

**PETUNJUK PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN I**



Oleh:

Achmad Maududie
Dwiretno Istiyadi Swasono
Tio Dharmawan
Priza Pandunata

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER
2023**

LEMBAR PENGESAHAN

Nama Matakuliah : Algoritma dan Pemrograman I
Kode : KSU1104
Program Studi : Teknologi Informasi
Fakultas : Fakultas Ilmu Komputer

Tim Penyusun

Koordinator

- a. Nama : Achmad Maududie
- b. NIP/NRP : 197004221995121001
- c. Program Studi : Teknologi Informasi
- d. Fakultas : Fakultas Ilmu Komputer

Anggota 1

- a. Nama : Dwiretno Istiyadi Swasono
- b. NIP/NRP : 197803302003121003
- c. Program Studi : Teknologi Informasi
- d. Fakultas : Fakultas Ilmu Komputer

Anggota 2

- a. Nama : Tio Dharmawan
- b. NIP/NRP : 199111122022031011
- c. Program Studi : Informatika
- d. Fakultas : Fakultas Ilmu Komputer

Anggota 2

- a. Nama : Priza Pandunata
- b. NIP/NRP : 198301212015041001
- c. Program Studi : Teknologi Informasi
- d. Fakultas : Fakultas Ilmu Komputer

Mengetahui,
Wakil Dekan 1

Jember, 13 November 2023
Koordinator,

Achmad Maududie, ST., M.Sc.
NIP: 197004221995121001

Achmad Maududie, ST., M.Sc.
NIP: 197004221995121001

PRAKATA

Puji dan syukur kami haturkan kepada Tuhan Yang Maha Kuasa, karena atas berkat dan petunjuk-Nyalah, maka seluruh proses penyusunan buku Petunjuk Praktikum Algoritma dan Pemrograman I dapat terlaksana. Buku Petunjuk Praktikum ini berisi tentang panduan dari sejumlah kegiatan praktikum yang bertujuan untuk memberikan gambaran nyata dari teori-teori yang telah diberikan melalui tatap muka di dalam kelas.

Petunjuk di setiap kegiatan praktikum dalam buku ini, disusun dengan mengikuti struktur sebagai berikut, yaitu: 1) Capaian Pembelajaran, 2) Tujuan Percobaan, 3) Tinjauan Pustaka, 4) Alat dan Bahan, 4) Prosedur Kerja, 5) Analisis Data, 6) Kesimpulan, 7) Daftar Pustaka, dan 8) Hasil Penilaian. Dengan struktur tersebut diharapkan mahasiswa mendapatkan kesempatan dalam melakukan percobaan, belajar menganalisis hasil percobaan, serta menarik kesimpulan untuk membuat gambaran umum tentang hasil percobaan, analisis, pembahasan, dan/atau pengujian hipotesa yang ada.

Tim penyusun buku Petunjuk Praktikum Algoritma dan Pemrograman sadar bahwa buku ini masih belum sempurna. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang konstruktif dari para pembaca untuk perbaikan panduan praktikum. Semoga seluruh niat baik kita, diberkati oleh Tuhan Yang Maha Kuasa.

Jember, <tgl> <bulan> <tahun>

Tim Penyusun
Buku Petunjuk Praktikum
Algoritma dan Pemrograman I

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PRAKATA	iii
DAFTAR ISI	iv
TATA TERTIB PRAKTIKUM	vi
BAB 1. PENGENALAN BAHASA PEMROGRAMAN	8
1.1 CAPAIAN PEMBELAJARAN	8
1.2 TUJUAN PERCOBAAN	8
1.3 TINJAUAN PUSTAKA	8
1.4 ALAT DAN BAHAN	10
1.5 PROSEDUR KERJA	10
1.6 HASIL DAN ANALISIS DATA	12
1.7 KESIMPULAN	13
1.8 TUGAS MANDIRI	13
1.9 DAFTAR PUSTAKA	14
1.10 HASIL PENILAIAN	15
BAB 2. TIPE DATA, VARIABEL, DAN OPERATOR ARITMATIKA	16
2.1 CAPAIAN PEMBELAJARAN	16
2.2 TUJUAN PERCOBAAN	16
2.3 TINJAUAN PUSTAKA	16
2.4 ALAT DAN BAHAN	18
2.5 PROSEDUR KERJA	18
2.6 HASIL DAN ANALISIS DATA	23
2.7 KESIMPULAN	25

2.8	TUGAS MANDIRI	25
2.9	DAFTAR PUSTAKA	26
2.10	HASIL PENILAIAN	27
BAB 3.	PERCABANGAN	28
3.1	CAPAIAN PEMBELAJARAN	28
3.2	TUJUAN PERCOBAAN	28
BAB 4.	PERULANGAN	29
4.1	CAPAIAN PEMBELAJARAN	29
4.2	TUJUAN PERCOBAAN	29
BAB 5.	FUNGSI	30
5.1	CAPAIAN PEMBELAJARAN	30
5.2	TUJUAN PERCOBAAN	30
5.3	TINJAUAN PUSTAKA	30
5.4	ALAT DAN BAHAN	31
5.5	PROSEDUR KERJA	32
5.6	HASIL DAN ANALISIS DATA	36
5.7	KESIMPULAN	37
5.8	TUGAS MANDIRI	37
5.9	DAFTAR PUSTAKA	38
5.10	HASIL PENILAIAN	39
BAB 6.	PENGELOLAAN FILE DATA	40
6.1	CAPAIAN PEMBELAJARAN	40
6.2	TUJUAN PERCOBAAN	40
6.3	TINJAUAN PUSTAKA	40

TATA TERTIB PRAKTIKUM

1. Praktikan wajib mengenakan pakaian rapi standar kuliah dan memberikan perhatian penuh pada kegiatan praktikum saat pelaksanaannya.
2. Praktikan diwajibkan untuk menjaga ketenangan dalam pelaksanaan praktikum.
3. Praktikan dilarang meninggalkan ruangan tanpa seizin asisten.
4. Praktikan yang datang terlambat lebih dari 15 menit akan dikenakan sanksi.
5. Praktikan yang tidak mengikuti praktikum tanpa keterangan tidak akan mendapat nilai.
6. Praktikan wajib melakukan presensi menggunakan tanda tangan (non-mmp)
7. Segala bentuk pelanggaran terkait poin 1 s/d 4 akan dikenakan sanksi berupa pengurangan nilai).

BAB 1. PENGENALAN BAHASA PEMROGRAMAN

1.1 CAPAIAN PEMBELAJARAN

- Kode CPL : CPL-03
- Nama CPL : Mampu menciptakan program, aplikasi atau sistem informasi untuk menyelesaikan permasalahan, khususnya bidang agroindustry.
- Kode CPMK : CPMK-01
- Nama CPMK : Mengenal bahasa pemrograman dan kakas bantu yang digunakan dalam menyusun kode program.
- Nama Sub-CPMK : Mengenal bahasa pemrograman dan kakas bantu yang digunakan dalam menyusun kode program.

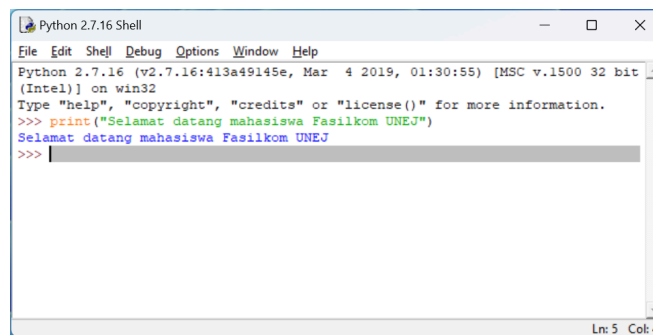
1.2 TUJUAN PERCOBAAN

Tujuan praktikum ini adalah mengenalkan dan memberikan pengalaman bagia mahasiswa dalam membuat aplikasi dengan menggunakan Bahasa Python sebagai Bahasa pemrogramannya. Dalam bab ini mahasiswa juga mulai diperkenalkan dengan *Integrated Developmen Environment* (IDE) melalui penyusunan kode program sederhana.

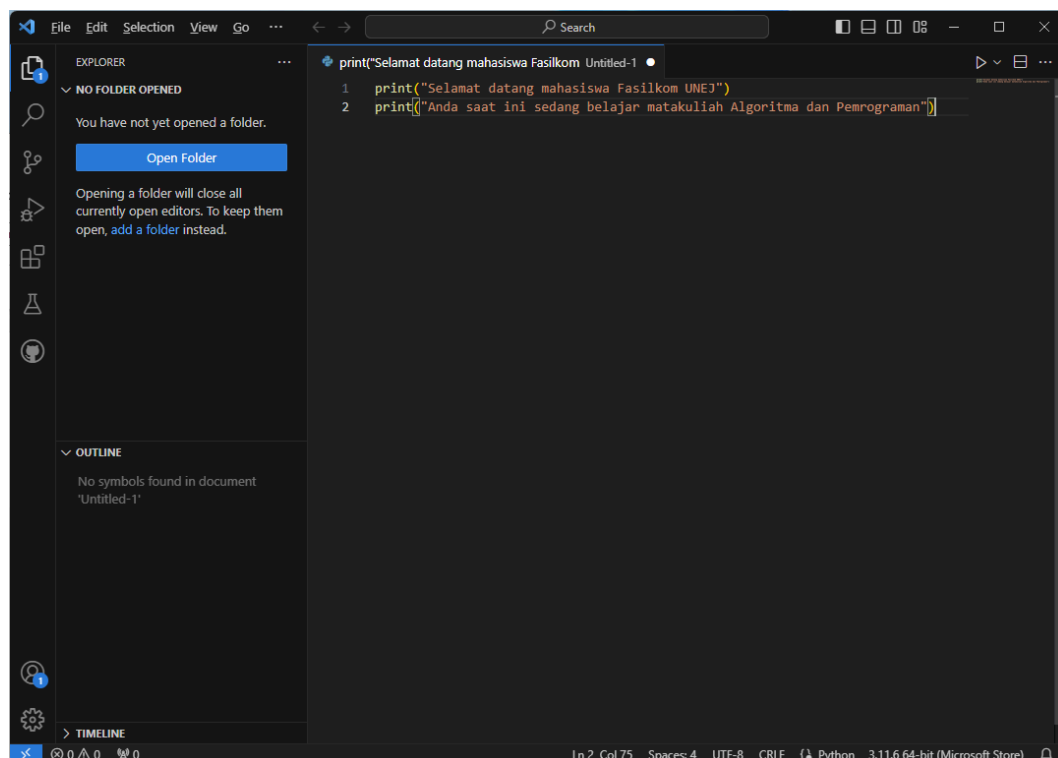
1.3 TINJAUAN PUSTAKA

Python adalah salah satu bahasa pemrograman tingkat tinggi modern yang awalnya dikembangkan oleh Guido van Rossum pada awal tahun 1990-an. Bahasa pemrograman ini berkembang sangat pesat hingga menjadi salah satu bahasa pemrograman utama yang digunakan di dunia industri maupun pendidikan. Pada saat ini (saat buku ini ditulis), ada dua versi mayor Python yang digunakan yaitu Phyton 2 yang mulai dikembangkan pada tahun 2000 dan Python 3 yang mulai dikembangkan pada tahun 2008. Meskipun kedua versi tersebut adalah bahasa pemrograman yang sama, namun kedua versi tersebut memiliki banyak perbedaan, sehingga terkadang aplikasi yang dikembangkan menggunakan Python 2 tidak dapat dijalankan dalam interpreter Python 3 dan begitu sebaliknya.

Salah satu hal yang sering menjadi pertimbangan dalam pengembangan perangkat lunak adalah ketersediaan kakas bantu (*tool*) yang digunakan dalam menyusun kode programnya. Berbagai tool telah tersedia secara luas, baik yang bersifat *proprietary* maupun *open-source*, berbayar maupun yang gratis atau sederhana maupun berfitur lengkap. Kakas bantu semacam ini dikenal dengan *Integrated Development Environment* (IDE). Contoh IDE yang sering digunakan dan tidak berbayar adalah IDLE dari python.org dan Visual Studio Code dari Microsoft.



Gambar 1.1: Integrated DeveLopment Environment (IDLE)



Gambar 1.2: Contoh tampilan Visual Studio Code

1.4 ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam percobaan ini terdiri dari perangkat keras dan perangkat lunak. Setiap mahasiswa diwajibkan mempersiapkan seluruh alat dan bahan yang diperlukan.


- Kebutuhan peralatan
 1. Perangkat keras berupa komputer PC atau Laptop/Notebook
 2. Perangkat lunak berupa kakas bantu (tool) IDE untuk menyusun kode program, yaitu Visual Studio Code 2022
- Kebutuhan bahan praktikum
 1. Tidak ada

1.5 PROSEDUR KERJA

Pada bab ini, pengenalan Bahasa program difokuskan pada pembuatan beberapa aplikasi menggunakan sejumlah statemen kode program yang sudah disediakan. Sebelum melakukan percobaan pastikan aplikasi Visual Studio Code sudah terinstall di dalam laptop yang akan digunakan.

Percobaan 1: Instalasi extensi Python dalam Visual Studio Code


Langkah-langkah percobaan:

1. Buka Visual Studio Code
2. Buka Extension View melalui menu File > Preferences > Extensions atau dengan cara mengklik tool Extensions  yang ada di Activity Bar.
3. Ketik “Python” pada text box “Search Extensions in Marketplace” dan perhatikan apa yang terjadi dalam Extension View
4. Klik tombol Install, tunggu sesaat dan perhatikan apa yang terjadi

Percobaan 2: Menentukan Folder dan Workspace dalam Visual Studio

Langkah-langkah percobaan:

1. Buat sebuah folder dengan nama “Bab1-“ + “NIM_Anda” untuk menyimpan semua percobaan dalam Bab 1 ini

2. Buka Visual Studio Code
3. Klik menu **File > Save Workspace As..**
4. Arahkan folder tempat menyimpan Workspace file ke folder yang telah anda siapkan
5. Ketikkan nama file workspace dengan format sebagai berikut “WSBab1-“ + “NIM_Anda”. Amati apa yang terjadi di dalam folder yang telah anda siapkan dan juga amati isi filenya.
6. Buatlah sebuah folder baru dengan nama “Aplikasi1” di dalam folder yang telah anda sediakan.
7. Klik menu **File > Add Folder to Workspace..** dan pilih folder “Aplikasi1” yang telah anda buat.
8. Tampilkan Explorer view melalui menu **View > Explorer** atau dengan mengklik tombol Explorer . Amati apa yang terjadi dalam Explorer view.
9. Menggunakan aplikasi Notepad, buka file “WSBab1-“ + “NIM_Anda” dan amati perubahan yang terjadi.
10. Klik menu **File > Close Workspace** dan amati apa yang terjadi
11. Klik menu **File > Open Workspace from file..** kemudian cari dan pilih file “WSBab1-“ + “NIM_Anda”. Amati apa yang terjadi.

Percobaan 3: Membuat file Python

Langkah-langkah percobaan:

1. Buka Visual Studio Code dan pastikan membuka workspace dari file “WSBab1-“ + “NIM_Anda”.
2. Klik menu **File** klik **New File** dan pastikan file yang dibuat adalah “Python File”
3. Tuliskan kode program berikut ini dalam area *Editor*.

```
print("Hello World")
```
4. Klik menu **File > Save** dan beri nama “helloworld” kemudian tekan tombol “Save”. Perhatikan apa yang terjadi folder “Aplikasi1” dalam Workspace ini.

5. Klik menu **Run > Run Without Debugging** atau dengan menekan tombol Ctrl+F5. Perhatikan tampilan dalam tab “TERMINAL”.

Percobaan 4: Fungsi print()

Langkah-langkah percobaan:

1. Pastikan file helloworld.py masih dibuka dan tambahkan kode program pada baris berikutnya hingga menjadi seperti berikut.

```
print("Hello World")
print('Saya adalah mahasiswa FASILKOM UNEJ')
print("""Saat ini saya masih semester satu""")
```

2. Klik menu **Run > Run Without Debugging** atau dengan menekan tombol Ctrl+F5. Perhatikan tampilan dalam tab “TERMINAL”.
3. Tambahkan kode program dalam file yang sama kode program berikut dan kemudian jalankan (melalui menu menu **Run > Run Without Debugging** atau dengan menekan tombol Ctrl+F5). Perhatikan tampilan dalam tab “TERMINAL”.

```
print()
print("Saat ini saya sedang belajar"+"Algoritma dan Pemrograman")
print("Saat ini saya sedang belajar","Algoritma dan Pemrograman")
print("Saat ini saya sedang belajar","Algoritma dan Pemrograman", sep="")
print("Saat ini saya sedang belajar","Algoritma dan Pemrograman", sep=" -> ")
```

4. Tambahkan kode program dalam file yang sama kode program berikut dan kemudian jalankan (melalui menu menu **Run > Run Without Debugging** atau dengan menekan tombol Ctrl+F5). Perhatikan tampilan dalam tab “TERMINAL”.

```
print("\nAlamat kampus kami")
print("Alamat: {0}\n{1}".format("Jl. Kalimantan 37", "Kampus Tegalbot"))
print("Kabupaten ", end='')
print("Jember", sep=' - ', end=" ")
print("68126")
```

1.6 HASIL DAN ANALISIS DATA

Berdasarkan seluruh percobaan yang telah dilakukan, maka hasil dan pembahasan berdasarkan teori pendukung atas setiap hasilnya disajikan berikut ini.

Percobaan 1: Instalasi extensi Python dalam Visual Studio Code

1. Tampilkan hasil *percobaan 1* pada langkah 3 dan berikan penjelasan atas hasil saat anda mengetikkan “Python” pada text box “Search Extensions in Marketplace”.
2. Tampilkan hasil *percobaan 1* pada langkah 4 dan berikan penjelasan atas hasil saat anda telah mengklik tombol Install.

Percobaan 2: Menentukan Folder dan Workspace dalam Visual Studio

1. Tampilkan hasil *percobaan 2* pada langkah 5 dan berikan penjelasan atas isi file Workspace tersebut.
2. Tampilkan hasil *percobaan 2* pada langkah 9 dan berikan penjelasan atas perbandingan isi file Workspace dengan isi sebelumnya.
3. Tampilkan hasil *percobaan 2* pada langkah 11 dan berikan penjelasannya.

Percobaan 3: Membuat file Python

1. Tampilkan hasil *percobaan 3* pada langkah 4 dan berikan penjelasan atas tampilan tersebut.
2. Tampilkan hasil *percobaan 3* pada langkah 5 dan berikan penjelasan atas tampilan tersebut.

Percobaan 4: Fungsi print()

1. Tampilkan hasil *percobaan 4* pada langkah 2 dan berikan penjelasan atas tampilan tersebut.
2. Tampilkan hasil *percobaan 4* pada langkah 3 dan berikan penjelasan atas tampilan tersebut.
3. Tampilkan hasil *percobaan 4* pada langkah 4 dan berikan penjelasan atas tampilan tersebut.

1.7 KESIMPULAN

Buatlah kesimpulan untuk bab ini berdasarkan semua percobaan yang telah dilakukan.

1.8 TUGAS MANDIRI

Buatlah program sederhana untuk menampilkan nim, nama lengkap, tanggal lahir, alamat tinggal, nama program studi, nama fakultas, dan lima nama lengkap teman anda satu angkatan.

1.9 DAFTAR PUSTAKA

Uzayr, S.B., 2021. Optimizing Visual Studio Code for Python Development. Apress.

1.10 HASIL PENILAIAN

Identitas Penilai (Asisten Praktikum)

NIM	
Nama	

Penilaian

No	Item	Skor
1	Pre-test (maks 5)	
2	Hasil analisis data (maks 30)	
3	Penarikan kesimpulan (maks 25)	
4	Tugas Mandiri (maks 15)	
5	Post-test (maks 15)	
6	Sikap (maks 10)	
Total Nilai		

Jember, <tgl> <bulan> <tahun>

<Nama Lengkap Penilai>

NIM: xxxxxxxxxxxxxx

BAB 2. TIPE DATA, VARIABEL, DAN OPERATOR ARITMATIKA

2.1 CAPAIAN PEMBELAJARAN

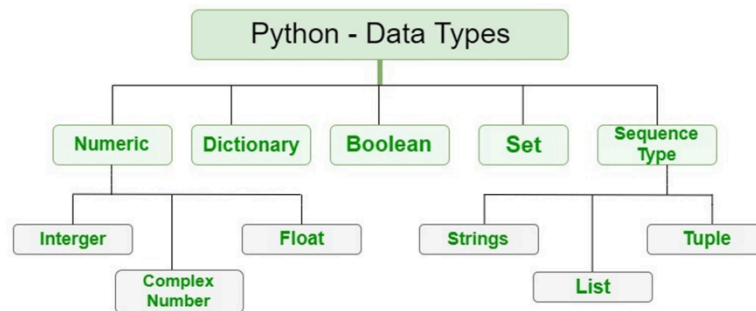
- Kode CPL : CPL-03
- Nama CPL : Mampu menciptakan program, aplikasi atau sistem informasi untuk menyelesaikan permasalahan, khususnya bidang agroindustry.
- Kode CPMK : CPMK-02; CPMK-03
- Nama CPMK : Mampu menerapkan konsep tipe data dan variabel dalam penanganan data; Mampu menerapkan operasi dasar aritmatika dalam kode program.
- Nama Sub-CPMK : Mampu menerapkan konsep tipe data dan variabel dalam penanganan data; Mampu menerapkan operasi dasar aritmatika pada proses perhitungan dalam kode program.

2.2 TUJUAN PERCOBAAN

Tujuan praktikum ini adalah pemahaman tentang tipe data dan variable serta operasi-operasi dasar matematika yang digunakan dalam menangani dan memproses data.

2.3 TINJAUAN PUSTAKA

Tipe data dan variabel adalah konsep dasar dalam Bahasa pemrograman, tidak terkecuali Python. Variabel digunakan untuk menyimpan nilai atau data secara sementara yang akan digunakan dalam program, sedangkan tipe data merujuk pada jenis nilai atau data yang disimpan dalam variabel.



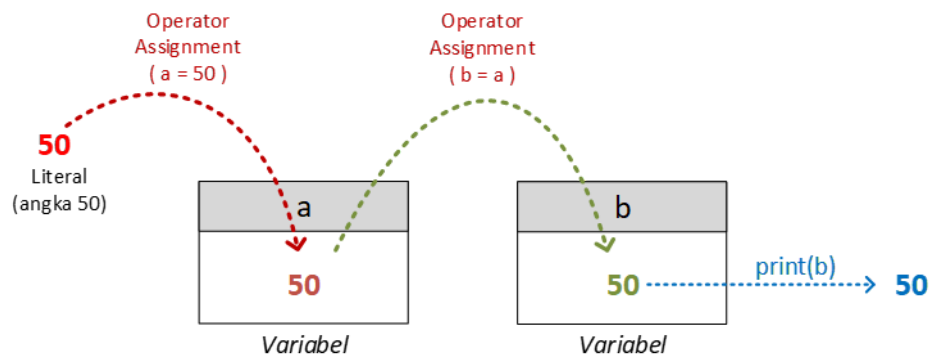
Gambar 2.1: Tipe data dalam Bahasa Python

Python tidak termasuk sebagai bahasa pemrograman *strongly type* karena bahasa ini tidak membutuhkan mekanisme pendeklarasian tipe data terlebih dahulu sebelum digunakan. Python secara otomatis menyesuaikan tipe data yang digunakan oleh sebuah variabel (*identifier*) berdasarkan obyek yang disimpan di dalamnya. Dengan demikian, sebuah variabel sewaktu-waktu dapat berubah tipe datanya sesuai dengan isi yang ada di dalamnya. Bahasa pemrograman yang demikian disebut *dynamically typed language*.

Pada prinsipnya, Python memberikan kebebasan bagi programmer untuk menentukan nama variabel yang digunakan. Namun demikian, penamaan variabel tersebut harus mengikuti aturan yang ditetapkan oleh bahasa pemrograman tersebut, yaitu: 1) nama variabel bersifat *case sensitive*; 2) nama variabel tidak diperbolehkan memakai kata yang telah digunakan bahasa pemrograman atau yang dikenal dengan istilah *reserved word*; 3) nama variabel tidak diperbolehkan mengandung karakter spasi; dan 4) nama variabel tidak boleh diawali dengan angka. Selain itu, penamaan variabel sangat disarankan mencerminkan data yang dikandung oleh variabel tersebut serta menggunakan nama yang sesingkat mungkin.

Dalam implementasinya, Python menggunakan sebuah operator khusus yang dinamakan sebagai *assignment* untuk menyimpan nilai dalam sebuah variabel. Setiap statemen *assignment* selalu menggunakan operator *assignment* yang diwakili dengan karakter sama dengan (“=”). Setiap penggunaan operator

assignment selalu memberikan dampak perubahan nilai yang disimpan dalam variabel.



Gambar 2.2: Ilustrasi perubahan nilai dalam variable saat dikenai operator assignment

Python memiliki kemampuan dasar untuk melakukan perhitungan matematika dasar, seperti penjumlahan, pengurangan, perkalian, dan pembagian. Operator penjumlahan dan pengurangan dalam Python menggunakan karakter yang sama seperti operator matematika biasa yaitu karakter plus (“+”) untuk operator penjumlahan dan karakter minus (“-“) untuk operator pengurangan. Khusus untuk operator perkalian, Python menggunakan karakter asterisk (“*”) sedangkan operator pembagian Python menggunakan karakter (“/”).

2.4 ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam percobaan ini terdiri dari perangkat keras dan perangkat lunak. Setiap mahasiswa diwajibkan mempersiapkan seluruh alat dan bahan yang diperlukan.

- Kebutuhan peralatan
 1. Perangkat keras berupa komputer PC atau Laptop/Notebook
 2. Perangkat lunak Visual Studio Code 2022
- Kebutuhan bahan praktikum
 1. Tidak ada

2.5 PROSEDUR KERJA

Bab ini terdiri dari lima percobaan, dua percobaan (1 dan 2) digunakan untuk memahami konsep tipe data dan variable sedangkan percobaan sisanya

untuk memahami konsep operator aritmatika dalam Bahasa pemrograman. Sebelum memulai, buatlah sebuah folder dengan nama “Bab2-“ + “NIM_Anda” untuk menyimpan semua percobaan dalam Bab 2 ini serta buat file Workspace dengan nama “WSBab2-“ + “NIM_Anda” dalam folder tersebut. Pastikan file Workspace yang anda gunakan di setiap percobaan adalah file Workspace ini.

Percobaan 1: Pengenalan tipe data

Langkah-langkah percobaan:

1. Buat sebuah folder dengan nama Aplikasi2 dan tambahkan folder tersebut dalam Workspace ini.
2. Buatlah sebuah file baru melalui menu **File > New File** dan berikan nama “NIM” + “tipe_data”. Pastikan file yang dibuat adalah “Python File”.
3. Tuliskan kode program berikut ini dalam area *Editor* dan kemudian jalankan. Perhatikan tampilan dalam tab “TERMINAL”.

```
print(type(10))
print(type(10.05))
print(type("Fasilkom"))
print(type(True))
print(type((10, 10.05, "Fasilkom", True)))
print(type([10, 10.05, "Fasilkom", True]))
print(type({"nama_ayah" : "Sukriman", "nama_ibu" : "Romlah Surjannah"}))
```

4. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
print("=====")
print(float(10), type(float(10)))
print(int(10.05), type(int(10.05)))
print(int(True), type(int(True)))
print(list((10, 10.05, "Fasilkom", True)), type(list((10, 10.05, "Fasilkom", True))))
print(tuple([10, 10.05, "Fasilkom", True]), type(tuple([10, 10.05, "Fasilkom", True])))
```

5. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
print("=====")
print(list({"nama_ayah" : "Sukriman", "nama_ibu" : "Romlah Surjannah"}.keys()))
print(list({"nama_ayah" : "Sukriman", "nama_ibu" : "Romlah Surjannah"}.values()))
```

Percobaan 2: Variabel

1. Buat sebuah file Python baru dengan nama “NIM”+“variabel” dalam folder “Aplikasi2”
2. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```

a = 10
b = 10.05
c = "Fasilkom"
d = True
e = (10, 10.05, "Fasilkom", True)
f = [10, 10.05, "Fasilkom", True]
g = {"nama_ayah" : "Sukriman", "nama_ibu" : "Romlah Surjannah"}

print("Nilai: {0} - tipe: {1}".format(a, type(a)))
print("Nilai: {0} - tipe: {1}".format(b, type(b)))
print("Nilai: {0} - tipe: {1}".format(c, type(c)))
print("Nilai: {0} - tipe: {1}".format(d, type(d)))
print("Nilai: {0} - tipe: {1}".format(e, type(e)))
print("Nilai: {0} - tipe: {1}".format(f, type(f)))
print("Nilai: {0} - tipe: {1}".format(g, type(g)))

```

3. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```

b = a
c = a
d = a
e = a
f = a
g = a

print("Nilai: {0} - tipe: {1}".format(a, type(a)))
print("Nilai: {0} - tipe: {1}".format(b, type(b)))
print("Nilai: {0} - tipe: {1}".format(c, type(c)))
print("Nilai: {0} - tipe: {1}".format(d, type(d)))
print("Nilai: {0} - tipe: {1}".format(e, type(e)))
print("Nilai: {0} - tipe: {1}".format(f, type(f)))
print("Nilai: {0} - tipe: {1}".format(g, type(g)))

```

Percobaan 3: Penamaan Variabel

1. Buat sebuah file Python baru dengan nama “NIM”+“penamaan_variabel” dalam folder “Aplikasi2”
2. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```

namaFakultas = "Fasilkom"
NamaFakultas = "FEB"
NAMAFAKULTAS = "FKIP"

print(namaFakultas)
print>NamaFakultas)
print(NAMAFAKULTAS)

```

3. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```

_namaFakultas = "Fak. Hukum"
nama_Fakultas = "Fak. Kedokteran"
nama1Fakultas = "FTP"
namaFakultas1 = "Fak. Pertanian"

print(_namaFakultas)
print(nama1Fakultas)
print(namaFakultas1)

```

4. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```

nama = input("Nama lengkap: ")
nim = input("NIM: ")
alamat = input("Alamat tinggal: ")
jmlSaudara = int(input("Jml saudara dlm keluarga: "))
hasilPerhitungan = float(input("Hitung 3 dibagi 2:"))

print("Nama lengkap: {0} ({1})".format(nama, nim))
print("Alamat tinggal: ", alamat)
print("Jumlah saudara: ", jmlSaudara)
print("Jawaban saya: ", hasilPerhitungan)

```

Entrikan setiap input yang ada dalam tab “TERMINAL”.

5. Tambahkan kode program berikut. Jalankan dan amati apa yang terjadi.

```

-namaFakultas = "FISIP"
nama Fakultas = "FKep"
lnamaFakultas = "Fak. Ilmu Budaya"

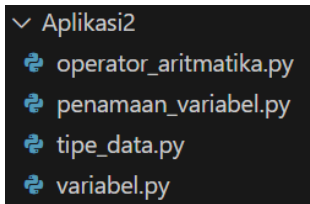
```

Percobaan 4: Operator Aritmatika

Format:

variabel = operan <operator> operan

1. Buat sebuah file Python baru dengan nama “operator_aritmatika” dalam folder “Aplikasi2”



2. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
jumlah = 9+1
print(jumlah)
jumlah = jumlah+5
print(jumlah)
```

3. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
jumlah = 10-5
print(jumlah)
jumlah = jumlah-5
print(jumlah)
```

4. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
jumlah = 2*2
print(jumlah)
jumlah = jumlah*2
print(jumlah)
```

5. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
jumlah = 2*2
print(jumlah)
jumlah = jumlah*2
print(jumlah)
```

6. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
jumlah = 2/2
print(jumlah)
jumlah = jumlah/2
print(jumlah)
```

7. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
jumlah = 2//2
print(jumlah)
jumlah = jumlah//2
print(jumlah)
```

8. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
jumlah = 10%2
print(jumlah)
jumlah = 9%2
print(jumlah)
```

9. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
string1 = 'Algoritma'
string2 = 'Pemrograman'
print(string1+string2)
```

10. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
pattern = "Saya tidak akan nakal lagi"
print(pattern*10)
```

2.6 HASIL DAN ANALISIS DATA

Berdasarkan seluruh percobaan yang telah dilakukan, maka hasil dan pembahasan berdasarkan teori pendukung atas setiap hasilnya disajikan berikut ini.

Percobaan 1: Pengenalan tipe data

1. Tampilkan hasil *percobaan 1* pada langkah 3 dan berikan penjelasan atas tampilan tersebut.
2. Tampilkan hasil *percobaan 1* pada langkah 4 dan berikan penjelasan atas tampilan tersebut.
3. Tampilkan hasil *percobaan 1* pada langkah 5 dan berikan penjelasan atas tampilan tersebut.

Percobaan 2: Variabel

1. Tampilkan hasil *percobaan 2* pada langkah 2 dan berikan penjelasan atas tampilan tersebut.
2. Tampilkan hasil *percobaan 2* pada langkah 3 dan berikan penjelasan atas tampilan tersebut.

Percobaan 3: Penamaan Variabel

1. Tampilkan hasil *percobaan 3* pada langkah 2 dan berikan penjelasan atas tampilan tersebut.
2. Tampilkan hasil *percobaan 3* pada langkah 3 dan berikan penjelasan atas tampilan tersebut.
3. Tampilkan hasil *percobaan 3* pada langkah 4 dan berikan penjelasan atas tampilan tersebut.
4. Tampilkan hasil *percobaan 3* pada langkah 5 dan berikan penjelasan atas tampilan tersebut.

Percobaan 4: Operator Aritmatika

1. Tampilkan hasil *percobaan 3* pada langkah 2 dan berikan penjelasan atas tampilan tersebut.
2. Tampilkan hasil *percobaan 3* pada langkah 3 dan berikan penjelasan atas tampilan tersebut.
3. Tampilkan hasil *percobaan 3* pada langkah 4 dan berikan penjelasan atas tampilan tersebut.
4. Tampilkan hasil *percobaan 3* pada langkah 5 dan berikan penjelasan atas tampilan tersebut.

5. Tampilkan hasil *percobaan 3* pada langkah 6 dan berikan penjelasan atas tampilan tersebut.
6. Tampilkan hasil *percobaan 3* pada langkah 7 dan berikan penjelasan atas tampilan tersebut.
7. Tampilkan hasil *percobaan 3* pada langkah 8 dan berikan penjelasan atas tampilan tersebut.
8. Tampilkan hasil *percobaan 3* pada langkah 9 dan berikan penjelasan atas tampilan tersebut.
9. Tampilkan hasil *percobaan 3* pada langkah 10 dan berikan penjelasan atas tampilan tersebut.

2.7 KESIMPULAN

Buatlah kesimpulan untuk bab ini berdasarkan semua percobaan yang telah dilakukan.

2.8 TUGAS MANDIRI

Buatlah program toko alat pertanian sederhana untuk **mengentri** dan **menampilkan** tiga barang dengan atribut kode barang, nama barang, jumlah, harga, dan diskon. Berikut contoh tampilan saat program dijalankan dan telah dientrikan semua datanya.

Catatan:

1. Nama toko dan Alamat bukan berupa entrain.
2. Nama toko disesuaikan dengan nama lengkap anda.
3. Semua nilai dalam tampilan daftar barang merupakan nilai yang diperoleh saat dientrikan data.
4. Harga jual merupakan perhitungan dari harga dikurangi harga diskon.

```

=====
TOKO PERTANIAN ARUMI SUKANDAR
Jl. Melati No 9 Jember
=====

Barang 1
Kode: 001
Nama: Benih Padi
Jumlah: 50
Harga: 155000
Diskon (%):5.5

Barang 2
Kode: 002
Nama: Cangkul
Jumlah: 25
Harga: 75000
Diskon (%):10

Barang 3
Kode: 003
Nama: Pupuk Urea
Jumlah: 100
Harga: 350000
Diskon (%):10

Daftar barang
1. 001 - Benih Padi; Jml: 50; Harga: 155000; Diskon: 5.5; Harga jual: 146475.0
2. 002 - Cangkul; Jml: 25; Harga: 75000; Diskon: 10.0; Harga jual: 67500.0
3. 003 - Pupuk Urea; Jml: 100; Harga: 350000; Diskon: 10.0; Harga jual: 315000.0

```

2.9 DAFTAR PUSTAKA

- Alchin, M., Browning, J.B., 2014. Pro Python. Apress.
- Payne, J., 2010. Beginning Python. John Wiley & Sons.

2.10 HASIL PENILAIAN

Identitas Penilai (Asisten Praktikum)

NIM	
Nama	

Penilaian

No	Item	Skor
1	Pre-test (maks 5)	
2	Hasil analisis data (maks 30)	
3	Penarikan kesimpulan (maks 25)	
4	Tugas Mandiri (maks 15)	
5	Post-test (maks 15)	
6	Sikap (maks 10)	
Total Nilai		

Jember, <tgl> <bulan> <tahun>

<Nama Lengkap Penilai>

NIM: xxxxxxxxxxxxxxxxx

BAB 3. PERCABANGAN

3.1 CAPAIAN PEMBELAJARAN

Kode CPL	: CPL-03
Nama CPL	: Mampu menciptakan program, aplikasi atau sistem informasi untuk menyelesaikan permasalahan, khususnya bidang agroindustri
Kode CPMK	: CPMK-04
Nama CPMK	: Mampu menerapkan pengaturan alur program dengan menerapkan konsep operator logika dalam statemen <code>if</code>
Nama Sub-CPMK	: Mampu menerapkan pengaturan alur program dengan menerapkan konsep operator logika dalam statemen <code>if</code>

3.2 TUJUAN PERCOBAAN

Tujuan praktikum ini adalah memahami bagaimana pengaturan alur profram dengan menerapkan konsep operator logika dalam statemen perintah `if` yang digunakan dalam menangani dan memproses data.

1.1 TINJAUAN PUSTAKA

Percabangan dalam pemrograman adalah konsep yang memungkinkan eksekusi kode untuk beralih berdasarkan kondisi yang diberikan. Dalam Python, terdapat beberapa pernyataan percabangan yang umum digunakan:

Pernyataan If:

Pernyataan `if` digunakan untuk mengevaluasi suatu kondisi dan menjalankan blok kode jika kondisi tersebut benar (True).

Contoh:

```
x = 10
if x > 5:
    print("Nilai x lebih besar dari 5")
```

Pernyataan If-Else:

Pernyataan `if-else` digunakan untuk menjalankan blok kode tertentu jika kondisi adalah benar, dan blok lain jika kondisi adalah salah.

Contoh:

```
x = 3
if x > 5:
    print("Nilai x lebih besar dari 5")
else:
    print("Nilai x tidak lebih besar dari 5")
```

Pernyataan If-Elif-Else:

Pernyataan `if-elif-else` digunakan untuk mengevaluasi beberapa kondisi berbeda dan menjalankan blok kode yang sesuai dengan kondisi yang pertama kali terpenuhi.

Contoh:

```
x = 20
if x > 5:
    print("Nilai x lebih besar dari 5")
elif x == 5:
    print("Nilai x sama dengan 5")
else:
    print("Nilai x kurang dari 5")
```

Percabangan Bersarang (Nested If):

Ini adalah penggunaan pernyataan percabangan di dalam pernyataan percabangan lainnya.

Contoh:

```
x = 10
if x > 5:
    if x < 15:
        print("Nilai x di antara 5 dan 15")
    else:
        print("Nilai x lebih besar dari 15")
```

Percabangan memungkinkan pengendalian alur program berdasarkan kondisi-kondisi tertentu. Python juga mendukung operator logika seperti `and`, `or`, dan `not` untuk menggabungkan kondisi-kondisi dalam pernyataan percabangan.

```
x = 10
y = 5

if x > 5 and y > 5:
    print("Kedua x dan y lebih besar dari 5")
elif x > 5 or y > 5:
    print("Salah satu x atau y lebih besar dari 5")
else:
    print("Kedua x dan y kurang dari atau sama dengan 5")
```

Percabangan menjadi dasar yang sangat penting dalam algoritma dan pemrograman karena memungkinkan program untuk membuat keputusan berdasarkan berbagai situasi yang mungkin terjadi. Hal ini sangat bermanfaat dalam mengontrol alur eksekusi kode untuk menangani berbagai skenario.

1.2 ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam percobaan ini terdiri dari perangkat keras dan perangkat lunak. Setiap mahasiswa diwajibkan mempersiapkan seluruh alat dan bahan yang diperlukan.

- Kebutuhan peralatan
 1. Perangkat keras berupa komputer PC atau Laptop/Notebook
 2. Perangkat lunak berupa kakas bantu (tool) IDE untuk menyusun kode program, yaitu Visual Studio Code 2022
- Kebutuhan bahan praktikum
 1. Tidak ada

1.3 PROSEDUR KERJA

Bab ini terdiri dari dua percobaan, dua percobaan (1 dan 2) digunakan untuk memahami konsep percabangan dalam Bahasa pemrograman Python. Sebelum memulai, buatlah sebuah folder dengan nama “Bab3-“ + “NIM_Anda”

untuk menyimpan semua percobaan dalam Bab 3 ini serta buat file Workspace dengan nama “WSBab3-“ + “NIM_Anda” dalam folder tersebut. Pastikan file Workspace yang anda gunakan di setiap percobaan adalah file Workspace ini.

Percobaan 1: Keputusan Pemupukan Berdasarkan Kondisi Tanah

Percobaan ini berupa pembuatan kode program untuk membantu petani dalam menentukan jenis pemupukan yang tepat berdasarkan kondisi tanah. Program akan mengevaluasi kelembaban tanah dan menentukan jenis pemupukan yang sesuai.

Langkah-langkah:

- Membaca Kondisi Tanah:
Program akan meminta input kelembaban tanah dari pengguna.
- Percabangan:
 - Jika kelembaban tanah rendah (misalnya di bawah 30%), program akan mencetak "Gunakan pupuk nitrogen".
 - Jika kelembaban tanah cukup (misalnya antara 30% dan 60%), program akan mencetak "Gunakan pupuk fosfor".
 - Jika kelembaban tanah tinggi (misalnya di atas 60%), program akan mencetak "Gunakan pupuk kalium".

Tulis dan jalankan contoh kode program berikut:

```
kelembaban_tanah = float(input("Masukkan kelembaban tanah (%): "))  
  
if kelembaban_tanah < 30:  
    print("Gunakan pupuk nitrogen")  
elif 30 <= kelembaban_tanah <= 60:  
    print("Gunakan pupuk fosfor")  
else:  
    print("Gunakan pupuk kalium")
```

Percobaan 2: Pengendalian Irigasi Berbasis Kondisi Cuaca

Percobaan ini berupa pembuatan kode program sederhana untuk mensimulasikan pengendalian irigasi pertanian berdasarkan kondisi cuaca. Program akan

memutuskan apakah sistem irigasi harus diaktifkan atau tidak berdasarkan prediksi cuaca.

Langkah-langkah:

- **Membaca Prediksi Cuaca:**
Program memperoleh data cuaca dari suatu sumber (bisa di-simulasikan dengan input manual atau menggunakan data dari sumber eksternal).
- **Percabangan:**
 - Jika prediksi cuaca adalah "cerah" atau "sedikit berawan", program akan mencetak "Sistem irigasi dimatikan".
 - Jika prediksi cuaca adalah "hujan" atau "badai", program akan mencetak "Sistem irigasi diaktifkan".

Tulis dan jalankan contoh kode program berikut:

```
prediksi_cuaca = "hujan" # Data cuaca (simulasi)

if prediksi_cuaca == "cerah" or prediksi_cuaca == "sedikit berawan":
    print("Sistem irigasi dimatikan")
elif prediksi_cuaca == "hujan" or prediksi_cuaca == "badai":
    print("Sistem irigasi diaktifkan")
```

1.4 HASIL DAN ANALISIS

Percobaan 1: Keputusan Pemupukan Berdasarkan Kondisi Tanah

Data Input (Simulasi):

Kelembaban tanah: 25%

Hasil:

Berdasarkan kelembaban tanah di bawah 30%, rekomendasi pemupukan adalah menggunakan pupuk nitrogen.

Analisis:

Dalam kondisi kelembaban tanah rendah, keputusan untuk menggunakan pupuk nitrogen dapat membantu dalam meningkatkan ketersediaan nutrisi bagi tanaman.

Percobaan 2: Pengendalian Irigasi Berbasis Kondisi Cuaca

Data Input (Simulasi):

Prediksi cuaca: "hujan"

Hasil:

Berdasarkan prediksi cuaca "hujan", sistem irigasi diaktifkan.

Analisis:

Dalam kondisi "hujan", keputusan untuk mengaktifkan sistem irigasi sesuai dengan kebutuhan tanaman yang memerlukan air tambahan selama periode hujan.

1.5 KESIMPULAN

Peran Struktur Percabangan:

- Struktur percabangan, seperti if, elif, dan else, digunakan untuk membuat keputusan yang berbeda tergantung pada kondisi yang dievaluasi.
- Dalam kedua skenario, struktur percabangan digunakan untuk mengatur alur program berdasarkan kondisi lingkungan yang berbeda.

Keputusan Berbasis Data:

- Struktur percabangan memungkinkan program untuk membuat keputusan berdasarkan informasi yang didapat, seperti prediksi cuaca atau kelembaban tanah, yang penting dalam pengelolaan agroindustri.

1.6 TUGAS MANDIRI

Judul Tugas: "Pengelolaan Panen Berdasarkan Kondisi Cuaca"

Deskripsi Tugas:

Anda diminta untuk membuat program sederhana yang dapat memprediksi tindakan yang tepat dalam pengelolaan panen berdasarkan kondisi cuaca. Gunakan struktur percabangan dalam Python untuk menentukan keputusan berdasarkan prediksi cuaca yang diberikan. Tugas ini bertujuan untuk mempraktikkan penggunaan struktur percabangan dalam konteks agroindustri.

Langkah-langkah Tugas:

- Buatlah sistem input prediksi cuaca menggunakan variabel atau fungsi dalam Python.
- Gunakan struktur percabangan (if, elif, else) untuk menentukan tindakan yang sesuai berdasarkan prediksi cuaca.
- Program harus memberikan output yang menentukan aksi yang harus dilakukan dalam pengelolaan panen berdasarkan prediksi cuaca yang diinput.
- Anda dapat menambahkan fitur tambahan sesuai kreativitas Anda untuk meningkatkan fungsionalitas program.

Catatan:

Pastikan program Anda mampu merespons berbagai kondisi cuaca yang berbeda untuk memberikan rekomendasi yang tepat terkait pengelolaan panen. Anda juga dapat menyertakan penjelasan tambahan atau dokumentasi yang menjelaskan logika keputusan yang diambil berdasarkan kondisi cuaca.

Tugas ini adalah kesempatan bagi Anda untuk berlatih penggunaan struktur percabangan dalam membuat keputusan adaptif yang relevan dengan kondisi lingkungan, khususnya dalam pengelolaan panen dalam konteks agroindustri.

1.7 DAFTAR PUSTAKA

Downey, A. (2015). *Think Python: How to Think Like a Computer Scientist*. O'Reilly Media.

Lutz, M. (2013). *Learning Python*. O'Reilly Media.

Peterson, J. (2018). "Control Structures and Conditional Statements in Python." *Journal of Programming Languages*. 6(2), 87-104.

Python Software Foundation. Python Documentation - Official Website. [Online]. Tersedia: <https://docs.python.org/3/tutorial/controlflow.html>.

GeeksforGeeks. "Python Control Flow Statements: if, elif, else, while, for loops." [Online]. Tersedia: <https://www.geeksforgeeks.org/python-programming-language/>.

1.8 HASIL PENILAIAN

Identitas Penilai (Asisten Praktikum)

NIM	
Nama	

Penilaian

No	Item	Skor
1	Pre-test (maks 5)	
2	Hasil analisis data (maks 30)	
3	Penarikan kesimpulan (maks 25)	
4	Tugas Mandiri (maks 15)	
5	Post-test (maks 15)	
6	Sikap (maks 10)	
Total Nilai		

Jember, <tgl> <bulan> <tahun>

<Nama Lengkap Penilai>

NIM: xxxxxxxxxxxxxxxxx

BAB 2. PERULANGAN

2.1 CAPAIAN PEMBELAJARAN

Kode CPL	: CPL-03
Nama CPL	: Mampu menciptakan program, aplikasi atau sistem informasi untuk menyelesaikan permasalahan, khususnya bidang agroindustri
Kode CPMK	: CPMK-05
Nama CPMK	: Mampu menerapkan pengaturan alur program menggunakan konsep perulangan dengan memanfaatkan statemen <code>while</code> dan <code>for</code>
Nama Sub-CPMK	: Mampu menerapkan pengaturan alur program menggunakan konsep perulangan dengan memanfaatkan statemen <code>while</code> dan <code>for</code>

2.2 TUJUAN PERCOBAAN

Tujuan praktikum ini memberikan pemahaman tentang konsep perulangan yang ada dalam Bahasa pemrograman. Pada bab ini diperkenalkan statemen `while` dan `for` untuk menyusun kode program yang membutuhkan eksekusi berulang terhadap satu atau sekelompok (blok) statemen.

2.3 TINJAUAN PUSTAKA

Perulangan atau looping adalah salah satu konsep fundamental dalam pemrograman yang memungkinkan program untuk mengeksekusi serangkaian instruksi secara berulang. Dalam Python, terdapat beberapa jenis perulangan yang umum digunakan, seperti perulangan `for` dan `while`.

Perulangan menggunakan `for`:

Perulangan `for` digunakan ketika Anda mengetahui berapa kali iterasi harus dilakukan, misalnya untuk mengulangi sebuah aksi untuk setiap elemen dalam sebuah list atau rentang tertentu.

Contoh:

```
# Perulangan menggunakan for untuk mencetak setiap elemen dalam list
fruits = ["apel", "jeruk", "pisang"]
for fruit in fruits:
    print(f"Saya suka {fruit}")
```

Hasinya akan mencetak:

```
Saya suka apel
Saya suka jeruk
Saya suka pisang
```

Perulangan menggunakan while:

Perulangan `while` digunakan ketika Anda ingin menjalankan serangkaian instruksi selama kondisi tertentu terpenuhi.

Contoh:

```
# Perulangan menggunakan while untuk menghitung dan mencetak angka
dari 1 hingga 5
count = 1
while count <= 5:
    print(count)
    count += 1
```

Hasilnya akan mencetak:

```
1
2
3
4
5
```

Dalam contoh di atas, `count` akan terus bertambah sampai mencapai nilai 6, dan perulangan akan berhenti karena kondisi `count <= 5` tidak lagi terpenuhi.

Penggunaan kontrol perulangan:

Dalam perulangan, Anda juga dapat menggunakan kontrol seperti `break` untuk menghentikan perulangan sebelum selesai dan `continue` untuk melompati iterasi tertentu.

Contoh penggunaan `break`:

```
# Penggunaan break untuk menghentikan perulangan saat nilai yang
dicari ditemukan
fruits = ["apel", "jeruk", "pisang"]
for fruit in fruits:
    if fruit == "jeruk":
        print("Buah jeruk ditemukan!")
        break
    print(f"Saya suka {fruit}")
```

Hasilnya akan mencetak:

```
Saya suka apel
Buah jeruk ditemukan!
```

Contoh penggunaan `continue`:

```
# Penggunaan continue untuk melompati iterasi saat nilai tertentu
ditemukan
fruits = ["apel", "jeruk", "pisang"]
for fruit in fruits:
    if fruit == "jeruk":
        print("Buah jeruk dilewati!")
        continue
    print(f"Saya suka {fruit}")
```

Hasilnya akan mencetak:

```
Saya suka apel
Buah jeruk dilewati!
Saya suka pisang
```

2.4 ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam percobaan ini terdiri dari perangkat keras dan perangkat lunak. Setiap mahasiswa diwajibkan mempersiapkan seluruh alat dan bahan yang diperlukan.

- Kebutuhan peralatan
 3. Perangkat keras berupa komputer PC atau Laptop/Notebook
 4. Perangkat lunak berupa kakas bantu (tool) IDE untuk menyusun kode program, yaitu Visual Studio Code 2022
- Kebutuhan bahan praktikum
 2. Tidak ada

2.5 PROSEDUR KERJA

Bab ini terdiri dari dua percobaan, dua percobaan (1 dan 2) digunakan untuk memahami konsep percabangan dalam Bahasa pemrograman Python. Sebelum memulai, buatlah sebuah folder dengan nama “Bab4-“ + “NIM_Anda” untuk menyimpan semua percobaan dalam Bab 4 ini serta buat file Workspace dengan nama “WSBab4-“ + “NIM_Anda” dalam folder tersebut. Pastikan file Workspace yang anda gunakan di setiap percobaan adalah file Workspace ini.

Percobaan 1: Penggunaan Perulangan untuk Pemantauan Kondisi Tanaman Setiap Hari

Deskripsi:

Seorang petani ingin membuat program sederhana untuk memantau kondisi tanaman setiap hari selama seminggu. Dia ingin mencatat tingkat kelembaban tanah setiap hari dan menentukan jika perlu dilakukan penyiraman.

Eksperimen:

- Program akan mengambil data kelembaban tanah setiap hari selama seminggu.

- Menggunakan perulangan `for` untuk mengiterasi melalui hari-hari dalam seminggu.
- Pada setiap iterasi, program akan meminta pengguna untuk memasukkan tingkat kelembaban tanah.
- Berdasarkan nilai yang dimasukkan, program akan menentukan apakah tanaman perlu disiram atau tidak.

Tulis dan jalankan kode program berikut:

```

days_of_week = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu",
"Minggu"]

for day in days_of_week:
    soil_moisture = float(input(f"Masukkan tingkat kelembaban tanah
untuk {day}: "))

    if soil_moisture < 30: # Misalkan 30 adalah ambang batas
kelembaban tanah yang memicu penyiraman
        print(f"Tanaman perlu disiram pada hari {day}")
    else:
        print(f"Tanaman tidak perlu disiram pada hari {day}")

```

Percobaan 2: Pemantauan Pertumbuhan Tanaman Setiap Minggu

Deskripsi:

Sebuah proyek riset agronomi mengharuskan pemantauan pertumbuhan tanaman setiap minggu. Mereka ingin membuat program sederhana untuk mencatat tinggi tanaman setiap minggu selama sebulan.

Eksperimen:

- Program akan mencatat tinggi tanaman setiap minggu selama satu bulan.
- Menggunakan perulangan `while` untuk menghitung empat minggu.
- Setiap minggu, program akan meminta pengguna untuk memasukkan tinggi tanaman.
- Program akan mencatat tinggi tanaman dan melanjutkan ke minggu berikutnya.

Tulis dan jalankan kode program berikut:

```
week = 1
while week <= 4:
    plant_height = float(input(f"Masukkan tinggi tanaman untuk minggu ke-{week}: "))
    print(f"Tinggi tanaman untuk minggu ke-{week}: {plant_height} cm")
    week += 1
```

2.6 HASIL DAN ANALISIS

Percobaan 1: Penggunaan Perulangan untuk Pemantauan Kondisi Tanaman Setiap Hari

Data Masukan:

- Misalkan tingkat kelembaban tanah yang dimasukkan pengguna adalah:
 - Senin: 25, Selasa: 35, Rabu: 20, Kamis: 40, Jumat: 28, Sabtu: 30, Minggu: 22

Analisis:

- Berdasarkan data yang dimasukkan, program memberikan rekomendasi penyiraman atau tidak untuk setiap hari.
- Tanaman perlu disiram pada hari Senin, Rabu, Jumat, dan Minggu karena kelembaban di bawah ambang batas yang ditetapkan.
- Hari Selasa, Kamis, dan Sabtu, tanaman tidak perlu disiram karena kelembaban tanah sudah mencukupi.

Percobaan 2: Pemantauan Pertumbuhan Tanaman Setiap Minggu

Data Masukan:

- Misalkan tinggi tanaman yang dimasukkan pengguna adalah:
 - Minggu 1: 12 cm, Minggu 2: 15 cm, Minggu 3: 18 cm, Minggu 4: 20 cm

Analisis:

- Program mencatat tinggi tanaman setiap minggu selama empat minggu.
- Terdapat peningkatan tinggi tanaman dari minggu ke minggu: dari 12 cm di minggu pertama hingga 20 cm di minggu keempat.
- Terjadi peningkatan rata-rata pertumbuhan sebesar 3 cm setiap minggu selama empat minggu.
- Dapat diasumsikan bahwa tanaman dalam kondisi yang baik karena mengalami pertumbuhan yang stabil.

2.7 KESIMPULAN

Percobaan 1: Pemantauan Kelembaban Tanah Setiap Hari

Penerapan Struktur Perulangan:

- Perulangan for untuk Mengiterasi Hari-hari: Struktur perulangan for digunakan untuk mengiterasi melalui daftar hari dalam seminggu.
- Pengambilan Data Setiap Hari: Dalam setiap iterasi, program meminta pengguna untuk memasukkan data kelembaban tanah.

Poin-poin Kesimpulan yang Terkait dengan Struktur Perulangan:

- Kebutuhan Penyiraman yang Tepat: Perulangan for memungkinkan program memberikan rekomendasi penyiraman yang tepat sesuai dengan kebutuhan tanaman pada setiap hari.
- Optimalisasi Pemantauan Rutin: Struktur perulangan memungkinkan pemantauan kelembaban tanah secara terjadwal, memungkinkan petani atau agroindustri untuk mengikuti perkembangan tanaman setiap hari.
- Kontrol Pengambilan Keputusan: Dengan struktur perulangan, pengambilan keputusan terkait penyiraman tanaman menjadi lebih terstruktur dan dapat dijadwalkan secara sistematis.

Percobaan 2: Pemantauan Pertumbuhan Tanaman Setiap Minggu

Penerapan Struktur Perulangan:

- Perulangan while untuk Menghitung Minggu: Program menggunakan perulangan while untuk menghitung empat minggu.
- Pencatatan Tinggi Tanaman Setiap Minggu: Setiap minggu, program meminta pengguna untuk memasukkan data tinggi tanaman.

Poin-poin Kesimpulan yang Terkait dengan Struktur Perulangan:

- Pencatatan dan Analisis Pada Interval Waktu: Struktur perulangan while memungkinkan pencatatan tinggi tanaman pada interval waktu tertentu (mingguan) untuk analisis performa tanaman.
- Pertumbuhan Konsisten yang Terpantau: Dengan perulangan, program dapat mencatat pertumbuhan yang konsisten dari minggu ke minggu, memungkinkan analisis performa tanaman dengan lebih terperinci.
- Perencanaan dan Pengaturan Pemantauan: Struktur perulangan while memungkinkan perencanaan pemantauan yang terjadwal, memungkinkan pengguna untuk melakukan analisis performa tanaman dalam jangka waktu tertentu.

2.8 TUGAS MANDIRI

Judul Tugas: Pemantauan Inventarisasi Tanaman dengan Penggunaan Perulangan dalam Python

Deskripsi Tugas:

Anda diminta untuk membuat sebuah program dalam bahasa Python yang akan membantu dalam menginventarisasi tanaman di sebuah kebun. Program ini akan meminta pengguna untuk memasukkan nama-nama tanaman beserta jumlahnya dan kemudian menampilkan inventarisasi yang telah terkumpul. Pastikan Anda menggunakan konsep perulangan untuk memungkinkan penggunaan berulang dalam inventarisasi tanaman.

Spesifikasi Tugas:

- Program harus memungkinkan pengguna memasukkan nama tanaman dan jumlahnya untuk inventarisasi.
- Inventarisasi tanaman harus diakumulasi jika nama tanaman yang sama dimasukkan lebih dari satu kali.
- Program harus menampilkan inventarisasi tanaman yang telah terkumpul setelah pengguna selesai memasukkan data.
- Gunakan struktur perulangan untuk memfasilitasi penggunaan berulang dalam melakukan inventarisasi.

Catatan Tambahan:

- Pastikan program memberikan kesempatan kepada pengguna untuk selesai dengan memberikan opsi atau kata kunci tertentu yang menunjukkan penyelesaian inventarisasi.
- Berikan petunjuk kepada pengguna tentang cara menggunakan program dengan jelas.
- Harap serahkan kode program Python dan deskripsi singkat mengenai cara menggunakan program beserta contoh input/output yang diharapkan.

2.9 DAFTAR PUSTAKA

Downey, A. (2015). *Think Python: How to Think Like a Computer Scientist*. O'Reilly Media.

Guttag, J. V. (2016). *Introduction to Computation and Programming Using Python: With Application to Understanding Data*. MIT Press.

Lutz, M. (2013). *Learning Python*. O'Reilly Media.

McGrath, M. (2017). *Python in easy steps*. In Easy Steps Limited.

Pilgrim, M., & Willison, S. (2009). *Dive Into Python 3*. Apress.

2.10 HASIL PENILAIAN

Identitas Penilai (Asisten Praktikum)

NIM	
Nama	

Penilaian

No	Item	Skor
1	Pre-test (maks 5)	
2	Hasil analisis data (maks 30)	
3	Penarikan kesimpulan (maks 25)	
4	Tugas Mandiri (maks 15)	
5	Post-test (maks 15)	
6	Sikap (maks 10)	
Total Nilai		

Jember, <tgl> <bulan> <tahun>

<Nama Lengkap Penilai>

NIM: xxxxxxxxxxxxxxxx

BAB 3. FUNGSI

3.1 CAPAIAN PEMBELAJARAN

Kode CPL	: CPL-03
Nama CPL	: Mampu menciptakan program, aplikasi atau sistem informasi untuk menyelesaikan permasalahan, khususnya bidang agroindustri
Kode CPMK	: CPMK-06
Nama CPMK	: Mampu memahami konsep fungsi dalam bahasa pemrograman serta menerapkannya dalam sebuah kode program aplikasi
Nama Sub-CPMK	: Mampu memahami konsep fungsi dalam bahasa pemrograman serta menerapkannya dalam sebuah kode program aplikasi

3.2 TUJUAN PERCOBAAN

Tujuan praktikum ini adalah memahami konsep fungsi dalam Bahasa pemrograman serta memberikan pengalaman kepada mahasiswa cara membuat dan menerapkan fungsi dalam kode program.

3.3 TINJAUAN PUSTAKA

Dalam bahasa pemrograman modern, ada satu istilah yang disebut dengan fungsi. Secara garis besar, fungsi adalah satu atau lebih statemen yang diletakkan diluar program utama yang mempunyai tugas untuk menyelesaikan suatu proses tertentu. Kode program sebuah fungsi dapat dituliskan dalam satu file yang sama dengan program utama atau dituliskan dalam file yang terpisah.

Ada dua jenis fungsi yang dikenal luas dalam bahasa pemrograman, yaitu build-in function dan user-defined function. Build-in function adalah fungsi yang telah ada dan disediakan oleh bahasa pemrograman. Fungsi tersebut dapat langsung dipergunakan oleh programmer untuk menyusun sebuah aplikasi tanpa harus membuatnya terlebih dahulu. Contoh Build-in function yang sering digunakan dalam bab sebelumnya adalah fungsi print dan input. Programmer

tidak perlu memikirkan bagaimana kode program yang digunakan untuk menampilkan teks di layar dan juga bagaimana kode program untuk menangkap tombol keyboard yang ditekan saat entri data. Programmer dapat langsung menggunakan kedua fungsi tersebut dengan cara memanggil nama fungsi yang dimaksud.

Tipe fungsi yang kedua adalah user-defined function yang merupakan fungsi buatan programmer sendiri dan umumnya dibangun untuk kebutuhan yang spesifik. Dalam membuat fungsi, programmer harus memberikan nama yang unik. Dengan demikian programmer harus memastikan bahwa nama yang dipergunakan bukanlah nama yang telah dipakai dalam bahasa pemrograman atau yang dikenal dengan istilah *reserved word*. Contoh *reserved word* dalam python adalah: print, input, try, except, int, float, tuple, list, dictionary, type, len, if, while dan for. Hal yang sangat dianjurkan dalam pemilihan nama fungsi adalah nama yang mencerminkan proses yang dikerjakan oleh fungsi yang dimaksud. Misalnya fungsi untuk memproses jumlah pembayaran dinamai dengan jumlahPembayaran.

Sintak dalam Bahasa Python mengacu pada format penulisan fungsi sebagai berikut.

```
def <nama_fungsi> (<parameters>):  
    block statement
```

Dimulai dengan keyword “def” kemudian dituliskan nama fungsinya dan diikuti dengan parameter-parameter yang digunakan yang ditempatkan dalam tanda kurung dan diakhiri dengan karakter titik dua (“:”). Block statement adalah satu atau lebih statement yang akan dieksekusi saat fungsi tersebut dipanggil.

3.4 ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam percobaan ini terdiri dari perangkat keras dan perangkat lunak. Setiap mahasiswa diwajibkan mempersiapkan seluruh alat dan bahan yang diperlukan.

- Kebutuhan peralatan

5. Perangkat keras berupa komputer PC atau Laptop/Notebook
 6. Perangkat lunak Visual Studio Code 2022
- Kebutuhan bahan praktikum
3. Tidak ada

3.5 PROSEDUR KERJA

Bab ini terdiri dari lima percobaan, dua percobaan (1 dan 2) digunakan untuk memahami konsep tipe data dan variable sedangkan percobaan sisanya untuk memahami konsep operator aritmatika dalam Bahasa pemrograman. Sebelum memulai, buatlah sebuah folder dengan nama “Bab5-“ + “NIM_Anda” untuk menyimpan semua percobaan dalam Bab 2 ini serta buat file Workspace dengan nama “WSBab5-“ + “NIM_Anda” dalam folder tersebut. Pastikan file Workspace yang anda gunakan di setiap percobaan adalah file Workspace ini.

Percobaan 1: membuat fungsi baru

Langkah-langkah percobaan:

1. Buat sebuah folder dengan nama Aplikasi5 dan tambahkan folder tersebut dalam Workspace ini.
2. Buatlah sebuah file baru melalui menu **File > New File** dan berikan nama “NIM” + “udf”. Pastikan file yang dibuat adalah “Python File”.
3. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
# Mendefinisikan fungsi cetak_aviliasi
def cetak_aviliasi():
    print("Perguruan tinggi: Universitas Jember")
    print("Fakultas: Ilmu Komputer")
    print("Alamat: Jl. Kalimantan No. 37 Kampus Tegalboto Jember")
    print()

print("Memanggil fungsi #1")
cetak_aviliasi()
print("Memanggil fungsi #2")
cetak_aviliasi()
print("Memanggil fungsi #3")
cetak_aviliasi()
```

4. Ubah kode program yang ada seperti berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```

# Mendefinisikan fungsi cetak_aviliasi
def cetak_aviliasi():
    print("Perguruan tinggi: Universitas Jember")
    print("Fakultas: Ilmu Komputer")
    print("Alamat: Jl. Kalimantan No. 37 Kampus Tegalboto Jember")
    print()

def cetak_identitas_pribadi():
    print("Nama: Joko Santoso Wijoyo")
    print("Alamat: Jl. Dumai I No 3 Kebonsari Jember")
    print("No. HP: 08123456789")

print("Memanggil fungsi #1")
cetak_aviliasi()
print("Memanggil fungsi #2")
cetak_aviliasi()
print("Memanggil fungsi #3")
cetak_aviliasi()
print("Memanggil fungsi #4")
cetak_identitas_pribadi()

```

Percobaan 2: menggunakan parameter dalam fungsi

Langkah-langkah percobaan:

1. Buatlah sebuah file baru melalui menu **File > New File** dan berikan nama "NIM" + "using_parameter". Pastikan file yang dibuat adalah "Python File".
2. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab "TERMINAL".

```

# Mendefinisikan fungsi printNama dengan dua parameter
def printNama(nama_depan, nama_belakang):
    namaLengkap = nama_depan + " " + nama_belakang
    print("\nHasil")
    print("Nama depan: %s" % nama_depan)
    print("Nama belakang: %s" % nama_belakang)
    print("Nama lengkap: %s" % namaLengkap)
    print()

data1 = input("Nama depan anda: ")
data2 = input("Nama belakang anda: ")
printNama(data1, data2)

```

3. Ubah kode program yang ada seperti berikut. Jalankan dan perhatikan tampilan dalam tab "TERMINAL".

```

# Mendefinisikan fungsi printNama dengan dua parameter
def printNama(nama_depan, nama_belakang, fakultas = "Fasilkom - UNEJ"):
    namaLengkap = nama_depan + " " + nama_belakang
    print("\nHasil")
    print(["Nama depan: %s" % nama_depan])
    print("Nama belakang: %s" % nama_belakang)
    print("Nama lengkap: %s" % namaLengkap)
    print("Fakultas: %s" % fakultas)
    print()

data1 = input("Nama depan anda: ")
data2 = input("Nama belakang anda: ")
printNama(data1, data2)

```

- Ubah kode program yang ada seperti berikut. Jalankan dengan mengentrikan nama fakultas selain “Fasilkom – UNEJ”. Perhatikan tampilan dalam tab “TERMINAL”.

```

# Mendefinisikan fungsi perhitungan pembayaran barang
# default diskon = 0 dan default pajak = 11.5
def perhitunganPembayaran(jml_item, harga_satuan, diskon = 0, pajak = 11.5):
    diskonPerItem = harga_satuan * diskon / 100.00
    totalHargaDiskon = (harga_satuan - diskonPerItem) * int(jml_item)
    totalPajak = totalHargaDiskon * pajak / 100.00
    pembayaran = totalHargaDiskon + totalPajak
    return pembayaran

pembayaranPupukNPK = perhitunganPembayaran(5, 35000, 10)
pembayaranPupukUrea = perhitunganPembayaran(20, 15000)
pembayaranCangkul = perhitunganPembayaran(10, 50000, 25, 0)
pembayaranBenihJagung = perhitunganPembayaran(5, 40000, 5, 15)

totalPembayaran = pembayaranPupukNPK + pembayaranPupukUrea \
    + pembayaranCangkul + pembayaranBenihJagung

print("Pembayaran buku tulis: Rp. %.2f" % pembayaranPupukNPK)
print("Pembayaran Pena: Rp. %.2f" % pembayaranPupukUrea)
print("Pembayaran Buku Wajib: Rp. %.2f" % pembayaranCangkul)
print("Pembayaran Majalah: Rp. %.2f" % pembayaranBenihJagung)
print()
print("Total pembayaran:Rp. %.2f" % totalPembayaran)

```

Percobaan 3: fungsi dalam fungsi

Langkah-langkah percobaan:

- Buatlah sebuah file baru melalui menu **File > New File** dan berikan nama “NIM” + “fungsi_dalam_fungsi”. Pastikan file yang dibuat adalah “Python File”.

2. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
## dengan mengembalikan daftar pembayaran dalam format list
def perhitunganPembayaran(itemPembelanjaan):
    def pembayaranPerItem(jml_item, harga_satuan, diskon, pajak):
        diskonPerItem = harga_satuan * diskon / 100.00
        TotalHargaDiskon = (harga_satuan - diskonPerItem) * int(jml_item)
        TotalPajak = TotalHargaDiskon * pajak / 100.00
        Pembayaran = TotalHargaDiskon + TotalPajak
        return Pembayaran

    listPembayaran = []
    if type(itemPembelanjaan) is list:
        for item in itemPembelanjaan:
            if type(item) is dict:
                namaBarang = item["nama_barang"]
                jmlItem = item["jml_item"]
                hargaSatuan = item["harga_satuan"]
                diskon = item["diskon"]
                pajak = item["pajak"]

                pembayaranItem = pembayaranPerItem(jmlItem, hargaSatuan, diskon, pajak)
                listPembayaran.append(
                    {"nama_barang" : namaBarang, "pembayaran_item" : pembayaranItem})

    return listPembayaran

daftarPembelanjaan = []
daftarPembelanjaan.append(
    {"nama_barang" : "Pupuk npk", "jml_item" : 5,
     "harga_satuan" : 35000, "diskon" : 10, "pajak" : 5})
daftarPembelanjaan.append(
    {"nama_barang" : "Benih Jagung", "jml_item" : 8,
     "harga_satuan" : 15000, "diskon" : 15, "pajak" : 10})

daftarPembayaran = perhitunganPembayaran(daftarPembelanjaan)
totalPembayaran = 0
for pembayaran in daftarPembayaran:
    print("Pembayaran barang %s: Rp. %.2f" % (pembayaran["nama_barang"], \
                                             pembayaran["pembayaran_item"]))
    totalPembayaran += pembayaran["pembayaran_item"]

print()
print("Total Pembayaran: Rp. %.2f" % totalPembayaran)
```

Percobaan 4: fungsi_rekursif

Langkah-langkah percobaan:

1. Buatlah sebuah file baru melalui menu **File > New File** dan berikan nama “NIM” + “fungsi_rekursif”. Pastikan file yang dibuat adalah “Python File”.

2. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```
# Mendefinisikan fungsi rekursif
def faktorial(n):
    if n == 1:
        return 1
    else:
        return n * faktorial(n - 1)

angka = int(input("Ketikkan sembarang bilangan bulat: "))
print("Faktorial dari angka tersebut adalah: %i" % faktorial(angka))
```

3.6 HASIL DAN ANALISIS DATA

Berdasarkan seluruh percobaan yang telah dilakukan, maka hasil dan pembahasan berdasarkan teori pendukung atas setiap hasilnya disajikan berikut ini.

Percobaan 1: membuat fungsi baru

1. Tampilkan hasil *percobaan 1* pada langkah 3 dan berikan penjelasan atas tampilan tersebut.
2. Tampilkan hasil *percobaan 1* pada langkah 4 dan berikan penjelasan atas tampilan tersebut.

Percobaan 2: menggunakan parameter dalam fungsi

1. Tampilkan hasil *percobaan 2* pada langkah 2 dan berikan penjelasan atas tampilan tersebut.
2. Tampilkan hasil *percobaan 2* pada langkah 3 dan berikan penjelasan atas tampilan tersebut.
3. Tampilkan hasil *percobaan 2* pada langkah 4 dan berikan penjelasan atas tampilan tersebut.

Percobaan 3: fungsi dalam fungsi

Tampilkan hasil *percobaan 3* pada langkah 2 dan berikan penjelasan atas tampilan tersebut.

Percobaan 4: fungsi_rekursif

Tampilkan hasil *percobaan 4* pada langkah 2 dan berikan penjelasan atas tampilan tersebut.

3.7 KESIMPULAN

Buatlah kesimpulan untuk bab ini berdasarkan semua percobaan yang telah dilakukan.

3.8 TUGAS MANDIRI

Buatlah program toko alat pertanian sederhana untuk **mengentri** dan **menampilkan** semua barang yang dijual oleh toko tersebut. Atribut setiap barang terdiri dari kode barang, nama barang, jumlah, harga, dan diskon. **Gunakan fungsi** untuk proses entri data dan proses menampilkan daftar barang. Berikut contoh tampilan saat program dijalankan dan telah dientrikan semua datanya.

Catatan:

1. Nama toko dan Alamat bukan berupa entrain.
2. Nama toko disesuaikan dengan nama lengkap anda.
3. Semua nilai (kode, nama barang, jumlah, harga, dan diskon) dalam tampilan daftar barang merupakan nilai yang diperoleh saat dientrikan data.
4. Harga jual merupakan perhitungan dari harga dikurangi harga diskon.
5. Aset total merupakan perhitungan total dar semua jumlah barang dikalikan harga barang.

```
Ada barang yang akan dientri? [y]/[t]y
Barang 1
Kode: 001
Nama: Benih Padi
Jumlah: 50
Harga: 350000
Diskon (%):10
Ada barang yang akan dientri? [y]/[t]y
Barang 2
Kode: 002
Nama: Benih Jagung
Jumlah: 75
Harga: 425000
Diskon (%):5
Ada barang yang akan dientri? [y]/[t]y
Barang 3
Kode: 003
Nama: Insektisida 100 ml
Jumlah: 30
Harga: 100000
Diskon (%):5.5
Ada barang yang akan dientri? [y]/[t]t
=====
TOKO PERTANIAN ARUMI SUKANDAR
Jl. Melati No 9 Jember
=====

Jml. item barang: 3
Aset total: 52375000.00
Daftar barang
1. 001 - Benih Padi; Jml: 50; Harga: 350000; Diskon: 10.0; Harga jual: 315000.0
2. 002 - Benih Jagung; Jml: 75; Harga: 425000; Diskon: 5.0; Harga jual: 403750.0
3. 003 - Insektisida 100 ml; Jml: 30; Harga: 100000; Diskon: 5.5; Harga jual: 94500.0
PS E:\Modul\Python VSC\Codes\Bab 5 Fungsi\Aplikasi5>
```

3.9 DAFTAR PUSTAKA

Alchin, M., Browning, J.B., 2014. Pro Python. Apress.
Payne, J., 2010. Beginning Python. John Wiley & Sons.
a.

3.10 HASIL PENILAIAN

Identitas Penilai (Asisten Praktikum)

NIM	
Nama	

Penilaian

No	Item	Skor
1	Pre-test (maks 5)	
2	Hasil analisis data (maks 30)	
3	Penarikan kesimpulan (maks 25)	
4	Tugas Mandiri (maks 15)	
5	Post-test (maks 15)	
6	Sikap (maks 10)	
Total Nilai		

Jember, <tgl> <bulan> <tahun>

<Nama Lengkap Penilai>

NIM: xxxxxxxxxxxxxxxxx

BAB 4. PENGELOLAAN FILE DATA

4.1 CAPAIAN PEMBELAJARAN

Kode CPL	: CPL-03
Nama CPL	: Mampu menciptakan program, aplikasi atau sistem informasi untuk menyelesaikan permasalahan, khususnya bidang agroindustri
Kode CPMK	: CPMK-07
Nama CPMK	: Mampu memahami konsep penyimpanan file sebagai tempat penyimpanan data serta penerapannya dalam sebuah kode program aplikasi
Nama Sub-CPMK	: Mampu memahami konsep penyimpanan file sebagai tempat penyimpanan data serta penerapannya dalam sebuah kode program aplikasi

4.2 TUJUAN PERCOBAAN

Tujuan praktikum ini adalah memahami cara untuk berinteraksi dengan file dan sekaligus menerapkan file sebagai tempat penyimpanan data. Disamping itu, mahasiswa juga diperkenalkan dengan konsep dataframe sebagai obyek untuk mengelola data.

4.3 TINJAUAN PUSTAKA

Dalam lingkungan digital, semua data, dokumen, hingga program disimpan dalam bentuk digital dan ditempatkan dalam sebuah tempat penyimpanan (storage) digital. Tempat penyimpanan data berupa file maupun system basisdata. Proses menyimpan dan membaca file selalu melibatkan penentuan nama file, drive yang digunakan, dan lokasi tempat penyimpanannya yang dikenal dengan direktori atau folder.

Saat mengelola file, Python menggunakan obyek file yang merepresentasikan koneksi pada file yang dimaksud. Dengan pengertian tersebut maka sebuah obyek file bukanlah sebuah file fisik yang tersimpan dalam storage. Ketika mengakses sebuah file menggunakan mode tulis, namun pada saat itu file tersebut belum ada,

maka Python langsung membuat file tersebut secara otomatis dan kemudian diaksesnya dengan mode tulis.

4.4 ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam percobaan ini terdiri dari perangkat keras dan perangkat lunak. Setiap mahasiswa diwajibkan mempersiapkan seluruh alat dan bahan yang diperlukan.

- Kebutuhan peralatan
 7. Perangkat keras berupa komputer PC atau Laptop/Notebook
 8. Perangkat lunak Visual Studio Code 2022
- Kebutuhan bahan praktikum
 4. Tidak ada

4.5 PROSEDUR KERJA

Bab ini terdiri dari lima percobaan, dua percobaan (1 dan 2) digunakan untuk memahami konsep tipe data dan variable sedangkan percobaan sisanya untuk memahami konsep operator aritmatika dalam Bahasa pemrograman. Sebelum memulai, buatlah sebuah folder dengan nama “Bab6-“ + “NIM_Anda” untuk menyimpan semua percobaan dalam Bab 6 ini serta buat file Workspace dengan nama “WSBab6-“ + “NIM_Anda” dalam folder tersebut. Pastikan file Workspace yang anda gunakan di setiap percobaan adalah file Workspace ini.

Percobaan 1: membuat file

Langkah-langkah percobaan:

1. Buat sebuah folder dengan nama Aplikasi6 dan tambahkan folder tersebut dalam Workspace ini.
2. Buatlah sebuah file baru melalui menu **File > New File** dan berikan nama “NIM” + “membuat_file”. Pastikan file yang dibuat adalah “Python File”.
3. Tambahkan kode program berikut. Jalankan dan perhatikan tampilan dalam tab “TERMINAL”.

```

import os

# Fungsi membuat dan menuliskan file teks
def tulisFileTeks(nama_file, teks):
    file = open(nama_file, "w")
    file.writelines(teks)
    file.close()

namaFile = "teks01.txt"
current_path = os.getcwd()
path = current_path+"/Files"

fullPathFile = path + "/" + namaFile
teks = "Memulai pemrograman menggunakan bahasa Python"
tulisFileTeks(fullPathFile, teks)

```

- Ubah kode program yang ada seperti berikut. Jalankan dan perhatikan tampilan dalam tab "TERMINAL".

```

import os

# Fungsi membuat, menuliskan, dan menambahkan konten file teks
def tulisFileTeks(nama_file, teks, mode):
    file = open(nama_file, mode)
    file.writelines(teks)
    file.close()

namaFile = "teks01.txt"
current_path = os.getcwd()
path = current_path+"/Files"

fullPathFile = path + "/" + namaFile

# Menggunakan mode tulis
teks = "Baris pertama: Memulai pemrograman menggunakan bahasa Python\n"
tulisFileTeks(fullPathFile, teks, "w")

# Menggunakan mode penambahan
teks = "Baris kedua: Betapa indahya dunia ini\n"
tulisFileTeks(fullPathFile, teks, "a")

```

4.6 HASIL DAN ANALISIS DATA

Berdasarkan seluruh percobaan yang telah dilakukan, maka hasil dan pembahasan berdasarkan teori pendukung atas setiap hasilnya disajikan berikut ini.

Percobaan 1: membuat file

1. Tampilkan hasil *percobaan 1* pada langkah 3 dan berikan penjelasan atas tampilan tersebut.
2. Tampilkan hasil *percobaan 1* pada langkah 4 dan berikan penjelasan atas tampilan tersebut.

4.7 KESIMPULAN

Buatlah kesimpulan untuk bab ini berdasarkan semua percobaan yang telah dilakukan.

4.8 TUGAS MANDIRI

Buatlah program toko alat pertanian sederhana untuk **mengentri** dan **menampilkan** semua barang yang dijual oleh toko tersebut. Atribut setiap barang terdiri dari kode barang, nama barang, jumlah, harga, dan diskon. **Gunakan fungsi** untuk proses entri data dan proses menampilkan daftar barang. Berikut contoh tampilan saat program dijalankan dan telah dientrikan semua datanya.

4.9 DAFTAR PUSTAKA

Alchin, M., Browning, J.B., 2014. Pro Python. Apress.
Payne, J., 2010. Beginning Python. John Wiley & Sons.

b.

4.10 HASIL PENILAIAN

Identitas Penilai (Asisten Praktikum)

NIM	
Nama	

Penilaian

No	Item	Skor
1	Pre-test (maks 5)	
2	Hasil analisis data (maks 30)	
3	Penarikan kesimpulan (maks 25)	
4	Tugas Mandiri (maks 15)	
5	Post-test (maks 15)	
6	Sikap (maks 10)	
Total Nilai		

Jember, <tgl> <bulan> <tahun>

<Nama Lengkap Penilai>

NIM: xxxxxxxxxxxxxxxx