

## BAB II

### TINJAUAN PUSTAKA

#### Penelitian Terdahulu

Penelitian Terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dalam hal ini peneliti mengambil skripsi sebelumnya sebagai penelitian terdahulu yang relevan :

Tabel 2.1 Penelitian Terdahulu

<b>Peneliti Judul / Jurnal</b>	<b>Rumusan Masalah</b>	<b>Solusi</b>	<b>Hasil</b>	<b>Pembeda</b>
Mendeteksi Hama Tanaman Mangga Dengan Metode Certainty Factor Penulis Gatot Fitriyanto, Nur Ahmad Azhar dan Muhammad Kurniawan Tahun 2016	Merancang aplikasi untuk mendeteksi hama dan penyakit pada tanaman mangga dengan metode certainty factor	Membuat Aplikasi Desktop berbasis java	Hasil uji coba sistem nilai certainty factor sangat cocok dengan perhitungan manual	Menggunakan bahasa pemograman java dan menggunakan metode Certainty Factor
Sistem Pakar Untuk Mendiagnosa Hama dan	Membangun atau merancang aplikasi untuk mendeteksi	Membuat Aplikasi Desktop	Memberikan solusi untuk membantu diagnosa hama	Mendiagnosa dengan metode perhitungan

Penyakit Tanaman Bawang Merah Menggunakan Certainty Factor Tuswanto dan Abdul Fadlil Tahun 2013	hama dan penyakit pada tanaman bawang merah dengan metode certainty factor		dan penyakit pada tanaman bawang merah dengan jumlah hama 6 dan 16 penyakit yang sudah di verifikasi ahli pakar	certainty factor
Sistem Pakar Identifikasi Penyakit dan Hama Tumbuhan Teh dengan Metode Forward Chaining Berbasis Android Ian Muhlisin Tahun 2015	Bagaimana merancang dan membangun sebuah sistem yang mampu mengidentifikasi hama dan penyakit tumbuhan teh berbasis Android	Membuat aplikasi android	Aplikasi sistem pakar identifikasi penyakit dan hama tumbuhan teh dibuat berbasis mobile, sehingga dapat diakses dengan cepat dan mudah.	Bahasa pemrograman menggunakan mobile
Sistem Pakar Pengidentifikasi Hama Pada Tanaman Anggrek Terrestrial Berbasis Mobile	Merancang sistem pakar untuk mengidentifikasi hama pada tanaman anggrek terrestrial	Membuat Aplikasi Android	Aplikasi sistem pakar ini memberikan pengetahuan berbagai jenis hama beserta gejala yang ditimbulkannya	Bahasa pemrograman menggunakan mobile

Satia Suhada dan Taufik Hidayatulloh	dengan basis mobile			
Sistem Pakar Hama dan Penyakit Pada Tanaman Jeruk Manis di Kabupaten Karo Tanti Kristanti dan Theopilus Sitepu Tahun 2013	Membangun website untuk mendiagnosa hama dan penyakit pada tanaman jeruk manis	Membuat website	Berhasil diimplementasikan dan dapat memberikan informasi tentang penyebaran berbagai hama dan penyakit tanaman jeruk manis	Mendiagnosa pada tanaman jeruk manis

#### A. Tanaman Mangga

Tanaman mangga (*Mangifera indica L.*) berasal dari India, Srilanka, dan Pankistan. Mangga asli Indonesia yang kemungkinan berasal dari Kalimantan adalah kebembem / kweni (*Mangifera Odorata*). Tanaman ini merupakan buah tropis yang bisa tumbuh baik di daerah beriklim kering. Sentra produksi mangga di Indonesia diantaranya adalah Indramayu, Cirebon, dan Majalengka di Jawa Barat, Tegal, Kudus, Pati, Magelang, dan Boyolali di Jawa Tengah, Pasuruan, Probolinggo, Nganjuk, dan Pamekasan di Jawa Timur, juga di daerah Istimewa Yogyakarta, Sumatera Utara, Sumatera Barat, Sulawesi Selatan, Maluku, Nusa Tenggara Barat dan Nusa Tenggara Timur.

