

Tugas Besar I IF2211 Strategi Algoritma
Semester II Tahun 2022/2023

Pemanfaatan Algoritma *Greedy* dalam Aplikasi Permainan “Galaxio”

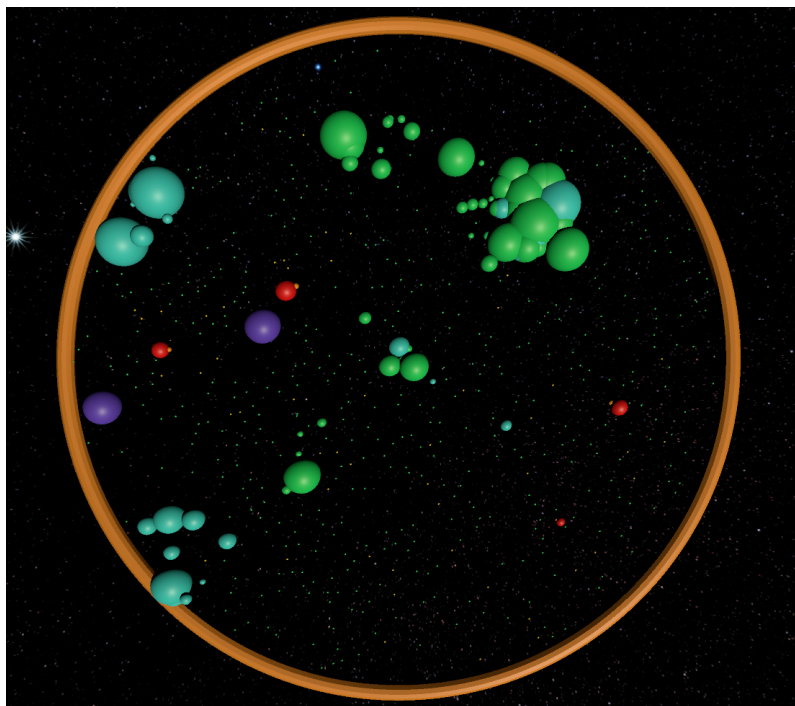
Batas pengumpulan : Hari Jumat, 17 Februari 2023 sebelum pukul 23.59 WIB

Arsip pengumpulan :

- *Source* program yang bisa dijalankan pada *game engine Galaxio* disertai README
- Laporan (*soft copy*)

Deskripsi tugas:

Galaxio adalah sebuah *game battle royale* yang mempertandingkan bot kapal anda dengan beberapa bot kapal yang lain. Setiap pemain akan memiliki sebuah bot kapal dan tujuan dari permainan adalah agar bot kapal anda yang tetap hidup hingga akhir permainan. Penjelasan lebih lanjut mengenai aturan permainan akan dijelaskan di bawah. Agar dapat memenangkan pertandingan, setiap bot harus mengimplementasikan strategi tertentu untuk dapat memenangkan permainan.



Gambar 1. Ilustrasi permainan *Galaxio*

Pada tugas besar pertama Strategi Algoritma ini, gunakanlah sebuah *game engine* yang mengimplementasikan permainan *Galaxio*. *Game engine* dapat diperoleh pada laman berikut:

<https://github.com/EntelectChallenge/2021-Galaxio>

Tugas mahasiswa adalah mengimplementasikan bot kapal dalam permainan *Galaxio* dengan menggunakan **strategi greedy** untuk memenangkan permainan. Untuk mengimplementasikan bot tersebut, mahasiswa disarankan melanjutkan program yang terdapat pada *starter-bots* di dalam *starter-pack* pada laman berikut ini:

<https://github.com/EntelectChallenge/2021-Galaxio/releases/tag/2021.3.2>

Spesifikasi permainan yang digunakan pada tugas besar ini disesuaikan dengan spesifikasi yang disediakan oleh *game engine Galaxio* pada tautan di atas. Beberapa aturan umum adalah sebagai berikut.

