

Penerapan Algoritma *Greedy* untuk Menentukan Pembelian Bibit Tanaman pada Stardew Valley

Delisha Azza Naadira / 13519133
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 13519133@std.stei.itb.ac.id

Abstrak—Bermain *game* tidak hanya bertujuan untuk bersenang-senang atau menghabiskan waktu, tetapi dapat juga melatih berbagai kemampuan yang dapat digunakan pada kehidupan sehari-hari, salah satu contohnya adalah kemampuan menggunakan uang dengan sebaik-baiknya. Stardew Valley adalah salah satu dari banyak contoh *game* yang meminta pemainnya untuk bijak dalam mengelola uang yang dimilikinya.

Keywords—*Greedy; Stardew Valley; menentukan; membeli*

I. PENDAHULUAN

Saat ini, *video game* sudah bermacam-macam jenisnya. Mulai dari aksi, petualangan, *puzzle*, strategi, simulasi, dan lain-lain. Salah satu *game* bergenre simulasi adalah Stardew Valley. Lebih tepatnya, "*farming simulator*" atau simulasi bertani. Diinspirasi oleh *game* klasik Harvest Moon, Stardew Valley merupakan suatu *game* yang menyediakan berbagai macam hal kepada pemainnya, seperti bertani, beternak, memancing, dan lain-lain.



Gambar 1. Logo "*Stardew Valley*"
(Sumber: <https://id.wikipedia.org/>)

Pada *game* ini, kita bermain sebagai suatu karakter yang diberikan warisan dari kakeknya berupa sebuah lahan pertanian di suatu kota bernama Pelican Town. Di sana pemain diminta untuk membersihkan lahan yang sudah lama tidak digunakan, menanam berbagai macam tanaman, berteman dengan warga kota, memulihkan dan memperbaiki berbagai macam bangunan kota, dan sebagainya. Hal-hal tersebut memerlukan waktu yang banyak dan sumber daya yang banyak pula. Karena itu, pemain harus menghasilkan banyak uang untuk dapat memenuhi keperluan-keperluan tersebut.

Uang tidak mudah didapatkan, apalagi pada awal permainan. Pemain harus bijaksana dalam membeli bibit tanaman untuk ditanam dan nantinya dijual. Terdapat cara-cara

lain untuk menghasilkan uang pada *game* ini, namun pada makalah ini difokuskan untuk pembelian bibit tanaman saja. Untuk mencari tahu tanaman (*crops*) mana saja yang harus dibeli, akan digunakan algoritma *greedy*.

II. LANDASAN TEORI

A. Stardew Valley

Stardew Valley merupakan suatu *game farming simulator*. Pada *game* ini terdapat empat musim—musim semi, musim panas, musim gugur, dan musim dingin—dengan masing-masing musim terdapat 28 hari. Pada tiap musim terdapat berbagai jenis tanaman yang bisa ditanam, dan hanya bisa ditanam di musim tersebut. Tanaman yang ditanam tidak sesuai dengan musimnya akan mati. Jenis tanaman yang bisa ditanam bervariasi, mulai dari sayuran, buah-buahan, dan bunga.



Gambar 2. Lahan bertani pada Stardew Valley dan beberapa tanaman yang ditanamkan
(Sumber: <https://screenrant.com/stardew-valley-best-crops-every-season/>)

Tanaman-tanaman pada Stardew Valley memiliki tiga aspek, yaitu:

1. Harga beli (*cost*), yaitu uang yang dibutuhkan pemain untuk membeli satu bibit tanaman.
2. Harga jual (*sell price*), yaitu uang yang akan didapatkan pemain ketika menjual satu hasil panen.
3. Waktu tumbuh (*growth time*), yaitu waktu yang diperlukan tanaman untuk tumbuh hingga dapat dipanen. Beberapa jenis tanaman dapat kembali memproduksi hasil panen setelah tanaman dewasa,

dengan waktu untuk tumbuh kembali (*regrowth time*) yang bervariasi.

