

Pembuatan Sugesti Pengolahan Makanan Dalam Permainan “Don’t Starve” Menggunakan Strategi *Dynamic Programming* Dengan Pemodelan *Integer Knapsack*

Yoga Prasetyo/13515148¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

¹13515148@std.stei.itb.ac.id

Abstrak – “Don’t Starve” merupakan salah satu permainan komputer bergaya *survival*. Sebagai salah satu permainan dengan gaya *survival*, kemampuan bertahan hidup adalah salah satu *skill* yang diutamakan dalam memainkan permainan ini. Sesuai namanya, yang berarti “Jangan Kelaparan”, permainan ini difokuskan untuk bertahan hidup dengan cara jangan kelaparan. Sering kali para pemain kebingungan mengolah makanan yang bahan mereka punya dalam permainan don’t starve. Dengan maraknya perkembangan teknologi, maka permasalahan tersebut bisa diselesaikan, contohnya dengan strategi *Dynamic Programming* dengan pemodelan *knapsack*. Oleh karena itu, penulis akan membahas cara pembuatan sugesti pengolahan makanan dalam permainan “Don’t Starve” menggunakan strategi *Dynamic Programming* dengan pemodelan *Knapsack*.

Keywords—Permainan “Don’t Starve”, *Dynamic Programming*, *Knapsack*.

I. PENDAHULUAN

“Don’t Starve” merupakan salah satu permainan komputer bergaya *survival* dengan tampilan grafis 2D. “Don’t Starve” dapat dibeli secara digital dari *online marketplace* bernama “Steam” dengan kisaran Rp.96.000,- atau \$14.99. “Don’t Starve” mendapatkan rating sebesar 97% sejak dirilis pada tahun 2013 sampai sekarang.



Gambar 1. Menu Utama Permainan “Don’t Starve” (sumber: komputer pribadi)

Sebagai salah satu permainan dengan gaya *survival*, kemampuan bertahan hidup adalah salah satu *skill* yang diutamakan dalam memainkan permainan ini. Kemampuan bertahan hidup tersebut meliputi *researching*, *crafting*, *hunting*, *farming* dan *fighting* tanpa adanya instruksi atau petunjuk. Dengan memiliki kemampuan tersebut, pemain bisa bertahan hidup dalam waktu yang cukup lama. Pada saat memulai permainan, pemain akan memilih karakter (apabila baru main untuk pertama kalinya, maka karakter yang bisa dipilih hanya satu) dan kemudian dibawa kepada sebuah dunia dengan *resource*, *enemy* dan lain-lain yang di *generate* secara *random*. Karakter sendiri memiliki *stats* seperti *health*, *sanity* dan *hunger*. Pada mulai permainan, pemain memiliki *health*, *sanity* dan *hunger* yang penuh.

Permainan ini memiliki 4 musim dalam setahun yang berdurasi selama 70 hari, dengan keuntungan dan kesulitannya masing-masing. Secara *default*, saat permainan dimulai, pemain akan bermain pada musim gugur. Secara *default*, musim gugur berlangsung selama 20 hari. Pada musim gugur, belum ada tantangan yang terlalu sulit, cukup dengan hanya bertahan hidup



Gambar 2. Cuplikan permainan “Don’t Starve” (sumber: komputer pribadi)

dengan tidak kelaparan. Pada musim gugur juga sumber makanan cukup banyak, mulai dari buah-buahan, sayur-sayuran, *birchnut*, daging-dagingan sampai es. Karena itulah musim gugur dijadikan musim *default* saat permainan dimulai. Kemudian setelah musim gugur ada musim dingin. Musim

