

BAB II

TINJAUAN PUSTAKA

2.1 *Radio Frequency Identification (RFID)*

RFID adalah sebuah teknologi penangkapan data yang memanfaatkan frekuensi radio dalam sistem kerjanya yang dapat digunakan secara elektronik untuk mengidentifikasi, melacak dan menyimpan informasi yang tersimpan dalam *tag RFID*. Perhatian terhadap *RFID* dalam lingkungan media massa maupun akademis yang populer telah meningkat dalam beberapa tahun ini. Salah satu buktinya adalah usaha dari organisasi-organisasi yang besar seperti Wal-Mart, Procter and Gamble, dan Departemen Pertahanan Amerika Serikat untuk menggunakan *RFID* sebagai suatu alat untuk mengontrol secara otomatis terhadap rantai supply mereka. Harga *tag* yang menurun dan standarisasi yang dinamis telah menyebabkan kita berada pada ambang ledakan penggunaan *RFID*.

Para pengamat *RFID* menganggap *RFID* sebagai suksesor dari *Barcode* optik yang banyak dicetak pada barang-barang dagangan dengan dua keunggulan pembeda.

a. Identifikasi yang unik

Sebuah *Barcode* mengindikasikan tipe obyek tempat ia dicetak, misalnya “Ini adalah sebatang coklat merek ABC dengan kadar 70% dan berat 100 gram”. Sebuah *tag RFID* selangkah lebih maju dengan mengemisikan sebuah nomor seri unik di antara jutaan obyek yang identik, sehingga ia dapat mengindikasikan “Ini adalah sebatang coklat merek ABC dengan kadar 70%

dan berat 100 gram, nomor seri 897348738”. Identifikasi yang unik dalam *RFID* ini dapat berperan sebagai pointer terhadap masukan basis data yang menyimpan banyak *history* transaksi untuk item-item individu.

b. Otomasi

Barkode di-*scan* secara optik, memerlukan kontak *line-of-sight* dengan *Reader*, dan tentu saja peletakan fisik yang tepat dari obyek yang di-*scan*. Kecuali pada lingkungan yang benar-benar terkontrol, *scanning* terhadap

Barcode memerlukan campur tangan manusia, sebaliknya *tag-tag RFID* dapat dibaca tanpa kontak *line of sight* dan tanpa penempatan yang presisi. *Reader RFID* dapat melakukan *scan* terhadap *tag-tag* sebanyak ratusan perdetik.

Sebagai suksesor dari *Barcode*, *RFID* dapat melakukan kontrol otomatis untuk banyak hal. Sistem-sistem *RFID* menawarkan peningkatan efisiensi dalam pengendalian inventaris (*inventory control*), logistik dan manajemen rantai *supply* (*supply chain management*).

2.1.1 Komponen-komponen Utama Sistem RFID

Secara garis besar sebuah sistem *RFID* terdiri atas tiga komponen utama, yaitu *tag*, *Reader* dan basis data (Gambar 2.1). Secara ringkas, mekanisme kerja yang terjadi dalam sebuah sistem *RFID* adalah bahwa sebuah *Reader* frekuensi radio melakukan *scanning* terhadap data yang tersimpan dalam *tag*, kemudian mengirimkan informasi tersebut ke sebuah basis data yang menyimpan data yang terkandung dalam *tag* tersebut.

a. Tag RFID

Sebuah *tag RFID* atau *transponder*, terdiri atas sebuah mikro (*microchip*) dan sebuah antena seperti yang terlihat pada. Chip tersebut menyimpan nomor seri yang unik atau informasi lainnya tergantung kepada tipe memorinya. Tipe memori itu sendiri dapat *read-only*, *read-write*, atau *write-onceread-many*. Antena yang terpasang pada chip mikro mengirimkan informasi dari chip ke *Reader*. Biasanya rentang pembacaan diindikasikan dengan besarnya antena. Antena yang lebih besar mengindikasikan rentang pembacaan yang lebih jauh. *Tag* tersebut terpasang atau tertanam dalam obyek yang akan diidentifikasi. *Tag* dapat di-*scan* dengan *Reader* bergerak maupun stasioner menggunakan gelombang radio.

