

Laporan Tugas Besar I IF2211 Strategi Algoritma

Pemanfaatan Algoritma Greedy dalam Aplikasi

Permainan Overdrive



Disusun Oleh:
Kelompok 3 Heavymetal
Muhammad Akmal Arifin - 13520037
Azmi Alfatih Shalahuddin - 13520158
Daffa Romyz Aufa - 13520162

Institut Teknologi Bandung

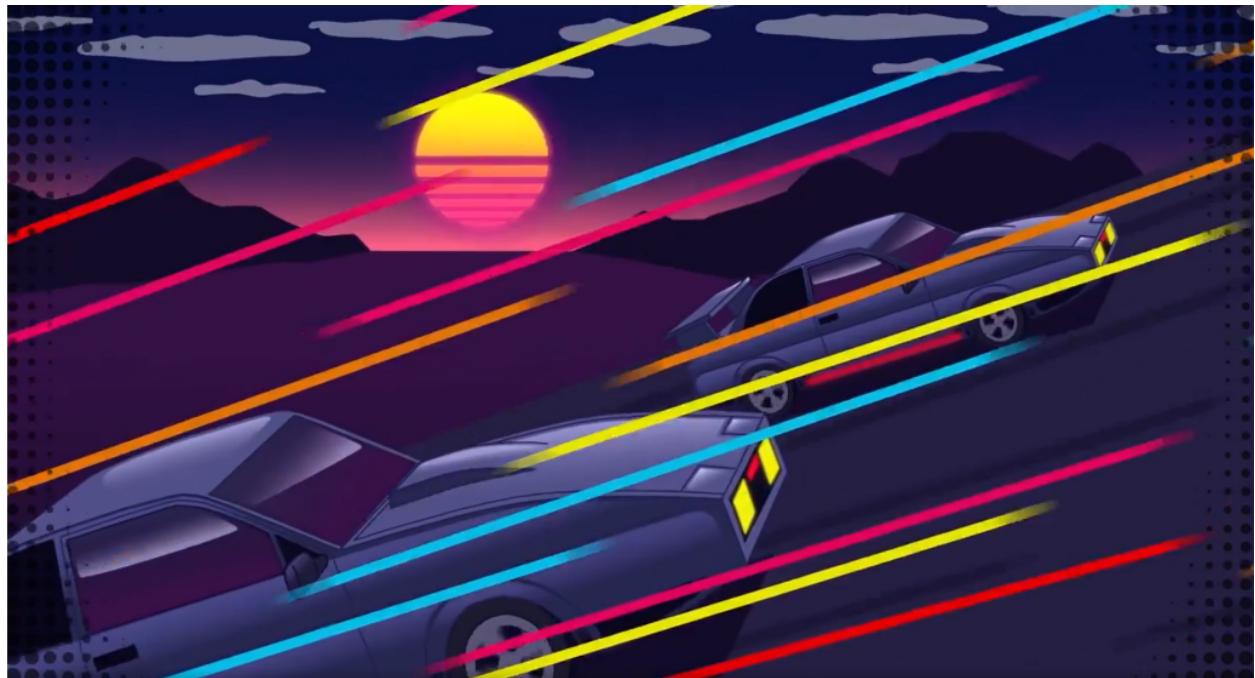
2022

DAFTAR ISI

Bab 1 Deskripsi Tugas	2
Bab 2 Landasan Teori	3
Algoritma Greedy	3
Aturan Umum Game Engine Overdrive	4
Implementasi Algoritma Greedy ke dalam Bot	5
Pengaturan Game Engine	7
Bab 3 Aplikasi Strategi Greedy	8
Persoalan Overdrive	8
Eksplorasi Alternatif Solusi Greedy	8
Strategi Greedy yang Dipilih	9
Bab 4 Implementasi dan Pengujian	12
Implementasi Algoritma Greedy dalam Game Engine	12
Struktur Data	13
Analisis Desain Solusi	14
Bab 5 Kesimpulan dan Saran	21
Kesimpulan	21
Saran	21
DAFTAR PUSTAKA	22
LAMPIRAN	22
Link Github	22

Bab 1 Deskripsi Tugas

Overdrive adalah sebuah game yang mempertandingkan 2 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 1.1. Contoh Ilustrasi Overdrive

Pada tugas besar pertama Strategi Algoritma ini, gunakanlah sebuah *game engine* yang mengimplementasikan permainan *Overdrive*. *Game engine* dapat diperoleh pada laman berikut: [EntelectChallenge/2020-Overdrive: Main repository for Entelect Challenge 2020](https://github.com/EntelectChallenge/2020-Overdrive)

Tugas mahasiswa adalah 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/EntelectChallenge/2020-Overdrive/releases/tag/2020.3.4>

Bab 2 Landasan Teori

A. Algoritma Greedy

Algoritma *greedy* merupakan salah satu algoritma atau metode untuk menyelesaikan sebuah permasalahan secara algoritmik. Algoritma ini merupakan metode yang paling populer dan sederhana dalam memecahkan persoalan optimasi, yaitu persoalan untuk mencari solusi paling optimal dari suatu permasalahan. Langkah-langkah algoritma *greedy* mencerminkan dari kata *greedy* yang dalam Bahasa Inggris artinya rakus atau tamak. Pada setiap langkah yang kita pilih, terdapat banyak pilihan yang perlu untuk dievaluasi dan kita perlu memilih langkah yang terbaik dari setiap langkah yang ada. Prinsip *greedy* adalah “*take what you can get now!*” yaitu pada setiap langkah, algoritma memilih pilihan optimal lokal. Dan dari setiap langkah tersebut, diharapkan akan membawa ke solusi yang optimal.

