SKRIPSI

PENGEMBANGAN APLIKASI SISTEM PEMBACA BARCODE BERBASIS CITRA KAMERA

Disusun Oleh :

FITROH AMALUDDIN
06.12.911

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010

LEMBAR PERSETUJUAN

PENGEMBANGAN APLIKASI SISTEM PEMBACA BARCODE BERBASIS CITRA KAMERA

SKRIPSI

Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik Komputer Dan Informatika Strata Satu (S-1

Disusun Oleh :

FITROH AMALUDDIN
NIM : 06.12.911

Diperiksa dan Disetujui

Mengetahui

Ketua Jurusan Teknik Elektro S-1	Dosen Pembimbing
Ir. Yusuf Ismail Nakhoda, MT	Dr. Eng. Aryuanto S., ST., MT.
NIP.Y.1018800189	NIP.Y.1030800417

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010

PENGEMBANGAN APLIKASI SISTEM PEMBACA BARCODE BERBASIS CITRA KAMERA

Fitroh Amaluddin

Jurusan Teknik Elektro S-1, Konsentrasi T.Komputer dan Informatika
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jln. Hokky no 48 Tasikmadu Malang
Scoobydoo_1410@yahoo.co.id

Dosen Pembimbing : I. Dr. Eng. Aryuanto Soetedjo,ST., MT.

Abstraksi

Barcode adalah sistem pengkodean informasi menggunakan bar untuk memudahkan dalam proses pemeriksaan. Biasanya barcode digunakan untuk aplikasi retail dan industri, dan dirancang untuk dapat dibaca dengan menggunakan pemindai laser. investasi yang cukup mahal. Belum lagi harga perangkat lunak yang kompatibel dengan perangkat pindai tersebut. Jadi memaksa pengguna membutuhkan pengembangan dalam metode pemindaian barcode.

Skripsi ini mengusulkan penggunaan kamera digital sebagai pembaca barcode. Pengenalan barcode dapat dilakukan secara konkuren pada beberapa kode.dan juga di lakukan beberapa langkah yaitu melakukan koreksi pada informasi citra yang ditangkap oleh kamera digital beserta proses kalibrasi citra, konversi citra dalam bentuk grayscale threshold yang nantinya di proses menjadi citra biner (0/1). Kemudian melakukan pengidentifikasian area citra barcode yang di seleksi dan menterjemahkan kedalam Angka.

Kata kunci : Barcode, Pengolahan Citra, Kamera

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya, sehingga dapat diselesaikan skripsi yang berjudul **"PENGEMBANGAN APLIKASI SISTEM PEMBACA BARCODE BERBASIS CITRA KAMERA"** ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi pada Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan peyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Prof. Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Sidik Noertjahjono, MT, selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Teknik Elektro S-1.
4. Bapak Dr. Eng. Aryuanto Soetedjo,ST., MT. selaku Dosen Pembimbing.
5. Ayah dan Ibu, saudara-saudara, sahabat-sahabat yang telah memberikan do'a restu, semangat, dukungan, dan biaya.
6. Rekan-rekan instruktur di Laboratorium Pemrograman Komputer dan Multimedia ITN Malang.

7. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penyusun semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Agustus 2010

penyusun

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Barcode adalah susunan garis cetak vertikal hitam putih dengan lebar berbeda untuk menyimpan data-data spesifik seperti kode produksi, nomor identitas, dll sehingga sistem komputer dapat mengidentifikasi dengan mudah, informasi yang dikodekan dalam barcode. Untuk membaca kode barcode, saat ini biasanya masih menggunakan Barcode reader jenis laser. Sedangkan barcode jenis ini masih relatif cukup mahal, Juga rentan rusak[6].

Dengan adanya faktor – faktor diatas menjadi salah satu pemicu berkembangnya sistem pencitraan. Seperti kita jumpai pada system pendeteksi wajah. Dan salah satunya adalah proses identifikasi produk kemasan menggunakan kamera. Dalam pengidentifikasian kode barcode menggunakan kamera banyak kelebihan yang ditawarkan yang tidak dapat dilakukan oleh barcode scanner jenis laser. Seperti pembacaan kode barcode 2 Dimensi, padahal barcode jenis ini mulai banyak digunakan oleh industri manufaktur besar[4].

Pada tahap selanjutnya identifikasi barcode menggunakan kamera juga dapat di kembangkan dan implementasikan pada Handphone berkamera. Yang nantinya bisa lebih bersifat mobile dan praktis[5].

Maka dari itu tugas dari Proyek akhir ini adalah Untuk mencari suatu alternatif lain mesin identifikasi yang dapat mengenali kode barcode. Juga untuk mengurangi kelemahan yang ada pada sistem barcode scanner jenis Laser. Dibalik kelebihan yang ditawarkan teknologi pencitraan menggunakan kamera, terdapat tingkat kesulitan atau kerumitan yang jauh lebih tinggi dibanding sistem - sitem sebelumnya[3].

Pada pembuatan teknologi pencitraan barcode membutuhkan perancangan peranti keras (sensor) untuk data akuisisi, teknik analisa atau pengolahan sinyal (citra,video), pengenalan pola, komputasi cerdas, teknik pemrograman komputer dan beberapa kemampuan lainnya.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka permasalahan yang diangkat pada Skripsi ini adalah bagaimana proses pengolahan dan pengenalan gambar kode barcode dari kamera dan menterjemahkannya kedalam bentuk desimal.

1.3 Tujuan Penelitian

Tujuan dari pelaksanaan skripsi ini adalah sebagai berikut :

1. Memahami tentang pengolahan citra digital.
2. Memahami cara kerja sistem pembacaan kode barcode.
3. Memahami dan mencari solusi untuk menangani masalah pencitraan seperti pengolahan warna, cahaya.

1.4 Batasan Masalah

Berdasarkan rumusan masalah yang telah diuraikan sebelumnya, maka pada pelaksanaan skripsi ini batasan masalah adalah sebagai berikut:

1. Pengambilan gambar barcode dapat dilakukan, selama pencahayaan tidak kontras dengan kode barcodenya, dan dengan jarak yang sesuai serta dengan kemiringan tertentu[2].
2. Hanya menangani kode barcode jenis EAN-13*(Europe Article Number)* / IAN[7] *(International Article Number)*.
3. Pada perangkat memakai kamera digital 4 Megapixel dan PC/Notebook.
4. Citra asli barcode berukuran 360x240 pixel.

1.5 Metodologi

1. Studi literatur

   dengan cara mempelajari dari berbagai macam sumber referensi maupun textbook mengenai image-processing (pengolahan citra digital) serta metode-metode yang akan dipakai dalam proses pembacaan kode barcode.

2. Analisa kebutuhan sistem

   Menganalisa masalah dan mencari solusi pada masalah yang dihadapi dalam pembacaan kode barcode.

3. Perancangan dan Implementasi Perangkat Lunak

   Dalam tahap ini perancangan perangkat lunak meliputi perancangan dan antarmuka dari perangkat lunak yang akan dibangun.

4. Pengujian

   Pada tahapan pengujian untuk menganalisa performansi dari sistem pembaca kode barcode .

1.6 Sistematika Penulisan

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 laporan skripsi yang di dalamnya berisi tentang latar belakang masalah, rumusan masalah, tujuan skripsi, batasan masalah, metodologi pengembangan sistem, dan sistematika pembahasan skripsi.

Bab II : Landasan Teori

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

Bab III : Analisis dan Perancangan Sistem

Bab ini berisi tentang analisis dan perancangan terhadap sistem pembacaan pada citra barcode yang kemudian akan di hasilkan keluaran berupa angka barcode.

Bab IV : Implementasi dan Pengujian

Bab ini berisi tentang hasil pengujian akurasi dalam impelemntasi pembacaan kode barcode dengan menggunakan kamera.

Bab V : Penutup

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

BAB II

LANDASAN TEORI

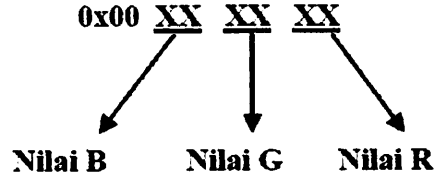
Dalam Bab ini akan dipaparkan mengenai teori – teori dasar yang digunakan sebagai landasan yang digunakan dalam menyelesaikan proyek akhir ini. Berikut teori – teori yang digunakan :

2.1 Pengolahan Citra Digital

Sebuah citra diubah ke bentuk digital agar dapat disimpan dalam memori komputer atau media lain. Proses mengubah citra ke bentuk digital bisa dilakukan dengan beberapa perangkat, misalnya scanner, kamera digital, dan handycam. Ketika sebuah citra sudah diubah ke dalam bentuk digital (selanjutnya disebut citra digital), bermacam-macam proses pengolahan citra dapat diperlakukan terhadap citra tersebut. Image processing atau sering disebut dengan pengolahan citra digital merupakan suatu proses dari gambar asli menjadi gambar lain yang sesuai dengan keinginan kita. Misal suatu gambar yang kita dapatkan terlalu gelap maka dengan image processing gambar tersebut bisa kita proses sehingga mendapat gambar yang jelas. Secara garis besar dapat diilustrasikan sebagai berikut :

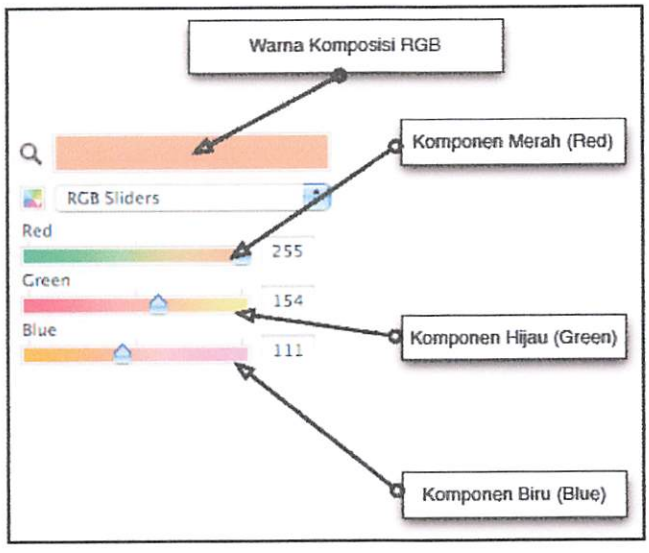
Gambar 2.1. Blok Diagram Pengolahan Citra

Prinsip dasar dari pengolahan citra adalah pengolahan warna RGB pada posisi tertentu. Dalam pengolahan citra warna dipresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00ffffff. Definisi nilai warna di atas seperti gambar 2.2., variabel 0x00 menyatakan angka dibelakangnya adalah hexadesimal.



Gambar 2.2. Nilai warna *RGB* dalam *hexadesimal*

Terlihat bahwa setiap warna mempunyai range nilai 00 (angka desimalnya adalah 0) dan ff (angka desimalnya adalah 255), atau mempunyai nilai derajat keabuan 256 = 28. Dengan demikian range warna yang digunakan adalah (28) (28) (28) = 224 (atau yang dikenal dengan istilah True Colour pada Windows). Nilai warna yang digunakan di atas merupakan gambungan warna cahaya merah, hijau dan biru seperti yang terlihat pada gambar 2.3. Sehingga untuk menentukan nilai dari suatu warna yang bukan warna dasar digunakan gabungan skala kecerahan dari setiap warnanya.



Gambar 2.3. Komposisi warna *RGB*

Dari definisi diatas untuk menyajikan warna tertentu dapat dengan mudah dilakukan, yaitu dengan mencampurkan ketiga warna dasar RGB.

Sebuah citra diubah ke bentuk digital agar dapat disimpan dalam memori komputer atau media lain. Proses mengubah citra ke bentuk digital bisa dilakukan dengan beberapa perangkat, misalnya scanner, kamera digital, dan handycam. Ketika sebuah citra sudah diubah ke dalam bentuk digital (selanjutnya disebut citra digital), bermacam-macam proses pengolahan citra dapat diperlakukan terhadap citra tersebut. seperti *magnified*, *reduced*, atau *rotated*, *kontras*, *brightness* pada citra dapat diubah sesuai keinginan kita.

Tujuan dari pengolahan citra adalah memperbaiki informasi pada gambar sehingga mudah terbaca atau memerpbaiki kualitas dari gambar itu sendiri.

Pengolahan citra mempunyai dua tujuan utama, yaitu sebagai berikut :

1. Memperbaiki kualitas citra, dimana citra yang dihasilkan dapat menampilkan informasi secara jelas. Hal ini berarti manusia sebagai pengolah informasi.
2. Mengekstraksi informasi yang menonjol pada suatu citra, dimana hasilnya adalah informasi citra dimana manusia mendapatkan informasi ciri dari citra secara numerik atau komputer melakun interpretasi terhadap informasi yang ada pada citra melalui besar-besaran data yang dapat dibedakan secara jelas.

2.1.1. Ciri Gambar

Ciri merupakan suatu tanda yang khas, yang membedakan antara satu dengan yang lain. Ciri – ciri dasar dari gambar adalah:

Warna

- Ciri warna suatu gambar dapat dinyatakan dalam bentuk histogram dari gambar tersebut yang dituliskan dengan: H(r,g,b), dimana H(r,g,b) adalah jumlah munculnya pasangan warna r (red), g (green) dan b (blue) tertentu.

Bentuk

- Ciri bentuk suatu gambar dapat ditentukan oleh tepi (sketsa), atau besaran moment dari suatu gambar. Pemakaian besaran moment pada ciri bentuk ini banyak digunakan orang dengan memanfaatkan nilai-nilai transformasi fourier dari gambar.
- Proses yang dapat digunakan untuk menentukan ciri bentuk adalah deteksi tepi, threshold, segmentasi dan perhitungan moment seperti (mean, median dan standard deviasi dari setiap lokal gambar).

Tekstur

- Ciri tekstur dari suatu gambar dapat ditentukan dengan menggunakan filter.
- Ciri tekstur ini sangat handal dalam menentukan informasi suatu gambar bila digabungkan dengan ciri warna gambar.

2.1.2. Macam-Macam Format Citra

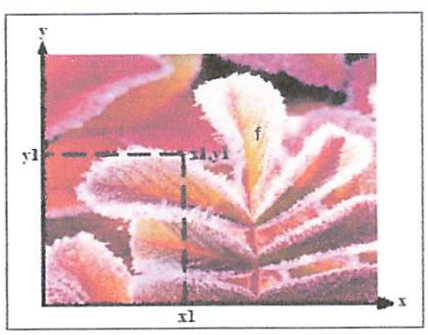
Berikut ini adalah macam-macam format citra :

Ekstensi	Nama	Keterangan
bmp	Windows Bitmap	Format file ini merupakan format grafis yang fleksibel untuk platform Windows sehingga dapat dibaca oleh program grafis manapun. Format ini mampu menyimpan informasi dengan kualitas tingkat 1 bit samapi 24 bit.
gif	Graphics Interchange Format	Format file ini merupakan format standar untuk publikasi elektronik dan internet. Format file mampu menyimpan animasi dua dimensi yang akan dipublikasikan pada internet, desain halaman web dan publikasi elektronik.
jpg/jpeg	Joint Photographic Experts Group	Format file sering dimanfaatkan untuk menyimpan gambar yang akan digunakan untuk keperluan halaman web, multimedia, dan publikasi elektronik lainnya.

png	Portable Network Graphics	Format file ini digunakan untuk menampilkan objek dalam halaman web. Kelebihan dari format file ini dibandingkan dengan GIF adalah kemampuannya menyimpan file dalam bit depth hingga 24 bit serta mampu menghasilkan latar belakang (background) yang transparan dengan pinggiran yang halus.
tiff	Tagged Image File Format	Format file ini mampu menyimpan gambar dengan kualitas hingga 32 bit. Format file ini juga dapat digunakan untuk keperluan pertukaran antar platform (PC, Machintosh, dan Silicon Graphic). Sangat bgus dalam grafis.

