Penerapan Algoritma Branch and Bound untuk Menemukan Jalur untuk Puzzle Ice Path pada Permainan Pokémon Crystal

Alifia Rahmah - 13520122
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13520122@std.stei.itb.ac.id

***Abstract*— Pokémon adalah permainan Pada permainan Pokémon Crystal, terdapat beberapa puzzle di dalam wilayah Ice Path. Puzzle ini terdiri dari sebuah lapangan es dan beberapa batu penghalang. Ketika pemain melangkah dalam lapangan es ini, pemain akan meluncur hingga bertabrakan dengan penghalang. Untuk dapat melewati puzzle ini, pemain harus memikirkan jalur sedemikian rupa dari awal melewati puzzle dalam Ice Path hingga mencapai titik tujuan. Solusi optimal untuk puzzle ini dapat dibangkitkan dengan menggunakan algoritma branch and bound, dengan cost untuk tiap simpul adalah jumlah dari langkah yang telah diambil dan jarak menuju titik tujuan.**

Kata kunci—Branch and Bound, Puzzle, Ice Path, Pokémon Crystal

I. PENDAHULUAN

Pokémon (Pocket Monsters) adalah sebuah *franchise* yang berpusat pada monster fiksi yang disebut Pokémon. *Franchise* ini terdiri dari permainan konsol, serial *anime*, film, *Trading Card Game* (TCG), *manga*, hingga *merchandise*. Permainan dalam seri utama Pokémon memiliki genre *Role-Playing Game* (RPG). Dalam permainan ini, pemain dapat mengoleksi beragam jenis monster yang disebut Pokémon, memperkuat Pokémon dengan melatihnya dalam pertarungan Pokémon, membuat Pokémon berevolusi, hingga berpartisipasi dalam Pokémon League dan menjadi Pokémon Champion.

Salah satu permainan Pokémon adalah Pokémon Crystal, yang merupakan salah satu dari permainan dari seri utama Pokémon. Pada permainan Pokémon Crystal, terdapat beberapa puzzle di dalam wilayah Ice Path. Puzzle ini terdiri dari sebuah lapangan yang terdiri dari lantai es dan beberapa batu penghalang. Ketika pemain melangkah dalam lapangan es ini, pemain akan meluncur hingga mencapai dinding penghalang. Untuk dapat melewati puzzle ini, pemain harus memikirkan jalur sedemikian rupa dari awal melewati puzzle dalam Ice Path hingga mencapai titik tujuan.

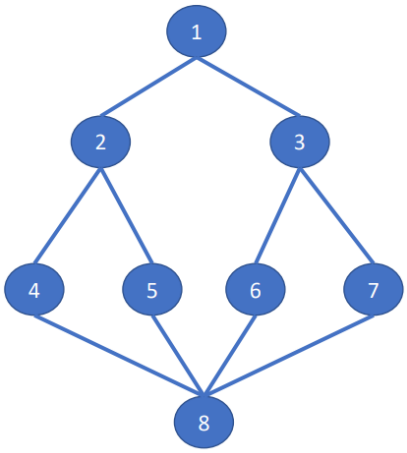
Solusi optimal untuk puzzle ini dapat dibangkitkan dengan menggunakan algoritma branch and bound. Algoritma branch and bound adalah algoritma pencarian rute yang digunakan untuk mencari rute optimal dalam suatu kasus. Dalam algoritma ini, setiap simpul memiliki cost masing-masing, yang merupakan nilai yang dicari secara heuristik. Nilai ini berupa taksiran bobot dari tiap rute yang dipilih. Algoritma ini membangkitkan rute dari simpul dengan *cost* paling optimal. Dalam perancangan solusi, akan dibuat program sederhana dengan masukan file berisi *layout* peta dari puzzle yang ingin dicari solusinya, dan output berupa arah yang harus diambil dalam tiap langkah menuju titik tujuan.

