

# **LAPORAN TUGAS BESAR 1**



## **KELOMPOK 12 (LIFT)**

1. Rahmat Al Fajri	-	16520171
2. Rava Naufal A	-	16520411
3. Ray Clement	-	16520371
4. Rio Alexander	-	16520181

**KU1102  
PENGENALAN KOMPUTASI  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
2020**

# DAFTAR ISI

BAB I: TUGAS 1.....	3
Dekomposisi persoalan .....	3
BAB II: TUGAS 2 .....	4
2.1 Deskripsi Simulasi .....	4
2.2 Flowchart .....	5
2.3 Pseudocode .....	8
2.4 Antarmuka.....	9
BAB III: TUGAS 3 .....	10
3.1 Definisi prosedur untuk menjemput penumpang .....	10
3.2 Definisi fungsi untuk menentukan jumlah penumpang .....	10
3.3 Definisi fungsi untuk menginput lantai tujuan penumpang .....	11
3.4 Definisi prosedur bila sudah sampai ke lantai yang dituju .....	11
3.5 Definisi prosedur mengubah arah gerak lift.....	12
3.6 Definisi prosedur untuk menginput kondisi awal lift.....	12
3.7 Main Program .....	13
BAB IV: KESIMPULAN DAN <i>LESSON LEARNED</i> .....	15
PEMBAGIAN TUGAS .....	15
DAFTAR RFERENSI.....	16

# **BAB I**

## **TUGAS 1**

### **Dekomposisi persoalan**

#### **1. Komponen Lift**

##### **1.1 Luar lift**

- 1.1.1 Tombol *Recall*
- 1.1.2 Tombol Fireman
- 1.1.3 Indikator Lantai
- 1.1.4 Pintu Luar

##### **1.2 Dalam lift**

- 1.2.1 Sensor Berat
- 1.2.2 Tombol Lantai dan Pintu
- 1.2.3 Tombol Emergency dan Interphone
- 1.2.4 Indikator Lantai
- 1.2.5 Pintu Dalam

##### **1.3 Penggerak lift**

- 1.3.1 Mesin
- 1.3.2 Sistem Katrol
- 1.3.3 Tali Beban
- 1.3.4 Rel Lift
- 1.3.5 Buffer

#### **2. Cara Kerja Lift**

- 2.1 Pintu Lift Terbuka
- 2.2 Penumpang masuk
- 2.3 Cek berat badan penumpang
- 2.4 Pintu lift tertutup
- 2.5 Lift bergerak ke lantai yang dituju
- 2.6 Pintu lift terbuka
- 2.7 Penumpang keluar
- 2.8 Pintu lift tertutup

