

LAPORAN PRAKTIKUM

# PEMROGRAMAN KOMPUTER (PYTHON)



Disusun Oleh:  
Rafli Pratama  
H1101241008

PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS TANJUNGPURA PONTIANAK  
2024

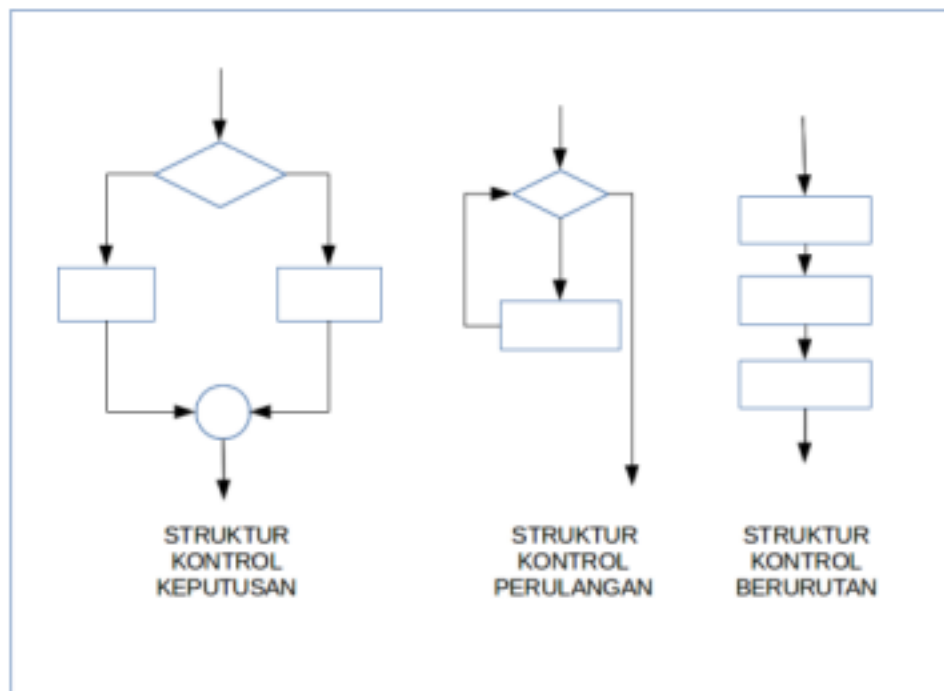
# Praktikum 3

## Sequential, Selection

### 1.1. Dasar Teori

Struktur kontrol di dalam pemrograman adalah kumpulan perintah dengan bentuk tertentu yang digunakan untuk melakukan pengontrolan terhadap jalannya program. Pada dasarnya struktur kontrol terdapat beberapa macam diantaranya adalah:

1. Struktur kontrol berurutan (Sequence Control Structure)
2. Struktur kontrol keputusan/percabangan/kondisi (Selection Control Structure)
3. Struktur kontrol pengulangan (Iteration/ Repetition Control Structure)



#### 1.1.1 Instruksi Runtutan (Sequential)

Sequence pada struktur algoritma adalah sebuah instruksi yang ada dalam algoritma dan diproses secara beruntun selangkah demi selangkah dari awal hingga akhir. Pada struktur dasar tahap runtunan ini juga berlaku pada bahasa pemrograman. Hal ini dapat ditinjau dari adanya tindakan dari komputer pada saat diberikan instruksi melalui bahasa pemrograman yang ditulis di komputer. Kemudian sistem di komputer akan melakukan proses penerjemahan secara runtun mulai dari awal hingga rangkaian yang terakhir tanpa ada loncatan.

Instruksi runtunan adalah instruksi yang dikerjakan secara beruntun atau berurutan baris per-baris mulai dari baris pertama hingga baris terakhir, tanpa ada loncatan atau pengulangan.

- a. Tiap instruksi dikerjakan sekali satu per-satu

- b. Urutan pelaksanaan instruksi sama dengan urutan penulisan algoritma
- c. Instruksi terakhir merupakan akhir dari algoritma
- d. Urutan penulisan instruksi bisa menjadi penting, bila diubah dapat menyebabkan hasil yang berbeda.

