

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data STI

PURRMART


Dipersiapkan oleh:

Kelompok 3

Muhammad Aidan Fathullah I	18223002
Vandega Arozan Musholine	18223010
Irdina Ilmuna Yosapat	18223060
Vincentia Belinda Sumartoyo	18223078
Gabriela Jennifer Sandy	18223092

Sekolah Teknik Elektro dan Informatika 2024 - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB-02-03		<jml hlm>
		Revisi	0	11 November 2024

Daftar Isi

Update daftar isi di bawah ini.

1	Ringkasan	3
2	Penjelasan Tambahan Spesifikasi Tugas	4
2.1	<Spesifikasi Fitur Tambahan 1>	4
2.2	<Spesifikasi Fitur Tambahan 2>	5
3	Struktur Data (ADT)	5
3.1	Struktur Data Mesin Karakter	5
3.2	Struktur Data Mesin Kata	5
3.3	Struktur Data Barang	6
3.4	Struktur Data User	6
3.5	Struktur Data List Statik	7
3.6	Struktur Data List Dinamik	7
3.7	Struktur Data Queue	8
4	Program Utama	8
5	Algoritma-Algoritma Menarik	10
5.1	<Algoritma 1>	10
5.2	<Algoritma 2>	10
6	Data Test	10
6.1	Data Test START	10
6.2	Data Test LOAD	10
6.3	Data Test LOGIN	10
6.4	Data Test REGISTER	11
6.5	Data Test LOG OUT	11
6.6	Data Test WORK	11
6.7	Data Test WORK CHALLENGE	11
6.8	Data Test STORE LIST	11
6.9	Data Test STORE REQUEST	11
6.10	Data Test STORE SUPPLY	11
6.11	Data Test STORE REMOVE	11
6.12	Data Test HELP	12
6.13	Data Test SAVE	12
6.14	Data Test QUIT	12
7	Test Script	12
8	Pembagian Kerja dalam Kelompok	15
9	Lampiran	16
9.1	Deskripsi Tugas Besar	16

System Mechanic	16
1. About the System	16
2. Menu Program	16
3. Command	17
a. START	17
b. LOAD <filename>	17
c. LOGIN	17
d. LOGOUT	17
e. REGISTER	17
f. WORK	17
g. WORK CHALLENGE	17
a) Tebak Angka	18
b) WORDL3	18
h. STORE LIST	18
i. STORE REQUEST	18
j. STORE SUPPLY	18
k. STORE REMOVE	19
l. HELP	19
m. SAVE <filename>	19
n. QUIT	19
9.2 Notulen Rapat	19
9.3 Log Activity Anggota Kelompok	21
9.4 <Lain-Lain>	23

1 Ringkasan

PURRMART adalah sebuah sistem jual beli yang diperlukan oleh Agen Purry. harus dibuat oleh tim *programmer* untuk membantu Agen Purry dari OWCA. Ia mendapat misi untuk menghentikan proses pembuatan mesin penghancur yang dilakukan oleh Dr. Asep Spakbor. Dalam menghentikannya, Agen Purry terlibat dalam peperangan yang membutuhkan banyak alat perang. Namun, akses terhadap pasokan alat-alat tersebut yang berada di Toko Borma tidaklah mudah. Oleh karena itu, Agen Purry membutuhkan sistem jual beli untuk mempermudah akses tersebut. Untuk itu, Ia menghubungi tim *programmer* untuk merancang PURRMART.

PURRMART adalah sebuah toko yang mengimplementasikan CLI atau *command-line interface*. Toko ini memungkinkan pelanggan untuk melakukan beberapa aktivitas, yaitu memperoleh uang dan mengolah barang yang ada dalam toko. Untuk memperoleh uang, pelanggan perlu memilih pekerjaan terlebih dahulu. Lalu, pelanggan dapat melakukan beberapa *challenge* dengan aturan bermain yang berbeda. Pada pengolahan barang dalam toko, pelanggan dapat melihat daftar barang yang tersedia di toko beserta detailnya, menambahkan barang baru ke dalam toko, atau meminta barang tertentu untuk disuplai. Selain itu, pengguna dapat menyimpan barang ke dalam keranjang sebelum melakukan pembelian serta melihat daftar barang yang telah dibeli lengkap dengan riwayat transaksinya. Toko ini juga memungkinkan pengguna untuk membuat daftar wishlist barang yang diinginkan dan menghapus item dari daftar tersebut.

Laporan ini mendeskripsikan mengenai program-program yang kami kerjakan. Pada laporan ini, terdapat lima bagian penjelasan. Bagian pertama berisi rangkuman yang berisi penjelasan secara garis besar tentang latar belakang pembuatan program dan penjabaran laporan secara ringkas. Bagian kedua adalah penyertaan spesifikasi tugas besar tambahan. Bagian ketiga mengandung struktur data (ADT) yang diimplementasikan dalam program yang kami kerjakan. Bagian keempat berisi penjelasan *main* program. Bagian kelima merupakan penjelasan mengenai algoritma-algoritma menarik yang terdapat dalam program yang kami kerjakan.

Program PURRMART dibuat menggunakan beberapa ADT yang terdapat dalam pembelajaran mata kuliah Algoritma dan Struktur Data, dengan dasar bahasa pemrograman C, seperti ADT User, List Dinamik, List Statik, Mesin Karakter, Mesin Kata, dan Queue.

2 Penjelasan Tambahan Spesifikasi Tugas

Program PURRMART memiliki alur kerja dan interaksi pengguna. Pertama, keberjalanan program dimulai dari opsi START, LOAD, dan HELP. Dengan memilih START, program akan membaca file konfigurasi default yang berisi username? money? daftar yang tersedia. Kemudian, LOAD memungkinkan pengguna untuk membuat save file sebelumnya dari folder tertentu untuk melanjutkan sesi sebelumnya.

2.1 BIO WEAPON

Fitur tambahan yang memungkinkan pemain untuk menambahkan barang baru berupa senjata biologis ke PURRMART. Senjata biologis ini menggunakan patogen yang telah

direkayasa secara genetis untuk menyerang target, namun memiliki risiko tinggi sabotase. Dalam implementasi, pemain harus memasukkan nama senjata, sekuens DNA, dan kode rahasia.

3 Struktur Data (ADT)

Dalam program PURRMART, kami menggunakan beberapa ADT, yaitu ADT User, ADT List Dinamis dan Statis, ADT Barang, ADT Mesin Karakter, ADT Mesin Kata, dan ADT Queue.

