BAB II
TINJAUAN PUSTAKA

2.1. Pengolahan Citra

Citra adalah representasi dari dua dimensi untuk bentuk fisik nyata tiga dimensi. Dalam perwujudannya, citra dibagi menjadi 2 yaitu citra diam (*still images*) dan citra bergerak (*moving images*). Citra diam adalah citra tunggal yang tidak bergerak. Sedangkan citra bergerak adalah rangkaian citra diam yang ditampilkan secara sekuensial sehingga memberi kesan pada mata kita sebagai gambar yang bergerak. Agar citra yang mengalami gangguan mudah diinterpretasi baik oleh manusia maupun mesin, maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik atau disebut pengolahan citra (*image processing*). Karena pengolahan citra dilakukan dalam komputer digital, maka citra yang akan diolah terlebih dahulu ditransformasikan kedalam bentuk besaran-besaran diskrit pada titik-titik elemen citra, dan bentuk dari citra ini disebut citra digital.

2.1.1. Komponen Citra Digital

Setiap citra *digital* memiliki beberapa karakteristik, antara lain ukuran citra, resolusi, dan format nilainya. Umumnya citra digital berbentuk persegi panjang yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya titik atau *pixel* (*Picture Element*), sehingga ukuran citra selalu bernilai bulat.

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

Format citra digital ada bermacam-macam. Karena sebenarnya citra merepresentasikan informasi tertentu, sedangkan informasi tersebut dapat dinyatakan secara bervariasi, maka citra yang mewakilinya dapat muncul dalam berbagai format. Citra yang merepresentasikan informasi hanya bersifat biner untuk membedakan 2 keadaan tertentu tidak sama dengan informasi yang lebih kompleks sehingga memerlukan lebih banyak keadaan yang diwakilinya. Pada citra digital semua informasi

tadi disimpan dalam bentuk angka, sedangkan penampilan angka tersebut biasanya dikaitkan dengan warna.

Citra digital tersusun atas titik-titik yang biasanya berbentuk persegi panjang atau bujursangkar yang secara beraturan membentuk baris dan kolom. Setiap titik memiliki koordinat sesuai posisinya dalam citra. Koordinat ini biasanya dinyatakan dalam bilangan bulat positif, yang dapat dimulai dari 0 atau 1 tergantung pada sistem yang digunakan.

2.1.2. Citra Skala Keabuan (*Grayscale*)

Citra skala keabuan memberi kemungkinan warna yang lebih banyak daripada citra biner, karena ada nilai-nilai lain diantara nilai minimum (0) dan nilai maksimumnya (1). Banyaknya kemungkinan nilai dan nilai maksimumnya bergantung pada jumlah bit yang digunakan. Contohnya untuk skala keabuan 4 bit, maka jumlah kemungkinan nilainya adalah $2^4 = 16$, dan nilai maksimumnya adalah $2^4 - 1 = 15$; sedangkan untuk skala keabuan 8 bit, maka jumlah kemungkinan nilainya adalah $2^8 = 256$ dan nilai maksimumnya adalah $2^8 - 1 = 255$.

Format citra ini disebut skala keabuan karena pada umumnya warna yang dipakai adalah antara hitam sebagai warna minimal dan warna putih sebagai warna maksimalnya, sehingga warna diantaranya keduanya adalah warna abu-abu. Pada umumnya citra skala keabuan menggunakan jumlah bit 8, sesuai dengan satuan memori komputer (*byte*).

Dalam hal ini, proses diatas dapat dilakukan dengan menerapkan fungsi linier untuk memetakan skala citra *true color* menjadi skala citra *grayscale*, berikut ini adalah persamaannya:

$$Ko = \frac{Ri + Gi + Bi}{3} \quad \text{.....(2.1)}$$

Keterangan :

$K_o = NilaiOutput$

$Ri = NilaiInputWarnaMerah$

$Gi = NilaiInputWarnaHijau$

$Bi = NilaiInputWarnaBiru$

2.1.3. Modifikasi Kecemerlangan Citra

Di ruangan yang gelap perlu menyalakan lampu atau sumber cahaya lainnya agar ruangan menjadi lebih terang dan dapat melihat benda-benda yang ada didalam ruangan tersebut dengan lebih jelas. Dalam pengolahan citra, hal itu analog dengan penambahan nilai warna putih (analog dengan sumber cahaya) yaitu dengan cara meningkatkan skala keabuan dari seluruh bagian (setiap titik) dalam citra tersebut untuk meningkatkan kecemerlangannya (*brightness*). Sebaliknya apabila citra terlalu cemerlang atau kelihatan pucat, maka tingkat keabuan dari setiap titik dalam citra tersebut perlu diturunkan.

Dalam hal ini, proses diatas dapat dilakukan dengan menerapkan fungsi linier untuk memetakan skala keabuan citra, berikut ini adalah persamaannya:

$$K_o = K_i + C \quad \text{.....(2.2)}$$

Keterangan :

$K_o =$ *NilaiOutput*

$K_i =$ *NilaiInput*

Dimana *C* adalah konstanta yang bernilai positif jika hendak meningkatkan kecemerlangan citra, dan sebaliknya bernilai negatif jika hendak mengurangi kecemerlangannya.

