

Penerapan Algoritma Greedy dalam pemilihan makanan dan minuman di Warung Tegal Simpang Dago Berdasarkan Kalori dan Harga

Irsyad Nurwidiyanto Basuki - 13521072¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

¹13521072@std.stei.itb.ac.id

Abstract—Bagi mahasiswa, berburu makanan enak adalah sebuah keharusan, tetapi berburu makanan murah adalah sebuah kewajiban. Warteg, yang merupakan singkatan dari Warung Tegal, banyak ditemui di berbagai lokasi, terutama di sekitar lingkungan kampus, karena mereka mencari pelanggan di kalangan mahasiswa. Para mahasiswa sering memilih makanan yang terjangkau secara finansial untuk menghemat pengeluaran mereka, dan itulah mengapa Warteg menjadi pilihan yang populer bagi mereka. Warteg adalah jenis usaha makanan sederhana yang menyediakan berbagai masakan dengan harga yang terjangkau. Salah satu Warteg yang terkenal di kalangan mahasiswa ITB adalah Warteg Simpang Dago. Warteg ini dipilih dikarenakan letaknya yang cukup strategis dan juga pilihan lauk yang bervariasi ditambah dengan harga yang sangat terjangkau. Dengan keuangan seadanya, mahasiswa kebingungan untuk menentukan makanan yang dapat digunakan sebagai sumber energi mereka dalam beraktivitas. Dengan menggunakan algoritma *greedy*, pemilihan makanan dan minuman dapat ditentukan lebih cepat dan sesuai kebutuhan.

Keywords—Algoritma *Greedy*, Knapsack Problem, Warteg, Warteg Simpang Dago.

I. PENDAHULUAN

Mahasiswa dalam beraktivitas tentunya memerlukan sumber energi yang berasal dari makanan. Mahasiswa dapat memesan makanan secara *online* tapi dengan kekurangan harga beli makanan tersebut terbilang cukup mahal dikarenakan mengharuskan membayar ongkos kirim, biaya pelayanan dan sebagainya. Banyak mahasiswa yang merantau harus menghadapi keterbatasan keuangan yang menyebabkan tidak terpenuhinya kebutuhan sumber energi yang diperlukan. Mahasiswa lebih memilih untuk membeli makanan pada Warteg terdekat karena harga yang terjangkau serta makanan yang bervariasi.

Warteg adalah singkatan dari Warung Tegal, sebuah warung makan sederhana yang menyajikan masakan khas Tegal. Nama warteg berasal dari singkatan tersebut. Warung makan ini awalnya berasal dari tiga desa di Tegal, yaitu Sidapurna, Sidakanton, dan Krandon. Namun, seiring berjalannya waktu, bisnis ini berkembang menjadi bisnis kuliner lokal yang tersebar di seluruh Indonesia. Meskipun nama warung ini berasal dari Tegal, model warteg telah disesuaikan dengan masakan khas daerah masing-masing kota. Konsep yang digunakan oleh

warteg adalah prasmanan, di mana pelanggan dapat mengambil beberapa menu yang diinginkan dan membayarnya sesuai dengan menu yang diambil. Kemunculan warteg di berbagai daerah tidak hanya dipengaruhi oleh faktor ekonomi agar terjangkau, tetapi juga telah menjadi tradisi di Indonesia.

Warung Tegal Simpang Dago, yang terletak di sekitar wilayah ITB, merupakan salah satu warteg yang populer oleh kalangan mahasiswa ITB. Warteg ini dikenal karena lokasinya yang strategis dan harga makanannya yang terjangkau. Keunggulan utama warteg ini adalah kemampuannya untuk menerima pembayaran non-tunai, yaitu dengan menggunakan QRIS (Quick Response Code Indonesian Standard). Warteg ini berlokasi tepatnya di Jl. Ir. H. Juanda No. 172, Lebakgede, Kecamatan Coblong, Kota Bandung, Jawa Barat. Secara spesifik, warteg ini berada di seberang timur restoran McDonald's Simpang Dago.

Warteg Simpang Dago ini berhasil memikat para mahasiswa untuk memenuhi kebutuhan sumber energi mereka. Warteg ini menyajikan berbagai pilihan makanan dengan harga yang terjangkau. Bagi mahasiswa, berburu makanan enak adalah sebuah keharusan, tetapi berburu makanan murah adalah sebuah kewajiban.

