 4.5.



Gambar 4.3 Diagram alir data level 2, detil dari pengambilan nilai bobot hasil *training*



Gambar 4.4 Diagram alir data level 2, detil dari proses *feed forward*

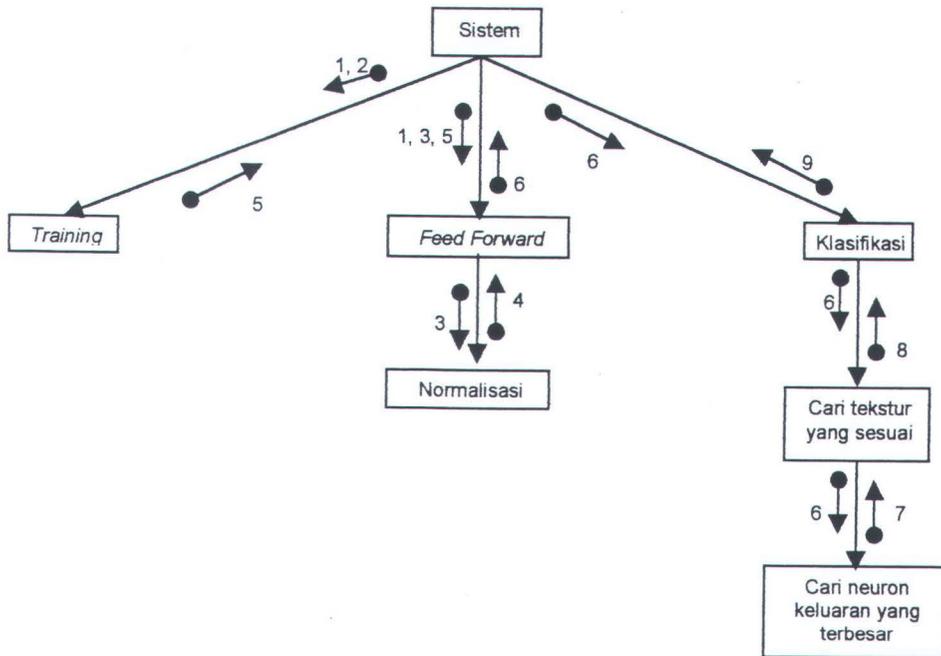


Gambar 4.5 Diagram alir data level 2, detil dari proses klasifikasi tekstur

Gambar 4.3, merupakan proses untuk mendapatkan karakteristik dari tekstur, yang diwakili oleh nilai bobot pada jaringan saraf. Gambar 4.4, merupakan proses *feed forward* dalam jaringan saraf. Sedangkan gambar 4.5, merupakan proses pengelompokan citra ke dalam kelas-kelas tekstur yang bersesuaian.

4.3.2 Hirarki Modul

Diagram disain modul yang diimplementasikan pada tugas akhir ini untuk proses segmentasi secara keseluruhan dapat dijabarkan pada gambar 4.6.



Gambar 4.6 Hirarki modul dari segmentasi tekstur

Keterangan :

1. data jaringan saraf (jumlah *layer*, ukuran *mask*)
2. citra pelatihan
3. citra masukan
4. citra normalisasi
5. nilai bobot
6. nilai keluaran jaringan saraf
7. neuron keluaran jaringan saraf terbesar

8. jenis tekstur
9. citra hasil segmentasi

4.4 IMPLEMENTASI PERANGKAT LUNAK

Pada bagian ini akan dijelaskan mengenai implementasi dari berbagai perancangan yang telah dijelaskan di atas, yang meliputi implementasi struktur data beserta implementasi dalam tahapan-tahapan proses.

4.4.1 Implementasi Struktur Data

Program yang dirancang terdiri dari program utama dan tiga buah unit, yaitu UType, UTraining dan USegmen. UType berisi deklarasi dari tipe-tipe data yang digunakan. Struktur dari unit ini dapat dilihat sebagai berikut :

```

Arr1      = array of double;
neuron    = record
            wgh, bWgh      : Arr1;
            act, errGrad  : double
            end;
list      = ^layer;
layer     = record
            node : array of neuron;
            before, next : list
            end;
Arr2      = array of array of double;
Arrcol    = array of array of Tcolor;
Toutput   = array of shortInt;
Arrbitmap = array of TBitmap;

const
  clArrTable : array[0..5] of TColor
    = (clYellow, clRed, clBlue, clAqua, clGreen, clBlack);

```

Unit UTraining berisi prosedur-prosedur yang berhubungan dengan proses pelatihan jaringan saraf untuk mendapatkan nilai bobot yang akan

dipergunakan pada proses segmentasi. Prosedur-prosedur pada unit ini dapat dilihat sebagai berikut :

```
procedure normalisasiImg(lb, tg:byte);
procedure reImg(var Timg:Arr2; lb, tg:byte);
procedure forwardProp;
procedure training(trMax:longword);
```

Unit USegmen berisi prosedur yang berhubungan dengan proses segmentasi citra dimana nilai bobot yang dipergunakan merupakan nilai bobot hasil dari proses pelatihan. Adapun prosedur dalam unit tersebut adalah sebagai berikut :

```
procedure segmen;
```

Program utama dari perancangan perangkat lunak ini menerima masukan berupa ukuran *mask* yang akan dipergunakan untuk memfilter citra, serta jumlah *layer* dalam jaringan saraf yang berfungsi untuk segmentasi citra tersebut. Selain itu masukan juga berupa citra bitmap yang disimpan dalam bentuk nilai *gray level*-nya pada sebuah data gb dengan tipe Arrcol (array of array of Tcolor). Selanjutnya data gb akan diproses lebih lanjut hingga akhirnya didapatkan citra hasil segmentasi. Data yang digunakan pada program utama ini dapat dilihat sebagai berikut :

```
{data untuk menyimpan nilai gray level dari citra masukan}
gb          : Arrcol;

{data untuk menyimpan gb pada saat proses}
img         : Arr2;

{data untuk menyimpan masukan ke jaringan saraf}
xInp       : Arr1;

{data untuk menyimpan keluaran dari jaringan saraf}
tOut       : TOutput;
```

```

{data untuk menyimpan ukuran mask}
maskSize      : byte;

{data untuk menyimpan layer pertama dan terakhir}
layer1, layer3 : list;

{data untuk menyimpan konfigurasi dari jaringan saraf}
lnRate, mmt, beta : double;

```

4.4.2 Implementasi Proses

Inti dari program ini adalah menjalankan tahapan-tahapan proses segmentasi, yaitu proses *training* (pelatihan) dan proses *feed forward* dari jaringan saraf. Implementasi dari tahapan-tahapan proses tersebut secara global dijelaskan di bawah ini.

4.4.2.1 Proses *training* (pelatihan)

Proses *training* merupakan proses awal yang harus dijalankan sebelum jaringan saraf dapat mengklasifikasikan citra berdasarkan tekstur-teksturnya. Dari proses ini didapatkan serangkaian nilai bobot yang akan digunakan pada proses segmentasi.

Yang dilakukan pada bagian ini adalah memproses masing-masing tekstur secara *feed forward* dengan nilai bobot awal yang diset secara acak dengan nilai antara $-0.5 - 0.5$. Masing-masing tekstur dilatih sebanyak 10% dari ukuran citra pelatihan untuk mendapatkan semua kemungkinan yang ada. Pengulangan proses sebanyak 10% dari ukuran citra ini membutuhkan waktu yang tidak sedikit.

Setelah itu nilai bobot diperbarui dengan cara menghitung nilai kesalahan secara *backpropagation* dan kemudian dimasukkan kembali ke dalam jaringan saraf. Proses ini terus diulang hingga didapat nilai total kesalahan yang

minimum. Dari sinilah didapatkan nilai bobot yang sesuai, yang nantinya akan digunakan dalam proses segmentasi.

```

procedure training(trMax:longword);
var err,temp,s          : double;
    count              : longword;
    f                  : string;
    Timg               : Arr2;
    z                  : word;
    x,y,lb,tg,i,p,q    : byte;
    aBmp               : Arrbitmap;
    sv                 : boolean;
begin
    inisialisasi nilai bobot dengan nilai acak;

    {f adalah file yang digunakan untuk menyimpan nilai bobot}
    f :='wgh'
      +IntToStr(maskSize)+MainForm.numlayerLabel.Caption
      +'_2.txt';

    membaca citra-citra pelatihan disimpan dalam aBmp;

    count:=0; //jumlah pelatihan
    err:=5;   //nilai kesalahan
    s:=0.5;   //nilai kesalahan minimal yang sudah disimpan
    sv:=false; //boolean tentang keberadaan file bobot

    randomize;

    While (count<trMax) and (err>=0.005) do
    begin
        temp := 0;

        for z := 0 to 1637 do //10% dari citra pelatihan
        begin
            p:=random(lb);
            q:=random(tg);

            {length(tOut) adalah jumlah citra pelatihan}
            for i := 0 to length(tOut)-1 do
            begin
                ubah citra pelatihan[i] ke nilai gray level-nya;

                normalisasiImg(lb,tg);
                reImg(Timg,lb,tg); //pembentukan citra untuk di-mask
                maskImg(Timg,lb,tg); //mengambil masukan dari citra
                setTarget(i);
                forwardProp;
                backProp;

                temp:=temp + errTotal;
            end
        end;
        err:=0.5*temp/8190; //perhitungan kesalahan
    end;
end;

