

BAB 2

LANDASAN TEORI

2.1 Kecerdasan Buatan

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 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). Namun bidang kecerdasan buatan baru dianggap sebagai bidang tersendiri di konferensi Dartmouth tahun 1956, dimana sepuluh peneliti muda memimpikan mempergunakan komputer untuk memodelkan bagaimana cara berpikir manusia. Hipotesis mereka adalah “Mekanisme berpikir manusia dapat secara tepat dimodelkan dan disimulasikan pada komputer digital”, dan ini yang menjadi landasan dasar kecerdasan buatan.

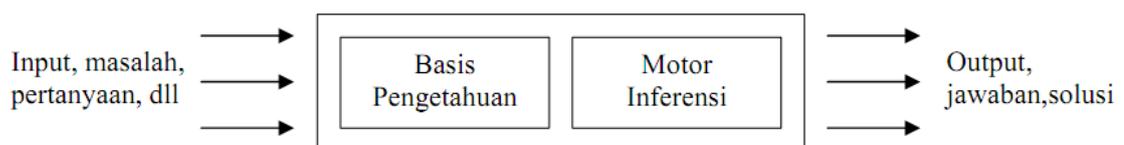
Kecerdasan buatan atau *Artificial Intelligence* merupakan bagian dari ilmu pengetahuan komputer yang khusus ditujukan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer. Sistem memperlihatkan sifat-sifat khas yang dihubungkan dengan kecerdasan dalam kelakuan atau tindak-tanduk yang sepenuhnya bisa menirukan beberapa fungsi otak manusia, seperti pengertian bahasa, pengetahuan, pemikiran, pemecahan masalah dan lain sebagainya [4].

Inteligensia adalah seseorang yang pandai melaksanakan pengetahuan yang dimilikinya. Dengan demikian, walaupun seseorang memiliki banyak pengetahuan, tetapi apabila tidak dapat melaksanakannya dalam praktik, maka ia tidak dapat digolongkan inteligensia. Pada batas-batas tertentu, kecerdasan buatan memungkinkan komputer dapat menerima pengetahuan melalui input manusia dan menggunakan pengetahuannya melalui simulasi proses penalaran dan berpikir manusia untuk memecahkan berbagai masalah. Kecerdasan buatan mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia [2].

Ada dua bagian utama dari aplikasi kecerdasan buatan [5], yaitu:

1. Basis pengetahuan (*knowledge base*), yakni suatu pengertian tentang beberapa wilayah subyek yang diperoleh melalui pendidikan dan pengalaman. Pengetahuan berisi fakta-fakta, teori, pemikiran, prosedur dan hubungan antara satu dengan lainnya.
2. Motor inferensi (*inference engine*), yakni kemampuan menarik kesimpulan berdasarkan pengalaman.

Pada saat basis pengetahuan sudah terbentuk, maka teknik kecerdasan buatan bisa digunakan untuk memberikan kemampuan baru kepada komputer agar bisa berpikir, menalar, dan membuat inferensi, serta pertimbangan yang didasarkan pada fakta dan hubungan-hubungannya yang terkandung dalam basis pengetahuan. Dengan basis pengetahuan dan kemampuan inferensi, komputer dapat disejajarkan sebagai alat bantu yang bisa digunakan secara praktis dalam memecahkan masalah dan pengambilan keputusan [6].



Gambar 2.1 Penerapan Konsep Kecerdasan Buatan dalam Komputer

(Sri Kusumadewi, 2003)

2.1.1 Perbandingan Kecerdasan Buatan dengan Kecerdasan Alami

Teknik pembuatan program kecerdasan buatan berbeda dengan teknik pemrograman menggunakan bahasa konvensional (alami). Program komputer konvensional didasarkan pada suatu algoritma yang disusun dengan jelas, rinci, serta langkah demi langkah sampai pada hasil yang sudah ditentukan sebelumnya. Algoritma tersebut kemudian dikonversi ke dalam program komputer yang memberitahu komputer secara pasti instruksi apa yang harus dikerjakan. Selanjutnya, algoritma bisa digunakan untuk mengolah data bilangan, huruf, atau kata lainnya untuk menyelesaikan masalah.