Elemen-elemen algoritma *greedy*:

1. Himpunan kandidat: berisi kandidat yang akan dipilih pada setiap langkah
2. Himpunan solusi: berisi kandidat yang sudah dipilih
3. Fungsi solusi: menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi
4. Fungsi seleksi: memilih kandidat berdasarkan strategi *greedy* tertentu. Strategi *greedy* ini bersifat heuristik.
5. Fungsi kelayakan: memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi (layak atau tidak)
6. Fungsi obyektif: memaksimumkan atau meminimumkan

Skema umum algoritma *greedy*:

```
function greedy(C : himpunan_kandidat) → himpunan_solusi
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma
greedy }
```

Deklarasi

```
x : kandidat
S : himpunan_solusi
```

Algoritma:

```
S ← {}
while (not SOLUSI(S)) and (C ≠ {}) do
    X ← SELEKSI(C)
    C ← C - {x}
    if LAYAK(S ∪ {x}) then
        S ← S ∪ {x}
    endif
endwhile
{ SOLUSI(S) or C = {} }

if SOLUSI(S) then
    return S
else
    write('tidak ada solusi')
endif
```

B. Aturan Umum Game Engine Overdrive

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 *power ups* 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 *power ups*. 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 <*lane*> <*block*>
 - 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.

C. Implementasi Algoritma Greedy ke dalam Bot

Algoritma greedy diimplementasikan ke dalam file bot.java pada fungsi run di dalam struktur sebagai berikut

- Starter-pack
 - Starter-bots

- Java
 - src/main/java/za/co/entelect/challenge
 - bot.java
- Target
 - classes/za/co/entelect/challenge
 - bot.classes
 - main.class
 - bot.json

Fungsi run sebagai tempat pengimplementasian algoritma greedy berada dalam file bot.java. Terdapat beberapa command yang dapat dimasukkan ke bot untuk menentukan hal yang akan dipilih oleh bot dalam menjalani setiap turn nya. *Command* yang kami bentuk yaitu :

- ACCELERATE : berfungsi untuk menambah kecepatan ke *state* berikutnya (hingga *maximum speed*). Bot akan tetap berada di jalur yang sama
- USE_BOOST : Menggunakan powerup boost yang telah dimiliki oleh bot, berfungsi untuk meningkatkan kecepatan menjadi *boost speed* selama 5 giliran. Jika tidak memiliki powerup boost, maka efek dari command seperti tidak melakukan apa-apa.
- USE_EMP : menembak EMP lurus didepan jalur kita dan juga ke kanan dan kiri. EMP akan menyebabkan musuh yang terkena berhenti dan mengurangi kecepatannya menjadi 3.
- FIX : Menggunakan *fix* kendaraan akan diam dan mengurangi 2 *damage points*.
- USE_LIZARD : Membuat bot lompat selama satu giliran, yang berarti mengabaikan semua *pickups*, *obstacles*, *player* lain dalam giliran itu kecuali di *block* terakhir. Jika block terakhir ternyata sama dengan *player* lain, maka bot kita akan berada di belakang *player* tersebut.
- USE_OIL : Menggunakan *powerups oil* yang telah kita ambil, meletakkan *block oil* tepat dibawah mobil kita. Jika tidak memiliki *powerups oil*, maka bot tidak melakukan apa-apa.
- TURN_RIGHT : Mengubah jalur mobil untuk giliran selanjutnya. Block pertama bot bergerak akan langsung mengambil jalur yang lebih kanan, dan block selanjutnya lurus mengikuti jalur yang sudah diambil. Jika TURN_RIGHT

digunakan ketika berada di jalur paling kanan, kecepatan akan dikenai *penalty* -1 dan mengurangi *score*.

- TURN_LEFT : Mengubah jalur mobil untuk giliran selanjutnya. Block pertama bot bergerak akan langsung mengambil jalur yang lebih kiri, dan block selanjutnya lurus mengikuti jalur yang sudah diambil. Jika TURN_LEFT digunakan ketika berada di jalur paling kiri, kecepatan akan dikenai *penalty* -1 dan mengurangi *score*.

D. Pengaturan Game Engine

1. Masukan folder bot ke dalam folder starter-pack
2. Edit file “game-runner-config.json” dengan mengubah directory yang ditunjuk pada player-a dan player-b ke tempat dimana “bot.json” berada.
3. Run program dengan menjalankan file “run.bat” atau buka command prompt pada root directory starter-pack dan run perintah run “run.bat”. Pada linux buka terminal di root directory starter-pack dan jalankan perintah “make run”

Bab 3 Aplikasi Strategi *Greedy*

A. Persoalan Overdrive

Mapping persoalan overdrive menjadi elemen-elemen algoritma greedy; himpunan kandidat, himpunan solusi, fungsi solusi, fungsi seleksi, fungsi kelayakan, dan fungsi objektif. Adapun uraian mapping persoalan *Overdrive* ke dalam elemen-elemen algoritma *greedy* adalah:

1. Himpunan kandidat : Perintah NOTHING, perintah ACCELERATE, perintah DECELERATE, perintah TURN_LEFT, perintah TURN_RIGHT, perintah USE_BOOST, perintah USE_OIL, perintah USE_LIZARD, perintah USE_TWEET, perintah USE_EMP, perintah FIX
2. Himpunan solusi : Perintah-perintah yang valid dan sesuai aturan
3. Fungsi solusi : Memeriksa apakah perintah membuat mobil mencapai garis finish
4. Fungsi seleksi : Algoritma memprioritaskan FIX apabila damage yang dimiliki lebih dari 2 untuk menghindari rintangan. Jika tidak perlu FIX maka mobil algoritma akan memprioritaskan menghindari rintangan. Hal ini dilakukan dengan cara menghitung nilai prioritas pada masing-masing lane, depan, kiri, maupun kanan. Kemudian memilih lane yang memiliki obstacles dengan nilai terkecil. Jika jalan lancar maka algoritma akan menggunakan powerups yang dimiliki. Jika tidak memiliki powerups maka akan melakukan accelerate. Jika sudah kecepatan maksimum maka tidak akan melakukan apa-apa.
5. Fungsi kelayakan : Perintah valid seperti menggunakan powerups hanya jika sudah diambil.
6. Fungsi objektif : memeriksa apakah bot mencapai garis *finish* pertama. Jika kedua bot mencapai garis *finish* secara bersamaan, maka bot yang memiliki kecepatan tercepat, dan jika kecepatannya sama, maka bot yang memiliki skor terbesar.

B. Eksplorasi Alternatif Solusi *Greedy*

Eksplorasi alternatif solusi greedy yang mungkin dipilih dalam persoalan overdrive, analisis efisiensi dari kumpulan alternatif solusi greedy yang dirumuskan, analisis efektivitas dari kumpulan alternatif solusi greedy yang dirumuskan

Setelah kami berdiskusi bersama mengenai permainan ini, kami akhirnya mengusulkan ada 4 alternatif solusi yang dapat digunakan, yaitu *greedy by evade*, *greedy by powerups*, and *greedy by speed*.

Pertama, kelompok kami merancang strategi *greedy by evade*. Dalam permainan ada yang dinamakan *obstacles* yang apabila mobil atau bot pemain mengenai *obstacles* ini, akan mengurangi kecepatan dan memberikan kerusakan pada bot. Selain itu juga, posisi mobil musuh dapat mempengaruhi kecepatan mobil juga. Sehingga untuk mempertahankan kecepatan dan meminimalisir kerusakan mobil, kita perlu memilih lokasi lane yang tidak akan mengenai *obstacles* atau mobil lawan. Strategi ini dirasa cukup efektif dalam menjaga konsistensi capaian mobil dan meminimalisasi penggunaan perintah FIX yang membuang giliran. Efisiensi dari algoritma ini adalah O(n).

Kedua, kelompok kami merancang strategi *greedy by powerups*. Dalam permainan ini ada yang dinamakan *power ups* yaitu semacam item yang dapat kita gunakan untuk menyerang lawan dengan cara meletakkan *obstacles* yaitu *cyber truck* atau dengan serangan seperti *EMP*. Apabila mobil mengenai *cyber truck* atau *EMP* maka kecepatannya akan berkurang menjadi 3, sehingga itu akan membuat lawan tertinggal. Strategi ini akan memanfaatkan *power ups* tweet dan emp dengan sebaik mungkin, dan memperbesar kemungkinan agar terkena lawan. Strategi ini dirasa kurang efektif karena mobil tidak selalu memiliki *powerups* sehingga mobil kebanyakan hanya akan diam saja. Efisiensi dari algoritma ini adalah O(n).

Ketiga, kelompok kami merancang strategi *greedy by speed*. Strategi ini berfokus pada kecepatan yaitu memaksimalkan kecepatan kita dan meminimalisir kecepatan lawan. Mobil selalu melakukan perintah ACCELERATE saja untuk memaksimalkan kecepatan. Hanya jika mobil mendapatkan damage perintah FIX dilakukan. Strategi ini dirasa kurang efektif karena mobil akan menggunakan perintah FIX dengan sering sehingga membuang banyak giliran. Efisiensi dari algoritma ini adalah O(n).

C. Strategi *Greedy* yang Dipilih

Strategi *greedy* yang dipilih (yang akan diimplementasikan dalam program) beserta alasan dan pertimbangan pemilihan strategi tersebut.

Strategi yang akhirnya kelompok kami pilih sebagai alternatif terbaik untuk persoalan ini adalah strategi *greedy by evade*, yang berarti dalam implementasinya bot kami selalu menginginkan posisi terbaik. Hal itu dilakukan dengan cara menghindari *obstacles* dan menghindari mobil lawan untuk meminimalisir terjadinya kerusakan pada mobil yang berakibat pengurangan kecepatan.