2.1.4. Peningkatan Kontras

Jika sebuah citra yang mempunyai nilai keabuan yang tidak terlalu berbeda untuk semua titik, maka citra tersebut akan kelihatan kurang kontras. Hal ini disebabkan citra tersebut memiliki kurva histogram yang sempit, dengan tepi kiri dan kanan yang berdekatan, sehingga titik tergelap dalam citra tersebut tidak mencapai hitam pekat dan titik paling terang dalam citra itu tidak berwarna putih cemerlang. Peningkatan kontras dapat dilakukan dengan menggunakan persamaan berikut:

$$K_o = G(K_i - P) + P \quad \text{.....(2.3)}$$

Dengan *G* adalah koefisien penguatan kontras, dan *P* adalah nilai skala keabuan yang dipakai sebagai pusat pengontrasan.

2.1.5. Pengambangan (*Thresholding*)

Operasi pengambangan digunakan untuk mengubah citra dengan format skala keabuan yang mempunyai nilai lebih dari 2 ke citra biner yang hanya memiliki 2 buah

nilai (0 dan 1). Dalam hal ini, titik dengan nilai rentang nilai keabuan tertentu diubah menjadi berwarna hitam dan sisanya menjadi putih atau sebaliknya.

$$K_o = \begin{cases} 1, \; jikaambangbawah \le K_i \le ambangatas \\ 0, lainnya \end{cases} \quad \text{.....(2.4)}$$

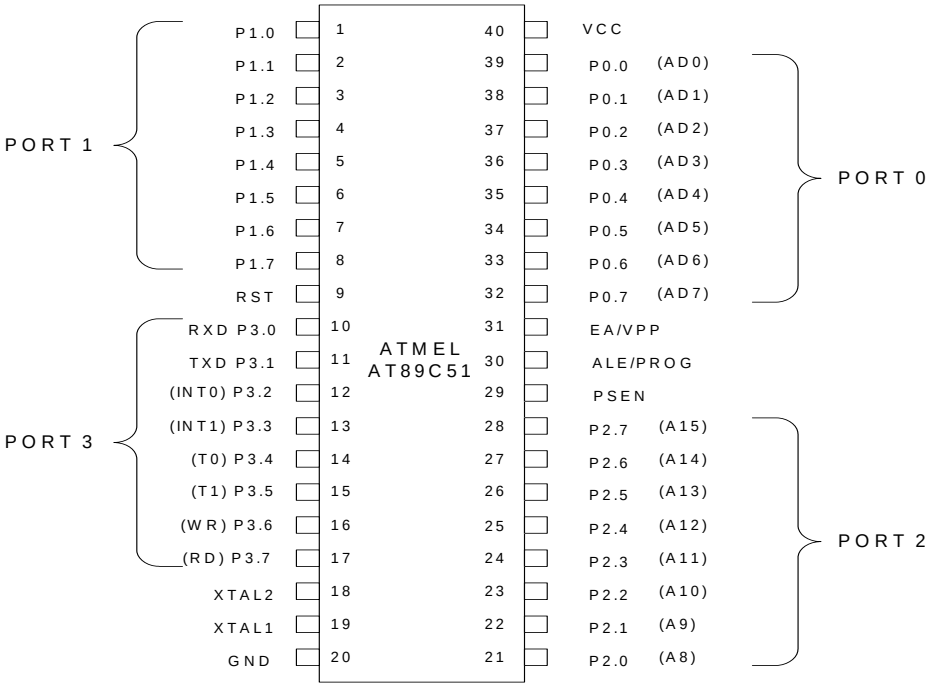
Pengambangan dilakukan untuk menampilkan titik-titik yang mempunyai rentang nilai skala keabuan tertentu, seperti penggabungan citra (*image blending*), perbaikan ketidaklinieran pencahayaan, deteksi gerakan, pembuatan citra diferensi untuk deteksi tepi, serta *digital angiography*.

2.2. Perangkat Keras

2.2.1. Mikrokontroler AT89C51

Komponen utama sistem identifikasi posisi benda ini adalah mikrokontroler AT89C51. AT89C51 adalah mikrokontroler keluaran atmel dengan 4K byte (*In-System Programmable Flash Memory*), AT89C51 merupakan memori dengan teknologi *nonvolatile* memori dan memori ini dapat diisi ulang ataupun dihapus berkali-kali.

Memori ini bisa digunakan untuk menyimpan instruksi berstandar pada kode MCS-51 sehingga memungkinkan mikrokontroler ini bekerja dalam mode *single chip operation* (mode operasi keping tunggal) yang tidak memerlukan memori luar untuk menyimpan *source code* tersebut.



Gambar 2.1 Konfigurasi pin Mikrokontroler AT89C51

2.2.1.1.Deskripsi kaki AT89C51

Pada AT89C51 memiliki masing-masing fungsi untuk setiap kakinya. Adapun masing-masing fungsi dari setiap kakinya adalah sebagai berikut:

Port 0

Port 0 terdiri dari 8 kaki (p0.0-p0.7), yaitu kaki ke 39 sampai kaki ke 32. Port ini biasa berfungsi sebagai I/O biasa, *low order multiplex address*/data ataupun menerima kode byte pada saat *flash progamming*. Port ini dapat memberikan output sink ke 8 buah TTL (*transistor transistor logic*) input atau bisa di ubah sebagai input dengan memberikan logika 1 pada port tersebut.

Port 1

Port 1 terdiri dari 8 kaki (p1.0 – p1.7), yaitu kaki ke 1 sampai ke 8. Port ini bisa berfungsi sebagai I/O biasa atau menerima *low order address byte* selama *flash programming*. Port ini mempunyai *internal pull up* dan untuk digunakan sebagai input maka harus di berikan logika 1 pada port ini. Pada saat digunakan sebagai output port ini dapat memberikan output sink ke 4 buah TTL (*transistor transistor logic*).

