

Laporan Project UAS SDA

Kelompok 5 Matematika 2020D

Anggota Kelompok :

1. Muhammad Arif Hunaifi (20030214015)
2. Iva Imana Amalia (20030214017)

Maksimalisasi Pengangkutan Barang dengan Algoritma Greedy

A. Pendahuluan

1. Latar Belakang :

Dengan pesatnya perkembangan perekonomian Indonesia. Perkembangan industri distribusi juga semakin meningkat untuk memenuhi permintaan pasar dalam mengantarkan barang kepada konsumen. Dengan pertumbuhan tersebut, diharapkan industri distribusi semakin meningkatkan kinerja dan kualitas pelayanan pengiriman barang. Untuk menghemat pengiriman dapat dilakukan dengan cara mengoptimalkan pengiriman dengan memaksimalkan daya angkutnya.

Barang yang akan dikirim harus meminimalkan total berat atau volume kapasitas tanpa melebihi batas maksimum. Penyusunan barang yang tidak optimal membutuhkan tambahan kontainer lain untuk mengangkut sisa barang yang seharusnya dapat dioptimalkan dengan menggunakan satu kontainer saja. Biaya yang dikeluarkan pun berlipat ganda untuk sekali ekspedisi pengiriman. Hal tersebut, dapat merugikan perusahaan yang menggunakan jasa pengiriman karena harus membayar biaya pengiriman yang berlipat untuk satu kali pengiriman barang.

Rencana penyusunan barang yang akan dibuat dapat dibantu dengan program python yang menggunakan knapsack problem dengan algoritma greedy yang nantinya akan memudahkan simulasi penataan barang yang optimal pada kontainer.

2. Rumusan Masalah :

- a. Bagaimana menentukan keuntungan maksimum pada permasalahan knapsack pada kontainer dengan menggunakan algoritma greedy by weight.
- b. Bagaimana menentukan keuntungan maksimum pada permasalahan knapsack pada kontainer dengan menggunakan algoritma greedy by profit.
- c. Bagaimana menentukan keuntungan maksimum pada permasalahan knapsack pada kontainer dengan menggunakan algoritma greedy by density.

3. Tujuan :

Berdasarkan rumusan masalah di atas, maka tujuan yang akan dibahas adalah.

- a. Mencari keuntungan maksimum pada permasalahan knapsack pada kontainer dengan menggunakan algoritma greedy by weight.

- b. Mencari keuntungan maksimum pada permasalahan knapsack pada kontainer dengan menggunakan algoritma greedy by profit.
- c. Mencari keuntungan maksimum pada permasalahan knapsack pada kontainer dengan menggunakan algoritma greedy by density.

4. Manfaat :

- a. Untuk mengetahui pola penyusunan barang teroptimal sehingga barang dapat tersusun dengan baik dalam kontainer.
- b. Menghasilkan sistem yang efektif dan efisien dalam manajemen kontainer yang dapat membuat produktivitas meningkat.
- c. Memaksimalkan pemanfaatan ruang yang tersedia dalam kontainer.

B. Landasan Teori

1. Knapsack Problem

Knapsack problem merupakan suatu permasalahan pemilihan barang yang akan disimpan atau dimasukkan ke suatu tempat dengan memiliki keterbatasan kapasitas (Juwita, Susanto, dan Halomoan (2017)). Pengertian lainnya, Knapsack Problem adalah masalah penempatan item (barang) ke dalam suatu tempat (biasa disebut Knapsack) yang mempunyai kapasitas tertentu, dimana setiap item memiliki berat dan nilai, sehingga total berat dari item-item yang ditempatkan tidak melebihi kapasitas.

2. Algoritma Greedy

Algoritma greedy adalah metode yang tepat untuk memecahkan masalah optimasi. Ada dua macam masalah optimasi, yaitu maksimasi dan minimasi. Prinsip dari algoritma greedy menurut Ghozali, Setiawan, & Furqon (2017) adalah mengambil sebanyak mungkin dari apa yang dapat diperoleh sekarang. Algoritma Greedy merupakan metode yang biasa digunakan untuk menyelesaikan knapsack problem.

Ada 3 Macam Algoritma Greedy untuk menyelesaikan permasalahan Knapsack , yaitu:

a. Greedy By Weight

Setiap langkah di knapsack problem diisi dengan barang yang mempunyai berat paling ringan. Strategi ini bertujuan untuk memaksimalkan keuntungan dengan memasukkan barang sebanyak mungkin. Tahap awalnya adalah mengurutkan secara menurun barang-barang berdasarkan berat barang yang paling ringan.

b. Greedy By Profit

Pada setiap langkah, knapsack diisi dengan barang-barang dengan keuntungan terbesar. Strategi ini berusaha untuk memaksimalkan keuntungan dengan memilih yang paling menguntungkan terlebih dahulu. Hal pertama yang

harus dilakukan adalah mengurutkan objek berdasarkan keuntungannya. Kemudian ambil barang-barang yang bisa ditampung knapsack satu per satu sampai knapsack penuh.

c. Greedy By Density

Pada setiap langkah, knapsack diisi dengan obyek yang mempunyai densitas terbesar (perbandingan nilai dan berat terbesar). Strategi ini mencoba memaksimalkan keuntungan dengan memilih objek yang mempunyai keuntungan per unit berat terbesar. Pertama kali yang dilakukan adalah mencari nilai profit per unit/ density dari tiap-tiap objek. Kemudian obyek-obyek diurutkan berdasarkan densitasnya.

C. Implementasi Algoritma

Knapsack merupakan suatu permasalahan bagaimana memilih objek dari sekian banyak objek dan berapa besar objek tersebut akan disimpan sehingga diperoleh suatu penyimpanan yang optimal dengan memperhatikan objek yang terdiri dari n objek (1, 2, 3, ..., n) dimana setiap objek memiliki bobot (w_i) dan nilai profit (p_i) dengan memperhatikan juga kapasitas dari media penyimpanan sebesar (M).

Algoritma greedy digunakan untuk memperoleh penyelesaian dari suatu permasalahan optimasi. Suatu permasalahan dengan n masukkan data dilakukan secara bertahap. Pertama dilakukan pemilihan solusi yang mungkin kemudian dari himpunan solusi yang mungkin tersebut akan diperoleh solusi optimal.

Metode ini bekerja secara bertahap dengan memperhatikan setiap input data pada setiap keadaan. Pada setiap tahap, dibuat keputusan dengan memperhatikan ada atau tidak sebuah input data yang memberikan solusi optimal, dan memperhatikan pula urutan data dalam proses pengambilannya. Pendekatan yang digunakan pada algoritma Greedy adalah membuat pilihan yang dapat memberikan perolehan yang terbaik yaitu dengan membuat pilihan optimum lokal pada setiap langkah dengan tujuan bahwa sisanya mengarah ke solusi optimum global.

Terdapat beberapa strategi Greedy yang dapat digunakan untuk memilih objek yang akan dimasukkan ke dalam M antara lain.

1) Greedy by profit

Algoritma greedy by profit:

- a) Tetapkan nilai kapasitas maksimum knapsack;
- b) Urutkan objek-objek berdasarkan keuntungan (profit) dari yang terbesar;
- c) Isi knapsack dengan objek yang memiliki keuntungan terbesar terlebih dahulu;
- d) Ambil satu-persatu objek yang dapat ditampung sampai kapasitas knapsack penuh;
- e) Hitung jumlah bobot dan keuntungan.

2) Greedy by weight

Algoritma greedy by weight:

- Tetapkan nilai kapasitas maksimum knapsack;
- Urutkan objek-objek berdasarkan berat dari yang teringan;
- Isi knapsack dengan objek yang memiliki berat teringan terlebih dahulu;
- Ambil satu-persatu objek yang dapat ditampung sampai kapasitas knapsack penuh;
- Hitung jumlah bobot dan keuntungan.11

3) Greedy by density

Algoritma greedy by density:

