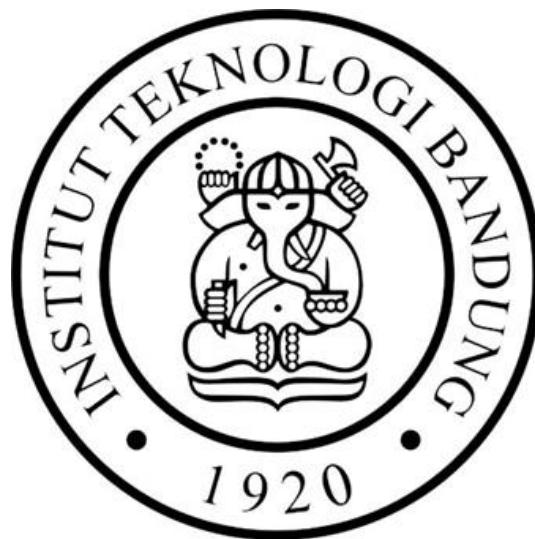


LAPORAN TUGAS BESAR
IF1210 DASAR PEMROGRAMAN

Kelas Mahasiswa (K-11) / Kelompok 07

Dosen: Rizal Dwi Prayogo, S.Si., M.Si., M.Sc.



Disusun Oleh:

M. Andiananto Bismaramadhan	16521499
Fariz Putra Hanggara	16521505
Nabilah Amanda Putri	16521511
Jason Rivalino	16521541

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2022

Pernyataan Kelompok

“Saya menyatakan bahwa saya mengerjakan tugas besar ini dengan sejujur-jujurnya, tanpa menggunakan cara yang tidak dibenarkan. Apabila di kemudian hari diketahui saya mengerjakan tugas besar ini dengan cara yang tidak jujur, saya bersedia mendapatkan konsekuensinya, yaitu mendapatkan nilai E pada mata kuliah IF1210 Dasar Pemrograman Semester 2 2021/2022”

1. Nama : M. Andiananto Bismaramadhan
NIM : 16521499
2. Nama : Fariz Putra Hanggara
NIM : 16521505
3. Nama : Nabilah Amanda Putri
NIM : 16521511
4. Nama : Jason Rivalino
NIM : 16521541

Daftar Isi

Pernyataan Kelompok	i
Daftar Isi	ii
Daftar Tabel	iii
Daftar Gambar	iv
A. Deskripsi Persoalan.....	1
B.Daftar Pembagian Kerja Anggota Kelompok	2
Table B.1 Daftar Pembagian Tugas Anggota Kelompok pada Source Code.....	2
Tabel B.2 Daftar Pembagian Tugas Anggota Kelompok pada Laporan.....	4
C.Checklist Hasil Rancangan, Implementasi, dan Testing Setiap Primitif.....	6
Tabel C.1 Checklist Hasil Rancangan, Implementasi, dan Testing Setiap Primitif.....	6
D.Desain Command untuk Setiap Primitif	7
E.Desain Kamus Data	11
F.Desain Dekomposisi Algoritmik dan Fungsional Program	15
G.Spesifikasi untuk Setiap Modul/Fungsi/Prosedur yang Dibuat	32
H.Struktur Data File Eksternal.....	37
I. Screenshot Hasil Pengujian Program Berdasarkan Fitur – Fitur pada Spesifikasi.....	56
Lampiran	64
Form MoM Asistensi 1	65
Form MoM Asistensi 2	67

Daftar Tabel

Tabel B.1 Daftar Pembagian Tugas Anggota Kelompok pada Source Code.....	2
Tabel B.2 Daftar Pembagian Tugas Anggota Kelompok pada Laporan.....	4
Tabel C.1 Checklist Hasil Rancangan, Implementasi, dan Testing Setiap Primitif.....	6

Daftar Gambar

Gambar F.1 Dekomposisi Register.....	24
Gambar F.2 Dekomposisi Login.....	25
Gambar F.3 Dekomposisi Menambah Game.....	25
Gambar F.4 Dekomposisi Mengubah Game.....	26
Gambar F.5 Dekomposisi Mengubah Stok Game.....	26
Gambar F.6 Dekomposisi Listing Game pada Toko.....	27
Gambar F.7 Dekomposisi Membeli Game.....	27
Gambar F.8 Dekomposisi Melihat Game yang dimiliki.....	28
Gambar F.9 Dekomposisi Mencari Game yang dimiliki.....	28
Gambar F.10 Dekomposisi Mencari Game di Toko.....	29
Gambar F.11 Dekomposisi Top-Up Saldo.....	29
Gambar F.12 Dekomposisi Melihat Riwayat Pembelian.....	30
Gambar F.13 Dekomposisi Help.....	30
Gambar F.14 Dekomposisi Load.....	31
Gambar F.15 Dekomposisi Save.....	31
Gambar F.16 Dekomposisi Exit.....	31
Gambar I.1 Screenshot Hasil Pengujian Load.....	56
Gambar I.2 Screenshot Hasil Pengujian Register.....	56
Gambar I.3 Screenshot Hasil Pengujian Register Apabila Username Sudah Terpakai.....	56
Gambar I.4 Screenshot Hasil Pengujian Login Salah.....	56
Gambar I.5 Screenshot Hasil Pengujian Login Benar.....	56
Gambar I.6 Screenshot Hasil Pengujian Menambah Game ke Toko Game Apabila Data Tidak Lengkap (Gagal).....	56
Gambar I.7 Screenshot Hasil Pengujian Menambah Game ke Toko Game Berhasil.....	57

Gambar I.8 Screenshot Hasil Pengujian Mengubah Game pada Toko Game saat ID Tidak Dimasukkan.....	57
Gambar I.9 Screenshot Hasil Pengujian Mengubah Game pada Toko Game Berhasil.....	57
Gambar I.10 Screenshot Hasil Pengujian Mengubah Stok Game di Toko Apabila Stok Kurang.....	57
Gambar I.11 Screenshot Hasil Pengujian Mengubah Stok Game di Toko Apabila Tidak Ada Game dengan ID Tersebut.....	57
Gambar I.12 Screenshot Hasil Pengujian Mengubah Stok Game di Toko Berhasil Dikurangi.....	58
Gambar I.13 Screenshot Hasil Pengujian Mengubah Stok Game di Toko Berhasil Ditambah.....	58
Gambar I.14 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Tidak Valid.....	58
Gambar I.15 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Kosong.....	58
Gambar I.16 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Harga dari Terkecil.....	58
Gambar I.17 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Harga dari Terbesar.....	59
Gambar I.18 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Tahun dari Terkecil.....	59
Gambar I.19 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Tahun dari Terbesar.....	59
Gambar I.20 Screenshot Hasil Pengujian Membeli Game.....	59
Gambar I.21 Screenshot Hasil Pengujian Membeli Game Apabila Game Sudah Dimiliki.	59
Gambar I.22 Screenshot Hasil Pengujian Membeli Game Apabila Saldo Tidak Cukup....	60
Gambar I.23 Screenshot Hasil Pengujian Membeli Game Apabila Stok Game Habis.....	60

Gambar I.24 Screenshot Hasil Pengujian Melihat Game yang Dimiliki Apabila Tidak Memiliki Game.....	60
Gambar I.25 Screenshot Hasil Pengujian Melihat Game yang Dimiliki Apabila Tidak Ada Parameter yang di Input.....	60
Gambar I.26 Screenshot Hasil Pengujian Mencari Game yang Dimiliki Apabila Salah Satu Parameter di Input.....	60
Gambar I.27 Screenshot Hasil Pengujian Mencari Game yang Dimiliki Apabila Tidak Ada yang Memenuhi Kriteria.....	60
Gambar I.28 Screenshot Hasil Pengujian Mencari Game di Toko Jika Hanya Salah Satu Parameter di Input.....	61
Gambar I.29 Screenshot Hasil Pengujian Mencari Game di Toko Jika Semua Parameter di Input.....	61
Gambar I.30 Screenshot Hasil Pengujian Mencari Game di Toko Jika Tidak Ada Parameter di Input.....	61
Gambar I.31 Screenshot Hasil Pengujian Mencari Game di Toko Jika Tidak Ada Game yang Memenuhi Kriteria.....	61
Gambar I.32 Screenshot Hasil Pengujian Top Up Saldo.....	62
Gambar I.33 Screenshot Hasil Pengujian Melihat Riwayat Pembelian Apabila Sudah Pernah Membeli Game.....	62
Gambar I.34 Screenshot Hasil Pengujian Melihat Riwayat Pembelian Apabila Tidak Memiliki Riwayat Pembelian Game.....	62
Gambar I.35 Screenshot Hasil Pengujian Help Untuk User.....	62
Gambar I.36 Screenshot Hasil Pengujian Help Untuk Admin.....	62
Gambar I.37 Screenshot Hasil Pengujian Help Apabila Login Tidak Dilakukan.....	62
Gambar I.38 Screenshot Hasil Pengujian Save.....	63
Gambar I.39 Screenshot Hasil Pengujian Exit Apabila Setuju Save.....	63
Gambar I.40 Screenshot Hasil Pengujian Exit Apabila Input di Luar Y/N.....	63

A. Deskripsi Persoalan

BNMO (dibaca: Binomo) adalah sebuah robot *game* milik Indra dan Doni yang membantu mereka melepas stress ketika mendapatkan tugas selama di Institut Teknologi Bandung. BNMO dulunya memiliki sistem inventarisasi & toko game yang baik. Indra dan Doni menjalani kuliah 2 semester di ITB dan merasa kesulitan dan stress. Doni menghabiskan waktu dengan BNMO untuk bermain *game*. Namun, Indra lebih suka bersenang-senang bermain *gacha*, akan tetapi ia rugi terus. Sehingga, pada suatu saat, Indra membanting BNMO sehingga BNMO pun rusak. Doni merasa depresi saat ia tahu BNMO rusak. Doni pun segera memperbaiki BNMO dan ia pun meminta bantuan kalian untuk memperbaiki BNMO karena Doni tidak cukup ahli dalam ngoding.

