

LAPORAN TUGAS BESAR 2

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 - Institut Teknologi Bandung

Jl. Ganeshha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen	Halaman
		<i>IF2111-TB-K2-03</i>	<i><jml hlm></i>
		Revisi	0
			<i>20 Desember 2024</i>

Daftar Isi

1	Ringkasan	3
2	Penjelasan Tambahan Spesifikasi Tugas	4
2.1	Riwayat Maksimal	4
3	Struktur Data (ADT)	5
3.1	Struktur Data User	5
3.2	Struktur Data Barang	6
3.3	Struktur Data Cart	7
3.4	Struktur Data History	7
3.5	Struktur Data Wishlist	8
4	Program Utama	8
5	Algoritma-Algoritma Menarik	10
5.1	Wishlist remove	10
5.2	Cart Pay	11
6	Data Test	11
6.1	Data Test Profile	12
6.2	Data Test Cart Add <nama> <n>	12
6.3	Data Test Cart Remove <nama> <n>	12
6.4	Data Test Cart Show	12
6.5	Data Test Cart Pay	12
6.6	Data Test History <n>	12
6.7	Data Test Wishlist Add	12
6.8	Data Test Wishlist Swap <i> <j>	12
6.9	Data Test Wishlist Remove	12
6.10	Data Test Wishlist Clear	12
6.11	Data Test Wishlist Show	13
6.12	Data Test Start	13
6.13	Data Test Load Save	13
6.14	Data Test Store List	13
7	Test Script	13
8	Pembagian Kerja dalam Kelompok	16
9	Lampiran	16
9.1	Deskripsi Tugas Besar	16
System Mechanic		17
1.	About the System	17
2.	Menu Program	17
3.	Command	17

a. PROFILE	17
b. CART ADD <nama> <n>	17
c. CART REMOVE <nama> <n>	18
d. CART SHOW	18
e. CART PAY	18
f. HISTORY <n>	18
g. WISHLIST ADD	18
h. WISHLIST SWAP <i> <j>	18
i. WISHLIST REMOVE	19
j. WISHLIST CLEAR	19
k. WISHLIST SHOW	19
4. Perubahan Command	19
a. START, LOAD, dan SAVE	19
b. STORE LIST	19
9.2 Notulen Rapat	19
9.3 Log Activity Anggota Kelompok	23

1 Ringkasan

PURRMART adalah sebuah sistem jual beli yang diperlukan oleh Agen Purry dan 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 perperangan 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 mengelola toko, keranjang belanja, dan *wishlist* pengguna. Selain itu, PURRMART juga dilengkapi dengan beberapa fitur menarik, seperti fitur *cart*, yang memungkinkan pengguna menambahkan, menghapus, melihat, dan membayar barang di keranjang. Adanya fitur *wishlist* yang menyediakan kemampuan untuk menyimpan daftar barang yang diinginkan, menghapus barang, menukar posisi barang, atau mengosongkan daftar tersebut. Fitur lain yaitu *history*, yang mencatat dan menampilkan riwayat pembelian pengguna berdasarkan urutan waktu, sehingga mempermudah pelacakan transaksi.

Program ini merupakan lanjutan dari program yang kami buat di Tugas Besar 1, dengan adanya pengembangan lebih lanjut untuk mendukung fitur-fitur baru dan meningkatkan efisiensi bagi pengguna yang menggunakan. Pengembangan dilakukan dengan melakukan beberapa perubahan perintah yang digunakan di PURRMART, seperti *start*, *load*, *save* serta *store list*. Program ini dibuat menggunakan beberapa ADT yang terdapat dalam pembelajaran mata kuliah Algoritma dan Struktur Data, dengan dasar bahasa pemrograman C, seperti ADT *User*, *Stack*, *Map*, dan *LinkedList*.

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.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Riwayat Maksimal

Fitur Riwayat Maksimal dalam PURRMART merupakan perluasan dari fungsi riwayat pembelian (*history*) dalam sistem. Fitur ini bertujuan untuk memberikan detail lebih lengkap tentang transaksi pengguna. Sebelumnya, ADT Stack digunakan untuk menyimpan riwayat pembelian hanya didasarkan pada nama barang dengan total harga terbesar. Dalam Riwayat Maksimal ini, seluruh detail barang yang dibeli dalam setiap transaksi akan disimpan.

STEI- ITB	IF2111-TB2-K2-03	Halaman 4 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

Command HISTORY akan ditingkatkan untuk menampilkan daftar riwayat pembelian pengguna secara rinci, termasuk total barang dan harga untuk setiap transaksi. Pengguna dapat meminta sejumlah riwayat transaksi terbaru dengan memberikan angka N, di mana program akan menampilkan hingga N riwayat transaksi terbaru atau semua transaksi jika jumlah riwayat lebih sedikit dari N. Setiap riwayat transaksi mencakup total harga, jumlah barang yang dibeli, serta daftar barang dengan kuantitas dan total harga masing-masing.

Untuk mendukung fitur ini, format file konfigurasi juga diperbarui agar detail riwayat pembelian dapat disimpan secara konsisten. Dalam format baru, setiap elemen riwayat dimulai dengan dua bilangan, yaitu L yang menunjukkan jumlah barang yang dibeli, dan X yang menunjukkan total harga transaksi tersebut. Baris-baris berikutnya menyimpan rincian setiap barang yang dibeli, dalam format <Total Biaya> <Kuantitas Barang> <Nama Barang>. Konfigurasi ini juga mencakup informasi, seperti nama pengguna, jumlah riwayat, dan detail transaksi sehingga memungkinkan sistem untuk memuat riwayat secara lengkap saat program dijalankan.

Dengan adanya fitur ini, pengguna akan mendapatkan transparansi penuh atas transaksi yang dilakukan, memenuhi kebutuhan audit OWCA, sekaligus meningkatkan pengalaman pengguna dengan akses yang lebih mudah dan detail terhadap riwayat pembelian mereka. Format file konfigurasi baru juga memastikan bahwa informasi ini disimpan dan dikelola secara efisien dalam sistem.

3 Struktur Data (ADT)

Dalam membuat program PURRMART lanjutan ini, kami menggunakan beberapa ADT baru, yaitu ADT User, ADT Barang, ADT Cart, ADT History, dan ADT Wishlist. Selain ADT baru, ADT yang telah dibuat pada Tugas Besar 1 (Mesin Kata serta Mesin Karakter) juga tetap digunakan dalam program lanjutan ini dengan beberapa penambahan di dalam fungsinya.

3.1 Struktur Data User

Pada penggeraan program PURRMART, kami menggunakan struktur data Pengguna (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 **reduce_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. ADT User ini diimplementasikan dalam file **user.c** untuk mendukung keberlangsungan dari aplikasi PURRMART.

