LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data STI

PURRMART

Dipersiapkan oleh:

Borma Torserba Dago

Nabilah Amanda Putri	18221021
Muhammad Rifky Fachrizain	18221027
Ali Syauqie	18223045
Stanislaus Ardy Bramantyo	18223057

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

Daftar Isi

1. Ringkasan 4
2. Struktur Data (ADT) 6
    2.1. Stack 6
    2.2. Setmap 7
    2.3. Linked list 8
3. Program Utama 9
    3.1. Main 9
    3.2. Pembacaan dan Inisialisasi File Konfigurasi 9
    3.3. Pemanggilan Command 10
    3.3.1. PROFILE 10
    3.3.2. CART ADD <nama> <n> 10
    3.3.3. CART REMOVE <nama> <n> 10
    3.3.4. CART SHOW 10
    3.3.5. CART PAY 11
    3.3.6. HISTORY <n> 11
    3.3.7. WISHLIST ADD 11
    3.3.8. WISHLIST SWAP <i> <j> 11
    3.3.9. WISHLIST REMOVE <i> 11
    3.3.10. WISHLIST REMOVE 11
    3.3.11. WISHLIST CLEAR 11
    3.3.12. WISHLIST SHOW 12
4. Data Test 12
    4.1. Data Test 1: PROFILE 12
    4.2. Data Test 2: CART ADD 12
    4.3. Data Test 3: CART REMOVE 12
    4.4. Data Test 4: CART SHOW 12
    4.5. Data Test 5: CART PAY 13
    4.6. Data Test 6: HISTORY 13
    4.7. Data Test 7: WISHLIST ADD 13
    4.8. Data Test 8: WISHLIST SWAP 13
    4.9. Data Test 9: WISHLIST REMOVE <i> 14
    4.10. Data Test 10: WISHLIST REMOVE 14

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.

4.11. Data Test 11: WISHLIST CLEAR 14
4.12. Data Test 2: WISHLIST SHOW 14
5. Test Script 15
6. Pembagian Kerja dalam Kelompok 18
7. Lampiran 20
7.1. Deskripsi Tugas Besar 20
Spesifikasi Umum 20
System Mechanic 20
1. About the System 20
2. Menu Program 20
3. Command 21
a. PROFILE 21
b. CART ADD <nama> <n> 21
c. CART REMOVE <nama> <n> 21
d. CART SHOW 22
e. CART PAY 22
f. HISTORY <n> 24
g. WISHLIST ADD 25
h. WISHLIST SWAP <i> <j> 25
i. WISHLIST REMOVE <i> 26
j. WISHLIST REMOVE 27
k. WISHLIST CLEAR 27
l. WISHLIST SHOW 28
4. Perubahan Command 28
a. START, LOAD, dan SAVE 28
b. STORE LIST 28
Konfigurasi Sistem 29
Daftar ADT 30
Bonus 31
7.2. Notulen Rapat 36
7.3. Log Activity Anggota Kelompok 37

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.

1. Ringkasan

PURRMART adalah sebuah aplikasi yang dibuat dengan fitur untuk mensimulasikan sebuah *e-commerce* yang interaktif. Aplikasi ini memungkinkan *user* untuk mengerti *flowchart* dan proses dari sebuah aplikasi *e-commerce* seperti *wishlist*, daftar keranjang, riwayat pembelanjaan, toko, dan lainnya.

Dalam pengembangan aplikasi ini, PURRMART menggunakan beberapa struktur data yang mencakup ArrayDin, ArrayStat, Mesin Karakter, Mesin Kata, Barang, User, *Queue, Stack, Setmap,* dan *Linkedlist*. Struktur data ini dipilih untuk menangani operasi penting seperti penyimpanan daftar barang, manajemen antrian *orderi* dan *supply*, serta pemrosesan kata dari *input* user. Terdapat juga beberapa prosedur yang berisi fungsi-fungsi utama seperti *start, help, load, login, work, register, logout, save, store_request, store_list*, *store_remove*, dan *store_supply* yang menjalankan logika fitur aplikasi. Terdapat juga dua *work challenge* yaitu WORDL3 dan Tebak Angka.

Dengan *milestone* terbaru, terdapat penambahan fungsi seperti *profile, cartAdd, cartRemove, cartShow, cartPay, history, wishlist_add, wishlist_swap, wishlist_remove, wishlist_clear,* dan *wishlist_show*.

Secara keseluruhan, tugas besar PURRMART dapat memberi gambaran dan membentuk sebuah aplikasi *e-commerce* yang sederhana dan mudah untuk dipahami dengan memanfaatkan berbagai variasi struktur data yang telah dipelajari selama keberjalanan IF2111 Algoritma dan Sturktur Data.

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.

2. Struktur Data (ADT)

Untuk *milestone* ke-2 ini, kami menambahkan 3 struktur data baru dan memodifikasi ADT User untuk mengimplementasikannya ke beberapa tipe *fungsi* yaitu *cart, history,* dan *wishlist*. Struktur data yang kami gunakan adalah *stack, setmap*, dan *linked list* dengan variasi *doubly*.

2.1. User

ADT User adalah abstraksi untuk merepresentasikan seorang pengguna dalam sistem belanja. Setiap pengguna memiliki atribut berupa informasi pribadi, saldo uang, keranjang belanja, riwayat pembelian, dan daftar wishlist. ADT ini dirancang untuk memudahkan pengelolaan aktivitas pengguna dalam aplikasi belanja.

- `void setName(User* user, const char* name)`

  Mengatur nama pengguna dengan string yang diberikan sebagai parameter.
- `void setPassword(User* user, const char* password)`

  Mengatur password pengguna dengan string yang diberikan sebagai parameter.
- `void setMoney(User* user, int money)`

  Mengatur saldo uang (money) pengguna.
- `User createUser(const char* name, const char* password, int money)`

  Membuat dan menginisialisasi pengguna baru dengan nama, password, dan saldo awal sesuai parameter yang diberikan.
- `void addUser(ListUser *LUser, User u)`

  Menambahkan pengguna baru ke dalam daftar pengguna jika kapasitas daftar masih mencukupi.

2.2. Stack

ADT Stack adalah sebuah struktur data untuk menyimpan data dengan konsep LIFO (*Last In First Out*) dimana data terakhir disimpan akan menjadi data yang pertama kali keluar. Dalam dunia nyata, ini menggunakan konsep *tower of hanoi* atau sebuah tumpukan keranjang belanja.

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.

- `void StackInitEmpty(Stack* st)`

  Menginisialisasi stack kosong dengan panjang len = 0 dan top = NULL.

- `int isEmpty(Stack st)`

  Menentukan apakah *stack* kosong atau tidak.

- `void push(Stack* stack, StackElType el)`

  Mendorong atau memasukan data ke dalam *stack*.

- `void pushNT(Stack* stack, const char* nama_barang, int total_harga)`

  Menambahkan elemen baru ke atas stack dengan nama barang dan total harga secara langsung.

- `void pop(Stack* stack)`

  Mengeluarkan data dari *stack* menggunakan prinsip LIFO.

- `StackElType top(Stack stack)`

  Menampilkan data teratas pada *stack*.

2.3. Setmap

ADT SetMap adalah struktur data yang digunakan untuk menyimpan pasangan key-value, di mana key berupa nama barang yang unik dan value adalah kuantitas barang tersebut. ADT ini cocok untuk digunakan dalam pengelolaan keranjang belanja karena memastikan setiap barang memiliki kuantitas yang terdefinisi dengan baik tanpa duplikasi key.

- `void MapCreateEmpty(Map *map);`

  Menginisialisasi sebuah Map kosong dengan elemen awal 0.

- `int MapIsEmpty(Map map);`

  Memeriksa apakah Map kosong. Mengembalikan 1 jika kosong dan 0 jika tidak.

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.

- `void MapInsert(Map *map, const char *key, int value);`

  Menambahkan pasangan key-value baru ke dalam Map. Jika key sudah ada, fungsi akan menambahkan value pada elemen yang sudah ada.

