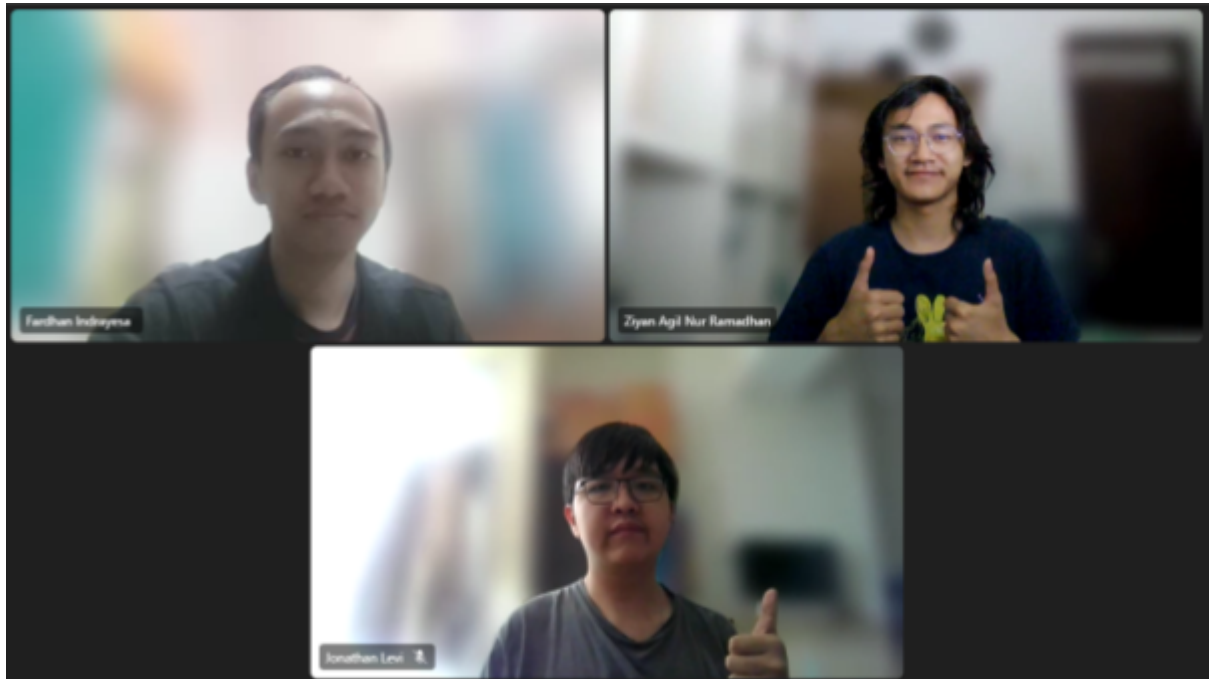


Pemanfaatan Algoritma Greedy Dalam Pembuatan *Bot* Permainan Robocode Tank Royale

Laporan Tugas Besar 1
IF2211 Strategi Algoritma



Disusun Oleh Kelompok 42:

12821046	Fardhan Indrayesa
13523132	Jonathan Levi
13622076	Ziyan Agil Nur Ramadhan

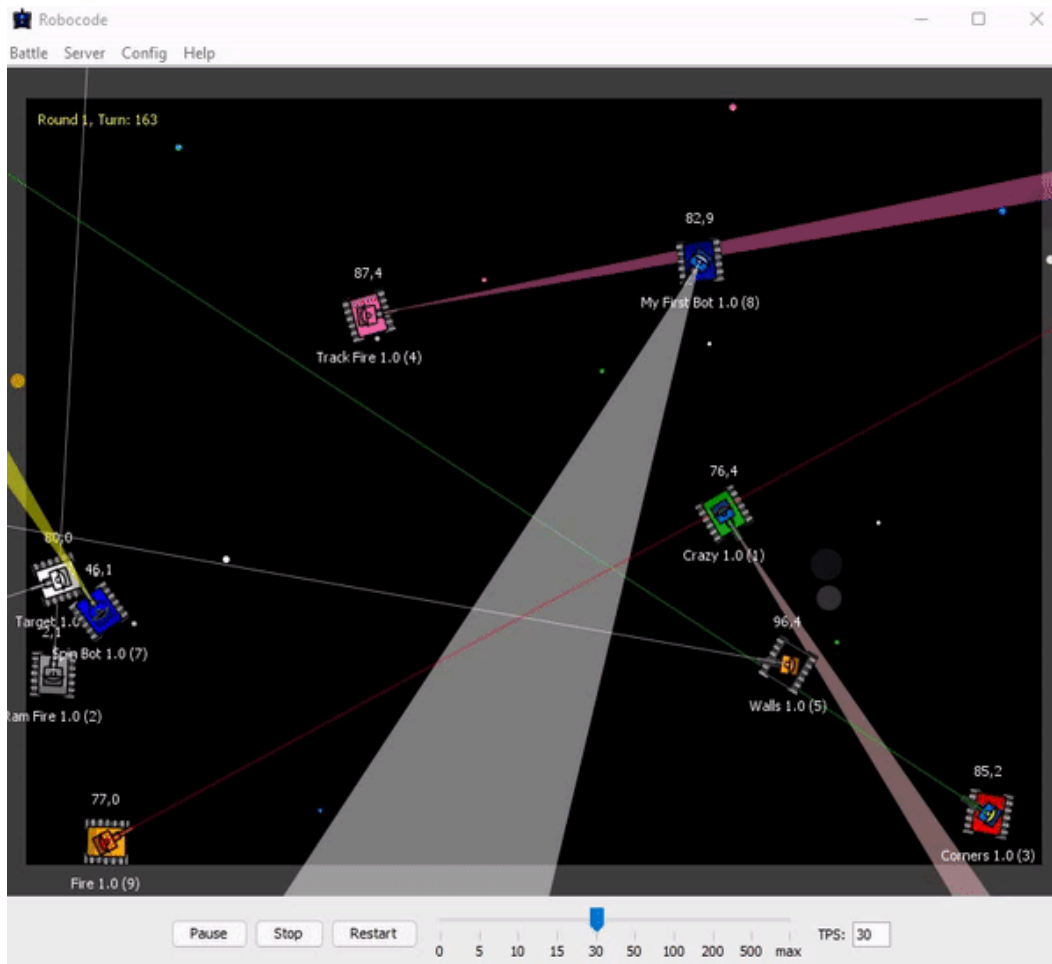
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2025

DAFTAR ISI

DAFTAR ISI.....	1
Bab 1: Deskripsi Tugas.....	2
Bab 2: Landasan Teori.....	8
a. Dasar Teori.....	8
b. Cara Kerja Bot.....	8
Bab 3: Aplikasi Strategi Greedy.....	10
Bab 4: Implementasi dan Pengujian.....	13
Bab 5: Kesimpulan dan Saran.....	35
a. Kesimpulan.....	35
b. Saran.....	35
PEMBAGIAN TUGAS.....	36
LAMPIRAN.....	37
DAFTAR PUSTAKA.....	38

Bab 1

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 melewati turn tersebut. Jika bot melewati 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.

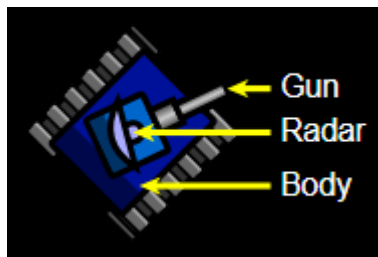
6. Tabrakan

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.

