

BAB II

TINJAUAN PUSTAKA

II.1. Kecerdasan Buatan (*Artificial Intelligent*)

Kecerdasan buatan adalah suatu ilmu yang mempelajari cara membuat komputer melakukan sesuatu seperti yang dilakukan oleh manusia. Kecerdasan buatan (*artificial intelligence*) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal, dalam pandangan manusia adalah cerdas. (Kusrini; 2006: 1)

Rich and Knight (1991) mendefinisikan kecerdasan buatan sebagai sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia. (Kusrini; 2006: 1)

Kecerdasan buatan berbeda dengan program konvensional. Pemrograman konvensional berbasis pada algoritma yang mendefinisikan setiap langkah dalam penyelesaian masalah. Pemrograman konvensional dapat menggunakan rumus matematika atau prosedur sekuensial untuk menghasilkan solusi. Lain halnya dengan pemrograman dalam kecerdasan buatan yang berbasis pada representasi simbol dan manipulasi. Dalam kecerdasan buatan, sebuah simbol dapat berupa kalimat, kata, atau angka yang digunakan untuk merepresentasikan objek, proses, dan hubungannya. Objek dapat berupa manusia, benda, ide, konsep, kegiatan, atau pernyataan dari suatu fakta. Proses digunakan untuk memanipulasi simbol untuk menghasilkan saran atau pemecahan masalah. Selain itu kecerdasan buatan dapat melakukan penalaran terhadap data yang tidak komplit. Hal ini sangat mustahil

dilakukan oleh pemrograman konvensional. Kemampuan penalaran dan penjelasan terhadap setiap langkah dalam pengambilan keputusan menjadi kelebihan dari kecerdasan buatan. (Turban; 2005: 2)

II.2. Sistem Pakar

Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Dengan sistem pakar ini, orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman. (Sri, Kusumadewi; 2003:109)

Menurut Durkin: sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan oleh seorang pakar.

Menurut Ignizi: sistem pakar adalah suatu model dan prosedur yang berkaitan, dalam suatu domain tertentu, yang mana tingkat keahliannya dapat dibandingkan dengan keahlian seorang pakar.

Menurut Giarratano dan Riley: Sistem pakar adalah suatu sistem komputer yang bisa menamai atau meniru kemampuan seorang pakar.

Dari definisi diatas dapat disimpulkan bahwa sistem pakar adalah suatu sistem yang menyamai seorang pakar dari program komputer yang dapat menyelesaikan masalah. Sistem ini merupakan program komputer cerdas yang menggunakan pengetahuan inferensi untuk menyelesaikan masalah yang cukup sulit sehingga membutuhkan seorang yang ahli dalam menyelesaikannya. Sistem ini diharapkan dapat bekerja seperti halnya seorang pakar.

Seorang pakar dengan sistem pakar mempunyai banyak perbedaan. Dakin (1994) mengemukakan perbandingan kemampuan antara seorang pakar dengan sebuah sistem pakar seperti pada tabel berikut:

Tabel II.1 Perbandingan Kemampuan Seorang Pakar dengan Sistem Pakar

<i>Factor</i>	<i>Human Expert</i>	<i>Expert Sistem</i>
<i>Time Availability</i>	Hari Kerja	Setiap Saat
Geografis	Lokal/Tertentu	Dimana Saja
Keamanan	Tidak Tergantikan	Dapat Digantikan
<i>Pershable/dapat habis</i>	Ya	Tidak
Perormansi	Variable	Konsisten
Kecepatan	Variable	Konsisten
Biaya	Tinggi	Terjangkau

(Muhammad, Arhami; 2005:6)

Dari tabel di atas dapat dikembangkan penjelasan lebih lanjut tentang keunggulan sistem pakar dibanding seorang pakar, yaitu:

1. Sistem pakar bisa digunakan setiap hari menyerupai sebuah mesin sedangkan seorang pakar tidak mungkin bekerja terus menerus setiap hari tanpa beristirahat.
2. Sistem pakar merupakan suatu *software* yang dapat diperbanyak dan kemudian dibagikan ke berbagai lokasi maupun tempat yang berbeda-beda untuk digunakan, sedangkan seorang pakar hanya bekerja pada satu tempat dan pada saat yang bersamaan.
3. Suatu sistem pakar dapat diberi pengamanan untuk menentukan siapa saja yang mempunyai hak akses untuk menggunakannya dan jawaban yang diberikan oleh sistem yang terbebas dari proses ancaman, sedangkan seorang pakar bisa saja mendapatkan ancaman atau tekanan pada saat menyelesaikan permasalahan.
4. Pengetahuan (*knowledge*) yang disimpan pada sistem pakar tidak akan bisa hilang/lupa, yang dalam hal ini tentu harus didukung oleh *maintenance* yang baik, sedangkan pengetahuan seorang pakar manusia lambat laun akan hilang karena meninggal, usia yang semakin tua, maupun menderita suatu penyakit.
5. Kemampuan memecahkan masalah pada suatu sistem pakar tidak dipengaruhi oleh faktor dari luar seperti intimidasi, perasaan kejiwaan, faktor ekonomi ataupun perasaan tidak suka. Akan tetapi sebaliknya dengan seorang pakar yang dapat dipengaruhi oleh faktor-faktor luar seperti yang disebutkan di atas ketika sedang menyelesaikan atau memecahkan suatu masalah, sehingga dapat

memunculkan jawaban yang berbeda-beda atas pertanyaan yang diajukan walaupun masalahnya sama.

6. Umumnya kecepatan dalam memecahkan masalah pada suatu sistem pakar *relative* lebih cepat dibandingkan oleh seorang pakar manusia. Hal ini sudah dibuktikan pada beberapa sistem pakar yang terkenal didunia.
7. Biaya menggaji seorang pakar lebih mahal bila dibandingkan dengan penggunaan program sistem pakar (dengan asumsi bahwa program sistem pakar itu sudah ada).

II.2.1. Ciri-ciri Sistem Pakar

Berikut ini adalah ciri-ciri sistem pakar yaitu:

1. Memiliki informasi yang handal.
2. Mudah di modifikasi.
3. Dapat digunakan dalam berbagai jenis komputer.
4. Memiliki kemampuan untuk belajar beradaptasi.

II.2.2. Keuntungan Sistem Pakar

Berikut ini adalah keuntungan sistem pakar yaitu:

1. Membuat seorang yang awam dapat bekerja seperti layaknya seorang pakar.
2. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.

3. Meningkatkan *output* dan produktivitas. Sistem pakar dapat bekerja lebih cepat dari manusia. Keuntungan ini berarti mengurangi jumlah pekerja yang dibutuhkan, dan akhirnya akan mereduksi biaya.
4. Meningkatkan kualitas.
5. Sistem pakar menyediakan nasehat yang konsisten dan dapat mengurangi tingkat kesalahan.
6. Membuat peralatan yang kompleks lebih mudah dioperasikan karena sistem pakar dapat melatih pekerja yang tidak berpengalaman.
7. Handal (*reliability*).
8. Sistem pakar tidak dapat lelah atau bosan juga konsisten dalam memberi jawaban dan selalu memberikan perhatian penuh.
9. Memiliki kemampuan untuk memecahkan masalah yang kompleks.
(Kusrini; 2006: 5)

II.2.3. Kelemahan Sistem Pakar

Di samping memiliki beberapa keuntungan, sistem pakar juga memiliki beberapa kelemahan, antara lain :

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
2. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar di bidangnya.
3. Sistem pakar tidak 100% bernilai benar. (Kusrini; 2006: 5)

II.2.4. Alasan pengembangan Sistem Pakar

Ada beberapa alasan mendasar mengapa sistem pakar dikembangkan untuk menggantikan seorang pakar, diantaranya:

1. Dapat menyediakan kepakaran setiap waktu dan diberbagai lokasi.
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar pensiun atau pergi.
4. Seorang pakar adalah mahal.
5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat.

II.3. Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar. Sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar (*non expert*) guna memperoleh pengetahuan pakar.

Komponen-komponen yang terdapat dalam arsitektur/struktur sistem pakar :

1. Antarmuka Pengguna (*User Interface*)
2. Basis Pengetahuan (*Knowledge Base*)
3. Akuisisi Pengetahuan (*Knowledge Acquisition*)
4. Mesin Inferensi

5. *Workplace*
6. Fasilitas Penjelasan
7. Perbaikan Pengetahuan

II.3.1. Antarmuka Pengguna (*user interface*)

User interface Merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

II.3.2. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Pengetahuan itu dapat berasal dari ahli, buku, basis data, penelitian dan gambar.

Ada 2 bentuk pendekatan basis pengetahuan yaitu:

- a. Penalaran berbasis aturan (*rule-based reasoning*)

Pada penalaran berbasis aturan, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk *IF-THEN*. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan seorang pakar dapat menyelesaikan masalah tersebut secara berurutan. Disamping itu, bentuk ini juga digunakan

apabila dibutuhkan penjelasan tentang jejak (langkah-langkah) pencapaian solusi.

b. Penalaran berbasis kasus (*case-based reasoning*)

Penalaran berbasis kasus, basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (fakta yang ada). Bentuk ini digunakan apabila *user* menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama (mirip). Selain itu bentuk ini juga digunakan bila kita telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan.

