

**APLIKASI PENGOLAHAN CITRA DIGITAL
UNTUK MEMILAH BIJI KACANG KEDELAI
BERBASIS MATLAB**

SKRIPSI



Disusun Oleh :
CLARA WILDEA MARTHA
NIM. 09.12.506

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
2013**

LEMBAR PERSETUJUAN

APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK
MEMILAH BIJI KACANG KEDELAI BERBASIS MATLAB

SKRIPSI

*Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan
guna mencapai gelar Sarjana Teknik*

Disusun oleh :

CLARA WILDEA MARTHA

NIM : 09.12.506

Mengetahui,

Ketua Program Studi Teknik Elektro S-1



M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Mengetahui
Pembimbing I

Mengetahui
Pembimbing II

Dr. Eng. Aryuanto Soetedjo, ST, MT
NIP.Y.1030800417

Yuli Wahyuni, ST, MT
NIP.P.1031200456

PRODI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013



SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Clara Wildea Martha

NIM : 09.12.506

Program Studi : Teknik Elektro S-1

Konsentrasi : Teknik Komputer

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 13 September 2013

Yang membuat Pernyataan,



Clara Wildea Martha

NIM : 09.12.506

ABSTRAK

APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK MEMILAH BIJI KACANG KEDELAI BERBASIS MATLAB

Clara Wildea Martha NIM 09.12.506

Dosen Pembimbing: Dr. Eng. Aryuanto Soetedjo, ST, MT dan
Yuli Wahyuni, ST, MT

Ketersediaan kacang kedelai produksi lokal dirasa kurang dibandingkan dengan ketersediaan kacang kedelai impor, ini disebabkan oleh tingginya harga jual kacang kedelai lokal dibandingkan dengan kacang kedelai impor. Salah satu upaya yang dapat dilakukan untuk meningkatkan ketersediaan kacang kedelai dan juga untuk mengurangi ketergantungan terhadap kacang kedelai impor adalah dengan penanganan pascapanen yang baik sehingga kehilangan hasil selama kegiatan pascapanen dapat ditekan. Namun, untuk pemilahan bibit kacang kedelai selama ini, sebagian petani masih menggunakan cara manual yaitu pemilihan butir per butir, jelas ini merupakan cara yang tidak efektif dan tidak efisien.

Morphing merupakan suatu teknik dari pengolahan citra yang didasarkan pada pengolahan bentuk. Nilai dari tiap piksel citra yang diolah merupakan bentuk perbandingan antara piksel yang bersesuaian dari citra masukan dengan nilai piksel tetangganya. Dengan memilih ukuran dan bentuk dari lingkungan, kita dapat membangun operasi analisis untuk mengolah citra yang dimasukkan agar lebih spesifik. Matlab juga digunakan dalam aplikasi ini untuk menampilkan data-data citra (images). Hal ini disebabkan data citra pada dasarnya berupa matriks warna dengan derajat keabuan (grayscale) tertentu.

Dengan dibuatnya aplikasi yang mempermudah proses pemilahan biji kacang kedelai berbasis matlab ini maka cara memilah biji kacang kedelai tanpa harus merusak sampel biji kacang kedelai tersebut dapat dilakukan. Dimana aplikasi ini merupakan pengembangan bagian sortasi tanpa menyentuh dan merusak objeknya.

Kata kunci : Kacang kedelai, Matlab, Pengolahan Citra Digital, Pemilahan.

KATA PENGANTAR

Dengan mengucapkan syukur kehadirat Tuhan Yang Maha Esa yang dengan segala Kasih dan Anugerah-Nya, telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi dengan judul:

"APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK MEMILAIH BIJI KACANG KEDELAI BERBASIS MATLAB"

Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata I di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materiil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Ir. Soeparno Djiwo, MT. selaku rektor ITN Malang
2. Bapak Ir. H. Anang subardi, MT. selaku Dekan Fakultas Teknologi Industri.
3. Bapak M. Ibrahim Ashari, ST, MT. selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
4. Bapak Dr. Eng. Aryuanto Soetedjo, ST, MT. selaku Sekretaris Jurusan Teknik Elektro S-1 ITN Malang.
5. Bapak Ir. Yusuf Ismail Nakhoda, MT. Selaku Dosen Wali.
6. Bapak Dr. Eng. Aryuanto Soetedjo, ST, MT. selaku Dosen Pembimbing I.
7. Ibu Yuli Wahyuni, ST, MT. selaku Dosen Pembimbing II.
8. Bapak Irawan Iswandi dan Ibu Widiastuti, orangtua saya yang selalu mendukung dalam segi apapun, terimakasih banyak saya ucapkan. Kalian berdua keren sekali, I love you.
9. Rizky Mertanila SE, kakak perempuan saya yang tidak kenal lelah menyemangati saya dalam pengerjaan skripsi ini.
10. Made Agus Anggasana Putra ST, terimakasih banyak untuk kesabaran dan pengertiannya.
11. Teman-teman seperjuangan 2009, Ila', Cebong, Vikong, Mas Aji, Apip, Yoga dan semuanya yang selalu berbagi suka dan duka.
12. Mimot, Momo, dan Pona sebagai tempat curhat dan berbagi keceriaan.

13. Dan semua pihak yang telah membantu dalam menyelesaikan skripsi ini yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan. Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, Juli 2013

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAK	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	viii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	3
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	
2.1 Kacang Kedelai	5
2.2 Citra Digital	9
2.2.1 Komponen Citra Digital	9
2.2.2 Representasi Citra Digital	11
2.2.3 Jenis-jenis Citra Digital	12
2.2.4 Elemen-elemen Citra Digital	13
2.2.5 Format Citra Digital	15
2.2.6 Representasi Warna	16
2.3 Edge Detection	17
2.3.1 Metode Robert	20
2.3.2 Metode Prewitt	20
2.3.3 Metode Sobel	20
2.4 MATLAB	21
2.5 Morphological Operation	24
2.5.1 Morphological Operation pada Binary Image dengan Matlab	25
2.5.2 Metode Morphological Image Processing	27

BAB III DESAIN SISTEM

3.1 Tujuan Pembuatan Sistem	32
3.2 Perancangan Sistem	33
3.3 Desain Sistem	33
3.4 Model Pengembangan	34
3.5 Perancangan Sistem Aplikasi	34
3.6 Diagram Alir	35
3.7 Proses Pengolahan Citra	35
3.7.1 Citra Awal	36
3.7.2 Mengubah Citra Awal Menjadi Citra Biner	37
3.7.3 Deteksi dan Klasifikasi Citra	39
3.8 Perancangan Interface	41
3.8.1 Tampilan Form Awal	41
3.8.2 Tampilan Form Kedua	41
3.8.3 Tampilan Form Ketiga	42
3.8.4 Tampilan Form Keempat	43

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem	44
4.2 Implementasi GUI	44
4.2.1 Form Awal Aplikasi	44
4.2.2 Form Kedua	45
4.2.3 Form Ketiga	45
4.2.4 Tampilan Form Keempat	46
4.2.5 Tampilan Form Kelima	46
4.2.6 Tampilan Form Keenam	47
4.2.7 Pengujian Citra Pertama	48
4.2.8 Pengujian Citra Kedua	49
4.2.9 Pengujian Citra Ketiga	50
4.2.10 Pengujian Citra Keempat	51
4.2.11 Pengujian Citra Kelima	52
4.2.12 Pengujian Citra Keenam	53
4.2.13 Pengujian Citra Ketujuh	54

4.2.14 Pengujian Citra Kedelapan	55
BAB V PENUTUP	
5.1 Kesimpulan	56
5.2 Saran	56
DAFTAR PUSTAKA	57
LAMPIRAN	

BAB I PENDAHULUAN

1.1 Latar Belakang

Produksi kacang kedelai di Indonesia selama beberapa tahun terakhir mengalami perubahan fase, yakni dari produktivitas kedelai yang semula 1,38 kuintal per hektar atau menurun 11,07 persen dibandingkan tahun 2011 (BPS Provisi Banten, Maret 2013). Hal ini membuat Pemerintah memberikan kebijakan untuk mengimpor kacang kedelai demi memenuhi kebutuhan konsumen. Sementara pada kasus impor kacang kedelai, harga impor hampir pasti selalu di bawah harga jual dari produksi kacang kedelai di dalam negeri. Perbedaan harga jual tersebut disebabkan oleh produktivitas kacang kedelai impor yang jauh lebih tinggi, kacang kedelai impor ditanam dalam skala usaha yang besar sehingga bersifat lebih ekonomis, serta adanya keberpihakan sangat tinggi dari pemerintah produsen utama kedelai di tingkat dunia kepada para petaninya.

Salah satu upaya yang dapat dilakukan untuk meningkatkan ketersediaan kacang kedelai dan juga untuk mengurangi ketergantungan terhadap kacang kedelai impor adalah dengan penanganan pascapanen yang baik sehingga kehilangan hasil selama kegiatan pascapanen dapat ditekan.

Untuk pemilahan bibit unggul selama ini, sebagian petani masih menggunakan cara manual yaitu pemilihan butir per butir. Bahkan tidak hanya para petani yang masih menggunakan cara manual, tetapi kegiatan pemilihan manual ini juga masih dilakukan oleh Balai Penelitian Tanaman kacang-kacangan dan Umbi-umbian (BALITKABI) Malang yang bertaraf nasional di Jawa Timur. jelas ini merupakan cara yang tidak efektif dan tidak efisien.

Pemilahan biji kacang kedelai masih dilakukan secara manual melalui pengamatan visual ini mempunyai beberapa kelemahan, antara lain membutuhkan waktu lama dan menghasilkan produk dengan mutu yang tidak konsisten karena keterbatasan visual manusia, kelelahan dan adanya perbedaan persepsi tentang mutu pada masing-masing pengamat. Pengolahan citra merupakan alternatif untuk mengatasi persoalan tersebut. Teknik pengolahan citra bisa mempermudah pekerjaan pemilahan sehingga dapat memberikan hasil yang diharapkan.

Oleh karena itu, untuk mempermudah proses pemilahan biji kacang kedelai maka dibuatlah suatu aplikasi untuk memilah biji kacang kedelai berbasis matlab yang diharapkan akan bermanfaat bagi pemilahan biji kacang kedelai yang cepat tanpa harus merusak sampel biji kacang kedelai tersebut, merupakan pengembangan bidang sortasi tanpa menyentuh dan merusak objeknya.

1.2 Rumusan Masalah

Dari latar belakang yang telah dijelaskan, maka masalah yang akan dibahas adalah mengenai : Bagaimana membuat aplikasi citra tersebut sehingga dapat memilah biji kacang kedelai yang dapat diimplementasikan pada pertanian?

1.3 Tujuan Penulisan

Tujuan dari penulisan makalah ini adalah untuk membuat suatu program aplikasi citra tersebut sehingga dapat memilah biji kacang kedelai diimplementasikan pada pertanian, dimana biji kacang kedelai akan dipilah menjadi dua bagian yaitu biji baik dan buruk.

1.4 Batasan Masalah

Batasan – batasan masalah dalam makalah ini adalah:

1. Masukan sistem adalah citra dalam aras RGB (*Red, Green, Blue*). Tanpa membahas proses pengambilan, pemotretan dan pemrosesan citra sebelum digunakan.
2. Format citra asli merupakan citra berwarna 24-bit dalam format JPEG (ekstensi *.jpg).
3. Pemilahan berdasarkan bentuk yang nampak pada citra.
4. Kacang kedelai yang digunakan adalah jenis kacang kedelai lokal.
5. Hanya membahas program aplikasi menggunakan pengolahan citra digital.

1.5 Metode Penelitian

Metode yang digunakan dalam penyusunan skripsi ini adalah :

1) Studi literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

2) Analisa Kebutuhan Sistem

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem baik hardware maupun software, di mana nantinya akan digunakan sebagai acuan perancangan sistem.

3) Perancangan sistem

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistem ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat dan diimplementasikan ke dalam sistem.

4) Eksperimen dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

1.6. Sistematika Pembahasan

Penyusunan laporan skripsi ini menggunakan kerangka pembahasan yang terbentuk dalam susunan bab, yang dapat dijelaskan sebagai berikut :

Bab I : Pendahuluan

Bab ini merupakan dasar penyusunan skripsi yang di dalamnya berisi tentang latar belakang masalah, rumusan masalah, tujuan skripsi, batasan masalah, metodologi pengembangan sistem, dan sistematika pembahasan skripsi.

Bab II : Dasar Teori

Bab ini berisi tentang permasalahan yang berhubungan dengan penelitian yang dilakukan pada skripsi ini.

Bab III : Desain Sistem

Berisi perancangan aplikasi sesuai dengan data yang didapatkan.

Bab IV : Implementasi dan Pengujian

Bab ini berisi tentang tahap implementasi yaitu identifikasi untuk menyelesaikan masalah yang dihadapi berdasarkan teori pada Bab II dan Bab III. Bab ini juga berisi hasil pengujian sistem.

Bab V : Penutup

Bab ini berisi tentang kesimpulan dan saran dari hasil penyusunan skripsi yang telah disusun.

BAB II

TINJAUAN PUSTAKA

2.1 Kacang Kedelai

Kedelai merupakan tanaman pangan berupa semak yang tumbuh tegak. Kedelai jenis liar *Glycine ururiensis*, merupakan kedelai yang menurunkan berbagai kedelai yang dikenal sekarang kedelai (*Glycine max* (L.) Merrill). Berasal dari daerah Manshukuo (Cina Utara), di Indonesia, dibudidayakan mulai abad ke-17 sebagai tanaman makanan dan pupuk hijau. Penyebaran tanaman kedelai ke Indonesia berasal dari daerah Manshukuo menyebar ke daerah Mansyuria : Jepang (Asia Timur) dan negara-negara lain di Amerika dan Afrika. (AAK,1989).

Akar tanaman kedelai terdiri atas akar tunggang, akar lateral, dan akar serabut. Pada tanah yang gembur, akar ini dapat menembus tanah sampai kedalaman 1,5 m. Pada akar lateral terdapat bintil-bintil akar yang merupakan kumpulan bakteri rhizobium pengikat N dari udara. Bintil akar ini biasanya akan terbentuk 15-20 hari setelah tanam, selain sebagai penyerap unsur hara dan penyangga tanaman, pada perakaran merupakan tempat terbentuknya bintil/nodul akar yang berfungsi sebagai pabrik alami terfiksasinya nitrogen udara oleh aktivitas bakteri Rhizobium (Tambas dan Rakhman, 1986).

Kedelai berbatang semak, dengan tinggi batang antara 30-100 cm. setiap batang dapat membentuk 3-6 cabang. Pertumbuhan batang dibedakan menjadi dua tipe, yaitu tipe determinate dan indeterminate. Perbedaan system pertumbuhan batang ini didasarkan atas keberadaan bunga dan pucuk batang. Pertumbuhan batang tipe determinate ditunjukkan dengan batang yang tidak tumbuh lagi pada saat tanaman mulai berbunga. Pertumbuhan batang tipe indeterminate dicirikan bila pucuk batang tanaman masih bisa tumbuh daun, walaupun tanaman sudah mulai berbunga. Begitu juga dengan bentuk daun kedelai ada dua macam, yaitu bulat (oval) dan lancip (lanceolate) (Adisarwanto, 2005).

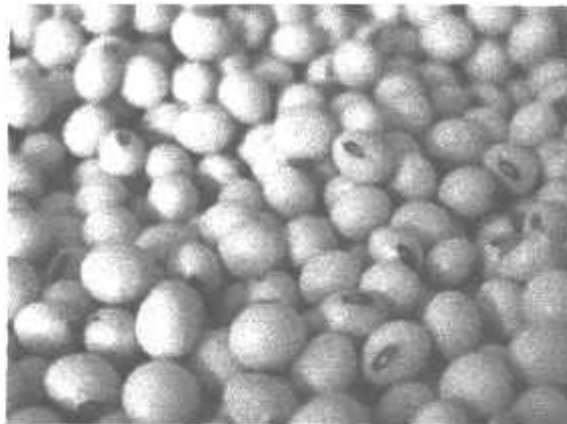
Bunga kedelai termasuk bunga sempurna, artinya dalam setiap bunga terdapat alat kelamin jantan dan alat kelamin betina. Penyerbukan terjadi pada saat mahkota bunga masih menutup, sehingga kemungkinan terjadinya kawin silang secara alami sangat kecil. Bunga terletak pada ruas-ruas batang, berwarna ungu atau putih. Tidak semua bunga dapat menjadi polong walaupun telah terjadi penyerbukan secara sempurna (Suprpto, 2001).