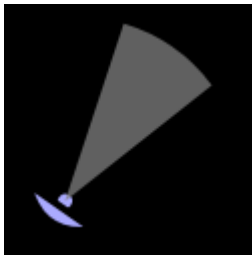
Bagian paling lambat adalah tubuh tank, yang dalam kondisi terbaik dapat berputar hingga 10 derajat per giliran. Namun, ini bergantung pada kecepatan bot saat ini. Semakin cepat bot bergerak, semakin lambat kemampuannya untuk berbelok.

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

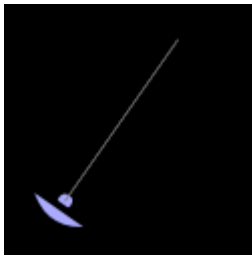
10. Pemindaian

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](#).

Bab 2

Landasan Teori

a. Dasar Teori

Algoritma *greedy* adalah algoritma yang memecahkan suatu masalah langkah demi langkah tanpa memikirkan konsekuensi jangka panjang. Cara kerja algoritma ini adalah sebagai berikut:

1. Ambil solusi yang terbaik dari langkah **saat ini** (tanpa mempertimbangkan konsekuensi langkah berikutnya).
2. Lanjut ke langkah berikutnya.
3. Ulangi tahap 1 dan 2 sampai langkah selesai.

Harapannya, dengan pemilihan solusi paling optimal pada setiap langkah (tanpa mempertimbangkan konsekuensi langkah berikutnya), algoritma ini menghasilkan solusi yang paling optimal secara keseluruhan.

b. Cara Kerja Bot

- Voratank

Jika bot mendapat giliran pertama, bot akan bergerak maju; bot juga memutar radarnya ke searah jarum jam mula-mula dan berbelok maju.

Saat bot mendeteksi bot musuh, tembak berdasarkan kecepatan bot musuh, lalu ubah arah perputaran radar. Jika bot dan bot musuh saling menabrak, tembak bot musuh, kemudian pindah posisi. Jika menabrak dinding, bot akan berbalik arah dan maju. Jika bot musuh berhasil menembak Voratank, bot ini akan menghadap ke arah peluru dan membalas bot musuh, lalu berbelok. Radar bot akan berputar saat mengetahui pelurunya menembak dinding. Saat bot musuh berhasil tertembak, Voratank akan menembak kembali ke arah peluru musuh. Bot ini akan melakukan scan ulang saat ada bot musuh yang mati. Setelah menembak bot musuh, bot ini akan langsung berpindah tempat.

- VoraBot

VoraBot bekerja dengan mendeteksi musuh menggunakan radar. Radar akan berputar untuk mencari keberadaan musuh. Jika tidak ada musuh yang terdeteksi, bot bergerak maju dan berbelok. Jika terdapat bot musuh yang terdeteksi, VoraBot akan mengarahkan meriam kepada target yang terdeteksi. Untuk mendapatkan akurasi yang tinggi, bot akan berhenti sejenak dan menembakkan meriam berdasarkan jarak VoraBot dengan musuh (semakin dekat, semakin besar kekuatannya). Saat bertabrakan dengan bot musuh, VoraBot menggunakan strategi untuk mengarahkan meriam ke bot musuh, menembaknya dengan kekuatan maksimal, lalu menjauhi bot tersebut.

VoraBot juga memiliki mekanisme untuk segera menjauhi dinding jika bot telah menabrak dinding. Terakhir, VoraBot memiliki mekanisme untuk perlawanan jika terkena serangan musuh. VoraBot akan mendeteksi asal serangan musuh untuk memberi serangan balasan, lalu melakukan pergerakan secara zigzag untuk menghindari serangan lanjutan dari bot musuh.

- UnpredictaBot

Bot akan mengubah arah perputaran radar secara reguler agar dapat melacak bot musuh lebih cepat. Bot juga akan menghindari dari dinding, dan jika menabrak dinding, UnpredictaBot akan berbelok dengan sudut berdasarkan jumlah energi.

Saat bot mendeteksi bot musuh, tembak berdasarkan kecepatan bot musuh, kemudian bot melakukan scan ulang saat bot musuh tepat di hadapan senapan agar dapat mengunci sasaran. Bot ini akan menembak dengan kekuatan lebih rendah jika energinya rendah, untuk menghemat energi. Jika bot menabrak bot musuh, tembak bot musuh (tembak dengan kekuatan lebih rendah jika energi rendah), lalu pindah posisi. Jika tertembak bot musuh, bot ini akan menghadap ke arah peluru musuh dan menembaknya, lalu menghindar. Radar bot akan berputar saat mengetahui pelurunya menembak dinding. Saat bot musuh berhasil tertembak, bot ini akan menembak kembali berdasarkan kekuatan tembakan musuh. Bot ini akan melakukan scan ulang saat ada bot musuh yang mati. Setelah menembak bot musuh, UnpredictaBot ini akan langsung berpindah tempat berdasarkan jumlah energi yang dimilikinya.

- SmartVoraRam

Bot pada awalnya berputar dengan sudut tertentu untuk memeriksa bot lawan. Apabila terdapat bot lawan yang terdeteksi, akan diperiksa kecepatan dan energy bot lawan tersebut. Apabila memenuhi suatu kondisi, bot ini akan mengarah ke bot lawan tersebut lalu menabraknya. Ketika menabrak bot lawan, bot ini akan terus menembak dan maju ke arah bot lawan sampai bot lawan kabur atau hancur. Apabila bot lawan kabur atau hancur, bot ini akan berhenti menembak. Apabila saat mendeteksi bot lawan tidak ada kondisi yang memenuhi, bot ini akan menembak bot lawan yang terdeteksi, lalu berpindah tempat setelah menembak. Strategi greedy bot ini berada pada cara menabrak bot lawan yang memenuhi kondisi tertentu. Bot ini akan melemahkan bot lawan dengan cara menembaknya dari jarak jauh terlebih dahulu. Lalu, apabila memenuhi suatu kondisi, bot ini akan menabrak bot lawan sampai bot lawan tersebut pergi atau hancur.

Bab 3

Aplikasi Strategi *Greedy*

- Voratank

Voratank merupakan bot yang menyerang bot lawan berdasarkan kecepatan bot lawan yang terdeteksi. Bot ini mengutamakan serangan tembakan langsung ke lawan saat bot lawan terdeteksi.

Elemen algoritma *greedy* bot Voratank.

Himpunan kandidat	Seluruh bot lawan
Himpunan solusi	Satu target bot lawan
Fungsi solusi	Mengunci bot lawan yang sudah terdeteksi oleh radar bot.
Fungsi seleksi	Menentukan apakah bot lawan memiliki kecepatan yang cukup kecil untuk ditembak dengan <i>firepower</i> yang besar.
Fungsi kelayakan	Memeriksa bot lawan berdasarkan jangkauan radar dan ambil nilai kecepatannya, apakah layak untuk ditembak secara <i>greedy</i> (dengan <i>firepower</i> tinggi) atau tidak.
Fungsi objektif	Memperoleh skor setinggi mungkin dengan cara terus menembak bot lawan yang terdeteksi.

- VoraBot

VoraBot menyerang siapapun yang dapat terdeteksi olehnya dan lebih mengutamakan efisiensi dalam menyerang bot musuh (serangan berdasarkan jarak dengan bot musuh).

Elemen algoritma *greedy* bot VoraBot.

