**LAPORAN RESMI**

**MODUL V**

**LISTS DAN TUPEL**

**ALGORITMA PEMROGRAMAN**



**NAMA : GIRALDO STEVANUS**

**N.R.P : 220441100064**

**DOSEN : SIGIT SUSANTO PUTRO,S.Kom,M.Kom.**

**ASISTEN : M.BADRUT TAMAM**

**TGL PRAKTIKUM : 25 OKTOBER 2022**

**Disetujui : 1 November 2022**

**Asisten**

**M.BADRUT TAMAM**

**21.04.411.00032**

**LABORATORIUM BISNIS INTELIJEN SISTEM**

**PRODI SISTEM INFORMASI**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Tiga (dari 4) tipe data kolektif pada python –yaitu list, set dan tuple– memiliki berbagai macam kesamaan dan juga berbagai macam perbedaan.

Masing-masing sifat pada tiap tipe data tersebut memberikan kita keleluasaan untuk memilih mana tipe data yang pas untuk suatu kasus, dan mana tipe data yang cocok untuk suatu kasus yang lain.

Oke karena ini adalah pertemuan terakhir dari seri belajar python dasar untuk pemula, pada pada pertemuan selanjutnya kita akan memulai seri baru belajar python untuk tingkat menengah.

Daftar adalah bagian teks di dalam dokumen yang menyimpan item dalam urutan tertentu. Daftar memungkinkan anda untuk menyimpan beberapa item atau bahkan jutaan item dalam satu tempat. Daftar adalah salah satu fitur dari bahasa python yang sangat kuat. Di kuliah online sebelumnya kita telah membahas tentang variabel dan string. pada kuliah online kali ini kita akan membahas daftar dan tuple. kita akan mempelajari lebih jauh tentang daftar dan tuple pada bahasa pemrograman python

## **Tujuan**

1. Dapat mengetahui dan memahami perintah dasar dalam Python
2. Dapat mengetahui bentuk umum dari list & tupel data
3. Dapat menggunakan dan mendeklarasikan python
4. Dapat membuat program sederhana untuk list dan tupel
5. Mengetahui manfaat dan kekurangan tupel & list dalam python

# **BAB II DASAR TEORI**

## **2.1 Pengertian List**

**A. List Python**

Dalam bahasa pemrograman Python, struktur data yang paling dasar adalah urutan atau lists. Setiap elemen-elemen berurutan akan diberi nomor posisi atau indeksnya. Indeks pertama dalam list adalah nol, indeks kedua adalah satu dan seterusnya.

Python memiliki enam jenis urutan built-in, namun yang paling umum adalah list dan tuple. Ada beberapa hal yang dapat Anda lakukan dengan semua jenis list. Operasi ini meliputi pengindeksan, pengiris, penambahan, perbanyak, dan pengecekan keanggotaan. Selain itu, Python memiliki fungsi built-in untuk menemukan panjang list dan untuk menemukan elemen terbesar dan terkecilnya.

**B. Membuat List Python**

Membuat List Python List adalah tipe data yang paling serbaguna yang tersedia dalam bahasa Python, yang dapat ditulis sebagai daftar nilai yang dipisahkan koma (item) antara tanda kurung siku. Hal penting tentang daftar adalah item dalam list tidak boleh sama. jenisnya. Membuat list sangat sederhana, tinggal memasukkan berbagai nilai yang dipisahkan koma di antara tanda kurung siku. Dibawah ini adalah contoh sederhana pembuatan list dalam bahasa Python.

list1 =('physics','chemistry', 1997, 2000)

list2 = [1, 2, 3, 4, 5];

list3 = ["a", "b", "c", "d"]

Untuk mengakses nilai dalam list python, gunakan tanda kurung siku untuk mengiris beserta indeks atau indeks untuk mendapatkan nilai yang tersedia pada indeks tersebut.

list1 = ['physics', 'chemistry', 1997, 2000];

list2 = [1, 2, 3, 4, 5, 6, 7]

print ("Nilai list1[0]:", list1[0])

print ("Nilai list1[1:5]:", list2[1:5])

Outputnya:

Nilai list1[0]: physics

Nilai list1[1:5]: [2, 3, 4, 5]

Anda dapat memperbarui satu atau beberapa nilai di dalam list dengan memberikan potongan di sisi kiri operator penugasan, dan Anda dapat menambahkan nilai ke dalam list dengan metode append ().

