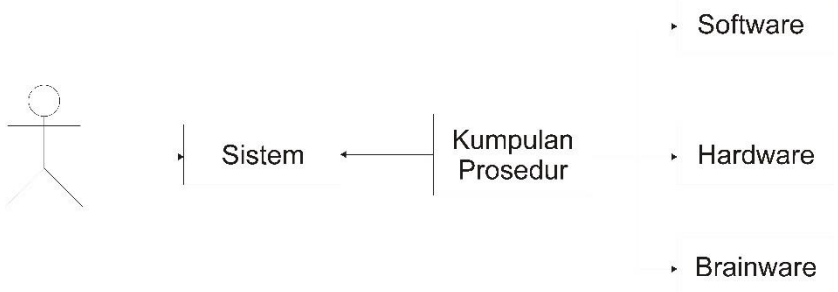
BAB II

LANDASAN TEORI

2.1. Konsep Sistem

Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan satu sama lain dan saling terhubung untuk melakukan suatu tugas secara bersama-sama.[3] Sebuah sistem sendiri terdiri dari tiga komponen utama antara lain *software*, *hardware*, dan *brainware*. Komponen-komponen tersebut saling berhubungan satu dengan yang lainnya.



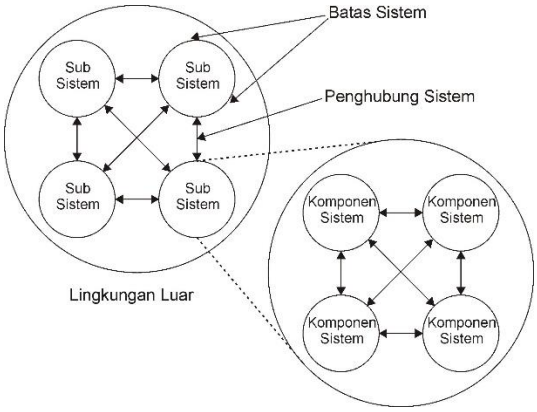
Gambar 2.1 Bagan Sistem, Prosedur, Pengguna dan Komponen (Sumber : Sistem Informasi dan Implementasinya[3, p. 7])

Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, untuk dilakukannya suatu kegiatan atau untuk diselesaikannya suatu tujuan tertentu.[4]

Pendekatan sistem yang menekankan pada komponennya mendefinisikan sistem sebagai dibentuknya satu kesatuan elemen-elemen yang berinteraksi satu dengan yang lainnya yang dikumpulkan untuk mencapai suatu sasaran tertentu.[4]

2.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik tersendiri diantaranya mempunyai komponen-komponen, batas sistem, lingkungan luar sistem, penghubung, dan tujuan.



Gambar 2.2 Karakteristik Suatu Sistem
(Sumber : Sistem Teknologi Informasi[4, p. 54])

1. Komponen Sistem

   Suatu sistem terdiri dari komponen yang berhubungan, artinya saling bekerja sama membentuk satu kesatuan. Komponen- komponen dapat berupa bagian-bagian dari sistem. Setiap bagian mempunyai sifat-sifat dari sistem yang menjalankan suatu tujuan tertentu dan mempengaruhi proses secara menyeluruh.[4]

2. Batas Sistem (*Boundary*)

   Batasan sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup sistem tersebut.[4]

3. Lingkungan Luar (*Environment*)

   Lingkungan luar (*environment*) dari suatu sistem merupakan sesuatu diluar batas dari sistem yang mempengaruhi proses sistem. Lingkungan luar yang menguntungkan merupakan masukan positif untuk sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan jika dimungkinkan untuk dihindari, kalau tidak maka akan mengganggu sistem tersebut.[4]

4. Penghubung (*Interface*)

   Penghubung (I*nterface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.[4]

5. Tujuan (*Goal*)

   Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Jika suatu tidak mempunyai sasaran maka operasi sistem tidak akan berguna. Sasaran dari sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang dihasilkan sistem.[4]

2.1.2 Klasifikasi Sistem

Berikut macam-macam klasifikasi sistem :

1. Sistem Abstrak dan Sistem Fisik

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (tidak terlihat). Sistem fisik merupakan sistem yang ada secara fisik (terlihat).[5]

2. Sistem Alamiah dan Sistem Buatan Manusia

   Sistem alamiah merupakan sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia, sistem ini melibatkan interaksi antara manusia dengan mesin. [5]

3. Sistem Tertentu dan Sistem Tak Tentu

   Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi antara bagian-bagiannya dapat dideteksi dengan pasti. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas. [5]

4. Sistem Tertutup dan Sistem Terbuka

   Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengn lingkungan luarnya. [5]

2.2. Konsep Informasi

Informasi merupakan hasil pengolahan data dari sumber yang memberikan nilai dan manfaat. Proses pengolahan data ini memerlukan teknologi untuk menjalankannya.[3]

Informasi merupakan salah satu bagian terpenting dalam suatu perusahaan atau organisasi, tanpa informasi suatu sistem menjadi tidak berjalan. Kualitas informasi ditentukan oleh bagaimana informasi tersebut dapat memberikan suatu dorongan kepada penerima informasi untuk bertindak sesuai informasi tersebut.

