

BAB II

TINJAUAN PUSTAKA

2.1 *Wireless IP Camera*



Gambar 2.1 *Wireless IP Camera*

(Sumber: Dokumentasi Pribadi, 2017)

IP Camera adalah CCTV (*Closed-circuit television*) kamera yang menggunakan *Internet Protokol* untuk mengirimkan data gambar dan sinyal kendali atas *Fast Ethernet link*. Dengan demikian, *IP Camera* juga sering disebut sebagai kamera jaringan. *IP Camera* yang terutama digunakan dengan cara yang sama seperti analog televisi sirkuit tertutup. Sejumlah *IP Camera* biasanya ditempatkan bersama-sama dengan perekam *video digital* (DVR) atau jaringan perekam *video* (NVR) untuk membentuk sistem pengawasan video. *IP Camera* merupakan perkembangan dari CCTV. Yang membedakannya dengan CCTV biasa adalah setiap kamera memiliki IP sendiri sehingga kita bisa memilih kamera mana yang ingin dilihat. (Rudi Arfiansyah *dkk*, 2012: 3)

Teknologi *IP Camera* ini sangat sederhana dengan menggabungkan rongga *ethernet* yang dihubungkan ke sistem jaringan, perangkat ini langsung dapat digunakan. Bahkan pada beberapa teknologi *IP Camera* sudah dimungkinkan untuk menggunakan nirkabel karena teknologi nirkabel memiliki kecepatan *bandwidth* agar dapat mengalirkan aliran tayangan video digital. (Andrew Joewono *dkk*, 2013: 27)

2.2 *Adaptor*



Gambar 2.2 *Adaptor Wireless IP Camera*

(Sumber: Dokumentasi Pribadi, 2017)

Adaptor adalah salah satu sumber tegangan DC yang sering kita jumpai dalam perangkat rumah tangga. Penggunaan *adaptor* ini diantaranya sebagai pemberi sumber tegangan *laptop* dan perangkat elektronika yang lain. *Adaptor* merupakan alat yang berfungsi untuk menurunkan tegangan AC PLN dari 220 Volt kemudian mengubahnya menjadi tegangan DC. Oleh karena itu *adaptor* sering disebut atau dinamakan dengan istilah AC DC *Adaptor*.

Dilihat dari tegangan keluarannya maka *adaptor* dapat dibedakan menjadi 2 jenis yaitu:

1. *Adaptor* Variabel
2. *Adaptor* Tegangan Tetap

1. *Adaptor* Variabel adalah *adaptor* yang memiliki tegangan *output* dapat diatur. Pada umumnya tegangan *output adaptor* variabel adalah 1.5 volt, 3 volt, 4.5 volt, 6 volt, 7.5 volt, 9 volt dan 12 volt. Pada *adaptor* variabel ini dilengkapi dengan saklar *selector* tegangan yang berfungsi untuk memilih tegangan *output* yang diinginkan. Saklar *selector* dalam *adaptor* variabel ada yang berbentuk rotari dan berbentuk geser.
2. *Adaptor* Tegangan Tetap adalah *adaptor* yang memiliki tegangan *output* permanen atau tidak dapat diatur. *Adaptor* tegangan tetap ini salah satunya adalah *adaptor laptop* dan *charger handphone*. Kedua jenis *adaptor* tersebut

memiliki tegangan *output* yang tetap dan didesain sesuai dengan kebutuhan.
(Zona Elektro, 2014)

2.3 Modem MiFi



Gambar 2.3 Modem MiFi Andromax M3Y

(Sumber: Dokumentasi Pribadi, 2017)

