

LAPORAN PRAKTIKUM

PEMROGRAMAN KOMPUTER (PYTHON)



Disusun Oleh:
Rafli Pratama

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

Praktikum 2

TIPE DATA, VARIABEL DAN INPUT OUTPUT

1.1. Dasar Teori

1.1.1 Tipe Data

Tipe data merupakan atribut yang berkaitan dengan data yang akan memberi tahu sistem komputer. Sehingga nantinya bisa menafsirkan nilai dari data tersebut. Secara khusus, tipe data adalah format penyimpanan data. Data bisa dalam bentuk variabel untuk tipe data tertentu

Tipe data dibagi menjadi beberapa bagian yakni

a. Integer

Integer merupakan tipe data bilangan bulat baik itu bilangan bulat positif maupun bilangan bulat negatif tanpa desimal. Contohnya adalah 8, -3, 0, 4

b. Float atau Real

Merupakan tipe data bilangan pecahan. Semua bilangan yang mengandung tanda desimal tergolong dalam float. Tanda desimal yang dipakai adalah tanda titik, bukan tanda koma. Sebagai contohnya, nilai 0,5 ditulis menjadi 0.5. Tipe ini juga mendukung penulisan dalam bentuk ilmiah, misalnya 2.64×10^{-2} artinya 2.64×10^{-2} . Berikut ini merupakan tabel contoh-contoh tipe data bilangan real:

5.6
3.4
3.67566

c. Char

Merupakan tipe data karakter. Semua data yang hanya terdiri dari 1 karakter tergolong dalam tipe ini. Misalnya data jenis kelamin yang hanya diisikan huruf L. atau P. Penulisan data tipe char harus diapit oleh tanda petik tunggal. Karakter-karakter yang diperbolehkan terdefinisi dalam tabel ASCII. Berikut ini merupakan tabel contoh-contoh tipe data char: "b", "?"

d. String

Merupakan tipe data kalimat. Semua data yang terdiri dari 1 karakter atau lebih dapat digolongkan ke dalam tipe ini. Syaratnya sama dengan tipe char, yaitu harus diapit oleh tanda petik tunggal. Berikut ini merupakan tabel contoh-contoh tipe data string "mahasiswa"

e. Boolean

Merupakan tipe data yang hanya mempunyai nilai True atau False, Penulisan TRUE ataupun FALSE tidak membedakan huruf kapital ataupun non-kapital. Hanya saja penulisannya tidak boleh disingkat menjadi huruf T atau huruf F saja. Berikut ini merupakan tabel contoh-contoh tipe data boolean True False

```
Print( 9 < 10) // True
```

```
Print (10 < 9 ) //False
```

1.1.2 Konstanta

Konstanta adalah nilai yang tidak dapat diubah pada waktu program dijalankan dan tidak berubah selama masa pakai program

Variabel yang mempunyai nilai yang sifatnya tidak bisa diubah, nilai ditentukan pada saat pendefinisian.

Misal:

<nama konstanta> <nilai>

Phi = 3.14;

Konstanta merupakan suatu nilai yang telah ditetapkan di awal pembuatan algoritma dan nilainya tidak dapat diubah oleh proses dalam algoritma. Cara mendefinisikan konstanta adalah dengan menambahkan kata kunci const diawal nama konstanta dan diletakkan di bagian deklarasi

1.1.3 Variabel dan Nilai

Variabel adalah suatu pengenalan (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variabel bisa diubah-ubah sesuai kebutuhan. Variabel ialah tempat penyimpanan untuk menyimpan data. Variabel bersifat mutable yaitu nilai yang ada di dalam variabel dapat diubah. Nilai yang ada di dalam variabel dapat berupa string maupun integer. Adapun literals merupakan simbol yang dapat digunakan untuk mengisi suatu variabel.

Nama dari suatu variabel dapat ditentukan sendiri oleh pemrogram dengan aturan sebagai berikut:

- Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf Bahasa python bersifat case-sensitive artinya huruf besar dan kecil dianggap berbeda. Jadi antara nim, NIM dan Nim dianggap berbeda.
- Tidak boleh mengandung spasi. Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah (underscore).
- Yang termasuk simbol khusus yang tidak diperbolehkan antara lain: \$, %, #, !, &, *, ., -, +, = dsb.
- Panjangnya bebas, tetapi hanya 32 karakter pertama yang terpakai.
 - Contoh variabel yang benar firstName, Alamat, Nilai_Ujian
 - Contoh variabel yang salah: 4XYZ, IP rata, Var:-4na ,458,

