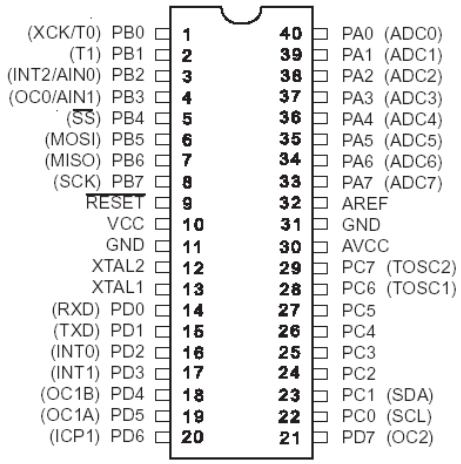
BAB II
TEORI PENUNJANG

2.1 *Mikrokontroler*

Mikrokontroler merupakan suatu komponen elektronika yang didalamnya terdapat rangkaian *mikroprosesor*, memori (*RAM*/*ROM*) dan *I/O*, rangkaian tersebut terdapat dalam *level chip*atau biasa disebut *single chip microcomputer*. Pada *mikrokontroler* sudah terdapat komponen-komponen *mikroprosesor* dengan bus-bus *internal* yang saling berhubungan. Komponen-komponen tersebut adalah *RAM, ROM, timer,* komponen *I/O* paralel dan serial, dan *interrupt kontroller*. Adapun keunggulan dari *Mikrokontroler* adalah adanya sistem *interrupt*. Sebagai perangkat kontrol penyesuaian, *mikrokontroler* sering disebut juga untuk menaikkan respon*external* (*interrupt*) pada waktu yang nyata. Perangkat tersebut harus melakukan hubungan *switching*cepat, menunda satu proses ketika adanya respon eksekusi yang lain.

AVR merupakan seri *mikrokontrolerCMOS* 8-bit buatan *Atmel*, berbasis arsitektur *RISC* (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu siklus *clock*. *AVR* mempunyai 32 *register general-purpose, timer/counter* fleksibel dengan mode *compare*, *interupt internal* dan *external*, *serial UART*, *programmable Watchdog Timer,* dan mode *power saving*. Beberapa diantaranya mempunyai *ADC* dan *PWM* internal. *AVR* juga mempunyai *In-System Programmable Flash on-chip* yang mengijinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial *SPI* (*Serial Peripheral Inteface*). *Chip AVR* yang digunakan untuk tugas akhir ini adalah *ATmega 8535*. *ATmega 8535* adalah mikrokontroler *CMOS* 8-bit daya-rendah berbasis arsitektur *RISC* yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus *clock*, *ATmega 8535* mempunyai *throughput* mendekati 1 *MIPS* per *MHz* membuat desain sistem untuk mengoptimalkan konsumsi daya dengan kecepatan proses.

AVR ATmega8535 mempunyai 40 kaki, 32 kaki yang digunakan untuk keperluan *port* paralel setiap *port* terdiri dari 8 pin, sehingga terdapat 4 *port*, yaitu *port* A, *port* B, *port* C, *port* D. seperti gambar dibawah ini :



Gambar II-1Konfigurasi Pin AVR ATmega 8535

Berikut adalah penjelasan fungsi tiap kaki yang biasa ada pada seri *mikrokontroler AVR ATmega8535*yaitu :

1. *VCC(power supply)*
2. *GND(ground)*
3. *Port* A (PA7..PA0)

Port A berfungsi sebagai *input* analog pada A/D Konverter. *Port* A juga berfungsi sebagai suatu *Port* I/O 8-bit dua arah, jika A/D Konverter tidak digunakan. *Pin - pin Port* dapat menyediakan *resistorinternal pull-up* (yang dipilih untuk masing-masing *bit*). *Port*A*output buffer* mempunyai karakteristik gerakan simetris. Ketika *pin* PA0 ke PA7 digunakan sebagai *input* dan secara *external* ditarik rendah, *pin–pin*akan memungkinkan arus sumber jika *resistor internal pull-up* diaktifkan. *Pin Port*A adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.

4. *Port* B (PB7..PB0)

Port B adalah suatu *PortI/O* 8-bit dua arah dengan *resistorinternal pull-up* (yang dipilih untuk beberapa *bit*). *Port* B *output buffer* mempunyai karakteristik gerakan simetris. Sebagai *input*, *pin port* B yang secara *external* ditarik rendah

akan arus sumber jika *resistor pullup* diaktifkan. *Pin Port* B adalah *tri-stated* pada suatu kondisi tertentu.