```

```

menggambar grafik dari konvergenitas proses pelatihan;

If (err<0.5) and (err<s) then
begin
  saveWgh(err,f,Count);
  s:=err;
  If sv=false then sv:=true;
end;
inc(count);
end;
destroyImage(aBmp); //aBmp adalah citra-citra pelatihan
end;

procedure TMainForm.training_2ToolButtonClick
(Sender:TObject);
var tr:longword;
begin
  lnRate:=StrToFloat(lnRateEdit.Text);
  mmt :=StrToFloat(mmtEdit.Text);
  beta :=StrToFloat(betaEdit.Text);
  tr :=StrToInt(Edit1.Text); //jumlah training maksimal
  training(tr);
end;

```

4.4.2.2 Proses segmentasi (*testing*)

Proses segmentasi dilakukan setelah proses *training* selesai dijalankan secara sempurna. Implementasi dalam program secara global dapat dilihat berikut ini :

```

procedure segmen;
var error,epx,trNum      : double;
    p,q,z,lb,tg,bt,i,j  : byte;
    s                    : string;
    time1,time2         : TDateTime;
    ls                   : longWord;
    fl                   : boolean;
    max                  : Arr1;
    Timg                 : Arr2;
    hsl                  : Arrbitmap;
    temp                 : Arrcol;
begin
  bt:=maskSize div 2;
  p:=bt;
  q:=bt;
  lb:=MainForm.Imagel.Picture.Width;
  tg:=MainForm.Imagel.Picture.Height;
  error:=0;

```

```

ls:= lb*tg;
epx:=1/ls;

membaca citra hasil disimpan dalam hsl;

setNode; //set jaringan saraf

normalisasiImg(lb,tg);
reImg(Timg,lb,tg);

fl:=false; //boolean tentang ketiadaan file nilai bobot

{membaca file nilai bobot}
readWghOn('wgh'
  +IntToStr(maskSize)+MainForm.numlayerLabel.Caption
  +'_2.txt',error,trNum,fl);

If fl=true then Exit; //fl=true berarti belum di-training

While q<=(tg-1+bt) do
begin
  maskInp(Timg,p,q);
  forwardProp;

  for z:=0 to length(layer3.node)-1 do
    max[z]:=layer3.node[z].act;

  z:=getMax(max); //mencari neuron dengan output terbesar

  piksel diberi warna sesuai dengan kelasnya;

  If gb[p-bt,q-bt]<>temp[p-bt,q-bt]
    then error:=error+epx; //perhitungan error

  If p<(lb-1+bt) then inc(p)
  else
  begin
    p:=bt;
    inc(q);
  end;
end;

dispResult(gb,MainForm.Image2); //menampilkan citra hasil
end;

```

BAB V

**HASIL UJI COBA DAN
EVALUASI PERANGKAT LUNAK**

BAB V

HASIL UJI COBA DAN EVALUASI PERANGKAT LUNAK

Dalam bab ini diuraikan mengenai hasil uji coba perangkat lunak dengan beberapa masukan dan parameter yang berbeda. Terlebih dahulu dipaparkan mengenai lingkungan tempat perangkat lunak ini dijalankan, selain itu juga akan dipaparkan beberapa pembahasan dari uji coba yang dilakukan.

Berbagai uji coba di bawah ini dijalankan pada komputer dengan spesifikasi sebagai berikut:

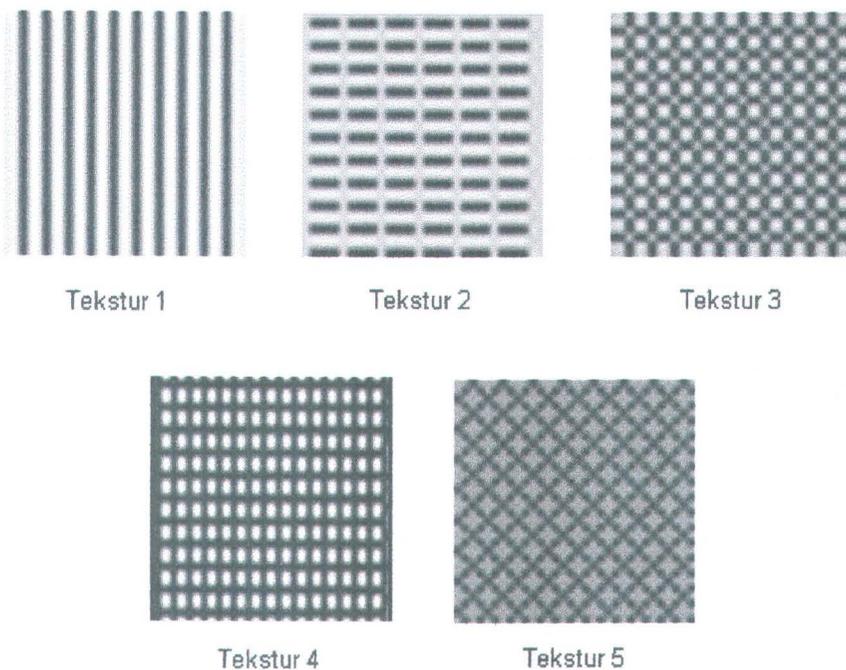
- ✍ Processor AMD-K6™ 3D
- ✍ Memory 32 MegaByte
- ✍ Hard Disk 3,5 GigaByte
- ✍ VGA Memory 1 MegaByte

Semakin tinggi spesifikasi komputer yang digunakan akan memberikan peningkatan pada kecepatan proses. Perangkat lunak ini dibuat dengan memakai bahasa pemrograman Delphi 5.0 dan menggunakan sistem operasi Windows 98.

Berikut ini diberikan beberapa hasil uji coba terhadap perangkat lunak yang dibuat. Tekstur yang digunakan dalam uji coba ini merupakan data bitmap 256 warna yang dibuat dengan menggunakan Adobe Photoshop 5.5. Untuk membentuk citra masukan, dilakukan penggabungan dari masing-masing tekstur. Hasil uji coba dibagi menjadi dua proses, yaitu proses pelatihan dan proses segmentasi.

5.1 PROSES PELATIHAN

Agar dapat dibandingkan, semua hasil uji coba menggunakan kombinasi dari tekstur yang sama. Setiap tekstur diubah menjadi nilai *gray-level* dan kemudian dinormalisasikan dengan nilai mean 0 dan nilai varian 1. Setelah itu secara acak dipilih 10% dari tekstur-tekstur pelatihan (semua kemungkinan dari *mask*) untuk melatih jaringan saraf agar didapatkan nilai bobot yang sesuai. Tekstur yang digunakan dalam proses pelatihan ini dapat dilihat dalam gambar 5.1.

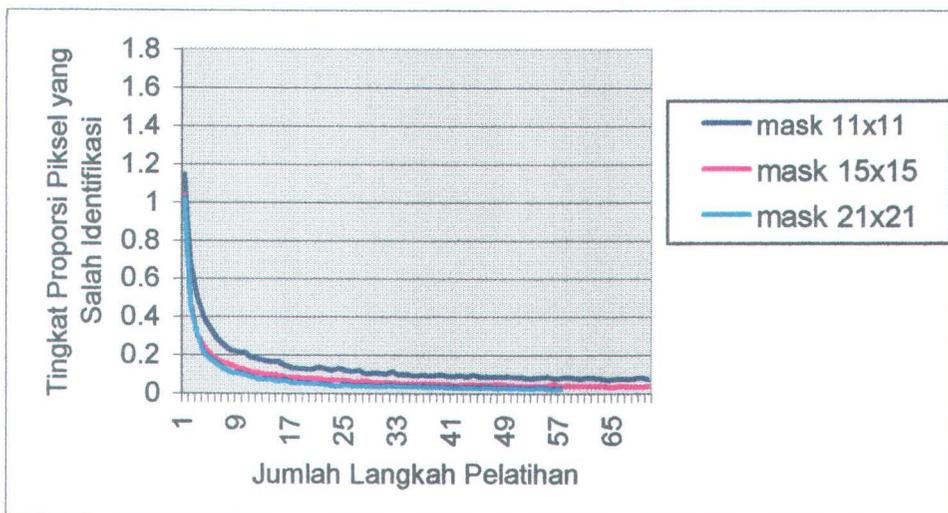


Gambar 5.1 Tekstur-tekstur untuk proses pelatihan

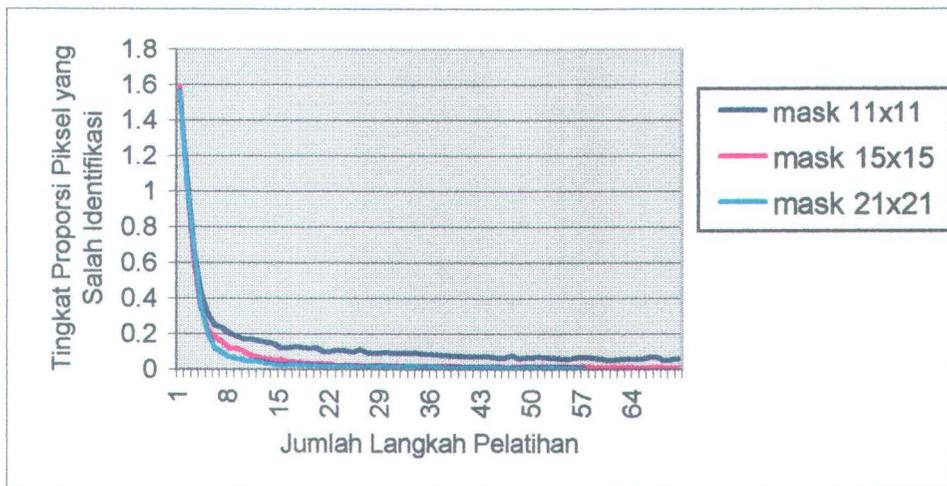
Parameter jaringan saraf yang dapat diubah-ubah adalah jumlah *layer*, jumlah node di *layer* 1 dan ukuran *mask*. Konstanta untuk *learning rate* adalah 0,01, momentumnya 0,5 dan derajat kemiringan dari fungsi aktivasi adalah 0,25.