dingin adalah saat dimana tantangan dimulai. Secara *default*, musim dingin berlangsung selama 15 hari. Pada musim dingin, karakter bisa merasakan kedinginan (*freezing*), oleh karena itu pemain harus tetap hangat agar bisa bertahan hidup. Selain itu juga, musim dingin tidak memungkinkan pemain untuk melakukan *farming*, karena selama musim dingin, tanaman tidak tumbuh. Sumber makanan pada musim dingin, sumber makanan meliputi daging-dagingan, telur dan es. Oleh karena itu, pemain harus mencari makanan yang banyak pada musim gugur agar persediaan makanannya cukup juga untuk musim dingin. Kemudian setelah musim dingin ada musim semi. Secara *default*, musim semi berlangsung selama 20 hari. Pada musim semi, hujan sering turun yang mengakibatkan *sanity* pemain cepat turun dan makanan menjadi cepat basi. Tetapi keuntungannya, kegiatan *farming* berlangsung lancar karena hujan akan mempercepat pertumbuhan tanaman. Pada musim semi juga, sumber daging yang paling mudah didapat, yaitu kelinci, tidak keluar. Jadi sumber makanan pada musim semi meliputi sayur-sayuran dan buah-buahan. Kemudian setelah musim semi ada musim panas. Secara *default*, musim panas berlangsung selama 15 hari. Pada musim panas, pemain bisa merasa kepanasan (*overheating*), oleh karena itu pemain harus tetap merasa sejuk agar bisa bertahan hidup. Pada musim panas, akan terjadi kebakaran pada tanaman, *enemy*, bahkan barang-barang yang sudah kita *craft*. Kebakaran ini ditandai dengan adanya asap kecil. Kebakaran bisa diredakan dengan es, sayangnya es tidak selalu ada saat musim panas. Karena itulah kemampuan bertahan hidup dan kemampuan berfikir cerdas diperlukan untuk bisa bertahan hidup dalam permainan ini.

II. MAKANAN PADA “DON’T STARVE”

Makanan pada permainan “Don’t Starve” terbagi menjadi 2 jenis, yaitu *Food Ingredients* dan *Crock Pot Meals*. *Food Ingredients* adalah makanan yang bisa kita dapatkan di jalan, saat *farming* maupun yang kita dapatkan saat *fighting*. *Food Ingredients* bisa dibagi menjadi beberapa kategori seperti pada tabel berikut.

Jenis	Contoh
Daging	Drumstick, Fried Drumstick, Meat, Cooked Meat, Monster Meat, Cooked Mosnter Meat, Morsel, Cooked Morsel, Frog Legs, Cooked Frog Legs
Ikan	Fish, Cooked Fish

Sayur	Carrot, Roasted Carrot, Corn, Popcorn, Eggplant, Braised Eggplant, Pumpkin, Hot Pumpkin, Blue Cap, Cooked Blue Cap, Green Cap, Cooked Green Cap, Red Cap, Cooked Red Cap
Buah	Pomegranate, Sliced Pomegranate, Dragon Fruit, Prepared Dragon Fruit, Durian, Extra Smelly Durian, Berries, Roasted Berries, Watermelon, Grilled Watermelon
Telur	Egg, Cooked Egg, Tallbird Egg, Fired Tallbird Egg

Tabel 1. Tabel *Food Ingredients* Umum

Sedangkan *Crock Pot Meals* adalah *Food Ingredients* yang kita olah menjadi sebuah makanan baru dengan alat yang bernama *Crock Pot*, dengan maksud untuk menambah lebih banyak *health*, *hunger*, atau *sanity*. *Crock Pot Meals* juga terbagi menjadi beberapa kategori seperti pada tabel berikut.