5. *Port* C (PC7..PC0)

Port C adalah suatu *PortI/O* 8-bit dua arah dengan *resistorinternal pull-up* (yang dipilih untuk beberapa *bit*). *Port* C *output buffer* mempunyai karakteristik gerakan simetris. Sebagai *input, pin port* C yang secara *external* ditarik rendah akan arus sumber jika *resistor pullup* diaktifkan. *Pin Port* C adalah *tri-stated* pada suatu kondisi tertentu.

6. *Port* D (PD7..PD0)

Port D adalah suatu *PortI/O* 8-bit dua arah dengan *resistorinternal pull-up* (yang dipilih untuk beberapa *bit*). *Port* D *output buffer* mempunyai karakteristik gerakan simetris. Sebagai *input, pinport* D yang secara *external* ditarik rendah akan arus sumber jika *resistorpullup* diaktifkan. *Pin Port* D adalah *tri-stated* pada suatu kondisi tertentu.

7. *RESET (Reset input)*

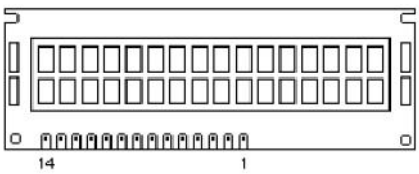
8. *XTAL1(Input Oscillator)*

9. *XTAL2 (Output Oscillator) AVCC* adalah *pin* penyedia tegangan untuk *port* A dan *A/D* Konverter.

10. AREF adalah *pin* referensi analog untuk *A/D* konverter.

2.2 *LCD* (*Liquid Crystal Display*)

LCD (Liquid Crystal Display) merupakan penampil karakter yang berfungsi untuk menampilkan karakter yang dimasukan baik oleh *keypad* maupun berdasarkan program yang dibuat pada *mikrokontroler*. Data yang ditampilkan berupa karakter baik angka maupun huruf. Adapun struktur dan deskripsi dari *LCD* 2 *line* dengan 16 karakter tampak pada gambar dan tabel di bawah ini.



Gambar II-2 LCD 16x2

Tampilan layar *LCD* yang digunakan adalah *LCD Dot Matrix Kompatibel Hitachi* yang memiliki *onboard mikrokontroler* HD44870. *LCD* ini mampu menampilkan 2 baris dengan 16 karakter tiap barisnya.

Tabel II-1Deskripsi LCD 16x2

Sinyal	I/O	Koneksi dengan ATmega 8535	Penjelasan
Vss	1	*GND*	*Ground*
Vdd	1	*VCC*	Tegangan catu daya +5*VDC*
VD	1	-1 s/d 5 (dari VR1)	Tegangan untuk mengatur kontras *LCD*
Rs	1	*Port* D	Memilih *register* internal dari *LCD*
R/W	1	*Port* D	Sinyal *Read/Write*
E	1	*Port* D	Sinyal untuk *enable*
DB0 .DB7	I/O	PD.0 hingga PD.7 (BUS)	8 bit data BUS

Pin data (D0....D7) dapat dihubungkan dengan bus data dari rangkaian *mikrokontroler* dengan lebar data 8 *bit* atau 4 *bit*, sementara *pin* kontrol terdiri dari *Rs*, *R/W* dan E. *Pin* Rs (*Register select*) berfungsi sebagai indikator untuk menetukan jenis data yang masuk, apakah data atau perintah. Logika *low* menunjukan yang masuk adalah perintah, sementara logika *high* menunjukan data.

Mikrokontroler akan mengirim suatu karakter atau kata pada *input* data 8 *bit* modul *LCD*, serta mengirim sinyal atau logika *high* pada *pinRs*. Kemudian, *mikrokontroler* akan mengirim perintah penulisan pada baris ke 1 ataupun baris 2 melalui *input* data 8 *bit* dengan memberikan logika *high* pada pin*Rs*. *PinR/W* berfungsi sebagai instruksi pada *LCD*, apakah *Read* atau *Write*. Ketika *R/W* diberi logika *low* berarti penulisan data (*write*), sedangkan logika *high* berarti pembacaan data (*read*). *Pin* E (*Enable*) digunakan untuk memegang data, baik masuk atau keluar dari *LCD*.

2.3 *WebCam* (*Web Camera*)

Web Camera atau yang biasa disingkat *WebCam* ini termasuk ke dalam perangkat keras periferal berupa kamera *video* sederhana dengan ukuran yang relatif lebih kecil. Biasanya ini digunakan untuk kebutuhan *video call* maupun kamera pemantau. Jadi *WebCam* umumnya tidak memerlukan kaset maupun tempat untuk menyimpan data melainkan data yang didapat akan langsung ditransferkan kepada komputer.