Sebaliknya, kecerdasan buatan tidak didasarkan pada algoritma, tetapi pada representasi dan manipulasi simbol. Simbol bisa berupa cetakan atau elektronik yang digunakan untuk menggambarkan objek, proses, dan hubungannya. Sebuah obyek bisa jadi seorang manusia, benda, pikiran, konsep, kejadian, atau pernyataan. Basis pengetahuan yang berisi fakta, konsep, dan hubungan di antara keduanya dapat diciptakan dengan menggunakan simbol. Kemudian beberapa proses dapat digunakan untuk memanipulasi simbol tersebut untuk menghasilkan nasehat atau rekomendasi untuk penyelesaian suatu masalah.

Kelebihan dari kecerdasan buatan [2] antara lain adalah:

1. Kecerdasan buatan lebih bersifat permanen karena tidak berubah selama sistem komputer dan program tidak mengubahnya. Kecerdasan alami bisa berubah karena sifat manusia yang pelupa.
2. Kecerdasan buatan lebih mudah diduplikasi dan disebarkan karena mentransfer pengetahuan manusia dari satu orang ke orang lain membutuhkan proses yang sangat lama dan keahlian tidak akan pernah dapat diduplikasi dengan lengkap. Jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.
3. Kecerdasan buatan akan lebih mudah dan murah dibandingkan mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.

4. Kecerdasan buatan bersifat konsisten karena bagian dari teknologi komputer sedangkan kecerdasan alami senantiasa berubah-ubah.
5. Kecerdasan buatan dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut.
6. Kecerdasan buatan cenderung dapat mengerjakan pekerjaan lebih baik dan lebih cepat dibanding dengan kecerdasan alami.

Kelebihan dari kecerdasan alami [2] yaitu:

1. Memungkinkan orang untuk menggunakan pengalaman secara langsung. Sedangkan pada kecerdasan buatan harus bekerja dengan input-input simbolik.
2. Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.
3. Kecerdasan alami bersifat kreatif karena dapat berkembang dengan mudah dan dapat menciptakan kreasi baru.

Dari uraian di atas diperoleh kesimpulan dari perbandingan antara kecerdasan buatan dan kecerdasan alami yang terlihat seperti pada Tabel 2.1 berikut.

Tabel 2.1 Perbandingan Antara Kecerdasan Buatan dan Kecerdasan Alami

Aspek	Kecerdasan Buatan	Kecerdasan Alami
Pemrosesan	Sebagian simbolik	Algoritmik
Pencarian	Heuristik	Algoritma
Input	Bisa tidak lengkap	Harus lengkap
Penjelasan	Disediakan	Tidak disediakan
Struktur	Kontrol dipisahkan dengan pengetahuan	Kontrol terintegrasi dengan informasi
Output	Kuantitatif	kualitatif
Fokus	Pengetahuan	Data
Pemeliharaan dan peningkatan	Relatif mudah	Sulit
Perangkat keras	<i>Workstation</i> dan PC	Semua tipe
Kemampuan menalar	Ya	Tidak
Mencari jawaban	Memuaskan	Optimal
Modifikasi	Berkali-kali	Jarang

(Anita Desiani dan Muhammad Arhami, 2006)

2.1.2 Area Aplikasi Kecerdasan Buatan

Kecerdasan buatan seperti bidang ilmu lainnya juga memiliki beberapa bidang disiplin ilmu yang sering digunakan untuk pendekatan yang esensial bagi penyelesaian suatu masalah dan dengan aplikasi bidang kecerdasan buatan yang berbeda [7], diantaranya adalah:

1. Sistem Pakar (*Expert System*)

Sistem pakar adalah sistem komputer yang menggabungkan dasar pengetahuan (*knowledge base*) dengan sistem inferensi untuk bisa menyamai atau meniru kemampuan seorang pakar dalam menyelesaikan suatu masalah tertentu. Contohnya adalah sistem pakar untuk diagnosis kasus medis, sistem pakar untuk analisis keuangan, dan sebagainya.

