

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam dunia hiburan, kata “*game*” sudah tidak asing lagi ditelinga kita. *Game* juga banyak jenisnya, ada *game* yang dimainkan di *console game* dan ada yang dimainkan di PC (*Personal Computer*). Permainan komputer (bahasa Inggris: *computer game*) adalah permainan video yang dimainkan pada komputer pribadi, dan bukan pada konsol permainan, maupun mesin ding-dong. Permainan komputer telah berevolusi dari sistem grafis sederhana sampai menjadi kompleks dan mutakhir.

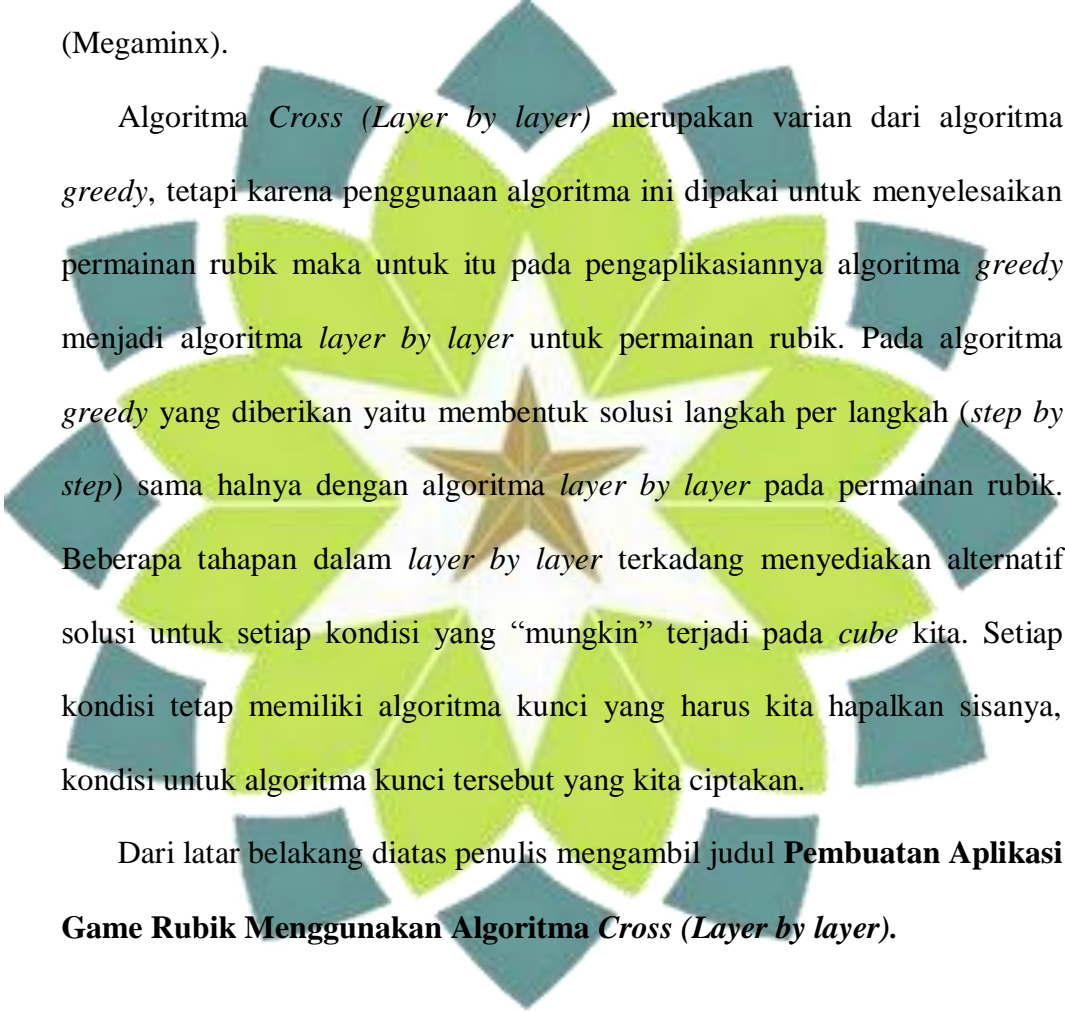
Kubus Rubik's adalah sebuah permainan teka-teki mekanik yang ditemukan pada tahun 1974 oleh pemahat dan profesor arsitektur Hungaria, Ernő Rubik. Kubus ini terbuat dari plastik dan terdiri atas 26 bagian kecil yang berputar pada poros yang terlihat. Setiap sisi dari kubus ini memiliki sembilan permukaan yang terdiri dari enam warna yang berbeda. Ketika terselesaikan/ terpecahkan, setiap sisi dari kubus ini memiliki satu warna dan warna yang berbeda dengan sisi lainnya.