II.3.3. Akuisisi Pengetahuan (*Knowledge Acquisition*)

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahua. Pengetahuan yang diakuisisi adalah pengetahuan procedural (apa yang harus dilakukan, berupa aturan, prosedur, metode, dan lain-lain) serta pengetahuan deklaratif (termasuk dan tidak termasuk, berupa fakta, konsep, dan lain-lain). Pengetahuan diperoleh dari pakar, dilengkapi dengan buku, basis data, laporan penelitian dan pengalaman pemakai. Ada beberapa tantangan dalam melakukan akuisisi, yaitu pengetahuan yang tidak lengkap, pengetahuan yang salah, kemampuan menjelaskan pengetahuan dan pandangan yang berbeda dari beberapa pakar.

Metode akuisisi pengetahuan terdiri dari:

- a. Wawancara: metode yang paling banyak digunakan, yang melibatkan pembicaraan dengan pakar secara langsung dalam suatu wawancara
- b. Analisis protokol: dalam metode ini pakar diminta untuk melakukan pekerjaan dan mengungkapkan proses pemikirannya dengan menggunakan kata-kata. Pekerjaan tersebut direkam, dituliskan, dan dianalisis.

II.3.4. Mesin/Motor Inferensi (*Inference Engine*)

Mesin Inferensi (*Inference Engine*), merupakan otak dari Sistem Pakar juga dikenal sebagai penerjemah aturan (*rule interpreter*). Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan.

Kerja mesin inferensi meliputi:

- a. Menentukan aturan mana akan dipakai.
- b. Menyajikan pertanyaan kepada pemakai, ketika diperlukan.
- c. Menambahkan jawaban ke dalam memori Sistem Pakar.
- d. Menyimpulkan fakta baru dari sebuah aturan.

e. Menambahkan fakta tadi ke dalam memori.

Ada 2 cara dalam melakukan inferensi yaitu:

- a. *Forward Chaining*: pencocokkan fakta atau pernyataan dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis. Metode inferensi cocok digunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognosis*).

- b. *Backward Chaining*: pencocokkan fakta atau pernyataan dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan.

II.3.5. Workplace / Blackboard

Workplace merupakan area dari sekumpulan memori kerja (*working memory*), digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara.

II.3.6. Fasilitas Penjelasan (*Explanation Facility*)

Kemampuan untuk menjejak (*tracing*) bagaimana suatu kesimpulan diambil merupakan hal yang sangat penting untuk transfer pengetahuan dan pemecahan masalah. Fasilitas penjelasan merupakan komponen tambahan yang akan meningkatkan kemampuan sistem pakar.

II.3.7. Perbaikan Pengetahuan

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya dan juga mengevaluasi apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan di masa mendatang. (Sri, Kusumadewi; 2003: 116).

II.4. Penyakit Osteoporosis

Penyakit Osteoporosis adalah penyakit pengkeroposan tulang. Osteoporosis ditandai oleh dua hal, yaitu pertama *densitas* (kepadatan) tulang berkurang, dan kedua kualitas tulang juga menurun. *Densitas* tulang adalah kepadatan tulang, yaitu beberapa gram mineral per volume tulang. Sedangkan kualitas tulang menyangkut arsitektur, penghancuran, dan pembentukan kembali (mineralisasi) tulang.

Tulang yang normal itu kuat, karena mengandung protein, kolagen, dan kalsium. Namun, bila tulang sudah keropos trauma sedikit saja sudah menimbulkan fraktur. Bentuk fraktur bisa berupa patah jadi dua bagian seperti pada lengan bawah, retak pada tulang panggul, atau kompresi pada tulang belakang.

Osteoporosis dapat dibagi dalam dua kelompok, yaitu:

1. Osteoporosis primer yang terdapat pada wanita pasca-menopause (*postmenopause osteoporosis*) dan pada pria atau wanita yang berusia lanjut (*senile osteoporosis*)
2. Osteoporosis sekunder yang disebabkan oleh penyakit atau kelainan tertentu, atau bisa pula akibat tindakan pembedahan atau pemberian obat yang mempercepat pengeroposan tulang. Contohnya adalah penyakit *cushing* (kelainan hormone adrenal).

