

# Optimasi Pemilihan Barang yang Dibawa Traveling dengan Algoritma Program Dinamis

Renita Napitupulu - 13517134  
Program Studi Teknik Informatika  
Institut Teknologi Bandung  
Bandung, Indonesia  
renitairen1613@gmail.com

**Abstraksi**— Traveling sudah menjadi hal yang lumrah bagi masyarakat terutama di kalangan anak muda. Saat akan melakukan traveling, pelancong pastinya membutuhkan barang yang akan dipakai selama traveling. Namun, seringkali para pelancong kesulitan menentukan barang apa saja yang harus dibawa selama traveling. Hal ini dapat menimbulkan permasalahan yang baru akibat kelupaan membawa barang ataupun membawa lebih banyak jenis barang. Barang yang dibawa juga memiliki beberapa batasan yaitu berat, volume dari barang tersebut, dan juga prioritas. Namun ada juga faktor lain, seperti musim yang akan mempengaruhi daftar barang yang dihasilkan. Untuk itu, dibutuhkan sebuah metode untuk mengoptimalkan prioritas dari barang yang dibawa. Pada makalah ini akan dibahas bagaimana memutuskan kombinasi barang apa saja yang akan dibawa pelancong untuk mengoptimalkan prioritas dari barang tersebut dengan menggunakan algoritma program dinamis.

**Kata Kunci**—Traveling, pelancong, barang, program dinamis

## I. PENDAHULUAN

Traveling merupakan kegiatan yang sudah dijadikan lumrah bahkan dijadikan menjadi kewajiban yang sudah dijadwalkan untuk sebagian masyarakat Indonesia terutama di kalangan anak muda yang sering juga dikatakan generasi milenial. Mereka yang suka melakukan kegiatan traveling disebut juga pelancong biasanya sudah mencari-cari tentang informasi destinasi wisata yang akan dituju dan juga keperluan apa saja yang dibutuhkan jauh hari sebelum mereka melakukan traveling. Persiapan yang perlu dilakukan mulai dari bagaimana agar bisa sampai ke destinasi yang dituju, barang apa saja yang dibutuhkan dalam rentang waktu itu, dan juga beberapa hal lainnya.

Melakukan persiapan membutuhkan waktu yang sebenarnya tergantung dengan pribadi orang yang bersangkutan. Namun, secara umum dalam hal untuk memilih barang apa saja yang akan dibawa, cenderung terburu-buru dan juga bingung untuk menentukannya. Hal ini dapat menimbulkan masalah baru yaitu bisa saja barang yang dibawa pada akhirnya jumlahnya melebihi apa yang sudah diperkirakan di awal yang juga mengakibatkan biaya perjalanan yang juga ikut naik atau juga malah sebaliknya, para pelancong lupa membawa barang tertentu yang juga dapat menimbulkan masalah baru yaitu saat traveling, pelancong tersebut harus

membeli barang yang tadi lupa dibawa sehingga juga akan mengakibatkan biaya yang dikeluarkan melebihi yang sebelumnya sudah ditentukan.

Untuk mengurangi banyak permasalahan yang tadi sudah dipaparkan di atas, pelancong seharusnya sudah memiliki daftar prioritas dari barang apa saja yang dibutuhkan selama traveling. Daftar prioritas dari barang yang dibutuhkan akan tergantung pada pelancong, namun dapat digeneralkan berdasarkan kebutuhan dasar manusia untuk melakukan kegiatan sehari-hari.

Setelah memiliki daftar prioritas maka akan lebih mudah memilih kombinasi barang apa saja yang akan dipilih untuk dibawa selama traveling. Kombinasi yang dipilih akan berkaitan dengan batasan dari si barang itu sendiri yaitu volume dari barang yang akan di pilih. Terdapat volume maksimum yang telah ditentukan sebelumnya oleh si pelancong, sehingga tidak perlu takut bahwa barang yang akan dibawa akan berukuran lebih besar dari ukuran tas yang dipersiapkan ataupun hal lainnya..

Namun, belum cukup jika hanya mempertimbangkan hal itu saja. Mungkin saja kombinasi barang yang dipilih, beratnya melebihi bobot tas yang sudah dipersiapkan. Untuk mengatasi hal tersebut maka perlu ada juga batasan dengan berat dari barang yang akan dipilih. Yaitu akan disesuaikan dengan keinginan atupun standar yang sudah ditetapkan oleh pelancong sebelumnya. Untuk itu, dapat dikatakan bahwa terdapat beberapa faktor yang akan menentukan kombinasi barang yang dihasilkan yang akan dibawa saat traveling yaitu berat dari barang, volume dari barang tersebut, dan prioritas barang tersebut.

