BAB II

LANDASAN TEORI

2.1 Sistem

2.1.1 Pengertian Sistem

Menurut JOG[9], Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut : Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut : Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Pendekatan sistem yang merupakan kumpulan dari elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang lebih luas. Definisi ini lebih banyak diterima, karena kenyataannya suatu sistem dapat terdiri dari beberapa subsistem atau sistem-sistem bagian. Komponen-komponen atau subsistem-subsistem dalam suatu sistem tidak dapat berdiri lepas sendiri-sendiri. Komponen-komponen atau subsistem-subsistem saling berinteraksi dan

saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai.

2.1.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, karakteristik sistem tersebut menurut JOG[9] yaitu :

a. Komponen-komponen (*Components*)

   Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

b. Batas Sistem (*Boundary*)

   Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

c. Lingkungan Luar Sistem (*Environments*)

   Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

d. Penghubung (*Interface*)

   Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem lainnya.

e. Masukan (*Input*)

   Masukan (*input*) adalah energi yang dimasukan ke dalam sistem.

f. Keluaran (*Output*)

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

g. Pengolah (*Processor*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

h. Sasaran (*Objectives*) atau tujuan (*Goal*)

Tujuan atau sasaran mutlak dimiliki oleh sebuah sistem.

2.2 Representasi Pengetahuan

Representasi pengetahuan berasal dari dua kata, yaitu terdiri dari kata Representasi dan kata pengetahuan. Menurut Kamus Besar Bahasa Indonesia, representasi berarti perbuatan mewakili atau keadaan diwakili. Sedangkan pengetahuan memiliki arti segala sesuatu yang diketahui berkenaan dengan hal tertentu. Berdasarkan pengertian kedua kata tersebut, maka representasi pengetahuan dapat diartikan sebagai suatu keadaan yang mewakili segala sesuatu yang diketahui berkenaan dengan hal tertentu. Suatu pengetahuan dapat direpresentasikan kedalam suatu program aplikasi yang disebut sistem representasi pengetahuan atau sistem pakar. Suatu sistem aplikasi dapat disebut sebagai sistem representasi pengetahuan jika program aplikasi yang digunakan bukan *tools* sistem pakar (seperti PROLOG, LISP, atau SHELL) atau domain pengetahuan yang terdapat pada basis pengetahuan tidak bersumber pada pakar

yang ada (bersumber pada buku referensi atau *manual book*). Sedangkan suatu sistem aplikasi disebut sebagai sistem pakar jika program aplikasi menggunakan *tools* sistem pakar (seperti PROLOG, LISP, atau SHELL) dan domain pengetahuan yang terdapat pada basis pengetahuan bersumber pada pakar yang berkompeten di bidangnya.

2.3 Pengertian Kecerdasan Buatan

Ada beberapa Pengertian *Artificial Intelligence* (AI) atau Kecerdasan buatan, antara lain :

a. Definisi AI menurut SRI[11] adalah sebagai berikut :

   "Kecerdasan buatan atau Artificial Intelligence merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia".

b. Menurut SUP[20], AI didefinisikan sebagai berikut :

   "Artificial Intelligence merupakan sub-bidang pengetahuan komputer yang khusus ditujukan untuk membuat software dan hardware yang sepebuhnya bisa menirukan beberapa fungsi otak manusia".

Berdasarkan beberapa pengertian tersebut, Kecerdasan Buatan dapat diartikan sebagai bagian dari ilmu komputer yang membuat agar mesin (komputer) dapat menirukan fungsi otak manusia sehingga dapat melakukan pekerjaan sebaik yang dilakukan manusia.

2.4 Sistem Pakar

2.4.1 Pengertian Sistem Pakar

Beberapa pengertian tentang sistem pakar, antara lain :

a. Menurut SRI[11], sistem pakar didefinisikan sebagai berikut :

"Sistem pakar (expert system) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli".

b. Menurut SUP[20], sistem pakar didefinisikan sebagai berikut :

"Sistem pakar adalah program artificial intelligence yang menggabungkan pangkalan pengetahuan (knowledge base) dengan sistem inferensi".

c. Menurut DUR[6], sistem pakar didefinisikan sebagai berikut :

"*Sistem Pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan pemecahan masalah dari seorang pakar*."

2.4.2 Struktur Sistem Pakar

Struktur sistem pakar menurut DUR[6] terdiri dari 5 bagian, yaitu basis pengetahuan, memori kerja, mekanisme inferensi, fasilitas penjelasan, dan antarmuka (*interface*).

2.4.2.1 Basis Pengetahuan

Basis pengetahuan adalah bagian dari sistem pakar yang berisi domain pengetahuan. Cara untuk memindahkan (mengkodekan) pengetahuan pakar kedalam basis pengetahuan disebut teknik representasi pengetahuan.

2.4.2.2 Memori kerja

Memori kerja adalah bagian dari suatu sistem pakar yang berisi tentang fakta permasalahan yang ditemui selama proses konsultasi berlangsung. Dengan demikian memory kerja melakukan proses pengingatan (*recording*) terhadap setiap jawaban yang diinputkan oleh pengguna.

2.4.2.3 Mekanisme Inferensi

Mekanisme inferensi (*inference engine*) adalah pemroses dalam suatu sistem pakar yang melakukan pencocokan terhadap fakta yang tersimpan dalam memori kerja dengan domain pengetahuan yang tersimpan dalam basis pengetahuan, untuk memberikan kesimpulan tentang suatu permasalahan. Cara untuk proses penarikan kesimpulan disebut dengan teknik inferensi.

2.4.2.4 Fasilitas Penjelasan

Sebuah *trademark* dari sistem pakar adalah kemampuan sistem untuk menjelaskan alasannya. Suatu sistem pakar memiliki modul tambahan yang disebut fasilitas penjelasan (*explanation facility*). Dengan menggunakan fasilitas

ini, suatu sistem pakar dapat memberikan penjelasan kepada *user* tentang mengapa (*why*) sistem menanyakan pertanyaan tersebut dan bagaimana (*how*) sistem mencapai kesimpulan-kesimpulannya.

2.4.2.5 **Antarmuka (*interface*)**

Interaksi antara sistem pakar dengan user difasilitasi dalam sebuah gaya bahasa alami. Interaksi ini juga sangat interaktif dan mendekati percakapan antar manusia. Desain dasar dari *requirement interface* adalah menanyakan pertanyaan.

2.4.3 Karakteristik Sistem Pakar

Sistem pakar memiliki karakteristik khusus yang membedakan dengan sistem aplikasi lainnya. Karakteristik sistem pakar tersebut menurut DUR[6] antara lain pemisahan pengetahuan dari mekanisme inferensi, memproses pengetahuan pakar, fokus terhadap bidang tertentu, mempertimbangkan dengan simbol (teknik representasi pengetahuan), berpikir secara heuristik, tidak mengabaikan ketidakpastian, keterbatasan pemecahan masalah, berkembang pada kompleksitas pemikiran, dan tidak terlepas dari kesalahan.

2.4.4 Teknik Representasi Pengetahuan

Teknik representasi pengetahuan adalah metode yang digunakan untuk mengkodekan pengetahuan kedalam basis pengetahuan sistem pakar (DUR[6]). Beberapa teknik representasi pengetahuan dalam sistem pakar menurut DUR[6] diantaranya teknik *Object-Attribute-Value Triplets*, teknik aturan-aturan (*Rules*), jaringan semantik (*Semantic Networks*), *frames*, dan logika (*Logic*).

2.4.4.1 ***Object-Attribute-Value* (OAV) *Triplets***

Semua teori kognitif dari pengetahuan manusia menggunakan fakta sebagai blok dasar pengembangan. Fakta adalah sebuah bentuk pengetahuan deklaratif. Fakta ini memberikan beberapa pemahaman tentang suatu kejadian atau permasalahan.

Dalam sistem pakar, fakta digunakan untuk membantu menggambarkan bagian dari *frame*, jaringan semantik, atau *rule*. Dalam OAV, fakta-fakta tersebut dapat bernilai tunggal, ganda, atau bahkan bernilai tidak tentu (*uncertain*). *Object* dapat berupa bentuk fisik atau konsep, *Atribut* adalah karakteristik atau sifat dari object tersebut, *Values* (Nilai) spesifik dari atribut tersebut pada situasi tertentu. Dapat berupa numerik, string atau boolean.

