



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231050
Nama Lengkap	Samuel Natanael
Minggu ke / Materi	03 / 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%)

Boolean Expression dan Logical Operator

Dalam pemrograman, terkadang kita butuh mencari tahu apakah sebuah expression itu benar atau salah. Dalam Python, kita dapat menggunakan Boolean Expression yang terdiri hanya satu values yang bisa berupa True atau False. Berikut adalah contoh cara menggunakan Boolean Expression dalam kasus menentukan apakah usia pengguna mencukupi untuk mengakses suatu website dalam Python mode interaktif.

```
# Memasukan value usia
```

```
>>> usia = 18
```

```
# Mengecek apakah usianya melebihi atau sama dengan 18
```

```
>>> usia >= 18
```

```
True
```

```
>>> usia = 12
```

```
>>> usia >= 18
```

```
False
```

```
>>> usia = 30
```

```
>>> usia >= 18
```

```
True
```

Untuk menyusun Boolean expression, kita dapat menggunakan operator-operator perbandingan dibawah ini.

Operator	Contoh	Penjelasan
Sama dengan ==	1 == 1	bernilai True Jika masing-masing operan memiliki nilai yang sama, maka kondisi bernilai benar atau True.
Tidak sama dengan !=	2 != 2	bernilai False Akan menghasilkan nilai kebalikan dari kondisi sebenarnya.
Tidak sama dengan <>	2 <> 2	bernilai False Akan menghasilkan nilai kebalikan dari kondisi sebenarnya.
Lebih besar dari >	5 > 3	bernilai True Jika nilai operan kiri lebih besar dari nilai operan kanan, maka kondisi menjadi benar.
Lebih kecil dari <	5 < 3	bernilai True Jika nilai operan kiri lebih kecil dari nilai operan kanan, maka kondisi menjadi benar.
Lebih besar atau sama dengan >=	5 >= 3	bernilai True Jika nilai operan kiri lebih besar dari nilai operan kanan, atau sama, maka kondisi menjadi benar.
Lebih kecil atau sama dengan <=	5 <= 3	bernilai True Jika nilai operan kiri lebih kecil dari nilai operan kanan, atau sama, maka kondisi menjadi benar.

Gambar 3.1 Operator-Operator Perbandingan (diambil dari <https://belajarpython.com/tutorial/operator-python/>)

Untuk membuat Boolean Expression dengan benar, penggunaan operator harus sesuai dan tepat untuk masalah anda. Ada beberapa hal yang perlu diperhatikan saat membuat ungkapan Boolean yaitu :

- Bentuk boolean expression pasti menghasilkan hanya dua pilihan, yaitu benar atau salah.
- Perhatikan istilah khusus seperti minimum, maksimum, tidak lebih dari, tidak kurang dari, tidak sama, dan tidak berbeda.
- Perhatikan dengan cermat dan tentukan variabel yang harus dibandingkan dengan benar sesuai dengan masalah.

Logical operator juga dapat digunakan untuk menggabungkan beberapa ekspresi boolean. Logical operator Python adalah and, or, dan not. Contoh penggunaannya adalah ketika memberi akses masuk ke dalam gym yang memiliki dua syarat yaitu memiliki member dan melebihi umur 18.

Usia ≥ 18 and member == True

Makanan gratis diberikan bagi yang sudah menjadi member lebih dari 6 bulan atau member premium.

Bulan_member > 6 or premium == True

Contoh masalah dalam Boolean expression :

- Siswa dengan surat peringatan melebihi tiga akan dikeluarkan dari kelas

Surat_Peringatan > 3

- Jona akan mendapatkan bonus jika nilainya lebih dari 90

Nilai > 90

- Minimal 2 medali emas dibutuhkan untuk masuk ke kuliah ini

Medali ≥ 2

- Uang saku saya sebulan 1 juta, Timmy 2 juta

Saya $<$ Timmy

- Nilai AIPro saya 100, dan Jona 100

Saya == Jona

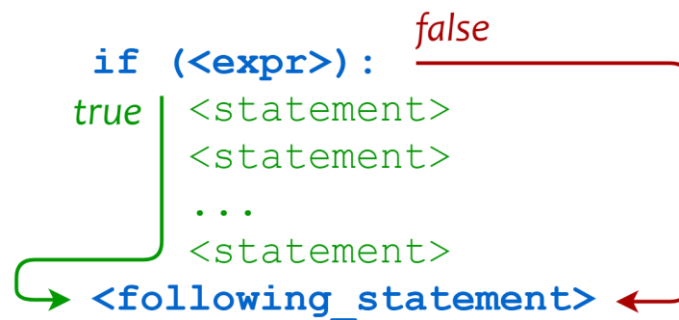
- Jumlah baju saya ada 2, Jona ada 3, dan Imel ada 5. Apakah baju Jona dan saya digabung sama dengan baju Imel

Saya + Jona == Imel

Bentuk-Bentuk Percabangan

Ada tiga bentuk percabangan pada Python yaitu: **conditional**, **alternative** dan **chained conditional**.