Gambar 1 Gambar *cartridge* Pokémon Crystal
(sumber: https://www.Pokémon.com/us/Pokémon-video-games/Pokémon-crystal-version/)

II. LANDASAN TEORI

A. *Algoritma Breadth First Search (BFS)*

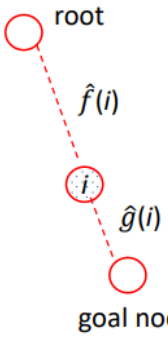
Algoritma breadth first search adalah salah satu metode traversal graf, yaitu mengunjungi simpul yang terhubung satu persatu dengan cara pencarian melebar. Dimulai dari simpul v, pencarian ini dilakukan dengan mengunjungi semua simpul yang bertetangga dengan simpul v terlebih dahulu, lalu mengunjungi simpul yang belum dikunjungi dan bertetangga dengan simpul yang tadi dikunjungi. Langkah tersebut dilakukan hingga mencapai simpul tujuan.



Gambar 2 Contoh urutan traversal dalam BFS
(sumber: Bahan Kuliah IF2211)

B. Algoritma Branch and Bound

Algoritma branch and bound adalah algoritma pencarian rute yang digunakan untuk persoalan optimasi, yang meminimalkan atau memaksimalkan suatu fungsi objektif, namun tetap tidak melanggar batasan (*constraint*) dalam persoalan. Pada algoritma ini, masing-masing simpul yang dibangkitkan memiliki *cost* yang dihitung berdasar nilai tertentu. Algoritma branch and bound merupakan penggabungan antara BFS (*breadth first search*) dan *least cost search*. Pada BFS, simpul yang akan dibangkitkan terurut berdasar urutan pembangkitannya (*First In First Out*/FIFO), sedangkan pada branch and bound, simpul yang akan dibangkitkan diurutkan dari simpul dengan *cost* paling optimal. *Cost* dalam simpul hidup dihitung dari estimasi *cost* termurah lingtasan dari simpul ke solusi, yang dihitung secara heuristik. *Cost* umumnya berupa taksiran dari nilai $c(i) = f(i) + g(i)$, dengan $c(i)$ ongkos untuk simpul i, $f(i)$ ongkos mencapai simpul i dari akar, dan $g(i)$ ongkos mencapai simpul tujuan dari simpul i.



Gambar 3 Ilustrasi penghitungan *cost* dengan taksiran $c(i) = f(i) + g(i)$
(sumber: Bahan Kuliah IF2211)

Pada algoritma branch and bound, dilakukan pemangkasan terhadap simpul yang dianggap tidak lagi mengarah ke solusi. Simpul yang dipangkas adalah simpul dengan nilai *cost* tidak lebih baik dari nilai terbaik sejauh ini, melanggar batasan yang ditentukan, atau solusi hanya terdiri dari satu titik.

C. Pokémon Crystal

Pokémon adalah sebuah *media franchise* yang berpusat pada monster fiksi yang disebut Pokémon. Pokémon berasal dari kata *pocket monsters*, yang berarti monster dalam saku. Hal ini merujuk pada monster Pokémon yang dapat dimasukkan ke dalam Pokeball (Pokémon ball) untuk disimpan dalam saku. *Franchise* ini terdiri dari permainan konsol, serial *anime*, film, *Trading Card Game* (TCG), *manga*, hingga *merchandise*. Permainan dalam seri utama Pokémon memiliki genre *Role-Playing Game* (RPG). Dalam permainan ini, pemain menjadi seorang trainer Pokémon dan dapat mengoleksi beragam jenis monster yang disebut Pokémon, memperkuat Pokémon dengan melatihnya dalam pertarungan Pokémon, membuat Pokémon berevolusi, hingga berpartisipasi dalam Pokémon League dan menjadi Pokémon Champion.

Gambar 4 Layar judul Pokémon Crystal
(sumber: https://www.tcrf.net)

Pokémon Crystal adalah salah satu permainan dari seri utama permainan Pokémon yang dikembangkan oleh Game Freak dan dirilis oleh Nintendo pada tahun 2000 di wilayah Jepang dan 2001 secara global untuk platform GameBoy Color. Pokémon Crystal merupakan salah satu game generasi kedua dari main series Pokémon, yang sejauh ini sudah memiliki 8 generasi permainan. Selain Pokémon Crystal, terdapat permainan generasi kedua lain yang rilis sebelum Pokémon Crystal, yaitu Pokémon Gold dan Pokémon Silver, yang rilis tahun 1999 di Jepang dan 2000 secara global. Alur permainan Pokémon Crystal seperti permainan Pokémon lainnya, yaitu mengoleksi Pokémon, berpetualang, melatih Pokémon hingga kuat, dan melakukan pertarungan hingga mendapat 8 *badge* untuk kemudian bertarung di Pokémon League dan menjadi Pokémon Champion.

