Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**<71230989>**
Nama Lengkap	**<Yohanes Nevan Adventus Wibawa>**
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

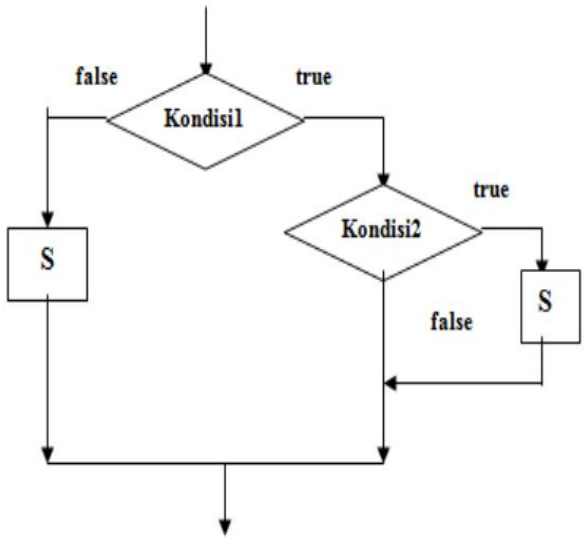
BAGIAN 1: MATERI MINGGU INI (40%)

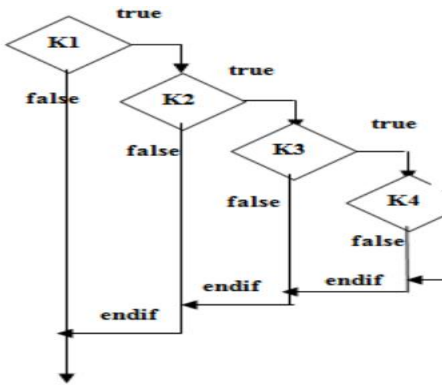
Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

Struktur Percabangan Kompleks

Percabangan kompleks bentuk 1 dibawah ini mengacu pada struktur percabangan dalam pemrograman di mana ada banyak alternatif kondisi pemilihan, dan setiap kondisi dapat memiliki lebih dari satu perintah yang dieksekusi. Dalam konteks ini, alur program akan bercabang ke berbagai jalur tergantung pada kondisi-kondisi yang terpenuhi, dan setiap jalur dapat mengeksekusi serangkaian perintah tertentu.

```
if kondisi1:
    if kondisi2:
        S
        S
    else:
        S
        S
else:
    S
    S
```





IF bertingkat adalah salah satu pendekatan untuk menangani situasi di mana terdapat banyak alternatif kondisi pemilihan. Pendekatan ini mengurangi kebutuhan untuk mengevaluasi semua kondisi secara berurutan, karena jika satu kondisi terpenuhi, cabang lain tidak akan dievaluasi. Contoh yang tepat adalah saat mengonversi nilai angka menjadi nilai huruf

Struktur Perulangan kompleks

```
for i in range(1000):
    print(i)
    if i == 10:
        break
```

Output :

```
TERMINAL

0
1
2
3
4
5
6
7
8
9
10
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 6>
```

Program tersebut akan mencetak angka dari 1 hingga 10 meskipun perulangan sudah ditetapkan dari 1 hingga 1000. Hal ini terjadi karena perintah "break" diberikan ketika nilai i adalah 10. Angka 10 tetap dicetak karena perintah untuk mencetak angka ditempatkan sebelum perintah "break".

```
for i in range(1000):
    if i == 10:
        break
    print(i)
```

Output:

```
"c:/Users/LENOVO/Tugas PrakAlPro/Tugas 6/coba.py"
0
1
2
3
4
5
6
7
8
9
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 6>
```

Program di atas akan menampilkan angka 1 hingga 9 meskipun perulangan telah diatur dari 1 hingga 1000. Hal ini disebabkan oleh perintah "break" yang diberikan ketika nilai i adalah 10. Angka 10 tidak akan ditampilkan karena perintah untuk mencetak angka diletakkan setelah perintah "break".

