BAB II

LANDASAN TEORI

2.1 Pengertian Game

Game adalah suatu sistem yang memiliki aturan dengan suatu variabel dan hasil yang dapat dihitung, di mana hasil yang berbeda ditugaskan dengan nilai-nilai yang berbeda, dimana pemain melakukan usaha dalam mempengaruhi hasil perhitungan tersebut.

Game secara umum dibagi menjadi lima bagian diantaranya:

1. *Arcade/ Action, game* yang memiliki interaksi dengan manusia yang tinggi dan menggunakan graphis yang beragam, seperti: *game* Quake dan Freelancer.
2. *Puzzle*, *game* yang didalamnya terdapat unsur logika, seperti : *game* Tetris.
3. *Card*, *game* seperti Poker dan BlackJack.
4. *Strategy*, *game* yang memiliki suatu teknik dan pemikiran serta manajemen didalamnya seperti : game Command and Conquer dan Warcraft.
5. *Sport*, *game* seperti permainan olah raga yang sesungguhnya seperti *game* Fifa 2000.

Dalam *game* terdapat banyak unsur grafika komputer. Grafika komputer adalah seperangkat alat yang terdiri dari hardware dan software untuk membuat gambar, grafik atau citra realistik untuk seni, *game* komputer, foto dan film animasi.

Grafika tiga dimensi dapat menggambarkan sebuah benda dalam dunia nyata menjadi sebuah objek dari berbagai sudut pandang. Tidak seperti grafika dua dimensi, dalam grafika tiga dimensi banyak sekali hal yang harus dipertimbangkan.

2.2 Teka-Teki Silang

Pada 2 Desember 1913, Arthur Wynne TTS dalam majalah *New York World* dengan format seperti yang kita kenal saat ini. Teka-teki ini, yang bisa dilihat di situs web ini, sering disebut sebagai TTS pertama, dan Wynne sebagai penemunya.

TTS kemudian menjadi fitur mingguan di majalah tersebut. Buku kumpulan TTS pertama terbit pada 1924, diterbitkan oleh Simon and Schuster. Bukunya terbukti laris dan TTS menjadi salah satu benda terpopuler pada tahun 1924.

Pada tahun 1970-an di Jakarta terbit "Asah Otak", sebuah majalah TTS dan berbagai teka-teki lainnya. Penerbitan ini ternyata sukses sehingga banyak terbitan serupa yang segera mengikutinya.

Teka-teki silang merupakan game yang sangat populer dikalangan masyarakat Indonesia, Teka-teki Silang adalah suatu permainan yang berupa pengisian jawaban dalam kotak-kotak dimana pertanyaannya itu berupa mendatar dan menurun . permainan ini hanya bisa dimainkan oleh satu pemain. Dalam

permainan ini seorang pemain memiliki suatu tantangan karena harus mengisi kotak-kotak yang kosong dengan jawaban yang benar dan tepat. Permainan ini termasuk jenis permainan tebakan yang mendidik (*Educational Quiz*). Selain mempunyai tingkat popularitas tinggi, permainan ini juga memiliki unsur mendidik dan menuntut kreatifitas tinggi, baik saat pembuatan maupun saat memainkan.

2.3 MATRIK

2.3.1 Teori matrik

Matrik adalah himpunan skalar yang disusun secara empat persegi panjang (menurut baris dan kolom). Skalar – skalar itu disebut elemen matrik. Untuk batasnya biasanya digunakan: ( ), [ ], || ||Notasi Matrik Matrik diberi nama dengan huruf besar.

Operasi pada Matrik

1. Penjumlahan matrik

   Syarat : ukuran matrik harus sama. Jika A = (aij) dan B = (bij), matrik berukuran sama, maka A + B adalah suatu matrik C = (cij) dimana cij = aij + bij untuk setiap I dan j

2. Perkalian skalar terhadap matrik

Kalau μ suatu skalar (bilangan) dan A = (aij), maka matrik μ A = (?aij), dengan kata lain, matrik μ A diperoleh dengan mengalikan semua elemen matrik A dengan μ.