Berikut ini adalah contoh algoritma sequential

#### 1. Algoritma Sequential 1

```
{ menunjukkan urutan yang berbeda memberi hasil yang berbeda } Deklarasi
integer A, B;
Deskripsi
A  $\square$  6;
A  $\square$  3 * A;
B  $\square$  A;
write(B);
```

algoritma diatas menghasilkan tampilan = 18

#### 2. Algoritma Sequential 2

```
{ algoritma untuk menghitung luas sebuah segitiga }
Deklarasi
real Alas, Tinggi, Luas;
Deskripsi
write ("Masukkan panjang alasnya : ");
read (Alas);
write ("Masukkan tingginya : ");
read (Tinggi);
Luas  $\square$  Alas * Tinggi / 2;
write ("Luas segitiga = ", Luas);
```

#### 3. Algoritma Sequential 3

```
{ algoritma menampilkan gaji bersih pegawai }
Deklarasi
string nama;
real gajipokok, tunjangan, pajak;
real gajibersih;
write ("Masukkan nama pegawai : ");
```

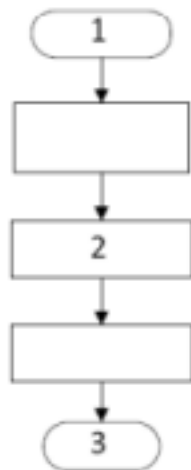
```

read (nama);
write ("Masukkan gaji pokoknya : ");
read (gajipokok);
tunjangan  $\square$  0.25 * gajipokok;
pajak  $\square$  0.15 * (gajipokok + tunjangan);
gajibersih  $\square$  gajipokok + tunjangan - pajak;
write ("Gaji saudara : ", nama);
write ("adalah = ", gajibersih);

```

Suatu struktur program dimana setiap baris program akan dikerjakan secara urut dari atas ke bawah sesuai dengan urutan penulisannya.

Dari flowchart disamping mula-mula pemroses akan melaksanakan instruksi baris program 1, instruksi baris program 2 akan dikerjakan jika instruksi baris program 1 telah selesai dikerjakan. Selanjutnya instruksi baris program 3 dikerjakan setelah instruksi baris program 2 selesai dikerjakan. Setelah instruksi baris program 3 selesai dilaksanakan maka algoritma



berhenti.

### 1.2.1 Instruksi Pemilihan (Selection)

digunakan untuk mengambil suatu keputusan atau memilih bagian program yang akan dikerjakan sesuai dengan kondisi atau syarat yang diberikan. atau melakukan Percabangan yaitu suatu pemilihan statemen yang akan dieksekusi dimana pemilihan tersebut didasarkan atas kondisi tertentu. Salah satu kondisi yang umum digunakan adalah berupa keadaan benar atau salah (true or false).

Dalam bahasa indonesia, if berarti "jika" dan then artinya "maka". Kondisi adalah persyaratan yang dapat bernilai salah atau benar. Aksi hanya dilakukan jika kondisi bernilai benar. Perhatikan kata yang digarisbawahi, if dan then merupakan kata kunci(keywords) untuk struktur pemilihan ini. Dalam kehidupan sehari-hari, kita sering menuliskan pernyataan tindakan bila suatu persyaratan dipenuhi.. Biasanya sebuah pernyataan pemilihan terdiri dari operand-operand yang dihubungkan dengan operator relasi dan digabungkan dengan operator logika.

Contohnya :

- $7 = 7$  (Bernilai benar, sebab 7 sama dengan 7)
- $5 = 9$  (Bernilai salah, sebab 5 tidak sama dengan 9)

- $4 > 2$  (Bernilai benar, sebab 4 lebih besar dari pada 2)
- $3 \neq 8$  (Bernilai benar, sebab 3 tidak sama dengan 8)
- $X = 10$  (Dapat bernilai benar atau salah, tergantung isi variabel X)
- $(X > 3) \text{ And } (Y < 12)$  (Dapat bernilai benar atau salah, tergantung isi variabel X dan Y)

Struktur pemilihan dalam penulisan program diimplementasikan dengan instruksi IF. Ada kalanya sebuah instruksi dikerjakan jika kondisi tertentu dipenuhi.

