

Penerapan Algoritma Greedy dalam Pengambilan Keputusan Penggunaan Senjata In Game Pada Permainan VALORANT

Fajari Bagas Imami 118140121
Program Studi Teknik Informatika
Jurusan Teknik Produksi dan Industri
Institut Teknologi Sumatera, Jalan terusan ryacudu
Fajari.118140121@student.itera.ac.id

Abstract— Dalam permainan FPS seperti VALORANT pemilihan senjata merupakan hal utama yang mempengaruhi aim seorang untuk meraih kemenangan. Namun pemilihan senjata pada tiap game nya akan dibatasi dengan kepemilikan credit pada game. Dalam makalah ini saya akan membahas penggunaan algoritma greedy untuk menentukan senjata yang dapat digunakan dalam in game.

Keywords—FPS, greedy, knapsack problem.

I. PENDAHULUAN

VALORANT merupakan salah satu game *First Person Shooter (FPS)*. Permainan ini dimainkan oleh 10 orang pemain dimana dari sisi lawan akan dibagi menjadi 5 melawan 5, permainan ini sangat mengandalkan aim dan strategi yang baik untuk bias memenangkan permainan.

Dalam permainan FPS diperlukannya senjata untuk memenangkan game, dalam pemilihan senjatapun dapat mempengaruhi aim dalam permainan yang membuat kita dapat mudah untuk menembak lawan

Salah satu solusi untuk membantu pemain dapat menggunakan rekomendasi penggunaan senjata untuk memenangkan permainan. Pada makalah ini akan membahas bagaimana rekomendasi penggunaan senjata menggunakan algoritma greedy

II. LANDASAN TEORI

A. Algoritma Greedy

Merupakan algoritma yang paling populer untuk digunakan dan algoritma greedy merupakan algoritma yang sederhana untuk memecahkan persoalan optimasi. Pada pencarian solusi optimasi dapat menentukan persoalan maksimasi dan minimasi. Algoritma Greedy biasanya tidak akan memberikan solusi yang optimal, namun Algoritma Greedy dapat menghasilkan “optimum lokal” yang dapat

mengaproksimasi solusi optimum sesungguhnya dari suatu persoalan.

B. Integer Knapsack Problem

Persoalan Knapsack adalah persoalan optimasi kombinatorial. Knapsack problem merupakan salah satu masalah yang membandingkan dari kapasitas bobot dari object. Setiap object pasti memiliki bobot (weight) dan keuntungan (profit) dari object tersebut dipilih cara terbaik untuk mendapatkan atau memperoleh total yang paling terbaik dari cara yang sederhana dan tidak boleh melebihi batas kapasitas yang ada.

disebut masalah knapsack 1/0 karena object dapat dimasukkan ke dalam ransel (1) atau tidak ada masukan sama sekali (0). Dari hal itu cara yang dilakukan knapsack problem yang pertama merupakan penentuan akan mengambil cara dengan menentukan akan memasukan object berdasarkan strategi heuristic seperti

- *greedy by profit*
pada strategi ini algoritma akan mencari object yang memiliki keuntungan terbesar

- *greedy by weight*
pada strategi ini algoritma akan mencari object yang memiliki berat teringan

- *greedy by density.*
pada strategi ini algoritma akan mencari object yang memiliki perbandingan dari keuntungan per unit nya.

Formulasi untuk persoalan 1/0 knapsack sendiri seperti berikut:

$$\text{Maksimasi } F = \sum_{i=1}^n p_i x_i$$

dengan kendala (*constraint*)

$$\sum_{i=1}^n w_i x_i \leq K$$

yang dalam hal ini, $x_i = 0$ atau $1, \quad i = 1, 2, \dots, n$

C. Fractional knapsack

Fractional knapsack problem biasa dikenal ransel pecahan merupakan carian dari knapsack akan tetapi hal ini memiliki solusi yang bisa memberikan berbentuk pecahan atau fraksi.

Beda hal nya dengan integer knapsack, fraksional knapsack dapat mengisi bobot dengan penuh tanpa menyisakan sisa bobot.

Formulasi untuk persoalan fraksional knapsack sendiri seperti berikut:

$$\text{Maksimasi } F = \sum_{i=1}^n p_i x_i$$

dengan kendala (*constraint*)

$$\sum_{i=1}^n w_i x_i \leq K$$

yang dalam hal ini, $0 \leq x_i \leq 1, \quad i = 1, 2, \dots, n$

III. IMPLEMENTASI

Dalam proses implementasi akan dijelaskan mengenai tahapan – tahapan implementasi untuk memberikan solusi dari pemilihan senjata pada in game VALORANT. Dari implementasi ini senjata pada VALORANT memiliki harga untuk dibeli yang dimana harga dikontekskan sebagai weighth dan tier yang telah dianalisis sebagai nilai dari senjata tersebut.

