

Aplikasi Algoritma Program Dinamis dalam Tantangan Pembelian Bibit Tanaman pada Permainan *Rune Factory 3*

Jeremy Rionaldo Pasaribu - 13520082
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13520082@std.stei.itb.ac.id

Abstract—*Rune Factory 3* merupakan permainan simulasi *role-playing-game* yang memiliki mekanisme bertani yang salah satunya manfaatnya untuk menghasilkan keuntungan (*profit*) bagi pemain melalui hasil panen. Untuk membuat permainan *Rune Factory 3* menjadi lebih menarik, diberikan tantangan kepada pemain dengan pemain hanya boleh membeli 1 buah bibit saja untuk setiap jenis bibit tanaman yang tersedia menggunakan uang pemain yang terbatas untuk menghasilkan keuntungan kepada pemain semaksimal mungkin sepanjang musim tersebut. Makalah ini akan membahas cara untuk menyelesaikan tantangan tersebut dengan menggunakan metode program dinamis.

Keywords—*program dinamis; Rune Factory 3*

I. PENDAHULUAN

Rune Factory 3 merupakan permainan simulasi *role-playing-game* keempat dalam seri permainan *Rune Factory*. Permainan ini memiliki beberapa mekanisme untuk menghasilkan uang bagi pemain, yaitu mekanisme bertani (*farming*), mekanisme bertarung atau berburu, mekanisme memasak, dan lain-lain.



Gambar 1.1. *Gameplay* permainan *Rune Factory 3*.
(Sumber: Arsip Penulis)

Mekanisme bertani merupakan salah satu cara untuk menghasilkan uang secara mudah. Namun, pemain harus menunggu bibit tanaman yang telah ditanam untuk tumbuh menjadi hasil panen agar dapat dijual untuk menghasilkan uang. Harga jual hasil panen dari bibit tanaman tersebut akan selalu memiliki nilai yang lebih besar daripada harga awal

pembelian bibit tanaman. Sehingga, penjualan hasil panen tersebut akan menghasilkan keuntungan (*profit*) kepada pemain.

Untuk membuat permainan ini menjadi lebih menarik, diberikan tantangan kepada pemain dengan pemain hanya boleh membeli 1 buah bibit saja untuk setiap jenis bibit tanaman menggunakan uang pemain yang terbatas untuk menghasilkan keuntungan semaksimal mungkin sepanjang musim. Dengan tantangan ini, pemain akan merasa sulit untuk mencari bibit-bibit tanaman yang dapat menghasilkan keuntungan semaksimal mungkin kepada pemain. Maka, makalah ini akan membahas bagaimana cara untuk memilih bibit-bibit tanaman yang dapat menghasilkan keuntungan sebesar mungkin menggunakan metode program dinamis.

II. LANDASAN TEORI

A. Bertani dalam *Rune Factory 3*

Pemain dapat menghasilkan uang dengan cara bertani. Dalam bertani, pemain akan membutuhkan bibit tanaman yang akan ditanam. Salah satu cara untuk mendapatkan bibit tanaman yaitu membelinya dari toko yang tersedia. Berikut merupakan karakteristik bibit tanaman yang didapatkan secara eksplisit (melalui deskripsi bibit tanaman) dan implisit (melalui mekanisme bertani):

- **buy price:** harga beli bibit tanaman.
- **sell price:** harga jual dari hasil panen bibit tanaman.
- **season:** musim agar bibit tanaman dapat tumbuh.
- **growth:** waktu yang dibutuhkan suatu bibit tanaman untuk tumbuh menjadi hasil panen.
- **regrowth:** waktu yang dibutuhkan untuk tanaman memberikan hasil panen kembali.

Permainan ini memiliki 4 kategori musim (*season*) yaitu *Spring*, *Summer*, *Autumn*, dan *Winter* dengan rentang waktu sebanyak 30 hari. Setiap bibit tanaman perlu ditanam pada musim yang sesuai serta disiram dengan air setiap hari. Jika bibit tanaman ditanam pada musim yang tidak sesuai atau bibit tanaman tidak disiram untuk beberapa hari, bibit tanaman tersebut akan tumbuh menjadi rumput mati (*withered grass*). Bibit tanaman juga dapat dipercepat

pertumbuhannya dan ditingkatkan harga jual panennya menggunakan barang bantu berupa *formula* dan *greenifier*. *Greenifier* membantu untuk meningkatkan kualitas harga jual panen sedangkan *formula* membantu mempercepat pertumbuhan bibit tanaman. Barang bantu tersebut juga dijual pada toko yang tersedia.