Penulisan pemilihan secara umum :

<p><b>If kondisi then</b></p> <p>Aksi</p>
---

Dalam bahasa Indonesia, if berarti “jika” dan then artinya “maka”. Kondisi adalah persyaratan yang dapat bernilai salah atau benar. Aksi hanya dilakukan jika kondisi bernilai benar. Perhatikan kata yang digarisbawahi, if dan then merupakan kata kunci (keywords) untuk struktur pemilihan ini. Dalam kehidupan sehari-hari, kita sering menuliskan pernyataan tindakan bila suatu persyaratan dipenuhi. Misalnya

**If hari sedang hujan then**

Budi tidak pergi bermain bola

**If Yanto pergi ke pasar then**

Yanto membeli Ikan

Struktur pemilihan if-then hanya memberikan satu pilihan aksi jika kondisi dipenuhi (bernilai benar), dan tidak memberi pilihan aksi lain jika bernilai salah. Bentuk pemilihan yang lebih umum ialah memilih satu dari dua buah aksi bergantung pada nilai kondisinya :

**If kondisi then**

Aksi 1

**Else**

Aksi 2

Else artinya “kalau tidak”. Bila kondisi bernilai benar, aksi 1 akan dikerjakan, tetapi kalau tidak, aksi 2 yang akan dikerjakan. Misalnya pada pernyataan berikut:

If hari ini cuaca cerah then

Yanto bermain sepakbola

Else

Yanto bermain di rumah saja

Jika kondisi “hari cerah” bernilai benar, maka aksi “Yanto bermain sepakbola” dilakukan, sebaliknya aksi “Yanto bermain di rumah” akan dilakukan jika “hari cerah” tidak benar.

Ada dua bentuk instruksi pemilihan yang sering digunakan yaitu:

Ada dua bentuk instruksi pemilihan yang sering digunakan yaitu:

#### 1. Instruksi if/then/else

Dalam struktur kondisi if.....else minimal terdapat dua pernyataan. Jika kondisi yang diperiksa bernilai benar atau terpenuhi maka pernyataan pertama yang dilaksanakan dan jika kondisi yang diperiksa bernilai salah maka pernyataan yang kedua yang dilaksanakan.

Dalam struktur kondisi if.....elif...else minimal terdapat 3 pernyataan. Pada struktur ini setiap kondisi diperiksa, jika kondisi tersebut benar maka pernyataan yang berada di ruang lingkungannya yang dijalankan, namun bila semua kondisi tidak benar maka pernyataan pada bagian terakhir yang dijalankan.

#### 2. Instruksi case

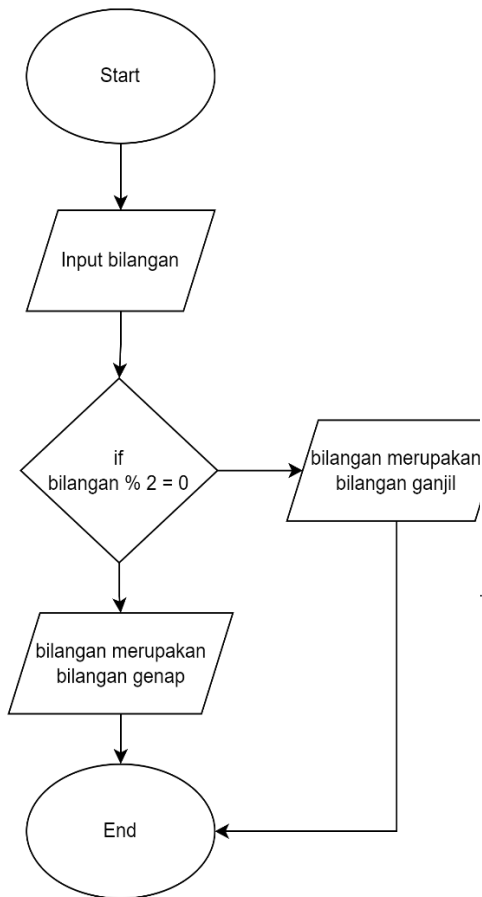
Suatu struktur kontrol switch case berguna untuk variabel yang akan diuji kesamaannya terhadap daftar nilai. Setiap dari nilai-nilai tersebut disebut sebagai case, dan pemilihan dari suatu aksi akan bergantung dari persyaratan yang ada pada switch. Namun pada Python tidak menyediakan "switch" atau "case" seperti bahasa pemrograman lain. Tapi kita bisa menggunakan statemen if, elif yang bisa menggantikan "switch" atau "case".