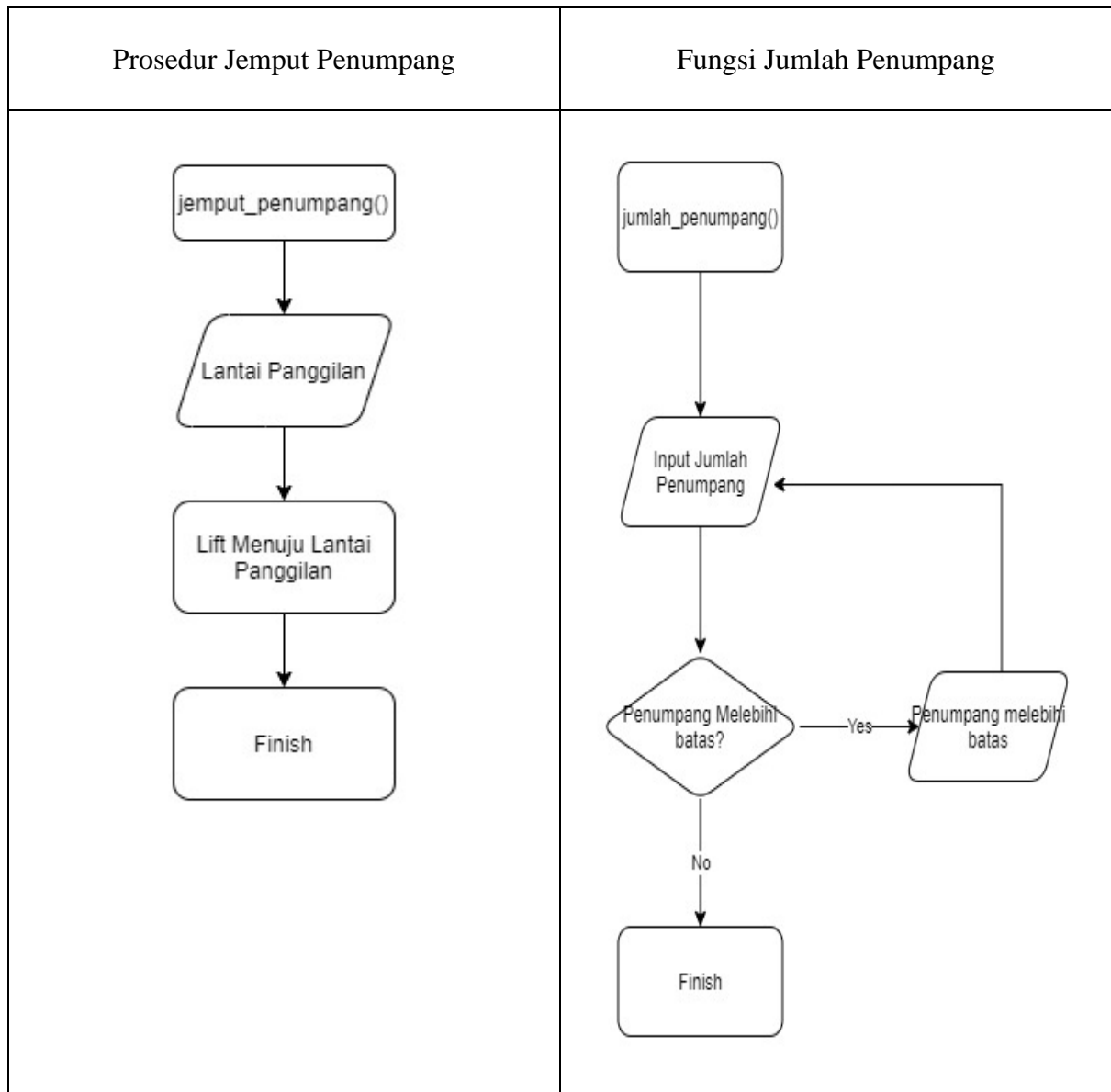
## BAB II

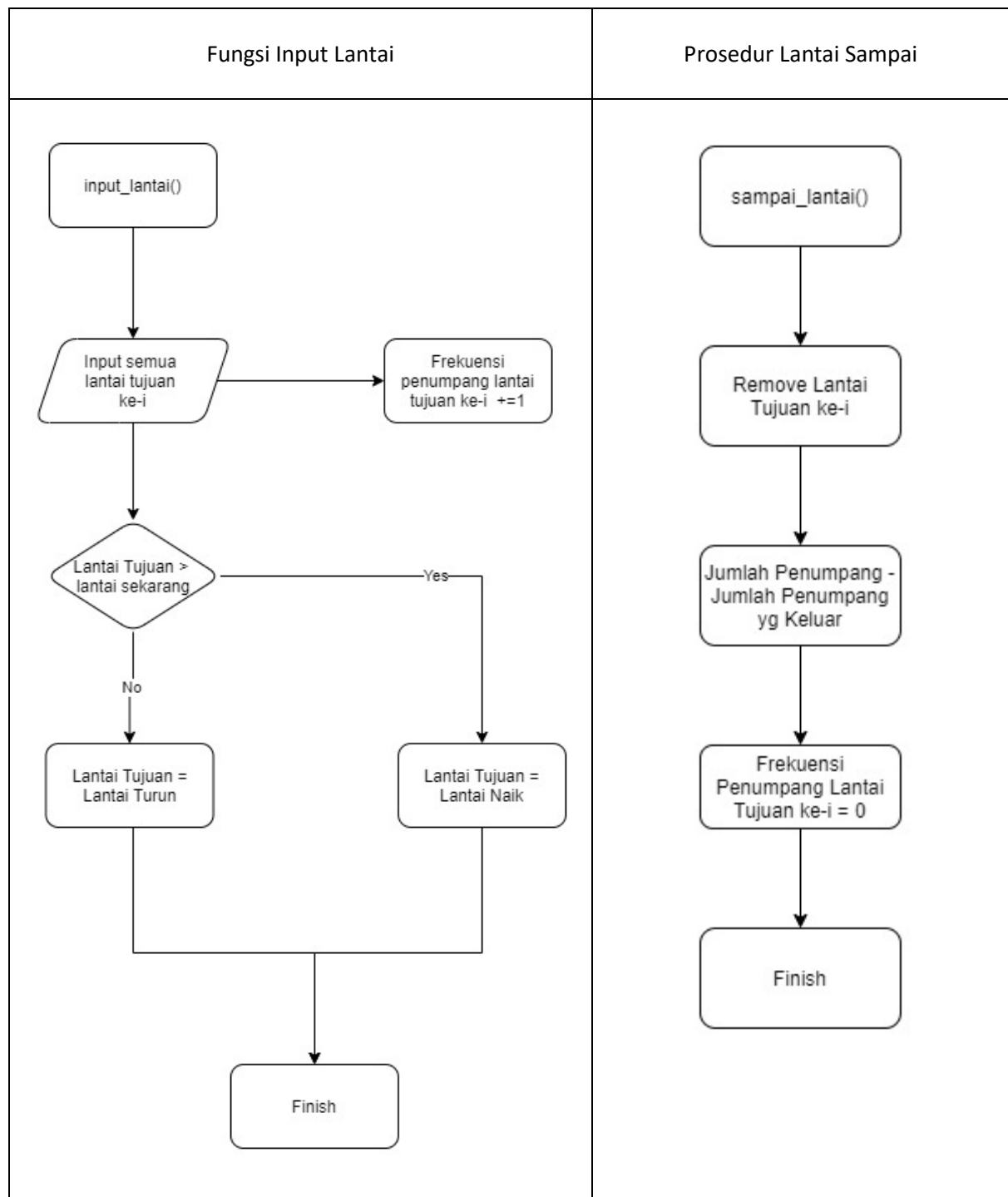
### TUGAS 2

#### 2.1 Deskripsi Simulasi

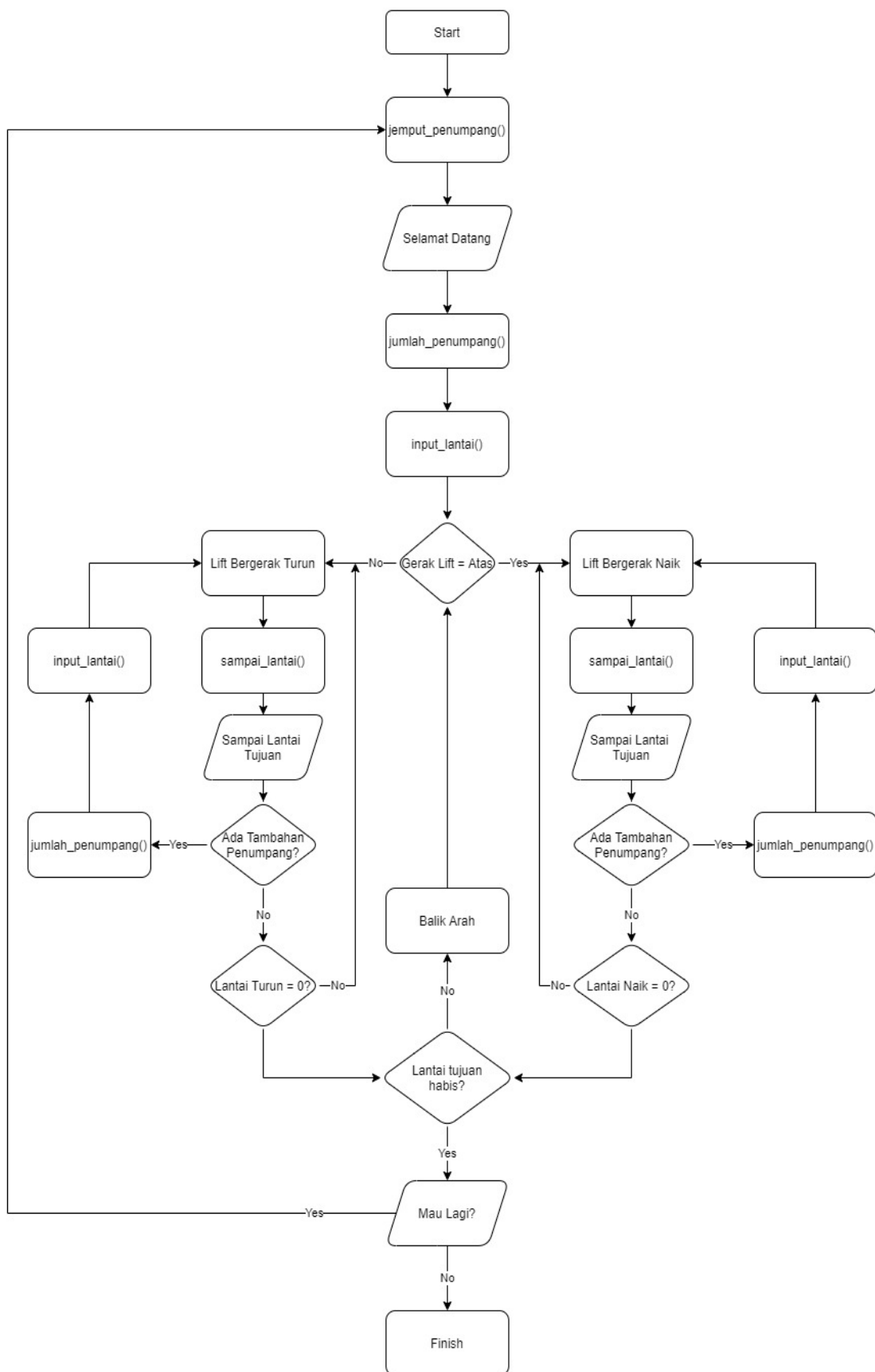
Bagian	Input	Proses	Output	Keterangan
Luar lift	Tombol <i>recall</i>	Lift bergerak ke lantai penginput (Asumsi suatu Gedung dengan 10 lantai)	Lift sampai dan pintu terbuka	Lift akan datang ke lantai penginput dari posisi lift terakhir
Dalam lift	Sensor berat	Jumlah penumpang dihitung. (Diasumsikan setiap penumpang memiliki berat 60 Kg dan kapasitas berat lift maksimal 600 Kg)	Jika jumlah penumpang memenuhi syarat, lift dapat digunakan	<p>Arah gerak lift akan didasarkan pada arah gerak sebelumnya.</p> <p>Urutan pemilihan lantai tujuan akan didasarkan pada arah gerak lift. Jika sebelumnya lift sedang bergerak ke atas, maka lantai tujuan yg berada di atas lantai saat ini akan didahulukan, begitu juga sebaliknya</p> <p>Lift dapat menerima penambahan penumpang saat rombongan penumpang sebelumnya belum mencapai tujuannya masing-masing</p>
	Tombol angka	Menyimpan lantai tujuan penumpang dan lift bergerak ke lantai tujuan	Lift sampai di lantai tujuan dan pintu terbuka	