Tag RFID sangat bervariasi dalam hal bentuk dan ukuran. *Tag* versi paling sederhana adalah *tag* pasif, yaitu *tag* yang tidak memiliki catu daya sendiri serta tidak dapat menginisiasi komunikasi dengan *Reader*. Sebagai gantinya, *tag*

merespon emisi frekuensi radio dan menurunkan dayanya dari gelombang-gelombang energi yang dipancarkan oleh *Reader*. Sebuah *tag* pasif minimum mengandung sebuah indentifier unik dari sebuah item yang dipasang *tag* tersebut. Data tambahan dimungkinkan untuk ditambahkan pada *tag*, tergantung kepada kapasitas penyimpanannya.

Harga *tag* pasif lebih murah dibandingkan harga versi lainnya. Perkembangan *tag* murah ini telah menciptakan revolusi dalam adopsi *RFID* dan memungkinkan penggunaannya dalam skala yang luas baik oleh organisasi-organisasi pemerintah dan industri.

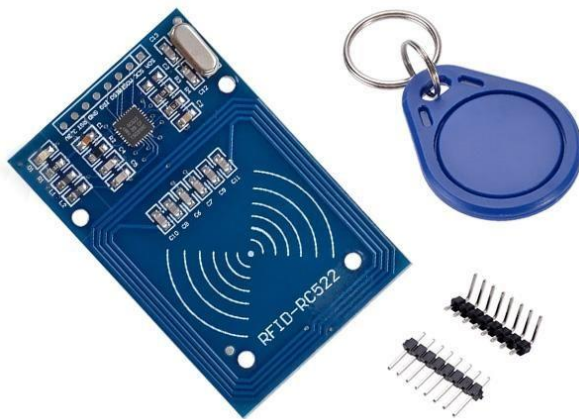
Tag semipasif adalah versi *tag* yang memiliki catu daya sendiri (baterai) tetapi tidak dapat menginisiasi komunikasi dengan *Reader*. Dalam hal ini baterai digunakan oleh *tag* sebagai catu daya untuk melakukan fungsi yang lain seperti pemantauan keadaan lingkungan dan mencatu bagian elektronik internal *tag*, serta untuk memfasilitasi penyimpanan informasi. *Tag* versi ini tidak secara aktif memancarkan sinyal ke *Reader*. *Tag* semi pasif dapat dihubungkan dengan sensor untuk menyimpan informasi untuk peralatan keamanan kontainer.

Tag aktif adalah *tag* yang selain memiliki antena dan chip juga memiliki catu daya dan pemancar serta mengirimkan sinyal kontinyu. *Tag* versi ini biasanya memiliki kemampuan baca tulis, dalam hal ini data *tag* dapat ditulis ulang dan/atau dimodifikasi. *Tag* aktif dapat menginisiasi komunikasi dan dapat berkomunikasi pada jarak yang lebih jauh, hingga 750 kaki, tergantung kepada daya baterainya. Harga *tag* ini merupakan yang paling mahal dibandingkan dengan versi lainnya.

Dalam keadaan yang sempurna, sebuah *tag* dapat dibaca dari jarak sekitar 10 hingga 20 kaki. *Tag* pasif dapat beroperasi pada frekuensi rendah (*low frequency*, LF), frekuensi tinggi (*high frequency*, HF), frekuensi ultra tinggi (*ultrahigh frequency*, UHF), atau gelombang mikro (*microwave*).

b. *Reader RFID*

RFID Reader, terdiri atas RFID Reader dan antena yang akan mempengaruhi jarak optimal identifikasi. RFID Reader akan membaca atau mengubah informasi yang tersimpan didalam Tag melalui frekuensi radio. Reader RFID tipe MRC522 digunakan dalam 13.56 MHz kartu chip komunikasi conctactless yang sangat terintegrasi untuk membaca dan menulis. RFID reader adalah merupakan alat pembaca RFID Tag.



