

# LAPORAN TUGAS BESAR

## IF2111 Algoritma dan Struktur Data STI

### PURRMART


Dipersiapkan oleh:

#### Kelompok 11

18223053 Daffa Athalla Rajasa  
18221026 Syasya Umaira  
18222041 Kezia Caren Cahyadi  
18223079 Atharizza Muhammad Athaya  
18223037 Michelle Hamdani  
18223013 Raka Adhitya Nugraha

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB-02-02		<jml hlm>
		Revisi	0	25/11/2024

# Daftar Isi

<b>1 Ringkasan</b>	<b>3</b>
2 Penjelasan Tambahan Spesifikasi Tugas	5
2.1 Quantum WORDL3	5
2.2 Bioweapon	5
3 Struktur Data (ADT)	5
3.1 Struktur Data Custom	6
3.2 Struktur Data Mesin Karakter	6
3.3 Struktur Data Mesin Kata	7
3.4 Struktur Data Queue	7
3.5 Struktur Data List	8
3.6 Struktur Data Listdin	9
4 Program Utama	9
5 Data Test	11
5.1 Data Test START	11
5.2 Data Test LOAD	11
5.3 Data Test LOGIN	12
5.4 Data Test LOGOUT	13
5.5 Data Test REGISTER	13
5.6 Data Test WORK	14
5.7 Data Test WORK CHALLENGE	15
5.7.1 Data Test Tebak Angka	15
5.7.2 Data Test WORDL3	16
5.7.3 Data Test Quantum WORDL3	18
5.8 Data Test STORE LIST	20
5.9 Data Test STORE REQUEST	20
5.10 Data Test STORE SUPPLY	22
5.11 Data Test STORE REMOVE	24
5.12 Data Test HELP	25
5.13 Data Test SAVE	25
5.14 Data Test QUIT	26
6 Test Script	26
7 Pembagian Kerja dalam Kelompok	32
<b>8 Lampiran</b>	<b>33</b>
8.1 Deskripsi Tugas Besar	33
8.2 Notulen Rapat	37
8.3 Log Activity Anggota Kelompok	38

# 1 Ringkasan

**PURRMART** adalah sebuah aplikasi berbasis command-line interface (CLI) yang mensimulasikan aktivitas proses jual-beli barang di situs *e-commerce*. Aplikasi ini dikembangkan menggunakan bahasa C dengan memanfaatkan berbagai struktur data untuk memenuhi kebutuhan pengguna. Aplikasi ini menggunakan library C seperti `stdio.h`, `stdlib.h`, `time.h`, dan `math.h`. **PURRMART** memiliki fitur utama seperti menampilkan barang yang ada pada toko, meminta dan menyuplai barang baru ke dalam toko, menyimpan dan membeli barang dalam keranjang, menampilkan barang yang sudah dibeli, membuat dan menghapus *wishlist*, serta untuk bekerja dan menghasilkan uang.

Pengguna dapat memulai aplikasi melalui beberapa opsi, yakni **START**, **LOAD**, atau **HELP**. Saat memilih **START**, aplikasi akan membaca file konfigurasi default yang berisi username, password, barang, dan harga barang yang tersedia. Kemudian, untuk opsi **LOAD**, pengguna mampu untuk memuat save file untuk melanjutkan sesi dari sebelumnya yang disimpan dalam file, sementara **HELP** memberikan panduan tentang command yang tersedia.

Selain fitur **START**, **LOAD** & **HELP**, aplikasi **PURRMART** juga menyediakan fitur pendaftaran akun baru. Pengguna dapat mendaftarkan akun baru dengan memasukkan atribut *username* dan *password*, dengan restriksi untuk *username* dan *password* harus terdiri hanya dari satu kata. Jika sudah memiliki akun, pengguna bisa untuk masuk ke akun menggunakan command **LOGIN**. Jika *username* belum didaftarkan atau *password* yang dimasukkan salah, maka pengguna tidak akan mendapatkan akses ke dalam aplikasi **PURRMART**.

Setelah melakukan registrasi dan *login*, pengguna bisa bekerja dan mendapatkan uang untuk berbelanja melalui *command* **WORK**. Dalam *command* **WORK** ini, ditawarkan beberapa pilihan pekerjaan dengan durasi dan upah yang bervariasi. Selama pengguna bekerja, sistem tidak bisa digunakan hingga pekerjaan selesai. Selain itu, pengguna juga dapat mendapatkan uang dengan melakukan challenge-challenge di OWCA, dengan menggunakan *command* **WORK CHALLENGE**.

Kemudian, untuk fungsi manajemen barang yang ada di toko, dapat dilakukan melalui beberapa *command*, diantaranya ada **STORE LIST** untuk menampilkan barang yang ada, **STORE REQUEST** untuk meminta barang baru, **STORE SUPPLY** untuk menambahkan barang baru berdasarkan daftar permintaan, dan **STORE REMOVE** untuk menghapus barang.

Saat pengguna ingin keluar dari akun mereka, pengguna dapat menggunakan *command* **LOGOUT** untuk keluar dari akun. Dalam menggunakan *command* **LOGOUT**, aplikasi **PURRMART** akan memastikan bahwa pengguna telah melakukan **LOGIN** sebelumnya sebelum memproses *command* ini. Lalu ada *command* **SAVE** yang memungkinkan pengguna menyimpan status aplikasi saat ini ke dalam suatu file dalam satu folder, sehingga bisa status aplikasi saat ditutup bisa diakses kembali di sesi berikutnya. Terakhir, ada *command* **QUIT** jika pengguna keluar dari aplikasi, baik dengan atau tanpa menyimpan status sesi terkini.

Laporan ini mencakup spesifikasi tugas besar Algoritma dan Struktur Data IF2111, struktur data apa saja yang digunakan, permasalahan yang ingin diselesaikan, serta alasan pemilihan dan implementasi struktur data yang digunakan. Laporan ini menjelaskan setiap *command*, dilengkapi dengan data uji dan skrip pengujian untuk setiap fitur dalam aplikasi PURRMART. Dengan berbagai *command* yang disediakan, aplikasi PURRMART memungkinkan pengguna melakukan manajemen barang, mengelola keranjang dan wishlist, serta memperoleh uang .

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Quantum WORDL3

Bonus Quantum WORDL3 adalah spesifikasi tambahan yang merupakan bagian dari *work challenge* bagi *user*. Fitur ini merupakan tingkatan lebih lanjut dari WORDL3 dengan pemain diminta untuk menebak empat kata sekaligus dalam satu waktu. Berbeda dengan WORDL3, *mini game* ini memperbolehkan user untuk menebak hingga 9 kali. Untuk mekanisme lainnya sama dengan spesifikasi wajib, WORDL3

### 2.2 Bioweapon

Dalam pengerjaan bonus Bioweapon, kelompok kami menggunakan ADT Mesin Karakter dan ADT Mesin Kata untuk pembacaan input, ADT Queue dan ADT List untuk mengecek keberadaan barang di *list\_barang* maupun dalam antrian yang ada. Untuk fungsi Bioweapon sendiri, pengguna diminta untuk menginput Nama Barang terlebih dahulu, dilanjut dengan Sekuens DNA yang nanti akan ditranslasi ke RNA, lalu kode rahasia yang jika terdapat di dalam Protein RNA, maka akan mengembalikan status berhasil ditambahkan ke dalam antrian pada STORE REQUEST. Terdapat beberapa fungsi baru yang digunakan di dalam fungsi Bioweapon ini, yaitu:

**Bioweapon():** Fungsi ini akan dipanggil jika user menginput STORE REQUEST BIOWEAPON pada terminal.

**DNAToRNA():** Fungsi ini ada di dalam fungsi **Bioweapon()**, tujuannya adalah untuk menerjemahkan DNA ke dalam bentuk RNA.

**IsCode():** Fungsi ini akan membandingkan antara protein dari RNA dengan kode rahasia yang diberikan oleh user.

**Decoding():** Fungsi ini ada di dalam fungsi **IsCode()**, tujuannya adalah mengubah bentuk RNA ke dalam protein dari masing-masing set tiga huruf.