Uji coba menggunakan ukuran *mask* 11x11 dengan jumlah masukan ke *layer* satu adalah 41, 15x15 dengan jumlah masukan ke *layer* satu adalah 57 dan 21x21 dengan jumlah masukan ke *layer* satu adalah 81. Jumlah *layer* yang digunakan ditentukan 2 dan 3 *layer*, sedangkan jumlah node di *layer* satu ditentukan sama dengan 20.

Gambar 5.2 dan 5.3 menunjukkan konvergenitas dari kesalahan klasifikasi (tingkat proporsi piksel yang salah identifikasi) pada sumbu y. Sedangkan sumbu x menunjukkan jumlah langkah pelatihan yang sudah dilakukan.



Gambar 5.2 Grafik konvergenitas dari metoda pelatihan dengan 2 *layer*



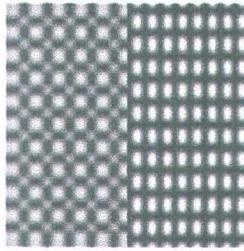
Gambar 5.3 Grafik konvergenitas dari metoda pelatihan dengan 3 *layer*

Dari gambar 5.2 dapat dilihat bahwa *mask* dengan ukuran 21x21 lebih cepat mencapai konvergen daripada ukuran 11x11 dan 15x15. Dari gambar 5.2 dan 5.3 didapatkan menambah ukuran *layer* menjadi tiga *layer* juga dapat meningkatkan kecepatan konvergen dibandingkan dengan dua *layer*. Proses pelatihan dihentikan jika nilai rata-rata kesalahan kurang dari 0,005 atau sudah tidak ada penurunan nilai kesalahan dalam waktu enam jam. Jika salah satu dari dua kondisi ini terpenuhi, maka jaringan saraf sudah konvergen.

Untuk menghindari lokal minimal, nilai bobot didapatkan dengan melatih program sebanyak lima kali untuk setiap nilai parameter yang sama dan memilih salah satu yang mempunyai nilai kesalahan terkecil.

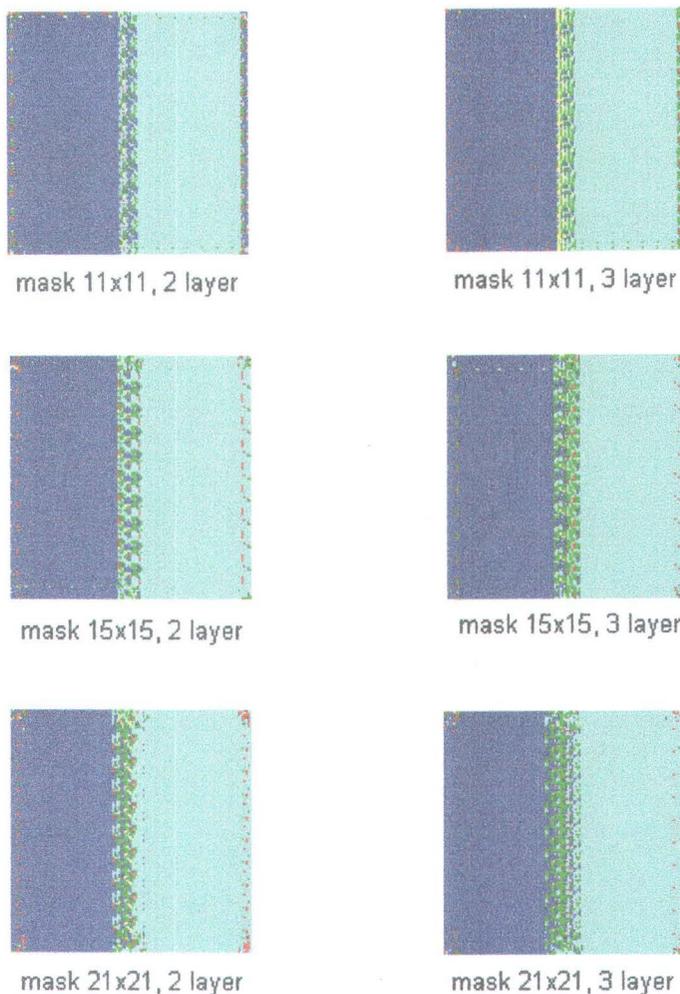
5.2 PROSES SEGMENTASI

Pada uji coba pertama diberikan masukan citra berukuran 128x128 piksel yang merupakan gabungan dari tekstur 3 dan tekstur 4, seperti pada gambar 5.4.



Gambar 5.4 Citra masukan untuk uji coba pertama

Setelah diuji cobakan pada program, hasilnya dapat dilihat pada gambar 5.5.

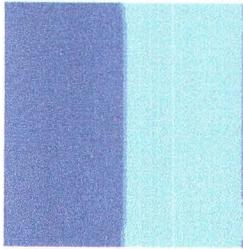


Gambar 5.5 Hasil uji coba 1

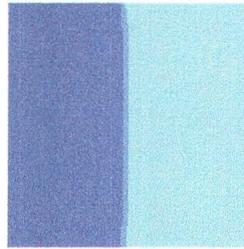
Hasil segmentasi di atas didapat tanpa melalui proses pemfilteran lebih lanjut untuk menghilangkan noise. Nilai kesalahan proses segmentasi ini dapat dilihat dalam tabel berikut ini :

Jumlah layer	Ukuran mask		
	11x11	15x15	21x21
2	0,029	0,032	0,043
3	0,035	0,059	0,065

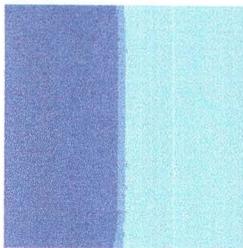
Dengan menggunakan fasilitas filter median pada Adobe Photoshop 5.5, didapatkan hasil seperti pada gambar 5.6 :



