



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231051
Nama Lengkap	Amida A Ronsumbre
Minggu ke / Materi	03/ Struktur Kontrol Percabangan

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

BAGIAN 1: MATERI MINGGU INI (40%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

MATERI 1

Penjelasan materi 1

3. Struktur Kontrol Percabangan

3.1 Tujuan Praktikum

1. Menyusun Boolean expression dengan benar dan menggabungkan beberapa expression menggunakan operator logical.
2. Menggunakan percabangan if sesuai dengan permasalahan yang dihadapi.
3. Menangani kesalahan masukan pengguna dengan exception Sederhana.

```
1  try:
2      angka = int(input("Masukkan sebuah angka: "))
3      print("Angka yang dimasukkan:", angka)
4  except ValueError:
5      print("Masukan tidak valid. Mohon masukkan angka.")
```

```
PS C:\Users\asus\Documents\Pratikum ALPRO> python -u "c:\Users\asus\Documents\Pratikum ALPRO\Latihan.py"
Masukkan sebuah angka: 5
Angka yang dimasukkan: 5
PS C:\Users\asus\Documents\Pratikum ALPRO> █
```

ni adalah alasan paling langsung. Jika nilai x tidak lebih besar dari 5, maka pernyataan dalam blok `if` tidak akan dieksekusi. Secara keseluruhan, keempat kemampuan di atas merupakan fondasi penting dalam pengembangan perangkat lunak yang baik, dan dapat membantu mahasiswa dalam membangun pemahaman yang kuat dalam logika pemrograman, pengambilan keputusan, dan penanganan kesalahan yang mungkin terjadi. Kesalahan masukan pengguna adalah hal yang umum terjadi dalam pengembangan perangkat lunak.

3.2 Alat dan Bahan

Praktikum ini membutuhkan perangkat komputer yang memiliki spesifikasi minimum sebagai berikut, Perangkat lunak yang diperlukan untuk mendukung praktikum ini adalah sebagai berikut yang terinstall menggunakan:

- Anaconda atau installer python lainnya.
- Command prompt(jika menggunakan Windows.)
- **Python 3.7 atau 3.8:** Python adalah bahasa pemrograman utama yang digunakan dalam praktikum. Versi 3.7 atau 3.8 diperlukan karena beberapa library atau package mungkin tidak kompatibel dengan versi yang lebih lama atau lebih baru.
- Dengan memenuhi semua persyaratan perangkat keras dan lunak di atas, peserta praktikum akan dapat mengikuti kegiatan pembelajaran dengan lancar dan mendapatkan pengalaman praktis yang efektif dalam pengembangan perangkat lunak dengan Python.

Materi

3.3.1 Boolean Expression dan Logical Operator

Perhatikan kasus berikut: Voucher diskon 30% pat dipakai jika minimum pembelian anda adalah Rp.

Minimum pembelian tersebut dapat dinyatakan dalam Python sebagai berikut: pembelian \geq 100000 Bentuk tersebut dinamakan sebagai boolean expression, karena hasilnya hanya ada dua kemungkinan, yaitu True atau False, tergantung dari variabel pembelian.

Jika anda masukkan perintah tersebut ke dalam Python mode interaktif, hasilnya dapat dilihat pada Gambar 3.

Boolean expression dapat disusun menggunakan operator-operator perbandingan seperti pada Tabel 3.

$x > y$ Apakah x lebih besar dari y?

$x \geq y$ Apakah x lebih besar atau sama dengan y?

$x < y$ ~~xss=removed~~ 1000000 Pengendara dengan kecepatan lebih dari 90 km/jam akan mendapatkan tilang kecepatan > 90 Wahana Rollercoaster hanya bisa dinaiki oleh mereka yang tinggi badannya lebih dari 110 cm tinggi > 110 Nilai ujian Hanna adalah 75 sedangkan Robby mendapatkan nilai 75.

Apakah gabungan sepatu Junaedi dan Ricky lebih banyak dari sepatu milik Arnold?

$junaedi + ricky > arnold$ hanya dapat dinaiki oleh penumpang dengan usia minimal 10 tahun dan tinggi badan minimal 110 cm.