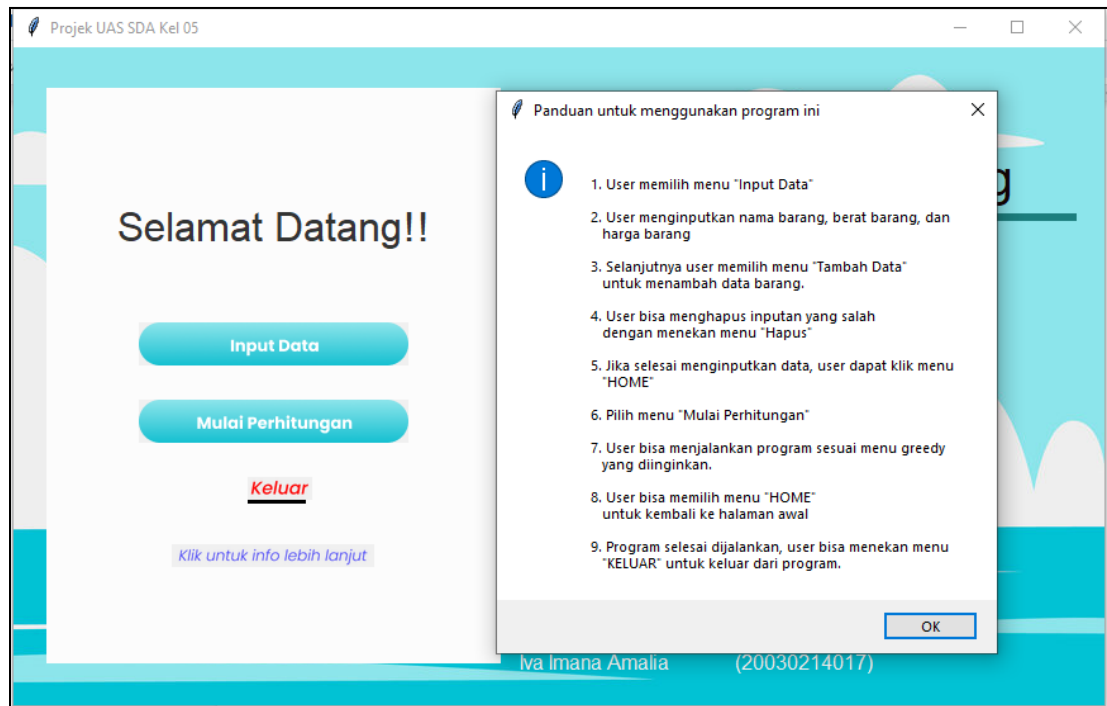
- Tetapkan nilai kapasitas maksimum knapsack;
- Hitung rasio (/) dari tiap-tiap barang;
- Urutkan objek-objek berdasarkan rasio (density) terbesar terlebih dahulu;
- Ambil satu-persatu objek yang dapat ditampung sampai kapasitas knapsack penuh;
- Hitung jumlah bobot dan keuntungan

D. Desain Interface

1) Tampilan Awal Program



2) Tampilan Menu “Info atau Panduan Pengguna”

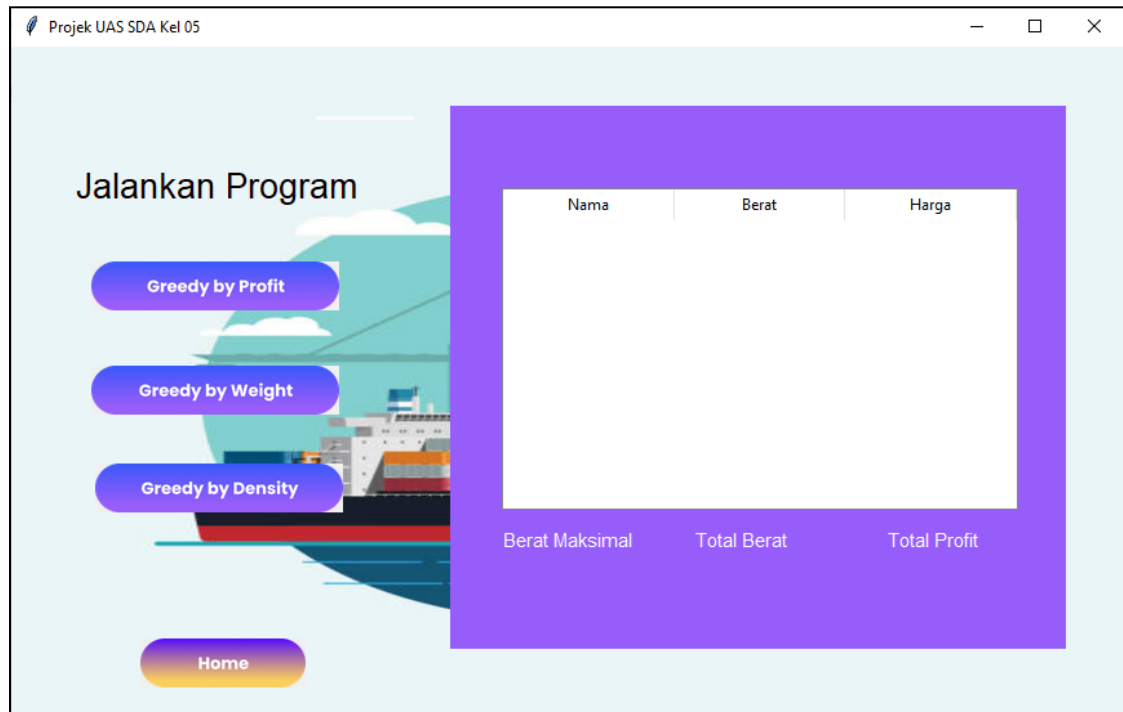


3) Tampilan Menu “Input Data”

The screenshot shows the 'Input Data' menu. It includes a text input for 'Kapasitas Maksimal (Kg)', a 'Tambah data' button, a 'Hapus' button, and text inputs for 'Nama Barang', 'Berat Barang (Kg)', and 'Harga Barang (Rp)'. A 'Home' button is at the bottom. A table overlay is visible on the right side of the form.

Nama	Berat	Harga
------	-------	-------

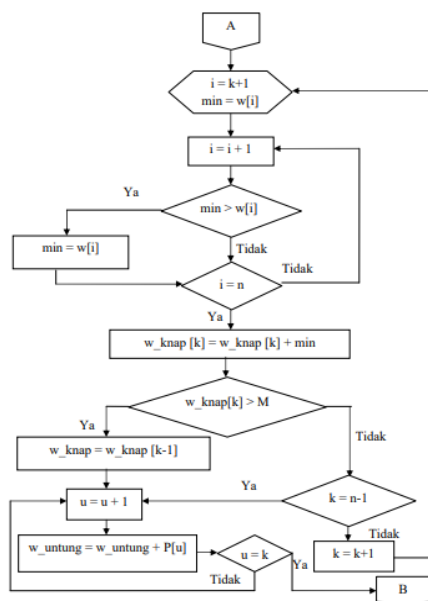
4) Tampilan Menu “Mulai Perhitungan”



