**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

CV. JUVE Komputer merupakan salah satu toko di kota Gorontalo yang memperbaiki segala jenis merek computer dan laptop. Toko ini berada di Jalan Jendral Sudirman belakang SMP N.6 kota Gorontalo. Toko ini berdiri pada tahun 2010 dan mempunyai karyawan berjumlah tiga orang.

Dalam sebulan toko ini dapat memperbaiki kurang lebih 7 laptop dengan berbagai macam jenis kerusakan yang dapat di lihat dari table di bawah ini:

**Tabel 1.1 Laptop dan Jenis Kerusakannya**

|  |  |  |
| --- | --- | --- |
| No | JENIS LAPTOP | KERUSAKAN |
| 1 | Axioo | Layar biru pada saat mulai loading |
| 2 | Toshiba | Tambah kapasistas RAM |
| 3 | Acer | Layar monitor tidak menyala |
| 4 | Axioo | Tidak dapat masuk booting |
| 5 | Acer | Menyala tapi muncul garis hitam |
| 6 | Axioo | Lambat masuk di windows |
| 7 | Acer | lampu LCD menyala tapi tidak terang |

Sistem pakar adalah sistem yang menggunakan pengetahuan manusia yang terekam dalam komputer untuk memecahkan persoalan yang biasanya memerlukan keahlian manusia. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Sistem pakar dapat ditampilkan dalam dua lingkungan yaitu pengembangan dan konsultasi. Lingkungan pengembangan digunakan oleh pembangun sistem pakar untuk membangun komponen dan memasukkan pengetahuan ke dalam basis pengetahuaan. Lingkungan konsultasi digunakan oleh orang yang bukan ahli untuk memperoleh pengetahuan dan berkonsultasi.

1

Laptop merupakan suatu perkembangan teknologi komputer yang bisa dibawah kemana-mana dengan kata lain komputer jinjing. Komputer jinjing (populer dalam bahasa Inggris: laptop, notebook, atau powerbook) adalah komputer bergerak yang berukuran relatif kecil dan ringan, beratnya berkisar dari 1-6 kg, tergantung ukuran, bahan, dan spesifikasi laptop tersebut. Sumber daya komputer jinjing berasal dari baterai atau adaptor A/C yang dapat digunakan untuk mengisi ulang baterai dan menyalakan laptop itu sendiri. Baterai laptop pada umumnya dapat bertahan sekitar 1 hingga 6 jam sebelum akhirnya habis, tergantung dari cara pemakaian, spesifikasi, dan ukuran baterai.

Sebagai komputer pribadi, laptop memiliki fungsi yang sama dengan personal komputer pada umumnya. Komponen yang terdapat di dalamnya sama persis dengan komponen pada laptop, hanya saja ukurannya diperkecil, dijadikan lebih ringan, lebih tidak panas, dan lebih hemat daya. laptop kebanyakan menggunakan layar LCD (Liquid Crystal Display) berukuran 10 inci hingga 17 inci tergantung dari ukuran laptop itu sendiri. Selain itu, papan ketik yang terdapat pada notebook juga dilengkapi dengan papan sentuh yang berfungsi sebagai “pengganti” mouse.

Berbeda dengan personal komputer (PC), laptop memiliki komponen pendukung yang didesain secara khusus untuk mengakomodasi sifat komputer jinjing yang portabel. Sifat utama yang dimiliki oleh komponen penyusun laptop adalah ukuran yang kecil, hemat konsumsi energi, dan efisien.

## Merek laptop yang sangat banyak disukai pembeli yang ada di Gorontalo yaitu Toshiba, Acer, Sony, Axioo, Asus . (survey pada setiap mahasiswa yang mengunakan laptop, toko computer)

Pemakai laptop baru menyadari kerusakan setelah laptop tidak dapat beroperasi sebagaimana mestinya. Oleh karena itu dalam penggunaan laptop kemungkinan besar membutuhkan perawatan berkala. Ini dapat dilakukan dengan cara mengetahui kerusakan apa yang terjadi pada laptop seperti yang telah di sebutkan di atas masalah-masalah apa saja yang ada pada laptop hal inilah yang mendorong pembangunan sistem pakar untuk *Troubleshooting* laptop Sehingga pemakai laptop dapat mengetahui masalah yang terjadi pada laptop tersebut dan melakukan perbaikan lebih awal tanpa harus membawa ke tempat perbaikan computer atau laptop.

Beberapa tugas tertentu membutuhkan pengetahuan yang sangat khusus sehingga dibutuhkan para pakar. Konsep pakar didasarkan pada asumsi bahwa pengetahuan pakar dapat ditangkap dalam penyimpanan komputer dan kemudian diterapkan oleh orang lain saat dibutuhakan.

Sistem pakar pada penelitian ini merupakan *system* berbasis web yang menghadirkan solusi dalam mengatasi kerusakan laptop atau *notebook*.

Sistem menyediakan fasilitas *troubleshooting* untuk membantu *user* dalam mengidentifikasi permasalahan komputernya. Proses identifikasi ini dilakukan melalui interaksi tanya - jawab antara sistem dengan pemakai.

Sistem juga menyediakan fasilitas edit pengetahuan yang dapat digunakan oleh *Knowledge Engineer* dalam melakukan perubahan data pada basis pengetahuan. Metode penelusuran solusi yang digunakan pada sistem ini adalah penelusuran runut maju (*forwardchainning*). Aturan pada basis pengetahuan dimodelkan sebagai tree dengan memanfaatkan database MySQL serta bahasa pemrograman PHP.

Pemanfaatan *database* untuk menyimpan basis pengetahuan dari sistem pakar akan mempermudah dalam pembuatan fasilitas penambahan pengetahuan.

Dengan adanya fasilitas penambahan pengetahuan, perubahan aturan pada basis pengetahuan dan pengembangan system melalui akuisisi pengetahuan yang baru dapat langsung dilakukan tanpa harus membongkar sistem yang sudah jadi. Hal ini akan memungkinkan system menjadi tetap *uptodate*.

Apabila dikaitkan dengan kemampuan seorang teknisi komputer dalam mendiagnosa kerusakan laptop. Maka dapat dibuat suatu program computer atau laptop yang bertugas untuk mengetahui dan menganalisis masalah – masalah kerusakan laptop untuk kemudian memberikan anjuran cara memperbaiki laptop tersebut.

Jenis – jenis kerusakan yang penulis maksudkan khusus kerusakan laptop yang sebenarnya dapat diperbaiki sendiri oleh pengguna laptop di rumah, tetapi biasanya pengguna laptop takut memperbaikinya sendiri. Pada aplikasi dengan konsep sistem pakar ini terdiri dari banyak macam masalah yang biasanya ditemui pada para pengguna laptop.

Atas dasar uraian di atas, maka penulis ingin membahas mengenai bagaimana caranya membuat suatu sistem yang dapat dengan mudah melayani para pengguna laptop untuk memperbaiki laptopnya pada saat bermasalah, yang selain menghemat waktu, tenaga, dan biaya, juga tanpa harus datang ke tempat reparasi laptop.

Oleh karena itu penulis ingin mewujudkannya dengan mencoba membuat aplikasi yang menggunakan konsep sistem pakar, dengan judul “**SISTEM PAKAR *TROUBLESHOOTING* LAPTOP BERBASIS *WEBSITE* STUDI KASUS PADA CV. JUVE KOMPUTER”**

* 1. **Rumusan Masalah**

Dari uraian latar belakang yang telah penulis paparkan diatas, maka dapat penulis rumuskan beberapa rumusan masalah, sebagai berikut;

1. Bagaimana cara merekayasa Sistem Pakar *Troubleshooting* kerusakan laptop atau notebook Berbasis Web ?
2. Apakah sistem pakar yang direkayasa dapat digunakan untuk *Troubleshooting* kerusakan dan masalah pada laptop atau notebook ?
   1. **Maksud dan Tujuan Penelitian**
      1. **Maksud penelitian**

Maksud dari penelitian ini adalah membangun sebuah aplikasi sistem pakar *troubleshooting* pada laptop dan memberikan petunjuk penanganan kerusakan laptop.

1. **Tujuan Penelitian**

Tujuan yang akan dicapai dalam penelitian ini adalah :

* 1. Untuk mengetahui cara merekayasa sistem pakar untuk *troubleshooting* kerusakan pada laptop atau *notebook* berbasis web.
  2. Sistem pakar *troubleshooting*  laptop atau *notebook* berbasis *website* yang sudah direkayasa dapat digunakan pada Juve komputer dan masyarakat umum penguna laptop.
  3. **Manfaat Penelitian**

1. **Pengembangan Pengembangan Ilmu Pengetahuan dan Teknologi.**

Penelitian ini diharapk dan dapat mengembangkan ilmu pengetahuan dibidang teknologi informasi pada umumnya dan dapat membuat aplikasi sistem pakar *troubleshooting* kerusakan personal laptop atau *notebook***.**

1. **Praktisi**

Sebagai bahan masukan bagi semua elemen-elemen ataupun unsur-unsur yang terlibat dalam pembuatan aplikasi sistem pakar untuk *troubleshooting* kerusakan laptop atau *notebook*.

1. **Peneliti**

Sebagai bahan masukan dan bahan pembelajaran kepada peneliti berikutnya yang akan meneliti masalah sistem pakar untuk *troubleshooting* kerusakan laptop atau *notebook*.

**BAB II**

**LANDASAN TEORI**

* 1. **Sistem Berbasis Pengetahuan**

Sistem berbasis pengetahuan (SBP) atau *knowledge-based System* adalah sistem yang melakukan sebuah tugas dengan menggunakan aturan-aturan yang diubah ke sebuah representasi simbolik pengetahuan, dan juga memakai banyak metode merepresentasikan dan melakukan penalaran pengetahuan suatu subjek tertentu untuk memecahkan persoalan atau memberikan saran berdasar oleh pengetahuan yang diambil dari pakar dibidang tersebut. Contohnya sebuah program yang bisa menerjemahkan cuaca termasuk SBP karena tidak membutuhkan tenaga ahli khusus, memprediksi cuaca adalah *Expert System* (Sistem Pakar) karena membutuhkan keahlian khusus oleh pakar dalam bidang meteorologi. Oleh karena itu, dapat dikatakan ES merupakan varian dari SBP.

Sistem pakar atau *expert sistem* merupakan suatu cabang dari ilmu komputer yang berdasarkan kecerdasan buatan (*artificial intelligence*). *Artificialintelligence* adalah sebuah rancangan program yang memungkinkan komputer melakukan suatu tugas atau mengambil keputusan dengan meniru suatu cara berpikir dan penalaran manusia.

8

Adapun beberapa definisi *artificialintelligence* adalah sebagai berikut :

a. Definisi *artificial intelligence* menurut Jogiyanto adalah :

“*Artificial Intelligence* didefinisikan sebagai suatu mesin atau alat pintar (biasanya adalah komputer) yang dapat melakukan suatu tugas yang bilamana tugas tersebut dilakukan oleh manusia akan dibutuhkan suatu kepintaran untuk melakukannya”.

b. Definisi *artificial intelligence* menurut Kusnadi adalah :

“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”.

c. Definisi *artificial intelligence* menurut Sarwono [9] adalah :

“Kecerdasan buatan adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia”.

Dengan perancangan *artificialintelligence* yang baik, maka diharapkan peran manusia dapat diminimalkan dan dapat meringankan beban kerja manusia. Cara kerja *artificialintelligence* adalah menerima *input*, kemudian diproses dan kemudian mengeluarkan *output* berupa suatu keputusan. Implementasi dari *artificial intelligence* saat ini umum ditemui dalam bidang-bidang seperti berikut :

a. *Computer Vision* : yaitu suatu metode *artificialintelligence* yang memungkinkan sebuah sistem komputer mengenali gambar sebagai inputnya. Contohnya adalah mengenali dan membaca tulisan yang ada didalam gambar.

b. *FuzzyLogic* : yaitu suatu metode *artificialintelligence* yang banyak terdapat pada alat-alat elektronik dan robotika. Dimana alat-alat elektronik atau robotika tersebut mampu berpikir dan bertingkah laku sebagaimana layaknya manusia.

c. *Game* : yaitu suatu metode *artificial intelligence* yang berguna untuk meniru cara berpikir seorang manusia dalam bermain *game*. Contohnya adalah program *Perfect Chessmate* yang mampu berpikir setara dengan seorang *grandmaster* catur.

d. *GeneralProblemSolving* : yaitu suatu metode *artificialintelligence* yang berhubungan dengan pemecahan suatu masalah terhadap suatu situasi yang akan diselesaikan oleh komputer. Biasanya permasalahan tersebut akan diselesaikan secara *trial* and *error* sampai sebuah solusi dari sebuah masalah didapatkan. Contohnya adalah program *Eureka* yang dapat memecahkan model linier programming.

e. *Speech Recognition* : yaitu suatu metode *artificial intelligence* yang berguna untuk mengenali suara manusia dengan cara dicocokkan dengan acuan yang telah diprogramkan sebelumnya. Contohnya adalah suara dari *user* dapat diterjemahkan menjadi sebuah perintah bagi komputer.

f. *Expert Sistem* : yaitu suatu metode *artificial intelligence* yang berguna untuk meniru cara berfikir dan penalaran seorang ahli dalam mengambil keputusan berdasarkan situasi yang ada. Dengan *expert sistem* seorang *user* dapat melakukan konsultasi kepada komputer, seolah-olah *user* tersebut berkonsultasi langsung kepada seorang ahli. Contohnya adalah program aplikasi yang mampu meniru seorang ahli medis dalam mendeteksi demam berdarah berdasarkan keluhan-keluhan pasiennya.

