

**LAPORAN TUGAS BESAR**  
**IF2111 Algoritma dan Struktur Data STI**

**“PURRMART”**


Dipersiapkan oleh:

Kelompok 6

Izhar Alif Akbar / 18223129  
Harfhan Ikhtiar Ahmad R. / 18223123  
Stefany Josefina Santono / 18223116  
Nakeisha Valya Shakila / 18223133  
Sharon Darma Putra / 18223107  
Anggita Najmi Layali / 18223122

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB-K3-6		38
		Revisi	-	25/11/2024

# Daftar Isi

<b>1 Ringkasan</b>	<b>6</b>
<b>2 Penjelasan Tambahan Spesifikasi Tugas</b>	<b>7</b>
2.1 Bio Weapon (Bonus)	7
<b>3 Struktur Data (ADT)</b>	<b>9</b>
3.1 ADT Kustom	9
3.2 ADT List	10
3.3 ADT Mesin Karakter	11
3.4 ADT Mesin Kata	11
3.5 ADT Mesin Kalimat	12
3.6 ADT Queue	12
<b>4 Program Utama</b>	<b>13</b>
<b>5 Algoritma-Algoritma Menarik</b>	<b>18</b>
5.1 IsSubstring di Bio Weapon	18
5.2 TandaTebakan di WORLD3	19
<b>6 Data Test</b>	<b>19</b>
6.1 START	19
6.2 LOAD <filename>	20
6.3 LOGIN	21
6.4 LOGOUT	21
6.5 REGISTER	22
6.6 WORK	22
6.7 WORK CHALLENGE	23
6.8 TEBAK ANGKA	23
6.9 WORLD3	24
6.10 STORE LIST	25
6.11 STORE REQUEST	25
6.12 STORE SUPPLY	26
6.13 STORE REMOVE	27
6.14 HELP	27
6.15 SAVE	27
6.16 QUIT	28
6.17 BIO WEAPON (Bonus)	29
6.18 QUANTUM WORLD3 (Bonus)	29
<b>7 Test Script</b>	<b>31</b>

<b>8</b>	<b>Pembagian Kerja dalam Kelompok</b>	<b>33</b>
<b>9</b>	<b>Lampiran</b>	<b>34</b>
9.1	Deskripsi Tugas Besar	34
9.2	Notulen Rapat	35
9.3	Log Activity Anggota Kelompok	37

## Daftar Gambar

Gambar 2.2.1 Interface PURRMART	7
Gambar 6.1.1 Menampilkan START yang valid	18
Gambar 6.2.1 Menampilkan LOAD dengan file konfigurasi yang valid	19
Gambar 6.3.1 Menampilkan LOGIN yang valid	20
Gambar 6.4.1 Menampilkan LOGOUT yang valid	20
Gambar 6.5.1 Menampilkan REGISTER yang valid	21
Gambar 6.5.2 Menampilkan REGISTER yang tidak valid	21
Gambar 6.6.1 Menampilkan WORK yang valid	22
Gambar 6.6.2 Menampilkan WORK yang tidak valid	22
Gambar 6.7.1 Menampilkan WORK CHALLENGE yang valid	22
Gambar 6.7.2 Menampilkan WORK CHALLENGE yang tidak valid	22
Gambar 6.8.1 Menampilkan angka terlalu besar	23
Gambar 6.8.3 Menampilkan challenge telah selesai dan pengguna memenangkan challenge	23
Gambar 6.8.2 Menampilkan angka terlalu kecil	23
Gambar 6.8.4 Menampilkan challenge telah selesai dan pengguna tidak memenangkan challenge	23
Gambar 6.9.1 Menampilkan salah satu huruf benar	24
Gambar 6.9.2 Menampilkan semua huruf salah	24
Gambar 6.9.3 Menampilkan challenge telah selesai dan pengguna memenangkan challenge	24
Gambar 6.9.4 Menampilkan challenge telah selesai dan pengguna tidak memenangkan challenge	24
Gambar 6.10.1 Menampilkan STORE LIST	24
Gambar 6.11.1 Menampilkan barang unik	25
Gambar 6.11.2 Menampilkan barang yang sudah ada di toko	25
Gambar 6.11.3 Menampilkan barang yang sudah ada di antrian	25
Gambar 6.12.1 Menampilkan barang yang diterima	25
Gambar 6.12.2 Menampilkan barang yang ditolak	25
Gambar 6.12.3 Menampilkan barang yang ditunda	25
Gambar 6.13.1 Menampilkan barang yang tidak ada di toko	26
Gambar 6.13.2 Menampilkan barang yang ada di toko	26
Gambar 6.14.1 Menampilkan Panduan Penggunaan Program	26
Gambar 6.15.1 Menampilkan SAVE yang valid	26
Gambar 6.15.2 Menampilkan SAVE yang tidak valid	27
Gambar 6.16.1 Menampilkan QUIT dengan opsi N	27
Gambar 6.16.2 Menampilkan QUIT dengan opsi Y	27

Gambar 6.16.3 Menampilkan QUIT tanpa opsi apapun	27
Gambar 6.17.1 Menampilkan BIO WEAPON yang berhasil	28
Gambar 6.17.1 Menampilkan BIO WEAPON yang gagal	28

### ● **Gambar 6.18.1 Tampilan Program Quantum WORDL3**

Gambar 6.1.1 Menampilkan START yang valid	18
Gambar 6.2.1 Menampilkan LOAD dengan file konfigurasi yang valid	19
Gambar 6.3.1 Menampilkan LOGIN yang valid	19
Gambar 6.4.1 Menampilkan LOGOUT yang valid	20
Gambar 6.5.1 Menampilkan REGISTER yang valid	20
Gambar 6.5.2 Menampilkan REGISTER yang tidak valid	20
Gambar 6.6.1 Menampilkan WORK yang valid	21
Gambar 6.6.2 Menampilkan WORK yang tidak valid	21
Gambar 6.7.1 Menampilkan WORK CHALLENGE yang valid	21
Gambar 6.7.2 Menampilkan WORK CHALLENGE yang tidak valid	21
Gambar 6.8.1 Menampilkan angka terlalu besar	22
Gambar 6.8.3 Menampilkan challenge telah selesai dan pengguna memenangkan challenge	22
Gambar 6.8.2 Menampilkan angka terlalu kecil	22
Gambar 6.8.4 Menampilkan challenge telah selesai dan pengguna tidak memenangkan challenge	22
Gambar 6.9.1 Menampilkan salah satu huruf benar	23
Gambar 6.9.2 Menampilkan semua huruf salah	23
Gambar 6.9.3 Menampilkan challenge telah selesai dan pengguna memenangkan challenge	23
Gambar 6.9.4 Menampilkan challenge telah selesai dan pengguna tidak memenangkan challenge	23
Gambar 6.10.1 Menampilkan STORE LIST	23
Gambar 6.11.1 Menampilkan barang unik	24
Gambar 6.11.2 Menampilkan barang yang sudah ada di toko	24
Gambar 6.11.3 Menampilkan barang yang sudah ada di antrian	24
Gambar 6.12.1 Menampilkan barang yang diterima	24
Gambar 6.12.2 Menampilkan barang yang ditolak	24
Gambar 6.12.3 Menampilkan barang yang ditunda	24
Gambar 6.13.1 Menampilkan barang yang tidak ada di toko	25
Gambar 6.13.2 Menampilkan barang yang ada di toko	25
Gambar 6.14.1 Menampilkan Panduan Penggunaan Program	25
Gambar 6.15.1 Menampilkan SAVE yang valid	26

Gambar 6.15.2 Menampilkan SAVE yang tidak valid	26
Gambar 6.16.1 Menampilkan QUIT dengan opsi N	26
Gambar 6.16.2 Menampilkan QUIT dengan opsi Y	26
Gambar 6.16.3 Menampilkan QUIT tanpa opsi apapun	27
Gambar 6.17.1 Menampilkan BIO WEAPON yang berhasil	27
Gambar 6.17.1 Menampilkan BIO WEAPON yang gagal	27
Gambar 6.18.1 Tampilan Program Quantum WORDL3	28
Gambar 6.18.2 Menampilkan Quantum WORDL3 yang berhasil	28
Gambar 6.18.2 Menampilkan Quantum WORDL3 yang gagal	28

### **Daftar Tabel**

Tabel 1 Test Script	31
Tabel 2 Pembagian Kerja dalam Kelompok	32
Tabel 3 Log Activity	36

# 1 Ringkasan

PURRMART merupakan aplikasi simulasi berbasis command-line interface (CLI) yang dirancang untuk menggantikan proses operasional tatap muka di Toko Borma, sebuah pemasok strategis untuk kebutuhan OWCA dalam menghadapi ancaman Dr. Asep Spakbor. Sistem ini dibuat sebagai solusi inovatif untuk mensimulasikan aktivitas e-commerce, meliputi pengelolaan daftar barang, pemrosesan permintaan barang, penyimpanan dan pembelian di keranjang belanja, hingga pembuatan dan pengelolaan wishlist. Aplikasi ini bertujuan untuk meningkatkan efisiensi pengelolaan logistik OWCA, sekaligus menghadirkan simulasi yang interaktif dan mendidik.

Dari deskripsi permasalahan tersebut, kami mengembangkan PURRMART dengan lima fitur utama, yaitu menampilkan daftar barang, membuat permintaan barang, menyuplai barang baru ke toko, menghapus barang dari toko, serta melaksanakan pekerjaan atau tantangan untuk mendapatkan uang. Laporan ini mencakup deskripsi umum permasalahan, penjelasan tambahan mengenai spesifikasi tugas, struktur data yang digunakan seperti list, queue, dan mesin kata, penjelasan program utama, serta data dan script yang digunakan dalam pengujian aplikasi. Selain itu, pembagian tugas anggota tim serta dokumentasi rapat dan asistensi juga dilampirkan untuk mencatat proses pengembangan secara menyeluruh.

