

BAB 2

LANDASAN TEORI

2.1 Kecerdasan Buatan

Pada topik ini akan membahas tentang definisi kecerdasan buatan, konsep kecerdasan buatan dan lingkup kecerdasan buatan pada aplikasi komersial.

2.1.1 Definisi Kecerdasan Buatan

Kecerdasan Buatan atau *Artificial Intelligence* didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Kecerdasan buatan ini merupakan cabang dari ilmu komputer yang *concerned* dengan pengotomatisasi tingkah laku cerdas (Anita Desiani dan Muhammad Arhami, 2006). Karena itu kecerdasan buatan harus didasarkan pada prinsip-prinsip teoretikal dan terapan yang menyangkut struktur data yang digunakan dalam representasi pengetahuan (*knowledge representation*), algoritma yang diperlukan dalam penerapan pengetahuan itu, serta bahasa dan teknik pemrograman yang dipakai dalam implementasinya. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan jaman, maka peran komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia.

Lebih detailnya, pengertian kecerdasan buatan dapat dipandang dari berbagai sudut pandang, antara lain:

1) Sudut pandang kecerdasan

Kecerdasan buatan akan membuat mesin menjadi 'cerdas' (mampu berbuat seperti apa yang dilakukan oleh manusia).

2) Sudut pandang penelitian

Kecerdasan buatan adalah suatu studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia.

Domain penelitian dalam kecerdasan buatan yaitu:

a) *Mundane task*

- Persepsi (*vision & speech*).
- Bahasa alami (*understanding, generation & translation*).
- Pemikiran yang bersifat *commonsense*.
- Robot control.

b) *Formal task*

- Permainan/*games*.
- Matematika (geometri, logika, kalkulus integral, pembuktian).

c) *Expert task*

- Analisis finansial.
- Analisis medikal.
- Analisis ilmu pengetahuan.
- Rekayasa (desain, pencarian kegagalan, perencanaan manufaktur).

3) Sudut pandang bisnis

Kecerdasan buatan adalah kumpulan peralatan yang sangat *powerful* dan metodologis dalam menyelesaikan masalah-masalah bisnis.

4) Sudut pandang pemrograman

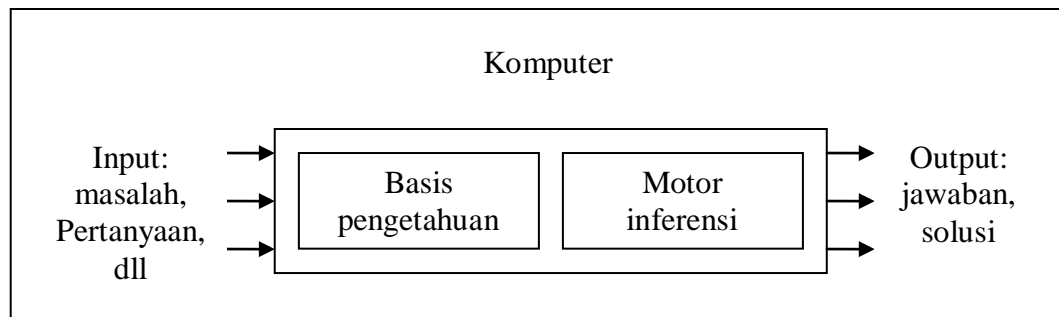
Kecerdasan buatan meliputi studi tentang pemrograman simbolik, penyelesaian masalah (*problem solving*) dan pencarian (*searching*).

Untuk melakukan aplikasi kecerdasan buatan ada dua bagian utama yang sangat dibutuhkan yaitu:

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

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

Gambar 2.1 berikut memperlihatkan penerapan konsep kecerdasan buatan di komputer.



Gambar 2.1 Penerapan konsep kecerdasan buatan di komputer

2.1.2 Konsep Kecerdasan Buatan

Ada beberapa konsep yang harus dipahami dalam kecerdasan buatan, diantaranya:

1. Turing Test – Metode Pengujian Kecerdasan.

Turing Test merupakan sebuah metode pengujian kecerdasan yang dibuat oleh Alan Turing. Dalam konsep ini, penanya (manusia) akan diminta untuk membedakan yang mana merupakan jawaban manusia dan mana yang merupakan jawaban komputer. Apabila tidak dapat membedakan, maka Turing berpendapat bahwa mesin tersebut dapat di asumsikan CERDAS.

2. Pemrosesan Simbolik

Komputer semula didesain untuk pemrosesan numerik, sedangkan manusia dalam berfikir dan menyelesaikan masalah lebih bersifat simbolik. Sifat penting dari AI adalah bagian dari ilmu komputer yang

melakukan proses secara simbolik dan non algoritmik dalam menyelesaikan masalah.

3. Heuristic

Heuristic merupakan suatu strategi untuk melakukan proses pencarian (search) ruang problem secara selektif, yang memandu proses pencarian yang kita lakukan sepanjang jalur yang memiliki kemungkinan sukses paling besar.

4. Penarikan Kesimpulan (Inferencing)

AI mencoba membuat mesin memiliki kemampuan berfikir atau mempertimbangkan (reasoning). kemampuan berfikir (reasoning) termasuk didalamnya proses penarikan kesimpulan (inferencing) berdasarkan fakta-fakta dan aturan dengan menggunakan metode heuristik atau pencarian lainnya.

5. Pencocokan Pola (Pattern Matching)

AI bekerja dengan metode pencocokan pola (Pattern Matching) yang berusaha untuk menjelaskan objek, kejadian (event) atau proses, dalam hubungan logik atau komputasional.