Himpunan kandidat	Seluruh bot lawan yang berada di dalam arena.
Himpunan solusi	Bot musuh yang terdeteksi oleh radar.
Fungsi solusi	Memeriksa bot musuh yang telah terdeteksi oleh radar.
Fungsi seleksi	Menyerang apapun yang terdeteksi pertama kali oleh bot.
Fungsi kelayakan	Memeriksa apakah bot berada dalam jangkauan radar atau tidak. Jika berada dalam jangkauan radar, memberikan damage berdasarkan jarak dengan bot musuh. Jika tidak, melakukan <i>scanning</i> kembali.
Fungsi objektif	Memperoleh skor tinggi dengan menembakkan meriam

	kepada bot musuh.
--	-------------------

- UnpredictaBot

UnpredictaBot adalah bot yang memiliki pergerakan/strategi yang bersifat *pseudo-random*, karena pergerakan dalam beberapa kasus adalah berdasarkan jumlah energi.

Elemen algoritma *greedy* bot UnpredictaBot:

Himpunan kandidat	Seluruh bot musuh yang berpotensi ditarget.
Himpunan solusi	Bot musuh yang terpilih.
Fungsi solusi	Memeriksa bot musuh yang dapat ditembak dengan cara membalikkan arah putaran radar secara reguler/rutin.
Fungsi seleksi	Mengunci target dan menentukan kekuatan tembakan berdasarkan kecepatan bot musuh sekarang dengan tiga tingkat kekuatan.
Fungsi kelayakan	Memeriksa apakah kekuatan tembakan besar layak dipakai atau tidak, berdasarkan energi UnpredictaBot sekarang.
Fungsi objektif	Mendapat skor setinggi-tingginya dengan cara menembak bot musuh.

- SmartVoraRam

Bot SmartVoraRam merupakan bot yang mengalahkan bot lawan dengan cara ditabrak. Bot ini mendeteksi kecepatan dan energi bot lawan. Apabila kecepatan dan energi bot cukup rendah, bot ini akan menabrak bot lawan tersebut. Tapi, apabila energi bot lawan masih cukup banyak, bot ini menembak bot lawan terlebih dahulu (melemahkan bot lawan, agar saat ditabrak, langsung hancur dan mendapatkan bonus ram damage).

Elemen algoritma *greedy* bot SmartVoraRam:

Himpunan kandidat	Seluruh bot lawan
Himpunan solusi	Satu target bot lawan
Fungsi solusi	Memeriksa semua bot yang mungkin untuk ditabrak.
Fungsi seleksi	Menentukan apakah bot lawan memiliki kecepatan dan energi minimum berdasarkan energi yang sudah ditentukan (4,5 dan 40).

Fungsi kelayakan	Memeriksa apakah bot lawan layak untuk ditabrak atau tidak berdasarkan fungsi seleksi.
Fungsi objektif	Memperoleh skor setinggi mungkin dengan cara menabrak bot lawan.

Dari hasil pengujian sebanyak 7 kali dengan mempertandingkan 1 bot utama dan 3 bot alternatif secara bersamaan, didapatkan bahwa Voratank mendominasi pertandingan dengan menang sebanyak 3 kali, diikuti oleh VoraBot dan UnpredictaBot masing-masing sebanyak 2 kali. Untuk juara kedua, UnpredictaBot dan Smart Vora Ram masing-masing terletak pada juara tersebut sebanyak 3 kali, Voratank 1 kali. Pada juara ketiga, VoraBot dan Voratank masing-masing mendapat juara tersebut 3 kali, Smart Vora Ram 1 kali. Pada juara keempat, Smart Vora Ram mendapat juara tersebut sebanyak 3 kali, UnpredictaBot dan VoraBot masing-masing 2 kali.

Dapat disimpulkan bahwa solusi *greedy* yang paling efisien dan efektif dari antara keempat bot adalah Spam Fire Damage dari Voratank karena cenderung menang dan tidak pernah terletak pada juara keempat.

Berdasarkan empat hasil eksplorasi strategi *greedy* pada keempat alternatif bot di atas, berikut merupakan alasan dan pertimbangan pemilihan strategi tersebut.

Strategi	<i>Heuristic</i>	Alasan
Spam Fire Damage (Voratank)	Menembak bot lawan yang terdeteksi oleh radar secara terus menerus dengan damage yang besar.	Agar bot lawan cepat kehabisan energi ketika ditembak oleh bot ini.
Pergerakan zig zag (Vorabot)	Belok kiri dan belok kanan dengan sudut yang sama.	Agar sulit ditembak oleh bot musuh.
Kunci target (UnpredictaBot)	Fungsi Rescan() saat <i>bearing</i> = 0.	Agar bot dapat fokus terhadap satu bot musuh dan mengurangi serangan yang meleset.
Spam Ram Fire (SmartVoraRam)	Bot berjalan ke bot musuh dan terus menabraknya sambil menembak.	Agar dapat menghabiskan energi bot musuh jauh lebih cepat.

Bab 4

Implementasi dan Pengujian

a. Implementasi

Pseudocode dan penjelasan setiap fungsi dan prosedur masing-masing bot adalah sebagai berikut.

- Voratank

Berikut merupakan pseudocode dari bot Voratank.

```
START Voratank

DECLARE turnDirect AS INTEGER ← 1

FUNCTION Main()
    CALL Start()

FUNCTION Run()
    SET BodyColor ← GRAY
    SET TurretColor ← BLUE
    SET RadarColor ← YELLOW

    ENABLE AdjustGunForBodyTurn
    ENABLE AdjustRadarForGunTurn
    ENABLE AdjustRadarForBodyTurn

    WHILE bot is running DO
        IF TurnNumber = 1 THEN
            Forward(100)
        ENDIF

        TurnRadarRight(270 * turnDirect)
        TurnLeft(45)
        Forward(80)
    ENDWHILE

FUNCTION OnScannedBot(event e)
    CALL ToTarget(e.X, e.Y)

    Stop()
    IF |e.Speed| < 5 THEN
        Fire(2)
        TurnRadarLeft(30 * turnDirect)
    ELSE
        Fire(1)
        TurnRadarLeft(30 * turnDirect)
    ENDIF

    turnDirect ← turnDirect * -1
```

```

    Resume()

FUNCTION OnHitBot(event e)
    CALL ToTarget(e.X, e.Y)

    Stop()
    Fire(2.5)

    TurnLeft(NormalizeRelativeAngle(90 - DirectionTo(e.X, e.Y)))
    SET MaxSpeed ← 5
    Forward(90)

    Resume()

FUNCTION OnHitWall(event e)
    TurnLeft(80)
    Forward(60)

FUNCTION OnHitByBullet(event e)
    CALL ToTarget(e.Bullet.X, e.Bullet.Y)
    Fire(1)
    TurnRight(NormalizeRelativeAngle(90 - (Direction -
        e.Bullet.Direction)))
    Forward(60)

FUNCTION OnBulletHitWall(event e)
    TurnRadarLeft(45)

FUNCTION OnBulletHit(event e)
    CALL ToTarget(e.Bullet.C, e.Bullet.Y)
    Fire(2)

FUNCTION OnBotDeath(event e)
    Rescan()

FUNCTION OnBulletFired(event e)
    TurnLeft(30)
    Forward(70)

FUNCTION ToTarget(x, y)
    DECLARE direct AS VAR ← DirectionTo(x, y)
    DECLARE gunBear AS VAR ← NormalizeRelativeAngle(direct -
        GunDirection)

    TurnGunLeft(gunBear)

END Voratank

```

Berdasarkan pseudocode di atas, berikut merupakan penjelasan dari masing-masing fungsi dan prosedur yang digunakan

