

Aplikasi Algoritma Greedy pada Pertarungan di Persona 4 Golden

Bariza Haqi -13520018

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13520018@std.stei.itb.ac.id

Abstract—Persona 4 Golden adalah *game Turn-based RPG* yang populer bagi kalangan gamer. Pada permainan ini pemain akan melakukan pertarungan dengan musuh untuk menyelesaikan cerita permainan. Pada pertarungan tersebut, pemain dan musuh akan diberi kesempatan giliran satu sama lain untuk menyerang atau mengeluarkan kemampuan. Oleh karena itu, dibutuhkan sebuah strategi untuk mengalahkan musuh dan karakter yang dimainkan pemain tidak mati. Salah satu cara yang dapat digunakan yaitu menggunakan algoritma greedy untuk mendapatkan solusi optimal. Algoritma ini akan memilih pilihan yang terbaik saat melawan musuh.

Kata Kunci—Persona-4, algoritma-greedy, RPG, turn-based, pertarungan, game

I. PENDAHULUAN

Bermain game sudah menjadi hal yang biasa dilakukan oleh kalangan masyarakat diberbagai umur baik sebagai hobi atau untuk *refreshing*. Banyak sekali genre *game* yang populer pada zaman ini. Salah satunya adalah *turn-based RPG*. Pada genre ini pemain akan bermain bergiliran, tidak secara simultan saat bertarung melawan musuh. Pemain hanya dapat mengatur strategi saat berada digilirannya sendiri. Salah satu game turn-based RPG yang cukup terkenal adalah Persona 4 Golden.

Tujuan dari *game* Persona 4 Golden adalah memecahkan suatu misteri di sebuah kota. Pada perjalanannya, karakter utama beserta tiga temannya akan bertarung melawan monster.

Untuk mengalahkan monster, pemain dapat menentukan strategi penyerangan saat mendapati gilirannya. Dibutuhkan strategi serangan yang apik sehingga membuat musuh tidak bisa menyerang. Pada makalah ini akan dijelaskan pemanfaatan algoritma *greedy* dalam pemilihan serangan terbaik agar pemain dapat menyerang terus tanpa membuat musuh bisa menyerang atau serangan yang didapati dari musuh kecil.



Gambar 1.1 Cover Game Persona 4 Golden

Sumber :

https://www.pcgamingwiki.com/wiki/Persona_4_Golden

II. LANDASAN TEORI

A. Algoritma Greedy

Algoritma *greedy* merupakan metode yang paling populer dan sederhana untuk memecahkan persoalan optimasi. Terdapat dua macam persoalan optimasi yaitu maksimasi (*maximization*) dan minimasi (*minimization*).

Algoritma *greedy* menggunakan pendekatan masalah dengan mencari nilai terbaik atau optimal dari setiap langkah (*step by step*) tanpa memperhatikan konsekuensi ke depannya dan berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global. Oleh karena itu, algoritma *greedy* tidak selalu mendapati solusi yang paling optimal. Algoritma *greedy* tidak melakukan pencarian penuh sehingga kompleksitas waktunya cukup rendah. Algoritma ini biasa digunakan ketika kita tidak membutuhkan solusi yang dekat dengan solusi sempurna. Prinsip algoritma *greedy* adalah “take what you can get now!”.

Berikut merupakan elemen-elemen algoritma *greedy*.

1. Himpunan kandidat yaitu himpunan yang berisi kandidat yang akan dipilih pada setiap langkah untuk menjadi bagian dari solusi yang dipilih.
2. Himpunan solusi yaitu himpunan yang berisi kandidat yang sudah dipilih.
3. Fungsi solusi yaitu fungsi yang menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi.
4. Fungsi seleksi (selection function) yaitu fungsi yang digunakan untuk memilih kandidat berdasarkan strategi *greedy* tertentu. Strategi *greedy* ini bersifat heuristik.
5. Fungsi kelayakan (feasible) yaitu fungsi untuk memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi (layak atau tidak)
6. Fungsi obyektif yaitu fungsi untuk menyatakan tujuan dari persoalan yang ingin diselesaikan

Berikut adalah contoh-contoh persoalan yang dapat diselesaikan dengan algoritma *greedy*.