Jenis yang telah dilepas oleh Menteri Pertanian diantaranya adalah Arumanis 143, Golek 31, dan Manalagi 69. Ketiga jenis mangga tersebut mampu banyak menghasilkan buah, daging buahnya tebal dan rasanya manis. Mangga gadung, gedong, dan durih termasuk jenis yang populer di masyarakat, mempunyai mutu tinggi, daging tebal dan rasanya manis.

Kondisi lingkungan yang ideal bagi tanaman mangga adalah iklim yang agak kering dengan curah hujan 750 – 2.000 mm, dengan 4 – 7 bulan kering, ketinggian < 300 m dpl, dan suhu udara rata – rata berkisar antara 25°C - 32°C. Namun, mangga dapat juga ditanaman pada ketinggian hingga 1.200 m.

Di daerah beriklim basah dengan musim kering <3 bulan pertumbuhannya subur, tetapi buahnya lebih sedikit dibandingkan dengan di daerah beriklim kering. Tanaman akan mudah terserang penyakit blendok dan mati pucuk serta rasa buahnya agak asam.

Untuk mempercepat pertumbuhan tanaman muda dan mengganti hara yang terangkut panen dan pemulihan pohon tanaman yang sudah berproduksi diperlukan pemupukan. Tanaman mangga muda lebih banyak membutuhkan pupuk P dengan perbandingan pupuk Urea, SP-36 dan KCI adalah 4:12:3 sedangkan jumlah pupuk untuk tanaman berproduksi berkisar antara 3 – 8% dari bobot buah yang dihasilkan, diberikan dua kali setahun, masing – masing menjelang musim kemarau dan awal musim hujan.

Pupuk organik, berupa pupuk kandang atau kompos dibutuhkan pada saat tanam sebanyak 20 kg untuk setiap lubang tanam.

## **B. Sistem Pakar**

Sistem pakar adalah sistem informasi berbasis komputer yang menggunakan pengetahuan pakar untuk mencapai performa keputusan tingkat tinggi dalam domain persoalan yang sempit (Turban, Aronson, & Liang, 2015).

Sedangkan Pakar adalah orang yang memiliki pengetahuan, penilaian, pengalaman dan metode khusus, serta kemampuan untuk menerapkan bakat ini dalam memberi nasihat dan memecahkan masalah adalah tugas pakar untuk menyediakan pengetahuan tentang bagaimana melaksanakan suatu tugas yang akan dijalankan oleh sistem berbasis-pengetahuan. Pakar mengetahui fakta mana yang penting dan memahami arti hubungan diantaranya. Misalnya dalam mendiagnosis persoalan sistem listrik mobil, pakar mekanik mengetahui bahwa pengikat kipas dapat putus dan menyebabkan baterai discharge.

Keahlian adalah pengetahuan ekstensif yang spesifik terhadap tugas yang dimiliki pakar. Tingkat keahlian menentukan performa keputusan. Keahlian sering

dicapai dari pelatihan, membaca, dan mempraktikkan. Keahlian mencakup pengetahuan eksplisit, misalnya teori yang dipelajari dari buku teks atau kelas, dan pengetahuan implisit yang diperoleh dari pengalaman. (Sumber: Efraim Turban, Jay E. Aronson, Ting Peng Liang, 2005, 714-715).

#### 1. Manfaat Sistem Pakar Dan Kemampuan Sistem Pakar

Menurut (Efraim Turban, Jay E. Aronson, Ting Peng Liang, 2005, 730-732)

ada beberapa manfaat serta kemampuan sistem pakar, antara lain :

