

Tugas Besar 1 IF2211 Strategi Algoritma
Semester II tahun 2024/2025

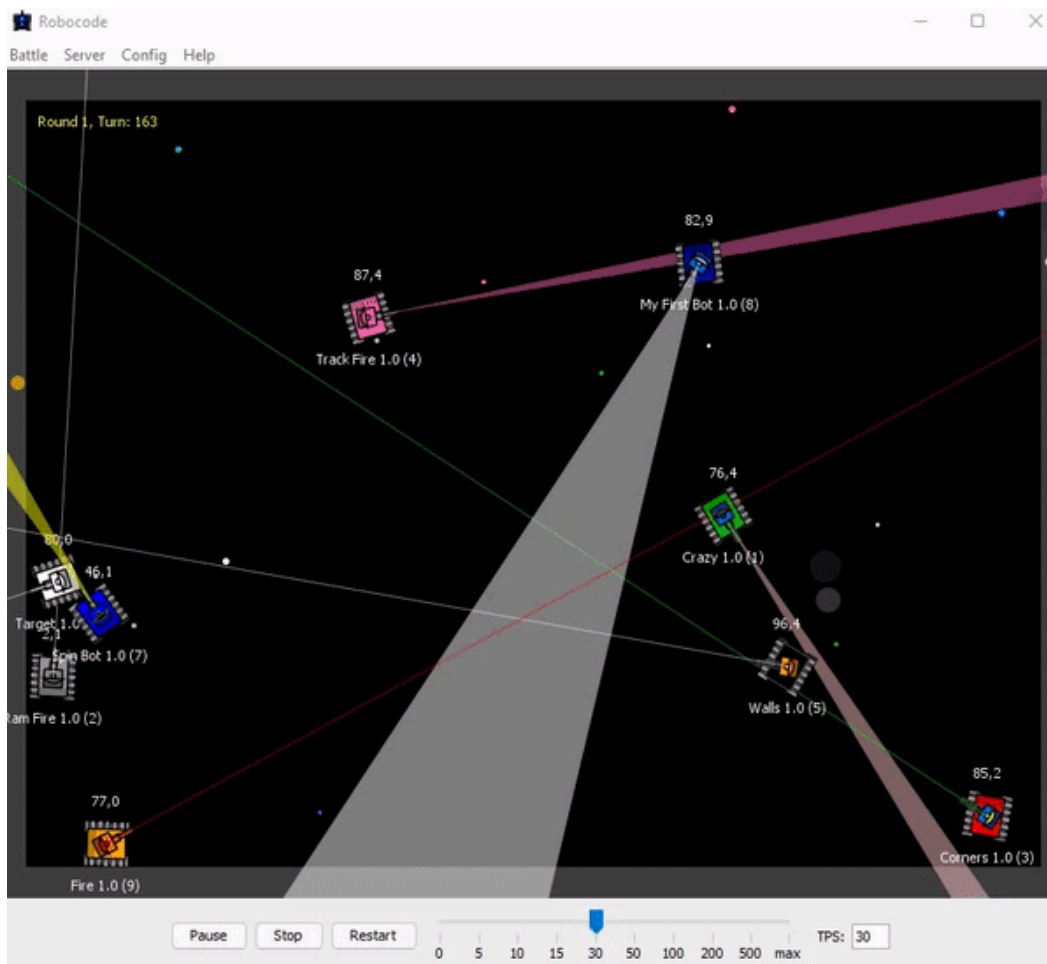
Pemanfaatan **Algoritma Greedy** dalam pembuatan *bot* permainan **Robocode Tank Royale**

Batas pengumpulan : Hari Senin, 24 Maret 2025 pukul 22.11 WIB

Arsip pengumpulan :

- *Source* program yang dapat dijalankan disertai README
- Laporan (*soft copy*)

Deskripsi Tugas:



Gambar 1 Robocode Tank Royale

Robocode adalah permainan pemrograman yang bertujuan untuk membuat kode bot dalam bentuk tank virtual untuk berkompetisi melawan bot lain di arena. Pertempuran Robocode berlangsung hingga bot-bot bertarung hanya tersisa satu seperti permainan Battle Royale, karena itulah permainan ini dinamakan Tank Royale. Nama Robocode adalah singkatan dari "Robot code," yang berasal dari [versi asli/pertama permainan ini](#). Robocode Tank Royale adalah evolusi/versi berikutnya dari permainan ini, di mana bot dapat berpartisipasi melalui Internet/jaringan. Dalam permainan ini, pemain berperan sebagai programmer bot dan tidak memiliki kendali langsung atas permainan. Pemain hanya bertugas untuk membuat program yang menentukan logika atau "otak" bot. Program yang dibuat akan berisi instruksi tentang cara bot bergerak, mendeteksi bot lawan, menembakkan senjatanya, serta bagaimana bot bereaksi terhadap berbagai kejadian selama pertempuran.