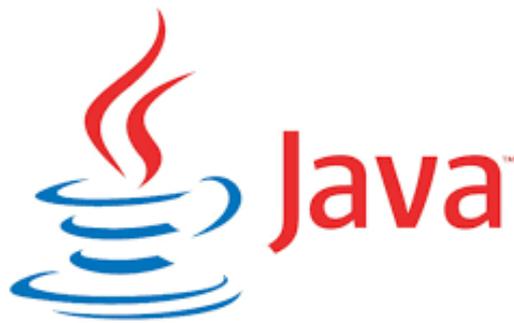
MiFi adalah perangkat *modem* internet yang bisa langsung membagikan koneksi internet kepada beberapa klien-klien yang terkoneksi secara WiFi. Perangkat *MiFi* merupakan perangkat yang sangat tepat untuk menyediakan koneksi internet kepada beberapa kliennya hanya dengan satu sambungan atau langganan ke ISP, akan tetapi *WiFi* memiliki tingkat mobilitas yang tinggi karena mudah dibawa-bawa dan sumber energi listriknya bisa menggunakan baterai.

Perangkat *MiFi* ini menggunakan koneksi nirkabel melalui jaringan seluler baik 3G GSM maupun CDMA, semua pengguna yang terhubung ke perangkat *MiFi* selain dapat mengakses Internet juga dapat saling berbagi *file* dan data. Kecepatan data yang dikirim dari perangkat *MiFi* ke komputer bisa mencapai hingga melebihi 1.5 Mbps, sementara jarak jangkauan antara perangkat *MiFi* ke komputer pengguna maksimal bisa berjarak 10 meter.

MiFi juga memungkinkan kita untuk menggunakan *Voice over IP* untuk membuat panggilan gratis melalui Internet dengan sangat murah, baik lokal maupun internasional. kita dapat melakukannya dengan menggunakan sebuah *laptop* (PC),

dengan berbagai aplikasi VoIP atau melalui ponsel yang dilengkapi dengan teknologi *WiFi*. (Mifiku, 2016)

2.4 *Java*



Gambar 2.4 Bahasa Pemrograman *Java*

(Sumber: Zamachsari.F, 2012)

Java merupakan bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya di suatu perusahaan perangkat lunak yang bernama *Sun Microsystems*, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama “*Oak*”. Namun pada tahun 1995 diganti namanya menjadi “*Java*”.

Menurut definisi *Sun Microsystems*, di dalam buku M. Shalahuddin dan Rosa A.S. (2010:1) *Java* adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri ataupun pada lingkungan jaringan. *Java* berdiri di atas sebuah mesin penterjemah (*interpreter*) yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca kode bit (*bytecode*) dalam *file class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa *Java* disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM. Alasan utama pembentukan bahasa *Java* adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat elektronik, sehingga *Java* harus bersifat tidak bergantung pada platform (*platform independent*). Itulah yang menyebabkan

dalam dunia pemrograman *Java* dikenal adanya istilah “*write once, run everywhere*”, yang berarti kode program hanya ditulis sekali, akan tetapi dapat dijalankan di bawah kumpulan pustaka (*platform*) manapun tanpa harus melakukan perubahan kode program. (Universitas Mercu Buana : 1)

Versi awal *Java* pada tahun 1996 sudah merupakan versi *release* sehingga dinamakan *Java* Versi 1.0. *Java* versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya:

1. *java.lang*: Peruntukan kelas elemen-elemen dasar.
2. *java.io*: Peruntukkan kelas *input* dan *output*, termasuk penggunaan berkas.
3. *java.util*: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas-kelas penanggalan.
4. *java.net*: Peruntukkan kelas TCP/IP yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
5. *java.awt*: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI)
6. *java.applet*: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

Contoh program *Hello World* yang ditulis menggunakan bahasa pemrograman *Java* adalah sebagai berikut:

```
// Outputs "Hello, world!" and then exits  
public class HelloWorld {  
  public static void main(String args[]) {  
    System.out.println("Hello, world!");  
  }  
}
```

2.5 XML (*Extensible Markup Language*)



Gambar 2.5 Bahasa Pemrograman XML

(Sumber: Dalan. S, 2016)

