

Penerapan Algoritma *Greedy* untuk Optimasi Pembelian *Item* dalam *Training* pada Gim *Uma Musume: Pretty Derby*

Dzaky Fattan Rizqullah - 13520003

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 13520003@std.stei.itb.ac.id

Abstract—*Uma Musume: Pretty Derby* adalah gim *social-simulation* berbasis *Visual Novel* yang memosisikan pemain sebagai pelatih yang mempunyai misi untuk melatih “gadis kuda” (*uma musume*) pada mode *training* agar dapat berkompetisi dalam pacuan gadis kuda. *Training items* adalah *items* yang dapat dibeli saat melakukan *training* yang dapat membantu pemain dalam melatih gadis kuda. Untuk membeli *items* tersebut, dibutuhkan suatu *coin* yang hanya dapat diperoleh saat gadis kuda memenangkan suatu balapan. Keterbatasan ini membuat pemain harus mengoptimasi penggunaan *coin* agar dapat memaksimalkan penggunaan *items* dan memaksimalkan hasil keluaran gadis kuda setelah dilatih. (Abstract)

Keywords—Algoritma, Greedy, Optimasi, *Uma Musume: Pretty Derby*, *Training*.

I. PENDAHULUAN

Gim bergenis *simulation* merupakan salah satu jenis gim yang sangat menarik dan menantang untuk dimainkan. Tidak jarang saat bermain gim bergenis ini, pemain tidak hanya sekedar “bermain”, tetapi harus memahami konsep dan mekanik gim dengan baik dan tidak jarang pula harus berpikir keras untuk menyelesaikan objektif dari gim tersebut. Ada banyak sekali *website* yang dibangun khusus untuk memberikan pemahaman serta alternatif strategi untuk menyelesaikan objektif dari gim-gim bergenis simulasi. Salah satu variasi dari game simulasi adalah *social-simulation* yang menekankan hubungan interaksi pemain dengan karakter di dalam gim. Variasi ini biasanya mengambil suatu tema, misalnya berkarir di dunia olahraga, *entertainment*, dan sebagainya.

Uma Musume: Pretty Derby merupakan salah satu jenis gim bergenis *social-simulation* yang dirilis oleh *Cygames*. Gim ini mengambil tema olahraga dengan sedikit modifikasi, tentunya untuk menarik minat pemain dan menyesuaikan target pasar. *Gameplay* utama dari gim ini ialah pada mode *Training*, yang mana pemain harus melatih makhluk fiksi “gadis kuda”—identitas mereka diadaptasi dari kuda versi dunia nyata—dan menemani mereka dalam ajang kompetisi fiksi “pacuan gadis kuda” hingga mencapai kemenangan.



Gambar 1.1: Icon gim “*Uma Musume: Pretty Derby*” saat event “Golden Week”.

(Sumber: <https://apps.qoo-app.com/en/app/6172>)

Fokus dari makalah ini ialah pada aspek penggunaan *items* pada mode *training* yang dapat membantu pemain saat melatih gadis kuda. *Items* ini hanya dapat diperoleh menggunakan suatu *coin* yang hanya didapat ketika gadis kuda memenangkan suatu balapan. Keterbatasan *items* ini membuat pemain harus mengoptimisasi penggunaan *coin* tersebut agar dapat memaksimalkan penggunaan *items* dan memaksimalkan keluaran gadis kuda saat selesai dilatih.

II. LANDASAN TEORI

A. Algoritma Greedy

Algoritma *greedy* adalah algoritma yang paling populer dan sangat sederhana untuk memecahkan persoalan berbasis optimasi. Algoritma ini memecahkan persoalan secara terurut langkah per langkah tanpa kembali ke langkah sebelumnya, dengan setiap langkahnya diberikan opsi tindakan yang dapat diambil. Tindakan yang terbaik saat itu, disebut optimum lokal, dipilih, dengan harapan langkah-langkah seterusnya akan menuju solusi terbaik secara keseluruhan, atau disebut optimum global. Sesuai namanya, algoritma *greedy* ini bersifat “rakus”, karena pada setiap langkahnya algoritma akan langsung memilih tindakan yang terbaik saat itu juga, tanpa memerhatikan konsekuensi pada langkah-langkah selanjutnya.

Algoritma *greedy* ini memiliki keunggulan dan juga kelemahan. Keunggulannya ialah algoritma ini berjalan sangat cepat dan bersifat sangat sederhana dibandingkan algoritma penyelesaian masalah optimasi lainnya, sehingga sering digunakan. Kekurangannya, algoritma ini tidak selalu menghasilkan solusi optimal, melainkan solusi yang menekati

optimal (*pseudo-optimum*). Meskipun demikian, untuk permasalahan yang memiliki kompleksitas sangat rumit sehingga mencari solusi eksak akan membutuhkan waktu yang sangat lama, solusi ini tetap dapat diterima. Persoalan optimasi hanya terdiri dari dua jenis, yaitu persoalan maksimasi dan persoalan minimasi.