Dengan ini, untuk mendapatkan gabungan makanan untuk mendapatkan kalori (sumber energi) yang maksimal dan harga yang murah diperlukan algoritma yang tepat agar mahasiswa tidak salah memilih makanan dan minuman yang ada. Pada makalah ini, penerapan algoritma *greedy* akan digunakan.



Gambar 1. Warteg Simpang Dago
Sumber : Dokumentasi pribadi

II. LANDASAN TEORI

A. Algoritma *Greedy*

Algoritma *greedy* adalah algoritma yang memecahkan persoalan secara langkah per langkah (step by step) sedemikian sehingga, pada setiap langkah:

1. Mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip "take what you can get now!")
2. "berharap" bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

Elemen-elemen algoritma *greedy*:

1. Himpunan kandidat, C : berisi kandidat yang akan dipilih pada setiap Langkah (misal: simpul/sisi di dalam graf, job, task, koin, benda, karakter, dsb).
2. Himpunan solusi, S : berisi kandidat yang sudah dipilih.
3. Fungsi solusi: menentukan apakah himpunan kandidat yang dipilih sudah memberikan solusi.
4. Fungsi seleksi (selection function): memilih kandidat berdasarkan strategi *greedy* tertentu. Strategi *greedy* ini bersifat heuristik.
5. Fungsi kelayakan (feasible): memeriksa apakah kandidat yang dipilih dapat dimasukkan ke dalam himpunan solusi (layak atau tidak).
6. Fungsi obyektif : memaksimumkan atau meminimumkan.

Dengan menggunakan elemen-elemen di atas, maka dapat dikatakan bahwa: Algoritma *greedy* melibatkan pencarian sebuah himpunan bagian, S , dari himpunan kandidat, C ; yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu S menyatakan suatu solusi dan S dioptimisasi oleh fungsi obyektif.

Optimum global belum tentu merupakan solusi optimum (terbaik), bisa jadi merupakan solusi sub-optimum atau pseudo-optimum dengan alasan sebagai berikut :

1. Algoritma *greedy* tidak beroperasi secara menyeluruh terhadap semua kemungkinan solusi yang ada (sebagaimana pada metode exhaustive search).
2. Terdapat beberapa fungsi "seleksi" yang berbeda, sehingga kita harus memilih fungsi yang tepat jika kita ingin algoritma menghasilkan solusi optimal.

B. Kalori

Kalori adalah satuan pengukuran energi dalam makanan. Dalam konteks nutrisi dan makanan, kalori sering disebut sebagai kilokalori yang setara dengan 1.000 kalori. Jadi, ketika kita membicarakan kalori dalam makanan, biasanya mengacu pada kilokalori atau istilah "kalori" yang lebih umum digunakan.

Kalori yang diperoleh dari makanan dan minuman berperan sebagai sumber energi dalam tubuh manusia untuk menjalankan fungsi-fungsi tubuh seperti bernapas, menjaga suhu tubuh, mengolah makanan, beraktivitas fisik, dan menjaga fungsi organ-organ lainnya. Kebutuhan kalori seseorang bervariasi tergantung pada faktor-faktor seperti usia, jenis kelamin, tingkat aktivitas fisik, dan komposisi tubuh.

Dalam konteks gizi, makanan dan minuman memiliki nilai kalori yang berbeda bergantung pada kandungan makronutrien (karbohidrat, protein, dan lemak) yang terdapat di dalamnya.

III. PENERAPAN DALAM PEMILIHAN MAKANAN DAN MINUMAN DI WARTEG SIMPANG DAGO

Warteg Simpang Dago menyediakan berbagai macam makanan serta minuman. Tentunya tiap makanan dan minuman terdapat perbedaan kalori yang dihasilkan, Ada makanan yang dan minuman yang menghasilkan kalori yang tinggi, ada juga yang menghasilkan kalori rendah. Pemilihan makanan ini dapat bergantung pada kebutuhan tiap mahasiswa.