Aplikasi ini dibuat menggunakan bahasa pemrograman C dengan memanfaatkan ADT yang telah dipelajari dalam mata kuliah IF2111 Algoritma dan Struktur Data. Aplikasi diawali dengan menu utama yang memungkinkan pengguna untuk memulai aplikasi dengan command seperti *START*, membaca data dari file menggunakan *LOAD*, atau mendapatkan informasi melalui *HELP*. Setelah aplikasi dimulai, pengguna dapat menggunakan berbagai command seperti *store list* untuk melihat barang di toko, *store request* untuk menambahkan permintaan barang, *store supply* untuk memasukkan barang ke toko, *store remove* untuk menghapus barang, *work* untuk mendapatkan uang, serta *work challenge* untuk menjalankan tantangan. Command tambahan seperti *SAVE* dan *QUIT* juga tersedia untuk menyimpan data atau keluar dari aplikasi.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Bio Weapon (Bonus)

Fitur *Bio Weapon* diuji dengan menggunakan command *STORE REQUEST BIOWEAPON*, yang bertujuan untuk memasukkan senjata biologis ke dalam antrian barang dengan validasi kode rahasia dalam sekuens DNA. Pada data test pertama, senjata biologis bernama "Batuksius fififafae" diuji dengan sekuens DNA TACTAGAAGCATTAT dan kode rahasia MIFVI. Proses validasi melibatkan konversi DNA ke RNA (AUGAUCUUCGUAAUA), kemudian diterjemahkan menjadi protein dengan representasi satu huruf MIFVI, yang cocok dengan kode rahasia yang diberikan. Hasil yang diharapkan adalah senjata biologis berhasil dimasukkan ke dalam antrian,

dengan output: *"Senjata biologis mengandung kode. Barang akan ditambahkan ke dalam queue! Batuksius fififafae masuk ke dalam queue barang."*

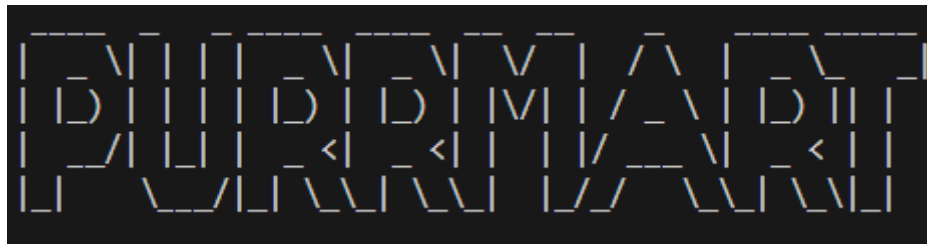
Pada data test kedua, senjata biologis bernama "Achoosius pilekae" diuji dengan sekuens DNA CCAATGAACCGCTCGGCATACCATGCCGCTGAT dan kode rahasia SKIBIDI. Dalam uji ini, kode rahasia tidak ditemukan dalam hasil konversi sekuens DNA yang diberikan. Hasil yang diharapkan adalah senjata biologis tidak dimasukkan ke dalam antrian, dengan output: *"Kode rahasia tidak ditemukan. Barang tidak akan ditambahkan ke dalam queue!"* Pengujian ini memastikan bahwa hanya senjata biologis dengan kode rahasia yang valid dapat masuk ke dalam antrian, sesuai dengan aturan fitur.

## 2.2 Quantum WORDL3

Fitur Quantum WORDL3 diuji dengan menggunakan command WORK CHALLENGE, yang bertujuan untuk memainkan permainan tebak kata dengan mekanisme menebak empat kata sekaligus dalam satu waktu. Pada data test pertama, pemain diberikan empat slot kosong untuk kata target, dan tebakan pertama adalah "ADIEU", "OMBRE", "PLUCK", dan "SERIN". Algoritma mengevaluasi setiap huruf pada kata yang ditebak, memberikan tanda penanda: huruf benar pada posisi yang tepat (ditampilkan biasa), huruf benar tetapi salah posisi (diberi tanda "\*"), dan huruf yang tidak ada dalam kata target (diberi tanda "%"). Setelah beberapa putaran, pemain berhasil menebak kata target dengan kombinasi "TRULY", "GLORY", "SIGMA", dan "MUSIC". Hasil yang diharapkan adalah pemain menang, dengan output: *"Selamat, Anda menang! +1500 rupiah telah ditambahkan ke akun Anda."*

Pada data test kedua, pemain diberikan kesempatan yang sama tetapi gagal menebak keempat kata target dalam batas maksimal 9 tebakan. Kombinasi terakhir dari pemain tidak mengungkap semua kata target, sehingga permainan berakhir dengan output: *"Boo! Anda kalah. Tidak ada hadiah untuk Anda."* Pengujian ini memastikan bahwa mekanisme permainan berjalan sesuai spesifikasi, dengan validasi posisi huruf yang tepat dan pemberian skor yang sesuai.

## 2.3 Interface



• **Gambar 2.2.1 Interface PURRMART**



PURRMART diatas sebagai identitas visual yang dirancang dengan seni ASCII, memadukan kreativitas dan estetika sederhana. Desain ini menjadikan "Purrmart" lebih dari sekadar nama namun merupakan sebuah elemen unik yang menarik perhatian dan memperkuat kesan modern.

### 3 Struktur Data (ADT)

Pada program PURRMART, kami memanfaatkan 6 jenis struktur data (ADT). Berikut merupakan beberapa ADT yang digunakan dalam program ini:

#### 3.1 ADT Kustom

##### **SKETSA STRUKTUR DATA :**

```
/* Kamus Umum */
#define MAX_LEN 100

typedef struct {
    char name[MAX_LEN];
    int price;
} Barang;

#endif
```

***barang.h***

```
/* Kamus Umum */
#define MAX_LEN 100

typedef struct {
    char name[MAX_LEN];
    char password[MAX_LEN];
    int money;
} User;

#endif
```

***user.h***

Dalam program Purrmart, kami menggunakan ADT Kustom untuk merepresentasikan dua entitas, yaitu **User** dan **Barang**. Entitas **User** dirancang dengan tiga atribut yaitu *name* berupa string untuk menyimpan nama pengguna, *password* berupa string untuk kata sandi pengguna, dan *money* berupa integer untuk mencatat jumlah uang yang dimiliki oleh pengguna. Sementara itu, Entitas **Barang** memiliki dua atribut utama, yaitu *name* berupa string untuk nama barang dan *price* berupa integer untuk harga barang. Dengan struktur ini, kami dapat mengelola data pengguna dan barang secara spesifik, yang memungkinkan kami untuk mengimplementasikannya pada command yang digunakan.

Kami mengimplementasikan kedua ADT ini dalam file bertipe header `user.h` dan `barang.h`. Pada `user.h`, ADT **User** didefinisikan sebagai struct dengan atribut *name*, *password*, dan *money*. Sedangkan pada `barang.h`, ADT **Barang** didefinisikan sebagai struct dengan atribut *name* dan *price*. Penggunaan kedua entitas tersebut digunakan untuk memisahkan setiap bagian program berdasarkan fungsinya.

### 3.2 ADT List

#### SKETSA STRUKTUR DATA :

```
#define boolean unsigned char
#define true 1
#define false 0

#define InitialSize 10

typedef int IdxType;
typedef Barang ElType2;
typedef struct {
    ElType2 *A;
    int Capacity;
    int Neff;
} ArrayDin;
```

#### *list.h*

Dalam program Purrmart, kami menggunakan ADT List yang terdiri dari dua jenis struktur data utama, yaitu **List Statis** dan **List Dinamis**, yang diimplementasikan menggunakan array dengan karakteristik berbeda berdasarkan pengalokasiannya. **List Statis** digunakan untuk setiap data yang tidak mengalami perubahan selama program berjalan sehingga memiliki kapasitas tetap dan jumlah elemen maksimum yang sudah ditentukan sejak awal. Sedangkan, **List Dinamis** digunakan untuk setiap data yang mengalami perubahan selama program berjalan sehingga memiliki ukuran yang memungkinkan array bertambah atau berkurang kapasitasnya sesuai kebutuhan.

Struktur data List Statis berada pada file bertipe header `list.h`. Di dalamnya, terdapat struktur **IdxType** digunakan untuk merepresentasikan indeks dalam array yang memiliki tipe data berupa integer. Struktur **ElType** digunakan untuk merepresentasikan indeks dalam array yang memiliki tipe data berupa User. Struktur **List** terdiri dari satu elemen utama, yaitu **A**, yang merupakan array statis dengan ukuran maksimum yang telah ditentukan oleh konstanta **MaxEl** yaitu sebanyak **100**. Array ini menyimpan elemen-elemen bertipe **ElType**.

Struktur data List Dinamis berada pada file bertipe header `arraydin.h`. Di dalamnya, terdapat struktur **IdxType** digunakan untuk merepresentasikan indeks dalam array yang memiliki tipe data berupa integer. Struktur **ElType2** digunakan untuk setiap elemen dalam array yang memiliki tipe data Barang. Sedangkan, struktur **ArrayDin** dari

tiga atribut berupa **A** yang merupakan pointer ke array dinamis bertipe **ElType2**, **Capacity** yang menyimpan kapasitas maksimum array bertipe integer, dan **Neff** yang menunjukkan jumlah elemen yang efektif bertipe integer.

### 3.3 ADT Mesin Karakter

#### SKETSA STRUKTUR DATA :

```
#define MARK '\n'

/* State Mesin Kata */
extern char currentChar;
extern boolean EOP;
```

*mesinkarakter.h*

Dalam program Purrmart, kami menggunakan ADT Mesin Karakter untuk membaca teks karakter demi karakter dari suatu pita, baik itu input langsung dari pengguna melalui stdin atau dari file. ADT Mesin Karakter ini berada pada file bertipe header *mesinkarakter.h*. Di dalamnya, terdapat dua state yaitu **currentChar** digunakan untuk merepresentasikan karakter yang sedang dibaca yang memiliki tipe data berupa character, dan **EOP (End of Process)**, digunakan untuk menunjukkan apakah pembacaan telah mencapai akhir pita yang memiliki tipe data boolean. Struktur pendukung lainnya yaitu **pita**, yang menjadi pointer ke sumber data (file atau standar input), dan **retval**, variabel untuk memantau hasil operasi pembacaan.

### 3.4 ADT Mesin Kata

#### SKETSA STRUKTUR DATA :

```
#define NMax 50
#define BLANK ' '

typedef struct
{
    char TabWord[NMax]; /* container penyimpan kata, indeks yang dipakai
[0..NMax-1] */
    int Length;
} Word;
/* State Mesin Kata */
extern boolean endWord;
extern Word currentWord;
```

*mesinkata.h*

Dalam program Purrmart, kami menggunakan ADT Mesin Kata digunakan untuk membaca teks pada satu kata demi satu kata dengan memanfaatkan **MesinKarakter** sebagai dasar untuk membaca per karakter. Mesin Kata juga dapat dimanfaatkan dalam

pengolahan teks dengan memisahkan kata-kata dari input, seperti mengabaikan spasi dan mengenali akhir input seperti MARK atau '!'. Selain itu, pada ADT Mesin Kata digunakan dalam melakukan *converting* tipe data tertentu agar sesuai dengan spesifikasi.