2.1.3. Citra Digital

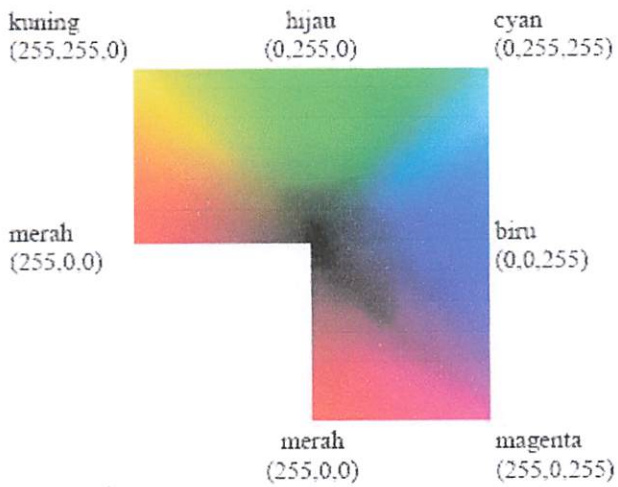
Citra digital dapat didefinisikan sebagai fungsi dua variabel, f(x,y), dimana x dan y adalah koordinat spasial dan nilai f(x,y) adalah intensitas citra pada koordinat tersebut, hal tersebut diilustrasikan pada gambar dibawah ini. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (Red, Green, Blue - RGB).



Gambar 2.4. Citra Digital

2.1.4. Citra RGB

RGB adalah suatu model warna yang terdiri dari merah(red-R), hijau (green-G), dan biru (blue-B), digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang nilai. Untuk monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak 256 x 256 x 256 = 16.777.216 jenis warna. Sebuah jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang 3 dimensi yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y, dan komponen-z. Misalkan sebuah vektor dituliskan sebagai r=(x, y, z). Untuk warna, komponen-komponen tersebut digantikan oleh komponen R(re d), G(g r e e n), B(bl ue). Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = RGB (30, 75, 255). Putih = RGB (255, 255, 255), sedangkan untuk hitam = RGB (0, 0, 0).



Gambar 2.5. Gradasi warna

2.1.5. Teknik Pengolahan Citra Digital

Teknik pengolahan citra dibedakan menjadi 3 tingkat pengolahan[4], yaitu :

1. Tahap 1 *Low Level Processing*, merupakan pengolahan cira, paling dasar dalam pengolahan citra, sebagai contohnya : pengurangan derau (*noise reduction*), perbaikan citra (*image enhancement*) dan resolusi citra (*image restoration*).
2. Tahap 2 *Mid Level Processing* pengolahan citra ini meliputi segmentasi pada citra, deteksi objek dan klasifikasi objek secara terpisah.
3. Tahap 3 *High Level Processing* merupakan tahap analisa citra.

Dari ketiga tahap diatas, dapat dinyatakan suatu gambaran mengenai teknik-teknik pengolahan citra digital.

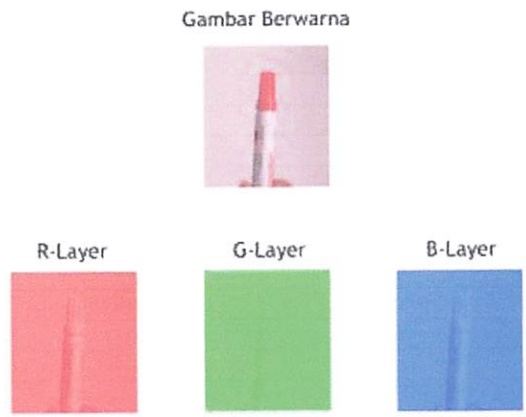
1. *Image enhancement*, berupa proses perbaikan citra dengan meningkatkan kualitas citra baik kontras maupun *brightness*.
2. *Image restoration*, proses memperbaiki model citra, biasannya berhubungan dengan bentuk citra yang sesuai.
3. *Color image processing*, berhubungan dengan citra berwarna baik itu *image enhancement*, *image restoration* atau yang lain.
4. *Wavelet* dan *Multiresolution Processing*, proses yang menyatakan citra dalam beberapa resolusi.
5. *Image compression*, merupakan proses yang digunakan untuk mengubah ukuran data pada citra.
6. *Morphological processing*, proses untuk memperoleh informasi yang menyatakan deskripsi dari suatu bentuk pada citra.

7. *Segmentation*, proses untuk membedakan atau memisahkan objek-objek yang ada dalam suatu citra, seperti memisahkan objek dengan *background*nya.
8. *Object recognition*, proses untuk mengetahui ada tidaknya objek dalam citra.

2.1.6 Citra Grayscale

Proses awal yang banyak dilakukan dalam Pengolahan Citra adalah mengubah citra berwarna menjadi citra grayscale, hal ini digunakan untuk menyederhanakan model citra. Pada awalnya citra terdiri dari 3 layer matrik yaitu R-layer, G-layer dan B-layer. Maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai r,g dan b[5] :

$$Ko = \frac{Ri+Gi+Bi}{3} \tag{2.1}$$



Gambar 2.6. Komposisi warna *RGB*

Sehingga untuk melakukan proses-proses selanjutnya tetap diperhatikan tiga layer di atas. Bila setiap proses perhitungan dilakukan menggunakan tiga layer, berarti dilakukan tiga perhitungan yang sama. Sehingga konsep itu diubah dengan mengubah 3 layer di atas menjadi 1 layer matrik grayscale dan hasilnya adalah citra grayscale. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing r, g dan b menjadi citra grayscale dengan nilai s, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai r, g dan b.

Gambar 2.7. Contoh gambar *derajat keabuan*

Berikut ini adalah proses pengolahan citra digital darii citra RGB menjadi citra Grayscale :

Gambar 2.8. konversi citra *RGB* menjadi citra *Grayscale* dengan *Matlab*

2.1.7 Threshold

Tresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Dengan menggunakan tresholding maka derajat keabuan bisa diubah sesuai keinginan, misalkan diinginkan menggunakan derajat keabuan 16, maka tinggal membagi nilai derajat keabuan dengan 16. Proses tresholding ini pada dasarnya adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan tresholding dengan derajat keabuan dapat digunakan rumus:

$$x = \frac{w}{b} \tag{2.2}$$

dimana :

x adalah nilai derajat keabuan setelah tresholding

w adalah nilai derajat keabuan sebelum tresholding

b adalah jumlah derajat keabuan yang diinginkan

2.1.8 Citra Biner

Citra biner diperoleh melalui proses pemisahan piksel-piksel berdasarkan derajat keabuan yang dimilikinya. Piksel yang dimiliki derajat keabuan lebih kecil dari nilai nilai batas yang ditentukan akan diberikan nol, sementara piksel yang memiliki derajat keabuan yang lebih besar dari batas akan diubah menjadi bernilai satu.

$$f(x,y) = \begin{cases} a_1 & f(x,y) < T \\ a_2 & f(x,y) \geq T \end{cases} \tag{2.3}$$

Jika,a0 = 0 danal = 1, maka operasi ini akan mentransformasikan suatu citra menjadi citra biner. Misal suatu citra memilikigray level 256, dipetakan menjadi citra biner, maka fungsi-fungsi transformasinya adalah sebagai berikut:

$$f(x,y)'=\begin{cases}0. & f(x,y)<128\\ 1. & f(x,y)\geq 128\end{cases} \qquad (2.4)$$

Piksel-piksel yang nilai intensitasnya di bawah 128 diubah menjadi hitam (nilai intensitas = 0), sedangkan piksel-piksel yang nilai diatas 128 diubah menjadi putih (nilai intensitas = 1).

Berikut ini adalah proses pengolahan citra digital dari citra Threshold menjadi citra Biner :

Gambar 2.9. konversi citra *Threshold* menjadi citra *Biner* dengan *Matlab*

2.1.9 Erosi

Erosi merupakan proses penghapusan titik-titik batas objek menjadi bagian dari latar, berdasarkan structuring element yang digunakan. Pada operasi ini, ukuran obyek diperkecil dengan mengikis sekeliling objek[8].

Cara yang dapat dilakukan ada 2:

- *Dengan mengubah semua titik batas menjadi titik latar*
- Dengan menset semua titik di sekeliling titik latar menjadi titik latar.

(a) Citra asli (b) Citra erosi-3 (c) Citra erosi-4

Gambar 2.10. Proses Erosi

2.1.10 Invert

Fungsi invert disini berguna untuk membalikkan nilai. dimama data yang di dapat citra biner pada sebuah gambar adalah jika citra hitam nilainya 0, sedangkan jika citra putih nilai 1. Pada proses pembalikan ini, data dalam bentuk citra biner yang mempunyai nilai 1 dikonversi menjadi nilai 0, sebaliknya yang bernilai 0 dikonversi menjadi 1.

2.2 Matlab

Matlab merupakan suatu software pemrograman perhitungan dan analisis yang banyak digunakan dalam semua area penerapan matematika baik bidang pendidikan maupun penelitian pada universitas dan industri. Dengan matlab,

maka perhitungan matematis yang rumit dapat diimplementasikan dalam program dengan lebih mudah[1].

Matlab merupakan singkatan dari MATriks LABoratory dan berarti software ini dibuat berdasarkan vektor-vektor dan matrik-matrik. Hal ini mengakibatkan software ini pada awalnya banyak digunakan pada studi aljabar linier, serta juga merupakan perangkat yang tepat untuk menyelesaikan persamaan aljabar dan diferensial dan juga untuk integrasi numerik.

Matlab memiliki perangkat grafik yang powerful dan dapat membuat gambar-gambar dalam 2D dan 3D. Dalam hal pemrograman, Matlab serupa dengan bahasa C dan bahkan salah satu dari bahasa pemrograman termudah dalam hal penulisan program matematik. Matlab juga memiliki beberapa **toolbox** yang berguna untuk pengolahan sinyal (signal processing), pengolahan gambar (image processing), dan lain-lain.

Antara Matlab dengan software pemrograman lainnya (C/C++, Visual Basic, Java, dan lain-lain). Terdapat perbedaan yang signifikan. Perbedaan yang utama antara keduanya dapat dilihat dari tiga faktor yaitu tujuan penggunaannya, fitur yang disediakan dan orientasi hasil masing-masing.

Ditinjau dari segi penggunaannya, software pemrograman biasanya berfungsi umum untuk berbagai kebutuhan (misalnya sistem informasi dan database), sedangkan Matlab digunakan spesifik sebagai alat bantu komputasi untuk bidang-bidang ilmiah (pendidikan, riset penelitian akademis, riset penelitian

industri, dan lain-lain) yang membutuhkan library program perhitungan dan tools disain dan analisis sistem matematis.

Ditinjau dari segi fiturnya, bahasa pemrograman umumnya hanya merupakan alat bantu membuat program, sedangkan Matlab dalam softwarenya selain membuat program juga terdapat fitur lain yang memungkinkan Matlab sebagai tools untuk disain dan analisis matematis dengan mudah.

Ditinjau dari segi orientasi hasilya, software pemrograman lain lebih berorientasi sebagai program untuk menghasilkan solusi program baru yang eksekusinya cepat, reliable dan efektif terhadap berbagai kebutuhan. Sedangkan Matlab lebih berorientasi spesifik untuk memudahkan penuangan rumus perhitungan matematis. Dalam hal ini dengan Matlab maka pembuatan program matematis yang kompleks bisa menjadi lebih singkat waktunya namun bisa jadi eksekusi program Matlab ini jauh lebih lambat dibandingkan bila dibuat dengan software pemrograman lainnya.

Matlab dibangun dari bahasa induknya yaitu bahasa C, namun tidak dapat dikatakan sebagai varian dari C, karena dalam sintak maupun cara kerjanya sama sekali berbeda dengan C. Namun dengan hubungan langsungnya terhadap bahasa C, Matlab mempunyai kelebihan-kelebihan bahasa C bahkan mampu berjalan pada semua *platform* Sistem Operasi tanpa mengubah sintak sama sekali. Selain itu Matlab memberikan sistem interaktif yang menggunakan konsep *array*/matrik sebagai standar variabel elemennya tanpa membutuhkan pen-deklarasi-an array seperti pada bahasa lainnya.

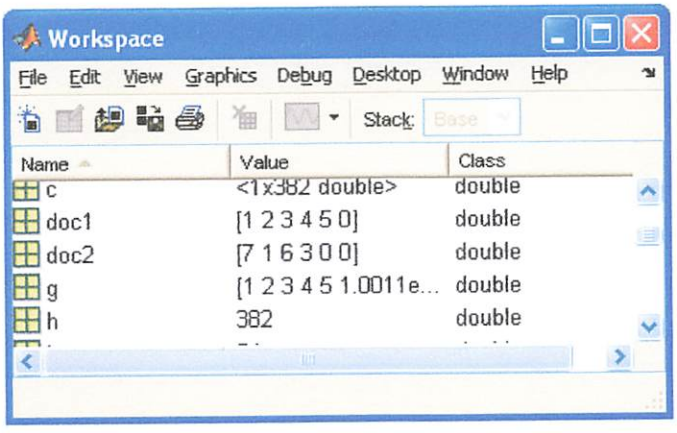
Matlab dikembangkan oleh MathWorks, yang pada awalnya dibuat untuk memberikan kemudahan mengakses data matrik pada program LINPACK dan EISPACK. Selanjutnya menjadi sebuah aplikasi untuk komputasi matrik. Dari sejak awal dipergunakan, Matlab memperoleh masukan ribuan pemakai. Kehadiran Matlab memberikan jawaban sekaligus tantangan. Matlab menyediakan beberapa pilihan untuk dipelajari, mempelajari metoda visualisasi saja, pemrograman saja atau kedua-duanya. Kemudahan yang lain, karena bahasa pemrograman yang lain memang tidak menawarkan kemudahan serupa. Selain itu Matlab juga memberikan keuntungan bagi programmer-developer program yaitu untuk menjadi program pembanding yang sangat handal, hal tersebut dapat dilakukan karena kekayaannya akan fungsi matematika, fisika, statistik dan visualisasi[1].

2.2.1. Pengenalan Matlab 7.0.4

Sebagaimana bahasa pemrograman lainnya, Matlab juga menyediakan lingkungan kerja terpadu yang sangat mendukung dalam pembangunan aplikasi. Tampilan jendela Matlab dapat dibagi menjadi beberapa bagian, yaitu :

1. Jendela Utama
2. Workspace

   Tampilan *workspace* dalam Matlab adalah :

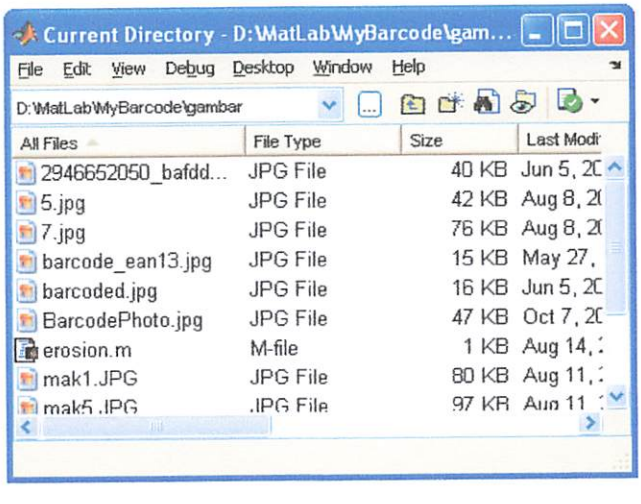


Gambar 2.11. Tampilan *Workspace*

Fungsi *workspace* adalah sebagai navigator bagi pemakai dalam penyediaan informasi mengenai variabel yang sedang aktif dalam workspace pada saat pemakaian. *Workspace* adalah suatu lingkungan abstrak yang menyimpan seluruh variabel dan perintah yang pernah digunakan selama penggunaan Matlab berlangsung.

3. Current directory

Tampilan *current directory* dalam Matlab adalah :

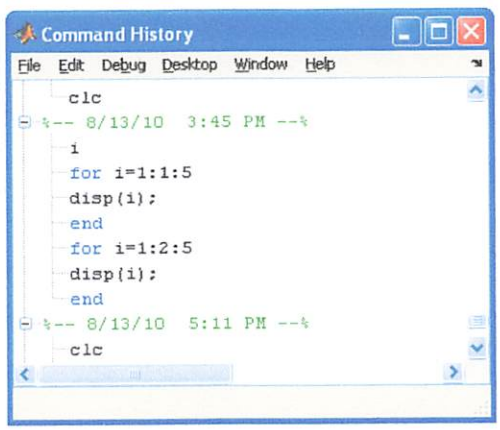


Gambar 2.12. Tampilan *Current Directory*

Fungsi *current directory* adalah memilih direktori yang aktif dan akan digunakan selama penggunaan Matlab berlangsung.

