


LAPORAN TUGAS BESAR
IF2111 Algoritma dan Struktur Data STI

PURRMART

Dipersiapkan oleh:
Kelompok 1

18221160	Seren Elizabeth Siahaan
18223006	Zaka Hanif Nabalah
18223022	Rayhan Hidayatul Fikri
18223036	Favian Rafi Laftiyanto
18223088	Wijasara Aptaluhung

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

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2111-TB1-K02-G08</i>		<i>51</i>
		<i>Revisi</i>	<i>0</i>	<i>25 November 2024</i>

Daftar Isi

Daftar Gambar	5
1 Ringkasan	8
2 Penjelasan Tambahan Spesifikasi Tugas	11
2.1 LOGIN	11
2.2 WORK	11
2.3 STORE SUPPLY	11
2.4 QUIT	12
2.5 Bio Weapon	12
3 Struktur Data (ADT)	12
3.1 Array	13
3.2 Arraydin	13
3.3 Mesinfrasa	14
3.4 Mesinkarakter	14
3.5 Mesinkata	14
3.6 Queue	15
4 Program Utama	15
5 Algoritma-Algoritma Menarik	18
5.1 Algoritma work_timer	18
5.2 Algoritma RNG	18
6 Data Test	19
6.1 Welcome Page	19
6.2 Start	20
6.3 Load <filename.txt>	20
6.4 Login	21
6.5 Logout	21
6.6 Register	22
6.7 Work	22

6.8 Work Challenge	23
a. Tebak Angka	24
6.9 Store List	26
6.10 Store Request	27
6.11 Store Supply	27
6.12 Store Remove	28
6.13 Help	28
6.14 Save	30
6.15 Quit	30
6.16 Bioweapon	31
7 Test Script	31
8 Pembagian Kerja dalam Kelompok	34
9 Lampiran	36
9.1 Deskripsi Tugas Besar	36
Command	
Pengguna dapat memasukkan command-command berikut.	36
a. START	36
b. LOAD <filename>	37
c. LOGIN	37
d. LOGOUT	37
e. REGISTER	37
f. WORK	37
g. WORK CHALLENGE	38
a) Tebak Angka	38
b) WORDL3	38
h. STORE LIST	39
i. STORE REQUEST	39
j. STORE SUPPLY	39
k. STORE REMOVE	39

l. HELP	39
m. SAVE <filename>	40
n. QUIT	40
9.2 Notulen Rapat	43
9.3 Log Activity Anggota Kelompok	45

Daftar Gambar

Gambar 1. Tampilan Welcome Page.....	20
Gambar 2. Tampilan Start ketika dijalankan.....	20
Gambar 3 Tampilan fitur Load <filename.txt> ketika filename.txt merupakan file dengan data yang tepat.....	20
Gambar 4 Tampilan fitur Load <filename.txt> ketika filename.txt merupakan file dengan data yang tidak tepat.....	20
Gambar 5 Tampilan fitur Load <filename.txt> ketika filename.txt merupakan file yang tidak ditemukan dalam sistem.....	21
Gambar 6. Tampilan Fitur login yang meminta inputan username dan password.....	21
Gambar 7. Tampilan ketika pengguna memasukkan data yang benar.....	21
Gambar 8. Tampilan ketika pengguna memasukkan data yang salah atau tidak ada data tersebut.....	21
Gambar 7. Tampilan jika pengguna menjalankan fungsi logout.....	22
Gambar 8. Tampilan saat pengguna menjalankan register.....	22
Gambar 9. Tampilan jika data yang dimasukkan sudah sesuai.....	22
Gambar 10. Tampilan jika data yang dimasukkan tidak sesuai atau sudah ada.....	22
Gambar 11. Tampilan saat pengguna menjalankan Work.....	23
Gambar 12. Tampilan saat pengguna telah memilih dan melakukan pekerjaan.....	23
Gambar 13. Tampilan saat pengguna menjalankan Work Challenge.....	24
Gambar 14. Tampilan saat pengguna memilih Tebak Angka sebagai challenge.....	24
Gambar 15. Tampilan saat pengguna benar menebak angka.....	24
Gambar 16. Tampilan saat pengguna salah dalam menebak angka.....	25
Gambar 17. Tampilan saat pengguna memilih WORDL399 sebagai challenge....	25

Gambar 18. Tampilan saat pengguna benar menebak kata.....	26
Gambar 19. Tampilan saat pengguna salah dalam menebak kata.....	26
Gambar 20. Tampilan jika pengguna menjalankan Store List.....	26
Gambar 21. Tampilan jika tidak ada barang di Store List.....	26
Gambar 22. Tampilan saat pengguna menjalankan Store Request.....	27
Gambar 23. Tampilan saat barang yang diminta sudah ada.....	27
Gambar 24. Tampilan saat barang yang diminta berhasil masuk ke dalam antrean.....	27
Gambar 25. Tampilan saat pengguna memasukkan command Store Supply.... 27	
Gambar 26. Tampilan saat barang yang diminta diterima.....	27
Gambar 27. Tampilan saat barang yang diminta ditunda.....	28
Gambar 28. Tampilan saat barang yang diminta tidak diterima / ditolak.....	28
Gambar 29. Tampilan saat pengguna memasukkan command Store Remove 28	
Gambar 30. Tampilan saat barang berhasil dihapus.....	28
Gambar 31. Tampilan saat barang gagal dihapus.....	28
Gambar 32. Tampilan pengguna memasukkan command Help saat pertama kali masuk ke dalam PURRMART.....	29
Gambar 33. Tampilan pengguna memasukkan command Help saat telah berhasil masuk ke dalam PURRMART.....	29
Gambar 32. Tampilan pengguna memasukkan command Help saat sudah masuk ke dalam akunnya.....	30
Gambar 33. Tampilan saat pengguna memasukkan command Save.....	30
Gambar 34. Tampilan saat pengguna memasukkan command Quit.....	31
Gambar 35. Tampilan saat pengguna berhasil menambahkan senjata biologis.....	31
Gambar 36. Ilustrasi Central Dogma.....	41
Gambar 37. Tabel Kodon-Asam Amino.....	42
Gambar 38. Singkatan Asam Amino.....	42

Gambar 39. Ilustrasi Sliding Window Pembacaan.....	43
--	----

1 Ringkasan

PURRMART merupakan program simulasi *e-commerce* yang memungkinkan pengguna untuk melakukan transaksi pembelian barang. Untuk menggunakan program, pengguna perlu mengakses akun lamanya atau membuat akun baru. Setelah mengakses akun, pengguna dapat melihat daftar barang, membuat permintaan masuknya barang baru atau tambahan stok barang yang ada, menampilkan riwayat pembelian barang, serta membuat dan menghapus daftar barang yang diinginkan. Pengguna dapat memanfaatkan uang yang didapatkannya untuk membeli barang. Hal ini dapat dilakukan pengguna dengan bekerja yang kemudian pengguna akan mendapatkan uang.

Di samping itu, terdapat cara lain untuk memperoleh uang, hal ini bisa dilakukan dengan berpartisipasi pada permainan. Permainan ini dapat dilakukan dengan syarat pengguna harus membayar nominal tertentu yang tidak akan dikembalikan ketika kalah maupun menang. Terdapat 2 permainan yang dapat dipilih pengguna, yakni Tebak Angka dan WORDL3. Pengguna juga dapat menyimpan kondisi sistem terkait barang dan user ke dalam sebuah *save file*. Apabila pengguna sudah selesai menggunakan sistem, pengguna dapat melakukan logout untuk keluar dari sesi atau quit untuk keluar dari sistem. Jika pengguna hanya logout, maka pengguna dapat kembali login atau memulai sesi baru dengan membuat akun baru. Jika pengguna memilih untuk keluar, maka pengguna dapat memilih untuk menyimpan atau tidak program lalu keluar dari sistem.

Program PURRMART menggunakan bahasa C dan sejumlah *Abstract Data Type* (ADT) sebagai struktur data yang diproses. Terdapat 6 ADT yang digunakan, yakni ADT User, ADT Barang, ADT List, ADT Mesin Karakter, ADT Mesin Kata, dan ADT Queue. ADT User dan ADT Barang digunakan untuk menyimpan informasi barang di toko atau di daftar permintaan serta menyimpan informasi akun user yang menggunakan program. ADT List diimplementasikan menggunakan struktur data array statis dan dinamis. ADT List digunakan untuk menyimpan struktur data barang dan user. ADT Mesin

Karakter dan Mesin Kata digunakan untuk memproses masukan pengguna dan pembacaan file dengan struktur data Word. ADT Queue digunakan untuk memproses daftar permintaan barang.

Ketika program PURRMART dijalankan, pengguna akan diberikan pilihan untuk melakukan START, LOAD, atau HELP. Masukan pengguna yang diterima hanyalah salah satu dari pilihan tersebut. Apabila masukan pengguna tidak diterima, akan muncul permintaan untuk memasukkan kembali perintah yang valid. Perintah START akan memproses *file* konfigurasi yang memuat daftar barang dan user standar bawaan program. Perintah LOAD yang disertai nama *file* dengan format txt akan memuat *file* hasil proses SAVE yang disimpan oleh pengguna pada sesi penggunaan program sebelumnya. Isi dari *file* tersebut belum tentu sama dengan *file* konfigurasi.