E. Hasil dan Pembahasan

Perhitungan Kompleksitas Waktu

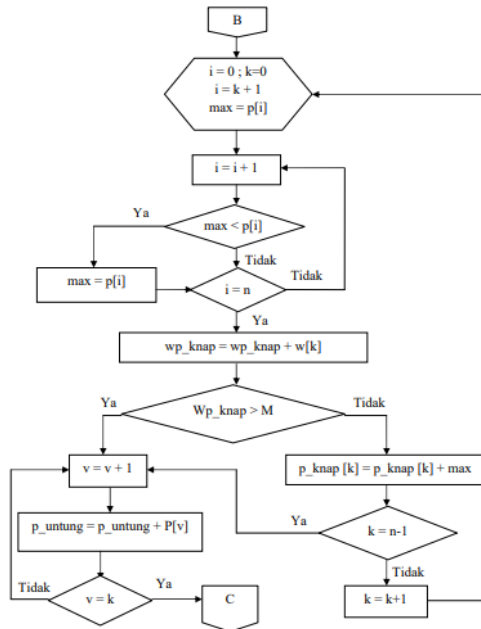
a. Greedy By Weight



Jumlah perhitungan kompleksitas waktu Greedy by Weight adalah

$$T(n) = 3n^2 + 2n + k - 2, O(n^2)$$

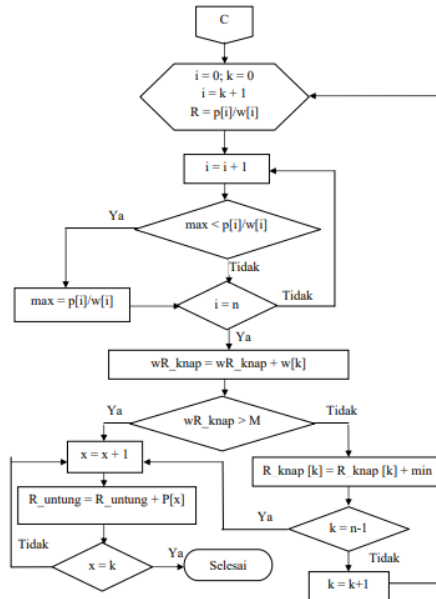
b. Greedy by profit



Jumlah perhitungan kompleksitas waktu Greedy by Profit adalah

$$T(n) = 3n^2 + 4n + k - 3, O(n^2)$$

c. Greedy by density



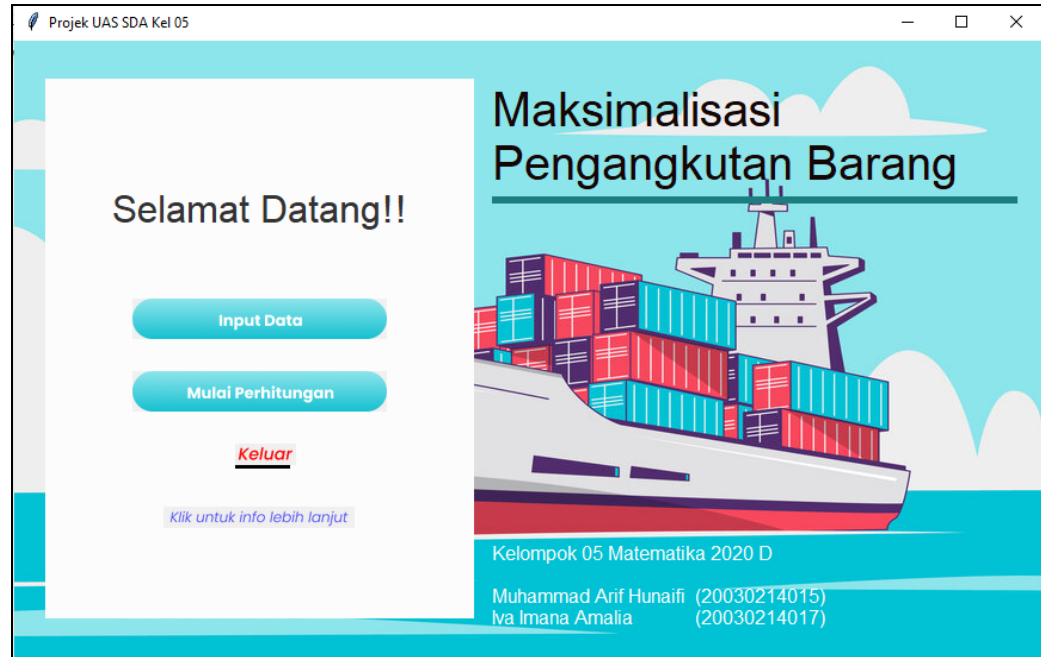
Jumlah perhitungan kompleksitas waktu Greedy by Density adalah

$$T(n) = 9n^2 + 13n + 3k - 6, O(n^2)$$

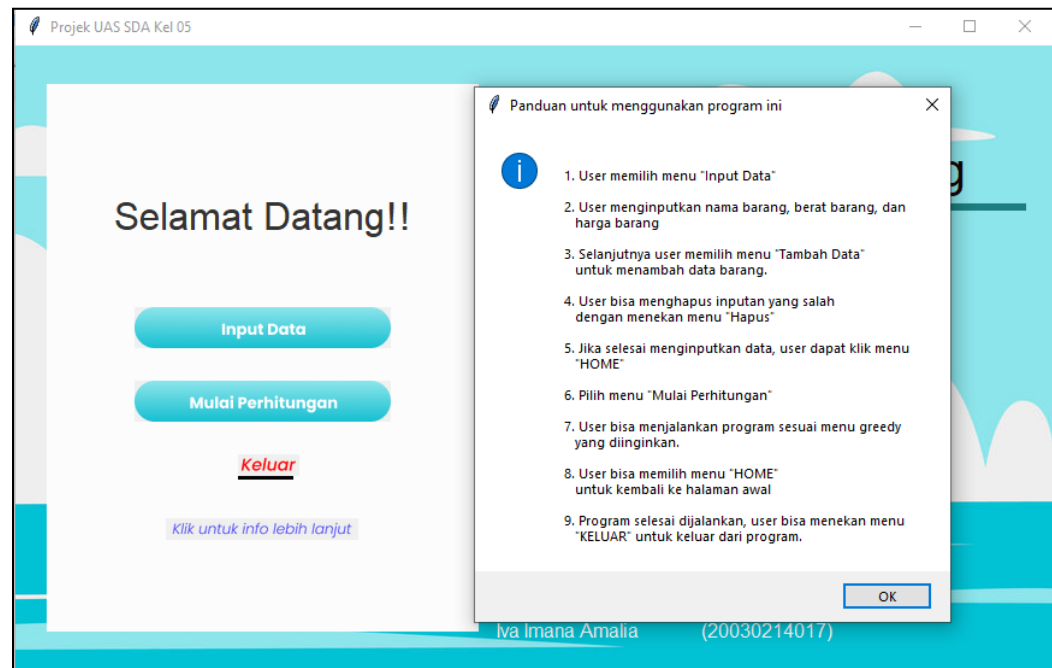
F. Lampiran

1. User Manual Guide

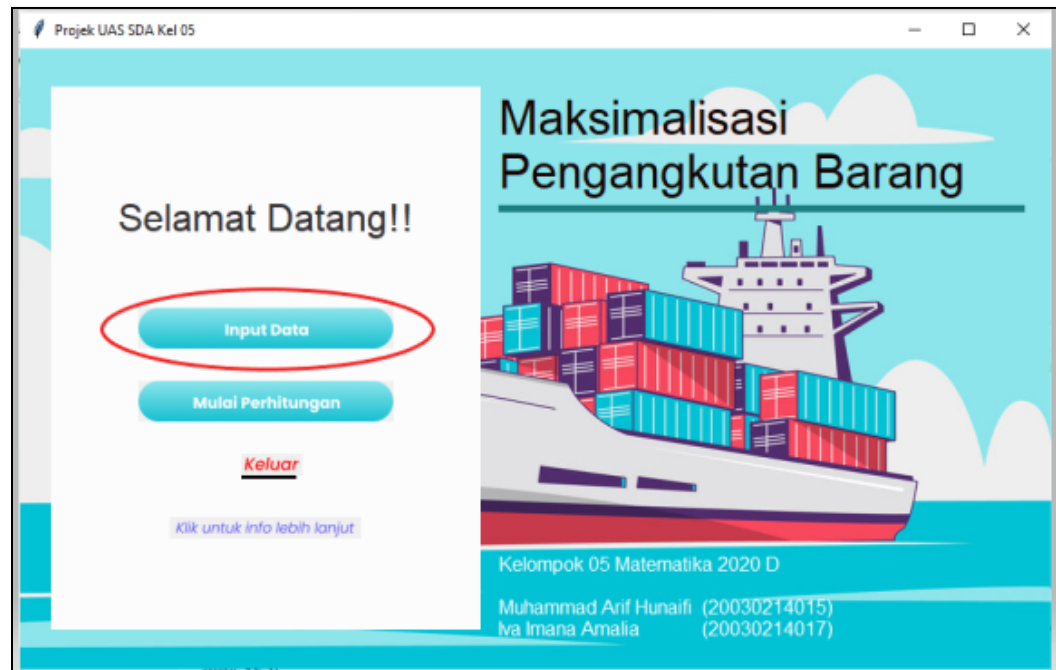
- 1) User menjalankan program “Maksimalisasi Pengangkutan Barang”. Maka tampilan program akan menjadi seperti dibawah.



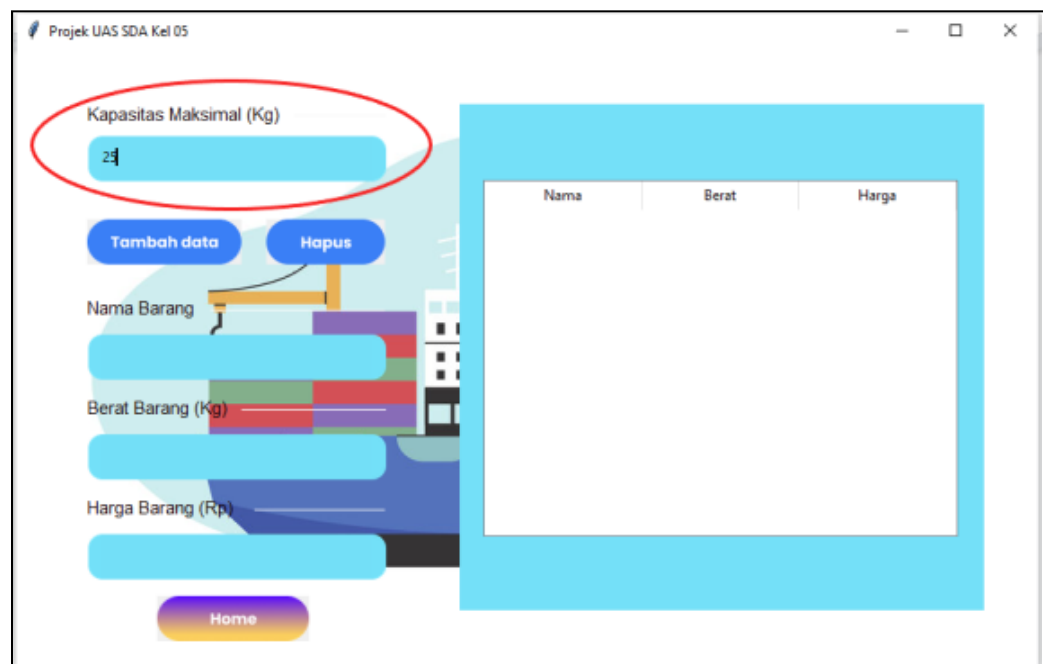
- 2) Langkah selanjutnya, user bisa membaca panduan penggunaan program pada menu “Klik untuk info lebih lanjut” yang berada dibagian bawah agar memudahkan user dalam menjalankan program.



