



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231051
Nama Lengkap	Amida A Ronsumbre
Minggu ke / Materi	06/ Percabangan dan Perulangan Kompleks

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

- 6.3.1 Struktur Percabangan Kompleks

Percabangan kompleks bentuk 1 yang Anda sebutkan menggunakan konsep nested if, yaitu percabangan di dalam percabangan. Ini memungkinkan program untuk mengevaluasi kondisi lebih dari satu dan menjalankan blok kode yang berbeda berdasarkan hasil evaluasi tersebut. Strukturnya melibatkan pemeriksaan kondisi pertama, lalu di dalamnya memeriksa kondisi kedua. Ada blok kode yang dieksekusi jika kedua kondisi terpenuhi, dan blok kode lain yang dieksekusi jika hanya kondisi pertama yang terpenuhi (kondisi kedua tidak terpenuhi). Ini memperluas kemungkinan logika yang dapat diterapkan dalam pemrograman untuk menangani situasi yang lebih kompleks.

- Percabangan kompleks bentuk 2:

Percabangan kompleks bentuk 3 dimulai dengan kondisi1, diikuti oleh dua cabang tambahan bergantung pada evaluasi kondisi2 dan kondisi3 secara berturut-turut. Jika kondisi1 terpenuhi, baris 2 dieksekusi, kemudian dilakukan pemeriksaan kondisi2. Jika kondisi2 terpenuhi, baris 4 dan 5 dieksekusi; jika tidak, program melanjutkan ke baris 6. Jika kondisi1 tidak terpenuhi, program langsung ke baris 6. Pada baris 6, dilakukan pemeriksaan kondisi3. Jika kondisi3 terpenuhi, baris 8 dan 9 dieksekusi; jika tidak, baris 11 dan 12 dieksekusi.

- Percabangan kompleks bentuk 3:

Percabangan kompleks bentuk 2 ini terdiri dari dua tingkat percabangan. Jika kondisi1 terpenuhi, maka ada pemeriksaan tambahan pada kondisi2. Jika kondisi2 terpenuhi, dijalankan dua perintah S pada baris 4 dan 5. Jika kondisi2 tidak terpenuhi, program melanjutkan ke percabangan kedua. Pada percabangan kedua, jika kondisi3 terpenuhi, dijalankan dua perintah S pada baris 8 dan 9; jika tidak, dijalankan dua perintah S pada baris 11 dan 12.

- Percabangan kompleks

Penggunaan if bertingkat memungkinkan evaluasi kondisi yang efisien karena hanya kondisi yang relevan yang dievaluasi, menghemat waktu eksekusi. Ini berbeda dengan if biasa di mana semua kondisi dievaluasi satu per satu, bahkan jika kondisi sebelumnya sudah terpenuhi. Contoh penggunaan yang cocok untuk if bertingkat adalah dalam kasus konversi nilai angka menjadi nilai huruf dengan batasan nilai yang sudah ditentukan sebelumnya.

- 6.3.2 Struktur Perulangan Kompleks Break

Perintah `break` digunakan untuk menghentikan loop sebelum waktunya berdasarkan kondisi tertentu, `continue` membuat loop langsung melanjutkan ke iterasi berikutnya mengabaikan baris kode setelahnya dalam loop saat itu, dan perulangan bertingkat adalah ketika satu loop berjalan di dalam loop lain, sering digunakan untuk masalah yang melibatkan struktur data multi-dimensi.

Perintah `break` digunakan untuk menghentikan proses perulangan sebelum mencapai akhir secara alami, biasanya dikontrol oleh kondisi tertentu dengan `if`. Contoh yang diberikan menunjukkan bagaimana `break` menghentikan loop `for` setelah mencapai nilai tertentu, dengan perbedaan hasil output tergantung pada posisi perintah `print` relatif terhadap `break`.

Perintah `continue` menyebabkan perulangan langsung kembali ke awal, mengabaikan perintah-perintah yang berada di bawahnya dan belum dieksekusi dalam iterasi saat itu. Contoh yang diberikan menunjukkan `continue` yang digunakan untuk melewati sebuah nilai tertentu dalam loop, sehingga nilai tersebut tidak muncul dalam output.