Setelah menjalankan salah satu perintah, pengguna akan memulai sesi dan diberikan pilihan untuk menjalankan perintah REGISTER, LOGIN, atau QUIT. Perintah REGISTER digunakan untuk membuat akun baru dengan *username* dan *password* yang hanya terdiri atas satu kata sesuai masukan pengguna lalu memulai sesi. Perintah LOGIN digunakan untuk masuk ke akun yang sudah terdaftar di sistem sesuai daftar *user* yang terdapat di *file* konfigurasi atau *save file* dan memulai sesi. Perintah QUIT dapat digunakan untuk keluar dari program. Jika pengguna menjalankan perintah QUIT, pengguna dapat memilih untuk menyimpan kondisi sesi saat itu dalam sebuah *save file* dengan nama sesuai masukan pengguna atau tidak. Pada kondisi ini, masukan selain ketiga perintah tersebut akan ditolak dan perintah baru akan diminta oleh program.

Setelah pengguna melakukan LOGIN, pengguna dapat menjalankan perintah WORK, WORK CHALLENGE, STORE LIST, STORE REQUEST, STORE SUPPLY, STORE REMOVE, LOGOUT, SAVE, dan QUIT. Perintah STORE LIST dapat digunakan untuk menampilkan daftar barang yang ada di toko. Perintah STORE REQUEST dapat digunakan untuk membuat permintaan penambahan barang baru ke daftar barang di toko dengan masukan pengguna sebagai nama barang. Permintaan tersebut akan diproses menggunakan perintah STORE SUPPLY.

Jika permintaan diterima, program akan meminta masukan pengguna sebagai harga barang dan memasukkan barang tersebut ke daftar barang di toko. Jika permintaan ditolak, program akan menghapus barang tersebut dari daftar barang di toko. Jika permintaan ditunda, barang akan dikembalikan ke daftar permintaan. Ada pula perintah STORE REMOVE yang dapat digunakan untuk menghapus barang dari daftar barang yang tersedia di toko dengan masukan pengguna sebagai nama barang yang akan dihapus dari daftar barang.

Perintah WORK akan menampilkan daftar pekerjaan yang tersedia beserta uang yang dihasilkan dan durasi pekerjaan. Program akan meminta masukan pengguna berupa nama pekerjaan yang dipilih. Kemudian, program akan memulai sesi bekerja dengan kondisi pengguna tidak dapat mengakses fitur program lainnya dan harus menunggu hingga durasi pekerjaan berakhir. Kemudian, uang yang dihasilkan dari sesi bekerja akan disimpan untuk akun pengguna yang sedang digunakan.

Perintah WORK CHALLENGE akan menampilkan daftar permainan yang tersedia dan biaya permainan. Program akan meminta masukan pengguna berupa nama permainan yang ingin dimainkan, yaitu Tebak Angka atau WORDL3. Jika pengguna ingin keluar dari sesi, pengguna dapat menjalankan perintah LOGOUT dan kembali ke menu REGISTER, LOGIN, dan QUIT yang dimasuki setelah START atau LOAD. Jika pengguna ingin menyimpan kondisi program saat itu, pengguna dapat menjalankan perintah SAVE. Pengguna dapat keluar dari program dengan menjalankan perintah QUIT.

2 Penjelasan Tambahan Spesifikasi Tugas

Pada pengembangan sistem aplikasi PURRMART ini, kami melakukan penambahan – penambahan spesifikasi berupa asumsi yang kemudian akan dijelaskan dan kami rincikan. Spesifikasi tambahan yang dimaksud adalah sebagai berikut:

2.1 LOGIN

Pada fungsi ini, kami menambahkan kondisi saat *user* sudah LOGIN, tetapi *user* memanggil fungsi lagi. Pada kondisi ini, program akan mencetak tulisan:

```
Anda masih tercatat sebagai {username}. Silakan LOGOUT terlebih dahulu.
```

2.2 WORK

Pada fungsi ini, kami menambahkan kondisi saat *user* memasukkan pekerjaan yang tidak tersedia pada daftar pekerjaan. Pada kondisi ini, program akan mencetak tulisan:

```
Pekerjaan Yang Dipilih Tidak Valid.
```

2.3 STORE SUPPLY

Pada fungsi ini, kami menambahkan kondisi ketika *user* memasukkan perintah yang tidak valid, yaitu bukan 'Terima', 'Tunda', dan 'Tolak'. Pada kondisi ini, program akan menunjukkan *error text* dan *user* akan diarahkan kembali ke Menu. Error text-nya adalah sebagai berikut.

```
Perintah tidak valid.
```

Lalu, kami juga menambahkan kondisi ketika *user* memanggil fungsi ini dalam kondisi antrian kosong. Pada kondisi ini, program akan mencetak tulisan:

```
Tidak ada barang yang ada di antrian.
```

Kami juga menambahkan kondisi ketika *user* menerima *request* penambahan barang dan memasukkan harga barang yang tidak valid, yaitu ketika harga barang bukanlah angka. Pada kondisi ini, program akan menunjukkan *error text* dan *user* akan diarahkan kembali ke Menu.

Error text-nya adalah sebagai berikut.

Angka yang kamu masukkan tidak valid.

2.4 QUIT

Pada fungsi ini, kami berasumsi bahwa ketika *user* ingin menyimpan data pada sesi sekarang, data akan disimpan pada file config. Asumsi ini kami buat karena tidak ada keterangan pada spesifikasi ketika pengguna SAVE melalui fungsi QUIT, file harus disimpan ke mana.

2.5 Bio Weapon

Pada fungsi ini, kami berasumsi bahwa maksimal panjang sekuens DNA yang dapat diproses adalah 60 huruf. Lalu, kami juga membuat asumsi bahwa ada kemungkinan *user* ingin memasukkan barang berupa senjata biologis yang telah ada di toko atau di antrian. Maka dari itu, kami menambahkan kondisi ketika *user* menambahkan barang berupa senjata biologis yang telah ada di toko. Pada kondisi ini, program akan mencetak tulisan:

Barang dengan nama yang sama sudah ada di toko.

Lalu, kami juga menambahkan kondisi ketika *user* menambahkan barang berupa senjata biologis yang telah ada di antrian. Pada kondisi ini, program akan mencetak tulisan:

Barang dengan nama yang sama sudah ada di antrian.
--

3 Struktur Data (ADT)

Pada pengembangannya *Abstract Data Type* (ADT) menjadi pondasi utama pada perancangan fitur - fitur yang ada. Mengutip dari MIT OpenCourseWare, sejatinya struktur data ini dapat didefinisikan sebagai model konseptual untuk mengorganisasi dan memanipulasi data yang didefinisikan

berdasarkan operasi yang dapat dilakukan terhadap data. Berikut merupakan penjelasan mengenai struktur data yang digunakan pada setiap command dalam program ini.

3.1 Array

File `array.h` mengimplementasikan ADT Array Statis, di mana array memiliki ukuran tetap yang telah ditentukan sebelumnya, dan jumlah elemen yang dapat disimpan dibatasi. Pada ADT ini, array digunakan untuk menyimpan data pengguna yang terdiri dari nama, kata sandi, dan saldo uang dalam bentuk struktur `User`. Beberapa komponen utama dalam file ini adalah konstanta seperti `IdxMax`, `IdxMin`, dan `IdxUndef`, yang digunakan untuk mendefinisikan batas indeks array. Selain itu, ada tipe data `IdxType` untuk indeks array bertipe `int` dan struktur `TabInt` yang menyimpan array pengguna dengan panjang maksimal $\text{IdxMax} - \text{IdxMin} + 1$ dan variabel `Neff` untuk menyimpan jumlah elemen yang terisi. Fungsi-fungsi dalam file ini digunakan untuk mengelola array, seperti menambah, mengubah, atau mengakses elemen, serta memeriksa apakah array kosong atau penuh. Dalam program ini, kami membatasi ukuran array di angka 100 sehingga banyak elemen yang disimpan hanya bisa termuat hingga sebanyak 100 elemen.

3.2 Arraydin

ADT `arraydin.h` mengimplementasikan ADT Arraydin, di mana array dapat berubah ukurannya sesuai dengan jumlah elemen yang dimilikinya, memberikan fleksibilitas lebih dibandingkan array statis. ADT ini digunakan untuk menyimpan data barang dalam struktur `Barang`, yang mencakup nama dan harga barang. Beberapa komponen dalam file ini antara lain konstanta seperti `InitialSize` untuk menentukan ukuran awal array dan `MAX_LEN` yang menentukan panjang maksimal nama barang. File ini juga mendefinisikan tipe data `IdxType` untuk indeks array yang bertipe `int` dan struktur `ArrayDin` yang menyimpan array barang beserta kapasitas dan jumlah elemen yang terisi. Fungsi-fungsi dalam file ini digunakan untuk mengelola array dinamis, seperti menambah, menghapus, atau memodifikasi elemen, serta memeriksa apakah array penuh atau kosong.

Jadi, array.h mengimplementasikan ADT Array Statis untuk mengelola data pengguna, sementara arraydin.h mengimplementasikan ADT Array Dinamis untuk mengelola data barang.