Pada Tugas Besar 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.

Komponen-komponen dari permainan ini antara lain:

1. Rounds dan Turns

Pertempuran dapat terdiri dari beberapa rounds. Secara default, satu pertempuran berisi 10 rounds, di mana setiap rounds akan memiliki pemenang dan yang kalah.

Setiap round dibagi menjadi beberapa turns, yang merupakan unit waktu terkecil. Satu turn adalah satu ketukan waktu dan satu putaran permainan. Jumlah turn dalam satu round tergantung pada berapa lama waktu yang dibutuhkan hingga hanya tersisa bot terakhir yang bertahan.

Pada setiap turn, sebuah bot dapat:

- Menggerakkan bot, memindai musuh, dan menembakkan senjata.
- Bereaksi terhadap peristiwa seperti saat bot terkena peluru atau bertabrakan dengan bot lain atau dinding.
- Perintah untuk bergerak, berputar, memindai, menembak, dan sebagainya dikirim ke server untuk setiap turn.

Perlu diperhatikan bahwa [API \(Application Programming Interface\)](#) bot resmi secara otomatis mengirimkan niat bot ke server di balik layar, sehingga Anda tidak perlu mengkhawatirkannya, kecuali jika Anda membuat API Bot sendiri.

Pada setiap turn, bot akan secara otomatis menerima informasi terbaru tentang posisinya dan orientasinya di medan perang. Bot juga akan mendapatkan informasi tentang bot musuh ketika mereka terdeteksi oleh pemindai.

Perlu diketahui bahwa game engine yang akan digunakan pada tugas besar ini tidak mengikuti aturan default mengenai komponen Round & Turns.

2. Batas Waktu Giliran

Penting untuk dicatat bahwa setiap bot memiliki batas waktu untuk setiap turn yang disebut turn timeout, biasanya antara 30-50 ms (dapat diatur sebagai aturan pertempuran). Ini berarti bahwa bot tidak bisa mengambil waktu sebanyak yang mereka inginkan untuk bergerak dan menyelesaikan turn saat ini.

Setiap kali turn baru dimulai, penghitung waktu ulang diatur ulang dan mulai berjalan. Jika batas waktu tercapai dan bot tidak mengirimkan pergerakannya untuk turn tersebut, maka tidak ada perintah yang dikirim ke server. Akibatnya, bot akan melewatkan turn tersebut. Jika bot melewatkan turn, ia tidak akan bisa menyesuaikan gerakannya atau menembakkan senjatanya karena server tidak menerima perintah tepat waktu sebelum turn berikutnya dimulai.

3. Energi

Semua bot memulai permainan dengan jumlah energi awal sebanyak 100 poin energi.

- Bot akan kehilangan energi jika ditembak atau ditabrak oleh bot musuh.
- Bot juga akan kehilangan energi jika menembakkan meriamnya.
- Bot akan mendapatkan energi jika peluru dari meriamnya mengenai musuh. Energi yang didapat akan lebih banyak 3 kali lipat dari energi yang digunakan untuk menembakkan peluru.
- Bot dengan energi nol akan dinonaktifkan dan tidak bisa bergerak. Jika bot terkena serangan dalam keadaan ini, bot akan hancur.

4. Peluru

Semakin banyak energi (daya tembak) yang digunakan untuk menembakkan peluru, semakin berat peluru tersebut dan semakin lambat gerakannya. Namun, peluru yang lebih berat juga menghasilkan lebih banyak kerusakan dan memungkinkan bot mendapatkan lebih banyak energi saat mengenai bot musuh.