1. Persoalan penukaran uang (*coin exchange problem*)
2. Persoalan memilih aktivitas (*activity selection problem*)
3. Minimisasi waktu di dalam sistem
4. Persoalan knapsack (*knapsack problem*)
5. Penjadwalan Job dengan tenggat waktu (*job scheduling with deadlines*)
6. Pohon merentang minimum (*minimum spanning tree*)
7. Lintasan terpendek (*shortest path*)
8. Kode Huffman (*Huffman code*)
9. Pecahan Mesir (*Egyptian fraction*)

B. Persona 4 Golden

Persona 4 adalah *game* yang dikembangkan oleh Atlus dan dirilis pada tanggal 10 Juli 2008. Setelah itu rilis Persona 4 Golden yang merupakan Persona 4 dengan beberapa tambahan fitur pada tanggal 14 Juni 2012. *Game* ini dapat dimainkan pada platform PC dan Playstation.

Pada permainan ini, pemain diharuskan untuk memecahkan misteri di sebuah kota. Pemain bebas berkeliling dan berinteraksi saat berada dalam permainan. Pada waktu tertentu, pemain diharuskan pergi ke sebuah *dungeon* atau ruangan berisi tantangan yang harus dilalui pemain. Pada ruangan ini pemain harus membawa satu karakter utama dan boleh mengajak temannya bertarung maksimal 3 orang. Pada ruangan ini banyak monster yang harus dilawan pemain. Ketika karakter utama berinteraksi dengan monster maka pertarungan dimulai secara otomatis.



Gambar 2.1 Tampilan saat dalam pertarungan Persona 4 Golden

Sumber: Arsip penulis

Terdapat beberapa istilah yang digunakan dalam Persona 4 yaitu

1. Elemen

Terdapat 7 elemen utama pada permainan Persona 4 yaitu *fire, ice, elec, wind, light, darkness, phys* dan satu elemen khusus yaitu *almighty*. Setiap karakter dan monster dapat memiliki kekebalan dan kelemahan terhadap serangan dari elemen-elemen tersebut kecuali *almighty*. *Almighty* merupakan elemen tambahan yang unik karena hanya berupa *skill* dan tidak bisa memanfaatkan kelemahan elemen sehingga kurang efektif dalam beberapa pertarungan dan biasa digunakan saat monster kebal terhadap segala elemen.

2. Karakter

Terdapat beberapa karakter yang dapat dimainkan saat pertarungan di persona 4 yaitu satu karakter utama dan 3 karakter yang dapat dibawa ke *dungeon* dari 5 karakter yang ada. Setiap karakter memiliki *hp (health point)* dan *sp (skill points)* tertentu. Ketika *hp* karakter habis, maka karakter tersebut akan mati. Ketika *sp* karakter habis maka karakter tidak dapat mengeluarkan *skill*. Setiap karakter memiliki kekuatan yang didapat dari makhluk yang disebut *persona*. Setiap karakter memiliki tepat 1 *persona* kecuali karakter utama yang memiliki lebih dari 1 *persona* sehingga karakter utama memiliki lebih banyak *skill* dan mampu melawan musuh dengan kekebalan elemen apapun.

3. Skill

Skill merupakan kemampuan yang dimiliki setiap karakter. *Skill* digunakan untuk menyerang musuh atau melakukan pertahanan. Ketika karakter menggunakan *skill* maka *sp* karakter akan berkurang terkecuali *skill phys* maka *hp* karakter yang akan berkurang. Karakter dapat ditambah *hp*-nya menggunakan *sp* ketika *hp*-nya sedikit namun untuk *sp* hanya bisa ditambah dari *item*. *Skill* karakter terdiri dari *magic skill* yaitu *skill* untuk menyerang dengan 8 elemen yang ada, *ailment skill* untuk menyebabkan

status ailment terhadap musuh, *recovery skill* untuk menyembuhkan karakter dan *support skill* untuk memperkuat karakter atau melemahkan musuh. *Magic skill darkness* dan *light* merupakan *skill* yang unik karena dapat mengalahkan monster dalam satu kali serang namun memiliki kemungkinan kecil untuk terkena serangan tersebut dan banyak monster yang tahan terhadap kedua elemen tersebut. *Skill* ini biasanya dipakai saat diawal permainan untuk menghabisi musuh yang lemah terhadap elemen tersebut karena kemungkinan terkena *skill* menjadi besar untuk musuh yang lemah terhadap elemen tersebut. *Skill* juga terbagi menjadi *skill to all* yaitu *skill* yang berefek ke seluruh karakter atau seluruh monster dan *skill to one* yaitu *skill* yang berefek ke satu karakter atau satu monster