**2.2 Sistem Pakar (*ExpertSystem*)**

**2.2.1 Konsep Dasar Sistem Pakar**

Sistem pakar (*expert system*) adalah suatu sistem komputer berbasis pengetahuan yang menyamai kemampuan pengambilan keputusan dari seorang pakar. Adapun beberapa definisi tentang sistem pakar (*expert system*) lainnya, adalah sebagai berikut :

a. Menurut Kusnadi, sistem pakar didefinisikan sebagai berikut :

“Sistem pakar adalah suatu sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli”.

b. Menurut Sarwono, sistem pakar didefinisikan sebagai berikut :

“Sistem pakar adalah salah satu cabang dari *artificial intelligence* yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar”.

Sistem pakar telah dibuat untuk memecahkan masalah-masalah dalam berbagai bidang, antara lain matematika, teknik, kedokteran, ilmu komputer, sampai bidang hukum. Walaupun sistem pakar sebagai sistem komputer yang dalam berbagai hal bekerjanya jauh lebih baik dari manusia atau ahli, tetapi kita tidak bisa menghilangkan begitu saja faktor manusia dan digantikan oleh sistem komputer, karena pada banyak situasi keahlian manusia tetap dibutuhkan, sebab kemampuan komputer terbatas.

Pengetahuan dari suatu sistem pakar mungkin dapat direpresentasikan dalam sejumlah cara. Salah satu metode yang paling umum untuk merepresentasikan pengetahuan adalah dalam bentuk tipe aturan (*rule*) IF THEN (jika..maka). Walaupun cara diatas sangat sederhana, namun banyak hal yang berarti dalam membangun sistem pakar dengan mengekspresikan pengetahuan pakar dalam bentuk aturan diatas.

Konsep dasar dari suatu sistem pakar mengandung beberapa unsur atau elemen, yaitu keahlian, ahli, pengalihan keahlian, inferesi, aturan dan kemampuan menjelaskan. Keahlian merupakan suatu penguasaan pengetahuan dibidang tertentu yang didapatkan dari pelatihan, membaca atau pengalaman. Seorang ahli adalah seorang yang mempunyai pengetahuan tertentu dan mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topik permasalahan (*domain*), menyusun kembali pengetauan jika dipandang perlu, dan menentukan relevan atau tidaknya keahlian mereka.

Pengalihan keahlian dari para ahli untuk kemudian dialihkan lagi ke orang lain yang bukan ahli, merupakan tujuan utama dari sistem pakar. Proses ini membutuhkan empat aktivitas, yaitu tambahan pengetahuan dari para ahli atau sumber-sumber lainnya, representasi pengetahuan kedalam komputer, inferensi pengetahuan dan pengalihan pengetahuan ke pengguna. Pengetahuan yang disimpan dikomputer dinamakan dengan basis pengetahuan (*knowledgebase*).

Salah satu fitur yang harus dimiliki oleh sistem pakar adalah kemampuan untuk menalar (*reasoning*). Jika keahlian-keahlian sudah tersimpan sebagai basis pengetahuan dan sudah tersedia program yang mampu mengakses basis data, maka komputer harus dapat diprogram untuk membuat inferensi. Proses ini dibuat dalam bentuk motor inferensi (*inferenceengine*).

Sistem pakar menggunakan basis pengetahuan (*knowldegebase*) sebagai dasar pemikirannya. *Knowledgebase* tersebut terdiri dari heuristik dan sejumlah *rule-rule* dan aturan-aturan yang tersusun secara sistematis dan spesifik, juga relasi antara data dan aturan (*rule*) dalam pengambilan kesimpulan. *Knowledge base* tersebut disimpan dalam sebuah basis data pada *database*. Sedangkan sebagai pusat pemrosesannya adalah *inferenceengine*, yaitu suatu rancangan aplikasi yang berfungsi untuk memberikan pertanyaan dan menerima *input* dari *user*, kemudian melakukan proses logika sesuai dengan *knowledgebase* yang tersedia, untuk selanjutnya menghasilkan *output* berupa suatu kesimpulan atau bisa juga berupa keputusan sebagai hasil akhir konsultasi.

*Knowledgeacquisitionsource* berfungsi sebagai penterjemah dari *knowledgebase* menjadi sebuah bahasa yang dapat dimengerti oleh *user*. Bagian ini diperlukan karena *knowledgebase* yang disimpan dalam sebuah *database*, disimpan dalam format tertentu, sedemikian rupa sehingga *user* sulit mengartikannya. *Disk* (*workingmemory*) adalah sejumlah modul *memory* yang menyimpan informasi sementara dari suatu proses konsultasi. Setiap proses baru dijalankan, memory tersebut akan di set ke kondisi awal. Dalam menjalankan proses, memory tersebut menyimpan informasi dari *rule-rule* yang dipakai dalam *knowledge base.*

Oleh karena itu, langkah-langkah perancangan sistem pakar sebaiknya mengikuti urutan sebagai berikut :

1. Menentukan batasan-batasan atau bidang konsentrasi dari sebuah sistem pakar yang akan dirancang.

2. Memilih jenis keputusan apa yang diambil.

3. Membuat pohon keputusan (*decision tree*).

4. Menuliskan *IF-THEN rules.*

5. Merancang antarmuka pengguna (*user interface*).

**2.2.2 Ciri-ciri Sistem Pakar**

Sistem pakar memiliki ciri-ciri sebagai berikut :

1. Terbatas pada bidang yang spesifik.

2. Dapat memberikan penalaran untuk data-data yang tidak pasti.

3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.

4. Berdasarkan pada *rule* atau kaidah tertentu.

5. *Outputnya* bersifat nasihat atau anjuran dan tergantung dari dialog *user*.

6. *Knowledgebase* dan *inferenceengine* terpisah.

**2.2.3 Orang Yang Terlibat Dalam Sistem Pakar**

Untuk memahami perancangan sistem pakar, perlu dipahami mengenai siapa saja yang berinteraksi dengan sistem. Mereka adalah :

1. Pakar (*domainexpert*) adalah seseorang ahli yang dapat menyelesaikan masalah yang sedang diusahakan untuk dipecahkan oleh sistem.

2. Pembangun pengetahuan (*knowledge engineer*) merupakan seseorang yang menerjemahkan pengetahuan seorang pakar dalam bentuk deklaratif sehingga dapat digunakan oleh sistem pakar.

3. Pengguna (*user*) merupakan seseorang yang berkonsultasi dengan sistem untuk mendapatkan saran yang disediakan oleh pakar.

4. Pembangun sistem (*sistem engineer*) merupakan seseorang yang membuat antarmuka pengguna, merancang bentuk basis pengetahuan secara deklaratif dan mengimplementasikan mesin inferensi.

**2.2.4 Kategori Masalah Sistem Pakar**

Masalah-masalah yang dapat diselesaikan dengan sistem pakar,diantaranya adalah :

1. Interpretasi, yaitu membuat kesimpulan atau deskripsi dari sekumpulan data mentah, termasuk diantaranya juga pengawasan, pengenalan ucapan, analisis citra, interpretasi sinyal dan beberapa analisis kecerdasan.

2. Proyeksi, yaitu memprediksi akibat-akibat yang dimungkinkan dari situasi-situasi tertentu, diantaranya peramalan, prediksi demografis, peramalan ekonomi, prediksi lalu lintas atau peramalan keuangan.

3. *Troubleshooting*, yaitu sebuah istilah dalam bahasa inggris merupakan pencarian sumber masalah secara sistematis sehingga masalah tersebut dapat diselesaikan. pada umumnya digunakan dalam berbagai bidang, seperti halnya dalam bidang komputer, administrasi sistem, dan juga bidang electronika.

4. Desain, yaitu menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu, diantaranya layout sirkuit dan perancangan bangunan.

5. Perencanaan, yaitu merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu, diantaranya perencanaan keuangan, komunikasi militer dan manajemen proyek.

6. Monitoring, yaitu membandingkan tingkah laku suatu sistem yang teramati dengan tingkah laku yang diharapkan darinya, daiantaranya *Computer Aided Monitoring Sistem.*

7. *Debugging* dan *repair*, yaitu menentukan dan mengimplementasikan cara-cara untuk mengatasi malfungsi, diantaranya memberikan resep obat terhadap suatu kegagalan.

8. Instruksi, yaitu mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subjek, diantaranya melakukan instruksi untuk diagnosis, *debugging* dan perbaikan kinerja.

9. Pengendalian, yaitu mengatur tingkah laku suatu *environment* yang kompleks seperti kontrol terhadap interpretasi-interpretasi, prediksi, perbaikan dan monitoring kelakuan sistem.

10. Seleksi, mengidentifikasi pilihan terbaik dari sekumpulan kemungkinan.

11. Simulasi, pemodelan interaksi antara komponen-komponen sistem.

**2.2.5 Kelebihan Sistem Pakar**

Seorang pakar dengan sistem pakar mempunyai banyak perbedaan. Perbandingan kemampuan antara seorang pakar dengan sebuah sistem pakar dapat

dilihat seperti pada tabel 2.1 berikut ini :

**Tabel 2.1 Perbandingan kemampuan seorang pakar**

**dengan sistem pakar**

|  |  |  |
| --- | --- | --- |
| **Factor** | **Human expert** | **Expert system** |
| Time availability | Hari kerja | Setiap saat |
| Geografis | Local/tertentu | Dimana saja |
| Keamanan | Tidak tergantikan | Dapat diganti |
| Perishable/dapat habis | Ya | Tidak |
| Performasi | Variable | Konsisten |
| Kecepatan | Variable | Konsisten |
| Biaya | Tinggi | Terjangkau |

Dari tabel 2.1 diatas dapat dikembangkan penjelasan lebih lanjut tentang kelebihan sistem pakar.

1. Sistem pakar bisa digunakan setiap hari menyerupai sebuah mesin sedangkan seorang pakar tidak mungkin bekerja terus menerus setiap hari tanpa beristirahat.

2. Sistem pakar merupakan suatu software yang dapat diperbanyak dan kemudian dibagikan keberbagai lokasi maupun tempat yang berbeda-beda untuk digunakan, sedangkan seorang pakar hanya berkerja pada satu tempat dan pada saat yang bersamaan.

3. Suatu sistem pakar dapat diberi pengamanan untuk menentukan siapa saja yang mempunyai hak akses untuk menggunakannya dan jawaban yang diberikan oleh sistem terbebas dari proses intimidasi atau ancaman, sedangkan seorang pakar bisa saja mendapat ancaman atau tekanan pada saat menyelesaikan permasalahan.

4. Pengetahuan (*knowledge*) yang disimpan pada sistem pakar tidak akan bisa hilang atau lupa, yang dalam hal ini tentu harus didukung oleh *maintenance* yang baik, sedangkan pengetahuan seorang pakar manusia lambat laun akan hilang karena meninggal, usia yang semakin tua, maupun menderita suatu penyakit.

5. Kemampuan memecahkan masalah pada suatu sistem pakar tidak dipengaruhi oleh faktor dari luar seperti intimidasi, perasaan kejiwaan, faktor ekonomi ataupun perasaan tidak suka. Akan tetapi sebaliknya dengan seorang pakar yang disebutkan dipengaruhi oleh faktor-faktor luar seperti yang disebutkan diatas ketika sedang menyelesaikan atau memecahkan suatu masalah, sehingga dapat memunculkan jawaban yang berbeda-beda atas pertanyaan yang diajukan walaupun masalahnya sama atau dengan kata lain, seorang pakar boleh jadi tidak konsisten.

6. Umumnya kecepatan dalam memecahkan masalah pada suatu sistem pakar relatif lebih cepat dibandingkan oleh seorang pakar manusia. Hal ini sudah dibuktikan pada beberapa sistem pakar yang tekenal didunia.

