

Aplikasi Algoritma Greedy pada Sistem Pertandingan dalam Permainan Digimon Cyber Sleuth

Saul Sayers - 13520094

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13520094@std.stei.itb.ac.id

Abstract—Digimon Cyber Sleuth merupakan video game yang diterbitkan oleh Bandai Namco Entertainment sebagai bagian dari series digimon. Permainan yang memiliki genre RPG ini memiliki sistem pertandingan berbasis giliran antara pemain dengan bot ataupun pemain lain. Tiap pemain akan memiliki beberapa digimon yang dapat ditandingkan, dengan tiap langkah dapat memilih alternatif tindakan yang tersedia. Untuk meraih kemenangan dari suatu pertandingan, diperlukan suatu strategi yang mangkus dalam pemilihan tindakan untuk tiap langkahnya. Makalah ini akan membahas penggunaan algoritma greedy dalam membangun solusi pemilihan tindakan dalam sistem pertandingan permainan Digimon Cyber Sleuth. Perlu dicatat bahwa solusi yang dibangun bukanlah solusi optimum global, tetapi solusi yang dibentuk dengan aproksimasi sehingga cukup benar untuk batas waktu yang singkat dan sederhana.

Keywords—Greedy, Digimon, Cyber Sleuth, Giliran, Game

I. PENDAHULUAN

Digital Monsters atau yang disingkat dengan digimon merupakan waralaba asal Jepang yang diciptakan oleh Akiyoshi Hongo. Pada awalnya, digimon diluncurkan ke media sebagai serial animasi yang menceritakan mengenai sekumpulan anak yang memasuki dunia paralel yang tercipta dari jaringan komunikasi bumi dan berisi makhluk hidup berupa monster digital. Anak – anak tersebut kemudian berteman, atau juga bisa dibidang memelihara, beberapa digimon yang kemudian membantu mereka dalam berpetualang untuk kembali ke dunia asal mereka. Tengah petualangan mereka, akan ditemukan beberapa digimon musuh. Digimon musuh akan melakukan pertandingan dengan digimon yang dimiliki oleh tokoh utama untuk menghambat mereka mencapai tujuannya. Series ini kemudian berakhir dengan perpisahan antara tokoh utama dan digimon yang mereka miliki karena berhasil kembali ke dunia sendiri.

Faktanya, series digimon ini kemudian mendunia dan berhasil meraih popularitas yang cukup tinggi di kalangan anak – anak. Hal tersebut terjadi karena beberapa faktor, salah satunya adalah tokoh utama yang seumuran dengan mereka sehingga anak – anak dapat merasakan keterhubungan dengan kisah para tokoh. Faktor lain adalah konsep dari berteman digimon yang serupa dengan bagaimana anak – anak kecil membuat teman imajinasi untuk mengisi waktu luang mereka. Faktor yang terakhir adalah sistem evolusi yang disediakan pada series digimon. Beberapa musuh yang dilawan terkadang terlalu kuat dan tidak bisa dikalahkan. Untuk mengatasi hal

tersebut, digimon memiliki sistem evolusi untuk melakukan metamorfosis menjadi bentuk yang lebih besar atau kuat agar dapat mengalahkan musuh mereka. Hal tersebut memberi nilai kepada penonton bahwa mereka juga dapat berkembang dan menjadi lebih kuat menghadapi masalah apapun.

Larisnya series digimon ini mendorong pihak waralaba digimon untuk membuat beberapa *merchandise* dalam bentuk sequel anime, manga, mainan, boneka, trading cards, dan media lain. Namun, para penggemar mulai memiliki keinginan untuk merasakan sensasi berinteraksi ataupun memelihara digimon secara langsung. Seiring berkembangnya teknologi, permasalahan tersebut dapat diatasi dengan diciptakannya *merchandise* dalam bentuk video game terkait digimon. Pada awalnya, Series video game tersebut hanya terbatas bagaimana seseorang bisa memiliki sebuah digimon dan memeliharanya agar dapat berevolusi menjadi yang diinginkan. Konsep video game digimon tersebut mulai menarik tidak hanya anak – anak kecil saja, tetapi target pasarnya juga berubah menjadi untuk semua umur.

Sebagaimana manusia selalu memiliki rasa ketidakpuasan, timbul permasalahan baru di mana penggemar juga ingin menandingkan digimon yang mereka miliki. Pengembangan permainan selalu dilakukan, hingga berhasil diluncurkan beberapa permainan digimon yang dapat memiliki fitur pertandingan. Dari antara puluhan permainan digimon tersebut, salah satunya adalah Digimon Cyber Sleuth. Cyber Sleuth merupakan video game digimon paling kini yang diluncurkan pada tahun 2015 dan menawarkan fitur yang cukup lengkap. Mulai dari petualangan, pemburuan dan koleksi digimon, pemeliharaan dan pelatihan digimon, evolusi digimon, hingga sistem pertandingan digimon yang dapat dilakukan dengan bot ataupun pemain lain secara online.