4. Item

Item merupakan barang yang didapat dari *dungeon* atau barang yang dibeli dari toko. *Item* terdiri dari barang-barang yang sangat berguna saat pertarungan dan efek item sama dengan menggunakan *skill* seperti menyembuhkan karakter pemain dari *status ailment*, kabur dari pertempuran, menghidupkan teman karakter utama yang mati, memperkuat karakter dan melemahkan musuh. Kelebihan dari *item* adalah terdapat *item* untuk mengisi *sp* yang digunakan untuk memakai *skill*. Penambahan *sp* hanya bisa untuk 1 karakter.

5. Dungeon

Dungeon adalah ruangan berisi tantangan yang harus dilalui pemain. *Dungeon* ini berisi monster yang harus dilewati atau dilawan pemain saat berada dalam permainan. Terdapat *dungeon* yang mudah dan sulit tergantung kemajuan cerita dari *game*-nya. Semakin jauh cerita *game*-nya semakin sulit *dungeon* yang harus dilewati. Level karakter pemain juga sangat berpengaruh. Semakin besar level karakter pemain, semakin mudah untuk mengalahkan monster di *dungeon*. Ketika pemain melawan monster dan dalam 1-2 kali giliran di pertarungan karakter langsung mati maka pemain harus menaikkan level karakter lebih dahulu dengan melawan monster yang lebih mudah saat dilawan.

6. Monster

Monster adalah makhluk yang bermunculan di *dungeon*. Setiap monster dapat menyerang, menggunakan *skill* dan kabur dari pertarungan. Setelah mengalahkan monster, maka karakter pemain akan mendapatkan *exp* (*experience point*) untuk menaikkan level. Bos monster merupakan monster unik yang sangat kuat dengan serangan yang besar dan tahan terhadap *skill ailment*. Pemain juga tidak bisa kabur dari pertarungan saat melawan bos monster sehingga pemain biasanya akan melawan bos monster ketika level karakter sudah cukup dan biasanya diberi peringatan terlebih dahulu di *dungeon*nya saat akan bertarung melawan bos monster.

7. Critical

Serangan *phys* memiliki kemungkinan menyebabkan *critical*. Ketika musuh atau karakter terkena *critical* maka serangan yang diterima menjadi dua kali lipat dan menyebabkan *status ailment down*.

8. Status Ailment

Status Ailment adalah keadaan suatu karakter atau monster ketika terkena *skill ailment* kecuali *down* dan *dizzy*. Efek *status ailment* dapat menghilang dengan sendirinya setelah beberapa kali giliran. Berikut adalah beberapa kondisi dari *status ailment*.

a. Poison

Karakter akan berkurang darahnya 10% setiap gilirannya selesai.

b. Silence

Karakter tidak dapat menggunakan *skill* saat berada dalam pertarungan

c. Panic

Karakter tidak dapat menyerang atau malah menyerang temannya sendiri.

d. Rage

Karakter akan melakukan *attack* secara otomatis dengan serangan 2 kali lipat namun pertahanan dan akurasinya menjadi setengahnya.

e. Fear

Karakter memiliki kemungkinan kabur dari permainan.

f. Exhaustion

sp karakter akan berkurang setiap gilirannya dan menerima karakter serangan 1.5 kali lipat dari awal

g. Enervation

Status atau kekuatan karakter berkurang sebesar 50%

h. Down dan Dizzy

Down dapat terjadi ketika karakter terkena serangan oleh kelemahannya atau *critical* saat tidak dalam kondisi *dizzy* atau *down* sehingga menyebabkan serangan 2 kali lipat. Ketika monster terkena *down* oleh karakter maka karakter yang menyerang akan mendapatkan tambahan gilirannya lagi sebanyak 1 kali. Karakter yang terkena *down* ketika diserang oleh kelemahannya atau *critical* lagi menyebabkan efek *Dizzy* yaitu kehilangan gilirannya sehingga tidak bisa menyerang atau menggunakan *skill*. Ketika semua monster yang ada dalam pertarungan dalam kondisi *down*, pemain dapat melakukan serangan ke semua monster dengan elemen *almighty-phys* yang disebut *All-Out*

Attack. Pemain bebas untuk tidak menggunakan *All-Out Attack* dan menggunakan tambahan giliran 1 kali untuk strategi lain.

