Penerapan Algoritma Greedy untuk Memenangkan Permainan Buckshot Roulette

Salsabiila - 13522062
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13522062@std.stei.itb.ac.id

***Abstract*— Makalah ini membahas penerapan algoritma greedy dalam permainan Buckshot Roulette, sebuah permainan meja indie bergenre horor. Buckshot Roulette menuntut pemain untuk mengambil keputusan optimal di setiap giliran dengan tujuan mengurangi nyawa dealer sebanyak mungkin untuk memenangkan permainan. Algoritma greedy dipilih karena kemampuannya dalam mencari solusi optimum lokal pada setiap tahap permainan dengan harapan mencapai solusi optimum global. Implementasi algoritma ini dilakukan dengan mendefinisikan setiap elemen yang diperlukan, seperti himpunan kandidat, himpunan solusi, fungsi solusi, fungsi seleksi, fungsi kelayakan, dan fungsi objektif. Pengujian menunjukkan bahwa strategi algoritma greedy yang digunakan berhasil memenangkan permainan dengan tingkat kemenangan sebesar 56% setelah dilakukan 25 kali percobaan. Meskipun demikian, strategi ini memiliki kelemahan berupa potensi pemborosan item dan tidak menjamin solusi optimal. Penelitian ini menyimpulkan bahwa algoritma greedy dapat menjadi strategi yang efektif untuk memenangkan permainan Buckshot Roulette, namun masih memiliki ruang untuk pengembangan lebih lanjut guna meningkatkan efisiensi dan efektivitas.**

Keywords—Buckshot Roulette; algoritma greedy; optimalisasi

I. PENDAHULUAN

Buckshot Roulette adalah permainan video meja yang menggabungkan elemen strategi dan keberuntungan, terinspirasi dari Russian Roulette yang terkenal. Permainan ini cukup populer karena mekaniknya yang simpel dan mudah dimengerti, dapat dimainkan berulang kali, serta waktu main yang sangat singkat. Pada permainan ini, pemain tidak hanya memiliki pilihan untuk menembak diri sendiri atau lawan, tetapi juga terdapat berbagai item yang dapat digunakan untuk membuat kemungkinan memenangkan permainan ini lebih besar.

Karena permainan Buckshot Roulette yang dimainkan per-ronde dan setiap rondenya pemain dan seorang *dealer* secara bergantian memiliki giliran untuk menentukan aksi yang akan diambil, sangat mungkin untuk permainan ini dimenangkan dengan cara selalu mencoba untuk mengambil aksi yang paling menguntungkan pada suatu giliran pemain. Secara formal, pendekatan ini dapat dilakukan dengan menggunakan strategi algoritma *greedy* yang dalam penentuan solusi akhirnya menggunakan pendekatan tersebut.

Implementasi algoritma greedy dalam permainan ini memberikan pendekatan yang terstruktur dan sistematis untuk mencapai solusi yang paling optimal. Dengan mempertimbangkan setiap aksi yang paling menguntungkan pada setiap giliran, pemain dapat memaksimalkan peluang kemenangan. Algoritma ini bekerja dengan cara mengambil keputusan berdasarkan kondisi saat ini dan memprioritaskan langkah yang memberikan hasil terbaik pada saat itu. Meskipun tidak selalu menghasilkan solusi optimal secara global, strategi ini dapat meningkatkan efisiensi permainan dan membantu pemain dalam membuat keputusan yang lebih baik dan lebih cepat.

II. DASAR TEORI

A. Permainan Buckshot Roulette

Buckshot Roulette merupakan permainan meja *(tabletop video game)* indie singkat bergenre horor yang dibuat dan dipublikasikan oleh Mike Klubnika di platform itch.io pada 28 Desember 2023 dan baru pada 4 April 2024 dirilis di platform Steam oleh Critical Reflex. Permainan ini memiliki waktu main yang cukup singkat, yakni sekita 15-20 menit. Pada versi terbarunya, terdapat dua jenis mode permainan, mode *story* dan mode *double or nothing*. Dalam penulisan makalah ini, cakupan mode yang akan dibahas hanya terbatas pada mode *story* yang merupakan mode standar pada permainan ini.

