

Optimasi Pemilihan Tanaman Petani dengan Algoritma Program Dinamis

Hagai Raja Sinulingga - 13516136

Program Studi Teknik Informatika

Institut Teknologi Bandung

Bandung, Indonesia

hagairaja@gmail.com

Abstraksi—Indonesia adalah negara agraris yang dapat ditanami oleh begitu banyak jenis tanaman. Tidak heran, hal ini membuat banyak mata pencaharian rakyat Indonesia adalah melalui bertani. Untuk mengurangi resiko gagal panen dan memanfaatkan potensi tersebut, banyak petani yang kemudian memilih untuk menanam secara terdiversifikasi. Petani memilih untuk membagi-bagi ukuran lahannya dan kemudian ditanami dengan tanaman yang berbeda. Hal ini agar ketika satu tanaman gagal panen masih ada tanaman lain yang mungkin dapat berhasil dan menutupinya. Namun masalahnya adalah setiap tanaman memiliki modal, nilai jual dan lama waktu panen yang berbeda-beda satu sama lain. Oleh karena itu, dibutuhkan suatu cara untuk menentukan tanaman-tanaman apa yang menjadi pilihan tepat bagi petani untuk memaksimalkan keuntungan yang didapatnya. Pada makalah ini akan dibahas bagaimana memutuskan tanaman apa yang sebaiknya dipilih untuk mendapatkan keuntungan yang maksimal dalam rangka diversifikasi penanaman petani menggunakan algoritma program dinamis.

Kata Kunci—Petani, tanaman, diversifikasi, program dinamis

I. PENDAHULUAN

Indonesia adalah negara yang berada dalam garis khatulistiwa bersama dengan Brazil, Taiwan dan negara-negara tropis lainnya. Posisinya yang strategis membuat tanah Indonesia menjadi tanah yang sejak dahulu menjadi tempat bertumbuhnya berbagai makhluk hidup sehingga saat ini Indonesia memiliki tanah yang begitu subur dan kaya akan zat hara yang dibutuhkan tanaman. Hal ini membuat Indonesia menjadi tempat bertumbuhnya banyak jenis tanam-tanaman termasuk tanaman hijau yang dapat dimakan. Ditambah lagi dengan musim yang hanya dua yaitu musim kemarau dan musim hujan, Indonesia menjadi negara yang dapat ditumbuhi oleh tanaman di sepanjang tahunnya.

Tidak heran hal ini membuat banyak rakyat Indonesia akhirnya memilih bertani sebagai mata pencahariannya sehari-hari. Hasil konsensus pada tahun 2013 bahkan menunjukkan bahwa jumlah rakyat Indonesia yang memilih ladang pencaharian melalui sektor pertanian mencapai 32,7 jiwa[1]. Oleh karena itu Indonesia pantas disebut sebagai negara agraris dan memang sudah seharusnya demikian mengingat potensi alam yang begitu besar ini. Namun, masih banyak ketidak-idealan dalam mengolah potensi alam di Indonesia melalui pertanian ini. Dalam bertani, para petani harus melawan resiko

gagal panen dan menerima nilai harga jual hasil panen yang dapat naik-turun berdasarkan kondisi pasar. Hal inilah yang menyebabkan masih banyak petani Indonesia yang kurang merasakan kemakmuran dari hasil jerih payahnya.

Untuk mengurangi risiko kedua hal tersebut, beberapa petani sebenarnya sudah menerapkan metode penanaman terdiversifikasi yaitu menanam tanaman yang jenis berbeda dalam satu lahan. Hal ini sangat baik agar menjaga pendapatan petani dalam posisi aman ketika suatu harga tanaman turun tiba-tiba karena sudah ada tanaman lain yang siap menutupi kerugiannya. Selain itu, dengan metode ini petani juga membantu melestarikan kesuburan tanah karena setiap tanaman memiliki kebutuhan zat hara yang berbeda sehingga tanah tidak tereksplorasi hanya di satu jenis mineral saja. Apabila dilakukan dengan baik, metode ini harusnya dapat menjadi cara petani dapat membuat pertaniannya menjadi pertanian yang berkelanjutan (*sustainable*).

