

BAB II

KAJIAN TEORI

A. Penyakit tulang, sendi, dan otot (TSO)

Ada berbagai macam penyakit tulang, sendi, dan otot (TSO) pada manusia beberapa penyakit tersebut antara lain: fraktur tertutup, fraktur terbuka, *osteoporosis*, *osteoarthritis*, *osteomalacia*, *osteomielitis*, *polimialgia reumatik*, *artritis gout*, *artritis reumatoid*, dislokasi sendi, *tetanus*, dan *polio*

1. Fraktur Tertutup

Fraktur adalah terputusnya kontinuitas tulang, tulang rawan sendi, tulang rawan epifisis baik yang bersifat total maupun parsial. Fraktur tertutup adalah suatu fraktur yang tidak mempunyai hubungan dengan dunia luar sehingga pada fraktur tertutup tidak terdapat luka luar. Fraktur tertutup biasanya terjadi pada pasien yang memiliki riwayat trauma seperti terjatuh atau pernah mengalami kecelakaan. Biasanya gejala yang dikeluhkan pasien adalah nyeri pada tulang dan sulit digerakkan serta terjadi pembengkakan (Kementerian Kesehatan, 2014).

2. Fraktur Terbuka

Fraktur terbuka adalah suatu fraktur dimana terjadi hubungan dengan lingkungan luar melalui kulit sehingga ada kemungkinan terjadi kontaminasi bakteri yang dapat menimbulkan komplikasi berupa infeksi. Pada fraktur terbuka biasanya juga ikut terjadi pendarahan, tulang yang patah juga ikut terlihat menonjol keluar dari permukaan kulit, namun tidak semua fraktur terbuka membuat tulang terlihat menonjol keluar (Faswita Wirda, 2016).

3. *Osteoporosis*

Osteoporosis adalah kelainan dimana terdapat reduksi atau penurunan massa total tulang. Kecepatan resorbsi tulang lebih cepat dari pembentukan tulang. Tulang menjadi keropos seara progresif, rapuh, mudah patah. Biasanya *Osteoporosis* terjadi pada orang yang berusia diatas 35 tahun dan resiko wanita terserang osteoporosis lebih tinggi daripada pria (Kanis John A, 1994). Faktor resiko terserang *Osteoporosis* antara lain adalah wanita menopause, gaya hidup yang tidak baik seperti merokok, konsumsi kafein, konsumsi alkohol, dan kurang aktivitas fisik (Guide W., 2017). Gejala pada *osteoporosis* biasanya antara lain: sakit punggung yang berkelanjutan dalam jangka panjang, dan postur tubuh menjadi bungkuk (Faqih Ruhyanudin, 2011)...

4. *Osteoarthritis*

Penyakit sendi degeneratif yang berkaitan dengan kerusakan kartilago sendi Pasien sering datang berobat pada saat sudah ada deformitas sendi yang bersifat permanen. Secara simtomatis penyakit sendi degeneratif terjadi pada usia 50-70, diantara yang menderita termuda ialah pada usia 20 tahun (Kementrian Kesehatan, 2014). Faktor utama yang dihubungkan dengan kejadian OA adalah penuaan, trauma sebelumnya, kecenderungan genetik, dan obesitas (Lukman Zulkifli Amin, 2015). Ada dua jenis osteoarthritis, yang primer penyebab belum diketahui, yang sekunder akibat trauma, infeksi, atau pernah terjadi fraktur. Gejala yang dikeluhkan pasien *osteoarthritis* biasanya nyeri pada persendian yang bergerak, terutama sendi penerima beban (panggul-lutut), dan persendian tangan tetapi tidak menutup kemungkinan bahwa sendi-sendi yang lain juga dapat terserang. Gejala

osteoarthritis umumnya berkembang secara perlahan-lahan dan semakin parah seiring waktu. Tingkat keparahan gejala penyakit ini dapat berbeda-beda pada tiap penderita serta lokasi sendi yang diserang (Kementerian Kesehatan, 2014).

5. *Osteomielitis*

Osteomielitis adalah suatu bentuk infeksi tulang yang menyebabkan kerusakan dan pembentukan tulang baru. Ada beberapa mekanisme infeksi yang dapat menyebabkan *osteomielitis* antara lain: infeksi (misalnya. setelah trauma, operasi, atau penyisipan sendi prostetik), insufisiensi vaskular (misal pada diabetes mellitus atau gangguan pembuluh darah perifer), dan penyebaran hematogen dari infeksi, misalnya diosteomielitis vertebral pada anak-anak (Gunawan, 2010). Penyebab utama *osteomielitis* adalah bakteri *Staphylococcus aureus*. Bakteri ini dapat menginfeksi tulang melalui aliran darah. Sebenarnya bakteri *Staphylococcus* adalah bakteri yang jarang menyebabkan masalah kesehatan dan umum terdapat di kulit, namun bakteri ini dapat berbalik menyerang ketika sistem kekebalan tubuh sedang lemah. Kebanyakan *osteomielitis* terjadi pada orang-orang yang berusia di atas 60 tahun. Biasanya pada wanita terjadi diatas usia 50 tahun atau menopause. Kegemukan/ obesitas juga memiliki resiko yang besar terserang *osteomielitis*. (Faqih Ruhyanudin, 2011).

6. Polimialgia Reumatik

Polymyalgia rheumatica (PMR) adalah suatu sindrom klinis dengan etiologi yang tidak diketahui yang mempengaruhi individu usia lanjut. *Polymyalgia* biasanya terjadi pada orang-orang lanjut usia dan lebih sering terjadi pada wanita.

Gejala-gejala yang dialami pasien biasanya nyeri dan kekakuan leher, bahu dan pinggul. Kekakuan pada pasien biasanya akan menyebabkan pasien mengalami kesulitan bangkit dari kursi, berbalik di tempat tidur, atau mengangkat tangan mereka di atas bahu tinggi. Kekakuan terjadi setelah periode istirahat (fenomena gel) serta kekakuan pada pagi hari lebih dari 1 jam biasanya terjadi. Pasien juga mungkin menggambarkan sendi distal bengkak, pembengkakan tungkai. *Carpal tunnel syndrome* dapat terjadi pada beberapa pasien. Kebanyakan pasien selalu lebih tua dari 50 tahun dan biasanya lebih tua dari 65 tahun (Carlos Alvarani M.D., 2002).