Kebutuhan mahasiswa dapat berbeda – berbeda. Tentunya perbedaan ini tergantung terhadap kebutuhan tiap mahasiswa. Ada mahasiswa yang ingin menaikkan berat badan. Dengan ini biasanya mahasiswa memprioritaskan untuk memilih makanan yang dapat menghasilkan surplus kalori atau kalori yang berlebih.

Ada pula mahasiswa lebih memprioritaskan harga makanan yang dibeli makanan dibanding mementingkan kalori yang diperlukan. Biasanya mahasiswa sudah berada di masa "akhir bulan" karena uangnya sudah habis untuk jajan dari awal bulan hingga pertengahan bulan.

Ada juga mahasiswa memperhatikan kalori dan harga dari suatu makanan dan minuman.

Algoritma *greedy* sangat cocok untuk menghadapi masalah ini. Algoritma *greedy* dapat mencari kombinasi makanan untuk mahasiswa yang mementingkan kalori, yang mementingkan harga serta yang mementingkan keduanya. Berikut merupakan nama, harga, serta kalori dari makanan dan minuman yang ada di Warteg Simpang Dago :

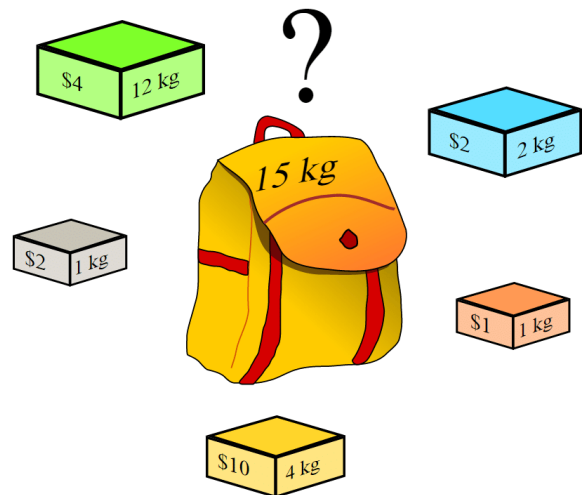
No	Nama	Harga	Kalori
1	Nasi Putih	4000	204
2	Nasi Putih 1/2	4000	102
3	Ayam Balado	7000	246
4	Ayam Cabe Ijo	7000	382
5	Ayam Gulai	7000	293
6	Ayam Kecap	7000	223
7	Ayam Goreng	7000	284
8	Ayam Opor	7000	392
9	Ikan Kembung	6000	112
10	Ikan Lele	6000	105
11	Ikan Tongkol	6000	200
12	Ikan Patin	6000	350
13	Ikan Gurame	6000	127
14	Udang Balado	5000	279
15	Kerang Ijo	3000	217
16	Tempe Balok	3000	195
17	Tempe Orek Kering	2000	254
18	Tempe Orek Basah	2000	265

19	Tahu Kecil	2000	131
20	Tahu Besar	3000	115
21	Telur Ceplok Balado	4000	216
22	Telur Bulat Balado	4000	202
23	Telur Asin	4000	137
24	Telur Dadar	4000	93
25	Rolade	5000	280
26	Sosis	5000	70
27	Otak - otak	5000	189
28	Sayur Kangkung	3000	28
29	Sayur Bayam	3000	23
30	Sayur Capcai	3000	216
31	Sayur Buncis	3000	31
32	Sayur Daun Sinkong	3000	38
33	Sayur Pare	3000	17
34	Sayur Pakcoy	3000	9
35	Kerupuk Kaleng	1000	65
36	Kerupuk Puli	1000	65
37	Kentang Mustofa	2000	153
38	Kentang Balado	3000	303
39	Bakwan Goreng	1000	137
40	Tempe Mendoan	1000	200
41	Perkedel	3000	21
42	Air Putih	0	0
43	Teh Tawar	0	2
44	Teh Manis	3000	55
45	Jeruk	3000	50
46	Kopi Hitam	3000	30
47	Kopi Susu	3000	160
48	Susu	3000	130

Tabel 1.

Permasalahan memilih makanan dan minuman ini mirip dengan masalah 1/0 Knapsack Problem. Knapsack Problem adalah persoalan dimana diberikan n buah objek dan sebuah *knapsack* dengan kapasitas bobot K . Setiap objek memiliki properti bobot (weight) w_i dan keuntungan (profit) p_i . Bagaimana cara memilih objek -objek yang dimasukkan ke dalam *knapsack* sedemikian sehingga diperoleh total

keuntungan yang maksimal. Total bobot objek yang dimasukkan tidak boleh melebihi kapasitas *knapsack*.



