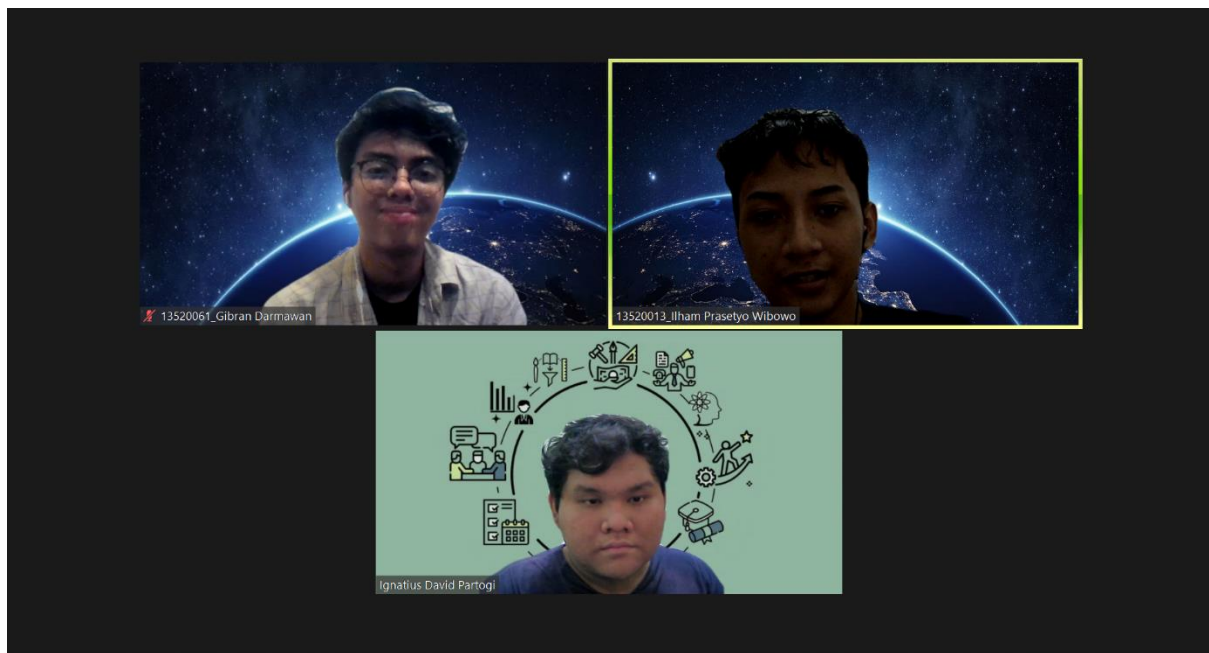


LAPORAN TUGAS BESAR STRATEGI ALGORITMA

“Pemanfaatan Algoritma Greedy dalam Aplikasi Permainan “Overdrive””

IF2211 STRATEGI ALGORITMA



Dosen pengajar : Dr. Masayu Leylia Khodra, S.T., M.T.

Kelompok : Mix Team 2

Disusun oleh :

Ignatius David Partogi (13518014)

Ilham Prasetyo Wibowo (13520013)

Gibran Darmawan (13520061)

**PROGRAM STUDI TEKNIK INFORMATIKA
SETKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2021**

BAB I

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.

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/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>

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:
 1. Oil item, dapat menumpahkan oli di bawah mobil anda berada.
 2. Boost, dapat mempercepat kecepatan mobil anda secara drastis.
 3. Lizard, berguna untuk menghindari lizard yang mengganggu jalan mobil anda.
 4. Tweet, dapat menjatuhkan truk di block spesifik yang anda inginkan.
 5. 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 <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.

BAB II

LANDASAN TEORI

1. Algoritma Greedy

Algoritma greedy adalah sebuah algoritma yang memecahkan persoalan secara langkah per langkah. Sehingga pada algoritma greedy pada setiap langkah mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan. Dengan demikian diharapkan dapat memilih solusi optimum lokal terlebih dahulu dan pada saat algoritma selesai berakhir dengan solusi yang optimum global.

