

Laporan Tugas Besar 1

IF2211 Strategi Algoritma

Pemanfaatan Algoritma *Greedy* dalam pembuatan *bot* permainan
Diamonds



Disusun oleh:

- | | |
|----------------------------------|----------|
| 1. Aurelius Justin Philo Fanjaya | 13522020 |
| 2. Francesco Michael Kusuma | 13522038 |
| 3. Maximilian Sulistiyo | 13522061 |

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2024

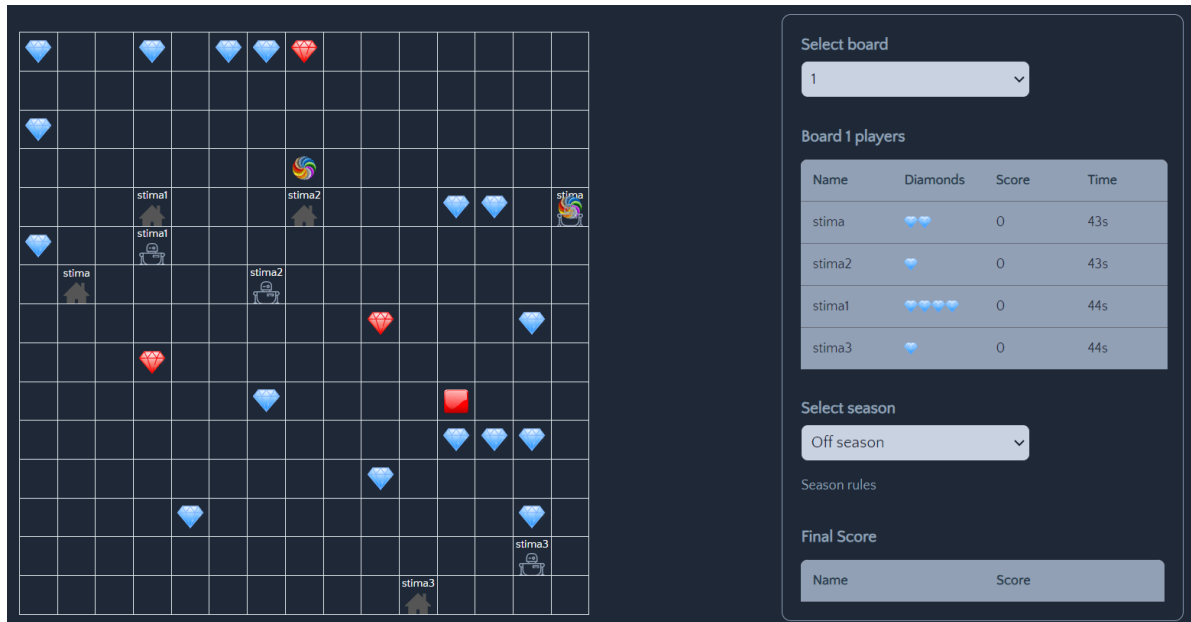
Daftar Isi

Daftar Isi.....	2
BAB 1	
Deskripsi Tugas.....	3
BAB 2	
Landasan Teori.....	6
1.1. Dasar Teori Algoritma Greedy.....	6
1.2. Cara Kerja Program.....	6
BAB 3	
Aplikasi Strategi Greedy.....	8
3.1. Proses Mapping Persoalan Menjadi Elemen Algoritma Greedy.....	8
3.2. Eksplorasi Alternatif Solusi Greedy.....	10
3.3. Analisis Efisiensi dan Efektivitas Alternatif Solusi Greedy.....	11
3.4. Strategi Greedy yang Dipilih.....	12
BAB 4	
Implementasi dan Pengujian.....	13
4.1. Implementasi Algoritma Greedy pada Program Bot yang digunakan dengan.....	13
4.2. Penjelasan Struktur Data yang Digunakan dalam Program Bot Diamonds.....	16
4.3. Analisis dan Pengujian.....	16
BAB 5	
Kesimpulan dan Saran.....	18
5.1 Kesimpulan.....	18
5.2 Saran.....	18
Daftar Pustaka.....	19

