

Penerapan Alogoritma Greedy dalam Memilih Bibit di Permainan Rune Factory 3

Trian Annas Thoriq Sumarjadi - 13516148

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

13516148@std.stei.itb.ac.id

Abstract—*Rune Factory 3* adalah sebuah permainan video game ber-genre *role-playing simulation* yang merupakan seri ke tiga dari permainan *Rune Factory*. Di dalam permainan ini, kita harus mengalahkan monster-monster yang jahat, untuk mengalahkan monster yang jahat tersebut, kita membutuhkan peralatan senjata yang bagus, sedangkan untuk mendapatkan senjata itu salah satu caranya adalah dengan membelinya, dan untuk membelinya kita membutuhkan uang. Untuk mendapatkan uang, kita harus bekerja bercocok tanam. Untuk mendapatkan hasil penjualan tanaman yang memiliki penghasilan terbesar, kita menggunakan algoritma *greedy* untuk menyeleksi bibit yang menghasilkan penjualan terbesar.

Keywords—*Greedy, bibit, Rune Factory 3*

I. PENDAHULUAN

Rune Factory 3 adalah game besutan *Neverland Co., Ltd.* Permainan I ini merupakan serial ketiga dari Permainan *Rune Factory* yang dirilis pada tanggal 22 Oktober 2009 di Jepang dan tanggal 9 November 2010 di Amerika Utara. *Console* untuk memainkan Permainan ini menggunakan NDS atau Nintendo DS, namun kita juga bisa memainkannya di Komputer dengan menggunakan *emulator* yang bernama deSmuMe.

Rune Factory 3 ini memiliki genre *role-game simulator* dengan tambahan *Action* di dalamnya. *Rune Factory* itu sendiri merupakan Permainan seperti *Harvest Moon* tetapi disini kita bisa bertarung melawan monster. Dengan fitur tambahan seperti itu, Pemain tidak hanya berfokus pada bercocok tanam dan bersosialisasi dengan warga, tetapi pemain juga memiliki misi untuk memainkan permainan ini, misi tersebut di bagi menjadi dua, yaitu misi utama dan misi sampingan.

Misi utama merupakan cerita utama yang membumbui permainan ini, dengan cerita yang bergenre drama dan romance, pemain tidak akan bosan mengikuti cerita utama. Sedangkan misi sampingan adalah misi yang tidak berkaitan dengan cerita utama, tetapi tidak kalah menarik ceritanya dengan cerita utama.

Penghasilan untuk mendapatkan Uang didalam permainan ini adalah dengan menjual barang, namun jalan yang paling efektif dalam mencari uang dengan cepat adalah dengan bercocok tanam. Disini penulis akan memberikan cara dalam memilih bibit untuk ditanam, dimana nantinya bibit tersebut jika sudah siap panen, hasil panennya memiliki nilai jual yang maksimum per bulannya, dengan 1 bulannya terdapat 30 hari.

II. DASAR TEORI

A. Algoritma Greedy

Greedy yang dalam bahasa indonesia adalah rakus, serakah atau tamak. Algoritma Greedy merupakan algoritma atau metode yang paling populer untuk memecahkan persoalan optimasi. Persoalan optimasi (*optimization problems*) yang berarti persoalan mencari solusi optimum dapat dibagi menjadi 2, yaitu maksimasi (*maximization*) dan minimasi (*minimization*). Solusi optimum (terbaik) adalah solusi yang bernilai minimum atau maksimum dari sekumpulan alternatif solusi yang mungkin. Solusi yang memenuhi semua kendala disebut solusi layak (*feasible solution*). Solusi layak yang mengoptimalkan fungsi optimasi disebut solusi optimum. Prinsip greedy adalah: "take what you can get now!"[1].

Algoritma greedy membentuk solusi langkah per langkah (step by step). Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya. Pendekatan yang digunakan di dalam algoritma greedy adalah membuat pilihan yang "tampaknya" memberikan perolehan terbaik, yaitu dengan membuat pilihan optimum lokal (*local optimum*) pada setiap langkah dengan harapan bahwa sisanya mengarah ke solusi optimum global (*global optimum*). Pada setiap langkah diperoleh optimum lokal. Bila algoritma berakhir, kita berharap optimum lokal menjadi optimum global.

Algoritma greedy disusun oleh elemen-elemen berikut:

1. Himpunan kandidat, Berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi, Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.
3. Fungsi seleksi (*selection function*) Memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.