1. Peta permainan berbentuk kartesius yang memiliki arah positif dan negatif. Peta hanya menangani angka bulat. Kapal hanya bisa berada di *integer* x,y yang ada di peta. Pusat peta adalah 0,0 dan ujung dari peta merupakan radius. Jumlah ronde maximum pada game sama dengan ukuran radius. Pada peta, akan terdapat 5 objek, yaitu *Players*, *Food*, *Wormholes*, *Gas Clouds*, *Asteroid Fields*. Ukuran peta akan mengecil seiring batasan peta mengecil.
2. Kecepatan kapal dilambangkan dengan x . Kecepatan kapal akan dimulai dengan kecepatan 20 dan berkurang setiap ukuran kapal bertambah. Ukuran (radius) kapal akan dimulai dengan ukuran 10. *Heading* dari kapal dapat bergerak antar 0 hingga 359 derajat. Efek *afterburner* akan meningkatkan kecepatan kapal dengan faktor 2, tetapi mengecilkan ukuran kapal sebanyak 1 setiap tick. Kemudian kapal akan menerima 1 salvo charge setiap 10 tick. Setiap kapal hanya dapat menampung 5 salvo charge. Penembakan salvo torpedo (ukuran 10) mengurangi ukuran kapal sebanyak 5.
3. Setiap objek pada lintasan punya koordinat x,y dan radius yang mendefinisikan ukuran dan bentuknya. *Food* akan disebar pada peta dengan ukuran 3 dan dapat dikonsumsi oleh kapal *player*. Apabila *player* mengkonsumsi *Food*, maka *Player* akan bertambah ukuran yang sama dengan *Food*. *Food* memiliki peluang untuk berubah menjadi *Super Food*. Apabila *Super Food* dikonsumsi maka setiap makan *Food*, efeknya akan 2 kali dari *Food* yang dikonsumsi. Efek dari *Super Food* bertahan selama 5 tick.
4. *Wormhole* ada secara berpasangan dan memperbolehkan kapal dari *player* untuk memasukinya dan keluar di pasangan satu lagi. *Wormhole* akan bertambah besar setiap tick game hingga ukuran maximum. Ketika *Wormhole* dilewati, maka *wormhole* akan mengecil sebanyak setengah dari ukuran kapal yang melewatinya dengan syarat *wormhole* lebih besar dari kapal *player*.
5. *Gas Clouds* akan tersebar pada peta. Kapal dapat melewati *gas cloud*. Setiap kapal bertabrakan dengan *gas cloud*, ukuran dari kapal akan mengecil 1 setiap tick game. Saat kapal tidak lagi bertabrakan dengan *gas cloud*, maka efek pengurangan akan hilang.
6. *Torpedo Salvo* akan muncul pada peta yang berasal dari kapal lain. *Torpedo Salvo* berjalan dalam lintasan lurus dan dapat menghancurkan semua objek yang berada pada

lintasannya. Torpedo Salvo dapat mengurangi ukuran kapal yang ditabraknya. *Torpedo Salvo* akan mengecil apabila bertabrakan dengan objek lain sebanyak ukuran yang dimiliki dari objek yang ditabraknya.

7. *Supernova* merupakan senjata yang hanya muncul satu kali pada permainan di antara quarter pertama dan quarter terakhir. Senjata ini tidak akan bertabrakan dengan objek lain pada lintasannya. *Player* yang menembakannya dapat meledakannya dan memberi *damage* ke *player* yang berada dalam zona. Area ledakan akan berubah menjadi *gas cloud*.
8. *Player* dapat meluncurkan *teleporter* pada suatu arah di peta. *Teleporter* tersebut bergerak dalam direksi dengan kecepatan 20 dan tidak bertabrakan dengan objek apapun. *Player* tersebut dapat berpindah ke tempat *teleporter* tersebut. Harga setiap peluncuran *teleporter* adalah 20. Setiap 100 tick *player* akan mendapatkan 1 *teleporter* dengan jumlah maximum adalah 10.
9. Ketika kapal *player* bertabrakan dengan kapal lain, maka kapal yang lebih besar akan dikonsumsi oleh kapal yang lebih kecil sebanyak 50% dari ukuran kapal yang lebih besar hingga ukuran maximum dari ukuran kapal yang lebih kecil. Hasil dari tabrakan akan mengarahkan kedua dari kapal tersebut lawan arah.
10. Terdapat beberapa *command* yang dapat dilakukan oleh *player*. Setiap tick, *player* hanya dapat memberikan satu *command*. Berikut jenis-jenis dari *command* yang ada dalam permainan:
 - a. FORWARD
 - b. STOP
 - c. START_AFTERBURNER
 - d. STOP_AFTERBURNER
 - e. FIRE_TORPEDOES
 - f. FIRE_SUPERNOVA
 - g. DETONATE_SUPERNOVA
 - h. FIRE_TELEPORTER
 - i. TELEPORTUSE_SHIELD
11. Setiap *player* akan memiliki score yang hanya dapat dilihat jika permainan berakhir. Score ini digunakan saat kasus *tie breaking* (semua kapal mati). Jika mengonsumsi kapal *player* lain, maka score bertambah 10, jika mengonsumsi *food* atau melewati wormhole, maka score bertambah 1. Pemenang permainan adalah kapal yang bertahan paling terakhir dan apabila *tie breaker* maka pemenang adalah kapal dengan score tertinggi.

