

Laporan Tugas Besar

IF2211 Strategi Algoritma



Dibuat oleh:

13520026 Muhammad Fajar Ramadhan

13520079 Ghebyon Tohada Nainggolan

13520120 Afrizal Sebastian

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

Daftar Isi

Bab I Deskripsi Tugas	2
Bab II Landasan Teori	4
2.1 Algoritma Greedy	4
2.1.1 Elemen - Elemen Algoritma Greedy	4
2.2 Game Engine	4
2.3 Game Engine Entellect Overdrive	4
2.3.1 Membangun Starter Bot dalam Bahasa Java	5
Bab III Aplikasi Strategi Greedy	6
3.1. Solusi-Solusi Algoritma Greedy	6
3.1.1 Greedy Tweet	6
3.1.2 Greedy Turning Lane	7
3.1.3 Greedy Fix	8
3.1.4 Greedy Lizard	9
3.1.5 Greedy EMP	10
3.1.6 Greedy Boost	12
3.1.7 Greedy Oil	13
3.2 Solusi Algoritma Greedy yang Dipilih dan Pertimbangannya	14
Bab IV Implementasi	16
4.1 Pseudocode	16
4.2 Struktur Data	24
4.3 Analisis dan Pengujian	26
4.3.1 Tweet V1	26
4.3.2 Tweet V2	27
4.3.3 Fix	29
4.3.4 Turning	30
4.3.5 Lizard	31
4.3.6 Boost	32
4.3.7 Oil	33
4.3.8 EMP	34
Bab V Kesimpulan dan Saran	36
Daftar Pustaka	37
Lampiran	38

Bab I Deskripsi Tugas

Overdrive merupakan sebuah permainan yang mempertandingkan dua bot mobil dalam sebuah ajang balapan. Setiap pemain akan memiliki sebuah bot mobil dan masing-masing bot akan saling bertanding untuk mencapai garis finish dan memenangkan pertandingan. Agar dapat memenangkan pertandingan, setiap pemain harus mengimplementasikan strategi tertentu untuk dapat mengalahkan lawannya. Gambar 1. Ilustrasi permainan Overdrive Pada tugas besar pertama Strategi Algoritma ini, gunakanlah sebuah game engine yang mengimplementasikan permainan Overdrive. Game engine dapat diperoleh pada laman berikut:



<https://github.com/EnteletecChallenge/2020-Overdrive>

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

<https://github.com/EnteletecChallenge/2020-Overdrive/releases/tag/2020.3.4>

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

1. Peta permainan memiliki bentuk array 2 dimensi yang memiliki 4 jalur lurus. Setiap jalur dibentuk oleh block yang saling berurutan, panjang peta terdiri atas 1500 block. Terdapat 5 tipe block, yaitu Empty, Mud, Oil Spill, Flimsy Wall, dan Finish Line yang masing-masing karakteristik dan efek berbeda. Block dapat memuat powerups yang bisa diambil oleh mobil yang melewati block tersebut.

2. Beberapa powerups yang tersedia adalah:
 - a. Oil item, dapat menumpahkan oli di bawah mobil anda berada.
 - b. Boost, dapat mempercepat kecepatan mobil anda secara drastis.
 - c. Lizard, berguna untuk menghindari lizard yang mengganggu jalan mobil anda.
 - d. Tweet, dapat menjatuhkan truk di block spesifik yang anda inginkan.
 - e. EMP, dapat menembakkan EMP ke depan jalur dari mobil anda dan membuat mobil musuh (jika sedang dalam 1 lane yang sama) akan terus berada di lane yang sama sampai akhir pertandingan. Kecepatan mobil musuh juga dikurangi 3.
3. Bot mobil akan memiliki kecepatan awal sebesar 5 dan akan maju sebanyak 5 block untuk setiap round. Game state akan memberikan jarak pandang hingga 20 block di depan dan 5 block di belakang bot sehingga setiap bot dapat mengetahui kondisi peta permainan pada jarak pandang tersebut.
4. Terdapat command yang memungkinkan bot mobil untuk mengubah jalur, mempercepat, memperlambat, serta menggunakan powerups. Pada setiap round, masing-masing pemain dapat memberikan satu buah command untuk mobil mereka. Berikut jenis-jenis command yang ada pada permainan:
 - a. NOTHING
 - b. ACCELERATE
 - c. DECELERATE
 - d. TURN_LEFT
 - e. TURN_RIGHT
 - f. USE_BOOST
 - g. USE_OIL
 - h. USE_LIZARD
 - i. USE_TWEET
 - j. USE_EMP
 - k. FIX
5. Command dari kedua pemain akan dieksekusi secara bersamaan (bukan sekuensial) dan akan divalidasi terlebih dahulu. Jika command tidak valid, bot mobil tidak akan melakukan apa-apa dan akan mendapatkan pengurangan skor.
6. Bot pemain yang pertama kali mencapai garis finish akan memenangkan pertandingan. Jika kedua bot mencapai garis finish secara bersamaan, bot yang akan memenangkan pertandingan adalah yang memiliki kecepatan tercepat, dan jika kecepatannya sama, bot yang memenangkan pertandingan adalah yang memiliki skor terbesar.

Adapun peraturan yang lebih lengkap dari permainan Overdrive, dapat dilihat pada laman :
<https://github.com/EntelectChallenge/2020-Overdrive/blob/develop/game-engine/game-rules.md>

Bab II Landasan Teori

2.1 Algoritma Greedy

Algoritma *greedy* merupakan sebuah paradigma dalam penyelesaian persoalan secara bertahap. Solusi yang diberikan oleh algoritma *greedy* disusun *step by step* atau langkah demi langkah. Pada setiap langkah tersebut akan diambil keputusan yang paling optimal. Keputusan optimal yang diambil adalah secara lokal dengan harapan akan membawa solusi akan optimal secara global.

2.1.1 Elemen - Elemen Algoritma *Greedy*

Beberapa elemen peninjauan pada algoritma *greedy* adalah sebagai berikut.

1. Himpunan kandidat, C : berisi kandidat yang akan dipilih pada setiap Langkah (misal: simpul/sisi di dalam graf, job, task, koin, benda, karakter, dsb)
2. Himpunan solusi, S : berisi kandidat yang sudah dipilih
3. Fungsi solusi: menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi
4. Fungsi seleksi (selection function): memilih kandidat berdasarkan strategi greedy tertentu. Strategi greedy ini bersifat heuristik.
5. Fungsi kelayakan (feasible): memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi (layak atau tidak)
6. Fungsi obyektif : memaksimumkan atau memminimumkan

2.2 Game Engine

Game engine adalah sebuah perngkat lunak yang dapat digunakan untuk mengembangkan dan menciptakan sebuah *game*. Fungsi *game engine* adalah rendering game ke 2 dimensi atau 3 dimensi, physics engine, pengaturan audio, scripting, pengatur penampilan animasi, pengaturan memori, pengaturan grafis, networking dan streaming data dan masih banyak lagi

2.3 *Game Engine* Entellect Overdrive

Game Entellect – 2020 Overdrive memiliki beberapa komponen bawaan agar game dapat dijalankan, yaitu :

1. *Game-engine* : *game-engine* bertanggung jawab atas jalannya permainan. *Game engine* akan mengeksekusi perintah yang dijalankan oleh bot jika perintah yang diberikan bot valid dan dapat dijalankan.
2. *Game-runner* : *Game-runner* bertanggung jawab menjalankan pertandingan antarpemain.
3. *Reference-bot* : *bot* yang disediakan untuk menguji hasil bot yang telah dibuat. Default *reference-bot* ditulis dalam bahasa java.

4. *Starter-bot* : bot dengan logika dasar dan akan dikembangkan menggunakan algoritma greedy.

2.3.1 Membangun Starter Bot dalam Bahasa Java

Pertama adalah men-download starter-pack pada <https://github.com/EntelectChallenge/2020-Overdrive/releases/tag/2020.3.4>. Dalam pembuatan bot, pada tugas ini digunakan bahasa java dalam membangun bot. Karena menggunakan bahasa java, pada folder *java/src* akan terdapat 3 folder dan 2 file, yaitu:

1. Folder Command, yang berisi file *class-class Command* yang nantinya dapat digunakan dalam mengembangkan strategi dalam membangun bot.
2. Folder entities, yang berisikan file yang berisikan objek-objek yang ada pada game seperti *car, wall, oil*, dll
3. Folder enums, yang berisikan file pendukung game, seperti *powerups*, objek pada lane, dll
4. Bot.java, File yang berisikan logika dan strategi yang akan dibangun. Pada file ini akan diimplementasikan algoritma greedy dengan memanfaatkan file-file diatas.
5. Main.java, file yang akan menjalankan *bot.java*

Untuk menjalankan bot, dibutuhkan beberapa aplikasi seperti Java JDK 8, IntelliJ IDEA, dan NodeJS. File *bot.java* dan *main.java* terlebih dahulu *dibuild* sebagai *file executable java* (.jar) yang nantinya digunakan untuk menjalankan bot di game. Namun, sebelum menjalankan game terlebih dahulu dilakukan beberapa konfigurasi file pada :

1. *game-runner-config.json* : file yang diberisikan konfigurasi lokasi file kedua pemain, hasil pertandingan, *game engine yang digunakan, dan konfigurasi game yang digunakan*.
2. *game-config.json* : file yang berisikan konfigurasi game, seperti panjang lintasan, banyak lane, dll.
3. *bot.json* : file yang berisikan identitas bot yang dibuat.

Untuk menggunakan *visualizer*, dapat menggunakan (<https://entelect-replay.raezor.co.za/#>) agar hasil dapat terlihat lebih rinci. Untuk menggunakan *visualizer* hasil pertandingan yang telah dijalankan menggunakan *run.bat* akan tersimpan pada folder *match-log*. Folder hasil pertandingan harus di *zip* terlebih dahulu. Setelah folder telah di-zip, *upload* folder pada *visualizer*.