7. Artritis Gout

Gout atau *arthritis gout* adalah suatu kelainan metabolismik yang mana laki-laki delapan sampai sembilan kali lebih sering terkena daripada wanita. Penyakit ini dapat terjadi pada berbagai usia, usia yang sering terkena adalah sekitar 50 tahunan. 85% dari penderita gout mempunyai faktor genetik. Gout terjadi sebagai akibat dari *hyperuricemia* yang berlangsung lama (asam urat serum meningkat) disebabkan oleh karena penumpukan purin atau rekresi asam urat yang kurang dari ginjal. Faktor resiko penyakit *Gout* antara lain: konsumsi alkohol, daging merah dan makanan yang banyak mengandung purin (Barry L, 2014). Gejala dari *gout* antara lain: Susah bergerak atau menggerakkan anggota gerak tubuh tertentu (pada

bagian yang sakit), Rasa sakit pada sendi biasanya terjadi pada malam hari, Sendi terasa sakit secara tiba-tiba (terutama sendi jempol kaki) atau sendi ujung bagian tubuh (Faqih Ruhyanudin, 2011).

8. *Artritis Reumatoid*

Penyakit autoimun yang ditandai dengan terdapatnya sinovitis erosif simetrik terutama mengenai jaringan persendian, seringkali juga melibatkan organ tubuh lainnya. *Atritis Rhematoid* lebih banyak terjadi pada wanita (3:1 dengan kasus pria) pada usia 25 – 35 tahun. Faktor resiko *Atritis Rhematoid* terjadi pada orang orang yang berusia diatas 60 tahun. Biasanya pada wanita terjadi diatas usia 50 tahun atau menopause. Kegemukan/ obesitas juga memiliki resiko yang besar terserang *Atritis Rhematoid*. Selain itu pekerja berat dengan penggunaan satu sendi terus menerus juga memiliki resiko terkena penyakit ini. Faktor genetik juga memiliki resiko yang besar untuk terkena *Atritis Rhematoid*. Gejala *Atritis Rhematoid* antara lain: nyeri dan bengkak pada sendi yang berlangsung terus menerus, kaku pada pagi hari berlangsung selama lebih dari 30 menit, persendian mengalami bengkak dan hangat jika diraba (Lutfi Chabib, 2016).

9. Dislokasi Sendi

Dislokasi sendi terjadi ketika permukaan tulang sendi tidak sesuai dengan posisi anatomi. Dislokasi merupakan keadaan emergensi karena berhubungan dengan kerusakan aliran darah dan persarafan disekitarnya. Diskolasi umumnya terjadi pada jari dan bahu. Meski demikian, persendian lain seperti lutut, pinggul, siku tangan, maupun pergelangan kaki juga dapat mengalami cedera ini. Gejala utama dislokasi biasanya akan terlihat melalui kejanggalan yang muncul pada

bentuk sendi, misalnya muncul benjolan aneh di dekat tempurung atau soket sendi. Sendi tersebut juga akan mengalami pembengkakan, lebam, terasa sangat sakit, serta tidak dapat digerakkan (Kementerian Kesehatan, 2014).

10. *Tetanus*

Tetanus merupakan infeksi yang tergolong serius dan disebabkan oleh bakteri *Clostridium tetani*. Bakteri ini umumnya terdapat dalam debu, tanah, serta kotoran hewan dan manusia. Bakteri *tetanus* sering kali masuk ke tubuh melalui luka terbuka akibat cidera atau luka bakar. Saat berhasil memasuki tubuh bakteri *tetanus* akan berkembang biak dan melepaskan *neurotoksin*. *Neurotoksin* adalah racun yang menyerang sistem saraf. Racun tersebut dapat mengacaukan kinerja saraf dan dapat menyebabkan kejang dan kekakuan otot yang merupakan gejala utama *tetanus* (Kementerian Kesehatan, 2014).

11. *Polio*

Polio atau *poliomyelitis* adalah penyakit virus yang sangat mudah menular dan menyerang sistem saraf. Pada kondisi penyakit yang bertambah parah, dapat menyebabkan kesulitan bernapas, kelumpuhan, dan pada sebagian kasus menyebabkan kematian. Penyakit polio disebabkan oleh virus yang umumnya masuk melalui makanan atau minuman yang terkontaminasi dengan tinja dan virus polio. Sama halnya seperti cacar, *polio* hanya menjangkiti manusia. Dalam tubuh manusia, virus *polio* menjangkiti tenggorokan dan usus. Selain melalui kotoran, virus *polio* juga dapat menyebar melalui tetesan cairan yang keluar saat penderitanya batuk atau bersin. Penderita *polio* biasanya mengalami gejala seperti

lemah otot, demam, merasa keletihan, sakit pada tenggorokan, serta terasa kaku dan sakit pada bagian kaki, tangan, leher, dan punggung (Kementerian Kesehatan, 2014).

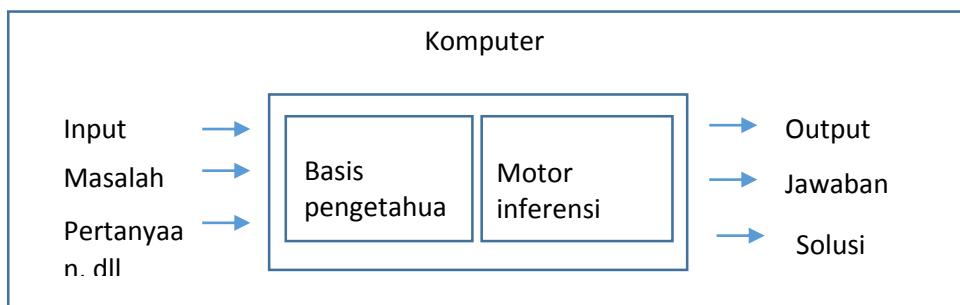
B. Kecerdasan Buatan

Kecerdassan buatan atau *artificial intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Sementara itu menurut (*Avrin Barr* dan *Edward A. Feigenbaum*, 1981) kecerdasan buatan adalah bagian dari ilmu komputer yang mempelajari perancangan sistem komputer yang cerdas, yaitu sistem yang memiliki karakteristik berpikir seperti manusia.

