

LAPORAN TUGAS BESAR I

IF2211 - STRATEGI ALGORITMA



Disusun oleh :

“Ramadhan Kareem”

Muhammad Raihan Nazhim Oktana – 13523021

Faqih Muhammad Syuhada – 13523057

Barru Adi Utomo – 13523101

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

2025

Daftar Isi

Daftar Isi.....	2
BAB I DESKRIPSI TUGAS.....	3
1.1. Deskripsi Permainan.....	3
1.2. Deskripsi Tugas.....	3
1.2.1. Spesifikasi Wajib.....	3
1.2.2. Spesifikasi Bonus.....	4
BAB II LANDASAN TEORI.....	5
2.1. Algoritma Greedy.....	5
2.2. Game Engine.....	6
2.3. Komponen Program Robocode Tank Royale.....	6
2.4. Alur Jalan Program Robocode Tank Royale.....	6
2.5. Cara Menjalankan Bot dan Implementasi Algoritma Greedy pada Bot.....	7
2.6. Mekanisme Teknis Permainan Robocode Tank Royale.....	7
BAB III APLIKASI STRATEGI GREEDY.....	9
3.1. Program Utama.....	9
3.2. Program Bot Alternatif 1.....	9
3.3. Program Bot Alternatif 2.....	10
3.4. Program Bot Alternatif 3.....	10
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	11
4.1. Implementasi.....	11
4.1.1. Bot Utama.....	11
4.1.2. Bot Alternatif 1.....	12
4.1.3. Bot Alternatif 2.....	14
4.1.4. Bot Alternatif 3.....	16
4.2. Pengujian.....	17
4.2.1. Pengujian 4 Bot Terpilih.....	17
4.2.2. Pengujian Bot Utama Dengan 3 Sample Bot.....	17
4.2.3. Pengujian 4 Bot Terpilih Dengan 4 Sample Bot.....	18
4.2.4. Analisis Hasil Pengujian.....	19
BAB V KESIMPULAN DAN SARAN.....	20
5.1 Kesimpulan.....	20
5.2 Saran.....	20
LAMPIRAN.....	21
DAFTAR PUSTAKA.....	22

BAB I

DESKRIPSI TUGAS

1.1. Deskripsi Permainan

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 hingga hanya tersisa satu tank 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.

1.2. Deskripsi Tugas

1.2.1. 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](#)

1.2.2. 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.

BAB II

LANDASAN TEORI

2.1. Algoritma Greedy

Algoritma greedy adalah metode sederhana yang cukup populer dalam menyelesaikan masalah optimasi. Algoritma ini beroperasi dengan memilih pilihan terbaik pada setiap langkahnya tanpa mempertimbangkan implikasi di masa depan, dengan keyakinan bahwa memilih opsi terbaik di setiap langkah akan membawa kita menuju solusi optimal secara keseluruhan. Persoalan optimasi yang diselesaikan oleh algoritma greedy dapat dibagi menjadi dua jenis: maksimasi dan minimasi. Algoritma ini didasarkan pada pengambilan keputusan lokal yang optimal pada setiap langkahnya, tanpa memperhatikan kemungkinan konsekuensi di langkah-langkah berikutnya. Sebagai contoh, dalam persoalan penukaran uang, algoritma greedy akan memilih koin dengan nilai tertinggi yang tersedia pada setiap iterasi.

Algoritma greedy melibatkan beberapa elemen penting, antara lain himpunan kandidat (kumpulan pilihan yang tersedia untuk dipilih pada setiap langkah), himpunan solusi (kumpulan pilihan yang sudah dipilih), fungsi solusi (yang memeriksa apakah himpunan kandidat memberikan solusi yang diinginkan), fungsi seleksi (memilih kandidat berdasarkan strategi greedy), fungsi kelayakan (memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi), dan fungsi objektif (memaksimalkan atau meminimalkan nilai yang ingin dicapai).

Meskipun algoritma greedy sering kali dapat memberikan solusi hampiran yang baik dalam waktu komputasi yang relatif singkat dibandingkan dengan algoritma yang eksak, membuktikan optimalisasinya secara matematis bisa menjadi tantangan. Terkadang, algoritma ini dapat menghasilkan solusi yang tidak optimal tergantung pada sifat dari masalah yang diselesaikan. Meskipun demikian, algoritma greedy tetap menjadi alat yang berguna dalam menyelesaikan masalah optimasi dengan cepat, meskipun dengan risiko tidak mendapatkan solusi optimal dalam beberapa kasus.

Secara keseluruhan, algoritma greedy merupakan alat yang sangat berguna dalam menyelesaikan berbagai masalah optimasi. Meskipun tidak selalu menghasilkan solusi 7 optimal, algoritma ini seringkali memberikan solusi yang baik dengan kecepatan yang optimal pula. Karena itu, algoritma greedy menjadi sangat penting bagi para pemrogram dalam menciptakan solusi yang efektif dan efisien. Dalam konteks pembuatan program bot dalam permainan, penggunaan algoritma greedy membantu dalam mendapatkan aksi bot yang paling efektif dan efisien.

Algoritma greedy terdiri dari beberapa komponen atau elemen yang penting yakni:

- a. Himpunan Kandidat (C)

Himpunan ini berisi kandidat yang akan dipilih pada setiap Langkah (misal: simpul/sisi di dalam graf, job, task, koin, benda, karakter, dsb).

- b. Himpunan Solusi (S)

Himpunan yang berisi kandidat yang sudah dipilih.

