BAB II
TINJAUAN PUSTAKA

Bab ini akan menguraikan tentang dasar-dasar teori yang menunjang dalam perencanaan dan pembuatan alat penyortiran barang berdasarkan tujuan. Uraian teori dalam bab ini meliputi IC Mikrokontroler ATMega16 dan perangkat pendukung lainnya.

2.1 Perangkat Keras *(Hardware)*

Adapun *hardware* yang digunakan dalam perancangan peralatan alat pemisah barang berdasarkan warna sesuai dengan tujuan penerbangan ini adalah, RFID, *Conveyor belt*, *LCD*, Mikrokontroler ATMega 16, Motor *DC*, dan *Power Supply*. Pengenalan *hardware* ini digunakan sebagai referensi acuan dalam mendukung teori pemrograman.

2.1.1 Pengenalan Mikrokontroler

Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umunya dapat menyimpan program dimana pada umumnya terdiri dari *CPU (Central Processing Unit)*, memori, *I/O* tertentu dan unit pendukung seperti *Analog-to-Digital Converter (ADC)* yang sudah terintegrasi di dalamnya(Widodo,2005). Kelebihan utama dari mikrokontroler adalah tersedianya RAM dan peralatan *I/O* pendukung sehingga ukuran board mikrokontroler menjadi sangat ringkas.

Mikrokontroler dapat dikelompokkan dalam satu keluarga, masing-masing mikrokontroler memiliki spesifikasi tersendiri namun tetap sama dalam pemrogramannya. Berikut beberapa contoh mikrokontroler yang banyak di pasaran saat ini:

1. Keluaran MCS-51
2. Keluaran MC68HC05
3. Keluaran MC68HCH
4. Keluaran AVR
5. Keluaran PIC 8

2.1.2 Mikrokontroler ATMega 16

Pada laporan akhir ini, penulis menggunakan mikrokontroller ATMega16 karena kemampuan kecepatan eksekusi yang lebih tinggi menjadi alasan kuat bagi banyak orang untuk memilih ATMega16, yang juga mikrokontroler keluarga AVR, dibandingkan dengan mikrokontroler pendahulunya yaitu keluarga MCS-51. Seiring perkembangan teknologi mikrokontroler, mikrokontroler dibuat semakin baik dengan bahasa pemrograman yang juga ikut berubah. Salah satunya adalah mikrokontroler *AVR (Alf and Vegards Risc processor)* Mikrokontroler *AVR (Alf and Vegard's Risc processor)* dari Atmel ini menggunakan arsitektur *RISC (Reduced Instruction Set Computer)* yang artinya prosesor tersebut memiliki set instruksi program yang lebih sedikit dibandingkan dengan *MCS*-51 yang menerapkan arsitektur *CISC (Complex Instruction Set Computer).* Hampir semua instruksi prosesor *RISC* adalah instruksi dasar (belum tentu sederhana), sehingga instruksi-instruksi ini umumnya hanya memerlukan 1 siklus mesin untuk menjalankannya.

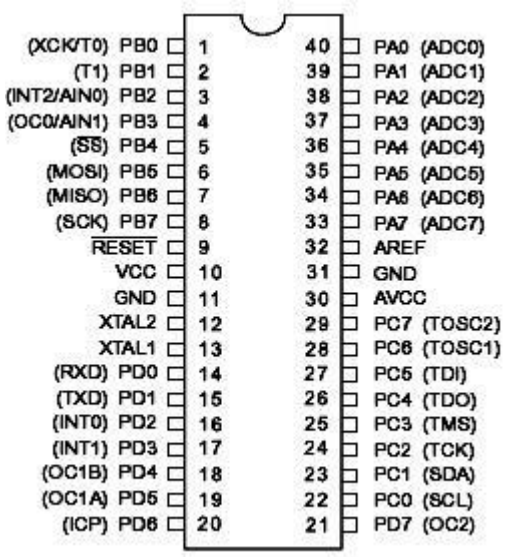
Sekarang ini, *AVR* dapat dikelompokkan menjadi 6 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, keluarga AT90CAN, keluarga AT90PWM dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya, sedangkan dari segi arsitektur dan instruksi yang digunakan, mereka hampir sama. Sebagai pengendali utama dalam pembuatan alat ini, digunakan salah satu produk ATMEL dari keluarga ATmega yaitu Atmega16.

Mikrokontroler *AVR* ATMega16 memiliki fitur yang cukup lengkap. Mikrokontroler *AVR* ATMega16 telah dilengkapi dengan *ADC internal*, EEPROM *internal*, *Timer/Counter*, *PWM*, *analog comparator*, dll (M.Ary Heryanto,2008). Sehingga dengan fasilitas yang lengkap ini memungkinkan kita belajar mikrokontroler keluarga AVR dengan lebih mudah dan efisien, serta dapat mengembangkan kreativitas penggunaan mikrokontroler ATMega16.

Untuk memenuhi kebutuhan dan aplikasi industri yang sangat beragam, mikrokontroler keluarga AVR ini muncul di pasaran dengan tiga seri utama yaitu, TinyAVR, ClassicAVR, MegaAVR. Keseluruhan seri AVR ini pada dasarnya memiliki organisasi memori dan set instruksi yang sama (sehingga dengan

demikian jika kita telah mahir menggunakan salah satu seri AVR, untuk beralih ke seri yang lain akan relatif lebih mudah). Perbedaan antara *tinyAVR*, *ClassicAVR* dan *MegaAVR* pada kenyataannya hanya mereflesikan tambahan-tambahan fitur yang ditawarkannya saja (misal adanya tambahan ADC internal pada seri AVR tertentu, jumlah port *I/O* serta memori yang berbeda, dan sebagainya). Diantara ketiganya *MegaAVR* yang umumnya memiliki fitur yang paling lengkap, disusul oleh *ClassicAVR*, dan terakhir *TinyAVR*.

2.1.3 Pin-Pin pada Mikrokontroler ATMega16



Gambar 2.1 Konfigurasi pin ATMega16

Konfigurasi pin ATMEGA16 dengan kemasan 40 pin *Dual In-line Package* (DIP) dapat dilihat pada Gambar 2.1 dari gambar diatas dapat dijelaskan fungsi dari masing-masing pin ATMega16 sebagai berikut.

1. VCC merupakan pin yang brfungsi sebagai masukan catu daya
2. GND merupakan pin Ground
3. Port A (PA0 – PA7) merupakan pin input/output dua arah *(full duplex)* dan selain itu merupakan pin masukan ADC
4. Port B (PB0 – PB7) merupakan pin input/output dua arah *(full duplex)* dan selain itu merupakan pin khusus, seperti dapat dilihat pada tabel 2.1 dibawah ini.

Tabel 2.1 Fungsi Khusus Port B

Pin	Fungsi Khusus
PB0	XCK *(USART External Clock Input/Output)* T0 *(Timer/Counter0 External Counter Input)*
PB1	T1 *(Timer/Counter1 External Counter Input)*
PB2	INT2 *(External Interupt 2 Input)* AIN0 *(Analaog Comparator Negative Input)*
PB3	OC0 *(Timer/Counter0 Output Compare Macth Output)* AIN1 *(Analaog Comparator Negative Input)*
PB4	SPI *(Slave Select Input)*
PB5	MOSI *(SPI Bus Master Output /Slave Input)*
PB6	MISO *(SPI Bus Master Input/Slave Output)*
PB7	SCK *(SPI Bus Serial Clock)*

5. Port C (PC0 – PC7) merupakan pin input/output dua arah *(full duplex)* dan selain itu merupakan pin khusus, seperti dapat dilihat pada tabel 2.2 dibawah ini.

   Tabel 2.2 Fungsi Khusus Port C

Pin	Fungsi Khusus
PC0	SCL *(Two-wire Serial Bus Clock Line)*
PC1	SDA *(Two-wire Serial BusData Input/Output Line)*
PC2	TCK *(Joint Test Action Group Test Clock)*
PC3	TMS *(JTAG Test Mode Select)*
PC4	TDO *(JTAG Data Out)*
PC5	TDI *(JTAG Test Data In)*
PC6	TOSC1 *(Timer Oscillator pin 1)*
PC7	TOSC2 *(Timer Oscillator pin 2)*

6. Port D (PD0 – PD7) merupakan pin input/output dua arah *(full duplex)* dan

selain itu merupakan pin khusus, seperti dapat dilihat pada tabel 2.3 dibawah ini.

Tabel 2.3 Fungsi Khusus Port D

Pin	Fungsi Khusus
PD0	RXD *(USART Input Pin)*
PD1	TXD *(USART Output Pin)*
PD2	INT0 *(External Interupt 0 Input)*
PD3	INT1 *(External Interupt 1 Input)*
PD4	OC1B *(Timer/Counter1 Output Compare B Macth Output)*
PD5	OC1A *(Timer/Counter1 Output Compare A Macth Output)*
PD6	ICP *(Timer/Counter1 Input Capture Pin)*
PD7	OC2 *(Timer/Counter2 Output Compare Macth Output)*

7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler12
8. XTAL1 dan XTAL2, merupakan pin masukan external clock
9. AVCC merupakan pin masukan tegangan untuk ADC
10. AREF merupakan pin masukan tegangan referensi untuk ADC.

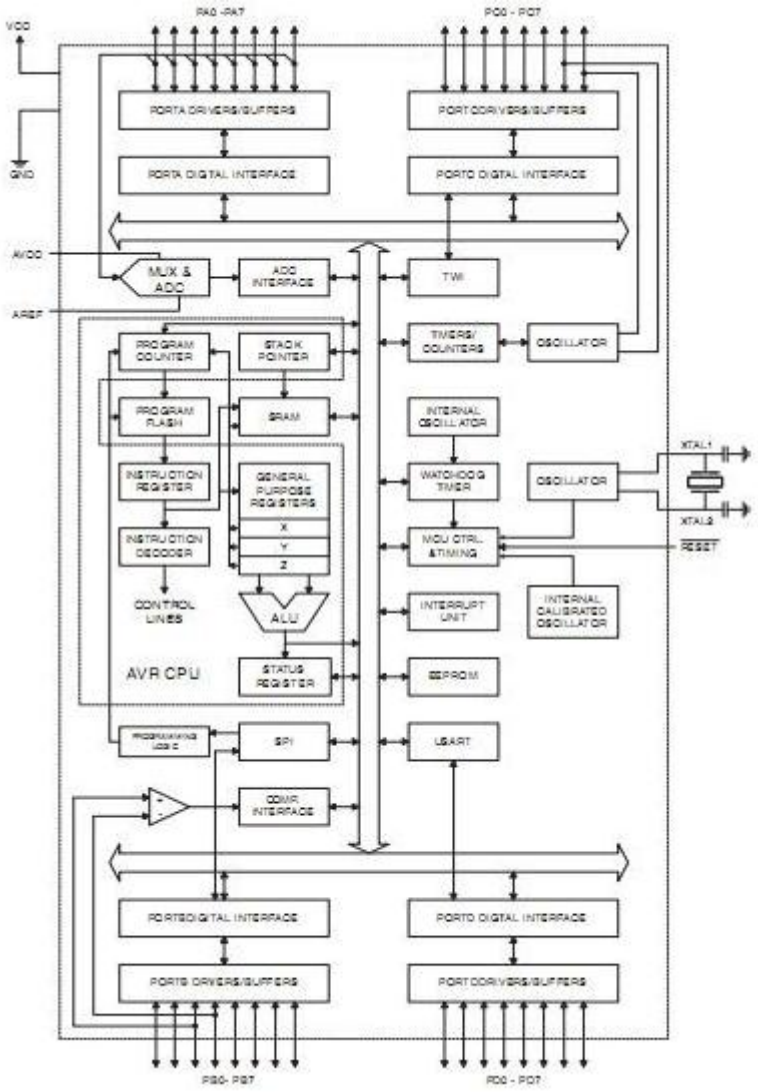
2.1.4 Diagram Blok

ATMEGA16 mempunyai *throughput* mendekati 1 Millions Instruction Per Second (MIPS) per MHz, sehingga membuat konsumsi daya menjadi rendah terhadap kecepatan proses eksekusi perintah.