ADT Mesin Kata ini berada pada file bertipe header mesinkarakter.h. Di dalamnya, terdapat struktur data yang terdiri dari **Word**, yang memiliki atribut array bertipe character (TabWord) untuk menyimpan kata hingga panjang maksimum NMax, dan atribut Length bertipe integer untuk mencatat panjang kata yang sedang diproses. Selain itu, terdapat state berupa EndWord bertipe boolean yang menunjukkan apakah pembacaan telah mencapai akhir input, serta currentWord bertipe Word yang menyimpan kata yang sedang diproses.

### 3.5 ADT Mesin Kalimat

#### SKETSA STRUKTUR DATA :

```
#define NMaks 100
#define NEWLINE '\n'
#define MARK2 '\0'

typedef struct {
    char TabLine[NMaks+1];
    int Length;
} Kalimat;

extern boolean EndKalimat; // Deklarasi variabel global
extern Kalimat CLine;
extern Kalimat CInput;
extern Kalimat CCommand;
extern boolean endWord;
```

*mesinkalimat.h*

Dalam program Purrmart, kami menggunakan ADT Mesin Kalimat digunakan untuk membaca, memproses, dan menyimpan data teks berbasis baris. ADT ini memanfaatkan MesinKata sebagai dasar pemrosesan setiap kata-nya, yang dibuat dengan tujuan untuk mempermudah proses dalam melakukan load dan save suatu file config karena mampu membaca teks dari file ataupun input terminal, hingga menulis kembali teks ke file. Selain itu, ADT ini berfungsi untuk memisahkan bagian-bagian dalam kalimat, seperti angka atau kata, menggunakan fungsi parsing

ADT Mesin Kata ini berada pada file bertipe header mesinkalimat.h. Di dalamnya, terdapat struktur data yang terdiri dari Kalimat, yang memiliki atribut array bertipe character (TabLine) untuk menyimpan teks dan atribut Length bertipe integer untuk mencatat panjang teks tersebut. Selain itu, terdapat state berupa EndKalimat bertipe boolean yang menandakan apakah pembacaan kalimat sudah selesai atau belum, CLine bertipe Kalimat yang menyimpan kalimat yang sedang diproses, CInput bertipe Kalimat yang menginput kalimat dari pengguna, dan Command bertipe Kalimat yang memproses perintah yang ada.

### 3.6 ADT Queue

#### SKETSA STRUKTUR DATA :

```
#define IDX_UNDEF -1
#define CAPACITY 100
/* Definisi elemen dan address */
typedef Barang ElType2;
typedef struct
{
    ElType2 buffer[CAPACITY];
    int idxHead;
    int idxTail;
} Queue;
```

*queue.h*

Dalam program Purrmart, kami menggunakan ADT Queue digunakan untuk memproses kalimat atau input dalam program dengan menggunakan konsep **Queue** (antrian). Queue berfungsi untuk menampung elemen kalimat (seperti kata atau simbol) secara berurutan dan diproses mengikuti prinsip First In and First Out, di mana elemen pertama yang masuk adalah yang pertama kali keluar. Selain itu, ADT ini juga mampu berfungsi untuk menambahkan elemen ke antrian, menghapus elemen, serta mengecek apakah antrian kosong atau penuh .

ADT Mesin Kata ini berada pada file bertipe header mesinkalimat.h. Di dalamnya, terdapat dua struktur data yang terdiri dari Eltype2 bertipe Barang. Dan Queue yang memiliki buffer untuk menyimpan elemen-elemen di dalam antrian bertipe ElType2, dan array bertipe integer dengan dua indeks, yaitu idxHead untuk menandakan elemen pertama dalam antrian dan idxTail untuk menandakan elemen terakhir.

## 4 Program Utama

Program utama dimulai dengan pendefinisian header yang diperlukan untuk spesifikasi program, yaitu <stdio.h>, <stdlib.h>, dan "boolean.h". Selain itu, terdapat header tambahan yang berasal dari ADT dan Spesifikasi fungsi yang telah dibuat. Header dari ADT meliputi barang.h , user.h ,mesinkarakter.h , mesinkata.h , queue.h , arraydin.h , list.h , dan mesinkalimat.h. Sementara itu, header untuk Spesifikasi meliputi load.h , save.h , login.h , logout.h , help.h , register.h , StoreList.h , StoreRemove.h , StoreRequest.h , StoreSupply.h , work.h , workchallenge.h, dan StoreRequestBioWeapon.h.

Setelah pendefinisian header, program dilanjutkan dengan deklarasi fungsi main() yang menjadi pusat dari seluruh proses. Fungsi ini diawali dengan inisialisasi struktur data seperti List userList, ArrayDin barangList, dan Queue barangQueue. Selain itu, variabel penting lainnya, seperti currentIndex, filename, dan level, juga diinisialisasi. Program dimulai dengan memanggil fungsi printPurrmart() untuk menampilkan interface.

Pada bagian utama program, terdapat proses looping untuk menjalankan berbagai command yang diinputkan pengguna berdasarkan tingkatan menu. Pada Level 1, pengguna dapat memasukkan command START, LOAD, HELP, atau QUIT. Command START akan memuat file default, sedangkan command LOAD memungkinkan pengguna memuat file khusus yang ditentukan. Jika command tidak sesuai, maka pesan error akan ditampilkan, dan pengguna diarahkan untuk menggunakan command HELP.

Selanjutnya, pada Level 2, pengguna dapat memilih command LOGIN, REGISTER, HELP, atau QUIT. Pada tahap ini, pengguna dapat melakukan autentikasi dengan command LOGIN atau mendaftarkan akun baru menggunakan REGISTER. Setelah berhasil login, pengguna akan diarahkan ke Level 3.

Pada Level 3, pengguna memiliki akses ke command utama seperti WORK, STORE, LOGOUT, SAVE, QUIT, dan HELP. Command WORK memungkinkan pengguna untuk bekerja, termasuk opsi CHALLENGE. Command STORE menawarkan berbagai operasi terkait barang, seperti menampilkan daftar barang (LIST), meminta barang (REQUEST), memasok barang (SUPPLY), atau menghapus barang (REMOVE). Command lain, seperti SAVE, digunakan untuk menyimpan data ke file, sementara QUIT memberikan opsi untuk keluar dari program dengan menyimpan data terlebih dahulu jika diinginkan.

Setiap command yang valid akan memanggil fungsi/prosedur yang sesuai untuk dieksekusi. Jika command tidak dikenali, program akan memberikan peringatan dan mengarahkan pengguna untuk memasukkan command yang valid. Program akan terus berjalan hingga pengguna memilih untuk keluar dengan command QUIT. Sebelum benar-benar keluar, program akan menampilkan antarmuka penutup melalui fungsi printClosing().

Berikut ini adalah algoritma dari program utama yang kami buat dengan file yang diberi nama “main.c”:

```
#include <stdio.h>
#include <stdlib.h>
#include "boolean.h"

/*-----HEADER ADT-----*/
#include "ADT/Kustom/barang.h"
#include "ADT/Kustom/user.h"
#include "ADT/MesinKarakter/mesinkarakter.h"
#include "ADT/MesinKata/mesinkata.h"
#include "ADT/Queue/queue.h"
#include "ADT/ListDinamis/arraydin.h"
#include "ADT/List/list.h"
#include "ADT/MesinKalimat/mesinkalimat.h"

/*-----HEADER SPESIFIKASI-----*/
#include "SPESIFIKASI/Load/load.h"
#include "SPESIFIKASI/Save/save.h"
#include "SPESIFIKASI/Login/login.h"
#include "SPESIFIKASI/Logout/logout.h"
#include "SPESIFIKASI/Help/help.h"
```

```

#include "SPESIFIKASI/Register/register.h"
#include "SPESIFIKASI/Help/help.h"
#include "SPESIFIKASI/StoreList/StoreList.h"
#include "SPESIFIKASI/StoreRemove/StoreRemove.h"
#include "SPESIFIKASI/StoreRequest/StoreRequest.h"
#include "SPESIFIKASI/StoreSupply/StoreSupply.h"
#include "SPESIFIKASI/Work/work.h"
#include "SPESIFIKASI/workchallenge/workchallenge.h"
#include "SPESIFIKASI/StoreRequestBioWeapon/StoreRequestBioWeapon.h"

int main()
{
    printPurrmart();

    // INISIALISASI STRUKTUR DATA
    IdxType currentIndex = IDX_UNDEF;
    char filename[MAX_LEN];
    List userList = MakeList();
    ArrayDin barangList = MakeArrayDin();
    Queue barangQueue;
    CreateQueue(&barangQueue);

    // ALGORITMA

    int level = 1;
    boolean isRunning = true;
    printPurrmart();

    while (isRunning)
    {
        if (level == 1)
        {
            PrintLevel1Menu();
            printf("\n> ");
            STARTWORD();

            // Tingkatan 1: START, LOAD, HELP, QUIT
            if (IsWordEqual(currentWord, StringToWord("START")))
            {
                char defaultFilename[] = "default.txt";
                Load(defaultFilename, &barangList, &userList);
                level = 2;
            }
            else if (IsWordEqual(currentWord, StringToWord("LOAD")))
            {
                printf("Nama File (.txt): ");
                printf("\n> ");
                STARTWORD();
                for (int i = 0; i < currentWord.Length; i++)
                {
                    filename[i] = currentWord.TabWord[i];
                }
                filename[currentWord.Length] = '\0';
                Load(filename, &barangList, &userList);
                if (!EndKalimat)
                {
                    printf("File berhasil dimuat. Masuk ke autentikasi pengguna.\n");
                    level = 2;
                }
            }
            else
            {
                printf("Gagal memuat file.\n");
            }
        }
    }
}