Algoritma *greedy* memiliki beberapa elemen yang penjelasannya adalah sebagai berikut.

- Himpunan kandidat (C), merupakan himpunan yang memiliki elemen-elemen kandidat yang akan dipilih pada setiap langkah.
- Himpunan solusi (S), merupakan himpunan yang berisi kandidat-kandidat yang sudah dipilih. Himpunan ini adalah himpunan bagian dari himpunan kandidat.
- Fungsi solusi, yaitu fungsi yang menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi.
- Fungsi seleksi (selection function), merupakan fungsi yang memilih kandidat pada himpunan kandidat yang mengimplementasikan suatu strategi *greedy* yang bersifat heuristik.
- Fungsi kelayakan (feasible), merupakan fungsi yang memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi tanpa melanggar batasan-batasan tertentu yang sudah ditetapkan.
- Fungsi objektif, yaitu fungsi yang berkaitan langsung dengan persoalan optimasi yang ingin di selesaikan, apakah itu memaksimalkan ataupun meminimumkan.

Algoritma *greedy* ini sering digunakan sebagai alternatif untuk penyelesaian masalah optimasi yang membutuhkan waktu penyelesaian secara cepat. Beberapa contoh persoalan yang dapat diselesaikan dengan algoritma *greedy* adalah persoalan penukaran uang, pemilihan aktivitas, persoalan *knapsack*, lintasan terpendek, dan sebagainya.

Lebih jelasnya, algoritma *greedy* ditulis dalam *pseudo-code* menjadi seperti berikut.

```
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
```

B. Persoalan Knapsack

Persoalan *Knapsack* adalah salah satu persoalan optimasi yang tujuannya memaksimalkan keuntungan. Misalkan terdapat sekumpulan objek yang masing-masing memiliki bobot (*weight*) dan nilai (*profit*). Objek-objek ini harus dimasukkan ke dalam suatu media penyimpanan sedemikian sehingga didapat total nilai terbesar tanpa melebihi kapasitas dari media penyimpanan itu sendiri. Terdapat dua jenis persoalan *knapsack*, yaitu integer *knapsack problem* atau 1/0 *knapsack problem* dan *fractional knapsack problem*.

Persoalan *knapsack* dapat diselesaikan dengan algoritma *greedy*. Dengan memanfaatkan suatu strategi *greedy* heuristik tertentu, objek dimasukkan satu per satu tanpa mengeluarkan objek yang sudah dimasukkan sebelumnya. Strategi heuristik yang dapat digunakan adalah sebagai berikut. Perlu diketahui bahwa seluruh strategi di bawah tidak ada yang menjamin akan memberikan solusi yang paling optimal.

- *Greedy by profit*, yaitu strategi yang mencoba memaksimalkan keuntungan dengan memilih objek dengan nilai paling besar terlebih dahulu. Bila ada nilai yang sama, maka diutamakan *weight* terkecil terlebih dahulu
- *Greedy by weight*, yaitu strategi yang mencoba memaksimalkan keuntungan dengan memilih objek dengan berat paling ringan terlebih dahulu, yang artinya strategi ini akan memasukkan objek sebanyak-banyaknya sehingga diharapkan solusi yang optimal. Bila ada *weight* yang sama, maka diutamakan nilai terbesar terlebih dahulu.
- *Greedy by density*, yaitu strategi yang mencoba memaksimalkan keuntungan dengan memilih objek dengan densitas terbesar terlebih dahulu, dalam hal ini densitas adalah rasio antara nilai dengan berat suatu objek. Bila ada *density* yang sama, maka diutamakan nilai terbesar terlebih dahulu.

Integer knapsack problem adalah variasi dari persoalan *knapsack* dengan syarat objek yang dimasukkan harus utuh, tidak merupakan fraksi. Oleh karenanya, seluruh strategi heuristik yang sudah dijelaskan sebelumnya tidak akan menjamin akan memberikan solusi yang paling optimal.

C. Uma Musume: Pretty Derby

Uma Musume: Pretty Derby adalah gim *social-simulation* berbasis *Visual Novel* yang memosisikan pemain sebagai pelatih yang dapat berinteraksi dengan makhluk fiksi “gadis kuda”. Pemain dapat berbincang dengan mereka, membaca cerita mereka, menikmati lagu dan *live performance* mereka, dan yang terpenting ialah melatih mereka dalam ajang kompetisi fiksi “pacuan gadis kuda”. Gadis kuda keluaran latihan ini dapat digunakan dalam *event* lainnya, seperti

balapan harian, PvP dengan pemain lain dalam mode *Team Stadium*, mengikuti *event* dan kompetisi bulanan, dan lain-lain.