- c. Fungsi Solusi
Fungsi ini akan menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi.
- d. Fungsi Seleksi
Fungsi yang akan memilih kandidat berdasarkan strategi greedy tertentu. Strategi greedy ini bersifat heuristik.
- e. Fungsi Kelayakan
Fungsi akan memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi (layak atau tidak).
- f. Fungsi Objektif
Fungsi ini menjelaskan apakah untuk memaksimumkan atau meminimumkan.

2.2. Game Engine

Game Engine adalah sebuah sistem perangkat lunak yang berperan dalam mengembangkan dan menciptakan permainan video game. Sebagai library, game engine menyediakan fungsi-fungsi inti yang digunakan untuk mempermudah pengembang permainan dalam proses pembuatan game. Penggunaan game engine tidak hanya terbatas untuk satu permainan saja, namun pengembang game dapat membuat lebih dari satu permainan dengan mengoptimalkan proses pengembangan game melalui adaptasi atau pemanfaatan game engine yang telah ada sebelumnya. Game engine biasanya juga disebut dengan Software Development Environment (SDE) yang membantu pengembang dalam pembuatan video game. Terdiri dari berbagai komponen perangkat keras dan lunak seperti 2D dan 3D graphics rendering, animasi, Artificial Intelligence (AI), manajemen memori, audio, dan lain-lain, game engine memfasilitasi seluruh proses desain dan pengembangan video game. Dengan menggunakan game engine sebagai pondasi awal, pengembang dapat mengurangi biaya pengembangan video game karena tidak perlu membuat semua komponen dari nol. Selain itu, satu game engine dapat digunakan untuk mengembangkan berbagai jenis video game tanpa melakukan banyak perubahan.

2.3. Komponen Program Robocode Tank Royale

Program ini terdiri dari tiga komponen yaitu game engine, bot starter pack, dan bot template. Ketiga komponen dari permainan Robocode Tank Royale ini bersumber dari <https://github.com/Ariel-HS/tubes1-if2211-starter-pack/releases/tag/v1.0> yang game enginenya sendiri sudah disesuaikan agar sesuai spesifikasi tubes ini.

2.4. Alur Jalan Program Robocode Tank Royale

Untuk menjalankan permainan bot ini, diperlukan beberapa tools yang perlu diinstal terlebih dahulu sebelum menjalankan program yaitu java untuk menjalankan server.jar (Robocode Tank Royale Game Server), dan dotnet(.NET) untuk mengembangkan bot menggunakan C#, dan NuGet Package: Robocode.TankRoyale.BotApi sebagai library komunikasi antara bot dan server.. Setelah dependensi terinstal, selanjutnya kita dapat melakukan

build dengan menjalankan perintah `./gradlew :gui-app:build`, clean jika bukan pertama kali build dengan perintah `./gradlew :gui-app:clean`, dan kemudian menjalankan game engine dengan perintah `java -jar ./gui-app/build/libs/robocode-tankroyale-gui-0.30.0.jar`. Jika berhasil, kita akan melihat tampilan terminal sesuai dengan yang diharapkan, dan gui terbuka setelah perintah dijalankan.

2.5. Cara Menjalankan Bot dan Implementasi Algoritma Greedy pada Bot.

Untuk menjalankan bot, pastikan komputer telah terinstall dengan dotnet(.NET) dan memiliki semua persyaratan yang tercantum. Setelah mengunduh source code bot starter pack dan mengekstraknya, buka terminal dan navigasikan ke direktori root proyek bot. Untuk setiap bot sesuaikan dotnet(.NET) (dapat dilihat di dalam file bereksistensi `.csproj`) sesuai versi yang digunakan pada perangkat. Jika dotnet yang digunakan di perangkat berbeda dengan yang ada di *starter pack*, maka folder bin dan obj harus di hapus karena dapat berkemungkinan berisi yang berbeda atau *error*, lalu jalankan `<nama bot>.cmd` didalam direktori bot tersebut dengan cara menjalankan perintah `./<nama bot>`.

Untuk mengimplementasikan bot baru kita bisa menggunakan template bot yang sudah disediakan di *starter pack*, di dalam file dengan eksistensi `.cs` kita dapat menentukan logik yang kita inginkan, lalu didalam file dengan eksistensi `.json` kita dapat menentukan info dari bot tersebut. Implementasi dari bot sendiri menginherit kelas Bot yang api nya berasal dari Robocode.TankRoyale.BotApi;

2.6. Mekanisme Teknis Permainan Robocode Tank Royale

Mekanisme dan Teknis Permainan Robocode Tank Royale sendiri yaitu setiap pemain mempunyai kode yang sudah di tulisnya untuk tank virtual yang bertarung dalam arena dua dimensi. Pemain tidak memiliki kendali langsung atas bot saat permainan berlangsung, sehingga kode yang ditulis merupakan kode yang menentukan bagaimana tank bergerak, memindai musuh, menembak, dan bereaksi terhadap berbagai peristiwa dalam pertempuran ini.

Permainan ini merupakan permainan berbasis server local, sehingga setiap aksi yang dilakukan mulai dari mendaftarkan bot hingga menjalankan aksi bot akan memerlukan request terhadap API *endpoint* tertentu yang disediakan oleh backend dalam hal ini adalah dotnet dalam bahasa C#. Berikut adalah urutan request yang terjadi dari awal mula permainan :

1. Registrasi Bot, ketika permainan dimulai, setiap bot perlu mengirimkan *request* untuk didaftarkan ke server. Proses ini akan mengirimkan data seperti info bot hingga kemampuan bot tersebut. Server kemudian mengembalikan ID unik yang digunakan bot untuk berkomunikasi selama permainan berlangsung.
2. Inisialisasi permainan. Setelah semua bot didaftarkan. Server mengirimkan semua informasi yang di kemungkinan digunakan bot, seperti ukuran arena atau kondisi awal permainan. Bot menerima data ini dan menyimpannya sebagai referensi yang mungkin digunakan dalam permainan.