```

```

    }
    else if (IsWordEqual(currentWord, StringToWord("HELP")))
    {
        DisplayHelp1();
    }
    else if (IsWordEqual(currentWord, StringToWord("QUIT")))
    {
        isRunning = false;
    }
    else
    {
        printf("Command tidak dikenal. Silakan masukkan command yang valid.\n");
    }
}
else if (level == 2)
{
    PrintLevel2Menu();
    printf("\n> ");
    STARTWORD();

    // Tingkatan 2: LOGIN, REGISTER, HELP, QUIT
    if (IsWordEqual(currentWord, StringToWord("LOGIN")))
    {
        Login(&userList, &currentIndex);
        ;
        level = 3;
    }
    else if (IsWordEqual(currentWord, StringToWord("REGISTER")))
    {
        RegisterUser(&userList);
    }
    else if (IsWordEqual(currentWord, StringToWord("HELP")))
    {
        DisplayHelp2();
    }
    else if (IsWordEqual(currentWord, StringToWord("QUIT")))
    {
        isRunning = false;
    }
    else
    {
        printf("Command tidak dikenal. Silakan masukkan command yang valid.\n");
    }
}
else if (level == 3)
{
    PrintLevel3Menu();
    printf("\n> ");
    STARTWORD();

    // Tingkatan 3: WORK, STORE, LOGOUT, SAVE, QUIT
    if (IsWordEqual(currentWord, StringToWord("WORK")))
    {
        ADVWORD();
        if (IsWordEqual(currentWord, StringToWord("CHALLENGE")))
        {
            WorkChallenge(&userList, currentIndex); // Done testing
        }
        else
        {
            Work(&userList, currentIndex);
        }
    }
    else if (IsWordEqual(currentWord, StringToWord("STORE")))

```



```

{
    ADVWORD();
    if (IsWordEqual(currentWord, StringToWord("LIST")))
    {
        StoreList(&barangList);
    }
    else if (IsWordEqual(currentWord, StringToWord("REQUEST")))
    {
        ADVWORD();
        if (IsWordEqual(currentWord, StringToWord("BIO")))
        {
            ADVWORD();
            if (IsWordEqual(currentWord, StringToWord("WEAPON")))
            {
                StoreRequestBioWeapon(&barangQueue, &barangList);
            }
            else
            {
                printf("Command tidak dikenal.\n");
            }
        }
        else if (endWord)
        {
            StoreRequest(&barangQueue, &barangList);
        }
        else{
            printf("Command tidak dikenal.\n");
        }
    }
    else if (IsWordEqual(currentWord, StringToWord("SUPPLY")))
    {
        StoreSupply(&barangQueue, &barangList);
    }
    else if (IsWordEqual(currentWord, StringToWord("REMOVE")))
    {
        StoreRemove(&barangList);
    }
    else
    {
        printf("Command tidak dikenal.\n");
    }
}
else if (IsWordEqual(currentWord, StringToWord("BIOWEAPON")))
{
    StoreRequestBioWeapon(&barangQueue, &barangList);
}
else if (IsWordEqual(currentWord, StringToWord("LOGOUT")))
{
    Logout(currentIndex);
    level = 2;
}
else if (IsWordEqual(currentWord, StringToWord("SAVE")))
{
    Save(&barangList, &userList, filename);
    printf("Berhasil menyimpan data!");
}
else if (IsWordEqual(currentWord, StringToWord("QUIT")))
{
    printf("Apakah Anda ingin menyimpan sesi ini? (Y/N)\n");
    boolean quit = false;
    while (!quit)
    {
        printf("> ");
        STARTWORD();
    }
}

```

```

        if (IsWordEqual(currentWord, StringToWord("Y")))
        {
            Save(&barangList, &userList, filename);
            printf("Berhasil menyimpan data!");
            quit = true;
        }
        else if (IsWordEqual(currentWord, StringToWord("N")))
        {
            quit = true;
        }
        else
        {
            printf("Command tidak dikenal. Silakan masukkan command yang
valid.\n");
            printf("Apakah Anda ingin menyimpan sesi ini? (Y/N)\n");
        }
        isRunning = false;
    }
}
else if (IsWordEqual(currentWord, StringToWord("HELP")))
{
    DisplayHelp3();
}
else
{
    printf("Command tidak dikenal. Silakan masukkan command yang valid.\n");
}
}

printClosing();
return 0;
}

```

## 5 Algoritma-Algoritma Menarik

### 5.1 *IsSubstring di Bio Weapon*

```

boolean IsSubstring(Word sequence, Word subSequence) {
    if (subSequence.Length > sequence.Length) return false;

    for (int i = 0; i <= sequence.Length - subSequence.Length; i++) {
        boolean match = true;
        for (int j = 0; j < subSequence.Length; j++) {
            if (sequence.TabWord[i + j] != subSequence.TabWord[j]) {
                match = false;
                break;
            }
        }
        if (match) return true;
    }

    return false;
}

```

Fungsi **IsSubstring** merupakan inti dalam validasi kode rahasia pada proses **StoreRequestBioWeapon**. Fungsi ini memeriksa apakah kode rahasia, berupa string tertentu, terdapat dalam hasil translasi protein yang diperoleh dari urutan DNA senjata biologis (merupakan *substring* dari hasil translasi protein). Jika kode ditemukan, barang dianggap valid dan ditambahkan ke dalam antrian, sementara jika tidak, barang ditolak karena dianggap telah disabotase. Dalam konteks ini, fungsi **IsSubstring** memastikan validitas senjata biologis dengan mendeteksi pola spesifik dalam protein secara efisien. Keberadaannya penting untuk menjamin kontrol kualitas barang dan menjaga integritas proses validasi.

## 5.2 TandaTebakan di *WORLD3*

```
void TandaTebakan(char *guess, Word *wordAns, char grid[5][6], int attempt,
char answer[6]) {

    for (int i = 0; i < 5; i++) {
        if (guess[i] == answer[i]) {
            grid[attempt][i] = guess[i];
        } else if (CharExist(guess[i], wordAns)) {
            grid[attempt][i] = '#';
        } else {
            grid[attempt][i] = '*';
        }
    }
    grid[attempt][5] = '\0';
}
```

Fungsi **TandaTebakan** digunakan untuk memeriksa dan memberi tanda pada setiap huruf dalam tebakan yang dimasukkan oleh pemain dalam permainan. Fungsi ini membandingkan setiap huruf dari tebakan dengan kata yang benar, yang tersimpan dalam parameter **wordAns**. Jika huruf dalam tebakan tepat berada di posisi yang benar, maka huruf tersebut ditambahkan ke dalam grid tanpa tanda apa pun. Jika huruf ada dalam kata jawaban tetapi tidak pada posisi yang tepat, maka huruf tersebut digantikan dengan simbol #. Sedangkan jika huruf tidak ada sama sekali dalam kata jawaban, akan digantikan dengan simbol \*. Fungsi **TandaTebakan** sangat penting untuk memberi umpan balik yang jelas kepada pemain mengenai kesalahan dalam tebakan mereka, sekaligus membantu mereka memperbaiki jawaban pada kesempatan berikutnya. Dengan cara ini, fungsi ini membantu pemain untuk mengevaluasi setiap tebakan secara lebih efektif dan mendekati jawaban yang benar.

## 6 Data Test

### 6.1 *START*

Program *Purrmart* dimulai dengan memasukkan command **START** pada terminal yang telah di-*run* sebelumnya. Perlu diperhatikan bahwa penulisan command harus sesuai

dengan opsi yang ada, jika tidak maka program akan meminta input ulang seperti pada *Gambar 6.1.1*. command ini digunakan untuk membaca file konfigurasi default yang tersimpan dalam **default.txt**. File ini berisi daftar barang-barang yang tersedia di toko dan dapat digunakan selama permainan berlangsung. Setelah file konfigurasi berhasil dibuka, program akan menampilkan pesan bahwa file telah berhasil dibaca, diikuti dengan daftar command yang bisa digunakan. Daftar command yang tersedia yaitu **LOGIN** untuk masuk ke akun, **REGISTER** untuk mendaftarkan akun baru, **HELP** untuk melihat panduan penggunaan, dan **QUIT** untuk keluar dari program.

```

=====
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 26
```

```

=====
|                               |
| WELCOME TO PURRMART         |
|                               |
|-----|
| Commands:                   |
| > START                     |
| > LOAD                      |
| > HELP                      |
| > QUIT                      |
|                               |
|-----|
Silakan pilih perintah:
> LOAD
Nama File (.txt):
> dummy.txt
File berhasil dimuat. Masuk ke autentikasi pengguna.

```

```

=====
|                               |
| WELCOME TO PURRMART         |
|                               |
|-----|
| Commands:                   |
| > START                     |
| > LOAD                      |
| > HELP                      |
| > QUIT                      |
|                               |
|-----|
Silakan pilih perintah:
> LOAD
Nama File (.txt):
> load.pdf
Gagal memuat file.

```

Gambar 6.2.2 Menampilkan LOAD dengan file konfigurasi yang tidak valid

- **Gambar 6.2.1 Menampilkan LOAD dengan file konfigurasi yang valid**

### 6.3 LOGIN

Setelah pengguna memasukkan command **START**, command **LOGIN** dapat digunakan untuk mengakses seluruh fitur yang tersedia pada program Purrmart. Dalam proses login, pengguna diminta untuk memasukkan **username** dan **password** yang telah terdaftar di file konfigurasi. Jika **password** yang dimasukkan tidak sesuai, program akan menampilkan pesan: *"Password salah. Coba lagi."* Sebaliknya, jika **username** yang dimasukkan tidak ditemukan dalam daftar yang terdaftar, program akan memberikan pesan: *"Username tidak ditemukan. Coba lagi."* Pastikan bahwa **username** dan **password** yang dimasukkan benar agar pengguna dapat masuk ke akun dan menggunakan program untuk mengakses fitur-fitur di dalam Purrmart.

```
=====
|                                     |
| AUTENTIKASI PENGGUNA               |
|-----|
| Commands:                          |
| > LOGIN                            |
| > REGISTER                         |
| > HELP                             |
| > QUIT                             |
|-----|
| Silakan pilih perintah:            |
| > LOGIN                            |
| Masukkan Username: admin           |
| Masukkan Password: alstrukdatkeren |
| Login berhasil. Selamat datang, admin |
| !                                  |
|=====|
|                                     |
| PURRMART - MAIN MENU               |
|-----|
| Commands:                          |
| > WORK                             |
| > WORK CHALLENGE                   |
| > STORE LIST                       |
| > STORE REQUEST                    |
| > STORE REQUEST BIOWEAPON           |
| > STORE SUPPLY                     |
| > STORE REMOVE                     |
| > LOGOUT                           |
| > SAVE                             |
| > HELP                             |
| > QUIT                             |
|-----|
| Silakan pilih perintah:            |
| > |
```