3.3 Mesinfrasa

ADT ini mengimplementasikan ADT Mesinfrasa, yang merupakan pengembangan dari Mesin Karakter dan **Mesinkata**. Mesin Frasa ini berfungsi untuk memproses input dalam bentuk frasa (kelompok kata) yang lebih panjang. Tujuannya adalah agar sistem bisa menangani lebih dari satu kata sekaligus, membentuk sebuah frasa yang utuh. Sehingga, ADT ini memperluas kapasitas pengolahan dari kata menjadi kalimat atau frasa yang lebih kompleks. Fungsi dalam file ini memungkinkan pemisahan dan pengolahan frasa dengan memanfaatkan operasi dari Mesin Karakter. Dalam program ini, kami membatasi frasa maksimal terdiri atas 100 karakter.

3.4 Mesinkarakter

ADT ini mengimplementasikan ADT Mesinkarakter yang digunakan untuk membaca dan memproses input karakter dari sebuah file atau sumber lain. Pada ADT ini, setiap karakter yang dibaca akan diproses satu per satu menggunakan fungsi-fungsi yang tersedia. File ini berfungsi untuk mempermudah pengelolaan input karakter dan menyediakan operasi seperti membaca karakter, memeriksa apakah karakter tersebut adalah EOF (*End Of File*), atau mengubah status pembacaan karakter. Struktur data yang digunakan dalam file ini biasanya adalah buffer yang menyimpan karakter yang dibaca, dengan fungsi untuk mengambil karakter satu per satu dan mengelola posisi pembacaan.

3.5 Mesinkata

ADT ini mengimplementasikan ADT Mesin Kata yang merupakan perluasan dari **Mesinkarakter**, di mana fungsinya untuk membaca dan memproses kata-kata dalam input, bukan hanya karakter. ADT ini menggunakan **mesinkarakter** untuk membaca karakter-karakter hingga membentuk sebuah kata. Setelah sebuah kata terbentuk, mesin kata akan memberikan akses untuk mengolah kata tersebut. Struktur data dalam file ini

memungkinkan pengelolaan kata dengan cara memisahkan input berdasarkan pemisah seperti spasi atau tanda baca. Fungsi-fungsi yang tersedia di dalam file ini antara lain untuk membaca kata, memeriksa apakah masih ada kata yang tersisa, dan mengolah kata yang terbaca untuk diproses lebih lanjut. Dalam program ini, kami membatasi kata maksimal terdiri atas 100 karakter.

3.6 Queue

ADT ini mengimplementasikan ADT Queue atau antrian, yang merupakan struktur data untuk menyimpan elemen secara berurutan dengan prinsip FIFO (*First In, First Out*), di mana elemen pertama yang masuk akan menjadi elemen pertama yang keluar. Queue ini biasanya digunakan untuk mengelola data yang datang dalam urutan tertentu, seperti proses antrian pada sistem. Struktur data yang digunakan dalam file ini adalah array atau linked list yang menyimpan elemen-elemen dalam urutan yang tetap. Fungsi-fungsi yang tersedia di dalam file ini termasuk menambah elemen ke dalam antrian (*enqueue*), menghapus elemen dari antrian (*dequeue*), serta memeriksa apakah antrian kosong atau penuh. Queue sering digunakan dalam aplikasi yang membutuhkan antrian proses atau data, seperti antrian pelanggan atau antrian pesan. Dalam program ini, kami memanfaatkan ADT Queue dalam membuat antrian dua variabel: Barang dan Char. Oleh karena itu, terdapat dua ADT Queue yang kami buat, yaitu Queue dan QueueBarang. Kapasitas Queue yang kami buat kami batasi di angka 100 sehingga banyak elemen yang berada dalam barisan hanya bisa termuat hingga 100 elemen.

4 Program Utama

Program utama dari PURRMART sejatinya diimplementasikan pada file main.c yang di dalamnya mengandung file *header* dari ADT, fungsi - fungsi, beserta file config. Ketika program utama ini dipanggil, maka program kemudian akan menampilkan *landing page* dari program kami. Pada *landing page* akan ditampilkan ASCII art yang terdapat di dalam file banner.txt. Setelah *landing page* ditampilkan, pengguna akan diminta untuk memasukkan

command pertama kali. Pada kesempatan kali ini, *command* yang dapat diterima adalah START, LOAD, dan QUIT.

Jika pengguna memberi input START, maka program akan membaca file konfigurasi default dari program. Namun, jika pengguna memberi input LOAD <filename.txt>, maka program akan membaca file.txt yang dipanggil tersebut. Terakhir, jika pengguna memberi input QUIT, maka program akan berhenti dan pengguna akan dikeluarkan dari sistem PURRMART.

Setelah file terbaca oleh program, pengguna dapat memanggil *set command* yang berbeda dari sebelumnya, *set command* yang dimaksud yaitu *command* REGISTER, LOGIN, dan QUIT. Melalui *command* REGISTER, pengguna dapat mendaftarkan *username* beserta *password* baru yang belum terdapat pada file yang dibaca saat mereka memanggil *command* START ataupun LOAD. Sedangkan melalui *command* LOGIN, pengguna dapat masuk ke dalam PURRMART dengan menggunakan *username* dan *password* yang sudah terdaftar pada file yang dibaca. Meskipun bukan pada *Main Menu*, pengguna dapat melakukan SAVE melalui *command* QUIT, dimana data pengguna terbaru akan disimpan pada file konfigurasi default.

Setelah melakukan LOGIN, pengguna memasuki *main menu* dari program dan dapat memanggil *set command* baru yang berbeda dari pada *set command* sebelumnya. *Set Command* yang dimaksud yaitu WORK, WORK CHALLENGE, STORE LIST, STORE REQUEST, STORE REQUEST BIOWEAPON, STORE SUPPLY, STORE REMOVE, LOGOUT, SAVE, dan QUIT. Melalui *command* WORK, pengguna dapat memperoleh uang sesuai dengan pekerjaan yang diambil. Sebelum memilih, pengguna akan ditampilkan terlebih dahulu data pekerjaan beserta gaji yang akan didapat jika kita melakukan pekerjaan tersebut. Disamping itu, pengguna juga dapat memanfaatkan *command* WORK CHALLENGE untuk menambahkan uang selain dengan cara bekerja. Pengguna dapat bermain *mini game* berupa Tebak Angka dan Wordl3 dengan membayar sejumlah biaya yang telah ditentukan pada *minigame*. Dalam *game* Tebak Angka, pengguna harus menebak angka dari *range* 0 hingga 100. Sedangkan pada Wordl3, pengguna diberi enam kali kesempatan

untuk menebak sebuah kata bahasa Inggris yang terdiri atas lima huruf. Melalui *minigame* tersebut, pengguna bisa mendapatkan reward berupa uang tergantung kecepatan pengguna menebak angka atau kata secara tepat.

Pada PURRMART, juga terdapat 4 opsi *command* STORE yang dapat dipanggil oleh pengguna. Pertama, pada *command* STORE LIST program akan menampilkan kepada pengguna Barang yang terdapat pada toko. Kedua, melalui *command* STORE REQUEST, di mana pengguna dapat memberi masukan ke toko berupa Barang yang ingin dijual, adapun masukan tersebut akan disimpan dalam sebuah Queue Request. Dalam hal ini, data barang yang ingin dimasukkan tidak akan langsung masuk ke dalam etalase toko. Ketiga, pada *command* STORE SUPPLY, pengguna akan memiliki opsi **terima**, **tunda**, atau **tolak** kepada Barang yang ada pada Queue Request (mulai dari urutan pertama Queue, sesuai konsep FIFO pada Queue). Apabila pengguna memberi *input* terima, maka pengguna dapat memberikan harga untuk Barang tersebut. Jika pengguna memberi *input* tunda, maka Barang yang di-*request* akan dikembalikan ke urutan terakhir pada Queue Request. Jika pengguna memberi *input* tolak, maka Barang akan dihapus dari Queue Request. Terakhir, pada *command* STORE REMOVE, Pengguna dapat memberi *input* berupa Nama Barang yang ingin dihapus dari etalase toko. Program kemudian akan menghapus Barang yang diinput pengguna akan dihapus dari array *items*.

Selain fungsi-fungsi tersebut, terdapat Command LOGOUT, SAVE, dan QUIT. Melalui *command* LOGOUT, pengguna dapat keluar dari pengguna, dan kembali ke sesi atau menu sebelumnya. Dalam hal ini, pengguna dapat kembali melakukan LOGIN atau REGISTER. Selain itu, pengguna juga dapat memberi *input command* SAVE <filename.txt>, untuk menyimpan data terbaru Barang dari arrayItems dan data terbaru Users dari arrayUsers ke dalam file txt tujuan sesuai *input* pengguna. Terakhir, melalui *command* QUIT, pengguna akan ditawarkan untuk melakukan SAVE. Apabila pengguna menjawab "Y" atau "y" (*read: Yes*), **maka program akan melakukan SAVE**

pada file **config.txt**. Apabila menjawab "N" atau "n" (*read: No*), maka program akan berhenti tanpa melakukan SAVE apapun.

5 Algoritma-Algoritma Menarik

Pada pengerjaan setiap spesifikasi yang ada pada PURRMART, kami menemukan beberapa algoritma menarik. Kami mendefinisikan algoritma menarik sebagai algoritma baru dan unik yang tidak terdapat di dalam spesifikasi. Algoritma ini kemudian menantang kami untuk mengerjakan dengan semangat sebab konsepnya yang menantang.