Gambar 2. Knapsack Problem
Sumber : [4]

Pada kasus pemilihan makanan dan minuman pada Warteg Simpang Dago ini kapasitas *knapsack* yang dimaksud adalah uang yang telah kita siapkan untuk membeli makanan dan minuman. Harga (atau bobot) makanan dan minuman yang dipilih tidak boleh melebihi uang yang telah kita siapkan – atau jika melebihi, kita akan diminta untuk mencuci piring.

Sementara kalori pada makanan dan minuman merupakan *profit* yang bisa didapat dari suatu makanan dan minuman. Tujuan pada kasus ini adalah bagaimana caranya mendapatkan kalori sebanyak – banyaknya dengan berbagai metode *Greedy*. Metode *greedy* yang akan digunakan adalah *greedy by calories*, berdasarkan kalori, *greedy by price*, berdasarkan harga dan *greedy by density*, berdasarkan kalori per harga.

A. Greedy by calories

Pada strategi algoritma ini, algoritma akan ‘rakus’ dalam mencari kalori terbesar terlebih dahulu dari makanan dan minuman yang ada hingga harga mencapai atau kurang dari uang yang telah disiapkan. Berikut elemen – elemen *greedy* yang digunakan :

1. Himpunan Kandidat, C : himpunan semua makanan atau minuman yang tersedia untuk dipilih. Setiap makanan atau minuman memiliki informasi tentang kalori dan harga.
2. Himpunan Solusi, S : himpunan makanan atau minuman yang dipilih dari himpunan kandidat C . Awalnya, himpunan solusi S kosong lalu diisi dengan makanan atau minuman yang telah dipilih dari strategi *greedy by calories*.
3. Fungsi Solusi: mengecek apakah makanan atau minuman di himpunan solusi sudah merupakan solusi dari permasalahan ini.
4. Fungsi Seleksi (Selection Function): memilih makanan atau minuman dengan kalori tertinggi dari himpunan kandidat C yang memenuhi kriteria kelayakan.
5. Fungsi Kelayakan (Feasible): melakukan pengecekan apakah suatu makanan atau minuman layak untuk dimasukkan ke dalam himpunan solusi S . Dalam konteks ini, fungsi kelayakan memeriksa apakah harga dan kalori dari makanan atau minuman yang dipilih tidak melebihi uang yang telah disiapkan

6. Fungsi Obyektif: Jumlah kalori seluruh makanan atau minuman yang dipilih maksimum

Dengan menggunakan elemen-elemen di atas, algoritma *greedy* berbasis kalori dapat memilih makanan atau minuman dengan kalori tertinggi pada setiap langkah dan secara iteratif membangun himpunan solusi S yang optimal berdasarkan kalori yang ada.

Berikut diberikan *pseudocode* untuk algoritma *greedy by calories*:

```
function byCalories(nasi, ayam, seafood,
lauklain, tambahan, sayur, minum, budget):
solution_set <- []
remaining_budget <- budget
while remaining_budget > 0 do
max_calories <- 0
selected_item <- None

for items in [nasi, ayam, seafood, lauklain,
tambahan, sayur, minum] do
valid_items <- []
for item in items do
if item.price <= remaining_budget then
valid_items.append(item)
if valid_items is not empty then
sort valid_items by item.calories in
descending order
for current_item in valid_items then
if (current_item.calories >
max_calories) and (current_item not in
solution_set) then
max_calories <- current_item.calories
selected_item <- current_item
break
if selected_item is not null then
add selected_item to solution_set
remaining_budget <- remaining_budget -
selected_item.price
else
break
if solution_set is not empty then
total_price <- sum of price for each item in
solution_set
total_calories <- sum of calories for each
item in solution_set
print "Selected Items:"
```

```
for item in solution_set do
print "Name: item.name, Price:
Rpitem.price, Calories: item.calories"
print "Total Harga: Rp" + total_price
print "Total Kalori: " + total_calories
else
print "Tidak ada kombinasi yang dapat
memenuhi budget"
```