Polong kedelai pertama terbentuk sekitar 7-10 hari setelah munculnya bunga pertama. Panjang polong muda sekitar 1 cm, jumlah polong yang terbentuk pada setiap ketiak tangkai daun sangat beragam, antara 1-10 buah dalam setiap kelompok. Pada setiap tanaman, jumlah polong dapat mencapai lebih dari 50, bahkan ratusan. Kecepatan pembentukan polong dan pembesaran biji akan semakin cepat setelah proses pembentukan bunga berhenti. Ukuran dan bentuk polong menjadi maksimal pada saat awal periode pemasakan biji. Hal ini kemungkinan diikuti oleh perubahan warna polong, dari hijau menjadi kuning kecoklatan pada saat masak (Adisarwanto, 2005). Biji kedelai berkeping dua yang terbungkus oleh kulit biji. Embrio terletak di antara keping biji. Warna kulit biji bermacam-macam, ada yang kuning, hitam, hijau atau coklat.

Kedelai yang tumbuh secara liar di Asia Tenggara meliputi sekitar 40 jenis. Penyebaran geografis dari kedelai mempengaruhi jenis tipenya. Terdapat 4 tipe kedelai yakni : tipe Mansyuria, Jepang, India, dan Cina. Dasar-dasar penentuan varietas kedelai adalah menurut: umur, warna biji dan tipe batang

Di salah satu negara bagian Amerika Serikat, terdapat areal pertumbuhan kedelai yang sangat luas sehingga menghasilkan 57% produksi kedelai dunia. Di Indonesia, saat ini kedelai banyak ditanam di dataran rendah yang tidak banyak mengandung air, seperti di pesisir Utara Jawa Timur, Jawa Tengah, Jawa Barat, Sulawesi Utara (Gorontalo), Lampung, Sumatera Selatan dan Bali. Kedelai (*Glycine max* (L) merrill) merupakan salah satu tanaman budidaya dengan kandungan nutrisi yang tinggi, diantaranya mengandung protein 30-50% (Richard et al., 1984). Kandungan

protein yang tinggi memberi indikasi bahwa tanaman kedelai memerlukan hara nitrogen yang tinggi pula. .



Gambar 2.1
Kacang kedelai

Menurut Kalshoven (2006), klasifikasi kacang kedelai (*Glycine max* (L.) adalah sebagai berikut:

- Divisi : Magnoliophyta
- Kelas : Magnoliopsida
- Ordo : Fabales
- Famili : Fabaceae
- Genus : *Glycine*
- Spesies : *Glycine max* (L.)

Kacang kedelai terkenal dengan nilai gizinya dan merupakan salah satu makanan yang mengandung 8 asam amino yang penting dan dibutuhkan oleh tubuh manusia. Tidak seperti makanan lain yang mengandung lemak jenuh dan tidak dapat dicerna, kacang kedelai tidak mengandung kolesterol, mempunyai rasio kalori rendah dibandingkan dengan protein dan bertindak sebagai makanan yang tidak menggemukkan bagi penderita obesitas. Kacang kedelai juga mengandung kalsium, besi, potassium, dan phosphorus. Kedelai juga kaya akan vitamin B kompleks yang baik bagi kesehatan manusia.

Adapun persyaratan yang perlu dipenuhi untuk mendukung pertumbuhan kacang kedelai antara lain :

- Iklim

Kedelai merupakan tanaman yang menyukai wilayah tropis pun subtropics. Ia sama seperti tanaman jagung. Bahkan dalam kondisi tertentu, ketahanan kacang kedelai jauh lebih tinggi ketimbang jagung. Kedelai menyukai tanah yang kering ketimbang lembab. Selain itu, kedelai juga tumbuh optimal di tempat dengan tingkat curah hujan 100 sampai 400 mm per bulannya. Sementara itu suhu yang paling baik adalah 21 sampai 34 derajat Celsius.

- Medium Tanam

Kedelai menyukai tanah yang tidak lembab namun cukup air. Kabar baiknya, tanaman ini tidak "rewel" soal jenis tanah yang cocok. Ia bisa tumbuh bahkan di lahan yang kurang subur sekalipun. Yang Paling penting adalah tanah tersebut tidak lembab dan air tidak menggenang. Meski demikian, untuk hasil paling optimal Anda bisa menanam kedelai di tanah jenis alluvial, regosol, grumoso, lantosoil juga andosol. Tingkat keasaman tanah sebaiknya berkisar di angka 5,8 sampai 7,0. Apabila kurang dari 5,5 maka pertumbuhan kacang kedelai akan lambat. Dalam budidaya kacang kedelai, sebaiknya Anda memilih medium tanam yang datar. Kalaupun tanahnya landai, Anda harus membutuhkan terasering.

- Ketinggian Tempat

Kacang kedelai dengan ukuran kecil sangat baik ditanam di lahan pada ketinggian 0,5 sampai 300 meter di atas permukaan laut. Sementara itu, kacang kedelai dengan ukuran biji lebih besar jauh lebih baik ditanam di ketinggian mulai dari 300 sampai 500 meter di atas permukaan laut.

Untuk hasil panen yang lebih baik, bibit yang digunakan dalam budidaya kacang kedelai sebaiknya bibit yang berkualitas. harus memenuhi unsur-unsur seperti:

1. Memiliki daya tumbuh yang jauh lebih baik.
2. Tidak tercemar dengan varietas lainnya.
3. Bersih dari kotoran apapun.
4. Memiliki ukuran yang seragam.
5. Merupakan varietas yang unggul dengan tingkatan produksi yang tinggi.

6. Dikenal kuat terhadap serangan hama juga penyakit.

Setelah diperoleh benih yang unggul, langkah selanjutnya adalah menyiapkan benih atau bibit. Sebelum ditanam, bibit kacang kedelai harus dicampur dengan legin terlebih dahulu. Legin adalah inokulum buatan bakteri. Bakteri ini akan hidup di bagian bintil akar dan sangat baik membantu kedelai mengikat unsur N dari udara kelak. Setelah pemberian legin, benih kacang kedelai disemaikan beberapa waktu. Setelah itu baru dipindahkan ke lahan tanam yang sebenarnya.

Setelah ditanam, sama seperti tanaman lainnya, dalam budidaya kacang kedelai diperlukan proses pemeliharaan yang meliputi penjarangan juga penyulaman, penyiangan, pembumbunan, pemupukan, pengairan juga penyiraman, pemberian pupuk, penyemprotan pestisida, dan penentasan hama juga penyakit jika ia menyerang kacang kedelai. Adapun hama yang sering menghantui para petani kacang kedelai adalah Aphis SPP, Melano Agromyza Phaseoli, Kumbang Daun tembukur, Cantalan, Ulat Polong, Kepala Polong, Kepik Hijau atau Nezara Viridula, juga ulat grayak. Sementara itu penyakit yang sering menyerang adalah penyakit layu bakteri, penyakit layu akibat jamur tanah, penyakit lapu, penyakit karat, penyakit anthracnose dan masih banyak lagi lainnya.

2.2 Citra Digital

2.2.1 Komponen Citra Digital

(Teori pengolahan Citra Digital, 2009) Citra adalah representasi dua dimensi untuk bentuk fisik nyata tiga dimensi. Citra dalam perwujudannya dapat bermacam-macam, mulai dari gambar hitam-putih pada sebuah foto (yang tidak bergerak) sampai pada gambar berwarna yang bergerak pada pesawat televisi. Proses transformasi dari bentuk tiga dimensi ke bentuk dua dimensi untuk menghasilkan citra akan dipengaruhi oleh bermacam-macam factor yang mengakibatkan penampilan citra suatu benda tidak sama persis dengan bentuk fisik nyatanya. Faktor-faktor tersebut merupakan efek degradasi atau penurunan kualitas yang dapat berupa rentang kontras benda yang terlalu sempit atau terlalu lebar, distorsi geometrik, kekaburan (*blur*),

kekaburan akibat obyek yang bergerak (*motion blur*), noise atau gangguan yang disebabkan oleh interferensi peralatan pembuat citra, baik berupa transduser, peralatan elektronik ataupun peralatan optik.

Teknik dan proses untuk mengurangi atau menghilangkan efek degradasi pada citra digital meliputi perbaikan/peningkatan citra (*image enhancement*), restorasi citra (*image restoration*), dan transformasi spasial (*spasial transformation*). Subyek lain dari pengolahan citra digital diantaranya adalah pengkodean citra (*image coding*), segmentasi citra (*image segmentation*), representasi dan deskripsi citra (*image representation and description*).

Pengolahan citra dilakukan dengan komputer digital maka citra yang akan diolah terlebih dahulu ditransformasikan ke dalam bentuk besaran-besaran diskrit dari nilai tingkat keabuan pada titik-titik elemen citra. Bentuk citra ini disebut citra digital. Setiap citra digital memiliki beberapa karakteristik, antara lain ukuran citra, resolusi dan format lainnya. Umumnya citra digital berbentuk persegi panjang yang memiliki lebar dan tinggi tertentu, yang biasanya dinyatakan dalam banyaknya titik atau piksel (*picture elemen/pixel*).

Ukuran citra dapat juga dinyatakan secara fisik dalam satuan panjang (misalnya mm atau inch). Dalam hal ini tentu saja harus ada hubungan antara ukuran titik penyusun citra dengan satuan panjang. Hal tersebut dinyatakan dengan resolusi yang merupakan ukuran banyaknya titik untuk setiap satuan panjang. Biasanya satuan yang digunakan adalah dpi (*dot per inch*). Makin besar resolusi makin banyak titik yang terkandung dalam citra dengan ukuran fisik yang sama. Hal ini memberikan efek penampakan citra menjadi semakin halus.

Format citra digital ada bermacam-macam. Karena sebenarnya citra merepresentasikan informasi tertentu, sedangkan informasi tersebut dapat dinyatakan secara bervariasi, maka citra yang mewakilinya dapat muncul dalam berbagai format. Citra yang merepresentasikan informasi yang hanya bersifat biner untuk membedakan dua keadaan tentu tidak sama citra dengan informasi yang lebih kompleks sehingga memerlukan lebih banyak keadaan

yang diwakilinya. Pada citra digital semua informasi tadi disimpan dalam bentuk angka, sedangkan penampilan angka tersebut biasanya dikaitkan dengan warna.

Citra digital (*digital image*) adalah citra kontinu $f(x,y)$ yang sudah didiskritkan baik koordinat spasial maupun tingkat kecerahannya. Setiap titik biasanya memiliki koordinat sesuai dengan posisinya dalam citra. Koordinat ini biasanya dinyatakan indeks x dan y hanya bernilai bilangan bulat positif, yang dapat dimulai dari 0 atau 1. Citra digital yang selanjutnya akan disingkat "citra" sebagai matrik ukuran $M \times N$ yang baris dan kolomnya menunjukkan titik-titiknya yang diperlihatkan pada persamaan 1 (Munir, R., 2004).

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \dots\dots\dots(2-1)$$

Setiap titik juga memiliki nilai berupa angka digital yang merepresentasikan informasi yang diwakili titik tersebut. Format nilai piksel sama dengan format citra keseluruhan. Pada kebanyakan sistem pencitraan, nilai ini biasanya berupa bilangan bulat positif.

2.2.2 Representasi Citra Digital

Komputer dapat mengolah isyarat-isyarat elektronik digital yang merupakan kumpulan sinyal biner (bernilai dua: 0 dan 1). Untuk itu, citra digital harus mempunyai format tertentu yang sesuai sehingga dapat merepresentasikan obyek pencitraan dalam bentuk kombinasi data biner.

Citra yang tidak berwarna atau hitam putih dikenal sebagai citra dengan derajat abu-abu (*citra graylevel/grayscale*). Derajat abu-abu yang dimiliki ini bisa beragam mulai dari 2 derajat abu-abu (yaitu 0 dan 1) yang dikenal juga sebagai citra monochrome, 16 derajat keabuan dan 256 derajat keabuan.

Dalam sebuah citra monochrome, sebuah piksel diwakili oleh 1 bit data yang berisikan data tentang derajat keabuan yang dimiliki piksel tersebut.

Data akan berisi 0 bila piksel berwarna hitam dan 1 bila piksel berwarna putih. Citra yang memiliki 16 derajat keabuan (mulai dari 0 yang mewakili warna hitam sampai dengan 15 yang mewakili warna putih) direpresentasikan oleh 4 bit data. Sedangkan citra dengan 256 derajat keabuan (nilai dari 0 yang mewakili warna hitam sampai dengan 255 yang mewakili warna putih) direpresentasikan oleh 8 bit data.

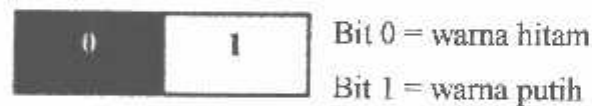
Dalam citra berwarna, jumlah warna bisa beragam mulai dari 16, 256, 65536 atau 16 juta warna yang masing-masing direpresentasikan oleh 4,8,16 atau 24 bit data untuk setiap pikselnya. Warna yang ada terdiri dari 3 komponen utama yaitu nilai merah (*red*), nilai hijau (*green*) dan nilai biru (*blue*).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

2.2.3 Jenis-jenis Citra Digital

2.2.3.1 Citra Biner (Monokrom)

Banyaknya warna: 2, yaitu hitam dan putih. Dibutuhkan 1 bit dalam memori untuk menyimpan kedua warna ini.

Gradasi warna:



Gambar 2.2
Citra Biner

2.2.3.2 Citra *Grayscale* (Skala Keabuan)

Banyaknya warna: tergantung pada jumlah bit yang disediakan dalam memori untuk menampung kebutuhan warna ini.

Citra 2 bit mewakili 4 warna dengan gradasi warna sebagai berikut:



Gambar 2.3
Citra *Grayscale* 2 bit

Citra 3 bit mewakili 8 warna dengan gradasi warna sebagai berikut:



Gambar 2.4
Citra *Grayscale* 3 bit

2.2.3.3 Citra Warna (*True Color*)

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = *Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = byte, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak $255 \times 255 \times 255 = 16.777.216$ = 16 juta warna lebih. Itulah sebabnya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna alam.

2.2.4 Elemen-elemen Citra Digital

Berikut ini adalah elemen-elemen yang terdapat pada citra digital.

1. Kecerahan (*Brightness*)

Kecerahan (*Brightness*) merupakan intensitas cahaya yang dipancarkan piksel dari citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah titik (piksel) di dalam citra merupakan intensitas rata-rata dari suatu area yang melingkupinya.

2. Kontras (*Contrast*)

Kontras (*Contrast*) menyatakan sebaran terang dan gelap dalam sebuah citra. Pada citra yang baik komposisi gelap dan terang tersebar secara merata.

3. Kontur (*Contour*)

Kontur (*Contour*) adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel-piksel yang bertetangga. Karena adanya perubahan intensitas inilah mata mampu mendeteksi tepi-tepi objek di dalam citra.

4. Warna

Warna sebagai persepsi yang ditangkap sistem visual terhadap panjang gelombang cahaya yang dipantulkan oleh objek.

5. Bentuk (*Shape*)

Bentuk (*Shape*) adalah properti intrinsik dari objek 3 dimensi, dengan pengertian bahwa bentuk merupakan properti intrinsik utama untuk sistem visual manusia.

6. Tekstur (*Texture*)

Tekstur (*Texture*) dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan piksel-piksel yang bertetangga. Tekstur adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut. Tekstur adalah keteraturan pola-pola tertentu yang terbentuk dari susunan piksel-piksel dalam citra digital. Informasi tekstur dapat digunakan untuk membedakan sifat-sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus, juga sifat-sifat spesifik dari kekasaran dan kehalusan permukaan tadi, yang sama sekali terlepas dari warna permukaan tersebut.

Berikut kegunaan analisis tekstur:

- Tekstur memainkan peranan penting dalam banyak tugas pada sistem visual, seperti pemeriksaan permukaan, pengelompokan objek pemandangan, orientasi permukaan dan penentuan bentuk objek.
- Digunakan untuk segmentasi citra, mengidentifikasi pola-pola yang teratur dan berulang, pola-pola intensitas, permukaan benda yang

berhubungan dengan sifat kasar dan halus, koloni mikroba, jalan raya, bahkan sampai pada sifat permukaan bumi atau planet lainnya.

- Untuk tujuan pengolahan citra, analisis tekstur adalah menjadikan pola variasi lokal intensitas yang berulang sebagai pembeda, manakala pola variasi tersebut terlalu kecil bila dibandingkan dengan objek yang diamati dalam resolusi yang dipakai.