4. Command History

Tampilan *command history* dalam Matlab adalah :

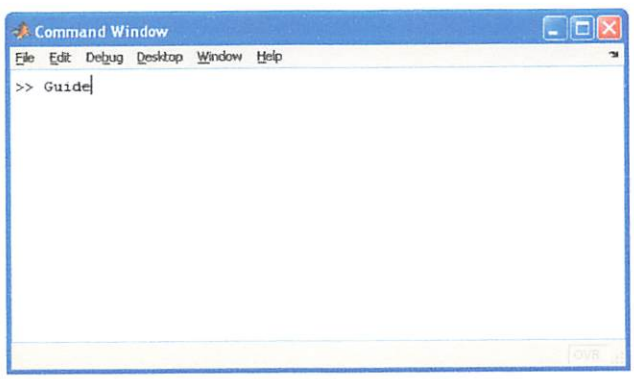


Gambar 2.13. Tampilan *Command History*

Fungsi *command history* adalah menyimpan perintah-perintah yang pernah ditulis pada *command window*.

5. Command window

Tampilan *command window* dalam Matlab adalah :

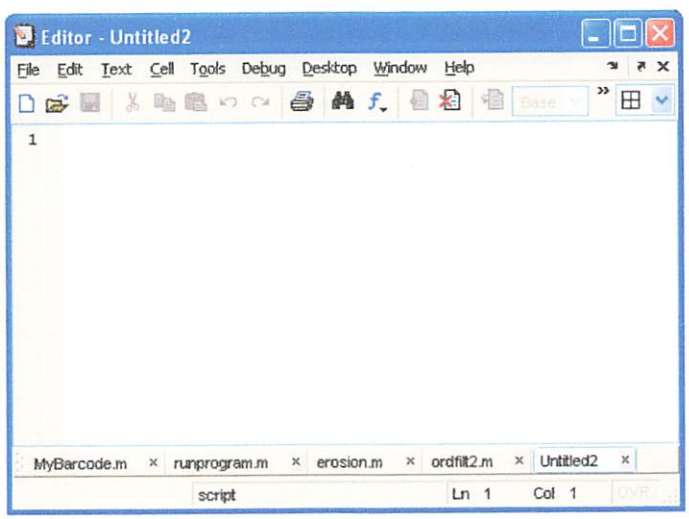


Gambar 2.14. Tampilan *Command Window*

Jendela ini berfungsi sebagai penerima perintah dari pemakai untuk menjalankan seluruh fungsi yang disediakan oleh Matlab. Pada dasarnya jendela ini adalah inti dari pemrograman Matlab yang menjadi media pengguna berinteraksi dengan Matlab.

6. Editor

Tampilan Matlab *Editor* adalah:

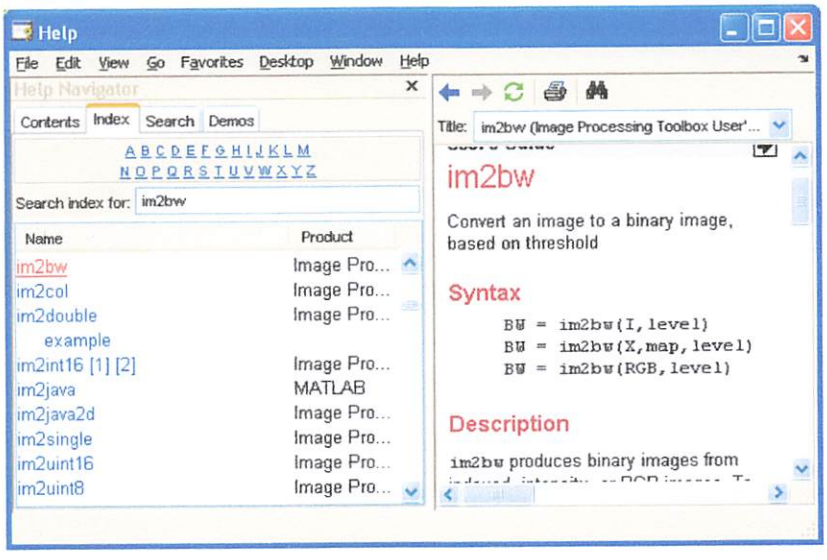


Gambar 2.15. Tampilan Matlab *Editor*

Fungsi Matlab *editor* adalah tempat membuat *script* program pada Matlab. Untuk memunculkan Matlab *Editor*, kita menggunakan perintah File – New – M-file atau dengan mengetikkan >>edit pada command window.

7. Help

Tampilan *Help* dalam Matlab adalah :



Gambar 2.16. Tampilan *Help*

2.2.2. Membuat Aplikasi Matlab 7

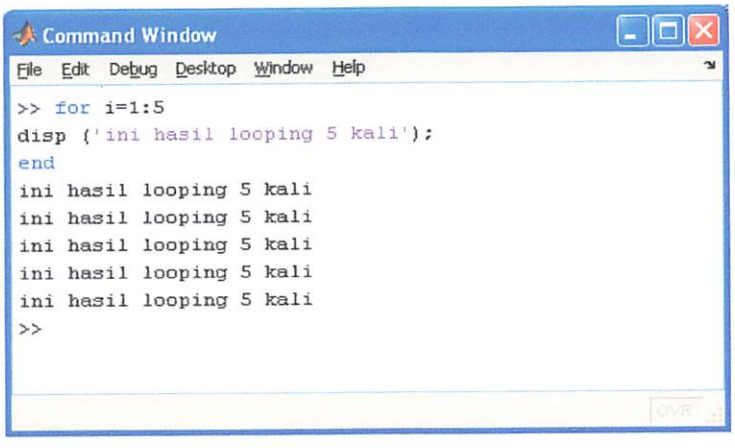
Pada aplikasi ini Matlab bisa berlaku seperti bahasa pemrograman C ataupun pascal yang mempunyai struktur kontrol program, biasanya pemrograman dengan matlab memerlukan lebih dari satu baris dan memungkinkan untuk didokumentasikan dalam m-file, kontrol program ini digunakan untuk memperbaiki tampilan atau membuat tampilan sesuai yang kita inginkan. Dalam bagian dibahas sebagian kontrol program yang diperlukan dalam pemrograman menggunakan matlab :

1. Pada Command Window

Untuk membuat program, hanya diperlukan untuk mengetikkan program pada prompt Matlab dalam Command Window, misalnya :

```
>> for i=1:5
>>disp('Ini hasil looping 5 kali');
>>end
```

Untuk skrip terakhir tidak diberikan tanda titik koma (;) sehingga hasil dari perhitungan diatas dapat langsung dilihat, dengan hasil yaitu :



Gambar 2.17 Hasil Perhitungan pada Command Window

2. Menggunakan Matlab Editor

Pada command window ketikkan:

```
>> edit
```

Tekan enter, selanjutnya muncul Matlab Editor dan ketikkan program dibawah ini:

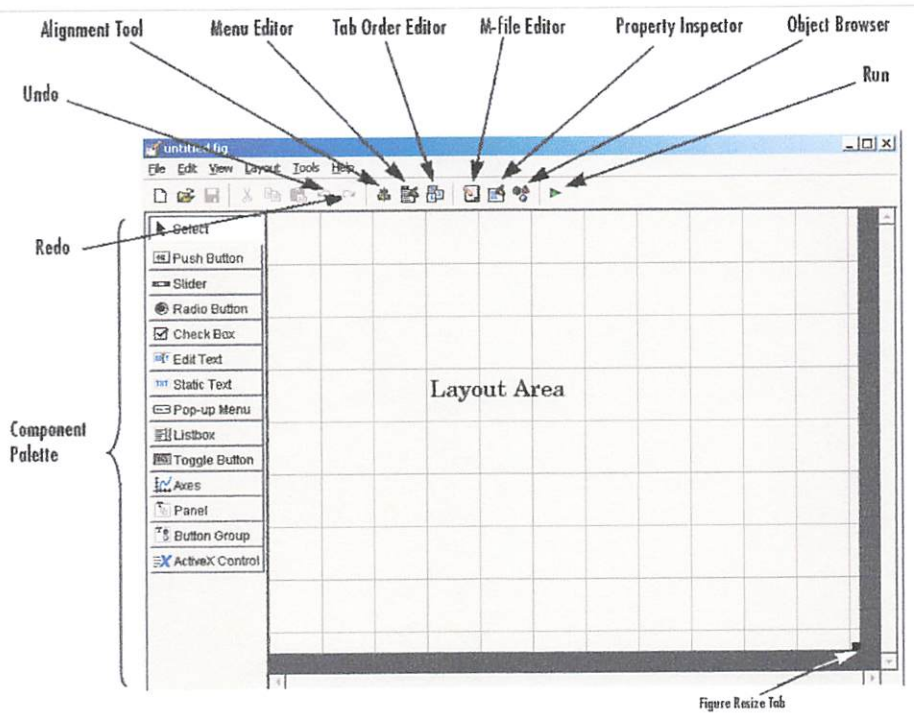
```
% fungsi_for.m - by:Fitroh Amaluddin
for i=1:5
disp('Ini hasil looping 5 kali');
end
```

2.2.3. Matlab GUI (Graphical User Interface)

GUI merupakan tampilan grafis yang memudahkan user berinteraksi dengan perintah teks. Dengan GUI, program yang dibuat menjadi lebih *user friendly*, sehingga user mudah menjalankan suatu aplikasi program.

Untuk membuka lembar kerja GUI dalam Matlab, kita menggunakan perintah File – New – GUI atau dengan mengetikkan `>>guide` pada command window.

Tampilan lembar kerja GUI dalam Matlab adalah :



Gambar 2.18 Tampilan Lembar Kerja GUI Matlab [13]

Pada lembar kerja GUI terdapat component pallete dimana komponen-komponen tersebut yang akan digunakan untuk membuat tampilan Matlab yang lebih user friendly, yaitu :

1. Push Button

   Push button merupakan tombol yang jika diklik akan menghasilkan suatu tindakan.

2. Slider

   Slider menerima masukan berupa angka pada suatu range tertentu dimana pengguna menggeser kontrol pada slider.

3. Radio Button

   Radio Button merupakan kontrol yang digunakan untuk memilih satu pilihan dari beberapa pilihan yang ditampilkan.

4. Check Box

   Check box merupakan kontrol yang digunakan untuk memilih antara satu atau lebih pilihan dari beberapa pilihan yang ditampilkan.

5. Edit Text

   Edit text merupakan kontrol untuk meng-inputkan atau memodifikasi teks.

6. Static Text

   Static text merupakan kontrol untuk membuat teks label.

7. Pop Up Menu

   Pop up menu merupakan kontrol yang digunakan untuk membuka tampilan daftar pilihan yang telah didefinisikan dengan mengklik tanda panah yang terdapat pada pop up menu.

8. List Box

   List Box merupakan kontrol yang digunakan untuk menampilkan semua daftar item. pengguna dapat memilih satu diantara item-item yang ada.

9. Toggle Button

   Toggle button hampir sama dengan push button, hanya push button diklik, tombol akan kembali ke posisi semula. Sebaliknya jika toggle button diklik, tombol tidak kembali ke posisi semula kecuali diklik kembali.

10. Axes

    Axes digunakan untuk menampilkan grafik atau gambar.

11. Panel

    Panel merupakan kotak yang digunakan untuk menandai atau mengelompokkan daerah tertentu pada figure.

12. Button group

    Button group hampir sama dengan panel, tetapi button group lebih digunakan untuk mengelompokkan radio button dan toggle button.

13. Active X Control

    Digunakan jika Matlab berjalan pada Sistem Operasi Microsoft Windows, dengan menggunakan komponen ini Matlab dapat terhubung dengan komponen Microsoft Windows.

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1. Analisa Sistem

Pada bab ini akan dibahas tentang desain sistem *pembacaan barcode* menggunakan menggunakan kamera, mengenai bagian-bagian dari sistem tersebut dan juga cara-cara serta metode yang digunakan.

Cara kerja dari sistem pembaca barcode menggunakan kamera adalah :

a) Input gambar dari hasil pengambilan gambar oleh kamera. Kemudian program menampilkan citra sesuai yang dicapture oleh kamera. Atau user bisa memasukkan gambar selain dari hasil capture oleh kamera. Seperti hasil scan.
b) Kemudian user akan melakukan scan line pada barcode secara horizontal.
c) Program akan menampilkan kode barcode.

3.2. Rancangan Program *Pembaca Barcode*

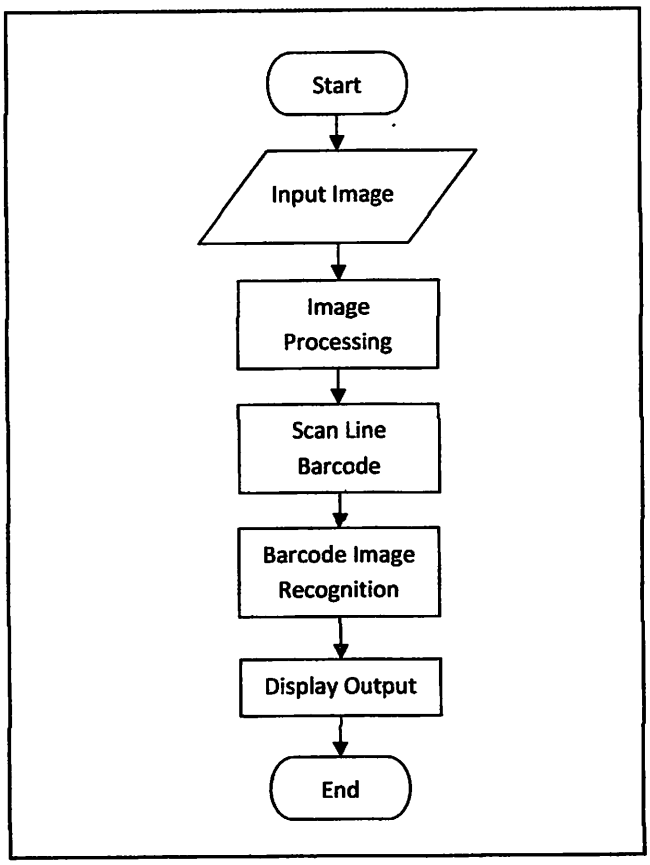
Program *pembaca barcode* yang akan dibangun, menggunakan Matlab 7.0.4. Dari bahasa pemrograman ini akan memanfaatkan *toolbox Image Acquisition* yang telah tersedia. *Toolbox* ini memiliki fungsi-fungsi yang digunakan untuk pengolahan citra, diantaranya yang digunakan untuk program ini adalah untuk mengkoneksikan perangkat keras citra seperti *frame grabber, webcam, firewire* dan lain-lain ke Matlab 7.0.4, melakukan *preview* (peninjauan) secara langsung dari perangkat citra, mengkonfigurasi fungsi *callback* yang dapat mengatur perangkat keras citra, serta membawa data citra ke Matlab.

Berikut adalah urutan rancangan pembuatan program *pembaca barcode* :

a) Kamera menangkap citra gambar berupa citra RGB (*citra berwarna*).
b) Kemudian user melakukan inputan citra yang didapat dari hasil capture oleh kamera pada program.
c) Kemudian program akan memproses inputan citra *RGB* tersebut menjadi outputan citra biner (hitam/putih), agar lebih mudah untuk proses pembacaan. Kemudian user akan melakukan scan line pada barcode secara horizontal.
d) Hasil dari scan line pada barcode berupa pixel-pixel dari citra biner tersebut. dan kemudian melakukan proses kompresi biner pada *guardbar* yang nantinya aka dibuat sample pada bar-bar yang lain.
e) Setelah proses kompresi pada *guardbar,* maka akan dilakukan proses segmentasi. Dari masing-masing kode desimal barcode, terdiri dari 7bit angka biner.
f) Setelah di ketahui 7bit angka biner dari tiap-tiap kode barcode, maka akan dilakukan proses pembacaan digit pertama barcode. Jika kombinasi cocok maka akan menampilkan kode pertama pada barcode.
g) Langkah terakhir adalah proses *check digit*. Dimana program akan melakukan perhitungan chek digit terakhir pada barcode apakah sudah sesuai atau tidak.

