**Lembar Laporan UAS**

Dasar-dasar Pemrograman

Ricky Setiadi S.Kom, M.Kom, CCISO, CEH, CTIA, CCSE, ISO 27001 Lead Auditor



**Program Studi Teknik Informatika**

Semester Ganjil Tahun Ajaran 2024/2025

Tanggal Pengumpulan : 13 Maret 2025

Kelompok Array Mavericks :

Fadilah Balfas (241552010022)

M. Zaki Marpiansyah (241552010015)

Fitri Ambar Sari (241552010023)

Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Tazkia

Jl. Raya Dramaga Blok Radar Baru No.8, RT.03/RW.03, Margajaya, Kec. Bogor Baru., Kota Bogor, Jawa Barat 16116, Indonesia

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc192753019)

[Abstrak 3](#_Toc192753020)

[**1.** **Pendahuluan** 3](#_Toc192753021)

[1.1 Latar Belakang 3](#_Toc192753022)

[1.2 Tujuan 3](#_Toc192753023)

[1.3 Rumusan Masalah 3](#_Toc192753024)

[1.4 Manfaat 3](#_Toc192753025)

[1.5 Ruang Lingkup 4](#_Toc192753026)

[**2.** **Tinjauan Pustaka** 4](#_Toc192753027)

[**3.** **Metodologi** 4](#_Toc192753028)

[3.1 Analisa dan Desain 4](#_Toc192753029)

[3.2 Desain Sistem 4](#_Toc192753030)

[3.2.1 PseudoCode : 4](#_Toc192753031)

[3.2.2 Diagram Alir (Flowchart) 7](#_Toc192753032)

[3.3 Langkah Pengembangan 7](#_Toc192753033)

[**4.** **Hasil dan Pembahasan** 7](#_Toc192753034)

[4.1 Bahasa Pemrograman dan Tools 7](#_Toc192753035)

[4.2 Source Code / Program, 8](#_Toc192753036)

[4.2.1 Rekursi 8](#_Toc192753037)

[4.2.2 Percabangan (Conditional Statement) 10](#_Toc192753038)

[4.2.3 Array (List dalam Python) 11](#_Toc192753039)

[Kesimpulan 12](#_Toc192753040)

[4.3 Pembahasan 12](#_Toc192753041)

[4.3.1 Struktur Program 12](#_Toc192753042)

[a. Deklarasi Variabel Global 13](#_Toc192753043)

[4.3.2 Test Case 14](#_Toc192753044)

[4.3.3 Hasil Pengujian, 14](#_Toc192753045)

[**5.** **Kesimpulan** 15](#_Toc192753046)

[**6.** **Daftar Pustaka** 15](#_Toc192753047)

[**7.** **Lampiran** 15](#_Toc192753048)

# Abstrak

* Ringkasan singkat (100-150 kata) tentang proyek yang dibuat.
* Tujuan proyek.
* Teknologi atau konsep yang digunakan.
* Hasil yang dicapai.

1. **Pendahuluan**
2. Latar Belakang

Dalam kehidupan sehari-hari, pencatatan daftar belanja merupakan hal yang umum dilakukan untuk memastikan kebutuhan yang diperlukan dapat dibeli tanpa ada yang terlewat. Namun, pencatatan daftar belanja secara manual dapat menimbulkan berbagai permasalahan, seperti daftar yang tercecer, sulit diperbarui, dan kurang efisien. Oleh karena itu, diberikan solusi berbasis teknologi untuk membantu mengelola daftar belanja dengan lebih mudah dan terstruktur.

1. Tujuan

Tujuan utama dari project ini adalah untuk mengembangkan aplikasi daftar belanja berbasis python. Ini memudahkan pengguna untuk mengelola dan mengakses daftar belanja dan juga memudahkan pengguna untuk menambah, menghapus, dan melihat daftar belanja. Selain itu project ini juga bertujuan meningkatkan pemahaman tentang implementasi struktur data array. Selain itu juga penggunaan loop dan rekrusi dapat meningkatkan efisiensi pemrosesan data.