Gambar 2.1: Contoh tampilan antarmuka gim, (a) menu utama, (b) menu *Gacha*, (c) mode *Training*, (d) mode *Team Stadium*
(Sumber: Arsip Penulis)

1) Gameplay

Mode utama dari gim ini ialah mode *Training*, yaitu mode yang mengharuskan pemain untuk memilih salah satu gadis kuda yang telah didapatkan, memilih dua gadis kuda lain dari hasil *Training* sebelumnya sebagai *parent*, menyusun dek kartu *support* yang berisi enam buah kartu, dan melatih gadis kuda tersebut dengan durasi selama tiga tahun penuh waktu dalam gim, yang disederhanakan menjadi 72 *turn*. Semakin cocok dan baik dua gadis kuda serta enam kartu *support* yang dipakai, maka akan semakin baik pula kualitas gadis kuda keluaran yang diharapkan.



Gambar 2.2: Balapan sebagai salah satu bagian dari alur *Training*
(Sumber: Arsip Penulis)

Gadis kuda keluaran *Training* akan ditampilkan pada menu *Hall of Fame*. Keluaran gadis kuda yang baik biasanya akan dipakai untuk balapan di luar *Training* dan/atau digunakan sebagai salah satu *parent* untuk gadis kuda yang ingin dilatih selanjutnya.

Terdapat tiga skenario *Training* yang dapat dipilih, yaitu “*URA Finals*”, skenario yang lebih mudah dan cocok untuk pemain kasual dan menengah; “*Aoharu Cup*”, skenario yang lebih sulit dan berfokus pada organisasi dan balapan tim; dan “*Make a New Track*”, skenario yang memberikan pemain kebebasan dalam memerintahkan gadis kuda untuk melakukan apapun sesuai opsi yang dapat dipilih sehingga skenario ini sangat tepat untuk mendapatkan keluaran gadis kuda yang maksimal.

2) Status pada Gadis Kuda

Setiap gadis kuda dapat diukur kemampuannya dengan memperhatikan status kompetensi yang dimiliki kuda tersebut. Kompetensi yang dimiliki gadis kuda ini mencakup beberapa aspek, yang dapat dibagi menjadi lima bagian, yaitu sebagai berikut.

a) *Status utama*, yaitu ukuran numerik kemampuan fisik dan psikis gadis kuda, yang nilainya antara 0 hingga 1200. Status utama ini terbagi lagi menjadi lima, yaitu:

- *Speed*, berkaitan dengan kecepatan maksimal gadis kuda, yang batasnya dapat ditembus saat gadis kuda tersebut berlari sekuat tenaga saat *last spurt*.
- *Stamina*, berkaitan dengan total stamina yang dimiliki gadis kuda, menentukan seberapa jauh gadis kuda tersebut mampu berlari.
- *Power*, berkaitan dengan kekuatan gadis kuda dalam menambah kecepatan, mengambil lajur, ataupun naik-turun bukit pada trek balapan.
- *Guts*, berkaitan dengan “kekuatan tambahan” gadis kuda saat berlari sekuat tenaga saat *last spurt*, serta “penangkal skill” lawan dengan tingkat tertentu.
- *Wisdom*, berkaitan dengan kemampuan gadis kuda mempertahankan posisi baik saat balapan, mengurangi kemungkinan *hanging* (kondisi di mana gadis kuda tidak sengaja menambah kecepatan di saat yang tidak perlu yang menyebabkan stamina terbuang sia-sia), serta memperbesar kemungkinan aktivasi *skill* yang dimiliki kuda tersebut.

b) *Aptitudes*, yaitu preferensi gadis kuda terhadap klasifikasi balapan yang dapat ia ikuti. Preferensi ini diukur menggunakan sistem *ranking* alfabetik terurut dari A (sangat baik) hingga G (sangat buruk). Preferensi yang bukan A dapat ditingkatkan menjadi maksimal A dengan memilih *parent* gadis kuda yang memiliki *factor* yang diinginkan, serta dapat ditingkatkan lebih lanjut menjadi S (*rank* tertinggi setelah A) saat *Training*. Preferensi ini terbagi lagi menjadi tiga, yaitu:

- Preferensi trek, yaitu antara tanah berumput atau tanah lumpur. Mayoritas gadis kuda memiliki preferensi tanah berumput.
- Preferensi jarak balapan, terbagi menjadi empat, menyesuaikan tipe jarak balapan yang ada dalam gim, yaitu *Short* ($\leq 1400m$), *Mile* (1401-1800m), *Medium* (1801-2400m), dan *Long* ($> 2400m$).
- Preferensi strategi balapan, yaitu posisi yang diambil gadis kuda saat balapan berlangsung, terbagi menjadi empat posisi, dari paling depan hingga paling belakang, yaitu *Runner/Escape*, *Leader/Preceding*, *Betweenner/Insertion*, dan *Chaser/Drive-in*.