3.2.1 Diagram Alir Program Utama

Secara umum sistem *pembaca barcode* dengan menggunakan kamera yang dirancang dapat dilihat pada diagram alir di bawah ini :

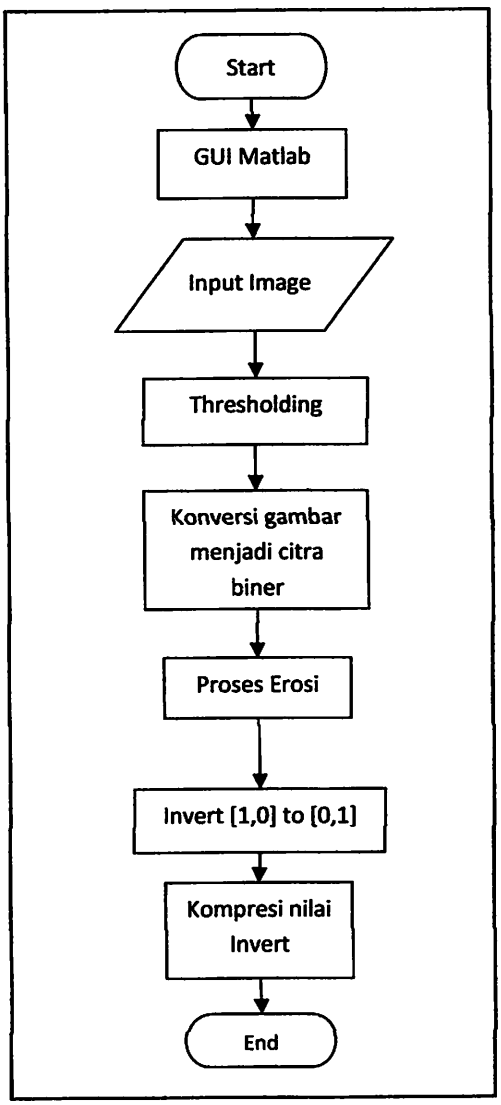


Gambar 3.1 Diagram alir proses *pembacaan barcode*

Dari diagram alir diatas dapat dilihat proses dari program sistem pembacaan kode barcode melalui inputan dari kamera. Langkah pertama adalah user menginputkan gambar hasil dari capture kamera. Kemudian, ketika program dijalankan maka program akan mengolah dan memproses gambar tersebut. Kemudian user akan melakukan scan line pada barcode secara horizontal. Jika kode nilai-nilai biner pada citra barcode tersebut terdeteksi maka program akan melakukan perhitungan seperti kompbinasi parity untuk mengecek digit desimal pertama barcode dan perhitungan untuk check digit terakhir. Kemudian program akan menampilkan hasil ke layar.

3.2.2 Image Processing

Pada proses pengolahan gambar, rincian diagram alirnya sebagai berikut:



Gambar 3.2 Diagram alir Proses Pengolahan Gambar[9]

Diagram alir diatas mendeskripsikan proses pengolahan gambar pada barcode. Proses ini digunakan untuk memfilter gambar yang di inputkan agar lebih mudah untuk dianalisa.

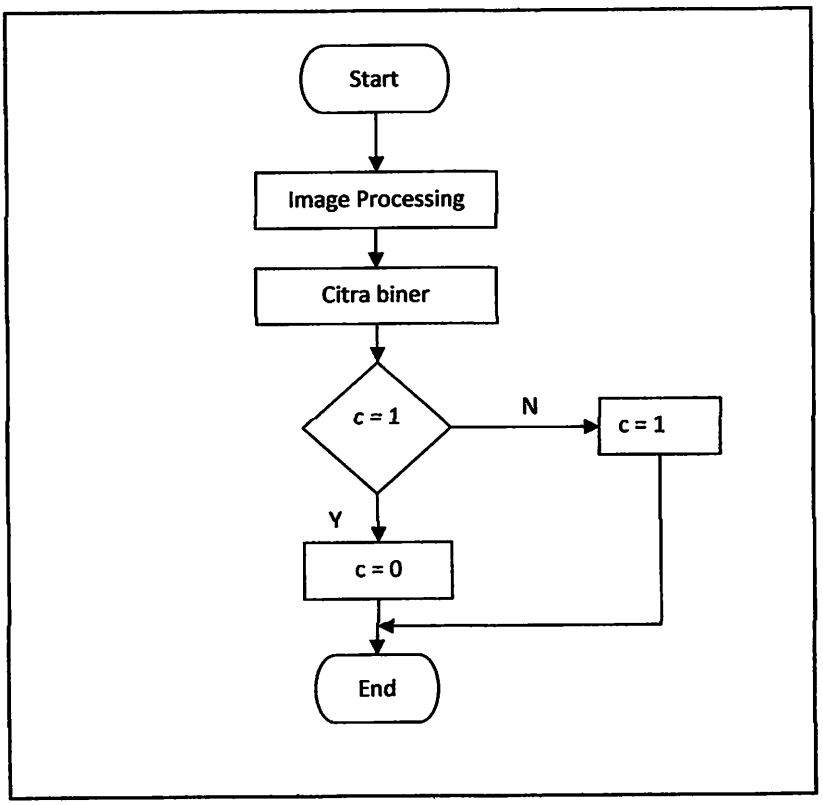
Jadi pada awalnya program akan melakukan pendeteksian nilai level threshold yang diinginkan. Kemudian dilakukan proses konversi pada gambar menjadi hitam-putih. Untuk menghilangkan noise/bluring pada citra barcode disini digunakan erosi dengan nilai-1, diberi nilai-1 agar citra asli barcode tidak hilang. Setelah proses scan line pada citra barcode program akan meng-*Invert(membalik)* dari citra [1,0] menjadi [0,1].

Operator yang digunakan pada proses diatas adalah menggunakan *image acquisition toolbox* yang disediakan oleh MATLAB 7.0.4. *Toolbox image acquisition* adalah kumpulan fungsi yang memperluas kemampuan MATLAB. Untuk program yang akan dibangun seperti fungsi *videoinput* digunakan untuk menginisialisasi objek *video* dari kamera yang terhubung ke PC secara langsung. Kemudian fungsi *preview* digunakan untuk menampilkan *video* secara langsung dari monitor. Fungsi *getsnapshot* juga digunakan pada program ini untuk membaca citra dari kamera dan menempatkannya di *workspace*. Atau fungsi *Pathname* yang digunakan untuk mecari file yang ada di dalam hardisk.

3.2.3 Inverting

Setelah keluaran citra berupa citra biner maka langkah selanjutnya adalah proses inverting, dimana mengubah nilai citra biner yang bernilai 1 menjadi 0, dan yang bernilai 0 diubah menjadi 1. Karena pada saat proses mengubah gambar menjadi citra biner, citra yang berwarna putih bernilai 1 dan yang hitam bernilai 0. Disini dilakukan proses inverting karena kode barcode yang berwarna hitam

nilainya harus 1 agar dapat dilakukan proses selanjutnya. Untuk diagram alir proses inverting dapat dilihat seperti pada gambar berikut:



Gambar 3.3 Diagram alir proses *Inverting*[9]

3.2.4 Compression

Untuk proses kompresi disini digunakan untuk mencari sample pembagi yang nantinya pembaginya tersebut akan dibuat sample pada bar-bar yang lain. Jadi pada proses ini kita hanya mengubah sekala data hasil scan line pada guard bar menjadi 1bit.

3.2.5 Segmentation

Proses segmentasi adalah proses pembagian kode-kode biner menjadi 7bit dari data hasil kompresi. Untuk mempermudah proses konversi 7bit kode biner menjadi 1bit kode desimal. Lebih jelasnya dapat dilihat pada tabel berikut:

Tabel 3.1 data segmentasi 7bit

Digit	L-code	G-code	R-code
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

Pada kolom barcode sebelah kiri terdapat dua jenis kode L dan G, kombinasi antara keduanya digunakan untuk mencari digit desimal pertama barcode. Karena digit pertama barcode tidak terdapat kode bar. Kemudian pada kolom barcode sebelah kanan hanya terdapat 1 jenis kode saja, yaitu R.

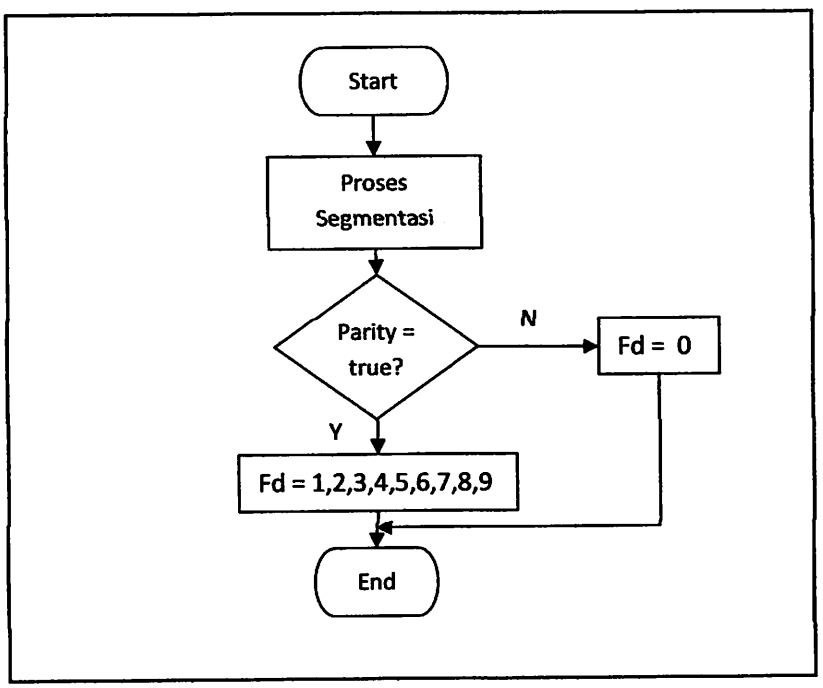
3.2.6 Parity Code

Parity code disini digunakan untuk mencari digit pertama decimal pada barcode. Jadi ada kombinasi-kombinasi tertentu untuk mengetahui digit desimal pertama. Digit pertama iniSeperti tertera pada tabel berikut :

Tabel 3.2 *Parity Code*

First digit	First group of 6 digits	Last group of 6 digits
0	LLLLLL	RRRRRR
1	LLGLGG	RRRRRR
2	LLGGLG	RRRRRR
3	LLGGGL	RRRRRR
4	LGLLGG	RRRRRR
5	LGGLLG	RRRRRR
6	LGGGLL	RRRRRR
7	LGLGLG	RRRRRR
8	LGLGGL	RRRRRR
9	LGGLGL	RRRRRR

Diagram alir untuk mencari *parity code.* Terlihat seperti alir diagram barikut :

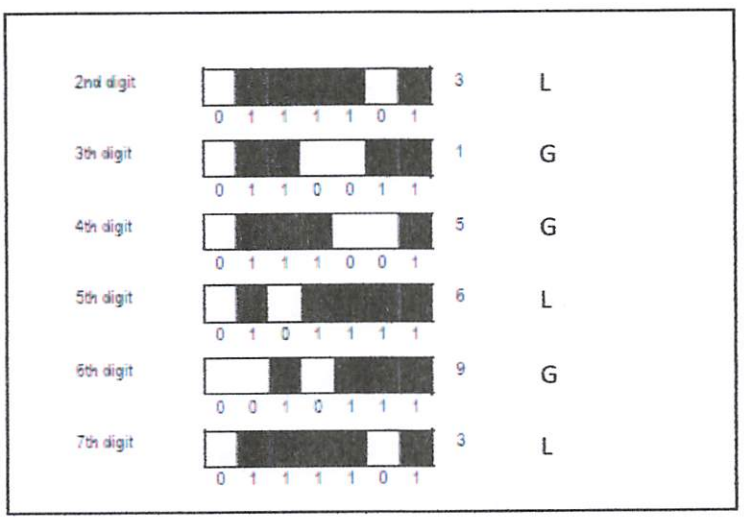


Gambar 3.4 Diagram alir mencari *Parity code.*

Seperti yang digambarkan pada diagram alir diatas, untuk mencari digit desimal pertama pada barcode kita hanya cukup menganalisa kolom sebelah kiri

saja atau menganalisa ke-6 digit pertama jika kombinasi L dan G sesuai (lihat pada tabel.3.2) maka program akan menampilkan digit desimal pertama barcode pada layar.

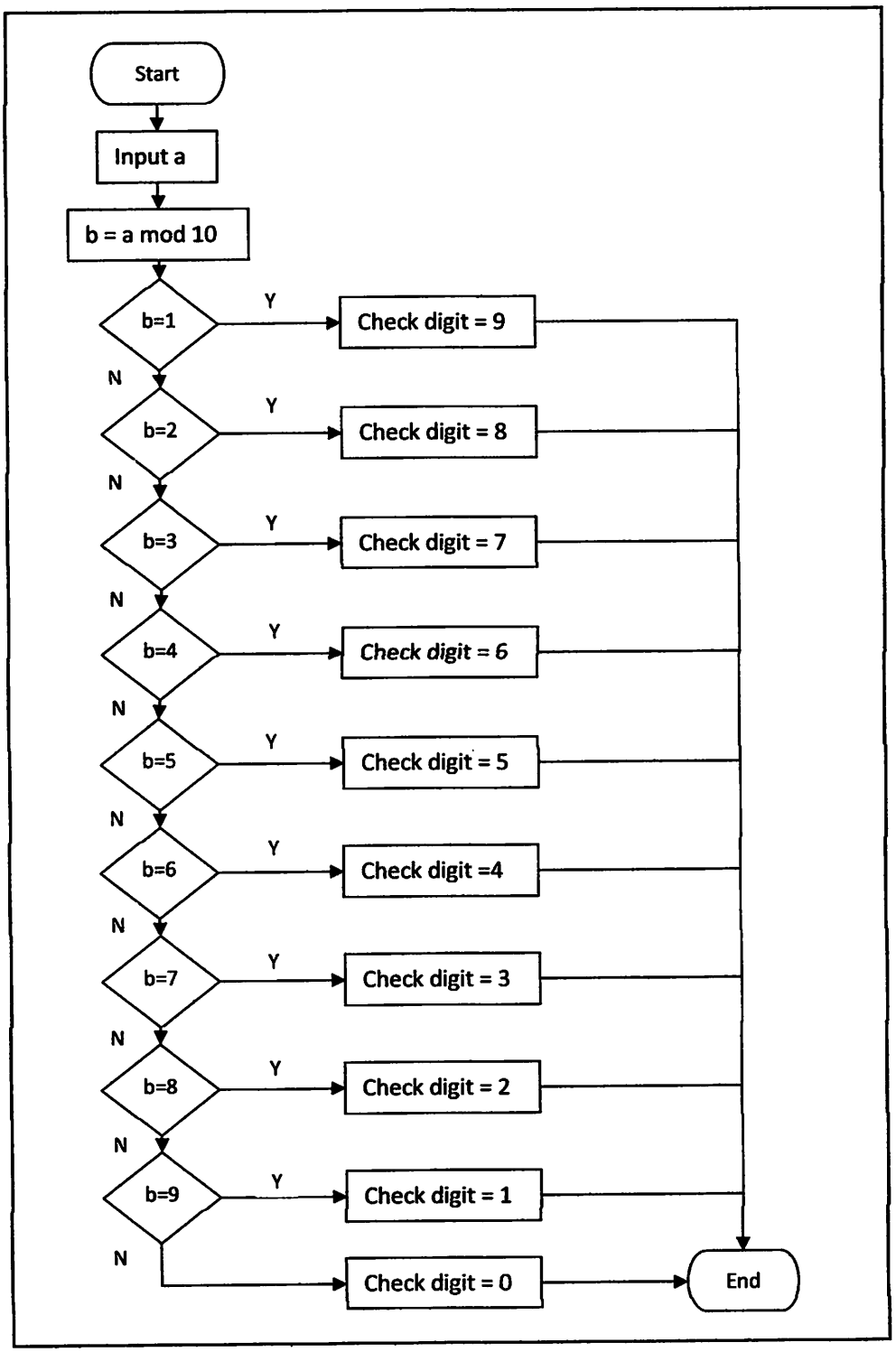
Sebagai permisalan jika 6 digit dari kolom pertama dengan kombinasi parity seperti gambar dibawah maka digit desimal pertama adalah 9



Gambar 3.5 Contoh kombinasi *parity code*[5]

3.2.7 Check Digit

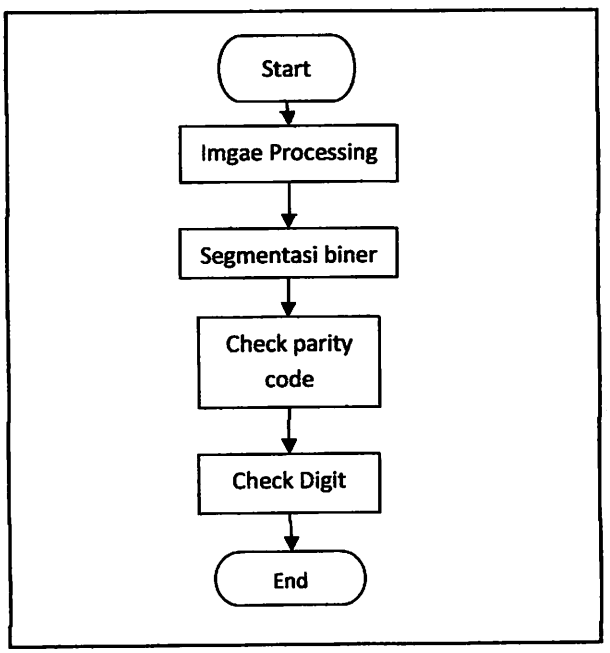
Setelah algoritma mencari digit pertama pada barcode, langkah selanjutnya program akan melakukan pengecekan kode digit terakhir pada barcode. Proses *Check digit* disini bergfungsi untuk mengecek semua digit-digit pada barcode apakah sudah terbaca dengan benar. Berikut ini adalah diagram alir untuk mencari kode digit terakhir pada barcode.