1. Rumusan Masalah
2. Bagaimana cara mengimplementasikan struktur data array dalam program?
3. Bagaimana penerapan rekursi dan perulangan dalam pemrogaman python?
4. Bagaimana cara mengoptimalkan pengelolaan daftar belanja agar lebih interaktif dan mudah digunakan oleh pengguna?
5. Manfaat

Dari sisi individu, mahasiswa mendapatkan pengalaman langsung dalam mengembangkan aplikasi berbasis Python mereka juga dapat meningkatkan keterampilan teknis dalam pemrograman, terutama dalam penggunaan struktur data seperti array, serta penerapan perulangan dan rekursi. Selain itu, proyek ini membantu mahasiswa dalam meningkatkan kemampuan pemecahan masalah, berpikir logis, serta memahami bagaimana cara membangun sebuah aplikasi yang efisien dan fungsional. Dari sisi akademik, proyek ini memberikan pemahaman yang lebih mendalam mengenai konsep dasar pemrograman yang telah dipelajari di kelas. Mahasiswa tidak hanya memahami teori tetapi juga mendapatkan kesempatan untuk menerapkannya dalam skenario nyata. Hal ini akan membantu mereka dalam menyelesaikan tugas-tugas akademik lainnya serta meningkatkan kesiapan mereka untuk menghadapi tantangan dalam dunia industri teknologi. aplikasi daftar belanja yang dikembangkan dalam proyek ini dapat digunakan dalam kehidupan sehari-hari. Dengan adanya aplikasi ini, pengguna dapat mencatat, mengelola, dan mengakses daftar belanja mereka dengan lebih mudah dan efisien. Proyek ini juga memberikan wawasan tentang bagaimana teknologi dapat dimanfaatkan untuk menyelesaikan masalah sederhana namun memiliki dampak besar dalam meningkatkan efisiensi dan produktivitas seperti peningkatan keterampilan teknis dalam pemrograman, pemahaman yang lebih baik terhadap struktur data dan algoritma dasar, dan pengalaman langsung dalam pengembangan perangkat lunak sederhana.

1. Ruang Lingkup
2. Penggunaan bahasa pemrogaman python.
3. Implementasi array,pengulangan dan rekursi.
4. Pengujian fungsi utama dalam program untuk memastikan validasi hasilnya. **[[1]](#footnote-1)**
5. **Tinjauan Pustaka**

Dalam buku Learning Python *oleh Mark Lutz (2013),* dijelaskan berbagai konsep fundamental dalam Python, termasuk penggunaan array dan struktur data lainnya. Buku ini menjadi acuan utama dalam memahami bagaimana data dapat dikelola secara efisien dalam Python serta bagaimana perulangan dan rekursi dapat digunakan untuk menyederhanakan kode program.

Selain itu, dokumentasi resmi Python (https://docs.python.org/3/) memberikan informasi yang komprehensif tentang berbagai fungsi bawaan Python, termasuk implementasi daftar (list), operasi pada array, serta metode manipulasi data seperti pencarian dan pengurutan. Dokumentasi ini menjadi sumber utama dalam memahami sintaksis serta praktik terbaik dalam menulis kode yang optimal dan efisien.

1. **Metodologi**
2. Analisa dan Desain
3. Desain Sistem
   * 1. PseudoCode :

Mulai

Inisialisasi daftar\_belanja sebagai daftar kosong

Definisi fungsi tambah\_item():

Minta pengguna memasukkan nama item

Tambahkan item ke daftar\_belanja

Cetak pesan bahwa item telah ditambahkan

Definisi fungsi hapus\_item():

Minta pengguna memasukkan nama item yang akan dihapus

Jika item ada di daftar\_belanja:

Hapus item dari daftar\_belanja

Cetak pesan bahwa item telah dihapus

Jika tidak:

Cetak pesan bahwa item tidak ditemukan

Definisi fungsi tampilkan\_daftar():

Jika daftar\_belanja kosong:

Cetak pesan bahwa daftar kosong

Jika tidak:

Cetak "Daftar Belanja:"

Untuk setiap item dalam daftar\_belanja:

Cetak nomor urut dan nama item

Definisi fungsi tampilkan\_daftar\_terbalik(index):

Jika index belum ditentukan:

Atur index ke panjang daftar\_belanja - 1

Jika index < 0:

Kembalikan (keluar dari fungsi)

Cetak nomor urut dan nama item pada index

Panggil kembali fungsi tampilkan\_daftar\_terbalik(index - 1)

Selama program berjalan:

Tampilkan menu:

1. Tambah Item

2. Hapus Item

3. Tampilkan Daftar

4. Tampilkan Daftar (Terbalik)

5. Keluar

Minta pengguna memilih menu

Jika pilihan adalah "1":

Panggil fungsi tambah\_item()

