Laporan Praktikum Algoritma & Pemrograman
Semester Genap 2025/2026

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.

NIM	**71251235**
Nama Lengkap	**NATALIA GRECIA PUTRI**
Minggu ke / Materi	**04 / Struktur Kontrol Percabangan**

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2026

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.

Boolean Expression dan Logic Operator

Ekspresi boolean merupakan pernyataan logika yang hanya menghasilkan dua kemungkinan nilai, yaitu benar (True) atau salah (False). Sebagai contoh dalam Python, kondisi seperti pembelian >= 100000' digunakan untuk memverifikasi apakah seseorang memenuhi syarat mendapat potongan harga.

Ekspresi boolean menggunakan operator perbandingan seperti:

- == : sama dengan
- != : tidak sama dengan
- > : lebih besar
- < : lebih kecil
- >= : lebih besar sama dengan
- <= : lebih kecil sama dengan

Selain itu, beberapa ekspresi boolean dapat digabungkan menggunakan operator logika, yaitu:

- and : Semua kondisi harus terpenuhi
- or : minimal satu kondisi harus terpenuhi
- not : membalikkan nilai boolean

Misalnya, persyaratan untuk menaiki wahana roller coaster adalah memiliki tinggi lebih dari 110 cm dan usia minimal 10 tahun, yang dapat diekspresikan dalam Python sebagai:

usia >= 10 and tinggi >= 110

Sementara itu, diskon dapat diberikan jika pelanggan adalah member atau memiliki total pembelian lebih dari Rp 500.000:

member == True or pembelian > 500000

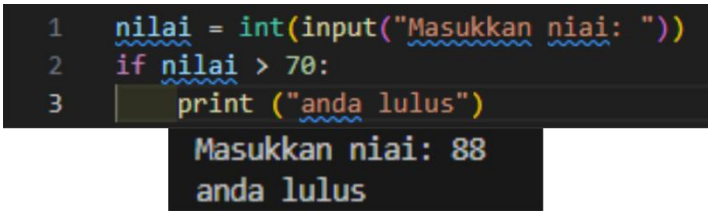
Ekspresi boolean dan operator logika sangat vital dalam pemrograman untuk membuat keputusan dan mengontrol alur program berdasarkan kondisi yang ditentukan.

Bentuk-bentuk Percabangan

Dalam pemrograman Python, struktur percabangan berfungsi untuk mengambil keputusan berdasarkan kondisi yang ada. Terdapat tiga bentuk utama percabangan yaitu, Condition (if), Alternative (if-else), dan Chained Conditional (if-elif-else).

1. Conditional (if)

   Percabangan ini berfungsi untuk mengeksekusi kode ketika suatu kondisi terpenuhi. Jika kondisi bernilai True, maka blok kode di dalamnya akan dijalankan. Contohnya:

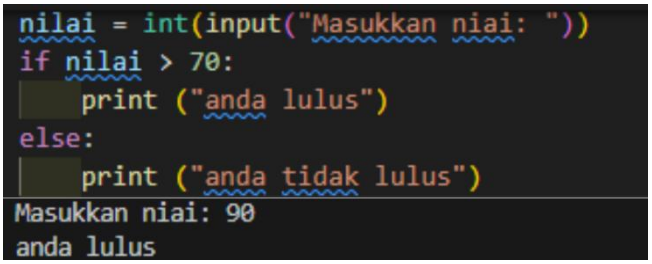


Gambar 4.1: Contoh conditional (If)

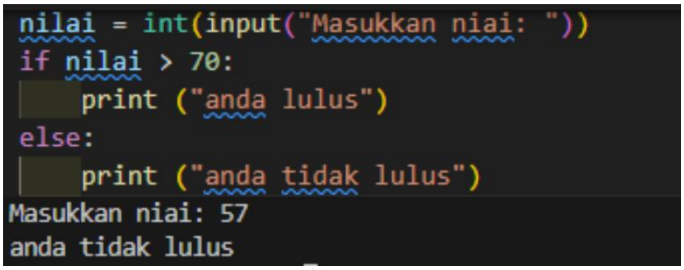
   Jika nilai lebih dari 70, akan ditampilkan pesan ‘anda lulus’

