

MODUL

PRAKTIKUM DASAR PEMROGRAMAN KOMPUTER



DISUSUN OLEH :

Brav Deva Bernadhi, ST., MT.

Imam Much Ibnu Subroto, ST., M.Sc., Ph.D

LABORATORIUM SIMULASI & KOMPUTER
JURUSAN TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
TAHUN AJARAN 2021/2022

Daftar Isi

HAK, KETENTUAN DAN TATA TERTIB PRAKTIKUM	3
PEDOMAN PENYUSUNAN LAPORAN.....	4
1. Penulisan source code	5
2. Sekilas tentang Python	5
3. Interpreter Python (Interaktif)	6
4. Script Python.....	6
5. Tipe Builtin dan operator.....	7
6. Seleksi/Kondisi.....	12
7. Perulangan.....	12
8. PETUNJUK PRAKTIKUM	15

HAK, KETENTUAN DAN TATA TERTIB PRAKTIKUM

HAK PRAKTIKAN :

1. Menerima materi sesuai dengan modul yang telah disusun.
2. Menerima pengarahan/bimbingan/asistensi baik dalam pembuatan tugas pendahuluan, penyampaian materi maupun penyusunan laporan.

KETENTUAN PRAKTIKUM

1. Praktikan adalah mahasiswa/i Jur. Teknik Industri FTI UNISSULA yang telah terdaftar dalam mata praktikum dasar pemrograman dan komputer, serta telah memenuhi semua persyaratan yang telah ditetapkan.
2. Penggunaan komputer dan alat bantu praktikum harus sesuai dengan petunjuk penggunaannya.
3. Peminjaman peralatan harus dengan persetujuan Asisten Praktikum tersebut.
4. Praktikum dilaksanakan secara daring.
5. Semua hasil praktikum harus diserahkan pada Asisten dan akan menjadi milik Laboratorium manufaktur

TATA TERTIB PRAKTIKUM

1. Praktikan diwajibkan hadir tepat pada jadwalnya. Keterlambatan lebih dari 15 menit mengakibatkan tidak boleh mengikuti praktikum pada jadwal tersebut dan diwajibkan menghadap Asisten yang bertugas.
2. Praktikan tidak boleh keluar dari laboratorium tanpa seizin Asisten Praktikum yang bertugas.
3. Praktikan yang berhalangan maka diwajibkan melapor dan memberikan surat keterangan berkenaan dengan ketidakhadirannya **minimal** 1 (satu) hari sebelum praktikum, serta bersedia untuk dipindahjadwalkan oleh Asisten Praktikum.
4. Praktikan diwajibkan mempersiapkan diri sebelum mengikuti praktikum dengan membaca, memahami materi, menunjukkan tugas pendahuluan yang telah di- Acc oleh Asisten Praktikum.
5. Pada saat pelaksanaan praktikum diharapkan untuk :
 - Memakai pakaian sopan, rapi dan berkerah (bukan jaket), bagi mahasiswi wajib untuk berbusana muslim yang rapi.
 - Memakai sepatu tertutup, tidak diperkenankan memakai sandal, jika sepatu sandal harus berkaos kaki.
 - Tidak merokok, makan, minum dan mengerjakan tugas lain yang tidak berhubungan dengan materi Praktikum.
 - Mengikuti kegiatan praktikum dengan baik, tertib dan menjaga kebersihan.
 - Tidak membawa peralatan selain yang terkait dengan kegiatan praktikum.
6. Kehadiran absensi dilakukan secara daring.
7. Praktikan dianggap gugur apabila :
 - Tidak mengikuti salah satu modul praktikum tanpa ijin.
 - Tidak melakukan asistensi tiap modul
 - Tidak mengikuti post test.
 - Tidak mengumpulkan laporan akhir sampai batas waktu yang sudah ditentukan.

Semarang, Mei 2022

Dosen Pengampu

Praktikum Dasar Pemrograman

Brav Deva Bernadhi, ST.,MT., dan Imam Much Ibnu Subroto, ST., M.Sc., Ph.D

PEDOMAN PENYUSUNAN LAPORAN

1. Penyusunan laporan dapat dimulai sejak praktikan selesai melakukan praktikum Modul I.
2. Laporan diketik rapi pada kertas putih berukuran **A4 – 80 gram**. Huruf yang dipergunakan adalah jenis Times New Roman ukuran **12 pt** dan **10 pt** untuk keterangan tabel dan gambar. Jarak antar baris adalah **1.5 spasi**. Margin **kiri – atas 4 cm, kanan – bawah 3 cm**.
3. Penomoran halaman diletakkan di bawah (**bottom – center**) dengan menggunakan huruf romawi kecil untuk halaman awal laporan. Untuk laporan utama dan lampiran penomoran halaman diletakkan pada pojok kanan atas, kecuali untuk setiap awal bab, nomor halaman berada di bagian bawah tengah. Laporan Utama dan Lampiran diberi nomor halaman dengan menggunakan angka arab.
4. Laporan dalam bentuk **softfile**.
5. Lembar Pengesahan Modul (pada masing-masing Modul) dibuat dan harus disahkan oleh Asisten Praktikum **setelah** laporan di acc.
6. Lembar Pengesahan Laporan Praktikum dibuat dan harus disahkan oleh Dosen Pengampu **sebelum** laporan tersebut dikumpulkan.

SISTEMATIKA PENULISAN LAPORAN

- Halaman Sampul Depan
- Lembar Pengesahan Laporan
- Kata Pengantar
- Daftar Isi
- BAB I : PENDAHULUAN
 - 1.1. Latar Belakang
 - 1.2. Perumusan Masalah
 - 1.3. Pembatasan Masalah
 - 1.4. Tujuan Praktikum
 - 1.5. Sistematika Penulisan
- BAB II : LANDASAN TEORI
- BAB III : PENGUMPULAN & PENGOLAHAN DATA
 - 3.1. Pengumpulan Data
 - 3.2. Pengolahan Data
- BAB IV : ANALISA
- BAB V : PENUTUP
- Daftar Pustaka
- Lampiran

1. Penulisan source code

- a) Indentasi adalah 4 karakter.
- b) Untuk indentasi, gunakan hanya spasi, jangan gunakan TAB.
- c) Panjang baris maksimal adalah 79 karakter. Pindah baris langsung dalam kurung dan hanya gunakan backslash (\) apabila diperlukan.
- d) Gunakan dua baris kosong antara class.
- e) Gunakan satu baris kosong antara fungsi.
- f) Import setiap modul dilakukan per baris, dengan urutan: standard library, pustaka pihak ketiga, pustaka lokal.
- g) Struktur program Python:
 1. Shebang Line
 2. Komentar dan docstring
 3. Import
 4. Global dan konstanta
 5. Class dan fungsi
- h) Aturan nama:
 1. Paket: lowercase
 2. Modul: lowercase, underscore bisa digunakan kalau diperlukan.
 3. Class: kapitalisasi per kata (huruf pertama setiap kata menggunakan kapital)
 4. Exception: kapitalisasi per kata (huruf pertama setiap kata menggunakan kapital). Gunakan tambahan Error untuk menandakan kesalahan.
 5. Fungsi dan method class: lowercase, setiap kata dipisahkan underscore.
 6. Variabel: lowercase, setiap kata dipisahkan underscore.
 7. Konstanta: lowercase atau UPPERCASE, setiap kata dipisahkan underscore.
- i) Case sensitive!

2. Sekilas tentang Python

- a) Open Source
 1. Sesuai dengan sertifikasi Open Source Initiative.
 2. Kompatibel dengan GPL, menurut Free Software Foundation. Walau demikian, tidak ada pembatasan copyleft GPL.
 3. Bebas digunakan, termasuk untuk produk proprietary.
 4. Selengkapnya, <http://www.python.org/psf/license/>.
- b) Sintaks sederhana, jelas, fleksibel, mudah dipelajari.
- c) Mendukung multi-paradigma, salah satunya Object-oriented.
- d) Tipe data very high level.

- e) Standard library yang sangat kaya.
- f) Berjalan di sangat banyak sistem.
- g) Dapat diextend dengan C/C++

3. Interpreter Python (Interaktif)

- a) Path default executable python:
 - 1. Windows: `c:\pythonXY\python.exe`. XY adalah versi Python. Contoh: `c:\python25\python.exe`
 - 2. Linux: `/usr/bin/python` atau `/usr/bin/pythonX.Y` (umumnya menggunakan symlink `/usr/bin/python`). XY adalah versi Python.
- b) Untuk menjalankan interpreter Python:
 - 1. Windows: Masuk ke command prompt, masuk ke direktori instalasi Python, jalankan `python.exe`. Dapat pula mengakses dari Start Menu. Daftarkanlah direktori instalasi Python ke PATH apabila diperlukan.
 - 2. Linux: Masuk ke terminal, berikan perintah: `'python'` (tanpa tanda kutip).
- c) Keluar dari sesi interaktif interpreter Python:
 - 1. Windows: `Ctrl-Z, ENTER`.
 - 2. Linux: `Ctrl-D`
- d) Untuk mencetak ke standard output, gunakan `print`. Contoh akan dibahas dalam sesi training.
- e) Untuk membaca string dari standard input, `raw_input()` bisa digunakan. Contoh akan dibahas dalam sesi training.
- f) Beberapa fungsi lain:
 - 1. `abs()`
 - 2. `chr()`
 - 3. `dir()`
 - 4. `max()`
 - 5. `min()`
 - 6. `pow()`

4. Script Python

- a) Simpan dengan ekstensi nama file `py`. Contoh: `hello.py`.
- b) Shebang
 - 1. `#!`, diikuti Path ke executable Python
 - 2. Contoh di Linux: `#!/usr/bin/python`
 - 3. Contoh di Windows: `#!c:\python25\python.exe`
 - 4. Di Linux, bisa gunakan bantuan `env`, sehingga shebang line menjadi `#!/usr/bin/env python`
 - a) Sesuaikan path ke `env`
- c) Jalankan di command line: `python <script.py>`

5. Tipe Builtin dan operator

Tipe	Detil	Catatan
None	None Object	
Number	<ul style="list-style-type: none"> • Boolean (bool) <ul style="list-style-type: none"> • True atau False • Integer (int) <ul style="list-style-type: none"> • Batas int bisa dilihat pada sys.maxint • Long integer (long) <ul style="list-style-type: none"> • Gunakan suffix L atau l • Batas long sesuai memori • Floating point (float) <ul style="list-style-type: none"> • pecahan • bisa ditulis dalam e atau E • Complex (complex) <ul style="list-style-type: none"> • bilangan kompleks • memiliki atribut real (real) dan imag (imajiner). • imajiner bisa dituliskan dengan suffix j atau J. • dir: conjugate, imag, real 	
Set	<ul style="list-style-type: none"> • Set (set): mutable <ul style="list-style-type: none"> • dir: add, clear, copy, difference, difference_update, discard, intersection, intersection_update, issubset, issuperset, pop, remove, symmetric_difference, symmetric_difference_update, union, update • Frozen Set (frozenset): immutable <ul style="list-style-type: none"> • dir: copy, difference, intersection, issubset, issuperset, symmetric_difference, union 	<ul style="list-style-type: none"> • Unordered • Semua item dalam set harus immutable • Dapat dibuat dari item iterable • Unik • Dapat dioperasi

Tipe	Detil	Catatan kan sebagai himpunan
Sequence	<ul style="list-style-type: none"> • String (str) <ul style="list-style-type: none"> • immutable • dapat dibentuk dengan pasangan: <ul style="list-style-type: none"> • kutip tunggal (') • kutip ganda (") • triple kutip tunggal ('') • triple kutip ganda (" " ") 	<ul style="list-style-type: none"> • Memiliki sifat iterable

- Pasangan triple-kutip bisa mengembed newline.
- Berlaku penggunaan escape character.
- Dapat dituliskan dalam bentuk raw string (diawali r atau R). Contoh: r'c:\window'
- Raw string, apabila diakhiri dengan backslash (\), harus berjumlah genap.
- dir: capitalize, center, count, decode, encode, endswith, expandtabs, find, index, isalnum, isalpha, isdigit, islower, isspace, istitle, isupper, join, ljust, lower, lstrip, partition, replace, rfind, rindex, rjust, rpartition, rsplit, rstrip, split, splitlines, startswith, strip, swapcase, title, translate, upper, zfill.
- Unicode String (unicode)
 - immutable
 - Diawali dengan u atau U, diikuti 4 digit hexa kode karakter unicode.
 - Berlaku penggunaan escape character.
 - Bisa diisi dengan \N{Name}, dimana Name sesuai aturan unicode.
Contoh: \N{Copyright Sign}.
 - Apabila menggunakan raw string, ditulis dengan ur.

- dir: method pada str, ditambah isdecimal dan isnumeric.
- Tuple (tuple)
 - immutable
 - Akses lebih cepat
 - Dapat dibuat dengan tuple(), atau dengan menempatkan anggota dalam kurung (dan).
 - Koma tambahan bisa dituliskan setelah anggota terakhir.
Contoh: (1,2,3,)
 - Tuple dengan satu anggota harus dituliskan dengan menambahkan koma. Contoh: (1,) dan bukan (1)
- List (list)
 - mutable
 - Lihat range()
 - Dapat dibuat dengan list(), atau dengan menempatkan anggota dalam kurung [dan].
 - Koma tambahan bisa dituliskan setelah anggota terakhir.
Contoh: [1,2,3,]
 - dir: append, count, extend, index, insert, pop, remove, reverse, sort.
- Xrange (xrange)
 - immutable
 - Dibuat dengan xrange()
 - Untuk perulangan, relatif lebih cepat dari penggunaan range().

Mapping

- Dictionary (dict)
 - Key harus berupa tipe immutable
 - Value dapat berupa hampir semua object Python
 - Dapat dibuat dengan dict(), atau dengan menempatkan key:value dalam kurung { dan }.
 - dir: clear, copy, fromkeys, get, has_key, items, iteritems, iterkeys,
- Unordered

itervalues, keys, pop, popitem,
setdefault, update, values

- File
 - File (file)
 - Dibuka dengan `open(name[, mode[, buffering]])` atau `file(name[, mode[, buffering]])`
 - Mode file: r(read, default), w(write), a(append)
 - Ditutup dengan method `close()` objek file
 - dir: `close`, `closed`, `encoding`, `fileno`, `flush`, `isatty`, `mode`, `name`, `newline`, `next`, `read`, `readinto`, `readline`, `readlines`, `seek`, `softspace`, `tell`, `truncate`, `write`, `writelines`, `xreadlines`

Operator

- Perbandingan boolean
 - `<`, lebih kecil
 - `<=`, lebih kecil sama dengan
 - `>`, lebih besar
 - `>=`, lebih besar sama dengan
 - `==`, sama dengan
 - `!=`, tidak sama dengan
- Operator Logical
 - `not`, logical negasi
 - `and`, logical and
 - `or`, logical or
- Operator aritmatika
 - `*`, perkalian
 - `/`, pembagian
 - `//`, pembagian integer
 - `%`, sisa bagi
 - `+`, penjumlahan

- -, pengurangan
- Operator bit
 - ~, bitwise complement
 - <<, shift left
 - >>, shift right
 - &, bitwise and
 - |, bitwise or
 - ^, bitwise xor
- Operator pada sequence
 - in: terdapat di dalam. Gunakan not in untuk kebalikannya.
- Operator tambahan string:
 - %: pemformatan

Index dan Slice:

Beberapa sequence seperti string, unicode string, list dan tuple mendukung index dan slicing, dengan sintaks:

- s[i]: item ke i dari sequence s, dimulai dari 0
- s[i:j]: item ke i sampai j dari sequence s
- s[i:j:k]: item ke i sampai j dari sequence s, dengan step k

Catatan untuk i, j dan k:

- Apabila nilai i dan j negatif, maka index relatif dari akhir sequence.
- Jika i tidak diberikan, maka default ke 0
- Jika j tidak diberikan, maka default ke len(s)
- Jika k tidak diberikan, maka default ke 1. Nilai k tidak bisa diisi dengan 0.
- Jika i lebih besar atau sama dengan j, maka menghasilkan slice kosong.