4. Fungsi kelayakan (*feasible*) Memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (*constraints*) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.

5. Fungsi obyektif, yaitu fungsi yang memaksimumkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain).

Pseudo-code algoritma *Greedy*-nya adalah sebagai berikut:

```
procedure greedy(input C: himpunan_kandidat;
                 output S: himpunan_solusi)
{ menentukan solusi optimum dari persoalan
  optimasi dengan algoritma greedy
  Masukan: himpunan kandidat C
  Keluaran: himpunan solusi S
}
Deklarasi
  x : kandidat;
Algoritma:
  S ← {} { inisialisasi S dengan kosong }
  while (belum SOLUSI(S)) and (C ≠ {}) do
    x ← SELEKSI(C); { pilih sebuah
    kandidat dari C }
    C ← C - {x} { elemen himpunan
    kandidat berkurang satu }
    if LAYAK(S ∪ {x}) then
      S ← S ∪ {x}
    endif
  endwhile
  {SOLUSI(S) sudah diperoleh or C = {} }
```

Dilihat dari *pseudo-code* di atas dapat kita ketahui langkah-langkah algoritma *greedy* adalah pertama kita harus memiliki himpunan kandidat terlebih dahulu, lalu langkah kedua adalah mengambil salah satu kandidat dari himpunan kandidat, sehingga berkurang 1 kandidat dari himpunan kandidat tersebut, lalu langkah ketiga adalah periksa apakah kandidat yang diambil tersebut layak, jika iya masukan ke dalam himpunan solusi, lalu langkah ke empat periksa apakah himpunan solusi sudah memberikan solusi yang lengkap atau belum, jika belum maka lakukan kembali dari langkah 2, jika sudah, maka berhenti, tetapi jika himpunan kandidatnya sudah habis, bisa jadi himpunan solusinya nol.

B. Persoalan Knapsack

Knapsack dapat diartikan sebagai karung atau kantung. Karung digunakan untuk memuat sesuatu dan tentunya tidak semua objek dapat ditampung di dalam karung. Karung tersebut hanya dapat menyimpan beberapa objek dengan total ukurannya (*weight*) lebih kecil atau sama dengan ukuran kapasitas karung. Setiap objek itupun tidak harus kita masukkan seluruhnya. Tetapi bisa juga sebagian saja.

knapsack 0/1, yaitu suatu objek diambil seluruh bagiannya atau tidak sama sekali. Setiap objek mempunyai nilai keuntungan atau yang disebut dengan profit. Tujuan ingin mendapatkan profit maksimal. Belum tentu menggunakan banyak objek yang masuk akan menguntungkan. Bisa saja hal yang sebaliknya yang terjadi.

Persoalan Knapsack bisa diselesaikan dengan algoritma *Greedy*, dan memiliki 3 cara yaitu:

1. *Greedy by profit*

Greedy berdasarkan profit adalah proses mencari object yang memiliki profit paling besar yang dapat dimasukan ke dalam knapsack. Pada setiap langkah Knapsack diisi dengan objek yang mempunyai profit terbesar. Strategi ini mencoba memaksimumkan keuntungan dengan memilih objek yang paling menguntungkan terlebih dahulu. Pertama kali dilakukan adalah menurunkan secara menurun obyek-obyek berdasarkan profitnya. Kemudian obyek-obyek yang dapat ditampung oleh knapsack diambil satu persatu sampai knapsack penuh atau (sudah tidak ada obyek lagi yang bisa dimasukan).

2. *Greedy by Weight*

Greedy berdasarkan *weight* adalah proses mencari object yang memiliki berat paling kecil, sehingga object yang dimasukan ke dalam knapsack lebih banyak. Pada setiap langkah, knapsack diisi dengan objek yang mempunyai berat paling ringan. Strategi ini mencoba memaksimumkan keuntungan dengan memasukan sebanyak mungkin objek kedalam knapsack. Pertama kali yang dilakukan adalah mengurutkan secara menaik objek-objek berdasarkan *weight*-nya. Kemudian obyek-obyek yang dapat ditampung oleh knapsack diambil satu persatu sampai knapsack penuh atau (sudah tidak ada obyek lagi yang bisa dimasukan).