Kedua persyaratan tersebut dapat dinyatakan dalam bentuk sebagai berikut: usia ≥ 10 and tinggi ≥ 110 Diskon diberikan kepada member atau jumlah pembelian lebih dari Rp.75.

```

PS C:\Users\asus\Documents\prtk Alpro> python -u "c:\Users\asus\Documents\pr
tk Alpro\py2"
Masukkan usia Anda: 20
Masukkan tinggi Anda (dalam cm): 165 cm
Traceback (most recent call last):
  File "c:\Users\asus\Documents\prtk Alpro\py2", line 3, in <module>
    tinggi = int(input("Masukkan tinggi Anda (dalam cm): "))
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: '165 cm'
PS C:\Users\asus\Documents\prtk Alpro> 

```

3.3.2 Bentuk-bentuk Percabangan

Percabangan pada Python secara umum ada tiga bentuk, yaitu: conditional, alternative dan chained conditional.

Bentuk conditional secara umum dinyatakan dalam kode program sebagai berikut: if : .

Sebagai contoh, jika nilai akhir > 70 maka akan mendapatkan sertifikat kelulusan.

Kode programnya sebagai berikut: if nilai_akhir > 70: print("Anda lulus dan mendapatkan sertifikat kelulusan!") Bentuk alternative conditional adalah bentuk percabangan yang memiliki dua alternatif langkah yang harus dijalankan berdasarkan kondisi tertentu.

Sebagai contoh, jika nilai akhir > 60, tampilkan tulisan Lulus.

Pada contoh ini, ada dua kemungkinan tulisan yang muncul, yaitu Lulus atau Tidak Lulus.

Implementasinya sebagai berikut: if nilai_akhir > 60: print("Lulus") else: print("Tidak Lulus")

Bentuk chained conditional digunakan jika kemungkinan langkah yang harus dijalankan berikutnya lebih dari dua.

Bentuknya secara umum adalah sebagai berikut: if : .

Bentuknya secara umum adalah sebagai berikut: if : .

Sebagai contoh, misalnya sebuah toko pakaian memberi diskon yang besarnya ditentukan oleh nilai pembelian anda.

Pembelian lebih dari Rp.1.000.000

000 tidak mendapatkan diskon.

Kemungkinan diskon yang diberikan ada 4, yaitu: 30%, 20%, 15% dan 0% (tidak diskon).

Perhatikan implementasi berikut ini: if pembelian > 1000000: diskon = 0.

```

py2 > ...
1  pembelian = float(input("Masukkan total pembelian Anda (dalam Rp): ")) # Input
2
3  if pembelian > 1000000:
4      diskon = 0.3 # diskon 30%
5  elif pembelian > 500000 and pembelian <= 1000000:
6      diskon = 0.2 # diskon 20%
7  elif pembelian >= 100000 and pembelian <= 500000:
8      diskon = 0.15 # diskon 15%
9  else:
10     diskon = 0 # tidak ada diskon
11
12  print("Diskon yang diberikan:", diskon * 100, "%") # Output besar diskon d

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\asus\Documents\prtk Alpro> python -u "c:\Users\asus\Documents\pr
tk Alpro\py2"
Masukkan total pembelian Anda (dalam Rp): 1000000
Diskon yang diberikan: 20.0 %
PS C:\Users\asus\Documents\prtk Alpro>

```

Dengan demikian, perubahan-perubahan tersebut bertujuan untuk meningkatkan kejelasan, keakuratan, dan fungsionalitas kode sehingga program dapat bekerja dengan baik dalam menghitung diskon berdasarkan jumlah pembelian yang diberikan.

Fungsi dari kode yang telah disesuaikan adalah untuk menghitung jumlah diskon berdasarkan jumlah pembelian yang dimasukkan pengguna, kemudian menghitung total bayar setelah diskon diterapkan, dan akhirnya mencetak total bayar tersebut.

Dengan menggunakan ini, pengguna dapat memasukkan jumlah pembelian, dan program akan memberikan informasi tentang jumlah diskon yang diberikan berdasarkan rentang jumlah pembelian, serta total pembayaran setelah diskon diterapkan.

3.3.3 Penanganan Kesalahan Input Menggunakan Exception Handling

Dalam menangani input dari pengguna, kita juga perlu memperhatikan potensi kesalahan yang mungkin terjadi sehingga program tidak bisa berjalan dengan semestinya.

Untuk lebih jelasnya, perhatikan program yang meminta input usia pengguna, kemudian program akan menampilkan apakah pengguna termasuk lansia, dewasa, remaja, kanak-kanak atau balita.

Kategori usia tersebut mengikuti aturan sebagai berikut: • Balita: 0-5 tahun.

