

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data STI

PURRMART


Dipersiapkan oleh:

Kelompok 05 Kelas K02

Florecita Natawirya	18223040
Kenlyn Tesalonika Winata	18223098
Nurul Na'im Natifah	18223106
Stevan Einer Bonagabe	18223028
Valereo Jibril Al Buchori	18223030

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2111-TB-02-05</i>		<i><jml hlm></i>
		<i>Revisi</i>	<i>0</i>	<i>25 Nov 2024</i>

Daftar Isi

1 Ringkasan	3
1.1 Deskripsi umum persoalan	4
1.2 Isi umum laporan	4
1.3 Kesimpulan hasil tugas besar	5
2 Tugas	5
2.1 Work	5
2.2 Store Request	6
3 Struktur Data (ADT)	6
3.1 ADT_List_Statis	6
3.2 ADT_List_Dinamis	6
3.3 ADT_Queue	7
3.4 ADT_Kustom	7
3.5 ADT_Mesin_Karakter	8
3.6 ADT_Mesin_Kata	8
4 Program Utama	9
4.1 ADT_Start and ADT_Quit	9
4.2 ADT_Login	9
4.3 ADT_Register	10
4.4 ADT_Load	10
4.5 ADT_Save	10
4.6 ADT_LogOut	10
4.7 WORK	10
4.8 WORK_CHALLENGE	10
4.8.1 Tebak Angka	11
4.8.2 WORDL3	11
4.9 STORE List	11
4.10 STORE Request	11
4.11 STORE Supply	11
4.12 STORE Remove	11
4.13 HELP	12
5 Data Test	12
5.1 STORE LIST	12
5.2 STORE REQUEST	13
5.3 STORE SUPPLY	15

5.4 STORE REMOVE	18
5.5 SAVE	19
5.5 LOAD	19
5.6 HELP	20
5.7 WORK CHALLENGE (Tebak Angka)	21
5.8 WORK CHALLENGE (WORDL3)	23
5.9 WORK	24
6 Test Script	26
7 Pembagian Kerja dalam Kelompok	30
8 Lampiran	31
8.1 Deskripsi Tugas Besar	31
8.2 Notulen Rapat	31
8.3 Log Activity Anggota Kelompok	33
8.4 Lampiran Daftar Gambar	34
Gambar 1. ADT LIST.H	34
Gambar 2. ADT LIST_DIN.H	37
Gambar 3. ADTQueue.H	41
Gambar 4. ADT Kustom.H	43
Gambar 5. ADT Mesin Karakter.h	44
Gambar 6 .ADT Mesin Kata.h	46

1 Ringkasan

1.1 Deskripsi Umum Persoalan

PURRMART adalah aplikasi simulasi berbasis CLI yang dirancang menggunakan bahasa C. Aplikasi ini mensimulasikan pengalaman berbelanja di e-commerce, dengan fitur seperti menampilkan barang toko, mengelola wishlist, bekerja untuk menghasilkan uang, dan melakukan transaksi pembelian barang. Sistem hanya menggunakan library standar C (stdio.h, stdlib.h, time.h, math.h), dan implementasinya memanfaatkan ADT khusus.

Untuk mengimplementasikan aplikasi PURRMART, kita harus membuat beberapa command/fungsi yang ada. Untuk pembuatan fungsi ini, akan dibagi menjadi 2 milestone, yaitu milestone 1 dan milestone 2. Untuk milestone 1 ini, akan dibuat beberapa fungsi, antara lain:

1. Main Menu (START-LOAD-HELP-QUIT-SAVE)
2. Login Menu (START-REGISTER-HELP)
3. Fitur Utama (WORK, WORK CHALLENGE, STORE LIST, STORE REQUEST, STORE SUPPLY, STOREE REMOVE, SAVE/QUIT)

Fungsi-Fungsi di atas akan diimplementasikan dengan fungsi-fungsi ADT. Ada beberapa Fungsi ADT primitif yang dapat digunakan untuk membuat fungsi utama, seperti ADT_List Statis, ADT_List Dinamis, ADT_Queue, ADT_Kustom, ADT_Mesin_Kata, ADT_Mesin_Karakter

1.2 Isi Umum Laporan

Laporan ini menjelaskan berbagai aspek penting dalam pembuatan aplikasi PURRMART, meliputi:

- **Ringkasan Tugas Besar:**
Laporan dimulai dengan memberikan gambaran umum tentang tujuan aplikasi PURRMART sebagai simulasi e-commerce berbasis CLI. Dijelaskan juga bagaimana aplikasi ini dirancang untuk memenuhi kebutuhan spesifikasi teknis dari tugas besar, yaitu penggunaan ADT khusus dan pengelolaan file konfigurasi.
- **Penjelasan Tambahan Spesifikasi Tugas:**
Secara mendetail, dijelaskan batasan dan fitur yang harus diimplementasikan dalam aplikasi ini.
- **Struktur Data (ADT):**
Penjelasan tentang ADT yang digunakan dalam aplikasi ini, termasuk struktur data User yang menyimpan informasi nama, password, dan uang pengguna, serta Barang yang menyimpan nama dan harga barang. Laporan ini juga mencakup ADT lain yang diintegrasikan ke dalam sistem.
- **Program Utama:**
Laporan menjelaskan alur utama aplikasi, mulai dari **main menu** dengan opsi START, LOAD, HELP, hingga login menu untuk LOGIN atau REGISTER. Setelah login,

pengguna dapat mengakses fitur-fitur seperti menampilkan daftar barang toko, bekerja untuk mendapatkan uang, menambah barang ke wishlist, dan membeli barang.

- **Data Test:**
Laporan menyajikan data uji yang digunakan untuk memastikan aplikasi berjalan sesuai harapan. Misalnya, daftar awal barang yang tersedia di toko, pengguna yang sudah terdaftar, dan beberapa skenario pengujian untuk memvalidasi fitur.
- **Test Script:**
Penjelasan tentang script pengujian yang digunakan untuk mengotomatisasi beberapa skenario pengujian, seperti login pengguna, simulasi supply barang, dan permainan worok_challenge. Hal ini membantu memastikan semua fitur aplikasi dapat berfungsi dengan baik.
- **Pembagian Kerja:**
Dalam laporan juga dijelaskan bagaimana tugas-tugas didelegasikan dalam tim, termasuk peran masing-masing anggota tim dalam pengembangan ADT, implementasi fitur utama, dan pengujian aplikasi.
- **Lampiran:**
Berisi file konfigurasi yang digunakan, kode lengkap dari aplikasi PURRMART, serta screenshot hasil pengujian aplikasi untuk mendokumentasikan keluaran dari berbagai skenario yang diuji.

Laporan ini dirancang untuk memberikan pemahaman yang menyeluruh tentang proses pengembangan, implementasi, dan pengujian aplikasi PURRMART sehingga pembaca dapat mengevaluasi kualitas dan kinerja dari tugas besar yang telah diselesaikan.

1.3 Kesimpulan Hasil Tugas Besar

Tugas besar ini menghasilkan aplikasi simulasi e-commerce yang interaktif, modular, dan memenuhi spesifikasi teknis. PURRMART memberikan pengalaman berbelanja yang realistis melalui berbagai fitur inovatif, mengintegrasikan pengelolaan data berbasis ADT, serta mendukung interaksi berbasis CLI yang intuitif. Aplikasi ini berhasil menunjukkan penguasaan konsep struktur data dan implementasi program berbasis file konfigurasi.

2 Tugas

2.1 Work

Pada bagian work terdapat modifikasi berupa ketika user memasukkan nilai 9090, maka akan memunculkan “Kode pekerjaan rahasia mata-mata telah dikonfirmasi... silakan tunggu sebentar, Tuan.” dan user akan mendapatkan pendapatan yang cukup banyak yaitu sebesar 99.999.999.

2.2 Store Request

Pada bagian Store Request. Ketika memasukkan barang baru kedalam store, store request dapat memasukkan lebih dari satu barang.

3 Struktur Data (ADT)

3.1 ADT_List_Statis

Sketsa Struktur Data:

ADT ini menggunakan array statis untuk menyimpan daftar pengguna. Setiap elemen array merepresentasikan satu entitas *User* yang berisi atribut yaitu nama, kata sandi, dan jumlah uang yang dimiliki. Kapasitas array ditentukan di awal, sehingga jumlah maksimum pengguna dibatasi oleh kapasitas tersebut.

