LAPORAN RESMI
MODUL III
LOOPING
ALGORITMA PEMROGRAMAN

NAMA	**: DINI AULIYA MULHA**
N.R.P	**: 24044100103**
DOSEN	**: FITRI DAMAYANTI, S.KOM., M.KOM.**
ASISTEN	**: AKBAR MAULANA HUSADA**
TGL PRAKTIKUM	**: 9 OKTOBER 2024**

Disetujui : 13 OKTOBER 2024
Asisten

AKBAR MAULANA HUSADA
23.04.411.00060

LABORATORIUM BISNIS INTELIJEN SISTEM
PRODI SISTEM INFORMASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJO MADURA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Looping (For dan While) adalah konsep penting dalam pemrograman yang memungkinkan eksekusi kode berulang kali. Dalam Python, loop digunakan untuk berbagai keperluan memahami cara menggunakan loop ini sangatlah penting bagi setiap programmer yang ingin mengembangkan keterampilan coding mereka. Artikel ini akan membahas cara kerja dan penggunaan loop For dan While dalam Python secara rinci.

Python menyediakan dua jenis loop utama: For dan While. Keduanya memiliki kegunaan yang berbeda dan dapat diaplikasikan pada situasi yang bervariasi. Dengan memahami perbedaan dan aplikasi praktis dari kedua jenis loop ini, kamu akan lebih mudah menulis kode yang efisien. Artikel ini akan memberikan panduan lengkap tentang cara mengimplementasikan loop For dan While dalam proyek Python kamu.

Menguasai loop di Python akan membantu kamu menyederhanakan tugas-tugas yang membutuhkan pengulangan. Baik itu iterasi melalui daftar, pengolahan data, atau bahkan pengembangan algoritma kompleks, loop adalah alat yang sangat berguna. Dalam artikel ini, kita akan mempelajari sintaks dasar dan contoh-contoh praktis penggunaan loop For dan While.

1.2 Tujuan

Tujuan looping dalam Python adalah untuk memungkinkan eksekusi berulang dari sekelompok pernyataan atau instruksi. Berikut adalah beberapa tujuan spesifik dari penggunaan looping:

1. Otomatisasi Tugas Berulang
2. Pengolahan Koleksi Data
3. Pengulangan Berdasarkan Kondisi
4. Pencarian dan Penyaringan Data
5. Interaksi dengan Pengguna
6. Implementasi Algoritma
7. Membuat Struktur Data Dinamis

BAB II
DASAR TEORI

2.1 Pengertian Perulangan

Perintah perulangan digunakan untuk mengulang pengeksekusikan statemen-statemen hingga berkali-kali sesuai dengan iterasi yang diinginkan. Dalam python, perintah untuk perulangan (loop) adalah while dan for.

2.2 Perintah While

Perintah while pada python merupakan perintah yang paling umum digunakan untuk proses iterasi. Konsep sederhana dari perintah while adalah ia akan mengulang mengeksekusi statemen dalam blok while selama nilai kondisinya benar. Dan ia akan keluar atau tidak melakukan eksekusi blok statemen jika nilai kondisinya salah.

Bentuk umum statemen *while*,

```
while (kondisi) :
      statemen
```

Pada contoh 1, merupakan contoh sederhana penggunaan while. Pada contoh di atas program akan terus mengeksekusi statemen dalam badan while, dikarenakan kondisinya selalu benar (true). Kondisi seperti ini disebut infinite loop.i

Contoh 1:

```
1  x = "Wuland"
2  while x:
3
4      print (x)
5      x = x[1:]
```

Contoh 2 :

```
1  a = 0; b = 10
2  while a < b :
3      print (a)
4      a = a + 1
```

2.3 Perintah For

Perintah for dalam python mempunyai ciri khas tersendiri dibandingkan dengan bahasa pemrograman lain. Tidak hanya mengulang bilangan-bilangan sebuah ekspresi aritmatik, atau memberikan keleluasaan dalam mendefinisikan iterasi perulangan dan menghentikan perulangan pada saat kondisi tertentu. Dalam python, statemen for bekerja mengulang berbagai macam tipe data sekuensial seperti List,String, dan Tuple.

Bentuk umum perintah *for*,

```
for (variabel) in (objek) :
     statemen
else:
     statemen
```

Contoh sederhana menggunakan perintah for

```
1    for i in [5, 4, 3, 2, 1]:
2         print (i)
```

Pada contoh 2, perintah perulangan terjadi dimana data-data untuk iterasi (objek) berada dalam List. Jadi elemen-elemen yang berada dalam List akan di masukkan (assign) ke dalam variabel target yaitu i.