Syarat terbentuknya tekstur adalah sebagai berikut.

- Adanya pola-pola primitif yang terdiri dari satu atau lebih piksel. Bentuk pola primitif ini dapat berupa titik, garis lurus, garis lengkung, luasan dan lain-lain yang merupakan elemen dasar dari sebuah bentuk.
- Pola-pola primitif tadi muncul berulang-ulang dengan interval jarak dan arah tertentu sehingga dapat diprediksi atau ditemukan karakteristik pengulangannya.

2.2.5 Format Citra Digital

Citra Digital memiliki beberapa format yang memiliki karakteristik tersendiri. Format pada citra digital ini umumnya berdasarkan tipe dan cara kompresi yang digunakan pada citra digital tersebut. Ada dua format citra digital yang sering dijumpai, antara lain:

1. Bitmap (BMP)

Merupakan format gambar yang paling umum dan merupakan format standard windows. Ukuran filenya sangat besar karena bisa mencapai ukuran megabyte. File ini merupakan format yang belum terkompresi dan menggunakan sistem warna RGB (*Red, Green, Blue*) di mana masing-masing warna pixel-nya terdiri dari 3 komponen R, G, dan B yang dicampur menjadi satu. File BMP dapat dibuka dengan berbagai macam software pembuka gambar seperti ACDSee, Paint, Irfan View dan lain-lain. File BMP tidak bisa (sangat jarang) digunakan di web (internet) karena ukurannya yang besar.

2. Joint Photographic Expert Group (JPEG/JPG)

Format JPEG merupakan format yang paling terkenal sampai sekarang ini. Hal ini karena sifatnya yang berukuran kecil (hanya puluhan/ratusan KB saja), dan bersifat portable. Format file ini sering digunakan pada bidang fotografi untuk menyimpan file foto hasil perekaman analog to digital converter (ADC). Karena ukurannya kecil maka file ini banyak digunakan di web (internet).

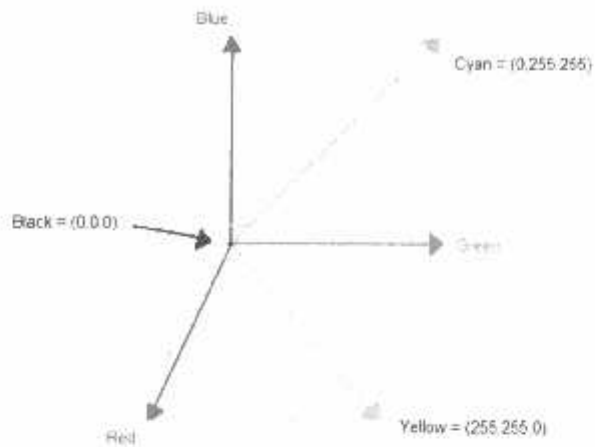
2.2.6 Representasi Warna

(Arie Murdianto, 2007) Metode yang paling mudah dan sederhana dalam merepresentasikan warna di dalam komputer adalah bitmap. Tipe file berformat seperti ini dapat diibaratkan dengan sebuah matriks yang terdiri dari baris-baris piksel. Dan setiap piksel dari baris tersebut merepresentasikan nilai tampilan warna piksel tersebut. Nilai tampilan warna tersebut disusun berdasarkan tiga warna dasar, yaitu: merah, biru, dan hijau. Metode ini dikenal dengan metode *RGB encoding*.

Setiap satu warna dasar (merah, hijau atau biru) memiliki 256 level warna. Sehingga total warna yang dapat dikenali oleh komputer sekarang adalah $256 \times 256 \times 256$ atau 16,8 juta warna yang berbeda. Jumlah tersebut didapatkan dengan mengkombinasikan setiap level warna dasar yang berbeda.

Misalnya saja warna hitam dapat direpresentasikan ke dalam sistem *RGB* sebagai $(0, 0, 0)$ yang berarti level warna merah 0, warna hijau 0, dan warna biru 0. Warna merah dapat direpresentasikan dengan $(255, 0, 0)$ yang berarti level warna merah sebanyak 255, warna hijau sebanyak 0, dan warna biru sebanyak 0. Sedangkan warna putih direpresentasikan ke dalam sistem *RGB* menjadi $(255, 255, 255)$.

Pada kebanyakan pemrosesan gambar membutuhkan properti-properti yang dimiliki oleh vektor, seperti normalisasi, rotasi dan proyeksi. Sehingga representasi warna dapat juga dilakukan dalam bentuk vektor. Berikut ini akan ditampilkan satu contoh vektor warna dalam ruang 3 dimensi.



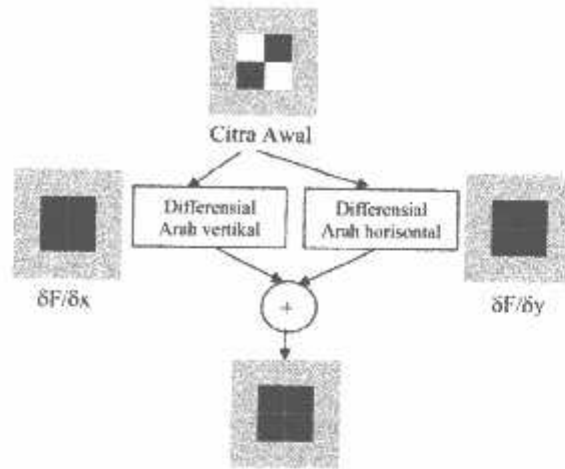
Gambar 2.5
Vektor representasi warna

2.3 *Edge Detection*

(Gonzalez, 2010) Deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah:

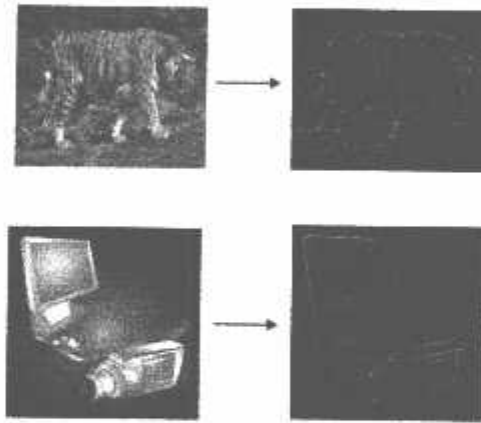
- Untuk menandai bagian yang menjadi detail citra
- Untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra

Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya. Gambar 2.6 berikut ini menggambarkan bagaimana tepi suatu gambar diperoleh.



Gambar 2.6
Proses Deteksi Tepi Citra

Perhatikan hasil deteksi dari beberapa citra menggunakan model differensial di atas:



Gambar 2.7
Hasil beberapa deteksi tepi

Pada gambar 2.7 terlihat bahwa hasil deteksi tepi berupa tepi-tepi dari suatu gambar. Bila diperhatikan bahwa tepi suatu gambar terletak pada titik-titik yang memiliki perbedaan tinggi. Berdasarkan prinsip-prinsip filter pada citra maka tepi suatu gambar dapat diperoleh menggunakan High Pass Filter (HPF), yang mempunyai karakteristik:

$$\sum_x \sum_y H(x, y) = 0 \dots\dots\dots(2-2)$$

Contoh:

Diketahui fungsi citra $f(x,y)$ sebagai berikut:

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

Dengan menggunakan filter : $H(x,y) = [-1,1]$

Maka Hasil filter adalah :

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

Bila digambarkan maka proses filter di atas mempunyai masukan dan keluaran sebagai berikut:



Macam-macam metode untuk proses deteksi tepi ini, antara lain:

1. Metode Robert
2. Metode Prewitt
3. Metode Sobel

Metode yang banyak digunakan untuk proses deteksi tepi adalah metode Robert, Prewitt dan Sobel.

2.3.1 Metode Robert

Metode Robert adalah nama lain dari teknik differensial yang dikembangkan di atas, yaitu differensial pada arah horisontal dan differensial pada arah vertikal, dengan ditambahkan proses konversi biner setelah dilakukan differensial. Teknik konversi biner yang disarankan adalah konversi biner dengan meratakan distribusi warna hitam dan putih. Metode Robert ini juga disamakan dengan teknik DPCM (Differential Pulse Code Modulation).

Kernel filter yang digunakan dalam metode Robert ini adalah:

$$H = \begin{bmatrix} -1 & 1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \dots\dots\dots(2-3)$$

2.3.2 Metode Prewitt

Metode Prewitt merupakan pengembangan metode robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF.

Kernel filter yang digunakan dalam metode Prewitt ini adalah:

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \dots\dots\dots(2-4)$$

2.3.3 Metode Sobel

Metode Sobel merupakan pengembangan metode robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode sobel ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi.

Kernel filter yang digunakan dalam metode Sobel ini adalah:

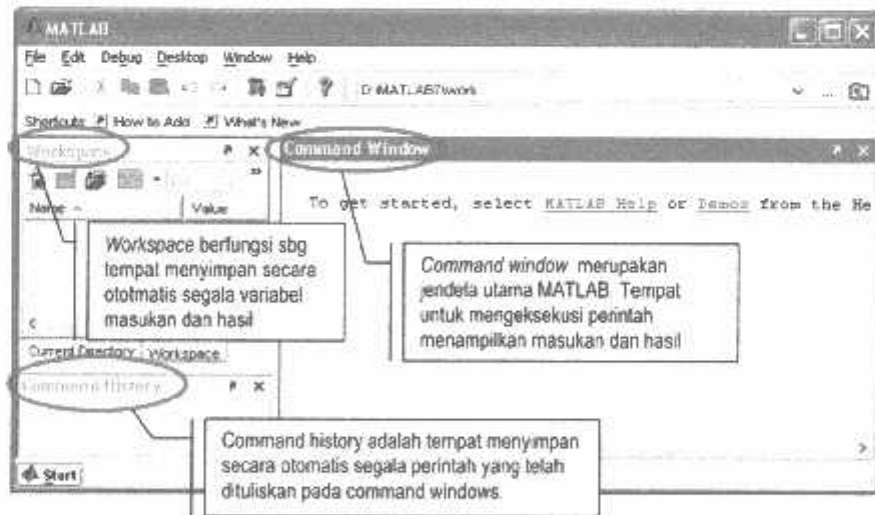
$$H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \dots\dots\dots(2-5)$$

2.4 MATLAB

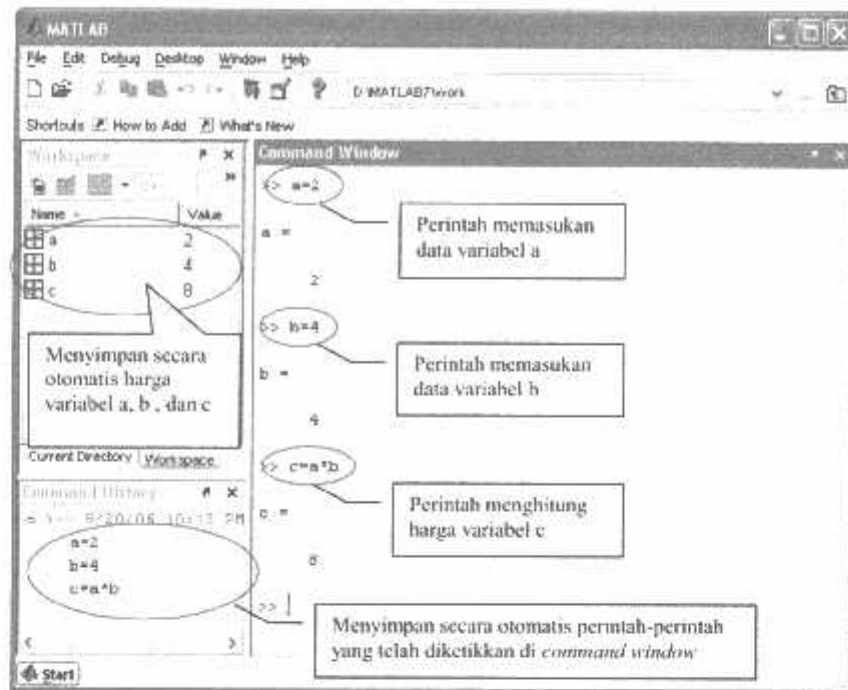
MATLAB merupakan perangkat lunak produk dari The MathWorks, Inc yang memadukan kemampuan perhitungan, pencitraan, dan pemrograman dalam satu paket. MATLAB merupakan bahasa komputasi teknik yang lebih mudah dan lebih canggih dalam penggunaannya dibandingkan dengan bahasa teknik pendahulunya seperti FORTRAN, BASIC, PASCAL. Sebetulnya MATLAB tidaklah berbeda dengan kalkulator scientific yang sehari-hari kita (orang teknik) kenal.

Secara garis besar lingkungan kerja MATLAB terdiri atas beberapa unsur, yaitu:

1. Command window (layar kendali)
2. Workspace (rak data)
3. Command history (layar pengingat)
4. M-file (editor) → akan dibahas pada bagian khusus.



Gambar 2.8
Lingkungan Kerja MATLAB 7.0



Gambar 2.9
Sistem Kerja MATLAB

MATLAB adalah sebuah bahasa dengan (high-performance) kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar. Penggunaan Matlab meliputi bidang-bidang:

- Matematika dan Komputasi
- Pembentukan Algorithm
- Akusisi Data
- Pemodelan, simulasi, dan pembuatan prototipe
- Analisa data, eksplorasi, dan visualisasi
- Grafik Keilmuan dan bidang Rekayasa

MATLAB merupakan suatu sistem interaktif yang memiliki elemen data dalam suatu array sehingga tidak lagi dipusingkan dengan masalah dimensi. Hal ini memungkinkan untuk memecahkan banyak masalah teknis yang terkait dengan komputasi, khususnya yang berhubungan dengan matrix dan formulasi vektor, yang mana masalah tersebut merupakan momok apabila

harus menyelesaikannya dengan menggunakan bahasa level rendah seperti Pascal, C dan Basic.

Nama MATLAB merupakan singkatan dari matrix laboratory. MATLAB pada awalnya ditulis untuk memudahkan akses perangkat lunak matrik yang telah dibentuk oleh LINPACK dan EISPACK. Saat ini perangkat MATLAB telah menggabung dengan LAPACK dan BLAS library, yang merupakan satu kesatuan dari sebuah seni tersendiri dalam perangkat lunak untuk komputasi matrix.

Dalam lingkungan perguruan tinggi teknik, Matlab merupakan perangkat standar untuk memperkenalkan dan mengembangkan penyajian materi matematika, rekayasa dan kelimuan. Di industri, MATLAB merupakan perangkat pilihan untuk penelitian dengan produktifitas yang tinggi, pengembangan dan analisisnya.

Fitur-fitur MATLAB sudah banyak dikembangkan, dan lebih dikenal dengan nama toolbox. Sangat penting bagi seorang pengguna Matlab, toolbox mana yang mendukung untuk learn dan apply teknologi yang sedang dipelajarinya. Toolbox toolbox ini merupakan kumpulan dari fungsi-fungsi MATLAB (M-files) yang telah dikembangkan ke suatu lingkungan kerja MATLAB untuk memecahkan masalah dalam kelas particular. Area-area yang sudah bisa dipecahkan dengan toolbox saat ini meliputi pengolahan sinyal, system kontrol, neural networks, fuzzy logic, wavelets, dan lain-lain.

Kelengkapan pada Sistem MATLAB. Sebagai sebuah system, MATLAB tersusun dari 5 bagian utama:

1. Development Environment. Merupakan sekumpulan perangkat dan fasilitas yang membantu untuk menggunakan fungsi-fungsi dan file-file MATLAB. Beberapa perangkat ini merupakan sebuah graphical user interfaces (GUI). Termasuk didalamnya adalah MATLAB desktop dan Command Window, command history, sebuah editor dan debugger, dan browsers untuk melihat help, workspace, files, dan search path.
2. MATLAB Mathematical Function Library. Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti: sum, sin, cos, dan complex arithmetic, sampai dengan fungsi-fungsi yang lebih kompleks

seperti matrix inverse, matrix eigenvalues, Bessel functions, dan fast Fourier transforms.

