

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Parkir**

Menurut PP No. 43 tahun 1993 parkir didefinisikan sebagai kendaraan yang berhenti pada tempat-tempat tertentu baik yang dinyatakan dengan rambu atau tidak, serta tidak semata-mata untuk kepentingan menaikkan atau menurunkan orang dan atau barang. Sedangkan definisi lain tentang parkir adalah keadaan dimana suatu kendaraan berhenti untuk sementara (menurunkan muatan) atau berhenti cukup lama (Warpani, 1990).

Beberapa definisi parkir dari beberapa sumber diantaranya adalah sebagai berikut :

1. Menurut Poerwadarmita (1976), parkir adalah tempat pemberhentian kendaraan beberapa saat.
2. Pignataro (1973) dan Sukanto (1985) menjelaskan bahwa parkir adalah memberhentikan dan menyimpan kendaraan (mobil, sepeda motor, sepeda, dan sebagainya) untuk sementara waktu pada suatu ruang tertentu. Ruang tersebut dapat berupa tepi jalan, garasi atau pelataran yang disediakan untuk menampung kendaraan tersebut.
3. Dijelaskan dalam buku peraturan lalu lintas (1998) pengertian dari parkir yaitu tempat pemberhentian kendaraan dalam jangka waktu yang lama atau sebentar tergantung kendaraan dan kebutuhan.
4. Sedangkan menurut Kepmen Perhub No. 4 Th. 1994, parkir adalah keadaan tidak bergerak suatu kendaraan yang tidak bersifat sementara.

#### **2.2. Fingerscan**

*Fingerscan* adalah terminal pengesahan yang memeriksa identitas seorang dari gambar sidik jarinya. Setiap orang mempunyai sesuatu yang bersifat unik, sidik jari yang berbeda. Sistem ini membuat pengenalan melalui metode pola sidik jari, alur dan titik simpul serta garis akhir pola garis (Collins, 2001).

Penggunaan *fingerscan* memiliki keuntungan diantaranya:

- a. Sidik jari tidak dapat dipalsukan atau digandakan.
- b. Kesalahan pencatatan dan manipulasi data dapat diminimalkan.
- c. Sistem pelaporan terintegrasi dengan sistem informasi kepegawaian.



**Gambar 2.1.** *Fingerscan*

*Fingerscan* memiliki tiga proses yaitu proses pemasukan data, proses identifikasi, dan proses verifikasi. Dalam proses pemasukan data, program akan menyimpan identitas yang berkaitan dengan sidik jari seperti nama dan *finger ID*. Proses identifikasi merupakan proses pencocokan yang dilakukan dengan mencari data dari *database* yang cocok dengan sidik jari yang di akuisisi terhadap sampel sidik jari. Program kemudian akan melakukan proses pencocokan dengan data yang tersedia pada *database*. Proses verifikasi merupakan proses pencocokan data transaksi dan *image* dalam *database* yang tersimpan dari identitas seseorang (Collins, 2001).

## **2.3. Sidik Jari**

### **2.3.1. Pengertian Sidik Jari**

Sidik jari merupakan salah satu bagian biometrik pada makhluk hidup selain darah, retina, dan DNA (*Deoxiribo Nuclead Acid*) yang tidak bisa ditiru ataupun digandakan.

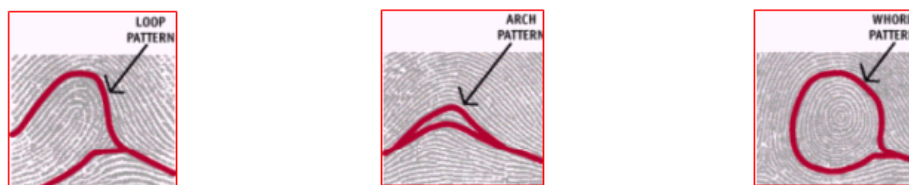
Sidik jari adalah gurat-gurat yang terdapat di kulit ujung jari yang berfungsi untuk memberi gaya gesek lebih besar agar jari dapat memegang benda-benda lebih erat. Sidik jari dapat digunakan sebagai sarana pengamanan dalam melakukan akses ke komputer karena sidik jari mempunyai ciri yang unik, setiap manusia memilikinya, dan selalu ada perbedaan antara yang satu dengan yang lain (Angraini, 2009).

### 2.3.2. Pola Sidik Jari

Berdasarkan klasifikasi, pola sidik jari dapat dinyatakan secara umum kedalam lima bentuk, yaitu (Collins, 2001) :

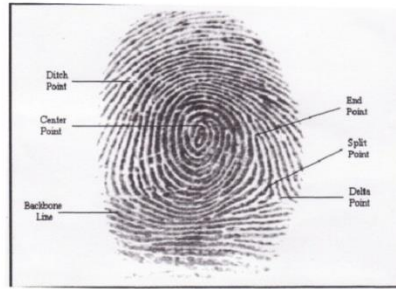
- a. *Loop kiri*
- b. *Loop kanan*
- c. *Arch*
- d. *Tented arch*
- e. *Whorl*

Namun dari pola yang ada diatas ada tiga pola yang paling umum yaitu *whorl*, *arch* dan *loop*.



**Gambar 2.2.** Pola Sidik Jari

Beberapa pola ini dapat dibedakan dengan kasat mata namun ada beberapa karakteristik sidik jari lainnya yang kadang tidak bisa dibedakan karena memiliki pola yang sama, masing-masing memiliki perubahan arah, percabangan yang berbeda, hal tersebut yang menyebabkan anatomi sidik jari ini memiliki beberapa karakteristik yang disebut *ridge* (Collins, 2001).