Gambar 2.4 Bentuk Fisik Modul RFID MRC522

Sumber : <http://www.instructables.com/id/Arduino-RFID-Reader-MFRC522-Tutorial/>

2.2. Pemograman Mikrokontroller Arduino

2.2.1 Arduino UNO ATmega 328

Arduino UNO adalah salah satu produk berlabel Arduino yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroller Atmega328 (sebuah keping yang secara fungsional bertindak seperti sebuah komputer). Piranti ini dapat dimanfaatkan untuk mewujudkan rangkaian elektronik dari yang sederhana hingga yang kompleks. Pengendalian LED hingga pengontrolan robot dapat diimplementasikan dengan menggunakan papan yang berukuran relatif kecil

ini. Bahkan, dengan penambahan komponen tertentu, piranti ini bisa dipakai untuk pemantauan jarak jauh melalui internet, misalnya pemantauan kondisi pasien di rumah sakit dan pengendalian alat-alat di rumah (Kadir, 2012:16).



Gambar 2.1 Arduino UNO

Sumber : <http://www.arduino.org/products/boards/arduino-uno>

2.2.2 Mikrokontroler Atmega328

Mikrokontroler Atmega328 adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur Reduce Instruction Set Computer (RISC) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur Complex Instruction Set Computer (CISC). Mikrokontroler ATmega328 memiliki arsitektur Harvard, yakni memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan kerja dan parallelism. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat instruksi dikerjakan instruksi berikutnya sudah diambil dari memori program. (Ferdynal. 2015)

Fitur-fitur yang terdapat pada mikrokontroler Atmega328 antara lain :

A. Memiliki EEPROM (Electrically Erasable Programmable Read Only Memory) sebesar 1KB sebagai tempat penyimpanan data semi permanen karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan.

B. Memiliki SRAM (Static Random Access Memory) sebesar 2KB.

C. Memiliki pin I/O digital sebanyak 14 pin 6 diantaranya PWM (Pulse Width Modulation) output.

D. 32 x 8-bit register serba guna.

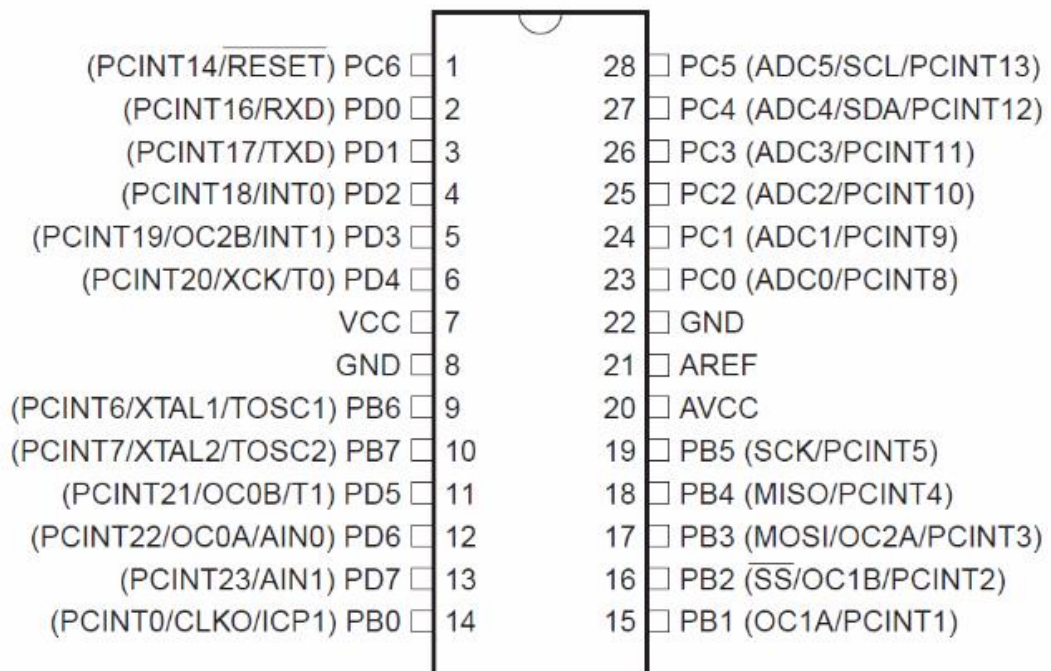
E. Dengan clock 16 MHz kecepatan mencapai 16 MIPS.

F. 32 KB Flash memory dan pada arduino memiliki bootloader yang menggunakan 2 KB dari flash memori sebagai bootloader.

G. 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus clock.

ATMega328 memiliki 3 buah PORT utama yaitu PORTB, PORTC, dan PORTD dengan total pin input/output sebanyak 23 pin. PORT tersebut dapat difungsikan sebagai input/output digital atau difungsikan sebagai peripheral lainnya.