## 2.2 Flowchart





## Main Program



## 2.3 Pseudocode

```
{Program Simulasi_Lift}

(Lantai Awal Penumpang)

{Jemput Penumpang}
(Jumlah Penumpang)

{Cek Overweight}
while (Jumlah Penumpang > 10) do
    overweight = True
    (Jumlah Penumpang)

{Lantai Tujuan}
i traversal [1..Jumlah Penumpang]
    (Lantai tujuan orang ke-(i))

{Lift berjalan ke lantai tujuan terdekat, beberapa penumpang keluar}

{Tambahan Penumpang}
if (Tambahan Penumpang = True) then
    (Banyak Tambahan)
    i traversal [1..Banyak Tambahan]
        (Lantai tujuan orang ke-(i))

{Lift kembali berjalan ke lantai tujuan dan menerima tambahan penumpang,
serta kembali berjalan sampai penumpang habis.}

(Mau lagi?)

if (Mau lagi? = True) then
    {Ulangi ke awal program, lift menjemput penumpang dari posisi
    terakhirnya}
else {Mau lagi? = False}
    {Selesai}
```



## 2.4 Antarmuka

```
C:\Users\rrioau\PycharmProjects\AiTiBi\venv
---Selamat Datang---
Lantai Penumpang: 5
1 ↑
2 ↑
3 ↑
4 ↑
----Lantai 5-----

Masukkan jumlah penumpang : 2
Masukkan lantai tujuan orang ke-1: 3
Masukkan lantai tujuan orang ke-2: 8

5 ↑
6 ↑
7 ↑
----Lantai 8 sisa 1 orang----
[3]
Apakah ada tambahan orang? (Y/N) Y
Masukkan jumlah penumpang : 2
Masukkan lantai tujuan orang ke-1: 2
Masukkan lantai tujuan orang ke-2: 3

7 ↓
6 ↓
5 ↓
4 ↓
----Lantai 3 sisa 1 orang----
[2]
Apakah ada tambahan orang? (Y/N)

----Lantai 2 sisa 0 orang----
[]
Apakah ada tambahan orang? (Y/N)
```

```
----Lantai 2 sisa 0 orang----
[]
Apakah ada tambahan orang? (Y/N)

~Selamat Jalan~
Mau Lagi? (Y/N) Y

Lantai Penumpang: 8
2 ↑
3 ↑
4 ↑
5 ↑
6 ↑
7 ↑
----Lantai 8-----

Masukkan jumlah penumpang : 1
Masukkan lantai tujuan orang ke-1: 1

7 ↓
6 ↓
5 ↓
4 ↓
3 ↓
2 ↓
----Lantai 1 sisa 0 orang----
[]
Apakah ada tambahan orang? (Y/N)

~Selamat Jalan~
Mau Lagi? (Y/N)

Process finished with exit code 0
```

## BAB III

### TUGAS 3

#### 3.1 Definisi prosedur untuk menjemput penumpang

```
def jemput_penumpang():
    global lantai_sekarang
    global gerak_lift

    j = int(input("Lantai Penumpang: "))

    for i in range(lantai_sekarang, j + 1):
        if i == j:
            print(f"----Lantai {i}-----")
            lantai_sekarang = j
            return
        else:
            print(f"{i} ↑")

    for i in range(lantai_sekarang, j - 1, -1):
        if i == j:
            print(f"----Lantai {i}-----")
            lantai_sekarang = j
            return
        else:
            print(f"{i} ↓")
```

#### 3.2 Definisi fungsi untuk menentukan jumlah penumpang

```
def jumlah_penumpang():
    # KAMUS Lokal
    # n      : int untuk menyimpan jumlah penumpang yang masuk
    # Asumsi berat tiap penumpang adalah 60 Kg
    # Asumsi kapasitas berat maksimum lift adalah 600 Kg

    # Algoritma
    while True:
        n = int(input("Masukkan jumlah penumpang : "))

        if n + total_penumpang > 10:
            print("Penumpang melebihi kapasitas")
        else: # n <= 10
            return n + total_penumpang
```

### 3.3 Definisi fungsi untuk menginput lantai tujuan penumpang

```
def input_lantai(n):
    global gerak_lift
    global total_penumpang

    lantai_temp = []

    for i in range(n):
        j = int(input(f"Masukkan lantai tujuan orang ke-{i + 1}: "))
        lantai_temp.append(j)
        if j == lantai_sekarang:
            lantai_penumpang[j - 1] -= 1
            total_penumpang -= 1
            lantai_penumpang[j - 1] += 1

    for i in lantai_temp:
        if not (i in lantai) and i != lantai_sekarang:
            if i > lantai_sekarang:
                lantai_naik.append(i)
            elif i < lantai_sekarang:
                lantai_turun.append(i)
            lantai.append(i)

    lantai.sort()

    if gerak_lift: # True = naik
        return lantai_naik
    else:
        return lantai_turun
```

### 3.4 Definisi prosedur bila sudah sampai ke lantai yang dituju

```
def sampai_lantai(n):
    # KAMUS Lokal:
    # n : int untuk menyimpan lantai sekarang

    # Algoritma
    global total_penumpang
    global lantai_penumpang

    lantai.remove(n)
    lantai_tujuan.remove(n)
    total_penumpang -= lantai_penumpang[n - 1]
    lantai_penumpang[n - 1] = 0
```

### 3.5 Definisi prosedur mengubah arah gerak lift

```
def balik_arah():  
    # Algoritma  
    global lantai_tujuan  
    global gerak_lift  
    global lantai_sekarang  
  
    if gerak_lift:  
        lantai_sekarang -= 1  
        lantai_tujuan = lantai_turun  
    else:  
        lantai_sekarang += 1  
        lantai_tujuan = lantai_naik  
  
    gerak_lift = not gerak_lift
```

### 3.6 Definisi prosedur untuk menginput kondisi awal lift

```
def input_awal():  
    # Algoritma  
    global total_penumpang  
    global lantai_sekarang  
    global lantai_tujuan  
    global gerak_lift  
  
    jemput_penumpang()  
    print()  
    total_penumpang = jumlah_penumpang()  
    lantai_tujuan = input_lantai(total_penumpang)  
  
    if lantai_tujuan == []:  
        balik_arah()  
    print()
```