Port 2

Port 2 terdiri dari 8 kaki (p2.0 – p2.7), yaitu kaki ke 21 sampai kaki ke 28. Port ini juga bisa digunakan sebagai I/O biasa atau *high order address* pada saat mengakses memori external secara 16 bit (movx @DPTR). Pada saat mengakses memori external secara 8 bit (mov@RI) port ini akan mengeluarkan isi dari port 2 *special function register*. Port ini mempunyai *internal pull up* dan berfungsi sebagai input dengan memberikan nilai logika 1. Dan sebagai output port ini bisa memberikan output sink ke 4 buah input TTL (*transistor transistor logic*).

Port 3

Port 3 terdiri dari 8 kaki juga (p3.0 – p3.7) yaitu kaki ke 10 sampai kaki ke 17. sebagai I/O biasa port 3 mempuyai sifat yang sama dengan port 1 dan 2. sedangkan sebagai fungsi spesial *port-port* ini mempunyai keterangan sebagai berikut:

P3.0 (RXD) berfungsi sebagai *port serial input*.

P3.1 (TXD) berfungsi sebagai *port serial output*.

P3.2 (INT0) berfungsi sebagai *port external interupt* 0.

P3.3 (INT1) berfungsi sebagai *port external interupt* 1.

P3.4 (T0) berfungsi sebagai *port external timer input* 0.

P3.5 (T1) berfungsi sebagai *port external timer input* 1.

P3.6 (WR) berfungsi sebagai *external* data memori *write strobe*.

P3.7 (RD) berfungsi sebagai *external* data memori *read strobe.*

Pin 9 (RST) *reset* akan aktif dengan memberikan *input high* selama 2 *cycle*.

Pin 18 (XTAL2) merupakan *output oscillator*.

Pin 19 (XTAL1) merupakan *input oscillator*.

Pin 20 (GND) merupakan *ground*.

Pin 29 (PSEN) pin ini berfungsi pada saat mengeksekusi program yang terletak pada memori *external*. *PSEN* akan aktif dua kali setiap *cycle*.

Pin 30 (ALE) pin ini bisa berfungsi sebagai *address latch enable* (ALE) yang melakukan *latch low byte address* pada saat mengakses memori *external*. Sedangkan pada saat *flash programing* berfungsi sebagai *pulsa input* pada operasi normal, *ALE* akan mengeluatkan sinyal clock sebesar 1/6 frekwensi oscillator kecuali pada saat memori *external*, sinyal *clock* pada pin ini dapat *disable* dengan di set *bit* 0 dari *special junction register* di alamat 8Eh *ALE* hanya akan aktif pada saat mengakses memori *external* (movx dan movc).

Pin 31 (EA) Pada kondisi rendah, pin ini akan berfungsi sebagai EA yaitu mikrokontroler akan menjalankan program yang ada pada memori *external* setelah sistem di *reset*. Jika kondisi tinggi, pin ini akan berfungsi untuk menjalankan program yang ada pada memori *internal*. Pada saat *flash programing* pin ini akan mendapatkan tegangan 12 Volt.

2.2.1.2.Struktur Memori

AT89C51 mempunyai struktur memori yang terdiri atas:

1. RAM internal adalah memori sebesar 128 *Byte* yang biasanya digunakan untuk menyimpan *variabel* atau data yang bersifat sementara, terdiri dari:
   a. *Register Banks*

      AT89C51 mempunyai delapan buah *register* yang terdiri atas R0 hingga R7. ke 8 *register* ini selalu terletak pada alamat 00h hingga 07h pada setiap kali sistem di *reset*. Namun posisi R0 hingga R7 dapat di pindahkan ke *bank 1* (08H hingga 0FH), *bank 2* (10H hingga 17H) atau *bank 3* (18H hingga 1FH) dengan mengatur *bit RS0* dan *RS1*.

   b. *Bit addressable RAM*

      RAM pada alamat 20H hingga 2FH dapat diberi akses secara pengalamatan bit, sehingga hanya dengan sebuah instruksi saja setiap *bit* dalam area ini dapat di set, *clear*, *AND* dan *OR*.

c. *RAM* keperluan umum

   RAM keperluan umum dimulai dari alamat 30H hingga 7FH dan dapat diakses dengan pengalamatan langsung maupun tidak langsung.

2. *Special function register* adalah memori yang berisi banyak *register* yang mempunyai fungsi-fungsi khusus yang disediakan oleh mikrokontroler tersebut, seperti *timer*, *serial* dan lain-lain.
3. *In-system programable flash Memory* adalah memori yang digunakan untuk menyimpan instruksi-instruksi *MCS 51*.

2.2.2. Serial Port

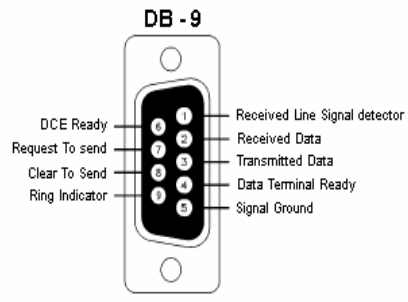
Komunikasi data pada umumnya dapat dilakukan dengan dua cara, yaitu secara *serial* dan secara *paralel*. Komunikasi data *serial* dilakukan dengan mengirimkan dan menerima data 8 *bit* secara satu per satu, sedangkan komunikasi data *paralel* dilakukan dengan mengirimkan dan menerima data 8 *bit* secara bersamaan atau sekaligus. *RS232* (*recommended standard number 232*) merupakan seperangkat alat yang diciptakan oleh *Electrical Industries Assosiation* yang berfungsi sebagai antarmuka dalam transfer data dengan komputer yang mana pengiriman data dilakukan dengan mengirimkan kode *biner*. Pada seperangkat komputer biasanya tersedia *Communication Port* atau sering disebut dengan COM. Biasanya terdapat dua buah *Communication Port*, yaitu COM1 dan COM2.