7. Biaya menggaji seorang pakar lebih mahal bila dibandingkan dengan penggunaan program sistem pakar (dengan asumsi bahwa program sistem pakar itu sudah ada).

**2.2.6 Keuntungan Sistem Pakar**

Sistem pakar merupakan paket perangkat lunak atau paket program komputer yang ditujukan sebagai penyedia nasihat dan sarana bantu dalam memecahkan masalah. Ada beberapa keunggulan dari sistem pakar, diantaranya dapat :

1. Menghimpun data dalam jumlah yang sangat besar.

2. Menyimpan data tersebut untuk jangka waktu yang panjang dalam suatu bentuk tertentu.

3. Mengerjakan perhitungan secara cepat dan tepat serta mencari kembali data yang tersimpan dengan kecepatan tinggi.

Selanjutnya ada banyak keuntungan bila menggunakan sistem pakar, keuntungan-keuntungan tersebut diatantaranya adalah :

1. Menjadikan pengetahuan dan nasihat lebih mudah didapat.

2. Meningkatkan output dan produktivitas.

3. Menyimpan kemampuan dan pengetahuan pakar.

4. Meningkatkan penyelesaian masalah.

5. Meningkatkan reliabilitas.

6. Memberikan jawaban yang cepat.

7. Merupakan panduan yang cerdas.

8. Dapat bekerja dengan informasi yang kurang lengkap dan tidak pasti.

**2.2.7 Kelemahan Sistem Pakar**

Adapun kelemahan-kelemahan yang dimiliki oleh sebuah sistem pakar diantaranya adalah :

1. Masalah dalam mendapatkan pengetahuan dimana pengetahuan tidak selalu bisa didapatkan dengan mudah, karena kadangkala pakar dari masalah yang kita buat tidak ada, dan kalaupun ada kadang-kadang pendekatan yang dimiliki oleh pakar berbeda-beda.

2. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dan pemeliharaanya.

3. Boleh jadi sistem tak dapat membuat keputusan.

4. Sistem pakar tidaklah 100% menguntungkan, walaupun seorang tetap tidak sempurna atau tidak selalu benar. Oleh karena itu perlu diuji ulang secara teliti sebelum digunakan. Dalam hal ini manusia tetap merupakan faktor dominan.

Kelemahan-kelemahan atau kekurangan dari sistem pakar tersebut bukanlah sama sekali tidak bisa diatasi, tetapi dengan terus melakukan perbaikan dan pengolahan berdasarkan pengalaman yang telah ada maka hal itu diyakini akan dapat diatasi walaupun dalam waktu yang panjang dan terus menerus.

**2.2.8 Struktur Sistem Pakar**

Sistem pakar terdiri dari dua bagian utama yaitu *developmentenvironment* dan *consultationenvironment*. *Developmentenvironment* dipakai oleh pembangun sistem pakar untu membangun komponen-komponen dan mengenalkan suatu sistem pakar pengetahuan kepada *knowledgebase*.*Consultatitionenvironment* dipakai oleh *user* untuk mendapatkan suatu pengetahuan yang berhubungan dengan suatu keahlian. Komponen-komponen sistem pakar dalam kedua bagian tersebut dapat dilihat dalam Gambar 2.1 dibawah ini :



**Gambar 2.1 Struktur Sistem Pakar**

Komponen-komponen yang terdapat dalam sistem pakar adalah seperti yang terdapat pada gambar 2.1, yaitu basis pengetahuan (*knowledgebase*), akuisisi pengetahuan, mesin *inferensi* (*inferenceenine*), area memori kerja (*workplace*), antarmuka pengguna (*userinterface*), fasilitas penjelasan, perbaikan pengetahuan.

Secara umum pemakai berinteraksi dengan sistem pakar melalui *interface* pemakai. *Interface* tersebut terdapat berupa pertanyaan-jawaban, menu pilihan, bahasa alami atau grafik. Sedangkan subsistem penjelasan dalam sistem pakar memungkinkan program untuk menjelaskan penalarannya kepada pemakai yang antara lain menyangkut kebutuhan sistem terhadap data tertentu.

**2.2.8.1 *KnowledgeBase***

*Knowledgebase* adalah suatu struktur data yang menyimpan informasi data, *rule*, relasi antara data dengan *rule* dalam pengambilan kesimpulan. *Knowledgebase* dapat dikatakan sebagai kumpulan informasi dan pengalaman seorang ahli pada suatu bidang tertentu. *Knowledgebase* terdiri dari fakta seperti halnya buku teori dan heuristik yang merupakan langkah-langkah seorang ahli dalam mengambil keputusan.

Setiap kasus yang ditemui, diproses berdasarkan pengalaman yang pernah tercatat, baru dapat diambil kesimpulan. Apabila terjadi suatu kasus yang belum pernah dialami, maka sistem pakar tidak mampu mengambil kesimpulan dari kasus tersebut. Untuk itu diperlukan penambahan data baru yang dapat dijadikan acuan yang melengkapi *knowledgebase*. Jadi, suatu *knowledgebase* dapat dikatakan baik apabila memiliki sejumlah *rule* yang mampu digunakan dalam setiap kemungkinan kasus dalam ruang lingkup tertentu. Dalam perancangannya, perancang harus mampu menggali sebanyak mungkin informasi dari narasumbernya, seorang ahli dan memindahkannya kedalam *knowledgebase*.

Dalam studi kasus pada sistem berbasis pengetahuan terdapat beberapa karakteristik yang dibangun untuk membantu kita di dalam membentuk serangkaian prinsip-prinsip arsitekturnya. Prinsip tersebut meliputi :

a. Pengetahuan merupakan kunci kekuatan sistem pakar.

b. Pengetahuan sering tidak pasti dan tidak lengkap.

c. Pengetahuan sering miskin spesifikasi.

d. Amatir menjadi ahli secara bertahap.

e. Sistem pakar harus fleksibel.

f. Sistem pakar harus transparan.

**2.2.8.2 Akuisisi Pengetahuan**

Akuisisi pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan kedalam program komputer. Tahap akuisisi pengetahuan ini biasanya dikerjakan oleh seorang *knowledgeengineer*. *Knowledgeengineer* adalah seorang yang memiliki latar belakang pengetahuan tentang komputer dan mengerti cara pengembangan sistem pakar serta yang menerjemahkan pengetahuan seorang pakar dalam bentuk deklaratif sehingga dapat digunakan oleh sistem pakar. Dalam tahap ini *knowledgeengineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer kedalam basis pengetahuan. Pengetahuan diperoleh dari pakar, dilengkapi dengan buku, basis data, pengalaman pemakai, laporan penelitian dan sebagainya. Dalam akuisisi pengetahuan (*knowledge acquisition*) ini terdapat empat metode utama, yaitu :

1. Wawancara

Wawancara adalah metode akuisisi yang paling banyak digunakan.Metode ini melibatkan pembicaraan dengan pakar secara langsung dalam suatu wawancara.

2. Analisis Protokol

Dalam metode akuisisi ini, pakar diminta untuk melakukan suatu pekerjaan dan mengungkapkan proses pemikirannya dengan menggunakan kata-kata. Pekerjaan tersebut direkam, dituliskan dan dianalisis.

3. Observasi pada pekerjaan pakar

Dalam metode ini, pekerjaan dalam bidang tertentu yang dilakukan pakar direkam dan diobservasi.

4. Induksi aturan

Metode ini dibatasi untuk sistem berbasis aturan. Induksi adalah suatu proses penalaran dari khusus ke umum. Suatu sistem induksi aturan diberi contoh-contoh dari suatu masalah yang hasilnya telah diketahui. Setelah diberikan beberapa contoh, sistem induksi aturan tersebut dapat membuat aturan yang benar untuk kasus-kasus contoh. Selanjutnya aturan dapat digunakan untuk menilai kasus lain yang hasilnya tidak diketahui.

**2.2.8.3 Mesin Inferensi**

Mesin inferensi atau *inferenceengine* adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan. Seperti halnya *knowledgebase*, *inferenceengine* juga terdiri dari *rule-rule* dan fakta-fakta, bedanya apabila didalam *knowledge base*, *rule-rule* dan fakta-fakta yang ada berisi sebuah *domain* yang spesifik dari sifat-sifat seorang ahli. Sedangkan *rule-rule* dan fakta-fakta pada *inferenceengine*, berisi strategi pencarian kesimpulan yang dilakukan oleh *expert sistem*.

Pemisahan kedua jenis *rule-rule* dan fakta-fakta tersebut memiliki beberapa keuntungan, yang pertama, memudahkan untuk melakukan perubahan dalam *knowledgebase* tanpa berimbas banyak pada *inferenceengine*, dan sebaliknya. Kedua, berguna untuk pengembangan dan komponen-komponen lainnya yang terdapat pada sistem pakar (terkecuali *knowledgebase*). Tujuan utama dari sistem pakar adalah membentuk dan mengeluarkan saran.

Untuk menyelesaikan tugas ini, sistem pakar harus melakukan pencarian solusi yang mana adalah tanggung jawab dari *inferenceengine* untuk melakukannya secara efisien dan efektif. Seringkali dalam proses pencarian, *user* dihadapkan pada sejumlah alternatif dan berbagai jenis batasan. Terdapat dua pendekatan untuk mengontrol inferensi dalam sistem pakar berbasis aturan, yaitu pelacakan kebelakang (*backwardchaining*) dan pelacakan kedepan (*forwardchaining*). Pelacakan kebelakang adalah pendekatan yang dimotori tujuan (*goal*-*driven*).

Dalam pendekatan ini, pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujujuan tersebut untuk kesimpulannya. Selanjutnya proses pelacakan menggunakan premis untuk aturan tersebut sebagai tujuan baru dan mencari aturan lain dengan tujuan baru sebagai kesimpulannya. Proses berlanjut sampai semua kemungkinan ditemukan. Gambar 2.2 menunjukan proses pelacakan kebelakang (*backwardchaining*).



**Gambar 2.2 Proses *Backward Chaining***

Pelacakan kedepan (*forward chaining*) adalah pendekatan yang dimotori oleh data (*data-driven*). Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Pelacakan kedepan mencari fakta yang sesuai dengan bagian *IF* dari aturan *IF-THEN*. Gambar 2.3 menunjukkan proses forward chaining.



**Gambar 2.3 Proses *Forward Chaining***

Kedua metode inferensi tersebut dipengaruhi oleh dua macam penelusuran, yaitu *depthfirstsearch* dan *breadthfirstsearch*. *Depthfirstsearch* melakukan penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ketingkat dalam yang berurutan. Gambar 2.4 dibawah ini menunjukkan metode penelusuran *depth first search.*



**Gambar 2.4 Teknik Penelusuran *Depth First Search***

*Breadth first* search bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah ketingkat selanjutnya. Gambar 2.5 dibawah ini menunjukkan metode penelusuran *breadth first search*.

****

**Gambar 2.5 Teknik Penelusuran *Breadth First Search***

Dalam memilih apakah akan menggunakan pelacakan kedepan atau pelacakan kebelakang, semuanya bergantung masalah yang akan tetap dibuatsistem pakarnya, dan belum dapat dibuktikan mana yang lebih baik diantara kedua metode inferensi ini.

**2.2.8.4 Antarmuka Pengguna**

Antamuka pengguna atau *user interface* merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. *User interface* atau antamuka pengguna menerima informasi dari pemakai dan mengubahnya kedalam bentuk yang dapat diterima oleh sistem. Selain itu, antarmuka pengguna menerima informasi dari sistem dan menyajikannya kedalam bentuk yang dapat dimengerti oleh pemakai. Karena itu, perancangan *user interface* haruslah memenuhi prinsip *user friendly*, yaitu mudah digunakan oleh *user* dan sejelas mungkin agat tidak terjadi adanya salah interpretasi.

Sistem pakar sering kali interaktif, yaitu saling tukar informasi dan konklusi dengan seorang pemakai dalam suatu dialog. Penampilan kecerdasannya diperbaharui melalui kenyataan bahwa program dapat menerima masukan bentuk bebas dalam kalimat-kalimat *user* yang sederhana dan dapat menyatakan konklusi-konklusinya dengan cara yang sama. Suatu program yang dapat melakukan ini dikatakan memiliki suatu tatap muka bahasa natural.