Persoalan yang Diselesaikan:

Struktur ini digunakan untuk menyimpan dan mengakses data pengguna dengan efisiensi waktu yang tinggi. Dengan array statis, akses langsung ke data pengguna tertentu dimungkinkan menggunakan indeks, tanpa harus melihat kuncinya terlebih dahulu (HashMap).

Alasan Pemilihan:

Array statis dipilih karena aplikasi memiliki jumlah pengguna yang relatif tetap dan terbatas, sehingga tidak memerlukan fleksibilitas dalam kapasitas penyimpanan.

Berikut penjelasan dari fungsi yang berada pada ADT_List_Statis:

[Gambar 1. ADT LIST.H](#)

3.2 ADT_List_Dinamis

Sketsa Struktur Data:

ADT ini menggunakan array dinamis untuk menyimpan daftar barang di toko. Elemen array merepresentasikan *Barang* yang berisi nama barang dan harga. Ukuran array dapat berubah secara dinamis sesuai dengan jumlah barang yang ada.

Persoalan yang Diselesaikan:

Daftar barang di toko membutuhkan fleksibilitas karena jumlah barang dapat bertambah atau berkurang sesuai aktivitas pengguna, seperti menambah barang melalui STORE Supply atau menghapus barang dengan STORE Remove, dan melihat jumlah barang yang ada saat ini dengan STORE List.

Alasan Pemilihan:

Array dinamis dipilih untuk mengatasi persoalan kebutuhan memori yang variatif. Struktur ini memungkinkan alokasi ulang memori saat kapasitas array penuh atau berkurang. Keunggulannya terletak pada efisiensi dalam akses data melalui indeks sekaligus fleksibilitas dalam ukuran.

Berikut penjelasan dari fungsi yang berada pada ADT_List_Dinamis:

[Gambar ADT LIST_DIN.H](#)

3.3 ADT_Queue

Sketsa Struktur Data:

ADT_Queue menggunakan struktur *queue* berbasis array untuk menyimpan permintaan barang yang diajukan oleh pengguna. Elemen *queue* berisi nama barang yang diminta. Operasi utama dari ADT_Queue ini adalah *enqueue* untuk menambahkan permintaan baru dan *dequeue* untuk memproses permintaan yang paling awal.

Persoalan yang Diselesaikan:

Permintaan barang di toko harus diatur berdasarkan urutan pengajuan. Struktur *queue* menjamin bahwa barang yang diminta lebih dulu akan diproses lebih awal (first-in-first-out) sehingga dapat menggambarkan antrian barang.

Alasan Pemilihan:

Struktur *queue* dipilih karena sesuai dengan sifat antrian permintaan, di mana pemrosesan dilakukan secara berurutan. Pendekatan ini memastikan transparansi dan efisiensi dalam manajemen permintaan barang di toko. Selain itu, kami menggunakan *queue* tipe 3 untuk memperkecil adanya resiko *queue* yang penuh dan menyederhanakan kompleksitas algoritma yang digunakan.

Berikut penjelasan dari fungsi yang berada pada ADT_Queue:

[Gambar ADTQueue.H](#)

3.4 ADT_Kustom

Sketsa Struktur Data:

ADT_Kustom menyediakan dua struktur data khusus:

- **Barang:** Berisi atribut seperti nama barang (*char[]*) dan harga (*int*).
- **User:** Berisi atribut nama pengguna(*char[]*), kata sandi(*char[]*), dan jumlah uang (*int*).

Persoalan yang Diselesaikan:

ADT ini menyederhanakan representasi data entitas utama dalam aplikasi, yaitu pengguna dan barang. Dengan atribut yang terstruktur, data dapat dengan mudah diakses dan dimodifikasi.

Alasan Pemilihan:

Penggunaan ADT_Kustom membuat kode lebih mudah dikelola.

Berikut penjelasan dari fungsi yang berada pada ADT_Kustom:

[Gambar ADT Kustom.H](#)

3.5 ADT_Mesin_Karakter

Sketsa Struktur Data:

ADT ini bertugas membaca karakter dari file atau input tanpa menggunakan fungsi bawaan seperti *scanf* atau *getchar*. Mesin ini memiliki indikator akhir (*end-of-process/EOP*) yang disebut *MARK*. Operasi utama meliputi fungsi *start* untuk memulai pembacaan, *ADV* untuk pindah karakter berikutnya untuk dilakukan pembacaan kembali, dan *getCC* untuk mendapatkan karakter yang sedang diproses. Selain itu, terdapat fungsi *startFile* dan *ADVfile* untuk membaca karakter pada sebuah file.

Persoalan yang Diselesaikan:

Struktur ini mengatasi pembatasan penggunaan fungsi bawaan untuk membaca karakter dan digunakan secara luas dalam membaca file konfigurasi atau data lainnya.

Alasan Pemilihan:

Untuk menggantikan penggunaan fungsi bawaan secara langsung seperti *scanf* dan *getchar*.

Berikut penjelasan dari fungsi yang berada pada ADT_Mesin_Karakter:

[Gambar ADT Mesin Karakter.h](#)

3.6 ADT_Mesin_Kata

Sketsa Struktur Data:

ADT_Mesin_Kata menggunakan fungsi dari ADT_Mesin_Karakter untuk membaca kata demi kata dari aliran karakter. Mesin ini memiliki fungsi seperti *startKata* untuk memulai pembacaan, *ADVKata* untuk mendapatkan kata berikutnya, dan *copyKata* untuk mengakuisisi kata dan menyimpannya ke dalam *currentWord*. Selain itu, untuk membaca kata yang berada di dalam sebuah file akan digunakan *startFileKata* dan *ADVFileKata*.

Persoalan yang Diselesaikan:

ADT ini digunakan untuk memproses input yang terdiri atas kata-kata, seperti nama barang atau nama pengguna, yang tidak dapat diolah langsung oleh ADT_Mesin_Karakter. Fitur ini juga berguna saat membaca file konfigurasi yang memuat data berbasis teks.

Alasan Pemilihan:

Untuk menggantikan penggunaan fungsi bawaan secara langsung seperti *scanf* dan *getchar*.

Berikut penjelasan dari fungsi yang berada pada ADT_Mesin_Kata:

[Gambar ADT Mesin Kata.h](#)

4 Program Utama

Program utama dalam aplikasi PURRMART bertanggung jawab untuk mengatur alur aplikasi secara keseluruhan. Program ini memulai eksekusi dari menu utama, yang menyediakan opsi START untuk memulai aplikasi, LOAD untuk membaca data dari file konfigurasi, dan HELP untuk memberikan panduan kepada pengguna. Setelah itu, pengguna diarahkan ke menu login, di mana mereka dapat memilih opsi LOGIN atau REGISTER untuk melanjutkan, sebelum akhirnya dapat mengakses berbagai fitur e-commerce yang tersedia.

4.1 Start and Quit

Start bertugas mempersiapkan semua elemen yang dibutuhkan saat aplikasi dimulai. Fungsi ini mencakup inisialisasi data awal, seperti memuat daftar barang di toko atau membaca file konfigurasi jika ada. Berikut adalah spesifikasi Kodenya:

Quit digunakan untuk menutup aplikasi secara aman dengan membersihkan memori yang digunakan program, memastikan semua data sementara telah tersimpan jika diperlukan, dan menampilkan pesan keluar.

4.2 Login

Fungsi ini menangani proses autentikasi pengguna. Sistem akan memeriksa apakah nama dan kata sandi yang dimasukkan cocok dengan data pengguna yang tersimpan. Jika autentikasi berhasil, pengguna dapat melanjutkan ke aplikasi utama. Jika tidak, pesan kesalahan akan ditampilkan, dan pengguna diminta mencoba lagi.

4.3 Register

Register memungkinkan pengguna baru untuk mendaftar dengan memasukkan nama dan kata sandi. Setelah data pengguna baru ditambahkan, sistem secara otomatis menyimpan

data tersebut agar pengguna dapat login di lain waktu. Fungsi ini juga memeriksa apakah nama pengguna yang dimasukkan sudah terdaftar untuk menghindari duplikasi.