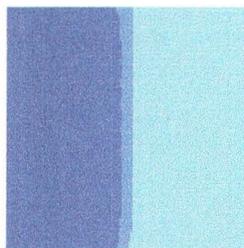
mask 11x11, 2 layer



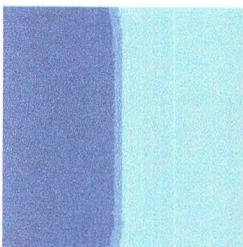
mask 11x11, 3 layer



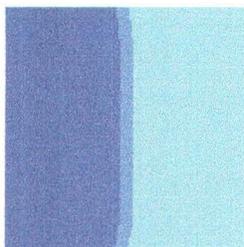
mask 15x15, 2 layer



mask 15x15, 3 layer



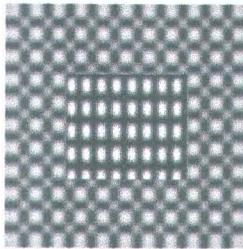
mask 21x21, 2 layer



mask 21x21, 3 layer

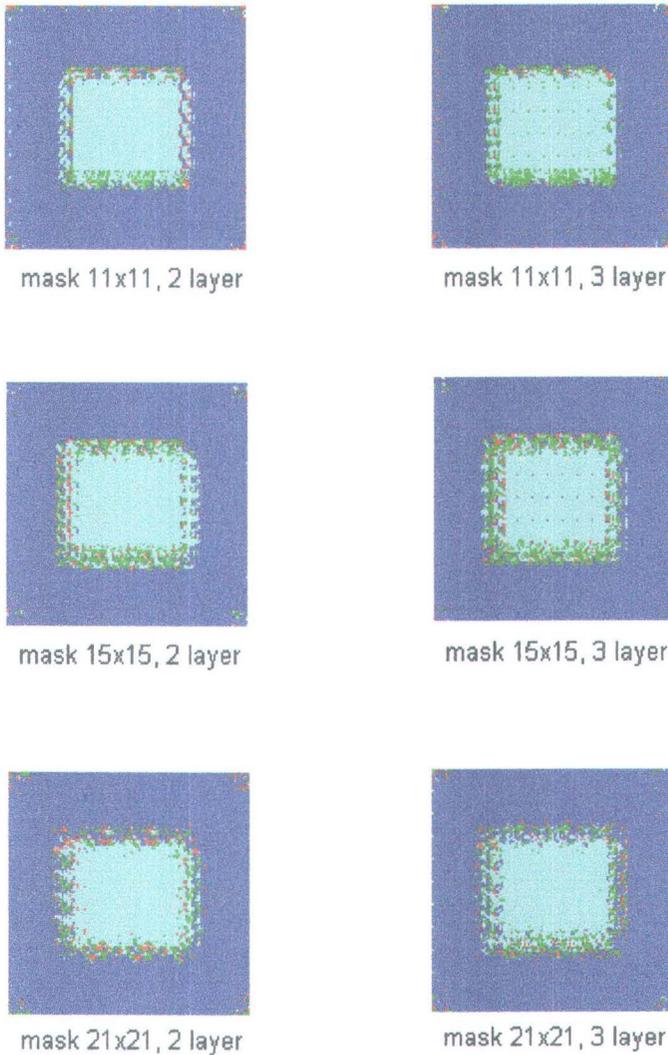
Gambar 5.6 Hasil uji coba 1 dengan menggunakan filter median

Pada uji coba kedua diberikan masukan citra yang merupakan gabungan dari tekstur 3 dan tekstur 4 dengan bentuk yang berbeda, seperti pada gambar 5.7.



Gambar 5.7 Citra masukan untuk uji coba kedua

Setelah diuji cobakan pada program, hasilnya dapat dilihat pada gambar 5.8.

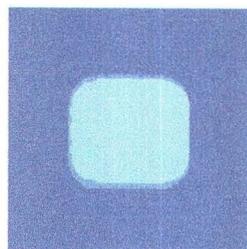


Gambar 5.8 Hasil uji coba 2

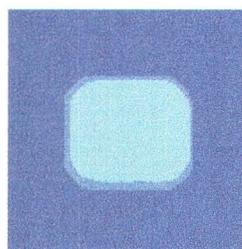
Hasil segmentasi di atas didapat tanpa melalui proses pemfilteran lebih lanjut untuk menghilangkan noise. Nilai kesalahan proses segmentasi ini dapat dilihat dalam tabel berikut ini :

Jumlah layer	Ukuran mask		
	11x11	15x15	21x21
2	0,036	0,052	0,042
3	0,056	0,073	0,054

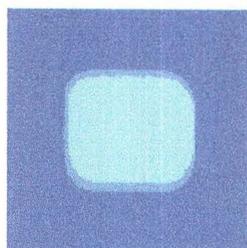
Dengan menggunakan fasilitas filter median pada Adobe Photoshop 5.5, didapatkan hasil seperti pada gambar 5.9.



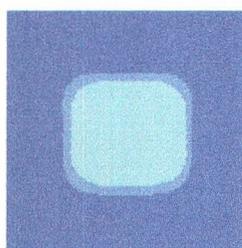
mask 11x11, 2 layer



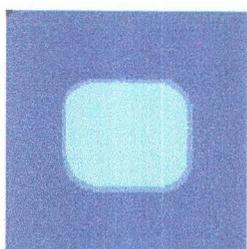
mask 11x11, 3 layer



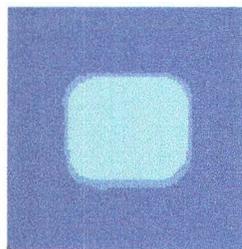
mask 15x15, 2 layer



mask 15x15, 3 layer



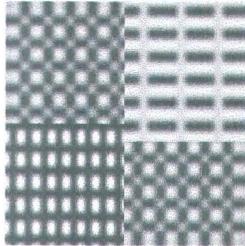
mask 21x21, 2 layer



mask 21x21, 3 layer

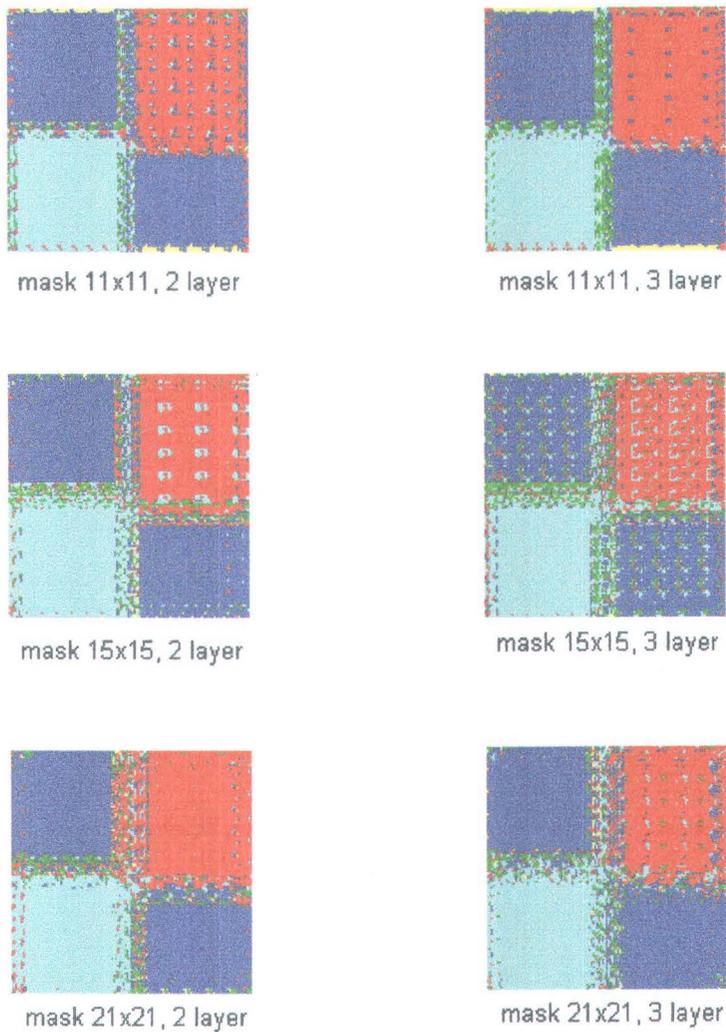
Gambar 5.9 Hasil uji coba 2 dengan menggunakan filter median

Pada uji coba ketiga diberikan masukan citra yang merupakan gabungan dari tekstur 2, tekstur 3 dan tekstur 4 dengan bentuk seperti pada gambar 5.10.



Gambar 5.10 Citra masukan untuk uji coba ketiga

Setelah diuji cobakan pada program, hasilnya dapat dilihat pada gambar 5.11.

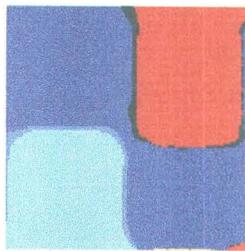


Gambar 5.11 Hasil uji coba 3

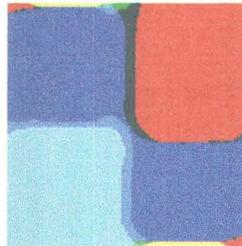
Hasil segmentasi di atas didapat tanpa melalui proses pemfilteran lebih lanjut untuk menghilangkan noise. Nilai kesalahan proses segmentasi ini dapat dilihat dalam tabel berikut ini :

Jumlah layer	Ukuran mask		
	11x11	15x15	21x21
2	0,074	0,067	0,070
3	0,100	0,119	0,082

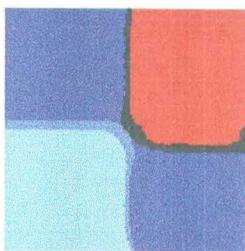
Dengan menggunakan fasilitas filter median pada Adobe Photoshop 5.5, didapatkan hasil seperti pada gambar 5.12.