Jenis	Contoh
Daging	Meatballs, Turkey Dinner, Froggle Bunwich, Bacon and Eggs, Pierogi, Meaty Stew, Kabobs
Ikan	Fish Tacos, Fish Sticks
Sayur	Pumpkin Cookie, Ratatouille, Stuffed Eggplant
Buah	Dragonpie, Fist Full of Jam, Melonsicle, Trail Mix

Tabel 2. Tabel *Crock Pot Meals* Umum

III. LANDASAN TEORI

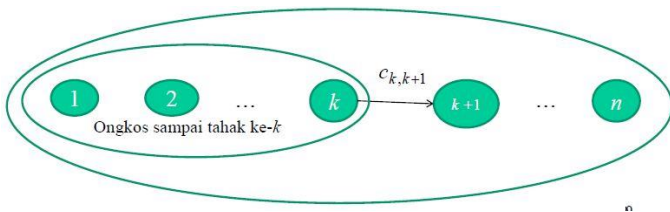
A. Dynamic Programming

Dynamic Programming adalah metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan (*stage*) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan. Istilah “*Dynamic Programming*” muncul karena perhitungan solusi menggunakan tabel-tabel. *Dynamic Programming* sering dikaitkan dengan Algoritma *Greedy*, karena secara umum, cara menyelesaikan persoalannya hampir sama. Tetapi ada sedikit perbedaan antara Algoritma *Greedy* dan *Dynamic Programming*, yaitu Algoritma *Greedy* hanya mempertimbangkan satu rangkaian keputusan, sedangkan *Dynamic Programming* mempertimbangkan lebih dari satu rangkaian keputusan.

B. Prinsip Optimalitas

Pada *Dynamic Programming*, rangkaian keputusan yang optimal dibuat dengan menggunakan Prinsip Optimalitas. Prinsip Optimalitas ini sendiri berarti “jika solusi total optimal, maka bagian solusi sampai tahap ke- k juga optimal”.

Prinsip Optimalitas berarti bahwa jika kita bekerja dari tahap k ke tahap $k+1$, kita dapat menggunakan hasil optimal dari tahap k tanpa harus kembali ke tahap awal. Ongkos pada tahap $k+1$ = (ongkos yang dihasilkan pada tahap k) + (ongkos yang dihasilkan dari tahap k ke tahap $k+1$)



Gambar 3. Ilustrasi Prinsip Optimalitas (sumber: [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Program-Dinamis-\(2018\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Program-Dinamis-(2018).pdf))

C. Karakteristik Persoalan Dynamic Programming

1. Persoalan dapat dibagi menjadi beberapa tahap (*stage*), yang pada setiap tahap hanya diambil satu keputusan
2. Masing-masing tahap terdiri dari sejumlah status (*state*) yang berhubungan dengan tahap tersebut. Secara umum, status merupakan bermacam kemungkinan masukan yang ada pada tahap tersebut.
3. Hasil dari keputusan yang diambil pada setiap tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
4. Ongkos (*cost*) pada suatu tahap meningkat secara teratur (*steadily*) dengan bertambahnya jumlah tahapan.
5. Ongkos pada suatu tahap bergantung pada ongkos tahap-tahap yang sudah berjalan dan ongkos pada tahap tersebut.
6. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan pada tahap sebelumnya.

7. Adanya hubungan rekursif yang mengidentifikasi keputusan terbaik untuk setiap status pada tahap k memberikan keputusan terbaik untuk setiap status pada tahap $k+1$.
8. Prinsip Optimalitas berlaku pada persoalan tersebut.

D. Integer (1/0) Knapsack

Pada persoalan ini,

1. Tahap (k) adalah proses memasukan barang ke dalam karung (*knapsack*) (ada 3 tahap)
2. Status (y) menyatakan kapasitas muat karung yang tersisa setelah memasukan barang pada tahap sebelumnya.

Dari tahap ke-1, kita masukkan objek ke-1 ke dalam karung untuk setiap satuan kapasitas karung sampai batas kapasitas maksimumnya. Karena kapasitas karung adalah bilangan bulat, maka pendekatan ini praktis.

Misalkan ketika memasukkan objek pada tahap k , kapasitas muat karung sekarang adalah $y-w_k$. Untuk mengisi kapasitas sisanya, kita menerapkan Prinsip Optimalitas dengan mengacu pada nilai optimum dari tahap sebelumnya untuk kapasitas sisa $y-w_k$ (yaitu $f_{k-1}(y-w_k)$).