4.4 Load

Fungsi ini bertugas membaca file konfigurasi yang berisi data pengguna dan daftar barang toko. Load memastikan semua data yang dibaca diinterpretasikan dengan benar, seperti menciptakan struktur ADT dari informasi yang dimuat.

4.5 Save

Save digunakan untuk menyimpan data aplikasi saat ini ke dalam file, seperti informasi pengguna, daftar barang, dan status lainnya. Fungsi ini memastikan bahwa data disimpan dengan format yang sesuai, sehingga dapat dimuat kembali tanpa kehilangan informasi.

4.6 LogOut

Fungsi ini memungkinkan pengguna keluar dari akun mereka dan kembali ke menu login. LogOut memastikan semua perubahan yang dibuat oleh pengguna, seperti pembaruan pada wishlist atau keranjang belanja, telah disimpan.

4.7 WORK

Dalam fungsi WORK, akan diberikan beberapa pilihan pekerjaan dengan lama waktu dan total uang yang dapat diperoleh dari pekerjaan tersebut.

4.8 WORK_CHALLENGE

Dalam fungsi work_Challenge dibagi menjadi 2 fungsi utama yaitu tebak angka dan wordl3. Terdapat perbedaan antara work challenge dan work. work tidak memerlukan uang sebelum masuk kedalam pekerjaannya, sedangkan work-challenge memerlukan uang sebelum masuk kedalam pekerjaannya. Berikut spesifikasi kodenya:

4.8.1 Tebak Angka

Dalam mini-game ini, pengguna harus menebak angka acak dalam batas tertentu (10 tebakan). Sistem memberikan petunjuk apakah angka tebakan pengguna terlalu tinggi atau terlalu rendah hingga mereka menemukan angka yang benar.

4.8.2 WORDL3

WORDL3 adalah permainan menebak kata, di mana pengguna dapat menembak dalam batas tertentu (5 tebakan). Jika pengguna berhasil menebak, maka uangnya

akan bertambah, sedangkan jika gagal tidak memperoleh apa pun. Sistem akan memberikan petunjuk apakah huruf yang ia masukkan sudah tepat posisinya, sudah tepat, tetapi posisinya salah, atau masih salah.

4.9 STORE List

Fungsi STORE List menampilkan daftar barang yang tersedia di toko, termasuk nama barang dan harga masing-masing.

4.10 STORE Request

STORE Request memungkinkan pengguna meminta barang tertentu yang belum tersedia di toko. Barang yang diminta akan masuk ke antrian untuk diproses di kemudian hari.

4.11 STORE Supply

Fungsi STORE Supply digunakan untuk menambah barang ke dalam toko berdasarkan permintaan yang sebelumnya diajukan. Barang dari antrian akan ditambahkan ke daftar toko, sehingga bisa dilihat pada STORE List.

4.12 STORE Remove

STORE Remove memungkinkan penghapusan barang dari toko. Penghapusan barang dari toko ini membuat barang tidak akan terlihat lagi dalam STORE list.

4.13 HELP

Fungsi ini memberikan panduan kepada pengguna terkait cara menggunakan fitur-fitur aplikasi. Pengguna dapat mengakses informasi tentang fungsi apa saja yang dapat digunakan pada status ia saat ini. Status tersebut dibagi menjadi 3, yaitu saat program dimulai, saat sebelum user register/login, saat user telah melakukan login/register.

5 Data Test

5.1 STORE LIST

Fitur yang dites: STORE List
Hasil yang diharapkan:

Jumlah barang: 5

Data barang berhasil dibaca. PURRMART berhasil dijalankan.

Daftar Barang (5 item):

1. AK47 (Harga: 10)
2. Lalabu (Harga: 20)
3. Ayam (Harga: 20)
4. akupada (Harga: 12)
5. eamng (Harga: 5)

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: 1

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: abc

Input tidak valid! Harap masukkan angka antara 1 sampai 5.

Hasil yang ditampilkan:

```
Jumlah barang: 5
Data barang berhasil dibaca. PURRMART berhasil dijalankan.
```

```
Daftar Barang (5 item):
```

1. AK47 (Harga: 10)
2. Lalabu (Harga: 20)
3. Ayam (Harga: 20)
4. akupada (Harga: 12)
5. eamng (Harga: 5)

```
PURRMART Main Menu:
```

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

```
Masukkan pilihan Anda: 1
```

```
List barang yang ada di toko:
```

- AK47
- Lalabu
- Ayam
- akupada
- eamng

```
PURRMART Main Menu:
```

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

```
Masukkan pilihan Anda: abc
```

```
Input tidak valid! Harap masukkan angka antara 1 sampai 5.
```

```
Masukkan pilihan Anda: 
```

5.2 STORE REQUEST

Fitur yang dites: STORE Request

Hasil yang diharapkan:

```
PURRMART Main Menu:
```

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

```
Masukkan pilihan Anda: 2
```

```
Masukkan nama barang yang ingin diminta: Apel
```

```
Berhasil menambah Barang Apel ke antrian!
```

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: 2

Masukkan nama barang yang ingin diminta: Mangga

Berhasil menambah Barang Mangga ke antrian!

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: 2

Masukkan nama barang yang ingin diminta: Jeruk

Berhasil menambah Barang Jeruk ke antrian!

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: 2

Masukkan nama barang yang ingin diminta: Apel

Barang dengan nama yang sama sudah ada di antrian!

Hasil Yang Ditampilkan:

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 2
Masukkan nama barang yang ingin diminta: Apel
Berhasil menambah Barang Apel ke antrian!
```

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 2
Masukkan nama barang yang ingin diminta: Mangga
Berhasil menambah Barang Mangga ke antrian!
```

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 2
Masukkan nama barang yang ingin diminta: Jeruk
Berhasil menambah Barang Jeruk ke antrian!
```

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 2
Masukkan nama barang yang ingin diminta: Apel
Barang dengan nama yang sama sudah ada di antrian!
```

5.3 STORE SUPPLY

Fitur yang dites: STORE Supply

Hasil yang diharapkan:

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 3
```

Apakah kamu ingin menambahkan barang Apel: Terima
Harga barang: 10
Apel dengan harga 10 telah ditambahkan ke toko.

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: 1

List barang yang ada di toko:

- AK47
- Lalabu
- Ayam
- akupada
- eamng
- Apel

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: 3

Apakah kamu ingin menambahkan barang Mangga: Tunda
Mangga dikembalikan ke antrian.

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: 3

Apakah kamu ingin menambahkan barang Jeruk: Tolak
Jeruk dihapuskan dari antrian.

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: 3
Apakah kamu ingin menambahkan barang Mangga: Tentu
< Balik ke menu >

Hasil Yang Ditampilkan:

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 3
Apakah kamu ingin menambahkan barang Apel: Terima
Harga barang: 10
Apel dengan harga 10 telah ditambahkan ke toko.
```

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 1
List barang yang ada di toko:
- AK47
- Lalabu
- Ayam
- akupada
- eamng
- Apel
```

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 3
Apakah kamu ingin menambahkan barang Mangga: Tunda
Mangga dikembalikan ke antrian.
```

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 3
Apakah kamu ingin menambahkan barang Jeruk: Tolak
Jeruk dihapuskan dari antrian.
```

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 3
Apakah kamu ingin menambahkan barang Mangga: Tentu
< Balik ke menu >
```

5.4 STORE REMOVE

Fitur yang dites: STORE Remove

Hasil yang diharapkan:

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: 4

Nama barang yang akan dihapus: Apel

Apel telah berhasil dihapus.

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: 1

List barang yang ada di toko:

- AK47
- Lalabu
- Ayam
- akupada
- eamng

PURRMART Main Menu:

1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR

Masukkan pilihan Anda: 4

Nama barang yang akan dihapus: Durian

Toko tidak menjual Durian

Hasil yang ditampilkan:

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 4
Nama barang yang akan dihapus: Apel
Apel telah berhasil dihapus.
```

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 1
List barang yang ada di toko:
- AK47
- Lalabu
- Ayam
- akupada
- eamng
```

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 4
Nama barang yang akan dihapus: Durian
Toko tidak menjual Durian
```

```
PURRMART Main Menu:
1. STORE LIST
2. STORE REQUEST
3. STORE SUPPLY
4. STORE REMOVE
5. KELUAR
Masukkan pilihan Anda: 5
Apakah Anda yakin ingin keluar? (Y/N): Y
Keluar dari PURRMART. Sampai jumpa!
```

5.5 SAVE

Fitur yang dites: SAVE

Hasil yang diharapkan: (File .txt berhasil diubah sesuai variabel yang di save)

Data berhasil disimpan ke file default.txt.

Hasil yang ditampilkan:

Data berhasil disimpan ke file default.txt.

5.5 LOAD

Fitur yang dites: LOAD

Hasil yang diharapkan:

```
Jumlah barang: 4
Data barang berhasil dibaca. PURRMART berhasil dijalankan.