3. Pertukaran Data Bot dan Server secara Realtime, selama pertempuran setiap bot mengirimkan kepada server melalui request API. perintah ini dapat berupa pergerakan bot, pengaturan radar dan meriam, atau penembakan meriam. Server kemudian dapat mengembalikan berbagai respon yang diberi oleh api berisi status permainan terbaru.
4. Event Handling, Selanjutnya setelah bot menerima berbagai status/event dari server maka bot dapat member respon, eventnya sendiri berbagai macam diantaranya seperti OnScannedRobot yaitu ketika membaca bot, OnHitByBullet yaitu ketika bot terkena peluru, OnHitWall yaitu ketika bot menabrak dinding, dan OnHitRobot yaitu saat bot bertabrakan dengan bot lain. Untuk responnya sendiri ada yang default maupun di override.
5. Scoring, Setelah pertempuran selesai server menghitung skor berdasarkan performa setiap bot dalam setiap round. Ada jumlah serangan yang berhasil, banyak menabrak bot lain dan mati atau tidaknya dalam keberjalanan suatu round. Hasil ini dikirimkan dalam bentuk tabel yang setiap bot memiliki atributnya masing-masing.

BAB III

APLIKASI STRATEGI GREEDY

3.1. Program Utama

Bot Utama ini menggunakan strategi greedy yang sangat simple. Pertama untuk pergerakan dasarnya ia hanya akan maju atau mundur sesuai keadaan dan jika ia maju ia akan mengatur akan belok kanan sebanyak 30 derajat atau 30 derajat ke kiri jika mundur yang keduanya dieksekusi ketika On Hit Event di trigger. Untuk tembakannya sendiri menggunakan kecepatan maksimum gun, sehingga radar dan gun bekerja dengan berbarengan. Untuk pergerakan event seperti hit wall ia menggunakan belok 90 derajat kemudian untuk hit by bullet dia akan merubah arah pergerakannya saja yang awalnya maju ia akan mundur. Selanjutnya untuk event hit bot ia akan mencari darimana arah dia menabrak bot, dan jika jaraknya kurang dari 10 derajat maka bot akan menembak jika tidak ia akan belok ke kiri lalu kabur dari arah sebaliknya.

3.2. Program Bot Alternatif 1

Bot Alternatif 1, menggunakan strategi greedy dengan memaksimalkan poin yang akan didapat dengan berusaha bertahan atau survive selama mungkin dan mengatur penggunaan peluru seefektif mungkin. Pergerakan bot diatur dengan menggunakan konsep bentuk fungsi sinusoidal yang didasarkan pada nilai waktu, frekuensi gelombang sebesar 0,20 satuan, dan amplitudo gelombang sebesar 30,0 satuan. Perancangan pergerakan bot dibuat sedemikian sehingga susah ditembak ataupun ditabrak oleh musuh karena kebanyakan trajektori peluru ataupun pergerakan bot adalah lurus sedangkan ini cenderung agak melengkung. Kemudian, untuk mengurangi poin yang terbuang sia-sia akibat menabrak tembok, pergerakan khusus juga ditambahkan saat bot berada pada area pinggir map dengan jarak 50 satuan untuk bermanuver berdasarkan sudut yang ditentukan dari posisi (X dan Y)-nya di saat tersebut. Adapun manuver kecil berbentuk arc atau lengkungan kecil sebesar 90 derajat yang dilakukan bertahap jika bot terkena tembakan agar tidak terkena tembak lagi. Serta, saat menabrak bot musuh langsung coba untuk bergeser atau mencari jalan lain yang kosong. Kemudian, untuk konsep pengefektifan penggunaan peluru dan pemanfaatan energi yang optimal, bot diberi skema penembakan berdasarkan jarak antara bot kita dengan bot musuh yang terdeteksi di radar dan kondisi jumlah energi yang dimiliki saat ini. Pada penerapannya, semakin jauh targetnya maka akan semakin kecil pelurunya, begitupun sebaliknya, semakin dekat targetnya maka akan semakin besar pelurunya. Pada kasus bot mengalami minim energi yang sudah tidak bisa apa-apa lagi, bot akan dicegah untuk menembak dan berusaha untuk bermanuver sebaik mungkin sebagai upaya pemaksimalan poin di akhir permainan. Bot ini akan cukup konsisten hampir di berbagai situasi, unggul saat bot-bot lain tidak spesifik melawan bot ini, namun kurang saat sebaliknya.