**IsInString():** Fungsi ini juga ada di dalam fungsi **IsCode()**, tujuannya adalah mengecek kode rahasia merupakan substring dari protein. Jika pada offset 0 tidak ditemukan, maka akan dilanjutkan sampai dengan offset 2.

## 3 Struktur Data (ADT)

Dalam program ini, kelompok kami menggunakan beberapa ADT, diantaranya ADT Custom, ADT Mesin Karakter, ADT Mesin Kata, ADT Queue, ADT List, ADT Listdin.

### 3.1 Struktur Data Custom

Struktur data *Custom* terdiri dari dua tipe data utama, yakni *User* & *Barang*. *User* merupakan tipe data bentukan yang menyimpan informasi pengguna seperti nama yang terdiri dari maksimal 50 karakter, password yang terdiri dari maksimal 50 karakter, dan uang. Lalu, ada *barang* merupakan tipe data bentukan yang menyimpan informasi barang seperti nama yang terdiri dari maksimal 50 karakter dan harga. Primitif yang digunakan untuk program ini adalah:

**makeUser():** Fungsi ini digunakan untuk membuat objek *User* baru dengan nama, password, dan saldo uang 0.

**makeBarang():** Fungsi ini digunakan untuk membuat objek *Barang* baru dengan nama dan harga barang.

**IsMarkUser():** Fungsi ini digunakan untuk mengecek apakah pencarian sudah sampai ke akhir (*Mark User*).

**IsSameUser():** Fungsi ini digunakan untuk mengecek apakah user yang dibandingkan sama atau tidak.

**IsSameBarang():** Fungsi ini digunakan untuk mengecek apakah barang yang dibandingkan sama atau tidak.

**IsSameString():** Fungsi ini digunakan untuk mengecek apakah string yang dibandingkan sama atau tidak.

ADT *Custom* ini digunakan untuk mengelola data pengguna dan data barang dalam sistem PURRMART. Dengan menggunakan ADT ini, informasi pengguna dan barang dapat disimpan dan dikelola dengan cara yang terstruktur dan mudah diakses. Pemilihan ADT *Custom* dalam PURRMART didasarkan pada kebutuhan untuk menyimpan dan mengelola data pengguna dengan cara yang terstruktur. Diimplementasikan sebagai ADT *Custom* dengan nama file header “custom.h”.

### 3.2 Struktur Data Mesin Karakter

Pada struktur data ini, terdapat nilai *EOP* yang bertipe boolean untuk menyatakan apakah pita sudah mencapai akhir (*true*) atau belum (*false*), nilai *currentChar* yang bertipe character untuk menyimpan karakter yang sedang dibaca, dan *retval* nilai bertipe integer yang menunjukkan hasil operasi pembacaan dari fungsi *fscanf*, digunakan untuk mendeteksi akhir file. Primitif yang digunakan untuk program adalah sebagai berikut:

**START():** Fungsi yang digunakan untuk memulai membaca pita input dari standar input (*stdin*).

**ADV():** Fungsi untuk memajukan pita ke karakter berikutnya.

**GetCC():** Fungsi untuk mengambil karakter saat ini

**isEOP():** Fungsi untuk mengecek apakah pita telah mencapai akhir (*EOP*).

ADT *Mesin Karakter* digunakan untuk melakukan input string, parsing command program, dan membaca file konfigurasi ke dalam aplikasi PURRMART.

Mesin karakter dapat digunakan untuk melakukan pembacaan dan pemrosesan data teks secara sekuensial, baik dari input pengguna melalui *stdin* maupun dari file teks menggunakan *STARTFILE*. Dengan mendeteksi akhir pembacaan menggunakan *MARK* atau *EOF*, mesin

karakter memastikan data dibaca sepenuhnya untuk kebutuhan seperti parsing input, membaca konfigurasi, atau memuat data awal sistem.

Diimplementasikan sebagai ADT Mesin Karakter dengan nama file header “mesin\_karakter.h”.

### 3.3 Struktur Data Mesin Kata

Pada ADT Mesin kata, terdapat struktur data Word yang terdiri dari TabWord yang merupakan array of character, dan length yang bertipe integer untuk menghitung jumlah karakter dalam sebuah kata. Selain itu, ada juga boolean endWord untuk menandai pembacaan pita dan currentWord bertipe struktur data word yang merupakan pembacaan kata saat ini. Primitif yang digunakan adalah sebagai berikut:

**IgnoreBlanks():** Fungsi untuk mengabaikan satu atau lebih karakter kosong (BLANK) pada pita input.

**STARTWORD():** Fungsi untuk memulai pembacaan kata pada pita atau file input.

**ADVWORD():** Fungsi untuk melanjutkan pembacaan ke kata berikutnya pada pita.

**CopyWORD():** Fungsi untuk menyalin sebuah kata dari pita ke dalam currentWord.

**isEndWord():** Fungsi untuk memeriksa apakah pita telah mencapai akhir kata.

ADT Mesin Kata digunakan untuk mengelola input kata dan kalimat, serta memudahkan pemrosesan teks yang dimasukkan dari input pengguna maupun file yang dibaca. Alasan pemilihan ADT Mesin Kata dalam program PURRMART adalah adanya kebutuhan untuk pemrosesan kata dan kalimat yang ditandai dengan spasi atau karakter kosong yang sangat penting untuk aplikasi yang membutuhkan analisis teks. ADT ini dibuat untuk menggantikan fungsi input standar seperti scanf dengan mengolah masukan secara kata per kata. Implementasi Diimplementasikan sebagai ADT Mesin Kata dengan nama file header “mesin\_kata.h”.

### 3.4 Struktur Data Queue

Struktur data Queue yang diimplementasikan menggunakan representasi array dengan alokasi statik, dengan buffer yang menyimpan elemen-elemen queue. Queue ini menggunakan dua indeks utama, idxHead dan idxTail, yang berfungsi untuk menunjuk posisi elemen pertama dan terakhir dalam queue. Terdapat tipe data ElTypeQueue yang merupakan tipe data elemen dalam queue. Adapun tipe data yang disimpan dalam queue adalah tipe data bentukan *Word* yang diambil dari ADT Mesin Kata. Primitif yang digunakan untuk program ini adalah:

**CreateQueue():** Fungsi untuk membuat sebuah queue kosong.

**IsEmpty():** Fungsi untuk memeriksa apakah sebuah queue kosong atau tidak.

**IsFull():** Fungsi untuk memeriksa apakah queue sudah penuh.

**Length():** Fungsi untuk mendapatkan jumlah elemen yang ada di dalam queue.

**enqueue():** Fungsi untuk menambahkan elemen ke dalam queue dengan mekanisme FIFO (*first in first out*).

**dequeue():** Fungsi untuk mengeluarkan satu elemen dari sebuah queue menggunakan mekanisme FIFO (*first in first out*).

**displayQueue():** Fungsi untuk menampilkan isi suatu queue.

ADT ini digunakan untuk merepresentasikan antrian permintaan penambahan (*enqueue*) barang pada sistem. ADT Queue ini digunakan untuk mengelola antrian elemen bertipe data baranga dengan prinsip *First In First Out* (FIFO). Dengan menggunakan mekanisme penggunaan queue melingkar (direpresentasikan dengan modulo dalam fungsi **enqueue()**), elemen-elemen baru akan ditambahkan pada posisi terakhir yang tersedia, dan elemen pertama yang masuk akan dikeluarkan pertama kali. Hal ini sangat berguna untuk merepresentasikan urutan permintaan penambahan barang. Struktur data Queue ini diimplementasikan sebagai ADT Queue dengan nama file header “queue.h”.