3.1 Struktur Data Mesin Karakter

Struktur data pertama yang digunakan untuk menyelesaikan persoalan adalah Mesin Karakter yang diimplementasikan dalam bentuk ADT mesin_karakter. Sketsa struktur dasar dari ADT ini meliputi beberapa elemen utama, yaitu **static FILE *pita** yang berfungsi untuk menyimpan pointer ke file atau input yang sedang dibaca, **char current_char** untuk menyimpan karakter yang sedang diakses dari pita, dan **boolean eop** yang menandakan apakah telah mencapai akhir pita (end of process/EOP). Fungsi-fungsi utama yang mendukung operasi ADT ini meliputi **start(char *filename)** untuk memulai pembacaan pita dari file tertentu atau dari stdin, **adv()** untuk membaca karakter berikutnya dari pita, **get_current_char()** untuk mengembalikan karakter saat ini dari pita, dan **is_eop()** untuk mengecek apakah telah mencapai akhir pita.

Mesin karakter ini dirancang untuk menyelesaikan persoalan dengan membaca file konfigurasi karakter demi karakter, serta melakukan parsing data seperti nama barang, harga, nama pengguna, dan saldo secara fleksibel. Selain itu, mesin karakter mempermudah proses pemisahan berdasarkan spasi, newline, atau karakter khusus lainnya, sehingga aplikasi menjadi lebih fleksibel dalam memahami dan memproses berbagai macam perintah. Alasan utama pemilihan ADT ini, selain karena sifatnya yang wajib, adalah karena ADT ini berperan sebagai dasar dalam membaca tipe data paling dasar yang dikenali oleh bahasa C. Fungsi-fungsi pada mesin karakter ini juga terpisah dari logika utama, sehingga membuat kode menjadi lebih bersih, modular, dan membantu dalam menciptakan program yang lebih terstruktur.

Mesin karakter diimplementasikan sebagai ADT dengan nama header **mesin_karakter.h** dan file implementasinya diberi nama **mesin_karakter.c**.

3.2 Struktur Data Mesin Kata

Struktur data kedua yang digunakan untuk menyelesaikan persoalan PURRMART adalah Mesin Kata yang diimplementasikan dalam bentuk ADT mesin_kata. Mesin Kata adalah ADT yang digunakan untuk membaca dan memproses input berupa kata-kata dari file konfigurasi maupun perintah pengguna di CLI. Struktur data ini terdiri dari variabel global seperti **current_word** (menyimpan kata yang sedang diproses), **end_word** (penanda akhir input), serta fungsi-fungsi utama seperti **start_word**, **adv_word**, **tab_word** dan **copy_word**. Ada beberapa variabel yang sama dengan mesin_karakter, seperti **adv**, **start**, **eop**.

Mesin Kata mengatasi persoalan *parsing* data dengan memisahkan kata-kata berdasarkan pembatas seperti spasi atau *newline*, serta memproses perintah CLI seperti LOAD atau WORK CHALLENGE. Mesin Kata dibuat untuk memastikan input diproses secara seragam dan terstruktur, serta menghindari kesalahan akibat format data yang tidak sesuai.

Alasan penggunaan Mesin Kata adalah efisiensi yang cukup tinggi dimana dengan fungsi pembantu seperti **is_same_word** untuk membandingkan kata dan **string_to_word** untuk konversi *string*, sehingga cocok untuk mengelola input dalam aplikasi PURRMART. Mesin Kata diimplementasikan sebagai ADT dimana headernya diberi nama `mesin_kata.h` dan file implementasinya dengan nama `mesin_kata.c`.

3.3 Struktur Data Barang

Struktur data ketiga yang digunakan untuk menyelesaikan persoalan PURRMART adalah Barang. ADT Barang digunakan untuk mengelola data barang yang tersedia dalam aplikasi PURRMART, yang mencakup informasi tentang nama dan harga barang. Struktur data ini menyelesaikan beberapa persoalan utama, seperti pengaturan stok barang, pengecekan apakah barang sudah ada atau tidak dalam sistem, serta menambah barang baru ke dalam daftar. Fungsi-fungsi dalam ADT ini, seperti **add_barang** berfungsi untuk menambahkan barang baru, **barang_count** digunakan untuk menghitung jumlah barang yang ada, dan **is_barang_exist** untuk memeriksa apakah barang dengan nama tertentu sudah ada.

Alasan memilih struktur data ini adalah untuk menyimpan informasi barang secara terstruktur dan memudahkan pencarian serta penambahan barang dalam sistem. Dengan menggunakan array `barangs[MAX_BARANG]`, sistem dapat dengan cepat mengakses dan memanipulasi data barang. ADT Barang diimplementasikan dalam file `barang.c` untuk mendukung operasi seperti menambah barang dan mengecek keberadaan barang, yang sangat penting dalam aplikasi PURRMART untuk mengelola stok barang secara dinamis.

3.4 Struktur Data User

Pada pengerjaan program PURRMART, kami menggunakan struktur data Pengguna (struct User) untuk merepresentasikan data pengguna dalam sistem. Struktur data ini menyimpan informasi penting seperti *name*, *password*, dan *money* yang dimiliki pengguna, semuanya diimplementasikan menggunakan array statis. Data ini digunakan untuk mencatat semua pengguna yang terdaftar dalam sistem PURRMART, mendukung fitur seperti **REGISTER**, **LOGIN**, **add_money**, dan **subtract_money**.

Struktur data ini menyelesaikan beberapa persoalan utama, seperti manajemen pengguna dengan memastikan pengguna yang terdaftar tidak duplikat, serta memvalidasi kredensial login (*username* dan *password*) dengan tepat sehingga menjaga keamanan akses ke dalam sistem. Struktur ini juga memastikan pengelolaan data pengguna berjalan dengan baik, termasuk memperbarui saldo pengguna secara tepat sesuai transaksi yang dilakukan.

Alasan pemilihan struktur data ini adalah untuk menyediakan mekanisme yang efisien dalam menyimpan dan mengakses data pengguna, serta memungkinkan pengecekan dan pembaruan data dengan cepat.

Agar implementasinya terstruktur dan mudah dikelola, fungsi-fungsi yang terkait dengan *struct User* ditempatkan dalam modul terpisah. Modul ini mencakup beberapa fungsi utama, seperti:

1. **add_user** – menambahkan pengguna baru dengan memastikan tidak ada nama pengguna yang sama.
2. **is_user_exist** – memeriksa apakah pengguna dengan nama tertentu sudah terdaftar di sistem.
3. **is_user_valid** – memverifikasi nama pengguna dan kata sandi saat proses login.
4. **update_money** – memperbarui saldo pengguna untuk transaksi, baik berupa penambahan maupun pengurangan, dengan memastikan saldo tidak negatif.