XML (*Extensible Markup Language*) merupakan bahasa web turunan dari SGML (*Standart Generalized Markup Language*) yang ada sebelumnya. XML hampir sama dengan HTML, dimana keduanya sama-sama turunan dari SGML. Teknologi XML dikembangkan mulai tahun 1966 dan mendapatkan pengakuan dari *World Wide Web Consortium* (W3C) pada bulan Februari 1998. Sedangkan SGML sendiri telah dikembangkan pada awal tahun 1980-an. Pada saat HTML dikembangkan pada tahun 1990, para penggagas XML mengadopsi bagian paling penting SGML dan dengan berpedoman pada pengembangan HTML menghasilkan bahasa *markup* yang tidak kalah hebatnya dengan SGML.

Menurut Hunter et al. (2007:3), *Extensible Markup Language* (XML) merupakan teknologi dengan aplikasi dunia nyata, khususnya untuk manajemen, tampilan, dan organisasi data. XML bekerja dengan tujuan *markup* dari setiap jenis data tetapi dengan kompleksitas yang dieliminasi. XML tidak benar-benar merupakan bahasa, tetapi lebih pada sintaks yang digunakan untuk menjelaskan *markup* lain.

Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa *meta-markup* yang menyediakan format tertentu untuk dokumen-dokumen yang mempunyai data terstruktur. Bahasa *markup* adalah mekanisme untuk mengenal secara terstruktur di dokumen. XML

adalah suatu aplikasi profil dari SGML. Seperti yang didefinisikan dari ISO 8879, SGML adalah cara standar dan *vendor-independent*.

XML tidak mempunyai definisi secara tepat karena ada yang berpendapat bahwa XML bukanlah suatu bahasa pemrograman, melainkan XML merupakan sintaks yang digunakan untuk menjelaskan bahasa markup lain sehingga dinamakan *meta-language*. Meskipun demikian pendapat yang XML bukan merupakan bahasa *markup*, didasarkan bahwa XML merupakan bahasa *markup* terpisah untuk tujuan terpisah. Selain itu XML bukanlah solusi semua hal untuk tujuan semua *user*. Sedangkan peran dari *markup* itu sendiri berupa:

1. *Markup* dapat menambah maksud arti (*semantic*) suatu data.
2. Dapat memisahkan data.
3. Dapat mendefinisikan peran data.
4. Dapat mendefinisikan batasan data.
5. Dapat mendefinisikan keterhubungan data.

XML dapat memungkinkan pertukaran informasi atau data antar *device* (*server*, PCs, *smart device*, aplikasi, dan situs web). Data ini akan menjadi *independent (unlocked)*, memudahkannya untuk diorganisir, diprogram, diubah dan ditukar antar situs web atau aplikasi apa saja. Karena kebutuhan ini, maka makin banyak teknologi berbasis XML yang keluar. Contohnya adalah SOAP (*Simple Project Acces Protocol*) dan UDDI (*Universal Description Discovery and Integration*).

Contoh program XML:

- Sebuah contoh dokumen XML seperti pada contoh dibawah ini dan simpan kedalam *file* contoh1.xml :

```
<?xml version="1.0"?>
<product barcode="2394287410">
  <manufacturer>Verbatim</manufacturer>
  <name>DataLife MF 2HD</name>
  <quantity>10</quantity>
  <size>3.5"</size>
  <color>black</color>
```

```
<description>floppy
disks</description></product>
```

- jika dokumen tersebut kita buka menggunakan web browser, maka tampak seperti gambar dibawah ni :

```
<?xml version="1.0" ?>
- <product barcode="2394287410">
  <manufacturer>Verbatim</manufacturer>
  <name>DataLife MF 2HD</name>
  <quantity>10</quantity>
  <size>3.5"</size>
  <color>black</color>
  <description>floppy disks</description>
</product>
```

- Dokumen diatas menjelaskan sebuah data produk yang ada pada suatu supermarket. Produk tersebut memiliki kode *barcode= 2394287410*, *manufacturer=Verbatim*, *name=DataLife MF 2HD*, *quantity=10*, *size=3.5"*, *color=black*, *description=floppy disks*. (Internet Programming Pens : 3)

2.6 *Android Studio*



Gambar 2.6 Perangkat Lunak *Android Studio*

(Sumber: Juansyah. A, 2015)

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi *Android* dan bersifat *open source* atau gratis. Peluncuran *Android Studio* ini diumumkan oleh *Google* pada 16 mei 2013 saat

event Google I/O Conference untuk tahun 2013. Sejak saat itu, *Android Studio* menggantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi *Android*.

