

IMPLEMENTASI ALGORITMA X DALAM PEMECAHAN BOT PERMAINAN DIAMOND (contoh judul)

Tugas Besar

Diajukan sebagai syarat menyelesaikan mata kuliah Strategi Algoritma (IF2211)
Kelas RA di Program Studi Teknik Informatika, Fakultas Teknologi Industri,
Institut Teknologi Sumatera



Oleh: Kelompok BOTDFI

Danar Prayogo 123140015

Ilyas Ramadhan 123140016

Fanisa Aulia Safitri 123140121

Dosen Pengampu: Imam Ekowicaksono, S.Si., M.Si.

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2025**

DAFTAR ISI

Daftar Isi	ii
BAB I Deskripsi Tugas	1
1.1 Latar Belakang	1
1.2 Tujuan Tugas Besar	2
1.3 Ruang Lingkup Tugas	2
1.4 Spesifikasi Tugas	3
BAB II Landasan Teori	5
2.1 Dasar Teori	5
2.2 Integer Knapsack Problem	6
2.3 Cara Kerja Permainan Diamonds	6
2.3.1 Implementasi Algoritma Greedy	7
2.3.2 Menjalankan Bot Program	9
2.3.3 Fitur Tambahan atau Aspek Lain	10
BAB III Aplikasi Strategi <i>Greedy</i>	11
3.1 Proses <i>Mapping</i>	11

BAB I

DESKRIPSI TUGAS

Tugas besar ini merupakan bagian dari mata kuliah strategi algoritma yang mengangkat topik permainan bot berjudul Diamonds. Dalam permainan ini setiap peserta diminta untuk mengembangkan sebuah bot dan mampu untuk bersaing dengan bot peserta lain, permainannya yaitu dengan mengumpulkan diamond sebanyak mungkin pada papan permainan yang dinamis. Peserta tidak hanya mengumpulkan diamond namun, peserta harus menghadapi berbagai rintangan. Peserta diharapkan mampu merancang algoritma yang efektif dan efisien. Melalui tugas ini mahasiswa diharapkan mampu menerapkan konsep-konsep pemrograman, algoritma pencarian, serta logika berbasis strategi dalam konteks permainan yang menantang dan kompetitif.

1.1 Latar Belakang

Dalam tugas besar mata kuliah Strategi Algoritma memberikan kesempatan kepada mahasiswa untuk menerapkan pemahaman mereka dalam bentuk permainan kompetitif berjudul Diamonds. Permainan ini merupakan sebuah tantangan pemrograman dimana setiap peserta diminta untuk mengembangkan suatu bot yang otomatis dan dapat bersaing dengan bot peserta lawan. Tujuan dari permainan ini adalah mengumpulkan diamonds sebanyak-banyaknya. Namun permainan tersebut tidaklah mudah, karena permainan ini dilengkapi dengan berbagai rintangan seperti inventory yang terbatas, serta interaksi antar bot yang saling menjatuhkan.

Agar permainan menjadi unggul, setiap peserta harus merancang strategi yang efisien dan adaptif, terutama dengan menerapkan pendekatan algoritma Greedy. Algoritma ini dipilih karena mampu mengambil keputusan yang terbaik secara cepat, dan algoritma ini sangat sesuai dalam permainan dengan waktu terbatas serta kondisi lingkungan yang dapat berubah-ubah. Dengan strategi yang tepat, bot dapat secara efektif mengejar diamond, menghindari bahaya dan menyimpan hasil diamond ke dalam base.

