

BAB II

LANDASAN TEORI

2.1 Sistem Pakar

2.1.1 Pengertian Sistem Pakar

Menurut Sutojo, Mulyanto Edy, dan Suhartono Vincent (2011, p159), sistem pakar merupakan cabang dari *Artificial Intelligence* (AI) yang cukup tua karena sistem ini mulai dikembangkan pada pertengahan 1960. Secara umum, sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang dilakukan oleh para ahli. Berikut adalah beberapa pengertian sistem pakar:

1. Turban (2001, p402)

“Sistem pakar adalah sebuah system yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan ke dalam sebuah computer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya menggunakan kepakaran atau keahlian manusia”.

2. Jackson (1999, p3)

“Sistem pakar adalah program komputer yang mempresentasikan dan melakukan penalaran dengan pengetahuan beberapa pakar untuk memecakan masalah atau memberikan saran”.

3. Luger dan Stubblefield (1993, p308)

“Sistem pakar adalah program yang berbasiskan pengetahuan yang menyediakan solusi ‘kualitas pakar’ kepada masalah-masalah dalam bidang (domain) yang spesifik”.

2.1.2 Manfaat Sistem Pakar

Menurut Sutojo, Mulyanto Edy, dan Suhartono Vincent (2011, p160), Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, di antaranya:

1. Meningkatkan produktifitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas dengan member nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.

2.1.3 Ciri-ciri Sistem Pakar

Menurut Sutojo, Mulyanto Edy, dan Suhartono Vincent (2011, p162),

Ciri-ciri dari sistem pakar adalah sebagai berikut:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah/rule tertentu
5. Mudah dimodifikasi
6. Keluarannya bersifat anjuran

2.1.4 Basis Pengetahuan

Menurut Kusumadewi Sri (2003, p115), Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah, tentu saja di dalam domain tertentu. Ada dua bentuk pendekatan basis pengetahuan yang sangat umum digunakan, yaitu :

4. 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 si pakar dapat menyelesaikan masalah tersebut secara berurutan. Disamping itu, bentuk ini juga

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

5. Penalaran berbasis kasus (*Case-Based Reasoning*)

Pada 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 apabila kita telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan.

2.1.5 Motor Inferensi (*Inference Engine*)

Inferensi yang dilakukan yaitu Forward Chaining. Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kiri (IF dulu). Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis.

Tabel 2.1 Contoh aturan-aturan

No.	Aturan
R-1	IF A & B THEN C
R-2	IF C THEN D
R-3	IF A & E THEN F
R-4	IF A THEN G
R-5	IF F & G THEN D
R-6	IF G & E THEN H
R-7	IF C & H THEN I
R-8	IF I & A THEN J
R-9	IF G THEN J
R-10	IF J THEN K

Contoh kasus:

Pada tabel 2.1 terlihat ada 10 aturan yang tersimpan dalam basis pengetahuan. Fakta awal yang diberikan hanya A dan F (artinya A & F bernilai benar). Ingin dibuktikan apakah K bernilai benar (hipotesis:K)?

Langkah-langkah inferensi adalah sebagai berikut:

- Dimulai dari R-1. A merupakan fakta sehingga bernilai benar, sedangkan B belum bisa diketahui kebenarannya, sehingga C-pun juga belum bisa diketahui kebenarannya. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-1 ini. Sehingga kita menuju ke R-2.
- Pada R-2, kita tidak mengetahui informasi apapun tentang C, sehingga kita juga tidak bisa memastikan kebenaran D. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-1 ini. Sehingga kita menuju ke R-3.
- Pada R-3, baik A maupun E adalah fakta sehingga jelas benar. Dengan demikian F sebagai konsekuen juga ikut benar. Sehingga kita sekarang mempunyai fakta baru yaitu F. Karena F bukan hipotesis yang hendak kita buktikan (=K), maka penelusuran kita lanjutkan ke R-4.
- Pada R-4, A adalah fakta sehingga jelas benar. Dengan demikian G sebagai konsekuen juga ikut benar. Sehingga sekarang kita mempunyai fakta baru. Karena G bukan hipotesis yang hendak kita buktikan (=K), maka penelusuran kita lanjutkan ke R-5.
- Pada R-5, baik F maupun G bernilai benar berdasarkan aturan R-3, dan R-4. Dengan demikian D sebagai konsekuen jika itu benar. Sehingga sekarang kita mempunyai fakta baru yaitu D. Karena D bukan hipotesis yang hendak kita buktikan (=K), maka penelusuran kita lanjutkan ke R-6.
- Pada R-6, baik A maupun G adalah benar berdasarkan fakta, dan R-4, dengan demikian H sebagai konsekuen juga ikut benar. Sehingga kita sekarang mempunyai fakta baru yaitu H. Karena H bukan hipotesis yang hendak kita buktikan (=K), maka penelusuran kita lanjutkan ke R-7.
- Pada R-7, meskipun H benar berdasarkan R-6, namun kita tidak tahu kebenarannya C, sehingga I-pun juga belum bisa diketahui kebenarannya. Oleh karena itu kita tidak dapat mengetahui Informasi apapun pada R-7 ini. Maka penelusuran kita lanjutkan ke R-8.
- Pada R-8, meskipun A benar karena fakta, namun kita tidak tahu kebenarannya I, sehingga J-pun juga belum bisa diketahui kebenarannya. Oleh karena itu kita tidak mendapatkan Informasi apapun pada R-8 ini sehingga menuju ke R-9.

- Pada R-9, J bernilai benar karena G benar berdasarkan R-4. Karena J bukan hipotesis yang hendak kita buktikan ($=K$), maka penelusuran kita lanjutkan ke R-10.
- Pada R-10, K bernilai benar karena J benar berdasarkan R-9. Karena H sudah merupakan hipotesis yang hendak kita buktikan ($=K$), maka terbukti K adalah benar.

Tabel munculnya fakta baru pada saat inferensi terlihat pada tabel 2.2

Tabel 2.2 Fakta Baru

Aturan	Fakta Baru
R-3	F
R-4	G
R-5	D
R-6	H
R-9	J
R-10	K

2.2 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan merupakan salah satu produk perangkat lunak yang dikembangkan secara khusus untuk membantu dalam proses pengambilan keputusan. Sesuai dengan namanya, tujuan dari dipergunakannya sistem ini adalah sebagai “*second opinion*” atau “*information sources*” yang dapat dipakai sebagai bahan pertimbangan sebelum memutuskan kebijakan tertentu.