Sistem pertandingan yang dimiliki akan menandingkan sekumpulan digimon yang dimiliki oleh seorang pemain dengan pemain lain. Sistem pertandingan yang tersedia ini dilakukan dengan berbasis giliran untuk tiap digimon-nya. Untuk tiap giliran, pemain akan diminta untuk memilih suatu tindakan dari beberapa tindakan yang tersedia. Namun, banyaknya alternatif tindakan yang dapat dipilih cukup banyak sehingga proses pemilihan tindakan menjadi cukup kompleks. Salah satu cara untuk dapat memperoleh solusi optimum dalam pemilihan tindakan tersebut adalah dengan menggunakan pendekatan algoritma greedy. Agar daftar

tindakan yang tersedia konsisten, maka harus dipastikan bahwa versi permainan adalah complete edition.



Gambar 1 : Versi Permainan yang Digunakan dan Dianalisis dalam Makalah

Sumber :

<https://fs-prod-cdn.nintendo-europe.com/>

Diakses pada 19 Mei 2022 pukul 03.39 WIB

II. TEORI DASAR

A. Algoritma Greedy

Algoritma greedy adalah algoritma yang digunakan untuk memecahkan masalah heuristik dengan cara membangun solusi langkah per langkah. Untuk tiap pilihan langkah dalam algoritma Greedy, akan diambil langkah yang terlihat paling optimal dan menguntungkan tanpa melihat konsekuensi kedepannya. Strategi ini tidak mempertimbangkan gambaran besar permasalahan sehingga hanya mendapatkan solusi optimum lokal saja. Pada akhir algoritma Greedy, tiap alternatif solusi optimum lokal akan digabungkan dengan harapan memperoleh solusi optimum global.

Sesuai dengan namanya yang apabila diterjemahkan berarti rakus, prinsip utama dari algoritma ini adalah *"take what you can get now!"*. Arti dari kalimat tersebut adalah keputusan yang diambil saat menerapkan algoritma greedy adalah keputusan yang optimal dan paling menguntungkan pada langkah tersebut dengan harapan tercapai optimum global. Solusi yang dibangun menggunakan algoritma ini cenderung tidak menghasilkan solusi yang paling optimal. Akan tetapi, algoritma ini berhasil membangun solusi yang cukup mendekati solusi optimum global dalam waktu dan tingkat kesulitan yang cukup wajar.

Algoritma ini memiliki beberapa elemen umum, yaitu :

1. Himpunan kandidat (C): Himpunan yang berisi elemen kandidat solusi (dapat berupa simpul ataupun sisi dalam sebuah graf).
2. Himpunan solusi (S): Himpunan yang berisi kandidat solusi terpilih.
3. Fungsi solusi: Fungsi yang menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi.
4. Fungsi seleksi: Fungsi yang memilih kandidat berdasarkan strategi greedy tertentu (bersifat heuristik).
5. Fungsi kelayakan: Fungsi yang mempertimbangkan kelayakan kandidat yang dipilih untuk dapat dimasukkan ke dalam himpunan solusi.