Algoritma greedy memiliki beberapa elemen yaitu :

- i. Himpunan kandidat
Himpunan kandidat bersisi dengan kandidat yang akan dipilih pada setiap langkah.
- ii. Himpunan solusi
Himpunan solusi menyimpan kandidat yang sudah dipilih
- iii. Fungsi solusi
Fungsi solusi menentukan apakah kandidat yang dipilih memberikan solusi untuk tujuan yang akan dicapai.
- iv. Fungsi seleksi
Fungsi seleksi memilih kandidat dari himpunan kandidat berdasarkan strategi greedy tertentu. strategi greedy ini bersifat heuristic.
- v. Fungsi kelayakan
Fungsi kelayakan memeriksa apakah kandidat yang dipilih dari fungsi seleksi dapat dimasukkan ke dalam himpunan solusi atau tidak.
- vi. Fungsi obyektif
Fungsi yang memaksimumkan atau meminimumkan.

Algoritma greedy tidak selalu memberikan solusi yang optimum. Artinya jika algoritma greedy dibandingkan dengan algoritma pencarian exhaustive search, maka exhaustive search akan menghasilkan solusi yang lebih optimal. Namun, kompleksitas waktu dari algoritma greedy jauh lebih sederhana dibandingkan dengan exhaustive search. Sehingga algoritma greedy dapat digunakan untuk mencari solusi hampiran, bukan untuk mencari solusi terbaik mutlak.

Contoh-contoh persoalan yang diselesaikan dengan Algoritma Greedy :

- a. Persoalan penukaran uang
- b. Persoalan memilih aktivitas
- c. Minimisasi waktu di dalam sistem
- d. Persoalan knapsack
- e. Penjadwalan job dengan tenggat waktu
- f. Pohon merentang minimum (minimum spanning tree)
- g. Lintasan terpendek (shortest path)
- h. Kode Huffman (Huffman code)
- i. Pecahan Mesir (Egyptian fraction)

Contoh algoritma terkenal yang menggunakan strategi greedy adalah algoritma Dijkstra. Algoritma Dijkstra digunakan untuk menentukan lintasan terpendek. Strategi greedy pada algoritma Dijkstra adalah pada setiap langkah, pilih lintasan yang memiliki bobot minimum yang menghubungkan simpul yang sudah terpilih dengan simpul yang belum terpilih. Lintasan dari simpul asal ke simpul baru tersebut harus memiliki lintasan yang terpendek di antara semua lintasannya ke simpul-simpul yang belum terpilih.

B. Permainan Overdrive

Algoritma greedy dapat diaplikasikan pada permainan Overdrive. Overdrive adalah permainan lomba yang dilombakan tiap tahun. Untuk mengunduh dan mencoba permainan dapat dilakukan pada tautan berikut.

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

Permainan akan mempertandingkan dua bot dengan satu mobil tiap bot. Lintasan terdiri dari 4 jalur dan 1500 block. Mobil yang bisa mencapai garis finish lebih cepat adalah pemenangnya. Game runner akan meminta sebuah Command dari bot. Bot akan menghasilkan command berdasarkan situasi dalam permainan. Command tersebut terdiri dari :

- 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

Setiap mobil bisa melihat keadaan 20 block di depannya dan 5 block di belakangnya. Pada setiap block terdapat kondisi seperti memiliki mud, oil spill, wall, atau powerup.

Mobil memulai permainan dengan kecepatan sebesar 5. Kecepatan bisa turun jika menabrak wall, atau melewati mud. Kecepatan mobil bisa juga naik jika menggunakan command accelerate. Mobil juga bisa mendapat powerup yang bisa digunakan.

Untuk mengubah bot sesuai dengan keinginan, perlu diubah terlebih dahulu file game-config.json dan mengubah directory tempat bot yang digunakan. Kemudian buka file bot yang ada pada folder pilihan yang diinginkan contohnya file java yang didalamnya terdapat file bot.java. Setelah itu masukkan algoritma yang diinginkan pada file bot.

BAB III

APLIKASI STRATEGI GREEDY

1. Pemecahan Persoalan

Himpunan kandidat, C : Himpunan kandidat terdiri command yang akan dipilih.

Command tersebut terdiri dari :

- 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

Himpunan solusi, S : Command yang dipilih.

