

BAB II

LANDASAN TEORI

2.1. Bengkel

Pada kondisi tertentu, kendaraan bermotor memerlukan perawatan atau perbaikan. Perawatan dan perbaikan kendaraan harus dilakukan agar umur pakai kendaraan lebih panjang atau paling tidak sama dengan umur pakai yang telah diprediksikan dan dirancang oleh pabrik pembuat. Meskipun demikian, perawatan dan perbaikan kendaraan bukan merupakan pekerjaan yang mudah. Hal tersebut memerlukan pengetahuan khusus.

Untuk memperoleh pengetahuan tersebut, tentu saja dibutuhkan kemauan dan waktu. Namun sebagian besar pemilik kendaraan bermotor biasanya merasa dirinya tidak memiliki kedua hal tersebut. Berdasarkan hal tersebut, terbuka peluang bagi pihak lain yang memiliki keahlian dan peralatan kerja di bidang kendaraan bermotor (otomotif) untuk membuka usaha perbengkelan. Terjadilah transaksi antara orang yang membutuhkan perawatan atau perbaikan di bidang otomotif dan mereka yang memiliki keahlian serta peralatan di bidang tersebut. Hal ini dilakukan di bengkel otomotif.

Bengkel mobil diklasifikasikan berdasarkan dua kriteria, yaitu fasilitas pelayanan dan skala usaha yang dijalankan (Meliputi jumlah tenaga kerja, modal, dan kapasitas kerja).

2.1.1. Fasilitas

Berdasarkan fasilitas pelayanan, bengkel mobil dapat dibedakan menjadi empat, yaitu:

A. Bengkel Dealer

Bengkel dealer merupakan bagian dari *dealer* otomotif yang memberikan pelayanan purnajual kepada konsumen. Bengkel jenis ini biasanya hanya melayani kendaraan dengan merek tertentu yang dijual di dealer tersebut. Pelayanan yang ditawarkan oleh bengkel *dealer* meliputi perawatan rutin hingga perbaikan yang memerlukan penggantian suku cadang. Bengkel jenis ini biasanya terdiri dari beberapa bagian khusus yang memberikan pelayanan perawatan atau perbaikan tertentu pada komponen mobil (*mesin, balancing, body repair*, dan sebagainya). Oleh karena itu, teknisi yang bekerja di bengkel ini juga memiliki spesialisasi tertentu dan dilengkapi peralatan yang mendukung pekerjaan.

B. Bengkel Pelayanan Umum

Bengkel pelayanan umum merupakan bengkel independen yang mampu melakukan perawatan dan perbaikan beberapa komponen mobil. Bengkel Semacam ini dapat dipandang sebagai beberapa bengkel khusus yang menggabungkan diri menjadi sebuah bengkel yang lebih besar. Berbeda dengan bengkel *dealer*, bengkel ini bukan merupakan bagian dari *dealer* otomotif. Oleh karena itu, pelayanan yang diberikan bengkel ini tidak ditujukan untuk pelayanan purnajual sebuah produk otomotif. Selain

itu, bengkel pelayanan umum biasanya memberikan pelayanan perawatan dan perbaikan untuk berbagai merek kendaraan.

C. Bengkel Pelayanan Khusus

Bengkel pelayanan khusus adalah bengkel otomotif yang memiliki spesialisasi dalam hal perawatan dan perbaikan salah satu elemen mobil. Sebagai contoh bengkel reparasi bodi, radiator, AC, *spooring* dan *balancing*, dan sebagainya. Spesialisasi yang dilakukan oleh bengkel tersebut menuntut peralatan khusus sesuai dengan jenis operasi yang akan dilakukan. Bagian terpenting dari bengkel pelayanan khusus adalah spesialisasi keahlian tenaga kerja sesuai dengan kualifikasi pekerjaan yang akan dilakukan.

D. Bengkel Unit Keliling

Bengkel unit keliling memberikan pelayanan berupa perbaikan yang dilakukan di lokasi mobil konsumen. Bengkel jenis ini terdiri dari beberapa buah mobil *van* dan derek yang secara periodik berpatroli di daerah tertentu, atau kadang-kadang menerima panggilan untuk memberi pelayanan kepada konsumen.

2.1.2. Skala Usaha

Berdasarkan skala usaha yang dijalankannya, bengkel mobil dapat diklasifikasikan menjadi dua, yaitu:

A. Bengkel Kecil

Bengkel kecil adalah bengkel yang meliputi bengkel skala garasi rumah dengan satu sampai lima orang pekerja, hingga bengkel permanen dengan tenaga kerja hingga 19 orang (definisi Biro Pusat Statistik tentang Usaha Kecil).

B. Bengkel Besar

Biro Pusat Statistik mengklasifikasikan usaha besar sebagai usaha yang mempekerjakan lebih dari 20 orang. Berdasarkan hal tersebut, sebuah bengkel dapat diklasifikasikan sebagai bengkel besar apabila memiliki pegawai lebih dari 20 orang. Bengkel besar dapat diklasifikasikan berdasarkan aset yang dimilikinya. Biasanya, orang-orang juga mengklasifikasikan bengkel besar apabila dilengkapi peralatan canggih sebagai peralatan kerjanya.