3.3. Program Bot Alternatif 2

Bot Alternatif 2, menggunakan strategi greedy menargetkan lawan hingga radar lepas dari lawan dengan tujuan mendapatkan sebanyak point tanpa perlu menghabiskan waktu untuk menggerakkan radar secara penuh (360 derajat). Tujuannya sendiri yaitu selama bot terbaca, ada kemungkinan bahwa bot masih berada di sekitar tempat yang dibaca diawal. Selain itu ada juga greedy yang digunakan untuk penembakan, dimana peluru yang ditembakkan sesuai dengan jarak dan ketersediaan energi dari bot tersebut. Semakin besar energi dan semakin dekat dengan lawan, semakin besar peluru yang ditembakkan. Untuk pergerakan awalnya menggunakan greedy random by turn, dimana pergerakannya sesuai sudah berapa banyak turn yang berjalan. Dimana terdiri dari maju serong kanan, maju serong kiri, mundur serong kiri, mundur serong kanan. Hal ini bertujuan untuk membuat bot tidak di titik horizontal dan vertikal yang sama sehingga tembakan yang diarahkan ke bot dapat terhindarkan. Bot ini juga memberi respon jika menabrak dinding yaitu membalikkan target kecepatan bot sehingga bergerak berlawanan arah dan jika terkena peluru bot akan menghindari dengan merotasi badan tanknya sebanyak 5 derajat.

3.4. Program Bot Alternatif 3

Pada bot alternatif 3, digunakan strategi greedy yaitu menembakkan peluru sebesar 3 tegak lurus dengan ArenaHeight. Tujuan menggunakan peluru sebesar tiga adalah semakin besar bobot peluru, akan semakin lama juga pergerakannya sehingga kemungkinan bot lain untuk tidak sengaja bergerak ke arah peluru tersebut meningkat. Strategi tersebut termasuk dalam kategori greedy karena menggunakan peluru berbobot tiga berharap mengenai bot yang di tengah arena. Untuk memaksimalkan lintasan peluru, maka bot ini akan menghadap sejajar dengan nilai maksimal antar ArenaHeight dan ArenaWidth. Bot ini akan bergerak secara konstan dari ujung wall ke ujung wall untuk menghindari bullet dari lawan karena untuk kemungkinan besar, peluru akan berjalan lebih lama dengan pergerakan bot ini. Selain itu, bot akan bergerak ke arah berlawanan jika mengenai bot lawan, terkena peluru, dan menabrak wall. Namun, jika bot menabrak wall, maka energi bot tersebut akan berkurang, oleh karena itu digunakan jarak pergerakan dikurangi sedikit agar tidak kena dengan wall.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi

Berikut merupakan implementasi kode C# setiap bot yang sudah dijelaskan pada Bab III:

4.1.1. Bot Utama

Flowers
<pre> Class FlowersBot inherits from Bot: movingForward <-- true Constructor(botInfo): Call base constructor with botInfo Function Main(): configBuilder <-- create configuration builder Set base path to current directory Add JSON file "Flowers.json" config <-- build configuration botInfo <-- create BotInfo from config bot <-- new FlowersBot(botInfo) bot.Start() Function Run(): BodyColor <-- Pink TurretColor <-- Purple RadarColor <-- Pink ScanColor <-- Pink BulletColor <-- Black GunTurnRate <-- 50 While IsRunning: MaxSpeed <-- 8 If movingForward: SetTurnRight(30) Forward(500) Else: SetTurnLeft(30) Back(500) Event OnScannedBot(event): </pre>

```

    Fire(3)

Event OnHitBot(event):
    bearing <-- BearingTo(event.X, event.Y)
    If bearing > -10 AND bearing < 10:
        Fire(3)

    If event.IsRammed:
        TurnLeft(10)
        movingForward <-- NOT movingForward

Event OnHitByBullet(event):
    movingForward <-- NOT movingForward

```

4.1.2. Bot Alternatif 1

Freeze
<pre> Class Freeze inherits from Bot: Constructor(botInfo): Call base constructor with botInfo Function Main(): configBuilder <-- create configuration builder Set base path to current directory Add JSON file "Freeze.json" config <-- build configuration botInfo <-- create BotInfo from config bot <-- new Freeze(botInfo) bot.Start() Function Run(): Set colors: BodyColor <-- LightBlue TurretColor <-- Blue RadarColor <-- Yellow ScanColor <-- Orange BulletColor <-- Black Set GunTurnRate <-- 20.0 Initialize: time <-- 0.00 amplitude <-- 30.0 frequency <-- 0.20 </pre>

```

    safeDistance <-- 50.0

While IsRunning:
    If X position < safeDistance:
        target <-- 270 if Y < half ArenaHeight else 90
        TurnToHeading(target)
        SetForward(30)
        Execute movement
    Else If X position > ArenaWidth - safeDistance:
        target <-- 270 if Y < half ArenaHeight else 90
        TurnToHeading(target)
        SetForward(30)
        Execute movement
    Else If Y position < safeDistance:
        target <-- 0 if X < half ArenaWidth else 180
        TurnToHeading(target)
        SetForward(30)
        Execute movement
    Else If Y position > ArenaHeight - safeDistance:
        target <-- 0 if X < half ArenaWidth else 180
        TurnToHeading(target)
        SetForward(30)
        Execute movement
    Else:
        Turn angle <-- amplitude * Sin(time * frequency)
        SetTurnRight(Turn angle)
        SetForward(30)
        Execute movement
        Increment time by 1.0

Event OnScannedBot(event):
    distance <-- DistanceTo(event.X, event.Y)
    energy <-- Energy
    If energy < 1.5:
        Return
    ElseIf energy <= 5:
        Fire(1)
    ElseIf energy <= 15:
        Fire(1.5)
    ElseIf energy <= 40:
        If distance < 150:
            Fire(2.5)
        Else:
            Fire(1.5)
    Else:
        If distance < 100:

```

```

        Fire(3)
    ElseIf distance < 200:
        Fire(2.5)
    Else:
        Fire(1.5)

Event OnHitWall(event):
    Back(90)
    TurnLeft(135)
    Forward(90)

Event OnHitBot(event):
    Back(60)
    TurnRight(30)
    Forward(60)

Event OnHitByBullet(event):
    Loop from 1 to 9:
        SetTurnLeft(10)
        SetForward(10)
        Execute movement

Private Function TurnToHeading(target):
    currentDirection <-- Direction
    Turn angle <-- NormalizeAngle(target - currentDirection)
    If Turn angle < 0:
        TurnLeft(-Turn angle)
    Else:
        TurnRight(Turn angle)

Private Function NormalizeAngle(angle):
    While angle > 180:
        angle -= 360
    While angle < -180:
        angle += 360
    Return angle

```

4.1.3. Bot Alternatif 2

Flows
<p>Class Flows inherits from Bot:</p> <p>Declare turnCounter as integer</p> <p>Declare movingForward as boolean</p>

```

Function Main():
    Create a new Flows bot
    Start the bot

Constructor():
    Call base constructor with BotInfo from file "Flows.json"

Function Run():
    Set bot colors:
        BodyColor <-- Purple
        TurretColor <-- Blue
        RadarColor <-- Cyan
        BulletColor <-- Black
        ScanColor <-- Purple

    Initialize turnCounter to 0
    Initialize movingForward to true
    Set GunTurnRate to 45
    Set TurnRate to 5

    While IsRunning:
        If turnCounter modulo 64 equals 0:
            TurnRate <-- 5
            TargetSpeed <-- 8 if movingForward, else -8
        If turnCounter modulo 64 equals 32:
            TurnRate <-- -5
            TargetSpeed <-- -8 if movingForward, else 8
        If turnCounter modulo 128 equals 0:
            Toggle movingForward

        Increment turnCounter
        Execute movement with Go()

Event OnScannedBot(event):
    distance <-- DistanceTo(event.X, event.Y)
    energy <-- Energy
    If energy < 1.5:
        Return
    ElseIf energy <= 5:
        Fire(1)
    ElseIf energy <= 15:
        Fire(1.5)
    ElseIf energy <= 40:
        If distance < 150:
            Fire(2.5)
    Else:

```

```

        Fire(1.5)
    Else:
        If distance < 100:
            Fire(3)
        ElseIf distance < 200:
            Fire(2.5)
        Else:
            Fire(1.5)

    GunTurnRate <-- -GunTurnRate

    Event OnHitWall(event):
        TargetSpeed <-- -TargetSpeed

    Event OnHitByBullet(event):
        TurnRate <-- 5

```

4.1.4. Bot Alternatif 3

Bot Pinggiran
<pre> Class BotPinggiran inherits from Bot: turnDirection <-- 1 moveAmount <-- uninitialized Function Main(): bot <-- new BotPinggiran() bot.Start() Constructor BotPinggiran(): Call base constructor with "BotPinggiran.json" Function Run(): turnDirection <-- 1 moveAmount <-- Maximum of ArenaWidth and ArenaHeight TurnRight(Direction % 180) Forward(moveAmount) TurnRight(90) TurnGunRight(90) While IsRunning: Forward((ArenaHeight * turnDirection) - 20) </pre>


```

Event OnScannedBot(event):
    Fire(3)

Event OnHitBot(event):
    turnDirection <-- turnDirection * -1

Event OnHitByBullet(event):
    turnDirection <-- turnDirection * -1

Event OnHitWall(event):
    turnDirection <-- turnDirection * -1

```

4.2. Pengujian

4.2.1. Pengujian 4 Bot Terpilih

Pada pengujian ini, diuji permainan dengan 4 bot (1v1v1v1), yaitu Flowers, Freeze, Flows, dan Bot Pinggiran sebagai 4 bot terbaik yang telah dibuat. Hasilnya sebagai berikut:

Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bo...	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	Flowers 1.0	1333	600	90	576	41	25	0	3	1	1
2	Freeze Bot 1.0	1013	500	60	422	17	13	0	2	2	1
3	Bot Pinggiran 1.0	993	600	30	336	16	11	0	1	2	3
4	Flows 1.0	882	400	30	346	14	79	13	1	2	2

Gambar 1. Hasil Pengujian 1 Atas 4 Bot Terpilih

Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bo...	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	Flowers 1.0	1498	650	90	672	58	28	0	3	1	2
2	Flows 1.0	988	550	60	335	13	30	0	2	3	1
3	Bot Pinggiran 1.0	729	400	30	256	25	17	0	2	2	1
4	Freeze Bot 1.0	487	200	0	261	6	20	0	0	1	3

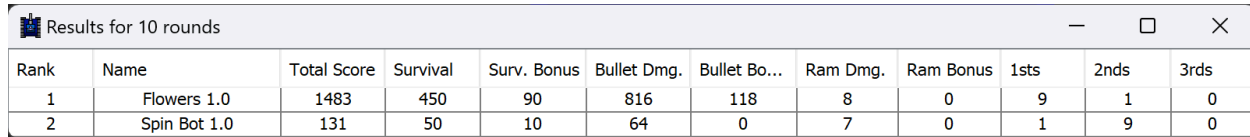
Gambar 2. Hasil Pengujian 2 Atas 4 Bot Terpilih

Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bo...	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	Freeze Bot 1.0	1606	850	150	557	25	23	0	5	0	3
2	Flowers 1.0	1278	500	30	640	64	43	0	2	2	3
3	Flows 1.0	1069	500	30	463	19	56	0	1	4	1
4	Bot Pinggiran 1.0	528	250	0	256	0	22	0	0	2	2