Hal prioritas pertama yang implementasikan dalam algoritma bot ialah mekanisme FIX. Hal ini kami rasa urgensinya berlaku secara umum, tidak hanya berlaku untuk algoritma greedy by evade. Hal ini penting karena bagaimanapun, apabila tidak mempedulikan FIX, maka mobil akan bergerak semakin lambat dan semakin lama bahkan bisa tidak bergerak sama sekali atau memiliki speed 0. Namun yang menjadi pembeda dari strategi FIX yang kami susun ialah bagaimana kita menganalisis dan memaksimalkan command FIX itu sendiri. Berdasarkan penjelasannya, Command FIX berfungsi untuk memperbaiki *damage points* sebanyak 2 poin. Dari penjelasan awal tersebut, kami mengambil strategi untuk memanfaatkannya dengan lebih maksimal dengan cara menggunakan hanya jika *damage points* lebih dari 2. Selain itu, hal lain dari command FIX yang kami maksimalkan ialah adanya potensi bahwa *damage* yang diterima oleh mobil tidak berpengaruh signifikan, yakni ketika kondisi awal kecepatan mobil rendah dan jauh dari kecepatan maksimal. Sehingga, meskipun kita menggunakan FIX untuk meningkatkan kecepatan maksimal, akan menjadi hal yang sia-sia karena kecepatan mobil kita masih jauh dari maksimal. Sebagai contoh, kecepatan mobil berada di *initial speed* atau 5 dan belum menerima *damage* sama sekali atau 0. Jika mobil terkena 1 *damage*, maka sesuai aturan, *damage* 1 menyebabkan kecepatan maksimum di 9, atau tidak berpengaruh sama sekali jika tidak memakai *powerups boost*. Begitupun ketika *damage* 2, meski kecepatan maksimum mobil menurun menjadi 8, akan tetapi karena kondisinya saat ini kecepatan masih berada di 5, maka belum terlalu penting bagi kita untuk mengembalikan kecepatan maksimum menjadi 15. Oleh karena itu, algoritma yang kami gunakan ialah menggunakan FIX jika kecepatan berada di maksimum dan *damage points* lebih dari 2.

Setelah itu, sesuai dengan algoritma yang kami pilih yakni *greedy by evade*, maka prioritas setelahnya ialah pencarian posisi terbaik untuk mempertahankan kecepatan dan mengurangi *damage* yang terjadi. Oleh karena itu, kami mencari cara untuk menghindari obstacle berupa wall, truck, mud, dan juga oil. Yang diprioritaskan ialah menghindari wall dan juga mobil lain yang ada di depan kita, karena wall mengurangi kecepatan secara drastis hingga ke level 1 (hampir tidak bergerak sama sekali) sedangkan mobil lain berpotensi untuk memberikan *damage* dan pengurangan kecepatan kepada mobil kita dengan berbagai cara. Untuk menghindari hal tersebut, kita menggunakan fungsi seleksi untuk memutuskan apakah akan belok kiri atau kanan dan mengecek apakah memiliki *powerups LIZARD* yang diprioritaskan jika kita memilikinya. Selain itu, yang berada di urutan kedua dari prioritas obstacle ialah *mud*

dan *oil* karena impact yang diberikan tidak sebesar wall dan mobil musuh. Seperti halnya menghindari wall dan mobil, kita juga menggunakan fungsi seleksi untuk memutuskan langkah yang akan diambil.

Dalam penggunaan strategi FIX dan menghindari obstacles, kami mempertimbangkan pola penyusunan prioritasnya. Kami memprioritaskan untuk melakukan FIX seperti yang dijelaskan di atas di urutan pertama, setelah itu, di urutan kedua kami memprioritaskan untuk menghindari *wall* dan *truck*. Setelah itu, apabila tidak terdapat *wall* ataupun *truck*, kami prioritaskan untuk FIX jika damage 1 atau 2 dan jika memiliki *powerups boost*. Setelah itu, baru kami prioritaskan untuk menghindari *mud* dan *oil*.

Jika tidak ada FIX dan *obstacles* yang harus diprioritaskan untuk ditangani, maka selanjutnya baru kami memprioritaskan *powerups* jika dimiliki. *Powerups* EMP digunakan jika terdapat mobil musuh dalam jangkauan EMP. begitupun *powerups* Tweet dan juga Oil.

Di prioritas terakhir, digunakan *command ACCELERATE* jika tidak memiliki prioritas untuk dihindari, FIX, ataupun *powerups* untuk digunakan, dengan syarat area di depan aman. Apabila tidak aman untuk dilakukan ACCELERATE, maka dilakukan command DO NOTHING.

Bab 4 Implementasi dan Pengujian

A. Implementasi Algoritma *Greedy* dalam *Game Engine*

```
run () → command

{melakukan perintah fix jika speed mencapai kecepatan maksimum sesuai
ukuran damage}
if (damage = 5) then → fix
if (damage = 4 and speed = 3) then → fix
if (damage = 3 and speed = 6) then → fix
if (damage = 2 and speed = 8) then → fix

{menghindari terrain wall atau cyber truck}
if (nexttile(car.lane) = wall or nexttile(car.lane) = cybertruck)
then
    if (nexttile(car.lane + 1) = empty and car.lane != 4) then →
turn_right
    Else if (nexttile(car.lane - 1) = empty and car.lane != 1) then
→ turn_left
    Else if (9nexttile(car.lane + 1) = mud or nexttile(car.lane +
1) = oil) and car.lane != 4) the → turn_right
    Else if ((nexttile(car.lane - 1) = mud or nexttile(car.lane -
1) = oil) and car.lane != 1) the → turn_left
    Else
        If (car.lane != 4) → turn_right
        Else → turn_left

{menghindari terrain mud atau oil}
if (nexttile(car.lane) = mud or nexttile(car.lane) = oil) then
    if (nexttile(car.lane + 1) = empty and car.lane != 4) then →
turn_right
    Else if (nexttile(car.lane - 1) = empty and car.lane != 1) then
→ turn_left
```

```

{mengaktifkan powerups jika ada}

if (haspowerup(boost)) then → boost
if (haspowerup(emp)) then → emp
if (haspowerup(tweet)) then → tweet car.lane car.block-1
if (haspowerup(oil)) then → oil

{accelerate jika belum kecepatan maksimum, do nothing jika tidak}
if (car.speed != max_speed) then → accelerate else → nothing

{fungsi dan prosedur}
nexttile(lane) → obstacle
{menerima lane dan mengeluarkan apakah ada rintangan pada lane tersebut}

haspowerup(powerup) → boolean
{menerima sebuah powerup dan memeriksa apakah powerup tersebut dimiliki atau tidak}