Adapun peraturan yang lebih lengkap dari permainan *Galaxio*, dapat dilihat pada laman :

<https://github.com/EntelectChallenge/2021-Galaxio/blob/develop/game-engine/game-rules.md>

Spesifikasi tugas:

Pada tugas besar kali ini, anda diminta untuk membuat sebuah *bot* untuk bermain permainan *Galaxio* yang telah dijelaskan sebelumnya. Untuk memulai, anda dapat mengikuti panduan singkat sebagai berikut.

1. *Download latest release starter pack.zip* dari tautan berikut
<https://github.com/EntelectChallenge/2021-Galaxio/releases/tag/2021.3.2>
2. Untuk menjalankan permainan, kalian butuh beberapa *requirement* dasar sebagai berikut.
 - a. Java (minimal Java 11):
<https://www.oracle.com/java/technologies/downloads/#java>
 - b. IntelliJ IDEA: <https://www.jetbrains.com/idea/>
 - c. NodeJS: <https://nodejs.org/en/download/>
 - d. .Net Core 3.1: <https://dotnet.microsoft.com/en-us/download/dotnet/3.1>
3. Panduan mengenai cara menjalankan permainan, membuat bot, *build* src code, dan melihat visualizer bisa dicek melalui tautan berikut:
https://docs.google.com/document/d/1Ym2KomFPLIG_KAbm3A0bnhw4_XQAsOKzpTa70lgnLNU/edit#
4. Silahkan bersenang-senang dengan memodifikasi bot yang disediakan di starter-bot. Ingat bahwa bot kalian harus menggunakan bahasa **Java**. Dilarang menggunakan kode program yang sudah ada, mahasiswa wajib membuat program/strategi sendiri. Tetapi, belajar dari program yang sudah ada tidak dilarang.

Strategi *greedy* yang diimplementasikan tiap kelompok harus dikaitkan dengan fungsi objektif dari permainan itu sendiri, yaitu memenangkan permainan dengan cara mempertahankan kapal pemain paling terakhir untuk hidup. Salah satu contoh pendekatan *greedy* yang bisa digunakan (pendekatan tak terbatas pada contoh ini saja) adalah menghindari objek yang dapat mengurangi ukuran kapal. Buatlah strategi *greedy* terbaik, karena setiap bot dari masing-masing kelompok akan diadu dalam suatu kompetisi Tubes 1 (TBD).

Strategi *greedy* harus dijelaskan dan ditulis secara eksplisit pada laporan, karena akan diperiksa pada saat demo apakah strategi yang dituliskan sesuai dengan yang diimplementasikan. Tiap kelompok dapat menggunakan kreativitas mereka dalam menyusun strategi *greedy* untuk memenangkan permainan. Implementasi pemain harus dapat dijalankan pada game engine yang telah disebutkan pada spesifikasi tugas besar, serta dapat dikompetisikan dengan pemain dari kelompok lain.

Lain-lain:

1. Tugas dikerjakan berkelompok dengan jumlah anggota minimal 2 orang dan maksimal 3 orang. Anggota kelompok diperbolehkan lintas kelas.
2. Program dibuat dalam **Bahasa Java**, sebagai salah satu bahasa yang dapat difasilitasi oleh *game engine* pada Tugas Besar ini. Dengan Java berarti anda sekaligus dapat mempraktekkan konsep pemrograman berorientasi objek (OOP) pada tugas ini.