Kubus ini dibuat kembali dan dipasarkan di kawasan eropa pada Mei, 1980. Rubik's tercatat sebagai mainan yang paling banyak terjual di dunia, dengan sekitar 300 juta kubus Rubik's termasuk imitasinya.

Ada berbagai variasi Cubes Rubik dengan sampai yaitu: $2 \times 2 \times 2$ (Rubik mini), rubik $3 \times 3 \times 3$ (standar), $4 \times 4 \times 4$ (Revenge Rubik/ Master Cube),

dan $5 \times 5 \times 5$ (Professor's Cube), $6 \times 6 \times 6$ (V-Cube 6), dan $7 \times 7 \times 7$ (V-Cube 7).

Terdapat beberapa perkembangan dalam ukuran dan bentuk geometris lainnya. Beberapa bentuk seperti termasuk tetrahedron/ rubik segi tiga (Pyraminx), rubik segi delapan (Skewb Diamond), rubik segi duabelas (Megaminx).



Algoritma *Cross (Layer by layer)* merupakan varian dari algoritma *greedy*, tetapi karena penggunaan algoritma ini dipakai untuk menyelesaikan permainan rubik maka untuk itu pada pengaplikasiannya algoritma *greedy* menjadi algoritma *layer by layer* untuk permainan rubik. Pada algoritma *greedy* yang diberikan yaitu membentuk solusi langkah per langkah (*step by step*) sama halnya dengan algoritma *layer by layer* pada permainan rubik. Beberapa tahapan dalam *layer by layer* terkadang menyediakan alternatif solusi untuk setiap kondisi yang “mungkin” terjadi pada *cube* kita. Setiap kondisi tetap memiliki algoritma kunci yang harus kita hapalkan sisanya, kondisi untuk algoritma kunci tersebut yang kita ciptakan.

Dari latar belakang diatas penulis mengambil judul **Pembuatan Aplikasi Game Rubik Menggunakan Algoritma *Cross (Layer by layer)***.

1.2. Rumusan Masalah

Melihat latar belakang masalah tersebut, maka masalah yang dibahas dapat dirumuskan sebuah pertanyaan sebagai berikut:

- a. Bagaimana membangun dan menerapkan permainan rubik dalam bentuk *software* komputer dengan menggunakan algoritma *Cross (Layer by layer)*?
- b. Bagaimana cara penyelesaian permainan tersebut dengan menggunakan waktu tercepat.
- c. Bagaimana aplikasi tersebut dibuat dalam bahasa pemrograman.

1.3. Tujuan

Berdasarkan rumusan masalah di atas, dapat disimpulkan bahwa penelitian ini bertujuan untuk :

- a. Membuat dan penerapan Aplikasi *Game* Rubik dalam bentuk aplikasi atau *software* komputer dengan algoritma *Cross (Layer by layer)*l.
- b. Membuat tutorial/ cara penyelsaian permainan rubik dengan waktu tercepat.
- c. Membuat aplikasi *game* rubik dengan bahasa pemrograman.