Pinout ATMEGA328P



Gambar 2.2 Gambar Datasheet ATmega 328P

Sumber : <http://www.nerdkits.com/forum/thread/1632/>

2.2.3 Input / Output Digital dan Input Analog

Input/output digital atau digital pin adalah pin-pin untuk menghubungkan arduino dengan komponen atau rangkaian digital, contohnya, jika ingin membuat LED berkedip, LED tersebut bisa dipasang pada salah satu pin input atau output digital dan *ground* komponen lain yang menghasilkan output digital atau menerima input digital bisa disambungkan ke pin ini.

Input analog atau analog pin adalah pin-pin yang berfungsi untuk menerima sinyal dari komponen atau rangkaian analog, contohnya; potensiometer, sensor suhu, sensor cahaya, dan lain-lain.

2.2.4 Catu daya

Pin catu daya adalah pin yang memberikan tegangan untuk komponen atau rangkaian yang dihubungkan dengan arduino. Pada bagian catu daya ini pin Vinput dan Reset. Vinput digunakan untuk memberikan tegangan langsung kepada arduino tanpa melalui tegangan pada USB atau adaptor, sedangkan Reset adalah pin untuk memberikan sinyal reset melalui tombol atau rangkaian eksternal.

2.2.5 Pemrograman Integrated Development Environment (IDE)

Pemrograman board Arduino UNO dilakukan dengan menggunakan Arduino Software (IDE), dan bahasa yang digunakan adalah bahasa C. Chip ATmega 328 yang terdapat pada Arduino UNO telah diisi program awal yang sering disebut bootloader. Bootloader tersebut yang bertugas untuk memudahkan melakukan pemrograman lebih sederhana menggunakan Arduino Software, tanpa harus menggunakan tambahan hardware lain. Hanya dengan cara hubungkan Arduino dengan kabel USB ke PC, lalu jalankan software Arduino Software (IDE), dan setelah itu pemrograman menggunakan chip ATmega2560 dapat dilakukan.

2.2.6 Sketch Arduino

Program Arduino dalam bahasa C yang harus diisikan menggunakan Arduino IDE ke dalam mikrokontroler. Sketch harus memiliki dua bagian blok penting yaitu 'void setup {}' untuk menginisialisasi program dan 'void loop {}' yang berisi mengenai program utamanya.

2.3 Visual Basic 2010

2.3.1 Pengertian *Visual Basic 2010*

Hendra, ST. dalam E-booknya yang berjudul “Pemrograman Berorientasi Objek pada *Visual Basic 2010* menjelaskan tentang Visual Basic, beliau menerangkan bahwa *Visual Basic 2010* merupakan bahasa yang Object-Based (komponen-komponen program dibuat dalam bentuk objek), sedangkan *Visual Basic 2010* adalah Object Oriented, hal ini berarti bahwa *Visual Basic 2010* merupakan bahasa yang benar-benar berorientasi object dengan mendukung empat pilar utama dari OOP yaitu Polymorphism, Inheritance, Abstraction dan Encapsulation.

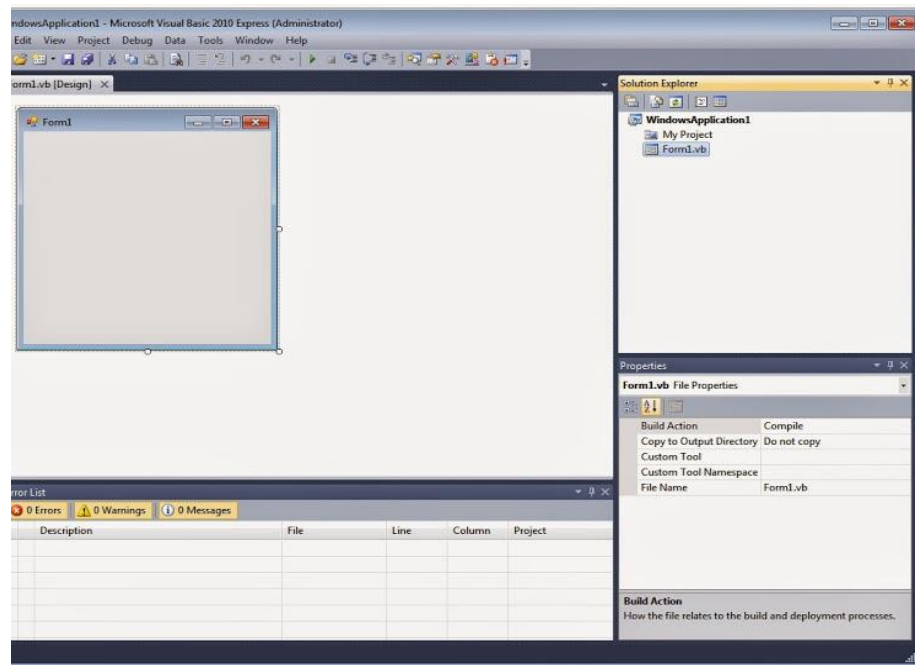
Visual Basic 2010 dapat kita jadikan alat Bantu untuk membuat berbagai macam program komputer. Aplikasi *Visual Basic 2010* hanya dapat dijalankan pada system Operasi *Windows*.