Contoh 2 :

```
1    T = [(1,2), (3,4), (5,6)]
2    for (a,b) in T :
3         print (a,b)
```

Pada contoh 3, merupakan penggunaan tipe data Tuple untuk proses perulangan. Elemen pada tuple akan di assign kedalam variabel a dan b.

```
1    T = [(1,2), (3,4), (5,6)]
2    for (a,b) in T :
3         print (a,b)
```

2.4 Perintah Break, Continue, dan Pass

1. Perintah Break

Perintah break digunakan untuk menghentikan jalannya proses iterasi padastatemen for atau while. Statemen yang berada di bawah break tidak akan di eksekusi dan program akan keluar dari proses looping.

Contoh 1:

```
x = 4
while x < 5:
    if x == 3:
        break
    print (x)
    x = x+1
else:
    print ("Loop sdh selesai dikerjakan" )
```

2. Perintah Continue

Statemen continue menyebabkan alur program kembali ke perintah looping. Jadi jika dalam sebuah perulangan terdapat statemen continue, maka program akan kembali ke perintah looping untuk iterasi selanjutnya.

Contoh 2 :

```
n = 10
while n:
    n = n - 1
    if n % 2 != 0:
            continue
    print (n)
```

3. Perintah Pass

Statemen pass mengakibatkan program tidak melakukan tindakan apa-apa. Perintah pass biasanya digunakan untuk mengabaikan suatu blok statemen perulangan, pengkondisian, class, dan fungsi yang belum didefinisikan badan programnya agar tidak terjadi error ketika proses compilasi.

Contoh 3 :

```
#program tidak akan melakukan
# #proses looping
# while True : pass
while True : pass
```

BAB III
TUGAS PENDAHULUAN

3.1 Soal

1. Jelaskan pengertian perulangan dalam python menggunakan bahasa sendiri!
2. Sebutkan apa saja perbedaan For loop dan While loop
3. Sebutkan apa saja perbedaan dari perintah Break, Continue, dan pass!
4. Jelaskan cara pengimplementasian perintah Break, Continue, dan pass!
5. Tulislah program singkat dengan menggunakan salah satu perintah Break, Continue, dan pass!

3.2 Jawaban

1. Perulangan dalam python adalah proses menyalankan blok kode atau satu bagian kode berkali-kali secara otomatis.
2. For Loop : digunakan untuk mengulangi sesuatu sesuai jumlah atau daftar yang sudah kita tentukan sebelumnya.
   While Loop : digunakan untuk mengulangi kode selama kondisi yang ditentukan masih benar.
3. Break : menghentikan seluruh Loop, keluar dari Loop sepenuhnya
   Continue : melewati iterasi saat ini dan lanjut ke iterasi berikutnya.
   Pass : tidak melakukan apa-apa, hanya placeholder ketika blok kode diperlukan tapi belum diimplementasikan
4. Dalam python Break memungkinkan untuk keluar dari loop saat kondisi eksternal dipicu Anda akan meletakkan break pernyataan tersebut didalam blok kode dibawah pernyataan Loop, biasanya setelah If pernyataan kondisional.
   Continue memungkinkan untuk melewati bagian loop tempat kondisi eksternal dipicu, tetapi melanjutkan untuk menyelesaikan sisa loop. iterasi loop saat ini akan terganggu, tetapi program akan kembali keawal Loop saat kondisi eksternal dipicu, pass pernyataan tersebut memungkinkan anda menangani kondisi tersebut tanpa loop terpengaruh dengan cara apapun, semua kode akan terus dibaca kecuali break pernyataan tertentu terjadi.
5. • Break

```
for i in range(10):
    if i == 5:
        break
    Print(i)
```

* perulangan akan berjalan dari i = 0 hingga i = 9
* ketika i sama dengan 5, kondisi ini terpenuhi
* perintah Break menghentikan perulangan
* mencetak nilai sebelum perulangan dilanjutkan

BAB IV

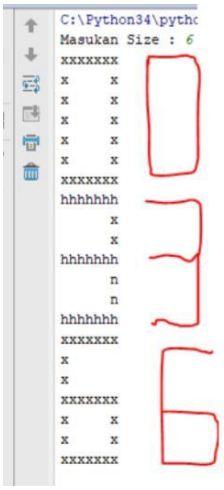
IMPLEMENTASI

4.1 Tugas Praktikum

4.1.1 Tugas Praktikum No. 1

Buatlah program dengan bentuk angka NIM terakhir kalian, misalnya 036

Outputnya



4.1.2 Tugas Praktikum No. 2

Sudarso ingin membuat program untuk membalikkan urutan dari sebuah angka bulat yang dimasukkan secara dinamis oleh pengguna. Bantulah ia dalam membuat program tersebut !

