

**LAPORAN PRAKTIKUM 2**  
**“BOOLEAN, PERULANGAN, & FUNGSI”**



Disusun oleh:

Nama : Muhammad Fathurrahman  
NIM : 2024573010004  
Kelas/ Semester : TI 2C/ 1  
Mata Kuliah : Praktikum Metode Numerik  
No. Prak : 02/PMN/TI/2025  
Program Studi : Teknik Informatika

**TEKNOLOGI INFORMASI DAN KOMPUTER**  
**POLITEKNIK NEGERI LHOKSEUMAWE**  
**2025**

## LEMBAR PENGESAHAN

Nomor Praktikum : 02/PNM/TI/2025  
Nama Praktikum : Laporan Praktikum 2 Boolean, Perulangan & Fungsi  
Nama Praktika : Muhammad Fathurrahman  
NIM : 2024573010004  
Kelas : TI 2C  
Jurusan : Teknologi Informasi dan Komputer  
Prodi : Teknik Informatika  
Tanggal Praktikum : 29 September 2025  
Tanggal Penyerahan : 6 Oktober 2025  
Nilai :  
Keterangan :

Buket Rata, 6 Oktober 2025

Dosen Pembimbing,

**Radhiyatammardhiyyah, SST., M.Sc.**

**NIP. 199208262022032011**

## **KATA PENGANTAR**

Assalamualaikum warahmatullahi wabarakatuh

Puji syukur ke hadirat Tuhan Yang Maha Esa. Atas rahmat dan hidayah NYA, penulis dapat menyelesaikan Tugas Praktikum tentang “Boolean, Perulangan & Fungsi” dengan tepat waktu.

Makalah disusun untuk memenuhi tugas Mata Kuliah Praktikum Metode Numerik. Selain itu, makalah ini bertujuan menambah wawasan tentang Boolean, Perulangan & Fungsi pada Bahasa pemrograman Python

Penulis mengucapkan terima kasih kepada Ibu Radhiyatammardhiyyah selaku guru Mata Pelajaran Sejarah. Ucapan terima kasih juga disampaikan kepada semua pihak yang telah membantu diselesaikannya makalah ini.

Penulis menyadari makalah ini masih jauh dari sempurna. Oleh sebab itu, saran dan kritik yang membangun diharapkan demi kesempurnaan makalah ini.

Wassalamualaikum warahmatullahi wabarakatuh

Lhokseumawe, 2 Oktober 2025

Muhammad Fathurrahman

## DAFTAR ISI

<b>LEMBAR PENGESAHAN .....</b>	<b>i</b>
<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1    Dasar Teori .....	1
<b>BAB 2 PRAKTIKUM.....</b>	<b>2</b>
2.1    Nilai Boolean.....	2
2.2    Mengevaluasi Nilai dan Variabel .....	2
2.3    Kebanyakan Nilai adalah Benar .....	2
2.4    Beberapa Nilai Salah .....	2
2.5    Fungsi dapat Mengembalikan Boolean .....	2
2.6    List.....	3
2.7    Tuple.....	3
2.8    Set.....	3
2.9    Dictionary .....	3
2.10   Pernyataan If .....	3
2.11   Elif.....	4
2.12   Else .....	4
2.13   If Else Pendek.....	4
2.14   And .....	4
2.15   Or.....	4
2.16   Nested If .....	5
2.17   Perulangan While .....	5
2.18   Pernyataan Break.....	5
2.19   Pernyataan Continue .....	5
2.20   Perulangan For .....	6
2.21   Perulangan For Menggunakan Fungsi range() .....	6
2.22   Else di Perulangan For .....	6
2.23   Perulangan For Bersarang .....	6
2.24   Membuat dan Memanggil Fungsi .....	6
2.25   Argumen.....	7
2.26   Argumen Kata Kunci .....	7

<b>BAB 3 ANALISIS .....</b>	<b>8</b>
<b>BAB 4 KESIMPULAN .....</b>	<b>11</b>
<b>DAFTAR PUSTAKA.....</b>	<b>12</b>

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Dasar Teori**

Boolean adalah tipe data yang hanya memiliki dua nilai logika, yaitu true (benar) dan false (salah). Dalam pemrograman, Boolean digunakan untuk membuat keputusan atau percabangan logika berdasarkan kondisi tertentu. Misalnya, dalam pernyataan if-else, program akan mengevaluasi ekspresi Boolean untuk menentukan apakah blok kode tertentu akan dijalankan atau tidak. Konsep ini sangat penting dalam pengendalian alur program karena memungkinkan komputer untuk “memilih” tindakan berdasarkan kondisi logis.

Perulangan atau looping adalah struktur kontrol dalam pemrograman yang digunakan untuk mengeksekusi sekumpulan perintah secara berulang selama kondisi tertentu terpenuhi. Tujuan utama dari perulangan adalah untuk menghemat waktu dan menghindari penulisan kode yang berulang-ulang. Ada beberapa jenis perulangan, seperti for, while, dan do-while, yang masing-masing memiliki cara kerja dan penggunaan berbeda tergantung pada kebutuhan. Dengan perulangan, programmer dapat melakukan proses berulang seperti menghitung data dalam array atau menampilkan output berulang tanpa menulis kode yang sama berulang kali.

Fungsi adalah blok kode yang dirancang untuk melakukan tugas tertentu dan dapat digunakan kembali di berbagai bagian program. Fungsi membantu meningkatkan efisiensi dan keteraturan kode karena memungkinkan programmer untuk memecah program besar menjadi bagian-bagian kecil yang lebih mudah dikelola. Sebuah fungsi biasanya memiliki nama, parameter (opsional), dan nilai balik (return value). Dengan menggunakan fungsi, programmer dapat menghindari duplikasi kode serta mempermudah proses debugging dan pemeliharaan program. Fungsi juga meningkatkan modularitas program sehingga setiap bagian memiliki tanggung jawab yang jelas dan terpisah.

## BAB 2

### PRAKTIKUM

#### 2.1 Nilai Boolean

Code:

```
print(10 < 9)
print(10 == 9)
print(10 > 9)
```

Output:

```
False
False
True
```

#### 2.2 Mengevaluasi Nilai dan Variabel

Code:

```
print(bool("Hello"))
print(bool(15))
```

Output:

```
True
True
```

#### 2.3 Kebanyakan Nilai adalah Benar

Code:

```
print(bool("abc"))
print(bool(123))
print(bool(["apple", "cherry", "banana"]))
```

Output:

```
True
True
True
```

#### 2.4 Beberapa Nilai Salah

Code:

```
print(bool(False))
print(bool(None))
print(bool(0))
print(bool(""))
print(bool(()))
print(bool([]))
print(bool({}))
```

Output:

```
False
False
False
False
False
False
False
```

#### 2.5 Fungsi dapat Mengembalikan Boolean

Code:

```
def myFunction():
    return True
if myFunction():
    print("YES!")
else:
    print("NO!")
```

Output:

```
YES!
```

## 2.6 List

Code:

```
listbuah = ["apel",  
            "jeruk",  
            "pisang"]  
print(listbuah)
```

Output:

```
['apel', 'jeruk', 'pisang']
```

## 2.7 Tuple

Code:

```
tuplebuah = ("apel",  
             "jeruk",  
             "pisang")  
print(tuplebuah)
```

Output:

```
('apel', 'jeruk', 'pisang')
```

## 2.8 Set

Code:

```
setbuah = {"apel",  
           "jeruk",  
           "pisang"}  
print(setbuah)
```

Output:

```
{'jeruk', 'apel', 'pisang'}
```

## 2.9 Dictionary

Code:

```
thisdict = {  
    "brand": "Lenovo",  
    "model": "ThinkPad",  
    "year": 2020  
}  
print(thisdict)
```

Output:

```
{'brand': 'Lenovo', 'model': 'ThinkPad', 'year': 2020}
```

## 2.10 Pernyataan If

Code:

```
a = 33  
b = 200  
if b > a:  
    print("b lebih besar dari a")
```

Output:

```
b lebih besar dari a
```



### 2.11 Elif

Code:

```
a = 33
b = 33
if b > a:
    print("b lebih besar dari a")
elif a == b:
    print("a sama dengan b")
```

Output:

```
a sama dengan b
```

### 2.12 Else

Code:

```
a = 200
b = 33
if b > a:
    print("b lebih besar dari a")
elif a == b:
    print("a sama dengan b")
else:
    print("a lebih besar dari b")
```

Output:

```
a lebih besar dari b
```

### 2.13 If Else Pendek

Code:

```
a = 4
b = 28
print("A") if a > b else print("B")
```

Output:

```
B
```

### 2.14 And

Code:

```
a = 198
b = 34
c = 300
if a > b and a < c:
    print("dua dua benar")
```

Output:

```
dua dua benar
```

### 2.15 Or

Code:

```
a = 198
b = 34
c = 300
if a > b or a > c:
    print("satu kondisi benar")
```

Output:

```
satu kondisi benar
```

## 2.16 Nested If

Code:

```
a = 8
if a > 5:
    print("a diatas 5")
    if a > 10:
        print("a diatas 10")
    else:
        print("tetapi tidak diatas 10")
```

Output:

```
a diatas 5
tetapi tidak diatas 10
```

## 2.17 Perulangan While

Code:

```
i = 1
while i ≤ 5:
    print(i)
    i += 1
```

Output:

```
1
2
3
4
5
```

## 2.18 Pernyataan Break

Code:

```
i = 1
while i < 6:
    print(i)
    if i == 3:
        break
    i += 1
```

Output:

```
1
2
3
```

## 2.19 Pernyataan Continue

Code:

```
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```

Output:

```
1
2
4
5
6
```

## 2.20 Perulangan For

Code:

```
fruits = ["apel", "pisang", "ceri"]
for x in fruits:
    print(x)
```

Output:

```
apel
pisang
ceri
```

## 2.21 Perulangan For Menggunakan Fungsi range()

Code:

```
for x in range(6):
    print(x)
```

Output:

```
0
1
2
3
4
5
```

## 2.22 Else di Perulangan For

Code:

```
for x in range(6):
    print(x)
else:
    print("Loop completed")
```

Output:

```
0
1
2
3
4
5
Loop completed
```

## 2.23 Perulangan For Bersarang

Code:

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]

for x in adj:
    for y in fruits:
        print(x, y)
```

Output:

```
red apple
red banana
red cherry
big apple
big banana
big cherry
tasty apple
tasty banana
tasty cherry
```

## 2.24 Membuat dan Memanggil Fungsi

Code:

```
def my_func():
    print("ini adalah fungsi")

my_func()
```

Output:

```
ini adalah fungsi
```

## 2.25 Argumen

Code:

```
def my_function(fname, lname):  
    print(fname + " " + lname)  
my_function("Hu", "Tao")
```

Output:

```
Hu Tao
```

## 2.26 Argumen Kata Kunci

Code:

```
def my_function(anak3, anak2, anak1):  
    print(anak3 + " adalah anak ketiga")  
my_function(anak1 = "Nina", anak2 = "Nino", anak3 = "Agus")
```

Output:

```
Agus adalah anak ketiga
```

## **BAB 3**

### **ANALISIS**





## **BAB 4**

### **KESIMPULAN**

Dari praktikum yang telah dilakukan, dapat disimpulkan bahwa konsep dasar seperti Boolean, perulangan, dan fungsi sangat penting dalam pemrograman. Boolean membantu menentukan alur logika program dengan nilai benar atau salah. Struktur percabangan seperti if, elif, dan else digunakan untuk mengambil keputusan berdasarkan kondisi tertentu, sedangkan operator logika seperti and dan or membantu menggabungkan beberapa kondisi sekaligus.

Selain itu, perulangan seperti for dan while memudahkan kita menjalankan kode secara berulang tanpa menulis perintah yang sama berkali-kali. Fungsi berperan penting untuk membuat kode lebih rapi, efisien, dan mudah dipahami karena dapat digunakan kembali di berbagai bagian program. Dengan memahami ketiga konsep ini, kita dapat membuat program yang lebih terstruktur dan mudah dikembangkan.



## **DAFTAR PUSTAKA**

Ma'arif, A. 2020. Buku Ajar Pemrograman Lanjut Bahasa Pemrograman Python. Yogyakarta: Universitas Ahmad Dahlan.