2.2.1 Proses Pengambilan Keputusan

Pengambilan keputusan adalah pemilihan beberapa tindakan alternatif yang ada untuk mencapai satu atau beberapa tujuan yang telah ditetapkan (Turban, 2005).

Pengambilan keputusan meliputi beberapa tahap dan melalui beberapa proses. Menurut Supranto (2005), pengambilan keputusan meliputi empat tahap yang saling berhubungan dan berurutan. Empat proses tersebut adalah :

a. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.

b. *Design*

Tahap ini merupakan proses menemukan dan mengembangkan alternatif. Tahap ini meliputi proses untuk mengerti masalah, menurunkan solusi dan menguji kelayakan solusi.

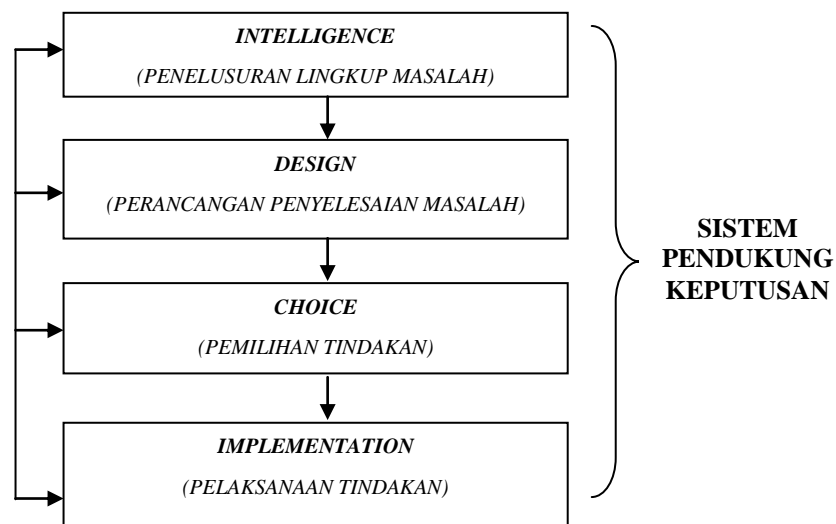
c. *Choice*

Pada tahap ini dilakukan proses pemilihan di antara berbagai alternatif tindakan yang mungkin dijalankan. Tahap ini meliputi pencarian, evaluasi, dan rekomendasi solusi yang sesuai untuk model yang telah dibuat. Solusi dari model merupakan nilai spesifik untuk variabel hasil pada alternatif yang dipilih.

d. *Implementation*

Tahap implementasi adalah tahap pelaksanaan dari keputusan yang telah diambil. Pada tahap ini perlu disusun serangkaian tindakan yang terencana, sehingga hasil keputusan dapat dipantau dan disesuaikan apabila diperlukan perbaikan.

Proses pengambilan keputusan, seperti terlihat pada gambar berikut.



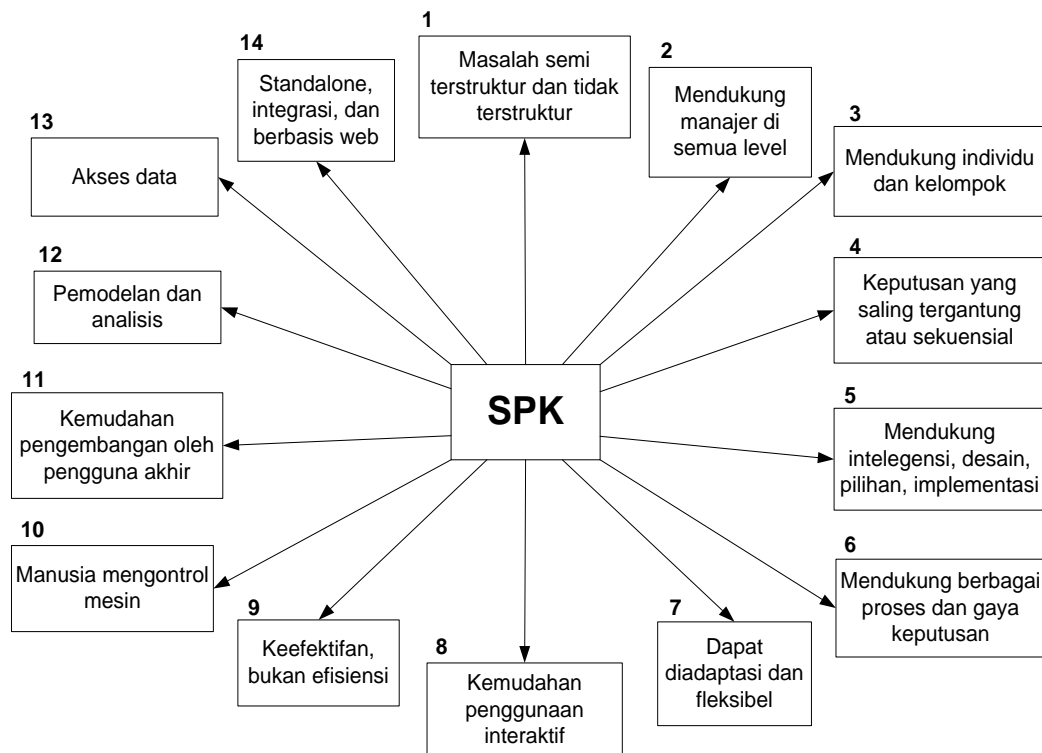
Gambar 2.1 Fase Proses Pengambilan Keputusan

2.2.2 Karakteristik Sistem Pendukung Keputusan

Turban (2005) mengemukakan karakteristik dan kapabilitas kunci dari Sistem Pendukung Keputusan adalah sebagai berikut (Gambar 2.2):

1. Dukungan untuk pengambil keputusan, terutama pada situasi semi terstruktur dan tak terstruktur.
2. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok.
4. Dukungan untuk semua keputusan independen dan atau sekuensial.
5. Dukungan di semua fase proses pengambilan keputusan: inteligensi, desain, pilihan, dan implementasi.
6. Dukungan pada berbagai proses dan gaya pengambilan keputusan.
7. Kemampuan sistem beradaptasi dengan cepat dimana pengambil keputusan dapat menghadapi masalah-masalah baru dan pada saat yang sama dapat menanganinya dengan cara mengadaptasikan sistem terhadap kondisi-kondisi perubahan yang terjadi.
8. Pengguna merasa seperti di rumah. *User-friendly*, kapabilitas grafis yang kuat, dan sebuah bahasa interaktif yang alami.
9. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, *timelines*, kualitas) dari pada efisiensi (biaya).
10. Pengambil keputusan mengontrol penuh semua langkah proses pengambilan keputusan dalam memecahkan masalah.
11. Pengguna akhir dapat mengembangkan dan memodifikasi sistem sederhana.
12. Menggunakan model-model dalam penganalisisan situasi pengambilan keputusan.
13. Disediakkannya akses untuk berbagai sumber data, format, dan tipe, mulai dari sistem informasi geografi (GIS) sampai sistem berorientasi objek.
14. Dapat dilakukan sebagai alat *standalone* yang digunakan oleh seorang pengambil keputusan pada satu lokasi atau didistribusikan di satu

organisasi keseluruhan dan di beberapa organisasi sepanjang rantai persediaan.