2.2. Sistem Pakar

Menurut Kusrini (2006) sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tertentu.

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan yang dimaksud antara lain: pembuatan keputusan, pemandu pengetahuan, pembuatan desain, perencanaan, perkiraan, pengaturan, pengendalian, diagnosis, perumusan, penjelasan,

pemberian nasihat dan pelatihan. Sistem pakar juga dapat berfungsi sebagai asisten yang pandai dari seorang pakar.

Alasan mendasar mengapa sistem pakar dikembangkan untuk menggantikan seorang pakar adalah:

1. Dapat menyediakan kepakaran setiap waktu dan di berbagai lokasi.
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar akan pensiun atau pergi.
4. Menghadirkan/menggunakan jasa seorang pakar memerlukan biaya yang mahal.
5. Pakar dibutuhkan juga pada lingkungan yang tidak bersahabat.

Ciri-ciri sistem pakar (Kusrini, 2006) adalah sebagai berikut:

1. Terbatas pada bidang yang spesifik.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan *rule* tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. *Output* bersifat anjuran.
7. *Output* tergantung dari dialog dengan *user*.
8. *Knowledge base* dan *inference engine* terpisah.

Keuntungan pemakaian sistem pakar (Kusrini, 2006) yaitu:

1. Membuat seseorang yang awam dapat bekerja seperti layaknya seorang pakar.
2. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti
3. Meningkatkan *output* dan produktivitas.
4. Meningkatkan kualitas.
5. Sistem pakar menyediakan nasihat yang konsisten dan dapat mengurangi tingkat kesalahan.
6. Membuat peralatan yang kompleks lebih mudah dioperasikan karena sistem pakar dapat melatih pekerja yang tidak berpengalaman.
7. Andal (*reliability*).
8. Sistem pakar tidak dapat lelah atau bosan, konsisten dalam memberi jawaban dan selalu memberikan perhatian penuh.
9. Memiliki kemampuan dalam memecahkan suatu permasalahan yang kompleks.
10. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan pakar, dapat diperoleh dan dipakai di mana saja.

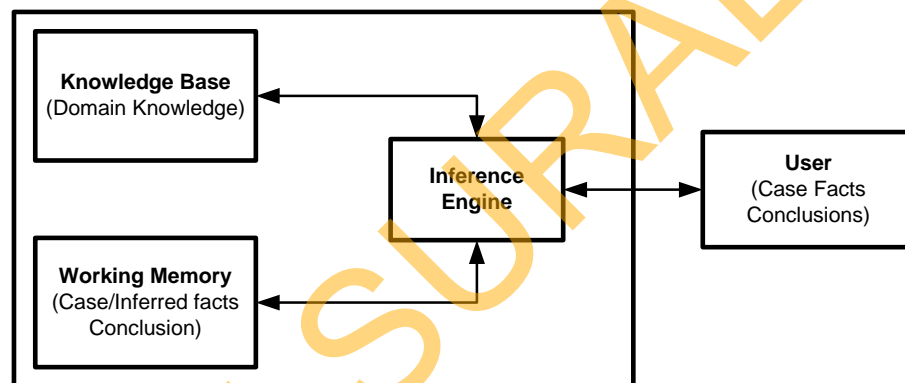
Menurut Gonzales dan Dankel (1993), terdapat beberapa kelemahan sistem pakar, antara lain:

1. Jawaban yang diberikan tidak selalu benar, seperti halnya pakar yang tidak selalu benar.
2. Pengetahuan terbatas pada keahlian pakar.

3. Pengetahuan akan kebiasaan umum sulit direpresentasikan pada sistem, sehingga sistem kurang sadar akan hal lazim (kebiasaan yang sudah umum).

2.2.1 Komponen Sistem Pakar

Menurut Irawan (2007), sistem pakar terdiri atas tiga komponen utama, yaitu *knowledge base*, *working memory* dan *inference engine*. Struktur dasar sistem pakar dapat dilihat pada Gambar 2.1.



Gambar 2.1 Struktur dasar sistem pakar (Irawan, 2007: 7)

1. Knowledge Base

Knowledge base adalah bagian dari sebuah sistem pakar yang mengandung/menyimpan pengetahuan. Pengetahuan ini merupakan representasi pengetahuan dari seorang pakar. *Knowledge base* yang dikandung oleh sebuah sistem pakar berbeda, tergantung pada bidang kepakaran dari sistem yang dibangun. *Knowledge base* direpresentasikan dalam berbagai macam bentuk, salah satunya adalah dalam bentuk sistem berbasis aturan (*ruled-based system*).

2. Working Memory

Working memory mengandung/menyimpan fakta-fakta (baik yang dimasukkan oleh pengguna maupun fakta baru) yang ditemukan selama proses konsultasi dengan sistem pakar. Selama proses konsultasi pengguna umum memasukkan fakta-fakta yang dibutuhkan, kemudian sistem akan mencari padanan fakta tersebut dengan informasi yang ada dalam *knowledge base* untuk menghasilkan fakta baru. Fakta baru yang merupakan hasil kesimpulan dari sistem akan dimasukkan ke dalam *working memory*.