Seperti disebutkan sebelumnya, peluru yang lebih berat akan bergerak lebih lambat. Ini berarti akan membutuhkan waktu lebih lama untuk mencapai target, meningkatkan risiko peluru tidak mengenai sasaran. Sebaliknya, peluru yang lebih ringan bergerak lebih cepat, sehingga lebih mudah mengenai target, tetapi peluru ringan tidak memberikan banyak poin energi saat mengenai bot musuh.

5. Panas Meriam (Gun Heat)

Saat menembakkan peluru, meriam akan menjadi panas. Peluru yang lebih berat menghasilkan lebih banyak panas dibandingkan peluru yang lebih ringan. Ketika meriam terlalu panas, bot tidak dapat menembak hingga suhu meriam turun ke nol. Selain itu, meriam juga sudah dalam keadaan panas di awal round dan perlu waktu untuk mendingin sebelum bisa digunakan untuk pertama kalinya.

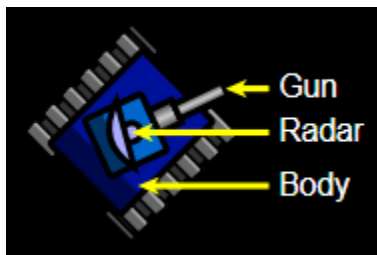
6. Tabrakan

Perlu diperhatikan bahwa bot akan menerima kerusakan jika menabrak dinding (batas arena), yang disebut wall damage. Hal yang sama juga terjadi jika bot bertabrakan dengan bot lain.

Jika bot menabrak bot musuh dengan bergerak maju, ini disebut ramming (menabrak dengan sengaja), yang akan memberikan sedikit skor tambahan bagi bot yang menyerang.

7. Bagian Tubuh Tank

Tubuh tank terdiri dari 3 bagian:



Body adalah bagian utama dari tank yang digunakan untuk menggerakkan tank.

Gun digunakan untuk menembakkan peluru dan dapat berputar bersama *body* atau independen dari *body*.

Radar digunakan untuk memindai posisi musuh dan dapat berputar bersama *body* atau independen dari *body*.

8. Pergerakan

Bot dapat bergerak maju dan mundur hingga kecepatan maksimum. Dibutuhkan beberapa giliran untuk mencapai kecepatan maksimum. Bot dapat mengalami percepatan maksimum sebesar 1 unit per giliran dan pengereman dengan perlambatan maksimum 2 unit per giliran. Percepatan dan perlambatan maksimum tidak bergantung pada kecepatan bot saat itu.

9. Berbelok

Seperti yang disebutkan sebelumnya, bagian tubuh, turret (meriam), dan radar dapat berputar secara independen satu sama lain. Jika turret atau radar tidak diputar, maka keduanya akan mengarah ke arah yang sama dengan tubuh bot.

Setiap bagian tubuh memiliki kecepatan putar yang berbeda. Radar adalah bagian tercepat dan dapat berputar hingga 45 derajat per giliran, yang berarti dapat berputar 360 derajat dalam 8 giliran. Turret dan meriam dapat berputar hingga 20 derajat per giliran.

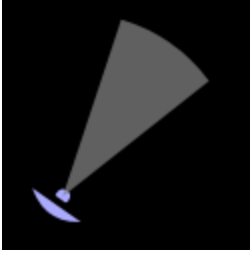
Bagian paling lambat adalah tubuh tank, yang dalam kondisi terbaik dapat berputar hingga 10 derajat per giliran. Namun, ini bergantung pada kecepatan bot saat ini. Semakin cepat bot bergerak, semakin lambat kemampuannya untuk berbelok.

Perlu diperhatikan bahwa tidak ada energi yang dikonsumsi saat bot bergerak atau berbelok.

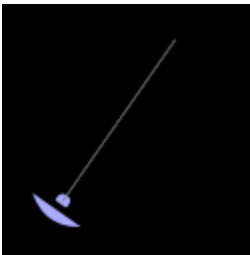
10. Pemindaian

Aspek penting dalam Robocode adalah memindai bot musuh menggunakan radar. Radar dapat mendeteksi bot dalam jangkauan hingga 1200 piksel. Musuh yang berada lebih dari 1200 piksel dari bot tidak dapat terdeteksi atau dipindai oleh radar.

Penting untuk diperhatikan bahwa sebuah bot hanya dapat memindai bot musuh yang berada dalam jangkauan sudut pemindaian (scan arc)-nya. Sudut pemindaian ini merupakan "sapuan radar" dari arah radar sebelumnya ke arah radar saat ini dalam satu giliran.