6. Fungsi obyektif: Fungsi yang mengoptimalkan solusi.

```
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 2 : 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)

Diakses pada 19 Mei 2022 pukul 01.51 WIB

B. Sistem Pertandingan

Sistem pertandingan yang terkandung dalam permainan ini akan menandingkan kumpulan digimon yang dimiliki oleh seorang pemain dengan digimon yang dimiliki oleh pemain lain. Tiap digimon mengandung atribut status yang menentukan kemampuan yang dia miliki. Status tersebut adalah sebagai berikut :

- HP : Health points, menyatakan banyaknya sisa nyawa yang dimiliki oleh digimon tersebut.
- SP : Skill points, menyatakan banyaknya sisa skill points yang dimiliki oleh digimon tersebut
- Attack : Besarnya damage yang diberikan oleh suatu digimon apabila melakukan basic attack
- Defense : Besarnya damage reduction yang didapat apabila suatu digimon diserang oleh digimon lain
- Intelligence : Besarnya damage yang diberikan apabila digimon menyerang dengan spells
- Speed : Menyatakan seberapa cepat giliran digimon tersebut akan datang
- Type : Merupakan tipe dari suatu digimon. Pengaruh stat ini akan dijelaskan di bagian penyerangan
- Attribute : Menyatakan dampak lain dari spell penyerangan yang dilakukan digimon. Dampak tersebut akan menurunkan stats musuh, stat yang diturunkan bergantung jenis atribut.
- Memory : Menyatakan cost dari tiap digimon apabila dibawa ke pertandingan

Untuk tiap pertandingannya, banyaknya maksimal digimon yang dapat ditandingkan secara bersamaan hanyalah tiga. Apabila suatu pemain membawa lebih dari 3 digimon, maka digimon yang lain akan berada akan berada dalam mode non-aktif atau dapat dibilang berada pada reserve list. Apabila suatu digimon yang sedang aktif dimainkan diserang hingga mati, terindikasi dengan health points yang habis, maka digimon tersebut dapat diganti

dengan digimon lain yang berada dalam mode non-aktif. Suatu pemain dinyatakan kalah apabila seluruh digimon yang dia miliki mati sebelum pemain lain. Sebaliknya, seorang pemain dikatakan menang apabila dia berhasil mematikan seluruh digimon lawan. Perlu dicatat bahwa kematian yang disebut tidak bersifat permanen, digimon hanya mati untuk pertandingan tersebut sehingga masih bisa digunakan untuk pertandingan kedepannya.

Sistem pertandingan ini akan dibagi menjadi 3 tahap, yakni sebagai berikut :

1) Pemilihan digimon

Pada tahap ini, pemain akan memilih digimon yang akan digunakan untuk pertandingan. Banyaknya digimon yang dapat dibawa akan bergantung pada memory size pemain dan memory cost digimon. Permasalahan ini dapat dimodelkan seperti integer knapsack problem. Pemain dapat membawa sejumlah digimon asalkan total memory cost nya tidak melebihi memory size tiap pemain. Terdapat sedikit restriksi untuk permainan digimon versi ini, bahwa banyak digimon yang dapat dibawa dalam satu pertandingan adalah maksimal 11 sehingga perlu sedikit memodifikasi permasalahan knapsack.



Gambar 3 : Cuplikan game saat pemilihan digimon

2) Pemilihan tindakan

Pada tahap ini, pemain akan memilih tindakan untuk tiap digimon yang sedang berada pada gilirannya. Beberapa alternatif tindakan yang dapat dipilih adalah :

- Attack : Melakukan basic attack kepada satu musuh sebesar status attack.
