

LAPORAN TUBES
PEMROGRAMAN 1
KELOMPOK CHUNCHUNMARU



Oleh:

Putra Strata Tandika Setyawan (2311104050)

Dimastian Aji Wibowo (2311104058)

Zhafir Zaidan Avail (2311104059)

Aulia Jasifa Br Ginting (2311104060)

Alya Rabani (2311104076)

PRODI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024

Tujuan

Tujuan dari praktikum ini adalah untuk mengimplementasikan sebuah program Python yang mampu mengelola daftar belanjaan. Program ini memungkinkan pengguna untuk menambahkan, mencari, menghapus, menampilkan, serta menghitung jumlah dan total harga item dalam daftar belanjaan.

Langkah Kerja

1. Inisialisasi Variabel dan Struktur Data

- Buat dua variabel, `daftar_belanjaan` sebagai list untuk menyimpan nama item, dan `daftar_harga` sebagai dictionary untuk menyimpan harga dari setiap item.

```
```python
daftar_belanjaan = []
daftar_harga = { }
```
```

Penjelasan Fungsi:

- `daftar_belanjaan = []` variabel tersebut diinisialisasi sebagai sebuah daftar kosong yang nantinya akan berisi item-item yang akan dibeli.
- `daftar_harga = { }` Variabel tersebut diinisialisasi sebagai kamus kosong yang digunakan untuk menyimpan harga dari setiap item dalam daftar belanjaan

2. Fungsi untuk Menambahkan Item ke Daftar Belanjaan

- Fungsi `tambah_item` menerima input nama item yang dipisahkan oleh koma, kemudian meminta input harga untuk setiap item. Item yang valid akan ditambahkan ke `daftar_belanjaan` dan `daftar_harga`.

```
```python
def tambah_item(items):
 new_items = items.split(",")
 for item in new_items:
 item = item.strip().capitalize()
 if item:
 harga = input_ulang(f"Masukkan harga untuk {item}: ")
```

```

while not harga.isdigit():
 print("Harga harus berupa angka. Silakan coba lagi.")
 harga = input_ulang(f"Masukkan harga untuk {item}: ")
daftar_belanjaan.append(item)
daftar_harga[item] = int(harga)
print(f'{item} telah ditambahkan ke daftar belanjaan dengan harga
{harga}.')
 sortir_belanjaan()
'''

```

#### Penjelasan Fungsi:

Agar user dapat menambahkan item ke dalam program tersebut, maka dibuatlah fungsi untuk menambahkan item ke daftar belanjaan.

- 'def tambah\_item(items):' Digunakan def untuk mendeklarasikan fungsi 'tambah item'. Fungsi ini akan menerima satu parameter yaitu 'items' yang merupakan string berisi item-item yang akan ditambahkan ke dalam daftar belanjaan.
- 'new\_items = items.split(","):' Baris ini mengambil string 'item' dan membaginya beberapa bagian berdasarkan koma (',').
- 'for item in new\_items:' yang menjadi loop yang akan mengeksekusi serangkaian pernyataan untuk setiap item dalam daftar 'new\_items'.
- 'Item = item.strip().capitalize()'. Di dalam loop tersebut, setiap item diproses. 'strip()' digunakan untuk menghapus spasi putih tambahan di awal dan akhir setiap item. Kemudian 'capitalize()' digunakan untuk mengubah huruf pertama setiap item menjadi huruf kapital. Hal ini dilakukan untuk memastikan bahwa format item konsisten, sehingga tidak ada item yang memiliki spasi ekstra atau huruf kecil di awalnya.
- 'If item'. Dibuat sebuah kondisi yang memeriksa apakah 'item' memiliki nilai yang tidak kosong. Jadi, jika pengguna memasukkan item kosong, maka kondisi ini akan bernilai false dan item tersebut tidak akan ditambahkan ke dalam daftar belanjaan.
- 'Harga = input\_ulang(f"Masukkan harga untuk {item}: ")'. Proses selanjutnya adalah meminta pengguna untuk memasukkan harga untuk item tertentu. Format f string yang digunakan memungkinkan kita untuk menyisipkan nilai variabel ke dalam string, sehingga pengguna tahu item mana yang akan mereka masukkan harganya.
- 'while not harga.isdigit():' lalu dibuat perulangan yaitu, loop while yang akan terus berjalan selama input yang dimasukkan pengguna bukanlah angka. Dalam loop ini, pesan kesalahan akan dicetak dan pengguna diminta untuk memasukkan harga lagi
- 'daftar\_belanjaan.append(item)'. Setelah harga yang valid dimasukkan oleh pengguna, item tersebut ditambahkan ke dalam daftar belanjaan 'daftar\_belanjaan'.
- 'Daftar\_harga[item] = int(harga)'. Pada proses selanjutnya, ini mengaitkan harga yang dimasukkan pengguna dengan item tersebut dalam kamus 'daftar\_harga'. Harga diubah menjadi integer sebelum disimpan dalam kamus.
- 'print(f'{item} telah ditambahkan ke daftar belanjaan dengan harga {harga}.')'. Lalu, cetak pesan mengkonfirmasi bahwa item telah ditambahkan ke daftar belanjaan bersama dengan harga yang telah dimasukkan pengguna.

- Setelah selesai menambahkan semua item dan harganya, panggil kembali fungsi 'sortir\_belanjaan()' tadi. Fungsi ini bertugas untuk mengurutkan daftar belanjaan sesuai dengan suatu kriteria tertentu.

### 3. Fungsi untuk Mencari Item dalam Daftar Belanjaan

- Fungsi 'cari\_item' mencari item dalam daftar berdasarkan nama yang diberikan dan menampilkan harga item tersebut jika ditemukan.