2.3.2 Aplikasi *Visual Basic 2010*

Aplikasi yang dapat dihasilkan dengan bahasa pemrograman *Visual Basic 2010* antara lain :

1. Sistem Aplikasi Bisnis
2. *Software* Aplikasi
3. Permainan (*Games*) dan Lain-lain.

Berikut adalah penjelasan mengenai fungsi dari jendela utama *Visual Basic 2010* :



Gambar 2.3 Tampilan Utama Aplikasi Visual Basic

1. *Menu Bar*
Berisi Menu-menu yang masing-masing menu memiliki fungsi tersendiri.
2. *ToolBar*
Tombol-tombol *Icon* Yang berfungsi mewakili suatu perintah yang berada pada *Menu bar*.
3. *ToolBox*
Jendela yang mengandung semua *Object* atau *control* yang dapat di tempelkan dan dibutuhkan untuk membentuk suatu *program*.
4. *Project(Solution) Explorer*
Jendela yang mengandung semua *File* yang ada didalam aplikasi yang akan kita buat :
Contoh : *Form,Module,Class,Report*, dll.
5. *Design View*
Daerah kerja utama Untuk Mendesain *program-program* Aplikasi.
6. *Code View*

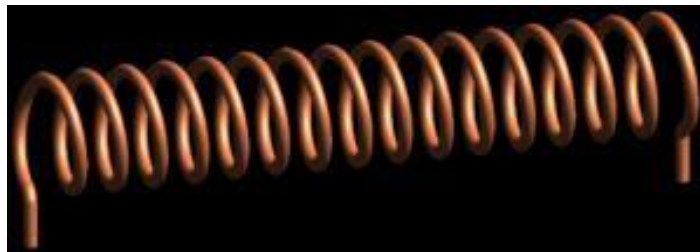
Tempat Mengetikkan baris *program* yang menjadi instruksi-instruksi.

7. *Project(Object) Properties*

Jendela yang mengandung semua informasi/Sifat dari *Object* yang terdapat pada aplikasi yang dibuat dan terseleksi.

2.4 *Electronic Door Lock (Solenoid doorlock)*

Solenoid adalah sebuah lilitan kawat tembaga yang kemudian dililitkan dengan rapat pada sebuah inti besi untuk menghasilkan medan elektromagnet. Lilitan tersebut disebut solenoida, solenoida ini merupakan medan magnet yang sangat kuat pada inti besinya, dengan asumsi bahwa panjang lilitan tersebut lebih besar dari diameter kabel atau tembaganya. Secara ideal, solenoid memiliki panjang lilitan yang tak berhingga dengan lilitan dari kabelnya yang rapat saling berhimpit satu sama lainnya. Maka akan menghasilkan medan elektromagnet yang sama dan konstan yang bersifat paralel terhadap inti besi yang menjadi sumbunya.



