

Tugas Besar 1 IF2211 Strategi Algoritma
Semester II tahun 2023/2024

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

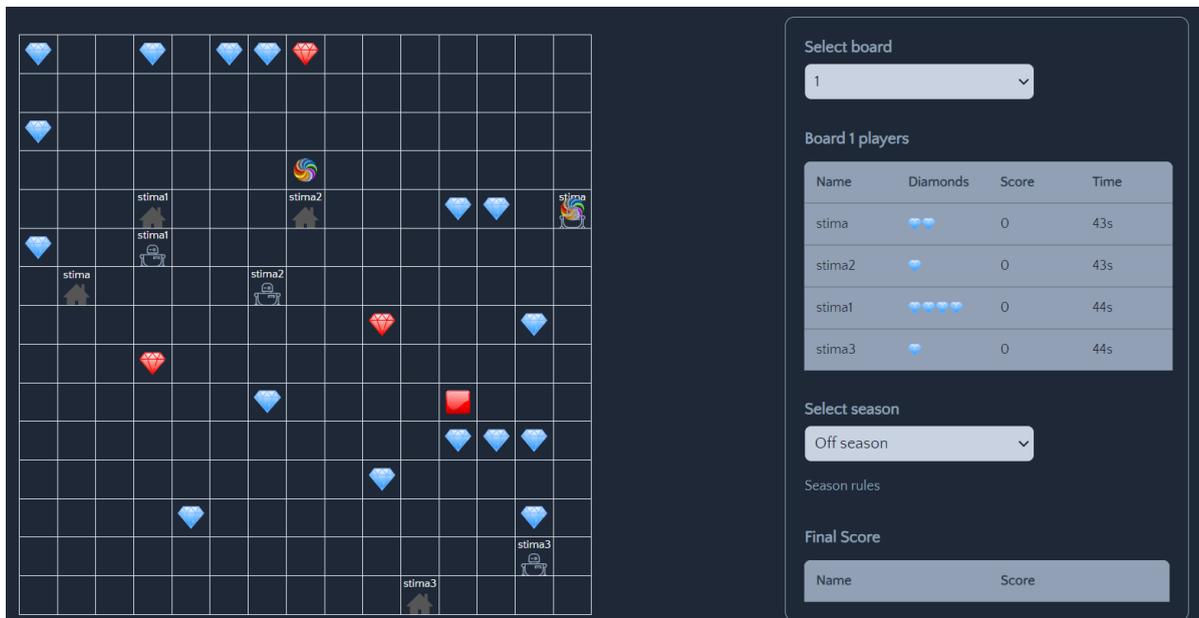
Batas pengumpulan : Hari Sabtu, 9 Maret 2024 pukul 23.59 WIB

Arsip pengumpulan :

- *Source* program *bot* yang dapat dijalankan pada permainan Diamonds disertai README
- Laporan (*soft copy*)

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:
 - a. Kode *backend* permainan, yang berisi *logic* permainan secara keseluruhan serta API yang disediakan untuk berkomunikasi dengan *frontend* dan program bot
 - b. Kode *frontend* permainan, yang berfungsi untuk memvisualisasikan permainan
2. *Bot starter pack*, yang secara umum berisi:
 - a. Program untuk memanggil API yang tersedia pada *backend*
 - b. Program *bot logic* (bagian ini yang akan kalian implementasikan dengan algoritma *greedy* untuk bot kelompok kalian)
 - c. Program utama (*main*) dan utilitas lainnya

Untuk mengimplementasikan algoritma pada bot tersebut, mahasiswa dapat menggunakan *game engine* dan membuat bot dari *bot starter pack* yang telah tersedia pada pranala berikut.

- **Game engine :**
<https://github.com/haziqam/tubes1-IF2211-game-engine/releases/tag/v1.1.0>
- **Bot starter pack :**
<https://github.com/haziqam/tubes1-IF2211-bot-starter-pack/releases/tag/v1.0.1>

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



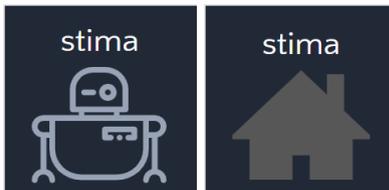
Ketika *red button* ini dilewati/dilangkahi, semua *diamond* (termasuk *red diamond*) akan di-*generate* kembali pada *board* dengan posisi acak. Posisi *red button* ini juga akan berubah secara acak jika *red button* ini dilangkahi.