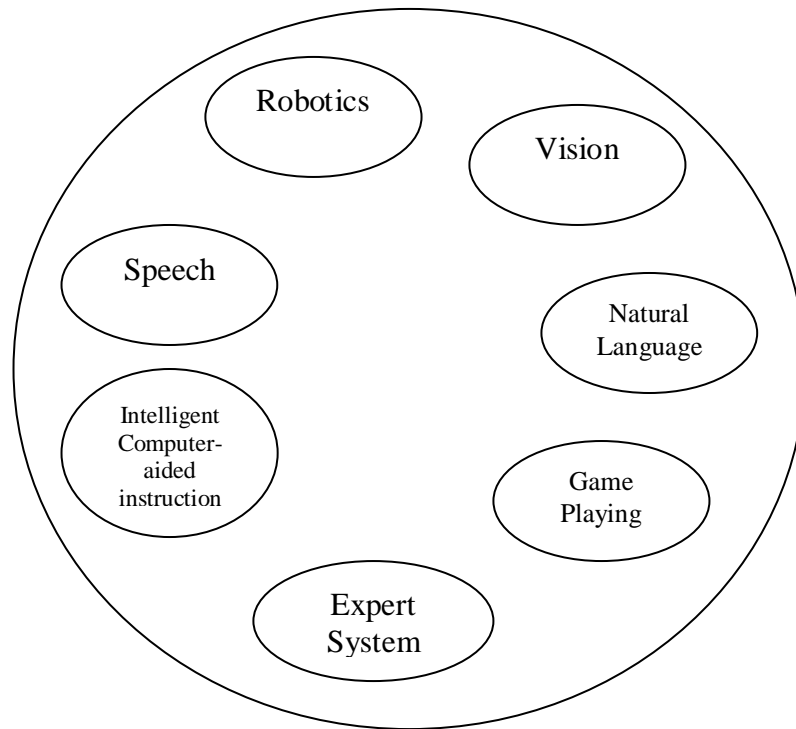
2.1.3 Lingkup Kecerdasan Buatan pada Aplikasi Komersial

Makin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran kecerdasan buatan. Karakteristik 'cerdas' sudah mulai dibutuhkan di berbagai disiplin ilmu dan teknologi. Kecerdasan buatan tidak hanya dominan di bidang ilmu komputer (informatika), namun juga sudah merambah di berbagai disiplin ilmu yang lain. Irisan antara psikologi dan kecerdasan buatan melahirkan sebuah area yang dikenal dengan nama *Cognition & Psycholinguistics*. Irisan antara teknik elektro dengan kecerdasan buatan melahirkan berbagai ilmu seperti: pengolahan citra, teori kendali, pengenalan pola dan robotika.

Adanya irisan penggunaan kecerdasan buatan di berbagai disiplin ilmu tersebut menyebabkan cukup rumitnya untuk mengklasifikasikan kecerdasan buatan menurut disiplin ilmu yang menggunakannya. Untuk memudahkan hal tersebut, maka pengklasifikasian lingkup kecerdasan buatan didasarkan pada output yang diberikan yaitu aplikasi komersial (meskipun sebenarnya kecerdasan buatan itu sendiri bukan merupakan medan komersial). Lingkup utama dalam kecerdasan buatan adalah:

- 1) Sistem Pakar (*Expert System*). Disini komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.
- 2) Pengolahan Bahasa Alami (*Natural Language 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 menggunakan suara.
- 4) Robotika & Sistem Sensor (*Robotics & Sensory Systems*).
- 5) *Computer Vision*, mencoba untuk dapat menginterpretasikan Gambar atau objek-objek tampak melalui komputer.
- 6) *Intelligent Computer-Aided Instruction*. Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
- 7) *Game Playing*.

Lingkup kecerdasan buatan yang dijelaskan di atas dapat digambarkan pada Gambar 2.2.



Gambar 2.2 Lingkup kecerdasan buatan

(Muhammad Arhami ,2004)

Metodologi-metodologi yang digunakan dalam Soft computing adalah :

1. Sistem Fuzzy (mengakomodasi ketidaktepatan) □ Logika Fuzzy (fuzzy logic)
2. Jaringan Syaraf (menggunakan pembelajaran) □ Jaringan Syaraf Tiruan (neurall network)
3. Probabilistic Reasoning (mengakomodasi ketidakpastian)
4. Evolutionary Computing (optimasi) □ Algoritma Genetika

2.2 Sistem Pakar

Pada topik ini akan membahas tentang defenisi sistem pakar, ciri-ciri sistem pakar, manfaat sistem pakar, kelebihan sistem pakar, kelemahan sistem pakar, konsep umum sistem pakar dan struktur sistem pakar.

2.2.1 Defenisi Sistem Pakar

Sistem pakar merupakan cabang dari kecerdasan buatan dan juga merupakan bidang ilmu yang muncul seiring perkembangan ilmu komputer saat ini. Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut (Kusrini, 2006). Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Dengan sistem pakar ini, orang awampun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman.

Ada beberapa defenisi tentang sistem pakar, antara lain:

- 1) Menurut Durkin: sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan oleh 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 bisa menyamai atau meniru kemampuan seorang pakar.

Sistem pakar pertama kali dikembangkan oleh komunitas AI pada pertengahan tahun 1960. Sistem pakar yang muncul pertama kali adalah *General-purpose Problem Solver* (GPS) yang dikembangkan oleh Newel dan Simon. GPS ini mengalami kegagalan dikarenakan cakupannya terlalu luas sehingga terkadang

justru meninggalkan pengetahuan-pengetahuan penting yang seharusnya disediakan.

