



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231052
Nama Lengkap	GABRIEL SACHIO ATMADJAJA
Minggu ke / Materi	10 / List

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: Modul Pratikum Algoritma dan Pemrograman, Pratikum AIPro

MATERI 1

Pengertian dan Cara Akses List

List sederhananya adalah kumpulan nilai yang bisa berbeda jenis dan ditampung dalam variabel. List dibuat dengan kurung siku ([]). Untuk mengakses nilai yang ada pada list, bisa menggunakan index yang dimulai dari 0. Lalu nilai pada list juga dapat diubah. Berikut contohnya:

```
list.py > ...
  Click here to ask Blackbox to help you code faster
1 list_nilai = ["Asep", 100, True]
2 print(f>Nama: {list_nilai[0]}, nilai: {list_nilai[1]}")
3
4 list_nilai[0] = "Dimas"
5 list_nilai[1] = 50
6 print(f>Nama: {list_nilai[0]}, nilai: {list_nilai[1]}")
```

Gambar 1.1: Contoh list dan cara aksesnya

Penambahan, Perkalian List

```
8 # #PENAMBAHAN LIST
9 lista = [1, 2, 3, 4, 5]
10 listb = [4, 5, 6]
11 hasil = lista + listb
12 print(hasil)
13
14 print(hasil * 3)
15
16 ulang = [0, 1, 2] * 3
17 print(ulang)
18
19 akses = ['b', 'a', 'c', 'e']
20 print(akses[0])
21 print(akses[2])
22 print(akses[-1]) # terakhir pertama -> 'e'
```

Gambar 1.2: Program

Output:

```
PS D:\Sachio\Kampus\PrakAIPro\pert10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
[1, 2, 3, 4, 5, 4, 5, 6]
[1, 2, 3, 4, 5, 4, 5, 6, 1, 2, 3, 4, 5, 4, 5, 6, 1, 2, 3, 4, 5, 4, 5, 6]
[0, 1, 2, 0, 1, 2, 0, 1, 2]
b
c
e
```

Gambar 1.3: Output

Method List

- 1) Append → Menambahkan/memasukkan nilai ke dalam list (lewat kanan)

```
25 lst = []
26
27 lst.append(1)
28 lst.append("satu")
29 lst.append(True)
30
31 print(lst)
```

Gambar 1.4: Method append

```
PS D:\Sachio\Kampus\PrakAlPro\pert10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
[1, 'satu', True]
```

Gambar 1.5: Output

- 2) Extend → mungkin gampangnya memberikan nilai

```
33 list1 = [1, 2, 3]
34 list2 = [4, 5, 6]
35 list3 = [7, 8, 9]
36
37 list1.extend(list2)
38 list1.extend(list3)
39 print(list1)
40 print(list1 + list2 + list3)
```

Gambar 1.6: Method extend

```
PS D:\Sachio\Kampus\PrakAlPro\pert10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 4, 5, 6, 7, 8, 9]
```

Gambar 1.7: output

- 3) Sort → mengurutkan nilai dari yang terkecil ke terbesar

```
48 lst1 = [1, 2, 3, 23, 565, 40, 94, 120, 99]
49 lst1.sort()
50 print(lst1)
```

Gambar 1.8: Method sort

```
PS D:\Sachio\Kampus\PrakAlPro\pert10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
[1, 2, 3, 23, 40, 94, 99, 120, 565]
```

Gambar 1.9: output

- 4) Pop → defaultnya menghapus nilai paling kanan pada list (bisa mengembalikan nilai)

```
52 lst1 = [0, 1, 2, 3, 4, 5]
53 terpop = lst1.pop(2) # intinya pop itu bisa mengembalikan nilai
54 print(lst1)
55 print(terpop)
```

Gambar 1.10: Method pop

```
PS D:\Sachio\Kampus\PrakAlPro\pert10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d
[0, 1, 3, 4, 5]
2
```

Gambar 1.11: Output

5) Copy → menyalin list

```
57 lst1 = [0, 1, 2, 3, 4, 5]
58 lst2 = lst1.copy()
59 del lst2[3] # tidak mengembalikan nilai
60 print(lst2)
```

Gambar 1.12: Method copy

```
PS D:\Sachio\Kampus\PrakAlPro\pert10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
[0, 1, 2, 4, 5]
```