```
```python
def cari_item(item):
    item = item.capitalize()
    if item in daftar_belanjaan:
        print(f'{item} ada dalam daftar belanjaan dengan harga
{daftar_harga[item]}'.")
    else:
        print(f'{item} tidak ada dalam daftar belanjaan.')
```
```

#### Penjelasan Fungsi:

- 'Def cari\_item(item):'. Digunakan pendefinisian untuk mendeklarasikan fungsi 'cari\_item' lalu fungsi ini akan menerima parameter 'item' yang merupakan item yang ingin dicari dalam daftar belanjaan
- 'Item = item.capitalize()'. Fungsi ini mengonversi huruf pertama item menjadi huruf kapital, untuk memastikan kesesuaian dalam pencarian item.
- 'If item in daftar\_belanjaan:' selanjutnya fungsi ini adalah kondisi untuk memeriksa apakah item telah diubah menjadi huruf besar dalam daftar belanjaan.
- 'print(f'{item} ada dalam daftar belanjaan dengan harga {daftar\_harga[item]}'.')'. Jika item ada dalam daftar belanjaan maka akan dicetak pesan seperti pada baris tersebut. Pesan ini mencakup nama item dan harganya, yang diambil dari 'daftar\_harga' menggunakan item sebagai kunci.
- 'Else:'. Jika lainnya, maka akan dicetak: 'print(f'{item} tidak ada dalam daftar belanjaan.')'. Pesan tersebut memberi tahu pengguna bahwa item yang dicari tidak tersedia dalam daftar belanjaan.

### 4. Fungsi untuk Menghapus Item dari Daftar Belanjaan

- Fungsi 'hapus\_item' menghapus item dari 'daftar\_belanjaan' dan 'daftar\_harga' jika ditemukan.

```
```python
```

```
def hapus_item(item):
    item = item.capitalize()
    if item in daftar_belanjaan:
        daftar_belanjaan.remove(item)
        del daftar_harga[item]
        print(f'{item} telah dihapus dari daftar belanjaan.')
    else:
        print(f'{item} tidak ada dalam daftar belanjaan.')
    ...
```

Penjelasan Fungsi:

- 'def hapus_item(item):'. Digunakan pendefinisian untuk mendeklarasikan fungsi 'hapus_item' dan fungsi ini akan menerima parameter 'item' yang merupakan item yang ingin dihapus dari daftar belanjaan
- 'item = item.capitalize()' fungsi ini mengubah item menjadi format yang kapitalisasi dengan huruf pertama besar. Ini dilakukan agar item yang dicari dengan konsistensi, terlepas dari bagaimana item tersebut dimasukkan saat fungsi dipanggil.
- 'if item in daftar_belanjaan:'. Lalu, terdapat kondisional if yang memeriksa apakah item yang telah dibuat menjadi huruf kapital ada dalam daftar belanjaan.
- 'daftar_belanjaan.remove(item)'. Jika item ada dalam daftar belanjaan, maka akan otomatis terhapus dari daftar menggunakan metode 'remove()'.
- 'del daftar_harga[item]'. Setelah menghapus item dari daftar belanjaan, digunakan pernyataan 'del' yang digunakan untuk menghapus informasi harga terkait dengan item dari 'daftar_harga'.
- 'print(f'{item} telah dihapus dari daftar belanjaan.')'. Jika item berhasil dihapus dari daftar belanjaan, maka cetak pesan tersebut agar pengguna tahu bahwa item tersebut telah dihapus.
- 'Else:'. Jika lainnya, maka akan dicetak 'print(f'{item} tidak ada dalam daftar belanjaan.')'. Pesan ini memberi tahu pengguna bahwa item yang ingin dihapus tidak ada dalam daftar belanjaan.

