

**LAPORAN TUGAS BESAR 1 IF2211**  
**Strategi Algoritma**  
**Pemanfaatan Algoritma Greedy dalam Aplikasi Permainan “Overdrive”**  
**Kelompok Mix Team 3**

Foto :



Disusun Oleh

1. Ilham Pratama(13520041)
2. Steven Gianmarg Haposan Siahaan(13520145)
3. Dwi Kalam Amal Tauhid (13519210)

Sekolah Teknik Elektro dan Informatika - Institut  
Teknologi Bandung Jl. Ganesha 10, Bandung 40132

# BAB 1

## DESKRIPSI TUGAS

Overdrive adalah sebuah game yang mempertandingan 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.

Mahasiswa ditugaskan 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>

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 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
  - 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 2

### LANDASAN TEORI

Algoritma greedy merupakan metode yang populer dan sederhana untuk memecahkan persoalan optimasi. Algoritma ini memecahkan persoalan secara langkah per langkah ( step by step ) sedemikian sehingga, pada setiap langkah dilakukan pengambilan pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensinya ke depan (prinsip “take what you can get now!”) dan “berharap” bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global. Optimum lokal ini dapat ditentukan dari banyak faktor tergantung persoalan, seperti contohnya selalu mengambil pilihan dengan nilai variabel tertentu yang terbesar

Elemen-elemen algoritma greedy :

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 objektif : memaksimumkan atau meminimumkan

Secara umum algoritma greedy adalah sebagai berikut :

```
function greedy( $C$  : himpunan_kandidat)  $\rightarrow$  himpunan_solusi
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy }
Deklarasi
 $x$  : kandidat
 $S$  : himpunan_solusi
Algoritma:
 $S \leftarrow \{\}$  { inialisasi  $S$  dengan kosong }
while ( not SOLUSI( $S$ )) and ( $C \neq \{\}$  ) do
 $x \leftarrow$  SELEKSI( $C$ ) { pilih sebuah kandidat dari  $C$  }
 $C \leftarrow C - \{x\}$  { buang  $x$  dari  $C$  karena sudah dipilih }
if LAYAK( $S \cup \{x\}$ ) then {  $x$  memenuhi kelayakan untuk dimasukkan ke dalam himpunan
solusi }
 $S \leftarrow S \cup \{x\}$  { masukkan  $x$  ke dalam himpunan solusi }
endif
endwhile
```

```
{SOLUSI(S) or C = {} }  
if SOLUSI(S) then { solusi sudah lengkap }  
    return S  
else  
    write ('tidak ada solusi')  
endif
```

## BAB 3

### APLIKASI STRATEGI *GREEDY*

Dalam mengimplementasikan strategi *greedy* pada persoalan ini, penulis *me-mapping* persoalan *Overdrive* menjadi elemen-elemen algoritma *Greedy*.

- Himpunan kandidat  
{Nothing, accelerate, decelerate, turn\_left, tur\_right, use\_boost, use\_oil, use\_lizard, use\_tweet, isEnd}
- Himpunan solusi  
Setiap *command* yang valid dan sesuai aturan main untuk setiap ronde serta permainan selesai.
- Fungsi solusi  
Memeriksa apakah permainan telah selesai.
- Fungsi seleksi  
Pilihlah
- Fungsi kelayakan  
Memeriksa apakah *command* yang dipilih valid. Misal, mobil tidak melebihi batas jalur *lane* dan *power-ups* yang hendak digunakan tersedia.
- Fungsi objektif  
Memenangkan permainan sesuai aturan yang ditetapkan.

#### Eksplorasi strategi *greedy*

Penulis telah mengeksplorasi beberapa strategi *greedy* yang dapat diimplementasikan.

Strategi *greedy* yang pertama adalah *greedy by max\_speed*. Pada strategi ini, mobil akan berfokus melewati setiap hambatan yang ada di hadapannya atau meningkatkan kecepatan, yakni mulai dengan cara menggunakan power ups lizard, pindah lane, power up boost, atau penggunaan power up lainnya yang dimiliki. Strategi ini dapat menjadi alternatif sebab salah satu aspek penilaian kemenangan adalah berdasarkan kecepatan maksimumnya. Namun, pemilihan strategi ini bisa jadi tidak terlalu optimal jika ternyata di suatu kesempatan, pengambilan dan penggunaan power ups lebih memiliki *value* dalam mencapai kemenangan. Efisiensi strategi ini diperkirakan adalah  $O(n)$ .