2. Alternative (If-else)

   Bentuk ini digunakan jika terdapat dua kemungkinan hasil. Jika kondisi pertama tidak terpenuhi, maka blok kode alternatif akan dieksekusi. Contohnya:



Gambar 4.2: Contoh Alternative (if-else) lulus



Gambar 4.3: Contoh Alternative (if-else) tidak lulus

Jika nilai lebih dari 70, akan menampilkan 'lulus', jika tidak maka akan menampilkan 'tidak lulus'.

3. Chained Conditional (If-Elif-Else)

   Jika terdapat lebih dari dua kemungkinan hasil, maka digunakan bentuk percabangan berantai (if-elif-else).

   Contohnya dalam sistem diskon:

```
if pembelian > 1000000:
        Diskon = 0.3            #30% diskon
elif pembelian > 500000:
        Diskon = 0.2            #20% diskon
elif pembelian >= 100000:
        diskon = 0.15           #15% diskon
else:
        diskon = 0              #Tidak ada diskon
```

   Dalam contoh ini, persentase diskon ditentukan berdasarkan jumlah pembelian menggunakan serangkaian kondisi yang diperiksa secara berurutan.

4. Ternary Operator (If-Else dalam satu baris)

   Python juga menyediakan cara lebih ringkas untuk menuliskan percabangan sederhana menggunakan **ternary operator**.

   Contohnya:

   Diskon = 0.1 if pembelian > 100000 else 0

   Kode ini setara dengan:

```
if pembelian > 100000:
        Diskon = 0.1
else:
```

Diskon = 0

Dengan pemahaman percabangan ini, kita dapat mengontrol alur eksekusi program secara lebih dinamis berdasarkan kondisi yang terjadi.

Penanganan Kesalahan Input Menggunakan Exception Handing

Dalam pemrograman, kesalahan input dari pengguna dapat menyebabkan program berhenti secra mendadak. Karena itu, penting untuk menangani kemungkinan kesalahan ini agar program tetap berjalan dengan baik.

Sebagai contoh, sebuah program yang meminta pengguna memasukkan usia dan menentukan kategori umur berdasarkan aturan berikut:

- Balita: 0-5 tahun
- Kanak-kanak: 6-11 tahun
- Remaja: 12-25 tahun
- Dewasa: 26-45 tahun
- Lansia: lebih dari 45 tahun

Kode Program Tanpa Exception Handling:

```
usia= int(input("Masukkan usia anda: "))
if usia <= 5:
    print ("Balita")
elif usia >= 6 and usia <= 11:
    print ("Kanak-kanan")
elif usia >= 12 and usia <= 25:
    print ("Remaja")
elif usia >= 26 and usia <- 45:
    print ("Dewasa")
else:
    print ("Lansia")
Masukkan usia anda: 17
Remaja
```

Gambar 4.4: Contoh Program Tanpa Exception Handling

```
usia= int(input("Masukkan usia anda: "))
if usia <= 5:
    print ("Balita")
elif usia >= 6 and usia <= 11:
    print ("Kanak-kanan")
elif usia >= 12 and usia <= 25:
    print ("Remaja")
elif usia >= 26 and usia <- 45:
    print ("Dewasa")
else:
    print ("Lansia")
Masukkan usia anda: tujuh
Traceback (most recent call last):
  File "e:\Praktek AlPro\handling.py", line 1, in <module>
    usia= int(input("Masukkan usia anda: "))
ValueError: invalid literal for int() with base 10: 'tujuh'
```

Gambar 4.5: Contoh Program Tanpa Exception Handling

Kode tersebut berfungsi dengan baik jika pengguna memasukkan angka yang valid. Namun, jika pengguna salah memasukkan input, misalnya teks atau karakter lain, program akan mengalami error dan berhenti.