### 3.5 Struktur Data List

Struktur data List adalah sebuah representasi array statis dengan kapasitas maksimal MaxEl (dalam implementasi program ini di set 100) dan elemen isi list bertipe ElType. Adapun tipe data yang disimpan dalam list adalah tipe data bentukan *User* yang diambil dari ADT Custom. Struktur ini digunakan untuk menyediakan berbagai fungsi yang dapat dimanfaatkan untuk memanipulasi daftar elemen pengguna secara terstruktur. Primitif yang digunakan untuk program ini adalah:

**MakeList():** Fungsi untuk membuat list kosong dengan kapasitas MaxEl.

**IsEmpty():** Fungsi untuk mengecek apakah list kosong.

**Get():** Fungsi untuk mendapatkan element list dengan indeks i.

**Set():** Fungsi untuk meng-*update* nilai pada elemen list dengan indeks i.

**Length():** Fungsi untuk menghitung jumlah elemen efektif dalam list (elemen selain Mark).

**FirstIdx():** Fungsi untuk mengembalikan indeks elemen pertama dalam list.

**LastIdx():** Fungsi untuk mengembalikan indeks terakhir dalam list.

**IsIdxValid():** Fungsi untuk mengecek apakah indeks tertentu valid dalam konteks ukuran list yang digunakan.

**IsIdxEff):** Fungsi untuk mengecek apakah indeks tertentu efektif dan terdefinisi dalam konteks list yang digunakan.

**Search():** Fungsi untuk mencari elemen tertentu dalam list.

**InsertFirst():** Fungsi untuk menyisipkan elemen baru di awal list.

**InsertAt():** Fungsi untuk menyisipkan elemen baru di indeks tertentu dalam list.

**InsertLast():** Fungsi untuk menyisipkan elemen baru di akhir list.

**DeleteFirst():** Fungsi untuk menghapus elemen pertama dari list.

**DeleteAt():** Fungsi untuk menghapus elemen di indeks tertentu dalam list.

**DeleteLast():** Fungsi untuk menghapus elemen terakhir dari list.

**Concat():** Fungsi untuk menggabungkan dua list menjadi satu list baru.

ADT list ini digunakan untuk menyelesaikan berbagai permasalahan terkait pengelolaan data linear berbasis array. ADT List yang memiliki struktur array ini mendukung akses langsung menuju elemen yang diinginkan dengan diidentifikasi melalui indeks, yakni daftar pengguna yang terdaftar di sistem PURRMART. Dukungan fungsi lengkap untuk manipulasi elemen

(penyisipan, penghapusan, pencarian) memudahkan pengguna dalam pengelolaan list. Implementasi dari ADT List adalah file header dengan nama “list.h”.

### 3.6 Struktur Data Listdin

Struktur data Listdin adalah sebuah representasi array dinamis dengan indeks bertipe `IdxType` dan elemen isi list bertipe `ElType`. Adapun tipe data yang digunakan sebagai `IdxType` adalah integer dan tipe data elemen yang disimpan dalam list adalah tipe data bentukan `Barang` yang diambil dari ADT `Custom`. Struktur ini digunakan untuk menyediakan berbagai fungsi yang dapat dimanfaatkan untuk memanipulasi daftar elemen barang secara dinamis. Primitif yang digunakan untuk program ini adalah:

**MakeListDin():** Fungsi untuk membuat list dinamis kosong.

**DeallocateListDin():** Fungsi untuk dealokasi list dinamis yang terdefinisi.

**IsEmptyListDin():** Fungsi untuk mengecek apakah list kosong.

**GetListDin():** Fungsi untuk mendapatkan element list dengan indeks *i*.

**GetCapacityListDin():** Fungsi untuk mendapatkan kapasitas yang tersedia.

**LengthListDin():** Fungsi untuk mendapatkan banyaknya elemen efektif array (elemen selain `Mark`).

**SearchListDin():** Fungsi untuk mencari elemen tertentu dalam list.

**InsertFirstListDin():** Fungsi untuk menyisipkan elemen baru di awal list.

**InsertAtListDin():** Fungsi untuk menyisipkan elemen baru di indeks tertentu dalam list.

**InsertLastListDin():** Fungsi untuk menyisipkan elemen baru di akhir list.

**DeleteFirstListDin():** Fungsi untuk menghapus elemen pertama dari list.

**DeleteAtListDin():** Fungsi untuk menghapus elemen di indeks tertentu dalam list.

**DeleteLastListDin():** Fungsi untuk menghapus elemen terakhir dari list.

**CopyListDin():** Fungsi untuk menyalin list dinamis.

**ReverseListDin():** Fungsi untuk membalikan (melakukan *reverse*) terhadap suatu list.

**PrintListDin():** Fungsi untuk menampilkan list ke layar.

**IsMemberListDin():** Fungsi untuk mengecek apakah suatu elemen merupakan bagian dari list yang digunakan.

ADT listdin ini digunakan untuk menyelesaikan berbagai permasalahan terkait pengelolaan data barang yang dinamis dengan berbasis array. ADT Listdin yang memiliki struktur array ini mendukung akses langsung menuju barang yang diinginkan dengan diidentifikasi melalui indeks, serta fleksibilitas untuk menyimpan daftar produk yang tersedia di PURRMART, termasuk harga dan stok secara dinamis. Dukungan fungsi lengkap untuk manipulasi elemen (penyisipan, penghapusan, pencarian) memudahkan pengguna dalam pengelolaan list. Implementasi Diimplementasikan sebagai ADT Listdin dengan nama file header “listdin.h”

## 4 Program Utama



Pada program sistem aplikasi berbasis CLI PURRMART ini, program dijalankan dengan memulai program utama 'main.c'. Program ini kemudian akan memerintahkan untuk input beberapa *command*, antara lain **START**, **LOAD**, dan **HELP** sebagai masukan pertama dari aplikasi PURRMART. *Command START* berfungsi untuk memulai aplikasi dengan file konfigurasi default yang sudah berisi daftar barang, harga, dan jumlah. *Command LOAD <filename>* digunakan untuk melakukan *save file* yang berisi daftar barang dan *user*. Yang terakhir, *command HELP* berfungsi untuk menampilkan daftar *command* yang dapat dieksekusi.

Program ini memiliki beberapa *command* utama setelah file konfigurasi telah berhasil dibaca, diantaranya:

1. **LOGIN**: Masuk ke akun di sistem dengan *username* dan *password* yang sudah terdaftar dalam sistem PURRMART sebelumnya.
2. **LOGOUT**: Keluar dari akun yang sedang digunakan saat sesi terakhir disimpan..
3. **REGISTER**: Mendaftarkan akun baru ke dalam sistem dengan *username* dan *password*.
4. **WORK**: Menampilkan daftar pekerjaan yang dapat dipilih lalu menambahkan *money* pada *user* setelah selesai melakukan pekerjaan.
5. **Tebak Angka**: Sebuah *challenge* yang meminta pemain menebak sebuah angka dalam 10 kesempatan, dan akan memberikan *feedback* apakah angka tebakannya lebih besar, kecil, atau sama dengan angka target. Semakin cepat pengguna dapat menebak angka target, maka uang yang diterima pun semakin banyak.
6. **WORDL3**: *Challenge* tebak kata dengan lima karakter. Program akan memberikan *feedback* sesuai dengan huruf - huruf yang dimasukkan oleh pengguna, apakah huruf tersebut ada dan berada di posisi yang benar, ada tapi berada di posisi yang salah, atau tidak ada sama sekali.
7. **STORE LIST**: Menampilkan semua barang yang terdapat di toko, dengan ketentuan bahwa tidak mungkin ada barang yang sama (setiap barang unik)
8. **STORE REQUEST**: Meminta penambahan barang baru ke toko, yang masa semua permintaan ini akan disimpan dalam sebuah antrian dan akan dimasukkan ke dalam toko menggunakan *command STORE SUPPLY*. Antrian penambahan barang ini akan selalu dikosongkan kembali setiap kali program dijalankan dari awal.
9. **STORE SUPPLY**: Menambahkan barang baru ke dalam toko berdasarkan antrian yang dibuat pada **STORE REQUEST**. Barang yang berada pada antrian paling depan akan dimasukkan ke toko, dan pengguna dapat menerima, menunda, atau menolak permintaan. Jika diterima maka program akan memasukkan barang tersebut ke toko dan meminta nilai harga, jika ditunda maka barang tersebut akan dimasukkan ke dalam antrian lagi, dan jika barang ditolak maka barang akan dihapus dari antrian
10. **STORE REMOVE**: Menghapus barang yang ada di toko. Program akan meminta nama barang yang akan dihapus dan memberi tahu apakah barang berhasil terhapus dari toko
11. **HELP**: Menampilkan daftar *command* yang mungkin dieksekusi dengan deskripsi pada setiap *command* yang ada dalam program ini.
12. **SAVE <filename>**: Menyimpan status aplikasi saat ini ke dalam file.
13. **QUIT**: Keluar dari aplikasi dengan opsi untuk menyimpan data sesi saat keluar.