2. Pengolahan Bahasa Alami (*Natural Language Processing*)

Pengolahan bahasa alami adalah program yang mampu memahami dan membangkitkan bahasa manusia dengan mengidentifikasi sintaksis, semantik, dan konteks dalam satu kalimat agar bisa sampai pada kesimpulan untuk memberikan jawaban. Pengolahan bahasa alami bisa digunakan sebagai sistem penerjemah, pengenalan ucapan (*speech recognition*), dan sebagainya.

3. Robotik dan Sistem Sensor (*Robotics and Sensory Systems*)

Robotik adalah suatu bidang rekayasa yang membahas duplikasi kemampuan fisik manusia, yaitu komplemen alami dalam bidang kecerdasan buatan yang berusaha menirukan kemampuan mental manusia. Robot adalah mesin yang mampu melaksanakan fungsi fisik manusia secara terbatas. Robot harus menggunakan sensor agar bisa menerima input lingkungannya dan dengan dilengkapi perangkat lunak cerdas maka robot dapat melakukan suatu pekerjaan seperti yang sudah diprogramkan.

4. Komputer Vision (*Computer Vision*)

Komputer vision merupakan alat analisis dan evaluasi informasi visual dengan menggunakan komputer dengan menguji sebuah gambar atau adegan nyata

dengan mengidentifikasi objek, ciri-ciri, atau pola-polanya. Salah satu contoh aplikasi adalah analisis potret suatu wilayah untuk keperluan intelijen.

5. *Computer Based Training (CBT)*

CBT adalah aplikasi kecerdasan buatan yang menghasilkan komputer sebagai tutor untuk dapat melatih dan mengajar. Bahkan *CBT intelligence* bisa melakukan dialog dengan menggunakan bahasa alami.

6. Permainan (*Game Playing*)

Permainan dalam kecerdasan buatan berupa permainan antara manusia melawan mesin yang memiliki intelektual untuk berpikir. Komputer dapat bereaksi dan menjawab tindakan-tindakan yang diberikan oleh lawan mainnya. Banyak permainan komputer telah dibuat dan dikembangkan. Salah satu contohnya adalah permainan catur.

2.1.3 Metode Pencarian

Ada dua hal yang sangat mendasar mengenai penelitian kecerdasan buatan, yaitu *knowledge representation* (representasi pengetahuan) dan *search* (pencarian). Basis pengetahuan hanyalah sekumpulan fakta mengenai suatu subjek. Untuk memanfaatkan basis pengetahuan yang sudah ada di dalam memori komputer, diperlukan beberapa macam mekanisme inferensi agar dapat melakukan penalaran dan mengambil keputusan. Hal itu dilakukan oleh suatu program yang menggunakan teknik pencarian dan pencocokan pola.

Pencarian adalah teknik mencari jawaban atau solusi dengan cara menguji sekumpulan kemungkinan ruang keadaan (*state space*) yang ada dari suatu permasalahan dengan cara "*trial and error*". Pencarian lebih mengarah pada pendekatan menuju fakta-fakta baru dan akhirnya sampai pada suatu jawaban atau solusi. Ada dua macam pencarian ruang keadaan, yaitu *blind search* (pencarian buta) dan *heuristic search* (pencarian heuristik). Keduanya digunakan dalam aplikasi kecerdasan buatan, tetapi pencarian heuristik merupakan metode utama dalam kecerdasan buatan [6].

Pencarian buta merupakan pencarian yang penelusurannya dimulai dengan tidak ada informasi awal yang digunakan dalam proses pencarian. Ada empat teknik utama pencarian buta yang digunakan pada setiap proses pencarian kecerdasan buatan, yaitu *depth first*, *breadth first*, *forward chaining*, dan *backward chaining*. Keempat teknik ini selalu bisa dikombinasikan dengan berbagai cara.