## 1.2. Percobaan Praktikum

### 1.2.1 Soal 1

Buatlah program untuk menentukan bilangan genap atau bilangan ganjil

Menentukan bilangan ganjil genap



```
#Menentukan bilangan genap dan ganjil
ulang = True
```

```
while ulang == True :
```

```
    bil = int(input('Masukkan Angka : '))
    print('-----')
    print(f'Bilangan : {bil}')
```

```
    if bil % 2 == 0 :
        print(f'Bilangan {bil} merupakan bilangan genap')
    else:
        print(f'Bilangan {bil} merupakan bilangan ganjil')
```

```
    ulang = bool(input('Ulangi ("YA") / Jangan Ulangi (Enter)'))
    print('-----')
    print('Terima Kasih')
```

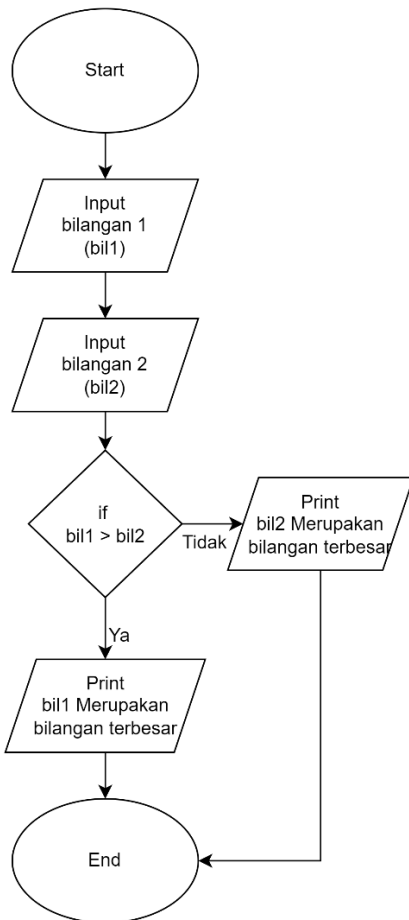
```
-----
Bilangan : 5
Bilangan 5 merupakan bilangan ganjil
-----
Terima Kasih
```

Pada program tersebut user di minta untuk memasukkan sebuah inputan bertipe data integer yang akan disimpan dalam variabel bil, lalu akan di lakukan pengkondisian apakah bil mod 2 = 0, jika iya maka print bil merupakan bilangan genap, jika tidak maka print bil merupakan bilangan ganjil

### 1.2.2 Soal 2

Buatlah program untuk menerima 2 bilangan bulat kemudian menetapkan bilangan yang terbesar

Menentukan bilangan terbesar  
dari 2 bilangan



```
#Menentukan bilangan terbesar
bil1 = int(input('Masukkan Bilangan pertama : '))
bil2 = int(input('Masukkan Bilangan kedua : '))

if bil1 > bil2 :
    print(f'{bil1} Merupakan bilangan terbesar!')
else :
    print(f'{bil2} Merupakan bilangan terbesar!')
```

