

# Dasar-Dasar Pemrograman 1 Gasal 2024/2025

## Lab 06 Sesi 2: List, Tuples, and Mutability



FAKULTAS  
ILMU  
KOMPUTER

**Deadline: Selasa, 29 Oktober 2024, pukul 16:20 WIB (80 menit)**

### Komponen yang Diuji:

1. Sub-CPMK 6: Mampu merancang *functions* untuk abstraksi dan modularisasi program
2. Sub-CPMK 7: Mampu memanipulasi *list, tuples, sets & dictionaries* untuk menyelesaikan permasalahan komputasi sederhana

### Riwayat Versi

Versi	Timestamp	Keterangan	Warna
1.	29-10-2024; 15:00 WIB	Rilis Pertama	-

# DekDepe Storage



*Illustrator: Shaney Zoya Fiandi 2023*

Dekdepe adalah seorang mahasiswa yang bekerja paruh waktu sebagai pengelola stok di sebuah *storage*. Untuk memudahkan pekerjaannya, ia ingin membuat sistem sederhana yang bisa membantu proses pengelolaan inventaris barang di gudangnya. Kamu diminta untuk membantu Dekdepe membuat beberapa fungsi sederhana agar ia bisa mengelola data inventaris barangnya dengan lebih mudah dan cepat.

## Ketentuan Program

Kamu diminta untuk membuat sebuah program utama, dua buah list **KOSONG** DI AWAL PROGRAM sebagai tempat menyimpan data barang pada *storage*, dan dua buah fungsi dengan detail sebagai berikut:

### 1. Program Utama

Program utama berfungsi untuk mencetak salam pembuka dan penutup, mengatur *flow* program, meminta *input* yang dibutuhkan dari user, memanggil fungsi yang sesuai, serta mengeluarkan *output* untuk diberikan pada *user*. Program utama **TIDAK BOLEH** menjalankan *logic* apapun selain menentukan menu mana yang akan dipilih oleh user dan meminta *input* yang sesuai.

## 2. Ketentuan List

Pada program, terdapat dua buah *list* yang **HARUS** diimplementasikan dengan ketentuan sebagai berikut:

Variabel	Penjelasan	Contoh
<i>List</i> barang	<i>List</i> ini berisi kumpulan barang barang yang terdapat di <i>storage</i> . <i>List</i> berisi <i>tuple</i> yang merupakan pasangan kode dengan nama barang.	<pre>barang = [("Code1", "ItemName1"), ("Code2", "ItemName2")]</pre>
<i>List</i> stok	<i>List</i> ini berisi stok dari barang-barang yang terdapat di <i>storage</i> . <i>List</i> berisi kumpulan <i>integer</i>	<pre>Stok = [StokItem1, StokItem2]</pre>

- Perhatikan juga bahwa item pada *list* *barang* memiliki stok yang tercatat pada **index yang sama** pada *list* *stok*. Sebagai contoh, *ItemName1* dengan kode *Code1* memiliki stok sebanyak *StockItem1* buah, sementara barang *ItemName2* dengan kode *Code2* memiliki stok sebanyak *StockItem2* buah.
- Kedua *list* haruslah **KOSONG** pada awal program berjalan. ("Code1", "ItemName1"), ("Code2", "ItemName2"), *StokItem1*, *StokItem2*, hanyalah **CONTOH** dan bukan merupakan barang sebenarnya yang berada pada *list*.
- Contoh implementasi pada fungsi:

```
1| barang = []      # Taruh list barang disini  
2| stok   = []      # Taruh list stok disini  
3|  
4| def    tambah_barang(<param1>, <param2>, ...):  
5| ... # Lanjutan kode
```

## 3. Fungsi tambah\_barang

- Fungsi ini bertujuan untuk menambahkan barang baru ke dalam *storage*.
- Parameter fungsi:
  - kode *string*
  - nama *string*
  - stok\_awal *integer*
- Fungsi **tidak** perlu me-*return* apapun.
- Parameter **kode** merepresentasikan **KODE** barang, nama adalah nama barang, dan *stok\_awal* adalah jumlah stok awal barang tersebut.
- Jika *kode* barang sudah ada di dalam daftar, **JANGAN** tambahkan barang baru dan tampilkan pesan **"GAGAL! Kode barang {kode} sudah ada di dalam sistem."**

- Jika BERHASIL, tambahkan tuple (id, nama) ke **list barang** dan tambahkan stok\_awal ke **list stok**. “**BERHASIL! Barang ({kode}, {nama}) berjumlah {stok\_awal} berhasil ditambahkan ke dalam storage.**”

#### 4. Fungsi kurangi\_stok

- Fungsi ini bertujuan untuk mengurangi stok dari barang berdasarkan `kode` barang.
- Parameter fungsi:
  - `kode` *string*
  - `jumlah` *integer*
- Fungsi **tidak** perlu me-return apapun.
- Parameter `kode` adalah kode barang yang ingin dikurangi stoknya, dan `jumlah` adalah jumlah stok yang akan dikurangi.
- Jika jumlah stok barang cukup, tampilkan pesan “**BERHASIL! Stok barang ID {kode} tersisa {stok\_baru}**”
- Jika jumlah yang dikurangi melebihi stok barang, tampilkan pesan “**GAGAL! Jumlah barang {kode} tidak cukup.**”
- Jika stok barang setelah dikurangi menjadi 0 setelah pengurangan, **hapus elemen** barang dan stok dari kedua list, serta tampilkan pesan “**BERHASIL! Barang I {kode} dihapus dari sistem.**”
- Jika kode barang tidak ditemukan, tampilkan pesan “**GAGAL! Kode barang tidak di dalam sistem.**”