**2.2.8.5 *Workplace***

*Workplace* merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam hasil-hasil antara dan kesimpulan yang dicapai. Ada tiga tipe keputusan yang dapat direkam, yaitu :

1. Rencana : bagaimana menghadapi masalah.

2. Agenda : aksi-aksi yang potensial yang sedang menunggu untuk dieksekusi.

3. Solusi : calon aksi yang akan dibangkitkan.

**2.2.8.6 Fasilitas Penjelasan**

Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Komponen ini menggambarkan penalaran sistem kepada pemakai. Fasilitas penjelasan dapat menjelaskan perilaku sistem pakar dengan menjawab pertanyaan-pertanyaan sebagai berikut :

1. Mengapa pertanyaan tertentu ditanyakan oleh sistem pakar ?

2. Bagaimana kesimpulan tertentu dipeoleh ?

3. Mengapa alternatif tertentu ditolak ?

**2.2.8.7 Perbaikan Pengetahuan**

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerja serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya.

**2.2.9 Pohon**

Suatu pohon (*tree*) adalah suatu hierarki struktur yang terdiri dari simpul (*node*) yang menyimpan informasi atau pengetahuan dan cabang yang menghubungkan simpul. Cabang disebut juga *link* atau *edge* dan simpul disebut juga dengan *vertex*. Akar simpul adalah simpul yang tertinggi dalam hierarki dan daun adalah simpul yang paling bawah. *Tree* dapat dianggap sebagai suatu tipe khusus dari jaringan semantik yang setiap simpulnya, kecuali akar pasti mempunyai satu simpul orang tua dan mempunyai nol atau lebih simpul anak.

Untuk tipe biasa dari *binarytree*, maksimum mempunyai dua anak untuk setiap simpul, dan sisi kiri dan kanan dari simpul anak dibedakan. Jika simpul mempunyai lebih dari satu orang tua maka disebut dengan jaringan.Gambar 2.6 menunjukkan *binarytree* yang mempunyai 0, 1 atau 2 cabang per simpul.



**Gambar 2.6 *BinaryTree***

Aplikasi dari *tree* adalah pembuatan keputusan dan biasa disebut dengan istilah *decisiontree* (pohon keputusan). Struktur keputusan dapat diterjemahkan secara mekanis kedalam kaidah produksi (*productionrule*) dengan menggunakan aturan *IF*-*THENRules*.

**2.2.10 IF-THEN Rules**

Dari pohon keputusan (*decisiontree*) yang telah tereduksi, dapat dikonversikan menjadi *IF-THEN rule*. Setiap *rule* pada *decisiontree* akan membentuk satu set *rulefinal*. Struktur dan sintaks penulisan rule adalah sebagai berikut :

RULE Label : Label berisi nama rule tersebut

IF : Sebagai penanda awal kondisi pada sebuah rule

THEN : Sebagai penanda awal kesimpulan pada sebuah rule

ELSE : Sebagai penanda awal alternatif kesimpulan pada sebuah

rule, bersifat opsional, jadi boleh tidak ada.

Sedangkan operator yang dapat digunakan pada *IF-THEN rule*adalah :

AND : Semua kondisi yang dihubungkan oleh operator ini harus bernilai benar, agar kondisi keseluruhan rule tersebut bernilai benar. Bila ada satu kondisi yang bernilai salah, maka rule tersebut bernilai salah.

OR : Bila semua kondisi yang dihubungkan oleh operator ini harus bernilai salah, maka kondisi keseluruhan rule tersebut bernilai salah. Bila ada satu atau lebih kondisi yang bernilai benar, keseluruhan rule tersebut bernilai benar.

Untuk gambar 2.6 diatas, maka *decision tree* dapat diterjemahkan kedalam aturan berikut :

IF pertanyaan = “Apakah arus listrik masuk ke dalam baterai ?”

And jawaban = “Tidak ”

THEN pertanyaan = “Apakah lampu adaptor berkedip-kedip saat di cars”

IF pertanyaan = “lampu baterai brkedip saat cars lalu mati?”

And jawaban = “Ya”

THEN pertanyaan = “Apakah adaptor panas tidak seperti biasanya ?”

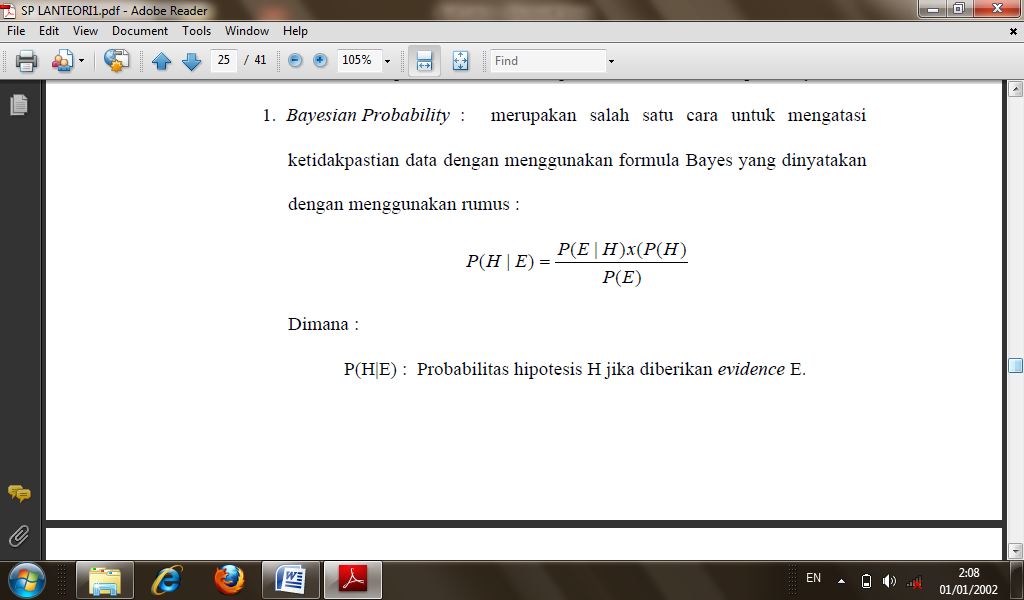
IF pertanyaan = “Apakah lampu power berkedip tak menentu ?”

And jawaban = “Ya”

THEN kesimpulan = “Adaptor”

**2.2.11 Ketidakyakinan**

Dalam penyusunan *rule*, *user* sering kali dihadapkan pada suatu premis yang meminta suatu kebenaran yang absolut. Sedangkan pada kenyataannya sering kali ditemui fakta yang kebenarannya tidak bersifat absolut. Untuk itu perlu ditentukan tingkat ketidakyakinan atau *uncertainty* terhadap suatu fakta. Misalnya pada sebuah pertanyaan, “Apakah pasien menderita nyeri pada perut?”, untuk memasukan suatu value kedalam *inferenceengine*, *user* terkadang merasa bingung, rasa nyeri yang dimaksudkan adalah rasa nyeri yang bagaimana, apakah nyeri sedikit saja sudah dapat dimasukkan sebagai *value*, atau nyeri yang hebat saja yang dimasukan sebagai value. Ataukah dalam *rule* tersebut rasa nyeri dapat diabaikan saha. Bisa saja dibuat pertanyaan pelengkap “Nyatakan intensitas nyeri yang dirasakan dalam skala 0 sampai 10”. Tetapi 2 orang yang berbeda dengan kasus yang sama bisa saja memberikan dua jawaban yang berbeda pula. Untuk itu diperlukan sebuah metode ketidakpastian atau *uncertainty* untuk memberikan toleransi jawaban atau pertanyaan yang jawabannya tidak pasti. Tingkat ketidakpastian ditentukan dari validitas *knowledgebaserule* dan validitas respon dari user. Ada 4 cara pendekatan dalam implementasi teori ketidakpastian yaitu :

1. *BayesianProbability* : merupakan salah satu cara untuk mengatasi ketidakpastian data dengan menggunakan formula Bayes yang dinyatakan dengan menggunakan rumus :

Dimana :

P(H|E) : Probabilitas hipotesis H jika diberikan evidence E.

P(H|E) : Probabilitas munculnya evidence E jika diketahui hipetosesis H. P(H) : Porabilitas hipotesis H tanpa memandang evidence apapun.

P(E) : Probabilitas evidece E.

2. EXSYS : ada tiga cara untuk menyatakan ketidakpastian yaitu dengan;

a. Sistem 0 atau 1, yaitu hanya ada dua kemungkinan yaitu ya dan tidak.

b. Sistem 0 sampai 10, yaitu menyatakan ketidakpastian dengan *confidence factor* dari 0 sampai 10. Apabila *confidence factor* suatu *rule* final berasal dari *confidencefactor rule*-*rule* lain, maka kesimpulan rule final tersebut memiliki *confidence factor* berupa rata-rata *confidence factor rule-rule* tersebut.

c. Sistem -100 sampai 100, yaitu menyatakan ketidakpastian dengan *confidence factor* -100 sampai 100. Apabila *confidence factor* bernilai -100, maka value suatu premis diyakini pasti salah.Apabila *confidence factor* bernilai 100 berarti kebenaran suatu premis diyakini pasti benar. Apabila *confidence factor* bernilai 0 berarti kebenaran suatu premis dianggap masih ragu-ragu.

3. *Fuzzy Set* : Tidak seperti dua pendekatan diatas, *fuzzyset* beranggapan bahwa ketidakpastian sebagai konsekuensi *input* dari *user*. *Confidencefactor* yang digunakan dalam metode ini berkisar antara 0 sampai 1.Untuk menentukan *fuzzyvalue* dari kesimpulan suatu *rule*, maka diambil nilai *fuzzyvalue* minimum dari semua premis yang ada kemudian dikalikan dengan *confidence factor rule* tersebut apabila premisnya dihubungkan dengan operator AND. Sedangkan apabila premisnya dihubungkan dengan operator OR, maka diambil nilai *fuzzyvalue* maksimum dari semua premis yang ada kemudian dikalikan dengan *confidence factor rule* tersebut.

Sebagai contoh adalah :

Rule 1 :

If disk drive berisik = yes (fuzzy value=0,8)

And menghasilkan bad sector bila memformat disket = yes

(fuzzy value = 0,3)

Then status disk drive = rusak (confidence factor = 0,9)

Maka tingkat keyakinan akan hasil kesimpulan rule tersebut adalah;

min(0,8 ; 0,3) x 0,9

0,3 x 0,9 = 0,27

Rule 2 :

If disk drive berisik = yes (fuzzy value=0,9)

Or menghasilkan bad sector bila memformat disket = yes

(fuzzy value = 0,2)

Then status disk drive = rusak (confidence factor = 0,7)

Maka tingkat keyakinan akan hasil kesimpulan rule tersebut adalah;

max(0,9 ; 0,2) x 0,7

0,9 x 0,7 = 0,63

4. Metode *Confidence Factor Union*

Dalam proses pengeksekusian *knowledge base*, nantinya kita akan dihadapkan dalam *multiple value* yang mengacu pada satu kesimpulan yang sama. Misalkan pada penggabungan dua *value*, *value* pertama memiliki *confidencefactor* sebesar 0,2 dan *value* kedua memiliki *confidencefactor* sebesar 0,5 maka *confidencefactor* gabungan kedua *value* tersebut adalah :

C(cf) = (0,2 + 0,5) – (0,2 x 0,5) = 0,6.

**2.3 Siklus Pengembangan Sistem**

1. **Analisis Sistem**

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi dengan maksud untuk mengidentifikasikan dan mengevaluasikan permasalahan-permasalahan, hambatan-hambatan yang terjadi dalam kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Didalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem sebagai berikut :

1. Identify, yaitu mengidentifikasi masalah.

2. Understand, yaitu memahami kerja dari sistem yang ada.

3. Analyze, yaitu menganalisa sistem.

4. Report, yaitu membuat laporan hasil analisis.

Model analisis merupakan representasi teknis yang pertama dari sistem, saat ini ada dua yang mendominasi pemodelan analisis, yaitu :

1. Analisis Terstruktur. Dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional, kita menciptakan model yang menggambarkan muatan dan aliran inforasi, kita membagi sistem secara fungsional dan secara *behavior*, dan menggambarkan esensi dari apa yang harus dibangun.

2. Analisis Berorientasi Objek, mendefinisikan semua kelas yang relevan terhadap masalah berserta operasi-operasi dan atribut-atribut yang diasosiasikan dengan kelas itu, keterhubungan dikelas-kelas dan perilaku yang dimilikinya..

Alat-alat yang digunakan dalam analisis terstruktur adalah :

1. **Diagram Konteks**

Diagram konteks adalah arus data yang berfungsi untuk menggambarkan keterkaitan aliran-aliran data antara sistem dengan bagian-bagian luar. Kesatuan luar ini merupakan sumber arus data atau tujuan data yang berhubungan dengan sistem informasi tersebut.Diagram konteks memberikan batasan yang jelas mengenai besaran-besaran entitas yang berada diluar sistem yang sedang dibuat, artinya diagram ini menggambarkan secara jelas batasan-batasan dari sebuah sistem yang sedang dibuat.