Jika pilihan adalah "2":

Panggil fungsi hapus\_item()

Jika pilihan adalah "3":

Panggil fungsi tampilkan\_daftar()

Jika pilihan adalah "4":

Panggil fungsi tampilkan\_daftar\_terbalik()

Jika pilihan adalah "5":

Cetak pesan keluar dari program

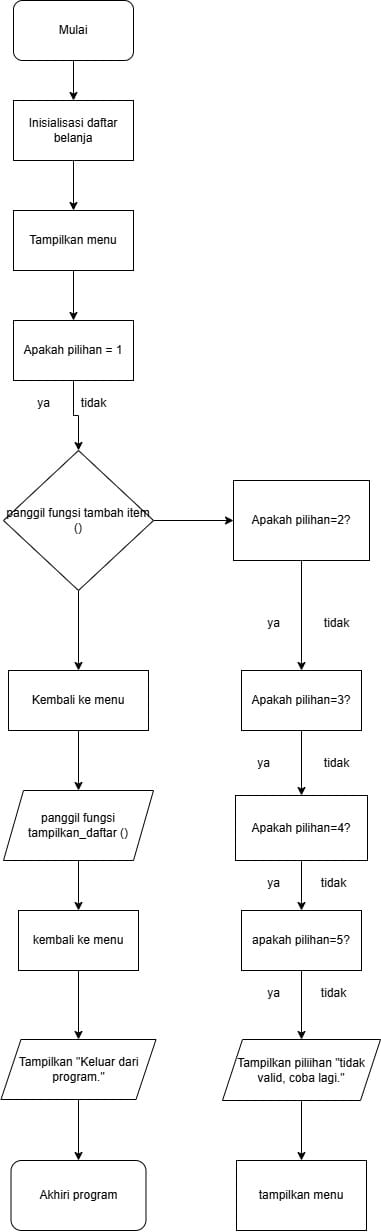
Hentikan program

Jika pilihan tidak valid:

Cetak pesan kesalahan

Selesai

* + 1. Diagram Alir (Flowchart)



1. Langkah Pengembangan

***berisi narasi proses pengembangan aplikasi mulai dari analisa kebutuhan, hipotesa, design, pengembangan, pengujian dan lain sebagainya. Gunakan diagram atau alat bantu lain yang dibutuhkan dalam penjelasan proses pengembangannya.***

1. **Hasil dan Pembahasan**
2. Bahasa Pemrograman dan Tools

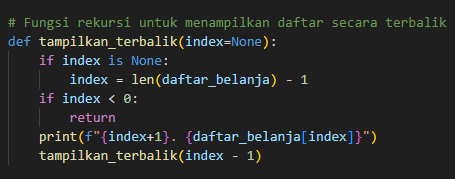
* + 1. Python : digunakan karena sintaksnya yang sederhana dan fleksibel
    2. Visual Studio Code : Digunakan untuk menginput code
    3. Draw.io : Digunakan untuk menggambar diagram alir (flowchart)

1. Source Code / Program,

### **Rekursi**

Rekursi adalah fungsi yang memanggil dirinya sendiri. Dalam program ini, rekursi digunakan pada fungsi tampilkan\_terbalik() untuk menampilkan daftar belanja secara terbalik.

Kode rekursi dalam program:

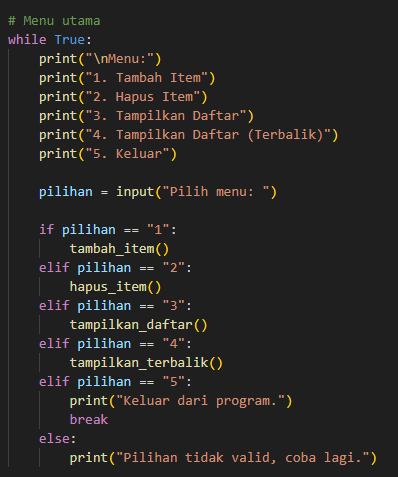


Penjelasan:

1. Jika index belum ditentukan (None), maka diberikan nilai indeks terakhir dari daftar\_belanja.
2. Jika index < 0, maka rekursi berhenti (return).
3. Cetak item pada indeks tertentu.
4. Panggil tampilkan\_terbalik() lagi dengan index - 1, sehingga berulang hingga mencapai indeks 0.

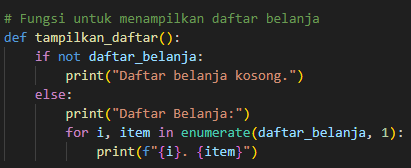
Perulangan digunakan untuk menjalankan kode berulang kali. Dalam program ini, ada dua jenis perulangan:

1. Perulangan dalam menampilkan daftar belanja (menggunakan for)
2. Perulangan dalam menu utama (menggunakan while True)
   1. Kode perulangan dalam menu utama:

**

Penjelasan:

* While True: membuat program terus berjalan sampai pengguna memilih keluar (break).
* Setiap iterasi, program menampilkan menu dan meminta input dari pengguna.
* Program terus berjalan sampai pengguna memilih "5" untuk keluar.
  1. Kode perulangan dalam menampilkan daftar belanja:

**

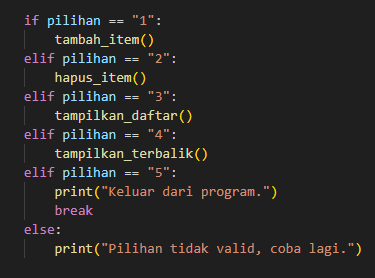
Penjelasan:

* for i, item in enumerate(daftar\_belanja, 1): mengulang setiap item dalam daftar belanja dan mencetaknya dengan nomor urut.

### **Percabangan (Conditional Statement)**

Percabangan adalah kode yang memungkinkan program memilih jalur eksekusi yang berbeda berdasarkan suatu kondisi (if-elif-else).

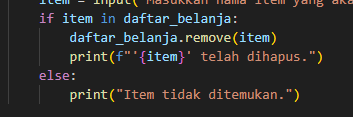
1. Contoh percabangan dalam program:

**

Penjelasan:

* Jika pengguna memilih "1", program memanggil fungsi tambah\_item().
* Jika "2", program memanggil hapus\_item(), dan seterusnya.
* Jika input bukan "1", "2", "3", "4", atau "5", program menampilkan pesan kesalahan "Pilihan tidak valid, coba lagi.".

1. Contoh percabangan lain dalam hapus\_item():



Penjelasan:

* Jika item ditemukan di daftar\_belanja, item tersebut dihapus.
* Jika tidak, program menampilkan "Item tidak ditemukan.".

### **Array (List dalam Python)**

Array atau list digunakan untuk menyimpan beberapa data dalam satu variabel. Dalam program ini, daftar belanja disimpan dalam daftar\_belanja, yang merupakan sebuah list.

1. Deklarasi list dalam program:



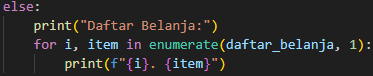
1. Penambahan item ke list dalam tambah\_item():



1. Penghapusan item dari list dalam hapus\_item():



1. Menampilkan isi list dalam tampilkan\_daftar():



Penjelasan:

* daftar\_belanja.append(item) menambah item ke daftar belanja.
* daftar\_belanja.remove(item) menghapus item dari daftar belanja.
* for i, item in enumerate(daftar\_belanja, 1): digunakan untuk menampilkan semua item dalam daftar.

**5. Input dari User**

Program menerima input dari pengguna menggunakan fungsi input().

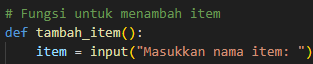
**Contoh input dalam program:**



Penjelasan:

* Program meminta pengguna memilih menu dan menyimpan input dalam variabel pilihan.

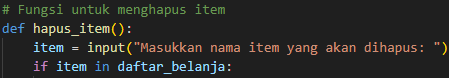
Input dalam tambah\_item():



Penjelasan:

* Program meminta pengguna memasukkan nama item yang ingin ditambahkan.

Input dalam hapus\_item():



Penjelasan:

* Program meminta pengguna memasukkan nama item yang ingin dihapus.

### **Kesimpulan**

Program ini menggunakan beberapa konsep pemrograman dasar:  
 **Rekursi** dalam tampilkan\_terbalik()  
 **Perulangan** while untuk menu utama dan for untuk daftar belanja  
 **Percabangan** if-elif-else untuk menentukan jalur eksekusi program  
 **Array (List dalam Python)** untuk menyimpan daftar belanja  
 **Input dari user dengan** input() untuk menerima data dari pengguna

1. Pembahasan

### **4.3.1 Struktur Program**

Program ini terdiri dari beberapa bagian utama yang membentuk keseluruhan alur kerja aplikasi daftar belanja. Struktur programnya adalah sebagai berikut:

### **Deklarasi Variabel Global**

* daftar\_belanja = [] : Variabel global yang digunakan untuk menyimpan daftar item belanja.
  1. **Fungsi dan Prosedur yang Digunakan**

Program ini menggunakan lima fungsi utama:

1. **Tambah\_item()**

* Menerima input nama item dari pengguna dan menambahkannya ke daftar belanja.
* Menampilkan konfirmasi setelah item berhasil ditambahkan.