Android Studio awalnya dimulai dengan versi 0.1 pada bulan Mei 2013, kemudian dibuat versi beta 0.8 yang dirilis pada bulan Juni 2014 dan yang paling stabil dirilis pada bulan Desember 2014 yang dimulai dari versi 1.0.

Sebagai pengembangan dari *Eclipse*, *Android Studio* memiliki banyak fitur-fitur baru dibandingkan dengan *Eclipse* IDE. *Android Studio* menggunakan *Gradle* sebagai *build environment* sedangkan *Eclipse* menggunakan *Ant*. (Juansyah. A, 2015: 3)

2.7 Weevcam



Gambar 2.7 Aplikasi *Weevcam* pada *Smartphone*

(Sumber: Dwi Rahma P.S, 2017)

Weevcam adalah sebuah aplikasi yang dirancang untuk produk rumah pintar (*smart home*). Nama *Weevcam* memiliki arti “kita dapat memantau *Wireless IP Camera* secara *live view*”. Sesuai dengan namanya, aplikasi ini dibuat dengan tujuan untuk memudahkan kinerja manusia agar dapat memantau kondisi ruangan tanpa harus berada ditempat memantau dan secara *real time*. Nama *Weevcam* dibuat pada tanggal 27 Maret 2017 oleh seorang mahasiswi Politeknik Negeri Sriwijaya jurusan Teknik Elektro Program Studi Teknik Telekomunikasi yang bernama Dwi Rahma PutriSari.

Aplikasi *Weevcam* dibuat dengan menggunakan sebuah perangkat lunak yang bernama *Android Studio* dengan didukung oleh bahasa pemrograman *Java* dan

XML (*Extensible Markup Language*). Pada aplikasi *Weevcam*, *Java* digunakan untuk pengkodean kerja fungsi aplikasi yang mana *Java* merupakan bahasa pemrograman yang bersifat umum atau non-spesifik (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi *Java* mampu berjalan di beberapa *platform* sistem operasi yang berbeda, *Java* dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Saat ini *Java* merupakan bahasa pemrograman yang paling populer digunakan dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

Bahasa pemrograman XML digunakan sebagai *layout* pada rancangan aplikasi *Weevcam*. XML (*Extensible Markup Language*) merupakan kelanjutan dari HTML (*HyperText Markup Language*) yang merupakan bahasa standar untuk melacak Internet.

Aplikasi *Weevcam* ini menggunakan *protocol RTSP port 554*. *Real Time Streaming Protocol (RTSP)* merupakan *protocol* jaringan komputer yang dirancang untuk digunakan dalam hiburan dan sistem komunikasi untuk mengendalikan *server* aliran media (*media streaming*). Protokol ini digunakan untuk menetapkan dan mengendalikan sesi media antara dua titik ujungnya. Klien dari *server* media mengeluarkan perintah seperti *VCR*, *play* dan *pause* untuk mendukung kendali waktu nyata dari berkas media yang dijalankan dari server. Transmisi aliran data tersebut bukan merupakan tugas protokol RTSP.