Class	
<code>public class Voratank</code>	Kelas inheritance dari Robocode Bot. Kelas ini merupakan kelas utama bot yang berisi seluruh data, fungsi, dan prosedur yang akan digunakan untuk menjalankan bot.
Attribute	
<code>turnDirect</code>	Arah putar radar, searah (negatif) atau berlawanan arah (positif) jarum jam.
Fungsi dan Prosedur	
<code>FUNCTION Main() CALL Start()</code>	Memulai bot dengan memanggil Start()
<code>FUNCTION Run()</code>	Menjalankan logika utama bot seperti mengatur warna body, gun, dan radar bot, serta mengatur pergerakan <i>default</i> bot.
<code>FUNCTION OnScannedBot(event e)</code>	Fungsi ketika radar mendeteksi adanya bot musuh. Bekerja dengan cara menyimpan posisi terakhir musuh yang terdeteksi, kemudian berhenti untuk mengarahkan senjata ke target dan menembaknya dengan kekuatan yang disesuaikan berdasarkan kecepatan bot target.
<code>FUNCTION OnHitBot(event e)</code>	Fungsi ketika bot bertabrakan dengan bot lain. Bekerja dengan cara mengarahkan senjata ke bot musuh yang menabrak atau ditabrak, lalu menembaknya dengan firepower tertentu, kemudian pindah posisi setelah menembak bot musuh tersebut.
<code>FUNCTION OnHitWall(event e)</code>	Fungsi ketika bot bertabrakan dengan dinding. Bekerja dengan cara berbelok kemudian maju.
<code>FUNCTION OnHitByBullet(event e)</code>	Fungsi ketika bot ditembak oleh musuh. Bekerja dengan cara mencari sumber tembakan, mengarahkan peluru ke sumber tembakan, lalu menembaknya dengan

	<i>firepower</i> tertentu, kemudian berpindah posisi setelah menembak.
FUNCTION OnBulletHitWall(event e)	Fungsi ketika peluru mengenai dinding. Bekerja dengan cara memutar radar dengan sudut tertentu.
FUNCTION OnBulletHit(event e)	Fungsi ketika peluru berhasil mengenai bot musuh. Bekerja dengan cara mengarahkan kembali senjata ke musuh yang tertembak dan menembak lagi dengan <i>firepower</i> tertentu.
FUNCTION OnBotDeath(event e)	Fungsi ketika terdapat bot yang sudah hancur. Bekerja dengan scan ulang seluruh arena permainan.
FUNCTION OnBulletFired(event e)	Fungsi ketika bot menembak peluru. Bekerja dengan berpindah posisi, setelah bot menembakkan peluru.
FUNCTION ToTarget(x, y)	Fungsi untuk mengarahkan senjata bot ini kepada bot lawan yang terdeteksi.

- VoraBot

Berikut merupakan pseudocode dari bot Vorabot.

```

START VoraBot

DECLARE lastTargetX, lastTargetY AS DOUBLE {Posisi terakhir musuh yang terdeteksi}
DECLARE movementDirection AS INTEGER ← 1 {Arah pergerakan bot (1 = maju, -1 = mundur)}
DECLARE radarDirection AS INTEGER ← 1 {Arah perputaran radar (1 = searah jarum jam, -1 = berlawanan)}

FUNCTION Main()
    CALL Start()

FUNCTION Run()
    {Warna elemen-elemen pada bot}
    SET BodyColor ← RED
    SET TurretColor ← YELLOW

```

```

SET RadarColor ← GREEN
SET BulletColor ← WHITE
SET ScanColor ← CYAN
  {Mengatur agar radar dan senjata bergerak secara independen
dari body bot}
ENABLE AdjustGunForBodyTurn
ENABLE AdjustRadarForGunTurn
ENABLE AdjustRadarForBodyTurn

WHILE bot is running DO
  {Radar terus berputar untuk mencari musuh}
  TurnRadarRight(360 * radarDirection)

  {Bot bergerak dalam pola memutar untuk menghindari musuh}
  TurnLeft(45)
  Forward(60)
ENDWHILE

FUNCTION OnScannedBot(event e)
  {Simpan posisi terakhir musuh}
  lastTargetX ← e.X
  lastTargetY ← e.Y

  {Arahkan senjata ke target}
  CALL ToTarget(lastTargetX, lastTargetY)

  {Hentikan pergerakan sementara untuk menembak lebih akurat}
  StopMovement()

  {Hitung kekuatan tembakan berdasarkan jarak musuh}
  firePower ← MIN(3, 500 / DistanceTo(lastTargetX, lastTargetY))
  Fire(firePower)

  {Ubah arah radar agar tetap mengikuti musuh}
  radarDirection ← -radarDirection

  {Lanjutkan pergerakan bot}
  ResumeMovement()

FUNCTION OnHitBot(event e)

```

```

    {Arahkan senjata ke bot lawan}
    CALL ToTarget(e.X, e.Y)

    {Berhenti sejenak untuk menembak}
    StopMovement()
    Fire(3)
    Back(50)
    ResumeMovement()

FUNCTION OnHitWall(event e)
    Back(60)
    TurnRight(90)
    Forward(100)

FUNCTION OnHitByBullet(event e)
    {Arahkan senjata ke posisi musuh yang menembak}
    CALL ToTarget(e.Bullet.X, e.Bullet.Y)

    distance ← DistanceTo(e.Bullet.X, e.Bullet.Y)
    firePower ← MIN(3, MAX(0.1, 500 / distance))
    Fire(firePower)
    {Gunakan gerakan zig-zag untuk menghindari serangan lanjutan}
    CALL ZigZagMovement()

FUNCTION OnBulletHitWall(event e)
    TurnRadarLeft(90)
    Rescan()

FUNCTION OnBulletHit(event e)
    CALL ToTarget(e.Bullet.X, e.Bullet.Y)

    {Hitung firepower berdasarkan jarak ke musuh}
    distance ← DistanceTo(e.Bullet.X, e.Bullet.Y)
    firePower ← MIN(3, MAX(0.1, 500 / distance))
    Fire(firePower)

FUNCTION OnBulletFired(event e)
    TurnLeft(20)
    Forward(50)

```

```

FUNCTION OnBotDeath(event e)
    Rescan()

FUNCTION ToTarget(x, y)
    DECLARE direction AS VAR ← DirectionTo(x, y)
    DECLARE gunBearing AS VAR ← NormalizeBearing(direction -
                                                    GunDirection)

    TurnGunLeft(gunBearing)

FUNCTION NormalizeBearing(angle)
    WHILE angle > 180 DO
        angle ← angle - 360
    ENDWHILE
    WHILE angle < -180 DO
        angle ← angle + 360
    ENDWHILE
    RETURN angle
{Metode untuk melakukan gerakan zig-zag agar sulit ditembak}
FUNCTION ZigZagMovement()
    FOR i ← 1 to 3 DO
        TurnRight(40 * movementDirection)
        Forward(40)
        TurnLeft(40 * movementDirection)
        Forward(40)
    ENDFOR
    movementDirection ← -movementDirection

END VoraBot

```

Berdasarkan pseudocode di atas, berikut merupakan penjelasan dari masing-masing fungsi dan prosedur yang digunakan