- 3) Jika user sudah memahami penggunaan program ini, langkah berikutnya adalah menginputkan data barang pada menu “Input Data”.



- 4) Sebelum menginputkan data barang, user dapat memasukkan kapasitas maksimal muatan berupa kilogram (kg) pada entry “Kapasitas Maksimal”.



- 5) Setelah itu, user dapat menginputkan data barang berupa nama barang, berat barang, dan harga barang pada entry-entry yang telah tersedia.

Projek UAS SDA Kel 05

Kapasitas Maksimal (Kg)

25

Tambah data Hapus

Nama Barang

Sepatu

Berat Barang (Kg)

4

Harga Barang (Rp)

40000

Home

Nama	Berat	Harga
------	-------	-------

- 6) Lalu pilih tombol “Tambah Data” agar data terinput dalam knapsack berupa tabel.

Projek UAS SDA Kel 05

Kapasitas Maksimal (Kg)

25

Tambah data Hapus

Nama Barang

Sepatu

Berat Barang (Kg)

4

Harga Barang (Rp)

40000

Home

Nama	Berat	Harga
------	-------	-------

7) Data akan terinput pada tabel disebelah kanan.

Projek UAS SDA Kel 05

Kapasitas Maksimal (Kg)

25

Tambah data Hapus

Nama Barang

I

Berat Barang (Kg)

Harga Barang (Rp)

Home

Nama	Berat	Harga
Sepatu	4	400000

8) Kemudian user dapat menginputkan kembali data barang menggunakan langkah yang sama dengan langkah No.5

Projek UAS SDA Kel 05

Kapasitas Maksimal (Kg)

25

Tambah data Hapus

Nama Barang

I

Berat Barang (Kg)

Harga Barang (Rp)

Home

Nama	Berat	Harga
Sepatu	4	400000

- 9) Jika terjadi kesalahan dalam menginputkan data, user dapat menghapus inputan dengan memilih tombol “Hapus”

Projek UAS SDA Kel 05

Kapasitas Maksimal (Kg)

25

Tambah data Hapus

Nama Barang

Berat Barang (Kg)

Harga Barang (Rp)

Home

Nama	Berat	Harga
Sepatu	4	400000

- 10) Setelah semua data barang selesai diinputkan, langkah selanjutnya user memilih menu “Home”

Projek UAS SDA Kel 05

Kapasitas Maksimal (Kg)

25

Tambah data Hapus

Nama Barang

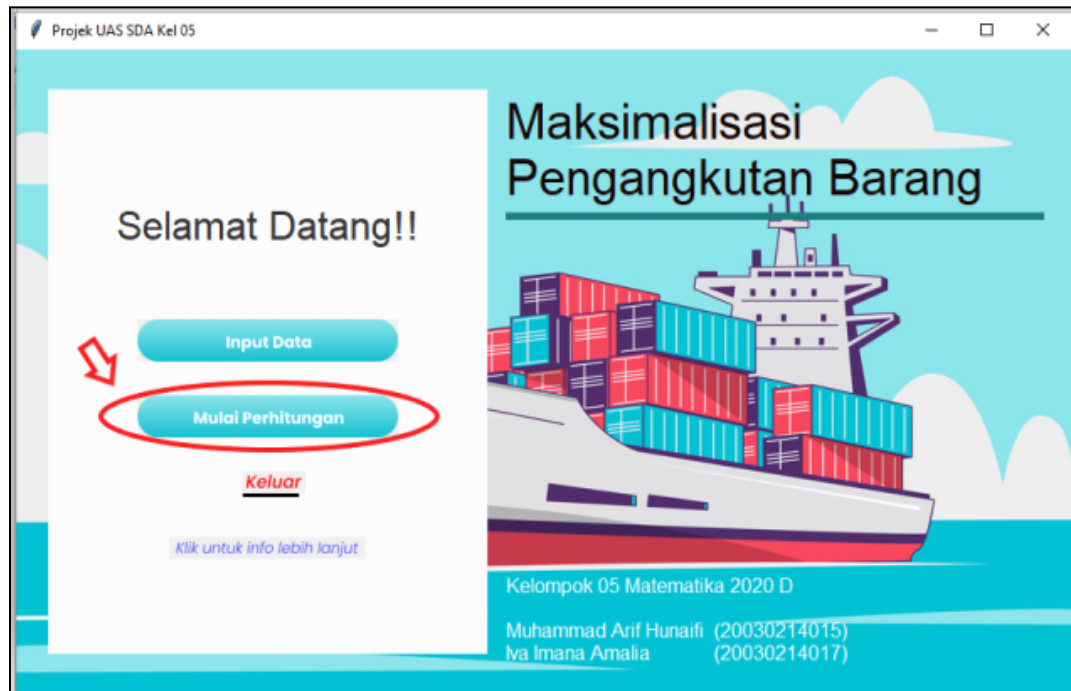
Berat Barang (Kg)

Harga Barang (Rp)

Home

Nama	Berat	Harga
Sepatu	4	400000
Kaos	3	240000
Topi	4	480000
Dasi	1	80000
Celana	5	750000
Sabuk	7	490000
Baju	2	220000
Tas	5	800000

11) User akan dibawa ke halaman awal. Lalu pilih menu “Mulai Perhitungan”



12) User dapat memulai perhitungan dengan memilih 3 algoritma yang telah disediakan, seperti greedy by profit, greedy by weight, atau greedy by density. Hasil perhitungan akan muncul di dalam tabel yang tersedia disebelah kanan.



Projek UAS SDA Kel 05

Jalankan Program

Greedy by Profit

Greedy by Weight

Greedy by Density

Home

Menggunakan Greedy by Weight

Nama	Berat	Harga
Dasi	1	80000
Baju	2	220000
Kaos	3	240000
Sepatu	4	400000
Topi	4	480000
Celana	5	750000
Tas	5	800000