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

Name	Diamonds	Score	Time
stima	💎💎	0	43s
stima2	💎	0	43s
stima1	💎💎💎💎	0	44s
stima3	💎	0	44s

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.

Panduan Penggunaan

Adapun panduan mengenai cara instalasi, menjalankan permainan, membuat bot, melihat *visualizer/frontend*, dan mengatur konfigurasi permainan dapat dilihat melalui tautan berikut.

 [Get Started with Diamonds](#)

Mekanisme Teknis Permainan Diamonds

DISCLAIMER:

Kalian **tidak wajib** memahami bagian ini, karena semua aspek teknis sebenarnya sudah disediakan pada *game engine* dan *bot starter pack* yang diberikan (kalian hanya perlu merancang *logic* dari botnya saja), namun ini hanya sekedar *fun fact* bagi yang penasaran mengenai cara kerja *game* ini (siapa tau bisa menjadi inspirasi untuk membuat *game* berbasis *web*).

Permainan ini merupakan permainan berbasis *web*, sehingga setiap aksi yang dilakukan – mulai dari mendaftarkan bot hingga menjalankan aksi bot – akan memerlukan HTTP *request*

terhadap API *endpoint* tertentu yang disediakan oleh *backend*. Berikut adalah urutan *requests* yang terjadi dari awal mula permainan.

1. Program bot akan mengecek apakah bot sudah terdaftar atau belum, dengan mengirimkan *POST request* terhadap *endpoint* `/api/bots/recover` dengan *body* berisi *email* dan *password* bot. Jika bot sudah terdaftar, maka *backend* akan memberikan *response code* 200 dengan *body* berisi id dari bot tersebut. Jika tidak, *backend* akan memberikan *response code* 404.
2. Jika bot belum terdaftar, maka program bot akan mengirimkan *POST request* terhadap *endpoint* `/api/bots` dengan *body* berisi *email*, *name*, *password*, dan *team*. Jika berhasil, maka *backend* akan memberikan *response code* 200 dengan *body* berisi id dari bot tersebut.
3. Ketika id bot sudah diketahui, bot dapat bergabung ke *board* dengan mengirimkan *POST request* terhadap *endpoint* `/api/bots/{id}/join` dengan *body* berisi board id yang diinginkan (*preferredBoardId*). Apabila bot berhasil bergabung, maka *backend* akan memberikan *response code* 200 dengan *body* berisi informasi dari *board*.
4. Program bot akan mengkalkulasikan *move* selanjutnya secara berkala berdasarkan kondisi *board* yang diketahui, dan mengirimkan *POST request* terhadap *endpoint* `/api/bots/{id}/move` dengan *body* berisi *direction* yang akan ditempuh selanjutnya ("NORTH", "SOUTH", "EAST", atau "WEST"). Apabila berhasil, maka *backend* akan memberikan *response code* 200 dengan *body* berisi kondisi *board* setelah *move* tersebut. Langkah ini dilakukan terus-menerus hingga waktu bot habis. Jika waktu bot habis, bot secara otomatis akan dikeluarkan dari *board*.
5. Program *frontend* secara periodik juga akan mengirimkan *GET request* terhadap *endpoint* `/api/boards/{id}` untuk mendapatkan kondisi *board* terbaru, sehingga tampilan *board* pada *frontend* akan selalu *ter-update*.

Spesifikasi Tugas Besar 1

- Buatlah program sederhana dalam bahasa **Python** yang mengimplementasikan **algoritma Greedy** pada *bot* permainan Diamonds dengan tujuan memenangkan permainan.
- Tugas dikerjakan berkelompok dengan anggota minimal 2 orang dan maksimal 3 orang, boleh lintas kelas dan lintas kampus.
- Strategi *greedy* yang diimplementasikan setiap kelompok harus dikaitkan dengan fungsi objektif dari permainan ini, yaitu memenangkan permainan dengan memperoleh *diamond* sebanyak banyak nya dan jangan sampai *diamond* tersebut diambil oleh bot lain. Buatlah strategi *greedy* terbaik, karena setiap bot dari masing-masing kelompok akan diadu dalam kompetisi Tubes 1.
- Strategi *greedy* yang kelompok anda buat harus dijelaskan dan ditulis secara eksplisit pada laporan, karena akan diperiksa saat demo apakah strategi yang dituliskan sesuai dengan yang diimplementasikan. Tiap kelompok dapat menggunakan kreativitas yang bermacam macam dalam menyusun strategi *greedy* untuk memenangkan

permainan. Implementasi pemain harus dapat dijalankan pada *game engine* yang telah disebutkan diatas serta dapat dikompetisikan dengan bot dari kelompok lain.