Tidak semua tanaman bisa diakses pada awal permainan. Beberapa jenis bibit tanaman hanya bisa dibeli setelah tahun kedua, dan beberapa jenis lainnya hanya dapat dibeli pada hari-hari tertentu. Contohnya, bibit stroberi hanya dapat dibeli pada Egg Festival di musim semi. Terdapat beberapa tempat yang menjual bibit tanaman, seperti Pierre's General Store di Pelican Town dan Oasis di Calico Desert.

Pemain dapat menghasilkan uang dengan menanam tanaman dan menjual hasil panennya. Setiap tanaman memiliki kegunaan masing-masing untuk menghasilkan keuntungan paling besar—langung dijual mentah, diolah menjadi produk *artisanal* atau produk masakan, dan lain-lain. Ketika sudah panen, tanaman juga memiliki empat macam kualitas (*normal, silver, gold, iridium*), dengan semakin tinggi kualitas nilai jual akan semakin besar. Namun, untuk mempermudah dalam algoritma ini, semua tanaman yang panen diasumsikan memiliki kualitas normal. Selain itu, diasumsikan juga bahwa pertumbuhan tanaman tidak dipengaruhi oleh hal-hal eksternal yang membuatnya tumbuh lebih cepat.

B. Algoritma Greedy

Algoritma *greedy* merupakan algoritma yang memecahkan persoalan secara langkah per langkah sedemikian sehingga pada setiap langkah diambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan dengan harapan dengan memilih optimum lokal pada setiap langkah akan berakhir pada optimum global. Pada beberapa kasus, algoritma *greedy* tidak memberikan solusi yang optimal, namun akan memberikan solusi yang mendekati nilai optimum (pseudo-optimum) dengan waktu yang sangat cepat, sehingga algoritma ini paling sering digunakan untuk permasalahan optimasi.

Terdapat dua macam persoalan optimasi:

1. Maksimasi (*maximization*)
2. Minimasi (*minimization*)

Elemen-elemen algoritma *greedy* adalah sebagai berikut.

1. Himpunan kandidat (C), berisi kandidat yang akan dipilih pada setiap langkah.
2. Himpunan solusi (S), berisi kandidat yang sudah dipilih.
3. Fungsi solusi, menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi.
4. Fungsi seleksi (*selection function*), memilih kandidat berdasarkan strategi *greedy* tertentu yang bersifat heuristik.
5. Fungsi kelayakan (*feasible*), memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi.
6. Fungsi objektif, memaksimalkan atau meminimumkan.

Terdapat berbagai macam persoalan yang dapat diselesaikan dengan algoritma *greedy*, contohnya persoalan penukaran uang, persoalan pemilihan aktivitas, persoalan *Knapsack*, lintasan terpendek, dan lain-lain. Untuk permasalahan ini, digunakan penyelesaian mirip dengan persoalan *Knapsack*.

C. Persoalan Knapsack

Persoalan *Knapsack* merupakan suatu persoalan optimasi yaitu memilih objek dari sekumpulan objek yang masing-masing memiliki bobot (*weight*) dan nilai (*profit*) untuk dimuat ke dalam suatu media penyimpanan tanpa melebihi kapasitas media penyimpanan tersebut. Sering kali dicari solusi keuntungan maksimum. Terdapat dua jenis persoalan *knapsack*, yaitu *integer knapsack problem* (disebut juga *1/0 knapsack problem*) dan *fractional knapsack problem*.

Formula matematis persoalan *Knapsack* adalah:

$$\text{Maksimasi } F = \sum_{i=1}^n p_i x_i$$

dengan kendala (*constraint*)

$$\sum_{i=1}^n w_i x_i \leq K$$

Gambar 3. Formula persoalan *Knapsack*
(Sumber: dokumentasi penulis)

Dengan algoritma *greedy*, permasalahan *Knapsack* dapat diselesaikan dengan memasukkan objek satu per satu ke dalam *knapsack*. Sekali objek dimasukkan ke dalam *knapsack*, objek tersebut tidak bisa dikeluarkan lagi. Beberapa strategi *greedy* yang heuristik yang dapat digunakan untuk memilih objek yang akan dimasukkan ke dalam *knapsack* adalah sebagai berikut.