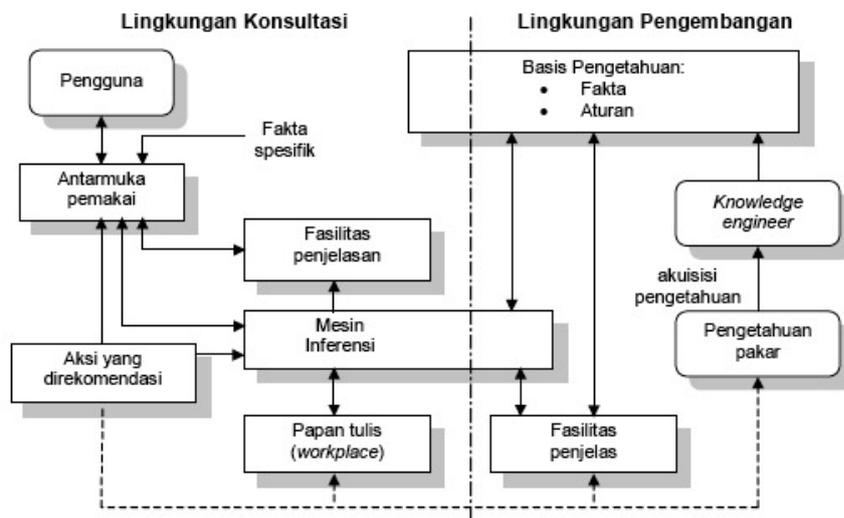
1. Meningkatkan output dan produktivitas.
2. Menurunkan waktu pengambilan keputusan.
3. Meningkatkan kualitas proses dan produk.
4. Mengurangi *downtime*.
5. Menyerap keahlian langka.
6. Fleksibilitas.
7. Operasi peralatan yang lebih mudah.
8. Eliminasi kebutuhan peralatan yang mahal.
9. Operasi dilingkungan berbahaya.
10. Aksesibilitas ke pengetahuan *help desk*.

#### 2. Keterbatasan Sistem Pakar

Metodologi ES yang tersedia mungkin tidak langsung dan efektif, bahkan untuk banyak aplikasi dalam kategori umum. Persoalan-persoalan berikut telah memperlambat penyebaran komersial ES :

1. Pengetahuan tidak selalu siap sedia.
2. Akan sulit mengekstrak keahlian dari manusia.
3. Pendekatan tiap pakar pada suatu penilaian situasi mungkin berbeda tetapi benar.
4. Sulit, bahkan bagi pakar kemampuan tinggi, untuk mengikhtisarkan penilaian situasi yang baik pada saat berada dalam tekanan waktu.
5. Pengguna sistem pakar memiliki batasan kognitif alami.

### 3. Struktur Sistem Pakar



**Gambar 2.1** Struktur Sistem Pakar

(Sri Kusumadewi, 2013, 114)

Sistem pakar dapat ditampilkan dengan dua lingkungan yaitu; lingkungan pengembangan dan lingkungan konsultasi. Lingkungan pengembangan digunakan oleh ESbuilder untuk membangun komponen dan memasukkan pengetahuan kedalam basis pengetahuan. Lingkungan konsultasi digunakan non-pakar. Lingkungan konsultasi digunakan non-pakar untuk memperoleh pengetahuan dan nasihat pakar. Lingkungan ini dapat dipisahkan setelah sistem lengkap. Komponen-komponen yang ada pada sistem pakar :

a. Subsistem akuisisi pengetahuan

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian pemecahan masalah dari pakar atau sumber pengetahuan terdokumentasi ke program komputer, untuk membangun atau memperluas basis pengetahuan.

b. Basis pengetahuan

Basis pengetahuan berisi pengetahuan relevan yang diperlukan untuk memahami, merumuskan, dan memecahkan persoalan. Basis tersebut mencakup dua elemen dasar yaitu :

- 1) Fakta, misalnya situasi persoalan dan teori area persoalan
- 2) Heuristik atau aturan khusus yang mengarahkan penggunaan pengetahuan untuk memecahkan persoalan khusus

c. Mesin inferensi

“Otak” ES adalah mesin inferensi, yang dikenal juga sebagai struktur kontrol atau penterjemah aturan (dalam ES berbasis aturan). Komponen ini sebenarnya adalah program komputer yang menyediakan metodologi untuk mempertimbangkan informasi dalam basis pengetahuan dan blackboard, dan merumuskan kesimpulan.

d. Antarmuka pengguna

Sistem pakar berisi prosesor bahasa untuk komunikasi berorientasi-persoalan yang mudah antara pengguna dan komputer. Komunikasi ini paling baik dilakukan dalam bahasa alami. Dikarenakan batasan teknologi, maka kebanyakan sistem yang ada menggunakan pendekatan pertanyaan dan jawaban untuk berinteraksi dengan pengguna. Acapkali ditambahi dengan menu, formulir elektronik, dan grafik.

e. *Blackboard* (tempat kerja)