1.4. Batasan Masalah

Permasalahan yang ditemukan selama penelitian ini dibatasi oleh hal-hal yang tercantum berikut ini:

- a. Aplikasi *game* rubik yang dibuat adalah jenis rubik 3 x 3 x 3.
- b. Aplikasi yang dibuat terdapat tutorial/ *user manual* pemecahan masalah dari *game* tersebut.

- c. Aplikasi ini juga terdapat *Stopwatch* untuk mengukur kecepatan dalam menyelesaikan permainan.
- d. Aplikasi hanya untuk mengukur kemampuan seseorang (dimainkan 1 orang).

1.5. State of The Art

Pada contoh jurnal yang ditulis oleh Dody Irwin, Alumnus Fakultas Teknologi Industri, Jurusan Teknik Elektro, Universitas Kristen Petra (2001), menjelaskan perbandingan Algoritma *Hidden Surface Removal* dalam pembuatan suatu objek.

Dalam jurnal tersebut dijelaskan penggambaran yang dilakukan dengan komputer memungkinkan untuk menghasilkan gambar objek benda tiga dimensi yang menyerupai dengan bentuk yang sebenarnya. Objek tiga dimensi ini mempunyai kedalaman arah x, y dan z. Untuk menggambarkan objek tersebut secara nyata maka bagian yang tidak tampak dari titik pandang harus dihilangkan, permukaan objek yang berada di belakang permukaan objek yang lainnya harus disembunyikan. Bilamana ada dua permukaan bidang yang berpotongan maka bagian bidang yang tidak terlihat dari titik pandang juga harus disembunyikan.

Contoh jurnal kedua yang ditulis oleh Ina Agustina, Jurusan Sistem Informasi, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional (2009).

Pada jurnal tersebut, dijelaskan perbandingan dari beberapa jenis algoritma *Hidden Surface Removal*, yaitu algoritma *Scan Line* dan algoritma *Z-Buffer*.

Algoritma *Scan Line* lebih hemat dalam penggunaan memori dibandingkan dengan algoritma *Z Buffer*. Algoritma *scan Line* akan membutuhkan waktu lebih sedikit daripada algoritma *Z Buffer*, bila gambar yang dihasilkan mempunyai baris ($y_{maks}-y_{min}+1$) yang jauh lebih kecil dari maksimum tinggi bidang gambar atau gambar mengumpul untuk daerah y (pada suatu baris) dan hamper membentuk baris, karena hanya menganalisis dan menampilkan baris y_{min} sampai y_{maks} yang diperlukan saja dan arena ($y_{maks}-y_{min}+1$) yang kecil dan komponen waktu yang diperlukan untuk melalui suatu rangkaian link sebanyak ($y_{maks}-y_{min}+1$) akan menjadi kurang berpengaruh.

1.6. Metode Penelitian

Dalam penyusunan laporan tugas akhir ini, penulis menggunakan metodologi penelitian sebagai berikut:

1. Metode pengumpulan data

Dalam mengumpulkan data-data lapangan, penulis menggunakan beberapa metode pengumpulan data, yaitu :

a. Study literatur

1) Studi Pustaka

Dalam penyusunan laporan tugas akhir ini, penulis menggunakan beberapa buku sebagai bahan landasan teoritis untuk memperoleh suatu keterangan yang dapat menunjang penyusunan laporan tugas akhir ini.