Bentuk Conditional:



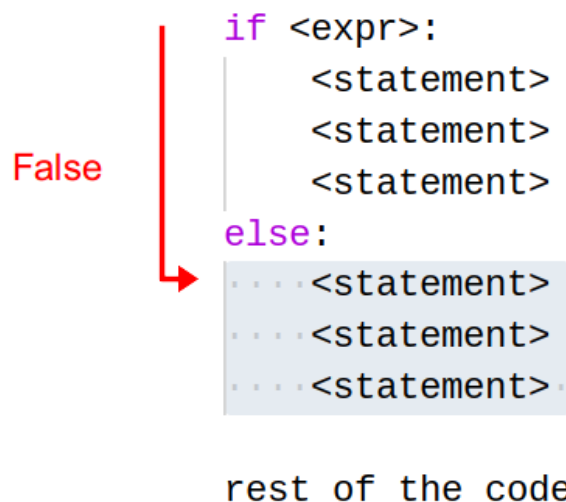
Gambar 3.2 Python Conditional (diambil dari <https://realpython.com/python-conditional-statements/>)

Contohnya seperti menentukan apakah cairan bersifat asam ($\text{ph} < 7$):

if $\text{ph} < 7$:

```
    print("asam")
```

Bentuk conditional juga bisa ditambahkan dengan "else" sebagai langkah alternatif yang harus dijalankan berdasarkan kondisi tertentu.

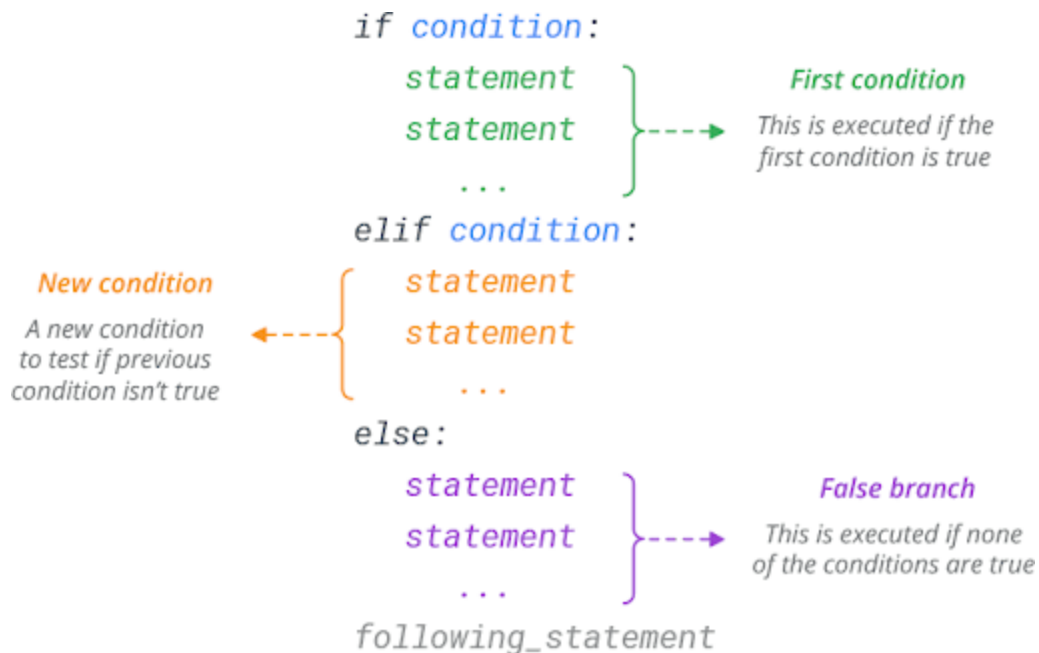


Gambar 3.3 Python Alternative Conditional (diambil dari <https://pythonprogramminglanguage.com/if-statements/>)

Contohnya seperti menentukan apakah cairan bersifat netral atau tidak:

```
if ph == 7:
    print("netral")
else:
    print("tidak netral")
```

Bentuk conditional yang ketiga adalah bentuk chained conditional, yang digunakan ketika ada kondisi yang harus dijalankan melebihi 2.



Gambar 3.4 Python Chained Conditional (diambil dari <https://internetpositif.id/blog/python-conditionals-if-else-elif-statements/>)

Contohnya adalah ketika sebuah restoran yang memberikan roti gratis dengan melihat umur pembeli. Jika berumur kurang dari 15 tahun maka mendapatkan 1 buah roti, 15 sampai 30 tahun mendapatkan 3 buah roti, dan 30 sampai 40 mendapatkan 4 roti, dan 40 keatas mendapatkan 5 roti. Ada 4 kemungkinan output yaitu 1, 3, 4, dan 5 buah roti.

```
if umur < 15:
    print("1 roti")
elif umur >= 15 and umur < 30:
    print("3 roti")
elif umur >= 30 and umur < 40:
    print("4 roti")
```

else:

```
print("5 roti")
```

Boolean Expression “and” bisa digunakan untuk menggabungkan kondisi seperti “15 sampai 30 tahun mendapatkan 3 buah roti” karena dua kondisi tersebut harus dipenuhi untuk berjalan.

```
umur >= 15 and umur < 30
```