Namun dengan metode diversifikasi tersebut muncul sebuah permasalahan baru bagi petani. Pertanyaan mengenai tanaman apa saja yang harus ditanam harus dapat dijawab dan diperhatikan matang-matang karena menyangkut keuntungan dan keberhasilan metode ini. Dalam mempertimbangkannya, para petani diberikan batasan bahwa setiap tanaman memiliki harga modal, harga jual dan lama waktu panen. Ketiga hal ini menjadi faktor yang senantiasa berubah melihat kondisi pasar. Berdasarkan hal ini, petani kemudian dapat menentukan tanaman-tanaman apa yang sebaiknya dipilih untuk kemudian ditanam.

Makalah ini akan mengaplikasikan pemrograman dinamis (dynamic programming) yang merupakan salah satu topik mata kuliah “Strategi Algoritma” untuk menyelesaikan permasalahan penentuan tanaman-tanaman apa saja yang sebaiknya ditanam seorang petani yang hendak menerapkan metode diversifikasi. Masalah ini dapat dimodelkan menjadi permasalahan Integer (1/0) Knapsack yang memiliki beberapa batasan (constraint) yaitu modal, harga jual dan lama waktu panen sebuah tanaman. Melalui strategi algoritma ini diharapkan petani dapat dengan mudah menentukan kombinasi tanaman apa yang paling menguntungkannya untuk boleh ditanam di ladangnya kelak. Sehingga walaupun ketiga faktor ini berubah di setiap tahunnya, petani tidak perlu berpikir panjang akan langkah selanjutnya yang harus ia ambil.

II. DASAR TEORI

A. Pertanian

Pertanian merupakan salah satu sejarah kebudayaan manusia. Pertanian muncul ketika manusia mulai tinggal menetap di suatu lokasi dan tidak nomaden lagi.[2] Istilah pertanian berarti menanam suatu tanaman pada lahan tertentu untuk kemudian dapat diolah untuk dikonsumsi.

Ketika menyebutkan kata pertanian maka akan ada dua terminologi yang muncul yaitu petani dan bertani. Petani adalah orang yang melakukan pertanian itu sendiri (menanam tanaman) dan bertani adalah kegiatan menanam dan merawat tanaman itu sendiri.

Dalam bertani, para petani dapat memilih menanam tanaman dalam metode monokultur dan multikultur yang sering disebut juga terdiversifikasi. Monokultur berarti menanam satu jenis tanaman saja dalam satu lahan sedangkan multikultur berarti menanam beberapa jenis tanaman dalam satu lahan. Pada makalah ini kita akan berfokus pada metode multikultur.



Gambar 1 – Multikultur [3]

B. Integer (1/0) Knapsack Problem [4]

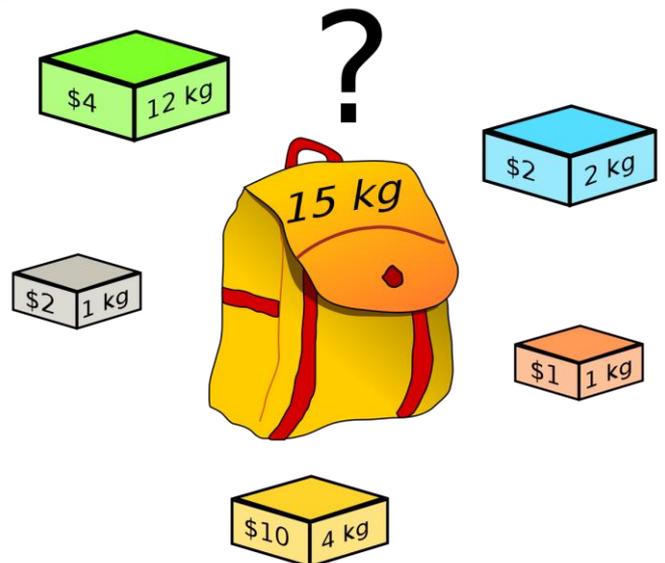
Integer (1/0) Knapsack Problem adalah permasalahan yang sangat familiar di dunia strategi algoritma. Permasalahannya adalah memilih barang-barang (X) dari n banyak barang yang masing-masing memiliki berat (W) tertentu dan keuntungan (V) tertentu sehingga dengan kapasitas k, kita mendapat keuntungan terbesar dengan memilih kombinasi barang tersebut. Jadi untuk memperjelas, setiap barang tersebut memiliki berat tertentu sehingga kita tidak dapat sembarang mengambil barangnya karena harus menyesuaikan dengan kapasitas berat yang dimiliki. Di sisi lain kita juga harus memaksimalkan keuntungan yang didapat melalui keuntungan yang dimiliki barang yang kita ambil. Berikut adalah bentuk matematis dari persoalan ini.