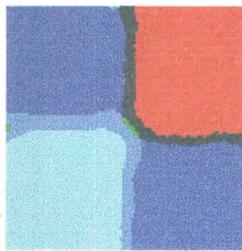
mask 11x11, 2 layer



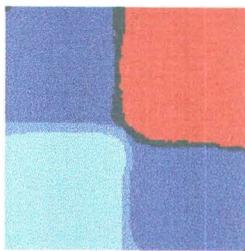
mask 11x11, 3 layer



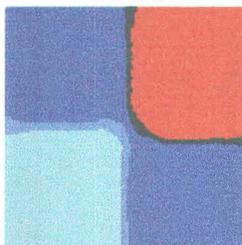
mask 15x15, 2 layer



mask 15x15, 3 layer



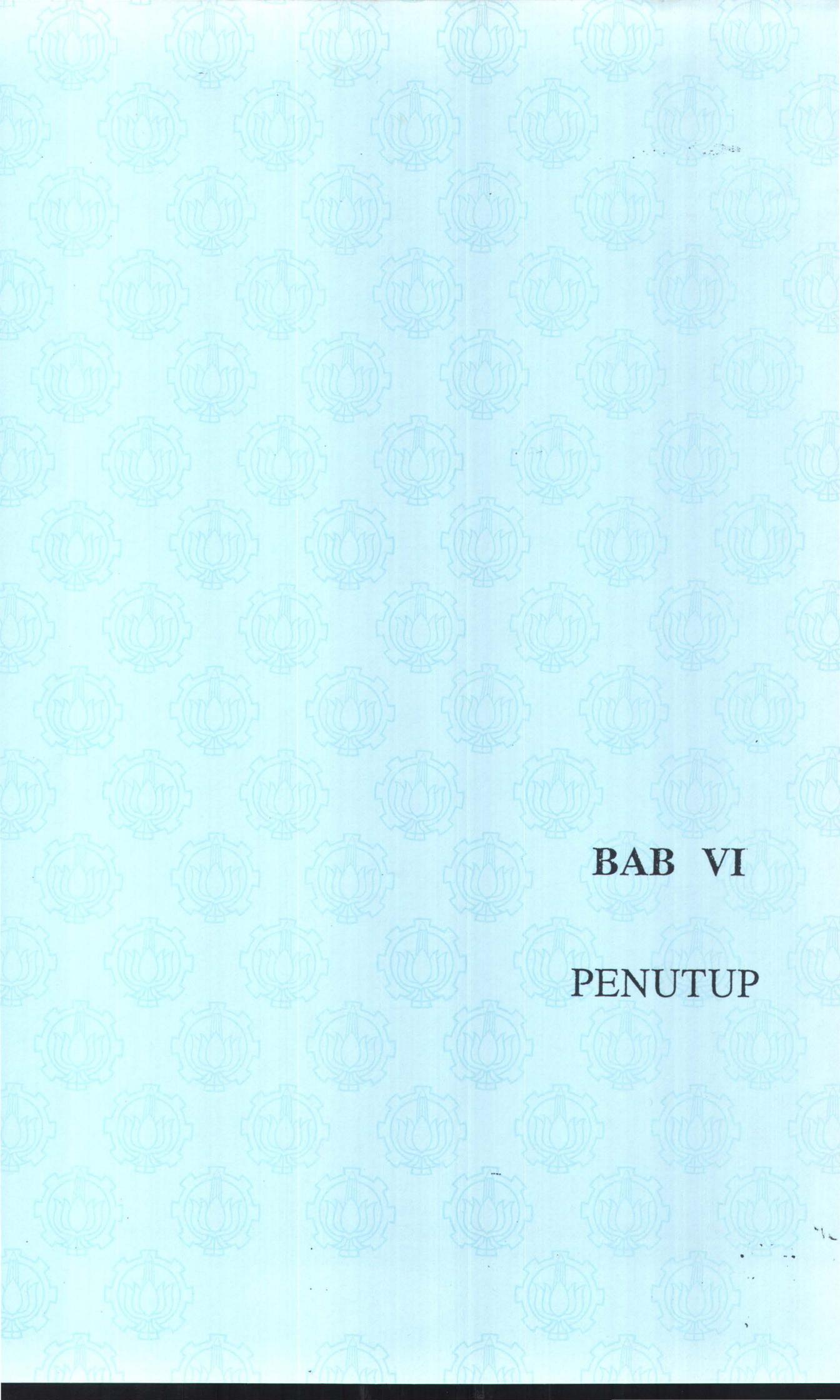
mask 21x21, 2 layer



mask 21x21, 3 layer

Gambar 5.12 Hasil uji coba 3 dengan menggunakan filter median

Dari uji coba pertama, kedua dan ketiga dapat dilihat bahwa dua *layer* pada umumnya lebih baik daripada tiga *layer*. Sedangkan meningkatkan ukuran *mask* tidak selalu berarti meningkatkan kesalahan. Uji coba ini membutuhkan dua sampai tiga hari untuk proses pelatihan dan satu sampai dua menit untuk mengklasifikasikan gambar.



BAB VI

PENUTUP

BAB VI

PENUTUP

Pada bab-bab sebelumnya, yaitu dari bab I sampai bab V telah diuraikan mengenai perancangan perangkat lunak ini, mulai dari latar belakang, teori-teori pendukung, perumusan tahapan-tahapan yang dilalui, sampai dengan implementasi program disertai hasil uji coba dan analisa. Pada bab ini diuraikan beberapa hal yang dapat disimpulkan dari hasil pengujian program dan beberapa saran dengan harapan untuk lebih menyempurnakan perancangan yang telah dibuat ini.

6.1 KESIMPULAN

Dari hasil uji coba perangkat lunak ini, dapat diketahui bahwa jaringan saraf dapat digunakan untuk mengklasifikasikan citra ke dalam tekstur-teksturnya. Jumlah *layer* dalam jaringan saraf mempengaruhi ukuran kesalahan segmentasi. Dari uji coba yang sudah dilakukan, jaringan saraf dengan dua *layer* memiliki tingkat kesalahan yang optimal.

Ukuran *mask* juga mempengaruhi tingkat kesalahan segmentasi. Tetapi peningkatan ukuran *mask* tidak selalu berarti pengurangan tingkat kesalahan. Hal ini dapat dipahami, karena ukuran *mask* harus cukup besar untuk bisa menangkap karakteristik dari tekstur. Tetapi sebaliknya ukuran *mask* harus cukup kecil, sehingga bisa mendeteksi batasan dari tekstur secara tepat. Sehingga penggunaan ukuran *mask* juga harus disesuaikan dengan pola tekstur yang akan disegmentasikan.

Hasil akhir dari proses segmentasi sangat tergantung dari proses training yang sudah dilakukan. Karena nilai bobot yang didapat pada proses training digunakan untuk mengenali tekstur-tekstur dalam sebuah citra dan juga untuk mengklasifikasikannya ke dalam kelompok yang sesuai.

Ukuran kesalahan yang digunakan untuk menghitung ketelitian jaringan saraf memiliki hubungan dengan citra hasil segmentasi. Apabila nilai kesalahan kecil, maka hasil akhir dari pengelompokan citra berdasarkan kelas teksturnya juga semakin bagus. Dan sebaliknya, semakin besar nilai kesalahan, citra yang dihasilkan juga semakin jelek.

6.2 SARAN

Perancangan perangkat lunak ini masih bisa dikembangkan lebih lanjut, beberapa saran untuk pengembangan diberikan berikut ini,

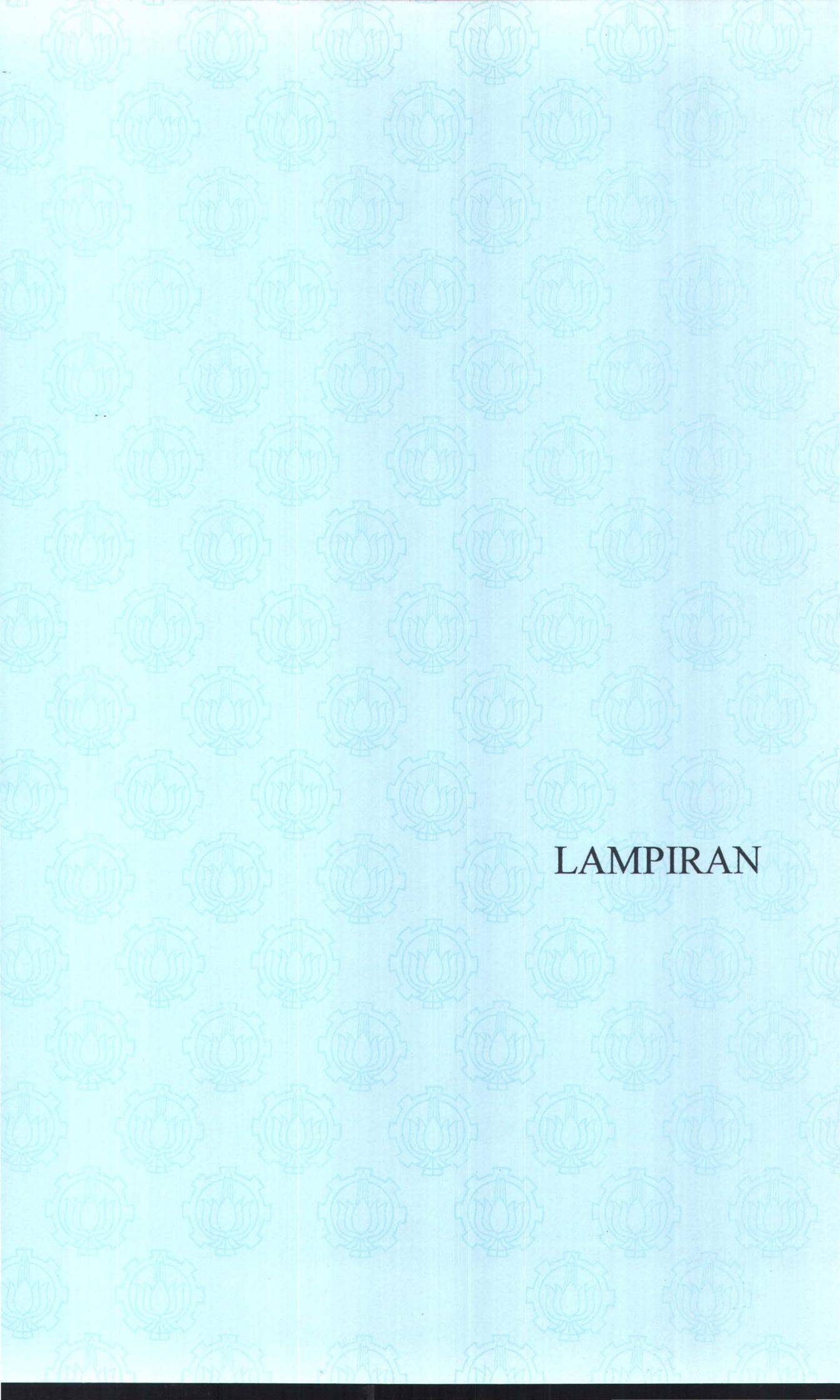
- ❖ Algoritma untuk melakukan proses pelatihan menggunakan metoda *supervised*, sehingga terbatas pada tekstur-tekstur yang sudah dikenali oleh sistem, dalam hal ini bisa dikembangkan menggunakan metoda *unsupervised*.
- ❖ Pengembangan dengan menambahkan filter median untuk menangani kesalahan pengelompokan yang biasa terjadi pada bagian batas antara kelas tekstur yang berbeda.
- ❖ Penambahan metoda node pruning untuk meningkatkan kecepatan konvergen pada proses pelatihan.