Beberapa keistimewaan dari AVR ATMEGA16 antara lain:

1. Mikrokontroler AVR 8 bit yang memilliki kemampuan tinggi dengan konsumsi daya rendah
2. Arsitektur RISC dengan throughput mencapai 16 MIPS pada frekuensi 16MHz
3. Memiliki kapasitas Flash memori 16 Kbyte, EEPROM 512 Byte dan SRAM 1 Kbyte
4. Saluran I/O sebanyak 32 buah, yaitu Port A, Port B, Port C dan Port D
5. CPU yang terdiri dari 32 buah register.

Pada gambar 2.2 dibawah ini menununjukan diagram blok pada mikrokontroler ATMega 16:



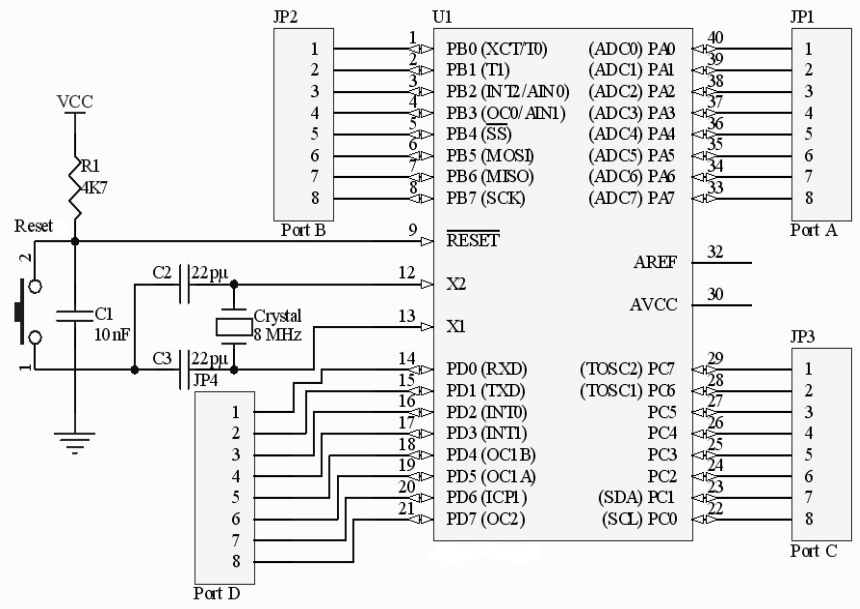
Gambar 2.2 Diagram Blok ATMega16

6. Unit interupsi dan eksternal
7. Port USART untuk komunikasi serial
8. Fitur peripheral
   a. Tiga buah *Timer/Counter* dengan kemampuan perbandingan *(compare)*
   b. Dua buah *Timer/Counter* 8 bit dengan *Prescaler* terpisah dan Mode *Compare*
   c. Satu buah *Timer/Counter* 16 bit dengan *Prescaler* terpisah, Mode *Compare* dan Mode *Capture*
   d. *Real Time Counter* dengan Oscillator sendiri
   e. Empat kanal PWM

f. 8 kanal ADC
g. 8 *Single-ended Channel* dengan keluaran hasil konversi 8 dan 10 resolusi (register ADCH dan ADCL)
h. 7 *Diferrential Channel* hanya pada kemasan *Thin Quad Flat Pack (TQFP)*
i. 2 *Differential Channel* dengan *Programmable Gain*
j. Antarmuka *Serial Peripheral Interface (SPI)* Bus
k. *Watchdog Timer* dengan *Oscillator Internal*
l. On-chip Analog Comparator

9. *Non-volatile* program memori

2.1.5 Rangkaian Sistem Minimum Mikrokontroler ATMega16

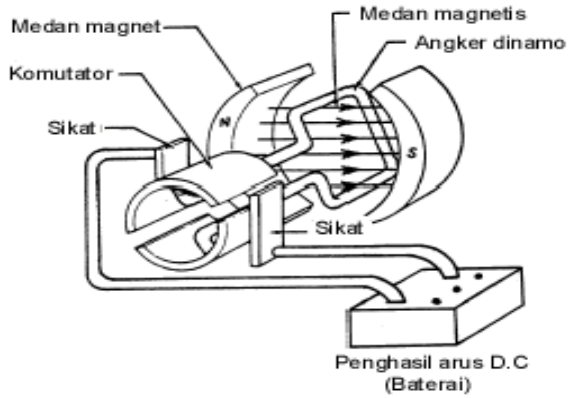


Gambar 2.3 Sistem Minimum Mikrokontroler ATMega16

Gambar 2.3 diatas adalah gambar sistem minimum mikrokontroler ATMega16. Sistem minimum mikrokontroler adalah sebuah rangkaian elektronik yang di perlukan untuk beroprasinya IC mikrokontroler. Sistem minimum ini bisa di hubungkan dengan rangkaian lain untuk menjalankan fungsi tertentu, contohnya dapat di hubungkan dengan berbagai sensor untuk menerima data dan di olah oleh IC mikrokontroler. Pada perancangan ini mikrokontroler yang digunakan adalah ATMega16 yang merupakan keluarga mikrokontroler dari ATMEL.

2.2 Motor DC

Pada laporan akhir ini, penulis menggunakan Motor DC *(Direct Current)* sebagai penggerak sabuk konveyor. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Energi mekanik ini pada umumnya digunakan untuk memutar impeller pompa, kipas atau blower, menggerakan kompresor, mengangkat bahan,dll. Motor listrik digunakan juga di rumah (mixer, bor listrik, kipas angin) dan di industri. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. Komponen-komponen pada motor dc dapat dilihat pada gambar 2.4 di bawah ini:



Gambar 2.4 Komponen-Komponen Pada Motor DC

Catu tegangan dc dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan pada gambar di atas disebut angker dinamo. Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet.

2.2.1 Prinsip Kerja Motor DC

Motor DC adalah piranti elektronik yang mengubah energi listrik menjadi energi mekanik berupa gerak rotasi. Pada motor DC terdapat jangkar dengan satu atau lebih kumparan terpisah. Tiap kumparan berujung pada cincin belah

(komutator). Dengan adanya *insulator* antara *komutator,* cincin belah dapat berperan sebagai saklar kutub ganda (*double pole, double throw switch*). Motor DC bekerja berdasarkan prinsip gaya Lorentz, yang menyatakan ketika sebuah konduktor beraliran arus diletakkan dalam medan magnet, maka sebuah gaya (yang dikenal dengan gaya Lorentz) akan tercipta secara ortogonal diantara arah medan magnet dan arah aliran arus. Pada motor DC, kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konverter energi baik energi listrik menjadi energi mekanik (motor) maupun sebaliknya dari energi mekanik menjadi energi listrik (generator) berlangsung melalui medium medan magnet. Energi yang akan diubah dari suatu sistem ke sistem yang lain, sementara akan tersimpan pada medium medan magnet untuk kemudian dilepaskan menjadi energi lainnya. Dengan demikian, medan magnet disini selain berfungsi sebagi tempat penyimpanan energi juga sekaligus proses perubahan energi, dimana proses perubahan energi pada motor arus searah dapat digambarkan pada gambar 2.5(Zuhal, 1991).

Gambar 2.5. Proses Konversi Energi pada Motor DC

Dengan mengingat hukum kekekalan energi, proses konversi energi listrik menjadi energi mekanik dapat dinyatakan sebagai berikut: Energi listrik sebagai input = Energi mekanik sebagai output + energi yang diubah menjadi panas + Energi yang tersimpan dalam medan magnet. kerja motor DC terjadi jika suatu lilitan jangkar dialiri arus listrik searah dengan arah i didalam medan magnet B, maka akan terbangkit gaya [F] sebesar :

$$\mathbf{F = B.i.l} \qquad (2.1)$$

Arah gaya ini ditentukan oleh aturan tangan kiri, dengan ibu jari, jari telunjuk, dan jari tengah saling tegak lurus menunjukan masing – masing arah , dan . Persamaan di atas merupakan prinsip dari sebuah motor arus searah, dimana

terjadi proses perubahan energi listrik menjadi energi mekanik . Bila jari-jari rotor adalah r, maka torsi yang akan dibangkitkan adalah :

T = F.r = B.i.l.r (2.2)

dimana :

F = gaya lorentz (N)

T = Torsi (N/m)

B = kuat medan magnet (Tesla)

i = kuat arus listrik (A)

l = panjang penghantar (m)

r = jari – jari rotor (m)

Hubungan kecepatan, torsi, dan daya motor

$P = \frac{M}{t}$ dimana, $M = F.l$

$P = \frac{F.l}{t}$ sedangkan $v = \frac{l}{t}$

Dalam satu putaran poros jarak ditempuh :

$l = 2\,.r.\pi$ sehingga $v = n\,.2\,.r\,.\pi$

Jadi, $\boldsymbol{P = n\,.2\,.r\,.\pi\,.F}$ atau $\boldsymbol{P = v\,.F}$ (2.3)

Dimana :

P = Daya (watt)

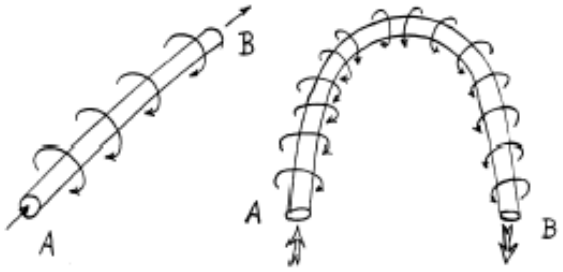
T = Momea/ Torsi (N/m)

t = waktu (s)

v = kecepatan putaran motor (rpm)

n = jumlah putaran

Aturan Genggaman Tangan Kanan bisa dipakai untuk menentukan arah garis fluks di sekitar konduktor. Genggam *konduktor* dengan tangan kanan dengan jempol mengarah pada arah aliran arus, maka jari-jari anda akan menunjukkan arah garis fluks. Pada gambar 2.6 dibawah ini menunjukkan medan magnet yang terbentuk di sekitar konduktor berubah arah karena bentuk U.



Gambar 2.6 Medan magnet yang membawa arus mengelilingi *konduktor.*

Medan magnet hanya terjadi di sekitar sebuah *konduktor* jika ada arus mengalir pada *konduktor* tersebut. Berikut dijelaskan mekanisme kerja motor DC secara umum yang bisa dilihat di bawah ini.

Mekanisme kerja untuk seluruh jenis motor secara umum :

1. Arus listrik dalam medan magnet akan memberikan gaya.
2. Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran / *loop,* maka kedua sisi *loop,* yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
3. Pasangan gaya menghasilkan tenaga putar / *torque* untuk memutar kumparan.
4. Motor-motor memiliki beberapa *loop* pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan.

2.3 Relay SPDT *(Single Pole Double Throw)*

Relay adalah suatu piranti yang bekerja berdasarkan elektromagnetik untuk menggerakan sejumlah kontaktor (saklar) yang tersusun. Kontaktor akan tertutup *(On)* atau terbuka *(Off)* karena efek induksi magnet yang dihasilkan kumparan (induktor) ketika dialiri arus listrik. Berbeda dengan saklar dimana pergerakan

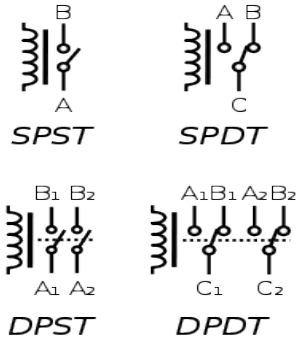
kontaktor *(On/Off)* dilakukan manual tanpa perlu arus listrik. Relay terdiri dari 3 bagian utama, yaitu:

1. *koil* : lilitan dari relay
2. *common* : bagian yang tersambung dengan NC(dlm keadaan normal)
3. kontak : terdiri dari NC dan NO

Keterangan:

1. *Normally Closed* (NC) merupakan bagian sakelar relay yang dalam keadaan normal (relay tidak diberi tegangan) terhubung dengan common.
2. *Normally Open* (NO) *Normally Open* (NO) merupakan bagian saklar relay yang dalam keadaan normal (relay tidak diberi tegangan) tidak terhubung dengan common. Tetapi *Normally Open* akan terhubung dengan *common* apabila relay diberi tegangan.