Maksimasi $\sum ViXi$ sampai n dari $i=1$
dengan batasan (constraint) $\sum WiXi \leq k$
dengan $Xi \in \{0, 1\}$
Solusi: $\{X1, X2, \dots, Xn\}$

Permasalahan ini dapat digunakan untuk memodelkan berbagai persoalan optimasi yang serupa seperti penyimpanan barang di Gudang dan pengaturan pengiriman barang seefisien mungkin untuk mendapatkan keuntungan terbesar. Jawaban dari persoalan ini akan berupa subset dari himpunan yang disebut sebagai barang-barang yang dinilai merupakan kombinasi paling menguntungkan namun tetap memenuhi batasan.

Dalam Teori komputasi, persoalan ini termasuk dalam persoalan NP-Complete. Artinya saat ini belum ditemukan sebuah algoritma yang dapat menyelesaikan persoalan ini dalam waktu polinomial. Walaupun demikian, masih ada setidaknya algoritma yang dapat menyelesaikan persoalan ini dalam waktu pseudo-polinomial yaitu dengan menggunakan pemrograman dinamis (*dynamic programming*). Selain dengan menggunakan pemrograman dinamis (*dynamic programming*), persoalan ini juga dapat diselesaikan dengan strategi algoritma yang lain seperti exhaustive search (mengenumerasi semua kemungkinan solusi dan melihat yang mana paling optimum), branch and bound, dan greedy (pendekatan dengan greedy tidak selalu menghasilkan solusi yang optimal namun lebih cepat untuk mencapai solusi).

Knapsack Problem juga memiliki beberapa variasi. Variasi ini muncul dengan mengubah parameter permasalahan seperti jumlah batasan, jumlah tujuan atau jumlah knapsack yang digunakan. Variasi tersebut diantaranya adalah Bounded Knapsack Problem. Setiap objek bias dipilih beberapa kali sejumlah c kali. Hal sebenarnya dapat dianggap dengan cara benda tersebut terdapat sebanyak c. Selanjutnya Unbounded Knapsack Problem yang tidak ada batas pemilihan untuk setiap objek. Ada juga Multi-objective Knapsack Problem yang tujuan/hal yang ingin dioptimasi lebih dari satu. Serta untuk multi yang lainnya terdapat Multi-dimensional Knapsack Problem yang batasan/constraint lebih dari satu dan Multiple Knapsack Problem yang jumlah knapsaknya lebih dari satu.



Gambar 2 – Knapsack Problem [5]

C. Pemrograman Dinamis (Dynamic Programming)

Pemrograman Dinamis adalah salah satu strategi pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan (stage) yang terpisah sedemikian sehingga solusi dapat dipandang sebagai serangkaian keputusan yang diambil secara berkaitan. Istilah ini pertama kali diperkenalkan oleh seorang professor dari Universitas Princeton yang juga bekerja di RAND corporation, bernama Richard Bellman, pada era tahun 1950-an. Ia mencetuskan hal ini karena perhitungan solusi dilakukan dengan menggunakan tabel-tabel yang isinya terisi secara dinamis. Pemrograman dinamis umumnya digunakan untuk menyelesaikan persoalan optimasi.

Strategi algoritma *Dynamic Programming* (DP) sebenarnya mirip dengan strategi *Greedy* yang membentuk solusi secara bertahap dan mengambil langkah terbaik untuk setiap fasenya. Hanya saja perbedaan yang sangat jelas pada DP ialah ia masih menyimpannya solusi terbaik pada setiap stage dan selalu mengujinya kembali ketika masuk ke stage berikutnya dan berikutnya hingga sampai ke stage akhir. Berbeda dengan strategi *greedy* yang benar-benar mengambil keputusan terbaik pada saat itu yang dianggap sebagai maksimum lokal yang harapannya menuju kepada jawaban optimum yang disebut sebagai maksimum global. Hal inilah yang sering kali membuat strategi *greedy* tidak memberikan solusi yang benar, termasuk dalam persoalan Knapsack ini.