Menggunakan Exception Handling dengan Try-Except:

```
try:
    usia= int(input("Masukkan usia anda: "))
    if usia <= 5:
        print ("Balita")
    elif usia >= 6 and u
        print ("Kanak-ka  (variable) usia: int
    elif usia >= 12 and usia <= 25:
        print ("Remaja")
    elif usia >= 26 and usia <- 45:
        print ("Dewasa")
    else:
        print ("Lansia")
except:
    print ("Anda salah memasukkan input usia")
Masukkan usia anda: tiga puluh
Anda salah memasukkan input usia
```

Gambar 4.6: Contoh Program Menggunakan Exception Handling (Try-Except)

Untuk menghindari error akibat kesalahan input, kita bisa menggunakan try-except yang akan menangkap kesalahan dan menampilkan pesan yang lebih informatif.

Jika pengguna memasukkan angka, program akan berjalan seperti biasa. Namun, jika input yang dimasukan bukan angka misalnya, huruf atau simbol, program tidak akan error, tetapi akan menampilkan pesan kesalahan yang informatif.

Keuntungan Menggunakan Exception Handling:

- Program tidak crash ketika terjadi kesalahan input.
- Pengguna mendapatkan feedback yang jelas tentang kesalahan yang terjadi.
- Meningkatkan user experience dan keandalan program.
- Memungkinkan program untuk menjalankan program.
- Memungkinkan program untuk melanjutkan eksekusi meskipun terjadi error.

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

https://github.com/grec12/71251235_nataliagrecia.git

Implementasi penanganan kesalahan input pengguna.

A. Program Demam atau Tidak

Gambar 4.7: Program menentukan demam atau tidak

Program ini berfungsi untuk memeriksa apakah seseorang mengalami demam berdasarkan suhu tubuh yang diinputkan.

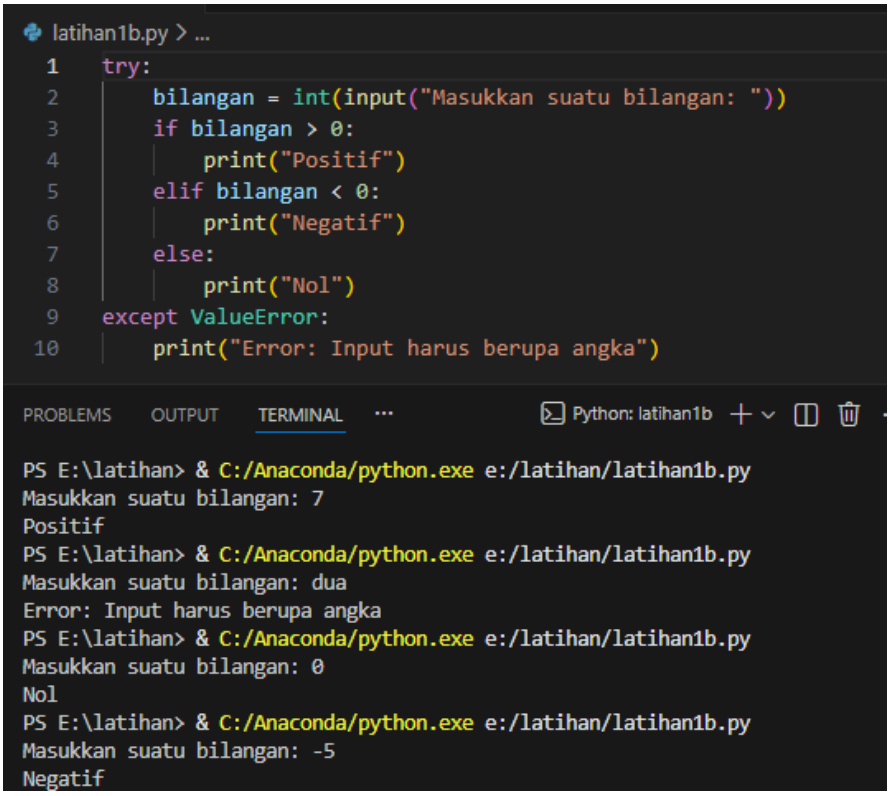
Logika Program:

1. Program meminta input suhu tubuh dari pengguna.
2. Menggunakan try-except untuk menangani kesalahan input.
3. Jika suhu >=38 derajat Celcius, maka dinyatakan demam.
4. Jika suhu < 38 derajat Celcius, maka dinyatakan tidak demam.
5. Jika input bukan angka, maka program akan menampilkan pesan error yang informatif.