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah. Beberapa aktivitas pemecahan yang dimaksud antara lain: pembuatan keputusan (*decision making*), pemanduan pengetahuan (*knowledge fusing*), pembuatan desain (*designing*), perencanaan (*planning*), prakiraan (*forecasting*), pengaturan (*regulating*), pengendalian (*controlling*), diagnosis (*diagnosing*), perumusan (*prescribing*), penjelasan (*explaining*), pemberian nasihat (*advising*) dan pelatihan (*tutoring*). Selain itu sistem pakar juga dapat berfungsi sebagai asisten yang pandai dari seorang pakar (Kusrini, 2006).

Sampai saat ini sudah banyak sistem pakar yang dibuat, beberapa contoh diantaranya terlihat pada Tabel 2.1.

Tabel 2.1 Contoh sistem pakar

Sistem pakar	Kegunaan
MYCIN	Diagnosa penyakit
DENDRAL	Mengidentifikasi struktur molekular campuran yang tak dikenal
XCON & XSEL	Membantu konfigurasi sistem komputer besar
SOPHIE	Analisis sirkit elektronik
Prospector	Digunakan di dalam geologi untuk membantu mencari dan menemukan deposit
FOLIO	Membantu memberikan keputusan bagi seorang manajer dalam hal stok <i>broken</i> dan investasi
DELTA	Pemeliharaan lokomotif listrik disel

2.2.2 Ciri-Ciri Sistem Pakar

Sistem pakar merupakan program-program praktis yang menggunakan strategi heuristik yang dikembangkan oleh manusia untuk menyelesaikan permasalahan-permasalahan yang spesifik (khusus), disebabkan oleh keheuristikannya dan

sifatnya yang berdasarkan pada pengetahuan sehingga umumnya sistem pakar mempunyai ciri-ciri sebagai berikut (E. Turban, 1995):

- 1) Terbatas pada domain keahlian tertentu.
- 2) Berdasarkan pada kaidah/*rule* tertentu.
- 3) Dapat digunakan dalam berbagai jenis komputer.
- 4) Mudah dimodifikasi, yaitu dengan menambah atau menghapus suatu kemampuan dari basis pengetahuannya.
- 5) Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pemakai.

2.2.3 Manfaat Sistem Pakar

Secara garis besar, banyak manfaat yang dapat diambil dengan adanya sistem pakar, antara lain (Sri Kusumadewi, 2003):

- 1) Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
- 2) Meningkatkan produktivitas akibat meningkatnya kualitas hasil pekerjaan, meningkatnya kualitas pekerjaan ini disebabkan meningkatnya efisiensi kerja.
- 3) Menghemat waktu kerja.
- 4) Menyederhanakan pekerjaan.
- 5) Merupakan arsip terpercaya dari sebuah keahlian, sehingga bagi pemakai sistem pakar seolah-olah berkonsultasi langsung dengan sang pakar, meskipun mungkin sang pakar telah tiada.
- 6) Memperluas jangkauan, dari keahlian seorang pakar. Di mana sebuah sistem pakar yang telah disahkan, akan sama saja artinya dengan seorang pakar yang tersedia dalam jumlah besar (dapat diperbanyak dengan kemampuan yang persis sama), dapat diperoleh dan dipakai di mana saja.

2.2.4 Kelebihan Sistem Pakar

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihanannya, seperti:

- 1) Meningkatkan ketersediaan (*increased availability*). Kepakaran/keahlian menjadi tersedia dalam sistem komputer. Dapat dikatakan bahwa sistem pakar merupakan produksi kepakaran secara masal (*mass production*).
- 2) Mengurangi biaya (*reduced cost*). Biaya yang diperlukan untuk menyediakan keahlian per satu orang *user* menjadi berkurang.
- 3) Mengurangi bahaya (*reduced danger*). Sistem pakar dapat digunakan di lingkungan yang mungkin berbahaya bagi manusia.
- 4) Permanen (*permanence*). Sistem pakar dan pengetahuan yang terdapat di dalamnya bersifat lebih permanen dibandingkan manusia yang dapat merasa lelah, bosan dan pengetahuannya hilang saat sang pakar meninggal dunia.
- 5) Keahlian multipel (*multiple expertise*). Pengetahuan dari beberapa pakar dapat dimuat ke dalam sistem dan bekerja secara simultan dan kontiniu menyelesaikan suatu masalah setiap saat. Tingkat keahlian/pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan satu orang pakar.
- 6) Meningkatkan kehandalan (*increased reliability*). Sistem pakar meningkatkan kepercayaan dengan memberikan hasil yang benar sebagai alternatif pendapat dari seorang pakar atau sebagai penengah jika terjadi konflik antara beberapa pakar. Namun hal tersebut tidak berlaku, jika sistem dibuat oleh salah seorang pakar sehingga akan selalu sama dengan pendapat pakar tersebut kecuali jika sang pakar melakukan yang mungkin terjadi pada saat tertekan atau stres.
- 7) Penjelasan (*explanation*). Sistem pakar dapat menjelaskan detail proses penalaran (*reasoning*) yang dilakukan hingga mencapai suatu kesimpulan. Seorang pakar mungkin saja terlalu lelah, tidak bersedia atau tidak mampu melakukannya setiap waktu. Dan hal ini akan meningkatkan tingkat kepercayaan bahwa kesimpulan yang dihasilkan adalah benar.
- 8) Respon yang cepat (*fast response*). Respon yang cepat atau *real-time* diperlukan pada beberapa aplikasi. Meskipun bergantung pada *hardware* dan *software* yang digunakan, namun sistem pakar relatif memberikan respon yang lebih cepat dibandingkan seorang pakar.