Berbeda dengan *greedy* yang mencari optimum lokal, DP mencari optimum global dengan menjamin solusi tersebut optimum untuk setiap step stage. Jika dianggap terdapat n stage dan kita melangkah mulai dari stage satu dengan iterasi k . Maka dalam DP kita akan melangkah ke stage $k+1$ menggunakan apa yang sudah optimum pada stage k . Hasil optimum ketika melangkah setiap ke setiap stage disimpan dalam sebuah tabel (memoisasi) yang dapat dipakai ketika setiap kali melangkah menuju stage berikutnya. Sehingga dijamin ketika masuk ke dalam stage terakhir, semua kemungkinan optimal lokal sudah saling diadu dan didapat optimum global saja.

Pemrograman dinamis tidak dapat diterapkan pada setiap persoalan. Berikut adalah beberapa karakteristik persoalan yang dapat diselesaikan melalui DP.

1. Persoalan dapat dibagi menjadi beberapa tahap (*stage*) yang terbatas dan pada stage hanya akan dilakukan satu pengambilan keputusan.
2. Masing-masing tahap terdiri dari sejumlah status (*state*) yang berhubungan dengan tahap tersebut. Status merupakan kemungkinan masukan yang ada pada tahap tersebut.
3. Hasil dari keputusan yang diambil pada setiap tahap dikalkulasi dari keterkaitan antar status ke status berikutnya. Jadi ada serangkaian tahap yang saling terkoneksi antara setiap statusnya.
4. Harga (*Cost*) pada suatu tahap meningkat secara teratur (*steadily*) dengan bertambahnya jumlah tahapan.

5. Harga pada suatu tahap bergantung pada Harga pada tahap-tahap sebelumnya dan pada tahap tersebut untuk kemudian dikalkulasi menjadi harga tahap itu.
6. Keputusan optimum pada suatu tahap tidak memandang tahap dua sebelumnya melainkan hanya pada tahap satu sebelumnya.
7. Adanya hubungan rekursif antara setiap tahap dalam mengidentifikasi solusi optimum pada tahap tersebut.
8. Persoalan tersebut berbicara tentang pencarian solusi paling optimum dengan batasan-batasan tertentu.

Selain itu, terdapat pula dua jenis pendekatan yang dapat dilakukan untuk menyelesaikan persoalan melalui DP yaitu sebagai berikut. [5]

1. Pemrograman dinamis maju (*forward* atau *up-down*). Dengan pendekatan ini DP bergerak mengevaluasi maju dari tahap 1 lalu ke tahap 2, 3 dan seterusnya sampai ke tahap n yang merupakan tahap akhir yang sudah dapat ditentukan solusi optimumnya.
2. Pemrograman dinamis mundur (*Backward* atau *Bottom-up*). Merupakan kebalikan dari maju yaitu evaluasi dimulai dari tahap akhir, lalu mundur sampai menuju tahap awal. Hal ini ibarat isi tas kita sudah penuh kemudian dikosongkan satu per satu sambil mencatat efek kita mengeluarkan barang.

Kedua pendekatan ini menghasilkan solusi yang sama dan keduanya merupakan solusi optimum.

Pada umumnya, untuk mengembangkan sebuah algoritma pemecahan masalah melalui strategi DP dilakukan langkah berikut.

1. Tentukan struktur solusi optimum yang ingin dicapai
2. Definisikan secara rekursif nilai solusi optimum.
3. Pilih pendekatan untuk menghitung nilai solusi optimum yaitu secara maju atau mundur.
4. Konstruksi solusi optimum.

Pada implementasinya, pemrograman dinamis dapat diimplementasikan tidak hanya secara rekursif melainkan dapat pula dengan cara iteratif dengan memanfaatkan array multi dimensi untuk menyimpan hasil kalkulasi dengan nilai optimum pada setiap tahap. Solusi yang dibangun melalui pendekatan Maju biasanya diimplementasi secara rekursif sedangkan untuk solusi yang dibangun melalui pendekatan Mundur diimplementasi secara iteratif.