- Guard : Mengurangi damage yang diterima oleh digimon tersebut sebesar 50% apabila diserang. Opsi ini tidak akan dianalisis dalam permainan dikarenakan tidak menyebabkan pemain selangkah lebih maju kepada kemenangan, melainkan berusaha mempertahankan kehidupan digimon. Guard lebih berguna untuk meningkatkan score points di akhir pertandingan, karena lebih banyak digimon yang hidup maka lebih banyak poin yang diperoleh
- Skill : Menggunakan skill yang dimiliki oleh sebuah digimon. Jenis skill dapat berupa power attack yang memiliki dampak serupa dengan basic attack, spell yang menyerang musuh namun memberikan efek mengurangi status musuh, boost yang dapat menaikkan status digimon sendiri, heal yang dapat menambah nyawa dari digimon sendiri ataupun teman

- Escape : Kabur atau menyerah dari pertandingan. Opsi ini tidak akan digunakan karena diinginkan untuk memaksimalkan kemenangan
- Item : Menggunakan item yang tersedia. Opsi ini tidak akan dianalisis sebab tidak semua pertandingan memperbolehkan penggunaan item, sehingga algoritma greedy cukup menganalisis opsi lain
- Change : Menukar digimon yang sedang hidup dengan digimon yang ada pada reserve list. Opsi ini tidak akan dianalisis sebab hanya membuang turn saja dan tidak menuju selangkah lebih dekat kepada kemenangan



Gambar 4 : Cuplikan game saat pemilihan tindakan

Digimon dapat diklasifikasikan menjadi 3 atribut, yakni vaccine, data, dan virus. Tiap atribut ini dapat menjadi kelebihan dan kekurangan untuk atribut yang lain. Atribut vaksin mengalahkan atribut data, atribut data mengalahkan atribut virus, dan atribut virus mengalahkan atribut vaksin. Untuk digimon yang menyerang musuh dengan atribut yang dikalahkan, maka damage yang ia berikan akan dikali dua. Apabila digimon menyerang musuh dengan atribut yang mengalahkannya, maka damage yang diberikan digimon akan dikali setengah.

3) Pemilihan target

Pada tahap ini, apabila digimon memilih tindakan menyerang musuh dengan basic attack, power move, ataupun spell maka digimon wajib memilih satu musuh untuk diserang. Namun, untuk kasus digimon memilih tindakan heal ataupun boost spell maka digimon dapat memilih dirinya sendiri ataupun temannya untuk menerapkan skill tersebut.



Gambar 5 : Cuplikan game saat pemilihan target

III. PENGAPLIKASIAN ALGORITMA GREEDY

Karena sistem pertandingan terdekomposisi menjadi 3 tahap, maka kita dapat mengaplikasikan algoritma greedy masing – masing untuk tiap tahap tersebut. Dalam mengaplikasikan algoritma greedy, pertama kita harus memetakan elemen tiap tahap menjadi elemen algoritma greedy. Kemudian, akan dilakukan eksplorasi alternatif fungsi seleksi yang ada untuk tiap algoritma greedy. Tiap alternatif fungsi seleksi akan dianalisis untuk dipertimbangkan kemangkusannya dalam mengoptimalkan solusi. Pada langkah terakhir, tiap alternatif solusi lokal akan digabungkan untuk memperoleh aproksimasi dari solusi global.

1) Tahap pemilihan digimon

a) Pemetaan elemen greedy

1. Himpunan kandidat (C): Himpunan digimon yang dimiliki oleh seorang pemain
2. Himpunan solusi (S): Himpunan digimon yang merupakan kandidat solusi terpilih
3. Fungsi solusi: Fungsi yang menentukan apakah banyaknya digimon yang dibawa sudah 11, atau sudah memenuhi kapasitas memory cost.
4. Fungsi seleksi: Fungsi yang memilih digimon untuk dibawa ke dalam permainan
5. Fungsi kelayakan: Fungsi yang memeriksa apakah memory cost total lebih besar daripada memory size yang dimiliki pemain
6. Fungsi obyektif: Fungsi yang mengoptimalkan digimon yang dibawa ke pertandingan, baik dari segi kuantitas ataupun kekuatan.

b) Eksplorasi fungsi seleksi