3.5 Struktur Data List Statik

Dalam aplikasi PURRMART, kami juga menggunakan List, dimana list yang digunakan dibagi menjadi list statik dan dinamik. Dalam driver list statik, **StaticIntList** digunakan untuk menyimpan daftar bilangan bulat, seperti ID barang atau data numerik lainnya, sedangkan **StaticCharList** digunakan untuk daftar karakter, seperti huruf atau simbol. ADT List Statik ini menyelesaikan beberapa persoalan utama, seperti menyimpan data secara terurut, mengecek keberadaan elemen, menambahkan elemen baru, dan menghapus elemen tertentu. Fungsi seperti **insert_static_int_list** dan **delete_static_int_list** mendukung operasi penambahan dan penghapusan elemen dengan mudah, serta fungsi **is_static_int_list_exist** memungkinkan pengecekan cepat apakah elemen tertentu sudah ada di dalam list.

Alasan pemilihan struktur data list statis adalah karena sifatnya list yang sederhana dan efisien sehingga cocok digunakan untuk data dengan ukuran tetap yang tidak berubah-ubah selama *runtime*. Elemen juga dapat diakses dengan indeks langsung melalui pendekatan array, serta memungkinkan operasi seperti pencarian dan penghapusan berjalan dengan kompleksitas yang dapat diprediksi. ADT List Statis ini diimplementasikan dalam file *list_statik.c* untuk mendukung keberlangsungan dari aplikasi PURRMART.

3.6 Struktur Data List Dinamik

Sedangkan dalam list dinamik, **DynamicIntList** digunakan untuk menyimpan daftar bilangan bulat, seperti ID barang maupun data numerik lainnya, sedangkan **DynamicCharList** digunakan untuk menyimpan daftar karakter, seperti simbol atau nama singkat. ADT ini menyelesaikan persoalan utama terkait pengelolaan data yang tidak memiliki ukuran tetap dan dapat berkembang seiring bertambahnya elemen, tanpa dibatasi kapasitas awal. Fungsi seperti **resize_dynamic_int_list** dan **resize_dynamic_char_list** memungkinkan kapasitas list bertambah secara dinamis, sedangkan fungsi seperti **insert_dynamic_int_list** dan **delete_dynamic_int_list** mendukung penambahan dan penghapusan elemen secara efisien.

Alasan pemilihan struktur data list dinamik ini adalah fleksibilitasnya dalam menangani data yang jumlahnya sulit diprediksi, karena kapasitasnya dapat ditingkatkan sesuai kebutuhan. Dengan menggunakan pointer dan alokasi memori dinamis, ADT ini memungkinkan pengelolaan memori yang lebih efisien dibandingkan list statis. List Dinamis diimplementasikan dalam file `list_dinamik.c` untuk mendukung operasi seperti menambah, menghapus, dan mencari elemen dalam daftar bilangan bulat atau karakter secara fleksibel, serta mendukung keberlangsungan dan kelancaran program aplikasi PURRMART.

3.7 Struktur Data Queue

Struktur data keenam yang diterapkan dalam pengembangan **PURRMART** adalah **ADT Queue** dan **CircularQueue**, yang berperan penting dalam pengelolaan antrian barang atau proses tertentu dalam aplikasi. Salah satu penggunaannya adalah untuk menangani permintaan penambahan barang ke toko melalui fitur **STORE REQUEST**. ADT Queue dirancang untuk mengelola antrian secara linier, sedangkan CircularQueue memungkinkan pengelolaan antrian berbentuk melingkar, sehingga ruang penyimpanan dapat dimanfaatkan secara optimal dengan mendaur ulang slot kosong setelah elemen dikeluarkan.

Kedua struktur data ini dirancang untuk menyelesaikan persoalan utama terkait manajemen antrian, terutama memastikan elemen dapat dimasukkan dan dikeluarkan secara teratur sesuai prinsip FIFO (*First In, First Out*). Operasi seperti **enqueue** digunakan untuk menambahkan elemen ke antrian, sedangkan **dequeue** berfungsi untuk mengeluarkan elemen dari antrian. Fungsi tambahan seperti **front** dan **rear** memberikan akses langsung ke elemen pertama dan terakhir dalam antrian, sehingga mempermudah pengelolaan data. Dengan pendekatan ini, PURRMART dapat menangani berbagai permintaan atau tugas yang datang secara berurutan tanpa kehilangan integritas data.

Alasan utama pemilihan struktur data Queue dan CircularQueue adalah untuk mendukung kebutuhan aplikasi yang memprioritaskan pengelolaan antrian secara efisien dan terstruktur. Mekanisme FIFO yang diimplementasikan memastikan bahwa setiap permintaan diproses dalam urutan yang sesuai, yang sangat penting untuk fitur seperti pengelolaan barang di toko. CircularQueue, secara khusus, memberikan efisiensi tambahan dengan memanfaatkan kembali ruang penyimpanan yang tersedia, sehingga cocok untuk aplikasi yang menangani volume data dinamis seperti PURRMART.

Implementasi struktur data ini ditempatkan dalam modul terpisah, yaitu file **queue.c**, yang mencakup berbagai fungsi inti untuk manajemen antrian. Modul ini menyediakan mekanisme antrian yang fleksibel, efisien, dan dapat diandalkan, mendukung kelancaran operasional aplikasi PURRMART dalam mengelola berbagai tugas dan permintaan pengguna.

4 Program Utama

Program utama PURRMART dimulai dengan inisiasi sistem, dimana pengguna dapat menjalankan program dengan perintah **START** atau memuat data dari file konfigurasi dengan menuliskan perintah **LOAD <filename>**. Proses load akan membaca data seperti daftar barang di toko, daftar pengguna terdaftar, dan nama pengguna, password, serta jumlah uang. Jika file

konfigurasi tidak ditemukan atau rusak, program akan menampilkan pesan kesalahan dan kembali ke menu utama.

Setelah data awal berhasil dimuat, program masuk ke menu **login**, di mana pengguna dapat memilih untuk melakukan login menggunakan akun yang sudah ada (**LOGIN**) atau mendaftarkan akun baru (**REGISTER**). Proses **REGISTER** memverifikasi bahwa nama pengguna belum digunakan sebelumnya, lalu menambahkan data pengguna baru ke sistem dengan atribut nama, password, dan saldo awal akan diinisiasi = 0. Sedangkan, perintah **LOGIN** memeriksa validitas nama pengguna dan password yang dimasukkan. Jika validasi berhasil, pengguna akan masuk ke program **PURRMART**, dan jika gagal, pengguna diminta untuk menuliskan inputan lagi.