Sebagai komponen elektronika, relay mempunyai peran penting dalam sebuah sistem rangkaian elektronika dan rangkaian listrik untuk menggerakan sebuah perangkat yang memerlukan arus besar tanpa terhubung langsung dengan perangakat pengendali yang mempunyai arus kecil. Dengan demikian relay dapat berfungsi sebagai pengaman. Simbol relay dapat dilihat pada gambar 2.7 di bawah ini:



Gambar 2.7 Simbol Relay

Jenis-Jenis Relay :

1. SPST (*Single Pole Single Throw) :* Relay ini memiliki empat terminal. Dua terminal kumparan (*coil*) dan dua terminal saklar (A dan B) yang dapat terhubung dan terputus.

2. SPDT *(Single Pole Double Pole)* : Relay ini memiliki lima terminal. Dua terminal kumparan *(coil)* dan tiga terminal saklar (A,B, dan C) yang dapat terhubung dan terputus dengan satu terminal pusat. Jika suatu saat terminal A terputus dengan terminal pusat (C) maka terminal lain (B) terhubung dengan terminal C, demikian juga sebaliknya.
3. DPST *(Double Pole Single Throw)* : Relay ini mempunyai enam terminal. Dua terminal kumparan *(coil),* dan empat terminal merupakan dua pasang saklar yang dapat terhubung dan terputus (A1 dan B1 - A2 dan B2).
4. DPDT *(Double pole Double Throw)* : Relay ini mempunyai delapan terminal. Dua terminal kumparan *(coil),* enam terminal merupakan dua set saklar yang dapat terputus dan terhubung (A1,B1,C1 dan A2, B2, C2)

2.4 *Belt Conveyor*

Belt Conveyor pada dasarnya mernpakan peralatan yang cukup sederhana. Alat tersebut terdiri dari sabuk yang tahan terhadap pengangkutan benda padat. Sabuk yang digunakan pada *belt conveyor* ini dapat dibuat dari berbagai jenis bahan misalnya dari karet, plastik, kulit ataupun logam yang tergantung dari jenis dan sifat bahan yang akan diangkut. Untuk mengangkut bahan -bahan yang panas, sabuk yang digunakan terbuat dari logam yang tahan terhadap panas.

Belt conveyor adalah mesin pemindah yang paling universal karena kapasitas cukup besar (500 s.d 5000 m^3/jam atau lebih), sanggup memindahkan material pada jarak relatif besar (500 s/d 1000 m atau lebih), desain yang sangat sederhana dan pengoperasian yang baik (http://www.hksystems.com,"conveyor"). Belt conveyor dapat digunakan untuk memindahkan berbagai unit material sepanjang arah horizontal atau pada suatu kemiringan tertentu pada berbagai industri. Contohnya pada industri pengecoran logam, tambang batubara, produksi beton, industri makanan dan lain-lain.

Belt Conveyor (Konveyor Sabuk) memiliki komponen utama berupa sabuk yang berada diatas roller-roller penumpu. Sabuk digerakkan oleh motor penggerak melalui suatu pulley, sabuk bergerak secara translasi dengan melintas datar atau miring tergantung kepada kebutuhan dan perencanaan. Material diletakkan diatas sabuk dan bersama sabuk bergerak kesatu arah. Pada pengoperasiannya konveyor sabuk menggunakan tenaga penggerak berupa motor listrik dengan perantara roda

gigi yang dikopel langsung ke puli penggerak. Sabuk yang berada diatas roller-roller akan bergerak melintasi roller-roller dengan kecepatan sesuai putaran dan puli penggerak. (Heryanto dan Adi,2008)

Beberapa informasi penting tentang material angkut yang perlu diketahui dalam perhitungan desain *conveyor,* antara lain :

1. Karakteristik pemakaian, hal ini menyangkut jenis dan ukuran material, sifat material, serta kondisi medan atau ruang kerja alat.
2. Proses produksi, mengangkut kapasitas perjam dari unit, kontinuitas pemindahan, metode penumpukan material dan lamanya alat beroperasi.
3. Prinsip-prinsip ekonomi, meliputi ongkos pembuatan, pemeliharaan, pemasangan, biaya operasi dan juga biaya penyusutan dari harga awal alat tersebut

Kelebihan *belt conveyor* :

1) Mampu membawa beban berkapasitas besar.
2) Kecepatan sabuk dapat diatur untuk menetapkan jumlah material yang dipindahkan persatuan waktu
3) Dapat bekerja dalam arah yang miring tanpa membahayakan operator yang mengoperasikannya
4) Memerlukan daya yang lebih kecil, sehingga menekan biaya operasinya
5) Tidak mengganggu lingkungan karena tingkat kebisingan dan polusi yang rendah.
6) Lebih ringan dari pada konveyor rantai maupun bucket conveyor.
7) Aliran pengangkutan berlansung secara terus menerus/kontinu

Kelemahan *belt conveyor* :

1) Sabuk sangat peka terhadap pengaruh luar, misalnya timbul kerusakan pada pinggir dan permukaan belt, sabuk bisa robek karena batuan yang keras dan tajam atau lepasnya sambungan sabuk.
2) Biaya perawatannya sangat mahal.
3) Jalur pemindahan *(transfer line).* Karena untuk satu unit *belt conveyor* hanya bisa dipasang untuk jalur lurus.
4) Kemiringan/sudut inklinasi yang terbatas.

Sabuk konveyor yang sederhana terdiri dari :

1. Rangka *(Frame)*
2. Puli penggerak *(Drive pulley)*
3. Puli yang digerakkan *(Tail pulley)*
4. Puli Pengencang (*Snub pulley)*
5. Sabuk *(Belt)*
6. Rol pembawa (*Carrying roller idler)*
7. Rol Kembali *(Return roller idler)*
8. Motor penggerak
9. Unit pemuat (*Chutes)*
10. Unit pengeluar (*Discharge spout)*
11. Pembersih sabuk (*Belt cleaner)*
12. Pengetat sabuk (*Belt take-up)*

2.5 *Radio Frequency Identification* (RFID)

Radio Frequency identification (RFID) adalah proses identifikasi suatu objek dengan menggunakan frekuensi transmisi radio. Frekuensi radio digunakan untuk membaca informasi dari sebuah device kecil yang disebut RFID *tag* atau *transponder (transmitter responder)* dan untuk selanjutnya disingkat menjadi ID tag. ID tag akan mengenali diri sendiri ketika mendeteksi sinyal dari device yang *compatible*, yaitu RFID *reader*.(Astono,2009). Perbandingan RFID dan *Barcode* dapat dilihat di tabel 2.4 di bawah ini.

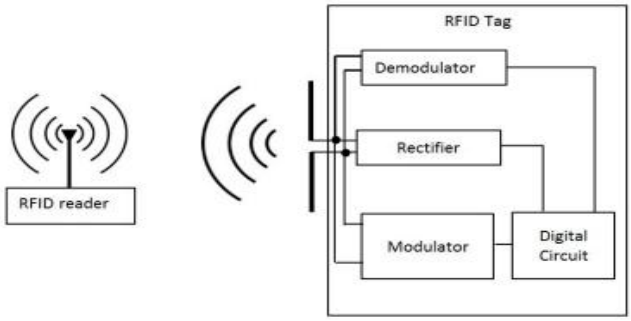
Tabel 2.4 Tabel Perbandingan RFID dan *Barcode*

Keadaan	*Barcode*	RFID
Kondisi baca	*Line of sight(LOS)*	*Non-los*
Posisi baca	Vertikal atau horizontal dengan toleransi tertentu	Bebas, segala kondisi memenuhi
Kecepatan baca	Relatif(2-5 detik)	<100 milidetik per item
Jarak baca maksimum	±7cm	±30cm(pendek), ±3m (menengah), ± 10M(jauh)
Kapasitas memori	Kecil	Hingga 64KB atau lebih
Proses pembacaan	Pemrosesan satu per satu	Banyak barang per proses
Kondisi buruk (debu,air)	Merusak label barcode, pembaca error	Tidak berpengaruh
Kemudahan duplikasi	Mudah	Hampir mustahil
Kemampuan	*Read only*	*Read and Write*

Pada Tabel 2.4 RFID lebih *reliable* dalam proses pembacaan karena dalam proses pembacaan bebas dengan segala kondisi. Dalam hal ini barang tidak perlu membuat design box untuk tempat RFID agar dapat terbaca. RFID lebih tahan terhadap kondisi seperti kotoran kimiawi debu dan lainnya dalam pembacaannya. Ukuran sangat kecil (untuk jenis pasif RFID) sehingga mudah ditanamkan dimana-mana.

2.5.1 TAG & Reader RFID

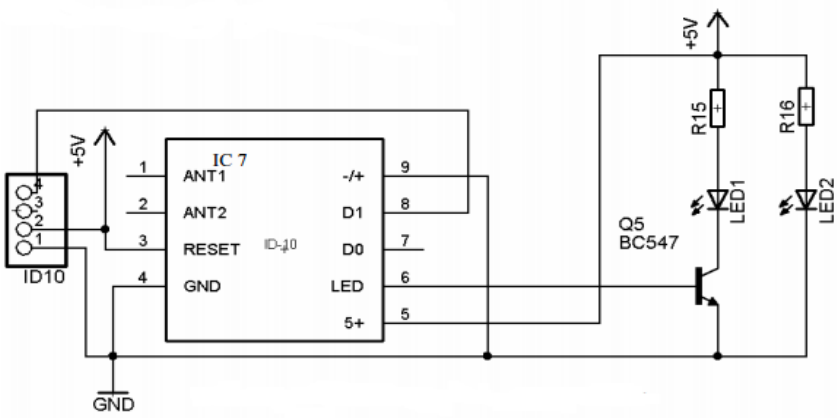
RFID *Reader* adalah device yang dibuat dari rangkaian elektronika tersusun menjadi sel-sel. Beberapa sel menyimpan data *read only*, misalnya serial number yang unik yang disimpan pada saat tag tersebut diproduksi. Sel lain pada RFID mungkin juga dapat ditulis dan dibaca secara berulang. Kontak antara ID tag dengan readernya dilakukan dengan pengiriman gelombang elektromagnetik. Sistem RFID dapat dilihat pada gambar 2.8 di bawah ini:



Gambar 2.8 Sistem RFID

Keunggulan utama RFID adalah pada aspek efisiensi dan kenyamanan dapat diuraikan sebagai berikut:

1. ID tag mampu diidentifikasi secara bersamaan, tanpa harus berada dalam jarak dekat.
2. ID tag mampu diidentifikasi menembus berbagai objek seperti kertas, plastic dan kayu (wireless data capture) dalam jarak ± 30 Cm. Rangkaian RFID reader dapat dilihat pada gambar 2.9 dibawah ini:

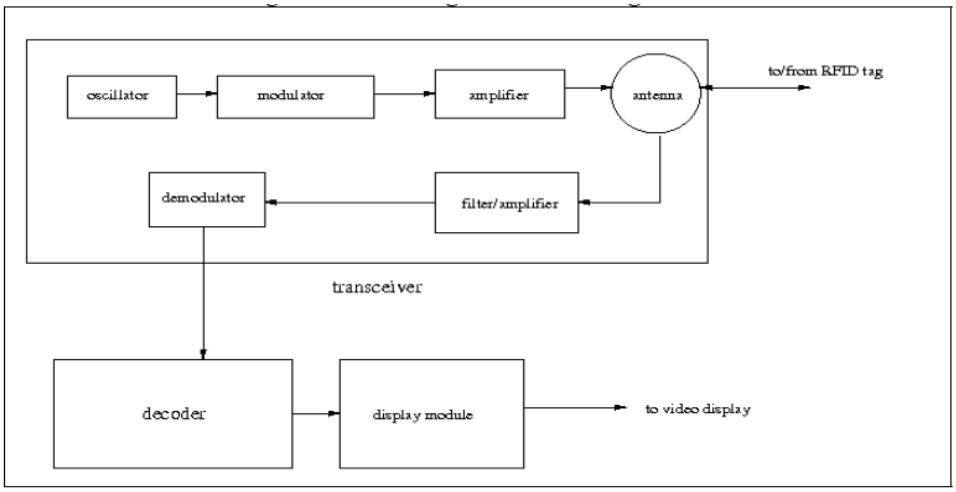


Gambar 2.9 Rangkaian *RFID Reader*

Berdasarkan catu daya tag, RFID dapat digolongkan menjadi

Tag aktif: yaitu tag yang catu dayanya diperoleh dari baterai, sehingga akan mengurangi daya yang diperlukan oleh pembaca RFID dan tag dapat mengirimkan informasi dalam jarak yang lebih jauh.(Winda, 2009). Kelemahan dari tipe tag ini adalah harga yang mahal dan ukuranny yang lebih besar karena lebih komplek. Semakin banyak fungsi yang dapat dilakukan oleh ID tag maka rangkaian akan semakin komplek dan ukuran akan semakin besar.