#### 5. Pilihan Opsi pada Program(T/K/Q)

- T : **Tambah Barang**
- K : **Kurangi Stok Barang**
- Q : **Quit Program**

### Batasan

1. Opsi (T/K/Q) **dijamin T/K/Q**
2. Stok barang **dijamin integer**.
3. Pengurangan stok **dijamin integer**.
4. Mahasiswa **DILARANG** menggunakan data type selain:
  - a. Integer,
  - b. String,
  - c. Boolean,
  - d. List, dan
  - e. Tuple.

### Test Case

- **Merah** untuk input

- **Biru** untuk output

## Test Case 1:

```
Masukkan Opsi yang diinginkan (T/K/Q): T
Kode barang: L01
Nama barang: Lepi ovoneL
Stok barang: 5

BERHASIL! Barang (L01, Lepi ovoneL) berjumlah 5 berhasil
ditambahkan ke dalam storage.

Masukkan Opsi yang diinginkan (T/K/Q): T
Kode barang: L02
Nama barang: Lepi Acus
Stok barang: 3

BERHASIL! Barang (L02, Lepi Acus) berjumlah 3 berhasil
ditambahkan ke dalam storage.

Masukkan Opsi yang diinginkan (T/K/Q): K
Kode barang: L02
Jumlah barang yang ingin dikurangi: 2

BERHASIL! Stok barang ID L02 tersisa 1.

Masukkan Opsi yang diinginkan (T/K/Q): Q
Bye-Bye!
```

## Test Case 2:

```
Masukkan Opsi yang diinginkan (T/K/Q): K
Kode barang: T04
Jumlah barang yang ingin dikurangi: 999

GAGAL! Kode barang tidak di dalam sistem.

Masukkan Opsi yang diinginkan (T/K/Q): T
Kode barang: T06
Nama barang: Darksaber
Stok barang: 1

BERHASIL! Barang (T06, Darksaber) berjumlah 1 berhasil
ditambahkan ke dalam storage.

Masukkan Opsi yang diinginkan (T/K/Q): T
```

```
Kode barang: Z04
Nama barang: Darksaber
Stok barang: 2

BERHASIL! Barang (Z04, Darksaber) berjumlah 2 berhasil
ditambahkan ke dalam storage.

Masukkan Opsi yang diinginkan (T/K/Q): T
Kode barang: T06
Nama barang: Red Lightsaber
Stok barang: 1

GAGAL! Kode barang T06 sudah ada di dalam sistem.

Masukkan Opsi yang diinginkan (T/K/Q): K
Kode barang: T06
Jumlah barang yang ingin dikurangi: 2

GAGAL! Jumlah barang T06 tidak cukup.

Masukkan Opsi yang diinginkan (T/K/Q): K
Kode barang: T06
Jumlah barang yang ingin dikurangi: 1

BERHASIL! Barang ID T06 dihapus dari sistem.

Masukkan Opsi yang diinginkan (T/K/Q): K
Kode barang: T06
Jumlah barang yang ingin dikurangi: 1

GAGAL! Kode barang tidak di dalam sistem.

Masukkan Opsi yang diinginkan (T/K/Q): T
Kode barang: T06
Nama barang: Darksaber
Stok barang: 18

BERHASIL! Barang (T06, Darksaber) berjumlah 18 berhasil
ditambahkan ke dalam storage.

Masukkan Opsi yang diinginkan (T/K/Q): Q
Bye-Bye!
```

## Ketentuan Umum Program

1. **Dilarang melakukan kerja sama, menyontek, dan tindakan kecurangan lainnya** dengan sesama mahasiswa/i maupun dengan menggunakan **artificial intelligence** dalam mengerjakan lab.
2. Penamaan variabel harus mengikuti Python Naming Convention ([referensi](#)).

3. Penamaan modul, class, method, dan variabel harus jelas dan tidak ambigu ([referensi](#)).
4. Dokumentasikan kode menggunakan *comment*.
5. Deadline pengumpulan adalah pukul **16:20**. Tempat pengumpulan akan ditutup pukul **16:40**. Setelah periode tersebut, submisi tidak akan diterima.

## Komponen Penilaian

Komponen penilaian dapat dilihat pada tautan [ini](#).

## Berkas yang Perlu Dikumpulkan

- lab06.py

Kumpulkan berkas lab06.py yang telah di-zip dengan format penamaan seperti berikut.

[KodeAsdos]\_[Kelas]\_[NPM]\_[NamaLengkap]\_Lab06.zip

Contoh:

SCAR\_C\_2406987612\_OscarRyandaPutra\_Lab06.zip

🤖 Gud Lak dan semangat!! 🤖