- 9) Stabil, tidak emosional dan memberikan respon yang lengkap setiap saat (*steady, unemotional, and complete response at all times*). Karakteristik ini diperlukan pada situasi *real-time* dan keadaan darurat (*emergency*) ketika seorang pakar mungkin tidak berada pada kondisi puncak disebabkan oleh stres atau kelelahan.
- 10) Pembimbing pintar (*intelligent tutor*). Sistem pakar dapat berperan sebagai *intelligent tutor* dengan memberikan kesempatan pada *user* untuk menjalankan contoh program dan menjelaskan proses *reasoning* yang dilakukan.
- 11) Basis data cerdas (*intelligent database*). Sistem pakar dapat digunakan untuk mengakses basis data secara cerdas.

Proses pengembangan sistem pakar memberikan keuntungan secara tidak langsung karena pengetahuan yang dimiliki oleh pakar harus dimuat ke dalam sistem secara eksplisit. Dengan demikian, pengetahuan ini dapat diperiksa kebenaran, konsisten dan kelengkapannya dan dapat disesuaikan untuk meningkatkan kualitasnya.

2.2.5 Kelemahan Sistem Pakar

Di samping memiliki beberapa manfaat dan kelebihan, sistem pakar juga memiliki beberapa kelemahan, antara lain (Sri Kusumadewi, 2003):

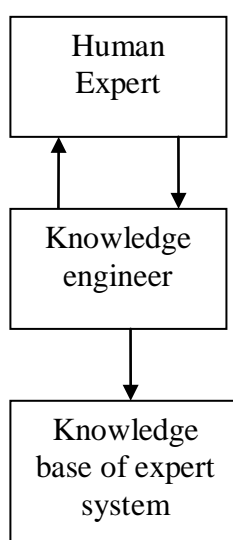
- 1) Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
- 2) Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar di bidangnya.
- 3) Sistem pakar tidak 100% bernilai benar

2.2.6 Konsep Umum Sistem Pakar

Pengetahuan yang dimiliki sistem pakar direpresentasikan dalam beberapa cara. Salah satu metode yang paling umum digunakan adalah tipe *rules* menggunakan format *IF THEN*. Banyak pendekatan berbasis pengetahuan (*knowledge approach*) untuk membangun sistem pakar telah mematahkan pendekatan awal

yang digunakan pada sekitar tahun 1950-an dan 1960-an yang menggunakan teknik penalaran (*reasoning*) yang tidak mengandalkan pengetahuan.

Pengetahuan tidak tertulis yang dimiliki oleh seorang pakar harus diekstraksi melalui wawancara secara ekstensif oleh *knowledge engineer*. Proses pengembangan sistem pakar yang berhubungan dengan perolehan pengetahuan dari pakar maupun sumber lain dan kodingnya disebut sebagai *knowledge engineering* yang dilaksanakan oleh *knowledge engineer*. Tahapan pengembangan sistem pakar secara umum terGambar pada Gambar 2.3.



Gambar 2.3 Tahapan pengembangan sistem pakar

Tahap awal, *knowledge engineer* melakukan diskusi dengan pakar untuk mengumpulkan pengetahuan yang dimiliki pakar yang bersangkutan. Tahap ini serupa dengan proses diskusi persyaratan/kebutuhan yang dilakukan *system engineer* pada sistem konvensional dengan kliennya. Setelah itu *knowledge engineer* melakukan koding pengetahuan secara eksplisit ke dalam *knowledge base*. Pakar kemudian mengevaluasi sistem pakar dan memberikan kritik. Proses ini berlangsung secara iteratif hingga dinilai sesuai oleh pakar.

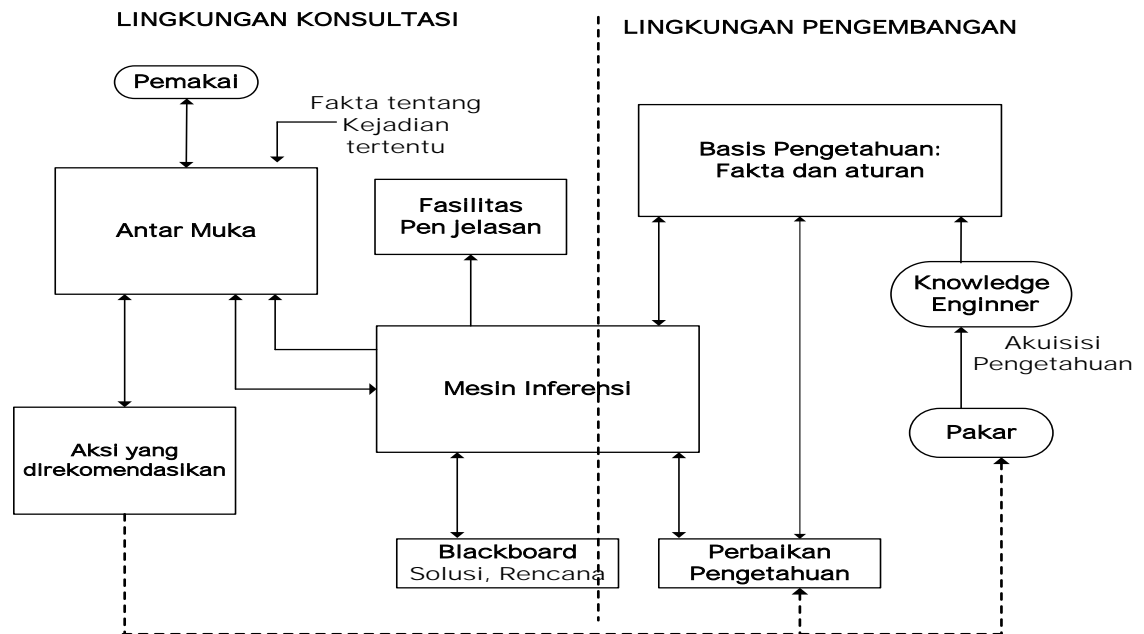
Sistem pakar umumnya dirancang dengan cara yang berbeda dengan sistem konvensional lain, terutama karena masalah yang dihadapi umumnya tidak