Ditemukan tiga alternatif algoritma greedy untuk tahap ini, yakni dengan spesifikasi sebagai berikut :

• Greedy by quantity :

Alternatif greedy ini dimaksudkan untuk memilih sebanyak – banyaknya digimon hingga team cost penuh, atau hingga sudah diperoleh 11 digimon pada reserve list. Pendekatan ini dapat dilakukan dengan melakukan sorting digimon yang dimiliki secara descending, kemudian melakukan pemilihan dari indeks pertama untuk digimon yang memiliki cost lebih kecil sama dengan memory size dibagi dengan 11. Hal tersebut dilakukan agar dapat memenuhi reserve list yang akan dibawa oleh pemain.

• Greedy by profit :

Profit yang dimaksud dengan pendekatan ini adalah akumulasi total dari status digimon. Alternatif greedy ini dimaksudkan untuk memilih digimon dengan total stats yang paling tinggi. Pendekatan ini dilakukan dengan mengakumulasi total semua stat untuk tiap digimon, kemudian melakukan sorting dari total stat tersebut secara descending. Kemudian, pemain dapat melakukan pemilihan digimon dari indeks pertama

hingga penuh. Hal tersebut dilakukan agar pemain dapat memilih digimon terkuat yang ia miliki.

• Greedy by density :

Density yang dimaksud dalam pendekatan ini adalah hasil pembagian total stats dengan memory cost. Pendekatan ini dilakukan dengan mengakumulasi total semua stat untuk tiap digimon, kemudian membaginya dengan memory cost. Untuk mempermudah pencarian, lakukan sorting density secara descending. Kemudian, pemain dapat melakukan pemilihan digimon dari indeks pertama hingga penuh. Alternatif algoritma greedy ini dimaksudkan untuk memilih digimon terkuat yang ada, namun masih mempertimbangkan cost agar tetap dapat membawa sebanyak mungkin digimon kuat.

c) Analisis fungsi seleksi

Untuk pendekatan greedy by quantity, akan selalu membawa 11 digimon agar memenuhi memory cost. Namun, digimon yang dibawa cenderung merupakan digimon yang tidak terlalu kuat. Hal tersebut dapat terlihat bahwa digimon yang dibawa mayoritas memiliki stats yang lebih rendah dibandingkan stats digimon musuh. Meskipun menang secara kuantitas, digimon pemain cenderung didominasi oleh digimon musuh sehingga lebih cepat kalah.

Untuk pendekatan greedy by profit, akan diperoleh banyak digimon yang terkuat. Namun, digimon yang dibawa cenderung memiliki cost terlalu besar sehingga digimon yang dibawa relatif sedikit, sekitar tiga hingga lima dari total kapasitas 11 digimon yang dapat dibawa. Meskipun menang secara kekuatan, digimon pemain kalah pada aspek pertahanan hidup selama pertandingan karena lebih sedikit digimon yang dibawa bermakna lebih cepat menuju kekalahan apabila mati.

Untuk pendekatan greedy by density, akan diperoleh digimon yangimbang antara kekuatan dengan kuantitas yang dibawa. Rata – rata uji kasus yang diperoleh menunjukkan bahwa algoritma ini cukup efektif karena mengimbangi kedua aspek. Digimon tidak terlalu lemah sehingga didominasi lawan dan cukup kuat pada aspek pertahanan hidup. Pendekatan ini menunjukkan persentase kemenangan yang lebih besar dibandingkan kedua pendekatan lainnya.

d) Penggabungan Alternatif Solusi

Ketiga alternatif algoritma greedy yang ditawarkan sebelumnya tidak dapat karena pendekatannya yang sangat berbeda. Apabila ditimbang dari kelebihan dan kekurangan masing – masing pendekatan, maka ditemukan bahwa algoritma greedy by density yang paling menguntungkan. Dengan demikian, pseudocode-nya adalah sebagai berikut :