DAFTAR PUSTAKA

DAFTAR PUSTAKA

1. Fausett, Laurene; *Fundamental of Neural Network*, Prentice-Hall International, Inc., New Jersey, 1994.
2. Fu, Limin; *Neural Network in Computer Intelligence*, McGraw Hill, 1994.
3. Gonzales, Rafael C. and Woods, Richard E; *Digital Image Processing*, Addison-Wesley Publishing Company Inc., 1993.
4. Gose, Earl; Johnsonbaugh, Richard and Jost, Steve; *Pattern Recognition and Image Analysis*, Prentice Hall, Inc., 1996.
5. Jain, Anil K. and Karu, Kalle; *Learning Texture Discrimination Masks*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 2, February 1996, halaman 195-198.
6. Pitas, Ioannis; *Parallel Algorithms for Digital Image Processing Computer Vision and Neural Networks*, Wiley Series in Parallel Computing, 1993.
7. Purcell, Edwin J; University of Arizona, *Calculus with Analytic Geometry*, 3rd Edition, Prentice Hall, Inc., 1978.
8. Schalkoff, Robert J; *Pattern Recognition Statistical, Structural and Neural Approaches*, Clemson University, John Wiley and Sons, Inc., 1992.
9. Walpole, Ronald E. and Myers, Raymond H; *Ilmu Peluang dan Statistika untuk Insinyur dan Ilmuwan*, Edisi ke-4, diterjemahkan oleh Dr. R.K. Sembiring, Penerbit ITB, Bandung, 1995.
10. Winston, Patrick Henry; *Artificial Intelligence*, Third Edition, Addison-Wesley Publishing Company, 1992.
11. Young, Tzay Y. and Fu, King-Sun; *Handbook of Pattern Recognition and Image Processing*, Academic Press, Inc., 1986.



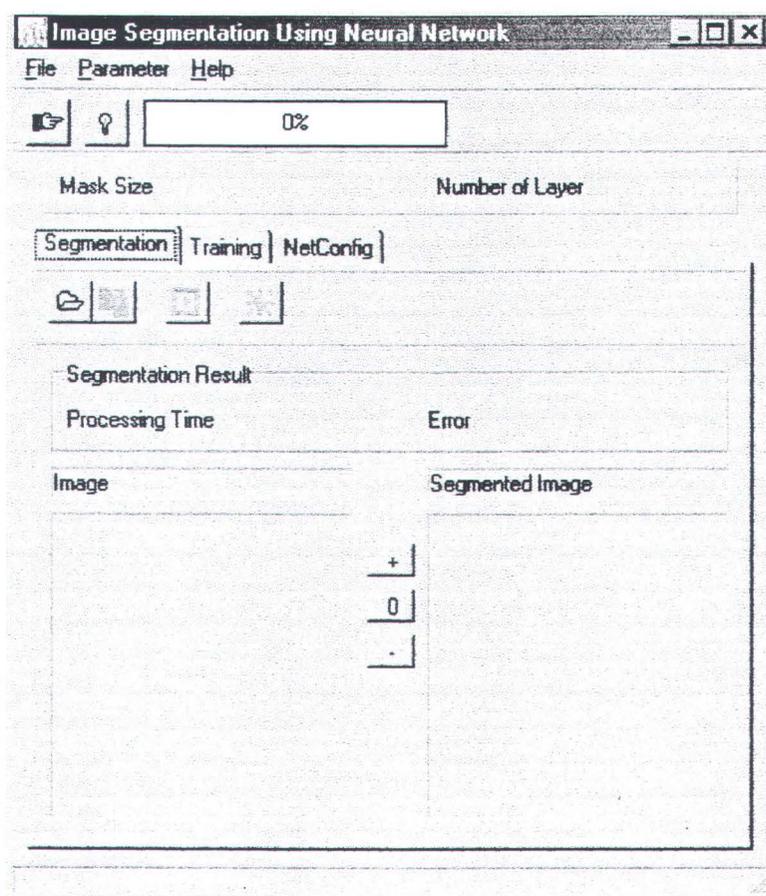
LAMPIRAN

LAMPIRAN

A. PETUNJUK PEMAKAIAN (*USER MANUAL*)

Antarmuka Perangkat Lunak

Perangkat lunak ini memiliki tiga menu utama, yaitu menu **File**, **Parameter** dan **Help**. Selain menu utama, perangkat lunak ini juga dilengkapi dengan dua buah tombol cepat (*speedbutton*), yaitu **Exit** dan **About**, sebuah penghitung prosentase proses (*gauge*), serta tiga *tabsheet*, yaitu **Segmentation**, **Training** dan **NetConfig**.

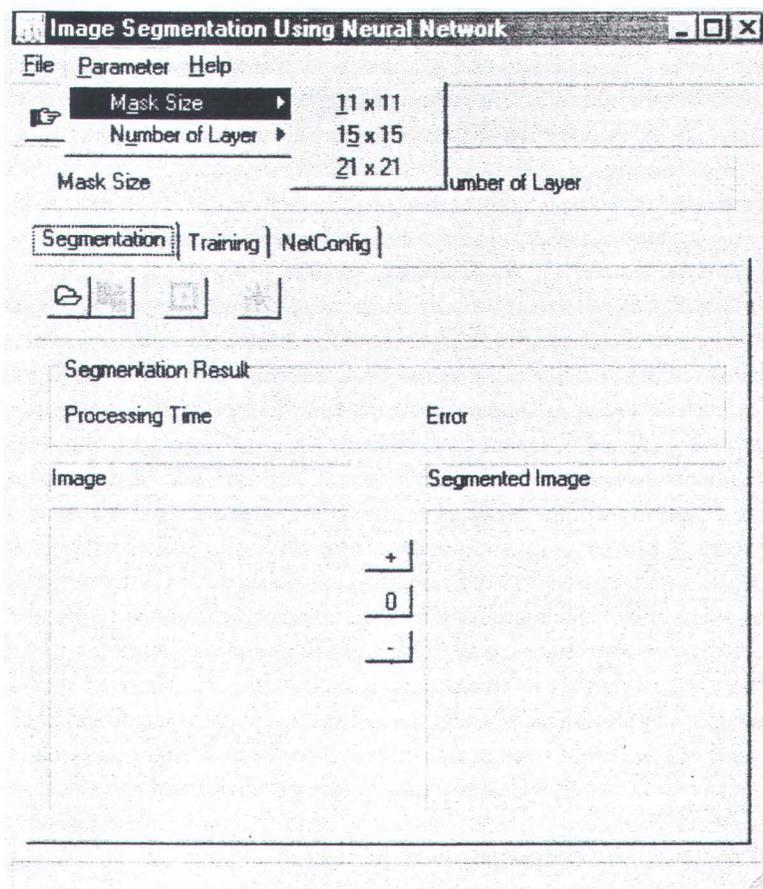


Gambar L.1 Tampilan Program

Menu yang pertama adalah menu **File** yang terdiri atas submenu **Exit** untuk menghentikan program.

Menu yang kedua adalah menu **Parameter** yang berhubungan dengan pilihan parameter yang disediakan oleh perangkat lunak. Menu ini terdiri dari dua submenu, yaitu :

- **Mask Size**, untuk menentukan ukuran *mask* yang digunakan. Submenu ini terdiri dari tiga pilihan submenu, yaitu **11 x 11**, **15 x 15** dan **21 x 21**.
- **Number of Layer**, untuk menentukan jumlah *layer* yang digunakan oleh jaringan saraf. Submenu ini terdiri dari dua pilihan submenu, yaitu **2 layers** dan **3 layers**.