Dalam tugas besar ini, persoalan yang ingin diselesaikan adalah proses untuk memperbaiki BNMO dengan melakukan pembuatan program untuk menjalankan sistem pengelolaan dari sebuah aplikasi platform penjualan game *online*. Untuk membentuk program, diperlukan berbagai fungsi dengan manfaat yang bermacam-macam agar dapat menjadi sebuah kesatuan aplikasi program. Dalam program yang dijalankan ini, terdapat dua jenis pengguna yang berperan dalam aplikasi yaitu pengguna sebagai admin dan sebagai user. Antara admin dan user ini sendiri terdapat perbedaan dalam berbagai role fungsi yang dapat dijalankannya. Admin disini berperan sebagai pengelola atau penjual dalam program dan dapat menjalankan fungsi-fungsi seperti untuk mendaftarkan pengguna dengan role User yang baru ke dalam aplikasi, mengubah spesifikasi dan jumlah stok game, melakukan pencarian pada game berdasarkan parameter tertentu dan juga untuk menambah saldo kepada role User. Sedangkan, User sendiri dalam program ini berperan sebagai pembeli game yang dapat menjalankan fungsi-fungsi seperti untuk melakukan pembelian game, melihat daftar game yang pernah dibeli sebelumnya dan riwayat pembeliannya, serta dapat melakukan pencarian pada game berdasarkan parameter tertentu (untuk fungsi pencarian dapat dilakukan baik untuk role admin maupun user). Untuk menjalankan aplikasi ini, terdapat fungsi untuk menjalankan loading data. Program ini juga memiliki fungsi untuk melakukan penyimpanan data apabila penggunanya ingin berhenti untuk menjalankan aplikasi ini.

B. Daftar Pembagian Kerja Anggota Kelompok

Table B.1 Daftar Pembagian Tugas Anggota Kelompok pada Source Code

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F01 – Dekomposisi, Abstraksi, dan Generalisasi Pola dengan Modular Programming	-	16521499 16521505 16521511 16521541	16521499 16521505 16521511 16521541	16521499 16521505 16521511 16521541
F02 – Register	Procedure register	16521505	16521505	16521499 16521505 16521511 16521541
F03 – Login	Procedure login	16521505	16521505	16521499 16521505 16521511 16521541
F04 – Menambah Game ke Toko Game	Procedure tambah_game	16521505	16521505	16521499 16521505 16521511 16521541
F05 – Mengubah Game pada Toko Game	Procedure ubah_game	16521505	16521505	16521499 16521505 16521511 16521541
F06 – Mengubah Stok Game di Toko	Procedure ubah_stok	16521541	16521541	16521499 16521505 16521511 16521541

F07 – Listing Game di Toko Berdasarkan ID, Tahun Rilis, dan Harga	Procedure list_game_toko	16521541	16521541	16521499 16521505 16521511 16521541
F08 – Membeli Game	Procedure buy_game	16521541	16521541	16521499 16521505 16521511 16521541
F09 – Melihat Game yang Dimiliki	Procedure list_game	16521541	16521541	16521499 16521505 16521511 16521541
F10 – Mencari Game yang Dimiliki dari ID dan Tahun Rilis	Procedure search_my_game	16521499	16521499	16521499 16521505 16521511 16521541
F11 – Mencari Game di Toko dari ID, Nama Game, Harga, Kategori, dan Tahun Rilis	Procedure search_game_at_store	16521499	16521499	16521499 16521505 16521511 16521541
F12 – Top Up Saldo	Procedure topup	16521499	16521499	16521499 16521505 16521511 16521541
F13 – Melihat Riwayat Pembelian	Procedure riwayat	16521499	16521499	16521499 16521505 16521511 16521541

F14 – Help	Procedure help	16521511	16521511	16521499 16521505 16521511 16521541
F15 – Load	Function load	16521511	16521511	16521499 16521505 16521511 16521541
F16 – Save	Procedure save	16521511	16521511	16521499 16521505 16521511 16521541
F17 – Exit	Procedure exit	16521511	16521511	16521499 16521505 16521511 16521541

Tabel B.2 Daftar Pembagian Tugas Anggota Kelompok pada Laporan

No	Bagian Laporan	NIM
1	Halaman Cover	16521511
2	Daftar Isi	16521511
3	Daftar Tabel	16521499 16521505 16521511 16521541
4	Daftar Gambar	16521499 16521505

		16521511 16521541
5	Deskripsi Persoalan	16521541
6	Daftar Pembagian Kerja Anggota Kelompok	16521499 16521505 16521511 16521541
7	Checklist Hasil Rancangan, Implementasi, dan Testing Setiap Primitif	16521511
8	Desain Command untuk Setiap Primitif	16521541
9	Desain Kamus Data	16521505
10	Desain Dekomposisi Algoritmik dan Fungsional Program	16521499 16521505 16521511 16521541
11	Spesifikasi Untuk Tiap Modul/Fungsi/Prosedur yang Dibuat	16521499 16521505 16521511 16521541
12	Screenshot Hasil Pengujian Program Berdasarkan Fitur-Fitur Pada Spesifikasi	16521511 16521541
13	Lampiran	16521499 16521505 16521511 16521541

C. Checklist Hasil Rancangan, Implementasi, dan Testing Setiap Primitif

Tabel C.1 Checklist Hasil Rancangan, Implementasi, dan Testing Setiap Primitif

Fitur	Desain	Implementasi	Testing
F01 – Dekomposisi, Abstraksi, dan Generalisasi Pola dengan Modular Programming	✓	✓	✓
F02 – Register	✓	✓	✓
F03 – Login	✓	✓	✓
F04 – Menambah Game ke Toko Game	✓	✓	✓
F05 – Mengubah Game pada Toko Game	✓	✓	✓
F06 – Mengubah Stok Game di Toko	✓	✓	✓
F07 – Listing Game di Toko Berdasarkan ID, Tahun Rilis, dan Harga	✓	✓	✓
F08 – Membeli Game	✓	✓	✓
F09 – Melihat Game yang Dimiliki	✓	✓	✓
F10 – Mencari Game yang Dimiliki dari ID dan Tahun Rilis	✓	✓	✓
F11 – Mencari Game di Toko dari ID, Nama Game, Harga, Kategori, dan Tahun Rilis	✓	✓	✓
F12 – Top Up Saldo	✓	✓	✓
F13 – Melihat Riwayat Pembelian	✓	✓	✓
F14 – Help	✓	✓	✓
F15 – Load	✓	✓	✓

F16 – Save	✓	✓	✓
F17 – Exit	✓	✓	✓

D. Desain Command untuk Setiap Primitif

1. F02 – Register

Nama command: register

Masukan: nama, username, password

Keluaran: pesan “Username (username) telah berhasil register ke dalam “Binomo”.” atau pesan “Username (username) sudah terpakai, silakan menggunakan username lain.”

2. F03 - Login

Nama command: login

Masukan: username, password

Keluaran: pesan “Halo (nama)! Selamat datang di “Binomo”.” atau pesan “Password atau username salah atau tidak ditemukan.” atau pesan “Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain “login”.” atau pesan “Maaf, anda harus menjadi user untuk melakukan hal tersebut.” atau pesan “Maaf, anda tidak memiliki izin untuk menjalankan perintah berikut. Mintalah ke administrator untuk melakukan hal tersebut.”

3. F04 - Menambah Game ke Toko Game

Nama command: tambah_game

Masukan: nama game, kategori, tahun rilis, harga, stok awal

Keluaran: pesan “Selamat! Berhasil menambahkan game (nama game).”

4. F05 - Mengubah Game pada Toko Game

Nama command: ubah_game

Masukan: ID game, nama game, kategori, tahun rilis, harga

Keluaran: -

5. F06 - Mengubah Stok Game di Toko

Nama command: ubah_stok

Masukan: ID game, jumlah

Keluaran: pesan “Stok game (nama game) berhasil ditambahkan. Stok sekarang: (jumlah_baru)” atau pesan “Stok game (nama game) berhasil dikurangi. Stok sekarang: (jumlah_baru)” atau pesan “Stok game (nama game) gagal dikurangi karena stok kurang. Stok sekarang: (jumlah_lama) (<(abs(jumlah))” atau pesan “Tidak ada game dengan ID tersebut!”

6. F07 - Listing Game di Toko Berdasarkan ID, Tahun Rilis dan Harga

Nama command: list_game_toko

Masukan: skema sorting (hanya salah satu antara tahun rilis dan harga dan sifatnya antara terurut naik atau turun)

Keluaran: ID game | nama | harga | kategori | tahun rilis | stok (kondisi berurut berdasarkan parameter dan tipe sorting pada masukan) atau pesan “Skema sorting tidak valid!”

7. F08 - Membeli Game

Nama command: buy_game

Masukan: ID game

Keluaran: pesan “Game “(nama game)” berhasil dibeli!” atau pesan “Anda sudah memiliki Game tersebut!” atau pesan “Saldo anda tidak cukup untuk membeli Game tersebut!” atau pesan “Stok Game tersebut sedang habis!”

8. F09 - Melihat Game yang dimiliki

Nama command: list_game

Masukan: -

Keluaran: ID game | nama game | kategori | tahun rilis | harga atau pesan “Maaf, kamu belum membeli game. Ketik perintah beli_game untuk beli.

9. F10 - Mencari Game yang dimiliki dari ID dan Tahun Rilis

Nama command: search_my_game

Masukan: ID game, tahun rilis (boleh hanya salah satu ataupun tidak ada masukan sama sekali)

Keluaran: ID game | nama game | harga | kategori | tahun rilis (berdasarkan kesesuaian dengan parameter masukan yang diberikan, jika tidak ada parameter maka semua game milik user akan ditampilkan) atau pesan “Tidak ada game pada inventory-mu yang memenuhi kriteria”

10. F11 - Mencari Game di Toko dari ID, Nama Game, Harga, Kategori dan Tahun Rilis

Nama command: search_game_at_store

Masukan: ID game, nama game, harga, kategori, tahun rilis (boleh hanya salah satu ataupun tidak ada masukan sama sekali)

Keluaran: ID game | nama game | harga | kategori | tahun rilis | stok (berdasarkan kesesuaian dengan parameter masukan yang diberikan, jika tidak ada parameter maka semua game yang ada di toko akan ditampilkan)

11. F12 - Top Up Saldo

Nama command: topup

Masukan: username, saldo

Keluaran: pesan “Top up berhasil. Saldo (nama) bertambah menjadi (saldo_baru).” atau pesan “Masukan tidak valid.” atau pesan “Username “(username)” tidak ditemukan.”