Gambar 1. Tampilan Permainan

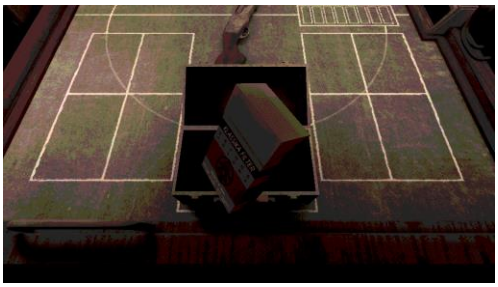
Sumber: Buckshot Roulette, diambil oleh penulis

Gambar 2. Tampilan Peluru dan *Charges*

Sumber: Buckshot Roulette, diambil oleh penulis

Dalam mode *story*, terdapat tiga ronde yang harus dimenangkan oleh pemain. Setiap rondenya, *shotgun* akan diisi dengan 2 jenis peluru, *live round* dan *blank round* dengan jumlah yang berbeda dan urutan yang tidak menentu. Jika pemain ataupun *dealer* ditembak dengan *live round*, maka *charge* atau nyawa dari pemain atau *dealer* akan berkurang sebanyak satu, namun jika peluru yang ditembakkan adalah *blank round*, maka tidak akan terjadi apapun. Pada ronde pertama, pemain dan *dealer* memiliki dua *charges* atau nyawa dan tidak terdapat item yang dapat digunakan dan baru pada ronde kedua penggunaan item dimulai. Berikut merupakan daftar item yang dapat digunakan beserta efek yang diberikan.

1. *Cigarettes*



Gambar 3. Tampilan Item *Cigarettes*

Sumber: Buckshot Roulette, diambil oleh penulis

Setelah menggunakan item ini, maka jumlah *charge* yang dimiliki penggunanya akan bertambah. Item ini tidak akan memiliki efek jika *charge* yang dimiliki sudah berjumlah maksimum.

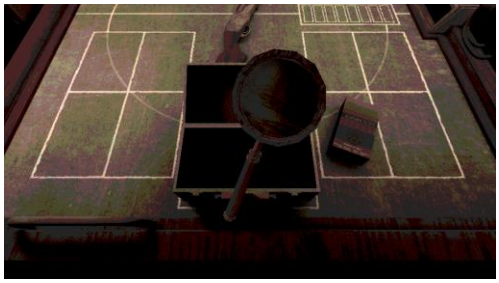
2. *Beer*

Gambar 4. Tampilan Item *Beer*

Sumber: Buckshot Roulette, diambil oleh penulis

Setelah menggunakan item ini, maka peluru saat ini akan dikeluarkan tanpa menggunakannya. Jika digunakan pada peluru terakhir, maka sesi permainan saat ini akan selesai.

3. *Magnifying Glass*

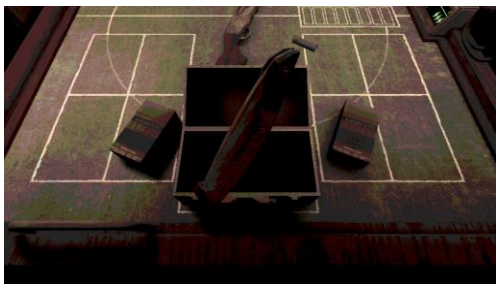


Gambar 5. Tampilan Item *Magnifying Glass*

Sumber: Buckshot Roulette, diambil oleh penulis

Setelah menggunakan item ini, maka penggunanya dapat melihat peluru saat ini yang terdapat dalam ruang peluru shotgun.

4. *Hand saw*



Gambar 6. Tampilan Item *Hand Saw*

Sumber: Buckshot Roulette, diambil oleh penulis

Setelah menggunakan item ini, jika jenis peluru yang ditembakkan merupakan *live round*, maka jumlah *charge* dari pemain yang terkena tembakannya akan berkurang sebanyak dua. Namun, item ini tidak memiliki efek apa pun jika peluru yang ditembakkan merupakan *blank round*.