memiliki solusi algoritmik dan bergantung pada inferensi untuk mendapatkan solusi yang terbaik yang paling mungkin (*reasonable*). Oleh karena itu sistem pakar harus mampu menjelaskan inferensi yang dilakukannya sehingga hasil yang diperoleh dapat diperiksa. *Explanation facility* (fasilitas untuk menjelaskan) merupakan bagian terintegrasi dari sebuah sistem pakar. Sebuah *explanation facility* yang detail dirancang untuk memungkinkan *user* mengeksplorasi *rules* melalui tipe pertanyaan “*what if*” yang disebut *hypothetical reasoning* dan bahkan menerjemahkan *natural language* ke dalam *rules*. Beberapa sistem pakar bahkan mampu belajar membentuk *rules* (*learn rules by example*) dengan cara induksi (*rule induction*) dari Tabel data.

Memformalisasi pengetahuan pakar ke dalam *rules* tidaklah sederhana, terutama jika pengetahuan tersebut belum pernah disusun secara sistematis sebelumnya. Mungkin terjadi masalah-masalah seperti inkonsistensi, ambiguitas dan duplikasi. Seorang pakar juga mengetahui batas pengetahuan yang mereka miliki dan membatasi saran yang mereka berikan.

2.2.7 Struktur Sistem Pakar

Sistem pakar terdiri dari dua bagian pokok, yaitu: lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*) (Sri kusumadewi, 2003). Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar. Komponen-komponen sistem pakar dalam kedua bagian tersebut terlihat dalam Gambar 2.4, yaitu *user interface* (antarmuka pengguna), basis pengetahuan, akuisisi pengetahuan, mesin inferensi, *workplace*, fasilitas penjelasan dan perbaikan pengetahuan.



Gambar 2.4 Arsitektur sistem pakar (sumber: Turban, 1995).

Keterangan Gambar:

1. Antar Muka

Antar Muka merupakan mekanisme yang disediakan oleh pengguna sistem pakar untuk berkomunikasi.

2. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang objek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

3. Akuisisi Pengetahuan

Akuisisi Pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer.

4. Mesin Inferensi

Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada didalam basis pengetahuan

dan dalam *workplace dan* untuk memformulasikan kesimpulan (Turban, 1995).

5. *Workplace*

Workplace merupakan area dari sekumpulan memory kerja.

6. Fasilitas Penjelasan

Fasilitas penjelasan merupakan komponen tambahan yang akan meningkatkan kemampuan sistem pakar.

2.3 Representasi Pengetahuan

Akuisisi pengetahuan dilakukan untuk memperoleh pengetahuan dari pakar atau sumber lain (sumber terdokumentasi, buku, sensor, file komputer, dll.). Dalam proses akuisisi pengetahuan, seorang perekayasa pengetahuan menjembatani antara pakar dengan basis pengetahuan. Perekayasa pengetahuan mendapatkan pengetahuan dari pakar, mengolahnya bersama pakar tersebut dan menaruhnya dalam basis pengetahuan, dengan format tertentu.

2.3.1 Model Representasi Pengetahuan

Setelah pengetahuan berhasil diakuisisi, pengetahuan harus diorganisasi dan diatur dalam suatu konfigurasi dengan suatu format/ representasi tertentu. Metode representasi pengetahuan yang penting adalah:

1. Logika (logic)

Logika merupakan suatu proses pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah dan prosedur yang membantu proses penalaran.

2. Jaringan Semantik (semantic nets)

Jaringan Semantik merupakan teknik representasi kecerdasan buatan klasik yang digunakan untuk informasi proporsional (Giarrantano dan Riley, 1994)

3. Object Atributte Value (OAV)

4. Bingkai (frame)

Bingkai berupa ruang- ruang yang berisi atribut untuk mengdeskripsikan pengetahuan.

5. Kaidah Produksi (production rule).

2.3.2 Kaidah Produksi

Kaidah menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Pada aturan produk atau kaidah produksi pengetahuan disajikan dalam aturan-aturan yang berbentuk pasangan keadaan-aksi (*condition-action*): “JIKA keadaan terpenuhi atau terjadi MAKA suatu aksi akan terjadi”. Sistem Pakar yang basis pengetahuannya selalu disajikan dalam bentuk aturan produk disebut sistem berbasis-aturan (*rule-based sistem*).

Dari *decision table* yang telah tereduksi, setiap barisnya dapat dikonversikan menjadi *IF_THEN Rule*. Setiap baris pada *decision table* yang telah tereduksi akan membentuk satu *set rule* final. Struktur dan penulisan *rule* adalah sebagai berikut :

1. *RULE* label : Tabel yang berisi nama rule tersebut.
2. *IF* : sebagai penanda awal kondisi.
3. *THEN* : sebagai penanda awal kesimpulan pada sebuah *rule*.
4. *ELSE* : sebagai penanda awal alternatif kesimpulan pada sebuah *rule*, bersifat opsional, jadi boleh tidak ada.

Operator yang dapat digunakan pada *IF-THEN rule* adalah :

- a. 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, keseluruhan *rule* tersebut bernilai salah.
- b. OR : Bila semua kondisi yang dihubungkan oleh operator ini bernilai salah, maka kondisi keseluruhan *rule* tersebut bernilai salah, bila ada salah satu kondisi atau lebih yang bernilai benar, keseluruhan *rule* tersebut bernilai benar.