list =('physics','chemistry', 1997, 2000)

print("nilai indeks ke-2 saat ini:")

print (list[2])

list[2]= 2001;

print("nilai indeks ke-2 setelah diperbaharui:")

print (list[2])

outputnya:

nilai indeks ke-2 saat ini:

1997

nilai indeks ke-2 setelah diperbaharui:

2001

C. Menghapus List

Untuk menghapus nilai di dalam list python, Anda dapat menggunakan salah satu pernyataan del jika Anda tahu persis elemen yang Anda hapus. Anda dapat menggunakan metode remove() jika Anda tidak tahu persis item mana yang akan dihapus.

print("nilai indeks ke-2 saat ini:")

print (list[2])

F. Operasi Dasar Pada List Python

List Python merespons operator + dan \* seperti string; Itu artinya penggabungan dan pengulangan di sini juga berlaku, kecuali hasilnya adalah list baru, bukan sebuah String. Sebenarnya, list merespons semua operasi urutan umum yang kami gunakan pada String di bab sebelumnya. Dibawah ini adalah tabel daftar operasi dasar pada list python.

|  |  |  |
| --- | --- | --- |
| Python Expression | Hasil | Penjelasan |
| Len ([1,2,3,4]) | 4 | length |
| [1, 2, 3] + [4, 5, 6] | 1, 2, 3, 4, 5, 6 | concenation |
| [‘Halo!’] \* 4 | [‘Halo’!, ‘Halo!’, ‘Halo!’, ‘Halo!’] | Repetition |
| 2 in [1, 2, 3] | True | Membership |
| for x in [1, 2, 3] : print (x, end = “) | 1, 2, 3 | Iteration |

Indexing, Slicing dan Matrix Pada List Python

Karena list adalah urutan, pengindeksan dan pengiris bekerja dengan cara yang sama untuk list seperti yang mereka lakukan untuk String.

Dengan asumsi input berikut :

L= [‘C++”, “Java, “Python]

|  |  |  |
| --- | --- | --- |
| Python Expression | Hasil | Penjelasan |
| L[2] | [‘Python’] | Offset mulai dari nol. |
| L[-2] | [‘Java’] | Negatif: hitung dari kanan. |
| L[1:] | [‘Java’, ‘Python’] | Negatif: hitung dari kanan. |

Method dan Fungsi Build-in Pada List Python

Python menyertakan fungsi built-in sebagai berikut:

|  |  |
| --- | --- |
| Python Function | Penjelasan |
| cmp (list1, list2) | #tidak lagi tersedia dengan python 3. |
| len (list) | Memberikan total panjang list. |
| max (list) | Mengembalikan item dari list dengan nilai maks. |
| min (list) | Mengembalikan item dari list dengan nilai min. |
| list (seq) | Mengembalikan item dari list dengan nilai min. |

Mengembalikan item dari list dengan nilai min.

|  |  |
| --- | --- |
| Python Methods | Penjelasan |
| list.append (obj) | Menambahkan objek ke list. |
| list.count (obj) | Jumlah pengembalian berapa kali obj terjadi dalam list. |
| Jumlah pengembalian berapa kali obj terjadi dalam list | Jumlah pengembalian berapa kali obj terjadi dalam list |
| Jumlah pengembalian berapa kali obj terjadi dalam list | Mengembalikan indeks terendah dalam list yang muncul obj |
| list.insert (index, obj) | Sisipkan objek obj ke dalam list di indeks offset. |
| list.pop(obj = list[- 1]) | Menghapus dan mengembalikan objek atau obj terakhir dari list |
| list.remove(obj) | Removes object obj from list |
| list.reverse() | Membalik list objek di tempat |
| list.sort([func]) | Urutkan objek list, gunakan compare func jika diberikan |

## **2.2 Pengertian Tupel**

**A. Tuple**

Sebuah tupel adalah urutan objek Python yang tidak berubah. Tupel adalah urutan, seperti daftar. Perbedaan utama antara tupel dan daftarnya adalah bahwa tupel tidak dapat diubah tidak seperti List Python. Tupel menggunakan tanda kurung, sedangkan List Python menggunakan tanda kurung siku.

