BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Kakuro adalah permainan *puzzle* yang bersifat logika dan biasa disebut sebagai transliterasi matematis dari *puzzle* silang. Permainan *puzzle* kakuro merupakan inovasi dari permainan *puzzle* dengan menerapkan unsur matematika dan logika. Permainan ini adalah permainan penjumlahan angka dengan aturan mainnya yaitu dengan mengisi setiap kotak yang tersedia (kotak jawaban) dengan angka bulat dari 1 hingga 9 dengan beberapa syarat dalam pengisiannya. Soal-soal dalam permainan *puzzle* kakuro bersifat unik, karena soal-soal ini akan saling terkait baik dengan soal lainnya maupun jawaban dalam menyelesaikan permainan *puzzle* kakuro [5]. Untuk menyelesaikan permainan *puzzle* kakuro ini, jumlah pada nilai jawaban harus bernilai sama dengan nilai soal dan tidak boleh terdapat dua angka yang sama pada satu lajur (kolom maupun baris).

Oleh karena itu dibutuhkan teknik untuk membangkitkan soal yang bersifat unik dengan sistem *random* yang akan menghasilkan soal yang akan saling terkait dari soal satu dengan soal lainnya dan mencari solusi dalam menyelesaikan jawaban pada permainan *puzzle* kakuro dengan beberapa syarat didalamnya. Algoritma *backtracking* dapat menjadi pilihan dalam menyelesaikan permasalahan pada *puzzle* kakuro karena pada dasarnya algoritma ini berfungsi untuk mencari segala kemungkinan solusi dengan menerapkan ruang keadaan dalam bentuk pohon solusi hingga menghasilkan solusi yang layak. Secara umum algoritma *backtracking* dapat berfungsi dengan baik untuk memecahkan masalah-masalah yang berkembang secara dinamis *(dynamic problem solving)* sehingga menjadi dasar algoritma untuk kecerdasan buatan *(artificial intelligence)* [1]. Pencarian solusi pada algoritma *backtracking* dilakukan dengan menelusuri simpul-simpul di dalam pohon dengan teknik *Depth-First Search* (DFS).

Algoritma *backtracking* adalah algoritma yang berbasiskan *Depth-First Seacrh* (DFS). Mekanisme penyelesaian dengan menggunakan algoritma

backtracking berprinsip pada metode rekursif. Untuk menyelesaikan keseluruhan masalah, dibutuhkan sebuah solusi untuk permasalahan pertama kemudian permasalahan-permasalahan lainnya akan dicoba untuk diselesaikan secara rekursif berdasarkan solusi pertama. Apabila pada kemungkinan solusi yang dicari tidak ditemukan, maka dilakukan *backtrack* untuk menguji kemungkinan solusi selanjutnya. Proses *backtrack* akan selesai ketika tidak ada lagi solusi yang mungkin untuk diselesaikan pada permasalahan awal. Pada algoritma *baktracking* tidak perlu mencari kemungkinan-kemungkinan solusi yang ada sehingga kemungkinan-kemungkinan solusi yang ada akan dipangkas *(pruning)* [2].

Oleh karena itu, algoritma *backtracking* akan diterapkan pada permainan *puzzle* kakuro dengan permasalahannya dan mengambil pokok bahasan skripsi bagaimana menerapkan algoritma tersebut pada perangkat lunak *game* dengan judul "IMPLEMENTASI ALGORITMA BACKTRACKING PADA GAME PUZZLE KAKURO".

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang masalah diatas, didapatkan perumusan masalahnya adalah bagaimana mengimplementasikan algoritma *backtracking* pada *game puzzle* kakuro.

1.3 Maksud Dan Tujuan

Berdasarkan permasalahan yang diteliti, maka maksud dari penulisan tugas akhir ini adalah untuk mengimplementasikan algoritma *backtracking* pada *game puzzle* kakuro. Adapun tujuannya adalah sebagai berikut :

1. Membangkitkan nilai soal pada permainan dengan sistem *random* menggunakan algoritma *backtracking.*
2. Mendapatkan solusi penyelesaian dalam menemukan jawaban pada permainan *puzzle* kakuro dengan beberapa syarat dalam permainannya.

1.4 Batasan Masalah

Batasan masalah dalam pembangunan *game* ini adalah sebagai berikut :

1. *Game* berbasis desktop.
2. *Game* dibangun dengan desain 2D.
3. *Game* bersifat *offline.*
4. *Game* bergenre *puzzle game.*
5. Aplikasi yang dibangun dibatasi hanya untuk *puzzle* kakuro berukuran *grid* (9x9).
6. Algoritma yang digunakan untuk membangkitkan nilai soal dengan sistem *random* serta pencarian solusi jawaban pada permainana *puzzle* kakuro adalah menggunakan algoritma *backtracking*.
7. Data masukan untuk mengisi jawaban pada *game puzzle* kakuro diisi dengan angka bilangan bulat dari 1 hingga 9.
8. Kotak jawaban dibatasi hanya menampilkan 2 hingga 3 kotak jawaban dalam satu lajur (baris maupun kolom) pada *game puzzle* kakuro.
9. Metode analisis perancangan yang digunakan adalah analisis perancangan dan pengembangan perangkat lunak berorientasi objek, dengan menggunakan pemodelan *Unified Modeling Language* (UML).
10. Aplikasi yang digunakan untuk membangun *game* ini adalah :
    a. Aplikasi yang digunakan untuk membangun *game* adalah *greenfoot.*
    b. Bahasa pemograman yang digunakan adalah *java*.