12. F13 - Melihat Riwayat Pembelian

Nama command: riwayat

Masukan: -

Keluaran: ID game | nama game | harga | tahun beli | (data ditampilkan secara terurut berdasarkan tahun terbaru pembelian game dari user) atau pesan “Maaf, kamu tidak ada riwayat pembelian game. Ketik perintah beli_game untuk membeli.”

13. F14 – Help

Nama command: help

Masukan: -

Keluaran: penjelasan dari berbagai macam fungsi yang ada dalam program ini. Keluaran akan dibedakan berdasarkan role yang dimiliki oleh pengguna aplikasi apakah sebagai admin atau user dengan menampilkan penjelasan fungsi yang hanya dapat dijalankan oleh rolenya masing-masing saja. Jika belum login, maka kondisi tampilan akan dianggap sama seperti tampilan pada role user.

14. F15 – Load

Nama command: load

Masukan: nama_folder

Keluaran: pesan “Selamat datang di antarmuka “Binomo”.” atau pesan “Tidak ada nama folder yang diberikan!” atau pesan “Folder “(nama_folder)” tidak ditemukan.”

15. F16 – Save

Nama command: save

Masukan: save_folder

Keluaran: pesan “Data telah disimpan pada folder (save_folder)!

16. F17 – Exit

Nama command: exit

Masukan: y, Y, n, atau N

Keluaran: -

E. Desain Kamus Data

```
type user: < id: string;           {user ID}
          username: string;        {username dari pengguna aplikasi yang unik}
          nama: string;      {nama dari pengguna}
          password: string;       {kata sandi untuk akun pengguna}
          role: string;        {peran pengguna apakah admin atau user}
          saldo: string       {saldo dari akun pengguna}

type game: < id: string; {game ID}
          nama: string;    {nama dari game}
          kategori: string; {jenis kategori dari game}
          tahun_rilis: string; {tahun game dikeluarkan}
          harga: string;    {harga dari game yang dijual}
          stok: string       {stok game yang tersisa di toko}

type riwayat: < game_id: string;     {ID dari game}
              nama: string;    {nama dari game}
              harga: string;   {harga dari game}
              user_id: string;  {user ID yang membeli game}
              tahun_beli: string {tahun saat pembelian game}

type kepemilikan: < game_id: string;     {ID dari game}
                  user_id: string  {user ID yang membeli game}
```

KAMUS GLOBAL

```
fileusers: array of SEQFILE {array file csv user}
filegame: array of SEQFILE {array file csv game}
fileriwayat: array of SEQFILE {array file csv riwayat}
filekepemilikan: array of SEQFILE {array file csv kepemilikan}
tuser: array of array of string {array dalam program untuk user}
tgame: array of array of string {array dalam program untuk game}
triwayat: array of array of string {array dalam program untuk riwayat}
```

tkepemilikan: array of array of string {array dalam program untuk kepemilikan}
tpengguna: array of string {array untuk menambah riwayat dan kepemilikan baru}
tinventory : array of array of string {array untuk game pada inventory}
turut : array of array of string {array untuk mengurutkan game}
Admin: boolean {menentukan role}
Loginn: boolean {menentukan kondisi sudah masuk dalam akun atau belum}

KAMUS LOKAL

{Prosedur Register} {F02}

nama: string
username: string
password: string

{Prosedur Login} {F03}

nama: string
username: string

{Prosedur tambah_game} {F04}

namagame: string
kategori: string
tahun: string
harga: string
stok: string
tbaru: array of array of string

{Prosedur ubah_game} {F05}

idgame: string
namagame: string
kategori: string

tahun: string

harga: string

{Prosedur ubah_stok} {F06}

idgame: string

jumlah: integer

{Prosedur list_game_toko} {F07}

skema: string

{Prosedur buy_game} {F08}

stok: integer

harga: integer

idgame: string

saldo: integer

punya: boolean

{Prosedur list_game} {F09}

cek: integer

a: integer

{Prosedur search_my_game} {F10}

idgame: string

tahun: string

cek: integer

{Prosedur search_game_at_store} {F11}

idgame: string

nama: string

harga: string

kategori: string

tahun: string

cek: integer

a: integer

{Prosedur topup} {F12}

username: string

saldo: integer

a: integer

total: integer

{Prosedur riwayat} {F13}f:

cek: integer

a: integer

{Prosedur help} {F14}

-

{Fungsi load} {F15}

f: SEQFILE (parser, args, cwd)

ada: boolean

{Prosedur save} {F16}

f: SEQFILE (parent_dir, path)

directory: string

{Prosedur exit} {F17}

keluar: string

F. Desain Dekomposisi Algoritmik dan Fungsional Program

a. Dekomposisi Algoritmik

{ Fungsi Load }

{ I.S. : berbagai data untuk kebutuhan program masih berbentuk csv }

Membaca file user.csv dan masukkan ke dalam array user

{ F.S./I.S. : array user terisi dengan data file user.csv }

Membaca file game.csv dan masukkan ke dalam array game

{ F.S./I.S. : array game terisi dengan data file game.csv }

Membaca file riwayat.csv dan masukkan ke dalam array riwayat

{ F.S./I.S. : array riwayat terisi dengan data file riwayat.csv }

Membaca file kepemilikan.csv dan masukkan ke dalam array kepemilikan

{ F.S./I.S. : array kepemilikan terisi dengan data file kepemilikan.csv }

{ Prosedur Login }

{ I.S. : admin atau user belum login dan data sudah dalam bentuk array }

Menerima input berupa username dan password

{ F.S./I.S. : variabel username dan password sudah terisi dan siap untuk diproses }

Melakukan pengecekan username dan password dengan array user

{ F.S. : Jika username dan password sudah benar, program akan berlanjut dengan pengguna sudah berstatus login. Jika tidak, program akan menampilkan pesan kesalahan dan berhenti }

{ Prosedur Register }

{ I.S. : array user terisi dengan data dari file user.csv }

Mengecek role pengguna dengan mengecek array user pada bagian role

{ F.S./I.S. : Jika role pengguna adalah admin, program akan berlanjut. Jika tidak, prosedur register akan berhenti }

Menerima input berupa nama, username, dan password

{ F.S./I.S. : variabel nama, username, dan password sudah terisi dan siap untuk diproses }

Melakukan pengecekan username dengan array user

{ F.S/I.S : Jika username sudah ada pada array user, program akan menuliskan pesan kesalahan dan berhenti. Jika sudah unik, program baru akan berlanjut }

Memasukkan nama, username, password dan role sebagai user ke dalam array user

{ F.S : array user siap diproses dan menampilkan pesan bahwa username telah berhasil didaftarkan}

{ Prosedur tambah_game }

{ I.S. : array user terisi dengan data dari file user.csv }

Mengecek role pengguna dengan mengecek array user pada bagian role

{ F.S./I.S. : Jika role pengguna adalah admin, program akan berlanjut. Jika tidak, prosedur tambah_game akan berhenti }

Menerima input berupa nama game, kategori, tahun rilis, harga dan stok

{ F.S./I.S. : Jika semua variabel input sudah dimasukkan, program akan berlanjut. Jika ada salah satu variabel yang tidak dimasukkan, program akan menuliskan pesan kesalahan dan meminta input ulang }

Memasukkan nama game, kategori, tahun rilis, harga dan stok ke dalam array game

{ F.S : menuliskan pesan bahwa game sudah berhasil ditambahkan, array game dengan tambahan game yang baru siap untuk diproses }

{ Prosedur ubah_game }

{ I.S. : array user terisi dengan data dari file user.csv }

Mengecek role pengguna dengan mengecek array user pada bagian role

{ F.S./I.S. : Jika role pengguna adalah admin, program akan berlanjut. Jika tidak, prosedur ubah_game akan berhenti }

Menerima input berupa ID game, nama game, kategori, tahun rilis, dan harga (input bersifat opsional, tidak harus semua variabel diinput)

{ F.S./I.S. : variabel yang dimasukkan siap untuk diproses }

Melakukan pengecekan ID game dengan array game

{ F.S./I.S. : Jika variabel ID game tidak ada dalam array atau tidak dimasukkan, program akan berhenti. Jika ada, pengecekan program akan berlanjut }

Melakukan pengecekan variabel selain ID game yang dimasukkan dengan array game sesuai dengan ID game yang telah dimasukkan sebelumnya

{ F.S. : melakukan perubahan informasi pada array game sesuai dengan variabel yang dimasukkan sebelumnya, }

{ Prosedur ubah_stok }

{ I.S. : array user terisi dengan data dari file user.csv }

Mengecek role pengguna dengan mengecek array user pada bagian role

{ F.S./I.S. : Jika role pengguna adalah admin, program akan berlanjut. Jika tidak, prosedur ubah_stok akan berhenti }

Menerima input berupa ID game dan jumlah

{ F.S./I.S. : variabel yang dimasukkan siap untuk diproses }

Melakukan pengecekan ID game dengan array game

{ F.S./I.S. : Jika variabel ID game tidak ada dalam array atau tidak dimasukkan, program akan menampilkan pesan kesalahan dan berhenti. Jika ada, pengecekan program akan berlanjut }

Melakukan pengecekan variabel jumlah dengan array game berdasarkan stok sesuai dengan ID game yang telah dimasukkan sebelumnya

{ F.S. : Jika variabel jumlah tidak dimasukkan, program akan berhenti. Jika jumlah pada variabel ditambah dengan jumlah yang ada pada array game bagian stok kurang dari 0, program akan menampilkan pesan kesalahan dan berhenti. Jika jumlah pada variabel ditambah dengan jumlah yang ada pada array game bagian stok lebih dari 0, program akan menuliskan pesan bahwa stok game sudah berhasil diubah dan nilai yang ada dalam array game bagian stok juga akan mengalami perubahan dari kondisi penambahan atau pengurangan yang ada }