Tag pasif: yaitu tag yang catu daya diperoleh dari medan yang dihasilkan oleh pembaca RFID rangkaianya lebih sederhana, harganya jauh lebih murah, ukurannya lebih kecil, dan lebih ringan kelemahannya adalah tag hanya dapat mengirimkan informasi dalam jarak yang dekat dan pembaca RFID harus menyediakan daya tambahan untuk ID tag.(Winda 2009) Pada sistem RFID biasanya tag atau atau transponder ditempelkan pada suatu object. Setiap tag dapat membawa informasi yang unik, diantaranya, serial number, model, warna, tempat perakitan, dan data lain. Pada sistem RFID tag atau transponder ditempelkan pada suatu object. Blok diagram RFID dapat dilihat pada gambar 2.10 dibawah ini:



Gambar 2.10 Blok Diagram RFID Tag dan Reader

2.5.2 Jenis-jenis Frekuensi Yang Digunakan RFID

Faktor penting yang harus diperhatikan dalam RFID adalah frekuensi kerja dari sistem RFID adalah frekuensi yang digunakan untuk komunikasi wireless antara reaer RFID dengan ID tag. Ada beberapa band frekuensi yang digunakan untuk sistem RFID pemilihan dari frekuensi kerja sistem sistem RFID akan mempengaruhi jarak komunikasi, interferensi dengan frekuensi sistem radio lain, kecepatan komunikasi data dan ukuran antenna untuk frekuensi yang rendah umumnya digunakan tag pasif frekuensi kerjanya bekisar antara 13,56 MHz, dan untuk frekuensi tinggi digunakan tag aktif frekuensi kerja dari tag pasif bekisar antara 125KHz-145KHz.

Pada frekuensi rendah, tag pasif tidak dapat mentransmisikan data dengan jarak yang jauh dikarenakan keterbatasan daya yang diperoleh dari medan elektromagnetik, akan tetapi komunikasi tetap dapat dilakukan tanpa kontak langsung hal yang perlu mendapatkan perhatian adalah tag pasif tidak dihalangi oleh objek logam.

Pada frekuensi tinggi, jarak komunikasi antara tag aktif dengan RFID reader dapat lebih jauh, tetapi masih terbatas oleh daya yang ada sinyal elektromagnetik pada frekuensi tinggi juga mendapatkan pelemahan (atenuasi) ketika tag tertutupi oleh es atau air. Pada kondisi terburuk, tag yang tertutup oleh logam tidak terdeteksi oleh pembaca RFID seperti yang dikutip dari penelitian denagn judul

analisa dan perancangan tracking sistem dengan RFID pada PT. Ekasari lorena ekspress.(Erwin, 2010).

2.5.3 Regulasi Dan Standarisasi RFID

Sampai saat ini belum ada lembaga atau badan dunia yang mengatur mengenai penggunaan frekuensi pada RFID, setiap negara dapat membuat peraturan sendiri mengenai hal ini. Badan-badan utama yang tugasnya member alokasi frekuensi untuk RFID adalah sebagai berikut

1. Amerika Serikat : *Federal Communication Commission* (FCC)
2. Kanada : *Department Of Communication* (DOC)
3. Eropa : ERO, CEPT dan ETSI
4. Jepang : *Ministry of Public Management, Home affairs, Post and Telecommunication* (MPHPT)
5. China : *Ministry of information industry*
6. Australia : *Australian communication authority*
7. New zealand : *Ministry of economic Development*

Frekuensi rendah (125 – 134 KHz dan 140 – 148.5 KHz) dan frekuensi tinggi (13.56 MHz) dari RFID tag dapat digunakan secara global tanpa lisensi. Frekuensi ultra tinggi (UHF 868 MHz – 928 MHz), tidak boleh digunakan secara global karena belum ada standar global yang mengaturnya. Regulasi juga ada pada sisi kesehatan dan isu lingkungan. Sebagai contoh, di eropa, regulasi dari waste electrical dan electronic equipment menyatakan bahwa ID tag tidak boleh dibuang. Hal tersebut berarti bahwa jika suatu kemasan kosong mau dibuang, maka ID tag-nya harus dilepas terlebih dahulu.

2.6 *Liquid Crystal Display* (LCD)

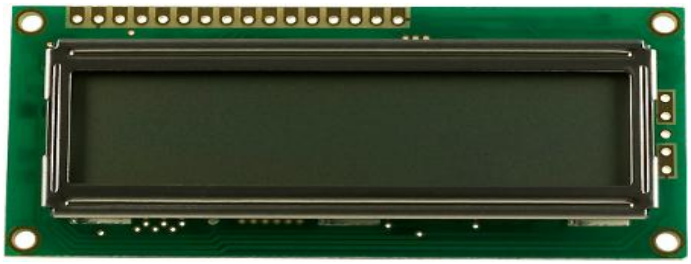
Liquid Crystal Display (LCD) adalah suatu jenis media tampilan yang menggunakan Kristal cair sebagai penampil utama. LCD terdiri dari lapisan-lapisan cairan Kristal diantara dua pelat kaca. Film transparan yang dapat menghantarkan listrik atau back plane, diletakan pada lembaran belakang kaca. Bagian transparan dari film yang dapat menghantarkan arus listrik pada bagian luar dari karakter yang diinginkan dilapiskan pada pelat bagian depan. Pada saat

terdapat tegangan antara segmen dan back plane, bagian yang berarus listrik ini mengubah transmisi cahaya melalui daerah dibawah segmen film.

LCD yang digunakan merupakan LCD tipe karakter karena LCD ini dapat menampilkan data berupa karakter. Keuntungan yang dapat diperoleh dengan menggunakan LCD adalah:

1. Dapat menampilkan karakter ASCII, sehingga memudahkan untuk membuat program tampilannya.
2. Mudah dihubungkan dengan port I/O karena hanya menggunakan 8 bit data dan 3 bit control.

LCD yang digunakan dalam tugas akhir ini merupakan tipe karakter 16x2 baris dan dapat menampilkan 16 karakter perbaris dan mempunyai 2 baris. Bentuk Fisik LCD dapat dilihat pada gambar 2.11 dibawah ini sedangkan konfigurasi untyk pin-pin LCD dapat dilihat di tabel 2.5 dibawah ini.



Gambar 2.11 LCD 16x2

Tabel 2.5 Konfigurasi Pin LCD 16x2

Nomor Kaki	**Simbol**	**Fungsi**
1	Gnd	Ground
2	Vcc	5 V
3	Vee	Pengatur Kontras
4	RS	Bernilai “1” untuk input berupa Data dan Bernilai “0” untuk input berupa instruksi
5	RW	Bernilai “1” untuk input proses Baca dan Bernilai “0” untuk input proses Tulis
6	E	Sinyal Enable bernilai “1”
7	DB0	Jalur Data
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	

15	V+BL	4,2 Volt
16	V-BL	Ground

2.7 Komponen Pendukung

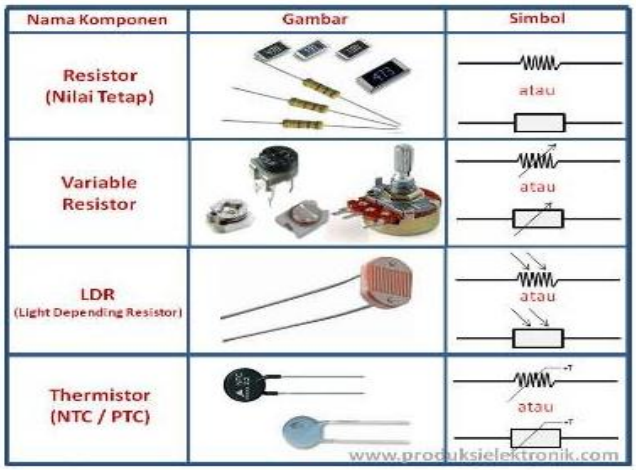
2.7.1 Resistor

Resistor adalah komponen dasar elektronika yang fungsinya untuk menghambat atau membatasi arus listrik yang mengalir ke komponen lainnya dalam suatu rangkaian elektronika (Taranggono, 1996). Kemampuan resistor menghambat arus bermacam-macam tergantung dari nilai resistansinya. Makin besar resistansi resistor maka arus akan semakin kecil keluarannya

Jenis-jenis Resistor diantaranya adalah :

1. Resistor yang Nilainya Tetap
2. Resistor yang Nilainya dapat diatur, Resistor Jenis ini sering disebut juga dengan Variable Resistor ataupun Potensiometer.
3. Resistor yang Nilainya dapat berubah sesuai dengan intensitas cahaya, Resistor jenis ini disebut dengan LDR atau *Light Dependent Resistor*
4. Resistor yang Nilainya dapat berubah sesuai dengan perubahan suhu, Resistor jenis ini disebut dengan PTC (*Positive Temperature Coefficient*) dan NTC *(Negative Temperature Coefficient)*

Resistor ini dilambangkan dengan Ohm (Ω) yang mana bentuk resistor ini seperti gambar 2.12 dibawah ini.



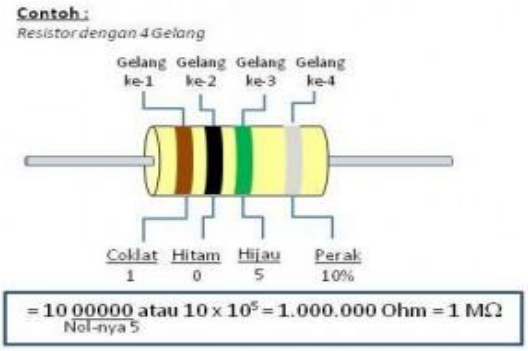
Gambar 2.12 Gambar dan Simbol Resistor

Kode warna yang ada pada resistor yang berfungsi standar untuk menghitung nilai resistansi yang terdapat pada resistor itu sendiri. Umumnya terdapat 4 warna di resistor, tetapi ada juga yang 5 warna. Warna emas dan perak biasanya terletak agak jauh dari warna lainnya sebagai tanda warna terakhir (warna toleransi). Tabel 2.6 dibawah ini adalah warna-warna yang terdapat di resistor:

Tabel 2.6 Warna Pada Resistor

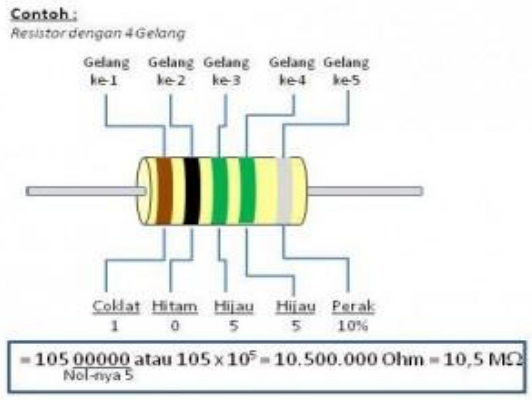
Warna		**Nilai**
	Hitam	0
	Cokelat	1
	Merah	2
	Orange	3
	Kuning	4
	Hijau	5
	Biru	6
	Ungu	7
	Abu-Abu	8
	Putih	9
	Emas	5%
	Perak	10%

Perhitungan untuk Resistor dengan 4 warna dapat dilihat pada gambar 2.13:



Gambar 2.13 Cara Baca Resistor 4 Warna

Perhitungan untuk Resistor dengan 5 Gelang warna dapat dilihat pada gambar 2.14 dibawah ini:



Gambar 2.14 Cara Baca Resistor 5 Warna

2.7.2 Kapasitor

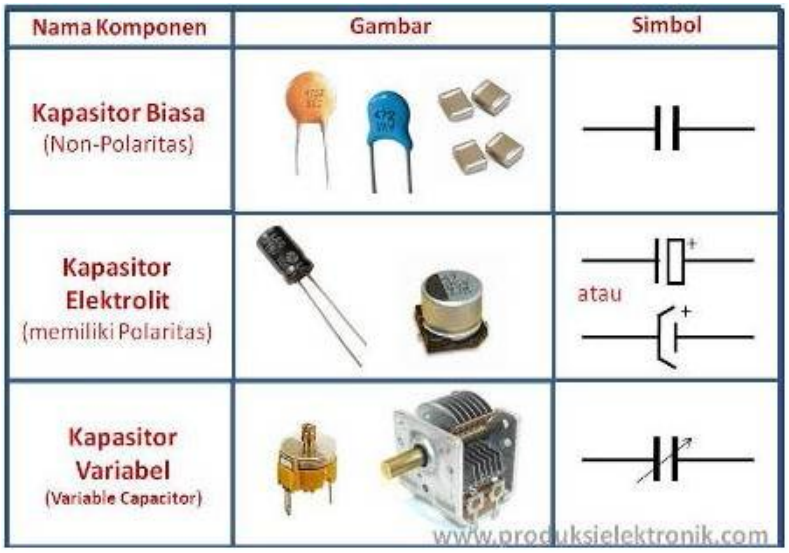
Kapasitor adalah suatu komponen elektronika yang berfungsi untuk menyimpan arus listrik dalam bentuk muatan, selain itu kapasitor juga dapat digunakan sebagai penyaring frekuensi. Kapasitas untuk menyimpan kemampuan kapasitor dalam muatan listrik disebut Farad (F) sedangkan simbol dari kapasitor adalah C (kapasitor). sebuah kapasitor pada dasarnya terbuat dari dua buah lempengan logam yang saling sejajar satu sama lain dan diantara kedua logam tersebut terdapat bahan isolator (Taranggono, 1996).

Jenis-jenis Kapasitor diantaranya adalah :

1. Kapasitor yang nilainya Tetap dan tidak ber-polaritas. Jika didasarkan pada bahan pembuatannya maka Kapasitor yang nilainya tetap terdiri dari Kapasitor Kertas, Kapasitor Mika, Kapasitor Polyster dan Kapasitor Keramik.
2. Kapasitor yang nilainya Tetap tetapi memiliki Polaritas Positif dan Negatif, Kapasitor tersebut adalah Kapasitor Elektrolit atau *Electrolyte Condensator* (ELCO) dan Kapasitor Tantalum
3. Kapasitor yang nilainya dapat diatur, Kapasitor jenis ini sering disebut dengan Variable Capasitor.

Kapasitor sering juga disebut sebagai kondensator. Kapasitor memiliki berbagai macam bentuk dan ukuran, tergantung dari kapasitas, tegangan kerja, dan

lain sebagainya. Gambar dan simbol kapasitor dapat dilihat pada gambar 2.15 dibawah ini:



Gambar 2.15 Gambar dan Simbol Kapasitor

Fungsi kapasitor pada rangkaian elektronika biasanya adalah sebagai berikut:

1. Kapasitor sebagai kopling, dilihat dari sifat dasar kapasitor yaitu dapat dilalui arus ac dan tidak dapat dilalui arus dc dapat dimanfaatkan untuk memisahkan 2 buah rangkaian yang saling tidak berhubungan secara dc tetapi masih berhubungan secara ac *(signal)*, artinya sebuah kapasitor berfungsi sebagai kopling atau penghubng antara 2 rangkaian yang berbeda.
2. Kapasitor berfungsi sebagai filter pada sebuah rangkaian *power supply*, maksudnya adalah kapasitor sebagai *ripple filter*, disini sifat dasar kapasitor yaitu dapat menyimpan muatan listrik yang berfungsi untuk memotong tegangan *ripple.*
3. Kapasitor sebagai penggeser fasa.
4. Kapasitor sebagai pembangkit frekuensi pada rangkaian oscilator.
5. Kapasitor digunakan juga untuk mencegah percikan bunga api pada sebuah saklar.

2.7.3 Transistor

Transistor adalah alat semikonduktor yang dipakai sebagai penguat, sebagai sirkuit pemutus dan penyambung *(switching)*, stabilisasi tegangan, modulasi sinyal atau sebagai fungsi lainnya. Transistor dapat berfungsi semacam kran

listrik, dimana berdasarkan arus inputnya (BJT) atau tegangan inputnya (FET), memungkinkan pengaliran listrik yang sangat akurat dari sirkuit sumber listriknya (Taranggono, 1996).

Jenis Transistor

1. Transistor NPN

   Adalah transistor positif,dimana transistor dapat bekerja mengalirkan arus listrik apabila basis dialiri tegangan arus positif.

2. Transistor PNP

   Adalah transistor negatif,dapat bekerja mengalirkan arus apabila basis dialiri tegangan negatif. Gambar dan Simbol Transistor dapat dilihat pada gambar 2.16 di bawah ini:

Gambar 2.16 Gambar dan Simbol Transistor

Pada umumnya, transistor memiliki 3 terminal. Tegangan atau arus yang dipasang di satu terminalnya mengatur arus yang lebih besar yang melalui 2 terminal lainnya. Transistor adalah komponen yang sangat penting dalam dunia elektronik modern. Dalam rangkaian analog, transistor digunakan dalam amplifier (penguat). Rangkaian analog melingkupi pengeras suara, sumber listrik stabil, dan penguat sinyal radio. Dalam rangkaian-rangkaian digital, transistor digunakan sebagai saklar berkecepatan tinggi. Beberapa transistor juga dapat dirangkai sedemikian rupa sehingga berfungsi sebagai *logic gate*, memori, dan komponen-komponen lainnya.

Berikut fungsi Transistor :

1. Sebagai penyearah arus.

2. Menahan sebagian arus.
3. Menguatkan arus.
4. Membangkitkan frekuensi rendah maupun tinggi.

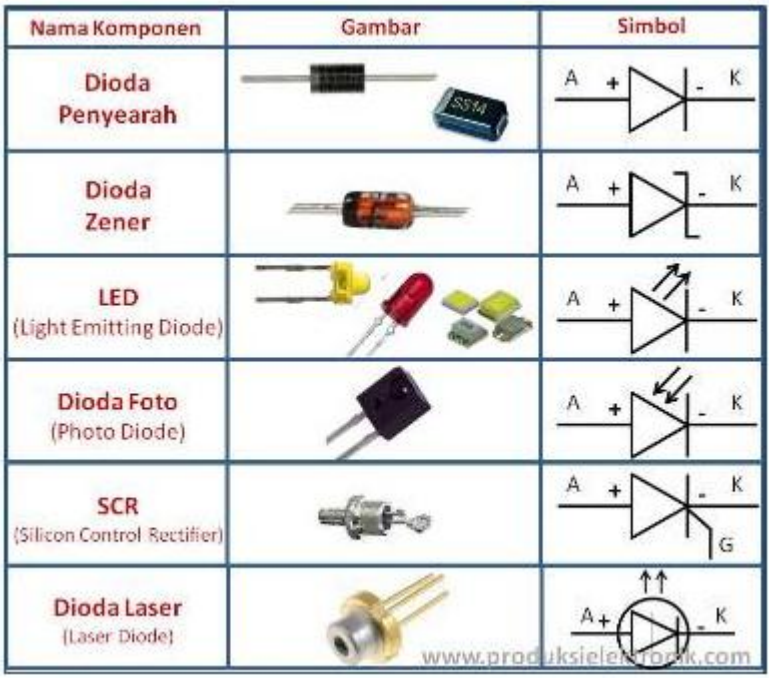
2.7.4 Dioda

Dioda adalah komponen aktif semikonduktor yang terdiri dari persambungan *(junction*) P-N.Sifat dioda yaitu dapat menghantarkan arus pada tegangan maju dan menghambat arus pada tegangan balik. Dioda berasal dari pendekatan kata dua elektroda yaitu anoda dan katoda (Taranggono, 1996). Dioda semikonduktor hanya melewatkan arus searah saja sehingga banyak digunakan sebagai komponen penyearah arus. Secara sederhana sebuah dioda bisa kita asumsikan sebuah katup, dimana katup tersebut akan terbuka manakala air yang mengalir dari belakang katup menuju kedepan, sedangkan katup akan menutup oleh dorongan aliran air dari depan katup.

Berdasarkan Fungsi Dioda terdiri dari :

1. Dioda Biasa atau dioda penyearah yang umumnya terbuat dari silikon dan berfungsi sebagai penyearah arus bolak balik (AC) ke arus searah (DC).
2. Dioda Zener yang berfungsi sebagai pengamanan rangkaian setelah tegangan yang ditentukan oleh dioda zener yang bersangkutan.
3. LED *(Light Emitting Diode)* yaitu dioda yang dapat memancarkan cahaya monokromatik.
4. Photodioda yaitu dioda yang peka dengan cahaya sehingga sering digunakan sebagai sensor.
5. Dioda *Schottky* (SCR atau *Silicon Control Rectifier*) adalah dioda yang berfungsi sebagai pengendali .
6. Dioda Laser yaitu dioda yang dapat memancar cahaya laser. dioda laser sering disingkat dengan LD.

Dioda disimbolkan dengan gambar anak panah yang pada ujungnya terdapat garis yang melintang. Simbol tersebut sebenarnya adalah sebagai perwakilan dari cara kerja dioda itu sendiri. Pada pangkal anak panah disebut juga sebagai anoda (kaki positif = P) dan pada ujung anak panah disebut sebagai katoda (kaki negative = N). Gambar dan simbol dioda dapat dilihat pada gambar 2.17 di bawah ini :



Gambar 2.17 Gambar dan Simbol Dioda

Fungsi Dioda

1. Sebagai penyearah, untuk dioda bridge
2. Sebagai penstabil tegangan *(voltage regulator),* untuk dioda zener
3. Pengaman / sekering
4. Sebagai rangkaian *clipper*, yaitu untuk memangkas / membuang level sinyal yang ada di atas atau di bawah level tegangan tertentu.
5. Sebagai rangkaian *clamper*, yaitu untuk menambahkan komponen DC kepada suatu sinyal AC
6. Sebagai pengganda tegangan.
7. Sebagai indikator, untuk LED *(light emiting diode)*
8. Sebagai sensor panas, contoh aplikasi pada rangkaian power amplifier
9. Sebagai sensor cahaya, untuk photodioda
10. Sebagai rangkaian VCO *(voltage controlled oscilator),* untuk dioda *varactor*

2.8 Perangkat Lunak *(Software)*

Adapun software yang digunakan dalam penyortiran barang penumpang berdasarkan tujuan ini adalah bahasa pemrograman C dan Pemrograman Visual

Basic. Pengenalan software ini digunakan sebagai referensi acuan dalam mendukung teori pemrograman.

2.8.1 *CodeVision AVR*

CodeVision AVR adalah suatu kompiler berbasis bahasa C, yang terintegrasi untuk memprogram dan sekaligus compiler aplikasi AVR *(Alf and Vegard's Risc processor)* terhadap mikrokontroler dengan sistem berbasis window. *CodeVision* AVR ini dapat mengimplematasikan hampir semua interuksi bahasa C yang sesuai dengan arsitektur AVR, bahkan terdapat beberapa keunggulan tambahan untuk memenuhi keunggulan spesifikasi dari *CodeVision* AVR yaitu hasil kompilasi studio debugger dari ATMEL. (Budi,1993)

Khusus untuk *library* fungsi, disamping *library* standar (seperti fungsi-fungsi matematik, manipulasi string, pengaksesan memori dan sebagainya), *CodeVisionAVR* juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol. Beberapa fungsi *library* yang penting diantaranya adalah fungsi-fungsi untuk pengaksesan LCD, komunikasi I2C, IC RTC *(Real time Clock),* sensor suhu LM75, SPI *(Serial Peripheral Interface)* dan lain sebagainya.