Program tersebut adalah sebagai berikut: 1 usia = int(input("Masukkan usia anda: ")) 2 if usia = 6 and usia = 12 and usia = 26 and usia 45: 11 print("Lansia") Jika program tersebut dijalankan beberapa kali dengan input yang berbeda-beda, hasilnya telah sesuai dengan yang diharapkan.

Hasil dari program tersebut dapat dilihat pada Gambar 3.

2. Bagian dari program tersebut yang meminta pengguna memasukkan usianya ada di baris 1, yaitu: `usia = int(input("Masukkan usia anda: "))` Gambar 3.

Program tersebut akan berjalan dengan baik selama pengguna tidak memasukkan input yang salah atau tidak sesuai yang diharapkan.

3 yang menunjukkan pengguna memasukkan input yang tidak sesuai.

3: Jika input tidak sesuai yang diharapkan, program akan berhenti.

Penggunaannya dalam kasus kategori usia dapat dilihat pada source code program berikut ini: 1 `inputuser = input("Masukkan usia anda: ")` 2 `try:` 3 `usia = int(inputuser)` 4 `if usia <= 5:` 5 `print("Balita")` 6 `elif usia >= 6 and usia <= 11:` 7 `print("Kanak-kanak")` 8 `elif usia >=12 and usia <= 25:` 9 `print("Remaja")` 10 `elif usia >= 26 and usia <= 45:` 11 `print("Dewasa")` 12 `elif usia > 45:` 13 `print("Lansia")` 14 `except:` 15 `print("Anda salah memasukkan input usia")`

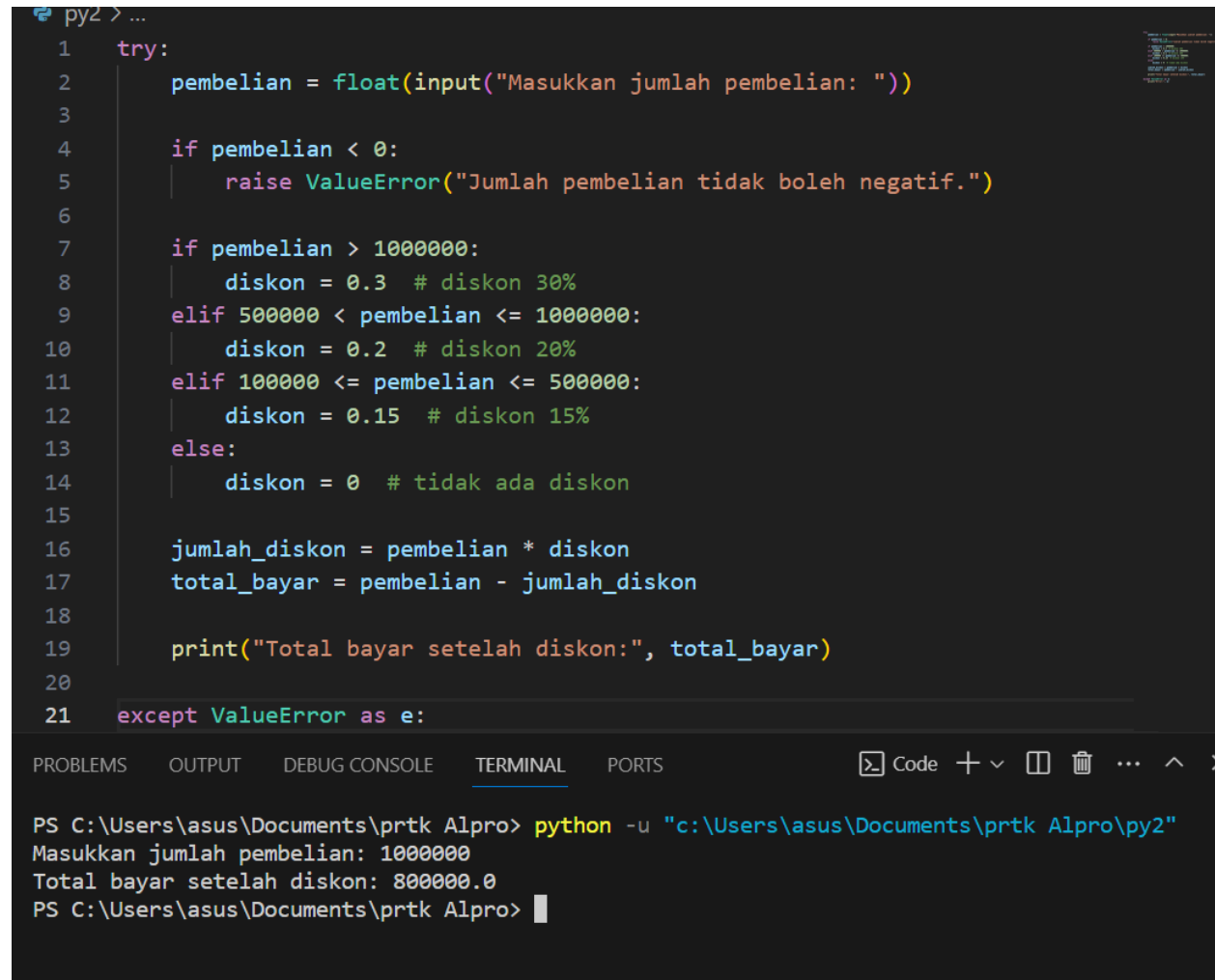
Misalnya contohnya:

```
1  usia = int(input("Masukkan usia anda: "))
2  if usia <= 5:
3      print("Balita")
4  elif usia >= 6 and usia <= 11:
5      print("Kanak-kanak")
6  elif usia >=12 and usia <= 25:
7      print("Remaja")
8  elif usia >= 26 and usia <= 45:
9      print("Dewasa")
10 elif usia > 45:
11     print("Lansia")
12
13
```