2.4.4.2 Rules

dalam sistem pakar, teknik representasi pengetahuan yang sering digunakan adalah teknik aturan (*rules*). Adapun penggunaan teknik representasi pengetahuan yang lainnya disesuaikan dengan jenis pemecahan masalah yang dihadapi.

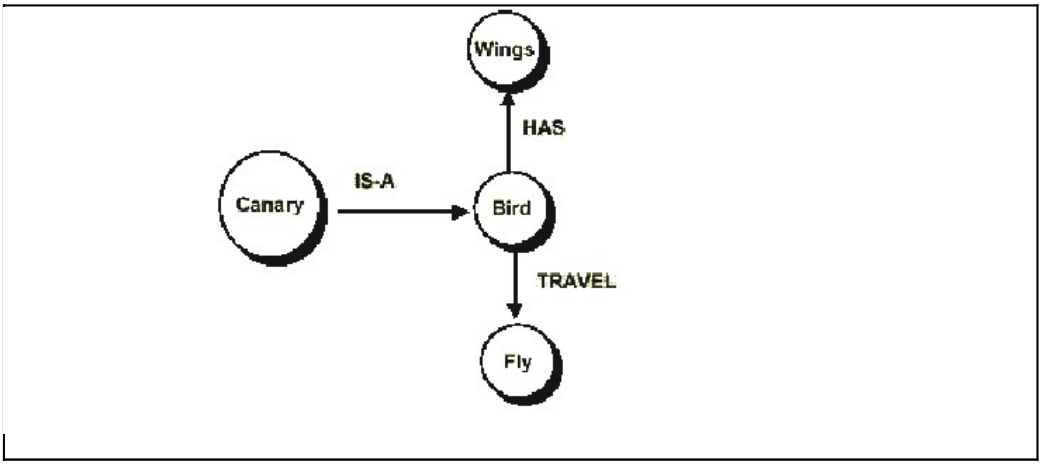
Teknik aturan didefinisikan sebagai suatu struktur pengetahuan yang menghubungkan beberapa informasi yang diketahui dengan informasi lainnya yang dapat disimpulkan atau ditelusuri untuk diketahui.

Berdasarkan bentuk pengetahuannya, teknik aturan dapat digolongkan menjadi hubungan relasi, rekomendasi, strategi, dan *heuristik*. Sedangkan berdasarkan strategi pemecahan masalahnya, teknik aturan dapat digolongkan menjadi interpretasi masalah, diagnosa masalah, dan desain. Contoh dari teknik aturan adalah sebagai berikut :

IF *stain* dari organisme adalah *grampos*
AND morfologi dari organisme adalah *coccus*
AND pertumbuhan dari organisme adalah *chains*
THEN organisme tersebut adalah *streptococcus*

2.4.4.3 Jaringan Semantik

Jaringan semantik merupakan pengetahuan secara grafis yang menunjukkan hubungan antar berbagai objek. Suatu jaringan semantik sederhana bisa dilihat pada gambar berikut :



Gambar 2.1 Contoh jaringan semantik

[Sumber : Durkin, "*Expert System Design and Development*"]

Jaringan semantik disusun dari *node* dan *arc* (*Lines*). *Node* merupakan representasi dari objek, objek properti atau properti *value*. (digambarkan dengan lingkaran)

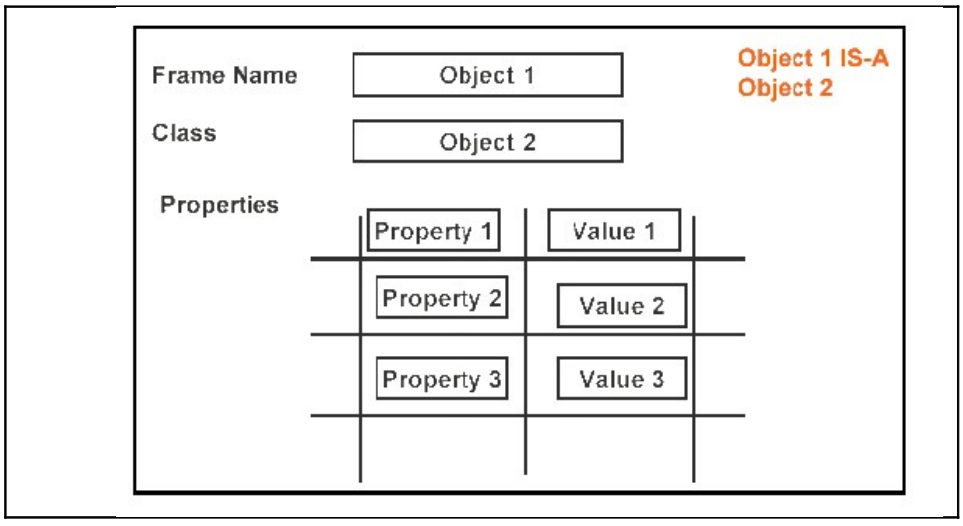
Arc merupakan representasi dari hubungan antar *node*. (digambarkan dengan garis). Baik *node* maupun *arc* memiliki label yang secara jelas menjelaskan representasi dari objek dan hubungan antar *node*.

Hampir semua objek, atribut, pemikiran atau apapun dapat dirumuskan dan dihubungkan antara satu dengan lainnya oleh garis-garis (*arc*). *Arc* umumnya menggunakan istilah seperti : "IS-A**"**, "HAS-A**"**, dll.

Pada gambar contoh tadi: Link “IS-A” diantara node “Canary” dan “Bird” menjelaskan hubungan “*spesific-to-general*” diantara keduanya.

2.4.4.4 Frames

Frame adalah struktur data untuk merepresentasi pengetahuan stereotip dari suatu konsep atau objek. Frames pertama kali diusulkan oleh Minsky (1975). Perancang sistem pakar menggunakan ide yang sama pada SKEMA (*schema*) untuk menangkap dan merepresentasi pengetahuan konseptual dalam Sistem Pakar. Skema merupakan suatu satuan (unit) yang berisi pengetahuan khusus tentang suatu konsep atau objek yang mengandung baik pengetahuan deklaratif maupun prosedural. Skema merupakan perluasan natural dari Jaringan Semantik dan diusulkan pertama kali oleh Barlett (1932). Struktur dari sebuah *frame* ditunjukkan dalam gambar berikut :



Gambar 2.2 Struktur *Frame*

[**Sumber : Durkin, “*Expert System Design and Development*”**]

Secara umum frame memiliki 2 elemen dasar slot dan facet. *Slot* merupakan kumpulan atribut/properti yang menjelaskan objek yang direpresentasi oleh frame

Facet (atau disebut juga: subslot) menjelaskan pengetahuan atau prosedur dari atribut pada slot. *Facet* dapat berbentuk *value*, *default*, *range*, *if added*, *if needed*, dan *other*.

Value menjelaskan tentang nilai dari suatu atribut, *Default* yaitu nilai yang digunakan jika suatu slot kosong atau tidak dideskripsikan pada frame instansiasi, *Range* menandakan jenis dari informasi yang dapat muncul pada slot tersebut (misalnya 0 sampai 100), *If Added* berisi informasi prosedural yang berupa suatu tindakan yang akan dikerjakan jika nilai dari slot diisi (atau berubah), *If Needed* digunakan pada kasus dimana tidak ada value pada slot. Suatu prosedur akan dikerjakan untuk memperoleh atau menghitung sebuah value. Sedangkan *Other*, slot bisa berisi frame, rule, jaringan semantik ataupun tipe lain dari informasi

2.4.4.5 Logika

Logika merupkan suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah dan prosedur yang membantu proses penalaran. Merupakan bentuk representasi pengetahuan yang paling tua, dan menjadi dasar

dari teknik representasi *high level*. Terdiri dari penalaran deduktif, penalaran induktif, dan logika proporsional.

2.4.4.5.1 Penalaran Deduktif

Penalaran ini bergerak dari penalaran umum menuju ke konklusi khusus. Umumnya dimulai dari suatu sylogisme, atau pernyataan premis dan inferensi. Umumnya terdiri dari 3 bagian yaitu premis mayor, premis minor dan konklusi.