1.1.4 Pemberian Nilai

Penugasan atau Assignment merupakan pemberian nilai ke variabel secara langsung. Notasi yang digunakan adalah Nilai yang dapat diberikan adalah tetapan, peubah, ekspresi, maupun nilai yang dihasilkan oleh fungsi. Syarat penugasan adalah nilai yang diberikan harus sesuai dengan tipe variabel. Apabila tipenya tidak sama, maka berlaku tipe yang lebih luas dapat menampung tipe yang lebih sempit. Tipe integer dapat ditampung oleh tipe real, sebaliknya tipe real tidak dapat ditampung oleh tipe integer. Begitu pula dengan string dan char, char dapat diberikan ke string, namun tidak sebaliknya.

1.1.5 Menampilkan Nilai

Didalam setiap bahasa pemrograman tentunya ada fungsi fungsi yang dapat digunakan untuk menampilkan sebuah nilai, seperti di java dengan fungsi println, javascript dengan fungsi console.log(), c++ dengan fungsi cout, nah pada bahasa pemrograman Python kita akan menggunakan fungsi print (), dalam fungsi print kita bisa menampilkan tipe data apapun, variabel yang memiliki nilai bahkan hasil dari operasi

Contoh :

- `print('Hello World') // Hello World`
- `print(2 + 2) // 4`
- `print (2 < 1) //True`

1.2.6 Operator dan Operand

Operator adalah simbol-simbol khusus yang merepresentasikan komputasi seperti penambahan dan perkalian. Nilai yang digunakan oleh operator, kemudian disebut sebagai operand. Berikut adalah ekspresi ekspresi yang benar dalam Python.

Operator ialah simbol yang digunakan untuk mengubah nilai dari variabel dengan melibatkan satu. atau lebih variabel dan nilai. Terdapat beberapa operator aritmatika yang dapat digunakan seperti penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), dan lain-lain. Berikut ini adalah contoh penggunaan operator aritmatika

a. Operator Logika

Operator logika digunakan untuk menggabungkan atau mennegasi kondisi-kondisi dalam sebuah ekspresi. Dalam Python, terdapat tiga operator logika utama:

- **and:** Mengembalikan nilai True jika kedua kondisi yang dihubungkan oleh and bernilai True. Jika salah satu atau keduanya bernilai False, maka hasil keseluruhannya adalah False. Contoh: `x > 8 and x < 20` akan bernilai True jika nilai x berada di antara 8 dan 20.
- **or:** Mengembalikan nilai True jika salah satu atau kedua kondisi yang dihubungkan oleh or bernilai True. Hanya jika kedua kondisi bernilai False, maka hasil keseluruhannya adalah False. Contoh: `x % 2 == 0 or x % 3 == 0` akan bernilai True jika nilai x habis dibagi 2 atau habis dibagi 3.
- **not:** Membalikkan nilai kebenaran dari sebuah kondisi. Jika kondisi awalnya True, maka setelah diaplikasikan not menjadi False, dan sebaliknya. Contoh: `not (x == y)` akan bernilai True jika x tidak sama dengan y.

b. Operator Modulus

Operator modulus, dilambangkan dengan simbol %, digunakan untuk mencari sisa pembagian dari dua bilangan bulat. Hasil dari operasi modulus adalah sisa pembagian yang selalu bernilai lebih kecil dari pembagi

1.2.7 Aturan Pada Operasi

a. Aturan Presedensi Operator dalam Python

Ketika sebuah ekspresi Python mengandung lebih dari satu operator, urutan di mana operasi-operasi tersebut dilakukan sangat penting. Urutan ini ditentukan oleh presedensi operator. Presedensi operator adalah aturan

yang menentukan operasi mana yang akan dijalankan terlebih dahulu. Aturan ini mirip dengan aturan matematika yang telah kita pelajari sebelumnya.

Hierarki Presedensi:

1. Kurung : Segala sesuatu di dalam kurung akan dihitung terlebih dahulu, terlepas dari jenis operator yang ada di dalamnya. Ini seperti memberi penekanan pada bagian tertentu dari ekspresi.
 - o Contoh: $(2 + 3) * 4$ akan menghasilkan 20, karena penjumlahan dalam kurung dihitung terlebih dahulu.
2. Pangkat : Operasi pangkat memiliki prioritas lebih tinggi dibandingkan perkalian, pembagian, penjumlahan, atau pengurangan.
 - o Contoh: $2 ** 3 + 1$ akan menghasilkan 9, karena pemangkatan dilakukan terlebih dahulu.
3. Perkalian dan Pembagian: Operasi perkalian (*) dan pembagian (/) memiliki prioritas yang sama. Jika dalam satu ekspresi terdapat beberapa perkalian dan pembagian, maka operasi akan dilakukan dari kiri ke kanan.
 - o Contoh: $6 / 2 * 3$ akan menghasilkan 9, karena pembagian dilakukan terlebih dahulu.
4. Penjumlahan dan Pengurangan: Operasi penjumlahan (+) dan pengurangan (-) memiliki prioritas yang sama. Sama seperti perkalian dan pembagian, jika terdapat beberapa penjumlahan dan pengurangan, maka operasi akan dilakukan dari kiri ke kanan.
 - o Contoh: $5 - 3 + 2$ akan menghasilkan 4.

