



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231050
Nama Lengkap	Samuel Natanael
Minggu ke / Materi	06 / Perulangan dan Percabangan 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.

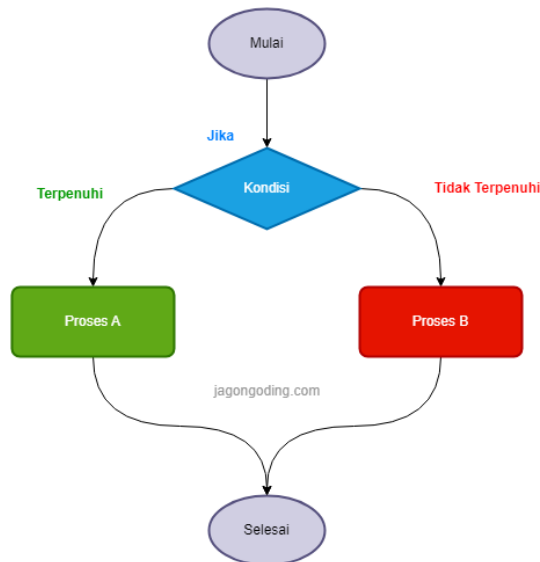
PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

BAGIAN 1: MATERI MINGGU INI (40%)

Struktur Percabangan Kompleks

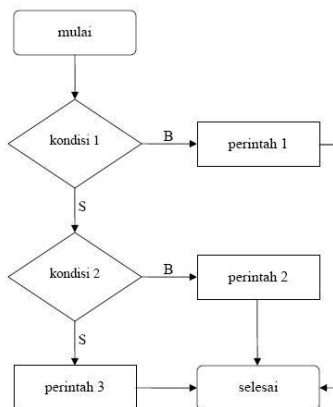
Percabangan adalah proses penentuan keputusan, bisa juga disebut dengan istilah conditional statement dalam bahasa inggris.

Sederhananya, kita memberikan perintah kepada komputer untuk melakukan suatu aksi tertentu (X) hanya jika kondisi Y terpenuhi, atau kita memberikan perintah untuk melakukan suatu aksi tertentu (A) hanya jika kondisi tidak terpenuhi.



Gambar 6.1 Flowchart Percabangan Sederhana (diambil dari <https://jagongoding.com/python/dasar/percabangan/>)

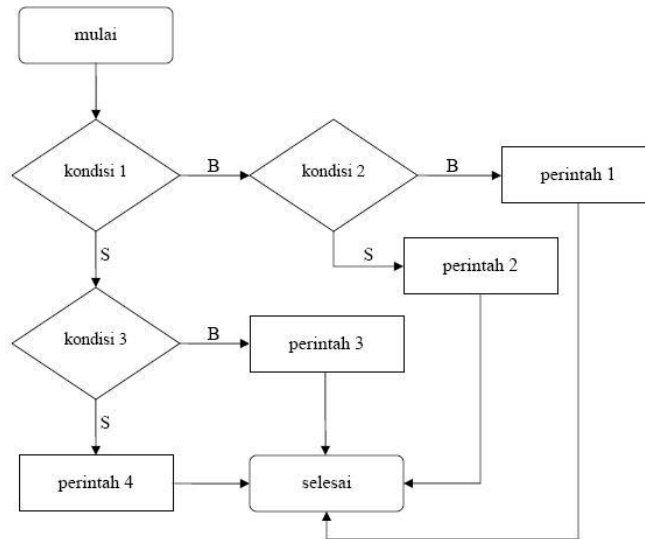
Percabangan juga bisa diperluas lagi menjadi percabangan kompleks di mana kondisi pemilihan dapat terdiri dari banyak pilihan. Selain itu, perintah dapat dibuat lebih dari satu.



Gambar 6.2 Flowchart Percabangan Kompleks 2 kondisi (diambil dari <https://ndoware.com/flowchart-percabangan.html>)

Bentuk code:

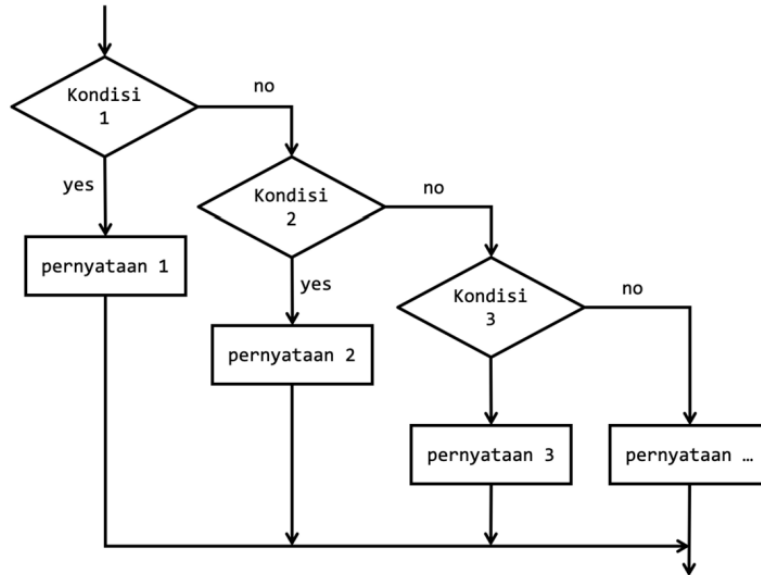
```
if {kondisi1}:
    {statements}
else:
    if{kondisi2}:
        {statements}
    else:
        {statements}
```



Gambar 6.3 Flowchart Percabangan Kompleks 3 kondisi (diambil dari <https://ndoware.com/flowchart-percabangan.html>)

Bentuk code:

```
if {kondisi1}:
    if {kondisi2}:
        {statements}
    else:
        {statements}
else:
    if{kondisi3}:
        {statements}
    else:
        {statements}
```



Gambar 6.3 Flowchart Percabangan Kompleks 3 kondisi Bentuk Kedua (diambil dari <https://ftmm.unair.ac.id/wp-content/uploads/filr/4747/6.%20Struktur%20Percabangan.pdf>)

Bentuk code:

```

if {kondisi1}:
    {statements}
else:
    if {kondisi2}:
        {statements}
    else:
        if {kondisi3}:
            {statements}
        else:
            {statements}
  
```

Percabangan Kompleks bersifat sangat fleksible, sehingga memiliki banyak bentuk yang berbeda tergantung kondisi.

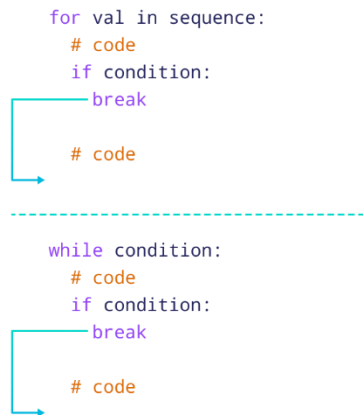
IF bertingkat memiliki manfaat khusus dalam beberapa situasi. Dalam IF biasa, semua kondisi IF harus dicoba satu per satu, bahkan mungkin hanya satu perintah IF yang terpenuhi; namun, IF bertingkat membuat eksekusi perintah lebih efisien dan lebih cepat. Ini pasti mempersingkat proses. Konversi nilai angka menjadi nilai huruf dengan batasan nilai yang sudah ditentukan adalah contoh kasus yang cocok untuk IF bertingkat.

Struktur Perulangan Kompleks

Dalam pemrograman, “break” dan “continue” statements digunakan untuk mengubah jalur perulangan.

- Break, keluar dari loop
- Continue, melewati pengulangan tersebut dan lanjut ke perulangan selanjutnya

Break



Gambar 6.4 Working of Python break Statement (diambil dari <https://www.programiz.com/python-programming/break-continue>)

Contoh:

```
for i in range(5):
    if i == 3:
        break
    print(i)
```

Output:

0

1

2

Continue

```
for i in range(5):
    if i == 3:
        continue
    print(i)
```

Output:

0
1
2
4

```
for val in sequence:
    # code
    if condition:
        continue

    # code

while condition:
    # code
    if condition:
        continue

    # code
```

Gambar 6.5 Working of continue Statement in Python (diambil dari <https://www.programiz.com/python-programming/break-continue>)

Perulangan Bertingkat

Perulangan kompleks adalah jenis per-ulangan di mana ada perulangan di dalam perulangan, yang menghasilkan waktu proses yang lebih lama. Perulangan kompleks digunakan dalam banyak algoritma untuk menyelesaikan beberapa masalah tertentu, contohnya adalah masalah matriks yang menggunakan array 2 dimensi, seperti catur, chekcers, dan masih banyak lagi. Pada dasarnya, masalah yang dapat diselesaikan melalui perulangan kompleks biasanya memiliki pola grid, atau kotak, yang memiliki lebar dan panjang yang berbeda.