Daftar Barang (4 item):
1. M4 (Harga: 10)
2. APEL (Harga: 20)
3. Ayam (Harga: 20)
4. Goreng (Harga: 30)

Data user berhasil dibaca.

Daftar User (2 user):
1. admin (Money: 100, Password: alstrukdatkeren)
2. praktikan (Money: 25, Password: kerenbangetkak)
```

Hasil yang ditampilkan:

```
Jumlah barang: 4
Data barang berhasil dibaca. PURRMART berhasil dijalankan.

Daftar Barang (4 item):
1. M4 (Harga: 10)
2. APEL (Harga: 20)
3. Ayam (Harga: 20)
4. Goreng (Harga: 30)

Data user berhasil dibaca.

Daftar User (2 user):
1. admin (Money: 100, Password: alstrukdatkeren)
2. praktikan (Money: 25, Password: kerenbangetkak)
```

5.6 HELP

Fitur yang dites: HELP

Hasil yang diharapkan:

```
=====[ Welcome Menu Help PURRMART ]====

START -> Untuk masuk sesi baru

LOAD -> Untuk memulai sesi berdasarkan file konfigurasi
```

QUIT -> Untuk keluar dari program

=====[Login Menu Help PURRMART]=====

REGISTER -> Untuk melakukan pendaftaran akun baru

LOGIN -> Untuk masuk ke dalam akun dan memulai sesi

QUIT -> Untuk keluar dari program

=====[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

LOGOUT -> Untuk keluar dari sesi

SAVE -> Untuk menyimpan state ke dalam file

QUIT -> Untuk keluar dari program

Hasil yang ditampilkan:

```
=====[ Login Menu Help PURRMART ]=====
REGISTER -> Untuk melakukan pendaftaran akun baru
LOGIN -> Untuk masuk ke dalam akun dan memulai sesi
QUIT -> Untuk keluar dari program
```

```
=====[ Welcome Menu Help PURRMART ]=====
START -> Untuk masuk sesi baru
LOAD -> Untuk memulai sesi berdasarkan file konfigurasi
QUIT -> Untuk keluar dari program
```

```

===== [ 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
LOGOUT -> Untuk keluar dari sesi
SAVE -> Untuk menyimpan state ke dalam file
QUIT -> Untuk keluar dari program

```

5.7 WORK CHALLENGE (Tebak Angka)

Fitur yang dites: Tebak Angka

Hasil yang diharapkan:

```

>> WORK CHALLENGE
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main=200)
2. WORDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan: 1
Selamat datang di Tebak Angka!
Anda memiliki 10 kesempatan untuk menebak angka antara 1 -
100.
Biaya main: 200 rupiah telah dipotong dari akun Anda.
Tebak angka (1-100): 50
Tebakanmu lebih kecil!
Tebak angka (1-100): 1
Tebakanmu lebih kecil!
Tebak angka (1-100): 100
Tebakanmu lebih besar!
Tebak angka (1-100): 75
Tebakanmu lebih kecil!
Tebak angka (1-100): 90
Tebakanmu lebih besar!
Tebak angka (1-100): 77
Tebakanmu lebih kecil!
Tebak angka (1-100): 85
Tebakanmu benar! +380 rupiah telah ditambahkan ke akun
Anda.
ini saldo kamu 2180

```

Hasil yang ditampilkan:

```
>> WORK CHALLENGE
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main=200)
2. WORDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan: 1
Selamat datang di Tebak Angka!
Anda memiliki 10 kesempatan untuk menebak angka antara 1 - 100.
Biaya main: 200 rupiah telah dipotong dari akun Anda.
Tebak angka (1-100): 50
Tebakanmu lebih kecil!
Tebak angka (1-100): 1
Tebakanmu lebih kecil!
Tebak angka (1-100): 100
Tebakanmu lebih besar!
Tebak angka (1-100): 75
Tebakanmu lebih kecil!
Tebak angka (1-100): 90
Tebakanmu lebih besar!
Tebak angka (1-100): 77
Tebakanmu lebih kecil!
Tebak angka (1-100): 85
Tebakanmu benar! +380 rupiah telah ditambahkan ke akun Anda.
ini saldo kamu 2180
```

5.8 WORK CHALLENGE (WORDL3)

Fitur yang dites: Wordl3

Hasil yang diharapkan:

```
>> WORK CHALLENGE
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main=200)
2. WORDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan: 2
WELCOME TO WORDL3, YOU HAVE 6 CHANCES TO ANSWER BEFORE YOU
LOSE!

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
```

```
Masukan kata tebakan Anda: abuee
Hasil:
a* b u e% e
statusMenang: 0, attempts: 6

Masukan kata tebakan Anda: Abuse
Selamat Anda menang!!
+1500 rupiah telah ditambahkan kesaldo akun Anda.
SALDONYA SEGINI 3000

Saldo Anda saat ini: 3000 rupiah
```

Hasil yang ditampilkan:

```
>> WORK CHALLENGE
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main=200)
2. W0RDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan: 2
WELCOME TO W0RDL3, YOU HAVE 6 CHANCES TO ANSWER BEFORE YOU LOSE!

- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

Masukan kata tebakan Anda: abuee
Hasil:
a* b u e% e
statusMenang: 0, attempts: 6

Masukan kata tebakan Anda: Abuse
Selamat Anda menang!!
+1500 rupiah telah ditambahkan kesaldo akun Anda.
SALDONYA SEGINI 3000

Saldo Anda saat ini: 3000 rupiah
```

5.9 WORK

Fitur yang dites: WORK

Hasil yang diharapkan:

```
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 (1-5): 9090

Kode pekerjaan rahasia mata-mata telah dikonfirmasi...
 silakan tunggu sebentar, Tuan.

Pekerjaan selesai, +99999999 rupiah telah ditambahkan ke
 akun Anda.
 Saldo Anda saat ini: 99999999 rupiah

Apakah Anda ingin bekerja lagi? (y/n): y
 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 (1-5): 1

Anda sedang bekerja sebagai Evil Lab Assistant... harap
 tunggu.

Pekerjaan selesai, +100 rupiah telah ditambahkan ke akun
 Anda.
 Saldo Anda saat ini: 100000099 rupiah

Apakah Anda ingin bekerja lagi? (y/n): n

Terima kasih telah bekerja. Saldo akhir Anda: 100000099
 rupiah

Hasil yang ditampilkan:

```

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 (1-5): 9090

Kode pekerjaan rahasia mata-mata telah dikonfirmasi... silakan tunggu sebentar, Tuan.

Pekerjaan selesai, +99999999 rupiah telah ditambahkan ke akun Anda.
Saldo Anda saat ini: 99999999 rupiah

Apakah Anda ingin bekerja lagi? (y/n): y
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 (1-5): 1

Anda sedang bekerja sebagai Evil Lab Assistant... harap tunggu.

Pekerjaan selesai, +100 rupiah telah ditambahkan ke akun Anda.
Saldo Anda saat ini: 100000099 rupiah

Apakah Anda ingin bekerja lagi? (y/n): n

Terima kasih telah bekerja. Saldo akhir Anda: 100000099 rupiah

```

5.10 REGISTER

Fitur yang dites: REGISTER
 Hasil yang diharapkan:

```

Selamat datang di PURRMART!

=== MENU PURRMART ===
1. REGISTER
2. LOGIN
3. LOGOUT
4. EXIT
Pilih menu: 1

=== REGISTER ===
Username: flori
Password: cantik

```

Akun dengan username flori telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.