1. **Diagram Alir Data (Data Flow Diagram)**

Merupakan representasi grafik dari suatu sistem yang menunjukkan proses atau fungsi, aliran data, tempat penyimpanan data dan entitas eksternal. *DataFlowDiagram* (DFD) juga digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan. Beberapa simbol yang digunakan di DFD adalah :

1. External Entity, merupakan kesatuan dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan input atau memeriksa output dari sistem. Disimbolkan dengan suatu notasi kotak. (Jogiyanto HM, 2005 : 701)

**Gambar 2.7 Kesatuan Luar (*ExternalEntity*)**

1. Data Flow (arus data), di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (process), simpanan data (data store) dan kesatuan luar (external entity). Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. (Jogiyanto HM, 2005 : 701)

**Gambar 2.8 Arus Data (*DataFlow*)**

1. *Process* (proses). Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus dara yang masuk kedalam proses untuk dihasilkan arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. (Jogiyanto HM, 2005 : 705)

Identifikasi

Nama

Proses

**Gambar 2.9 Proses (*Process*)**

1. Suatu proses adalah kegiatan atau kerja yang dilakukan orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. (Jogiyanto HM, 2005 : 705)

Media Data Store

**Gambar 2.10 Penyimpanan Data (*DataStorage*)**

1. **Kamus Data**

Salah satu komponen kunci dalam sistem manajemen *database* adalah *file* khusus yang disebut kamus data (*datadictionary*). Kamus data merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir berisi informasi tentang struktur *database*.

Untuk setiap elemen data yang disimpan dalam *database* seperti nomor pokok pegawai, diuraikan secara lengkap mulai dari nama, tempat penyimpanan, program komputer yang berhubungan dan lain-lain. Kamus data biasanya dipelihara secara otomatis oleh sistem manajemen *database*. Cara mendefinisikan kamus data yaitu :

a. Menggambarkan arti aliran data atau penyimpanan yang ditunjuk dalam DFD.

b. Menggabungkan komponen dari kumpulan data yang mengalir yaitu kumpulan komponen yang mungkin bisa dipecah lagi menjadi data elementer.

c. Menggambarkan data yang tersimpan.

d. Menentukan nilai dibagian elementer dari informasi yang relevan di DFD dan data storenya.

1. **Desain Sistem**

Setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analis sistem untuk memikirkan bagaimana membentuk sistem tersebut. Tahap ini disebut dengan desain sistem (*systems design*).

Menurut Verzello dan Reuter, dalam Jogiyanto (2005:196) desain sistem adalah tahap setelah analisis dari siklus pengembangan sistem; pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi menggambarkan bagaimana suatu sistem dibentuk.

Menurut Burch dan Grudnitski, dalam Jogiyanto ( 2005:196 ) desain sistem dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

Tahap desain sistem mempunyai dua tujuan utama, yaitu:

1. Untuk memenuhi kebutuhan kepada pemakai sistem.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya.

Desain sistem dapat dibagi dalam dua bagian, yaitu desain sistem secara umum (*general systems design*) dan desain sistem terinci (*detailed system design*).

* + 1. Desain Sistem Secara Umum

Tujuan dari desain sistem secara umum adalah untuk memberikan gambaran secara umum kepada *user* tentang sistem yang baru, yang mana merupakan persiapan dari desain sistem secara rinci. Desain secara umum dilakukan oleh analis sistem untuk mengidentifikasikan komponen-komponen sistem informasi yang akan didesain secara rinci oleh pemrogram komputer dan ahli teknik lainnya.

Pada tahap ini, komponen-komponen sistem informasi dirancang dengan tujuan untuk dikomunikasikan kepada *user*. Komponen sistem informasi yang didesain adalah model, output, input, database, teknologi dan kontrol.

* + 1. Desain Sistem secara Rinci (*Detailed System Design****)***
* Desain Output Terinci

Desain output terinci dimaksudkan untuk mengetahui bagaimana dan seperti apa bentuk output-output dari sistem yang baru. Desain Output Terinci terbagi atas dua, yaitu desain output berbentuk laporan di media kertas dan desain output dalam bentuk dialog di layar terminal.

* 1. Desain output dalam bentuk laporan

Desain ini dimaksudkan untuk menghasilkan output dalam bentuk laporan di media kertas. Bentuk laporan yang paling banyak digunakan adalah dalam bentuk tabel dan berbentuk grafik atau bagan.

* 1. Desain output dalam bentuk dialog layar terminal

Desain ini merupakan rancang bangun dari percakapan antara pemakai sistem (*user*) dengan komputer. Percakapan ini dapat terdiri dari proses memasukkan data ke sistem, menampilkan output informasi kepada *user*, atau keduanya

Beberapa stategi dalam membuat layar dialog terminal :

* + 1. Dialog pertanyaan / jawaban
    2. Menu

Menu banyak digunakan karena merupakan jalur pemakai yang mudah dipahami dan mudah digunakan. Menu berisi beberapa alternatif atau *option* atau pilihan yang disajikan kepada user. Pilihan menu akan lebih baik bila dikelompokkan sesuai fungsinya.

* Desain Input Terinci

Masukan merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan oleh organisasi. Data hasil dari transaksi merupakan masukan untuk sistem informasi. Hasil dari sistem informasi tidak lepas dari data yang dimasukkan. Desain input terinci dimulai dari desain dokumen dasar sebagai penangkap input yang pertama kali. Jika dokumen dasar tidak didesain dengan baik, kemungkinan input yang tercatat dapat salah bahkan kurang.

Fungsi dokumen dasar dalam penanganan arus data :

1. Dapat menunjukkan macam dari data yang harus dikumpulkan dan ditangkap
2. Data dapat dicatat dengan jelas, konsisten dan akurat

3. Dapat mendorong lengkapnya data, disebabkan data yang dibutuhkan disebutkan satu persatu di dalam dokumen dasarnya

* Desain Database Terinci

Basis data *(database)* merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. Database merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Penerapan database dalam sistem informasi disebut *database system.*

Sistem basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam didalam suatu organisasi. Dengan sistem basis data ini tiap-tiap orang atau bagian dapat memandang database dari beberapa sudut pandang yang berbeda. Bagian kredit dapat memandangnya sebagai data piutang, bagian penjualan dapat memandangnya sebagai data penjualan, bagian personalia dapat memandangnya sebagai data karyawan, bagian gudang dapat memandangnya sebagai data persediaan. Semuanya terintegrasi dalam sebuah data yang umum. Berbeda dengan sistem pengolahan data tradisional, sumber data ditangani sendiri-sendiri untuk tiap aplikasinya.

Pada tahap ini, desain database dimaksudkan untuk mendefinisikan isi atau struktur dari tiap-tiap file yang telah diidentifikasikan di desain secara umum.

* Desain Teknologi

Tahap desain teknologi terbagi atas dua, yaitu desain teknologi secara umum dan terinci. Pada tahap ini kita menentukan teknologi yang akan dipergunakan dalam menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi yang dimaksud meliputi :

1. Perangkat keras (*hardware*), yang terdiri dari alat masukan, alat pemroses, alat output dan simpanan luar
2. Perangkat lunak (*software*), yang terdiri dari perangkat lunak sistem operasi (O*perating system*), perangkat lunak bahasa (*language software*) dan perangkat lunak (*application software*)
3. Sumber daya manusia (*brainware*), misalnya operator komputer, pemrogram, spesialis telekomunikasi, sistem analis dan lain sebagainya.

Desain teknologi sangat diperlukan pada tahap implementasi dan pengujian untuk membuktikan bahwa sistem dapat berjalan secara semestinya.

1. **Implementasi Sistem**

Whitten, et al (2004: 34) mengungkapkan ” Sistem Implementasi adalah konstruksi, instalasi, pengujian, dan pengiriman sistem kedalam produksi (artinya operasi sehari-hari)”.

Sistem telah dianalisa dan didesain secara rinci dan teknologi telah diseleksi dan dipilih. Tiba saatnya sekarang sistem untuk diimplementasikan (diterapkan). Tahap implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperasikan. Tahap implementasi sistem dapat terdiri dari langkah-langkah sebagai berikut:

1. Menerapkan Rencana Implementasi

Rencana implementasi merupakan kegiatan awal dari tahap implementasi sistem. Rencana implementasi dimaksudkan terutama untuk mengatur biaya dan waktu yang dibutuhkan selama tahap implementasi.

1. Melakukan Kegiatan Implementasi

Kegiatan implementasi dilakukan dengan dasar kegiatan yang telah direncanakan dalam rencana implementasi. Kegiatan-kegiatan yang dapat dilakukan dalam tahap implementasi ini adalah sebagai berikut :

Telah diketahui bahwa manusia merupakan faktor yang perlu dipertimbangkan dalam sistem informasi. Jika sistem informasi ingin sukses, maka personil-personil yang terlibat harus diberi pengertian dan pengetahuan yang cukup tentang sistem informasi dan posisi serta tugas mereka.

* 1. Persiapan Tempat dan Instalasi Perangkat Keras dan Perangkat Lunak

Jika peralatan baru akan dimiliki, maka tempat atau ruangan untuk peralatan ini perlu dipersiapkan terlebih dahulu. Keamanan fisik dari tempat ini perlu juga dipertimbangkan. Sistem komputer yang besar membutuhkan tempat dengan lingkungan yang lebih harus diperhitungkan. Langkah selanjutnya setelah persiapan fisik tempat adalah menginstalasi perangkat keras yang sudah dikirim dan menginstalasi perangkat lunak yang sudah ada.

* 1. Pemrograman dan Pengetesan Sistem

Pemrograman merupakan kegiatan menulis kode program yang akan dieksekusi oleh komputer. Kode program yang ditulis oleh pemrogram harus berdasarkan dokumentasi yang disediakan oleh analis sistem hasil dari desain sistem secara rinci. Sebelum program diterapkan, maka program harus terlebih dahulu bebas dari kesalahan-kesalahan. Oleh sebab itu, program harus diuji untuk menemukan kesalahan-kesalahan yang mungkin dapat terjadi. Program dites untuk tiap-tiap modul dan dilanjutkan dengan pengetesan untuk semua modul yang telah dirangkai.

* 1. Pengetesan Sistem

Pengetesan sistem biasanya dilakukan setelah pengetesan program. Pengetesan sistem dilakukan untuk memeriksa kekompakan antar komponen sistem yang diimplementasikan. Tujuan utama dari pengetesan sistem ini adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan.

1. **Operasi dan Pemeliharaan**

Setelah masa sistem berjalan sepenuhnya menggantikan sistem lama, sistem memasuki pada tahapan operasi dan pemeliharaan. Zwass (1999) membagi pemeliharaan perangkat lunak menjadi tiga macam, yaitu :

1. Pemeliharaan Perfektif

Pemeliharaan Perfektif ditujukan untuk memperbarui sistem lama sebagai tanggapan atas perubahan kebutuhan pemakai dan kebutuhan organisasi, meningkatkan efesiensi sistem, dan memperbaiki dokumentasi.

1. Pemeliharaan Adaptif

Pemeliharaan Adaptif berupa perubahan aplikasi untuk menyesuaikan diri terhadap lingkungan perangkat keras dan perangkat lunakbaru. Sebagai contoh pemeliharaan ini dapat berupa perubahan aplikasi dari *mainframe* ke lingkungan *client/ server* atau mengkonversi dari sistem berbasis berkas ke lingkungan basis data.

1. Pemeliharaan Korektif

Pemeliharaan korektif berpa pembetulan atas kesalahan-kesalahan yang ditemukan pada saat sistem berjalan.

**2.4 Teknik Pengujian Sistem**

**2.4.1 *WhiteBox***

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean.

Pengujian sistem/ perangkat lunak memiliki sejumlah aturan yang berfungsi sebagai sasaran pengujian, diantaranya adalah sebagai berikut :

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan
2. *Test case* yang baik adalah *test case* yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

Pengujian *White Box*, adalah metode pengujian yang menggunakan struktur kontrol desain prosedur untuk memperoleh *test case*. Dengan menggunakan metode *white box*, perekayasa sistem dapat melakukan *test case* yang memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali, menggunakan semua keputusan logis pada sisi *true* dan *false*, mengeksekusi semua *loop* pada batasan mereka dan pada batas operasional mereka, dan menggunakan stuktur data internal untuk menjamin validitasnya. Pengujian *Basis Path* adalah teknik pengujian *white box* yang diusulkan pertama kali oleh Tom McCabe. Metode *basis path* ini memungkinkan desainer *test case* mengukur kompleksitas logis dari desain prosedural dan menggunakannya sebagai pedoman untuk menetapkan basis set dari jalur eksekusi. (Roger S. Pressman, 2002 : 536).