c) *Training Bonus*, yaitu kecenderungan peningkatan satu atau lebih status utama yang dapat diperoleh gadis kuda dalam *Training*. Nilainya ditentukan dalam persentase yang bernilai 30% dan disebar dalam lima aspek dalam status utama milik gadis kuda.

d) *Skill*, yaitu keahlian yang dapat diaktifkan gadis kuda yang akan membantunya di tengah balapan. Setiap gadis kuda memiliki skill khas mereka masing-masing, dan beberapa skill umum harus “dibeli” dengan poin skill yang dapat diperoleh selama *Training*. Hampir setiap skill memiliki dua tingkat (normal dan rare).

e) *Factor*, dalam hal ini ialah stat yang dimiliki oleh gadis kuda parent yang dapat diwariskan untuk gadis kuda yang akan dilatih selanjutnya.

Keseluruhan aspek status yang dimiliki gadis kuda yang satu dengan yang lainnya bersifat unik, namun tidak menurut kemungkinan ada. Memperhatikan status yang dimiliki gadis kuda yang akan dilatih akan berpengaruh pada pemilihan *parent* gadis kuda dan kartu *support* serta akan memiliki gaya *Training* yang unik pula.

3) Item dalam Skenario “Make a New Track”

Item adalah objek eksklusif dalam skenario “Make a New Track” yang dapat diperoleh dan dipakai pemain untuk membantu mereka dalam melatih gadis kuda. Saat makalah ini ditulis, terdapat 53 jenis *items* yang dapat diperoleh melalui penukaran *coin*. *Coin* ini hanya dapat diperoleh ketika gadis kuda memenangkan suatu balapan.



Gambar 2.3: Shop pada mode Training
(Sumber: Arsip Penulis)

Referensi [6] memberikan daftar seluruh *items* yang dapat diperoleh dan dipakai. Keseluruhannya terbagi menjadi tujuh kategori. Beberapa item memberikan efek yang sama namun dengan nilai yang berbeda.

- *Energy / Motivation Recovery*, berkaitan dengan penambahan energi dan motivasi gadis kuda untuk berlatih.
- *Status*, berkaitan dengan status utama yang dimiliki gadis kuda.
- *Bond Gauge*, berkaitan dengan tingkat persahabatan dengan karakter *support*. *Bond Gauge* atau tingkat persahabatan adalah suatu ukuran yang bila sudah terisi

80% akan mengaktifkan efek yang dapat meningkatkan nilai penambahan status saat *training*.

- *Good Condition*, berkaitan dengan pemberian kondisi baik pada gadis kuda.
- *Bad Condition*, berkaitan dengan pemulihan kondisi buruk pada gadis kuda.
- *Training*, berkaitan dengan objek yang dapat membantu latihan gadis kuda secara langsung.
- *Race*, berkaitan dengan *bonus* yang diperoleh setelah menyelesaikan balapan.

Items ini dapat dibeli pada *shop* yang akan dibuka dalam *Training* setelah menyelesaikan balapan wajib paling pertama. Jumlah maksimal *item* yang dapat disimpan per jenisnya adalah lima. Beberapa *item* hanya dapat diperoleh dari balapan, yang hanya bertahan di dalam *shop* selama tiga *turn*. *Items* ini akan selalu ditukar setiap enam *turn* sekali. Ada kemungkinan saat pertukaran *item* sekaligus diberikan potongan harga, mulai dari 10% hingga 25% dengan pembulatan ke atas.

Keterbatasan *items* ini membuat pemain harus mengoptimisasi penggunaan *coin* tersebut agar dapat memaksimalkan penggunaan *items* dan memaksimalkan keluaran gadis kuda saat selesai dilatih.

III. PENERAPAN ALGORITMA

Masalah yang akan diselesaikan adalah maksimasi *profit* dari pembelian *item* untuk *Training*. Masalah ini merupakan salah satu jenis persoalan *knapsack*. Persoalan ini dapat diselesaikan dengan memanfaatkan algoritma *greedy*. Penjabaran dari penyelesaian masalah ini adalah sebagai berikut.

A. Pemetaan Masalah

Algoritma *greedy* akan dimanfaatkan untuk menyelesaikan masalah optimasi ini, oleh karena itu, perlu dilakukan pemetaan elemen-elemen permasalahan ke dalam elemen algoritma *greedy*. Hasil pemetaannya adalah sebagai berikut.