✓ 5.9s

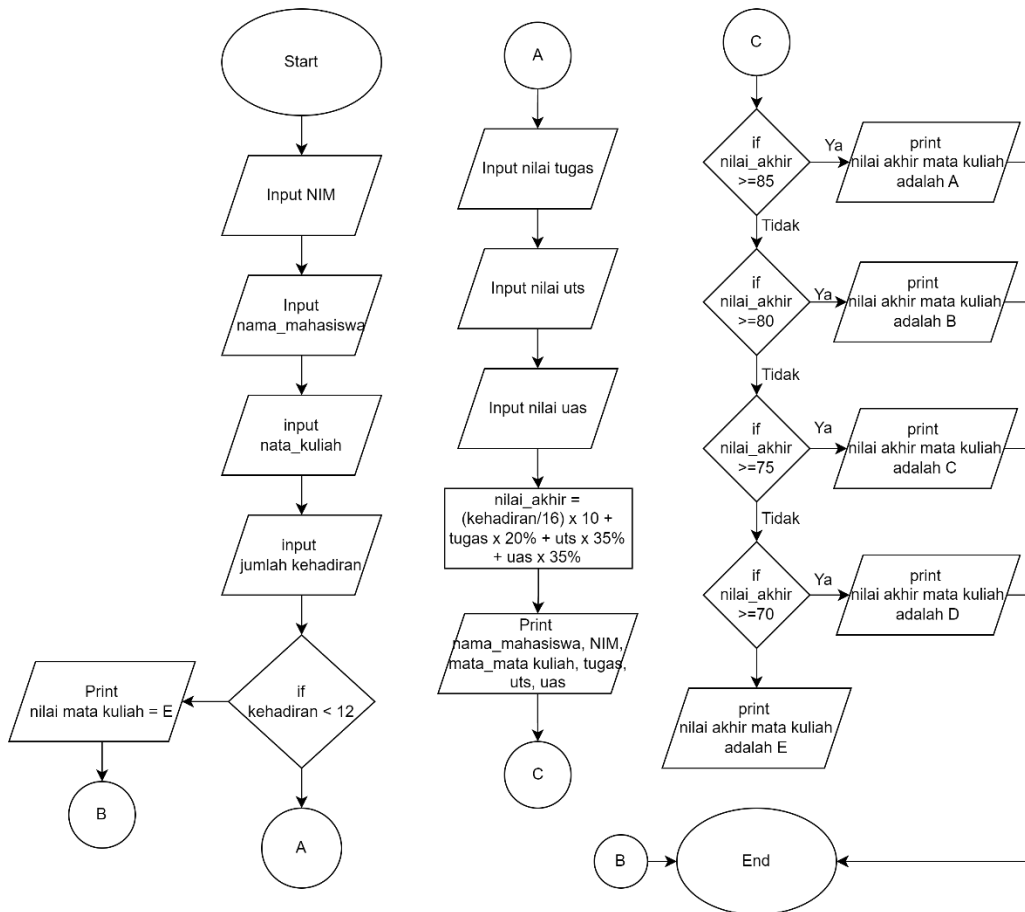
3 Merupakan bilangan terbesar!

Pada program tersebut user akan diminta memasukkan 2 buah inputan bertipe data integer yang akan di simpan pada variabel bil1 dan bil2, jika bil1 lebih besar daripada bil2 maka output nya adalah bil1 merupakan bilangan terbesar, jika tidak maka outputnya adalah bil2 merupakan bilangan terbesar

### 1.2.3 Soal 3

Buatlah program dengan input nim, nama, mata kuliah, kehadiran, tugas, uts, uas dengan mengeluarkan tampilan nim, nama, mata kuliah, kehadiran, tugas, uts, uas serta nilai akhirnya dan skala A, B, C, D, dan E sesuai dengan kebijakan akademik.  $\text{NilaiAkhir} = (\text{abs}/16) * 10 + \text{tgs} * 20\% + \text{uts} * 35\% + \text{uas} * 35\%$