Makalah ini akan mengaplikasikan salah satu bahan ajar dalam kuliah Strategi Algoritma yaitu program dinamis (*dynamic programming*) yang akan memberikan solusi untuk permasalahan pemilihan barang yang akan dibawa saat traveling. Pemodelan yang dipilih untuk permasalahan ini yaitu Integer (1/0) Knapsack yang memiliki beberapa batasan yaitu berat barang, volume barang, dan juga prioritas dari barang tersebut. Melalui makalah ini, diharapkan pelancong dapat menentukan kombinasi dari barang yang akan dibawa saat traveling sehingga tidak perlu lagi berhadapan dengan permasalahan berat total barang yang dibawa melebihi

kuantitas yang sudah ditentukan sebekumnya ataupun beberapa masalah yang telah disebutkan di atas.

## II. DASAR TEORI

### A. Traveling

Traveling merupakan kegiatan yang semakin sering dijumpai saat ini. Traveling dalam bahasa Indonesia dapat diartikan sebagai perjalanan seseorang ke tempat yang berbeda atau belum pernah dikunjungi sebelumnya. Orang yang melakukan traveling seringkali disebut dengan kata *traveler* atau jika dalam Bahasa Indonesia disebut sebagai *pelancong*.



Gambar 1 Pelancong sedang traveling

(Sumber: [https://www.google.com/search?q=traveling&safe=strict&source=lnms&tbn=isch&sa=X&ved=0ahUKewi8vfjEq-zhAhVKsi8KHd4pDUkQ\\_AUIDigB&biw=801&bih=765#imgdii=xGRWRRfmzohKTM:&imgrc=LQIDFIB15pRJ9M](https://www.google.com/search?q=traveling&safe=strict&source=lnms&tbn=isch&sa=X&ved=0ahUKewi8vfjEq-zhAhVKsi8KHd4pDUkQ_AUIDigB&biw=801&bih=765#imgdii=xGRWRRfmzohKTM:&imgrc=LQIDFIB15pRJ9M): diakses pada 25 April pukul 10.00)

Destinasi yang dituju saat melakukan traveling dapat dibedakan menjadi beberapa tipe yaitu pantai, gunung, bangunan bersejarah, dan juga bangunan modern. Untuk setiap destinasi, memiliki kombinasi barang tersendiri yang memang sesuai untuk dibawa oleh pelancong.

### B. Integer (1/0) Knapsack Problem

*Knapsack* merupakan tas atau karung yang berfungsi untuk menampung sesuatu. Namun, tas ini memiliki limit (batasan) sehingga hanya beberapa objek yang dapat disimpan dalam tas tersebut. *Knapsack* merupakan permasalahan bagaimana memilih objek yang sesuai dari beberapa objek yang ditawarkan yang memiliki berat dan profit, sehingga total dari berat objek yang dipilih harus lebih kecil atau sama dengan kapasitas dari penampung (tas/karung) tersebut.

*Knapsack* memiliki tiga jenis persoalan yaitu *Integer (1/0) Knapsack Problem*, *Knapsack Bounded*, dan *Knapsack*

*Unbounded*. Permasalahan *Integer Knapsack* adalah permasalahan pemilihan objek dari beberapa objek yang ditawarkan dengan setiap objek hanya memiliki dua status yaitu terpilih atau tidak terpilih.

Misalkan *knapsack* yang diberikan memiliki bobot maksimum sebesar  $K$ . Diperlukan melakukan maksimasi dari profit yang akan diperoleh dari objek yang dipilih. Setiap barang akan memiliki berat yaitu  $w_i$  dan juga profit yang dihasilkan yaitu  $p_i$ . Maksimasi yang dihasilkan yaitu :