Class	
Public class Vorabot	Berisi seluruh atribut, fungsi, dan prosedur yang menjalankan VoraBot.
Attribute	
lastTargetX, lastTargetY	Menyimpan posisi dari bot musuh yang

	terdeteksi dalam koordinat kartesius.
<code>movementDirection</code>	Menentukan arah pergerakan bot (1 = maju, -1 = mundur)
<code>radarDirection</code>	Menentukan arah rotasi radar (1 = searah jarum jam, -1 = berlawanan arah jarum jam)
Fungsi dan Prosedur	
<code>FUNCTION Main() CALL Start()</code>	Memulai bot dengan memanggil <code>Start()</code>
<code>FUNCTION Run()</code>	Menjalankan logika utama bot seperti mengatur warna pada elemen-elemen bot, mengatur pergerakan radar, <i>body</i> , dan <i>gun</i> pada bot, serta mengatur <i>scanning</i> dan pergerakan bot.
<code>FUNCTION OnScannedBot(event e)</code>	Fungsi ketika radar mendeteksi adanya bot musuh. Bekerja dengan cara menyimpan posisi terakhir musuh yang terdeteksi, kemudian berhenti untuk mengarahkan senjata ke target dan menembaknya dengan kekuatan yang disesuaikan berdasarkan jarak ke target.
<code>FUNCTION OnHitBot(event e)</code>	Fungsi ketika bot bertabrakan dengan bot lain. Bekerja dengan cara mengarahkan senjata ke bot musuh yang menabrak atau ditabrak, menembak dengan kekuatan penuh, kemudian mundur.
<code>FUNCTION OnHitWall(event e)</code>	Fungsi untuk mengatasi ketika bot menabrak dinding. Bekerja dengan cara mundur, berbelok, kemudian maju.
<code>FUNCTION OnHitByBullet(event e)</code>	Fungsi untuk mengatasi ketika bot ditembak oleh bot musuh. Bekerja dengan cara mencari asal sumber tembakan, mengarahkan peluru ke sumber tembakan, menembaknya dengan kekuatan yang disesuaikan dengan jarak, kemudian bergerak dengan zigzag untuk menghindari serangan lanjutan.
<code>FUNCTION OnBulletHitWall(event e)</code>	Fungsi ketika peluru mengenai dinding. Bekerja dengan cara memutar radar kemudian

	melakukan <i>scan</i> ulang.
FUNCTION OnBulletedHit(event e)	Fungsi ketika peluru mengenai bot musuh. Bekerja dengan cara mengarahkan kembali senjata ke musuh yang tertembak dan menembaknya dengan kekuatan yang disesuaikan berdasarkan jarak.
FUNCTION OnBulletFired(event e)	Fungsi yang dijalankan setelah bot menembakkan peluru. Bekerja dengan cara berbelok dan maju untuk menghindari jika bot musuh memiliki mekanisme serupa untuk serangan balasan.
FUNCTION OnBotDeath(event e)	Fungsi ketika terdapat bot yang mati. Bekerja dengan melakukan scan ulang untuk mendapatkan target baru.
FUNCTION ToTarget(x, y)	Fungsi untuk menghitung arah dan mengarahkan senjata ke bot musuh. Bekerja dengan cara menghitung sudut yang diperlukan agar senjata dapat diarahkan ke target.
FUNCTION NormalizeBearing(angle)	Fungsi untuk normalisasi sudut antara -180 hingga 180 derajat.
FUNCTION ZigZagMovement()	Fungsi untuk pergerakan bot secara zigzag. Bekerja dengan cara melakukan pergerakan secara zigzag yaitu belok kanan, maju, belok kiri, maju.

- UnpredictaBot

Berikut merupakan pseudocode dari bot UnpredictaBot.

```

START UnpredictaBot

DECLARE turnDirection AS INTEGER ← 1
DECLARE moveDirection AS INTEGER ← 1
DECLARE oscillationCounter AS INTEGER ← 0
DECLARE justFired AS BOOLEAN ← false

DECLARE WALL_MARGIN AS DOUBLE ← 50

```

```

DECLARE angleToCenter, turnAngle AS DOUBLE { avoidWall() }
DECLARE firePower, gunBear AS DOUBLE { OnScannedBot() }
DECLARE perpendicular AS DOUBLE { OnHitBot() }
DECLARE dodgeAngle AS DOUBLE { OnHitByBullet() }

FUNCTION Main()
    CALL Start()

FUNCTION Run()
    SET BodyColor ← GRAY
    SET TurretColor ← YELLOW
    SET RadarColor ← GREEN
    SET BulletColor ← KHAKE

    ENABLE AdjustGunForBodyTurn
    ENABLE AdjustRadarForGunTurn
    ENABLE AdjustRadarForBodyTurn

    WHILE bot is running DO
        CALL avoidWall()

        TurnRadarLeft(90 * turnDirection)
        oscillationCounter ← oscillationCounter + 1
        IF oscillationCounter == 4 THEN
            oscillationCounter ← 0
            turnDirection ← turnDirection * -1
        ENDIF
    ENDWHILE

FUNCTION avoidWall()
    IF ((X < WALL_MARGIN) OR (X > ArenaWidth - WALL_MARGIN) OR
        (Y < WALL_MARGIN) OR (Y > ArenaHeight - WALL_MARGIN)) THEN
        angleToCenter ← DirectionTo(ArenaWidth / 2, ArenaHeight / 2)
        turnAngle ← NormalizeRelativeAngle(angleToCenter - Direction)

        TurnRight(turnAngle)
        Forward(100)
        moveDirection ← moveDirection * -1
    
```

```

ELSE
    if (NOT justFired)
        TurnLeft(45)
        Forward(80 * moveDirection)
    ENDIF

    justFired ← false

FUNCTION OnHitWall(event e)
    turnAngle ← 90 + (Energy MOD 70)
    TurnLeft(turnAngle)
    Forward(100)
    moveDirection ← moveDirection * -1

{ Strategi greedy: bot lawan terdeteksi -> arahkan ke bot tersebut ->
ukur kecepatannya -> kunci target dan tembak }
FUNCTION OnScannedBot(event e)
    gunBear ← GunBearingTo(e.X, e.Y)
    TurnGunLeft(gunBear)

    IF |e.Speed| < 2 THEN
        firePower ← 4
    ELSE
        IF |e.Speed| < 5 THEN
            firePower ← 2
        ELSE
            firePower ← 1
        ENDIF
    ENDIF

    IF Energy < 20 THEN
        firePower ← 1
    ENDIF

    Fire(firePower)

    justFired ← true

    IF gunBear = 0 THEN
        Rescan()

```



```

ENDIF

FUNCTION OnHitBot(event e)
    gunBear ← GunBearingTo(e.X, e.Y)
    TurnGunLeft(gunBear)

    Stop(true)
    IF Energy < 2.5 THEN
        Fire(0.2)
    ELSE
        Fire(2.5)
    ENDIF

    perpendicular ← NormalizeRelativeAngle(90 - DirectionTo(e.X,
                                                                e.Y))

    TurnLeft(perpendicular)
    SET MaxSpeed ← 5
    Forward(100)

    Resume()
    justFired ← true

FUNCTION OnHitByBullet(event e)
    gunBear ← GunBearingTo(e.Bullet.X, e.Bullet.Y)
    TurnGunLeft(gunBear)
    Fire(1.5)

    dodgeAngle ← NormalizeRelativeAngle(90 - (Direction -
                                                e.Bullet.Direction))

    TurnRight(dodgeAngle)
    SET MaxSpeed ← 5
    Forward(100)

    justFired ← true

FUNCTION OnBulletHitWall(event e)
    TurnRadarRight(60)

FUNCTION OnBulletHit(event e)
    gunBear ← GunBearingTo(e.Bullet.X, e.Bullet.Y)

```

```

    TurnGunLeft(gunBear)
    Fire(MIN(e.Bullet.Power + 1, 3))

    justFired ← true

FUNCTION OnBulletFired(event e)
    TurnLeft(35 + INT(Energy MOD 30))
    Forward(70)

    justFired ← true

FUNCTION OnBotDeath(event e)
    Rescan()

END UnpredictaBot