BAB 1

Deskripsi Tugas

Diamonds merupakan suatu programming challenge yang mempertandingkan bot yang anda buat dengan bot dari para pemain lainnya. Setiap pemain akan memiliki sebuah bot dimana tujuan dari bot ini adalah mengumpulkan diamond sebanyak-banyaknya. Cara mengumpulkan diamond tersebut tidak akan sesederhana itu, tentunya akan terdapat berbagai rintangan yang akan membuat permainan ini menjadi lebih seru dan kompleks. Untuk memenangkan pertandingan, setiap pemain harus mengimplementasikan strategi tertentu pada masing-masing bot-nya. Penjelasan lebih lanjut mengenai aturan permainan akan dijelaskan di bawah.



Pada tugas pertama Strategi Algoritma ini, mahasiswa diminta untuk membuat sebuah bot yang nantinya akan dipertandingkan satu sama lain. Tentunya mahasiswa harus menggunakan strategi greedy dalam membuat bot ini.

Program permainan Diamonds terdiri atas:

1. Game engine, yang secara umum berisi:
2. Kode backend permainan, yang berisi logic permainan secara keseluruhan serta API yang disediakan untuk berkomunikasi dengan frontend dan program bot

3. Kode frontend permainan, yang berfungsi untuk memvisualisasikan permainan
4. Bot starter pack, yang secara umum berisi:
5. Program untuk memanggil API yang tersedia pada backend
6. Program bot logic (bagian ini yang akan kalian implementasikan dengan algoritma greedy untuk bot kelompok kalian)
7. Program utama (main) dan utilitas lainnya

Komponen-komponen dari permainan Diamonds antara lain:

1. Diamonds

Untuk memenangkan pertandingan, kita harus mengumpulkan diamond ini sebanyak-banyaknya dengan melewati/melangkahnya. Terdapat 2 jenis diamond yaitu diamond biru dan diamond merah. Diamond merah bernilai 2 poin, sedangkan yang biru bernilai 1 poin. Diamond akan di-regenerate secara berkala dan rasio antara diamond merah dan biru ini akan berubah setiap regeneration.

2. Red Button/Diamond Button

Ketika red button ini dilewati/dilangkahi, semua diamond (termasuk red diamond) akan di-generate kembali pada board dengan posisi acak. Posisi red button ini juga akan berubah secara acak jika red button ini dilangkahi.

3. Teleporters

Terdapat 2 teleporter yang saling terhubung satu sama lain. Jika bot melewati sebuah teleporter maka bot akan berpindah menuju posisi teleporter yang lain.

4. Bots and Bases

Pada game ini kita akan menggerakkan bot untuk mendapatkan diamond sebanyak banyaknya. Semua bot memiliki sebuah Base dimana Base ini akan digunakan untuk menyimpan diamond yang sedang dibawa. Apabila diamond disimpan ke base, score bot akan bertambah senilai diamond yang dibawa dan inventory (akan dijelaskan di bawah) bot menjadi kosong.

5. Inventory

Bot memiliki inventory yang berfungsi sebagai tempat penyimpanan sementara diamond yang telah diambil. Inventory ini memiliki kapasitas maksimum sehingga sewaktu waktu bisa

penuh. Agar inventory ini tidak penuh, bot bisa menyimpan isi inventory ke base agar inventory bisa kosong kembali.

Untuk mengetahui flow dari game ini, berikut ini adalah cara kerja permainan Diamonds.