Fungsi solusi: Apakah sudah mencapai garis finish.

Fungsi seleksi (selection function): jika tidak kecepatan tidak maksimum, tambah kecepatan. Kemudian jika kecepatan mobil sudah maksimum, hindari command decelerate, minimumkan damage pada mobil, dan gunakan powerup.

Fungsi kelayakan (feasible): -

Fungsi obyektif : Memaksimalkan kecepatan mobil sampai kecepatan maksimum dan meminimalkan damage pada mobil. Jika mobil sudah mencapai kecepatan maksimum, gunakan powerup.

2. Alternatif Solusi

- a) Memaksimalkan speed. Beberapa karakteristik pemaksimalan speed dari bot adalah:
 - Bot tidak bisa sama sekali memilih command DECELERATE
 - Akan mengaktifkan Boost jika speed rendah (saat damage = 1)
- b) Meminimalisasikan pengurangan top speed. Beberapa karakteristik penjagaan top speed dari bot adalah:
 - Saat damage = 0, tidak akan mengaktifkan Boost jika di depan terdapat pengurangan speed dan tidak punya Lizard. Perlunya ada Lizard untuk mengaktifkan boost memiliki tujuan untuk menjaga agar speed tetap 15 minimal selama 2 round
 - Jika damage mencapai 2 atau lebih, mobil langsung diperbaiki hingga value damage nya dibawah 2.
 - Bot akan memilih lane yang memiliki pengurangan speed paling rendah, meskipun damage dari lane tersebut lebih tinggi. Hal ini karena tidak ada obstacle yang memberikan lebih banyak damage, namun ada obstacle yang perbedaan damage dan speed reductionnya signifikan. Contoh skenario adalah jika terdapat 2 lane, lane pertama memiliki 3 mud, sedangkan lane kedua memiliki 1 wall, maka mobil akan

memilih lane dengan 3 mud, karena pengurangan speed nya lebih rendah bila dibandingkan dengan 1 wall yang langsung mengurangi speed ke level terendah.

3. Analisis efektivitas alternatif solusi

Bot diprogram untuk lebih efektif/*greedy apabila* melawan musuh yang kompetitif. *Source code* yang dibuat agar bot menggunakan macam-macam *power-up* hanya akan aktif apabila musuh berada dalam map atau dekat dengan bot.

4. Solusi yang dipilih

Strategi *greedy* yang diimplementasikan berfokus pada pergerakan mobil (ACCELERATE, DECELERATE, TURN_LEFT, TURN_RIGHT, dll.) karena implementasi yang paling mudah. Variasi *power-up* yang banyak akan membuat kode program rumit untuk dijalankan.