- `int MapGetEl(Map map, const char *key);`

  Mengembalikan value dari key tertentu. Jika key tidak ditemukan, fungsi akan mengembalikan -1.

- `void MapDelete(Map *map, const char *key);`

  Menghapus elemen dengan key tertentu dari Map dan menggeser elemen setelahnya untuk mengisi posisi yang kosong.

- `int MapContains(Map map, const char *key);`

  Memeriksa apakah Map mengandung elemen dengan key tertentu. Mengembalikan 1 jika ditemukan, dan 0 jika tidak.

2.4. Linked list

ADT Linked list adalah *list linear* yang merupakan sebuah kumpulan data yang sekuensial dan saling berhubungan. Variasi yang dipakai untuk *list linear* adalah *doubly linked list* yang merupakan sebuah variasi yang dapat tidak hanya menghubungkan data di depannya, melainkan data sebelumnya juga terhubung. Ini akan membantu untuk mengakses data sehingga tidak perlu mulai dari awal selalu.

- `void createEmptyLinkedList(DoublyLinkedList *list)`

  Membuat *linked list* baru yang kosong.

- `void LinkedListInsertBeginning(DoublyLinkedList* list, const char *elem)`

  Menambahkan sebuah *element* baru di posisi awal pada *linked list*.

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.

- `void LinkedListInsertElem(DoublyLinkedList* list, const char *elem, int pos)`

  Menambahkan sebuah *element* baru di posisi yang diingkan *user* pada *linked list*.

- `void LinkedListInsertEnd(DoublyLinkedList* list, const char *elem)`

  Menambahakan sebuah *element* baru di posisi akhir pada *linked list*.

- `void LinkedListDeleteElem(DoublyLinkedList* list, int pos)`

  Menghapus sebuah *element* di posisi yang diinginkan *user* pada *linked list*.

- `void LinkedListDeleteBeginning(DoublyLinkedList* list)`

  Menghapus sebuah *element* di posisi awal pada *linked list*.

- `void LinkedListDeleteEnd(DoublyLinkedList* list)`

  Menghapus sebuah *element* di posisi akhir pada *linked list*.

- `void printDoublyLinkedList(DoublyLinkedList list)`

  Menampilkan seluruh *element* yang pada *linked list currently*.

- `const char* LinkedListGetElmt(DoublyLinkedList list, int pos)`

  Mendapatkan posisi dari *element* yang diinginkan.

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.

3. Program Utama

3.1. Main

Kode utama (main) pada program ini merupakan entry point aplikasi PURRMART, sebuah simulasi toko yang dikelola dengan berbagai fitur interaktif. Program dimulai dengan menampilkan pesan selamat datang dan memberikan informasi mengenai perintah awal yang tersedia, seperti START, LOAD, dan HELP. Selanjutnya, pengguna diarahkan untuk memilih dan menjalankan sesi konfigurasi awal melalui file atau login ke sistem melalui proses otentikasi. Setelah berhasil masuk, pengguna dapat mengakses berbagai fitur utama, seperti bekerja (WORK), berinteraksi dengan inventori toko (STORE), menjalankan tantangan kerja (WORK CHALLENGE), atau menyimpan dan keluar dari aplikasi (SAVE dan QUIT). Logika program mengatur alur melalui serangkaian validasi input dan eksekusi perintah berbasis kata menggunakan modul pendukung, seperti mesin kata, array dinamis, dan queue, untuk mendukung berbagai operasi sistem.

Selain itu, untuk Milestone 2 pada tugas besar ini terdapat beberapa tambahan fitur yang ditambahkan pada program. Terdapat fitur *cart* atau keranjang pembelian untuk menyimpan barang yang ingin dibeli untuk sementara dengan spesifikasi dapat menambahkan, menghapus, menunjukan, dan membayar isi dari keranjang pembelian. Terdapat juga fitur *wishlist* yaitu sebuah daftar keinginan barang dengan spesifikasi dapat menambahkan, menukar urutan, menghapus, menunjukan, dan menghapus isi dari daftar keinginan *user*. Terdapat juga fitur *profile* untuk menunjukan nama dan saldo *user* yang sedang ter-*login* serta fitur *history* untuk menunjukan riwayat pembelian barang *user*.

3.2. Pembacaan dan Inisialisasi File Konfigurasi

Proses pembacaan dan inisialisasi file konfigurasi dalam program ini dilakukan pada tahap awal, di mana pengguna diberikan opsi untuk memulai aplikasi dengan perintah START atau LOAD. Perintah START akan membaca file konfigurasi default yang sudah ditentukan, sedangkan LOAD memungkinkan pengguna untuk memasukkan nama file konfigurasi tertentu. Program menggunakan fungsi seperti startKata() untuk membaca

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.

input dari pengguna dan menambahkan path default ke nama file saat diperlukan. Setelah file ditemukan, program memuat data seperti daftar pengguna (ArrayStat) dan barang di toko (ArrayDin) ke dalam memori untuk digunakan dalam sesi aplikasi. Jika file gagal ditemukan atau ada kesalahan dalam proses inisialisasi, pengguna akan diberikan pesan error untuk mencoba kembali. Inisialisasi yang berhasil memastikan bahwa aplikasi siap digunakan dengan data yang sesuai dari file konfigurasi.

3.3. Pemanggilan Command

3.3.1. PROFILE

Perintah ini digunakan untuk menunjukan rincian nama dan saldo dari *user* yang sedang *login*.

3.3.2. CART ADD <nama> <n>

Perintah ini digunakan untuk menambahkan sebuah barang dengan spesifikasi nama dan jumlah barang yang ingin ditambahkan ke dalam keranjang pembelian *user*.

3.3.3. CART REMOVE <nama> <n>

Perintah ini digunakan untuk menghapus atau mengurangi suatu barang dengan jumlah yang diinginkan *user* dari keranjang pembeliannya.

3.3.4. CART SHOW

Perintah ini akan menampilkan kuantitas, nama, dan total harga tiap barang serta jumlah harga dari seluruh barang yang ada pada keranjang pembelian *user*.

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.

3.3.5. CART PAY

Perintah ini akan menampilkan keranjang pembelian seperti pada perintah *CART SHOW* dan akan meminta *input* apakah ingin membeli seluruh barang yang ada pada keranjang pembelian.

3.3.6. HISTORY <n>

Perintah ini akan menunjukan *n* baris dari riwayat pembelian barang dari *user* yang berisikan nama serta <...>.

3.3.7. WISHLIST ADD

Perintah ini akan meminta *input* sebuah nama barang dan akan menambahkannya ke *wishlist* bila barang tersebut ada pada toko dan belum ada pada *wishlist*.

3.3.8. WISHLIST SWAP <i> <j>

Perintah ini akan menukar urutan dari barang indeks-*i* dengan indeks-*j* bila nilai *i* dan *j* merupakan indeks yang *valid* (indeks-*i* dan *j* berisikan barang dan tidak *NULL*).

3.3.9. WISHLIST REMOVE <i>

Perintah ini akan menghapus barang yang ada pada *wishlist* di indeks-*i* bila nilai *i* merupakan indeks yang *valid*.

3.3.10. WISHLIST REMOVE

Perintah ini akan menerima *input* sebuah nama barang dan menghapus barang tersebut dari *wishlist* bila barang tersebut sudah ada pada *wishlist*.

3.3.11. WISHLIST CLEAR

Perintah ini akan menghapus *wishlist* hingga kosong.

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.

3.3.12. WISHLIST SHOW

Perintah ini akan menampilkan daftar *wishlist* dari *user*.

4. Data Test

4.1. Data Test 1: PROFILE

Tabel 4.1 Menjelaskan Hasil *Data Test* PROFILE.

