

**LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN
KOMPUTER
MODUL 5
SELECTION**

Laporan ini disusun untuk memenuhi Tugas Mata Kuliah
Praktikum Basis Data



Disusun Oleh :

AHSAN MAULANA RIZQI

104062400071

S1 BD - 04 - B

**PROGRAM STUDI S1 BISNIS DIGITAL
TELKOM UNIVERSITY PURWOKERTO
2025**

1. Dasar Teori

1.1. Percabangan

Percabangan adalah cara untuk menjalankan kode yang berbeda tergantung pada apakah kondisi yang diberikan bernilai benar (*True*) atau salah (*False*). Bentuk paling sederhana dari percabangan ini adalah dengan menggunakan *if* untuk memeriksa kondisi.

1.2. *if statement*

if statement digunakan ketika ada satu pilihan keputusan. Percabangan ini digunakan untuk menjalankan program jika kondisinya bernilai *True* dan jika kondisi bernilai *False* maka *if statement* tidak akan dijalankan.

1.3. *if else statement*

If else statement adalah percabangan program yang memiliki dua kondisi. Jika kondisi pertama terpenuhi, maka blok kode program di dalam *if statement* akan dieksekusi. Jika kondisi pertama tidak terpenuhi, maka blok kode program di dalam *else statement* akan dieksekusi.

1.4. *if elif statement*

If elif statement adalah percabangan program yang memiliki lebih dari dua kondisi. Jika kondisi pertama terpenuhi, maka blok kode program di dalam *if statement* akan dieksekusi. Jika kondisi pertama tidak terpenuhi, maka kondisi kedua akan dicek. Jika kondisi kedua terpenuhi, maka blok kode program di dalam *if elif statement* akan dieksekusi. Jika kondisi kedua tidak terpenuhi, maka *statement* tidak akan ada yang dieksekusi.

1.5. *if elif else statement*

If elif else statement adalah percabangan program yang memiliki lebih dari dua kondisi. Jika kondisi pertama terpenuhi, maka blok kode program di dalam *if statement* akan dieksekusi. Jika kondisi pertama tidak terpenuhi, maka kondisi kedua akan dicek. Jika kondisi kedua terpenuhi, maka blok kode program di dalam *elif statement* akan dieksekusi. Jika kondisi kedua tidak terpenuhi, maka blok kode program di dalam *else statement* akan dieksekusi.

1.6. *Nested if statement*

Nested if statement adalah cara untuk memasukkan satu atau lebih pernyataan *if-else* di dalam pernyataan *if-else* lainnya. Dalam *nested conditional statement*, kita dapat mengevaluasi beberapa kondisi dalam beberapa tingkat, tergantung pada kebutuhan program. Pernyataan *if-else* yang ditempatkan di dalam *if-else* lainnya disebut sebagai blok bersarang (*nested block*).

2. Pembahasan Tugas Guided

2.1. Latihan *if statement*

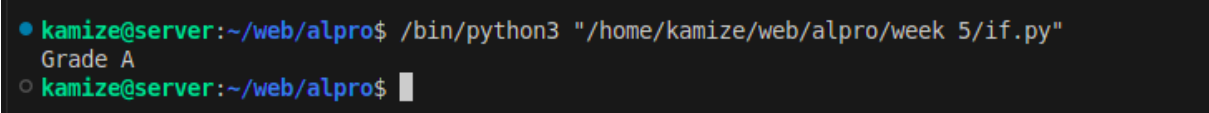


```
alpro - if.py

1  nilai = 91 # variabel nilai
2  if nilai >= 90: # jika nilai lebih besar sama dengan 90
3      print("Grade A") # tampilkan grade A
```

Gambar 1. Contoh code *if statement*

Pada gambar 1 variabel nilai memiliki nilai 91 dan `if nilai >= 90` sebagai *statement*. Jika *statement* pada variabel nilai lebih dari sama dengan 90 bernilai *True*, maka *if statement* akan dieksekusi dan teks Grade A di tampilkan ke layar. Jika tidak maka *statement* tersebut tidak dieksekusi dan tidak ditampilkan ke layar.