Pada dasarnya ada dua jenis komunikasi data *serial*, yaitu komunikasi data *serial sinkron* dimana pengiriman *clock* dilakukan secara bersamaan dengan data *serial* dan komunikasi data *serial asinkron* dimana pengiriman *clock* dilakukan secara dua tahap, yaitu saat data dikirimkan dan saat data diterima *RS232*. Pada komputer mempunyai dua jenis konektor, yaitu konektor dengan 25 pin atau sering disebut konektor DB 25 dan konektor dengan 9 pin atau sering disebut konektor DB 9. Untuk DB 9 pada dasarnya hanya 3 pin yang terpakai, yaitu pin pengirim, penerima, dan *ground*. Perlu diperhatikan bahwa dalam pengiriman data serial semakin jauh jarak kirim maka kemungkinan *noise* atau gangguan semakin besar.

Dalam setiap proses *transfer* data *serial*, RS232 memerlukan sebuah *Data Terminal Equipment* atau sering disebut *DTE* dan *Data Communication Equipment* atau sering disebut *DCE* pada masing-masing terminal. Pengiriman data dilakukan secara *bit* per *bit*, misalnya jika ingin mengirim suatu karakter (A) yang dalam format *ASCII*

adalah 41H atau 01000001 *biner* maka data akan dikirim mulai dari bit pertama, kedua, sampai bit terakhir. Kecepatan transfer data harus sama antara penerima dan pengirim, jika kecepatannya tidak sama maka akan terjadi *overflow*. Kecepatan transmisi tersebut biasanya sering disebut dengan *baudrate*. *Baudrate* yang sering dipakai diantaranya adalah 110, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 230400, 460800, dan 921600. panjang data *bit* yang sering digunakan diantaranya adalah 4, 5, 6, 7, dan 8 *bit*.

Pada komunikasi data *serial* pada dasarnya yang dikirimkan adalah tegangan dan kemudian dibaca dalam data *bit*. Besar level tegangannya adalah antara -25 Volt sampai +25 Volt. Untuk bit dengan logika 1 maka besar level tegangannya adalah antara -3 Volt sampai -25 Volt, sedangkan untuk bit dengan logika 0 maka besar level tegangannya adalah antara +3 Volt sampai +25 Volt. Ada beberapa besar level tegangan yang tidak mempunyai logika, yaitu antara -3 Volt sampai +3 Volt, lebih kecil dari -25 Volt, dan lebih besar dari +25 Volt.



Gambar 2.2 Konektor DB 9

Berikut ini adalah tabel penggunaan pin, dan jenis sinyal konektor DB-9.

Tabel 2.1 Nama Sinyal DB-9

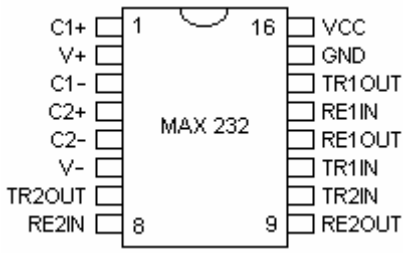
Pin (DB-9)	Nama Sinyal	Jenis
1	*Data Carrier Detect*	*Input*
2	*Received Data(RxD)*	*Input*
3	*Transmite data(TxD)*	*Output*
4	*Data Terminal Ready(DTR)*	*Output*
5	*Ground*	-
6	*Data Set Ready(DSR)*	*Input*
7	*Request To Send(RTS)*	*Output*
8	*Clear To Send(CTS)*	*Input*
9	*Ring Indicator*	*Input*

Keterangan:

1. Pin 1 (*Data Carrier Detect*) berfungsi untuk mendeteksi boleh atau tidaknya *DTE* menerima data.
2. Pin 2 (*Received Data*) berfungsi sebagai jalur penerimaan data dari *DCE* ke *DTE*.
3. Pin 3 (*Transmited Data*) berfungsi sebagai jalur pengiriman data dari *DTE* ke *DCE*.
4. Pin 4 (*Data Terminal Ready*) berfungsi untuk memberitahu kesiapan terminal *DTE*.
5. Pin 5 (*Ground*) berfungsi sebagai saluran .
6. Pin 6 (*Data Set Ready*) berfungsi untuk menyatakan bahwa status data tersambung pada *DCE*.
7. Pin 7 (*Request To Send*) berfungsi untuk mengirim sinyal informasi dari *DTE* ke *DCE* bahwa akan ada data yang akan dikirim.
8. Pin 8 (*Clear To Send*) berfungsi untuk memberitahu pada *DTE* bahwa *DCE* siap untuk menerima data.
9. Pin 9 (*Ring Indicator*) berfungsi untuk memberitahu *DTE* bahwa ada terminal yang menginginkan komunikasi dengan *DCE*.