```
function greedyPilihDigimon(C : himpunan_kandidat) ->
himpunan_solusi
{Himpunan digimon yang merupakan kandidat solusi terpilih}
Deklarasi
X : kandidat
S : himpunan_solusi

Algoritma
S <- {}
```

```

sortByDensity(C) {Melakukan sorting himpunan tersebut
berdasarkan densitynya}
while (not SOLUSI(S) and (C != {})) do
  x <- C.get(0)
  C <- C - {X}
  If LAYAK (S union {X}) then
    S <- S union {X}
  endif
endwhile
return S

```

2) Tahap pemilihan tindakan

a) Pemetaan elemen greedy

1. Himpunan kandidat (C): Himpunan tindakan yang dapat dipilih oleh pemain
2. Himpunan solusi (S): Himpunan tindakan yang merupakan kandidat solusi terpilih
3. Fungsi solusi: Fungsi yang menentukan apabila tindakan yang dipilih sudah yang paling optimal
4. Fungsi seleksi: Fungsi yang memilih tindakan yang akan dipilih oleh pemain
5. Fungsi kelayakan: Fungsi yang menentukan apakah tindakan yang dipilih legal, seperti skill points yang cukup untuk penggunaan skill
6. Fungsi obyektif: Fungsi yang mengoptimalkan dampak dari tindakan yang dipilih oleh pemain

b) Eksplorasi fungsi seleksi

Ditemukan tiga alternatif algoritma greedy untuk tahap ini, yakni dengan spesifikasi sebagai berikut :

- Greedy by heal :

Alternatif greedy ini dimaksudkan untuk menghemat skill point sebanyak – banyaknya untuk penggunaan heal. Dengan pendekatan ini, kita hanya menggunakan skill points hanya untuk heal digimon pemain. Digimon yang dipilih dapat berupa digimon dengan nilai pengurangan health points yang paling besar. Apabila tidak perlu melakukan heal, maka digimon pemain dapat cukup melakukan basic attack untuk menyerang musuh. Hal tersebut dimaksudkan agar digimon pemain dapat tahan lama selama pertandingan.

- Greedy by spell :

Alternatif greedy ini dimaksudkan agar digimon pemain menggunakan spell secara terus menerus. Dengan pendekatan ini, kita bisa memberikan dampak pengurangan stats pada digimon lawan. Penggunaan spell dilakukan secara terus menerus hingga skill points digimon pemain habis. Apabila skill points sudah habis, kita dapat melanjutkan penyerangan dengan basic attack. Hal tersebut dilakukan agar digimon pemain lawan dapat menjadi lebih lemah.

- Greedy by boost :

Alternatif greedy ini dimaksudkan untuk menghemat skill point sebanyak – banyaknya untuk penggunaan boost. Dengan pendekatan ini, kita hanya menggunakan skill points hanya untuk boost digimon pemain. Digimon yang dipilih dapat berupa digimon dengan nilai status yang paling kecil. Untuk penyerangan, kita cukup menggunakan basic attack yang tidak menggunakan skill points. Hal tersebut dimaksudkan agar digimon pemain dapat menjadi lebih kuat.

- Greedy by power move :

Alternatif greedy ini dimaksudkan agar digimon pemain menggunakan power move secara terus menerus. Dengan pendekatan ini, kita bisa memberikan damage yang cukup besar pada digimon lawan. Penggunaan spell dilakukan secara terus menerus hingga skill points digimon pemain habis. Apabila skill points sudah habis, kita dapat melanjutkan penyerangan dengan basic attack. Hal tersebut dilakukan agar digimon pemain lawan lebih cepat mati.

c) Analisis fungsi seleksi

Untuk pendekatan greedy by heal, digimon pemain akan bertahan hidup untuk waktu yang relatif cukup lama. Namun, pada beberapa kasus ditemukan bahwa pemain musuh juga bertahan cukup lama sebab digimon pemain tidak dapat memberikan penyerangan yang cukup signifikan ke lawan. Meskipun menang pada aspek ketahanan, pendekatan ini cukup lemah pada aspek penyerangan sehingga musuh sukar mati.

Untuk pendekatan greedy by spell, digimon lawan akan mendapatkan pengurangan stats yang signifikan. Namun, pada beberapa kasus ditemukan bahwa skill points digimon pemain habis dengan cukup cepat sehingga digimon menjadi kurang relevan karena tidak bisa menggunakan skill apapun, hanya menggunakan basic attack. Selain itu, ditemukan kasus bahwa efek pengurangan status tidak dapat ditumpuk sehingga pendekatan ini kurang efektif.

Untuk pendekatan greedy by power move, digimon lawan akan mendapatkan damage yang cukup signifikan. Namun, pada beberapa kasus ditemukan bahwa skill points digimon pemain habis dengan cukup cepat sehingga digimon menjadi kurang relevan karena tidak bisa menggunakan skill apapun, hanya menggunakan basic attack. Meskipun menang pada aspek penyerangan, pendekatan ini cukup kalah pada aspek ketahanan.

Untuk pendekatan greedy by boost, digimon pemain akan mendapatkan peningkatan stats yang signifikan. Namun, pada beberapa kasus ditemukan bahwa skill points digimon pemain habis dengan cukup cepat sehingga digimon menjadi kurang relevan karena tidak bisa menggunakan skill apapun, hanya menggunakan basic attack. Selain itu, ditemukan kasus bahwa efek peningkatan status tidak dapat ditumpuk sehingga pendekatan ini kurang efektif.

d) Penggabungan Alternatif Solusi

Apabila meninjau keempat alternatif algoritma greedy sebelumnya, terlihat bahwa masing – masing pendekatan memiliki kelebihan dan kekurangan tertentu. Namun, pendekatan tersebut bisa diintegrasikan. Solusi gabungan yang

ditawarkan adalah digimon pemain mengutamakan boost apabila tidak dalam kondisi boost sebelumnya untuk meningkatkan dasar status. Kemudian, digimon menggunakan spell untuk mengurangi status musuh apabila musuh belum memiliki efek pengurangan status. Penggunaan spell juga terpengaruh oleh boost sehingga dampaknya lebih signifikan. Lalu, digimon pemain memeriksa health yang dimilikinya ataupun rekannya. Apabila ada yang dalam kondisi kritis, yakni health points di bawah 25 persen maka kita menggunakan tindakan heal untuk pemain tersebut. Apabila tidak, kita dapat menggunakan power move. Pseudocode untuk algoritma gabungannya kira kira sebagai berikut :