Integrated Development Environtment (IDE) telah diadaptasikan pada chip AVR yaitu *In-System Programmer software,* memungkinkan *programmer* untuk mentransfer program ke chip mikrokontroler secara otomatis setelah proses assembly/kompilasi berhasil. *In-System Programmer software* di desain untuk bekerja dan dapat berjalan dengan perangkat lunak lain seperti AVR *Dragon,* AVRISP, Atmel STK500, dan lain sebagainya

Disamping *library standar* C, *CodeVisionAVR* C compiler memiliki *library* lain untuk:

- Modul LCD Alphanumerik
- *Delays*
- *Protokol semikonduktor Maxim/Dallas*

CodeVisionAVR juga memiliki *CodeWizardAVR* sebagaki generator program otomatis, yang memungkinkan kita untuk menulis, segala bentuk pengaturan *chip*

dalam waktu singkat, dan semua kode yang dibutuhkan untuk mengimplementasikan fungsi-fungsi seperti:

1. **Pengaturan akses *External Memory***

   Untuk *chip-chip* AVR yang memungkinkan koneksi memori eksternal SRAM, dapat juga mengatur ukuran memori dan wait state (tahap tunggu) dari memori ketika memori tersebut diakses.

2. **Identifikasi *chip reset source***

   Adalah suatu layanan dimana kita dapat membuat kode secara otomatis yang dapat mengidentifikasi kondisi yang menyebabkan c*hip* di *reset.*

3. **Inisialisasi *port input/output***

   Pengaturan port-port yang kan dijadikan gerbang masukan dan keluaran dapat secara otomatis digenerate codenya. Yang kita lakukan hanya memilih port-port yang akan digunakan sebagai input atau output.

4. **Inisialisasi *Interupsi external***

   Pengaturan interupsi eksternal yang nantinya akan digunakan untuk menginterupsi program utama

5. **Inisialisasi *timers/counters***

   Pengaturan timers yang berfungsi untuk mengatur frekwensi yang nantinya digunakan pada interupsi.

6. **Inisialisasi *timer watchdog***

   Pengaturan *timers* yang berfungsi untuk mengatur frekuensi yang nantinya digunakan pada interupsi, sehingga interupsi akan dilayani oleh suatu fungsi *wdt_timeout_isr* .

7. **Inisialisasi UART(USART) dan komunuikasi serial**

   Pengaturan komunikasi serial sebagai penerima atau pengirim data.

8. **Inisialisasi komparasi analog**

   Pengaturan yang berkaitan dengan masukan data yang digunakan dalam aplikasi yang membutuhkan komparasi pada ADC nya.

9. **Inisialisasi ADC**

   Pengaturan ADC(*Analog-Digital Converter)* yang berfungsi untuk merubah format analog menjadi format digital untuk diolah lebih lanjut.

10. Inisialisasi antarmuka SPI

Pengaturan chip yang berkaitan dengan *Clock rate, Clock Phase,* dan lainnya.

11. Inisialisasi antarmuka *Two Wire* BUS

Pengaturan Chip yang berhubungan dengan pola jalur komunikasi antara register yang terdapat pada chip AVR.

12. Inisialisasi antarmuka CAN

Pengaturan chip yang lebih kompleks, yang dapat mengatur interupsi, transmisi data, timers, dan lainnya.

13. Inisialisasi sensor temperatur, thermometer, dan lainnya

Pengaturan yang berhubungan dengan sensor temperatur *one wire bus*, memiliki fungsi-fungsi yang ada pada librari *CodeVisionAVR.*

14. Inisialisasi *one wire bus*

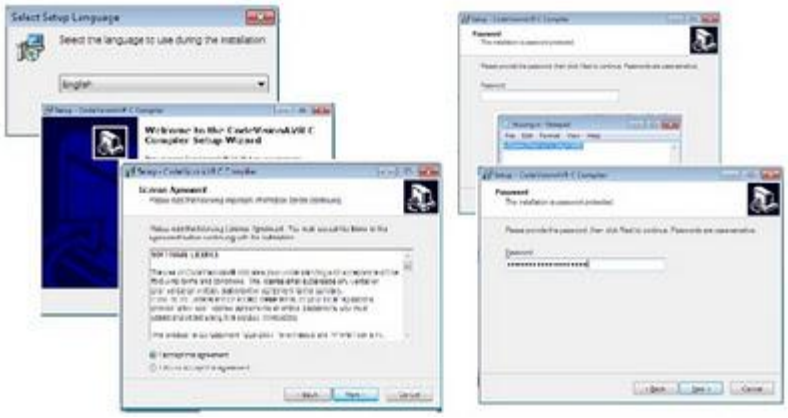
Pengaturan yang berhubungan dengan sensor temperatur yang memiliki fungsi-fungsi yang ada pada librari *CodeVisionAVR*. Seperti *Maxim/Dallas Semiconductor.*

15. Inisialisasi modul LCD

Pengaturan port-port yang kan digunakan sebagai penghubung dengan LCD alphanumerik.

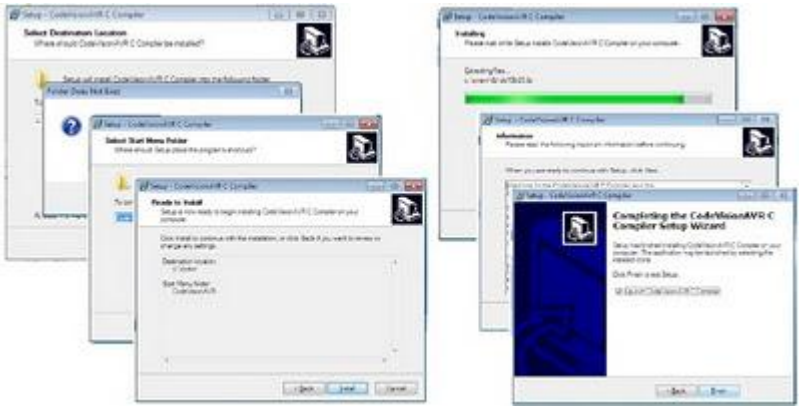
Berikut adalah langkah-langkah Instalasi *CodeVisionAVR:*

a. Buka folder tempat kita menyimpan master *software CodeVisionAVR,* pilih file ***Setup*** dengan melakukan double click atau **klik kanan - *Run Adsministrator* - OK**.
b. Pilih **English** pada jendela ***Select Setup Language***, klik **OK**.
c. Pada jendela ***Welcome to the CodeVisionAVR C Compiler Setup Wizard*** pilih menu **Next**.
d. Pilih point ***I accept the agreement*** pada jendela ***Licence Agreement***, kemudian pilih ***Next*** jika kita setuju dengan perjanjian yang diajukan pada jendela tersebut.
e. Selanjutnya akan tampil jendela yang meminta *password*, pada kolom ***password*** silahkan ***copy-paste password*** dari file yang ada. Dapat di;ihat pada gambar 2.18 dibawah ini.



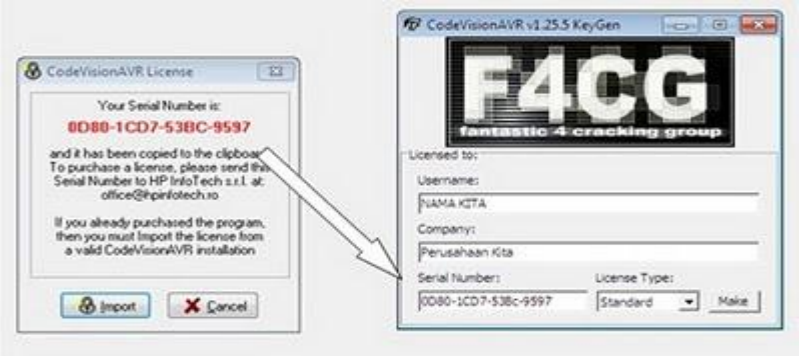
Gambar 2.18 Proses Instal *CodeVisionAVR (Part1)*

f. Lanjutkan proses instalasi dengan memilih menu ***Next/Yes/OK*** pada jendela-jendela berikutnya. Dapat di;ihat pada gambar 2.19 dibawah ini.



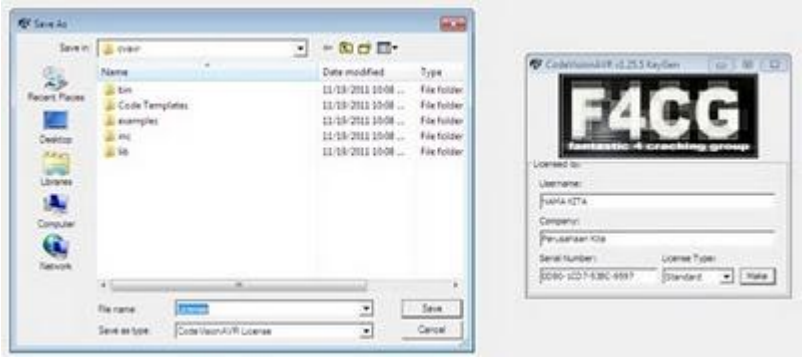
Gambar 2.19 Proses Instal *CodeVisionAVR (Part 2)*

g. Proses Instal memang seperti sudah selesai, namun *software* belum dapat digunakan. Pada saat CodeVisionAVR akan digunakan maka akan muncul jendela **CodeVisionAVR Licence**. Jendela tersebut meminta ***file licence***. Dapat di;ihat pada gambar 2.20 dibawah ini.



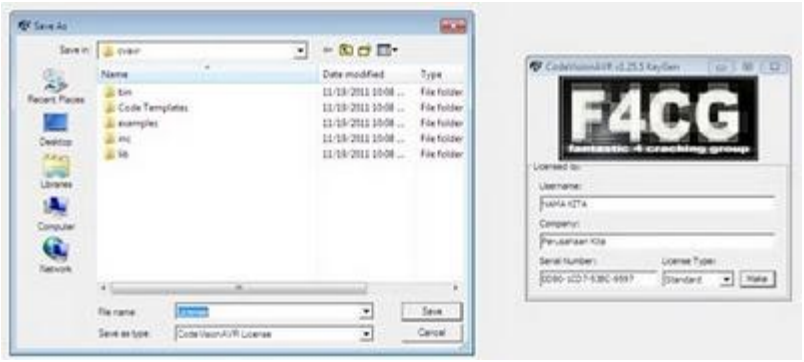
Gambar 2.20 Proses Instal *CodeVisionAVR (Part 3)*

h. Langkah yang perlu kita lakukan adalah membuka file Keygen yang juga terletak pada folder tempat Kita menyimpan master CodeVisionAVR.
i. Masukkan Nama Kita pada kolom *Username*, Nama Perusahaan Kita pada kolom *Company* dan masukkan serial number yang telah tertulis pada jendela ***CodeVisionAVR Licence***:
j. Kemudian pilih menu ***Make*** pada jendela ***KeyGen***. Arahkan pada folder tempat kita menyimpan instalasi *CodeVisionAVR* kemudian pilih menu ***Save*** maka akan muncul jendela dengan tulisan *Licence file generated! - OK*. Dapat di;ihat pada gambar 2.21 dibawah ini.



Gambar 2.21 Proses Instal *CodeVisionAVR (Part4)*

k. Selanjutnya silahkan kembali ke jendela ***CodeVisionAVR Licence***, kemudian pilih ***Import*** dan arahkan pada folder kita menyimpan file *Licence* dari KeyGen pada langkah sebelumnya tadi. Pilih file ***Licence*** kemudian pilih ***Open***. Dapat di;ihat pada gambar 2.22 dibawah ini.



Gambar 2.22 Proses Instal *CodeVisionAVR (Part5)*

l. Proses Instalasi *CodeVision AVR* telah selesai.

2.8.2 Pengenalan Bahasa C

Bahasa C diciptakan oleh Dennis Ritchie tahun 1972 di *Bell Laboratories.*

Kelebihan Bahasa C: (Anton,2009)

a. Bahasa C tersedia hampir di semua jenis komputer.
b. Kode bahasa C sifatnya adalah portable dan fleksibel untuk semua jenis computer.
c. Bahasa C hanya menyediakan sedikit kata-kata kunci. hanya terdapat 32 kata kunci.
d. Proses executable program bahasa C lebih cepat
e. Dukungan pustaka yang banyak.
f. C adalah bahasa yang terstruktur
g. Bahasa C termasuk bahasa tingkat menengah

Kekurangan Bahasa C:

a. Banyaknya operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai.
b. Bagi pemula pada umumnya akan kesulitan menggunakan pointer.