```

Berdasarkan pseudocode di atas, berikut merupakan penjelasan dari masing-masing fungsi dan prosedur yang digunakan

Class	
public class UnpredictaBot	Kelas inheritance yang merupakan kelas utama bot yang berisi seluruh data, fungsi, dan prosedur yang akan digunakan untuk menjalankan bot.
Attribute	
turnDirection	Menentukan arah rotasi radar (1 = ke kiri atau berlawanan arah jarum jam; -1 = ke kanan atau searah jarum jam).
moveDirection	Menentukan arah gerakan bot (1 = maju; -1 = mundur).
oscillationCounter	Menentukan kapan arah rotasi radar berubah (setiap 4 satuan waktu).
justFired	Menentukan apakah bot baru saja menembak (true = barusan menembak; false = tidak barusan menembak).
angleToCenter	Sudut untuk dapat berjalan lurus ke bagian

	tengah arena.
turnAngle	Sudut/ketajaman berbelok yang diperlukan agar bot menghadap ke tengah arena (sehingga dapat berjalan lurus ke bagian tengah arena).
firePower	Kekuatan tembakan.
gunBear	Sudut perputaran senjata ke arah musuh.
perpendicular	Sudut berbelok yang diperlukan agar bot dapat bergerak tegak lurus terhadap bot musuh (sehingga menghindari dari bot yang tertabrak).
dodgeAngle	Sudut berbelok yang diperlukan agar bot dapat bergerak tegak lurus terhadap peluru dari bot musuh (sehingga menghindari dari tembakan bot musuh).
Fungsi dan Prosedur	
FUNCTION Main() CALL Start()	Memulai bot dengan memanggil Start().
FUNCTION Run()	Menjalankan logika utama bot seperti mengatur warna pada elemen-elemen bot, mengatur pergerakan radar, <i>body</i> , dan <i>gun</i> pada bot, serta mengatur <i>scanning</i> dan pergerakan bot.
FUNCTION avoidWall()	Fungsi yang mendeteksi dinding di sekitar bot. Jika bot berada dekat dinding, bot akan berputar ke tengah arena dan maju, lalu mengubah arah gerakan; jika tidak (dan tidak barusan menembak), belok kiri dan maju atau mundur.
FUNCTION OnHitWall(event e)	Fungsi yang mendeteksi jika bot tetap menabrak dinding, yaitu berbelok berdasarkan jumlah energi bot sekarang dan maju, lalu mengubah arah gerakan.
FUNCTION OnScannedBot(event e)	Fungsi yang mendeteksi bot musuh, yaitu mengunci target dan menembak target (bot

	musuh) berdasarkan kecepatan bot musuh dan jumlah energi bot sekarang. Prioritas melihat apakah jumlah energi UnpredictaBot rendah. Jika ya, tembak dengan kekuatan tembakan rendah; jika tidak, tembak berdasarkan kecepatan bot musuh.
FUNCTION OnHitBot(event e)	Fungsi yang mendeteksi jika bot dan bot musuh tabrakan, yaitu mengarahkan senjata ke bot musuh yang tertabrak, menembak dengan kekuatan tembakan berdasarkan jumlah energi bot (UnpredictaBot) sekarang, lalu menghindar secara tegak lurus.
FUNCTION OnHitByBullet(event e)	Fungsi yang mendeteksi jika bot ditembak bot musuh, yaitu menembak ke sumber tembakan, lalu menghindar secara tegak lurus.
FUNCTION OnBulletHitWall(event e)	Fungsi yang mendeteksi jika bot menembak dinding, yaitu memutar radar agar tidak terus menembak dinding.
FUNCTION OnBulletHit(event e)	Fungsi yang mendeteksi jika bot menembak bot musuh, yaitu mengarahkan kembali senjata ke bot musuh yang tertembak dan menembak dengan kekuatan tembakan berdasarkan kekuatan tembakan bot musuh.
FUNCTION OnBulletFired(event e)	Fungsi yang dijalankan setelah bot menembakkan peluru, yaitu berbelok berdasarkan jumlah energi bot sekarang dan maju.
FUNCTION OnBotDeath(event e)	Fungsi yang mendeteksi jika ada bot yang mati, yaitu melakukan <i>scan</i> ulang untuk mendapatkan target baru.

- SmartVoraRam

Berikut merupakan pseudocode dari bot SmartVoraRam.

```
START SmartVoraRam

DECLARE turnDirect AS INTEGER ← 1
DECLARE hittet AS BOOLEAN ← false
```

```

DECLARE botId AS INTEGER ← 0

FUNCTION Main()
    CALL Start()

FUNCTION Run()
    SET BodyColor ← BLUE
    SET TurretColor ← RED
    SET RadarColor ← YELLOW

    WHILE bot is running DO
        TurnRight(90 * turnDirect)
    ENDWHILE

FUNCTION OnScannedBot(event e)
    DECLARE dir AS VAR ← DirectionTo(e.X, e.Y)
    dir ← NormalizeRelativeAngle(dir - Direction)
    SET TurnRate ← 10
    TurnLeft(dir)

    IF (e.Speed < 4.5) AND (e.Energy < 40) THEN
        CALL ToTarget(e.X, e.Y)
        CALL ToBot(e.X, e.Y, 50)
    ELSE
        CALL ToTarget()
        IF Energy > 10 THEN
            Fire(1.5)
        ELSE
            Fire(1)
        ENDIF

        TurnLeft(45 * turnDirect)
        SET TargetSpeed ← 8
        Forward(50)
    ENDIF

FUNCTION OnHitBot(event e)
    hitted ← true
    botId ← e.VictimId

    CALL ToTarget(e.X, e.Y)
    WHILE hitted
        IF Energy > 15 THEN
            Fire(4)
        ELSE
            Fire(1.5)
        ENDIF
        ToBot(e.X, e.Y, 100)

        IF NOT e.IsRammed THEN
            hitted ← false
            BREAK

```

```

        ENDIF
    ENDWHILE

    FUNCTION OnHitWall(event e)
        Rescan()

    FUNCTION OnBulletHit(event e)
        CALL ToTarget(e.Bullet.X, e.Bullet.Y)
        turnDirect ← turnDirect * -1

    FUNCTION OnHitByBullet(event e)
        CALL ToTarget(e.Bullet.X, e.Bullet.Y)

    FUNCTION OnRoundEnded(event e)
        hitted ← false
        Rescan()

    FUNCTION OnBotDeath(event e)
        IF botId == e.VictimId
            hitted ← false
        ENDIF

    FUNCTION ToTarget(x, y)
        DECLARE direct AS VAR ← DirectionTo(x, y)
        DECLARE gunBear AS VAR ← NormalizeRelativeAngle(direct -
                                                    GunDirection)

    FUNCTION ToBot(x, y, far)
        DECLARE dist AS VAR ← DistanceTo(x, y)
        SET TargetSpeed ← 8
        Forward(dist + far)

    END SmartVoraRam

```

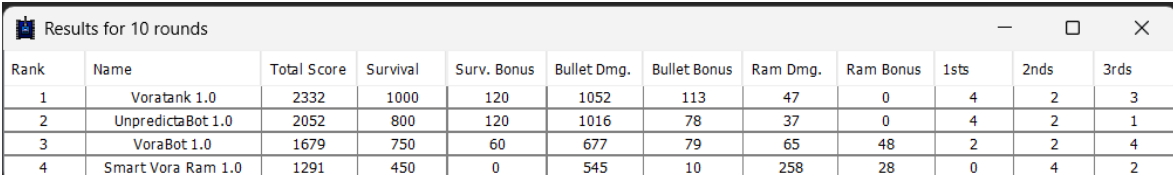
Berdasarkan pseudocode di atas, berikut merupakan penjelasan dari masing-masing fungsi dan prosedur yang digunakan