Blackboard adalah area kerja memori yang tersimpan sebagai database untuk deskripsi persoalan terbaru yang ditetapkan oleh data input, digunakan juga untuk perekaman hipotesis dan keputusan sementara. Tiga tipe keputusan dapat direkam dalam blackboard: rencana (bagaimana mengatasi persoalan), agenda (tindakan potensial sebelum eksekusi), dan solusi (hipotesis kandidat dan arah tindakan alternatif yang telah dihasilkan sistem sampai dengan saat ini).

f. Subsistem penjelasan (*justifier*)

Kemampuan untuk melacak tanggung jawab suatu kesimpulan terhadap sumbernya adalah penting untuk transfer keahlian dan dalam pemecahan masalah. Subsistem penjelasan(disebut juga justifier) dapat melacak tanggung jawab tersebut dan menjelaskan perilaku ES dengan menjawab pertanyaan berikut secara interaktif :

g. Sistem perbaikan pengetahuan

Pakar manusia memiliki sistem perbaikan-pengetahuan,yakni mereka dapat menganalisis pengetahuannya sendiri dan kegunaannya, belajar darinya, dan meningkatkannya untuk konsultasi mendatang. Serupa pula, evaluasi tersebut diperlukan dalam pembelajaran komputer sehingga program dapat menganalisis alasan keberhasilan atau kegagalannya. Hal ini dapat mengarah kepada peningkatan sehingga menghasilkan basis pengetahuan yang lebih akurat dan pertimbangan yang lebih *efektif*. Komponen tersebut tidak tersedia dalam sistem pakar komersial saat ini, tetapi sedang dikembangkan dalam *ESexperimental* pada beberapa universitas dan lembaga riset. (Sumber: Efraim Turban, Jay E. Aronson, Ting Peng Liang, 2005, 722-724).

4. Cara Kerja Sistem Pakar

Diantara komponen-komponen dalam Gambar 2.1 di atas, basis pengetahuan dan mesin inferensi adalah modul paling kritis agar sistem pakar dapat berfungsi dengan baik. Pengetahuan harus direpresentasikan dan diatur secara tepat dalam basis pengetahuan. Mesin inferensi kemudian dapat menggunakan pengetahuan tersebut untuk menarik kesimpulan baru dari fakta dan aturan yang ada. Dalam bagian ini, struktur berbasis pengetahuan dan mesin inferensi pada sistem berbasis-aturan.

## 1. Representasi dan Organisasi Pengetahuan

Pengetahuan pakar harus direpresentasikan dalam format yang dapat dipahami komputer dan diatur dengan tepat dalam basis pengetahuan sistem pakar. Terdapat beberapa cara yang berbeda untuk merepresentasikan pengetahuan manusia, antara lain aturan produksi, jaringan semantik, dan pernyataan logika.

Dalam sistem berbasis aturan, pengetahuan dalam basis pengetahuan direpresentasikan dalam aturan JIKA MAKA yang menggabungkan kondisi dan kesimpulan untuk menangani situasi tertentu.

Bagian JIKA mengindikasikan kondisi aturan tersebut diaktifkan, dan bagian MAKA menunjukkan tindakan atau kesimpulan jika semua kondisi JIKA dipenuhi.

Keuntungan menggunakan aturan produksi adalah aturan tersebut mudah dipahami dan aturan baru dapat ditambahkan dengan mudah ke dalam basis pengetahuan tanpa memengaruhi aturan yang telah ada. Ketidakpastian yang dihubungkan dengan tiap aturan dapat ditambahkan untuk meningkatkan keakuratannya.

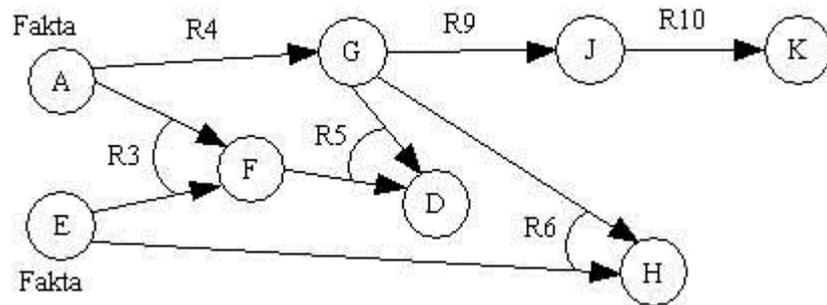
Tugas utama pengembangan sistem pakar adalah memperoleh pengetahuan dari manusia dan mengubahnya menjadi aturan produksi yang dapat ditangani mesin inferensi. Mesin inferensi memilih aturan yang dapat diterapkan dari basis pengetahuan, mengintegrasikannya, dan mempertimbangkannya untuk mendapatkan kesimpulan.