Hasil Data Test	Deskripsi
``` ========== User Atma: Saldo : 1000 ========== ```	Program berhasil menampilkan nama user dan jumlah saldonya.

4.2. Data Test 2: CART ADD

Tabel 4.2 Menjelaskan Hasil *Data Test* CART ADD.

Hasil Data Test	Deskripsi
``` Masukkan perintah CART ADD: CART ADD PULPEN 14 Barang tidak ada di toko! Masukkan perintah CART ADD: CART ADD MINYAKGORENG 20 Berhasil menambahkan 20 MINYAKGORENG ke keranjang belanja! Masukkan perintah CART ADD: CART ADD BERAS 50 Berhasil menambahkan 50 BERAS ke keranjang belanja! ```	Program berhasil mendeteksi barang yang ada di keranjang dan menambahkan jumlahnya sesuai ketersediaan barang pada toko.

4.3. Data Test 3: CART REMOVE

Tabel 4.3 Menjelaskan Hasil *Data Test* CART REMOVE.

Hasil Data Test	Deskripsi

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.

Masukkan perintah CART REMOVE: CART REMOVE BERAS 70 Tidak berhasil mengurangi, hanya terdapat 50 BERAS pada keranjang! Masukkan perintah CART REMOVE: CART REMOVE MINYAKGORENG 20 Berhasil mengurangi 20 MINYAKGORENG dari keranjang belanja!	Program berhasil mendeteksi barang yang ada di keranjang dan mengurangi jumlahnya sesuai ketersediaan barang pada keranjang.

4.4. Data Test 4: CART SHOW

Tabel 4.4 Menjelaskan Hasil *Data Test* CART SHOW.

Hasil Data Test	Deskripsi
Masukkan perintah CART SHOW: CART SHOW Keranjang kamu kosong! Berikut adalah isi keranjangmu. Kuantitas Nama Total 20 MINYAKGORENG 400000 50 BERAS 500000 Total biaya yang harus dikeluarkan adalah 900000. Masukkan perintah CART SHOW: CART SHOW Berikut adalah isi keranjangmu. Kuantitas Nama Total 50 BERAS 500000 Total biaya yang harus dikeluarkan adalah 500000.	Keranjang berhasil ditampilkan, baik keranjang kosong, keranjang dengan barang yang sudah ditambahkan maupun dikurangi, serta total biayanya.

4.5. Data Test 5: CART PAY

Tabel 4.5 Menjelaskan Hasil *Data Test* CART PAY.

Hasil Data Test	Deskripsi
Berikut adalah isi keranjangmu. Kuantitas Nama Total 1 Skibidi 2000 1 Toilet 3000 1 Kai Cenat 20000 1 Mewing 30000 Total biaya yang harus dikeluarkan adalah 55000, apakah jadi dibeli? Ya/Tidak : Ya Barang-barang sudah terbeli :D	Pembayaran berhasil dijalankan, baik itu berhasil karena uang *user* cukup, tidak berhasil karena uang *user* tidak cukup, *user* tidak jadi melakukan pembayaran, *user* memberikan input aneh, dan ketika *cart user* kosong.

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.

Berikut adalah isi keranjangmu. Kuantitas Nama Total 1 Skibidi 2000 1 Toilet 3000 1 Kai Cenat 20000 1 Mewing 30000 Total biaya yang harus dikeluarkan adalah 55000, apakah jadi dibeli? Ya/Tidak : Ya Uangmu kurang 5000 bro, Balik lagi kalau uangmu cukup ya :3c Berikut adalah isi keranjangmu. Kuantitas Nama Total 1 Skibidi 2000 1 Toilet 3000 1 Kai Cenat 20000 1 Mewing 30000 Total biaya yang harus dikeluarkan adalah 55000, apakah jadi dibeli? Ya/Tidak : Tidak Baik, Kembali ke Menu Berikut adalah isi keranjangmu. Kuantitas Nama Total 1 Skibidi 2000 1 Toilet 3000 1 Kai Cenat 20000 1 Mewing 30000 Total biaya yang harus dikeluarkan adalah 55000, apakah jadi dibeli? Ya/Tidak : Sigma Masukkanmu aneh :skull:, Kembali ke Menu Keranjang kamu kosong!	

4.6. Data Test 6: HISTORY

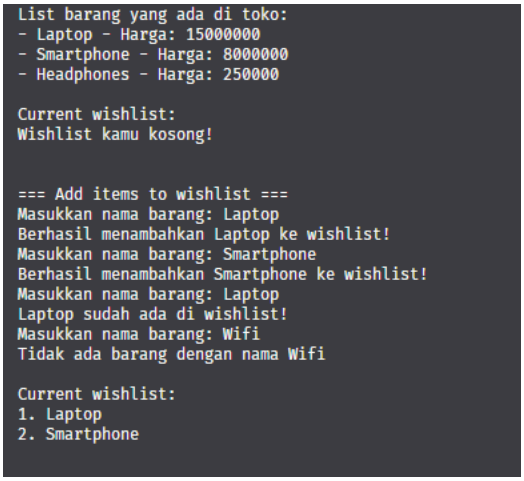
Tabel 4.6 Menjelaskan Hasil *Data Test* HISTORY.

Hasil Data Test	Deskripsi
Berapa Transaksi yang Ingin Ditampilkan? 1 1 ========== RIWAYAT TRANSAKSI ANDA ========== Anda belum beli apapun -w- ========== RIWAYAT TRANSAKSI ANDA ========== Berapa Transaksi yang Ingin Ditampilkan? 3 3 ========== RIWAYAT TRANSAKSI ANDA ========== 1. Pensil 111 2. Dorito 789 3. Albarda 456 ========== RIWAYAT TRANSAKSI ANDA ==========	Program berhasil menampilkan riwayat transaksi

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.

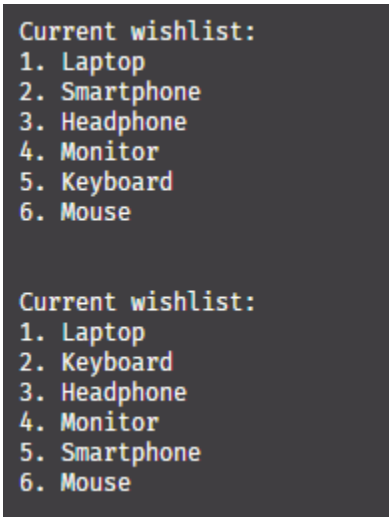
4.7. Data Test 7: WISHLIST ADD

Tabel 4.7 Menjelaskan Hasil *Data Test* WISHLIST ADD.

Hasil Data Test	Deskripsi
	Dapat menambahkan barang yang ada pada toko saja pada *wishlist* bila barang tersebut belum ada di *wishlist*.

4.8. Data Test 8: WISHLIST SWAP

Tabel 4.8 Menjelaskan Hasil *Data Test* WISHLIST SWAP.

Hasil Data Test	Deskripsi
	Dapat menukar posisi barang di *wishlist* bila *index* valid dan tidak sama. Akan ada *error* bila tidak valid ataupun sama *index* yang diberikan *user* dan tidak menjalankan penukaran posisi barang.

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.

```
Current wishlist:
1. Laptop
2. Smartphone
3. Headphone
4. Monitor
5. Keyboard
6. Mouse


Current wishlist:
1. Mouse
2. Smartphone
3. Headphone
4. Monitor
5. Keyboard
6. Laptop
```

```
Current wishlist:
1. Laptop
2. Smartphone
3. Headphone
4. Monitor
5. Keyboard
6. Mouse

Tidak bisa ditukar, index yang diberikan tidak valid [1 - 6]

Current wishlist:
1. Laptop
2. Smartphone
3. Headphone
4. Monitor
5. Keyboard
6. Mouse
```

```
Current wishlist:
1. Laptop
2. Smartphone
3. Headphone
4. Monitor
5. Keyboard
6. Mouse

Tidak perlu ditukar, index-3 dan index-3 sudah sama

Current wishlist:
1. Laptop
2. Smartphone
3. Headphone
4. Monitor
5. Keyboard
6. Mouse
```

4.9. Data Test 9: WISHLIST REMOVE <i>

Tabel 4.9 Menjelaskan Hasil *Data Test* WISHLIST REMOVE <i>.

Hasil Data Test	Deskripsi

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.

