

LAPORAN PRAKTIKUM

DAFTAR / LIST

DOSEN PENGAMPU Dian Septiani Santoso M. Kom.



WAHYU IKBAL MAULANA - 3323600056

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

PRODI SAINS DATA TERAPAN

SEPTEMBER 2023

Latihan 1

```
lst = [1,2,3,4,5]
lst.insert(1,6)
del lst[0]
lst.append(1)

print(lst)

[6, 2, 3, 4, 5, 1]
```

list lst berisi lima elemen: 1, 2, 3, 4, dan 5.

function insert Menambahkan elemen 6 ke dalam indeks ke-1 dalam list lst.

Menghapus elemen pertama dalam list lst menggunakan del, lalu Menambahkan elemen 1 ke akhir list lst

Lalu cetak list lst.

```
lst = [1, [2,3], 4]
print(lst[1])
print(len(lst))

[2, 3]
3
```

Baris 1: Mendefinisikan list lst dengan tiga elemen: 1, [2,3], dan 4.

Baris 2: Mencetak elemen kedua dari list lst.

Baris 3: Mencetak panjang list lst.

```
lst = []
del lst
print(lst)

[]
```

Kode tersebut error karena list lst telah dihapus pada baris kedua. Ketika kita mencoba untuk mencetak list lst pada baris ketiga, list tersebut sudah tidak ada lagi.

Untuk memperbaiki kode tersebut seharusnya del dihapus atau diletakkan di akhir sehingga outputnya berupa lst

```
lst = [1,2,3,4,5]
lst_2 = []
add = 0
```

```
for number in lst:
    add += number
    lst_2.append(add)

print(lst_2)

[1, 3, 6, 10, 15]
```

list lst yang berisi elemen 1, 2, 3, 4, dan 5.

lalu lst_2 berisi array kosong

add = 0: adalah perintah untuk membuat variabel add yang berisi nilai 0.

lalu menggunakan perulangan untuk mengulangi proses berikut untuk setiap elemen list lst: Menambahkan elemen list lst saat ini ke variabel add. Menambahkan nilai variabel add ke list lst_2. print(lst_2): adalah perintah untuk mencetak list lst_2 ke konsol.

B. Latihan 2

```
# Definisikan list
list_buah = ["Pisang", "Nanas", "Melon", "Durian"]

# Tambahkan elemen ke list menggunakan metode append
list_buah.append("Anggur")
print("List buah setelah append:", list_buah)

# Tambahkan beberapa elemen ke list menggunakan metode extend
list_buah.extend(["Mangga", "Jeruk"])
print("List buah setelah extend:", list_buah)

# Masukkan elemen ke list pada indeks tertentu menggunakan metode insert
list_buah.insert(2, "Apel")
print("List buah setelah insert:", list_buah)

# Hapus elemen dari list menggunakan metode remove
list_buah.remove("Melon")
print("List buah setelah remove:", list_buah)

List buah setelah append: ['Pisang', 'Nanas', 'Melon', 'Durian', 'Anggur']
List buah setelah extend: ['Pisang', 'Nanas', 'Melon', 'Durian', 'Anggur', 'Mangga', 'Jeruk']
List buah setelah insert: ['Pisang', 'Nanas', 'Apel', 'Melon', 'Durian', 'Anggur', 'Mangga', 'Jeruk']
List buah setelah remove: ['Pisang', 'Nanas', 'Apel', 'Durian', 'Anggur', 'Mangga', 'Jeruk']
```

Berikut adalah penjelasan dari kode di atas

Fungsi	Keterangan	Contoh
<code>append()</code>	Menambahkan elemen baru ke akhir list	<code>list.append(elemen)</code>
<code>extend()</code>	Menambahkan elemen-elemen dari list lain ke akhir list	<code>list.extend(list_lain)</code>
<code>insert()</code>	Menambahkan elemen baru ke list pada indeks tertentu	<code>list.insert(indeks, elemen)</code>
<code>remove()</code>	Menghapus elemen dari list	<code>list.remove(elemen)</code>

C. Latihan 3

```
# Deklarasikan list untuk menyimpan input pengguna
list_angka = []

# Pilih mode pengurutan
while True:
    mode = int(input("Pilih mode pengurutan (1: Ascending atau 2: Descending): "))
    if mode == 1 or mode == 2:
        break
    else:
        print("pilih 1 atau 2")
        continue

# Masukkan bilangan bulat
while True:
    bil_bulat = int(input("Masukkan bilangan bulat:"))
    list_angka.append(bil_bulat)

    lanjut = input("Masih ada bilangan yang dimasukkan (y/t)")
    if lanjut != "y":
        break

# Urutkan input_list sesuai mode yang dipilih
if mode == 1:
    # Selection Sort
    for i in range(len(list_angka)):
        min_idx = i
        for j in range(i + 1, len(list_angka)):
            if list_angka[min_idx] > list_angka[j]:
                min_idx = j

    # Swap elemen minimum dengan elemen pertama
```

```

        list_angka[i], list_angka[min_idx] = list_angka[min_idx],
list_angka[i]
elif mode == 2:
    # Selection Sort
    for i in range(len(list_angka)):
        min_idx = i
        for j in range(i + 1, len(list_angka)):
            if list_angka[min_idx] > list_angka[j]:
                min_idx = j

        # Swap elemen minimum dengan elemen pertama
        list_angka[i], list_angka[min_idx] = list_angka[min_idx],
list_angka[i]

# Cetak input_list yang telah diurutkan
print("Hasil pengurutan:")
print(list_angka)