- Program harus mengandung komentar yang jelas, dan untuk setiap strategi *greedy* yang disebutkan, harus dilengkapi dengan kode sumber yang dibuat.
- Mahasiswa dilarang menggunakan kode program yang diunduh dari Internet. Mahasiswa harus membuat program sendiri, diperbolehkan untuk belajar dari program yang sudah ada.
- Mahasiswa dianggap sudah melihat dokumentasi dari *game engine*, sehingga tidak terjadi kesalahpahaman spesifikasi antara mahasiswa dan asisten.
- BONUS (maks 10): Membuat video tentang aplikasi *greedy* pada bot serta simulasinya pada game kemudian mengunggahnya di Youtube. Video dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Untuk contoh video tubes stima tahun-tahun sebelumnya dapat dilihat di Youtube dengan kata kunci “Tubes Stima”, “strategi algoritma”, “Tugas besar stima”, dll.
- Jika terdapat kesulitan selama mengerjakan tugas besar sehingga memerlukan bimbingan, maka dapat melakukan asistensi tugas besar kepada asisten (opsional). Dengan catatan asistensi hanya bersifat membimbing, bukan memberikan “jawaban”.
- Terdapat juga demo dari program yang telah dibuat. Pengumuman tentang demo menunggu pemberitahuan lebih lanjut dari asisten.
- Bot yang telah dibuat akan dikompetisikan dengan kelompok lain dan disaksikan oleh seluruh peserta kuliah. Terdapat hadiah menarik bagi kelompok yang memenangkan kompetisi.
- Setiap kelompok harap mengisi nama kelompok dan anggotanya pada link berikut, paling lambat **Sabtu, 24 Februari pukul 22.11 WIB**.

 [Pendataan Kelompok Tubes 1 Stima](#)

- Diwajibkan untuk memilih asisten meskipun tidak melakukan asistensi, karena asisten yang dipilih akan menjadi asisten saat asistensi (opsional) dan demo tugas besar. Pemilihan asisten dapat dilakukan pada link berikut, paling lambat **Sabtu, 24 Februari pukul 22.11 WIB**.

 [Pendataan Kelompok Tubes 1 Stima](#)

- Program disimpan dalam repository yang bernama **Tubes1_NamaKelompok** dengan nama kelompok sesuai dengan yang di sheets diatas. Berikut merupakan struktur dari isi repository tersebut:
 - a. Folder src berisi source code.
 - b. Folder doc berisi laporan tugas besar dengan format **NamaKelompok.pdf**
 - c. README untuk tata cara penggunaan yang minimal berisi:
 - i. Penjelasan singkat algoritma *greedy* yang diimplementasikan
 - ii. Requirement program dan instalasi tertentu bila ada
 - iii. Command atau langkah-langkah dalam meng-compile atau build program
 - iv. Author (identitas pembuat)

- Laporan dikumpulkan hari **Sabtu, 9 Maret 2024** pada alamat Google Form berikut paling lambat pukul **23.59** :
<https://bit.ly/tubes1stima24>
- Adapun pertanyaan terkait tugas besar ini bisa disampaikan melalui QnA berikut:
<https://bit.ly/qnastima24>