## Kode pemograman :

```

# Menentukan Nilai Matakuliah
nim = input('Masukkan NIM Mahasiswa : ')
nama_mhs = input('Masukkan nama mahasiswa : ')
matkul = input('Masukkan nama mata kuliah : ')
kehadiran = int(input('Masukkan Jumlah kehadiran dari 16 pertemuan'))
while kehadiran > 16 :
    print('Jumlah kehadiran salah, masukkan kehadiran kembali')
    kehadiran = int(input('Masukkan Jumlah kehadiran dari 16 pertemuan'))

if kehadiran < 12 :
    print(f'Nama mahasiswa : {nama_mhs} \nNIM : {nim} \nMata Kuliah : {matkul} \nJumlah kehadiran : {kehadiran}')
    print(f'Nilai mata kuliah {matkul} adalah : E')
else:
    tugas = int(input('Masukkan nilai tugas mata kuliah : '))
    uts = int(input('Masukkan nilai uts : '))
    uas = int(input('Masukkan nilai uas : '))

    nilai_akhir = (kehadiran/16) * 10 + tugas*0.2 + uts * 0.35 + uas * 0.35

    if nilai_akhir >=85 :
        nilai = 'A'
    elif nilai_akhir >= 80:
        nilai= 'B'
    elif nilai_akhir >= 75:
        nilai= 'C'
    elif nilai_akhir >= 70:
        nilai = 'D'
    else :
        nilai = 'E'

    print(f'Nama mahasiswa : {nama_mhs} \nNIM : {nim} \nMata Kuliah : {matkul} \nJumlah kehadiran : {kehadiran} \nNilai tugas : {tugas} \nNilai UTS : {uts} \nNilai UAS : {uas}')
    print(f'Total Nilai mata kuliah {matkul} adalah : {nilai_akhir} \nNilai akhir mata kuliah {matkul} Adalah {nilai}')
  
```

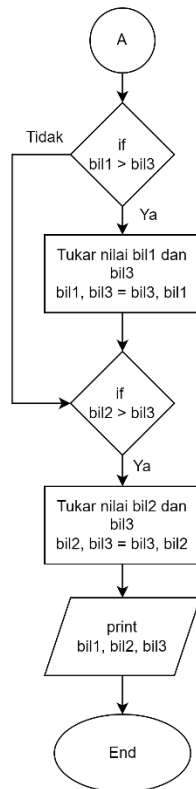
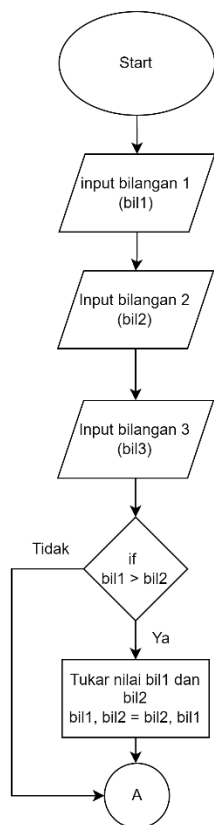
#### Output :

```
Nama mahasiswa : Rafli Pratama
NIM : H1101241008
Mata Kuliah : Pemograman Komputer
Jumlah kehadiran : 16
Nilai tugas : 100
Nilai UTS : 95
Nilai UAS : 95
Total Nilai mata kuliah Pemograman Komputer adalah : 96.5
Nilai akhir mata kuliah Pemograman Komputer Adalah A
```

Pada program tersebut, user akan di minta menginputkan nim, nama mahasiswa, mata kuliah, kehadiran dari 16 pertemuan, jika pertemuan kurang dari 12 maka nilai mata kuliah tersebut di pastikan E, jika tidak maka program akan berlanjut, user akan di minta memasukkan nilai tugas, uts, dan uas, lalu setelah itu di lakukan perhitungan nilai akhir dengan rumus  $(\text{kehadiran}/16) \times 10 + \text{tugas} \times 20\% + \text{uts} \times 35\% + \text{uas} \times 35\%$ , jika nilai akhir lebih dari sama dengan 85 maka nilai mata kuliah A, jika  $\geq 80$  nilainya B, jika  $>75$  maka nilai nya C, jika  $\geq 70$  nilainya D, dan selain dari itu adalah E, lalu pada output akan di tampilkan nama mahasiswa, nim, mata kuliah, kehadiran, nilai tugas, uts, dan uas lalu nilai akhir dari mata kuliah tersebut