Berikut adalah Operator-operator pada bahasa pemrograman python :

1. Operator Aritmatika

Operator	Deskripsi	Contoh	Hasil
*	Perkalian	$8 * 5$	40
/	Pembagian	$8 / 5$	1
%	Modulus	$8 \% 5$	3
+	Penjumlahan	$8 + 5$	13
-	Pengurangan	$8 - 5$	3

2. Operator Perbandingan

Operator	Deskripsi	Contoh	Hasil
\geq	Lebih besar atau sama dengan	$8 \geq 5$	True
\leq	Lebih kecil atau sama dengan	$8 \leq 5$	False
\neq	Tidak sama dengan	$8 \neq 5$	True
$<$	Lebih Kecil	$14 < 6$	False
$>$	Lebih besar	$5 > 3$	True
$==$	Sama dengan	$4 == 4$	True

3. Operator Penugasan

Operator	Contoh	Hasil
<code>*=</code>	<code>X *= 100</code>	<code>X = X * 100</code>
<code>/=</code>	<code>X /= 100</code>	<code>X = X / 100</code>
<code>%=</code>	<code>X %= 100</code>	<code>X = X % 100</code>
<code>+=</code>	<code>X += 100</code>	<code>X = X + 100</code>
<code>-=</code>	<code>X -= 100</code>	<code>X = X - 100</code>

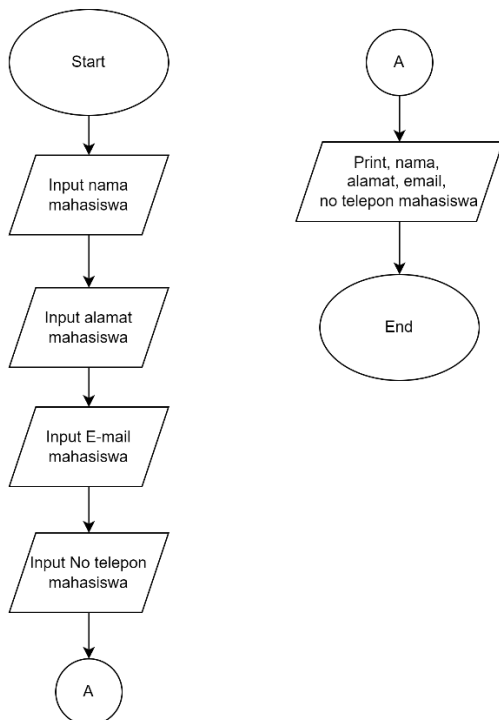
1.2. Percobaan Praktikum

1.2.1. Percobaan A

Susun algoritma dan buat program yang meminta data dasar mahasiswa (mis: Nama, Alamat, e_mail, dan telepon) kemudian menampilkan-nya kembali tersusun.

Berikut adalah flowchart dan kode program beserta outputnya

algoritma dan buat program yang meminta data
dasar mahasiswa



```
44 # Meminta data dasar mahasiswa
45 nama_mahasiswa = input('Masukkan nama mahasiswa : ')
46 alamat = input('Masukkan alamat mahasiswa : ')
47 email = input('masukkan Email (hanya nama) : ')
48 no_telepon = input('Masukkan nomor telepon : ')
49
50 print('Data mahasiswa : ')
51 print(f'Nama Mahasiswa : {nama_mahasiswa}')
52 print(f'Alamat Mahasiswa : {alamat}')
53 print(f'Email Mahasiswa : {email}@student.untan.ac.id')
54 print(f'Nomor Telepon Mahasiswa : {no_telepon}')
```

Output :