2.2.2.1.MAX 232

Kita menggunakan DB 9 untuk komunikasi dengan komputer secara *serial*, untuk itu mikrokontroler memerlukan sebuah piranti yang berfungsi sebagai pengubah level tegangan. *Max232* menggunakan level/karakteristik *elektrik* yang berbeda dengan level *TTL* (*transistor transistor logic*). *Max232* bekerja pada level tegangan +3 s/d +25 *Volt* untuk *space* (logic 0) dan -3 s/d -25 *Volt* untuk *mark* (logic 1). Sedangkan *TTL* (*transistor transistor logic*) bekerja pada level tegangan -5 s/d +5 *Volt*. Piranti tambahan yang kita butuhkan adalah *IC MAX232*. Pada dasarnya *IC* ini hanya digunakan sebagai pengubah level tegangan ke level *Transistor Transistor Logic* (*TTL*), tidak berfungsi sebagai pengkodean sinyal yang melewati *DB 9*, dan juga tidak mengkonversikan data *serial* ke *paralel*.



Gambar 2.3 IC MAX232

DB 9 sebagai komunikasi *serial* mempunyai 9 pin yang memiliki fungsi masing-masing. Pin yang biasa digunakan adalah pin 2 sebagai *received* data, pin 3 sebagai *transmited* data, dan pin 5 sebagai *ground signal*. Karakteristik *elektrik* dari *DB 9* adalah sebagai berikut :

1. *Space* (logic 0) mempunyai level tegangan sebesar +3 s/d +25 *Volt*.
2. *Mark* (logic 1) mempunyai level tegangan sebesar -3 s/d -25 *Volt*.
3. Level tegangan antara +3 s/d -3 *Volt* tidak terdefinisikan.
4. Arus yang melalui rangkaian tidak boleh melebihi dari 500 mA, ini dibutuhlan agar sistem yang dibangun bekerja dengan akurat.

2.2.3. Motor Stepper

Motor stepper adalah salah satu tipe motor yang sangat populer digunakan sebagai penggerak/pemutar peralatan industri. Prinsip kerja motor stepper ini mirip dengan motor DC, yaitu sama-sama dicatu dengan tegangan DC untuk memperoleh medan magnet. Bila DC motor memiliki magnet tetap pada *stator*, motor stepper mempunyai magnet tetap pada rotor. Suatu motor stepper biasanya cukup dinyatakan dengan spesifikasi “berapa *phasa*“, “berapa derajat per step”, “berapa volt tegangan catu untuk tiap lilitan” dan “berapa ampere/miliampere arus yang dibutuhkan untuk tiap lilitan”. Motor stepper berbeda dengan motor DC. Motor stepper tidak dapat bergerak dengan sendirinya, motor stepper bergerak secara per step sesuai dengan spesifikasinya, dan bergerak dari satu step ke step berikutnya memerlukan waktu. Secara umum motor DC tidak menghasilkan torsi yang besar pada kecepatan rendah, sebaliknya motor stepper dapat menghasilkan torsi yang besar pada kecepatan rendah. Motor stepper juga memiliki karakteristik yang lain yaitu *holding torque*, yang tidak dimiliki oleh motor DC. *Holding torque* memungkinkan motor stepper dapat menahan posisinya ketika tidak berputar. Hal ini sangat berguna untuk aplikasi dimana suatu sistem memerlukan keadaan start dan stop.

Motor stepper tidak merespon sinyal clock, motor stepper mempunyai beberapa lilitan dimana lilitan-lilitan tersebut harus dicatu (tegangan) dahulu dengan suatu urutan tertentu agar dapat berotasi. Membalik urutan pemberian tegangan tersebut akan menyebabkan putaran motor stepper yang berbalik arah. Jika sinyal kontrol tidak terkirim sesuai dengan perintah maka motor stepper tidak akan berputar secara tepat, mungkin hanya akan bergetar dan tidak bergerak. Untuk mengontrol motor stepper biasanya kita menggunakan suatu rangkaian *driver* yang menangani kebutuhan arus dan tegangan.

Karakteristik dari motor stepper adalah sebagai berikut:

1. Voltage.

Tiap motor stepper mempunyai tegangan rata-rata yang biasanya tertulis pada tiap unitnya atau tercantum pada *datasheet* masing-masing motor stepper. Tegangan rata-rata ini harus diperhatikan dengan seksama karena bila melebihi dari tegangan rata-rata ini akan menimbulkan panas yang terlalu besar pada motor stepper yang menyebabkan kinerja putarannya tidak maksimal atau bahkan motor stepper akan rusak dengan sendirinya.

2. *Resistance.*

Resistance per winding adalah karakteristik yang lain dari motor stepper. *Resistance* ini akan menentukan arus yang mengalir, selain itu juga akan mempengaruhi torsi dan kecepatan maksimum dari motor stepper.

3. Derajat per Step.

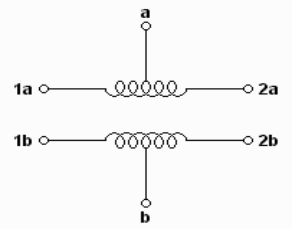
Derajat per step adalah faktor terpenting dalam pemilihan motor stepper sesuai dengan aplikasinya. Tiap-tiap motor stepper mempunyai spesifikasi masing-masing, antara lain: 0.720 derajat per step, 1.80 derajat per step, 3.60 derajat per step, 7.50 derajat per step, 150 derajat per step, dan bahkan ada yang 900 derajat per step. Dalam pengoperasiannya kita dapat menggunakan 2 prinsip yaitu *full-step* atau *half-step*. Dengan *full-step* berarti motor stepper akan berputar sesuai dengan spesifikasi derajat per stepnya, sedangkan *half-step* berarti motor stepper akan berputar setengah derajat/step dari spesifikasi motor stepper tersebut.