2.4 Mekanisme Inferensi

Mekanisme Inferensi adalah bagian dari sistem pakar yang melakukan penalaran dengan menggunakan isi daftar aturan berdasarkan urutan dan pola tertentu. Selama proses konsultasi antar sistem dengan pemakai, mekanisme inferensi menguji aturan satu sampai kondisi aturan itu benar.

Secara umum ada dua teknik utama yang digunakan dalam mekanisme inferensi untuk pengujian aturan, yaitu penalaran maju (*Forward Chaining*) dan penalaran mundur (*Backward Chaining*).

2.4.1 Metode Forward Chaining

Metode *Forward Chaining* adalah suatu metode pengambilan keputusan yang umum digunakan dalam system pakar. Proses pencarian dengan metode Forward Chaining berangkat dari kiri ke kanan, yaitu dari premis menuju kepada kesimpulan akhir, metode ini sering disebut *data driven* yaitu pencarian dikendalikan oleh data yang diberikan (Sri Hartati dan Iswanti, Sari,2008).

Aktivitas sistem dilakukan berdasarkan siklus mengenal-beraksi. Mula-mula, sistem mencari semua aturan yang kondisinya terdapat di memori kerja, kemudian memilih salah satunya dan menjalankan aksi yang bersesuaian dengan aturan tersebut. Pemilihan aturan yang akan dijalankan berdasarkan strategi tetap

yang disebut strategi penyelesaian konflik. Aksi tersebut menghasilkan memori kerja baru dan siklus diulangi lagi sampai tidak ada aturan yang dapat dipicu, atau tujuan yang dikehendaki sudah terpenuhi. Untuk lebih jelasnya dapat melihat contoh Tabel 2.2.

Tabel 2.2 Contoh Aturan menggunakan penalaran *Forward Chaining* (Alur Maju)

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

Pada Tabel 2.2 terlihat ada 10 aturan yang tersimpan dalam basis pengetahuan. Fakta awal yang diberikan hanya : A dan E (artinya : A dan E bernilai benar). Ingin dibuktikan apakah K bernilai benar (hipotesis K) ?

Langkah-langkah inferensi adalah sebagai berikut :

1. Dimulai dari R-1, A merupakan fakta sehingga bernilai benar, sedangkan B belum bias diketahui kebenarannya, sehingga C pun juga belum bisa diketahui kebenarannya. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-1 ini. Sehingga kita menuju ke R-2.
2. Pada R-2 kita tidak mengetahui informasi apapun tentang C, sehingga kita juga tidak bisa memastikan kebenaran D. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-1 ini. Sehingga kita menuju ke R-3.

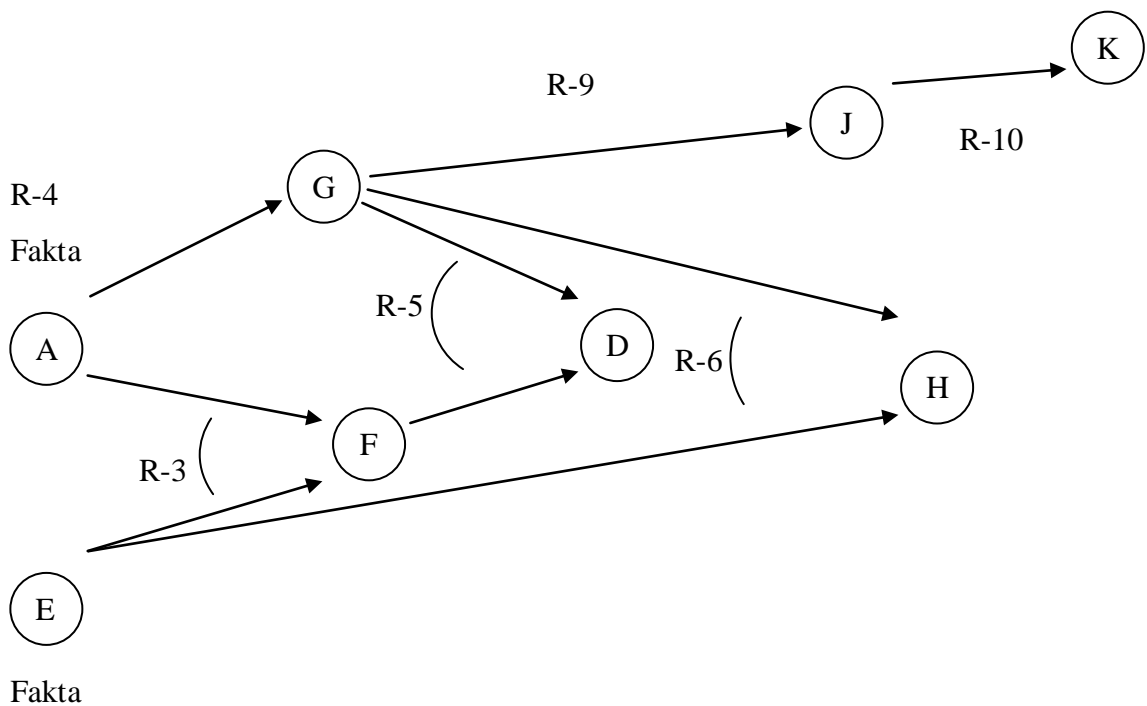
3. Pada R-3, baik A maupun E adalah fakta sehingga jelas benar. Dengan demikian F sebagai konsekuen juga ikut benar. Sehingga sekarang kita mempunyai fakta baru yaitu F. Karena F bukan hipotesis yang hendak kita buktikan maka penelusuran kita lanjutkan ke R-4.
4. Pada R-4, A adalah fakta sehingga jelas benar. Dengan demikian G sebagai konsekuen juga ikut benar. Sehingga sekarang kita mempunyai fakta baru yaitu G. Karena G bukan hipotesis yang hendak kita buktikan, maka penelusuran kita lanjutkan ke R-5.
5. Pada R-5, baik F maupun G bernilai benar berdasarkan aturan R-3 dan R-4. Dengan demikian G sebagai konsekuen juga ikut benar. Sehingga sekarang kita mempunyai fakta baru yaitu D. Karena D bukan hipotesis yang hendak kita buktikan, maka penelusuran kita lanjutkan ke R-6.
6. Pada R-6, baik A maupun G adalah benar berdasarkan fakta dari R-4. Dengan demikian H sebagai konsekuen juga ikut benar. Sehingga sekarang kita mempunyai fakta baru yaitu H. Karena H bukan hipotesis yang hendak kita buktikan, maka penelusuran kita lanjutkan ke R-7.
7. Pada R-7, meskipun H benar berdasarkan R-6, namun kita tidak tahu kebenaran C sehingga, I pun juga belum bisa diketahui kebenarannya. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-7 ini. Sehingga kita menuju ke R-8.
8. Pada R-8, meskipun A benar karena fakta, namun kita tidak tahu kebenaran I, sehingga J pun juga belum bisa diketahui kebenarannya. Oleh karena itu kita tidak mendapatkan informasi apapun pada R-8 ini. Sehingga kita menuju ke R-9.
9. Pada R-9, J bernilai benar karena G benar berdasarkan R-4. Karena J bukan hipotesis yang hendak kita buktikan, maka penelusuran kita lanjutkan ke R-10.