2.3. Konsep Sistem Informasi

Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.[4]

2.3.1. Komponen Sistem Informasi

Komponen sistem informasi memiliki fungsi dan tugas yang berhubungan satu sama lain yang membentuk suatu kesatuan yang membantu untuk mencapai tujuan atau sasaran. Berikut komponen-konponen tersebut :

1. *Input* (masukan)

   Komponen input ini berfungsi sebagai penerima semua masukan dari pengguna *(user)* berupa data. Data tersebut berasal dari satu atau beberapa sumber.[3]

2. *Output* (keluaran)

   Sistem informasi akan menghasilkan keluatan *(output)* berupa informasi. Komponen ini berfungsi untuk menyajikan hasil akhir untuk pengguna *(user)* sistem informasi tersebut. Informasi yang dihasilkan akan sesuai dengan data yang diinputkan dan fungsionalitas dari sistem.[3]

3. *Software* (perangkat lunak)

   Adanya komponen perangkat lunak ini akan membantu sistem informasi dalam menjalankan tugasnya untuk mencapai tujuan yang ditentukan. Tugas komponen ini ialah melakukan proses pengolahan data, penyajian

informasi, menghitung data, dan lain-lain. Komponen perangkat lunak mencakup sistem operasi, aplikasi, dan *driver*.[3]

4. *Hardware* (perangkat keras)

   Komponen ini mencakup semua perangkat keras komputer yang digunakan secara fisik dalam sistem informasi. Komponen ini meliputi komputer beserta komponen didalamnya. Termasuk juga hub, switch, router, yang berperan di dalam jaringan komputer.[3]

5. *Database* (basis data)

   Komponen ini berfungsi sebagai penyimpanan semua data dan informasi kedalam satu atau beberapa tabel. Setiap tabel memiliki tugas atau fungsi masing-masing dan juga dapat berelasi satu sama lain.[3]

6. *Control and Procedure* (kontrol dan prosedur)

   Kontrol dan prosedur adalah dua buah komponen yang menjadi satu. Fungsinya adalah untuk mencegah terjadinya beragam gangguan dan ancaman terhadap data dan informasi yang ada didalam sistem informasi .[3]

7. Teknologi dan Jaringan Komputer

   Komponen ini memegang peranan yang penting untuk sistem informasi. Komponen ini berfungsi untuk mengatur *software, hardware, database, input, output,* kontrol dan prosedur agar sistem terkendali dengan baik.[3]

2.4. Kecerdasan Buatan

Kecerdasan buatan atau *artificial intelligence* merupakan bidang ilmu komputer yang mempunyai peran penting di era kini dan masa yang akan datang.

Bidang ini telah tumbuh berkembang pesat di 20 tahun terakhir seiring dengan pertumbuhan kebutuhan akan perangkat cerdas pada industri rumah tangga.[2, p. 2]

Bidang keilmuan kecerdasan buatan sampai saat ini terus mencoba untuk melakukan proses cara berpikir manusia seperti mengetahui, memahami, memprediksi, dan melakukan manipulasi hal – hal yang lebih besar dan lebih rumit dari yang pernah ada. Tidak hanya memecahkan berbagai masalah, tetapi juga untuk membangun sebuah sistem atau alat yang memiliki kecerdasan.

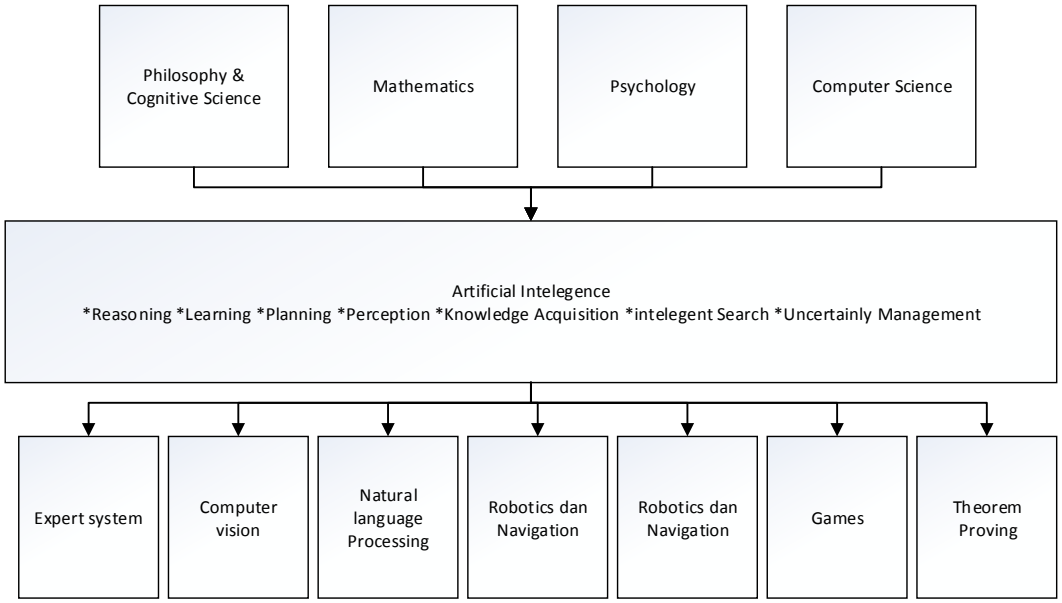
Kata *intelligence* berasal dari bahasa latin *intelligo* yang berarti ‘saya paham’. Jadi, dasar dari *intelligence* adalah kemampuan memahami dan melakukan aksi. Sebenarnya, area kecerdasan buatan bermula dari kemunculan komputer sekitar tahun 1940-an, meskipun sejarah perkembangannya dapat dilacak hingga zaman mesir kuno. Pada masa sekarang, perhatian difokuskan pada kemampuan komputer untuk mengerjakan sesuatu yang dapat dilakukan oleh manusia. Dalam hal ini, komputer tersebut dapat meniru kemampuan prilaku manusia.[2, p.3]

2.4.1. Bidang Ilmu Kecerdasan Buatan

Supaya dapat bertindak seperti manusia, komputer harus diberi bekal pengetahuan dan kemampuan menalar. Tujuan dari sistem kecerdasan buatan dapat dibagi menjadi 4 kategori : [2, p. 7]