5. Fungsi untuk Menghapus Semua Item dari Daftar Belanjaan

- Fungsi 'hapus_semua_item' menghapus semua item dari 'daftar_belanjaan' dan 'daftar_harga'.

```
```python
def hapus_semua_item():
 daftar_belanjaan.clear()
 daftar_harga.clear()
```

```

 print("Semua item telah dihapus dari daftar belanjaan.")
 ...

```

Penjelasan Fungsi:

- 'def hapus\_semua\_item()'. Dibuat pendefinisian untuk mendeklarasikan fungsi 'hapus\_semua\_item' untuk menghapus semua item dari daftar belanjaan dan daftar harga.
- 'daftar\_belanjaan.clear()'. Dengan menggunakan metode 'clear()' untuk membuat daftar belanjaan menjadi kosong.
- 'daftar\_harga.clear()'. Digunakan lagi metode 'clear()' untuk menghapus semua informasi harga yang terkait dengan item-item yang ada dalam daftar belanjaan
- 'print("Semua item telah dihapus dari daftar belanjaan.")'. Setelah menghapus semua item, maka cetak pesan tersebut untuk memberi tahu pengguna bahwa semua item telah dihapus dari daftar belanjaan.

## 6. Fungsi untuk Menampilkan Daftar Belanjaan

- Fungsi 'tampilkan\_daftar' menampilkan semua item beserta harga yang ada dalam daftar belanjaan.

```

```python
def tampilkan_daftar():
    if not daftar_belanjaan:
        print("Daftar belanjaan kosong.")
    else:
        print("Daftar Belanjaan:")
        for item in daftar_belanjaan:
            print(f"- {item}: {daftar_harga[item]}")
    ...

```

Penjelasan Fungsi:

- 'def tampilkan_daftar()'. Digunakan pendefinisian untuk mendeklarasikan fungsi 'tampilkan_daftar' lalu fungsi ini akan menampilkan 'daftar' yang merupakan daftar yang ingin dicari pada daftar belanjaan
- 'if not daftar_belanjaan:'. Fungsi variabel ini untuk memeriksa apakah daftar_belanjaan kosong. Jika kosong, maka kondisi ini benar
- 'print("Daftar Belanjaan kosong.")'. Jika kosong, akan dicetak pesan 'Daftar belanjaan kosong'. Menunjukkan bahwa daftar tersebut kosong.
- 'else:'. Jika tidak kosong, kode di dalam akan dieksekusi. 'daftar_belanjaan' 'else'

- `Print("Daftar Belanjaan:")`. Baris ini mencetak header "Daftar Belanjaan:" untuk menunjukkan awal daftar
- `'for item in daftar_belanjaan:'`. Baris ini memulai perulangan yang mengulangi setiap item di 'daftar_belanjaan'
- `'print(f"- {item}: {daftar_harga[item]}")'`. Baris ini mencetak setiap item beserta harga yang sesuai dari performatan digunakan untuk memasukkan nilai dan ke dalam string. Symbol digunakan untuk menunjukkan item daftar 'daftar_belanjaan' 'daftar_harga' 'f-string' 'item' 'daftar_harga[item]'

7. Fungsi untuk Mengurutkan Daftar Belanjaan

- Fungsi `'sortir_belanjaan'` mengurutkan item dalam daftar belanjaan secara alfabetis.

```
```python
def sortir_belanjaan():
 daftar_belanjaan.sort()
```
```

Penjelasan Fungsi:

- `'def sorting_belanjaan()'`. Digunakan pendefinisian untuk menyortir daftar belanjaan
- `'daftar_belanjaan.sort()'`. Baris kode ini menggunakan metode `sort()` pada daftar bernama `daftar_belanjaan`. Metode `sort()` akan mengurutkan item-item dalam daftar secara default berdasarkan urutan abjad.