Hukum pada penjumlahan dan perkalian skalar :

Jika A, B, C adalah matrik berukuran sama, dan μ adalah skalar maka :

1. A + B = B + A (komutatif)
2. (A + B) + C = A + (B+C) (asosiatif)
3. μ (A + B) = μ A + μ B (distributif)
4. Selalu ada matrik D sedemikian hingga A + D = B

2.3.2 Perkalian matrik

Pada umumnya matrik tidak komutatif terhadap operasi perkalian : AB ? BA. Pada perkalian matrik AB, matrik A disebut matrik pertama dan B matrik kedua.

Syarat : Jumlah kolom matrik pertama = jumlah baris matrik kedua

Definisi :

Pandang A = (aij) berukuran (p x q) dan B = (bij) berukuran (q x r). Maka perkalian AB adalah suatu matrik C = (cij) berukuran (p x r) dimana :

cij = ai1 b1j + ai2 b2j + … + aiq bqj, untuk setiap i = 1,2,…,p dan j = 1,2,… r

Hukum pada perkalian matrik :

1. A(B + C) = AB + AC, dan (B + C) A = BA + CA, memenuhi hukum distributif
2. A(BC) = (AB)C , memenuhi hukum asosiatif
3. Perkalian tidak komutatif, AB ? BA
4. Jika AB = 0 (matrik 0) , yaitu matrik yang semua elemennya adalah = 0, kemungkinan kemungkinannya adalah :

   (i). A = 0 dan B = 0

   (ii) A = 0 atau B = 0

   (iii) A ? 0 dan B ? 0
5. Bila AB = AC belum tentu B = C

2.3.3 Transpose dari suatu Matrik

Pandang suatu matrik A = (a) berukuran (m x n) maka transpose dari A adalah matrik A^t berukuran (n x m) yang didapatkan dari A dengan menuliskan baris ke – i dari A, i = 1,2,…,m sebagai kolom ke –i dari $A^{t.}$ Dengan kata lain : A^t = (a).

Sifat – sifat matrik transpose :

1. $(A + B)^T = A^T + B^T$
2. $(A^T)^T = A$
3. $\mu(A^T) = (\mu A)^T$
4. $(AB)^T = B^T A^T$

2.3.4 Beberapa Jenis Matrik Khusus

Beberapa jenis matrik khusus sebagai berikut :

1. Matrik bujursangkar

   Adalah matrik dengan jumlah baris = jumlah kolom, sehingga disebut berordo n.

   Barisan elemen a11, a22, … ann disebut diagonal utama dari matrik bujursangkar A

2. Matrik nol

   Adalah matrik yang semua elemennya adalah 0

3. Matrik diagonal

   Matrik bujursangkar yang semua elemen di luar diagonal utamanya 0.

4. Matrik identitas

   Adalah matrik diagonal yang elemen – elemen diagonal utama adalah 1.

5. Matrik skalar

   Adalah matrik diagonal dengan semua elemen diagonal utamanyanya = k

6. Matrik segitiga bawah (lower triangular)

   Adalah matrik bujursangkar yang semua elemen di atas diagonal utama = 0.

7. Matrik segitiga atas (upper triangular)

   adalah matrik bujursangkar yang semua elemen di bawah diagonal utama = 0.

8. Matrik simetris

Adalah matrik yang transposenya sama dengan dirinya sendiri.

9. Matrik anti simetris

   Adalah matrik yang transposenya adalah negatifnya.

10. Matrik hermitian

    Adalah matrik yang bila transpose hermitiannya adalah sama dengan dirinya sendiri.

11. Matrik idempoten, nilpotent

    Bila berlaku A.A = A = A, maka A dikatakan matrik idempoten. R Bila A = 0, maka A nilpotent dengan index r (dimana r adalah bilangan bulat positif terkecil yang memenuhi hubungan tersebut)

2.4 Proses Pencarian Data

Jenis pencarian yang diimplementasikan pada aplikasi permainan teka-teki silang ini adalah jenis pencarian *String Matching*, pencarian ini adalah pencarian string didalam teks.

Persoalan pencarian *string* dirumuskan sebagai berikut:

Diberikan:

1. Teks (*text*), yaitu (*long*) *string* yang panjangnya *n* karakter
2. *Pattern*, yaitu *string* dengan panjang *m* karakter ($m < n$) yang akan dicari di dalam teks.

Carilah (*find* atau *locate*) lokasi pertama di dalam teks yang bersesuaian dengan *pattern*. Aplikasi dari masalah pencocokan string antara lain pencarian suatu kata di dalam dokumen (misalnya menu *Find* di dalam *Microsoft Word*).