Gambar 2.1. Kalender musim *Spring* dengan rentang 30 hari. (Sumber: *Arsip Penulis*)

Untuk mempermudah pembuatan makalah ini, pemain hanya akan membeli bibit tanaman dengan uang pemain tanpa membeli alat bantu untuk mempercepat pertumbuhan bibit tanaman ataupun meningkatkan harga jual hasil panen.

B. Knapsack Problem

Persoalan *knapsack* (*knapsack problem*) merupakan persoalan optimasi dengan diberikan n buah objek dan sebuah *knapsack* dengan kapasitas bobot M . Setiap objek memiliki bobot w_i dan keuntungan p_i . Persoalan ini meminta solusi untuk memilih objek-objek yang dimasukkan ke dalam *knapsack* sedemikian rupa sehingga diperoleh total keuntungan yang maksimal dengan total bobot objek tidak boleh melebihi kapasitas (M) *knapsack*. Terdapat banyak jenis dari persoalan *knapsack*, salah satunya yaitu *integer knapsack problem* atau *1/0 knapsack problem*.

$$\text{Maksimasi } F = \sum_{k=1}^n p_k x_k$$

dengan kendala (*constraint*)

$$\sum_{k=1}^n p_k x_k \leq M$$

yang dalam hal ini, $x_i = 0$ atau $1, i = 1, 2, \dots, n$

Gambar 2.2. Formulasi matematis *integer knapsack problem* (Sumber: [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))

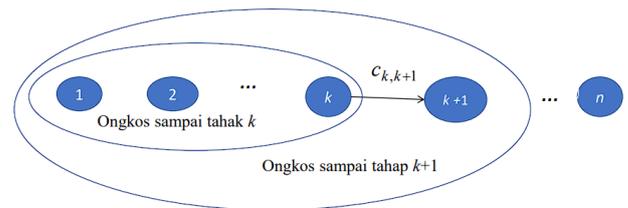
C. Program Dinamis

Program dinamis (*dynamic programming*) merupakan metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan (*stage*) serta membagi masalah persoalan menjadi sejumlah upamalah. Metode ini digunakan

untuk menyelesaikan persoalan-persoalan optimasi (maksimasi atau minimisasi).

Perbedaan metode program dinamis dengan algoritma *greedy* berada pada pembuatan rangkaian keputusan dalam menyelesaikan masalah. Algoritma *greedy* hanya menghasilkan satu rangkaian keputusan saja sebagai solusi optimal sedangkan program dinamis menghasilkan lebih dari satu rangkaian keputusan yang akan dipertimbangkan sebagai solusi optimal.

Metode program dinamis menggunakan prinsip optimalitas serta tahapan-tahapan dalam membuat rangkaian keputusan yang optimal. Prinsip optimalitas menyatakan jika solusi total optimal, maka bagian solusi sampai tahap ke- k juga optimal. Prinsip optimalitas juga menyatakan bahwa ketika algoritma bekerja dari tahap k sampai tahap $k+1$, algoritma dapat menggunakan hasil optimal dari tahap k tanpa kembali lagi ke tahap awal.



Gambar 2.2. Prinsip Optimalitas. (Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian1.pdf>)

Terdapat dua jenis pendekatan yang digunakan dalam program dinamis:

1. **Program dinamis maju (*forward / up-down*)**
Program dinamis bergerak mulai dari tahap 1, kemudian maju ke tahap 2, 3, dan seterusnya sampai tahap n .
2. **Program dinamis mundur (*backward / bottom-up*)**
Program dinamis bergerak mulai dari tahap n , kemudian mundur ke tahap $n - 1, n - 2$, dan seterusnya sampai tahap 1.