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. **add_money** – menambah saldo setelah pengguna mendapatkan penghasilan dari pekerjaan.
5. **reduce_money** – mengurangi saldo ketika pengguna melakukan transaksi seperti membayar barang dalam keranjang atau mengikuti tantangan berbayar.

3.2 Struktur Data Barang

Struktur data kedua yang digunakan untuk menyelesaikan persoalan di PURRMART adalah Barang. ADT Barang dirancang untuk mengelola data barang yang tersedia dalam aplikasi, termasuk nama dan harga barang. Dalam pengelolaan barang, terdapat dua jenis struktur data utama: **ListBarang** untuk menyimpan daftar barang yang ada di toko dan **QueueBarang** untuk mengelola permintaan barang secara antrian. ListBarang menggunakan array dinamis yang memungkinkan ukuran array bertambah secara otomatis ketika kapasitas penuh, melalui fungsi seperti **resize_list_barang**. Fungsi seperti **add_barang** menambahkan barang baru ke dalam daftar, **barang_count** menghitung jumlah barang saat ini, dan **is_barang_exist** memeriksa keberadaan barang dalam daftar. Selain itu, fungsi **delete_barang** berfungsi untuk menghapus barang berdasarkan nama, sementara **harga_barang** memberikan akses ke harga barang tertentu.

QueueBarang, di sisi lain, digunakan untuk mengelola permintaan barang baru yang diimplementasikan menggunakan array statis. Fungsi seperti **enqueue_barang** digunakan untuk menambahkan barang ke antrian, **dequeue_barang** untuk menghapus barang dari antrian, dan **is_barang_exist_queue** untuk memeriksa apakah suatu barang sudah ada dalam antrian. Fungsi tambahan seperti **print_list_barang** dan **print_queue_barang** mempermudah menampilkan data barang di toko maupun permintaan yang sedang menunggu.

Pemilihan struktur data ini didasarkan pada kebutuhan untuk efisiensi tentunya. ListBarang menggunakan array dinamis untuk memungkinkan pengelolaan daftar barang yang dapat bertambah seiring waktu tanpa pembatasan kapasitas tetap. QueueBarang dipilih untuk mengelola permintaan barang secara berurutan (FIFO) dengan efisiensi tinggi pada operasi penambahan dan penghapusan elemen. ADT Barang ini diimplementasikan dalam file **barang.h**.

STEI- ITB	IF2111-TB2-K2-03	Halaman 6 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

dan **barang.c**, mendukung fitur-fitur penting seperti penambahan, penghapusan, pencarian, dan pencetakan data barang di PURRMART.

3.3 Struktur Data Cart

Pada program PURRMART, ADT Cart dirancang untuk merepresentasikan keranjang belanja dalam sistem PURRMART. Struktur data ini terdiri dari dua elemen utama yaitu **CartElement**, yang berfungsi untuk merepresentasikan setiap item dalam keranjang dengan atribut name (nama barang) dan amount (jumlah barang), serta **Cart**, yang merupakan array statis dari CartElement untuk menyimpan daftar barang yang dimasukkan ke keranjang. Selain itu, atribut count digunakan untuk mencatat jumlah total barang yang ada di dalam keranjang. Struktur data Cart ini memiliki kesamaan dengan ADT Map dari segi prinsip *key-value relationship*, dimana nama barang (name) dapat dianggap sebagai *key*, dan jumlah barang (amount) sebagai value.

Struktur data Cart ini dirancang untuk mengelola keranjang belanja, seperti menyimpan barang dengan fungsi **insert_cart**, menghapus barang dengan **delete_cart**, dan memvalidasi keberadaan barang menggunakan **is_exist_in_cart** atau **cart_value**. Program juga dilengkapi fungsi validasi kapasitas keranjang, seperti **is_cart_empty** untuk memeriksa apakah keranjang kosong, dan **is_cart_full** untuk memastikan keranjang tidak melebihi kapasitas maksimum (MAX_LIST). Fungsi **create_cart** berfungsi untuk menginisialisasi struktur data Cart, yaitu keranjang belanja. Fungsi ini mengatur atribut count pada struktur Cart ke 0, yang menandakan bahwa keranjang masih kosong dan belum ada barang di dalamnya.

ADT Cart ini diimplementasikan menggunakan array statis karena sifat keranjang belanja yang biasanya memiliki kapasitas tetap. Pemilihan array statis memungkinkan manipulasi elemen dengan efisien, seperti pencarian, penambahan, dan penghapusan elemen. Struktur ini diimplementasikan dalam file **cart.h** sebagai header dan **cart.c** untuk definisi fungsinya.

3.4 Struktur Data History

Dalam program PURRMART, ADT History dirancang untuk mencatat riwayat transaksi dalam sistem PURRMART menggunakan struktur data berbasis stack. Struktur History terdiri dari dua elemen utama, yaitu **HistoryElement**, yang merepresentasikan suatu transaksi dengan atribut **HistoryBarangs** (daftar barang yang dibeli), **HistoryBarcodeCount** (jumlah barang dalam transaksi), dan **TotalPrice** (total harga transaksi), serta **History**, yang merupakan stack berisi elemen-elemen HistoryElement. Stack ini menggunakan atribut **IndexTop** untuk menunjuk elemen teratas dalam riwayat transaksi.

Persoalan utama yang diselesaikan oleh ADT History adalah pencatatan transaksi secara terurut berdasarkan waktu. Fungsi **create_history** berfungsi untuk membuat stack kosong yang akan menyimpan riwayat transaksi pengguna. Fungsi **add_history_element** digunakan untuk menambahkan detail barang beserta jumlahnya ke dalam elemen transaksi. Fungsi **push_history** digunakan untuk menambahkan transaksi baru ke puncak stack, sementara fungsi **pop_history**

memungkinkan penghapusan atau pengambilan elemen transaksi terbaru dari stack. Dengan pendekatan berbasis stack, transaksi terakhir selalu berada di posisi teratas, sehingga mudah diakses atau ditampilkan sebagai riwayat terbaru.

Struktur berbasis stack dipilih karena sifatnya yang mendukung prinsip LIFO (*Last In, First Out*), yang cocok untuk pengelolaan riwayat, di mana transaksi terbaru diprioritaskan untuk ditampilkan atau diproses. Implementasi ini ditempatkan dalam file **history.h** sebagai header dan **history.c** untuk definisi fungsi. ADT ini menjadi komponen penting dalam mendukung fitur seperti HISTORY, dan memungkinkan pengguna untuk melihat transaksi terbaru hingga terlama secara efisien.