`Current wishlist:`
`1. Laptop`
`2. Smartphone`
`3. Headphone`
`4. Monitor`
`5. Keyboard`
`6. Mouse`
 	
`Berhasil menghapus barang posisi ke-1 dari wishlist!`
 	
`Current wishlist:`
`1. Smartphone`
`2. Headphone`
`3. Monitor`
`4. Keyboard`
`5. Mouse`
 	
`Current wishlist:`
`1. Laptop`
`2. Smartphone`
`3. Headphone`
`4. Monitor`
`5. Keyboard`
`6. Mouse`
 	
`Berhasil menghapus barang posisi ke-6 dari wishlist!`
 	
`Current wishlist:`
`1. Laptop`
`2. Smartphone`
`3. Headphone`
`4. Monitor`
`5. Keyboard`
 	
`Current wishlist:`
`1. Laptop`
`2. Smartphone`
`3. Headphone`
`4. Monitor`
`5. Keyboard`
`6. Mouse`
 	
`Penghapusan barang WISHLIST gagal dilakukan, Barang ke-9 tidak ada di WISHLIST!`
`Current wishlist:`
`1. Laptop`
`2. Smartphone`
`3. Headphone`
`4. Monitor`
`5. Keyboard`
`6. Mouse`
	Dapat menghapus barang di *wishlist* dengan *index* yang sesuai dan *valid*. Bila tidak, akan muncul *error* bahwa *index* yang diberikan tidak *valid*.

4.10. Data Test 10: WISHLIST REMOVE

Tabel 4.10 Menjelaskan Hasil *Data Test* WISHLIST REMOVE.

Hasil Data Test	Deskripsi

`Current wishlist:`
`1. Laptop`
`2. Smartphone`
`3. Headphone`
`4. Monitor`
`5. Keyboard`
`6. Mouse`
 	
`Masukkan nama barang yang akan dihapus: Wifi Edunex`
`Penghapusan barang wishlist gagal dilakukan, Wifi Edunex tidak ada di wishlist!`
 	
`Current wishlist:`
`1. Laptop`
`2. Smartphone`
`3. Headphone`
`4. Monitor`
`5. Keyboard`
`6. Mouse`
	Dapat menghapus barang yang ada pada *wishlist* sesuai dengan nama yang diberikan. Bila nama tidak ada pada *wishlist*, akan ada *error*.

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.

4.11. Data Test 11: WISHLIST CLEAR

Tabel 4.11 Menjelaskan Hasil *Data Test* WISHLIST CLEAR.

Hasil Data Test	Deskripsi

`Current wishlist:`
`1. Laptop`
`2. Smartphone`
`3. Headphone`
`4. Monitor`
`5. Keyboard`
`6. Mouse`
 	
`Wishlist kamu berhasil dihapus!`
 	
`Current wishlist:`
`Wishlist kamu kosong!`
	Wishlist dapat dihapus.

4.12. Data Test 12: WISHLIST SHOW

Tabel 4.12 Menjelaskan Hasil *Data Test* WISHLIST SHOW.

Hasil Data Test	Deskripsi

`Current wishlist:`
`1. Laptop`
`2. Smartphone`
`3. Headphone`
`4. Monitor`
`5. Keyboard`
`6. Mouse`
 	
`Current wishlist:`
`Wishlist kamu kosong!`
	Wishlist dapat ditampilkan dengan rincian barang. Bila kosong, akan *output* bahwa *wishlist* kosong.

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.

4.13. Data Test 13: STORE LIST

Tabel 4.13 Menjelaskan Hasil *Data Test* STORE LIST.

Hasil Data Test	Deskripsi
List barang yang ada di toko: - Laptop - Harga: 15000000 - Smartphone - Harga: 8000000 - Headphones - Harga: 250000	List dalam *store* dapat ditampilkan dengan rincian barang serta harganya.

4.14. Data Test 14: HELP

Tabel 4.14 Menjelaskan Hasil *Data Test* HELP.

Hasil Data Test	Deskripsi
Masukkan menu: 1 =====[Welcome Menu Help PURRMART]===== START -> Untuk masuk sesi baru LOAD -> Untuk memulai sesi berdasarkan file konfigurasi QUIT -> Untuk keluar dari program Masukkan menu: 2 =====[Login Menu Help PURRMART]===== REGISTER -> Untuk melakukan pendaftaran akun baru LOGIN -> Untuk masuk ke dalam akun dan memulai sesi QUIT -> Untuk keluar dari program Masukkan menu: 3 =====[Menu Help PURRMART]===== WORK -> Untuk bekerja WORK CHALLENGE -> Untuk mengerjakan challenge STORE LIST -> Untuk melihat barang-barang di toko STORE REQUEST -> Untuk meminta penambahan barang STORE SUPPLY -> Untuk menambahkan barang dari permintaan STORE REMOVE -> Untuk menghapus barang PROFILE -> Untuk melihat data diri pengguna (hanya saat login) CART ADD <nama> <n> -> Untuk menambahkan barang ke keranjang belanja CART REMOVE <nama> <n> -> Untuk mengurangi barang dari keranjang belanja CART SHOW -> Untuk melihat isi keranjang belanja CART PAY -> Untuk melakukan pembayaran keranjang belanja HISTORY <n> -> Untuk melihat riwayat pembelian WISHLIST ADD -> Untuk menambahkan barang ke wishlist WISHLIST SWAP <i> <j> -> Untuk menukar posisi barang di wishlist WISHLIST REMOVE <i> -> Untuk menghapus barang dari wishlist berdasarkan posisi WISHLIST REMOVE -> Untuk menghapus barang dari wishlist berdasarkan nama WISHLIST CLEAR -> Untuk mengosongkan seluruh isi wishlist WISHLIST SHOW -> Untuk melihat barang-barang di wishlist LOGOUT -> Untuk keluar dari sesi SAVE -> Untuk menyimpan state ke dalam file QUIT -> Untuk keluar dari program	Help untuk setiap menu berhasil ditampilkan.

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.

5. Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Fitur PROFILE	Memeriksa apakah fitur *profile* dapat ditampilkan.	Memanggil fungsi showProfile()	Data Test 1	Program berhasil menampilkan *profile* pengguna	.Hasil yang keluar sesuai dengan hasil yang diharapkan
2	Fitur CART ADD	Memeriksa apakah fitur CART ADD berhasil dijalankan	Memanggil fungsi cartadd() Memasukan perintah CART ADD yang diikuti dengan nama barang pada toko dan jumlah barang tersebut yang ingin ditambahkan ke keranjang. Apabila barang tidak ada dalam toko, menampilkan barang tidak ada pada toko dan barang tidak ditambahkan ke cart.	Data Test 2	Program berhasil validasi dan menambahkan barang ke *cart*.	Hasil yang keluar sesuai dengan hasil yang diharapkan
3	Fitur CART REMOVE	Memeriksa apakah fitur CART REMOVE berhasil menghapus barang dari keranjang.	Memanggil fungsi cartremove() Memasukan perintah CART REMOVE yang diikuti dengan nama barang pada toko dan jumlah barang tersebut yang ingin dihilangkan dari keranjang. Apabila	Data Test 3	Program berhasil memvalidasi dan menghapus barang sesuai dengan nama dan jumlah pada *cart*.	Hasil yang keluar sesuai dengan hasil yang diharapkan

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.