Penulisan percabangan juga bisa ditulis dengan ternary operator dengan bentuk seperti berikut

Normal:

```
if ph == 7:
```

```
    rasa = "normal"
```

else:

```
    rasa = "tidak normal"
```

Ternary:

```
rasa = "normal" if ph == 7 else "tidak normal"
```

Penanganan Kesalahan Input Menggunakan Exception Handling

Dalam menangani input pengguna, kita juga harus mempertimbangkan kesalahan yang mungkin terjadi sehingga program tidak berfungsi dengan benar. Contohnya adalah ketika sebuah code meminta integer, tetapi pengguna memasukan string.

```
umur = int(input("Masukan Umur : "))
```

Masukan Umur : tujuh belas

Traceback (most recent call last):

File "d:\Ngampus\Semester 2\PrakAIPro\Tes.py", line 30, in <module>

```
    umur = int(input("Masukan Umur : "))
```

```
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

ValueError: invalid literal for int() with base 10: 'tujuh belas'

Salah satu cara yang bisa digunakan adalah dengan **try** dan **except**.



Gambar 3.5 **try** dan **except** Pada Python (diambil dari <https://www.learnbyexample.org/python-exceptions-try-except/>)

Supaya bisa dimengerti lebih dalam, perhatikan program dibawah ini yang menentukan nilai murid. Berikut adalah sistem penilaiannya.

A = 90 - 100

B = 80 - 89

C = 70 - 79

D = 60 - 69

Remedial = < 60

```
1 # Code "try" akan dirun terlebih dahulu
2 try:
3     nilai = int(input("Masukan Nilai : "))
4     if nilai >= 90 and nilai <= 100:
5         print("A")
6     elif nilai >= 80 and nilai < 90:
7         print("B")
8     elif nilai >= 70 and nilai < 80:
9         print("C")
10    elif nilai >= 60 and nilai < 70:
11        print("D")
12    else:
13        print("Remedial")
14 # Jika Code "try" mengalami error maka code "except" akan dijalankan
15 except ValueError:
16     print("Masukan angka 1-100")
```

Terminal (Ctrl+)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/tes2.py"

Masukan Nilai : 100

A

PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/tes2.py"

Masukan Nilai : seratus

Masukan angka 1-100

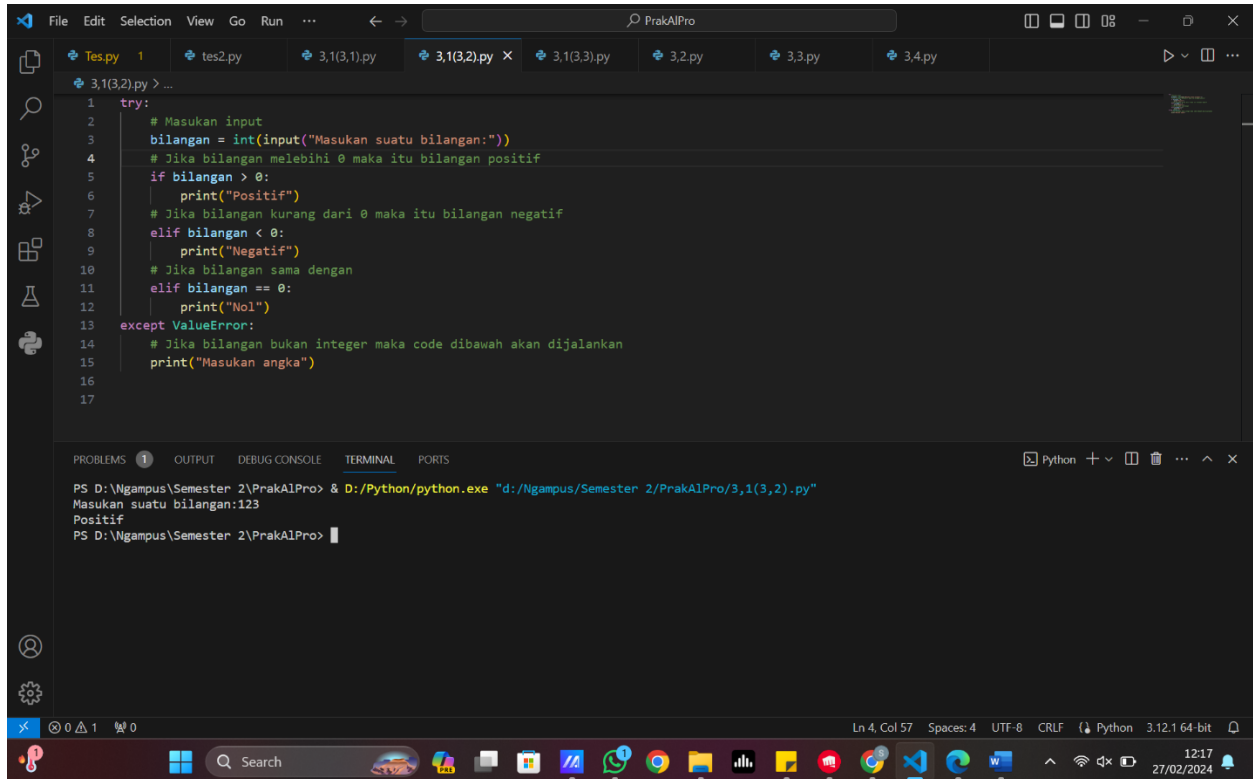
PS D:\Ngampus\Semester 2\PrakAlPro>

ValueError di depan **except** artinya code didalam **except** akan berjalan ketika ada ValueError didalam code **try**

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 3.1

Contoh 3,1



The screenshot shows a Visual Studio Code editor window with a Python script named `3,1(3,2).py`. The script is a program that checks if a user input is a positive, negative, or zero integer. It uses a `try` block for the input and an `except ValueError` block for non-integer inputs. The terminal at the bottom shows the command to run the script, the input `123`, and the output `Positif`.