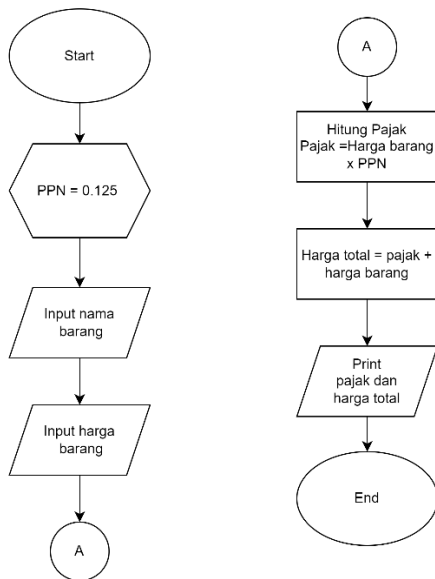
```
Data mahasiswa :
Nama Mahasiswa : Rafli Pratama
Alamat Mahasiswa : Sungai rengas
Email Mahasiswa : h1101241008@student.untan.ac.id
Nomor Telepon Mahasiswa : 089694226167
```

Penjelasan :

Didalam algoritma atau program, pertama-tama user akan diminta untuk memasukkan inputan berupa nama, alamat, email dan nomor telepon mahasiswa lalu masing-masing data inputan tersebut di simpan di dalam sebuah variabel, lalu variabel tersebut akan di panggil melalui fungsi print yang menampilkan data data yang telah di inputkan user sebelumnya.

1.2.2. Percobaan B

Susun algoritma dan buat program yang menghitung pajak pertambahan nilai (ppn) 12.50% dengan meminta harga barang yang dibeli dari pengguna program.



```
24 namaBarang = input('Input nama barang : ')
25 hargaBarang = int(input('Input masukkan harga barang : '))
26 tax = 12.5 / 100 * hargaBarang
27 totalHarga = tax + hargaBarang
28 print(f'Nama barang : {namaBarang}')
29 print(f'Harga barang : Rp.{hargaBarang}')
30 print(f'Pajak yang harus di bayar adalah : Rp.{int(tax)}')
31 print(f'Total harga yang harus di bayar adalah Rp.{int(totalHarga)}')
```

Output :

```
PS D:\Programming\Python> & C:/Users/HP/AppData/Local
Input nama barang : Milo
Input masukkan harga barang : 7000
Nama barang : Milo
Harga barang : Rp.7000
Pajak yang harus di bayar adalah : Rp.875
Total harga yang harus di bayar adalah Rp.7875
PS D:\Programming\Python> █
```

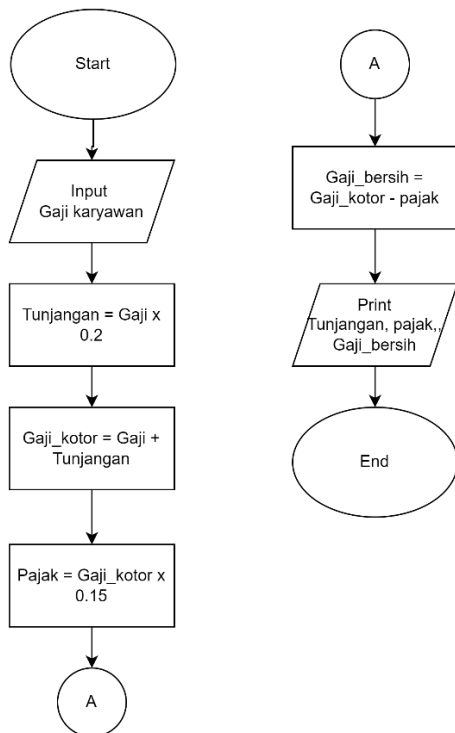
Penjelasan :

Dalam program tersebut user akan di minta menginputkan nama barang dan harga barang, lalu harga barang tersebut akan di kalikan dengan PPN yang bernilai 12.5% sehingga dapatlah pajak dari harga barang tersebut kemudian total harga didapat dari penjumlahan antara harga barang dan pajak, lalu setiap data seperti nama, harga pajak dan total harga di simpan dalam sebuah variabel, lalu variabel tersebut akan di panggil ke dalam sebuah fungsi print untuk menampilkan output data yang telah di inputkan user sebelumnya, dan output perhitungan pajak dan total harga.