4.1.3 Tugas Praktikum No. 3

Juminten memiliki sebuah toko penyewaan DVD dengan syarat bahwa setiap DVD hanya boleh dipinjam selama 5 hari. Apabila DVD tidak dikembalikan sesuai waktu yang diberikan, penyewa akan dikenai denda sebesar Rp.2500/hari. Selain itu apabila keterlambatan si penyewa mencapai lebih dari 10 hari, maka penyewa akan dikenai denda tambahan Rp5500 untuk setiap 5 hari keterlambatan. Bantulah Juminten dengan membuat program untuk menghitung total denda keterlambatan berdasarkan lama waktu penyewaan dan saat perhitungan sudah selesai, program akan menanyakan apakah user ingin menghitung kembali atau tidak !

4.1 Source Code

4.2.1 Tugas Praktikum Soal No. 1

```
NIM = 103
size = int(input("Masukkan ukuran pola: "))
for i in range(size):
    if i == 0:
        print(" " * (size - 1) + "1")
    else:
        print(" " * (size - 1) + "1")
print()
for i in range(size):
    if i == 0 or i == size - 1:
        print("0" * size)
    else:
        print("0" + " " * (size - 2) + "0")
print()
for i in range(size):
    if i == 0 or i == size // 2 or i == size - 1:
        print("3" * size)
    else:
        print(" " * (size - 1) + "3")
```

4.2.2 Tugas Praktikum Soal No. 2

```
# Meminta input angka dari pengguna
angka = input("Masukkan angka bulat: ")
# Membalik urutan angka menggunakan looping
angka_terbalik = "" i = len(angka) - 1
while i >= 0:
    angka_terbalik += angka[i]
    i -= 1
```

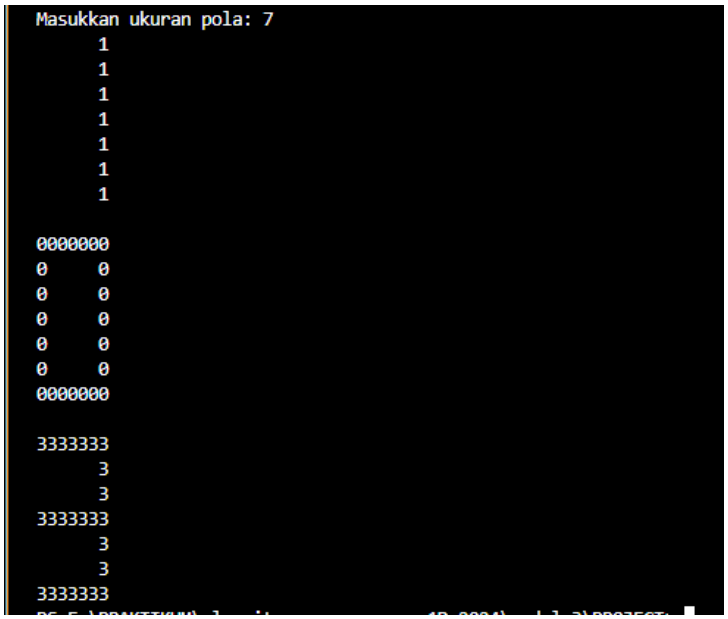
```
# Menampilkan hasil
print("Angka setelah dibalik: ",angka_terbalik)
```