Countinue

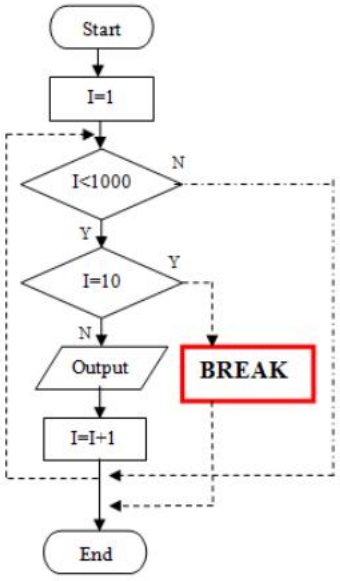
Perintah continue dalam pemrograman menyebabkan proses perulangan untuk melanjutkan ke iterasi berikutnya, mengabaikan statement-statement yang ada setelahnya dalam iterasi saat ini. Biasanya, perintah continue diimplementasikan dengan menggunakan pernyataan if untuk menentukan kapan harus menggunakannya. Berikut adalah contoh penggunaannya dalam struktur if

```
for i in range(10):
    if i == 5:
        continue
    print(i)
```

Output:

```
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 6> & "C:/Progr
"c:/Users/LENOVO/Tugas PrakAlPro/Tugas 6/coba.py"
0
1
2
3
4
6
7
8
9
```

Melewatkan angka 5



Perintah `continue` mengakibatkan proses perulangan melompat langsung ke iterasi berikutnya, mengabaikan pernyataan-pernyataan yang ada di bawahnya dalam iterasi saat ini. Sehingga, perintah-perintah yang ada setelah `continue` tidak akan dieksekusi dalam iterasi tersebut dan langsung dilanjutkan ke iterasi berikutnya.

Perulangan Bertingkat

Struktur perulangan kompleks merujuk pada bentuk perulangan di mana terdapat perulangan bersarang di dalam perulangan lainnya, sehingga terjadi perulangan bertingkat yang dapat menyebabkan peningkatan waktu proses. Ada banyak algoritma yang cocok untuk menggunakan struktur perulangan kompleks. Masalah yang bisa dipecahkan dengan perulangan kompleks meliputi masalah matriks yang menggunakan array 2 dimensi, masalah game board seperti catur dan minesweeper, serta masalah pengolahan citra digital seperti algoritma untuk mendeteksi tepi citra, mengubah citra berwarna menjadi grayscale, dan banyak lagi.

```
for i in range(m):
    <lakukan perintah ini>
    <lakukan perintah itu>
```

Sama seperti dalam percabangan kompleks, perulangan sendiri-sendiri juga dapat menjadi kompleks, seperti dalam kasus perulangan bersarang, misalnya dengan menggunakan dua perulangan for, satu di dalam yang lain. Jika kita ingin menyatukan perulangan kedua ke dalam perulangan pertama, maka kedua perulangan tersebut akan saling bersarang, yang berarti perulangan kedua akan dieksekusi pada setiap iterasi perulangan pertama.

```
for i in range(m):
    for j in range(n):
        <lakukan perintah ini di inner>
        <lakukan perintah itu di inner>
    <lakukan perintah lain di outer>
    <lakukan perintah lain lagi di outer>
```

```
n = int(input("masukan n = "))
for i in range(n, 0, -1):
    for j in range(i, 0, -1):
        print(i," ", end="")
    print()
```

Output:

```
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 6> & "
PrakAlPro/Tugas 6/coba.py"
masukan n = 5
5  5  5  5  5
4  4  4  4
3  3  3
2  2
1
```

Kegiatan Praktikum

Contoh 6.1

```
n=int (input("Masukkan n = "))
for i in range (1,n+1):
    for j in range (1,i+1):
        print(i," ",end='')
    print ()
```

Output:

```
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 6> & "C:/Prog
PrakAlPro/Tugas 6/kegiatan praktikum 1.py"
Masukkan n = 5
1
2  2
3  3  3
4  4  4  4
5  5  5  5  5
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 6>
```

Kode ini merupakan implementasi dari pola segitiga angka yang dijelaskan sebelumnya. Berikut adalah penjelasan langkah demi langkah: n=int(input("Masukkan n = ")): Baris ini meminta pengguna untuk memasukkan nilai n, yang akan menentukan jumlah baris dalam segitiga angka. for i in range(1, n+1):: Ini adalah perulangan luar (outer loop) yang akan berjalan dari 1 hingga n.