Strategi kedua adalah *greedy by annoy*. Pada strategi ini, mobil akan berfokus menyerang mobil lawan dengan mengeluarkan power ups yang dimilikinya. Strategi ini dapat menjadi alternatif sebab lawan akan terus terganggu lajunya. Namun, hal ini dapat berisiko jika mobil tidak memiliki amunisi power ups yang cukup untuk mengganggu lawannya, terlebih jika jalur untuk mengambil power ups-nya banyak hadangan. Efisiensi strategi ini diperkirakan adalah  $O(n)$

Akhirnya, strategi greedy yang kami pilih adalah *greedy by max\_speed*. Pertimbangan penulis adalah konsistensi performa yang dapat dihasilkan oleh mobil. Sebab, strategi ini berfokus pada konsistensinya dalam menjaga atau meningkatkan kecepatan serta menghasilkan kecepatan maksimal yang menjadi poin yang sangat penting pada aspek kemenangan *overdrive* ini. Sebagai akibatnya, strategi kedua tidak kami pilih dengan alasan konsistensi performa yang relatif lebih rendah dibandingkan strategi pertama.

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

Berikut adalah implementasi bot penulis dalam bentuk pseudocode.

```
If (di depan ada hambatan) then
    If (punya power ups lizard) then
        gunakan lizard
    else
        pindah lane

if (mobil.lane > 1) then
    hitung reduksi speed ketika mobil pindah ke kiri

If (mobil.lane < 4) then
    hitung reduksi speed ketika mobil pindah ke kanan

If (mobil.damage >= 2) then
    FIX mobil
else
    If (mobil.speed < 15) then
        if (mobil tidak perlu direm) then
            If (mobil punya BOOST) then
                if(car.speed = 3) then
                    Gunakan BOOST
                else
                    If (mobil.speed < maxSpeed) then
                        percepat
                    else
                        if (mobil lawan di depan) then
                            If (punya EMP) then
                                Gunakan EMP
                            Else if (punya TWEET) then
                                Gunakan tweet
                            Else
                                Tidak melakukan apa apa
                        else
                            If (punya OIL) then
                                Gunakan oil
                            Else
                                Tidak melakukan apa apa
                    else
                        If (mobil.speed < maxSpeed) then
                            percepat
                        else
                            if (mobil lawan di depan) then
                                If (punya EMP) then
                                    Gunakan EMP
```



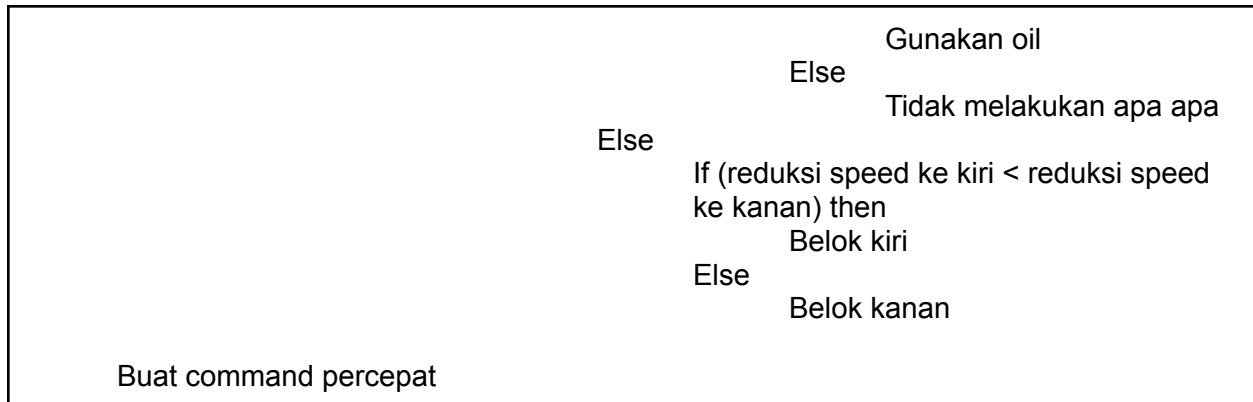
```

        Else if (punya TWEET) then
            Gunakan tweet
        Else
            Tidak melakukan apa apa
    else
        If (punya OIL) then
            Gunakan oil
        Else
            Tidak melakukan apa apa

else //mobil perlu di rem
    If (mobil.speed < maxSpeed) then
        If (reduksi speed ke depan <= reduksi speed ke kiri OR reduksi
            speed ke depan <= reduksi speed ke kanan)
            Percepat
        Else
            If (reduksi speed ke kiri < reduksi speed ke kanan)
                Belok kiri
            Else
                Belok kanan

    Else
        If (reduksi speed ke kiri > 0 AND reduksi speed ke kanan = 0)
            then
                Belok kanan
        Else if (reduksi speed ke kanan > 0 AND reduksi speed ke kiri =
            0 then
                Belok kiri
        Else if ( reduksi speed ke kanan = 0 AND reduksi speed ke kiri =
            0) then
                Belok kanan atau belok kiri
        Else
            If (punya Lizard) then
                Gunakan Lizard
            Else
                If (reduksi speed ke depan < reduksi speed ke
                    kiri OR reduksi speed ke depan < reduksi speed
                    ke kanan) then
                    If (mobil lawan di depan) then
                        If (punya EMP) then
                            Gunakan EMP
                        Else if (punya TWEET) then
                            Gunakan tweet
                        Else
                            Tidak melakukan apa apa
                    else
                        If (punya OIL) then