5.1 Algoritma work_timer

```
void work_timer(int second){
    int i = 0;
    clock_t start = clock();
    while (i < second){
        printf("%d/%d\n", i, second);
        i++;
        while (clock() < start + (i * 1000));
    }
    printf("%d/%d out\n", i, second);
}
```

Algoritma work_timer merupakan algoritma yang fungsinya sebagai timer pada fungsi WORK. Algoritma ini bekerja dengan mencatat waktu mulai dipanggilnya fungsi, lalu membandingkannya dengan waktu mulai yang sudah ditambahkan oleh jumlah detik yang diinginkan. Selain itu juga algoritma ini akan mencatat progres timer tiap detiknya agar pengguna dapat mengetahui progres dari WORK yang dijalankan.

5.2 Algoritma RNG

```
int RNG(int min, int max){
    srand(time(NULL));
```

```
int v1 = rand();
srand(v1);
return (rand() * v1) % (max - min + 1) + min;
}
```

Algoritma RNG merupakan algoritma yang berfungsi untuk mendapatkan angka random pada suatu range. Algoritma ini digunakan pada WORK CHALLENGE (Tebak Angka dan Word13). Algoritma ini memanfaatkan time dan hasil rand v1 sebagai seed-nya, sehingga angka hasil RNG semakin acak dan tidak beraturan.

6 Data Test

Pada bagian ini, akan disajikan beberapa data uji yang digunakan untuk menguji berbagai fitur dari program yang telah dikembangkan. Setiap data uji dirancang untuk menguji fitur tertentu dalam program dan diharapkan menghasilkan keluaran yang sesuai dengan yang diinginkan. Penjelasan untuk tiap data uji akan mencakup fitur apa saja yang diuji, hasil yang seharusnya diberikan oleh sistem, serta informasi tambahan yang relevan jika diperlukan. Data uji yang diberikan akan disertai dengan input yang digunakan, proses yang dilakukan oleh sistem, dan output yang dihasilkan sebagai hasil dari eksekusi program. Dengan data uji ini, dapat dievaluasi apakah sistem berfungsi sesuai dengan ekspektasi dan memenuhi persyaratan yang ditetapkan.

6.1 Welcome Page

Percobaan ini akan menguji fitur pembuka dari PURRMART. Pengecekan dilakukan dengan menjalankan perintah berikut di terminal.

```
cd bin
mingw32-make -f makefile
./output
```



Gambar 1. Tampilan Welcome Page

6.2 Start

Pengujian ini berfungsi untuk menguji fitur START dengan mengambil data dari config.txt. Setelah memasukkan perintah "START", pengguna dapat memasukkan perintah - perintah lain untuk menjalankan program.

```
Masukkan command: START
File konfigurasi berhasil dibaca. PURRMART berhasil dijalankan.
```

Gambar 2. Tampilan Start ketika dijalankan

6.3 Load <filename.txt>

Pengujian ini bertujuan dalam menguji fitur **load** saat menjalankan program PURRMART dan mengambil data dari file txt yang diinginkan. Setelah memasukkan perintah ini, pengguna dapat menjalankan

```
Masukkan command: LOAD dummyfile.txt
Save file berhasil dibaca. PURRMART berhasil dijalankan.
```

Gambar 3 Tampilan fitur Load <filename.txt> ketika filename.txt merupakan file dengan data yang tepat

```
Masukkan command: LOAD invalid.txt
File tidak memuat data yang sesuai. PURRMART gagal dijalankan.
```

Gambar 4 Tampilan fitur Load <filename.txt> ketika filename.txt merupakan file dengan data yang tidak tepat

```
Masukkan command: LOAD arealfile.txt
Save file tidak ditemukan. PURRMART gagal dijalankan.
```

Gambar 5 Tampilan fitur Load <filename.txt> ketika filename.txt merupakan file yang tidak ditemukan dalam sistem

6.4 Login

Pengujian ini bertujuan dalam menguji fitur **login** yang digunakan untuk pengguna dapat masuk ke dalam akunnya. Dalam konteks ini data pengguna telah ada disimpan di config.txt. Saat menjalankan perintah ini, pengguna harus memasukkan data username dan password yang sesuai. Jika data telah sesuai, pengguna akan diarahkan ke menu selanjutnya.

```
Masukkan command: LOGIN
Username: admin
Password: 
```

Gambar 6. Tampilan Fitur login yang meminta inputan username dan password

```
Masukkan command: LOGIN
Username: admin
Password: alstrukdatkeren

Anda telah login ke PURRMART sebagai admin.
```

Gambar 7. Tampilan ketika pengguna memasukkan data yang benar

```
Username: arealusername
Password: verysecurepassword

Username atau password salah.
```

Gambar 8. Tampilan ketika pengguna memasukkan data yang salah atau tidak ada data tersebut

6.5 Logout

Pengujian **logout** ditujukan untuk menguji pengguna dapat keluar dari akun yang telah *login* sebelumnya. Saat menjalankan perintah ini pengguna akan langsung keluar dari akunnya dan membuat `statusLogin` menjadi `False`.

```
Masukkan command: LOGOUT
admin@ telah logout dari sistem PURRMART. Silakan REGISTER/LOGIN kembali untuk melanjutkan.
```

Gambar 7. Tampilan jika pengguna menjalankan fungsi logout

6.6 Register

Pengujian **register** ditujukan untuk pengguna dapat mendaftarkan akunnya ke dalam sistem. Saat perintah dijalankan, pengguna akan diminta untuk memasukkan *username* dan *password* yang diinginkan. Dalam hal ini, *username* dan *password* harus dalam kondisi yang unik. Sehingga diharapkan apabila *username* dan *password* dalam kondisi tidak unik, maka pengguna diharapkan memasukkan ulang data *username* dan *password* yang diinginkan.

```
Masukkan command: REGISTER
Username: zack2301
Password: 
```

Gambar 8. Tampilan saat pengguna menjalankan register

```
Masukkan command: REGISTER
Username: zack2301
Password: verysecurepassword
Akun dengan username zack2301 telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.
```

Gambar 9. Tampilan jika data yang dimasukkan sudah sesuai

```
Masukkan command: REGISTER
Username: admin
Password: password
Akun dengan username admin telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.
```

Gambar 10. Tampilan jika data yang dimasukkan tidak sesuai atau sudah ada

6.7 Work

Pengujian **work** ditujukan untuk melihat keberhasilan fitur ini menambahkan uang setelah pengguna melakukan pekerjaan yang dipilihnya. Saat menjalankan program, pengguna akan ditampilkan terlebih dahulu jenis pekerjaan dan gaji yang didapat jika melakukan pekerjaan tersebut. Pengguna kemudian diminta untuk memilih pekerjaan yang dia mau. Setelah itu program diharapkan bekerja dan kemudian akan menunggu hingga

pekerjaan selesai. Jika pekerjaan telah selesai, diharapkan pengguna mendapatkan uang atas kerja yang telah dilakukannya.

```
Masukkan command: work
Daftar pekerjaan:
  1. Evil Lab Assistant (pendapatan=100, durasi=14)
  2. OWCA Hiring Manager (pendapatan=4200, durasi=21)
  3. Cikapundunginator Caretaker (pendapatan=7000, durasi=30)
  4. Mewing Specialist (pendapatan=1000, durasi=22)
  5. Inator Connoisseur (pendapatan=997, durasi=15)
Masukkan pekerjaan yang dipilih: █
```

Gambar 11. Tampilan saat pengguna menjalankan Work

```
Masukkan pekerjaan yang dipilih: 5
Anda sedang bekerja sebagai INATOR CONNOISSEUR... harap tunggu.
0/15
1/15
2/15
3/15
4/15
5/15
6/15
7/15
8/15
9/15
10/15
11/15
12/15
13/15
14/15
15/15 out
Pekerjaan selesai, +997 rupiah telah ditambahkan ke akun Anda.(Uang Sekarang = 997)
```

Gambar 12. Tampilan saat pengguna telah memilih dan melakukan pekerjaan

6.8 Work Challenge

Pengujian fitur **work challenge** ini ditujukan untuk melihat keberhasilan fitur ini menambahkan uang setelah pengguna melakukan *challenge* di OWCA yang dipilihnya. Saat menjalankan program ini pengguna akan ditampilkan daftar *challenge* di OWCA beserta dengan biaya bermain. Untuk melakukan *challenge*, pengguna harus membayar sesuai dengan biaya bermain yang tertera. Ketika pengguna telah memilih *challenge*, kemudian mereka akan diarahkan untuk bermain *game*-nya.

```
Masukkan command: WORK CHALLENGE
Daftar pekerjaan:
  1. TebakAngka (biaya=200)
  2. WORDL399 (biaya=500)
Masukkan challenge yang dipilih: |
```

Gambar 13. Tampilan saat pengguna menjalankan Work Challenge

a. Tebak Angka

Saat pengguna memilih game “Tebak Angka”, sistem akan memilih angka secara *random*. Kemudian pengguna akan diminta untuk mengerjakan *challenge* yang diberikan dengan menebak angka yang sesuai.

```
Masukkan challenge yang dipilih: 1
Uang Berkurang 200 (Uang Sekarang = 797)
Tebak Angka: |
```