Penjelasan code:

- Baris 1: Memulai try untuk menangkap exception.
- Baris 2: Menerima input suhu dari pengguna, menggunakan float() agar dapat menerima desimal.
- Baris 3-4: Kondisi pertama, jika suhu >= 38, maka akan menampilkan 'Anda demam'.
- Baris 5-6: Kondisi alternatif jika suhu < 38, maka akan menampilkan 'Anda tidak demam'.
- Baris 7-8: Block except untuk menangkap ValueError jika input bukan angka.

B. Program Pengecekan Positif atau Negatif



Gambar 4.8: Program pengecekan positif atau negatif

Program ini berfungsi untuk mencari nilai terbesar dari tiga bilangan yang diinput oleh pengguna.

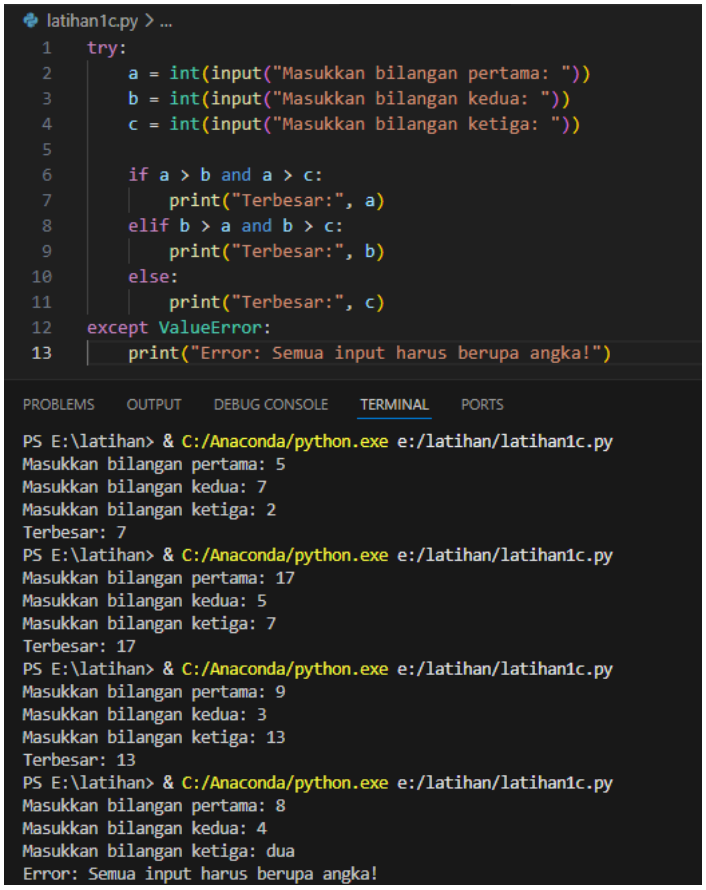
Logika Program:

1. Program meminta input suhu tubuh dari pengguna.
2. Menggunakan try-except untuk menangani kesalahan input.
3. Jika bilangan > 0, maka akan dinyatakan positif.
4. Jika bilangan < 0, maka akan dinyatakan negatif.
5. Jika bilangan = 0, maka akan dinyatakan nol.
6. Jika input tidak valid (bukan angka), maka program akan menampilkan pesan error.

Penjelasan Kode:

- Baris 1: Memulai blok try.
- Baris 2: Input bilangan menggunakan int().
- Bars 3-4: Kondisi pertama untuk bilangan positif (>0).
- Baris 5-6: Kondisi kedua untuk bilangan negatif (<0).
- Baris 7-8: Kondisi else untuk bilangan nol (otomatis == 0).
- Baris 9-10: Exception handling untuk input tidak valid.