1. *Greedy by profit*, yaitu pada setiap langkah dipilih objek yang mempunyai keuntungan terbesar. Strategi ini mencoba memaksimalkan keuntungan dengan memilih objek yang paling menguntungkan terlebih dahulu.
2. *Greedy by weight*, yaitu pada setiap langkah dipilih objek yang mempunyai berat teringan. Strategi ini mencoba memaksimalkan keuntungan dengan memasukkan sebanyak-banyaknya objek ke dalam *knapsack*.
3. *Greedy by density*, yaitu pada setiap langkah dipilih objek yang mempunyai keuntungan per unit berat terbesar. Strategi ini mencoba memaksimalkan keuntungan dengan memilih objek yang memiliki keuntungan per berat terbesar.

Pemilihan objek berdasarkan salah satu dari ketiga strategi *greedy* tersebut tidak menjamin akan memberikan solusi optimal.

Fractional Knapsack Problem merupakan variasi dari persoalan *Knapsack* tetapi dengan solusi berbentuk pecahan atau fraksi. Objek yang dimasukkan ke dalam *knapsack* dapat berupa sebagian saja ($0 \leq x_i \leq 1$). Algoritma *greedy* akan selalu

berhasil memberikan solusi optimal untuk persoalan *fractional knapsack* jika strategi memilih objek adalah berdasarkan densitas terlebih dahulu.

III. PENERAPAN ALGORITMA

Pada konteks penyelesaian masalah pemilihan bibit untuk dibeli, digunakan metode penyelesaian masalah *fractional knapsack*. Karena tiap tanaman tidak memiliki *weight* namun memiliki *cost*, *sell price*, dan *growth time*, maka dilakukan modifikasi pada strategi *greedy*. Karena itu, strategi *greedy* yang digunakan untuk permasalahan ini adalah sebagai berikut.

1. *Greedy by cost*, yaitu pada setiap langkah dipilih tanaman yang memiliki harga beli bibit terkecil. Strategi ini akan mencoba memaksimalkan keuntungan dengan membeli sebanyak-banyaknya bibit tanaman.
2. *Greedy by sell price*, yaitu pada setiap langkah dipilih tanaman yang memiliki harga jual terbesar.
3. *Greedy by profit*, yaitu pada setiap langkah dipilih tanaman yang memiliki keuntungan (*sell price - cost*) terbesar.
4. *Greedy by profit per day*, yaitu pada setiap langkah dipilih tanaman yang memiliki keuntungan per hari paling besar. Strategi ini diikutkan karena setiap tanaman memiliki waktu tumbuh yang berbeda, dan beberapa jenis tanaman dapat memproduksi hasil panen lebih dari satu kali dalam satu kali tanam.

Jika pada *knapsack problem* diberikan sebuah *knapsack* dengan suatu kapasitas tertentu, pada permasalahan ini sebagai gantinya diberikan kuantitas uang yang dapat digunakan untuk membeli bibit (*budget*). Selain itu, pemain dapat melakukan produksi massal dengan membeli satu jenis bibit tanaman yang memiliki keuntungan terbesar. Namun, dapat juga ditetapkan banyak bibit maksimum untuk satu jenis tanaman yang dapat dibeli oleh pemain. Perlu diperhitungkan juga tanggal pembelian bibit tanaman, karena *profit per day* dari tanaman yang dapat dipanen lebih dari sekali dan mulai ditanam pada awal musim akan berbeda dengan yang mulai ditanam pada pertengahan musim.

Dengan itu, variabel-variabel yang digunakan pada penggunaan algoritma *greedy* ini adalah sebagai berikut.

1. $B = \text{budget}$
2. $m = \text{max. buy}$
3. $c = \text{cost}$
4. $s = \text{sell price}$
5. $p = \text{profit}$
6. $g = \text{growth time}$
7. $d = \text{days left in season} = 28 - \text{current date}$
8. $r = \text{regrowth time}$
9. $n = \text{number of produce}$