Hasil yang ditampilkan:

```
Selamat datang di PURRMART!

=== MENU PURRMART ===
1. REGISTER
2. LOGIN
3. LOGOUT
4. EXIT
Pilih menu: 1

=== REGISTER ===
Username: flori
Password: cantik
Akun dengan username flori telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.
```

5.11 LOGIN

Fitur yang dites: LOGIN

Hasil yang diharapkan:

```
=== MENU PURRMART ===
1. REGISTER
2. LOGIN
3. LOGOUT
4. EXIT
Pilih menu: 2

=== LOGIN ===
Username: flori
Password: cantik
Anda telah login ke PURRMART sebagai flori.
```

```
=== MENU PURRMART ===
1. REGISTER
2. LOGIN
3. LOGOUT
4. EXIT
Pilih menu: 2

=== LOGIN ===
Username: flori
Password: jelek
Username atau password salah.
```

Hasil yang ditampilkan:

```
=== MENU PURRMART ===
1. REGISTER
2. LOGIN
3. LOGOUT
4. EXIT
Pilih menu: 2

=== LOGIN ===
Username: flori
Password: cantik
Anda telah login ke PURRMART sebagai flori.
```

```
=== MENU PURRMART ===
1. REGISTER
2. LOGIN
3. LOGOUT
4. EXIT
Pilih menu: 2

=== LOGIN ===
Username: flori
Password: jelek
Username atau password salah.
```

5.12 LOGOUT

Fitur yang dites: LOGOUT

Hasil yang diharapkan:

```
=== MENU PURRMART ===
1. REGISTER
2. LOGIN
3. LOGOUT
4. EXIT
Pilih menu: 3

=== LOGOUT ===
Anda telah logout dari PURRMART.
```

Hasil yang ditampilkan:

```
=== MENU PURRMART ===
1. REGISTER
2. LOGIN
3. LOGOUT
4. EXIT
Pilih menu: 3

=== LOGOUT ===
Anda telah logout dari PURRMART.
```

6 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Store Request	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	Jumlah barang yang mau dimasukkan dan nama barang	Memasukkan barang yang diminta ke queue barang	Berhasil memasukkan barang yang diminta ke queue barang
2	Store List	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	-	Menampilkan daftar barang yang ada di store	berhasil menampilkan daftar barang yang ada di store
3	Store Remove	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	Nama barang yang mau dihapus	Menghapus barang yang diminta oleh user	berhasil Menghapus barang yang diminta oleh user
4	Store Supply	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	Terima, tunda, atau tolak barang untuk dimasukkan ke store	Memunculkan barang yang paling atas pada antrian lalu akan memberikan beberapa opsi perlakuan pada barang	Berhasil Memunculkan barang yang paling atas pada antrian lalu akan memberikan beberapa opsi perlakuan pada barang
5	WORK	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	Masukkan pilihan pekerjaan yang mau dilakukan	Pekerjaan akan dilakukan selama waktu yang ditentukan dan menghasilkan uang	Pekerjaan berhasil dilakukan selama waktu yang ditentukan dan menghasilkan uang
6	Tebak Angka	Untuk mengetahui apakah programnya sudah	Langsung run kode driver	Memasukkan tebakan angka (1-100)	Mengurangi uang pengguna , lalu Memberi tahu jika tebakan	Berhasil mengurangi uang pengguna , lalu berhasil Memberi tahu jika tebakan

		berjalan dengan baik			angkanya masih lebih besar/lebih kecil dari tebakan angka, serta memberi tahu jika tebakan sudah benar dan menambah uang pengguna	angkanya masih lebih besar/lebih kecil dari tebakan angka, serta berhasil memberi tahu jika tebakan sudah benar dan menambah uang pengguna
7	Wordl3	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	Memasukkan tebakan kata	Mengurangi uang pengguna, Memberi tahu jika tebakan kata sudah benar atau huruf berada di posisi yang salah atau huruf tidak ada dalam tebakan kata, lalu menambah uang pengguna	Berhasil Mengurangi uang pengguna, Berhasil Memberi tahu jika tebakan kata sudah benar atau huruf berada di posisi yang salah atau huruf tidak ada dalam tebakan kata, lalu Berhasil menambah uang pengguna
8	HELP	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	-	Menampilkan penjelasan terkait fungsi yang ada di menu tempat user berada	Menampilkan penjelasan terkait fungsi yang ada di menu tempat user berada
9	ADT- List statis	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	-	Dapat menghasilkan list of user dan fungsi yang berada pada list of user dapat berjalan dengan baik	Berhasil menghasilkan list of user dan fungsi yang berada pada list of user berjalan dengan baik
10	ADT- List Dinamis	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	-	Dapat menghasilkan list of barang dan fungsi yang berada pada list of barang dapat	Berhasil menghasilkan list of barang dan fungsi yang berada pada list of barang t

					berjalan dengan baik	berjalan dengan baik
11	ADT-Queue	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	-	Dapat menghasilkan queue of barang dan fungsi yang berada pada queue of barang dapat berjalan dengan baik	Berhasil menghasilkan queue of barang dan fungsi yang berada pada queue of barang berjalan dengan baik
12	ADT-Mesin kata & Mesin huruf	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	-	Dapat membaca hasil input baik secara langsung maupun dari file lain	Dapat membaca hasil input baik secara langsung maupun dari file lain
13	Save	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	nama file tempat menyimpan data	Dapat menyimpan data yang sudah diubah ke .txt	Berhasil menyimpan data yang sudah diubah ke .txt
14	Load	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	nama file untuk di load datanya	Dapat mengambil data dari suatu file .txt	Dapat mengambil data dari suatu file .txt
15	Register	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	Nama User dan Password	Dapat memasukkan pengguna baru ke dalam data .txt, dan menambah jumlah pengguna	Berhasil memasukkan pengguna baru ke dalam data .txt dan menambah jumlah pengguna
16	Login	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	Nama user dan Password	Mengganti suatu variabel "user" menjadi sesuai nama dan password yang di input	Berhasil Mengganti suatu variabel "user" menjadi sesuai nama dan password yang di input
17	LogOut	Untuk mengetahui apakah programnya sudah	Langsung run kode driver	-	Mengganti suatu variabel "user" menjadi sesuai undefined	Berhasil Mengganti suatu variabel "user" menjadi undefined

		berjalan dengan baik				
18	Start	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	-	Memulai aplikasi PURRMART	Berhasil memulai aplikasi PURRMART
19	Quit	Untuk mengetahui apakah programnya sudah berjalan dengan baik	Langsung run kode driver	-	Memulai aplikasi PURRMART	Berhasil memulai aplikasi PURRMART

7 Pembagian Kerja dalam Kelompok

Tugas		Peran
Tukang ngurus asistensi (chat kak asprak & nulis MoM)		Reo-18223030 ▾
Laporan		SEMUA ▾
User Interface		SEMUA ▾
Kerangka dan alur program utama		SEMUA ▾
ADT Wajib	ADT - Kustom	Nafa-18223106 ▾
	ADT - List (Array Statik)	Reo-18223030 ▾
	ADT - List (Array Dinamik)	Stevan Einer-18223028 ▾
	ADT - Queue	Florecita-18223040 ▾
	ADT-Mesinkata & mesin huruf	Kenlyn-18223098 ▾
User Management	ADT - Kustom	Nafa-18223106 ▾
	Register	
	Login	
	Logout	

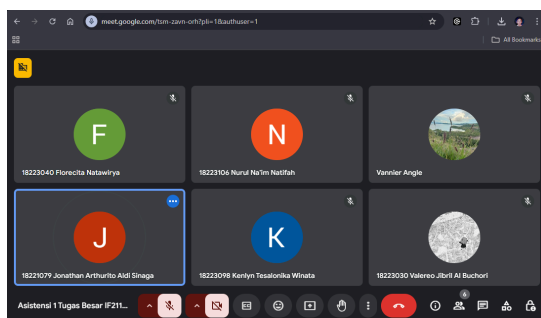
Tugas		Peran
	Quit	
Save & Load		Stevan Einer-18223028 ▾
WORK & WORK Challenge	WORK	Reo-18223030 ▾
	Tebak Angka	Reo-18223030 ▾
	WORDL3	Kenlyn-18223098 ▾
STORE	STORE LIST	Florecita-18223040 ▾
	STORE REQ	Stevan Einer-18223028 ▾
	STORE SUPPLY	Kenlyn-18223098 ▾
	STORE REMOVE	Florecita-18223040 ▾
	HELP	Reo-18223030 ▾
Konfigurasi Sistem		Reo-18223030 ▾