10. Pada R-10, K bernilai benar karena J benar berdasarkan R-9. Karena K sudah merupakan hipotesis yang hendak kita buktikan, maka terbukti bahwa K adalah benar.

Tabel munculnya fakta baru pada saat inferensi terlihat pada Tabel 2.3 sedangkan alur inferensi terlihat pada Gambar 2.5

Tabel 2.3 Fakta baru pada saat inferensi

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



Gambar 2.5 Alur inferensi Forward Chaining

2.4.2 Metode Backward Chaining

Metode *Backward Chaining* adalah suatu metode pengambilan keputusan yang juga umum digunakan dalam sistem pakar. Metode *backward chaining* adalah kebalikan dari *Forward Chaining*. Percobaan fakta atau pernyataan dimulai dari bagian sebelah kanan (*THEN* dulu). Dengan kata lain, penalaran dimulai dari hipotesis terlebih dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan. Proses pencarian dengan metode *backward chaining* berangkat dari kanan ke kiri, yaitu dari kesimpulan sementara menuju kepada premis, metode ini sering disebut *goal driven* pencarian dikendalikan oleh tujuan yang diberikan (Hartati, Sri dan Iswanti, Sari, 2008).

Metode *Backward Chaining* merupakan strategi pencarian yang arahnya kebalikan dari *Forward Chaining*. Proses pencarian dimulai dari tujuan, yaitu kesimpulan yang menjadi solusi permasalahan yang dihadapi. Mesin inferensi mencari kaidah-kaidah dalam basis pengetahuan yang kesimpulannya merupakan solusi yang ingin dicapai, kemudian dari kaidah-kaidah yang diperoleh, masing-masing kesimpulan *Backward Chaining* jalur yang mengarah ke kesimpulan tersebut.

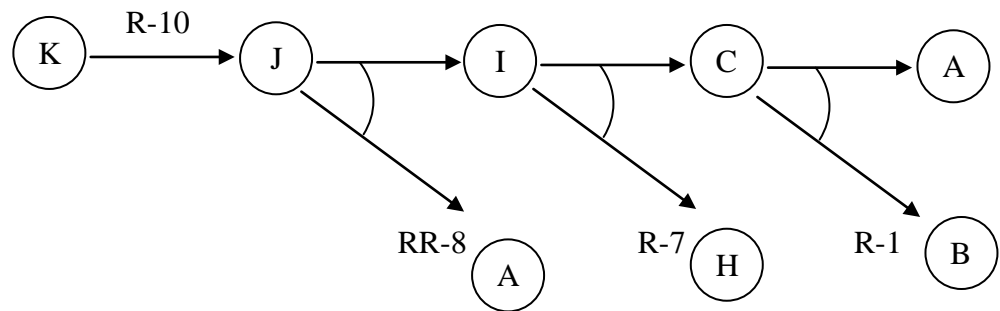
Jika informasi-informasi atau nilai dari atribut-atribut yang mengarah ke kesimpulan tersebut sesuai dengan data yang diberikan maka kesimpulan tersebut merupakan solusi yang dicari, jika tidak sesuai maka kesimpulan tersebut bukan merupakan solusi yang dicari. *Backward Chaining* memulai proses pencarian dengan suatu tujuan sehingga strategi ini disebut juga *goal-driven*.

Seperti halnya pada Tabel 2.2, terlihat ada 10 aturan yang tersimpan dalam basis pengetahuan. Fakta awal yang diberikan : A dan E (artinya A dan E bernilai benar). Ingin dibuktikan apakah A bernilai benar (hipotesis hanya K)?.