Contoh :

Premis mayor : Jika hujan turun saya tidak akan lari pagi

Premis Minor : Pagi ini hujan turun

Konklusi : Oleh karena itu pagi ini saya tidak akan lari pagi

2.4.4.5.2 Penalaran Induktif

Merupakan kebalikan dari deduktif, yaitu dimulai dari masalah khusus menuju ke masalah umum. Menggunakan sejumlah fakta atau premis yang mantap untuk menarik kesimpulan umum.

Contoh:

Premis : Dioda yang salah menyebabkan peralatan elektronik rusak

Premis : Transistor rusak menyebabkan elektronik rusak

Premis : IC rusak menyebabkan peralatan elektronik tidak berfungsi

Konklusi : Maka, peralatan semi konduktor rusak merupakan penyebab utama rusaknya peralatan elektronik.

K o n k l u s i tidak selalu mutlak, dapat berubah jika ditemukan fakta-fakta baru

2.4.4.5.3 Logika Proporsional

Dalam melakukan penalaran dengan komputer, komputer harus dapat menggunakan proses penalaran deduktif dan induktif ke dalam bentuk yang sesuai dengan manipulasi komputer. Metode ini disebut logika komputasional. Bentuk logika komputasional ada 2 macam yaitu Logika Proporsional atau Kalkulus dan Logika Predikat.

Suatu Proposisi merupakan suatu statemen atau pernyataan yang menyatakan benar (*TRUE*) atau salah (*FALSE*).

2.4.5 Teknik Inferensi

Teknik inferensi dalam sistem pakar dimaksudkan untuk memberikan kesimpulan permasalahan dan dapat juga dijadikan bahan penjelasan bagaimana proses pencarian kesimpulan dilakukan.

Teknik inferensi terdiri dari beberapa bagian, penggolongan teknik inferensi yang sering dipakai menurut DUR[6] adalah *forward chaining* dan *backward chaining*.

2.4.5.1 Forward Chaining

Forward chaining atau penelusuran maju adalah strategi inferensi yang diawali dengan sekumpulan fakta yang diketahui, menambah fakta baru dan mencocokkan hasilnya dengan fakta yang ada, dan melanjutkan proses ini hingga kesimpulan akhir dicapai.

2.4.5.2 Backward Chaining

Backward chaining atau penelusuran mundur adalah strategi inferensi yang mencoba untuk membuktikan hipotesis dengan mengumpulkan informasi pendukung.

2.5 Internet

Internet merupakan jaringan global yang menghubungkan suatu jaringan (*network*) dengan *network* lainya di seluruh dunia (AHM[2]).

TCP/IP merupakan protokol penghubung antar jaringan yang beragam di seluruh dunia dalam melakukan komunikasi internet.

Internet merupakan aset yang besar bagi semua penggunanya, mengingat begitu cepatnya jaringan ini dalam menyalurkan informasi tanpa terhambat masalah perbedaan ruang dan waktu.

Berbagai lembaga mulai dari pendidikan, pemerintahan, perusahaan dagang, penerbitan sampai marketing dan komersial telah memanfaatkan fasilitas jaringan ini untuk melakukan transfer informasi, baik itu berupa komunikasi, berita atau

iklan. Dikarenakan kebutuhan masyarakat terhadap informasi dalam internet semakin meningkat, banyak pihak terdorong untuk meningkatkan kualitas internet menjadi lebih baik.

2.5.1 Jaringan Internet

Jaringan internet adalah suatu jaringan komputer global yang terbentuk dari jaringan-jaringan komputer lokal dan regional yang memungkinkan terjadinya komunikasi antar jaringan di seluruh dunia (AHM[2]).

Internet dimulai sekitar tahun 1960, ketika Departemen Pertahanan Amerika Serikat memulai suatu proyek yang bernama *ARPANET*. Beberapa universitas di Amerika Serikat diminta bantuan dalam mengerjakan proyek ini dan awalnya telah berhasil menghubungkan 4 komputer dilokasi universitas yang berbeda. Perkembangan *ARPANET* ini cukup pesat jika dilhat dari perekembangan komputer pada saat itu. Sebagai gambarannya pada tahun 1977 *ARPANET* telah menghubungkan lebih dari 100 *mainframe* komputer dan saat ini terdapat sekitar 4 juta *host* jaringan yang terhubung pada jaringan ini. Jumlah sebenarnya dari komputer yang terhubung tidak dapat diketahui dengan pasti, karena perkembangan jumlah komputer yang terhubung dengan suatu jaringan semakin lama semakin besar.

Karena perkembangannya yang sangat pesat jaringan komputer ini tidak dapat lagi dikatakan sebagai *ARPANET*. Konsep ini yang kemudian berkembang

dan dikenal dengan konsep *Internetworking* (jaringan antar jaringan), dan orang menyebut jaringan besar komputer tersebut dengan istilah internet.

2.5.2 World Wide Web

Salah satu layanan aplikasi internet adalah *World Wide Web*, aplikasi yang cukup baru dikembangkan pada internet dan menjadi layanan aplikasi yang paling populer dikalangan pengguna jaringan internet, perkembangan terus dilakukan sampai saat ini untuk menyempurnakan teknologi ini. *World Wide Web* atau yang biasa disebut dengan web, bekerja dengan menggunakan teknologi yang disebut *hypertext*, yang kemudian dikembangkan menjadi suatu protokol aplikasi yang disebut dengan HTTP (*HyperText Transfer Protocol*).

Web merupakan aplikasi yang paling mudah untuk menjelajahi internet. Dengan menggunakan web, pengaksesan beragam sumber informasi di internet misalnya *gopher, wais, ftp dan e-mail*, dapat dilakukan dengan satu cara yang menggabungkan beberapa jenis representasi dan metode pengaksesan informasi kemudian menyajikannya dalam beragam bentuk informasi.

2.5.2.1 Protokol HTTP

HTTP (*HyperText Transfer Protocol*) adalah protokol yang dipergunakan untuk mentransfer dokumen web (WIK[23]).

Protokol ini adalah protokol ringan, tidak berstatus dan generik yang dapat dipergunakan di berbagai macam tipe dokumen. Pengembangan HTTP dikoordinasi oleh *World Wide Web Consortium* (W3C) dan grup kerja *Internet Engineering Task Force* (IETF), bekerja dalam publikasi satu seri RFC, yang paling terkenal RFC 2616, yang menjelaskan HTTP/1.1, versi HTTP yang digunakan umum sekarang ini.

HTTP adalah sebuah protokol meminta dan menjawab antara *client* dan *server*. Sebuah client HTTP seperti *web-browser* memulai permintaan dengan membuat hubungan *TCP/IP* ke port tertentu di tuan rumah yang jauh (port 80 merupakan standar untuk HTTP). Sebuah server HTTP yang mendengarkan di port tersebut menunggu *client* mengirim kode permintaan (*request*), seperti "GET / HTTP/1.1" (meminta halaman yang sudah ditentukan), diikuti dengan pesan *MIME* yang memiliki beberapa informasi kode *header* dan menjelaskan aspek dari permintaan tersebut, diikut dengan badan dari data tertentu (*body*).

Beberapa *header* bebas untuk tidak dituliskan, sementara *header* lainnya (seperti *host*) harus selalu dituliskan ketika melakukan *request*. Begitu menerima kode permintaan (dan pesan, bila ada), server mengirim kembali kode jawaban, seperti "200 OK", dan sebuah pesan yang diminta.

