

# **LAPORAN TUGAS BESAR**


## **IF2111 Algoritma dan Struktur Data STI**

### **PURRMART**

Dipersiapkan oleh:  
Kelompok 01:

Bagas Noor Fadhilah	(18223115)
Khairunnisa Azizah	(18223117)
Maria Vransiska P. C. T. D. P	(18223119)
Theresia Ivana M. S	(18223126)
Aulia Azka Azzahra	(18223131)

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung  
Jl. Ganesha 10, Bandung 40132

	<b>Sekolah Teknik Elektro dan Informatika ITB</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<i>IF2111-TB-03-01</i>		<i>44</i>
		<i>Revisi</i>	-	<i>20/11/2024</i>

# Daftar Isi

<b>1. Ringkasan.....</b>	<b>4</b>
<b>2. Penjelasan Tambahan Spesifikasi Tugas.....</b>	<b>4</b>
<b>3. Struktur Data (ADT).....</b>	<b>4</b>
3.1 ADT array.....	5
3.2 ADT arraydin.....	5
3.3 ADT boolean.....	5
3.4 ADT mesinkarakter.....	5
3.5 ADT mesinkata.....	5
3.6 ADT queue.....	6
3.7 ADT misc.....	6
3.8 ADT dplinkedlist.....	6
3.9 ADT map.....	7
3.10 ADT stack.....	7
<b>4. Program Utama.....</b>	<b>7</b>
<b>5. Algoritma-Algoritma Menarik.....</b>	<b>8</b>
5.1 Double Linked List.....	8
5.2 Stack.....	8
<b>6. Data Test.....</b>	<b>9</b>
6.1 START.....	9
6.2 LOAD.....	10
6.3 LOGIN.....	11
6.4 LOGOUT.....	12
6.5 REGISTER.....	13
6.6 WORK.....	14
6.7 WORK CHALLENGE.....	15
6.8 STORE LIST.....	20
6.9 STORE REQUEST.....	20
6.10 STORE SUPPLY.....	21
6.11 STORE REMOVE.....	23
6.12 HELP.....	24
6.13 SAVE.....	26
6.14 QUIT.....	26
6.15 PROFILE.....	27
6.16 CART ADD <nama> <n>.....	27
6.17 CART REMOVE <nama> <n>.....	28
6.18 CART SHOW.....	29

6.19 CART PAY.....	29
6.20 HISTORY <n>.....	31
6.21 WISHLIST ADD.....	32
6.22 WISHLIST SWAP <i> <j>.....	33
6.23 WISHLIST REMOVE <i>.....	33
6.24 WISHLIST REMOVE.....	34
6.25 WISHLIST CLEAR.....	35
6.26 WISHLIST SHOW.....	35
<b>7. Test Script.....</b>	<b>36</b>
<b>8. Pembagian Kerja dalam Kelompok.....</b>	<b>40</b>
<b>9. Lampiran.....</b>	<b>41</b>
9.1 Deskripsi Tugas Besar 2.....	41
9.2 Notulen Rapat.....	41
9.3 Log Activity Anggota Kelompok.....	42

## 1. Ringkasan

Persediaan senjata dan barang OWCA sudah mulai menipis setelah melalui pertarungan sengit melawan Dr. Asep Spakbor selama 3 bulan 13 hari 2 jam 47 menit dan 2 detik. OWCA pun menugaskan tim programmer andalannya untuk merancang aplikasi jual beli barang bernama PURRMART. PURRMART ini dibuat dengan tujuan agar dapat membantu Toko Borma, toko pemasok barang-barang perang, untuk melakukan sistem jual beli secara online.

PURRMART adalah sebuah *e-commerce* berbasis CLI (Command Line Interface) yang diimplementasikan menggunakan bahasa C yang dikembangkan untuk menyelesaikan tugas besar untuk mata kuliah IF2111 Algoritma dan Struktur Data. Fitur-fitur utama PURRMART adalah menampilkan barang toko, meminta dan menyuplai barang baru ke toko, menyimpan dan membeli barang dalam keranjang, menampilkan barang yang sudah dibeli, membuat dan menghapus *wishlist*, serta bekerja untuk menghasilkan uang. Implementasi ADT menjadi dasar dalam penerapan fitur-fitur tersebut.

Ketika program PURRMART pertama kali dijalankan, main menu akan ditampilkan dengan welcome menu dan beberapa perintah seperti START, LOAD, HELP, dan QUIT. Setelah itu, program akan memasuki login menu yang menyediakan command LOGIN REGISTER, dan HELP. Program dapat menerima input sesuai perintah hingga perintah QUIT diberikan.

Laporan ini berisi penjelasan secara rinci mengenai aplikasi *e-commerce* yang telah dibuat, beserta spesifikasi-spesifikasi yang terdiri dari fitur-fitur utama, implementasi ADT yang telah dimodifikasi, serta algoritma yang digunakan dalam aplikasi PURRMART. Pada bagian akhir laporan, dilampirkan hasil pengetesan program yang dilakukan untuk memastikan fungsionalitas aplikasi PURRMART. Dengan demikian, tugas besar ini membantu memahami dan mengimplementasikan ADT dalam konsep yang berbeda, yaitu sebagai sebuah aplikasi *e-commerce* berbasis *command-line*.

## 2. Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Bonus: Quantum WORDL3

Quantum WORDL3 adalah fitur tambahan yang menantang pemain untuk menebak empat kata berbeda, masing-masing terdiri dari 5 huruf, dalam satu waktu. Pemain diberi 9 kesempatan untuk menebak keempat kata target dengan benar, yang diambil secara acak dari daftar kata valid dan dijamin berbeda satu sama lain. Setiap tebakan harus dituliskan dalam satu baris, dipisahkan spasi, dan dianggap tidak valid jika terdapat kata duplikat, kurang dari empat kata, atau salah satu kata tidak berisi lima huruf. Setelah setiap tebakan, pemain menerima umpan balik visual untuk keempat kata, menandai huruf yang benar posisinya, huruf yang ada tetapi salah posisi, dan huruf yang tidak ada dalam kata target. Jika pemain berhasil menebak semua kata dengan benar, permainan langsung berakhir dengan kemenangan dan hadiah +10000 rupiah untuk akun pemain, sedangkan jika gagal, kata target akan diungkapkan di akhir permainan. Permainan ini mengadopsi logika dan format umpan balik dari WORDL3, namun menambah kompleksitas dengan lebih banyak kata untuk ditebak, sehingga memberikan pengalaman bermain yang lebih menantang dan strategis.

## 3. Struktur Data (ADT)

Berikut adalah ADT tambahan yang digunakan untuk mengatasi persoalan PURRMART.

STEI- ITB	IF2111-TB-03-01	Halaman 4 dari 45 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

### 3.1 ADT *array*

#### 3.1.1 Persoalan yang Diselesaikan

ADT array digunakan untuk mengelola data user yang ada di dalam sistem.

#### 3.1.2 Alasan Pemilihan

Array statik dipilih untuk menyimpan data pengguna karena diasumsikan jumlah pengguna (ukuran data) tidak akan berubah drastis selama runtime. Lalu operasi seperti menambah elemen, memvalidasi indeks, atau mencetak isi tabel lebih efisien bila diimplementasikan dengan menggunakan array statik.

#### 3.1.3 Implementasi

Implementasi array dalam program ini ada di dalam kode `array.c` pada direktori `src/ADT/array/array.c`

### 3.2 ADT *arraydin*

#### 3.2.1 Persoalan yang Diselesaikan

ADT arraydin digunakan untuk mengelola data barang.

#### 3.2.2 Alasan Pemilihan

Array dinamis dipilih sebagai struktur data karena memudahkan dalam pengelolaan memori secara otomatis dengan memperbesar atau memperkecil ukuran array sesuai kebutuhan sehingga lebih fleksibel tanpa harus menghapus data barang jika array sudah penuh.

#### 3.2.3 Implementasi

Implementasi arraydin terdapat pada kode `arraydin.c` pada direktori `src/ADT/arraydin.c`

### 3.3 ADT *boolean*

#### 3.3.1 Persoalan yang Diselesaikan

ADT boolean digunakan untuk menyelesaikan masalah terkait dengan penulisan dan pembacaan logika dalam kode. Dengan mendefinisikan tipe data boolean, serta nilai true dan false, program menjadi lebih mudah dibaca dan dipahami.

#### 3.3.2 Alasan Pemilihan

ADT boolean dipilih supaya kode menjadi lebih konsisten, mudah dibaca, dan mudah dipahami.

#### 3.3.3 Implementasi

Implementasi boolean terdapat pada kode `boolean.c` pada direktori `src/ADT/boolean.c`

### 3.4 ADT *mesinkarakter*

#### 3.4.1 Persoalan yang Diselesaikan

ADT Mesin Karakter dapat digunakan untuk membaca file konfigurasi, memproses input pengguna, dan menjadi dasar untuk implementasi modul lanjutan seperti mesin kata.

#### 3.4.2 Alasan Pemilihan

ADT Mesin Karakter dipilih karena efisien dalam membaca input satu per satu, memberi kontrol lebih pada pengolahan data teks. Selain itu, pemisahan pembacaan karakter dari bagian lain kode membuat program lebih mudah dibaca dan dikelola, serta memastikan alur pembacaan data yang konsisten.

#### 3.4.3 Implementasi

Implementasi mesinkarakter terdapat pada kode `mesinkarakter.c` pada direktori `src/ADT/mesinkarakter.c`

### 3.5 ADT *mesinkata*

#### 3.5.1 Persoalan yang Diselesaikan

ADT mesinkata digunakan untuk mempermudah proses membaca kata demi kata dari suatu file atau input, dalam hal ini dikembangkan sebagai pengganti dari `scanf`.

### 3.5.2 Alasan Pemilihan

ADT mesinkata dipilih untuk menerima input dan command dari user.

### 3.5.3 Implementasi

Implementasi mesinkata terdapat pada kode `mesinkata.c` pada direktori `src/ADT/mesinkata.c`

## 3.6 ADT *queue*

### 3.6.1 Persoalan yang Diselesaikan

ADT queue digunakan untuk menyelesaikan persoalan pengelolaan struktur data antrian (queue) untuk menyimpan dan memproses data barang yang masuk, keluar, dan dikelola secara FIFO (First In First Out).