5. *Handcuffs*

Gambar 7. Tampilan Item *Handcuffs*

Sumber: Buckshot Roulette, diambil oleh penulis

Setelah item ini digunakan, lawan dari pengguna akan diborgol dan tidak dapat melakukan apapun pada

giliran selanjutnya atau dengan kata lain, pemain yang diborgol akan melewatkan gilirannya sebanyak satu kali. Item ini tidak dapat digunakan kembali pada pemain yang sudah diborgol.

Penggunaan item dapat dilakukan hingga ronde terakhir, yaitu ronde ketiga. Pada ronde ini, pemain dan *dealer* masing-masing memiliki empat *charges* normal dan dua *faded charges*. Jika setelah permainan berlangsung tinggal tersisa satu *faded charge*, maka semua *damage* yang dilakukan pada pemain tersebut akan bersifat *lethal*.

Pada setiap giliran di tiap ronde, pemain dapat memilih salah satu dari dua pilihan, menembak lawan atau menembak diri sendiri. Jika pemain memilih untuk menembak lawan, maka giliran pemain akan diakhiri, namun jika memilih untuk menembak diri sendiri dan peluru yang ditembakkan adalah *blank round* maka giliran pemain akan tetap berlanjut. Seluruh peraturan permainan ini berlaku terhadap pemain dan juga *dealer* sehingga dibutuhkan strategi yang tepat untuk memenangkan permainan ini.

B. Algoritma Greedy

Algoritma Greedy adalah algoritma dengan hasil solusi yang pada suatu tahap atau langkah dalam pemecahan masalah dianggap paling optimal tanpa memedulikan solusi optimal global [1]. Pada algoritma ini, setiap pilihan yang diambil pada setiap tahap atau langkah harus merupakan solusi optimum lokal, meskipun pilihan yang diambil belum tentu dapat memberikan solusi akhir yang optimal. Prinsip dari algoritma ini adalah mencari optimum lokal pada setiap tahapnya dengan harapan dapat mengantarkan ke pada solusi optimum global. Algoritma ini biasanya digunakan untuk permasalahan-permasalahan yang memiliki sub struktur yang dapat diselesaikan satu per satu untuk mendapat solusi optimum lokal.

Algoritma *greedy* memiliki enam elemen pembentuknya, yaitu [2]:

1. Himpunan kandidat (C)

   Himpunan kandidat adalah elemen kandidat yang akan dipilih pada setiap langkahnya untuk masuk ke dalam himpunan solusi.

2. Himpunan solusi (S)

   Himpunan solusi terdiri dari elemen himpunan kandidat yang sudah terpilih.

3. Fungsi solusi

   Fungsi solusi merupakan fungsi yang menentukan apakah himpunan kandidat yang dipilih dapat memberikan solusi atau tidak.

4. Fungsi seleksi

   Fungsi seleksi merupakan fungsi yang menyeleksi apakah suatu elemen dapat masuk ke dalam himpunan kandidat atau tidak berdasarkan suatu startegi *greedy* tertentu yang umumnya bersifat heuristik.

5. Fungsi kelayakan

   Fungsi kelayakan merupakan fungsi yang memeriksa apakah elemen dari himpunan kandidat dapat masuk ke dalam himpunan solusi atau tidak.

6. Fungsi objektif

   Fungsi objektif merupakan fungsi yang memeriksa apakah solusi yang dipilih memberikan hasil minimum atau maksimum.

III. Hasil Penelitian

Dalam implementasi algoritma *greedy* untuk memenangkan permainan Buckshot Roulette, diperlukan definisi untuk setiap elemen dari algoritma tersebut. Berikut merupakan definisi dari setiap elemen untuk strategi algoritma *greedy by attack* yang memprioritaskan mengurangi jumlah *charges* milik dealer.