4.2.3 Tugas Praktikum Soal No. 3

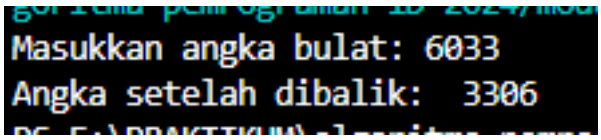
```
while True:
    hari_pinjam = int(input("masukkan lama waktu penyewaan: "))
    batas_pinjam = 5


    denda_per_hari = 2500
    denda_tambahan = 5500
    if hari_pinjam <= batas_pinjam:
        print("kamu tidak mendapat denda")
    else:
        keterlambatan = hari_pinjam - batas_pinjam
        total_denda = keterlambatan * denda_per_hari
        if hari_pinjam > 10:
            tambahan_hari = keterlambatan - 10
            jumlah_tambahan = tambahan_hari // 5
            total_denda += jumlah_tambahan * denda_tambahan
            print("total denda: ", total_denda)
        elif hari_pinjam > batas_pinjam <10:
            print("total denda: ", total_denda)
            while True:
                if hari_pinjam < 10:
                    print("itu saja denda nya")
                    break
    lagi = input("apa kamu ingin menghitung denda lagi?
(ya/tidak: )")
    if lagi != "ya":
        break
```

4.3 Hasil

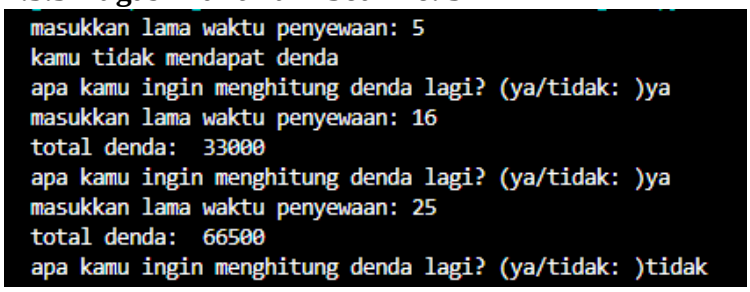
4.3.1 Tugas Praktikum Soal No. 1



4.3.2 Tugas Praktikum Soal No. 2



4.3.3 Tugas Praktikum Soal No. 3



4.4 Penjelasan

4.4.1 Tugas Praktikum Soal No. 1

Pada soal no. 1 diminta untuk membuat sebuah kode Python yang digunakan untuk mencetak pola karakter berdasarkan ukuran yang dimasukkan oleh pengguna. Program ini mencetak pola untuk tiga digit berbeda ('1', '0', dan '3'), dengan aturan yang spesifik untuk masing-masing digit.

Pada awal program, pengguna diminta untuk memasukkan ukuran pola, yang kemudian disimpan dalam variabel size. Variabel ini menentukan ukuran pola yang akan dicetak.

Untuk digit pertama ('1'), program menggunakan loop untuk mencetak sejumlah baris sebanyak size. Setiap baris berisi angka '1' yang ditempatkan di bagian paling kanan, dengan spasi sebanyak size - 1 di depannya. Hasilnya adalah pola vertikal dengan angka '1' yang selalu berada di tepi kanan.

Setelah itu, program mencetak baris kosong untuk memisahkan pola digit pertama dari pola digit berikutnya.Pola untuk digit kedua ('0') dihasilkan dengan cara yang sedikit berbeda. Jika berada di baris pertama atau baris terakhir, angka '0' dicetak penuh sepanjang size. Namun, untuk baris-baris di tengah, hanya angka '0' pertama dan terakhir yang dicetak, dengan ruang kosong di antaranya. Ini menghasilkan pola seperti bingkai persegi panjang yang menggambarkan angka '0'.

Lalu, baris kosong dicetak untuk memisahkan pola angka berikutnya dan untuk digit ketiga ('3') memiliki pola yang mencetak angka '3' secara penuh pada baris pertama, baris tengah, dan baris terakhir. Untuk baris lainnya, angka '3' ditempatkan di tepi kanan, dengan spasi kosong di depannya. Pola ini menggambarkan bentuk angka '3' secara vertikal, dengan bagian atas, tengah, dan bawah yang tebal.

Secara keseluruhan, program ini dirancang untuk menghasilkan pola yang menyerupai tampilan angka '1', '0', dan '3' sesuai dengan ukuran yang diinginkan, memberikan gambaran visual yang berbeda untuk masing-masing digit.

4.4.2 Tugas Praktikum Soal No. 2

Pertama, program meminta input dari pengguna dan menyimpannya dalam variabel bernama angka. Selanjutnya, variabel angka_terbalik digunakan untuk menyimpan angka yang akan dihasilkan setelah urutannya dibalik.

Untuk membalik urutan angka, program menggunakan loop while yang dimulai dari indeks terakhir angka (dengan i = len(angka) - 1) dan terus berjalan ke belakang hingga mencapai indeks pertama. Setiap karakter dari angka diambil berdasarkan indeks i dan ditambahkan ke variabel angka_terbalik. Setelah itu, i dikurangi satu setiap kali iterasi agar program bisa melanjutkan ke karakter sebelumnya.