1. Sistem yang dapat berpikir seperti manusia (Bellman, 1978)
2. Sistem yang dapat berpikir secara rasional (Winston, 1992)
3. Sistem yang dapat beraksi seperti manusia (Rich dan Knight, 1991)
4. Sistem yang dapat beraksi secara rasional (Nilsson, 1998)

Secara garis besar, bidang ilmu yang dipelajari AI dapat dilihat dalam gambar di bawah ini :



Gambar 2.3 Domain Area AI
(Sumber : Artificial Intelligence Konsep dan Penerapannya [2, p.7])

2.5. Definisi Kasus yang Dianalisis

2.5.1. Definisi Pakar

Pakar menurut Kamus Besar Bahasa Indonesia (KBBI) adalah (orang) ahli, spesialis.[6] Pakar adalah seseorang yang memiliki pengetahuan yang lebih dalam dalam suatu bidang.

2.5.2. Sistem Pakar

Sistem pakar adalah program komputer yang mensimulasi penilaian dan perilaku manusia atau organisasi yang memiliki pengetahuan dan pengalaman ahli dalam bidang tertentu.[2, p. 132] Sistem pakar juga merupakan sistem komputer yang mengemulasi kemampuan pengambilan keputusan seorang manusia ahli. [2, p. 133]

2.5.2.1. Kelebihan dan Karakteristik Sistem Pakar

Sistem pakar banyak digunakan pada aplikasi terkini dan kompleks karena [2, p. 134] :

1. Sistem pakar dapat bertindak sebagai konsultan, instruktur, atau pasangan/rekan
2. Meningkatkan *availability* atau kepakaran tersedia pada sebuah perangkat komputer
3. Mengurangi bahaya
4. Permanen
5. Pengetahuan dapat tidak lengkap, namun keahlian dapat diperluas sesuai kebutuhan. Program konvensional harus "lengkap" sebelum mereka dapat digunakan.
6. Database yang cerdas, sistem pakar dapat digunakan untuk mengakses database secara cerdas, misalnya data mining.

Sistem pakar biasanya didesain untuk memiliki karekteristik sebagai berikut :

1. *High Performance*

   Melakukan perhitungan dengan cepat dan akurat berdasarkan sumber pengetahuan dari banyak pakar.

2. *Adequate Response Time*

   Merespon dari request user dengan cepat karena sistem pakar di desain untuk bekerja dengan produktivitas tinggi.

3. *Good Reliability*

Memiliki kehandalan yang baik karena sistem pakar mengambil sumber pengetahuan dari banyak pakar

4. *Understandable*

   Informasi yang diberikan mudah dimengerti karena informasi yang diberikan oleh sistem pakar merupakan anjuran atau suatu nasihat yang dapat dipercaya.

Kesimpulannya, dengan sistem pakar dapat membantu banyak orang dengan cepat dan efisien dalam kebutuhan akan suatu informasi yang dapat memberikan petunjuk untuk memutuskan suatu pilihan keputusan.

2.5.3. Definisi Kerusakan

Kerusakan berasal dari kata rusak yang mempunyai arti sudah tidak sempurna (baik, utuh).[6] Kerusakan itu sendiri merupakan penjelasan suatu benda yang tidak sempurna.

2.5.4. Definisi Kendaraan

Kendaraan berasal dari kata kendara yang mempunyai arti sesuatu yang dirancang untuk perjalan dan kendaraan merupakan sesuatu yang digunakan untuk dikendarai atau dinaiki.[6]

2.5.5. *Forward Chaining*

Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui. Kemudian mencocokan dengan fakta – fakta tersebut dengan bagian IF dari rules IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan. Dimulai dari rule teratas untuk setiap kali pencocokan. Setiap rule

hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi rule yang bisa dieksekusi. Metode pencarian yang digunakan adalah *Depth - First Search* (DFS), *Breadth - First Search* (BFS) atau *Best First Search*.[7, p. 171]

Untuk memahami cara kerja *forward chaining*, perhatikan contoh berikut ini :

Misalkan diketahui sistem pakar menggunakan 10 rule berikut :

R1 : IF (hari != ‘minggu’) AND (baba == ‘sehat’) THEN baba_kuliah = true;

R2 : IF (baba_kuliah) THEN sylvia_kuliah = true;

R3 : IF (hari != ‘minggu’) AND (baba_sakit) THEN baba_berada = ‘RS’;

R4 : IF (hari != ‘minggu’) THEN mhs_UNIKOM_kuliah = true;

R5 : IF (baba_berada == ‘RS’) AND (mhs_UNIKOM_kuliah) THEN sylvia_kuliah = true;

R6 : IF (mhs_UNIKOM_kuliah) AND (baba_sakit) THEN sylvia_kuliah = true;

R7 : IF (baba_kuliah) AND (!sylvia_kuliah) THEN sylvia_sakit = true;

R8 : IF (hari != ‘minggu’) AND (sylvia_sakit) THEN kuliah_libur = false;

R9 : IF (mhs_UNIKOM_kuliah) THEN kuliah_libur = false;

R10 : IF (!kuliah_libur) THEN baba_berada = ‘KAMPUS’;

Semula diberikan fakta dua buah fakta berikut :

hari = ‘senin’;

baba_kuliah = true;

Untuk membuktikan apakah baba berada dikampus dan sylvia tidak kuliah, maka dapat dibuktikan dengan langkah – langkah berikut :

Berdasarkan fakta yang ada, sistem pakar berusaha menelusuri *rule – rule* dari bagian IF, yang dimulai dari *rule* R1 sampai dengan R10.