{ Prosedur list_game_toko }

{ I.S. : array game terisi dengan data dari file game.csv }

Menerima input berupa skema sorting (input berupa kategori dan sistem pengurutan) (opsional)

{ F.S./I.S. : Jika skema sorting yang dimasukkan merupakan tahun dan harga beserta ada sistem pengurutannya antara + atau -, program akan berlanjut, variabel skema sorting siap untuk diproses. Jika skema sorting tidak dimasukkan, program akan berlanjut juga. Diluar itu, masukan dianggap tidak valid dan program akan menampilkan pesan kesalahan dan berhenti }

Melakukan pengecekan variabel yang diinput antara parameter harga atau tahun dan pengurutan secara naik atau turun dengan array game

{ F.S. : menampilkan urutan game yang ada di toko berdasarkan parameter harga atau tahun serta berdasarkan urutan naik atau turun tergantung variabel yang dimasukan. Jika tidak ada variabel yang dimasukkan, maka program akan terurut naik berdasarkan id gamenya }

{ Prosedur buy_game }

{ I.S. : array user terisi dengan data dari file user.csv }

Mengecek role pengguna dengan mengecek array user pada bagian role

{ F.S./I.S. : Jika role pengguna adalah user, program akan berlanjut. Jika tidak, prosedur buy_game akan berhenti }

Menerima input berupa ID game

{ F.S./I.S. : Jika variabel ID game tidak dimasukkan, program akan menampilkan pesan kesalahan dan berhenti. Jika sudah dimasukkan, program akan berlanjut dan variabel ID game siap untuk diproses }

Melakukan pengecekan variabel ID Game dengan array game

{ F.S./I.S. : Jika variabel ID Game tidak ditemukan dalam array game, program akan menampilkan pesan kesalahan dan berhenti. Jika ditemukan, maka pengecekan program akan berlanjut}

Melakukan pengecekan kepemilikan user pada array kepemilikan berdasarkan masukan variabel ID Game

{ F.S./I.S. : Jika variabel ID game sudah ada dalam array kepemilikan, program akan menampilkan pesan bahwa game sudah dimiliki dan program akan berhenti. Jika belum ada, program akan melanjutkan pengecekan }

Melakukan pengecekan saldo user pada array user berdasarkan masukan variabel ID Game

{ F.S./I.S. : Jika saldo user yang ingin membeli game pada array lebih kecil dibandingkan dengan parameter harga pada array game berdasarkan ID game yang dimasukkan, program akan menampilkan pesan bahwa saldo user tidak mencukupi untuk melakukan pembelian game dan program akan berhenti. Jika saldo user lebih besar dari parameter harga pada array game berdasarkan ID game yang dimasukkan, program akan melanjutkan pengecekan }

Melakukan pengecekan stok game pada array game berdasarkan masukan variabel ID Game

{ F.S./I.S. : Jika ID game yang dimasukkan memiliki nilai 0 untuk parameter stok dalam array game, program akan menampilkan pesan bahwa stok game sedang habis dan program akan berhenti. Jika bernilai lebih dari 0, program akan menampilkan pesan bahwa game dengan ID yang terdapat dalam array game berdasarkan variabel input berhasil dibeli, array user bagian saldo akan berkurang sejumlah harga game, array

game untuk game id yang dibeli bagian stok akan berkurang 1, dan array riwayat serta kepemilikan akan mengalami penambahan data pembelian }

{ Prosedur list_game }

{ I.S. : array user terisi dengan data dari file user.csv }

Mengecek role pengguna dengan mengecek array user pada bagian role

{ F.S./I.S. : Jika role pengguna adalah user, program akan berlanjut. Jika tidak, prosedur list_game akan berhenti }

Melakukan pengecekan user_id dari user pada array kepemilikan

{ F.S./I.S. : Jika user_id dari user tidak ada dalam array kepemilikan, program akan menampilkan pesan kesalahan dan berhenti. Jika user_id dari user ada dalam array kepemilikan, program akan berlanjut }

Melakukan pengecekan dari ID game yang ada pada array kepemilikan dengan array game

{ F.S : menampilkan daftar game yang telah dibeli oleh user berdasarkan kesamaan antara array kepemilikan dan array game }

{ Prosedur search_my_game }

{ I.S. : array user terisi dengan data dari file user.csv }

Mengecek role pengguna dengan mengecek array user pada bagian role

{ F.S./I.S. : Jika role pengguna adalah user, program akan berlanjut. Jika tidak, prosedur search_my_game akan berhenti }

Menerima input berupa ID game dan tahun (opsional)

{ F.S./I.S. : Jika semua parameter masukan dikosongkan, program akan menampilkan semua game yang dimiliki kemudian berhenti. Jika ada salah satu atau kedua parameter diisi, maka program akan berlanjut, variabel siap untuk diproses }

Melakukan pengecekan variabel ID game dengan array kepemilikan dan pengecekan tahun dengan array game berdasarkan masukan variabel ID game sebelumnya

{ F.S. : Jika variabel ID game atau tahun tidak ada dalam array kepemilikan (kondisi satu inputan saja) atau variabel ID game tidak sesuai dengan variabel tahun dalam array game (kondisi dua inputan), program akan menampilkan pesan bahwa tidak ada game yang memenuhi kriteria. Jika variabel ID game atau variable tahun ada dalam array kepemilikan (kondisi satu inputan saja) atau variabel ID game sesuai dengan variabel tahun dalam array game (kondisi dua inputan), program akan menampilkan program akan menampilkan informasi lengkap game sesuai dengan variabel yang diberikan dan juga sesuai dengan array kepemilikan }

{ Prosedur search_game_at_store }

{ I.S. : array game terisi dengan data dari file game.csv }

Menerima input berupa ID game, nama game, harga game, kategori, dan tahun rilis (opsional)

{ F.S./I.S. : Jika kelima parameter variabel tidak dimasukkan, program akan menampilkan semua game yang ada dalam aplikasi. Jika ada parameter variabel yang diinput, program akan berlanjut dan variabel siap untuk diproses }

Melakukan pengecekan variabel yang diinput dengan array game

{ F.S. : Jika parameter variabel yang dimasukkan tidak ada yang sama dengan array game, program akan menampilkan pesan bahwa tidak ada game yang memenuhi kriteria. Jika ada parameter variabel yang sesuai dengan array game, program akan menampilkan informasi lengkap game dengan variabel yang sesuai tersebut }

{ Prosedur topup }

{ I.S. : array user terisi dengan data dari file user.csv }

Mengecek role pengguna dengan mengecek array user pada bagian role

{ F.S./I.S. : Jika role pengguna adalah admin, program akan berlanjut. Jika tidak, prosedur topup akan berhenti }

Menerima input berupa username dan saldo

{ F.S./I.S. : Jika variabel username dan saldo tidak dimasukkan, program akan berhenti. Jika sudah dimasukkan, program akan berlanjut dan variabel username dan saldo siap untuk diproses }

Melakukan pengecekan variabel username dengan array user

{ F.S/I.S : Jika username tidak ada pada array user, program akan menuliskan pesan kesalahan dan berhenti. Jika username ada, program akan berlanjut }

Melakukan pengecekan variabel saldo dengan array user

{ F.S : Jika masukkan pada variabel saldo ditambah dengan nilai saldo yang ada pada array user kurang dari 0, program akan menuliskan pesan kesalahan dan berhenti. Jika masukkan pada variabel saldo ditambah dengan nilai saldo yang ada pada array user lebih dari 0, program akan menjumlahkan nilai saldo yang ada pada array user serta memberikan pesan bahwa proses topup saldo sudah berhasil }

{ Prosedur riwayat }

{ I.S. : array user terisi dengan data dari file user.csv }

Mengecek role pengguna dengan mengecek array user pada bagian role

{ F.S./I.S. : Jika role pengguna adalah user, program akan berlanjut. Jika tidak, prosedur riwayat akan berhenti }

Melakukan pengecekan user_id dari user pada array riwayat

{ F.S./I.S. : Jika user_id dari user tidak ada dalam array riwayat, program akan menampilkan pesan kesalahan dan berhenti. Jika user_id dari user ada dalam array riwayat, program akan berlanjut }

Melakukan pengecekan dari ID game yang ada pada array riwayat dengan array game

{ F.S : menampilkan daftar game yang telah dibeli oleh user berdasarkan kesamaan antara array riwayat dengan array game berurut berdasarkan tahun terbaru pembelian }

{ Prosedur help }

{ I.S. : masukan ‘help’ dari pengguna aplikasi pada pemilihan prosedur }

Menampilkan list program beserta dengan penjelasannya

{ F.S. : List program dengan penjelasannya ditampilkan pada layar }

{ Prosedur save }

{ I.S. : array-array dengan data yang sudah dimodifikasi dari berbagai proses pada prosedur-prosedur sebelumnya }