Untuk strategi *greedy by profit per day*, harus diperhitungkan juga keuntungan tumbuhan yang dapat panen lebih dari sekali. Tanaman sekali panen memiliki *profit per day* berupa:

$$p/d = (s - c)/d \tag{1}$$

Sedangkan, untuk tumbuhan yang dapat dipanen lebih dari satu kali pada satu musim, kita harus menghitung berapa kali tanaman panen hingga akhir musim. Rumus yang digunakan adalah sebagai berikut:

$$n = \lfloor ((d + g)/r) + 1 \rfloor \tag{2}$$

Kemudian, diambil n terbesar. Setelah itu, *profit per day* dapat dihitung dengan rumus:

$$p/d = ((n \times s) - c) / d + 1 \tag{3}$$

Pada permasalahan pemilihan bibit tanaman diberikan suatu budget (B) dan bibit maksimum yang dapat dibeli untuk satu tanaman (m). Karena itu, properti objek selain waktu tumbuh akan dikalikan dengan m . Contohnya, apabila diberikan $m = 9$, maka untuk dua contoh tanaman (garlic dan Hops) tabel akan seperti berikut.

TABLE I. CONTOH PROPERTI OBJEK

Nama	Properti Objek						Dikali m			
	g	r	c	s	p	p/d	c	s	p	p/d
Garlic	4	0	40	60	20	5	360	540	180	45
Hops	11	1	60	25	-35	13	540	225	-315	117

Setelah dihitung masing-masing properti, dapat dilakukan penyelesaian masalah seperti pada *fractional knapsack problem*. Setelah itu, dapat dicari solusi algoritma *greedy* dengan *greedy by cost*, *sell price*, *profit*, dan *profit per day*. Total keuntungan merupakan banyak uang yang didapatkan dari mulai menanam tanaman hingga akhir musim, yaitu dihitung dengan:

$$\text{Total keuntungan} = p/d \times \text{fraksi solusi} \times \text{sisa hari} \tag{4}$$

Walaupun semua strategi *greedy* dicoba, yang memiliki kemungkinan akan menghasilkan solusi optimum paling besar adalah dengan menggunakan strategi *greedy by profit per day*.

Pada permasalahan ini, elemen-elemen *greedy* yang terbentuk adalah sebagai berikut.

1. Himpunan kandidat (C), yaitu semua jenis tanaman yang terdapat pada *game* bergantung pada musim yang dipilih.
2. Himpunan solusi (S), berisi semua tanaman yang terpilih untuk dibeli sesuai dengan penyelesaian *fractional knapsack problem*.

- Fungsi solusi, yaitu pengecekan jumlah semua bibit yang sudah terambil apakah sudah mendekati budget yang diberikan atau belum.
- Fungsi seleksi (*selection function*), yaitu keempat strategi greedy yang digunakan untuk mencari solusi permasalahan.
- Fungsi kelayakan (*feasible*), yaitu fungsi yang memeriksa apabila jika jenis tanaman baru dipilih harga belinya jika dijumlahkan dengan semua jenis tanaman yang sudah dipilih tidak akan melebihi budget yang diberikan.
- Fungsi objektif, tergantung pada strategi *greedy* yang dipakai. Harga beli minimum, harga jual maksimum, keuntungan maksimum, dan keuntungan per hari maksimum.

IV. STUDI KASUS

Pada bab ini akan diberikan beberapa contoh kasus permasalahan pemilihan jenis tanaman pada Stardew Valley. Karena tidak ada perbedaan pada tiap musim terkait pembelian tanaman (kecuali musim dingin yang tidak dapat menanam tanaman), maka untuk studi kasus ini akan dilakukan pada kasus musim panas.

Tabel tanaman musim semi adalah sebagai berikut:

TABLE II. TABEL TANAMAN MUSIM SEMI

No.	Nama	g	r	c	s	p
1	Blueberry	13	4	80	150	70
2	Corn	14	4	150	50	-100
3	Hops	11	1	60	25	-35
4	Hot Pepper	5	3	40	40	0
5	Melon	12	0	80	250	170
6	Poppy	7	0	100	140	40
7	Radish	6	0	40	90	50
8	Red Cabbage	9	0	100	260	160
9	Starfruit	13	0	400	750	350
10	Summer Spangle	8	0	50	90	40
11	Sunflower	8	0	200	80	-120
12	Tomato	11	4	50	60	10
13	Wheat	4	0	10	25	15

Untuk meminimalkan ukuran tabel solusi algoritma *greedy*, property objek langsung dikalikan dengan m.