C. Program Mencari Nilai Terbesar



Gambar 4.9: Mencari nilai terbesar

Program ini meminta tiga input bilangan pengguna, kemudian menentukan dan menampilkan bilangan yang terbesar dari ketiganya.

Logika Program:

1. Ada tiga kemungkinan bilangan terbesar.
2. Kriteria a terbesar: a > b and a > c.
3. Kriteria b terbesar: b > a and b > c.
4. Kriteria c terbesar: c > a and c > b.
5. Setiap input dibungkus dengan try-except terpisah.

Penjelasan Kode:

- Baris 1: Memulai blok try untuk menangkap exception.
- Baris 2-4: Menerima tiga input bilangan dari pengguna.
- Baris 6-7: Cek apakah a adalah yang terbesar menggunakan logical operator AND.
- Baris 8-9: Cek apakah b adalah yang terbesar.
- Baris 10-11: Jika bukan a atau b, maka c adalah yang terbesar.
- Baris 12-13: Exception handling untuk input tidak valid.

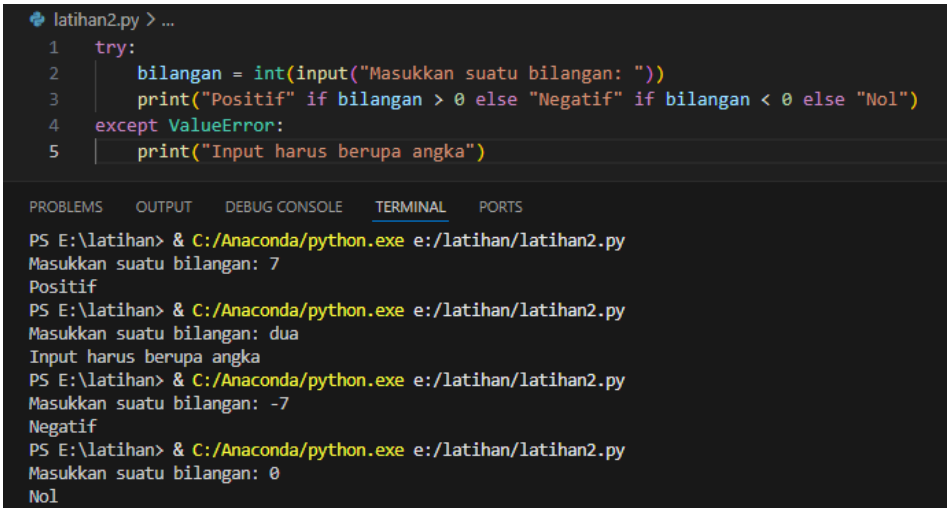
Note:

Menggunakan else untuk kondisi terakhir lebih efisien daripada elif c > a and c > b karena, jika a dan b bukan yang terbesar, maka otomatis c adalah yang terbesar.

SOAL 2

Implementasi percabangan menggunakan ternary operator.

Program ini mengimplementasikan pengecekan bilangan positif, negatif, atau nol menggunakan ternary operator. Ternary operator adalah cara penulisan percabangan yang lebih ringkas dalam satu baris.



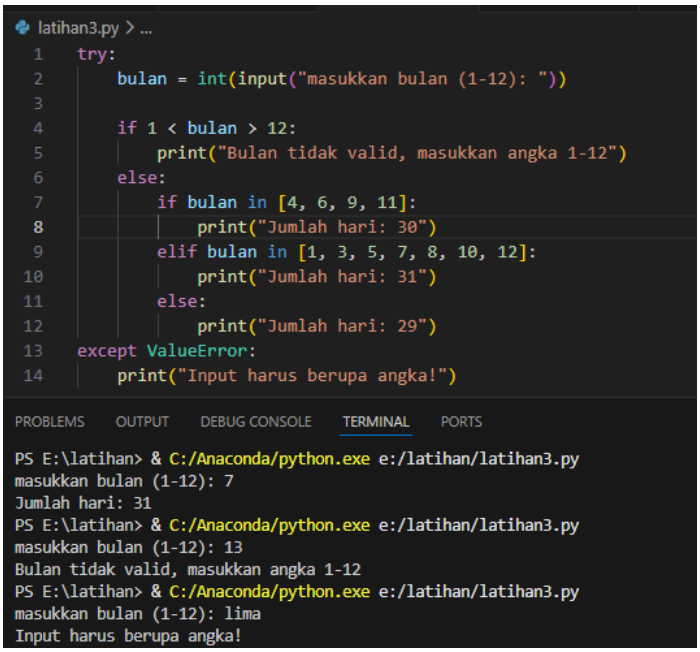
Gambar 4.10: Implementasi percabangan menggunakan ternary operator.