```



### Struktur Data pada Program

Secara garis besar, struktur data program ini memiliki 3 bagian utama, yaitu *command*, *enum*, dan *entities*, yang turut disertai *bot.java* dan *main.java*

1. Command, berfungsi untuk menyimpan kelas-kelas yang berhubungan dengan prosedur *game-engine*.
  - a. AccelerateCommand, untuk mempercepat laju mobil
  - b. BoostCommand, untuk menggunakan power ups boost
  - c. ChangeLaneCommand, untuk berpindah lane
  - d. Command, interface untuk command-command lainnya
  - e. DecelerateCommand, untuk memperlambat mobil
  - f. DoNothingCommand, tidak melakukan apa pun di suatu round
  - g. EmpCommand, menggunakan power up EMP
  - h. FixCommand, memperbaiki mobil
  - i. LizardCommand, menggunakan power up lizard
  - j. OilCommand, menggunakan power up oil
  - k. TweetCommand, menggunakan power up tweet
2. Enum, berguna sebagai alat iterasi yang dibuat untuk mencocokkan setiap kemungkinan yang ada.
  - a. Direction, berisi objek-objek yang mendefinisikan arah
  - b. PowerUps, berisi objek-objek power ups
  - c. State, objek-objek status ketika game telah dijalankan
  - d. Terrain, objek-objek pada suatu blok
3. Entities, sebagai objek yang ada di gamestate
  - a. Car: objek mobil yang ada di permainan
  - b. GameState: Objek berupa status-status pada game
  - c. Lane: Objek berupa informasi pada lane
  - d. Position: Objek berupa posisi lane dan blok
4. Bot.java, berisi implementasi/ algoritma pada bot
5. Main.java, berisi mekanisme pembacaan state dan eksekusi state

## **BAB 5**

### **KESIMPULAN DAN SARAN**

Penulis berhasil mengimplementasikan algoritma greedy pada bot di persoalan ini. Tujuan dari pengimplementasiannya adalah memenangkan pertandingan. Strategi greedy yang penulis pilih adalah greedy by max\_speed, yaitu berfokus pada menghindari hambatan atau meningkatkan kecepatan.

Saran untuk pengerjaan tubes ini adalah perbanyak eksplorasi strategi *greedy* sehingga lebih dapat mempertimbangkan strategi greedy yang paling optimal.

## DAFTAR PUSTAKA

1. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Tugas-Besar-1-IF2211-Strategi-Algoritma-2022.pdf>
2. <https://github.com/EntelectChallenge/2020-Overdrive/releases/tag/2020.3.4>
3. <https://github.com/EntelectChallenge/2020-Overdrive/blob/master/game-engine/game-rules.md#obstacles>