3. Inference Engine

Inference engine adalah bagian dari sistem pakar yang bertugas mencari padanan antara fakta yang ada di dalam *working memory* dengan fakta-fakta tentang pengetahuan tertentu dari pakar yang ada di dalam *knowledge base*. Selanjutnya *inference engine* akan menarik kesimpulan dari masalah yang diajukan kepada sistem. Terdapat dua metode yang dapat digunakan untuk mencari kesimpulan, yaitu *forward chaining* dan *backward chaining*.

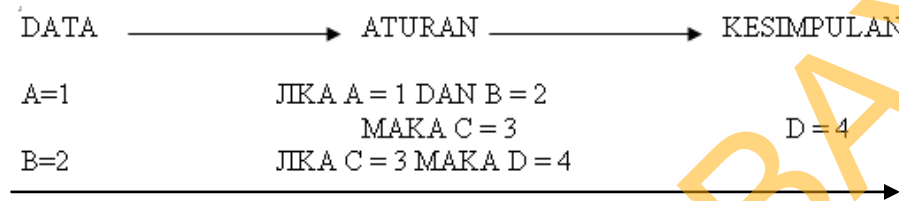
4. User (Kusrini, 2006)

User adalah seseorang yang berkonsultasi dengan sistem untuk mendapatkan saran / kesimpulan yang disediakan oleh pakar.

2.2.2 Forward Chaining

Metode *forward chaining (data driven)* adalah suatu metode yang menghasilkan kesimpulan dari seperangkat data yang diketahui (Irawan, 2007). Menurut Kusrini (2006), runut maju (*forward chaining*) berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode *forward chaining*, data digunakan

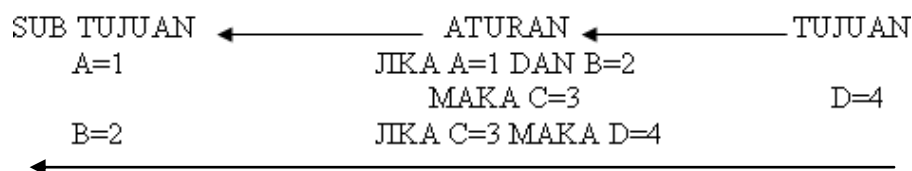
untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai menemukan suatu hasil. Cara kerja metode *forward chaining* dapat dilihat pada Gambar 2.2.



Gambar 2.2 *Forward Chaining* (Kusrini, 2006:36)

2.2.3 Backward Chaining

Metode *backward chaining* (*goal driven*) adalah suatu metode yang berawal dari proses memilih beberapa kesimpulan yang mungkin dan mencoba membuktikan kesimpulan tersebut dari bukti-bukti yang ada (Irawan, 2007). Menurut Kusrini (2006), runut balik (*backward chaining*) merupakan metode penalaran kebalikan dari runut maju. Dalam metode *backward chaining*, penalaran dimulai dengan tujuan yang merunut balik ke jalur yang akan mengarahkan ke tujuan tersebut. Cara kerja metode *backward chaining* dapat dilihat pada Gambar 2.3.



Gambar 2.3 *Backward Chaining* (Kusrini, 2006:37)

2.2.4 Verifikasi

Verifikasi adalah suatu proses yang bertujuan untuk memastikan bahwa sistem telah berlaku dalam kondisi yang ditetapkan. Verifikasi terdiri dari dua proses. Proses pertama adalah memeriksa pelaksanaan suatu sistem secara spesifik. Proses kedua adalah memeriksa konsistensi dan kelengkapan dari basis pengetahuan (*knowledge base*). Verifikasi dijalankan ketika ada penambahan atau perubahan pada *rule*, karena *rule* sudah terdapat pada sistem. Tujuan verifikasi adalah untuk memastikan adanya kecocokkan antara sistem dengan apa yang sistem kerjakan dan juga untuk memastikan bahwa sistem terbebas dari kesalahan. Berikut ini adalah beberapa metode pemeriksaan aturan-aturan dalam suatu basis pengetahuan (Gonzales dan Dankel, 1993).

1. Redundant rules

Dikatakan *redundant rules* jika dua *rule* atau lebih mempunyai *premise* dan *conclusion* yang sama.

Contoh :

*Rule 1: If the humidity is high and the temperature is hot
Then there will be thunderstorms*

*Rule 2: If the temperature is hot and the humidity is high
Then there will be thunderstorms*

2. Conflicting rules

Conflicting rules terjadi ketika dua *rule* atau lebih mempunyai *premise* yang sama tetapi *conclusion* yang berbeda.

Contoh :

Rule 1: If the temperature is hot and the humidity is high

Then there will be sunshine

Rule 2: If the temperature is hot and the humidity is high

Then there will be no sunshine

3. Subsumed rules

Suatu keadaan dapat dikatakan *subsumed rules* jika *rule* mempunyai *constraint* yang lebih atau kurang tetapi mempunyai *conclusion* yang sama.

Contoh :

Rule 1: If the temperature is hot and the humidity is high

Then there will be thunderstorms

Rule 2: If the temperature is hot

Then there will be thunderstorms

4. Circular rules

Circular rules merupakan proses perulangan dari suatu *rule* karena *premise* dari salah satu *rule* merupakan *conclusion* dari *rule* yang lain, atau kebalikannya.