**Gambar 2.3.** Gambar Anatomi Sidik Jari Manusia

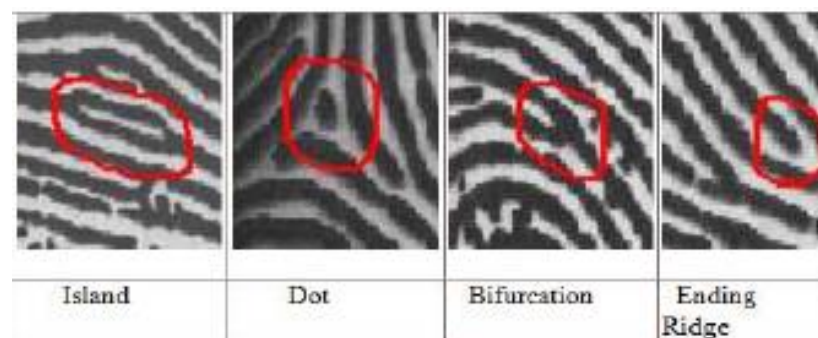
Keterangan :

1. *Ditch Point* : Titik terdalam dari pola sidik jari
2. *Center Point* : Titik paling tengah dimana pola sidik jari berawal.
3. *Backbone Line* : Garis paling tebal pada pola sidik jari.
4. *End Point* : Titik akhir atau terputusnya pola sidik jari.
5. *Split Point* : Titik yang membentuk cabang dari pola sidik jari.
6. *Delta Point* : Titik terdalam pada percabangan (terletak didaerah yang terpisah dari dua jenis garis).

Sensor sidik jari akan menangkap kontur kulit jari. Kontur jari adalah hal yang sangat penting karena bisa menghindari kecurangan pemalsuan dengan menggunakan *photocopy* sidik jari, dengan karet, atau cetakan yang lain.

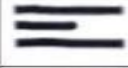
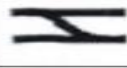










### 2.3.3. Karakteristik Sidik Jari

Identifikasi oleh sidik jari juga memiliki karakteristik pola tertentu, ada tiga karakteristik pola dasar yaitu *ridge ending*, bifurkasi, titik dan pulau.



**Gambar 2.4.** Karakteristik Sidik Jari

Tabel 2.1. Variasi Karakter Sidik Jari

| <i>Basic and composite ridge characteristics</i> |                                                                                   |                                         |                                                                                     |
|--------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------|-------------------------------------------------------------------------------------|
| <i>pola</i>                                      | <i>Example</i>                                                                    | <i>pola</i>                             | <i>Example</i>                                                                      |
| <i>ridge ending</i>                              |  | <i>bridge</i>                           |  |
| <i>bifurcation</i>                               |  | <i>double bifurcation</i>               |  |
| <i>dot</i>                                       |  | <i>trifurcation</i>                     |  |
| <i>island ridge</i> (short)                      |  | <i>opposed bifurcations</i>             |  |
| <i>lake (enclosure)</i>                          |  | <i>ridge crossing</i>                   |  |
| <i>hook (spur)</i>                               |  | <i>opposed bifurcation/ridge ending</i> |  |

#### 2.4. Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah *chip*. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan *input output*. Mikrokontroler merupakan komputer didalam *chip* yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini.

Mikrokontroler dapat diartikan lain yaitu sebuah versi mini atau mikro dari sebuah komputer karena mikrokontroler sudah mengandung beberapa peripheral yang langsung bisa dimanfaatkan, misalnya *port parallel*, *port serial*, komparator, konversi digital ke analog (DAC), konversi analog ke digital (ADC) dan sebagainya hanya menggunakan sistem minimum yang tidak rumit atau kompleks. Secara teknik, hanya ada 2 macam mikrokontroler. Pembagian ini didasarkan pada kompleksitas instruksi-instruksi yang dapat diterapkan pada mikrokontroler tersebut. Pembagian itu yaitu RICS (*Reduce Instruction Set Computer*) yaitu instruksi yang dimiliki terbatas, tetapi memiliki fasilitas yang lebih banyak contohnya mikrokontroler keluarga MCS51 yaitu AT89S52. CISC (*Complex*

*Instruction Set Computer*) yaitu instruksi bisa dikatakan lebih lengkap tapi dengan fasilitas secukupnya contohnya mikrokontroler keluarga AVR yaitu ATmega8535 (Budiharto, 2004).

Mikrokontroler disebut sebagai "*one chip solution*" karena terdiri dari :

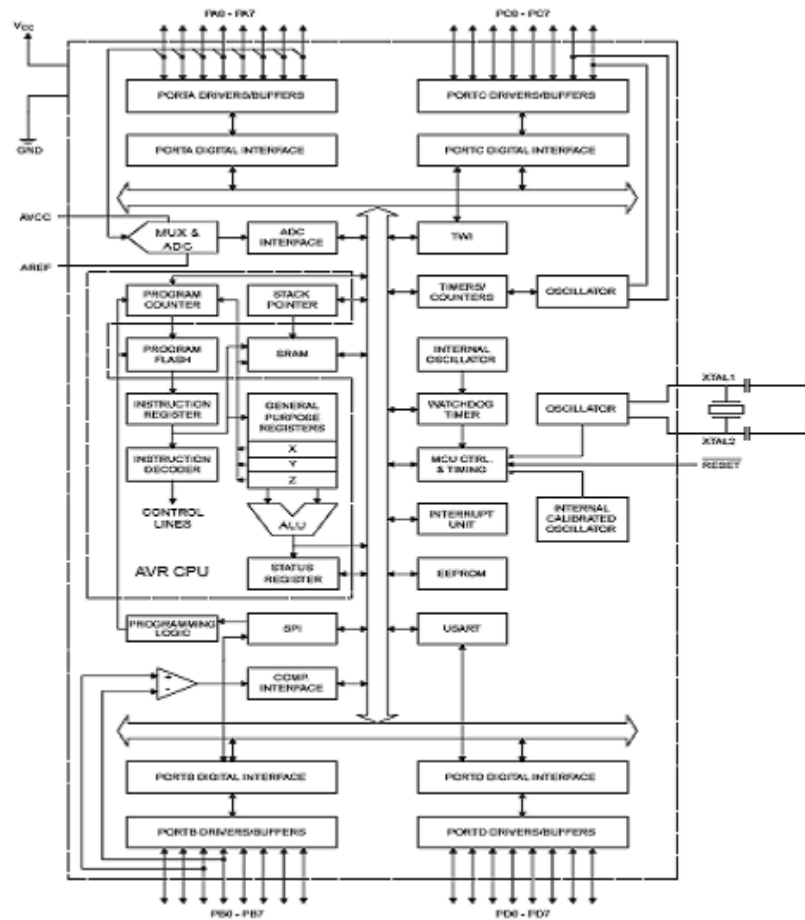
- CPU
- RAM
- EPROM/PROM/ROM
- I/O (*Input/Output*) - *serial* dan *parallel*
- *Timer*
- *Interrupt Controller*

Mikrokontroler AVR (*Alf and Vegard's Risc processor*) memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode *16-bit* (*16-bits word*) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus *clock* atau dikenal dengan teknologi RISC (*Reduced Instruction Set Computing*), berbeda dengan instruksi MCS51 yang membutuhkan 12 siklus *clock* atau dikenal dengan teknologi CISC (*Complex Instruction Set Computing*).

