

Aplikasi Algoritma *Greedy* dalam Pengoptimalan Kombinasi *Skill Hero* di *Game Arena of Valor*

Farnas Rozaan Iraquee - 13520067
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13520067@std.stei.itb.ac.id

Abstrak—Arena of Valor merupakan sebuah mobile video games bergenre MOBA (Multiplayer Online Battle Arena) yang cukup terkenal, terutama di Asia. Game ini tidak hanya memerlukan keahlian dalam bermain, tetapi juga dibutuhkan kecerdasan, seperti memikirkan strategi yang akan digunakan dalam suatu match atau menganalisis cara bermain lawan. Oleh karena itu akan dijelaskan salah satu komponen penting yang perlu diperhatikan dalam bermain game ini, yaitu penentuan kombinasi serangan yang akan diluncurkan dengan memanfaatkan algoritma *greedy* sehingga dapat menghasilkan *damage* yang besar serta optimal.

Kata Kunci—Arena of Valor, Strategi, Serangan, Algoritma *Greedy*.

I. PENDAHULUAN

Saat ini game online merupakan hobi yang banyak digandrungi orang, mulai dari kalangan anak-anak sampai orang dewasa. Tidak hanya dimainkan sebagai hiburan atau penghilang stres/penat, game online tersebut sekarang sudah berkembang cukup pesat dan bahkan menjadi mata pencaharian bagi sebagian orang. Bidang tersebut lebih kerap disebut electronic sports (e-sports). Salah satu game online yang masuk ke e-sports adalah Arena of Valor. Sudah banyak turnamen-turnamen Arena of Valor yang diadakan mulai dari tingkat komunitas kecil, nasional, hingga internasional. Game ini sangat terkenal di dunia, terutama di daerah Asia, sehingga prize pool dari turnamen internasionalnya dapat menyentuh angka hingga miliaran rupiah. Oleh karena itu, tak heran banyak kalangan yang melirik ke bidang ini.

Kemudian untuk menjadi atlet esports Arena of Valor ataupun player biasa tentunya dibutuhkan dua skill penting, yaitu skill dalam bermain game tersebut dan skill dalam menentukan strategi bermain yang efektif. Dan salah satu aspek penting dari strategi tersebut adalah penggunaan *skill* hero yang baik dan benar serta kombinasi serangan yang dapat menghasilkan *damage* yang besar. Supaya tujuan tersebut dapat dicapai, maka akan dimanfaatkan salah satu algoritma yang diajarkan dalam mata kuliah Strategi Algoritma ini, yaitu algoritma *greedy*.

II. TEORI DASAR

A. Algoritma *Greedy*

Algoritma *greedy* merupakan algoritma yang paling populer dan sederhana untuk memecahkan persoalan optimasi. Persoalan optimasi itu sendiri merupakan persoalan mencari solusi optimal. Terdapat dua macam persoalan optimasi:

1. Maksimasi (*maximation*)
2. Minimasi (*minimation*)

Algoritma *greedy* menyelesaikan masalah dengan membentuk solusi tahap per tahap (*step by step*). Pada setiap tahap tersebut kita harus mengambil pilihan yang terbaik yang dapat diperoleh saat itu, tidak peduli apa yang terjadi di depan dan berharap pilihan yang kita ambil tersebut dapat mengantarkan kita ke solusi optimum global karena algoritma ini tidak selalu menghasilkan solusi yang paling optimal. Dengan begitu, algoritma *greedy* memiliki prinsip “*Take what you can get now!*”.

Algoritma *greedy* memiliki beberapa elemen. Elemen-elemen tersebut adalah sebagai berikut:

1. Himpunan kandidat (C)
Himpunan kandidat berisi kandidat yang akan dipilih pada setiap Langkah.
2. Himpunan solusi (S)
Himpunan solusi berisi kandidat yang sudah dipilih.
3. Fungsi solusi
Fungsi solusi digunakan untuk menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi.
4. Fungsi seleksi
Fungsi seleksi digunakan untuk memilih kandidat berdasarkan strategi *greedy* tertentu. Strategi ini bersifat heuristic.
5. Fungsi kelayakan
Fungsi kelayakan berfungsi untuk memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi.
6. Fungsi objektif
Fungsi objektif berisi tujuan dari algoritma tersebut, memaksimumkan atau meminimumkan.