Gambar L.2 Tampilan Menu Parameter

Menu yang ketiga adalah menu **Help** yang terdiri atas submenu **About** untuk memberikan informasi tentang perangkat lunak yang sedang aktif.

Dua *speedbutton* yang ada pada perangkat lunak ini berfungsi sebagaimana submenu yang bersesuaian, sedangkan *gauge* berfungsi untuk mengetahui prosentase dari laju perangkat lunak.

Adapun tiga buah *tabsheet* yang terdapat pada perangkat lunak ini dapat dijelaskan sebagai berikut :

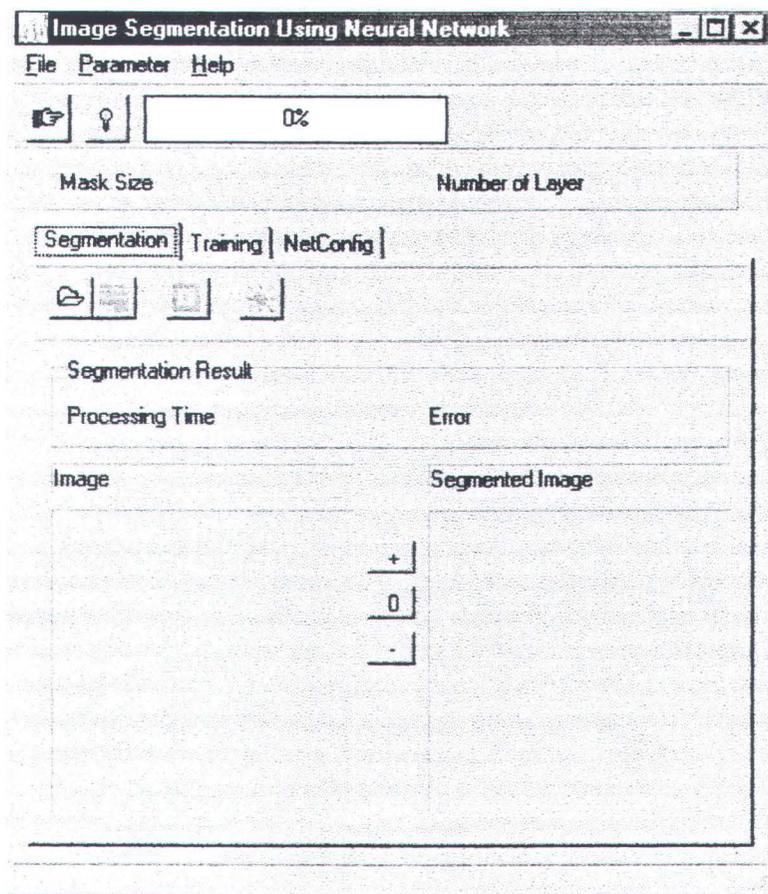
- **TabSheet Segmentation**

Berhubungan dengan proses segmentasi yang akan dilakukan terhadap berkas citra masukan. Pada bagian ini terdapat dua kelompok *speedbutton*. Kelompok pertama terdiri dari empat *speedbutton*, yaitu :

- **Open Image**, untuk membuka berkas citra masukan yang akan disegmentasi.
- **Save Segmented Image**, untuk menyimpan citra hasil proses segmentasi.
- **Segmentation**, untuk melakukan proses segmentasi.
- **Segmented Image with Median Filter**, untuk menampilkan citra hasil proses segmentasi yang telah difilter dengan filter median.

Sedangkan kelompok kedua terdiri dari tiga *speedbutton*, yaitu :

- **Zoom In**, untuk memperbesar ukuran citra yang ditampilkan.
- **Zoom None**, untuk mengembalikan ukuran citra ke ukuran aslinya.
- **Zoom Out**, untuk memperkecil ukuran citra yang ditampilkan.

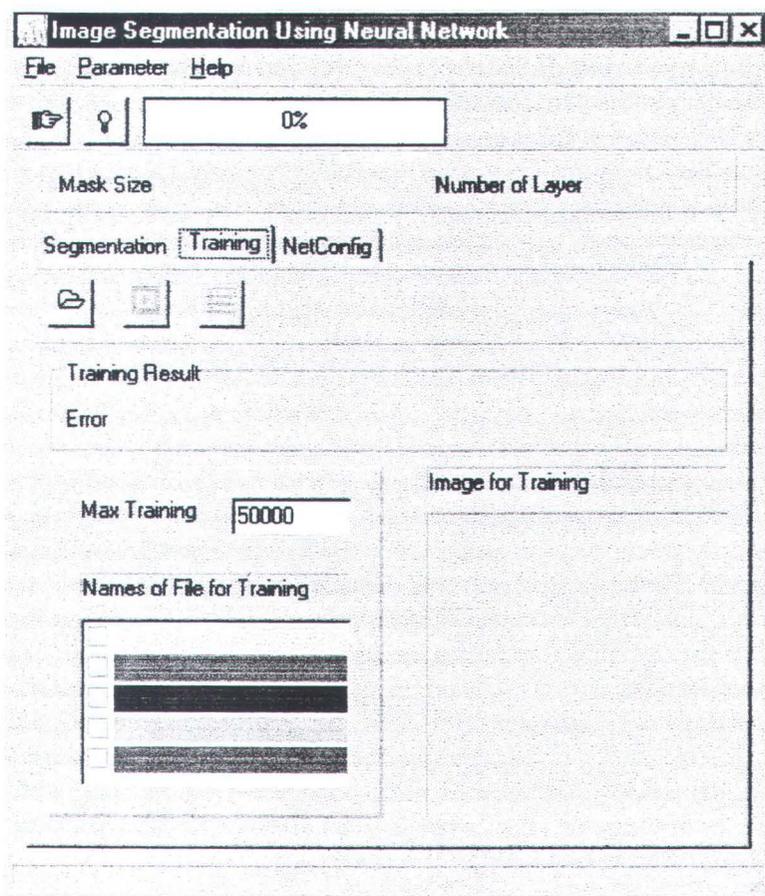


Gambar L.3 Tampilan TabSheet Segmentation

- **TabSheet Training**

Berhubungan dengan proses pelatihan yang akan dilakukan terhadap berkas-berkas citra pelatihan. Pada bagian ini terdapat tiga *speedbutton*, yaitu :

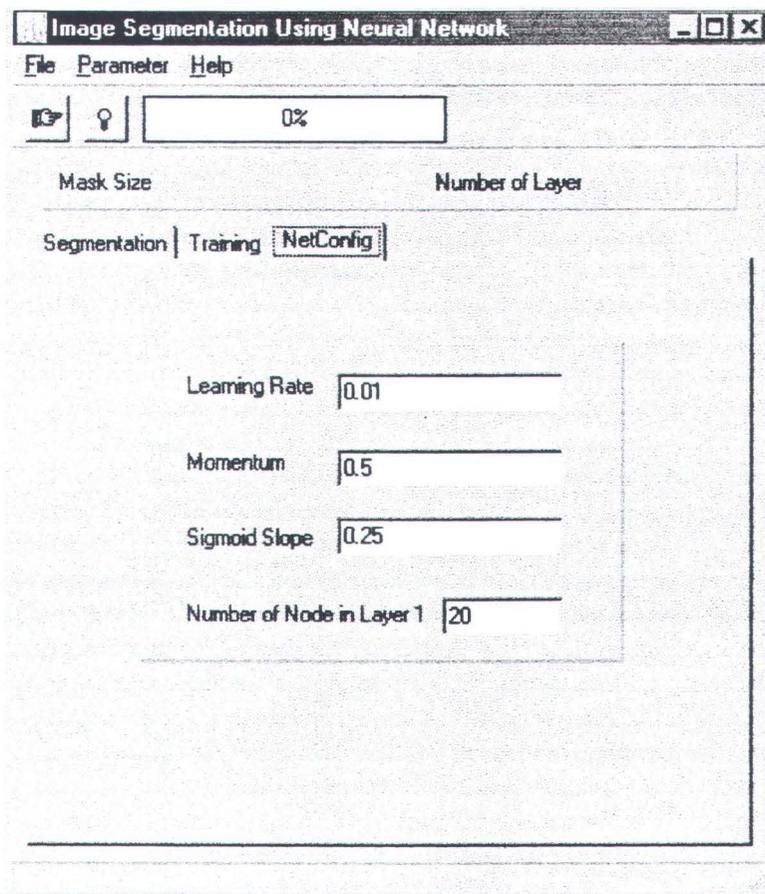
- **Open Image**, untuk membuka berkas-berkas citra pelatihan yang akan diproses untuk mendapatkan nilai bobot yang sesuai.
- **Training**, untuk melakukan proses pelatihan.
- **Show Graphic of Training**, untuk menunjukkan grafik konverginitas dari metoda pelatihan.



Gambar L.4 Tampilan TabSheet Training

- **TabSheet NetConfig**

Berhubungan dengan konfigurasi dari parameter jaringan saraf yang dapat diubah-ubah oleh pengguna.