### 3.6.2 Alasan Pemilihan

ADT queue berbasis circular buffer dipilih agar sesuai dengan kebutuhan pengelolaan data barang yang menggunakan aturan FIFO. Circular buffer dipilih untuk efisiensi memori dan kemudahan implementasi pada kasus dengan kapasitas tetap.

### 3.6.3 Implementasi

Implementasi queue terdapat pada kode `queue.c` pada direktori `src/ADT/queue.c`

## 3.7 ADT *misc*

### 3.7.1 Persoalan yang Diselesaikan

ADT Misc digunakan untuk mencari data, memeriksa input pengguna, dan mengubah tipe data. Selain itu, misc juga berisi fungsi-fungsi yang mendukung program utama.

### 3.7.2 Alasan Pemilihan

ADT Misc dipilih karena menyediakan fungsi-fungsi penting seperti pencarian elemen, validasi input, dan konversi data yang mempermudah pengelolaan aplikasi. Dengan memisahkan fungsi-fungsi pendukung ini, kode menjadi lebih terstruktur, mudah dibaca, dan lebih mudah dikelola.

### 3.7.3 Implementasi

Implementasi misc terdapat pada kode `misc.c` pada direktori `misc.c`

## 3.8 ADT *dplinkedlist*

### 3.8.1 Persoalan yang Diselesaikan

ADT *dplinkedlist* digunakan untuk menyelesaikan persoalan semua *wishlist*, *store list*, *start*, *load*, dan *save*.

### 3.8.2 Alasan Pemilihan

Pemilihan algoritma *doubly linked list* didasarkan pada fleksibilitasnya dalam manipulasi data dinamis, seperti penambahan dan penghapusan elemen di awal, akhir, atau tengah daftar dengan efisiensi tinggi. Struktur ini memungkinkan navigasi dua arah, memudahkan pencarian dan penyusunan elemen secara fleksibel tanpa memindahkan elemen lain, serta memanfaatkan memori secara dinamis.

### 3.8.3 Implementasi

Implementasi *dplinkedlist* terdapat pada kode `dplinkedlist.c` pada direktori `src/ADT/listlinier/dplinkedlist.c`

### 3.9 ADT *map*

#### 3.9.1 Persoalan yang Diselesaikan

ADT *map* digunakan untuk menyelesaikan persoalan semua *cart*, *store list*, *start*, *load*, dan *save*.

#### 3.9.2 Alasan Pemilihan

Pemilihan algoritma *map* dalam implementasi ini bertujuan untuk menyimpan dan mengelola pasangan *key-value* secara efisien, khususnya untuk mencari, menambah, dan memodifikasi data berdasarkan *key*. Struktur ini sangat cocok untuk aplikasi yang membutuhkan pengelompokan data unik, seperti inventori barang dalam keranjang belanja, dimana setiap *key* merepresentasikan nama barang dan *value* adalah jumlahnya. Algoritma *map* meminimalkan kompleksitas operasi pencarian dan manipulasi data dibandingkan struktur data lainnya karena sifatnya yang memungkinkan pencarian langsung berdasarkan *key*.

#### 3.9.3 Implementasi

Implementasi *map* terdapat pada kode `map.c` pada direktori `src/ADT/setmap/map.c`

### 3.10 ADT *stack*

#### 3.10.1 Persoalan yang Diselesaikan

ADT *stack* digunakan untuk menyelesaikan persoalan *cart pay*, *history*, *store list*, *start*, *load*, dan *save*.

#### 3.10.2 Alasan Pemilihan

Pemilihan algoritma *stack* dalam implementasi ini didasarkan pada kebutuhan pengelolaan data secara LIFO (Last In First Out), yang sangat cocok untuk skenario seperti *history*. Struktur *stack* memungkinkan operasi penambahan (*push*) dan penghapusan (*pop*) elemen dilakukan dengan efisien hanya pada satu ujung, yaitu *top*. Hal ini menjamin bahwa elemen terakhir yang ditambahkan akan menjadi elemen pertama yang dihapus, sesuai dengan kebutuhan kasus penggunaan seperti navigasi kembali atau manajemen riwayat perubahan. Dengan sifatnya yang sederhana dan efektif, *stack* dapat memastikan penggunaan memori yang terkontrol sesuai kapasitas maksimal (*MaxElStack*).

#### 3.10.3 Implementasi

Implementasi *stack* terdapat pada kode `stack.c` pada direktori `src/ADT/stack/stack.c`

## 4. Program Utama

Program utama terletak pada file `main.c` yang dapat dijalankan dengan menggunakan MakeFile 'mingw32-make run'. Dalam program `main.c` dimulai dengan pendefinisian header yaitu `main.h`. Dalam header `main.h` terdapat 28 header, yaitu `<stdlib.h>`, `misc.h` yang berisi fungsi-fungsi penting seperti pencarian elemen, validasi input, dan konversi data yang mempermudah pengelolaan aplikasi, `start.h`, `load.h`, `login.h`, `logout.h`, `register.h`, `work.h`, `work_challenge.h`, `store_list.h`, `store_req.h`, `store_sup.h`, `store_rmv.h`, `help.h`, `save.h`, `quit.h`, `profile.h`, `cart_add.h`, `cart_remove.h`, `cart_show.h`, `cart_pay.h`, `history.h`, `wishlist_add.h`, `wishlist_swap.h`, `wishlist_remove_ver1.h`, `wishlist_remove_ver2.h`, `wishlist_clear`, dan `wishlist_show.h`

Program dimulai dengan inisialisasi struktur data, dan setelah MakeFile dijalankan, pengguna akan melihat tampilan PURRMART dengan menu utama yang terdiri dari beberapa perintah, yaitu START, LOAD, HELP, dan QUIT. Perintah START akan menampilkan pesan "STARTING PROGRAM..." yang menyajikan informasi jumlah barang di toko, jumlah pengguna dalam program, data pengguna, riwayat pembelian, serta wishlist pengguna. Jika memilih LOAD, pengguna akan diminta memasukkan nama file yang ingin digunakan. Jika file valid, program akan menampilkan pesan "LOAD

PROGRAM..." dan "Save file berhasil dibaca. PURRMART berhasil dijalankan," sebagai tanda bahwa program berhasil dimuat. Perintah HELP menyediakan panduan penggunaan program, mencakup penjelasan mengenai Welcome Menu, Login Menu, dan Menu Help. Jika pengguna ingin keluar dari program, mereka dapat mengetik perintah QUIT dan memilih "Y" untuk keluar atau "N" untuk tetap berada di dalam program.

Program ini dilengkapi dengan command LOGIN dan REGISTER yang memungkinkan pengguna memulai sesi dengan akun yang sudah ada atau membuat akun baru. Setelah login, pengguna dapat menggunakan command LOGOUT untuk keluar dari sesi. Pengguna juga dapat memperoleh uang tambahan dengan command WORK atau menyelesaikan tantangan WORK CHALLENGE, seperti Tebak Angka, WORLD3, dan Quantum WORDL3. Selain itu, terdapat fitur manajemen barang, termasuk STORE REQUEST untuk mengajukan penambahan barang baru ke toko, STORE SUPPLY untuk menambahkan barang berdasarkan antrian, STORE LIST untuk melihat daftar barang yang tersedia, dan STORE REMOVE untuk menghapus barang dari toko.

Program ini juga menyediakan command tambahan lainnya, seperti PROFILE untuk melihat informasi pengguna setelah login. Terdapat fitur keranjang (*cart*) yang mencakup CART ADD dan CART REMOVE untuk menambah atau mengurangi barang di keranjang dengan jumlah tertentu, CART SHOW untuk melihat barang yang ada di keranjang, dan CART PAY untuk membeli barang yang sudah dimasukkan ke dalam keranjang. Fitur HISTORY memungkinkan pengguna melihat riwayat pembelian mereka.

Selain itu, program ini juga menyediakan fitur wishlist yang mencakup berbagai command, seperti WISHLIST ADD untuk menambahkan barang ke dalam wishlist, WISHLIST SWAP untuk menukar barang di wishlist, WISHLIST REMOVE untuk menghapus barang berdasarkan posisi atau nama barang, WISHLIST CLEAR untuk menghapus semua barang di wishlist, dan WISHLIST SHOW untuk menampilkan barang-barang yang ada di dalam wishlist. Semua fitur ini dirancang untuk memberikan pengalaman pengguna yang terstruktur dan mudah digunakan

## 5. Algoritma-Algoritma Menarik

### 5.1 Double Linked List

Penggunaan algoritma *double linked list* sangatlah menarik karena dibutuhkannya `<stdlib>` untuk mengatasi alokasi memori. Yang membuat algoritma ini special adalah penanganan traversal kedepan dan kebelakangnya. Dalam dunia nyata pengaplikasian DLL dapat digunakan pada sebuah aplikasi yang mempunyai fitur *back* dan *forward* seperti browser anda atau mungkin *spotify* yang tiap lagu nya pasti akan tertaut sehingga sangat mudah bagi pengguna untuk maju dan mundur ke lagu yang diinginkan. Selain itu, DLL juga mempunyai keunggulan dalam kemudahan pengatasan memori karena handlingnya yang menggunakan malloc sehingga *insert* dan *delete* sangat fleksibel. Berbeda dengan *static array* yang membutuhkan penanganan secara eksplisit.

### 5.2 Stack

Algoritma *stack* merupakan salah satu algoritma struktur data yang mudah dipahami sehingga penggunaanya sangat banyak dipakai untuk berbagai aplikasi. Algoritma *Stack* hanya memerlukan 2 operasi utama yaitu *push* dan *pop*. Dengan 2 operasi ini kita sudah bisa melakukan penanganan data, dalam kasus aslinya seperti pembacaan sintaks bahasa program tertentu, alokasi memori dan lainnya.





## START

>> START

STARTING PROGRAM . . .