Membuat tuple semudah memasukkan nilai-nilai yang dipisahkan koma. Secara opsional, Anda dapat memasukkan nilai-nilai yang dipisahkan koma ini di antara tanda kurung juga.

tup1 = (‘physics’, ‘chemistry’, 1997, 2000)

tup2 = (1, 2, 3, 4, 5 ) ;

tup3 = “a”, ”b”, “c”, “d”;

**B. Akses Nilai Dalam Tuple**

Untuk mengakses nilai dalam tupel, gunakan tanda kurung siku untuk mengiris beserta indeks atau indeks untuk mendapatkan nilai yang tersedia pada indeks tersebut.

tup1 = (‘’, ‘’, 1997, 2000);

tup2 = (1, 2, 3, 4, 5, 6, 7)

print (“ tup1[0]: “, tup1[0])

print (“ tup2[1:5]: “, tup2[1,5])

**Output**

tup1 [0] : physics

tup2 [1:5] : [2, 3, 4, 5]

**C. Memperbarui Nilai Dalam Tuple**

Tupel tidak berubah, yang berarti Anda tidak dapat memperbarui atau mengubah nilai elemen tupel. Anda dapat mengambil bagian dari tupel yang ada untuk membuat tupel baru seperti ditunjukkan oleh contoh berikut.

**Output**

12, 34.56, ‘abc’, ‘xyz’

tup1 = (12, 34.56);

tup2 = (‘abc’, ‘xyz’);

# Syntaks seperti dibawah tidak bisa diterapkan di Tupel

# tup1[0] = 100

# Tupel hannya dapat mengambil isi dimasukkan ke Tupel baru

tup3 = tup1 + tup2;

print (tup3)

**D. Menghapus nilai dalam Tupel**

Menghapus elemen tuple individual tidak mungkin dilakukan. Tentu saja, tidak ada yang salah dengan menggabungkan tupel lain dengan unsur-unsur yang tidak diinginkan dibaung. Untuk secara eksplisit menghapus keseluruhan tuple, cukup gunakan del statement.

tup = (‘physics’, ‘chemistry’, 1997, 2000);

print (tup)

del tup;

print “setelah tup dihapus: ”

print (tup)

Output:

Physics, chemistry, 1997, 2000

Setelah tup dihapus:

Traceback (most recent call last):

File ”test.py”, nilai 9, in <module>

Print tup;

NameError: name ‘tup’ is not defined

**E. Operasi Sederhana Pada Tupel**

Tupel merespons operator + dan \* sama seperti String; Mereka berarti penggabungan dan pengulangan di sini juga berlaku, kecuali hasilnya adalah tupel baru, bukan string. Sebenarnya, Tuple merespons semua operasi urutan umum yang kami gunakan pada String di bab sebelumnya. Dibawah ini adalah tabel daftar operasi dasar pada Tuple python.

|  |  |  |
| --- | --- | --- |
| Python Expression | Hasil | Penjelasan |
| Len ((1, 2, 3)) | 3 | lenght |
| (1, 2, 3) + (4, 5, 6) | (1, 2, 3, 4, 5, 6) | Concatenation |
| (‘Halo!’,) \* 4 | (‘Halo!’, ‘Halo!’, ‘Halo!’, ‘Halo!’) | Repetition |
| 3 in (1, 2, 3) | True | Membership |
| For x in (1, 2, 3) : print (x, end = ‘ ‘ ) | 1 2 3 | Iteration |