Gambar 14. Tampilan saat pengguna memilih Tebak Angka sebagai challenge

```
Tebak Angka: 50
Tebakanmu lebih kecil!
Tebak Angka: 75
Tebakanmu lebih kecil!
Tebak Angka: 85
Tebakanmu lebih kecil!
Tebak Angka: 90
Tebakanmu lebih kecil!
Tebak Angka: 95
Tebakanmu lebih kecil!
Tebak Angka: 97
Tebakanmu benar!
+500 rupiah telah ditambahkan ke akun Anda.(Uang Sekarang = 1297)
```

Gambar 15. Tampilan saat pengguna benar menebak angka


```

Tebak Angka: 50
Tebakanmu lebih besar!
Tebak Angka: 25
Tebakanmu lebih besar!
Tebak Angka: 75
Tebakanmu lebih besar!
Tebak Angka: 1
Tebakanmu lebih kecil!
Tebak Angka: 0
Tebakanmu lebih kecil!
Tebak Angka: 4
Tebakanmu lebih kecil!
Tebak Angka: 6
Tebakanmu lebih kecil!
Tebak Angka: 8
Tebakanmu lebih kecil!
Tebak Angka: 9
Tebakanmu lebih kecil!
Tebak Angka: 10
Failed

```

Gambar 16. Tampilan saat pengguna salah dalam menebak angka

b. WORDL399

Saat pengguna memilih game “WORDL399”, sistem akan memilih kata secara *random* yang diambil dari file word.txt. Kemudian pengguna akan diminta untuk mengerjakan *challenge* yang diberikan dengan menebak kata.

```

Masukkan challenge yang dipilih: 2
Uang Berkurang 500 (Uang Sekarang = 597)
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
Masukan Kata Tebakan: 

```

Gambar 17. Tampilan saat pengguna memilih WORDL399 sebagai challenge

```

Masukan Kata Tebakan: COIFS
Hasil:
W% H% O* L% E%
C H% A% I* R%
P% L% A% N% K%
C O I F S
- - - - -
- - - - -
Menang!!
+1500 rupiah telah ditambahkan ke akun Anda.(Uang Sekarang = 1997)

```

Gambar 18. Tampilan saat pengguna benar menebak kata

```

Masukan Kata Tebakan: LIDAR
Hasil:
W% H% O* L% E%
C% H% A* I% R
R* A* Z% O* R
A* M% O* U* R
O* R* D* E% R
L% I% D* A R
Kalah
Jawaban: DOUAR

```

Gambar 19. Tampilan saat pengguna salah dalam menebak kata

6.9 Store List

Pengujian fitur ini ditujukan untuk melihat apakah fitur ini bisa menampilkan data - data barang yang ada di dalam store. Ketika pengguna menjalankan *command* ini, kemudian akan ditampilkan data - data barang yang ada di dalam store.

```

Masukkan command: STORE LIST
List barang yang ada di toko :
- Shrinkinator
- Landslideinator

```

Gambar 20. Tampilan jika pengguna menjalankan Store List

```

Masukkan command: STORE LIST
TOKO KOSONG

```

Gambar 21. Tampilan jika tidak ada barang di Store List

6.10 Store Request

Pengujian ini ditujukan untuk melihat apakah pengguna dapat melakukan *request* produk saat *command* ini dijalankan. Saat pengguna menjalankan program ini, pengguna akan diminta memasukkan barang apa yang ingin ia *request*. Jika nama barang belum ada, maka kemudian akan dimasukkan ke dalam *queue* atau antrian.

```
Masukkan command: STORE REQUEST
Nama barang yang diminta: 
```

Gambar 22. Tampilan saat pengguna menjalankan Store Request

```
Masukkan command: STORE REQUEST
Nama barang yang diminta: Landslideinator
Barang dengan nama yang sama sudah ada di toko!
Masukkan command: STORE REQUEST
Nama barang yang diminta: Bantal
Barang dengan nama yang sama sudah ada di antrian!
```

Gambar 23. Tampilan saat barang yang diminta sudah ada

```
Masukkan command: STORE REQUEST
Nama barang yang diminta: Bantal
Barang telah berhasil dimasukkan ke dalam antrian.
```

Gambar 24. Tampilan saat barang yang diminta berhasil masuk ke dalam antrian

6.11 Store Supply

Pengujian ini ditujukan untuk melihat data *request* barang yang masuk. Apakah kemudian data tersebut dapat diterima atau tidak. Terdapat tiga kondisi dalam hal ini, barang diterima, ditunda, atau ditolak.

```
Masukkan command: STORE SUPPLY
Apakah kamu ingin menambahkan barang Guling: 
```

Gambar 25. Tampilan saat pengguna memasukkan command Store Supply

```
Masukkan command: STORE SUPPLY
Apakah kamu ingin menambahkan barang Bantal: Terima
Harga barang: 150
Bantal dengan harga 150 telah ditambahkan ke toko.
```

Gambar 26. Tampilan saat barang yang diminta diterima

```
Masukkan command: STORE SUPPLY
Apakah kamu ingin menambahkan barang Guling: Tunda
Guling dikembalikan ke antrian.
```

Gambar 27. Tampilan saat barang yang diminta ditunda

```
Masukkan command: STORE SUPPLY
Apakah kamu ingin menambahkan barang Selimut: Tolak
Selimut dihapuskan dari antrian.
```

Gambar 28. Tampilan saat barang yang diminta tidak diterima / ditolak

6.12 Store Remove

Pengujian *command* ini bertujuan untuk menghapus barang yang ada di dalam *list* barang pada *store*. Ketika pengguna menjalankan program ini, maka pengguna akan diminta untuk memasukkan nama barang yang ingin dihapus. Barang dapat dihapus jika barang tersebut benar - benar ada.

```
Masukkan command: STORE REMOVE
Nama barang yang akan dihapus: 
```

Gambar 29. Tampilan saat pengguna memasukkan command Store Remove

```
Masukkan command: STORE REMOVE
Nama barang yang akan dihapus: Shrinkinator
Shrinkinator telah berhasil dihapus.
```

Gambar 30. Tampilan saat barang berhasil dihapus

```
Masukkan command: STORE REMOVE
Nama barang yang akan dihapus: Ayam goreng crisbar
Toko tidak menjual Ayam goreng crisbar.
```

Gambar 31. Tampilan saat barang gagal dihapus

6.13 Help

Pengujian ini untuk mengetahui informasi fungsi yang dapat dijalankan tertampilkan kepada pengguna. Ketika program ini dijalankan pengguna akan diperlihatkan dengan fitur - fitur serta bagaimana fitur tersebut dapat berfungsi.

```

      s e l e c t << START >> o r << LOAD [file.txt] >>
Masukkan command: HELP
===== [Selamat datang di PURRMART] =====
| Berikut adalah Command yang dapat Anda lakukan: |
| 1. START |
|   Untuk masuk sesi baru |
| 2. LOAD <filename.txt> |
|   Untuk memulai sesi berdasarkan suatu file konfigurasi |
| 3. QUIT |
|   Untuk keluar dari program |
=====

```

Gambar 32. Tampilan pengguna memasukkan command Help saat pertama kali masuk ke dalam PURRMART

```

Save file berhasil dibaca. PURRMART berhasil dijalankan.
Masukkan command: 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 33. Tampilan pengguna memasukkan command Help saat telah berhasil masuk ke dalam PURRMART

```
Anda telah login ke PURRMART sebagai dummy.
Masukkan command: 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 REQUEST BIOWEAPON
|   Untuk meminta penambahan senjata biologis
| 6. STORE SUPPLY
|   Untuk menambahkan barang dari permintaan
| 7. STORE REMOVE
|   Untuk menghapus barang
| 8. LOGOUT
|   Untuk keluar dari sesi
| 9. SAVE <filename.txt>
|   Untuk menyimpan state ke dalam file
| 10. QUIT
|   Untuk keluar dari program
=====
```

Gambar 32. Tampilan pengguna memasukkan command Help saat sudah masuk ke dalam akunnya

6.14 Save

Pengujian ini ditujukan untuk mempertunjukkan seluruh data yang dijalankan selama program dijalankan dapat disimpan. Saat program dijalankan, pengguna akan diminta untuk menyimpan pada <filename.txt>. Kemudian file ini dapat digunakan kembali oleh pengguna dengan menjalankan *command* load.

```
Masukkan command: SAVE dummyfile.txt
File konfigurasi berhasil disimpan di ../save/dummyfile.txt
```

Gambar 33. Tampilan saat pengguna memasukkan command Save

6.15 Quit

Pengujian ini bertujuan untuk melihat keberhasilan pengguna keluar dari PURRMART. Ketika pengguna menjalankan *command* ini, maka pengguna ditawarkan untuk menjalankan *command* save. Jika tidak, maka data disimpan pada config.txt.

```

Masukkan command: QUIT
Apakah kamu ingin menyimpan data sesi kali ini (Y/N)? Y
File konfigurasi berhasil disimpan di ../save/config.txt
Data sesi kali ini telah berhasil disimpan!