Manusia dapat menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar, namun bekal pengetahuan saja tidak cukup bagi manusia untuk menyelesaikan permasalahan, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan yang dimiliki. Sama halnya seperti manusia untuk membuat mesin cerdas dan dapat bekerja layaknya manusia mesin tersebut juga harus diberi pengetahuan dan kemampuan untuk menalar. Menurut Sri Kusumadewi (2003: 3) dari sudut pandang pemrograman untuk membuat aplikasi kecerdasan buatan dibutuhkan 2 bagian utama yaitu :

1. Basis pengetahuan (*knowledge base*), berisi fakta fakta, teori, pemikiran, dan hubungan antara satu dengan lainnya.

2. Motor inferensi (*inference engine*), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.



Gambar 2.1. Penerapan konsep kecerdasan buatan di komputer

Kecerdasan buatan mempunyai beberapa lingkup utama, menurut (Sri Kusumadewi, 2003) lingkup utama dari kecerdasan buatan adalah :

1. Sistem pakar (*Expert System*), disini komputer digunakan sebagai saraa untuk menyimpan pengetahuan para pakar dengan demikian komputer akan memiliki keahlian untuk menyelesaikan suatu permasalahan dengan meniru keahlian yang dimiliki oleh pakar.
2. Pengolahan bahasa alami (*Natural Languege Processing*), dengan pengolahan bahasa alami ini diharapkan *user* dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari hari.
3. Pengenalan ucapan (*Speech recognition*), melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan komputer dengan suara.
4. Robotika dan sistem sensor (*Robotic and Sensory System*).
5. *Computer Vision*, mencoba untuk mengintrepetasikan gambar atau obyek obyek tampak melalui komputer.

6. *Intelligent Computer-aided Instruction*, komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.

Jika dibandingkan dengan kecerdasan alami atau kecerdasan yang dimiliki oleh manusia kecerdasan buatan memiliki beberapa kelebihan antara lain :

1. Kecerdasan buatan lebih bersifat permanen
2. Kecerdasan buatan lebih mudah diduplikasi dan disebarluaskan
3. Kecerdasan buatan lebih murah dibanding kecerdasan alami
4. Kecerdasan buatan bersifat konsisten
5. Kecerdasan buatan dapat didokumentasi
6. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibanding kecerdasan alam

C. Sistem Pakar

Secara umum sistem pakar adalah suatu sistem yang berusaha mengadopsi pengetahuan manusia ke dalam komputer, agar komputer dapat menyelesaikan suatu masalah seperti yang biasa dilakukan para ahli. Menurut Rika Rosnelly (2011: 2) sistem pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek (*emulates*) kemampuan pengambilan keputusan (*desicion making*) seorang pakar. Pendapat lain tentang sistem pakar (Sri Kusumadewi, 2003) :

1. Menurut Durkin: Sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan seorang pakar

2. Menurut Ignizio: Sistem pakar adalah suatu model dan prosedur yang berkaitan dalam suatu domain tertentu yang mana tingkat keahliannya dapat dibandingkan dengan keahlian seorang pakar
3. Menurut Giarratano dan Riley: Sistem pakar adalah suatu sistem komputer yang dapat menyamai atau meniru kemampuan seorang pakar.

Ada beberapa ciri dan karakteristik yang membedakan sistem pakar dengan sistem yang lain. Ciri dan karakteristik sistem pakar antara lain :

1. Pengetahuan sistem pakar merupakan suatu konsep, bukan berbentuk numeris. Hal ini dikarenakan komputer melakukan proses pengolahan data secara numerik sedangkan keahlian dari seorang pakar adalah fakta dan aturan aturan bukan numerik.
2. Informasi dalam sistem pakar tidak selalu lengkap, subjektif, tidak konsisten subjek terus berubah dan tergantung pada kondisi lingkungan sehingga keputusan yang diambil bersifat tidak pasti dan tidak mutlak akan tetapi menurut ukuran kebenaran tertentu. Oleh karena itu dibutuhkan kemampuan sistem untuk belajar secara mandiri dalam menyelesaikan masalah masalah dengan pertimbangan khusus.
3. Kemungkinan solusi sistem pakar terhadap suatu permasalahan adalah bervariasi dan mempunyai banyak pilihan jawaban yang dapat diterima, semua faktor yang ditelusuri memiliki ruang masalah yang luas dan tidak pasti, oleh karena itu diperlukan fleksibilitas sistem dalam menangani kemungkinan solusi dari berbagai permasalahan.

4. Perubahan atau pengembangan pengetahuan dalam sistem pakar dapat terjadi setiap saat bahkan sepanjang waktu sehingga diperlukan kemudahan dalam modifikasi sistem untuk menampung jumlah pengetahuan yang semakin besar dan semakin bervariasi.
5. Pandangan dan pendapat setiap pakar tidaklah selalu sama, oleh karena itu tidak ada jaminan bahwa solusi sistem pakar merupakan jawaban yang pasti benar. Setiap pakar akan memberikan pertimbangan pertimbangan pada faktor subjektif.
6. Keputusan merupakan bagian terpenting dari sistem pakar. Sistem pakar harus memberikan solusi yang akurasi berdasarkan masukan pengetahuan meskipun solusinya sulit sehingga fasilitas informasi sistem harus selalu diperlukab.

Ada beberapa keuntungan dalam penggunaan sistem, menurut Sri kusumadewi (2003: 110) manfaat yang dapat diambil dari sistem pakar antara lain:

1. Memungkinkan orang awam dapat mengerjakan pekerjaan para ahli.
2. Dapat melakukan proses berulang secara otomatis.
3. Menyimpan pengetahuan dan keahlian para pakar.
4. Meningkatkan output dan produktivitas.
5. Meningkatkan kualitas.
6. Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka).
7. Mampu beroperasi dalam lingkungan yang berbahaya.
8. Memiliki kemampuan untuk mengakses pengetahuan.
9. Memiliki reliabilitas.