MATERI 2

Penjelasan materi 2, dst... sesuai format ini.

- 6.4 Kegiatan Praktikum

Program ini menciptakan sebuah deret angka berdasarkan input pengguna (n). Deret tersebut menampilkan angka sebanyak n baris, dengan setiap baris menampilkan angka yang sama sebanyak nomor baris tersebut. Logika programnya menggunakan nested loop, di mana loop luar (outer loop) mengontrol jumlah baris, sementara loop dalam (inner loop) mengontrol jumlah angka yang ditampilkan dalam setiap baris. Outputnya akan menampilkan deret angka sesuai dengan pola yang dijelaskan dalam contoh.

■ Contoh 6.2 Anda diminta untuk membuat suatu deret dengan tampilan sebagai berikut:

Program ini menciptakan sebuah pola deret angka berdasarkan input pengguna (n). Pola tersebut terdiri dari serangkaian baris dengan pola tertentu, di mana pada baris ganjil angka dari 1 hingga n ditampilkan secara berurutan, sedangkan pada baris genap angka dari n hingga 1 ditampilkan secara berurutan. Logika programnya menggunakan nested loop, di mana loop luar (outer loop) mengontrol jumlah baris, dan loop dalam (inner loop) mengontrol pola angka dalam setiap baris, dengan memperhatikan apakah baris tersebut ganjil atau genap. Outputnya akan menampilkan pola deret angka sesuai dengan pola yang dijelaskan dalam contoh.

■ Contoh 6.3 Anda diminta untuk membuat suatu deret dengan tampilan sebagai berikut:

Program ini menciptakan sebuah pola berbasis karakter (X dan O) berdasarkan input pengguna (n). Pola tersebut terdiri dari serangkaian baris dengan pola tertentu, di mana pada setiap baris karakter 'X' dan 'O' ditampilkan secara bergantian, dimulai dengan 'X'. Logika programnya menggunakan nested loop, di mana loop luar (outer loop) mengontrol jumlah baris, dan loop dalam (inner loop) mengontrol pola karakter dalam setiap baris, dengan menggunakan kondisi untuk menentukan karakter yang harus ditampilkan pada posisi tertentu dalam baris. Outputnya akan menampilkan pola karakter 'X' dan 'O' sesuai dengan pola yang dijelaskan dalam contoh.

```
ug6.py >
1 n = int(input("Masukkan n = "))
2 for i in range(0, n + 1):
3     for j in range(1, n - i + 1):
4         if j % 2 == 1:
5             print("X", end='')
6         else:
7             print("O", end='')
8     print()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\asus\Documents\A.Kuliah\B.Semester 2\pra alpro> python -u "C:\Users\asus\Documents\A.Kuliah\B.Semester 2\pra alpro\ug6.py"

Masukkan n = 6

XOXOXO

XOXOX

XOXO

XOX

XO

X

■ Contoh 6.4 Anda diminta untuk membuat suatu deret dengan tampilan:

Berikut adalah ringkasan logika program untuk mencetak pola segitiga bintang:

Loop luar mengontrol jumlah baris dari atas ke bawah, dimulai dari n turun ke 1.

Setiap baris memiliki spasi kosong sesuai dengan jumlah $(n - i - 1)$, di mana i adalah nomor baris saat ini.

Setelah spasi kosong ditambahkan, bintang (*) ditambahkan sebanyak $(1 \text{ sampai } n - i - 1)$ untuk membentuk pola segitiga.

Proses ini diulang untuk setiap baris sampai baris terakhir.

Berikut adalah implementasi kode programnya:

```
1 hasil = ""
2 x = int(input("Masukkan jumlah :"))
3 bar = x
4 while bar >= 0:
5     kol = bar
6     while kol > 0:
7         hasil += " "
8         kol -= 1
9     kanan = 1
10    while kanan < (x - (bar-1)):
11        hasil += " "
12        kanan += 1
13    hasil = hasil + "\n"
14    bar -= 1
15    print(hasil)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\asus\Documents\A.Kuliah\B.Semester 2\pra alpro> python -u "C:\Users\asus\Documents\A.Kuliah\B.Semester 2\pra alpro\ug6.py"

Masukkan jumlah :10

*
 *
 *
 *
 *
 *
 *
 *
 *
 *

Program tersebut menciptakan pola segitiga bintang berdasarkan jumlah baris yang dimasukkan pengguna. Langkah-langkahnya adalah sebagai berikut:

1. Meminta pengguna untuk memasukkan jumlah baris segitiga.
2. Melakukan iterasi untuk setiap baris dari atas ke bawah.
3. Di setiap baris, menambahkan spasi kosong sesuai dengan nomor baris saat ini.
4. Menambahkan bintang (*) sesuai dengan pola segitiga pada setiap baris.