Gambar 2.2 Karakteristik dan Kapabilitas SPK

2.2.3 Keuntungan Sistem Pendukung Keputusan

Keuntungan dari sistem pendukung keputusan sebagai berikut :

1. Dapat memperluas kemampuan seseorang untuk mengambil keputusan dalam memproses data atau informasi pemakainya.
2. Membantu mengambil keputusan dalam hal penghematan waktu yang dibutuhkan untuk memecahkan masalah, terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
3. Dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.
4. Dapat menjadi stimulan bagi pengambil keputusan dalam memahami permasalahannya, karena sistem pendukung keputusan mampu menyajikan berbagai alternatif.

5. Mampu menyediakan bukti tambahan untuk memberikan pembenaran, sehingga dapat memperluas posisi pengambilan keputusan.

Sistem Pendukung Keputusan terdiri atas 3 komponen atau subsistem, sebagai berikut:

1. Subsistem Data (*Data Subsystem*)

Merupakan komponen sistem penunjang keputusan sebagai penyedia data bagi sistem yang mana data disimpan dalam *Database Manajement Sistem* (DBMS), sehingga dapat diambil dan diekstraksi dengan cepat.

2. Subsistem Model (*Model Subsystem*)

Keunikan dari sistem ini adalah kemampuannya dalam mengintegrasikan data dengan model-model keputusan.

3. Subsistem Dialog (*User Sistem Interface*)

Melalui sistem dialog ini, sistem dapat diartikulasikan dan diimplementasikan sehingga pengguna atau pemakai dapat berkomunikasi dengan sistem yang dirancang.

2.3 Kesesuaian Lahan

Kesesuaian lahan (*land suitability*) merupakan tingkat kecocokan sebidang lahan untuk suatu penggunaan tertentu. KBBI (2009), Kesesuaian lahan diartikan sebagai hal sesuai dan tidak sesuaiya tanah untuk pemanfaatan tertentu.

Evaluasi lahan adalah suatu proses penilaian sumber daya lahan untuk tujuan tertentu dengan menggunakan suatu pendekatan atau cara yang sudah teruji. Hasil evaluasi lahan akan memberikan informasi dan/atau arahan penggunaan lahan sesuai dengan keperluan. Kesesuaian lahan adalah tingkat kecocokan sebidang lahan untuk penggunaan tertentu. Kesesuaian lahan tersebut dapat dinilai untuk kondisi saat ini (*kesesuaian lahan aktual*) atau setelah diadakan perbaikan (*kesesuaian lahan potensial*).

Kesesuaian lahan aktual adalah kesesuaian lahan berdasarkan data sifat biofisik tanah atau sumber daya lahan sebelum lahan tersebut diberikan masukanmasukan yang diperlukan untuk mengatasi kendala. Data biofisik tersebut berupa karakteristik tanah dan iklim yang berhubungan dengan persyaratan tumbuh tanaman yang dievaluasi. Kesesuaian lahan potensial menggambarkan kesesuaian lahan yang akan dicapai apabila dilakukan usaha-usaha perbaikan. Lahan yang dievaluasi dapat berupa hutan konversi, lahan terlantar atau tidak produktif, atau lahan pertanian yang produktivitasnya kurang memuaskan tetapi masih memungkinkan untuk dapat ditingkatkan bila komoditasnya diganti dengan tanaman yang lebih sesuai.

2.3.1 Klasifikasi kesesuaian lahan

Struktur klasifikasi kesesuaian lahan menurut kerangka FAO (1976) dapat dibedakan menurut tingkatannya, yaitu tingkat Ordo, Kelas, Subkelas dan Unit.

Ordo adalah keadaan kesesuaian lahan secara global. Pada tingkat ordo kesesuaian lahan dibedakan antara lahan yang tergolong sesuai (S=Suitable) dan

lahan yang tidak sesuai (N=Not Suitable). Kelas adalah keadaan tingkat kesesuaian dalam tingkat ordo. Berdasarkan tingkat detail data yang tersedia pada masing-masing skala pemetaan, kelas kesesuaian lahan dibedakan menjadi:

1. Untuk pemetaan tingkat semi detail (skala 1:25.000-1:50.000) pada tingkat kelas, lahan yang tergolong ordo sesuai (S) dibedakan ke dalam tiga kelas, yaitu: lahan sangat sesuai (S1), cukup sesuai (S2), dan sesuai marginal (S3). Sedangkan lahan yang tergolong ordo tidak sesuai (N) tidak dibedakan ke dalam kelas-kelas.
2. Untuk pemetaan tingkat tinjau (skala 1:100.000-1:250.000) pada tingkat kelas dibedakan atas Kelas sesuai (S), sesuai bersyarat (CS) dan tidak sesuai (N).

Kelas S1 : Lahan tidak mempunyai faktor pembatas yang berarti atau nyata terhadap penggunaan secara berkelanjutan, atau faktor pembatas bersifat minor dan tidak akan berpengaruh terhadap produktivitas lahan secara nyata.

Kelas S2 : Lahan mempunyai faktor pembatas, dan factor pembatas ini akan berpengaruh terhadap produktivitasnya, memerlukan tambahan masukan (input). Pembatas tersebut biasanya dapat diatasi oleh petani sendiri.

Kelas S3 : Lahan mempunyai faktor pembatas yang berat, dan faktor pembatas ini akan sangat berpengaruh terhadap produktivitasnya, memerlukan tambahan masukan yang lebih banyak daripada lahan yang tergolong S2. Untuk mengatasi factor pembatas pada S3 memerlukan modal tinggi, sehingga perlu adanya bantuan atau campur tangan (intervensi) pemerintah atau pihak swasta.