3.5 Struktur Data Wishlist

Struktur data terakhir yang digunakan dalam program PURRMART adalah struktur data Wishlist. ADT Wishlist digunakan untuk mengelola daftar barang yang diinginkan oleh pengguna dalam sistem PURRMART. Struktur data ini diimplementasikan menggunakan *linked list* yang terdiri dari elemen-elemen **WishlistElement**, di mana setiap elemen menyimpan informasi tentang nama barang (Nama) dan alamat elemen berikutnya (Next). Struktur utama, **Wishlist**, memiliki pointer First yang menunjuk ke elemen pertama dalam daftar, atau NULL jika daftar kosong.

Persoalan utama yang diselesaikan oleh ADT ini adalah pengelolaan dinamis dari daftar barang yang diinginkan pengguna. Fungsi seperti **insert_wishlist** berfungsi untuk menambahkan barang baru di akhir daftar, sedangkan **delete_wishlist** digunakan untuk menghapus barang tertentu berdasarkan nama. Fungsi **swap_wishlist** memungkinkan pengguna menukar posisi dua barang dalam daftar, sedangkan **search_wishlist** membantu mencari barang tertentu. Fungsi-fungsi ini memastikan daftar dapat dikelola secara fleksibel, termasuk mencetak seluruh isi wishlist dengan **print_wishlist** atau menghitung jumlah barang dengan **count_wishlist**.

Pemilihan struktur *linked list* sebagai ADT dikarenakan perlu untuk mengelola data secara dinamis. *Linked list* memungkinkan penambahan dan penghapusan elemen tanpa perlu menggeser elemen lainnya. Hal ini sangat efisien untuk operasi-operasi di *wishlist*, seperti menambah barang di akhir atau menghapus barang dari daftar. Selain itu, *linked list* memungkinkan ukuran daftar bertambah sesuai kebutuhan pengguna.

ADT Wishlist diimplementasikan dalam file **wishlist.h** sebagai header dan **wishlist.c** untuk definisi fungsi. Pendekatan berbasis linked list menjadikan ADT ini fleksibel dan efisien, mendukung fitur-fitur seperti WISHLIST ADD, WISHLIST REMOVE, WISHLIST SWAP, dan WISHLIST PRINT dalam sistem PURRMART.

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

STEI- ITB	IF2111-TB2-K2-03	Halaman 8 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

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. Selain fitur yang ada di Tugas Besar 1, fitur-fitur tambahan yang dapat diakses adalah sebagai berikut.

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. Mengelola keranjang belanja → **CART**
 - a. **CART ADD** → pengguna dapat menambahkan barang dengan kuantitas tertentu ke dalam keranjang belanja. Apabila barang tidak tersedia di toko, akan muncul pesan bahwa barang tidak tersedia.
 - b. **CART REMOVE** → pengguna dapat mengurangi jumlah barang tertentu di keranjang. Apabila jumlah yang diminta untuk dihapus melebihi yang ada di keranjang, maka proses akan gagal.
 - c. **CART SHOW** → program menampilkan isi keranjang pengguna, termasuk jumlah barang, total harga masing-masing barang, dan total keseluruhan harga keranjang. Apabila keranjang kosong, akan ditampilkan pesan keranjang kosong.
 - d. **CART PAY** → pengguna dapat membayar semua barang di keranjang. Program akan memastikan saldo pengguna mencukupi sebelum transaksi dilakukan. Jika sukses, barang akan dihapus dari keranjang dan saldo pengguna akan berkurang. Riwayat pembelian (*history*) akan diperbarui sesuai barang dengan harga total terbesar.
3. Mengelola wishlist → **WISHLIST**
 - a. **WISHLIST ADD** → pengguna dapat menambahkan barang ke daftar wishlist jika barang tersedia di toko. Apabila barang sudah ada di wishlist, program akan memberi pesan.
 - b. **WISHLIST REMOVE** → pengguna dapat menghapus suatu barang berdasarkan posisi barang yang ada di wishlist dan nama barang yang ada di wishlist.
 - c. **WISHLIST SWAP** → pengguna dapat menukar posisi dua barang di wishlist berdasarkan urutan barang pada wishlist.

- d. **WISHLIST CLEAR** → pengguna dapat mengosongkan seluruh isi barang dalam wishlist.
- e. **WISHLIST SHOW** → program menampilkan barang-barang yang sudah dimasukkan oleh pengguna ke wishlist.

Setelah fitur manajemen barang (cart dan wishlist), PURRMART juga memiliki fitur **HISTORY** yang berfungsi untuk menunjukkan riwayat pembelian pengguna di PURRMART berdasarkan jumlah pembelian (misalnya, HISTORY 5 yang berarti 5 riwayat terbaru). Riwayat diurutkan dari transaksi terbaru ke transaksi terlama. Jika pengguna belum melakukan pembelian, akan muncul pesan bahwa riwayat pembelian kosong. Selain itu, program juga dapat menampilkan informasi data diri pengguna (nama beserta saldo pengguna) yang hanya bisa diakses ketika pengguna sudah login ke program.

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 program 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 *Wishlist remove*

Algoritma **remove_wishlist()** merupakan metode penghapusan barang dari wishlist pengguna yang menawarkan dua fitur secara langsung. Pengguna dapat melakukan penghapusan berdasarkan nama barang atau berdasarkan posisi barang dalam daftar. Algoritma ini dimulai dengan validasi apakah wishlist pengguna kosong menggunakan fungsi **is_wishlist_empty()**. Jika wishlist kosong, notifikasi akan diberikan dan algoritma berhenti. Sebaliknya, jika pengguna ingin menghapus barang berdasarkan nama, algoritma akan meminta input nama barang lalu memvalidasinya menggunakan fungsi **search_wishlist()**. Setelah itu, algoritma akan menghapus barang tersebut dengan **delete_wishlist()** jika barang ditemukan. Apabila nama barang tidak ditemukan, pesan kesalahan ditampilkan kepada pengguna.

Selain penghapusan berdasarkan nama, algoritma ini juga mendukung penghapusan berdasarkan posisi barang dalam wishlist. Input berupa angka diolah menjadi integer untuk menentukan posisi barang dan algoritma akan memvalidasi apakah posisi tersebut valid berdasarkan jumlah item dalam wishlist. Jika valid, algoritma menggunakan traversal linked list

untuk mencapai node pada posisi yang ditentukan dan menghapus barang tersebut dengan fungsi **delete_wishlist()**. Namun, jika input berada di luar rentang atau tidak valid maka akan muncul notifikasi kesalahan.

Algoritma ini menarik karena menggabungkan fleksibilitas dan efisiensi dalam penghapusan data di linked list. Validasi input yang ketat menjamin integritas data dan mengurangi risiko kesalahan selama eksekusi. Dengan kompleksitas waktu $O(n)$ untuk traversal, algoritma tetap efisien bahkan untuk daftar dengan banyak barang. Kombinasi fitur penghapusan berdasarkan nama dan posisi memberikan pengalaman pengguna yang lebih baik, membuat algoritma ini menjadi salah satu solusi menarik dalam pengelolaan data wishlist.