1. Ayam Goreng Crisbar
2. AK47
3. Meong
4. Ayam Geprek Haji Mansur

Loaded Users:

User 1: user1 (Money: 100)

User 2: user2 (Money: 25)

Loaded Items:

Item 1: AK47 (Price: 10)

Item 2: M14 (Price: 10)

Item 3: Lalabu (Price: 20)

Item 4: Ayam Goreng Crisbar (Price: 20)

Item 5: Meong (Price: 500)

Item 6: Ligmaintator (Price: 200)

## 6.2 LOAD

Data yang di test : load.c

```
LOAD
>> LOAD
LOAD PROGRAM
Masukkan nama file yang akan di load :
struk.txt
Tidak ada file dengan nama tersebut!
Masukkan nama file yang akan di load :
default.txt
1. Ayam Goreng Crisbar
2. AK47
3. Meong
4. Ayam Geprek Haji Mansur
Save file berhasil dibaca. PURRMART berhasil dijalankan.
LOADING PROGRAM . . .
Loaded Users:
User 1: user1 (Money: 100)
User 2: user2 (Money: 25)

Loaded Items:
Item 1: AK47 (Price: 10)
Item 2: M14 (Price: 10)
Item 3: Lalabu (Price: 20)
Item 4: Ayam Goreng Crisbar (Price: 20)
Item 5: Meong (Price: 500)
Item 6: Ligmaintator (Price: 200)
```

Hasil yang seharusnya:

## LOAD

>> LOAD

LOAD PROGRAM

Masukkan nama file yang akan di load :

struk.txt

Tidak ada file dengan nama tersebut!

Masukkan nama file yang akan di load :

default.txt

1. Ayam Goreng Crisbar

2. AK47

3. Meong

4. Ayam Geprek Haji Mansur

Save file berhasil dibaca. PURRMART berhasil dijalankan.

LOADING PROGRAM . . .

Loaded Users:

User 1: user1 (Money: 100)

User 2: user2 (Money: 25)

Loaded Items:

Item 1: AK47 (Price: 10)

Item 2: M14 (Price: 10)

Item 3: Lalabu (Price: 20)

Item 4: Ayam Goreng Crisbar (Price: 20)

Item 5: Meong (Price: 500)

Item 6: Ligmaintator (Price: 200)

## 6.3 LOGIN

Data yang di test : login.c

>>LOGIN

Masukkan username: ligma

Masukkan password: alstrukdatkeren

Username atau password salah. Coba lagi.

Masukkan password: alstrukdatkeren

Anda telah login ke PURRMART sebagai user1.

LOGIN

Anda masih berada di sesi user1, silahkan logout terlebih dahulu!

>>LOGIN

Masukkan username: Aulia

Masukkan password: 12345

Username atau password salah. Coba lagi.

Masukkan username: Aulia

Masukkan password: 123456

Username atau password salah. Coba lagi.

Masukkan username: Aulia

Masukkan password: 123457

Username atau password salah. Coba lagi.

Sudah terlalu banyak percobaan, anda akan dikembalikan ke laman utama!

Hasil yang seharusnya:

```
// Contoh login yang gagal karena password salah
```

```
>>LOGIN
```

```
Masukkan username: ligma
```

```
Masukkan password: alstrukdatkeren
```

```
Username atau password salah. Coba lagi.
```

```
// Contoh login yang berhasil untuk username user1 dan password alstrukdatkeren
```

```
>>LOGIN
```

```
Masukkan username: user1
```

```
Masukkan password: alstrukdatkeren
```

```
Anda telah login ke PURRMART sebagai user1.
```

```
// Contoh login yang gagal karena terlalu banyak percobaan
```

```
>>LOGIN
```

```
Masukkan username: Aulia
```

```
Masukkan password: 12345
```

```
Username atau password salah. Coba lagi.
```

```
Masukkan username: Aulia
```

```
Masukkan password: 123456
```

```
Username atau password salah. Coba lagi.
```

```
Masukkan username: Aulia
```

```
Masukkan password: 1234567
```

```
Username atau password salah. Coba lagi.
```

```
Sudah terlalu banyak percobaan, anda akan dikembalikan ke laman utama!
```

```
// Contoh login yang gagal karena pengguna belum LOGOUT
```

```
>>LOGIN
```

```
Anda masih berada di sesi user1, silahkan logout terlebih dahulu!
```

## 6.4 LOGOUT

Data yang di test : logout.c

```
>> LOGOUT
```

```
user1 telah logout dari sistem PURRMART. Silakan REGISTER/LOGIN kembali untuk melanjutkan.
```

```
LOGOUT
```

```
Anda tidak berada di suatu sesi!
```

Hasil yang seharusnya:

```
// Contoh logout yang berhasil
```

```
>>LOGOUT
```

```
user1 telah logout dari sistem PURRMART. Silakan REGISTER/LOGIN kembali untuk melanjutkan.
```

```
// Contoh logout yang gagal karena pengguna belum login
>>LOGOUT
Anda tidak berada di suatu sesi!
```

## 6.5 REGISTER

Data yang di test : register.c

```
>>REGISTER
Masukkan username (maksimum 49 karakter, tanpa spasi): ivanaazizah
Masukkan password (maksimum 20 karakter): hahaha
Masukkan jumlah uang awal: 100
Akun dengan username ivanaazizah telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.
```

```
>>REGISTER
Masukkan username (maksimum 49 karakter, tanpa spasi): bagas
Akun dengan username bagas gagal dibuat karena sudah ada. Silakan lakukan REGISTER ulang.
```

```
>>REGISTER
Masukkan username (maksimum 49 karakter, tanpa spasi): ivana azizah
Username tidak boleh mengandung spasi.
```

```
Masukkan username (maksimum 49 karakter, tanpa spasi): maria
Masukkan password (maksimum 20 karakter): uhuy
Password tidak valid! Panjang password harus 6-20 karakter.
```

```
>>REGISTER
Masukkan username (maksimum 49 karakter, tanpa spasi): maria
Masukkan password (maksimum 20 karakter): uhuybanget
Masukkan jumlah uang awal: aaa
Input jumlah uang tidak valid. Masukkan angka positif.
Masukkan jumlah uang awal: -50
Input jumlah uang tidak valid. Masukkan angka positif.
```

Hasil yang seharusnya:

```
// Contoh register yang berhasil
>>REGISTER
Masukkan username (maksimum 49 karakter, tanpa spasi): ivanaazizah
Masukkan password (maksimum 20 karakter): hahaha
Masukkan jumlah uang awal: 100
Akun dengan username ivanaazizah telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.
```

```
// Contoh register yang gagal karena username sudah ada
>>REGISTER
Masukkan username (maksimum 49 karakter, tanpa spasi): bagas
Akun dengan username bagas gagal dibuat karena sudah ada. Silakan lakukan REGISTER ulang.
```

```
// Contoh register yang gagal karena username mengandung spasi
>>REGISTER
Masukkan username (maksimum 49 karakter, tanpa spasi): ivana azizah
Username tidak boleh mengandung spasi.
```

//Contoh register yang gagal karena password tidak sesuai

>>REGISTER

Masukkan username (maksimum 49 karakter, tanpa spasi): maria

Masukkan password (maksimum 20 karakter): uhuy

Password tidak valid! Panjang password harus 6-20 karakter.

//Contoh register yang gagal karena uang bukan interger dan bernilai negatif

>>REGISTER

Masukkan username (maksimum 49 karakter, tanpa spasi): maria

Masukkan password (maksimum 20 karakter): uhuybanget

Masukkan jumlah uang awal: aaa

Input jumlah uang tidak valid. Masukkan angka positif.

Masukkan jumlah uang awal: -50

Input jumlah uang tidak valid. Masukkan angka positif.

## 6.6 WORK

Data yang di test : work.c

```
>>WORK
Saldo sekarang : Rp 1000
Daftar pekerjaan:
1. Evil Lab Assistant (pendapatan=100, durasi=14s)
2. OWCA Hiring Manager (pendapatan=4200, durasi=21s)
3. Cikapundunginator Caretaker (pendapatan=7000, durasi=30s)
4. Mewing Specialist (pendapatan=10000, durasi=22s)
5. Inator Connoisseur (pendapatan=997, durasi=15s)
Masukkan nama pekerjaan yang dipilih: Evil Lab
Pekerjaan tidak ditemukan. Silakan coba lagi.
Masukkan nama pekerjaan yang dipilih: Evil Lab Assistant
Anda sedang bekerja sebagai Evil Lab Assistant... harap tunggu.
Pekerjaan selesai, +100 rupiah telah ditambahkan ke akun Anda.
Saldo sekarang setelah bekerja : Rp 1100
```

Hasil yang seharusnya:

>>WORK

Saldo sekarang : Rp 1000

Daftar pekerjaan:

1. Evil Lab Assistant (pendapatan=100, durasi=14s)
2. OWCA Hiring Manager (pendapatan=4200, durasi=21s)
3. Cikapundunginator Caretaker (pendapatan=7000, durasi=30s)
4. Mewing Specialist (pendapatan=10000, durasi=22s)
5. Inator Connoisseur (pendapatan=997, durasi=15s)

Masukkan nama pekerjaan yang dipilih: **Evil Lab**

Pekerjaan tidak ditemukan. Silakan coba lagi.

Masukkan nama pekerjaan yang dipilih: **Evil Lab Assistant**

Anda sedang bekerja sebagai Evil Lab Assistant... harap tunggu.

Pekerjaan selesai, +100 rupiah telah ditambahkan ke akun Anda.

Saldo sekarang setelah bekerja : Rp 1100

## 6.7 WORK CHALLENGE

Data yang di test : tebakangka.c

```
>> WORK CHALLENGE

Untuk keluar ketik (Purry)
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main=200)
2. W0RDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan (1/2): 1
Mulai challenge Tebak Angka!
30
Tebakanmu lebih besar!
40
Tebakanmu lebih besar!
23
Tebakanmu lebih besar!
7
Tebakanmu lebih kecil!
10
Tebakanmu lebih kecil!
19
Tebakanmu lebih besar!
15
Tebakanmu lebih besar!
14
Tebakanmu benar! +150 rupiah telah ditambahkan ke akun anda.
```

Hasil yang seharusnya:

```
>> WORK CHALLENGE

Untuk keluar ketik (Purry)
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main=200)
2. W0RDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan (1/2): 1
Mulai challenge Tebak Angka!
30
Tebakanmu lebih besar!
40
Tebakanmu lebih besar!
23
Tebakanmu lebih besar!
7
Tebakanmu lebih kecil!
10
Tebakanmu lebih kecil!
19
Tebakanmu lebih besar!
15
Tebakanmu lebih besar!
14
Tebakanmu benar! +150 rupiah telah ditambahkan ke akun anda.
```