```

B. Struktur Data

Adapun struktur data yang digunakan pada permainan Overdrive ini terbagi menjadi 3 yaitu *command*, *entities*, dan *enums* yang disertakan dengan Bot.java dan Main.java.

1. Command berisi perintah-perintah yang bot dapat lakukan.
 - a. DoNothingCommand : perintah untuk tidak melakukan apapun
 - b. AccelerateCommand : perintah untuk menambah kecepatan ke level kecepatan berikutnya
 - c. DecelerateCommand : perintah untuk mengurangi kecepatan ke level kecepatan sebelumnya
 - d. ChangeLaneCommand : perintah untuk mengubah posisi mobil ke lane yang ada diatas atau dibawahnya.
 - e. BoostCommand : perintah untuk menggunakan powerup boost untuk menjadikan kecepatan mobil maksimum

- f. OilCommand : perintah untuk menggunakan powerup oil untuk menaruh tile oil pada posisi mobil.
 - g. LizardCommand : perintah untuk menggunakan powerup lizard untuk melompati rintangan
 - h. TweetCommand : perintah untuk menggunakan powerup tweet untuk menaruh cyber truck pada lane dan blok tertentu
 - i. EmpCommand : perintah untuk menggunakan powerup emp untuk menembak emp pada mobil lawan
 - j. FixCommand : perintah untuk memperbaiki mobil. Perintah mengurangi 2 poin damage pada mobil.
2. Entities adalah objek yang ada dalam game.
 - a. Car : objek mobil dengan atribut id, position, speed, damage, state, powerups, boosting, dan boostcounter.
 - b. GameState : objek game dengan atribut currentRound, maxRounds, player, opponent, worldMap
 - c. Lane : objek jalur balapan dengan attribut position, surfaceObject, dan occupiedByPlayerId
 - d. Position : objek posisi berupa koordinat x dan y
 3. Enums adalah alat iterasi.
 - a. Direction : arahan mobil untuk bergerak
 - b. PowerUps : list powerups yang ada
 - c. State : list status mobil
 - d. Terrain : list tipe tile
 4. Bot.java berisi bot yang telah diimplementasikan
 5. Main.java berisi program utama yang membaca current state, mengirimkannya ke bot, mendapat output dari bot, dan menuliskan ke layar

C. Analisis Desain Solusi

Match 1:

Awal Permainan

MATCHES ADD A MATCH

2022.02.18.21.16.15.zip (selected)

Round 001

Reset Remove this match

[1, 1] GUNNAR	[2, 1]	[3, 1]	[4, 1]	[5, 1]	[6, 1]	[7, 1] PLAYERS	[8, 1]	[9, 1]	[10, 1]	[11, 1]	[12, 1]	[13, 1] GUNNAR	[14, 1]	[15, 1]	[16, 1]	[17, 1]	[18, 1]	[19, 1]	[20, 1] GUNNAR
[1, 2]	[2, 2]	[3, 2]	[4, 2]	[5, 2]	[6, 2]	[7, 2]	[8, 2]	[9, 2]	[10, 2]	[11, 2]	[12, 2]	[13, 2]	[14, 2]	[15, 2]	[16, 2]	[17, 2]	[18, 2]	[19, 2]	[20, 2]
[1, 3]	[2, 3]	[3, 3]	[4, 3]	[5, 3]	[6, 3]	[7, 3]	[8, 3]	[9, 3]	[10, 3]	[11, 3]	[12, 3]	[13, 3]	[14, 3]	[15, 3]	[16, 3]	[17, 3]	[18, 3]	[19, 3] GUNNAR	[20, 3]
[1, 4] GUNNAR	[2, 4]	[3, 4]	[4, 4]	[5, 4]	[6, 4]	[7, 4]	[8, 4]	[9, 4]	[10, 4]	[11, 4]	[12, 4]	[13, 4]	[14, 4]	[15, 4]	[16, 4]	[17, 4]	[18, 4]	[19, 4]	[20, 4]

1 | < First < Prev 1 Next > Last >

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

Round Details Max Rounds: 600 Current Round: 1

A - HeavyMetal

Position Speed Lane Distance Boosts Boosting Powerups

1 [c 1, y: 1]	5	1	1	0	No	
------------------	---	---	---	---	----	--

Bot Command

RESTART ENGINE

Tengah Permainan

Entelect Challenge 2020 - Replay https://entelect-replay.raezor.co.za/#

SIK Edunex Chat Gmail Calendar Fangath New Nor... DatCen (-ω-) Tubes 1 STIMA Semester 4 - Googl... Establish 6 Core Ha... Muhammad Akmal...