BAB IV IMPLEMENTASI DAN PENGUJIAN

1. Implementasi Algoritma *Greedy*

```
function run(GameState gameState) -> Command
    Car myCar = gameState.player;
    Car opponent = gameState.opponent;

    //Basic fix logic
    List<Lane> frontBlocks = new ArrayList<>();
    frontBlocks = getBlocksInFront(myCar.position.lane, myCar.position.block,
gameState);

    int frontSpeedReduction;
    int leftSpeedReduction;
    int rightSpeedReduction;

    frontSpeedReduction = countSpeedReduction(frontBlocks, 15);
    if (myCar.position.lane > 1)
        List<Lane> leftBlocks = getBlocksOnLeft(myCar.position.lane,
myCar.position.block, gameState);
        leftSpeedReduction = countSpeedReduction(leftBlocks, 15);
    else
        List<Lane> leftBlocks = new ArrayList<>();
        leftSpeedReduction = 99;

    if (myCar.position.lane < 4)
        List<Lane> rightBlocks = getBlocksOnRight(myCar.position.lane,
myCar.position.block, gameState);
        rightSpeedReduction = countSpeedReduction(rightBlocks, 15);
    else
        List<Lane> rightBlocks = new ArrayList<>();
        rightSpeedReduction = 99;

    /*Fix jika Damage >= 2*/
    if (myCar.damage >= 2)
        -> FIX;
    else if (myCar.speed == 0)
        -> ACCELERATE;
    else
        /* speed < 15 */
        if (myCar.speed < 15)
            /* Jika tidak ada pengurangan speed di depan
```



```

        Tidak belok ke kiri ataupun ke kanan*/
    if (frontSpeedReduction == 0)
        /* Jika punya boost */
        if (hasPowerUp(PowerUps.BOOST, myCar.powerups))
            /* Jika damage = 0 dan punya Lizard ATAU damage = 1 dan speed = 3 */
            if ((myCar.damage == 0 && hasPowerUp(PowerUps.LIZARD,
myCar.powerups)) || (myCar.damage == 1 && myCar.speed == 3))
                -> BOOST;
            /* Jika damage = 0 dan tidak punya Lizard ATAU damage = 1 dan speed >
3 */

        else
            /* Jika speed belum maksimum */
            if (myCar.speed < maxSpeed)
                -> ACCELERATE;
            /* Jika speed sudah maksimum */
            else
                /* Jika musuh terlihat di map */
                if (opponent.position.block > myCar.position.block - 6 &&
opponent.position.block < myCar.position.block + 21)
                    /* Jika musuh di depan mobil player */
                    if (opponent.position.block > myCar.position.block)
                        /* Jika punya EMP, tembak EMP */
                        if (hasPowerUp(PowerUps.EMP, myCar.powerups) &&
opponent.position.lane != myCar.position.lane)
                            -> EMP;
                        /* Jika punya Tweet, prediksi posisi musuh dan letakkan
CyberTruck */

                    else if (hasPowerUp(PowerUps.TWEET, myCar.powerups))
                        Random rand1 = new Random();
                        int x1 = rand1.nextInt(3);
                        if (x1 == 0)
                            int placeLane = opponent.position.lane - 1;
                            int placeBlock = opponent.position.block + 1;
                            -> new TweetCommand(placeLane, placeBlock);
                        else if (x1 == 1)
                            int placeLane = opponent.position.lane;
                            int placeBlock = opponent.position.block + 1;
                            -> new TweetCommand(placeLane, placeBlock);
                        else
                            int placeLane = opponent.position.lane + 1;
                            int placeBlock = opponent.position.block + 1;
                            -> new TweetCommand(placeLane, placeBlock);

                        /* Jika tidak punya EMP ataupun Tweet */
                    else
                        -> DO_NOTHING;

```

```

        /* Jika musuh di belakang mobil player */
        else
            -> DO_NOTHING;

        /* Jika musuh tidak terlihat di map */
        else
            /* Jika musuh ada di belakang player dan player punya Oil, gunakan
oil */
            if (opponent.position.block < myCar.position.block &&
hasPowerUp(PowerUps.OIL, myCar.powerups))
                -> OIL;
            /* Jika musuh di depan player atau tidak punya Oil */
            else
                -> DO_NOTHING;
        /* Jika tidak punya Boost */
        else
            /* Jika speed belum maksimum, Accelerate */
            if (myCar.speed < Bot.maxSpeed)
                -> ACCELERATE;
            /* Jika speed sudah maksimum */
            else
                /* Jika musuh terlihat di map */
                if (opponent.position.block > myCar.position.block - 6 &&
opponent.position.block < myCar.position.block + 21)
                    /* Jika musuh ada di depan player */
                    if (opponent.position.block > myCar.position.block)
                        /* Jika punya EMP, tembak EMP */
                        if (hasPowerUp(PowerUps.EMP, myCar.powerups) &&
opponent.position.lane != myCar.position.lane)
                            -> EMP;
                        /* Jika punya Tweet, prediksi posisi musuh dan letakkan
CyberTruck */
                        else if (hasPowerUp(PowerUps.TWEET, myCar.powerups))
                            Random rand1 = new Random();
                            int x1 = rand1.nextInt(3);
                            if (x1 == 0)
                                int placeLane = opponent.position.lane - 1;
                                int placeBlock = opponent.position.block + 1;
                                -> new TweetCommand(placeLane, placeBlock);
                            else if (x1 == 1)
                                int placeLane = opponent.position.lane;
                                int placeBlock = opponent.position.block + 1;
                                -> new TweetCommand(placeLane, placeBlock);
                            else
                                int placeLane = opponent.position.lane + 1;