8 Lampiran




8.1 Deskripsi Tugas Besar




☰ Spesifikasi Tugas Besar 1 IF2111 2024/2025

8.2 Notulen Rapat



Moments of Meeting

Hari	Senin	Tanggal	19 November 2024
Kelas	K2	Kelompok	5
Waktu	21.00	Tempat	Online (Google Meet)
Anggota Kelompok	NIM	Nama	
	18223098	Kenlyn Tesalonika Winata 	
	18223030	Valereo Jibril Al Buchori 	
	18223106	Nurul Na'im Natifah 	
	18223028	Stevan Einer Bonagabe	

		
	18223040	Florencia Natawirya 
Nama Asisten	Jonathan Arthurito Aldi Sinaga 	

No.	Pertanyaan	Jawaban
1	“Untuk di dalam mesin karakter apakah diperbolehkan menggunakan scanf atau tidak sama sekali?”	“Harusnya tidak boleh sama sekali.”
2	“Kalau tidak boleh sama sekali berarti kita membuat pita kita sendiri?”	“Iya, kita akan membuat mesin karakter untuk semua input di dalam command”

No.	Pertanyaan	Jawaban
3	“Berarti inputnya diletakkan dalam pita karakter terlebih dahulu lalu dibaca?”	“Iya, seperti itu”
4	“Berarti semua command tidak diperbolehkan menggunakan scanf?”	“Iya,, semua command jika memerlukan input menggunakan mesin kata atau mesin karakter”
5	“Untuk Milestone 1 apakah sudah membuat main?”	“Iya, harusnya sudah dibuat”
6	“Apakah Printf tetap diperbolehkan?”	“Tetap boleh”
7	“Apakah wajib membuat file .c, .h, dan test file? lalu apakah di file h hanya deklarasi dan penjelasan fungsi?”	“file .c dan .h wajib, namun test file tidak wajib dan iya, di file .h hanya deklarasi fungsi dan penjelasannya, untuk implementasi di file .c”
8	“Apakah jika tidak mengerjakan bonus tetap dapat nilai penuh?”	“Iya, bonus hanya untuk memperbaiki nilai jika ada kesalahan di spesifikasi wajib.”
9	“Apakah jika kita login otomatis ke load, ataukah saat kita login hanya ke load username?”	“kita bakal load dulu baru login, jadi bisa saja banyak file .txt”
10	“Untuk konfigurasi, apakah display merupakan cara kita menyimpan di txt?”	“Ya, harusnya format menyimpan di txt seperti itu”
11	“Untuk pertanyaan apakah harus melalui perwakilan atau boleh langsung ke kakaknya?”	“Untuk pertanyaan yang spesifik ke kelompok boleh langsung ke saya, tapi jika pertanyaan teknis yang orang lain perlu tau juga bisa ditanyakan di sheets QnA”
12	“Bagian file save name apakah membuat txt baru atau pakai yang lama?”	“Dibebaskan”

8.3 Log Activity Anggota Kelompok

Berikut merupakan Log-Activity setiap anggota kelompok:

1. Log-Activity Stevan Einer Bonagabe -18223028:

Task Name	11-Nov-2024	13-Nov-2024	15-Nov-2024	17-Nov-2024	19-Nov-2024	21-Nov-2024	23-Nov-2024	24-Nov-2024	25-Nov-2024
Dibagikan Tubes									
Diskusi Pembagian Tubes									
Mengerjakan fungsi Load, Save, dan Store Request									
Mengerjakan laporan									
Menggabungkan fungsi yang telah dibuat									

2. Log-Activity Kenlyn Tesalonika Winata - 18223098:

Task Name	11-Nov-2024	13-Nov-2024	15-Nov-2024	17-Nov-2024	19-Nov-2024	21-Nov-2024	23-Nov-2024	24-Nov-2024	25-Nov-2024
Dibagikan Tubes									
Diskusi Pembagian Tubes									
Mengerjakan fungsi ADT_Mesin kata & Mesin karakter, Word13, dan Store Sunnly									
Mengerjakan laporan									
Menggabungkan fungsi yang telah dibuat									

3. Log-Activity Nurul Na'im Natifah - 18223106:

Task Name	11-Nov-2024	13-Nov-2024	15-Nov-2024	17-Nov-2024	19-Nov-2024	21-Nov-2024	23-Nov-2024	24-Nov-2024	25-Nov-2024
Dibagikan Tubes									
Diskusi Pembagian Tubes									
Mengerjakan fungsi User Management dan ADT_Kustom									
Mengerjakan laporan									
Menggabungkan fungsi yang telah dibuat									

4. Log-Activity Florecita Natawirya - 18223040:

Task Name	11-Nov-2024	13-Nov-2024	15-Nov-2024	17-Nov-2024	19-Nov-2024	21-Nov-2024	23-Nov-2024	24-Nov-2024	25-Nov-2024
Dibagikan Tubes									
Diskusi Pembagian Tubes									
Mengerjakan fungsi ADT_Queue, Store list & Remove									
Mengerjakan laporan									
Menggabungkan fungsi yang telah dibuat									

5. Log-Activity Valereo Jibril Al Buchori - 18223030:

Task Name	11-Nov-2024	13-Nov-2024	15-Nov-2024	17-Nov-2024	19-Nov-2024	21-Nov-2024	23-Nov-2024	24-Nov-2024	25-Nov-2024
Dibagikan Tubes									
Diskusi Pembagian Tubes									
Mengerjakan fungsi Help, tebak angka, WORK, ADT_list									
Mengerjakan laporan									
Menggabungkan fungsi yang telah dibuat									

8.4 Lampiran Daftar Gambar

Gambar 1. ADT LIST.H

```

#ifndef LIST_H
#define LIST_H

#include "ADT_Kustom.h"
#include "../boolean.h"

// Konstanta
#define Mark -1

// Struktur data list
typedef struct {
    User A[MAX_USERS]; // Array untuk menyimpan User
    int Count;          // Banyaknya elemen efektif dalam list
} List;

// ***** KONSTRUKTOR *****
void CreateList(List *L);
/* I.S. sembarang */
/* F.S. Terbentuk list L kosong dengan elemen efektif 0 */

// ***** TEST KOSONG / PENUH *****
boolean IsEmpty(List L);
/* Mengirimkan true jika list kosong */

boolean IsFull(List L);
/* Mengirimkan true jika list penuh */

// ***** OPERASI DASAR *****
int Length(List L);
/* Mengirimkan jumlah elemen efektif dalam list */

User Get(List L, int idx);
/* Prekondisi: idx adalah indeks valid dalam list */
/* Mengembalikan elemen User pada indeks ke-idx */

```

```

void Set(List *L, int idx, User U);
/* Mengubah elemen User pada indeks ke-idx dengan U */

// ***** SELEKTOR INDEKS *****
int IdxFirst(List L);
/* Mengirimkan indeks elemen pertama (0 jika list tidak kosong) */

int IdxLast(List L);
/* Mengirimkan indeks elemen terakhir (Count - 1 jika list tidak kosong) */

boolean IsIdxValid(int idx);
/* Mengirimkan true jika idx berada dalam rentang indeks array [0..MAX_USERS-1] */

boolean IsIdxEff(List L, int idx);
/* Mengirimkan true jika idx adalah indeks efektif dalam list */

// ***** OPERASI PENAMBAHAN *****
void InsertFirst(List *L, User U);
/* Menambahkan elemen baru di awal list */

void InsertLast(List *L, User U);
/* Menambahkan elemen baru di akhir list */

void InsertAt(List *L, User U, int idx);
/* Menambahkan elemen baru di indeks ke-idx */

// ***** OPERASI PENGHAPUSAN *****
void DeleteFirst(List *L);
/* Menghapus elemen pertama dari list */

void DeleteLast(List *L);
/* Menghapus elemen terakhir dari list */

void DeleteAt(List *L, int idx);
/* Menghapus elemen pada indeks ke-idx */

```

```

// ***** OPERASI PENCARIAN *****
int SearchInList(List L, Word key);
/* Mencari elemen dalam list berdasarkan nama menggunakan Mesin Kata.
 * IS: List terdefinisi dan kata key terdefinisi.
 * FS: Mengembalikan indeks elemen pertama yang cocok dengan key, atau -1 jika tidak ditemukan.
 */
#endif