III. PENYELESAIAN MULTI-DIMENSIONAL KNAPSACK PROBLEM DENGAN MENGGUNAKAN PEMROGRAMAN DINAMIS

Untuk merancang algoritma memecahkan masalah ini kita akan mengikuti empat langkah yang telah dipaparkan sebelumnya. Pada persoalan ini dimisalkan tahap (k) adalah proses memasukkan objek ke- k ke dalam knapsack dan status (y) menyatakan setiap batasan yang ada. Jadi jika ada dua batasan dalam kasus multi-dimensional knapsacknya maka akan ada y_1 dan y_2 . Selanjutnya M merupakan kapasitas untuk masing-masing batasan yang diberikan. Jadi jika kondisinya

seperti yang tadi maka akan ada M1 dan M2. Terakhir akan dihasilkan solusi berbentuk (x_1, x_2, \dots, x_k) dimana x merupakan sebuah Boolean 1 atau 0.

Proses mencari solusi optimumnya dapat dilakukan dengan cara sebagai berikut. Misalkan kita berada tahap k yaitu mempertimbangkan apakah memasukkan objek ke- k atau tidak ke dalam knapsack. Untuk mencari solusi optimum pada tahap itu maka akan dipertimbangkan antara mengambilnya yaitu menambahkan laba dengan hasil optimum pada tahap sebelumnya dengan batasan sudah ditambah dengan batasan yang dibawa objek tersebut atau tidak mengambilnya yaitu tahap selanjutnya tetapi dengan batasan yang sama. Keuntungan yang paling besar kemudian akan dipilih menjadi solusi optimum untuk tahap itu.

Berikut adalah gambaran fungsi relasi rekurens yang dipakai dalam penyelesaian masalah soal ini.

$$F_0(y) = 0, y_i = 0, 1, 2, \dots, M_i$$

Untuk $1 \leq i \leq m$ (basis)

$$F_k(y) = -\infty, \text{ jika terdapat } y_i < 0$$

Untuk $1 \leq i \leq m$ (basis)

$$F_k(y) = \max \{F_{k-1}(y), P_k + F_{k-1}(y - W_k)\} \quad (\text{rekurens})$$

Untuk $k = 1, 2, \dots, n$ dan $y_i \geq 0$ untuk $1 \leq i \leq m$

$F_k(y)$ adalah keuntungan optimum untuk tahap ke- k untuk kapasitas y pada persoalan Multi-dimensional Knapsack Problem. Untuk nilai yang tidak ada pada list diberi nilai $-\infty$ untuk menghilangkannya dari pertimbangan. Demikianlah solusi optimum akan didapatkan. Algoritma dengan desain seperti ini memiliki kompleksitas $O(nM)$.

Berikut adalah contoh penerapan pemrograman dinamis untuk menyelesaikan persoalan Multi-dimensional Knapsack Problem. Misalkan batasan yang digunakan adalah kapasitas bobot dan volume yaitu masing-masing 2 dan 3 dengan jumlah objek yang ingin dimasukkan ada 3 dengan rincian sebagai berikut.

Tabel I - Rincian Barang Knapsack

Objek ke - i	Bobot (W_1)	Volume (W_2)	Nilai (P_i)
1	1	2	30
2	2	2	45
3	1	1	25

Berikut adalah proses perhitungan untuk setiap tahapnya

1. Tahap I:

$$F_1(y) = \max \{F_0(y_1, y_2), P_1 + F_0(y_1 - W_1, y_2 - W_2)\}$$

$$= \max \{F_0(y_1, y_2), 30 + F_0(y_1 - W_1, y_2 - W_2)\}$$

Tabel II - Tahap I

y_1	y_2	$F_0(y_1, y_2)$	$30 + F_0(y_1 - W_1, y_2 - W_2)$	$F_1(y_1, y_2)$	(x_1, x_2, x_3)
-------	-------	-----------------	----------------------------------	-----------------	-------------------

0	0	0	$-\infty$	0	(0,0,0)
0	1	0	$-\infty$	0	(0,0,0)
0	2	0	$-\infty$	0	(0,0,0)
0	3	0	$-\infty$	0	(0,0,0)
1	0	0	$-\infty$	0	(0,0,0)
1	1	0	$-\infty$	0	(0,0,0)
1	2	0	30	30	(1,0,0)
1	3	0	30	30	(1,0,0)
2	0	0	$-\infty$	0	(0,0,0)
2	1	0	$-\infty$	0	(0,0,0)
2	2	0	30	30	(1,0,0)
2	3	0	30	30	(1,0,0)