Gambar 3.6 Diagram alir mencari *check digit*

3.2.8 Barcode Recognition

Setelah proses *Inverting*, maka langkah selanjutnya akan dilakukan proses seperti kompresi dan segmentasi data biner hasil scan line. Dan melakukan check pada kombinasi parity code. Dan melakukan proses perhitungan check digit. Untuk lebih terperinci dapat dilihat seperti pada diagram alir di bawah ini :



Gambar 3.7 Diagram Alir Proses *Barcode Recognition*

Setelah proses scan line pada citra barcode maka langkah selanjutnya adalah proses kompresi hasil scan line tersebut. Proses kompresi digunakan untuk mendapatkan nilai terkecil dari barcode. Sebagai acuan adalah *guard bar* sebelah kiri, setelah didapat nilai terkecil maka nilai tersebut akan dibuat sampel untuk bar-bar selanjutnya.

3.3. Desain Graphical User Interface

GUI didesain untuk memfasilitasi sistem operasi program yang interaktif. GUI dapat digunakan untuk pengaturan awal program, menjalankan program, mengakhiri program dan menampilkan hasil dari kalkulasi dalam program.

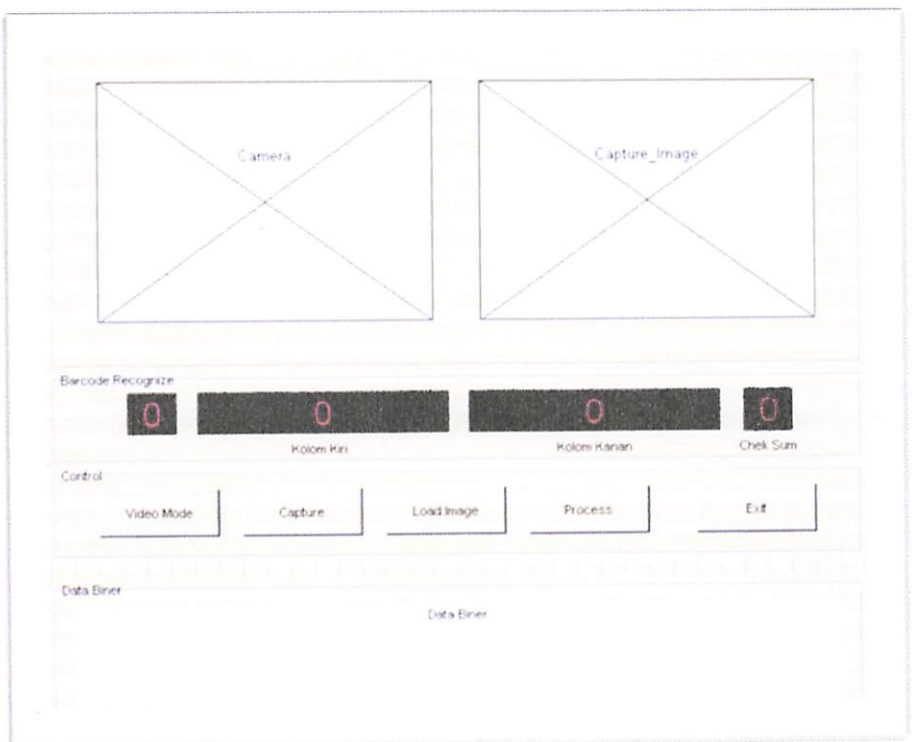
Untuk memulai program user harus menginputkan gambar dengan cara menekan tombol *Load Image* setelah ditekan maka program akan menampilkan window *Open Dialog* yang nanti kita di suruh untuk memilih file gambar barcode mana yang ingin digunakan sebagai inputan. Setelah di tekan tombol *Open* pada *Opend Dialog* maka secaraotomatis program akan menampilkan gambar yang diinginkan tersebut ke dalam layar.

Di dalam GUI program *Pembaca Barcode*, terdapat beberapa komponen yang digunakan, yaitu *button, axes, push button, static text, edit text.* Berikut ini perencanaan tombol-tombol yang akan digunakan dalam program *Pembaca Barcode* :

- *Video Mode* : untuk menampilkan cita yang ditangkap oleh kamera.
- *Capture* : untuk mengambil gambar secara langsng dari kamera.
- *Load Image* : untuk mencari file gambar yangtelah disimpan.
- *Process* : untuk memproses gambar menjadi citra biner.
- *Exit* : untuk keluar dari program.
- *Code3* : untuk menampilkan digit pertama barcode.
- *Code1* : untuk menampilkan kode digit barcode pada kolom kiri.
- *Code2* : untuk menampilkan kode digit barcode pada kolom kanan.
- *Check Sum* : untuk menampilkan nilai hasil perhitungan *chek sum*

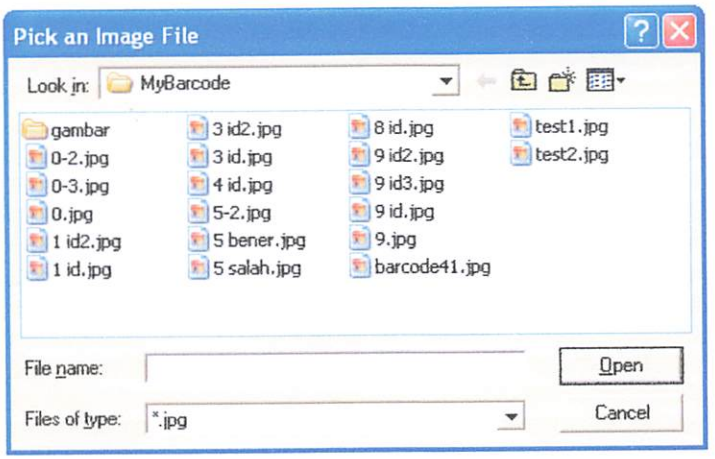
- *Data Biner* : untuk menampilkan data biner hasil *scan line*.
- *Edit Text* : untuk menampilkan kode tiap digit.
- *Axes1* : untuk menampilkan citra dari kamera secara live.
- *Axes2* : untuk menampilkan gambar hasil *searching*.

Berikut ini adalah rancangan tampilan dan tata letak komponen yang akan dibuat :



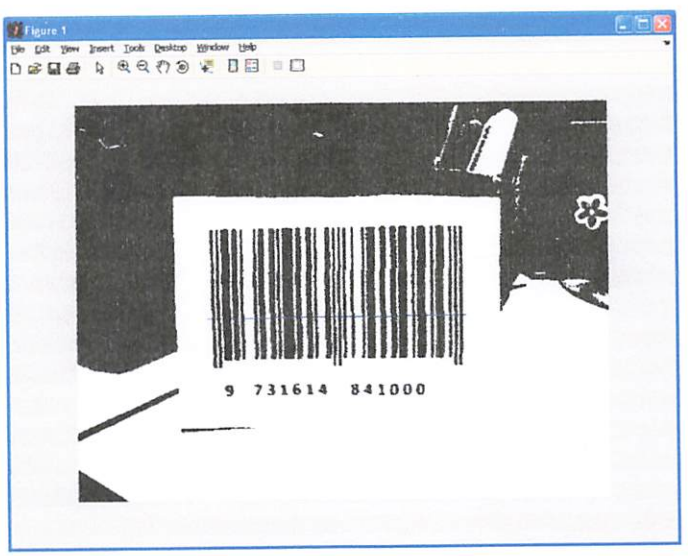
Gambar 3.8 Rencana tampilan program *Pembaca Barcode*

Untuk proses awalnya kita lakukan penekanan pada tombol *Load Image*. Ketika tombol *Load Image* di tekan maka akan memanggil fungsi *Open Dialog*. Seperti terlihat pada gambar berikut

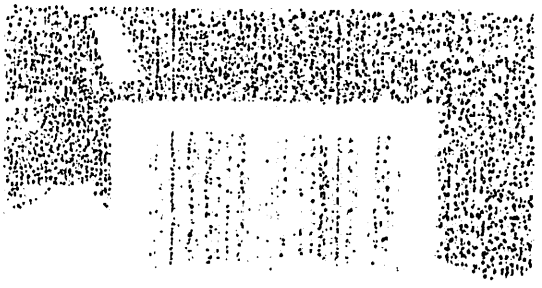


Gambar 3.9 Tampilan *Open Dialog*

Setelah itu langkah selanjutnya adalah melakukan proses pengolahan citra gambar yang telah dipilih dengan cara kita tekan pada tombol *Process*. Fungsi dalam tombol ini adalah mengubah citra *RGB* menjadi cita *Biner*. Untuk mengubah citra biner dilakukan proses seperti proses *Thresholding, Erosi,* dan *Black-White*. Kemudian hasil pengolahan citra akan di tampilkan dalam *Figure*. Seperti gambar dibawah ini:



Gambar 3.10 Hasi dari proses citra *RGB* menjadi citra *Biner* dan proses *Scanning*



Setelah dari proses cita *RGB* menjadi citra *Biner*. Langkah berikutnya kita akan melakukan *scan line* pada barcode secara horizontal. Didalam proses *scan line* terdapat fungsi-fungsi yang harus dijalankan, seperti fungsi *Compressing, Segmentation,* mengecek kebenaran dari kombinasi *Parity Code,* dan fungsi perhitungan dari *Check Digit.* Setelah fungsi dan perhitungan tersebut selesai maka semua data dari hasil pemrosesan tersebut akan di tampilkan pada layar program.

BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

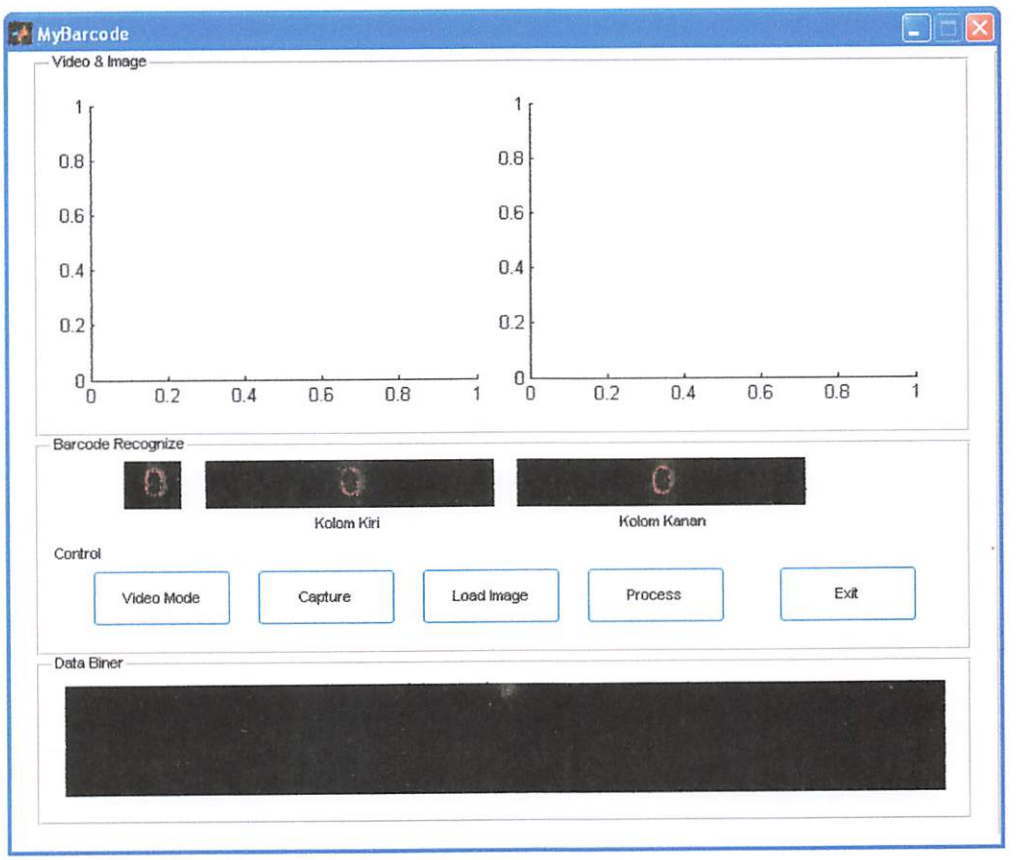
Implementasi dilakukan dengan menerapkan hasil desain yang telah dibuat kedalam bahasa pemrograman (*Coding*) Matlab 7.0.4, sehingga prosedur-prosedur yang telah dibuat dapat dimengerti oleh mesin dan menghasilkan keluaran seperti apa yang diharapkan. Berikut ini adalah perlengkapan yang digunakan dalam implementasi sistem :

Tabel 4.1 Spesifikasi Perlengkapan Implementasi

NO	Perlengkapan	Spesifikasi	Keterangan
1	*Software*	Sistem Operasi	Windows XP SP3
		Bahasa Pemrograman	Matlab 7.04
2	PC	*Processor*	Dual Core 1.6 GHz
		Memori	1,00 GB DDR2
		Hard disk	160 GB
4	Notebook	*Processor*	Core 2 Duo 2.0 GHz
		Memori	1,00 GB DDR2
		Hard disk	160 GB
5	*Digital camera* FinePix	Resolusi	Ture 4.0 Mega Pixels
		Zoom	6x optical zoom
		Lensa	6.36 mm
		Transfer data	USB 2.0 Port

4.1.1 Tampilan Program *Pembaca Barcode*

Pada program *pembaca barcode* ini, akan terdapat beberapa komponen yang digunakan untuk mempermudah pengguna menggunakan program *pembaca kode barcode* ini.



Gambar 4.1 Tampilan GUI Program *pembaca barcode*

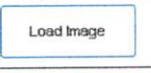
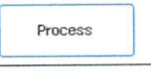
Pada program ini terdapat satu file *.m yaitu MyBarcode.m dan 3 window interface yaitu window *main*, window *open dialog*, dan window *Figure* untuk proses scan line kode barcode. Di dalam program ini terdapat fungsi-fungsi dan perhitungan dalam pembacaan kode barcode.

Cara kerja program *pembaca barcode* ini adalah, pertama-tama user akan melakukan proses input gambar. Selanjutnya image yang berupa format *RGB*

diubah menjadi citra *grayscale*. Dan kemudian di roses lagi menjadi citra biner (hitam dan putih) berikut ini adalah listing program untuk mengolah gambar dari citra *RGB* menjadi citra biner. Kemudian user akan melakukan proses *scan line* secara horizontal pada barcode. Nilai citra biner hasil *scan line* akan di invert, yaitu mengubah nilai [1,0] menjadi [0,1]. Kemudian dilakukan proses kompresi, segmentasi, kemudian menentukan kombinasi dari *parity code*. Setelah ditemukan kode digit pertama barcode maka akan dilakukan proses *check digit*.

