

BAB 2

LANDASAN TEORI

2.1 Intelegensia Semu

Intelegensia semu atau kecerdasan buatan merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. 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.

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki oleh seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik.

Agar komputer bisa bertindak seperti dan sebaik manusia, maka komputer juga harus diberi bekal pengetahuan, dan kemampuan untuk menalar. Untuk itu pada intelligenza semu, akan mencoba untuk memberikan beberapa metoda untuk membekali komputer dengan kedua komponen tersebut agar komputer bisa menjadi mesin yang pintar.

Lebih detilnya, 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 yang sering dibahas oleh para peneliti meliputi:

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 integrasi, pembuktian).

c. *Expert task.*

- Analisis finansial.
 - Analisis *medical*.
 - 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*).

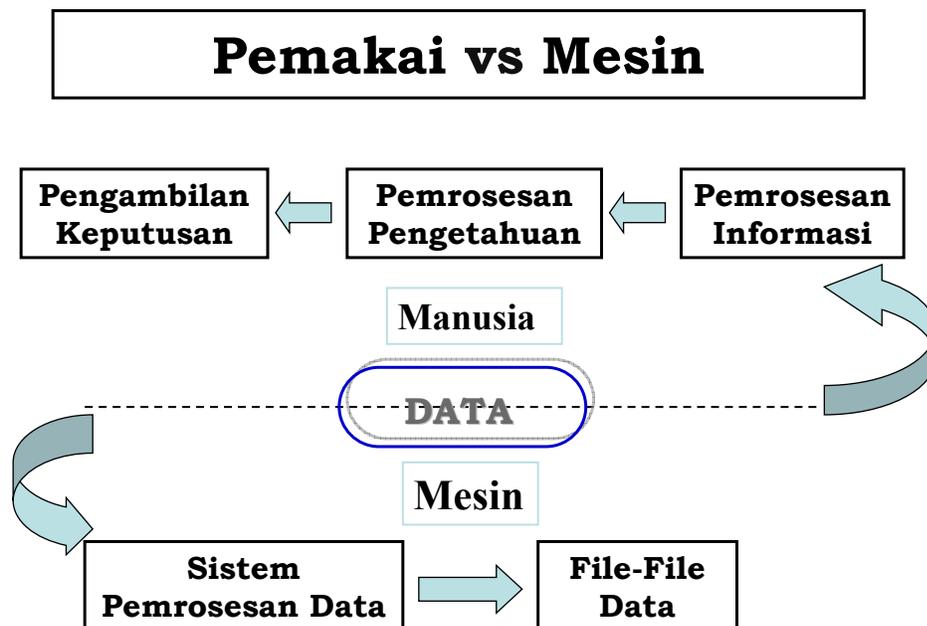
Untuk melakukan aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan, yaitu:

- **Basis Pengetahuan (*Knowledge base*)**, berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
- **Motor Inferensi (*Inference Engine*)**, yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.

2.1.1 Tujuan inteligensia Semu

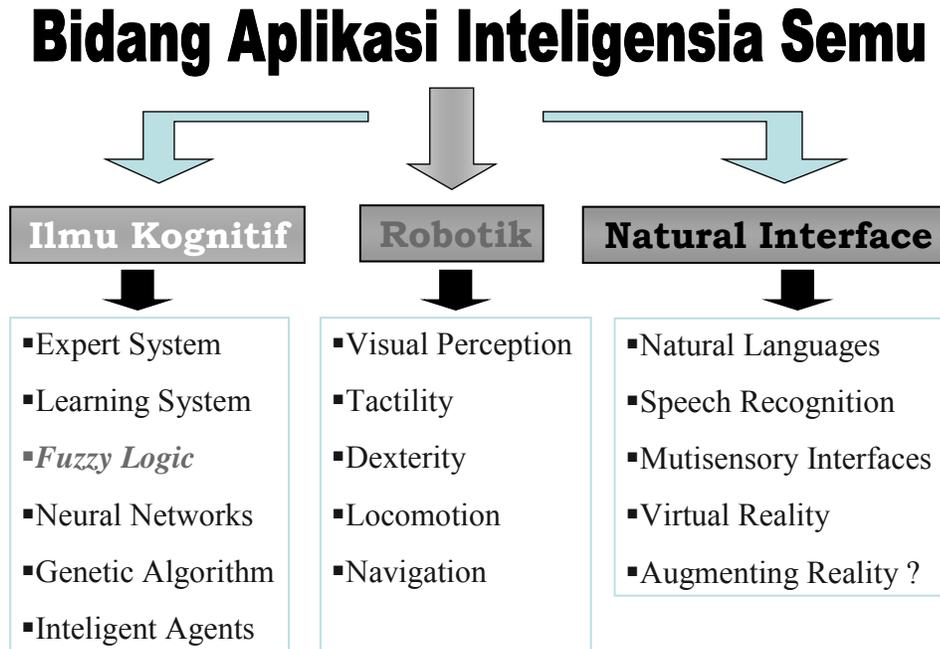
- Membuat mesin lebih pintar (tujuan utama)
- Mengerti apa itu intelijen (tujuan pemenang Noble)
- Membuat mesin lebih berguna (tujuan usahawan)

- Sudut pandang Kecerdasan
Membuat mesin lebih “Cerdas”
- Sudut pandang Penelitian
Membuat mesin (komputer) dapat melakukan sesuatu sebaik yang dilakukan manusia
- Sudut pandang Bisnis
Sekumpulan peralatan yang sangat *powerfull* dan metodologis untuk menyelesaikan masalah bisnis
- Sudut pandang Pemrograman
Studi tentang pemrograman simbolik, *problem solving* dan *searching*
(sumber: Kusumadewi, Sri)



Gambar 2.1 - Pemakai vs Mesin

2.1.2 Bidang Aplikasi Inteligencia Semu



Gambar 2.2 - Bidang aplikasi inteligensia semu

2.1.3 Kecerdasan Buatan dan Kecerdasan Alami

Jika dibandingkan dengan kecerdasan alami (kecerdasan yang dimiliki oleh manusia), kecerdasan buatan memiliki beberapa keuntungan secara komersial antara lain:

- Kecerdasan buatan lebih bersifat permanen. Kecerdasan alami akan cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. Kecerdasan buatan tidak akan berubah sepanjang sistem komputer & program tidak mengubahnya.
- Kecerdasan buatan lebih mudah diduplikasi & disebar. Mentransfer pengetahuan manusia dari satu orang ke orang lain

membutuhkan proses yang sangat lama; dan juga dsuatu keahlian itu tidak akan pernah dapat diduplikasi degan lengkap. Oleh karena itu, jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.

- c. Kecerdasan buatan lebih murah dibanding kecerdasan alami. Menyediakan layanan komputer akan lebih mudah & lebih murah dibandingkan dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
- d. Kecerdasan buatan bersifat konsisten. Hal ini disebabkan karena kecerdasan buatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami akan senantiasa berubah-ubah.
- e. Kecerdasan buatan dapat didokumentasi. Keputusan yang dibuat oleh komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.
- f. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibanding dengan kecerdasan alami.
- g. Kecerdasan buatan dapat mengerjakan pekerjaan lebih baik dibanding dengan kecerdasan alami.

Sedangkan keuntungan dari kecerdasan alami adalah:

- a. Kreatif. Kemampuan untuk menambah ataupun memnuhi pengetahuan itu sanagat melekat pada jiwa manusia. Pada kecerdasan buatan, untuk menambah pengethuan harus dilakukan melalui sistem yang dibangun.
- b. Kecerdasan alami memungkinkan orang untuk menggunakan pengalaman secara langsung. Sedangkan pada kecerdasan buatan harus bekerja secara input-input simbolik.
- c. Pemikiran manusia dapat digunaka secara luas, sedangkan kecerdasan buatan sangat terbatas.