Setiap iterasi dari perulangan ini akan mengatur jumlah angka yang akan dicetak pada setiap baris segitiga. for j in range(1, i+1):: Ini adalah perulangan dalam (inner loop) yang akan berjalan dari 1 hingga nilai i pada setiap iterasi perulangan luar. Ini bertujuan untuk mencetak angka pada setiap baris, dengan jumlah angka yang dicetak sesuai dengan nilai i pada baris tersebut. print(i, " ", end=''):

Pada setiap iterasi perulangan dalam, angka i dicetak tanpa melakukan enter (ganti baris), dengan menggunakan parameter end=''. Ini berarti setiap angka akan dicetak di samping angka lainnya tanpa spasi tambahan. print(): Setelah selesai perulangan dalam, perintah ini mencetak enter (ganti baris), sehingga baris berikutnya akan dimulai pada baris baru. Dengan cara ini, pola segitiga angka akan terbentuk sesuai dengan nilai n yang dimasukkan pengguna. Pada setiap baris, angka akan dicetak sesuai dengan jumlah baris yang bersesuaian dengan nilai i pada saat itu.

Contoh 6.2

```
n = int(input("masukan n = "))
for i in range(1, n+1):
    if i%2 == 1:
        for j in range(1, n + 1):
            print(j," ", end="")
    else:
        for j in range(n, 0, -1):
            print(j," ", end='')
    print()
```

Output:

```
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 6> & "C
PrakAlPro/Tugas 6/kegiatan praktikum 6.2.py"
masukan n = 5
1  2  3  4  5
5  4  3  2  1
1  2  3  4  5
5  4  3  2  1
1  2  3  4  5
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 6>
```

Program tersebut menerima input dari pengguna dalam bentuk variabel n, yang menentukan jumlah baris dalam pola angka yang akan dicetak. Selanjutnya, dilakukan perulangan luar (outer loop) dengan rentang nilai dari 1 hingga n.

Pada setiap iterasi dari perulangan luar, dilakukan pengecekan apakah nilai i adalah ganjil atau genap. Jika i ganjil, maka dilakukan perulangan dalam (inner loop) dari 1 hingga n. Setiap angka dari 1 hingga n akan dicetak secara berurutan pada baris tersebut. Namun, jika i genap, perulangan dalam akan melakukan pencetakan dari n hingga 1 secara terbalik.

Setelah pencetakan untuk setiap baris selesai, dilakukan pemindahan ke baris berikutnya dengan mencetak enter (ganti baris). Proses ini berlanjut hingga mencapai baris ke-n sesuai dengan nilai yang dimasukkan pengguna. Dengan cara ini, pola angka yang berbeda akan terbentuk pada setiap baris, bergantung pada nilai i (ganjil atau genap) pada perulangan luar.

Contoh 6.3

```
n=int(input("Masukkan n = "))
for i in range(0,n+1):
    for j in range(1,n-i+1):
        print("X", end=' ') if j%2==1 else print("0",end=' ')
    print()
```

Output:

```
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 6> &
PrakAlPro/Tugas 6/kegiatan praktikum 6.3.py"
Masukkan n = 5
X 0 X 0 X
X 0 X 0
X 0 X
X 0
X
```

Program tersebut menerima input dari pengguna berupa nilai n yang menentukan tinggi segitiga terbalik. Setiap baris segitiga terbalik memiliki karakter "X" dan "0" yang dicetak secara bergantian, dengan jumlah karakter yang berkurang dari kiri ke kanan pada setiap baris. Proses pencetakan karakter dilakukan dengan menggunakan perulangan dalam (inner loop) yang mengiterasi dari 1 hingga n-i+1, di mana i adalah nomor baris yang sedang diproses. Selama pencetakan, dilakukan pengecekan apakah nomor iterasi ganjil atau genap, dan sesuai dengan itu, karakter "X" atau "0" dicetak. Setelah pencetakan selesai untuk setiap baris, dilakukan pemindahan ke baris berikutnya dengan mencetak enter (ganti baris). Proses ini berlanjut hingga mencapai baris ke-n sesuai dengan nilai yang dimasukkan pengguna.

Contoh 6.4

```
hasil = ""

x = int(input("Masukkan jumlah :"))
bar = x

while bar >= 0:

    kol = bar
    while kol > 0:
        hasil += "   "
        kol -= 1

    kanan = 1
    while kanan < (x - (bar-1)):
        hasil += " * "
        kanan += 1

    hasil = hasil + "\n"
    bar -= 1

print(hasil)
```

Output:

```
PrakAlPro/Tugas 6/kegiatan praktikum 6.4.py"
Masukkan jumlah :10
                        *
                      *  *
                    *  *  *
                  *  *  *  *
                *  *  *  *  *
              *  *  *  *  *  *
            *  *  *  *  *  *  *
          *  *  *  *  *  *  *  *
        *  *  *  *  *  *  *  *  *
      *  *  *  *  *  *  *  *  *  *
```