10. Meningkatkan kapabilitas sistem komputer.
11. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian.
12. Sebagai media pelengkap dalam pelatihan.
13. Meningkatkan kapabilitas dalam penyelesaian masalah.
14. Menghemat waktu dalam pengambilan keputusan.

Selain memiliki keuntungan sistem pakar juga memiliki beberapa kelemahan antara lain :

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
2. Sulit dikembangkan, hal ini terkait dengan ketersediaan pakar di bidangnya
3. Sistem pakar tidak 100% benar.

Menurut Sri kusumadewi (2003: 113) ada 4 bentuk sistem pakar yaitu :

1. Berdiri sendiri, sistem pakar jenis ini merupakan *software* yang berdiri sendiri tidak bergabung dengan *software* lainnya.
2. Tergabung, sistem pakar jenis ini merupakan bagian program yang terkandung di dalam suatu algoritma (konvensional), atau merupakan program dimana didalamnya memanggil algoritma subrutin lain (konvensional)
3. Menghubungkan ke *software* lain, bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu pakar program tertentu, misalnya dengan DBMS
4. Sistem mengabdi, sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang membantu menganalisis data radar.

1. Representasi pengetahuan

Menurut Kusrini (2008: 6) representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sistem pakar. Reprsntasi dimaksudkan untuk menangkap sifat sifat penting masalah dan membuat informasi itu dapat diakses oleh prosedur pemecahan masalah. Ada beberapa karakteristik dari metode representasi pengetahuan yaitu

- a. Harus dapat diprogram dalam bahasa pemrograman dan hasilnya disimpan dalam memori.
- b. Dirancang sedemikian sehingga isinya dapat digunakan untuk proses penalaran.
- c. Model representasi pengetahuan merupakan struktur data yang dapat dimanipulasi oleh mesin inferensi dan pencarian untuk aktivitas pencocokan.

Beberapa teknik yang sering digunakan untuk merepresentasikan pengetahuan antara lain adalah :

- a. Jaringan semantik (*Semantic network*).

Dalam jaringan semantik pengetahuan diorganisasikan dengan menggunakan jaringan yang disusun oleh dua komponen dasar, yaitu *node* dan *arc*. *Node* menyatakan objek, konsep atau situasi yang ditunjukkan oleh kotak atau lingkaran, sedangkan *arc* menyatakan hubungan antar node yang ditunjukkan oleh tanda panah yang menghubungkan *node-node* dalam jaringan.

Menurut Kusumadewi dalam (Rika Rosnelly, 2011), jaringan semantik merupakan gambaran pengetahuan secara grafis, yang menunjukkan hubungan antara berbagai objek. Jaringan semantik terdiri dari lingkaran lingkaran yang

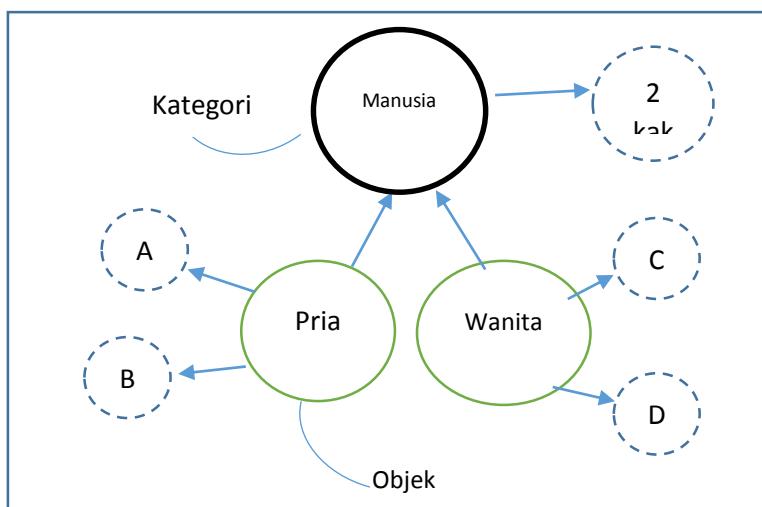
menunjukkan objek objek dan informasi tentang objek objek tersebut. Objek disini dapat berupa benda atau peristiwa, antara dua objek dihubungkan dengan *arc* yang menunjukkan hubungan antar objek.

1) Kategori

Objek objek yang mempunyai kemiripan karakteristik dapat digolongkan dalam kategori tertentu. Kategori merupakan pengorganisasian objek yang merupakan representasi yang vital dalam jarigan semantik.

2) Objek

Objek merupakan individu tersendiri yang mempunyai sifat sifat karakteristik yang spesifik. Dalam kaitannya dengan kategori, jika sebuah objek diturunkan dari kategori maka objek akan memiliki sifat dari kategori secara keseluruhan dan juga mempunyai sifat spesifik dari objek itu sendiri. Gambar 2.2 menunjukkan conoh jaringan semantik

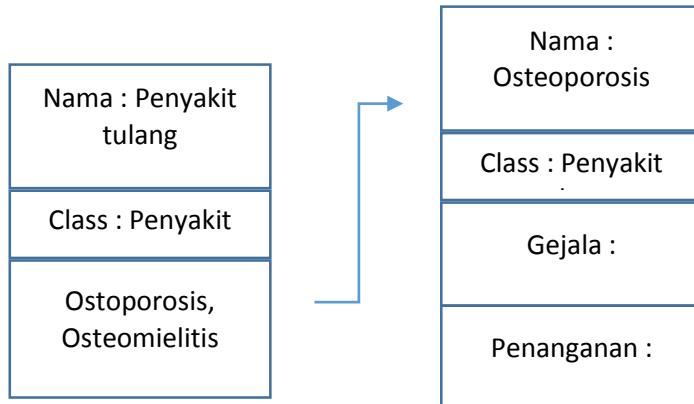


Gambar 2.2 Contoh jaringan

b. *Frame*

Menurut Minsky dalam (Rika Rosnelly, 2011), *frame* dapat dipandang sebagai struktur data statik yang digunakan untuk merepresentasikan situasi situasi yang telah dipahami dan *stereotype*. Sebuah *frame* digambarkan dengan menggunakan jaringan dari *node node* dan hubungan hubungan. Level teratas dari *frame* menyatakan atribut atribut sedangkan level terendah memiliki terminal dan slot yang harus diisi oleh data. *Frame* merupakan kumpulan slot slot yang merupakan atribut untuk mendeskripsikan pengetahuan. Pengetahuan yang termuat dalam slot dapat berupa kejadian, lokasi, situasi ataupun elemen elemen lain.. *Frame* biasanya digunakan untuk merepresentasikan pengetahuan yang didasarkan pada karakteristik yang sudah dikenal, yang merupakan hasil dari pengalaman.