Secara umum, AVR dapat dikelompokkan ke dalam 4 kelas, yaitu keluarga AT90Sxx, keluarga ATmega dan AT86RFxx. Dari segi arsitektur dan perintah yang digunakan adalah sama, yang membedakannya adalah kelas memori, *peripheral* dan fungsinya.

## **2.5. Mikrokontroler ATmega8535**

Mikrokontroler tipe AVR terdiri dari 3 jenis yaitu AT Tyny, AVR Klasik, dan AT Mega. Perbedaannya hanya pada fasilitas dan I/O yang tersedia serta fasilitas lain seperti ADC, EEPROM dan lain sebagainya, salah satu jenisnya mikrokontroler ATmega8535. ATmega8535 memiliki teknologi RICS (*Reduce Instruction Set Computer*) dengan kecepatan maksimal 16 MHz membuat ATmega8535 lebih cepat dibandingkan dengan varian MCS51. Adapun blok diagram ATmega8535 adalah sebagai berikut (Budiharto, 2004) :



**Gambar 2.5.** Blok Diagram ATmega8535  
(Budiharto, 2004)

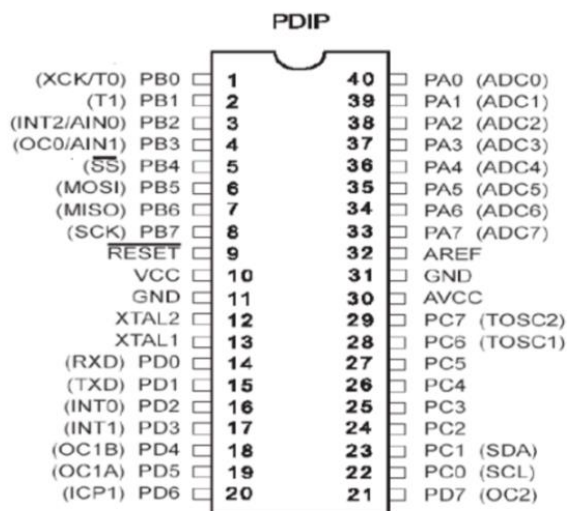
### 2.5.1. Arsitektur Mikrokontroler ATmega8535

Fitur yang tersedia pada ATmega 8535 adalah sebagai berikut (Heryanto, M. Ary, dkk, 2008:1) :

1. 8 bit AVR berbasis RISC dengan performa tinggi dan konsumsi daya rendah.
2. Kecepatan maksimal 16 Mhz
3. Memori :
  - a. 8 Kb *Flash*,
  - b. 512 *byte* SRAM,
  - c. 512 *byte* EEPROM
4. *Timer/Counter* :
  - a. 2 buah 8 bit *timer/counter*,

- b. 1 buah 16 bit *timer/counter*,
- c. 4 kanal PWM
- 5. 8 kanal 10/8 bit ADC
- 6. *Programmable Serial* USART
- 7. Komparator Analog
- 8. 6 pilihan *sleep mode* untuk penghematan daya listrik
- 9. 32 jalur I/O yang bisa di program

### 2.5.2. Konfigurasi Pin Mikrokontroler ATmega8535



**Gambar 2.6.** *PinOut* ATmega8535

Penjelasan dari masing-masing kaki adalah sebagai berikut (Heryanto, M. Ary, dkk, 2008:3) :

1. Power, VCC dan GND (*Ground*).
2. *Port* A (PA7..PA0) (kaki 32-39) Merupakan port 8 bit dua arah (*bidirectional*) I/O. *Port* ini berfungsi sebagai *port* data/alamat I/O ketika menggunakan SRAM eksternal.
3. *Port* B (PB7..PB0) (kaki 1-8) merupakan *port* 8 bit dua arah (*bidirectional*) I/O, untuk berbagai keperluan (*multi purpose*)



4. *Port C* (PC7..PC0) (kaki 21-28) adalah *port* 8 bit dua arah I/O, dengan *internal pull-up resistor*. *Port C* ini juga berfungsi sebagai *port* alamat ketika menggunakan SRAM eksternal.
5. *Port D* (PD7..PD0) (kaki 10-17) adalah *port* 8 bit dua arah I/O dengan resistor *pull-up* internal. *Port D* juga dapat berfungsi sebagai terminal khusus.
6. RESET (kaki 9) ketika kondisi rendah rendah yang lebih lama dari 50 nS mikrokontroler akan reset walaupun detak tidak berjalan.
7. XTAL1 (kaki 19) masukan bagi penguat osilator terbalik dan masukan bagi rangkaian operasi detak internal dan XTAL2 (kaki 18) Keluaran dari penguat osilator terbalik.
8. AREF adalah pin masukan untuk tegangan referensi eksternal ADC.

## 2.6. Motor DC

Motor DC adalah suatu motor yang mengubah energi listrik searah menjadi mekanis yang berupa tenaga penggerak torsi. Motor DC digunakan dimana kontrol kecepatan dan kecepatan torsi diperlukan untuk memenuhi kebutuhan. Bagian motor DC yang paling penting adalah rotor (bagian yang berputar) dan stator (bagian yang tidak berputar). Bagian stator adalah badan motor, sikat-sikat dan inti kutub magnet. Bagian rotor adalah bagian yang berputar dari suatu motor DC, yaitu ialah lilitan jangkar, jangkar, komutator, tali isolator, poros, bantalan dan kipas. (Harahap, 1996).



**Gambar 2.7.** Motor DC

### 2.6.1. Prinsip Kerja Motor DC

Prinsip kerja motor DC yaitu suatu kumparan atau lilitan kawat yang dialiri arus listrik untuk memperkuat medan magnetic akan mendapatkan gaya yang dikeluarkan medan magnet tersebut dengan arah tegak lurus pada garis medan yang dialiri arus Motor DC biasanya digunakan dalam rangkaian yang memerlukan kepresisian yang tinggi untuk pengaturan kecepatan, pada torsi yang konstan. Semua motor DC beroperasi atas dasar arus yang melewati konduktor yang berada dalam medan magnet. Motor DC disini digunakan sebagai motor penggerak utama (Harahap, 1996).