II.5. Microsoft Visual Basic 6.0

Microsoft Visual Basic 6.0 merupakan salah satu aplikasi pemrograman visual yang memiliki bahasa pemrograman yang cukup populer dan mudah untuk dipelajari. Dengan *Visual Basic*, kita bisa membuat program yang memungkinkan pengguna komputer berkomunikasi dengan komputer tersebut. *Microsoft Visual Basic 6.0* menyediakan berbagai perangkat kontrol yang dapat digunakan untuk membuat program aplikasi sederhana hingga ke aplikasi pengolahan *database*. (Adi, Kurniadi; 2003: 3)

II.6. SQL Server 2005

Pada dasarnya pengertian dari *SQL Server* itu sendiri adalah bahasa yang dipergunakan untuk mengakses data dalam basis data *relation*. Bahasa ini secara *defacto* adalah bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa

ini dalam manajemen datanya. *SQL server* 2005 merupakan salah satu produk dari *Relational Database Management System* (RDBMS).

Pada *SQL Server* 2005, Anda dapat membuat, mengubah dan menghapus *Trigger* lewat menu atau pilihan-pilihan dalam *SQL Server Manajement Studio* atau lewat *query* (dahulu *Query Analyzer*). (M, Agus, J. Alam; 2005: 120)

II.7. *Data Flow Diagram* (DFD)

Data Flow Diagram adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

Data Flow Diagram sering digunakan untuk menggambarkan suatu sistem baru yang akan dikembangkan secara logika mempertimbangkan lingkungan fisik dimana data tersebut mengalir dan akan disimpan.

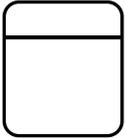
Data Flow Diagram menggambarkan komponen-komponen sebuah sistem, aliran-aliran data diantara komponen-komponen tersebut terdapat asal tujuan serta penyimpanan data.

Beberapa tipe *data flow diagram* antara lain :

1. *Data Flow Diagram* Konteks
2. *Data Flow Diagram* Fisik
3. *Data Flow Diagram* Logis

Beberapa simbol yang digunakan DFD dapat dilihat pada tabel II.2.

Tabel II.2 Simbol-Simbol *Data Flow Diagram (DFD)*

Simbol	Nama	Arti
	Entity	Entity adalah kesatuan luar yang dapat berupa orang, organisasi, objek, atau sistem lain yang berhubungan.
	Proses	Proses merupakan suatu yang dapat mengubah laporan menjadi output.
	Data Store	Data Store merupakan tempat penyimpanan data.
	Arus Data	Arus Data berupa masukan dan keluaran.

(Sumber: Jogianto, H.M: 2005; 701)

Data Flow Diagram terdiri dari :

1. *Context Diagram*

Context Diagram berfungsi memetakan model lingkungan (menggambarkan antara entitas luar, masukan dan keluaran sistem), yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.

2. *Diagram Rinci (DFD levelled)*

Dalam *DFD levelled* akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut

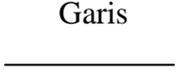
kedalam spesifikasi proses yang jelas. Jadi dalam DFD *levelled* bisa dimulai dari DFD level 0 kemudian turun ke DFD level 1 dan seterusnya.

II.8. *Entity Relationship Diagram (ERD)*

ERD digunakan untuk memodelkan struktur data dan hubungan antara data karena hal ini relative kompleks. Dengan *ERD* kita dapat menguji model dengan mengabaikan proses yang harus dilakukan. *ERD* menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Setiap *ERD* bisa terdapat lebih dari satu atribut. Isi atribut mempunyai sesuatu yang dapat mengidentifikasi isi *entity* satu dengan yang lain. Atribut mewakili oleh simbol *ellips* sebagai salah satu cara menggambarkan atribut.

Berikut ini macam-macam simbol *ERD* yang dapat dilihat pada gambar II.3.

Tabel II.3 Simbol-Simbol *Entity Relationship Diagram (ERD)*

<i>Simbol</i>	<i>Keterangan</i>
1 dan 1, 1 dan N, N dan N	Relasi dapat dinyatakan dengan pemakaian angka (1 & 1 untuk relasi satu-ke-satu, 1 & N untuk relasi satu-ke-banyak atau N & N untuk relasi banyak-ke-banyak).
	Menyatakan himpunan entitas.
	Menyatakan himpunan relasi.
	Menyatakan atribut yang berfungsi sebagai <i>Key</i> .