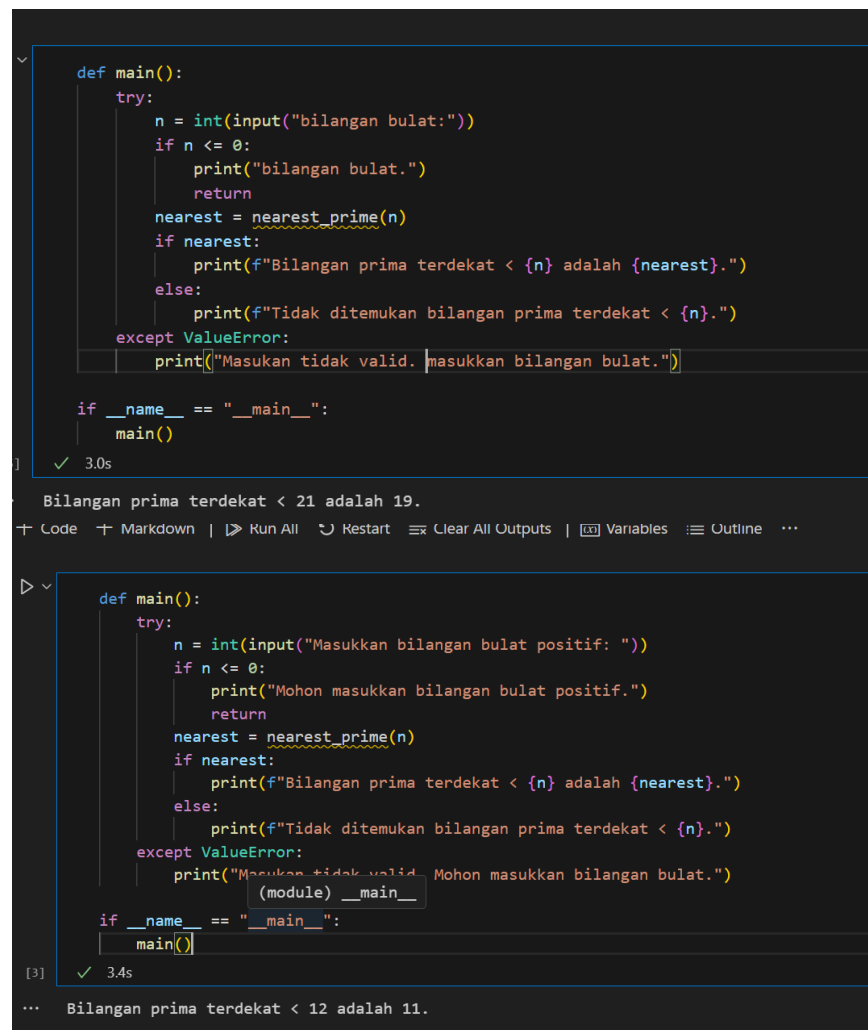
5. Mengurangi jumlah baris setelah selesai satu iterasi.
6. Mencetak pola segitiga yang telah dibuat.

Hasilnya adalah pola segitiga bintang yang sesuai dengan jumlah baris yang dimasukkan pengguna.