4.1.2 Komponen pada Program *Pembaca Barcode*

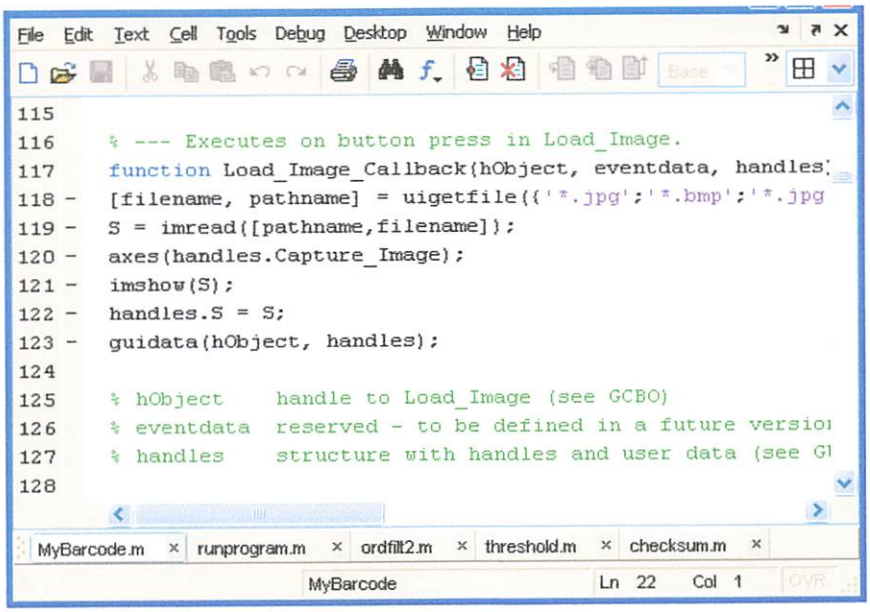
Dalam program ini terdapat beberapa komponen-komponen yang telah disediakan oleh Matlab 7.0.4. Berikut ini adalah komponen-komponen yang ada pada program *pembaca barcode* :

No.	Komponen	Gambar
1	Push button *Load Image*	
2	Push button Proses	
3	Push button *Exit*	

Berikut ini cara kerja tiap komponen pada program *pembaca barcode:*

- Push button : *Load Image*

  Berikut ini adalah listing program untuk searching file gambar yang telah disimpan yang digunakan untuk inputan :



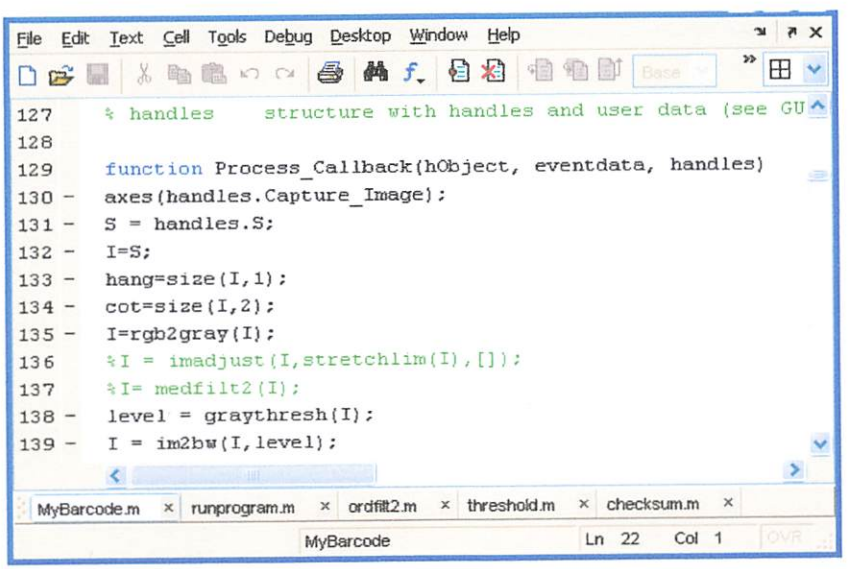
Gambar 4.2 Listing program mencari file gambar

- Push button : *Process*

Pada tombol *Process* ada beberapa fungsi yang di jalankan:

- Fungsi mengubah gambar menjadi citra biner

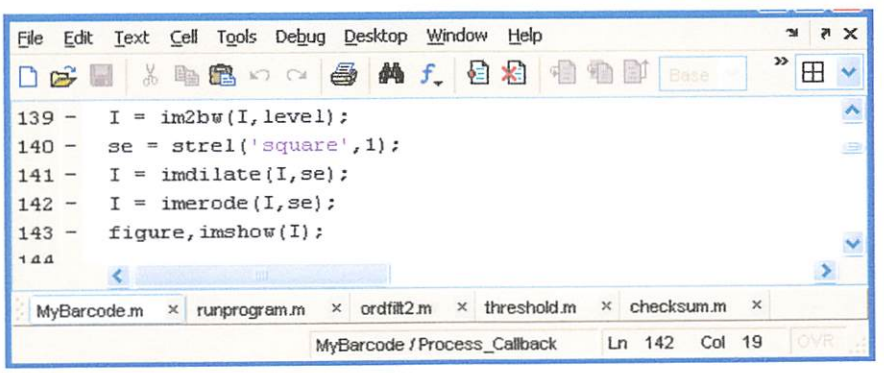
Berikut ini adalah contoh Listing program untuk menampilkan gambar dan memprosesnya menjadi citra biner :



Gambar 4.3 Listing program mengubah citra *RGB* menjadi cita biner

- Fungsi *Erotion*

Setelah proses mengubah gambar menjadi citra biner, maka selanjutnya adalah proses *erotion.* Proses ini digunakan untuk memfilter garis tepi ditiap kode bar. Proses ini dilakukan agar gambar yang dihasilkan terlihat jelas. Berikut ini adalah penambahan listing program untuk proses erosi.

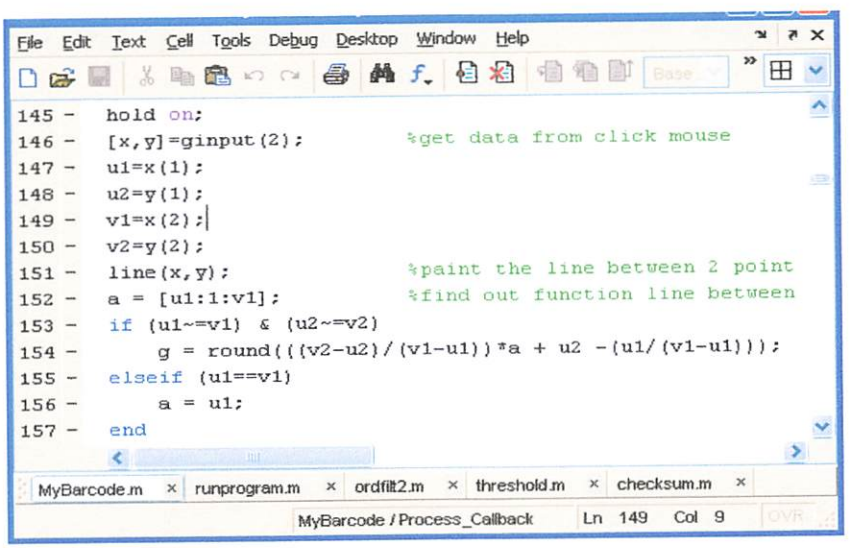


Gambar 4.4 Listing program untuk memfilter kode bar

Setelah proses filtering pada kode bar, user akan melakukan *scan line* secara horizontal pada barcode.

- Fungsi *scan line*

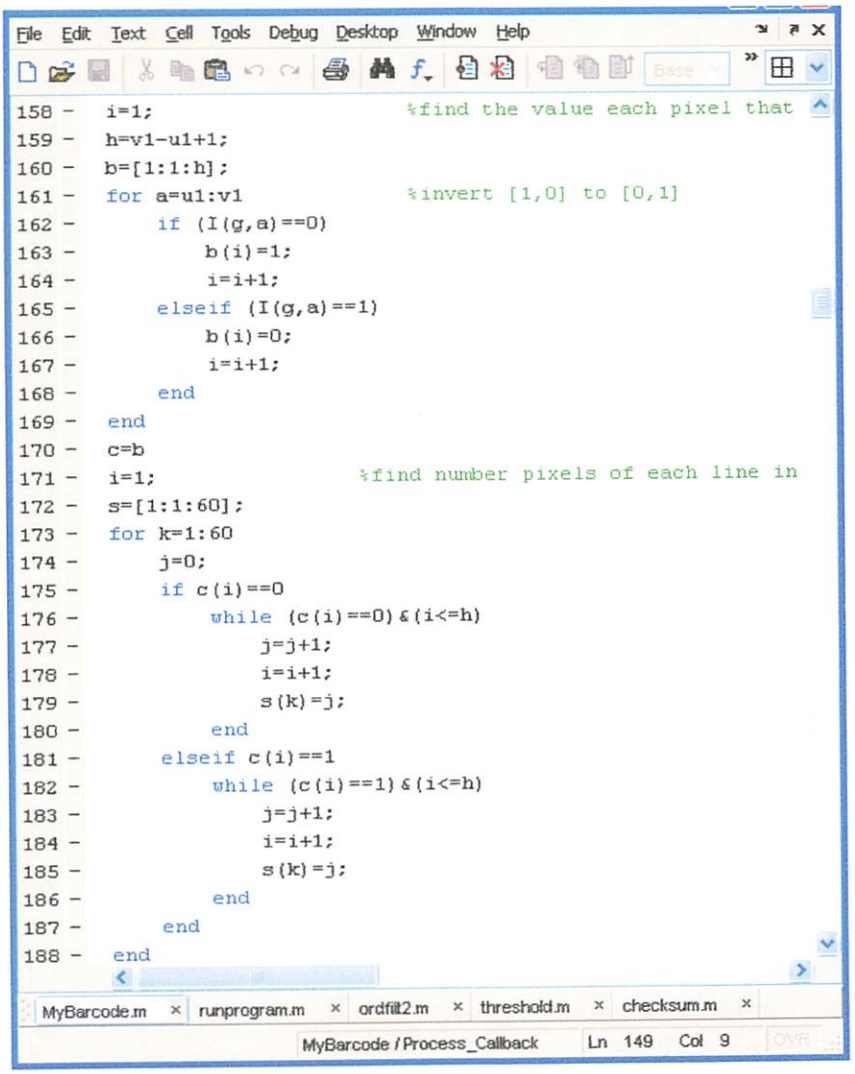
Berikut ini adalah listing program untuk proses *scan line*



Gambar 4.5 Listing program untuk proses *scan line*

- Proses *Inverting*

Proses inverting ini bertujuan untuk membalik nilai citra biner [1,0] nenjadi [0,1]. Berikut adalah listing program proses inverting dan mencari nilai ditiap baris bar.

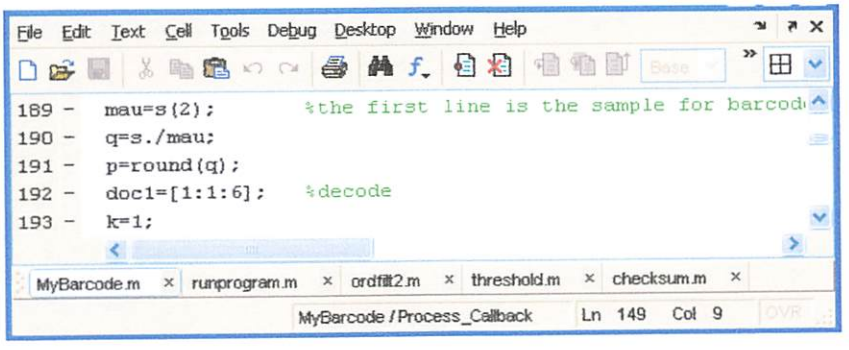


Gambar 4.6 Listing program proses *inverting* dan mencari nilai di tiap garis bar

- Proses *Compression*

Pada proses ini program akan melakukan inisialisasi pada *guard bar*. Disitu program akan melakukan proses pembagian nilai biner pada *guard bar* untuk di kompres menjadi 1 bit biner saja. Yang nantinya pembaginya

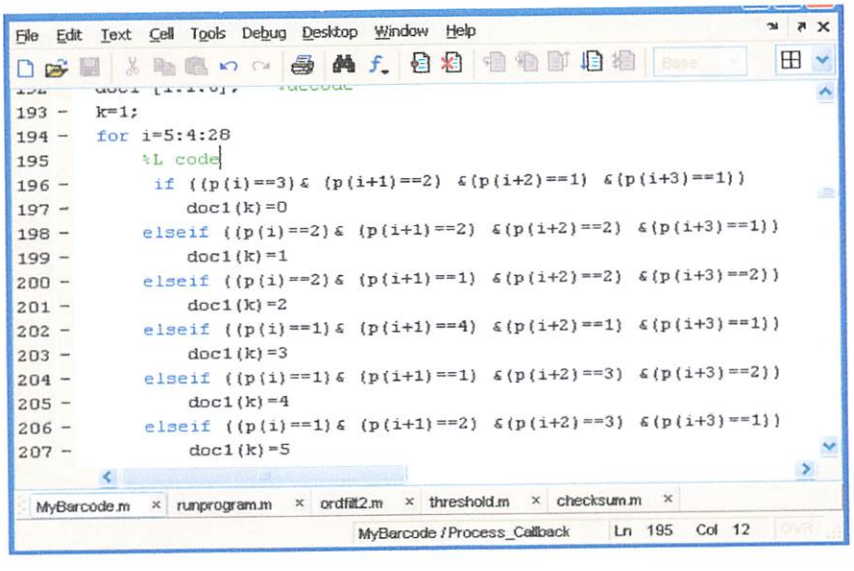
akan dibut sample untuk membagi biner pada bar-bar lainnya. Jika sisa hasil pembagian sama dengan setengah atau lebih dari setengah dari nilai pembaginya, maka program tersebut akan tetap menghitung, jika sisa pembaginya lebih kecil dari setengah dari nilai pembaginya maka sisa tersebut tidak akan dihitung. Dengan menggunakan fungsi *round* program akan mencari nilai yang paling mendekati nilai pembaginya. Berikut ini adalah listing programnya.



Gambar 4.7 Listing program proses *compression*

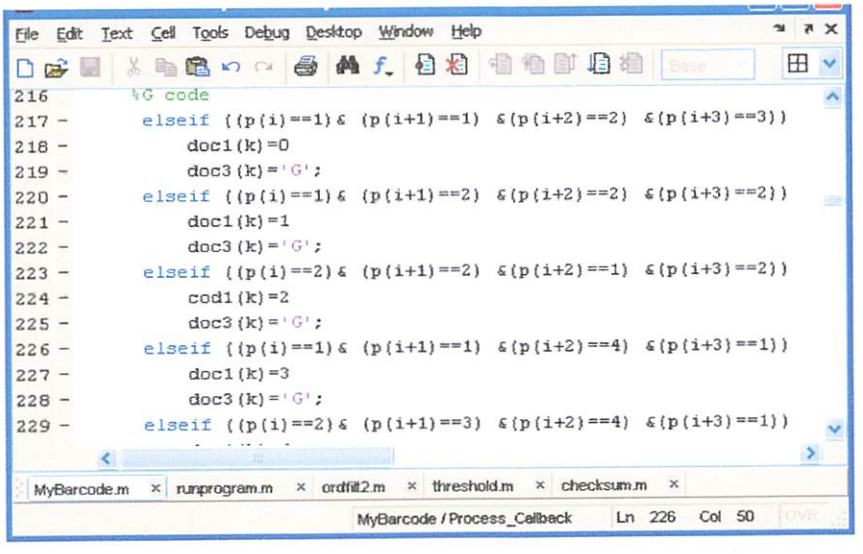
- Proses segmentasi

berikut ini adalah kombinasi dari kode L



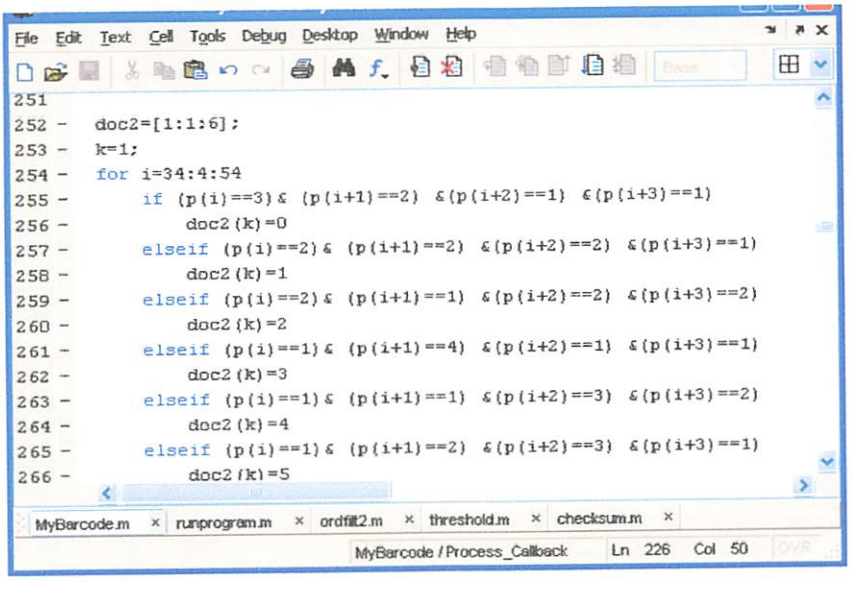
Gambar 4.8 kombinasi data dari kode L

Berikut adala kombinasi data dari kode G



Gambar 4.9 kombinasi data dari kode G

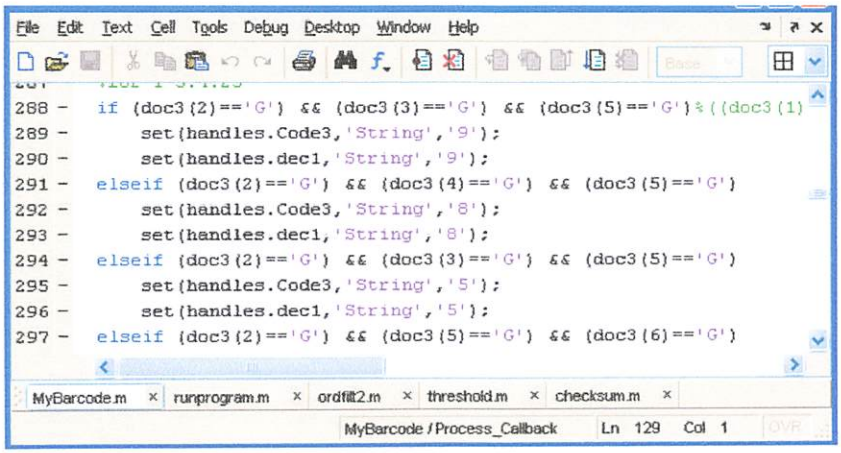
Berikut ini adalah kombinasi data dari kode R