Gambar 5 Skrinsut permainan Pokémon Crystal
(sumber: https://www.polygon.com/)

Selain bertarung, dalam permainan ini juga terdapat puzzle yang dapat ditemui sepanjang perjalanan. Pemain harus menyelesaikan puzzle tersebut untuk dapat melanjutkan perjalanan, atau mendapat hadiah berupa *item* yang membantu pemain, baik dalam perjalanan maupun untuk memperkuat Pokémon.

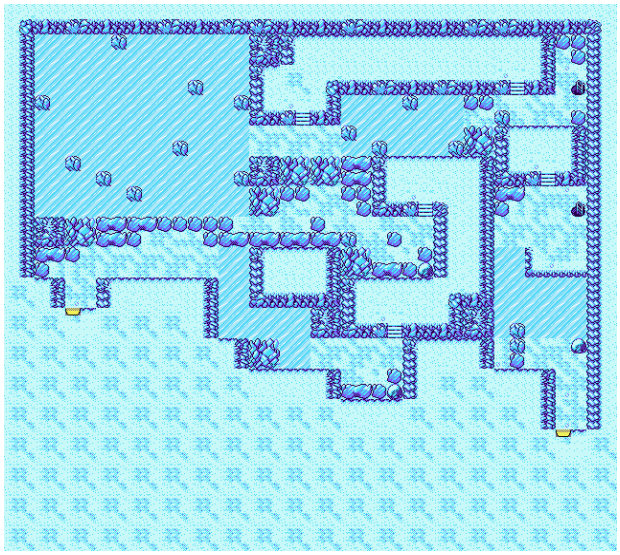
Pada tahun 2009-2010, telah dibuat remake dari Pokémon Gold/Silver/Crystal, yaitu Pokémon HeartGold dan Pokémon SoulSilver yang tersedia di platform Nintendo DS.

D. Ice Path

Ice Path adalah sebuah wilayah terowongan dalam permainan Pokémon Crystal yang menghubungkan dua wilayah, yaitu Route 44 dan Blackthorn City di region Johto. Untuk melanjutkan perjalanan, pemain harus melewati Ice Path ini. Dalam Ice Path, terdapat empat lantai, dengan lantai utama merupakan terowongan dari Route 44 ke Blackthorn City, dan tiga lantai lainnya berupa wilayah buntu dengan berbagai *item* yang tersebar di tiap lantai. Wilayah Ice Path terdiri dari salju dan bebatuan es.

Gambar 5 Wilayah luar Ice Path
(sumber:
https://strategywiki.org/wiki/Pokémon_Gold_and_Silver/Route_44)

Dalam beberapa wilayah dalam Ice Path, terdapat lapangan es. Jika lantai es ini dilewati, pemain akan meluncur dengan arah yang sama dengan arah saat melewati lantai ini hingga menabrak penghalang atau mencapai lantai salju. Penghalang ini dapat berupa batu es yang berada di tengah atau pinggir dari lapangan es, maupun dinding.



Gambar 6 Peta lantai pertama Ice Path
(sumber: https://bulbapedia.bulbagarden.net/wiki/Ice_Path)

E. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek adalah melakukan pemrograman dengan paradigma berorientasi objek. Dalam paradigma ini, elemen hasil abstraksi persoalan dibuat dalam bentuk objek dengan cetakan yang dibuat dari kelas tertentu. Dalam kelas, terdapat atribut, yaitu properti dimiliki dalam objek tersebut, serta method, yang mendeskripsikan bagaimana operasi atribut dalam objek.