Aplikasi PURRMART memungkinkan pengguna untuk melakukan pekerjaan yang menghasilkan uang, melakukan mini games seperti wordl3 dan tebak angka, aktivitas perbelanjaan serta meminta barang baru pada *store* yang tersedia. Sebelum mengakses semua fitur tersebut, pengguna dapat memilih *save file* yang ingin mereka gunakan, dan pengguna juga dapat mengakses PURRMART dengan *username* yang sudah ada atau membuat *username* baru.

## 5 Data Test

### 5.1 Data Test START

```
>> START

Save file berhasil dibaca. PURRMART berhasil dijalankan.
USER KOSONG!

ISI TOKO:
=====
| No | Nama Barang | Harga |
=====
| 1 | AK47 | 10 |
| 2 | Lalabu | 20 |
| 3 | Ayam Goreng Crisbar | 20 |
=====
```

Penjelasan: Tes ini dilakukan untuk memulai pertama kali dalam Toko PURRMART dalam file konfigurasi *default*.

### 5.2 Data Test LOAD

```
>> LOAD save1.txt

Save file berhasil dibaca. PURRMART berhasil dijalankan.
ISI USER:
=====
| Name | Password | Money |
=====
| suci | suci123 | 1000 |
| budi | abc132 | 400 |
=====

ISI TOKO:
=====
| No | Nama Barang | Harga |
=====
| 1 | Nasi Bakar | 30 |
| 2 | Bakso Pak Budi | 40 |
| 3 | Emping Bu Sucipto | 100 |
=====
```

```
>> LOAD asal.txt
```

```
Save file tidak ditemukan. PURRMART gagal dijalankan.  
USER KOSONG!
```

```
TOKO KOSONG!
```

Penjelasan: Tes ini dilakukan untuk memulai pertama kali dalam Toko PURRMART, tetapi dengan membuka file yang diharapkan.

### 5.3 *Data Test LOGIN*

```
>> LOGIN
```

```
Username: atha
```

```
Password: atha123
```

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

```
>> LOGIN
```

```
Username: suci
```

```
Password: suci123
```

```
Username atau password salah.
```

```
>> LOGIN

Username: atha
Password: atha1
Username atau password salah.
```

Penjelasan: Tes ini dilakukan untuk memulai sesi dengan *login*, yaitu masuk ke akun di sistem PURRMART yang sudah didaftarkan sebelumnya.

#### 5.4 Data Test LOGOUT

```
>> LOGOUT

Selamat Jalan atha!
>> █
```

Penjelasan: Tes ini dilakukan untuk mengeksekusi fitur *logout*, yaitu keluar dari akun yang digunakan untuk mengakses sistem PURRMART yang sedang digunakan.

#### 5.5 Data Test REGISTER

```
>> REGISTER

Username: atha
Password: atha123
Akun dengan username atha telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.
ISI USER:
=====
| Name                               | Password                               | Money   |
=====
| atha                               | atha123                               | 0       |
=====
```

Penjelasan: Tes ini dilakukan untuk memulai sesi dengan *register*, yaitu mendaftarkan akun baru ke dalam sistem PURRMART yang belum didaftarkan sebelumnya.

## 5.6 Data Test WORK

```
>> WORK
```

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 pekerjaan yang dipilih (1-5): 4

Anda sedang bekerja sebagai Mewing Specialist. Harap tunggu...

Pekerjaan selesai. +10000 rupiah telah ditambahkan ke akun Anda.

ISI USER:

Name	Password	Money
RAK	123	0
atha	atha123	10000

Penjelasan: Tes ini dilakukan untuk mendapatkan uang dengan *work*, yaitu ‘melakukan’ pekerjaan yang nantinya akan mendapatkan uang dan ditambahkan ke akun di sistem PURRMART yang sudah didaftarkan sebelumnya.

## 5.7 Data Test WORK CHALLENGE

### 5.7.1 Data Test Tebak Angka

>> WORK CHALLENGE

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. W0RDL399 (biaya main=500)
3. QUANTUM W0RDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan: 1

Hai! Aku telah memilih angka antara 1 sampai 100. Coba tebak ya!

Kamu memiliki 10 kesempatan untuk menebak.

Setiap kesalahan akan mengurangi kesempatan menebak dan skormu.

1x percobaan benar: 500 poin

2-3x percobaan benar: 300 poin

4-6x percobaan benar: 100 poin

7-10x percobaan benar: 50 poin

Masukkan tebakanmu: 50

Tebakanmu terlalu besar. Coba lagi!

Masukkan tebakanmu: 30

Tebakanmu terlalu kecil. Coba lagi!

Masukkan tebakanmu: 40

Tebakanmu terlalu kecil. Coba lagi!

Masukkan tebakanmu: 43

Tebakanmu terlalu besar. Coba lagi!

Masukkan tebakanmu: 42

Selamat! Tebakanmu benar! Angka yang aku pilih adalah 42.

Kamu berhasil menebak angka tersebut dalam 5 kali percobaan.

kamu mendapatkan 100 poin.

>> WORK CHALLENGE

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. W0RDL399 (biaya main=500)
3. QUANTUM W0RDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan: 1

Penjelasan: Tes ini dilakukan untuk melakukan sesi *work challenge*, yaitu melakukan mini game dengan menebak angka yang nantinya akan mendapatkan uang dengan restriksi sudah memiliki uang yang cukup untuk bermain mini game tersebut.

### 5.7.2 Data Test W0RDL3

```
>> WORK CHALLENGE
```

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. W0RDL399 (biaya main=500)
3. QUANTUM W0RDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan: 2

WELCOME TO W0RDL3, YOU HAVE 5 CHANCES TO ANSWER BEFORE YOU LOSE!

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

Masukan kata tebakan Anda: SUCIA

S% U\* C% I\* A%

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

Masukan kata tebakan Anda: AAAAA

S% U\* C% I\* A%

A% A% A% A% A%

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

Masukan kata tebakan Anda: ASAsa

S% U\* C% I\* A%

A% A% A% A% A%

A% S% A% s% a%

```
>> WORK CHALLENGE
```

```
Daftar challenge yang tersedia:
```

1. Tebak Angka (biaya main=200)
2. WØRDL399 (biaya main=500)
3. QUANTUM WØRDL3 (biaya main=500)

```
Masukan challenge yang hendak dimainkan: 2
```

```
Uang tidak cukup! Silakan bekerja dulu.
```

Penjelasan: Tes ini dilakukan untuk melakukan sesi *work challenge*, yaitu melakukan mini game dengan menebak satu kata yang nantinya akan mendapatkan uang jika bisa ditebak dengan benar. Program ini akan berjalan dengan restriksi pengguna sudah memiliki uang yang cukup untuk bermain mini game tersebut.



### 5.7.3 Data Test Quantum W0RDL3

>> WORK CHALLENGE

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. W0RDL399 (biaya main=500)
3. QUANTUM W0RDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan: 3

Selamat datang di Quantum W0RDL3!

Tugas Anda: Tebak 4 kata berbeda dalam 9 kesempatan.