Bab III Aplikasi Strategi *Greedy*

3.1. Solusi-Solusi Algoritma *Greedy*

3.1.1 *Greedy Tweet*

Algoritma *greedy tweet* ini merupakan salah satu strategi dalam penggunaan powerup *tweet*. Strategi yang diimplementasikan adalah dengan melihat kondisi dan posisi musuh sehingga penggunaan *Tweet* diharapkan akan selalu mengenai musuh. Pada setiap ronde, mobil akan melakukan pengecekan ketersediaan power up *Tweet* ini, jika tersedia maka akan digunakan berdasarkan kondisi musuh. *Tweet* akan digunakan ketika musuh sedang melakukan *fix* atau posisi musuh sudah jauh didepan atau tertinggal dibelakang. *Tweet* akan digunakan untuk memunculkan *obstacle Cybertruck* di depan musuh. Penempatan *Cybertruck* bervariasi tergantung kondisi musuh. Jika di depan musuh tidak terdapat *obstacle* maka akan langsung dimunculkan didepannya. Jika terdapat *obstacle* di depan musuh akan dimunculkan pada *lane* sebelah kiri atau kanan tergantung *obstacle* pada *lane* tersebut. Jika *lane* sebelah kiri musuh tidak terdapat *obstacle* dan *lane* sebelah kanan terdapat *obstacle*, maka akan dimunculkan pada *lane* sebelah kiri begitu juga untuk keadaan sebaliknya. Strategi ini digunakan dengan harapan musuh akan menabrak *Cybertruck* ketika sedang menghindari *obstacle*.

a. Mapping elemen-elemen *greedy*

- Himpunan kandidat : *Lane* mobil musuh saat ini, *lane* mobil musuh sebelah kiri, dan *lane* mobil musuh sebelah kanan
- HImpunan solusi : *Lane* musuh jika tidak terdapat *obstacle* di depannya atau *lane* sebelah kanan atau kiri tergantung *obstacle* pada *lane* tersebut
- Fungsi solusi : Memeriksa *lane* musuh saat ini, sebelah kiri, dan sebelah kanannya
- Fungsi kelayakan : Memeriksa *obstacle* pada *lane* saat ini jika ada periksa *lane* sebelah kiri dan kanan, jika tidak kembalikan *lane* saat ini
- Fungsi objektif : Gunakan *Tweet Boost* pada *lane* di himpunan solusi

b. Analisis Efisiensi Solusi

Algoritma akan memeriksa 3 *lane* yaitu *lane* musuh saat ini, *lane* sebelah kiri, dan *lane* sebelah kanan. Pada setiap *lane* akan diperiksa n buah blok, untuk mengetahui apakah terdapat *obstacle* atau tidak pada *lane* tersebut. Sehingga kompleksitasnya adalah:

$$T(n) = 3 * n = O(n)$$

dengan n adalah kecepatan mobil saat ini

c. Analisis Efektivitas Solusi

Strategi ini efektif apabila :

- Terdapat *obstacle* pada *lane* ini dan salah satu *lane* kanan atau kiri sehingga musuh akan terkena *Cybertruck* ketika belok menghindari *obstacle*.
- Musuh sedang melakukan *fix* pada mobilnya sehingga *Cybertruck* dapat langsung dimunculkan di depannya

Strategi ini tidak efektif apabila :

- Terdapat *obstacle* pada *lane* musuh namun tidak di sebelah kanan dan kirinya sehingga tidak dapat diprediksi penggunaan *Tweet Boost* ditempatkan pada *lane* sebelah kiri atau kanan musuh

3.1.2 Greedy Turning Lane

Algoritma *Greedy Turning Lane* merupakan salah satu strategi dalam menentukan penggunaan perintah belok atau tetap di *lane*. Strategi yang diimplementasikan adalah dengan memeriksa *lane* pemain saat ini, *lane* sebelah kiri, dan *lane* sebelah kanan. Setiap ronde akan diperiksa apakah terdapat *obstacle* di depan pemain dan *damage* yang akan diterima pada *lane* yang menjadi kandidat untuk ditempati ronde selanjutnya. Algoritma lalu menentukan perintah optimal dengan membandingkan *damage* setiap *lane*. *Lane* yang akan dipilih adalah *lane* dengan *damage* minimal. Jika terdapat *lane* lebih dari 1 dengan *damage* minimal maka akan dipilih berdasarkan banyaknya *powerup* pada *lane* tersebut.

a. Mapping elemen-elemen greedy

- Himpunan kandidat : *Lane* mobil pada saat ini, *lane* sebelah kiri, dan *lane* sebelah kanan.
- Himpunan solusi : *Lane* yang memiliki *damage* paling minimal.
- Fungsi Solusi : Memeriksa *lane* apakah memiliki *obstacle* atau tidak
- Fungsi Seleksi : Memeriksa *damage* untuk setiap *lane*
- Fungsi Kelayakan : Memeriksa *lane* yang dipilih ialah memiliki *damage* paling minimal
- Fungsi Objektif : Minimisasi *damage* yang akan diterima ketika akan pindah *lane*

b. Analisis Efisiensi Solusi

Algoritma akan memeriksa 3 *lane* yaitu *lane* pemain saat ini, *lane* sebelah kiri, dan *lane* sebelah kanan. Pada setiap *lane* akan diperiksa n buah blok, untuk mengetahui apakah terdapat *obstacle* atau tidak pada *lane* tersebut. Pada setiap *lane* juga akan diperiksa

sebanyak n buah blok untuk kalkulasi *damage* yang ditimbulkan. Sehingga kompleksitasnya adalah :

$$T(n) = (3 * n) + (3 * n) = 6n = O(n)$$

dengan n adalah kecepatan mobil saat ini

Untuk kasus tambahan akan diperiksa jumlah *powerups* pada lane tersebut sehingga kompleksitasnya menjadi :

$$T(n) = (3 * n) + (3 * n) + (3 * n) = 9n = O(n)$$

dengan n adalah kecepatan mobil saat ini

c. Analisis Efektivitas Solusi

Strategi ini efektif apabila :

- Terdapat perbedaan *damage* yang signifikan pada setiap lane nya sehingga akan meminimalkan damage yang akan diterima pemain.

Strategi ini tidak efektif apabila :

- Terdapat kemungkinan bahwa *lane* yang dihindari mengandung *powerups* lebih banyak daripada *lane* yang menjadi solusi serta perbedaan damage yang tidak terlalu signifikan.

3.1.3 Greedy Fix

Algoritma *Greedy Fix* merupakan salah satu strategi dalam menentukan penggunaan perintah *fix* untuk memperbaiki *damage*. Strategi yang diimplementasikan adalah dengan memeriksa *damage* mobil saat ini. Jika sudah lebih besar dari 3 maka akan langsung melakukan perintah *fix*. Jika *damage* sudah terlalu kecil akan diperiksa apakah mobil berada pada *lane* yang aman atau tidak untuk melakukan *fix*. Aman yang dimaksud adalah tidak dalam jangkauan *emp* musuh dan tidak ada *obstacle* pada *block* di depan pemain.

a. Mapping elemen-elemen greedy

- Himpunan kandidat : *fix* digunakan, *fix* tidak digunakan
- Himpunan solusi : *fix* digunakan.
- Fungsi Solusi : Memeriksa *lane* pemain
- Fungsi Seleksi : Memeriksa *obstacle* dan posisi musuh
- Fungsi Kelayakan : Memeriksa apakah posisi pemain aman

- Fungsi Objektif : Minimisasi kemungkinan *damage* yang diterima lebih besar setelah *fix*

b. Analisis Efisiensi Solusi

Algoritma akan memeriksa keadaan *damage* pemain (1) maka kompleksitasnya adalah :

$$T(n) = 1 = O(1)$$

Untuk kasus tambahan jika damage terlalu kecil, maka akan diperiksa *lane* pemain sebanyak n buah blok dan memeriksa posisi musuh sehingga kompleksitasnya :

$$T(n) = n + 1 = O(n)$$

c. Analisis Efektivitas Solusi

Strategi ini efektif apabila :

- Musuh mempunyai *powerup emp* sehingga bisa dihindari penggunaannya ketika kita sedang melakukan *fix*

Strategi ini tidak efektif apabila :

- Posisi pemain tidak aman karena terlalu banyak *obstacle* untuk berada dalam posisi aman sehingga akan secara konstan melakukan *fix* hingga kondisi yang memungkinkan untuk *accelerate*

3.1.4 Greedy Lizard

Greedy by Power Ups Lizard adalah strategi dalam melewati *obstacle* berupa *wall*, *mud*, dan/atau *oil_spill*. Pada setiap ronde, mobil akan melakukan pengecekan terhadap jalur yang ada dihadapannya sebanyak kecepatan mobil saat ini, serta melakukan pengecekan terhadap mobil musuh yang mungkin ada di depan dan dapat didahului. Jika pada *range* tersebut ada *obstacle* dan/atau mobil musuh dan mobil saat ini memiliki *lizard* maka mobil akan mengaktifkan *power ups* tersebut.

a. Mapping elemen-elemen greedy

- Himpunan kandidat : *Lane* mobil pada saat ini.
- Himpunan solusi : *Lane* yang memiliki *obstacle* dan/atau mobil musuh.

- Fungsi Solusi : Memeriksa *lane* apakah memiliki *obstacle* dan/atau mobil musuh pada *lane* yang sama
- Fungsi Seleksi : Pilih *lane* memiliki *obstacle* dan/atau mobil musuh pada *lane* yang sama
- Fungsi Kelayakan : Memeriksa apakah *obstacle* dan/atau mobil musuh masih dalam jangkauan kecepatan mobil dan mobil memiliki *lizard*.
- Fungsi Objektif : Mempercepat gerakan mobil tanpa berpindah *lane* dan tanpa menabrak *obstacle*.

b. Analisis Efsiensi Solusi

Terdapat n blok yang akan dicek isi dari blok tersebut apakah ada *obstacle* dan/atau mobil musuh lalu menjalankan *Command*. Pengecekan ada-tidaknya *Lizard* (m) sehingga kompleksitasnya adalah :

$$T(n + m) = n + m = O(n + m),$$

dengan n adalah kecepatan mobil dan m banyak powerups saat ini

c. Analisis Efektivitas Solusi

Strategi ini efektif apabila :

- Terdapat *obstacle* pada jalur saat ini dan kecepatan yang dimiliki mobil adalah kecepatan maksimal atau kecepatan dengan *boost* sehingga tidak perlu terkena *obstacle* atau mengurangi kecepatan mobil saat ini.

Strategi ini tidak efektif apabila :

- Terdapat kemungkinan bahwa mobil lawan teradapat pada blok yang sama dengan mobil saat telah menggunakan skill ini sehingga akan terjadi kecelakan antarmobil.