8. Fungsi untuk Menghitung Jumlah Item dalam Daftar Belanjaan

- Fungsi `'hitung_jumlah_item'` menghitung dan menampilkan jumlah total item dalam daftar belanjaan.

```
```python
def hitung_jumlah_item():
 jumlah_item = len(daftar_belanjaan)
 print(f"Jumlah total item dalam daftar belanjaan adalah {jumlah_item}.")
```
```

Penjelasan Fungsi:

- `'def hitung_jumlah_item():'`. Digunakan pendefinisian untuk mendeklarasikan fungsi `'hitung_jumlah_item'` lalu fungsi ini akan menampilkan `'jumlah_item'` yang merupakan pada daftar belanjaan
- `'jumlah_item = len(daftar_belanjaan)'`. Baris kode ini menggunakan fungsi `len()` untuk menghitung jumlah item dalam daftar bernama `'daftar_belanjaan'` Hasil perhitungan disimpan dalam variabel `jumlah_item`
- `'print(f"Jumlah total item dalam daftar belanjaan adalah {jumlah_item}.")'`. baris kode ini merupakan pernyataan untuk mencetak pesan ke konsol. Pesan yang ditampilkan adalah "Jumlah total item dalam daftar belanjaan adalah" diikuti dengan nilai variabel `jumlah_item`. Nilai variabel diformat menggunakan f-string untuk memasukkan ke dalam string. Secara keseluruhan, fungsi ini menghitung jumlah item dalam daftar `'daftar_belanjaan'` dan kemudian menampilkan hasilnya ke konsol

9. Fungsi untuk Menghitung Total Harga Semua Item dalam Daftar Belanjaan

- Fungsi `'hitung_total_harga'` menghitung dan menampilkan total harga dari semua item dalam daftar belanjaan.

```
```python
def hitung_total_harga():
 total_harga = sum(daftar_harga[item] for item in daftar_belanjaan)
 print(f"Total harga semua item dalam daftar belanjaan adalah {total_harga}.")
```
```

Penjelasan Fungsi:

- `'def hitung_total_harga():'`. Digunakan pendefinisian untuk mendeklarasikan fungsi `'hitung_total_harga'` lalu fungsi ini akan menampilkan `'total_harga'` dengan tujuan mengetahui berapa total harga pada setiap item daftar belanjaan
- `'total_harga = sum(daftar_harga[item] for item in daftar_belanjaan)'`. Fungsi `'sum()'` pada baris kode ini untuk menghitung total harga semua item dalam daftar belanjaan serta menjumlahkan semua nilai yang dihasilkan dari ekspresi `daftar_belanjaan[item]` untuk setiap item dalam daftar. Fungsi `'daftar_harga[item]'` untuk mengakses harga item pada indeks yang ditentukan oleh variabel `item`. Fungsi `'for item in daftar_belanjaan'` berulang melalui setiap item dalam daftar `daftar_belanjaan[item]`
- Lalu, cetak pesan `'total harga semua item dalam daftar belanjaan adalah'` dengan menggunakan f string lalu panggil variabel `'total_harga'`

10. Fungsi untuk Meminta Input Ulang Jika Input Salah

- Fungsi `'input_ulang'` memastikan bahwa input yang diberikan pengguna tidak kosong.


```

```python
def input_ulang(prompt):
 while True:
 data = input(prompt)
 if data.strip():
 return data
 else:
 print("Input tidak valid. Silakan coba lagi.")
```

```

Penjelasan Fungsi:

- 'def input_ulang(prompt):'. Fungsi ini di gunakan untuk meminta input dari pengguna berulang kali sampai pengguna memasukkan input yang valid
- 'while True:'. Menciptakan loop tak terbatas yang akan terus berjalan sampai ada pernyataan break atau return yang dieksekusi
- 'data = input(prompt)'. Pada baris kode ini menggunakan fungsi input() untuk menampilkan prompt ke pengguna dan menerima input dari pengguna. Prompt yang ditampilkan adalah nilai dari parameter prompt. Input pengguna disimpan dalam variabel data.
- 'if data.strip():'. Pernyataan if ini memeriksa apakah variabel data berisi nilai yang tidak kosong. Metode strip() digunakan untuk menghapus spasi putih di awal dan akhir string data. Jika data.strip() bernilai True, maka string data tidak kosong.
- 'return data'. Pernyataan return data mengembalikan nilai variabel data dari fungsi. Ini berarti fungsi akan berhenti dan mengembalikan nilai input pengguna kepada kode yang memanggil fungsi input_ulang()
- 'else:'. Blok else dieksekusi jika kondisi if bernilai False, yaitu jika variabel data kosong.
- 'print("Input tidak valid. Silakan coba lagi.)'. Baris kode ini mencetak pesan bahwa inputan yang diberikan tidak valid, dipersilahkan untuk mencoba lagi.