Gambar II-3 WebCam (Web Camera)

Jenis-jenis dari *WebCam* dibagi menjadi beberapa bagian, yaitu :

1. **Jenis *Serial* dan *Paralelport WebCam***

WebCam jenis ini telah jarang digunakan sekarang ini, karena juga sudah tidak lagi diproduksi. Jenis *WebCam* pertama kali ini juga menghasilkan gambar dengan kualitas yang rendah serta *frame rate* yang sangat rendah pula.

2. **Jenis *USB WebCam***

WebCam ini adalah solusi untuk pengguna yang masih baru. *WebCam* yang telah mendukung fasilitas *PnP*(*Plug and Play*) ini bisa dihubungkan secara langsung ke *port* tanpa perlu mematikan komputer. Akan tetapi syarat yang harus dipenuhi yaitu *OS*-nya harus sudah mendukung fasilitas *portusb*.

3. **Jenis *Firewire and Card Based WebCam***

Firewire merupakan teknologi *videocapture device* yang diperlukan untuk kamera yang telah mendukungnya. Biasanya *WebCam* yang memerlukan teknologi tersebut harganya relatif mahal. Akan tetapi kemampuan yang ditawarkan pun bisa menghasilkan 24 hingga 30 *fps*.

4. **Jenis *Network and Wireless Camera***

Network Camera merupakan salah satu perangkat kamera yang tidak membutuhkan fasilitas komputer sama sekali. Karena kamera jenis ini bisa tehubung langsung ke jaringan dengan *modem*.

- **Jenis *Webcam* Yang digunakan**

Dari berbagai jenis *WebCam* yang ada dipasaran,*WebCam*yang digunakan untuk mengerjakan tugas akhir ini adalah*Logitech C270*jenis *USB WebCam*, karena mudah didapat, dan mudah untuk dihubungkan ke *PC*.

2.4 *Visual Basic*

Visual Basic merupakan bahasa pemograman komputer. Bahasa pemograman adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. *Visual Basic* selain disebut sebagai bahasa pemograman, juga sering disebut sebagai sarana (*tool*) untuk menghasilkan program-program aplikasi berbasiskan *windows*.

Visual Basic memungkinkan pembuatan aplikasi *Graphical User Interface* (*GUI*) atau pemograman yang menggunakan tampilan grafis sebagai alat komunikasi dengan pemakainya. Dalam *Visual Basic* untuk pembuatan tampilan *user interface* relatif mudah dilakukan karena hanya perlu meletakkan objek-objek grafis ke sumber (*form*) yang sudah disediakan *Visual Basic*. Setelah itu cukup mengatur properti dari objek-objek tersebut.

- **Lingkungan *Visual Basic 6.0***

Saat kita masuk ke tampilan utama *Visual Basic,* kita akan melihat banyak menu yang dapat kita pilih, dan berikut adalah beberapa menu di lingkungan kerja yang ada di visual basic.

1. Main Menu

Main menu terdiri dari dua komponen yaitu *Menubar* dan*TitleBar*. *Menubar* menampilkan menu yang berisi perintah-perintah pada *Visual Basic*,

sedangkan *TitleBar* menampilkan judul proyek *Visual Basic* yang sedang dikerjakan.

2. Menu ToolBar

Menu ToolBar merupakan menu berbentuk *icon* yang berisi perintah. *ToolBar* yang disediakan *Visual Basic* untuk mengakses berbagai fungsi yang ada dalam menu secara lebih cepat dan lebih mudah.

3. Menu ToolBox

ToolBox mengandung semua objek atau kontrol yang dibutuhkan untuk membentuk suatu program aplikasi. Kontrol adalah suatu objek yang akan menjadi *interface* (penghubung) antara program aplikasi dan *user*.

4. Form Window

Form Window atau jendela *form* adalah daerah kerja utama, *Form Window* untuk meletakkan berbagai macam objek interaktif seperti teks, gambar, tombol-tombol perintah, *scrollbar* dan sebagainya. Pada saat program aplikasi dijalankan, semua yang terdapat di dalam *form*akan ditampilkan pada layar *window*. Jendela *form* inilah yang akan menjadi latar belakang dari program aplikasi.