Protokol HTTP pertama kali dipergunakan dalam web pada tahun 1990. Pada saat tersebut protokol yang digunakan adalah protokol HTTP versi 0.9. Versi 0.9 ini adalah protokol transfer dokumen secara mentah, maksudnya adalah data dokumen dikirim sesuai dengan isi dari dokumen tersebut tanpa

memandang tipe dari dokumen. Kemudian pada tahun 1996 protokol HTTP diperbaiki menjadi HTTP versi 1.0. Perubahan ini untuk mengakomodasi tipe-tipe dokumen yang hendak dikirim beserta enkoding yang dipergunakan dalam pengiriman data dokumen. Sesuai dengan perkembangan infrastruktur internet maka pada tahun 1999 dikeluarkan HTTP versi 1.1 untuk mengakomodasi *proxy* dan *cache*. Berikut adalah contoh koneksi HTTP:

```
S = Server C = Client

C : (Inisialisasi koneksi)
C : GET /index.htm HTTP/1.1
C : Host: www.wikipedia.org
S : 200 OK
S : Mime-type: text/html
S : <CRLF>
S : -- data dokumen --
S : (close connection)
```

2.5.2.2 Protokol HTTPS

HTTPS adalah versi aman dari HTTP, protokol komunikasi dari *World Wide Web*. Ditemukan oleh *Netscape Communications Corporation* untuk menyediakan autentikasi dan komunikasi tersandi dan penggunaan dalam *e-commerce* (WIK[23]).

Selain menggunakan komunikasi plain text, *HTTPS* menyandikan data *session* menggunakan protokol SSL (*Secure Socket layer*) atau protokol TLS (*Transport Layer Security*). Kedua protokol tersebut memberikan perlindungan yang memadai dari serangan *eavesdroppers*, dan *man in the middle attacks*. Pada umumnya port HTTPS adalah 443. Tingkat keamanan tergantung pada

ketepatan dalam mengimplementasikan pada *browser* web dan perangkat lunak *server* dan didukung oleh algoritma penyandian yang aktual.

Oleh karena itu, pada halaman web digunakan HTTPS, dan *url* yang digunakan dimulai dengan '**https://**' bukan dengan '**http://**'.

Kesalahpahaman yang sering terjadi pada pengguna kartu kredit dalam web adalah dengan menganggap HTTPS sepenuhnya melindungi transaksi. Sedangkan pada kenyataannya, HTTPS hanya melakukan enkripsi informasi antara *browser* dengan *web server* yang menerima informasi. Pada *web server*, informasi tersebut tersimpan dalam *database server* (terkadang tidak langsung dikirimkan ke pemroses kartu kredit), dan *database server* inilah yang paling sering menjadi sasaran penyerangan oleh pihak-pihak yang tidak berkepentingan.

2.5.2.3 XHTML

XHTML *(Extensible Hypertext Markup Language*) adalah pengembangan standarisasi terbaru dari generasi HTML 4 sebagai bahasa *markup* untuk website (AHM[2]).

Keunikan dokumen ini adalah dapat digunakan pada protokol yang sangat simpel, ukuran file yang kecil, dan mempunyai fasilitas *hyperlink*, yang dapat dengan mudah berpindah dari satu halaman ke halaman lainnya.

XHTML ditulis dalam format *plain-text* yang terdiri dari tag-tag XHTML. Interpretasi script XHTML dilakukan pada *browser*, sehingga semua orang dapat melihat *source code* XHTML tersebut dengan mudah. File XHTML memiliki ekstensi *default* **html** dan **htm**. Penulisan source code XHTML dapat dilakukan dengan menggunakan *word processor* biasa, seperti *notepad*, *Ultra Edit, Cute HTML, Kate, gedit* dan *vi.*

Untuk menampilkan hasil dari file XHTML tersebut diperlukan *browser*. *Browser* merupakan *software* yang pada *client* yang berfungsi untuk menterjemahkan tag-tag XHTML menjadi halaman web. *Browser* yang sering di gunakan oleh para *surfer* adalah *Microsoft Internet Explorer, Opera, Mozilla Suite* dan *Firefox.*

2.5.2.4 Javascript

Javascript adalah suatu bahasa pemrograman berbasis "*web client*". Bahasa pemrograman ini tidak dapat berdiri sendiri. Penulisan *script* harus disipkan dalam file HTML baik itu secara langsung maupun melalui penghubung (*link*). Interpretasi *javascript* dilakukan pada *browser* tanpa harus merubah setting pada *server*, tetapi tidak semua *browser* yang ada dapat melakukan interpretasi *javascript*. Selain itu hasil interpretasi suatu *browser* belum tentu sama dengan interpretasi pada *browser* lainnya, sehingga pada

beberapa kasus suatu *script* yang dapat berjalan pada *browser* tertentu tidak akan berjalan dengan baik pada *browser* lainnya.

Solusi pemecahan masalah ini adalah dengan mencoba *script* yang telah dibuat pada beberapa *browser*.

2.5.2.5 Browser

Browser atau penjelajah web adalah suatu paket perangkat lunak yang memungkinkan seorang pengguna untuk menampilkan dan berinteraksi dengan dokumen-dokumen yang disimpan pada *server web*. Beberapa *browser* yang sering digunakan adalah *Microsoft Internet Explorer, Opera* dan *Mozilla Firefox.*

2.6 LAMP

LAMP adalah paket *web-development* yang berisikan aplikasi Linux, Apache, MySQL, dan PHP. LAMP merupakan paket aplikasi *open source* yang paling populer untuk mengembangkan Aplikasi Online berbasis web.

2.6.1 Linux (GNU/Linux)

Linux atau *GNU/Linux* adalah sistem operasi gratis yang sangat populer. Istilah *linux* atau *GNU/Linux* (*GNU*) juga digunakan sebagai rujukan kepada

keseluruhan *distro linux* (*Linux distribution*), yang di dalamnya selalu disertakan program-program lain yang mendukung sistem operasi ini (WIK[22]).

Contoh program-program tersebut adalah *web server*, bahasa pemrograman, *database*, *desktop*, dan aplikasi/*software* perkantoran (*office suite*) seperti *OpenOffice.org*. *Distro linux* telah mengalami pertumbuhan yang pesat dari segi kepopuleran, sehingga lebih populer dari versi *UNIX* yang *proprietary* dan mulai menandingi dominasi *Microsoft Windows* dalam beberapa sisi. Linux mendukung banyak *device* komputer, dan telah diguna di dalam berbagai jenis komputer, *super-computer* dan *embedded system*.

Kernel Linux pada awalnya ditulis sebagai hobi oleh pelajar Universitas Finland *Linus Torvalds* yang belajar di Universitas Helsinki, untuk membuat kernel Minix yang gratis dan dapat diedit (Minix adalah proyek pembelajaran menyerupai UNIX dengan tujuan). Versi 0.01 dikeluarkan ke internet pada September 1991, Versi 0.02 pada 5 Oktober 1991.

2.6.2 Apache HTTPD2

Apache HTTPD 2 adalah *server* web yang bertugas sebagai penyedia layanan web pada suatu komputer, baik itu layanan penerimaan *request* atau pengiriman *response* (AHM[3]).

Apache HTTPD 2 adalah proyek *open source* dan *platform independent* dari Apache Group dan merupakan server web yang paling banyak digunakan di

internet. Standarisasi web, kecepatan parsing, dan kemudahan konfigurasi menjadikan Apache sangat digemari oleh para penggunanya. Apache HTTPD2 dapat di *download* gratis di (http://httpd.apache.org).

2.6.3 MySQL

MySQL adalah sebuah *database server* yang didukung oleh PHP untuk menyimpan berbagai macam data. PHP memang mendukung banyak database, tetapi MySQL merupakan salah satu yang paling populer.

Keunggulan MySQL dari database lainnya adalah kecepatan dalam parsing query dan dukungan lintas platform yang sangat beragam, selain itu MySQL adalah proyek *open source*. MySQL dikembangkan oleh MySQL AB, dan dapat di download gratis pada website (http://www.mysql.com).

2.6.4 PHP

PHP adalah *recursive ancronym* dari *Hypertext Preprocessor* yang awalnya dibuat oleh Rasmus Lerdorf pada tahun 1995 sebagai bahasa pemrograman untuk websitenya yang pada waktu itu masih merupakan *ancronym* dari *Personal Home Page*. Tapi dikarenakan kemudahan dan kefleksibelan bahasa pemrograman ini, banyak pihak yang ingin mengembangkannya, diantaranya Andi Gutmans dan Zeev Suraski yang berhasil menulis ulang *core* dari PHP tersebut dan

menamainya sebagai *Zend Engine* yang berasal dari nama mereka berdua (*Zeev and Andi*).