Jika radar tidak bergerak dalam suatu giliran, artinya radar tetap mengarah ke arah yang sama seperti pada giliran sebelumnya, maka sudut pemindaian akan menjadi nol derajat, dan bot tidak akan dapat mendeteksi musuh.



Oleh karena itu, sangat disarankan untuk selalu mengubah arah radar agar tetap dapat memindai musuh.

11. Skor

Pada akhir pertempuran, setiap bot akan diranking berdasarkan total skor yang diperoleh masing-masing bot selama keseluruhan pertempuran. Tentunya, tujuan utama pada tugas besar ini adalah membuat bot yang memberikan skor setinggi mungkin. Berikut adalah rincian komponen skor pada pertempuran:

- **Bullet Damage:** Bot mendapatkan **poin sebesar *damage*** yang dibuat kepada bot musuh menggunakan peluru.
- **Bullet Damage Bonus:** Apabila peluru berhasil membunuh bot musuh, bot mendapatkan **poin sebesar 20% dari *damage*** yang dibuat kepada musuh yang terbunuh.
- **Survival Score:** Setiap ada bot yang mati, bot lainnya yang masih bertahan pada ronde tersebut mendapatkan **50 poin**.
- **Last Survival Bonus:** Bot terakhir yang bertahan pada suatu ronde akan mendapatkan **10 poin** dikali dengan banyaknya musuh.
- **Ram Damage:** Bot mendapatkan **poin sebesar 2 kalinya *damage*** yang dibuat kepada bot musuh dengan cara menabrak.

- **Ram Damage Bonus:** Apabila musuh terbunuh dengan cara ditabrak, bot mendapatkan **poin sebesar 30% dari *damage*** yang dibuat kepada musuh yang terbunuh.

Skor akhir bot adalah akumulasi dari 6 komponen diatas. Perlu diperhatikan bahwa game akan menampilkan berapa kali suatu bot meraih peringkat 1, 2, atau 3 pada setiap ronde. Namun, hal ini tidak dihitung sebagai komponen skor maupun untuk perangkingan akhir. Bot yang dianggap menang pertempuran adalah bot dengan akumulasi skor tertinggi.

Untuk informasi lebih lengkap, silahkan buka dokumentasi Tank Royale pada link [berikut](#).

Starter Pack


Untuk tugas besar ini, game engine yang akan digunakan sudah dimodifikasi oleh asisten. Berikut adalah beberapa perubahan yang dibuat oleh asisten dari game engine Tank Royale:

- **Theme GUI:** diubah menjadi light theme
- **Skor & Energi:** masing-masing bot ditampilkan di samping arena permainan agar lebih mudah diamati saat pertarungan
- **Turn Limit:** Durasi pertarungan tidak akan bergantung pada banyak ronde. Pada game engine ini, pertarungan akan berakhir apabila banyaknya turn sudah mencapai batas tertentu. Apabila batasan ini tercapai, ronde otomatis langsung berakhir dan pemenang pertarungan akan ditampilkan. Turn Limit dapat diatur pada menu “setup rules”.

Source Code untuk game engine dan template bot telah disediakan pada tautan berikut.

[tubes1-if2211-starter-pack](#)

Adapun panduan mengenai cara menjalankan game engine, membuat bot, dan melihat referensi API dapat dilihat melalui tautan berikut.

 [Get Started With Robocode](#)

Spesifikasi Wajib

- Buatlah 4 bot (1 utama dan 3 alternatif) dalam bahasa **C# (.net)** yang mengimplementasikan ***algoritma Greedy*** pada *bot* permainan Robocode Tank Royale 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 memperoleh skor setinggi mungkin pada akhir

pertempuran. Hal ini dapat dilakukan dengan mengoptimalkan komponen skor yang telah dijelaskan diatas.