Data yang di test : wordl399.c

```
>> WORK CHALLENGE
```

```
Untuk keluar ketik (Purry)
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main=200)
2. W0RDL3 (biaya main=500)
```

```
Masukan challenge yang hendak dimainkan (1/2): 2
```

```
WELCOME TO W0RDL3! YOU HAVE 5 CHANCES TO GUESS THE WORD!
```

```
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
```

```
Masukkan kata tebakan Anda (input dalam lowercase): aiueo
Hasil:
a*i%u%e%o%
```

```
Masukkan kata tebakan Anda (input dalam lowercase): galon
Hasil:
g%a1%o%n%
```

```
Masukkan kata tebakan Anda (input dalam lowercase): katak
Hasil:
katak
```

```
Selamat, Anda menang!
+1500 rupiah telah ditambahkan ke akun Anda.
Apakah kamu ingin bekerja kembali? (YA/Purry)
```

```
>> WORK CHALLENGE
```

```
Untuk keluar ketik (Purry)
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main=200)
2. W0RDL3 (biaya main=500)
```

```
Masukan challenge yang hendak dimainkan (1/2): 2
Saldo tidak mencukupi untuk memainkan challenge ini.
```

Hasil yang seharusnya:

```
//Contoh Work Challenge jika uang cukup
```

```
>> WORK CHALLENGE
```

```
Untuk keluar ketik (Purry)
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main=200)
2. W0RDL3 (biaya main=500)
```

```
Masukan challenge yang hendak dimainkan (1/2): 2
```

```
WELCOME TO W0RDL3! YOU HAVE 5 CHANCES TO GUESS THE WORD!
```

```
- - - - -
- - - - -
```



- - - - -  
- - - - -  
- - - - -

Masukkan kata tebakan Anda (input dalam lowercase): aiueo

Hasil:

a\*i%u%e%o%

Masukkan kata tebakan Anda (input dalam lowercase): galon

Hasil:

g%a%l%n%

Masukkan kata tebakan Anda (input dalam lowercase): katak

Hasil:

katak

Selamat, Anda menang!

+1500 rupiah telah ditambahkan ke akun Anda.

//Contoh Work Challenge jika uang tidak cukup

>> WORK CHALLENGE

Untuk keluar ketik (Purry)

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)

2. WORDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan (1/2): 2

Saldo tidak mencukupi untuk memainkan challenge ini.

Data yang di test : quantum\_wordl399.c

>> WORK CHALLENGE

Untuk keluar ketik (Purry)

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)

2. WORDL3 (biaya main=500)

3. Quntum WORDL3 (biaya main=10)

Masukan challenge yang hendak dimainkan (1/2/3): 3

WELCOME TO QUANTUM WØRDL3! YOU HAVE 9 CHANCES TO GUESS FOUR WORDS!

```
-----
-----
-----
-----
-----
-----
-----
-----
-----
```

Masukkan empat kata tebakan Anda (pisahkan dengan spasi): maria katak demon dokja

Hasil:

Tebakan 1: m%ar\*ia%

Tebakan 2: k\*a\*t%a%k%

Tebakan 3: d%e%a%o%n\*

Tebakan 4: d%o%k%j%a

Masukkan empat kata tebakan Anda (pisahkan dengan spasi): tapir kocak demon tapir

Hasil:

Tebakan 1: tapir

Tebakan 2: k\*o%a\*k%

Tebakan 3: d%e%a%o%n\*

Tebakan 4: t%ap%ir\*

Masukkan empat kata tebakan Anda (pisahkan dengan spasi): tapir kotak ivana maria

Hasil:

Tebakan 1: tapir

Tebakan 2: k\*o%a\*k%

Tebakan 3: ivana

Tebakan 4: maria

Masukkan empat kata tebakan Anda (pisahkan dengan spasi): tapir dokja ivana maria

Hasil:

Tebakan 1: tapir

Tebakan 2: dokja

Tebakan 3: ivana

Tebakan 4: maria

Selamat, Anda menang Quantum WØRDL3!

+10000 rupiah telah ditambahkan ke akun Anda 0-0000-0.

Apakah kamu ingin bekerja kembali? (YA/Purry)

Purry

Kamu sudah keluar dari WORK!

Hasil yang seharusnya:

>> WORK CHALLENGE

Untuk keluar ketik (Purry)

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. WØRDL3 (biaya main=500)

### 3. Quntum WORDL3 (biaya main=10)

Masukan challenge yang hendak dimainkan (1/2/3): 3

WELCOME TO QUANTUM WORDL3! YOU HAVE 9 CHANCES TO GUESS FOUR WORDS!

```
- - - - -      - - - - -      - - - - -      - - - - -  
- - - - -      - - - - -      - - - - -      - - - - -  
- - - - -      - - - - -      - - - - -      - - - - -  
- - - - -      - - - - -      - - - - -      - - - - -  
- - - - -      - - - - -      - - - - -      - - - - -  
- - - - -      - - - - -      - - - - -      - - - - -  
- - - - -      - - - - -      - - - - -      - - - - -  
- - - - -      - - - - -      - - - - -      - - - - -  
- - - - -      - - - - -      - - - - -      - - - - -
```

Masukkan empat kata tebakan Anda (pisahkan dengan spasi): maria katak demon dokja  
Hasil:

Tebakan 1: m%ar\*ia%

Tebakan 2: k\*a\*t%a%k%

Tebakan 3: d%e%m%o%n\*

Tebakan 4: d%o%k%j%a

Masukkan empat kata tebakan Anda (pisahkan dengan spasi): tapir kocak demon tapir  
Hasil:

Tebakan 1: tapir

Tebakan 2: k\*oc%a\*k%

Tebakan 3: d%e%m%o%n\*

Tebakan 4: t%ap%ir\*

Masukkan empat kata tebakan Anda (pisahkan dengan spasi): tapir kotak ivana maria  
Hasil:

Tebakan 1: tapir

Tebakan 2: k\*ot%a\*k%

Tebakan 3: ivana

Tebakan 4: maria

Masukkan empat kata tebakan Anda (pisahkan dengan spasi): tapir dokja ivana maria  
Hasil:

Tebakan 1: tapir

Tebakan 2: dokja

Tebakan 3: ivana

Tebakan 4: maria

Selamat, Anda menang Quantum WORDL3!

+10000 rupiah telah ditambahkan ke akun Anda 0~@0æà0~ğ.

Apakah kamu ingin bekerja kembali? (YA/Purry)

Purry

Kamu sudah keluar dari WORK!

## 6.8 STORE LIST

Data yang di test : store\_list.c

&gt;&gt; STORE LIST

List barang yang ada di toko:

- AK47 - Harga: 10
- M14 - Harga: 10
- Lalabu - Harga: 20
- Ayam Goreng Crisbar - Harga: 20
- Meong - Harga: 500
- Ligmaintator - Harga: 200

STORE LIST  
>>STORE LIST  
TOKO KOSONG

Hasil yang seharusnya:

>> STORE LIST

List barang yang ada di toko:

- AK47 - Harga: 10
- M14 - Harga: 10
- Lalabu - Harga: 20
- Ayam Goreng Crisbar - Harga: 20
- Meong - Harga: 500
- Ligmaintor - Harga: 200

```
>>STORE LIST
```

TOKO KOSONG

## 6.9 STORE REQUEST

Data yang di test : store req.c

STORE REQUEST

/ ) ( \_ ) ( \_ ) ( \_ \ ( \_ ) ( \_ \ ( \_ ) ( \_ ) ( ) ( ) ( \_ ) / ) ( \_ ) ( \_ )  
 \ \ ) ( ) ( ) ( \_ / ) \_ ) / ) \_ ) ( ) ( ) ( ) ( \_ ) \ \ ) ( \_ )  
 ( \_ / ( \_ ) ( \_ ) ( \_ ) \ ) ( \_ ) \ ) ( \_ ) ( \_ / \ ( \_ ) ( \_ ) ( \_ / ( \_ )

Write "Purrry" to exit

```
>>STORE REQUEST
===== [STORE REQUEST] =====
Nama barang yang diminta:
bom
Barang berhasil dimasukkan ke dalam antrian
Nama barang yang diminta:
bom
Barang sudah berada di dalam antrian!
Nama barang yang diminta:
a
Barang sudah ada di toko!
Nama barang yang diminta:
```

Hasil yang seharusnya:

Write "Purry" to exit

```
>>STORE REQUEST
===== [STORE REQUEST] =====
Nama barang yang diminta:
bom
Barang berhasil dimasukkan ke dalam antrian

Nama barang yang diminta:
bom
Barang sudah berada di dalam antrian!

Nama barang yang diminta:
a
Barang sudah ada di toko!
```

## 6.10 STORE SUPPLY

Data yang di test : store\_sup.c

```

  /  _  )  (  _  )  /  \  (  _  )  /  _  )  /  \  (  _  )  (  \  )
  \  _  \  )  (  o  )  /  )  )  \  _  \  \  (  )  \  )  \  (  \  )  /
  (  _  /  (  )  \  /  (  \  )  (  )  (  _  \  /  (  )  (  )  \  _  /  (  )

```

-----  
Write "Purry" to exit

```
>>STORE SUPPLY
===== [STORE SUPPLY] =====
Apakah kamu ingin menambahkan barang bom? (Terima/Tolak/Tunda)
Terima
Harga Barang :
150
```

```

bom dengan harga 150 telah ditambahkan ke toko.
Uangmu sekarang : Rp 850
Apakah kamu ingin menambahkan barang paku? (Terima/Tolak/Tunda)
Tunda
paku dikembalikan ke antrian.
Apakah kamu ingin menambahkan barang paku? (Terima/Tolak/Tunda)
Tolak
paku dibuang dari antrian.
Tidak ada barang di antrian

```

```

>>STORE SUPPLY
===== [STORE SUPPLY] =====
Apakah kamu ingin menambahkan barang bom? (Terima/Tolak/Tunda)
Terima
Harga Barang :
2000
Uang kamu tidak mencukupi!
Apakah kamu ingin menambahkan barang bom? (Terima/Tolak/Tunda)
Terima
Harga Barang :
aaaa
Masukkan sebuah integer yang lebih besar dari 0!

```

```

Apakah kamu ingin menambahkan barang paku? (Terima/Tolak/Tunda)
Purry
Kamu keluar dari Store Supply!