Contoh :

Rule 1: If X and Y are brothers

Then X and Y have the same parents

Rule 2: If X and Y have the same parents

Then X and Y are brothers

5. Unnecessary IF condition

Unnecessary IF terjadi ketika dua *rule* atau lebih mempunyai *conclusion* yang sama tetapi salah satu dari *rule* tersebut mempunyai *premise* yang tidak perlu dikondisikan dalam *rule* karena tidak mempunyai pengaruh apapun.

Contoh :

Rule 1: If the patient has the pink spots and the patient has a fever
Then the patient has measles

Rule 2: If the patient has the pink spots and the patient does not have fever
Then the patient has measles

6. Dead-end rules

Dead-end rules adalah tindakan yang tidak mempengaruhi *conclusion* dan tidak digunakan oleh *rule* yang lain untuk menghasilkan suatu *conclusion*.

Contoh :

Rule 1: If the gauge reads empty
Then the gas tank is empty

7. Missing Rules

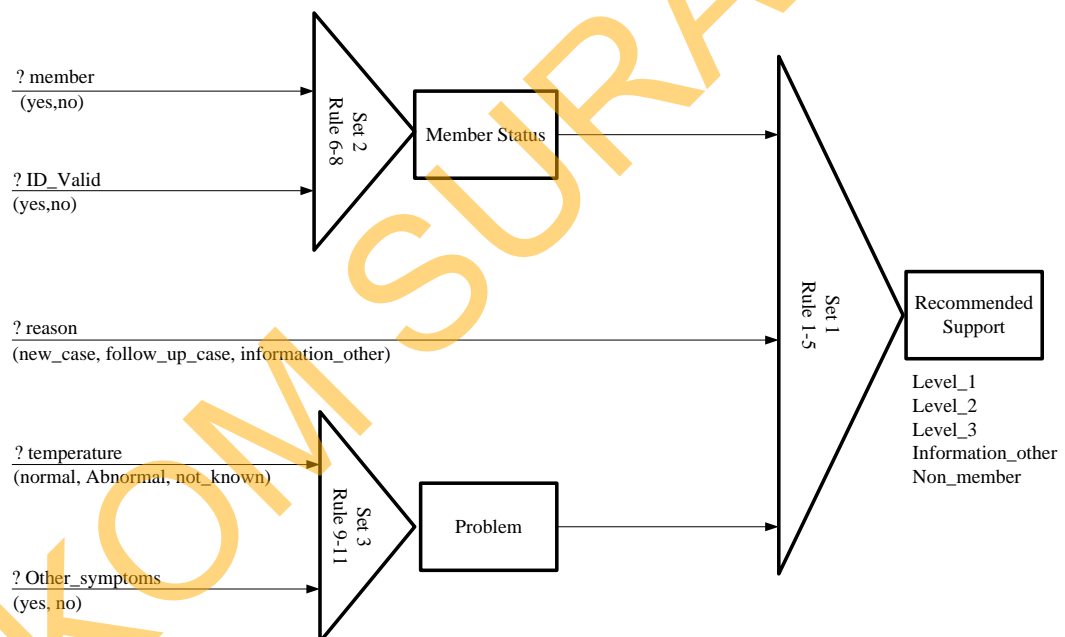
Missing rules merupakan suatu aturan yang ditandai dengan fakta yang tidak pernah digunakan dalam *inference process*.

8. Unreachable Rules

Unreachable rules merupakan suatu aturan yang *premisnya* tidak akan pernah cocok dengan keadaan sistem, baik karena *missing rule* atau kurangnya data masukan.

2.2.5 Dependency Diagram

Setelah diketahui urutan kerja sistem dalam mencari keputusan dari diagram blok, langkah selanjutnya adalah membuat *dependency diagram* (diagram ketergantungan). Menurut Dologite (1993) *dependency diagram* adalah suatu relasi yang menunjukkan hubungan atau ketergantungan antara inputan jawaban, aturan-aturan (*rule*), nilai-nilai dan direkomendasikan ke dalam sistem berbasis pengetahuan. Contoh *dependency diagram* dapat dilihat pada Gambar 2.5



Gambar 2.4 Dependency Diagram (Dologite, 1993:23)

2.2.6 Decision Table

Setelah diagram ketergantungan dibuat, langkah berikutnya adalah membuat *Decision table*. Menurut Dologite (1993), *decision table* diperlukan untuk menunjukkan hubungan timbal balik antara nilai-nilai pada hasil fase antara atau rekomendasi akhir sistem. Sebelum membuat *decision table*, diperlukan informasi mengenai jumlah *rule* yang akan dibuat. Jumlah *rule* dapat diketahui dari jumlah

parameter dan kemungkinan jawaban yang ada, seperti pada Gambar 2.6. Sebagai contoh dari pembuatan *decision table* dapat dilihat pada Tabel 2.1.