Setelah login, pengguna akan masuk ke program utama aplikasi serta dapat mengakses beberapa fitur. Fitur yang dapat diakses seperti:

1. Melihat daftar barang → **STORE LIST** → program akan menampilkan semua barang yang tersedia di toko beserta harga barang tersebut. Jika toko kosong, akan ditampilkan pesan bahwa toko kosong.
2. Meminta barang baru → **STORE REQUEST** → pengguna dapat menambahkan permintaan barang baru ke dalam antrian. Barang yang diminta harus memiliki nama unik. Apabila barang sudah ada di toko atau sudah berada dalam antrian, akan ditampilkan pesan juga.
3. Menambahkan barang ke toko → **STORE SUPPLY** → pengguna dapat memproses barang dari antrian permintaan untuk ditambahkan ke toko. Harga barang dimasukkan oleh pengguna saat proses menambahkan barang ke toko.
4. Menghapus barang dari toko → **STORE REMOVE** → barang yang tidak diperlukan dapat dihapus dari toko oleh pengguna dengan memastikan nama barang valid dan ada di dalam daftar toko.

Selain fitur manajemen barang, **PURRMART** juga memiliki fitur untuk mengelola saldo pengguna. Pengguna dapat bekerja menggunakan perintah **WORK** dan mendapatkan uang. **WORK** akan menampilkan daftar pekerjaan dengan penghasilan dan durasi kerja tertentu. Pengguna dapat memilih pekerjaan, dan setelah durasi selesai, saldo pengguna akan bertambah sesuai penghasilan yang dihasilkan. Alternatif lain, pengguna dapat mengikuti tantangan interaktif menggunakan perintah **WORK CHALLENGE**. Tantangan ini melibatkan mini-games seperti **Tebak Angka** atau **WORDL3**, di mana pengguna dapat bertaruh uang untuk mendapatkan hadiah lebih besar jika menang.

Jika pengguna bingung dengan perintah yang tersedia, pengguna dapat menggunakan perintah **HELP**, yang menampilkan daftar rangkuman semua perintah yang dapat dieksekusi di menu saat ini beserta deskripsinya. Fungsi ini sangat penting untuk membantu pengguna memahami fitur-fitur yang ada, terutama bagi yang baru menggunakan aplikasi.

Program **PURRMART** akan terus berjalan hingga pengguna memilih perintah **QUIT**. Sebelum keluar, pengguna dapat memilih untuk menyimpan status aplikasi saat ini ke dalam file konfigurasi menggunakan perintah **SAVE <filename>**. Proses **SAVE** akan menyimpan daftar

barang, data pengguna, dan status aplikasi lainnya sehingga sesi berikutnya dapat dilanjutkan dari keadaan terakhir. Fitur-fitur ini memberikan pengalaman pengguna yang lebih interaktif dan terorganisir dalam menjalankan program PURRMART yang lengkap, modular, dan efisien.

5 Algoritma-Algoritma Menarik

Pada tugas besar ini, kami menemukan algoritma yang menarik untuk digunakan serta algoritma yang kami anggap efisien sehingga dapat mengurangi efisiensi program.

5.1 Store request

Prosedur store request terutama fungsi **store_request** adalah algoritma yang menarik karena mencerminkan proses nyata dalam manajemen *e-commerce*, menggabungkan konsep struktur data antrian (Queue), dan interaksi langsung dengan pengguna. Algoritma ini unik karena memanfaatkan logika validasi yang sistematis dan mengelola permintaan barang secara FIFO (*First In, First Out*), serta memastikan barang yang diminta lebih dulu diproses lebih dahulu. Selain itu, prosedur store_request sangat mengimplementasikan materi Queue yang sudah dipelajari di kelas sehingga kami bersemangat untuk mempelajari hal ini lebih dalam.

```
void store_request(char *name) {
    // apakah barang sudah ada di toko atau belum
    if (is_barang_exist(barangs, name)) {
        printf("Barang dengan nama yang sama sudah ada di toko!\n");
        return;
    }

    // apakah barang sudah ada di dalam antrian
    int queueindex = queue.head;
    while (queueindex != queue.tail) {
        if (is_same_string(queue.buffer[queueindex], name)) {
            printf("Barang dengan nama yang sama sudah ada di antrian!\n");
            return;
        }
        queueindex = (queueindex + 1) % MAX_QUEUE;
    }

    // tambahkan barang ke antrian
    if (!is_queue_full(&queue)) {
        enqueue(&queue, name);
        printf("Barang berhasil diminta.\n");
    } else {
        printf("Antrian penuh.\n");
    }
}
```

Gambar 5.2.1. store_request sebagai algoritma yang menarik

6 Data Test

Program PURRMART ini terdiri dari beberapa fitur atau command yang dapat diberikan oleh pengguna. Oleh karena itu, diperlukan testing pada kondisi-kondisi tertentu untuk memastikan apakah tiap fitur tersebut dapat berjalan dengan tepat dan benar. Berikut adalah testing yang kami lakukan terhadap command yang terdapat pada program PURRMART ini,

beserta dengan penjelasan mengenai hasil atau output yang akan diberikan oleh program pada tiap kondisinya.

6.1 Data Test START

Tes ini dilakukan untuk mengecek keberjalanan program utama dan kondisi awal program.

```
>>> START
Konfigurasi aplikasi berhasil dibaca. PURRMART berhasil dijalankan.

*****
*           Silahkan Bergabung di PURRMART           *
*****
* 1. LOGIN                                           *
* 2. REGISTER                                       *
* 3. LOGOUT                                          *
*****
* Pilih Command atau ketik 'HELP'! *
*****
>>> |
```

Gambar 6.1.1 Tampilan ketika berhasil memulai start

```
WELCOME TO PURRMART

*****
* Selamat Datang di PURRMART *
*****
* 1. START                   *
* 2. LOAD                   *
* 3. QUIT                   *
*****
* Pilih Command atau ketik 'HELP'! *
*****
>>> |
```

Gambar 6.1.2 Kondisi awal program

6.2 Data Test LOAD

Tes ini dilakukan untuk mengecek keberjalanan program utama dan kondisi awal program.

```
*****
*      Selamat Datang di PURRMART      *
*****
*      1. START                        *
*      2. LOAD                        *
*      3. QUIT                        *
*****
*      Pilih Command atau ketik 'HELP'!  *
*****
>>> LOAD
Save file tidak ditemukan. PURRMART gagal dijalankan.
```