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



```
def main():
    try:
        n = int(input("bilangan bulat:"))
        if n <= 0:
            print("bilangan bulat.")
            return
        nearest = nearest_prime(n)
        if nearest:
            print(f"Bilangan prima terdekat < {n} adalah {nearest}.")
        else:
            print(f"Tidak ditemukan bilangan prima terdekat < {n}.")
    except ValueError:
        print("Masukan tidak valid. Masukkan bilangan bulat.")

if __name__ == "__main__":
    main()
```

✓ 3.0s

Bilangan prima terdekat < 21 adalah 19.

```
def main():
    try:
        n = int(input("Masukkan bilangan bulat positif: "))
        if n <= 0:
            print("Mohon masukkan bilangan bulat positif.")
            return
        nearest = nearest_prime(n)
        if nearest:
            print(f"Bilangan prima terdekat < {n} adalah {nearest}.")
        else:
            print(f"Tidak ditemukan bilangan prima terdekat < {n}.")
    except ValueError:
        print("Masukan tidak valid. Mohon masukkan bilangan bulat.")

if __name__ == "__main__":
    main()
```

[3] ✓ 3.4s

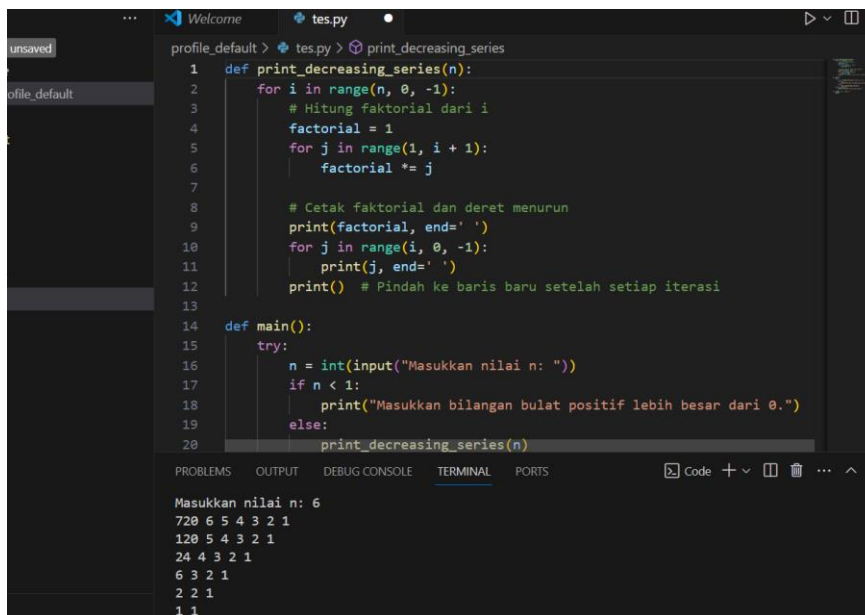
... Bilangan prima terdekat < 12 adalah 11.

Fungsi main() adalah titik masuk program, meminta input dari pengguna, dan kemudian menampilkan bilangan prima terdekat yang lebih kecil dari bilangan tersebut. Jika input pengguna kurang dari 2, program akan memberitahu bahwa tidak ada bilangan prima yang memenuhi kriteria tersebut.

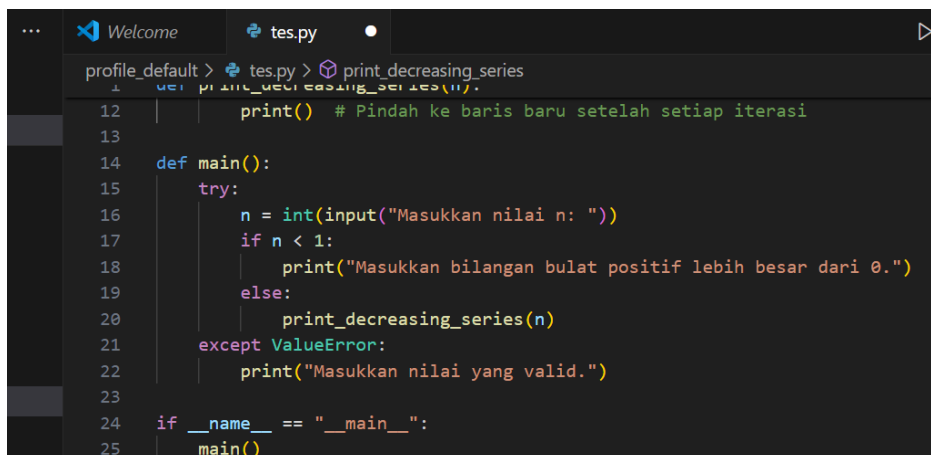
Fungsi main() adalah titik masuk program, meminta input dari pengguna, dan kemudian menampilkan bilangan prima terdekat yang lebih kecil dari bilangan tersebut. Jika input pengguna kurang dari 2, program akan memberitahu bahwa tidak ada bilangan prima yang memenuhi kriteria tersebut. Fungsi main() adalah titik masuk program, meminta input dari pengguna, dan kemudian menampilkan bilangan prima terdekat yang lebih kecil dari bilangan tersebut. Jika input pengguna kurang dari 2, program akan memberitahu bahwa tidak ada bilangan prima yang memenuhi kriteria tersebut.