Indexing, Slicing dan Matrix Pada Tuple Python

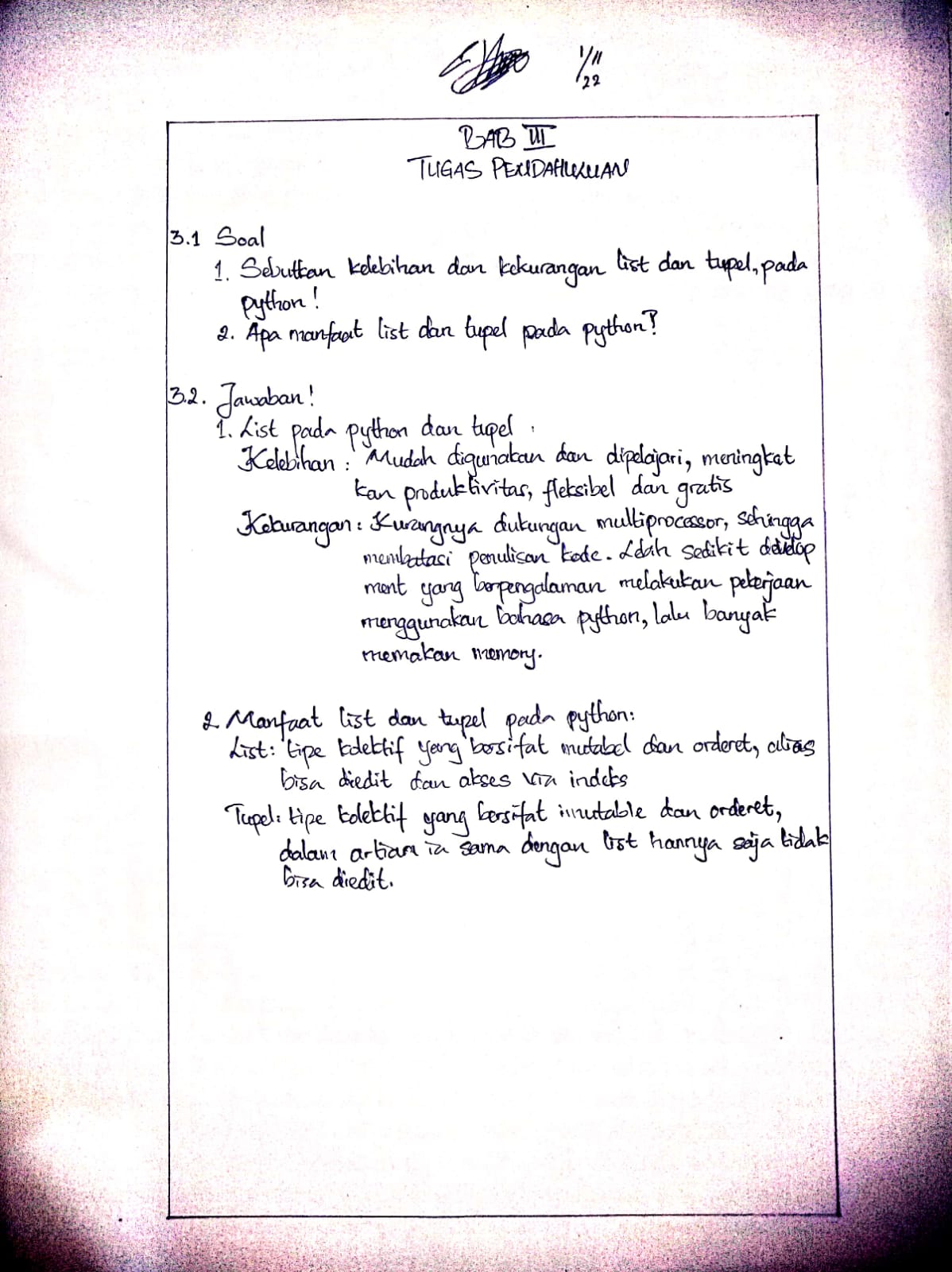
Karena tupel adalah urutan, pengindeksan dan pengiris bekerja dengan cara yang sama untuk tupel seperti pada String, dengan asumsi masukan berikut **T = ('C++', 'Java', 'Python')**

|  |  |  |
| --- | --- | --- |
| **Python Expression** | **Hasil** | **Penjelasan** |
| T [2] | ‘python’ | Offset mulai dari nol |
| T [-2] | ‘java’ | Negatif: dihitung dari kanan |
| T [1:] | (‘java’, ‘python’) | Slicing mengambil bagian |

Fungsi build in pada tupel Python

Python menyertakan fungsi build-in sebagai berikut

|  |  |
| --- | --- |
| Python function | penjelasan |
| cmp (tupel1, tupel2) | # tidak lagi tersedia dengan python 3 |
| len (tupel) | Memberikan total panjang tupel |
| max (tupel) | Mengembalikan item dari tupel dengan nilai max |
| min (tupel) | Mengembalikan item dari tupel dengan nilai min |
| Tupel (seq) | Mengubah tupel menjadi tupel |

* program utamanya.