### 3.7 Main Program

```
print("---Selamat Datang---")

gerak_lift = True
lantai = []
lantai_naik = []
lantai_turun = []
lantai_sekarang = 1

total_penumpang = 0
lantai_penumpang = [0 for i in range(10)] # Tabel frekuensi tujuan penumpang
temp_penumpang = 0

input_awal()

while True:
    # Lift mulai bergerak
    while total_penumpang!=0:

        if gerak_lift:
            # Jika lift bergerak ke atas
            if lantai_sekarang in lantai_tujuan:
                # Jika lantai sekarang adalah lantai yang dituju
                sampai_lantai(lantai_sekarang)

                # Output user
                print(f'---Lantai {lantai_sekarang} sisa {total_penumpang} orang---')
                total_penumpang -= lantai_penumpang[lantai_sekarang - 1]
                print(lantai)

            # Tambahan penumpang
            if input("Apakah ada tambahan orang? (Y/N) ").lower() == "y":
                # Jika ada tambahan penumpang
                temp_penumpang = total_penumpang
                total_penumpang = jumlah_penumpang()
                lantai_tujuan = input_lantai(total_penumpang - temp_penumpang)
                print()

            else:
                # Jika lantai sekarang bukan lantai yang dituju
                print(f"{lantai_sekarang} ↑")

            if lantai_tujuan == []:
                # Jika lantai tujuan sudah kosong (habis)
                balik_arah()
                break

            # Lantai bergerak naik
            lantai_sekarang += 1

        elif not gerak_lift:
            # Jika lantai bergerak ke bawah
            if lantai_sekarang in lantai_tujuan:
                # Jika lantai sekarang adalah lantai yang dituju
                sampai_lantai(lantai_sekarang)

                # Output user
                print(f'---Lantai {lantai_sekarang} sisa {total_penumpang} orang---')
                total_penumpang -= lantai_penumpang[lantai_sekarang - 1]
                print(lantai)
```

```

# Tambahkan penumpang
if input("Apakah ada tambahan orang? (Y/N) ").lower() == "y":
    # Jika ada tambahan penumpang
    temp_penumpang = total_penumpang
    total_penumpang = jumlah_penumpang()
    lantai_tujuan = input_lantai(total_penumpang - temp_penumpang)
    print()

else:
    # Jika lantai sekarang bukan lantai yang dituju
    print(f"{lantai_sekarang} ↓")

if lantai_tujuan == []:
    # Jika lantai yang dituju sudah kosong (habis)
    balik_arah()
    break

# Lantai bergerak turun
lantai_sekarang -= 1

# Ketika semua penumpang telah keluar
if total_penumpang == 0:
    print("~Selamat Jalan~")

    if input("Mau Lagi? (Y/N) ").lower() == "y":
        # Jika ada penumpang baru yang masuk
        print()
        if gerak_lift:
            lantai_sekarang -= 1
        else:
            lantai_sekarang += 1

        input_awal()
    else:
        # Keluar Program
        break

```

## **BAB IV**

### **KESIMPULAN DAN *LESSON LEARNED***

1. Algoritma thinking dan computational thinking sangat membantuk kami dalam membedah konsep pada suatu system tertentu
2. Algoritma cara kerja lift tidak semudah yang kami bayangkan
3. Python merupakan bahasa pemrograman yang sederhana dalam penggunaannya
4. Penggunaan definisi fungsi dan prosedur dalam pyhton sangat membantu untuk mempersingkat algortima

### **PEMBAGIAN TUGAS**

Pada tugas besar kali ini, kami semua berpartisi dalam pengerjaannya dan mengerjakan setiap komponen yang ada bersama-sama. Beberapa komponen tersebut kami adakan *leader* sebagai penentu keputusan dan pembimbing dalam pengerjaannya, yaitu:

#### ***TUGAS 2:***

- Ray Clement
- Rahmat Al Fajri

#### ***TUGAS 3:***

- Rio Alexander

#### ***LAPORAN, PPT, dan TEKNIKAL VIDEO:***

- Rava Naufal A

## **DAFTAR RFERENSI**

<https://softwarerecs.stackexchange.com/questions/32612/gui-drag-drop-style-gui-builder-for-python-tkinter>

<http://engineeringbuilding.blogspot.com/2011/06/lift-pada-gedung-bertingkat.html>