Dengan menggunakan elemen-elemen di atas, maka dapat dikatakan bahwa algoritma *greedy* melibatkan pencarian

sebuah himpunan bagian, S , dari himpunan kandidat, C ; yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu S menyatakan suatu solusi dan S dioptimisasi oleh fungsi obyektif.

Berikut ini adalah beberapa 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*)

Skema umum algoritma *greedy*:

```

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 2.1 Skema Umum Algoritma *Greedy*

B. Arena of Valor

Arena of Valor merupakan sebuah mobile video games bergenre MOBA (Multiplayer Online Battle Arena) yang mempertandingkan 2 buah tim untuk saling menghancurkan tower inti musuh mereka. Berikut adalah gambar map dari arena pertandingan game ini:



Gambar 2.2 Map Arena of Valor

(sumber: <https://jurnalapps.co.id/di-aov-horizon-valley-adalah-battlefield-baru-pengganti-arena-antaris-14505>)

Map arena of valor itu memiliki berbagai bagian, yaitu toplane, midlane, dan bottomlane yang masing-masing memiliki 3 tower penjaga sebelum mencapai ke daerah tower inti yang memunculkan creep yang berfungsi untuk

mendapatkan gold dan exp (farming) ketika dibunuh dan juga untuk push tower ataupun menahan damage dari tower musuh. Kemudian ada jungle juga yang di dalamnya terdapat monster yang berfungsi untuk resources farming hero. Lalu, ada 3 monster lain yang bisa dibunuh untuk mendapatkan gold dan exp tambahan serta buff yang masing-masing monster memiliki efek yang berbeda.

Dalam upaya menghancurkan tower inti musuh dibutuhkan kerjasama yang baik dan strategi yang matang juga serta skill dalam bermain juga tentunya karena game ini tidak dihitung dari jumlah kill yang diperoleh, tetapi siapa yang lebih dulu menghancurkan tower inti (core) musuh yang menang.



Gambar 2.3 Tower Inti (Core) (sumber: game Arena of Valor)

Berikut adalah komposisi role hero yang harus dimiliki setiap tim:

1. Jungler

Jungler merupakan carry di dalam Arena of Valor. Role jungler bertugas untuk membersihkan creep yang ada di jungle sehingga lebih cepat untuk naik level dan mendapatkan gold lebih banyak serta mendapatkan 2 buff yang sangat penting bagi role jungler ini. Tugas jungler berikutnya adalah melakukan gank atau pick off hero musuh, terutama hero seperti archer atau mage, dan membantu team fight. Selain itu, jungler juga harus mengambil objektif yang ada pada game seperti membunuh abyssal dragon, dark slayer, atau spirit sentinel agar tim mendapatkan buff serta gold dan exp tambahan, atau push tower, apabila memungkinkan. Hero yang bisa digunakan pada role ini adalah assassin, archer, warrior, ataupun mage. Namun, yang umum dipakai adalah assassin atau archer.

2. Abyssal Dragon Laner (ADL)

Role ini bertugas untuk menjaga lane di dekat tempat spawn Abyssal Dragon. Pada umumnya hero yang mengisi role ini adalah hero archer atau warrior yang memiliki damage output yang tinggi. Tugas dari ADL adalah menjaga tower yang ada pada lane tersebut, mengawasi abyssal dragon agar tidak dibunuh musuh (dibantu oleh support), membantu rekan setim untuk membunuh abyssal dragon, serta membantu team fight karena damage

output tertinggi biasanya terletak di role ini. Dengan memiliki damage yang tinggi, role ini juga merupakan carry selain jungler, tetapi jungler biasanya memiliki impact lebih banyak di early - midgame sedangkan ADL mid - lategame.