Gambar 2.13Kumparan Solenoid

(Sumber : Halliday, 2001)

Apabila kita alirkan listrik kepada batang besi yang kita tempatkan di tengah lilitan, maka batang besi tersebut akan mendapatkan induksi magnet dan akhirnya dapat menjadi magnet. Dengan penempatan sebagian batang besi tersebut berada di dalam solenoid dan sebagiannya lagi di sebelah luarnya. Batang besi yang terinduksi magnet tersebut akan menarik masuk benda berbahan logam ke dalam solenoid. Hal ini yang dimanfaatkan untuk menggerakkan tuas, menutup dan mengunci pintu, atau menggerakkan slot

kunci pintu. Prinsip kerja dari sebuah solenoid DC cukup mirip dengan sebuah solenoid AC, keduanya dirancang khusus dan menghasilkan medan elektromagnet.

Inti besi yang berbentuk bulat dan kerucut itu, salah satu ujungnya memiliki kutub positif. Ketika inti besi tersebut dimasukkan ke tengah kumparan yang penuh dengan medan magnet, maka permukaan ujung yang satunya lagi memiliki kutub negatif. Sementara di bagian bawahnya terdapat area yang cukup luas untuk menyalurkan aliran fluks magnet tersebut.

2.5 Relay

Transistor tidak dapat berfungsi sebagai *switch* (saklar) tegangan AC atau tegangan tinggi. Selain itu umumnya tidak digunakan sebagai *switching* untuk arus besar (>5 A). dalam hal ini, menggunakan relay sangat tepat. Relay berfungsi sebagai saklar yang bekerja berdasarkan input yang dimilikinya (Budiharto, 2005).

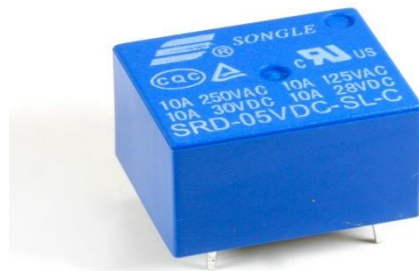
Keuntungan relay:

1. Dapat switch AC dan DC, transistor hanya switch DC.
2. Relay dapat switch tegangan tinggi, transistor tidak dapat.
3. Relay pilihan yang tepat untuk switching arus yang besar.
4. Relay dapat switch banyak kontak dalam 1 waktu.

Kekurangan relay:

1. Relay ukurannya jauh lebih besar daripada transistor.
2. Relay tidak dapat switch dengan cepat.
3. Relay butuh daya lebih besar dibanding transistor.
4. Relay membutuhkan arus input yang besar.

Transistor berdaya kecil juga kadang kala membutuhkan relay sebagai saklar tegangan tinggi. Relay akan aktif apabila ada input tegangan yang cukup pada basis transistor. Dibutuhkan diode proteksi untuk mencegah tegangan balik yang dapat merusak transistor pada gambar 2.10 menunjukkan bentuk dari relay.



Gambar 2.16Bentuk FisikRelay

(<http://www.buildcircuit.com/wp-content/uploads/2014/02/relay-2.jpg>, diakses 10 Mei 2016)

2.6 Flowchart

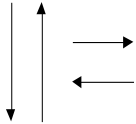

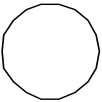
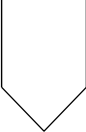

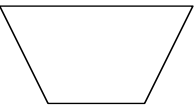
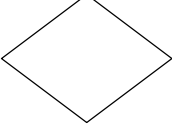
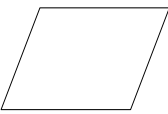
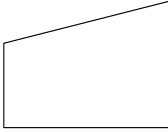
Menurut Mulyadi dalam buku Sistem Akuntansi definisi *Flowchart* yaitu : **“Flowchart adalah bagan yang menggambarkan aliran dokumen dalam suatu sistem informasi.”**

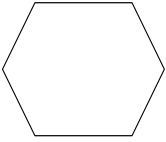

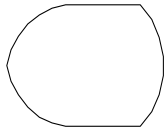
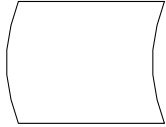
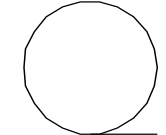
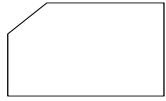
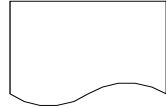
Dengan adanya flowchart urutan poses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah. Setelah flowchart selesai disusun, selanjutnya pemrogram (programmer) menerjemahkannya ke bentuk program dengan bahasa pemrograman.

2.6.1 Simbol – Simbol Flochart

Flowchart disusun dengan symbol - simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program.