9

10

6

7

8

1

2

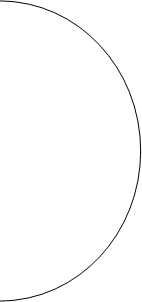
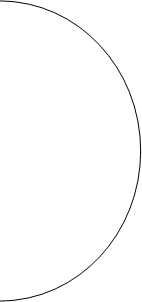
3

5

4

**Gambar 2.11**  Bagan Alir

Bagan alir digunakan untuk menggambarkan struktur kontrol program. Dan untuk menggambarkan grafik alir, harus memperhatikan representasi desain prosedural pada bagan alir. Pada gambar dibawah ini, grafik alir memetakan bagan alir tersebut ke dalam grafik alir yang sesuai (dengan mengasumsikan bahwa tidak ada kondisi senyawa yang diisikan di dalam diamond keputusan dari bagan alir tersebut). Masing-masing lingkaran, yang disebut *simpul* grafik alir, merepresentasikan satu atau lebih statemen prosedural. Urutan kotak proses dan permata keputusan dapat memetakan simpul tunggal. Anak panah tersebut yang disebut *edges* atau *links*, merepresentasikan aliran kontrol dan analog dengan anak panah bagan alir. Edge harus berhenti pada suatu simpul, meskipun bila simpul tersebut tidak merepresentasikan statemen prosedural. (Roger S. Pressman, 2002 : 536).



R3

R2

R1

R4

**Gambar 2.12** Grafik Alir

*Kompleksitas siklomatis* adalah metriks perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program.Bila metriks ini digunakan dalam konteks metode pengujian basis path, maka nilai yang terhitung untuk kompleksitas siklomatis menentukan jumlah *jalur independen*. Jalur independen adalah jalur yang melalui program yang mengintroduksi sedikitnya satu rangkaian statemen proses baru atau suatu kondisi baru. Bila dinyatakan dengan terminologi grafik alir, jalur independen harus bergerak sepanjang paling tidak satu edge yang tidak dilewatkan sebelum jalur tersebut ditentukan. Sebagai contoh, serangkaian jalur independen untuk grafik alir yang ditunjukkan pada gambar 2.7 adalah :

Jalur 1 : 1 – 11

Jalur 2 : 1 – 2 – 3 – 4 – 5 – 10 – 1 – 11

Jalur 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Jalur 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Jalur 1, 2, 3, dan 4 yang ditentukan di atas terdiri dari sebuah basis set untuk grafik alir pada gambar 2.7. Bagaimana kita tahu banyaknya jalur yang dicari? Komputasi kompleksitas siklomatis memberikan jawaban. Fondasi kompleksitas siklomatis adalah teori grafik, dan memberi kita metriks perangkat lunak yang sangat berguna. Kompleksitas dihitung dalam salah satu dari tiga cara berikut :

1. Jumlah region grafik alir sesuai dengan kompleksitas siklomatis.
2. Kompleksitas siklomatis, *V(G),* untuk grafik alir *G* ditentukan sebagai *V(G)* = *E – N* + 2 di mana *E* adalah jumlah edge grafik alir dan *N* adalah jumlah simpul grafik alir.
3. Kompleksitas siklomatis, *V(G),* untuk grafik alir G juga ditentukan sebagai *V(G)* = P + 1, dimana P adalah jumlah simpul predikat yang diisikan dalam grafik alir *G*.

Pada gambar 2.7 grafik alir, kompleksitas siklomatis dapat dihitung dengan menggunakan masing-masing dari algoritma yang ditulis di atas :

1. Grafik alir mempunyai 4 region
2. *V(G)* = 11 edge – 9 simpul + 2 = 4
3. *V(G)* = 3 simpul yang diperkirakan + 1 =4

Dengan demikian, kompleksitas siklomatis dari grafik alir pada gambar 2.7 adalah 4. Yang lebih penting, nilai untuk *V(G)* memberi kita batas atas untuk jumlah jalur independen yang membentuk basis set, dan implikasinya, batas atas jumlah pengujian yang harus didesain dan dieksekusi untuk menjamin semua statemen program.

**2.4.2 *BlackBox***

*Black boxaproach* adalah suatu sistem dimana input dan outputnya dapat didefinisikan tetapi prosesnya tidak diketahui atau tidak terdefinisi. Metode ini hanya dapat dimengerti oleh pihak dalam (yang menangani sedangkan pihak luar hanya mengetahui masukan dan hasilnya). Sistem ini terdapat pada sub sistem tingkat terendah.

Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari *software*. Karena itu ujicoba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *whitebox*. Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang.

2. Kesalahan interface.

3. Kesalahan dalam struktur data atau akses database eksternal.

4. Kesalahan performa dan kesalahan inisialisasi.

Tidak seperti metode whitebox yang dilaksanakan diawal proses, ujicoba blackbox diaplikasikan dibeberapa tahapan berikutnya. Karena ujicoba blackbox dengan sengaja mengabaikan struktur kontrol, sehingga perhatiannya difokuskan pada informasi *domain*. Ujicoba didesain untuk dapat menjawab pertanyaan-pertanyaan berikut :

1. Bagaimana validitas fungsionalnya diuji?

2. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik ?

3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu ?

4. Bagaimana batasan-batasan kelas data diisolasi?

5. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem?

6. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem?

Dengan mengaplikasikan ujicoba blackbox, diharapkan dapat menghasilkan sekumpulan kasus uji yang memenuhi kriteria berikut :

1. Kasus uji yang berkurang, jika jumlahnya lebih dari 1, maka jumlah dari uji kasus tambahan harus didesain untuk mencapai ujicoba yang cukup beralasan.

2. Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan, dari pada kesalahan yang terhubung hanya dengan suatu ujicoba yang spesifik.

**2.5 Software Pendukung**

**2.5.1 PHP (PHP; *HypertextPreprocessor*)**

PHP dikenal sebagai sebuah bahasa *scripting* yang menyatu dengan tag-tag HTML, ditempakan dalam *server* dan dieksekusi di *server*, dan digunakan untuk membuat halaman web yang dinamis, yang hasilnya dikirimkan ke *client* tempat pemakai menggunakan *browser*.

PHP diciptakan oleh Rasmus Lerdoff ([rasmus@php.net](mailto:rasmus@php.net)), yang pada awalnya digunakan untuk melihat siapa saja yang mengunjungi website-nya dan untuk melihat biodata pengunjung. Pada awal tahun 195, PHP versi pertama di release dan dikenal sebagai *tool personal home page*. Namun kini PHP telah mengalami banyak kemajuan. Sehingga secara resmi PHP merupakan kependekan dari PHP: *hypertext preprocessor*, merupakan bahasa pemrograman *script* web *server-side* yang disisipkan pada HTML. Ini berarti bahwa data akan diekstrak dari *database* dalam sebuah halam akan diproses terlebih dahulu sebelum dikirim ke *client* (ditampilkan di halaman web).

Kelebihan-kelebihan PHP dibanding program lain;

1. Gratis/*free* karena PHP merupakan *open source software*.
2. Tidak ada virus yang menginfeksi program PHP, sampai saat ini program PHP belum dapat diinfeksi virus, kebanyakan virus menginfeksi file berekstensi \*.exe, sangat awet dan aman.
3. Sangat multi user program PHP tidak akan bentrok dengan program lain yang sama-sama menggunakan program dalam satu jaringan.
4. PHP mampu membuat halaman dinamis, memanipulasi *form*, dan dapat dihubungkan dengan *database*.
5. Sangat stabil di semua *operating sistem*, program PHP walaupun dipakai dalam waktu yang sangat lama tidak akan memberatkan sistem dan tidak akan mempengaruhi komputer untuk berjalan sangat lambat, sangat cocok diterapkan pada komputer yang selalu nyala 24 jam.



**Gambar 2.13 PHP**

**2.5.2 MySQL**

MySQL adalah sebuah [perangkat lunak](http://id.wikipedia.org/wiki/Perangkat_lunak) sistem manajemen [basis data](http://id.wikipedia.org/wiki/Basis_data)[SQL](http://id.wikipedia.org/wiki/SQL) (*database management system*) atau DBMS yang [*multithread*](http://id.wikipedia.org/w/index.php?title=Multithread&action=edit&redlink=1), [*multi*-*user*](http://id.wikipedia.org/wiki/Multi-user), dengan sekitar 6 juta instalasi di seluruh dunia. [MySQL AB](http://id.wikipedia.org/w/index.php?title=MySQL_AB&action=edit&redlink=1) membuat MySQL tersedia sebagai [perangkat lunak gratis](http://id.wikipedia.org/wiki/Perangkat_lunak_gratis) dibawah lisensi [GNU *General Public License*](http://id.wikipedia.org/wiki/GNU_General_Public_License)(GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti [*Apache*](http://id.wikipedia.org/wiki/Apache_HTTP_Server), dimana perangkat lunak dikembangkan oleh komunitas umum, dan [hak cipta](http://id.wikipedia.org/wiki/Hak_cipta) untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial [Swedia](http://id.wikipedia.org/wiki/Swedia)[MySQL AB](http://id.wikipedia.org/w/index.php?title=MySQL_AB&action=edit&redlink=1), dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang [Finlandia](http://id.wikipedia.org/wiki/Finlandia) yang mendirikan MySQL AB adalah: [David Axmark](http://id.wikipedia.org/w/index.php?title=David_Axmark&action=edit&redlink=1), [Allan Larsson](http://id.wikipedia.org/w/index.php?title=Allan_Larsson&action=edit&redlink=1), dan [Michael "Monty" Widenius](http://id.wikipedia.org/w/index.php?title=Michael_%28Monty%29_Widenius&action=edit&redlink=1).

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional ([*RDBMS*](http://id.wikipedia.org/wiki/RDBMS)) yang didistribusikan secara gratis dibawah [lisensi GPL](http://id.wikipedia.org/wiki/GPL) (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; [SQL](http://id.wikipedia.org/wiki/SQL) (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata ([DBMS](http://id.wikipedia.org/wiki/DBMS)) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web ([*wordpress*](http://id.wikipedia.org/wiki/Wordpress)), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai [perangkat lunak sumber terbuka](http://id.wikipedia.org/wiki/Perangkat_lunak_sumber_terbuka), dibawah lisensi [GPL](http://id.wikipedia.org/wiki/GPL) sehingga dapat digunakan secara gratis.
3. *Multi*-*user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *'Performance tuning'*, MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed* / *unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level [*subnetmask*](http://id.wikipedia.org/w/index.php?title=Subnetmask&action=edit&redlink=1), nama[*host*](http://id.wikipedia.org/w/index.php?title=Host&action=edit&redlink=1), dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol [TCP/IP](http://id.wikipedia.org/wiki/TCP/IP), [Unix soket](http://id.wikipedia.org/w/index.php?title=Unix_soket&action=edit&redlink=1) ([UNIX](http://id.wikipedia.org/wiki/UNIX)), atau [Named Pipes](http://id.wikipedia.org/w/index.php?title=Named_Pipes&action=edit&redlink=1) ([NT](http://id.wikipedia.org/wiki/NT)).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi [API](http://id.wikipedia.org/wiki/API) (*Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan ([*tool*](http://id.wikipedia.org/wiki/Tool)) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam [PostgreSQL](http://id.wikipedia.org/wiki/PostgreSQL) ataupun [Oracle](http://id.wikipedia.org/wiki/Oracle).

MySQL sangat populer dalam [aplikasi web](http://id.wikipedia.org/wiki/Aplikasi_web) seperti [MediaWiki](http://id.wikipedia.org/wiki/MediaWiki) (perangkat lunak yang dipakai [Wikipedia](http://id.wikipedia.org/wiki/Wikipedia) dan proyek-proyek sejenis) dan [PHP-Nuke](http://id.wikipedia.org/w/index.php?title=PHP-Nuke&action=edit&redlink=1) dan berfungsi sebagai komponen basis data dalam [LAMP](http://id.wikipedia.org/wiki/LAMP). Popularitas sebagai aplikasi web dikarenakan kedekatannya dengan popularitas [PHP](http://id.wikipedia.org/wiki/PHP), sehingga seringkali disebut sebagai Dynamic Duo.