5.2 Cart Pay

Algoritma **cart_pay()** adalah bagian penting dalam sistem PURRMART, yang bertujuan untuk menyelesaikan proses checkout dengan cara yang sistematis. Proses dimulai dengan menghitung total harga barang dalam keranjang, yang dilakukan oleh fungsi **totalharga**. Fungsi ini menjumlahkan biaya berdasarkan jumlah dan harga satuan masing-masing barang. Setelah itu, sistem memverifikasi saldo pengguna, dan jika saldo mencukupi, transaksi diteruskan dengan mengurangi saldo sesuai total harga dan mengosongkan keranjang. Selain itu, data barang yang telah dibeli dipindahkan ke dalam riwayat transaksi pengguna menggunakan fungsi **add_cart_to_history**.

Algoritma ini juga memberikan pengguna gambaran ringkas tentang belanja mereka, termasuk daftar barang yang akan dibeli, jumlahnya, serta total biaya. Dengan informasi ini, pengguna dapat memutuskan apakah mereka ingin melanjutkan transaksi atau membatalkannya. Jika saldo tidak mencukupi atau keranjang kosong, sistem akan menampilkan pesan kegagalan. Yang menarik dari algoritma ini adalah integrasi berbagai Struktur Data Abstrak (ADT), seperti **ADT Cart** untuk pengelolaan keranjang, **ADT User** untuk pengecekan saldo dan pembaruan, serta **ADT History** untuk mencatat transaksi yang berhasil.

Secara keseluruhan, algoritma ini mengadopsi konsep yang sangat relevan dengan skenario dunia nyata, terutama dalam konteks *e-commerce*. Dengan demikian, **cart_pay()** tidak hanya penting dalam memastikan konsistensi data dan kelancaran transaksi, tetapi juga memberikan pengalaman pengguna yang intuitif dan mudah dipahami.

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,

STEI- ITB	IF2111-TB2-K2-03	Halaman 11 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

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

6.1 Data Test Profile

Tes ini dilakukan untuk mengecek apakah pengguna bisa melihat data dirinya saat status pengguna telah log in.

```
Ketik command atau "Help"  
>>> profile  
Nama : jenni  
Saldo : 0  
  
Ketik command atau "Help"  
>>> █
```

Gambar 6.1.1 Tampilan profile pengguna sebelum work dijalankan

```
Ketik command atau "Help"  
>>> 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 6.1.2 Tampilan ketika work dijalankan

```
Ketik command atau "Help"  
>>> profile  
Nama : jenni  
Saldo : 10000  
  
Ketik command atau "Help"  
>>> █
```

Gambar 6.1.3 Tampilan profile pengguna setelah work dijalankan

6.2 Data Test Cart Add <nama> <n>

Tes ini dilakukan untuk mengecek keberjalanannya fungsi CART ADD dimana pengguna bisa menambahkan barang dengan kuantitas tertentu ke dalam keranjang belanja.

STEI- ITB	IF2111-TB2-K2-03	Halaman 12 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

```
Ketik command atau "Help"  
>>> cart add AK47 2  
Berhasil menambahkan 2 AK47 ke keranjang belanja!
```

Gambar 6.2.1 Tampilan fungsi CART ADD ketika berhasil dijalankan

6.3 Data Test Cart Remove <nama> <n>

Tes ini dilakukan untuk mengecek keberjalanannya fungsi CART REMOVE dimana pengguna bisa mengurangi barang sejumlah kuantitas tertentu dari keranjang belanja.

6.4 Data Test Cart Show

Tes ini dilakukan untuk mengecek keberjalanannya fungsi CART SHOW dimana digunakan untuk menunjukkan barang-barang yang sudah dimasukkan ke dalam keranjang.

```
Ketik command atau "Help"  
>>> cart show  
Keranjang belanja kosong!
```

Gambar 6.4.1. Tampilan ketika cart kosong

```
Ketik command atau "Help"  
>>> cart add AK47 2  
Berhasil menambahkan 2 AK47 ke keranjang belanja!  
  
Ketik command atau "Help"  
>>> cart add Meong 6  
Berhasil menambahkan 6 MEONG ke keranjang belanja!  
  
Ketik command atau "Help"  
>>> cart show  
Berikut adalah isi keranjangmu.  
  
Kuantitas      Nama      Total  
2              AK47      20  
6              Meong     3000
```

Gambar 6.4.2. Tampilan ketika fungsi CART SHOW berhasil dijalankan

6.5 Data Test Cart Pay

Tes ini dilakukan untuk mengecek keberjalanannya fungsi CART PAY dimana digunakan untuk membeli barang-barang yang sudah dimasukkan ke dalam keranjang.

```
>>> cart pay
Kamu akan membeli barang-barang berikut.

Kuantitas      Nama      Total
2                  AK47     20
Total biaya yang harus dikeluarkan adalah 20, apakah jadi dibeli? (Ya/Tidak): ya
Keranjang kosong.
Ketik command atau "Help"
```

Gambar 6.5.1. Tampilan ketika fungsi CART PAY berhasil dijalankan

```
Kuantitas      Nama      Total
3                  AK47     30
5                  Lalabu   100
Total biaya yang harus dikeluarkan adalah 130, apakah jadi dibeli? (Ya/Tidak):
```

Gambar 6.5.2. Tampilan ketika fungsi CART PAY berhasil dijalankan

6.6 Data Test History <n>

Tes ini dilakukan untuk mengecek keberjalanannya fungsi HISTORY dimana digunakan untuk menunjukkan riwayat pembelian seorang pengguna.

```
Ketik command atau "Help"
>>> history
Riwayat pembelian barang:
```

Gambar 6.6.1. Tampilan ketika HISTORY berhasil menambahkan barang ke wishlist

6.7 Data Test Wishlist Add

Tes ini dilakukan untuk mengecek keberjalanannya fungsi WISHLIST ADD dimana digunakan untuk menambahkan suatu barang ke wishlist.

```
Ketik command atau "Help"
>>> wishlist add
Masukkan nama barang yang ingin ditambahkan ke wishlist: AK47
AK47 berhasil ditambahkan ke wishlist!
```

Gambar 6.7.1. Tampilan ketika berhasil menambahkan barang ke wishlist

```
Ketik command atau "Help"
>>> wishlist add
Masukkan nama barang yang ingin ditambahkan ke wishlist: bunga & kumbang
Tidak ada barang dengan nama bunga & kumbang di toko!
```

Gambar 6.7.2 Tampilan ketika barang yang ingin ditambahkan tidak tersedia di toko

6.8 Data Test Wishlist Swap <i><j>

Tes ini dilakukan untuk mengecek keberjalanannya fungsi WISHLIST SWAP dimana digunakan untuk menukar barang posisi ke-i dengan barang posisi ke-j pada wishlist.