```

function greedyPilihTindakan(C : himpunan_kandidat,
pemain : digimon, lawan : digimon) -> himpunan_solusi
{Himpunan tindakan yang merupakan kandidat solusi terpilih}
Deklarasi
S : himpunan_solusi

Algoritma
S <- {}
if not BOOST(pemain) and MANACUKUP(pemain) then
  S <- C.BOOST
  return S
else if not DEBOOST(lawan) and MANACUKUP(pemain)
then
  S <- C.SPELL
  return S
else if HEALTH(pemain) < MAXHP(pemain)*0.25 and
MANACUKUP(pemain) then
  S <- C.HEAL
  return S
else if MANACUKUP(pemain) then
  S <- C.POWERMOVE
  return S
else
  S <- C.BASIC_ATTACK
  return S

```

3) Tahap pemilihan target

a) Pemetaan elemen greedy

1. Himpunan kandidat (C): Himpunan target yang dapat dipilih oleh pemain
2. Himpunan solusi (S): Himpunan target yang telah dipilih pemain sebagai solusi
3. Fungsi solusi: Fungsi yang menentukan target yang dipilih merupakan target paling optimal
4. Fungsi seleksi: Fungsi yang memilih target untuk dipilih oleh pemain
5. Fungsi kelayakan: Fungsi yang memeriksa apakah untuk attack, spell, dan power dipilih target musuh dan apakah heal dan boost memilih target teman.
6. Fungsi obyektif: Fungsi yang mengoptimalkan dampak pemilihan target

b) Eksplorasi fungsi seleksi

Ditemukan tiga alternatif algoritma greedy untuk tahap ini, yakni dengan spesifikasi sebagai berikut :

• Greedy by attributes :

Alternatif greedy ini dimaksudkan untuk memanfaatkan fitur atribut pada sistem pertandingan. Perhatikan bahwa dengan memilih musuh yang atributnya lebih lemah, maka damage yang diberikan dapat dikali dua. Pendekatan ini dilakukan dengan memilih target yang elemennya lebih lemah dari digimon pemain, dan apabila tidak ada cukup memilih secara random. Apabila memilih tindakan heal atau boost, maka efeknya juga dapat dikali dua. Dengan demikian, pendekatan ini dilakukan agar melipatgandakan dampak tindakan yang kita pilih.

• Greedy by stats :

Alternatif greedy ini dimaksudkan untuk dengan cepat menyingkirkan musuh dengan status yang lebih kuat. Hilangnya digimon musuh yang kuat sangat berpengaruh ke alur pertandingan karena damage yang dia berikan ke digimon pemain cukup banyak. Dengan demikian, lebih efektif apabila digimon musuh yang kuat mati sejak awal pertandingan. Apabila memilih tindakan heal atau boost, maka kita bisa melakukannya kepada karakter dengan stats/health point yang paling rendah di saat itu agar tidak terlalu lemah. Pendekatan ini dilakukan agar memperbesar kontribusi digimon pemain, dan meminimalkan kontribusi digimon musuh.

c) Analisis fungsi seleksi

Untuk pendekatan greedy by attributes, akan diperoleh dampak tindakan yang maksimal. Namun, pada beberapa kasus ditemukan bahwa terdapat kontribusi digimon pemain yang relatif rendah karena stats yang kecil. Selain itu, ditemukan juga beberapa kasus di mana digimon musuh yang terkuat sangat berpengaruh pada alur pertandingan karena tidak segera dimusnahkan. Meskipun menang secara dampak tindakan sendiri, pendekatan ini kalah pada aspek meminimalkan dampak penyerangan musuh serta menyamaratakan kontribusi tiap digimon pemain karena masih ada yang terlalu lemah.

Untuk pendekatan greedy by stats, akan diperoleh kontribusi digimon pemain yang cukup rata dan meminimalisir kontribusi dari digimon musuh. Namun, pada beberapa kasus ditemukan kekalahan dengan margin yang cukup sedikit karena sedikit lebih lambat dalam dampak penyerangan. Meskipun menang pada aspek meminimalkan dampak penyerangan musuh dan menyamaratakan kontribusi tiap digimon pemain, pendekatan ini kalah pada dampak tindakan yang dipilih.

d) Penggabungan Alternatif Solusi

Karena hanya ada dua alternatif solusi yang memiliki kelebihan dan kekurangan masing – masing, maka algoritma dapat diintegrasikan. Kedua pendekatan juga dapat dilakukan secara koheren dengan pseudocode sebagai berikut :