3. *Greedy by Density*

Greedy berdasarkan *density*(profit/*weight*) adalah proses mencari object dengan *density* paling besar. Pada setiap langkah, knapsack diisi dengan obyek yang mempunyai *densitas* terbesar (perbandingan nilai dan berat terbesar). Strategi ini mencoba memaksimumkan keuntungan dengan memilih objek yang mempunyai keuntungan per unit berat terbesar. Pertama kali yang dilakukan adalah mencari nilai profit per unit/ *density* dari tiap-tiap objek. Kemudian obyek-obyek diurutkan berdasarkan *densitas*nya. Kemudian obyek-obyek yang dapat ditampung oleh knapsack diambil satu persatu sampai knapsack penuh atau (sudah tidak ada obyek lagi yang bisa dimasukan).

Dari ketiga strategi *greedy* di atas tidak menjamin hasil yang didapat selalu menghasilkan nilai yang paling optimum, namun

nilai yang dihasilkan tetap mendekati nilai optimum. Sebagai contoh mari kita lihat persoalan knapsack di bawah ini:

Diketahui :

$w_1 = 7$	$p_1 = 14$
$w_2 = 12$	$p_2 = 40$
$w_3 = 5$	$p_3 = 10$
$w_4 = 10$	$p_4 = 20$
$w_5 = 5$	$p_5 = 15$
Kapasitas Knapsack = 20	

Properti objek			
i	w_i	p_i	p_i/w_i
1	7	14	2
2	12	40	3.34
3	5	10	2
4	10	20	2
5	5	15	3

i	Greedy by			Solusi optimal
	profit	weight	density	
1	0	1	0	0
2	1	0	1	1
3	0	1	0	0
4	0	0	0	0
5	1	1	1	1
Σ bobot	17	17	17	17
Σ keuntungan	55	39	55	55

Pada contoh diatas, kita lihat dengan menggunakan 3 strategi *greedy* 2 diantaranya menghasilkan nilai optimum, yaitu *greedy by profit* dan *greedy by density*, namun pada *greedy by weight*, ketebulan tidak menghasilkan nilai optimum. Pada contoh kasus yang lain bisa jadi tiga-tiganya strategi *greedy* tersebut tidak menghasilkan solusi optimum sama sekali.

C. Rune Factory 3

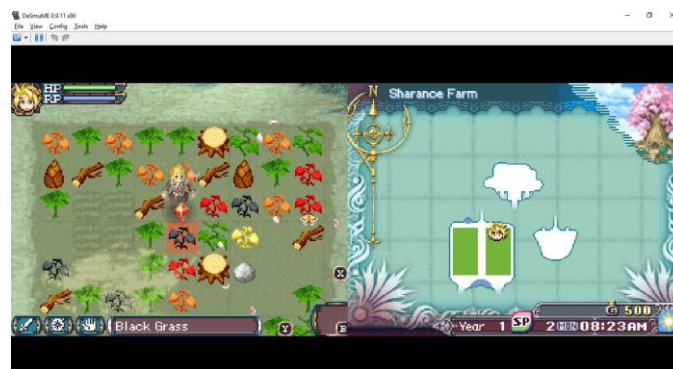
Pertama kali kita memainkan permainan *Rune Factory* kita diperlihatkan sebuah *scene* dimana karakter kita yang bernama *default*-nya Micah terjatuh dari langit yang pada saat itu sedang terjadi hujan badai, Micah tergeletak didepan sebuah rumah, lalu ditolong oleh seorang perempuan yang bernama Shara, Micah ini pingsan dengan wujud sebagai domba(Micah ini setengah manusia dan setengah domba), dibawahlah masuk kedalam rumahnya untuk dirawat. Keesokan harinya Micah terbangun dengan wujud manusianya, lalu pergi meninggalkan rumah tersebut, setelah di luar rumah, micah bertemu perempuan yang menolong Micah. Micah ini mengalami hilang ingatan yang tidak diketahui sebabnya, karena kasihan, Shara lalu memberinya tempat tinggal di sebuah Rumah Pohon yang bernama Sarance, di bawah pohon tersebut terdapat kebun yang tidak terawat. Shara memintanya untuk tinggal di Rumah pohon tersebut dan merawat kebun yang terbegkalai tersebut, dan bekerja menjadi petani agar bisa mendapatkan penghasilan. Tidak hanya disitu saja Micah juga harus mencari petunjuk untuk mendapatkan kembali ingatannya dengan melawan monster-monster jahat.