3. MATLAB Language. Merupakan suatu high-level matrix/array language dengan control flow statements, functions, data structures, input/output, dan fitur-fitur object-oriented programming. Ini memungkinkan bagi kita untuk melakukan kedua hal baik "pemrograman dalam lingkup sederhana" untuk mendapatkan hasil yang cepat, dan "pemrograman dalam lingkup yang lebih besar" untuk memperoleh hasil-hasil dan aplikasi yang kompleks.
4. Graphics. MATLAB memiliki fasilitas untuk menampilkan vector dan matrices sebagai suatu grafik. Didalamnya melibatkan high-level functions (fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan data tiga dimensi, image processing, animation, dan presentation graphics. Ini juga melibatkan fungsi level rendah yang memungkinkan bagi anda untuk membiasakan diri untuk memunculkan grafik mulai dari bentuk yang sederhana sampai dengan tingkatan graphical user interfaces pada aplikasi MATLAB.
5. MATLAB Application Program Interface (API). Merupakan suatu library yang memungkinkan program yang telah ditulis dalam bahasa C dan Fortran mampu berinteraksi dengan MATLAB. Ini melibatkan fasilitas untuk pemanggilan routines dari MATLAB (dynamic linking), pemanggilan MATLAB sebagai sebuah computational engine, dan untuk membaca dan menuliskan MAT-files.

2.5 Morphological Operation

Telah disebutkan di atas, bahwa Matlab adalah salah satu perangkat lunak yang berbasis matematika, matlab juga merupakan software yang dapat dimanfaatkan untuk image processing (pengolahan citra). Dengan menggunakan matlab, serta fungsi-fungsi yang telah disediakan oleh matlab, maka proses pengolahan citra pada Binary Image dengan menggunakan Morphological Operation dapat diterapkan dengan lebih baik Morphological Operation berisikan operasi-operasi dasar yang berbasis matematika. Operasi-

operasi ini digunakan sebagai dasar dalam Morphological Image Processing pada Binary image. Dengan menggunakan Matlab, proses pengerjaan image processing khususnya untuk Morphological Operation diharapkan dapat lebih mudah.

Kata Morphology biasanya dikaitkan dengan cabang Ilmu Biologi yang mengacu pada bentuk dan struktur dari hewan dan tanaman. Kita menggunakan kata yang sama disini, namun dalam konteks Mathematicat Morphology, yaitu sebuah alat untuk mengekstrak komponen-komponen dari image dengan menggunakan operasi matematika. Dalam Morphological Image ini, sebuah image akan diubah ke dalam bentuk Binary Image. Binary Image adalah image yang termasuk dalam kelas logika, dimana hanya bernilai 0 dan 1. Piksel dengan nilai 0 akan ditampilkan dengan warna hitam, dan piksel dengan warna 1 akan ditampilkan dengan warna putih. Namun dalam Binary Image ini, kita dapat menampilkan nilai 0 sebagai warna putih dan nilai 1 sebagai warna hitam dengan menggunakan notasi NOT ("") pada variabel yang menampung image tersebut.

2.5.1 Morphological Operation pada Binary Image dengan Matlab

Berikut ini fungsi-fungsi yang ada pada Matlab yang dapat digunakan untuk pengolahan citra pada Binary Image berdasarkan Morphological operation.

1) Untuk memasukkan image yang akan diproses ke dalam Morphological Operation Pada Matlab digunakan Perintah:

```
Imagname = input ('n n Enter the name of the image file  
(filename.ext)'); 's')
```

Dimana variabel imagname adalah variabel yang digunakan untuk menampung image yang diinput oleh user. Perintah input sendiri digunakan untuk peng-inputan file.

```
a = imread(imagname)
```

```
imshow(a)
```

a adalah variabel yang digunakan untuk image yang sudah dibaca oleh Matlab tersebut.

`a = im2bw(a,0,5)`

`siz = size(a)`

`im2bw` digunakan untuk merubah RGB atau grayscale image menjadi Binary Image, sehingga image tersebut hanya memiliki warna hitam dan putih saja. Setelah merubah image ini menjadi bentuk binary image maka akan dilakukan proses morphological operation.

2) Morphological Operation yang digunakan pada program aplikasi dengan menggunakan Matlab, adalah dengan menggunakan perintah `bwmorph`. Dengan menggunakan perintah tersebut selanjutnya kita dapat menggunakan perintah-perintah yang telah disediakan oleh Matlab untuk memproses image dengan menggunakan Morphological Operation. Perintah yang digunakan untuk proses Morphological Operation adalah sebagai berikut:

- 'erode' Digunakan untuk melakukan proses erosion pada binary image.
- 'dilate' Digunakan untuk melakukan proses dilation pada binary image.
- 'open' Digunakan untuk melakukan Morphological Opening (erosion kemudian dilation).
- 'close' Digunakan untuk melakukan Morphological Closing (dilation kemudian erosion).
- 'tophat' Top Hatt, mengurangi hasil Morphological Opening (erosion diikuti dilation) dari input image.
- 'bothat' Bottom Hatt, menampilkan Morphological Closing (dilation diikuti erosion) dan mengurangi piksel dari image asli yang diinput.
- 'clean' Menghilangkan piksel yang terisolasi.
- 'fill' Mengisi piksel yang berada didalam (0 dikelilingi oleh 1).
- 'remove' Menjadikan semua piksel menjadi 0 jika 4 piksel tetangga yang terhubung adalah 1, hanya meninggalkan piksel yang menjadi batas-batas.
- 'shrink' Menyusutkan objek ke poin-poin. Perintah ini menghilangkan piksel sehingga objek tanpa celah atau lubang menyusut ke sebuah poin, objek dengan lubang ke sebuah cincin terhubung diantara setiap lubang dan batas luar

- 'thicken' Menebalkan objek dengan menambahkan piksel ke objek luar sehingga objek yang tadinya tidak terhubung menjadi terhubung (B-connected).
- 'thin' Menipiskan objek ke dalam bentuk garis.

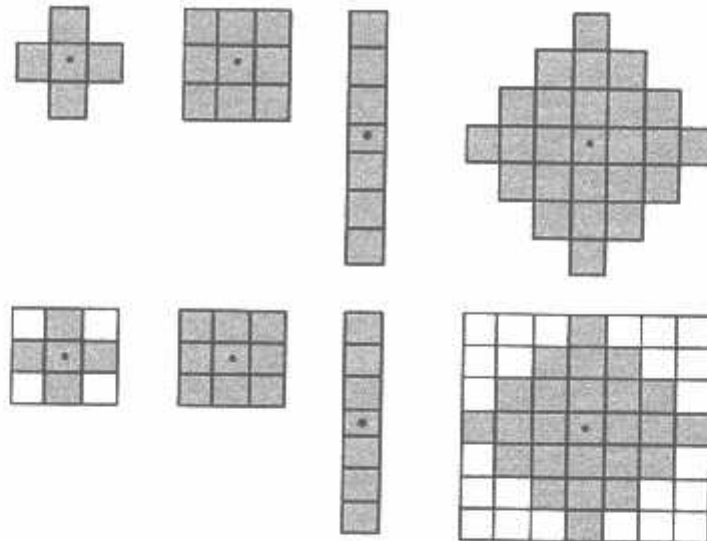
2.5.2 Metode Morphological Image Processing

Metode ini digunakan sebagai tool untuk mengekstrakan komponen citra yang berguna dalam representasi dan deskripsi bentuk daerah, seperti boundaries, skeletons, dan convex hull.

2.5.2.1 Structure Element (STREL)

Strel adalah himpunan sub-image kecil yang digunakan untuk meneliti citra dalam pembelajaran propertinya.

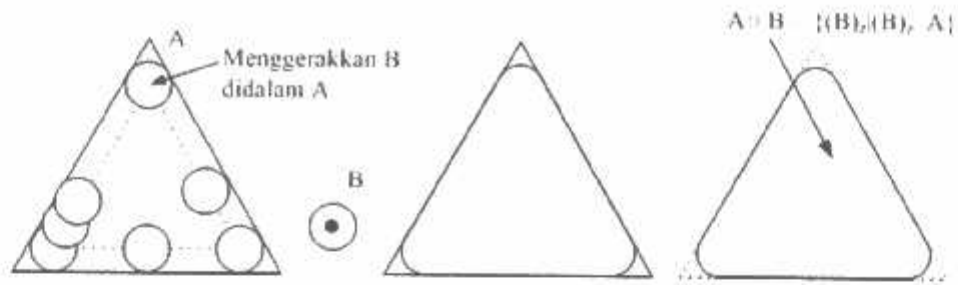
- Untuk elemen yang menjadi anggota strel, origin strel, juga harus ditetapkan.
- Origin dari strel ditandai dengan tanda titik hitam.
 - Jika tidak ada tanda titik hitam maka diasumsikan origin berada di pusat simetri
 - Karena origin tidak harus berada di pusat, tetapi juga bisa berada di pinggir strel.



Gambar 2.10
Contoh strel

2.5.2.2 Operasi Opening

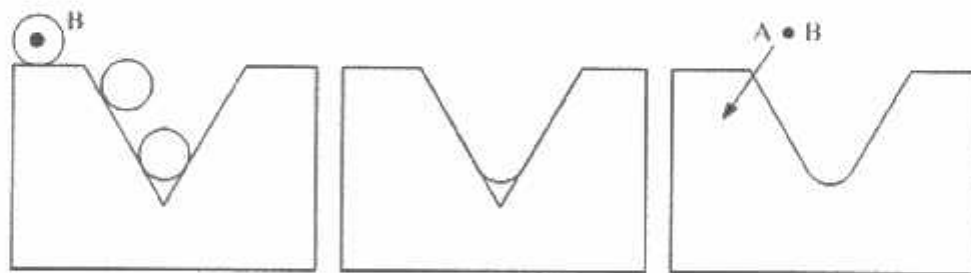
- Opening berfungsi untuk:
 - Menghaluskan garis-garis bentuk obyek,
 - Menghilangkan bagian-bagian yang sempit, dan
 - Menghilangkan penonjolan-penonjolan yang tipis.
- Formula:
 - $A \circ B = (A \ominus B) \oplus B$(3-1)
 - Operasi opening A oleh B adalah erosi A oleh B, diikuti dengan dilasi hasil tersebut dengan B
- Padanan fungsi:
 - $A \circ B = \cup \{(B)_z | (B)_z \subseteq A\}$(3-2)



Gambar 2.11
Gambaran Operasi Opening

2.5.2.3 Operasi Closing

- Closing cenderung menghaluskan garis-garis bentuk tetapi kebalikan dari opening,
 - Menolak pecahan-pecahan sempit dan teluk yang panjang dan tipis,
 - Menghilangkan lubang kecil dan
 - Mengisi gap pada garis-garis bentuk (contour).
- Formula:
 - $A \bullet B = (A \oplus B) \ominus B$(3-3)
 - Operasi closing A oleh B adalah dilasi A oleh B, diikuti erosi hasilnya dengan B.



Gambar 2.12
Gambaran dari Operasi Closing

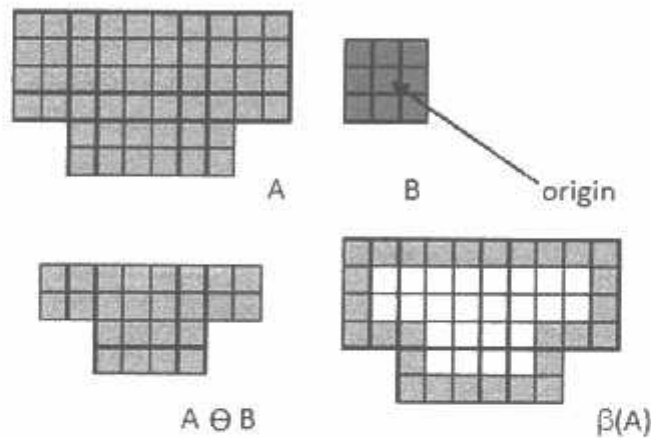
2.5.2.4 Boundary Extraction

Boundary himpunan A dinyatakan oleh $\beta(A)$, bisa didapatkan dengan pertama mengerosi A oleh B dan kemudian melakukan himpunan perbedaan (pengurangan) antara A dan hasil erosi.

- Formula:

$$\beta(A) = A - (A \ominus B) \dots \dots \dots (3-4)$$

di mana B adalah strel yang tepat untuk mengerosi A



Gambar 2.13
Gambaran Boundary Extraction

2.5.2.5 Region Filling

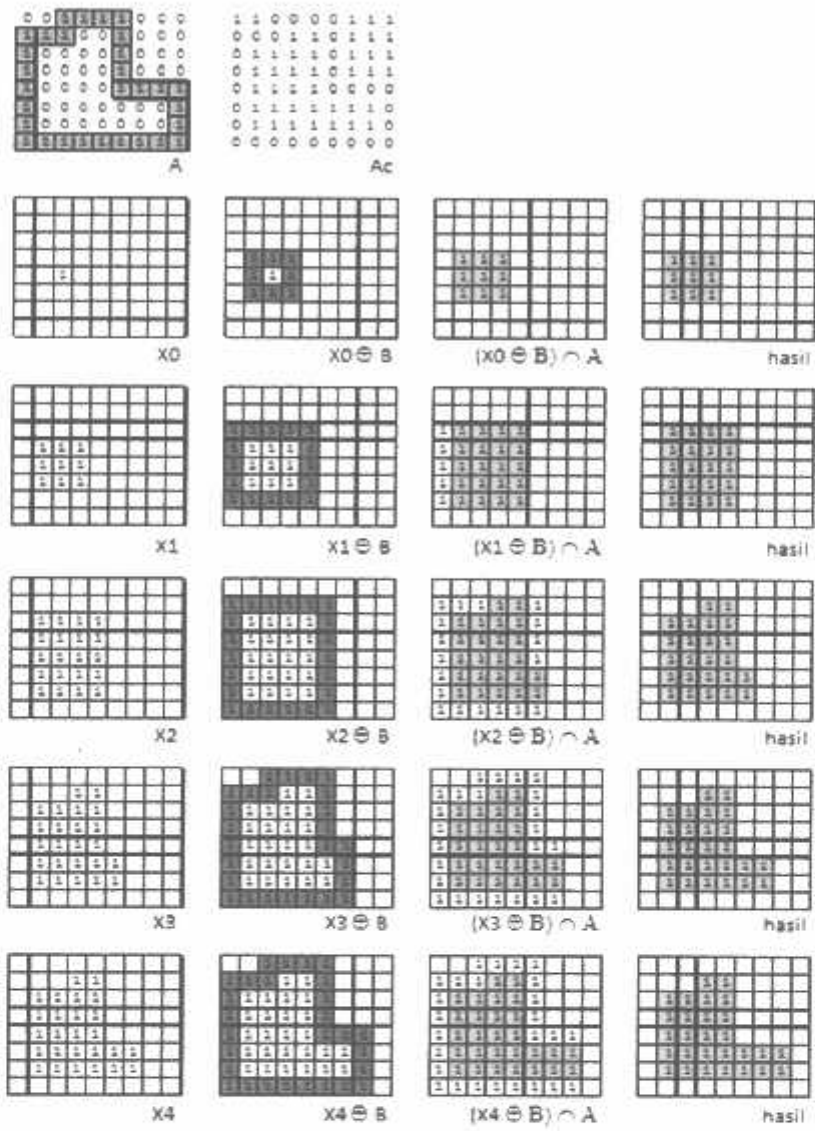
Operasi ini didasarkan pada sejumlah dilasi, komplementasi, dan interseksi. Dimulai dengan sebuah titik p di dalam boundary, tujuannya adalah untuk mengisi semua region dengan nilai 1. Jika mengikuti konvensi bahwa semua titik non-boundary (background) diberi nilai 0, maka p harus diberi nilai 1 untuk memulai. Prosedur di bawah ini adalah untuk mengisi region dengan nilai 1:

- $X_k = (X_{k-1} \oplus B) \cap A^c$ $k = 1, 2, 3, \dots$ (3-5)
di mana $X_0 = p$ dan B adalah strel simetrik. Algoritma berhenti pada iterasi langkah ke k jika $X_k = X_{k-1}$.
- Himpunan union dari X dan A mengisi daerah isian dan boundary-nya.

Toolbox: Bentuk 1 : $BW2 = \text{imfill}(BW, \text{locations})$

Bentuk 2 : $BW2 = \text{imfill}(BW, \text{'holes'})$

Bentuk 3 : $BW2 = \text{imfill}(BW)$



Gambar 2.14
Gambaran Operasi Region Filling

BAB III

DESAIN SISTEM

Pada bab ini dijelaskan mengenai analisa dan perancangan system aplikasi. Analisa ditujukan memberikan gambaran secara umum terhadap aplikasi. Hal ini guna untuk menunjang perancangan aplikasi yang akan dikembangkan sehingga kebutuhan akan aplikasi tersebut diketahui sebelumnya. Kemudian hasil analisa akan menjadi dasar untuk melakukan perancangan atau desain aplikasi sesuai kebutuhan sistem.