Berat Maksimal	Total Berat	Total Profit
25	24.0	2970000.0

Projek UAS SDA Kel 05

Jalankan Program

Greedy by Profit

Greedy by Weight

Greedy by Density

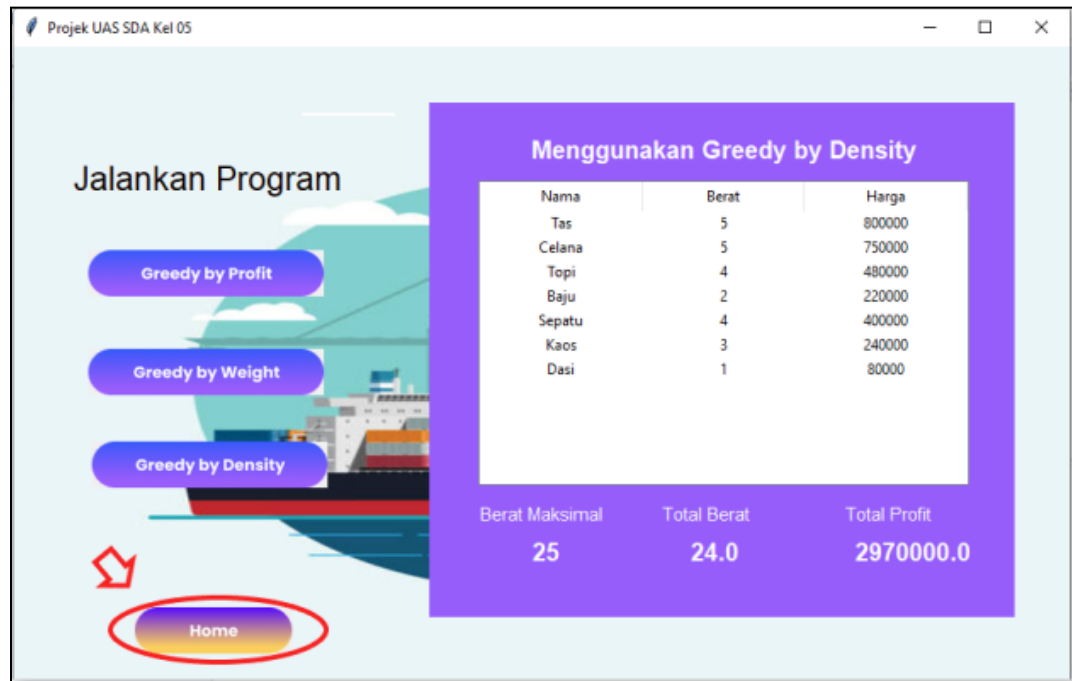
Home

Menggunakan Greedy by Density

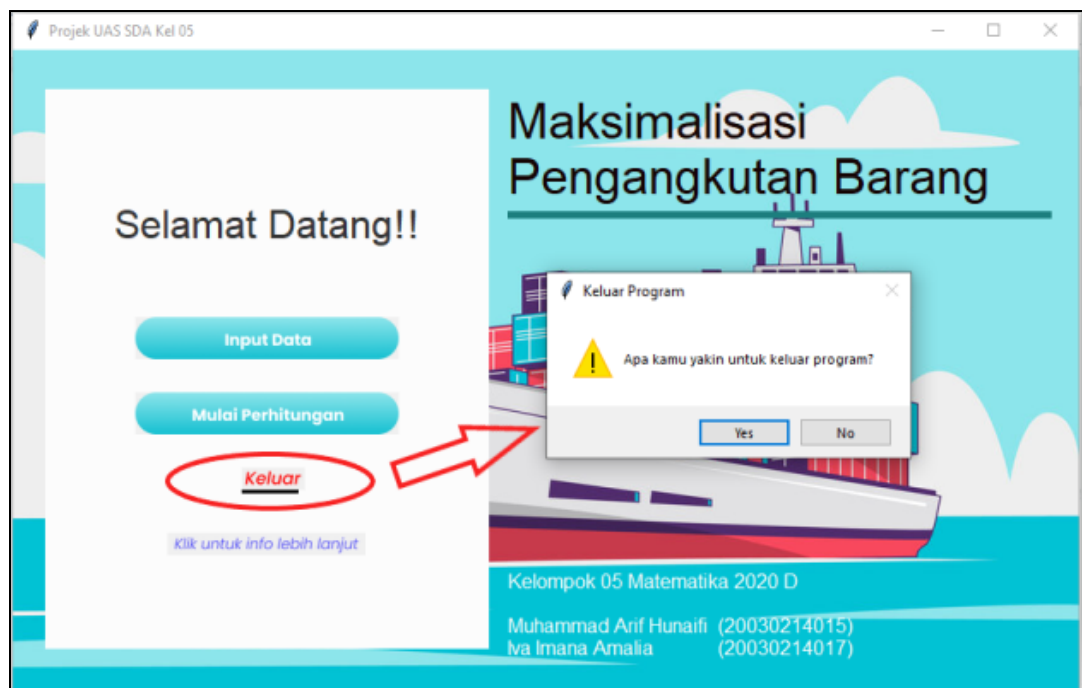
Nama	Berat	Harga
Tas	5	800000
Celana	5	750000
Topi	4	480000
Baju	2	220000
Sepatu	4	400000
Kaos	3	240000
Dasi	1	80000

Berat Maksimal	Total Berat	Total Profit
25	24.0	2970000.0

13) Jika selesai perhitungan, user dapat kembali ke tampilan awal dengan memilih menu “Home”

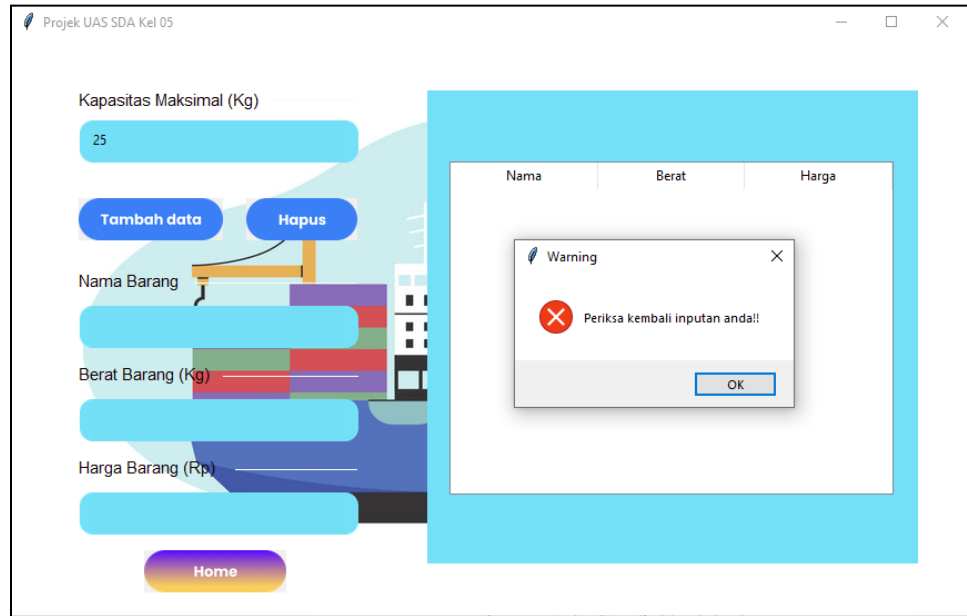


14) Program telah selesai digunakan. User dapat keluar dari program dengan memilih menu “Keluar”

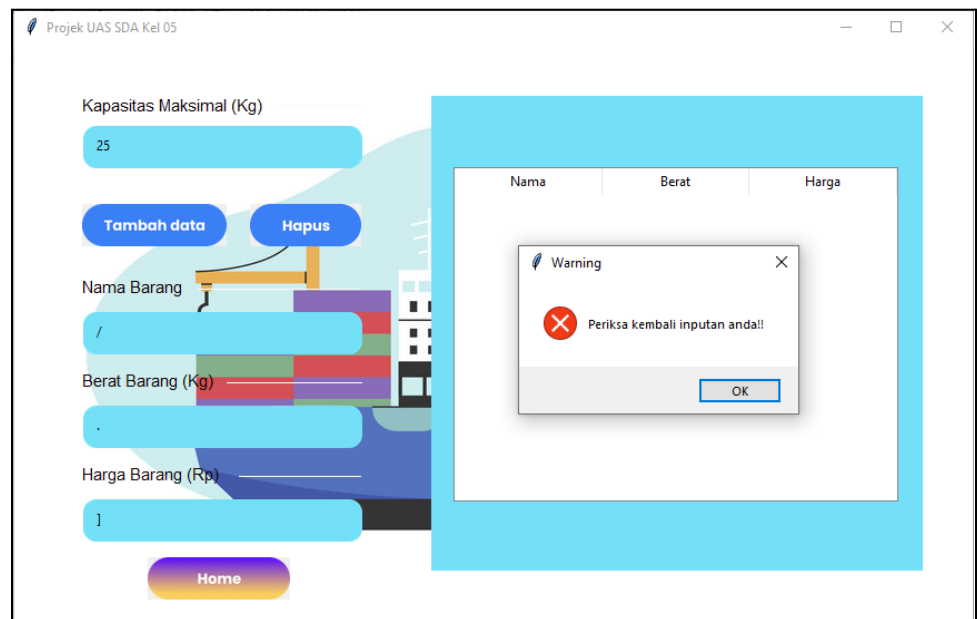


15) Kesalahan-kesalahan/Error yang dapat terjadi saat User menginputkan data.

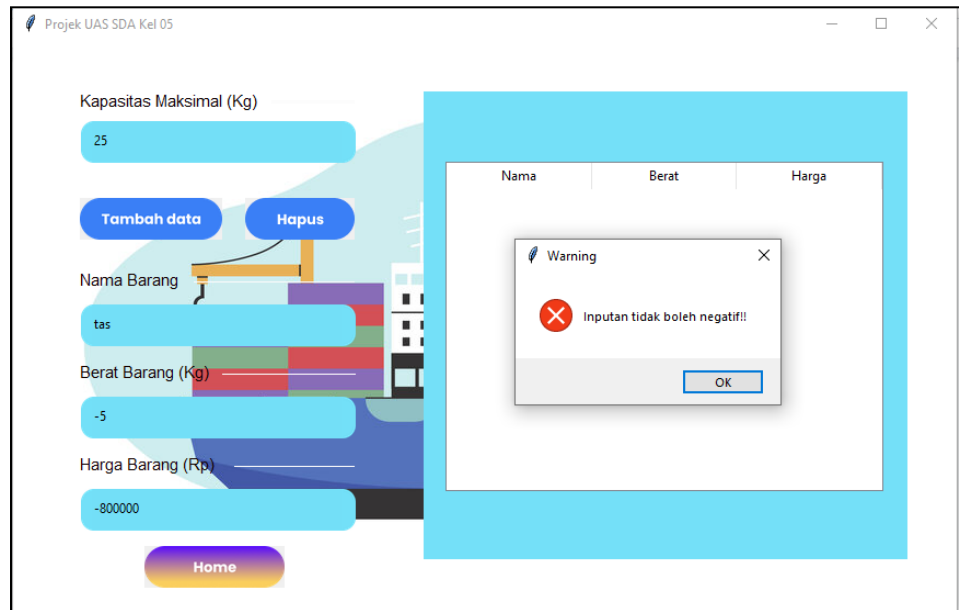
a. Saat user tidak menginputkan apa-apa/ inputan kosong