```

```

        int placeBlock = opponent.position.block + 1;
        -> new TweetCommand(placeLane, placeBlock);

        /* Jika tidak punya EMP ataupun Tweet */
        else
            -> DO_NOTHING;

        /* Jika musuh ada di belakang player */
        else
            -> DO_NOTHING;
        /* Jika musuh tidak terlihat di map */
        else
            /* Jika musuh ada di belakang player dan player punya Oil, gunakan oil
*/
            if (opponent.position.block < myCar.position.block &&
hasPowerUp(PowerUps.OIL, myCar.powerups))
                -> OIL;
            /* Jika musuh ada di depan player atau player tidak punya Oil */
            else
                -> DO_NOTHING;
        /* Jika ada speed reduction di depan */
        else
            /* Jika speed belum maksimum */
            if (myCar.speed < Bot.maxSpeed)
                /* Jika frontSpeedReduction paling rendah */
                if (frontSpeedReduction <= leftSpeedReduction && frontSpeedReduction <=
rightSpeedReduction)
                    -> ACCELERATE;
                /* Jika speed reduction di kiri atau kanan lebih rendah */
                else
                    /* Jika leftSpeedReduction lebih rendah, belok kiri, belok kiri */
                    if (leftSpeedReduction < rightSpeedReduction)
                        -> TURN_LEFT;
                    /* Jika speed reduction di kanan lebih rendah dari kiri, belok kanan */
                    else if (leftSpeedReduction > rightSpeedReduction)
                        -> TURN_RIGHT;
                    /* Jika speed reduction di kanan sama dengan speed reduction di kiri,
random antara kiri dan kanan */
                    else
                        Random rand1 = new Random();
                        int x1 = rand1.nextInt(2);
                        if (x1 == 0)
                            -> TURN_LEFT;
                        else
                            -> TURN_RIGHT;
                /* Jika speed sudah maksimum */

```

```

else
    if (frontSpeedReduction < leftSpeedReduction && frontSpeedReduction <
rightSpeedReduction)
        /* Jika musuh terlihat di map */
        if (opponent.position.block > myCar.position.block - 6 &&
opponent.position.block < myCar.position.block + 21)
            /* Jika musuh ada di depan player */
            if (opponent.position.block > myCar.position.block)
                /* Jika punya EMP, tembak EMP */
                if (hasPowerUp(PowerUps.EMP, myCar.powerups) &&
opponent.position.lane != myCar.position.lane)
                    -> EMP;
                /* Jika punya Tweet, prediksi posisi musuh dan letakkan
CyberTruck */
            else if (hasPowerUp(PowerUps.TWEET, myCar.powerups))
                Random rand1 = new Random();
                int x1 = rand1.nextInt(3);
                if (x1 == 0)
                    int placeLane = opponent.position.lane - 1;
                    int placeBlock = opponent.position.block + 1;
                    -> new TweetCommand(placeLane, placeBlock);
                else if (x1 == 1)
                    int placeLane = opponent.position.lane;
                    int placeBlock = opponent.position.block + 1;
                    -> new TweetCommand(placeLane, placeBlock);
                else
                    int placeLane = opponent.position.lane + 1;
                    int placeBlock = opponent.position.block + 1;
                    -> new TweetCommand(placeLane, placeBlock);

                /* Jika tidak punya EMP ataupun Tweet */
            else
                -> DO_NOTHING;

            /* Jika musuh ada di belakang player */
        else
            -> DO_NOTHING;

        /* Jika musuh tidak terlihat di map */
    else
        /* Jika musuh ada di belakang player dan player punya Oil, gunakan oil
*/
        if (opponent.position.block < myCar.position.block &&
hasPowerUp(PowerUps.OIL, myCar.powerups))
            -> OIL;
        /* Jika musuh ada di depan player atau player tidak punya Oil */