Terdapat beberapa pilihan saat dalam berada pertarungan yaitu

1. *Analysis*

Melakukan analisis terhadap musuh yaitu melihat kelemahan dan kekebalan musuh dari elemen. Melakukan analisis tidak membuat giliran karakter berakhir



Gambar 2.2 Tampilan saat melakukan *analysis*

Sumber: Arsip penulis

Terdapat beberapa keterangan dalam *analysis* yaitu:

a. Normal

Ditandai dengan “-” yang berarti musuh tidak memiliki kekebalan dan kelemahan terhadap elemen tersebut

b. *Resist*

Ditandai dengan “Str” yang berarti serangan yang diterima musuh akan berkurang ketika diserang dengan elemen tersebut.

c. *Block*

Ditandai dengan “Nul” yang berarti serangan yang diterima musuh dengan elemen tersebut tidak akan berefek atau musuh tidak akan terkena *damage*.

d. *Drain*

Ditandai dengan “Dr” yang berarti serangan yang diterima musuh dengan elemen tersebut akan diubah menjadi penambahan darah terhadap musuh yang terkena.

e. *Repel*

Ditandai dengan “Rpl” yang berarti serangan yang diterima musuh dengan elemen tersebut tidak akan terkena *damage* dan karakter yang menyerang akan terkena serangannya sendiri.

f. Weak

Ditandai dengan “Wk” yang berarti serangan yang diterima musuh dengan elemen tersebut menjadi 2 kali lipat dan menyebabkan musuh terkena efek *status ailment down*.

g. Unknown

Ditandai dengan “?” yang berarti keterangan musuh terhadap elemen tersebut belum diketahui. Saat melakukan pertarungan dengan musuh baru, semua keterangan elemen berupa unknown. Keterangan elemen musuh akan diketahui dan diperbarui setelah pemain mencoba menyerang monster dengan elemen yang dipilih.

2. *Tactics*

Mengubah taktik serangan seperti karakter menyerang dengan sendiri atau diatur oleh pemain

3. Guard

Melakukan pertahanan sehingga kelemahan karakter terhadap suatu elemen tidak dapat diserang dan *damage* yang diterima menjadi setengahnya.

4. *Attack*

Melakukan serangan biasa secara *phys* tanpa mengurangi *sp* dan *hp* pada satu musuh. *damage* yang didapat musuh tidak semaksimal menggunakan *skill*.

5. *Skill*

Melakukan *skill* dari *skill* yang dimiliki karakter

6. *Persona*

melakukan *analysis* terhadap *persona* karakter atau mengganti *persona* (untuk karakter utama)

7. *Item*

Menggunakan item terhadap karakter, teman karakter atau musuh.

8. *Escape*

Kabur dari pertarungan jika kondisi sudah tidak memungkinkan seperti monster sangat kuat dan tidak bisa dilawan atau banyak teman karakter utama yang mati. Agar lebih pasti gunakan *skill escape* atau *item escape*.

III. APLIKASI ALGORITMA GREEDY PADA PERTARUNGAN DI PERSONA 4 GOLDEN

Pada implementasinya, akan ditentukan asumsi dan batasan terlebih dahulu. Tujuan *greedy*-nya adalah memaksimalkan kemenangan pertarungan yaitu dengan memaksimalkan serangan terhadap musuh dan meminimalkan serangan yang diterima karakter.