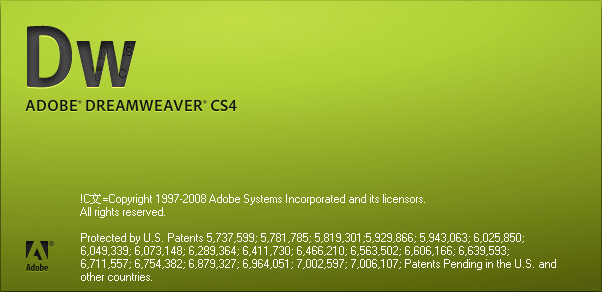
**Gambar 2.14 MySQL**

**2.5.3 Adobe Dreamweaver**

Adobe Dreamweaver CS4, atau biasa disebut “Dreamweaver CS4”, adalah sebuah perangkat lunak aplikasi untuk mendesain dan membuat halaman web. Dengan menggunakan Dreamweaver CS4, ketika membuat sebuah halaman web, Anda tidak perlu lagi mengetik kodekode HTML atau kode-kode lainnya secara manual. Anda cukup melakukan klik beberapa kali, maka simsalabin, halaman web yang Anda inginkan sudah jadi.

Ketika Anda mempelajari buku SGS Pemrograman Web dengan HTML, CSS, dan JavaScript, Anda harus menuliskan beberapa baris kode HTML yang panjang, hanya untuk sekadar membuat sebuah Tabel. Namun, dengan Dreaweaver CS4, Anda dapat membuat tabel hanya dengan melakukan dua kali klik saja. Dreamweaver CS4 akan menciptakan kode-kode HTML yang sesuai untuk membuat tabel yang Anda inginkan tersebut. Selain HTML, Dreamweaver CS4 juga mendukung CSS, JavaScript, PHP, ASP, dan bahasa pemrograman lainnya untuk membuat web. Hal ini akan sangat menguntungkan Anda. Sebagai contoh, jika dahulu Anda harus mengetikkan kode-kode CSS untuk membuat Style tertentu, maka dengan Dreaweaver CS4, Anda cukup melakukan klik beberapa kali saja.

Dreaweaver CS4 adalah versi terbaru dari keluarga Dreamweaver. Versi pertamanya sendiri diluncurkan sekitar tahun 1994 oleh Macromedia Inc. Dalam versi terbaru ini, banyak sekali fasilitas baru yang ditambahkan. Contohnya, Anda akan dapat membuat dan menggunakan Style dalam CSS dengan mudah dan fleksibel. Panel untuk pengolahan CSS juga sudah diperbarui dan lebih mudah digunakan. Dreaweaver CS4 juga menyediakan beberapa template halaman web baru, termasuk fasilitas Starter Pages. (Dominikus Juju, 2007;1-2)



**Gambar 2.15** Dreamweaver

**2.5.4 Adobe Photoshop**

Adobe Photoshop, atau biasa disebut Photoshop, adalah perangkat lunak editor citra buatan Adobe Systems yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh [fotografer](http://id.wikipedia.org/wiki/Fotografer) [digital](http://id.wikipedia.org/wiki/Digital) dan perusahaan [iklan](http://id.wikipedia.org/wiki/Iklan) sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolah gambar/foto, dan, bersama [Adobe Acrobat](http://id.wikipedia.org/wiki/Adobe_Acrobat), dianggap sebagai produk terbaik yang pernah diproduksi oleh Adobe Systems. Versi kedelapan aplikasi ini disebut dengan nama Photoshop CS (Creative Suite), versi sembilan disebut Adobe Photoshop CS2, versi sepuluh disebut Adobe Photoshop CS3 , versi kesebelas adalah Adobe Photoshop CS4 dan versi yang terakhir (keduabelas) adalah Adobe Photoshop CS5. Photoshop tersedia untuk [Microsoft Windows](http://id.wikipedia.org/wiki/Microsoft_Windows), [Mac OS X](http://id.wikipedia.org/wiki/Mac_OS_X), dan [Mac OS](http://id.wikipedia.org/wiki/Mac_OS); versi 9 ke atas juga dapat digunakan oleh sistem operasi lain seperti [Linux](http://id.wikipedia.org/wiki/Linux) dengan bantuan perangkat lunak tertentu seperti [CrossOver](http://id.wikipedia.org/w/index.php?title=CrossOver&action=edit&redlink=1). Pada tahun 1987, [Thomas Knoll](http://id.wikipedia.org/w/index.php?title=Thomas_Knoll&action=edit&redlink=1), mahasiswa PhD di Universitas Michigan, mulai menulis sebuah program pada Macintosh Plus-nya untuk menampilkan gambar grayscale pada layar monokrom. Program ini, yang disebut Display, menarik perhatian saudaranya [John Knoll](http://id.wikipedia.org/w/index.php?title=John_Knoll&action=edit&redlink=1), seorang karyawan di Industrial Light & Magic, yang merekomendasikan Thomas agar mengubah programnya menjadi program penyunting gambar penuh. Thomas mengambil enam bulan istirahat dari studi pada tahun 1988 untuk berkolaborasi dengan saudaranya pada program itu, yang telah diubah namanya menjadi ImagePro. Setelah tahun itu, Thomas mengubah nama programnya menjadi Photoshop dan bekerja dalam jangka pendek dengan produsen scanner Barneyscan untuk mendistribusikan salinan dari program tersebut dengan slide scanner; "total sekitar 200 salinan Photoshop telah dikirimkan" dengan cara ini. Selama waktu itu, John bepergian ke Silicon Valley di California dan memberikan demonstrasi program itu kepada insinyur di [Apple Computer Inc.](http://id.wikipedia.org/wiki/Apple_Inc.) dan Russell Brown, direktur seni di [Adobe](http://id.wikipedia.org/wiki/Adobe_Systems). Kedua demonstrasi itu berhasil, dan Adobe memutuskan untuk membeli lisensi untuk mendistribusikan pada bulan September 1988.[[1]](http://id.wikipedia.org/wiki/Adobe_Photoshop#cite_note-history-0) Sementara John bekerja pada plug-in di California, Thomas tetap di Ann Arbor untuk menulis kode program. Photoshop 1.0 dirilis pada 1990 khusus untuk Macintosh. (Wikipedia/diakses 30 Januari 2011)



**Gambar 2.16** Adobe Photoshop CS

**2.5. Laptop (*Notebook*)**

Perkembangan teknologi saat ini membawa pengaruh luar biasa bagi gaya hidup setiap orang. Terutama dibidang teknologi informasi baik segi perangkat keras maupun perangkat lunak. Tidak dipungkiri lagi perkembangan komputer cukup mengesankan, dahulu komputer yang hanya digunakan untuk keperluan kantor, sekarang berkembang ke berbagai bidang seperti design, game, dan hiburan. Pada awal-awal pertamanya komputer hanya dapat digunakan terbatas di meja saja atau yang sering disebut sebagai computer desktop  (komputer duduk). Dengan terus berkembangnya pola pikir dan peradaban manusia, muncul konsep atau ide bahwa komputer tidak hanya digunakan sebatas di atas meja saja, tetapi harus dapat digunakan untuk berpindah-pindah tempat (mobile).

Berikut ini awal pentahapan perkembangan laptop yang dikutip dari (www.kaskus.us) adalah sebagai berikut :

**Tahap pertama** Hal ini berawal dari ide  pada 1970-an oleh Alan Kay dari Xerox Palo Alto Research Center memiliki suatu visi untuk sebuah komputer jinjing yang tidak memerlukan kabel, seukuran buku catatan. Ia menamakannya Dynabook. Dynabook milik Kay ini diciptakan dengan kemampuan network wireless. Dan hal ini mulai mengerakkan roda perkembangan sebuah komputer jinjing yang sebenarnya seperti sekarang ini.

**Tahap kedua** Di era 1979, William Moggridge dari korporasi Grid System menciptakan komputer jinjing yang pertama: The Grid Compass Computer 1109. Memiliki 340 kilobyte memory, sebuah kotak die-cast magnesium dan sebuah layar lipat electrominescent. Nasa membeli banyak darinya pada kisaran $800 perbiji, untuk digunakan pada program ruang angkasa.

**Tahap ketiga** pada 1983 Gavilan Computer memproduksi sebuah komputer laptop yang bekerja dengan baik sebaik MS-DOS pada processor 8088. Diproduksi dengan 64K RAM, dapat di up-grade ke 128K. Dengan berat sekitar 9 pounds, memiliki touchpad/mouse didepan sebelah atas keyboard.

**Tahap keempat** Pada tahun 1984 Apple memperkenalkan Apple model IIc. Apple IIc adalah sebuah notebook-seukuran komputer, akan tetapi bukan Laptop sebenarnya. Memiliki microprosesor 65C02, Memory 128 kilobytes dan floppy drive 5.25-inch internal, dua port serial, port mouse, modem card, power supply external, dan dapat dilipat. Komputer itu sendiri memiliki berat 10 sampai 12 lb (5kg), akan tetapi monitornya sedikit lebih berat. Apple IIc memiliki monitor monochrome 9-inci dan LCD panel optional.

Kombinasi antara komputer/panel LCD telah menjadikannya sebuah komputer portable. Apple IIc dipasarkan ke rumah dan bidang pendidikan, dan meraup sukses untuk sekitar lima tahun. Perusahaan lain seperti IBM, memperkenalkan komputer jinjing lainnya pada tahun berikutnya. Komputer jinjing pertama yang dilepas secara komersial adalah IBM PC, tidak seperti Apple IIc, PC convertible adalah benar-benar sebuah computer Laptop. Seperti Komputer Gavilan diperkenalkan pada tahun 1986.

Dengan fitur: Microprosesor 8088,Memory 256 kilobyte,Dua floppy drive 3,5-inch (8.9 cm),Sebuah layar LCD,Paralel dan port serial printer,Ruang untuk modem internal danSebuah software dasar mencakup pengolahan kata, penangalan kalendar, buku alamat, dan software kalkulator. Dengan berat yang kokoh 12 lbs (5,4 kg), PC ini dijual pada kisaran $3500. Menjadi komputer jinjing yang pertama dengan desain clamshell seperti laptop sekarang. Sukses dari PC yang dapat dirubah menjadi katalisator untuk parapesaing seperti Compaq dan Toshiba mengubah desain clamshell ini menjadi komputer jinjing mereka sendiri. Dan mulai jaman dari sebuah komputer laptop.

Dari situlah laptop muncul dan berkembang seperti sekarang ini, yang sudah mempunyai tampilan yang lebih cangih dan lebih baik. Dari sejarah perkembangannya laptop yang telah dijelaskan bisa di artikan sebagai komputer jinjing atau komputer yang dapat di bawah kemana-mana.

Laptop atau istilah kerennya notebook, merupakan komputer kecil yang dapat dibawa kemana-mana dengan mudah atau biasa disebut portabel yang terintergrasi dengan casing. Baterai atau adaptor merupakan sumber listrik dari laptop ketika kita tidak menghubungkannya dengan arus listrik seperti halnya komputer. Baterai laptop umumnya dapat bertahan sekitar satu hingga enam jam, tergantung dari pemakaian, spesifikasi, dan ukuran baterai. Kenapa laptop mudah dibawa kemana-mana? Karena berat laptop cukup ringan, antara 1 hingga 6 kg tergantung bahan, ukuran, dan spesifikasinya. Laptop berukuran kecil dan ringan jika dibandingkan dengan komputer, memiliki fungsi dan komponen yang sama dengan komputer yang designnya disesuaikan dengan portabilitas laptop. Komponen yang ada pada laptop lebih kecil dibandingkan dengan komputer, dimana mengikuti ukuran dari laptop itu sendiri. Selain kecil, komponen laptop juga hemat energi dan efisien sesuai ukuran laptop. Layar yang digunakan pada laptop adalah LCD (Liquid Crystal Display) yang biasanya berukuran 10 inchi hingga 17 inchi bergantung dari ukuran laptop. Pada keyboard laptop dilengkapi dengan touchpad atau trackpad sebagaimana mouse pada komputer yang digunakan untuk menggerakkan kursor.