Gambar 6.2.1 Tampilan ketika filename tidak ditemukan

6.3 Data Test LOGIN

Tes ini dilakukan untuk masuk ke dalam program dengan akun pengguna yang sudah terdaftar

```
>>> LOGIN
Username: admin
Password: 12345
Username atau Password salah. Coba lagi.
Username: █
```

Gambar 6.3.1 Tampilan ketika input username atau password salah

```
>>> LOGIN
Username: admin
Password: 12345
Username atau Password salah. Coba lagi.
Username: admin
Password: alstrukdatkeren
Login berhasil. Selamat datang, admin!
>>> █
```

Gambar 6.3.2 Tampilan ketika login berhasil dimana input username atau password benar

6.4 Data Test REGISTER

Tes ini dilakukan untuk mendaftarkan akun yang belum tercatat dalam *file* konfigurasi.

```
>>> register
Username: aidan
Password: baiksolehammin
Akun dengan username "aidan" berhasil dibuat. Silahkan LOGIN untuk melanjutkan!
>>> login
Anda sudah login sebagai aidan. Jika ingin login sebagai user lain, silakan logout terlebih dahulu.
>>> logout
aidan telah logout dari sistem PURRMART. Silakan REGISTER/LOGIN kembali untuk melanjutkan.
>>> register
Username: aidan
Password: baiksolehbanget
Akun dengan username "aidan" sudah digunakan. Silakan coba username lain!
```

Gambar 6.4.1 Tampilan ketika REGISTER berhasil dijalankan

6.5 Data Test LOG OUT

Tes ini dilakukan untuk mengeluarkan akun dari *file* konfigurasi

```
>>> LOGOUT
admin telah logout dari sistem PURRMART. Silakan REGISTER/LOGIN kembali untuk melanjutkan.
```

Gambar 6.5.1 Tampilan ketika LOGOUT berhasil dijalankan

6.6 Data Test WORK

Tes ini dilakukan untuk memastikan pengguna dapat memilih pekerjaan dan mendapatkan uang dari pekerjaan yang dilakukan.

```
>>> 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: Mewing Specialist
Anda sedang bekerja sebagai Mewing Specialist... harap tunggu.
Pekerjaan selesai, +10000 rupiah telah ditambahkan ke akun Anda.
```

Gambar 6.6.1 Tampilan ketika pekerjaan yang dipilih merupakan Mewing Specialist

```

>>> 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: Inator Connoisseur
Anda sedang bekerja sebagai Inator Connoisseur... harap tunggu.
Pekerjaan selesai, +997 rupiah telah ditambahkan ke akun Anda.

```

Gambar 6.6.2 Tampilan ketika pekerjaan yang dipilih merupakan Inator Connoisseur

6.7 Data Test WORK CHALLENGE

Tes ini dilakukan untuk mengecek apakah pengguna bisa mendapatkan uang dengan melakukan challenge di OWCA, dimana terdapat dua challenge yang dapat dipilih : Tebak angka dan WORDL3.

```

>>> work challenge
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main = 200)
2. WORDL399 (biaya main = 500)
Masukkan nomor challenge yang dipilih: 1
Tebak angka (1-100) (sisa kesempatan:10) : 31
Angka terlalu kecil
Tebak angka (1-100) (sisa kesempatan:9) : 67
Angka terlalu besar
Tebak angka (1-100) (sisa kesempatan:8) : 45
Angka terlalu kecil
Tebak angka (1-100) (sisa kesempatan:7) : 57
Angka terlalu besar
Tebak angka (1-100) (sisa kesempatan:6) : 52
Angka terlalu besar
Tebak angka (1-100) (sisa kesempatan:5) : 49
Angka terlalu besar
Tebak angka (1-100) (sisa kesempatan:4) : 47
Angka terlalu kecil
Tebak angka (1-100) (sisa kesempatan:3) : 48
Selamat, kamu berhasil menebak angka! Anda mendapatkan uang sebanyak 150
>>>

```

Gambar 6.9.1 Tampilan ketika Tebak Angka berhasil dijalankan

```
Welcome to WORDL399, you have 6 chance to guess the word
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
Masukan kata tebakan Anda: aidan
a* i% d% a* n%
- - - - -
- - - - -
- - - - -
- - - - -
Masukan kata tebakan Anda: keren
a* i% d% a* n%
k% e% r% e% n%
- - - - -
- - - - -
- - - - -
Masukan kata tebakan Anda: soleh
a* i% d% a* n%
k% e% r% e% n%
s* o l% e% h%
- - - - -
- - - - -
- - - - -
Masukan kata tebakan Anda: 
```

Gambar 6.9.2 Tampilan ketika WORDL3 berhasil dijalankan

6.8 Data Test STORE LIST

Tes ini dilakukan untuk mengecek *list* barang yang tersedia di toko dimana barang bersifat *unique*.

```
>>> STORE LIST
Daftar barang di toko:
- AK47
- Lalabu
- Ayam Goreng Crisbar
```

Gambar 6.8.1 Tampilan ketika STORE LIST berhasil dijalankan

6.9 Data Test STORE REQUEST

Tes ini dilakukan untuk mengecek keberjalanan fungsi *STORE REQUEST* dimana menerima permintaan penambahan barang baru ke dalam toko.

```
>>> STORE REQUEST
Nama barang yang diminta: Lemon Tea
Barang berhasil diminta.
```

Gambar 6.8.2 Tampilan ketika STORE REQUEST berhasil dijalankan

6.10 Data Test STORE SUPPLY

Tes ini dilakukan untuk mengecek keberjalanan fungsi *STORE SUPPLY* dimana sistem menambahkan barang baru ke toko berdasarkan antrian permintaan.

```
>>> STORE SUPPLY
Apakah kamu ingin menambahkan barang Lemon tea? (Terima/Tunda/Tolak/Purry) : Terima
Harga barang: 100
Lemon tea dengan harga 100 telah ditambahkan ke toko.
```

Gambar 6.10.1 Tampilan ketika STORE SUPPLY berhasil dijalankan dimana pengguna menerima penambahan barang

```
>>> STORE REQUEST
Nama barang yang diminta: Kue
Barang berhasil diminta.
>>> STORE SUPPLY
Apakah kamu ingin menambahkan barang Kue? (Terima/Tunda/Tolak/Purry) : Tolak
Kue dihapuskan dari antrian.
```

Gambar 6.10.2 Tampilan ketika STORE SUPPLY berhasil dijalankan dimana pengguna menolak penambahan barang.

```
>>> STORE SUPPLY
Tidak ada barang pada antrian.
```

Gambar 6.10.3 Tampilan ketika STORE SUPPLY berhasil dijalankan dimana tidak ada barang pada antrian


```
>>> STORE SUPPLY
Apakah kamu ingin menambahkan barang Dum Dum? (Terima/Tunda/Tolak/Purry) : Tunda
Dum Dum dikembalikan ke antrian.
```

Gambar 6.10.4 Tampilan ketika STORE SUPPLY berhasil dijalankan dimana pengguna menunda penambahan barang.

6.11 Data Test STORE REMOVE

Tes ini dilakukan untuk mengecek keberjalanan fungsi *STORE REMOVE* dimana menghapus barang yang ada di toko.

```
>>> STORE REMOVE
Nama barang yang akan dihapus: d
Toko tidak menjual d.
>>> STORE LIST
Daftar barang di toko:
- AK47
- Lalabu
- Ayam Goreng Crisbar
- Lemon tea
```

Gambar 6.11.1 Tampilan ketika STORE REMOVE berhasil dijalankan dimana pengguna memasukkan input yang tidak ada di toko

```
>>> STORE REMOVE
Nama barang yang akan dihapus: Lemon tea
Lemon tea berhasil dihapus dari toko.
>>> STORE LIST
Daftar barang di toko:
- AK47
- Lalabu
- Ayam Goreng Crisbar
```

Gambar 6.11.2 Tampilan ketika STORE REMOVE berhasil dijalankan dimana berhasil menghapus barang dari toko

6.12 Data Test HELP

Tes ini dilakukan untuk mengecek keberjalanan fungsi *HELP* sebelum dan setelah *START* atau *LOAD*.

```

>>> HELP
===== [Selamat datang di PURRMART] =====
| Berikut adalah Command yang dapat Anda lakukan:
| 1. START
|   Untuk masuk sesi baru
| 2. LOAD
|   Untuk memulai sesi berdasarkan suatu file konfigurasi
| 3. QUIT
|   Untuk keluar dari program
|
=====
>>>

```

Gambar 6.12.1 Tampilan ketika HELP berhasil dan perintah dipanggil pada welcome menu

```

>>> HELP
===== [Login Menu Help PURRMART] =====
| Berikut adalah Command yang dapat Anda lakukan:
| 1. REGISTER
|   Untuk melakukan pendaftaran akun baru
| 2. LOGIN
|   Untuk masuk ke dalam akun dan memulai sesi
| 3. QUIT
|   Untuk keluar dari program
|
=====
>>>

```

Gambar 6.12.2 Tampilan ketika HELP berhasil dan perintah dipanggil pada login menu

```

Login berhasil. Selamat datang, admin!
>>> HELP
===== [Main Menu Help PURRMART] =====
| Berikut adalah Command yang dapat Anda lakukan:
| 1. WORK
|   Untuk bekerja
| 2. WORK CHALLENGE
|   Untuk mengerjakan challenge
| 3. STORE LIST
|   Untuk melihat barang-barang di toko
| 4. STORE REQUEST
|   Untuk meminta penambahan barang
| 5. STORE SUPPLY
|   Untuk menambahkan barang dari permintaan
| 6. STORE REMOVE
|   Untuk menghapus barang
| 7. LOGOUT
|   Untuk keluar dari sesi
| 8. SAVE
|   Untuk menyimpan state ke dalam file
| 9. QUIT
|   Untuk keluar dari program
|
=====
>>>

```

Gambar 6.12.3 Tampilan ketika HELP berhasil dan perintah dipanggil pada main menu

6.13 Data Test SAVE

Tes ini dilakukan untuk mengecek keberjalanan fungsi *SAVE* dimana menyimpan *state* aplikasi terbaru ke dalam suatu *file*.

```
>>> store list
Daftar barang di toko:
- AK47
- Lalabu
- Ayam Goreng Crisbar
>>> store remove ak47
Unknown command: store remove ak47
>>> store remove
Nama barang yang akan dihapus: ak47
ak47 berhasil dihapus dari toko.
>>> store list
Daftar barang di toko:
- Lalabu
- Ayam Goreng Crisbar
>>> save aidanbaik.txt
Save file berhasil disimpan. save berhasil.
```

Gambar 6.13.1 Tampilan ketika SAVE berhasil dijalankan

6.14 Data Test QUIT

Tes ini dilakukan untuk mengecek keberjalanan fungsi *QUIT*

```
>>> quit
Kamu keluar dari PURRMART, quit
Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? n
Kamu keluar dari PURRMART
Dadah ^_^/
PS C:\Users\aidan\Kuliah\Institut Teknologi Bandung\Semester 3\IF2111 - Algoritma dan Struktur Data STI\Tugas-Besar-IF2111-Algoritma-dan-Struktur-Data-STI-2024-2025\src>
```

Gambar 6.13.1 Tampilan ketika QUIT berhasil dijalankan

7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	START	Memeriksa keberhasilan keberjalanan <i>file</i> konfigurasi	Menginput <i>command</i> START pada saat pertama kali memulai program	START	Gambar 6.1.1 Gambar 6.1.2	Sesuai dengan hasil yang diharapkan

2	LOAD	Menjalankan <i>file</i> konfigurasi	Menginput <i>command</i> LOAD yang memiliki satu argumen berupa <i>filename</i> .	LOAD < <i>filename</i> >	Gambar 6.2.1	Sesuai dengan hasil yang diharapkan
3	LOGIN	Masuk ke dalam program dengan akun pengguna yang sudah terdaftar	<ul style="list-style-type: none"> - Menginput <i>command</i> LOGIN. - Memasukkan <i>username</i> dan <i>password</i> dari pengguna. - Sistem melakukan verifikasi terhadap masukan pengguna. 	LOGIN	Gambar 6.3.1 Gambar 6.3.2	Sesuai dengan hasil yang diharapkan
4	REGISTER	Mendaftarkan akun yang belum tercatat dalam <i>file</i> konfigurasi	<ul style="list-style-type: none"> - Menginput <i>command</i> REGISTER setelah pengguna memulasi sesi. - Sistem mendaftarkan akun baru ke dalam sistem. - Sistem menerima pembuatan akun baru hanya jika <i>username</i> dan <i>password</i> terdiri dari 1 kata dan akun belum dibuat sebelumnya. 	REGISTER	Gambar 6.4.1	Sesuai dengan hasil yang diharapkan
5	LOGOUT	Mengeluarkan akun dari <i>file</i> konfigurasi	Menginput <i>command</i> LOGOUT ketika sudah melakukan LOGIN	LOGOUT	Gambar 6.5.1	Sesuai dengan hasil yang diharapkan
6	WORK	Memastikan pengguna dapat memilih pekerjaan dan mendapatkan uang dari pekerjaan yang dilakukan.	<ul style="list-style-type: none"> - Menginput <i>command</i> WORK. - Memasukkan nama pekerjaan yang tertera di daftar pekerjaan. - Sistem menolak perintah lain ketika pengguna sedang bekerja. - Sistem menambahkan pendapatan setelah pekerjaan selesai. 	WORK	Gambar 6.6.1 Gambar 6.6.2	Sesuai dengan hasil yang diharapkan
7	WORK CHALLENGE	Mendapatkan uang dengan melakukan challenge di OWCA, dimana terdapat dua challenge yang dapat dipilih :	<ul style="list-style-type: none"> - Menginput <i>command</i> WORK CHALLENGE. - Sistem menunjukkan daftar <i>challenge</i> yang tersedia beserta biaya main. - Sistem akan menerima <i>challenge</i> 	WORK CHALLENGE	Gambar 6.7.1 Gambar 6.7.2	Sesuai dengan hasil yang diharapkan