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

SOAL 1



```
py2 > ...
1  try:
2      pembelian = float(input("Masukkan jumlah pembelian: "))
3
4      if pembelian < 0:
5          raise ValueError("Jumlah pembelian tidak boleh negatif.")
6
7      if pembelian > 1000000:
8          diskon = 0.3 # diskon 30%
9      elif 500000 < pembelian <= 1000000:
10         diskon = 0.2 # diskon 20%
11     elif 100000 <= pembelian <= 500000:
12         diskon = 0.15 # diskon 15%
13     else:
14         diskon = 0 # tidak ada diskon
15
16     jumlah_diskon = pembelian * diskon
17     total_bayar = pembelian - jumlah_diskon
18
19     print("Total bayar setelah diskon:", total_bayar)
20
21 except ValueError as e:
```





PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [] [X] ... ^ >

```
PS C:\Users\asus\Documents\prtk Alpro> python -u "c:\Users\asus\Documents\prtk Alpro\py2"
Masukkan jumlah pembelian: 1000000
Total bayar setelah diskon: 800000.0
PS C:\Users\asus\Documents\prtk Alpro> |
```

Dengan penanganan kesalahan input pengguna seperti ini, program akan lebih tangguh dan akan memberikan pesan yang lebih informatif kepada pengguna jika terjadi kesalahan dalam input yang diberikan. Sehingga dapat memperoleh ini.

SOAL 2

```
pyz > ...
1  # Menerima input dari pengguna
2  angka = float(input("Masukkan sebuah angka: "))
3
4  hasil = "Positif" if angka > 0 else "Negatif" if angka < 0 else "Nol"
5
6  # Mencetak hasilnya
7  print("Angka tersebut adalah:", hasil)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  Code    ... ^

```
PS C:\Users\asus\Documents\prtk Alpro> python -u "c:\Users\asus\Documents\prtk Alpro\py2"
Masukkan sebuah angka: 2
Angka tersebut adalah: Positif
PS C:\Users\asus\Documents\prtk Alpro> 3
3
PS C:\Users\asus\Documents\prtk Alpro> python -u "c:\Users\asus\Documents\prtk Alpro\py2"
Masukkan sebuah angka: 7
Angka tersebut adalah: Positif
PS C:\Users\asus\Documents\prtk Alpro> 
```

Dalam kode ini, ketika kita menentukan apakah angka itu negatif atau tidak, kita menggunakan "else" tambahan untuk mengecek apakah angka itu nol. Ini memastikan bahwa jika angka tersebut bukan positif, dan juga bukan negatif, maka itu harus nol. Dengan menambahkan "Nol Else", kita menangani kasus di mana angka tersebut adalah nol dengan benar.

SOAL 3

```
def jumlah_hari_bulan(tahun, bulan):
    # Daftar jumlah hari dalam setiap bulan
    jumlah_hari = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

    # Penanganan tahun kabisat (Februari memiliki 29 hari)
    if bulan == 2 and (tahun % 400 == 0 or (tahun % 4 == 0 and tahun % 100 != 0)):
        return 29
    else:
        return jumlah_hari[bulan - 1] # Mengembalikan jumlah hari berdasarkan indeks bulan

try:
    bulan = int(input("Masukkan bulan (1-12): "))