```
Ketik command atau "Help"
>>> wishlist show
Berikut adalah isi wishlist-mu:
1. Ayam Goreng Crisbar
2. AK47
3. Lalabu

Ketik command atau "Help"
>>> wishlist swap 2 3
Barang Lalabu dan AK47 berhasil ditukar posisinya di wishlist!

Ketik command atau "Help"
>>> wishlist show
Berikut adalah isi wishlist-mu:
1. Ayam Goreng Crisbar
2. Lalabu
3. AK47
```

Gambar 6.8.1 Tampilan ketika WISHLIST SWAP berhasil dijalankan

6.9 Data Test Wishlist Remove

Tes ini dilakukan untuk mengecek keberjalanannya fungsi WISHLIST REMOVE dimana digunakan untuk menghapus barang dengan posisi ke-i dari wishlist dan menghapus barang dari wishlist berdasarkan nama barang yang dimasukkan pengguna.

```
Ketik command atau "Help"
>>> wishlist remove 3
AK47 berhasil dihapus dari wishlist!
```

Gambar 6.9.1 Tampilan ketika WISHLIST REMOVE berdasarkan posisi berhasil dijalankan (sebelum)

```
Ketik command atau "Help"
>>> wishlist show
Berikut adalah isi wishlist-mu:
1. Ayam Goreng Crisbar
2. Lalabu
```

Gambar 6.9.2 Tampilan ketika isi wishlist setelah WISHLIST REMOVE dijalankan

6.10 Data Test Wishlist Clear

Tes ini dilakukan untuk mengecek keberjalanannya fungsi WISHLIST CLEAR dimana digunakan untuk menghapus semua barang yang terdapat di dalam WISHLIST.

STEI- ITB	IF2111-TB2-K2-03	Halaman 15 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

```
Ketik command atau "Help"
>>> wishlist clear
Wishlist berhasil dihapus!
```

Gambar 6.10.1 Tampilan ketika WISHLIST CLEAR berhasil dijalankan

```
Ketik command atau "Help"
>>> wishlist show
Wishlist kosong!
```

Gambar 6.10.2 Tampilan isi wishlist setelah WISHLIST CLEAR dijalankan

6.11 Data Test Wishlist Show

Tes ini dilakukan untuk mengecek keberjalanannya fungsi WISHLIST SHOW dimana digunakan untuk menunjukkan barang-barang yang sudah dimasukkan ke dalam wishlist.

```
Ketik command atau "Help"
>>> wishlist show
Berikut adalah isi wishlist-mu:
1. Ayam Goreng Crisbar
2. AK47
```

Gambar 6.11.1. Tampilan ketika Wishlist Show berhasil menambahkan barang ke wishlist

```
Ketik command atau "Help"
>>> wishlist show
Wishlist kosong!
```

Gambar 6.11.2. Tampilan ketika wishlist kosong

6.12 Data Test Start

Tes ini dilakukan untuk mengecek keberjalanannya program utama dan kondisi awal program dimana terdapat perubahan command pada Tugas Besar 2 ini

```

=====
|   / \   |-----|   / \   |
|  / \  |-----|  / \  |
| / \ \ |-----| / \ \ |
|/ \ \ \|-----|/ \ \ \|
=====

*****
* Selamat Datang di PURRMART *
*****
* 1. START      *
* 2. LOAD       *
* 3. QUIT       *
*****
* Pilih Command atau ketik 'HELP'! *
*****
Ketik command atau "Help"
>>> █

```

Gambar 6.12.1 Kondisi awal program

```

Ketik command atau "Help"
>>> start
Konfigurasi aplikasi berhasil dibaca. PURRMART berhasil dijalankan.
*****
* Silahkan Bergabung di PURRMART *
*****
* 1. LOGIN      *
* 2. REGISTER   *
* 3. LOGOUT     *
*****
* Pilih Command atau ketik 'HELP'! *
*****
Ketik command atau "Help"
>>> █

```

Gambar 6.12.2 Tampilan ketika berhasil menjalankan fungsi START

6.13 Data Test Load Save

Tes ini dilakukan untuk mengecek keberjalanannya fungsi *SAVE* untuk menyimpan *state* aplikasi terbaru ke dalam suatu *file* dimana terdapat perubahan command pada Tugas Besar 2 ini.

```

Ketik command atau "Help"
>>> load default.txt
Konfigurasi aplikasi berhasil dibaca. PURRMART berhasil dijalankan.
*****
* Silahkan Bergabung di PURRMART *
*****
* 1. LOGIN      *
* 2. REGISTER   *
* 3. LOGOUT     *
*****
* Pilih Command atau ketik 'HELP'!
*****

```

Gambar 6.13.1 Tampilan LOAD ketika berhasil dijalankan

STEI- ITB	IF2111-TB2-K2-03	Halaman 17 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

```
Ketik command atau "Help"
>>> save testing.txt
Save file berhasil disimpan. save berhasil.
```

Gambar 6.13.2 Tampilan SAVE ketika berhasil dijalankan

6.14 Data Test Store List

Tes ini dilakukan untuk mengecek *list* barang yang tersedia di toko dimana barang bersifat *unique* dimana terdapat perubahan command pada Tugas Besar 2 ini.

```
Ketik command atau "Help"
>>> store list
Daftar barang di toko:
- "AK47" - harga: 10
- "Lalabu" - harga: 20
- "Ayam Goreng Crisbar" - harga: 20
- "Meong" - harga: 500
```

Gambar 6.14.1 Tampilan ketika STORE LIST berhasil dijalankan

6.15 Data Test Help

Tes ini dilakukan untuk mengecek keberjalanannya fungsi *HELP*.

```

Ketik command atau "Help"
>>> help
=====
||          MAIN MENU HELP
=====
|| Berikut adalah Command yang dapat Anda lakukan:
|| -----
|| 1. PROFILE
||     Untuk menampilkan data diri dan jumlah saldo user
|| 2. WORK
||     Bekerja untuk mendapatkan uang
|| 3. WORK CHALLENGE
||     Mengerjakan tantangan
|| 4. STORE LIST
||     Melihat barang-barang di toko
|| 5. STORE REQUEST
||     Meminta penambahan barang
|| 6. STORE SUPPLY
||     Menambahkan barang dari permintaan
|| 7. STORE REMOVE
||     Menghapus barang dari toko
|| 8. BIO WEAPON
||     Menambah barang berupa senjata biologis
|| 9. CART ADD <nama barang> <jumlah barang>
||     Untuk menambahkan barang ke dalam keranjang
|| 10. CART REMOVE <nama barang> <jumlah barang>
||     Untuk mengurangi/menghapus barang dari keranjang
|| 11. CART SHOW
||     Untuk menampilkan barang yang ada di keranjang
|| 12. CART PAY
||     Untuk membeli barang yang ada di keranjang
|| 13. HISTORY <n>
||     Untuk menampilkan n riwayat pembelian terakhir
=====
```

Gambar 6.15.1 Tampilan ketika fungsi HELP berhasil dijalankan