1. **Hapus\_item()**

* Menerima input nama item dari pengguna dan menghapusnya dari daftar belanja jika ditemukan.
* Menampilkan pesan kesalahan jika item tidak ditemukan.

1. **Tampilkan\_daftar()**

* Menampilkan seluruh item dalam daftar belanja dalam urutan penambahan.
* Jika daftar kosong, menampilkan pesan bahwa daftar belanja kosong.

1. **Tampilkan\_terbalik(index=None)** (Menggunakan Rekursi)

* Menampilkan daftar belanja dalam urutan terbalik.
* Menggunakan rekursi untuk mengakses elemen dari belakang ke depan.

1. **Menu Utama (while True)**

* Menampilkan menu interaktif.
* Memproses input dari pengguna untuk menjalankan fungsi yang sesuai.
  1. Proses Pemanggilan Fungsi
* Program dijalankan dalam loop utama (while True) yang terus menampilkan menu kepada pengguna.
* Berdasarkan input pengguna, fungsi yang sesuai akan dipanggil:

1. Jika pengguna memilih **1**, tambah\_item() dipanggil.
2. Jika pengguna memilih **2,** hapus\_item() dipanggil.
3. Jika pengguna memilih **3**, tampilkan\_daftar() dipanggil.
4. Jika pengguna memilih **4**, tampilkan\_terbalik() dipanggil.
5. Jika pengguna memilih **5**, program keluar dari loop dan berhenti.
   * 1. Test Case

|  |  |  |  |
| --- | --- | --- | --- |
| No | Fitur yang di uji | Skenario Pengujian | Metode Pengujian |
| 1. | Tambah item | Menambahkan item baru ke daftar | Input valid dan cek output |
| 2. | Tambah item (negatif) | Menambahkan item yang sama dua kali | Input duplikat dan cek output |
| 3. | Hapus item | Menghapus item yang ada dalam daftar | nput valid dan cek daftar |
| 4. | Hapus item (Negatif) | Menghapus item yang tidak ada | Input tidak valid dan cek pesan error |
| 5. | Tampilkan daftar | Menampilkan daftar setelah beberapa item ditambahkan | Periksa output |
| 6. | Tampilkan daftar kosong | Menampilkan daftar saat masih kosong | Periksa output |
| 7. | Tampilkan terbalik | Menampilkan daftar secara terbalik | Periksa urutan output |

* + 1. Hasil Pengujian,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | **Fitur yang Diuji** | Ekspetasi Awal | **Hasil Pengujian** | Kesimpulan |
| 1. | Tambah Item | Item ditambahkan ke daftar | Sesuai ekspektasi | Berhasil |
| 2. | Tambah Item (Negatif) | Mencegah duplikasi item | Tidak ada pencegahan duplikasi | Perlu perbaikan |
| 3. | Hapus Item | Item berhasil dihapus | Sesuai ekspektasi | Berhasil |
| 4. | Hapus Item (Negatif) | Menampilkan pesan error | Sesuai ekspektasi | Berhasil |
| 5. | Tampilkan Daftar | |  |  | | --- | --- | | Menampilkan daftar dengan benar |  | | Sesuai ekspektasi | Berhasil |
| 6. | Tampilkan Daftar kosong | Menampilkan pesan daftar kosong | Sesuai ekspektasi | Berhasil |
| 7. | Tampilkan Terbalik | Menampilkan daftar dengan urutan terbalik | Sesuai ekspektasi | Berhasil |

Perbandingan Hipotesisi Awal dengan Hasil Akhir

Hipotesis awal dari program ini adalah bahwa program dapat:

1. Menambah dan menghapus item dalam daftar belanja dengan benar.
2. Menampilkan daftar belanja secara normal dan terbalik.
3. Menjalankan menu interaktif tanpa error.
4. **Kesimpulan**