Dalam merancang aplikasi pada proyek ini terlebih dahulu dilakukan pembuatan desain sistem, desain proses, serta desain antar muka aplikasi. Desain proses berguna untuk mengintegrasikan semua proses yang terjadi dalam aplikasi yang di buat. Desain data berguna untuk mengetahui apa saja yang dibutuhkan dalam proses yang akan dikerjakan sedangkan perancangan antarmuka berfungsi sebagai antar muka interaksi antara pengguna dengan sistem aplikasi yang dibuat, sehingga pengguna dapat mengoperasikan aplikasi yang dibuat.

3.1. Tujuan Pembuatan Sistem

Perancangan dan pembuatan sistem merupakan hal yang sangat penting dalam skripsi ini. Sebagai langkah awal dalam pembuatan sistem dibutuhkan suatu perancangan terhadap segala komponen yang diperlukan dalam pembuatan sistem, karna dengan perancangan tersebut diharapkan nantinya akan mendapatkan suatu sistem yang baik dan siap untuk beroperasi dengan yang diharapkan.

Tujuan pembuatan sistem ini adalah untuk mempermudah pemilahan biji kacang kedelai tanpa harus bergantung pada pemilahan biji kacang kedelai secara manual.

3.2 Perancangan Sistem

Perancangan sistem yang dibuat adalah mengenai proses pemilahan biji kacang kedelai berbasis matlab, dimana proses pemilahan biji kacang kedelai akan dilakukan berdasarkan tekstur citra dan bentuk citra yang diinputkan ke dalam sistem.

3.3 Desain Sistem

Demi mencapai hasil yang maksimal maka dibuat suatu desain, dimana desain sistem disajikan untuk mengetahui bagaimana aplikasi ini akan berjalan, adapun konsep-konsep dari desain aplikasi pemilahan biji kacang kedelai berbasis matlab pada gambar di bawah ini.



Gambar 3.1 Konsep dasar aplikasi pemilahan biji kacang kedelai berbasis matlab

Dari gambar 3.1 dapat dijelaskan secara umum dan fungsi dari masing-masing blok sebagai berikut :

1. User memasukkan citra biji kacang kedelai yang akan menjalani proses pemilahan.
2. Citra biji kacang kedelai diubah menjadi citra biner, kemudian dilakukan pendeteksian tepi citra dan mendeteksi bentuk dari masing-masing bentuk biji kacang kedelai yang nampak,
3. Bentuk biji kacang kedelai telah dapat dikenali atau diidentifikasi.

3.4 Model Pengembangan

Secara umum aplikasi pemilahan biji kacang kedelai berbasis matlab ini memiliki tahapan seperti gambar di bawah ini.



Gambar 3.2 Alur proses aplikasi pemilahan biji kacang kedelai

Berdasarkan gambar 3.2 diatas, alur proses aplikasi pemilahan biji kacang kedelai menunjukkan bahwa inputan yang akan diolah berupa sebuah citra kemudian citra tersebut diubah dari awalnya citra berwarna menjadi citra biner atau citra keabu-abuan. Proses selanjutnya dari citra tersebut dilakukan pendeteksian tepi agar dapat diketahui dengan jelas garis batas antar citra.

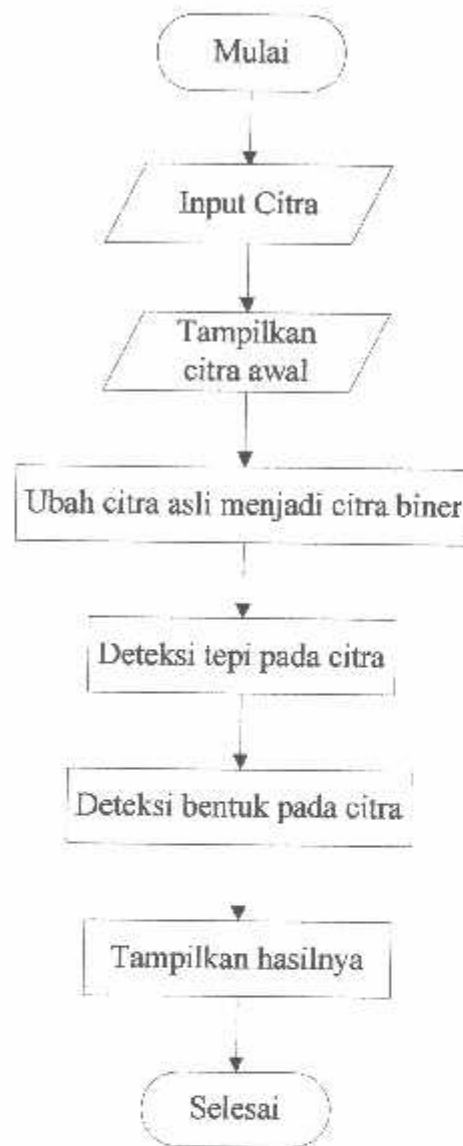
Dan dilanjutkan dengan proses pendeteksian bentuk pada citra untuk mendapatkan bentuk citra yang sesuai dengan parameter yang telah diberikan. Dari proses yang telah dijelaskan diatas, aplikasi tersebut memberikan hasil akhir berupa klasifikasi citra yang memenuhi parameter.

3.5 Perancangan Sistem Aplikasi

Tujuan dari perancangan secara umum adalah untuk memberikan gambaran secara umum tentang aplikasi pemilahan biji kacang kedelai berbasis matlab. Karena aplikasi pemilahan biji kacang kedelai ini dimulai dari penentuan fakta awal terlebih dahulu sebelum menemukan sebuah hasil atau kesimpulan. Maka perancangan yang akan dibuat pada aplikasi pemilah biji kacang kedelai ini menggunakan beberapa form yang berkesinambungan agar mendapat tampilan aplikasi yang baik sesuai dengan yang diharapkan.

3.6 Perancangan Diagram Alir

Perancangan diagram alir akan menjelaskan bagaimana proses dan urutan dari aplikasi yang dibuat seperti pada gambar dibawah ini :



Gambar 3.3 Diagram Alir Program Aplikasi

3.7 Proses Pengolahan Citra

Pengolahan citra merupakan proses pengolahan dan analisa citra. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas,

pengolahan citra digital juga mencakup semua data dua dimensi. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit – bit tertentu . Citra digital dapat didefinisikan sebagai fungsi dua variabel $f(x,y)$, dimana x dan y adalah koordinat spatial dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut.

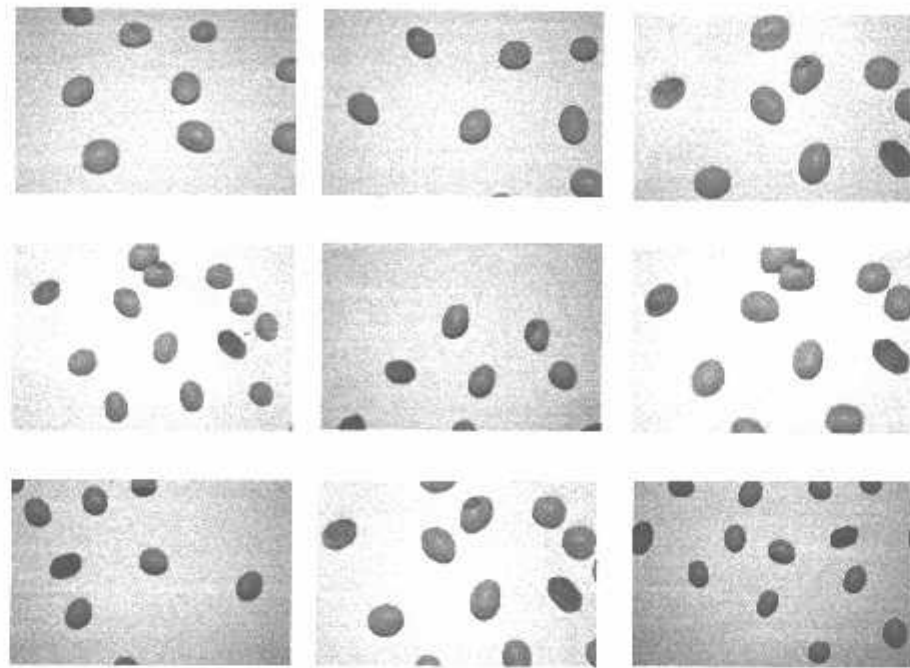
3.7.1 Citra Awal

Sebagaimana yang telah dijelaskan awal bab ini, terhadap citra awal ini merupakan awal dari proses pengolahan citra yang akan berlangsung selanjutnya.



Gambar 3.4
Biji Kacang Kedelai

Rata-rata ukuran biji kacang kedelai adalah dengan panjang 8mm dan lebar 6 mm. Berikut ini spesifikasi citra awal yang digunakan sebagai inputan untuk pemrosesan aplikasi.



Gambar 3.5

Sembilan buah citra inputan yang akan diproses

Ukuran pixel citra : 4608 x 3702

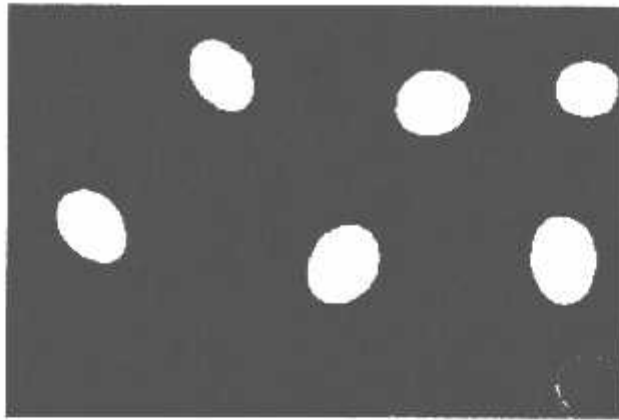
Citra warna: 24 bit

Format citra: jpeg

3.7.2 Mengubah Citra Awal Menjadi Citra Biner

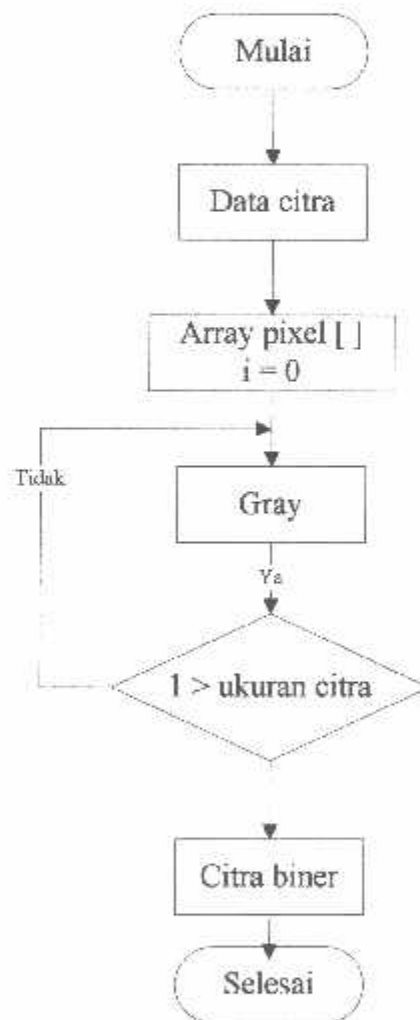
Mengubah citra awal yang berupa citra berwarna menjadi citra biner adalah pengolahan citra digital membutuhkan memori yang jauh lebih besar dibanding pengolahan teks pada komputer. Untuk menghemat kebutuhan memori, warna citra diubah ke tingkat grayscale terendah yang berarti citra hanya mengandung informasi hitam dan putih pada piksel-piksel penyusunnya, yang disebut juga dengan citra biner, sehingga operasi dapat tetap dijalankan walaupun memori yang tersedia relatif kecil. Selain itu, sifat-sifat pengolahan citra biner umumnya lebih sederhana, cepat dan mudah diimplementasikan.

Setelah pengubahan citra warna menjadi citra biner dilakukan maka hasil yang akan diperoleh adalah seperti gambar di bawah ini.



Gambar 3.6

Hasil proses pengubahan citra biner pada citra



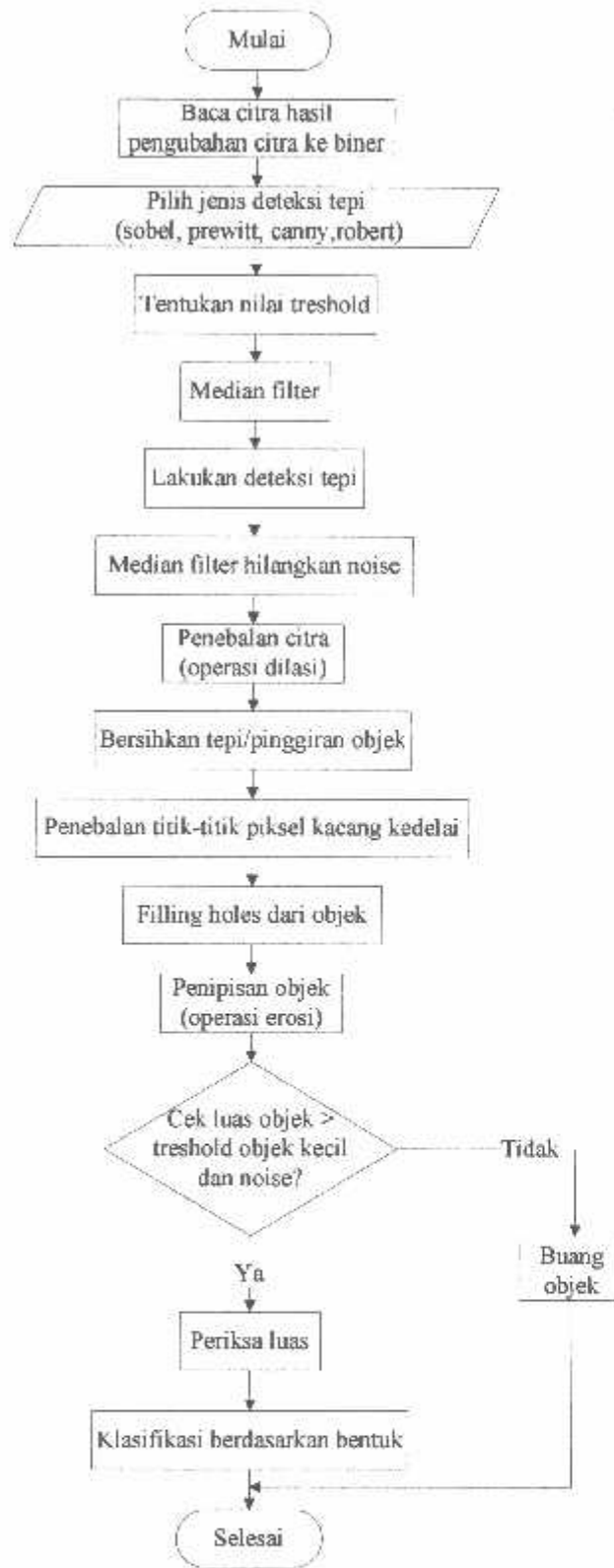
Gambar 3.7

Diagram Alir proses pengubahan citra asli menjadi citra biner

3.7.3 Deteksi dan Klasifikasi Citra

Pendeteksian tepi citra menggunakan metode Sobel yang merupakan pengembangan metode robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode sobel ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi.

Pada tahap selanjutnya dilakukan proses deteksi bentuk untuk selanjutnya diklasifikasikan berdasarkan bentuk elips dan circle. Sebelum melakukan proses deteksi, terlebih dahulu dilakukan penghilangan *noise* dengan median filter kemudian melakukan pemilihan jenis deteksi tepi dan menentukan nilai *threshold* untuk proses deteksi tepi, setelah dilakukan deteksi tepi, dilakukan proses penghilangan *noise* yang kecil berupa bintik-bintik menggunakan median filter dengan ukuran *kernel* 2x2. Selanjutnya dilakukan proses penebalan titik-titik objek sebelum, penebalan dilakukan menggunakan operasi dilasi dengan ukuran struktur element ($SE=2$) kemudian dilanjutkan dengan membersihkan batas tepi (*clear border*) menggunakan hubungan antar piksel sebesar 8, kemudian dilakukan penebalan kembali terhadap titik-titik piksel objek untuk kemudian dilakukan pengisian lubang (*filling holes*). Kemudian dilakukan penipisan objek untuk mengembalikan objek ke ukuran semula. Selanjutnya diperiksa apakah mencukupi nilai *threshold* yang ditetapkan atau tidak, dalam hal ini nilai *threshold* yang dipilih adalah 250.