Selanjutnya, kita bandingkan nilai keuntungan dari objek pada tahap k (yaitu p_k) plus nilai $f_{k-1}(y-w_k)$ dengan keuntungan pengisian hanya $k-1$ macam objek, $f_{k-1}(y)$. Jika $p_k + f_{k-1}(y-w_k)$ lebih kecil dari $f_{k-1}(y)$, maka objek yang ke- k tidak dimasukkan kedalam karung, tetapi jika lebih besar, maka objek yang ke- k dimasukkan.

Relasi rekurens untuk persoalan ini adalah

$f_0(y) = 0,$	$y = 0, 1, 2, \dots, M$	(basis)
$f_k(y) = -\infty,$	$y < 0$	(basis)
$f_k(y) = \max\{f_{k-1}(y), p_k + f_{k-1}(y-w_k)\},$		(rekurens)
$k = 1, 2, \dots, n$		

- $f_k(y)$ adalah keuntungan optimum dari persoalan *Integer (1/0) Knapsack* pada tahap k untuk kapasitas karung sebesar y .
- $f_0(y) = 0$ adalah nilai dari persoalan *knapsack* kosong (tidak ada persoalan *knapsack*) dengan kapasitas y .
- $f_k(y) = -\infty$ adalah nilai dari persoalan *knapsack* untuk kapasitas negatif. Solusi optimum dari persoalan *Integer (1/0) Knapsack* adalah $f_n(M)$.

IV. METODOLOGI PENELITIAN

Sebelum masuk ke metodologi penelitian, ada beberapa asumsi yang dipakai dalam metodologi penelitian kali ini :

1. Asumsikan pemain dalam keadaan sehat (*health* penuh) dan waras (*sanity* penuh), sehingga kesehatan dan kewarasan bisa diabaikan dan hanya berfokus pada kelaparan pada metodologi kali ini.
2. Asumsikan pemain memiliki tidak lebih dari 10 *Food Ingredients*.
3. Pada metodologi ini, *knapsack* akan disebut sebagai *Crock Pot*, dengan kapasitas 4.
4. Asumsi *weight* untuk setiap barang sama, yaitu 1.

5. Dikarenakan harus ada penghematan terhadap pengolahan makanan yang memiliki *hunger* yang tinggi, maka penghitungan profit adalah 50-*hunger*
6. Berikut adalah tabel profit setiap *Food Ingredients* yang umum ditemukan dalam permainan “Don’t Starve” beserta dengan kategori untuk masing-masing makanan

Nama	Hunger	Profit	Kategori
Meat	25	25	Daging
Monster Meat	18.75	31.25	Daging
Morsel	12.5	37.5	Daging
Carrot	12.5	37.5	Sayur
Berries	9.375	40.625	Buah
Pumpkin	37.5	12.5	Sayur
Eggplant	25	25	Sayur
Red Cap	12.5	37.5	Sayur

Tabel 3. Tabel Profit Setiap Makanan Yang Umum Ditemukan Pada Permainan “Don’t Starve”

7. Berikut adalah tabel resep untuk *Crock Pot Meals* yang bisa dibuat dari *Food Ingredients* diatas, beserta *hunger* yang dihasilkan. Pada bagian daging, sayur dan buah, angka menunjukkan jumlah minimal makanan dengan kategori tersebut harus ada dalam *Crock Pot*. Kemudian tanda ✖ berarti tidak boleh ada makanan dengan kategori tersebut di dalam *Crock Pot*. Kemudian tanda ✓ berarti makanan dengan kategori tersebut boleh ada dan boleh tidak dalam *Crock Pot*.