Pada gambar 2.3 adalah contoh dari *frame*.



Gambar 2.3. Contoh *frame*

c. *Script*

Script merupakan skema representasi pengetahuan yang hampir sama dengan *frame*, hanya saja *frame* menggambarkan objek sedangkan *script* menggambarkan urutan peristiwa. Penggambaran urutan peristiwa pada *script*

menggunakan serangkaian slot yang berisi tentang informasi orang, objek, dan tindakan yang terjadi dalam suatu peristiwa. *Script* memiliki beberapa elemen yaitu:

- 1) Kondisi input : kondisi yang harus dipenuhi sebelum terjadinya suatu peristiwa
- 2) Jalur (*track*): variasi kejadian yang mungkin dalam suatu *script*.
- 3) Pendukung (*prop*): objek objek pendukung yang digunakan pada suatu peristiwa.
- 4) Peran (*roles*)
- 5) Adegan (*scene*): adegan yang dimainkan.
- 6) Hasil (*goal*): hasil atau kondisi yang mungkin terjadi setelah urutan peristiwa terjadi.

Contoh sederhana dari *script* sebagai berikut:

Script rumah makan

- Jalur (*track*) : rumah makan
Peran (*roles*) : Pelanggan
Pendukung (*prop*) : Pelayan, meja, makanan, kasir, tempat duduk
Kondisi masukan : pelanggan lapar, pelanggan punya uang.

Adegan (*scene*) 1 : Masuk

- Pelanggan masuk restoran
- Pelanggan antri
- Pelanggan memilih tempat duduk

Adegan (*scene*) 2 : Pesan makanan

- Pelanggan melihat menu yang ada di meja
- Pelanggan memilih makanan yang ada di menu

- Pelanggan menulis makanan yang dipesan.
- Pelanggan memberikan pesanan kepada pelayan.

Adegan (*scene*) 3 : Makan

- Pelayan mengantar pesanan ke meja pelanggan
- Pelanggan menerima makanan yang dipesan.
- Pelanggan memakan makanan yang dipesan.

Adegan (*scene*) 4 : Pulang

- Pelanggan membayar makanan yang sudah dimakan di kasir
- Pelanggan meninggalkan rumah makan

Hasil :

- Pelanggan merasa kenyang
- Uang pelanggan berkurang
- Pelanggan merasa kecewa

d. Kaidah produksi

Di sini pengetahuan disajikan dalam aturan-aturan yang berbentuk pasangan keadaan-aksi (condition-action). Sistem Pakar yang basis pengetahuannya disajikan dalam bentuk aturan produk disebut sistem berbasis-aturan (*rule-based system*).

Contoh dari kaidah produksi:

- Jika suhu berada di bawah 20°C maka udara terasa dingin
- Jika terluka maka berikan obat luka

Aturan dalam kaidah produksi terkadang juga menggunakan operator logika *and* atau *or*, contohnya

- Jika kepala pusing dan pilek maka terserang influenza
- Jika tidak memiliki uang atau mobil tidak tersedia maka tidak dapat membeli mobil

Aturan dalam kaidah produksi diklasifikasikan menjadi kaidah derajat pertama dan kaidah meta. Kaidah meta adalah aturan yang bagian konklusinya merupakan premis bagi kaidah yang lain, sebaliknya kaidah derajat pertama adalah aturan yang bagian konklusinya bukan premis bagi kaidah yang lain.

Contoh kaidah meta:

- jika pusing dan cepet lelah dan sering kesemutan maka mengalami anemia

Contoh kaidah derajat pertama :

- Jika anemia dan batuk kronis maka terserang TBC

2. Komponen sistem pakar

Menurut Rika Rosnelly (2011: 14) komponen yang terdapat dalam struktur sistem pakar adalah *knowledge base* (basis pengetahuan), *inference engine*, *working memory*, *explanation facility*, *knowledge acquisition facility*, dan *user interface*.

a. Basis pengetahuan

Basis pengetahuan berisi pengetahuan-pengetahuan untuk penyelesaian suatu masalah. Ada dua bentuk pendekatan basis pengetahuan yang umum digunakan yaitu :

1) Penalaran berbasis aturan (*rules based reasoning*)

Pada penalaran berbasis aturan , pengetahuan direpresentasikan dengan menggunakan aturan berbentuk *if-then*. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu dan pakar dapat menyelesaikan masalah secara berurutan. Selain itu bentuk pendekatan ini juga

digunakan apabila dibutuhkan penjelasan tentang jejak atau langkah-langkah pencapaian solusi.

2) Penalaran berbasis kasus (*case based reasoning*)

Pada penalaran berbasis kasus, basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang. Bentuk ini digunakan apabila *user* menginginkan untuk mengetahui lebih banyak lagi pada kasus-kasus yang hampir sama. Selain itu bentuk pendekatan ini juga digunakan apabila kita telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan.