3. Midlaner

Midlaner adalah role yang berada di lane bagian tengah. Biasanya diisi oleh hero mage. Tugas dari role ini adalah menjaga lane tengah dan segera membersihkan creep agar dapat melakukan roaming lebih cepat untuk membantu lane Abyssal Dragon ataupun Dark Slayer apabila dibutuhkan. Biasanya roaming dilakukan bersama-sama dengan support apabila support tidak sedang menjaga archer di lane Abyssal Dragon. Dengan damage magic yang tinggi, midlaner dapat membantu team fight, gank, ataupun mengambil objektif.

4. Dark Slayer Laner (DSL)

Role ini bermain pada lane di dekat Dark Slayer. Hero yang mengisi role ini biasanya adalah warrior ataupun tank yang memiliki damage yang cukup tinggi, tetapi memiliki defense yang cukup kuat juga. Tugasnya adalah menjaga tower di lane tersebut, mengawasi Dark Slayer atau Spirit Sentinel, melakukan split push (push tower di tempat yang berlawanan dengan tempat terjadinya team fight) apabila memungkinkan (dilakukan di mid - lategame), dan juga membantu team fight apabila dibutuhkan karena pada early - midgame biasanya war terjadi di lane Abyssal Dragon ataupun mid. Oleh karena itu, skill individu DSL yang baik sangat dibutuhkan untuk bisa 1v1 dengan DSL musuh.

5. Support / Observer

Support bertugas untuk membuka map dan mencari tahu keberadaan player musuh di map serta menjaga carry yang berada di tim terhindar dari upaya gank dari lawan. Biasanya archer sangat rentan untuk id-gank sehingga para pemain support seringkali menjaga carry, terutama ADL, dengan sekuat tenaga. Hero yang digunakan pada role ini biasanya adalah hero yang memiliki skillset yang sangat berguna dalam team fight, seperti memiliki skill crowd control (CC) ataupun buff bagi rekan setim. Role ini biasanya tidak mempunyai damage yang cukup untuk membunuh musuh sehingga role ini hanya bertugas untuk menciptakan ruang untuk bisa menyerang musuh ataupun melindungi teman musuh dari serangan tim musuh. Role ini cenderung susah untuk dimainkan karena tidak hanya diperlukan skill bermain yang mumpuni, tetapi harus bisa memikirkan strategi yang efektif untuk menaklukkan musuh seperti apa.

Dalam makalah ini akan dijelaskan bagaimana cara

melancarkan kombinasi serangan yang optimal. Optimal dalam artian dapat menghasilkan *damage* yang besar baik ke *hero* musuh, *monster*, ataupun *tower*. Karena penyerangan yang optimal tersebut merupakan kunci kemenangan dari game ini. Meskipun, banyak aspek lain yang perlu diperhatikan seperti penentuan role hero dalam game dan lain sebagainya.

III. PEMBAHASAN

A. Algoritma Greedy Pengoptimalan Kombinasi Skill

Pada umumnya, setiap *hero* pada game Arena of Valor memiliki 4 *skill* yang berbeda yang terdiri dari 1 *skill* pasif dan 3 *skill* aktif, meskipun ada yang memiliki lebih dari itu. *Skill* dari setiap *hero* juga berbeda satu sama lain. Untuk dapat menghasilkan *damage* yang besar, cara dari setiap *hero* berbeda-beda. Ada yang memanfaatkan *skill* dan ada juga yang memanfaatkan *attack* normal. Pada pembahasan kali ini, algoritma *greedy* digunakan untuk mengoptimalkan *damage* yang dihasilkan dari kombinasi *skill* bukan *attack* normal.

Salah satu algoritma *greedy* yang dapat digunakan adalah strategi menggunakan *skill* aktif berdasarkan prioritas. Berikut adalah prioritas urutan penggunaan *skill* aktif mulai dari prioritas tertinggi:

1. Menggunakan *skill movement* yang dapat mendekati musuh (opsional, tidak perlu dilakukan apabila musuh masih berada di jangkauan penyerangan kita atau musuh sudah terbunuh)



Gambar 3.1 Skill 2 Nakroth (Skill Movement)
(sumber: Youtube AoV)

2. Menggunakan *skill* yang memiliki efek *crowd control*. Efek *crowd control* sendiri dapat berupa *stun*, *slow*, *knock back*, *knock up*, *silent*, ataupun *suppress*.



Gambar 3.2 Skill 3 Tel'annas (Skill Crowd Control)
(sumber: Youtube AoV)

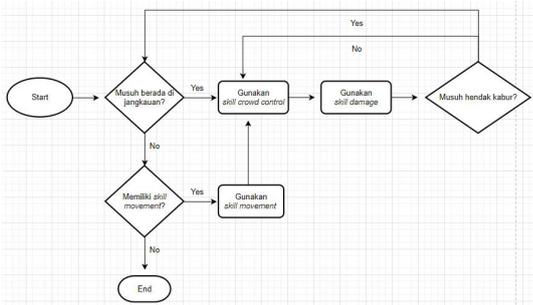
3. Menggunakan *skill* yang menghasilkan *damage* dengan *range* jauh ataupun dekat. Dengan catatan, *skill* dengan *range* jauh biasa digunakan ketika musuh berusaha kabur sehingga jarak antara kita dengan musuh menjadi cukup jauh, sedangkan *skill* dengan *range* dekat biasa digunakan ketika musuh masih berada di jarak yang dekat dengan kita.



Gambar 3.3 Skill 1 Liliana (Skill Damage)
(sumber: Youtube AoV)

Setiap *skill* memiliki durasi *cooldown* yang berbeda-beda. Setelah *skill* digunakan, *skill* tersebut akan masuk ke masa *cooldown* tersebut. Sebelum durasi *cooldown* menyentuh angka nol, maka *skill* tersebut tidak akan bisa dipakai. Apabila dengan kombinasi *skill* di atas musuh masih dapat bertahan, maka gunakan *skill* yang tersedia terlebih dahulu. Apabila beberapa *skill* tersedia dalam waktu yang relatif bersamaan, gunakan *skill* berdasarkan urutan prioritas di atas.

Agar lebih mudah untuk dipahami, berikut adalah *flowchart* untuk algoritma di atas. *Flowchart* ini memiliki asumsi bahwa *cooldown* selesai di waktu yang relatif bersamaan sehingga *skill* dapat selalu digunakan berdasarkan urutan prioritas di atas



Gambar 3.4 Flowchart Algoritma

Seperti yang sudah dijelaskan sebelumnya, algoritma *greedy* memiliki beberapa elemen. Berikut adalah uraian elemen-elemen tersebut berdasarkan algoritma *greedy* yang digunakan:

1. Himpunan Kandidat
Himpunan kandidat dari kasus ini adalah 4 skill yang dimiliki hero yang dipilih.
2. Himpunan solusi
Himpunan solusi dari kasus ini berisi skill yang dipilih untuk menyerang.
3. Fungsi solusi
Fungsi solusi pada kasus ini yaitu memeriksa apakah serangan sudah optimal.
4. Fungsi seleksi
Fungsi seleksi dari kasus ini yaitu memilih skill dengan *stun* terlebih dahulu yang mana kemudian dilanjutkan dengan skill dengan damage terbesar.
5. Fungsi kelayakan
Fungsi kelayakan mengecek apakah solusi dapat dimasukkan ke dalam himpunan seleksi.
6. Fungsi objektif
Fungsi objektif dari kasus ini adalah memaksimalkan damage yang dapat diberikan ke lawan.

B. Implementasi Algoritma

Berikut ini adalah contoh implementasi algoritma *greedy* yang dituliskan di subbab sebelumnya. Hero yang akan digunakan pada contoh implementasi algoritma adalah Natalya dan Arthur.

1. Natalya



Gambar 3.5 Hero Natalya

(sumber: <https://arenaofvalor.fandom.com/wiki/Natalya>)

Berikut ini adalah rincian *skill* yang dimiliki oleh Natalya:

a. Arcane Fire

Skill menimbulkan 11(+2% dari *Magic Power*) *Magic Damage* pada musuh dan dapat ditumpuk hingga 10 kali dalam rentang waktu 3 detik. *Skill* ini merupakan *skill* pasif dan tiga lainnya merupakan *skill* aktif.

b. Arcane Spirits