A. Asumsi dan Batasan

Pada permainan *Persona 4* nya sendiri dibutuhkan strategi yang rumit dan banyak sekali *skill* yang dapat dipilih saat dalam pertarungan. Saat karakter menyerang menggunakan

skill elemen *wind*, bisa saja karakter memiliki beberapa *skill* elemen *wind* dengan *damage* yang berbeda dan pengurangan *sp* yang berbeda. Karena menggunakan strategi *greedy* memaksimalkan serangan terhadap musuh maka *skill* elemen yang dipilih otomatis adalah yang memiliki *damage* terbesar. *Skill* yang unik dan terlalu kuat atau musuh dengan kekuatan khusus akan diabaikan seperti *Ghastly Wail* dengan elemen *darkness* yang dapat membunuh semua musuh 100% dalam satu serangan ketika musuh dalam keadaan *Fear* karena *skill* tersebut hanya ada di akhir *game*.

Pada saat menggunakan *magic skill*, *skill to all* menjadi pilihan utama karena dapat menyerang ke semua musuh sehingga serangan menjadi maksimal. Jika hanya ada satu musuh saja atau terdapat musuh dengan status elemen *dr* atau *rpl* maka *skill to one* otomatis dipilih karena dapat menguntungkan musuh atau merugikan karakter.

Karena item dan *skill* memiliki efek yang sama maka akan disatukan menjadi *action* dan penambahan *sp* dimasukkan ke *recovery*. Untuk *ailment skill* akan otomatis memakai *skill to one* jika ada satu musuh dan *skill to all* untuk musuh lebih dari satu. Jika karakter tidak memiliki *skill to all* saat terdapat banyak musuh maka *ailment skill* tidak digunakan. Untuk *recovery* akan terbagi menjadi *heal* untuk menambah darah karakter termasuk menghidupkan karakter, *recover sp* untuk memulihkan *sp* karakter dan *cures ailment* untuk menyembuhkan status *ailment*. Karena *recovery* bisa berupa *item* maka *recovery* dapat digunakan oleh karakter apapun.

Untuk *heal* dan *cures ailment* akan otomatis memakai *skill to one* jika hanya satu karakter membutuhkan efeknya dan *skill to all* jika lebih dari satu karakter yang membutuhkan efeknya. *Skill to one* terpaksa dipakai jika karakter tidak memiliki *skill to all* pada *heal* dan *cures ailment*. Untuk *support skill* hanya akan menjadi *support skill* karena banyaknya *skill* yang bermacam-macam dan semuanya hanya bertujuan untuk melemahkan musuh atau memperkuat karakter. Untuk *command* atau perintah saat pertarungan hanya terdiri dari *attack*, *guard*, *action* dan *escape*.

B. Implementasi

Agar serangan yang dilakukan terhadap musuh optimal tentu kita harus mengatur pilihan saat berada dalam pertarungan. Tentu saja harus mengambil pilihan terbaik disaat itu juga tanpa memperhatikan konsekuensi ke depannya. Oleh karena itu, harus dibuat sebuah susunan prioritas pilihan agar memaksimalkan serangan terhadap musuh.

Untuk penyusunan datanya sendiri adalah :

1. *Action* terdiri dari *magic*, *ailment*, *recovery* dan *support*.
2. *Magic* terdiri dari *fire*, *ice*, *elec*, *wind*, *light*, *darkness*, *phys* dan *almighty*.
3. *Ailment* terdiri dari *poison*, *fear*, *silence*, *rage*, *panic*, *exhaustion* dan *enervation*.
4. Status elemen terdiri dari “-” (*normal*), “str” (*resist*), “nul” (*block*), “dr” (*drain*), “rpl” (*repel*), “wk” (*weak*) dan “?” (*unknown*).

5. Status *ailment* terdiri dari *poison*, *fear*, *silence*, *rage*, *panic*, *exhaustion*, *enervation*, *dizzy* dan *down*.
6. Musuh kebal terhadap elemen jika status elemennya “str”, “nul”, “dr” atau “rpl”.