## 2. Mesin Inferensi

Dalam keputusan kompleks, pengetahuan pakar sering tidak dapat direpresentasikan dalam aturan tunggal. Sebaliknya, aturan dapat digabungkan secara dinamis untuk mencakup berbagai kondisi. Proses penggabungan banyak aturan berdasarkan data yang tersedia, disebut inferensi. Komponen yang melakukan inferensi dalam sistem pakar disebut mesin inferensi. Ada dua pendekatan populer untuk menarik kesimpulan adalah sebagai berikut :

a. *Forward Chaining*

*Forward chaining* adalah mencari bagian JIKA terlebih dahulu. Setelah semua kondisi JIKA dipenuhi, aturan dipilih untuk mendapatkan kesimpulan. Jika kesimpulan diambil dari keadaan pertama, bukan dari yang terakhir, maka ia akan digunakan sebagai fakta untuk disesuaikan dengan kondisi JIKA aturan yang lain untuk mendapatkan kesimpulan yang lebih baik. Proses ini berlanjut hingga dicapai kesimpulan terbaik. Yang digambarkan pada Gambar 2.



**Gambar 2.2** *Forward Chaining*  
(Sri Kusumadewi, 2003, 116 - 118)

b. *Backward Chaining*

*Backward chaining* merupakan metode inferensi yang dimulai dari harapan apa yang akan terjadi (hipotesis) dan kemudian mencaribukti yang mendukung atau yang berlawanan dengan tujuan yang dimaksud, dalam hal ini memerlukan perumusan dan pengujian hipotesis terlebih dahulu.

### C. PHP

PHP pertama kali ditemukan pada 1995 oleh seorang Software Developer bernama Rasmus Lerdorf. Ide awal PHP adalah ketika itu Rasmus ingin mengetahui jumlah pengunjung yang membaca resume onlinenya. Script yang dikembangkan dapat melakukan dua pekerjaan, yakni merekam informasi visitor,

dan menampilkan jumlah pengunjung dari suatu website, dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia web saat ini.

Lerdrof mencoba mengembangkan PHP menggunakan bahasa C dari pada menggunakan Perl. Tahun 1997, PHP versi 2 dirilis, dengan nama *Personal Home Page Form Interpreter* (PHP-FI). PHP semakin populer dan diminati oleh programmer dalam bidang web. Banyak sekali Team Developer yang ikut bergabung dengan Lerdrof untuk mengembangkan PHP hingga menjadi seperti sekarang, hingga akhirnya dirilis versi ke 3-nya, pada Juni 1998, dan tercatat lebih dari 50.000 programmer menggunakan PHP dalam membuat website dinamis.

Pengembangan demi pengembangan terus berlanjut, ratusan fungsi ditambahkan sebagai fitur dari bahasa PHP, dan di awal tahun 1999, *netcraft* mencatat, ditemukan 1.000.000 situs di dunia telah menggunakan PHP. Ini membuktikan bahwa PHP merupakan bahasa yang paling populer digunakan dalam bidang *web development*. Hal ini mengagetkan para developernya termasuk Rasmus sendiri yang tidak menyangka bahasa PHP paling populer dalam bidang *web*. Kemudian Zeev Suraski dan Andi Gutsman selaku *core developer* (programmer inti) mencoba untuk menulis ulang PHP Parser, dan diintegrasikan dengan menggunakan *Zend scripting engine*, dan mengubah jalan alu operasi PHP. Dan semua fitur baru tersebut di rilis dalam PHP 4.