Memindahkan array user ke dalam file user.csv

{ F.S./I.S. : file user.csv terisi dengan data array user yang baru }

Memindahkan array game ke dalam file game.csv

{ F.S./I.S. : file game.csv terisi dengan data array game yang baru }

Memindahkan array riwayat ke dalam file riwayat.csv

{ F.S./I.S. : file riwayat.csv terisi dengan data array riwayat yang baru }

Memindahkan array kepemilikan ke dalam file kepemilikan.csv

{ F.S. : file kepemilikan.csv terisi dengan data array kepemilikan yang baru }

{ Prosedur exit }

{ I.S. : masukan ‘exit’ dari pengguna aplikasi pada pemilihan menu }

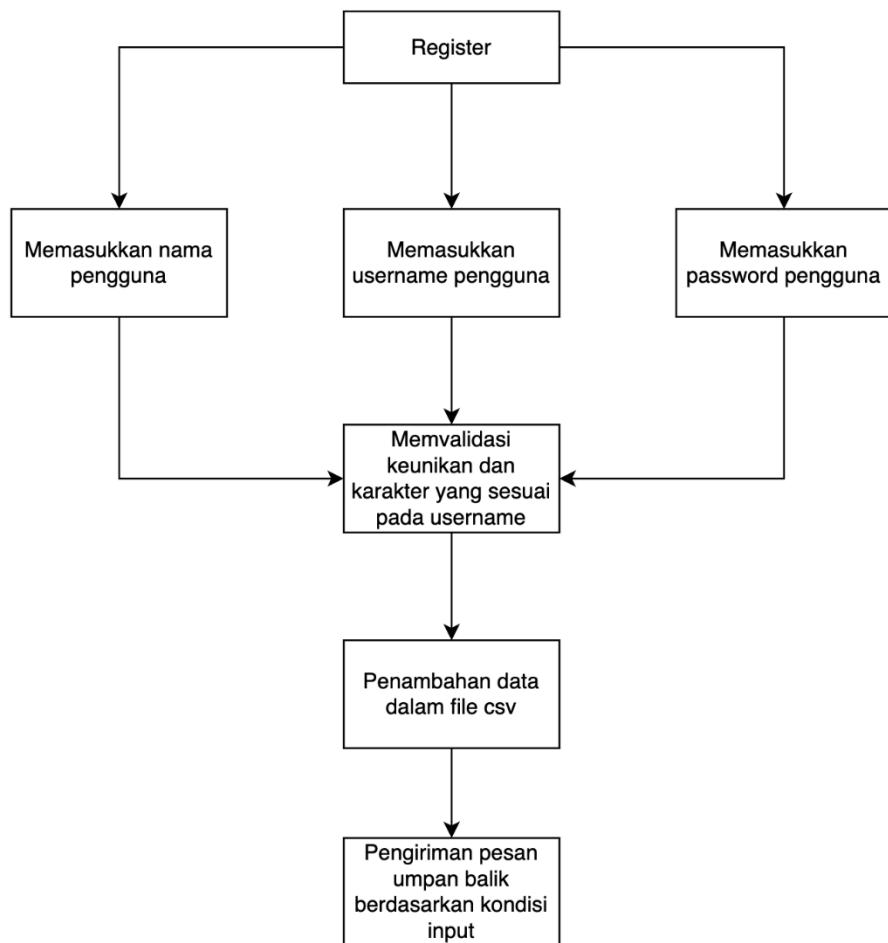
Menampilkan pertanyaan opsi kepada pengguna aplikasi apakah ingin melakukan penyimpanan pada data yang telah dimodifikasi

{ F.S./I.S. : Teks untuk konfirmasi ditampilkan pada layar }

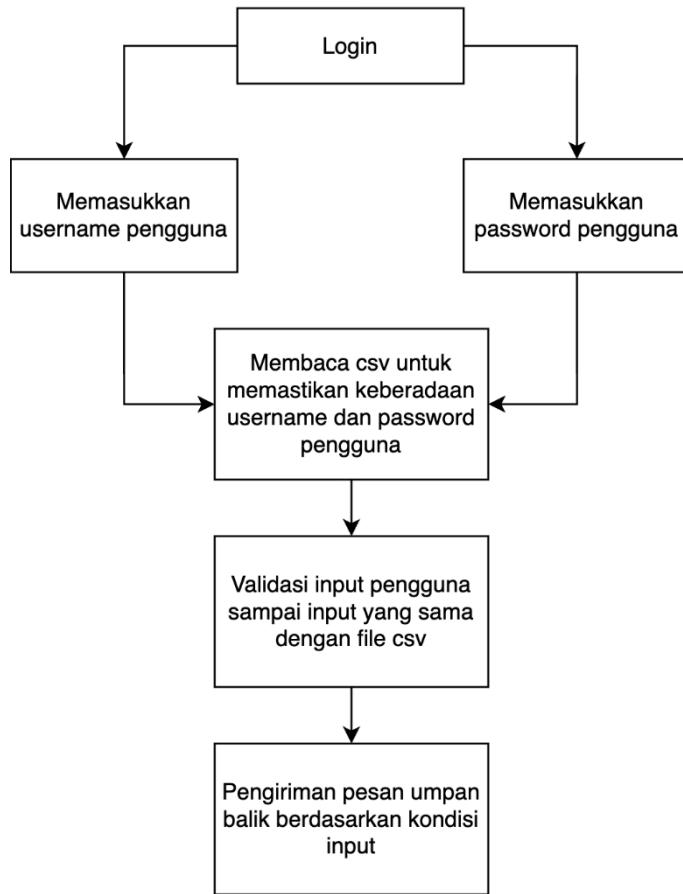
Melakukan pengecekan pada jawaban dari pengguna

{ F.S. : Jika jawaban pengguna adalah ‘Y’ atau ‘y’, program akan memanggil prosedur Save kemudian berhenti. Jika jawaban pengguna adalah ‘N’ atau ‘n’, program akan langsung berhenti tanpa melakukan prosedur Save. Jika bukan keduanya, jawaban menjadi tidak valid dan program akan meminta ulang masukkan pada pengguna hingga jawaban valid }