### 2.7. Driver Motor DC

*Driver* motor ini berfungsi untuk mengendalikan motor DC. Untuk menggerakkan motor DC ini biasanya membutuhkan arus yang besar, untuk itu digunakan *H-bridge* yang terdiri dari rangkaian penguat transistor atau menggunakan IC penguat daya sebagai pengendali putaran motor.

Driver motor digunakan untuk menggerakkan motor DC menggunakan mikrokontroler. Arus yang mampu diterima atau yang dikeluarkan oleh mikrokontroler sangat kecil (dalam satuan *miliampere*) sehingga agar mikrokontroler dapat menggerakkan motor DC diperlukan suatu rangkaian *driver* motor yang mampu mengalirkan arus sampai dengan beberapa *ampere*. Rangkaian *driver* motor DC dapat berupa rangkaian *transistor*, *relay*, atau IC (*Integrated Circuit*).

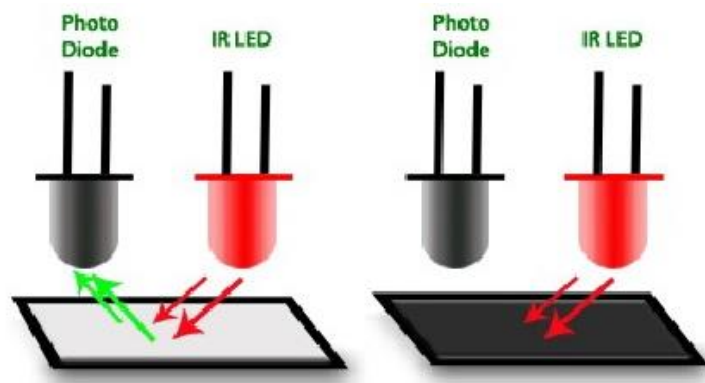
### 2.8. Sensor Infra Merah

Infra merah adalah radiasi elektromagnetik dari panjang gelombang lebih panjang dari cahaya tampak, tetapi lebih pendek dari radiasi gelombang radio. Infra merah berarti "bawah merah" (dari bahasa Latin *infra*, "bawah"), merah merupakan warna dari cahaya tampak dengan gelombang terpanjang.

Cahaya infra merah merupakan cahaya yang tidak tampak. Jika dilihat dengan spektroskop cahaya maka radiasi cahaya infra merah akan terlihat pada spektrum electromagnet. Radiasi infra merah memiliki panjang gelombang antara

700 nm sampai 1 mm dan berada pada spektrum berwarna merah. Dengan panjang gelombang ini maka cahaya infra merah tidak akan terlihat oleh mata namun radiasi panas yang ditimbulkannya masih dapat dirasakan atau dideteksi.

Pada dasarnya komponen yang menghasilkan panas juga menghasilkan radiasi infra merah termasuk tubuh manusia maupun tubuh binatang. Cahaya infra merah, walaupun mempunyai panjang gelombang yang sangat panjang tetap tidak dapat menembus bahan-bahan yang tidak dapat melewati cahaya yang nampak sehingga cahaya infra merah tetap mempunyai karakteristik seperti halnya cahaya yang nampak oleh mata. Pada pembuatan komponen yang dikhususkan untuk penerima infra merah, lubang untuk menerima cahaya (*window*) sudah dibuat khusus sehingga dapat mengurangi interferensi dari cahaya non-infra merah. Oleh sebab itu sensor infra merah yang baik biasanya memiliki jendela (pelapis yang terbuat dari silikon) berwarna biru tua keungu-unguan. Sensor ini biasanya digunakan untuk aplikasi infra merah yang digunakan diluar rumah (*outdoor*).



**Gambar 2.8.** Sensor Infra Merah

Sinar infra merah yang dipancarkan oleh pemancar infra merah tentunya mempunyai aturan tertentu agar data yang dipancarkan dapat diterima dengan baik pada penerima. Oleh karena itu baik di pengirim infra merah maupun penerima infra merah harus mempunyai aturan yang sama dalam mentransmisikan (bagian pengirim) dan menerima sinyal tersebut kemudian mendekodekannya kembali menjadi data *biner* (bagian penerima). Komponen yang dapat menerima infra merah ini merupakan komponen yang peka cahaya yang dapat berupa dioda

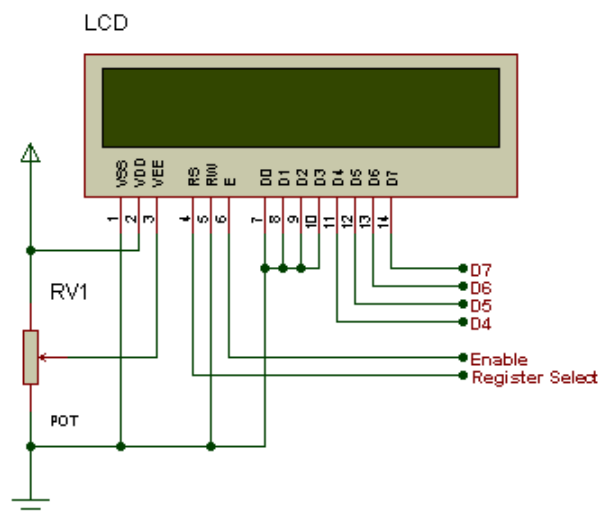
(*photodiode*) atau transistor (*phototransistor*). Komponen ini akan merubah energi cahaya, dalam hal ini energi cahaya infra merah, menjadi pulsa-pulsa sinyal listrik. Komponen ini harus mampu mengumpulkan sinyal infra merah sebanyak mungkin sehingga pulsa-pulsa sinyal listrik yang dihasilkan kualitasnya cukup baik.

## 2.9. Liquid Crystal Display (LCD)

*Liquid Crystal Display* merupakan rangkaian elektronika yang digunakan untuk menampilkan keterangan atau indikator yang diberikan ke dalam mikrokontroler (Suyadhi, 2010).

