

Dasar Pemrograman

07a Struktur data list

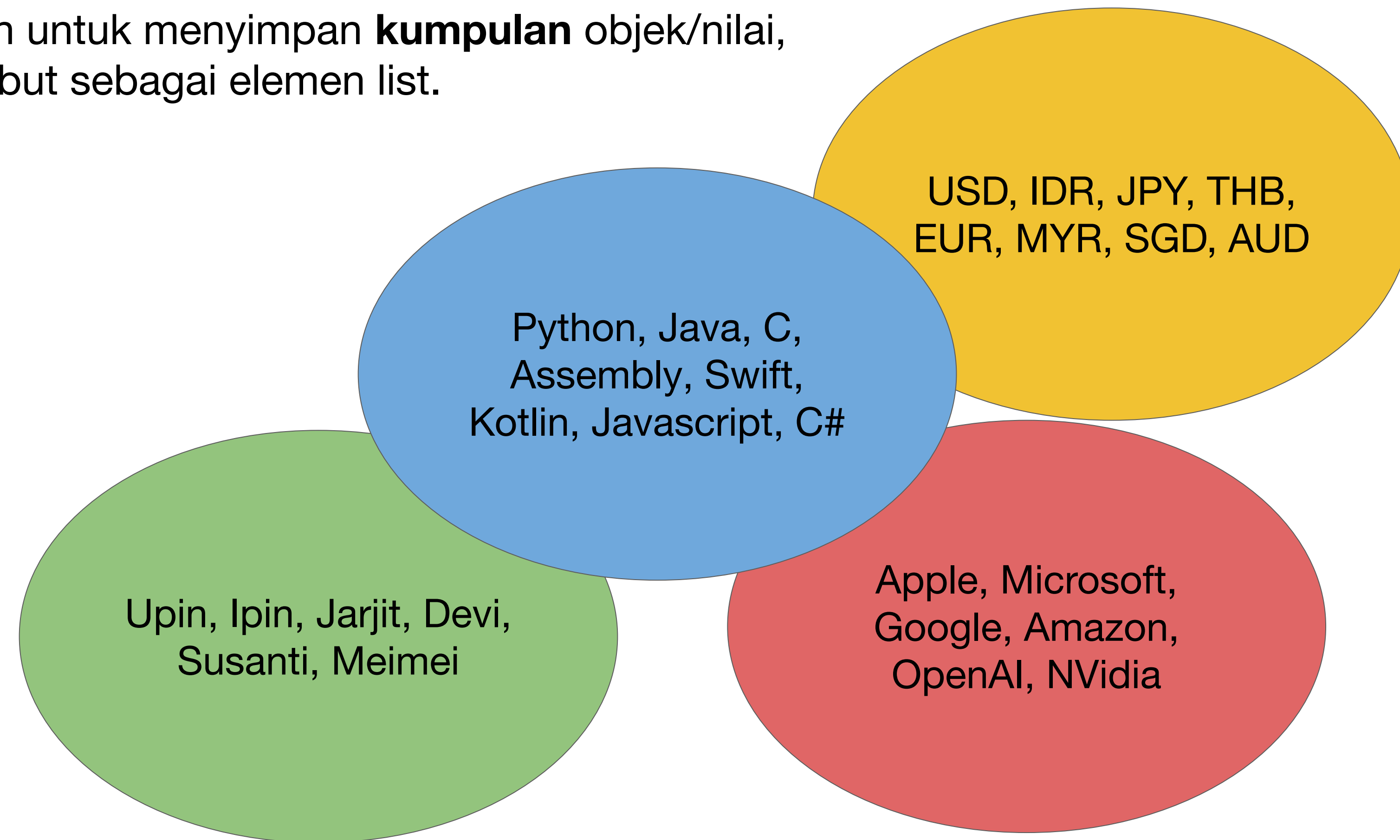
AES / 2024 05 22



Struktur data *list*

List adalah salah satu jenis struktur data dalam bahasa pemrograman python.

List dapat kita gunakan untuk menyimpan **kumpulan** objek/nilai, yang kemudian kita sebut sebagai elemen list.