Nama	Hunger	Daging	Sayur	Buah
Meatballs	62.5	1	✓	✓
Meaty Stew	150	3	✓	✓
Ratatouille	25	✖	1	✓
Fist Full of Jam	37.5	✖	✖	1

Tabel 4. Tabel Resep Crock Pot Meals Beserta Hunger Yang Dihasilkan

Berikut ini adalah langkah-langkah dari pembuatan sugesti pengoahan makanan pada permainan “Don’t Starve” menggunakan strategi *Dynamic Programming* dengan metode *Integer Knapsack* :

1. Lihat *Food Ingredients* yang dimiliki oleh pemain, pada kesempatan kali ini penulis hanya akan memberikan 2 buah contoh *Food Ingredients* yang dimiliki pemain :

- Contoh 1

Nama	Jumlah
Berries	4
Carrot	2
Pumpkin	1
Red Cap	2

Tabel 5. Tabel Food Ingredients Contoh 1

- Contoh 2

Nama	Jumlah
Meat	1
Carrot	2
Eggplant	2
Red Cap	2

Tabel 6. Tabel Food Ingredients Contoh 2

2. Kemudian lakukan perhitungan sesuai teori di atas dengan representasi menggunakan tabel (asumsi x_1 - x_9 mengurut sesuai dengan urutan pada tabel, beserta jumlahnya, jadi x_1 - x_4 adalah berries, x_5 - x_6 adalah carrot dst). Perhitungan dilakukan pada contoh 1

- Tahap 1

$$f_1(y) = \max\{f_0(y), p_1 + f_0(y-w_1)\}$$

$$= \max\{f_0(y), 40.625 + f_0(y-1)\}$$

		Solusi Optimum		
y	$f_0(y)$	$p_1 + f_0(y-1)$	$f_1(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*)$
0	0	$-\infty$	0	(0,0,0,0,0,0,0,0,0)
1	0	40.625	40.625	(1,0,0,0,0,0,0,0,0)
2	0	40.625	40.625	(1,0,0,0,0,0,0,0,0)
3	0	40.625	40.625	(1,0,0,0,0,0,0,0,0)
4	0	40.625	40.625	(1,0,0,0,0,0,0,0,0)

Tabel 7. Tabel Perhitungan Tahap 1

- Tahap 2

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

$$= \max\{f_1(y), 40.625 + f_1(y-1)\}$$

		Solusi Optimum		
Y	$f_1(y)$	$p_2 + f_1(y-1)$	$f_2(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*)$
0	0	$-\infty$	0	(0,0,0,0,0,0,0,0,0)
1	40.625	40.625	40.625	(1,0,0,0,0,0,0,0,0)
2	40.625	81.25	81.25	(1,1,0,0,0,0,0,0,0)
3	40.625	81.25	81.25	(1,1,0,0,0,0,0,0,0)
4	40.625	81.25	81.25	(1,1,0,0,0,0,0,0,0)

Tabel 8. Tabel Perhitungan Tahap 2

- Tahap 3

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

$$= \max\{f_2(y), 40.625 + f_2(y-1)\}$$

- Tahap 6

$$f_6(y) = \max\{f_5(y), p_6 + f_5(y-w_6)\}$$

$$= \max\{f_5(y), 37.5 + f_5(y-1)\}$$

Solusi Optimum				
y	$f_5(y)$	$p_6 + f_5(y-1)$	$f_6(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*)$
0	0	$-\infty$	0	(0,0,0,0,0,0,0,0,0)
1	40.625	37.5	40.625	(1,0,0,0,0,0,0,0,0)
2	81.25	78.125	81.25	(1,1,0,0,0,0,0,0,0)
3	121.875	118.75	121.875	(1,1,1,0,0,0,0,0,0)
4	162.5	159.375	162.5	(1,1,1,1,0,0,0,0,0)

Tabel 12. Tabel Perhitungan Tahap 6

- Tahap 7

$$f_7(y) = \max\{f_6(y), p_7 + f_6(y-w_7)\}$$

$$= \max\{f_6(y), 12.5 + f_6(y-1)\}$$

Solusi Optimum				
y	$f_6(y)$	$p_7 + f_6(y-1)$	$f_7(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*)$
0	0	$-\infty$	0	(0,0,0,0,0,0,0,0,0)
1	40.625	12.5	40.625	(1,0,0,0,0,0,0,0,0)
2	81.25	53.125	81.25	(1,1,0,0,0,0,0,0,0)
3	121.875	93.75	121.875	(1,1,1,0,0,0,0,0,0)
4	162.5	134.375	162.5	(1,1,1,1,0,0,0,0,0)