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

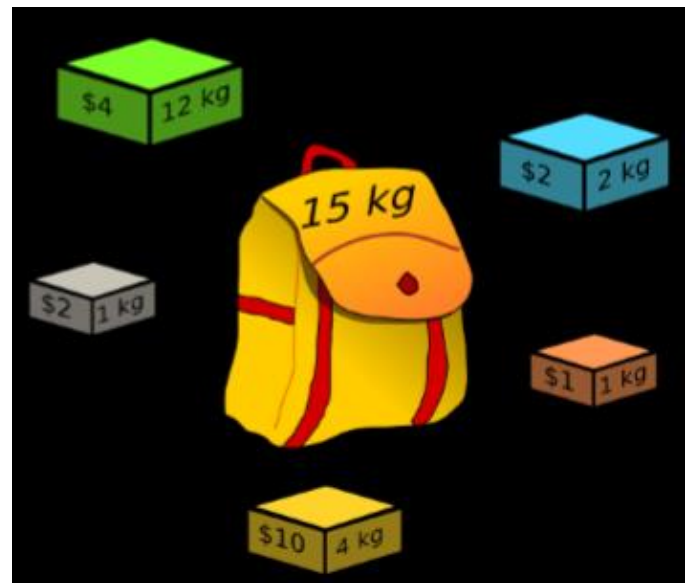
dan memiliki batasan

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

Solusi dari permasalahan tersebut yaitu dinyatakan dalam bentuk :

$$X = \{x_1, x_2, \dots, x_n\}$$

$x_i$  akan menghasilkan nilai 1 jika objek ke- $i$  dipilih dan sebaliknya yaitu bernilai 0 jika tidak dipilih.



Gambar 2 Integer (1/0) Knapsack

(Sumber: [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/Algoritma-Greedy-\(2019\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/Algoritma-Greedy-(2019).pdf) diakses pada 25 April 2019 pukul 10.30)

### C. Program Dinamis (Dynamic Programming)

Program dinamis merupakan merupakan sebuah metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan (stage) sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan.

Karakteristik dari persoalan yang dapat dapat diselesaikan dengan Program Dinamis, yaitu:

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. Cost pada suatu tahap bergantung pada cost tahap-tahap yang sebelumnya sudah dijalankan dan juga cost pada tahap tersebut.
6. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan pada tahap sebelumnya.
7. Terdapat hubungan rekursif antara setiap tahap yang akan menghasilkan keputusan terbaik untuk setiap status pada tahap tersebut.
8. Prinsip optimalitas berlaku pada persoalan tersebut.

Algoritma Greedy sebenarnya memiliki kemiripan dengan Program Dinamis karena sama-sama mencari solusi yang optimal dari persoalan yang diberikan. Perbedaannya terletak pada bagian yaitu yang dipertimbangkan selalu adalah optimal lokal hingga akhirnya menghasilkan optimum global. Jadi setelah melewati satu tahap, optimum local sebelumnya tidak dipertimbangkan lagi, meskipun mungkin saja mala nilai akhir optimum local yang sebelumnya yang lebih besar.

Sedangkan untuk program dinamis, optimum global yang dihasilkan itu diperoleh dari optimum yang selalu disimpan dari setiap tahap dan selalu akan dicek kembali, nilai mana yang memang akan menghasilkan nilai lebih optimum. Hal ini ditunjukkan dengan adanya prinsip optimalitas pada program dinamis bertujuan untuk menghasilkan rangkaian keputusan yang optimal. Prinsip optimalitas berarti jika solusi total optimal, maka bagian solusi sampai tahap ke-k juga optimal.

Pada program dinamis, terdapat dua pendekatan yang dapat digunakan yaitu Pohon Dinamis maju (*forward* atau *up-down*) dan juga Pohon Dinamis mundur (*backward* atau *bottom-up*). Misalkan dinyatakan perubahan keputusan yang harus dibuat untuk setiap tahapan dari 1 sampai n yaitu  $x_1, x_2, \dots, x_n$ . Untuk Pohon Dinamis maju, Program Dinamis akan bergerak mulai dari tahap 1, terus maju ke tahap 2,3, dan seterusnya sampai tahap n. Sehingga dikatakan bahwa evaluasi yang dikerjakan secara berurutan maju dari peubah keputusan  $x_1$  sampai dengan  $x_n$ . Biasanya implementasi untuk Program Dinamis maju akan dibangun secara rekursif. Sedangkan untuk Program Dinamis mundur, Program Dinamis akan bergerak mulai tahap n(tahap akhir) dan akan terus mundur ke tahap n-1, n-2, dan smapi akhirnya mencaapai tahap 1. Sehingga dikatakan bahwa evaluasi yang dikerjakan secara berurutan mundur dari peubah keputusan  $x_n$  sampai dengan  $x_1$ . Dan untuk Program Dinamis mundur

biasanya akan dibangun secara iteratif. Sehingga implementasi dari Program Dinamis tidak selalu harus bersifat rekursif.