Gambar 4.10 kombinasi data dari kode R

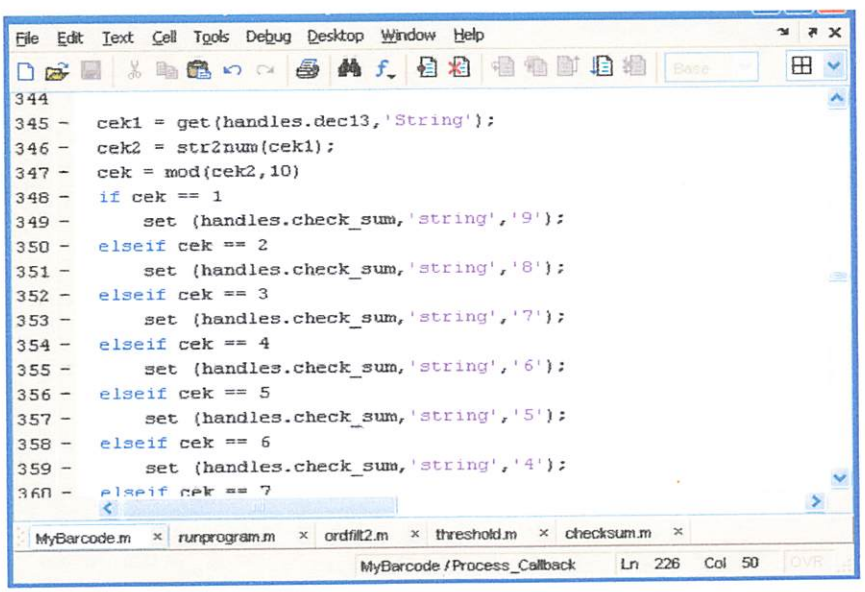
- Proses mencari digit pertama

  Berikut ini adalah listing program untuk mencari kode digit pertama pada barcode :



Gambar 4.11 Listing Program untuk mencari digit pertama barcode

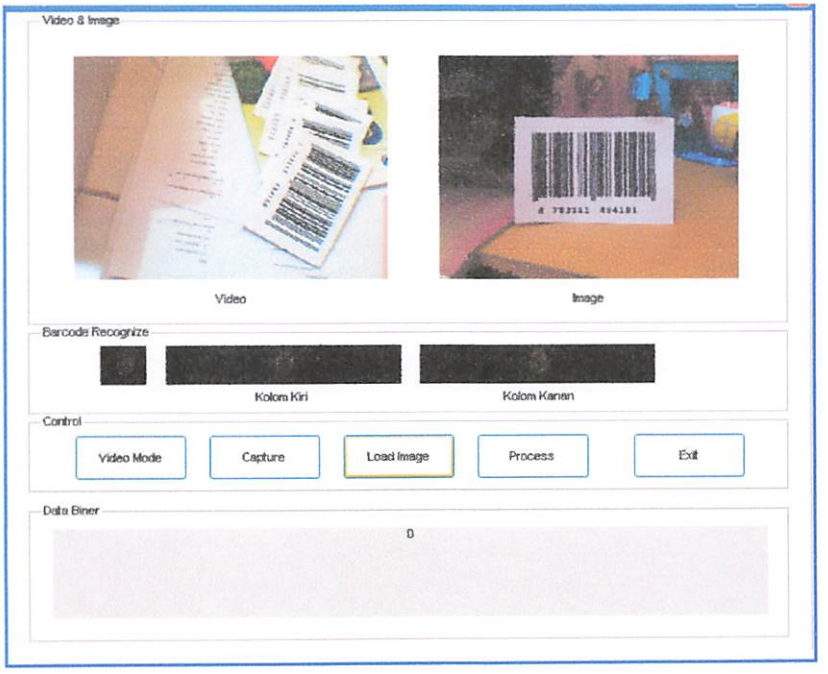
- Proses perhitungan *check digit*



Gambar 4.12 Listing perhitungan *check digit*

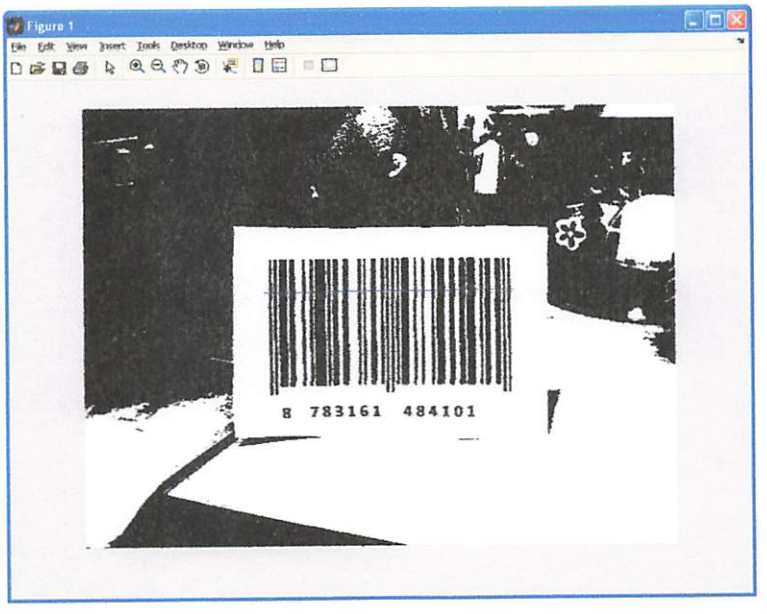
4.1.3 Hasil Implementasi

Berikut ini adalah hasil implementasi program *pembaca barcode* yang telah di implementasikan. berikut adalah tampilan pada saat proses input gambar di jalankan.

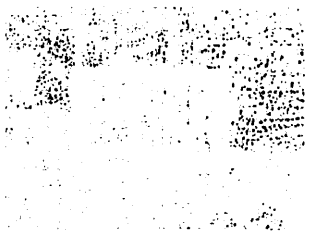


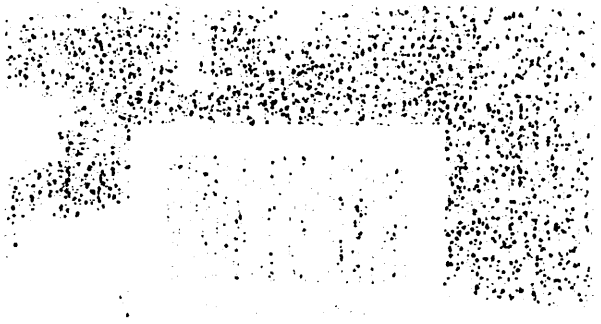
Gambar 4.13 Tampilan pada saat proses input gambar

Setelah menginputkan gambar, langkah selanjutnya adalah melakukan proses pengolahan gambar dan scan line, berikut ini adalah tampilan ketika tombol *process* ditekan sekaligus proses *scan line* :

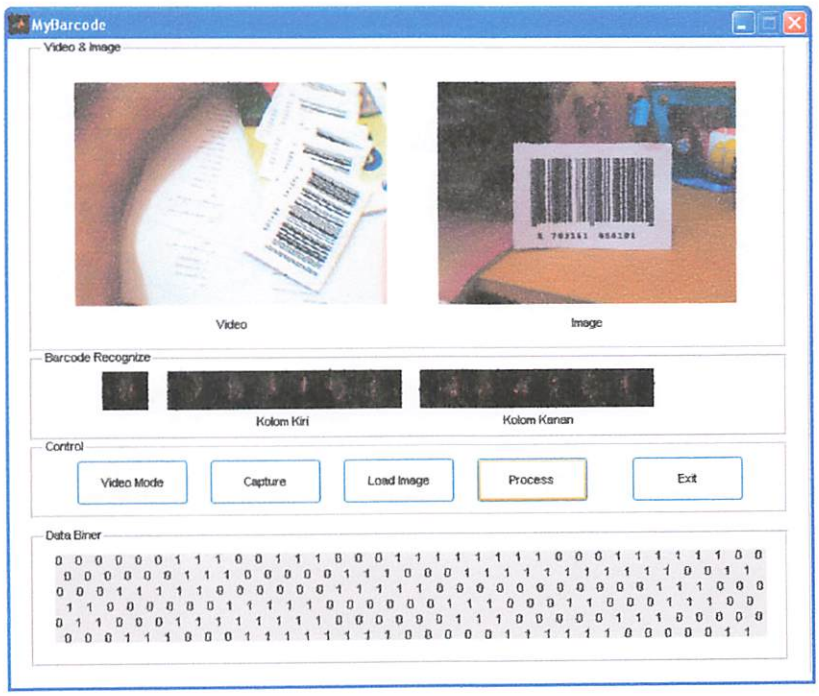


Gambar 4.14 Tampilan hasil *Image processing* dan proses *scan line*





Pada saat proses scan line, gambar yang terscan akan memberikan data biner. berikut ini adalah data biner hasil dari proses scan line:



Gambar 4.15 Tampilan setelah proses scan line

4.2. Pengujian Sistem

Pengujian ini dilakukan untuk mengetahui kinerja program pembaca barcode beserta algoritma yang digunakan serta pengujian pada suatu kondisi yang dapat berdampak negatif pada saat proses pembacaan barcode.

Dalam pengujian ini dilakukan suatu langkah untuk menguji program pembaca barcode dengan beberapa kondisi. Pada pengujian ini dilakukan dengan 7 faktor. Dari ke-7 faktor tersebut dapat dilihat pada tabel di bawah ini :

Tabel 4.1 Kondisi barcode dengan beberapa faktor dan kondisi

No.	KONDISI	CITRA
1.	Malam dengan cahaya	
2.	Siang dengan cahaya	
3.	Siang tanpa cahaya	
4.	Warna biru muda	
5.	Warna biru gelap	
6.	Warna ukuran 640x480 Pixel	

7.	Rotasi 15 derajat	
8.	Rotasi 10 derajat	
9.	Resolusi citra 800x600	
10.	Resolusi cita 640x480	
11.	Jarak kamera dengan objek 25cm	
12.	Jarak kamera dengan objek 30 cm	
13.	Pola barcode cembung	

Data hasil pengujian pembacaan barcode dapat dilihat pada tabel dibawah ini:

Tabel 4.3 Data pengujian sensitifitas algoritma pembaca barcode

No.	Kondisi	faktor	Hasil Deteksi	
			berhasil	Tidak berhasil
1	Malam dengan cahaya	Cahaya	√	-
2	Siang dengan cahaya	Cahaya	√	-
3	Siang tanpa cahaya	Cahaya	√	-
4	Warna biru muda	Warna	-	√
5	Warna biru tua	warna	-	√
6	Warna ukuran 640x480	Ukuran	√	-
7	Rotasi 15 derajat	Rotasi	-	√
8	Rotasi 10 derajat	Rotasi	√	-
9	Resolusi citra 800x600	Resolusi	-	√
10	Resolusi cita 640x480	Resolusi	√	-
11	Jarak kamera 25cm	jarak	√	-
12	Jarak kamera 30cm	jarak	-	√
13	*Pola barcode cembung*	Pola	√	-

Tabel 4.3 menunjukkan bahwa pada pengujian sensitifitas terhadap faktor-faktor diatas dapat disimpulkan bahwa pada kondisi malam dan siang dengan menggunakan cahaya lampu TL 15 watt, atau kondisi siang tanpa cahaya lampu kode barcode dengan ukuran 320x240 tetap dapat terdeteksi dengan baik. Sedangkan untuk faktor warna pada kode bar dengan ukuran gambar dengan ukuran 320x240 pixel tidak dapat terdeteksi. Akan tetapi jika pada faktor warna, dan citra asli kode bar di perbesar menjadi ukuran 640x480 pixel, program akan

dapat membaca kode barcode dengan baik. pada faktor rotasi, agar kode barcode tetap terbaca kemiringan kode barcode harus berada pada 0 derajat sampai 10 derajat, jika lebih dari 10 derajat kode barcode tidak dapat terbaca. Untuk faktor resolusi, maksimal ukuran resolusi citra yang diambil adalah 640x480 pixel dan minimal ukuran 448x336. Sedangkan pada faktor jarak, agar kode barcode tetap dapat terdeteksi, Maka jarak yang idealnya untuk jarak antara kamera dengan objek adalah berkisar antara 25 cm. Sedangkan untuk jarak maksimalnya adalah 27 cm dan minimal 20 cm. Kemudian untuk pola bidang barcode yang berkondisi cembung juga mempengaruhi proses pembacaan kode barcode.

BAB V

PENUTUP

Dari beberapa uraian yang telah dikemukakan pada bab-bab sebelumnya dapat diambil kesimpulan dan saran sebagai berikut :

5.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan pada proyek akhir ini, maka disimpulkan bahwa :

1. Pola rotasi dan kemiringan barcode mempengaruhi akurasi pembacaan.
2. Kualitas citra asli barcode akan sangat berpengaruh ketika pada saat waktu proses pengolahan gambar.
3. Pencahayaan yang berbeda dapat mempengaruhi proses pembacaan barcode.
4. Selain pencahayaan, jarak antara subjek denga kamera juga mempengaruhi tingkat keberhasilan pembacaan barcode.
5. Tingkat ketajaman warna pada kode bar akan sangat berpengaruh dalam pembacaan kode barcode. Maka dari itu untuk menangani masalah itu gambar asli kode barcode harus diperbesar.
6. Ukuran besar kecilnya resolusi dan besar kecilnya citra asli barcode juga mempengaruhi proses pembacaan kode barcode.
7. Bidang pada barcode juga berpengaruh pada proses pembacaan.

5.2 Saran

1. Untuk pengembanganya dapat menggunakan algoritma yang lebih handal agar proses pembacaan barcode lebih akurat.
2. Disarankan untuk menggunakan kamera yang berukuran 5 mega pixel atau yang lebih canggih agar didapat hasil citra yang lebih bagus.
3. Untuk pengembangan system berikutnya sebaiknya ditambahkan fitur-fitur lain, seperti data hasil proses scaning kode barcode.
4. Sebagai implementasi selanjutnya diharapkan system dapat mengenali citra barcode secara real time dengan tingkat akurasi yang tinggi.

DAFTAR PUSTAKA

[1]. Abdia Gunaidi, "*Matlab Programming*", Informatika Bandung, 2010.