- Kapasitas *knapsack* analog dengan jumlah *coin* yang dimiliki pemain.
- Himpunan kandidat (C) dalam permasalahan ini adalah kumpulan *items* yang dapat dibeli pada suatu waktu dalam *Training*.
- Himpunan solusi (S) dalam permasalahan ini adalah kumpulan *items* yang akan dibeli, yang diharapkan akan memberikan *profit* optimal.
- Fungsi solusi dalam permasalahan ini adalah pengecekan seluruh *items* yang terpilih apakah memiliki total harga yang sudah mendekati jumlah *coin* yang dimiliki pemain.
- Fungsi seleksi (selection function) dalam permasalahan ini adalah strategi *greedy* yang bersifat heuristik.
- Fungsi kelayakan (feasible) dalam permasalahan ini adalah pengecekan apakah jika suatu *item* dipilih, maka

hasil penambahan total harganya masih lebih kecil dari jumlah *coin* yang dimiliki pemain.

- Fungsi objektif dalam permasalahan ini adalah memaksimalkan total *profit* yang didapat.

Items adalah objek yang akan dipilih oleh algoritma. Dalam hal ini, *weight* (berat) adalah harga dari item. Masing-masing *item* memiliki efek yang berbeda-beda sehingga akan sulit untuk mengukur *profit* (keuntungan) antara *item* yang satu dengan yang lainnya.

B. Pengukuran Profit item dan Strategi Heuristik

Untuk mengatasi masalah perbandingan ukuran *profit*, diperlukan suatu skala tingkat keuntungan universal numerik yang dapat diterapkan pada himpunan kandidat. Keseluruhan *item* perlu dianalisis agar nilai signifikansi dapat diberikan untuk tiap *item* dengan tepat.

Hasil analisis dari keseluruhan *item* adalah sebagai berikut.

1) *Bad Condition items* hanya akan memiliki *profit* bila gadis kuda sedang memiliki kondisi buruk. *Item* jenis ini memiliki tingkat keuntungan paling rendah.

2) *Good Condition items* yang penting hanyalah “Learning Hat” yang memiliki harga 280 *coin*. *Item* ini bersifat sangat langka sehingga memiliki *profit* paling besar. *Item* tersebut memiliki tingkat keuntungan paling tinggi.

3) *Bond Gauge items* yang penting hanyalah “Carrot BBQ Set” yang memiliki harga 40 *coin*. Meskipun demikian, *item* ini langsung kehilangan nilainya jika keseluruhan karakter *support* sudah memiliki tingkat persahabatan minimal 80% (sudah tidak perlu ditambah lagi karena sudah dapat mengaktifkan efek penambahan status). Saat masih memiliki nilai, *item* tersebut memiliki tingkat keuntungan tepat di bawah “Bootcamp Megaphone”.

4) *Race items* yang penting hanyalah “Horseshoe Hammer (Superior)” yang memiliki harga 40 *coin*. *Item* ini akan memberikan *profit* terbaik saat dipakai untuk tiga balapan terakhir, artinya jika sudah membeli tiga *item* ini, maka akan langsung kehilangan nilainya. Tingkat keuntungannya di bawah segala varian “Ankle Weight”.

5) *Status items* yang penting hanyalah varian *writings* (15 *coin*) dan *scroll* (30 *coin*). *Item* ini memiliki tingkat keuntungan tepat di atas *Bad Condition items*.

6) *Energy / Motivation Recovery items* memiliki beberapa *item* yang penting, yaitu:

- Segala jenis varian “Vital” yang tingkat keuntungannya diukur berdasarkan penambahan *energy* yang diperoleh.
- Kedua jenis “Cupcake” yang tingkat keuntungannya diukur berdasarkan penambahan *motivation* yang diberikan. “Plain Cupcake” (30 *coin*) akan kehilangan nilainya setelah sudah tersimpan sebanyak dua buah, dan “Sweet Cupcake” (55 *coin*)

akan langsung kehilangan nilainya setelah sudah tersimpan sebanyak satu buah saja.

- “Royal Bitter Juice” (70 *coin*) memiliki tingkat keuntungan di bawah “Cupcake”.

7) *Training items* memiliki *item* yang penting, namun harus memperhatikan beberapa aspek di luar *shop*. Penjabarannya sebagai berikut.

- “Reset Whistle” (20 *coin*) memiliki tingkat keuntungan tepat setelah “Carrot BBQ Set”.
- “Amulet for Health Prayer” (40 *coin*) memiliki tingkat keuntungan di antara “Vital 65” (75 *coin*) dan “Vital 40” (55 *coin*).
- Varian “Megaphone” yang penting adalah “Spartan Megaphone” (55 *coin*) dan “Bootcamp Megaphone” (70 *coin*) yang tingkat keuntungannya berdasarkan persentase penambahan status dengan sedikit koreksi.
- Untuk “Ankle Weight” (50 *coin*), varian yang memiliki nilai hanyalah varian yang cocok dengan tipe kartu *support* yang dipakai (Misalnya hanya menggunakan kartu *support* bertipe *Speed* dan *Power*, maka yang memiliki nilai adalah varian *Speed* dan *Power*). Secara umum, tingkat keuntungannya tepat di bawah “Sparta Megaphone”.
- Untuk “Training Petition” (150 *coin*), varian yang memiliki nilai hanyalah varian yang cocok dengan tipe kartu *support* yang dipakai (Misalnya hanya menggunakan kartu *support* bertipe *Speed* dan *Power*, maka yang memiliki nilai adalah varian *Speed* dan *Power*). Meskipun demikian, tingkat keuntungan *item* ini hanyalah tepat di atas *Bad Condition items*.