2. Metode pengembangan perangkat lunak

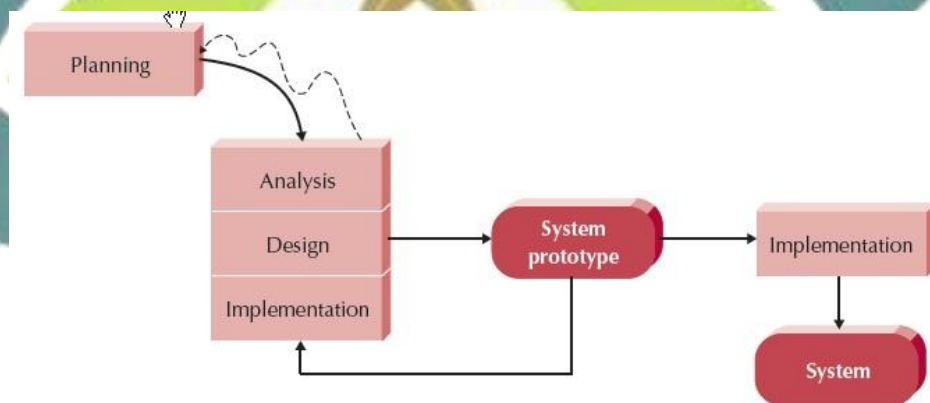
Dalam proses pengembangan perangkat lunak, metode yang digunakan adalah metode pengembangan *prototype*. Metode ini merupakan metode yang cepat dan cocok digunakan untuk aplikasi dengan *deadline* yang singkat.

Prototype dimulai dengan pengumpulan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan obyek keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui dan area garis besar di mana definisi lebih jauh merupakan keharusan kemudian dilakukan “perancangan kilat”. Perancangan berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai (contohnya pendekatan input dan format output). Perancangan kilat membawa kepada konstruksi sebuah *prototype*. *Prototype* tersebut dievaluasi oleh pelanggan atau pemakai dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. Iterasi terjadi pada saat disetel untuk memenuhi kebutuhan pelanggan

dan pada saat yang sama memungkinkan pengembang untuk secara lebih baik memahami apa yang harus dilakukannya. [Pressman, 2002]

Dengan model prototipe ini perancang dan pengguna bertemu untuk mendefinisikan secara obyektif keseluruhan perangkat lunak, mengidentifikasi kebutuhan yang diketahui, dan area lebih besar di mana definisi lebih jauh merupakan keharusan kemudian dilakukan perancangan kilat berupa maket atau prototipe sistem untuk kemudian dievaluasi pengguna untuk menyaring kebutuhan pengembangan perangkat lunak. Iterasi dimulai sampai kebutuhan pengguna terpenuhi dan pada saat yang sama memungkinkan pengembang untuk memahami apa yang harus dilakukan.

Tahapan-tahapan dalam *Prototyping* adalah sebagai berikut :



Gambar 1.1. Tahapan *Prototype*

1. Pengumpulan kebutuhan

Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun prototyping

Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output)

3. Evaluasi prototyping

Evaluasi ini dilakukan oleh pelanggan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak prototyping direvisi dengan mengulangi langkah 1, 2, dan 3.