1.2 Tujuan Tugas Besar

Tujuan dari tugas besar permainan Diamonds adalah sebagai berikut:

1. Menerapkan algoritma Greedy dalam konteks yang nyata khususnya dalam perancangan bot otomatis.
2. Dapat mengembangkan kemampuan berpikir analisis dan strategis.
3. Dapat meningkatkan keterampilan pemrograman serta penguasaan struktur data
4. Dapat melatih kerjasama antar tim
5. Mengevaluasi efektivitas strategi algoritma yang telah dirancang
6. Mengembangkan solusi yang algoritmik saat dijalankan dalam sistem bot otomatis

1.3 Ruang Lingkup Tugas

Ruang Lingkup tugas besar permainan diamonds adalah sebagai berikut:

1. Perancangan bot
Bot dirancang untuk dapat bergerak secara otomatis berdasarkan strategi algoritma yang diterapkan, bot harus mampu mengambil keputusan untuk mengambil diamond, kembali ke base atau menghindari serangan dari musuh.
2. Penerapan Algoritma Greedy
Strategi utama yang diterapkan adalah Algoritma Greedy, dengan logika pemilihan langkah terbaik berdasarkan kondisi lokal saat itu.
3. Lingkungan Permainan
Bot dioperasikan dalam papan dua dimensi yang dinamis, dan terdapat berbagai elemen seperti diamond yang memiliki warna merah dan biru, red button, base, teleport serta interaksi antar bot.
4. Batasan Teknis
Bot harus dikembangkan dalam bahasa pemrograman Python dengan memanfaatkan game engine dan bot stater pack. Bot harus dapat dijalankan dan siap untuk berkompetisi dengan bot peserta lain.
5. Evaluasi dan Dokumentasi
Evaluasi dilakukan dengan melakukan pengujian strategi bot dalam simulasi

ataupun dalam kompetisi, seluruh proses bot harus didokumentasikan dalam bentuk laporan.

1.4 Spesifikasi Tugas

Spesifikasi Tugas ini adalah

1. Tujuan Bot

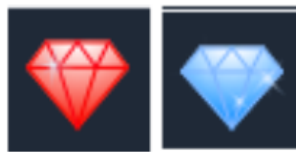
Bot harus dapat mengumpulkan diamond sebanyak mungkin, dengan keterangan diamond merah berisikan 2 point dan diamond biru berisikan 1 point, ketika inventory bot sudah penuh bot harus membawa diamond ke dalam base untuk menambahkan skor, karena diamond yang berhasil dibawa ke base akan dihitung sebagai point.

2. Ketentuan Bot

Bot memiliki inventory yang terbatas, jika inventory penuh bot harus kembali ke base untuk menyimpan diamond, bot yang ditabrak oleh bot lawan akan kehilangan seluruh diamond yang ada di inventorynya dan bot akan kembali lagi ke dalam base.

3. Elemen Permainan

(a) Diamond



Gambar 1.1: Diamonds

Terdapat 2 jenis diamond yaitu red diamond dan blue diamond dengan point yang berbeda. Red diamond memiliki 2 point dan blue diamond hanya memiliki 1 point

(b) Red Button

Ketika Bot Peserta menginjak Red Button, posisi diamond akan diubah secara acak



Gambar 1.2: Red Button



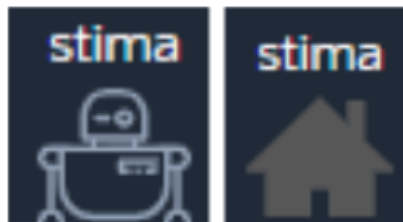
Gambar 1.3: Teleporter

(c) Teleporter

Terdapat dua teleportasi yang terhubung satu sama lain dan dapat digunakan untuk berpindah tempat.

(d) Bots dan Base

Bot akan menampung beberapa diamond diinventorynya, jika inventory bot sudah penuh maka bot akan kembali ke base untuk menyimpan diamond. Apabila diamond disimpan ke dalam base maka score bot akan bertambah, dengan posisi base tidak akan pernah berubah sampai akhir permainan.



Gambar 1.4: Bots dan Base

BAB II

LANDASAN TEORI

2.1 Dasar Teori

Algoritma Greedy adalah metode yang paling populer untuk memecahkan persoalan optimasi, dalam bahasa inggris greedy diartikan sebagai tamak atau rakus. Algoritma Greedy adalah algoritma yang memecahkan masalah langkah per langkah dengan setiap langkahnya dapat mengambil pilihan yang terbaik yang dapat diperoleh saat itu tanpa memperhatikan konsekuensi ke depan dan berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global [1]

Berikut elemen-elemen yang terdapat dalam Algoritma Greedy yang harus ditentukan dan dipertimbangkan