Untuk tipe xrange, index didukung namun slicing tidak didukung.

Lain-lain:

- Beberapa fungsi:
 - type(object): mengembalikan tipe object
 - id(object): mengembalikan alamat memori object
 - isinstance(object, class-or-type-or-tuple): mengembalikan apakah object merupakan instance dari class.
- Kategori tipe juga mencakup Callable (dapat dipanggil), Modules (setelah diimport), Classes dan Type.

6. Seleksi/Kondisi

Sintaks if

```
if <expression>:  
    <statement>  
    <statement>  
    ..  
[elif <expression>:  
    <statement>  
    <statement>  
    ..  
]  
[else:  
    <statement>  
    <statement>  
    ..  
]
```

Contoh if:

```
>>> a=10  
>>> if a > 5:  
...     print 'a besar dari 5'  
...  
a besar dari 5
```

Catatan:

- pass dapat digunakan untuk blok kosong.
- Beberapa contoh akan diberikan pada sesi training.
- Python tidak mendukung switch/case

7. Perulangan

Sintaks for

```
for <target_list> in <expression_list>:  
    <statement>  
    <statement>  
    ...  
[else:  
    <statement>
```

```
<statement>
```

```
..
```

```
]
```

Contoh for:

```
>>> for i in range(1,10,2):
```

```
...print i
```

```
...
```

```
1
```

```
3
```

```
5
```

```
7
```

```
9
```

Sintaks while

```
while <expression>:
```

```
<statement>
```

```
<statement>
```

```
...
```

```
[else:
```

```
<statement>
```

```
<statement>
```

```
..
```

```
]
```

Contoh while:

```
>>> a=1
```

```
>>> while a<5:
```

```
...print a
```

```
...a += 1
```

```
...
```

```
1
```

```
2
```

```
3
```

```
4
```

Catatan:

- pass dapat digunakan untuk blok kosong.
- Beberapa contoh akan diberikan pada sesi training.

CONTOH SCRIPT :

Program Penjumlahan Dua Bilangan

Meminta inputan dari user

```
bil1 = input('Masukkan bilangan pertama: ')
```

```
bil2 = input('Masukkan bilangan kedua: ')
```

Menjumlahkan bilangan

```
jumlah = float(bil1) + float(bil2)
```

Menampilkan jumlah

```
print('Jumlah {0} + {1} adalah {2}'.format(bil1, bil2, jumlah))
```

Program python untuk menentukan bilangan prima atau tidak

Meminta input bilangan dari user

```
num = int(input("Masukkan bilangan: "))
```

bilangan prima harus lebih besar dari 1

```
if num > 1:
```

```
    for i in range(2,num):
```

```
        if (num % i) == 0:
```

```
            print(num, "bukan bilangan prima")
```

```
            print(i, "kali", num//i, "=", num)
```

```
            break
```

```
    else:
```

```
        print(num, "adalah bilangan prima")
```

bila bilangan kurang atau sama dengan satu

```
else:
```

```
    print(num, "bukan bilangan prima")
```

Program python perulangan

```
count = 0
```

```
while (count < 5):
```

```
    print("The count is:", count)
```

```
    count = count + 1
```

```
print('Good bye!')
```

Program python perulangan

```
count = 0
```

```
while (count < 6):
```

```
    print(count, "kurang dari 6")
```

```
    count = count + 1
```

```
else:
```




```
    print(count, "tidak kurang dari 6")
```

8. PETUNJUK PRAKTIKUM

<https://www.programiz.com/python-programming/online-compiler/>




01. Hello World

```
>>> print("Hello World")
```

main.py	  	Shell
<pre>1 # Online Python compiler (interpreter) to run Python online. 2 # Write Python 3 code in this online editor and run it. 3 print("Hello world")</pre>		Hello world >