```


Gambar 2. ADT LIST_DIN.H

```
/*          READMEEE
LIST INI DIGUNAKAN UNTUK MEMNYIMPAN LIST BARANG YANG ADA DI TOKO

*/
#ifndef LISTDIN_H
#define LISTDIN_H
#include "ADT_Kustom.h"
#include "boolean.h"

/* Kamus Umum */
#define IDX_MIN 0
/* Indeks minimum list */
#define IDX_UNDEF -1
/* Indeks tak terdefinisi*/

/* Definisi elemen dan koleksi objek */
typedef Barang ElType; /* type elemen list */
typedef int IdxType;
typedef struct
{
    ElType *buffer; /* memori tempat penyimpan elemen (container) */
    int nEff;       /* >=0, banyaknya elemen efektif */
    int capacity;   /* ukuran elemen */
} ListDin;

/* Indeks yang digunakan [0..capacity-1] */
/* Jika l adalah : ListDin, cara deklarasi dan akses: */
/* Deklarasi : l : ListDin */
/* Maka cara akses:
    l.nEff      untuk mengetahui banyaknya elemen
    l.buffer    untuk mengakses seluruh nilai elemen list
    l.buffer[i] untuk mengakses elemen ke-i */
/* Definisi :
    list kosong: l.nEff = 0
    Definisi elemen pertama : l.buffer[i] dengan i=0
    Definisi elemen terakhir yang terdefinisi: l.buffer[i] dengan i=l.capacity */
```

```

boolean isIdxEff(ListDin l, IdxType i);
/* Mengirimkan true jika i adalah indeks yang terdefinisi utk list */
/* yaitu antara 0..NEFF(l) */

/* ***** TEST KOSONG/PENUH ***** */
/* *** Test list kosong *** */
boolean isEmpty(ListDin l);
/* Mengirimkan true jika list l kosong, mengirimkan false jika tidak */
/* *** Test list penuh *** */
boolean isFull(ListDin l);
/* Mengirimkan true jika list l penuh, mengirimkan false jika tidak */

/* ***** BACA dan TULIS dengan INPUT/OUTPUT device ***** */
/* *** Mendefinisikan isi list dari pembacaan *** */
void readList(ListDin *l);
/* I.S. l sembarang dan sudah dialokasikan sebelumnya */
/* F.S. List l terdefinisi */
/* Proses : membaca banyaknya elemen l dan mengisi nilainya */
/* 1. Baca banyaknya elemen diakhiri enter, misalnya N */
/*   Pembacaan diulangi sampai didapat N yang benar yaitu 0 <= N <= CAPACITY(l) */
/*   Jika N tidak valid, tidak diberikan pesan kesalahan */
/* 2. Jika 0 < N <= CAPACITY(l); Lakukan N kali: Baca elemen mulai dari indeks
   |   0 satu per satu diakhiri enter */
/*   Jika N = 0; hanya terbentuk l kosong */
void printList(ListDin l);
/* Proses : Menuliskan isi list dengan traversal, list ditulis di antara kurung siku;
   |   antara dua elemen dipisahkan dengan separator "koma", tanpa tambahan karakter di depan,
   |   di tengah, atau di belakang, termasuk spasi dan enter */
/* I.S. l boleh kosong */
/* F.S. Jika l tidak kosong: [e1,e2,...,en] */
/* Contoh : jika ada tiga elemen bernilai 1, 20, 30 akan dicetak: [1,20,30] */
/* Jika list kosong : menulis [] */

/* ***** OPERATOR ARITMATIKA ***** */
/* *** Aritmatika list : Penjumlahan, pengurangan, perkalian, ... *** */

```

```

ListDin plusMinusList(ListDin l1, ListDin l2, boolean plus);
/* Prekondisi : l1 dan l2 memiliki Neff sama dan tidak kosong */
/* Jika plus = true, mengirimkan l1+l2, yaitu setiap elemen l1 dan l2 pada indeks yang sama dijumlahkan */
/* Jika plus = false, mengirimkan l1-l2, yaitu setiap elemen l1 dikurangi elemen l2 pada indeks yang sama */

/* ***** OPERASI LAIN ***** */
void copyList(ListDin lIn, ListDin *lOut);
/* I.S. lIn terdefinisi tidak kosong, lOut sembarang */
/* F.S. lOut berisi salinan dari lIn (identik, nEff dan capacity sama) */
/* Proses : Menyalin isi lIn ke lOut */
ElType sumList(ListDin l);
/* Menghasilkan hasil penjumlahan semua elemen l */
/* Jika l kosong menghasilkan 0 */
int countVal(ListDin l, ElType val);
/* Menghasilkan berapa banyak kemunculan val di l */
/* Jika l kosong menghasilkan 0 */

/* ***** SORTING ***** */
void sort(ListDin *l, boolean asc);
/* I.S. l boleh kosong */
/* F.S. Jika asc = true, l terurut membesar */
/*      Jika asc = false, l terurut mengecil */
/* Proses : Mengurutkan l dengan salah satu algoritma sorting,
            algoritma bebas */

/* ***** MENAMBAH DAN MENGHAPUS ELEMEN DI AKHIR ***** */
/* *** Menambahkan elemen terakhir *** */
void insertLast(ListDin *l, ElType val);
/* Proses: Menambahkan val sebagai elemen terakhir list */
/* I.S. List l boleh kosong, tetapi tidak penuh */
/* F.S. val adalah elemen terakhir l yang baru */

```

```

/* ***** MENGHAPUS ELEMEN ***** */
void deleteLast(ListDin *l, ElType *val);
/* Proses : Menghapus elemen terakhir list */
/* I.S. List tidak kosong */
/* F.S. val adalah nilai elemen terakhir l sebelum penghapusan, */
/*      Banyaknya elemen list berkurang satu */
/*      List l mungkin menjadi kosong */

/* ***** MENGUBAH UKURAN ARRAY ***** */
void expandList(ListDin *l, int num);
/* Proses : Menambahkan capacity l sebanyak num */
/* I.S. List sudah terdefinisi */
/* F.S. Ukuran list bertambah sebanyak num */

void shrinkList(ListDin *l, int num);
/* Proses : Mengurangi capacity sebanyak num */
/* I.S. List sudah terdefinisi, ukuran capacity > num, dan nEff < capacity - num. */
/* F.S. Ukuran list berkurang sebanyak num. */

void compressList(ListDin *l);
/* Proses : Mengubah capacity sehingga nEff = capacity */
/* I.S. List tidak kosong */
/* F.S. Ukuran nEff = capacity */

#endif

```

Gambar 3. ADTQueue.H

```
#ifndef QUEUE_H
#define QUEUE_H

#include "boolean.h"
#include "ADT_Kustom.h"
#define IDX_UNDEF -1
#define KAPASITAS 100

/* Definisi elemen dan address */
typedef Barang EType;
typedef struct {
    EType buffer[KAPASITAS];
    int idxHead;
    int idxTail;
} Queue;

/* ***** AKSES (Selektor) ***** */
/* Jika q adalah Queue, maka akses elemen : */
#define IDX_HEAD(q) (q).idxHead
#define IDX_TAIL(q) (q).idxTail
#define HEAD(q) (q).buffer[(q).idxHead]
#define TAIL(q) (q).buffer[(q).idxTail]

/* *** Kreator *** */
void CreateQueue(Queue *q);
/* I.S. sembarang */
/* F.S. Sebuah q kosong terbentuk dengan kondisi sbb: */
/* - Index head bernilai IDX_UNDEF */
/* - Index tail bernilai IDX_UNDEF */
/* Proses : Melakukan alokasi, membuat sebuah q kosong */

/* ***** Prototype ***** */
boolean isEmpty(Queue q);
/* Mengirim true jika q kosong: lihat definisi di atas */
```

```

boolean isPenuh(Queue q);
/* Mengirim true jika tabel penampung elemen q sudah penuh */
/* yaitu IDX_TAIL akan selalu di belakang IDX_HEAD dalam buffer melingkar*/

int length(Queue q);
/* Mengirimkan banyaknya elemen queue. Mengirimkan 0 jika q kosong. */

/* *** Primitif Add/Delete *** */
void enqueue(Queue *q, ElType val);
/* Proses: Menambahkan val pada q dengan aturan FIFO */
/* I.S. q mungkin kosong, tabel penampung elemen q TIDAK penuh */
/* F.S. val menjadi TAIL yang baru, IDX_TAIL "mundur" dalam buffer melingkar. */

void dequeue(Queue *q, ElType *val);
/* Proses: Menghapus val pada q dengan aturan FIFO */
/* I.S. q tidak mungkin kosong */
/* F.S. val = nilai elemen HEAD pd I.S., IDX_HEAD "mundur";
   q mungkin kosong */

/* *** Display Queue *** */
void displayQueue(Queue q);
/* Proses : Menuliskan isi Queue dengan traversal, Queue ditulis di antara kurung
   siku; antara dua elemen dipisahkan dengan separator "koma", tanpa tambahan
   karakter di depan, di tengah, atau di belakang, termasuk spasi dan enter */
/* I.S. q boleh kosong */
/* F.S. Jika q tidak kosong: [e1,e2,...,en] */
/* Contoh : jika ada tiga elemen bernilai 1, 20, 30 akan dicetak: [1,20,30] */
/* Jika Queue kosong : menulis [] */

#endif