```

|| 11. CART SHOW
||     Untuk menampilkan barang yang ada di keranjang
|| 12. CART PAY
||     Untuk membeli barang yang ada di keranjang
|| 13. HISTORY <n>
||     Untuk menampilkan n riwayat pembelian terakhir
|| 14. WISHLIST ADD
||     Untuk menambahkan barang ke dalam wishlist
|| 15. WISHLIST SWAP <i> <j>
||     Untuk menukar posisi dua barang pada wishlist
|| 16. WISHLIST REMOVE <i>
||     Untuk menghapus sebuah barang pada wishlist
||     berdasarkan posisi barang di wishlist
|| 17. WISHLIST REMOVE
||     Untuk menghapus sebuah barang pada wishlist
||     berdasarkan nama barang
|| 18. WISHLIST CLEAR
||     Untuk menghapus semua barang pada wishlist
|| 19. WISHLIST SHOW
||     Untuk melihat semua barang pada wishlist
|| 20. LOGOUT
||     Keluar dari sesi
|| 21. SAVE
||     Menyimpan state ke dalam file
|| 22. QUIT
||     Keluar dari program
=====
```

Gambar 6.15.2 Tampilan ketika fungsi HELP berhasil dijalankan

6.16 Data Test Log In

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

STEI- ITB	IF2111-TB2-K2-03	Halaman 19 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