Tabel 13. Tabel Perhitungan Tahap 7

- Tahap 8

$$f_8(y) = \max\{f_7(y), p_8 + f_7(y-w_8)\}$$

$$= \max\{f_7(y), 37.5 + f_7(y-1)\}$$

Solusi Optimum				
y	$f_7(y)$	$p_8 + f_7(y-1)$	$f_8(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*)$
0	0	$-\infty$	0	(0,0,0,0,0,0,0,0,0)
1	40.625	37.5	40.625	(1,0,0,0,0,0,0,0,0)
2	81.25	78.125	81.25	(1,1,0,0,0,0,0,0,0)
3	121.875	118.75	121.875	(1,1,1,0,0,0,0,0,0)
4	162.5	159.375	162.5	(1,1,1,1,0,0,0,0,0)

Tabel 14. Tabel Perhitungan Tahap 8

Solusi Optimum				
y	$f_2(y)$	$p_3 + f_2(y-1)$	$f_3(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*)$
0	0	$-\infty$	0	(0,0,0,0,0,0,0,0,0)
1	40.625	40.625	40.625	(1,0,0,0,0,0,0,0,0)
2	81.25	81.25	81.25	(1,1,0,0,0,0,0,0,0)
3	81.25	121.875	121.875	(1,1,1,0,0,0,0,0,0)
4	81.25	121.875	121.875	(1,1,1,0,0,0,0,0,0)

Tabel 9. Tabel Perhitungan Tahap 3

- Tahap 4

$$f_4(y) = \max\{f_3(y), p_4 + f_3(y-w_4)\}$$

$$= \max\{f_3(y), 40.625 + f_3(y-1)\}$$

Solusi Optimum				
y	$f_3(y)$	$p_4 + f_3(y-1)$	$f_4(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*)$
0	0	$-\infty$	0	(0,0,0,0,0,0,0,0,0)
1	40.625	40.625	40.625	(1,0,0,0,0,0,0,0,0)
2	81.25	81.25	81.25	(1,1,0,0,0,0,0,0,0)
3	121.875	121.875	121.875	(1,1,1,0,0,0,0,0,0)
4	121.875	162.5	162.5	(1,1,1,1,0,0,0,0,0)

Tabel 10. Tabel Perhitungan Tahap 4

- Tahap 5

$$f_5(y) = \max\{f_4(y), p_5 + f_4(y-w_5)\}$$

$$= \max\{f_4(y), 37.5 + f_4(y-1)\}$$

Solusi Optimum				
y	$f_4(y)$	$p_5 + f_4(y-1)$	$f_5(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*)$
0	0	$-\infty$	0	(0,0,0,0,0,0,0,0,0)
1	40.625	40.625	40.625	(1,0,0,0,0,0,0,0,0)
2	81.25	81.25	81.25	(1,1,0,0,0,0,0,0,0)
3	121.875	121.875	121.875	(1,1,1,0,0,0,0,0,0)
4	121.875	162.5	162.5	(1,1,1,1,0,0,0,0,0)

Tabel 11. Tabel Perhitungan Tahap 5

- Tahap 9

$$f_9(y) = \max\{f_8(y), p_9 + f_8(y-w_9)\} \\ = \max\{f_8(y), 37.5 + f_8(y-1)\}$$

			Solusi Optimum	
y	$f_8(y)$	$p_9 + f_8(y-1)$	$f_9(y)$	$(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*)$
0	0	$-\infty$	0	(0,0,0,0,0,0,0,0,0)
1	40.625	37.5	40.625	(1,0,0,0,0,0,0,0,0)
2	81.25	78.125	81.25	(1,1,0,0,0,0,0,0,0)
3	121.875	118.75	121.875	(1,1,1,0,0,0,0,0,0)
4	162.5	159.375	162.5	(1,1,1,1,0,0,0,0,0)