Created by RAEZOR

MATCHES ADD A MATCH

2022.02.18.21.16.15.zip (selected)

Round 084

Reset Remove this match

[797, 1]	[798, 1]	[799, 1]	[800, 1]	[801, 1]	[802, 1]	[803, 1]	[804, 1]	[805, 1]	[806, 1]	[807, 1]	[808, 1]	[809, 1]	[810, 1]	[811, 1] GUNNAR	[812, 1]	[813, 1]	[814, 1]	[815, 1]	[816, 1] GUNNAR	[817, 1]	[818, 1]	[819, 1]	[820, 1]	[821, 1]	[822, 1]
[797, 2]	[798, 2]	[799, 2]	[800, 2]	[801, 2]	[802, 2]	[803, 2]	[804, 2] PLAYERS	[805, 2]	[806, 2]	[807, 2]	[808, 2]	[809, 2]	[810, 2]	[811, 2] GUNNAR	[812, 2]	[813, 2]	[814, 2] PLAYERS	[815, 2]	[816, 2] GUNNAR	[817, 2]	[818, 2]	[819, 2]	[820, 2]	[821, 2]	[822, 2]
[797, 3]	[798, 3]	[799, 3]	[800, 3]	[801, 3]	[802, 3]	[803, 3]	[804, 3] PLAYERS	[805, 3]	[806, 3]	[807, 3]	[808, 3]	[809, 3]	[810, 3]	[811, 3]	[812, 3]	[813, 3]	[814, 3]	[815, 3]	[816, 3] PLAYERS	[817, 3]	[818, 3]	[819, 3]	[820, 3]	[821, 3]	[822, 3]
[797, 4]	[798, 4]	[799, 4]	[800, 4]	[801, 4]	[802, 4]	[803, 4]	[804, 4]	[805, 4]	[806, 4]	[807, 4]	[808, 4]	[809, 4]	[810, 4]	[811, 4] GUNNAR	[812, 4]	[813, 4]	[814, 4] GUNNAR	[815, 4]	[816, 4] GUNNAR	[817, 4] GUNNAR	[818, 4]	[819, 4]	[820, 4]	[821, 4]	[822, 4]

1 | < First < Prev 84 Next > Last >

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

Round Details Max Rounds: 600 Current Round: 84

A - HeavyMetal

Position Speed Lane Distance Boosts Boosting Powerups

1 [c 802, y: 3]	15	3	802	5	YES	EMP,LIZARD,EMPO ILIZARD,LIZARD, LIZARD,OIL,EMP,E MP,OIL
--------------------	----	---	-----	---	-----	--

RESTART ENGINE

Match 2:

Awal Permainan

MATCHES ADD A MATCH

2022.02.18.21.16.30.zip (selected)

[1, 1] -PLAYER	[2, 1]	[3, 1]	[4, 1]	[5, 1]	[6, 1]	[7, 1]	[8, 1]	[9, 1]	[10, 1]	[11, 1]	[12, 1] -WALLS	[13, 1] -OBSTACLES	[14, 1]	[15, 1]	[16, 1]	[17, 1]	[18, 1]	[19, 1]	[20, 1]	[21, 1]
[1, 2]	[2, 2]	[3, 2]	[4, 2]	[5, 2]	[6, 2]	[7, 2]	[8, 2]	[9, 2]	[10, 2]	[11, 2]	[12, 2]	[13, 2]	[14, 2]	[15, 2]	[16, 2]	[17, 2]	[18, 2]	[19, 2]	[20, 2]	[21, 2]
[1, 3]	[2, 3]	[3, 3]	[4, 3]	[5, 3]	[6, 3]	[7, 3]	[8, 3]	[9, 3]	[10, 3]	[11, 3]	[12, 3]	[13, 3]	[14, 3]	[15, 3]	[16, 3]	[17, 3]	[18, 3]	[19, 3]	[20, 3]	[21, 3]
[1, 4] -PLAYER	[2, 4]	[3, 4]	[4, 4]	[5, 4]	[6, 4]	[7, 4]	[8, 4]	[9, 4]	[10, 4]	[11, 4]	[12, 4]	[13, 4]	[14, 4] -OBSTACLES	[15, 4] -WALLS	[16, 4] -OBSTACLES	[17, 4] -WALLS	[18, 4]	[19, 4]	[20, 4]	[21, 4]

First Prev **1** Next > Last >

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

Round Details Max Rounds: 600 Current Round: 1

A - HeavyMetal (selected)