Sebenarnya, pencarian heuristik pun merupakan pencarian buta yang telah diberi bimbingan atau pengarahan. Pencarian heuristik merupakan pencarian yang penelusurannya dimulai dengan adanya informasi awal yang digunakan dalam proses pencarian. Heuristik adalah kriteria, metode, atau prinsip-prinsip untuk memutuskan di antara beberapa alternatif dari aksi yang menjanjikan untuk menjadi yang paling efektif untuk mencapai tujuan tertentu. Contohnya *generate and test*, *hill climbing*, *best first search*, algoritma genetika, algoritma semut, dan sebagainya.

2.2 Algoritma Runut Balik (*Backtracking*)

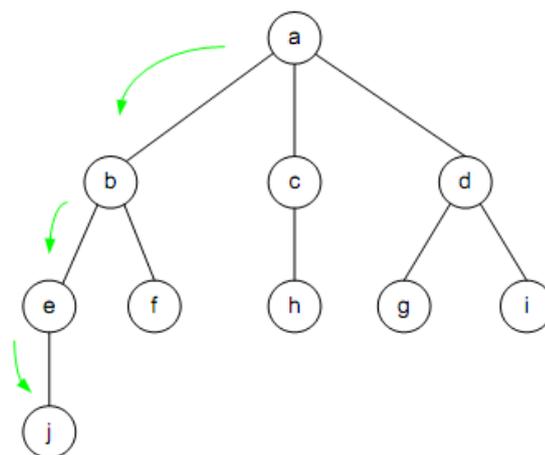
Algoritma *backtracking* pertama kali diperkenalkan oleh D.H. Lehmer pada tahun 1950. Dalam perkembangannya beberapa ahli seperti RJ Walker, Golomb, dan Baumert menyajikan uraian umum tentang *backtracking* dan penerapannya dalam berbagai persoalan dan aplikasi. Algoritma *backtracking* (runut balik) merupakan salah satu metode pemecahan masalah yang termasuk dalam strategi yang berbasis pencarian pada ruang status. Algoritma *backtracking* bekerja secara rekursif dan melakukan pencarian solusi persoalan secara sistematis pada semua kemungkinan solusi yang ada.

Algoritma *Backtracking* atau runut balik adalah algoritma yang cara kerjanya berdasarkan pada DFS (*Depth First Search*). Algoritma ini merupakan perbaikan dari algoritma *Brute Force*. Pada algoritma *Brute Force*, semua simpul kemungkinan yang ada akan dibangkitkan tanpa melihat apakah simpul tersebut mendekati solusi yang diinginkan atau tidak sehingga memakan waktu yang lama untuk menyelesaikan suatu masalah. Lain halnya dengan algoritma runut balik yang secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada kemudian membangkitkan simpul dari solusi yang mendekati penyelesaian. Dengan kata lain,

tidak perlu dibangkitkan simpul yang menjauhi solusi sehingga waktu pencarian solusi dapat dikurangi.

Pada *backtracking* semua kemungkinan solusi dibuat dalam bentuk pohon solusi (berbentuk abstrak) terlebih dahulu baru kemudian pohon tersebut dijelajahi secara DFS sampai ditemukan solusi yang layak. Algoritma ini berfungsi dengan baik untuk memecahkan masalah-masalah yang keadaannya (*state*) selalu berubah dari waktu ke waktu (*dynamic problem solving*) sehingga menjadi dasar algoritma untuk *Artificial Intelligence* (kecerdasan buatan) [9].

Nama *backtrack* didapatkan dari sifat algoritma ini yang memanfaatkan karakteristik himpunan solusinya yang sudah disusun menjadi suatu pohon solusi. Agar lebih jelas bisa dilihat pada pohon solusi berikut.