```

Hasil yang seharusnya:

Write "Purry" to exit

```

>>STORE SUPPLY
===== [STORE SUPPLY] =====
Apakah kamu ingin menambahkan barang bom? (Terima/Tolak/Tunda)
Terima
Harga Barang :
150

bom dengan harga 150 telah ditambahkan ke toko.
Uangmu sekarang : Rp 850

Apakah kamu ingin menambahkan barang paku? (Terima/Tolak/Tunda)
Tunda
paku dikembalikan ke antrian.

Apakah kamu ingin menambahkan barang paku? (Terima/Tolak/Tunda)
Tolak
paku dibuang dari antrian.
Tidak ada barang di antrian

```

```

>>STORE SUPPLY
===== [STORE SUPPLY] =====
Apakah kamu ingin menambahkan barang bom? (Terima/Tolak/Tunda)
Terima
Harga Barang :
2000
Uang kamu tidak mencukupi!

Apakah kamu ingin menambahkan barang bom? (Terima/Tolak/Tunda)

```

```
>>STORE SUPPLY
===== [STORE SUPPLY] =====
Apakah kamu ingin menambahkan barang paku? (Terima/ToLak/Tunda)
Purry
Kamu keluar dari Store Supply!
```

Data yang di test : store\_rmv.c

```
( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )  
 \ / ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )  
 ( / ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )
```

-----

```
>>STORE REMOVE  
===== [STORE REMOVE] =====  
Items in Store:  
Item 1: a (Price: 10)  
Item 2: b (Price: 20)  
Item 3: c d e (Price: 20)  
Item 4: Doofincshzmert (Price: 4100)  
Item 5: bom (Price: 77)  
Nama barang yang dihapus:  
bom  
Barang berhasil dihilangkan.  
Items in Store:  
Item 1: a (Price: 10)  
Item 2: b (Price: 20)  
Item 3: c d e (Price: 20)  
Item 4: Doofincshzmert (Price: 4100)
```

```
Nama barang yang dihapus:  
martabak  
Barang tidak ada di toko!  
Items in Store:  
Item 1: a (Price: 10)  
Item 2: b (Price: 20)  
Item 3: c d e (Price: 20)  
Item 4: Doofincshzmert (Price: 4100)
```

```
Write "Purrr" to exit
//Contoh Store Remove jika barang ada
>>STORE REMOVE
===== [STORE REMOVE] =====
Items in Store:
Item 1: a (Price: 10)
Item 2: b (Price: 20)
Item 3: c d e (Price: 20)
Item 4: Doofincshzmert (Price: 4100)
Item 5: bom (Price: 77)
```

```

Nama barang yang dihapus:
bom
Barang berhasil dihilangkan.
Items in Store:
Item 1: a (Price: 10)
Item 2: b (Price: 20)
Item 3: c d e (Price: 20)
Item 4: Doofincshzmert (Price: 4100)

```

```

//Contoh Store Remove jika barang tidak ada
>>STORE REMOVE
===== [STORE REMOVE] =====
Nama barang yang dihapus:
martabak
Barang tidak ada di toko!
Items in Store:
Item 1: a (Price: 10)
Item 2: b (Price: 20)
Item 3: c d e (Price: 20)
Item 4: Doofincshzmert (Price: 4100)

```

## 6.12 HELP

Data yang di test : help.c

```

HELP
====[ Welcome Menu Help PURRMART ]====
1. START -> Untuk masuk sesi baru
2. LOAD -> Untuk memulai sesi berdasarkan file konfigurasi
3. QUIT -> Untuk keluar dari program

```

```

HELP
====[ Login Menu Help PURRMART ]====
1. REGISTER -> Untuk melakukan pendaftaran akun baru
2. LOGIN -> Untuk masuk ke dalam akun dan memulai sesi
3. QUIT -> Untuk keluar dari program

```

```

====[ Menu Help PURRMART ]====
1. PROFILE -> Untuk melihat informasi akun
2. WORK -> Untuk bekerja
3. WORK CHALLENGE -> Untuk mengerjakan challenge
4. STORE LIST -> Untuk melihat barang-barang di toko
5. STORE REQUEST -> Untuk meminta penambahan barang
6. STORE SUPPLY -> Untuk menambahkan barang dari permintaan
7. STORE REMOVE -> Untuk menghapus barang
8. CART ADD <nama> <i>-> Untuk menambah barang ke keranjang
9. CART REMOVE <nama> <i>-> Untuk menghapus barang dari keranjang
10. CART SHOW -> Untuk melihat isi keranjang

```



11. CART PAY -> Untuk melakukan pembayaran
12. HISTORY -> Untuk melihat riwayat transaksi
13. WISHLIST ADD -> Untuk menambah barang ke wishlist
14. WISHLIST SWAP <i> <j>-> Untuk menukar barang di wishlist
15. WISHLIST REMOVE <i> -> Untuk menghapus barang dengan posisi ke-i dari wishlist
16. WISHLIST REMOVE -> Untuk menghapus barang berdasarkan nama barang dari wishlist
17. WISHLIST SHOW -> Untuk melihat wishlist
18. WISHLIST CLEAR -> Untuk membersihkan wishlist
19. LOGOUT -> Untuk keluar dari sesi
20. SAVE -> Untuk menyimpan state ke dalam file
21. QUIT -> Untuk keluar dari program

Hasil yang seharusnya:

// Ketika perintah dipanggil pada welcome menu

HELP

====[ Welcome Menu Help PURRMART ]====

1. START → Untuk masuk sesi baru
2. LOAD → Untuk memulai sesi berdasarkan file konfigurasi
3. QUIT → Untuk keluar dari program

// Ketika perintah dipanggil pada login menu

HELP

====[ Login Menu Help PURRMART ]====

1. REGISTER → Untuk melakukan pendaftaran akun baru
2. LOGIN → Untuk masuk ke dalam akun dan memulai sesi
3. QUIT → Untuk keluar dari program

// Ketika perintah dipanggil pada main menu

HELP

====[ Menu Help PURRMART ]====

1. PROFILE → Untuk melihat informasi akun
2. WORK → Untuk bekerja
3. WORK CHALLENGE → Untuk mengerjakan challenge
4. STORE LIST → Untuk melihat barang-barang di toko
5. STORE REQUEST → Untuk meminta penambahan barang
6. STORE SUPPLY → Untuk menambahkan barang dari permintaan
7. STORE REMOVE → Untuk menghapus barang
8. CART ADD <nama> <i> → Untuk menambah barang ke keranjang
9. CART REMOVE <nama> <i> → Untuk menghapus barang dari keranjang
10. CART SHOW → Untuk melihat isi keranjang
11. CART PAY → Untuk melakukan pembayaran
12. HISTORY → Untuk melihat riwayat transaksi
13. WISHLIST ADD → Untuk menambah barang ke wishlist
14. WISHLIST SWAP <i> <j> → Untuk menukar barang di wishlist
15. WISHLIST REMOVE <i> → Untuk menghapus barang dengan posisi ke-i dari wishlist
16. WISHLIST REMOVE → Untuk menghapus barang berdasarkan nama barang dari wishlist
17. WISHLIST SHOW → Untuk melihat wishlist
18. WISHLIST CLEAR → Untuk membersihkan wishlist
19. LOGOUT → Untuk keluar dari sesi
20. SAVE → Untuk menyimpan state ke dalam file

21. QUIT → Untuk keluar dari program

### 6.13 SAVE

Data yang di test : save.c

```
SAVE
Masukkan nama file untuk SAVE state program:
default.txt
save/default.txt
40 AK47
100 AK47
35 Lalabu
10 AK47
500 Meong
20 Ayam Goreng Crisbar
Save file berhasil disimpan
```

Hasil yang seharusnya:

```
SAVE
Masukkan nama file untuk SAVE state program:
default.txt
save/default.txt
Save file berhasil disimpan
```

### 6.14 QUIT

Data yang di test : quit.c

```
QUIT
Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? N

Kamu keluar dari PURRMART.
Dadah ^_^/
```

```
QUIT
Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? Y
Masukkan nama file untuk SAVE state program:
default.txt
save/default.txt
Save file berhasil disimpan

Kamu keluar dari PURRMART.
Dadah ^_^/
```

Hasil yang seharusnya:

>> QUIT

Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? N

Kamu keluar dari PURRMART.

Dadah ^\_^/

>> QUIT

Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? Y

Masukkan nama file untuk SAVE state program:

default.txt

save/default.txt

Save file berhasil disimpan

Kamu keluar dari PURRMART.

Dadah ^\_^/

## 6.15 PROFILE

Data yang di test : profile.c

```
PROFILE
>> PROFILE
Nama: user1
Saldo: 100
```

Hasil yang seharusnya:

```
>> PROFILE
Nama: user1
Saldo: 100
```

## 6.16 CART ADD <nama> <n>

Data yang di test : cart\_add.c

```
>> CART ADD AK47 10
AK47 sebanyak 10 telah ditambahkan ke keranjang.
AK47 : 10
```

```
>> CART ADD porsche 911
Barang tidak ada di toko!
AK47 : 10
```

Hasil yang seharusnya:

```
// Contoh Cart Add ketika barang ada di toko
>> CART ADD AK47 10
AK47 sebanyak 10 telah ditambahkan ke keranjang.
AK47 : 10
```

```
// Contoh Cart Add ketika barang tidak ada di toko
>> CART ADD porsche 911
Barang tidak ada di toko!
AK47 : 10
```

### 6.17 CART REMOVE <nama> <n>

Data yang di test : cart\_remove.c

```
>> CART REMOVE AK47 10
Berhasil mengurangi 10 AK47 dari keranjang belanja!
AK47 : 60
```

```
>> CART REMOVE porsche 911
Tidak ada barang dengan nama porsche dalam keranjang anda!
AK47 : 60
```

```
>> CART REMOVE AK47 80
Angka yang anda masukkan melebihi jumlah barang saat ini!
Pengurangan gagal dilakukan!
AK47 : 60
```

Hasil yang seharusnya:

```
// Contoh Cart Remove berhasil
>> CART REMOVE AK47 10
Berhasil mengurangi 10 AK47 dari keranjang belanja!
```

```
// Contoh Cart Remove ketika barang tidak ada di keranjang
>> CART REMOVE porsche 911
Tidak ada barang dengan nama porsche dalam keranjang anda!
```

```
// Contoh Cart Remove ketika barang melebihi jumlah barang saat ini
>> CART REMOVE AK47 80
Angka yang anda masukkan melebihi jumlah barang saat ini!
Pengurangan gagal dilakukan!
```

## 6.18 CART SHOW

Data yang di test : cart\_show.c

```
>> CART SHOW
Berikut adalah isi keranjangmu.
Kuantitas Nama                               Total
60          AK47                               600
Total biaya yang harus dikeluarkan adalah 600.
Ketik Purry untuk keluar
Masukkan nama barang untuk melihat detail atau ketik Purry untuk keluar:
```

```
CART SHOW
Keranjang anda kosong!
Perintah tidak dijalankan!
```

Hasil yang seharusnya:

```
// Contoh Cart Show ketika keranjang memiliki isi
>> CART SHOW
Berikut adalah isi keranjangmu.
Kuantitas Nama                               Total
60          AK47                               600
Total biaya yang harus dikeluarkan adalah 600.
Ketik Purry untuk keluar
Masukkan nama barang untuk melihat detail atau ketik Purry untuk keluar:
```

```
// Contoh Cart Show ketika keranjang kosong
>> CART SHOW
Keranjang anda kosong!
Perintah tidak dijalankan!
```

## 6.19 CART PAY

Data yang di test : cart\_pay.c

```
CART PAY
Keranjang anda kosong!
Perintah tidak dijalankan!
```

```
CART PAY
Berikut adalah isi keranjangmu.
Kuantitas Nama                               Total
2          AK47                               20
Total biaya yang harus dikeluarkan adalah 20, apakah jadi dibeli? (Ya/Tidak): Ya
Selamat kamu telah membeli barang-barang tersebut!
```

CART PAY

Berikut adalah isi keranjangmu.