Penggunaan perangkat LCD sebagai peraga pada alat ini karena LCD banyak memiliki kelebihan :

1. Pemakaian arusnya kecil.
2. Dapat menampilkan semua simbol ASCII maupun simbol yang dibuat sendiri.
3. Pengendaliannya sangat mudah karena sudah dilengkapi dengan unit pengendali didalam.
4. Mudah dirangkakan ke sistem mikrokontroler.



**Gambar 2.9.** *Liquid Crystal Display (LCD)*

Berikut ini adalah tabel konfigurasi pin dari LCD 16 x 2 :

**Tabel 2.2.** Konfigurasi pin LCD 16x2

| Pin | Simbol | Level | Tujuan           | Fungsi                                                   |
|-----|--------|-------|------------------|----------------------------------------------------------|
| 1   | VSS    | -     | Power Supply     | Ground                                                   |
| 2   | VDD    | -     | Power Supply     | Tegangan Supply (+5Volt)                                 |
| 3   | VLS    | -     | Power Supply     | Power supply untuk mendrive LCD guna mengatur kontrasnya |
| 4   | RS     | H/L   | $\mu$ C          | H : Data ; L : Instruction Code                          |
| 5   | R/W    | H/L   | $\mu$ C          | H : Read ; L : Write                                     |
| 6   | E      | H/L   | $\mu$ C          | Enable                                                   |
| 7   | DB0    | H/L   | $\mu$ C          | Data Bus Line                                            |
| 8   | DB1    | H/L   | $\mu$ C          | Data Bus Line                                            |
| 9   | DB2    | H/L   | $\mu$ C          | Data Bus Line                                            |
| 10  | DB3    | H/L   | $\mu$ C          | Data Bus Line                                            |
| 11  | DB4    | H/L   | $\mu$ C          | Data Bus Line                                            |
| 12  | DB5    | H/L   | $\mu$ C          | Data Bus Line                                            |
| 13  | DB6    | H/L   | $\mu$ C          | Data Bus Line                                            |
| 14  | DB7    | H/L   | $\mu$ C          | Data Bus Line                                            |
| 15  | V+BL   | -     | Back Ligh Supply | Tegangan Supply (+5 Volt)                                |
| 16  | V+BL   | -     | Back Ligh Supply | Ground                                                   |

Karakteristik yang ada pada LCD antara lain :

- Mempunyai 16 karakter dengan 2 baris tampilan yang terbentuk dari matrik titik (*dot matrix*).
- Duty ratio* : 1/16
- ROM pembangkit karakter untuk 192 jenis karakter dengan bentuk karakter huruf : 5 x 7 matrik titik.
- Mempunyai 8 tipe *RAM* pembangkit karakter.
- RAM* data tampilan beberapa dan *RAM* pembangkit karakter dapat dibaca dari unit Mikrokontroler.
- Dilengkapi dengan beberapa perintah yaitu penghapusan tampilan, posisi awal kursor, tampilan karakter kedip (*display clear*), posisi awal kursor (*cursor home*), tampilan karakter kedip (*display character blink*), dan penggeseran tampilan (*display shift*).
- Rangkaian pembangkit detak (*clock*) internal.

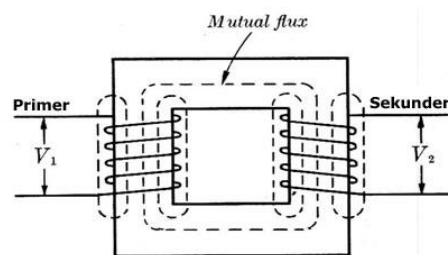
- h. Catu daya tunggal +5V.
- i. Rangkaian otomatis rest saat daya dihidupkan.
- j. Pemrosesan dengan CMOS.
- k. Jangkauan suhu 0°C sampai 50° C.

## 2.10. Power Supply

*Power supply* adalah suatu sistem yang dapat bekerja mengkonversikan tegangan arus bolak – balik (AC) ketegangan searah (DC) pada nilai tertentu. Dalam setiap peralatan elektronika, *power supply* merupakan bagian yang terpenting dalam suatu sistem rangkaian elektronika agar rangkaian tersebut dapat digunakan. Rangkaian *power supply* memberikan masukan tegangan pada alat pengendali.

## 2.11. Transformator

*Transformator* adalah suatu alat untuk mempertinggi atau memperendah suatu tegangan bolak-balik. Pada dasarnya sebuah transformator terdiri dari sebuah kumparan primer dan sebuah kumparan sekunder yang digulung pada sebuah inti besi lunak. Arus bolak-balik pada kumparan primer menimbulkan medan magnet yang berubah-ubah dalam inti besi. Medan magnet ini menginduksi GGL bolak-balik dalam kumparan sekunder (Budiman, 1992).



**Gambar 2.10.** *Transformator*

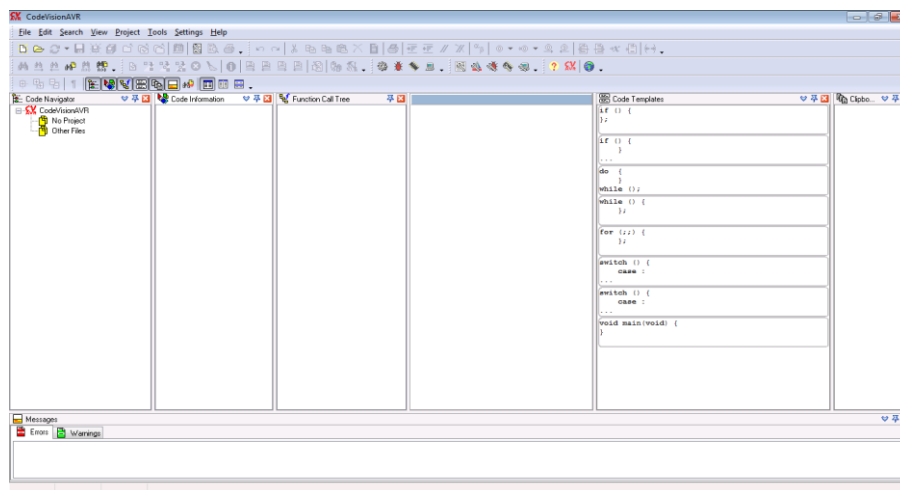
Prinsip kerja tranformator adalah sebagai berikut:

1. Kumparan primer dihubungkan kepada sumber tegangan yang hendak diubah besarnya. Karena tegangan primer itu tegangan bolak-balik, maka besar dan arah tegangan itu berubah-ubah.