Sebagian besar *server* RTSP menggunakan *Real-time Transport Protocol (RTP)* yang saling melengkapi dengan *Real-time Control Protocol (RTCP)* untuk pengiriman aliran media. Namun beberapa penyedia menerapkan protokol pengiriman dengan hak milik pribadi, misalnya *server* RTSP dari *Real Networks* menyediakan protokol miliknya yaitu *Real Data Transport*. Walaupun dalam beberapa hal mirip dengan HTTP, RTSP dapat berguna dalam mengendalikan jalannya *multimedia*. RTSP memiliki *state* sedangkan HTTP tidak memilikinya. Sebuah pengenal digunakan jika diperlukan untuk melacak *session* yang berurutan. Sama halnya dengan HTTP, RTSP menggunakan

TCP untuk mengurus koneksi *end-to-end* dan sebagian besar pesan kendali RTSP dikirim oleh klien ke *server*.

Weevcam hanya dapat diunduh melalui *smartphone*. Dimana prinsip kerja dari aplikasi *Weevcam* ini yaitu dapat melihat gambar yang sedang ditangkap oleh *Wireless IP Camera* secara *online* dengan cara *login* aplikasi *Weevcam* dan kemudian pada awal menu aplikasi *Weevcam*, dapat kita pilih secara langsung kamera mana yang ingin kita gunakan untuk memantau yang mana kita telah mendaftarkan *IP address* dari masing-masing *Wireless IP Camera* terlebih dahulu. Aplikasi *Weevcam* ini lebih praktis karena tidak diperlukan *reset* pada saat akan menggunakan *Wireless IP Camera*, kita hanya perlu mendaftarkan alamat IP yang sedang digunakan oleh *Wireless IP Camera* tersebut. Setelah mendaftarkan *IP address* dari masing-masing *Wireless IP Camera* dan memilih *channel* kamera mana yang akan dilihat, maka secara otomatis aplikasi *Weevcam* akan menampilkan gambar yang sedang ditangkap oleh *Wireless IP Camera* secara *online* dan *real time*. (Dwi Rahma P.S, 2017)

2.8 *Android Jelly Bean* Versi 4.1.2



Gambar 2.8 *Android Jelly Bean* Versi 4.1.2

(Sumber: Dyah Ayu, 2013)

Android 4.1.2 Jelly Bean diumumkan pada 29 Oktober 2012, versi ini menawarkan peningkatan kecepatan dan kemudahan *Android* versi 4.1.1 serta mencakup semua fitur baru seperti *Photo Sphere* dan desain baru aplikasi kamera,

keyboard Gesture Typing, *Google Now* dan lainnya. *Jelly Bean* adalah sejenis permen yang juga populer disebut kacang jeli. Di bawah ini merupakan fitur-fitur yang ada pada *Android* Versi 4.1.2 *Jelly Bean*:

1. *Photo Sphere* dan *Keyboard Gesture*. *Photo Sphere* adalah salah satu fitur menakjubkan dari *Android 4.1.2 Jelly Bean*. *Photo Sphere* memungkinkan pengguna mengambil foto dari setiap arah, mirip dengan *Street View* dan *Panoramic Photo*. Selain *Photo Sphere*, *Android 4.1.2* juga mendukung fitur *keyboard gestures*. Fitur *keyboard gesture* ini sebenarnya telah hadir pada aplikasi pihak ketiga, kini fitur ini diadopsi oleh *Google* dengan berbagai penyempurnaan
2. *User Login*. Fasilitas *user login* ini adalah salah satu fitur baru yang dibawa *Android 4.1.2 Jelly Bean*. Fitur ini memungkinkan pengguna untuk *login* ke perangkat layaknya *login* di komputer. Melalui fitur *user login* ini, pengguna dapat mengkustomisasi *homescreen*, *widget* hingga aplikasi sendiri. Fitur ini juga bermanfaat untuk mengakomodasi jika pengguna *tablet* lebih dari satu orang, sehingga masing-masing memiliki kustomisasi sendiri-sendiri. Fitur *user login* ini untuk sementara hanya tersedia pada *tablet*, akan tetapi untuk kedepannya akan dikembangkan untuk *smartphone*.
3. *Wireless Display*. *Android 4.1.2* juga dilengkapi dengan sebuah fitur yang bernama *wireless display* dimana pengguna dapat menghubungkan perangkat *Android* dengan *monitor* lain atau televisi (TV) tanpa kabel. Untuk menggunakan fitur ini dibutuhkan *wireless display adaptor* untuk *monitor* atau TV yang memiliki konektor HDMI. *Google* memperluas fungsi notifikasi pada *Android 4.1.2 Jelly Bean* yang memungkinkan pengguna untuk menindaklanjuti pemberitahuan atau notifikasi yang muncul hanya dengan sekali sentuh
4. *Google Now*. *Google Now* telah hadir sejak *Android 4.1 Jelly Bean*. Namun, fitur *Google Now* ini telah diperbarui oleh *Google*. Sebagai contoh pada *Weather* kini tak hanya memberitahukan keadaan cuaca, akan tetapi dapat memberitahukan perkiraan cuaca hari ini maupun yang akan datang. Sistem