Gambar 2.2 Urutan Pencarian pada *Depth First Search*

(Adhhal Huda Bakri, 2010)

Misalkan pohon diatas menggambarkan solusi dari suatu permasalahan. Untuk mencapai solusi (j), maka jalan yang ditempuh adalah (a,b,e,j), demikian juga dengan solusi-solusi yang lain. Algoritma *backtrack* akan memeriksa mulai dari solusi yang pertama yaitu solusi (j). Jika ternyata solusi (j) bukan solusi yang layak maka penelusuran dibatalkan dan akan dilanjutkan ke solusi berikutnya. Pada pohon yang lebih rumit, cara ini akan jauh lebih efisien daripada *brute force*. Pada beberapa kasus, hasil perhitungan sebelumnya harus disimpan, sedangkan pada kasus yang lainnya tidak perlu.

2.2.1 Properti Umum dalam Algoritma Runut Balik

Properti-properti dalam algoritma runut balik adalah sebagai berikut:

1. Solusi persoalan

Solusi dinyatakan sebagai vektor dengan n-tuple : $X = (x_1, x_2, \dots, x_n)$, $x_i \in S_i$

Mungkin saja $S_1 = S_2 = \dots = S_n$.

Contoh: $S_i = \{0, 1\}$, $x_i = 0$ atau 1

2. Fungsi pembangkit nilai x_k

Dinyatakan sebagai : $T(k)$

$T(k)$ membangkitkan nilai x_k yang merupakan komponen vektor solusi

3. Fungsi pembatas (fungsi kriteria)

Dinyatakan sebagai : $B(x_1, x_2, \dots, x_k)$

Fungsi pembatas menentukan apakah (x_1, x_2, \dots, x_k) mengarah ke solusi. Jika ya, maka pembangkitan nilai untuk x_{k+1} dilanjutkan, tetapi jika tidak, maka dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

2.2.2 Prinsip Kerja Algoritma Runut Balik

Berikut ini adalah skema umum Algoritma Runut Balik:

```
procedure RunutBalik(input k:integer)
{Mencari semua solusi persoalan dengan metode runut balik
Masukan: k, yaitu indeks komponen vektor solusi, x[k]
Keluaran: solusi x = (x[1], x[2], ..., x[n])}
Algoritma:
  for tiap x[k] yang belum dicoba sedemikian sehingga
    (x[k] ← T(k)) and B (x[1], x[2], ..., x[k] = true do
  if (x[1], x[2], ..., x[k]) adalah lintasan dari akar ke daun
  then
    CetakSolusi(x)
  Endif
  RunutBalik(k+1){Tentukan nilai untuk x[k+1]}
  Endfor
```

Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti aturan pencarian mendalam (DFS). Simpul-simpul yang sudah dilahirkan dinamakan simpul hidup (*live node*). Simpul

hidup yang diperluas dinamakan simpul-E (*expand node*). Tiap kali simpul-E diperluas, lintasan yang dibangun olehnya bertambah panjang. Jika lintasan yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut “dibunuh” sehingga menjadi simpul mati (*dead node*). Fungsi yang digunakan untuk membunuh simpul-E adalah dengan menerapkan fungsi pembatas (*bounding function*). Simpul yang sudah mati tidak akan pernah diperluas lagi.

Jika lintasan berakhir dengan simpul mati, maka proses pencarian diteruskan dengan membangkitkan simpul anak lainnya. Bila tidak ada lagi simpul anak yang dapat dibangkitkan, maka pencarian solusi dilanjutkan dengan melakukan runut balik ke simpul hidup terdekat (simpul orang tua). Selanjutnya simpul ini menjadi simpul-E yang baru. Pencarian dihentikan bila telah ditemukan solusi atau tidak ada lagi simpul hidup untuk runut balik.

Dalam penelitian yang dilakukan Irwanti, 2010 algoritma *backtracking* digunakan dalam mencari anagram dari sebuah kata. Pada penelitian tersebut algoritma *backtracking* dipakai untuk membangun daftar kata *valid* yang merupakan kumpulan kata yang dapat disusun oleh huruf-huruf yang telah diacak dan terdapat di *database*. Program permainan yang dirancang Irwanti, 2010 melakukan tiga hal, yaitu memilih kata dari *database* secara acak, membangun daftar kata *valid* dari huruf-huruf kata yang terpilih, dan terakhir mencocokkan *string* masukan pemain dengan daftar yang telah dibuat [3].