#### 1.2.4 Soal 4

Buatlah program untuk menerima 3 bilangan bulat kemudian bentuk bersusun dari bilangan terbesar ke terkecil



```

# Mengurutkan bilangan terkecil hingga terbesar
bil1 = int(input('Masukkan bilangan pertama : '))
bil2 = int(input('Masukkan bilangan kedua : '))
bil3 = int(input('Masukkan bilangan ketiga : '))

if bil1 > bil2 :
    bil1, bil2 = bil2, bil1
if bil1 > bil3 :
    bil1, bil3 = bil3, bil1
if bil2 > bil3 :
    bil2, bil3 = bil3, bil2
print(f'Urutan bilangan dari terkecil ke terbesar adalah {bil1}, {bil2}, {bil3}')
  
```

Urutan bilangan dari terkecil ke terbesar adalah 2, 1, 3  
 Urutan bilangan dari terkecil ke terbesar adalah 1, 2, 3  
 Urutan bilangan dari terkecil ke terbesar adalah 1, 2, 3

Pada program ini, user akan di minta memasukkan 3 inputan bertipe integer yang masing masing akan di simpan pada variabel bil1, bil2, bil3 lalu akan di lakukan pengecekan pertama, apakah bil1 > bil2 jika iya maka tukar nilai bil1 dengan bil2, lalu pengecekan kedua apakah bil1 > bil3, jika iya maka tukar nilai bil1 dengan bil3, lalu pengecekan terakhir apakah bil2 > bil3 jika iya maka tukar nilai bil2 dengan bil3, lalu print bil1, bil2, bil3

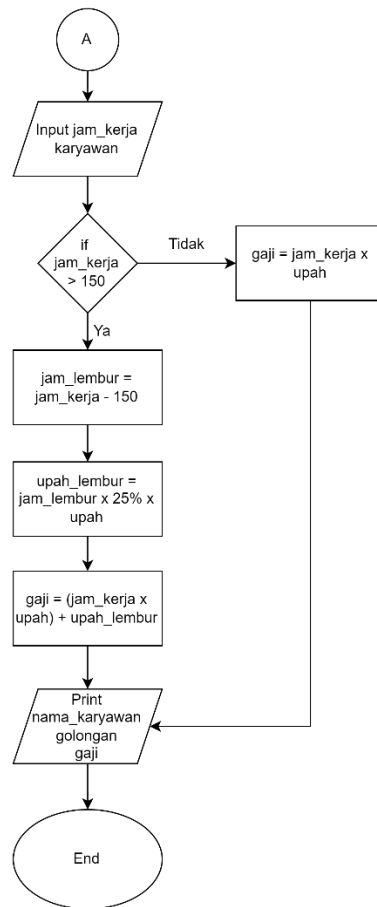
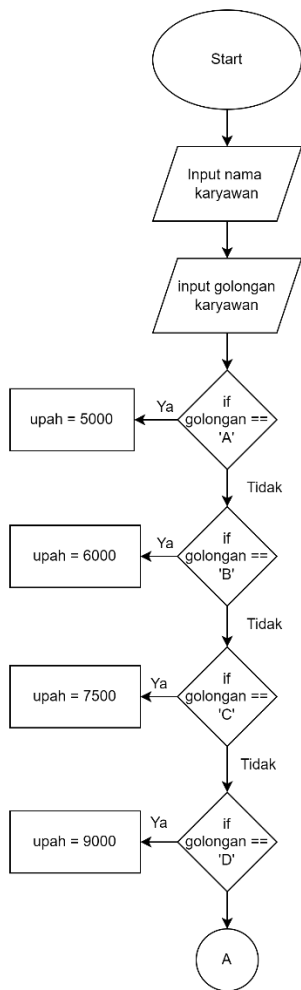
### 1.2.5 Soal 5