Selain pendekatan dalam program dinamis, terdapat juga langkah-langkah dalam pengembangan algoritma program dinamis sebagai berikut:

1. **Karakteristikan struktur solusi optimal**
Hal ini berisi penentuan tahap (*stage*), variabel keputusan, status (*state*), dan lain-lain.
2. **Definisikan secara rekursif nilai solusi optimal**
Tentukan hubungan nilai optimal suatu tahap dengan tahap sebelumnya.
3. **Hitung nilai solusi optimal secara maju atau mundur**
Perhitungan nilai solusi optimal akan dilakukan setiap tahap bisa secara maju yaitu dimulai dari tahap-1 atau

secara mundur yaitu dari tahap- n dengan tahap- n merupakan tahap terakhir dalam program dinamis.

4. Rekonstruksi solusi optimal (opsional)

Melakukan rekonstruksi solusi secara mundur

Dengan adanya pendekatan serta langkah-langkah pengembangan algoritma dinamis, persoalan optimasi dapat diselesaikan menggunakan metode program dinamis. Salah satu contoh persoalan optimasi yaitu persoalan *integer knapsack* (*integer knapsack problem*). Metode program dinamis akan digunakan dalam makalah ini untuk menyelesaikan tantangan pembelian bibit tanaman yang memiliki kemiripan dengan *integer knapsack problem*.

III. PEMBAHASAN DAN IMPLEMENTASI

A. Pemetaan Persoalan

Tantangan pembelian bibit tanaman ini mirip seperti *integer knapsack problem*. Berikut merupakan pemetaan persoalan bibit tanaman menjadi *integer knapsack problem*:

1. **Objek:** bibit tanaman yang tersedia dalam toko
2. **Kapasitas knapsack (M):** jumlah uang yang dimiliki oleh pemain
3. **Bobot (w):** harga beli dari bibit tanaman.
4. **Keuntungan (p):** keuntungan dari menjual hasil panen bibit tanaman

Sebelum membahas rumus keuntungan (p) dari bibit tanaman, berikut merupakan keterangan yang akan digunakan dalam rumus keuntungan:

- p = keuntungan hasil panen (*profit*)
- s = harga jual hasil panen (*sell price*)
- b = harga beli bibit tanaman (*buy price*)
- t = tanggal dalam permainan
- r = waktu yang dibutuhkan untuk tanaman memberikan hasil panen kembali (*regrowth*)
- g = waktu yang dibutuhkan bibit tanaman untuk tumbuh menghasilkan panen (*growth*)

Terdapat dua jenis rumus keuntungan (p) yang akan digunakan yaitu keuntungan untuk tanaman yang hanya sekali dipanen dan keuntungan untuk tanaman yang dapat beberapa kali dipanen. Rumus keuntungan untuk tanaman yang hanya sekali dipanen sebagai berikut:

$$p = (s - b) * [(30 - t) / g] \quad (1)$$

Sedangkan keuntungan untuk tanaman yang dapat beberapa kali dipanen sebagai berikut:

$$p = [d / 30] * (s - b + (s * [d / r])) \quad (2)$$

dengan

$$d = 30 - (t + g) \quad (3)$$

B. Analisis Persoalan dengan Program Dinamis

Karakteristik struktur optimal dalam tantangan pembelian bibit tanaman sebagai berikut:

- Tahap (k) adalah proses pembelian bibit tanaman menggunakan uang pemain.
- Status (y) adalah uang pemain yang tersisa setelah membeli bibit tanaman pada tahap sebelumnya.

Dalam penentuan solusi optimal, pemain membeli bibit tanaman pada tahap k sehingga uang pemain pada tahap selanjutnya akan menjadi $y - w_k$ dengan w merupakan harga beli bibit tanaman. Untuk membeli bibit tanaman selanjutnya, akan diterapkan prinsip optimalitas yang mengacu pada nilai optimum dari tahap sebelumnya yaitu $f_{k-1}(y - w_k)$. Kemudian, akan dibandingkan nilai keuntungan pembelian pada tahap k berupa $p_k + f_{k-1}(y - w_k)$ dengan nilai keuntungan $f_{k-1}(y)$ pada tahap sebelumnya. Jika $p_k + f_{k-1}(y - w_k)$ lebih besar dari $f_{k-1}(y)$, bibit tanaman ke- k akan dibeli. Relasi rekurens untuk persoalan tantangan ini adalah:

$$f_k(y) = 0, y = 0, 1, 2, \dots, M \quad (\text{basis})$$

$$f_k(y) = -\infty, y < 0 \quad (\text{basis})$$

$$f_k(y) = \max \{f_{k-1}(y), p_k + f_{k-1}(y - w_k)\}, \quad (\text{rekurens})$$

$$k = 1, 2, \dots, n$$

Gambar 3.1. Relasi rekurens untuk tantangan pembelian bibit tanaman. (Sumber: Arsip Penulis)