• **Gambar 6.3.1 Menampilkan LOGIN yang valid**

```
=====
|                                     |
| AUTENTIKASI PENGGUNA               |
|-----|
| Commands:                          |
| > LOGIN                            |
| > REGISTER                         |
| > HELP                             |
| > QUIT                             |
|-----|
| Silakan pilih perintah:            |
| > LOGIN                            |
| Masukkan Username: praktikan       |
| Masukkan Password: kerenbangetdek  |
| Password salah. Coba lagi.         |
| Masukkan Username: 4dmin           |
| Username tidak ditemukan. Coba lagi.|
| Masukkan Username: oke             |
| Username tidak ditemukan. Coba lagi.|
|-----|
```

*Gambar 6.3.2 Menampilkan LOGIN yang tidak valid*

## 6.4 LOGOUT

Setelah pengguna berhasil masuk menggunakan command **LOGIN**, command **LOGOUT** dapat digunakan untuk keluar dari akun dan mengakhiri sesi pengguna. Jika command yang dimasukkan sesuai, seperti ditunjukkan pada *Gambar 6.4.1*, program akan menampilkan pesan: *"Logout berhasil. Sampai jumpa!"* dan secara otomatis akan kembali ke tampilan **START**. Namun, jika command yang dimasukkan tidak valid, seperti pada *Gambar 6.4.2*, program akan memberikan pesan: *"Command tidak dikenal. Silakan masukkan command yang valid."*

```
Silakan pilih perintah:
> LOGOUT
Logout berhasil. Sampai jumpa!

=====
|                                     |
| AUTENTIKASI PENGGUNA               |
|-----|
| Commands:                          |
| > LOGIN                            |
| > REGISTER                         |
| > HELP                             |
| > QUIT                             |
|-----|
| Silakan pilih perintah:            |
| > |
```

```
Silakan pilih perintah:
> logout
Command tidak dikenal. Silakan masukkan command yang valid.

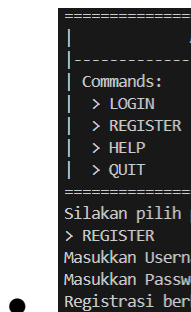
=====
|                                     |
| PURRMART - MAIN MENU               |
|-----|
| Commands:                          |
| > WORK                             |
| > WORK CHALLENGE                   |
| > STORE LIST                       |
| > STORE REQUEST                    |
| > STORE REQUEST BIOWEAPON           |
| > STORE SUPPLY                     |
| > STORE REMOVE                     |
| > LOGOUT                           |
| > SAVE                             |
| > HELP                             |
| > QUIT                             |
|-----|
| Silakan pilih perintah:            |
| > |
```

Gambar 6.4.2 Menampilkan LOGOUT yang tidak valid

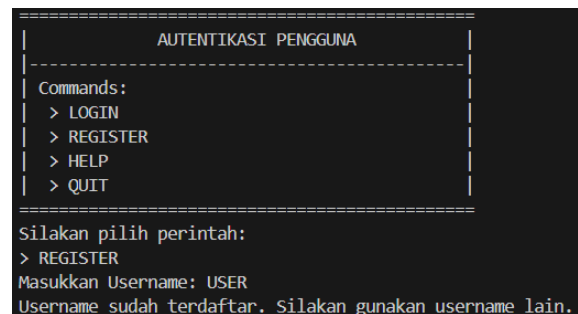
- **Gambar 6.4.1 Menampilkan LOGOUT yang valid**

## 6.5 REGISTER

Setelah pengguna memasukkan command **START**, command **REGISTER** dapat digunakan untuk membuat akun baru jika pengguna belum memiliki akun untuk login. Dalam proses registrasi akun, setiap **username** yang dibuat harus bersifat unik, sedangkan **password** tidak harus unik dan dapat dibuat sama dengan akun lain. Jika proses pendaftaran berhasil, seperti pada *Gambar 6.5.1*, program akan menampilkan "*Registrasi berhasil, silakan login.*" Namun, jika pendaftaran gagal karena **username** yang dimasukkan sudah terdaftar sebelumnya seperti pada *Gambar 6.5.2*, program akan menampilkan "*Username sudah terdaftar. Silakan gunakan username lain.*" .



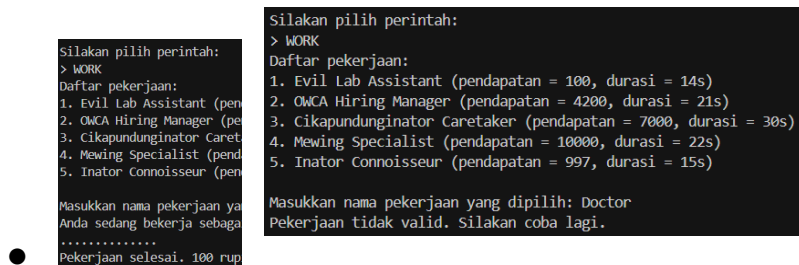
- **Gambar 6.5.1 Menampilkan REGISTER yang valid**



- **Gambar 6.5.2 Menampilkan REGISTER yang tidak valid**

## 6.6 WORK

Setelah pengguna berhasil melakukan **LOGIN**, command **WORK** dapat digunakan untuk menampilkan daftar pekerjaan yang tersedia dengan informasi pendapatan dan durasi waktu yang diperlukan untuk menyelesaikannya. Pengguna dapat memilih salah satu pekerjaan dari daftar tersebut dengan memasukkan nama pekerjaan yang diinginkan. Jika pekerjaan yang dipilih sesuai dengan daftar, seperti pada *Gambar 6.6.1*, program akan menampilkan *"Anda sedang bekerja sebagai <Nama Pekerjaan>... Harap tunggu selama <Durasi> detik."* Selama durasi tersebut, pengguna perlu menunggu hingga pekerjaan selesai. Setelah waktu selesai, program akan menampilkan *"Pekerjaan selesai. <Pendapatan> rupiah telah ditambahkan ke saldo Anda."* menandakan bahwa pendapatan dari pekerjaan tersebut telah ditambahkan ke saldo pengguna. Namun, jika nama pekerjaan yang dimasukkan tidak sesuai dengan daftar yang tersedia, seperti yang ditunjukkan pada *Gambar 6.6.2*, program akan memberikan pesan: *"Pekerjaan tidak valid. Silakan coba lagi."*



● **Gambar 6.6.1 Menampilkan WORK yang valid**

● **Gambar 6.6.2 Menampilkan WORK yang tidak valid**

## 6.7 WORK CHALLENGE

Setelah pengguna berhasil melakukan **LOGIN**, command **WORK CHALLENGE** dapat digunakan untuk menampilkan daftar challenge yang tersedia yaitu Tebak Angka dan WORDL3 dengan informasi biaya main. Selain itu, ada pula opsi quit apabila pengguna tidak ingin memainkan challenge. Pengguna dapat memilih salah satu challenge dari daftar tersebut dengan memasukkan nama challenge yang diinginkan dalam bentuk nomor urutan challenge. Jika saldo pengguna cukup untuk bermain, seperti pada *Gambar 6.7.1*, program akan menampilkan challenge yang harus diselesaikan oleh pengguna. Namun, jika saldo pengguna tidak cukup untuk bermain, seperti pada *Gambar 6.7.2*, program akan menampilkan *"Uang Anda tidak cukup untuk memainkan game ini."*



```
Silakan pilih p
> WORK CHALLENGE

WELCOME TO WORK
Daftar challenge
1. Tebak Angka
2. WORDL3 (biaya
3. Keluar dari v
Saldo Anda saat
>> 1
Tebak angka: █
```

- **Gambar 6.7.1 Menampilkan WORK CHALLENGE yang valid**

```
Silakan pilih p
> WORK CHALLENGE

WELCOME TO WORK
Daftar challenge
1. Tebak Angka
2. WORDL3 (biaya
3. Keluar dari v
Saldo Anda saat
>> 1
Uang Anda tidak
```

- **Gambar 6.7.2 Menampilkan WORK CHALLENGE yang tidak valid**

## 6.8 **TEBAK ANGKA**

Setelah pengguna berhasil melakukan **WORK CHALLENGE**, input 1 yang merepresentasikan challenge **TEBAK ANGKA** dapat digunakan untuk memainkan challenge menebak satu angka random dalam 10 kali percobaan, apabila pengguna memasukkan angka yang lebih besar seperti pada *Gambar 6.8.1*, maka akan menampilkan “Terlalu besar!” dan “Sisa percobaan : <Jumlah percobaan yang tersisa>”. Namun, jika pengguna memasukkan angka yang lebih kecil seperti pada *Gambar 6.8.2*, maka akan menampilkan “Terlalu kecil!” dan “Sisa percobaan : <Jumlah percobaan yang tersisa>”. Apabila pengguna berhasil menebak angka seperti pada *Gambar 6.8.3* maka pengguna akan mendapatkan reward berupa penambahan saldo dan menampilkan “Selamat! Anda berhasil menebak angka: <Angka yang benar>” dan “Selamat! <Saldo> rupiah telah ditambahkan ke akun Anda.”, semakin besar sisa percobaan saat menebak angka, maka semakin besar pula saldo yang didapat. Sebaliknya, apabila pengguna gagal menebak angka seperti pada *Gambar 6.8.4*, maka tidak akan mendapatkan penambahan ataupun pengurangan saldo dan menampilkan “Sayang sekali! Angka yang benar adalah: <Angka yang benar>” dan “Sayang sekali, Anda tidak mendapatkan hadiah.”.

```
Tebak angka: 20

Terlalu besar!
Sisa percobaan: 7
```

```
Terlalu besar!
Sisa percobaan: 0

Sayang sekali! Angka yang benar adalah: 90
Sayang sekali, Anda tidak mendapatkan hadiah.
```

- 

- **Gambar 6.8.3**  
**Menampilkan**  
**challenge telah selesai**  
**dan pengguna**  
**memenangkan**  
**challenge**

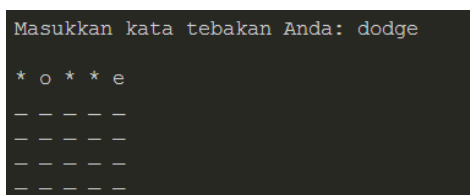
```
Tebak angka: 17  
Terlalu kecil!  
Sisa percobaan: 6
```

```
Tebak angka: 18  
Selamat! Anda berhasil menebak angka: 18  
Selamat! 380 rupiah telah ditambahkan ke akun Anda.
```