*Tuangkan ringkasan hasil proyek secara naratif termasuk kesimpulan terkait dengan informasi apakah proyek ini berhasil memenuhi tujuan awal atau tidak*

*Tuliskan juga saran dan rekomendasi sebagai bagian dari perbaikan untuk pengembangan proyek di masa depan serta saran untuk mahasiswa lain yang ingin mengerjakan proyek serupa.*

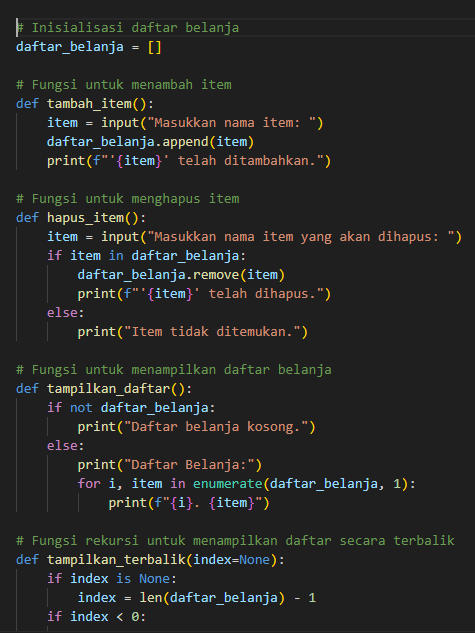
1. **Daftar Pustaka**

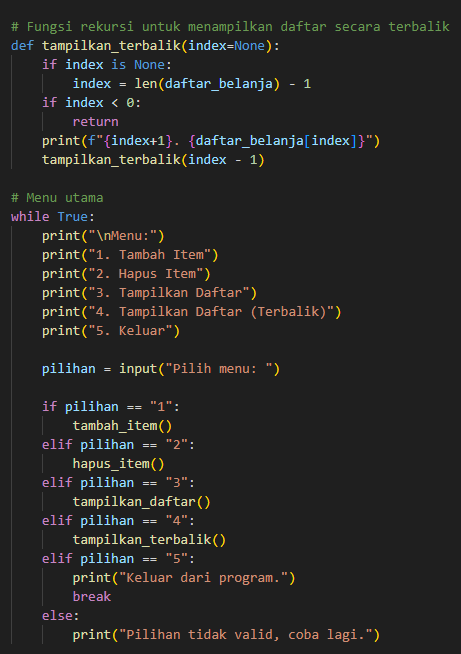
*Tuangkan sumber referensi yang digunakan selama pengerjaan project seperti :*

* *Buku*
* *Artikel*
* *Dokumentasi*
* *Video tutorial*

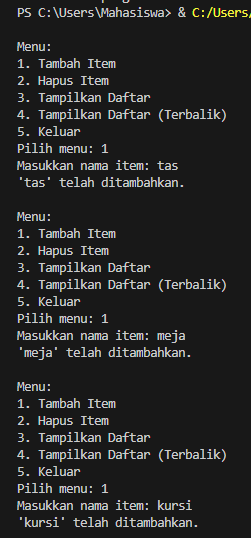
1. **Lampiran**

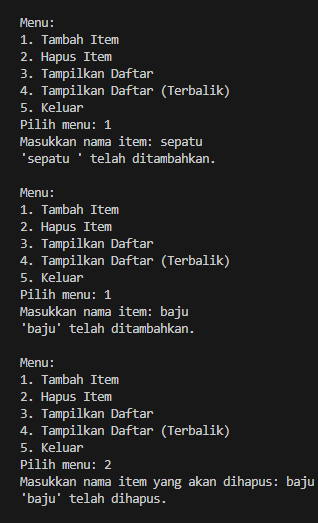
**Kode Program Lengkap**:

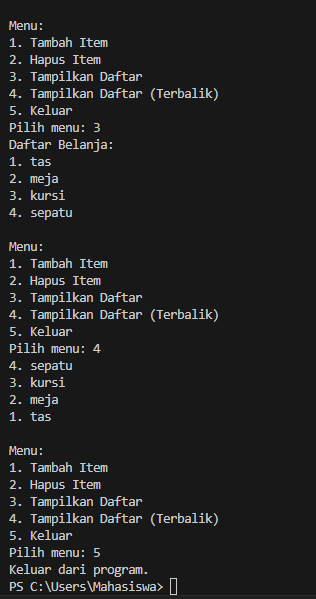




**File Tambahan**:







1. [↑](#footnote-ref-1)