Percobaan ke-1 dari 9

Masukkan 4 tebakan Anda (pisahkan dengan spasi):

RUMAH RUMAH KAMAR BULAN

Hasil percobaan:

Kata 1 salah:

Kata 2 salah:

Kata 3 salah:

Kata 4 salah:

Percobaan ke-2 dari 9

Masukkan 4 tebakan Anda (pisahkan dengan spasi):

RUMAH RUMAH KAMAR BULAN

Hasil percobaan:

Kata 1 salah:

Kata 2 salah:

Kata 3 salah:

Kata 4 salah:

```
Percobaan ke-9 dari 9
Masukkan 4 tebakan Anda (pisahkan dengan spasi):
RUMAH RUMAH KAMAR BULAN

Hasil percobaan:
Kata 1 salah:
Kata 2 salah:
Kata 3 salah:
Kata 4 salah:

Anda kehabisan kesempatan! Jawaban yang benar adalah:
Kata 1: ANGIN
Kata 2: PUNYA
Kata 3: PUNYA
Kata 4: RUMAH
```

## >> WORK CHALLENGE

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. W0RDL399 (biaya main=500)
3. QUANTUM W0RDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan: 3  
Uang tidak cukup! Silakan bekerja dulu.

Penjelasan: Tes ini dilakukan untuk melakukan sesi *work challenge*, yaitu melakukan mini game dengan menebak 4 kata yang nantinya akan mendapatkan uang jika bisa ditebak dengan benar. Program ini akan berjalan dengan restriksi pengguna sudah memiliki uang yang cukup untuk bermain mini game tersebut.

## 5.8 Data Test STORE LIST

```
>> STORE LIST

ISI TOKO:

=====
| No | Nama Barang | Harga |
=====
| 1  | Nasi Bakar | 30    |
| 2  | Bakso Pak Budi | 40    |
| 3  | Emping Bu Sucipto | 100   |
=====
```

```
>> STORE LIST

TOKO KOSONG!
```

Penjelasan: Tes ini dilakukan untuk melihat barang-barang apa saja yang ada dalam toko.

## 5.9 Data Test STORE REQUEST

```
>> STORE REQUEST

Nama barang yang diminta: UAG
ISI ANTRIAN:

=====
| No | Content |
=====
| 1  | UAG     |
=====
```

```
>> STORE REQUEST
```

```
Nama barang yang diminta: UAG
```

```
Barang dengan nama yang sama sudah ada di antrian!
```

```
ISI ANTRIAN:
```

```
=====
| No  | Content                                     |
=====
| 1   | UAG                                         |
=====
```

```
>> STORE REQUEST
```

```
Nama barang yang diminta: AK47
```

```
Barang dengan nama yang sama sudah ada di toko!
```

```
ANTRIAN KOSONG!
```

Penjelasan: Tes ini dilakukan untuk meminta penambahan barang baru ke dalam toko. Barang-barang yang diminta akan disimpan dalam antrian.

### 5.10 Data Test STORE SUPPLY

>> STORE SUPPLY

Apakah kamu ingin menambahkan barang UAG: Terima

Harga barang: 10

UAG, 3"UAG" dengan harga 10 telah ditambahkan ke toko.

ISI TOKO:

No	Nama Barang	Harga
1	AK47	10
2	Lalabu	20
3	Ayam Goreng Crisbar	20
4	UAG	10

>> STORE SUPPLY

Apakah kamu ingin menambahkan barang MK-14: Tunda

"MK-14" dikembalikan ke antrian.

ISI TOKO:

No	Nama Barang	Harga
1	AK47	10
2	Lalabu	20
3	Ayam Goreng Crisbar	20
4	UAG	10

```
>> STORE SUPPLY

Apakah kamu ingin menambahkan barang MK-14: Tolak
"MK-14" dihapuskan dari antrian.
ISI TOKO:
=====
| No | Nama Barang | Harga |
=====
| 1 | AK47 | 10 |
| 2 | Lalabu | 20 |
| 3 | Ayam Goreng Crisbar | 20 |
| 4 | UAG | 10 |
=====
```

```
>> STORE SUPPLY

Apakah kamu ingin menambahkan barang M4: Purry
ISI TOKO:
=====
| No | Nama Barang | Harga |
=====
| 1 | AK47 | 10 |
| 2 | Lalabu | 20 |
| 3 | Ayam Goreng Crisbar | 20 |
| 4 | UAG | 10 |
=====
```

Penjelasan: Tes ini dilakukan untuk menambahkan barang baru ke dalam toko berdasarkan antrian permintaan. Barang yang berada pada antrian paling depan akan dimasukkan ke toko. Pengguna dapat menerima, menunda, atau menolak permintaan.

### 5.11 Data Test STORE REMOVE

```
>> STORE REMOVE
```

```
Nama barang yang ingin dihapus: AK47
```

```
Barang dengan nama yang sama sudah ada di toko!
```

```
"AK47" telah dihapus dari toko.
```

```
ISI TOKO:
```

=====			
No	Nama Barang	Harga	
=====			
1	Lalabu	20	
2	Ayam Goreng Crisbar	20	
=====			

```
>> STORE REMOVE
```

```
Nama barang yang ingin dihapus: a
```

```
Barang tidak ditemukan di toko.
```

```
ISI TOKO:
```

=====			
No	Nama Barang	Harga	
=====			
1	Lalabu	20	
2	Ayam Goreng Crisbar	20	
=====			

Penjelasan: Tes ini dilakukan untuk menghapus barang yang ada di toko dengan menerima *input* barang yang akan dihapus.

### 5.12 Data Test HELP

```
>> HELP

===== [ Welcome Menu Help PURRMART ] =====
START      -> Untuk masuk sesi baru
LOAD       -> Untuk memulai sesi berdasarkan file konfigurasi
QUIT       -> Untuk keluar dari program
```

```
>> HELP

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

```
>> HELP

===== [ Main Menu Help PURRMART ] =====
WORK       -> Untuk bekerja
WORK CHALLENGE -> Untuk mengerjakan challenge
STORE LIST -> Untuk melihat barang-barang di toko
STORE REQUEST -> Untuk meminta penambahan barang
STORE SUPPLY -> Untuk menambahkan barang dari permintaan
STORE REMOVE -> Untuk menghapus barang
LOGOUT     -> Untuk keluar dari sesi
SAVE       -> Untuk menyimpan state ke dalam file
QUIT       -> Untuk keluar dari program
```

Penjelasan: Tes ini dilakukan untuk menampilkan daftar *command* untuk dieksekusi dengan deskripsinya.

### 5.13 Data Test SAVE

Fitur yang dites: SAVE <test1.txt>



```
>> SAVE test1.txt
```

```
Data berhasil disimpan ke ../save/test1.txt
```

```
1    4
2    10 AK47
3    20 Lalabu
4    20 Ayam Goreng Crisbar
5    10 Chitato
6    1
7    100 atha atha123
```

Penjelasan: Tes ini dilakukan untuk melakukan operasi *save* atau menyimpan state dari aplikasi untuk dieksekusi di kemudian hari.

#### 5.14 Data Test QUIT

```
>> QUIT
```

```
Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? Y
Tuliskan nama file tempat kamu ingin menyimpan: a.txt
Data berhasil disimpan ke ../save/a.txt
Kamu keluar dari PURRMART.
Dadah ^_^/
```