Anda telah keluar dari PURRMART
Sampai jumpa! Semoga datang kembali ke PURRMART!
PS C:\ITB_Thingy\C_Alstrukdat\TubesAlstrukdatK2G1\bin>

```

Gambar 34. Tampilan saat pengguna memasukkan command Quit

6.16 Bioweapon

Pengujian ini bertujuan untuk fungsi BONUS Bioweapon, dimana pengguna dapat melakukan request untuk sebuah senjata biologis yang memiliki kode rahasia ke Queue Request.

```

Masukkan command: STORE REQUEST BIOWEAPON
Masukkan nama senjata biologis: Batuksius fififafae
Masukkan sekuens DNA: TAAATGATGAGATAACCATACCGGCCGCAATT
Masukkan kode rahasia: SIGMA
Senjata biologis mengandung kode, barang akan ditambahkan ke dalam queue!
Masukkan command: STORE SUPPLY
Apakah kamu ingin menambahkan barang Batuksius fififafae(Terima/Tunda/Tolak): Terima
Harga barang: 20000
Batuksius fififafae dengan harga 20000 telah ditambahkan ke toko.

```

Gambar 35. Tampilan saat pengguna berhasil menambahkan senjata biologis

7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Start	Mengetahui keberhasilan <i>command</i> dalam membaca file	1. Masukkan perintah untuk ke dalam terminal 2. Masukkan input "START"	START	File dapat dibaca	isi file config.txt terbaca kedalam program
2	Load	Mengetahui keberhasilan Load dalam membuka dan mengakses sebuah file	1. Masukkan input berupa LOAD <filename.txt>	LOAD dummyfile.txt	File dummyfile.txt berhasil dibuka dan diakses jika ada dalam sistem dan memuat data yang tepat	isi file dummyfile.txt terbaca kedalam program
3	Login	Mengetahui apakah pengguna dapat masuk	1. Masukkan input LOGIN	LOGIN Username : admin	Pengguna dapat masuk ke dalam akunnya	user masuk kedalam program

		ke dalam sistem dengan akunnya.	2. Masukkan <i>username</i> dan <i>password</i> yang sesuai	<i>Password</i> : alstrukdatkern		menggunakan akun admin
4	Logout	Mengetahui apakah pengguna dapat keluar dari akun yang sedang login	1. Masukkan perintah LOGOUT	LOGOUT	Pengguna dapat keluar dari akunnya	user keluar dari akun yang sebelumnya dimasukan saat LOGIN
5	Register	Mengetahui apakah pengguna dapat mendaftarkan akunnya dengan ketentuan yang ada	1. Masukkan inputan REGISTER 2. Masukkan <i>username</i> yang unik beserta <i>password</i>	REGISTER <i>Username</i> : zack2301 <i>Password</i> : verysecurepassword	Pengguna dapat menambahkan akun untuk melakukan <i>login</i>	user dapat LOGIN menggunakan akun yang baru terdaftar
6	Work	Menambahkan uang pengguna dengan melakukan pekerjaan	1. Jalankan perintah WORK 2. Masukkan input nomer pekerjaan	WORK Masukkan pekerjaan yang dipilih: 5	Pengguna mendapatkan uang berdasarkan gaji yang diperoleh di tiap pekerjaannya	user menunggu selama 15 detik lalu mendapat uang berjumlah 997
7	Work Challenge	Menambahkan uang pengguna dengan melakukan <i>challenge</i>	1. Jalankan perintah WORK CHALLENGE 2. Masukkan input berupa nomer challenge	WORK CHALLENGE Masukkan challenge yang dipilih: 1 WORK CHALLENGE Masukkan challenge yang dipilih: 2	Pengguna mendapatkan uang berdasarkan bonus ketika melakukan <i>challenge</i> berupa <i>game</i>	1: user dapat memainkan challenge TebakAngka dan uangnya dikurangi sebanya biaya 2: user dapat memainkan challenge WORDL399 dan uangnya dikurangi sebanya biaya
8	Store List	Mengetahui bahwa barang - barang di dalam toko dapat dilihat	1. Jalankan perintah STORE LIST	STORE LIST	Pengguna dapat melihat data barang apa saja yang ada di dalam toko	Program akan menampilkan list Barang pada etalase <i>store</i>
9	Store Request	Mengetahui apakah pengguna dapat	1. Jalankan perintah STORE REQUEST	STORE REQUEST	Pengguna dapat meminta tambahan barang Bantal	Barang Bantal akan masuk ke Queue Request

		meminta penambahan barang baru ke dalam <i>store</i>	2. Masukkan inputan barang yang ingin di- <i>request</i>	Nama barang yang diminta: Bantal	untuk dimasukkan ke dalam toko	dari urutan belakang
10	Store Supply	Mengetahui apakah barang baru dapat ditambahkan ke dalam <i>store</i>	1. Jalankan perintah STORE SUPPLY 2. Masukkan inputan berupa Terima / Tolak / Tunda	STORE SUPPLY Apakah kamu ingin menambahkan barang Bantal: Terima STORE SUPPLY Apakah kamu ingin menambahkan barang Selimut: Tolak STORE SUPPLY Apakah kamu ingin menambahkan barang Guling: Tunda	Terima: Barang yang di- <i>request</i> dapat ditambahkan ke dalam Store List Tolak: Barang Selimut pada queue request akan dihapus Tunda: Barang yang di- <i>request</i> dikembalikan ke queue request urutan belakang	Terima: Barang Bantal sekarang dijual oleh <i>store</i> Tolak: Barang Selimut menghilang dari Queue Request Tunda: Barang Guling berada di urutan terakhir pada Queue Request
11	Store Remove	Mengetahui apakah barang yang ada di <i>store</i> dapat dihapus	1. Jalankan perintah STORE REMOVE 2. Masukkan inputan berupa nama barang yang ingin dihapus	STORE REMOVE Nama barang yang akan dihapus: Shrinkinator	Barang Shrinkinator yang ada di dalam <i>store</i> dapat dihapus	Shrinkinator menghilang pada etalase toko/ <i>store</i>
12	Help	Mengetahui apakah pengguna dapat memanfaatkan daftar fitur di <i>command Help</i>	1. Jalankan perintah HELP 2. Masukkan inputan pilihan dari <i>set command</i>	HELP	Pengguna dapat menggunakan fitur - fitur di <i>command help</i>	Ditampilkan list <i>command</i> yang dapat digunakan oleh pengguna, sesuai dengan tahap pengguna dalam program
13	Save	Mengetahui apakah data terbaru dapat tersimpan di dalam file	1. Panggil SAVE dengan memasukkan inputan SAVE <file name.txt>	SAVE dummyfile.txt	Terdapat Data baru untuk Barang pada dummyfile.txt	Ada Data baru untuk Barang pada dummyfile.txt, yaitu Bantal (dengan harga 150)

14	Quit	Mengetahui apakah pengguna dapat keluar dari sesi aplikasi	<ol style="list-style-type: none"> 1. Jalankan perintah QUIT 2. Masukkan inputan untuk melakukan SAVE (Y/N) 	QUIT	Pengguna keluar dari aplikasi	Program berhenti dan ditampilkan sebuah pesan meninggalkan program
15	Bioweapon	Mengetahui apakah pengguna dapat meminta penambahan barang spesifik senjata biologis baru ke dalam <i>store</i>	<ol style="list-style-type: none"> 1. Jalankan perintah STORE REQUEST BIOWEAPON 2. Masukkan inputan nama senjata biologis 3. Masukkan inputan sekuens DNA 4. Masukkan inputan kode rahasia 	STORE REQUEST BIOWEAPON Masukkan nama senjata biologis: Batuksius Fififafae Masukkan sekuens DNA: TAAATGAT GAGATAAC CATACCGG GCCGCAAT T Masukkan kode rahasia: SIGMA	Barang spesifik senjata biologis yang baru dapat ditambahkan	Program akan memasukkan senjata biologis Batuksius Fififafae ke dalam queue request pada urutan paling belakang.

8 Pembagian Kerja dalam Kelompok

Nama Lengkap - NIM	Deskripsi Tugas
Seren Elizabeth Siahaan - 18221160	<ul style="list-style-type: none"> - Mengerjakan kode bagian fungsi Start, Load, dan Readme - Menggabungkan fungsi-fungsi dalam file Main - Mengerjakan laporan bagian Ringkasan - Membuat ADT mesinkarakter, mesinkata, dan arraydin - Membuat driver ADT mesinkarakter dan mesinkata - Membuat banner ASCII Art

Zaka Hanif Nabalah - 18223006	<ul style="list-style-type: none"> - Mengerjakan kode bagian fungsi Work dan Work Challenge (Tebak Angka dan Word13) - Menggabungkan fungsi-fungsi dalam file Main - Membuat Makefile - Melakukan debugging pada program - Mengerjakan laporan bagian Data Test dan Test Script - Membuat driver ADT array dan arraydin