Kuantitas	Nama	Total
68	AK47	680

Total biaya yang harus dikeluarkan adalah 680, apakah jadi dibeli? (Ya/Tidak): Ya  
Uang kamu hanya 80, tidak cukup untuk membeli keranjang!

CART PAY

Berikut adalah isi keranjangmu.

Kuantitas	Nama	Total
68	AK47	680

Total biaya yang harus dikeluarkan adalah 680, apakah jadi dibeli? (Ya/Tidak): Purry  
Anda telah keluar dari CART PAY.

CART PAY

Berikut adalah isi keranjangmu.

Kuantitas	Nama	Total
68	AK47	680

Total biaya yang harus dikeluarkan adalah 680, apakah jadi dibeli? (Ya/Tidak): Tidak  
Pembelian dibatalkan anda dikeluarkan dari CART PAY!

Hasil yang seharusnya:

// Contoh pembayaran berhasil

CART PAY

Berikut adalah isi keranjangmu.

Kuantitas	Nama	Total
2	AK47	20

Total biaya yang harus dikeluarkan adalah 20, apakah jadi dibeli? (Ya/Tidak): Ya  
Selamat kamu telah membeli barang-barang tersebut!

// Contoh pembayaran gagal (keranjang kosong)

CART PAY

Keranjang anda kosong!

Perintah tidak dijalankan!

// Contoh pembayaran gagal (pengguna tidak memiliki uang yang cukup)

CART PAY

Berikut adalah isi keranjangmu.

Kuantitas	Nama	Total
68	AK47	680

Total biaya yang harus dikeluarkan adalah 680, apakah jadi dibeli? (Ya/Tidak): Ya  
Uang kamu hanya 80, tidak cukup untuk membeli keranjang!

// Contoh pembayaran gagal (pengguna memasukan Tidak)

CART PAY

Berikut adalah isi keranjangmu.

Kuantitas	Nama	Total
68	AK47	680

Total biaya yang harus dikeluarkan adalah 680, apakah jadi dibeli? (Ya/Tidak): Tidak  
Pembelian dibatalkan anda dikeluarkan dari CART PAY!

```
// Contoh pembayaran gagal (pengguna memasukan masukan purry)
```

#### CART PAY

Berikut adalah isi keranjangmu.

Kuantitas Nama

Total

68 AK47

680

Total biaya yang harus dikeluarkan adalah 680, apakah jadi dibeli? (Ya/Tidak): Purry

Anda telah keluar dari CART PAY.

## 6.20 HISTORY <n>

Data yang di test : history.c

```
>> HISTORY 4
Riwayat pembelian barang:
1. AK47 40
2. AK47 100
3. Lalabu 35
4. AK47 10
Penampilan riwayat selesai!
```

```
HISTORY 10
>> HISTORY 10
Riwayat pembelian barang:
1. AK47 40
2. AK47 100
3. Lalabu 35
4. AK47 10
5. Meong 500
6. Ayam Goreng Crisbar 20
7. AK47 20
Penampilan riwayat selesai!
```

```
HISTORY 1
Tidak ada riwayat pembelian!
```

Hasil yang seharusnya:

```
// Contoh history menunjukan riwayat pembelian N < total riwayat
```

```
>> HISTORY 4
```

Riwayat pembelian barang:

1. AK47 40

2. AK47 100

3. Lalabu 35

4. AK47 10

Penampilan riwayat selesai!

```
// Contoh history menunjukan riwayat pembelian N ≥ total riwayat
```

```
>> HISTORY 10
```

```

 Riwayat pembelian barang:
 1. AK47 40
 2. AK47 100
 3. Lalabu 35
 4. AK47 10
 5. Meong 500
 6. Ayam Goreng Crisbar 20
 7. AK47 20
 Penampilan riwayat selesai!

```

```

// Contoh history menunjukan riwayat pembelian kosong
HISTORY 1
Tidak ada riwayat pembelian!

```

## 6.21 WISHLIST ADD

Data yang di test : wishlist\_add.c

```

Masukan nama barang wishlist:
Ayam Goreng Crisbar
Barang sudah ada di dalam wishlist!

```

```

Masukan nama barang wishlist:
dimsum boss
Tidak ada barang tersebut di dalam toko!

```

```

>>> WISHLIST ADD
Ketik Purry untuk keluar
Masukan nama barang wishlist:
AK47
debug alokasi AK47
Anda sudah keluar dari WISHLIST ADD!

```

Hasil yang seharusnya:

```

>> WISHLIST ADD
Masukan nama barang wishlist:
dimsum boss
Tidak ada barang tersebut di dalam toko!

```

```

>> WISHLIST ADD
Masukan nama barang wishlist:
Ayam Goreng Crisbar
Barang sudah ada di dalam wishlist!

```

```

>>> WISHLIST ADD
Ketik Purry untuk keluar
Masukan nama barang wishlist:
AK47
debug alokasi AK47

```



Anda sudah keluar dari WISHLIST ADD!

## 6.22 WISHLIST SWAP <i> <j>

Data yang di test : wishlist\_swap.c

```
>> WISHLIST SWAP 1 2
Berhasil menukar AK47 dengan Ayam Geprek Haji Mansur pada wishlist!
```

```
WISHLIST SWAP 1 7
Pastikan anda masukkan angka yang valid untuk melakukan swap!
e.g. WISHLIST SWAP <i> <j> dengan i dan j merupakan angka yang valid dalam wishlist anda! (integer positive dan tidak sama)
```

```
WISHLIST SWAP 1 2
Hanya terdapat satu element dalam wishlist anda!
```

Hasil yang seharusnya:

```
// Contoh Wishlist Swap berhasil menukar
>> WISHLIST SWAP 1 2
Berhasil menukar AK47 dengan Ayam Geprek Haji Mansur pada wishlist!
```

```
// Contoh Wishlist Swap gagal (wishlist tidak tersedia)
WISHLIST SWAP 1 7
Pastikan anda masukkan angka yang valid untuk melakukan swap!
e.g. WISHLIST SWAP <i> <j> dengan i dan j merupakan angka yang valid dalam wishlist anda!
(integer positive dan tidak sama)
```

```
// Contoh Wishlist Swap gagal (hanya terdapat satu barang)
WISHLIST SWAP 1 2
Hanya terdapat satu element dalam wishlist anda!
```

## 6.23 WISHLIST REMOVE <i>

Data yang di test : wishlist\_remove\_ver1.c

```
>> WISHLIST REMOVE 2
Berhasil menghapus barang posisi ke-2 dari wishlist!
```

```
WISHLIST REMOVE 10
Pastikan anda masukkan angka yang valid untuk melakukan remove!
e.g. WISHLIST REMOVE <i> dengan i merupakan angka yang valid dalam wishlist anda! (integer positive dan didalam indeks efektif)
```

```
WISHLIST REMOVE KDJ
Perintah tidak valid!
```

```
WISHLIST REMOVE 2
wishlist anda kosong!
```

Hasil yang seharusnya:

```
// Contoh Wishlist Remove berhasil menghapus barang ke-i dari WISHLIST
>> WISHLIST REMOVE 2
Berhasil menghapus barang posisi ke-2 dari wishlist!

// Contoh Wishlist Remove gagal (wishlist tidak tersedia)
>> WISHLIST REMOVE 10
Pastikan anda masukkan angka yang valid untuk melakukan remove!
e.g. WISHLIST REMOVE <i> dengan i merupakan angka yang valid dalam wishlist anda!
(integer positive dan didalam indeks efektif)

// Contoh Wishlist Remove gagal (pengguna memasukkan perintah yang tidak valid)
WISHLIST REMOVE KDJ
Perintah tidak valid!

// Contoh Wishlist Remove ketika WISHLIST kosong
WISHLIST REMOVE 2
wishlist anda kosong!
```

## 6.24 WISHLIST REMOVE

Data yang di test : wishlist\_remove\_ver2.c

```
WISHLIST REMOVE
Ketik Purry untuk keluar
Masukan nama barang yang akan dihapus:
Meong
1. Ayam Goreng Crisbar
2. Ayam Geprek Haji Mansur
Anda sudah keluar dari WISHLIST REMOVE!
```

```
WISHLIST REMOVE
Ketik Purry untuk keluar
Masukan nama barang yang akan dihapus:
porsche 911
Barang tidak ada di wishlist!
```

Hasil yang seharusnya:

```
// Contoh menghapus barang "Meong" dari WISHLIST
WISHLIST REMOVE
Ketik Purry untuk keluar
Masukan nama barang yang akan dihapus:
Meong
1. Ayam Goreng Crisbar
2. Ayam Geprek Haji Mansur
Anda sudah keluar dari WISHLIST REMOVE!
```

```
// Contoh menghapus barang gagal (barang tidak ada di WISHLIST)
WISHLIST REMOVE
Ketik Purry untuk keluar
```

Masukan nama barang yang akan dihapus:  
porsche 911  
Barang tidak ada di wishlist!