Tabel 15. Tabel Perhitungan Tahap 9

Solusi optimum $X = (1,1,1,1,0,0,0,0,0)$ dengan $\Sigma p = f = 162.5$

- Periksa apakah sesuai dengan resep yang ada atau tidak
 - Makanan yang diambil adalah 4 buah *berries*, yang berada di kategori buah. Berarti 4 buah *berries* bisa menghasilkan makanan sesuai resep, yaitu *Fist Full of Jam*, masukkan makanan ke *Crock Pot* lalu lanjut ke langkah 4
 - Jika tidak sesuai dengan resep diatas, maka lanjut ke langkah 5, dan makanan hasil perhitungan tadi tidak dimasukkan ke dalam *Crock Pot*, namun tidak diikutsertakan dalam perhitungan selanjutnya (jika ada).
- Tinjau makanan sisa yang pemain miliki. Jika jumlah makanan $\geq$ kapasitas *Crock Pot*, maka lanjut ke langkah 5. Jika tidak, maka lanjut ke langkah 6.

Nama	Jumlah
Carrot	2
Pumpkin	1
Red Cap	2

Tabel 16. Tabel Sisa Makanan Yang Dimiliki Pemain

- Ulangi langkah 2-4 sampai tidak ada lagi makanan yang bisa dimasukkan ke dalam *Crock Pot*. Lalu lanjut ke langkah 6
- Meninjau semua makanan hasil dari perhitungan diatas.
 - Pada contoh 1 diatas, didapatlah 1 *Fist Full of Jam*, 1 *Ratatouille* dan sisa 1 *Pumpkin*.

V. KESIMPULAN

Jadi, dengan strategi *dynamic programming* dan metode *integer knapsack*, kita bisa membuat sugesti pengolahan makanan dalam permainan "Don't Starve".

Caranya adalah pertama meninjau makanan yang pemain miliki. Lalu lakukan perhitungan berdasarkan teori yang telah

dijelaskan di atas sebanyak n tahap, dimana n adalah jumlah makanan yang dimiliki pemain saat itu. Kemudian setelah didapatkan solusi optimum, periksa apakah kombinasi dari solusi optimum tersebut sesuai dengan salah satu resep yang ada. Jika sesuai maka masukkan makanan tersebut ke dalam *knapsack*. Jika tidak sesuai, lakukan langkah berikutnya tanpa mengikutsertakan solusi optimum tadi. Langkah berikutnya adalah meninjau jumlah sisa makanan yang pemain miliki. Jika $\geq$ kapasitas *knapsack*, maka lanjut ke langkah berikutnya. Jika tidak, maka berhenti. Lalu langkah berikutnya adalah mengulang langkah dari langkah perhitungan sampai langkah sebelum ini. Setelah selesai, tinjau kembali hasil makanan yang didapatkan beserta sisa makanan yang tidak masuk ke dalam *knapsack*.

Dengan pengembangan lebih lanjut, aplikasi sugesti pengolahan makanan pada permainan "Don't Starve" ini bisa menjadi lebih besar dan mengcover seluruh *Food Ingredients* dan *Crock Pot Meals* yang ada pada permainan "Don't Starve".

DAFTAR PUSTAKA

- [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Program-Dinamis-\(2018\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Program-Dinamis-(2018).pdf)
- <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2017-2018/Makalah2017/Makalah-Matdis-2017-136.pdf>
- <https://www.klei.com/games/dont-starve>
- https://store.steampowered.com/app/219740/Dont_Starve/
- <https://dontstarve.gamepedia.com/Seasons>
- <http://dontstarve.wikia.com/wiki/Food>

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, 13 Mei 2018



Yoga Prasetyo 13515148