Kasus 1: Pemain memiliki uang sebanyak 1000G untuk membeli bibit tanaman pada awal musim panas. Pemain ingin membeli maksimal 5 bibit untuk satu jenis tanaman.

Diketahui: $B = 1000$; $m = 5$; $d = 28 - 1 = 27$

Solusi algoritma *greedy*:

TABLE III. SOLUSI ALGORITMA GREEDY KASUS 1

Properti Objek							Greedy by			
i	g	r	c	s	p	p/d	c	s	p	p/d
1	13	4	400	750	350	92.9	0	0	0	0
2	14	4	750	250	-500	8.9	0	0	0	0
3	11	1	300	125	-175	65.2	1/6	0	0	0
4	5	3	200	200	0	50	1	0	0	0
5	12	0	400	1250	850	70.8	0	0	0	0
6	7	0	500	700	200	8.6	0	0	0	0
7	6	0	200	450	250	41.7	1	0	0	0
8	9	0	500	1300	800	88.9	0	0	0	0
9	13	0	2000	3750	1750	134.6	0	1/2	1/2	1/2
10	8	0	250	450	200	25	1	0	0	0
11	8	0	1000	400	-600	-75	0	0	0	0
12	11	4	250	300	50	44.6	1	0	0	0
13	4	0	50	125	75	18.8	1	0	0	0
Total Budget							1000	1000	1000	1000
Total Keuntungan							5347	1884	1884	1884

Solusi optimal dari permasalahan tersebut adalah:

$$X = (1, 0, 0, 0, \frac{1}{4}, 0, 0, 1, 0, 0, 0, 0, 0)$$

yang memberikan keuntungan maksimum 5588G.

Berdasarkan percobaan tersebut, dapat disimpulkan bahwa tidak ditemukan solusi optimal menggunakan algoritma *greedy* manapun, namun paling mendekati menggunakan *greedy by cost*. Hal ini dapat disebabkan pada kasus 1 uang yang tersedia kurang banyak, sedangkan terdapat satu jenis tanaman yang memiliki harga beli sangat tinggi namun memiliki keuntungan jual tinggi pula. Algoritma akan memilih tanaman tersebut dibanding tanaman lain yang memiliki harga beli jauh lebih rendah namun memiliki keuntungan cukup tinggi, sehingga akhirnya hanya tanaman tersebut yang terbeli.

Kasus 2: Pemain memiliki uang sebanyak 5000G untuk membeli bibit tanaman pada awal musim panas. Pemain ingin membeli maksimal 10 bibit untuk satu jenis tanaman.

Diketahui: $B = 5000$; $m = 10$; $d = 28 - 1 = 27$

Solusi algoritma *greedy*:

TABLE IV. SOLUSI ALGORITMA GREEDY KASUS 2

Properti Objek							Greedy by			
i	g	r	c	s	p	p/d	c	s	p	p/d
1	13	4	800	1500	750	185.8	0	0	0	1

Properti Objek							Greedy by			
<i>i</i>	<i>g</i>	<i>r</i>	<i>c</i>	<i>s</i>	<i>p</i>	<i>p/d</i>	<i>c</i>	<i>s</i>	<i>p</i>	<i>p/d</i>
2	14	4	1500	500	-1000	17.8	0	0	0	0
3	11	1	600	250	-350	130.4	0	0	0	0
4	5	3	400	400	0	100	1	0	0	0
5	12	0	800	2500	1700	141.6	0	0	1	0
6	7	0	1000	1400	400	17.2	0	0	0	0
7	6	0	400	900	500	83.4	1	0	0	0
8	9	0	1000	2600	1600	177.8	0	1	1/5	1/5
9	13	0	4000	7500	3500	269.2	0	1	1	1
10	8	0	500	900	400	50	1	0	0	0
11	8	0	2000	800	-1200	-150	0	0	0	0
12	11	4	500	600	100	89.2	1	0	0	0
13	4	0	100	250	150	37.5	1	0	0	0
Total Budget							5000	5000	5000	5000
Total Keuntungan							10,694	12,516	12,498	13,736

Solusi optimal dari permasalahan tersebut adalah:

$$X = (1, 0, 0, 0, 0, 0, 0, 1/5, 1, 0, 0, 0, 0)$$

yang memberikan keuntungan maksimum 13736G.