4. Mengkodekan sistem

Dalam tahap ini prototyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai

5. Menguji sistem

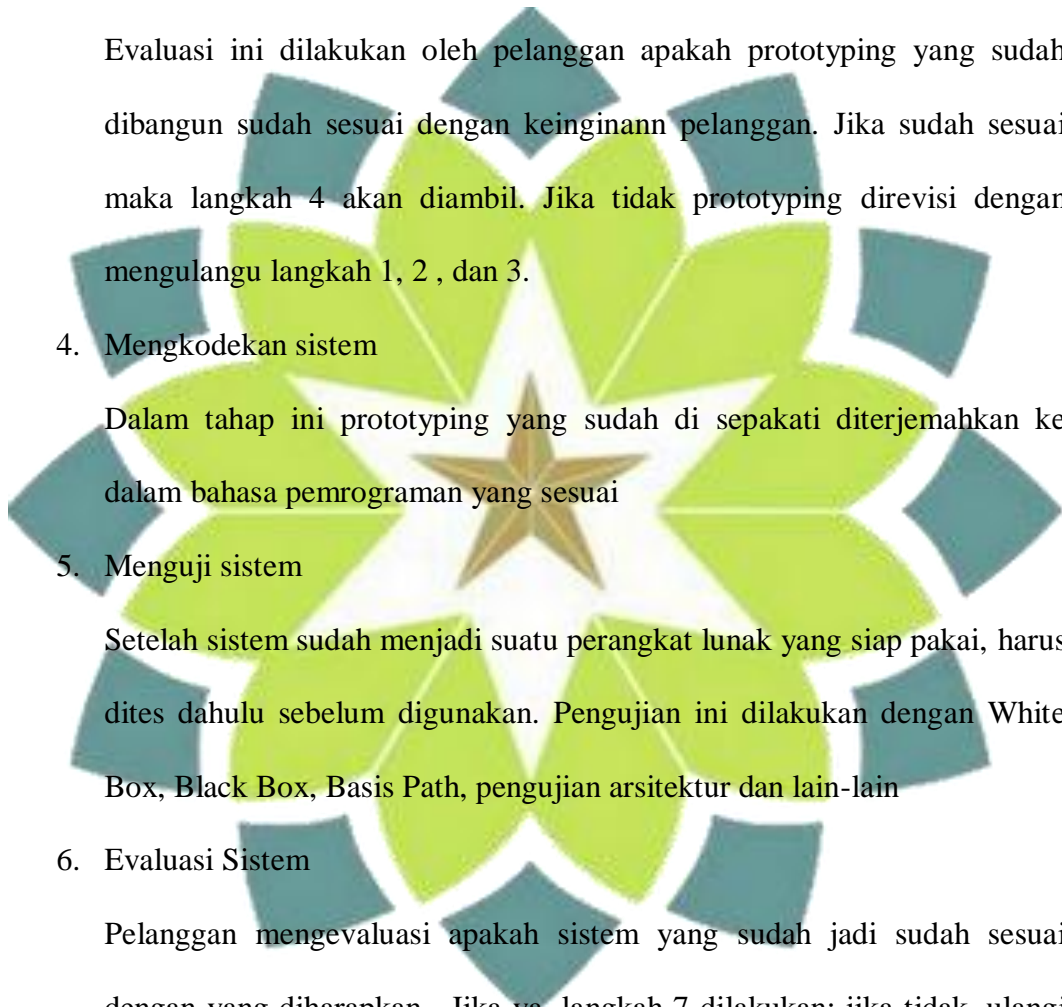
Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain

6. Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan 5.

7. Menggunakan sistem

Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.



1.7. Jadwal Penelitian

Tabel 1.1 Jadwal Penyusunan TA

No	Deskripsi Penelitian	November				Desember				Januari				Februari			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Pengumpulan kebutuhan			■	■	■	■										
2	Perancangan					■	■	■	■								
3	Evaluasi prototype									■	■	■	■	■	■		
4	Penyusunan laporan			■	■	■	■	■	■	■	■	■	■	■	■		

1.8. Sistematika Penyusunan

Sistematika Penyusunan Tugas Akhir ini adalah :

BAB I. PENDAHULUAN

Berisi tentang kajian secara global apa yang menjadi latar belakang masalah, rumusan masalah, tujuan, batasan masalah, *state of the art*, metode penelitian, jadwal penelitian dan sistematika penyusunan pada penyusunan tugas akhir ini.

BAB II. LANDASAN TEORI

Menjelaskan tentang teori-teori yang digunakan untuk menjelaskan dan menyelesaikan permasalahan yang akan dikaji.

BAB III. ANALISIS DAN PERANCANGAN

Bab ini juga membahas tentang rancangan aplikasi yang akan dibangun.

BAB IV. IMPLEMENTASI DAN PENGUJIAN

Implementasi sistem merupakan penerapan dari analisis dan desain sistem yang telah dilakukan menjadi suatu program aplikasi dengan menggunakan bahasa pemrograman tertentu.

BAB V. PENUTUP

Penutup merupakan bab terakhir yang berisikan tentang kesimpulan dari keseluruhan pembahasan yang serta pengajuan usulan atau saran-saran.