Condition : Member_Status (ok, not_ok)	=2
Reason (new_case, follow_up_case, information_other)	=3
Problem (serious, non_serious)	=2

Gambar 2.5 Penentuan Jumlah Rule (Dologite, 1993:24)

Tabel 2.1 *Decision Table* (Dologite, 1993:24)

Rule	Member Status	Reason	Problem	Concluding Recommendation for support level
A1	Ok	New_case	Serious	Level_1
A2	Ok	New_case	Non_serious	Level_2
A3	Ok	Follow_up_case	Serious	Level_1
A4	Ok	Follow_up_case	Non_serious	Level_3
A5	Ok	Information_other	Serious	Information_other
A6	Ok	Information_other	Non_serious	Information_other
A7	Not_ok	New_case	Serious	Non_member
A8	Not_ok	New_case	Non_serious	Non_member
A9	Not_ok	Follow_up_case	Serious	Non_member
A10	Not_ok	Follow_up_case	Non_serious	Non_member
A11	Not_ok	Information_other	Serious	Non_member
A12	Not_ok	Information_oher	Non_serious	Non_member

2.2.7 Reduced Decision Table

Reduced decision table adalah pembuatan tabel yang nilai-nilainya didapat dari mereduksi *decision table*. Setelah didapatkan nilai dari *decision table*,

akan direduksi untuk mendapatkan nilai dari kondisi terakhir. Sebagai contoh dari *reduced decision table* dapat dilihat pada Tabel 2.2

Tabel 2.2 *Reduced Decision Table* (Dologite, 1993:24)

Rule	Member Status	Reason	Problem	Concluding Recommendation for support level
B1	Ok	New_case	Serious	Level_1
B2	Ok	New_case	Non_serious	Level_2
B3	Ok	Follow_up_case	Serious	Level_1
B4	Ok	Follow_up_case	Non_serious	Level_3
B5	Ok	Information_other	-	Information_other
B6	Not_ok	-	-	Non_member

2.3. Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan. (Sommerville, 2003). Pada definisi ini, ada dua istilah kunci yaitu:

1. Disiplin rekayasa, perekayasa membuat suatu alat bekerja. Menerapkan teori, metode, dan alat bantu yang sesuai, selain itu mereka menggunakannya dengan selektif dan selalu mencoba mencari solusi terhadap permasalahan, walaupun tidak ada teori atau metode yang mendukung. Perekayasa juga menyadari bahwa mereka harus bekerja dalam batasan organisasi dan keuangan, sehingga mereka berusaha mencari solusi dalam batasan-batasan ini.

2. Semua aspek produksi perangkat lunak, rekayasa perangkat lunak tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak tetapi juga dengan kegiatan seperti manajemen proyek perangkat lunak dan pengembangan alat bantu, metode, dan teori untuk mendukung produksi perangkat lunak.

Secara umum, rekayasa perangkat lunak memakai pendekatan sistematis dan terorganisasi terhadap pekerjaan mereka karena cara ini seringkali paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Namun demikian, rekayasa ini sebenarnya mencakup masalah pemilihan metode yang paling sesuai untuk satu set keadaan dan pendekatan yang lebih kreatif, informal terhadap pengembangan yang mungkin efektif pada beberapa keadaan. Pengembangan informal sangat cocok untuk pengembangan sistem *e-commerce web* membutuhkan gabungan keahlian perangkat lunak dan perancangan grafis.

2.3.1. Proses Perangkat Lunak

Proses perangkat lunak adalah serangkaian kegiatan-kegiatan dan hasil-hasil relevannya yang menghasilkan perangkat lunak. Kegiatan-kegiatan ini sebagian besar dilakukan perreayasa perangkat lunak. Ada empat kegiatan proses dasar yang umum bagi seluruh kegiatan proses perangkat lunak. Kegiatan-kegiatan ini adalah :

1. Spesifikasi perangkat lunak, fungsionalitas perangkat lunak dan batasan kemampuan operasinya harus didefinisikan

2. Pengembangan perangkat lunak, perangkat lunak yang memenuhi spesifikasi tersebut harus diproduksi.
3. Validasi perangkat lunak, perangkat lunak harus divalidasi untuk menjamin bahwa perangkat lunak melakukan apa yang diinginkan oleh pelanggan.
4. Evolusi perangkat lunak, perangkat lunak harus berkembang untuk memenuhi kebutuhan pelanggan yang berubah-ubah.

Proses perangkat lunak yang berbeda mengatur kegiatan ini dengan cara berbeda dan dijelaskan dengan tingkat kerincian yang berbeda pula. Waktu kegiatan bervariasi, sebagaimana hasilnya. Pengaturan yang berbeda dapat menggunakan proses yang berbeda untuk menghasilkan produk dengan jenis yang sama. Namun demikian, untuk beberapa jenis aplikasi tertentu, beberapa proses lebih sesuai dari yang lainnya jika digunakan proses yang tidak sesuai, maka kualitas penggunaan produk perangkat lunak yang akan dikembangkan tersebut mungkin berkurang.

2.3.2. Model Proses Perangkat Lunak

Model proses pengembangan perangkat lunak adalah sebagai berikut :

1. Model air terjun (waterfall). Model ini mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi dan evolusi, dan merepresentasikannya sebagai fase-fase proses yang berbeda seperti

spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian dan seterusnya.