3.1.5 Greedy EMP

Greedy by Power Ups EMP adalah strategi menembak mobil musuh yang berada di depan mobil saat ini menggunakan *EMP*. Untuk setiap ronde, jika pada *lane* saat ini tidak ada *obstacle*. Jika mobil musuh berada 2 *block* di depan dan tidak berada pada *lane* serta mobil saat ini memiliki *EMP*, maka *EMP* akan ditembakkan. Namun, terdapat kasus khusus ketika mobil saat ini berada pada *lane* yang sama, maka harus melakukan cek terlebih dahulu jika mobil saat ini tidak melebihi posisi musuh pada *round* selanjutnya. Jika sudah melakukan tembakan, maka pada round selanjutnya tidak melakukan tembakan lagi.

a. Mapping elemen-elemen greedy

- Himpunan Kandidat : Posisi mobil dan posisi mobil musuh saat ini.
- Himpunan Solusi : Selesih *block* mobil musuh dengan *block* mobil lebih dari 2 dan jika *lane* mobil musuh sama dengan mobil dengan syarat *block* mobil tidak melebihi *block* musuh di *round* selanjutnya.
- Fungsi Solusi : Memeriksa *block* musuh dan diri sendiri dan memiliki selisih lebih dari 2 atau memeriksa *lane* musuh dan diri sendiri sama, lalu memeriksa *block* diri sendiri tidak melebihi lawan pada *round* selanjutnya.
- Fungsi Seleksi : Pilih keadaan Selesih *block* mobil musuh dengan *block* mobil lebih dari 2 jika *lane* mobil musuh sama dengan mobil dengan syarat *block* mobil tidak melebihi *block* musuh di *round* selanjutnya.
- Fungsi Kelayakan : Mobil memiliki *EMP*.
- Fungsi Objectif : Memperlambat kecepatan lawan yang berada di depan.

b. Analisis Efiseinsi Solusi :

Memeriksa ada-tidaknya *EMP* pada mobil saat ini (n) dengan linaer search. Perbandingan posisi saat ini (1), sehingga kompleksitasnya :

$$T(n) = n + 1 = O(n),$$

dengan n banyak powerups saat ini.

c. Analisis Efektivitas Solusi :

Strategi ini efektif apabila :

- Mobil musuh berada di 2 *block* atau lebih dari mobil sendiri dan tidak berada pada *lane* yang sama sehingga kemungkinan terkena *EMP* lebih besar sehingga mobil melambat.

Strategi ini tidak efektif apabila:

- Mobil musuh berada pada *lane* yang sama sehingga ada kemungkinan terjadi tabrakan antar mobil, mobil lawan mempunyai strategi yang dapat menghindari *EMP* atau Mobil pada lane 1 atau 4 dan lawan berada di lane 2 atau 3 yang berjarak 2 didepan, sehingga lawan mungkin dapat menghindar.

3.1.6 Greedy Boost

Greedy by Power Ups Boost adalah strategi dalam menggunakan *boost*. Jika menggunakan skill ini maka kecepatan mobil akan menjadi 15. Pada setiap ronde, diperiksa apakah pada *lane* saat memiliki *obstacle*. Jika tidak memiliki *obstacle*, memiliki *boost*, dan *boost counting* sama dengan 0 maka akan digunakan *boost*.

a. Mapping elemen-elemen Greedy :

- Himpunan Kandidat : *Lane* yang dilewati pada saat ini.
- Himpunan Solusi : *Lane* yang tidak memiliki *obstacle* pada 15 *block* di depan.
- Fungsi Solusi : Memeriksa apakah *lane* pada saat ini memiliki *obstacle* pada 15 *block* didepan.
- Fungsi Seleksi : Ambil *lane* yang tidak memiliki 15 *block* didepan.
- Fungsi Kelayakan : Mobil memiliki *boost* dan memiliki *boost-counting* sama dengan 0.
- Fungsi Objektif : Membuat kecepatan saat ini menjadi 15 (melebihi kecepatan maksimal) untuk 5 round kedepan.

b. Analisis Efisiensi Solusi

Terdapat 15 *block* yang akan diperiksa. Memeriksa ada-tidaknya *boost* pada mobil (*n*), sehingga efisiensi menjadi :

$$T(n) = n + 15 = O(n),$$

dengan *n* banyaknya powerups saat ini.

c. Analisis Efektifitas Solusi :

Strategi ini efektif apabila :

- Tidak ada *obstacle* pada *lane* saat ini hingga 5 round selanjutnya, sehingga kecepatan mobil akan berada pada kecepatan *boost* sehingga mempercepat laju mobil.

Strategi ini tidak efektif apabila :

- Jika pada *round-round* selanjutnya terdapat *obstacle* sehingga harus berpindah sehingga mengurangi kecepatan atau tidak dapat berpindah sehingga menabrak *obstacle* sehingga memiliki damage yang lebih besar.

3.1.7 Greedy Oil

Greedy by Power Ups Oil adalah strategi untuk dalam menggunakan *Oil*. *Oil* akan meletakan *Oil_Spill* pada posisi kita saat ini untuk *round* selanjutnya. Jika musuh dan diri sendiri berada pada *lane* yang sama dan mobil musuh berada di belakang dengan jarak kurang dari 5 maka *Oil* akan digunakan. Jika lebih dari lima, maka akan diprediksi dengan memeriksi *lane* kanan-kiri, jika pada *lane* kanan-kiri ada *obstacle* maka kemungkinan musuh akan tetap pada *lane* saat ini.

a. Mapping elemen-elemen greedy

- Himpunan Kandidat : *Lane* mobil saat ini dan mobil musuh yang berada pada satu jalur.
- Himpunan Solusi : *Lane* dengan jarak mobil musuh dan diri sendiri kurang dari 5 atau *lane* dengan *lane* kanan dan/atau kirinya terdapat *obstacle*.
- Fungsi Solusi : Memeriksa *lane* saat ini lalu jarak antar mobil sendiri dan lawan kurang dari 5 atau memeriksa *lane* kanan-kiri yang memiliki *obstacle*.
- Fungsi Seleksi : Ambil *lane* dengan jarak mobil musuh dan diri sendiri kurang dari 5 (prioritas) atau *lane* kanan dan/atau kirinya memiliki *obstacle*.
- Fungsi Kelayakan : Mobil memiliki *Oil* yang akan digunakan.
- Fungsi Objektif : Memperlambat mobil musuh pada saat berada dibelakang.

b. Analisis Efisiensi Solusi

Melakukan Pemeriksaan pada *lane* kanan dan kiri (*n*) dan melakukan pemeriksaan ada-tidaknya *Oil* pada mobil (*m*), sehingga :

$$T(n + m) = n + m = O(n + m),$$

dengan *n* adalah kecepatan dan *m* banyak powerups saat ini.

c. Analisis Efektivitas Solusi

Strategi efektif apabila :

- Mobil musuh tepat berada di belakang mobil sendiri dengan jarak kurang dari 5 sehingga kemungkinan musuh mengenai *oil_spill* lebih besar.

Strategi tidak efektif apabila :

- Mobil musuh berada di belakang dengan jarak lebih dari 5 sehingga musuh mungkin dapat menghindari *oil_spill* yang telah dibuat.

3.2 Solusi Algoritma Greedy yang Dipilih dan Pertimbangannya

Pada permainan *Game Entellect – 2020 Overdrive* yang kami mainkan ini terdapat beberapa solusi atau alternatif strategi dalam pembuatan bot untuk menaklukan permainan ini. Banyak solusi atau algoritma *greedy* yang menjadi pertimbangan dalam implementasinya pada pembuatan bot. Beberapa pertimbangan dalam menentukan strategi dalam pembuatan bot ini diantaranya mencapai tujuan utama yaitu sampai di garis *finish* terlebih dahulu dari pada lawan, menggunakan *powerups* yang diberikan oleh sistem permainan untuk menaklukan lawan, serta mendapat skor semaksimal mungkin pada permainan.

Untuk implementasi dalam pembuatan bot ini kami menerapkan beberapa strategi *greedy* untuk memaksimalkan performa bot saat dimainkan. Untuk menangani skema dalam mencapai garis *finish* terlebih dahulu kami mengimplementasikan beberapa strategi seperti *Greedy Fix*, *Greedy Boost*, *Greedy Turning Lane*. Pertimbangan dalam implementasinya adalah strategi *Greedy Fix* memberikan solusi yang cukup baik karena ketika bot terkena *damage* maka akan berpengaruh terhadap kecepatan maksimal yang dapat dicapai oleh bot tersebut sehingga dengan *Greedy Fix* tersebut dapat menangani kasus agar kecepatan maksimum tidak berkurang. Penggunaan *Greedy Boost* tentunya memberikan solusi optimal karena dapat menambah kecepatan maksimum bot sehingga jarak yang ditempuh dalam satu ronde menjadi lebih banyak. *Greedy Turning Lane* dipakai untuk mengatasi *obstacle-obstacle (oil,mud,wall)* pada permainan agar dapat dihindari baik *damage* yang diterima maupun pinalti pada kecepatan.

Skema selanjutnya adalah bagaimana menggunakan *powerups* yang diberikan oleh sistem permainan secara efektif. Kami mengimplementasikan strategi *Greedy Boost*, *Greedy Oil*, *Greedy Tweet*, *Greedy EMP*, dan *Greedy Lizard*. Penggunaan *powerups* ini berperan penting dalam permainan karena jika tidak dilakukan dengan strategi akan terbuang sia-sia. Beberapa pertimbangan dalam mengimplementasikan strategi tersebut yaitu, seperti yang sudah dijelaskan *Greedy Boost* dapat memberikan solusi optimal agar dapat menambah kecepatan. Penerapan *Greedy Oil* dapat membuat musuh mengalami penalti pengurangan kecepatan dan membuat kondisi pemain menjadi unggul. Penerapan *Greedy Lizard* ialah untuk menghindari beberapa penalti yang awalnya harus kita terima karena tidak ada cara lain untuk menghindarinya. Pertimbangan dalam *Greedy EMP* dan *Greedy Tweet* adalah kedua boost tersebut merupakan penggunaan boost yang sangat ofensif. Kedua strategi tersebut dapat mengakibatkan dampak negatif yang signifikan terhadap lawan. Sehingga dengan kelima strategi tersebut diharapkan penggunaan *powerups* dilakukan secara efektif dan dapat membantu memenangkan permainan.

Skema terakhir adalah untuk mendapatkan skor semaksimal mungkin dalam permainan. Skema ini sudah ditangani dengan adanya penerapan beberapa strategi *powerups* pada skema sebelumnya. Pengambilan dan penggunaan *powerups* akan meningkatkan skor yang diperoleh oleh pemain. Pada skema sebelumnya juga sudah diterapkan strategi *Greedy Turning Lane* untuk menghindari penalti ketika menabrak *obstacle*. Tentunya dengan menghindari *obstacle* tersebut skor yang sudah didapat tidak akan berkurang. Pada *Greedy Turning Lane* juga

terdapat alternatif dalam pemilihan *lane* dengan *powerups* terbanyak sehingga akan menambah skor dengan semakin banyaknya *powerups* yang diambil.