13 Juli 2004, evolusi PHP, PHP telah mengalami banyak sekali perbaikan disegalasasi, dan wajar jika *netcraft* mengumumkan PHP sebagai bahasa *web* populer di dunia, karena tercatat 19 juta domain telah menggunakan PHP sebagai *server side scripting*. PHP saat ini telah Mendukung XML dan *Web Services*, Mendukung *SQLite*. Tercatat lebih dari 19 juta domain telah menggunakan PHP sebagai server scriptingnya. Benar benar PHP sangat mengejutkan.

#### **D. MySQL**

MySQL adalah sebuah implementasi dari sistem manajemen basis data *relasional* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan

turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (*Structured Query Language*).

SQL adalah sebuah konsep pengoperasian Jurnal Teknologi Informasi Rancang Bangun Sistem Perpustakaan untuk Jurnal Elektronik 129 basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keunggulan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja pengoptimisannya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basis data transaksional maupun operasi basis data nontransaksional. Pada modus operasi nontransaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basis data kompetitor lainnya.

#### **E. XAMPP**

XAMPP adalah sebuah *software web server apache* yang didalamnya sudah tersedia *database server MySQL* dan dapat mendukung pemrograman PHP. XAMPP merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah cuma menginstall satu kali sudah tersedia *Apache Web Server, MySQL Database Server, PHP Support* (PHP 4 dan PHP 5) dan beberapa module lainnya.

#### **F. Entity Relationship Diagram (ERD)**

Merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Ada tiga simbol yang digunakan, yaitu:

##### 1. *Entity*

*Entity* adalah objek yang dapat dibedakan dalam dunia nyata. Sedangkan *Entity Set* adalah kumpulan dari *entity* yang sejenis. *Entity Set* dapat berupa objek secara fisik (Rumah, Kendaraan, Peralatan) atau objek secara konsep (Pekerjaan dan Perusahaan). *Entity* disimbolkan dengan persegi panjang.

## 2. *Relationship*

*Relationship* adalah hubungan yang terjadi antara satu atau lebih *entity*. Sedangkan *Relationship Set* adalah kumpulan *relationship* sejenis. *Relationship* disimbolkan dengan jajar genjang.

## 3. *Attribute*

*Attribute* adalah karakteristik dari tiap *entity* atau *relationship* yang menyediakan penjelasan detail mengenai *entity* atau *relationship* tersebut. Nilai dari *attribute* adalah data aktual atau informasi yang disimpan pada suatu *attribute* di dalam *entity* atau *relationship*, dimana tiap *attribute* memiliki domain (*value set*) tersendiri. Domain (*value set*) adalah batas – batas nilai yang diperbolehkan bagi suatu *attribute*. *Attribute* disimbolkan dengan *Ellipse*.

## G. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

#### 1. Fungsi DFD

Fungsi dari Data Flow Diagram adalah :

- a. Data Flow Diagram adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi.
- b. DFD salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi – fungsi sistem merupakan bagian yang lebih penting dan kompleks dari data yang dimanipulasi sistem.
- c. Merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

#### H. Flowchart

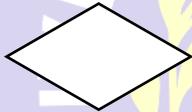
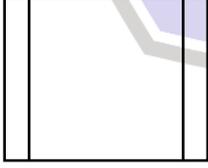
Karena komputer membutuhkan hal-hal yang rinci, maka bahasa pemrograman bukanlah alat baik untuk merancang sebuah algoritma awal. Alat yang banyak dipakai untuk membuat algoritma adalah diagram alur (flowchart).

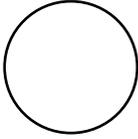
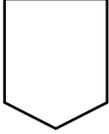
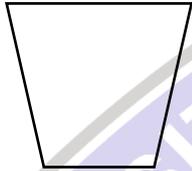
Diagram alur dapat menunjukkan secara jelas arus pengendalian suatu algoritma, yakni melaksanakan suatu rangkaian kegiatan secara logis dan sistematis. Suatu diagram alur dapat memberi gambaran dua dimensi berupa simbol-simbol grafis. Masing-masing simbol telah ditetapkan lebih dahulu fungsi dan artinya. Simbol-simbol tersebut dipakai untuk menunjukkan berbagai kegiatan operasi dan jalur pengendalian. Arti khusus dari sebuah flowchart adalah simbol-simbol yang digunakan untuk menggambarkan urutan proses yang terjadi di dalam suatu program komputer secara sistematis dan logis. (Sumber : Tata Sutabri, S.Kom. MM, 2004,21).