Position	Speed	Lane	Distance	Boosts	Boosting	Powerups
1 [c: 1, y: 1]	5	1	1	0	No	

Tengah Permainan

MATCHES ADD A MATCH

2022.02.18.21.16.30.zip

[11, 1]	[12, 1]	[13, 1]	[14, 1]	[15, 1]	[16, 1]	[17, 1]	[18, 1] -OBSTACLES	[19, 1]	[20, 1]	[21, 1]	[22, 1]	[23, 1]	[24, 1] -WALLS	[25, 1]	[26, 1]	[27, 1]	[28, 1]	[29, 1]	[30, 1] -OBSTACLES	[31, 1]	[32, 1]	[33, 1]	[34, 1]	[35, 1]	[36, 1]	[37, 1]	[38, 1]	[39, 1]	[40, 1]	[41, 1]	[42, 1]
[11, 2]	[12, 2]	[13, 2]	[14, 2]	[15, 2]	[16, 2]	[17, 2]	[18, 2]	[19, 2]	[20, 2]	[21, 2]	[22, 2]	[23, 2]	[24, 2]	[25, 2]	[26, 2]	[27, 2]	[28, 2]	[29, 2]	[30, 2]	[31, 2]	[32, 2]	[33, 2]	[34, 2]	[35, 2]	[36, 2]	[37, 2]	[38, 2]	[39, 2]	[40, 2]	[41, 2]	[42, 2]
[11, 3]	[12, 3]	[13, 3]	[14, 3]	[15, 3]	[16, 3]	[17, 3]	[18, 3]	[19, 3]	[20, 3]	[21, 3]	[22, 3]	[23, 3]	[24, 3]	[25, 3]	[26, 3]	[27, 3]	[28, 3]	[29, 3]	[30, 3]	[31, 3]	[32, 3]	[33, 3]	[34, 3]	[35, 3]	[36, 3]	[37, 3]	[38, 3]	[39, 3]	[40, 3]	[41, 3]	[42, 3]
[11, 4] -OBSTACLES	[12, 4]	[13, 4]	[14, 4]	[15, 4]	[16, 4]	[17, 4]	[18, 4]	[19, 4]	[20, 4]	[21, 4]	[22, 4]	[23, 4]	[24, 4]	[25, 4]	[26, 4]	[27, 4]	[28, 4]	[29, 4]	[30, 4]	[31, 4]	[32, 4]	[33, 4]	[34, 4]	[35, 4]	[36, 4]	[37, 4]	[38, 4]	[39, 4]	[40, 4]	[41, 4]	[42, 4]

First Prev **102** Next > Last >

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

Round Details Max Rounds: 600 Current Round: 102

A - HeavyMetal (selected)

Position	Speed	Lane	Distance	Boosts	Boosting	Powerups
2 [c: 722, y: 2]	3	2	722	0	No	LIZARD,LIZARD,EM PEMP,PEMP,PEMP P

Akhir Permainan

MATCHES **ADD A MATCH**

2022.02.18.21.16.30.zip (selected)

Round 210

[1432, 1]	[1433, 1]	[1434, 1]	[1435, 1]	[1436, 1]	[1437, 1] +GOLDEN+	[1438, 1]	[1439, 1] +GOLDEN+	[1440, 1] +GOLDEN+	[1441, 1]	[1442, 1]	[1443, 1]	[1444, 1]	[1445, 1]	[1446, 1]	[1447, 1]	[1448, 1]	[1449, 1]	[1450, 1]	[1451, 1]	[1452, 1] +GOLDEN+	[1453, 1] +GOLDEN+	[1454, 1] +GOLDEN+	[1455, 1] +GOLDEN+	[1456, 1] +GOLDEN+	[1457, 1] +GOLDEN+
[1432, 2]	[1433, 2]	[1434, 2]	[1435, 2]	[1436, 2]	[1437, 2]	[1438, 2]	[1439, 2]	[1440, 2]	[1441, 2]	[1442, 2]	[1443, 2]	[1444, 2]	[1445, 2]	[1446, 2]	[1447, 2]	[1448, 2]	[1449, 2]	[1450, 2]	[1451, 2]	[1452, 2] +GOLDEN+	[1453, 2] +GOLDEN+	[1454, 2] +GOLDEN+	[1455, 2] +GOLDEN+	[1456, 2] +GOLDEN+	[1457, 2] +GOLDEN+
[1432, 3]	[1433, 3]	[1434, 3]	[1435, 3]	[1436, 3]	[1437, 3]	[1438, 3]	[1439, 3]	[1440, 3]	[1441, 3]	[1442, 3]	[1443, 3]	[1444, 3]	[1445, 3]	[1446, 3]	[1447, 3] +GOLDEN+	[1448, 3] +GOLDEN+	[1449, 3]	[1450, 3]	[1451, 3]	[1452, 3] +GOLDEN+	[1453, 3] +GOLDEN+	[1454, 3] +GOLDEN+	[1455, 3] +GOLDEN+	[1456, 3] +GOLDEN+	[1457, 3] +GOLDEN+
[1432, 4]	[1433, 4]	[1434, 4]	[1435, 4]	[1436, 4]	[1437, 4]	[1438, 4] +GOLDEN+	[1439, 4] +GOLDEN+	[1440, 4] +GOLDEN+	[1441, 4]	[1442, 4]	[1443, 4]	[1444, 4]	[1445, 4]	[1446, 4]	[1447, 4]	[1448, 4]	[1449, 4]	[1450, 4]	[1451, 4]	[1452, 4]	[1453, 4]	[1454, 4]	[1455, 4]	[1456, 4]	[1457, 4] +GOLDEN+

[< First] [< Prev] **210** [Next >] [Last >]
(Click the button that displays the round number to quickly switch rounds)

Round Details Max Rounds: 600 Current Round: 210

A - HeavyMetal (selected)