2. Tahap II:

$$F_2(y) = \max \{F_1(y_1, y_2), P_2 + F_1(y_1 - W_1, y_2 - W_2)\}$$

$$= \max \{F_1(y_1, y_2), 45 + F_1(y_1 - W_1, y_2 - W_2)\}$$

y_1	y_2	$F_1(y_1, y_2)$	$30 + F_1(y_1 - W_1, y_2 - W_2)$	$F_2(y_1, y_2)$	(x_1, x_2, x_3)
0	0	0	$45 - \infty = -\infty$	0	(0,0,0)
0	1	0	$45 - \infty = -\infty$	0	(0,0,0)
0	2	0	$45 - \infty = -\infty$	0	(0,0,0)
0	3	0	$45 - \infty = -\infty$	0	(0,0,0)
1	0	0	$45 - \infty = -\infty$	0	(0,0,0)
1	1	0	$45 - \infty = -\infty$	0	(0,0,0)
1	2	30	$45 - \infty = -\infty$	30	(1,0,0)
1	3	30	$45 + 0 = 45$	45	(0,1,0)
2	0	0	$45 - \infty = -\infty$	0	(0,0,0)
2	1	0	$45 - \infty = -\infty$	0	(0,0,0)
2	2	30	$45 + 0 = 45$	45	(0,1,0)
2	3	30	$45 + 0 = 45$	45	(0,1,0)

3. Tahap III :

$$F_3(y) = \max \{F_2(y_1, y_2), P_3 + F_2(y_1 - W_1, y_2 - W_2)\}$$

$$= \max \{F_2(y_1, y_2), 25 + F_2(y_1 - W_1, y_2 - W_2)\}$$

y_1	y_2	$F_2(y_1, y_2)$	$25 + F_2(y_1 - W_1, y_2 - W_2)$	$F_3(y_1, y_2)$	(x_1, x_2, x_3)
0	0	0	$25 - \infty = -\infty$	0	(0,0,0)
0	1	0	$25 - \infty = -\infty$	0	(0,0,0)
0	2	0	$25 - \infty = -\infty$	0	(0,0,0)

0	3	0	$25 - \infty = -\infty$	0	(0,0,0)
1	0	0	$25 - \infty = -\infty$	0	(0,0,0)
1	1	0	$25 + 0 = 25$	25	(0,0,1)
1	2	30	$25 + 0 = 25$	30	(1,0,0)
1	3	45	$25 + 0 = 25$	45	(0,1,0)
2	0	0	$25 - \infty = -\infty$	0	(0,0,0)
2	1	0	$25 + 0 = 25$	25	(0,0,1)
2	2	45	$25 + 0 = 25$	45	(0,1,0)
2	3	45	$25 + 30 = 55$	55	(1,0,1)

Solusi optimum dari persoalan ini adalah $X = (1, 0, 1)$ dengan $\sum p = f_3(2, 3) = 55$.

IV. PENERAPAN PEMROGRAMAN DINAMIS UNTUK MENENTUKAN PILIHAN TANAMAN

Misalkan suatu pasar di tahun 2018 ini memiliki keadaan harga barang-barangnya seperti berikut.

Tanaman	Modal (Juta Rp)	Harga (Juta Rp)	Jual	Lama Panen (bulan)
Wortel	5	7	3	
Cabe	20	25	5	
Selada	3	5	2	
Kol	4	8	2	
Bayam	2	3	1	
Kangkung	4	5	1	
Tomat	7	12	4	
Bawang	12	18	6	
Jagung	5	8	3	
Ubi	4	7	2	
Kentang	6	12	2	

Maka untuk memulai tahun tersebut, petani dapat mengecek apa yang sebaiknya dipilih untuk ditanam di tahun tersebut menggunakan teknik yang sama dengan yang dipakai untuk persoalan Multi-dimensional Knapsack Problem. Dalam kasus ini W_1 adalah Modal yang bergantung kondisi ekonomi petani, W_2 adalah Lama Panen yaitu 12 bulan dan P adalah jumlah Harga Jual.