2. Dalam inti besi timbul medan magnet yang besar dan arahnya berubah-ubah pula. Perubahan medan magnet ini menginduksi tegangan bolak-balik pada kumparan sekunder.

## 2.12. Code Vision AVR

*Code Vision AVR* merupakan salah satu *software compiler* yang khusus digunakan untuk keluarga mikrokontroler. Meskipun *Code Vision AVR* termasuk *software* komersial, namun tetap dapat menggunakannya dengan mudah karena terdapat versi evaluasi yang tersedia secara gratis walaupun dengan kemampuan yang dibatasi (Soebhakti,2009).



**Gambar 2.11.** Tampilan Awal pada *Code Vision AVR*

*Code Vision AVR* merupakan yang terbaik bila dibandingkan dengan kompilator-kompilator yang lain karena beberapa kelebihan yang dimiliki oleh *Code Vision AVR* antara lain :

1. Menggunakan IDE (*Intergrated Development Environment*).
2. Fasilitas yang disediakan lengkap (mengedit program, meng-*compile* program, men-*download* program) serta tampilannya yang terlihat menarik dan mudah dimengerti. Kita dapat mengatur settingan editor sedemikian rupa sehingga membantu memudahkan kita dalam penulisan program.
3. Mampu membangkitkan kode program secara otomatis dengan menggunakan fasilitas *Code Wizard AVR*.

4. Memiliki fasilitas untuk *men-download* program langsung dari *Code Vision AVR* dengan menggunakan *hardware* khusus seperti Atmel STK500, Kanda Sysrem STK200+ / 300 dan beberapa *hardware* lain yang telah didefinisikan oleh *Code Vision AVR*.
5. Memiliki fasilitas *debugger* sehingga dapat menggunakan *software compiler* lain untuk mengecek kode *assembler*-nya, contohnya AVRStudio.
6. Memiliki terminal komunikasi serial yang terintegrasi dalam *Code Vision AVR* sehingga dapat digunakan untuk membantu pengecekan program yang telah dibuat khususnya yang menggunakan fasilitas komunikasi serial UART.

### 2.13. Bahasa Pemrograman C

Bahasa C luas digunakan untuk pemrograman berbagai jenis perangkat, termasuk mikrokontroler. Bahasa ini sudah merupakan bahasa pemrograman tingkat menengah dimana memudahkan *programmer* menuangkan algoritmanya. Bahasa C luas digunakan untuk pemrograman berbagai jenis perangkat, termasuk mikrokontroler. Berikut adalah dasar dari bahasa C :

1. *Preprocessor*(#) : Digunakan untuk memasukkan (*include*) *text* dari *file* lain, mendefinisikan macro dapat mengurangi beban kerja pemrograman dan meningkatkan *legibility source code* (mudah dibaca).

Contoh : `#include <delay.h>`

2. *#define* : digunakan untuk mendefinisikan macro.

Contoh : `#define T1 PINA.0`

3. Komentar

Penulisan komentar untuk beberapa baris komentar sekaligus

`/*`

`...komentar`



#### 4. Deklarasi variabel & konstanta

- a. Variabel adalah memori penyimpanan data yang nilainya dapat diubah-ubah.

Penulisan : [tipe data] [nama] = [nilai\_awal] ;'

- b. Konstanta adalah memori penyimpanan data yang nilainya tidak dapat diubah.

Penulisan : const [tipe data] [nama] = [nilai] ;

c. Tambahan:

- Global variabel/konstanta yang dapat diakses di seluruh bagian program.
- Lokal variabel/konstanta yang hanya dapat diakses oleh fungsi tempat dideklarasikannya.

#### 5. Tipe Data

**Tabel 2.3.** Tipe Data

| Type              | Size (Bits) | Range                               |
|-------------------|-------------|-------------------------------------|
| Bit               | 1           | 0, 1                                |
| Bool, _bool       | 8           | 0, 1                                |
| Char              | 8           | -128 to 127                         |
| Unsigned char     | 8           | 0 to 255                            |
| Signed char       | 8           | -128 to 127                         |
| Int               | 16          | -32768 to 32767                     |
| Short int         | 16          | -32768 to 32767                     |
| Unsigned int      | 16          | 0 to 65535                          |
| signed char       | 16          | -32768 to 32767                     |
| Long int          | 32          | -2147483648 to 2147483647           |
| Unsigned long int | 32          | 0 to 4294967295                     |
| Signed char       | 32          | -2147483648 to 2147483647           |
| Float             | 32          | $\pm 1.175e - 38$ to $\pm 3.402e38$ |
| Double            | 32          | $\pm 1.175e - 38$ to $\pm 3.402e38$ |

#### 6. Percabangan dan pengulangan

- a. *if else* : digunakan untuk penyeleksian kondisi.
- b. *for* : digunakan untuk *looping* dengan jumlah yang sudah diketahui.
- c. *while* : digunakan untuk *looping* jika dan selama memenuhi syarat tertentu.

d. *do while* : digunakan untuk *looping* jika dan selama memenuhi syarat tertentu.

e. *switch case* : digunakan untuk seleksi dengan banyak kondisi

## 7. Prosedur & Fungsi

Prosedur & Fungsi adalah bagian program yang dapat dipanggil oleh program utama. Bedanya kalau prosedur memberikan hasil yang tidak memiliki nilai balik melainkan berupa proses sedangkan fungsi memberikan hasil yang memiliki nilai balik yaitu berupa nilai.

## 8. Statement

*Statement* adalah setiap operasi dalam pemrograman, harus diakhiri dengan [ ; ] atau [ } ]. *Statement* tidak akan dieksekusi bila diawali dengan tanda [ // ] untuk satu baris. Lebih dari 1 baris gunakan pasangan [ /\* ] dan [ \*/ ]. *Statement* yang tidak dieksekusi disebut juga *comments* / komentar.