1. Himpunan kandidat(C): berisikan kandidat yang akan dipilih pada setiap Langkah (misal: simpul/sisi di dalam graf, job, task, koin, benda, karakter, dan sebagainya)
2. Himpunan Solusi(S): berisi kandidat yang sudah dipilih
3. Solusi: menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi
4. Fungsi seleksi(selection function): memilih kandidat berdasarkan strategi greedy tertentu. Strategi greedy ini bersifat heuristik.
5. Fungsi Kelayakan(feasible): memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi
6. Fungsi Obyektif: memaksimumkan atau meminimumkan nilai

Namun solusi yang dihasilkan oleh algoritma greedy tidak selalu menjadi solusi yang optimal, terkadang hasilnya menjadi sub-optimal atau pseudo-optimal. Karena algoritma greedy tidak akan mengeksplorasi seluruh kemungkinan yang terjadi dan keberhasilannya dalam mencapai solusi yang optimal sangat bergantung pada pemilihan fungsi seleksi yang tepat di antara berbagai pilihan yang ada.

2.2 Integer Knapsack Problem

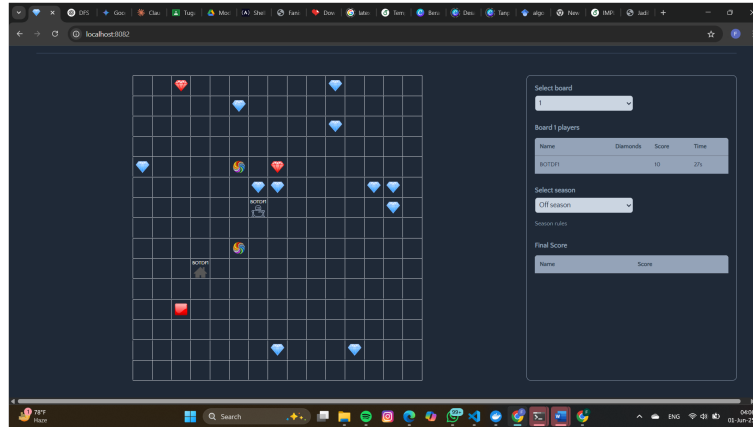
Integer Knapsack Problem adalah masalah yang ada pada riset operasi di program bilangan bulat, dimana bertujuan untuk memaksimalkan total nilai barang ke tempat yang diinginkan dengan memiliki batasan tertentu [2]. Hal ini membantu dalam mengoptimasi solusi, namun karena implementasi Algoritma Greedy juga menerapkan teori heuristik sehingga suatu langkah yang sudah dilakukan tidak dapat dilakukan backtracking. Dengan konsep ini, setiap objek memiliki properti beban yang dapat berupa jarak, bobot dan lainnya. Lalu, memiliki properti keuntungan dan juga terdapat rasio antara keuntungan dan beban, yakni densitas. Oleh karena itu, Algoritma Greedy dapat dibagi berdasarkan 3 properti yaitu:

1. Greedy by Profit (Keuntungan), yakni pada setiap langkah, pilih objek yang mempunyai keuntungan terbesar dan strategi ini mencoba memaksimalkan keuntungan dengan memiliki objek yang lebih menguntungkan.
2. Greedy by Weight (beban), yakni pada setiap langkah, dipilih objek yang mempunyai beban tercukup dari kapasitas maksimum dengan mencoba memaksimumkan keuntungan dengan memasukkan sebanyak mungkin objek ke dalam knapsack.
3. Greedy by Dencity (densitas) yaitu pada setiap langkah knapsack diisi dengan objek yang mempunyai profit atau weight terbesar dengan coba memaksimumkan keuntungan dengan mengambil oobjek yang memiliki keuntungan per unit.

2.3 Cara Kerja Permainan Diamonds

Berikut adalah cara kerja permainan Diamonds.