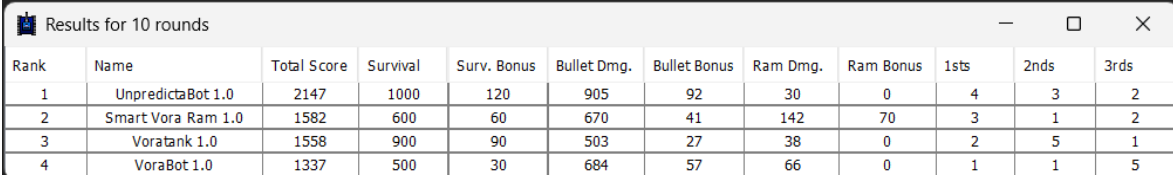
Class	
public class SmartVoraRam	Kelas inheritance dari Robocode Bot. Kelas ini merupakan kelas utama bot yang berisi seluruh data, fungsi, dan prosedur yang akan digunakan untuk menjalankan bot.
Attribute	
turnDirect	Arah putar radar, searah (negatif) atau berlawanan arah (positif) jarum jam.

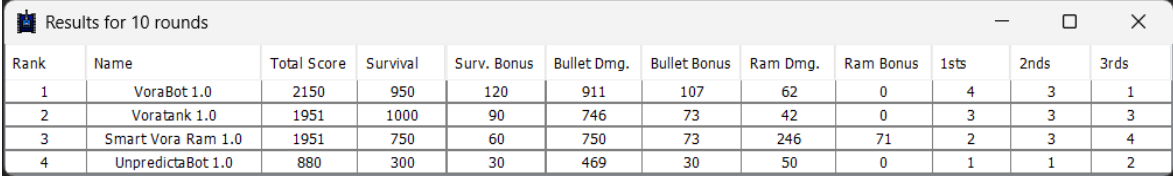
hitted	Kondisi apakah sedang menargetkan bot lawan atau tidak.
botId	ID bot yang sedang terdeteksi.
Fungsi dan Prosedur	
FUNCTION Main() CALL Start()	Memulai bot dengan memanggil Start()
FUNCTION Run()	Menjalankan logika utama bot seperti mengatur warna body, gun, dan radar bot, serta mengatur pergerakan <i>default</i> bot.
FUNCTION OnScannedBot(event e)	Fungsi ketika radar mendeteksi adanya bot musuh. Bekerja dengan cara menyimpan posisi terakhir musuh yang terdeteksi, kemudian berhenti untuk mengarahkan senjata ke target dan menembaknya dengan kekuatan yang disesuaikan berdasarkan kecepatan dan energi bot target.
FUNCTION OnHitBot(event e)	Fungsi ketika bot bertabrakan dengan bot lain. Bekerja dengan cara mengarahkan senjata ke bot musuh yang menabrak atau ditabrak, lalu menembaknya dengan <i>firepower</i> tertentu sampai bot lawan tersebut hancur atau pindah posisi.
FUNCTION OnHitWall(event e)	Fungsi ketika bot bertabrakan dengan dinding. Bekerja dengan cara scan ulang bot lawan yang tersedia.
FUNCTION OnBulletedHit(event e)	Fungsi ketika peluru berhasil mengenai bot musuh. Bekerja dengan cara mengarahkan radar ke bot lawan tersebut untuk mengecek apakah layak untuk ditabrak atau tidak.
FUNCTION OnHitByBullet(event e)	Fungsi ketika bot ditembak oleh musuh. Bekerja dengan cara mengarahkan radar ke posisi peluru tersebut untuk mengecek apakah terdapat bot lawan atau tidak.
FUNCTION OnRoundEnded(event e)	Fungsi yang dieksekusi saat ronde sudah berakhir. Bekerja dengan cara keluar dari

	while loop apabila sedang menembak bot lawan. Lalu, scan ulang seluruh area.
FUNCTION OnBotDeath(event e)	Fungsi ketika terdapat bot yang sudah hancur. Bekerja dengan mengidentifikasi ID bot lawan yang sudah hancur. Apabila bot yang hancur adalah bot lawan yang ditabrak, maka bot berhenti menembak, lalu melakukan scan dengan cara mengubah arah bot. Apabila bot yang hancur bukan bot yang sedang ditabrak, maka bot tetap menembak dan menabrak bot lawan tersebut.
FUNCTION ToTarget(x, y)	Fungsi untuk mengarahkan senjata bot ini kepada bot lawan yang terdeteksi.
FUNCTION ToBot(x, y, far)	Fungsi untuk mendatangi bot lawan yang layak untuk ditabrak.

b. Hasil pengujian

Percobaan 1											
											
Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bonus	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	Voratank 1.0	2332	1000	120	1052	113	47	0	4	2	3
2	UnpredictaBot 1.0	2052	800	120	1016	78	37	0	4	2	1
3	VoraBot 1.0	1679	750	60	677	79	65	48	2	2	4
4	Smart Vora Ram 1.0	1291	450	0	545	10	258	28	0	4	2

Percobaan 2											
											
Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bonus	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	UnpredictaBot 1.0	2147	1000	120	905	92	30	0	4	3	2
2	Smart Vora Ram 1.0	1582	600	60	670	41	142	70	3	1	2
3	Voratank 1.0	1558	900	90	503	27	38	0	2	5	1
4	VoraBot 1.0	1337	500	30	684	57	66	0	1	1	5

Percobaan 3											
											
Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bonus	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	VoraBot 1.0	2150	950	120	911	107	62	0	4	3	1
2	Voratank 1.0	1951	1000	90	746	73	42	0	3	3	3
3	Smart Vora Ram 1.0	1951	750	60	750	73	246	71	2	3	4
4	UnpredictaBot 1.0	880	300	30	469	30	50	0	1	1	2

| Percobaan 4 | | | | | | | | | | | |

Results for 10 rounds											
Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bonus	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	UnpredictaBot 1.0	1911	800	60	814	38	178	21	2	4	3
2	Smart Vora Ram 1.0	1905	700	60	884	91	169	0	2	2	4
3	VoratanK 1.0	1641	850	90	623	46	31	0	3	3	1
4	VoraBot 1.0	1437	650	90	604	38	54	0	3	1	2

Results for 10 rounds											
Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bonus	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	VoraBot 1.0	2088	1000	90	728	68	162	39	3	3	3
2	Smart Vora Ram 1.0	2080	650	90	950	73	257	59	3	2	2
3	VoratanK 1.0	1664	800	60	664	42	97	0	2	4	2
4	UnpredictaBot 1.0	1385	550	60	683	53	38	0	2	1	3

Results for 10 rounds											
Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bonus	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	VoratanK 1.0	2326	1050	150	1003	100	23	0	4	3	1
2	UnpredictaBot 1.0	1669	650	90	826	64	38	0	3	1	3
3	VoraBot 1.0	1594	800	60	605	43	56	29	3	1	4
4	Smart Vora Ram 1.0	1349	500	0	589	31	208	21	0	5	2

Results for 10 rounds											
Rank	Name	Total Score	Survival	Surv. Bonus	Bullet Dmg.	Bullet Bonus	Ram Dmg.	Ram Bonus	1sts	2nds	3rds
1	VoratanK 1.0	2045	1050	150	721	107	17	0	3	4	1
2	UnpredictaBot 1.0	2033	800	60	1016	88	68	0	5	1	1
3	VoraBot 1.0	1665	800	90	607	88	70	10	2	3	4
4	Smart Vora Ram 1.0	1048	350	0	491	0	202	5	0	2	4

Dari hasil pengujian sebanyak 7 kali dengan mempertandingkan 1 bot utama dan 3 bot alternatif secara bersamaan, didapatkan bahwa VoratanK mendominasi pertandingan dengan menang sebanyak 3 kali. Berikut merupakan analisis dari tiap bot dari segi penyerangan, pertahanan, dan agresivitas.