Adapun beberapa contoh algoritma *string matching* adalah:

1. Algoritma *Brute Force*

   Dengan asumsi bahwa teks berada di dalam *array T*[1..*n*] dan *pattern* berada di dalam *array P*[1..*m*], maka algoritma *brute force* pencocokan *string* adalah sebagai berikut:

   1. Mula-mula *pattern P* dicocokkan pada awal teks *T*.
   2. Dengan bergerak dari kiri ke kanan, bandingkan setiap setiap karakter di dalam *pattern P* dengan karakter yang bersesuaian di dalam teks T sampai:
      a. semua karakter yang dibandingkan cocok atau sama (pencarian berhasil), atau
      b. dijumpai sebuah ketidakcocokan karakter (pencarian belum berhasil)
   3. Bila *pattern P* belum ditemukan kecocokannya dan teks *T* belum habis, geser *pattern P* satu karakter ke kanan dan ulangi langkah 2.

2. Algoritma Knuth-Morris-Pratt (KMP)

   Pada algoritma *brute force*, setiap kali ditemukan ketidakcocokan *pattern* dengan teks, maka *pattern* digeser satu karakter ke kanan. Sedangkan pada algoritma KMP, kita memelihara informasi yang digunakan untuk melakukan jumlah pergeseran. Algoritma menggunakan informasi tersebut untuk membuat pergeseran yang lebih jauh, tidak hanya satu karakter seperti pada algoritma *brute force*.

Dengan algoritma KMP ini, waktu pencarian dapat dikurangi secara signifikan. Algoritma KMP dikembangkan oleh D. E. Knuth, bersama-sama dengan J. H. Morris dan V. R. Pratt.

2.5 Enkripsi

Enkripsi merupakan pengacakan terhadap suatu data (plaintext), hingga data tersebut tidak dapat dikenali lagi bentuk aslinya (ciphertext). Sedangkan dekripsi merupakan pengembalian terhadap suatu data yang telah diacak sehingga data tersebut dapat kembali ke keadaan semula (plaintext).

Enkripsi adalah proses mengacak data sehingga tidak dapat dibaca oleh pihak lain. Pada kebanyakan proses enkripsi, Anda harus menyertakan kunci sehingga data yang dienkripsi dapat didekripsikan kembali. Ilmu yang mempelajari teknik enkripsi disebut kriptografi. Gambaran sederhana tentang enkripsi, misalnya mengganti huruf a dengan n, b dengan m dan seterusnya. Model penggantian huruf sebagai bentuk enkripsi sederhana ini sekarang tidak dipergunakan secara serius dalam penyembunyian data. ROT-13 adalah program yang masih suka dipergunakan. Intinya adalah mengubah huruf menjadi 23 huruf didepannya. Misalnya b menjadi o dan seterusnya. Pembahasan enkripsi akan terfokus pada enkripsi password dan enkripsi komunikasi data.

2.6 Rekayasa Perangkat Lunak

Perangkat lunak merupakan suatu program yang berisi sejumlah instruksi yang dibuat untuk melakukan sejumlah proses dengan tujuan sebagai alat pengendalian sistem komputer atau untuk menyelesaikan aplikasi tertentu.

Dalam membangun suatu perangkat lunak, selalu timbul beberapa kesulitan baik dari lingkungan intern maupun lingkungan ekstern. Untuk mengatasi kesulitan tersebut disusunlah suatu tata cara atau metodologi yang terangkum dalam Rekayasa Perangkat Lunak.

Rekayasa Perangkat Lunak adalah *suatu disiplin ilmu untuk membangun perangkat lunak, yang menggabungkan metode-metode untuk seluruh tahap-tahap pengembangan perangkat lunak, penentuan alat bantu yang baik untuk mengotomatisasikan metode-metode ini, pembuatan model perangkat lunak yang sangat berpengaruh untuk penerapan perangkat lunak, serta teknik-teknik yang baik untuk menjamin kualitas perangkat lunak.* [PRE 92]