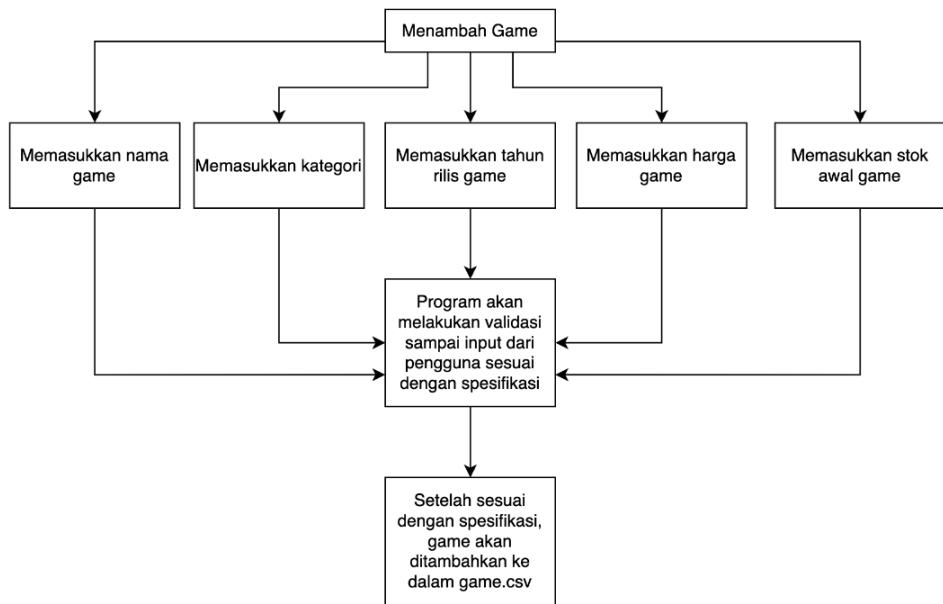
b. Dekomposisi Fungsional Program



Gambar F.1 Dekomposisi Register



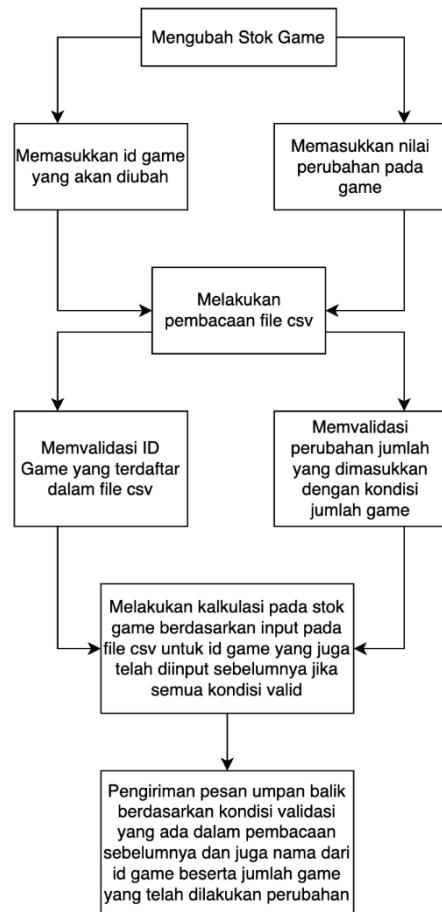
Gambar F.2 Dekomposisi Login



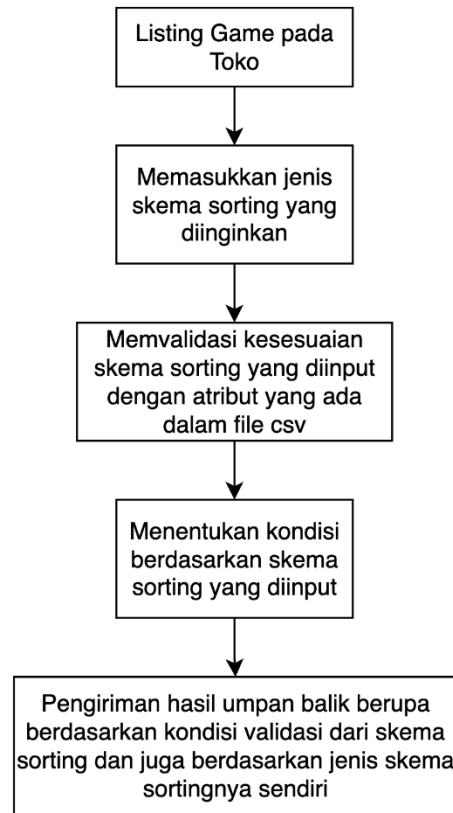
Gambar F.3 Dekomposisi Menambah Game



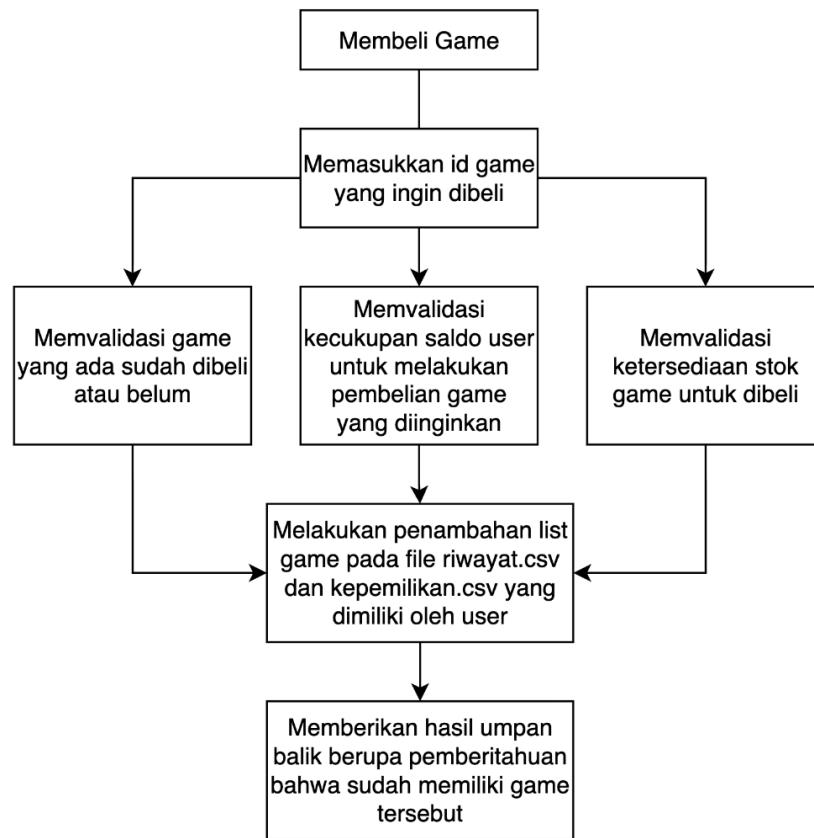
Gambar F.4 Dekomposisi Mengubah Game



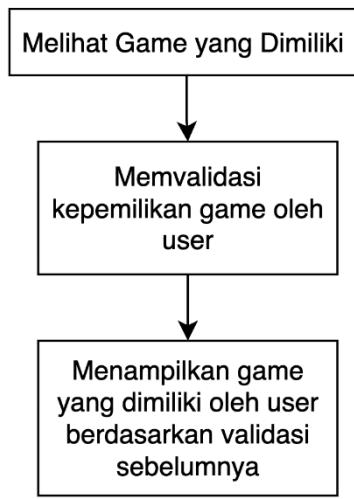
Gambar F.5 Dekomposisi Mengubah Stok Game



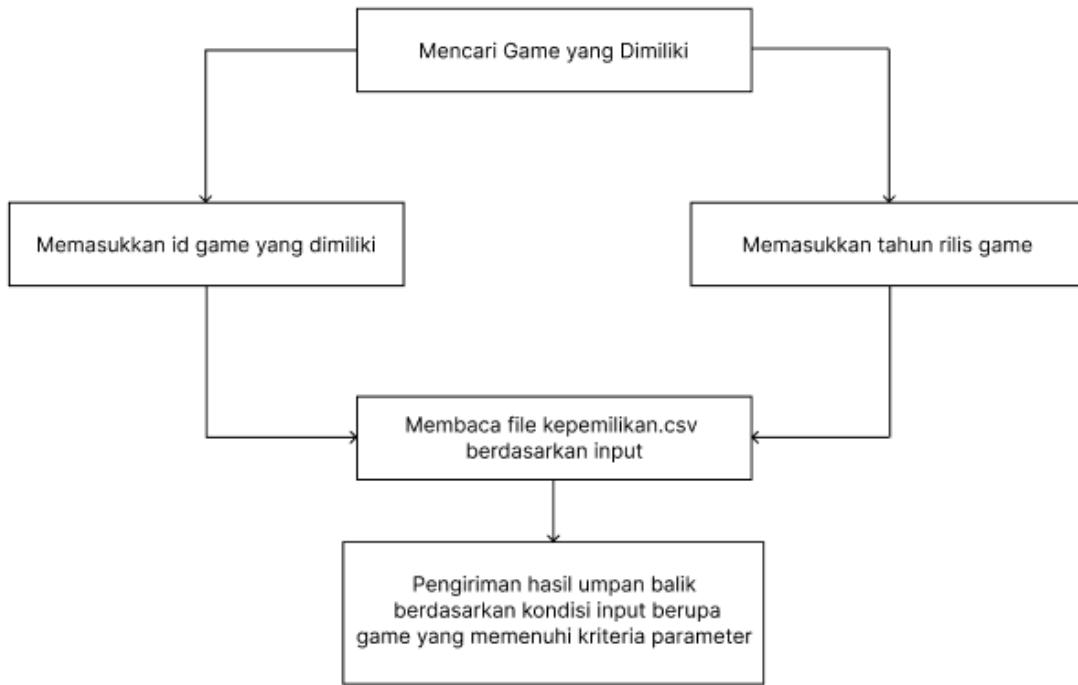
Gambar F.6 Dekomposisi Listing Game pada Toko



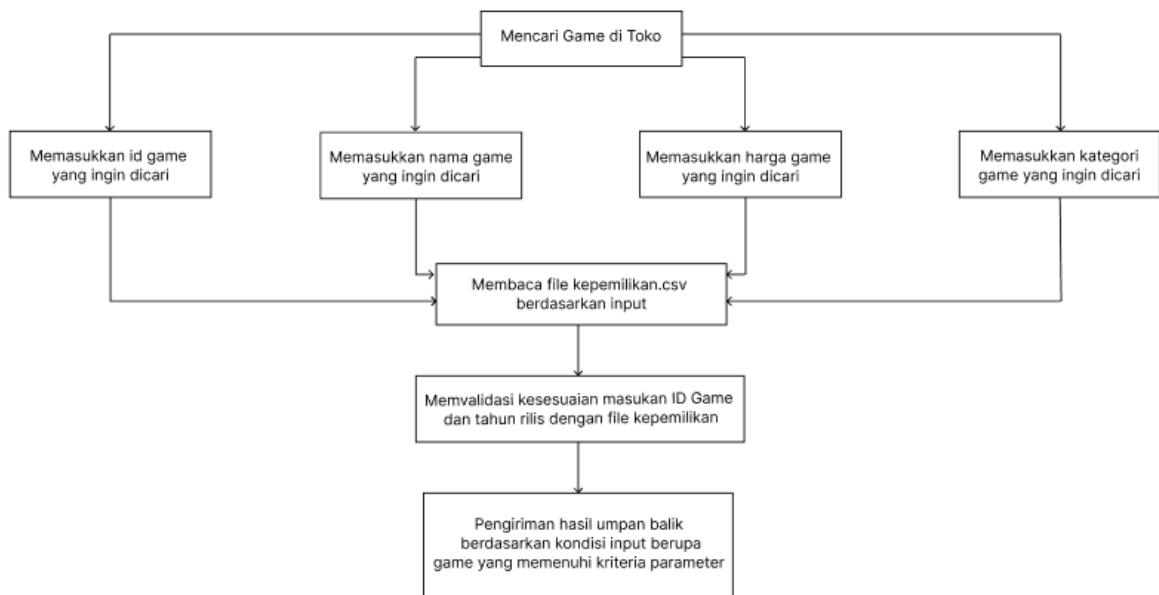
Gambar F.7 Dekomposisi Membeli Game



Gambar F.8 Dekomposisi Melihat Game yang dimiliki



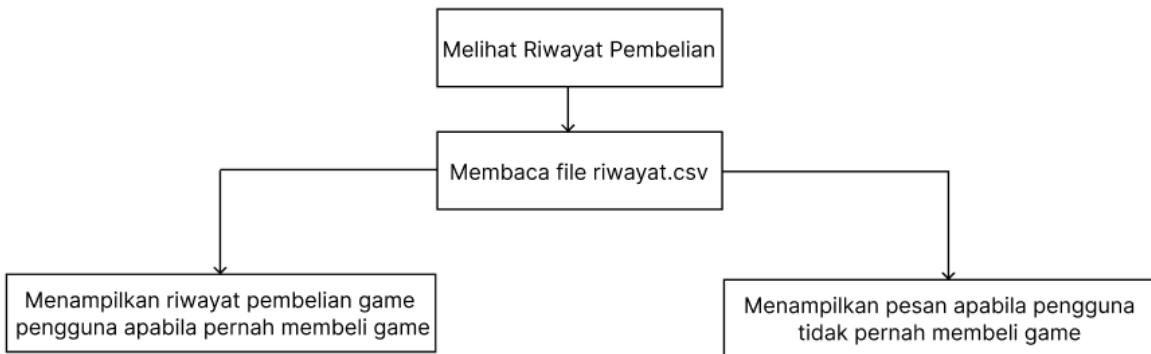
Gambar F.9 Dekomposisi Mencari Game yang dimiliki



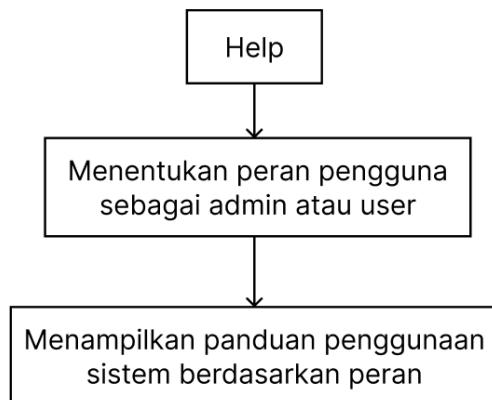
Gambar F.10 Dekomposisi Mencari Game di Toko