			barang tidak ada dalam toko, menampilkan barang tidak ada pada toko dan barang tidak ditambahkan ke cart. Apabila jumlah barang yang ingin dihapus lebih banyak dari jumlah barang pada cart, maka menampilkan *error message*.			
4	Fitur CART SHOW	Memeriksa apakah fitur CART SHOW berhasil menampilkan isi keranjang	Memanggil fitur cartshow()	Data Test 4	Program berhasil menampilkan *cart*.	Hasil yang keluar sesuai dengan hasil yang diharapkan
5	Fitur CART PAY	Memeriksa apakah fitur CART PAY berhasil untuk menambah pengguna baru	Memanggil fitur cartpay()	Data Test 5	Barang dalam keranjang berhasil dibayar pengguna apabila saldo pengguna cukup.	Hasil yang keluar sesuai dengan hasil yang diharapkan
6	Fitur HISTORY	Memeriksa apakah riwayat transaksi dapat ditampilkan dengan benar	Memanggil fungsi showHistory()	Data Test 6	Riwayat transaksi berhasil ditampilkan.	Hasil yang keluar sesuai dengan hasil yang diharapkan
7	Fitur WISHLIST ADD	Memeriksa apakah fitur WISHLIST ADD dapat validasi dan menambahkan barang ke *wishlist*.	Memanggil fungsi Wishlist_add() Memasukan nama barang dengan kriteria - Ada pada toko	Data Test 7	Program berhasil validasi dan menambahkan barang ke *wishlist*.	Hasil yang keluar sesuai dengan hasil yang diharapkan

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.

			namun belum di *wishlist*. - Ada pada *wishlist*. - Tidak ada pada toko.			
8	Fitur WISHLIST SWAP	Memeriksa apakah fitur WISHLIST SWAP dapat menukar urutan *i* menjadi *j* dan urutan *j* menjadi *i*.	Memanggil fungsi Wishlist_swap() dan memberi indeks *i* dan *j* dengan kriteria: - *i* dan *j* valid - *i* atau/dan j tidak valid.	Data Test 8	Program berhasil memvalidasi dan menukar barang sesuai dengan indeks dari *user*.	Hasil yang keluar sesuai dengan hasil yang diharapkan
9	Fitur WISHLIST REMOVE <i>	Memeriksa apakah fitur WISHLIST REMOVE <i> dapat memvalidasi dan menghapus barang sesuai indeks.	Memanggil fungsi Wishlist_remove_i() dan memberi indeks *i* dengan kriteria: - *i* valid. - *i* tidak valid.	Data Test 9	Program berhasil memvalidasi dan menghapus barang dari *wishlist* sesuai indeks	Hasil yang keluar sesuai dengan hasil yang diharapkan
10	Fitur WISHLIST REMOVE	Memeriksa apakah fitur WISHLIST REMOVE dapat memvalidasi dan menghapus barang sesuai nama yang diinginkan	Memanggil fungsi Wishlist_remove() dan memberi input berupa nama barang dengan kriteria: - ada di *wishlist.* - tidak ada di *wishlist*.	Data Test 10	Program berhasil memvalidasi dan menghapus barang sesuai dengan nama dari *wishlist*.	Hasil yang keluar sesuai dengan hasil yang diharapkan
11	Fitur WISHLIST CLEAR	Memeriksa apakah fitur WISHLIST CLEAR dapat menghapus kosong *wishlist*.	Memanggil fungsi Wishlist_clear().	Data Test 11	Program berhasil membuat *wishlist* kosong.	Hasil yang keluar sesuai dengan hasil yang diharapkan

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.

12	Fitur WISHLIST SHOW	Memeriksa apakah fitur WISHLIST SHOW apakah dapat menampilkan *wishlist*	Memanggil fungsi Wishlist_show().	Data Test 12	Program berhasil menampilkan *wishlist*.	Hasil yang keluar sesuai dengan hasil yang diharapkan
13	Fitur STORE LIST	Memeriksa apakah fitur STORE LIST dapat menampilkan daftar store yang tersedia	Memanggil fungsi store_list().	Data Test 13	Program berhasil menampilkan *list* dari toko dengan rincian nama barang dan harganya.	Hasil yang keluar sesuai dengan hasil yang diharapkan
14	Fitur HELP	Memeriksa apakah fitur HELP berhasil ditampilkan	Memanggil fungsi Help().	Data Test 14	Program berhasil menampilkan panduan lengkap mengenai fitur-fitur di sistem.	Hasil yang keluar sesuai dengan hasil yang diharapkan

6. Pembagian Kerja dalam Kelompok

No	Fitur/ADT	NIM Coder	NIM Tester
1.	Main Program	18223045	18221021 18221027 18223045 18223057
2.	ADT User	18223045	18223045

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.

3.	ADT Stack	18223045	18223045
4.	ADT Setmap	18223045	18223045
5.	ADT Linked list	18223045	18223045
6.	PROFILE	18221027	18221027
7.	CART ADD <nama> <n>	18221021	18221021
8.	CART REMOVE <nama> <n>	18221021	18221021
9.	CART SHOW	18221021	18221021
10.	CART PAY	18221027	18221027
11.	HISTORY <n>	18221027	18221027
12.	WISHLIST ADD	18223057 18223045	18223057 18223045
13.	WISHLIST SWAP <i> <j>	18223057 18223045	18223057 18223045
14.	WISHLIST REMOVE <i>	18223057 18223045	18223057 18223045
15.	WISHLIST REMOVE	18223057 18223045	18223057 18223045
16.	WISHLIST CLEAR	18223057 18223045	18223057 18223045
17.	WISHLIST SHOW	18223057 18223045	18223057 18223045
18.	START	18223045	18223045
19.	LOAD	18223045	18223045
20.	SAVE	18223045	18223045

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.

21.	STORE LIST	18223057	18223057
22.	HELP	18223021	18223021
23.	Notulen	18221021 18221027	-
24.	Laporan	18221021 18221027 18223057	-

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.

7. Lampiran

7.1. Deskripsi Tugas Besar

Spesifikasi Umum

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:

- Menampilkan barang toko
- Meminta dan menyuplai barang baru ke toko
- Menyimpan dan membeli barang dalam keranjang
- Menampilkan barang yang sudah dibeli
- Membuat dan menghapus *wishlist*
- Bekerja untuk menghasilkan uang

2. Menu Program

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

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

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.

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

```
>> PROFILE
Nama : Purry
Saldo: 2000

(Silahkan kreasikan atribut yang ditampilkan)
// Kembali ke menu utama
```

b. CART ADD <nama> <n>

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

```
>> CART ADD AK47 20
Berhasil menambahkan 20 AK47 ke keranjang belanja!
// Kembali ke menu utama
```

```
>> CART ADD BebekKaliya 240
Barang tidak ada di toko!
// Perintah invalid; Kembali ke menu utama
```

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

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.

lebih sedikit dari N maka perintah akan gagal.

```
>> CART REMOVE AK47 10
Berhasil mengurangi 10 AK47 dari keranjang belanja!
// Kembali ke menu utama
```

```
>> CART REMOVE AK47 70
Tidak berhasil mengurangi, hanya terdapat 10 AK47 pada keranjang!
// Asumsi di keranjang belanja jumlah AK47 <70 sehingga perintah invalid; Kembali ke menu utama
```

```
>> CART REMOVE BintangSkibidi 70
Barang tidak ada di keranjang belanja!
// Asumsi tidak ada Bintang Skibidi di keranjang belanja; Kembali ke menu utama
```

d. CART SHOW

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

```
// Contoh dimana keranjang memiliki isi
>> CART SHOW
Berikut adalah isi keranjangmu.
Kuantitas  Nama    Total
2          AK47    20
1          Lalabu  10
Total biaya yang harus dikeluarkan adalah 30.

// Command mati; Kembali ke menu utama
```

```
// Contoh yang kosong
>> CART SHOW
Keranjang kamu kosong!
```

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

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.