Gambar 1.13: output

6) Del → menghapus nilai dari list, tetapi tidak mengembalikan nilai (Perhatikan gambar 1.12 dan 1.13)

7) Remove → menghapus nilai yang sesuai

```
62 lst1 = [1, 2, 3, 4, 5, 3, 7, 5, 3]
63 lst1.remove(3)
64 print(lst1)
```

Gambar 1.14: Method remove

```
PS D:\Sachio\Kampus\PrakAlPro\pert10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
[1, 2, 4, 5, 3, 7, 5, 3]
```

Gambar 1.15: output

Semisal ingin hapus semua nilai 3 yang duplikat, bisa gunakan cara berikut:

```
66 for i in range(lst1.count(3)):
67     lst1.remove(3)
```

Gambar 1.16: Kode menghapus nilai 3 yang duplikat

8) Reverse → membalikkan urutan nilai dari terbesar ke terkecil

```
86 lst1 = [1, 2, 3]
87 lst1.reverse()
88 print(lst1)
```

Gambar 1.17: Method reverse

```
PS D:\Sachio\Kampus\PrakAlPro\pert10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
[3, 2, 1]
```

Gambar 1.18: Output

Perbedaan List dan String

```
96 # PERBEDAAN LIST DAN STRING  
97 a = "ini adalah string"  
98 b = ["ini", "ada", 1, 4, "h" "string"]  
99 print(type(a[0]))  
100 print(type(b[2]))  
101  
102 print(id(a))  
103 a = "tes"  
104 print(id(a))  
105  
106 print(id(b))  
107 b.append(2)  
108 print(id(b))
```

Gambar 1.19: contoh

```
PS D:\Sachio\Kampus\PrakAlPro\pert10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
<class 'str'>  
<class 'int'>  
2618386219760  
2618386051696  
2618384308352  
2618384308352
```

Gambar 1.20: output

List sebagai parameter fungsi

```
110 #LIST SEBAGAI PARAMETER FUNGSI  
111 def nama_fungsi(paramter):  
112     argument = *****  
113  
114 def hapus_index_pertama(list):  
115     return list[1:]  
116  
117 list1 = [1, 2, 3, 4]  
118 print(hapus_index_pertama(list1))
```

Gambar 1.21: List sbg parameter fungsi

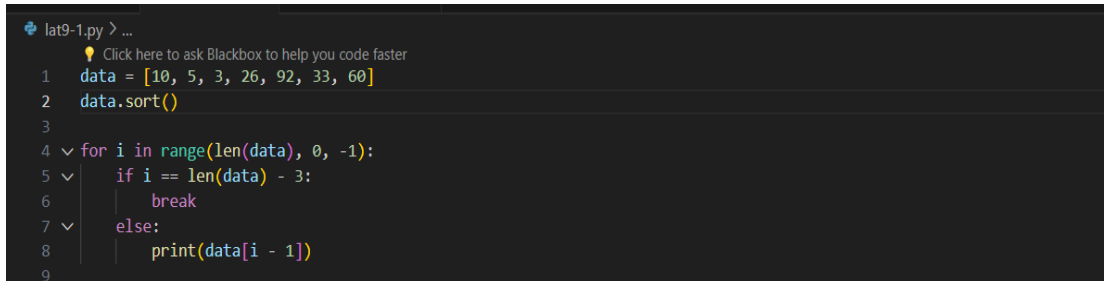
```
PS D:\Sachio\Kampus\PrakAlPro\pert10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
[2, 3, 4]
```

Gambar 1.22: output

BAGIAN 2: LATIHAN MANDIRI (60%)

Source: <https://github.com/gabrielsachioa/PrakAIPro9.git>

SOAL 1



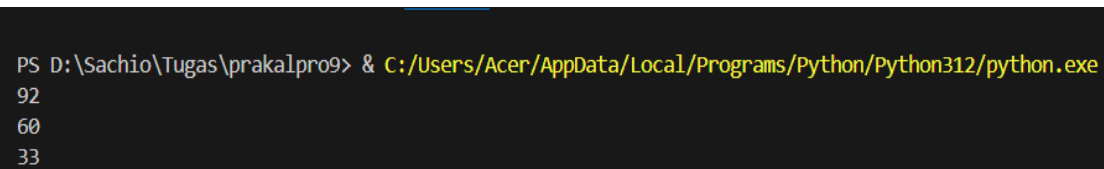
```
lat9-1.py > ...
  Click here to ask Blackbox to help you code faster
1 data = [10, 5, 3, 26, 92, 33, 60]
2 data.sort()
3
4 for i in range(len(data), 0, -1):
5     if i == len(data) - 3:
6         break
7     else:
8         print(data[i - 1])
9
```