Tabel 2.1 Simbol – simbol flowchart.

No	Simbol	Keterangan
1.		Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.
2.		Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan
3.		Yaitu simbol untuk keluar – masuk atau penyambungan proses dalam lembar / halaman yang sama.
4.		Yaitu simbol untuk keluar – masuk atau penyambungan proses pada lembar / halaman yang berbeda.
5.		Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer
6.		Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer
7.		Simbol pemilihan proses berdasarkan kondisi yang ada.
8.		Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
9.		Simbol untuk pemasukan data secara manual on-line keyboard

10.		Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
11.		Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
12.		Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
13.		Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
14.		Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
15.		Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
16.		Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Sumber :

https://www.academia.edu/6912277/PENGERTIAN_DASAR_DAN_SIMBOL_FLOWCHART

2.7 Bahasa C

Sejarah Perkembangan Bahasa C Berasal dari bahasa BCPL (Basic Combined Programming Language) oleh MARTIN RICHARD, Cambridge tahun 1967. KEN THOMPSON membuat bahasa B untuk dipakai pada komputer DEC

PDP -7 di bawah sistem operasi UNIX pada Bell laboratory, Murray Hill, New Jersey tahun 1970. Bahasa B merupakan suatu bahasa pemrograman yang tidak memiliki jenis suatu data seperti halnya PL/M. Berdasarkan gambaran bahasa B, DENNIS RITCHIE menulis bahasa C. Nama C diambil berdasarkan urutan sesudah B dari bahasa BCPL.

Tujuan bahasa C pada mulanya untuk membentuk suatu sistem operasi yang akan digunakan pada mesin komputer DEC PDP-11 yang baru. Pada tahun 1975, sistem operasi UNIX versi 6 dan bahasa C mulai diberikan kepada Universitas maupun Akademi. Dan pada tahun 1979, system operasi UNIX versi 7 dikeluarkan dengan bahasa C.

Sistem operasi ini (versi 7) seluruhnya ditulis dalam bahasa C. Pada 1978 Dennis Ritchie dan Brian Kernighan kemudian mempublikasikan buku *The C Programming Language* yang semakin memperluas pemakaiannya dan dijadikan standar oleh ANSI (American National Standard Institute) pada tahun 1989. C kemudian dikembangkan lagi oleh Bjarne Stroustrup menjadi C++ (1986). C dan/atau C++ banyak digunakan (sehingga menjadi „standar“) sebagai bahasa pemrograman untuk membuat sistem operasi.

2.7.1 Struktur Program C

Untuk dapat memahami bagaimana suatu program ditulis, maka struktur dari program harus dimengerti terlebih dahulu. Tiap bahasa komputer mempunyai struktur program yang berbeda. Jika struktur dari program tidak diketahui, maka akan sulit bagi pemula untuk memulai menulis suatu program dengan bahasa yang bersangkutan. Struktur dari program C terdiri dari koleksi satu / lebih fungsi-fungsi. Fungsi pertama yang harus ada di program C sudah ditentukan namanya, yaitu bernama `main()` . Suatu fungsi di dalam program C dibuka dengan kurung kurawal buka “{” dan ditutup dengan kurung kurawal tutup “}”.

Di antara kurung kurawal 7 dapat dituliskan statemen-statement program C dan pada setiap statemen diakhiri dengan tanda titik koma “;”. Berikut adalah struktur dari program C: Gambar 2.2 Struktur Umum Bahasa C Bahasa C dikatakan sebagai bahasa pemrograman terstruktur, karena strukturnya

menggunakan fungsi-fungsi sebagai program-program bagian (subroutine). Fungsi-fungsi selain fungsi utama merupakan program-program bagian. Fungsi-fungsi ini dapat ditulis setelah fungsi utama atau diletakkan pada file pustaka dan akan dipakai di suatu program, maka nama judulnya (header file) harus dilibatkan di dalam program yang menggunakan preprocessor directive `#include` .

Header file merupakan file yang berisi dengan prototype (judul, nama, sintak) dari sekumpulan fungsi-fungsi pustaka tertentu. Jadi file ini hanya berisi dengan prototipe dari fungsi-fungsi pustaka, sedangkan fungsi-fungsi pustakanya sendiri disimpan dalam file pustaka (library file dengan nama extension file -nya adalah `.LIB`).