02. Join two strings

```
string1 = "Linux"
string2 = "Hint"
joined_string = string1 + string2
print(joined_string)
```

main.py	  	Shell
<pre>1 # Online Python compiler (interpreter) to run Python online. 2 # Write Python 3 code in this online editor and run it. 3 string1 = "Linux" 4 string2 = "Hint" 5 joined_string = string1 + string2 6 print(joined_string) 7</pre>		LinuxHint >

03. Format floating point in the string

```
# Use of String Formatting
```



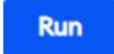
```
float1 = 563.78453
```

```
print("{:5.2f}".format(float1))
```

```
# Use of String Interpolation
```

```
float2 = 563.78453
```

```
print("%5.2f" % float2)
```

main.py	  	Shell
<pre> 1 float1 = 563.78453 2 print("{:5.2f}".format(float1)) 3 # Use of String Interpolation 4 float2 = 569.78453 5 print("%5.2f" % float2) </pre>	<pre> 563.78 569.78 > </pre>	

04. Raise a number to a power

```




import math
# Assign values to x and n
x = 4
n = 3

# Method 1
power = x ** n
print("%d to the power %d is %d" % (x,n,power))

# Method 2
power = pow(x,n)
print("%d to the power %d is %d" % (x,n,power))

# Method 3
power = math.pow(2,6.5)
print("%d to the power %d is %5.2f" % (x,n,power))

```

main.py	  	Shell
<pre> 1 import math 2 # Assign values to x and n 3 x = 4 4 n = 3 5 6 # Method 1 7 power = x ** n 8 print("%d to the power %d is %d" % (x,n,power)) 9 10 # Method 2 11 power = pow(x,n) 12 print("%d to the power %d is %d" % (x,n,power)) 13 14 # Method 3 15 power = math.pow(2,6.5) 16 print("%d to the power %d is %5.2f" % (x,n,power)) </pre>	<pre> 4 to the power 3 is 64 4 to the power 3 is 64 4 to the power 3 is 90.51 > </pre>	

05. Working with Boolean types

```

# Boolean value
val1 = True

```

```

print(val1)

# Number to Boolean
number = 10
print(bool(number))

number = -5
print(bool(number))

number = 0
print(bool(number))

# Boolean from comparison operator
val1 = 6
val2 = 3
print(val1 < val2)

```

main.py	Run	Shell
<pre> 1 # Boolean value 2 val1 = True 3 print(val1) 4 5 # Number to Boolean 6 number = 10 7 print(bool(number)) 8 9 number = -5 10 print(bool(number)) 11 12 number = 0 13 print(bool(number)) 14 15 # Boolean from comparison operator 16 val1 = 6 17 val2 = 3 18 print(val1 < val2) </pre>	<pre> True True True False False > </pre>	



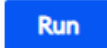
06. If else statement

```

# Assign a numeric value
number = 70

# Check the is more than 70 or not
if (number >= 70):
    print("Anda Lulus")
else:
    print("Anda Tidak Lulus")

```

main.py	  	Shell
<pre> 1 # Assign a numeric value 2 number = 70 3 4 # Check the is more than 70 or not 5 if (number >= 70): 6 print("Anda Lulus") 7 else: 8 print("Anda Tidak Lulus") 9 </pre>	<pre> Anda Lulus > </pre>	




07. Using AND and OR operators


```


# Take MCQ marks
mcq_marks = float(input("Enter the MCQ marks: "))
# Take theory marks
theory_marks = float(input("Enter the Theory marks: "))

# Check the passing condition using AND and OR operator
if (mcq_marks >= 40 and theory_marks >= 30) or (mcq_marks + theory_marks) >= 70:
    print("\nYou have passed")
else:
    print("\nYou have failed")