Iterasi ke – 1

Fakta baba_kuliah = true, memicu rule R2,

IF (baba_kuliah) THEN sylvia_kuliah = true;

Sehingga :

Fakta	Fakta Baru
hari = ‘senin’ baba_kuliah = true;	sylvia_kuliah = true;

Fakta hari = ‘senin’, memicu *rule* R4,

IF (hari != ‘minggu’) THEN mhs_UNIKOM_kuliah = true;

Sehingga :

Fakta	Fakta Baru
hari = ‘senin’ baba_kuliah = true;	sylvia_kuliah = true; mhs_UNIKOM_kuliah = true;

Iterasi ke – 2

Pada iterasi ke – 1 ditemukan fakta baru yaitu :

sylvia _kuliah = true;

mhs_UNIKOM_kuliah = true;

berdasarkan fakta baru maka berikut langkah – langkah yang dikerjakan oleh sistem pakar :

Fakta baru mhs_UNIKOM_kuliah = true memicu rule R9,

IF (mhs_UNIKOM_kuliah) THEN kuliah_libur = false;

Sehingga :

Fakta	Fakta Baru

hari = ‘senin’
baba_kuliah = true;
	sylvia_kuliah = true;
	mhs_UNIKOM_kuliah = true;
	kuliah_libur = false;

Iterasi ke – 3

Pada iterasi ke – 2 ditemukan fakta baru yaitu :

kuliah_libur = false;

berdasarkan fakta baru maka berikut langkah – langkah yang dikerjakan oleh sistem pakar :

Fakta baru kuliah_libur = false, memicu rule R10,

IF (!kuliah_libur) THEN baba_berada = ‘KAMPUS’;

Fakta	Fakta Baru

hari = ‘senin’
baba_kuliah = true;
	sylvia_kuliah = true;
	mhs_UNIKOM_kuliah = true;
	kuliah_libur = false;
	baba_berada = ‘KAMPUS’;

Sampai disini, fakta – fakta yang ada sudah tidak bisa digunakan untuk memicu *rule – rule* lagi sehingga proses dihentikan dan diperoleh kesimpulan – kesimpulan berupa fakta – fakta yang bernilai benar. Maka dari itu dapat disimpulkan apakah baba berada di kampus adalah benar dengan fakta yaitu baba_berada = ‘KAMPUS’ dan sylvia tidak kuliah tidak terbukti dengan fakta yang didapat yaitu sylvia_kuliah = true;

2.5.6. *Searching*

Searching merupakan mekanisme pemecahan masalah yang paling umum di dalam kecerdasan buatan. Metode *searching* pada kecerdasan buatan merupakan *searching* tehadap jalur problem *space*, bukan pada data tertentu.[2]

Contoh penerapan algoritma *searching* dapat dilihat pada penentuan rute, navigasi robot, perencanaan pada *block world*, *eight-queen problem* pada catur,

eight puzzle, dan berbagai macam contoh lainnya. Semua permasalahan ini memiliki struktur permasalahan yang serupa, diantaranya :[2]

1. Berhadapan dengan situasi awal dan ada *goal* tertentu yang hendak dicapai
2. Pada kondisi tertentu, ada pilihan dari tindak sederhana untuk dipilih, contohnya belok ke kiri atau belok ke kanan. Melakukan urutan tindakan tertentu dapat membuat goal tercapai atau juga tidak tercapai. Oleh karena itu, urutan harus terdefinisi dengan jelas.
3. *Searching* adalah proses meninjau beberapa urutan langkah dan memilih salah satu yang akan mencapai *goal*.
4. Untuk beberapa aplikasi, setiap urutan tindakan terkait dengan nilai tertentu. Permasalahan *searching* tidak hanya untuk mencapai *goal*, tetapi juga nilai minimum dalam pencapaian *goal* tersebut.

Secara umum algoritma *searching* di dalam kecerdasan buatan yang dikenal secara umum adalah :

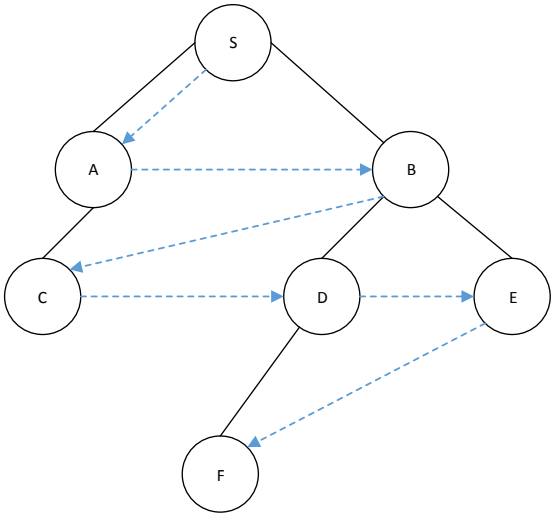
1. Uninformed Search Algorithm

Algoritma yang tidak memberikan informasi atau pengetahuan mengenai permasalahan yang ada. Isinya hanya sebatas definisi dari algoritma tersebut.