Natalya memanggil 5 *arcane spirit* untuk menyerang area target. Tiap *spirit* menimbulkan 350(+30% dari *magic power*) *magic damage* pada musuh. Saat ada musuh yang terkena *hit* beberapa *spirit*, tiap *spirit* lainnya hanya menimbulkan 30% *damage*. *Spirit* memulihkan 10 mana dan mengurangi *cooldown skill* ini sebanyak 0,6 detik saat mengenai hero musuh. *Cooldown skill* ini adalah 6 detik. *Base damage* dari *skill* ini akan meningkat tiap levelnya. Urutan peningkatannya adalah sebagai berikut:

	Lvl 1	Lvl 2	Lvl 3	Lvl 4	Lvl 5	Lvl 6
DMG	350	390	430	470	510	550

Tabel 3.1 Peningkatan *Base Damage* per Level
(Natalya)

c. Arcane Nova

Natalya menembakkan *arcane orb* ke arah target. Bola tersebut menjadi *arcane nova* dan meng-*stun* musuh yang terkena *hit* selama 1 detik. Kemudian bola membesar dan terbang perlahan untuk menimbulkan 130(+15% dari *magic power*) *magic damage* saat bersinggungan dengan musuh, mengurangi *movement speed* mereka selama 50% selama 1,5 detik. *Cooldown skill* ini adalah 10 detik yang akan berkurang setiap naik level. Seperti halnya pada *skill* Arcane Spirits, *base damage* dari *skill* ini juga akan meningkat tiap levelnya. Urutan peningkatan dan pengurangannya adalah sebagai berikut:

	Lvl 1	Lvl 2	Lvl 3	Lvl 4	Lvl 5	Lvl 6
DMG	130	145	160	170	190	205
CD	10	9,6	9,2	8,8	8,4	8

Tabel 3.2 Peningkatan *Base Damage* serta Pengurangan Waktu *Cooldown* per Level (Natalya)

d. Lethal Rays

Natalya memerintahkan Poison Fang untuk menembakkan sinar mematikan yang menimbulkan 120(+15% dari *magic power*) *magic damage* tiap 0,25 detik pada musuh. Selama *channeling*, Natalya menerima *shield* yang menyerap 850(+100% dari *magic power*) *damage*. Natalya kebal terhadap efek *crowd control* selama *shield* aktif. *Skill* ini bersifat *channeling* dan memiliki durasi maksimum sebanyak 3,5 detik. *Channeling* dapat dibatalkan selang 0,5 detik setelah pengaktifan dengan menekan tombol *skill* kembali. *Cooldown skill* ini adalah 30 detik. *Base damage* dan bonus *shield* akan meningkat serta waktu *cooldown* akan berkurang tiap levelnya. Urutan peningkatan dan pengurangannya adalah sebagai berikut:

	Lvl 1	Lvl 2	Lvl 3
DMG	120	150	180
CD	30	26	22
<i>Shield</i>	850	1275	1700

Tabel 3.3 Peningkatan *Base Damage* dan Bonus *Shield* serta Pengurangan Waktu *Cooldown* per Level (Natalya)

Berdasarkan algoritma *greedy* yang sudah dirumuskan, berikut adalah urutan penggunaan *skill hero* Natalya ini:

1. Arcane Nova
2. Arcane Spirits
3. Lethal rays

Dengan asumsi *magic power* sebesar 500, level sudah maksimal, dan *skill* selalu mengenai *hero* musuh maka berikut adalah penjelasan dari masing-masing *skill* yang digunakan.

Skill yang digunakan pertama kali, yaitu Arcane Nova, memberikan efek *stun* kepada *hero* musuh dan juga memberikan *damage* kepada *hero* musuh. Selain itu, *movement speed* musuh juga berkurang. Besar *damage* yang dihasilkan adalah 280. *Skill* yang digunakan berikutnya adalah Arcane Spirits. Apabila kelima *spirit* mengenai musuh, maka besar *damage* yang dihasilkan adalah 1540. Selain menghasilkan *damage* Natalya juga memulihkan mana serta mengurangi *cooldown skill* ini. *Skill* yang terakhir digunakan adalah Lethal Rays. Apabila *skill* ini digunakan secara terus menerus selama 3,5 detik, maka *damage* yang dihasilkan adalah sebesar 2730. Selama itu, Natalya juga mendapatkan *shield*. Karena penggunaan *skill* selalu mengenai musuh maka *skill* pasifnya, Arcane Fire, menghasilkan *damage* tambahan sebesar 210. Sehingga jumlah *damage* yang dihasilkan dari seluruh *skill*