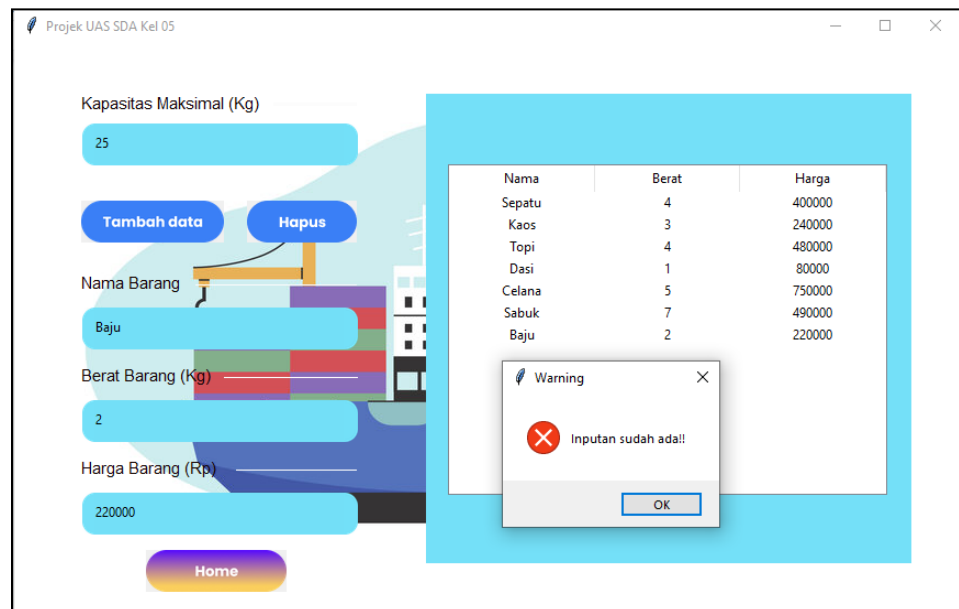
b. Saat user menginputkan simbol



- c. Saat user menginputkan nilai negatif pada berat dan harga barang



- d. Saat user menginputkan barang yang sama lebih dari satu kali



2. Listing Code

- a. Import berbagai macam modul yang dibutuhkan dalam program

```
from itertools import count
from tkinter import *
from tkinter import messagebox
from tkinter import ttk
from PIL import ImageTk, Image
```

```

from tkinter.messagebox import showinfo,askquestion
import pandas
import csv

```

- b. Buat tkinter untuk tampilan GUI program

```

root = Tk()
root.title("Projek UAS SDA Kel 05")
root.geometry("862x519")
root.configure(bg = "#FFFFFF")

```

- c. Membuat multiple frame untuk berpindah menu pada program

```

framehome = Frame(root)
frameinput = Frame(root)
frameoutput = Frame(root)
f4 = Frame(root)
f5 = Frame(root)
for frame in (framehome, frameinput, frameoutput, f4, f5):
    frame.config(bg="PeachPuff2")
    frame.grid(row=0, column=0, sticky='news')

```

- d. Tampilan pada tampilan awal

```

canvashome = Canvas(framehome,bg = "#FFFFFF",height = 519,width = 862,bd
= 0,highlightthickness = 0,relief = "ridge")
imghome = ImageTk.PhotoImage(Image.open("D:/My Downloads/KELOMPOK
5/Gambar/pp.jpg"))
canvashome.create_image(0, 0, anchor=NW, image=imghome)

canvashome.create_rectangle(26.0,32.0,385.0,486.0,fill="#FCFCFC",outline="")
canvashome.create_rectangle(400.0,131.0,840.0,137.0,fill="#1C7E7E",outline="")
)
canvashome.create_rectangle(185.0,339.0,231.0,360.0,fill="#000000",outline="")
canvashome.create_text(82.0,124.0,anchor="nw",text="Selamat
Datang!!",fill="#333333",font=("Roboto Bold", 32 * -1))
canvashome.create_text(400.0,36.0,anchor="nw",text="Maksimalisasi
\nPengangkutan Barang",fill="#150505",font=("Poppins Bold", 40 * -1))
canvashome.create_text(400.0,422.0,anchor="nw",text="Kelompok          05
Matematika 2020 D\n\nMuhammad Arif Hunaifi (20030214015)\nIva Imana
Amalia \
(20030214017)",fill="white",font=("RobotoSerifNormalRoman SemiBold", 16 *
-1))
canvashome.grid(row=0,column=0)

```

e. Tombol yang dibutuhkan pada tampilan awal

```
button_image_keluar = PhotoImage(file="D:/My Downloads/KELOMPOK
5/Gambar/button_1.png")
button_keluar =
Button(framehome,image=button_image_keluar,borderwidth=0,highlightthickness=
0,command=lambda: quit(), relief="flat")
button_keluar.place(x=185.0,y=339.0,width=51.0,height=18.0)

button_image_info = PhotoImage(file="D:/My Downloads/KELOMPOK
5/Gambar/button_2.png")
button_info =
Button(framehome,image=button_image_info,borderwidth=0,highlightthickness=
0,command= lambda : info(),relief="flat")
button_info.place(x=125.0,y=392.0,width=160.0,height=18.0)

button_image_input = PhotoImage(file="D:/My Downloads/KELOMPOK
5/Gambar/button_input.png")
button_input =
Button(framehome,image=button_image_input,borderwidth=0,highlightthickness=
0,command=lambda: show_frame(frameinput),relief="flat")
button_input.place(x=99.0,y=217.0,width=213.0,height=34.0)

button_image_mulai = PhotoImage(file="D:/My Downloads/KELOMPOK
5/Gambar/button_mulai.png")
button_mulai =
Button(framehome,image=button_image_mulai,borderwidth=0,highlightthickness=
0,command=lambda: show_frame(frameoutput),relief="flat")
button_mulai.place(x=99.0,y=278.0,width=213.0,height=34.0)
```

f. Tampilan pada tampilan input

```
canvasinput = Canvas(frameinput,bg = "#FFFFFF",height = 519,width = 862,bd =
0,highlightthickness = 0,relief = "ridge")
canvasinput.place(x = 0, y = 0)
img = ImageTk.PhotoImage(Image.open("D:/My Downloads/KELOMPOK
5/Gambar/abc.jpg"))
canvasinput.create_image(0, 0, anchor=NW, image=img)

canvasinput.create_rectangle(165.0,219.0,312.0,220.0,fill="#FCFCFC",outline="
")
canvasinput.create_rectangle(236.0,54.0,312.0,57.0,fill="#FCFCFC",outline="")
canvasinput.create_rectangle(202.0,387.0,312.0,388.0,fill="#FCFCFC",outline="
")
canvasinput.create_rectangle(191.0,303.0,313.0,304.0,fill="#FCFCFC",outline="
")
```

g. Tombol yang dibutuhkan pada tampilan input

```
button_home = PhotoImage(file="D:/My Downloads/KELOMPOK  
5/Gambar2/button_0.png")  
buttonhome =  
Button(frameinput,image=button_home,borderwidth=0,highlightthickness=0,com  
mand=lambda: show_frame(framehome),relief="flat")  
buttonhome.place(x=120.0,y=460.0,width=128.0,height=38.0)  
  
button_image_tambahdata = PhotoImage(file="D:/My Downloads/KELOMPOK  
5/Gambar2/button_1.png")  
button_tambahdata =  
Button(frameinput,image=button_image_tambahdata,borderwidth=0,highlightthic  
kness=0,command= lambda : tambahdt(),relief="flat")  
button_tambahdata.place(x=61.0,y=143.0,width=130.0,height=38.0)  
  
button_image_hapus = PhotoImage(file="D:/My Downloads/KELOMPOK  
5/Gambar2/button_hapus.png")  
button_hapus =  
Button(frameinput,image=button_image_hapus,borderwidth=0,highlightthickness  
=0,command=lambda: reset(),relief="flat")  
button_hapus.place(x=212.0,y=143.0,width=100.0,height=38.0)  
  
canvasinput.create_rectangle(375.0,46.0,817.0,472.0,fill="#74E0F8",outline="")
```

h. Masukan yang dibutuhkan pada tampilan input

```
entry_image_kapmaks = PhotoImage(file="D:/My Downloads/KELOMPOK  
5/Gambar2/entry_1.png")  
entry_bg_1 =  
canvasinput.create_image(186.5,92.0,image=entry_image_kapmaks)  
entry_kapmaks = Entry(frameinput,bd=0,bg="#73DFF7",highlightthickness=0)  
entry_kapmaks.place(x=73.0,y=73.0,width=227.0,height=36.0)  
  
entry_image_beratbrg = PhotoImage(file="D:/My Downloads/KELOMPOK  
5/Gambar2/entry_2.png")  
entry_bg_2 =  
canvasinput.create_image(186.5,343.0,image=entry_image_beratbrg)  
entry_beratbrg = Entry(frameinput,bd=0,bg="#73DFF7",highlightthickness=0)  
entry_beratbrg.place(x=73.0,y=324.0,width=227.0,height=36.0)  
  
entry_image_hargabrg = PhotoImage(file="D:/My Downloads/KELOMPOK  
5/Gambar2/entry_3.png")  
entry_bg_3 =  
canvasinput.create_image(186.5,427.0,image=entry_image_hargabrg)  
entry_hargabrg = Entry(frameinput,bd=0,bg="#73DFF7",highlightthickness=0)  
entry_hargabrg.place(x=73.0,y=408.0,width=227.0,height=36.0)
```

```

entry_image_namabrg = PhotoImage(file="D:/My Downloads/KELOMPOK
5/Gambar2/entry_4.png")
entry_bg_4
canvasinput.create_image(186.5,259.0,image=entry_image_namabrg)
entry_namabrg = Entry(frameinput,bd=0,bg="#73DFF7",highlightthickness=0)
entry_namabrg.place(x=73.0,y=240.0,width=227.0,height=36.0)