Berdasarkan percobaan tersebut, dapat disimpulkan bahwa cara mendapatkan solusi optimum pada kasus kedua adalah dengan menggunakan strategi *greedy by profit per day*.

Kasus 2 merupakan variasi dari kasus 1 apabila pemain memiliki uang yang jauh lebih banyak dan jumlah bibit maksimum lebih banyak pula. Kali ini pemilihan tanaman lebih akurat dan dapat menghasilkan solusi optimum dengan menggunakan salah satu strategi *greedy*.

Kasus 3: Pemain memiliki uang sebanyak 5000G untuk membeli bibit tanaman pada hari ke-12 musim panas. Pemain ingin membeli maksimal 10 bibit untuk satu jenis tanaman.

Diketahui: $B = 5000$; $m = 10$; $d = 28 - 12 = 16$

Solusi algoritma *greedy*:

TABLE V. SOLUSI ALGORITMA GREEDY KASUS 3

Properti Objek							Greedy by			
<i>i</i>	<i>g</i>	<i>r</i>	<i>c</i>	<i>s</i>	<i>p</i>	<i>p/d</i>	<i>c</i>	<i>s</i>	<i>p</i>	<i>p/d</i>
1	13	4	800	1500	750	41.2	0	0	0	0
2	14	4	1500	500	-1000	-58.8	0	0	0	0
3	11	1	600	250	-350	52.9	0	0	0	0
4	5	3	400	400	0	70.6	1	0	0	0
5	12	0	800	2500	1700	141.6	0	0	1	0
6	7	0	1000	1400	400	17.2	0	0	0	0

Properti Objek							Greedy by			
<i>i</i>	<i>g</i>	<i>r</i>	<i>c</i>	<i>s</i>	<i>p</i>	<i>p/d</i>	<i>c</i>	<i>s</i>	<i>p</i>	<i>p/d</i>
7	6	0	400	900	500	83.4	1	0	0	0
8	9	0	1000	2600	1600	177.8	0	1	1/5	1
9	13	0	4000	7500	3500	269.2	0	1	1	1
10	8	0	500	900	400	50	1	0	0	0
11	8	0	2000	800	-1200	-150	0	0	0	0
12	11	4	500	600	100	41.2	1	0	0	0
13	4	0	100	250	150	37.5	1	0	0	0
Total Budget							5000	5000	5000	5000
Total Keuntungan							4805	7599	7588	7599

Solusi optimal dari permasalahan tersebut adalah:

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

yang memberikan keuntungan maksimum 7599G.

Berdasarkan percobaan tersebut, dapat disimpulkan bahwa cara mendapatkan solusi optimum pada kasus kedua adalah dengan menggunakan strategi *greedy by profit per day*.

Jika dibandingkan kasus 2 dengan kasus 3, uang yang disediakan dan bibit maksimum yang dibeli sama. Yang membedakan hanya bahwa pada kasus 2 bibit tanaman dibeli pada hari ke-12 musim panas. Karena itu, tanaman-tanaman yang dapat panen berkali-kali tidak semenguntungkan pada kasus 2. Karena itu, dapat disimpulkan bahwa tanaman yang panen berkali-kali lebih baik ditanam di awal musim.

Kasus 4: Pemain memiliki uang sebanyak 700G untuk membeli bibit pada awal musim panas. Pemain ingin membeli maksimal 1 bibit untuk satu jenis tanaman.