Untuk memudahkan petani, berikut adalah contoh kode program dalam bahasa Python yang dapat dieksekusi untuk menjawab persoalan ini. (Diolah dari GeekForGeeks^[6])

```
def knapSack(W1, W2, W1t, W2t, sol, n, val):
    # Base Case
    if n == 0 or W1 == 0 or W2 == 0 :
        return 0, sol
```

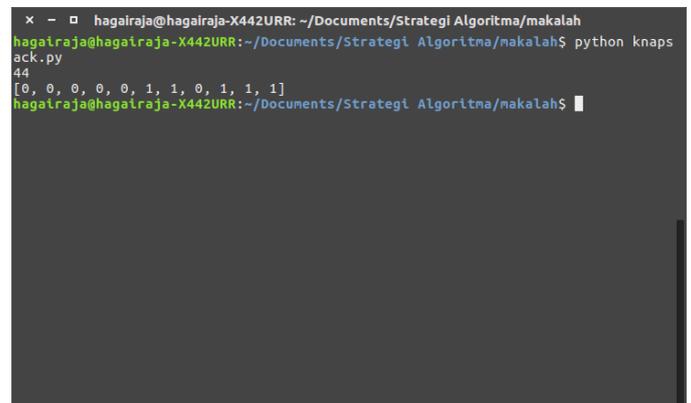
```
# If weight of the nth item is more than the budget
# W1 and W2, then this plant cannot be included in the optimal solution
if W1t[n-1] > W1 or W2t[n-1] > W2 :
    return knapSack(W1, W2, W1t, W2t, sol, n-1, val)

# return the maximum of two cases:
# (1) nth plant included
# (2) not included
else:
    num1, sol1 = val[n-1] + knapSack(W1-W1t[n-1], W2-W2t[n-1], W1t, W2t, sol, n-1, val)
    num2, sol2 = knapSack(W1, W2, W1t, W2t, sol, n-1, val)
    if num1 > num2 :
        sol1[n] = 1
        return num1, sol1
    else :
        return num2, sol2
```

Serta cara pemanggilannya untuk data diatas dengan Modal 40 Juta Rupiah dapat dilakukan dengan cara sebagai berikut.

```
val = [7, 25, 5, 8, 3, 5, 12, 18, 8, 7, 12]
W1t = [5, 20, 3, 4, 2, 4, 7, 12, 5, 4, 6]
W2t = [3, 5, 2, 2, 1, 1, 4, 6, 3, 2, 2]
W1 = 40
W2 = 12
n = len(val)
sol = [0,0,0,0,0,0,0,0,0,0,0]
print knapSack(W1, W2, W1t, W2t, sol, n, val)
```

Hasilnya akan berupa tampilan berikut



Gambar 3 – Hasil Eksperimen 1

Pada hasil percobaan ini ditunjukkan bahwa petani dapat meraih harga jual sebesar 44 juta rupiah yaitu dengan menanam Kangkung, Tomat, Jagung, Ubi dan Kentang. Hasil tersebut terlihat dari urutan array Boolean yang dihasilkan yang

1 artinya tanam dan 0 artinya tidak tanam. Setelah menghitung batasan, ternyata benar masih memenuhi batasan yaitu Modal 26 dari 40 dan Waktu 12 dari 12. Disini diperoleh benar jawaban optimum dari pilihan yang harus diambil petani untuk kasus demikian.

Saya juga mencoba kasus kedua dengan berasumsi beberapa tanaman memiliki perubahan harga jual dan beberapa bibit juga memiliki kemampuan panen yang berbeda di tahun ini. Hal itu semua dituliskan dalam tabel berikut.

Tanaman	Modal (Juta Rp)	Harga Jual (Juta Rp)	Lama Panen (bulan)
Wortel	5	7	3
Cabe	20	25	5
Selada	3	5	2
Kol	4	8	2
Bayam	2	3	1
Kangkung	4	5	1
Tomat	7	12	4
Bawang	12	18	6
Jagung	5	8	2
Ubi	4	7	2
Kentang	6	12	2

Dengan batasan Modal sebesar 30 juta rupiah dan waktu yang sama yaitu 12 bulan didapat hasil sebagai berikut.

```

x - □ hagairaja@hagairaja-X442URR: ~/Documents/Strategi Algoritma/makalah
hagairaja@hagairaja-X442URR:~/Documents/Strategi Algoritma/makalah$ python knapsack.py
46
[1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1]
hagairaja@hagairaja-X442URR:~/Documents/Strategi Algoritma/makalah$

```

Gambar 3 – Hasil Eksperimen 2

Pada hasil percobaan ini ditunjukkan bahwa petani dapat meraih harga jual sebesar 46 juta rupiah yaitu dengan menanam Wortel, Kol, Bayam, Jagung, Ubi dan Kentang. Setelah menghitung batasan, ternyata benar masih memenuhi batasan yaitu Modal 26 dari 30 dan Waktu 12 dari 12. Disini diperoleh benar jawaban optimum dari pilihan yang harus diambil petani untuk kasus demikian.