Gambar 3. Hasil Pengujian 3 Atas 4 Bot Terpilih

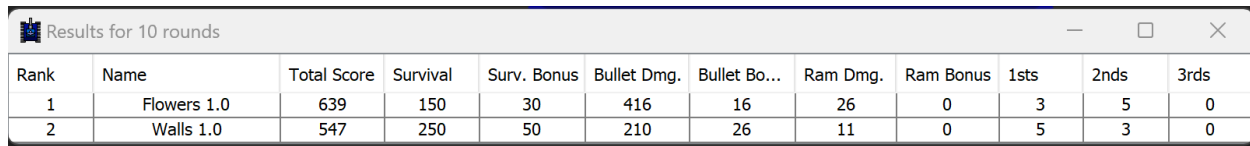
4.2.2. Pengujian Bot Utama Dengan 3 Sample Bot

Pada pengujian ini, diuji permainan dengan 4 bot (1v1), yaitu Flowers, Velocity Bot, Spin Bot, dan Walls sebagai 3 bot terbaik dari sample bot yang disediakan. Hasilnya sebagai berikut:



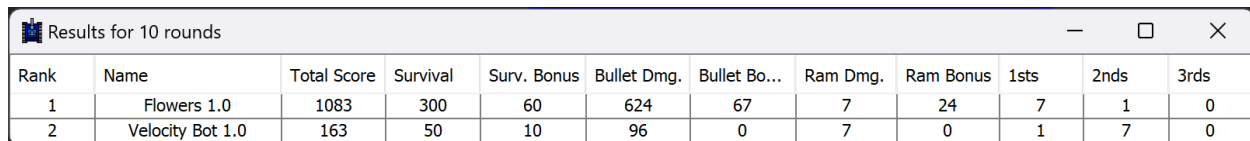
Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bo...	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	Flowers 1.0	1483	450	90	816	118	8	0	9	1	0
2	Spin Bot 1.0	131	50	10	64	0	7	0	1	9	0

Gambar 4. Hasil Pengujian 1 Atas Bot Utama Dengan 3 Sample Bot (Spin)



Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bo...	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	Flowers 1.0	639	150	30	416	16	26	0	3	5	0
2	Walls 1.0	547	250	50	210	26	11	0	5	3	0

Gambar 5. Hasil Pengujian 2 Atas Bot Utama Dengan 3 Sample Bot (Walls)

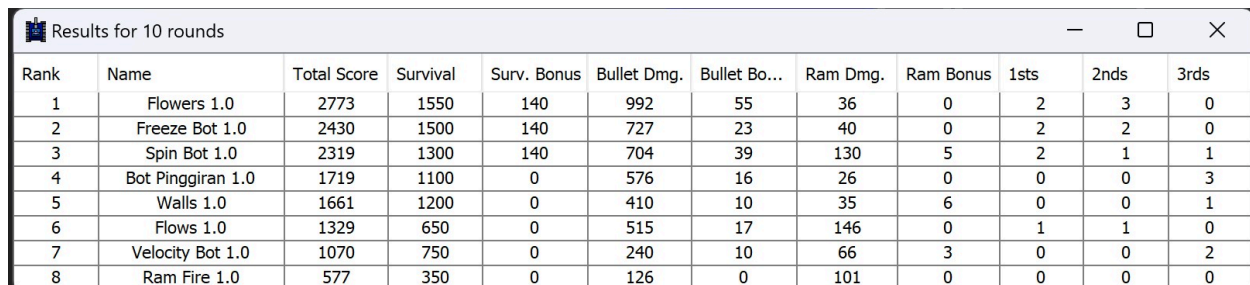


Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bo...	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	Flowers 1.0	1083	300	60	624	67	7	24	7	1	0
2	Velocity Bot 1.0	163	50	10	96	0	7	0	1	7	0

Gambar 6. Hasil Pengujian 3 Atas Bot Utama Dengan 3 Sample Bot (Velocity)

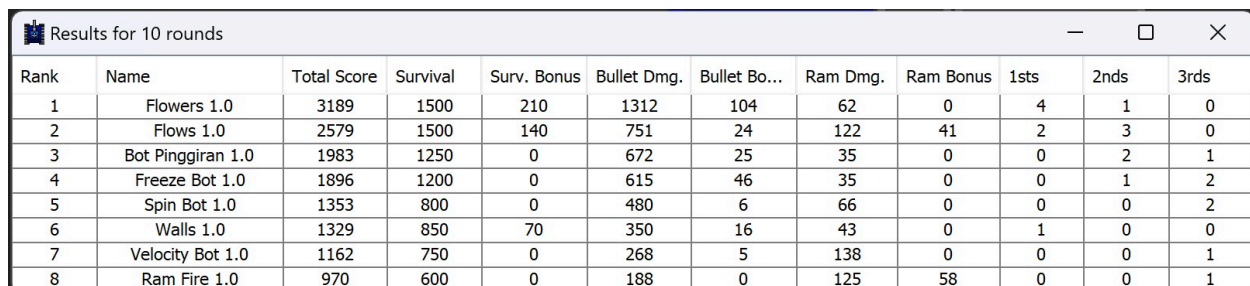
4.2.3. Pengujian 4 Bot Terpilih Dengan 4 Sample Bot

Pada pengujian ini, diuji permainan dengan 8 bot (1v1v1v1v1v1v1v1), yaitu Flowers, Freeze, Flows, Bot Pinggiran, Velocity Bot, Spin Bot, Walls, dan Ram Fire sebagai 4 bot terbaik dari sample bot yang disediakan. Hasilnya sebagai berikut:



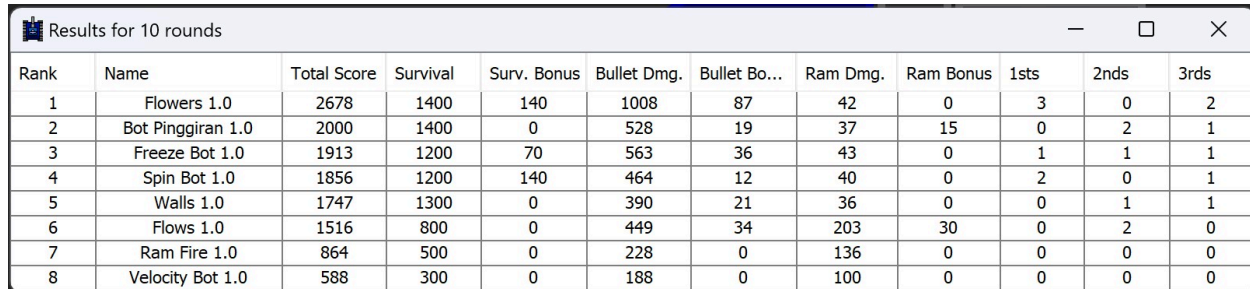
Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bo...	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	Flowers 1.0	2773	1550	140	992	55	36	0	2	3	0
2	Freeze Bot 1.0	2430	1500	140	727	23	40	0	2	2	0
3	Spin Bot 1.0	2319	1300	140	704	39	130	5	2	1	1
4	Bot Pinggiran 1.0	1719	1100	0	576	16	26	0	0	0	3
5	Walls 1.0	1661	1200	0	410	10	35	6	0	0	1
6	Flows 1.0	1329	650	0	515	17	146	0	1	1	0
7	Velocity Bot 1.0	1070	750	0	240	10	66	3	0	0	2
8	Ram Fire 1.0	577	350	0	126	0	101	0	0	0	0

Gambar 7. Hasil Pengujian 1 Atas 4 Bot Terpilih Dengan 4 Sample Bot



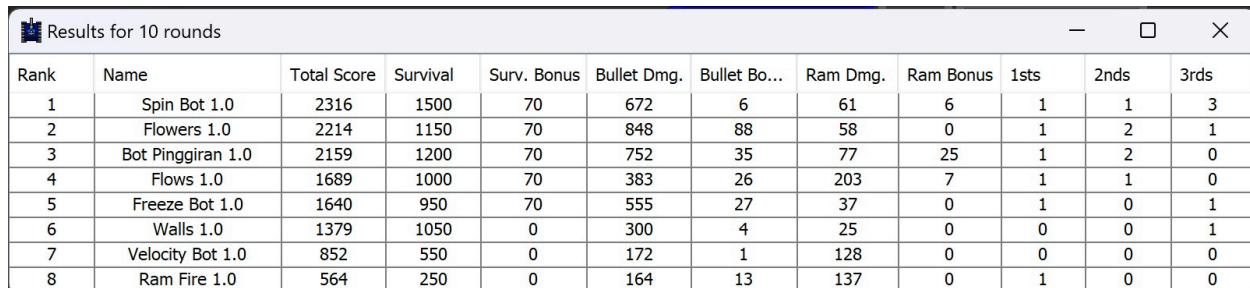
Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bo...	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	Flowers 1.0	3189	1500	210	1312	104	62	0	4	1	0
2	Flows 1.0	2579	1500	140	751	24	122	41	2	3	0
3	Bot Pinggiran 1.0	1983	1250	0	672	25	35	0	0	2	1
4	Freeze Bot 1.0	1896	1200	0	615	46	35	0	0	1	2
5	Spin Bot 1.0	1353	800	0	480	6	66	0	0	0	2
6	Walls 1.0	1329	850	70	350	16	43	0	1	0	0
7	Velocity Bot 1.0	1162	750	0	268	5	138	0	0	0	1
8	Ram Fire 1.0	970	600	0	188	0	125	58	0	0	1

Gambar 8. Hasil Pengujian 2 Atas 4 Bot Terpilih Dengan 4 Sample Bot



Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bo...	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	Flowers 1.0	2678	1400	140	1008	87	42	0	3	0	2
2	Bot Pinggiran 1.0	2000	1400	0	528	19	37	15	0	2	1
3	Freeze Bot 1.0	1913	1200	70	563	36	43	0	1	1	1
4	Spin Bot 1.0	1856	1200	140	464	12	40	0	2	0	1
5	Walls 1.0	1747	1300	0	390	21	36	0	0	1	1
6	Flows 1.0	1516	800	0	449	34	203	30	0	2	0
7	Ram Fire 1.0	864	500	0	228	0	136	0	0	0	0
8	Velocity Bot 1.0	588	300	0	188	0	100	0	0	0	0

Gambar 9. Hasil Pengujian 3 Atas 4 Bot Terpilih Dengan 4 Sample Bot



Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bo...	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	Spin Bot 1.0	2316	1500	70	672	6	61	6	1	1	3
2	Flowers 1.0	2214	1150	70	848	88	58	0	1	2	1
3	Bot Pinggiran 1.0	2159	1200	70	752	35	77	25	1	2	0
4	Flows 1.0	1689	1000	70	383	26	203	7	1	1	0
5	Freeze Bot 1.0	1640	950	70	555	27	37	0	1	0	1
6	Walls 1.0	1379	1050	0	300	4	25	0	0	0	1
7	Velocity Bot 1.0	852	550	0	172	1	128	0	0	0	0
8	Ram Fire 1.0	564	250	0	164	13	137	0	1	0	0