Algoritma ini menggambarkan bahwa teknik pencarian tidak memiliki informasi atau pengetahuan tambahan mengenai kondisi luar dari yang disediakan oleh definisi masalah. Algoritma ini akan melakukan *generate* dari *successor* dan membedakan *goal state* dari *non-goal state*. Pencarian dilakukan berdasarkan urutan node yang hendak di – *expand*.

a. Breadth First Search (BFS)

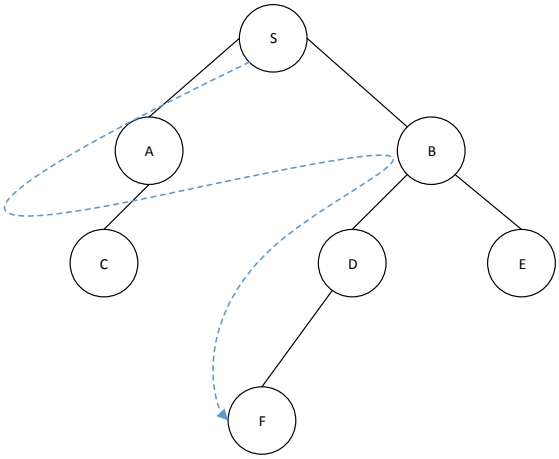
Pencarian yang menggunakan teknik dimana langkah pertamanya adalah *root node* diekspansi. Setelah dilanjut semua *successor* dari *root node* juga di *expand*. Hal ini terus berulang – ulang hingga *leaf* (*node* pada level bawah sidah tidak mempunyai *successor* lagi).



Gambar 2.4 Breadth First Search

b. Depth First Search (DFS)

Teknik pencarian yang melakukan ekspansi menuju *node* yang paling dalam pada *tree*. *Node* paling dalam dicirikan dengan tidak adanya *successor* dari *node* itu. Setelah selesai diekspansi, maka *node* tersebut akan ditinggalkan dan dilakukan ke *node* paling dalam lainnya yang masih memiliki *successor* yang belum diekspansi.

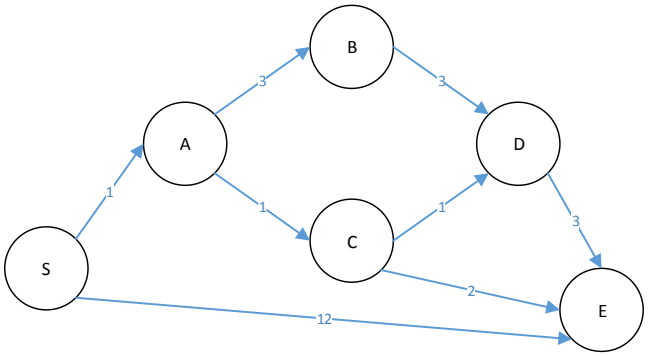


Gambar 2.5 Depth First Search

c. Uniform Cost Search (UCS)

Pencarian dengan *Breadth First Search* menjadi optimal ketika nilai pada semua *path* adalah sama. Dengan sedikit perluasan, dapat ditemukan sebuah algoritma yang optimal dengan melihat kepada nilai tiap *path* di antara *node – node* yang ada.

Selain menjalankan fungsi BFS, *Uniform Cost Search (UCS)* melakukan ekspansi *node* dengan nilai *path* yang paling kecil. Hal ini dapat dilakukan dengan membuat antrian pada *successor* yang ada berdasarkan kepada nilai *path*-nya (*node* disimpan dalam bentuk *priority queue*).



Gambar 2.6 Uniform Cost Search

2. Informed Search Algorithm

Walaupun menggunakan *Uninformed Search Algorithm* banyak masalah dapat dipecahkan, namun tidak semua algortima tersebut dapat menyelesaikan masalah dengan efisien. Oleh karena itu, dibutuhkan informasi atau pengetahuan tambahan ke dalam permasalahan yang ada.

Pencarian algoritma ini menggunakan *knowledge* yang spesifik kepada permasalahan yang dihadapi di samping definisi masalah itu sendiri. Metode ini mampu menemukan solusi secara efisien daripada yang dapat dilakukan pada metode *uninformed strategy*. Lebih lanjut lagi, dalam pencarian ini akan dikenal nilai estimasi (prediksi) dari satu *node* ke *node* yang lainnya. Nilai estimasi ini biasanya dilambangkan dengan h(n). Jika n adalah *goal node*, maka nilai h(n) adalah nol.[2]

a. *Greedy Best First Search*

Pencarian yang melakukan ekspansi *node* yang memiliki jarak terdekat dengan *goal*. Namun, ekspansi yang dilakukan pada metode ini menggunakan evaluasi *node* hanya dengan melihat kepada fungsi heuristiknya. Dengan kata lain, yang dibandingkan untuk penentuan ekspansi *node* adalah nilai estimasi/prediksinya saja.[2, p.27]

f(n) = h(n)

b. *A* Search*

Bentuk dari *Best First Search* yang paling dikenal adalah algoritma pencarian A* (dibaca dengan "*A-star*"). Sedikit berbeda dengan *Greedy* yang hanya melihat kepada nilai h(n), pencarian dengan A* melihat

kepada kombinasi nilai dari *path*-nya yaitu g(n) dengan nilai estimasi yaitu h(n).

f(n) = g(n) + h(n)

2.5.7. Teori Probabilitas Bayesian

Teori Probabilitas Bayesian ditemukan oleh Reverend Thomas Bayes (1701 – 1761). Pada umumnya, teori ini digunakan untuk menghitung nilai kebenaran probabilitas dari suatu *evidence* (fakta).[2, p. 86] Teori ini kemudian disempurnakan oleh Laplace.