Dalam proses eksekusi mencari jawaban paling optimum, komputer hanya membutuhkan waktu kurang dari satu detik.

Hal ini sangat baik dan wajar untuk membantu petani mengambil keputusan tanaman apa yang akan ia tanam pada tahun itu.

V. KESIMPULAN DAN SARAN

Algoritma pemrograman dinamis dapat dipakai untuk menyelesaikan persoalan optimasi dalam memilih kombinasi tanaman-tanaman yang baik untuk ditanam seorang petani pada suatu tahun. Hal ini dapat membantu petani untuk dengan cepat dan tepat mengambil keputusan dan menerima laba terbesar sebagai rezekinya di tahun itu. Persoalan ini sangat baik didekati dengan metode pemecahan masalah Integer (0/1) Multi-dimensional Knapsack Problem dimana setiap batasan diubah menjadi batasan tertentu oleh tanaman seperti dalam makalah ini ialah modal dan lama panen. Melalui pendekatan ini solusi dapat dicari dalam waktu yang singkat bahkan kurang dari satu detik.

Selanjutnya saya menyarankan untuk melakukan riset lanjutan akan hal ini. Saya secara khusus menyarankan untuk membuat GUI program ini agar benar-benar dapat digunakan langsung oleh petani atau camatnya agar kelak lebih mudah mereka pakai dan pertimbangkan. Selain itu dapat pula ditambahkan batasan-batasan lain untuk membuat hasil diversifikasi yang lebih baik. Hal ini tentu saja dengan mudah dapat dilakukan yaitu dengan menambahkan satu buah status baru (y).

VI. UCAPAN TERIMA KASIH

Pertama dan terutama penulis ingin mengucapkan puji syukur kepada Tuhan Yang Maha Esa karena dengan rahmat dan karunai-Nya penulis dapat menyelesaikan makalah dengan judul “Optimasi Pemilihan Tanaman Petani dengan Algoritma Program Dinamis” ini dengan baik. Penulis juga berterima kasih kepada para dosen pengajar mata kuliah IF2211 Strategi Algoritma, Bapak Dr. Ir. Rinaldi Munir, Ibu M.T., Masayu Leylia Khodra, S.T., M.T., dan Ibu Dr. Nur Ulfa Maulidevi, S.T., M. Sc., karena atas bimbingan mereka selama ini dalam mengajar dan memberikan ilmu sehingga penulis mampu membuat makalah ini. Penulis juga berterima kasih kepada rekan-rekan yang telah memberikan semangat dan dorongan kepada penulis maupun para asisten kuliah yang dengan sepenuh hati membentuk saya dan teman-teman sekelas untuk setia mengoprek di minggu-minggu kuliah kami. Kiranya kita semua hidup dalam damai sejahtera.

REFERENSI

- [1] <http://st2013.bps.go.id/dev/st2013/index.php/site/tabel?tid=23&wid=0>, diakses 13 Mei 2018 pukul 14:31 WIB
- [2] <http://himateta.lk.ipb.ac.id/2010/07/sejarah-pertanian/>, diakses 13 Mei 2018 pukul 15:31 WIB
- [3] <https://gardenerdy.com/difference-between-monoculture-polyculture-farming-techniques>, diakses 13 Mei 2018 pukul 15:21 WIB
- [4] Kellerer, et al., "Knapsacks Problems", Springer, 2004.
- [5] R. Munir, Program Dinamis (Dynamic Programming). 2015. [Online] Tersedia dalam [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2014-2015/Program%20Dinamis%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2014-2015/Program%20Dinamis%20(2015).pdf). [diakses 13 Mei 2018 pukul 16.30 WIB].
- [6] <https://www.geeksforgeeks.org/knapsack-problem/>, diakses 13 Mei 2018 pukul 23:21 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, 13 Mei 2018



Hagai Raja Sinulingga
13516136