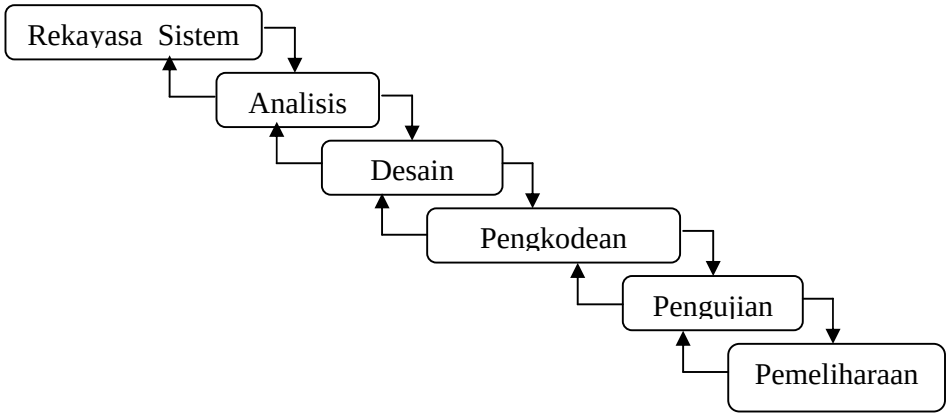
Untuk menghasilkan perangkat lunak yang baik maka :

a. Perangkat lunak yang dibuat harus mempunyai sifat yang mudah diperbaiki atau dikembangkan, sehingga dapat terus disesuaikan dengan perkembangan, baik dari teknologi maupun kebutuhan pengguna.
b. Perangkat lunak harus dapat diandalkan, yang mana kemampuan dari perangkat lunak diharapkan bekerja tanpa kesalahan.
c. Perangkat lunak harus bersifat efisien.
d. Perangkat lunak harus memberikan antarmuka yang tepat bagi pengguna.

2.7 Pendekatan Pembangunan Perangkat Lunak

Metodologi pembangunan perangkat lunak dikembangkan dan diterapkan agar dapat menghasilkan suatu perangkat lunak yang baik dan sesuai dengan kebutuhan pengguna.

Metodologi yang dipakai dalam pembangunan perangkat lunak teka-teki silang adalah pendekatan Siklus Kerja Sederhana (*Classic life Cycle*) yang biasa disebut juga model *Waterfall*. Siklus ini bersifat sistematik, dengan pendekatan sekuensial untuk membangun perangkat lunak, dimulai pada tingkat sistem dan berkembang sampai dengan pengujian (*testing*) dan perbaikan kesalahan (*maintenance*). [HAN 95]



Gambar 2.1

Siklus kerja Rekayasa Perangkat Lunak dengan Metode Waterfall

Keterangan mengenai tiap-tiap langkah pembuatan perangkat lunak adalah sebagai berikut :

1. **Rekayasa Sistem**

   Karena perangkat lunak selalu merupakan bagian dari sistem, maka harus diawali dengan menentukan kebutuhan-kebutuhan bagi seluruh elemen-elemen sistem, dan kemudian mengalokasikan beberapa *subset* dari kebutuhan-kebutuhan tersebut bagi perangkat lunak. Rekayasa sistem mencakup kumpulan kebutuhan-kebutuhan pada setiap tingkat sistem dengan jumlah yang kecil pada tingkat teratas perancangan dan analisis.

2. **Analisis**

   Merupakan proses pengumpulan kebutuhan perangkat lunak. Untuk memahami program yang akan dibangun, harus dipahami wawasan informasi dari perangkat lunak, fungsi yang dibutuhkan, performansi dan antarmuka. Beberapa target yang ingin dicapai antara lain :

   a. Mengetahui keinginan pengguna
   b. Efisiensi biaya, sumber daya dan waktu pembangunan perangkat lunak
   c. Ketersediaan dukungan teknik

3. **Desain**

   Perancangan perangkat lunak sebenarnya merupakan kumpulan proses yang difokuskan pada 4 (empat) atribut yang berbeda, yaitu struktur data, arsitektur perangkat lunak, rincian prosedur dan karakteristik antarmuka.

Tahap ini merupakan tahap penerjemahan dari keperluan atau data yang telah dianalisa ke dalam bentuk yang mudah dimengerti oleh pemakai.

Beberapa target dari tahap ini adalah :

a. Spesifikasi perangkat lunak harus dapat dimengerti oleh pembangun perangkat lunak.
b. Perancangan modul-modul memungkinkan digunakan kembali baik secara parsial maupun total.
c. Pembentukan dokumentasi perancangan perangkat lunak.