2.1.4 Sejarah Kecerdasan Buatan

Kecerdasan buatan termasuk bidang ilmu yang relatif muda. Pada tahun 1950-an para ilmuwan dan peneliti mulai memikirkan bagaimana caranya agar mesin dapat melakukan pekerjaannya seperti yang bisa dikerjakan oleh manusia. *Alan Turing*, seorang matematikawan dari *Inggris* pertama kali mengusulkan adanya tes untuk melihat bisa tidaknya sebuah mesin dikatakan cerdas. Hasil tes tersebut kemudian dikenal dengan *Turing Test*, dimana si mesin tersebut menyamar seolah-olah sebagai seseorang di dalam suatu permainan yang mampu memberikan respon terhadap serangkaian pertanyaan yang diajukan. *Turing* beranggapan bahwa, jika mesin dapat membuat seseorang percaya bahwa dirinya mampu

berkomunikasi dengan orang lain, maka dapat dikatakan bahwa mesin tersebut cerdas (seperti layaknya manusia).

Kecerdasan Buatan atau "*Artificial Intelligence*" itu sendiri dimunculkan oleh seorang profesor dari *Massachusetts Institute of Technology* yang bernama *John McCarthy* pada tahun 1956 pada *Dartmouth Conference* yang dihadiri oleh para peneliti AI. Pada konferensi tersebut juga didefinisikan tujuan utama dari kecerdasan buatan, yaitu: mengetahui dan memodelkan proses-proses berfikir manusia dan mendesain mesin agar dapat menirukan kelakuan manusia tersebut.

Beberapa program AI yang mulai dibuat pada tahun 1956-1966, antara lain:

- *Logic Theorist*, diperkenalkan pada *Dartmouth Conference*, program ini dapat membuktikan teorema-teorema matematika.
- *Sad Sam*, diprogram oleh *Robert K Lindsay* (1960). Program ini dapat mengetahui kalimat-kalimat sederhana yang ditulis dalam bahasa Inggris dan mampu memberikan jawaban dari fakta-fakta yang didengar dalam sebuah percakapan.
- *ELIZA*, diprogram oleh *Joseph Weizenbaum* (1967). Program ini mampu melakukan terapi terhadap pasien dengan memberikan beberapa pertanyaan.

2.1.5 Ruang Lingkup Inteligencia Semu

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.

Dewasa ini, kecerdasan buatan juga memberikan kontribusi yang cukup besar di bidang manajemen. Adanya sistem pendukung keputusan, dan Sistem Informasi Manajemen juga tidak terlepas dari andil kecerdasan buatan.

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

Lingkup utama kecerdasan buatan adalah:

1. Sistem Pakar (*Expert System*).

Di sini 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 obyek-obyek tampak melalui komputer.

6. *Intelligent Computer-aided Instruction*.

Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.

7. *Game Playing*.

Beberapa karakteristik yang ada pada sistem yang menggunakan *artificial intelligence* adalah pemrogramannya yang cenderung bersifat simbolik ketimbang algoritmik, bisa mengakomodasi input yang tidak lengkap, bisa melakukan interferensi, dan adanya pemisahan antara kontrol dengan pengetahuan.

Namun, seiring dengan perkembangan teknologi, muncul beberapa teknologi yang juga bertujuan untuk membuat agar komputer menjadi cerdas sehingga dapat menirukan kerja manusia sehari-hari.

Teknologi ini juga mampu mengakomodasi adanya ketidakpastian dan ketidaktepatan data input. Dengan didasari pada teori himpunan, maka pada tahun 1965 muncul Logika *Fuzzy*. Kemudian pada tahun 1975 *John Holland* mengatakan bahwa setiap problem berbentuk adaptasi (alami maupun buatan) secara umum dapat diformulasikan dalam terminologi genetika. Algoritma genetika ini merupakan simulasi proses evolusi Darwin dan operasi genetika atas kromosom.

2.2 Perkembangan Model AI Pada Industri Games

Pada dasarnya AI game terdiri dari dua katagori yaitu *Deterministic* dan *Nondeterministic*. *Deterministic* merupakan simple algoritma, dimana sebuah nonplayer-character pada komputer game bergerak sesuai dengan koordinat yang telah diatur sedemikian rupa, seakan-akan sebuah *nonplayer-character* dapat bergerak sesuai kemauannya sendiri, tetapi mudah ditebak apa yang akan dilakukannya selajutnya. Sedangkan *nondeterministic* adalah kebalikan dari

deterministic. Pergerakan *nondeterministik* tidak dapat di prediksi oleh pemain dan menggunakan algoritma yang lebih kompleks. Sebagai contoh dari *nondeterministik*, dinamika nonplayer-character dapat belajar beradaptasi lingkungan dan juga untuk mengatur strategi taktik untuk bertarung dengan pemain, dan memecahkan masalah yang dihadapi.

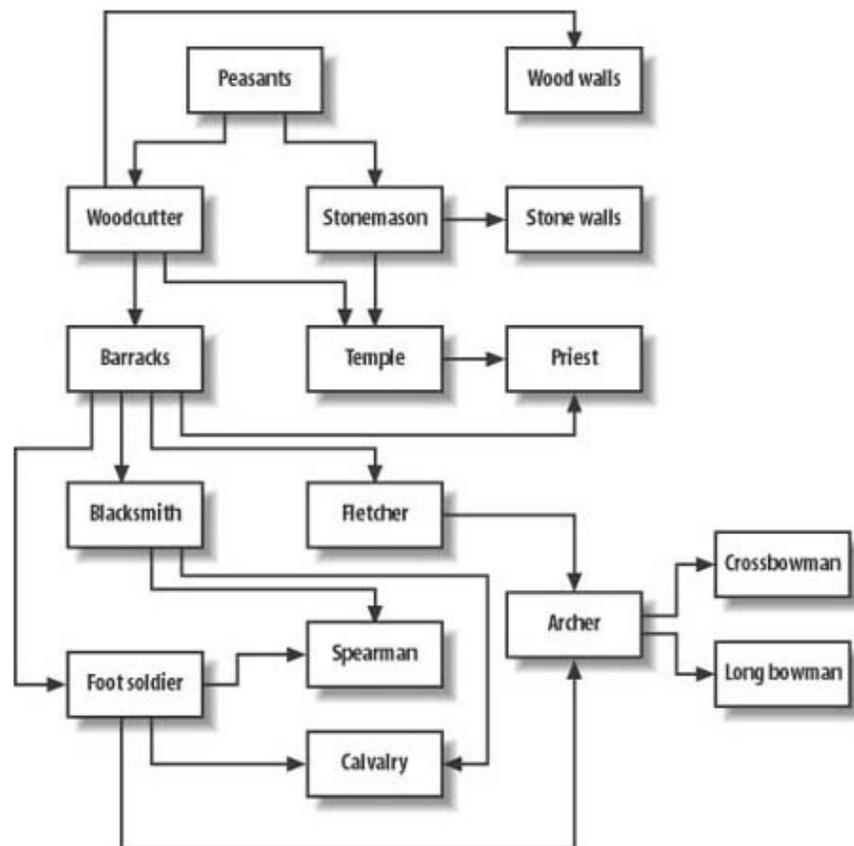
Pada point ini akan kami akan menganalisa beberapa metode algoritma yang paling banyak dan pasti digunakan pada *video games*. Kami juga akan memberikan contoh penerapan pada games yang telah beredar di pasaran dengan menggunakan salah satu metode ini.

2.2.1 Rule-Based AI

Rule-Based AI merupakan algoritma AI yang paling luas dan paling banyak digunakan. Tidak hanya pada game dan aplikasi saja tetapi juga system ini telah diterapkan di dunia kedokteran. System ini berusaha mengadopsi pengetahuan manusia kedalam komputer, agar komputer dapat menyelesaikan masalah seperti biasa yang dilakukan oleh para ahli. Pada dasar algoritma ini hanya berisi perintah *if-then rules* yang digunakan untuk mendapatkan kesimpulan. Sebelum rule tersebut diterapkan pada aplikasi, kita membuat semua rule ada, lalu rule tersebutlah komputer dapat memberikan jawaban atau output kepada user. Tetapi hasil output yang diberikan komputer kepada user tidak 100% bernilai benar dan akurat.