```
1 try:
2     # Masukan input
3     bilangan = int(input("Masukan suatu bilangan:"))
4     # Jika bilangan melebihi 0 maka itu bilangan positif
5     if bilangan > 0:
6         print("Positif")
7     # Jika bilangan kurang dari 0 maka itu bilangan negatif
8     elif bilangan < 0:
9         print("Negatif")
10    # Jika bilangan sama dengan
11    elif bilangan == 0:
12        print("Nol")
13 except ValueError:
14    # Jika bilangan bukan integer maka code dibawah akan dijalankan
15    print("Masukan angka")
16
17
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,1(3,2).py"

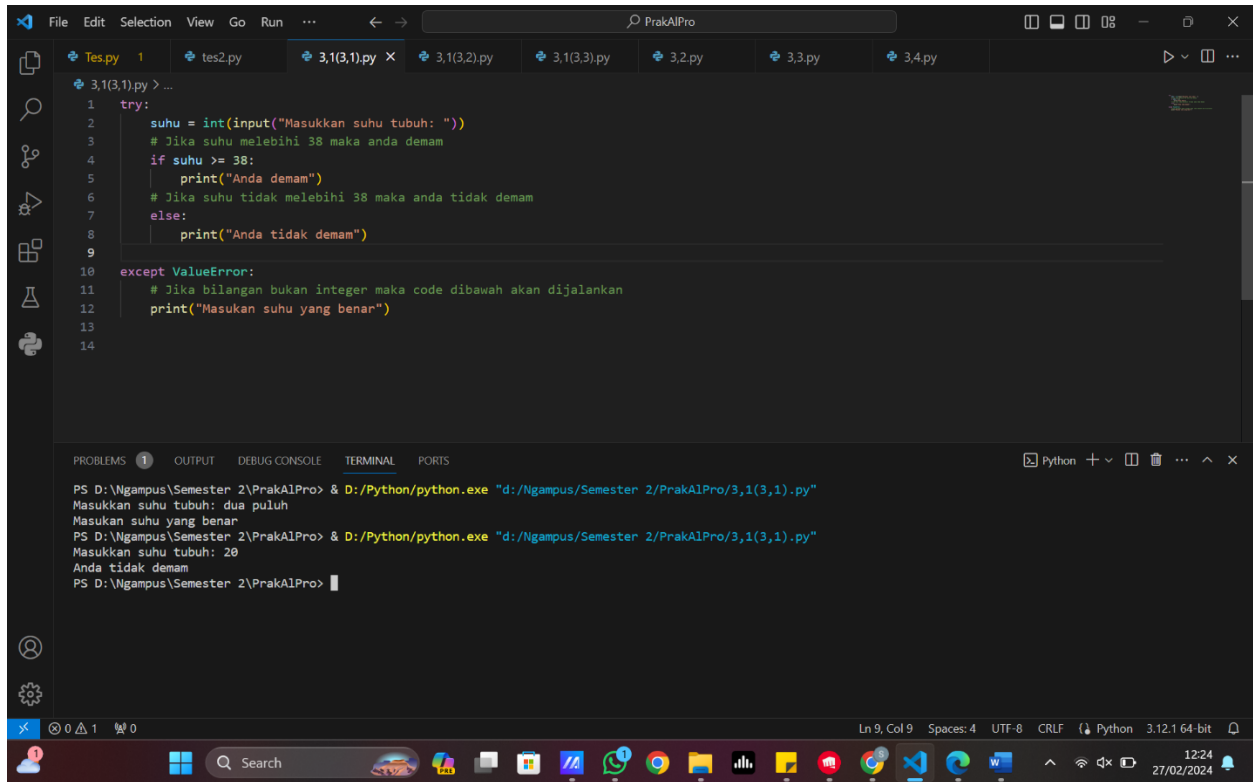
Masukan suatu bilangan:123

Positif

PS D:\Ngampus\Semester 2\PrakAlPro> |

Ln 4, Col 57 Spaces: 4 UTF-8 CRLF Python 3.12.1 64-bit

Contoh 3,2



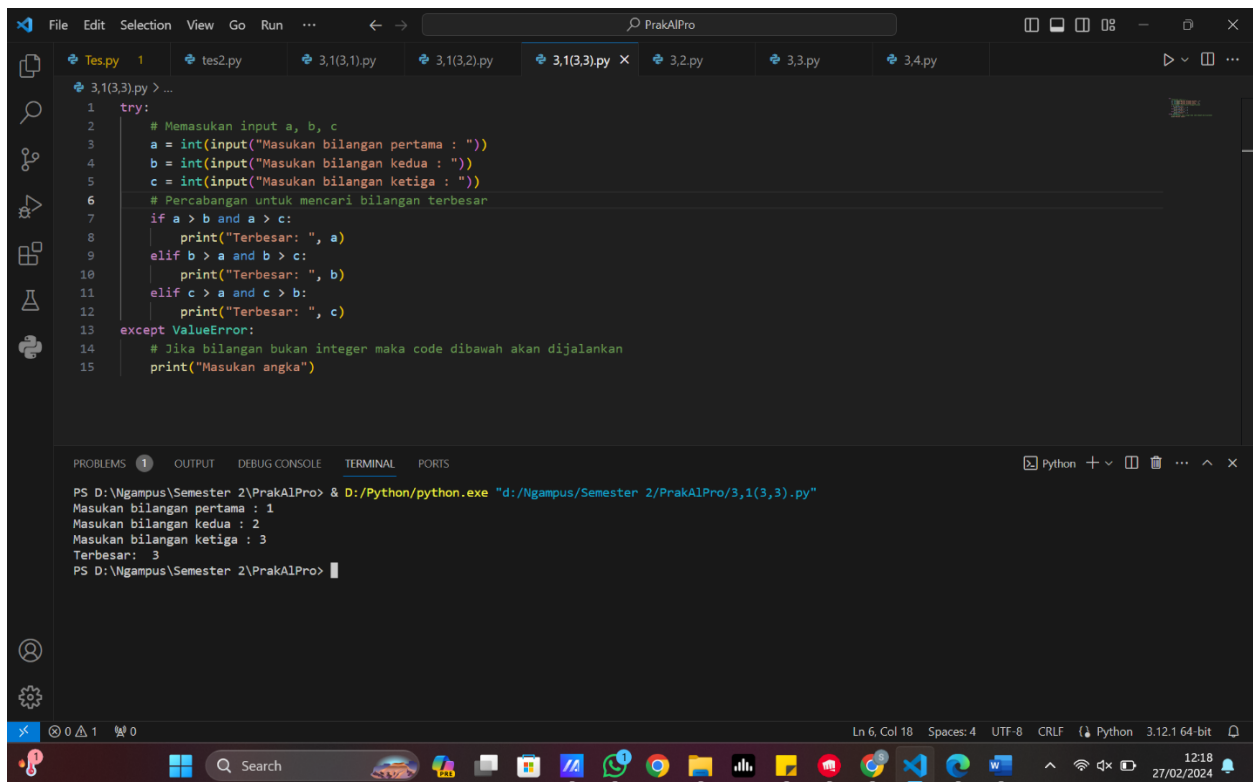
The screenshot shows the VS Code editor with a Python file named `3,1(3,1).py`. The code is a script that takes a temperature input and checks if it is above 38 degrees Celsius. If it is, it prints "Anda demam"; otherwise, it prints "Anda tidak demam". It also includes a try-except block to handle non-integer inputs, printing "Masukan suhu yang benar" if a `ValueError` occurs.

```
1 try:
2     suhu = int(input("Masukkan suhu tubuh: "))
3     # Jika suhu melebihi 38 maka anda demam
4     if suhu >= 38:
5         print("Anda demam")
6     # Jika suhu tidak melebihi 38 maka anda tidak demam
7     else:
8         print("Anda tidak demam")
9
10 except ValueError:
11     # Jika bilangan bukan integer maka code dibawah akan dijalankan
12     print("Masukan suhu yang benar")
13
14
```

The terminal output shows the execution of the script. It prompts the user to enter a temperature, and when the user enters 20, it prints "Anda tidak demam".

```
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,1(3,1).py"
Masukkan suhu tubuh: dua puluh
Masukan suhu yang benar
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,1(3,1).py"
Masukkan suhu tubuh: 20
Anda tidak demam
PS D:\Ngampus\Semester 2\PrakAlPro>
```