1. Setiap pemain bot akan ditempatkan pada board secara random. Masing-masing bot memiliki base dan score dan inventory awal adalah nol
2. Setiap pemain diberikan waktu yang sama untuk bergerak
3. Tujuan utama bot adalah mengambil diamond sebanyak-banyaknya dengan diamond merah memiliki 2 point dan diamond biru memiliki 1 poin.
4. Setiap bot memiliki inventory yang sewaktu-waktu bisa saja penuh, maka pemain



Gambar 2.5: Implementasi Algoritma Greedy Bot-Diamond

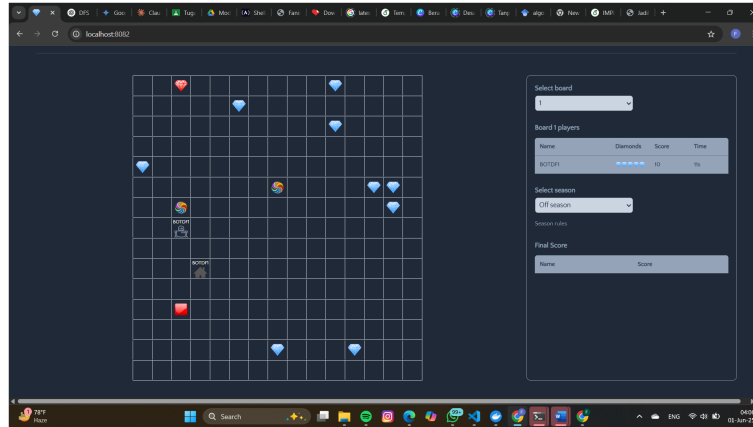
diharuskan untuk kembali ke dalam base untuk menyimpan diamond.

5. Jika diamond kembali menuju base, score bot akan bertambah senilai dengan diamond yang dibawa dan tersimpan di inventory, kemudian inventory bot akan kosong kembali dan bot akan mengambil diamond lagi.
6. Usahakan bot tidak bertemu dengan musuh atau bot peserta lain karena jika bot lawan menimpa bot peserta, maka bot Peserta akan kembali ke base dan diamond yang awalnya dimiliki oleh bot peserta akan hilang dan diambil oleh bot lawan, atau istilahnya adalah tackle.
7. Terdapat fitur tambahan seperti red button dan teleport
8. Apabila waktu seluruh bot telah berakhir, maka permainan akan berakhir. Score masing-masing pemain akan ditampilkan pada Final Score.

2.3.1 Implementasi Algoritma Greedy

Berdasarkan cara bermain dan kondisi permainan, dapat diketahui bahwa bot akan melangkah dengan kecepatan 1 langkah/detik dan juga terdapat 2 jenis diamond, yakni red diamond yang berbobot 2 poin dan blue diamond yang berbobot 1 poin. Artinya, objek yang menjadi properti beban adalah langkah/waktu (kecepatan) sedangkan objek yang menjadi properti keuntungan adalah diamond.

Dari posisi awal pada gambar 2.5 dan posisi akhir pada gambar 2.6, diperlihatkan bahwa terdapat 2 perbandingan dari diamond yang terdekat, yaitu red diamond pada posisi (2,0) dan blue diamond (0,4). Kondisi tersebut dapat diilustrasikan pada tabel



Gambar 2.6: Implementasi Algoritma Greedy Bot-Diamond

berikut:

Properti Objek				Greedy by			Solusi Optimal
i	w_i	p_i	p_i/w_i	profit	weight	density	
1	9	2	0,22	1	0	0	1
2	7	1	0,28	0	1	1	0
Total Bot				2	1	1	2
Total Keuntungan				2	1	1	2

Tabel 2.1: Implementasi Algoritma Greedy Bot-Diamond

Sebagai tindak lanjut dari hasil yang diperoleh, penulis memilih untuk mengimplementasikan Algoritma Greedy dengan pendekatan densitas (Greedy by density) untuk mendapatkan hasil yang paling mendekati kondisi optimal. Untuk memahami bagaimana bot akan bertindak, khususnya ketika berinteraksi dengan tiga komponen kunci permainan, berikut adalah penjabaran dari enam elemen inti Algoritma Greedy yang diterapkan:

1. Kumpulan Langkah Potensial (Himpunan Kandidat, C): Ini adalah semua opsi gerakan yang bisa dilakukan bot untuk meraih diamond, seperti bergerak ke lokasi-lokasi tempat diamond berada.
2. Kumpulan Langkah Terpilih (Himpunan Solusi, S): Urutan langkah dari opsi yang ada yang memberikan total diamond paling banyak, dengan memperhitungkan penggunaan Red Button untuk memaksimalkan hasil.
3. Kriteria Penyelesaian (Fungsi Solusi): Cara untuk menentukan apakah permainan atau tahap telah selesai, misalnya bot sudah mengumpulkan cukup diamond

atau waktu permainan usai. Sasaran utamanya adalah mengumpulkan diamond sebanyak mungkin sebelum kembali ke base.

4. Logika Pemilihan Langkah (Fungsi Seleksi): Metode untuk memutuskan langkah berikutnya, dengan prioritas pada langkah yang menghasilkan diamond terbanyak secara instan, setelah mempertimbangkan posisi diamond, Red Button, dan Teleporter.
5. Validitas Langkah (Fungsi Kelayakan): Sebuah langkah dinilai layak jika memungkinkan bot mencapai diamond atau Red Button tanpa risiko kehilangan diamond karena kapasitas inventaris yang penuh.
6. Tujuan Utama (Fungsi Objektif): Adalah untuk mendapatkan jumlah diamond semaksimal mungkin, dengan strategi yang mengutamakan pengambilan diamond berwarna merah dan pemanfaatan Red Button secara efektif.

2.3.2 Menjalankan Bot Program

Untuk menjalankan bot program permainan Diamonds, langkah pertama yang dilakukan adalah memastikan semua library dan dependensi yang diperlukan sudah terinstal. Program bot dituliskan dengan menggunakan bahasa Python, dan menggunakan file zip bot stater pack dan game engine yang telah disediakan. Langkah-langkah menjalankan bot:

1. Pastikan Python sudah terinstall.
2. Install dependensi dengan masuk ke direktori proyek dan jalankan instalisasinya
`pip install -r requirement.`
3. Gunakan perintah yang ada di run-bots.bat dan masukkan ke dalam terminal ganti MyBot dengan nama bot yang sudah didaftarkan di main.py
4. Jika ingin menjalankan banyak bot sekaligus maka jalankan perintah `./run-bots.bat`.
5. Verifikasi dan testing, setelah sesi permainan ini berakhir skor permainan akan ditampilkan dipapan final score.

2.3.3 Fitur Tambahan atau Aspek Lain

Untuk menunjang performa dan fleksibilitas bot dalam permainan, fitur tambahan yang digunakan adalah Handling Tackle untuk menghindari kehilangan diamond karena tabrakan dari lawan. bot dapat memprioritaskan menghindari posisi lawan atau mempercepat kembali ke base saat inventory penuh.

BAB III

APLIKASI STRATEGI *GREEDY*

3.1 Proses *Mapping*

proses mapping yang dilakukan Bot ini tidak membuat peta statis keseluruhan area permainan. Sebaliknya, "mapping" yang dilakukannya bersifat dinamis dan berfokus pada beberapa aspek:

1. Pelacakan Posisi yang Dikunjungi (*self.Visited_{position}*)

- Mekanisme: SuperBot memiliki atribut *self.visited_{position}* yang merupakan sebuah set. Setiap kali bot bergerak ke posisi baru, posisi tersebut ditambahkan ke dalam set *self.visited_{position}*. Tujuan utama dari *visited_{positions}* adalah untuk strategi eksplorasi. Dalam fungsi *putar* di tempat yang sama. Keterbatasan Pembersihan : *visited_{position}* tidak menyimpan semua posisi yang pernah dikunjungi selamanya. Ada batasan jumlah posisi yang disimpan, yaitu 50. Jika $\text{len}(\text{self.visited}_{\text{positions}}) > 50$, maka hanya 30 posisi terakhir yang akan disimpan. Ini berarti "peta" yang diingat bot.

2. Langkah Pelaksanaan:

- Pelaksanaan kegiatan
- Evaluasi hasil