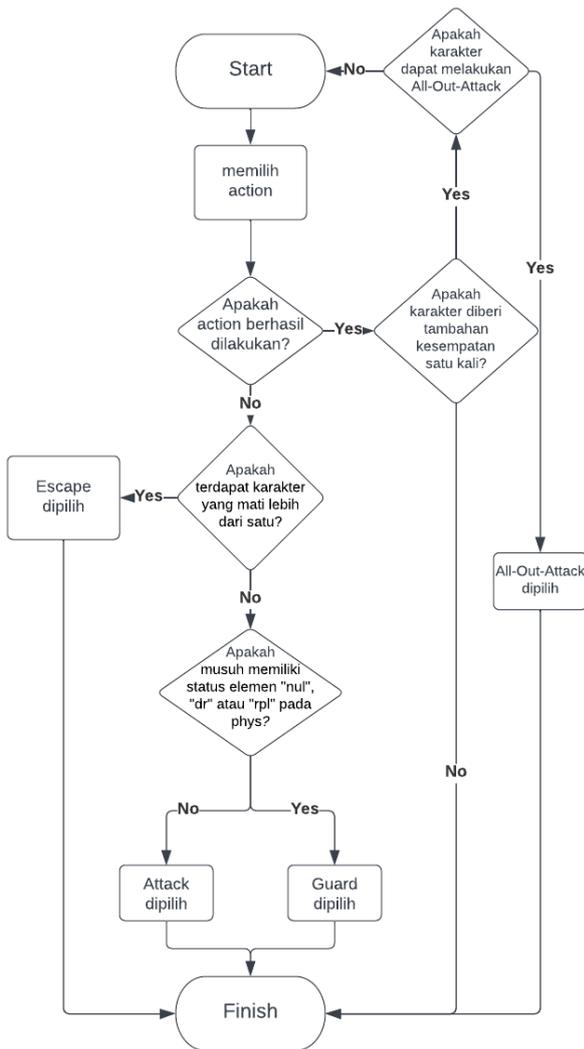
Himpunan data yang didapat adalah:

- Elemen = {fire, ice, elec, wind, light, darkness, phys}
- Command = {attack, guard, action, escape}
- Action = {magic, ailment, recovery, support}
- Magic = {fire, ice, elec, wind, light, darkness, phys, almighty}
- Ailment = {poison, fear, silence, rage, panic, exhaustion, enervation}
- Recovery = {heal, recover sp, cures ailment}
- StatusElemen = {-, str, nul, dr, rpl, wk, ?}
- StatusAilment = {poison, fear, silence, rage, panic, exhaustion, enervation, dizzy, down}

Dari data yang didapat kita buat prioritas pilihan terbaik saat dalam pertarungan untuk memaksimalkan kemenangan pertarungan. Pada *command*, untuk memaksimalkan serangan dan pertahanan tentu *action* menjadi paling prioritas dari yang lain. Jika pada bagian *action* tidak dilakukan karena tidak memungkinkan maka karakter akan melakukan pilihan lain. Jika terdapat karakter yang mati lebih dari satu maka akan melakukan *escape*. Jika musuh memiliki status elemen “nul”, “dr” atau “rpl” pada elemen *phys* maka karakter akan melakukan *guard*. Jika tidak, maka karakter akan melakukan *attack*.

Untuk bagian *action*, Jika *sp* karakter habis atau tidak cukup untuk melakukan *skill* maka karakter akan melakukan *recovery sp* karena tanpa *sp*, karakter tidak bisa menggunakan *action* lain. Jika terdapat musuh dengan status elemen “wk” dan tidak sedang dalam keadaan *down* dan karakter dapat melakukan *magic* elemen tersebut (memiliki skillnya dan hp mencukupi untuk serangan *phys*) maka *magic* dipilih sesuai elemen tersebut. Ketika musuh dalam keadaan tersebut diserang menggunakan elemen yang *weak* maka musuh akan *down* dan karakter akan mendapatkan tambahan giliran sebanyak satu kali sehingga menjadi pilihan prioritas utama jika didapati kesempatan tersebut. Jika semua musuh *down* dan memiliki kesempatan untuk melakukan serangan kepada semua musuh atau disebut dengan *All-Out Attack*, maka pilihan tersebut langsung dipilih dalam kondisi apapun karena serangan tersebut tidak bisa ditahan dan dihindari walaupun status elemen musuh “nul”, “dr” atau “rpl” dan serangan yang dihasilkan lumayan besar.