Contoh 3,3



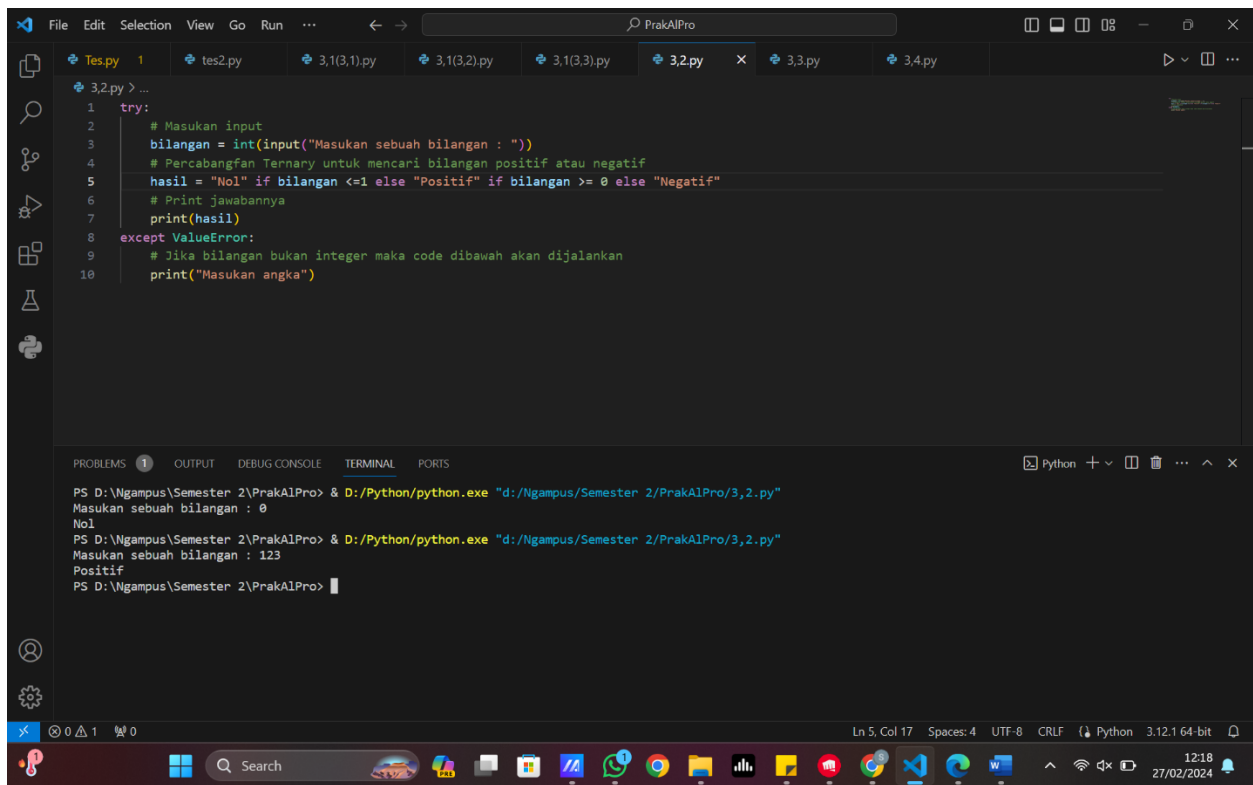
The screenshot shows the VS Code editor with a Python file named `3,1(3,3).py`. The code is a script that takes three integer inputs and finds the maximum among them. It uses a series of if-elif statements to compare the values. It also includes a try-except block to handle non-integer inputs, printing "Masukan angka" if a `ValueError` occurs.

```
1 try:
2     # Memasukan input a, b, c
3     a = int(input("Masukan bilangan pertama : "))
4     b = int(input("Masukan bilangan kedua : "))
5     c = int(input("Masukan bilangan ketiga : "))
6     # Percabangan untuk mencari bilangan terbesar
7     if a > b and a > c:
8         print("Terbesar: ", a)
9     elif b > a and b > c:
10        print("Terbesar: ", b)
11    elif c > a and c > b:
12        print("Terbesar: ", c)
13 except ValueError:
14    # Jika bilangan bukan integer maka code dibawah akan dijalankan
15    print("Masukan angka")
```

The terminal output shows the execution of the script. It prompts the user to enter three numbers: 1, 2, and 3. It then prints "Terbesar: 3".

```
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,1(3,3).py"
Masukan bilangan pertama : 1
Masukan bilangan kedua : 2
Masukan bilangan ketiga : 3
Terbesar: 3
PS D:\Ngampus\Semester 2\PrakAlPro>
```