1.2.3. Percobaan C

Susun algoritma dan buat program dengan sebuah proses algoritma suatu perusahaan mempunyai n orang karyawan dengan asumsi penggajian sebagai berikut:

1. Gaji setiap karyawan akan sama jumlahnya.
2. Gaji karyawan dihitung dengan cara gaji pokok ditambah tunjangan dikurangi pajak.
3. Pajak yang berlaku di perusahaan tersebut adalah 15% dari gaji pokok setelah ditambah tunjangan.
4. Tunjangan yang didapat dari perusahaan adalah 20%.
5. Gaji pokok bisa berubah tergantung kebijakan perusahaan.
6. Hitung gaji bersih karyawan dan dicetak ke layar



```

1  # #Program gaji karyawan
2
3  gaji_pokok = int(input('Masukkan Gaji Pokok : '))
4  tunjangan = int (20 / 100 * gaji_pokok)
5  gaji_kotor = gaji_pokok + tunjangan
6  pajak = int(15 / 100 * gaji_kotor)
7  gaji_bersih = gaji_kotor - pajak
8
9  print('-----')
10 print(f'Gaji pokok adalah Rp.{gaji_pokok}')
11 print(f'Tunjangan Gaji adalah Rp.{tunjangan}')
12 print(f'Gaji kotor karyawan sebesar Rp.{gaji_kotor}')
13 print(f'Potongan pajak sebesar 15% dari gaji kotor')
14 print(f'Pajak adalah sebesar Rp.{pajak}')
15 print(f'Gaji Bersih karyawan sebesar Rp.{int(gaji_bersih)}')
16 print('-----')

```

Output :

```

PS D:\Programming\Python> & C:/Users/HP/AppData/Lo
Masukkan Gaji Pokok : 1000000
-----
Gaji pokok adalah Rp.1000000
Tunjangan Gaji adalah Rp.200000
Gaji kotor karyawan sebesar Rp.1200000
Potongan pajak sebesar 15% dari gaji kotor
Pajak adalah sebesar Rp.180000
Gaji Bersih karyawan sebesar Rp.1020000
-----

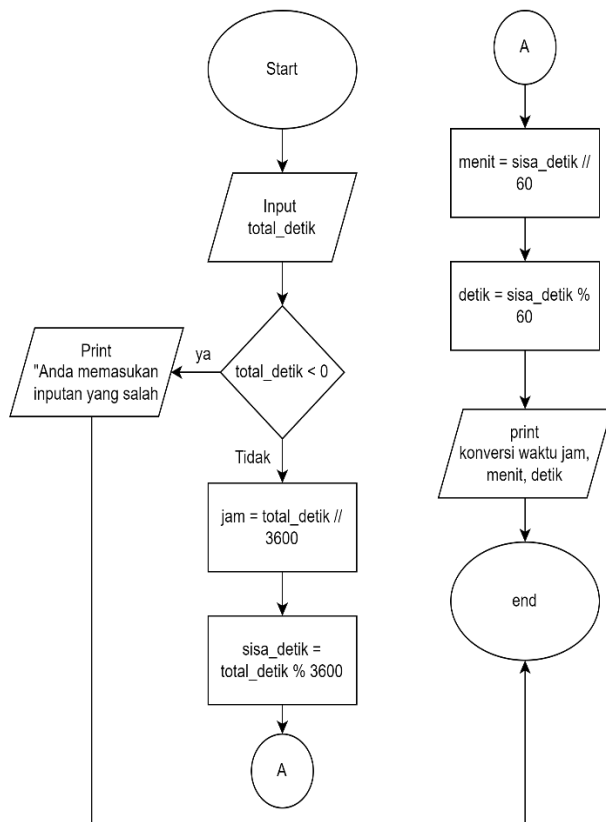
```

Penjelasan :

Pada program di atas, user akan di minta memasukkan gaji pokok karyawan dan di simpan dalam sebuah variabel yang bernama *gaji_pokok*, lalu setelah itu akan dilakukan perhitungan tunjangan yang bernilai 20% dari gaji pokok dan di simpan dalam variabel *tunjangan*, lalu di hitung juga gaji kotor karyawan yang merupakan hasil penjumlahan dari gaji pokok dan tunjangan dan di simpan dalam variabel *gaji_kotor*, lalu setelah itu di lakukan penghitungan pajak yang merupakan 15% dari gaji kotor dan disimpan dalam variabel *pajak*, setelah itu dilakukan perhitungan gaji bersih yang merupakan hasil kurang dari gaji kotor dan pajak dan hasilnya di simpan dalam variabel *gaji_bersih*, setelah itu variabel variabel yang menyimpan data data tersebut di panggil dalam sebuah fungsi print untuk menampilkan output hasil perhitungan gaji karyawan