Berikut merupakan hasil pengujian *greedy by calories* menggunakan *dataset* makanan atau minuman pada Tabel 1 dengan maksimum pengeluaran uang Rp25000 :

```
Makanan atau minuman yang dipilih:
Nama: Ayam Opor, Harga: Rp7000, Kalori: 392
Nama: Ayam Cabe Ijo, Harga: Rp7000, Kalori: 382
Nama: Ikan Patin, Harga: Rp6000, Kalori: 350
Nama: Udang Balado (100gr), Harga: Rp5000, Kalori: 279
Total Harga: Rp25000
Total Kalori: 1403
```

Gambar 3. Hasil uji *greedy by calories*

B. *Greedy by price*

Pada strategi algoritma ini, algoritma akan ‘rakus’ dalam mencari harga termurah terlebih dahulu dari makanan dan minuman yang ada hingga harga mencapai atau kurang dari uang yang telah disiapkan. Berikut elemen – elemen *greedy* yang digunakan :

1. Himpunan Kandidat, C: himpunan semua makanan atau minuman yang tersedia untuk dipilih. Setiap makanan atau minuman memiliki informasi tentang kalori dan harga.
2. Himpunan Solusi, S: himpunan makanan atau minuman yang dipilih dari himpunan kandidat C. Awalnya, himpunan solusi S kosong lalu diisi dengan makanan atau minuman yang telah dipilih dari strategi *greedy by price*.
3. Fungsi Solusi: mengecek apakah makanan atau minuman di himpunan solusi sudah merupakan solusi dari permasalahan ini.
4. Fungsi Seleksi (Selection Function): memilih makanan atau minuman dengan harga terendah dari himpunan kandidat C yang memenuhi kriteria kelayakan.
5. Fungsi Kelayakan (Feasible): melakukan pengecekan apakah suatu makanan atau minuman layak untuk dimasukkan ke dalam himpunan solusi S. Dalam konteks ini, fungsi kelayakan memeriksa apakah harga dari makanan atau minuman yang dipilih tidak melebihi uang yang telah disiapkan
6. Fungsi Obyektif: Jumlah kalori seluruh makanan atau minuman yang dipilih maksimum

Dengan menggunakan elemen-elemen di atas, algoritma *greedy* berbasis kalori dapat memilih makanan atau minuman dengan harga terendah pada setiap langkah dan secara iteratif membangun himpunan solusi S yang optimal berdasarkan kalori yang ada.

Berikut diberikan *pseudocode* untuk algoritma *greedy by price*:

```
function byPrice(nasi, ayam, seafood,
lauklain, tambahan, sayur, minum, budget):
```

```

solution_set <- []
remaining_budget <- budget
while remaining_budget > 0 do
  min_price <- positive infinity
  selected_item <- null
  for items in (nasi, ayam, seafood,
lauklain, tambahan, sayur, minum) do
    valid_items <- empty list
    for item in items do
      if item.price <= remaining_budget:
        add item to valid_items
        if valid_items is not empty then
          sort valid_items by item.price in
ascending order
          for current_item in valid_items do
            if (current_item.price < min_price)
and (current_item not in solution_set) then
              min_price <- current_item.price
              selected_item <- current_item
              break

        if selected_item is not null then
          add selected_item to solution_set
          remaining_budget <- remaining_budget -
selected_item.price
        else
          break

    if solution_set is not empty then
      total_price <- sum of price for each item
in solution_set
      total_calories <- sum of calories for each
item in solution_set
      print "Selected Items:"
      for item in solution_set do
        print "Name: item.name, Price:
Rpitem.price, Calories: item.calories"
        print "Total Harga: Rp" + total_price
        print "Total Kalori: " + total_calories
      else
        print "Tidak ada kombinasi yang dapat
memenuhi budget"