Kebanyakan game jenis *RPG* atau *MMRPG* yang menggunakan rule-based AI, tetapi tidak menutup kemungkinan untuk dipakai pada jenis game lain, terutama pada *fighting game* juga menerapkan algoritma ini. Kami akan memberikan contoh bagaimana alur algoritma ini bekerja. Contoh yang kami buat merupakan jenis *Role-Playing Games*.

Di bawah contoh Struktur rule-based AI,



Gambar 2.3 - Tree pada Rule-Based AI

Berdasarkan gambar di atas maka didapat algoritma sebagai berikut :

```

if(Priest == Yes)
{
    Temple = Yes;
    Barracks = Yes;
    Woodcutter= Yes;
    Stonemason= Yes;
  
```

```

}

if(Longbowman == Yes)
{
    Archer = Yes;
    Fletcher = Yes;
    FootSoldier = Yes;
    Barracks = Yes;
    Woodcutter = Yes;
}

if(Cavalry == Yes)
{
    Blacksmith = Yes;
    FootSoldier = Yes;
    Barracks = Yes;
    Woodcutter = Yes;
}
if(StoneWalls == Yes)
    Stonemason = Yes;

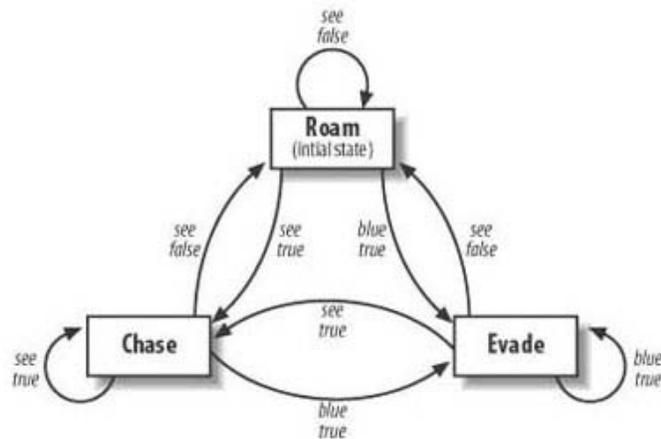
```

2.2.2 Finite State Machines

Final State Mechine merupakan simple algoritma yang paling umum digunakan oleh video games dari dulu hingga sekarang. Pada algoritma ini komputer hanya mengalami perubahan kondisi berdasarkan aksi dari player.

Sebagai contoh, “*Ghosts*” pada Pac Man yang menggunakan algoritma Finite State Machine. komputer dapat bergerak leluasa menjelajah, mengejar *player*, atau menghindari player. Di setiap state komputer dapat menunjukkan reaksi yang berbeda-beda. Keputusan yang komputer dapat berdasarkan aksi dari player. Contohnya jika *player* memakan pil “*power*” maka player dapat menghabisi komputer, maka komputer akan berubah dari pengejaran menjadi menghindari.

Dibawah terdapat contoh simpel diagram dan *algoritma Finale State Machine "Ghost"* pada game *Pac Man*



Gambar 2.4 - Diagram Ghost pada finite state machine

```

SWITCH(currentState){
  CASE Roam:
    IF (imBlue==TRUE) currentState=Evade;
    ELSE IF (canSeePlayer== TRUE) currentState=Chase;
    ELSE IF (canSeePlayer== FALSE) currentState=Roam;
    break;
  CASE Chase:
    IF (imBlue== TRUE) currentState=Evade;
    ELSE IF (canSeePlayer== FALSE) currentState=Roam;
    ELSE IF (canSeePlayer== TRUE) currentState=Chase;
    break;
  CASE Evade:
    IF (imBlue== TRUE) currentState=Evade;
    ELSE IF (canSeePlayer== TRUE) currentState=Chase;
    ELSE IF (canSeePlayer==FALSE) currentState=Roam;
    break;
}

```

2.2.3. Basic Probability

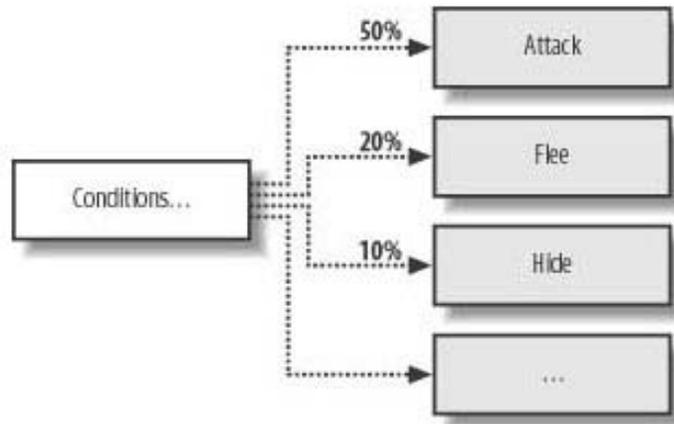
Algoritma ini menilai semua kemungkinan yang akan player lakukan, sebelum komputer mengambil keputusan. Sebelum algoritma ini dibuat, kita harus membuat sebuah nilai probabilitas pada setiap aksi yang akan player lakukan. komputer akan mencari nilai probabilitas terbesar atau

yang paling menguntungkan yang akan komputer putuskan komputer. Di bawah terdapat contoh tabel probabilitas dengan nilai masing-masing skill character.

Tabel 2.1 - Character class ability

Character class ability				
Ability	Wizard	Fighter	Rouge	Ranger
Use magic	0.9	0.05	0.2	0.1
Wield sword	0.1	0.9	0.7	0.75
Harvest wood	0.3	0.5	0.6	0.8
Pick locks	0.15	0.1	0.05	0.5
Find traps	0.13	0.05	0.2	0.7
Read map	0.4	0.2	0.1	0.8

Berdasarkan nilai di atas, sebagai contoh jika player adalah *wizard* dan komputer adalah *rouge*. *Player* menggunakan *ability magic* kemungkinan berhasilnya 90% ketimbang *rouge* yang hanya kemungkinan berhasilnya hanya 20%. Maka dari penilai tersebut komputer mencari probabilitas yang paling menguntungkan yaitu *wield sword* yang kemungkinan berhasilnya 70%.



Gambar 2.5 - Hirarki Probability game

Pada contoh diatas juga menggambarkan nilai probabilitas, komputer akan membaca kondisi yang terjadi pada player, lalu komputer akan mencari kemungkinan yang ada, dan memilih kemungkinan yang paling menguntungkan untuk melakukan tindakan.

Metode ini sangat bermanfaat pada games fighting, pada umumnya hampir semua game fighting menggunakan metode algoritma ini. Contoh game fighting yang menggunakan metode ini : Street Fighter, Tekken, dan lain-lain.

2.3 *Softcomputing*

Softcomputing merupakan metode yang dapat mengolah data-data yang bersifat tidak pasti, impresisi dan dapat diimplementasikan dengan biaya yang murah (low-cost solution). Berbagai macam definisi *softcomputing* diberikan oleh para ahli. Salah satu definisinya adalah sebagaimana disampaikan oleh pencetus

softcomputing yaitu Prof. Lotfi A. Zadeh sebagai berikut : “Berbeda dengan pendekatan konvensional *hardcomputing*, *softcomputing* dapat bekerja dengan baik walaupun terdapat ketidakpastian, ketidakakuratan maupun kebenaran parsial pada data yang diolah. Hal inilah yang melatarbelakangi fenomena dimana kebanyakan metode *softcomputing* mengambil *human-mind* sebagai model.”.