F. Prority Queue

Priority queue adalah sebuah *abstract data type* (ADT) modifikasi dari struktur data queue, dengan tambahan prioritas dari tiap elemen. Ketika dimasukkan suatu elemen, elemen tersebut akan menyesuaikan posisi di dalam queue tersebut sesuai prioritasnya. Dalam beberapa bahasa pemrograman seperti Java dan C++, priority queue sudah termasuk ke dalam library dan siap digunakan, dengan penentuan prioritas yang dapat dimodifikasi.

III. IMPLEMENTASI

A. Rancangan Awal

Garis besar dari program untuk mencari jalur melewati puzzle dalam Ice Path adalah membangkitkan simpul kemungkinan arah gerak dari posisi mulai ke posisi akhir menggunakan algoritma branch and bound, lalu memilih simpul dengan *cost* paling kecil. Arah gerak yang dapat dilakukan adalah bergerak ke atas (UP), bergerak ke bawah (DOWN), bergerak ke kiri (LEFT), dan bergerak ke kanan (RIGHT), sehingga satu simpul dapat membangkitkan maksimal empat simpul dengan arah tersebut.

Cost dari tiap simpul dibangkitkan dengan menggunakan $c(i) = f(i) + g(i)$, dengan $f(i)$ berupa langkah yang telah dilakukan dari start hingga posisi saat ini, dan $g(i)$ berupa jarak Manhattan dari posisi saat ini hingga posisi akhir. Jarak

manhattan dihitung dengan menjumlahkan jarak horizontal dan jarak vertikal dari posisi awal ke posisi akhir.

Garis besar algoritma penyelesaian dalam program ini adalah dilakukan pembangkitan dari simpul awal, yang merupakan hasil pembacaan file input, ke empat arah yang memungkinkan, kemudian membangkitkan simpul yang masih hidup dengan *cost* paling kecil. Langkah tersebut diulang hingga terdapat simpul dengan posisi yang sudah mencapai posisi akhir.

B. Implementasi Program

Untuk membuat program ini, digunakan bahasa Java dengan paradigma berorientasi objek. Terdapat beberapa kelas yang akan dibuat, yaitu Main, Board, Node, NodeComparator, dan Solver. Terdapat juga enumerasi arah yaitu UP, DOWN, LEFT, dan RIGHT.

Kelas Board merepresentasikan *state* peta dan posisi pemain. Dalam kelas ini, akan disimpan peta berbentuk matriks karakter, posisi awal, posisi saat ini, dan posisi akhir. Dalam konstruktor dari kelas ini, posisi saat itu dideklarasikan dengan nilai yang sama seperti posisi awal (S). Selain atribut, terdapat juga method-method yang membantu perhitungan, di antaranya method untuk menghitung jarak dari posisi saat itu ke posisi akhir dan method untuk bergerak, atau memindahkan posisi saat ini sesuai arah yang diinginkan.

```
public class Board {
    public
ArrayList<ArrayList<Character≫ board;
    public int startRow;
    public int startCol;
    public int endRow;
    public int endCol;
    public int currentRow;
    public int currentCol;
    ...
    public void moveUp(){
          ...
    }
    public void moveDown() {
          ...
    }
    public void moveLeft() {
          ...
    }
    public void moveRight() {
          ...
    }
    ...
}
```

Kelas Node merepresentasikan simpul yang akan dibangkitkan untuk tiap langkah pencarian solusi. Dalam satu simpul, akan disimpan atribut Board yang direpresentasikan dalam simpul tersebut, *cost* dari simpul, kedalaman simpul, enumerasi arah yang dihasilkan dari parent ke simpul tersebut, serta simpul parent. Konstruktor awal dari Node menginisialisasikan *cost* dan kedalaman dengan nilai nol, sedangkan konstruktor simpul dari simpul parent yang digunakan untuk pembangkitan menggunakan nilai kedalaman simpul parent ditambah satu untuk kedalaman, dan nilai *cost* berupa penjumlahan kedalaman, yang merepresentasikan juga banyak langkah yang telah dilakukan, dan jarak dari posisi saat itu ke posisi akhir.

```
public class Node {
    public Board board;
    public int cost;
    public int depth;
    public Move moveFromParent;
    public Node parent;

    public Node(Board board) {
          ...
    }
    public Node(Board board, Move
moveFromParent, Node parent) {
          this.board = board;
          this.depth = parent.depth + 1;
          this.moveFromParent =
          moveFromParent;
          this.parent = parent;
          this.cost = parent.depth +
          board.distanceToEnd();
    }
}
```