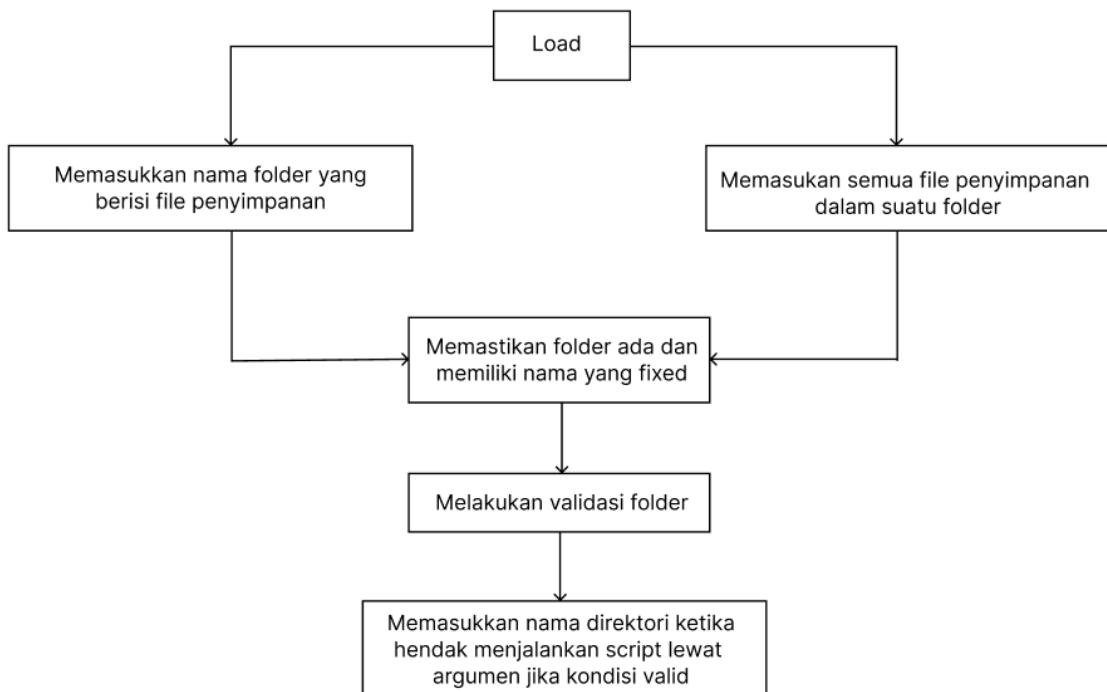
Gambar F.11 Dekomposisi Top-Up Saldo



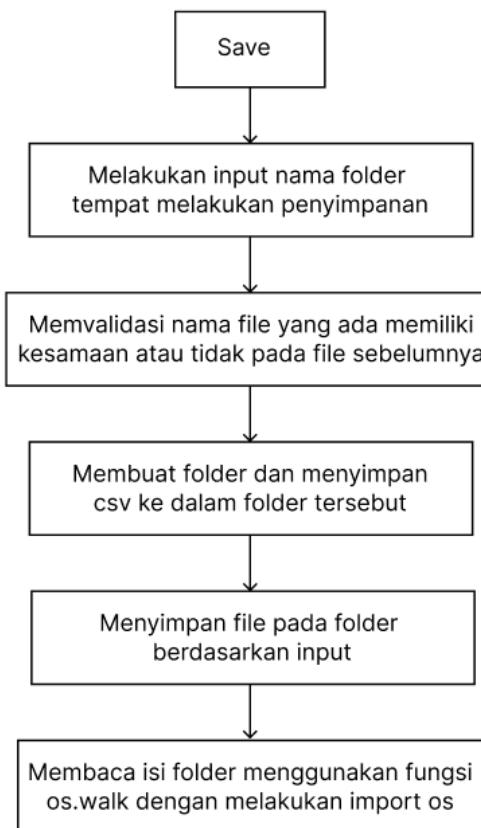
Gambar F.12 Dekomposisi Melihat Riwayat Pembelian



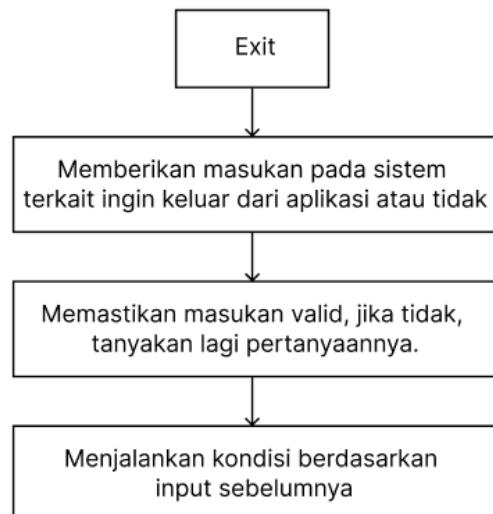
Gambar F.13 Dekomposisi Help



Gambar F.14 Dekomposisi Load



Gambar F.15 Dekomposisi Save



Gambar F.16 Dekomposisi Exit

G. Spesifikasi untuk Setiap Modul/Fungsi/Prosedur yang Dibuat

F02 – Register

Procedure Register (input nama : string, input username : string, input password : string, output case: array)

{ I.S. string nama, username, dan password yang akan dimasukkan ke dalam array user}

{ F.S. array user yang sudah ditambahkan informasi pengguna baru dengan role user }

F03 – Login

Procedure login (input username: string, input password : string, output case : boolean)

{ Prosedur yang memeriksa apakah password inputan sama dengan password di array }

{ I.S. string username yang valid dalam array dan string password yang digunakan untuk login}

{ F.S. Boolean case berdasarkan input password }

F04 – Menambah Game ke Toko Game

```
Procedure tambah_game (input nama_game: string, input kategori: string, input  
tahun_rilis: string, input harga: integer, input stok_awal: integer, output case: array)  
{ Prosedur yang digunakan oleh admin untuk menambahkan game dalam toko}  
{ I.S. string nama game, kategori, tahun serta integer harga dan stok yang akan  
dimasukkan dalam array game}  
{ F.S. array game yang sudah berhasil ditambahkan game yang baru}
```

F05 – Mengubah Game pada Toko Game

```
Procedure ubah_game (input ID_game: string, input nama_game: string, input kategori:  
string, input tahun_rilis: string, input harga: integer, output case: array)  
{ Prosedur yang digunakan oleh admin untuk mengubah informasi yang ada dalam suatu  
game}  
{ I.S. string ID game yang wajib dimasukkan untuk pencarian game, beserta string nama  
game, kategori, tahun, dan integer harga yang bersifat opsional yang akan digunakan  
untuk mengubah informasi dalam array game}  
{ F.S. array game yang sudah berhasil dilakukan perubahan informasi}
```

F06 – Mengubah Stok Game di Toko

```
Procedure ubah_stok (input ID_game : string, input jumlah : integer, output case : string,  
output case: array)  
{ prosedur yang digunakan oleh admin untuk mengubah jumlah stok game }  
{ I.S. string ID game untuk menentukan nama game dan integer jumlah untuk merubah  
stok dalam array game}  
{ F.S. array game bagian stok yang telah diubah berdasarkan input jumlah}
```

F07 – Listing Game di Toko Berdasarkan ID, Tahun Rilis, dan Harga

```
Procedure list_game_toko (input sorting : integer, output case: array)

{ prosedur untuk mengurutkan game pada toko berdasarkan parameter dan jenis pengurutan}

{ I.S. string skema sorting untuk menentukan pengurutan game pada toko}

{ F.S. game yang ada dalam toko akan ditampilkan berdasarkan urutan input parameter dan jenisnya}
```

F08 – Membeli Game

```
Procedure buy_game (input ID_game : string, output case: array)

{ prosedur yang digunakan oleh user untuk melakukan pembelian game yang ada di toko}

{ I.S. string ID game untuk menentukan game mana yang akan dibeli}

{ F.S. Penambahan data dalam array riwayat dan kepemilikan sesuai game yang telah dibeli}
```

F09 – Melihat Game yang Dimiliki

```
Procedure list_game

{ Prosedur yang menampilkan daftar game yang dimiliki oleh user berdasarkan array kepemilikan }
```

F10 – Mencari Game yang Dimiliki dari ID dan Tahun Rilis

```
Procedure search_my_game (input ID_game: string, input tahun_rilis: string, output case: array)

{ Prosedur yang digunakan oleh user untuk menampilkan list game yang dimiliki berdasarkan input parameter}

{ I.S. string ID game dan tahun sebagai parameter untuk menentukan game dalam inventory apa saja yang akan ditampilkan}
```

{ F.S. menampilkan daftar game yang dimiliki oleh user dengan parameter input yang sesuai }

F11 – Mencari Game di Toko dari ID, Nama Game, Harga, Kategori, dan Tahun Rilis

Prosedur search_game_at_store (input ID_game: string, input nama_game: string, input harga: integer, input kategori: string, input tahun_rilis: string, output case: array)

{ Prosedur untuk melakukan pencarian game yang ada dalam toko }

{ I.S. string ID game, nama game, kategori, tahun, dan integer harga sebagai parameter untuk menentukan game yang ada di toko apa saja yang akan ditampilkan }

{ F.S. menampilkan daftar game yang ada di toko dengan parameter input yang sesuai }

F12 – Top Up Saldo

Prosedur topup (input username: string, input saldo: integer, output case: array)

{ Prosedur yang digunakan oleh admin untuk mengubah jumlah saldo pada user }

{ I.S. string username untuk menentukan user dan integer saldo sebagai nilai saldo yang akan dirubah }

{ F.S. array pada user bagian saldo akan mengalami perubahan sesuai jumlah saldo yang diinput }

F13 – Melihat Riwayat Pembelian

Procedure riwayat

{ Prosedur yang digunakan oleh user untuk menampilkan riwayat pembelian game secara terurut dari parameter tahun pembelian berdasarkan array riwayat }

F14 – Help

Procedure Help

{ prosedur yang memberikan panduan penggunaan aplikasi }

F15 – Load

function load (input file_user: string, input file_game: string, input file_riwayat: string, input file_kepemilikan: string, output user : array)

{Fungsi kemudian membaca dan memasukkan data ke dalam array}

{I.S. string file_user, file_game, file_riwayat, file_kepemilikan}

{F.S. array dari berbagai file yang berisi data – data yang telah dibaca sebelumnya}

F16 – Save

Procedure Save

{ prosedur yang melakukan penyimpanan data array-array dari sistem ke file eksternal }

F17 – Exit

Procedure Exit

{ prosedur yang mengakhiri jalannya program dengan dua pilihan yaitu menyimpan atau tidak menyimpan perubahan yang telah dilakukan }

H. Struktur Data File Eksternal

Procedure register {F02}

```
global tuser
global admin
global loginn
if loginn then
    if admin then
        input(nama)
        input(username)
        input(password)
        i transversal [0..len(user)]
        if useri.username = username then
            output (“Username, useri.username, sudah terpakai, silahkan menggunakan
username yang lain”)
        else
            read nambahuser(tuser,nama,username,password)
            output (“Username, useri.username, telah berhasil register ke dalam Binomo!”)
        else
            output (“Bukan admin”)
    else
        output (“Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain login”)
```