```

```

Pilih mode pengurutan (1: Ascending atau 2: Descending): 1
Masukkan bilangan bulat:5
Masih ada bilangan yang dimasukkan (y/t)y
Masukkan bilangan bulat:1
Masih ada bilangan yang dimasukkan (y/t)y
Masukkan bilangan bulat:12
Masih ada bilangan yang dimasukkan (y/t)y
Masukkan bilangan bulat:-5
Masih ada bilangan yang dimasukkan (y/t)y
Masukkan bilangan bulat:16
Masih ada bilangan yang dimasukkan (y/t)y
Masukkan bilangan bulat:2
Masih ada bilangan yang dimasukkan (y/t)y
Masukkan bilangan bulat:12
Masih ada bilangan yang dimasukkan (y/t)y
Masukkan bilangan bulat:14
Masih ada bilangan yang dimasukkan (y/t)t
Hasil pengurutan:
[-5, 1, 2, 5, 12, 12, 14, 16]

```

Membuat variabel `list_angka` yang berisi array kosong untuk menyimpan input pengguna. Lalu menggunakan `while` untuk dan variabel `mode` pengurutan untuk menyimpan mode pengurutan.

Membaca input lalu menambahkan bilangan bulat tersebut ke dalam `list_angka`. Meminta pengguna untuk memasukkan pilihan "y" atau "t" untuk melanjutkan atau mengakhiri proses.

Jika pilihan pengguna adalah selain y, maka proses akan dihentikan.

Kali ini saya mengurutkan bilangan bulat dengan menggunakan Algoritma Selection Sort. Berikut penjelasan algoritma selection sort

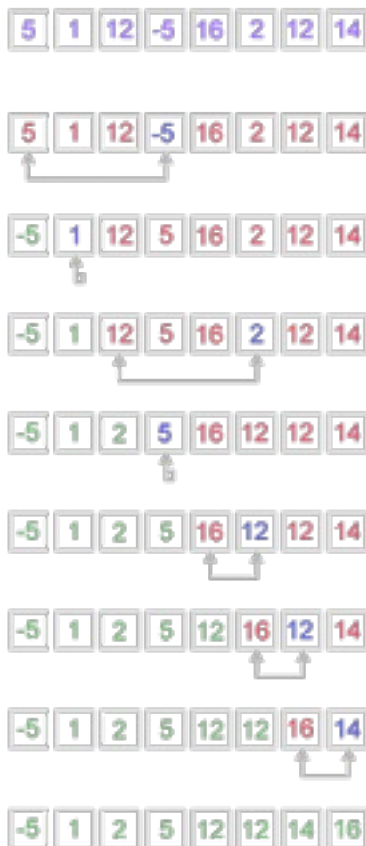
Selection sort merupakan metode pengurutan dengan mencari nilai data terkecil dimulai dari data diposisi 0 hingga diposisi N-1.

Algoritma Selection Sort bekerja dengan cara mencari elemen minimum dalam list, lalu menukar elemen minimum tersebut dengan elemen pertama. Proses ini diulangi hingga semua elemen dalam list terurut.

Jika terdapat N data dan data terkoleksi dari urutan 0 sampai dengan N-1 maka algoritma pengurutan dengan metode selection sort adalah sebagai berikut:

1. Cari data terkecil dalam interval $j = 0$ sampai dengan $j = N-1$
2. Jika pada posisi pos ditemukan data yang terkecil, tukarkan data diposisi pos dengan data di posisi i jika k.
3. Ulangi langkah 1 dan 2 dengan $j = j+1$ sampai dengan $j = N-1$, dan seterusnya sampai $j = N$

Jika kita memiliki elemen array : {5, 1, 12, -5, 16, 2, 12, 14} maka cara pengurutannya



Algoritma Selection Sort bekerja dengan cara mencari elemen minimum dalam list, lalu menukar elemen minimum tersebut dengan elemen pertama. Proses ini diulangi hingga semua elemen dalam list terurut.

D. Latihan 4

```
print("Program untuk memisahkan kata input user berdasarkan  
delimiter")  
kalimat = input("Masukkan kalimat: ")
```

```
# Pecah string menjadi list
list_hasil = kalimat.split(input("Masukkan delimiter: "))

# Tampilkan list hasil
print("List hasil:", list_hasil)

Program untuk memisahkan kata input user berdasarkan delimiter
Masukkan kalimat: ini hanyalah contoh saja
Masukkan delimiter: a
List hasil: ['ini h', 'ny', 'l', 'h contoh s', 'j', '']
```

Variabel kalimat berisi kalimat yang kita inputkan

Kemudian kita disuruh memasukkan input delimiter, yang kemudian dimasukkan ke variabel list hasil

Fungsi **split()** adalah fungsi bawaan Python yang digunakan untuk memecah string menjadi list. Fungsi ini menerima dua argumen: string yang akan dipecah dan pembatas.

Dalam contoh ini, string yang akan dipecah adalah kalimat "ini hanyalah contoh saja" dan pembatasnya adalah "a".

Setelah itu hasilnya kita print