Berdasarkan analisis di atas, tingkat keuntungan *item* ditentukan pada tabel di bawah. Skala yang digunakan antara 0 sampai dengan 100.

TABLE I. TINGKAT KEUNTUNGAN ITEM

Nama	Nilai	Nama	Nilai
Learning Hat	100	Ankle Weight	50
Bootcamp Megaphone	90	Horseshoe Hammer (Sup.)	45
Carrot BBQ Set	85	Sweet Cupcake	40
Reset Whistle	80	Plain Cupcake	35
Vital 65	75	Scrolls	30
Amulet for Health Prayer	70	Writings	25
Vital 40	65	Training Petition	20
Vital 20	60	Bad Condition Item	10
Spartan Megaphone	55	Lainnya	5

Tabel di atas akan menjadi nilai acuan *profit* yang akan dipilih oleh algoritma yang menggunakan strategi *greedy by profit*. Perlu diperhatikan bahwa “Lainnya” tidak termasuk *item* yang sudah kehilangan atau tidak ada nilainya, sebagaimana yang sudah dijelaskan sebelumnya. Terakhir, ada kemungkinan dua strategi atau lebih akan menghasilkan total keuntungan yang sama. Bila nilai tersebut merupakan nilai keuntungan terbesar, maka yang paling optimal dilihat dari sisa *coin* terbanyak.

C. Studi Kasus

Di bagian ini akan diberikan beberapa contoh kasus yang akan memanfaatkan algoritma *greedy by profit*, *weight*, dan *density*. *Weight* (w) merupakan harga *item*, *profit* (p) merupakan keuntungan yang akan didapat untuk suatu *item*, dan *density* (p/w). Perlu diperhatikan bahwa algoritma hanya akan mencoba membeli *item* dengan keuntungan terbesar saat itu juga, artinya tidak akan memperhitungkan kemungkinan ke depan, seperti waktu *item* habis dan waktu rotasi *shop*.

Pada suatu waktu, pemain ingin melatih gadis kuda baru, yang bernama “Agnes Digital”. Persiapan pemain sebelum memulai *Training* tertera pada tangkapan layar di bawah. Dapat dilihat bahwa kartu *support* yang dipakai bertipe *Guts* dan *Wisdom*, sehingga varian “Ankle Weight” serta “Training Petition” yang akan dipilih hanyalah yang bertipe kedua itu saja.



Gambar 2.4: Setup pemain sebelum *Training* dimulai
(Sumber: Arsip Penulis)

Pada *turn* tertentu, setelah mengumpulkan beberapa *coin*, pemain hendak membeli *item* yang akan membantunya dalam *Training*. Tingkat persahabatan ke *support* belum memadai. Daftar *item* yang dapat dibeli pemain tertera pada tabel di bawah.

TABLE II. DAFTAR *ITEM* YANG DAPAT DIBELI PADA KASUS I DAN II

No.	Nama Item	w	p	p/w
1.	Writings of Power	15	25	1.667
2.	Notepad of Speed	10	5	0.5
3.	Carrot BBQ Set	40	85	2.125
4.	Power Ankle Weight	50	0	0
5.	Guts Ankle Weight	50	50	1
6.	Spartan Megaphone	55	55	1
7.	Tricks to Efficient Training	150	5	0.033
8.	Speed Training Petition	150	20	0.133

Saat itu, pemain memiliki 100 *coin*. Solusi algoritma *greedy*-nya adalah sebagai berikut.

TABLE III. SOLUSI ALGORITMA *GREEDY* KASUS I

Properti Objek				Greedy by		
No.	w	p	p/w	w	p	p/w
1.	15	25	1.667	1	0	1
2.	10	5	0.5	1	0	1
3.	40	85	2.125	1	1	1
4.	50	0	0	0	0	0
5.	50	50	1	0	0	0
6.	55	55	1	0	1	0
7.	150	5	0.033	0	0	0
8.	150	20	0.133	0	0	0
Total Harga				65	95	65
Sisa <i>coin</i>				35	5	35
Total Keuntungan				115	140	115

Solusi optimal dari kasus tersebut adalah

$$X = (0, 0, 1, 0, 0, 1, 0, 0)$$

Dengan total harga 95 dan total keuntungan 140. Dalam kasus ini, *greedy by profit* memberikan keuntungan terbesar. Dalam kasus ini juga terlihat jumlah *coin* terlalu sedikit sehingga tidak mampu membeli cukup banyak *item*. Hal ini terlihat pada kedua algoritma *greedy* lainnya yang menyisakan banyak sekali *coin*, namun tidak cukup untuk membeli *item* yang juga memberikan keuntungan yang cukup besar.