1.5 Metode Penelitian

Metodologi penelitian yang digunakan adalah metodeologi analisis kualitatif. Analisis kualitatif adalah metode analisis yang meliputi observasi, dokumentasi, serta mendapatkan pemahaman yang mendalam tentang masalah-masalah yang sedang diteliti.

1.5.1 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah studi literatur, dengan cara mengumpulkan literatur, jurnal, *paper,* dan pencarian data yang bersumber dari internet berkaitan dengan aplikasi yang akan dibangun.

1.5.2 Metode Pembangunan Perangkat Lunak

Metode pembangunan perangkat lunak yang digunakan pada kasus ini adalah model *waterfall.* Model ini mengusulkan sebuah pendekatan kepada perkembangan aplikasi yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh *requirements definition, system and software design, implementation and unit testing, integration and system testing, operation maintenance*. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut :

1. *Requirements Definition.*

   Tahap *requirements definition* merupakan tahap pengumpulan kebutuhan data secara lengkap, kemudian dianalisis dan didefinisikan kebutuhan data tersebut sehingga terpenuhi oleh aplikasi yang akan dibangun. Fase ini harus terpenuhi secara lengkap untuk menghasilkan desain yang sesuai.

2. *System and Software Design.*

   Tahapan *system and software design* merupakan tahap dimana dilakukan penuangan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti diagram alir data (*data flow diagram*), diagram hubungan entitas (*entity relationship diagram*) serta struktur dan bahasan data.

3. *Implementation and Unit Testing.*

   Tahap *implementation and unit testing* merupakan tahap dimana penulisan kode program atau *coding* yang merupakan penerjemahan *design* dalam bahasa yang dapat dikenali oleh komputer. Dilakukan oleh *programmer* yang akan menterjemahkan transaksi yang diminta oleh pemain. Tahapan ini yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah

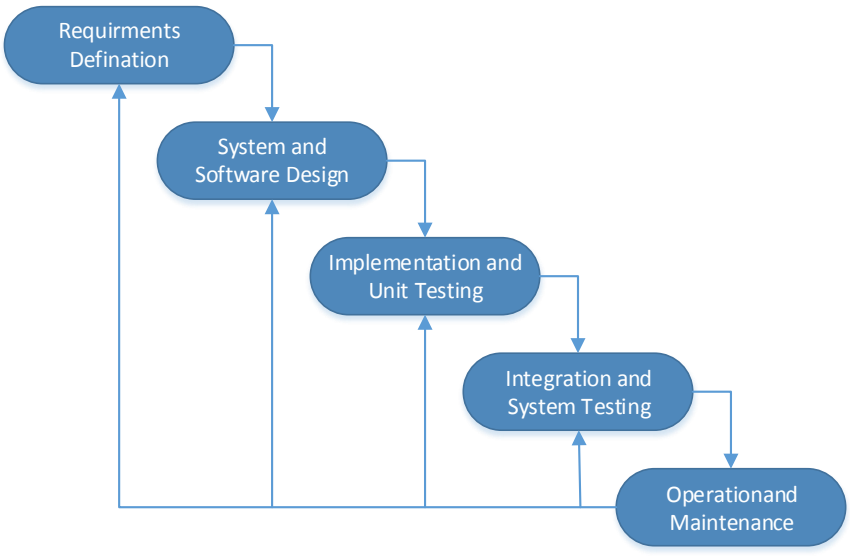
pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian diperbaiki.

4. *Integration and System Testing.*

   Tahap *integration and system testing* merupakan tahap dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna.

5. *Operationand Maintenance.*

   Tahap *operationand maintenance* merupakan tahap dimana penerapan dan pemeliharaan program secara normal tidak selalu diperlukan, tahap ini merupakan bagian siklus panjang. Sistem telah terpasang dan sedang dalam penggunaan. Pemeliharaan mencakup perbaikan kesalahan yang tidak ditemukan dalam tahap-tahap sebelumnya. Meningkatkan implementasi unit-unit sistem dapat mempertinggi pelayanan sistem yang disebabkan oleh kebutuhan baru.



Gambar 1.1 Metode pembangunan perangkat lunak *waterfall* [3]

1.6 Sistematika Penulisan

Sistematika penulisan laporan akhir penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penulisan tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi penjelasan latar belakang permasalahan pada *puzzle* kakuro dengan menerapkan algoritma *backtracking* untuk membangkitkan soal permainan dengan sistem *random* serta mendapatkan solusi penyelesaian jawaban pada permainan *puzzle* kakuro. Perumusan masalah yang didapat, dapat menjadi acuan untuk menghasilkan maksud dan tujuan pada *game puzzle* kakuro. Dikuti dengan metodologi penelitian serta sistematika penulisan untuk menggambarkan struktur penulisan yang lebih baik.

BAB II LANDASAN TEORI

Membahas berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Menganalisis masalah dari model penelitian, menggambarkan identifikasi masalah, menerapkan kedalam contoh kasus, *hardware, software,* diagram pembuatan sistem dan *UML (Unified Modeling Language).*

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Merupakan tahapan yang dilakukan dalam penelitian secara garis besar dimulai dari tahap persiapan sampai pada tahap penarikan kesimpulan, metode, dan kaidah yang diterapkan dalam penelitian. Pada bab ini juga, menerapkan kegiatan implementasi dan pengujian pada *game puzzle* kakuro menggunakan algoritma *backtracking.*

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang sudah diperoleh dari hasil yang di dapat pada implementasi algoritma *backtracking* pada *game puzzle* kakuro meliputi kecerdasan buatan atau *artificial intellegence* (AI), desain, dan *tools-tools* yang dikembangkan serta saran yang ditambahkan dalam implementasi algoritma *backtracking* pada *game puzzle* kakuro agar lebih baik.