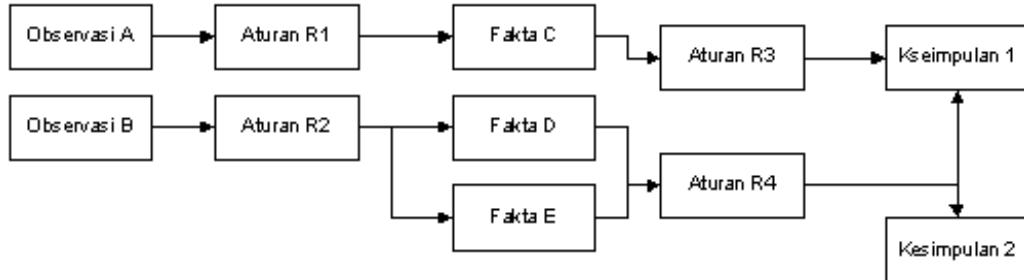
b. Mesin inferensi (*inference engine*)

Menurut Kusrini (2008: 8) inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar proses inferensi dilakukan dalam suatu modul yang disebut mesin inferensi (*inference engine*). Ada dua metode inferensi yang biasanya digunakan dalam sistem pakar yaitu runut maju (*forward chaining*) dan runut balik (*backward chaining*).

1) Runut maju (*forward chaining*)

Runut maju berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini untuk mencari kesimpulan diawali dari fakta-fakta yang diberikan kemudian dicari *rules* yang sesuai dengan fakta-fakta tersebut, kemudian dilakukan

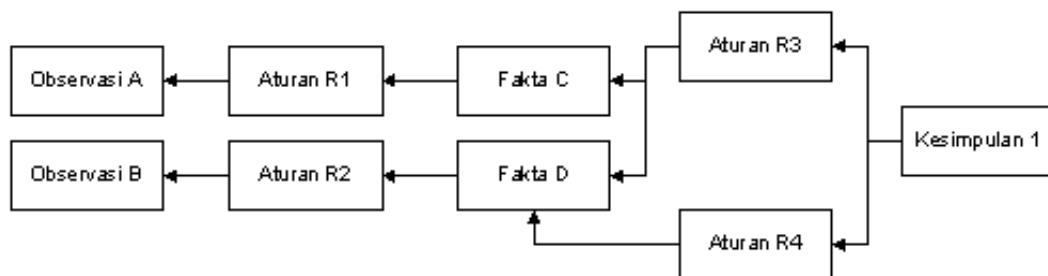
penarikan kesimpulan. Contoh sederhana gambaran penyelesaian masalah dengan forward chaining seperti pada gambar 2.4.



Gambar 2.4 Contoh penyelesaian masalah dengan *forward chaining*

2) Runut balik (*backward chaining*)

Runut balik merupakan metode penalaran kebalikan dari runut maju. Dalam metode runut balik penalaran dimulai dengan tujuan kemudian merunut balik ke jalur yang akan mengarahkan ke tujuan tersebut. Dengan kata lain penalaran dimulai dari hipotesis terlebih dahulu dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan. Contoh sederhana gambaran penyelesaian masalah dengan *backward chaining* sebagai berikut:



Gambar 2.5 Contoh penyelesaian masalah dengan backward chaining

c. *Working memory*

Working memory berguna untuk menyimpan fakta yang dihasilkan oleh *inference engine* dengan penambahan berupa derajat kepercayaan atau dapat juga dipakai sebagai global *database* dari fakta yang digunakan oleh rule rule yang ada.

d. *Explanation facility*

Kemampuan untuk menjelaskan (*tracing*) bagaimana suatu kesimpulan dapat diambil merupakan hal yang sangat penting untuk transfer pengetahuan dan pemecahan masalah. Fasilitas penjelasan merupakan komponen tambahan yang akan meningkatkan kemampuan sistem pakar.

e. *Knowledge acquisition facility*

Melibuti proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi ke program komputer yang bertujuan untuk memperbaiki atau mengembangkan basis pengetahuan

f. *User interface* (antarmuka pengguna)

User interface merupakan mekanisme untuk memberi kesempatan kepada *user* dan sistem pakar untuk berkomunikasi. Antar muka menerima informasi dari pengguna dan mengubahnya menjadi bentuk yang dapat diterima oleh sistem. Selain itu antar muka juga menerima informasi dari sistem pakar dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pengguna. Syarat utama membangun *user interface* adalah kemudahan dalam menjalankan sistem.

Semua kesulitan dalam membangun suatu program harus disembunyikan, yang ditampilkan hanyalah tampilan yang interaktif, komunikatif, dan kemudahan pakai

3. *Certainty factor (faktor Kepastian)*

Faktor kepastian (*certainty factor*) diperkenalkan oleh Shortliffe Buchanan dalam pembuatan MYCIN, *certainty factor* (CF) merupakan nilai parameter klinis yang diberikan MYCIN untuk menunjukkan besarnya kepercayaan (Kusrini, 2008). Tim pengembang MYCIN mencatat bahwa dokter sering kali menganalisa informasi yang ada dengan ungkapan seperti misalnya: mungkin, kemungkinan besar, hampir pasti, dan sebagainya. Untuk mengakomodasi hal ini tim MYCIN menggunakan *certainty factor* (CF) guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi. Nilai CF(*Rule*) serta bobot dari masing-masing fakta didapat dari interpretasi istilah dari pakar menjadi nilai CF serta bobot tertentu.

Certainty factor didefinisikan sebagai berikut:

$$CF(H, e) = MB(H, e) - MD(H, e)$$

$CF(H, e)$: Certainty factor dari hipotesis H yang dipengaruhi oleh gejala (evidence) E. Besarnya Cf berkisar antara -1 sampai dengan 1. Nilai -1 menunjukkan ketidakpercayaan mutlak, sedangkan nilai 1 menunjukkan kepercayaan mutlak

$MB(H, E)$: Ukuran kenaikan kepercayaan (measure of increased belief) terhadap hipotesis H yang dipengaruhi oleh gejala E

$MD(H, E)$: Ukuran kenaikan ketidakpercayaan (measure of increased disbelief) terhadap hipotesis H yang dipengaruhi oleh gejala E.