1. Himpunan kandidat: aksi-aksi yang dapat dilakukan pemain pada saat gilirannya.
2. Himpunan solusi: aksi yang terpilih.
3. Fungsi solusi: memeriksa apakah hasil aksi yang dipilih tidak mengurangi jumlah *charge* pemain dan/atau mengurangi jumlah *charge dealer*.
4. Fungsi seleksi: memilih aksi yang dapat mengurangi jumlah *charge* milik *dealer* semaksimal mungkin.
5. Fungsi kelayakan: memeriksa apakah aksi yang dilakukan mengurangi jumlah *charge* milik *dealer*.
6. Fungsi objektif: memenangkan permainan Buckshot Roulette.

A. Implementasi Program

Berikut merupakan hasil implementasi program untuk menentukan langkah di setiap giliran pemain dengan menggunakan algoritma *greedy* yang ditulis dalam bahasa Python.

```
# Initialize variables with state starting from the first round

round = 1

item_quantity = [0, 0, 0, 0, 0] # Order of items: cigarettes, beer, magnifying glass, hand saw, handcuffs

player_charges = 2

dealer_charges = 2

n_live_rounds = 1

n_blank_rounds = 2

max_charges = 2

current_round = None

curr_player = 1

is_game_finished = False

is_round_over = False

is_cuffed = 0


# Function to shoot dealer with 1 damage
```

```
def shoot_dealer(type, dmg):
    global dealer_charges, curr_player, is_game_finished, is_round_over, n_blank_rounds, n_live_rounds, is_cuffed

    print("Shoot the dealer!")
    if type == 1:
        dealer_charges -= dmg
        curr_player = 0
        n_live_rounds -= 1
    else:
        print("1. live round")
        print("2. blank round")
        bullet_shot = int(input("Enter the bullet shot: "))
        if bullet_shot == 1:
            dealer_charges -= dmg
            n_live_rounds -= 1
        else:
            n_blank_rounds -= 1

    if is_cuffed > 0:
        is_cuffed -= 1
    curr_player = 0

    if round == 3 and dealer_charges == 2:
        dealer_charges -= 1 # Dealer's charges will be set to 1 if on the 3rd round all of it doesn't have any more normal charges

    if is_round_finished() == 1:
        is_game_finished = True
    elif is_round_finished() == 0:
        is_round_over = True

def shoot_player(type=1):
    global player_charges, curr_player, is_game_finished, is_round_over, n_live_rounds, n_blank_rounds

    print("Shoot yourself!")
    if type != 1:
        print("1. live round")
        print("2. blank round")
        bullet_shot = int(input("Enter the bullet shot: "))
        if bullet_shot == 1:
            player_charges -= 1
            n_live_rounds -= 1
        else:
            n_blank_rounds -= 1
    else:
        n_blank_rounds -= 1

    curr_player = 1

    if is_round_finished() == 1:
        is_game_finished = True
    elif is_round_finished() == 0:
        is_round_over = True

# Function to check if player can use cigarettes if player charges are not at max, ensuring player is not dying
def check_player_charges():
    global player_charges, item_quantity

    if player_charges < max_charges and item_quantity[0] > 0:
        count = 0
        while player_charges < max_charges and item_quantity[0] > 0:
            player_charges += 1
            item_quantity[0] -= 1
            count += 1
        print(f"Use item: cigarettes x {count}")

# Function to use hand saw to increase damage, made to its own function to avoid redundancy
def use_handsaw(type, dmg):
    print("Use item: hand saw")
    item_quantity[3] -= 1
    shoot_dealer(type, dmg)

# Function to check if the round or game is finished
def is_round_finished():
    global round, max_charges, player_charges, dealer_charges, n_blank_rounds, n_live_rounds, is_cuffed, item_quantity

    if dealer_charges <= 0:
        if round < 3:
            print()
            round += 1
            print(f"=============== ROUND {round} ===============")
```

```
            max_charges = int(input("Enter the maximum charges for each players: "))
            player_charges = max_charges
            dealer_charges = max_charges
            n_live_rounds = 0
            n_blank_rounds = 0
            is_cuffed = 0
            item_quantity = [0, 0, 0, 0, 0]

            return 0
        else:
            print()
            print(f"===============      YOU      WIN ===============")
            return 1
    elif player_charges <= 0:
        print()