Dalam algoritma penyelesaian, dibuat simpula awal berupa board awal hasil pembacaan file input. Kemudian dibangkitkan simpul anak sesuai arah yang dapat dilalui dari suatu posisi. Simpul hasil pembangkitan disimpan dalam struktur data priority queue, dengan pembanding berupa *cost* dari simpul. Priority queue ini menggambarkan urutan simpul yang akan membangkitkan masing-masing simpul anaknya. Dalam priority queue ini, semakin kecil *cost* simpul, simpul akan berada di posisi paling awal. Setelah semua simpul anak dari satu simpul sudah dibangkitkan, dilakukan penggantian simpul untuk dibangkitkan kembali, hingga posisi pemain dalam simpul yang akan dibangkitkan sudah mencapai posisi akhir.

```
public static ArrayList<Node>
getSolution(Board initialBoard) {
```

```
...

while (!currentNode.board.isAtEnd()) {

if (currentNode.board.canMoveUp()) {
      // tambahkan node ke atas
}

if (currentNode.board.canMoveDown()) {
      // tambahkan node ke bawah
}
   if
(currentNode.board.canMoveLeft()) {
      // tambahkan node ke kiri
}
   if
(currentNode.board.canMoveRight()) {
      // tambahkan node ke kanan
}
currentNode = aliveNode.poll();

        }

   // backtrack jalur solusi-parent

   while (currentNode.parent != null) {
      ...
   }

   return solution;
}
```

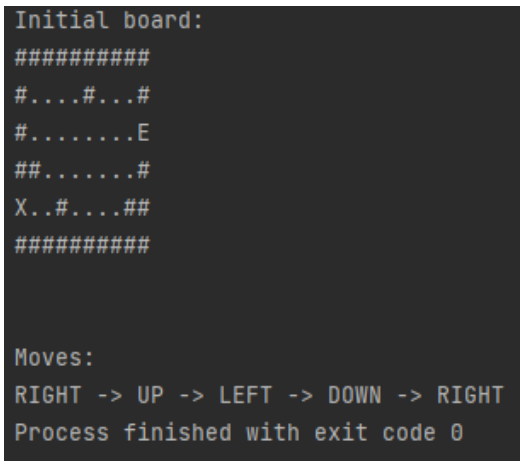
File *input* berupa struktur wilayah puzzle Ice Path dibuat dalam susunan karakter dalam file teks (.txt) dengan simbol-simbol berupa tagar (#) sebagai simbol penghalang, titik (.) sebagai simbol ruang kosong, huruf S sebagai posisi awal pemain, dan huruf E sebagai posisi akhir yang harus dicapai pemain.

C. Hasil Eksekusi

Untuk melakukan pengetesan terhadap program, dibuat testcase salah satu puzzle dalam Ice Path sebagai berikut.

```
##########
#....#...#
#........E
##.......#
S..#....##
##########
```

Dari testcase tersebut, program dijalankan dengan input file teks dengan isi tersebut.

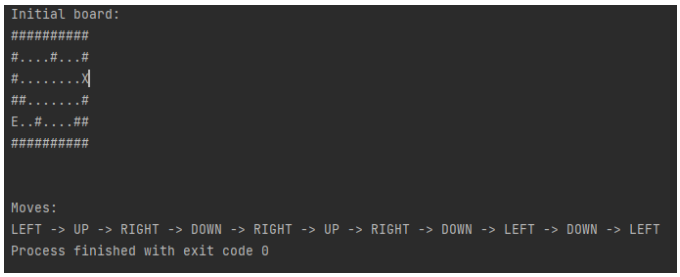


Gambar 7 Hasil eksekusi program dari testcase 1
(sumber: dokumentasi penulis)

Selain testcase tersebut, dibuat pula testcase untuk kembali ke posisi mulai dari posisi akhir dari testcase 1, dengan menukar posisi mulai dengan posisi akhir dan posisi akhir dengan posisi mulai.

```
##########
#....#...#
#........S
##.......#
E..#....##
##########
```

Dari testcase tersebut, program dijalankan dengan input file teks dengan isi tersebut, menghasilkan rute LEFT → UP → RIGHT → DOWN → RIGHT → UP → RIGHT → DOWN → LEFT → DOWN → LEFT.