Gambar 1. Cover Permainan *Rune Factory 3*
(http://therunefactory.wikia.com/wiki/Rune_Factory_3:_A_Fantasy_Harvest_Moon)

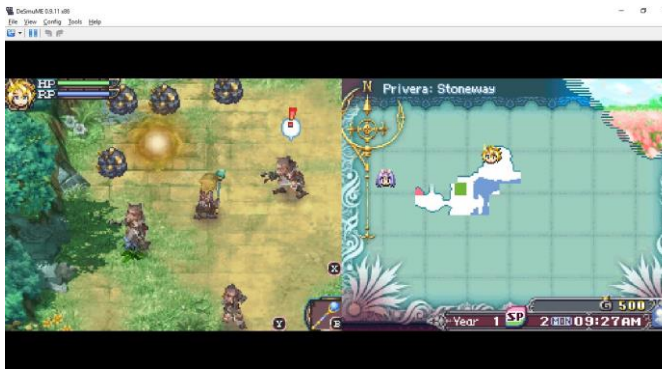
Pemain barulah bisa bereksplorasi fitur yang ada di dalam game seperti berkebun, melawan monster dan berinteraksi dengan NPC(Non Player Character).

Untuk fitur berkebun, pemain memiliki kebun yang luas untuk menanam tanaman yang bermacam-macam sesuai dengan musim yang ada, dalam game ini terdapat 4 musim. Tidak hanya sampai disitu pemain juga bisa membeli kandang untuk memelihara monster.



Gambar 2. Kebun Pemain
(Penulis)

Untuk fitur melawan monster, pemain bisa berpetualang ke luar Desa untuk melawan monster. Untuk menaklukan monster-monster jahat, pemain harus memiliki perlengkapan bertempur yang mumpuni agar tidak mati dalam melawan monster-monster jahat. Untuk mendapatkan perlengkapan yang bagus, salah satunya dengan membelinya di toko senjata dan harganya yang cukup relatif mahal. Untuk mendapatkan uang salah satu caranya pemain harus menjual hasil dari bercocok tanam.



Gambar 3. Gameplay Permainan ketika sedang melawan monster
(penulis)

Untuk fitur bersosialisasi dengan NPC, pemain bisa berbincang-bincang dengan warga desa. Di dalam permainan ini terdapat 11 wanita yang memiliki sifat kepribadian yang berbeda-beda yang bisa di kencana oleh pemain, pemain juga akan mendapatkan *even-even* yang penuh dengan *drama* dan *romance* yang membumbui ketika sedang berkencan dengan wanita-wanita tersebut, dan nantinya pemain bisa memilih 1 karakter wanita untuk dinikahinya, tidak sampai disitu pemain juga bisa memiliki anak.



Gambar 4. Daftar wanita yang bisa dinikahi di Rune Factory 3.

(<https://www.tehvidya.com/tag/rune-factory-3/>)

Pada saat memulai permainan, player akan mendapatkan uang sebagai modal, sebesar 500 G, dan dengan tanggal 2 bulan Spring tahun ke 1. Di dalam permainan ini terdapat 4 bulan dalam 1 tahun, dimana setiap bulannya mewakili 4 musim, dan setiap bulannya terdapat 30 hari, dan 1 minggunya hanya terdapat 6 hari.

III. PENERAPAN ALGORITMA GREEDY UNTUK MEMILIH BIBIT

Permainan *Rune Factory 3* ini memiliki 4 musim, yang setiap musimnya memiliki bibit tanaman yang berbeda-beda dan juga memiliki profit yang berbeda juga. Untuk mendapatkan profit maksimal setiap bulannya, dengan asumsi

uang modal perbulannya pemain memiliki uang sebesar 2000 G, dan menanam dimulai di tanggal 1. Pembelian juga diasumsikan hanya bisa membeli 1 bibit untuk setiap jenisnya.

Setiap bibit memiliki informasi harga beli (Hb), harga jual (Hj) dan juga informasi waktu untuk tumbuh hingga siap panen(t). Karena dalam 1 bulan terdapat 30 hari, maka terdapat kemungkinan 1 jenis bibit bisa ditanam lebih dari 1 kali sepanjang kurang dari sama dengan 30 hari. Maka rumus untuk menentukan profit per bulan setiap bibit adalah:

$$P = (Hj * 30) \text{ div } t \text{ (ambil bulat ke bawah)}$$

Dan rumus untuk rumus menentukan biaya modal per bulan setiap bibitnya adalah :

$$W = (Hb * 30) \text{ div } t \text{ (ambil bulat ke bawah)}$$

Dibulatkan kebawah karena, jika kita menanam di waktu akhir-akhir pergantian bulan, dan belum sempat memanen tanamannya, maka tanaman akan busuk dan tidak bisa di jual.