[2]. QualitySoft, "*bcTester barcode recognition.*" [Online]. Available: http://www.qualitysoft.de/index_en.html

[3]. Darma P., September 2008 "*Sistem Biometrika*", Mengwitani.

[4]. Barcode Indonesia "*Linear Image ungguli Laser Barcode*" [Online]. Available : http://blog.fastechindo.com/

[5]. D. Chai and F. Hock, "Locating and decoding EAN-13 barcodes from images captured by digital cameras," in *Information, Communications and Signal Processing, 2005 Fifth International Conference on, 2005*, pp. 1556-159 [Online]. Available: www.icics.org/2005/download/

[6]. Barcode Indonesia "*Definisi dan Sejarah Barcode*" [Online]. Available : http://blog.fastechindo.com/?p=5

[7]. Wikipedia "*European Article Number*" [Online]. Available : http://en.wikipedia.org/wiki/European_Article_Number

[8]. Mathwork, "*Dilation, erotion, and the morphological gradient*", [Online] Available : http://blogs.mathworks.com/steve/2006/09/25/dilation-erosion-and-the-morphological-gradient

[9]Software practice, Barcode Image Recognition, [Online] Available : http://www.softwarepractice.org/wiki/Barcode_Image_Recognition

[10]. EAN-13 Parity Pattern, "*EAN-13 check digitcalculation – product coding, bar codes*", [Online]. Available : http://www.export911.com/e911/coding/eanParity.htm

L A M P I R A N

```
function varargout = MyBarcode(varargin)
% MYBARCODE M-file for MyBarcode.fig
%      MYBARCODE, by itself, creates a new MYBARCODE or raises the existing
%      singleton*.
%
%      H = MYBARCODE returns the handle to a new MYBARCODE or the handle to
%      the existing singleton*.
%
%      MYBARCODE('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in MYBARCODE.M with the given input arguments.
%
%      MYBARCODE('Property','Value',...) creates a new MYBARCODE or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before MyBarcode_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to MyBarcode_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 MyBarcode
% Last Modified by GUIDE v2.5 18-Aug-2010 22:41:48
% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @MyBarcode_OpeningFcn, ...
                   'gui_OutputFcn',  @MyBarcode_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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

% End initialization code - DO NOT EDIT
 % --- Executes just before MyBarcode is made visible.

function MyBarcode_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 MyBarcode (see VARARGIN)
% Choose default command line output for MyBarcode
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
```

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

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

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : Fitroh Amaluddin
NIM : 06.12.911
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Judul Skripsi : Pengembangan Aplikasi Sistem Pembaca Barcode Berbasis Citra Kamera

Dipertahankan dihadapan tim penguji skripsi jenjang Strata Satu (S-1) pada:
Hari : Selasa
Tanggal : *24 Agustus 2010*
Dengan Nilai : 87.25 (A) BY

PANITIA UJIAN SKRIPSI

KETUA

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

ANGGOTA PENGUJI

Dosen Penguji I

Joseph Dedy Irawan, ST.MT.
NIP. 19740416 200501 1 002

Dosen Penguji II

Sandy Nataly Mantja, SKom.

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata satu (S-1) Jurusan Teknik Elektro konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : FITROH AMALUDDIN
NIM : 06.12.911
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer dan Informatika
Masa Bimbingan : 19 Mei 2010 s/d 19 November 2010
Judul Skripsi : Pengembangan Aplikasi Sistem Pembaca Barcode Berbasis Citra Kamera

Tanggal	Uraian	Paraf
Penguji II 24 Agustus 2010	1. Sesuai Kesimpulan, Jelaskan di BAB IV pengujian pada cahaya sekian, resolusi sekian,kemiringan sekian, hasilnya seperti apa	

Mengetahui,
Dosen Pembimbing I

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

Dosen Penguji,
PENGUJI II

Sandy Nataly Mantja, SKom.

INSTITUT TEKNOLOGI NASIONAL
Jl. Bendungan Sigura – gura No.2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Fitroh Amaluddin
Nim : 06.12.911
Masa Bimbingan : 19-Mei-2010 s/d 19-November-2010
Judul Skripsi : Pengembangan Aplikasi Sistem Pembaca Barcode Berbasis Kamera

No	Tanggal	Uraian	Paraf Pembimbing
1	12/07 10	REVISI PROGRAM	
2	20/07	ACC PROGRAM	
3	/07 10	REVISI MAKALAH	
4	24/07 10	ACC MAKALAH SEMINAR HASIL	
5	02/08 10	ACC BAB I, II	
6	08/08 10	REVISI BAB III	
7	10/8/10	ACC BAB III & REVISI BAB IV	
8	20/8/10	ACC BAB IV & V	
9			
10			

Malang,
Dosen pembimbing

DR. Eng. Aryuanto S, ST MT
NIP.Y.1030800417

Form S-4b

```
% UIWAIT makes MyBarcode wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
%variables for design

icon1=imread('stream.png');
icon2=imread('capture.png');
icon3=imread('search.png');
icon4=imread('scan.png');
icon5=imread('exit.png');
set(handles.StartCamera,'CData',icon1)
set(handles.Capture,'CData',icon2)
set(handles.Load_Image,'CData',icon3)
set(handles.Process,'CData',icon4)
set(handles.exit,'CData',icon5)

hback = axes('units','normalized','position',[0 0 1 1]);
uistack(hback,'bottom');

[back map]=imread('backbarcode.jpg');
image(back)
colormap(map)


function varargout = MyBarcode_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 Capture.

function StartCamera_Callback(hObject, eventdata, handles)
% hObject    handle to StartCamera (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Start/Stop Camera
global vid;
   cla(handles.Camera,'reset');
   guidata(hObject, handles); %updates the handles
   imaqreset;
   set(gcf,'CurrentAxes',handles.Camera);
   set(gcf,'DoubleBuffer','on');
   imaqhwinfo;
   imaqhwinfo('winvideo',1);
   vid=videoinput('winvideo',1);
   vid.ReturnedColorSpace = 'rgb'
   vidRes = get(vid, 'VideoResolution');
   nBands = get(vid, 'NumberOfBands');
   hImage = image( zeros(vidRes(2), vidRes(1), nBands) );
   preview(vid, hImage);
```

```
% --- Executes on button press in Load_Image.

function Capture_Callback(hObject, eventdata, handles)
 global vid;
   S=getsnapshot(vid);
   axes(handles.Capture_Image);
   imshow(S);
   handles.S = S;
   guidata(hObject, handles);

% hObject    handle to Capture (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 Load_Image.

function Load_Image_Callback(hObject, eventdata, handles)
[filename, pathname] = uigetfile({'*.jpg';'*.bmp';'*.jpg';'*.gif';'*.tif'}, 'Pick an Image File');
S = imread([pathname,filename]);
axes(handles.Capture_Image);
imshow(S);
handles.S = S;
guidata(hObject, handles);

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

function Process_Callback(hObject, eventdata, handles)
axes(handles.Capture_Image);
S = handles.S;
I=S;
hang=size(I,1);
cot=size(I,2);
I=rgb2gray(I);

%I = imadjust(I,stretchlim(I),[]);
%I= medfilt2(I);

level = graythresh(I);
I = im2bw(I,level);
se = strel('square',1);
I = imdilate(I,se);
I = imerode(I,se);
figure,imshow(I);

hold on;
[x,y]=ginput(2);        %get data from click mouse
u1=x(1);
u2=y(1);
v1=x(2);
v2=y(2);
line(x,y);
a = [u1:1:v1];
```

```
if (u1~=v1) & (u2~=v2)
   g = round(((v2-u2)/(v1-u1))*a + u2 -(u1/(v1-u1)));
elseif (u1==v1)
   a = u1;
end
i=1;                %find the value each pixel that the line go through
h=v1-u1+1;
b=[1:1:h];
for a=u1:v1            %invert [1,0] to [0,1]
   if (I(g,a)==0)
      b(i)=1;
      i=i+1;
   elseif (I(g,a)==1)
      b(i)=0;
      i=i+1;
   end
end
c=b
i=1;              %find number pixels of each line in barcode picture;
s=[1:1:60];
for k=1:60
   j=0;
   if c(i)==0
      while (c(i)==0)&(i<=h)
         j=j+1;
         i=i+1;
         s(k)=j;
      end
   elseif c(i)==1
      while (c(i)==1)&(i<=h)
         j=j+1;
         i=i+1;
         s(k)=j;
      end
   end
end
mau=s(2);      %the first line is the sample for barcode,in another line is ratio with this
q=s./mau;
p=round(q);
doc1=[1:1:6];   %decode
k=1;
for i=5:4:28

   %L code
    if ((p(i)==3)& (p(i+1)==2) &(p(i+2)==1) &(p(i+3)==1))
      doc1(k)=0
   elseif ((p(i)==2)& (p(i+1)==2) &(p(i+2)==2) &(p(i+3)==1))
      doc1(k)=1
   elseif ((p(i)==2)& (p(i+1)==1) &(p(i+2)==2) &(p(i+3)==2))
      doc1(k)=2
   elseif ((p(i)==1)& (p(i+1)==4) &(p(i+2)==1) &(p(i+3)==1))
      doc1(k)=3
   elseif ((p(i)==1)& (p(i+1)==1) &(p(i+2)==3) &(p(i+3)==2))
      doc1(k)=4
   elseif ((p(i)==1)& (p(i+1)==2) &(p(i+2)==3) &(p(i+3)==1))
      doc1(k)=5
```

```
  elseif ((p(i)==1)& (p(i+1)==1) &(p(i+2)==1) &(p(i+3)==4))
     doc1(k)=6
  elseif ((p(i)==1)& (p(i+1)==3) &(p(i+2)==1) &(p(i+3)==2))
     doc1(k)=7
  elseif ((p(i)==1)& (p(i+1)==2) &(p(i+2)==1) &(p(i+3)==3))
     doc1(k)=8
  elseif ((p(i)==3)& (p(i+1)==1) &(p(i+2)==1) &(p(i+3)==2))
     doc1(k)=9

 %G code
  elseif ((p(i)==1)& (p(i+1)==1) &(p(i+2)==2) &(p(i+3)==3))
     doc1(k)=0
     doc3(k)='G';
  elseif ((p(i)==1)& (p(i+1)==2) &(p(i+2)==2) &(p(i+3)==2))
     doc1(k)=1
     doc3(k)='G';
  elseif ((p(i)==2)& (p(i+1)==2) &(p(i+2)==1) &(p(i+3)==2))
     cod1(k)=2
     doc3(k)='G';
  elseif ((p(i)==1)& (p(i+1)==1) &(p(i+2)==4) &(p(i+3)==1))
     doc1(k)=3
     doc3(k)='G';
  elseif ((p(i)==2)& (p(i+1)==3) &(p(i+2)==4) &(p(i+3)==1))
     doc1(k)=4
     doc3(k)='G';
  elseif ((p(i)==1)& (p(i+1)==3) &(p(i+2)==2) &(p(i+3)==1))
     doc1(k)=5
     doc3(k)='G';
  elseif ((p(i)==4)& (p(i+1)==1) &(p(i+2)==1) &(p(i+3)==1))
     doc1(k)=6
     doc3(k)='G';
  elseif ((p(i)==2)& (p(i+1)==1) &(p(i+2)==3) &(p(i+3)==1))
     doc1(k)=7
     doc3(k)='G';
  elseif ((p(i)==3)& (p(i+1)==1) &(p(i+2)==2) &(p(i+3)==1))
     doc1(k)=8
     doc3(k)='G';
  elseif ((p(i)==2)& (p(i+1)==1) & (p(i+2)==1) &(p(i+3)==3))
     doc1(k)=9
     doc3(k)='G';
  end
  k=k+1;

end

doc2=[1:1:6];
k=1;
for i=34:4:54
%R code
   if (p(i)==3)& (p(i+1)==2) &(p(i+2)==1) &(p(i+3)==1)
      doc2(k)=0
   elseif (p(i)==2)& (p(i+1)==2) &(p(i+2)==2) &(p(i+3)==1)
      doc2(k)=1
   elseif (p(i)==2)& (p(i+1)==1) &(p(i+2)==2) &(p(i+3)==2)
      doc2(k)=2
```

```
  elseif (p(i)==1)& (p(i+1)==4) &(p(i+2)==1) &(p(i+3)==1)
     doc2(k)=3
  elseif (p(i)==1)& (p(i+1)==1) &(p(i+2)==3) &(p(i+3)==2)
     doc2(k)=4
  elseif (p(i)==1)& (p(i+1)==2) &(p(i+2)==3) &(p(i+3)==1)
     doc2(k)=5
  elseif (p(i)==1)& (p(i+1)==1) &(p(i+2)==1) &(p(i+3)==4)
     doc2(k)=6
  elseif (p(i)==1)& (p(i+1)==3) &(p(i+2)==1) &(p(i+3)==2)
     doc2(k)=7
  elseif (p(i)==1)& (p(i+1)==2) &(p(i+2)==1) &(p(i+3)==3)
     doc2(k)=8
  elseif (p(i)==3)& (p(i+1)==1) &(p(i+2)==1) &(p(i+3)==2)
     doc2(k)=9
  end
  k=k+1;
end
doc1 = num2str(doc1);
doc2 = num2str(doc2);
c = num2str(c);

set(handles.Code1,'String',doc1);
set(handles.Code2,'String',doc2);
set(handles.Data_Biner,'String',c);

if (doc3(2)=='G') && (doc3(3)=='G') && (doc3(5)=='G')
   set(handles.Code3,'String','9');
   set(handles.dec1,'String','9');
elseif (doc3(2)=='G') && (doc3(4)=='G') && (doc3(5)=='G')
   set(handles.Code3,'String','8');
   set(handles.dec1,'String','8');
elseif (doc3(2)=='G') && (doc3(3)=='G') && (doc3(5)=='G')
   set(handles.Code3,'String','5');
   set(handles.dec1,'String','5');
elseif (doc3(2)=='G') && (doc3(5)=='G') && (doc3(6)=='G')
   set(handles.Code3,'String','4');
   set(handles.dec1,'String','4');
elseif (doc3(3)=='G') && (doc3(4)=='G') && (doc3(5)=='G')
   set(handles.Code3,'String','3');
   set(handles.dec1,'String','3');
elseif (doc3(3)=='G')&&(doc3(5)=='G')&&(doc3(6)=='G')
   set(handles.Code3,'String','1');
   set(handles.dec1,'String','1');
else
   set(handles.Code3,'String','0');
   set(handles.dec1,'String','0');
end

fd=get(handles.dec1,'String');

bit1=(str2num(fd)*1);
bit2=(str2num(doc1(1))* 3);  %left colom
bit3=(str2num(doc1(2))* 1);
bit4=(str2num(doc1(3))* 3);
bit5=(str2num(doc1(4))* 1);
```

```
bit6=(str2num(doc1(5))* 3);
bit7=(str2num(doc1(6))* 1);

bit8=(str2num(doc2(1))*3);  %right colom
bit9=(str2num(doc2(2))*1);
bit10=(str2num(doc2(3))*3);
bit11=(str2num(doc2(4))*1);
bit12=(str2num(doc2(5))*3);

bit13=(bit1 + bit2 + bit3 + bit4 + bit5 + bit6 + bit7 + bit8 + bit9 + bit10 + bit11 + bit12);

set (handles.dec2,'String',bit2);
set (handles.dec3,'String',bit3);
set (handles.dec4,'String',bit4);
set (handles.dec5,'String',bit5);
set (handles.dec6,'String',bit6);
set (handles.dec7,'String',bit7);
set (handles.dec8,'String',bit8);
set (handles.dec9,'String',bit9);
set (handles.dec10,'String',bit10);
set (handles.dec11,'String',bit11);
set (handles.dec12,'String',bit12);
set (handles.dec13,'String',bit13);

 cek1 = get(handles.dec13,'String');
cek2 = str2num(cek1);
cek = mod(cek2,10)
if cek == 1
   set (handles.check_sum,'string','9');
elseif cek == 2
   set (handles.check_sum,'string','8');
elseif cek == 3
   set (handles.check_sum,'string','7');
elseif cek == 4
   set (handles.check_sum,'string','6');
elseif cek == 5
   set (handles.check_sum,'string','5');
elseif cek == 6
   set (handles.check_sum,'string','4');
elseif cek == 7
   set (handles.check_sum,'string','3');
elseif cek == 8
   set (handles.check_sum,'string','2');
elseif cek == 9
   set (handles.check_sum,'string','1');
else
   set (handles.check_sum,'string','0');
end
```