Position	Speed	Lane	Distance	Boosts	Boosting	Powerups
2 [x: 1437, y: 1]	9	1	1437	0	No	OIL,OIL,OIL,OIL,T WEET,TWEET,OIL,O IL,OIL,OIL,TWEET,T WEET,LIZARD,OIL, TWEET,TWEET,OIL

RESTART ENGINE

Match 3:

Awal Permainan

MATCHES **ADD A MATCH**

2022.02.18.21.16.15.zip

2022.02.18.21.16.30.zip

2022.02.18.21.16.50.zip (selected)

Round 001

[1, 1] +GOLDEN+	[2, 1]	[3, 1]	[4, 1]	[5, 1]	[6, 1]	[7, 1] +GOLDEN+	[8, 1] +GOLDEN+	[9, 1]	[10, 1]	[11, 1]	[12, 1]	[13, 1]	[14, 1]	[15, 1]	[16, 1]	[17, 1]	[18, 1]	[19, 1]	[20, 1]	[21, 1]
[1, 2]	[2, 2]	[3, 2]	[4, 2]	[5, 2]	[6, 2]	[7, 2]	[8, 2]	[9, 2]	[10, 2]	[11, 2]	[12, 2]	[13, 2]	[14, 2]	[15, 2]	[16, 2]	[17, 2]	[18, 2]	[19, 2]	[20, 2] +GOLDEN+	[21, 2]
[1, 3]	[2, 3]	[3, 3]	[4, 3]	[5, 3]	[6, 3]	[7, 3]	[8, 3]	[9, 3]	[10, 3]	[11, 3]	[12, 3]	[13, 3]	[14, 3]	[15, 3]	[16, 3]	[17, 3]	[18, 3]	[19, 3]	[20, 3]	[21, 3] +GOLDEN+
[1, 4] +GOLDEN+	[2, 4]	[3, 4]	[4, 4]	[5, 4]	[6, 4]	[7, 4] +GOLDEN+	[8, 4]	[9, 4]	[10, 4]	[11, 4]	[12, 4]	[13, 4] +GOLDEN+	[14, 4] +GOLDEN+	[15, 4]	[16, 4]	[17, 4] +GOLDEN+	[18, 4]	[19, 4]	[20, 4]	[21, 4]

[< First] [< Prev] **1** [Next >] [Last >]
(Click the button that displays the round number to quickly switch rounds)

Round Details Max Rounds: 600 Current Round: 1

A - HeavyMetal (selected)

Position	Speed	Lane	Distance	Boosts	Boosting	Powerups
1 [x: 1, y: 1]	5	1	1	0	No	

RESTART ENGINE

Bot Command

Tengah Permainan

secara default mobil akan accelerate jika belum mencapai kecepatan maksimum. Jika sudah mencapai kecepatan maksimum maka mobil tidak melakukan apa-apa.

Bab 5 Kesimpulan dan Saran

A. Kesimpulan

Dengan mengimplementasikan materi mata kuliah IF2211 Strategi Algoritma mengenai algoritma greedy pada bot dalam permainan Overdrive, kami berhasil membuat bot yang cukup baik dalam mencapai objektif yang ada, yakni mencapai garis finish secepat mungkin. Strategi greedy terbaik yang pada akhirnya kami pilih ialah strategi greedy by position yakni dalam setiap langkahnya berusaha mendapatkan posisi sebaik mungkin dan mempertahankan kecepatan dengan menghindari obstacles dan melakukan perbaikan dengan memanfaatkan detail-detail yang ada di setiap *command* semaksimal mungkin.

B. Saran

Dalam proses pengerjaan projek semacam ini di masa yang akan datang ataupun projek lanjutan dari hal ini, alangkah baiknya jika pemrogram mempelajari terlebih dahulu dasar-dasar paradigma pemrograman berbasis objek dan juga dasar pemrograman Java untuk mempermudah pemrogram mengimplementasikan algoritma yang telah dibuat. Selain itu, pemrogram juga direkomendasikan untuk mempelajari Dokumentasi program yang akan dikerjakan, sebagai contoh dalam hal ini ialah Overdrive. Untuk penelitian pencarian algoritma dalam permainan overdrive yang dilakukan selanjutnya, salah satu hal yang dapat dipertimbangkan ialah pengecekan obstacles sejauh dua ronde ke depan, yang dalam penelitian ini belum dilakukan karena terkendala waktu.

DAFTAR PUSTAKA

Munir, R., 2022. *Algoritma Greedy (Bagian 1)*. [online] Informatika.stei.itb.ac.id. Available at: <[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)> [Accessed 2 February 2022].

Munir, R., 2022. *Algoritma Greedy (Bagian 2)*. [online] Informatika.stei.itb.ac.id. Available at: <[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag2.pdf)> [Accessed 2 February 2022].

Munir, R., 2022. *Algoritma Greedy (Bagian 3)*. [online] Informatika.stei.itb.ac.id. Available at: <[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Greedy-\(2022\)-Bag3.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Greedy-(2022)-Bag3.pdf)> [Accessed 2 February 2022].

EntelectChallenge. 2020. Overdrive. [online] github.com. Available at: <<https://github.com/EntelectChallenge/2020-Overdrive/>> [Accessed 2 February 2022].

LAMPIRAN

Link Github

https://github.com/DaffaRomyz/Tubes1_Heavymetal