Untuk menyelesaikan suatu permasalahan menggunakan Program Dinamis, langkah-langkah pengembangan yang perlu dilakukan yaitu:

1. Pilih karakteristik dari solusi optimal yang akan dihasilkan.
2. Definisikan secara rekursif nilai solusi optimal.
3. Tentukan pendekatan yang akan digunakan untuk menghitung nilai solusi optimal, apakah dengan pendekatan Program Dinamis secara maju atau oendekatan Program Dinamis secara mundur.
4. Lakukan konstruksi solusi optimum.

### III. PENYELESAIAN MULTI-DIMENSIONAL KNAPSACK PROBLEM DENGAN MENGGUNAKAN ALGORITMA PROGRAM DINAMIS

Dalam proses perancangan sebuah algoritma Program Dinamis untuk menyelesaikan sebuah masalah, langkah-langkah yang perlu dilakukan yaitu seperti yang sudah dijelaskan sebelumnya pada dasar teori. Pada persoalan ini, tahap (k) merupakan proses memasukkan barang ke dalam karung (*knapsack*). Selanjutnya yaitu Status (y) yang akan menyatakan batasan (*constrain*) dari permasalahan yang diberikan. Midalnya batasan yang diberikan yaitu M yang menunjukkan total beban maksimum yang dapat ditampung oleh *knapsack*. Solusi yang akan ditampilkan yaitu dalam bentuk  $(x_1, x_2, \dots, x_n)$  yang akan menghasilkan nilai 0 ataupun 1. 0 berarti barang tersebut tidak akan dimasukkan k etas sedangkan 1 berarti barang tersebut akan dimasukkan k etas.

Proses pencarian solusi yang optimum yaitu dilakukan dengan cara saat berada dalam tahapan k, maka akan dipertimbangkan saat barang tersebut diambil atau tidak dengan cara yaitu menjumlahkan total profit yang akan diterima dengan profit dari solusi optimum sebelumnya dan juga dengan tetap membah batasan yang ada dan harus divalidasi terlebih dahulu apakah masih memenuhi batasan ata sudah melanggar batasan. Jika masih sesuai dengan batasan, maka akan dibandingkan lebih besar profit mana yang diperoleh, apakah dengan mengambil barang tersebut atau tidak mengambilnya.

Untuk fungsi relasi rekurens yang akan digunakan dalam pencarian solusi untuk permasalahan ini dapat digambarkan sebagai berikut.

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

$$\text{Untuk } 1 \leq i \leq m \text{ (basis)}$$

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

$$\text{Untuk } 1 \leq i \leq m \text{ (basis)}$$

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

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

Dari fungsi tersebut, profit maksimal yang dapat diperoleh untuk tahap-k dengan kapasitas y pada persoalan 0/1 Knapsack

digambarkan dengan fungsi  $F_k(y)$  sedangkan  $F_0(y) = 0$  adalah nilai dari persoalan knapsack kosong (tidak ada persoalan knapsack) dengan kapasitas  $y$  dan yang terakhir yaitu  $F_k(y) = -\infty$  adalah nilai dari persoalan knapsack untuk kapasitas negatif. Solusi optimum dari persoalan 0/1 Knapsack adalah  $F_n(M)$ .

Contoh persoalan yang dapat diselesaikan dengan algoritma Program Dinamis diberikan sebagai berikut.

Barang ke-i	$w_i$	$p_i$
1	2	65
2	3	80
3	1	30

Batasan yang diberikan yaitu beart maksimum barang yang dapat diambil yaitu  $M=5$ . Proses penyelesaiannya adalah sebagai berikut.

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	Solusi Optimum			
	$F_0(y)$	$65 + f_0(y - 2)$	$f_1(y)$	$(x_1^*, x_2^*, x_3^*)$
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)

Tahap 2:

$$f_2(y) = \max\{f_1(y), p_1 + f_1(y - w_2)\}$$

$$= \max\{f_1(y), 70 + f_1(y - 3)\}$$

y	Solusi Optimum			
	$f_1(y)$	$85 + f_0(y - 2)$	$f_2(y)$	$(x_1^*, x_2^*, x_3^*)$
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)

Untuk tahap 3, hanya mengikuti sebagaimana di tahap 1 dan juga tahap 2. Solusi optimum yang diperoleh yaitu  $X = (1,1,0)$  dengan total profit yang dihasilkan yaitu  $f = 145$

#### IV. IMPLEMENTASI PROGRAM DINAMIS UNTUK MENENTUKAN PILIHAN BARANG YANG DIBAWA TRAVELING

Misalnya Renita merupakan seorang pelancong yang telah merencanakan untuk melakukan traveling akhir Mei ini. Renita berencana bahwa ia dan teman-temannya akan melakukan pendakian ke salah satu gunung di Jawa Barat. Dan tercatat bahwa pada bulan Mei yang masih masuk ke dalam musim

hujan sehingga cuaca yang diperkirakan yaitu akan terkadang mendung bahkan hujan.