yang digunakan adalah sebesar 4760. Dengan *damage* sebesar ini, *hero* dengan HP dan *defense* yang kecil akan terbunuh seketika.

Hasil yang ditunjukkan di atas merupakan hasil yang paling optimal. Dalam kasus ini algoritma *greedy* berhasil membentuk solusi yang optimal.

2. Arthur



Gambar 3.6 Hero Arthur

(sumber: <https://arenaofvalor.fandom.com/wiki/Arthur>)

Berikut ini adalah rincian *skill* yang dimiliki oleh Arthur:

1. Paragon

Arthur diberkati kekuatan surgawi dan menerima 2 *armor* (bertambah sesuai level *hero*). *Skill* ini merupakan *skill* pasif dan tiga lainnya merupakan *skill* aktif.

2. Righteous Fervor

Arthur memperkuat diri, meningkatkan *movement speed*-nya sebesar 30% dan memperkuat *attack normal* berikutnya (setelah efek *attack normal* berakhir, maka akan memasuki *cooldown*). Di *attack normal* berikutnya, dia menimbulkan 180(+100% dari *attack damage*) *physical damage*, *stun* pada target selama 0.25 detik dan *slow* sebesar 20% selama 1 detik. Dia juga menerima bonus *shield* berdaya 500(+5% dari HP maks-nya) saat menghantam *hero* musuh. *Hero* yang terkena hantaman akan ditandai dan musuh yang bertanda akan menerima *damage* lebih banyak dari *attack normal* dan *skill* Arthur sebesar 1% dari HP maks mereka disertai *magic damage*. *Skill* ini juga meningkatkan *movement speed hero* kawan sekitar selama 5 detik. *Cooldown skill* ini adalah 7 detik. *Base damage* akan meningkat dan waktu *cooldown* akan berkurang tiap levelnya. Urutan peningkatan dan pengurangannya adalah sebagai berikut:

	Lvl 1	Lvl 2	Lvl 3	Lvl 4	Lvl 5	Lvl 6
DMG	180	205	230	255	280	305
CD	7	6,4	5,8	5,2	4,6	4

Tabel 3.4 Peningkatan *Base Damage* serta Pengurangan Waktu *Cooldown* per Level (*Skill* 1 Arthur)

3. Holy Guard

Arthur memanggil pedang suci yang mengitarinya

selama 5 detik, yang menimbulkan 50(+35% dari *attack damage*) *physical damage* tiap 0,5 detik. Selain itu, dia juga menerima *shield* yang mampu menyerap *damage* sebanyak 5% dari HP maksimum selama 2 detik. *Cooldown skill* ini adalah 10 detik. *Base damage* akan meningkat dan waktu *cooldown* akan berkurang tiap levelnya. Urutan peningkatan dan pengurangannya adalah sebagai berikut:

	Lvl 1	Lvl 2	Lvl 3	Lvl 4	Lvl 5	Lvl 6
DMG	50	65	80	95	110	125
CD	5%	6%	7%	8%	9%	10%

Tabel 3.5 Peningkatan *Base Damage* serta Pengurangan Waktu *Cooldown* per Level (*Skill 2* Arthur)

4. Deep Impact

Arthur melompat tinggi dan menghantam *hero* musuh, menimbulkan *magic damage* sebesar 14% dari HP maksimum target dan menghempaskan musuh sekitar ke udara selama 0,5 detik. Musuh yang berada dalam radius hantaman akan menerima 100 (+35% dari *attack damage*) *magic damage* per detik selama 5 detik. *Cooldown skill* ini adalah 42 detik. *Base damage* dan *burn damage* akan meningkat serta waktu *cooldown* akan berkurang tiap levelnya. Urutan peningkatan dan pengurangannya adalah sebagai berikut:

	Lvl 1	Lvl 2	Lvl 3
Base	14%	17%	20%
CD	42	35	28
Burn	100	125	150

Tabel 3.6 Peningkatan *Base Damage* dan *Burn Damage* serta Pengurangan Waktu *Cooldown* per Level (Arthur)

Berdasarkan algoritma *greedy* yang sudah dirumuskan, berikut adalah urutan penggunaan *skill hero* Arthur ini:

1. Deep Impact
2. Righteous Fervor
3. Attack normal
4. Holy Guard