2. Pengembangan evolusioner. Pendekatan ini berhimpitan dengan kegiatan spesifikasi, pengembangan, dan validasi. Suatu sistem awal dikembangkan dengan cepat dari spesifikasi abstrak. Sistem ini kemudian diperbaiki dengan masukan dari pelanggan untuk menghasilkan sistem yang memuaskan bagi kebutuhan pelanggan.
3. Pengembangan sistem formal. Pendekatan ini didasarkan atas pembuatan spesifik sistem matematis dan pentransformasian spesifikasi, dengan memakai metode matematis untuk membangun program. Verifikasi komponen sistem dilakukan dengan membuat argumen matematis yang disesuaikan dengan spesifikasi.

Pengembangan berdasarkan pemakaian ulang. Pendekatan ini didasarkan atas adanya komponen yang dapat dipakai untuk jumlah yang signifikan. Proses pengembangan sistem terfokus pada integrasi komponen-komponen ini ke dalam suatu sistem dan bukan mengembangkan dari awal.

2.4. Sistem Basis Data

Sistem basis data merupakan sistem yang terdiri atas kumpulan *file* (tabel) yang saling berhubungan (dalam sebuah basis data di sebuah komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan atau

program lain untuk mengakses dan memanipulasi *file-file* (tabel-tabel) tersebut (Fathansyah, 2007:9)

2.4.1. Komponen Basis Data

Menurut Kusri (2007 : 11), komponen-komponen sistem basis data meliputi :

1. Perangkat Keras (Hardware) sebagai pendukung operasi pengolahan data.

Perangkat keras computer adalah semua bagian fisik computer. Contoh dari perangkat keras computer yaitu : mouse, keyboard, monitor, CPU, memori, dan lain-lain

2. Sistem Operasi (*Operating System*) atau perangkat lunak untuk mengolah basis data.

Sistem operasi merupakan suatu software sistem yang bertugas untuk melakukan control dan manajemen hardware serta operasi-operasi dasar sistem, termasuk menjalankan software aplikasi seperti program-program pengolahan kata dan browser web. Secara umum sistem operasi adalah software pada lapisan pertama yang ditaruh pada memori computer pada saat computer dinyalakan.

3. Basis Data (*Database*) sebagai inti dari sistem basis data.
4. *Database Management System* (DBMS)

DBMS adalah *software* yang menangani semua akses ke basis data.

Secara konsep yang terjadi dalam DBMS adalah sebagai berikut :

- a. User melakukan pengaksesan basis data untuk informasi yang diperlukannya menggunakan bahasa manipulasi data, biasanya disebut SQL
- b. DBMS menerima *request* dari user dan menganalisanya.
- c. DBMS memeriksa skema eksternal user, pemetaan eksternal/konseptual, skema konseptual, pemetaan konseptual/internal, dan struktur penyimpanan.
- d. DBMS mengeksekusi operasi-operasi yang diperlukan untuk memenuhi permintaan user.

5. Pemakai (User)

Pemakai merupakan orang atau sistem yang akan mengakses dan merubah isi basis data. Beberapa jenis pemakai basis data yaitu :

- a. Programmer Aplikasi : orang yang mengkodekan aplikasi dengan bahasa pemrograman.
- b. User Mahir : orang yang mampu menggunakan basis data secara langsung dengan menggunakan DBMS.
- c. User Umum/End User : orang yang akan memakai basis data dengan perantara program aplikasi.
- d. User khusus : biasa berupa sistem lain.

6. Aplikasi Lain

Aplikasi lain merupakan *software* yang dibuat untuk memberikan *interface* kepada user sehingga lebih mudah dan terkontrol dalam mengakses basis data.

2.5. Web Developer

Menurut Kadir (2005:2), *World Wide Web* (WWW) atau biasa disebut dengan web merupakan salah satu sumber daya Internet yang berkembang pesat. Pertama kali aplikasi web dibangun hanya dengan menggunakan bahasa yang disebut HTML (*HyperText Markup Language*) dan protokol yang digunakan dinamakan HTTP (*HyperText Transfer Protocol*). Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML yang sekarang ini terdapat banyak skrip seperti: PHP dan ASP, sedangkan contoh yang berupa objek antara lain adalah applet (java) (Kadir, 2005:2). Jadi aplikasi web atau aplikasi berbasis web (*Web-based application*) adalah aplikasi untuk menyampaikan informasi kepada pengguna yang menggunakan layanan Internet berbasis web.

Dalam aplikasi tersebut, terjadi pertukaran antara klien (komputer yang meminta informasi) dengan server (komputer yang memasok atau menanggapi informasi). Web memberikan informasi secara *online* melalui internet langsung. Klien melakukan permintaan informasi dengan menggunakan *browser* (contoh *browser*: Internet Explorer, Opera, Mozilla, dan sebagainya). *Server* menerima informasi dan melayani permintaan dari *client*. Hal ini biasa disebut dengan web server (contoh web server: Apache, IIS, Xitami, dan sebagainya). Setelah itu, web server akan berkomunikasi dengan *middleware* (contoh *middleware*: ASP, JSP, PHP, dan sebagainya) untuk bisa berhubungan dengan basis data atau *database* (contoh *database*: access, oracle, sql, dan sebagainya). Setelah berinteraksi

dengan *database*, *server* yang telah mendapatkan informasi akan memberikan tanggapan terhadap klien yang meminta informasi tadi.