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.

```
// Contoh pembayaran yang berhasil (Pengguna memasukan Ya)
>> CART PAY
Kamu akan membeli barang-barang berikut.
Kuantitas  Nama    Total
2          AK47    20
1          Lalabu  10
Total biaya yang harus dikeluarkan adalah 30, apakah jadi dibeli?
(Ya/Tidak): Ya

Selamat kamu telah membeli barang-barang tersebut!

// Command mati; Kembali ke main menu
```

```
// Contoh pembayaran yang gagal (Pengguna tidak memiliki uang yang cukup)
>> CART PAY
Kamu akan membeli barang-barang berikut.
Kuantitas  Nama    Total
2          AK47    20
1          Lalabu  20
Total biaya yang harus dikeluarkan adalah 40, apakah jadi dibeli?
(Ya/Tidak): Ya

Uang kamu hanya 15, tidak cukup untuk membeli keranjang!

// Command mati; Kembali ke main menu
```

```
// Contoh pembayaran yang gagal (Pengguna memasukan Tidak)
>> CART PAY
Kamu akan membeli barang-barang berikut.
Kuantitas  Nama    Total
2          AK47    20
1          Lalabu  10
```


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.

```
Total biaya yang harus dikeluarkan adalah 30, apakah jadi dibeli?
(Ya/Tidak): Tidak

// Command mati; Kembali ke main menu
```

```
// Contoh pembayaran yang gagal (Pengguna memasukan masukan aneh)
>> CART PAY
Kamu akan membeli barang-barang berikut.
Kuantitas  Nama    Total
2          AK47    20
1          Lalabu  10
Total biaya yang harus dikeluarkan adalah 30, apakah jadi dibeli?
(Ya/Tidak): Purry

// Command mati; Kembali ke main menu
```

```
// Contoh pembayaran yang gagal (Keranjang kosong)
>> CART PAY
Keranjang kamu kosong!
```

f. HISTORY <n>

HISTORY adalah *command* yang digunakan untuk menunjukan 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.

```
// Contoh menunjukan riwayat pembelian N < total riwayat
>> HISTORY 3
Riwayat pembelian barang:
1. AK47 40
2. AK47 100
3. Lalabu 35

// Command mati; Kembali ke main menu
```

```
// Contoh menunjukan riwayat pembelian N ≥ total riwayat
>> HISTORY 10
Riwayat pembelian barang:
```


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.

```
1. AK47 40
2. AK47 100
3. Lalabu 35
4. AK47 10
5. Meong 500
6. Ayam Goreng Crisbar 20

// Command mati; Kembali ke main menu
```

```
// Contoh riwayat pembelian kosong
Kamu belum membeli barang apapun!
```

g. WISHLIST ADD

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

```
>> WISHLIST ADD
Masukkan nama barang: Ayam Geprek Bakar Crispy Besthal

Berhasil menambahkan Ayam Geprek Bakar Crispy Besthal ke wishlist!
```

```
>> WISHLIST ADD
Masukkan nama barang: Ayam Geprek Bakar Crispy Besthal

Ayam Geprek Bakar Crispy Besthal sudah ada di wishlist!
```

```
>> WISHLIST ADD
Masukkan nama barang: Ayam Geprek Sambalado Besthal

Tidak ada barang dengan nama Ayam Geprek Sambalado Besthal!
```

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.

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.

```
>> WISHLIST SWAP 1 2
Berhasil menukar posisi Ayam Geprek Bakar Crispy Besthal dengan Ayam Mangut Besthal pada wishlist!
// Urutan Ayam Geprek Bakar Crispy Besthal berubah dari 1 menjadi 2. Sebaliknya, urutan Ayam Mangut Besthal berubah dari 2 menjadi 1
```

```
>> WISHLIST SWAP 1 2
Gagal menukar posisi Ayam Geprek Bakar Crispy Besthal!
// Hanya terdapat satu barang (Ayam Geprek Bakar Crispy Besthal) pada wishlist sehingga posisinya tidak dapat ditukar
```

i. WISHLIST REMOVE <i>

WISHLIST REMOVE adalah *command* yang digunakan untuk menghapus barang dengan posisi ke-i dari *wishlist*.

```
// Contoh menghapus barang ke-i dari WISHLIST
>> WISHLIST REMOVE 2
Berhasil menghapus barang posisi ke-2 dari wishlist!

// Command mati; Kembali ke main menu
```

```
// Contoh penghapusan wishlist yang gagal (Pengguna memasukkan urutan barang yang tidak ada)

//Misalnya dalam kasus ini, hanya terdapat 5 barang di dalam wishlist
>> WISHLIST REMOVE 10
Penghapusan barang WISHLIST gagal dilakukan, Barang ke-10 tidak ada di WISHLIST!

// Command mati; Kembali ke main menu
```

```
// Contoh penghapusan wishlist yang gagal (Wishlist kosong)

//Misalnya dalam kasus ini, tidak ada barang di dalam wishlist
>> WISHLIST REMOVE 1
Penghapusan barang WISHLIST gagal dilakukan, WISHLIST kosong!

// Command mati; Kembali ke main menu
```


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.

```
// Contoh penghapusan wishlist yang gagal (Pengguna memasukkan perintah yang tidak valid)
>> WISHLIST REMOVE XY
Penghapusan barang WISHLIST gagal dilakukan, command tidak valid!

// Command mati; Kembali ke main menu
```

j. WISHLIST REMOVE

WISHLIST REMOVE adalah *command* yang digunakan untuk menghapus barang dari wishlist berdasarkan nama barang yang dimasukkan pengguna.

```
// Contoh menghapus barang "Lalabu" dari WISHLIST
>> WISHLIST REMOVE
Masukkan nama barang yang akan dihapus : Lalabu
Lalabu berhasil dihapus dari WISHLIST!

// Command mati; Kembali ke main menu
```

```
// Contoh penghapusan wishlist yang gagal (Barang tidak ada di WISHLIST)

>> WISHLIST REMOVE
Masukkan nama barang yang akan dihapus : LoremIpsum
Penghapusan barang WISHLIST gagal dilakukan, LoremIpsum tidak ada di WISHLIST!

// Command mati; Kembali ke main menu
```

k. WISHLIST CLEAR

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

```
>> WISHLIST CLEAR
Wishlist telah dikosongkan.
```


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.

I. WISHLIST SHOW

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

```
>> WISHLIST SHOW
Berikut adalah isi wishlist-mu:
1 Ayam Geprek Bakar Crispy Besthal
2 Ayam Mangut Besthal
3 Karaage Don
4 Torikatsu Don
```

```
>> WISHLIST SHOW
Wishlist kamu kosong!
```

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.

b. STORE LIST

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

```
>> STORE LIST
List barang yang ada di toko :
- Platypus Laser - Harga: 100
- Shrink Ray - Harga: 500
- Net Shooter - Harga: 250
- Camouflage Cloak - Harga: 150
- Sleep Dart Gun - Harga: 300
- Bubble Blaster - Harga: 200
```

```
>> STORE LIST
TOKO KOSONG
```


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.

Konfigurasi Sistem

File konfigurasi akan dibaca saat memulai permainan. File ini menyimpan data-data yang disimpan ketika sistem dijalankan sebelumnya atau data-data *default*. Spesifikasi dari *file* konfigurasi adalah sebagai berikut:

1. Barisan pertama adalah bilangan bulat positif **N** yang menunjukan banyaknya barang di dalam sistem
2. Selanjutnya, sejumlah **N** baris menyatakan nama barang beserta harganya dengan format **<Harga barang> <Nama barang>**
3. Baris selanjutnya adalah bilangan bulat positif **M** yang menunjukan banyaknya pengguna di dalam sistem
4. Selanjutnya, terdapat data M buah pengguna dengan masing-masing spesifikasi berikut:
    a. Baris pertama adalah bilangan bulat positif **K** yang menunjukan banyaknya riwayat pengguna tersebut
    b. Selanjutnya, sejumlah **K** baris menyatakan nama barang beserta total biaya dengan format **<Total biaya> <Nama barang>**
    c. Baris selanjutnya adalah bilangan bulat positif **J** yang menunjukan banyaknya *wishlist* pengguna tersebut
    d. Selanjutnya, sejumlah **J** baris menyatakan nama barang dengan format **<Nama barang>**

Berikut adalah contoh file konfigurasi yang dimuat di awal sebuah *session* sebagai inisialisasi:

3	# Banyaknya barang di dalam toko
10 AK47	
20 Lalabu	
20 Ayam Goreng Crisbar	
500 Meong	
2	# Banyaknya pengguna di dalam program
100 user1 alstrukdatkeren	# Data pengguna “user1”
6	# Banyaknya riwayat pembelian pengguna “user1”
40 AK47	
100 AK47	
35 Lalabu	
10 AK47	
500 Meong	

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.