1. Pertama, setiap pemain (bot) akan ditempatkan pada board secara random. Masing-masing bot akan mempunyai home base, serta memiliki score dan inventory awal bernilai nol.
2. Setiap bot diberikan waktu untuk bergerak, waktu yang diberikan semua sama untuk setiap pemain.
3. Objektif utama bot adalah mengambil diamond-diamond yang ada di peta sebanyak-banyaknya. Seperti yang sudah disebutkan di atas, diamond yang berwarna merah memiliki 2 poin dan diamond yang berwarna biru memiliki 1 poin.
4. Setiap bot juga memiliki sebuah inventory, dimana inventory berfungsi sebagai tempat penyimpanan sementara diamond yang telah diambil. Inventory ini sewaktu-waktu bisa penuh, maka dari itu bot harus segera kembali ke home base.
5. Apabila bot menuju ke posisi home base, score bot akan bertambah senilai diamond yang tersimpan pada inventory dan inventory bot akan menjadi kosong kembali.
6. Usahakan agar bot anda tidak bertemu dengan bot lawan. Jika bot A menempa posisi bot B, bot B akan dikirim ke home base dan semua diamond pada inventory bot B akan hilang, diambil masuk ke inventory bot A (istilahnya tackle).
7. Selain itu, terdapat beberapa fitur tambahan seperti teleporter dan red button yang dapat digunakan apabila anda menuju posisi objek tersebut.
8. Apabila waktu seluruh bot telah berakhir, maka permainan berakhir. Score masing-masing pemain akan ditampilkan pada tabel Final Score di sisi kanan layar.

BAB 2

Landasan Teori

1.1. Dasar Teori Algoritma Greedy

Algoritma Greedy merupakan salah satu metode yang paling populer dan sederhana untuk memecahkan masalah optimasi. Pengertian dari Algoritma Greedy sendiri adalah pendekatan dalam pemrograman untuk memecahkan persoalan optimasi dengan cara yang tampaknya rakus (*Greedy*). Pendekatan dari Algoritma Greedy memfokuskan algoritma untuk mengambil keputusan optimal pada tiap langkahnya dengan harapan bahwa setiap langkah akan membawa kita lebih dekat ke solusi akhir yang optimal.

Algoritma Greedy akan mengambil solusi optimal pada setiap langkahnya tanpa mempertimbangkan konsekuensi di masa depan. Oleh sebab itu, Algoritma Greedy tidak selalu menghasilkan solusi optimal pada beberapa persoalan seperti pada *Coin Exchange Problem*. Tapi, pada persoalan-persoalan tertentu seperti persoalan *Minimum Spanning Tree*, Algoritma Greedy akan selalu menghasilkan solusi optimal dengan pendekatan Algoritma Kruskal atau Prim.

1.2. Cara Kerja Program

Pada game ini, program dibagi menjadi 2 bagian besar yaitu bagian Game Engine dan Bot. Kedua bagian program ini dapat berkomunikasi menggunakan API yang telah tersedia pada program. Pada bagian Bot, program akan memanggil API yang tersedia pada backend. Bot melakukan POST request terhadap endpoint API untuk mendaftarkan dirinya berdasarkan email dan password. Setelah terdaftar, bot dapat bergabung/join ke dalam permainan dengan melakukan POST terhadap endpoint join dan mendapatkan informasi dari board. Berdasarkan informasi tersebut, Bot dapat melakukan kalkulasi move berdasarkan logic masing-masing bot dan mengirimkannya melalui POST request ke backend melalui *endpoint move*. Backend akan memberikan response dengan body berisi kondisi board setelah move tersebut, dan bot kembali mengkalkulasi move selanjutnya dan diulang sampai permainan selesai.

Algoritma Greedy dapat kita implementasikan ke dalam bot dengan membuat module berupa file `‘.py’` pada folder logic, kemudian meng-*import* fungsi dalam module tersebut ke file `main.py`. Dalam file `main.py`, fungsi dimasukkan ke dalam dictionary CONTROLLER dengan nama key tertentu. Kemudian, untuk menjalankan bot tersebut, kita menjalankan main menggunakan:

```
python main.py --logic {KEY} --email={EMAIL} --name={USERNAME} --password={PASSWORD}
--team etimo &
```

Dengan KEY adalah key dari fungsi logic yang terdapat pada CONTROLLER di main.py, EMAIL dan USERNAME yang unik dan password dari akun tersebut.