Teori ini menerangkan hubungan antara probabilitas terjadinya peristiwa A dengan syarat peristiwa B telah terjadi dan probabilitas terjadinya peristiwa B dengan syarat peristiwa A telah terjadi.[2] Teori ini didasarkan pada prinsip bahwa jika terdapat tambahan informasi atau *evidence* (fakta), maka nilai probabilitas dapat diperbaiki. Teori ini bermanfaat untuk mengubah atau memperbaiki nilai kemungkinan yang ada menjadi lebih baik dengan didukung informasi atau *evidence* (fakta) tambahan.[2]

Misalkan $\{H_1, H_2, H_3, \ldots, H_n\}$ suatu himpunan kejadian yang merupakan suatu sekatan ruang *sample* S dengan $P(H_i) \neq 0$ untuk i = 1, 2, 3, ... , n. Dan misalkan E suatu kejadian sembarang dalam S dengan $P(E) \neq 0$.

$$P(H_i|E) = \frac{P(H_i \cap E)}{\sum_{i=1}^{n} P(H_i \cap E)} = \frac{P(E|H_i)P(H_i)}{\sum_{i=1}^{n} P(E|H_i)P(H_i)}$$

Bukti :

Berdasarkan probabilitas bersyarat :

$$P(H_i|E) = \frac{P(H_i \cap E)}{P(E)} = \frac{P(H_i \cap E)}{P(H_1 \cap E) + P(H_2 \cap E) + \cdots + P(H_n \cap E)} = \frac{P(E|H_i)P(H_i)}{\sum_{i=1}^{n} P(H_i \cap E)}$$

$$= \frac{P(E|H_i)P(H_i)}{\sum_{i=1}^{n} P(E|H_i)P(H_i)}$$

Contoh :

Chandra mengalami gejala ada bintik – bintik di wajahnya. Dokter menduga bahwa Chandra terkena cacar dengan :

1. Probabilitas munculnya bintik – bintik di wajah, jika Chandra terkena cacar : p(bintik|cacar) = 0.8
2. Probabilitas Chandra terkena cacar tanpa memandang gejala apapun : p(cacar) = 0.4
3. Probabilitas munculnya bintik – bintik di wajah, jika Chandra terkena alergi : p(bintik|alergi) = 0.3
4. Probabilitas Chandra terkena alergi tanpa memandang gejala apapun : p(alergi) = 0.7
5. Probabilitas munculnya bintik – bintik di wajah, jika Chandra jerawatan : p(bintik|jerawat) = 0.9
6. Probabilitas Chandra jerawatan tanpa memandang gejala apapun p(jerawat) : 0.5

Dari formulasi data yang diketahui di atas, maka dapat dihitung :

1. Probabilitas Chandra terkena cacar karena ada bintik – bintik di wajah :

$$P(H_i|E) = \frac{P(E|H_i)P(H_i)}{\sum_{i=1}^{n} P(E|H_i)P(H_i)}$$

$$P(cacar|bintik) = \frac{P(bintik|cacar) * P(cacar)}{P(bintik|cacar) * P(cacar) + p(bintik|alergi) * P(alergi) + P(bintik|jerawat) * p(jerawat)}$$

$$P(cacar|bintik) = \frac{0.8 * 0.4}{0.8 * 0.4 + 0.3 * 0.7 + 0.9 * 0.5}$$

$$P(cacar|bintik) = \frac{0.32}{0.98} = 0.327$$

2. Probabilitas Chandra terkena alergi karena ada bintik – bintik di wajah :

$$P(H_i|E) = \frac{P(E|H_i)P(H_i)}{\sum_{i=1}^{n} P(E|H_i)P(H_i)}$$

$P(alergi|bintik)$

$$= \frac{P(bintik|alergi) * P(alergi)}{P(bintik|cacar) * P(cacar) + p(bintik|alergi) * P(alergi) + P(bintik|jerawat) * p(jerawat)}$$

$$P(alergi|bintik) = \frac{0.3 * 0.7}{0.8 * 0.4 + 0.3 * 0.7 + 0.9 * 0.5}$$

$$P(alergi|bintik) = \frac{0.21}{0.98} = 0.214$$

3. Probabilitas Chandra terkena jerawat karena ada bintik – bintik di wajah

$$P(H_i|E) = \frac{P(E|H_i)P(H_i)}{\sum_{i=1}^{n} P(E|H_i)P(H_i)}$$

$P(jerawat|bintik)$

$$= \frac{P(bintik|jerawat) * P(jerawat)}{P(bintik|cacar) * P(cacar) + p(bintik|alergi) * P(alergi) + P(bintik|jerawat) * p(jerawat)}$$

$$P(jerawat|bintik) = \frac{0.9 * 0.5}{0.8 * 0.4 + 0.3 * 0.7 + 0.9 * 0.5}$$

$$P(jerawat|bintik) = \frac{0.45}{0.98} = 0.459$$

Jika terdapat gejala baru, maka persamaan Bayes berubah menjadi :

$$P(H|E,e) = P(H|E)\frac{P(e|E,H)}{P(e|E)}$$

Contoh :

Adanya bintik – bintik di wajah merupakan gejala seseorang terkena cacar. Namun, ada observasi baru yang menunjukan bahwa selain bintik – bintik di wajah,

panas badan juga merupakan gejala orang terkena cacar. Dapat disimpulkan antara munculnya bintik – bintik di wajah dan panas badan juga memiliki keterkaitan satu sama lain.

Pada wajah Chandra terdapat bintik – bintik. Dokter menduga bahwa Chandra terkena cacar dengan probablitas bahwa Chandra terkena cacar bila ada bintik – bintik di wajah P(cacar|bintik) = 0.8. kemudian, ada observasi yang lain bahwa ada observasi yang lain bahwa apabila seseorang terkena cacar maka dia pasti mengalami panas badan. Diketahui probabilitas orang yang terkena cacar bila panas badan P(cacar|panas) = 0.5.