```

Gambar 4. ADT Kustom.H

```
#ifndef Kustom
#define Kustom

/* ***** Definisi Konstanta ***** */
#define MAX_ITEMS 100 /* Batas maksimum jumlah barang yang dapat disimpan */
#define MAX_USERS 50 /* Batas maksimum jumlah pengguna yang dapat terdaftar */
#define MAX_STR 100 /* Panjang maksimum string untuk input/output umum */
#define MAX_LEN 100 /* Panjang maksimum nama dan password */

/* ***** Definisi Tipe Data User ***** */
/* Struktur untuk menyimpan data pengguna */
typedef struct {
    char name[MAX_LEN]; /* Nama pengguna (maksimum MAX_LEN karakter) */
    char password[MAX_LEN]; /* Kata sandi pengguna (maksimum MAX_LEN karakter) */
    int money; /* Saldo uang yang dimiliki pengguna */
} User;

/* ***** Definisi Tipe Data Barang ***** */
/* Struktur untuk menyimpan data barang */
typedef struct {
    char name[MAX_LEN]; /* Nama barang (maksimum MAX_LEN karakter) */
    int price; /* Harga barang */
} Barang;
#endif
```

Gambar 5. ADT Mesin Karakter.h

```
#ifndef __MESIN_KAR_H_
#define __MESIN_KAR_H_

#include "../boolean.h"

#define MARK ';'
#define ENTER "\n"

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

void start();
/* Mesin siap dioperasikan dan pita disiapkan untuk dibaca.
   Karakter pertama merupakan karakter yang berada di jendela.
   Pita akan direset sebelum pembacaan dari stdin.
   I.S. : sembarang
   F.S. : currentChar adalah karakter pertama pada pita
   Jika currentChar != MARK maka EOP maka fungsi akan mengembalikan false
   Jika currentChar = MARK maka EOP maka fungsi akan mengembalikan true */

void ADV();
/* Pita dimajukan satu karakter.
   I.S. : Karakter pada jendela = currentChar, currentChar != MARK
   F.S. : currentChar yang baru adalah karakter berikutnya dari currentChar yang lama,
   currentChar mungkin = MARK
   Jika currentChar = MARK maka EOP akan bernilai true */
```



```

void startFile(char *file_name);
/* Menyiapkan file untuk dibuka dan dibaca
   I.S. : sembarang
   F.S. : Jika file berhasil dibuka maka success menjadi true
         | Jika file tidak ditemukan atau aksesnya ditolak maka success menjadi false */
/* Mesin siap dioperasikan dan pita disiapkan untuk dibaca.
   Pita yang akan dibaca berada di dalam file dengan path ../data/file_name.
   Pita yang dibaca diambil dari fopen(path).
   I.S. : sembarang
   F.S. : currentChar adalah karakter pertama pada pita
         | Jika currentChar != MARK maka EOP maka fungsi akan mengembalikan false
         | Jika currentChar = MARK maka EOP maka fungsi akan mengembalikan true */

void ADVFile();

char getCC();
/* Mengirimkan currentChar */

boolean isEOP();
/* Mengirimkan true jika currentChar = MARK */

#endif

```

Gambar 6 .ADT Mesin Kata.h

```
#ifndef __MESINKATA_H__
#define __MESINKATA_H__

#include "boolean.h"
#include "ADT Mesin Karakter.h"

#define NMax 100
#define BLANK ' '

typedef struct
{
    char TabWord[NMax]; /* container penyimpan kata, indeks yang dipakai [0..NMax-1] */
    int Length;
} Word;

/* State Mesin Kata */
extern boolean EndWord;
extern Word currentWord;

void resetCurrentKata();
/* Mengembalikan currentWord ke state awal (kosong / length=0).
   I.S. : currentWord sembarang
   F.S. : currentWord kosong atau length-nya = 0 */

void ignoreBlanks();
/* Mengabaikan satu atau beberapa BLANK.
   I.S. : currentChar sembarang
   F.S. : currentChar ≠ BLANK atau currentChar = MARK */

void startKata();
/* I.S. : currentChar sembarang
   F.S. : EndWord = true, dan currentChar = MARK;
          atau EndWord = false, currentWord adalah kata yang sudah diakuisisi,
          currentChar karakter pertama sesudah karakter terakhir kata */
```

```

void ADVKata();
/* I.S. : currentChar adalah karakter pertama kata yang akan diakuisisi
   F.S. : currentWord adalah kata terakhir yang sudah diakuisisi,
   |     | currentChar adalah karakter pertama dari kata berikutnya, mungkin MARK
   |     | Jika currentChar = MARK, EndWord = true.
   Proses : Akuisisi kata menggunakan procedure SalinWord */

void copyKata();
/* Mengakuisisi kata, menyimpan dalam currentWord.
   I.S. : currentChar adalah karakter pertama dari kata
   F.S. : currentWord berisi kata yang sudah diakuisisi;
   |     | currentChar = BLANK atau currentChar = MARK;
   |     | currentChar adalah karakter sesudah karakter terakhir yang diakuisisi.
   |     | Jika panjang kata melebihi NMax, maka sisa kata "dipotong" */

void startFileKata(char* file_name);
/* Kata dibaca dengan prosedur STARTFILE() yang akan membaca dari file pada path,
   akuisisi kata menggunakan copySentence.
   I.S. : currentChar sembarang
   F.S. : EndWord = true, dan currentChar = MARK;
   |     | atau EndWord = false, currentWord adalah kata yang sudah diakuisisi,
   |     | currentChar karakter pertama sesudah karakter terakhir kata */

void ADVFileKata();
/* I.S. : currentChar adalah karakter pertama kata yang akan diakuisisi
   F.S. : currentWord adalah kata terakhir yang sudah diakuisisi,
   |     | currentChar adalah karakter pertama dari kata berikutnya, mungkin MARK
   |     | Jika currentChar = MARK, EndWord = true.
   Proses : Akuisisi kata menggunakan procedure copySentence */

```

```

void copySentence();
/* Mengakuisisi kata dengan membolehkan blanks, menyimpan dalam currentWord.
   I.S. : currentChar adalah karakter pertama dari kata
   F.S. : currentWord berisi kata yang sudah diakuisisi;
   |
   | currentChar = BLANK atau currentChar = MARK;
   | currentChar adalah karakter sesudah karakter terakhir yang diakuisisi.
   | Jika panjang kata melebihi NMax, maka sisa kata "dipotong" */

void startLine();
/* Membaca satu line dari user, dengan membolehkan spasi.
   I.S. : currentChar sembarang
   F.S. : EndWord = true, dan currentChar = MARK;
   |
   | atau EndWord = false, currentWord adalah kata yang sudah diakuisisi,
   | currentChar karakter pertama sesudah karakter terakhir kata */

boolean isKataEqual(word kata1, word kata2);
/* Mengembalikan true bila Word kata1 sama dengan Word kata2. */

boolean isKataInt(word kata);
/* Mengembalikan true apabila seluruh isi Word berupa numerik */

void displayKata(word kata, boolean newLine);
/* Menampilkan isi Word ke layar.
   |
   | I.S. : sembarang
   | F.S. : Seluruh isi dalam Word telah ditampilkan pada layar */

int kataToInt(word kata);
/* Mengembalikan nilai integer dari Word kata */

#endif

```