Mengapakah *human-mind* merupakan model yang menarik bagi pengembangan *softcomputing* ? Kunci dari pertanyaan ini sebenarnya terletak pada struktur dan fungsi dari otak manusia. Otak manusia merupakan mesin molekuler, yang terdiri dari dua jenis sel: neuron dan glia. Dalam otak kita terdapat sekitar 10^{11} sel neuron, sedangkan sel glia sekitar 3 sampai 4 kali lipatnya. Sel neuron berfungsi sebagai pemroses informasi yang diterima oleh otak. Sel neuron terhubung antara satu dengan yang lain dengan benang-benang panjang. Berat otak manusia saat lahir sekitar 400 gram, sedangkan saat dewasa sekitar 1500 gram. Pertambahan berat ini disebabkan oleh bertambah panjangnya benang-benang tersebut, disamping pertambahan sel glia. Pertambahan panjang ini berkaitan erat dengan proses pembelajaran yang dialami oleh manusia. Hal ini merupakan ide awal bagi pengembangan metode *softcomputing*: artificial neural network, yang memiliki kemampuan pembelajaran terhadap informasi yang telah diterima.

Selain kemampuan pembelajaran, otak manusia juga memiliki kemampuan untuk mengambil keputusan walaupun informasi mengandung unsur ketidakpastian dan kekurangtegasan, seperti “manis”, “pahit”, “tinggi”, “rendah”,

dsb. Hal ini merupakan konsep yang mendasari mengembangkan metode fuzzy, yang mencerminkan cara berfikir manusia. Selain neural network dan fuzzy, masih banyak lagi jenis-jenis metode *softcomputing*, yang ide awalnya bersumber dari otak manusia maupun mekanisme biologi yang terdapat di alam semesta.

2.3.1 Metode *Softcomputing*

Mengacu pada definisi yang diberikan oleh Zadeh, metode-metode dalam *softcomputing* dapat dikategorikan ke dalam tiga kategori besar:

- *Fuzzy Logic* (FL)
- *Neural Network Theory* (NN)
- *Probabilistic Reasoning* (PR)

Metode-metode ini sebenarnya bukanlah sesuatu yang baru diadakan setelah konsep *softcomputing* dirumuskan. Yang terjadi justru sebaliknya. Metode-metode Fuzzy Logic, Neural Network maupun Probabilistic Reasoning telah ada lebih dahulu. Fuzzy Logic telah berkembang sejak tahun 1965. Konsep-konsep dasar neural network telah digali sejak tahun 1940-an. Probabilistic Reasoning juga bukanlah hal yang baru sama sekali. Karena itu, Zadeh menyebut *softcomputing* sebagai reinkarnasi dari metode-metode di atas.

Lebih lanjut lagi, dalam konsep *softcomputing*, ketiga jenis metode ini ibarat pilar, saling mendukung dan bekerjasama dalam memecahkan suatu permasalahan. Keunggulan yang diperoleh dari kerjasama metode-metode itu lebih ditekankan daripada keunggulan individual salah satu daripadanya. Kekurangan satu metode akan ditutup dengan kelebihan metode yang lain.

Keunggulan satu metode disumbangkan, sehingga segi-segi positif dari metode yang ada tersebut dapat dimanfaatkan secara optimal.

Berikut diuraikan konsep dan gambaran mengenai masing-masing pilar dalam *softcomputing*.

2.3.1.1 Logika Fuzzy

Aplikasi-aplikasi yang menggunakan sistem logika fuzzy sering sekali dianggap atau dinamakan sebagai pengendali fuzzy (fuzzy control). Padahal disamping pengendali fuzzy terdapat bermacam-macam teori yang digunakan pada aplikasi-aplikasi fuzzy seperti klasifikasi fuzzy (fuzzy clasification) dan diagnosis fuzzy (fuzzy diagnosis). Pada tulisan ini akan dipaparkan masalah dalam teknologi fuzzy dan perbedaan antara pengendali fuzzy dengan klasifikasi fuzzy dan fuzzy diagnosis.

Teknologi sistem fuzzy telah berkembang cukup jauh, dan memberikan berbagai keuntungan dan perbaikan unjuk kerja pada sistem kendali yang pernah ada. Perkembangan sistem ini menuntut mutu sumber daya manusia yang berpendidikan, seperti ilmuwan dan enjineer yang ahli di bidang teknik sistem fuzzy dan tidak lepas dari kondisi pendidikan dan kelengkapan alat-alat yang mendukung pengembangan teknologi ini.

Di bidang aplikasi fuzzy, tidak hanya terdapat sistem kendali fuzzy, melainkan juga ada klasifikasi fuzzy dan diagnosis fuzzy. Jadi sistem fuzzy pada keadaan riil, mempunyai ragam metode dan strategi

pengembangan yang dapat diterapkan pada masalah-masalah kendali saat ini.

Cara berekspresi mesin

- Semua ekspresi mesin (komputer) berbasiskan bilangan binari 0 dan 1. Untuk ekspresi bilangan maka dibuat kumpulan 0 dan 1 yang disebut dengan *binary word*.
- Dapat melakukan proses logika “JIKA MAKA.... ‘ yang merupakan satu kalimat. Kumpulan kalimat ini dikenal dengan “*Knowledge base*”

Problema ekspresi mesin

- Jika tidak ditemukan logika yang tepat benar (data matching) maka tidak akan keluar jawaban.
- Hanya dapat memproses informasi yang sudah jelas batasannya yang dapat diekspresikan dengan binary word.
- Tidak dapat memasukan beberapa aturan inklusi (aturan IF.. THEN..) yang tidak konsisten ke dalam “knowledge base”

Fuzzy merupakan representasi Suatu pengetahuan yang dikonstruksikan dengan *if-then rules*. Karakteristik dari metode ini adalah :

- Pemecahan masalah dilakukan dengan menjelaskan sistem bukan lewat angka-angka, melainkan secara linguistik, atau variable-variable yang mengandung ketakpastian/ketidaktegasan.

- Pemakaian if-then rules untuk menjelaskan kaitan antara satu variable dengan yang lain.
- Menjelaskan sistem memakai algoritma fuzzy

2.3.1.1.1 Sejarah *Fuzzy*

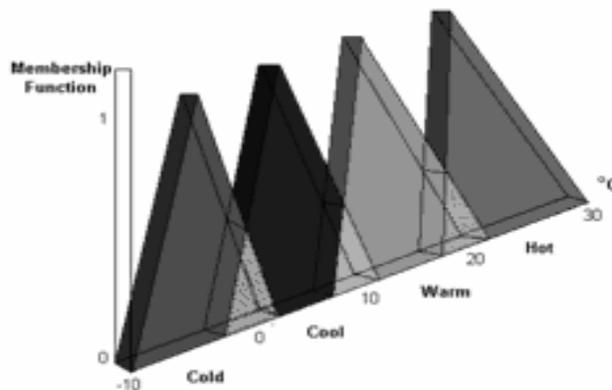
- Tahun 1965, Prof. Zadeh dari California Univ. menulis paper yang sangat terkenal dan menjadi bahan referensi utama dunia Fuzzy.
- Tahun 1974, Prof Mandani dari London Univ. menerapkan teori Fuzzy untuk mengendalikan steam engine.
- Tahun 1970-an, Institusi Ilmiah mengenai Fuzzy Logic terbentuk dan teori mengenai Fuzzy berkembang pesat.
- Tahun 1980-an, Fuzzy masuk pada dunia terapan seperti industri semen, IC fuzzy, komputer fuzzy, kendali fuzzy, dsb.

2.3.1.1.2 Sistem *Fuzzy*

Fungsi keanggotan

- Suatu fungsi yang mengekspresikan satu kata dalam bahasa manusia.
- Suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya yang memiliki interval antara 0 sampai 1.