# **BAB IV IMPLEMENTASI**

## **Source Code**

**4.1.1** Buatlah program List pada python (dinamis) dengan ada pertanyaan ingin mengulang iya atau tidak dan dengan Inputan yang sudah ditentukan

while True:

    list1=[]

    for i in range(5):

        item = input("Masukkan 15 nama makanan: ")

        list1.append(item)

    list2=[]

    for i in range(5):

        item2 = input("Masukkan 5 nama minuman: ")

        list2.append(item2)

    list3=[]

    for i in range(5):

        item3 = input("Masukkan 5 nama teman: ")

        list3.append(item3)

    list4=[]

    for i in range(5):

        item4 = input("Masukkan 5 bulan lahir teman: ")

        list4.append(item4)

    list5=[]

    for i in range(5):

        item5 = input("Masukkan 5 tanggal lahir teman: ")

        list5.append(item5)

    # rubah list3 index 3 dan 5

    index4 = [str(input("masukkan update output 2 yang diinginkan pada list3 index 3: "))]

    index5 = [str(input("masukkan update output 2 yang diinginkan pada list3 index 5: "))]

    # output pertama

    print (list1[0:5] + list2)

    # output kedua

    list3[3] = [4]

    list3[4] = [5]

    print(list3)

    print ("nilai baru dari list3 index 3 adalah",index4," dan nilai baru dari list3 index 4 adalah",index5)

    # output ketiga

    del list5[1]

    del list5[4]

    print("setelah index 1 dan 4 dihapus:",list5)

    # output keempat

    a = list4 + list5

    print(a)

    print ("nilai max dari keduanya adalah",max(a),"dan nilai min dari keduanya adalah",min(a))

    n = str(input("ingin mengulang y atau n?: "))

    if n == "y":

        continue

    elif n == "n":

        print("program berakhir")

        break

    else:

        print("tidak ada dalam pilihan program berakhir")

        break

**4.1.2** Program Tupel pada python (dinamis) dengan inputan yang sudah tertera

# Program 2

# Data\_Tuple

print("Tuple 1 : Nama Kota")

Tuple1 = (input(tuple()),input(tuple()),input(tuple()),input(tuple()),input(tuple()))

Tuple2 = (12,20,23,31,19,30,45,98)

Tuple3 = ("pEmograman", "pemograman", "Pemograman", "PemOgRamAn", "PEMOGRAMAN")

print("=========================================")

print("output 1 :")

print(Tuple3[1],Tuple3[3],Tuple3[4],Tuple1[3],Tuple1[1])

print("=========================================")

print("output 2 : ")

for i in range(1,5) :

    a = Tuple2[2]

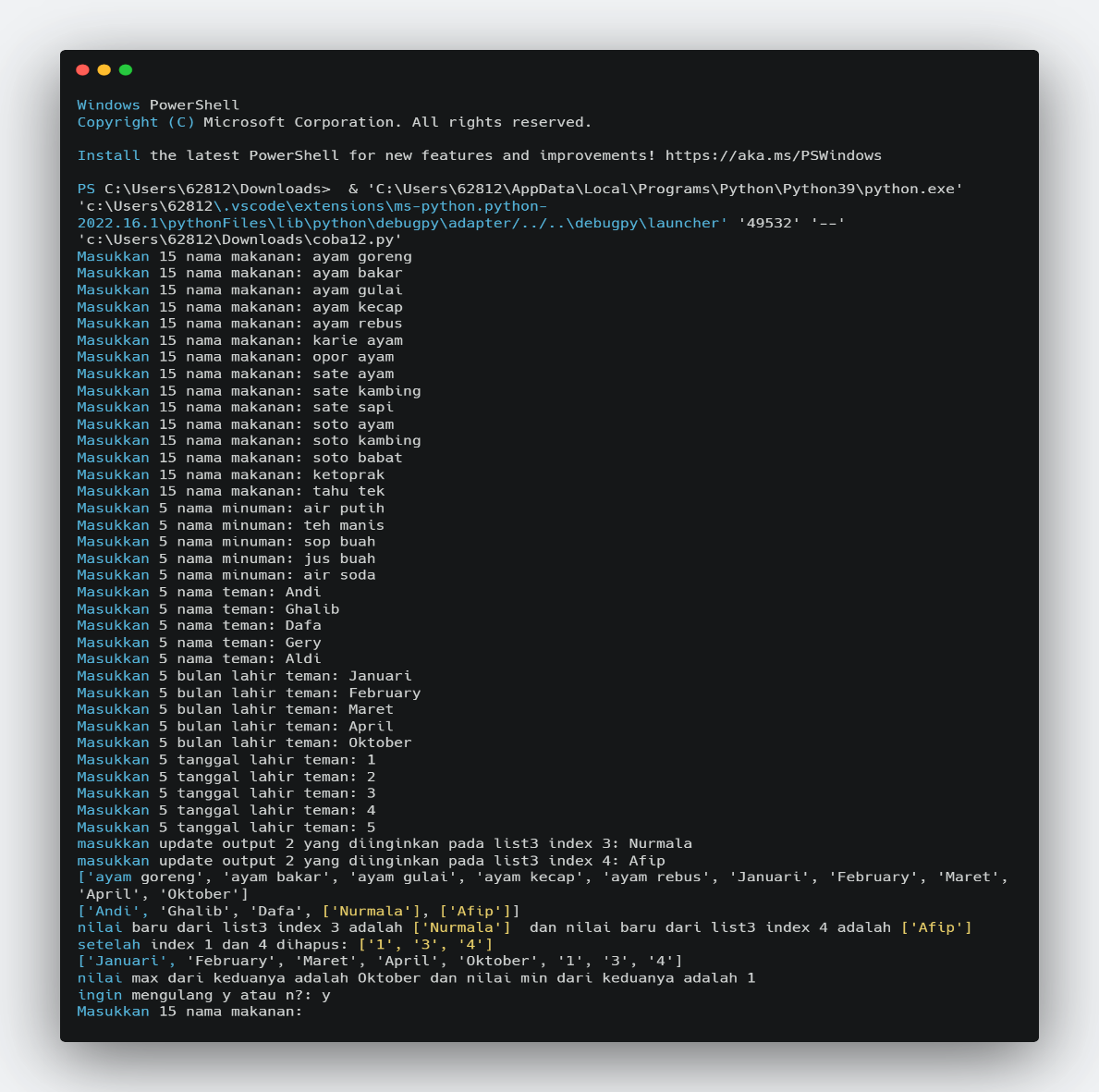
    print(a)