Gambar 3.8

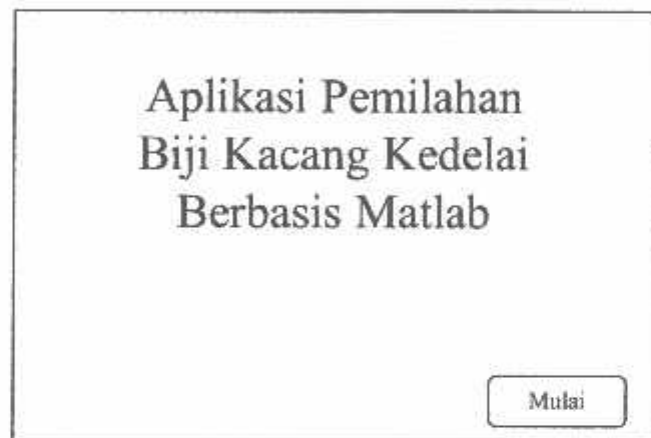
Diagram alir proses pendeteksian tepi pada citra.

3.8 Perancangan Interface

Perancangan interface merupakan rancang bangun dari interaksi user dengan computer. Interaksi ini dapat berupa proses penginputan data ke sistem, pemilihan menu dan menjalankan aplikasi.

3.8.1 Tampilan form awal

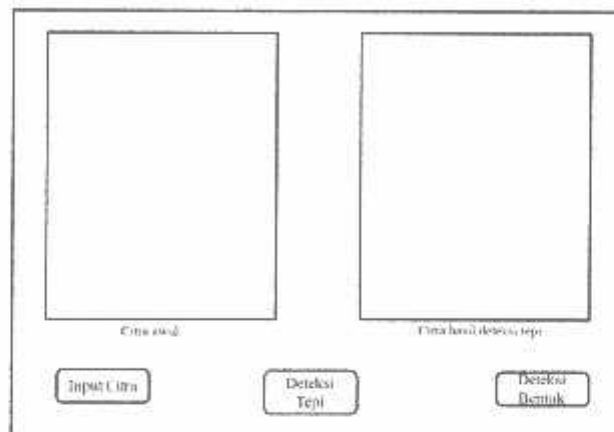
Pada aplikasi pemilahan biji kacang kedelai ini akan menampilkan form awal seperti pada gambar di bawah ini.



Gambar 3.9 Tampilan Form Awal Aplikasi

3.8.2 Tampilan Form Kedua

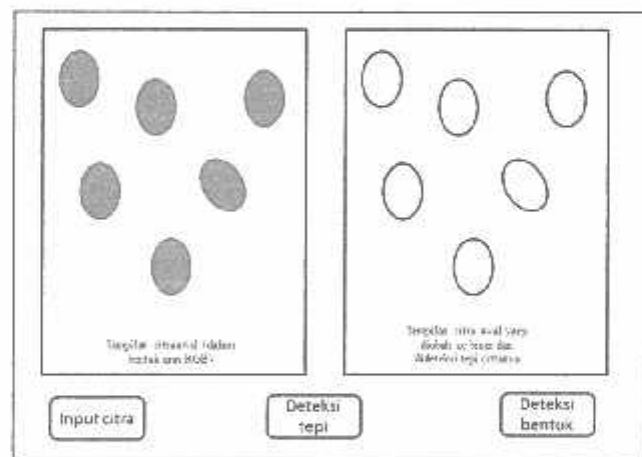
Setelah menekan button mulai pada form awal, maka akan muncul form kedua yaitu form deteksi tepi. Pada perancangan form deteksi tepi ini, user akan menginputkan citra dan mulai melakukan proses pengubahan citra berwarna menjadi citra biner dan kemudian dilakukan deteksi tepi pada citra. Untuk lebih jelasnya, dapat dilihat pada gambar di bawah ini.



Gambar 3.10 Tampilan Form Kedua

3.8.3 Tampilan Form Ketiga

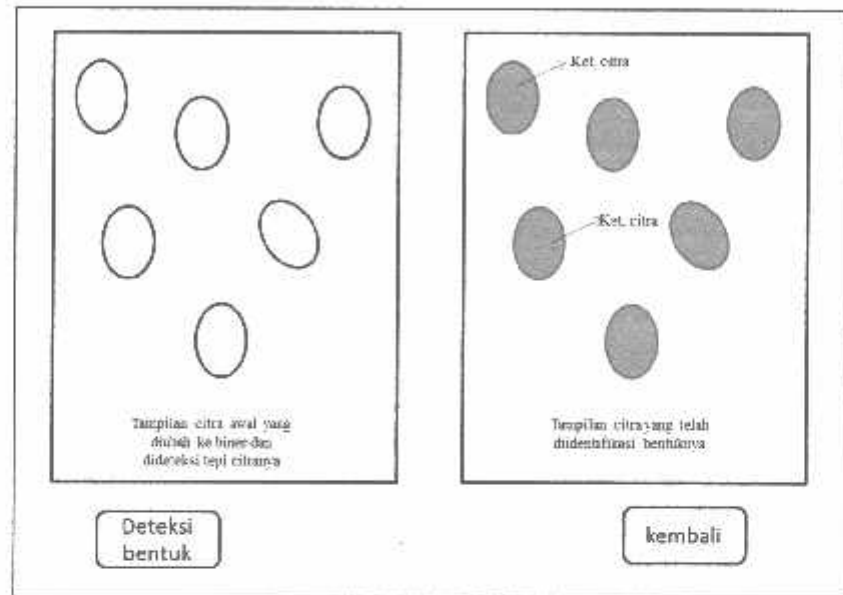
Setelah menekan button deteksi bentuk pada form deteksi tepi, maka akan muncul form ketiga yaitu form deteksi bentuk. Pada perancangan form deteksi bentuk ini, setelah pendeteksian tepi sukses, kemudian klik button deteksi bentuk maka akan pindah ke form deteksi bentuk seperti pada gambar di bawah ini.



Gambar 3.11 Tampilan hasil deteksi tepi

3.8.4 Tampilan Form Keempat

Setelah pendeteksian tepi pada citra sukses, kemudian klik button deteksi bentuk maka akan pindah ke form deteksi bentuk seperti pada gambar di bawah ini.



Gambar 3.12 Tampilan hasil akhir

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Dalam bab ini akan dijelaskan mengenai hasil uji coba perangkat lunak Aplikasi Pengolahan Citra Digital Untuk Memilah Biji Kacang Kedelai Berbasis Matlab pada perangkat komputer atau laptop. Selain itu dari hasil uji coba yang telah dilakukan akan dianalisa apakah rancangan ini dapat memenuhi tujuan yang akan di apai seperti yang telah dijelaskan pada Bab I.

4.1 Implementasi Sistem

Berikut ini adalah spesifikasi perangkat lunak sebagai sistem pendukung aplikasi yang dibuat :

1. Sistem Operasi Windows : Microsoft Windows 7
2. Software : Matlab R2008b
3. Kamera Nikon D5200 (sebagai alat pembantu pengambilan citra)

4.2 Implementasi GUI

4.2.1 Form Awal Aplikasi

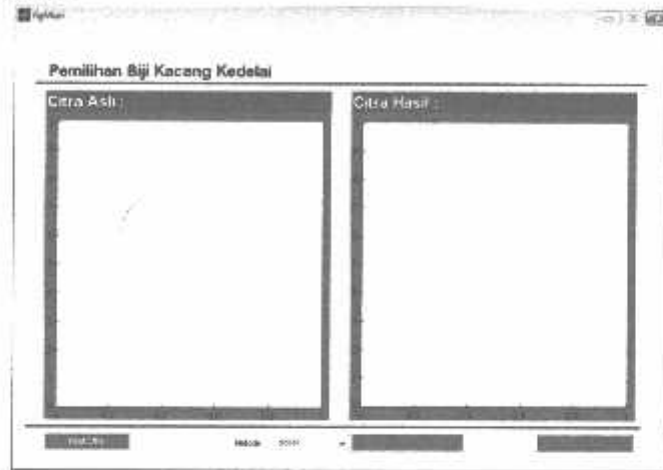
Dari rancangan yang telah dibuat pada bab sebelumnya, maka tahap selanjutnya yaitu mengimplementasikannya menjadi sebuah tampilan.



Gambar 4.1
Tampilan Awal Aplikasi

Gambar 4.1 merupakan gambar tampilan awal atau tampilan pembuka aplikasi pemilahan biji kacang kedelai.

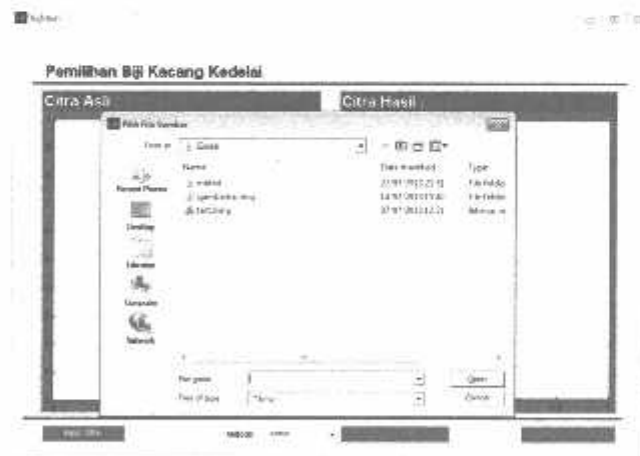
4.2.2 Form kedua



Gambar 4.2
Tampilan form kedua

Gambar 4.2 merupakan tampilan form kedua dari aplikasi pemilahan biji kacang kedelai. Dalam form ini terdapat button input citra yang berfungsi untuk menginputkan file citra yang ingin diproses.

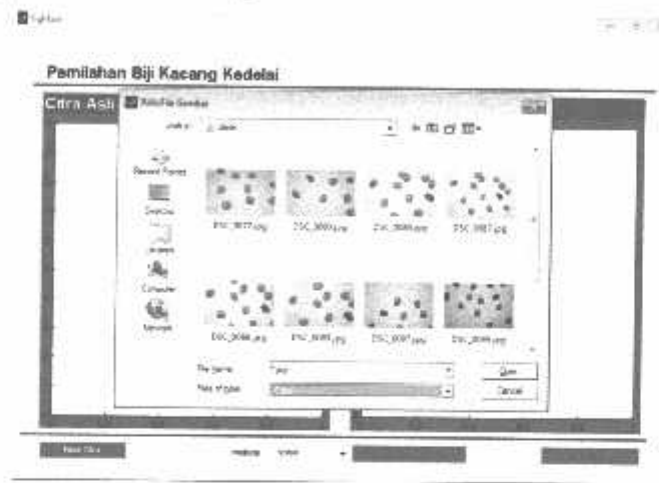
4.2.3 Form Ketiga



Gambar 4.3
Tampilan Form Ketiga

Gambar 4.3 merupakan tampilan form setelah kita menekan button 'input citra' maka akan muncul tampilan sebagai berikut, kemudian pilih file citra yang ingin diproses.

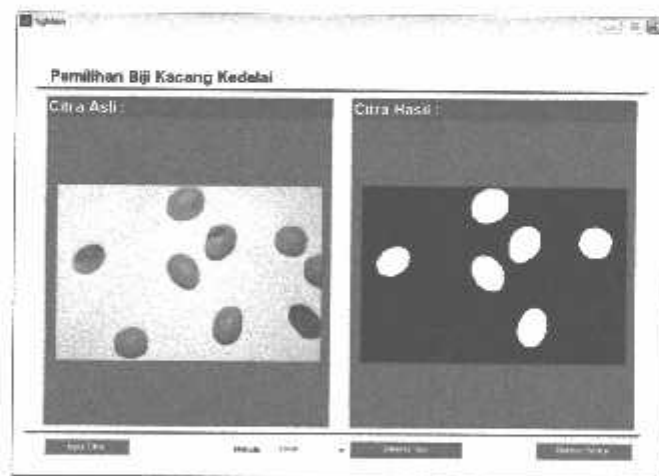
4.2.4 Tampilan Form Keempat



Gambar 4.4
Tampilan form keempat

Gambar 4.4 merupakan tampilan form untuk memilih file citra yang ingin diproses.

4.2.5 Tampilan Form Kelima

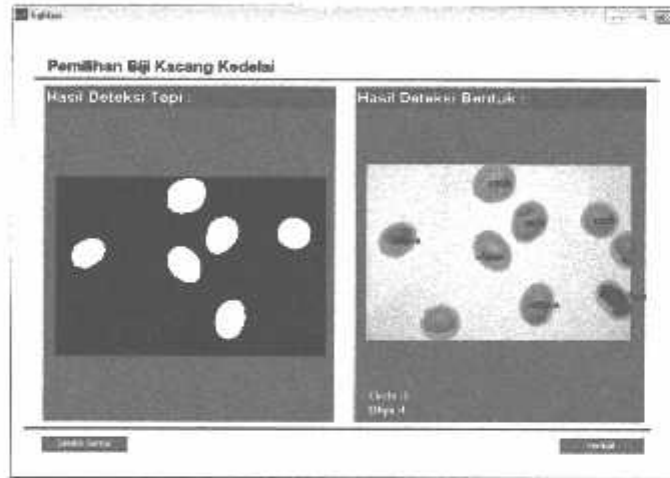


Gambar 4.5
Tampilan Form Kelima

Gambar 4.5 merupakan tampilan gambar apabila kita sudah memasukkan citra yang kita inginkan dan menekan button deteksi tepi untuk mendeteksi tepi dari setiap citra kacang kedelai yang nampak.

Pada citra hasil ditunjukkan bahwa kacang yang berhasil dideteksi adalah yang berwarna putih secara keseluruhan.

4.2.6 Tampilan Form Keenam

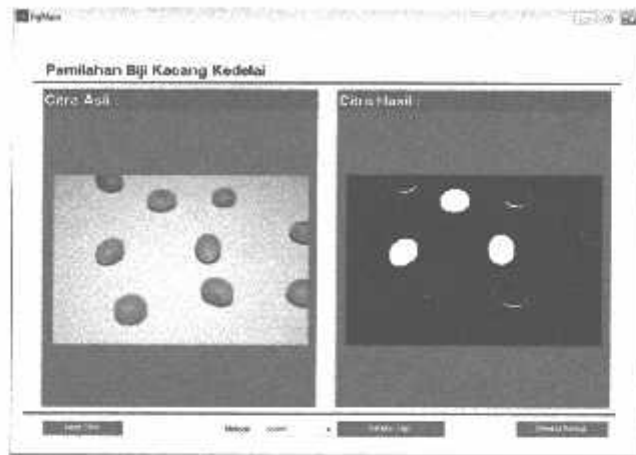


Gambar 4.6
Tampilan Form Keenam

Gambar 4.6 merupakan gambar tampilan ketika bentuk dari citra kacang kedelai yang tampak, sesuai dengan parameter bentuk yang telah ditentukan.

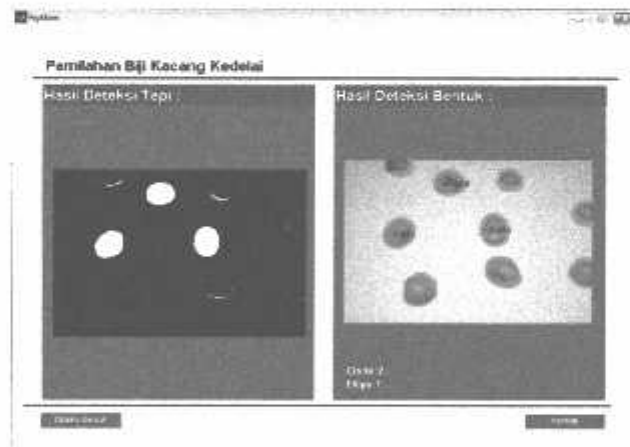
Untuk menguji sejauh mana aplikasi ini berjalan dilakukan 8 kali pengujian pada inputan citra yang berbeda-beda. Berikut ini adalah hasil pengujian 8 inputan citra.