20 Ayam Goreng Crisbar 2 Ayam Goreng Crisbar AK47 25 user2 kerenbangetkak 0 1 Meong	 # Banyaknya *wishlist* pengguna “admin” # Data pengguna “user1” # Banyaknya riwayat pembelian pengguna “user2” # Banyaknya *wishlist* pengguna “user2”

Perlu diperhatikan bahwa antrian permintaan barang DAN keranjang tiap pengguna tidak disimpan di konfigurasi! Jika sistem dimulai, maka antrian dan keranjang akan dibuat lagi dari 0.

Daftar ADT

1. **ADT Kustom**
   Terdapat perubahan pada ADT pengguna (User). ADT akan menyimpan keranjang, riwayat pembelian, dan *wishlist* yang dimiliki oleh seorang user.

```
typedef struct {
  char name[MAX_LEN];
  char password[MAX_LEN];
  integer money;
  Map keranjang;
  Stack riwayat_pembelian;
  Linkedlist wishlist;
} User;
```

2. **ADT Stack**
   ADT ini digunakan untuk menyimpan riwayat pembelian seorang pengguna. Riwayat pembelian akan menyimpan nama barang dengan ketentuan di bawah dan total biaya seluruh barang.

   Nama barang yang disimpan 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

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.

lebih besar. Total biaya dari seluruh pembelian juga disimpan pada suatu elemen

3. **ADT Setmap**
   ADT ini digunakan untuk menyimpan keranjang pembelian seorang pengguna. Keranjang akan menyimpan nama beserta kuantitas barang yang ingin dibeli. Key elemen dalam setmap berupa barang yang pasti unik sementara value adalah kuantitas dari elemen tersebut.

4. **ADT Linked List**
   ADT ini digunakan untuk menyimpan *wishlist* seorang pengguna. Elemen yang disimpan di dalam *wishlist* adalah nama barang yang ingin dibeli.

Bonus

Pada tugas besar ini, terdapat beberapa fitur tambahan yang bisa diimplementasikan. Fitur-fitur ini tidak wajib untuk dikerjakan dan bobotnya lebih kecil dibanding fitur utama. **Utamakan fitur-fitur utama yang diminta sebelum mengerjakan bonus.** Berikut adalah penjelasan dari masing-masing fitur bonus:

1. **Store List Gacor**
   Pada iterasi sebelumnya, barang yang masuk ke toko harus bersifat **unique.** Sebelumnya, kalian menggunakan ADT List Dinamis untuk menyimpan barang-barang tersebut. Apakah terdapat ADT lain yang lebih efisien untuk mengimplementasikan fungsionalitas tersebut? Jika ada, ubah kode *store* kalian untuk menggunakan ADT-nya.
   Catatan: *Insertion* dan *deletion* dengan kompleksitas ~O(1) akan diberikan nilai tambahan.

2. **Riwayat Maksimal**
   ADT Stack digunakan untuk menyimpan riwayat pembelian seorang pengguna. Namun, data yang disimpan hanya nama barang dengan total harga terbesar. Petinggi OWCA ingin memiliki detail setiap pembelian ~~untuk keperluan *tax fraud*~~, mereka ingin *command* HISTORY untuk menunjukan seluruh barang yang dibeli.

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.

```
// Contoh menunjukan riwayat pembelian N < total riwayat
>> HISTORY 3
Riwayat pembelian barang:
Pembelian 1 - Total 40
Kuantitas  Nama    Total
2          AK47    20
1          Lalabu  20

Pembelian 2 - Total 100
Kuantitas  Nama    Total
8          AK47    80
1          Lalabu  20

Pembelian 3 - Total 35
Kuantitas  Nama    Total
2          M14     15
1          Lalabu  20

// Command mati; Kembali ke main menu
```

Agar detail riwayat pembelian *persistent*, terdapat juga perubahan pada konfigurasi file, yaitu:

a. Tidak terdapat **K** baris riwayat, tetapi elemen riwayat.
b. Baris pertama setiap elemen riwayat adalah bilangan positif tidak nol **L** yang menunjukan banyaknya barang yang dibeli dalam pembelian dan bilangan **X** yaitu total harga pada pembelian tersebut
c. Selanjutnya, sejumlah **L** baris menyatakan barang yang dibeli dengan format **<Total biaya> <Kuantitas barang> <Nama barang>**

Contoh konfigurasi baru adalah sebagai berikut.

`/** Konfigurasi barang toko */`	
`/** Konfigurasi User lainnya */`	
`100 user1 alstrukdatkeren`	`# Data user`
`6`	`# Banyaknya elemen riwayat pengguna`
`2 40`	`# Banyaknya riwayat-1 dan total`
`20 2 AK47`	
`20 1 Lalabu`	
`2 100`	`# Banyaknya riwayat-2 dan total`
`80 8 AK47`	
`20 1 Lalabu`	

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.

2 35 15 2 M14 20 1 Lalabu 1 10 10 1 AK47 1 500 500 1 Meong 1 20 20 1 Ayam Goreng Crisbar /** Konfigurasi wishlist user*/ /** Konfigurasi User lainnya */	# Banyaknya riwayat-3 dan total # Banyaknya riwayat-4 dan total # Banyaknya riwayat-5 dan total # Banyaknya riwayat-6 dan total

3. **Deteksi Kebocoran Senjata Biologis**

   Pada *milestone* sebelumnya, beberapa dari Anda telah sukses membuat sistem untuk mendeteksi kode pada DNA dari pabrik untuk mencegah sabotase musuh. Namun, OWCA mencurigai adanya kebocoran pada gudang PURRMART akibat penyimpanan yang tidak sesuai prosedur operasional baku (POB). Untuk menyelidiki hal tersebut, OWCA mencoba melakukan metagenomik untuk mengetahui spesies yang terdapat pada sampel dari lingkungan. Proses tersebut membutuhkan proses *sequence alignment* untuk mengecek kesamaan antara sekuens senjata biologis dengan sekuens hasil metagenomik. Oleh karena itu, Anda diminta mengimplementasikan kakas *global alignment* dengan algoritma Needleman-Wunsch. Apabila skor akhir lebih besar dari 80% panjang sekuens yang lebih panjang (jumlah karakter/basa nukleotida), maka dapat disimpulkan bahwa terjadi kebocoran senjata biologis. Panjang sekuens maksimum 50 karakter. Panjang kedua sekuens tidak harus sama, tetapi cukup mirip (misal 48 karakter dan 44 karakter). **Perhatikan kompleksitas algoritma, tidak boleh lebih dari ~O(2^n).**

   Ketentuan *scoring*:
   - *Match:* +1
   - *Mismatch:* 0
   - *Gap penalty:* -1

```
>> GLOBALALIGNMENT
Masukan sekuens referensi: TAGTAGAATGGGAGAGGTT // Panjang: 19 karakter
```


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.

```
Masukan sekuens query:     TAGTAGGGTTAATGTT // Panjang: 16 karakter

Skor: 9
Sekuens yang telah disejajarkan:
TAGTAGAATGGGAGAGGTT
TAGTAG---GGTTAATGTT

Woah! Tidak ada kebocoran (≧◡≦) // 19*80% = 15.20 > 9 (lebih rendah)
```

```
>> GLOBALALIGNMENT
Masukan sekuens referensi: TAGTAGAATGGGAGAGGC // Panjang: 18 karakter
Masukan sekuens query:     TAGTAGAATGGGTAAGTC // Panjang: 18 karakter

Skor: 15
Sekuens yang telah disejajarkan:
TAGTAGAATGGGAGAGGC
TAGTAGAATGGGTAAGTC

Nawh! Ada kebocoran... (ಥ﹏ಥ) // 18*80% = 14.4 < 15 (lebih tinggi)
```

Fun Fact: Di dunia nyata, penggunaan *global alignment* untuk menjajarkan hasil metagenomik dengan tujuan identifikasi organisme bersifat tidak lazim. Kakas seperti BLAST yang menerapkan *local alignment* lebih tepat dan lazim digunakan. Namun, implementasinya rumit sehingga tidak dipilih dalam tugas ini.