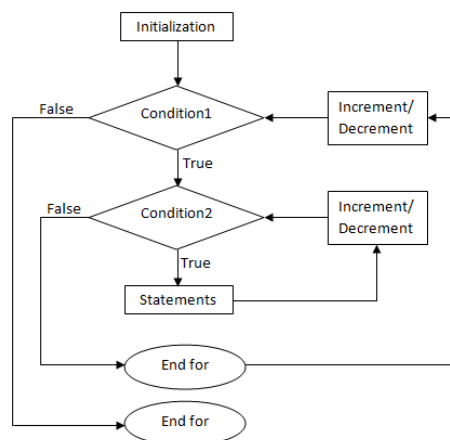


Fig: Flowchart for nested for loop

Gambar 6.6 Flowchart Perulangan Kompleks (diambil dari <https://stackoverflow.com/questions/43697634/showing-nested-for-loops-in-a-flowchart>)

Perulangan kompleks menggunakan while:

```
#statement(s)
while condition_1 :
    #statement(s)
    while condition_2 :
        #statement(s)
```

Perulangan kompleks menggunakan for:

```
for iterating_var in sequence:
    statement-1
    for iterating_var in sequence:
        statement-2
    statement-3
statement-4
```

Contoh penggunaan perulangan kompleks while:

```
i = 1
while i <= 4 :
    j = 0
    while j <= 3 :
        print(i*j, end=" ")
        j += 1
    print()
    i += 1
```

output:

0 1 2 3

0 2 4 6

0 3 6 9

0 4 8 12

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

```
1  # Input batas
2  batas = int(input("Masukan batas: "))
3
4  # Fungsi
5  def ganjil_prima_tertinggi(batas):
6      # Memulai Perulangan for dengan range batas dari besar ke kecil
7      for i in range(batas-1, 2, -1):
8          # j = 2 karena bilangan prima semua bisa dibagi 1
9          j = 2
10         # Perulangan untuk membagi i dengan j satu persatu untuk mencari bilangan prima
11         while i >= j:
12             # Jika bilangan bilangan j tidak bisa membagi i maka artinya i bilangan prima
13             if i == j:
14                 return i
15             else:
16                 # Jika bisa dibagi dengan j maka break karena bukan prima
17                 if i % j == 0:
18                     break
19                 # Jika tidak bisa dibagi dengan j maka lanjut ke bilangan j + 1
20                 # Dan seterusnya sampai mendapatkan bilangan prima
21                 else:
22                     j += 1
23                 continue
24
25  print(ganjil_prima_tertinggi(batas))
```

```
Masukan batas: 12
11
PS D:\Ngampus\Semester
Masukan batas: 21
19
PS D:\Ngampus\Semester
Masukan batas: 100
97
PS D:\Ngampus\Semester
Masukan batas: 200
199
```


SOAL 2

```
6,2.py > ...
1  # Input n
2  n = int(input("n = "))
3
4  # Fungsi untuk mencari faktorial
5  def factorial(n):
6      x = 1
7      # Factorial n didapatkan dengan perkalian semua bilangan dari 1 sampai n
8      for i in range(n, 0, -1):
9          x = x*i
10     return x
11
12     # Perulangan untuk menuliskan factorial dan baris bilangan
13     for i in range(n, 0, -1):
14         # Factorial diprint lebih dahulu lalu disambung dengan baris bilangan
15         print(factorial(i), end=' ')
16         # Perulangan untuk menghasilkan baris bilangan
17         for j in range(i, 0, -1):
18             print(j, end=' ')
19         # Jika perulangan berakhir maka akan lanjut perulangan berikutnya
20         print(" ")
```

```
n = 10
3628800 10 9 8 7 6 5 4 3 2 1
362880 9 8 7 6 5 4 3 2 1
40320 8 7 6 5 4 3 2 1
5040 7 6 5 4 3 2 1
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
PS D:\Ngampus\Semester 2\PrakAlPro>
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
```

SOAL 3

```
1  # Input tinggi dan lebar
2  tinggi = int(input("Masukan tinggi : "))
3  lebar = int(input("Masukan lebar : "))
4
5  # x dan y berawal dari 1
6  x = 1
7  y = 1
8
9  # Perulangan untuk hanya print sampai tinggi saja
10 while y <= tinggi:
11     # Perulangan untuk hanya print sampai lebar saja
12     for i in range(lebar):
13         # Kalau tidak melebihi lebar maka x akan di print
14         print(x, end=' ')
15         # x ditambah 1 tiap perulangan
16         x += 1
17     # Ketika sudah mencapai lebar, maka akan print di bawahnya
18     print("\n")
19     # y ditambah satu tiap kali perulangan lebar habis
20     y += 1
```

```
Masukan tinggi : 5
Masukan lebar : 4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20
PS D:\Ngampus\Semester 2\PrakAlPro>
Masukan tinggi : 10
Masukan lebar : 10
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100
```

Github : <https://github.com/SamuelN1508/PrakAlPro6.git>