4.2.7 Pengujian citra pertama



Gambar 4.7
Deteksi tepi citra pertama

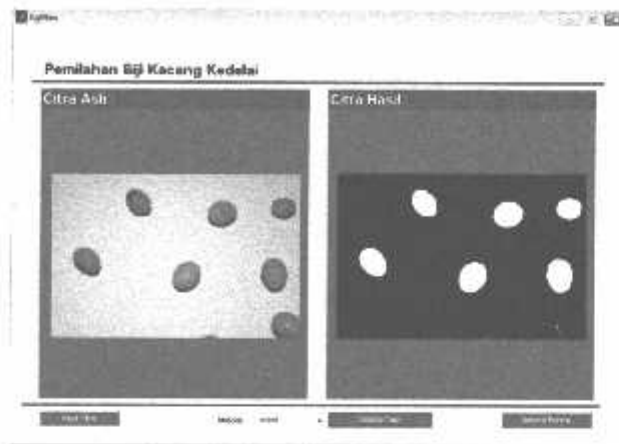
Gambar 4.7 merupakan tampilan gambar apabila kita sudah memasukkan citra yang kita inginkan dan menekan button deteksi tepi untuk mendeteksi tepi dari setiap citra kacang kedelai yang nampak. Pada citra hasil ditunjukkan bahwa kacang yang berhasil dideteksi adalah yang berwarna putih secara keseluruhan.



Gambar 4.8
Deteksi bentuk citra pertama

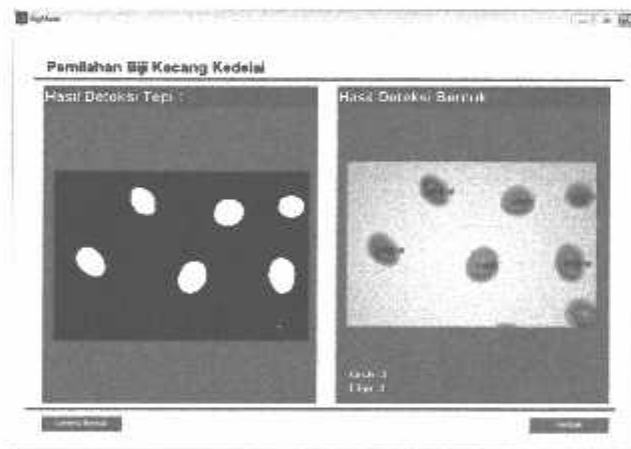
Gambar 4.8 merupakan gambar tampilan ketika bentuk dari citra kacang kedelai yang tampak, sesuai dengan parameter bentuk yang telah ditentukan.

4.2.8 Pengujian Citra Kedua



Gambar 4.9
Deteksi tepi citra kedua

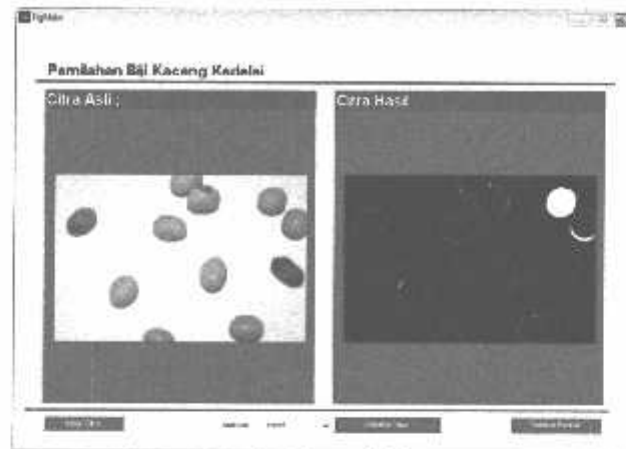
Gambar 4.9 merupakan tampilan gambar apabila kita sudah memasukkan citra yang kita inginkan dan menekan button deteksi tepi untuk mendeteksi tepi dari setiap citra kacang kedelai yang nampak. Pada citra hasil ditunjukkan bahwa kacang yang berhasil dideteksi adalah yang berwarna putih secara keseluruhan.



Gambar 4.10
Deteksi bentuk citra kedua

Gambar 4.10 merupakan gambar tampilan ketika bentuk dari citra kacang kedelai yang tampak, sesuai dengan parameter bentuk yang telah ditentukan.

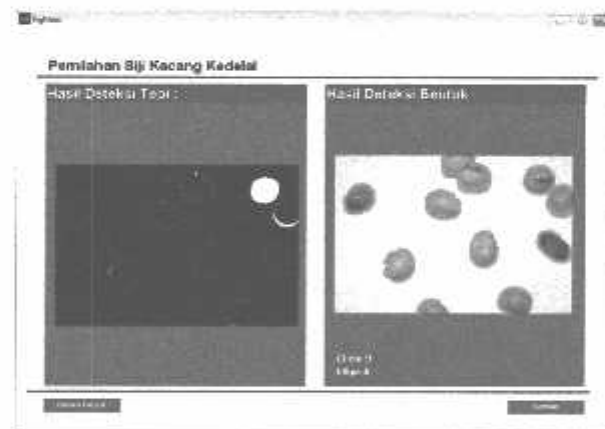
4.2.9 Pengujian Citra Ketiga



Gambar 4.11

Deteksi tepi citra ketiga

Gambar 4.11 merupakan tampilan gambar apabila kita sudah memasukkan citra yang kita inginkan dan menekan button deteksi tepi untuk mendeteksi tepi dari setiap citra kacang kedelai yang nampak. Pada citra hasil ditunjukkan bahwa kacang yang berhasil dideteksi adalah yang berwarna putih secara keseluruhan.

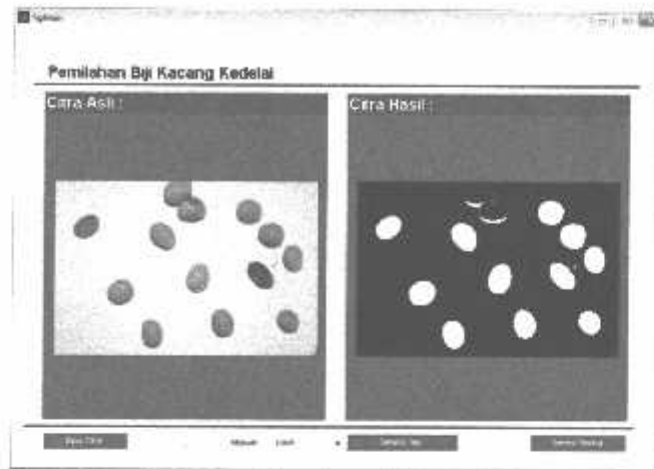


Gambar 4.12

Deteksi bentuk citra ketiga

Gambar 4.12 merupakan gambar tampilan ketika bentuk dari citra kacang kedelai yang tampak, sesuai dengan parameter bentuk yang telah ditentukan.

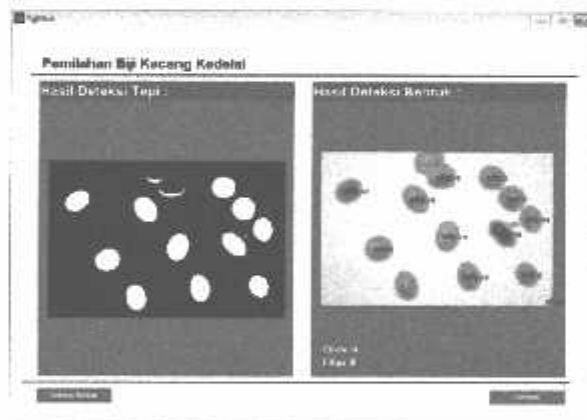
4.2.10 Pengujian Citra Keempat



Gambar 4.13

Deteksi tepi citra keempat

Gambar 4.13 merupakan tampilan gambar apabila kita sudah memasukkan citra yang kita inginkan dan menekan button deteksi tepi untuk mendeteksi tepi dari setiap citra kacang kedelai yang nampak. Pada citra hasil ditunjukkan bahwa kacang yang berhasil dideteksi adalah yang berwarna putih secara keseluruhan.

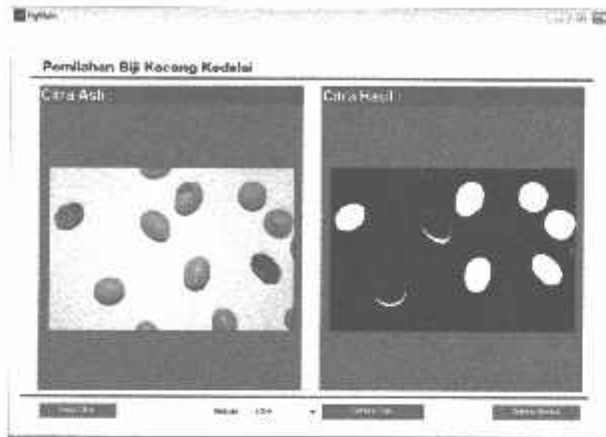


Gambar 4.14

Deteksi bentuk citra keempat

Gambar 4.14 merupakan gambar tampilan ketika bentuk dari citra kacang kedelai yang tampak, sesuai dengan parameter bentuk yang telah ditentukan.

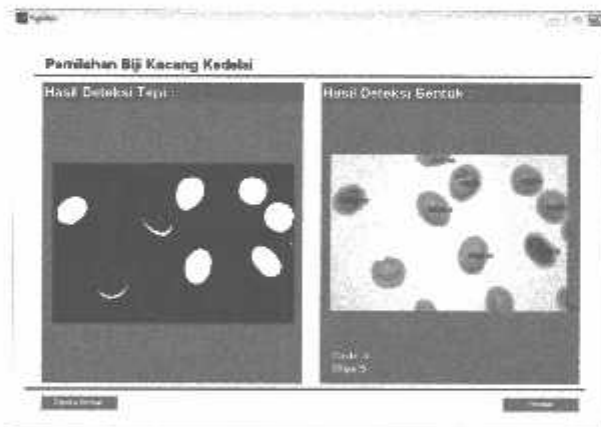
4.2.11 Pengujian Citra Kelima



Gambar 4.15

Deteksi tepi citra kelima

Gambar 4.15 merupakan tampilan gambar apabila kita sudah memasukkan citra yang kita inginkan dan menekan button deteksi tepi untuk mendeteksi tepi dari setiap citra kacang kedelai yang nampak. Pada citra hasil ditunjukkan bahwa kacang yang berhasil dideteksi adalah yang berwarna putih secara keseluruhan.

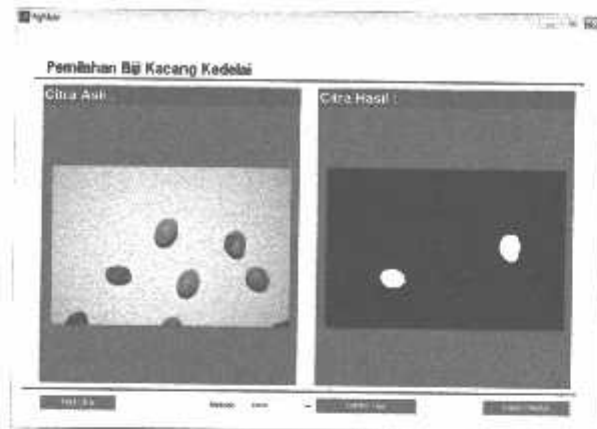


Gambar 4.16

Deteksi bentuk citra kelima

Gambar 4.16 merupakan gambar tampilan ketika bentuk dari citra kacang kedelai yang tampak, sesuai dengan parameter bentuk yang telah ditentukan.

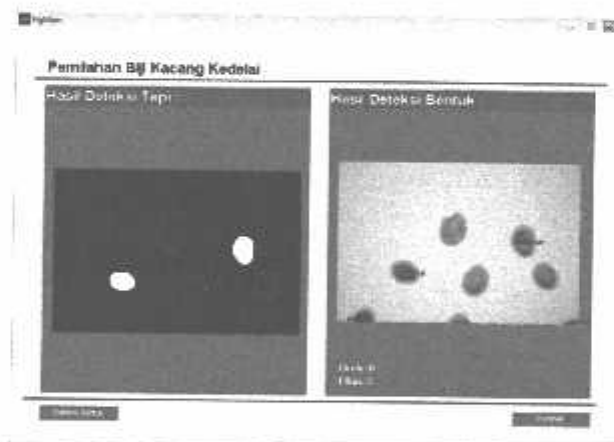
4.2.12 Pengujian Citra Keenam



Gambar 4.17

Deteksi tepi citra keenam

Gambar 4.17 merupakan tampilan gambar apabila kita sudah memasukkan citra yang kita inginkan dan menekan button deteksi tepi untuk mendeteksi tepi dari setiap citra kacang kedelai yang nampak. Pada citra hasil ditunjukkan bahwa kacang yang berhasil dideteksi adalah yang berwarna putih secara keseluruhan.

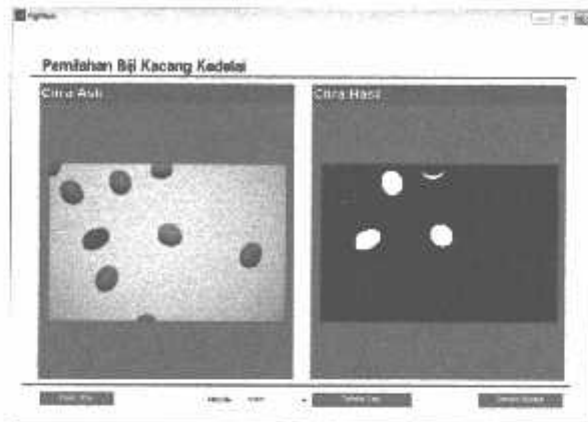


Gambar 4.18

Deteksi bentuk citra keenam

Gambar 4.18 merupakan gambar tampilan ketika bentuk dari citra kacang kedelai yang tampak, sesuai dengan parameter bentuk yang telah ditentukan.

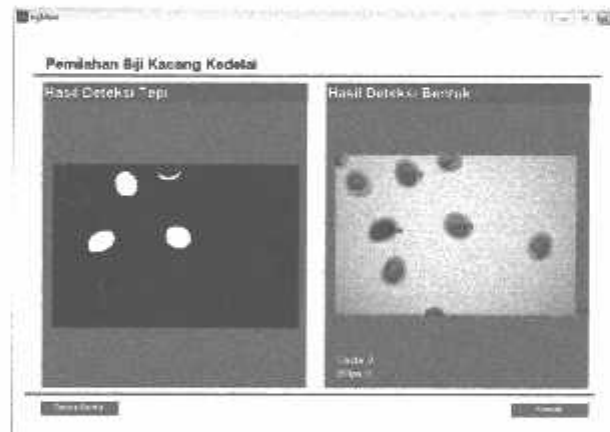
4.2.14 Pengujian Citra Kedelapan



Gambar 4.21

Deteksi tepi citra kedelapan

Gambar 4.21 merupakan tampilan gambar apabila kita sudah memasukkan citra yang kita inginkan dan menekan button deteksi tepi untuk mendeteksi tepi dari setiap citra kacang kedelai yang nampak. Pada citra hasil ditunjukkan bahwa kacang yang berhasil dideteksi adalah yang berwarna putih secara keseluruhan.



Gambar 4.22

Deteksi bentuk citra kedelapan

Gambar 4.22 merupakan gambar tampilan ketika bentuk dari citra kacang kedelai yang tampak, sesuai dengan parameter bentuk yang telah ditentukan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pengujian aplikasi pemilahan kacang kedelai berbasis matlab dapat disimpulkan bahwa:

1. Aplikasi pemilahan biji kacang kedelai berbasis matlab ini telah berhasil mendeteksi bentuk dari kacang kedelai yang nampak pada citra
2. Proses pendeteksian bentuk dapat dilakukan apabila jarak setiap kacang kedelai pada citra tidak terlalu dekat.
3. Aplikasi pemilahan biji kacang kedelai berbasis matlab ini dapat mempermudah proses pemilahan biji kacang dibandingkan dengan cara manual karena dapat dijadikan sampel dari keseluruhan kacang yang ada.

5.2 Saran

1. Untuk mendapatkan hasil pendeteksian bentuk yang baik, sebaiknya jarak tiap kacang kedelai pada citra tidak terlalu dekat
2. Masih banyak yang dapat dikembangkan dari aplikasi pemilahan biji kacang kedelai berbasis matlab, seperti pemilahan berdasarkan tekstur yang nampak pada citra.
3. Untuk memperoleh hasil yang optimal pada pemilahan ini bisa dikembangkan dengan JST dan penambahan hardware sebagai pengembangan dibidang elektronika dan komputer.