2.8.3 Mengkompilasi Program

Suatu *source* program C baru dapat dijalankan setelah melalui tahap kompilasi dan penggabungan. Tahap kompilasi dimaksudkan untuk memeriksa *source*-program sesuai dengan kaidah-kaidah yang berlaku di dalam bahasa pemrograman C. Tahap kompilasi akan menghasilkan *relocatable object file.* File-

file objek tersebut kemudian digabung dengan perpustakaan-fungsi yang sesuai. untuk menghasilkan suatu *executable-program.*

Shortcut yang digunakan untuk mengkompile:

- *ALT + F9* dipakai untuk melakukan pengecekan jika ada error pada program yang telah kita buat.
- *CTRL + F9* dipakai untuk menjalankan program yang telah kita buat atau bisa juga dengan mengklik tombol *debug* pada *toolbar.*

2.8.4 Struktur Bahasa Pemrograman C

Terdapat banyak struktur dalam Pemrograman C yaitu:

1. *Header File*

   adalah berkas yang berisi prototype fungsi. definisi konstanta. dan definisi variable. Fungsi adalah kumpulan code C yang diberi nama dan ketika nama tersebut dipanggil maka kumpulan kode tersebut dijalankan.

   Contoh :

   stdio.h

   math.h

   conio.h

2. Preprosesor *Directive* (#include)

   Preprosesor directive adalah bagian yang berisi pengikutsertaan file atau berkas-berkas fungsi maupun pendefinisian konstanta.

   Contoh:

   #include <stdio.h>

   #include phi 3.14

3. Void

   artinya fungsi yang mengikutinya tidak memiliki nilai kembalian (return).

4. Main ( )

   Fungsi main ( ) adalah fungsi yang pertama kali dijalankan ketika program dieksekusi.tanpa fungsi main suatu program tidak dapat dieksekusi namun dapat dikompilasi.

5. Statement

   Statement adalah instruksi atau perintah kepada suatu program ketika program itu

dieksekusi untuk menjalankan suatu aksi. Setiap statement diakhiri dengan titik-koma(;).

2.8.5 Kata Kunci *(Keyword)*

Kata kunci-kata kunci yang terdapat di C dapat dilihat di tabel 2.7 dibawah ini:

Tabel 2.7 Kata Kunci yang Terdapat di Bahasa C

auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

2.8.6 Identifier

Identifier atau nama pengenal adalah nama yang ditentukan sendiri oleh pemrogram yang digunakan untuk menyimpan nilai, misalnya nama variable, nama konstanta, nama suatu elemen (misalnya: nama fungsi, nama tipe data, dll). Identifier punya ketentuan sebagai berikut :

a. Maksimum 32 karakter (bila lebih dari 32 karakter maka yang diperhatikan hanya 32 karakter pertama saja).
b. Case sensitive: membedakan huruf besar dan huruf kecilnya.
c. Karakter pertama harus karakter atau underscore (_) . selebihnya boleh angka.
d. Tidak boleh mengandung spasi atau blank.
e. Tidak boleh menggunakan kata yang sama dengan kata kunci dan fungsi.

2.8.7 Variabel

Variabel adalah *identifier* yang nilainya dapat berubah atau diubah selama program berjalan (dieksekusi). Pengubahnya adalah user atau proses.

- Deklarasi variabel (tipe_data nama_variabel;)

Variabel yang akan digunakan dalam program haruslah dideklarasikan terlebih dahulu. Pengertian deklarasi di sini berarti memesan memori dan menentukan jenis data yang bisa disimpan di dalamnya.

Contoh:

Int a, b, c;

- Inisialisasi variabel (tipe_data nama_variabel = nilai;)

  Int a=12, b=7, c=0;

2.8.9 Konstanta

Konstanta adalah identifier yang nilainya tetap selama program berjalan/ dieksekusi. Cara untuk mengubahnya hanya melalui source codenya saja seperti halnya variabel, konstanta juga memiliki tipe. Penulisan konstanta mempunyai aturan tersendiri, sesuai dengan tipe masing-masing.

1. Konstanta karakter misalnya ditulis dengan diawali dan diakhiri dengan tanda petik tunggal, contohnya : 'A' dan '@'.
2. Konstanta integer ditulis dengan tanda mengandung pemisah ribuan dan tidak mengandung bagian pecahan, contohnya : –1 dan 32767.
3. Konstanta real (*float* dan *double*) bisa mengandung pecahan (dengan tanda berupa titik) dan nilainya bisa ditulis dalam bentuk eksponensial (menggunakan tanda e), contohnya : 27.5f (untuk tipe *float*) atau 27.5 (untuk tipe *double*) dan 2.1e+5 (maksudnya 2,1 x 105).
4. Konstanta string merupakan deretan karakter yang diawali dan diakhiri dengan tanda petik-ganda (“), contohnya :“Pemrograman Dasar C”.

   Contoh :

   ```
   #define phi 3.14
   #define max_data 50
   ```

2.8.10 Tipe Data Dasar

Data merupakan suatu nilai yang bisa dinyatakan dalam bentuk konstanta atau variabel. Konstanta menyatakan nilai yang tetap, sedangkan variabel menyatakan nilai yang dapat diubah-ubah selama eksekusi berlangsung. Untuk mengetahui ukuran memory bisa dipakai fungsi sizeof (<tipe_data>).

Catatan:

Ukuran dan kawasan dari masing-masing tipe data adalah bergantung pada jenis mesin yang digunakan (misalnya mesin 16 bit bisa jadi memberikan hasil berbeda dengan mesin 32 bit). Untuk ukuran memori tipe data dapat dilihat di tabel 2.8 dibawah ini:

Tabel 2.8 Ukuran Memori untuk Tipe Data

Tipe Data	Ukuran Memori	Kawasan
unsigned char	8 bits	0 s/d 255
char	8 bits	-128 s/d 127
short int	16 bits	-32.768 s/d 32.767
unsigned int	32 bits	0 s/d 4.294.967.295
int	32 bits	-2.147.483.648 s/d 2.147.483.647
unsigned long	32 bits	0 s/d 4.294.967.295
enum	16 bits	-2147483.648 to 2.147.483.648
long	32 bits	-2.147.483.648 s/d 2.147.483.647
float	32 bits	3,4 x 10-38 s/d 3,4 x 10+38
double	64 bits	1.7 x 10-308 to 1.7 x 10+308
long double	80 bits	3.4 x 10-4932 to 3.4 x 10+4932
near (pointer)	32 bits	not applicable
far (pointer)	32 bits	not applicable

Untuk menampilkan hasil output dibutuhkan kode format, berikut pada tabel 2.9 adalah daftar kode Format dari tipe data dasar:

Tabel 2.9 Kode Format Dalam Bahasa C

Kode format	Kegunaan
%c	Menampilkan sebuah karakter
%s	Menampilkan nilai string
%d	Menampilkan nilai decimal integer
%i	Menampilkan nilai decimal integer
%u	Menampilkan nilai decimal integer tidak bertanda (unsigned integer)
%ld	Menampilkan nilai decimal long integer
%lu	Menampilkan nilai decimal long integer tak bertanda
%li	Menampilkan nilai decimal long integer
%hu	Menampilkan nilai decimal short integer tak bertanda
%hi	Menampilkan nilai decimal short integer
%x	Menampilkan nilai heksa decimal integer
%o	Menampilkan nilai okta integer
%f	Menampilkan nilai pecahan / float
%e	Menampilkan nilai float scientific
%g	Sebagai pengganti %f atau %e tergantung yang terpendek
%lf	Menampilkan nilai pecahan double
%le	Menampilkan nilai pecahan double
%lg	Menampilkan nilai pecahan double
%p	Menampilkan suatu alamat memory untuk pointer

2.8.11 Karakter Escape

Karakter escape adalah karakter yang diawali dengan tanda backslash (/), yang masingmasing memiliki makna tertentu. Berikut adalah daftar karakter:

Tabel 2.10 Karakter Escape

KARAKTER ESCAPE	ARTI
\a	Bunyi bel (speaker komputer)
\b	Mundur satu spasi (backspace)
\f	Ganti halaman (form feed)
\n	Ganti baris baru (new line)
\r	Ke kolom pertama baris yang sama (carriage return)
\t	Tab horizontal
\v	Tab vertical
\0	Nilai kosong (NULL)
\'	Karakter petik tunggal
\"	Karakter petik ganda
\\	Garis miring terbalik (backslash)
\?	Karakter tanda tanya
\DDD	Menyatakan sebuah karakter yang nilai ASCII nya sama dengan nilai octal DDD
\xHH	Menyatakan sebuah karakter yang nilai ASCII nya sama dengan nilai heksadesimal HH

2.9 Pengenalan Pemrograman *Visual Basic*

Microsoft *Visual Basic* merupakan bahasa pemrograman yang berbasis microsoft windows, sebagai bahasa pemrograman yang mutakhir, **Visual Basic** 6.0 didesain untuk dapat memanfaatkan fasilitas yang tersedia dalam Microsoft windows. Microsoft *Visual Basic* 6.0 juga merupakan bahasa pemrograman *Object Oriented Programing* (OOP), yaitu pemrograman yang berorientasi objek. *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.

Visual Basic 6.0 adalah salah satu development tools untuk membangun aplikasi dalam lingkungan windows. Dalam pengembangan aplikasi, visual basic

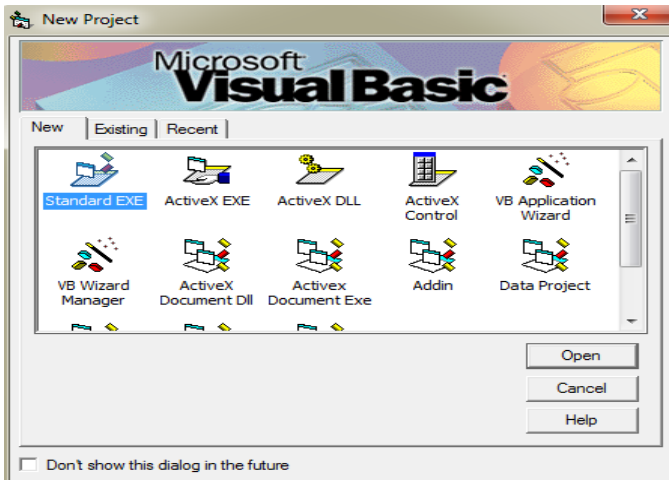
menggunakan pendekatan visual untuk merancang user interface atau tampilan dalam bentuk form, sedangkan untuk kodingnya menggunakan bahasa basic yang cenderung mudah dipelajari. Visual basic telah menjadi *tools* yang terkenal bagi para pemula maupun developer. Visual Basic 6.0 sebetulnya perkembangan dari versi sebelumnya dengan beberapa penambahan komponen yang sedang tren saat ini, seperti kemampuan pemrograman internet dengan DHTML *(Dynamic HyperText Mark Language),* dan beberapa penambahan fitur database dan multimedia yang semakin baik.

Secara umum ada beberapa manfaat yang diperoleh dari pemakaian program *Visual Basic*, diantaranya:

a. Dipakai dalam membuat program aplikasi berbasis Windows.
b. Dipakai dalam membuat obyek-obyek pembantu program, seperti fasilitas Help, kontrol ActiveX, aplikasi Internet, dan sebagainya.
c. Digunakan untuk menguji program *(Debugging)* dan menghasilkan program akhir EXE yang bersifat Executable, atau dapat langsung dijalankan. (Subari & Yuswanto, 2008)

2.9.1 *Interface Visual Basic* 6.0

Untuk memulai pemrograman dengan *visual basic,* jalankan program *visual basic* 6.0. Selanjutnya pada tampilan awal akan ditampilkan kotak dialog *new project* seperti pada gambar 2.23 di bawah ini.



Gambar 2.23 Tampilan Kotak Dialog New Project

Pada kotak dialog tersebut terdapat 3 buah tab yang terdiri dari:

a. ***New*** (menampilkan daftar pilihan untuk membuat *project* baru)

b. ***Existing*** (untuk *browsing* dan membuka *project*)

c. ***Recent*** (untuk membuka *project* yang sering digunakan).