Kelas N Lahan yang karena mempunyai faktor pembatas yang sangat berat dan/atau sulit diatasi. Subkelas adalah keadaan tingkatan dalam kelas kesesuaian lahan. Kelas kesesuaian lahan dibedakan menjadi subkelas berdasarkan kualitas dan karakteristik lahan (sifat-sifat tanah dan lingkungan fisik lainnya) yang menjadi faktor pembatas terberat, misal Subkelas S3rc, sesuai marginal dengan pembatas kondisi perakaran (rc =rooting condition). Unit adalah keadaan tingkatan dalam subkelas kesesuaian lahan, yang didasarkan pada sifat tambahan yang berpengaruh dalam pengelolaannya. Contoh kelas S3rc1 dan S3rc2, keduanya mempunyai kelas dan subkelas yang sama dengan faktor penghambat sama yaitu kondisi perakaran terutama factor kedalaman efektif tanah, yang dibedakan ke dalam unit 1 dan unit 2. Unit 1 kedalaman efektif sedang (50-75 cm), dan Unit 2 kedalaman efektif dangkal (<50 cm). Dalam praktek evaluasi lahan, kesesuaian lahan pada kategori unit ini jarang digunakan.

Berbagai sistem evaluasi kesesuaian lahan dilakukan dengan menggunakan pendekatan yang berbeda seperti sistem perkalian parameter, sistem

penjumlahan parameter dan sistem pencocokan (*matching*) antara kualitas lahan dan karakteristik lahan dengan persyaratan tumbuh tanaman yaitu : Sangat sesuai, Cukup sesuai, Sesuai marginal, dan tidak sesuai.

2.4 Metode Forward Chaining

Metode *Forward Chaining* adalah metode pencarian atau teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan. (Russel S, Norvig P, 2003). Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju. Berikut adalah diagram *Forward Chaining* secara umum untuk menghasilkan sebuah *goal*. *Forward chaining* merupakan metode inferensi yang melakukan penalaran dari suatu masalah kepada solusinya. Jika klausa premis sesuai dengan situasi (bernilai TRUE), maka proses akan menyatakan konklusi. Forward chaining adalah *data-driven* karena inferensi dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Jika suatu aplikasi menghasilkan tree yang lebar dan tidak dalam, maka gunakan forward chaining.

Tipe sistem yang dapat dicari dengan Forward Chaining :

1. Sistem yang dipersentasikan dengan satu atau beberapa kondisi.
2. Untuk setiap kondisi, sistem mencari rule-rule dalam knowledge base untuk rule-rule yang berkorespondensi dengan kondisi dalam bagian IF.
3. Setiap rule dapat menghasilkan kondisi baru dari konklusi yang diminta pada bagian THEN. Kondisi baru ini ditambahkan ke kondisi lain yang sudah ada.
4. Setiap kondisi yang ditambahkan ke sistem akan diproses. Jika ditemui suatu kondisi baru dari konklusi yang diminta, sistem akan kembali ke langkah 2 dan mencari rule-rule dalam knowledge base kembali. Jika tidak ada konklusi baru, sesi ini berakhir.

Contoh :

Terdapat 10 aturan yang tersimpan dalam basis pengetahuan yaitu :

R1 : if A and B then C

R2 : if C then D

R3 : if A and E then F

R4 : if A then G

R5 : if F and G then D

R6 : if G and E then H

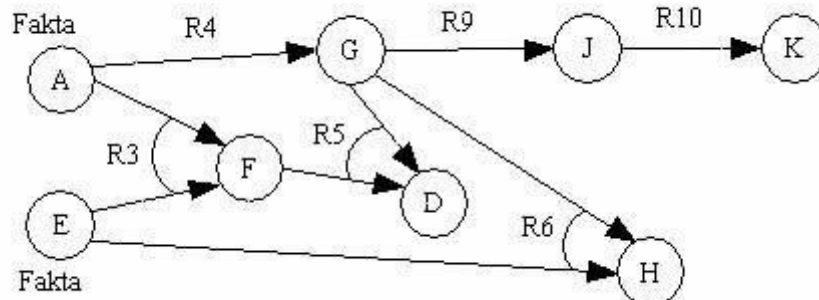
R7 : if C and H then I

R8 : if I and A then J

R9 : if G then J

R10 : if J then K

Fakta awal yang diberikan hanya A dan E, ingin membuktikan apakah K bernilai benar. Proses penalaran forward chaining terlihat pada gambar dibawah :



Gambar 2.3 Forward Chaining

Contoh Kasus

Sistem Pakar : Penasihat Keuangan

Kasus : Seorang user ingin berkonsultasi apakah tepat jika dia berinvestasi pada stock IBM?

Variabel-variabel yang digunakan :

A = memiliki uang \$10.000 untuk investasi

B = berusia < 30 tahun

C = tingkat pendidikan pada level college

D = pendapatan minimum pertahun \$40.000

E = investasi pada bidang Sekuritas (Asuransi)

F = investasi pada saham pertumbuhan (growth stock)

G = investasi pada saham IBM

Setiap variabel dapat bernilai TRUE atau FALSE

FAKTA YANG ADA :

§ Diasumsikan si user (investor) memiliki data:

o Memiliki uang \$10.000 (A TRUE)

o Berusia 25 tahun (B TRUE)

§ Dia ingin meminta nasihat apakah tepat jika berinvestasi pada IBM stock?

RULES :

R1 : IF seseorang memiliki uang \$10.000 untuk berinvestasi AND dia berpendidikan pada level college THEN dia harus berinvestasi pada bidang sekuritas

R2 : IF seseorang memiliki pendapatan per tahun min \$40.000 AND dia berpendidikan pada level college THEN dia harus berinvestasi pada saham pertumbuhan (growth stocks)

R3 : IF seseorang berusia < 30 tahun AND dia berinvestasi pada bidang sekuritas THEN dia sebaiknya berinvestasi pada saham pertumbuhan

R4 : IF seseorang berusia <> 22 tahun THEN dia berpendidikan college

R5 : IF seseorang ingin berinvestasi pada saham pertumbuhan THEN saham yang dipilih adalah saham IBM.

Rule simplification:

- R1: IF A and C, THEN E
- R2: IF D and C, THEN F
- R3: IF B and E, THEN F
- R4: IF B, THEN C
- R5: IF F, THEN G

Solusi dengan Forward Chaining :

- Step I : IF A and C Then E = R1

- Step II : IF B then C A,B,C -> True = R4
- Step III : If A and C then E A,B,C -> True = R2
- Step IV : If B and E then F A,B,C,E,F -> true = R3
- step V : if F then G. G->True

kesimpulan : Cocok untuk investasi saham IBM

2.5 Sistem Informasi Geografis

2.5.1 Pengertian Sistem

Sistem dapat didefinisikan sebagai sekumpulan objek, ide, berikut saling keterhubungannya dalam mencapai tujuan atau sasaran bersama. Pada saat ini banyak pihak yang telah mengalami masalah system untuk kebutuhannya hingga definisinya pun menjadi beragam. Definisi – definisi yang lain yaitu :

Menurut Simatu (1995, p.39), sistem adalah cara pandang terhadap dunia nyata yang terdiri dari elemen – elemen yang saling berinteraksi untuk mencapai tujuan dalam lingkungan yang kompleks.