4. Kategori sistem pakar

Menurut Sri Kusumadewi (2003: 122) ada beberapa permasalahan yang dicakup dalam sistem pakar antara lain :

- a. Interpretasi, pengambilan keputusan dari hasil observasi, termasuk diantaranya: pengawasan, pengenalan ucapan, analisis citra, interpretasi sinyal, dan beberapa analisis kecerdasan.
- b. Prediksi, termasuk diantaranya : peramalan, prediksi demografis, peramalan ekonomi, prediksi lalu lintas, estimasi hasil, militer, pemasaran atau peramalan keuangan
- c. Diagnosis, termasuk diantaranya: medis, elektronis, mekanis, dan diagnosis perangkat lunak.
- d. Perancangan, termasuk diantaranya: layout sirkuit dan perancangan bangunan
- e. Perencanaan, termasuk diantaranya: perencanaan keuangan, komunikasi, militer, pengembangan produk, *routing*, dan manajemen proyek
- f. Monitoring, misalnya *computer-Aided monitoring system*.
- g. Debugging, memberikan resep obat terhadap suatu kegagalan
- h. Instruksi, melakukan instruksi untuk diagnosis, *debugging*, dan perbaikan kinerja.
- i. Kontrol, melakukan kontrol terhadap interpretasi, prediksi, perbaikan, dan monitoring kelakuan sistem.

D. Basis data (*database*)

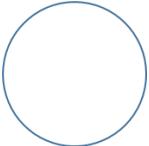
Basis data adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema

atau struktur tertentu, dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu. Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip dan tujuan utamanya adalah kemudahan serta kecepatan dalam pengambilan kembali data/arsip. Perbedaanya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronik.

Dalam pengelolaan basis data diperlukan suatu sistem pengelola basis data atau *database management system (DBMS)*. DBMS merupakan perantara bagi pemakai dengan basis data dalam disk. Perangkat DBMS akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali serta menerapkan mekanisme pengamanan data, pemakaian data bersama, dan pemaksaan keakuratan/konsistensi data.

E. DFD (*Data Flow Diagram*)

Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi atau simbol simbol untuk menggambarkan sistem jaringan kerja antar fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data. Penggunaan DFD sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas. Ada beberapa komponen dalam DFD, menurut Demarco dan Yordan komponen dalam DFD adalah sebagai berikut :

Simbol	Kategori
	Entitas luar (External entity)
	Proses (Process)
	Aliran data (Data flow)
	Penyimpanan (Data store)

Tabel 2.1 Keterangan gambar dalam DFD

Entitas luar (*external entity*) merupakan entitas yang berada di luar lingkungan sistem namun berinteraksi dengan sistem. Entitas luar akan memberikan input atau menerima output dari sistem. Entitas ini dapat berupa orang, benda atau sistem lain.

Proses (*process*) merupakan kegiatan atau pekerjaan yang dilakukan oleh orang, mesin, atau komputer, dimana aliran data masuk kemudian ditransformasikan dan menghasilkan aliran data keluar

Aliran data (*Data Flow*) disimbolkan dengan anak panah, yang menggambarkan aliran data yang masuk maupun keluar dari suatu proses. Aliran data mengalir diantara proses, simpanan data, dan entitas luar..

Penyimpanan (*Data store*) merupakan tempat penyimpanan data atau suatu sistem database dari suatu komputer yang dapat berupa suatu arsip/dokumen, suatu agenda/buku ataupun suatu tabel.

F. HTML dan PHP

HTML atau *HyperText Markup Language* merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman web (M. Rudyanto Arief, 2011: 23).

HTML dikembangkan oleh W3C (World Wide Web Consortium) semenjak awal teknologi internet. HTML terus dikembangkan agar dapat menampilkan lebih banyak konten selain teks dan gambar, misalnya suara, video dan lain sebagainya

PHP (PHP: *Hypertext Preprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis (M. Rudyanto Arief, 2011). Karena PHP merupakan *server-side scripting* maka perintah perintah PHP akan dieksekusi di server kemudian hasilnya dikirimkan ke browser dalam bentuk HTML, dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan web lebih tejamin. PHP dirancang untuk membentuk halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web.

G. MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengelolaan datanya. MySQL menggunakan SQL sebagai

bahasa dasar untuk mengakses database sehingga mudah untuk digunakan. SQL dibagi menjadi tiga bentuk query, yaitu (Haris Saputro, 2012) :

1. DDL (*Data Definition Language*)

DDL adalah sebuah metode Query SQL yang berguna untuk mendefinisikan data pada sebuah Database, Query yang dimiliki DDL adalah :

- a. CREATE, digunakan untuk membuat *database* dan tabel
- b. DROP , digunakan untuk menghapus tabel dan *database*
- c. ALTER digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah *Field* (*Add*), mengganti nama *Field* (*Change*) ataupun menamakannya kembali (*Rename*), dan menghapus *Field* (*Drop*).

2. DCL (*Data Control Language*)

DCL adalah sebuah metode Query SQL yang digunakan untuk memberikan hak otorisasi mengakses *database*, mengalokasikan space, pendefinisian space, dan pengauditan penggunaan database. Query yang dimiliki DCL adalah :

- a. GRANT, untuk memberikan hak akses *user* tertentu mengakses tabel dalam *database*.
 - b. REVOKE, untuk membatalkan hak akses *user*, yang ditetapkan oleh perintah GRANT
 - c. COMMIT, menetapkan penyimpanan *database*
 - d. ROLLBACK , membatalkan penyimpanan *database*
3. DML (Data Manipulation Language)

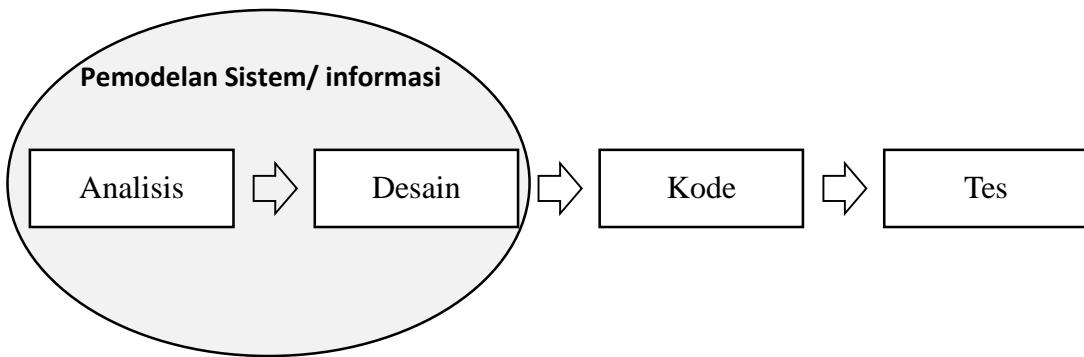
DML adalah sebuah metode Query yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari Query DML ini untuk melakukan pemanipulasi database yang telah dibuat. Query yang dimiliki DML adalah :