BAB 3

Aplikasi Strategi Greedy

3.1. Proses *Mapping* Persoalan Menjadi Elemen Algoritma Greedy

3.1.1. Himpunan Kandidat

Pada permasalahan ini terdapat beberapa objek yang dapat dimasukkan ke dalam himpunan kandidat yaitu

Nama Objek	Penjelasan
Base	Objek ini merupakan markas dari bot dimana jika bot kembali ke base membawa N diamonds, maka bot ini akan mendapatkan N points sesuai dengan banyaknya diamonds. Kemudian base juga merupakan tempat muncul kembalinya bot jika bot ter-tackle oleh bot lain.
Red Button	Jika objek ini dilewati oleh bot maka semua diamond yang terletak di board akan generate ulang dan diacak posisinya, posisi dari red button pun akan diacak setelah dilewati.
Diamond	Jika objek ini dilewati oleh bot maka bot akan menambahkan diamond sebanyak N ke dalam inventory-nya. Jumlah N tergantung dengan tipe diamond dimana jika bot mengambil red diamond maka N adalah dua dan jika bot mengambil blue diamond maka N adalah satu
Teleporter	Pada setiap game hanya terdapat dua teleporter pada setiap saat. Jika objek ini dilewati oleh bot maka bot akan “masuk” ke dalam teleporter A dan akan muncul pada teleporter B.
Inventory	Inventory merupakan tempat bot menyimpan diamonds yang terambil oleh bot selama keberjalanan game. Inventory memiliki batasan slot 5 diamonds dimana jika kita mengambil diamond saat slot sudah 5 maka diamond tidak akan terambil dan bot pun harus kembali ke base untuk mengosongkan inventory.
Bot Musuh	Pada setiap game terdapat bot musuh yang ikut bermain. Bot musuh memiliki sifat yang sama dengan bot kita dalam hal base, teleporter, dan diamonds, dan inventory. Mereka juga dapat

	men-tackle kita dengan cara melewati bot kita. Jika kita ter-tackle maka semua diamond pada inventory kita akan terambil oleh bot yang menyerang kita. Sebaliknya juga kita dapat men-tackle bot musuh dan mendapatkan semua diamond yang terdapat pada inventory mereka
--	--

3.1.2. Himpunan Solusi

Dengan pemilihan objek-objek tersebut maka dapat dikatakan bahwa himpunan solusi dari algoritma ini adalah path yang diambil oleh bot.

3.1.3. Fungsi Solusi

Algoritma ini memeriksa jarak terdekat untuk ke Base, RedButton, dan Diamond dan memilih berdasarkan syarat pada fungsi seleksi

3.1.4. Fungsi Seleksi

Seleksi tujuan dalam permainan dapat disimpulkan sebagai berikut :

Nama Objek	Syarat
Base	Bot akan menjadikan base sebagai tujuan jika: <ol style="list-style-type: none"> 1. Inventory penuh, bot membawa 5 diamond 2. Membawa lebih dari 2 diamond dan jarak base kurang dari 2 3. Membawa diamond dan batas waktu kurang dari 10 detik
Red Button	Bot akan menjadikan red button sebagai tujuan jika jumlah diamond yang tersisa pada board kurang dari sama dengan 6 dan jarak dari red button kurang dari jarak diamond terdekat.
Diamond	Terdapat beberapa diamond pada board namun diamond yang dijadikan tujuan oleh bot adalah diamond terdekat.
Teleporter	Teleporter dijadikan tujuan dari bot jika: <ol style="list-style-type: none"> 1. Jarak dari bot ke teleporter A ditambah jarak dari teleporter B ke diamond lebih dekat dibandingkan jarak langsungnya 2. Jarak dari bot ke teleporter A ditambah jarak dari teleporter B ke base lebih dekat dibandingkan jarak langsungnya
Bot Musuh	Bot musuh akan dijadikan tujuan oleh bot jika:

	<ol style="list-style-type: none"> 1. Bot musuh membawa lebih dari 2 diamond dan jarak dengan bot musuh kurang dari 3 2. Bot musuh terletak tepat 1 satuan dari bot kita
--	--

3.1.5. Fungsi Kelayakan

Algoritma ini akan memeriksa apakah inventory sudah penuh dan juga apakah waktu permainan kurang dari sama dengan 10 detik.