Dengan *Zend Engine* tersebut PHP dapat menerima *dynamic library* yang bertindak sebagai fungsi-fungsi diluar dari fungsi standar PHP itu sendiri. Banyak sekali external library *third party* (dari pihak ketiga) yang dapat digunakan pada PHP, mulai dari pembuatan dokumen *PDF, flash atau image,* penggunaan koneksi network, *enkripsi* data, sampai pada implementasi *e-commerce* menggunakan kartu kredit yang semuanya bisa didapatkan dengan gratis.

Selain gratis, PHP juga merupakan proyek o*pen source* yang berlisensi *open public* dan merupakan bahasa pemrograman *platform independent* yang dapat berjalan pada sistem operasi apapun.

Walaupun merupakan bahasa pemrograman *web-based,* tetapi PHP dapat melakukan lebih dari sekedar *hypertext preprocessor*, applikasi manajemen dan server binding juga dapat di implementasikan dengan bahasa pemrograman ini.

PHP dapat di download secara gratis pada website (http://www.php.net/download.php).

2.7 Pengertian Diagnosa

Menurut Kamus Besar Bahasa Indonesia, diagnosa (diagnosis) adalah penelitian jenis penyakit berdasarkan gejala-gejalanya.

Sedangkan pengertian mendiagnosa adalah menentukan jenis penyakit dengan meneliti atau memeriksa gejala-gejalanya.

2.8 Gangguan Kehamilan Pada Ibu Hamil

2.8.1 Definisi Gangguan Kehamilan

Gangguan kehamilan berasal dari dua kata, yaitu kata gangguan dan kehamilan. Menurut Kamus Besar Bahasa Indonesia, gangguan diartikan sebagai hal yang menyebabkan ketidak normalan / ketidak lancaran, sedangkan kehamilan diartikan sebagai keadaan hamil.

Berdasarkan pengertian kedua kata tersebut, maka gangguan kehamilan dapat didefinisikan sebagai suatu hal yang menyebabkan ketidak normalan dalam keadaan hamil. Sedangkan objek yang mengalami kehamilan ini adalah Ibu hamil yang didefinisikan sebagai wanita (ibu) yang mengandung janin bayi dalam rahimnya.

2.8.2 Macam-macam Gangguan Kehamilan

Gangguan kehamilan dapat terjadi dari beberapa jenis berikut, diantaranya (1) penyakit yang khusus diderita wanita hamil (*Gestose*), (2) gangguan yang disebabkan karena kondisi medis yang terdahulu, (3) Penyakit infeksi, dan (4) gangguan yang sering timbul dikarenakan suatu kondisi tertentu yang menjadi penyebabnya.

2.8.2.1 Gestose

Gestose adalah penyakit yang khas untuk kehamilan (*gestatio*=kehamilan), jadi penyakit ini tidak terdapat pada wanita yang tidak hamil (OBS[..]). Yang termasuk dalam golongan penyakit ini adalah (1) *Gestose* dini yang timbul pada hamil muda (*hyperemesis gravidarum*) dan (2) *Gestose* lambat yang timbul pada kehamilan lanjut atau *hypertensi* dalam kehamilan (*EPH Gestose*).

2.8.2.1.1 Hyperemesis Gravidarum

Gejala dari penyakit ini adalah mual dan muntah yang menyebabkan semua yang dimakan dan diminum keluar kembali hingga menyebabkan berat badan turun dan mempengaruhi keadaan umum penderitanya. Jika hal ini terjadi maka diperlukan perawatan di rumah sakit.

Penyakit ini biasanya mulai setelah minggu ke 6 dan baik sendiri sekitar minggu ke 12.

Pada bentuk yang ringan, penderita penyakit ini hanya merasa mual atau muntah pada pagi hari saja, tengah hari sudah biasa kembali, maka penyakit ini disebut "*morning sickness*" (*emesis gravidarum*). Keadaan ini tidak mempengaruhi keadaan umum penderita (BOG[4]).

2.8.2.1.2 Penyakit Hypertensi dalam kehamilan

Penyakit hipertensi dalam kehamilan merupakan kelainan jantung (*vaskuler*) yang terjadi sebelum kehamilan atau timbul dalam kehamilan atau pada permulaan nifas. Penyakit hypertensi dalam kehamilan ini terdiri dari preeklampsi dan eklampsi.

Hypertensi yang hanya terjadi dalam kehamilan dan khas untuk kehamilan. Pada tingkat tanpa kejang disebut *preeklampsi* dan pada tingkat dengan kejang disebut *eklampsi*.

Preeklampsi memperlihatkan gejala hypertensi, pembengkakan (*oedem*) dan kadar protein dalam darah tinggi (*proteinuria*), kadang-kadang hanya hypertensi dengan *proteinuria* atau hypertensi dengan pembengkakan (*oedem*).

Eklampsi sama gejala-gejalanya dengan preeklampsi ditambah dengan kejang dan / atau koma. Jadi *preeklampsi* dan *eklampsi* merupakan satu penyakit hanya tingkatnya yang berlainan (BOG[4]).

2.8.2.2 Gangguan Karena Kondisi Medis Terdahulu

2.8.2.2.1 Diabetes

Diabetes adalah gangguan *endokrin* yang ditandai dengan tingginya kadar glukosa darah (*glikosuria*). Diabetes menyerang 1 dalam 300 kehamilan sebelum ditemukan insulin pada tahun 1921 (HAM[7]).

Diabetes yang muncul saat kehamilan atau disebut diabetes semasa kehamilan (*gestational diabetic*), disebabkan saat hamil kebutuhan karbohidrat meningkat. Karena hormon insulin dalam tubuh tidak mencukupi untuk mengubah karbohidrat tersebut menjadi gula, maka terjadilah penimbunan kadar gula yang tinggi dalam darah. Namun diabetes ini bersifat sementara, artinya bisa hilang setelah melahirkan, meskipun bisa juga menjadi parah bila tak terdeteksi.

Baik diabetes semasa kehamilan maupun sejak sebelum hamil (*pregestational diabetic*) berdampak pada kelahiran bayi diatas 4000 gram (*giant baby* atau bayi besar), selain juga bisa bayi meninggal di kandungan terutama sering terjadi di usia kehamilan 34-36 minggu. Cacat multiple organ pada bayi juga bisa terjadi, seperti tidak ada tempurung kepala, tidak tertutupnya sumsum tulang belakang, tidak ada lubang dubur, kelainan jantung, ginjal, dan saraf. Saat persalinan dapat terjadi *distosia* (persalinan macet), rasa mulasnya bagus tetapi pembukaannya tidak maju dan kepala bayi pun tidak turun.

Gejala yang khas dari penyakit ini dan harus diwaspadai adalah 3P (*Polyphogie*/banyak makan, *Polyurie*/banyak kencing, *Polydipsie*/banyak minum).

Bagi pengidap diabetes sebelum hamil dan selama ini rutin mengkonsumsi obat, begitu tahu hamil harus berhenti minum obat oral dan diganti dengan suntikan. Sementara yang diabetesnya dipicu oleh

kehamilan, harus menjalani diet khusus selama beberapa waktu tertentu, diikuti pemeriksaan kadar gula darah kembali.

2.8.2.2.2 Penyakit Jantung

Ada beberapa gangguan jantung yang berkaitan dengan kehamilan. artinya, penyakit/gangguan jantung tersebut baru diketahui atau timbul gejala medisnya saat hamil berupa gangguan gagal jantung, semisal sesak napas atau jantung berdebar-debar. Salah satunya, penyakit jantung katup akibat adanya penyempitan atau terjadi kebocoran pada katup jantung. Semakin parah penyakit jantungnya, semakin besar kemungkinan terjadi keguguran. Masa rawan berlangsung di usia kehamilan 16 minggu dan minggu 32-34. hal tersebut terjadi karena pada minggu ke 16, peredaran darah meningkat hingga bisa menambah beban jantung, sementara volume darah terus meningkat sampai usia kehamilan 32-34 minggu, lalu bertahan pada posisi cukup tinggi sampai menjelang persalinan. Selain itu cairan tubuh juga maningkat maksimal. Situasi ini membuat beban jantung tambah berat, hingga suplai makanan dan oksigen dari peredaran darah ke ibu janin pun tambah berat gangguannya.