11. Fungsi Utama

- Fungsi `main` menyediakan antarmuka menu untuk pengguna agar dapat memilih aksi yang diinginkan seperti menambah, menghapus, mencari, menampilkan, menghitung item dan harga, serta keluar dari program.

```

```python
def main():

```

```

while True:
 print("\nPilihan:")
 print("1. Tambah item")
 print("2. Hapus item")
 print("3. Cari item")
 print("4. Tampilkan daftar belanjaan")
 print("5. Hapus semua item dari daftar belanjaan")
 print("6. Hitung jumlah total item")
 print("7. Hitung total harga")
 print("8. Keluar")

 pilihan = input_ulang("Masukkan pilihan Anda: ")

 if pilihan == "1":
 items = input_ulang("Masukkan nama item (pisahkan dengan koma untuk
menambahkan beberapa item sekaligus): ")
 tambah_item(items)
 elif pilihan == "2":
 item = input_ulang("Masukkan nama item yang ingin dihapus: ")
 hapus_item(item)
 elif pilihan == "3":
 item = input_ulang("Masukkan nama item yang ingin dicari: ")
 cari_item(item)
 elif pilihan == "4":
 tampilkan_daftar()
 elif pilihan == "5":
 hapus_semua_item()
 elif pilihan == "6":
 hitung_jumlah_item()
 elif pilihan == "7":
 hitung_total_harga()
 elif pilihan == "8":
 print("Terima kasih, sampai jumpa!")
 break

```

```

else:
 print("Pilihan tidak valid. Silakan coba lagi.")
'''

```

#### Penjelasan Fungsi:

- 'def main():'. Dibuat sebuah pendefinisian tanpa parameter yang merupakan titik awal dari eksekusi program.
- 'while true:'. Menciptakan loop tak terbatas yang akan terus berjalan hingga diberi perintah untuk berhenti ('break')
- 'print("")'. Statements tersebut digunakan untuk menampilkan menu pilihan kepada user, agar user dapat memilihnya.
- 'pilihan = input\_ulang'. Fungsi ini digunakan untuk meminta input dari pengguna. Nama fungsi menyiratkan bahwa mungkin ada validasi atau pengulangan permintaan input hingga input yang valid diterima
- 'if'. Statement digunakan untuk memeriksa pilihan pengguna dan memanggil fungsi yang sesuai berdasarkan pilihan tersebut
  - Pilihan 1: menambah item ke daftar belanjaan
  - Pilihan 2: menghapus item dari daftar belanjaan
  - Pilihan 3: mencari item dalam daftar belanjaan
  - Pilihan 4: menampilkan daftar belanjaan
  - Pilihan 5: menghapus semua item dari daftar belanjaan
  - Pilihan 6: menghitung jumlah total item dalam daftar belanjaan
  - Pilihan 7: menghitung total harga dari semua item dalam daftar belanjaan
  - Pilihan 8: keluar dari program
- 'break'. Digunakan untuk keluar dari loop dan mengakhiri program ketika pilihan 8 dipilih.
- 'else'. Menangani input yang tidak valid dengan mencetak pesan 'print("pilihan tidak valid. Silahkan coba lagi.")' agar user mengetahui dan mencoba lagi.

## Hasil dan Pembahasan

Setelah implementasi, program berhasil dijalankan dan dapat melakukan semua fungsi yang telah direncanakan. Berikut adalah beberapa contoh hasil eksekusi program:

### 1. Menambah item ke daftar belanjaan:

...

Masukkan nama item: apel, jeruk

Masukkan harga untuk Apel: 5000

Apel telah ditambahkan ke daftar belanjaan dengan harga 5000.

Masukkan harga untuk Jeruk: 7000

Jeruk telah ditambahkan ke daftar belanjaan dengan harga 7000.

...

### 2. Mencari item dalam daftar belanjaan:

...

Masukkan nama item yang ingin dicari: apel

Apel ada dalam daftar belanjaan dengan harga 5000.

...

### 3. Menghapus item dari daftar belanjaan:

...

Masukkan nama item yang ingin dihapus: jeruk

Jeruk telah dihapus dari daftar belanjaan.

...

### 4. Menampilkan daftar belanjaan:

...

Daftar Belanjaan:

- Apel: 5000

...

### 5. Menghitung jumlah item dan total harga:

...

Jumlah total item dalam daftar belanjaan adalah 1.

Total harga semua item dalam daftar belanjaan adalah 5000.

...