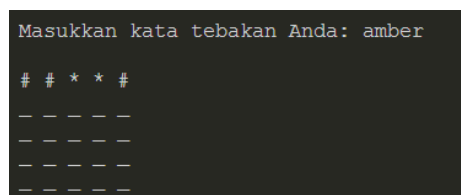
- 
- **Gambar 6.8.4**  
**Menampilkan**  
**challenge telah selesai**  
**dan pengguna tidak**  
**memenangkan**  
**challenge**

## 6.9 WORLD3

Setelah pengguna berhasil melakukan **WORK CHALLENGE**, input 2 yang merepresentasikan challenge WORDL3 dapat digunakan untuk memainkan challenge menebak satu kata random dengan 5 huruf dalam 5 kali percobaan sesuai dengan tampilan baris yang tersedia, apabila pengguna memasukkan kata dengan beberapa urutan huruf yang benar tetapi kata nya masih salah seperti pada *Gambar 6.9.1*, maka akan menampilkan letak huruf yang benar nya saja. Namun, jika pengguna memasukkan kata tanpa ada huruf yang benar seperti pada *Gambar 6.9.2*, maka akan menampilkan simbol-simbol seperti \* dan #. Apabila pengguna berhasil menebak angka seperti pada *Gambar 6.9.3* maka pengguna akan mendapatkan reward berupa penambahan saldo dan menampilkan “*Selamat! <Saldo> rupiah telah ditambahkan ke akun Anda.*”, semakin besar sisa percobaan saat menebak kata, maka semakin besar pula saldo yang didapat. Sebaliknya, apabila pengguna gagal menebak kata seperti pada *Gambar 6.9.4*, maka tidak akan mendapatkan penambahan ataupun pengurangan saldo dan menampilkan “*OH TIDAK Anda kehabisan kesempatan! Jawaban yang benar adalah: <Kata yang benar>*” dan “*Sayang sekali, Anda tidak mendapatkan hadiah.*”.



● **Gambar 6.9.1**  
**Menampil**  
**kan salah**



● **Gambar 6.9.2**  
**Menampi**  
**lkan**  
**semua**

**satu huruf  
benar**

**huruf  
salah**

```
Masukkan kata tebakan Anda: house
SELAMAT, ANDA MENEBAK KATA: house
Selamat! 2700 rupiah telah ditambahkan ke akun Anda.
```

- **Gambar 6.9.3 Menampilkan challenge telah selesai dan pengguna memenangkan challenge**

```
OH TIDAK Anda kehabisan kesempatan! Jawaban yang benar
Sayang sekali, Anda tidak mendapatkan hadiah.
```

- **Gambar 6.9.4 Menampilkan challenge telah selesai dan pengguna tidak memenangkan challenge**

## **6.10 STORE LIST**

Setelah pengguna berhasil melakukan **LOGIN**, command **STORE LIST** dapat digunakan untuk menampilkan daftar barang yang tersedia di toko dan setiap barang bersifat unik sehingga tidak terdapat duplikasi

```
Silakan pilih perintah:
> STORE LIST
- AK47
- Lalabu
- Ayam Goreng Crisbar
```

- **Gambar 6.10.1 Menampilkan STORE LIST**

## **6.11 STORE REQUEST**

Setelah pengguna berhasil melakukan **LOGIN**, command **STORE REQUEST** dapat digunakan untuk menambahkan barang yang baru dan tidak ada di daftar barang ataupun didalam antrian barang (barang yang sudah direquest sebelumnya) seperti pada *Gambar 6.11.1*. Apabila barang yang di-request sudah ada di toko seperti pada *Gambar 6.11.2* maka akan menampilkan “*Barang dengan nama yang sama sudah di toko!*”, kemudian apabila barang yang di-request sudah pernah di-request sebelumnya seperti pada *Gambar 6.11.3* maka akan menampilkan “*Barang dengan nama yang sama sudah di antrian!*”.

- 

- **Gambar 6.11.1 Menampilkan barang unik**

- 

- **Gambar 6.11.2 Menampilkan barang yang sudah ada di toko**

- 

- **Gambar 6.11.3 Menampilkan barang yang sudah ada di antrian**

## 6.12 STORE SUPPLY

Setelah pengguna berhasil melakukan **LOGIN**, command **STORE SUPPLY** dapat digunakan untuk mengkonfirmasi barang yang telah di-request pada **STORE REQUEST** akan diterima, ditolak, atau ditunda. Setiap barang yang diterima seperti pada *Gambar 6.12.1*, maka pengguna harus menetapkan harga barang dan menampilkan “<Nama barang> dengan harga <Harga barang> telah ditambahkan ke toko.” Kemudian apabila barang ditolak seperti pada *Gambar 6.12.2*, maka akan menampilkan “<Nama barang> telah dihapus dari antrian.”. Dan apabila barang ditunda seperti pada *Gambar 6.12.3*, maka akan menampilkan “<Nama barang> dikembalikan ke antrian.”.

- **Gambar 6.12.1 Menampilkan barang yang diterima**

```
Silakan pilih perintah:
> STORE SUPPLY
Apakah kamu ingin menambahkan barang Gomilk? (Terima, Tolak, Tunda): Tolak
Gomilk telah dihapus dari antrian.
```

- **Gambar 6.12.2 Menampilkan barang yang ditolak**

```
Silakan pilih perintah:
> STORE SUPPLY
Apakah kamu ingin menambahkan barang Gomilk? (Terima, Tolak, Tunda): Tunda
Gomilk dikembalikan ke antrian.
```

- **Gambar 6.12.3 Menampilkan barang yang ditunda**

## 6.13 STORE REMOVE

Setelah pengguna berhasil melakukan **LOGIN**, command **STORE REMOVE** dapat digunakan untuk menghapus setiap barang yang berada pada toko. Apabila barang yang ingin dihapus ada di toko seperti pada *Gambar 6.13.1*, maka akan menampilkan “*Toko tidak menjual: <Nama Barang>*” Namun, apabila barang yang ingin dihapus tidak ada di toko seperti *Gambar 6.13.2*, maka akan menampilkan “*<Nama barang> berhasil dihapus*”.

```
Silakan pilih perintah:
> STORE REMOVE
Nama barang yang akan dihapus: ambalabu
Toko tidak menjual: ambalabu
```

```
Silakan pilih perintah:
> STORE REMOVE
Nama barang yang akan dihapus: AK47
AK47
telah berhasil dihapus
```

- **Gambar 6.13.1 Menampilkan barang yang tidak ada di toko**

- **Gambar 6.13.2 Menampilkan barang yang ada di toko**

## 6.14 HELP

Setelah pengguna berhasil melakukan **LOGIN**, command **HELP** dapat digunakan untuk menjadi panduan pengguna dalam menggunakan command-command yang dapat diakses setelah **LOGIN**. Command **HELP** menjelaskan fungsi dan contoh setiap command yang dapat diakses oleh pengguna seperti pada *Gambar 6.14.1*.

```
Silakan pilih perintah:
> HELP

=====
PANDUAN PENGGUNAAN PROGRAM
=====

1. WORK
   Fungsi: Untuk bekerja.
   Contoh: Ketik WORK untuk bekerja dan mendapatkan uang.
2. WORK CHALLENGE
   Fungsi: Untuk mengerjakan challenge.
   Contoh: Ketik WORK CHALLENGE untuk menyelesaikan tantangan dan mendapatkan uang.
3. STORE LIST
   Fungsi: Untuk melihat barang-barang di toko.
   Contoh: Ketik STORE LIST untuk menampilkan daftar barang.
4. STORE REQUEST
   Fungsi: Untuk meminta penambahan barang ke toko.
   Contoh: STORE REQUEST
5. STORE REQUEST BIOWEAPON
   Fungsi: Untuk meminta penambahan senjata biologis ke toko.
   Contoh: STORE REQUEST BIOWEAPON
6. STORE SUPPLY
   Fungsi: Untuk menambahkan barang berdasarkan permintaan.
   Contoh: STORE SUPPLY
7. STORE REMOVE
   Fungsi: Untuk menghapus barang dari toko.
   Contoh: STORE REMOVE
8. LOGOUT
   Fungsi: Untuk keluar dari sesi.
   Contoh: Ketik LOGOUT untuk keluar.
9. SAVE
   Fungsi: Untuk menyimpan state program ke dalam file.
   Contoh: SAVE
10. QUIT
   Fungsi: Keluar dari program.
   Contoh: QUIT
```

- **Gambar 6.14.1 Menampilkan Panduan Penggunaan Program**

## 6.15 SAVE

Setelah pengguna berhasil melakukan **LOGIN**, command **SAVE** dapat digunakan untuk menyimpan seluruh data yang sudah ter-*update* selama menggunakan fitur di dalam Purrmart kedalam file **.txt** yang baru sehingga nama file konfigurasi harus bersifat unik dan belum pernah digunakan seperti pada *Gambar 6.15.1*, maka akan menampilkan “Berhasil menyimpan data!”. Namun, apabila file konfigurasi telah digunakan atau tidak dalam format **.txt** seperti pada *Gambar 6.15.2*, maka akan menampilkan “Nama file harus berakhiran .txt!” dan “Nama file tidak boleh berawalan '<Nama file>!'”.

```
Silakan pilih perintah:
> SAVE
Masukkan nama file untuk menyimpan data (harus .txt dan tidak berawalan 'default'): alstrupkdead.txt
Berhasil menyimpan data!=====
```

- **Gambar 6.15.1 Menampilkan SAVE yang valid**

```
Silakan pilih perintah:
> SAVE
Masukkan nama file untuk menyimpan data (harus .txt dan tidak berawalan 'default'): default.txt
Nama file tidak boleh berawalan 'default'!
Silakan coba lagi.
```

- **Gambar 6.15.2 Menampilkan SAVE yang tidak valid**



## 6.16 QUIT

Command **QUIT** tersedia di beberapa tampilan seperti di awal program, setelah pengguna berhasil melakukan **START**, dan setelah pengguna berhasil melakukan **LOGIN**. command **QUIT** dapat digunakan apabila pengguna ingin mengakhiri program dan setiap pengguna akan mengakhiri program, maka pengguna diberikan opsi untuk menyimpan sesi atau tidak seperti pada *Gambar 6.16.1* dan *Gambar 6.16.2*, ketika pengguna ingin menyimpan sesi maka perlu ada input file konfigurasi baru yang unik dalam bentuk file **.txt**. Namun, apabila pengguna melakukan **QUIT** ketika pengguna tidak menggunakan command yang berkaitan dengan perubahan data seperti pada *Gambar 6.16.3*, maka tidak akan ada opsi untuk menyimpan sesi atau tidak karena program tidak mengubah apapun dan tidak ada sesi yang disimpan (kosong).

```
Silakan pilih perintah:
> QUIT
Apakah Anda ingin menyimpan sesi ini? (Y/N)
> N

=====
THANK YOU!
=====

Terima kasih telah mengunjungi PURRMART!
Sampai jumpa lagi!
```

```
Silakan pilih perintah:
> QUIT
Apakah Anda ingin menyimpan sesi ini? (Y/N)
> Y
Masukkan nama file untuk menyimpan data (harus .txt dan tidak berawalan 'default'): loss.txt
Berhasil menyimpan data!=====
THANK YOU!
=====

Terima kasih telah mengunjungi PURRMART!
Sampai jumpa lagi!
```

- **Gambar 6.16.1 Menampilkan QUIT dengan opsi N**

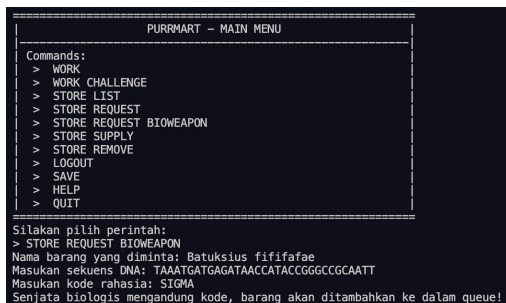
- **Gambar 6.16.2 Menampilkan QUIT dengan opsi Y**



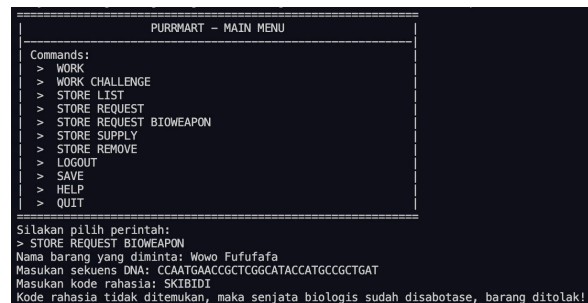
- **Gambar 6.16.3 Menampilkan QUIT tanpa opsi apapun**

## 6.17 BIO WEAPON (Bonus)

Setelah pengguna berhasil melakukan **LOGIN**, command **BIO WEAPON** dapat digunakan untuk memasukkan barang baru ke dalam antrian dengan validasi kode rahasia dalam sekuens DNA. Setiap barang yang diminta perlu unik dan belum ada di ada di toko ataupun di antrian barang. Setiap sekuens DNA yang di-*input* harus sesuai dengan hasil konversi RNA-nya, apabila DNA sesuai dengan hasil terjemahan RNA-nya seperti *Gambar 6.17.1*, maka akan menampilkan “*Senjata biologis mengandung kode, barang akan ditambahkan ke dalam queue!*”. Namun, apabila tidak sesuai maka seperti *Gambar 6.17.2*, maka akan menampilkan “*Kode rahasia tidak ditemukan, maka senjata biologis sudah disabotase, barang ditolak!*”.



- **Gambar 6.17.1 Menampilkan BIO WEAPON**



- **Gambar 6.17.2 Menampilkan BIO WEAPON yang gagal**

## ON yang berhasil

### 6.18 QUANTUM WORLD3 (Bonus)

Setelah pengguna berhasil melakukan **LOGIN**, command **QUANTUM WORLD** berupa *work challenge* ekstra bagi pemain. Berbeda dengan WORDL3 yang terdapat pada spesifikasi wajib, fitur Quantum WORDL3 memungkinkan pemain untuk menebak empat kata sekaligus dalam satu waktu (keempat kata harus dijamin berbeda). Pada permainan ini, pemain diberi kesempatan untuk menebak sebanyak 9 kali.

```
=====
WELCOME TO QUANTUM WORDL3!
Aturan main:
1. Terdapat 9 kesempatan.
2. Setiap tebakan harus merupakan kata valid 5 huruf kecil (lowercase) bahasa Inggris.
3. * menunjukkan huruf tidak ada di dalam jawaban
4. # menunjukkan huruf ada di dalam jawaban tetapi posisi salah
5. Jika huruf benar dan posisi benar, huruf akan dikembalikan tanpa tanda

=====
Percobaan yang tersisa: 9
Kata 1:
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----

Kata 2:
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----

Kata 3:
-----
-----
-----
-----
-----
-----
-----
-----
-----
=====
```

• **Gambar 6.18.1 Tampilan Program Quantum WORDL3**

```
Masukkan kata tebakan Anda: youth

Percobaan yang tersisa: 5
Kata 1:
z * * * #
z * * * #
z * * * *
* o * * *
-----
-----
-----
-----
-----

Kata 2 benar: zesty
Kata 3 benar: zebra
Kata 4 benar: youth

Masukkan kata tebakan Anda: zonal
=====
SELAMAT! Anda menebak semua kata dengan benar dalam 5 percobaan.
=====
Selamat! Anda berhasil menebak semua kata dan mendapatkan 5000 rupiah!
```

- **Gambar 6.18.2 Menampilkan Quantum WORDL3 yang berhasil**

```
Masukkan kata tebakan Anda: aiueo

=====
ANDA KEHABISAN KESEMPATAN!
Jawaban yang benar adalah:
Kata 1: zebra
Kata 2: youth
Kata 3: zonal
Kata 4: zesty

Anda berhasil menebak 0 dari 4 kata.

=====
Sayang sekali, Anda tidak mendapatkan hadiah. Coba lagi!

WELCOME TO WORK CHALLENGE!
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main=200)
2. WORDL3 (biaya main=500)
3. Quantum WORDL3 (biaya main=1000)
4. Keluar dari work challenge (quit)
Saldo Anda saat ini: 53000
```

- **Gambar 6.18.3 Menampilkan Quantum WORDL3 yang gagal**

## 7 Test Script

No	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Fitur Start	Menguji apakah command START dapat menjalankan keseluruhan program PURRMART.	Mengetikkan command START pada terminal.	<u>START</u>	Fungsi berhasil mengeluarkan output yang diharapkan dan berhasil membaca file config.txt	Sesuai yang diharapkan
2	Fitur Load	Menguji apakah command LOAD dapat membaca file dan berfungsi dengan baik.	Mengetikkan command LOAD dilanjutkan dengan file txt konfigurasinya	<u>LOAD</u> <u>&lt;filename&gt;</u>	Fungsi berhasil dijalankan, mengeluarkan output yang diharapkan, dan berhasil memuat file eksternal	Sesuai yang diharapkan
3	Fitur Login	Memastikan pengguna dapat login dengan username dan	Mengetikkan command LOGIN kemudian memasukkan username dan	<u>LOGIN</u>	Fungsi berhasil memverifikasi kredensial pengguna dan login berhasil	Sesuai yang diharapkan

		password yang valid.	password yang valid.			
4	Fitur Logout	Memastikan pengguna dapat keluar dari sesi login tanpa masalah.	Mengetikkan command LOGOUT.	<a href="#">LOGOUT</a>	Fungsi berhasil mengeluarkan pengguna dan mengubah status ke level login	Sesuai yang diharapkan
5	Fitur Register	Memastikan pengguna baru dapat mendaftar dengan username dan password.	Mengetikkan command REGISTER kemudian memasukkan username dan password baru.	<a href="#">REGISTER</a>	Fungsi berhasil mendaftarkan akun baru dan mengkonfirmasi pendaftaran	Sesuai yang diharapkan
6	Fitur Work	Memastikan pengguna dapat bekerja dan mendapatkan uang.	Mengetikkan command WORK pada menu dan memilih pekerjaan yang tersedia.	<a href="#">WORK</a>	Fungsi berhasil memberikan pekerjaan dan memperbarui status pengguna	Sesuai yang diharapkan
7	FiturWork Challenge	Memastikan pengguna dapat memainkan mini-game untuk uang tambahan.	Mengetikkan command WORK CHALLENGE	<a href="#">WORK CHALLENGE</a>	Fungsi berhasil memunculkan pilihan menu <b>Tebak Angka</b> dan <b>World3</b>	Sesuai yang diharapkan
8	Fitur Tebak Angka	Memastikan pengguna dapat bermain mini-game Tebak Angka untuk hadiah uang.	Mengetikkan command TEBAK ANGKA dan angka tebakan	<a href="#">TEBAK ANGKA</a>	Fungsi berhasil memulai mini-game <b>Tebak Angka</b> , memvalidasi tebakan pengguna dan memberikan hasil (menang/kalah)	Sesuai yang diharapkan
9	Fitur World3	Memastikan pengguna dapat bermain mini-game WORDL3 untuk hadiah uang.	Mengetikkan command WORLD3 dan kata tebakan	<a href="#">WORLD3</a>	Fungsi berhasil menjalankan permainan teka-teki <b>World3</b> dan memvalidasi hasil permainan	Sesuai yang diharapkan
10	Fitur Store List	Memastikan daftar barang di toko dapat ditampilkan.	Mengetikkan command STORE LIST	<a href="#">STORE LIST</a>	Fungsi berhasil menampilkan daftar barang yang ada di toko	Sesuai yang diharapkan
11	Fitur Store Request	Memastikan pengguna dapat mengajukan permintaan barang baru ke toko.	Mengetikkan command STORE REQUEST dan memasukkan nama barang	<a href="#">STORE REQUEST</a>	Fungsi berhasil menerima permintaan barang dan menambahkannya ke antrian	Sesuai yang diharapkan
12	Fitur Store Supply	Memastikan barang dari antrian dapat	Mengetikkan command STORE SUPPLY dan	<a href="#">STORE SUPPLY</a>	Fungsi berhasil menyuplai barang ke toko	Sesuai yang diharapkan

		ditambahkan ke toko.	memilih barang yang akan disuplai			
13	Fitur Store Remove	Memastikan barang dapat dihapus dari daftar toko.	Mengetikkan command STORE REMOVE dan memilih barang yang akan dihapus	<a href="#">STORE REMOVE</a>	Fungsi berhasil menghapus barang dari toko	Sesuai yang diharapkan
14	Fitur Help	Memastikan daftar command dan penjelasan dapat ditampilkan.	Mengetikkan command HELP pada level 1 atau 2, atau 3	<a href="#">HELP</a>	Fungsi berhasil menampilkan daftar command yang valid untuk setiap level, dengan penjelasan singkat	Sesuai yang diharapkan
15	Fitur Save	Memastikan state aplikasi dapat disimpan ke file.	Mengetikkan command SAVE diikuti dengan filename	<a href="#">SAVE</a>	Fungsi dapat menyimpan state aplikasi terbaru ke dalam suatu file	Sesuai yang diharapkan
16	Fitur Quit	Memastikan aplikasi dapat keluar dengan opsi menyimpan.	Mengetikkan command QUIT dan memilih untuk menyimpan atau tidak	<a href="#">QUIT</a>	Fungsi berhenti, dan menanyakan apakah akan menyimpan sesi yang sedang berlangsung	Sesuai yang diharapkan
17	Fitur Store Request Bio Weapon (Bonus)	Memeriksa apakah senjata biologis dapat dimasukkan ke antrian barang toko.	Mengetikkan command STORE REQUEST BIO WEAPON untuk senjata biologis	<a href="#">BIO WEAPON (Bonus)</a>	Fungsi berhasil menambahkan senjata biologis ke antrian barang toko dengan validasi DNA dan kode rahasia.	Sesuai yang diharapkan
18	Fitur Quantum WORLD3	Memeriksa apakah mini-game Quantum WORLD3 dapat dimainkan dengan benar	Mengetikkan command QUANTUM WORLD3	<a href="#">QUANTUM WORLD3 (Bonus)</a>	Fungsi berhasil menjalankan versi khusus WORLD3 dengan mekanisme kuantum, memberikan tantangan dan hasil	Sesuai yang diharapkan

*Tabel 1 Test Script*

## 8 Pembagian Kerja dalam Kelompok

Nama Lengkap / NIM	Deskripsi Tugas
Izhar Alif Akbar / 18223129	Program main, ADT, Command Load, Command Start, Command Login, Command Bio Weapon, Command Quantum WORLD3

Harfhan Ikhtiar Ahmad R. / 18223123	Command Register, Command Help, Command Save
Stefany Josefina Santono / 18223116	Command Logout, Command Store Remove, Menyusun Laporan
Nakeisha Valya Shakila / 18223133	Command Store List, Command Store Request, Command Store Supply, Menyusun Laporan
Sharon Darma Putra / 18223107	Program main, Command Work Challenge, Command Tebak Angka, Command WORDL3, Command Quantum WORLD3
Anggita Najmi Layali / 18223122	Command Load, Command Start, Command Work

*Tabel 2 Pembagian Kerja dalam Kelompok*

## 9 Lampiran

### 9.1 Deskripsi Tugas Besar

PURRMART adalah sebuah aplikasi simulasi berbasis CLI (Command-Line Interface). Sistem ini dibuat menggunakan bahasa C dengan memanfaatkan struktur data yang telah dipelajari selama perkuliahan. Struktur data yang digunakan meliputi ADT Kustom, ADT List, ADT Queue, dan Mesin Kata, yang dimanfaatkan untuk mengelola barang, pengguna, serta berbagai fitur aplikasi. PURRMART juga memungkinkan penggunaan file konfigurasi untuk membaca data awal sistem seperti daftar barang dan pengguna. Library yang diperbolehkan hanya **stdio.h**, **stdlib.h**, **math.h**, dan **time.h**.

#### System Mechanic

##### 1. About the System

PURRMART adalah aplikasi simulasi e-commerce yang dapat digunakan untuk:

- Menampilkan barang di toko.
- Mengajukan dan menyuplai barang baru ke toko.
- Membeli barang dan menyimpannya di keranjang.
- Menampilkan barang yang telah dibeli.
- Membuat dan menghapus wishlist.
- Menghasilkan uang melalui pekerjaan dan tantangan mini-game.

##### 2. Menu Program

Saat pertama kali dijalankan, PURRMART menampilkan main menu dengan command **START**, **LOAD**, dan **HELP**. Setelah itu, program memasuki login menu dengan command **LOGIN**, **REGISTER**, dan **HELP**. Setelah pengguna berhasil login, mereka dapat mengakses berbagai fitur tambahan melalui main menu.

### 3. Command

- **START**: Membaca file konfigurasi default untuk menginisialisasi daftar barang yang tersedia di toko.
- **LOAD <filename>**: Membaca file konfigurasi tertentu untuk memuat daftar barang dan pengguna dari state sebelumnya.
- **LOGIN**: Memungkinkan pengguna untuk login ke dalam sistem menggunakan username dan password.
- **REGISTER**: Mendaftarkan akun baru dengan username dan password untuk digunakan pada sistem.
- **WORK**: Memberikan kesempatan kepada pengguna untuk bekerja dan mendapatkan uang sesuai dengan pekerjaan yang dipilih.
- **WORK CHALLENGE**: Menawarkan mini-game untuk mendapatkan uang tambahan, seperti Tebak Angka atau WORDL3.
- **STORE LIST**: Menampilkan barang-barang yang tersedia di toko.
- **STORE REQUEST**: Mengajukan permintaan barang baru ke toko.
- **STORE SUPPLY**: Menyediakan barang dari antrian permintaan ke toko dengan opsi menerima, menunda, atau menolak barang.
- **STORE REMOVE**: Menghapus barang dari daftar toko.
- **SAVE <filename>**: Menyimpan state terbaru aplikasi ke file yang ditentukan.
- **QUIT**: Keluar dari aplikasi dengan pilihan untuk menyimpan data saat ini.
- **HELP**: Menampilkan daftar command yang tersedia beserta penjelasannya

### 4. Daftar ADT

- **ADT Kustom**: Untuk merepresentasikan data barang (Barang) dan pengguna (User).
- **ADT List**: Untuk menyimpan daftar barang dan pengguna, menggunakan list statis untuk pengguna dan list dinamis untuk barang.
- **ADT Queue**: Untuk mengelola permintaan barang yang diusulkan pengguna.
- **Mesin Kata**: Untuk membaca input dari file konfigurasi dan memproses command pengguna.

### 5. Bonus

- **QuantumWORDL3**: Fitur tambahan berupa challenge dengan menebak empat kata sekaligus.
- **Bio Weapon**: Fitur untuk menambahkan senjata biologis ke toko melalui analisis kode DNA





## 9.2 Notulen Rapat

**Form Asistensi Tugas Besar  
IF2111/Algoritma dan Struktur Data STI  
Sem. 1 2024/2025**

No. Kelompok/Kelas : 06 / 03  
Nama Kelompok : 6acor  
Anggota Kelompok (Nama/NIM) :  
1. Izhar Alif Akbar / 18223129  
2. Harfhan Ikhtiar Ahmad R. / 18223123  
3. Stefany Josefina Santono / 18223116  
4. Nakeisha Valya Shakila / 18223133  
5. Sharon Darma Putra / 18223107  
6. Anggita Najmi Layali / 18223122

Asisten Pembimbing : Jonathan Arthurito Aldi Sinaga

### Asistensi I

<b>Tanggal : 20 November 2024</b>	<b>Catatan Asistensi:</b>  Mesin karakter pita tidak akan tertutup sampai program selesai. Jadi, setelah menerima input, mesin tidak akan menerima input lagi. Pita bisa ditutup jika programnya sudah selesai, tapi itu terserah lagi ke implementasi kalian.  Isi dari fungsi utama (main) cukup dibuat sederhana, misalnya hanya berupa <i>while</i> untuk menerima input. Fungsi-fungsi yang kalian definisikan sebaiknya dibuat di file terpisah.  Segera selesaikan bagian ini dan integrasikan dengan keseluruhan program. Minimal, usahakan semuanya selesai besok. Lakukan juga uji coba dengan <i>enlight testing</i> , mulai dari register dan test, untuk memastikan tidak ada bug.  Untuk array, sebaiknya pisahkan antara array dinamis dan array biasa. Jika memungkinkan, gunakan mesin kata saja untuk bagian ini, sesuai nilai dari asisten kalian. Akan lebih baik jika hanya
<b>Tempat : Google Meetings</b>	
<b>Kehadiran Anggota Kelompok:</b>  1 18223129   2 18223123   3 18223116	



4  
18223133



5  
18223107



6  
18223122



mengedit mesin kata yang sudah ada, daripada membuat ADT baru.


Implementasi mesin kalimat boleh dibuat dari mesin kata. Namun, tetap gunakan mesin kata karena kalian perlu berinteraksi dengan file. Jika membutuhkan *path*, cukup buat ADT baru yang menggunakan mesin kata dan implementasi mesin kalimat dari mesin kata untuk menghindari redundansi.

Jadi, kalian tetap bisa memakai mesin kalimat, tetapi tidak perlu memasukkannya ke dalam mesin kata.

Untuk bagian world3 dan tebak angka, kalian bebas berkreasi sesuai ide masing-masing. Kembangkan fitur atau pendekatan yang kalian rasa menarik dan sesuai konsep program secara keseluruhan.

Selain itu, coba pelajari penggunaan Makefile atau OSIA Script untuk mempermudah proses pengelolaan dan pengaturan program. Dengan begitu, integrasi dan kompilasi akan menjadi lebih efisien. Tapi, semuanya kembali lagi ke kebebasan dan kreativitas masing-masing. Sesuaikan dengan cara yang paling nyaman bagi kalian.

**Tanda Tangan Asisten:**

	
--	--

### 9.3 Log Activity Anggota Kelompok

No.	Log Activities	NIM	Tanggal
1.	Kerja Kelompok (Meet Perdana) - Diskusi Konsep dan Pembagian Tugas	18223129 18223123 18223116 18223133 18223107 18223122	13 - 11 - 2024
2.	Finalisasi ADT dan fungsi main	18223129	17 - 11 - 2024
3.	Kerja Kelompok - Diskusi Penggunaan ADT	18223129 18223123 18223116 18223133 18223107 18223122	18 - 11 - 2024
4.	Push Load dan Start	18223129 18223122	19 - 11 - 2024
5.	Push Login	18223129	19 - 11 - 2024
6.	Push Work Challenge, Tebak Angka dan WORLD3	18223107	20 - 11 - 2024
7.	Push Store List dan Store Request	18223116	20 - 11 - 2024
8.	Push Work	18223122	21 - 11 - 2024
9.	Push Store Remove dan Logout	18223116	22 - 11 - 2024
10.	Push Store Supply	18223133	22 - 11 - 2024
11.	Push Register, Help, dan Save	18223123	23 - 11 - 2024

12.	Push Store Request Bio Weapon	18223129	23 - 11 - 2024
13.	Push Quantum WORLD3	18223129 18223107	25 - 11 - 2024
14.	Finalisasi Main	18223129 18223107	25 - 11 - 2024
15.	Finalisasi Laporan	18223116 18223133	25 - 11 - 2024

***Tabel 3 Log Activity***