Gambar 2.1: Contoh program

Keterangan:

Misal saja, ada variabel data yang isinya list dengan nilai acak. Lalu gunakan method sort untuk mengurutkan nilai dari terkecil ke terbesar. Setelah itu buat for loop, startnya adalah panjang list, lalu stop di 0, dan stepnya menurun (-1) karena ingin mencari 3 nilai terbaik. Setelah itu buat if → jika i sama dengan panjang(data) – 3 maka hentikan perulangan. Ini bertujuan untuk menghentikan perulangan setelah memprint 3 nilai terbaik. Lalu else → print(data[i – 1])), perlu dikurangi 1 supaya tdk error.. gampangnya karena index itu mulai dari 0

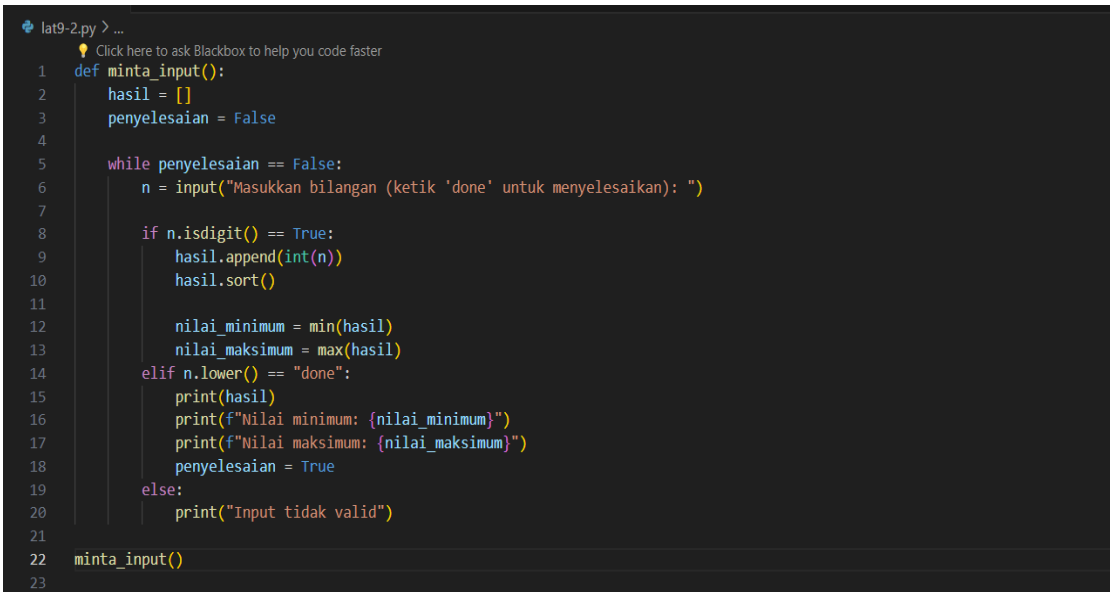
Output:



```
PS D:\Sachio\Tugas\prakalpro9> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
92
60
33
```