Jika keadaan sebelumnya tidak sesuai maka karakter akan melakukan *heal* jika terdapat karakter lain yang *hp*-nya kurang dari 50% atau terdapat karakter yang mati. *Cures ailment* akan menjadi pilihan selanjutnya jika tidak ada karakter dengan *hp* dibawah 50% dan semua karakter hidup dan terdapat karakter yang terkena status *ailment*. Jika musuh tidak dalam durasi atau tidak kebal terhadap *ailment* dan karakter memiliki



Gambar 3.2 Flowchart Greedy pada pemilihan command

Sumber: Arsip penulis

C. Eksperimen

Pada bagian ini akan dilakukan eksperimen dengan memainkan pertarungan pada Persona 4 Golden. Urutan pilihan akan disesuaikan dengan hasil dari Flowchart Greedy.

1) Elemen *weak* diketahui

Terdapat 2 musuh pada pertarungan dengan dimulai dari karakter utama yang akan membuat pilihan. *Bar* berwarna kuning menandakan *hp* dan *bar* berwarna hijau menandakan *sp*.



Gambar 3.3 Tampilan pertarungan

Sumber: Arsip penulis

Status musuh memiliki kelemahan terhadap elemen *phys* dan kebal terhadap semua elemen selain *phys*.



Gambar 3.4 Status musuh Blue Sigil

Sumber: Arsip penulis

pilihan karakter pertama:

Mulai → pilih action → sp cukup untuk melakukan skill? (yes) → apakah terdapat musuh dengan status elemen “wk” dan tidak sedang dalam keadaan down? (yes) → Apakah karakter dapat melakukan magic elemen tersebut? (yes) → magic elemen *phys* dipilih (karena musuh memiliki status elemen *wk* pada elemen *phys*) → (Kedua musuh down karena serangan ke kedua musuh dengan elemen *phys*) → Apakah action berhasil dilakukan? (yes) → Apakah karakter diberi tambahan kesempatan satu kali? (yes) → Apakah karakter dapat melakukan All-Out-Attack? (Yes) → All-Out-Attack dipilih → Selesai

Kedua musuh masih hidup dan giliran akan dilanjutkan oleh karakter kedua. Pilihan karakter kedua akan sama dengan karakter pertama dan kedua musuh mati.



Gambar 3.5 Tampilan saat musuh mati

Sumber: Arsip penulis

Hasil akhir menang dan musuh tidak diberi kesempatan untuk melawan.

2) Elemen Weak tidak diketahui

Terdapat 1 musuh pada pertarungan dengan semua status tidak diketahui.



Gambar 3.5 Status musuh Scarlet Turret

Sumber: Arsip penulis

pilihan karakter pertama:

Mulai → pilih action → sp cukup untuk melakukan skill? (yes) → apakah terdapat musuh dengan status elemen “wk” dan tidak sedang dalam keadaan down? (no) → Apakah terdapat karakter yang hp-nya kurang dari 50% atau mati? (No) → Apakah terdapat karakter yang terkena status ailment? (No) → Apakah musuh tidak dalam durasi atau tidak kebal terhadap ailment dan karakter memiliki ailment? (Yes) → Ailment panic dipilih karena prioritas utama → Apakah action berhasil dilakukan? (yes) → Apakah karakter diberi tambahan kesempatan satu kali? (No) → Selesai

Musuh terkena *status ailment panic* yang menyebabkan musuh tidak dapat menyerang.



Gambar 3.5 Tampilan musuh setelah terkena panic

Sumber: Arsip penulis

pilihan karakter kedua:

Mulai → pilih action → sp cukup untuk melakukan skill? (yes) → apakah terdapat musuh dengan status elemen “wk” dan tidak sedang dalam keadaan down? (no) → Apakah terdapat karakter yang hp-nya kurang dari 50% atau mati? (No) → Apakah terdapat karakter yang terkena status ailment? (No) → Apakah musuh tidak dalam durasi atau tidak kebal terhadap ailment dan karakter memiliki ailment? (No) → Apakah terdapat musuh dengan status

elemen “wk” dan karakter memiliki elemen tersebut? (No) → Apakah terdapat musuh dengan status elemen “?” dan karakter memiliki elemen tersebut? (yes) → magic elemen phys dipilih (karena pada prioritas, karakter tidak memiliki skill light dan darkness) → Apakah action berhasil dilakukan? (yes) → Apakah karakter diberi tambahan kesempatan satu kali? (No) → Selesai

Musuh terkena serangan dan status elemen *phys* musuh adalah “str”.