Itu sebabnya ibu harus banyak istirahat dan menghindari stress emosional di masa rawan. Mereka yang lebih beresiko terkena gangguan hamil selagi hamil adalah ibu dengan kehamilan kembar, ada riwayat keluarga yang mengalami tekanan darah tinggi pada kehamilan, atau juga

bila pada kehamilan sebelumnya ada gangguan jantung temporer, mengingat gangguan tersebut cenderung berulang.

2.8.2.2.3 Anemia

Penyakit ini bisa dibawa sebelum kehamilan atau timbul setelah kehamilan berlangsung. Jika sebelum kehamilan sudah anemia, maka akan bertambah berat ketika hamil. Anemia defisiensi zat besi paling sering dialami ibu hamil. Bisa akibat minimnya pemasukan unsur besi dari makanan ke tubuh, atau makanan tersebut memang kurang mengandung zat besi, bisa juga diakibatkan terlalu banyak zat besi yang keluar dari badan, seperti karena pendarahan, atau penyakit wasir yang kronis.

Anemia ini bisa berdampak *abortus* (keguguran), persalinan yang lama, pendarahan pasca melahirkan, kelahiran prematur dibawah 37 minggu, BBLR (Berat Bayi Lahir Rendah), kematian mudigah (terjadi saat kehamilan muda), serta kemungkinan lahir dengan cacat bawaan. Gejala yang dialami biasanya mudah lelah, lesu, dan sesak napas saat beraktivitas atau berolahraga berat, permukaan kulit dan wajah pucat, mudah pusing, dan gampang pingsan. Bila penyakit ini tergolong berat, diperlukan penanganan yang cepat berupa transfusi darah atau obat zat besi diberikan secara intravenal. Sementara untuk yang ringan dan sedang, cukup diberi suplemen zat besi dengan asam folat.

2.8.2.2.4 APS (Sindrom Antifosfolipid)

APS sering disebut pula sindrom ACA (*anticardiolipin*), merupakan kelainan pengentalan darah. Ibu hamil yang memiliki riwayat ginekologis yang buruk misalnya keguguran berulang, janin mati, dan preeklampsia semasa kehamilan terdahulu lebih berisiko terkena. dampak yang ditimbulkan adalah janin akan gugur pada masa embrio atau sebelum mencapai usia kehamilan 8 minggu karena suplai darah ibu yang membawa nutrisi dan oksigen lewat plasenta tak lancar. bila dicurigai terkena APS, dianjurkan menjalani tes ACA.

Pada penyakit ini tidak menunjukkan gejala yang spesifik. Pada ibu hamil, gejalanya mirip dengan yang biasa dialami ibu hamil, seperti cepat lelah, mengantuk, sering pusing, dan sulit konsentrasi. Gejala-gejala tersebut akan terus berlangsung sepanjang kehamilan, beda dengan kehamilan biasa yang kembali normal setelah 4 bulan (IBU[8]).

2.8.2.3 Penyakit Infeksi

Infeksi TORCH banyak menjangkiti ibu hamil. TORCH singkatan dari *Toksoplasmosis*, *Other* (antara lain *HIV*, *Sipilis*, *Klamidia*), *Rubella*, *Cytomegalovirus*, *Herpes*. Untuk mengetahui ada tidaknya infeksi ini, biasanya saat kontrol pertama kali, ibu hamil diminta melakukan serangkaian pemeriksaan darah.

Toksoplasmosis adalah infeksi yang diakibatkan oleh sejenis parasit *Toksoplasma gondii*, biasanya ditemukan pada kucing dan dapat menyebabkan pertumbuhan janin terhambat, kelainan mata, cacat otak, abortus, atau bayi lahir mati. Sebagian besar kasus infeksi toksoplasmosis tidak menunjukan gejala klinik. Bila ibu atau janin terbukti terinfeksi, dokter dapat memberikan obat-obat antibiotik. Untuk mencegah toksoplasma, ibu hamil jangan sekali-kali membersihkan kotoran kucing atau hewan peliharaan lainnya. Selain itu, masaklah daging hingga betul-betul matang dan cuci semua buah atau sayuran dengan air yang mengalir sebelum dimakan.

HIV (*Human Immunodeficiency Virus*) menyerang sistem kekebalan tubuh manusia. Jika ibu hamil terinfeksi HIV, maka persalinannya harus melalui bedah sesar untuk meminimalkan penularan virus pada bayi. Infeksi lainnya seperti sipilis dan klamidia dapat terinfeksi melalui hubungan seksual atau dari lingkungan yang tidak bersih. Janin pun dapat tertular, namun pemeriksaan dini pada awal kehamilan sangat membantu untuk dicari penanganan selanjutnya oleh dokter.

Rubella atau campak jerman adalah infeksi virus yang masuk melalui saluran napas dengan menunjukkan gejala demam disertai bercak kemerahan pada kulit selama 3 hari, dimulai dari wajah, lalu merambat ke tubuh dan tungkai. Ada kalanya juga disertai pembesaran kelenjar getah bening yang terdapat di belakang telinga, dan badan terasa pegal-pegal pada persendian. Namun sekitar 25-50 % infeksi *Rubella* justru tidak memberikan gejala klinis

yang jelas. Semakin awal ibu hamil terinfeksi, makin berat dan besar kemungkinan terjadi kecacatan. Bentuk kecacatannya antara lain cacat pada mata (katarak, *glukoma*, atau *mikroftalmia*), jantung, hati dan limfa, alat pendengaran, sistem saraf pusat, dan sebagainya. Upaya pencegahan penting dilakukan, yakni melalui pemberian imunisasi MMR (*Mumps-Measless-Rubella*).

Cytomegalovirus (CMV) adalah virus yang gejala terinfeksinya hampir sama dengan serangan flu. Sumber infeksi CMV berasal dari tenggorokan, ludah, lendir mulut rahim, sperma, urin (terutama pada anak-anak). Ibu hamil yang terkena infeksi ini dapat mengakibatkan keguguran spontan, keterbelakangan mental, hidrosefalus, dan mikrosefalus (kepala kecil).

Herpes disebabkan oleh virus herpes simpleks (VHS). VHS pada ibu hamil ditandai benjolan-benjolan kecil berisi cairan di permukaan kulit atau vagina, terasa gatal sampai nyeri dan dapat disertai demam ringan. Janin bisa terinfeksi VHS karena virus ini bisa masuk dalam sirkulasi darah janin melalui plasenta yang menghubungkan ibu dengan janinnya, ataupun melalui selaput ketuban. Jika infeksi terjadi di trimester awal, umumnya janin akan mati yang berakhir dengan keguguran. Sementara di trimester II dan III, akan lahir prematur, yang biasanya disertai kecacatan maupun kematian begitu dilahirkan.

Selain TORCH, infeksi yang dapat menjangkiti ibu hamil diantaranya campak, cacar air, infeksi saluran kemih, demam tifoid, hepatitis, dan tetanus (IBU[8])

2.8.2.4 Gangguan Karena Kondisi Tertentu

Ragam gangguan ini meliputi masalah berat badan yang tidak mengalami kenaikan atau bahkan kegemukan, gangguan seputar tangan dan kaki, varises, gatal-gatal di kulit, keluhan sakit panggul dan perut, masalah gusi dan gigi, serta sembelit.

2.8.2.4.1 Masalah Berat Badan

jika berat badan tidak mengalami kenaikan, maka ada kemungkinan adanya gangguan pada ibu, bisa karena kurangnya gizi yang masuk, sistem pencernaan yang kurang baik, dan faktor lainnya. Berat badan ideal sepanjang 5 bulan pertama harus ada peningkatan 1 kg setiap bulannya dan selanjutnya 2 kg per bulan.

Sedangkan masalah lainnya kelebihan berat badan. Bagi janin, pertambahan berat badan yang berlebihan bisa menghambat pertumbuhannya. Sebab apabila ibunya *odem*/bengkak dan terjadi hipertensi, suplai nutrisi ke janin pun berkurang karena terjadi penyempitan pembuluh darah.