3. Program harus mengandung komentar yang jelas, dan untuk setiap strategi Greedy yang disebutkan, harus dilengkapi dengan kode sumber yang dibuat.
4. Beri nama pemain anda tersebut dengan nama kelompok anda. Pengaturan konfigurasi nama pemain bisa diatur pada file `main.java` line 61 yang terdapat dalam direktori "JavaBot".
5. Terdapat contoh implementasi pemain pada *repository* dari *game engine* tersebut, namun mahasiswa dilarang menggunakan kode program tersebut untuk pemainnya atau kode program lain yang diunduh dari Internet. Mahasiswa harus membuat program sendiri, tetapi belajar dari program yang sudah ada tidak dilarang.
6. Mahasiswa dianggap sudah membaca dokumentasi dari *game engine* tersebut, sehingga tidak terjadi kesalahpahaman spesifikasi antara mahasiswa dan asisten.
7. Deadline Pengumpulan tugas besar adalah hari Jumat, 17 Februari 2023 pukul **23.59 WIB**. Keterlambatan akan berakibat pada **penalti**.
8. **Bonus** (nilai maksimal 10): Setiap kelompok membuat video aplikasi yang mereka buat kemudian mengunggahnya ke YouTube. Video yang dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Pada waktu demo aplikasi di depan asisten, mahasiswa mengakses video YouTube tersebut dan memutarinya di depan asisten. Beberapa contoh video tubes tahun-tahun sebelumnya dapat dilihat di YouTube dengan menggunakan kata kunci "Tubes Stima", "Tugas besar stima", "strategi algoritma", dll.
9. Semua pertanyaan menyangkut tugas ini harus dikomunikasikan melalui QnA yang disediakan agar dapat dicermati oleh semua peserta kuliah IF2211. Laman QnA dapat diakses pada <https://bit.ly/TugasStimaQnA>
10. Demo program menunggu pemberitahuan lebih lanjut dari asisten.
11. Program akan dikompertisikan (diadu) dengan kelompok lain dan disaksikan oleh seluruh peserta kuliah. **Pemenang kompetisi akan diberi hadiah dari Dosen Strategi Algoritma. Waktu kompetisi akan ditentukan kemudian oleh asisten.**
12. Setiap anggota harus memahami proses pembuatan program, karena akan ada pertanyaan-pertanyaan yang harus dijawab per individu.
13. Setiap kelompok harap mengisi nama kelompok dan anggotanya pada link <https://bit.ly/KelompokTubes1Stima>
14. Program disimpan dalam repository bernama Tubes1_NamaKelompok dengan NamaKelompok merupakan nama kelompok yang terdaftar pada <https://bit.ly/KelompokTubes1Stima>. Isi dari repository tersebut **adalah sama** dengan isi folder JavaBot yang sudah anda kerjakan ditambahkan laporan. Berikut adalah struktur dari repository tersebut:
 - a. Folder **src** berisi **source code**.
 - b. Folder **target** berisi hasil build dari source code, termasuk **executable** (File .jar) dengan format **nama_kelompok.jar**
 - c. Folder **doc** berisi **laporan tugas besar** dengan format **nama_kelompok.pdf**
 - d. 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
- e. Dockerfile
- f. pom.xml

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 *game engine*, dll).
- **Bab 3:** Aplikasi strategi *greedy*.
 - Proses mapping persoalan *Galaxio* 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 *Galaxio*
 - 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 dalam *game engine* yang digunakan (*pseudocode* yang cukup detail dengan komentar untuk pembaca kode agar mudah dipahami).
 - Penjelasan struktur data yang digunakan dalam program bot *Galaxio* dan struktur data tambahan jika ada.
 - 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.
- **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 (50%)**
 - a. Mapping persoalan *Galaxio* ke dalam elemen-elemen algoritma Greedy (5%)
 - b. Eksplorasi alternatif-alternatif solusi persoalan *Galaxio* dengan algoritma Greedy (10%). Semakin banyak alternatif yang dieksplorasi maka akan semakin bagus nilainya.
 - c. Analisis efisiensi secara teoritis dan alternatif solusi persoalan (5%)
 - d. Analisis efektivitas secara teoritis dari alternatif solusi persoalan (5%)
 - e. Analisis dari Desain solusi Algoritma Greedy yang diusulkan untuk diimplementasikan dalam program/coding (10%)
 - f. Kualitas Strategi Algoritma yang diusulkan (15%)
2. **Bagian 2: Implementasi Program / Coding (50%)**
 - a. Kesesuaian strategi Greedy yang dituliskan dengan implementasi dan saat demo (20%)
 - b. Modularitas/keterbacaan penulisan program (5%)
 - c. Demo pemahaman program (25%)
3. **Bagian 3: Kompetisi dan Komponen Lainnya**
 - a. Beberapa kelompok pemenang (akan ditentukan jumlahnya nanti) pada saat kompetisi akan mendapatkan bonus nilai (**bonus** 10% dari total nilai)
 - b. Bonus dalam membuat video kelompok (**bonus** 10% dari total nilai)

--- Selamat Mengerjakan! ---

“Hidup memang ga boleh *greedy*, tapi khusus tubes ini harus”

– Saul –

“”

– Steven –

“”

– Angel –

“”

– Vio –

“”

– Dimas –

“Enjoy the game!”

– Ziyad –

“”

– Bryan –

“”

– Willy –