        print(f"===============       GAME       OVER ===============")
        return 1
    return -1


# Function to determine the best move using greedy strategy focusing on attacking the dealer
def greedy_by_attack():
    global       n_live_rounds,       n_blank_rounds, current_round, curr_player, item_quantity, is_cuffed

    if n_live_rounds > 0 and n_blank_rounds == 0:
        if item_quantity[3] > 0:
            use_handsaw(0, 2)
        else:
            shoot_dealer(1, 1)
    elif n_live_rounds > 0:
        # Use handcuffs to reduce dealer's turn if there are live rounds
        if item_quantity[4] > 0 and n_live_rounds > 0 and is_cuffed == 0:
            print("Use item: handcuffs")
            item_quantity[4] -= 1
            is_cuffed = 2


        # Prioritize knowing the current round type inside the shotgun:
        if item_quantity[2] > 0: # If there is a magnifying glass
            print("Use item: magnifying glass")
            item_quantity[2] -= 1
            print("1. Live round")
            print("2. Blank round")
            current_round = int(input("Enter the current round type inside the shotgun: "))

            # If the current round is a live round
            if current_round == 1:
                # If there is a hand saw, use it to increase damage
                if item_quantity[3] > 0:
                    use_handsaw(1, 2)
                else:
                    shoot_dealer(1, 1)
            else:
                # Shoot yourself to keep the next turn with the player
                shoot_player(1)

        else: # If there is no magnifying glass
            while item_quantity[1] > 0 and n_live_rounds < n_blank_rounds: # If there is beer, use it repeatedly to increase the chance of shooting a live round
                print("Use item: beer")
                print("1. Live round")
                print("2. Blank round")
                beer_round = int(input("Enter the round type ejected: "))
                if beer_round == 1:
                    n_live_rounds -= 1
                else:
                    n_blank_rounds -= 1
                item_quantity[1] -= 1
            if n_live_rounds >= n_blank_rounds and item_quantity[3] > 0:
                use_handsaw(0, 2)
            else:
                shoot_dealer(0, 1)
    else:
        shoot_player(1)


# Function to start playing the game
def start_playing():
    global n_live_rounds, n_blank_rounds, round, curr_player, player_charges, dealer_charges, is_game_finished, is_round_over, is_cuffed, item_quantity

    print()
    is_round_over = False
```

```
    if round != 1:
        print("1. Cigarettes")
        print("2. Beer")
        print("3. Magnifying Glass")
        print("4. Hand Saw")
        print("5. Handcuffs")
        item_input = input("Enter the received items: ")
        arr = list(map(int, item_input.split()))
        for i in arr:
            item_quantity[i - 1] += 1


    n_live_rounds = int(input("Enter the number of live rounds: "))
    n_blank_rounds = int(input("Enter the number of blank rounds: "))


    while n_live_rounds > 0 or n_blank_rounds > 0:
        check_player_charges()
        greedy_by_attack()
        if is_game_finished or is_round_over:
            return
        print (is_cuffed)
        if (n_live_rounds != 0 or n_blank_rounds != 0) and is_cuffed == 0:
            while curr_player == 0:
                if input("Is it your turn now? (y/n): ") == 'y':
                    n_live_rounds -= int(input("Enter the number of live rounds used: "))
                    n_blank_rounds -= int(input("Enter the number of blank rounds used: "))
                    player_charges = int(input("Enter the number of charges left for player: "))
                    dealer_charges = int(input("Enter the number of charges left for dealer: "))
                    curr_player = 1
        if is_round_finished() == 1:
            is_game_finished = True
            return
        elif is_round_finished() == 0:
            is_round_over = True
            return


# Main program
print("============== START GAME ==============")
print(f"==============         ROUND         {round} ==============")
while not is_game_finished:
    start_playing()
    if is_game_finished:
        break
```