Bab IV Implementasi

4.1 Pseudocode

```
function run() -> Command

DEKLARASI
    nextBlock : list of Object
    countEMP : integer          // 0 atau 1
    tweetPosition1 : Position
    tweetPosition2 : Position

BODY
    //Inisiasi atribut pendukung
    if( gamestate.currentRound = 1) then
        countEMP <- 0
        opponentFixPosition <- opponent.position
    endif

    //Strategi Memperbaiki Mobil
    if(myCar.damage >= 3) then
        -> FIX
    else if(myCar.damage = 2 or myCar.damage = 1) then
        if(isSafePosition()) then
            -> FIX
        endif

    //Strategi Menggunakan Boost
    //Periksa myCar memiliki PowerUps Boost dan sedang tidak menggunakan boost,
    //kemudian lakukan prediksi apakah Boost layak dipakai
    if(havePowerUps(PowerUps.BOOST, myCar.powerups) and myCar.boostCounter = 0 and
predictToUseBoost()) then
        -> BOOST
    endif

    nextBlock <- getBlocksInFront(myCar.position.lane, myCar.position.block, myCar.speed)

    //Periksa apakah terdapat obstacle pada Block yang dipilih
    if(isThereObstacle(nextBlock)) then
        //Jika terdapat obstacle pada nextBlock
        //Strategi Menggunakan Lizard dan Strategi Perpindahan Lane
        if(havePowerUps(PowerUps.LIZARD, myCar.powerups) and
(predictToUseLizardCauseTerrain(nextBlock) or predictToUseLizardCauseOpponent())) then
            //Periksa apakah myCar memiliki PowerUps Lizard
            //kemudian lakukan prediksi apakah Boost layak dipakai
            -> LIZARD
            else
                //Jika tidak pilih arah yang effektif
                -> getEffectiveDirection_Calculate()
            endif
        else if(not isThereObstacle(nextBlock)) then
            //Jika tidak terdapat obstacle pada nextBlock
```

```

//Strategi Memprediksi apakah musuh akan menggunakan EMP dan cara yang mungkin untuk
menghindarinya
if (havePowerUps(PowerUps.EMP, opponent.powerups) and
myCar.position.block > opponent.position.block and
myCar.position.block - opponent.position.block <= 2) then
-> dodgeEMP()
endif

//Strategi Penggunaan EMP dan Strategi untuk overtake opponent
if (opponent.position.block > myCar.position.block ) then
//Jika myCar memiliki PowerUps EMP,
//lakukan pemeriksaan apakah EMP layak digunakan
/* Kondisi layak penggunaan EMP :
    Posisi opponent berada pada 2 blok di depan kita dan masih di dalam map pada state yg sama.
    Namun terdapat kasus khusus, jika opponent berada di lane yg sama maka harus dipenuhi kondisi :
        posisi myCar di state selanjutnya tidak melebihi posisi opponent di state selanjutnya.
*/
if (havePowerUps(PowerUps.EMP, myCar.powerups) and
countEMP <- 0 and
opponent.position.block - myCar.position.block > 2 and
opponent.position.block - myCar.position.block <= 20) then
if (opponent.position.lane = myCar.position.lane) then
if (opponent.position.block > myCar.position.block + myCar.speed) then
    countEMP <- 1
    -> EMP
endif
endif
else
    countEMP <- 1
    -> EMP
endif
else
    //Strategi Menggunakan Lizard dan Strategi Perpindahan Lane
    countEMP <- 0
    if (havePowerUps(PowerUps.LIZARD, myCar.powerups) and
        (predictToUseLizardCauseTerrain(nextBlock) or predictToUseLizardCauseOpponent()) ) then
        -> LIZARD
    else
        -> getEffectiveDirection_Calculate()
    endif
endif
endif

//Strategi Penggunaan Tweet
/*Terdiri dari strategi pemilihan :
 * 1. posisi didepan opponent saat melakukan Fix
 * 2. posisi saat opponent berada di map sebelumnya atau di map selanjutnya atau
 * di map saat ini namun berada di belakang */
tweetPosition1 <- predictToUseTweet1(opponentFixPosition)
tweetPosition2 <- predictToUseTweet2()
if(tweetPosition1.block != 0 and tweetPosition1.lane != 0) then
-> TweetCommand(tweetPosition1.lane, tweetPosition1.block +opponent.speed + 4)
else
    opponentFixPosition <- opponent.position
endif

```

```

if(tweetPosition2.block != 0 and tweetPosition2.lane != 0) then
-> TweetCommand(tweetPosition2.lane, tweetPosition2.block)
endif
//Strategi Penggunaan OIL
if (havePowerUps(PowerUps.OIL, myCar.powerups) and predictToUseOil()) then
-> OIL
endif

-> ACCELERATE

```

```

// ADDED STRATEGY

/* Hitung terlebih dahulu jumlah damage yang diberikan jika melewati lane tersebut tanpa menggunakan lizard
 * periksa apakah damage myCar + damage >= 3 atau
 * block myCar + speed myCar > block myCar + blok lastObstacle
 * dengan lastObstacle adalah obstalce terakhir di lane pada saat pemeriksaan lane */
function predictToUseLizardCauseTerrain(List<Object> nextBlock) -> boolean

DEKLARASI
    damage : integer
BODY
    damage <- calculatePossibleDamageFromChoosenWay(nextBlock)
    if (myCar.damage + damage >= 3 or
        (myCar.position.block + myCar.speed >
        myCar.position.block + idxLastObstacle (nextBlock, lastObstacleInFront(nextBlock)))) then
            -> true
        endif
            -> false

/* Periksa apakah mungkin melangkahi opponent dengan menggunakan lizard
 * berada di posisi lane yang sama, dan
 * posisi block myCar + speed myCar > block opponet + speed opponent
 */
function predictToUseLizardCauseOpponent() -> boolean
DEKLARASI
    ...
BODY
    ...
    if (opponent.position.lane = myCar.position.lane and
        opponent.position.block > myCar.position.block and
        myCar.position.block + myCar.speed > opponent.position.block + opponent.speed) then
            -> true
        endif
            -> false

/* opponent berada pada jarak 1 s.d 5 di belakang
 * periksa lane yang mungkin akan dilewati opponent, bila pada lane-lane tersebut belum ada obstacle,
 * kembalikan true
 */
function predictToUseOil() -> boolean

```

DEKLARASI

RightBlocks, LeftBlocks : list of Object

BODY**if** (opponent.position.lane = myCar.position.lane and opponent.position.block < myCar.position.block)**then****if** (myCar.position.block - opponent.position.block <= 5) **then**

-> true

endif

else

if (myCar.position.lane = 1) **then**

RightBlocks <- getBlocksInFront(myCar.position.lane+1, myCar.position.block, myCar.speed)

if (isThereObstacle(RightBlocks)) **then**

-> true

endif**endif**else **if** (myCar.position.lane = 4) **then**

LeftBlocks <- getBlocksInFront(myCar.position.lane - 1, myCar.position.block, myCar.speed)

if (isThereObstacle(LeftBlocks)) **then**

-> true

endif**endif**

else

RightBlocks <- getBlocksInFront(myCar.position.lane + 1, myCar.position.block, myCar.speed)

LeftBlocks <- getBlocksInFront(myCar.position.lane - 1, myCar.position.block, myCar.speed)

if (isThereObstacle(RightBlocks) or isThereObstacle(LeftBlocks)) **then**

-> true

endif**endif**

-> false

/* Memeriksa apakah di suatu posisi, opponent sedang melakukan fix

* Akan mengembalikan posisi opponent bila sedang melakukan fix */

function predictToUseTweet1(Position fixPosition) -> position**DEKLARASI**

target : Position

BODY

target.block <- 0

target.lane <- 0

if (opponent.position.lane = fixPosition.lane andopponent.position.block = fixPosition.block) **then****if**(havePowerUps(PowerUps.TWEET, myCar.powerups)) **then**

target.block <- opponent.position.block

target.lane <- opponent.position.lane

else

target.block <- 0

target.lane <- 0

endif**endif**

-> target

// Fungsi dengan -> Position

/* Posisi yang akan dikembalikan :

* Kasus 1 : opponent berada pada jarak 2 s.d 5 di belakang

periksa lane yang mungkin akan dilewati opponent, bila pada lane-lane tersebut belum ada obstacle,

posisi target menjadi di lane tersebut dan blocknya +1 dari posisi block opponent
 * Kasus 2 : opponent berada pada jarak > 5 di belakang atau berada pada jarak >20 di depan
 posisi target menjadi tepat di depan opponent
 */

```

function predictToUseTweet2() -> Position
DEKLARASI
  target : Position
  i : integer
BODY
  target.block <- 0
  target.lane <- 0
  if (havePowerUps(PowerUps.TWEET, myCar.powerups)) then
    // Prediksi Target peletakan bila pada state tersebut opponent berada pada jangkauan myCar
    if (myCar.position.block - opponent.position.block <= 5 and myCar.position.block -
opponent.position.block > 1) then
      i <- 1
      while(i <= 4 and i != myCar.position.lane) do
        blocks <- getBlocksInFront(i, opponent.position.block, opponent.speed)
        if (not isThereObstacle(blocks)) then
          target.block <- opponent.position.block + 1
          target.lane <- opponent.position.lane
        else
          if (i = 1) then
            RightBlocks <- getBlocksInFront(i+1, opponent.position.block, opponent.speed-1)
            if (not isThereObstacle(RightBlocks)) then
              target.lane <- i+1
              target.block <- opponent.position.block + 1
            endif
          else if(i = 4) then
            LeftBlocks <- getBlocksInFront(i-1, opponent.position.block, opponent.speed-1)
            if (not isThereObstacle(LeftBlocks)) then
              target.lane <- i-1
              target.block <- opponent.position.block + 1
            endif
          else
            LeftBlocks <- getBlocksInFront(i-1, opponent.position.block, opponent.speed-1)
            RightBlocks <- getBlocksInFront(i+1, opponent.position.block, opponent.speed-1)
            if (not isThereObstacle(LeftBlocks) and not isThereObstacle(RightBlocks)) then
              target.lane <- i - 1
              target.block <- opponent.position.block + 1
            else if (not isThereObstacle(LeftBlocks) and isThereObstacle(RightBlocks)) then
              target.lane <- i - 1
              target.block <- opponent.position.block + 1
            else if (isThereObstacle(LeftBlocks) and not isThereObstacle(RightBlocks)) then
              target.lane <- i + 1
              target.block <- opponent.position.block + 1
            endif
          endif
          i <- i + 1
        endif
      endwhile

      else if (myCar.position.block - opponent.position.block > 5 or myCar.position.block -
opponent.position.block < 0) then
  