### 6.25 WISHLIST CLEAR

Data yang di test : wishlist\_clear.c

```
>> WISHLIST CLEAR  
wishlist berhasil dikosongkan!
```

Hasil yang seharusnya:

```
>> WISHLIST CLEAR  
Wishlist berhasil dikosongkan!
```

### 6.26 WISHLIST SHOW

Data yang di test : wishlist\_show.c

```
>> WISHLIST SHOW  
wishlist kamu kosong!
```

```
>> WISHLIST SHOW  
Berikut adalah isi wishlist-mu:  
1. Meong  
2. AK47
```

Hasil yang seharusnya:

```
>> WISHLIST SHOW  
Wishlist kamu kosong!
```

```
>> WISHLIST SHOW  
Berikut adalah isi wishlist-mu:  
1. Meong  
2. AK47
```

## 7. Test Script

No .	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	F00-Start	Memastikan bahwa program PURRMART dapat dijalankan	Memasukkan command START	Data Test 6.1	PURRMART berhasil dimulai dan akan menampilkan tampilan awal yang dapat dilihat pada data test xx	Sesuai yang diharapkan
2	F01-Load	Memastikan bahwa file konfigurasi dapat dibaca dan disimpan	Memasukkan command LOAD pada menu utama untuk file yang sudah di save	Data Test 6.2	Program berhasil membaca file yang sudah di <i>save</i> dan menampilkan "Save file berhasil dibaca. PURRMART berhasil dijalankan" dan akan memuat user dan item yang tersimpan	Sesuai yang diharapkan
3	F02-Login	Memastikan pengguna dapat masuk ke akun yang sudah didaftarkan	Memasukkan command <i>LOGIN</i> dilanjutkan dengan memasukkan <i>username</i> dan <i>password</i> pengguna yang sudah didaftarkan sebelumnya	Data Test 6.3	Program berhasil memasukkan pengguna ke akun yang diinput saat melakukan <i>login</i> , tidak berhasil jika <i>username</i> atau <i>password</i> salah, telah login atau belum melakukan <i>logout</i> , terlalu banyak melakukan percobaan	Sesuai yang diharapkan
4	F03-Logout	Memastikan pengguna dapat keluar dari akun	Memasukkan command <i>LOGOUT</i>	Data Test 6.4	Program berhasil mengeluarkan pengguna dari sistem PURRMART	Sesuai yang diharapkan
5	F04-Register	Memastikan pengguna dapat melakukan pendaftaran akun baru dengan input yang valid	Memasukkan command <i>REGISTER</i> dilanjutkan dengan memasukkan <i>username</i> , <i>password</i> dan uang sesuai dengan ketentuan	Data Test 6.5	Program berhasil mendaftarkan akun baru yang dimasukkan oleh pengguna	Sesuai yang diharapkan
6	F05-Work	Memastikan apakah command <i>WORK</i> dapat berjalan dengan baik	Memasukkan command <i>WORK</i> dilanjutkan dengan memasukkan nama pekerjaan yang dipilih	Data Test 6.6	Program berjalan dengan baik sehingga pengguna mendapatkan uang dari pekerjaan yang dipilih	Sesuai yang diharapkan
7	F06-Work Challenge	Memastikan command <i>WORK CHALLENGE</i> dapat menjalankan <i>challenge</i> tebak angka, <i>wordl3</i> , dan <i>quantum_wordl399</i>	Memasukkan command <i>WORK CHALLENGE</i> dan memilih salah satu <i>challenge</i> yang tersedia. Jika pengguna memilih opsi 1, program tebak angka dimulai	Data Test 6.7	Program berjalan dengan baik pada semua fitur di <i>WORK CHALLENGE</i> . Program tebak angka, <i>wordl399</i> , dan <i>quantum_wordl399</i> berjalan dengan baik sehingga ketika tebakan angka atau huruf	Sesuai yang diharapkan

			dan pengguna akan diminta untuk menginput angka hingga benar. sedangkan pengguna memilih opsi 2, program wordl3 dimulai dan pengguna diminta memasukkan 5 huruf hingga benar. Jika pengguna memilih opsi 3, program quantum_wordl399 dan pengguna diminta untuk menebak empat kata berbeda sekaligus dalam satu waktu dengan 9 kesempatan.		pengguna benar, uang akan bertambah secara otomatis ke akun pengguna	
8	F07-Store List	Memastikan apakah <i>command STORE LIST</i> dapat menampilkan nama barang yang dijual beserta harganya	Memasukkan <i>command STORE LIST</i>	Data Test 6.8	Program berhasil menampilkan barang - barang yang ada di toko beserta harganya	Sesuai yang diharapkan
9	F08-Store Request	Memastikan apakah <i>command STORE REQUEST</i> dapat melakukan permintaan penambahan barang baru ke dalam toko	Memasukkan <i>command STORE REQUEST</i> dilanjutkan dengan memasukkan nama barang yang diminta	Data Test 6.9	Program berhasil menyimpan barang yang diminta ke dalam antrian serta memastikan bahwa barang tersebut belum tersedia di toko maupun di antrian.	Sesuai yang diharapkan
10	F09-Store Supply	Memastikan apakah <i>command STORE SUPPLY</i> dapat menambahkan barang dari antrian ke dalam toko, serta memungkinkan untuk memasukkan kembali atau menghapus barang dari antrian	Memasukkan <i>command STORE SUPPLY</i> dilanjutkan dengan memilih untuk menerima, menunda atau menolak barang tersebut	Data Test 6.10	Program berhasil menambahkan barang baru ke dalam toko serta menentukan harga barang. Selain itu, program juga dapat menghapus barang yang telah ditambahkan dan memasukkannya kembali	Sesuai yang diharapkan
11	F10-Store Remove	Memastikan apakah <i>command STORE REMOVE</i> dapat menghapus barang pada toko	Memasukkan <i>command STORE REMOVE</i> dilanjutkan dengan memasukkan nama barang yang ingin dihapus	Data Test 6.11	Program berhasil menghapus barang dari toko dan menampilkan pesan bahwa barang tersebut telah berhasil dihapus	Sesuai yang diharapkan
12	F11- Help	Memastikan apakah <i>command HELP</i> dapat diakses dan	Memasukkan <i>command HELP</i> . Menu Help akan	Data Test 6.12	Program berhasil menampilkan Menu Help berdasarkan	Sesuai yang diharapkan

		menampilkan cara penggunaan program sesuai dengan welcome menu, login menu, dan main menu	keluar berdasarkan posisi pengguna, yaitu pada saat di welcome menu, login menu, dan main menu		welcome menu, login menu, dan main menu	
13	F12-Save	Memastikan command <i>SAVE</i> dapat menyimpan keadaan terbaru aplikasi ke dalam file tertentu	Memasukkan command <i>SAVE</i> dan menuliskan nama file yang akan disimpan	Data Test 6.13	Program berhasil menyimpan keadaan terbaru aplikasi ke dalam suatu file	Sesuai yang diharapkan
14	F13-Quit	Memastikan command <i>QUIT</i> dapat mengeluarkan pengguna dari program	Memasukkan command <i>QUIT</i>	Data Test 6.14	Program berhasil mengeluarkan pengguna dari program yang sedang berlangsung	Sesuai yang diharapkan
15	F14-Profile	Memastikan apakah command <i>PROFILE</i> dapat memperlihatkan data diri pengguna setelah melakukan login	Memasukkan command <i>PROFILE</i>	Data Test 6.15	Program berhasil menampilkan data diri pengguna berupa Nama dan Saldo	Sesuai yang diharapkan
16	F15-Cart Add	Memastikan apakah command <i>CART ADD</i> dapat menambahkan barang dengan kuantitas tertentu ke dalam keranjang belanja	Memasukkan command <i>CART ADD &lt;nama&gt; &lt;n&gt;</i>	Data Test 6.16	Program berhasil menambahkan barang dengan kuantitas tertentu ke dalam keranjang belanja	Sesuai yang diharapkan
17	F16-Cart Remove	Memastikan apakah command <i>CART REMOVE</i> dapat mengurangi barang sejumlah kuantitas tertentu dari keranjang belanja	Memasukkan command <i>CART REMOVE &lt;nama&gt; &lt;n&gt;</i>	Data Test 6.17	Program berhasil mengurangi barang sejumlah kuantitas tertentu dari keranjang belanja, serta melakukan validasi terhadap kuantitas yang diberikan, bila kuantitas pada keranjang belanja lebih sedikit dari N maka perintah akan gagal	Sesuai yang diharapkan
18	F17-Cart Show	Memastikan apakah command <i>CART SHOW</i> dapat menunjukkan barang-barang yang sudah dimasukkan ke dalam keranjang	Memasukkan command <i>CART SHOW</i>	Data Test 6.18	Program berhasil menunjukkan barang-barang yang sudah dimasukkan ke dalam keranjang dan menampilkan total biaya yang harus dikeluarkan	Sesuai yang diharapkan
19	F18-Cart Pay	Memastikan apakah command <i>CART PAY</i> dapat membeli barang-barang yang sudah dimasukan ke dalam keranjang	Memasukkan command <i>CART PAY</i>	Data Test 6.19	Program berhasil membeli barang-barang yang sudah dimasukan ke dalam keranjang dan mengurangi uang yang dimiliki pengguna, serta	Sesuai yang diharapkan

					menambahkan riwayat pembelian	
20	F19-History	Memastikan apakah <i>command</i> HISTORY dapat menunjukan riwayat pembelian seorang pengguna	Memasukkan <i>command</i> <i>HISTORY</i> <n>	Data Test 6.20	Program berhasil menunjukan riwayat pembelian seorang pengguna dengan menampilkan dari riwayat yang paling baru ke paling tua	Sesuai yang diharapkan
21	F20-Wishlist Add	Memastikan apakah <i>command</i> WISHLIST ADD dapat menambahkan suatu barang ke wishlist	Memasukkan <i>command</i> <i>WISHLIST ADD</i>	Data Test 6.21	Program berhasil menambahkan suatu barang ke wishlist	Sesuai yang diharapkan
22	F21-Wishlist Swap	Memastikan apakah <i>command</i> WISHLIST SWAP dapat menukar barang posisi ke-i dengan barang posisi ke-j pada wishlist	Memasukkan <i>command</i> <i>WISHLIST SWAP</i> <i> <j>	Data Test 6.22	Program berhasil menukar barang posisi ke-i dengan barang posisi ke-j pada wishlist	Sesuai yang diharapkan
23	F22-Wishlist Remove Ver1	Memastikan apakah <i>command</i> WISHLIST REMOVE dapat menghapus barang dengan posisi ke-i dari wishlist	Memasukkan <i>command</i> <i>WISHLIST REMOVE</i> <i>	Data Test 6.23	Program berhasil menghapus barang dengan posisi ke-i dari wishlist dan jika tidak ada wishlist ke-i maka penghapusan barang di wishlist gagal	Sesuai yang diharapkan
24	F23-Wishlist Remove Ver2	Memastikan apakah <i>command</i> WISHLIST REMOVE dapat menghapus barang dari wishlist berdasarkan nama barang yang dimasukkan pengguna	Memasukkan <i>command</i> <i>WISHLIST REMOVE</i>	Data Test 6.24	Program berhasil menghapus barang dari wishlist berdasarkan nama barang yang dimasukkan pengguna	Sesuai yang diharapkan
25	F24-Wishlist Clear	Memastikan apakah <i>command</i> WISHLIST CLEAR dapat menghapus semua barang yang terdapat di dalam wishlist	Memasukkan <i>command</i> <i>WISHLIST CLEAR</i>	Data Test 6.25	Program berhasil menghapus semua barang yang terdapat di dalam wishlist	Sesuai yang diharapkan
26	F25-Wishlist Show	Memastikan apakah <i>command</i> WISHLIST SHOW dapat menunjukan barang-barang yang sudah dimasukkan ke dalam wishlist	Memasukkan <i>command</i> <i>WISHLIST SHOW</i>	Data Test 6.26	Program berhasil menunjukan barang-barang yang sudah dimasukkan ke dalam wishlist	Sesuai yang diharapkan