Berikut merupakan contoh penyelesaian persoalan tantangan pembelian bibit tanaman yang diambil dari referensi [1] dengan algoritma program dinamis yang diketahui jumlah bibit (n) = 3 dan uang pemain (M) = 5 Gold:

TABLE I. CONTOH PERSOALAN

i	Nama	W	P
1	Pink Turnip Seed	2	65
2	Cabbage Seeds	3	80
3	Strawberry Seed	1	30

Keterangan: i merupakan indeks dari bibit tanaman, $Nama$ merupakan nama dari bibit tanaman, P merupakan keuntungan dari bibit tanaman dengan rumus (1) atau (2), W merupakan harga beli dari bibit tanaman tersebut. Berikut merupakan tahap-tahap penyelesaian contoh persoalan tersebut:

TABLE II. TAHAP 1

$f_1(y) = \max \{f_0(y), p_1 + f_0(y - w_1)\}$ $= \max \{f_0(y), 65 + f_0(y - 2)\}$				
y	$f_0(y)$	$65 + f_0(y - 2)$	Solusi Optimum	
			$f_1(y)$	(x_1, x_2, x_3)
0	0	$-\infty$	0	(0, 0, 0)

1	0	$-\infty$	0	(0, 0, 0)
2	0	65	65	(1, 0, 0)
3	0	65	65	(1, 0, 0)
4	0	65	65	(1, 0, 0)
5	0	65	65	(1, 0, 0)

TABLE III. TAHAP 2

$$f_2(y) = \max\{f_1(y), p_2 + f_1(y - w_2)\}$$

$$= \max\{f_1(y), 80 + f_1(y - 3)\}$$

y	$f_0(y)$	$80 + f_1(y - 3)$	Solusi Optimum	
			$f_2(y)$	(x_1, x_2, x_3)
0	0	$80 + (-\infty) = -\infty$	0	(0, 0, 0)
1	0	$80 + (-\infty) = -\infty$	0	(0, 0, 0)
2	65	$80 + (-\infty) = -\infty$	65	(1, 0, 0)
3	65	$80 + 0 = 80$	80	(0, 1, 0)
4	65	$80 + 0 = 80$	80	(0, 1, 0)
5	65	$80 + 65 = 145$	145	(1, 1, 0)

TABLE IV. TAHAP 3

$$f_3(y) = \max\{f_2(y), p_3 + f_2(y - w_3)\}$$

$$= \max\{f_2(y), 30 + f_2(y - 1)\}$$

y	$f_0(y)$	$30 + f_2(y - 1)$	Solusi Optimum	
			$f_2(y)$	(x_1, x_2, x_3)
0	0	$30 + (-\infty) = -\infty$	0	(0, 0, 0)
1	0	$30 + (-\infty) = -\infty$	0	(0, 0, 0)
2	65	$30 + 0 = 30$	65	(1, 0, 0)
3	80	$30 + 65 = 95$	95	(1, 0, 1)
4	80	$30 + 80 = 110$	110	(0, 1, 1)
5	145	$30 + 80 = 110$	145	(1, 1, 0)

Solusi optimum yang didapatkan adalah: $X = (1, 1, 0)$ dengan total keuntungan yang didapatkan sebesar 45 Gold. Penyelesaian pada contoh persoalan ini menggunakan pendekatan program dinamis maju. Namun pada penyelesaian persoalan studi kasus selanjutnya, pencarian solusi optimum akan menggunakan pendekatan program dinamis mundur.