```

i. Tabel data knapsack pada halaman input

```

data_knapsack = ttk.Treeview(frameinput)
data_knapsack['columns'] = ('nama_barang','berat_barang','harga_barang')

data_knapsack.column("#0", width=0, stretch=NO)
data_knapsack.column("nama_barang",anchor=CENTER, width=10)
data_knapsack.column("harga_barang",anchor=CENTER,width=10)
data_knapsack.column("berat_barang",anchor=CENTER,width=10)

data_knapsack.heading("#0",text="",anchor=CENTER)
data_knapsack.heading("nama_barang",text="Nama",anchor=CENTER)
data_knapsack.heading("harga_barang",text="Harga",anchor=CENTER)
data_knapsack.heading("berat_barang",text="Berat",anchor=CENTER)

data_knapsack.place(x=395.0,y=110.0,width=400.0,height=300.0)

canvasinput.create_text(61.0,46.0,anchor="nw",text="Kapasitas           Maksimal
(Kg)",fill="#0C0000",font=("RalewayRoman Bold", 15 * -1))
canvasinput.create_text(61.0,209.0,anchor="nw",text="Nama
Barang",fill="#110000",font=("RalewayRoman Bold", 15 * -1))
canvasinput.create_text(61.0,377.0,anchor="nw",text="Harga           Barang
(Rp)",fill="#150505",font=("RalewayRoman Bold", 15 * -1))
canvasinput.create_text(61.0,293.0,anchor="nw",text="Berat           Barang
(Kg)",fill="#110000",font=("RalewayRoman Bold", 15 * -1))

```

j. Tampilan yang dibutuhkan pada halaman output

```

canvasoutput = Canvas(frameoutput,bg = "#FFFFFF",height = 519,width =
862,bd = 0,highlightthickness = 0,relief = "ridge")
canvasoutput.place(x = 0, y = 0)
image_image_01 = ImageTk.PhotoImage(Image.open("D:/My
Downloads/KELOMPOK 5/Gambar/z.jpg"))
image_01 = canvasoutput.create_image(431.0,259.0,image=image_image_01)

canvasoutput.create_rectangle(236.0,54.0,312.0,57.0,fill="#FCFCFC",outline="")
canvasoutput.create_rectangle(340.0,46.0,817.0,468.0,fill="#965DFB",outline="")
)

```

```

canvasoutput.create_text(381.0,375.0,anchor="nw",text="Berat
Maksimal",fill="#FCFCFC",font=("RalewayRoman Bold", 15 * -1))
canvasoutput.create_text(530.0,375.0,anchor="nw",text="Total
Berat",fill="#FCFCFC",font=("RalewayRoman Bold", 15 * -1))
canvasoutput.create_text(679.0,375.0,anchor="nw",text="Total
Profit",fill="#FCFCFC",font=("RalewayRoman Bold", 15 * -1))
canvasoutput.create_text(50.0,92.0,anchor="nw",text="Jalankan
Program",fill="#0C0000",font=("Poppins Bold", 27 * -1))

```

k. Tombol yang dibutuhkan pada halaman output

```

button_home1 = PhotoImage(file="D:/My Downloads/KELOMPOK
5/Gambar2/button_0.png")
buttonhome1 =
Button(frameoutput,image=button_home1,borderwidth=0,highlightthickness=0,c
ommand=lambda: show_frame(framehome),relief="flat")
buttonhome1.place(x=100.0,y=460.0,width=128.0,height=38.0)

button_image_weight =PhotoImage(file="D:/My Downloads/KELOMPOK
5/Gambar3/button_1.png")
button_weight =
Button(frameoutput,image=button_image_weight,borderwidth=0,highlightthickne
ss=0,command=lambda: gbw(),relief="flat")
button_weight.place(x=62.0,y=248.0,width=192.0,height=38.0)

button_image_profit =PhotoImage(file="D:/My Downloads/KELOMPOK
5/Gambar3/button_2.png")
button_profit =
Button(frameoutput,image=button_image_profit,borderwidth=0,highlightthicknes
s=0,command=lambda: gbp(),relief="flat")
button_profit.place(x=62.0,y=167.0,width=192.0,height=38.0)

button_image_density =PhotoImage(file="D:/My Downloads/KELOMPOK
5/Gambar3/button_3.png")
button_density =
Button(frameoutput,image=button_image_density,borderwidth=0,highlightthickn
ess=0,command=lambda: gbd(),relief="flat")
button_density.place(x=65.0,y=324.0,width=192.0,height=38.0)

```

l. Tabel data hasil Greedy pada halaman output

```

tabeloutput = ttk.Treeview(frameoutput)
tabeloutput['columns'] = ('nama_barang','berat_barang','harga_barang')

tabeloutput.column("#0", width=0, stretch=NO)
tabeloutput.column("nama_barang",anchor=CENTER, width=10)
tabeloutput.column("harga_barang",anchor=CENTER,width=10)

```

```

tabeloutput.column("berat_barang",anchor=CENTER,width=10)

tabeloutput.heading("#0",text="",anchor=CENTER)
tabeloutput.heading("nama_barang",text="Nama",anchor=CENTER)
tabeloutput.heading("harga_barang",text="Harga",anchor=CENTER)
tabeloutput.heading("berat_barang",text="Berat",anchor=CENTER)

tabeloutput.place(x=380.0,y=110.0,width=400.0,height=250.0)

labeljudul= Label(frameoutput, font=("arial" ,15, "bold"),bg="#965DFB",
fg="#FCFCFC")
labeljudul.place(x=420, y=70)
labelharga = Label(frameoutput, font=("arial" ,15, "bold"),bg="#965DFB",
fg="#FCFCFC")
labelharga.place(x=684, y=400)
labelmaks= Label(frameoutput, text="", font=("arial" ,15,
"bold"),bg="#965DFB", fg="#FCFCFC")
labelmaks.place(x=421, y=400)
labeltotalberat = Label(frameoutput, text="", font=("arial" ,15,
"bold"),bg="#965DFB", fg="#FCFCFC")
labeltotalberat.place(x=550, y=400)

```

m. Berbagai macam def yang dibutuhkan pada program

```

def gbp(kapsmaks = (entry_kapmaks.get())):
    labeljudul.config(text="")
    labelharga.config(text="")
    labelmaks.config(text="")
    labeltotalberat.config(text="")

    for row in tabeloutput.get_children():
        tabeloutput.delete(row)
    kapsmaks=int(entry_kapmaks.get())
    barang_ = daftarbarang_()
    print ("\nMenggunakan greedy by harga dengan kapasitas maksimal Knapsack:",
kapsmaks)
    testGreedy(barang_, kapsmaks, harga)
    taken,val,weight=greedy(barang_, kapsmaks,harga)
    for barang in taken:
        tabeloutput.insert(parent="",index='end',iid=barang,text="",values=barang)

    labeljudul.config(text="Menggunakan Greedy by Profit")
    labelharga.config(text=val)
    labelmaks.config(text=entry_kapmaks.get())
    labeltotalberat.config(text=weight)