Menurut Gordon (1989, p.39), sistem adalah suatu kumpulan objek yang terangkai dalam interaksi dan saling ketergantungan yang teratur.

Menurut Rober dan Michael (1991, p.39), sistem adalah kumpulan elemen yang saling berinteraksi membentuk kesatuan, dalam interaksi yang kuat maupun lemah dengan pembatas yang jelas.

2.5.2 Pengertian Informasi

Informasi adalah data yang ditempatkan pada konteks yang penuh arti oleh penerimanya. Menurut pustaka Institute Teknologi Surabaya (1986, p.32), informasi adalah makna atau pengertian yang dapat diambil dari suatu data dengan menggunakan konvensi – konvensi yang umum digunakan di dalam representasinya. Menurut Pustaka Kadir (1999, p.32), informasi adalah data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang, manajer, staf, atau orang lain di dalam suatu organisasi atau perpustakaan.

2.5.3 Pengertian Sistem Informasi

Sistem Informasi adalah *entity* (kesatuan) formal yang terdiri dari berbagai sumber daya maupun logika. Menurut pustaka Budiharto (1995, p.42), sistem informasi adalah suatu sistem manusia dan mesin yang terpadu untuk menyajikan informasi guna mendukung fungsi operasi, manajemen dan pengambilan keputusan dalam organisasi.

2.5.4 Pengertian Geografis

Menurut Richthoffen (1991, p.49), geografi adalah ilmu yang mempelajari permukaan bumi sesuai dengan refrensinya, atau studi mengenai area – area yang berbeda dipermukaan bumi di dalam pengertian karakteristik – karakteristiknya. Menurut Vidal de la Blache (1993, p.49), geografi adalah ilmu mengenai tempat – tempat yang sangat mengkonsentrasikan diri pada kualitas – kualitas dan potensi – potensi suatu negara.

2.5.5 Pengertian Sistem Informasi Geografis

Pada dasarnya, istilah sistem informasi geografis merupakan gabungan dari tiga unsur pokok yaitu sistem, informasi, dan geografis. Dengan demikian, pengertian terhadap ketiga unsur – unsur pokok ini akan sangat membantu dalam memahami SIG. Istilah geografis sendiri merupakan bagian dari *spasial* (keruangan), kata geografis mengandung pengertian suatu persoalan mengenai bumi, permukaan bumi dua atau tiga dimensi. Istilah informasi geografis mengandung pengertian informasi mengenai tempat – tempat yang terletak di permukaan bumi, pengetahuan mengenai posisi dimana suatu objek terletak dipermukaan bumi, dan informasi mengenai keterangan – keterangan yang terdapat di permukaan bumi yang posisinya diberikan atau diketahui.

Dengan demikian SIG merupakan suatu kesatuan formal yang terdiri dari berbagai sumberdaya fisik dan logika yang berkenaan dengan objek – objek yang terdapat di permukaan bumi. Jadi SIG juga merupakan sejenis

perangkat lunak yang dapat digunakan untuk pemasukan, penyimpanan, manipulasi, menampilkan, dan keluaran informasi geografis berikut keterangan – keterangannya.

2.5.6 Konsep Dasar Sistem Informasi Geografis

Sistem informasi geografis dapat didefinisikan sebagai suatu *system* komputer yang digunakan untuk memasukkan, menyimpan, memeriksa, menampilkan, memanipulasi data-data yang berhubungan dengan posisi-posisi dipermukaan bumi. Sistem informasi geografis juga dapat dikatakan sebagai suatu alat yang telah terkomputerisasi untuk pemetaan dan penganalisaan sesuatu yang ada dan peristiwa- peristiwa yang terjadi di permukaan bumi. Kurangnya pengetahuan mengenai penanganan masalah geografis suatu wilayah atau daerah dapat menimbulkan perselisihan antara kedua belah pihak di wilayah tertentu. Maka dengan adanya sistem informasi geografis diharapkan dapat menangani masalah tersebut.

2.5.7 Manfaat Sistem Informasi Geografis

Fungsi Sistem Informasi Geografis adalah meningkatkan kemampuan menganalisis informasi spasial secara terpadu untuk perencanaan dan pengambilan keputusan. Sistem Informasi Geografis dapat memberikan informasi kepada pengambil keputusan untuk analisis dan penerapan *database* keruangan.

Sistem Informasi Geografis mampu memberikan kemudahan-kemudahan yang diinginkan. Dengan Sistem Informasi Geografis kita akan dimudahkan dalam melihat fenomena kebumihannya dengan perspektif yang lebih baik. Sistem Informasi Geografis mampu mengakomodasi penyimpanan, pemrosesan, dan penayangan data spasial digital bahkan integrasi data yang beragam, mulai dari citra satelit, foto udara, peta bahkan data statistik. Sistem Informasi Geografis juga mengakomodasi dinamika data, pemutakhiran data yang akan menjadi lebih mudah.

2.5.8 Komponen Sistem (Subsistem) SIG (Sistem Informasi Geografis)

Beberapa subsistem dalam Sistem Informasi Geografis antara lain adalah :

a. Input

Pada tahap *input* (pemasukan data) yang dilakukan adalah mengumpulkan dan mempersiapkan data spasial dan atau atribut dari berbagai sumber data. Data yang digunakan harus dikonversikan menjadi format digital yang sesuai. Proses konversi yang dilakukan dikenal dengan proses digitalisasi.

b. Manipulasi

Manipulasi data merupakan proses *editing* terhadap data yang telah masuk, hal ini dilakukan untuk menyesuaikan tipe dan jenis data agar sesuai dengan sistem yang akan dibuat, seperti: penyamaan skala, perubahan sistem proyeksi, generalisasi dan sebagainya.

c. Manajemen Data

Tahap ini meliputi seluruh aktifitas yang berhubungan dengan pengolahan data (menyimpan, mengelola, dan menganalisis data) kedalam sistem penyimpanan permanen, seperti: sistem *file server* atau *database server* sesuai kebutuhan sistem.

