



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231052
Nama Lengkap	GABRIEL SACHIO ATMADJAJA
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%)

Sumber Materi: Modul Pratikum Algoritma dan Pemrograman

MATERI 6

Percabangan kompleks, bisa juga disebut percabangan bertingkat adalah kondisi pemilihan dalam percabangan tidak hanya satu atau bisa terdiri dari banyak alternatif.

Berikut salah satu bentuk percabangan kompleks:

```
1  if kondisi1:
2      S
3  else:
4      if kondisi2:
5          S
6      else:
7          if kondisi3:
8              S
9          else:
10             if kondisi4:
11                 S
12             else:
13                 S
```

Gambar 1.1: Salah satu bentuk percabangan kompleks

Lalu ada juga perulangan kompleks disebut juga perulangan bertingkat yaitu perulangan yang di dalamnya terdapat perulangan lagi.

Berikut contoh perulangan bertingkat:

```
1  for i in range(m):
2      for j in range(n):
3          <lakukan perintah ini di inner>
4          <lakukan perintah itu di inner>
5      <lakukan perintah lain di outer>
6      <lakukan perintah lain lagi di outer>
```

Gambar 1.2:

Pada gambar 1.2, bagian for i disebut sebagai outer loop, sedangkan bagian for j disebut sebagai inner loop.

Hal yang perlu diketahui lainnya adalah di dalam perulangan bisa terdapat percabangan kompleks atau sebaliknya.

BAGIAN 2: LATIHAN MANDIRI (60%)

Source: <https://github.com/gabrielsachioa/PrakAIPro6.git>

SOAL 1

Kode:

```
latihan6-1.py > prima_terdekat
1  n = int(input("Masukkan n: "))
2
3  def prima_terdekat(n):
4      if n <= 1:
5          return "Bukan Prima"
6      elif n == 3:
7          return 2
8      else:
9          angka = 0
10         for i in range(n):
11             if i % 2 == 1:
12                 for j in range(2, i):
13                     if i % j == 0:
14                         break
15                     else:
16                         angka = i
17         return angka
18
19 print(prima_terdekat(n))
```

Gambar 2.1: Program Mencari Prima Terdekat dari inputan

Keterangan:

- Bilangan prima adalah bilangan yang habis dibagi 1 dan bilangan itu sendiri
- Baris 1 -> menangkap input dari user
- Baris 3 s.d 17 -> fungsi untuk mencari prima terdekat, dan fungsi ini belum dijalankan
- Baris 4,5 -> kondisi semisal user meng-inputkan n = 1, maka akan me-return "Bukan prima" karena angka 1 bukan bilangan prima
- Baris 6 -> kondisi semisal user meng-inputkan n = 3. Angka 3 sendiri termasuk bilangan prima, tapi programnya meminta bilangan prima terdekat dan bilangan prima terdekat dari angka 3 adalah angka 2
- Baris 8 s.d 17 -> kondisi selain dari 2 kondisi diatasnya. Buat variable angka untuk mereturn angkanya nanti, lalu buat perulangan hingga n, dan kondisi angka ganjil karena bilangan prima yang genap itu hanya angka 2, jadi sisanya adalah ganjil. Lalu buat perulangan lagi dgn variable j dimulai dari 2 hingga i. Kalau sisa bagi dari i dan j sama dengan 0, maka hentikan perulangan. Selain dari itu isi variable angka dengan i. Lalu return angka untuk mendapat bilangan prima terdekatnya
- Baris 19 -> memanggil fungsi

Output:

```
PS D:\Sachio\Tugas\prakalpro6> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
tihan6-1.py
Masukkan n: 12
11
```

Gambar 2.2: Output jika inputan-nya 12

```
PS D:\Sachio\Tugas\prakalpro6> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
tihan6-1.py
Masukkan n: 21
19
```

Gambar 2.3: Output jika inputan-nya 21

SOAL 2

Kode:

```
latihan6-2.py > perkalian_terbalik
1  n = int(input("Masukkan n: "))
2
3  def faktorial(n):
4      if n == 0:
5          return 1
6      else:
7          return n * faktorial(n - 1)
8
9  def perkalian_terbalik():
10     for i in range(n, 0, -1):
11         print(faktorial(i), end=" ")
12         for j in range(i, 0, -1):
13             print(j, end=" ")
14         i -= 1
15         print()
16
17  perkalian_terbalik()
```

Gambar 2.4: Program menampilkan deret

Keterangan:

- Baris 1 -> meminta dan menangkap input bilangan dari user
- Baris 3 s.d 7 -> fungsi faktorial untuk menghitung perkalian atau angka yang paling kiri nantinya. Intinya rumusnya adalah $n * \text{faktorial}(n - 1)$. Fungsi ini akan memanggil dirinya sendiri
- Baris 9 s.d 15 -> fungsi untuk menampilkan deretnya. Polanya adalah angka paling kiri itu faktorial dari n, lalu angka selanjutnya itu adalah angka-angka yang mengalikannya. Cara buatnya adalah dengan `for i in range(n, 0, -1)` untuk menampilkan faktorial ke bawah. Lalu `print(faktorial(i), end=" ")` untuk menampilkan faktorial sebanyak i. Lalu `for j` untuk menampilkan

deretnya kesamping. Lalu pastikan juga terdapat `i -= 1` supaya perulangannya berjalan dengan baik. Lalu `print()` untuk baris baru

- Baris 17 -> memanggil fungsi untuk menampilkan deretnya

Output:

```
PS D:\Sachio\Tugas\prakalpro6> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/tihan6-2.py
Masukkan 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
```

Gambar 2.5: Output

SOAL 3

Kode:

```
latihan6-3.py > ...
1  tinggi = int(input("Tinggi: "))
2  lebar = int(input("Lebar: "))
3
4  def deret_dinamis(tinggi, lebar):
5      if tinggi <= 0 or lebar <= 0:
6          print("Tidak valid")
7      else:
8          counter = 0
9          for i in range(1, tinggi + 1):
10             for j in range(1, lebar + 1):
11                 counter += 1
12                 print(counter, "", end='')
13             print()
14
15  deret_dinamis(tinggi, lebar)
```

Gambar 2.6: Program menampilkan deret

Keterangan:

- Baris 1, 2 -> meminta dan menangkap input dari user
- Baris 4 s.d 13 -> fungsi untuk menampilkan deretnya, parameternya dari input-an user.
- Baris 5, 6 -> kondisi semisal inputnya lebih kecil sama dengan dari 0, maka akan menampilkan "Tidak Valid"
- Baris 7 s.d 13 -> kondisi selain dari itu, buat variable counter dengan nilai 0. Lalu buat for i untuk menampilkan barisnya. For j untuk menampilkan deretnya, tapi pastikan counternya dihitung terlebih dahulu baru di print. Lalu `print()` untuk membuat baris barunya
- Baris 15 -> memanggil fungsi `deret_dinamis`

Output:

```
PS D:\Sachio\Tugas\prakalpro6> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:
ro6/latihan6-3.py
Tinggi: 5
Lebar: 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20
```

Gambar 2.7: Output