Visual Basic 6.0 menyediakan 13 jenis *project* yang bisa dibuat seperti terlihat pada gambar 1.3 di atas. Ada beberapa *project* yang biasa digunakan oleh banyak pengguna *Visual Basic*, antara lain:

a. *Standard EXE: Project* standar dalam Visual Basic dengan komponen-komponen standar. Jenis *project* ini sangat sederhana, tetapi memiliki keunggulan bahwa semua komponennya dapat diakui oleh semua unit komputer dan semua user meskipun bukan administrator. Pada buku ini akan digunakan *project Standard EXE* ini, sebagai konsep pemrograman visualnya.

b. *ActiveX EXE: Project* ini adalah *project ActiveX* berisi komponen-komponen kemampuan intuk berinteraksi dengan semua aplikasi di sistem operasi *windows*.

c. *ActiveX DLL: Project* ini menghasilkan sebuah aplikasi *library* yang selanjutnya dapat digunakan oleh semua aplikasi di sistem operasi *Windows.*

d. *ActiveX Control: Project* ini menghasilkan komponen-komponen baru untuk aplikasi Visual Basic yang lain.

e. *VB Application Wizard: Project* ini memandu pengguna untuk membuat aplikasi secara mudah tanpa harus pusing-pusing dengan perintah-perintah pemrograman.

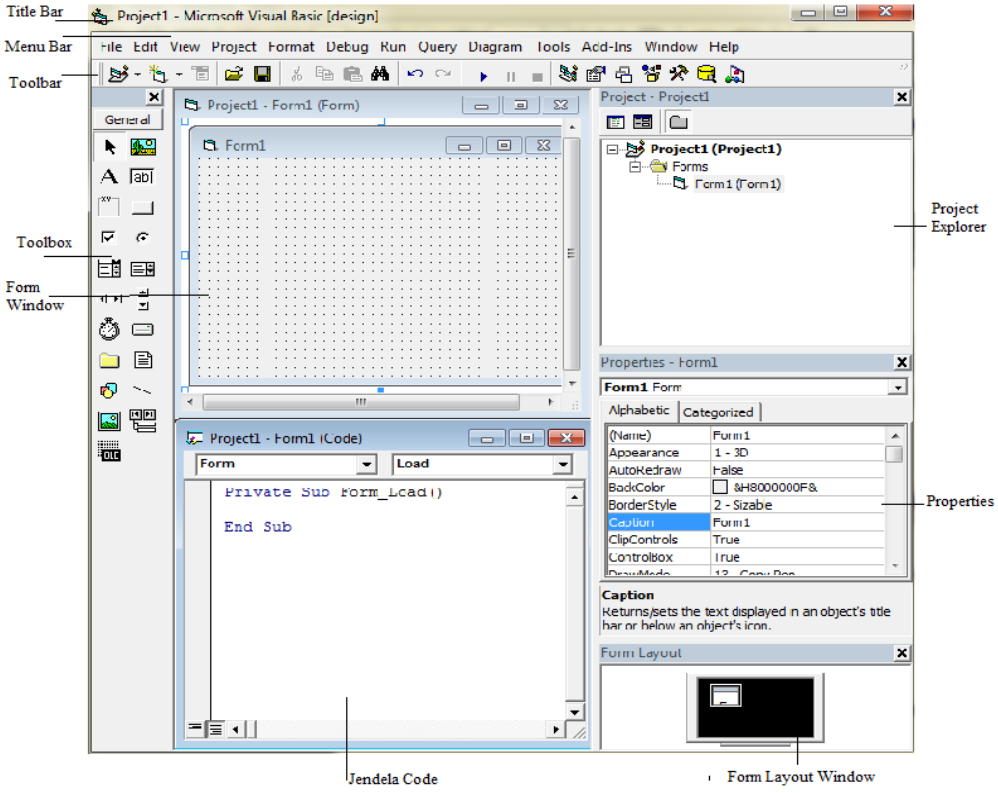
f. *Addin: Project* seperti *Standard EXE* tetapi dengan berbagai macam komponen tambahan yang memungkinkan kebebasan kreasi dari pengguna.

g. *Data project: Project* ini melengkapi komponennya dengan komponen-komponen database. Sehingga bisa dikatakan *project* ini memang disediakan untuk keperluan pembuatan aplikasi database.

h. *DHTML Application: Project* ini digunakan untuk membuat aplikasi internet pada sisi *client* (*client side*) dengan fungsi-fungsi DHTML.

i. *IIS Application: Project* ini menghasilkan aplikasi internet pada sisi *server* (*server side*) dengan komponen-komponen CGI (*Common Gateway Interface*).

Untuk pembuatan program pertama kali pilih tab New, pilih Standard EXE lalu klik Open. Selanjutnya muncul tampilan utama Visual Basic 6.0 seperti pada gambar 2.24 di bawah ini



Gambar 2.24 IDE Microsoft Visual Basic 6.0

1. *Title Bar*

   Title bar merupakan batang jendela dari program visual basic 6.0 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* dan *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.

2. *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 :

a. *File* : terdiri dari perintah-perintah untuk membuka, menutup, menyimpan, mencetak, mengcompile *Project* yang sedang kita kerjakan
b. *Edit* : kumpulan perintah yang membantu kita memanipulasi penulisan *code* (*listing* program)
c. *View* : terdiri dari perintah untuk melihat/berpindah antar *window* didalam lingkungan pengembangan
d. *Project* : sekumpulan perintah untuk menambah komponen *Project* (*Form*, *Module*, dll) dan tempat dimana kita akan men-*set-up properties Project*
e. *Format* : perintah-perintah untuk memanipulasi kontrol yang terdapat pada sebuah *Form*
f. *Debug* : Sekumpulan perintah yang berguna untuk melacak *bugs* ataupun melacak/*trace listing* program yang sedang kita kerjakan *line*-per-*line*
g. *Run* : perintah untuk menjalankan, *pause* dan menghentikan jalannya Program/aplikasi
h. *Tools* : terdiri dari *wizard-wizard* yang sangat membantu pengembangan aplikasi
i. *Add-Ins* : sekumpulan *component/wizard* yang bisa digunakan dalam pembuatan aplikasi
j. *Window* : kumpulan *window-window* yang terdapat didalam *Project*
k. *Help* : berisi *file-file* bantuan/*help* dan tentang Visual Basic itu sendiri

3. Toolbars

Toolbars merupakan sebuah batang yang berisi kumpulan tombol yang terletak dibagian bawah menu bar atau terdapat didalam Menu Bar *(shortcut)* yang dapat digunakan untuk menjalankan perintah memanipulasi *Project*. Pada kondisi *default* program visual basic hanya menampilkan *toolbars* standar. Namun dapat pula di-*set* sesuai dengan keinginan kita sendiri.

4. Project Explorer

Project Explorer merupakan suatu kumpulan *module* 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

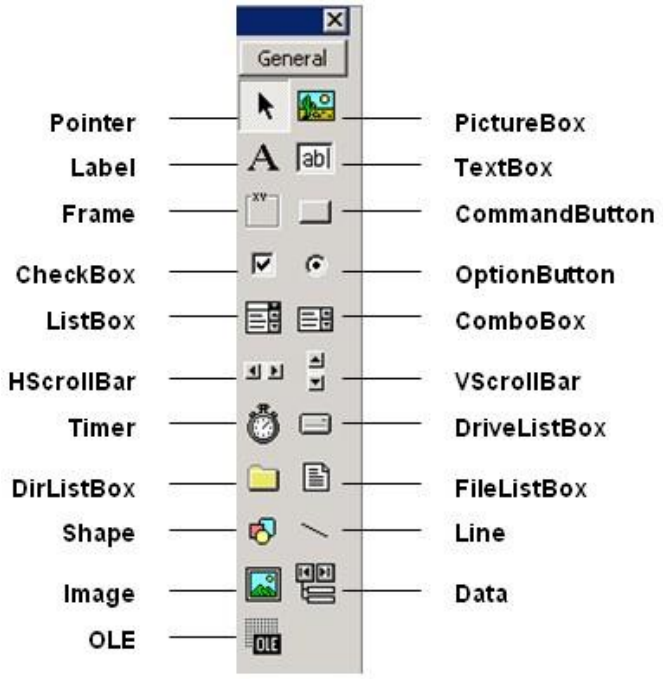
program. Apabila membuat suatu program aplikasi baru maka secara otomatis *project* tersebut akan diisi dengan *form* baru. Dalam jendela *project explorer* ditampilkan suatu struktur hirarki dari sebuah *project* itu sendiri yang berisi semua item yang terkandung di dalamnya.

5. *Form*

   Form merupakan *windows* atau jendela di mana akan dibuat *user interface*. Kita dapat menambahkan sebanyak mungkin *form* kedalam aplikasi kita sesuai dengan kebutuhan.

6. *Toolbox* atau kontrol

   Merupakan tampilan berbasis grafis yang dimasukkan pada *form* untuk membuat interaksi dengan pemakai. Bentuk *toolbox* visual basic dapat dilihat pada gambar 2.25 pada gambar di bawah ini



Gambar 2.25 Toolbox Pada Visual Basic 6.0

Adapun secara garis besar fungsi dari masing-masing control tersebut adalah sebagai berikut :

a. Pointer

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

b. Picturebox

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

c. Label

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

d. Textbox

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

e. Frame

Frame adalah kontrol yang digunakan sebagai kontainer bagi kontrol lainnya.

f. CommandButton

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

g. CheckBox

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

h. OptionButton

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

i. ListBox

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

j. ComboBox

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

k. HScrollbar/VScrollbar

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

l. Timer

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

m. DriveListBox, DirListBox, dan *File*ListBox

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

n. Shape dan Line

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

o. 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.*

p. Data dan Adodc

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

q. OLE

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

7. Properties

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

8. 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.

9. 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* dan *drop*. (Fajrillah Hasballah, 2009)

2.10 *Flowchart*

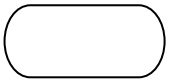
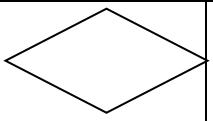
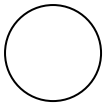
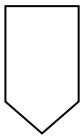
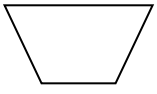
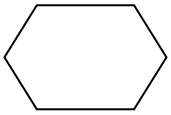
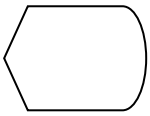
Menurut Malik (2005) menyebutkan bahwa Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung.

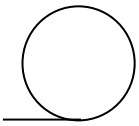
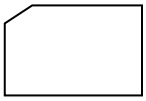
Flowchart ini merupakan langkah awal pembuatan program. Dengan adanya flowchart urutan poses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah. Setelah flowchart selesai disusun, selanjutnya pemrogram (programmer) menerjemahkannya ke bentuk program dengan bahsa pemrograman. Flowchart atau bagan alur merupakan metode untuk menggambarkan tahap-tahap penyelesaian masalah (prosedur) secara grafik dengan simbol-simbol standar yang mudah dipahami. Flowchart menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Pedoman-pedoman dalam membuat flowchart

1. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
2. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
3. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja.
4. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
5. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
6. Gunakan simbol-simbol flowchart yang standar.

Tabel 2.11 Simbol-Simbol Flowchart

No	Simbol	Nama	Keterangan
1		***Terminator***	simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan
2		***Flow***	simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain.
3		***Processing***	Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer
4		***Input-Output***	Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
5		***Predefined Process***	Simbol untuk pelaksanaan suatu bagian (sub-program) /prosedure
6		***Decision***	Simbol pemilihan proses berdasarkan kondisi yang ada.
7		***Connector***	simbol untuk keluar – masuk atau penyambungan proses dalam lembar / halaman yang sama.
8		***Off-line Connector***	simbol untuk keluar – masuk atau penyambungan proses pada lembar / halaman yang berbeda.
9		***Manual Operation***	Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer
11		***Preparation***	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
12		***Display***	Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.

No	Simbol	Nama	Keterangan
13		***Storage Data***	Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
14		***Magnetik Tape Unit***	Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
15		***Punch Card***	Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
16		***Dokumen***	Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.