- Strategi *greedy* yang diimplementasikan **harus berbeda** untuk setiap bot yang diimplementasikan dan setiap strategi *greedy* harus menggunakan **heuristic** yang berbeda.
- **Bot yang dibuat TIDAK BOLEH sama dengan SAMPEL yang diberikan sebagai CONTOH. Baik dari starter pack maupun dari repository engine asli.**
- Buatlah strategi *greedy* terbaik, karena setiap **bot utama** 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.
- Setiap 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**. Artinya semua strategi harus diimplementasikan
- Mahasiswa disarankan membaca [dokumentasi](#) dari *game engine*. Perlu diperhatikan bahwa game engine yang digunakan untuk tubes ini **SUDAH DIMODIFIKASI**. Untuk Perubahan dapat dilihat pada bagian [starter pack](#)

Spesifikasi 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. **Semakin menarik video, maka semakin banyak poin yang diberikan.**
- (maks 10) Beberapa kelompok pemenang lomba kompetisi akan mendapatkan nilai tambahan berdasarkan posisi yang diraih.
- 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”.

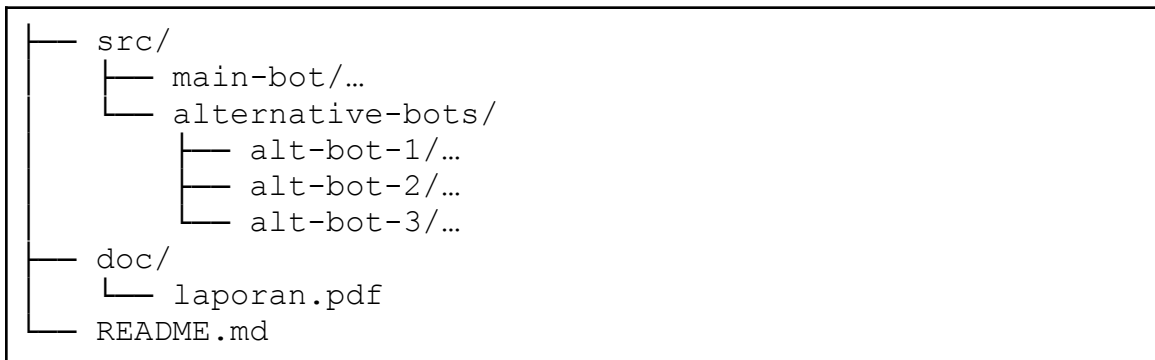
- Bot yang telah dibuat akan dikompertisikan dengan kelompok lain dan disaksikan oleh seluruh peserta kuliah. Terdapat hadiah menarik bagi kelompok yang memenangkan kompetisi.
- Setiap kelompok harap mengisi nama kelompok dan anggotanya pada link berikut, paling lambat **Kamis, 6 Maret pukul 22.11 WIB**.

[📄 Pendataan Kelompok Tubes 1 Stima 2024/2025](#)

- 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 **Kamis, 6 Maret 2025 pukul 22.11 WIB**.

[📄 Pendataan Kelompok Tubes 1 Stima 2024/2025](#)

- Program disimpan dalam repository yang bernama **Tubes1_NamaKelompok** dengan nama kelompok sesuai dengan yang di sheets diatas. Berikut merupakan struktur dari isi repository tersebut:



- Folder src berisi semua source code
 - Folder doc berisi laporan tugas besar dengan format **NamaKelompok.pdf** dengan ketentuan isi pada [isi laporan](#)
 - README untuk tata cara penggunaan yang minimal berisi:
 - Penjelasan singkat algoritma greedy yang diimplementasikan untuk setiap bot yang dibuat
 - Requirement program dan instalasi tertentu bila ada
 - Command atau langkah-langkah dalam meng-compile atau build program
 - Author (identitas pembuat)
- Sangat disarankan untuk menggunakan [semantic commit](#). Buatlah release dengan format **v1.x** dengan x adalah nomor revisi dimulai dari revisi 0. Contoh v1.0 untuk release pertama, v1.1 untuk revisi selanjutnya.
 - Pastikan untuk membuat repository bersifat **Public** paling lambat **H+1 deadline (1 hari setelah deadline)**. Sebelum deadline repository harus bersifat **Private**.

- Laporan dikumpulkan hari **Senin, 24 Maret 2025** pada alamat Google Form berikut paling lambat **pukul 22.11 WIB**:
<https://bit.ly/tubes1stima25>.
PERINGATAN: Keterlambatan akan mengurangi nilai sebanyak 1 poin untuk setiap menit keterlambatan.
- Adapun pertanyaan terkait tugas besar ini bisa disampaikan melalui QnA berikut:
<https://bit.ly/QnA-Stima-25>.