Fuzzy characterization



Gambar 2.6 - Karakteristik suhu dalam suatu ruangan dalam Fuzzy

- Himpunan Fuzzy dikatakan memiliki **bentuk normal maksimum**, jika paling sedikit satu elemennya memiliki nilai keanggotaan 1 (satu) dan satu elemennya memiliki nilai keanggotaan 0 (nol).
- Himpunan Fuzzy dikatakan memiliki **bentuk normal minimum**, jika paling sedikit satu elemennya memiliki nilai keanggotaan 1 (satu).

Komplemen (Negasi) Himpunan Fuzzy.

$$\mu' (x) = 1 - \mu (x)$$

2.3.1.1.3 *If – Then Rule*

Interpretasikan if-then aturan adalah suatu three-part memproses. Proses ini diterangkan secara detil di (dalam) bagian yang berikutnya: *Fuzzify inputs* : Selesaikan semua fuzzy statement pada antecedent yang mempunyai derajat keanggotaan antara 0 dan 1. Jika hanya ada satu bagian dari antecedent, ini adalah derajat tingkat pendukung untuk aturan itu.

Apply fuzzy operator to multiple part antecedents : Jika ada berbagai [part;bagian] kepada yang terdahulu, [menerapkan/berlaku] operator logika fuzzy dan memecahkan yang terdahulu [itu] ke nomor;jumlah tunggal antar[a] 0 dan 1. Ini adalah derajat tingkat pen;dukungan untuk aturan [itu].

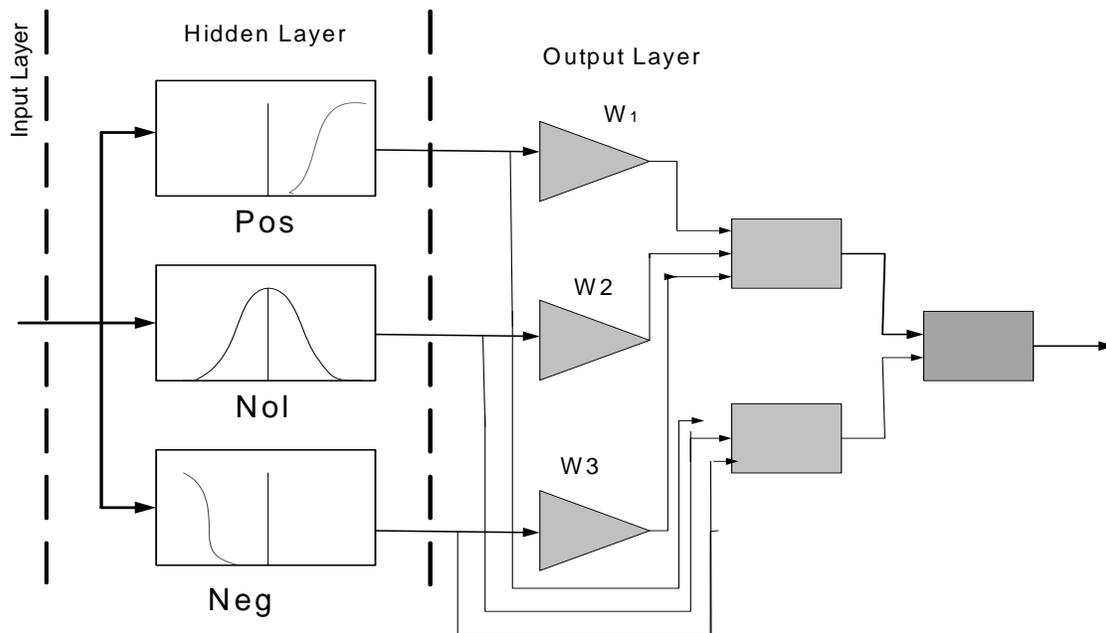
Apply implication method : Gunakan derajat tingkat pen;dukungan untuk keseluruhan aturan untuk membentuk output fuzzy set. Yang sebagai akibat untuk suatu aturan tidak jelas menugaskan suatu keseluruhan tidak jelas mulai keluaran [itu]. [yang] di-set tidak jelas ini Wakili oleh suatu fungsi keanggotaan yang di/terpilih untuk menandai (adanya) kualitas yang sebagai akibat. Jika yang terdahulu hanya secara parsial benar, (yaitu., ditugaskan suatu nilai kurang dari 1), kemudian keluaran yang tidak jelas di-set dipotong ujung menurut metoda implikasi.

Fungsi-fungsi Implikasi :

- *Conditional Fuzzy Proposition*
 - Digunakan pernyataan IF
 - IF x is A THEN y is B
 - (x dan y adalah skalar, sdg A dan B adalah variabel linguistik)
 - Ada 2 Fungsi implikasi : Min dan Dot
- *Unconditional Fuzzy Proposition*
 - Tidak digunakan pernyataan IF
 - X is A
 - (x adalah skalar, sdg A adalah variabel linguistik)

Pendekatan Fungsi Neuro Fuzzy

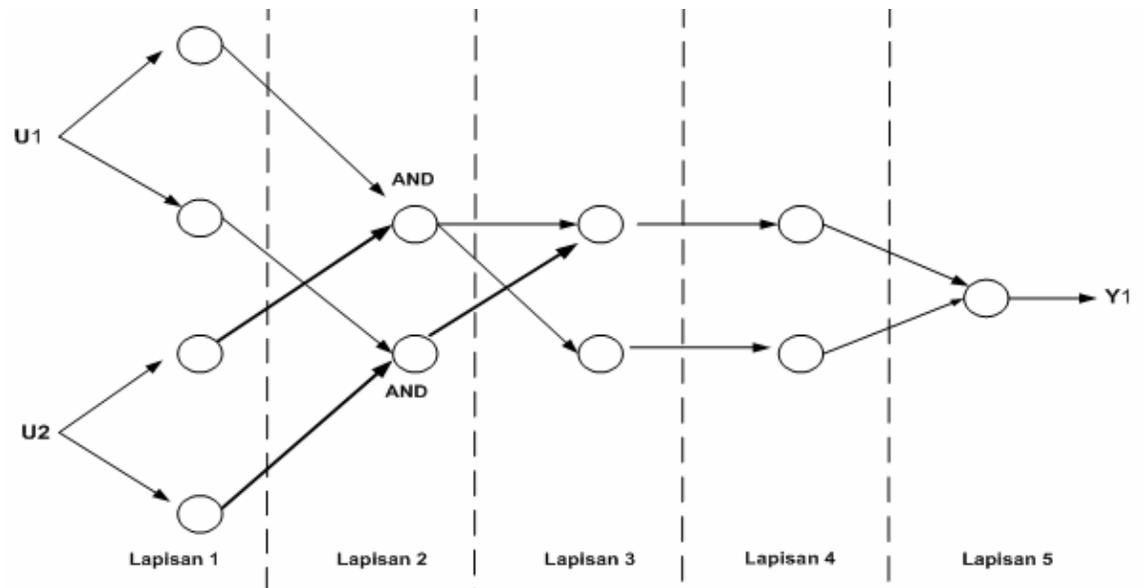
- Misalkan ada 3 Rule
 - IF e is POS Then u is 100
 - IF e is NOL Then u is 0
 - IF e is NEG Then u is - 100
- e adalah input
- U adalah output



Gambar 2.7 - Blok diagram uantuk 3 Rule

2.3.1.1.4 Adaptive Neuro Fuzzy Inference System (ANFIS)

- Arsitektur secara fungsional sama dengan fuzzy rule base model Sugeno
- Menggunakan algoritma pembelajaran terhadap sekumpulan data.
- Arsitektur ANFIS dengan batasan tertentu sama dengan arsitektur Jaringan Syaraf fungsi radial
- Arsitektur ANFIS (mis. untuk 2 input dan 1 output)



Gambar 2.8 - Struktur jaringan ANFIS

Penjelasan Arsitektur ANFIS

- Tiap-tiap neuron i lapisan ke 1 adaptif terhadap parameter fungsi aktivasi
- Tiap-tiap node lapisan ke 2 berupa node tetap yang outputnya adalah hasil masukan.
- Tiap-tiap node lapisan ke 3 berupa node tetap yang merupakan hasil perhitungan rasio dari α predikat dari aturan ke- i terhadap jumlah keseluruhan α predikat \diamond *normalized firing strength*

$$\alpha_i = \alpha_i / (\alpha_1 + \alpha_2) ; \text{ dengan } i = 1, 2.$$

- Tiap-tiap node lapisan ke 4 merupakan node adaptif terhadap suatu output.

$$\alpha_i y_i = \alpha_i (c_{i1} u_1 + c_{i2} u_2 + c_{i0}) ; \text{ dengan } i = 1, 2.$$

- Tiap-tiap node lapisan ke 5 adalah node tetap yang merupakan jumlahan dari semua masukan.