Laptop dirancang dengan bentuk flip untuk melindungi layar dan keyboard, namun semakin berkembangnya teknologi, sekarang ada beberapa laptop hanya berbentuk layar saja dan memiliki layar touchscreen. Laptop sangat cocok bagi seseorang yang mimiliki mobilitas yang tinggi dan selalu bekerja dengan laptop setiap saat. Ada juga beberapa orang memiliki laptop hanya untuk fashion atau gaya jaman sekarang, sehingga kadang sebagian orang tidak mengerti bagaimana cara merawat laptop yang mereka miliki. Namun, tak hanya itu, banyak pula yang membutuhkan tapi tak bisa merawat, sehingga laptop yang dimiliki gampang rusak. Untuk itu sebuah laptop perlu perawatan yang berkala agar kerja laptop tetap terjaga seperti baru.Seperti kata pepatah, “merawat lebih baik dari pada mengobati”, karena jika rusak tak sedikit biaya yang harus dikeluarkan. Untuk itu pada halaman selanjutnya juga akan dibahas bagaimana merawat laptop agar tetap terjaga fungsinya dan bisa bertahan lama.Sumber : (<http://id.shvoong.com>)

Cara merawat laptop agar tetap terjaga fungsinya dan bisa bertahan lama :

* + - 1. Jangan sembarangan mendownload software gratis dari internet.  
         Terlebih lagi misalnya software yang seolah-olah sebagai suatu antivirus. Gunakan software-software yang telah Anda dapatkan dari paket laptop yang Anda beli. Risiko virus bisa merusak ke dalam laptop Anda jika Anda sembarangan menggunakan software dari internet. Jika Anda tetap ingin menggunakan software hasil download, maka pastikan sudah Anda scan software tersebut dengan antivirus yang Anda miliki.
      2. Jangan memberikan Penutup apapapun termasuk sarung Laptop pada saat Laptop anda dalam keadaan menyala karena berdasarkan pengalaman Laptop orang lain Processor Laptopnya akan mati karena sirkulasi udara yang tidak ada dibawah di permukaan laptop.
      3. Menurut survei, kerusakan laptop yang paling umum terjadi pada harddisk dan LCD display/layar. Kerusakan harddisk diakibatkan benturan atau terjatuh. Kerusakan LCD biasanya karena terpapar sinar matahari dan tekanan fisik.
      4. Rapikanlah kabel-kabel adaptor atau kabel lain yang sedang terhubung dengan laptop, jangan sampai membuat orang lain tersandung.
      5. Selain kerusakan harddisk dan LCD, tumpahan cairan adalah penyebab kerusakan laptop yang paling umum, atau pakai selembar film-transparan yang disebut ‘keyboard protector’.
      6. Hindari Getaran dan Jauhkan laptop dari alat pengeras suara, misalnya loudspeaker, mesin/kendaraan berat, dan sumber getaran lainnya.
      7. Hindarkan laptop dari sinar x-ray di airport.
      8. Jaga kebersihan laptop, lap dengan kain bersih atau kapas agar bebas debu.
      9. Lindungi modem laptop. Gunakan modem yang mempunyai fitur digital-line guard. Karena fitur ini akan menjaga modem dari kerusakan jika secara tidak sengaja mencolokkan kabel modem ke jack telpon digital PABX atau jalur ISDN. Dan berhati-hatilah ketika akan mencolokkan kabel telepon ke laptop karena kawat logam dalam konektor modem laptop sangat tipis dan mudah bengkok.
      10. Listrik padam dan gangguan tegangan dapat terjadi sewaktu-waktu, di rumah, di kamar hotel atau di kantor. Jika memungkinkan, gunakanlah ’surge-protector’ jika sedang menggunakan AC outlet. Dan jangan lupa membuat back-up data secara rutin.
      11. Gunakanlah selalu tas laptop ketika sedang bepergian.

## Adapun ciri-ciri kerusakan Laptop yaitu yang terjadi di bagi dua secara hardware dan software.

## Ciri-ciri kerusakan laptop secara *hardware* :

## Kerusakan LCD Laptop / *Notebook*

## Layar tidak tampil gambar, menyala tapi keluar garis-garis vertikal, tampak blok hitam, dan gambar tidak simetris / acak. Solusinya coba cek dulu konektor atau pun soket - soket yang berhubungan dengan monitor.

## Ciri-ciri Kerusakan *Keyboard* pada Laptop/*Notebook* :

## Beberapa tuts tidak berfungsi, keluar bunyi beep panjang pada saat laptop dinyalakan, cursor berjalan tidak stabil / bergerak sendiri.

## Kerusakan yang lebih parah biasanya konslet dan ini menyebabkan Laptop atau Notebook setelah booting, melakukan restart-restart terus.

## Ciri-ciri Kerusakan Memory pada Laptop/*Notebook*

## Pada saat dihidupkan tidak tampak tampilan sama sekali, blue screen pada saat mulai loading Operating System. Bisa juga keluar suara beep berulang-ulang.

## Ciri kerusakan motherboard atau IC regulator Laptop/*Notebook*

## Dihidupkan agak sulit, baterai tidak mau di *charge*, Mati Total. Indikator *charger* nyala, setelah dicharge lampu indikator pada charger mati (konslet). Jadi terjadi arus balik pada powernya. Kerusakan ini sering terjadi.

## Ciri-ciri Kerusakan Charger pada Laptop/*Notebook*

## Baterai tidak mau di charge, tidak ada indikator masuk power, laptop di charge posisi hidup malah kemudian mati. Layar bergetar tidak stabil.

## Ciri-ciri Kerusakan DVD atau CD room Laptop/*Notebook*

## Tidak mau membaca CD, indikator CD off.

## Ciri-ciri Kerusakan Hardisk Laptop/*Notebook*

## Loading data / System lambat, berbunyi tidak normal, tidak bisa masuk windows, belum sampai login windows sudah restart sendiri.

## Ciri Kerusakan Chipset atau VGA Laptop/*Notebook*

## Layar tidak tampil, kalaupun tampil tidak mau akses ke Bios. Ada terdengar suara beep secara beraturan.

## Ciri-ciri kerusakan Baterai pada Laptop/*Notebook*

## Lampu charge baterai menyala tetapi kalau adaptor di cabut laptop ikut mati atau baterai tidak berfungsi normal karena kondisi ini biasanya karena safety circuit di board battery sudah rusak, bisa di akibatkan over current (i), over voltage (v) maupun heat atau panas. dan didalam board battery terdapat safety circuit 3 lapis dan cukup cerdas.

## Ciri-ciriKerusakan Laptop pada Software

## Tidak mau *booting* ke OS.

## Pada *Loading* awal desktop banyak informasi error.

## Proses *Loading* sangat Lambat.

## Tidak bisa membuka Aplikasi.

## Flasdisk Tidak bisa di baca.

## Folder Option pada *Explorer* Tidak bisa di buka.

## File regedit tidak bisa di eksekusi.

(file:///D:/judulpenelitian/datakerusakanlaptop/merk-laptop-terbaik-dalam-hal-tahan.html)

[](http://3.bp.blogspot.com/-lnlZNo0q_e8/Td907-iPotI/AAAAAAAAAWM/f0CMafY7oXM/s1600/Apple-MacBook-Pro-Winter-2011-From-The-Front.png)

**Gambar 2.17 Laptop atau notebook**

**2.6. Kerangka Pemikiran**

CV. Juve Komputer dan masyarakat umum

**Implementasi Sistem**

**Masalah**

1. Bagaimana cara merekayasa Sistem Pakar *Troubleshooting* kerusakan laptop atau notebook Berbasis Web?
2. Apakah sistem pakar yang direkayasa dapat digunakan untuk *Troubleshooting* kerusakan dan masalah pada laptop atau notebook ?
3. Tersedianya fasilitas dan teknologi yang mendukung.
4. Masyarakat membutuhkan software system pakar untuk troubleshooting laptop.

**Peluang**

**Desain Sistem**

1. Desain Model
2. Desain User Interface
   1. Desain Output
   2. Desain Input
3. Desain Teknologi
4. Desain Database

**Analisis Sistem**

* Diagram konteks
* Diagram arus data

**Solusi**

Aplikasi Sistem pakar troubleshooting laptop

**Tujuan**

1. Untuk mengetahui cara merekayasa sistem pakar untuk *troubleshooting* kerusakan pada laptop atau *notebook* berbasis web.
2. Sistem pakar *troubleshooting*  laptop atau *notebook* berbasis *website* yang sudah direkayasa dapat digunakan pada Juve komputer dan masyarakat umum penguna laptop.

**Pembangunan Sistem**

1. PHP
2. MySql
3. Dreamweaver
4. Photoshop

**Pengujian Sistem**

1. White Box
2. Black Box

**Gambar 2.18 Bagan Kerangka Pikir**

**BAB III**

**OBJEK DAN METODE PENELITIAN**

* 1. **Objek Penelitian**

Berdasarkan latar belakang dan kerangka pemikiran yang telah diuraikan pada Bab I dan Bab II, maka yang menjadi obyek penelitian adalah Sistem pakar *troubleshooting* laptop berbasis website.

* 1. **Metode Penelitian**

Metode yang akan digunakan dalam penelitian ini adalah pakar yaitu penelitian yang berusaha menuturkan pemecahan masalah yang ada dari seorang pakar, sehingga bisa memudahkan seseorang dapat memperbaikii atau menganalisa sendiri kerusakan apa yang ada pada laptop atau notebooknya. Tahap penelitiannya di uraikan sebagai berikut :

* + 1. **Tahap analisis**

Tahap analisis merupakan tahap penguraian dari suatu sistem informasi dengan maksud untuk mengidentifikasikan dan mengevaluasikan permasalahan-permasalahan, hambatan-hambatan yang terjadi dalam kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Didalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem sebagai berikut :

71

1. *Identify*, yaitu mengidentifikasi masalah.

2. *Understand*, yaitu memahami kerja dari sistem yang ada.

3. *Analyze*, yaitu menganalisa sistem.

4. *Report*, yaitu membuat laporan hasil analisis.

Alat-alat yang digunakan dalam analisis terstruktur adalah :

1. Diagram Konteks (Context Diagram)

2. Diagram arus data (Data Flow Diagram).

* + 1. **Tahap desain**

1. Desain Model

Merupakan tahapan yang lebih berfokus pada spesifikasi detail berbasis komputer. Sistem yang digunakan adalah *Model-Driven Design*, yaitu sebuah pendekatan desain sistem yang menekankan penggambaran model sistem untuk mendokumentasikan aspek teknis dan implementasi dari sebuah sistem. Dimana pada tahap ini kita melakukan pertimbangan-pertimbangan mengenai bagaiman suatu sistem akan diterapkan, baik dalam teknologi dan lingkungan implementasi. Pada tahap ini digunakan *Data Flow Diagram* (DFD), dimana kita memodelkan persyaratan bisnis logis dari suatu sistem informasi.DFD memodelkan keputusan-keputusan teknis dan desain manusia untuk diimplementasikan sebagai bagian dari suatu sistem informasi.

1. Desain *Output*

Desain *output* dimaksudkan untuk mengetahui bagaimana dan seperti apa bentuk *output-output* dari sistem yang akan dibuat. Desain *output* terinci terbagi atas dua, yaitu desain *output* berbentuk laporan di media kertas dan desain *output* dalam bentuk dialog di layar terminal (*monitor*).

1. Desain *Input*

Masukan merupakan awal dimulainya proses pengolahan informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan oleh organisasi.Data hasil dari transaksi tidak lepas dari data yang dimasukkan. Desain *input* terinci dimulai dari desain dokumen dasar sebagai penangkap *input* yang pertamakali. Jika dokumen dasar tidak didesain dengan baik, kemungkinan *input* yang tercatat dapat salah bahkan kurang.

1. Desain *Database*

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya.Penerapan *database* dalam aplikasi disebut *databasesistem*.

1. Desain Teknologi

Pada tahap ini kita menentukan teknologi yang akan dipergunakan dalam menerima input, menjalankan model, menyimpan dan mengakses data menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan.

* + 1. **Tahap Produksi/Pembuatan**

Merupakan tahapan dimana kita melakukan pengembangan, melakukan tahap produksi sesuai dari hasil analisa dan desain sistem yang sebelumnya. Termasuk didalamnya membangun sebuah aplikasi, menulis *listing* program dan membangunya dalam bentuk sebuah antarmuka dan integrasi sistem-sistem program yang terdiri dari *input*, *proses*, dan *output*, yang tersusun dalam sebuah sistem *menu* sehingga dapat dijalankan oleh pengguna sistem. Dalam tahapan ini, penulis menggunaakan bahasa pemrograman PHP dan *Database* Mysql.

* + 1. **Tahap pengujian**

Tahap ini dilakukan setelah semua modul selesai dibuat, dan program dapat berjalan, dimana seluruh perangkat lunak, program tambahan, dan semua program yang terlibat dalam pembangunan sistem diuji untuk memastikan sistem dapat berjalan sesuai dengan rancangan atau belum. Pengujian yang dilakukan dengan dua teknik pengujian, yaitu;

1. *White Box*

Dalam pengujian *WhiteBox* ini dengan membuat bagan alir program, *listing* program, grafik alir, pengujian *basispath* serta perhitungan *Ciclomatic Complexity.*

1. *Black Box*

Pengujian *Black Box* yang termasuk dalam tahap ini yaitu menguji antarmuka sistem, apakah sebuah sistem setelah diberikan ke pengguna dapat dioperasikan atau tidak.

* + 1. **Tahap implementasi**

Tahap implementasi sistem *(System Implementation)* merupakan tahap meletakan sistem supaya siap untuk dioperasikan pada CV. JUVE Komputer dan masyarakat umum penguna laptop.