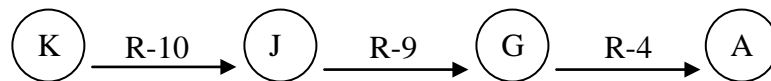
Langkah-langkah inferensi adalah sebagai berikut :

1. Pertama-tama kita cari terlebih dahulu mulai dari R-1, aturan yang mana memiliki konsekuen K. ternyata setelah ditelusur, aturan dengan konsekuen K baru ditemukan pada R-10. Untuk membuktikan bahwa K benar maka perlu dibuktikan bahwa J benar.
2. Kita cari aturan yang memiliki konsekuen J. kita mulai dari R-1, ternyata kita baru akan menemukan aturan dengan konsekuen J pada R-8. Untuk membuktikan bahwa J benar maka perlu dibuktikan bahwa I dan A benar. Untuk membuktikan kebenaran I, kita perlu cari aturan dengan konsekuen I, ternyata ada di R-7.
3. Untuk membuktikan bahwa I benar di R-7, kita perlu buktikan bahwa C dan H benar. Untuk itu kita pun perlu mencari aturan dengan konsekuen C dan ada di R-1.
4. Untuk membuktikan C benar di R-1, kita perlu buktikan bahwa A dan B benar. A jelas benar karena A merupakan fakta. Sedangkan B kita tidak bisa membuktikan kebenarannya, karena selain bukan fakta, di dalam basis pengetahuan juga tidak ada aturan dengan konsekuen B. Dengan demikian maka dari penalaran ini kita tidak bisa buktikan kebenaran dari hipotesis K. Namun demikian, kita masih punya alternative lain untuk melakukan penalaran.
5. Kita lakukan backtracking. Kita ulangi lagi dengan mencari aturan dengan konsekuen I, ternyata juga tidak ada.
6. Kita lakukan backtraining lagi dengan mencari aturan dengan konsekuen I, ternyata juga tidak ada.
7. Kita lakukan backtraining lagi dengan mencari aturan dengan konsekuen J, ternyata kita temukan pada R-9. sehingga kita perlu buktikan kebenaran G.
8. Kita mendapatkan di R-4 dengan konsekuen G. Kita perlu untuk membuktikan kebenaran A. karena A adalah fakta, maka terbukti bahwa K bernilai benar.

Alur inferensi dapat dilihat pada Gambar 2.6.



(a) Pertama : Gagal



(b) Kedua : Sukses

Gambar 2.6 Alur Inferensi Backward Chaining

2. 5 Gangguan Psikotik Akibat Zat

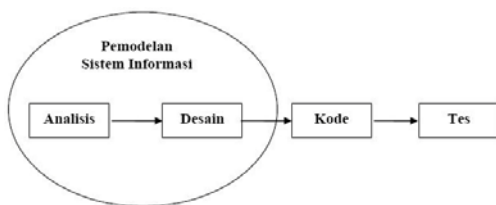
Psikotik adalah gangguan jiwa yang ditandai dengan ketidak mampuan individu menilai kenyataan yang terjadi, misalnya terdapat halusinasi, waham atau perilaku kacau/aneh. Zat psikoaktif adalah zat yang mempengaruhi aktivitas mental dan perilaku. Zat ini meliputi obat psikotropik dan narkotik. Obat psikotropik adalah obat yang bekerja secara selektif pada susunan saraf pusat (SSP) dan mempunyai efek utama terhadap aktivitas mental dan perilaku dan digunakan untuk terapi gangguan psikiatrik. Obat narkotik adalah obat yang bekerja secara selektif pada susunan saraf pusat (SSP) dan mempunyai efek utama terhadap perubahan atau penurunan kesadaran, hilangnya rasa dan mengurangi sampai menghilangkan rasa nyeri, digunakan untuk analgesik, antitusif, antispasmodik dan premedikasi anestesi.

Sekelompok gejala psikotik yang terjadi selama atau segera sesudah penggunaan zat psikoaktif. Gejalanya yaitu halusinasi, kekeliruan identifikasi,

waham dan/atau *ideas of reference* (gagasan tentang dirinya sebagai acuan) yang seringkali bersifat kecurigaan atau kejaran, gangguan psikomotor (excitement atau stupor) dan afek yang abnormal antara ketakutan yang mencekam hingga kesenangan yang berlebihan. Umumnya kesadarannya masih jernih. Variasi gejala dipengaruhi jenis zat yang digunakan dan kepribadian penggunanya.

2.6 Siklus Pengembangan Perangkat Lunak

Salah satu strategi yang sering dipakai sebagai *model proses* atau *paradigma rekayasa perangkat lunak* yaitu dengan menggunakan model *waterfall*. Model *waterfall* merupakan pendekatan perangkat lunak yang sistematis dan sekuensial yang dimulai dari tahap analisis, desain, kode, pengujian, dan pemeliharaan.



Tahap-tahap model *waterfall* meliputi aktivitas-aktivitas sebagai berikut :

1. Rekayasa dan pemodelan sistem/informasi.

Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan (*requirements*) pada tingkat sistem dengan sejumlah kecil analisis. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

2. Analisis kebutuhan perangkat lunak.

Tahap ini merupakan proses pengumpulan kebutuhan secara intensif dan terfokus, khususnya pada perangkat lunak.

3. Desain.

Tahap ini adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural.

4. Pengkodean .

Merupakan tahap penerjemahan dari proses desain ke dalam bentuk mesin yang bisa dibaca.

5. Pengujian.

Tahap ini berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Pemeliharaan.

Proses pemeliharaan perangkat lunak dan keseluruhan sistem bila terjadi kesalahan pada program, atau terjadi perubahan lingkungan perangkat lunak dan juga bila terjadi perubahan requirements dan maintenance yang bersifat preventif untuk mengantisipasi keadaan yang tidak diinginkan.