	Sebagai penghubung antara himpunan relasi.
---	--

(Sumber: Edhi, Sutanta; 2004: 80)

II.9. *Flowchart*

Flowchart sering juga disebut dengan bagan alir. *Flowchart* atau bagan alir adalah sebagai suatu skema yang menggambarkan urutan kegiatan dari suatu program dari awal sampai akhir.

Flowchart merupakan bagian yang menunjang arus pekerjaan secara keseluruhan dari sistem sedangkan menurut Jogiyanto, H.M (2005:793), *Flowchart* adalah bagan (*chart*) yang menunjukkan aliran (*flow*) didalam program atau prosedur sistem secara logika. Bagan alir ini terutama digunakan untuk alat bantu komunikasi dan untuk dokumentasi. Pada waktu akan menggambar suatu bagan alir, analis sistem atau pemrogram dapat mengikuti pedoman-pedoman sebagai berikut :

1. Bagan alir sebaiknya digambarkan dari atas kebawah dan mulai dari bagian kiri.
2. Kegiatan didalam bagan alir harus ditunjukkan dengan jelas.
3. Harus jelas kegiatan akan dimulai dan akan berakhirnya.
4. Kegiatan bagan alir sebaiknya digunakan suatu kata yang mewakili pekerjaan persiapan, dokumen, hitung inventaris peralatan kantor.
5. Setiap kegiatan didalam bagan alir harus didalam urutan yang semestinya.

6. Kegiatan yang terpotong dan akan disambung ditempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakan simbol-simbol bagan alir yang standart.

Ada lima macam bagan alir, yang mana penjelasannya sebagai berikut :

1. Bagan Alir Sistem (*System Flowchart*)

Bagan Alir Sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada didalam sistem.

2. Bagan Alir Dokumen (*Document Flowchart*)

Bagan Alir Dokumen atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusannya.

3. Bagan Alir Skematik (*Schematic Flowchart*)

Bagan Alir Skematik merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem. Juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan.

4. Bagan Alir Program (*Program Flowchart*)

Bagan Alir Program merupakan bagan alir logika program (*Program Logic Flowchart*) dan bagan alir program terinci (*detailed computer program flowchart*) bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah didalam program komputer secara logika. Bagan alir logika ini

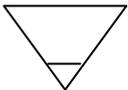
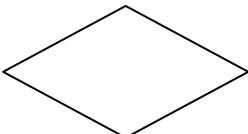
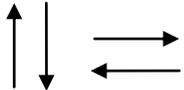
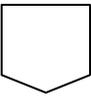
dipersiapkan oleh analis sistem. Bagan alir program terinci (*detailed computer program flowchart*) digunakan untuk menggambarkan instruksi-instruksi program komputer secara terinci.

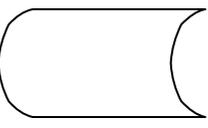
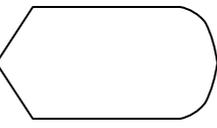
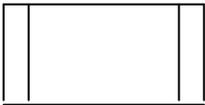
5 Bagan Alir Proses (*Process Flowchart*)

Bagan Alir Proses merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini berguna bagi analis sistem yang menggambarkan dalam suatu prosedur. Bagan alir proses selain dapat menunjukkan kegiatan dan simpanan yang digunakan dalam suatu prosedur, dapat juga menunjukkan jarak kegiatan yang satu dengan yang lainnya serta waktu yang diperlukan oleh suatu kegiatan (Jogiyanto, H.M; 2005: 794).

Ada beberapa simbol *Flowchart* antara lain dapat dilihat pada gambar II.4.

II.4 Simbol-Simbol *Flowchart*

<i>Simbol</i>	<i>Keterangan</i>	<i>Simbol</i>	<i>Keterangan</i>
	Simbol Awal/Akhir		Simbol <i>input</i> dari kartu/ <i>output</i> ditulis ke kartu
	Simbol <i>Output/Input</i>		Simbol penyimpanan <i>file</i> secara tetap
	Simbol pengolahan yang dilakukan oleh komputer		<i>File Storage offline</i> (Arsip)
	Simbol Kondisi/Keputusan		Tanda sambung pada halaman yang sama
	Simbol Aliran Proses		Tanda sambung pada halaman yang berbeda

	Simbol pengolahan yang tidak dilakukan oleh komputer		Simbol input dari <i>disk</i> atau <i>output</i> di simpan ke disk
	Menyatakan teks dokumen kerja		Simbol output yang digunakan yaitu layar, printer, dsb.
	Proses apa saja yang tidak terdefinisi		Multi Dokumen

(Sumber: Jogiyanto, H.M; 2005:796)

17,21-27