Contoh : suhu=adc/255\*100; //contoh rumus perhitungan suhu

## 9. Operasi Aritmetika

**Tabel 2.4.** Daftar Operator Kondisi

| Operator       | Keterangan                                                                                                       |
|----------------|------------------------------------------------------------------------------------------------------------------|
| +, -, *, /     | Tambah, kurang, kali dan bagi                                                                                    |
| +=, -=, *=, /= | Nilai di sebelah kiri operator di tambah, dikurangi, dikali, atau dibagi dengan nilai di sebelah kanan operator. |
| %              | Sisa pembagian                                                                                                   |
| ++, --         | Ditambah 1 ( <i>increment</i> ) atau dikurangi satu ( <i>decrement</i> )                                         |

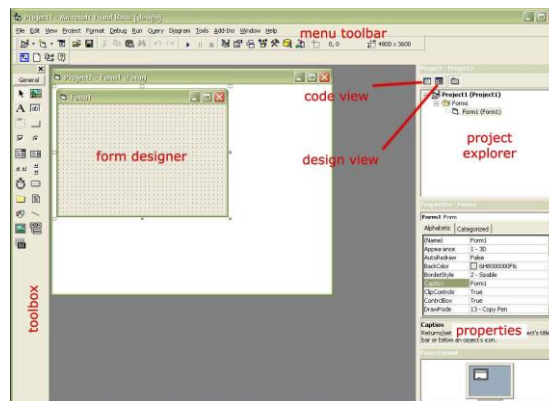
### 2.14. Visual Basic

*Visual Basic* adalah salah satu *development tools* untuk membangun aplikasi dalam lingkungan *windows*. Dalam pengembangan aplikasi, *visual basic* menggunakan pendekatan visual untuk merancang *user interface* atau tampilan dalam bentuk *form*, sedangkan untuk kodingnya menggunakan bahasa *basic* yang

cenderung mudah dipelajari. *Visual basic* telah menjadi *tools* yang terkenal bagi para pemula maupun *developer*. Dalam lingkungan *windows user interface* memegang peranan penting, karena dalam pemakaian aplikasi, pemakai senantiasa berinteraksi dengan *user interface* tanpa menyadari bahwa di belakangnya berjalan instruksi-instruksi program yang mendukung tampilan dan proses yang dilakukan (Daryanto. 2003).

*Visual Basic* merupakan salah satu *software* untuk membuat program yang cukup sederhana tetapi banyak cakupan yang dapat dikerjakan, karena *visual basic* dapat mengakses banyak *software* seperti *Excel*, *Access* dan sebagainya. *Visual basic* lebih sederhana dari pemrograman yang lain. Kesederhanaan *visual basic* terletak pada kemudahan membuat bahasa pemrograman dan bentuk tampilan yang dikehendaki. *Visual Basic* ini merupakan pengembangan bahasa *basic* yang diterapkan pada program yang berbasis *Windows*.

### 2.14.1. Struktur Aplikasi Visual Basic



**Gambar 2.12.** *Jendela Kerja Standar Visual Basic*

#### a. Title Bar

*Title bar* merupakan batang jendela dari program *visual basic* yang terletak pada bagian paling atas dari jendela program yang berfungsi untuk menampilkan judul atau nama jendela. Selain itu juga berfungsi untuk memindahkan posisi jendela dengan menggunakan *drag and drop* pada posisi *title bar* tersebut dan

untuk mengatur ukuran jendela dari ukuran *minimize* ke ukuran *restore* ataupun sebaliknya dengan melakukan klik ganda pada posisi *title bar* tersebut.

#### b. Menu Bar

*Menu bar* merupakan batang menu yang terletak di bawah *title bar* yang berfungsi untuk menampilkan pilihan menu atau perintah untuk mengoperasikan program *visual basic*. Saat pertama kali jendela program *visual basic* terbuka dapat dilihat tiga belas menu utama yaitu *file, edit, view, project, format, debug, run, query, diagram, tools, addins, window, dan help*. *Menu bar* mempunyai sederetan pilihan menu yang masing-masing mempunyai arti dan fungsi yang berbeda.

#### c. Toolbars

*Toolbars* merupakan sebuah batang yang berisi kumpulan tombol yang terletak dibagian bawah *menu bar* yang dapat digunakan untuk menjalankan sebuah perintah. Pada kondisi *default* program *visual basic* hanya menampilkan *toolbars* standar.

#### d. Project Explorer

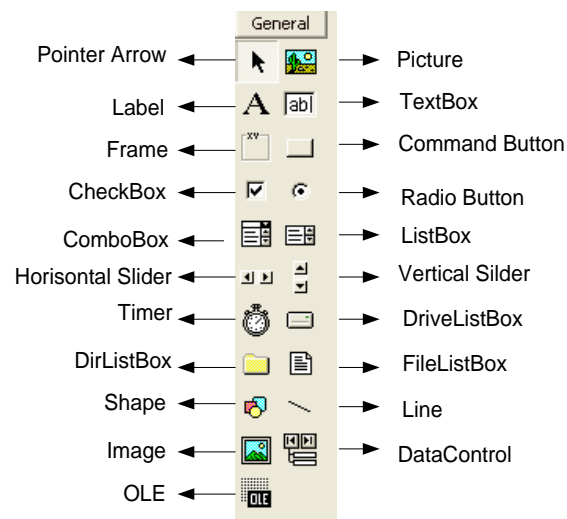
*Project Explorer* merupakan suatu kumpulan modul atau merupakan program aplikasi itu sendiri. Dalam *visual basic*, *file project* disimpan dengan nama *file* berakhiran *vbp*, dimana *file* ini berfungsi untuk menyimpan seluruh komponen program. Apabila membuat suatu program aplikasi baru maka secara otomatis *project* tersebut akan diisi dengan *form* baru. Dalam jendela *project explorer* ditampilkan suatu struktur hirarki dari sebuah *project* itu sendiri yang berisi semua item yang terkandung di dalamnya.

#### e. Form

*Form* merupakan *windows* atau jendela di mana akan dibuat *user interface*.

f. Toolbox atau control

*Toolbox* merupakan tampilan berbasis grafis yang dimasukkan pada *form* untuk membuat interaksi dengan pemakai. Bentuk *toolbox visual basic* adalah sebagai berikut.



**Gambar 2.13.** *Toolbox Visual Basic 6.0.*

Adapun secara garis besar fungsi dari masing-masing kontrol tersebut adalah sebagai berikut.

1. Pointer

*Pointer* bukan merupakan suatu kontrol tapi icon ini digunakan ketika memilih kontrol yang sudah berada pada form.

2. PictureBox

*PictureBox* adalah kontrol yang digunakan untuk menampilkan image dengan format BMP, DIB, (*Bitmap*), ICO (*Icon*), CUR (*Cursor*), WMF (*Metafile*), CMF (*Enhanced Metafile*), GIF, JPEG.

3. Label

*Label* adalah kontrol yang digunakan untuk menampilkan teks yang tidak dapat diperbaiki.

#### 4. Textbox

*TextBox* adalah kontrol yang mengandung string yang dapat dipakai oleh pemakai, dapat berupa satu baris tunggal atau banyak baris.

#### 5. Frame

*Frame* adalah kontrol yang digunakan sebagai kontainer bagi kontrol lainnya.

#### 6. CommandButton

*CommandButton* merupakan kontrol hampir ditemukan pada setiap form dan digunakan untuk membangkitkan *event* proses tertentu ketika pemakai melakukan klik padanya.

#### 7. CheckBox

*CheckBox* digunakan untuk pilihan yang isinya bernilai *yes* atau *no*, *true* atau *false*.

#### 8. Option Button

*OptionButton* sering digunakan lebih dari satu sebagai pilihan terhadap beberapa *option* yang hanya dapat dipilih satu.

#### 9. List Box

*ListBox* mengandung sejumlah item dan pemakai dapat memilih lebih dari satu.

#### 10. Combo Box

*ComboBox* merupakan kombinasi dari *TextBox* dan suatu *ListBox* di mana pemasukan data dapat dilakukan dengan pengetikan maupun pemilihan.

#### 11. HScrollbar/VScrollbar

*HScrollbar/VScrollbar* digunakan untuk membentuk *scrollbar* berdiri sendiri.

#### 12. Timer

*Timer* digunakan untuk proses *background* yang diaktifkan berdasarkan *interval* waktu tertentu. Ini merupakan kontrol non visual.

#### 13. DriveListBox, DirListBox, dan FileListBox

*DriveListBox*, *DirListBox*, dan *FileListBox* sering digunakan untuk membentuk *dialog box* yang berkaitan dengan *file*.

#### 14. Shape dan Line

*Shape* dan *Line* digunakan untuk menampilkan bentuk seperti garis, persegi, bulatan, oval.

#### 15. Image

*Image* berfungsi menyerupai *image box*, tetapi tidak dapat digunakan sebagai *kontainer* bagi kontrol lainnya. Sesuatu yang perlu diketahui bahwa kontrol *image* menggunakan *resource* yang lebih kecil dibandingkan dengan *Picture Box*.

#### 16. Data dan Adodc

Data dan Adodc digunakan untuk menampilkan *database* pada suatu *form*.

#### 17. OLE

*OLE* dapat digunakan sebagai tempat bagi program eksternal seperti *Microsoft Excel*, *Microsoft Word*, dan lain-lain.

## 18. SSTab

SSTab adalah objek berbentuk lembaran-lembaran. Setiap lembaran dapat berisi kelompok informasi.

## 19. DataGrid

*DataGrid* digunakan untuk menampilkan data dalam bentuk *grid* seperti di *worksheet excel*.

## 20. DalaList dan DataCombo

*DalaList* dan *DataCombo* digunakan untuk menampilkan data dalam sebuah *drop-down list box*.

### g. Properties

*Properties* merupakan nilai yang dimiliki oleh sebuah objek *visual basic*, merupakan sebuah jendela yang digunakan untuk menampung nama properti dari kontrol yang dipilih.

### h. Jendela Code

Jendela *Code* adalah salah satu jendela yang paling penting dalam *visual basic*, yang berisi kode-kode program yang merupakan instruksi-instruksi untuk aplikasi *visual basic*. Setiap objek pada *visual basic* dapat ditambahai kode-kode program untuk melaksanakan tugas-tugas tertentu, misalnya membatalkan perintah, menutup aplikasi dan sebagainya.

### i. Form Layout Window

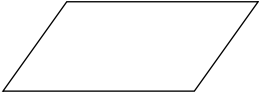

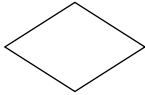



*Form Layout Window* merupakan sebuah jendela yang digunakan untuk mengatur posisi dari *form* pada *form* saat program dijalankan. Pada saat mengarahkan *pointer mouse* ke bagian *form*, maka *pointer mouse* akan berubah menjadi anak panah empat arah (*pointer* mengatur posisi) untuk memindah posisi *form* pada *layer* monitor dapat dilakukan dengan proses *drag and drop*.

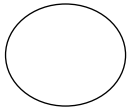
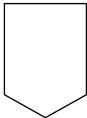



### 2.15. Flowchart

Menurut (Triwibisono, 2009) *flowchart* didefinisikan sebagai skema penggambaran dari algoritma atau proses. Tabel berikut menampilkan simbol-simbol yang digunakan dalam menyusun *flowchart*.

**Tabel 2.5.** Simbol-Simbol *Flowchart*

| Proses                                                                              | Keterangan                                                                                                                          |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
|    | <p><b><i>Input atau Output</i></b><br/>Digunakan untuk menuliskan proses menerima data atau mengeluarkan data.</p>                  |
|   | <p><b>Proses</b><br/>Digunakan untuk menyakan proses yang membutuhkan keputusan.</p>                                                |
|  | <p><b><i>Conditional atau Decision</i></b><br/>Digunakan untuk menyatakan proses yang membutuhkan keputusan.</p>                    |
|  | <p><b><i>Terminator</i></b><br/>Sebagai simbol “<i>START</i>” atau “<i>END</i>” untuk memulai atau mengakhiri <i>flowchart</i>.</p> |
|  | <p><b><i>Preparation</i></b><br/>Digunakan untuk memberikan nilai awal.</p>                                                         |
|  | <p><b><i>Display</i></b><br/>Digunakan untuk menampilkan data ke monitor.</p>                                                       |

|                                                                                     |                                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p><b><i>Connector (Off-page)</i></b><br/>Digunakan untuk menyatukan beberapa <i>arrow</i>.</p>                                                                                                                        |
|    | <p><b><i>Connector (Off-page)</i></b><br/>Digunakan untuk menghubungkan <i>flowchart</i> yang harus digambarkan pada halaman yang berbeda, biasanya pada simbol ini diberi nomor sebagai penanda misalnya angka 1.</p> |
|  | <p><b><i>Arrow</i></b><br/>Sebagai penunjuk arah dan alur proses.</p>                                                                                                                                                  |