C. Implementasi Program Dinamis

Karena uang pemain (M) bisa memiliki kapasitas yang sangat besar, metode program dinamis akan diimplementasikan dalam bentuk program untuk mempermudah perhitungan. Berikut merupakan implementasi algoritma program dinamis dalam bahasa Java:

```
import java.util.List;

public class Solution {

    public static void DynamicProgrammingSolution
    (List<Seed> selectedSeed, int M) {
        int total_profit = 0;
        int n = selectedSeed.size();

        // Membuat list tahapan dengan tahap maksimum
        // adalah n
        int f[][] = new int[n + 1][M + 1];
        for (int i = 0; i <= n; i++) {
            for (int j = 0; j <= M; j++) {
                f[i][j] = 0;
            }
        }

        // Menghitung nilai keuntungan (p) dari semua
        // bibit tanaman dari tahap ke-1 sampai tahap
        // ke-n menggunakan prinsip optimalitas
        for (int k = 1; k <= n; k++) {
            for (int y = 0; y <= M; y++) {
                // W[k]
                int w = selectedSeed.get(k-1).getBuyPrice();
                // P[k]
                int p = selectedSeed.get(k-1).getProfit();
                if (w > y) {
                    f[k][y] = f[k - 1][y];
                } else {
                    f[k][y] = Math.max(f[k - 1][y], p +
                    f[k - 1][y - w]);
                }
            }
        }

        // Pencarian solusi dengan pendekatan
        // program dinamis mundur
        System.out.println("Program dinamis mundur: ");
        int solusi[] = new int[n];
        int k = n;
        while (k != 0) {
            if (f[k][M] != f[k - 1][M]) {
                System.out.println("Membeli bibit ke-" + k);
                solusi[k - 1] = 1;
                total_profit += selectedSeed.
                get(k - 1).getProfit();
                M = M - selectedSeed.
                get(k - 1).getBuyPrice();
            } else {
                solusi[k - 1] = 0;
            }
            k--;
        }

        // Menampilkan solusi optimal, uang sisa pemain,
        // serta total keuntungan yang didapatkan pemain
        System.out.print("\nSolusi Optimal:
        X = (" + solusi[0]);
        for (int i = 1; i < n; i++) {
            System.out.print(", ");
            System.out.print(solusi[i]);
        }
        System.out.print(")");
        System.out.println("\n
        Uang pemain yang tersisa: " + M);
    }
}
```

```
System.out.println("Total keuntungan pemain yang
    didapatkan: " + total_profit);
}
}
```

Program akan memberikan tabel bibit tanaman sesuai dengan musim permainan masukan pengguna serta menampilkan solusi optimal, uang pemain yang tersisa, serta total keuntungan pemain yang didapatkan.

```
PS D:\Programming\Java\Program Makalah> java -cp class Driver
Uang pemain (M): 1000
Tanggal permainan (t): 1
Musim permainan (Season): Spring
+-----+-----+-----+
| i | Nama | W | P |
+-----+-----+-----+
| 1 | Pink Turnip Seed | 80 | 1080 |
| 2 | Cabbage Seeds | 100 | 1120 |
| 3 | Strawberry Seed | 260 | 7660 |
| 4 | Onion Seeds | 110 | 1340 |
| 5 | Pink Melon Seed | 620 | 14320 |
| 6 | Moondrop Seed | 70 | 2120 |
| 7 | Toyherb Seed | 60 | 1680 |
| 8 | Cherry Grass Seed | 120 | 2960 |
| 9 | Lamp Grass Seed | 480 | 4100 |
| 10 | Emery Flower Seed | 1000 | 11800 |
| 11 | Blue Crystal Seed | 700 | 6300 |
| 12 | Gold Cabbage Seed | 230 | 7140 |
+-----+-----+-----+
Program dinamis mundur:
Membeli bibit ke-12
Membeli bibit ke-7
Membeli bibit ke-6
Membeli bibit ke-5
Solusi Optimal: X = (0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1)
Uang pemain yang tersisa: 20
Total keuntungan pemain yang didapatkan: 25260
PS D:\Programming\Java\Program Makalah>
```

Gambar 3.2. Tampilan penyelesaian tantangan pembelian bibit tanaman. (Sumber: Arsip Penulis)

IV. PENGUJIAN

Berikut merupakan beberapa studi kasus persoalan tantangan pembelian bibit tanaman dalam permainan *Rune Factory 3* untuk diuji:

A. Studi Kasus 1

Pada tanggal 1 dengan musim *Spring*, pemain memiliki uang (*M*) sebanyak 2000 Gold. Pemain ingin membeli bibit tanaman yang menghasilkan keuntungan semaksimal mungkin sepanjang musim tersebut. Berikut merupakan solusi dari algoritma program dinamis:

```
PS D:\Programming\Java\Program Makalah> java -cp class Driver
Uang pemain (M): 2000
Tanggal permainan (t): 1
Musim permainan (Season): Spring
+-----+-----+-----+
| i | Nama | W | P |
+-----+-----+-----+
| 1 | Pink Turnip Seed | 80 | 1080 |
| 2 | Cabbage Seeds | 100 | 1120 |
| 3 | Strawberry Seed | 260 | 7660 |
| 4 | Onion Seeds | 110 | 1340 |
| 5 | Pink Melon Seed | 620 | 14320 |
| 6 | Moondrop Seed | 70 | 2120 |
| 7 | Toyherb Seed | 60 | 1680 |
| 8 | Cherry Grass Seed | 120 | 2960 |
| 9 | Lamp Grass Seed | 480 | 4100 |
| 10 | Emery Flower Seed | 1000 | 11800 |
| 11 | Blue Crystal Seed | 700 | 6300 |
| 12 | Gold Cabbage Seed | 230 | 7140 |
+-----+-----+-----+
Program dinamis mundur:
Membeli bibit ke-12
Membeli bibit ke-9
Membeli bibit ke-8
Membeli bibit ke-7
Membeli bibit ke-6
Membeli bibit ke-5
Membeli bibit ke-4
Membeli bibit ke-3
Solusi Optimal: X = (0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1)
Uang pemain yang tersisa: 50
Total keuntungan pemain yang didapatkan: 41320
```

Gambar 4.1. Tampilan program dinamis dalam penyelesaian Studi Kasus 1. (Sumber: Arsip Penulis)

Solusi optimum yang didapatkan adalah: $X = (0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1)$. Dengan pendekatan program dinamis mundur, solusi optimal tersebut menghasilkan keuntungan maksimum untuk sepanjang musim tersebut sebesar 41320 Gold dengan uang sisa pemain sebesar 50 Gold.

B. Studi Kasus 2

Pada tanggal 20 dengan musim *Summer*, pemain memiliki uang (*M*) sebanyak 1500 Gold. Pemain ingin membeli bibit tanaman yang menghasilkan keuntungan semaksimal mungkin sepanjang musim tersebut. Berikut merupakan solusi dari algoritma program dinamis:

```
Uang pemain (M): 1500
Tanggal permainan (t): 20
Musim permainan (Season): Summer
+-----+-----+-----+
| i | Nama | W | P |
+-----+-----+-----+
| 1 | Cucumber Seed | 80 | 460 |
| 2 | Tomato Seed | 140 | 0 |
| 3 | Corn Seed | 130 | 0 |
| 4 | Pumpkin Seed | 220 | 1020 |
| 5 | Eggplant Seed | 160 | 320 |
| 6 | Pink Cat Seeds | 50 | 440 |
| 7 | Fire Flower Seed | 260 | 940 |
| 8 | IronLeaf Seed | 420 | 0 |
| 9 | Clover Seed | 390 | 1710 |
| 10 | G. Crystal Seed | 700 | 0 |
+-----+-----+-----+
Program dinamis mundur:
Membeli bibit ke-9
Membeli bibit ke-7
Membeli bibit ke-6
Membeli bibit ke-5
Membeli bibit ke-4
Membeli bibit ke-1
Solusi Optimal: X = (1, 0, 0, 1, 1, 1, 1, 0, 1, 0)
Uang pemain yang tersisa: 340
Total keuntungan pemain yang didapatkan: 4890
PS D:\Programming\Java\Program Makalah>
```

Gambar 4.2. Tampilan program dinamis dalam penyelesaian Studi Kasus 2. (Sumber: Arsip Penulis)

Solusi optimum yang didapatkan adalah: $X = (1, 0, 0, 1, 1, 1, 1, 0, 1, 0)$. Dengan pendekatan program dinamis mundur, solusi optimal menghasilkan keuntungan maksimum untuk sepanjang musim tersebut sebesar 4890 Gold dengan uang sisa pemain sebesar 340 Gold.