4. Optimasi Rute Ekspedisi

Toko PURRMART memiliki banyak klien sehingga perusahaan SiLambat, rekan toko PURRMART, membutuhkan cara untuk mengirim barang dengan rute yang paling efisien. Seluruh titik harus dikunjungi dan suatu titik ke titik lainnya memiliki jarak tertentu. Awalnya, salah satu karyawan PURRMART, menggunakan algoritma BFS untuk menyelesaikan permasalahan ini. Namun, ternyata algoritma tersebut tidak efisien dan memerlukan kemampuan komputasi yang besar. Anda diminta menggunakan algoritma alternatif yang lebih efisien untuk menyelesaikan masalah ini, semakin efisien semakin baik. **Jelaskan alasan pemilihan algoritmanya di laporan.**

```
>> OPTIMASIRUTE
Masukkan jumlah lokasi pengiriman (node): 4
Masukkan jumlah rute (edge): 5
Masukkan jarak antarlokasi (weight):
0 1 10
0 2 15
0 3 20
```


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.

```
1 2 35
1 3 25

Data diterima, silakan tunggu…

A-ha! Rute paling efektif adalah 0-2-3-1.
```

5. **Pencarian Barang**

   Toko PURRMART memiliki terlalu banyak barang sehingga kamu diminta untuk membantu membuat sebuah mesin pencari baru. Mesin pencari yang lama tidak efektif dikarenakan mesin tersebut mencari barang yang memiliki nama secara *exact match*. Karena manusia tempatnya salah dan lupa, barang yang dicari sering tidak ketemu karena nama barang yang dicari seringlah sekelumit berbeda. Oleh karena itu, kamu diminta untuk membuat mesin pencari kuasi *match* menggunakan jarak Levenshtein dengan *threshold distance default* adalah 10. *Threshold distance default* dapat diganti.

```
- Command akan diterima dengan format seperti berikut
- STORE FIND <nama-barang> --distance=<X> --ignore-casing
  --ignore-space
-  Flag - flag berupa --distance, --ignore-casing, --ignore-space
  bersifat opsional
- Flag --distance=X menyatakan untuk mengganti threshold default
  menjadi X
- Flag --ignore-casing untuk menganggap bahwa 2 character yang
  hanya berbeda dalam casingnya dapat dianggap sebagai huruf yang
  sama. Lowercase letter and uppercase letter dianggap sebagai
  huruf yang sama.
- Flag --ignore-space menyatakan bahwa spasi di dalam pencarian
  dapat di ignore pada semua string sehingga matching akan
  dilakukan antar 2 string yang tidak memiliki whitespace
- Urutan order dari command tidaklah strict. Ketiga command
  tersebut akan mengeluarkan hasil yang sama.
    - STORE FIND AK47 --distance=10 --ignore-casing
    - STORE FIND --distance10 AK47 --ignore-casing
    - STORE FIND --ignore-casing AK47 -distance10
```

```
Asumsi store sudah memiliki barang-barang ini
- Wobbulous X200 Supreme
- GrumblySnort Extreme
```


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.

```
   -  Zizzle Pop Max Plus
   -  Floombastic Ultra Boost
   -  WubbleTronix 3000 Deluxe
   -  Zizzle Zap UltraMax
```

```
>> STORE FIND Zarflob Supreme Edition
Barang tidak ditemukan.
```

```
>> STORE FIND Floompbastic Ultrablast
Berikut adalah barang yang mirip:
   -  Floombastic Ultra Boost
```

```
>> STORE FIND --distance=7 Zizzle Zap Plus
Berikut adalah barang yang mirip:
   -  Zizzle Pop Max Plus
   -  Zizzle Zap UltraMax
```

```
>> STORE FIND WobbulousX202Supreme --ignore-space
Berikut adalah barang yang mirip:
   -  Wobbulous X200 Supreme
```

```
>> STORE FIND --ignore-casing zizzle zap ultraamax
Berikut adalah barang yang mirip:
   -  Zizzle Zap UltraMax
```

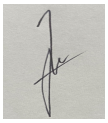

Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.

7.2. Notulen Rapat

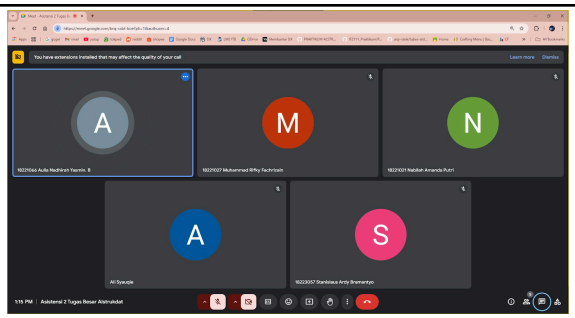
Form Asistensi Tugas Besar
IF2111/Algoritma dan Struktur Data STI
Sem. 1 2024/2025

No. Kelompok/Kelas : 07/01
Nama Kelompok : Kelompok 7
Anggota Kelompok (Nama/NIM) :
1. Nabilah Amanda Putri/18221021
2. Muhammad Rifky Fachrizain/18221027
3. Ali Syauqie/18223045
4. Stanislaus Ardy Bramantyo/1822305

Asistensi II

Tanggal : 18 Desember 2024	**Catatan Asistensi:** 1. Fungsi history bergantung pengguna mau menampilkan berapa, hanya menampilkan history terbaru aja 2. Untuk yang cart show, nampilin semua yang ada di cart + subtotal dari semua harganya
Tempat : https://meet.google.com/brq-cxbt-koe	
Kehadiran Anggota Kelompok: No NIM Tanda tangan 1 18221021 2 18221027 3 18223045 4 18223057 	

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.

	**Tanda Tangan Asisten:**

7.3. Log Activity Anggota Kelompok

No.	Tanggal	NIM	Nama	Aktivitas
1.	17/12/2024	18221027	Muhammad Rifky Fachrizain	Memperbaiki Login dan Register
2.	20/12/2024	18223045	Ali Syauqie	Mengerjakan ADT DoublyLinkedList, ADT Stack, ADT Map. dan ADT User
3.	20/12/2024	18223057	Stanislaus Ardy Bramantyo	Membuat *google docs* laporan dan mengisi bab ADT, Program Utama, TestScript, dan Lampiran
4.	21/12/2024	18221021	Nabilah Amanda Putri	Mengerjakan cart add, cart remove, dan cart show
5.	21/12/2024	18223057	Stanislaus Ardy Bramantyo	Upload semua fungsi *wishlist* (add, swap, remove, clear, dan show)
6.	22/12/2024	18223057	Stanislaus Ardy Bramantyo	Menambahkan *comment* ke semua fungsi *wishlist*. Revisi pada *wishlist* clear, , remove, show, dan swap.
7.	22/12/2024	18223057	Stanislaus Ardy Bramantyo	Revisi untuk format dari fungsi *print* dari ADT linked list.
8.	22/12/2024	18223057	Stanislaus Ardy Bramantyo	Upload penambahan spesifikasi pada

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.

				fungsi *store list*.
9.	22/12/2024	18223057	Stanislaus Ardy Bramantyo	Melaksanakan Testing dan menambahkan pada laporan bagian Data Test setiap fungsi *wishlist*.
10.	22/12/2024	18223057	Stanislaus Ardy Bramantyo	Menambahkan penjelasan tiap fungsi di ADT *linked list*.
11.	22/12/2024	18221021	Nabilah Amanda Putri	Memperbaiki cart remove, cart add, dan cart show
12.	22/12/2024	18221021	Nabilah Amanda Putri	Update fungsi help, Mesin Kata, Mesin Karakter. dan Set Map
13.	22/12/2024	18221021	Nabilah Amanda Putri	Mengerjakan laporan
14.	22/12/2024	18223045	Ali Syauqie	Update ADT Stack
15.	22/12/2024	18221027	Muhammad Rifky Fachrizain	Mengerjakan dan update cart pay, history, profile
16.	22/12/2024	18223045	Ali Syauqie	Mengerjakan main

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.