2.3.1.2 Neural Networks

Neural Networks (Jaringan Saraf Tiruan) menurut Haykin didefinisikan sebagai berikut : *“Sebuah neural network (Jaringan Saraf Tiruan) adalah prosesor yang terdistribusi paralel, terbuat dari unit-unit yang sederhana, dan memiliki kemampuan untuk menyimpan pengetahuan yang diperoleh secara eksperimental dan siap pakai untuk berbagai tujuan. Neural network ini meniru otak manusia dari sudut :*

1. *Pengetahuan diperoleh oleh network dari lingkungan, melalui suatu proses pembelajaran.*
2. *Kekuatan koneksi antar unit yang disebut synaptic weights, berfungsi untuk menyimpan pengetahuan yang telah diperoleh oleh jaringan tersebut.”*

Sebuah neural network dapat dianalisa dari dua sisi:

- Bagaimana neuron-neuron tersebut dirangkaikan dalam suatu jaringan (arsitektur).
- Bagaimana jaringan tersebut dilatih agar memberikan output sesuai dengan yang dikehendaki (algoritma pembelajaran). Algoritma pembelajaran ini menentukan cara bagaimana nilai penguatan yang optimal diperoleh secara otomatis.

Berdasarkan arsitekturnya, neural network dapat dikategorikan, antara lain, *single-layer neural network*, *multilayer neural network*, *recurrent neural network* dan sebagainya.

Berawal dari diperkenalkannya model matematika neuron oleh McCulloch & Pitts, penelitian di bidang *neural network* berkembang cukup pesat, dan mencapai puncak keemasan pertama pada era tahun 60, dan puncak kedua pada pertengahan tahun 80-an. Penelitian dalam bidang ini, dapat dibagi dalam tiga kategori:

1. Riset untuk meneliti proses informasi yang terjadi pada otak dan jaringan saraf. Tema ini merupakan porsi penelitian para ahli medis dan *neuroscientist*.
2. Penelitian teoritis untuk mendalami konsep dasar proses informasi pada otak. Kategori ini memerlukan ketajaman analisa matematika untuk menggali dasar-dasar teori dari proses tersebut.
3. Penelitian yang bertujuan memanfaatkan teori-teori yang telah ada untuk aplikasi. Dalam hal ini, perlu sekali memperhatikan tingkat akurasi sistem, dan menekan biaya serendah mungkin (*low cost solution*).

Dewasa ini, neural network telah diaplikasikan di berbagai bidang. Hal ini dikarenakan neural network memiliki kelebihan-kelebihan sbb.

1. Dapat memecahkan problema non-linear yang umum dijumpai di aplikasi.

2. Kemampuan memberikan jawaban terhadap *pattern* yang belum pernah dipelajari (*generalization*).
3. Dapat secara otomatis mempelajari data numerik yang diajarkan pada jaringan tersebut.

Untuk penggunaan *neural networks* pada *video game* masih terbilang masih cukup jarang, Sebenarnya AI dengan menggunakan neural network bisa terus dikembangkan dan sangat fleksibel., sayangnya karena proses training yang banyak memakan waktu, membuat para pembuat game terutama pada game action tidak memilih metode ini, tetapi tidak menutup kemungkinan untuk tidak menggunakan metode ini.

Game yang pernah menerapkan metode ini salah satunya merupakan jenis *game racing* yaitu “*GTI Racing*”. Pada game ini komputer dapat mempelajari cara player menancap gas, rem ditikungan. Pada putaran pertama komputer melakukan proses pembelajaran pada aksi player, setelah proses pembelajaran selesai maka komputer akan tahu bagaimana cara player bermain. Pada putaran kedua jangan harap player dapat melawan komputer dengan cara yang sama.

2.4 Algoritma

Kata algoritma berasal dari kata *Algoris* dan *Ritmis*, pertama kali diungkapkan oleh Abu Ja'far Mohammed ibn Musa al Khowarizmi dalam buku *Al-jabr W'almulqabala* (Horowitz, Ellis dan Sartaj Sahni, 1978, p1).

Beberapa pengertian algoritma, antara lain adalah sebagai berikut :

1. Menurut D.R. Stinson

Algoritma adalah sebuah kumpulan aturan-aturan untuk menyelesaikan suatu masalah dalam tahapan-tahapan yang terstruktur dan terbatas.

2. Menurut Abu Ja'far Mohammad Ibn Musa Al Khowarizmi

Algoritma adalah suatu fungsi khusus untuk menyelesaikan suatu persoalan.

3. Menurut Goodman Hedetniemi

Algoritma adalah urutan terbatas dari operasi-operasi yang terdefinisi dengan baik, yang masing-masing membutuhkan memori dan waktu yang terbatas untuk menyelesaikan suatu masalah.

4. Menurut Gamedev.net (<http://www.gamedev.net>)

Algoritma adalah sekumpulan instruksi untuk melaksanakan suatu tugas.

Berdasarkan definisi di atas maka pengertian algoritma bila dipandang dari ilmu komputer adalah suatu fungsi yang terdiri dari serangkaian langkah-langkah yang terstruktur dan dituliskan secara sistematis yang akan dikerjakan untuk menyelesaikan masalah dengan bantuan komputer.

2.5 Algoritma *Searching*

Dalam Teknologi Informatika, algoritma *searching* adalah algoritma yang mengambil suatu masalah sebagai input dan mengembalikannya sebagai solusi untuk masalah tersebut, umumnya setelah memeriksa sekumpulan kemungkinan solusi. Kebanyakan dari algoritma yang dipelajari oleh para ahli yang

menyelesaikan masalah algoritma *search*. Kumpulan dari semua kemungkinan solusi untuk masalah disebut ruang pencarian (*search space*). Algoritma pencarian tanpa informasi (*uninformed search algorithm*) menggunakan metode pencarian paling sederhana pada ruang pencarian, sedangkan algoritma pencarian dengan informasi menggunakan heuristik untuk mendapatkan pengetahuan tentang struktur dari ruang pencarian untuk mencoba mengurangi waktu pencarian.

Secara umum algoritma pencarian jalan dibagi menjadi 2 bagian, yaitu :

a. Uninformed Search Algorithm

Uninformed Search Algorithm adalah algoritma yang tidak memiliki keterangan tentang jarak atau biaya dari path dan tidak memiliki pertimbangan akan path mana yang lebih baik. Algoritma ini hanya dapat membedakan yang mana goal dan yang mana bukan goal. *Algoritma Uninformed Search* juga disebut sebagai algoritma pencarian buta (*blind search*).

b. Informed Search Algorithm

Informed Search Algorithm adalah algoritma yang memiliki keterangan tentang jarak atau biaya dari path dan menggunakan pertimbangan berdasarkan pengetahuan dan path mana yang lebih baik.