3.1.6. Fungsi Objektif

Algoritma ini memaksimumkan jumlah poin yang didapatkan oleh bot

3.2. Eksplorasi Alternatif Solusi Greedy

Beberapa Alternatif Solusi Greedy yang kami eksplorasi untuk persoalan ini adalah sebagai berikut.

3.2.1. Greedy by Distance

Greedy by Distance adalah Algoritma Greedy dengan mencari Diamond mana yang memiliki jarak paling dekat terhadap bot. Jarak yang digunakan pada persoalan ini adalah *Manhattan Distance*, yaitu $|X_{\text{Diamond}} - X_{\text{Bot}}| + |Y_{\text{Diamond}} - Y_{\text{Bot}}|$ yang merepresentasikan jumlah langkah yang dibutuhkan oleh bot untuk sampai ke diamond tersebut.

3.2.2. Greedy by Point

Greedy by Point adalah Algoritma Greedy dengan mencari Diamond mana yang memiliki point paling besar pada board. Karena dalam permainan ini hanya terdapat 2 kemungkinan nilai point Diamond, yaitu 1 point (Blue Diamond) dan 2 point (Red Diamond), maka bot akan menuju Red Diamond dengan jarak terdekat dengan bot selama masih ada Red Diamond pada board. Setelah Red Diamonds habis, baru bot menuju ke Blue Diamond terdekat sampai muncul kembali Red Diamond pada Board.

3.2.3. Greedy by Point per Distance

Greedy by Point per Distance adalah Algoritma Greedy dengan mencari Diamond mana yang memiliki nilai Point/Distance paling besar terhadap bot. Bot akan menuju ke Diamond dengan nilai Point/Distance yang paling besar.

3.2.4. Greedy by Tackling

Greedy by Point per Tackling adalah Algoritma Greedy dengan mencari Bot lawan yang memiliki Diamond dan berjarak terdekat dengan Bot. Bot akan menuju ke posisi bot lawan dan mencoba meng-*tackle* bot lawan tersebut.

3.2.5. Greedy by Point per Area

Greedy by Point per Area adalah Algoritma Greedy dengan mencari Area dengan total point pada Area tersebut yang paling besar. Bot akan menghitung jumlah point pada area yang telah dibagi-bagi pada board, kemudian menuju area dengan point total paling besar.

3.2.6. Greedy by Base Distance

Greedy by Base Distance adalah Algoritma Greedy dengan mencari Diamond dengan jarak yang dekat dengan base. Tujuan dari pendekatan Algoritma Greedy ini adalah untuk mencegah bot bergerak terlalu jauh dari base sehingga tidak memerlukan waktu lama untuk kembali ke base dan menyimpan diamond menjadi point.

3.3. Analisis Efisiensi dan Efektivitas Alternatif Solusi Greedy

No .	Alternatif Solusi (Greedy by)	Kelebihan	Kekurangan
1.	Distance	<ul style="list-style-type: none">• Semakin dekat diamond, semakin banyak diamond yang dapat diperoleh dalam satu permainan.• Bot bergerak semakin jauh dari base jika diamond sudah sedikit.	<ul style="list-style-type: none">• Bot melewatkan diamond merah (2 point) padahal perbedaan jarak sangat kecil.
2.	Point	<ul style="list-style-type: none">• Bot akan memprioritaskan mengambil diamond merah daripada diamond biru.	<ul style="list-style-type: none">• Bot tetap mengambil diamond berwarna merah walaupun jaraknya sangat jauh.• Diamond merah seringkali berjarak jauh satu sama lain sehingga menghabiskan waktu yang tidak seimbang dengan point yang didapatkan.
3.	Point per Distance	<ul style="list-style-type: none">• Bot akan menuju ke diamond yang paling efisien, yaitu point/move yang paling besar.• Bot akan memprioritaskan Red Diamond dibanding Blue Diamond yang berjarak sama.	<ul style="list-style-type: none">• Bot akan menuju ke Red Diamond yang berjarak 2 kali lipat dari sebuah blue diamond dan cenderung membuat bot lebih jauh dari base.
4.	Tackling	<ul style="list-style-type: none">• Bot dapat mendapatkan banyak poin dalam waktu yang singkat.• Bot lawan akan kehilangan point sehingga memperbesar	<ul style="list-style-type: none">• Bot lawan dapat kembali men-<i>tackle</i> bot kita.• Terdapat kemungkinan bot akan “kejar-kejaran” dengan bot