A. Spring

Plant	Buy	Sell	Days to Grow	Image
Pink Turnip Seed	80G	200G	3 Days	
Cabbage Seed	100G	480G	6 Days	
Moondrop Seed	70G	600G	7 Days	
Toyherb Seed	60G	180G	2 Days	
Cherry Grass Seed	120G	860G	6 Days	

Gambar 5. Daftar bibit pada bulan spring
([http://therunefactory.wikia.com/wiki/Crops_\(RF3\)](http://therunefactory.wikia.com/wiki/Crops_(RF3)))

Himpunan kandidatnya adalah kumpulan bibit yang bisa dibeli yaitu (*Pink Turnip*, *Cabbage*, *Moondrop*, *Toyherb*, *Cherry*).

Himpunan Solusinya adalah kumpulan bibit yang memungkinkan memiliki profit maksimum.

Fungsi seleksinya adalah memilih bibit yang menghasilkan keuntungan terbesar di akhir bulan.

Fungsi layaknya adalah membatasi pengeluaran tidak melebihi modal uang yang dimiliki.

Fungsi objectifnya adalah jumlah uang yang didapatkan maksimum.

Properti objek			
i	w_i	p_i	p_i/w_i
1	800	2000	2.5
2	500	2400	4.8
3	280	2400	8.57
4	900	2700	3
5	600	4300	7.167

I	Greedy by			Solusi optimal
	profit	weight	density	
1	0	0	0	0
2	0	1	1	0
3	1	1	1	1
4	1	0	0	1
5	1	1	1	1
Σ bobot	1780	1380	1380	1780
Σ keuntungan	9400	9100	9100	9400

Solusi optimum yang didapat di bulan *Spring* adalah $S = (0,0,1,1,1)$. Profit yang didapat di akhir bulannya adalah 9400 G.

B. Summer

Plant	Buy	Sell	Days to Grow	Image
Cucumber Seed	80G	180G	6 Days	
Tomato Seed	140G	300G	11 Days	
Pink Cat Seeds	50G	270G	4 Days	
Fire Flower Seed	260G	1,200G	6 Days	

Gambar 6. Daftar bibit pada bulan summer
([http://therunefactory.wikia.com/wiki/Crops_\(RF3\)](http://therunefactory.wikia.com/wiki/Crops_(RF3)))

Himpunan kandidatnya adalah kumpulan bibit yang bisa dibeli yaitu (*Cucumber, Tomato, Pink cat, Fire Flower*).

Himpunan Solusinya adalah kumpulan bibit yang memungkinkan memiliki profit maksimum.

Fungsi seleksinya adalah memilih bibit yang menghasilkan keuntungan terbesar di akhir bulan.

Fungsi layaknya adalah membatasi pengeluaran tidak melebihi modal uang yang dimiliki.

Fungsi objectifnya adalah jumlah uang yang didapatnya maksimum.

Properti objek			
i	w_i	p_i	p_i/w_i
1	400	900	2.25

2	280	600	2.14
3	350	1890	5.4
4	1300	6000	4.61

i	Greedy by			Solusi optimal
	profit	Weight	density	
1	0	1	0	0
2	1	1	1	1
3	1	1	1	1
4	1	0	1	1
Σ bobot	1930	1030	1930	1930
Σ keuntungan	8490	3390	8490	8490

Solusi optimum yang didapat di bulan *Summer* adalah $S = (0,1,1,1)$. Profit yang didapat di akhir bulannya adalah 8490 G.

C. Fall/Autumn

Plant	Lv 1-10 Buy	Lv 1 Sell	Days to Grow	Image
Potato Seed	90G	280G	3 Days	
Carrot Seed	100G	590G	8 Days	
Yam Seed	80G	140G	6 Days	
Spinach Seed	60G	140G	2 Days	
Charm Blue Seed	40G	320G	6 Days	

Gambar 7. Daftar bibit pada bulan autumn
([http://therunefactory.wikia.com/wiki/Crops_\(RF3\)](http://therunefactory.wikia.com/wiki/Crops_(RF3)))

Himpunan kandidatnya adalah kumpulan bibit yang bisa dibeli yaitu (*Potato, Carrot, Yam, Spinach, Charm Blue*).