B. Hasil dan Analisis Pengujian

Berikut merupakan salah satu contoh hasil keluaran program algoritma *greedy* yang telah dibuat:

```
============== START GAME ==============
============== ROUND 1 ==============

Enter the number of live rounds: 1
Enter the number of blank rounds: 2
Shoot the dealer!
1. live round
2. blank round
Enter the bullet shot: 2
Is it your turn now? (y/n): y
Enter the number of live rounds used: 1
Enter the number of blank rounds used: 0
Enter the number of charges left for player: 2
Enter the number of charges left for dealer: 1
Shoot yourself!

Enter the number of live rounds: 3
Enter the number of blank rounds: 2
Shoot the dealer!
1. live round
2. blank round
Enter the bullet shot: 2
Is it your turn now? (y/n): y
Enter the number of live rounds used: 1
Enter the number of blank rounds used: 0
Enter the number of charges left for player: 1
Enter the number of charges left for dealer: 1
Shoot the dealer!
1. live round
2. blank round
Enter the bullet shot: 1

============== ROUND 2 ==============
Enter the maximum charges for each players: 4

1. Cigarettes
2. Beer
3. Magnifying Glass
4. Hand Saw
```

```
5. Handcuffs
Enter the received items: 3 5
Enter the number of live rounds: 1
Enter the number of blank rounds: 1
Use item: handcuffs
Use item: magnifying glass
1. Live round
2. Blank round
Enter the current round type inside the shotgun: 2
Shoot yourself!
Shoot the dealer!

1. Cigarettes
2. Beer
3. Magnifying Glass
4. Hand Saw
5. Handcuffs
Enter the received items: 2 4
Enter the number of live rounds: 2
Enter the number of blank rounds: 2
Use item: hand saw
Shoot the dealer!
1. live round
2. blank round
Enter the bullet shot: 1
Is it your turn now? (y/n): n
Is it your turn now? (y/n): y
Enter the number of live rounds used: 1
Enter the number of blank rounds used: 1
Enter the number of charges left for player: 2
Enter the number of charges left for dealer: 2
Shoot yourself!

1. Cigarettes
2. Beer
3. Magnifying Glass
4. Hand Saw
5. Handcuffs
Enter the received items: 2 3
Enter the number of live rounds: 3
Enter the number of blank rounds: 2
Use item: magnifying glass
1. Live round
2. Blank round
Enter the current round type inside the shotgun: 1
Shoot the dealer!
Is it your turn now? (y/n): y
Enter the number of live rounds used: 1
Enter the number of blank rounds used: 1
Enter the number of charges left for player: 2
Enter the number of charges left for dealer: 0

=============== ROUND 3 ===============
Enter the maximum charges for each players: 6

1. Cigarettes
2. Beer
3. Magnifying Glass
4. Hand Saw
5. Handcuffs
Enter the received items: 4 2 1 4
Enter the number of live rounds: 1
Enter the number of blank rounds: 2
Use item: beer
1. Live round
2. Blank round
Enter the round type ejected: 1
Shoot the dealer!
1. live round
2. blank round
Enter the bullet shot: 2
Is it your turn now? (y/n): y
2. Beer
3. Magnifying Glass
4. Hand Saw
5. Handcuffs
Enter the received items: 5 1 3 5
Enter the number of live rounds: 4
Enter the number of blank rounds: 4
Use item: handcuffs
Use item: magnifying glass
1. Live round
2. Blank round
Enter the current round type inside the shotgun: 2
Shoot yourself!
Use item: hand saw
Shoot the dealer!
1. live round
2. blank round
Enter the bullet shot: 1
```

```
Use item: hand saw
Shoot the dealer!
1. live round
2. blank round
Enter the bullet shot: 2
Is it your turn now? (y/n): y
Enter the number of live rounds used: 2
Enter the number of blank rounds used: 1
Enter the number of charges left for player: 6
Enter the number of charges left for dealer: 5
Use item: handcuffs
Shoot the dealer!
1. live round
2. blank round
Enter the bullet shot: 2
Shoot the dealer!

1. Cigarettes
2. Beer
3. Magnifying Glass
4. Hand Saw
5. Handcuffs
Enter the received items: 5 3 5 3
Enter the number of live rounds: 3
Enter the number of blank rounds: 2
Use item: handcuffs
Use item: magnifying glass
1. Live round
2. Blank round
Enter the current round type inside the shotgun: 2
Shoot yourself!
Use item: magnifying glass
1. Live round
2. Blank round
Enter the current round type inside the shotgun: 2
Shoot yourself!
Shoot the dealer!
Shoot the dealer!
Is it your turn now? (y/n): y
Enter the number of live rounds used: 1
Enter the number of blank rounds used: 0
Enter the number of charges left for player: 4
Enter the number of charges left for dealer: 1

1. Cigarettes
2. Beer
3. Magnifying Glass
4. Hand Saw
5. Handcuffs
Enter the received items: 2 2 2 5
Enter the number of live rounds: 4
Enter the number of blank rounds: 2
Use item: cigarettes x 2
Use item: handcuffs
Shoot the dealer!
1. live round
2. blank round
Enter the bullet shot: 2
Shoot the dealer!
1. live round
2. blank round
Enter the bullet shot: 1

=============== YOU WIN ===============
```