		Tebak angka dan WORDL3.	yang dipilih oleh pengguna. Jika uang yang dimiliki tidak mencukupi, maka <i>game</i> tidak bisa dimainkan. - Sistem akan memulai <i>challenge</i> dimana untuk permainan Tebak kata, Sistem hanya memberikan 10 kesempatan dan program akan memberikan <i>feedback</i> terhadap tebakkan kita, jumlah kesempatan yang dipakai oleh pengguna akan mempengaruhi uang yang didapatkan. - Sementara untuk WORDL3, Pemain memiliki 6 kesempatan untuk menebak kata yang benar, dimana kata harus valid.			
8	STORE LIST	Memperlihatkan <i>list</i> barang yang ada di toko (barang bersifat <i>unique</i>)	- Menginput <i>command</i> STORE LIST. - Sistem memperlihatkan list barang yang ada di toko.	STORE LIST	Gambar 6.8.1	Sesuai dengan hasil yang diharapkan
9	STORE REQUEST	Menerima permintaan penambahan barang baru ke dalam toko	- Menginput <i>command</i> STORE REQUEST. - Memasukkan nama barang yang ingin ditambahkan. - Sistem mengecek apakah barang sudah ada di toko atau antrian, bila berbeda maka barang bisa masuk ke toko.	STORE REQUEST	Gambar 6.9.1	Sesuai dengan hasil yang diharapkan
10	STORE SUPPLY	Menambahkan barang baru ke toko berdasarkan antrian permintaan.	- Menginput <i>command</i> STORE SUPPLY. - Sistem akan menerima input pengguna dimana jika diterima maka program akan meminta harga dari barang dan dimasukan ke toko.	STORE SUPPLY	Gambar 6.10.1 Gambar 6.10.2 Gambar 6.10.3 Gambar 6.10.4	Sesuai dengan hasil yang diharapkan

			Jika pengguna menunda penambahan barang, barang akan kembali masuk ke antrian dan jika pengguna menolak, maka barang akan dihapus dari antrian.			
11	STORE REMOVE	Menghapus barang yang ada di toko.	<ul style="list-style-type: none"> - Menginput <i>command</i> STORE REMOVE. - Sistem akan menerima input barang yang akan dihapus. - Sistem akan mengeluarkan output berhasil atau tidaknya terhapusnya barang di toko. 	STORE REMOVE	Gambar 6.11.1 Gambar 6.11.2	Sesuai dengan hasil yang diharapkan
12	HELP	Menampilkan daftar <i>command</i> yang mungkin untuk dijalankan beserta dengan deskripsinya	<ul style="list-style-type: none"> - Menginput <i>command</i> HELP. - Sistem akan menampilkan daftar <i>command</i> dan deskripsinya sesuai dengan dimana perintah tersebut dipanggil . 	HELP	Gambar 6.12.1 Gambar 6.12.2 Gambar 6.12.3	Sesuai dengan hasil yang diharapkan
13	SAVE	Menyimpan <i>state</i> aplikasi terbaru ke dalam suatu <i>file</i> .	<ul style="list-style-type: none"> - Menginput <i>command</i> SAVE. - Sistem akan menyimpan <i>state</i> ke dalam suatu <i>file</i> beserta argumen yang merepresentasikan nama <i>file</i> yang akan disimpan. 	SAVE <filename>	Gambar 6.13.1	Sesuai dengan hasil yang diharapkan
14	QUIT	Mengeluarkan pengguna dari sesi aplikasi.	<ul style="list-style-type: none"> - Menginput <i>command</i> QUIT. - Sistem akan menerima <i>input</i> berupa Y/N - Jika pengguna menginput Y maka program akan memanggil SAVE - Jika pengguna menginput N maka akan langsung keluar dari program 	QUIT	Gambar 6.14.1	Sesuai dengan hasil yang diharapkan

8 Pembagian Kerja dalam Kelompok

Nama Anggota	Pembagian Kerja
Muhammad Aidan Fathullah 18223002	<ul style="list-style-type: none"> • Membuat Repository Github • Membuat fitur Save, Work Challenge • Merevisi fitur Start • Menggabungkan keseluruhan file dan fix bug • Merapikan ADT
Vandega Arozan Musholine 18223010	<ul style="list-style-type: none"> • Membuat fitur Save, Store Supply, Store Remove • Membuat laporan • Membuat driver
Irdina Ilmuna Yosapat 18223060	<ul style="list-style-type: none"> • Membuat fitur Start, Quit, Store List • Membuat laporan bagian Ringkasan, Test Script, Lampiran, Data Test, Ringkasan • Membantu membuat laporan bagian Struktur Data bagian Struktur Data Mesin Kata, Struktur Kata User, dan Struktur Kata Queue.
Vincentia Belinda Sumartoyo 18223078	<ul style="list-style-type: none"> • Membuat ADT Mesin Kata • Membuat command Register, Login, Store Request • Menotulen asistensi • Membantu membuat laporan bagian Struktur Data bagian Struktur Data Mesin Karakter, Struktur Mesin Kata, User, Struktur Kata Queue. • Membuat laporan bagian Program Utama, Log Activity Anggota Kelompok, dan Algoritma menarik
Gabriela Jennifer Sandy 18223092	<ul style="list-style-type: none"> • Membuat ADT User dan ADT Mesin Karakter • Membuat command Register, Login, Logout, Work, Help • Membuat laporan bagian Struktur Data, Lampiran, Test Script, Fitur Tambahan • Membuat fitur README • Membuat UI dan merapikan fitur

9 Lampiran

9.1 Deskripsi Tugas Besar

PURRMART merupakan suatu sistem jual beli ke Borma yang diperlukan oleh Agen Purry. Ia mendapat misi untuk menghentikan proses pembuatan mesin penghancur yang dilakukan oleh Dr. Asep Spakbor dimana PURRMART ini akan mempermudah akses terhadap pasokan alat-alat tersebut yang berada di Toko Borma.