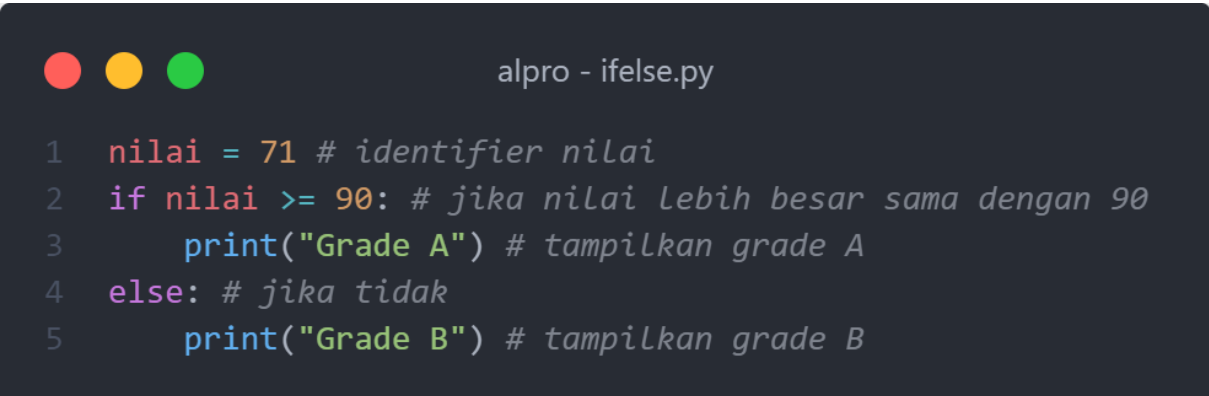


```
kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/if.py"
Grade A
kamize@server:~/web/alpro$
```

Gambar 2. hasil output pada latihan *if statement*

Pada gambar 2 menghasilkan *output* Grade A karena variabel nilai yang memiliki nilai 91 dan variabel nilai tersebut lebih dari sama dengan 90, maka *if statement* dieksekusi yang menghasilkan tulisan Grade A yang ditampilkan ke layar.

2.2. Latihan *if else statement*



```
alpro - ifelse.py

1  nilai = 71 # identifier nilai
2  if nilai >= 90: # jika nilai lebih besar sama dengan 90
3      print("Grade A") # tampilkan grade A
4  else: # jika tidak
5      print("Grade B") # tampilkan grade B
```

Gambar 3. Contoh code *if else statement*

Pada gambar 3 variabel nilai memiliki nilai 71 dan `if nilai >= 90` sebagai *statement*. Jika *statement* pada variabel nilai lebih dari sama dengan 90 bernilai *True*, maka *if statement* akan dieksekusi dan teks Grade A di tampilkan ke layar. Jika tidak maka *else statement* akan dieksekusi dan teks Grade B ditampilkan ke layar.

```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/ifelse.py"
Grade B
○ kamize@server:~/web/alpro$
```

Gambar 4. hasil output pada latihan *if else statement*

Pada gambar 4 menghasilkan *output* Grade B karena variabel nilai yang memiliki nilai 71 dan variabel nilai tersebut tidak lebih dari sama dengan 90, maka *if statement* tidak dieksekusi. Tetapi *else statement* akan dieksekusi dan teks Grade B yang ditampilkan ke layar.

2.3. Latihan *if elif statement*

```
alpro - ifelif.py

1  nilai = 61 # identifier nilai
2  if nilai >= 90: # jika nilai lebih besar sama dengan 90
3      print("Grade A") # tampilkan Grade A
4  elif nilai >= 70: # jika nilai lebih besar sama dengan 70
5      print("Grade B") # tampilkan Grade B
6  elif nilai >= 60: # jika nilai lebih besar sama dengan 60
7      print("Grade C") # tampilkan Grade C
```

Gambar 5. Contoh code *if elif statement*

Pada gambar 5 variabel nilai memiliki nilai 61. Jika *statement* pertama dengan kondisi variabel nilai lebih dari sama dengan 90 divalidasi dan menghasilkan nilai *True*, maka *if statement* pertama akan dieksekusi dan teks Grade A ditampilkan ke layar. Jika *statement* pertama *False*, maka *statement* kedua divalidasi dengan kondisi variabel nilai lebih dari sama dengan 70 dan menghasilkan nilai *True*, maka *if statement* kedua akan dieksekusi dan teks Grade B ditampilkan ke layar. Jika *statement* kedua *False*, maka *statement* ketiga divalidasi dengan kondisi variabel nilai lebih dari sama dengan 60 dan menghasilkan nilai *True*, maka *if statement* ketiga akan dieksekusi dan teks Grade C di tampilkan ke layar. Jika tidak ada *statement* yang bernilai *True* maka *statement* tidak akan dieksekusi dan tidak ditampilkan ke layar.

```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/ifelif.py"
Grade C
○ kamize@server:~/web/alpro$
```

Gambar 6. hasil output pada latihan *if elif statement*

Pada gambar 6 menghasilkan *output* Grade C karena variabel nilai yang memiliki nilai 61 dan variabel nilai tersebut memenuhi *statement* ketiga, maka *if statement* ketiga dieksekusi dan teks Grade C akan ditampilkan ke layar.

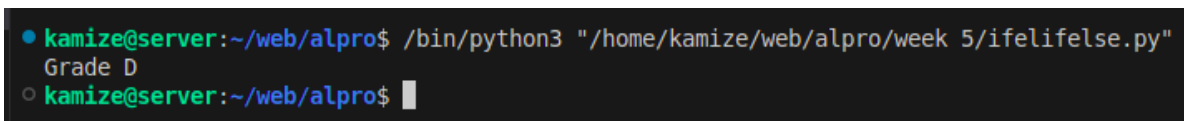
2.4. Latihan *if elif else statement*



```
1  nilai = 51 # identifier nilai
2  if nilai >= 90: # jika nilai lebih besar sama dengan 90
3      print("Grade A") # tampilkan Grade A
4  elif nilai >= 70: # jika nilai lebih besar sama dengan 70
5      print("Grade B") # tampilkan Grade B
6  elif nilai >= 60: # jika nilai lebih besar sama dengan 60
7      print("Grade C") # tampilkan Grade C
8  else: # jika tidak ada yang diatas
9      print("Grade D") # tampilkan Grade D
```

Gambar 7. Contoh code *if elif else statement*

Pada gambar 7 variabel nilai memiliki nilai 51. Jika *statement* pertama dengan kondisi variabel nilai lebih dari sama dengan 90 divalidasi dan menghasilkan nilai *True*, maka *if statement* pertama akan dieksekusi dan teks Grade A ditampilkan ke layar. Jika *statement* pertama *False*, maka *statement* kedua divalidasi dengan kondisi variabel nilai lebih dari sama dengan 70 dan menghasilkan nilai *True*. Maka *if statement* kedua akan dieksekusi dan teks Grade B ditampilkan ke layar. Jika *statement* kedua *False*, maka *statement* ketiga divalidasi dengan kondisi variabel nilai lebih dari sama dengan 60 dan menghasilkan nilai *True*, maka *if statement* ketiga akan dieksekusi dan teks Grade C di tampilkan ke layar. Jika tidak ada *statement* yang bernilai *True* maka *else statement* akan dieksekusi dan teks Grade D ditampilkan ke layar.



```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/ifelifelse.py"
Grade D
○ kamize@server:~/web/alpro$
```

Gambar 8. hasil output pada latihan *if elif else statement*

Pada gambar 8 menghasilkan *output* Grade D karena variabel nilai yang memiliki nilai 51 dan variabel nilai tersebut tidak memenuhi *statement* yang ada, maka *if statement* tidak dieksekusi. Tetapi *else statement* akan dieksekusi dan teks Grade D yang ditampilkan ke layar.

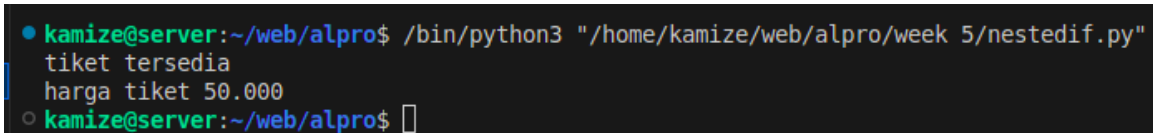
2.5. Latihan *Nested if statement*



```
1 # nested if statements
2 umur = 38 #identifikasi umur
3 if umur >= 11: # statement pertama jika umur lebih dari 11
4     print("tiket tersedia") # tampilkan tiket tersedia
5     if umur <= 20 or umur >= 60: # nested if statement jika umur kurang dari 20 atau lebih dari 60
6         print("harga tiket 35.000") # tampilkan harga tiket 35.000
7     else: # jika umur tidak memenuhi syarat diatas
8         print("harga tiket 50.000") # tampilkan harga tiket 50.000
9 else: # jika umur kurang dari 11
10     print("tiket tidak tersedia") # tampilkan tiket tidak tersedia
11
```

Gambar 9. Contoh code nested if statement

Pada gambar 1 variabel umur memiliki nilai 38. Jika *statement* pertama dengan kondisi variabel umur lebih dari sama dengan 11 divalidasi dan menghasilkan nilai *True*, maka *if statement* pertama akan dieksekusi. Yang menghasilkan *output* teks tiket tersedia yang ditampilkan ke layar dan *statement* lain yang terdapat pada *statement* pertama memiliki kondisi jika umur kurang dari sama dengan 20 atau umur lebih dari sama dengan 60, maka menghasilkan *output* teks harga tiket 35.000, jika tidak maka akan menghasilkan *output* teks harga tiket 50.000. jika *statement* pertama tidak terpenuhi maka *else statement* akan dieksekusi yang menghasilkan *output* teks tiket tidak tersedia.

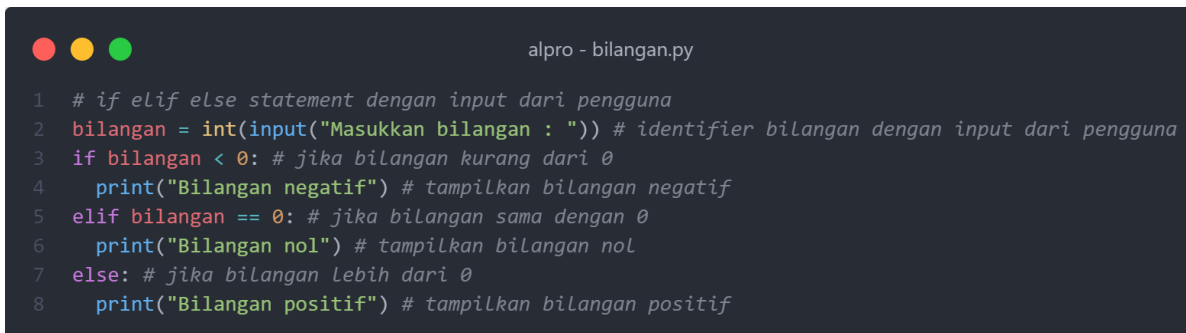


```
• kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/nestedif.py"
tiket tersedia
harga tiket 50.000
○ kamize@server:~/web/alpro$
```

Gambar 10. Hasil output pada latihan nested if statement

Pada gambar 10 menghasilkan *output* teks tiket tersedia dan harga tiket 50.000 karena variabel umur yang memiliki nilai 38 dan variabel umur tersebut memenuhi *statement* pertama, maka menghasilkan *output* teks tiket tersedia. Akan tetapi nested *if statement*nya tidak terpenuhi, maka akan menghasilkan *output* teks harga tiket 50.000 yang ditampilkan ke layar.

2.6. Latihan *if elif else statement* dengan input



```
1 # if elif else statement dengan input dari pengguna
2 bilangan = int(input("Masukkan bilangan : ")) # identifier bilangan dengan input dari pengguna
3 if bilangan < 0: # jika bilangan kurang dari 0
4     print("Bilangan negatif") # tampilkan bilangan negatif
5 elif bilangan == 0: # jika bilangan sama dengan 0
6     print("Bilangan nol") # tampilkan bilangan nol
7 else: # jika bilangan lebih dari 0
8     print("Bilangan positif") # tampilkan bilangan positif
```

Gambar 11. Contoh code if elif else statement dengan input

Pada gambar 11 variabel bilangan sebagai nilai input dari pengguna. Jika *statement* pertama yang memiliki kondisi bilangan kurang dari 0 divalidasi dan menghasilkan nilai *True*, maka *statement* pertama akan dieksekusi dan menghasilkan *output* teks berupa bilangan negatif. Jika *statement* pertama menghasilkan nilai *False*, maka *statement* kedua yang memiliki kondisi bilangan sama dengan 0 divalidasi dan menghasilkan nilai *True*. *Statement* kedua akan dieksekusi yang menghasilkan *ouput* teks berupa bilangan nol. Jika *if statement* tidak ada yang terpenuhi maka *else statement* akan dieksekusi yang menghasilkan *output* teks berupa bilangan positif.

```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/bilangan.py"
Masukkan bilangan : 12
Bilangan positif
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/bilangan.py"
Masukkan bilangan : 0
Bilangan nol
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/bilangan.py"
Masukkan bilangan : -34
Bilangan negatif
○ kamize@server:~/web/alpro$
```

Gamabar 12. Hasil output pada latihan *if elif else statement* dengan input

Pada gambar 12 menghasilkan *output* teks bilangan negatif jika pengguna memasukan bilangan kurang dari 0, contohnya -34. Jika pengguna memasukan bilangan 0, maka akan menghasilkan *output* teks bilangan nol. Jika pengguna tidak memasukan sesuai *statement* pertama atau *statement* kedua maka akan menghasilkan *output* teks bilangan positif, contohnya 12.

2.7. Latihan *if else statement* dengan operator

```
alpro - bil_terbesar.py
1 x = int(input("Masukkan bilangan X: ")) # identifier bilangan x dengan input dari pengguna
2 y = int(input("Masukkan bilangan Y: ")) # identifier bilangan y dengan input dari pengguna
3
4 if x > y: # jika x lebih besar dari y
5     print("Bilangan terbesar adalah X: ", x) # tampilkan bilangan terbesar adalah x : (nilai variabel x)
6 else: # jika x tidak lebih besar dari y
7     print("Bilangan terbesar adalah Y: ", y) # tampilkan bilangan terbesar adalah y : (nilai variabel y)
```

Gambar 13. Contoh code *if else statement* dengan operator

Pada gambar 13 variabel x dan y digunakan untuk menyimpan nilai input dari pengguna. Jika *statement* yang memiliki kondisi x lebih dari y divalidasi dan menghasilkan nilai *True*, maka *statement*nya akan dieksekusi yang menghasilkan *output* teks bilangan terbesar adalah x : (nilai variabel x). Jika *if statement* tidak terpenuhi maka *else statement* akan dieksekusi yang menghasilkan *output* bilangan terbesar adalah y : (nilai variabel y).

```

● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/bil_terbesar.py"
Masukkan bilangan X: 5
Masukkan bilangan Y: 4
Bilangan terbesar adalah X: 5
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/bil_terbesar.py"
Masukkan bilangan X: 8
Masukkan bilangan Y: 21
Bilangan terbesar adalah Y: 21
○ kamize@server:~/web/alpro$ 

```

Gambar 14. Hasil output pada latihan *if else statement*

Pada gambar 14 menunjukkan beberapa contoh *output* jika pengguna memasukan nilai variabel x dan y. Yang pertama pengguna memasukan nilai variabel x dengan nilai 5 dan variabel y dengan nilai 4 karena pengguna memasukan nilai variabel x lebih dari nilai variabel y. Maka *if statement* akan dieksekusi yang menghasilkan nilai *output* teks bilangan terbesar adalah x : 5. Yang kedua pengguna memasukan nilai variabel x dengan nilai 8 dan variabel y dengan nilai 21 karena pengguna memasukan nilai variabel x tidak lebih dari variabel y. Maka *if statement* tidak dieksekusi tetapi *else statement* akan dieksekusi yang menghasilkan *output* teks bilangan terbesar adalah y : 21.

3. Pembahasan Tugas Unguided

3.1. Terjemahkan *flowchart* berikut ke dalam kode program python simpan dengan nama seleksi.py:

```

alpro - seleksi.py

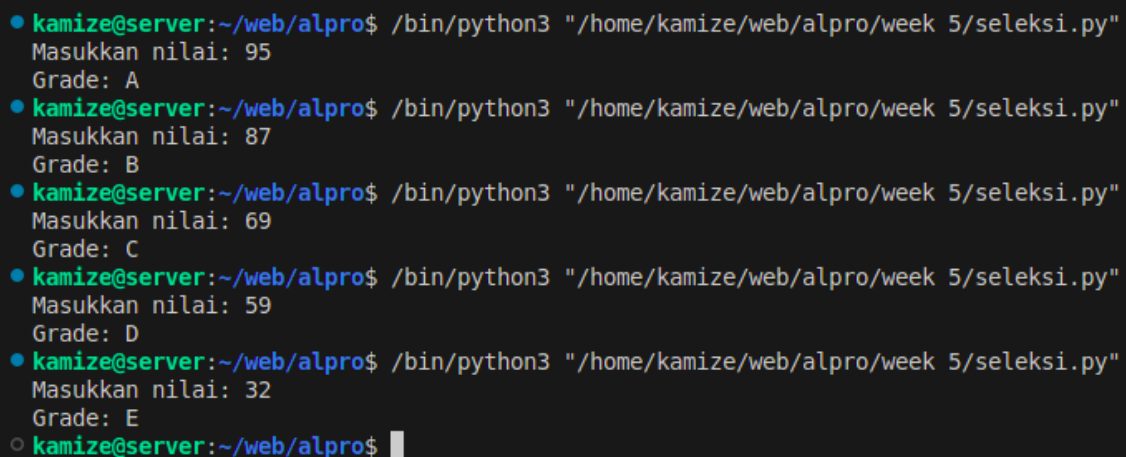
1 # Program to determine grade based on score
2 # Read the score
3 nilai = int(input("Masukkan nilai: ")) #identifier value with user input
4 # Determine the grade
5 if nilai >= 90: # # frist if statement to check the score, if the score is greater than or equal to 90
6     grade = "A" # assign grade A
7 elif nilai >= 70 and nilai < 90: # # second if statement to check the score, if the score is greater than or equal to 70
8     grade = "B" # assign grade B
9 elif nilai >= 60 and nilai < 70: # # third if statement to check the score, if the score is greater than or equal to 60
10    grade = "C" # assign grade C
11 elif nilai >= 50 and nilai < 60: # # fourth if statement to check the score, if the score is greater than or equal to 50
12    grade = "D" # assign grade D
13 else: # # else statement to check the score, if the score not meet the above conditions
14    grade = "E" # assign grade E
15 # Display the grade
16 print(f"Grade: {grade}")

```

Gambar 15. Jawaban tugas *unguided 1*

Pada gambar 15 variabel nilai digunakan untuk menyimpan *value* input dari pengguna. Jika *statement* pertama divalidasi dengan kondisi variabel nilai lebih dari sama dengan 90 divalidasi dan menghasilkan nilai *True*, maka *if statement* pertama akan dieksekusi yang menghasilkan variabel grade memiliki *value* A. Jika *statement* pertama *False*, maka *statement* kedua divalidasi dengan kondisi variabel nilai lebih dari sama dengan 70 and variabel nilai kurang dari 90 dan menghasilkan nilai *True*. Maka *if statement* kedua akan dieksekusi yang menghasilkan variabel grade memiliki *value* B. Jika *statement* kedua *False*, maka *statement* ketiga divalidasi dengan kondisi variabel nilai lebih dari sama

dengan 60 and variabel nilai kurang dari 70 dan menghasilkan nilai *True*. Maka *if statement* ketiga akan dieksekusi yang menghasilkan variabel grade memiliki *value* C. Jika *statement* ketiga *False*, maka *statement* keempat divalidasi dengan kondisi variabel nilai lebih dari sama dengan 50 and variabel nilai kurang dari 60 dan menghasilkan nilai *True*. Maka *if statement* keempat akan dieksekusi yang menghasilkan variabel grade memiliki *value* D. Jika tidak ada *statement* yang bernilai *True* maka *else statement* akan dieksekusi yang menghasilkan variabel grade memiliki *value* E. Hasil *output string* ditampilkan kelayar berupa teks Grade : (*value* dari variabel grade).



```
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/seleksi.py"
Masukkan nilai: 95
Grade: A
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/seleksi.py"
Masukkan nilai: 87
Grade: B
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/seleksi.py"
Masukkan nilai: 69
Grade: C
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/seleksi.py"
Masukkan nilai: 59
Grade: D
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/seleksi.py"
Masukkan nilai: 32
Grade: E
○ kamize@server:~/web/alpro$
```

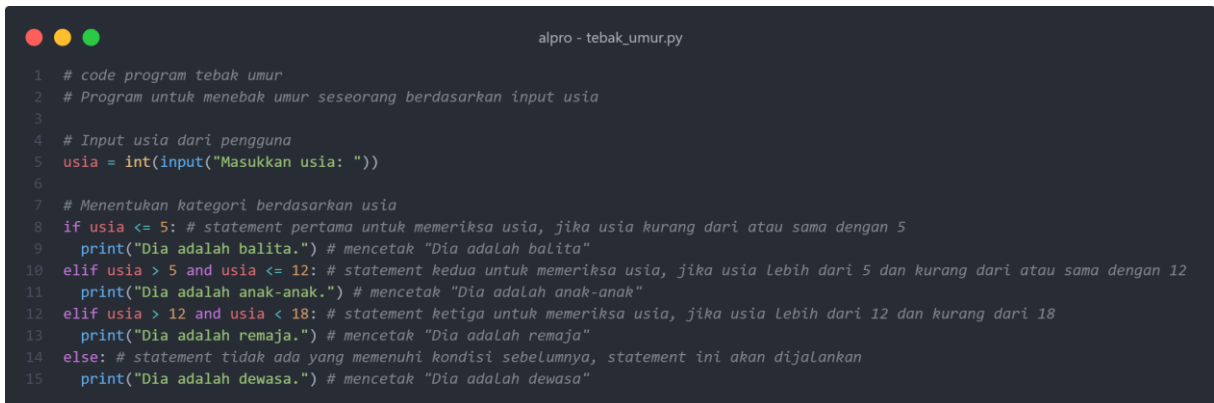
Gambar 16. Hasil output dari jawaban tugas unguided 1

Pada gambar 16 menunjukkan beberapa contoh *output* jika pengguna memasukan *value* ke variabel nilai:

1. Jika pengguna memasukan *value* 95 ke variabel nilai. Maka *if statement* pertama akan dieksekusi karena *value* variabel nilai memenuhi kondisi pertama yang menghasilkan *output string* Grade : A.
2. Jika pengguna memasukan *value* 87 ke variabel nilai. Maka *if statement* kedua akan dieksekusi karena *value* variabel nilai memenuhi kondisi kedua yang menghasilkan *output string* Grade : B.
3. Jika pengguna memasukan *value* 69 ke variabel nilai. Maka *if statement* ketiga akan dieksekusi karena *value* variabel nilai memenuhi kondisi ketiga yang menghasilkan *output string* Grade : C.
4. Jika pengguna memasukan *value* 59 ke variabel nilai. Maka *if statement* keempat akan dieksekusi karena *value* variabel nilai memenuhi kondisi pertama yang menghasilkan *output string* Grade : D.
5. Jika pengguna memasukan *value* 32 ke variabel nilai. Maka *if statement* tidak akan dieksekusi karena *value* variabel nilai tidak memenuhi kondisi yang ada, tetapi *else statement* akan dieksekusi yang menghasilkan *output string* Grade : E.

3.2. Terjemahkan skenario berikut ke dalam kode program python, simpan dengan nama `tebak_umur.py`:

- Jika usia 5 tahun ke bawah ($usia \leq 5$), maka dia adalah balita
- Jika usia di atas 5 tahun ke atas sampai 12 tahun ($usia > 5$ dan $usia \leq 12$), maka dia adalah anak-anak
- Jika usia di atas 12 tahun dan usia di bawah 18 tahun ($usia > 12$ dan $usia < 18$) maka dia adalah remaja
- Jika usia di atas 18 tahun maka dia adalah dewasa



```
alpro - tebak_umur.py
1 # code program tebak umur
2 # Program untuk menebak umur seseorang berdasarkan input usia
3
4 # Input usia dari pengguna
5 usia = int(input("Masukkan usia: "))
6
7 # Menentukan kategori berdasarkan usia
8 if usia <= 5: # statement pertama untuk memeriksa usia, jika usia kurang dari atau sama dengan 5
9     print("Dia adalah balita.") # mencetak "Dia adalah balita"
10 elif usia > 5 and usia <= 12: # statement kedua untuk memeriksa usia, jika usia lebih dari 5 dan kurang dari atau sama dengan 12
11     print("Dia adalah anak-anak.") # mencetak "Dia adalah anak-anak"
12 elif usia > 12 and usia < 18: # statement ketiga untuk memeriksa usia, jika usia lebih dari 12 dan kurang dari 18
13     print("Dia adalah remaja.") # mencetak "Dia adalah remaja"
14 else: # statement tidak ada yang memenuhi kondisi sebelumnya, statement ini akan dijalankan
15     print("Dia adalah dewasa.") # mencetak "Dia adalah dewasa"
```

Gambar 17. Jawaban tugas unguided 2

Pada gambar 17 variabel `usia` digunakan untuk menyimpan *value* dari input pengguna. Jika *statement* pertama dengan kondisi variabel `usia` kurang dari sama dengan 5 divalidasi dan menghasilkan nilai *True*, maka *if statement* pertama akan dieksekusi yang menghasilkan *output string* `Dia adalah balita`. Jika *statement* pertama *False*, maka *statement* kedua divalidasi dengan kondisi variabel umur lebih dari 5 and kurang dari sama dengan 12 dan menghasilkan nilai *True*. Maka *if statement* kedua akan dieksekusi yang menghasilkan *output string* `Dia adalah anak-anak`. Jika *statement* kedua *False*, maka *statement* ketiga divalidasi dengan kondisi variabel umur lebih dari 12 and kurang dari sama dengan 18 dan menghasilkan nilai *True*. Maka *if statement* kedua akan dieksekusi yang menghasilkan *output string* `Dia adalah remaja`. Jika tidak ada *statement* yang bernilai *True* maka *else statement* akan dieksekusi yang menghasilkan *output string* `Dia adalah dewasa`.

```

● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/tebak_umur.py"
Masukkan usia: 5
Dia adalah balita.
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/tebak_umur.py"
Masukkan usia: 10
Dia adalah anak-anak.
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/tebak_umur.py"
Masukkan usia: 15
Dia adalah remaja.
● kamize@server:~/web/alpro$ /bin/python3 "/home/kamize/web/alpro/week 5/tebak_umur.py"
Masukkan usia: 20
Dia adalah dewasa.
○ kamize@server:~/web/alpro$ █

```

Gambar 18. Hasil output dari jawaban tugas unguided 2

Pada gambar 18 menunjukkan beberapa contoh *output* jika pengguna memasukan *value* ke variabel nilai:

1. Jika pengguna memasukan *value* 4 ke variabel usia. Maka *if statement* pertama akan dieksekusi karena *value* variabel usia memenuhi kondisi pertama yang menghasilkan *output string* dia adalah balita.
2. Jika pengguna memasukan *value* 11 ke variabel usia. Maka *if statement* kedua akan dieksekusi karena *value* variabel usia memenuhi kondisi kedua yang menghasilkan *output string* dia adalah anak-anak.
3. Jika pengguna memasukan *value* 16 ke variabel usia. Maka *if statement* ketiga akan dieksekusi karena *value* variabel usia memenuhi kondisi ketiga yang menghasilkan *output string* dia adalah remaja.
4. Jika pengguna memasukan *value* 21 ke variabel usia. Maka *if statement* tidak akan dieksekusi karena *value* variabel usia tidak memenuhi kondisi yang ada, tetapi *else statement* akan dieksekusi yang menghasilkan *output string* Dia adalah remaja.

4. Ringkasan

Modul ini membahas dasar-dasar python tentang *selection*. *Selection* adalah suatu opsi dimana terdapat kondisi tertentu yang harus dipenuhi oleh program untuk menjalankan suatu perintah. Terdapat beberapa jenis *selection* didalam bahasa python seperti *if*, *if-else*, *if-else*, *if-elif-else*. *Selection* juga dapat digunakan untuk membuat *selection* dengan kondisi lebih kompleks dengan menggunakan *nested statement* atau pernyataan bersarang.

If statement adalah *selection* yang dieksekusi jika *statement* memiliki kondisi *True*, tetapi jika *statement* memiliki kondisi *False statement* itu tidak akan di eksekusi. *if-else statement* adalah *selection* yang dieksekusi jika kondisi *True* dan jika kondisi *False* maka *else statement* akan dieksekusi. *If-elif statement* adalah *selection* jika kondisi pertama *True* maka kondisi pertama akan dieksekusi, jika kondisi pertama *False* maka kondisi kedua dengan kondisi *True* akan dikesekusi dan seterusnya, jika tidak ada ada kondisi yang

terpenuhi atau *False* maka selection tidak akan dieksekusi. If-elif-else adalah *selection* if-elif yang tidak ada kondisi yang terpenuhi atau *False* maka *else statement* akan dieksekusi.

Laporan ini juga berisi tugas praktik yang mencakup implementasi selection pada bahasa python. Salah satu contoh latihan adalah penggunaan selection untuk mengetahui grade berdasarkan besaran *value* nilai dan ketersediaan tiket beserta harganya sesuai dengan *value* variabel umur. Selain itu terdapat juga tugas mandiri untuk memahami lebih dalam tentang penggunaan selection dengan cara diberi studi kasus yang dikerjakan dengan mandiri. Salah satu studi kasusnya berupa menentukan grade nilai berdasarkan *value* variabel nilai yang dimasukkan oleh pengguna dan menebak apakah usia pengguna termasuk kategori balita, anak-anak, remaja, atau dewasa berdasarkan input usia oleh pengguna.

Dari laporan ini disimpulkan bahwa pemahaman konsep selection dalam Python sangat penting. Dengan menguasainya programmer dapat menuliskan kode sesuai dengan kondisi tertentu. Sehingga kode yang ditulis akan menghasilkan *output* sesuai dengan kondisi yang telah ditentukan, lebih fleksibel, dan lebih mudah dipahami.