Berikut tier yang didapat untuk menjadi referensi nilai:



Gambar 1 Tier Senjata

sumber: <https://www.instagram.com/p/CY1vhV-J8rE/>

Berikut harga yang didapat untuk menjadi referensi weighth:



Gambar 1 Harga Senjata

sumber: valorant

a. Algoritma

Pada persoalan kali ini, tahapan implementasinya akan memiliki weight (w) sebagai harga senjata dan nilai (v) sebagai tier pada senjata. Kemudian menyimpan harga dan tier tersebut ke struct array. Pada struct dibagi menjadi primary gun dan secondary gun.

```
struct gun{
    string name;
    int tier;
    int credit;
};

struct gun secondary[4] = {
    {"shorty",4, 150},
    {"frenzy",3, 450},
    {"ghost",2, 500},
    {"sheriff",2, 800}};

struct gun primary[12] = {
    {"bucky",4, 850}, {"marshal",2, 950},
    {"stinger",4, 950}, {"specter",3, 1600},
    {"judge",2, 1650}, {"ares",1, 1650},
    {"bulldog",3, 2050}, {"guardian",3, 2250},
    {"phantom",1, 2900}, {"vandal",1, 2900},
    {"odin",2, 3200}, {"operator",1, 4700}};
```

Kemudian diinputkannya credit pada game untuk dijadikan batasan dari weighth yang diberikan.

Ketika batasan atau credit dari game telah didapat, dilakukannya pemilihan senjata dengan beberapa rekomendasi seperti gun by credit dan by value.

Pada proses by credit, yang pertama dilakukan melakukan looping untuk mencari primary gun dari harga yang terbesar maupun yang terkecil. Ketika mencari yang terbesar maupun terkecil maka primary hanya boleh diambil satu sebelum mengambil secondary terlebih dahulu. Proses ini dilakukan hingga mendapatkan senjata yang terbaik sehingga mengeluarkan output senjata rekomendasi.

Pada proses by value, proses yang dilakukan sama halnya dengan proses sebelumnya akan tetapi tier akan

menentukan pengambilan senjata dan senjata pada tier yang sama tidak akan merekomendasikan. Pengambilan akan ditentukan dengan harga yang sesuai dan tier yang paling terbaik kemudian dilakukan looping untuk mendapat senjata lainnya.

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

Algoritma:
S ← {} { inisialisasi S dengan kosong }
while (not SOLUSI(S) and (C ≠ {})) do
  x ← SELEKSI(C) { pilih sebuah kandidat dari C }
  C ← C - {x} { buang x dari C karena sudah dipilih }
  if LAYAK(S ∪ {x}) then { x memenuhi kelayakan untuk dimasukkan ke dalam himpunan solusi }
    S ← S ∪ {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
```

Gambar 3 Skema umum algoritma greedy

sumber: [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)

Dalam Proses pengujian dilakukan dengan memasukan dengan batas:

- Menggunakan credit yang memiliki nilai harga yang sama
- Menggunakan credit yang melebihi nilai harga
- Menggunakan credit yang kurang dari nilai harga

A. Percobaan menggunakan credit yang memiliki nilai harga yang sama

Masukan credit : 2900

```
by weight expensive
-----
vandal
by weight cheapest
-----
bucky
shorty
shorty
shorty
shorty
marshal
frenzy
```

B. Percobaan menggunakan credit yang melebihi nilai harga

Masukan credit : 5000

```
by weight expensive
-----
operator
shorty
by weight cheapest
-----
bucky
shorty
shorty
shorty
shorty
marshal
frenzy
frenzy
frenzy
frenzy
```

C. Percobaan menggunakan credit yang kurang dari nilai harga

Masukan credit : 100

```
by weight expensive
-----
by weight cheapest
-----
```

REFERENCES

- [1] [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). Diakses pada 27 Maret 2022..
- [2] Rehagana Kevin Christian Sembiring.2021. Penerapan Regular Expression (RegEx) pada Penyensoran Kata-Kata Tabu dalam In Game Chat Permainan VALORANT.Bandung:stei.itb
- [3] Muhammad Akram Al Bari.2021 .Penerapan Algoritma Greedy dalam Pengambilan Keputusan dengan Eisenhower Decision Matrix.bandung.stei.itb

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.

Bandar Lampung, 29 maret 2022

Fajarib

Fajari Bagas Imami 118140121