4. Pengkodean

Pada tahap ini perancangan harus diterjemahkan kedalam bentuk yang dapat dibaca oleh mesin. Yang menjadi target dari tahap ini adalah menuliskan program secara rinci pada setiap modul.

5. Pengujian

Proses pengujian difokuskan pada *logical internal* dari perangkat lunak untuk memastikan bahwa semua pernyataan-pernyataan telah diuji, dan pada *functional external* untuk menemukan kesalahan-kesalahan dan memastikan bahwa masukan sesuai dengan kebutuhan.

Yang menjadi target pada tahap ini adalah :

a. Mendeteksi setiap kesalahan yang terjadi
b. Memperbaiki setiap kesalahan yang terjadi
c. Mencegah kesalahan baru selama perangkat lunak dimodifikasi
d. Perangkat lunak dapat dipercaya / diandalkan.

6. **Pemeliharaan**

   Merupakan tahap perbaikan atau penyempurnaan perangkat lunak yang telah selesai akibat perubahan-perubahan atau penambahan yang terjadi sesuai kebutuhan atau permintaan pemakai. Untuk melakukannya, dipergunakan kembali setiap langkah pada siklus kerja terdahulu.

2.8 Pengembangan Perangkat Lunak

Secara umum, pengembangan perangkat lunak [MAH 96] terdiri dari 3 (tiga) tahap, yaitu :

1. **Definisi**

   Pada tahap ini dirumuskan apa yang akan diproses, fungsi dan performansi yang diinginkan, interface yang akan digunakan, serta prediksi kendala-kendala, batasan-batasan dan kriteria validasi yang diperlukan untuk menunjang keberhasilan sistem.

2. **Pengembangan**

   Pada tahap ini dijabarkan mengenai struktur data dan arsitektur perangkat lunak, prosedur rinci untuk penerapan, proses penerjemahan ke dalam pemrograman, serta perlakuan uji coba.

3. **Perubahan**

   Ada 3 (tiga) jenis perubahan atas perangkat lunak, yaitu :

   1. Perbaikan, yakni perubahan karena kekurangan.
   2. Adaptasi, yaitu perubahan yang mengacu terhadap lingkungan.
   3. Perluasan, yaitu perubahan karena permintaan pemakai.

2.9 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah teknik grafik yang digunakan untuk menjelaskan aliran informasi dan transformasi data yang bergerak dari masukan ke keluaran. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Teknik ini sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau akan disimpan.

DFD menjelaskan bagaimana data masukan diubah menjadi keluaran, dimana setiap bagian pada diagram menjelaskan proses transformasi yang berbeda. DFD dapat digunakan untuk menggambarkan sistem atau perangkat lunak pada banyak tingkatan pada dari suatu abstraksi. DFD juga dapat digunakan untuk menggambarkan sistem atau perangkat lunak pada setiap tingkat dan dapat dipecah-pecah.

DFD dapat dibagi ke dalam tingkatan-tingkatan yang menggambarkan bertambahnya aliran data dan rincian fungsional. Level 1 pada DFD menyatakan prosedur-prosedur dari keseluruhan sistem yang digambarkan pada model konteks (*Context Model*).

Model konteks pada DFD merupakan model sistem pokok yang menggambarkan keseluruhan seluruh elemen perangkat lunak sebagai sebuah lingkaran dengan data masukan dan data keluaran dinyatakan dengan anak panah masuk atau keluar, berturut-turut. Model konteks menjelaskan bagaimana gambaran sistem dan hubungan dengan pihak luar. Setiap proses dan jalur-jalur

aliran data digambarkan pada tingkat yang lebih tinggi (misalnya level 2) yang menyatakan proses-proses dari keseluruhan prosedur yang digambarkan pada DFD level 1.

Untuk membuat DFD, digunakan simbol-simbol. Simbol-simbol ini dirancang untuk mempermudah penggambaran kepada pemakai. Berikut ini adalah simbol-simbol menurut **Yourdan–DeMarco** :

Persegi empat : Menyatakan entitas luar (*external entity*), yaitu sebuah sistem atau sistem lain yang menerima / menghasilkan informasi dari / bagi perangkat lunak.