```

```

target.lane <- opponent.position.lane
target.block <- opponent.position.block + 1
endif
endif

-> target

/* Fungsi digunakan untuk mengecek apakah 15 blok di depan myCar
 * tidak memiliki jenis terrain yang merupakan obstacle */
function boolean predictToUseBoost() then
DEKLARASI
    blocks : list of Object
BODY
    blocks <- getBlocksInFront(myCar.position.lane, myCar.position.block, 15)
    if (not isThereObstacle(blocks)) then
        -> true
    endif
        -> false

function getEffectiveDirection_Calculate() -> Command
DEKLARASI
    direction : string
    damage_L, damage_C, damage_R : integer
    collectedPowerUps_L : integer
    collectedPowerUps_C : integer
    collectedPowerUps_R : integer
    CurrentBlocks, LeftBlocks, RightBlocks : list of Object
BODY
    direction <- "Stay"
    CurrentBlocks <- getBlocksInFront(myCar.position.lane , myCar.position.block, myCar.speed)

    damage_L <- 0, damage_R <- 0
    damage_C <- calculatePossibleDamageFromChoosenWay(CurrentBlocks)
    collectedPowerUps_C <- countPowerUpsFromChoosenWay(CurrentBlocks)

    if (myCar.position.lane = 1) then
        RightBlocks <- getBlocksInFront(myCar.position.lane + 1, myCar.position.block-1, myCar.speed)
        damage_R <- calculatePossibleDamageFromChoosenWay(RightBlocks)
        if (damage_C < damage_R) then
            direction <- "Stay"
        else if (damage_C > damage_R) then
            direction <- "Right"
        else
            collectedPowerUps_R <- countPowerUpsFromChoosenWay(RightBlocks)
            if (collectedPowerUps_C >= collectedPowerUps_R) then
                direction <- "Stay"
            else
                direction <- "Right"
            endif
        endif
    else if(myCar.position.lane = 4) then
        List<Object> LeftBlocks = getBlocksInFront(myCar.position.lane - 1, myCar.position.block-1,
myCar.speed)
        damage_L <- calculatePossibleDamageFromChoosenWay(LeftBlocks)
        if (damage_C < damage_L){

```

```

        direction <- "Stay"
else if (damage_C > damage_L){
    direction <- "Left"
else{
    collectedPowerUps_L = countPowerUpsFromChoosenWay(LeftBlocks)
    if (collectedPowerUps_C >= collectedPowerUps_L){
        direction <- "Stay"
    else{
        direction <- "Left"
    endif
endif
else
LeftBlocks <- getBlocksInFront(myCar.position.lane - 1, myCar.position.block - 1, myCar.speed)
RightBlocks <- getBlocksInFront(myCar.position.lane + 1, myCar.position.block - 1, myCar.speed)
damage_L <- calculatePossibleDamageFromChoosenWay(LeftBlocks)
damage_R <- calculatePossibleDamageFromChoosenWay(RightBlocks)
if (damage_C < damage_R) then
    if (damage_C > damage_L) then
        direction <- "Left"
    else if (damage_C < damage_L) then
        direction <- "Stay"
    else
        collectedPowerUps_L <- countPowerUpsFromChoosenWay(LeftBlocks)
        if (collectedPowerUps_C >= collectedPowerUps_L) then
            direction <- "Stay"
        else
            direction <- "Left"
        endif
    endif
else if (damage_C > damage_R) then
    if (damage_L < damage_R) then
        direction <- "Left"
    else if (damage_L > damage_R) then
        direction <- "Right"
    else
        collectedPowerUps_L <- countPowerUpsFromChoosenWay(LeftBlocks)
        collectedPowerUps_R <- countPowerUpsFromChoosenWay(RightBlocks)
        if (collectedPowerUps_L > collectedPowerUps_R) then
            direction <- "Left"
        else if (collectedPowerUps_L < collectedPowerUps_R) then
            direction <- "Right"
        else
            //Random, tapi untuk sementara pilih left
            direction <- "Left"
        endif
    endif
else
    if (damage_C > damage_L) then
        direction <- "Left"
    else if (damage_C <= damage_L) then
        direction <- "Stay"
    endif
endif

if (direction = "Stay") then

```

```

-> ACCELERATE
else if(direction = "Left")then
-> LEFT
else if(direction = "Right")then
-> RIGHT
endif
-> ACCELERATE

```

```

// FUNGSI TAMBAHAN

function dodgeEMP() -> Command
    // Periksa di lane mana opponent berada
    // Pindah ke lane yang aman :
    // 1. lane 1 atau lane 2, maka pindah ke kanan
    // 2. lane 3 atau lane 4, maka pindah ke kiri

function countPowerUpsFromChoosenWay(blocks : list of Object) -> integer
    // Mengembalikan banyak PowerUps yang akan diterima bila melewati kumpulan blocks yang dipilih

function calculatePossibleDamageFromChoosenWay(blocks : list of Object) -> integer
    // Mengembalikan damage yang akan diterima bila melewati kumpulan blocks yang dipilih

function havePowerUps(name : PowerUps, inCar : array of PowerUps) -> boolean
    // Memeriksa apakah pada suatu List PowerUps memiliki PowerUps yang diinginkan

function lastObstacleInFront(lane : list of Object) -> Object

function idxLastObstacle(lane : list of Object, nameTerrain : Object) -> integer

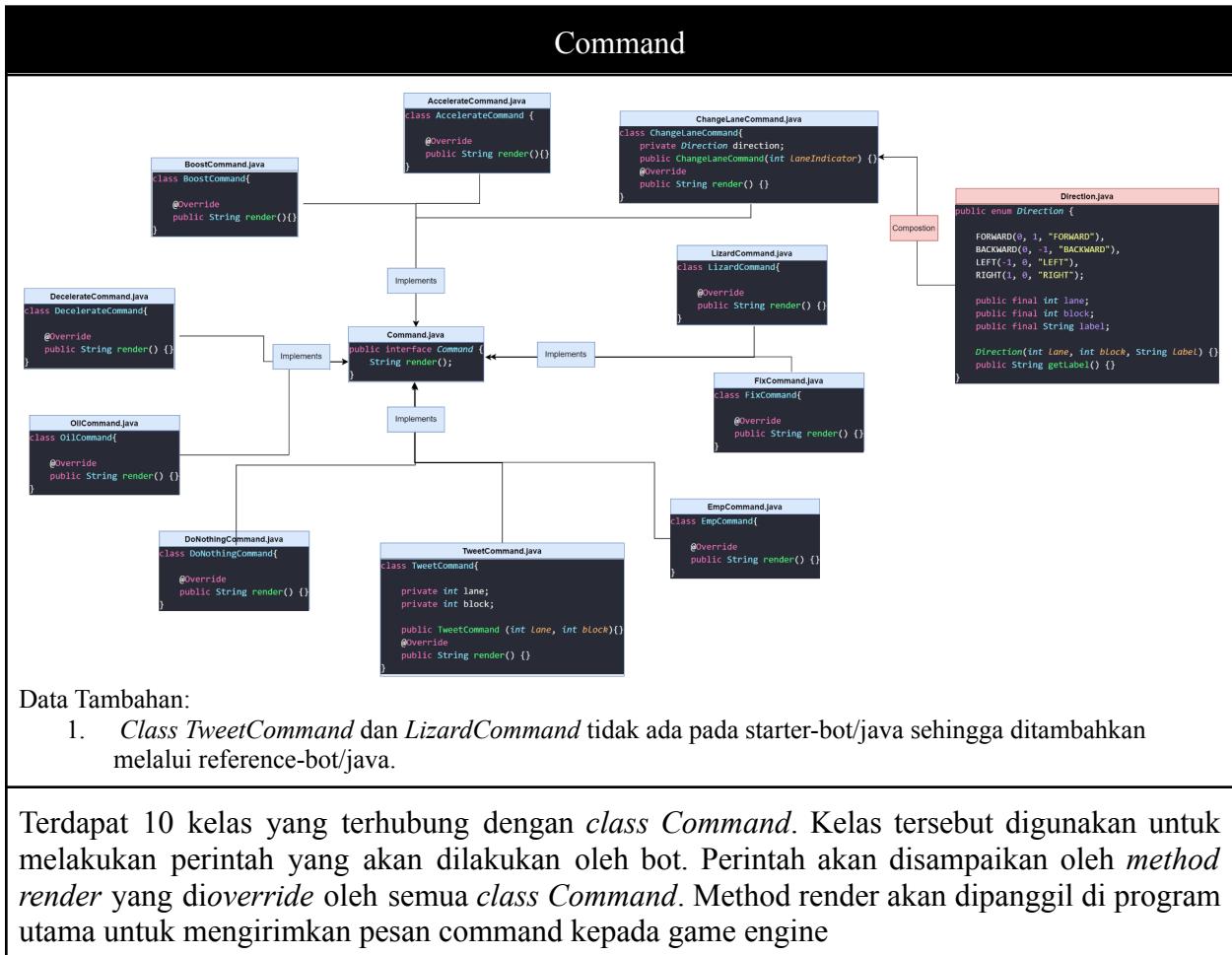
function isThereObstacle(blocks : list of Object) -> boolean
    // Digunakan untuk memeriksa apakah pada blocks memiliki salah satu jenis terrain berikut : WALL,
    MUD, OIL_SPILL */

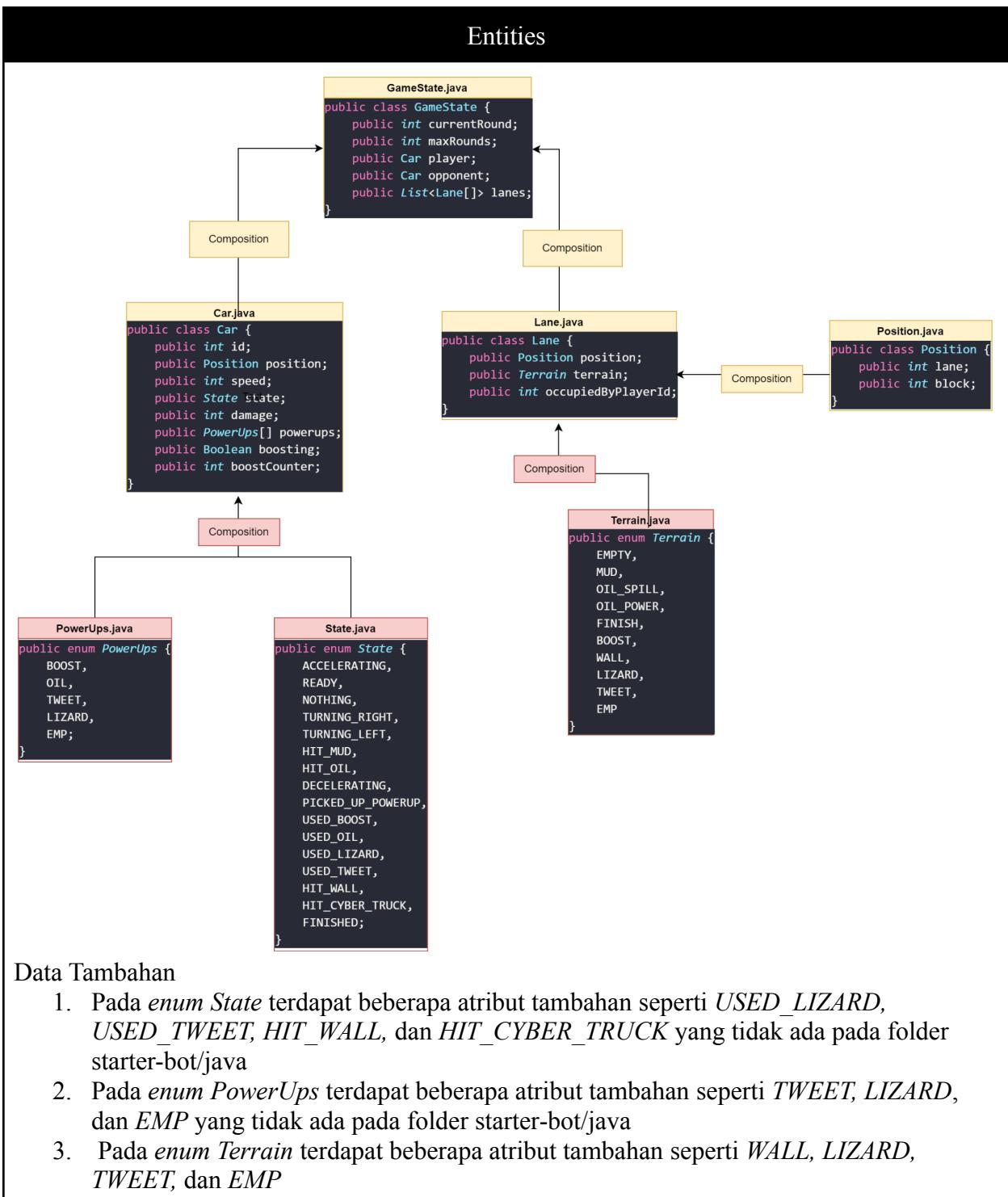
function isSafePosition() -> boolean
    // Digunakan untuk memeriksa apakah posisi myCar sekarang dalam posisi aman.
    /* Posisi aman berada pada :
     * Blok yang akan dilewati selanjutnya tidak memiliki bad terrain (oil spill, wall, mud) dan
     * (jarak opponent dan myCar > 20 dengan myCar berada lebih di depan atau
     * jarak opponent dan myCar antara 1 dan 20 serta opponent tidak memiliki EMP atau
     * jarak opponent dan myCar > 15 dengan opponent berada lebih di depan) */

function getBlocksInFront(lane : integer, block : integer, size : integer) -> list of Object
    // Mengambil block yang berada di depan posisi (lane,block) sebesar size