Deklarasi dan inisialisasi list

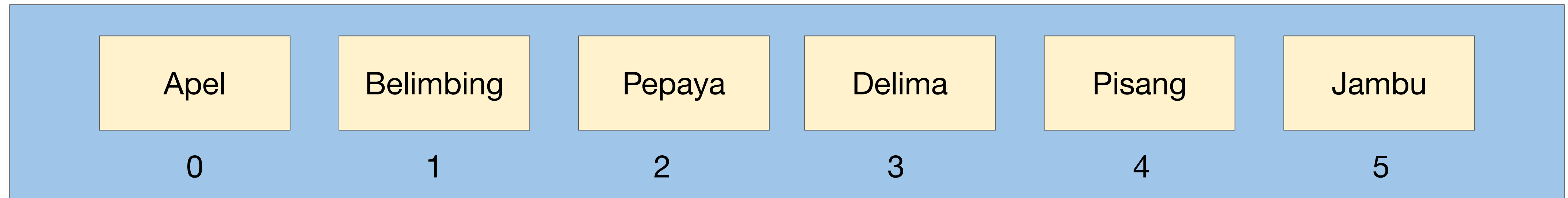
Deklarasi dan inisialisasi variabel dengan tipe list dilakukan dengan menggunakan kurung siku “[]”

Tiap elemen list dipisahkan dengan tanda koma

```
# contoh cara deklarasi dan inisialisasi list
list_kosong = []
buah_buahan = ["Apel", "Belimbing", "Pepaya", "Delima", "Pisang", "Jambu"]
bil_kuadrat = [1, 4, 9, 16, 25]
l_bhs_pemrograman = ["Java", "C", "Python", "Kotlin", "Dart", "Javascript", "Go"]
global_it_companies = ["Google", "Microsoft", "Apple", "NVidia", "Amazon"]
tdk_mesra_students = ["Upin", "Ipin", "Susanti", "Meimei", "Jarjit"]
```

Akses elemen list

elemen list dapat diakses menggunakan indeks/nomor urutnya, yg dimulai dari 0



```
buah_buahan = ["Apel", "Belimbing", "Pepaya", "Delima", "Pisang", "Jambu"]  
print(buah_buahan[2]) # akan mencetak "Pepaya"  
print(buah_buahan[4]) # akan mencetak "Pisang"
```

indeks

Update elemen list

Elemen tertentu pada list dapat diupdate/diberi nilai baru sebagaimana variabel biasa.
Sintaks yang digunakan `list[indeks] = nilai_baru`

Kondisi awal

```
buah_buahan = ["Apel", "Belimbing", "Pepaya", "Delima", "Pisang", "Jambu"]
```

Elemen pada indeks 3 (awal)



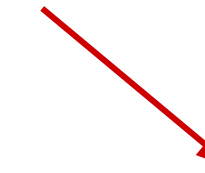
Ganti elemen pada indeks 3

```
buah_buahan[3] = "Durian"
```

Kondisi baru

```
buah_buahan = ["Apel", "Belimbing", "Pepaya", "Durian", "Pisang", "Jambu"]
```

Elemen pada indeks 3 (baru)



Menambah elemen list

Untuk menambahkan elemen baru ke dalam list, dapat digunakan fungsi ***append*** dan ***insert***. Fungsi ***append*** akan meletakkan elemen baru ke ujung list, sedangkan fungsi ***insert*** digunakan bila ingin menyisipkan elemen baru pada posisi tertentu.

```
# elemen list dapat ditambah
buah_buahan = ["Apel", "Belimbing", "Pepaya", "Delima", "Pisang", "Jambu"]
print(buah_buahan) # cetak isi list (kondisi awal)

# Cara 1: Menggunakan append, elemen baru akan diletakkan di akhir list
buah_buahan.append("Manggis")
print(buah_buahan)

# Cara 2: Menggunakan insert, elemen baru akan disisipkan pada indeks yang diberikan
buah_buahan.insert(2, "Markisa")
print(buah_buahan)
```

Menghapus elemen list

Untuk menghapus elemen pada list, dapat digunakan fungsi ***remove***, ***del*** dan ***pop***.

Fungsi ***remove*** untuk menghapus berdasarkan nilai, sedangkan fungsi ***del*** dan ***pop*** digunakan untuk menghapus berdasarkan indeksnya.

```
buah_buahan = ["Apel", "Belimbing", "Pepaya", "Delima", "Pisang", "Jambu"]
print(buah_buahan)

# menghapus menggunakan remove
buah_buahan.remove("Belimbing")
print(buah_buahan)

# menghapus menggunakan del
del buah_buahan[2]
print(buah_buahan)

# menghapus menggunakan pop
buang = buah_buahan.pop(1)
print(buah_buahan)
print(buang)
```

Fungsi `len()` untuk mendapatkan jumlah elemen pada list

Jumlah/banyaknya elemen pada list bisa didapatkan dengan menggunakan fungsi `len(list)`