```

function greedyPilihTarget(C : himpunan_kandidat, pemain :
digimon) -> himpunan_solusi
{Himpunan target yang merupakan kandidat solusi terpilih}

```

Deklarasi

S : himpunan_solusi

Target :

{Perlu dicatat bahwa algoritma ini melakukan traversal untuk tiap

Algoritma

S <- {}

target <- SEARCH(S)

{Perlu dicatat bahwa algoritma di bawah ini seharusnya dilakukan bersamaan dengan fungsi search, namun ditulis terpisah untuk mempermudah pembacaan implementasi saat pencarian target yang optimal}

if PENYERANGAN() then

{Kasus mencari target yang atributnya lebih lemah}

if attribute(pemain) > attribute(target) **then**

S <- S union {target}

return S

{Kasus tidak ada atribut lawan yang lebih lemah, maka mencari digimon yang terkuat namun atributnya tidak kuat}

else if stats(target) > stats(pemain) **and** attribute(target) = attribute(pemain) **then**

S <- S union {target}

return S

{Kasus cukup mencari digimon yang atributnya sama}

else if attribute(target) = attribute(pemain) **then**

S <- S union {target}

return S

else {kasus digimon pasti lebih lemah, mau gamau pilih}

S <- S union {target}

return S

else {kasus boost atau heal}

{Kasus mencari target yang atributnya lebih lemah}

if attribute(pemain) > attribute(target) **then**

S <- S union {target}

return S

{Kasus tidak ada atribut lawan yang lebih lemah, maka mencari digimon yang terkuat namun atributnya tidak kuat}

else if stats(target) < stats(pemain) **or** HEALTH(target) <= 0.25*MAX_HP(target) **then**

S <- S union {target}

return S

{Kasus lain, cukup pilih random karena semuanya berdampak sama}

else

S <- S union {target}

return S

4) Uji Kasus algoritma gabungan

Agar hasil uji kasus memberikan hasil yang cukup adil, percobaan pertandingan dilakukan dalam mode arena di mana pemain bisa bertemu dengan handicap yang rata dan status digimon yang serupa. Dalam mode ini, dilakukan sepuluh kali percobaan pertandingan. Hasil yang diperoleh adalah 8 kali kemenangan dari 10 pertandingan. Hasil tersebut membuktikan bahwa solusi yang dibangun dari algoritma

greedy bukanlah solusi yang optimum global, melainkan hanyalah solusi optimum lokal yang secara aproksimasi hasilnya mendekati yang terbaik. Algoritma ini relatif sederhana untuk digunakan, sehingga hasil yang diperoleh cukup memuaskan untuk batas waktu dan kondisi yang wajar.

IV. KESIMPULAN DAN SARAN

Algoritma greedy dapat diaplikasikan ke banyak permasalahan di dunia nyata, salah satunya pada sistem pertandingan dalam permainan Digimon Cyber Sleuth. Dalam makalah ini, diperoleh kelebihan dan kekurangan dari tiap alternatif algoritma greedy yang sudah ditawarkan. Makalah ini juga sudah berada pada tahap uji coba untuk hasil solusi gabungan yang sudah dibangun.

Data menunjukkan bahwa algoritma greedy yang dibentuk bukanlah merupakan solusi optimum global karena masih terdapat kasus yang berujung ke kekalahan pemain. Algoritma greedy ini digunakan karena memprioritaskan kesederhanaan dan kecepatan dibandingkan dengan ketepatan sehingga solusi yang dibangun merupakan pendekatan saja. Apabila ingin menggunakan solusi yang memastikan kemenangan pemain, lebih baik menggunakan algoritma lain yang lebih tepat dan terjamin. Dengan demikian, dapat disimpulkan bahwa apabila mengutamakan kompleksitas algoritma yang relatif sederhana dan batas waktu yang wajar, maka algoritma greedy menjadi alternatif solusi yang baik.

Perlu dicatat bahwa solusi alternatif greedy yang dibangun dalam makalah ini masih belum merupakan solusi greedy yang terbaik. Algoritma greedy yang ditawarkan masih dapat dikembangkan lagi dengan beberapa saran, misalnya untuk jenis skill move yang dapat memilih target lebih dari satu. Selain itu, tindakan change perlu dianalisis sebab terdapat kasus pada search pemilihan target memberikan dampak yang signifikan meskipun membuang – buang turn. Algoritma greedy yang ditawarkan juga dapat mencakup mode pertandingan di mana penggunaan item diperbolehkan. Selain itu, analisis stats yang digunakan masih pada tahap akumulasi total. Saran untuk penggunaan stats kedepannya dapat mempertimbangkan untuk tiap masing – masing stats, tidak hanya akumulasi total saja.

V. LINK MENUJU VIDEO

Penulis melampirkan video penjelasan verbal terkait penjelasan permainan dan algoritma greedy yang telah diaplikasikan dengan tautan <https://youtu.be/ihufxEfSFqk>

VI. UCAPAN TERIMA KASIH

Puji syukur dan terimakasih terhadap Tuhan yang Maha Esa telah penulis ucapkan karena berkat rahmat-Nya, penulis mampu menyelesaikan tugas yang berupa karya ilmiah ini dengan baik dan tepat waktu. Penulis juga mengucapkan terima kasihnya kepada Bu Masayu Leylia Khodra, S.T., M.T serta Bu Dr. Nur Ulfa Maulidevi, S.T., M.Sc. selaku dosen mata pelajaran Strategi Algoritma yang telah mengajar saya. Tidak lupa juga, penulis ingin mengucapkan terimakasih yang sebesar – besarnya kepada semua pihak yang telah membantu penulis dalam pembuatan karya ilmiah “Aplikasi Algoritma

Greedy pada Sistem Pertandingan dalam Permainan Digimon Cyber Sleuth” ini atas segala dukungan dan bantuan yang telah diberikan selama proses pembuatan tugas.

REFERENSI

- [1] Munir, Rinaldi. Algoritma Greedy (Bagian 1). Diakses melalui [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) pada 17 Mei 2022 pukul 20.00 WIB
- [2] Kusumaningsih, Ari. Optimising the tower-defense games with advanced local cultural content and a greedy algorithm. <https://iopscience.iop.org/article/10.1088/1742-6596/1569/2/022067/meta> pada 17 Mei 2022 pukul 20.30 WIB
- [3] Grindosaur. Digimon Cyber Sleuth: Digimon Field. Diakses melalui <https://www.grindosaur.com/en/games/digimon/digimon-story-cyber-sleuth/digimon> pada 19 Mei 2022 pukul 18.00
- [4] Toynanaka. Digimon Cyber Sleuth: Support Skill Guide. Diakses melalui <https://digimoncybersleuth.wordpress.com/support-skill-guide/> pada 19 Mei 2022 pukul 19.00

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, 20 Mei 2022



Saul Sayers 13520094