Ada beberapa macam motor stepper. Hal ini dibedakan menjadi dua kategori besar yaitu *permanent magnet* dan *variable reluctance*. Tipe *permanent magnet* terbagi menjadi dua motor stepper yaitu unipolar stepper motor dan bipolar stepper motor.

Unipolar stepper motor sangat mudah untuk dikontrol. Dengan rangkaian *counter* ‘-n’ yang sedehana dapat mengontrol putaran dari motor stepper dengan urutan

yang benar. Unipolar stepper motor mempunyai karakteristik khusus yaitu berupa lilitan *center-tapped* dan 1 lilitan sebagai *common*. Lilitan *common* akan mencatu tegangan pada *center-tapped* dan sebagai *ground* adalah rangkaian driver nya.

Unipolar stepper motor dapat dikenali dengan mengetahui adanya lilitan *center-tapped*. Jumlah *phase* dari motor stepper adalah dua kali dari jumlah *coil* nya. Biasanya pada unipolar stepper motor terdapat dua buah *coil*.



Gambar 2.4 Susunan coil unipolar motor stepper

Pada prinsipnya ada dua macam cara kerja unipolar stepper motor, yaitu *full-step* dan *half-step*. Terlihat pada tabel 2.3 dan tabel 2.4.

Tabel 2.2 Pemberian Tegangan untuk Bekerja FullStep

*FULL*STEP								
No	Tegangan yang diberikan pada lilitan							
	Arah putar searah jarum jam				Arah putar melawan jarum jam			
	a	b	c	d	a	b	c	d
1	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0
3	0	0	1	0	0	1	0	0
4	0	0	0	1	1	0	0	0

Pada *full step*, suatu titik pada sebuah kutub magnet di rotor akan kembali mendapat tarikan medan magnet stator pada lilitan yang sama setelah step ke 4. berikutnya dapat diberikan lagi mulai dari step 1. Setiap step, rotor bergerak searah atau berlawanan dengan jarum jam sebesar spesifikasi derajat per step dari motor stepper. Setiap step hanya menarik sebuah kutub saja. Tegangan ‘1’ adalah menunjukkan logika dalam level *Transistor Transistor Logic* (TTL). Besar tegangan sesungguhnya diatur dengan spesifikasi motor stepper yang dipakai.

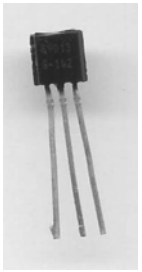
Tabel 2.3 Pemberian Tegangan untuk Bekerja Half Step

HALF STEP								
No	Tegangan yang diberikan pada lilitan							
	Arah putar searah jarum jam				Arah putar melawan jarum jam			
	a	b	c	d	a	b	c	d
1	1	0	0	0	0	0	0	1
2	1	1	0	0	0	0	1	1
3	0	1	0	0	0	0	1	0
4	0	1	1	0	0	1	1	0
5	0	0	1	0	0	1	0	0
6	0	0	1	1	1	1	0	0
7	0	0	0	1	1	0	0	0
8	1	0	0	1	1	0	0	1

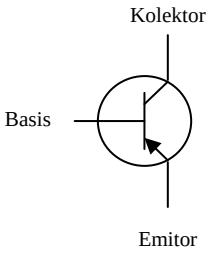
Untuk *half step*, setiap kutub magnet pada rotor akan kembali mendapatkan tarikan dari medan magnet lilitan yang sama setelah step ke 8. berikutnya kembali mulai step 1. Setiap step posisi rotor berubah sebesar setengah derajat dari spesifikasi derajat per step motor stepper.

2.2.4. Transistor

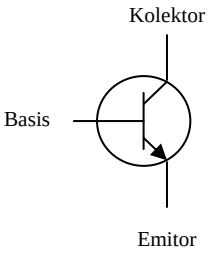
Transistor adalah komponen elektronik yang mempunyai fungsi sebagai penguat arus maupun tegangan dan sebagai saklar elektronik (dalam dunia digital).



(a)



(b)



(c)

Gambar 2.5 (a) Bentuk Fisik Transistor, (b) Transistor PNP, (c) Transistor NPN

Transistor pada umumnya mempunyai dua sambungan (*junction*), yaitu emiter-basis dan basis- kolektor. Karena inilah, sebuah transistor sama seperti dua buah dioda. Transistor *npn* disebut juga dioda emiter basis atau singkatnya dioda emiter. Transistor *pnp* juga disebut dioda kolektor basis atau dioda kolektor.

2.2.5. LM 7805

IC ini mempunyai tiga kaki yang digunakan sebagai komponen pendukung dari Vcc untuk menghasilkan tegangan 5V. IC regulator ini berfungsi untuk menstabilkan tegangan 5V dan dapat bekerja dengan baik jika tegangan input (Vin) lebih besar daripada tegangan output (Vout).



Gambar 2.6 IC LM7805

2.2.6. Kamera Webcam

Pada dasarnya setiap kamera terdiri dari tiga bagian utama yaitu :

1. Lensa.

   Lensa mempunyai fungsi untuk memilih bidang pandang tertentu dan ditangkap oleh optik yang menghasilkan gambar untuk diteruskan ke tabung kamera.

2. Tubuh Kamera.

   Tubuh kamera berisi tabung pengambil gambar, yang berfungsi untuk merubah gambar optik yang dihasilkan lensa menjadi sinyal elektrik.

3. *Recorder* / VCR (*Video Casette Recorder*).

   VCR (*Video Casette Recorder*) berfungsi sebagai perekam gambar dan suara, dibeberapa jenis kamera ada yang recorder nya terpisah dan ada juga yang menjadi satu bagian dengan bodi kamera.