Daftar barang-barang yang dimiliki dan sekiranya dibutuhkan oleh Renita untuk pendakian ini telah disesuaikan dengan tipe destinasi yang ingin dituju oleh pelancong yaitu ke gunung dan juga musim saat melakukan traveling yaitu pada saat musim hujan. Daftar barangnya adalah seperti berikut.

Nama Barang	Berat (kg)	Volume (l)	Prioritas
Jaket	3	4	1
Kaos	2	2	1
Celana	4	5	1
Pakaian dalam	1	3	1
Sarung Tangan + syal	0.5	2	0.8
Masker	0.1	0.5	0.7
Handuk	0.5	7	0.5
Jas hujan	1	4	1
Obat	0.3	1	1
Korek api	0.1	1	0.7
Senter anti air	1	1	1
Tissue	1	0.5	0.6
Sleeping bag	6	6	1
Trashbag	0.2	0.2	0.7
Drybag	4	4	0.6
Peralatan mandi	2	3	0.3
Tongkat gunung	1	2	0.3

Renita telah mempersiapkan tas carrier volume 28 l yang dapat menampung maksimum 20 kg barang.

Dalam kasus ini,  $W_1$  akan merepresentasikan berat dari barang yang akan diambil,  $W_2$  akan merepresentasikan volume dari barang yang terkait, sedangkan  $P$  akan merepresentasikan prioritas total barang yang diambil. Untuk menyelesaikan persoalan ini, dapat menggunakan kode program di bawah ini yang ditulis dengan menggunakan bahasa Python.

```

def knapSack(L_W1, L_W2, W1, W2, P, solusi, n):
    #Untuk kasus kosong atau bobot total = 0,
    atau volume total = 0
    if n == 0 or W1 == 0 or W2 == 0:
        return 0, solusi

    #untuk kasus berat dari barang ke-n <= berat total
    #perlu dibandingkan adan dipilih antara
    barang diambil atau tidak yang menghasilkan
    prioritas maksimum

    elif L_W1[n-1] <= W1 or L_W2[n-1] <= W2:
        prio1, solusi1 = P[n-1] + knapSack(L_W1, L_W2,
        W1-L_W1[n-1], W2-L_W2[n-1], P, solusi, n-1)
        prio2, solusi2 = knapSack(L_W1, L_W2, W1, W2,
        P, solusi, n-1)

        #ambil barang ke-n jika menghasilkan
        prio yang maksimum
        if prio1 > prio2:
            solusi1[n] = 1
            return prio1, solusi1
        else:
            return prio2, solusi2

    #untuk kasus berat dari barang ke-n > berat total
    #Barang ke-n tidak akan diambil dan lakukan
    knapSack kembali dengan menghapus barang ke-n
    else:
        return knapSack(L_W1, L_W2, W1, W2, P, solusi, n-1)

```

Untuk pemanggilan kode program yang telah diberikan di atas dapat dilakukan sebagai berikut.

```

L_W1 = [3, 2, 4, 1, 0.5, 0.1, 0.5, 1, 0.3, 0.1, 1, 1, 6, 0.2, 4, 2, 1]
L_W2 = [4, 2, 5, 3, 2, 0.5, 7, 4, 1, 1, 1, 0.5, 6, 0.2, 4, 3, 2]
P = [1, 1, 1, 1, 0.8, 0.7, 0.5, 1, 1, 0.7, 1, 0.6, 1, 0.7, 0.6, 0.3, 0.3]
W1 = 20
W2 = 28
n = len(L_W1)
solusi = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
print(knapSack(L_W1, L_W2, W1, W2, P, solusi, n))

```

Tampilan yang dihasilkan pada terminal adalah sebagai berikut:

```

E:\KULIAH\semester_4\STIMA>python knapSack.py
10.1
[1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0]
E:\KULIAH\semester_4\STIMA>