1. Simbol – Simbol Flowchart

Sudah dikemukakan di atas bahwa diagram alur atau flowchart memiliki beberapa simbol yang biasa digunakan untuk menggambarkan rangkaian proses yang harus dilaksanakan. Simbol tersebut dijelaskan dibawah ini:

Tabel 2.2 Simbol – Simbol Flowchart

Simbol Flowchart	Keterangan
1. 	<b>TERMINATOR</b> Simbol ini digunakan untuk mengawali atau mengakhiri suatu proses / kegiatan
2. 	<b>PREPARATION</b> Simbol ini digunakan untuk mempersiapkan nilai awal suatu variabel yang akan diproses dan digunakan untuk proses <i>loop</i>
3. 	<b>DECISION</b> Simbol ini digunakan untuk pengujian suatu kondisi yang sedang diproses
4. 	<b>PROSES</b> Simbol ini digunakan untuk menggambarkan suatu proses yang sedang dieksekusi
5. 	<b>INPUT / OUTPUT</b> Simbol ini digunakan untuk menggambarkan proses <i>input (read)</i> maupun proses <i>output (print)</i>
6. 	<b>SUBROUTINE</b> Simbol ini digunakan untuk menggambarkan proses pemanggilan subprogram dari mainprogram
7. 	<b>FLOW LINE</b> Simbol ini digunakan untuk menggambarkan arus proses dari suatu kegiatan ke kegiatan lain.

<p>8. </p>	<p><i>CONNECTOR</i></p> <p>Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya yang ada di dalam suatu lembar halaman</p>
<p>9. </p>	<p><i>PAGE CONECTOR</i></p> <p>Simbol ini digunakan sebagai penghubung antara suatu proses dengan proses lainnya tetapi berpindah halaman</p>
<p>10. </p>	<p><i>MANUAL OPERATION</i></p> <p>Simbol ini digunakan untuk menggambarkan suatu kegiatan atau proses yang bersifat manualisasi</p>
<p>11. </p>	<p><i>DOCUMENT</i></p> <p>Simbol ini digunakan untuk menggambarkan suatu dokumen atau suatu kegiatan mencetak suatu informasi dengan mesin printer</p>
<p>12. </p>	<p><i>DISPLAY</i></p> <p>Simbol ini digunakan untuk menggambarkan suatu kegiatan menampilkan data atau informasi melalui monitor atau CRT (<i>Cathode Ray Tube</i>)</p>
<p>13. </p>	<p><i>STORED DATA</i></p> <p>Simbol ini digunakan untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media <i>magnetic disk</i></p>
<p>14. </p>	<p><i>MANUAL INPUT</i></p> <p>Simbol ini digunakan untuk menggambarkan proses pemasukan data melalui <i>keyboard</i></p>

## 2. Jenis – Jenis Flowchart

Bentuk diagram alur (*flowchart*) yang sering digunakan dalam proses pembuatan suatu program komputer adalah sebagai berikut :

1) Program *Flowchart*

Simbol – simbol yang menggambarkan proses secara rinci dan *detail* antara intruksi yang satu dengan intruksi yang lainnya dalam suatu program komputer yang bersifat logik.

2) Sistem *Flowchart*

Simbol – simbol yang menggambarkan urutan prosedur secara detail dalam suatu sistem komputerisasi bersifat fisik.

3) Teknik Pembuatan *Flowchart*

Sebelum kita membuat sebuah program komputer, yang harus kita lakukan sebelumnya adalah membuat *flowchart*. Jenis *flowchart* yang sering digunakan adalah program *flowchart*. Teknik pembuatan program *flowchart* ini dibagi menjadi dua bagian, yaitu :

1. *General Way*

Teknik pembuatan *flowchart* dengan cara ini lazim digunakan untuk menyusun logika suatu program. Teknik ini menggunakan pengulangan proses secara tidak langsung

2. *Iteration Way*

Teknik pembuatan *flowchart* dengan cara ini biasanya dipakai untuk logika program yang cepat dan bentuk permasalahannya kompleks. Pengulangan proses yang terjadi bersifat langsung (*Direct-Loop*).

(Sumber : Tata Sutabri, S.Kom. MM; 2004; 24).