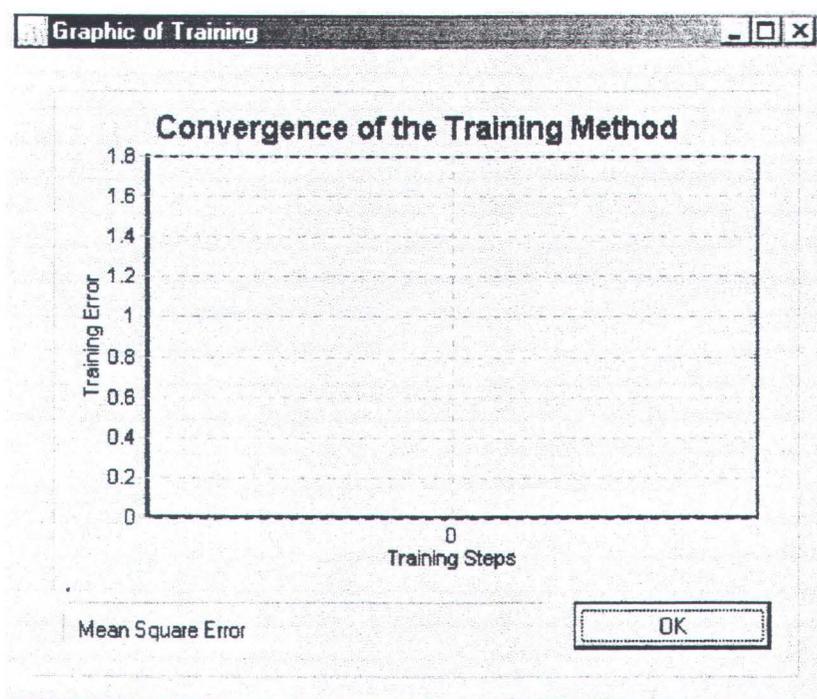


Gambar L.5 Tampilan TabSheet NetConfig

Tata Cara Penggunaan Perangkat Lunak

Untuk menjalankan perangkat lunak ini, pertama kali pengguna harus memilih parameter yang akan digunakan oleh perangkat lunak, yaitu ukuran *mask* dan jumlah *layer* pada jaringan saraf. Langkah berikutnya adalah proses pelatihan untuk mendapatkan nilai bobot yang sesuai dengan karakteristik dari tekstur-tekstur yang digunakan, dengan cara memberikan masukan berupa file-file bitmap (*.bmp) yang berisi citra pelatihan. Langkah tersebut dilakukan oleh pengguna dengan cara memilih *tabsheet Training* dan dilanjutkan dengan memilih *speedbutton Open Image*. Setelah kelima citra pelatihan yang

digunakan telah dibuka, selanjutnya pengguna dapat memilih *speedbutton* **Training** untuk memulai proses pelatihan. Pada saat proses pelatihan berlangsung, pengguna dapat melihat grafik konvergenitas dengan cara memilih *speedbutton* **Show Graphic of Training**. Proses pelatihan ini dilakukan untuk setiap parameter dari jaringan saraf (setiap pilihan pada menu **Parameter**).



Gambar L6. Tampilan untuk Membaca Konvergenitas dari Proses Pelatihan

Setelah tahapan-tahapan di atas dilakukan, berikutnya dijalankan tahapan proses segmentasi, secara berturut-turut pengguna memilih *tabsheet* **Segmentation** dan dilanjutkan dengan memilih *speedbutton* **Open Image** untuk memberikan masukan berupa file bitmap (*.bmp) yang berisi sebuah citra yang terdiri dari beberapa tekstur. Selanjutnya pengguna dapat memilih *speedbutton* **Segmentation** untuk memulai proses segmentasi. Setelah proses segmentasi selesai, pengguna dapat menyimpan hasil segmentasi dengan cara memilih

speedbutton Save Segmented Image. Kemudian pengguna dapat memilih *speedbutton Segmented Image with Median Filter* untuk melihat citra hasil proses segmentasi setelah difilter dengan filter median.

B. NILAI KESALAHAN DARI METODA PELATIHAN DENGAN 2 LAYER

Jumlah Langkah Pelatihan	Mean Square Error		
	Mask 11 x 11	Mask 15 x 15	Mask 21 x 21
1	1.143882225	1.036575038	1.020626641
2	0.676316254	0.470279699	0.452347591
3	0.4983147	0.296562055	0.302715632
4	0.388183236	0.239361996	0.205935947
5	0.333909633	0.195908585	0.176852945
6	0.280874429	0.171030184	0.152448267
7	0.25101938	0.157632433	0.124946136
8	0.224412286	0.147971958	0.105691736
9	0.212246777	0.129480913	0.102873223
10	0.209653796	0.121372201	0.095995493
11	0.187208105	0.108331618	0.083450469
12	0.177022005	0.105639828	0.072620165
13	0.168808432	0.096174569	0.069591759
14	0.163288371	0.097259082	0.069480044
15	0.164285845	0.092816095	0.060799286
16	0.143478371	0.085811022	0.065369151
17	0.132715421	0.080749314	0.053722265
18	0.130205735	0.078902529	0.052280488
19	0.124163029	0.075231301	0.053135716
20	0.122921794	0.07453295	0.050542175
21	0.135543315	0.075327967	0.04791834
22	0.124512679	0.067606643	0.046250687
23	0.117504108	0.069084287	0.041011532
24	0.12964404	0.065968624	0.038292763
25	0.117695639	0.06546759	0.044582506
26	0.11204578	0.06062412	0.035404168
27	0.116636465	0.058067828	0.04042882
28	0.100213153	0.062803165	0.036716467
29	0.102865615	0.053716471	0.038548539
30	0.103130055	0.053628106	0.0321043
31	0.099226885	0.056870555	0.037619467
32	0.111978606	0.051866513	0.036121778
33	0.095495092	0.051582012	0.032121176
34	0.098971349	0.046935567	0.036103693
35	0.091063886	0.049653835	0.031920684
36	0.093609746	0.047320581	0.033533139

37	0.093770748	0.05167054	0.028695917
38	0.091432476	0.046958472	0.030507381
39	0.094512892	0.046116653	0.031463109
40	0.090440777	0.048365649	0.026920165
41	0.087941515	0.040014967	0.026778413
42	0.092508512	0.042711831	0.029468606
43	0.086330657	0.044602131	0.027779651
44	0.094613996	0.044745088	0.028719159
45	0.083511885	0.042676528	0.024505865
46	0.089040233	0.043703827	0.030287007
47	0.082015665	0.044497908	0.024687922
48	0.081319119	0.04389125	0.023583374
49	0.087488742	0.043134889	0.027697686
50	0.080517325	0.040777096	0.025045153
51	0.083820853	0.034658522	0.023769336
52	0.073693468	0.03686167	0.020270638
53	0.076463938	0.036992892	0.02171557
54	0.079619487	0.036250228	0.026396642
55	0.082600503	0.04104345	0.020378952
56	0.073377529	0.043954707	0.019897256
57	0.078359044	0.034848982	0.020880623
58	0.083021904	0.039064694	
59	0.078152792	0.035477523	
60	0.071408574	0.037853813	
61	0.080656543	0.036061109	
62	0.076678984	0.034161802	
63	0.073914059	0.039068781	
64	0.06706204	0.031504242	
65	0.071152182	0.030454344	
66	0.072881531	0.039768544	
67	0.071538482	0.030693339	
68	0.073122767	0.034521748	
69	0.078723166	0.031705701	
70	0.070459222	0.03582285	
71	0.065611083	0.027971858	
72	0.070745177	0.032673204	
73	0.064962824		
74	0.072251669		
75	0.069263017		
76	0.072237711		
77	0.072259905		
78	0.06938055		
79	0.06081955		
80	0.064480739		
81	0.068096788		
82	0.071123278		
83	0.075423544		
84	0.058848384		

85	0.068602634	
86	0.069977441	
87	0.063650702	
88	0.062950122	
89	0.061336442	
90	0.070023727	
91	0.061026634	
92	0.061729175	
93	0.061345815	
94	0.058927126	
95	0.064608758	
96	0.062167552	
97	0.060641786	
98	0.058539294	
99	0.064227967	
100	0.057221326	
101	0.056145335	
102	0.0651356	
103	0.059450395	
104	0.061297153	
105	0.066817023	
106	0.054693988	
107	0.061512125	
108	0.058776609	
109	0.064452214	
110	0.060107317	
111	0.056509408	
112	0.052652082	
113	0.060922312	
114	0.053780553	

C. NILAI KESALAHAN DARI METODA PELATIHAN DENGAN 3 LAYER

Jumlah Langkah Pelatihan	Mean Square Error		
	Mask 11 x 11	Mask 15 x 15	Mask 21 x 21
1	1.564167161	1.592564516	1.563723292
2	1.098109746	1.050194702	1.042598343
3	0.685854973	0.610526585	0.674029156
4	0.44149724	0.352107211	0.36980483
5	0.307890169	0.223015544	0.194301079
6	0.246055499	0.182001677	0.120148633
7	0.230573407	0.150792138	0.092049806
8	0.200430207	0.114477	0.067706123
9	0.181432359	0.113328241	0.05777562
10	0.169075908	0.097401651	0.048874189