    b = Tuple2[4]

    print(b)

## **4.2 Hasil**

**4.1.1**

****

Penjelasannya: membuat data inputan terlebih dahulu dengan loopingan while True sehingga untuk mencari nilai list1 dengan perulangan for i in range(5): mengeluarkan nama makanan, untuk mencari nilai list2 dengan perulangan for i in range(5): mengeluarkan nama minuman, untuk mencari nilai list3 dengan perulangan for i in range(5): mengeluarkan nama teman, untuk mencari nilai list1 dengan perulangan for i in range(5): mengeluarkan bulan lahir teman, untuk mencari nilai list1 dengan perulangan for i in range(5): mengeluarkan TLT. Diperintahkan untuk mengubah lists indeks 3 dan 5, lalu untuk output pertama list 1 harus diawali dari awal bilangan sampai ke no.5 digabungkan dengan lists2.

**4.1.2**

****

Penjelasannya: ketika menampilkan menu untuk mengeluarkan nama kota (Tupel 1), sesudah itu dengan outputnya 1: pemograman Pem0gRamAn PEMOGRAMAN Surabaya Tangerang, outputnya ke-2: mengeluarkan perintah dengan perulangan (dinamis).

# **BAB V PENUTUP**

## **Analisa**

Dari hasil praktek dan pengamatan secara berturut-turut fungsi digunakan sebagi program untuk mempermudah bagaimana cara mentukan sebuah operasinya jika ada study kasus bahkan caranya juga tergolong simple.

Sama halnya dengan penggunaan perulangan tapi hannya sedikit perbedaan yakni bentuk pemanggilan pada sebuah program untuk lebih singkat saja. Dari hasil praktikum kali ini kita dapat memahami fungsi menghitung suhu, membuat program menentukan list\_data, dan membuat system dinamis.

Dalam python tedapat dua perintah yang digunakan untuk membuat sebuah perintah. Sehingga def dalam perintah python merupakan program yang execuble artinya function ini tidak akan aktif sampai python merunning perintah def tersebut. Sedangkan lamda dalam python lebih dikenal dengan nama Anonymous Function (fungsi yang tidak disebutkan namanya). Lamda sebuah perintah (statemen) namun lebih dikenal kepada ekspresi.

## **Kesimpulan**

Program besar dapat di pisah-pisah menjadi program-program kecil melaluifunction. Kemudahan dalam mencari kesalahan-kesalahan karena alur logika jelas dan kesalahan dapat dilokalisasi dalam suatu modul tertentu. Memperbaiki atau memodifikasi program dapat dilakukan pada suatu modul tertentu saja tanpa menggangu keseluruhan program. Dapat digunakan kembali (Reusability) oleh program atau fungsi lain. Meminimalkan penulisan perintah yang sama.