    # Memeriksa apakah bulan yang dimasukkan valid
    if bulan < 1 or bulan > 12:
        raise ValueError("Bulan yang dimasukkan tidak valid. Harap masukkan nomor bulan antara 1-12.")

    tahun = 2020 # Tahun yang ditetapkan dalam permasalahan

    # Memanggil fungsi jumlah_hari_bulan untuk mendapatkan jumlah hari
    jumlah_hari = jumlah_hari_bulan(tahun, bulan)
```

Dengan demikian, program memungkinkan pengguna untuk memasukkan nomor bulan, menghitung jumlah hari dalam bulan tersebut, dan menangani kesalahan dengan memberikan pesan yang sesuai jika diperlukan.

```
    # Memanggil fungsi jumlah_hari_bulan untuk mendapatkan jumlah hari
    jumlah_hari = jumlah_hari_bulan(tahun, bulan)

    print("Jumlah hari:", jumlah_hari)

except ValueError as e:
    print("Error:", e)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\asus\Documents\prtk Alpro> python -u "c:\Users\asus\Documents\prtk Alpro\py2"
Masukkan bulan (1-12): 07
Jumlah hari: 31
PS C:\Users\asus\Documents\prtk Alpro> █
```

- Fungsi ini mengembalikan jumlah hari dalam suatu bulan, termasuk penanganan tahun kabisat untuk bulan Februari.
- Pengguna diminta untuk memasukkan nomor bulan (1-12).
- Program memeriksa apakah nomor bulan yang dimasukkan pengguna valid (antara 1 dan 12).
- Jika nomor bulan valid, program menghitung jumlah hari dalam bulan tersebut menggunakan fungsi **Jumlah_hari_bulan**

- Program menangani kesalahan umum dan mencetak pesan kesalahan jika terjadi masalah selama eksekusi.

SOAL 4

```
1  try:
2      # Meminta pengguna untuk memasukkan panjang sisi segitiga
3      sisi_1 = float(input("Masukkan sisi 1: "))
4      sisi_2 = float(input("Masukkan sisi 2: "))
5      sisi_3 = float(input("Masukkan sisi 3: "))
6
7      # Memeriksa apakah semua sisi sama panjang
8      if sisi_1 == sisi_2 == sisi_3:
9          print("3 sisi sama")
10     # Memeriksa apakah ada dua sisi yang sama panjang
11     elif sisi_1 == sisi_2 or sisi_1 == sisi_3 or sisi_2 == sisi_3:
12         print("2 sisi sama")
13     # Jika tidak ada sisi yang sama panjang
14     else:
15         print("Tidak ada yang sama")
16
17 except ValueError:
18     print("Input tidak valid. Pastikan Anda memasukkan bilangan.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\asus\Documents\prtk Alpro> python -u "c:\Users\asus\Documents\prtk Alpro\py2"
Masukkan sisi 1: 4
Masukkan sisi 2: 8
Masukkan sisi 3: 12
Tidak ada yang sama
PS C:\Users\asus\Documents\prtk Alpro> |
```

```
PS C:\Users\asus\Documents\prtk Alpro> python -u "c:\Users\asus\Documents\prtk Alpro\py2"
Masukkan sisi 1: 8
Masukkan sisi 2: 8
Masukkan sisi 3: 8
3 sisi sama
PS C:\Users\asus\Documents\prtk Alpro> |
```

1. Program meminta pengguna untuk memasukkan panjang sisi-sisi segitiga menggunakan **Input ()** dan mengonversi nilai tersebut ke float menggunakan **Float()**.
2. Kemudian, program memeriksa kondisi-kondisi berikut:
 - Jika semua sisi segitiga sama Panjang **sisi_1 == sisi_2 == sisi_3**, program mencetak "3 sisi sama".

- Jika ada dua sisi yang sama Panjang **sisi_1 == sisi_2** atau **sisi_1 == sisi_3** atau **sisi_2 == sisi_3** program mencetak "2 sisi sama".
- Jika tidak ada sisi yang sama panjang, program mencetak "Tidak ada yang sama".

Jika terjadi kesalahan saat mengonversi input pengguna menjadi float (misalnya, pengguna memasukkan huruf atau karakter khusus bukan angka), program akan menangkap kesalahan tersebut dan mencetak pesan kesalahan yang sesuai.

Dengan cara ini, program dapat meminta input pengguna, menentukan jenis segitiga berdasarkan panjang sisi-sisinya, dan menangani kesalahan jika input pengguna tidak valid.