SOAL 2

Tulis jawaban anda untuk soal nomor 2 di sini. Format untuk soal nomor 3 dan seterusnya juga sama.



```
profile_default > tes.py > print_decreasing_series
1 def print_decreasing_series(n):
2     for i in range(n, 0, -1):
3         # Hitung faktorial dari i
4         factorial = 1
5         for j in range(1, i + 1):
6             factorial *= j
7
8         # Cetak faktorial dan deret menurun
9         print(factorial, end=' ')
10        for j in range(i, 0, -1):
11            print(j, end=' ')
12        print() # Pindah ke baris baru setelah setiap iterasi
13
14 def main():
15     try:
16         n = int(input("Masukkan nilai n: "))
17         if n < 1:
18             print("Masukkan bilangan bulat positif lebih besar dari 0.")
19         else:
20             print_decreasing_series(n)
21
22 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Masukkan nilai n: 6
720 6 5 4 3 2 1
120 5 4 3 2 1
24 4 3 2 1
6 3 2 1
2 2 1
1 1
```



```
profile_default > tes.py > print_decreasing_series
12 print() # Pindah ke baris baru setelah setiap iterasi
13
14 def main():
15     try:
16         n = int(input("Masukkan nilai n: "))
17         if n < 1:
18             print("Masukkan bilangan bulat positif lebih besar dari 0.")
19         else:
20             print_decreasing_series(n)
21     except ValueError:
22         print("Masukkan nilai yang valid.")
23
24 if __name__ == "__main__":
25     main()
```

Program ini pertama-tama menghitung faktorial dari n dan kemudian mencetaknya di awal setiap baris. Setelah itu, program mencetak deret angka yang menurun sampai 1. Ini dilakukan untuk setiap nilai mulai dari n sampai 1. Maaf atas kebingungan sebelumnya, dan semoga solusi ini sesuai dengan yang Anda butuhkan.

SOAL 3

```

profile_default > tes.py > ...
1  def print_series(height, width):
2      """Fungsi untuk menampilkan deret sesuai permintaan."""
3      start = 1
4      for i in range(height):
5          for j in range(width):
6              print(start, end=" ")
7              start += 1
8          print()
9
10 def main():
11     try:
12         height = int(input("Masukkan tinggi: "))
13         width = int(input("Masukkan lebar: "))
14         if height < 1 or width < 1:
15             print("Tinggi dan lebar harus lebih besar dari 0.")
16         else:
17             print_series(height, width)
18     except ValueError:
19         print("Masukkan nilai yang valid.")
20

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Code

+

-

📄

🗑️

⋮

```

PS C:\Users\asus\.ipython> python -u "c:\Users\asus\.ipython\profile_default\tes.py"
Masukkan tinggi: 5
Masukkan lebar: 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20

```

Untuk membuat program yang menampilkan deret angka dengan tinggi dan lebar tertentu:

Input: Tinggi (height) dan lebar (width) deret angka.

Output: Deret angka dengan dimensi yang ditentukan, dimulai dari 1, di mana setiap baris berisi angka berurutan sesuai lebar yang diinput, dan jumlah baris sesuai tinggi yang diinput.

Program ini meminta pengguna untuk memasukkan tinggi dan lebar, lalu mencetak deret sesuai dengan jumlah baris dan kolom yang dimasukkan. Setiap nilai di setiap baris akan terus bertambah, dimulai dari 1. Semoga ini sesuai dengan apa yang Anda perlukan.