C. Studi Kasus 3

Pada tanggal 5 dengan musim *Autumn*, pemain memiliki uang (*M*) sebanyak 1100 Gold. Pemain ingin membeli bibit tanaman yang menghasilkan keuntungan semaksimal mungkin sepanjang musim tersebut. Berikut merupakan solusi dari algoritma program dinamis:

```
PS D:\Programming\Java\Program Makalah> java -cp class Driver
Uang pemain (M): 1100
Tanggal permainan (t): 5
Musim permainan (Season): Autumn
+-----+-----+-----+-----+
| i | Nama          | W   | P   |
+-----+-----+-----+-----+
| 1 | Potato Seed   | 90  | 1520|
| 2 | Carrot Seed   | 100 | 1470|
| 3 | Yam Seed      | 80  | 2720|
| 4 | Spinach Seed  | 60  | 960  |
| 5 | Green Pepper Seed | 150 | 8010 |
| 6 | Charm Blue Seed | 40  | 1120 |
| 7 | Autumn Grass Seed | 440 | 4960 |
| 8 | Pom-Pom G. Seed | 510 | 6090 |
| 9 | Red Crystal Seed | 700 | 6300 |
| 10| Gold Potato Seed | 250 | 6600 |
+-----+-----+-----+-----+
Program dinamis mundur:
Membeli bibit ke-10
Membeli bibit ke-8
Membeli bibit ke-6
Membeli bibit ke-5
Membeli bibit ke-4
Membeli bibit ke-3
Solusi Optimal: X = (0, 0, 1, 1, 1, 1, 0, 1, 0, 1)
Uang pemain yang tersisa: 10
Total keuntungan pemain yang didapatkan: 25500
PS D:\Programming\Java\Program Makalah>
```

Gambar 4.3. Tampilan program dinamis dalam penyelesaian Studi Kasus 3. (Sumber: *Arsip Penulis*)

Solusi optimum yang didapatkan adalah: $X = (0, 0, 1, 1, 1, 1, 0, 1, 0, 1)$. Dengan pendekatan program dinamis mundur, solusi optimal menghasilkan keuntungan maksimum untuk sepanjang musim tersebut sebesar 25500 Gold dengan uang sisa pemain sebesar 10 Gold.

D. Studi Kasus 4

Pada tanggal 7 dengan musim *Winter*, pemain memiliki uang (M) sebanyak 500 Gold. Pemain ingin membeli bibit tanaman yang menghasilkan keuntungan semaksimal mungkin sepanjang musim tersebut. Berikut merupakan solusi dari algoritma program dinamis:

```
PS D:\Programming\Java\Program Makalah> java -cp class Driver
Uang pemain (M): 500
Tanggal permainan (t): 7
Musim permainan (Season): Winter
+-----+-----+-----+-----+
| i | Nama          | W   | P   |
+-----+-----+-----+-----+
| 1 | Turnip Seed   | 130 | 2030|
| 2 | Leek Seed     | 120 | 1140|
| 3 | Daikon Seed   | 160 | 1830|
| 4 | W. Cabbage Seed | 200 | 1150 |
| 5 | Hot-Hot F. Seed | 470 | 4330 |
| 6 | Noel Grass Seed | 330 | 0    |
| 7 | W. Crystal Seed | 700 | 6300 |
| 8 | Gold Turnip Seed | 210 | 4760 |
+-----+-----+-----+-----+
Program dinamis mundur:
Membeli bibit ke-8
Membeli bibit ke-3
Membeli bibit ke-1
Solusi Optimal: X = (1, 0, 1, 0, 0, 0, 0, 1)
Uang pemain yang tersisa: 0
Total keuntungan pemain yang didapatkan: 8620
PS D:\Programming\Java\Program Makalah>
```

Gambar 4.4. Tampilan program dinamis dalam penyelesaian Studi Kasus 4. (Sumber: *Arsip Penulis*)

Solusi optimum yang didapatkan adalah: $X = (1, 0, 1, 0, 0, 0, 0, 1)$. Dengan pendekatan program dinamis mundur, solusi optimal menghasilkan keuntungan maksimum untuk sepanjang musim tersebut sebesar 8620 Gold dengan uang sisa pemain sebesar 0 Gold.