```

main.py	  	Shell
<pre> 1 # Take MCQ marks 2 mcq_marks = float(input("Enter the MCQ marks: ")) 3 # Take theory marks 4 theory_marks = float(input("Enter the Theory marks: ")) 5 6 # Check the passing condition using AND and OR operator 7 if (mcq_marks >= 40 and theory_marks >= 30) or (mcq_marks + 8 theory_marks) >= 70: 9 print("\nYou have passed") 10 else: 11 print("\nYou have failed") </pre>	<pre> Enter the MCQ marks: 30 Enter the Theory marks: 35 You have failed > </pre>	

main.py		Shell
<pre> 1 # Take MCQ marks 2 mcq_marks = float(input("Enter the MCQ marks: ")) 3 # Take theory marks 4 theory_marks = float(input("Enter the Theory marks: ")) 5 6 # Check the passing condition using AND and OR operator 7- if (mcq_marks >= 40 and theory_marks >= 30) or (mcq_marks + theory_marks) >=70: 8 print("\nYou have passed") 9- else: 10 print("\nYou have failed") 11 </pre>		<pre> Enter the MCQ marks: 40 Enter the Theory marks: 45 You have passed > </pre>

main.py		Shell
<pre> 1 # Take MCQ marks 2 mcq_marks = float(input("Enter the MCQ marks: ")) 3 # Take theory marks 4 theory_marks = float(input("Enter the Theory marks: ")) 5 6 # Check the passing condition using AND and OR operator 7- if (mcq_marks >= 40 and theory_marks >= 30) or (mcq_marks + theory_marks) >=70: 8 print("\nAnda Lulus") 9- else: 10 print("\nAnda Gagah") 11 </pre>		<pre> Enter the MCQ marks: 30 Enter the Theory marks: 20 Anda Gagah > </pre>

08. Switch case statement

Switcher for implementing switch case options

```

def employee_details(ID):
    switcher = {
        "1004": "Employee Name: MD. Mehrab",
        "1009": "Employee Name: Mita Rahman",
        "1010": "Employee Name: Sakib Al Hasan",
    }
    """The first argument will be returned if the match found and
    nothing will be returned if no match found"""
    return switcher.get(ID, "nothing")

```

Take the employee ID

ID = input("Enter the employee ID: ")

Print the output

print(employee_details(ID))

<pre> main.py 1 # Switcher for implementing switch case options 2 def employee_details(ID): 3 switcher = { 4 "1004": "Employee Name: MD. Mehrab", 5 "1009": "Employee Name: Mita Rahman", 6 "1010": "Employee Name: Sakib Al Hasan", 7 } 8 '''The first argument will be returned if the match found and 9 nothing will be returned if no match found''' 10 return switcher.get(ID, "nothing") 11 12 # Take the employee ID 13 ID = input("Enter the employee ID: ") 14 # Print the output 15 print(employee_details(ID)) 16 </pre>	<div style="text-align: right; border-bottom: 1px solid #ccc; padding-bottom: 5px;"> Run </div> <div style="border-top: 1px solid #ccc; padding-top: 5px;"> Shell </div> <pre> Enter the employee ID: 1010 Employee Name: Sakib Al Hasan > </pre>
--	---

09. While Loop

```

# Initialize counter
counter = 1
# Iterate the loop 5 times
while counter < 6:
    # Print the counter value
    print ("The current counter value: %d" % counter)
    # Increment the counter
    counter = counter + 1

```



<pre> main.py 1 # Initialize counter 2 counter = 1 3 # Iterate the loop 5 times 4 while counter < 6: 5 # Print the counter value 6 print ("The current counter value: %d" % counter) 7 # Increment the counter 8 counter = counter + 1 9 </pre>	<div style="text-align: right; border-bottom: 1px solid #ccc; padding-bottom: 5px;"> Run </div> <div style="border-top: 1px solid #ccc; padding-top: 5px;"> Shell </div> <pre> The current counter value: 1 The current counter value: 2 The current counter value: 3 The current counter value: 4 The current counter value: 5 > </pre>
--	--

10. For Loop

```

# Initialize the list
weekdays = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]
print("Seven Weekdays are:\n")
# Iterate the list using for loop
for day in range(len(weekdays)):
    print(weekdays[day])

```

main.py	  Run	Shell
1 # Initialize the list 2 weekdays = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", 3 print("Seven Weekdays are:\n") 4 # Iterate the list using for loop 5 for day in range(len(weekdays)): 6 print(weekdays[day]) 7		Seven Weekdays are: Sunday Monday Tuesday Wednesday Thursday Friday Saturday >