Dengan asumsi *attack damage* sebesar 300, HP sebesar 4000, HP musuh sebesar 2500, level sudah maksimal, dan *skill* selalu mengenai *hero* musuh maka berikut adalah penjelasan dari masing-masing *skill* yang digunakan.

Skill yang digunakan pertama kali adalah Deep Impact. *Skill* ini memberikan efek *stun* kepada musuh serta memberikan *damage* sebesar 755. Kemudian, akan digunakan *skill* Righteous Fervor dan dilanjutkan dengan *attack* normal. Serangan ini akan menghasilkan *damage* sebesar 605. Karena efek *burn* dari *skill* pertama masih ada dan asumsi efek *burn* tersisa 4 detik, maka *burn damage* yang dihasilkan adalah sebesar 1120. Dan *skill* terakhir adalah Holy Guard yang menghasilkan *damage* sebesar 2325 selama 10 detik. Sehingga total *damage* yang dihasilkan secara keseluruhan adalah 4805.

Hasil yang ditunjukkan di atas bukan merupakan hasil yang paling optimal. Dalam kasus ini algoritma *greedy* tidak berhasil membentuk solusi yang optimal. Untuk hasil yang paling optimal adalah dengan menggunakan *skill* Righteous Fervor terlebih dahulu karena setelah kita menggunakan *skill* tersebut dan dilanjutkan dengan *attack* normal maka semua *damage* yang masuk ke *hero* musuh akan bertambah. Sehingga urutan penggunaan *skill* yang optimal adalah:

1. Righteous Fervor
2. Attack Normal
3. Deep Impact
4. Holy Guard

Total *damage* yang dihasilkan dari urutan penggunaan *skill* di atas adalah sebesar 4855. Meskipun tidak berbeda jauh, tetapi ini dapat membuktikan bahwa algoritma *greedy* tidak selalu menghasilkan solusi yang paling optimal.

IV. KESIMPULAN

Dengan adanya bantuan algoritma *greedy*, banyak permasalahan yang timbul di kehidupan sehari-hari dapat dengan mudah terselesaikan. Seperti halnya topik yang dibahas di makalah ini. Meskipun terkesan tidak berguna, tetapi nyatanya hal tersebut dapat berguna, terlebih bagi pro player yang bermain di level turnamen.

Algoritma *greedy* tidak selalu memberikan hasil pemilihan yang optimal pada semua *hero*. Hal ini dikarenakan prinsip *greedy* yang bergantung pada kondisi saat ini tanpa mempertimbangkan konsekuensi kedepan. Selain itu, jenis *skill* pada masing-masing *hero* dan waktu *cooldown skill* tersebut juga menjadi salah satu penyebab ketidakefektifan ini. Karena terdapat *hero* yang bukan mengandalkan *skill* melainkan *attack* normalnya.

V. UCAPAN TERIMA KASIH

Penulis berterima kasih kepada Allah SWT karena rahmat dan kesempatan yang sudah diberikan oleh-Nya, penulis dapat menyelesaikan makalah ini, dan penulis berterima kasih kepada pihak luar terutama komunitas Arena of Valor karena membantu penulis dalam penulisan makalah berjudul “Aplikasi Algoritma *greedy* dalam Pengoptimalan Kombinasi *Skill Hero* di Game Arena of Valor”. Selain itu, penulis juga berterima kasih pada Bapak Rinaldi Munir, Ibu Masayu Leyla Khodra, serta Ibu Nur Ulfa Maulidevi selaku dosen pengampu mata kuliah IF2211 Strategi Algoritma pada tahun pelajaran kali ini.

REFERENSI

- [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 20 Mei 2022
- [2] <https://arenaofvalor.fandom.com/wiki/Natalya> diakses pada 20 Mei 2022
- [3] <https://arenaofvalor.fandom.com/wiki/Arthur> diakses pada 20 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, 23 Mei 2022

A handwritten signature in black ink, consisting of stylized, cursive letters that appear to read 'Farnas' followed by a horizontal line and a flourish.

Farnas Rozaan Iraquee 13520067