```

4.2 Struktur Data





Entites merupakan *class* yang berisikan objek yang akan dapat digunakan pada saat game dijalankan. *Class* utama adalah *GameState* karena akan berisi *state-state* yang akan dipakai di tiap *round-nya*.

4.3 Analisis dan Pengujian

4.3.1 Tweet V1

Pengujian

myCar

Round 064
[Reset](#)
[Remove this match](#)

[754, 1]	[751, 1]	[748, 1]	[745, 1]	[742, 1]	[741, 1]	[740, 1]	[739, 1]	[738, 1]	[737, 1]	[736, 1]	[735, 1]	[734, 1]	[733, 1]	[732, 1]	[731, 1]	[730, 1]	[729, 1]	[728, 1]	[727, 1]	[726, 1]	[725, 1]	[724, 1]	[723, 1]	[722, 1]	[721, 1]	[720, 1]	[719, 1]	[718, 1]	[717, 1]	[716, 1]	[715, 1]	[714, 1]	[713, 1]	[712, 1]	[711, 1]	[710, 1]	[709, 1]	[708, 1]	[707, 1]	[706, 1]	[705, 1]	[704, 1]	[703, 1]	[702, 1]	[701, 1]	[700, 1]	[699, 1]	[698, 1]	[697, 1]	[696, 1]	[695, 1]	[694, 1]	[693, 1]	[692, 1]	[691, 1]	[690, 1]	[689, 1]	[688, 1]	[687, 1]	[686, 1]	[685, 1]	[684, 1]	[683, 1]	[682, 1]	[681, 1]	[680, 1]	[679, 1]	[678, 1]	[677, 1]	[676, 1]	[675, 1]	[674, 1]	[673, 1]	[672, 1]	[671, 1]	[670, 1]	[669, 1]	[668, 1]	[667, 1]	[666, 1]	[665, 1]	[664, 1]	[663, 1]	[662, 1]	[661, 1]	[660, 1]	[659, 1]	[658, 1]	[657, 1]	[656, 1]	[655, 1]	[654, 1]	[653, 1]	[652, 1]	[651, 1]	[650, 1]	[649, 1]	[648, 1]	[647, 1]	[646, 1]	[645, 1]	[644, 1]	[643, 1]	[642, 1]	[641, 1]	[640, 1]	[639, 1]	[638, 1]	[637, 1]	[636, 1]	[635, 1]	[634, 1]	[633, 1]	[632, 1]	[631, 1]	[630, 1]	[629, 1]	[628, 1]	[627, 1]	[626, 1]	[625, 1]	[624, 1]	[623, 1]	[622, 1]	[621, 1]	[620, 1]	[619, 1]	[618, 1]	[617, 1]	[616, 1]	[615, 1]	[614, 1]	[613, 1]	[612, 1]	[611, 1]	[610, 1]	[609, 1]	[608, 1]	[607, 1]	[606, 1]	[605, 1]	[604, 1]	[603, 1]	[602, 1]	[601, 1]	[600, 1]	[599, 1]	[598, 1]	[597, 1]	[596, 1]	[595, 1]	[594, 1]	[593, 1]	[592, 1]	[591, 1]	[590, 1]	[589, 1]	[588, 1]	[587, 1]	[586, 1]	[585, 1]	[584, 1]	[583, 1]	[582, 1]	[581, 1]	[580, 1]	[579, 1]	[578, 1]	[577, 1]	[576, 1]	[575, 1]	[574, 1]	[573, 1]	[572, 1]	[571, 1]	[570, 1]	[569, 1]	[568, 1]	[567, 1]	[566, 1]	[565, 1]	[564, 1]	[563, 1]	[562, 1]	[561, 1]	[560, 1]	[559, 1]	[558, 1]	[557, 1]	[556, 1]	[555, 1]	[554, 1]	[553, 1]	[552, 1]	[551, 1]	[550, 1]	[549, 1]	[548, 1]	[547, 1]	[546, 1]	[545, 1]	[544, 1]	[543, 1]	[542, 1]	[541, 1]	[540, 1]	[539, 1]	[538, 1]	[537, 1]	[536, 1]	[535, 1]	[534, 1]	[533, 1]	[532, 1]	[531, 1]	[530, 1]	[529, 1]	[528, 1]	[527, 1]	[526, 1]	[525, 1]	[524, 1]	[523, 1]	[522, 1]	[521, 1]	[520, 1]	[519, 1]	[518, 1]	[517, 1]	[516, 1]	[515, 1]	[514, 1]	[513, 1]	[512, 1]	[511, 1]	[510, 1]	[509, 1]	[508, 1]	[507, 1]	[506, 1]	[505, 1]	[504, 1]	[503, 1]	[502, 1]	[501, 1]	[500, 1]	[499, 1]	[498, 1]	[497, 1]	[496, 1]	[495, 1]	[494, 1]	[493, 1]	[492, 1]	[491, 1]	[490, 1]	[489, 1]	[488, 1]	[487, 1]	[486, 1]	[485, 1]	[484, 1]	[483, 1]	[482, 1]	[481, 1]	[480, 1]	[479, 1]	[478, 1]	[477, 1]	[476, 1]	[475, 1]	[474, 1]	[473, 1]	[472, 1]	[471, 1]	[470, 1]	[469, 1]	[468, 1]	[467, 1]	[466, 1]	[465, 1]	[464, 1]	[463, 1]	[462, 1]	[461, 1]	[460, 1]	[459, 1]	[458, 1]	[457, 1]	[456, 1]	[455, 1]	[454, 1]	[453, 1]	[452, 1]	[451, 1]	[450, 1]	[449, 1]	[448, 1]	[447, 1]	[446, 1]	[445, 1]	[444, 1]	[443, 1]	[442, 1]	[441, 1]	[440, 1]	[439, 1]	[438, 1]	[437, 1]	[436, 1]	[435, 1]	[434, 1]	[433, 1]	[432, 1]	[431, 1]	[430, 1]	[429, 1]	[428, 1]	[427, 1]	[426, 1]	[425, 1]	[424, 1]	[423, 1]	[422, 1]	[421, 1]	[420, 1]	[419, 1]	[418, 1]	[417, 1]	[416, 1]	[415, 1]	[414, 1]	[413, 1]	[412, 1]	[411, 1]	[410, 1]	[409, 1]	[408, 1]	[407, 1]	[406, 1]	[405, 1]	[404, 1]	[403, 1]	[402, 1]	[401, 1]	[400, 1]	[399, 1]	[398, 1]	[397, 1]	[396, 1]	[395, 1]	[394, 1]	[393, 1]	[392, 1]	[391, 1]	[390, 1]	[389, 1]	[388, 1]	[387, 1]	[386, 1]	[385, 1]	[384, 1]	[383, 1]	[382, 1]	[381, 1]	[380, 1]	[379, 1]	[378, 1]	[377, 1]	[376, 1]	[375, 1]	[374, 1]	[373, 1]	[372, 1]	[371, 1]	[370, 1]	[369, 1]	[368, 1]	[367, 1]	[366, 1]	[365, 1]	[364, 1]	[363, 1]	[362, 1]	[361, 1]	[360, 1]	[359, 1]	[358, 1]	[357, 1]	[356, 1]	[355, 1]	[354, 1]	[353, 1]	[352, 1]	[351, 1]	[350, 1]	[349, 1]	[348, 1]	[347, 1]	[346, 1]	[345, 1]	[344, 1]	[343, 1]	[342, 1]	[341, 1]	[340, 1]	[339, 1]	[338, 1]	[337, 1]	[336, 1]	[335, 1]	[334, 1]	[333, 1]	[332, 1]	[331, 1]	[330, 1]	[329, 1]	[328, 1]	[327, 1]	[326, 1]	[325, 1]	[324, 1]	[323, 1]	[322, 1]	[321, 1]	[320, 1]	[319, 1]	[318, 1]	[317, 1]	[316, 1]	[315, 1]	[314, 1]	[313, 1]	[312, 1]	[311, 1]	[310, 1]	[309, 1]	[308, 1]	[307, 1]	[306, 1]	[305, 1]	[304, 1]	[303, 1]	[302, 1]	[301, 1]	[300, 1]	[299, 1]	[298, 1]	[297, 1]	[296, 1]	[295, 1]	[294, 1]	[293, 1]	[292, 1]	[291, 1]	[290, 1]	[289, 1]	[288, 1]	[287, 1]	[286, 1]	[285, 1]	[284, 1]	[283, 1]	[282, 1]	[281, 1]	[280, 1]	[279, 1]	[278, 1]	[277, 1]	[276, 1]	[275, 1]	[274, 1]	[273, 1]	[272, 1]	[271, 1]	[270, 1]	[269, 1]	[268, 1]	[267, 1]	[266, 1]	[265, 1]	[264, 1]	[263, 1]	[262, 1]	[261, 1]	[260, 1]	[259, 1]	[258, 1]	[257, 1]	[256, 1]	[255, 1]	[254, 1]	[253, 1]	[252, 1]	[251, 1]	[250, 1]	[249, 1]	[248, 1]	[247, 1]	[246, 1]	[245, 1]	[244, 1]	[243, 1]	[242, 1]	[241, 1]	[240, 1]	[239, 1]	[238, 1]	[237, 1]	[236, 1]	[235, 1]	[234, 1]	[233, 1]	[232, 1]	[231, 1]	[230, 1]	[229, 1]	[228, 1]	[227, 1]	[226, 1]	[225, 1]	[224, 1]	[223, 1]	[222, 1]	[221, 1]	[220, 1]	[219, 1]	[218, 1]	[217, 1]	[216, 1]	[215, 1]	[214, 1]	[213, 1]	[212, 1]	[211, 1]	[210, 1]	[209, 1]	[208, 1]	[207, 1]	[206, 1]	[205, 1]	[204, 1]	[203, 1]	[202, 1]	[201, 1]	[200, 1]	[199, 1]	[198, 1]	[197, 1]	[196, 1]	[195, 1]	[194, 1]	[193, 1]	[192, 1]	[191, 1]	[190, 1]	[189, 1]	[188, 1]	[187, 1]	[186, 1]	[185, 1]	[184, 1]	[183, 1]	[182, 1]	[181, 1]	[180, 1]	[179, 1]	[178, 1]	[177, 1]	[176, 1]	[175, 1]	[174, 1]	[173, 1]	[172, 1]	[171, 1]	[170, 1]	[169, 1]	[168, 1]	[167, 1]	[166, 1]	[165, 1]	[164, 1]	[163, 1]	[162, 1]	[161, 1]	[160, 1]	[159, 1]	[158, 1]	[157, 1]	[156, 1]	[155, 1]	[154, 1]	[153, 1]	[152, 1]	[151, 1]	[150, 1]	[149, 1]	[148, 1]	[147, 1]	[146, 1]	[145, 1]	[144, 1]	[143, 1]	[142, 1]	[141, 1]	[140, 1]	[139, 1]	[138, 1]	[137, 1]	[136, 1]	[135, 1]	[134, 1]	[133, 1]	[132, 1]	[131, 1]	[130, 1]	[129, 1]	[128, 1]	[127, 1]	[126, 1]	[125, 1]	[124, 1]	[123, 1]	[122, 1]	[121, 1]	[120, 1]	[119, 1]	[118, 1]	[117, 1]	[116, 1]	[115, 1]	[114, 1]	[113, 1]	[112, 1]	[111, 1]	[110, 1]	[109, 1]	[108, 1]	[107, 1]	[106, 1]	[105, 1]	[104, 1]	[103, 1]	[102, 1]	[101, 1]	[100, 1]	[99, 1]	[98, 1]	[97, 1]	[96, 1]	[95, 1]	[94, 1]	[93, 1]	[92, 1]	[91, 1]	[90, 1]	[89, 1]	[88, 1]	[87, 1]	[86, 1]	[85, 1]	[84, 1]	[83, 1]	[82, 1]	[81, 1]	[80, 1]	[79, 1]	[78, 1]	[77, 1]	[76, 1]	[75, 1]	[74, 1]	[73, 1]	[72, 1]	[71, 1]	[70, 1]	[69, 1]	[68, 1]	[67, 1]	[66, 1]	[65, 1]	[64, 1]	[63, 1]	[62, 1]	[61, 1]	[60, 1]	[59, 1]	[58, 1]	[57, 1]	[56, 1]	[55, 1]	[54, 1]	[53, 1]	[52, 1]	[51, 1]	[50, 1]	[49, 1]	[48, 1]	[47, 1]	[46, 1]	[45, 1]	[44, 1]	[43, 1]	[42, 1]	[41, 1]	[40, 1]	[39, 1]	[38, 1]	[37, 1]	[36, 1]	[35, 1]	[34, 1]	[33, 1]	[32, 1]	[31, 1]	[30, 1]	[29, 1]	[28, 1]	[27, 1]	[26, 1]	[25, 1]	[24, 1]	[23, 1]	[22, 1]	[21, 1]	[20, 1]	[19, 1]	[18, 1]	[17, 1]	[16, 1]	[15, 1]	[14, 1]	[13, 1]	[12, 1]	[11, 1]	[10, 1]	[9, 1]	[8, 1]	[7, 1]	[6, 1]	[5, 1]	[4, 1]	[3, 1]	[2, 1]	[1, 1]	[0, 1]
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