```
>> QUIT
```

```
Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? N
Data sesi tidak disimpan. Tetap keluar (Y/N)? Y
```

Penjelasan: Tes ini dilakukan untuk keluar dari sesi aplikasi PURRMART.

## 6 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	START	Memeriksa apakah <i>file config</i> default berhasil dibaca	<ul style="list-style-type: none"> <li>- Melakukan <i>compile</i> semua <i>file</i> yang berkaitan dengan <i>main.c</i> pada terminal</li> <li>- Menginput <i>command</i> START pada saat pertama kali memulai program PURRMART</li> </ul>	START	Program mengeluarkan "File konfigurasi aplikasi berhasil dibaca. PURRMART berhasil dijalankan."	Program mengeluarkan "File konfigurasi aplikasi berhasil dibaca. PURRMART berhasil dijalankan"
2	LOAD	Memeriksa apakah <i>file</i> berhasil dibaca	<ul style="list-style-type: none"> <li>- Melakukan <i>compile</i> semua <i>file</i> yang berkaitan dengan <i>main.c</i> pada terminal</li> <li>- Memasukkan <i>command</i> LOAD &lt;filename&gt;</li> </ul>	LOAD <filename>	Kondisi 1: Program mengeluarkan "Save file tidak ditemukan. PURRMART gagal dijalankan." Kondisi 2: "Save file berhasil dibaca. PURRMART berhasil dijalankan."	Kondisi 1: Program mengeluarkan "Save file tidak ditemukan. PURRMART gagal dijalankan." Kondisi 2: "Save file berhasil dibaca. PURRMART berhasil dijalankan."
3	LOGIN	Login ke dalam PURRMART dengan username yang sudah ada pada file konfigurasi	<ul style="list-style-type: none"> <li>- Pengguna sudah melakukan START/LOAD</li> <li>- Memasukkan <i>command</i> "LOGIN"</li> </ul>	LOGIN	Program meminta pengguna untuk memasukkan <i>username</i> dan <i>password</i> . Program mengeluarkan pesan "Anda telah login ke PURRMART sebagai <username>" ketika <i>command</i> valid.	Program meminta pengguna untuk memasukkan <i>username</i> dan <i>password</i> .  Kondisi 1: Program mengeluarkan pesan "Anda telah login sebagai <username>" jika benar dan belum masuk sebelumnya  Kondisi 2: Program mengeluarkan pesan "Anda masih tercatat sebagai <username>. Silahkan logout terlebih dahulu." saat

						<p>sudah login sebelumnya</p> <p>Kondisi 3: Program akan mengeluarkan “Username atau password salah” jika ada yang salah</p>
4	LOGOUT	Logout dari akun pengguna	<ul style="list-style-type: none"> <li>- Pengguna sudah <i>compile</i>, loaded, dan login.</li> <li>- Memasukkan <i>command</i> “LOGOUT”</li> </ul>	LOGOUT	Program mengeluarkan pesan “<username> telah logout dari sistem PURRMART. Silakan REGISTER/LOGIN kembali untuk melanjutkan.”	Program akan keluar dari fungsi dan mengeluarkan pesan “<username> telah keluar dari sistem PURRMART. Silakan LOGIN/REGISTER untuk melanjutkan”
5	REGISTER	Mendaftarkan akun baru ke dalam PURRMART	<ul style="list-style-type: none"> <li>- Pengguna sudah melakukan START/LOAD</li> <li>- Memasukkan <i>command</i> “REGISTER”</li> </ul>	REGISTER	<p>Program meminta pengguna untuk memasukkan <i>username</i> dan <i>password</i>. Program mengeluarkan “Akun dengan username &lt;username&gt; telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.” ketika <i>command</i> valid.</p>	<p>Program meminta pengguna untuk memasukkan <i>username</i> dan <i>password</i>.</p> <p>Kondisi 1: Program mengeluarkan “Akun dengan username &lt;username&gt; telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.” ketika <i>command</i> valid.</p> <p>Kondisi 2: Program akan mengeluarkan pesan “Akun dengan username &lt;username&gt; gagal dibuat. Silakan lakukan REGISTER ulang.” jika gagal</p>

6	WORK	Menampilkan pilihan daftar pekerjaan dan menambahkan pendapatan ke saldo pengguna	<ul style="list-style-type: none"> <li>- Pengguna sudah <i>compile</i>, loaded, dan login.</li> <li>- Memasukkan <i>command</i> “WORK”</li> </ul>	WORK	Program mengeluarkan pesan “Daftar pekerjaan: ‘....’ Masukkan pekerjaan yang dipilih: ‘....’ Anda sedang bekerja sebagai ‘....’ harap tunggu. Pekerjaan selesai ‘....’ rupiah telah ditambahkan ke akun anda.	<p>Program mengeluarkan pesan “Daftar pekerjaan: ‘....’ Masukkan pekerjaan yang dipilih: ‘....’ Anda sedang bekerja sebagai ‘....’ harap tunggu. Pekerjaan selesai ‘....’ rupiah telah ditambahkan ke akun anda.</p> <p>Jika input tidak valid, program mengeluarkan pesan “Pekerjaan tidak valid!”</p>
7	WORK CHALLENGE: Tebak Angka	Menampilkan permainan dengan sepuluh kesempatan untuk menebak angka yang benar dan menambahkan pendapatan ke saldo pengguna	<ul style="list-style-type: none"> <li>- Pengguna sudah <i>compile</i>, loaded, dan login.</li> <li>- Memasukkan <i>command</i> “WORK CHALLENGE”</li> <li>- Memasukkan “1” pada pilihan <i>challenge</i> yang hendak dimainkan</li> </ul>	WORK CHALLENGE	Program mengeluarkan pesan “Daftar challenge yang tersedia: ‘....’ Masukkan challenge yang hendak dimainkan: ‘....’ Tebak angka: ‘....’ Tebakanmu benar! ‘....’ rupiah telah ditambahkan ke akun anda.	<p>Program mengeluarkan pesan “Daftar challenge yang tersedia: ‘....’ Masukkan challenge yang hendak dimainkan: ‘....’ Tebak angka: ‘....’ Tebakanmu benar! ‘....’ rupiah telah ditambahkan ke akun anda.</p> <p>Jika input daftar challenge tidak valid, maka program akan mengeluarkan “Angka tidak valid! Kembali ke menu utama...”</p> <p>Jika uang tidak cukup, maka program mengeluarkan “Uang tidak cukup! Silakan bekerja dulu.”</p>
8	WORK CHALLENGE: WORDL3	Menampilkan permainan tebak kata berjumlah lima karakter	<ul style="list-style-type: none"> <li>- Pengguna sudah <i>compile</i>, loaded, dan login.</li> <li>- Memasukkan <i>command</i> “WORK CHALLENGE”</li> </ul>	WORK CHALLENGE	Program mengeluarkan pesan “WELCOME TO WORDL3,	Program mengeluarkan pesan “WELCOME TO WORDL3,

		dan menambahkan pendapatan ke saldo pengguna	- Memasukkan “2” pada pilihan <i>challenge</i> yang hendak dimainkan		YOU HAVE 5 CHANCES TO ANSWER BEFORE YOU LOSE! ‘....’ Masukkan kata tebakan Anda: ‘....’ Hasil: ‘....’ Selamat, Anda menang! ‘....’ rupiah telah ditambahkan ke akun Anda.”	YOU HAVE 5 CHANCES TO ANSWER BEFORE YOU LOSE! ‘....’ Masukkan kata tebakan Anda: ‘....’ Hasil: ‘....’ Selamat, Anda menang! ‘....’ rupiah telah ditambahkan ke akun Anda.”  Jika kalah, program mengeluarkan “Boo! Anda kalah. Jawabannya adalah ..”
9	STORE LIST	Menampilkan <i>command</i> list untuk melihat barang di dalam toko	- Pengguna sudah <i>compile</i> , loaded, dan login. - Memasukkan <i>command</i> STORE LIST	STORE LIST	Program mengeluarkan pesan sebagai berikut:  Kondisi 1: “List barang yang ada di toko : ‘....’” Kondisi 2: “TOKO KOSONG”	Program mengeluarkan pesan sebagai berikut:  Kondisi 1: “List barang yang ada di toko : ‘....’” Kondisi 2: “TOKO KOSONG”
10	STORE REQUEST	Meminta penambahan barang baru di dalam toko	- Pengguna sudah <i>compile</i> , loaded, dan login. - Memasukkan <i>command</i> STORE REQUEST	STORE REQUEST	Program mengeluarkan pesan “Nama barang yang diminta: ‘....’” ketika masukan tidak <i>valid</i> , akan mengeluarkan pesan “Barang dengan nama yang sama sudah ada di toko!” atau “Barang dengan nama yang sama sudah ada di antrian!”	Program mengeluarkan pesan “Nama barang yang diminta: ‘....’” ketika masukan tidak <i>valid</i> , akan mengeluarkan pesan “Barang dengan nama yang sama sudah ada di toko!” atau “Barang dengan nama yang sama sudah ada di antrian!”