Keterkaitan antara adanya bintik – bintik di wajah dengan panas badan bila seseorang terkena cacar P(bintik|panas, cacar) = 0.4. sedangkan nilai keterkaitan antara adanya bintik – bintik di wajah dan panas badan P(bintik|panas) = 0.6

Maka dapat ditarik kesimpulan :

$$P(H|E,e) = P(H|E)\frac{P(e|E,H)}{P(e|E)}$$

$$P(cacar|panas, bintik) = P(cacar|bintik)\frac{P(bintik|panas, cacar)}{P(bintik|panas)}$$

$$P(cacar|panas, bintik) = 0.5\frac{0.4}{0.6} = 0.33$$

2.6. Perangkat Lunak Pendukung

Perangkat lunak pendukung merupakan perangkat lunak yang digunakan dalam merancang sistem dari mulai pengkodingan sampai implementasinya, berikut ini adalah perangkat lunak yang akan digunakan dalam merancang sistem pakar diagnosis kerusakan kendaraan :

2.6.1. *HTML*

HTML (Hypertext Markup Language) yaitu bahasa *scripting* yang memberikan *output* berupa halaman *Web Site* sehingga halaman tersebut dapat diakses di setiap komputer pengakses. Pada *Web page*, *HTML* menjadi bahasa *script* dasar yang dijalankan bersama bahasa *scripting* lainnya.[8]

HTML merupakan *script* pemrograman yang memberikan informasi di internet dan informasi itu membawa kita menjelajah dari satu tempat ke tempat lain.[9]

Dapat disimpulkan bahwa *HTML* merupakan sebuah bahasa untuk menampilkan konten *web* bagi pengaksesnya untuk mendapatkan informasi. Semua bahasa *scripting* yang berjalan di bawah *Web* dapat didukung oleh *HTML*, biasanya bahasa tersebut melakukan *embeded script* pada tag *HTML*. *HTML* juga dapat ditulis menggunakan *editor* yang kita sukai seperti *Macromedia Dreamwever, Sublime Text Editor*, dan bahkan menggunakan *NotePad* sekalipun.

HTML mempunyai struktur dasar dimana elemen *HTML* dimulai dengan tag awalan dilanjutkan dengan isinya dan tag untuk akhiran seperti : <body> ini HTML </body>. Sebuah elemen standar dapat bersarang pada elemen lainnya. Dokumen *HTML* ditandai dengan tag pembuka <html>, dan diakhiri dengan tag penutup </html>. Biasanya juga memuat tag <head> dan <body>.

2.6.2. *CSS*

CSS merupakan singkatan dari *Cascading Style Sheets*. Fungsinya untuk mendefinisikan elemen-elemen *HTML* untuk ditampilkan di halaman *Web*. *CSS* juga bisa diartikan sebagai pemberian desain pada halaman *Web*.[10]

2.6.3. *INK*

INK adalah *front-end framework* yang baik, bagus dan luar biasa yang mengedepankan tampilan yang ramah untuk *mobile device* (*Handphone, smartphone* dll.) guna mempercepat dan mempermudah pengembangan suatu *design website*. INK menyediakan kumpulan *setting layout* dan fungsi *javascript* yang siap pakai dan mudah untuk dikembangkan.

2.6.4. *Javascript*

Javascript merupakan pemograman java yang perintah - perintahnya ditulis dengan kode yang disebut *script*, *java* adalah bahasa pemrograman berorientasi objek, sedangkan *script* adalah serangkaian instruksi program.[10]

Jadi *javascript* merupakan *interface* pembantu dalam pemrograman web, *javascript* mengenali kode secara *case sensitive*, yang artinya javascript membedakan huruf besar dan huruf kecil.

2.6.5. JQuery

JQuery adalah yang tercepat, terkecil dan kaya akan fitur *library* javascript. Dengan jquery membuat hal-hal seperti *html* dokumen dan manipulasi, penanganan *event*, animasi, dan Ajax jauh lebih sederhana dengan *API* yang mudah digunakan yang bekerja di banyak *browser*.

Dengan JQuery, *developer* dapat mudah dalam penulisan kode javascript untuk penanganan aksi manipulasi dokumen *HTML* agar lebih interaktif.

2.6.6. *PHP*

PHP (PHP Hypertext Preprocessor) adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam *HTML*, banyak dipakai untuk membuat situs web dinamis.[11]

Jadi *PHP* merupakan bahasa yang membuat sebuah *web* dapat terkoneksi dengan database sehingga menjadi *web* yang dinamis, yang dapat berubah- ubah sesuai data yang tersimpan pada database.

2.6.7. *CodeIgniter*

CodeIgniter adalah *framework PHP* yang kuat dengan *size source* yang sangat kecil, dibangun untuk pengembang yang membutuhkan *toolkit* sederhana dan elegan untuk membuat aplikasi *web* dengan fitur lengkap.

Dengan *framework codeigniter* dapat memudahkan *developer* dalam membuat aplikasi *web* dinamis dengan cepat dan rapi karena fitur yang disediakan oleh *framework codeigniter* banyak dan mudah.

2.6.8. MySQL

MySQL merupakan sebuah bentuk database yang berjalan sebagai *server* artinya tidak harus meletakan database tersebut dalam satu mesin dengan aplikasi yang digunakan, sehingga anda dapat meletakan sebuah database pada sebuah mesin khusus dan dapat diletakan pada tempat yang jauh dari komputer pengaksesnya.[11]

MySQL memiliki beberapa kelebihan, di antara lain:

a. Portability

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti windows, Linux, FreeBSD, Solaris dan lain-lain.

b. Open source

MySQL didistribusikan secara *open source* (gratis), dibawah lisensi *GPL* sehingga dapat digunakan secara cuma - cuma.

c. Multi user

MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

d. Performance tuning

MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

e. Column types

MySQL memiliki tipe kolom yang sangat kompleks, seperti *integer*, *double*, *char*, *text*, *date* dan lain-lain.

f. Command and function

MySQL memiliki *operator* dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam *query*.