Misalkan pemain mendapatkan 20 *coin* tambahan dari balapan sehingga total *coin* milik pemain menjadi 120, maka solusi algoritma *greedy*-nya menjadi seperti berikut.

TABLE IV. SOLUSI ALGORITMA *GREEDY* KASUS II

Properti Objek				Greedy by		
No.	w	p	p/w	w	p	p/w
1.	15	25	1.667	1	1	1
2.	10	5	0.5	1	1	1
3.	40	85	2.125	1	1	1
4.	50	0	0	0	0	0
5.	50	50	1	1	0	0
6.	55	55	1	0	1	1
7.	150	5	0.033	0	0	0
8.	150	20	0.133	0	0	0
Total Harga				115	120	120
Sisa <i>coin</i>				5	0	0
Total Keuntungan				165	170	170

Solusi optimal dari kasus tersebut adalah

$$X = (1, 1, 1, 0, 0, 1, 0, 0)$$

Dengan total harga 120 dan total keuntungan 170. Hanya dengan ekstra 20 *coin* saja, strategi *greedy by density* dan *greedy by weight* sama-sama mampu menemukan solusi optimal.

Beberapa *turn* selanjutnya, pemain hendak membeli *item* kembali. Gadis kuda pemain tidak memiliki kondisi buruk apapun dan “Guts Ankle Weight” yang dimiliki pemain sudah berjumlah maksimal sehingga tidak dapat dibeli lagi. Kali ini kebanyakan *item* dikenai potongan harga 10% dibulatkan ke bawah. Daftar *item* yang dapat dibeli tertera pada tabel di bawah.

TABLE V. DAFTAR ITEM YANG DAPAT DIBELI PADA KASUS III DAN IV

No.	Nama Item	w	p	p/w
1.	Training Helper DVD	13	0	0
2.	Training Helper DVD	13	0	0
3.	Writings of Stamina	13	25	1.923
4.	Writings of Power	13	25	1.923
5.	Vital 20	31	60	1.935
6.	Horseshoe Hammer (Fine)	22	5	0.227
7.	Power Ankle Weight	45	0	0
8.	Guts Ankle Weight	45	0	0
9.	Spartan Megaphone	49	55	1.122
10.	Tricks to Efficient Training	135	5	0.037
11.	Speed Ankle Weight	50	0	0
12.	Bootcamp Megaphone	70	90	1.285

Saat itu, pemain memiliki 160 *coin*. Solusi algoritma *greedy*-nya adalah sebagai berikut.

TABLE VI. SOLUSI ALGORITMA GREEDY KASUS III

Properti Objek				Greedy by		
No.	w	p	p/w	w	p	p/w
1.	13	0	0	0	0	0
2.	13	0	0	0	0	0
3.	13	25	1.923	1	0	1
4.	13	25	1.923	1	0	1
5.	31	60	1.935	1	1	1
6.	22	5	0.227	1	0	1
7.	45	0	0	0	0	0
8.	45	0	0	0	0	0
9.	49	55	1.122	1	1	0
10.	135	5	0.037	0	0	0
11.	50	0	0	0	0	0
12.	70	90	1.285	0	1	1
Total Harga				128	150	149
Sisa <i>coin</i>				32	10	11
Total Keuntungan				170	205	205

Solusi optimal dari kasus tersebut adalah

$$X = (0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1)$$

Dengan total harga 149 dan total keuntungan 205. Dalam kasus ini, strategi *greedy by density* dan *greedy by weight* juga sama-sama mampu mendapatkan total keuntungan yang paling besar. Meskipun demikian, *greedy by density* memberikan sisa *coin* yang lebih besar, sehingga strategi tersebut merupakan strategi yang optimal.

Misalkan pemain menggunakan satu "Guts Ankle Weight" sehingga pemain menjadi dapat membeli *item* tersebut, kemudian mendapat tambahan 40 *coin* sehingga totalnya menjadi 200, maka solusi algoritma *greedy*-nya menjadi seperti berikut.