Berdasarkan atas pengujian studi kasus yang telah dilakukan, program dinamis berhasil menghasilkan solusi yang

optimum atas persoalan tantangan pembelian bibit tanaman. Sehingga, dapat disimpulkan bahwa program dinamis cocok dalam penyelesaian persoalan tantangan pembelian bibit tanaman.

V. PENUTUP

A. Kesimpulan

Rune Factory 3 merupakan permainan simulasi *role-playing-game* yang memiliki mekanisme bertani yang salah satunya manfaatnya untuk menghasilkan keuntungan (*profit*) bagi pemain melalui hasil panen. Untuk membuat permainan *Rune Factory 3* menjadi lebih menarik, diberikan tantangan kepada pemain dengan pemain hanya boleh membeli 1 buah bibit saja untuk setiap jenis bibit tanaman yang tersedia menggunakan uang pemain yang terbatas untuk menghasilkan keuntungan kepada pemain semaksimal mungkin sepanjang musim tersebut. Tantangan tersebut mengikuti persoalan *knapsack* dengan bobot (W) adalah harga beli dari bibit tanaman tersebut serta keuntungan (P) adalah harga keuntungan dari hasil panen bibit tanaman tersebut. Perhitungan keuntungan dari hasil panen bibit tanaman memiliki dua jenis yaitu keuntungan dari hasil sekali panen dan keuntungan dari hasil beberapa kali panen. Solusi optimal dari tantangan pembelian bibit tanaman didapatkan dengan algoritma program dinamis.

Algoritma program dinamis dibuat dengan menggunakan kode program mengikuti langkah-langkah yang mirip dalam penyelesaian persoalan *integer knapsack problem*. Program akan meminta masukan tanggal permainan (t), musim permainan (*Season*), serta uang pemain (M). Dilakukan pengujian untuk setiap studi kasus dan program dinamis menghasilkan solusi optimum untuk setiap studi kasus tersebut. Aplikasi algoritma program dinamis ini berhasil digunakan untuk menghasilkan solusi optimum dan cocok untuk penyelesaian persoalan tantangan pembelian bibit tanaman.

B. Saran

Jika diinginkan, penyelesaian tantangan pembelian bibit tanaman tersebut dapat dibandingkan dengan algoritma pendekatan lain seperti algoritma *greedy*, algoritma *branch and bound*, dan lain-lain. Kemudian, hasil dari pendekatan algoritma tambahan tersebut dapat dibandingkan dengan algoritma program dinamis untuk mengetahui apakah algoritma program dinamis lebih efisien atau tidak.

LINK KODE PROGRAM

<https://github.com/JeremyRio/ProgramMakalahStima>

LINK VIDEO YOUTUBE

<https://youtu.be/kErhi9Hmrrw>

UCAPAN TERIMA KASIH

Puji syukur penulis ucapkan kepada Tuhan yang Maha Esa atas berkat, rahmat, dan karunia-Nya yang membuat penulis dapat menyelesaikan makalah ini dengan baik dan tepat

waktu. Penulis juga mengucapkan terima kasih kepada ibu Dr. Masayu Leylia Khodra, S.T, M.T. dan ibu Dr. Nur Ulfa Maulidevi, S.T., M.Sc. selaku dosen pengajar mata kuliah IF221 Strategi Algoritma atas bimbingan dan pengajaran teori-teori dasar dari makalah ini serta bapak Dr. Ir. Rinaldi Munir, MT. selaku dosen pengajar mata kuliah IF221 Strategi Algoritma atas pengajaran teori-teori dasar serta situs referensi yang digunakan dalam makalah. Penulis ingin meminta maaf apabila makalah ini terdapat kesalahan dalam tutur kata dan kekurangan data yang dapat membuat informasi menjadi tidak akurat.

REFERENSI

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian1.pdf>, diakses pada tanggal 19 Mei 2022, pukul 16.35 WIB.
- [2] [https://therunefactory.fandom.com/wiki/Crops_\(RF3\)](https://therunefactory.fandom.com/wiki/Crops_(RF3)), diakses pada tanggal 19 Mei 2022, pukul 17.23 WIB.

- [3] <https://www.guru99.com/knapsack-problem-dynamic-programming.html>, diakses pada tanggal 20 Mei 2022, pukul 10.11 WIB.

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, 22 Mei 2022



Jeremy Ronaldo Pasaribu - 13520082