1.2.4 Percobaan D

Buatlah algoritma dan buat program untuk mengkonversi total detik ke bentuk jam- menit-detik. Data total detik diinput oleh user. Contohnya, misalnya data total detiknya adalah 5440 detik, maka outputnya adalah 1 jam 30 menit 40 detik



```

# Konversi waktu
total_detik = int(input('Masukkan total detik : '))

if total_detik < 0 :
    print('Anda memasukkan inputan yang salah')
else :
    jam = total_detik // 3600
    sisa_detik = total_detik % 3600
    menit = sisa_detik // 60
    detik = sisa_detik % 60

    print('-----')
    print(f'Total detik = {total_detik} Detik')
    print(f'Maka konversinya adalah = {jam} Jam {menit} Menit {detik} Detik.')
    print('-----')
  
```

Output :

```

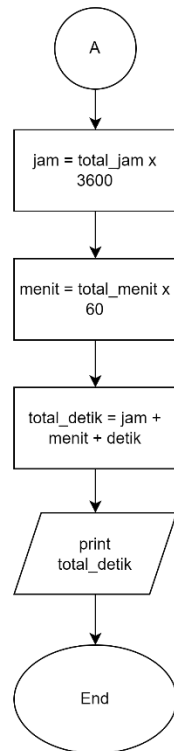
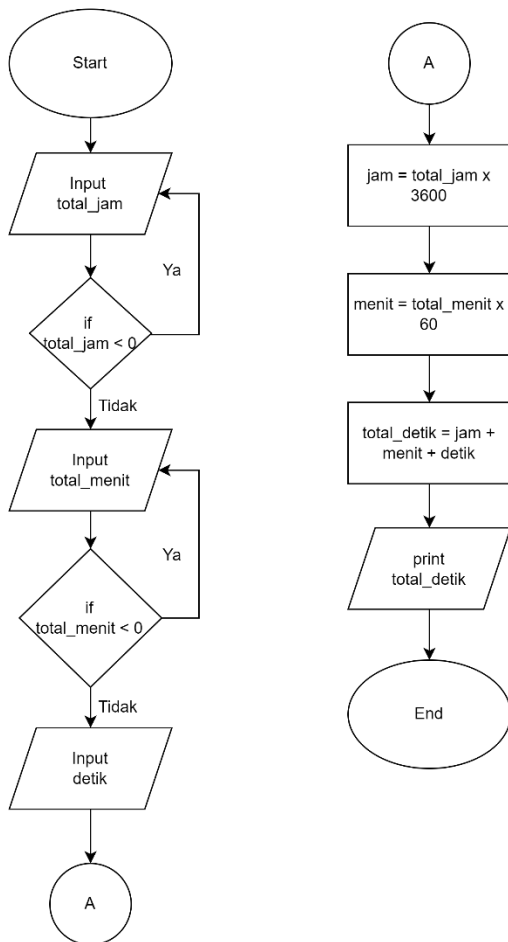
PS D:\Programming\Python\Praktikum2> & C:/Users/HP/Ap
Masukkan total detik : 5440
-----
Total detik = 5440 Detik
Maka konversinya adalah = 1 Jam 30 Menit 40 Detik.
-----
PS D:\Programming\Python\Praktikum2> & C:/Users/HP/Ap
Masukkan total detik : 2401
-----
Total detik = 2401 Detik
Maka konversinya adalah = 0 Jam 40 Menit 1 Detik.
-----
PS D:\Programming\Python\Praktikum2> & C:/Users/HP/Ap
Masukkan total detik : 46
-----
Total detik = 46 Detik
Maka konversinya adalah = 0 Jam 0 Menit 46 Detik.
-----
  
```

Penjelasan :

Pada program ini user akan diminta memasukkan total detik yang dimana inputan tersebut akan di simpan di variabel total_detik, lalu setelah itu di lakukan pengecekan apakah total detik < 0, jika iya maka program akan selesai dengan output "anda memasukkan inputan yang salah" namun jika tidak maka program akan di lanjutkan dengan perhitungan jam yang merupakan hasil dari pembagian total_detik /3600, lalu kita mencari sisa_detik dengan hitungan total_detik % 3600 yang akan di gunakan untuk mencari menit dengan hitungan sisa_detik / 60, dan mencari detik dengan rumus sisa_detik % 60, lalu semua variebel tersebut akan di panggil dengan fungsi print yang akan menampilkan output konversi total detik ke jam. Menit, detik

1.2.5 Percobaan E

Buatlah algoritma dan buat program untuk mengkonversi jam-menit-detik ke total detik. Data jam-menit-detik diinput dari user. Contoh, misalnya data jam-menit- detiknya adalah 1 jam 30 menit 40 detik, maka besarnya total detik adalah 5440 detik. Analisis: input: jam (j), menit (m), detik (d) output: total detik (total) Rumus: ingat bahwa 1 jam = 3600 detik dan 1 menit = 60 detik. maka total detik = jam x 3600 + menit x 60 + detik.



```

# konversi Jam menit detik ke total detik
total_jam = int(input('Masukkan Total jam : '))
while total_jam < 0 :
    print('-----Warning-----')
    print('Masukkan kembali total jam')
    total_jam = int(input('Masukkan Total jam : '))

total_menit = int(input('Masukkan Total menit : '))
while total_menit < 0 :
    print('-----Warning-----')
    print('Masukkan kembali total menit')
    total_menit = int(input('Masukkan Total menit : '))

detik = int(input('Masukkan Total detik : '))

print('-----')
jam = total_jam * 3600
menit = total_menit * 60
total_detik = jam + menit + detik
print(f'Waktu nya adalah {total_jam} Jam {total_menit} Menit {detik} Detik')
print(f'Total detik adalah : {total_detik} Detik')
print('-----')
  
```

Output :

```

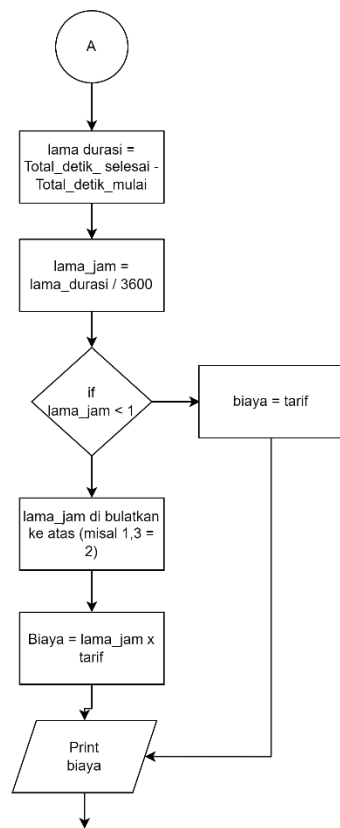
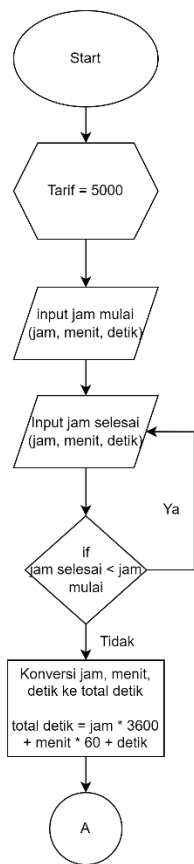
PS D:\Programming\Python\Praktikum2> python
Masukkan Total jam : 2
Masukkan Total menit : 12
Masukkan Total detik : 55
-----
Waktu nya adalah 2 Jam 12 Menit 55 Detik
Total detik adalah : 7975 Detik
-----
  
```

Penjelasan :

Dalam program ini user akan diminta memasukkan jam, menit dan detik, lalu setelah itu akan di lakukan pengecekan apakah jam atau menit yang di input kurang dari 0, jika iya maka akan diminta untuk menginputkan kembali jam dan menit yang akan disimpan ke dalam variabel total_jam dan total_menit, jika tidak maka akan di lakukan perhitungan yaitu jam dikali dengan 3600 disimpan pada variabel jam dan menit dikali 60, lalu akan di lakukan penjumlahan total detik yang akan di simpan dalam variabel total_detik dengan rumus $\text{total_detik} = \text{jam} + \text{menit} + \text{detik}$ setelah itu, fungsi print akan menampilkan output dari hasil perhitungan tadi.

1.2.6 Percobaan F

Seorang user di warnet mulai menggunakan internet pada pukul J1 dan selesai pada pukul J2. Bila tarip penggunaan di warnet tersebut 1 jam Rp. 5000,- maka buat program billing warnet untuk menghitung lama pemakaian (dalam jam menit detik) dan biaya yang harus dibayar user



```

print('-----')

jam_mulai = int(input('Masukkan jam mulai : '))
menit_mulai = int(input('Masukkan menit mulai : '))
detik_mulai = int(input('Masukkan detik mulai : '))
total_detik_jam_mulai = jam_mulai * 3600 + menit_mulai * 60 + detik_mulai
print('-----')

jam_selesai = int(input('Masukkan jam keluar : '))
while jam_selesai < jam_mulai:
    print('-----Warning-----')
    print('Jam yang anda masukkan salah, silahkan inputkan kembali')
    jam_selesai = int(input('Masukkan jam keluar : '))

menit_selesai = int(input('Masukkan menit keluar : '))
detik_selesai = int(input('Masukkan detik keluar : '))
total_detik_jam_selesai = jam_selesai * 3600 + menit_selesai * 60 + detik_selesai

lama_pemakaian = total_detik_jam_selesai - total_detik_jam_mulai

lama_jam = round(lama_pemakaian / 3600, 1)
jam_pakai = lama_pemakaian / 3600
jam_pakai = int(jam_pakai) if jam_pakai == int(jam_pakai) else int(jam_pakai) + 1 # Membulatkan ke

if jam_pakai <= 1:
    biaya = 5000
else:
    biaya =(jam_pakai - 1) * 10000

print('-----')
print(f'Jam mulai adalah pada Pukul {jam_mulai}:{menit_mulai}:{detik_mulai}')
print(f'Jam selesai adalah pada Pukul {jam_selesai}:{menit_selesai}:{detik_selesai}')
print(f'Lama pemakaian adalah {lama_jam} Jam')
print(f'Total tarif yang harus di bayar adalah Rp.{biaya}')
print('-----')

```

Output :

```

-----
Masukkan jam mulai : 12
Masukkan menit mulai : 30
Masukkan detik mulai : 00
-----
Masukkan jam keluar : 13
Masukkan menit keluar : 45
Masukkan detik keluar : 00
-----
Jam mulai adalah pada Pukul 12:30:0
Jam selesai adalah pada Pukul 13:45:0
Lama pemakaian adalah 1.2 Jam
Total tarif yang harus di bayar adalah Rp.10000
-----

```

Penjelasan :

Pada program ini user akan di minta memasukkan waktu mulai (jam, menit, detik) dan waktu selesai (jam, menit, detik) setelah itu waktu tadi akan di konversi menjadi total detik, lalu akan dilakukan perhitungan lama_durasi dari hasil pengurangan total detik mulai dan total detik selesai, lalu lama_durasi tadi akan di konversi kembali ke dalam jam, setelah itu kita menginsiasi harga tarif yaitu sebesar 5000, lalu jika lama jam durasi kurang dari 1 jam, maka harga tarif tetap 5000, dan jika lebih dari 1, maka akan di berlakukan pembulatan ke atas dan harga tarif adalah hasil kali dari tarif x lama jam yang telah di bulatkan ke atas, dan biaya tarif ini akan di tampilkan melalui fungsi print.

1.3. Kesimpulan dan Saran

1.3.1. Kesimpulan

Dari praktikum ini dapat disimpulkan :

1. Fungsi input yang di isikan oleh user akan selalu mengembalikan tipe data String
2. Tipe data yang bisa di operasikan hanyalah tipe data Float dan Integer
3. Jika ingin mengoperasikan hasil inputan user, kita perlu mengubah tipe datanya ke float dengan fungsi float() atau int()
4. Urutan presedensi dalam Python mengikuti hierarki matematika umum, seperti kurung, pangkat, perkalian, pembagian, penjumlahan, dan pengurangan.
5. Operator logika seperti and, or, dan not digunakan untuk melakukan operasi pada kondisi yang melibatkan tipe data boolean.
6. Variabel dalam Python bersifat mutable, yang artinya nilai yang tersimpan di dalam variabel dapat diubah selama program berjalan.
7. Komentar dalam kode (menggunakan #) digunakan untuk menjelaskan bagian-bagian penting dalam kode, sehingga memudahkan pemahaman dan pengembangan lebih lanjut.

1.3.2. Saran

Saran terhadap praktikum ini yaitu:

1. Pastikan untuk selalu memeriksa tipe data dari input pengguna sebelum melakukan operasi aritmatika
2. Gunakan penamaan variabel yang jelas dan deskriptif agar program dapat mudah dibaca dan dipahami.
3. Saat menulis program yang kompleks, tambahkan komentar di setiap bagian penting untuk membantu pemahaman dan pemeliharaan kode.
4. Saat eksekusi program, pastikan file dan terminal berada di direktori proyek untuk menghindari kesalahan jalur file. (tidak berlaku untuk .ipynb)
5. Selalu uji kode secara bertahap untuk memastikan setiap bagian program berfungsi dengan benar sebelum melanjutkan ke tahap berikutnya.
6. Pahami presedensi operator agar tidak terjadi kesalahan dalam urutan eksekusi operasi matematika dan logika.

DAFTAR PUSTAKA

1. Modul *PRAKTIKUM PEMOGRAMAN KOMPUTER (PYTHON)* oleh Ilhamsyah, S.Si, M.Cs.
2. <https://Dicoding.com>