d. Query

Suatu metode pencarian informasi untuk menjawab pertanyaan yang diajukan oleh pengguna Sistem Informasi Geografis. Pada Sistem Informasi Geografis dengan sistem *file server*, *query* dapat dimanfaatkan dengan bantuan *compiler* atau *interpreter* yang digunakan dalam mengembangkan sistem, sedangkan untuk Sistem Informasi Geografis dengan sistem *database server*, dapat memanfaatkan SQL (*structured query language*) yang terdapat pada DBMS yang digunakan.

e. Analisis

Terdapat dua jenis analisis dalam Sistem Informasi Geografis, yaitu : fungsi analisis spasial, dan analisis atribut. Fungsi analisis spasial adalah operasi yang dilakukan pada data spasial. Sedangkan, fungsi analisis atribut adalah fungsi pengolahan data atribut, yaitu data yang tidak berhubungan dengan ruang.

f. Visualisasi (Data Output)

Penyajian hasil berupa informasi baru atau *database* yang ada baik dalam bentuk *softcopy* maupun dalam bentuk *hardcopy* seperti dalam bentuk : peta (atribut peta dan atribut data), tabel, grafik dan lain-lain.

2.5.9 Cara Kerja SIG (Sistem Informasi Geografis)

Sistem Informasi Geografis dapat menyajikan *real world* (dunia nyata) pada monitor sebagaimana lembaran peta dapat merepresentasikan dunia nyata diatas kertas. Tetapi, Sistem Informasi Geografis memiliki kekuatan lebih dan fleksibilitas dari pada lembaran pada kertas. Peta merupakan representasi grafis dari dunia nyata, obyek-obyek yang dipresentasikan di atas peta disebut unsur peta atau *map features* (contohnya adalah sungai, taman, kebun, jalan dan lain-lain). Karena peta mengorganisasikan unsur-unsur berdasarkan lokasi-lokasinya. SIG menyimpan semua informasi deksriptif unsur-unsurnya sebagai atribut-atribut didalam basis data. Kemudian, Sistem Informasi Geografis membentuk dan menyimpannya didalam tabel-tabel (*relasional*) dengan demikian, atribut-atribut ini dapat diakses melalui lokasi-lokasi unsur-unsur peta dan sebaliknya, unsur-unsur peta juga dapat diakses melalui atribut-atributnya. (Prahasta 2002)

2.6 Google Map API

Google Maps adalah layanan gratis Google yang cukup populer. Para *programmer* dapat menambahkan fitur Google Maps dalam *web* yang dibuat dengan Google Maps API. Google Maps API adalah library JavaScript. Menggunakan/memprogram Google Maps API sangat mudah. Yang di butuhkan adalah pengetahuan tentang HTML dan JavaScript, serta koneksi

Internet. Dengan menggunakan Google Maps API *programmer* dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga para *programmer* dapat fokus hanya pada data-data yang akan diolah. Sedangkan data peta-peta dunia menjadi urusan Google.

Tahapan menggunakan Google Map API sebagai berikut :

1. Memasukkan Maps API JavaScript ke dalam HTML.
2. Membuat element div dengan nama map_canvas untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan property-properti pada peta.
4. Menuliskan fungsi JavaScript untuk membuat objek peta.
5. Meng-inisiasi peta dalam tag body HTML dengan event onload.

2.7 Perancangan Sistem

2.7.1 Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Ia akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh boundary (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada store dalam diagram konteks (Jogiyanto, 2005).

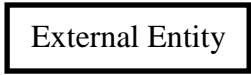


Diagram Konteks merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran data-aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data-

aliran data menuju dan dari sistem diketahui penganalisis dari wawancara dengan *user* dan sebagai hasil analisis dokumen (Jogiyanto, 2005).

Diagram Konteks menggaris bawahi sejumlah karakteristik penting dari suatu sistem (Jogiyanto, 2005):

1. Kelompok pemakai, pihak yang akan memberikan data ke sistem
2. Data, apa saja yang diterima/dihasilkan sistem dari/ke dunia luar
3. Penyimpanan data, tempat sistem harus memberi informasi atau laporan
4. Batasan, yang membedakan antara sistem dan lingkungan

Tabel 2.3. Simbol Diagram Konteks

Simbol Diagram Konteks	Keterangan
 External Entity	Pihak-pihak yang berada di luar sistem, tetapi secara langsung berhubungan dengan sistem dalam hal memberi data atau menerima informasi
 Process	Didalam diagram konteks, berisi mengenai sistem yang akan dibuat
 Data Flow	Berisi data atau informasi yang mengalir dari satu pihak ke sistem dan sebaliknya

2.7.2 DFD (*Data Flow Diagram*)

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 (Pressman, 2002).


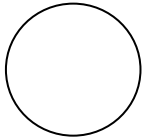
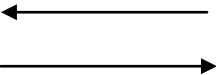

Pada saat informasi mengalir melalui perangkat lunak, dimodifikasi oleh suatu deretan transformasi. Diagram alir data/ DFD (*data flow diagram*) adalah sebuah teknis grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari *input* menjadi *output* (Pressman, 2002).

Berikut ini adalah aturan-aturan pembuatan (DFD) :

1. Didalam *Data Flow Diagram* (DFD) tidak boleh menghubungkan antara *external entity* dengan *entity* lainnya secara langsung.
2. Didalam *Data Flow Diagram* (DFD) tidak boleh menghubungkan *data store* yang satu dengan *data store* yang lainnya secara langsung.
3. Didalam *Data Flow Diagram* (DFD) tidak boleh menghubungkan *data store* dengan *external entity* secara langsung.
4. Setiap proses harus memiliki *data flow* yang masuk dan juga *data flow* yang keluar.

Berikut adalah simbol-simbol DFD:

Tabel 2.4 Simbol-simbol *Data Flow Diagram* (DFD)

Simbol	Keterangan
	<i>External Entity</i> Simbol ini digunakan untuk menggambarkan asal atau tujuan data
	Proses Simbol ini menunjukan proses pengolahan atau tranformasi data
	<i>Data Flow</i> Simbol ini digunakan untuk menggambarkan aliran data yang berjalan
	<i>Data Store</i> Simbol ini menggambarkan <i>data Flow</i> yang sudah disimpan atau diarsipkan

Jenis-jenis DFD (*Data Flow Diagram*):

1. *Context Diagram* (CD)

Jenis pertama *Context Diagram*, adalah *data flow diagram* tingkat atas (DFD *Top Level*), yaitu diagram yang paling tidak detail, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke

luar sistem dan ke dalam dan ke luar entitas-entitas eksternal. (CD menggambarkan sistem dalam satu lingkaran dan hubungan dengan entitas luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem).