Buatlah sebuah aplikasi simulasi berbasis CLI (*command-line interface*). Sistem ini dibuat dalam bahasa C dengan menggunakan struktur data yang sudah kalian pelajari di mata kuliah ini. Kalian boleh menggunakan (atau memodifikasi) struktur data yang sudah kalian buat untuk praktikum pada tugas besar ini. Daftar ADT yang wajib digunakan dapat dilihat pada bagian Daftar ADT. *Library* yang boleh digunakan hanya `stdio.h`, `stdlib.h`, `time.h`, dan `math.h`.

System Mechanic

1. About the System

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

1. Menampilkan barang toko
2. Meminta dan menyuplai barang baru ke toko
3. Menyimpan dan membeli barang dalam keranjang
4. Menampilkan barang yang sudah dibeli
5. Membuat dan menghapus *wishlist*
6. 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.
- 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 dimasukkan ke toko. Pengguna dapat menerima, menunda, atau menolak permintaan.

1. Jika diterima, maka program akan meminta harga dari barang dan dimasukan ke toko.
2. Jika ditunda, maka barang akan kembali masuk ke antrian
3. 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.

9.2 Notulen Rapat


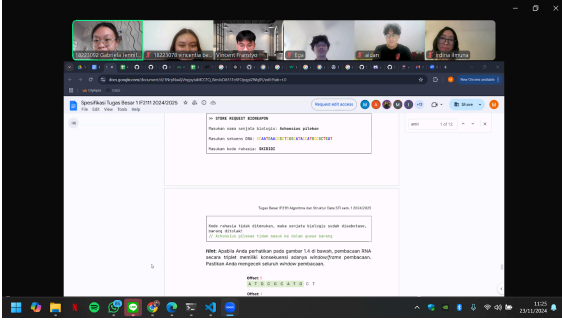



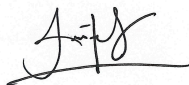
**Form Asistensi Tugas Besar
IF2111/Algoritma dan Struktur Data STI
Sem. 1 2024/2025**


No. Kelompok/Kelas : 03 / K-02
Nama Kelompok :
Anggota Kelompok (Nama/NIM) :

1. Muhammad Aidan Fathullah I 18223002
2. Vandega Arozan Musholine 18223010
3. Irdina Ilmuna Yosapat 18223060
4. Vincentia Belinda Sumartoyo 18223078
5. Gabriela Jennifer Sandy 18223092

Asisten Pembimbing : Vincent Franstyo

Asistensi I

Tanggal : Sabtu, 23 November 2024	Catatan Asistensi: <ol style="list-style-type: none"> 1. baiknya header dan ADT digabungkan driver diwajibkan jadi sebaiknya digabung satu file 2. help di spesifikasi, itu ada 3 state. welcome menu dulu dari start, load, help, quit. kalau udah ke login menu (register, login, help, quit), terus sisanya pas udah masuk. harus divalidasi, misal kayak work gak bisa dilakukan sebelum register/login. 3. di dalam folder ADT, ada file-file nya. struktur foldernya dirapihin dulu gitu, mau manggil headernya dari mana gitu, gabakal bingung lagi. 4. default config boleh mengikuti yang ada di spek. 5. enak nya ADT list nya dipisah, dinamis sama statis. drivernya jadinya juga misah gitu. 6. jangan lupa error handling, kayak masukin spasi doang/string/integer or something gitu. 7. disarankan buat make file 8. read me jangan lupa!!!
Tempat : Zoom	
Kehadiran Anggota Kelompok:	
No NIM Tanda tangan	
1 18223002	
	
2 18223010	
	
3 18223060	
	
4 18223078	
	
5 18223092	
	

	Tanda Tangan Asisten: 
--	--

9.3 Log Activity Anggota Kelompok

No.	Tanggal	Anggota	Aktivitas
1.	14/11/2024	18223002 Muhammad Aidan Fathullah I	Membuat repository Github
2.	14/11/2024	18223002 Muhammad Aidan Fathullah I	Kerja kelompok pertama dan pembagian tugas
		18223060 Irdina Ilmuna Yosapat	
		18223092 Gabriela Jennifer Sandy	
		18223078 Vincentia Belinda Sumartoyo	
		18223010 Vandega Arozan Musholine	
3.	18/11/2024	18223060 Irdina Ilmuna Yosapat	Push Start dan Quit
4.	19/11/2024	18223092 Gabriela Jennifer Sandy	Push ADT User
		18223078 Vincentia Belinda Sumartoyo	Push Register, ADT Mesin Kata
		18223002 Muhammad Aidan Fathullah I	Push all_headers.h dan update ADT Mesin Karakter
		18223010 Vandega Arozan Musholine	Push save
5.	19/11/2024	18223002 Muhammad Aidan Fathullah I	Kerja kelompok kedua
		18223060 Irdina Ilmuna Yosapat	

		18223092 Gabriela Jennifer Sandy	
		18223078 Vincentia Belinda Sumartoyo	
		18223010 Vandega Arozan Musholine	
6.	20/11/2024	18223078 Vincentia Belinda Sumartoyo	Push Login dan update Register, ADT Mesin Kata, Mesin Karakter dan user
7.	21/11/2024	18223002 Muhammad Aidan Fathullah I	Push ADT Barang dan update ADT Mesin Kata, update ADT function untuk User, Mesin Karakter, Mesin Kata dan push header. Finalisasi seluruh ADT yang diperlukan. Push load dan console.
8.	22/11/2024	18223078 Vincentia Belinda Sumartoyo	Update register, login
9.	23/11/2024	18223092 Gabriela Jennifer Sandy	Push logout dan update register serta login
		18223002 Muhammad Aidan Fathullah I	Update session-control
		18223060 Irdina Ilmuna Yosapat	Update start and quit
10.	24/11/2024	18223078 Vincentia Belinda Sumartoyo	Push store request
		18223010 Vandega Arozan Musholine	Push store supply dan store remove
		18223092 Gabriela Jennifer Sandy	Push work
		18223002 Muhammad Aidan Fathullah I	Update ADT
		18223060 Irdina Ilmuna Yosapat	Push store list

11.	25/11/2024	18223092 Gabriela Jennifer Sandy	Memisahkan ADT List, update logout, push help, dan create README
-----	------------	----------------------------------	--