SOAL 3.2



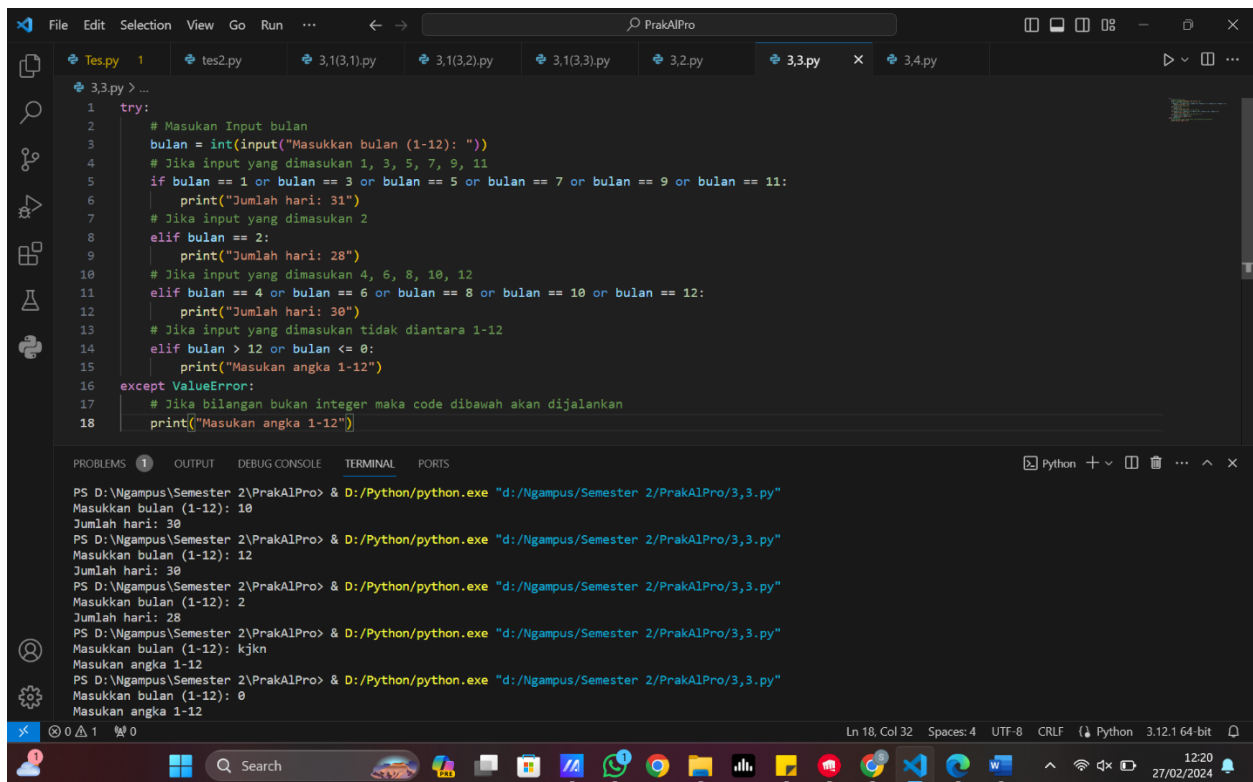
The screenshot shows the VS Code editor with a file named `3,2.py` open. The code is a Python script that prompts the user to enter a number. It uses a try-except block to handle `ValueError` if the input is not an integer. The script uses a ternary operator to determine if the number is positive, negative, or zero.

```
1 try:
2     # Masukan input
3     bilangan = int(input("Masukan sebuah bilangan : "))
4     # Percabangan Ternary untuk mencari bilangan positif atau negatif
5     hasil = "Noi" if bilangan <=1 else "Positif" if bilangan >= 0 else "Negatif"
6     # Print jawabannya
7     print(hasil)
8 except ValueError:
9     # Jika bilangan bukan integer maka code dibawah akan dijalankan
10    print("Masukan angka")
```

The terminal output shows the script being executed twice. The first time, the user enters `0` and the output is `Noi`. The second time, the user enters `123` and the output is `Positif`.

```
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,2.py"
Masukan sebuah bilangan : 0
Noi
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,2.py"
Masukan sebuah bilangan : 123
Positif
PS D:\Ngampus\Semester 2\PrakAlPro>
```