5. Project Explorer

Jendela *Project Explorer* adalah jendela yang mengandung semua *file* di dalam aplikasi *Visual Basic*. Setiap aplikasi dalam *Visual Basic* disebut dengan istilah *project* (proyek), dan setiap proyek bisa mengandung lebih dari satu *file*. Pada *Project Explorer* ditampilkan semua *file* yang terdapat pada aplikasi (proyek), misalnya *form*, modul, *class* dan sebaginya.

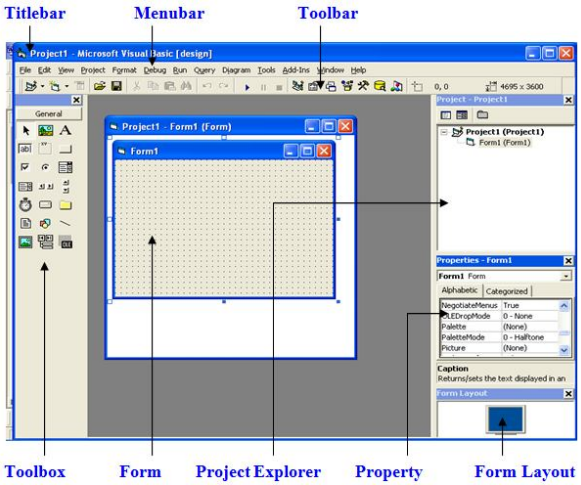
6. Jendela *Properties*

Jendela *Properties* adalah jendela yang mengandung semua informasi mengenai objek yang terdapat pada aplikasi *Visual Basic*. Properti adalah sifat dari sebuah objek, seperti nama, warna, ukuran, posisi dan sebagainya. Untuk mengatur bentuk dan karakteristik dari setiap objek melalui Jendela *Properties*. Jendela *Properties* ini memiliki dua buah *tab* yaitu *tabAlphabetic* dan *tabCategorized*. Pada *tabAlphabetic*, semua properti dari objek akan diurutkan berdasarkan abjad, sedangkan pada *tabCategorized*, semua properti dikelompokkan berdasarkan jenisnya. Tidak ada perbedaan antara kedua *tab*

tersebut terhadap properti yang ditampilkan, perbedaan keduanya hanyalah cara menampilkan.

7. Form Layout Window

*Form Layout Window*adalah jendela yang menggambarkan posisi dari *form* yang ditampilkan pada layar monitor. Posisi *form* pada *Form Layout Window* inilah yang merupakan petunjuk di mana aplikasi akan ditampilkan pada layar monitor saat dijalankan. Jika ingin memindahkan *form* ke posisi yang lain pada layar, dengan menggeser *form* tersebut pada jendela *FormLayout Window.*



Gambar II-4 Tampilan Visual Basic dengan form sebagai area kerja

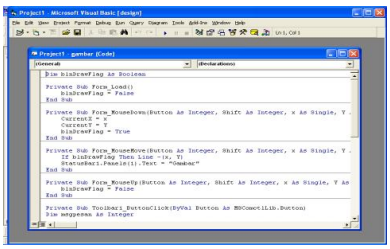
Keterangan:

- *Titlebar* : menampilkan judul proyek *Visual Basic*
- *Toolbar*: *shortcut* yang digunakan untuk membuat perintah suatu proyek.
- *Menubar*: menu untuk perintah *Visual Basic 6.0*
- *Toolbox*: objek yang digunakan untuk membuat suatu *form*.
- *Form*: bidang untuk membuat suatu *form*.
- *Project Explorer* : menampilkan semua *file* didalam aplikasi *Visual Basic 6.0*.
- *Property*: daftar *setting* properti.
- *Form Layout*: penampilan *form* pada layar.

8. Jendela *Code*

Jendela *Code* adalah salah satu jendela yang penting dalam *Visual Basic*. Jendela ini berisi kode-kode program yang merupakan instruksi-instruksi untuk aplikasi *Visual Basic*. Setiap objek pada *Visual Basic* dapat ditambahkan dengan kode-kode program untuk melakukan tugas-tugas tertentu, misalnya menutup aplikasi, membatalkan perintah dan sebagainya. Pada saat menjalankan *Visual Basic*, jendela ini tidak akan ditampilkan pada layar. Cara untuk menampilkan Jendela Kode :

- Pilih menu ***View>Code***
- Klik ganda objek tertentu pada *Form Window*, atau
- Klik ganda pada komponen yang diinginkan, lalu pilih ***View Code***



Gambar II-5Tampilan area kerja di Visual Basic

2.5 *Matlab* (*Matrix Laboratory*)