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 Robocode Tank Royale menjadi elemen-elemen algoritma Greedy (himpunan kandidat, himpunan solusi, fungsi solusi, fungsi seleksi, fungsi kelayakan, fungsi objektif)
 - Eksplorasi **4 alternatif solusi** greedy yang mungkin dipilih dalam persoalan Robocode Tank Royale. (setiap alternatif harus memiliki **heuristic** yang **berbeda**)
 - Analisis efisiensi dan efektivitas dari kumpulan alternatif solusi greedy yang dirumuskan
 - Strategi *greedy* **yang dipilih** dari 4 alternatif solusi greedy (yang akan diimplementasikan dalam program) beserta alasan dan pertimbangan pemilihan strategi tersebut.
- **Bab 4:** Implementasi dan pengujian.
 - Implementasi 4 alternatif solusi (pseudocode yang cukup detail sehingga **dapat dimengerti dengan mudah**. Berikan komen pada pseudocode jika merasa perlu)
 - Penjelasan struktur data, fungsi, dan prosedur yang digunakan pada bot dengan solusi greedy **yang dipilih**.
 - Pengujian dilakukan minimal sebanyak 3 kali berupa pertandingan antara bot utama (**yang dipilih**) dengan **bot asisten** (format 1 vs 1). Pengujian yang dicantumkan sebaiknya mencakup semua kejadian yang unik.

- Analisis hasil dari pengujian. Misalnya adalah apakah strategi *greedy* berhasil mendapatkan nilai optimal, lalu jika tidak, dalam kondisi seperti apa strategi *greedy* tidak berhasil mendapatkan nilai optimal, dsb.
- **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 Tank Royale ke dalam elemen-elemen algoritma *greedy* (10%)
- c. Eksplorasi alternatif-alternatif solusi persoalan Tank Royale 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)

Perhatian

- **Dilarang keras copy paste** program dari internet, AI, repository lain, ataupun program milik teman. Program harus dibuat sendiri, kecurangan akan diberikan sanksi berat yaitu nilai tugas menjadi **nol**.
- Pastikan program **dapat dikompilasi setidaknya** pada *windows* dan *linux*.
- Apabila program **tidak dapat dijalankan** maka tidak akan dinilai oleh asisten.
- **Keterlambatan pengumpulan akan mengurangi 1 poin untuk setiap menit keterlambatan. Harap diperhatikan karena masih banyak yang telat mengumpulkan pada tugas kecil 1.**
- Tambahkan tabel berikut yang diisi *checklist* (✓) pada bagian lampiran laporan Anda.

No	Poin	Ya	Tidak
1	Bot dapat dijalankan pada Engine yang sudah dimodifikasi asisten.		
2	Membuat 4 solusi greedy dengan heuristic yang berbeda.		
3	Membuat laporan sesuai dengan spesifikasi.		
4	Membuat video bonus dan diunggah pada Youtube.		

Referensi

“Robocode Tank Royale”

<https://github.com/robocode-dev/tank-royale>

Engine asli **sebelum** di modifikasi asisten.

“Robocode Tank Royale Docs”

<https://robocode-dev.github.io/tank-royale/>

Dokumentasi lengkap mengenai tank royale.

“C# .Net”

<https://dotnet.microsoft.com/en-us/>

Website resmi untuk .Net dari Microsoft.

“Learn to code C#”

<https://dotnet.microsoft.com/en-us/learntocode>

Tutorial bahasa C# dari situs resmi Microsoft yang cukup friendly.

"C# language documentation"

<https://learn.microsoft.com/en-us/dotnet/csharp/>

Dokumentasi bahasa C# dari microsoft.

"What is an API?"

<https://aws.amazon.com/what-is/api/>

Penjelasan apa itu API bagi yang memerlukan.

--- Selamat Mengerjakan! ---

"Bikin botnya buat tubes aja yaa, jangan buat ~~buzzer~~"

— Farhan —

"Warm up dulu gaess"

— Ciko —

"Tembak tembak dor dor dor"

— Haikal —

"Ditunggu bot-bot uniknyaa"

— Ariel —

"(TT~TT)"

— Rafi —

"My enemies are many, my equal are none"

— Aland —

"Asik ye"

— Eka —

"marhaban ya ramadhan"

— Ikhwan —