Gambar 2.2: output tiga nilai terbaik dari list

SOAL 2



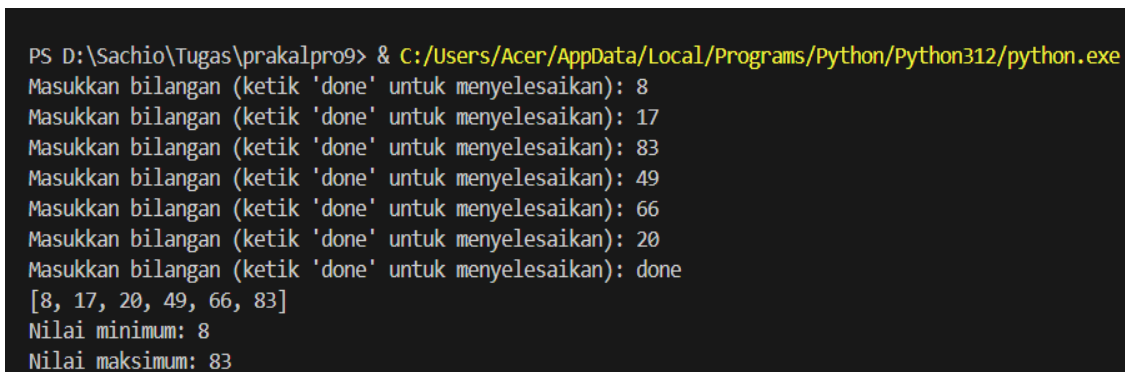
```
lat9-2.py > ...
Click here to ask Blackbox to help you code faster
1 def minta_input():
2     hasil = []
3     penyelesaian = False
4
5     while penyelesaian == False:
6         n = input("Masukkan bilangan (ketik 'done' untuk menyelesaikan): ")
7
8         if n.isdigit() == True:
9             hasil.append(int(n))
10            hasil.sort()
11
12            nilai_minimum = min(hasil)
13            nilai_maksimum = max(hasil)
14        elif n.lower() == "done":
15            print(hasil)
16            print(f"Nilai minimum: {nilai_minimum}")
17            print(f"Nilai maksimum: {nilai_maksimum}")
18            penyelesaian = True
19        else:
20            print("Input tidak valid")
21
22    minta_input()
23
```

Gambar 2.3: Program menentukan nilai minimum dan maksimum

Keterangan:

Buat fungsi `minta_input`. Di dalamnya buat variabel `hasil` dengan list kosong untuk menampung nilai yang diinputkan user. Lalu buat variabel `penyelesaian = False` agar bisa dimanipulasi untuk menghentikan perulangan `while`. Lalu buat `while penyelesaian == False` → buat variabel `n` yang meminta input dari user. Lalu kondisi `if n.isdigit() == True` itu maksudnya jika inputan user berupa digit (True) maka masukkan nilai yang diinputkan user itu ke dalam variabel `hasil`. Dan urutkan dengan `sort`. Lalu juga cari nilai min dengan menggunakan `min(hasil)` dan nilai max menggunakan `max(hasil)`. Pada kondisi `elif n.lower() == "done"` itu maksudnya jika input user adalah "done" maka print hasil, nilai minimum, dan nilai maksimum, lalu hentikan perulangan dengan membuat variabel `penyelesaian` menjadi True. Kondisi `else`, maka inputan dianggap tidak valid

Output:



```
PS D:\Sachio\Tugas\prakalpro> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
Masukkan bilangan (ketik 'done' untuk menyelesaikan): 8
Masukkan bilangan (ketik 'done' untuk menyelesaikan): 17
Masukkan bilangan (ketik 'done' untuk menyelesaikan): 83
Masukkan bilangan (ketik 'done' untuk menyelesaikan): 49
Masukkan bilangan (ketik 'done' untuk menyelesaikan): 66
Masukkan bilangan (ketik 'done' untuk menyelesaikan): 20
Masukkan bilangan (ketik 'done' untuk menyelesaikan): done
[8, 17, 20, 49, 66, 83]
Nilai minimum: 8
Nilai maksimum: 83
```

Gambar 2.4: Output nilai minimum dan maksimum

SOAL 3

Teks berita:

```
teks_berita.txt
Click here to ask Blackbox to help you code faster
1 Heboh Pedagang Pasar Induk Kramatjati Buang Puluhan Ton Pepaya
2
3 Jakarta, CNN Indonesia -- Pedagang buah di Pasar Induk Kramatjati, Jakarta, ramai-ramai membuang puluhan ton pepaya yang tidak laku pada Jumat (2
4 Meski harganya murah, buah berwarna oranye ini tetap tak laris. Kejadian ini sudah terjadi sejak dua pekan lalu, tepatnya setelah Hari Raya Idul
5
6 "Udah seminggu lalu, semenjak Lebaran, karena saking banyaknya pepaya, permintaannya nggak ada," kata petugas loket parkir dekat tempat pembuangan
7
8 Seorang pedagang mengatakan pepaya-pepaya tersebut sudah mulai membusuk sehingga memang harus dibuang.
9
10 "Masih banyak itu, pepaya kan ya udah tua, nggak mungkin dibiarkan," katanya.
11
12 Selain tak laku, ia juga mengeluhkan harga pepaya anjlok yakni Rp3.000 per kilogram (kg). Menurutny, permintaan pasar tidak sesuai dengan pasokan
13
14 "Iya itu mulai habis Lebaran panennya tinggi, tapi nggak ada yang mau beli. Bener-bener banyak itu yang dibuang," pungkasnya.
```