Penelitian yang dilakukan Amin, 2007 juga menggunakan algoritma runut balik pada permainan teka-teki silang untuk mengisi kotak-kotak permainan yang sebelumnya telah dibuat. Kotak-kotak tersebut direpresentasikan sebagai struktur data matriks sehingga setiap kotak akan memiliki indeks. Program akan menghitung jumlah kotak pada deretan kotak kemudian mencari dan memilih kata (dalam *database*) yang memiliki jumlah karakter sama dengan jumlah kotak tersebut. Program akan mencatat dimana letak hubungan antar deretan kotak tersebut kemudian mencatat indeks dan mengambil karakter yang terdapat di dalamnya untuk dibandingkan kembali dengan deretan kata yang ada di dalam *database* kata. Jika kata yang dimasukkan berikutnya cocok maka pencarian akan dilanjutkan, namun jika

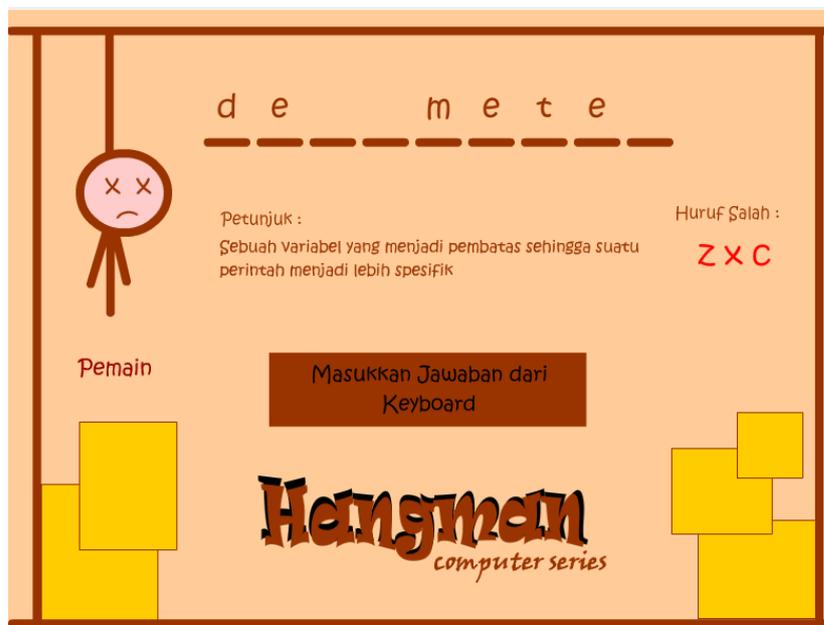
tidak terdapat kata yang cocok maka program akan mematikan kemungkinan jawaban berdasarkan pencarian tersebut dan program akan melakukan *backtrack* [8].

2.3 *Hangman*

Hangman adalah permainan menebak kata dengan media kertas dan pensil untuk dua atau lebih pemain. Salah satu pemain menentukan sebuah kata dan pemain lain mencoba menebaknya dengan mengajukan beberapa huruf.

"Asal-usul *Hangman* tidak jelas, tetapi tampaknya telah muncul di zaman Victoria," kata Tony Augarde, penulis buku "The Oxford Guide to Word Games" (Oxford University Press). Permainan ini disebutkan dalam buku Alice Bertha Gomme yang berjudul "Traditional Games" pada tahun 1894 dengan nama "Birds, Beasts and Fishes". Peraturannya sederhana, seorang pemain menuliskan huruf pertama dan terakhir dari sebuah kata untuk seekor binatang, dan pemain lain menebak huruf di antaranya. Sumber lain menyebut permainan ini "Gallows", "The Game of Hangin", atau "Hanger". *Hangman* telah muncul sejak tahun 1978 pada video game Speak dan Spell dengan nama "Mystery Word" dan kadang-kadang dimainkan saat ini sebagai permainan forum.