Bot	Efektivitas Penyerangan	Pertahanan	Agresivitas
VoratanK	Memiliki efektivitas menyerang yang tinggi karena serangan yang dilakukan berdasarkan kecepatan target yang terdeteksi. Kemudian, bot juga memiliki mekanisme untuk menyerang kembali bot yang menembaknya dengan	Mekanisme pertahanan VoratanK cukup baik dalam menangani ketika bot menabrak sebuah dinding. Kemudian, jika tertembak oleh bot musuh, VoratanK juga memiliki pertahanan dengan menghindari posisi awal ketika bot terkena serangan untuk menghindari serangan lanjutan. Terakhir, mekanisme pertahanannya yaitu ketika bot	Bot tidak terlalu agresif karena bot lebih mengutamakan efektivitas pada penyerangan. Penyerangan dan gerakan bot dilakukan secara sederhana tanpa ada mekanisme khusus yang

	mendeteksi arah serangan bot musuh. Terakhir, bot juga memiliki mekanisme untuk menembak lagi musuh yang berhasil tertembak oleh bot.	telah menembakkan peluru, bot akan berpindah tempat untuk menghindari serangan balasan dari bot musuh jika serangan mengenai bot musuh.	signifikan.
UnpredictaBot	UnpredictaBot memiliki efektivitas menyerang yang sangat tinggi karena dilakukan dengan cara mengunci target atau fokus pada satu target.	Bot ini memiliki pertahanan yang cukup baik karena akan menghindari dari bot musuh yang menembak atau tertabrak dengan bot ini.	Bot ini memiliki agresivitas yang kurang baik karena hanya menyerang bot musuh yang menembak bot ini sebanyak sekali.
VoraBot	VoraBot memiliki strategi penyerangan yang cukup baik dengan menyerang musuh berdasarkan jarak VoraBot dengan bot musuh. Tetapi, hal ini masih belum cukup baik karena tidak mendeteksi kecepatan bot musuh sehingga peluang untuk peluru meleset lebih tinggi dibandingkan Voratank. VoraBot juga memiliki mekanisme yang mirip dengan Voratank jika terdapat bot musuh yang menyerang dan jika peluru mengenai bot musuh.	VoraBot juga memiliki pertahanan yang serupa dengan Voratank seperti mekanisme menghindari dinding dan serangan lawan. Hanya saja, pergerakan VoraBot untuk menghindari serangan lawan adalah dengan menggunakan pergerakan zigzag.	Bot memiliki pergerakan zigzag dalam menghindari serangan lawan. Bot ini masih kurang agresif jika dibandingkan dengan SmartVoraRam tetapi masih lebih agresif dari Voratank dan UnpredictaBot walaupun tidak terlalu signifikan.
SmartVoraRam	Masih sama dengan bot-bot lainnya, SmartVoraRam memiliki efektivitas yang cukup tinggi dalam menyerang musuh karena menyerang	Pertahanan bot ini masih kurang karena hanya melakukan scan ulang jika tertabrak dinding. Hal ini berpotensi bot menabrak dinding secara terus menerus. Mekanisme menabrak bot pun membuat pertahanan pada bot	Bot memiliki agresivitas yang cukup tinggi karena akan menabrak bot musuh yang layak untuk ditabrak, dalam hal ini

	berdasarkan kecepatan bot dan sisa nyawa bot musuh.	ini kurang baik.	adalah bot dengan sisa nyawa sedikit.
--	---	------------------	--

Bab 5

Kesimpulan dan Saran

a. Kesimpulan

Pada tugas ini, penulis telah berhasil membuat 4 alternatif bot (1 bot utama dan 3 bot alternatif) yang menggunakan strategi yang berbeda-beda. Strategi yang digunakan, yaitu Spam Fire Damage, pergerakan zigzag, penguncian target, dan Spam Ram Fire, yang sudah dijelaskan pada tabel strategi di Bab 3. Berdasarkan keempat strategi tersebut, bot yang paling dominan menghasilkan skor tertinggi adalah Voratank dengan total dominan skor tertinggi sebanyak 3 kali dari 7 kali simulasi.

Voratank memiliki strategi Spam Fire Damage, yaitu menembak secara terus-menerus ke bot lawan yang sedang terdeteksi. Selain itu, bot ini menyerang lawan berdasarkan kecepatan bot lawan tersebut agar selalu tepat sasaran saat menembak bot lawan. Kemudian, apabila bot ini menabrak atau ditabrak bot lawan, bot ini akan menembak langsung bot lawan tersebut dengan kekuatan tembakan yang cukup tinggi agar bot lawan cepat hancur.

Oleh karena itu, penulis berhasil melakukan simulasi pada keempat bot yang sudah dibuat dengan strategi yang berbeda-beda, yaitu Voratank, Vorabot, UnpredictaBot, dan SmartVoraRam yang berhasil disimulasikan dalam bahasa C#.

b. Saran

Saran untuk kelompok ini adalah sebagai berikut.

- (1) Lebih mengeksplor lagi terkait algoritma greedy yang lebih efektif dan efisien
- (2) Meningkatkan komunikasi antar anggota kelompok
- (3) Mempelajari API Robocode lebih dalam agar menghasilkan bot yang optimal

PEMBAGIAN TUGAS

NIM	Nama	Pembagian Tugas
12821046	Fardhan Indrayesa	<p>Implementasi</p> <ul style="list-style-type: none"> - Voratank - SmartVoraRam <p>Laporan</p> <ul style="list-style-type: none"> - Bab 2: SmartVoraRam - Bab 3: Voratank, SmartVoraRam - Bab 4: Implementasi Voratank dan SmartVoraRam - Bab 5: Kesimpulan dan saran
13523132	Jonathan Levi	<p>Implementasi</p> <ul style="list-style-type: none"> - UnpredictaBot <p>Laporan</p> <ul style="list-style-type: none"> - Bab 1: Deskripsi Tugas - Bab 2: Konsep Algoritma <i>Greedy</i>, Voratank, UnpredictaBot - Bab 3: UnpredictaBot, analisis efisiensi dan efektivitas solusi greedy, alasan pemilihan strategi <i>greedy</i>. - Bab 4: Implementasi UnpredictaBot, analisis efektivitas serangan, pertahanan, dan agresivitas UnpredictaBot.
13622076	Ziyan Agil Nur Ramadhan	<p>Implementasi</p> <ul style="list-style-type: none"> - VoraBot <p>Laporan</p> <ul style="list-style-type: none"> - Bab 2: VoraBot - Bab 4: Implementasi VoraBot, percobaan pertandingan, serta analisis efektivitas serangan, pertahanan, dan agresivitas Voratank, VoraBot, SmartVoraRam

LAMPIRAN

Tautan *repository* GitHub : [Tubes1_VoratanK](#)
Tautan video : [Video Kelompok 42](#)

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, R. (2025). *Algoritma Greedy*. Homepage Rinaldi Munir. Retrieved March 20, 2025, from [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)

Robocode Tank Royale Docs. (2024, July 15). Retrieved March 10, 2025, from <https://robocode-dev.github.io/tank-royale/articles/intro.html>