2.8.2.4.2 Gangguan Seputar Tangan Dan Kaki

a. Kesemutan

kesemutan di daerah telapak tangan dan kaki disebabkan antara lain kekurangan vitamin B, gangguan metabolisme tubuh, penekanan ujung-

ujung saraf tepi oleh penumpukan cairan di daerah ujung-ujung jari, dan bisa juga akibat penekanan rahim yang membesar terhadap kumpulan jaringan saraf (*pleksus*) di daerah rongga panggul atau saraf *iskiadikus* (suatu saraf besar yang mengurus persarafan tungkai, dimulai dari paha sampai ujung jari). Setiap gangguan pada saraf tersebut beserta cabang-cabangnya akan menimbulkan rasa nyeri, kesemutan, bahkan sampai mati rasa.

b. Kram Betis / jari kaki

Salah satu faktor penyebab kram pada otot adalah makanan dan zat gizi. Yaitu apabila pemasukan zat gizi kurang dan metabolisme zat gizi terganggu atau berubah. Zat gizi yang berhubungan dengan kram otot adalah kalsium, sodium, magnesium. Untuk menghindari gejala ini maka disarankan untuk mengkonsumsi makanan yang mengandung zat gizi yang telah disebutkan.

c. Kaki bengkak

Odem atau kaki bengkak pada kehamilan bisa disebabkan faktor fisiologis maupun patologis. Yang patologis, patut dicurigai ibu mengalami preeklampsia, terlebih jika tangan dan wajahnya pun bengkak. Yang fisiologis disebabkan pembuluh darah balik / vena tertekan akibat kehamilan yang membasar, hingga terjadilah bendungan yang menyebabkan darah tidak mengalir selancar biasanya.

2.8.2.4.3 Varises

varises bisa muncul sejak kehamilan trimester I, tergantung sebelumnya sudah ada varises atau tidak. Varises saat hamil akan menambah parah jika sebelumnya ibu sudah terkena. Varises yang muncul saat hamil diakibatkan kehamilan itu sendiri, sebab ada perubahan pada fungsi dan alat tubuh, selain perubahan hormonal. Varises ini sering muncul pada wanita hamil diatas usia 40 tahun, telah melahirkan anak lebih dari 2, dan pada kehamilan kembar.

2.8.2.4.4 Gatal-gatal di Kulit

Kebanyakan rasa gatal yang dialami saat hamil disebabkan *pruritus gravidarum* dan terjadinya bisa di seluruh badan atau setempat. Rasa gatal yang menyeluruh diperkirakan terjadi karena munculnya penumpukan *bilirubin* dan asam empedu yang ringan. Ini terjadi karena perubahan kadar hormon *estrogen* dan *progestin* dalam tubuh ibu hamil. Jika ibu rentan, pengeluaran *bilirubin* dan asam empedu dari hati jadi terganggu, seperti terbendung, hingga kadarnya dalam darah jadi agak meninggi. Inilah yang menimbulkan gatal-gatal di kulit.

2.8.2.4.5 Keluhan Sakit Panggul dan Perut

Semakin tua usia kehamilan, ibu sering mengalami sakit panggul dan perut bagian bawah. Namun masing-masing penyebabnya berbeda. Sakit panggul lebih dikarenakan faktor bayi dan sering dialami usia kehamilan tua, terutama menjelang kelahiran. Bisa jadi ukuran bayi sangat besar sehingga beban ibu bertambah berat. Atau, kepala bayi masuk ke rongga panggul sehingga tulang panggul meregang dan ini menimbulkan rasa sakit pada panggul ibu. Hal lain yang bisa menjadi penyebabnya adalah kemungkinan posisi ibu kurang benar saat bangun, hampir tergelincir atau karena memakai sepatu hak tinggi.

2.8.2.4.6 Masalah Gusi dan Gigi

Karena pengaruh hormon kehamilan, pada gusi tumbuh *epulis* atau benjolan di sela-sela gigi, hingga gusi pun gampang berdarah ketika ibu sikat gigi. Problema ini (*epulis gravidarum*) merupakan masalah pada gusi yang hanya terjadi saat hamil. Biasanya *epulis gravidarum* terjadi pada trimester I dan bisa berlanjut hingga bayi lahir, baru kemudian hilang dengan sendirinya.

2.8.2.4.7 Sembelit

Ibu hamil sering mengalami kesulitan Buang Air Besar. Selain karena disebabkan perubahan hormonal, juga disebabkan pola makan yang salah dan kurang minum. Walaupun sembelit, tapi jika tidak segera diatasi akan menjadi masalah yang merepotkan, apalagi gangguan sembelit ini dapat terjadi sepanjang kehamilan, dan biasanya semakin terasa di masa akhir kehamilan karena kepala bayi mulai turun dan menekan usus (IBU[8]).

2.9 Object Oriented

Object oriented merupakan paradigma dalam rekayasa *software* yang didasarkan pada objek dan kelas. *Object oriented* memandang *software* bagian per bagian, dan menggambarkan suatu bagian dalam satu objek. Satu objek dalam sebuah model merupakan suatu fokus selama dalam proses analisis, desain dan implementasi dengan menekankan pada *state*, prilaku (*behavior*), dan interaksi objek-objek dalam model tersebut (ASH[1]).

2.9.1 *Object Oriented Analysis and Design* (OOAD)

Object oriented analysis adalah metode analisis yang memeriksa keperluan (*requirements*) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup permasalahan (ASH[1]).

Object oriented design adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

2.9.2 Objek

Objek (*object*) adalah benda, secara fisik ataupun konseptual. *Hardware, software,* dokumen dan manusia adalah beberapa contoh dari objek. Sebuah objek memiliki keadaan sesaat (*state*) dan prilaku (*behavior*) (ASH[1]).

State adalah kondisi objek yang menggambarkan objek tersebut, sedangkan *behaviour* adalah suatu definisi tindakan dan reaksi suatu objek.

2.9.3 Kelas

Kelas (*class*) adalah definisi umum untuk himpunan objek sejenis. Kelas menetapkan spesifikasi prilaku (*behavior*) dan atribut objek-objek tersebut. *Class* adalah abstraksi dari entitas dalam dunia nyata. Objek adalah contoh (*instance*) dari sebuah kelas (ASH[1]).

2.10. UML (*Unified Modeling Language*)

UML merupakan model yang dapat digunakan secara luas dalam pemodelan bisnis, pemodelan perangkat lunak dari semua fase pembentukan dan semua tipe

sistem, dan pemodelan secara umum dari berbagai pembentukan / konstruksi yang memiliki dua perilaku yaitu baik statis maupun dinamis (MUS[12]).

2.10.1 Sejarah UML

UML dimulai secara resmi pada oktober 1994, ketika Rumbaugh bergabung dengan Booch pada Relational Software Corporation. Proyek ini memfokuskan pada penyatuan metode Booch dan OMT. UML versi 0.8 merupakan metode penyatuan yang dirilis pada bulan Oktober 1995. Dalam waktu yang sama, Jacobson bergabung dengan Relational dan cakupan dari UML semakin luas sampai diluar perusahaan OOSE. Dokumentasi UML versi 0.9 akhirnya dirilis pada bulan Juni 1996. Meskipun pada tahun 1996 ini melihat dan menerima feedback dari komunitas Software Engineering . Dalam waktu tersebut, menjadi lebih jelas bahwa beberapa organisasi perangkat lunak melihat UML sebagai strategi dari bisnisnya. Kemudian dibangunlah UML Consortium dengan beberapa organisasi yang akan menyumbangkan sumber dayanya untuk bekerja, mengembangkan, dan melengkapi UML (SOF[18]).

Di sini beberapa partner yang berkontribusi pada UML 1.0, diantaranya Digital Equipment Corporation, Hewlett-Packard, I-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Relational, Texas Instruments dan Unisys. Dari kolaborasi ini dihasilkan UML 1.0 yang merupakan bahasa pemodelan yang ditetapkan secara baik, kuat, dan cocok

untuk lingkungan masalah yang luas. UML 1.0 ditawarkan menjadi standarisasi dari Object Management Group (OMG). Dan pada Januari 1997 dijadikan sebagai standar bahasa pemodelan.