2.5.1. Unsur - Unsur Dalam Penyediaan Website atau Situs

Untuk menyediakan sebuah website, maka harus tersedia unsure-unsur penunjang, unsure-unsur penunjang dari website antara lain adalah :

1. Nama Domain

Nama domain atau biasa disebut dengan Domain Name atau URL adalah alamat unik di dunia internet yang digunakan untuk mengidentifikasi sebuah website, atau dengan kata lain domain name adalah alamat yang digunakan untuk menemukan sebuah website pada dunia internet, Nama domain diperjual belikan secara bebas di internet dengan status sewa tahunan. Setelah Nama Domain itu terbeli di salah satu penyedia jasa pendaftaran, maka pengguna disediakan sebuah kontrol panel untuk administrasinya. Jika pengguna lupa/tidak memperpanjang masa sewanya, maka nama domain itu akan di lepas lagi ketersediaannya untuk umum. Nama domain sendiri mempunyai identifikasi ekstensi/akhiran sesuai dengan kepentingan dan lokasi keberadaan website tersebut. Contoh nama domain ber-ekstensi internasional adalah com, net, org, info, biz, name, ws. Contoh nama domain ber-ekstensi lokasi Negara Indonesia adalah :

- .co.id : Untuk Badan Usaha yang mempunyai badan hukum sah
- .ac.id : Untuk Lembaga Pendidikan
- .go.id : Khusus untuk Lembaga Pemerintahan Republik Indonesia

- .mil.id : Khusus untuk Lembaga Militer Republik Indonesia
- or.id : Untuk segala macam organisasi yang tidak termasuk dalam kategori ac.id, co.id, go.id, mil.id dan lain lain
- sch.id : Untuk Lembaga Pendidikan yang menyelenggarakan pendidikan seperti SD, SMP dan atau SMU
- web.id : Ditujukan bagi badan usaha, organisasi ataupun perseorangan yang melakukan kegiatannya di World Wide Web.

2. Rumah tempat website (Web hosting)

Web Hosting dapat diartikan sebagai ruangan yang terdapat dalam harddisk tempat menyimpan berbagai data, file-file, gambar, video, data email, statistik, database dan lain sebagainya yang akan ditampilkan di website. Besarnya data yang bisa dimasukkan tergantung dari besarnya web hosting yang disewa/dipunyai, semakin besar web hosting semakin besar pula data yang dapat dimasukkan dan ditampilkan dalam website.

Web Hosting juga diperoleh dengan menyewa. Pengguna akan memperoleh kontrol panel yang terproteksi dengan username dan password untuk administrasi websitenya. Besarnya hosting ditentukan ruangan harddisk dengan ukuran MB (Mega Byte) atau GB (Giga Byte). Lama penyewaan web hosting rata-rata dihitung per tahun. Penyewaan hosting dilakukan dari perusahaan-perusahaan penyewa web hosting yang banyak dijumpai baik di Indonesia maupun Luar Negeri. Lokasi peletakan pusat data (datacenter) web hosting

bermacam-macam. Ada yang di Jakarta, Singapore, Inggris, Amerika, dll dengan harga sewa bervariasi.

3. Bahasa Program (Scripts Program).

Bahasa Program Adalah bahasa yang digunakan untuk menerjemahkan setiap perintah dalam website yang pada saat diakses. Jenis bahasa program sangat menentukan statis, dinamis atau interaktifnya sebuah website. Semakin banyak ragam bahasa program yang digunakan maka akan terlihat website semakin dinamis, dan interaktif serta terlihat bagus.

Beragam bahasa program saat ini telah hadir untuk mendukung kualitas website. Jenis jenis bahasa program yang banyak dipakai para desainer website antara lain HTML, ASP, PHP, JSP, Java Scripts, Java applets, XML, Ajax dsb. Bahasa dasar yang dipakai setiap situs adalah HTML sedangkan PHP, ASP, JSP dan lainnya merupakan bahasa pendukung yang bertindak sebagai pengatur dinamis, dan interaktifnya situs. Bahasa program ASP, PHP, JSP atau lainnya bisa dibuat sendiri. Bahasa program ini biasanya digunakan untuk membangun portal berita, artikel, forum diskusi, buku tamu, anggota organisasi, email, mailing list dan lain sebagainya yang memerlukan update setiap saat.

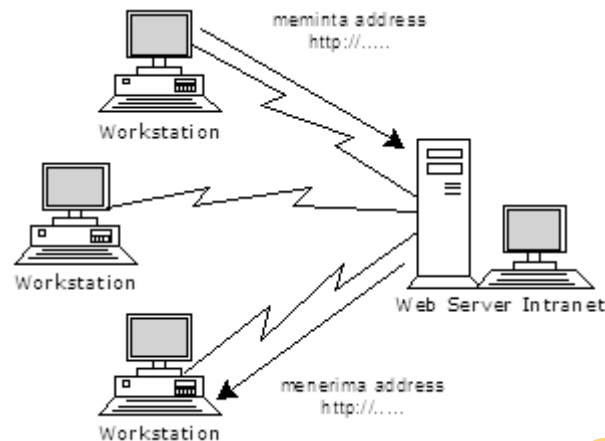
2.5.2. Intranet

Intranet (*Internal Network*) mulai didengung-dengungkan pada pertengahan tahun 1995 oleh beberapa penjual produk jaringan yang mengacu pada kebutuhan informasi dalam bentuk *Web* di dalam perusahaan. Intranet merupakan jaringan komputer dalam perusahaan yang menggunakan komunikasi

data standar seperti dalam Internet. Artinya, semua fasilitas Internet dapat digunakan untuk kebutuhan dalam perusahaan (atau dalam suatu organisasi). Dengan kata lain, Intranet dapat dikatakan ber-internet dalam lingkungan yang terbatas.