Rayhan Hidayatul Fikri - 18223022	<ul style="list-style-type: none"> - Membuat kode bagian fungsi Login, Logout, dan Register - Mengerjakan sebagian besar bagian laporan - Merapikan keseluruhan laporan
Favian Rafi Laftiyanto - 18223036	<ul style="list-style-type: none"> - Membuat kode bagian fungsi Help, Save, dan Quit - Menggabungkan fungsi-fungsi dalam file Main - Melakukan debugging pada program - Mengerjakan laporan bagian Data Test dan Test Script - Membuat driver ADT queue dan queuebarang - Melakukan testing edge case pada driver main
Wijasara Aptaluhung - 18223088	<ul style="list-style-type: none"> - Membuat kode bagian fungsi Store List, Store Request, Store Supply,

	Store Remove, Login, Logout, dan Register <ul style="list-style-type: none"> - Membuat kode bonus Bioweapon - Membuat ADT queue barang dan mesinfrasa - Membuat driver ADT mesinfrasa - Mengerjakan laporan bagian Penjelasan Tambahan Spesifikasi Tugas
--	--

9 Lampiran

9.1 Deskripsi Tugas Besar

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.

Command

Pengguna dapat memasukkan *command-command* berikut.

a. START

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

b. LOAD <filename>

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

c. LOGIN

Login merupakan *command* yang baru dapat dipanggil setelah pengguna memulai sesi. *Login* berguna untuk masuk ke akun di sistem PURRMART yang sudah didaftarkan sebelumnya. Pada fitur ini, pengguna dapat memasukkan data *username* dan *password*-nya dan kemudian akan divalidasi oleh data yang ada. Pengguna tidak akan dapat melakukan *login*, jika datanya tidak ada di sistem.

d. LOGOUT

LOGOUT merupakan salah satu *command* yang baru dapat digunakan setelah pengguna telah memasuki sebuah sesi. Pengguna baru dapat melakukan *logout* jika pengguna telah melakukan *login* terlebih dahulu.

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.** Data dari akun kemudian akan dimanfaatkan oleh sistem untuk memvalidasi data ketika pengguna menggunakan *command login*.

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

Selain Work, terdapat WORK CHALLENGE yang 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 tebakan 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 dimasukan, tetapi dengan penanda tertentu. Huruf yang benar dan berada pada tempat yang tepat dicetak biasa. Huruf yang benar, tetapi berada di tempat yang salah diberi tanda "*" setelah hurufnya. Huruf yang tidak ada sama sekali pada kata diberi tanda "%" setelah hurufnya.

h. **STORE LIST**

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

i. **STORE REQUEST**

STORE REQUEST adalah *command* yang digunakan untuk meminta penambahan barang baru ke dalam toko. Barang-barang yang diminta akan disimpan di dalam sebuah antrian dan akan dimasukkan ke toko menggunakan *command* selanjutnya. **Nama barang yang masuk tidak boleh sama dengan nama barang yang sudah ada di toko atau di antrian.**

j. **STORE SUPPLY**

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

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

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

k. **STORE REMOVE**

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

l. **HELP**

HELP merupakan *command* yang digunakan menampilkan daftar *command* yang mungkin untuk dieksekusi dengan deskripsinya.

Penjelasan dari deskripsi dibebaskan selama masih mendeskripsikan *command* sesuai dengan spek.

m. SAVE <filename>

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

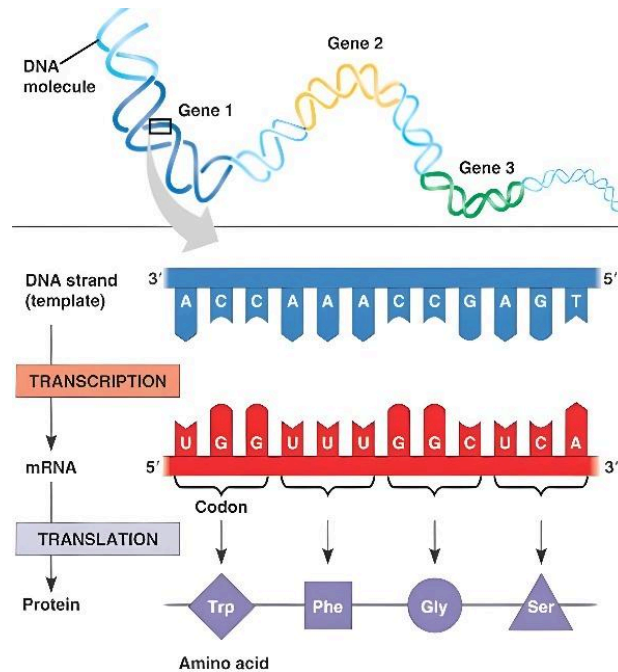
n. QUIT

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

o. Bio Weapon

Fitur tambahan yang memungkinkan pemain untuk menambahkan barang baru berupa senjata biologis ke PURRMART. Pada dasarnya, senjata biologis merupakan jenis senjata yang menggunakan patogen untuk menyerang target, mulai dari bakteri, fungi, maupun virus. Namun, apabila tidak berhati-hati, senjata biologis rentan disabotase dan justru dapat berakibat buruk bagi pemiliknya. Maka, dalam kasus ini pabrik telah memutuskan untuk **merekayasa genom patogen dan menyisipkan kode tersembunyi** di dalamnya.

Dalam biologi molekuler, dikenal istilah [central dogma](#) sebagai alur aliran informasi genetik. Informasi disimpan dalam bentuk DNA, ditranskripsi menjadi RNA, kemudian diterjemahkan menjadi protein.



Gambar 36. Ilustrasi Central Dogma

DNA (*deoxyribonucleic acid*) terdiri dari empat basa nukleotida: adenina (A), guanina (G), timina (T), dan sistein (C). RNA (*ribonucleic acid*) mirip seperti DNA, hanya saja timina (T) diganti dengan urasil (U). Pada proses transkripsi (DNA- \rightarrow RNA), Anda perlu mengganti nukleotida pada DNA dengan nukleotida komplementernya. Adenina (A) berpasangan dengan Urasil (U) sementara Guanina (G) berpasangan dengan Sistein (C).

		Second letter				
		U	C	A	G	
First letter	U	UUU } Phe UUC } UUA } Leu UUG }	UCU } UCC } Ser UCA } UCG }	UAU } Tyr UAC } UAA Stop UAG Stop	UGU } Cys UGC } UGA Stop UGG Trp	U C A G
	C	CUU } CUC } Leu CUA } CUG }	CCU } CCC } Pro CCA } CCG }	CAU } His CAC } CAA Gln CAG }	CGU } CGC } Arg CGA } CGG }	U C A G
	A	AUU } AUC } Ile AUA } AUG Met	ACU } ACC } Thr ACA } ACG }	AAU } Asn AAC } AAA Lys AAG }	AGU } Ser AGC } AGA } Arg AGG }	U C A G
	G	GUU } GUC } Val GUA } GUG }	GCU } GCC } Ala GCA } GCG }	GAU } Asp GAC } GAA Glu GAG }	GGU } GGC } Gly GGA } GGG }	U C A G

Gambar 37. Tabel Kodon-Asam Amino

Setelah DNA [ditranskripsikan](#) ke RNA, maka selanjutnya akan ditranslasi oleh ribosom dengan tabel kodon. Protein terdiri dari 20 asam amino yang tersusun sebagai rantai peptida. Perhatikan pada gambar 1.2 bahwa pembacaan RNA dilakukan secara triplet (kelipatan tiga). Satu asam amino dapat dikode oleh lebih dari satu kodon. Apabila terdapat kodon stop pada sekuens, maka lewati saja (anggap tidak ada apa-apa). Abaikan pula mekanisme kodon start.

G Glycine	Gly	P Proline	Pro
A Alanine	Ala	V Valine	Val
L Leucine	Leu	I Isoleucine	Ile
M Methionine	Met	C Cysteine	Cys
F Phenylalanine	Phe	Y Tyrosine	Tyr
W Tryptophan	Trp	H Histidine	His
K Lysine	Lys	R Arginine	Arg
Q Glutamine	Gln	N Asparagine	Asn
E Glutamic Acid	Glu	D Aspartic Acid	Asp
S Serine	Ser	T Threonine	Thr

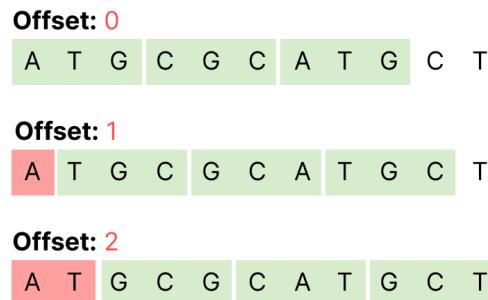
Gambar 38. Singkatan Asam Amino

Seperti yang dapat Anda lihat pada gambar 1.3, asam amino memiliki singkatan tiga huruf dan singkatan satu huruf. Singkatan asam amino

satu huruf inilah yang akan digunakan oleh pabrik untuk menyisipkan kode di senjata biologis.

Program akan meminta Anda untuk memasukkan sekuens DNA dan kode rahasia yang disepakati antara Anda dan pabrik. Apabila kode rahasia tersebut terdeteksi, maka barang dimasukkan ke *queue* seperti pada fitur STORE REQUEST. Namun, jika kode rahasia tidak ditemukan, maka barang tidak akan dimasukkan ke dalam *queue*.

Hint: Apabila Anda perhatikan pada gambar 1.4 di bawah, pembacaan RNA secara triplet memiliki konsekuensi adanya *window/frame* pembacaan. Pastikan Anda mengecek seluruh *window* pembacaan.



Gambar 39. Ilustrasi Sliding Window Pembacaan

9.2 Notulen Rapat

Asistensi I

Tanggal : Rabu, 20 November 2024	Catatan Asistensi : 1. Kalau memanggil fungsi quit(), itu save config file-nya ke mana, karena kalo pake quit tidak ada keterangan save ke mana, berbeda dengan