[< First] < Prev 64 Next > [Last >]

(Click the button that displays the round number to quickly switch rounds)

Round Details
Max Rounds: 600
Current Round: 64

A - Game-Haram

Position
Speed
Lane
Distance
Boosts
Boosting
Powerups

1
15
4
741
3
YES
EMP,EMP,EMP,EMP,
EMP,BOOST,LIZARD,LIZARD,EMP,OIL,TWEET

Bot Command

Command: USE_TWEET 1509

Opponent Car

Round Details

Max Rounds: 600

A - Game-Haram

B - GameHalal

Position	Speed	Lane	Distance	Boosts
2 [x: 502, y: 1]	3	1	502	0

Bot Command
Command: FIX

Round Details

Max Rounds: 600

A - Game-Haram

B - GameHalal

Position	Speed	Lane	Distance	Boosts
2	3	1	502	0

Bot Command
Command: USE_BOOST

Analisis

Tweet v1 digunakan pada saat mobil lawan melakukan fix, lalu ronde selanjutnya myCar akan menggunakan Command Tweet. Seperti pada gambar di atas, opponent(merah) melakukan fix pada 64 dan pada round 65 opponent terkena Cyber_Truck.

Strategi ini tidak efektif apabila Tweet dijalankan berdekatan dengan obstacle lain, sehingga opponent mungkin akan melakukan turning.

4.3.2 Tweet V2

Pengujian

myCar

Round 088

[Click the button that displays the round number to quickly switch rounds]

Round Details

Max Rounds: 600

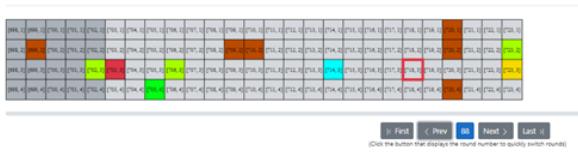
A - Game-Haram

Position	Speed	Lane	Distance	Boosts
1 [x: 1006, y: 4]	9	4	1006	0

Bot Command

Command: USE_TWEET 3 722
Execution time: 2ms
Exception: null

Opponent Car



Round Details

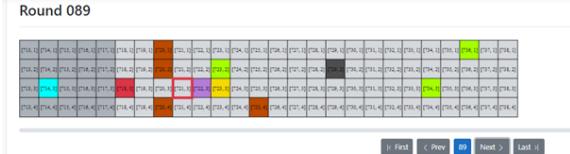
Max Rounds: 600

A - Game-Haram

B - GameHalal

Position	Speed	Lane	Distance	Boosts
2 [x: 703, y: 3]	15	3	703	4

Bot Command



Round Details

Max Rounds: 600

A - Game-Haram

B - GameHalal

Position	Speed	Lane	Distance	Boosts
2 [x: 718, y: 3]	15	3	718	3

Bot Command

Command: ACCELERATE

Analisis

Tweet v2 digunakan pada saat mobil lawan sudah jauh di depan atau di belakang. Cyber Truck akan muncul pada depan mobil lawan yang sudah berjarak jauh dari myCar. Seperti pada gambar myCar telah berada pada block 1006, sedangkan opponent berada pada block 703.

Strategi ini tidak efektif apabila Tweet dijalankan pada opponent yang telah jauh di depan karena lane musuh akan sulit diprediksi dan jika opponent sudah terlalu jauh, maka hanya akan mempersempit jarak sedikit saja.

4.3.3 Fix

Pengujian

myCar

Round 097

(Click the button that displays the round number to quickly switch rounds)

Round Details

Max Rounds: 600

A - Game-Haram

Position	Speed	Lane	Distance	Boosts
1 [x: 1063, y: 1]	8	1	1063	0

Bot Command

Command: FIX

Analisis

Fix digunakan pada saat aman. Dikatakan aman ketika

- Blok yang akan dilewati selanjutnya tidak memiliki bad terrain (oil spill, wall, mud) dan
- (jarak opponent dan myCar > 20 dengan myCar berada lebih di depan atau
- jarak opponent dan myCar antara 1 dan 20 serta opponent tidak memiliki EMP atau
- jarak opponent dan myCar > 15 dengan opponent berada lebih di depan)

Pada gambar diatas myCar telah berjarak jauh didepan dan pada block selanjutnya tidak ada obstacle

Strategi ini tidak efektif apabila opponent memiliki EMP dan berada dekat di belakang myCar, sehingga myCar akan menumpuk-numpuk damage.

4.3.4 Turning

Pengujian

myCar

Round 096

[First] [Prev] **96** [Next] [Last]

(Click the button that displays the round number to quickly switch rounds)

Round Details

Max Rounds: 600

A - Game-Haram

Position 1 [x: 1056, y: 2]	Speed 8	Lane 2	Distance 1056	Boosts 0

Bot Command

Command: TURN_LEFT

Analisis

Turning digunakan pada saat ada obstacle di depan. Penggunaan turning dengan memilih lane yang akan memberikan damage paling sedikit atau memberikan powerUps terbanyak. Pada gambar lane didepan myCar memiliki mud, dan dikiri myCar memiliki mud, sehingga akan diambil sebelah kiri yang paling sedikit damage.

Strategi ini tidak efektif apabila Opponent dekat pada myCar dan mengambil jalur yang sama dan memungkinkan adanya tabrakan.