		kemungkinan kita menang.	musuh sehingga menghabiskan waktu, terlebih jika bot musuh berhasil kembali ke base sebelum di- <i>tackle</i> .
5.	Point per Area	<ul style="list-style-type: none"> Bot akan pergi ke Area dengan point paling banyak sehingga jarak antar diamond cenderung lebih kecil. 	<ul style="list-style-type: none"> Jika area terlalu jauh, bot akan menghabiskan sangat banyak waktu untuk pergi ke Area tersebut kemudian kembali ke base. Komputasi yang dilakukan cukup banyak sehingga dapat mempengaruhi waktu menentukan bot bergerak.
6.	Base Distance	<ul style="list-style-type: none"> Bot akan berada pada posisi yang cenderung tidak terlalu jauh dari base sehingga waktu untuk kembali ke base tidak terlalu lama. 	<ul style="list-style-type: none"> Bot kehilangan kesempatan - kesempatan mengambil diamond yang dekat karena jaraknya sedikit lebih jauh dari base.

3.4. Strategi Greedy yang Dipilih

Berdasarkan hasil analisis kami, strategi Greedy yang kami pilih adalah gabungan antara beberapa algoritma Greedy di atas, yaitu sebagai berikut.

1. Greedy by Tackling

Greedy by Tackling kami pilih karena berdasarkan percobaan-percobaan yang kami lakukan, melakukan Tackle terhadap bot lawan sangatlah menguntungkan. Selain mendapatkan poin dengan jumlah yang besar dalam waktu yang singkat, Greedy by Tackling juga akan sangat meningkatkan peluang kita untuk menang karena lawan yang kita tackle akan kehilangan Diamond-nya, dan biasanya sangat sulit untuk kembali menjadi kompetitif pada permainan tersebut. Selain itu, dengan melakukan tackle juga kita mencegah bot kita untuk di-*tackle* oleh bot lain.

2. Greedy by Distance

Greedy by Distance kami pilih karena menurut kami jarak terdekat adalah hal yang paling penting dalam permainan ini. Karena waktu yang sangat terbatas (60 detik), bergerak dengan jarak yang pendek mengurangi risiko bot tidak sempat kembali ke base dan bot dapat mengambil lebih banyak diamond yang berdekatan. Selain itu, kami juga memperhitungkan teleporter untuk digunakan jika jaraknya memang menjadi lebih pendek sehingga meningkatkan efisiensi dan efektivitas dari algoritma ini. Pada Algoritma Greedy by distance ini, kami memperhitungkan jarak diamond dan diamond button.

BAB 4

Implementasi dan Pengujian

4.1. Implementasi Algoritma Greedy pada Program Bot yang digunakan dengan

4.1.1 Pseudocode

```
class GreedyLogic(BaseLogic):
    procedure init(self):
        Goal_tujuan = None

    procedure next_move(self, board_bot: GameObject, board: Board):
        Panggil bot, posisi bot, posisi base
        Kumpulkan data diamond, bot musuh, dan posisi teleport
        Menentukan posisi teleport dan jarak tiap teleport
        Menentukan jarak ke base
        Memeriksa jumlah diamond yang dibawa
        if(waktu < 10 detik dan masih membawa diamond)
            then goToBase(jarak ke base, jarak ke base
                menggunakan teleport, base, teleport terdekat, posisi
                sekarang)
        Mencari posisi bot lawan
        if(jarak ke bot = 1 atau (< 3 dan musuh memiliki diamond >
            2))
            then bunuh bot lawan
        if(inventory penuh atau jarak ke base dekat dan diamond >
            2)
            then goToBase(jarak ke base, jarak ke base
                menggunakan teleport, base, teleport terdekat, posisi
                sekarang)
        else
            diamondAlgorithm(diamond, teleport, redButton, base,
                posisi sekarang, properti bot, near_teleporter)
```

```
function goToBase(jarak ke base, jarak ke base menggunakan teleport,  
base, teleport terdekat, posisi sekarang)
```

```
    if ( jarak ke base <= jarak ke base menggunakan teleport)  
        then langsung ke base tanpa teleport  
    else  
        pergi ke base menggunakan teleport
```

```
return pergerakan bot
```

```
function diamondAlgorithm(diamond, teleport, redButton, base, posisi  
sekarang, properti bot, near_teleporter)
```

```
    Cari diamond terdekat
```

```
    if(jumlah diamond sudah empat dan ada red diamond)  
        Skip
```

```
    else
```

```
        Ambil diamond
```

```
        Cari posisi red button
```

```
        Menentukan jarak terdekat ke red button apakah direct atau  
        menggunakan teleport
```