TABLE VII. SOLUSI ALGORITMA GREEDY KASUS IV

Properti Objek				Greedy by		
No.	w	p	p/w	w	p	p/w
1.	13	0	0	0	0	0
2.	13	0	0	0	0	0
3.	13	25	1.923	1	0	1
4.	13	25	1.923	1	0	1
5.	31	60	1.935	1	1	1
6.	22	5	0.227	1	0	1
7.	45	0	0	0	0	0
8.	45	50	1.111	1	1	0
9.	49	55	1.122	1	1	1
10.	135	5	0.037	0	0	0
11.	50	0	0	0	0	0
12.	70	90	1.285	0	1	1
Total Harga				173	195	198
Sisa <i>coin</i>				27	5	2
Total Keuntungan				220	255	260

Solusi optimal dari kasus tersebut adalah

$$X = (0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1)$$

Dengan total harga 198 dan total keuntungan 260. Saat *coin* ditambahkan sebanyak 40, *greedy by profit* akan memilih "Guts Ankle Weight" dengan *profit* 50, sementara *greedy by density* akan memilih "Spartan Megaphone" dengan *profit* 55. Hal ini membuat *greedy by density* menjadi lebih unggul. Perlu diperhatikan bahwa jika *coin* ditambahkan lagi sebanyak 20, *greedy by profit* akan mengungguli *greedy by density* dengan membeli salah satu dari "Writings".

Dari empat studi kasus yang telah dibahas, terlihat bahwa *greedy by density* adalah strategi heuristik yang paling cocok diterapkan untuk permasalahan ini, disusul oleh *greedy by profit* yang keunggulannya beda tipis dengan strategi sebelumnya. Meskipun demikian, keunggulan kedua strategi ini sangat bergantung pada jumlah *coin* yang dimiliki pemain, terlihat pada kasus IV ketika jumlah *coin* ditambahkan sedikit lagi, strategi yang lebih unggul bisa berbeda.

IV. KESIMPULAN

Algoritma *greedy* merupakan salah satu metode yang tepat dalam menyelesaikan persoalan optimasi seperti pada persoalan *integer knapsack*, dalam hal ini untuk mengoptimasi *profit* yang didapatkan dalam membeli *item* pada mode *Training* dalam skenario "Make a New Track". Dalam percobaan ini, pemilihan *item* berhasil mendapatkan solusi optimal menggunakan strategi *greedy by profit* dan *greedy by density*. Salah satu ataupun keduanya dapat memberikan total keuntungan paling tinggi tergantung dari jumlah *coin* yang dimiliki pemain. Dari keempat kasus yang dibahas, *greedy by density* unggul tipis dibandingkan *greedy by profit*.

Meskipun demikian, dalam kenyataannya, masih ada pertimbangan lain yang diperlukan dalam membeli *item* tersebut. Faktor lama bertahannya *item* dalam shop serta jumlah *race* yang dapat diikuti gadis kuda sehingga dapat membeli *item* lebih banyak tidak dikaji dalam makalah ini. Meskipun demikian, Permasalahan ini sudah dapat mendemonstrasikan kemampuan algoritma *greedy* dalam menyelesaikan masalah tersebut.

VIDEO LINK AT YOUTUBE

Untuk video, link Youtube diberikan di bawah.

<https://youtu.be/cZ0wHitnnl8>

UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan ke hadirat Allah SWT. karena rahmat dan hidayah yang telah Beliau berikan membuat penulis dapat menyelesaikan makalah ini dengan baik dan tepat waktu. Penulis juga mengucapkan terima kasih kepada Bapak Ir. Rinaldi Munir selaku Dosen Strategi Algoritma K3 yang telah memberikan sumber materi serta bimbingan selama satu semester penuh, sehingga penulis dapat mengaplikasikan ilmu yang sudah didapatkan, terutama algoritma *greedy* dengan baik dan benar di dalam makalah ini.

REFERENCES

- [1] Munir, R. 2021. Algoritma Greedy (Bagian 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 9 Mei 2022.
- [2] Munir, R. 2021. Algoritma Greedy (Bagian 2). [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag2.pdf), diakses pada 9 Mei 2022.
- [3] Munir, R. 2021. Algoritma Greedy (Bagian 3). [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Greedy-\(2022\)-Bag3.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Greedy-(2022)-Bag3.pdf), diakses pada 9 Mei 2022.

[4] QooApp English – Uma Musume: Pretty Derby GamePage. <https://apps.qoo-app.com/en/app/6172>, diakses pada 9 Mei 2022.

[5] 【ウマ娘】序盤の進め方 | ゲームの流れとやることを解説 [Uma Musume - How to Proceed in Early Stages | Explaining The Flow of The Game and What to Do]. <https://gamewith.jp/uma-musume/article/show/257616>, diakses pada 9 Mei 2022.

[6] 【ウマ娘】クライマックスのおすすめアイテムと効果一覧 [Uma Musume – Recommended items and effects list of climax] <https://gamewith.jp/uma-musume/article/show/320296>, diakses pada 14 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, 20 Mei 2022



Dzaky Fattan Rizqullah 13520003