Hangman sebenarnya didesain agar terlihat seperti orang gantung diri. Meskipun muncul perdebatan tentang sejak kapan gambar ini dipergunakan, tetapi pemakaiannya masih sampai sekarang. Selain itu, aturan permainan *Hangman* bisa berbeda-beda. Beberapa pemain menggambar tiang gantungan sebelum bermain dan menggambar bagian tubuh manusia (seperti kepala, badan, tangan kiri, tangan kanan, kaki kiri dan kaki kanan satu per satu) saat permainan. Ada pula yang memulai tanpa gambar sama sekali untuk memberikan peluang lebih banyak kepada pemain lain untuk menebak. Jumlah bagian tubuh manusia juga dapat bervariasi, tergantung jumlah peluang yang akan diberikan. Alternatif lain yang membuat permainan sedikit lebih mudah yaitu dengan memberikan definisi dari kata yang ditebak sehingga memungkinkan juga bagi pemain untuk belajar bahasa.



Gambar 2.3 Bagan *Hangman* dengan Kata Rahasia "delimiter"

Dalam bahasa Inggris, ada 12 huruf yang paling sering digunakan, yaitu E, T, A, O, I, N, S, H, R, D, L dan U. Daftar frekuensi huruf ini digunakan oleh penebak untuk meningkatkan peluang ketika giliran mereka untuk menebak. Di sisi lain, daftar tersebut dapat digunakan oleh pembuat teka-teki untuk membingungkan lawan mereka dengan sengaja memilih kata yang menghindari huruf-huruf umum atau yang berisi huruf langka. Strategi lain yang umum adalah dengan menebak huruf vokal terlebih dahulu, karena bahasa Inggris hanya memiliki lima vokal (A, E, I, O dan U) dan hampir setiap kata setidaknya memiliki satu. Sedangkan untuk teks bahasa Indonesia huruf yang sering muncul adalah huruf A, N, I, E, L, T, R, D, S dan M [10].

Hangman umumnya dimainkan dengan aturan sebagai berikut:

1. Satu pemain memilih kata rahasia, kemudian menuliskan sejumlah garis sama banyak dengan panjang kata.
2. Pemain lain mulai menebak sebuah huruf. Jika pemain tersebut menebak dengan benar, maka pemain yang pertama akan mengungkapkan posisi huruf tersebut di dalam kata. Jika tidak, maka tebakan salah dan pemain pertama menggambarkan satu elemen dari bagan *Hangman*.
3. Permainan berakhir ketika semua huruf dalam kata telah terungkap atau bila penebak sudah kehabisan kesempatan untuk menebak.

Berikut ini contoh ilustrasi permainan *Hangman*. Seorang pemain akan menebak kata dengan menggunakan daftar frekuensi huruf dalam bahasa Inggris.



Kata: _ _ _ _ _

Tebakan: E

Salah:

(a)

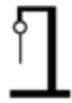


Kata: _ _ _ _ _

Tebakan: T

Salah: E

(b)



Kata: _ _ _ _ _

Tebakan: A

Salah: e,t

(c)



Kata: _ A _ _ _ A _

Tebakan: O

Salah: e,t

(d)



Kata: _ A _ _ _ A _

Tebakan: I

Salah: e,o,t

(e)



Kata: _ A _ _ _ A _

Tebakan: N

Salah: e,i,o,t

(f)



Kata: _ A N _ _ A N

Tebakan: S

Salah: e,i,o,t

(g)



Kata: _ A N _ _ A N

Tebakan: H

Salah: e,i,o,s,t

(h)



Kata: H A N _ _ A N

Tebakan: R

Salah: e,i,o,s,t

(i)



Kata: H A N _ _ A N

Tebakan:

Salah: e,i,o,r,s,t

(j)

Gambar 2.4 Contoh Ilustrasi Permainan *Hangman*