Bahasa pemrograman berfungsi sebagai media untuk membuat antarmuka di antara user dengan komputer dan pemrogramannya dibuat semakin mudah untuk digunakan, contohnya yaitu *pascal* terus memunculkan versi baru misalnya *turbo pascal for windows*, bahasa C dengan *turbo*Cnya dan lain sebagainya. *Matlab* muncul dengan sesuatu yang berbeda bila dibandingkan dengan bahasa pemrograman lain, *Matlab* dikembangkan sebagai bahasa pemrograman sekaligus alat visualisasiserta bisa menyelesaikan banyak permasalahan dan kasus yang berhubungan langsung dengan berbagai disiplin keilmuan seperti matematika, fisika, rekayasa teknik dan lain-lain.

*Matla*b muncul pada akhir tahun 1970-an dan pemrogramannya dirancang oleh Cleve Moler, maksud dia merancangnya adalah untuk memberikan kemudahan bagi mahasiswa. *Matlab* cepat menyebar dan dipakai oleh berbagai

kalangan, bukan hanya dipakai oleh universitas tetapi dipakai juga oleh ilmuan untuk visualisasi, pemodelan, simulasi, rekayasa dan sebagainya. Menyadari kegunaan dan potensi dari *Matlab* sangat bagus, kemudian Cleve Moler membentuk team untuk menulis ulang *Matlab* yang kemudian sekarang ini kita dapat memakai *Matlab* dengan berbagai kemudahannya, bahkan situs resmi untuk mengakses dan mempelajari *Matlab* sudah disediakan langsung oleh pihak *Matlab*.

2.5.1 Fasilitas yang tersedia di *Matlab*(*Matrix Laboratory*)

Matlab mempunyai banyak fitur-fitur yang sangat memudahkan dalam penggunaannya, dari awal keluarnya *Matlab* terus dikembangkan sehingga *toolbox* sangat membantu penggunanya dalam pemakaian. *Toolbox* ini merupakan kumpulan dari fungsi-fungsi *Matlab* yang memudahkan dalam memecahkan masalah seperti pengolahan sinyal, sistem kontrol, *neural networks* dan lain-lain. Dalam memakai dan melakukan pemrograman dengan *Matlab* ada dua cara untuk menggunakannya, yaitu:

1. Cara langsung di *command window.*

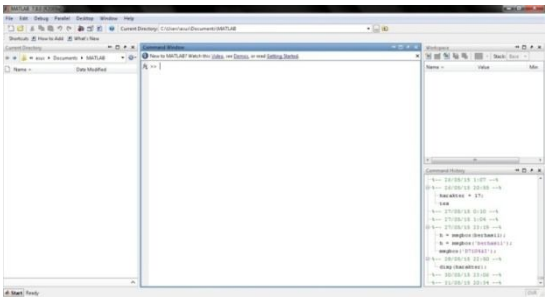
Cara pertama ini paling sering dilakukan dan hasil dari pengetikan program bisa langsung terlihat hasilnya di *command window* itu sendiri.

2. Menggunakan *File* M

Cara ini memakai *Matlabeditor*, biasanya cara ini dipakai oleh orang yang sudah mengenal lebih jauh tentang *Matlab*, kelebihan dari cara ini yaitu kemudahan dalam mengevaluasi perintah secara keseluruhan, terutama untuk program yang membutuhkan waktu pengerjaan yang cukup lama serta skrip yang panjang. Penjelasan untuk fitur-fitur yang ada dalam *Matlab* sebagai berikut:

A. *Window* utama *Matlab* (*Matrix Laboratory*)

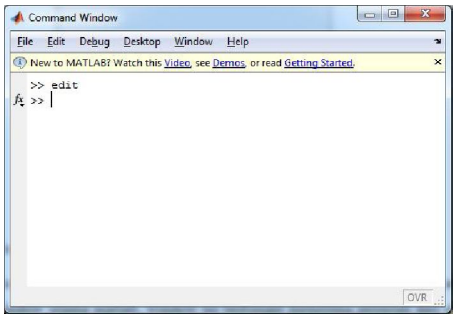
Window ini merupakan *window* induk untuk pengaksesan ke seluruh lingkungan kerja menu-menu yang ada dalam *Matlab*.