Tempat : Google Meet	

Kehadiran Anggota Kelompok:



1. Seren Elizabeth Siahaan
18221160



2. Zaka Hanif Nabalah
18223006



3. Rayhan Hidayatul Fikri
18223022



4. Favian Rafi Laftiyanto
18223036



5. Wijaksana Aptaluhung
18223088

fungsi save gitu? Bebas mau disimpan di config atau mau simpan di file baru, tidak di set dari kami.


2. Di wordle, kalo pakai fscanf di work challenge apakah boleh? Tidak apa-apa.

3. Berarti fscanf boleh di semua file? Boleh.

4. Laporan yang ada di drive itu untuk milestone 2 nanti atau untuk milestone ini aja? Nanti laporan untuk milestone 2 akan berlanjut dari laporan yang sekarang.

5. Untuk setiap fungsi apa harus dibuat file headernya? Apakah headernya harus dipisah-pisah apa boleh digabung saja seperti yang di readme di spesifikasi? Ya, harus dibuat header filenya. Boleh digabung, boleh dipisah, yang di spek hanyalah contoh saja.

6. Cara nge-run file-nya gimana kalau

	<p>yang di-include headernya? Misal ingin run runStack.c tapi pakai stack.c nge-include stack.h. Cara runnya:</p> <p>gcc stack.c runStack.c -o test, ./test</p>
	<p>Tanda Tangan Asisten:</p>  <p>Aulia Nadhirah Yasmin B.</p>

9.3 Log Activity Anggota Kelompok

Tanggal	Anggota	Aktivitas
20 November 2024	18223006 Zaka Hanif Nabalah	Menambahkan <i>command</i> Work dan Work Challenge
20 November 2024	18223036 Favian Rafi Laftiyanto	Menambahkan <i>command</i> Save, Quit, dan Help
22 November 2024	18221160 Seren Elizabeth Siahaan	Menambahkan <i>command</i> Start, Load dan config.txt
23 November 2024	18221160 Seren Elizabeth Siahaan	Memperbaiki <i>command</i> Start, Load, dan driver
23 November 2024	18221160 Seren Elizabeth Siahaan	Menambahkan file dummyfile.txt
23 November 2024	18221160 Seren Elizabeth	Push ADT pembacaan

	Siahaan	stdin, wordtoint, dan compare
23 November 2024	18221160 Seren Elizabeth Siahaan	Menambahkan <i>command</i> Load kondisi file invalid
23 November 2024	18223006 Zaka Hanif Nabalah	Melakukan perubahan pada fungsi CopyWord di ADT
23 November 2024	18223006 Zaka Hanif Nabalah	Memperbaiki metode <i>input</i> untuk <i>command</i> Work dan Work Challenge
23 November 2024	18223022 Rayhan Hidayatul Fikri	Menambahkan <i>command</i> Login, Logout, dan Register. Menambahkan file Users.txt beserta ADT pendukungnya.
24 November 2024	18223006 Zaka Hanif Nabalah	Menambahkan <i>null terminator</i> pada fungsi Word di ADT
24 November 2024	18223006 Zaka Hanif Nabalah	Menambahkan validasi input pada <i>command</i> wordI3
24 November 2024	18223006 Zaka Hanif Nabalah	Melakukan perubahan <i>word list</i> pada <i>command</i> WordI3
24 November 2024	18223088 Wijaksana Aptaluhung	Menambahkan ADT revisi dari mesinkarakter dan mesinkata
24 November 2024	18223088 Wijaksana	Menambahkan <i>command</i>

	Aptaluhung	store, ADT queue dan queue barang, serta bonus bioweapon
24 November 2024	18223036 Favian Rafi Laftiyanto	Melakukan perubahan pada nama variabel di <i>command</i> Save, Quit, dan Help
24 November 2024	18223006 Zaka Hanif Nabalah	Memindahkan semua ADT pada ADT folder di GitHub
24 November 2024	18223036 Favian Rafi Laftiyanto	Menambahkan versi pertama dari <i>command</i> Main.c serta melakukan fiksasi pada hal - hal kecil
24 November 2024	18223006 Zaka Hanif Nabalah	Menambahkan fungsi makefile pada ADT dan melakukan fiksasi pada komponen di folder
24 November 2024	18221160 Seren Elizabeth Siahaan	Melakukan perubahan untuk fungsi <i>compareWordToString</i> di ADT
24 November 2024	18223006 Zaka Hanif Nabalah	Melakukan fiksasi untuk fungsi <i>compareWordToString</i> di ADT
24 November 2024	18223088 Wijaksana Aptaluhung	Melakukan push untuk perubahan pada <i>command</i> bioweapon

24 November 2024	18223088 Wijaksana Aptaluhung	Melakukan push revisi untuk ADT mesinkata
24 November 2024	18223006 Zaka Hanif Nabalah	Melakukan perubahan pada <i>command</i> Main.c
24 November 2024	18223088 Wijaksana Aptaluhung	Menambahkan ADT mesinfrasa
24 November 2024	18221160 Seren Elizabeth Siahaan	Menambahkan header dummylogin pada <i>command</i> login
24 November 2024	18223088 Wijaksana Aptaluhung	Melakukan fiksasi dari ADT
24 November 2024	18223088 Wijaksana Aptaluhung	Push revisi dari <i>command</i> storeremove dan storerequest
24 November 2024	18223006 Zaka Hanif Nabalah	Melakukan perbaikan pada <i>command</i> Main dan ADT mesinfrasa
24 November 2024	18223088 Wijaksana Aptaluhung	Melakukan perbaikan pada ADT queuebarang
24 November 2024	18223088 Wijaksana Aptaluhung	Melakukan perbaikan dan push pada <i>command</i> bioweapon
24 November 2024	18223006 Zaka Hanif Nabalah	Melakukan perubahan pada fungsi makefile di ADT dan <i>command</i> main
25 November 2024	18223036 Favian Rafi	Menambahkan menu Help pada <i>command</i>

	Laftiyanto	Bioweapon dan menambahkan <i>newlines</i> pada Help
25 November 2024	18223036 Favian Rafi Laftiyanto	Menambahkan detail <filename.txt> pada <i>command</i> Save dan Load
25 November 2024	18223088 Wijaksana Aptaluhung	Memperbaiki dan <i>push command</i> Login, Logout, dan Register
25 November 2024	18223036 Favian Rafi Laftiyanto	Memindahkan convig.txt, dummyfile.txt, invalid.txt ke folder bernama Save
25 November 2024	18223006 Zaka Hanif Nabalah	Memperbaiki format <i>filename</i> pada fungsi Save
25 November 2024	18223088 Wijaksana Aptaluhung	Melakukan perubahan pada <i>command</i> Start, Load, dan makefile
25 November 2024	18223006 Zaka Hanif Nabalah	Melakukan revisi pada ADT meisnkata dan mesinkarakter
25 November 2024	18223006 Zaka Hanif Nabalah	Melakukan perbaikan pada input validasi pada <i>command</i> WORDL3
25 November 2024	18223006 Zaka Hanif Nabalah	Melakukan perubahan kecil pada Work dan Work Challenge
25 November 2024	18223006 Zaka Hanif	Memindahkan

	Nabalah	queuebarang ke ADT folder
25 November 2024	18223088 Wijaksana Aptaluhung	Melakukan perubahan dan push ulang pada <i>command</i> Store Supply
25 November 2024	18223088 Wijaksana Aptaluhung	Melakukan perubahan pada Bioweapon
25 November 2024	18223036 Favian Rafi Laftiyanto	Fiksasi <i>command</i> Save dan Start
25 November 2024	18223006 Zaka Hanif Nabalah	Melakukan konektivitas antara Work dan Work Challenge ke <i>command</i> User
25 November 2024	18223036 Favian Rafi Laftiyanto	Menambahkan <i>banner</i> ASCII <i>art</i> dan fiksasi SAVE
25 November 2024	18223006 Zaka Hanif Nabalah	Melakukan fiksasi di <i>parameter</i> main
25 November 2024	18223006 Zaka Hanif Nabalah	Menghapus <i>leftover</i> debug printf
25 November 2024	18221160 Seren Elizabeth Siahaan	Perubahan pada file readme, banner.txt, dan menghapus dummylogin
25 November 2024	18223088 Wijaksana Aptaluhung	Melakukan revisi Store Supply

25 November 2024	18223036 Favian Rafi Laftiyanto	Menambahkan / <i>push</i> queuedriver
25 November 2024	18223006 Zaka Hanif Nabalah	Menambahkan / <i>push</i> queuebarangdriver
25 November 2024	18221160 Seren Elizabeth Siahaan	Menambahkan driver Mesin Karakter
25 November 2024	18223036 Favian Rafi Laftiyanto	Menambakan komentar pada Work dan Work Challenge
25 November 2024	18223088 Wijaksana Aptaluhung	Menambahkan driver Mesin Frasa
25 November 2024	18221160 Seren Elizabeth Siahaan	Menambahkan driver Mesin Kata