Penjelasan kode:

- Baris 1-2: Try block dan input bilangan.
- Baris 3: Nested ternary operator yang bekerja dari kiri ke kanan:
    1. Cek bilangan > 0? Jika ya, akan menghasilkan Positif.
    2. Cek bilangan < 0? Jika ya, akan menghasilkan Negatif.
    3. Jika tidak, otomatis akan menghasilkan Nol.
- Baris 4-5: Exception handling

SOAL 3

Program Jumlah Hari dalam Bulan



Gambar 4.11: Jumlah bulan dalam bulan

Program ini menampilkan jumlah hari dalamsuatu bulan di tahun 2020 (tahun kabisat). Pengguna memasukkan nomor buan (1-12), kemudian program menampilkan jumlah hari pada bulan tersebut.

Logika Program:

1. Bulan dengan 31 hari: Januari (1), Maret (3), Mei (5), Juli (7), Agustus (8), Oktober (10), Desember (12).
2. Bulan dengan 30 hari: April (4), Juni (6), September (9).
3. Februari (2): 29 hari (karena tahun 2020 adalah hari kabisat).

Penjelasan Kode:

- Meminta Input dari Pengguna

  Program meminta pengguna memasukkan nomor bulan 1-12. Input dikoncersi menjadi int, sehingga jika pengguna memasukkan huruf, programdapat menangani error dengan try-except.
- Memeriksa Apakah Input Valid

Program mengecek apakah angka yang dimasukkan diantara 1 hingga 12. Jika valid, program akan melanjutkan untuk menentukan jumlah hari.

- Menentukan Jumlah Hari dalam Bulan

  Jika bulan 1, 3, 5, 7, 810, 12 = jumlah hari: 31

  Jika bulan 4, 6, 9, 11 = jumlah hari: 30

  Jika bulan 2 = jumlah hari: 29 (karena tahun 2020 adalah tahun kabisat).

- Menangani Input dari luar 1-12

  Jika pegguna memasukkan angka diluar rentan misalnya 13 atau -5, program akan menampilkan pesan bahwa bulan tersebut tidak valid.

- Menangani Kesalahan Input

  Jika pengguna memasukkan huruf atau karakter lain misalnya empat, program akan memberikan peringatan bahwa input harus berupa angka.

SOAL 4

Perbandingan Sisi Segitiga

Gambar 4.12: Perbandingan sisi segitiga

Penjelasan Kode:

1. Program Meminta Input dari Pengguna

   Program meminta memasukkan tiga angka sebagai panjang sisi.

2. Menangani kesalahan Input

   Jika pengguna memasukkan selain angka, program akan menampilkan error.

3. Memaksa Validari Input
   - Program memastikan semua angka leboh besar dari nol.
   - Memeriksa apakah jumlah dua sisi mana lebih besar dari sisi ketiga.
   - Jika tidak valid, program menampilkan “Input tidak valid”.
4. Mengecek Kesamaan Sisi
   - Jika semua angka sama, maka program akan menampilkan “3 sisi sama”,
   - Jika dua angka sama, maka program akan menampilkan “2 sisi sama”,
   - Jika tidak ada angka yang sama, maka program akan menampilkan “tidak ada yang sama”,
5. Menampilkan Hasil

   Program mencetak hasil berdasarkan kondisi yang sesuai dengan input pengguna.