Setelah semua karakter dari angka awal ditambahkan ke angka_terbalik dalam urutan terbalik, program akan menampilkan hasilnya. Misalnya, jika pengguna memasukkan "6033", maka hasil yang akan ditampilkan adalah "3306".

4.4.3 Tugas Praktikum Soal No. 3

Untuk soal no 3 ini berfungsi untuk menghitung denda keterlambatan penyewaan berdasarkan jumlah hari yang terlambat. Pertama-tama, program menggunakan loop while True yang berarti program akan terus berjalan hingga pengguna memilih untuk berhenti. Pengguna diminta untuk memasukkan jumlah hari penyewaan, yang disimpan dalam variabel hari_pinjam.

Program membandingkan nilai hari_pinjam dengan batas_pinjam, yang ditetapkan sebagai 5 hari. Jika hari_pinjam kurang dari atau sama dengan 5, maka tidak ada denda yang harus dibayar dan program menampilkan pesan "kamu tidak mendapat denda". Namun, jika hari_pinjam lebih dari 5, program menghitung jumlah keterlambatan dengan mengurangi batas_pinjam dari hari_pinjam, dan total denda dihitung dengan mengalikan jumlah hari keterlambatan dengan denda_per_hari, yang bernilai 2500.

Selanjutnya, jika jumlah hari yang terlambat melebihi 10, maka ada tambahan denda untuk setiap 5 hari di atas 10 hari. Program menghitung tambahan hari dengan mengurangi 10 dari hari_pinjam, lalu menghitung jumlah kelompok 5 hari tambahan. Setiap kelompok 5 hari ini dikenakan denda tambahan sebesar 5500, dan denda tersebut ditambahkan ke total denda. Setelah itu, program menampilkan total denda yang harus dibayar.

Jika hari_pinjam lebih dari 5 tapi kurang dari 10, program hanya menampilkan total denda tanpa tambahan. Ada pengecekan tambahan di dalam loop

lain yang memeriksa apakah hari_pinjam kurang dari 10. Jika benar, program menampilkan "itu saja dendanya" dan keluar dari loop.

Setelah setiap perhitungan denda selesai, program menanyakan apakah pengguna ingin menghitung denda lagi. Jika pengguna menjawab "ya", program mengulang dari awal. Jika pengguna menjawab selain "ya", program akan berhenti dan keluar dari loop utama.

BAB V
PENUTUP

4.1 Analisa

Menurut praktikaan, fungsi Looping atau perulangan ini adalah efisiensi kode, karena memungkinkan pengurangan jumlah kode yang ditulis. Looping sangat berguna dalam pengolahan data, seperti menghitung total nilai dalam array atau daftar dengan cara yang lebih sederhana dan cepat serta memfasilitasi otomatisasi tugas-tugas yang berulang, seperti pemrosesan file atau pengambilan data, yang meningkatkan produktivitas. Dalam pengembangan algoritma

Di dunia nyata, banyak kegiatan yang melibatkan pengolahan informasi berulang, seperti menghitung statistik dari survei. Di media sosial, algoritma untuk menampilkan konten sering menggunakan looping untuk memperbarui dan menyajikan informasi kepada pengguna.

Secara keseluruhan, looping merupakan alat yang sangat berguna baik dalam pemrograman maupun dalam kehidupan sehari-hari. Dalam pemrograman, ia meningkatkan efisiensi dan mempermudah pengolahan data yang kompleks. Sementara itu, dalam kehidupan sosial, konsep ini diterapkan dalam analisis data dan pengambilan keputusan, menjadikannya penting bagi programmer dan ilmuwan data.

4.2 Kesimpulan

Kesimpulan dari analisis ini menunjukkan bahwa looping adalah komponen penting dalam pemrograman yang meningkatkan efisiensi dan mempermudah pengolahan data. Dengan kemampuannya untuk mengurangi redundansi dalam kode, looping memungkinkan programmer untuk menulis kode yang lebih bersih dan lebih mudah dipahami. Dalam konteks kehidupan sosial, manfaat looping terlihat dalam berbagai aplikasi, mulai dari analisis data hingga algoritma media sosial yang menyajikan informasi. Secara keseluruhan, pemahaman yang mendalam tentang looping sangat berharga bagi para programmer dan ilmuwan data, karena ia mendukung otomatisasi dan pemrosesan informasi yang lebih efektif.