Gambar 8. Tampilan Akhir Permainan

Sumber: Buckshot Roulette, diambil oleh penulis

Berdasarkan hasil pengujian, algoritma *greedy* yang digunakan telah terbukti dapat memenangkan permainan Buckshot Roulette meski dengan *win rate* yang cukup rendah, yaitu sebesar 56%. Hasil ini didapatkan setelah melakukan percobaan sebanyak 25 kali, dan dari percobaan tersebut algoritma ini berhasil menang sebanyak 14 kali.

Secara umum, algoritma ini sudah cukup baik dalam mencoba mendapatkan solusi optimum lokal di setiap giliran pemain dengan harapan dapat memenangkan permainan. Namun, masih terdapat banyak kekurangan dari menggunakan strategi ini, yakni karena strategi *greedy* yang dipilih adalah *greedy by attack* di mana pemain menggunakan seluruh sumber daya atau *resource* yang dimiliki untuk mengurangi *charge* milik *dealer* sebanyak-banyak dalam suatu giliran, sering kali terjadi pemborosan item. Selain itu, layaknya strategi algoritma *greedy* pada umumnya, strategi ini tidak menjanjikan solusi yang paling optimal.

IV. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa algoritma *greedy* dapat digunakan sebegai

salah satu solusi untuk memenangkan permainan Buckshot Roulette dengan *win rate* sebesar 56% dengan memilih solusi optimum lokal di setiap langkah penyelesaian dengan harapan dapat mencapai optimum global.

Meskipun tidak dapat menjamin bahwa solusi yang dihasilkan akan menjadi solusi yang optimal, strategi algoritma *greedy by attack* yang digunakan pada penelitian terbukti dapat menjadi strategi yang sangkil untuk memenangkan permainan ini dan masih memiliki banyak potensi dan ruang untuk berkembang serta menyempurnakan algoritma ini dengan mempertimbangkan lebih banyak aspek dari permainan untuk meningkatkan efisiensi dan efektivitas.

TAUTAN VIDEO YOUTUBE

Berikut merupakan tautan video youtube: https://youtu.be/bxddAIQDrkw

UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena berkat, rahmat dan karunia-Nya penulis dapat menyelesaikan makalah ini. Ucapan terima kasih juga penulis sampaikan kepada orang tua penulis dan dosen pengampu mata kuliah Strategi Algoritma untuk kelas K2, yaitu Dr. Nur Ulfa Maulidevi, S.T, M.Sc. Terakhir, penulis juga berterima kasih atas motivasi dan dukungan dari seluruh teman penulis baik selama penulisan makalah ini, maupun sehari-hari.

REFERENCES

[1] Y. Wang, "Review on greedy algorithm," Theoretical and natural science, vol. 14, no. 1, pp. 233–239, Nov. 2023, doi: https://doi.org/10.54254/2753-8818/14/20241041.

[2] Munir, Rinaldi (2021), Algoritma Greedy Bagian 1, Slide Kuliah IF 2211 Strategi Algoritma, Bandung, Indonesia.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Juni 2024

Salsabiila
13522062