Pemilihan karakter ketiga sama dengan karakter kedua dengan elemen *ice* dan status elemen *ice* musuh adalah “_”.

Pemilihan karakter keempat sama dengan karakter kedua dengan elemen *darkness*. Serangan elemen *darkness* tidak terkena yang berarti musuh tidak memiliki kelemahan pada elemen *darkness*.

Pada giliran musuh karena musuh terkena *status ailment panic* maka musuh tidak dapat menyerang dan karakter mendapat gilirannya lagi pada putaran kedua.

Pemilihan karakter pertama sama dengan karakter kedua pada putaran pertama dengan elemen *light*. Serangan elemen *light* tidak terkena yang berarti musuh tidak memiliki kelemahan pada elemen *light*.

pemilihan karakter kedua sama dengan karakter kedua pada putaran pertama dengan elemen *wind* dan status elemen *wind* musuh adalah “_”.

pilihan karakter ketiga:

Mulai → pilih action → sp cukup untuk melakukan skill? (yes) → apakah terdapat musuh dengan status elemen “wk” dan tidak sedang dalam keadaan down? (no) → Apakah terdapat karakter yang hp-nya kurang dari 50% atau mati? (No) → Apakah terdapat karakter yang terkena status ailment? (No) → Apakah musuh tidak dalam durasi atau tidak kebal terhadap ailment dan karakter memiliki ailment? (No) → Apakah terdapat musuh dengan status elemen “wk” dan karakter memiliki elemen tersebut? (No) → Apakah terdapat musuh dengan status elemen unknown dan karakter memiliki elemen tersebut? (No) → Apakah terdapat musuh dengan status elemen “_” dan karakter memiliki elemen tersebut? (Yes) → magic elemen ice dipilih (karakter hanya memiliki skill phys dan ice, musuh kebal terhadap phys maka ice dipilih) → Apakah action berhasil dilakukan? (yes) → Apakah karakter diberi tambahan kesempatan satu kali? (No) → Selesai

Pada serangan ini musuh mati dan hasil akhir menang dengan musuh tidak diberi kesempatan untuk melawan.



Gambar 3.5 Tampilan saat musuh mati

Sumber: Arsip penulis

IV. KESIMPULAN

Banyak sekali aplikasi yang dapat diterapkan dari Strategi Algoritma. Salah satunya adalah penerapan algoritma *greedy* dari materi algoritma *greedy*. Algoritma *greedy* dapat dipakai untuk mendapatkan solusi optimal pada suatu permasalahan. Hasil yang optimal dapat diperoleh walaupun terkadang strategi *greedy* tidak selalu mendapat hasil yang paling optimal. Namun karena kompleksitas waktunya yang sangat cepat sehingga menjadi kelebihan dari algoritma *greedy* dibanding dengan algoritma yang lain. Oleh karena itu, algoritma *greedy* sangat bermanfaat dan dapat diaplikasikan dalam berbagai permasalahan.

TAUTAN VIDEO YOUTUBE

<https://youtu.be/x7TKyWSyT3o>

UCAPAN TERIMA KASIH

Pertama-tama, penulis mengucapkan puji syukur ke hadirat Allah SWT karena rahmat dan hidayahnya penulis dapat menyelesaikan tugas makalah ini. Penulis juga mengucapkan terima kasih kepada Bapak Rinaldi Munir selaku dosen mata kuliah Strategi Algoritma, yang selama ini membimbing pembelajaran Strategi Algoritma sehingga membantu pengerjaan makalah ini

REFERENSI

- [1] [https://megamitensei.fandom.com/wiki/Affinities_\(mechanic\)](https://megamitensei.fandom.com/wiki/Affinities_(mechanic)) Diakses pada 7 Mei 2022
- [2] https://megamitensei.fandom.com/wiki/Status_Changes_in_Persona_4 Diakses pada 7 Mei 2022
- [3] https://megamitensei.fandom.com/wiki/List_of_Persona_4_Skills Diakses pada 7 Mei 2022
- [4] [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 7 Mei 2022
- [5] <https://www.ign.com/wikis/shin-megami-tensei-persona-4-golden/Combat> Diakses pada 7 Mei 2022

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, 13 Mei 2022

Bariza Haqi 13520018