Program di atas menghasilkan pola segitiga siku-siku yang terdiri dari karakter bintang (*) dengan jumlah baris sesuai yang dimasukkan pengguna. Cara kerjanya adalah sebagai berikut: Program meminta pengguna untuk memasukkan jumlah baris pola segitiga. Selanjutnya, program menggunakan tiga buah perulangan while untuk mengatur pembentukan pola segitiga. Perulangan luar (outer loop) mengatur jumlah baris pola segitiga. Perulangan ini akan berjalan dari jumlah baris ke 0. Perulangan dalam pertama (inner loop) mengatur jumlah spasi pada setiap baris pola segitiga. Jumlah spasi yang ditambahkan akan berkurang seiring dengan penurunan jumlah baris. Perulangan dalam kedua (inner

loop) mengatur jumlah karakter bintang (*) yang akan ditampilkan pada sisi kanan segitiga. Jumlah bintang yang ditampilkan akan bertambah seiring dengan penurunan jumlah baris. Setelah selesai setiap baris, program akan mencetak hasil pembentukan pola segitiga ke layar. Dengan cara ini, program mencetak pola segitiga siku-siku dengan karakter bintang (*) sesuai dengan jumlah baris yang dimasukkan oleh 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 6.1

A. Source Code

```python
def det_prima(num):
    if num < 2:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True

def cari_prima(num):
    for i in range(num - 1, 1, -1):
        if det_prima(i):
            return i
    return None

n = int(input("Masukkan bilangan n: "))
prima_terdekat = cari_prima(n)
if prima_terdekat:
    print(f"Bilangan prima terdekat < {n} adalah {prima_terdekat}")
else:
    print(f"Tidak ditemukan bilangan prima terdekat < {n}")
```

B. Output:

C. Penjelasan

Pada awalnya, digunakan fungsi `prima(num)` untuk menerima sebuah nilai integer `num` dan memeriksa apakah `num` adalah bilangan prima atau bukan.

Fungsi ini melakukan iterasi dari angka 2 sampai akar kuadrat dari `num`, dengan tujuan memeriksa apakah `num` dapat dibagi oleh angka selain 1 dan dirinya sendiri. Jika dalam iterasi ditemukan bilangan yang memenuhi syarat tersebut, maka `num` bukanlah bilangan prima, dan fungsi mengembalikan `False`. Sementara itu, fungsi `cari_prima(num)` berperan dalam mengambil sebuah integer `num` sebagai input dan mengecek apakah itu merupakan bilangan prima. Fungsi ini melakukan iterasi dari `num - 1` hingga 2, mencari bilangan prima pertama yang ditemukan dengan memanggil fungsi `prima(i)`.

Jika sebuah bilangan prima ditemukan selama iterasi, fungsi akan langsung mengembalikan bilangan prima tersebut. Namun, jika tidak ditemukan bilangan prima dalam rentang tersebut, maka fungsi akan mengembalikan `None`. Di bawah definisi kedua fungsi tersebut, program meminta pengguna untuk memasukkan sebuah bilangan `n`. Setelah itu, program mencari bilangan prima terdekat yang lebih kecil dari `n` menggunakan fungsi `cari_prima(n)`, dan hasilnya dicetak. Jika bilangan prima terdekat ditemukan, program akan mencetak pesan yang menyatakan bahwa "Bilangan prima terdekat < n adalah prima_terdekat". Namun, jika tidak ditemukan bilangan prima terdekat, program akan mencetak pesan yang menyatakan bahwa "Tidak ditemukan bilangan prima terdekat < n".

SOAL 6.2

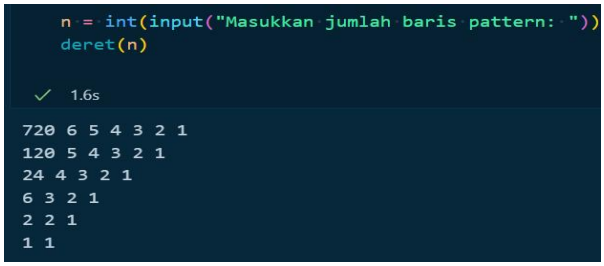
A. Source Code

```python
def deret(n):
    for i in range(n, 0, -1):
        print(str(faktorial(i)) + " ", end="")
        for j in range(i, 0, -1):
            print(str(j) + " ", end="")
        print()

def faktorial(num):
    if num == 1 or num == 0:
        return 1
    return num * faktorial(num - 1)

n = int(input("Masukkan jumlah baris pattern: "))
deret(n)
```

B. Output



C. Penjelasan

Program ini bertujuan untuk mencetak pola segitiga dengan baris-baris yang menurun, di mana setiap barisnya terdiri dari nilai faktorial dari nomor baris tersebut, diikuti dengan deret angka yang terus berkurang dari nomor baris tersebut hingga 1.