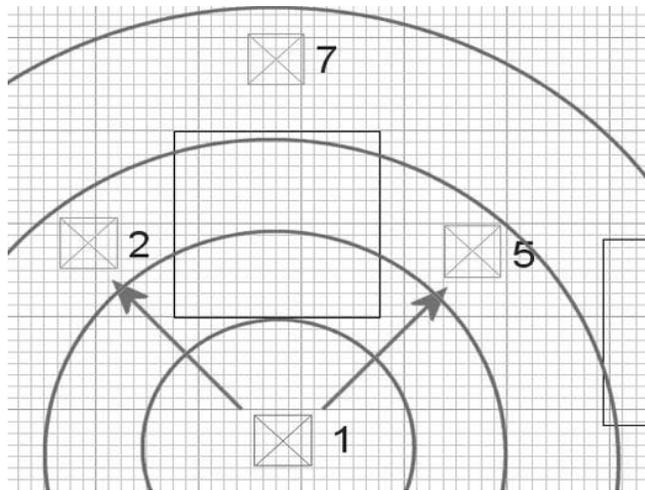
2.6 Waypoints Search

Waypoint search merupakan pencarian yang paling efisien dan menghemat konsumsi pada CPU-time. Hampir semua game sekarang ini menggunakan metode waypoints sebagai search, terutama pada 3D game action, contoh game yang menerapkan waypoint sebagai solusinya adalah “*Counter Striker*”. Terdapat dua metode pencarian dengan menggunakan waypoint search, yaitu *Best-first search* dan Data-Based Search.

2.6.1 Path Finding Algorithm

Path finding algorithm adalah konsep pencarian dengan menggunakan *best-first search*. Pada proses ini kami menempatkan node di dalam map dan diberi nomor id pada setiap node yang ada. Node tersebut akan men-scan setiap node yang satu dengan lain : node 1 scan node 2,3,4, ... , dan seterusnya. Dan node 2 scans node 1,3,4,... , dan seterusnya. Sewaktu semua node telah di scan maka akan dimasukkan ke dalam variabel yang telah kami beri nama `distances_computed` path finding algorithm dapat di jalankan

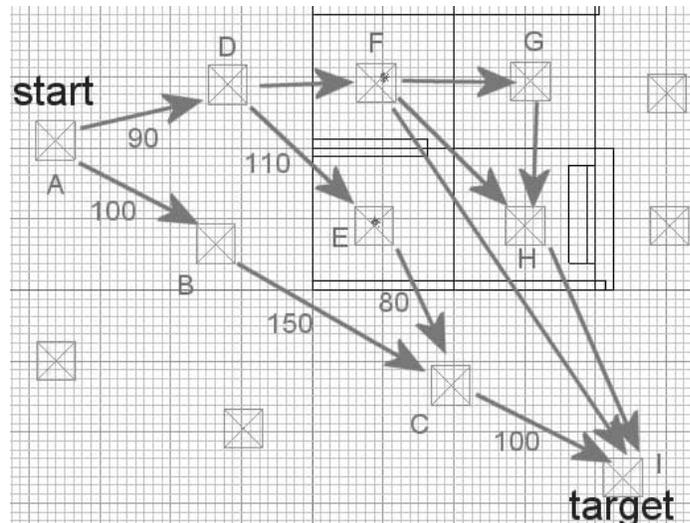
Tetap bagaimana setiap node dapat membaca satu sama lain? Lihatlah gambar dibawah ini.



Gambar 2.9 Scan dari node ke node

kita asumsikan bahwa node 1 men-scan node yang ada dikelilinginya dimana terdapat node 2,5,7. pada dasarnya scan dapat menembus semua obstacle yang ada. Maka untuk menghindari hal tersebut, kita menambahkan metode dimana node tersebut hanya mendapatkan nilai distance antara yang node satu dengan node yang lain dengan cara melihat langsung tanpa ada obstacle. Kita memanbakan sebuah vector di dalamnya yang vector tersebut yang memberikan nilai *distances* antara node dengan node yang lain atau dengan obstacle. Node 1 dapat melihat langsung ke node 2 dan 4 maka $node_to_node[1, 2] = 300$ dan $node_to_node[1, 5] = 340$, sedangkan node 1 tidak melihat node 7

karena ada penghalang maka $note_to_node[1, 7] = 99999$. Dan begitu seterusnya



Gambar 2.10 Menjumlahkan total node terpendek

Dibawah terdapat potongan algoritma path finding

```

IF (my.scan(radius) != null) {
    IF(my.vector(500,0,0) == my.scan(radius)) {
        copy_node();
    }
    Else {
        break;
    }
}

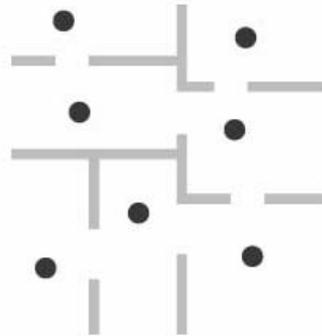
```

2.6.2 Data-Based Search

Pada proses ini komputer tidak perlu menghitung lagi setiap jarak setiap titik satu dengan lainnya untuk mencari jarak terpendek. Kita telah membuat semua nilai distance dari node yang satu dengan node yang lain yang sudah dimasukkan ke dalam map.

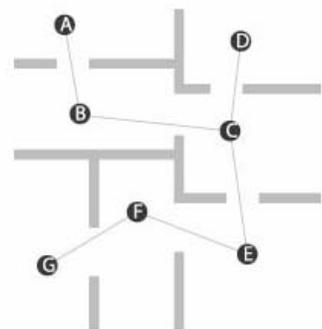
Node-node tersebut saling saling berhubungan dari titik yang satu ke titik yang lain. Dengan menggunakan metode ini sangat menguntungkan proses kerja komputer dan CPU-time karena sangat menghemat waktu, karena semuanya sudah dimasukkan ke dalam data-base.

Dibawah terdapat contoh gambar sebuah map yang sudah diberi titik-titik waypoint.



Gambar 2.11 Panempatan Node

Pada gambar 2.11, setiap map diberi node-node yang mewakili daerah tersebut, dimana node-node tersebut akan dibuat saling berhubungan.



Gambar 2.12 Tiap Nodes Diberi Label

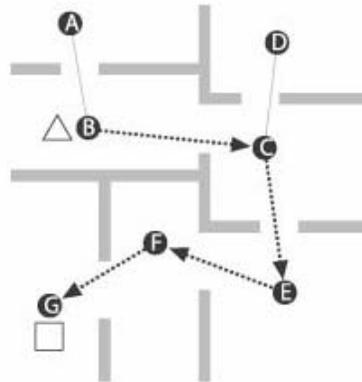
Selanjutnya kita tinggal membuat diagram hubungan antara node satu dengan yang lain, seperti pada gambar 2.12.

Tabel 2.2 - Table Node

		<i>End</i>						
		A	B	C	D	E	F	G
<i>Start</i>	A	—	B	B	B	B	B	B
	B	A	—	C	C	C	C	C
	C	B	B	—	D	E	E	E
	D	C	C	C	—	C	C	C
	E	C	C	C	C	—	F	F
	F	E	E	E	E	E	—	G
	G	F	F	F	F	F	F	—

Pada gambar 2.12, dibuat tabel yang menghubungkan antara tiap node. Jika posisi komputer berada di B ingin bergerak menuju G maka jalur yang bisa di lalui berdasarkan tabel di atas

- *Start* di baris *B* menuju kolom *End G* berdasarkan tabel kemungkinan yang bisa dilalui yaitu B
- Berikutnya *Start* di *C* menuju *End G* melalui *E*
- Berikutnya *Start* di *E* menuju *End G* melalui *F*
- Berikutnya *Start* di *F* menuju *End G* melalui *G*



Gambar 2.13 – Finding the path

B sebagai Komputer, G sebagai target / Player. Maka berdasarkan tabel pada gambar 2.13, node yang akan dilalui computer adalah sebagai berikut $B \rightarrow C \rightarrow E \rightarrow F \rightarrow G$.