```

Ketik command atau "Help"
>>> login
Username: user1
Password: alstrukdatkeren
Login berhasil. Selamat datang, user1!

```

Gambar 6.16.1 Tampilan ketika LOGIN berhasil dimana input password dan username benar

7 Test Script

No.	Fitur yang Dtes	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	PROFILE	Memastikan data pengguna seperti nama dan saldo ditampilkan dengan benar.	- Menginput command PROFILE saat pengguna telah login - Sistem akan mengeluarkan output berupa nama dan saldo yang dimiliki	PROFILE	Gambar 6.1.1 Gambar 6.1.2 Gambar 6.1.3	Sesuai dengan hasil yang diharapkan
2	CART ADD	Memastikan bahwa pengguna bisa memasukkan barang jika barang tersedia di toko.	- Menginput command CART ADD <nama> <n> - Sistem akan menambahkan barang dengan kuantitas yang ditulis pengguna ke dalam keranjang belanja	CART ADD <nama> <n>	Gambar 6.2.1	Sesuai dengan hasil yang diharapkan
3	CART REMOVE	Memastikan pengguna bisa menghapus atau mengurangi jumlah barang di keranjang sesuai dengan input.	- Menginput command CART REMOVE <nama> <n> - Sistem akan melakukan validasi terhadap kuantitas yang diberikan - Sistem akan mengurangi barang dengan kuantitas yang ditentukan oleh pengguna dari keranjang belanja jika validasi berhasil	CART REMOVE <nama> <n>	Gambar 6.3.1 Gambar 6.3.2	Sesuai dengan hasil yang diharapkan
4	CART SHOW	Memastikan bahwa sistem bisa menunjukkan barang-barang yang sudah dimasukkan ke dalam keranjang	- Menginput command CART SHOW - Sistem akan mengeluarkan output berupa barang-barang yang dimasukkan ke dalam keranjang	CART SHOW	Gambar 6.4.1 Gambar 6.4.2	Sesuai dengan hasil yang diharapkan

		dalam keranjang.	dimana apabila keranjang kosong, sistem akan mengeluarkan output yang memberitahu pengguna bahwa keranjang kosong			
5	CART PAY	Memastikan pembayaran berhasil jika saldo cukup.	<ul style="list-style-type: none"> - Menginput command CART PAY - Sistem akan memastikan terlebih dahulu apakah uang pengguna cukup untuk membeli seluruh barang di keranjang. - Jika pengguna jadi membeli barang dengan uang yang cukup maka barang tersebut berhasil dibeli dan jika tidak mencukup sistem akan menolak pembelian barang yang dilakukan 	CART PAY	Gambar 6.5.1	Sesuai dengan hasil yang diharapkan
6	HISTORY	Menampilkan riwayat pembelian berdasarkan transaksi terbaru hingga terlama.	<ul style="list-style-type: none"> - Menginput command HISTORY <n> - Sistem akan menampilkan riwayat pembelian sebanyak n transaksi terakhir. 	HISTORY <n>	Gambar 6.6.1 Gambar 6.6.2	Sesuai dengan hasil yang diharapkan
7	WISHLIST ADD	Memastikan sistem dapat menambahkan suatu barang ke wishlist.	<ul style="list-style-type: none"> - Menginput command WISHLIST ADD - Sistem akan menambahkan barang ke wishlist jika barang tersedia dan belum ada di wishlist. 	WISHLIST ADD	Gambar 6.7.1 Gambar 6.7.2	Sesuai dengan hasil yang diharapkan
8	WISHLIST SWAP	Memastikan sistem dapat menukar barang posisi ke-i dengan barang posisi ke-j pada wishlist	<ul style="list-style-type: none"> - Menginput command WISHLIST SWAP <i><j> - Sistem akan menukar posisi barang ke-i dengan barang ke-j di wishlist. Apabila jumlah barang hanya terdapat 1, barang gagal ditukar. 	WISHLIST SWAP <i><j>	Gambar 6.8.1	Sesuai dengan hasil yang diharapkan
9	WISHLIST REMOVE	Memastikan sistem dapat menghapus barang dengan	<ul style="list-style-type: none"> - Menginput command WISHLIST REMOVE - Sistem akan menghapus barang di 	WISHLIST REMOVE	Gambar 6.10.1 Gambar 6.10.	Sesuai dengan hasil yang diharapkan

		posisi ke-i dari <i>wishlist</i> berdasarkan nama barang yang dimasukkan pengguna.	posisi ke-i apabila valid (<i>wishlist</i> dan posisi barang ke-i tidak kosong) dan sistem akan menghapus barang dari <i>wishlist</i> berdasarkan nama yang dimasukkan. Apabila nama yang dimasukkan tidak valid (tidak ada di <i>wishlist</i>), perintah tidak valid.			
10	WISHLIST CLEAR	Memastikan sistem dapat menghapus semua barang yang terdapat di dalam WISHLIST.	- Menginput command WISHLIST CLEAR - Sistem akan mengosongkan seluruh isi <i>wishlist</i> .	WISHLIST CLEAR	Gambar 6.11.1 Gambar 6.11.2	Sesuai dengan hasil yang diharapkan
11	WISHLIST SHOW	Memastikan sistem dapat menunjukkan barang barang yang dimasukkan ke dalam keranjang	- Menginput command WISHLIST SHOW - Sistem akan menunjukkan barang barang di toko	WISHLIST SHOW	Gambar 6.12.1 Gambar 6.12.2	Sesuai dengan hasil yang diharapkan
12	START	Memeriksa keberhasilan keberjalanan file konfigurasi	Menginput command START pada saat pertama kali memulai program	START	Gambar 6.13.1 Gambar 6.13.2	Sesuai dengan hasil yang diharapkan
13	LOAD SAVE	Menjalankan file konfigurasi dan Menyimpan state aplikasi terbaru ke dalam suatu file.	- Menginput command LOAD yang memiliki satu argumen berupa <i>filename</i> - Menginput command SAVE. - Sistem akan menyimpan state ke dalam suatu file beserta argumen yang merepresentasikan nama file yang akan disimpan.	LOAD < <i>filename</i> > SAVE < <i>filename</i> >	Gambar 6.14.1	Sesuai dengan hasil yang diharapkan
14	STORE LIST	Memperlihatkan list barang yang ada di toko (barang bersifat unique)	- Menginput command STORE LIST. - Sistem memperlihatkan list barang yang ada di toko.	STORE LIST	Gambar 6.14.1	Sesuai dengan hasil yang diharapkan

15	HELP	Menampilkan daftar <i>command</i> yang mungkin untuk dijalankan beserta dengan deskripsinya	- Menginput <i>command</i> HELP. - Sistem akan menampilkan daftar <i>command</i> dan deskripsinya sesuai dengan dimana perintah tersebut dipanggil .	HELP	Gambar 6.15.1 Gambar 6.15.2	Sesuai dengan hasil yang diharapkan
16	LOGIN	Masuk ke dalam program dengan akun pengguna yang sudah terdaftar	- Menginput <i>command</i> LOGIN. - Memasukkan <i>username</i> dan <i>password</i> dari pengguna. - Sistem melakukan verifikasi terhadap masukan pengguna.	LOGIN	Gambar 6.16.1	Sesuai dengan hasil yang diharapkan

8 Pembagian Kerja dalam Kelompok

Nama Anggota	Pembagian Kerja
Muhammad Aidan Fathullah 18223002	<ul style="list-style-type: none"> • Mengerjakan ADT • Merevisi keseluruhan kode • Melakukan debugging
Vandega Arozan Musholine 18223010	<ul style="list-style-type: none"> • Menyusun cart_pay, save, dan driver ADT
Irdina Ilmuna Yosapat 18223060	<ul style="list-style-type: none"> • Menyusun fitur History, Cart Show, Wishlist Show, dan Wishlist Add • Membuat laporan bagian Ringkasan, Data Test, Test Script, Lampiran, Algoritma Algoritma menarik dan Log Activity Anggota Kelompok <ul style="list-style-type: none"> • Merevisi laporan dan Milestone 1
Vincentia Belinda Sumartoyo 18223078	<ul style="list-style-type: none"> • Menyusun cart_add, cart_remove, wishlist_add, wishlist_swap • Membuat laporan
Gabriela Jennifer Sandy 18223092	<ul style="list-style-type: none"> • Menjadi penghubung dengan asisten • Mengerjakan bagian Help, Wishlist Remove • Merevisi laporan dan Milestone 1 • Mengerjakan laporan bagian Data Testing,

9 Lampiran