Isi laporan

- **Cover:** Cover laporan ada foto anggota kelompok (foto bertiga). Foto ini menggantikan logo “gajah” ganesha.
- **Bab 1:** Deskripsi tugas (dapat menyalin spesifikasi tugas ini).
- **Bab 2:** Landasan Teori.
 - Dasar teori (algoritma *greedy*) secara umum
 - Bagaimana cara kerja program secara umum (bagaimana bot melakukan aksinya, bagaimana mengimplementasikan algoritma *greedy* ke dalam bot, bagaimana menjalankan bot, dll).
- **Bab 3:** Aplikasi strategi *greedy*.
 - Proses mapping persoalan Diamonds menjadi elemen-elemen algoritma Greedy (himpunan kandidat, himpunan solusi, fungsi solusi, fungsi seleksi, fungsi kelayakan, fungsi objektif)
 - Eksplorasi alternatif solusi *greedy* yang mungkin dipilih dalam persoalan Diamonds
 - Analisis efisiensi dan efektivitas dari kumpulan alternatif solusi *greedy* yang dirumuskan
 - Strategi *greedy* yang dipilih (yang akan diimplementasikan dalam program) beserta alasan dan pertimbangan pemilihan strategi tersebut.
- **Bab 4:** Implementasi dan pengujian.
 - Implementasi algoritma *greedy* pada program bot yang digunakan (*pseudocode* yang cukup detail dengan komentar untuk pembaca kode agar mudah dipahami).
 - Penjelasan struktur data yang digunakan dalam program bot Diamonds.
 - Analisis dari desain solusi algoritma *greedy* yang diimplementasikan pada setiap **pengujian** yang dilakukan. Misalnya adalah apakah strategi *greedy* berhasil mendapatkan nilai optimal, lalu jika tidak, dalam kondisi seperti apa strategi *greedy* tidak berhasil mendapatkan nilai optimal, dsb. Pengujian yang dicantumkan sebaiknya mencakup semua kejadian yang unik.
- **Bab 5:** Kesimpulan dan saran.
- **Lampiran:** Tautan *repository* GitHub dan video (jika membuat)
- **Daftar Pustaka**

Keterangan laporan:

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar.
2. Laporan mengikuti format pada *section* "Isi laporan" dengan baik dan benar.
3. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

Penilaian:

1. Bagian 1: Desain Solusi Algoritma Greedy ditulis dalam Laporan (45%)

- a. Pemahaman tugas besar (5%)
- b. Mapping persoalan Diamonds ke dalam elemen-elemen algoritma *greedy* (10%)
- c. Eksplorasi alternatif-alternatif solusi persoalan Diamonds dengan algoritma Greedy (10%). Semakin banyak alternatif yang dieksplorasi maka akan semakin bagus nilainya.
- d. Analisis efisiensi dan efektivitas secara teoritis dari alternatif solusi persoalan (5%)
- e. Analisis dari Desain solusi Algoritma Greedy yang diusulkan untuk diimplementasikan dalam program/coding (15%)

2. Bagian 2: Implementasi Program dan Demo (55%)

- a. Kesesuaian strategi *Greedy* yang dituliskan dengan implementasi dan saat demo (20%)
- b. Kualitas algoritma *Greedy* yang diimplementasikan (10%)
- c. Modularitas/keterbacaan penulisan program (5%)
- d. Demo pemahaman program (20%)

3. Bagian 3: Kompetisi dan Komponen Lainnya

- a. Beberapa kelompok pemenang (akan ditentukan jumlahnya nanti) pada saat kompetisi akan mendapatkan bonus nilai (**bonus** maksimal 10 poin)
- b. Bonus dalam membuat video kelompok (**bonus** maksimal 10 poin)

Perhatikan:

- **Dilarang keras** *copy paste* program dari internet (AI, repository lain, ataupun teman). Program harus dibuat sendiri, kecurangan akan mengakibatkan nilai tugas menjadi nol.
- Pastikan program **dapat setidaknya dijalankan** pada *windows* dan *linux*.
- Apabila program **tidak bisa dijalankan** maka tidak akan dinilai oleh asisten.

- - - Selamat Mengerjakan - - -

“Greedy nya cukup di tubes ini aja oke!”

- Leon -

“Baru tubes satu, harusnya kelarlah yah”

- Marthen -

“Jangan ada yang ngilang ya, kasian temen kelompoknya bebannya jadi lebih berat :(”

- Haziq -

“Satu kata lucu, greedy”

- Rifqi -

“ε=ε=ε= ∩(∇)∩”

∩∩∩∩∩ (∩∩∩∩∩)

- Alex -

“Wih dah tubes pertama aja nih, semangat 🙌”

- Sulthan -

“Semangat gais tubes perdana stimanya, ingat algoritmanya **greedy** ya”

- Nigel -

“Keren dah bikin bot”

- Bewe -