```

```

def gbw(kapsmaks = (entry_kapmaks.get())):
    labeljudul.config(text="")
    labelharga.config(text="")
    labelmaks.config(text="")
    labeltotalberat.config(text="")

    for row in tabeloutput.get_children():
        tabeloutput.delete(row)
    kapsmaks=int(entry_kapmaks.get())
    barang_ = daftarbarang_()
    testGreedy(barang_, kapsmaks, beratInverse)
    taken,val,weight = greedy(barang_, kapsmaks,beratInverse)
    for barang in taken:
        print (" ",barang)
        tabeloutput.insert(parent="",index='end',iid=barang,text="",values=barang)

    labeljudul.config(text="Menggunakan Greedy by Weight")
    labelharga.config(text=val)
    labelmaks.config(text=entry_kapmaks.get())
    labeltotalberat.config(text=weight)

def gbd(kapsmaks = (entry_kapmaks.get())):
    labeljudul.config(text="")
    labelharga.config(text="")
    labelmaks.config(text="")
    labeltotalberat.config(text="")

    for row in tabeloutput.get_children():
        tabeloutput.delete(row)
    kapsmaks=int(entry_kapmaks.get())
    barang_ = daftarbarang_()
    print ('\nMenggunakan greedy by density dengan kapasitas maksimal
Knapsack:', kapsmaks)
    testGreedy(barang_, kapsmaks, density)
    taken,val,weight = greedy(barang_, kapsmaks,density)
    print("Total harga dari barang \n" + str(val))
    for barang in taken:
        print (" ",barang)
        tabeloutput.insert(parent="",index='end',iid=barang,text="",values=barang)

    labeljudul.config(text="Menggunakan Greedy by Density")
    labelharga.config(text=val)
    labelmaks.config(text=entry_kapmaks.get())
    labeltotalberat.config(text=weight)

```



```

def tambahdt():
    try:
        for row in data_knapsack.get_children():
            data_knapsack.delete(row)
            data = pandas.read_csv("D:/My Downloads/KELOMPOK
5/databarang.csv",sep=',')

            namabarang.clear()
            beratbarang.clear()
            hargabarang.clear()

        for i in data.index :
            namabarang.append(data['namabarang'][i])
            beratbarang.append(data['beratbarang'][i])
            hargabarang.append(data['hargabarang'][i])

            namabrg = str(entry_namabrg.get())
            namabarang.append(namabrg)
            beratbrg = int(entry_beratbrg.get())
            beratbarang.append(beratbrg)
            assert beratbrg > 0
            hargabrg = int(entry_hargabrg.get())
            hargabarang.append(hargabrg)
            assert hargabrg > 0

            with open("D:/My Downloads/KELOMPOK 5/databarang.csv",
mode='w',encoding='UTF8',newline='') as csv_file:
                header=['namabarang','beratbarang','hargabarang']
                writer = csv.writer(csv_file, delimiter=',', quotechar='"',
quoting=csv.QUOTE_MINIMAL)
                writer.writerow(header)
                for i in range(len(hargabarang)) :
                    writer.writerow([namabarang[i],beratbarang[i],hargabarang[i]])

            barang_ = daftarbarang_()
            for barang in barang_:
                print(barang)
                data_knapsack.insert(parent="",index='end',iid=barang,text="",values=barang)

            entry_beratbrg.delete(0,END)
            entry_hargabrg.delete(0,END)
            entry_namabrg.delete(0,END)

```

```

except ValueError:
    messagebox.showerror('Warning','Periksa kembali inputan anda!!')
    barang_ = daftarbarang_()
    for barang in barang_:
        print(barang)
        data_knapsack.insert(parent=",index='end',iid=barang,text=",values=barang)

```

```

except AssertionError:
    messagebox.showerror('Warning','Inputan tidak boleh negatif!!')

```

```

except TclError:
    messagebox.showerror('Warning','Inputan sudah ada!!')
    entry_beratbrg.delete(0,END)
    entry_hargabrg.delete(0,END)
    entry_namabrg.delete(0,END)

```

```

def info():
    messagebox.showinfo('Panduan untuk menggunakan program ini',"
1. User memilih menu "Input Data"

2. User menginputkan nama barang, berat barang, dan
   harga barang

3. Selanjutnya user memilih menu "Tambah Data"
   untuk menambah data barang.

4. User bisa menghapus inputan yang salah
   dengan menekan menu "Hapus"

5. Jika selesai menginputkan data, user dapat klik menu
   "HOME"

6. Pilih menu "Mulai Perhitungan"

7. User bisa menjalankan program sesuai menu greedy
   yang diinginkan.

8. User bisa memilih menu "HOME"
   untuk kembali ke halaman awal

9. Program selesai dijalankan, user bisa menekan menu
   "KELUAR" untuk keluar dari program.")

```

```

def reset():
    namabarang.pop()
    beratbarang.pop()

```

```

hargabarang.pop()

        with open("D:/My Downloads/KELOMPOK 5/databarang.csv",
mode='w',encoding='UTF8',newline='') as csv_file:
            header=['namabarang','beratbarang','hargabarang']
            writer = csv.writer(csv_file, delimiter=',', quotechar='"',
quoting=csv.QUOTE_MINIMAL)
            writer.writerow(header)
            for i in range(len(hargabarang)) :
                writer.writerow([namabarang[i],beratbarang[i],hargabarang[i]])

for row in data_knapsack.get_children():
    data_knapsack.delete(row)
barang_ = daftarbarang_()
for barang in barang_:
    data_knapsack.insert(parent="",index='end',iid=barang,text="",values=barang)

def quit():
    MsgBox = askquestion('Keluar Program','Apa kamu yakin untuk keluar
program?',icon = 'warning')
    if MsgBox == 'yes':
        root.destroy()

def show_frame(frame):
    frame.tkraise()

```

n. Class barang untuk program greedy

```

namabarang = []
beratbarang = []
hargabarang = []

class barang(object):
    def __init__(self, n, v, w):
        self.nama = n
        self.harga = int(v)
        self.berat = int(w)
    def getnama(self):
        return self.nama
    def getharga(self):
        return self.harga
    def getberat(self):
        return self.berat
    def __str__(self):

```

```

        hasil = " + self.nama + ' ' + str(self.berat)\
                + ' ' + str(self.harga) + "
        return hasil

def daftarbarang_():
    barang_ = []
    for i in range(len(hargabarang)):
        barang_.append(barang(namabarang[i], hargabarang[i], beratbarang[i]))
    return barang_

def greedy(barang_, kapsmaks, keyFcn):
    assert type(barang_) == list and kapsmaks >= 0
    barang_Copy = sorted(barang_, key=keyFcn, reverse = True)
    hasil = []
    totalVal = 0.0
    totalberat = 0.0
    i = 0
    while totalberat < kapsmaks and i < len(barang_):
        if (totalberat + barang_Copy[i].getberat()) <= kapsmaks:
            hasil.append((barang_Copy[i]))
            totalberat += barang_Copy[i].getberat()
            totalVal += barang_Copy[i].getharga()
            i += 1
    print(totalberat)
    return (hasil, totalVal, totalberat)

def harga(barang):
    return barang.getharga()
def beratInverse(barang):
    return 1.0/barang.getberat()
def density(barang):
    return barang.getharga()/barang.getberat()

def testGreedy(barang_, constraint, getKey):
    print("Total berat dari barang" )
    taken, val, io = greedy(barang_, constraint, getKey)
    print("Total harga dari barang \n" + str(val))
    print("List barang diperoleh")
    for barang in taken:
        print (" ", barang)

```

```

def testGreedy(kapsmaks = (entry_kapmaks.get())):
    barang_ = daftarbarang_()
    print("\nList barang yang masuk dalam permasalahan knapsack :")
    for barang in barang_:
        print(barang)
    data_knapsack.insert(parent="", index='end', iid=barang, text="", values=barang)

```

o. Tampilan awal setelah program dijalankan

```

with open("D:/My Downloads/KELOMPOK 5/databarang.csv",
mode='r',encoding='UTF8',newline='') as csv_file:
    reader = csv.DictReader(csv_file, delimiter=',')
    for row in data_knapsack.get_children():
        data_knapsack.delete(row)
    for row in reader:
        nmbrg = row['namabarang']
        brtbrg = row['beratbarang']
        hrgbrg = row['hargabarang']
        data_knapsack.insert("", 0, values=(nmbrg,brtbrg,hrnbrg))

show_frame(framehome)
root.mainloop()

```

3. Pembagian Tugas

1. Muhammad Arif Hunaifi (20030214015)

- Kode perhitungan greedy by harga (value)
- Kode perhitungan greedy by density
- Kode tampilan list barang yang masuk dalam masalah knapsack
- Membuat desain tampilan halaman input pada figma
- Membuat desain tampilan halaman output
- Menggabungkan kode greedy yang dibutuhkan pada halaman input
- Menggabungkan file csv dan kodingan
- Membuat treeview untuk menampilkan tabel data halaman input
- Menambahkan hapus, tambah data
- Melengkapi kekurangan pada program

2. Iva Imana Amalia (20030214017)

- Kode perhitungan greedy by weight (berat)
- Membuat kode menampilkan list barang yang masuk dalam masalah knapsack

- c. Menggabungkan ketiga kode greedy dalam satu kodingan
- d. Membuat desain tampilan halaman awal pada figma
- e. Membuat try except untuk inputan pada halaman input
- f. Menggabungkan kode greedy yang dibutuhkan pada halaman output
- g. Membuat file csv
- h. Membuat treeview untuk menampilkan tabel data halaman output
- i. Menambahkan def info, keluar
- j. Melengkapi kekurangan pada program