Lingkaran : Menyatakan proses yang menggunakan dan mengubahnya informasi ke dalam bentuk lain.

Anak panah : Menyatakan satu atau banyak item data. Seluruh anak panah harus diberi nama.

Garis ganda : Menyatakan tempat penyimpanan data (*data storage*), yang menyimpan isi dari aliran data yang digunakan oleh sistem atau perangkat lunak.

2.10 Perancangan Antarmuka

Antarmuka suatu perangkat lunak menggambarkan bagaimana cara sebuah program akan dilihat dan digambarkan oleh pemakai. Perancangan sebuah antarmuka meliputi perumusan tampilan layar dan struktur menu.

Secara umum, fungsi antar muka bagi pemakai adalah sebagai berikut :

a. Penyampaian pesan dari pemakai ke sistem dan sebaliknya.

b. Validasi oleh sistem terhadap pesan dari pemakai dan pengiriman pesan kesalahan dari sistem ke pemakai.

Beberapa pertimbangan dari pedoman dalam melakukan perancangan antar muka, antara lain :

a. Sederhana
b. Konsisten
c. Membatasi jumlah parameter dan perintah
d. Menyediakan nilai dasar (*default value*)
e. Memberikan kemudahan penanganan kesalahan
f. Dapat membatalkan suatu aksi
g. Dukungan kontrol oleh pemakai
h. Memiliki fasilitas pembatalan (*undo*)

Berikut ini adalah penjelasan dari beberapa komponen antar muka yang penting dan harus dirancang :

1. Perancangan Menu

Beberapa pedoman perancangan menu adalah sebagai berikut :

a. Konsisten
b. Terorganisir
c. Terurut
d. Dapat kembali ke menu awal
e. Tersedia alternatif cara pemilihan item
f. Mempertimbangkan luas layar

g. Mempertimbangkan visual menu dalam hal warna, bentuk maupun letaknya
h. Memberikan keterangan tentang masukan
i. Menyediakan bantuan langsung
j. Mempertimbangkan waktu tanggap dan kecepatan tampilan

2. Perancangan Pengisian Masukan

Beberapa pedoman dalam perancangan pengisian masukan adalah sebagai berikut :

a. Penggunaan istilah yang konsisten
b. Memperhatikan frekuensi pemakaian

3. Perancangan Fasilitas Bantuan Langsung Dan Pesan Kesalahan

Beberapa pedoman dalam perancangan fasilitas bantuan adalah sebagai berikut :

1. Singkat, jelas, konsisten dan mudah dipahami
2. Pemanggilan bantuan dapat dilakukan dari semua titik pada sistem setiap saat
3. Penayangan bantuan tidak mengganggu pekerjaan

Beberapa pedoman dalam perancangan pesan kesalahan adalah sebagai berikut :

a. Dimunculkan jika pemakai membuat kesalahan saat interaksi
b. Menjelaskan tentang jenis kesalahan dan cara mengatasinya

c. Pesan tidak terlalu umum

d. Lokasi dalam penayangannya harus konsisten

2.11 Bahasa Pemrograman Visual Basic

Dalam pembuatan sebuah program, visual basic menggunakan sistem yang disebut RAD (*Rapid Aplication Development*). Sistem ini memanfaatkan bahasa pemrograman visual yang mempermudah bagi pemakainya mendesain tampilan program (*user interface*). Cara ini sangat bermanfaat untuk membuat program yang bekerja dalam sistem windows yang memang tampilan layarnya lebih rumit dibandingkan dengan sistem DOS dulu. Dengan bahasa pemrograman biasa (non visual), waktu seorang programmer akan lebih banyak dihabiskan untuk mendesain atau memperindah tampilan program daripada menulis program utamanya sendiri.

Program Visual Basic dapat dikembangkan dan bekerja di lingkungan windows 98 atau lebih. Visual Basic menjadikan pemakai tidak perlu waktu untuk membuat tampilan kotak dilayar, dialog, perangkat kontrol yang rumit dan menyita waktu dalam menuliskan program. Beberapa keunggulan lainnya adalah kecepatan eksekusinya yang cepat, kemudahan akses berbagai format database, sehingga menjadikan penulis memilih Visual Basic untuk merancang program aplikasi dari kecerdasan buatan yang dibuat.