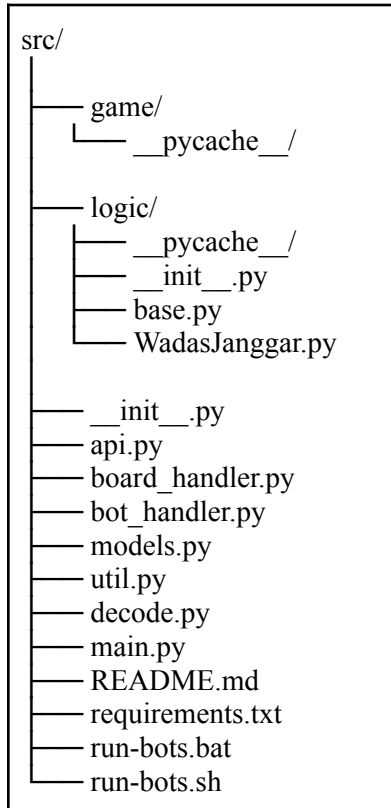
```
    if (diamond sisa sedikit dan jarak ke red diamond lebih dekat)  
        then ke red button
```

```
        Menentukan arah gerak bot
```

```
return pergerakan bot
```

4.1.2 Struktur Project

Berikut adalah struktur dari projek kita:



4.1.3 Penjelasan Fungsi-Fungsi

Berikut adalah penjelasan singkat dari fungsi yang dibuat :

Nama Fungsi	Penjelasan
distance(A: Position, B: Position)	Menghitung jarak dari B ke A menggunakan jarak Manhattan
distanceWithTeleporter(near_teleporter: int, Destination: Position, far_teleporter: Position)	Menghitung jarak dari A ke B dengan menggunakan teleporter, jarak Manhattan
goToBase(distance: int, distance_with_tele: int, base: Position, teleporter: Position, current_position: Position)	Mengeset tujuan bot agar ke base, dapat melalui teleporter dan dapat langsung ke base
diamondAlgorithm(diamonds, teleporter, redButton, base, current_position, props, near_teleporter)	Mencari diamond terdekat dengan pertimbangan red button dan teleporter

4.2. Penjelasan Struktur Data yang Digunakan dalam Program Bot Diamonds

4.2.1 Array/List

Array/List digunakan untuk menyimpan objek-objek pada board sesuai dengan jenisnya berdasarkan attribute type, yaitu “DiamondGameObject” sebagai *diamonds*, “BotGameObject” sebagai *bots*, dan “TeleportGameObject” sebagai *teleporter*.