SOAL 3.3



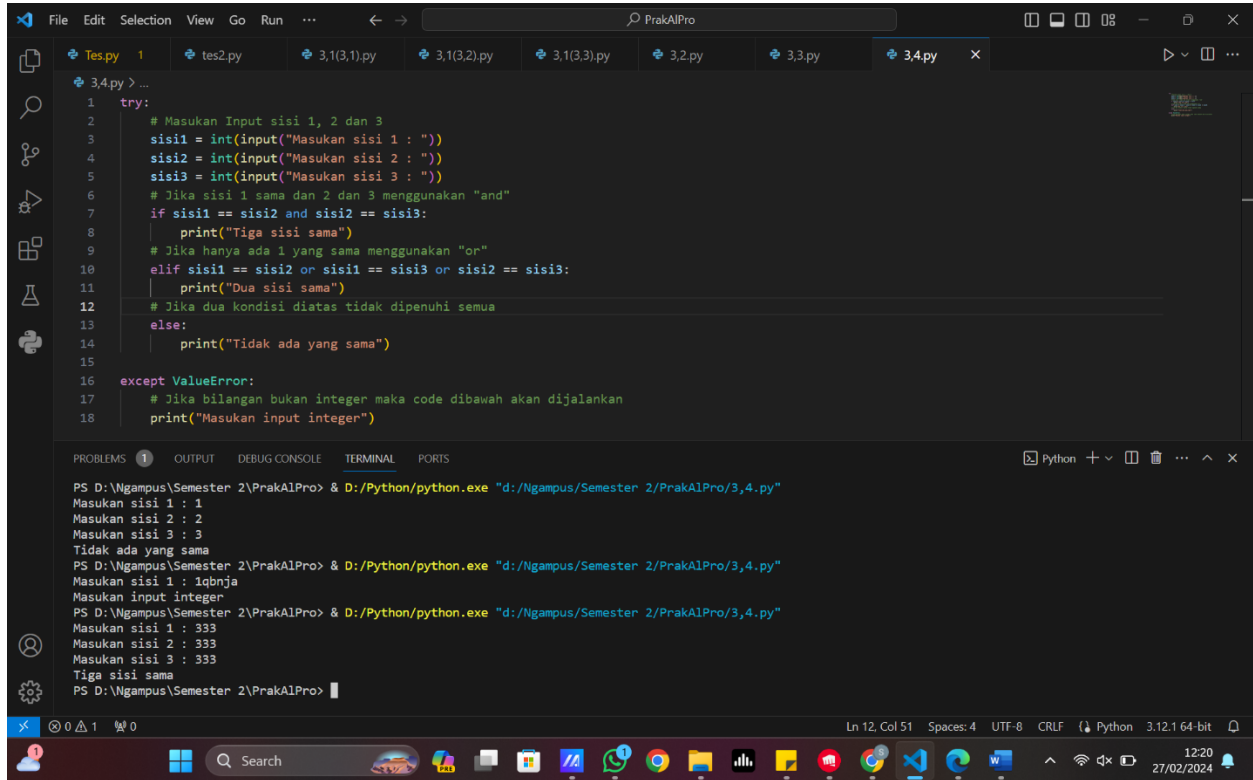
The screenshot shows the VS Code editor with a file named `3,3.py` open. The code is a Python script that prompts the user to enter a year. It uses a series of if-elif-else statements to determine if the year is a leap year. The script uses a try-except block to handle `ValueError` if the input is not an integer.

```
1 try:
2     # Masukan Input bulan
3     bulan = int(input("Masukkan bulan (1-12): "))
4     # Jika input yang dimasukan 1, 3, 5, 7, 9, 11
5     if bulan == 1 or bulan == 3 or bulan == 5 or bulan == 7 or bulan == 9 or bulan == 11:
6         print("Jumlah hari: 31")
7     # Jika input yang dimasukan 2
8     elif bulan == 2:
9         print("Jumlah hari: 28")
10    # Jika input yang dimasukan 4, 6, 8, 10, 12
11    elif bulan == 4 or bulan == 6 or bulan == 8 or bulan == 10 or bulan == 12:
12        print("Jumlah hari: 30")
13    # Jika input yang dimasukan tidak diantara 1-12
14    elif bulan > 12 or bulan <= 0:
15        print("Masukan angka 1-12")
16 except ValueError:
17    # Jika bilangan bukan integer maka code dibawah akan dijalankan
18    print("Masukan angka 1-12")
```

The terminal output shows the script being executed five times with different inputs. The first four times, the user enters a valid month (1, 12, 2, 1) and the output is the number of days in that month. The fifth time, the user enters `kjkn` and the output is `Masukan angka 1-12`.

```
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,3.py"
Masukkan bulan (1-12): 10
Jumlah hari: 30
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,3.py"
Masukkan bulan (1-12): 12
Jumlah hari: 30
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,3.py"
Masukkan bulan (1-12): 2
Jumlah hari: 28
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,3.py"
Masukkan bulan (1-12): kjkn
Masukan angka 1-12
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,3.py"
Masukkan bulan (1-12): 0
Masukan angka 1-12
```

SOAL 3.4



The screenshot shows a Visual Studio Code editor window with a Python file named `3,4.py`. The code is a script that takes three inputs and checks if they are all the same, two are the same, or none are the same. It also includes a try-except block to handle non-integer inputs.

```
1 try:
2     # Masukan Input sisi 1, 2 dan 3
3     sisi1 = int(input("Masukan sisi 1 : "))
4     sisi2 = int(input("Masukan sisi 2 : "))
5     sisi3 = int(input("Masukan sisi 3 : "))
6     # Jika sisi 1 sama dan 2 dan 3 menggunakan "and"
7     if sisi1 == sisi2 and sisi2 == sisi3:
8         print("Tiga sisi sama")
9     # Jika hanya ada 1 yang sama menggunakan "or"
10    elif sisi1 == sisi2 or sisi1 == sisi3 or sisi2 == sisi3:
11        print("Dua sisi sama")
12    # Jika dua kondisi diatas tidak dipenuhi semua
13    else:
14        print("Tidak ada yang sama")
15
16 except ValueError:
17     # Jika bilangan bukan integer maka code dibawah akan dijalankan
18     print("Masukan input integer")
```

The terminal output shows three test cases:

```
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,4.py"
Masukan sisi 1 : 1
Masukan sisi 2 : 2
Masukan sisi 3 : 3
Tidak ada yang sama
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,4.py"
Masukan sisi 1 : 1qbnja
Masukan input integer
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/3,4.py"
Masukan sisi 1 : 333
Masukan sisi 2 : 333
Masukan sisi 3 : 333
Tiga sisi sama
PS D:\Ngampus\Semester 2\PrakAlPro>
```

Github : <https://github.com/SamuelN1508/PrakAlPro3.git>