```

**Gambar 3** Hasil Eksperimen 1

Dari tampilan tersebut jika pada barang ke-i, nilai yang dihasilkan adalah 1, maka artinya barang ke-i akan dibawa saat traveling dari list barang yang sudah ditentukan. Dan sebaliknya, jika bernilai 0, maka barang tersebut tidak akan dibawa. Terlihat bahwa dari barang-barang yang sudah didaftarkan pada daftar barang yang dibuat oleh Renita, dia dapat memaksimalkan prioritas dari barang yang akan dibawa

saat traveling yaitu prioritas yang diperoleh sebesar 10,1 dan total berat maupun volume dari semua barang yang dibawa tidak melanggar batasan yang diberikan yaitu hasil yaitu total volume semua barang yaitu 27,7 liter dan juga total berat semua barang yaitu 18,4 kg. Rincian barang yang akan dibawa dalam tas saat traveling yaitu jaket, kaos, celana, pakaian dalam, sarung tangan, masker, jas hujan, obat, korek api, senter anti air, sleeping bag, dan trashbag.

Selain itu, dapat juga dicobakan untuk kasus yang berbeda yaitu misalkan peancong akan berlibur ke pantai dan musimnya yaitu musim panas. Daftar list barang yang akan dibawa diberikan sebagai berikut.

Nama Barang	Berat (kg)	Volume (l)	Prioritas
Pakaian ganti	0.6	0.5	1
Handuk	0.5	1	0.8
Sunblock	0.2	0.3	1
Kacamata hitam	0.3	0.3	0.9
Karpet	2	1	0.5
Kamera	2	1.5	0.8
Alat make-up	1	1.3	0.6

Diberikan keterangan tambahan bahwa si pelancong hanya akan membawa tas mini yang dapat menampung maksimum berat yaitu 1,2 kg dan volume yaitu 1,5 liter.

Dengan menggunakan algoritma program dinamis yang tadi sudah diberikan diatas, tampilan yang dihasilkan pada terminal adalah sebagai berikut:

```

E:\KULIAH\semester_4\STIMA>python knapSack.py
2.9
[1, 0, 1, 1, 0, 0, 0]
E:\KULIAH\semester_4\STIMA>

```

**Gambar 4** Hasil Eksperimen 2

Dari hasil yang diberikan terlihat bahwa prioritas yang diperoleh dari total barang yang akan dibawa ke pantai yaitu 2.9 dengan berat total dari barang yang dibawa yaitu 1,1 kg dan volume total dari semua barang yang akan dibawa yaitu 1,1 liter. Rincian dari barang yang akan dibawa yaitu pakaian ganti, sunblock, dan juga kacamata hitam.

## V. KESIMPULAN DAN SARAN

Pertama, penulis ingin mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa karena dengan berkat-Nya penulis dapat menyelesaikan makalah dengan judul “Optimasi Pemilihan Barang yang Dibawa Traveling dengan Program Dinamis” dengan baik. Penulis juga berterima kasih kepada para dosen pengajar mata kuliah IF2211 Strategi Algoritma,

Dr. Masayu Leylia Khodra, Dr. Nur Ulfa Maulidevi, S.T, M.Sc., dan Dr. Ir. Rinaldi Munir, M.T., atas bimbingannya selama ini dalam mengajar dan memberikan ilmu sehingga penulis mampu untuk membuat makalah ini, terlebih kepada pengimplementasian algoritma terhadap ilmu lain. Penulis juga berterima kasih kepada teman dan keluarga yang telah memberikan semangat dan dorongan kepada penulis

#### VI. UCAPAN TERIMA KASIH

Pertama, penulis ingin mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa karena dengan berkat-Nya penulis dapat menyelesaikan makalah dengan judul “Optimasi Pemilihan Barang yang Dibawa Traveling dengan Program Dinamis” dengan baik. Penulis juga berterima kasih kepada para dosen pengajar mata kuliah IF2211 Strategi Algoritma, Dr. Masayu Leylia Khodra, Dr. Nur Ulfa Maulidevi, S.T, M.Sc., dan Dr. Ir. Rinaldi Munir, M.T., atas bimbingannya selama ini dalam mengajar dan memberikan ilmu sehingga penulis mampu untuk membuat makalah ini, terlebih kepada pengimplementasian algoritma terhadap ilmu lain. Penulis juga berterima kasih kepada teman dan keluarga yang telah memberikan semangat dan dorongan kepada penulis


#### REFERENSI

- [1] Munir Rinaldi . 2016 .Matematika Diskrit Revisi Keenam. Bandung : InformatikaBandung
- [2] <http://eprints.undip.ac.id/18675/1/Pertemuan10.pdf> (Diakses pada 25 April pukul 18.00).
- [3] <https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/> (Diakses pada 25 April pukul 21.00).

#### 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, 26 April 2019



Renita Napitupulu - 13517134