Beberapa hal yang harus diperhatikan dalam menggambar CD :

- a. Terminologi sistem
- b. Batas Sistem adalah batas antara “daerah kepentingan sistem”
- c. Lingkungan Sistem adalah segala sesuatu yang berhubungan atau mempengaruhi sistem tersebut
- d. *Interface* adalah aliran yang menghubungkan sebuah sistem dengan lingkungan sistem tersebut

2. Diagram Level n / *Data Flow Diagram Levelled*

Dalam diagram n DFD dapat digunakan untuk menggambarkan diagram fisik maupun diagram diagram logis. Dimana Diagram Level n merupakan hasil pengembangan dari *Context Diagram* ke dalam komponen yang lebih detail tersebut disebut dengan *top-down partitioning*. Jika kita melakukan pengembangan dengan benar, kita akan mendapatkan DFD-DFD yang seimbang.

a. DFD Fisik

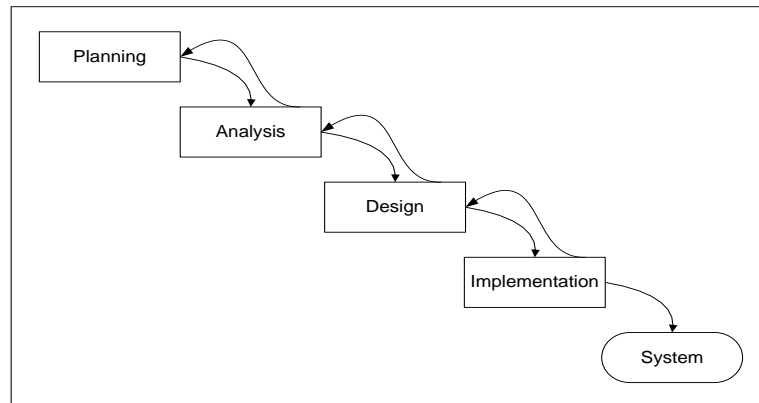
Adalah representasi grafik dari sebuah sistem yang menunjukkan entitas-entitas internal dan eksternal dari sistem tersebut, dan aliran-aliran data ke dalam dan keluar dari entitas-entitas tersebut.

b. DFD Logis

Adalah representasi grafik dari sebuah sistem yang menunjukkan proses-proses dalam sistem tersebut dan aliran-aliran data ke dalam dan ke luar dari proses-proses tersebut. Kita menggunakan DFD logis untuk membuat dokumentasi sebuah sistem informasi karena DFD logis dapat mewakili logika tersebut, yaitu apa yang dilakukan oleh sistem tersebut, tanpa perlu menspesifikasi dimana, bagaimana, dan oleh siapa proses-proses dalam sistem tersebut dilakukan.

2.7.3 Metode Pengembangan Perangkat Lunak Menggunakan Metode *System Development Life Cycle model Waterfall*

Pada metode penelitian ini dilakukan rekayasa perangkat lunak yang digunakan adalah model *Waterfall* seperti pada gambar berikut ini:



(Sumber : Alan Dennis, Barbara H Wixom. 2003)

Gambar 2.4 Metode Pengembangan Model *Waterfall*

Keterangan:

1. *Planning (Perencanaan)*

Tahap perencanaan merupakan proses penting untuk mengetahui mengapa sistem harus dibuat dan menentukan bagaimana cara membangun sistem tersebut. Langkah pertama dari proses tersebut adalah dengan mengidentifikasi peluang apakah dapat memberikan kemungkinan biaya rendah tetapi menghasilkan keuntungan.

2. *Analysis (Analisis)*

Analisis sistem dilakukan untuk memberikan jawaban pertanyaan siapa yang akan menggunakan sistem. Apa yang akan dilakukan oleh sistem, dimana dan kapan sistem tersebut digunakan. Pada tahap ini pembuat sistem akan melakukan observasi dan pengamatan terhadap sistem yang lama, kemudian mengidentifikasi, memanfaatkan dan mengembangkan peluang, dan membangun konsep untuk sebuah sistem baru.

3. *Design (perancangan)*

Tahap perancangan dilakukan untuk menetapkan bagaimana sistem akan dioperasikan. Hal ini berkaitan dengan menentukan perangkat keras,

perangkat lunak, jaringan, tampilan program, *form* dan laporan yang akan dipakai. Selain itu perlu juga menspesifikasi program, database dan file yang dibutuhkan.

4. *Implementation*

Merupakan tahap berikutnya untuk menerjemahkan data atau pemecahan masalah yang telah dirancang ke dalam bahasa pemrograman komputer yang telah ditentukan. Semua tahap ini desain perangkat lunak sebagai sebuah program lengkap atau unit program.

5. *System*

Tahapan ini, merupakan hasil sistem yang telah dibuat dalam bentuk perangkat lunak yang telah dipasang dan digunakan, termasuk didalamnya proses pemeliharaan dan perbaikan kesalahan. Perangkat lunak yang telah selesai dibuat dapat mengalami perubahan-perubahan atau penambahan sesuai dengan permintaan *user* atau perubahan sistem.

2.8 Basis Data

Database adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. (Kristanto: 2003)

Dengan memahami pengertian di atas, maka istilah basis data dapat dipahami sebagai suatu kumpulan data terhubung (*interrelated data*) yang disimpan secara bersama-sama pada suatu media, tanpa tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data (kalaupun ada maka kerangkapan data maka kerangkapan data tersebut harus seminimal mungkin dan terkontrol (*controlled redudancy*), data disimpan dengan cara-cara tertentu sehingga mudah untuk digunakan atau ditampilkan kembali, data dapat digunakan oleh satu atau lebih program-program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakannya, data

disimpan sedemikian rupa sehingga proses penambahan, pengambilan dan modifikasi data dapat dilakukan dengan mudah dan terkontrol (Sutanta, 2004).

Istilah-istilah yang digunakan dalam basis data:

1) *File*

Merupakan kumpulan dari atribut *record-record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda-beda dalam data *value*-nya.

2) *Record*

Merupakan kumpulan dari elemen-elemen yang saling berhubungan atau berkaitan menginformasikan tentang *entry* secara lengkap.

3) *Field*

Merupakan sekumpulan tanda-tanda yang berbentuk kesatuan tersendiri, merupakan bagian terkecil dari *record* dan bentuknya unik dijadikan *field* kunci yang dapat mewakili *record*-nya.

4) Entity

Merupakan tempat kejadian atau konsep yang informasikan direkam.