11	STORE SUPPLY	Menambahkan barang baru ke dalam toko berdasarkan antrian permintaan	- Pengguna sudah <i>compile</i> , loaded, dan login. - Memasukkan <i>command</i> STORE SUPPLY	STORE SUPPLY	Program mengeluarkan pesan “Apakah kamu ingin menambahkan barang ‘....’ Harga barang: ‘....’” apabila <i>command valid</i>	Program mengeluarkan pesan “Apakah kamu ingin menambahkan barang ‘....’ Harga barang: ‘....’” apabila <i>command valid</i>

					maka program akan mengeluarkan pesan "<barang> dengan harga '.....' telah ditambahkan ke toko."	maka program akan mengeluarkan pesan "<barang> dengan harga '.....' telah ditambahkan ke toko."
12	STORE REMOVE	Menghapus barang yang ada di toko.	<ul style="list-style-type: none"> <li>- Pengguna sudah <i>compile</i>, loaded, dan login.</li> <li>- Memasukkan <i>command</i> STORE REMOVE</li> </ul>	STORE REMOVE	Program mengeluarkan pesan "Nama barang yang akan dihapus: '.....'". Ketika masukan valid, maka program akan mengeluarkan pesan "'.....' telah berhasil dihapus."	Program mengeluarkan pesan "Nama barang yang akan dihapus: '.....'". Ketika masukan valid, maka program akan mengeluarkan pesan "'.....' telah berhasil dihapus dari toko." Ketika input tidak ada di store, maka program akan mengeluarkan "Barang tidak ditemukan di toko."
13	HELP	Memunculkan daftar <i>command</i> yang yang dapat digunakan oleh pengguna	<ul style="list-style-type: none"> <li>- Memasukkan <i>command</i> STORE REMOVE</li> </ul>	HELP	Program mengeluarkan menu Help	Program mengeluarkan menu Help.
14	SAVE	Menyimpan state aplikasi terbaru	<ul style="list-style-type: none"> <li>- Pengguna sudah <i>compile</i>, loaded, dan login.</li> <li>- Memasukkan <i>command</i> "SAVE s.txt"</li> <li>- Memeriksa, mengikuti, dan memilih keluaran dari setiap menu</li> </ul>	SAVE <filename>	Program mengeluarkan pesan "Save file berhasil disimpan." dan file .txt muncul pada folder src/save	Program mengeluarkan pesan "Save file berhasil disimpan." dan file .txt muncul pada folder src/save.
15	QUIT	Keluar dari program	<ul style="list-style-type: none"> <li>- Memasukkan <i>command</i> "QUIT"</li> </ul>	QUIT	Program akan mengeluarkan banner dan keluar dari program.	Program akan mengeluarkan banner dan keluar dari program.

## 7 Pembagian Kerja dalam Kelompok

No	NIM - Nama	Pembagian Tugas
1	18221026 - Syasya Umaira	<ul style="list-style-type: none"> <li>• Membuat command start, command work challenge bagian Tebak Angka, dan command store remove</li> <li>• Membuat testing driver ADT List dan Custom</li> </ul>
2	18222041 - Kezia Caren Cahyadi	<ul style="list-style-type: none"> <li>• Membuat command work dan bonus Quantum WORLD3</li> </ul>
3	18223013 - Raka Adhitya Nugraha	<ul style="list-style-type: none"> <li>• Membuat ADT Custom, ADT Queue</li> <li>• Membuat command login, command quit</li> <li>• Membuat driver untuk ADT List, Listdin, Mesinkarakter dan Mesinkata</li> <li>• Membuat notulensi asistensi</li> <li>• Membuat Laporan bagian Ringkasan, Struktur Data (ADT) dan program utama</li> </ul>
4	18223037 - Michelle Hamdani	<ul style="list-style-type: none"> <li>• Membuat ADT Mesin Kata</li> <li>• Membuat command logout, command help</li> <li>• Membuat laporan bagian Test Script</li> </ul>
5	18223053 - Daffa Athalla Rajasa	<ul style="list-style-type: none"> <li>• Membuat ADT List</li> <li>• Membuat command register, command work challenge bagian WORDL3</li> <li>• Menulis laporan bagian data test dan lampiran (deskripsi tugas besar, <i>log activity</i> kelompok, dan pembagian kerja)</li> </ul>
6	18223079 - Atharizza Muhammad Athaya	<ul style="list-style-type: none"> <li>• Membuat ADT Mesin Karakter</li> <li>• Membuat command load, command store list, command store supply</li> <li>• Membuat bonus Bioweapon</li> <li>• Membuat file driver ADT</li> <li>• Mengintegrasikan seluruh fungsi kedalam file console dan main.</li> </ul>

		<ul style="list-style-type: none"> <li>• Membuat makefile.</li> <li>• Membuat laporan bagian spesifikasi fitur tambahan.</li> </ul>
--	--	---

## 8 Lampiran

### 8.1 Deskripsi Tugas Besar

#### 1. About the System

PURRMART adalah sebuah aplikasi yang dapat mensimulasikan aktivitas beli barang pada *e-commerce*. PURRMART memiliki beberapa fitur utama, yaitu:

- 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*
- Bekerja untuk menghasilkan uang

#### 2. Menu Program

Ketika program pertama kali dijalankan, PURRMART akan memperlihatkan *main menu* yang berisi **welcome menu** dan beberapa *command* yaitu **START**, **LOAD**, dan juga **HELP**.

Setelah itu, program akan memasuki **login menu** yang memiliki command **LOGIN**, **REGISTER**, dan juga **HELP**. Jika pengguna berhasil memasuki kredensial suatu akun, maka mereka akan masuk ke menu selanjutnya.

**Main menu** menerima masukan berupa *command* yang akan dijelaskan pada bagian berikutnya. Program akan terus menerima *command* sampai diberikan *command* **QUIT** yang berlaku pada seluruh menu.

#### 3. Command

Pengguna dapat memasukkan *command-command* berikut.

##### a. START



START merupakan salah satu command yang dimasukkan pertama kali dalam Toko PURRMART. Setelah menekan Enter, dibaca file konfigurasi *default* yang berisi daftar barang pada toko.

**b. LOAD <filename>**

LOAD merupakan salah satu *command* yang dimasukkan pertama kali dalam PURRMART. Command ini memiliki satu argumen yaitu *filename* yang merepresentasikan suatu *save file* yang ingin dibuka. *File* didapatkan dari *folder* tertentu, contohnya *save*. Setelah menekan *Enter*, akan dibaca *save file <filename>* yang berisi daftar barang pada toko. Lebih detailnya bisa dilihat pada [Konfigurasi Aplikasi](#).

**c. LOGIN**

Login merupakan *command* yang baru dapat dipanggil setelah pengguna memulai sesi. *Login* berguna untuk masuk ke akun di sistem PURRMART yang sudah didaftarkan sebelumnya.

**d. LOGOUT**

LOGOUT merupakan salah satu *command* yang baru dapat digunakan setelah pengguna telah memasuki sebuah sesi.

**e. REGISTER**

Register merupakan *command* yang baru dapat dipanggil setelah pengguna memulai sesi. *Register* berguna untuk mendaftarkan akun baru ke dalam sistem PURRMART. Sebuah akun setidaknya memiliki atribut *username* dan *password*. **Username dan password hanya terdiri dari 1 kata.**

**f. WORK**

WORK merupakan *command* yang digunakan pengguna untuk mendapatkan uang. Terdapat sejumlah pekerjaan yang bisa dipilih. Setiap pekerjaan memiliki waktu tunggu yang berbeda-beda dan dengan nominal pendapatan yang berbeda-beda pula. Selama pengguna sedang bekerja, maka sistem tidak bisa digunakan hingga pekerjaan selesai dilakukan.