Gambar II-6 Tampilan utama Matlab

B. *Command window*

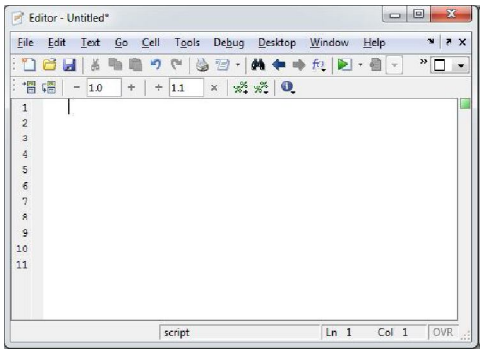
Command window ini dapat diakses dengan mengaksesnya pada menu *window* utama *Matlab*. *Window* ini berfungsi menerima perintah dari *user* untuk menjalankan seluruh fungsi yang ada di *Matlab*.



Gambar II-7 Tampilan tempat kerja Command Window

C. *MatlabEditor*

Ketika membuka *window* utama *Matlab* (*Matrix Laboratory*), *window editor* ini tidak akan langsung muncul, tetapi harus diakses lewat *promptMatlab*, atau dengan cara klik menu *file*, *new* ,dan pilih *script*, sehingga muncul *window editor* seperti pada gambar II-8.



Gambar II-8 Tampilan tempat kerja editor

2.5.2 Metode yang digunakan untuk pengolahan citra

Dari berbagai jenis metode yang biasa digunakan untuk pengolahan citra digital, metode yang dipilih penulis untuk mengerjakan tugas akhir ini yaitu menggunakan metode korelasi *template matching*.

Seperti yang telah dijelaskan sebelumnya, tentang metode yang digunakan penulis untuk menyelesaikan tugas akhir ini.Menurut Jurie *TemplateMatching* merupakan metode pengolahan citradigital untuk menemukan bagian-bagian kecil darigambar yang cocok dengan *template* gambar danmerupakan metode yang paling populer untukmendapatkan informasi dari media gambar.Dalam metode ini dikenal pencarian fitur tertentu atau target dari waktu ke waktu yang didasarkan pada perbandingan isi setiap gambar dengan *sample template*.[6] Sedangkan menurut Roberto Brunelli *template matching*adalah suatu teknik untuk mengukur kesamaan dari dua gambar digital, untuk menentukan apakah mereka adalah sama atau tidak.[7]

Secara umum tahap-tahap yang dilakukan dari metode korelasi *Template Matching* yang dibuat oleh penulis antara lain *Input* Citra, *Cropping*, *Grayscalling, thresholding*, Segmentasi citra, Ekstraksi ciri, hasil pengolahan. Untuk pengertian dari masing-masing tahapan diatas, dapat dilihat dari penjelasan berikut ini.

1. ***Input***

Pada proses ini akan dimasukkan sebuah citra/gambar plat nomor bus dengan format .jpg yang selanjutnya akan diolah untuk pengenalan karakter yang ada di plat nomor bus.

2. ***Cropping***

Croppingmerupakan usaha memberi batas sisi gambar sehingga dapat diketahui batas gambar yang dipakai dan sisi atau bagian gambar yang dibuang.

3. ***Grayscalling***

Grayscalling atau abu-abu pada gambar digital adalah gambar yang pada setiap pikselnya hanya berisikan informasi intensitas warna putih dan hitam. Gambar *grayscalling* memilki banyak variasi nuansa abu-abu sehingga berbeda dengan gambar hitam-putih.[8]

4. ***Thresholding***

Thresholding adalah proses mengubah citra berderajat keabuan menjadi citra *biner* atau hitam putih sehingga dapat diketahui daerah mana yang termasuk obyek dan *background* secara jelas. Citra hasil *thresholding* biasanya digunakan lebih lanjut untuk proses pengenalan obyek serta ekstraksi ciri.[9]

5. **Segmentasi Citra**

Segmentasi citra adalah membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan yang tertentu antara tingkat keabuan suatu piksel dengan tingkat keabuan piksel-piksel tetangganya. Proses segmentasi memiliki tujuan yang hampir sama dengan proses klasifikasi tidak terpandu. Segmentasi sering dideskripsikan sebagai proses analogi terhadap proses pmisahan latar depan dan latar belakang.[10]

6. **Ekstraksi Ciri**

Ekstraksi cirrimerupakan bagian dari teknik pengenalan yangbertujuan untuk mengambil atau mengekstraksi nilai-nilai unik dari suatu obyek yang membedakan dengan obyek yang lain.[11]

7. **Hasil Pengolahan**

Hasil pengolahan merupakan bagian yang akan menampilkan *output* dari pengenalan plat nomor berupa karakter yang terdeteksi oleh sistem.