4.3.5 Lizard

Pengujian

myCar

Round 067

[775, 1]	[776, 1]	[777, 1]	[778, 1] <small><MCID></small>	[779, 1]	[780, 1]	[781, 1]	[782, 1]	[783, 1]	[784, 1]	[785, 1] <small><MCID></small>	[786, 1]	[787, 1]	[788, 1]	[789, 1]	[790, 1]	[791, 1]	[792, 1]
[775, 2]	[776, 2] <small>HOTEL></small>	[777, 2]	[778, 2]	[779, 2]	[780, 2]	[781, 2]	[782, 2]	[783, 2]	[784, 2] <small><MCID></small>	[785, 2]	[786, 2]	[787, 2]	[788, 2]	[789, 2]	[790, 2]	[791, 2]	[792, 2]
[775, 3]	[776, 3]	[777, 3]	[778, 3]	[779, 3]	[780, 3]	[781, 3]	[782, 3]	[783, 3]	[784, 3] <small><MCID></small>	[785, 3]	[786, 3]	[787, 3]	[788, 3]	[789, 3]	[790, 3]	[791, 3]	[792, 3]
[775, 4]	[776, 4] <small><MCID></small>	[777, 4]	[778, 4]	[779, 4] <small><MCID></small>	[780, 4]	[781, 4]	[782, 4]	[783, 4]	[784, 4] <small><MCID></small>	[785, 4]	[786, 4]	[787, 4]	[788, 4]	[789, 4]	[790, 4]	[791, 4] <small><MCID></small>	[792, 4]

[First](#) [Prev](#) **67** [Next](#) [Last](#)

(Click the button that displays the round number to quickly switch rounds)

Round Details

Max Rounds: 600

A - Game-Haram



Position
1
[x: 780, y: 4]



Speed
9



Lane
4



Distance
780



Boosts
0

Bot Command

Command: USE_LIZARD

Analisis

Lizard digunakan saat ada *obstacle* atau opponent di depan dan myCar memiliki lizard untuk melangkahi *obstacle* tersebut. Seperti pada gambar myCar menggunakan Lizard karena memiliki Lizard dan ada *obstacle* di depan maupun dikiri.

Strategi ini tidak efektif terdapat kemungkinan bahwa mobil lawan teradapat pada blok yang sama dengan mobil saat telah menggunakan skill ini sehingga akan terjadi tabrakan antarmobil.

31

4.3.6 Boost

Pengujian

myCar

Round 021

[221, 1]	[222, 1]	[223, 1]	[224, 1]	[225, 1]	[226, 1] PLAYERS	[227, 1]	[228, 1]	[229, 1] BOOSTS	[230, 1]	[231, 1]	[232, 1]	[233, 1]	[234, 1]	[235, 1]	[236, 1]	[237, 1]	[238, 1]	[239, 1]	[240, 1]	[241, 1]	[242, 1]
[221, 2]	[222, 2]	[223, 2]	[224, 2] <OBSTACLE>	[225, 2] <OBSTACLE>	[226, 2]	[227, 2]	[228, 2] <OBSTACLE>	[229, 2]	[230, 2]	[231, 2]	[232, 2]	[233, 2]	[234, 2]	[235, 2]	[236, 2]	[237, 2]	[238, 2]	[239, 2]	[240, 2]	[241, 2]	[242, 2]
[221, 3]	[222, 3]	[223, 3]	[224, 3]	[225, 3] <OBSTACLE>	[226, 3]	[227, 3]	[228, 3]	[229, 3]	[230, 3]	[231, 3]	[232, 3]	[233, 3]	[234, 3]	[235, 3]	[236, 3]	[237, 3]	[238, 3]	[239, 3]	[240, 3]	[241, 3]	[242, 3]
[221, 4]	[222, 4]	[223, 4]	[224, 4]	[225, 4]	[226, 4]	[227, 4]	[228, 4]	[229, 4]	[230, 4]	[231, 4]	[232, 4]	[233, 4]	[234, 4]	[235, 4]	[236, 4]	[237, 4]	[238, 4]	[239, 4]	[240, 4]	[241, 4]	[242, 4]

[< First] [< Prev] [21] [Next >] [Last >]
(Click the button that displays the round number to quickly switch rounds)

Round Details

Max Rounds: 600

A - Game-Haram

					
Position 1 [x: 226, y: 1]	Speed 8	Lane 1	Distance 226	Boosts 0	Boosting No

Bot Command

Command: USE_BOOST

Analisis

Boost digunakan pada saat lane di depan myCar tidak memiliki *obstacle* dan memiliki *boost-counting* = 0. Seperti pada gambar lane di depan myCar tidak memiliki *obstacle* sehingga *boost* pada dapat dijalankan.

Strategi ini tidak efektif pada *round-round* selanjutnya terdapat *obstacle* yang tidak dapat dihindari sehingga menabrak *obstacle* sehingga memiliki damage yang lebih besar.

4.3.7 Oil

Pengujian

myCar

Round 043

Reset
Remove this match

(481, 1)	(484, 1)	(481, 1)	(484, 1)	(487, 1)	(486, 1)	(488, 1)	(485, 1)	(489, 1)	(482, 1)	(483, 1)	(480, 1)	(486, 1)	(487, 1)	(488, 1)	(489, 1)	(480, 1)	(481, 1)	(482, 1)	(483, 1)	(484, 1)	(485, 1)	(486, 1)	(487, 1)	(488, 1)	(489, 1)
(483, 2)	(484, 2)	(485, 2)	(486, 2)	(487, 2)	(488, 2)	(489, 2)	(480, 2)	(481, 2)	(482, 2)	(483, 2)	(484, 2)	(485, 2)	(486, 2)	(487, 2)	(488, 2)	(489, 2)	(481, 2)	(482, 2)	(483, 2)	(484, 2)	(485, 2)	(486, 2)	(487, 2)	(488, 2)	(489, 2)
(483, 3)	(484, 3)	(485, 3)	(486, 3)	(487, 3)	(488, 3)	(489, 3)	(480, 3)	(481, 3)	(482, 3)	(483, 3)	(484, 3)	(485, 3)	(486, 3)	(487, 3)	(488, 3)	(489, 3)	(481, 3)	(482, 3)	(483, 3)	(484, 3)	(485, 3)	(486, 3)	(487, 3)	(488, 3)	(489, 3)
(483, 4)	(484, 4)	(485, 4)	(486, 4)	(487, 4)	(488, 4)	(489, 4)	(480, 4)	(481, 4)	(482, 4)	(483, 4)	(484, 4)	(485, 4)	(486, 4)	(487, 4)	(488, 4)	(489, 4)	(481, 4)	(482, 4)	(483, 4)	(484, 4)	(485, 4)	(486, 4)	(487, 4)	(488, 4)	(489, 4)

< First < Prev 43 Next > Last >

(Click the button that displays the round number to quickly switch round)

Round Details

Max Rounds: 600

Current Round: 43

A - Game-Haram

(selected) ✓

Position	Speed	Lane	Distance	Boosts	Boosting	Powerups
1	15	1	488	1	YES	EMPOIL,EMP,EMP,EMP,OIL

Bot Command

Command: USE_OIL

B - GameHalal

(x: 468, y: 1)

Max Rounds: 600

Position: 2, Speed: 9, Lane: 2, Distance: 473, Boosts: 0, Boosting: No

Bot Command

Command: ACCELERATE

Analysis

33

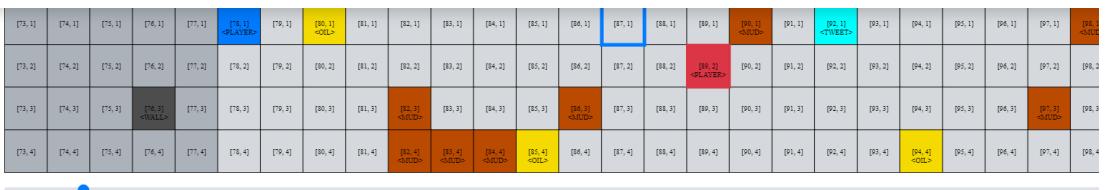
Oil akan dilepaskan saat opponent tepat berada kurang dari 5 dibelakang myCar atau memprediksi dengan *obstacle* yang ada seperti pada gambar. Terdapat *obstacle* pada lane 2, 3, 4 sehingga Oil akan dilepas pada lane 1.

Strategi ini tidak efektif apabila mobil musuh berada di belakang dengan jarak lebih dari 5 sehingga musuh mungkin dapat menghindari *oil_spill* yang telah dibuat.

4.3.8 EMP

Pengujian

myCar & opponent



(Click the button that displays the round number to quickly switch rounds)

Round Details																				Current Round:	
																				(selected) ✓	
Position		Speed		Lane		Distance		Boosts		Boosting		Powerups									
2		9		1		78		0		No		LIZARD,LIZARD,EM P,LIZARD,TWEET,T WEET									
Bot Command																					
Command: USE_EMP																					

Analisis

EMP digunakan pada saat opponent berada di depan dan memiliki jarak lebih dari sama dengan 2, sehingga kemungkinan opponent terkena EMP lebih besar karena EMP berbentuk *cone* pada saat ditembak. Seperti pada gambar, opponent berada di depan berjarak 2 sehingga lawan akan ditembak dengan EMP

Strategi ini tidak efektif myCar pada lane 1 atau 4 dan lawan berada di lane 2 atau 3 yang berjarak 2 didepan, sehingga lawan mungkin dapat menghindar.

Bab V Kesimpulan dan Saran

5.1 Kesimpulan

Dari Tugas Besar 1 IF 2211 Strategi Algorimta Semester II T.A. 2020/2021 yang berjudul “Pemanfaatan Algoritma Greedy dalam Aplikasi Permainan “Overdrive”, Tim Game Haram Berhasil membuat Bot untuk mengendalikan mobil dengan menggunakan strategi Greedy. Strategi Greedy yang dimanfaatkan juga bukan hanya satu atau dua strategi, melainkan banyak strategi yang disusun sedemikian rupa sehingga berhasil menciptakan strategi yang menurut tim kami adalah strategi terbaik bagi kami.

5.2 Saran

Adapun beberapa saran yang dapat bermanfaat sebagai acuan dan pengembangan tugas maupun strategi greedy adalah sebagai berikut

- Akan lebih baik bila dibuat terlebih dahulu skema prioritas strategi yang lebih rapi sehingga untuk merevisi strateginya dapat dilakukan dengan lebih mudah
- Pembuatan pseudocode akan lebih baik bila ditulis dengan menggunakan fungsi. Dalam hal ini dapat memperindah dan memperjelas pseudocode dan dapat digunakan untuk merevisi dan merapikan code program.

Daftar Pustaka

Entellect Challange. (2020, Agustus 17). EntelectChallenge/2020-Overdrive .
<https://github.com/EntelectChallenge/2020-Overdrive> (diakses tanggal 5 Agustus 2022).

Van den Berg Franché. (2020, April 12). <https://entelect-replay.raezor.co.za/>
[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf).

Lampiran

Link Github : https://github.com/ghebyon/TUBES01_IF2211

Link Demo : https://youtu.be/YgRG_JmRejM