**g. WORK CHALLENGE**

WORK CHALLENGE merupakan *command* alternatif sebagai cara mendapatkan uang dengan melakukan *challenge-challenge* di OWCA. Pemain membutuhkan uang dengan jumlah tertentu untuk bisa memainkan challenge. Uang yang dibayarkan untuk bermain *challenge* tidak akan dikembalikan, meskipun pemain kalah dalam permainan. Terdapat dua *challenge* yang dapat dipilih:

**a) Tebak Angka**

Challenge Tebak Angka merupakan permainan yang meminta pemain menebak sebuah angka yang ditentukan oleh program. Pemain memiliki 10 (sepuluh) kesempatan untuk menebak angka yang benar. Program akan memberikan *feedback* apakah angka tebakannya lebih besar, lebih kecil, atau sama dengan angka target. Jumlah kesempatan yang dipakai oleh pengguna akan mempengaruhi uang yang didapatkan.

**b) WORDL3**

Challenge WORDL3 merupakan permainan tebak kata berjumlah lima karakter. Pemain memiliki 6 (enam) kesempatan untuk menebak kata yang benar. Kata harus berupa kata valid, tidak boleh sekadar *string* acak, bahasa dibebaskan (disarankan bahasa Indonesia/Inggris). Pada setiap giliran, program akan mencetak ulang kata yang dimasukkan, tetapi dengan penanda tertentu. Huruf yang benar dan berada pada tempat yang tepat dicetak biasa. Huruf yang benar, tetapi berada di tempat yang salah diberi tanda "\*" setelah hurufnya. Huruf yang tidak ada sama sekali pada kata diberi tanda "%" setelah hurufnya.

**h. STORE LIST**

STORE LIST adalah *command* yang digunakan untuk melihat barang-barang apa saja yang ada di dalam toko. **Setiap barang yang ditampilkan haruslah bersifat *unique*.**

**i. STORE REQUEST**

STORE REQUEST adalah *command* yang digunakan untuk meminta penambahan barang baru ke dalam toko. Barang-barang yang diminta akan disimpan di dalam sebuah antrian dan akan dimasukkan ke toko menggunakan *command* selanjutnya. **Nama barang yang**

masuk tidak boleh sama dengan nama barang yang sudah ada di toko atau di antrian.

**j. STORE SUPPLY**

STORE SUPPLY adalah *command* yang digunakan untuk menambahkan barang baru ke dalam toko berdasarkan antrian permintaan. Barang yang berada pada antrian paling depan akan dimasukan ke toko. Pengguna dapat menerima, menunda, atau menolak permintaan.

- Jika diterima, maka program akan meminta harga dari barang dan dimasukan ke toko.
- Jika ditunda, maka barang akan kembali masuk ke antrian
- Jika ditolak, maka barang akan dihapus dari antrian

Harus terdapat validasi agar harga barang merupakan angka yang valid (berupa angka dan bernilai lebih dari nol).

**k. STORE REMOVE**

STORE REMOVE adalah *command* yang dapat menghapus barang yang ada di toko. Akan dilakukan *input* akan barang yang akan dihapus. Beri tahu apabila proses berhasil (barang terdapat pada toko dan berhasil dihapus) ataupun tidak (barang tidak terdapat di toko).

**l. HELP**

HELP merupakan *command* yang digunakan menampilkan daftar *command* yang mungkin untuk dieksekusi dengan deskripsinya. Penjelasan dari deskripsi dibebaskan selama masih mendeskripsikan *command* sesuai dengan spek.

**m. SAVE <filename>**





SAVE merupakan *command* yang digunakan untuk menyimpan *state* aplikasi terbaru ke dalam suatu *file*. Command SAVE memiliki satu argumen yang merepresentasikan nama *file* yang akan disimpan. Penyimpanan dilakukan pada *folder* tertentu, misal *folder save*.

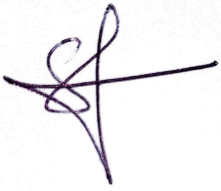
**n. QUIT**

QUIT merupakan *command* yang digunakan untuk keluar dari sesi aplikasi PURRMART.

## 8.2 Notulen Rapat

Asistensi I

<b>Tanggal : 22 November 2024 21.00-selesai</b>	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"> <li>- Buat ADT ada di satu folder, jadi semua ADT kaya list, queue, dibuat satu folder masing-masing.</li> <li>- Buat folder untuk masing-masing fungsi/command, biar gak berantakan. Di buat juga untuk file header dan file C nya.</li> <li>- Jangan lupa untuk rapikan dan perbaiki kode main.</li> <li>- Pindahkan file store request ke folder game,</li> <li>- <i>'Bagaimana untuk driver?'</i> Digunakan untuk mencoba masing-masing fungsi (buat main dan cek fungsionalitas program).</li> <li>- Harusnya buat branch dulu biar setiap orang bisa kerja, akan tetapi karena udah banyak juga jadi gapapa.</li> <li>- <i>'Untuk word challange dan world3, baik digabung apa dipisah? Kodenya dipisah apa digabung?'</i> Mending dipisah aja, ada tebak angka (header dan c) serta world3 (header dan c)</li> </ul>
<b>Tempat : GMeet(Online)</b>	
<b>Kehadiran Anggota Kelompok:</b>	
<p>1 18223079</p> 	
<p>2 18223013</p> 	
<p>3 18221026</p> 	
<p>4 18223037</p> 	
<p>5 18223053</p>	

 <p>6 18222041</p>	
<b>Dokumentasi</b> 	<b>Tanda Tangan Asisten:</b> 

### 8.3 Log Activity Anggota Kelompok

Tanggal	Anggota	Aktivitas
18/11/2024	18221026 Syasya Umaira 18222041 Kezia Caren Cahyadi 18223013 Raka Adhitya Nugraha 18223037 Michelle Hamdani 18223053 Daffa Athalla Rajasa 18223079 Atharizza Muhammad Athaya	Membahas spesifikasi tugas besar dan pembagian tugas besar
	18223079 Atharizza Muhammad Athaya	Initial commit Update README.md Add initial directory structure
19/11/2024	18223079 Atharizza Muhammad Athaya	Add MESINKAR and MESINKATA Add main and load. Edit MESINKATA logic

20/11/2024	18223037 Michelle Hamdani	Create command.h Delete src/ADT/command.h Add files via upload Delete src/command.h Update LOGOUT.c Update HELP.c
	18223079 Atharizza Muhammad Athaya	Add makeUser and makeBarang func Merge branch 'main' of <a href="https://github.com/ITBAtharizza/K01-11-TubesADT">https://github.com/ITBAtharizza/K01-11-TubesADT</a> Add ADT list and storing User's data Add ADT ListDin, pointers for lists Update Load() into separate file
	18223013 Raka Adhitya Nugraha	feat: Add custom, remove ADT from fungsi feat: Add queue, remove makeUser & makeBarang feat: Add quit feat: Add quit.h feat: Add comment at custom.h and listdin.h feat: Add login.h and login.c Merge remote-tracking branch 'refs/remotes/origin/main' feat: Add login feat: Repair Help
21/11/2024	18223079 Atharizza Muhammad Athaya	Repair login & mesinkar Add store list, update printlistdin Add Request, edit load and main Add supply Merge branch 'main' of <a href="https://github.com/ITBAtharizza/K01-11-TubesADT">https://github.com/ITBAtharizza/K01-11-TubesADT</a> Update main.c
	18221026 Syasya Umaira	add tebak angka & start
22/11/2024	18223079 Atharizza Muhammad	(Atha) Add register, repair main

	Athaya	Repair the structure Add console and update main
	18221026 Syasya Umaira	add store remove
	18223013 Raka Adhitya Nugraha	feat: Add Save feat: Repair Save feat: Complete Quit feat: Repair Save & Quit
	18221026 Syasya Umaira 18222041 Kezia Caren Cahyadi 18223013 Raka Adhitya Nugraha 18223037 Michelle Hamdani 18223053 Daffa Athalla Rajasa 18223079 Atharizza Muhammad Athaya	Asistensi I
23/11/24	18221026 Syasya Umaira	replace file adt add folder tebak angka add drive for adt list
	18223079 Atharizza Muhammad Athaya	Adjust include paths and remove files
24/11/24	18223053 Daffa Athalla Rajasa	Refactor console.h Feat wordl3 Refactor console.h
	18223079 Atharizza Muhammad Athaya	Repair mesinkar & move quit n save Repair logic on MESINKATA Repair mistypes