Menghasilkan nilai 0 karena tidak ada
elemen pada list

```
list_kosong = []  
print(f"Jumlah elemen pada list_kosong adalah {len(list_kosong)}")  
  
buah_buahan = ["Apel", "Belimbing", "Pepaya", "Delima", "Pisang", "Jambu"]  
print(f"Jumlah elemen pada list buah_buahan adalah {len(buah_buahan)}")
```

Menghasilkan nilai 6 sesuai banyaknya elemen pada
`buah_buahan`

Loop pada list

Cara 1 : Menggunakan while

```
# Loop menggunakan indeks dengan while dan len

buah_buahan = ["Apel", "Belimbing", "Pepaya", "Delima", "Pisang", "Jambu"]
i = 0
while i < len(buah_buahan):
    print(buah_buahan[i])
    i+=1
print()
```

Loop pada list

Cara 2 : Menggunakan for

```
# Loop menggunakan indeks dengan for-range dan len  
  
buah_buahan = ["Apel", "Belimbing", "Pepaya", "Delima", "Pisang", "Jambu"]  
for i in range(len(buah_buahan)) :  
    print(buah_buahan[i])
```

Loop pada list

Cara 3 : Menggunakan for each

```
# Loop menggunakan for each
```

```
buah_buahan = ["Apel", "Belimbing", "Pepaya", "Delima", "Pisang", "Jambu"]
```

```
for buah in buah_buahan:  
    print(buah)
```

Fungsi *sort()* dan *reverse()*

Elemen pada list dapat diurutkan dengan menggunakan fungsi *sort()*, fungsi ini akan mengurutkan dari nilai terkecil ke nilai terbesar.

Fungsi *reverse()*, membalikkan urutan elemen pada list. Fungsi ini bukan kebalikan dari fungsi *sort()*. Ia akan membalikkan urutan elemen pada list, terlepas dari sebelumnya list itu sudah terurut atau tidak.

```
buah_buahan = ["Apel", "Belimbing", "Pepaya", "Delima", "Pisang", "Jambu"]
buah_buahan.sort()
print(buah_buahan)

l_bhs_pemrograman = ["Java", "C", "Python", "Kotlin", "Dart", "Javascript", "Go"]
l_bhs_pemrograman.reverse()
print(l_bhs_pemrograman)
```

Memeriksa keberadaan elemen pada list dengan *in*

Keyword *in* dapat digunakan untuk memeriksa apakah suatu nilai ada dalam list.

```
buah_buahan = ["Apel", "Belimbing", "Pepaya", "Delima", "Pisang", "Jambu"]  
ada_durian = "Durian" in buah_buahan  
ada_belimbing = "Belimbing" in buah_buahan  
  
print(ada_durian)  
print(ada_belimbing)
```

Menghasilkan *False* karena "Durian"
tidak ada pada *buah_buahan*

Menghasilkan *True* karena "Belimbing" ada pada
buah_buahan

Contoh soal 1

Berikut adalah list berisi nilai hasil UTS mahasiswa. Buatlah program untuk menemukan nilai UTS terkecil, dengan menggunakan perulangan pada list.

```
# list berisi hasil UTS mahasiswa  
nilai_uts = [35, 65, 78, 75, 22, 87, 90, 32, 85, 87, 65]
```

Contoh soal 2

Berikut adalah list berisi nilai hasil UTS mahasiswa. Buatlah program untuk menemukan nilai rata-rata UTS, dengan menggunakan perulangan pada list.

```
# list berisi hasil UTS mahasiswa
nilai_uts = [35, 65, 78, 75, 22, 87, 90, 32, 85, 87, 65]
```

Contoh soal 3

Faktor sebuah bilangan bulat adalah suatu bilangan yang membagi habis bilangan tersebut.

Buatlah program yang menerima sebuah bilangan bulat kemudian mencari seluruh faktor dari bilangan tersebut menggunakan perulangan, meletakkannya di dalam sebuah list dan kemudian mencetaknya ke layar.

petunjuk:

gunakan perulangan dan operator modulus untuk memeriksa seluruh bilangan yang dapat membagi habis bilangan bulat tersebut

Refleksi

Setelah mempelajari dan mempraktikkan materi pada slide ini anda diharapkan dapat:

1. Mampu melakukan operasi-operasi dasar yang melibatkan list
2. Menggunakan list dalam persoalan-persoalan pemrograman yang anda jumpai