4.3. Analisis dan Pengujian

Pengujian	Strategi	Hasil
Base	Bot akan menjadikan base sebagai tujuan jika: <ol style="list-style-type: none">1. Inventory penuh, bot membawa 5 diamond2. Membawa lebih dari 2 diamond dan jarak base kurang dari 23. Membawa diamond dan batas waktu kurang dari 10 detik	Strategi ini dalam banyak kasus memberi hasil yang optimal dimana bot tidak membawa diamond pada inventorynya pada akhir game sehingga tidak ada poin yang terbuang sia-sia
Red Button	Bot akan menjadikan red button sebagai tujuan jika jumlah diamond yang tersisa pada board kurang dari sama dengan 6 dan jarak dari red button kurang dari jarak diamond terdekat.	Strategi ini berjalan dengan baik dalam kasus dimana jumlah diamond pada board sedikit dan juga jauh sehingga lebih baik untuk bot agar mereset diamond agar kemungkinan terdapat diamond di sekitar bot menambah
Diamond	Terdapat beberapa diamond pada board namun diamond yang dijadikan tujuan oleh bot adalah diamond terdekat.	Strategi ini berjalan dengan baik dimana dengan memilih diamond terdekat maka bot pun efektif dalam mengambil makanan sekitarnya
Teleporter	Teleporter dijadikan tujuan dari bot jika: <ol style="list-style-type: none">1. Jarak dari bot ke teleporter A ditambah jarak dari teleporter B ke diamond lebih dekat dibandingkan jarak langsungnya2. Jarak dari bot ke	Strategi ini berjalan dengan baik dimana bot memilih jarak diamond/base terdekat yang sebenarnya dengan memasuki teleporter dan berpindah lokasi dengan sangat cepat.

	teleporter A ditambah jarak dari teleporter B ke base lebih dekat dibandingkan jarak langsungnya	
Bot Musuh	<p>Bot musuh akan dijadikan tujuan oleh bot jika:</p> <ol style="list-style-type: none"> 1. Bot musuh membawa lebih dari 2 diamond dan jarak dengan bot musuh kurang dari 3 2. Bot musuh terletak tepat 1 satuan dari bot kita 	Strategi ini berjalan dengan baik di mana jika bot musuh membawa lebih dari 2 diamond dan jaraknya kurang dari 3 atau bot musuh terletak tepat 1 satuan dari bot kita, maka bot kita akan menuju ke posisi bot musuh tersebut.

BAB 5

Kesimpulan dan Saran

5.1 Kesimpulan

Kesimpulan dari tugas besar ini adalah kami dapat mengaplikasikan strategi greedy dalam permainan diamonds ini. Kami berhasil membuat algoritma yang kami bilang cukup baik setelah dibandingkan dengan beberapa algoritma kawan kami. Kemudian kami juga dapat mengatakan bahwa dengan menggunakan greedy yang sifatnya mengambil optimum dari lokal belum tentu akan menghasilkan global maksimum namun menurut kami hasil dari strategi greedy ini sudah mendekati hasil optimum global.

5.2 Saran

Saran terhadap tugas besar ini adalah masih kurangnya algoritma alternatif yang dibandingkan dengan algoritma bot yang sudah kami buat, contohnya algoritma di luar Greedy sehingga bisa dibandingkan performanya terhadap algoritma Greedy yang telah dibuat ini. Selain itu, kami masih belum melakukan percobaan pada kondisi board di luar kondisi *default* sehingga belum diketahui performanya di luar kondisi *default*.

Daftar Pustaka

- [1] B. Kuliah, I. Strategi, A. Oleh, and R. Munir, "Algoritma Greedy," 2021. Available: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf)
- [2] R. Munir, "Algoritma Greedy (Bagian 2) Bahan Kuliah IF2211 Strategi Algoritma," 2021. Available: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag2.pdf)
- [3] R. Munir, "Algoritma Greedy (Bagian 3) Bahan Kuliah IF2211 Strategi Algoritma," 2022. Accessed: Mar. 08, 2024. [Online]. Available: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Greedy-\(2022\)-Bag3.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Greedy-(2022)-Bag3.pdf)
- [4] A. Medina Sari, "Algoritma Greedy: Pengertian ,Jenis dan Contoh Program," *FIKTI*, Sep. 07, 2023. <https://fikti.umsu.ac.id/algoritma-greedy-pengertian-jenis-dan-contoh-program/>

Link Repository : https://github.com/FrancescoMichael/Tubes1_Wadas-Janggar

Link Video Youtube : https://youtu.be/mqophgqEggY?si=BO_JlWXg98cwhTgv