Gambar 8 Hasil eksekusi program dari testcase 2
(sumber: dokumentasi penulis)

Dari hasil eksekusi secara keseluruhan, program ini sudah berjalan dengan lancar sesuai rancangan program, dengan keluaran berupa arah langkah yang dapat digunakan untuk melewati puzzle Ice Path dengan baik. Walaupun program ini terhambat untuk kasus layout puzzle yang lebih besar dan kompleks karena keterbatasan *memory space* saat pembangkitan simpul baru, namun untuk kasus tertentu, puzzle dapat diselesaikan dengan baik.

IV. KESIMPULAN

Berbagai kasus yang ditemukan dalam kehidupan sehari-hari dapat diselesaikan menggunakan strategi tertentu, dengan abstraksi ke permasalahan komputasi, lalu menggunakan

pendekatan algoritma tertentu untuk menyelesaikan permasalahaan tersebut. Salah satu permasalahan tersebut adalah penyelesaian puzzle dalam permainan, salah satunya Ice Path pada Pokémon Crystal. Dengan menggunakan algoritma branch and bound, dapat dibangkitkan rute berupa daftar arah untuk tiap langkah untuk menuju ke tujuan akhir dari puzzle ini.

V. Saran

Dengan makalah penyelesaian puzzle Ice Path ini, diharapkan proses perancangan program untuk menyelesaikan puzzle ini dapat dimanfaatkan dengan baik, dan dapat dikembangkan lagi. Selain menggunakan branch and bound, program ini dapat dimodifikasi dengan menggunakan algoritma pencarian rute lain seperti A-Star, Greedy Best First Search, atau Uniform Cost Search.

VI. Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa karena atas berkat rahmat dan karunia-Nya makalah yang berjudul "Penerapan Algoritma Branch and Bound untuk Menemukan Jalur untuk Puzzle Ice Path pada Permainan Pokémon Crystal" dapat diselesaikan dengan baik. Penulis juga berterima kasih kepada Ibu Nur Ulfa Maulidevi selaku dosen pengampu mata kuliah IF2211 Strategi Algoritma Semester II 2021/2022 Kelas K2.

Pranala Video

https://youtu.be/lNdeFsu--wM

Source Code

https://github.com/alifiarahmah/icepath-solver

Daftar Pustaka

[1] Munir, Rinaldi. 2021. *Breadth/Depth First Search (BFS/DFS)*. Bahan kuliah IF2211 Strategi Algoritma Semester II Tahun 2021/2022, Institut Teknologi Bandung.

[2] Munir, Rinaldi. 2021. *Algoritma Branch & Bound*. Bahan kuliah IF2211 Strategi Algoritma Semester II Tahun 2021/2022, Institut Teknologi Bandung.

[3] Pokémon Crystal Version | Video Games & Apps. The Official Pokémon Website | Pokémon.com | Explore the World of Pokémon. https://www.Pokémon.com/us/Pokémon-video-games/Pokémon-crystal-version/ (diakses tanggal 22 Mei 2022)

[4] Ice Path – Bulbapedia, the community-driven Pokémon encyclopedia https://bulbapedia.bulbagarden.net/wiki/Ice_Path (diakses tanggal 22 Mei 2022)

[5] Priority Queue. GeeksforGeeks. https://www.geeksforgeeks.org/priority-queue-set-1-introduction/ (diakses tanggal 22 Mei 2022)

[6] Manhattan distance [Explained]. OpenGenus IQ. https://iq.opengenus.org/manhattan-distance/ (diakses tanggal 22 Mei 2022)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Mei 2022

Alifia Rahmah
13520122