Gaji karyawan pada sebuah perusahaan didasarkan pada jam-kerja dalam satu bulan serta posisi atau golongannya dalam perusahaan itu. Upah per jam menurut golongan adalah sebagai berikut:

Golongan	Upah/Jam (Rp)
A	5000
B	6000
C	7500
D	9000

Apabila karyawan bekerja lebih dari 150 jam perminggu, maka kelebihan jam kerja tersebut dihitung sebagai lembur dengan upah/jam 25% diatas upah reguler. Buat sebuah algoritma yang

menerima nama, golongan, serta jam-kerja karyawan, kemudian menampilkan gaji total-nya dalam satu minggu



```

# Menentukan Gaji karyawan
nama_karyawan = input('Masukkan nama karyawan: ')
golongan = input('Masukkan Golongan karyawan: ').upper()

while golongan not in ["A", "B", "C", "D"]:
    print("Masukkan golongan yang benar!")
    golongan = input('Masukkan Golongan karyawan: ').upper()

if golongan == "A":
    upah = 5000
elif golongan == "B":
    upah = 6000
elif golongan == "C":
    upah = 7500
elif golongan == "D":
    upah = 9000

jam_kerja = int(input('Masukkan jam kerja karyawan : '))
if jam_kerja > 150 :
    jam_lembur = jam_kerja - 150
    upah_lembur = jam_lembur * 0.25 * upah
    gaji = (jam_kerja * upah) + upah_lembur
else :
    gaji = jam_kerja * upah

print('-----')
print(f>Nama karyawan : {nama_karyawan}')
print(f>Golongan : {golongan}')
print(f>Jam kerja : {jam_kerja} Jam')
print(f>Gaji karyawan sebesar : Rp.{int(gaji)}')

```

```

-----
Nama karyawan : Rafli Pratama
Golongan : D
Jam kerja : 170 Jam
Gaji karyawan sebesar : Rp.1575000

```

pada program ini user akan diminta memasukkan nama karyawan beserta golongan karyawan, jika golongan karyawan == 'A' maka inisialisasi upah = 5000, jika golongan karyawan == 'B' maka inisialisasi upah = 6000, jika golongan karyawan == 'C' maka inisialisasi upah = 7500, jika golongan karyawan == 'D' maka inisialisasi upah = 9000. Lalu user di minta memasukkan jam kerja karyawan, jika jam kerja lebih dari 150 maka waktu lebih akan di anggap sebagai jam lembur, dan upah lembur = jam lembur 25% dari upah/jam, setelah itu menghitung gaji dengan rumus upah x jam kerja (jika tidak ada lembur) dan upah x jam kerja + upah lembur (jika ada jam lembur). Lalu print kembali nama karyawan, golongan, jam kerja, dan gaji karyawan.

## 1.3. Kesimpulan dan Saran

### 1.3.1. Kesimpulan

Dari praktikum pemograman komputer mengenai Sequential dan Selection ini dapat disimpulkan :

1. Struktur control pada pemograman ada 3 yaitu Sequential, Selection, dan Iteration
2. Pada intruksi runtutan, kode porgram di eksekusi dari atas ke bawah, atau dari awal ke akhir
3. **If/Then/Else**: Memeriksa kondisi dan menjalankan instruksi berdasarkan benar atau salahnya kondisi tersebut. Digunakan untuk membuat percabangan sederhana maupun kompleks.

### 1.3.2. Saran

Saran terhadap praktikum ini yaitu:

1. Hindari percabangan yang terlalu dalam (nested) karena dapat menyulitkan debugging dan memperlambat performa program.
2. Gunakan **if/elif/else** untuk percabangan kompleks di Python
3. perhatikan urutan instruksi karena perubahan kecil dapat menghasilkan output yang berbeda.

# DAFTAR PUSTAKA

1. <https://id.scribd.com/document/523667443/4-Instruksi-Pemilihan-Selection>
2. Modul *Perkuliahan dan Praktikum Algoritma dan Pemrograman* Oleh Ilhamsyah, S.Si., M.Cs.