Webcam adalah sebuah kamera yang bentuknya lebih kecil dari kamera pada umumnya. Webcam biasanya digunakan untuk mengambil gambar video secara langsung dari komputer, dan biasanya digunakan untuk *chatting*.

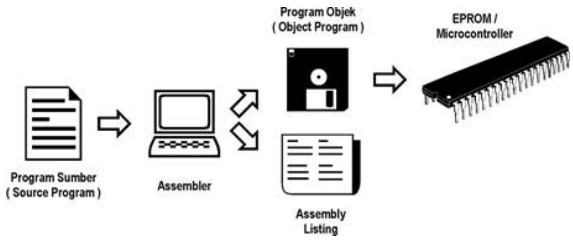


Gambar 2.7 Kamera Webcam

2.3. Bahasa *Assembly*

Assembler adalah program komputer yang mentranslasi program dari bahasa *assembly* ke bahasa mesin. Sedangkan bahasa *assembly* adalah ekuivalensi bahasa mesin. *Mnemonics* atau bisa juga disebut kode operasi (*Opcode*) adalah kode-kode yang akan dikerjakan oleh program *assembler* yang ada pada komputer ataupun mikrokontroler. Kode operasi yang dikerjakan oleh mikrokontroler merupakan perintah-perintah atau instruksi-instruksi yang sangat bergantung dengan jenis mikrokontroler yang digunakan. Contoh, untuk keluarga MCS51 digunakan MOVX, MOV, ADD dan lain-lain.

Sedangkan kode operasi yang dikerjakan oleh program assembler yang ada pada komputer atau *Assembler Directive* sangat bergantung pada program assembler yang digunakan. Contoh, ORG, EQU, DB dan lain-lain.



Gambar 2.8 Proses *Assembly*

Program sumber *assembly* terdiri dari kumpulan baris-baris perintah dan biasanya disimpan dengan extension.ASM dengan 1 baris untuk satu perintah, setiap baris perintah tersebut bisa terdiri atas beberapa bagian (*field*), yakni bagian label, bagian *mnemonic*, bagian *operand* yang bisa lebih dari satu dan terakhir bagian komentar.

Assembler Directive digunakan untuk mengatur kerja dari program *Assembler. Mnemonic* untuk *Assembler Directive* tergantung pada program *Assembler* yang dipakai. Ada beberapa *Assembler Directive* yang umum, yang sama untuk banyak macam program *Assembler* diantaranya adalah :

1. ORG (*Origin)*

   Digunakan untuk menyatakan lokasi memori tempat instruksi atau perintah yang ada di bawahnya disimpan.

2. EQU (*Equate)*

   EQU digunakan untuk mendefinisikan sebuah simbol atau lambang *assembl*er secara bebas.

3. DB (*Define Byte)*

   Digunakan untuk memberikan nilai tertentu pada memori-program.

4. DW (*Define Word)*

   Dipakai untuk memberi nilai 2 byte ke *memory-program* pada baris bersangkutan. *Assembler Directive* ini biasa dipakai untuk membentuk suatu tabel yang isinya adalah nomor-nomor *memory-program*.

5. DS (*Define Storage)*

   Assembler Directive ini dipakai untuk membentuk variabel. Sebagai variabel tentu saja memori yang dipakai adalah *memory-data* (RAM) bukan *memory-program* (ROM).

2.4. Pemograman *Delphi*

2.4.1. Pengenalan *Borland Delphi*

Borland Delphi adalah paket bahasa pemrograman yang mempunyai cakupan kemampuan yang luas dan sangat canggih. Secara umum, kemampuan *Delphi* adalah menyediakan komponen–komponen dan bahasa pemrograman yang andal, sehingga memungkinkan untuk membuat program aplikasi sesuai yang diinginan, dengan tampilan dan kemampuan yang canggih.

Borland Delphi dirilis pertama kali pada bulan Februari 1995, dan sekarang sudah mencapai perkembangannya sampai dengan *Delphi7*. *Delphi* merupakan bahasa pemrograman yang mudah, karena *Delphi* adalah bahasa perograman tingkat tinggi (*high level*) sehingga sangat memudahkan *user* untuk bermain-main di tingkat ini. Pada Pemrograman *Delphi* tinggal *click* dan geser ke *form*, maka jadilah program aplikasi yang diinginkan.

2.4.2. Penggunaan Pascal Pada Delphi

Secara kasar kita bisa menyebut *Delphi* sebagai generasi *Pascal*, sehingga macam- macam *syntax* yang digunakan oleh *Pascal* dapat berguna, dan bisa digunakan lewat *Delphi*. Untuk mengakses *Pascal* kita harus mengubah *compiler* yang digunakan *Delphi*.

2.4.3. Form

Berbeda dengan *Pascal*, pada *Delphi* akan mengenal OOP (*Object oriented programming*), jadi bila pada *Pascal* terlihat tampilan yang menjemukan, pada *Delphi* dapat diatur tampilan semenarik mungkin pada *form* yang digunakan. Caranya hanya menaruh komponen-komponen yang diinginkan pada *form* tersebut, dan memfungsikan masing-masing komponen sesuai dengan yang diinginkan.

2.4.4. Komponen-Komponen *Delphi*

Dalam membuat program, *Delphi* telah menyediakan banyak kemudahan, yaitu dengan disediakannya komponen-komponen. Komponen ini merupakan sebuah *procedure* program yang sudah di *compile* dan langsung dapat digunakan, sesuai dengan fungsinya masing-masing.