DAFTAR PUSTAKA

- 1) AAK. Kedelai. Kanisius, Yogyakarta, 1989.
- 2) Adi Sarwanto, T. , "*Kedelai*", Penebar Swadaya, 2005.
- 3) Ahmad, U. "*Pengolahan Citra Digital dan Teknik Pemrogramannya.*" Graha Ilmu, 2005.
- 4) Bank Pengetahuan Tanaman Pangan Indonesia 2010. Tersedia : deptan.go.id (diakses tanggal 10 April 2013)
- 5) C. Gonzales Rafael, E. Woods Richard, "*Digital Image Processing Third Edition*", Pearson Education, Inc, 2010.
- 6) Ch. Wijaya Marvin, Prijono Agus, "*Pengolahan Citra Digital Menggunakan MATLAB*", Informatika Bandung, 2007.
- 7) Efendi, Nanang. "*BPS Provinsi Banten*". Tersedia : www.bps.go.id (diakses tanggal 10 April 2013)
- 8) Karya Tulis Ilmiah IPB. Tersedia : www.ipb.ac.id (diakses tanggal 10 April 2013)
- 9) Prasetyo Eko, "*Morfologi Citra*", Teknik Informatika Universitas Muhamadiyah Gresik, 2011.
- 10) Rakhman, A.M dan D. Tambas. 1986. *Pengaruh Inokulasi Rhizobium japonicum* L'rank., *Pemupukan Molibdenum dan Kobalt terhadap Produksi dan Jumlah Bintil Akar Tanaman Kedelai pada Tanah Podsolik Plintik*. Direktorat Jendral Pendidikan Tinggi Departemen Pendidikan dan Kebudayaan Fakultas Pertanian Universitas Sriwijaya, Palembang. Hal 7,9.
- 11) Sugiharto, Aris, "*Pemrograman GUI dengan MATLAB*", Andi, Yogyakarta, 2006.
- 12) Supriatna Somantri Agus, Miskiyah dan Broto Wisnu. "*Identifikasi Mutu Fisik Jagung Dengan Menggunakan Pengolahan Citra Digital dan Jaringan Syaraf Tiruan*". tersedia : www.scribd.com (diakses tanggal 5 April 2012)

- 13) Sutoyo. T , Mulyanto Edy, Suhartono. Vincent, Dwi Nurhayati Oky, Wijanarko, *"Teori Pengenalan Citra Digital"*, Andi Yogyakarta dan UDINUS Semarang, 2009.

LAMPIRAN

LISTING PROGRAM

```
function varargout = figMain(varargin)
% FIGMAIN M-file for figMain.fig
%   FIGMAIN, by itself, creates a new FIGMAIN or raises the
existing
%   singleton*.
%
%   H = FIGMAIN returns the handle to a new FIGMAIN or the
handle to
%   the existing singleton*.
%
%   FIGMAIN('CALLBACK', hObject,eventData,handles,...) calls the
local
%   function named CALLBACK in FIGMAIN.M with the given input
arguments.
%
%   FIGMAIN('Property','Value',...) creates a new FIGMAIN or
raises the
%   existing singleton*. Starting from the left, property
value pairs are
%   applied to the GUI before figMain_OpeningFcn gets called.
An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to figMain_OpeningFcn via
varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help figMain

% Last Modified by GUIDE v2.5 21-Jul-2013 05:58:53

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @figMain_OpeningFcn, ...
                  'gui_OutputFcn',  @figMain_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```

% End initialization code - DO NOT EDIT

% --- Executes just before figMain is made visible.
function figMain_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to figMain (see VARARGIN)

% Choose default command line output for figMain
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UWAIT makes figMain wait for user response (see URESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = figMain_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
frm = guidata(gcbo);
set(frm.pnl1, 'visible', 'off');
set(frm.pnl2, 'visible', 'on');

% --- Executes on button press in btnTepi.
function btnTepi_Callback(hObject, eventdata, handles)
% hObject    handle to btnTepi (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

b= feval(@{x)
x{1}{x{2}},get(handles.cbMetode,{'string','value'}));

```

```

S = handles.S;
axes(handles.axHasil);
a=rgb2gray(S);
bw=edge(a,b);
bw = bwareaopen(bw,30);
se = strel('disk',2);
bw = imclose(bw,se);
bw = infill(bw,'holes');
imshow(bw);
handles.bw = bw;
set(handles.btnBentuk,'Enable','on');

% --- Executes on button press in btnInput.
function btnInput_Callback(hObject, eventdata, handles)

% hObject    handle to btnInput (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename, pathname] = uigetfile({'*.bmp'; '*.png'; '*.*'}, 'Pilih
File Gambar');
S = imread([pathname, filename]);
axes(handles.axAsli);
imshow(S);

handles.S = S;
guidata(hObject, handles);
set(handles.btnTepi,'Enable','on');

% --- Executes on button press in btnBentuk.
function btnBentuk_Callback(hObject, eventdata, handles)
% hObject    handle to btnBentuk (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.pnl2,'Visible','off');
set(handles.pnl3,'Visible','on');
b= feval(@x
x{1}{x{2}},get(handles.cbMetode,{'W','log','Vallis'}));
S = handles.S;
axes(handles.axTepi);
a=rgb2gray(S);
bw=edge(a,b);
bw = bwareaopen(bw,30);
se = strel('disk',2);
bw = imclose(bw,se);
bw = infill(bw,'holes');
imshow(bw);
handles.bw = bw;

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in btnDetek.
function btnDetek_Callback(hObject, eventdata, handles)
% hObject    handle to btnDetek (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global i;
b=feval(@(x) x{1}{x{2}}, get(handles.cbMetode, {'String', 'Value'}));
S = handles.S;
axes(handles.axCropping);
format long;
a=rgb2gray(S);

SE = eye(5);
a = imerode(a,SE);

bw=edge(a,b);
bw = bwareaopen(bw,30);
se = strel('disk',2);
bw = imclose(bw,se);
bw = imfill(bw,'holes');
L = bwlabel(bw);
s = regionprops(L, 'centroid');
dt = regionprops(L, 'area');
dim = size(s);
BW_filled = imfill(bw,'holes');
boundaries = bwboundaries(BW_filled);
imshow(S);
hold on;
cir = 0 ;
elp=0;

for k=1:dim(1)
    b= boundaries{k};

    plot(b(:,2),b(:,1), 'g', 'linewidth',3);

    dim = size(b);
    for i=1:dim(1)
        shapes{k}(1,i) = sqrt ( ( b(i,2) - s(k).Centroid(1) )^2 +
( b(i,1) - s(k).Centroid(2) )^2 );
    end

    a=max(shapes{k});
    b=min(shapes{k});
    c=dt(k).Area;
    root=a-b;
    diagonal = c/(4*b^2);
    rectangle=c/(4*b*(a^2-b^2)^0.5);
    triangle=(c*3^0.5)/((a+b)^2);
    elip =c/(a*b*pi);

```

```

diamon= (c*( a^2 - b^2 )^0.5) / (2*a^2*b);
if root <= 0
    text(s(k).Centroid(1)-20,s(k).Centroid(2),'circle');
    cir = cir + 1;
elseif (diagonal < 1.05) && (diagonal > 0.95)
    text(s(k).Centroid(1)-20,s(k).Centroid(2),'circle');
    cir = cir + 1;
elseif (elip < 1.05) && (elip > 0.95)
    text(s(k).Centroid(1)-20,s(k).Centroid(2),'ellipse');
    elp = elp + 1;
elseif (diamon < 1.05) && (diamon > 0.95)
    text(s(k).Centroid(1)-20,s(k).Centroid(2),'ellipse');
    elp = elp + 1;
elseif ((rectangle <1.05) && (rectangle >0.95))
    text(s(k).Centroid(1)-20,s(k).Centroid(2),'ellipse');
    elp = elp + 1;
elseif (triangle < 1.05) && (triangle > 0.95)
    text(s(k).Centroid(1)-20,s(k).Centroid(2),'circle');
    cir = cir + 1;
end
end
disp (cir);
set(handles.txtCir,'String',strcat('Circle : ', num2str( cir)
));
set(handles.txtElp,'String',strcat('Ellipses : ', num2str(elp) ));

% --- Executes on selection change in cbMetode.
function cbMetode_Callback(hObject, eventdata, handles)
% hObject    handle to cbMetode (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: contents = get(hObject,'String') returns cbMetode
contents as cell array
%   contents{get(hObject,'Value')} returns selected item from
cbMetode

% --- Executes during object creation, after setting all
properties.
function cbMetode_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cbMetode (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFns called

% Hint: popupmenu controls usually have a white background on
Windows.
%   See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultuicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```




PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : **CLARA WILDEA MARTHA**
Nim : **09.12.506**
Jurusan : **Teknik Elektro**
Konsentrasi : **Teknik Komputer S-1**
Masa Bimbingan : **Semester Genap 2012-2013**
Judul : **APLIKASI PEMILAHAN BIJI KACANG KEDELAI
BERBASIS MATLAB**

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Program Strata Satu (S-1)

Pada Hari : Jumat
Tanggal : 16 Agustus 2013
Dengan Nilai : 84,75 (A) *✓*

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Sekretaris Majelis Penguji

Dr. Eng. Anyuanjo Soetedjo, ST, MT
NIP.Y.1030800417

ANGGOTA PENGUJI

Penguji I

Irmalia Suryani Faradisa, ST, MT
NIP.P. 1030000365

Penguji II

Bambang Prio Hartono, ST, MT
NIP. Y. 1028400082



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi Teknik Komputer, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : CLARA WILDEA MARTHA
Nim : 09.12.506
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer
Masa Bimbingan : Semester Genap 2012-2013
Judul Skripsi : APLIKASI PEMILAHAN BIJI KACANG KEDELAI BERBASIS MATLAB

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	16 Agustus 2013	• Bab 3, mulai dari citra awal s/d penentuan bentuk	
2	Penguji II	16 Agustus 2013	• Halaman 44 s/d 50 untuk gambar diberi penjelasan. • Daftar pustaka disesuaikan dengan buku panduan	

Disetujui:

Penguji I

Irmalia Suryani Faradisa, ST, MT
NIP.P. 1030000365

Penguji II

Bambang Prio Hartono, ST, MT
NIP. Y. 1028400082

Mengetahui:

Dosen Pembimbing I

Dr. Eng. Arvianto Soetedjo, ST, MT
NIP. V.1030800417

Dosen Pembimbing II

Yuli Wahyuni, ST, MT
NIP.P.1031200456



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Clara Widia M
NIM :
Perbaikan melalui : 0912506

hal 44 s/d 50 untuk gambar
diberi penjelasan

Daftar Pustaka disesuaikan dg
buku panduan

Malang,

(_____)



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formullr Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokorn, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA
N I M
Perbaikan meliputi

Clara
0512506

1 Revisi bab 3. mulai citra awal
s/d ~~setor~~ penentuan bentuk.

Malang,




FORMULIR BIMBINGAN SKRIPSI

Nama : Clara Wildea Martha
Nim : 09.12.506
Masa Bimbingan : Semester Genap 2013
Judul Skripsi : APLIKASI PEMILAHAN BIJI KACANG KEDELAI BERBASIS
MATLAB

No	Tanggal	Uraian	Paraf Pembimbing
1	7 April 2013	Atk bab 1 dan 2	
2	10 Juni 2013	Revisi bab 3	
3	21 Juli 2013	Demo program	
4	1 Agustus 2013	Atk program aplikasi	
5	3 Agustus 2013	Revisi masalah seminar kean!	
6	5 Agustus 2013	Atk bab 3-5	
7			
8			
9			
10			

Malang,
Dosen Pembimbing I


Dr. Eng. Aryanto Soetedjo, ST, MT
NIP.P.1030800417



FORMULIR BIMBINGAN SKRIPSI

Nama : Clara Wildea Martha
Nim : 09.12.506
Masa Bimbingan : Semester Genap 2013
Judul Skripsi : APLIKASI PEMILAHAN Biji KACANG KEDELAI BERBASIS
MATLAB

No	Tanggal	Uraian	Paraf Pembimbing
1	5 April 2013	Revisi bab I dan II	
2	9 April 2013	ACC bab I dan II	
3	14 April 2013	Revisi makalah seminar proposal	
4	7 Juni 2013	Revisi bab III	
5	22 Juli 2013	Demo program aplikasi	
6	25 Juli 2013	Revisi makalah seminar hasil	
7	3 Agustus 2013	ACC bab III-V	
8			
9			
10			

Malang,
Dosen Pembimbing II

Yuli Wahyuni, ST, MT
NIP.P.1031200456



PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : CLARA WILDEA MARTHA
 NIM : 09.12.506
 Semester : VIII (DELAPAN)
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-1
 Konsentrasi : ~~TEKNIK ELEKTRONIKA~~
~~TEKNIK ENERGI LISTRIK~~
~~TEKNIK KOMPUTER DAN INFORMATIKA~~
 TEKNIK KOMPUTER
~~TEKNIK TELEKOMUNIKASI~~
 Alamat : Jalan Golf no. 73 Tunjung berto Karangploso

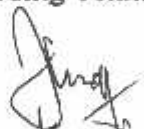
Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat **SKRIPSI Tingkat Sarjana**. Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan-persyaratan pengambilan **SKRIPSI** adalah sebagai berikut :


1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas
 Recording Teknik Elektro


 (..... Handayani))

Malang, 11 Februari 2013
 Pemohon


 (..... CLARA WILDEA M.))

Disetujui
 Ketua Jurusan Tel



PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :


Nama : **CLARA WILDEA MARTHA**
Nim : **0912506**
Semester : **XIII (Tigabelas)**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK MEMILAH BIJI KACANG KEDELAI BERBASIS MATLAB"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami


Dr. Eng. Anyuanto Soetedjo, ST, MT
NIP.P/1030800417

*) Coret yang tidak perlu



PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i : *

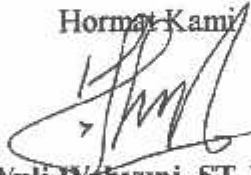
Nama : CLARA WILDEA MARTHA
Nim : 0912506
Semester : XIII (Tigabelas)
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK MEMILAH BIJI KACANG KEDELAI BERBASIS MATLAB"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami



Yuli Wahyuni, ST, MT
NIP.P. 1031200456

Catatan :

Setelah disetujui agar formulir ini Diserahkan mahasiswa/i yang bersangkutan kepada jurusan untuk diproses lebih lanjut

*) Coret yang tidak perlu



an : 1 (satu) berkas
Pembimbing Skripsi

1 : Yth. Bapak/Ibu **Yuli Wahyuni, ST, MT**
Dosen Teknik Elektro S-1
ITN Malang

Yang bertanda tangan dibawah


Nama : **CLARA WILDEA MARTHA**
Nim : **0912506**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing untuk penyusunan Skripsi dengan judul :

"APLIKASI PENGGLAHAN CITRA DIGITAL UNTUK MEMILAH Biji KACANG KEDELAI BERBASIS MATLAB"

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih.

Mengetahui
Program Studi Teknik Elektro S-1


Ibrahim Ashari, ST, MT
NIP.P. 1030100358

Hormat Kami


CLARA WILDEA MARTHA
NIM. 0912506



**BERITA ACARA SEMINAR PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Konsentrasi : Teknik Komputer**

1.	Nim	: 0912506		
2.	Nama	: CLARA WILDEA MARTHA		
3.	Konsentrasi Jurusan	: Teknik Komputer		
4.	Jadwal Pelaksanaan:	Waktu	Tempat	
	17 April 2013	09:00	III.1.3	
5.	Judul proposal yang diseminarkan Mahasiswa	APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK MEMILAH BIJI KACANG KEDELAI BERBASIS MATLAB		
6.	Perubahan judul yang diusulkan oleh Kelompok Dosen Keahlian			
7.	Catatan :	* LATAR BELAKANG LEBIH SPESIFIK TTG BUNIH BISI KEDELAI, TUJUAN & Babasan masalah.		
8.	Catatan :	* Tool yang digunakan harus standar (Kamera). * Studi awal utk mengetahui karakteristik buni kedelai		
	Persetujuan judul Skripsi			
	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III	
	(.....)	(.....)	(.....)	
	Mengetahui, Ketua Program Studi Teknik Elektro S-1	Disetujui, Calon Dosen Pembimbing ybs		
		Pembimbing I	Pembimbing II	
	 M. Ibrahim Ashari, ST, MT NIP. P 1030100358	(.....)	(Yuu wah yuni)	



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

**FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK**

PTN (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

nomor Surat : ITN-137/EL-FTI/2013

1 Mei 2013

kepada : -

keperluan : BIMBINGAN SKRIPSI

kepada : Yth. Bapak/Ibu Dr. Eng. Aryuanto Soetedjo, ST, MT
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **CLARA WILDEA MARTHA**
Nim : **0912506**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Genap Tahun Akademik 2012 - 2013 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Ketua Program Studi Teknik Elektro S-1

M. Irfan Ashari, ST, MT

NIP.P. 1030100358