Gambar 2.5: Teks berita

Program:

```
lat9-3.py > ...
Click here to ask Blackbox to help you code faster
1 with open("teks_berita.txt", "r") as file:
2     kata = file.read().split()
3
4     print(kata)
```

Gambar 2.6: Program

Keterangan:

Intinya, buka terlebih dahulu file txtnya.. dalam kasus ini saya gunakan with open, dengan mode read, dan simpan di variable file. Lalu file.read() untuk membaca seluruh baris dan kata, setelah itu .split() untuk membagi menjadi token.. Perlu diketahui split akan membuat list untuk daftar token-token itu. Lalu saya simpan di variabel kata. Setelah itu tinggal print variabel kata

Output:

```
PS D:\Sachio\Tugas\prakalpro9> & C:\Users\Acer\AppData\Local\Programs\Python\Python312\python.exe d:/Sachio/Tugas/prakalpro9/lat9-3.py
['Heboh', 'Pedagang', 'Pasar', 'Induk', 'Kramatjati', 'Buang', 'Puluhan', 'Ton', 'Pepaya', 'Jakarta,', 'CNN', 'Indonesia', '--', 'Pedagang', 'buah', 'di', 'Pasar', 'Induk', 'Kramatjati,', 'Jakarta,', 'ramai-ramai', 'membuang', 'puluhan', 'ton', 'pepaya', 'yang', 'tidak', 'laku', 'pada', 'Jumat', '(26/4).', 'Di', 'beberapa', 'lokasi', 'pembuangan', 'sampah', 'tampak', 'pepaya', 'rusak', 'menumpuk.', 'Meski', 'harganya', 'murah,', 'buah', 'berwarna', 'oranye', 'ini', 'tetap', 'tak', 'laris.', 'Kejadian', 'ini', 'sudah', 'terjadi', 'sejak', 'dua', 'pekan', 'lalu,', 'tepatnya', 'setelah', 'Hari', 'Raya', 'Idulfitri.', 'Udah', 'seminggu', 'lalu,', 'semenjak', 'Lebaran,', 'karena', 'saking', 'banyaknya', 'pepaya', 'permintaannya', 'nggak', 'ada,', 'kata', 'petugas', 'loket', 'parkir', 'dekat', 'tempat', 'pembuangan', 'pepaya', 'Pasar', 'Induk', 'Kramatjati,', 'Jakarta,', 'Jumat', '(26/4).', 'Seorang', 'pedagang', 'mengatakan', 'pepaya-pepaya', 'tersebut', 'sudah', 'mulai', 'membusuk', 'sehingga', 'memang', 'harus', 'dibuang.', 'Masih', 'banyak', 'itu', 'pepaya', 'kan', 'ya', 'udah', 'tua', 'nggak', 'mungkin', 'dibiarkan.', 'katanya.', 'Selain', 'tak', 'laku', 'ia', 'juga', 'mengeluhkan', 'harga', 'pepaya', 'anjlok', 'yakni', 'Rp3.000', 'per', 'kilogram', '(kg).', 'Menurutny,', 'permintaan', 'pasar', 'tidak', 'sesuai', 'dengan', 'pasokan', 'pepaya', 'yang', 'tengah', 'panen', 'raya.', 'Iya', 'itu', 'mulai', 'habis', 'Lebaran', 'panennya', 'tinggi', 'tapi', 'nggak', 'ada', 'yang', 'mau', 'beli.', 'Bener-bener', 'banyak', 'itu', 'yang', 'dibuang.', 'pungkasnya.']
```

Gambar 2.7: Output