Untuk menggunakan komponen ini dengan cara menekan komponen yang diinginkan, kemudian diletakan di *form*, maka komponen tersebut akan muncul di *form*. Kegunaan beberapa komponen:

1. *Button*

   Biasa digunakan sebagai tombol kendali. Perbedaan antara bitbtn dengan btn pada bitbtn dapat disisipkan warna pada tombol dan icon tertentu, lain halnya bila menggunakan btn.

2. *Panel*

   Panel berfungsi untuk mengelompokan komponen-komponen didalamnya.

3. *Label*

   Untuk menamakan atau memberi keterangan pada program.

4. *Edit*

   Edit berfungsi sebagai masukan data (*input*) dalam bentuk *string*, dari bentuk *string* ini dapat mengolahnya menjadi bentuk *integer* atau bentuk lainnya. Yang kemudian dapat digunakan untuk operasi selanjutnya.

5. *Chart*

Data-data yang telah dianalisa, dapat ditampilkan ke dalam grafik, sehinga memudahkan untuk menganalisanya.

6. *String grid*

String grid berguna untuk menaruh data *string* kedalam bentuk kolom tabel, seperti pada excel. Harus mengubah *type* data kedalam bentuk *string* bila data yang ingin ditampilkan bukan data string.

7. *Popup Menu*

Popup Menu berfungsi sebagai perintah yagn aktif bila meng-klik kanan *mouse*, untuk mengaktifkannya harus mengaktifkan *popup menu* pada komponen yang dinginkan, caranya ubah pada *object inspector*.

8. *Main Menu*

Contoh *main menu* adalah *Option* pada tiap aplikasi porgram, dengan komponen ini bisa menaruh fungsi-fungsi program seperti pada aplikasi umumnya.

9. *Combo Box*

Combo Box berfungsi sebagai petunjuk untuk pemilihan berbagai masukan.

10. *Check Box*

Bila komponen ini di *check* maka ada aplikasi yang bisa di *setting* untuk bekerja dibawahnya.

11. *Radio Button*

Prinsip kerjanya hampir sama dengan *check box*, cuma tampilannya saja yang berbeda.

12. *Media Player*

Biasa digunakan untuk menyalakan atau memainkan musik (format *wav* atau *midi*) dan menjalankan film (format *avi*).

13. *Timer*

Timer berfungsi sebagai jam yang telah disediakan *Delphi*. Dengan *timer* kita juga dapat *mendecode time*, sehingga dapat terjadi akuisisi data.

14. *Client socket*

Digunakan di *client* pada saat koneksi ke *server*.

15. *Server socket*

Digunakan di *server*, yang akan menyediakan layanan koneksi ke *client*.

2.4.5. Object Inspector

Object inspector berguna sebagai *options* dari masing-masing komponen. Dengan *object inspector* ini kita dapat memanipulasi komponen yang kita gunakan (walaupun sebenarnya kita juga dapat menggunakannya dengan menuliskannya lewat *text mode*).

Beberapa contoh pada *propertis*:

1. *Font*

   Menunjukan *font* yang akan kita gunakan. Dalam menu ini juga terdapat macam-macam *font* berikut *style* dan ukurannya.

2. *Caption*

   Merupakan judul dari *form* itu. Perhatikan perbedaannya dengan *name*.

3. *Name*

   Merupakan nama dari komponen itu. Biasanya bila kita mengambil sebuah komponen, *Delphi* otomatis menyediakan nama sesuai dengan nama komponen tersebut. Namun nama ini dapat dengan mudah kita ubah agar mudah mengingatnya. Berbeda dengan *caption,* pada *caption* kita tidak mengubah nama komponen, melainkan hanya mengubah judul yang tertera pada komponen tersebut. Sehingga bila kita ingin menggunakan komponen tersebut, yang kita gunakan adalah namanya, dan bukan nama pada *caption*.

4. *Enabled*

   Menunjukan bahwa komponen ini bisa digunakan atau tidak. Perintah ini cukup penting bila kita tidak ingin *user* menekan tombol yang tidak diinginkan.

5. *Height*

   Merupakan tinggi dari komponen yang bakalan kita taruh pada *form*, atau komponen lainnya.

6. *Width*

   Merupakan lebar komponen.

7. *Popup menu*

   Fungsi ini berguna saat kita mengkilik kanan dan akan tampil perintah. Pada penggunaanya kita mengambuil komponen *popup menu* dan menset *popup menu* pada *form*.

8. *Auto size*

   Fungsi ini berisi mengenai apakah form yang kita miliki bisa diperbesar atau tidak.

9. *Text*

   Biasanya tedapat pada komponen *edit* dan *mask edit*. Kegunaannya untuk menaruh kata-kata pada komonen ini.

2.4.6 DSPack234

DSPack adalah suatu komponen dan aplikasi multimedia menggunakan *MS Direct Show* dan teknologi *DirectX*. DSPack didisain untuk bekerja dengan *DirectX 9* di sistem operasi *Win9X, ME, Win2000,* dan *Win XP*, dan *DSPack* juga didisain untuk digunakan di pemrograman *Delphi* 5, *Delphi* 6, *Delphi* 7, dan *C++ Builder*.

DSPack pada pemrograman seperti delphi digunakan sebagai komponen tambahan untuk program yang menggunakan kamera sebagai tampilan.

2.4.7 Tcomport Driver

Tcomport Driver adalah suatu komponen tambahan dalam bahasa pemrograman *Delphi*, komponen ini digunakan untuk aplikasi penghubung antara mikrokontroler dan PC yang menggunakan program *Delphi* sebagai penghubungnya.