Adapun fasilitas standar Internet yang digunakan dalam Intranet adalah standar protokol TCP/IP. Standar tersebut memungkinkan protokol jaringan melakukan komunikasi, menerima dan mengirimkan data ke terminal yang lain. Standar yang lain adalah FTP (*File Transfer Protocol*) yang merupakan pelayanan *resource sharing*, sebuah fasilitas untuk mengambil *file* yang ada di Internet. SMTP (*Simple Mail Transfer Protocol*) yang merupakan dasar dari *e-mail* untuk berkomunikasi serta MIME (*Multipurpose Internet Mail Extensions*) yang merupakan standar untuk mendefinisikan format biner, grafik dan suara agar dapat ditransmisikan dengan *e-mail*. Selain itu terdapat protokol NNTP (*Network News Transfer Protocol*) dan POP (*Post Office Protocol*).

Protokol lainnya yang relatif baru dan sangat penting bagi Internet maupun Intranet adalah HTTP (*HyperText Transport Protocol*). Komputer yang memakai HTTP (dengan menggunakan perangkat lunak tertentu) disebut *Web server*. HTTP bertanggung jawab atas distribusi dan kolaborasi dokumen yang ada di Web server. HTTP tidak perlu tahu data yang dibawanya dan bahkan sistem operasi yang menjalankan Web server. Bila suatu *platform* mendukung TCP/IP dan *multitasking*, maka HTTP dapat digunakan. Kini, HTTP merupakan dasar dari layanan World Wide Web (WWW) Inter/Intranet.



Gambar 2.6 Aliran *hardware* dan logik dalam Intranet

Secara umum, seperti yang telah disebutkan sebelumnya bahwa teknologi yang digunakan antara Internet dan Intranet adalah sama. Namun demikian terdapat perbedaan antara Internet dengan Intranet dilihat dari perspektif jangkauan dan penggunaannya, yakni:

- Lingkup akses dan jangkauan.
- Cara teknologi yang digunakan untuk berkomunikasi.
- Tujuan dari terselenggaranya komunikasi.

Pada Internet, lingkupnya adalah global, komunikasi lewat saluran telekomunikasi publik, dan penggunanya bisa siapa saja tanpa membedakan posisi seseorang dalam kaitannya dengan isi informasi. Pada Intranet, cakupannya lebih terbatas, yakni di dalam organisasi; hubungannya antar kelompok kerja atau departemen di dalam perusahaan; penggunaannya oleh komunitas yang sudah ditentukan.

5.1. Testing dan Implementasi

Menurut standart ANSI/IEEE 1059, *Testing* adalah proses menganalisa suatu entitas *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (*defects/error/bugs*) dan mengevaluasi fitur-fitur dari entitas *software*.

Menurut Romeo (2003:3), *Testing software* adalah proses mengoperasikan *software* dalam suatu kondisi yang dikendalikan untuk:

1. Verifikasi

Apakah telah berlaku sebagaimana yang di tetapkan (menurut spesifikasi)?

2. Mendeteksi *Error*

3. Validasi

Apakah spesifikasi yang di tetapkan telah memenuhi keinginan atau kebutuhan pengguna yang sebenarnya ?

Menurut Romeo (2003:33), *Test Case* merupakan tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan kondisi ataupun hasil yang telah ditentukan sebelumnya. Metode *testing* ini dibagi menjadi dua, yaitu:

1. *White Box Testing*

White Box Testing atau *glass box testing* atau *clear box testing* adalah suatu metode *test case* yang menggunakan struktur kendali dari desain prosedural.

Metode desain *test case* ini dapat menjamin :

- a. Semua jalur (*path*) yang independen/terpisah dapat dites setidaknya sekali tes.
- b. Semua logika keputusan dapat dites dengan jalur yang salah atau jalur yang benar.

- c. Semua loop dapat dites terhadap batasannya dan ikatan operasional.
- d. Semua struktur internal data dapat dites untuk memastikan validasinya.

2. *Black box testing*

Black box testing atau *behavioral testing* atau *specification-based testing*, *input/output testing* atau *functional testing* dilakukan tanpa sepengetahuan detail struktur internal dari sistem atau komponen yang dites. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan spesifikasi kebutuhan dari *software*.

Menggunakan *black box testing*, perancang *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program. Kategori *error* dapat diketahui melalui *black box testing*, antara lain :

- a. Fungsi yang hilang atau tidak benar.
- b. *Error* dari antar muka.
- c. *Error* dari struktur data atau akses eksternal *database*.
- d. *Error* dari kinerja atau tingkah laku.
- e. *Error* dari inialisasi dan terminasi.