```

```

        else
            -> DO_NOTHING;
    else
        if (leftSpeedReduction < rightSpeedReduction)
            -> TURN_LEFT;
        else if (leftSpeedReduction > rightSpeedReduction)
            -> TURN_RIGHT;
        else
            Random rand3 = new Random();
            int x3 = rand3.nextInt(2);
            if (x3 == 0)
                -> TURN_LEFT;
            else
                -> TURN_RIGHT;
    else
        if (leftSpeedReduction == 0 && rightSpeedReduction == 0)
            Random rand2 = new Random();
            int x2 = rand2.nextInt(2);
            if (x2 == 0)
                -> TURN_LEFT;
            else
                -> TURN_RIGHT;

        else if (leftSpeedReduction > 0 && rightSpeedReduction == 0)
            -> TURN_RIGHT;
        else if (leftSpeedReduction == 0 && rightSpeedReduction > 0)
            -> TURN_LEFT;
        else
            if (hasPowerUp(PowerUps.LIZARD, myCar.powerups))
                -> LIZARD;
            else
                if (frontSpeedReduction <= leftSpeedReduction && frontSpeedReduction <=
rightSpeedReduction)
                    /* Jika musuh terlihat di map */
                    if (opponent.position.block > myCar.position.block - 6 &&
opponent.position.block < myCar.position.block + 21)
                        /* Jika musuh ada di depan player */
                        if (opponent.position.block > myCar.position.block)
                            /* Jika punya EMP, tembak EMP */
                            if (hasPowerUp(PowerUps.EMP, myCar.powerups) &&
opponent.position.lane != myCar.position.lane)
                                -> EMP;
                            /* Jika punya Tweet, prediksi posisi musuh dan letakkan
CyberTruck */
                                else if (hasPowerUp(PowerUps.TWEET, myCar.powerups))
                                    Random rand1 = new Random();

```

```

        int x1 = rand1.nextInt(3);
        if (x1 == 0)
            int placeLane = opponent.position.lane - 1;
            int placeBlock = opponent.position.block + 1;
            -> new TweetCommand(placeLane, placeBlock);
        else if (x1 == 1)
            int placeLane = opponent.position.lane;
            int placeBlock = opponent.position.block + 1;
            -> new TweetCommand(placeLane, placeBlock);
        else
            int placeLane = opponent.position.lane + 1;
            int placeBlock = opponent.position.block + 1;
            -> new TweetCommand(placeLane, placeBlock);

        /* Jika tidak punya EMP ataupun Tweet */
        else
            -> DO_NOTHING;

        /* Jika musuh ada di belakang player */
        else
            -> DO_NOTHING;
        /* Jika musuh tidak terlihat di map */
        else
            /* Jika musuh ada di belakang player dan player punya Oil, gunakan oil
*/
            if (opponent.position.block < myCar.position.block &&
hasPowerUp(PowerUps.OIL, myCar.powerups))
                -> OIL;
            /* Jika musuh ada di depan player atau player tidak punya Oil */
            else
                -> DO_NOTHING;
        else
            if (leftSpeedReduction < rightSpeedReduction)
                -> TURN_LEFT;
            else if (leftSpeedReduction > rightSpeedReduction)
                -> TURN_RIGHT;
            else
                Random rand3 = new Random();
                int x3 = rand3.nextInt(2)
                if (x3 == 0)
                    -> TURN_LEFT
                else
                    -> TURN_RIGHT

        /* fungsi untuk mengecek apakah mobil memiliki powerup tertentu */
        /* fungsi merupakan modifikasi dari fungsi yang ada pada reference bot */

```

```

function hasPowerUp(PowerUps powerUpToCheck, PowerUps[] available) -> Boolean
    i traversal (PowerUps powerUp: available)
        if (powerUp.equals(powerUpToCheck))
            -> true;
        -> false;

```

```

/* scan 15 blok di depan mobil */

```

```

/* mengembalikan list yang berisi blok di depan mobil */

```

```

function getBlocksInFront(int lane, int block, GameState gameState) -> List[Lane]
    List<Lane[]> map <- gameState.lanes
    List<Lane> blocks <- new ArrayList<>()
    int startBlock1 <- map.get(0)[0].position.block;

    Lane[] laneList <- map.get(lane - 1)
    I traversal (block - startBlock1 + 1, block - startBlock1 + 16)
        if (laneList[i] = null || laneList[i].terrain = Terrain.FINISH)
            break
        blocks.add(laneList[i])
    -> blocks