## 8. Pembagian Kerja dalam Kelompok

Nama Lengkap - NIM	Deskripsi Tugas
Bagas Noor Fadhilah - 18223115	<ul style="list-style-type: none"><li>- Membuat code wishlist add, wishlist swap, wishlist remove, dan revisi start, load, dan user.</li><li>- Membuat laporan</li></ul>
Khairunnisa Azizah - 18223117	<ul style="list-style-type: none"><li>- Membuat code revisi store list, save, help dan driver stack, map, listlinier</li><li>- Membuat Laporan</li></ul>
Maria Vransiska P. C. T. D. P - 18223119	<ul style="list-style-type: none"><li>- Membuat code history, wishlist clear, dan wishlist show</li><li>- Membuat laporan</li></ul>
Theresia Ivana M. S - 18223126	<ul style="list-style-type: none"><li>- Membuat code cart remove, cart show, dan cart pay</li><li>- Membuat laporan</li></ul>
Aulia Azka Azzahra - 18223131	<ul style="list-style-type: none"><li>- Membuat code profile dan cart add</li><li>- Membuat laporan</li></ul>



## 9. Lampiran





### 9.1 Deskripsi Tugas Besar 2



Tugas besar ini bertujuan untuk membuat aplikasi simulasi berbasis CLI (command-line interface) menggunakan bahasa C dan berbagai ADT, yaitu ADT Kustom, ADT List, ADT Mesin Karakter, ADT Mesin Kata, dan ADT Queue. Selain itu, juga digunakan beberapa ADT tambahan seperti ADT Listlinier, ADT Map, dan ADT Stack, serta beberapa library seperti `stdio.h`, `stdlib.h`, `time.h`, dan `math.h`.

PURRMART adalah aplikasi yang mensimulasikan aktivitas beli barang pada *e-commerce* dengan fitur utama seperti menampilkan barang toko, meminta dan menyuplai barang baru, menyimpan dan membeli barang dalam keranjang, menampilkan barang yang sudah dibeli, membuat dan menghapus wishlist, serta bekerja untuk menghasilkan uang. Ketika program PURRMART pertama kali dijalankan, main menu akan ditampilkan dengan welcome menu dan beberapa perintah seperti START, LOAD, HELP, dan QUIT. Setelah itu, program akan memasuki login menu yang menyediakan command LOGIN, REGISTER, dan HELP. Program dapat menerima input sesuai perintah hingga perintah QUIT diberikan.

### 9.2 Notulen Rapat

Asistensi I

<b>Tanggal : 18 Desember 2024</b>	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"><li>- Semua penjelasan struktur data ADT (dari milestone 1) dimasukkan ke laporan milestone 2</li><li>- Bagian 2 (Penjelasan Tambahan Spesifikasi Tugas) khusus untuk bagian bonus</li><li>- Gambar SS bagian 5 (Data Test) dimasukkan ke lampiran untuk menghemat perhitungan halaman</li></ul>
<b>Tempat : Online (Google Meet)</b>  <b>Kehadiran Anggota Kelompok:</b>  1 18223115   2 18223117   3 18223119   4 18223126 	

<p>5 18223131</p> 	
	<p><b>Tanda Tangan Asisten:</b></p> 

### 9.3 Log Activity Anggota Kelompok

Hari, tanggal	Nama Lengkap - NIM	Kegiatan
Rabu, 4 Desember 2024	<p>Bagas Noor Fadhilah 18223115</p>	- Memasukkan ADT map, stack, dan <i>linked list</i> ke dalam repo github
Sabtu, 7 Desember 2024		<ul style="list-style-type: none"> <li>- Mengganti <i>linked list</i> dengan <i>double linked list</i></li> <li>- Revisi kode start untuk dapat menangani konfigurasi file yang baru</li> </ul>
Minggu, 8 Desember 2024		- Memperbaiki ADT stack dan DLL agar bisa menangani manipulasi string
Senin, 9 Desember 2024		- Memperbaiki ADT map untuk bisa menangani manipulasi string
Rabu, 11 Desember 2024		- Mengganti elemen maksimum setiap ADT untuk menghindari <i>overflow</i> pada sistem dan <i>refactor</i> untuk menghilangkan warning pada compiler
Jumat, 13 Desember 2024		- Memperbaiki start karena adanya corrupt pada uang user dan merevisi perintah load
Sabtu, 14 Desember 2024		<ul style="list-style-type: none"> <li>- Pembikinan perintah <i>wishlist_swap</i>, <i>wishlist_add</i>, <i>wishlist_remove</i> versi 1, dan <i>wishlist_remove</i> versi 2</li> <li>- Revisi register untuk menangani konfigurasi file revisi</li> </ul>

Minggu, 15 Desember 2024		- Debugging kode Aulia agar bisa menangani inline input
Senin, 16 Desember 2024		- Debugging kode Theresia agar bisa menangani inline input
Selasa, 17 Desember 2024		- Debugging kode Maria dan Theresia - Debugging ADT DLL - Pengisian dokumen bagian algoritma menarik
Rabu, 18 Desember 2024		- Debugging kode cart add, cart remove, save, dan cart_pay
Minggu, 22 Desember 2024		- Debugging kode quantum_wordl399
Rabu, 11 Desember 2024	Khairunnisa Azizah 18223117	- membuat driver test untuk ADT stack
Jumat, 13 Desember 2024		- membuat driver test untuk ADT map
Senin, 16 Desember 2024		- revisi store list dan help
Selasa, 17 Desember 2024		- revisi codesave
Rabu, 18 Desember 2024		- membuat driver test untuk ADT list linier
Kamis, 19 Desember 2024		- merevisi driver test karena ada error dan function baru
Jumat, 20 Desember 2024		- Membuat laporan bagian data test
Sabtu, 21 Desember 2024		- Membuat kode quantum_wordl399
Minggu, 22 Desember 2024		- Melanjutkan quantum_wordl399 dan debugging
Minggu, 15 Desember 2024	Maria Vransiska P. C. T. D. P 18223119	- Membuat kode History dan header history
Senin, 16 Desember 2024		- Membuat kode wishlist_clear dan header wishlist_clear - Membuat kode wishlist show dan header wishlist show

Selasa, 17 Desember 2024		<ul style="list-style-type: none"> <li>- Revisi kode wishlist show dan wishlist clear</li> <li>- Mengisi laporan bagian test script</li> </ul>
Rabu, 18 Desember 2024		<ul style="list-style-type: none"> <li>- Melanjutkan laporan bagian test script</li> </ul>
Jumat, 20 Desember 2024		<ul style="list-style-type: none"> <li>- Melanjutkan test script</li> </ul>
Jumat, 13 Desember 2024		<ul style="list-style-type: none"> <li>- Membuat kode cart_add</li> </ul>
Sabtu, 14 Desember 2024		<ul style="list-style-type: none"> <li>- Membuat kode cart_remove</li> </ul>
Minggu, 15 Desember 2024		<ul style="list-style-type: none"> <li>- Melanjutkan kode cart_remove dan membuat kode cart_show</li> </ul>
Senin, 16 Desember 2024		<ul style="list-style-type: none"> <li>- Melanjutkan kode cart_show dan membuat kode cart_pay</li> </ul>
Selasa, 17 Desember 2024	Theresia Ivana M. S 18223126	<ul style="list-style-type: none"> <li>- Melanjutkan kode cart_pay</li> <li>- Membuat laporan Struktur Data (ADT) bagian sketsa struktur data dan persoalan yang diselesaikan</li> </ul>
Rabu, 18 Desember 2024		<ul style="list-style-type: none"> <li>- Membuat notulen asistensi dan Test Script</li> </ul>
Kamis, 19 Desember 2024		<ul style="list-style-type: none"> <li>- Melanjutkan membuat Test Script</li> </ul>
Minggu, 22 Desember 2024		<ul style="list-style-type: none"> <li>- Menambahkan penjelasan mengenai quantum_word1399 di bagian 2 dan Test Script</li> <li>- Finalisasi laporan</li> </ul>
Sabtu, 14 Desember 2024		<ul style="list-style-type: none"> <li>- Membuat code profile dan cart add</li> </ul>
Minggu, 15 Desember 2024	Aulia Azka Azzahra 18223131	<ul style="list-style-type: none"> <li>- Menyusun laporan secara menyeluruh untuk mempermudah teman dalam proses menginputan data.</li> <li>- Membuat laporan bagian ringkasan</li> </ul>
Senin, 16 Desember 2024		<ul style="list-style-type: none"> <li>- Membuat laporan bagian ringkasan</li> <li>- Membuat laporan Struktur Data</li> </ul>

		(ADT) bagian sketsa struktur data dan persoalan yang diselesaikan - Membuat laporan bagian Program Utama - Generate bug daftar isi - Membuat laporan bagian Deskripsi Tugas Besar 2 - Membantu mengisi pembagian kerja dalam kelompok
Rabu, 18 Desember 2024		- Formating laporan - Menambahkan Struktur Data (ADT) - Membantu membuat Test Script
Kamis, 19 Desember 2024		- Melanjutkan membuat Test Script
Minggu, 22 Desember 2024		- Membuat data test quantum_word1399 dan beberapa perubahan - Finalisasi laporan