9.1 Deskripsi Tugas Besar

PURRMART merupakan suatu sistem jual beli ke Borma yang diperlukan oleh Agen Purry. Ia mendapat misi baru 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. Seluruh *command* hanya berlaku pada menu utama.

a. PROFILE

PROFILE adalah *command* yang digunakan untuk melihat data diri pengguna. PROFILE hanya dapat dipanggil saat status pengguna telah login

b. CART ADD <nama> <n>

CART ADD adalah *command* yang digunakan untuk menambahkan barang dengan kuantitas tertentu ke dalam keranjang belanja.

c. CART REMOVE <nama> <n>

CART REMOVE adalah *command* yang digunakan untuk mengurangi barang sejumlah kuantitas tertentu dari keranjang belanja. Perlu dilakukan validasi terhadap kuantitas yang diberikan, bila kuantitas pada keranjang belanja lebih sedikit dari N maka perintah akan gagal.

d. CART SHOW

CART SHOW adalah *command* yang digunakan untuk menunjukkan barang-barang yang sudah dimasukkan ke dalam keranjang.

e. CART PAY

CART PAY adalah *command* yang digunakan untuk membeli barang-barang yang sudah dimasukan ke dalam keranjang. Perlu dipastikan bahwa **pengguna memiliki uang yang cukup** untuk membeli seluruh barang keranjang. Pembelian akan mengurangi uang yang dimiliki pengguna dan menambahkan riwayat pembelian.

Nama barang yang dimasukan ke riwayat pembelian adalah barang dengan total harga (harga barang * kuantitas) terbesar. Jika terdapat lebih dari 1 barang dengan total yang sama, maka yang disimpan adalah barang dengan urutan lexical yang lebih besar. Dimasukan juga total harga pada pembelian tersebut.

f. HISTORY <n>

HISTORY adalah *command* yang digunakan untuk menunjukkan riwayat pembelian seorang pengguna. N merupakan jumlah riwayat yang ditampilkan, contoh N=3 maka akan menampilkan 3 riwayat pembelian terbaru. Jika N melebihi jumlah riwayat pembelian yang ada, maka seluruh riwayat pembelian akan ditampilkan. Urutan penunjukan adalah dari yang paling baru ke paling tua.

g. WISHLIST ADD

WISHLIST ADD merupakan *command* yang digunakan untuk menambahkan suatu barang ke *wishlist*.

h. WISHLIST SWAP <i><j>

WISHLIST SWAP merupakan *command* yang digunakan untuk menukar barang posisi ke-i dengan barang posisi ke-j pada *wishlist*. Posisi i dan j merupakan urutan barang pada *wishlist*, urutan dimulai dari 1.

i. WISHLIST REMOVE

WISHLIST REMOVE adalah *command* yang digunakan untuk menghapus barang dengan posisi ke-i dari *wishlist* atau menghapus barang dari *wishlist* berdasarkan nama barang yang dimasukkan pengguna.

j. WISHLIST CLEAR

WISHLIST CLEAR adalah *command* yang digunakan untuk menghapus semua barang yang terdapat di dalam WISHLIST.

k. WISHLIST SHOW

WISHLIST SHOW adalah *command* yang digunakan untuk menunjukkan barang-barang yang sudah dimasukkan ke dalam *wishlist*.

4. Perubahan Command

Terdapat beberapa *command* iterasi sebelumnya yang berubah, yaitu sebagai berikut.

a. START, LOAD, dan SAVE

Terdapat perubahan konfigurasi yang harus ditambahkan dalam implementasi *command* START, LOAD, dan SAVE.

STEI- ITB	IF2111-TB2-K2-03	Halaman 26 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

b. STORE LIST

STORE LIST akan menampilkan nama barang yang dijual **beserta harganya**.

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: baiknya header dan ADT digabungkan driver diwajibkan jadi sebaiknya digabung satu file
Tempat : Zoom Kehadiran Anggota Kelompok: No NIM Tanda tangan 1 18223002  2 18223010 	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. di dalam folder ADT, ada file-file nya. struktur foldernya dirapihin dulu gitu, mau manggil headernya dari mana gitu, gabakal bingung lagi. default config boleh mengikuti yang ada di spek. enaknya ADT list nya dipisah, dinamis sama statis. drivernya jadinya juga misah gitu. jangan lupa error handling, kayak masukin spasi doang/string/integer or something gitu.

3
18223060



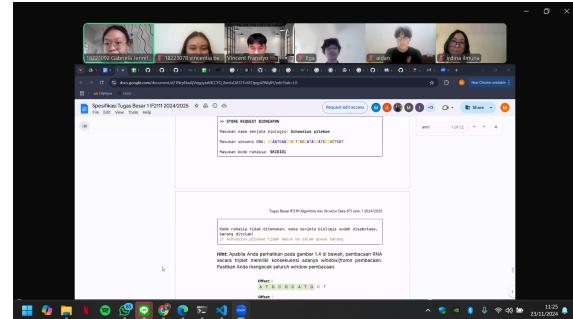
4
18223078



5
18223092



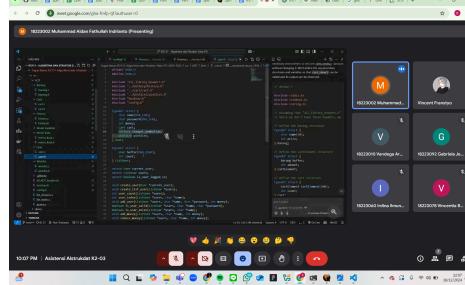
disarankan buat make file
read me jangan lupa!!!

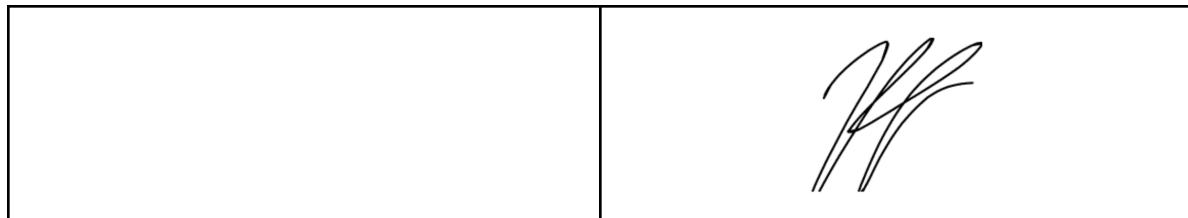


Tanda Tangan Asisten:



Asistensi II

Tanggal : 18 Desember 2024	Catatan Asistensi:
Tempat : Zoom	
Kehadiran Anggota Kelompok: No NIM Tanda tangan	<ol style="list-style-type: none">1. Isi stack nya array, bukan stack yang kayak di praktikum gitu. Masih menggunakan konsep stack, dan elemen di dalam stack nya diganti-ganti → gapapa2. Apabila ada perubahan di milestone 1 nya nanti, ubah di laporannya juga3. Kasih batasan nanti constraint nya sampai keberapa gitu juga. Di laporan dimention juga ya nanti. Kalau tetep mau pake BFS dan punya alasan yang kuat, silahkan aja.4. Help nya nanti tambahan kayak profile, cart_add dkk juga (disaranin)5. Revisi laporannya nanti update di repo juga
1 18223002 	
2 18223010 	
3 18223060 	
4 18223078 	
5 18223092 	
	Tanda Tangan Asisten:



9.3 Log Activity Anggota Kelompok

No.	Tanggal	Anggota	Aktivitas
1	03/12/2024	18223002 Muhammad Aidan Fathullah I 18223010 Vandega Arozan Musholine 18223060 Irdina Ilmuna Yosapat 18223078 Vincentia Belinda Sumartoyo 18223092 Gabriela Jennifer Sandy	Kerja kelompok pertama untuk merevisi Milestone 1
		18223092 Gabriela Jennifer Sandy	Merevisi Milestone 1
2	04/12/2024	18223078 Vincentia Belinda Sumartoyo 18223092 Gabriela Jennifer Sandy	Merevisi tebak angka Update README.md
3	11/12/2024	18223002 Muhammad Aidan Fathullah I 18223010 Vandega Arozan Musholine 18223060 Irdina Ilmuna Yosapat 18223078 Vincentia Belinda Sumartoyo	Kerja kelompok merevisi Milestone 1
4	11/12/2024	18223002 Muhammad Aidan Fathullah I	Fix minor bug

		18223010 Vandega Arozan Musholine	
		18223092 Gabriela Jennifer Sandy	Revisi ADT List
5	16/12/2024	18223078 Vincentia Belinda Sumartoyo	Push Cart add, Cart Remove, Wishlist Add dan Wishlist Swap
6	17/12/2024	18223010 Vandega Arozan Musholine	Push Profile
		18223060 Irdina Ilmuna Yosapat	Push Wishlist Show, Wishlist Clear, Cart Show, dan Cart Pay
7	18/12/2024	18223092 Gabriela Jennifer Sandy	Push Wishlist Remove <i>
		18223010 Vandega Arozan Musholine	Push Cart Pay
		18223060 Irdina Ilmuna Yosapat	Push History
8	19/12/2024	18223002 Muhammad Aidan Fathullah I	Kerja kelompok ketiga
		18223010 Vandega Arozan Musholine	
		18223060 Irdina Ilmuna Yosapat	
		18223078 Vincentia Belinda Sumartoyo	
		18223092 Gabriela Jennifer Sandy	
9	20/12/2024	18223092 Gabriela Jennifer Sandy	Update Console and Help dan All Wishlist
10	20/12/2024	18223002 Muhammad Aidan Fathullah I	Kerja kelompok keempat
		18223010 Vandega Arozan Musholine	
		18223060 Irdina Ilmuna Yosapat	

STEI- ITB	IF2111-TB2-K2-03	Halaman 31 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

		18223078 Vincentia Belinda Sumartoyo	
		18223092 Gabriela Jennifer Sandy	
10	20/12/2024	18223002 Muhammad Aidan Fathullah I	Fix all
		18223010 Vandega Arozan Musholine	
		18223078 Vincentia Belinda Sumartoyo	
		18223092 Gabriela Jennifer Sandy	