Sutanta (2004), mengatakan bahwa perancangan basis data dan model data secara umum dapat dibagi menjadi beberapa kelompok, yaitu:

1. Model data berbasis objek
2. Model data berbasis *record*
3. Model data fisik
4. Model data konseptual

Dan dari model-model data tersebut, model data yang sering digunakan pada umumnya adalah model data berbasis objek dan model data berbasis *record*.

1. Model Data Berbasis Objek (*Object Based Logical Model*)

Model data berbasis objek menggunakan konsep entitas, atribut dan hubungan antar entitas. Terdiri dari :


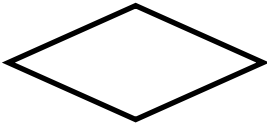
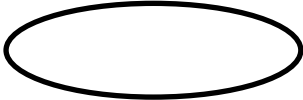
- a. *Entity Relationship Model*
- b. *Binary Model*

c. *Semantic Data Model*

d. *Infological Model*

Jenis model data ini yang paling populer dan sering digunakan adalah *Entity Relationship Model*, yaitu : Model Relasi-Entitas yang pada hakekatnya perwujudan dari model relasional dalam bentuk diagram, yaitu E-R Diagram. *Domain* data disebut juga sebagai himpunan entitas, diwakili oleh diagram kotak. *Field* data atau atribut diwakili oleh diagram lingkaran atau ellips. Hubungan atau relasi antar domain diwakili oleh jajaran-genjang.

Tabel 2.5. Simbol ERD

Simbol ERD	Nama Simbol
	Domain Data
	Relasi antara domain data
	Atribut dari domain data

2. Model Data Berbasis *Record* (*Record-Based Logical Models*)

Model ini berdasarkan pada *record* untuk menjelaskan kepada *user* tentang hubungan *logic* antar data dalam basis data. Terdiri dari :

1. Model Relasional (*Relational Model*)
2. Model Hirarkis (*Hierarchical Model*)
3. Model Jaringan (*Network Model*)

Jenis model data ini yang paling populer dan sering digunakan adalah *Relational Model*, yaitu: Model data yang diciptakan berdasarkan teori-relasional seperti *relational algebra*, dan *relational calculus*. Salah seorang pencetus awal dari basis data relasional adalah E.F.Codd yang juga telah menciptakan serangkaian operasi matematika relasional terhadap model data relasional.

Sutanta (2004), mengatakan bahwa pada prinsipnya model data relasional dapat di-representasikan dalam bentuk tabel data, dimana:

- a. satu tabel mewakili satu “domain” data atau *entity*, bila direkam merupakan satu file yang hanya memiliki satu tipe *record* saja, setiap *record* adalah baris
- b. setiap *record* terdiri atas beberapa *field* (atribut) atau *tuple*, atau kolom
- c. jumlah *tuple / field* pada setiap *record* sama
- d. setiap *record* memiliki atribut kunci utama (*primary key*) yang unik dan dapat dipakai untuk mengenali satu *record*
- e. *record* dapat diurutkan menurut kunci utama
- f. hubungan antara *domain* dinyatakan dalam bentuk relasi, ada tiga kemungkinan relasi antar *domain* yaitu: relasi satu-satu (*one-to-one relation*), relasi satu-banyak (*one-to-many relation*), relasi banyak-banyak (*many-to-many relation*)

2.9 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis (Sidik : 2005)..

2.10 HTML (*Hyper Text Markup Language*)

HTML merupakan kepanjangan dari *Hyper Text Markup Language* adalah suatu bahasa yang digunakan untuk membuat halaman-halaman hypertext (hypertext page) pada internet. Dengan konsep hypertext ini, untuk membaca suatu dokumen anda tidak harus melakukannya secara urut, baris demi baris, atau halaman demi halaman. Tetapi anda tidak dapat dengan mudah melompat dari satu topik ke topic lainnya yang anda sukai, seperti halnya jika anda melakukan pada online Help dari suatu aplikasi Windows. HTML dirancang untuk digunakan tanpa tergantung pada suatu platform tertentu (platform independent) (Kadir : 2006).

2.11 PHP (*Hypertext Preprocessor*)

PHP diciptakan oleh Rasmus Lerdorf, seorang pemrogram C yang handal. Semula PHP hanya digunakan untuk mencatat jumlah pengunjung pada homepagenya. Rasmus adalah seorang pendukung open source. Karena itulah ia mengeluarkan Personal Home Page Tools versi 1.0 secara gratis. Setelah mempelajari YACC dan GNU Bison, Rasmus menambah kemampuan PHP 1.0 dan menerbitkan PHP 2.0. PHP mudah dibuat dan cepat dijalankan, PHP dapat berjalan dalam web server yang berbeda dan dalam sistem operasi yang berbeda pula. PHP dapat berjalan di sistem operasi UNIX, Windows 98, Windows XP, Windows NT, dan Macintosh (Hakim : 2006)..

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP bernama FI (Form Interpreted). Pada saat tersebut PHP adalah sekumpulan script yang digunakan untuk mengolah data form dari web. Perkembangan selanjutnya adalah Rasmus melepaskan kode sumber tersebut dan menamakannya PHP/FI, pada saat tersebut kepanjangan dari PHP/FI adalah Personal Home Page/Form Interpreter. Dengan pelepasan kode sumber ini menjadi open source, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter sudah diimplementasikan dalam C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara

signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend, menulis ulang interpreter PHP menjadi lebih bersih, lebih baik dan lebih cepat. Kemudian pada Juni 1998 perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan nama rilis tersebut menjadi PHP 3.0. Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai. Versi ini banyak dipakai sebab versi ini mampu dipakai untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan proses dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis PHP 5.0. Versi ini adalah versi mutakhir dari PHP. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Dalam versi ini juga dikenalkan model pemrograman berorientasi objek baru untuk menjawab perkembangan bahasa pemrograman ke arah pemrograman berorientasi objek (Kurniawan : 2007)..

Kelebihan PHP :

1. PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web Server yang mendukung PHP dapat ditemukan dimana - mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis - milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena referensi yang banyak. PHP adalah bahasa open source yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara runtime melalui console serta juga dapat menjalankan perintah-perintah system.

Selain berbagai kemudahan, PHP memiliki beberapa masalah atau kekurangan. Pertama, dari segi bahasa ia bukanlah bahasa yang ideal untuk pengembangan berskala besar. Kekurangan utama adalah tidak adanya namespace. Namespace merupakan sebuah cara untuk mengelompokkan nama variabel atau fungsi dalam susunan hirarkis