- a. INSERT , digunakan untuk memasukkan data pada tabel *database*
- b. UPDATE , digunakan untuk pengubahan terhadap data yang ada pada tabel *database*
- c. DELETE, digunakan untuk Penghapusan data pada tabel *database*
- d. SELECT digunakan untuk menampilkan data pada tabel.

Pada MySQL sebuah *database* mengandung satu atau sejumlah tabel. Tabel terdiri dari sejumlah kolom dan baris, dimana setiap kolom berisi sekumpulan data yang memiliki tipe yang sejenis dan baris merupakan sekumpulan data yang saling berkaitan dan membentuk informasi. Kolom biasanya juga disebut sebagai *field* dan informasi yang tersimpan dalam baris disebut *record* (M. Rudyanto Arief, 2011).

H. Perancangan sistem

Perancangan sistem merupakan pengembangan sistem baru untuk mengatasi masalah pada sistem yang lama. Model perancangan sistem yang akan dipakai adalah metode analisis sistem terstruktur atau *Waterfall Model*. *Waterfall* adalah sebuah model perkembangan perangkat lunak dilakukan secara sekuensial, dimana satu tahap dilakukan setelah tahap sebelumnya selesai dilaksanakan. Model *waterfall* menurut Roger S. Pressman dalam (Udhi Sapto V., 2014) tahapan pada *Waterfall Model* disajikan pada gambar 4 sebagai berikut:



Gambar 2.6 Tahapan *Waterfall Model*

Uraian tahap-tahap pada *Waterfall Model* adalah sebagai berikut:

1. **Analisis** adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan perancangan sistem.
2. **Desain** adalah tahap penerjemah atau tahap perancangan dari keperluan-keperluan yang dianalisis dalam bentuk yang lebih mudah dimengerti oleh pemakai.
3. **Kode** adalah tahap implementasi dari hasil sistem yang telah dirancang dalam bahasa pemrograman yang telah ditentukan dan digunakan dalam pembuatan sistem.
4. **Tes** adalah tahap pengujian terhadap program yang telah dibuat. Pengujian dilakukan agar fungsi-fungsi dalam sistem bebas dari error, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

I. Kualitas sistem

Ada beberapa faktor yang mempengaruhi kualitas sistem. Faktor yang sering digunakan oleh pengembang *software* adalah faktor McCall. McCall dan kawan-kawan pada tahun 1977 telah mengusulkan suatu penggolongan faktor-

faktor atau kriteria yang mempengaruhi kualitas software. Pada dasarnya, McCall menitikberatkan faktor-faktor tersebut menjadi tiga aspek penting (Richardus E. Indrajit, 2012), aspek aspek tersebut adalah:

1. Sifat-sifat operasional dari *software* (*Product Operations*)

Pada aspek ini hal-hal yang diukur adalah yang berhubungan dengan teknis analisis, perancangan, dan konstruksi sebuah *software*. Faktor- faktor McCall yang berkaitan dengan sifat-sifat operasional *software* adalah

- a. ***Correctness***: sejauh mana suatu perangkat lunak memenuhi spesifikasi dan *mission objective* dari *user*.
- b. ***Reliability***: sejauh mana suatu perangkat lunak dapat diharapkan untuk melaksanakan fungsinya dengan ketelitian yang diperlukan.
- c. ***Efficiency***: banyaknya sumber daya komputasi dan kode program yang dibutuhkan suatu perangkat lunak untuk melakukan fungsinya.
- d. ***Integrity***: sejauh mana akses ke perangkat lunak dan data oleh pihak yang tidak berhak dapat dikendalikan.
- e. ***Usability***: usaha yang diperlukan untuk mempelajari, mengoperasikan, menyiapkan *input*, dan mengartikan *output* dari perangkat lunak.

2. Kemampuan *software* dalam menjalani perubahan (*Product Revision*)

Sebuah *software* yang dirancang dan dikembangkan dengan baik, akan dengan mudah dapat diperbaiki atau dimodifikasi jika diperlukan. Seberapa jauh *software* tersebut dapat diperbaiki merupakan faktor yang harus diperhatikan.

Faktor-faktor McCall yang berkaitan dengan kemampuan *software* untuk menjalani perubahan yaitu :

- a. **Maintainability:** usaha yang diperlukan untuk menetapkan dan memperbaiki kesalahan dalam program.
 - b. **Testability:** usaha yang diperlukan untuk menguji program untuk memastikan bahwa program melaksanakan fungsi yang ditetapkan.
 - c. **Flexibility:** usaha yang diperlukan untuk memodifikasi program operasional.
3. Daya adaptasi atau penyesuaian *software* terhadap lingkungan baru (*Product Transition*).

Faktor transisi yaitu bagaimana *software* tersebut dapat dijalankan pada beberapa platform atau kerangka sistem yang beragam. Faktor-faktor McCall yang berkaitan dengan tingkat adaptasi *software* terhadap lingkungan baru adalah :

- a. **Portability:** usaha yang diperlukan untuk memindahkan program dari perangkat keras / lingkungan sistem perangkat lunak tertentu ke yang lainnya.
- b. **Reusability:** tingkat kemampuan program / bagian dari program yang dapat dipakai ulang dalam aplikasi lainnya, berkaitan dengan paket dan lingkup dari fungsi yang dilakukan oleh program.
- c. **Interoperability:** usaha yang diperlukan untuk menggabungkan satu sistem dengan yang lainnya.