```

Berikut merupakan hasil pengujian *greedy by price* menggunakan *dataset* makanan atau minuman pada Tabel 1 dengan maksimum pengeluaran uang Rp25000 :

```

Makanan atau minuman yang dipilih:
Nama: Air Putih, Harga: Rp1, Kalori: 0
Nama: Teh Tawar, Harga: Rp1, Kalori: 2
Nama: Kerupuk Kaleng, Harga: Rp1000, Kalori: 65
Nama: Kerupuk Puli, Harga: Rp1000, Kalori: 65
Nama: Bakwan Goreng, Harga: Rp1000, Kalori: 65
Nama: Tempe Mendoan, Harga: Rp1000, Kalori: 65
Nama: Tempe Orek Kering, Harga: Rp2000, Kalori: 195
Nama: Tempe Orek Basah, Harga: Rp2000, Kalori: 195
Nama: Tahu Kecil, Harga: Rp2000, Kalori: 195
Nama: Kentang Mustofa, Harga: Rp2000, Kalori: 65
Nama: Tempe Balok, Harga: Rp3000, Kalori: 195
Nama: Tahu Besar, Harga: Rp3000, Kalori: 195
Nama: Kentang Balado, Harga: Rp3000, Kalori: 65
Nama: Perkedel, Harga: Rp3000, Kalori: 65
Total Harga: Rp24002
Total Kalori: 1432

```

Gambar 4. Hasil uji *greedy by price*

C. *Greedy by density*

Density yang dimaksud di sini adalah perbandingan antara kalori dari tiap makanan dan minuman dengan harga dari tiap makanan dan minuman. Berikut penjelasan elemen – elemen *greedy* yang digunakan :

1. Himpunan Kandidat, C: Ini adalah himpunan semua makanan atau minuman yang tersedia untuk dipilih. Setiap makanan atau minuman memiliki informasi tentang kalori dan harga.
2. Himpunan Solusi, S: himpunan yang berisi makanan atau minuman yang dipilih dari himpunan kandidat C. Awalnya, himpunan solusi S kosong lalu diisi dengan makanan atau minuman yang telah dipilih dari strategi *greedy by density*.
3. Fungsi Solusi: mengecek apakah makanan atau minuman di himpunan solusi sudah merupakan solusi dari permasalahan ini.
4. Fungsi Seleksi (Selection Function): memilih makanan atau minuman dengan *density* atau kepadatan tertinggi dari himpunan kandidat C yang memenuhi kriteria kelayakan. Kepadatan dihitung dengan membagi kalori dengan harga yang diberikan oleh makanan atau minuman tersebut.
5. Fungsi Kelayakan (Feasible): melakukan pengecekan apakah suatu makanan atau minuman layak untuk dimasukkan ke dalam himpunan solusi S. Dalam konteks ini, fungsi kelayakan memeriksa apakah harga dari makanan atau minuman yang dipilih tidak melebihi batas uang yang telah disiapkan.
6. Fungsi Obyektif: Jumlah kalori seluruh makanan atau minuman yang dipilih maksimum

Dengan menggunakan elemen-elemen di atas, algoritma *greedy* berbasis kepadatan (*density*) dapat memilih makanan atau minuman dengan kepadatan (*density*) tertinggi pada setiap langkah dan secara iteratif membangun himpunan solusi S yang optimal berdasarkan perbandingan kalori dengan harga.

Berikut diberikan *pseudocode* untuk algoritma *greedy by density* :

```

function byDensity(nasi, ayam, seafood,
lauklain, tambahan, sayur, minum, budget):

```

```

items <- [nasi, ayam, seafood, lauklain,
tambahan, sayur, minum]
solution_set <- []
remaining_budget <- budget
while remaining_budget > 0 do
  max_density <- 0
  selected_item <- None
  for category_items in items do
    valid_items <- []
    for item in category_items do
      if item.price <= remaining_budget then
        valid_items.append(item)
        if valid_items is not empty then
          sort valid_items by (item.calories /
item.price) in descending order
          current_item <- valid_items[0]
          if (current_item.calories /
current_item.price) > max_density then
            max_density <- (current_item.calories
/ current_item.price)
            selected_item <- current_item

        if selected_item is not null then
          add selected_item to solution_set
          remaining_budget <- remaining_budget -
selected_item.price
          for category_items in items do
            if selected_item exists in category_items
then
              remove selected_item from category_items
            else
              break

if solution_set is not empty then
  total_price <- sum of price for each item
in solution_set
  total_calories <- sum of calories for each
item in solution_set
  print "Selected Items:"
  for item in solution_set do
    print "Name: item.name, Price:
Rpitem.price, Calories: item.calories"
    print "Total Harga: Rp" + total_price

```

```

print "Total Kalori: " + total_calories
else
  print "Tidak ada kombinasi yang dapat
memenuhi budget."