2.7 Interaksi Manusia dan Komputer

Interaksi Manusia dan Komputer adalah sebuah cabang ilmu yang mempelajari perancangan, evaluasi, dan implementasi sistem komputer interaktif untuk digunakan oleh manusia, serta studi fenomena-fenomena besar yang berhubungan dengannya (*Shneiderman*, 1998, p8).

2.7.1 Tujuan Interaksi Manusia dan Komputer

Sistem yang efektif akan menghasilkan rasa keberhasilan, kompetensi, penguasaan, dan kejelasan dalam komunitas pemakai. Tujuan dari rekayasa sistem IMK adalah untuk menghasilkan sistem dengan (*Shneiderman*, 1998, p9-14).

1. Fungsionalitas yang sebenarnya

Sistem dengan fungsionalitas yang kurang memadai akan mengecewakan pemakai dan sering ditolak atau tidak digunakan. Sedangkan sistem yang berlebihan akan menyebabkan kesulitan dalam implementasi, pemeliharaan, dan penggunaan.

2. Keandalan, ketersediaan, keamanan, dan integritas data

- a. Keandalan (*reliability*): berfungsi seperti yang diinginkan.
- b. Ketersediaan (*availability*): tersedia ketika hendak digunakan.
- c. Keamanan (*security*): terlindung dari akses yang tidak diinginkan dan kerusakan yang disengaja.
- d. Integritas data (*data integrity*): keutuhan data terjamin.

3. Standarisasi, integrasi, konsistensi, dan portabilitas

- a. Standarisasi: keseragaman sifat-sifat antarmuka pemakai pada aplikasi yang berbeda, misalnya dengan menggunakan standar industri yang ada.
- b. Integrasi: keterpaduan antar paket aplikasi dan *software tools*.
- c. Konsistensi: keseragaman dalam suatu program aplikasi.
- d. Portabilitas: dimungkinkan data dikonversi pada berbagai *hardware* dan *software*.

4. Penjadwalan dan anggaran

Perencanaan yang hati-hati dan manajemen yang berani diperlukan karena proyek harus sesuai dengan jadwal dan dalam anggaran.

2.8 Data

Menurut Prahasta (2002) data adalah representasi dari kenyataan apa adanya di lapangan, konsep-konsep atau instruksi-instruksi yang diformalkan dan sesuai untuk komunikasi, interpretasi atau pemrosesan, baik yang dilakukan oleh manusia maupun secara otomatis dengan bantuan mesin dan alat-alatnya. Maka dari itu data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak sehingga perlu diolah lebih lanjut.

2.9 Informasi

Menurut Prahasta (2002, p30) informasi adalah makna atau pengertian yang dapat diambil dari suatu data dengan menggunakan konversi-konversi yang umum digunakan di dalam representasinya. Dengan demikian informasi dapat menjadi masukan yang berguna dalam pengambilan keputusan.

2.10 Game

Pada umumnya permainan merupakan sekumpulan aturan. Dalam permainan digunakan suatu pola pencarian ruang keadaan (*state space search*) dimana permainan akan menghasilkan sejumlah pencarian ruang. Untuk memperoleh hasil terbaik digunakan heuristik. Dalam permainan, komputer dapat

bertindak seolah-olah dijalankan oleh manusia, dimana dapat berpikir dan menganalisa setiap langkah.

2.11 Simulasi

Simulasi adalah pemalsuan dari suatu benda, keadaan, atau proses asli dari suatu hal. Dalam teknologi informatika, simulasi memiliki arti khusus: Alan Turing menggunakan istilah "simulasi" untuk menjelaskan yang terjadi saat komputer digital mengeksekusi suatu program yang menjelaskan keadaan transisi, input dan output. Dalam pemrograman komputer, simulasi sering digunakan untuk mengeksekusi program yang hanya dapat dijalankan dalam komputer tertentu atau dalam keadaan tertentu.

2.12 Database

2.12.1 Definisi Database

Menurut Prahasta (2002, p190) *database* adalah himpunan kelompok data (file/arsip) yang saling berhubungan dan terorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah. Dengan demikian *database* merupakan salah satu komponen yang penting dalam sistem karena dapat menyediakan informasi bagi pemakai.

Menurut Elmasri (2000 , p4), basis data adalah kumpulan data yang saling berhubungan. Pengertian basis data menurut Mcleod, R(2000 , p16-17) adalah suatu koleksi data komputer yang terintegrasi, diatur dan disimpan dengan suatu cara yang memudahkan untuk diambil kembali.

Menurut Date (2000, p10) database merupakan kumpulan data persistent yang digunakan oleh sistem aplikasi pada beberapa perusahaan.

Database menurut Connolly & Begg (2002, p14), adalah “*a shared collection of logically related data, and a description of this data, designed to meet the information needs of an organization*”. Dalam bahasa Indonesia, definisi tersebut dapat dikatakan sebagai berikut : “kumpulan data yang terhubung secara logis, beserta deskripsi data tersebut, yang dirancang untuk memenuhi kebutuhan informasi sebuah organisasi”

Dalam praktek, penggunaan istilah basis data menurut Elmasri lebih dibatasi pada arti implisit yang khusus yaitu:

1. Basis data merupakan penyajian suatu aspek dari dunia nyata. Terkadang disebut “*miniworld*” atau “*Universe of Disclosure*”. Perubahan-perubahan yang terjadi pada *miniworld* direfleksikan pada database.
2. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang terkumpul secara acak dan tanpa mempunyai arti tidak dapat disebut basis data.
3. Basis data perlu dirancang, dibangun, dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa pemakai dan beberapa aplikasi yang sesuai dengan kepentingan pemakai.

Sedangkan pengertian sistem basis data adalah komputerisasi sistem penyimpanan data yang bertujuan menyimpan dan memelihara informasi serta mengizinkan user untuk melakukan pengupdatean atau pengambilan data yang dibutuhkan (Date, 2000, p5).

2.12.2 Konsep Database

McLeod, R (2001, p182) menjelaskan konsep basis data sebagai sebuah integrasi logis atas sejumlah *file*.

Menurut Elmasri (2000 , p4), basis data adalah kumpulan data yang saling berhubungan. Pengertian basis data menurut Mcleod, R(2000 , p16-17) adalah suatu koleksi data komputer yang terintegrasi, diatur dan disimpan dengan suatu cara yang memudahkan untuk diambil kembali.

Menurut Date (2000, p10) database merupakan kumpulan data persistent yang digunakan oleh sistem aplikasi pada beberapa perusahaan.

Database menurut Connolly & Begg (2002, p14), adalah “*a shared collection of logically related data, and a description of this data, designed to meet the information needs of an organization*”. Dalam bahasa Indonesia, definisi tersebut dapat dikatakan sebagai berikut : “kumpulan data yang terhubung secara logis, beserta deskripsi data tersebut, yang dirancang untuk memenuhi kebutuhan informasi sebuah organisasi”

Dalam praktek, penggunaan istilah basis data menurut Elmasri lebih dibatasi pada arti implisit yang khusus yaitu:

4. Basis data merupakan penyajian suatu aspek dari dunia nyata. Terkadang disebut “*miniworld* ” atau “*Universe of Disclosure*”. Perubahan-perubahan yang terjadi pada *miniworld* direfleksikan pada database.
5. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang terkumpul secara acak dan tanpa mempunyai arti tidak dapat disebut basis data.
6. Basis data perlu dirancang, dibangun, dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa pemakai dan beberapa aplikasi yang sesuai dengan kepentingan pemakai.

Sedangkan pengertian sistem basis data adalah komputerisasi sistem penyimpanan data yang bertujuan menyimpan dan memelihara informasi serta mengizinkan user untuk melakukan pengupdatean atau pengambilan data yang dibutuhkan (Date, 2000, p5).