Diketahui: $B = 700$; $m = 1$; $d = 28 - 1 = 27$

Solusi algoritma *greedy*:

TABLE VI. SOLUSI ALGORITMA GREEDY KASUS 4

Properti Objek							Greedy by			
<i>i</i>	<i>g</i>	<i>r</i>	<i>c</i>	<i>s</i>	<i>p</i>	<i>p/d</i>	<i>c</i>	<i>s</i>	<i>p</i>	<i>p/d</i>
1	13	4	80	150	70	18.6	1	1	1	1
2	14	4	150	50	-100	1.8	3/5	0	0	0
3	11	1	60	25	-35	13	1	0	0	2/3
4	5	3	40	40	0	10	1	0	0	0
5	12	0	80	250	170	13.1	1	1	1	1
6	7	0	100	140	40	5.7	1	2/5	0	0
7	6	0	40	90	50	8.3	1	0	1	0
8	9	0	100	260	160	17.8	1	1	1	1
9	13	0	400	750	350	26.9	0	1	1	1
10	8	0	50	90	40	5	1	0	0	0

Properti Objek							Greedy by			
<i>i</i>	<i>g</i>	<i>r</i>	<i>c</i>	<i>s</i>	<i>p</i>	<i>p/d</i>	<i>c</i>	<i>s</i>	<i>p</i>	<i>p/d</i>
11	8	0	200	80	-120	-15	0	0	0	0
12	11	4	50	60	10	8.9	1	0	0	0
13	4	0	10	25	15	3.8	1	0	0	0
Total Budget							700	700	700	700
Total Keuntungan							2948	2203	2372	2382

Solusi optimal dari permasalahan tersebut adalah:

$$X = (1, 0, 1, 1, 1, 1, 1, 1, 9/40, 1, 0, 1, 1)$$

yang memberikan keuntungan maksimum 3087G.

Berdasarkan percobaan tersebut, dapat disimpulkan bahwa tidak ditemukan solusi optimal menggunakan algoritma *greedy* manapun, namun paling mendekati menggunakan *greedy by cost*.

V. KESIMPULAN

Algoritma *greedy* merupakan algoritma yang banyak digunakan untuk masalah optimasi. Salah satu masalah yang dapat diselesaikan dengan *greedy* adalah permasalahan *knapsack*. Permasalahan *knapsack* pun dapat diaplikasikan pada banyak hal, salah satunya dalam memilih tanaman yang akan dibeli pada *game* Stardew Valley. Hal-hal yang harus dipertimbangkan dalam pemilihan adalah harga beli, harga jual, keuntungan, dan keuntungan per hari dari masing-masing tanaman.

Berdasarkan percobaan-percobaan yang telah dilakukan, dapat disimpulkan bahwa pemilihan tanaman untuk dibeli dan ditanam menggunakan algoritma *greedy* cukup berhasil. Walaupun terdapat beberapa percobaan yang kurang sukses, yaitu tidak menghasilkan solusi optimum, solusi dapat diaproksimasi dan sudah cukup mendekati solusi optimum. Ketidaktepatan pada hasil percobaan dapat dikarenakan kurangnya pengkajian oleh penulis.

UCAPAN TERIMA KASIH

Penulis ingin berterima kasih Allah SWT., seluruh dosen IF2211 Strategi Algoritma terutama Bapak Prof. Ir. Dwi Hendratmo Widyantoro, M.Sc., Ph.D. sebagai dosen pengampu K-03 yang telah mengajarkan mengenai strategi-strategi algoritma, khususnya algoritma *greedy*, sehingga penulis dapat membuat makalah ini berdasarkan materi yang telah diajarkan. Penulis juga ingin berterima kasih kepada semua pihak lain yang telah membantu dalam penulisan makalah ini, baik dengan memberikan pengetahuan mengenai strategi *greedy* maupun motivasi untuk menyelesaikan makalah ini.

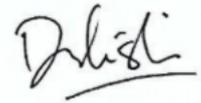
REFERENSI

- [1] Munir, Rinaldi. "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), accessed in 11/05/2021.
- [2] Munir, Rinaldi. "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), accessed in 11/05/2021.
- [3] Munir, Rinaldi. "Algoritma Greedy (Bagian 3)". [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag3.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag3.pdf), accessed in 11/05/2021.
- [4] "Algoritma Greedy". <http://dev.bertzzie.com/knowledge/analisis-algoritma/Greedy.html>, accessed in 10/05/2021.
- [5] http://a-research.upi.edu/operator/upload/s_d505_045711_chapter1.pdf, accessed in 11/05/2021.

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.

Jakarta, 11 Mei 2021



Delisha Azza Naadira 13519133