Gambar 10. Hasil Pengujian 4 Atas 4 Bot Terpilih Dengan 4 Sample Bot

4.2.4. Analisis Hasil Pengujian

Berdasarkan hasil pengujian diatas, kita bisa menarik analisis bahwa bot Flowers sebagai bot utama dari 4 bot yang telah dibuat terbukti mampu menunjukkan kapasitas dan kemampuannya dengan selalu konsisten berada di papan atas permainan (top 1 / top 2) baik dalam kondisi 1v1v1v1 yang berjumlah sedang, atau 1v1 yang berbentuk by1 berjumlah sedikit, ataupun 1v1v1v1v1v1v1v1v1 yang berjumlah banyak. Hal ini membuktikan algoritma greedy yang dibuat sudah cukup optimal dengan heuristik yang digunakan untuk memaksimalkan poin yang didapatkan tersebut. Menganalisis 3 bot lain yang telah dibuat juga, dapat dilihat bahwa kecenderungan 3 bot lainnya adalah tetap cukup konsisten juga, hanya saja jika diamati pada saat pertandingan berlangsung, terlihat bahwa bot Freeze mampu optimal saat permainan cenderung dilakukan jarak jauh, bot Flows mampu optimal saat permainan cenderung dilakukan jarak dekat dan jumlahnya sedikit, serta bot Pinggiran mampu optimal saat tidak ada yang mengganggu wall atau perjalanan menuju wall yang dimainkannya. Sedikit menganalisis bot-bot sampel yang menjadi bahan uji di sini, kebanyakan memiliki pola atau play style yang terstruktur dan sederhana yang sangat memungkinkan untuk dikalahkan dengan mudah, kecuali Spin Bot yang mampu menunjukkan kelebihanannya yang disebabkan oleh movement yang tidak lurus. Oleh karena itu, dapat disimpulkan bahwa bot Flowers yang dipilih menjadi bot utama sudah tepat, dan 3 bot lainnya yang telah dibuat juga menunjukkan konsistensi dan permainan yang baik dengan beragam algoritma greedy dengan heuristik masing-masing yang berbeda.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Jadi, kami mengimplementasikan strategi greedy pada gim Robocode TankRoyale. Strategi greedy ini adalah metode sederhana yang cukup populer dalam menyelesaikan masalah optimasi.. Algoritma ini beroperasi dengan memilih pilihan terbaik pada setiap langkahnya tanpa mempertimbangkan implikasi di masa depan, dengan keyakinan bahwa memilih opsi terbaik di setiap langkah akan membawa kita menuju solusi optimal secara keseluruhan.

Bot Utama fokus pada pergerakan dasar dan tembakan maksimum, Bot Alternatif 1 menekankan pada bertahan hidup dan efisiensi peluru dengan pergerakan sinusoidal dan penyesuaian ukuran peluru berdasarkan jarak dan energi. Bot Alternatif 2 menargetkan lawan hingga radar lepas dan menggunakan pergerakan acak berdasarkan giliran, serta penyesuaian ukuran peluru berdasarkan jarak dan energi. Terakhir, Bot Alternatif 3 menggunakan peluru berukuran 3 dan pergerakan konstan dari dinding ke dinding untuk menghindari tembakan, serta reaksi terhadap tabrakan dengan bot, peluru, dan dinding. Setiap bot memiliki keunggulan dan kelemahan tersendiri dalam berbagai situasi.

5.2 Saran

Pengimplementasian strategi greedy pada game Robocode TankRoyale memiliki banyak keterbatasan. Masih banyak strategi yang dapat dilakukan untuk bisa memaksimalkan kemampuan robot tank ini, seperti pembelajaran pergerakan, adaptasi lingkungan, dan lainnya. Namun, pada hal ini strategi greedy juga masih bisa dikembangkan dan di-*improve* dengan berkreasi dan menangani banyak kasus.

LAMPIRAN

1. Pranala Github : https://github.com/RNXFreeze/Tubes1_RamadhanKareem
2. Pranala Video Youtube : <https://youtu.be/KPIt3BNW69Q>
3. Tabel Checklist :

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.	✓	

DAFTAR PUSTAKA

Munir, Rinaldi. (2025). Algoritma Greedy (Bagian 1).
[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/04-Algoritma-Greedy-\(2025\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/04-Algoritma-Greedy-(2025)-Bag1.pdf). Diakses pada 24 Maret 2025.

Munir, Rinaldi. (2025). Algoritma Greedy (Bagian 2).
[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/05-Algoritma-Greedy-\(2025\)-Bag2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/05-Algoritma-Greedy-(2025)-Bag2.pdf). Diakses pada 24 Maret 2025.

Munir, Rinaldi. (2025). Algoritma Greedy (Bagian 3).
[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/06-Algoritma-Greedy-\(2025\)-Bag3.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/06-Algoritma-Greedy-(2025)-Bag3.pdf). Diakses pada 24 Maret 2025.

Starterpack Robocode.
<https://github.com/Ariel-HS/tubes1-if2211-starter-pack/releases/tag/v1.0>. Diakses pada 24 Maret 2025.

Dokumentasi API Robocode. <https://robocode-dev.github.io/tank-royale/>. Diakses pada 24 Maret 2025.