```

```

/* scan 15 blok dari kiri mobil */

```

```

function getBlocksInLeft(int lane, int block, GameState gameState) -> List[Lane]
    List<Lane[]> map <- gameState.lanes
    List<Lane> blocks <- new ArrayList<>()
    int startBlock1 <- map.get(0)[0].position.block;

    Lane[] laneList <- map.get(lane - 2)
    I traversal (block - startBlock1 + 1, block - startBlock1 + 15)
        if (laneList[i] = null || laneList[i].terrain = Terrain.FINISH)
            break
        blocks.add(laneList[i])
    -> blocks

```

```

/* scan 15 blok dari kanan mobil */

```

```

function getBlocksInLeft(int lane, int block, GameState gameState) -> List[Lane]
    List<Lane[]> map <- gameState.lanes
    List<Lane> blocks <- new ArrayList<>()
    int startBlock1 <- map.get(0)[0].position.block;

    Lane[] laneList <- map.get(lane )
    I traversal (block - startBlock1 + 1, block - startBlock1 + 15)
        if (laneList[i] = null || laneList[i].terrain = Terrain.FINISH)
            break
        blocks.add(laneList[i])
    -> blocks

```

```
function getSpeedState(int maxSpeed) -> int
```

```
  int speedState <- 0
  switch (maxSpeed) {
    case 15:
      speedState <- 5
      break
    case 9:
      speedState <- 4
      break
    case 8:
      speedState <- 3
      break
    case 6:
      speedState <- 2
      break
    case 5:
      speedState <- 1
      break
    case 3:
      speedState <- 1
      break
  }
  -> speedState
```

```
/* menghitung banyaknya blok yang bisa menurunkan kecepatan mobil */
```

```
function countSpeedReduction(List<Lane> blocks, int maxSpeed) -> int
```

```
  int total <- 0
  int tempMaxSpeed <- maxSpeed
  if (maxSpeed > blocks.size())
    tempMaxSpeed = blocks.size()

  i traversal (0..tempMaxSpeed)
    if (blocks.get(i).terrain == Terrain.MUD || blocks.get(i).terrain == Terrain.OIL_SPILL)
      total += 1
    else if (blocks.get(i).terrain == Terrain.WALL || blocks.get(i).isOccupiedByCyberTruck)
      total += 5
    if (total > 5)
      total = 5;
  if (getSpeedState(maxSpeed) - total == 2 && total != 0)
    total += 1
  else if (getSpeedState(maxSpeed) - total < 1)
    total = getSpeedState(maxSpeed) - 1
  -> total
```

2. Struktur Data yang Digunakan

Algoritma greedy diimplementasi kan pada file bot.java, file ini merupakan bot berbentuk class yang diintansiasi dalam file main.java. Dalam file main, akan didapatkan gamestate dari game engine dan melakukan method run pada bot. Argumen tersebut berbentuk sebuah gamestate untuk dibaca oleh objek bot. Bot kemudian melakukan algoritma greedy sesuai dengan gamestate diberikan.

3. Analisis Efisiensi Strategi yang Digunakan

Dengan algoritma yang sudah dibuat, bot seharusnya berfungsi dengan semestinya.

Penggunaan pilihan metoda greedy akan membuat bot lebih efisien untuk memenangkan permainan.

BAB V

KESIMPULAN DAN SARAN

Algoritma greedy dapat digunakan untuk memecahkan banyak masalah, salah satunya adalah untuk membuat bot dalam permainan overdrive. Walaupun algoritma greedy tidak menghasilkan solusi yang terbaik seperti pada brute force exhaustive search, kompleksitas waktu algoritma greedy lebih sederhana. Dalam permainan overdrive ini, strategi greedy yang digunakan adalah memaksimalkan kecepatan mobil mencapai kecepatan maksimum. Selain itu, meminimalkan kerusakan mobil. Dengan demikian mobil bisa cepat sampai garis finish.

DAFTAR PUSTAKA

[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)

[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)

[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)

Tautan menuju source code :

<https://github.com/ilhamwibowo/Tubes-1-STIMA>

Video demonstrasi program:

[YouTubehttps://www.youtube.com/watch?v=yBDupUubbaI](https://www.youtube.com/watch?v=yBDupUubbaI)