Fungsi `deret(n)` melakukan iterasi mundur dari nilai `n` hingga 1, dengan langkah mundur sebesar 1. Pada setiap iterasi, nilai faktorial dari nomor baris tersebut dicetak terlebih dahulu, diikuti dengan pencetakan deret angka yang terus berkurang dari nomor baris tersebut hingga 1. Pencetakan dilakukan dalam satu baris dengan menggunakan fungsi `print()` dengan argumen `end=""`, sehingga setiap angka atau nilai faktorial dipisahkan oleh spasi. Setelah satu baris selesai dicetak, baris baru ditambahkan dengan fungsi `print()` tanpa argumen.

Fungsi `faktorial(num)` digunakan untuk menghitung nilai faktorial dari suatu bilangan bulat `num`. Jika nilai `num` adalah 1 atau 0, maka fungsi mengembalikan nilai 1, karena faktorial dari 1 dan 0 adalah 1. Jika nilai `num` lebih besar dari 1, maka fungsi melakukan rekursi untuk mengalikan nilai `num` dengan faktorial dari `num-1`.

Setelah itu, program meminta pengguna untuk memasukkan jumlah baris pattern yang diinginkan menggunakan fungsi `input()`. Nilai yang dimasukkan oleh pengguna akan disimpan dalam variabel `n`.

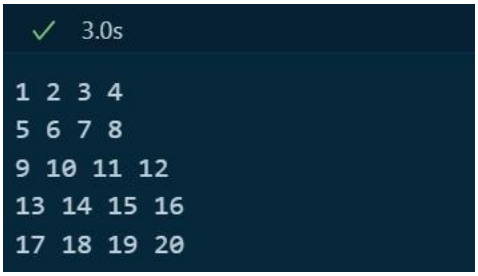
Selanjutnya, program memanggil fungsi `deret(n)` dengan menggunakan nilai `n` sebagai argumen. Fungsi ini akan mencetak pola segitiga sesuai dengan jumlah baris yang dimasukkan oleh pengguna, dengan setiap baris menampilkan nilai faktorial dari nomor baris tersebut, diikuti dengan deret angka yang terus berkurang hingga 1.

Soal 6.3

A. Source Code

```
tinggi = int(input("tinggi = "))
lebar = int(input("lebar = "))
angka = 1
for i in range(tinggi):
    for j in range(lebar):
        print(angka, end=" ")
        angka += 1
    print()
```

B. Output



C. Penjelasan

Program ini bertujuan untuk mencetak pola persegi panjang dengan tinggi dan lebar yang ditentukan oleh pengguna. Setiap sel dalam pola akan berisi angka berturut-turut, dimulai dari 1 hingga mencapai tinggi * lebar.

Program dimulai dengan meminta pengguna untuk memasukkan nilai tinggi dan lebar persegi panjang menggunakan fungsi `input()`. Nilai-nilai ini disimpan dalam variabel `tinggi` dan `lebar` secara berturut-turut.

Selanjutnya, program menggunakan dua loop bersarang untuk mencetak pola persegi panjang. Loop pertama (`for i in range(tinggi)`) digunakan untuk mengontrol tinggi dari pola, sedangkan loop kedua (`for j in range(lebar)`) digunakan untuk mengontrol lebarnya.

Pada setiap iterasi loop dalam loop pertama, program mencetak angka berturut-turut dimulai dari 1 menggunakan variabel `angka`. Setelah mencetak angka, nilai `angka` ditingkatkan dengan 1 agar pada iterasi berikutnya akan mencetak angka berikutnya.

Setelah mencetak angka sepanjang lebar persegi panjang, program memanggil fungsi `print()` tanpa argumen untuk mencetak baris baru sehingga pola akan terus berlanjut ke baris berikutnya.

Dengan demikian, program ini akan mencetak pola persegi panjang sesuai dengan tinggi dan lebar yang dimasukkan oleh pengguna, dengan setiap sel berisi angka berturut-turut dari 1 hingga tinggi * lebar.

Link Github

Link = https://github.com/YohanesNevan/Tugas-Laporan-AlPro-6.git