2.6.9. *Browser* dan *Web Server*

Web Server merupakan perangkat lunak yang dijalankan di sistem operasi pada komputer *server* maupun desktop, yang berfungsi untuk menerima permintaan (*request*) dalam bentuk *protocol*, misalkan *HTTP (Hyper Text Transfer Protocol)* dan *HTTPS (Hyper Text Transfer Protocol Secure)*.[3]

Berdasarkan penjelasan di atas, *web server* merupakan perangkat lunak untuk menerima permintaan pada klien yang biasanya dikenal dengan nama *web browser*.

2.6.10. XAMPP

XAMPP adalah aplikasi *web server* bersifat *instant* (siap saji) yang dapat digunakan baik di sistem operasi linux maupun di sistem operasi windows.[3] XAMPP dapat mempermudah pengguna karena dengan menginstal XAMPP kita tidak perlu lagi menginstal dan mengkonfigurasi Apache *server* ataupun MySQL database.

2.6.11. *UML (Unified Modeling Language)*

Unified Modeling Language (UML) memiliki beberapa pengertian diantaranya " adalah keluarga notasi grafis yang didukung oleh model-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi *object*" (Fowler, 2005).[12]

UML terdiri dari dua komponen utama yaitu bentuk analisis dan *design*. Analisis UML terdiri dari *Use Case Diagram, Class Diagram, Sequence Diagram, Collaboration Diagram, Activity Diagram* dan *Statechart Diagram*. *Design UML* terdiri dari *Component Diagram* dan *Deployment Diagram*.

2.7. Jaringan Komputer

Jaringan komputer adalah hubungan dari sejumlah perangkat yang dapat saling berkomunikasi satu sama lain (*a network is a interconnection of a set of devices capable of communication*).[14]

Jadi, dapat dibayangkan bahwa jaringan komputer dapat terjadi jika ada dua perangkat komputer atau lebih yang saling terhubung. Perangkat di sini dapat berupa komputer desktop, smartphone, laptop, dan PC tablet.

2.7.1. *LAN (Local Area Network)*

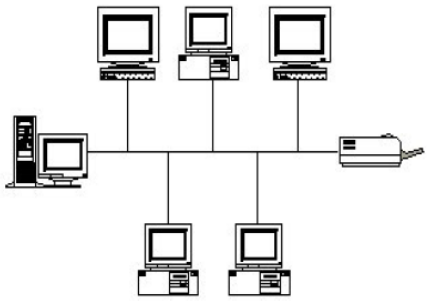
LAN (Local Area Network) merupakan jaringan komputer terkecil untuk pemakaian pribadi.[5] Biasanya jaringan *LAN* dapat mencakup 1 km hingga 10 km. Jaringan ini cocok diimplementasikan pada komputer kampus, gedung, kantor, dalam rumah, sekolah, atau yang lebih kecil.

2.7.2. Topologi Jaringan Komputer

Menurut Iwan Sofana topologi fisik jaringan yang digunakan dalam jaringan diantaranya:

1. *Linear Bus* (Garis Lurus)

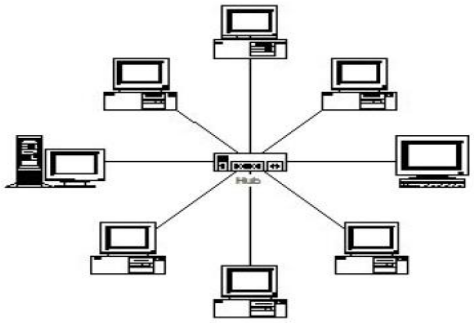
   Topologi linear bus terdiri dari satu jalur kabel utama dimana masing-masing ujungnya diberikan sebuah *terminator*. Semua *nodes* pada jaringan (*file server*, *workstation*, dan perangkat lainnya) terkoneksi sebuah kabel utama (*backbone*).



Gambar 2.7 Topologi Linear Bus
(Sumber : Teori dan modul Jaringan Komputer[14, p. 12])

2. *Star* (Bintang)

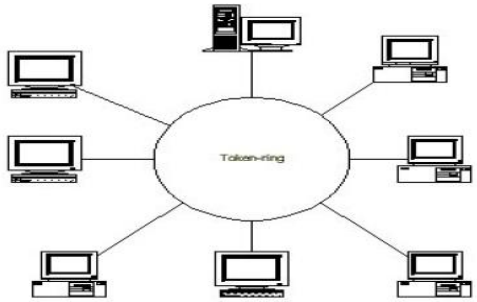
Topologi *star*, setiap *nodes* (*file server*, *workstation*, dan perangkat lainnya) terkoneksi ke jaringan melalui sebuah *concentrator*.



Gambar 2.8 Topologi Star
(Sumber : Teori dan modul Jaringan Komputer[14, p. 12,13])

3. *Ring* (Cincin)

   Topologi *ring* menggunakan teknik konfigurasi yang sama dengan topologi *star* tetapi pada topologi ini terlihat bahwa jalur media transmisi menyerupai suatu lingkaran tertutup.



Gambar 2.9 Topologi Ring
(Sumber : Teori dan modul Jaringan Komputer[14, p. 12])

4. *Tree* (Pohon)

   Topologi model ini merupakan perpaduan antara topologi *linear bus* dan *star*, yang terdiri dari kelompok-kelompok *workstation* dengan konfigurasi *star* yang terkoneksi ke kabel utama yang menggunakan topologi *linear bus*. Topologi ini memungkinkan untuk perkembangan

jaringan yang telah ada dan memungkinkan untuk mengkonfigurasi jaringan sesuai dengan kebutuhan.