```

Berikut merupakan hasil pengujian *greedy by density* menggunakan *dataset* makanan atau minuman pada Tabel 1 dengan maksimum pengeluaran uang Rp25000 :

```

Makanan atau minuman yang dipilih:
Nama: Teh Tawar, Harga: Rp1, Kalori: 2
Nama: Tempe Orek Kering, Harga: Rp2000, Kalori: 195
Nama: Tempe Orek Basah, Harga: Rp2000, Kalori: 195
Nama: Tahu Kecil, Harga: Rp2000, Kalori: 195
Nama: Sayur Capcai, Harga: Rp3000, Kalori: 216
Nama: Tempe Balok, Harga: Rp3000, Kalori: 195
Nama: Tahu Besar, Harga: Rp3000, Kalori: 195
Nama: Kerupuk Kaleng, Harga: Rp1000, Kalori: 65
Nama: Kerupuk Puli, Harga: Rp1000, Kalori: 65
Nama: Bakwan Goreng, Harga: Rp1000, Kalori: 65
Nama: Tempe Mendoan, Harga: Rp1000, Kalori: 65
Nama: Udang Balado (100gr), Harga: Rp5000, Kalori: 279
Total Harga: Rp24001
Total Kalori: 1732

```

Gambar 5. Hasil uji *greedy by density*

IV. KESIMPULAN

Setelah dilakukan pengujian dengan tiga strategi, yaitu *greedy by calories*, *greedy by price*, dan *greedy by density* didapatkan hasil yang berbeda – beda. Ini dikarenakan penggunaan algoritma *greedy* tidak selalu menghasilkan hasil yang paling optimal. Tapi hasil yang didapatkan sudah mendekati hasil yang optimal, dalam konteks ini mendapatkan kalori sebanyak – banyaknya.

Dari makalah ini kita dapat mendapatkan kesimpulan bahwa penerapan ilmu strategi algoritma sangat berguna untuk kehidupan sehari – hari tidak hanya permasalahan komputasi saja. Ilmu strategi algoritma ini tidak hanya digunakan saat ujian seperti UTS (Ujian Tengah Semester) atau UAS (Ujian Akhir Semester) saja tapi juga bisa digunakan ke hal – hal yang menurut kita kecil seperti memilih makanan atau minuman pada rumah makan dan ternyata menghasilkan hasil sangat berguna. Seperti dalam makalah ini, kita menggunakan algoritma *greedy* untuk mendapatkan kalori sebanyak – banyaknya yang digunakan sebagai sumber energi.

V. PENUTUP

Penulis memanjatkan puji syukur kepada Allah SWT karena atas kasih dan karunia-Nya, penulis dapat menyelesaikan Tugas Makalah Strategi Algoritma dengan tepat waktu. Penulis juga ingin mengucapkan terima kasih kepada Ibu Dr. Nur Ulfa Maulidevi, Bapak Dr. Rila Mandala, dan Bapak Dr. Rinaldi Munir selaku dosen mata kuliah IF2211 Strategi Algoritma yang telah memberi ilmu, materi, serta pengalaman yang membantu dalam penulisan makalah ini. Penulis juga mengucapkan terima kasih kepada keluarga dan teman penulis yang telah memberikan dukungan penuh. Penulis juga mengucapkan terima kasih kepada Warteg Simpang Dago beserta pelayan - pelayannya karena telah terus menemani penulis berkuliah di ITB disaat suka maupun duka. Penulis juga ingin meminta maaf apabila terdapat kekurangan atau kesalahan pada makalah ini

REFERENSI

- [1] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf). Diakses pada 20 Mei 2023
- [2] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag2.pdf). Diakses pada 20 Mei 2023
- [3] <https://evermos.com/home/panduan/warteg/> Diakses pada 20 Mei 2023
- [4] <https://www.theoptimizationexpert.com/case-studies/knapsack/>. Diakses pada 21 Mei 2023

LAMPIRAN

Pranala video YouTube : <https://youtu.be/a2ufPNwn-m8>

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 2023



Irsyad Nurwidiyanto Basuki - 13521072