operasi *Andorid 4.1.2 Jelly Bean* telah disematkan pada ponsel *Nexus 4* dan *tablet Nexus 10* yang menjadi ancaman bagi *iOS 6*. *Android 4.1.2 Jelly Bean* telah menjadi jawaban *Google* atas *Windows Phone 8* yang telah dirilis pada hari yang sama. (Dyah Ayu, 2013)

2.9 Flowchart

2.9.1 Pengertian Flowchart

Menurut Hidayat (2014 : Vol. 4 No. 2) *Flowchart* atau Diagram Alir adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* menolong *analyst* dan *programmer* untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* adalah bentuk gambar atau diagram yang mempunyai aliran satu atau dua arah secara sekuensial. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman.

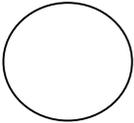
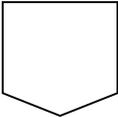
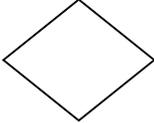
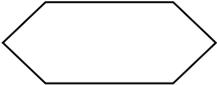
2.9.2 Pedoman Menggambar Flowchart

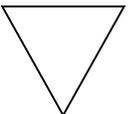
Pedoman dalam menggambar suatu *flowchart* atau bagan alir, analisis sistem atau pemrograman yaitu sebagai berikut:

- a. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman
- b. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas
- c. Harus ditunjukkan darimana kegiatan akan dimulai dan dimana akan berakhirnya.
- d. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan, misalnya “persiapkan” dokumen “hitung” gaji

- e. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya
- f. Kegiatan yang terpotong dan akan disambung ke tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
- g. Gunakanlah simbol-simbol bagan alir yang standar.

Tabel 2.1 Simbol-simbol *Flowchart*

No	Simbol	Keterangan
1		Simbol arus atau <i>flow</i> , yaitu menyatakan jalannya arus suatu proses
2		Simbol <i>connector</i> , berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
3		Simbol <i>offline connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman berbeda
4		Simbol <i>process</i> , yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
5		Simbol manual, yaitu menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
6		Simbol <i>decision</i> , yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya atau tidak
7		Simbol terminal, yaitu menyatakan permulaan atau akhir suatu program
8		Simbol <i>predefined process</i> , yaitu menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal

9		Simbol <i>keying operation</i> , menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
10		Simbol <i>offline-storage</i> , menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
11		Simbol manual <i>input</i> , memasukkan data secara manual dengan menggunakan <i>online keyboard</i>
12		Simbol <i>input/output</i> , menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya
13		Simbol <i>magnetic tape</i> , menyatakan <i>input</i> berasal dari pita magnetis atau <i>output</i> disimpan ke dalam pita magnetis
14		Simbol <i>disk storage</i> , menyatakan <i>input</i> berasal dari disk atau <i>output</i> disimpan ke dalam <i>disk</i>
15		Simbol <i>document</i> , mencetak keluaran dalam bentuk dokumen (melalui printer)
16		Simbol <i>punched card</i> , menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu