

LAPORAN PRAKTIKUM STRUKTUR DATA
PERCABANGAN (IF-ELSE, SWITCH-CASE, DAN DO WHILE)
MENGGUNAKAN PYTHON



Disusun Oleh :

Nama	: Muhammad Nuh
NIM	: 220504027
Unit	: 01
Prodi	: Informatika
Mata Kuliah	: Struktur Data dan Algoritma
Dosen Pengampu	: NURUL FADILLAH, S. ST., M.T

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS SAMUDRA
2023



**KEMENTERIAN PENDIDIKAN KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS SAMUDRA
FAKULTAS TEKNIK**

Jalan Prof. Dr. Syarif Thayeh, Meurandeh, Langsa-Aceh

Telp (0641) 426534, Fax (0641) 426535, 7445155

Laman: <http://www.unsam.ac.id> Email: info@unsam.ac.id Kode Pos 24416

Nama : Muhammad Nuh

NIM : 220504027

Laporan Studi : Informatika

Laporan Praktikum : Struktur Data Percabangan (If-Else, Switch-Case, dan
Do-While) Menggunakan Python

Isi dan format laporan ini telah disetujui serta disahkan sebagai syarat untuk melengkapi pelaksanaan praktikum Struktur Data Percabangan (If-Else, Switch-Case, dan Do-While) Menggunakan Python pada Program Studi Informatika, Fakultas Teknik, Universitas Samudra, Pada :

Hari : Senin

Tanggal : 16 Oktober 2023

Nilai :

Mengetahui
Pembimbing praktikum I

Mengetahui
Pembimbing Praktikum II

NURUL FADILLAH, S.ST. , M.T.
NIP. 198910012022032009

DEFY SYAFIRA, S.T. , M.Kom.

KATA PENGANTAR

Puji syukur atas kehadiran Allah Swt. atas segala rahmat-Nya sehingga penyusun dapat membuat laporan ini dapat tersusun sampai selesai. Tidak lupa pula shalawat dan salam atas kehadiran Nabi besar Muhammad saw yang telah membawa kita dari alam jahiliah ke alam yang penuh ilmu pengetahuan seperti yang kita rasakan saat ini.

Dengan terselesainya laporan ini, maka tidak lupa saya mengucapkan banyak terima kasih kepada semua pihak yang terlibat dalam penyusunan laporan ini, khususnya kepada Ibu Nurul Fadillah, S.ST.,M.T. selaku Dosen Pengampu.

Saya sangat berharap semoga laporan ini dapat menambah pengetahuan dan pengalaman bagi pembaca. Bahkan saya berharap lebih jauh lagi agar malah ini bisa pembaca praktikkan juga. Bagi saya sebagai penyusun mersa bahwa masih banyak kekurangan dalam penyusunan laporan ini karena keterbatasan pengetahuan dan pengalaman saya. Untuk itu saya sangat mengharapkan kritik dan saran yang membangun dari pembca demi kesempurnaan laporan ini

Langsa, 15 Oktober 2023

Muhammad Nuh

NIM. 220504027

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	v
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	1
1.3 Tujuan Laporan	2
BAB II DASAR TEORI.....	3
2.1 Kajian Pustaka.....	3
BAB III METODOLOGI PRAKTIKUM.....	5
3.1 Algoritma Program 1	5
3.2 Flowchart Program 1	5
3.3 Algoritma Program 2.....	6
3.4 Flowchart Program 2	7
3.5 Algoritma Program 3.....	8
3.6 Flowchart Program 3	8
3.7 Algoritma Program 4.....	9
3.8 Flowchart Program 4	10
3.9 Algoritma Tugas Program.....	11
3.10 Flowchart Tugas Program	12
3.11 Alat dan Bahan	13
BAB IV HASIL PRAKTIKUM DAN ANALISA	14
4.1 Analisis Program 1	14
4.2 Analisis Program 2	15
4.3 Analisis Program 3.....	17
4.4 Analisis Program 4.....	19
4.5 Analisis Program Tugas Program	22

BAB V PENUTUP	26
5.1 Kesimpulan.....	26
LAMPIRAN DOKUMENTASI	27
DAFTAR PUSTAKA	28

DAFTAR GAMBAR

Gambar 4.1 Program 1	14
Gambar 4.2 Output (ya) Program 1.....	15
Gambar 4.3 Output (tidak) Program	15
Gambar 4.4 Program 2	15
Gambar 4.5 Output Program 2	17
Gambar 4.6 Program 3	17
Gambar 4.7 Output 1 Program 3	19
Gambar 4.8 Output 2 Program 3	19
Gambar 4.9 Program 4	19
Gambar 4.10 Output Program 4	21
Gambar 4.11 Tugas Program If-Elif-Else	22
Gambar 4.12 Tugas Program Alternatif Do-While.....	23
Gambar 4.13 Tugas Program Alternatif Switch-Case	23
Gambar 4.14 Dokumentasi Praktikum	27

BAB I

PENDAHULUAN

1.1 Latar Belakang

Percabangan adalah salah satu konsep fundamental dalam pemrograman python yang memainkan peran krusial dalam memberikan dinamika dan kontrol terhadap alur eksekusi program. Dalam pengembangan perangkat lunak, seringkali kita dihadapkan pada kebutuhan untuk membuat keputusan berdasarkan kondisi tertentu. Inilah dimana struktur percabangan menjadi kunci untuk memahami bagaimana program dapat merespon secara adaptif terhadap situasi yang berbeda.

Dalam pemrograman python, percabangan memungkinkan pengembang untuk merancang logika program yang cerdas dan fleksibel. Dalam menggunakan pernyataan kondisional seperti if, elif, dan else, pengembang dapat memberikan arahan spesifik kepada program untuk menjalankan blok kode tertentu berdasarkan evaluasi kondisi-kondisi tertentu. Sebagai contoh, percabangan dapat digunakan untuk menangani kasus-kasus khusus, mengontrol aliran eksekusi, atau memberikan respons yang sesuai tergantung pada input yang diberikan.

Dengan memiliki pemahaman yang kuat tentang konsep percabangan, pengembang dapat menulis kode yang lebih bersih, lebih mudah dibaca, dan lebih responsive terhadap perubahan keadaan. Oleh karena itu, pemahaman tentang percabangan merupakan keterampilan dasar yang penting bagi setiap pengembang python yang ingin membuat program yang efektif dan adaptif.

1.2 Rumusan Masalah

Percabangan dalam pengembangan perangkat lunak menggunakan python menjadi krusial untuk mengatasi berbagai tantangan dan memenuhi kebutuhan spesifik. Salah satu aspek penting adalah kemampuan menanggapi kondisi atau situasi tertentu dengan eksekusi kode yang sesuai.

1.3 Tujuan Laporan

Tujuan dari pembahasan tentang percabangan dalam python adalah memberikan pemahaman mendalam kepada pembaca mengenai konsep, kegunaan, dan penerapan praktis dari percabangan dalam pengembangan perangkat lunak. Dengan memahami konsep dan penggunaan percabangan, pembaca diharapkan dapat meningkatkan keterampilan pemrograman python, memahami bagaimana membuat keputusan dalam kode program, dan merancang solusi yang lebih responsive dan adaptif terhadap berbagai kondisi dan situasi.

BAB II

DASAR TEORI

2.1 Kajian Pustaka

Percabangan adalah salah satu bentuk kontrol program. Percabangan berguna untuk mengatur alur jalannya program sesuai dengan suatu kondisi yang terpenuhi. Dalam python, terdapat tiga keywords untuk melakukan percabangan, yaitu if, elif, dan else.

1. IF

If merupakan penanda awal suatu percabangan. Suatu program akan menjalankan alur tertentu jika memenuhi kondisi yang ditentukan pada blok if.

Contoh:

```
X = 1
Y = 2
if (X<Y) :
print ("Benar")
```

Karena satu kurang dari dua, maka kondisi pada blok if terpenuhi (bernilai true) sehingga blok kode selanjutnya akan dijalankan. Jadi program akan mencetak kata "Benar".

2. ELIF

Elif atau kependekan dari else if adalah suatu percabangan setelah if. Sama seperti if, elif akan menjalankan program dibawahnya jika kondisi dalam blok if atau elif sebelumnya tidak terpenuhi (bernilai false) dan kondisi pada blok elif tersebut terpenuhi.

Contoh:

```
X = 12
Y = 25
if (X > Y) :
print ("X lebih besar")
```

```
elif(X == Y):  
    print("X sama dengan Y")  
elif(X < Y):  
    print("Y lebih besar")
```

Karena 25 lebih dari 12, maka pernyataan $X > Y$ tidak terpenuhi sehingga kode blok di bawahnya tidak dijalankan dan berlanjut ke blok percabangan elif. Karena elif 12 tidak sama dengan 25, maka pernyataan $X == Y$ juga tidak terpenuhi sehingga kode blok di bawahnya tidak dijalankan karena 12 kurang dari 25, maka pernyataan $X < Y$ terpenuhi sehingga kode blok di bawahnya akan dijalankan oleh program dan mencetak "Y lebih besar".

3. ELSE

Else merupakan kondisi penutup dari sebuah percabangan. Dapat dikatakan bahwa else adalah solusi percabangan terakhir, sehingga else tidak memerlukan syarat seperti halnya if dan elif.

Contoh:

```
X = 19  
if(X == 10):  
    print("Sepuluh")  
elif(X == 20):  
    print("Dua puluh")  
elif(X == 50):  
    print("Lima puluh")  
else:  
    print("angka lain")
```

Program di atas akan mencetak kalimat "angka lain" karena nilai X tidak memenuhi syarat pada percabangan if dan elif, sehingga program masuk pada kode blok else.

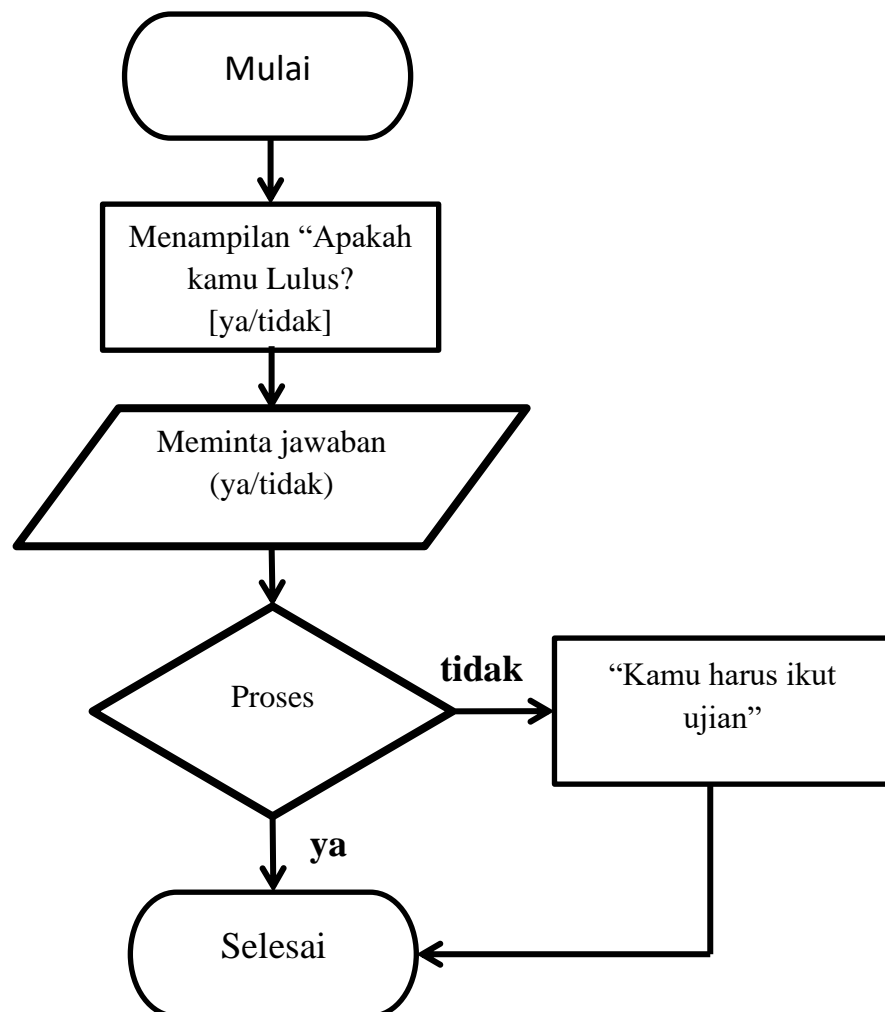
BAB III

METODOLOGI PENELITIAN

3.1 Algoritma Program 1

1. Mulai
2. Meminta pengguna untuk memasukkan jawaban (ya/tidak) apakah mereka lulus atau tidak.
3. Jika jawaban pengguna adalah “tidak”, output “Kamu harus ikut ujian.”
4. Jika jawaban pengguna adalah “ya”, program selesai.
5. Selesai.

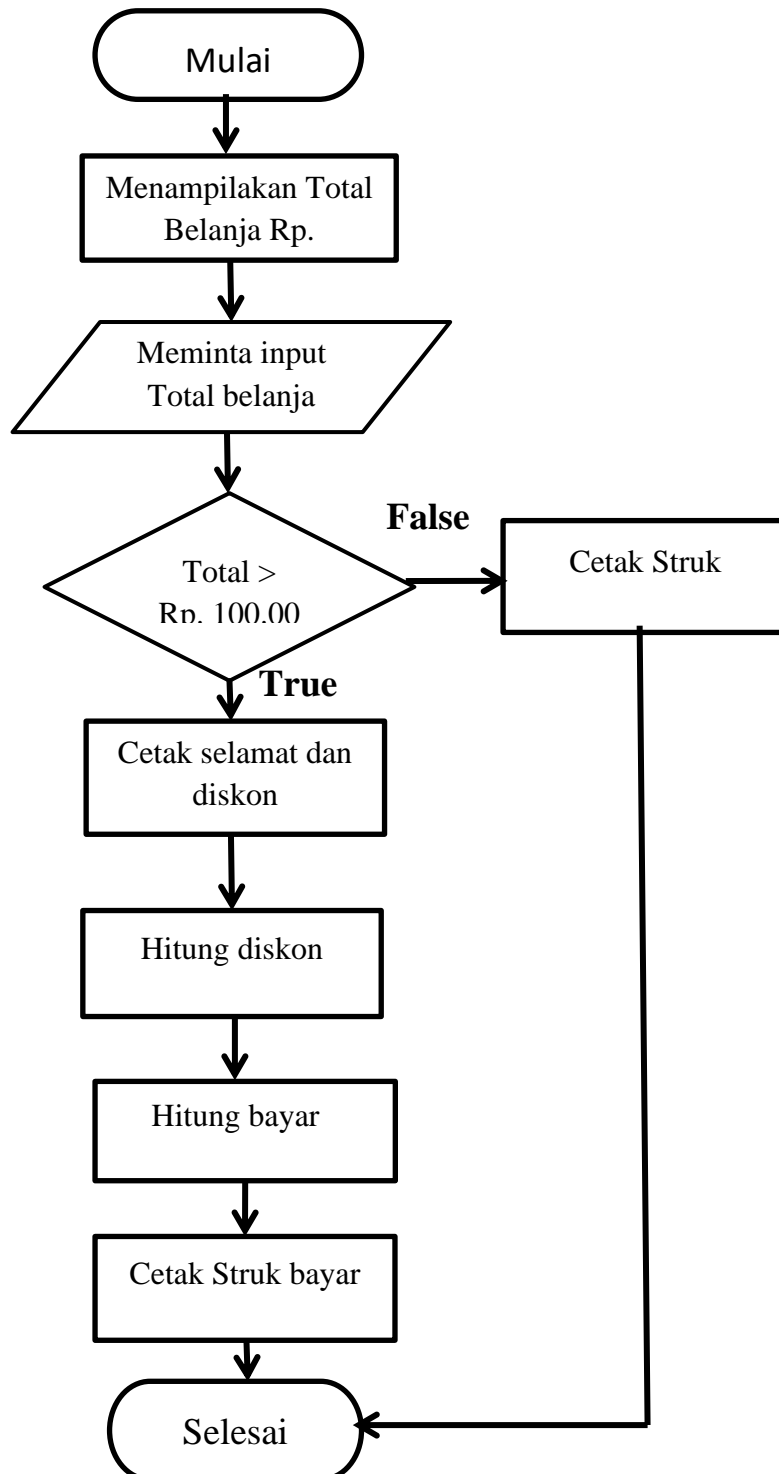
3.2 Flowchart Program 1



3.3 Algoritma Program 2

1. Mulai
2. Meminta pengguna untuk memasukkan total belanja (Total belanja:Rp).
3. Konversi total belanja ke dalam tipe data integer
(bayar=int(total_belanja)).
4. Jika total belanja lebih dari Rp.100.000, lanjut ke langkah 5. Jika tidak lanjut ke langkah 8.
5. Cetak”Selamat Karena Belanja > Rp. 100.000 anda mendapat voucher Makan.”
6. Cetak “dan diskon 5%.”
7. Hitung diskon (diskon=total belanja*5/100)
8. Hitung total yang harus dibayar (bayar=total belanja – diskon)
9. Cetak struk:
 - Cetak “total yang harus dibayar: Rp”
 - Cetak “Terima kasih sudah berbelanja di toko kami”
10. Selesai

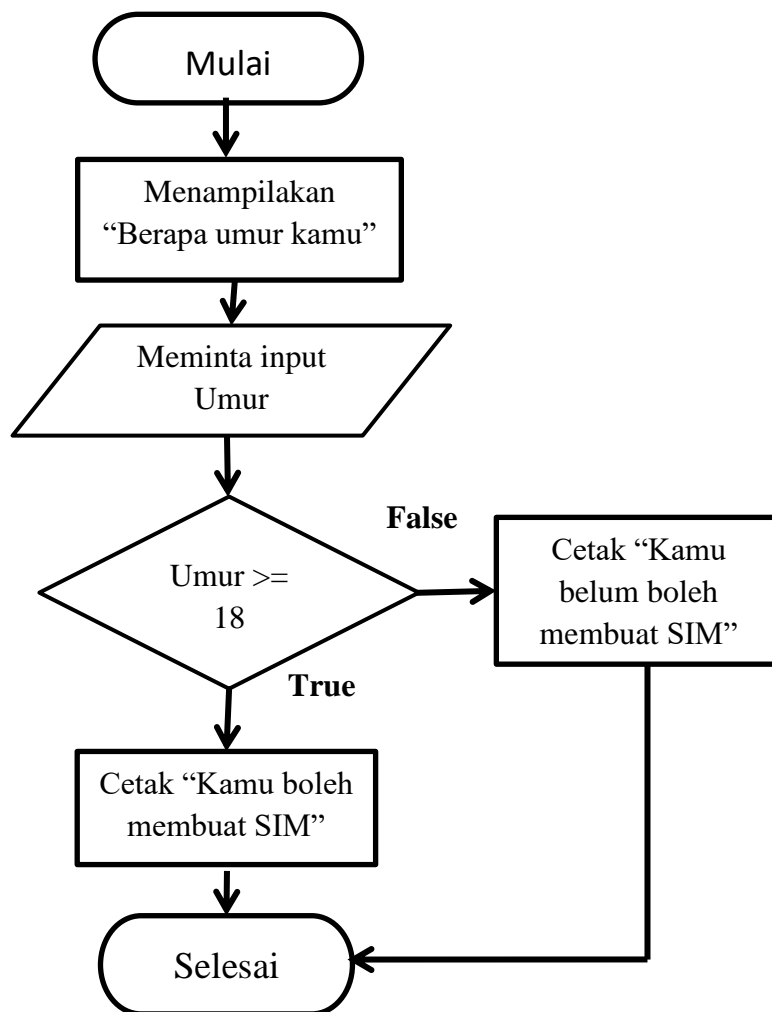
3.4 Flowchart Program 2



3.5 Algoritma Program 3

1. Mulai
2. Meminta pengguna untuk memasukkan umur
3. Konversi input umur ke tipe data integer
4. Jika umur lebih besar atau sama dengan 18, lanjut ke langkah 5. Jika tidak, lanjut ke langkah 7.
5. Cetak “Kamu boleh membuat SIM.”
6. Jika umur kurang dari 18:
7. Cetak “Kamu belum boleh membuat SIM.”
8. Selesai

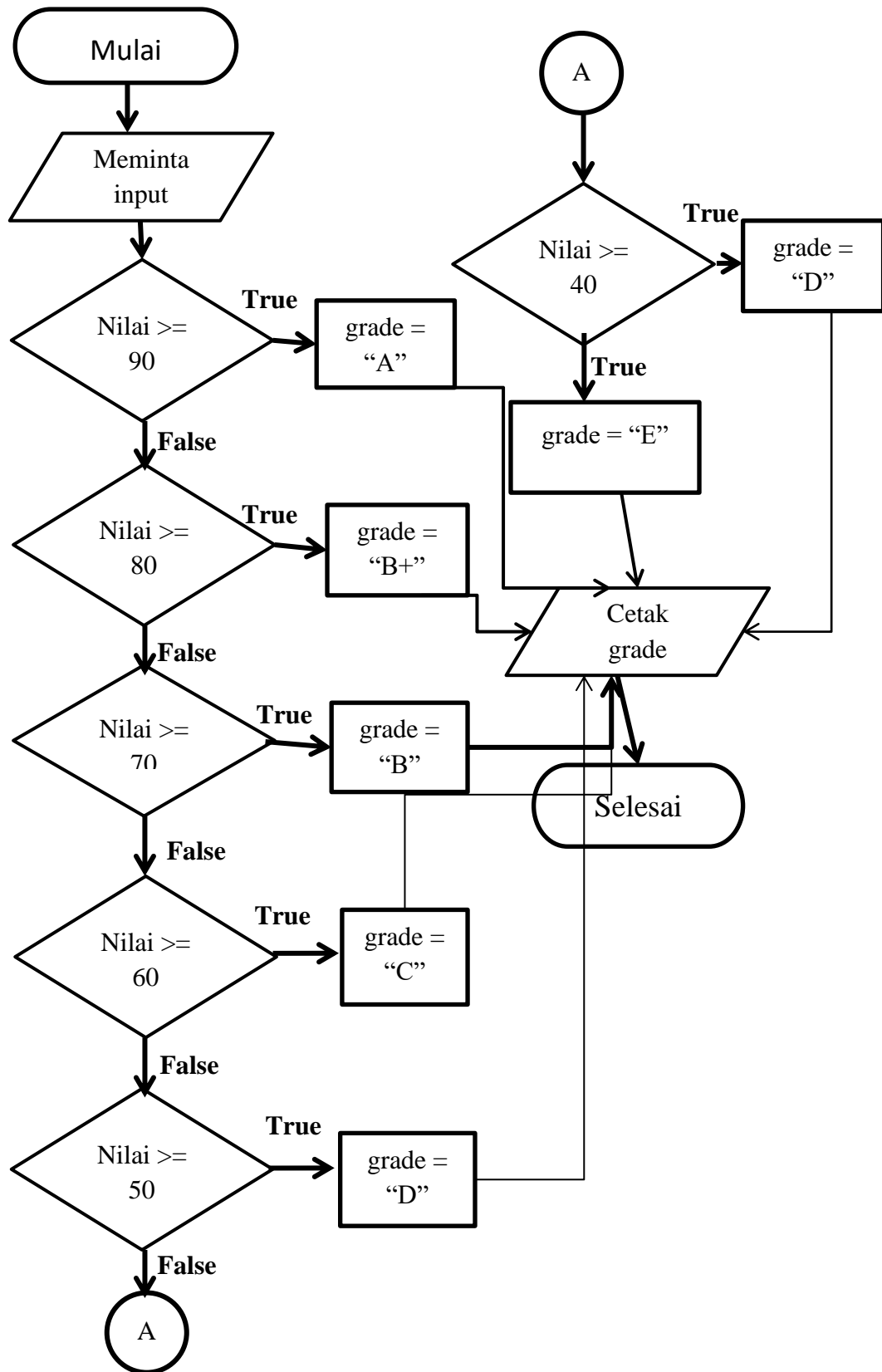
3.6 Flowchart Program 3



3.7 Flowchart Program 4

1. Mulai
2. Minta pengguna untuk memasukkan nilai
3. Inisialisasi variabel grade dengan nilai awal “ Anda tidak lulus.”
4. Jika nilai ≥ 90 , set grade =”A”
5. Jika nilai ≥ 90 , set grade =”A”
6. Jika nilai ≥ 80 , set grade =”B+”
7. Jika nilai ≥ 70 , set grade =”B”
8. Jika nilai ≥ 60 , set grade =”C+”
9. Jika nilai ≥ 50 , set grade =”C”
10. Jika nilai ≥ 40 , set grade =”D”
11. Cetak hasil grade
12. Selesai

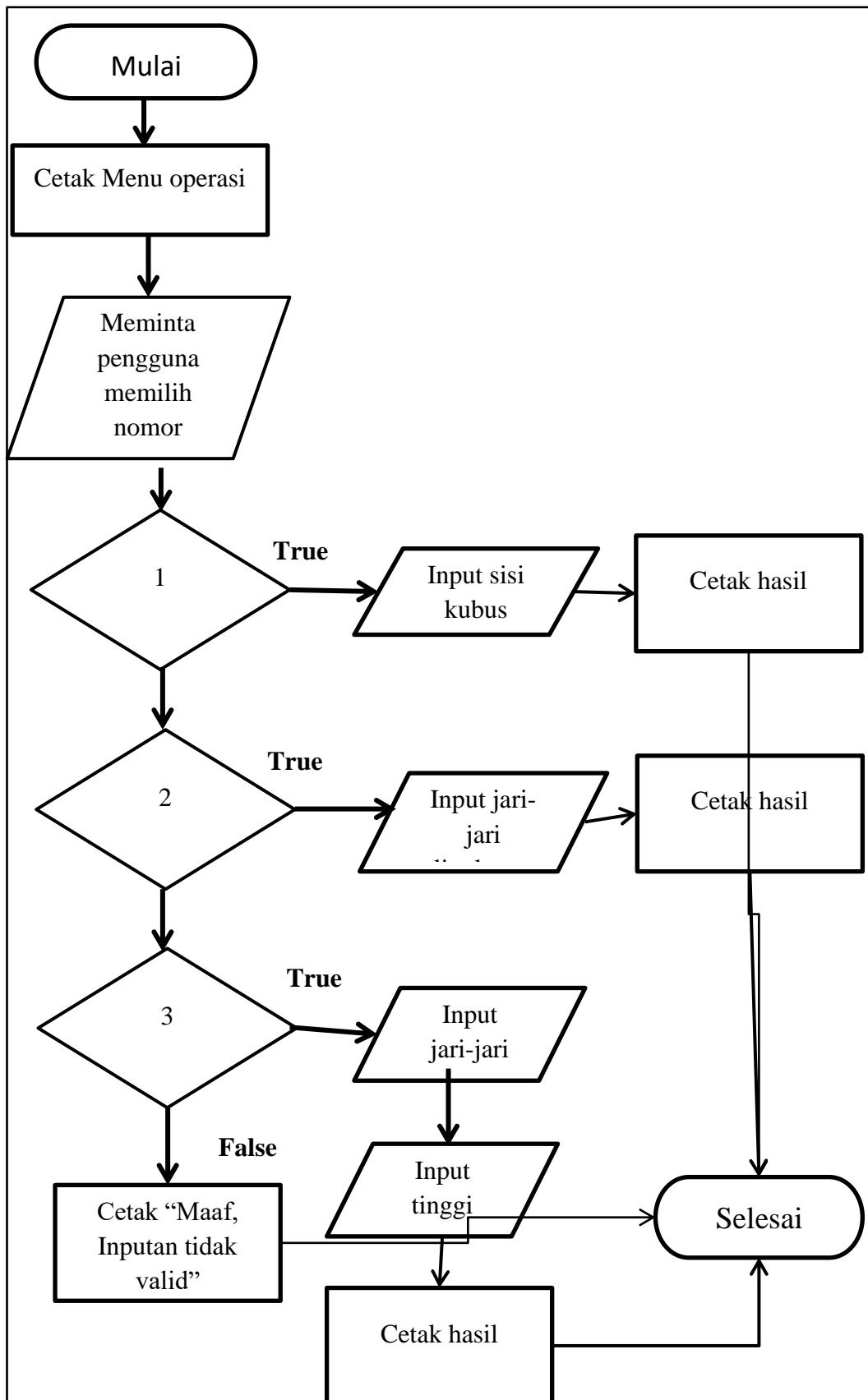
3.8 Flowchart Program 4



3.9 Algoritma Tugas Program

1. Mulai
2. Cetak menu operasi
3. Meminta pengguna memilih nomor operasi (1/2/3)
4. Inisialisasi variabel hasil dengan nilai awal “Maaf, inputan tidak valid”
5. Jika pilhan ‘1’, lanjut ke langkah 6. Jika pilihan adalah ‘2’, lanjut ke langkah 9. Jika pilihan adalah ‘3’, lanjut ke langkah 12.\
6. Meminta pengguna memasukkan panjang jari-jari lingkaran
7. Hitung volume kubus
8. Jika pilihan ‘2’, lanjut ke langkah 9. Jika tidak lanjut ke langkah 12.
9. Meminta pengguna memasukkan panjang jari-jari lingkaran
10. Hitung luas lingkaran
11. Jika piliha ‘3’, lanjut ke langkah 12. Jika tidak, lanjut ke langkah 15.
12. Meminta pengguna memasiukkan panjang jari-jari silinder dan tinggi silinder
13. Hitung volume silinder
14. Cetak hasil
15. Selesai

3.10 Flowchart Tugas Program



3.11 Alat dan Bahan

1. Komputer: Diperlukan sebuah komputer (laptop atau desktop) dengan sistem operasi seperti Windows, macOS, atau Linux.
2. Text editor atau Integrated Development Environment (IDE) yang mendukung penulisan dan eksekusi kode python.
3. Koneksi internet jika ada dibutuhkan untuk mengakses referensi program
4. Contoh-contoh kasus atau scenario yang akan digunakan sebagai dasar praktikum.

BAB IV

HASIL PRAKTIKUM DAN ANALISA

4.1 Analisis Program 1

```
1 lulus = input("Apakah kamu lulus? [ya/tidak]: ")
2 if lulus == "tidak":
3     print("Kamu harus ikut ujian")
```

Gambar 4.1 Program 1

1. `lulus = input("Apakah kamu lulus? [ya/tidak]: ")`:
 - Input pengguna: Program meminta pengguna untuk memasukkan jawaban dengan menampilkan pesan "Apakah kamu lulus?[ya/tidak]:".
 - Variabel 'lulus': Jawaban pengguna disimpan dalam variabel 'lulus'.
2. `if lulus == "tidak":`
 - Percabangan: Ini adalah struktur percabangan menggunakan if. Program akan mengevaluasi apakah nilai variabel lulus sama dengan string "tidak".
 - Kondisi: Jika jawaban pengguna adalah "tidak", maka blok di bawahnya akan dieksekusi.
3. `print("Kamu harus ikut ujian")`:
 - Cetak Pesan: Jika jawaban pengguna adalah "tidak", program akan mencetak pesan "Kamu harus ikut ujian".

Analisis Umum:

- Program ini memberikan informasi kepada pengguna bahwa jika mereka tidak lulus (jawaban "tidak"), mereka harus mengikuti ujian.
- Input dari pengguna dan kondisi if digunakan untuk mengambil keputusan tentang tindakan selanjutnya yang akan diambil oleh program.

Contoh Eksekusi:

```
Apakah kamu lulus? [ya/tidak]: ya
PS C:\Python> 
```

Gambar 4.2 Output (ya) program 1

Output: Tidak ada output, karena tidak ada blok else yang mengatasi jawaban "ya". Program akan berakhir tanpa melakukan tindakan lebih lanjut.

```
Apakah kamu lulus? [ya/tidak]: tidak
Kamu harus ikut ujian
```

Gambar 4.3 Output (tidak) Program 1

Output: "Kamu harus ikut ujian". Program ini memanfaatkan penggunaan percabangan untuk memberikan respons yang sesuai tergantung pada input yang diberikan oleh pengguna.

4.2 Analisis Program 2

```
1 total_belanja = input("Total belanja: Rp ")
2 # jumlah yang harus dibayar adalah berapa total belanjanya
3 # tapi kalau dapat diskon akan berkurang
4 bayar = int(total_belanja)
5
6 # jika dia belanja di atas 100rb maka berikan bonus dan diskon
7 if int(total_belanja) > 100000:
8     print("Selamat Karena Belanja > Rp. 100.000 anda mendapat Voucher Makan ")
9     print("dan diskon 5%")
10
11 # hitung diskonnya
12 diskon = int(total_belanja) * 5/100 #5%
13 bayar = int(total_belanja) - diskon
14
15 # cetak struk
16 print("Total yang harus dibayar: Rp %s" %bayar)
17 print("Terima kasih sudah berbelanja di Toko Kami")
18 print("Kami Tunggu kunjungan anda selanjutnya.")
```

Gambar 4.4 Program 2

1. `total_belanja = input("Total belanja: Rp ")`:
 - `total_belanja = input("Total belanja: Rp ")` Input Pengguna: Program meminta pengguna untuk memasukkan total belanja dengan menampilkan pesan "Total belanja: Rp ".
 - Variabel `total_belanja`: Nilai yang dimasukkan oleh pengguna disimpan dalam variabel 'total_belanja'.
2. `'bayar = int(total_belanja)'`:
 - Konversi ke Integer: Nilai `total_belanja` yang awalnya string dikonversi menjadi integer dan disimpan dalam variabel `bayar`.
3. `if int(total_belanja) > 100000`:
 - Percabangan: Program menggunakan percabangan `if` untuk mengevaluasi apakah nilai `total_belanja` lebih dari 100.000.
 - Kondisi: Jika total belanja melebihi 100.000, program akan menjalankan blok di bawahnya.
4. `print("Selamat Karena Belanja > Rp. 100.000 anda mendapat Voucher Makan ")`
 - Cetak Pesan: Jika total belanja lebih dari 100.000, program mencetak pesan memberi tahu pengguna bahwa mereka mendapatkan Voucher Makan.
5. `print("dan diskon 5%")`:
 - Cetak Pesan: Program mencetak pesan memberi tahu pengguna bahwa mereka mendapatkan diskon 5%.
6. `diskon = int(total_belanja) * 5/100`:
 - Perhitungan Diskon: Program menghitung diskon sebesar 5% dari total belanja dan menyimpannya dalam variabel `diskon`.
7. `bayar = int(total_belanja) - diskon`:
 - Perhitungan Pembayaran: Program mengurangi diskon dari total belanja untuk mendapatkan jumlah yang harus dibayar dan menyimpannya dalam variabel `bayar`.
8. `print("Total yang harus dibayar: Rp %s" %bayar)`:
 - Cetak Total Pembayaran: Program mencetak total pembayaran yang harus dibayarkan oleh pengguna setelah diskon.

9. `print("Terima kasih sudah berbelanja di Toko Kami"):`
 - Cetak Pesan: Program mencetak pesan terima kasih kepada pengguna atas pembeliannya.
10. `print("Kami Tunggu kunjungan anda selanjutnya."):`
 - Cetak Pesan: Program mencetak pesan mengundang pengguna untuk berkunjung kembali.

Analisis Umum:

- Program ini memberikan informasi kepada pengguna tentang total pembayaran setelah mendapatkan diskon berdasarkan total belanja.
- Pengguna mendapatkan voucher makan dan diskon 5% jika total belanja melebihi 100.000.

Contoh Eksekusi:

```
Total belanja: Rp 120000
Selamat Karena Belanja > Rp. 100.000 anda mendapat Voucher Makan
dan diskon 5%
Total yang harus dibayar: Rp 114000.0
Terima kasih sudah berbelanja di Toko Kami
Kami Tunggu kunjungan anda selanjutnya.
```

Gambar 4.5 Output Program 2

Program ini menciptakan pengalaman berbelanja yang lebih menyenangkan dengan memberikan insentif kepada pelanggan untuk pembelian di atas batas tertentu.

4.3 Analisis Program 3

```
1 umur = input("Berapa umur kamu: ")
2 if int(umur) >= 18:
3     print("Kamu boleh membuat SIM")
4 else:
5     print("Kamu belum boleh membuat SIM")
```

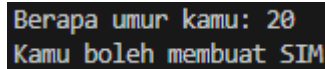
Gambar 4.6 Program 3

1. umur = input("Berapa umur kamu: "):
 - Input Pengguna: Program meminta pengguna untuk memasukkan umur dengan menampilkan pesan "Berapa umur kamu: ".
 - Variabel umur: Nilai yang dimasukkan oleh pengguna disimpan dalam variabel umur.
2. if int(umur) >= 18:
 - Percabangan: Program menggunakan percabangan if untuk mengevaluasi apakah nilai umur setelah dikonversi ke integer adalah lebih besar atau sama dengan 18.
 - Kondisi: Jika umur pengguna lebih besar atau sama dengan 18, program akan menjalankan blok di bawahnya.
3. print("Kamu boleh membuat SIM"):
 - Cetak Pesan: Jika umur pengguna lebih besar atau sama dengan 18, program mencetak pesan memberi tahu pengguna bahwa mereka boleh membuat SIM.
4. Else:
 - Else: Jika kondisi pada langkah 2 tidak terpenuhi (umur kurang dari 18), program akan menjalankan blok di bawahnya.
5. print("Kamu belum boleh membuat SIM"):
 - Cetak Pesan: Jika umur pengguna kurang dari 18, program mencetak pesan memberi tahu pengguna bahwa mereka belum boleh membuat SIM.

Analisis Umum:

- Cetak Pesan: Jika umur pengguna kurang dari 18, program mencetak pesan memberi tahu pengguna bahwa mereka belum boleh membuat SIM. Program ini memberikan informasi kepada pengguna tentang apakah mereka boleh membuat SIM atau belum berdasarkan umur mereka.
- Pengguna mendapatkan jawaban yang spesifik sesuai dengan kondisi umur mereka.

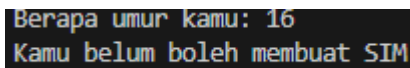
Contoh Eksekusi:



```
Berapa umur kamu: 20
Kamu boleh membuat SIM
```

Gambar 4.7 Output 1 Program 3

Program memberikan jawaban positif karena umur yang dimasukkan (20) memenuhi syarat (lebih dari atau sama dengan 18).



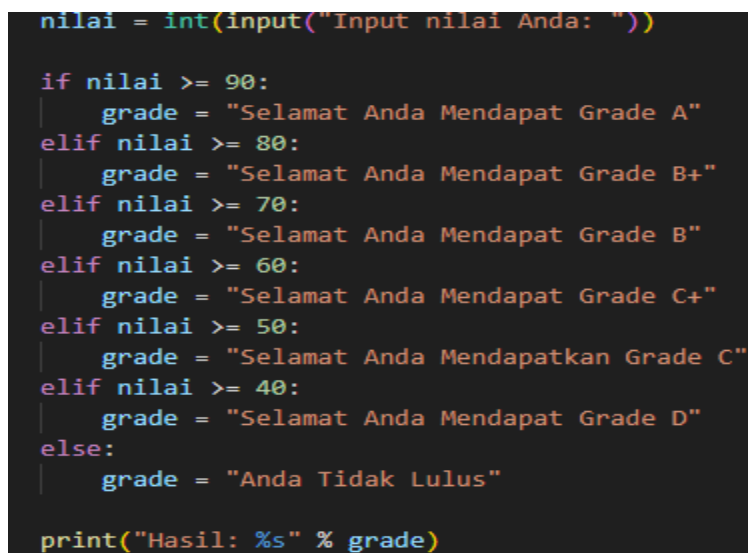
```
Berapa umur kamu: 16
Kamu belum boleh membuat SIM
```

Gambar 4.8 Output 2 program 3

Program memberikan jawaban negatif karena umur yang dimasukkan (16) tidak memenuhi syarat (kurang dari 18).

Program ini memberikan solusi yang jelas berdasarkan kondisi umur pengguna dan memberikan informasi yang relevan terkait kemampuan membuat SIM.

4.4 Analisis Program 4



```
nilai = int(input("Input nilai Anda: "))

if nilai >= 90:
    grade = "Selamat Anda Mendapat Grade A"
elif nilai >= 80:
    grade = "Selamat Anda Mendapat Grade B+"
elif nilai >= 70:
    grade = "Selamat Anda Mendapat Grade B"
elif nilai >= 60:
    grade = "Selamat Anda Mendapat Grade C+"
elif nilai >= 50:
    grade = "Selamat Anda Mendapatkan Grade C"
elif nilai >= 40:
    grade = "Selamat Anda Mendapat Grade D"
else:
    grade = "Anda Tidak Lulus"

print("Hasil: %s" % grade)
```

Gambar 4.9 Program 4

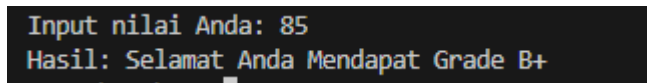
1. `nilai = int(input("Input nilai Anda: "))`:
 - Input Pengguna: Program meminta pengguna untuk memasukkan nilai dengan menampilkan pesan "Input nilai Anda: ".
 - Variabel nilai: Nilai yang dimasukkan oleh pengguna dikonversi menjadi tipe data integer dan disimpan dalam variabel nilai.
2. `if nilai >= 90`:
 - Percabangan: Program menggunakan percabangan `if` untuk mengevaluasi apakah nilai lebih besar atau sama dengan 90.
 - Kondisi: Jika nilai pengguna lebih besar atau sama dengan 90, program akan menjalankan blok di bawahnya.
3. `grade = "Selamat Anda Mendapat Grade A"`:
 - Pemberian Grade: Jika kondisi pada langkah 2 terpenuhi, variabel `grade` diberi nilai "Selamat Anda Mendapat Grade A".
4. `elif nilai >= 80`:
 - Percabangan Alternatif: Jika kondisi pada langkah 2 tidak terpenuhi, program akan mengevaluasi kondisi pada langkah ini.
 - Kondisi: Jika nilai pengguna lebih besar atau sama dengan 80, program akan menjalankan blok di bawahnya.
5. `grade = "Selamat Anda Mendapat Grade B+"`:
 - Pemberian Grade: Jika kondisi pada langkah 4 terpenuhi, variabel `grade` diberi nilai "Selamat Anda Mendapat Grade B+".
6. `elif nilai >= 70`:
 - Percabangan Alternatif: Program akan terus mengevaluasi kondisi secara berurutan hingga salah satu kondisi terpenuhi.
 - Kondisi: Jika nilai pengguna lebih besar atau sama dengan 70, program akan menjalankan blok di bawahnya.
7. `grade = "Selamat Anda Mendapat Grade B"`:
 - Pemberian Grade: Jika kondisi pada langkah 6 terpenuhi, variabel `grade` diberi nilai "Selamat Anda Mendapat Grade B". Proses ini berlanjut untuk setiap blok `elif` yang sesuai.
8. `else`:

- Else: Jika tidak ada kondisi sebelumnya yang terpenuhi, program akan menjalankan blok di bawahnya.
9. `grade = "Anda Tidak Lulus":`
- Pemberian Grade: Jika tidak ada kondisi sebelumnya yang terpenuhi, variabel `grade` diberi nilai `"Anda Tidak Lulus"`.
10. `print("Hasil: %s" % grade):`
- Cetak Grade: Program mencetak `grade` yang telah ditentukan ke layar.

Analisis Umum:

- Program ini memberikan grade berdasarkan rentang nilai tertentu.
- Setiap grade sesuai dengan rentang nilai yang berbeda, dimulai dari A dan turun ke D, dan `"Anda Tidak Lulus"` untuk nilai di bawah 40.

Contoh Eksekusi:



```
Input nilai Anda: 85
Hasil: Selamat Anda Mendapat Grade B+
```

Gambar 4.10 Output Program 4

Program memberikan grade B+ karena nilai yang dimasukkan (85) memenuhi kondisi pada blok `elif nilai >= 80`.

4.5 Analisis Tugas Program

```
print("""Pilih Operasi :
1. Menghitung Volume Kubus
2. Menghitung Luas Lingkaran
3. Menghitung Volume Silinder""")

pilihan = input('Masukkan nomor operasi yang dipilih (1/2/3): ')

if pilihan == '1':
    sisi = float(input("Masukkan panjang sisi kubus: "))
    volume_kubus = sisi ** 3
    print("Volume Kubus adalah:", volume_kubus)
elif pilihan == '2':
    jari_jari = float(input("Masukkan panjang jari-jari lingkaran: "))
    luas_lingkaran = 3.14 * jari_jari ** 2
    print("Luas Lingkaran adalah:", luas_lingkaran)
elif pilihan == '3':
    jari_jari = float(input("Masukkan panjang jari-jari silinder: "))
    tinggi = float(input("Masukkan tinggi silinder: "))
    volume_silinder = 3.14 * jari_jari ** 2 * tinggi
    print("Volume Silinder adalah:", volume_silinder)
else:
    print("Maaf, inputan tidak valid")
```

Gambar 4.11 Tugas Program If-elif-else

Analisis :

- Penggunaan **print** untuk menampilkan pilihan operasi.
- Penggunaan **input** untuk mendapatkan input dari pengguna.
- Penggunaan **if-elif-else** untuk struktur percabangan berdasarkan pilihan pengguna.
- Penggunaan **float** untuk mengonversi input ke tipe data float.
- Penggunaan operator perpangkatan ****** untuk menghitung volume kubus.
- Penggunaan operator matematika dan konstanta 3.14 untuk menghitung luas lingkaran dan volume silinder.

```

1  print("""
2  =====PILH SALAH SATU=====
3  - PILIHAN 1 MENGHITUNG VOLUME KUBUS -
4  - PILIHAN 2 MENGHITUNG LUAS LINGKARAN -
5  - PILIHAN 3 MENGHITUNG VOLUME SILINDER -
6  =====TEKAN 'X' UNTUK KELUAR DARI PROGRAM=====A
7  =====PILH SALAH SATU=====
8  """)
9  pi = 3.14
10 ulangProgram = True
11
12 while ulangProgram:
13     print("Pilih operasi yang ingin Anda lakukan:")
14     opsiPilihan = input("Masukkan opsiPilihan (1/2/3/x): ")
15
16     if opsiPilihan == '1':
17         sisi = int(input("Masukkan panjang sisi kubus: "))
18         volume = sisi * sisi * sisi
19         print(f"Volume kubus: {volume}")
20     elif opsiPilihan == '2':
21         jariJari = int(input("Masukkan jari-jari lingkaran: "))
22         luas = pi * (jariJari * jariJari)
23         print(f"Luas lingkaran: {luas}")
24     elif opsiPilihan == '3':
25         jariJari = int(input("Masukkan panjang jari-jari lingkaran: "))

```

Gambar 4.12 Tugas Program Alternatif Do-While

Analisis:

- Pada baris pertama program akan menjalankan method **Print()** yang berfungsi untuk mencetak . Yang berisikan Interface buat user apa aja yang dapat dilakukan pada program ini.
- Kemudian **Variable** pi itu berisikan nilai PHI yaitu 3,14.
- Penggunaan While yang berisikan variable ulangProgram yang bernilai **True** dimaksudkan agar program berjalan terus menerus selama program bernilai **True** jika jika program **False** maka program akan berhenti.
- Struktur pengulangan menggunakan **break** untuk keluar dari perulangan.

```

22     luas = pi * (jariJari * jariJari)
23     print(f"Luas lingkaran: {luas}")
24     elif opsiPilihan == '3':
25         jariJari = int(input("Masukkan panjang jari-jari lingkaran: "))
26         tinggi = int(input("Masukkan tinggi silinder: "))
27         volume = pi * (jariJari * jariJari) * tinggi
28         print(f"Volume silinder: {volume}")
29     elif opsiPilihan == 'x' or opsiPilihan == 'X':
30         print("Anda Keluar Dari Program.")
31         break
32     else:
33         print("Pilihan tidak valid. Silakan pilih lagi.")
34
35     ulangProgram = input('Ingin Menghitung lagi? (ya/tidak)')
36     if ulangProgram == 'ya':
37         True
38     elif ulangProgram == 'tidak':
39         False
40         print('Terimakasih :)')
41         break
42     else:
43         print('')
44

```

Gambar 4.13 Tugas Program Altrnatif Switch-Case

Analisis:

- Penggunaan fungsi (**def**) untuk mengorganisir perhitungan berdasarkan operasi.
- Penggunaan dictionary (operasi) untuk menghubungkan nomor operasi dengan fungsi yang sesuai.
- Penggunaan **in** untuk memeriksa apakah nomor operasi valid.
- Penggunaan **print** untuk menampilkan hasil perhitungan.

Perbedaan dari ketiga program:

1. Program If-elif-else:
 - Meminta pengguna untuk memilih operasi.
 - Bergantung pada pilihan pengguna, program akan menghitung dan mencetak volume kubus, luas lingkaran, atau volume silinder.
2. Program menggunakn perulangan while True:
 - Meminta pengguna untuk memilih operasi.

- Bergantung pada pilihan pengguna, program akan menghitung dan mencetak volume kubus, luas lingkaran, atau volume silinder.
- Terdapat perulangan tak terbatas (`while True`), sehingga program akan terus meminta input hingga pengguna memilih operasi yang valid.
- Program menggunakan fungsi sebagai pengganti struktur `do-while` yang tidak secara eksplisit didukung dalam Python.

3. Program fungsi sebagai pengganti `switch-case`:

- Meminta pengguna untuk memilih operasi.
- Bergantung pada pilihan pengguna, program akan menggunakan fungsi yang sesuai untuk menghitung dan mencetak hasil perhitungan.
- Program menggunakan fungsi sebagai pengganti struktur `switch-case` yang tidak secara eksplisit didukung dalam Python.

Semua program memiliki logika yang berbeda dalam mengatur alur eksekusi dan menghasilkan output yang sesuai dengan tujuan dan struktur masing-masing program.

BAB V

PENUTUP

5.1 Kesimpulan

Dalam penutup laporan ini, dapat disimpulkan bahwa percabangan memainkan peran krusial dalam pengembangan perangkat lunak menggunakan bahasa pemrograman Python. Konsep dasar teori percabangan, seperti if, elif, dan else, membuka peluang untuk membuat program yang responsif terhadap berbagai kondisi. Melalui praktikum yang disajikan, kami telah memperlihatkan contoh penggunaan percabangan dalam skenario berbeda, mulai dari pengambilan keputusan sederhana hingga penggunaan dalam struktur perulangan.

Penting untuk dicatat bahwa dalam Python, tidak ada struktur langsung yang setara dengan do-while maupun switch case seperti dalam beberapa bahasa pemrograman lain. Meskipun demikian, kami telah menunjukkan bagaimana dapat menggunakan pernyataan while True dan struktur kontrol alur untuk mencapai hasil yang serupa.

Dengan pemahaman yang mendalam tentang konsep-konsep percabangan dalam Python, para pengembang dapat meningkatkan fleksibilitas dan responsivitas program mereka. Kesimpulan ini mencerminkan pentingnya memahami percabangan sebagai keterampilan esensial dalam mengembangkan solusi perangkat lunak yang efektif dan dapat diandalkan.

Laporan Dokumentasi



Gambar 4.14 Dokumentasi Praktikum

DAFTAR PUSTAKA

Python.org

Situs web resmi Python: <https://www.python.org>

Python Software Foundation. (2021). Python 3.9.7 Documentation.

*Dokumentasi resmi Python yang berisi petunjuk instalasi:
<https://docs.python.org/3/>*

Lutz, M. (2013). Learning Python, 5th Edition. O'Reilly Media.

*Buku yang mencakup instalasi Python dan konsep dasar:
<https://www.oreilly.com/library/view/learning-python-5th/9781449355722/>*

McKinney, W. (2017). Python for Data Analysis, 2nd Edition. O'Reilly Media.

*Buku yang fokus pada instalasi Python untuk analisis data:
<https://www.oreilly.com/library/view/python-for-data/9781491957653/>*

GeeksforGeeks. (2021). Python Installation on Windows, macOS, and Linux