Himpunan Solusinya adalah kumpulan bibit yang memungkinkan memiliki profit maksimum.

Fungsi seleksinya adalah memilih bibit yang menghasilkan keuntungan terbesar di akhir bulan.

Fungsi layaknya adalah membatasi pengeluaran tidak melebihi modal uang yang dimiliki.

Fungsi objectifnya adalah jumlah uang yang didapatnya maksimum.

Properti objek			
i	w_i	p_i	p_i/w_i
1	900	2800	3.11
2	300	1770	5.9
3	400	700	1.25
4	900	2100	2.33
5	200	1600	8

I	Greedy by			Solusi optimal
	profit	weight	density	
1	1	1	1	1
2	0	1	1	1
3	0	1	1	1
4	1	0	0	0
5	0	1	1	1
Σ bobot	1800	1800	1800	1800
Σ keuntungan	4900	6870	6870	6870

Solusi optimum yang didapat di bulan *Autumn* adalah $S = (1,1,1,0,1)$. Profit yang didapat di akhir bulannya adalah 6870 G.

D. Winter

Plant	Buy	Sell	Days to Grow	Image
Turnip Seed	130G	420G	3 Days	
Leek Seed	120G	500G	6 Days	
Daikon Seed	160G	770G	7 Days	
W. Cabbage Seed	200G	1,350G	12 Days	

Gambar 8. Daftar bibit pada bulan winter

Himpunan kandidatnya adalah kumpulan bibit yang bisa dibeli yaitu (*Turnip, Leek, Daikon, W. Cabbage*).

Himpunan Solusinya adalah kumpulan bibit yang memungkinkan memiliki profit maksimum.

Fungsi seleksinya adalah memilih bibit yang menghasilkan keuntungan terbesar di akhir bulan.

Fungsi layaknya adalah membatasi pengeluaran tidak melebihi modal uang yang dimiliki.

Fungsi objectifnya adalah jumlah uang yang didapatnya maksimum.

Properti objek			
i	w_i	p_i	p_i/w_i
1	1300	4200	3.23
2	600	2500	4.167
3	640	3080	4.8
4	400	2700	6.75

I	Greedy by			Solusi optimal
	profit	weight	density	
1	1	0	0	0
2	0	1	1	1
3	1	1	1	1
4	0	1	1	1
Σ bobot	1940	1640	1640	1640
Σ keuntungan	7280	8280	8280	8280

Solusi optimum yang didapat di bulan *Autumn* adalah $S = (0,1,1,1)$. Profit yang didapat di akhir bulannya adalah 8280 G.

IV. KESIMPULAN

Banyak sekali strategi algoritma yang bisa dimanfaatkan dalam menyelesaikan suatu masalah di kehidupan sehari-hari, tidak harus dengan masalah yang sangat rumit dan serius, tetapi juga dalam hal-hal kecil namun menyenangkan, seperti perosalan dalam memilih bibit pada game *Rune Factory 3*.

V. UCAPAN TERIMA KASIH

Penulis ingin mengucapkan syukur kepada Tuhan Yang Maha Esa atas anugerah-Nya sehingga pembuatan makalah ini berjalan dengan lancar dan dapat terselesaikan. Ucapan terima kasih juga kepada Ibu Masayu, selaku dosen yang mengajar saya pada mata kuliah ini. Ucapan terima kasih juga kepada kedua Orang tua saya yang selalu mendoakan saya, dan terimakasih kepada teman-teman saya yang selalu mendukung saya.

REFERENSI

- [1] Munir, Rinaldi. 2004. Diktat Kuliah IF2251 Strategi Algoritmik. Algoritma Greedy. Bandung: Institut Teknologi Bandung.
- [2] <https://dosen.perbanas.id/menyelesaikan-knapsack-problem-dengan-menggunakan-algoritma-greedy/> diakses pada 11 Mei 2018
- [3] http://therunefactory.wikia.com/wiki/Rune_Factory_3:_A_Fantasy_Harvest_Moon diakses pada 11 Mei 2018

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, 29 April 2012



Trian Annas Thoriq Sumarjadi dan 13516148