Antara Januari–Juli 1997 gabungan group tersebut memperluas kontribusinya sebagai hasil respon dari OMG dengan memasukkan Adersen Consulting, Ericsson, ObjectTimeLimited, Platinum Technology, Ptech, Reich Technologies, Softeam, Sterling Software dan Taskon. Revisi dari versi UML (versi 1.1) ditawarkan kepada OMG sebagai standarisasi pada bulan Juli 1997. Dan pada bulan September 1997, versi ini dierima oleh OMG Analysis dan Design Task Force (ADTF) dan OMG ArchitectureBoard. Dan Akhirnya pada Juli 1997 UML versi 1.1 menjadi standarisasi.

Pemeliharaan UML terus dipegang oleh OMG Revision Task Force (RTF) yang dipimpin oleh Cris Kobryn. RTP merilis editorial dari UML 1.2 pada Juni 1998. Dan pada tahun 1998 RTF juga merilis UML 1.3 disertai dengan user guide dan memberikan technical cleanup.

2.10.2 Tujuan Penggunaan UML

Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi *object*. Dan menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin (SOF[18]).

2.10.3 Gambaran dari UML

UML dapat dimanfaatkan untuk berbagai macam kebutuhan, diantaranya untuk melakukan pemodelan, visualisasi, spesifikasi, konstruksi dan dokumentasi.

2.10.3.1UML sebagai Bahasa Pemodelan

UML merupakan bahasa pemodelan yang memiliki pembendaharaan kata dan cara untuk mempresentasikan secara fokus pada konseptual dan fisik dari suatu sistem. Contoh untuk sistem *software* yang *intensive* membutuhkan bahasa yang menunjukkan pandangan yang berbeda dari arsitektur sistem, ini sama seperti menyusun/mengembangkan *software development life cycle*. Dengan UML akan memberitahukan kita bagaimana untuk membuat dan membaca bentuk model yang baik, tetapi UML tidak dapat memberitahukan model apa yang akan dibangun dan kapan akan membangun model tersebut. Ini merupakan aturan dalam *software development process*.

2.10.3.2UML sebagai bahasa Visualizing

UML tidak hanya merupakan rangkaian simbol grafikal, cukup dengan tiap simbol pada notasi UML merupakan penetapan semantik yang baik. Dengan cara ini, satu pengembang dapat menulis model UML dan pengembang lain atau perangkat yang sama lainnya dapat mengartikan bahwa model tersebut tidak ambigu. Hal ini akan mengurangi error yang terjadi karena perbedaan bahasa dalam komunikasi model konseptual dengan model

lainnya. UML menggambarkan model yang dapat dimengerti dan dipresentasikan ke dalam model tekstual bahasa pemograman. Contohnya kita dapat menduga suatu model dari sistem yang berbasis web tetapi tidak secara langsung dipegang dengan mempelajari *code* dari sistem. Dengan model UML maka kita dapat memodelkan suatu sistem web tersebut dan direpresentasikan ke bahasa pemrograman. UML merupakan suatu model eksplisit yang menggambarkan komunikasi informasi pada sistem. Sehingga kita tidak kehilangan informasi *code* implementasi yang hilang dikarenakan *developer* memotong *coding* dari implementasi.

2.10.3.3UML sebagai bahasa Specifying

Maksudnya membangun model yang sesuai, tidak ambigu dan lengkap. Pada faktanya UML menunjukan semua spesifikasi keputusan analisis, desain dan implementasi yang penting yang harus dibuat pada saat pengembangan dan penyebaran dari sistem *software* intensif.

2.10.3.4UML sebagai bahasa *Constructing*

UML bukan bahasa pemograman visual, tetapi model UML dapat dikoneksikan secara langsung pada bahasa pemograman visual. Maksudnya membangun model yang dapat dimapping ke bahasa pemograman seperti java,

C++, VB atau tabel pada *database relational* atau penyimpanan tetap pada *database* berorientasi *object*.

2.10.3.5UML sebagai bahasa Documenting

Maksudnya UML menunjukan dokumentasi dari arsitektur sistem dan detail dari semuanya.

UML hanya memberikan bahasa untuk memperlihatkan permintaan dan untuk tes. UML menyediakan bahasa untuk memodelkan aktifitas dari perencanaan project dan manajemen pelepasan (*release management*).

2.10.4 Bagian-Bagian dari UML

Bagian-bagian utama dari UML adalah *view, diagram, model element, dan general mechanism*.

2.10.4.1View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis *view* dalam UML antara lain: *use case view*, *logical view*, *component view*, *concurrency view*, dan *deployment view*.

2.10.4.1.1 Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa user atau sistem lainnya.

View ini digambarkan dalam *use case diagrams* dan kadang-kadang dengan *activity diagrams*. View ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

2.10.4.1.2 Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class*, *object*, dan *relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu. View ini digambarkan dalam *class diagrams* untuk struktur statis dan dalam *state*, *sequence*, *collaboration*, dan *activity diagram* untuk model dinamisnya. View ini digunakan untuk perancang (*designer*) dan pengembang (*developer*).

2.10.4.1.3 Component view

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur

dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya. View ini digambarkan dalam component view dan digunakan untuk pengembang (*developer*).

2.10.4.1.4 Concurrency view

Membagi sistem ke dalam proses dan prosesor. View ini digambarkan dalam diagram dinamis (*state*, *sequence*, *collaboration*, dan *activity diagrams*) dan diagram implementasi (*component* dan *deployment diagrams*) serta digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

2.10.4.1.5 Deployment view

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya. View ini digambarkan dalam *deployment diagrams* dan digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

2.10.4.2Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem.

Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika digambarkan biasanya dialokasikan untuk view tertentu. Adapun jenis diagram antara lain :

2.10.4.2.1 Use Case Diagram

Menggambarkan sejumlah external actors dan hubungannya ke use case yang diberikan oleh sistem. Use case adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari use case symbol namun dapat juga dilakukan dalam *activity diagrams*. Use case digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

2.10.4.2.2 Class Diagram

Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem.

Class dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu class tergantung/menggunakan class yang lain), *specialed* (satu class merupakan spesialisasi dari class lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram*.

2.10.4.2.3 State Diagram

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

2.10.4.2.4 Sequence Diagram

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

2.10.4.2.5 Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

2.10.4.2.6 Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

2.10.4.2.7 Component Diagram

Menggambarkan struktur fisik kode dari komponent. Komponent dapat berupa *source code*, komponent *biner*, atau *executable component*. Sebuah komponent berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.

2.10.4.2.8 Deployment Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

2.10.5 Area Penggunaan UML

UML digunakan paling efektif pada domain seperti :

1. Sistem Informasi Perusahaan
2. Sistem Perbankan dan Perekonomian
3. Bidang Telekomunikasi
4. Bidang Transportasi
5. Bidang Penerbangan
6. Bidang Perdagangan

Namun UML tidak terbatas untuk pemodelan *software*. Pada faktanya UML banyak untuk memodelkan sistem non *software* seperti aliran kerja pada sistem perundangan, desain *hardware*, dan lain-lain.

2.11 Sistem Representasi Pengetahuan Berbasis Web

Berdasarkan beberapa pengertian sebelumnya, sistem representasi pengetahuan berbasis web dapat diartikan sebagai kumpulan dari elemen-elemen program aplikasi yang mewakili segala sesuatu yang diketahui berkenaan dengan hal tertentu yang dapat diakses melalui jaringan internet.

2.12 Skala Pengukuran Statistik

Data yang akan dianalisis dengan statistik harus berbentuk angka-angka. Angka-angka yang digunakan dalam analisis statistik pada dasarnya dapat

dikategorikan menjadi 4 yaitu skala nominal, skala ordinal, skala interval, dan skala ratio.

Skala nominal yaitu angka yang tidak mempunyai arti hitung. Angka yang diterapkan hanya merupakan simbol / tanda dari objek yang akan dianalisis. Skala ordinal yaitu suatu skala yang sudah mempunyai daya pembeda, tetapi perbedaan antara angka yang satu dengan angka yang lainnya tidak konstan. Skala interval yaitu skala yang mempunyai rentangan konstan antara tingkat satu dengan yang aslinya, tetapi tidak mempunyai angka 0 mutlak. Sedangkan skala ratio adalah suatu skala yang mempunyai rentangan konstan dan mempunyai angka 0 mutlak (SUG[19]).