Procedure login {F03}

```
global admin
global tpengguna
global tinventory
global loginn
global tuser
input(username)
```

```

input(password)
i transversal[0..len(user)]

if useri.username = username and useri.password = password then
    output ("Hallo, useri.username, selamat datang di binomo!")
    if useri.role = "admin" then
        admin <- True
    else
        admin <- False
    read(Inventory())
    loginn <- True
    return True
else:
    output ("Username atau password tidak ditemukan atau salah")

```

Procedure tambah_game {F04}

```

global tgame
global admin
global loginn
if loginn then
    if admin then
        input (namagame)
        input (kategori)
        input (tahun)
        input (harga)
        input (stok)
    while namagame="" or kategori="" or tahun="" or harga="" or stok="" do
        output ("Mohon masukkan semua informasi mengenai game agar disimpan dalam
        BNMO")
        input (namagame)
        input (kategori)

```

```

input (tahun)
input (harga)
input (stok)      Tbaru<- [[“”]]
j transversal(0..len(game)+1)
i transversal(0...6)
if j=len(game) then
    Tbaru[j][1] <- namagame
    Tbaru[j][2] <- kategori
    Tbaru[j][3] <- tahun
    Tbaru[j][4] <- harga
    Tbaru[j][5] <- stok+”\n”
else
    Tbaru[j][i]<-tgame[j][i]
gamei<-tbaru
output (“Selamat berhasil menambahkan game (tbaru[j][i])”)
else
output (“Bukan admin”)
else
output (“Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain login”)

```

Procedure ubah_game {F05}

```

global tgame
global admin
global loginn
if loginn then
    if admin then
        input (idgame)
        input (namagame)
        input (kategori)
        input (tahun)

```

input (harga)

```
j transversal [0..(len(game))]
if gamei.id=idgame then
    if namagame="" then
        namagame <- gamei.nama
    if kategori="" then
        kategori <- gamei.kategori
    if tahun="" then
        tahun <- gamei.tahun_rilis
    if harga="" then
        harga <- gamei.harga

    gamei.nama <- namagame
    gamei.kategori <- kategori
    gamei.tahun_rilis <- tahun
    gamei.harga <- harga
    return True
else
    output ("Game ID tidak ditemukan!")
else
    output ("Bukan admin")
else
    output ("Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain login")
```

Procedure ubah_stok {F06}

```
global admin
global tgame
global login
if loginn then
```

```

if admin then
    input (idgame)
    input int(jumlah)
        i transversal [0..(len(game))]
            if gamei.id=idgame then
                total <- int(jumlah)+ int(gamei.stok)

depend on (total, jumlah):
    total>0 and jumlah<0 then
        gamei.stok <- str(total)
        output ("Stok game",gamei.nama, "berhasil dikurangi. Stok sekarang",gamei.stok)
    total>0 and jumlah>0 then
        gamei.stok <- str(total)
        output ("Stok game",gamei.nama, "berhasil ditambah. Stok sekarang",gamei.stok)
    total=0 then
        output ("Stok game",gamei.nama, "gagal dikurangi karena stok kurang. Stok
        sekarang", gamei.stok)
    return True
else
    output ("Tidak ada game dengan ID tersebut!")
else
    output ("Bukan admin")
else
    output ("Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain login")

```

Procedure list_game_toko {F07}

```

global loginn
if loginn then
    input (skema)
    read urutan(skema)
    if skema = "harga+" or "harga-" then

```

```

if skema=="harga-" then
    a <- 1
    i transversal [1..(len(game))]
        output ("{a}.", "|", turut[i][0], "|", turut[i][1], "|", turut[i][4], "|", turut[i][2], "|"
        ,turut[i][3], "|", turut[i][5])
    a <- a+1

elif skema=="harga+" then
    a <- 1
    i transversal [(len(game))..1]
        output ("{a}.", "|", turut[i][0], "|", turut[i][1], "|", turut[i][4], "|", turut[i][2], "|"
        ,turut[i][3], "|", turut[i][5])
    a <- a+1

elif skema=="tahun-" then
    a <- 1
    i transversal [1..(len(game))]
        output ("{a}.", "|", turut[i][0], "|", turut[i][1], "|", turut[i][4], "|", turut[i][2], "|"
        ,turut[i][3], "|", turut[i][5])
    a <- a+1

elif skema=="tahun+":
    a <- 1
    i transversal [(len(game))..1]
        output ("{a}.", "|", turut[i][0], "|", turut[i][1], "|", turut[i][4], "|", turut[i][2], "|"
        ,turut[i][3], "|", turut[i][5])
    a <- a+1

elif skema=="":
    a <- 1
    i transversal [1..(len(game))]
        output ("{a}.", "|", gamei.id, "|", gamei.nama, "|", gamei.harga, "|",
        gamei.kategori, "|", gamei.tahun_rilis, "|", gamei.stok)
    a <- a+1

```

```

else:
    output ("Skema sorting tidak valid!")

else:
    output ("Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain login")

```

Procedure buy_game {F08}

```

global tuser
global tgame
global tkepemilikan
global tpengguna
global admin
global loginn
if loginn then
    if admin=False then
        Stok <- 0
        Harga <-0
        input (idgame)
        saldo <- int(tpengguna[5])
        punya <- False
        i <- 0

        i transversal [0..(len(kepemilikan))]
        if kepemilikani.idgame = idgame then
            if int(kepemilikani.user_id) = int(tpengguna[0]) then
                punya <- True
                break
        else
            punya <- False

```

```

i transversal [0..(len(game))]
if gamei.id=idgame:
    stok <- gamei.stok
    harga<- gamei.harga
    i <- i
    break
else
    output ("Game ID tidak ditemukan!")
    return False

```

depend on (saldo,harga,stok,punya):

saldo>harga and stok>0 and punya=False:

```

output ("Game",gamei.nama," berhasil dibeli!")
read nambahkepemilikan(tkepemilikan,idgame)
tpengguna[5] <- str(saldo – harga)
gamei.stok <- str(stok – 1)
read nambahriwayat(triwayat,idgame)
i transversal [0..(len(user))]
if useri.id=tpengguna[0] then
    useri.saldo=tpengguna[5]

```

punya=True:

output ("Anda sudah memiliki game tersebut!")

saldo<harga:

output ("Saldo anda tidak cukup untuk membeli game tersebut")

stok=0:

output ("Stok tersebut sedang habis!")

else

output ("Hanya user")

else

output ("Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain login")

Procedur list_game {F09}

```
global tgame
global tkepemilikan
global tpengguna
global admin
global loginn
if loginn then
    if admin=False then
        cek <- 0
        output ("Daftar game: ")
        a <- 1
        i transversal [1..(len(kepemilikan))]
        if int(kepemilikan[i].user_id) = int(tpengguna[0]) then
            j transversal [0..(len(game))]
            if game[j].id = kepemilikan[i].game_id then
                output (game[j].id, "|", game[j].nama, "|", game[j].kategori, "|", game[j].tahun_rilis,
                "|", game[j].harga)
                a <- a+1
            cek <- cek+1
            break
        else:
            continue
    else:
        if cek=0 then
            output ("Maaf, kamu belum memiliki game.")
        else:
            output ("Hanya user")
    else:
        output ("Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain login")
```

Procedure search_my_game {F10}

```
global admin

global loginn

if loginn then

    if admin=False then

        input(idgame)

        input(tahun)

        output (“Daftar game pada inventory yang memenuhi kriteria:")

        if (id_game = “”) and (tahun_rilis = “”) then

            i transversal [0..(len(inventory))]

            if (tinventory[i][0] = id_game) and (tinventory[i][3] = tahun) then

                output(tinventory[i][0], “|” ,tinventory[i][1], “|” ,tinventory[i][4], “|”

                ,tinventory[i][2], “|” ,tinventory[i][3])

            else

                if cek=0 then

                    output (“Tidak ada game yang memenuhi kriteria!”)

            elif (id_game ≠ “”) and (tahun = “”) then

                i transversal [0..(len(inventory))]

                if (tinventory[i][0] = id_game) then

                    output(tinventory[i][0], “|” ,tinventory[i][1], “|” ,tinventory[i][4], “|”

                    ,tinventory[i][2], “|” ,tinventory[i][3])

                else

                    if cek=0 then
```

output (“Tidak ada game yang memenuhi kriteria!”)

elif (id_game = "") and (tahun ≠ "") then

i transversal [0..(len(inventory))]

if (tinventory[i][0] = id_game) then

```
output(tinventory[i][0], "|", tinventory[i][1], "|", tinventory[i][4], "|")
```

,tinventory[i][2], “|” ,tinventory[i][3]))

else

if cek=0 then

output (“Tidak ada game yang memenuhi kriteria!”)

elif (id_game = "") and (tahun = "") then

i transversal [0..(len(inventory))]

```
output(tinventory[i][0], "|", tinventory[i][1], "|", tinventory[i][4], "|")
```

,tinventory[i][2], “|” ,tinventory[i][3])

else

output ("Hanya user")

else

```
output ("Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain login")
```

Procedure search_game_at_store {F11}

global tgame

global loginn

if loginn then

input(idgame)

```
input(nama)
```

```

input(harga)
input(kategori)
cek <- 0
a <- 1
i transversal [1..(len(game))]
    if (gamei.id = id_game) or (id_game = "") then
        if (gamei.nama = nama) or (nama = "") then
            if (gamei.harga = harga) or (harga = "") then
                if (gamei.kategori = kategori) or (kategori = "") then
                    if (gamei.tahun_rilis = tahun) or (tahun = "") then
                        output (gamei.id, "|", gamei.nama, "|", gamei.harga, "|", gamei.kategori, "|"
                                , gamei.tahun_rilis, "|", gamei.stok)
                        cek <- cek+1
                    a <- a+1
                else
                    if cek=0 then
                        output ("Tidak ada game yang memenuhi kriteria")
                    else
                        output ("Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain login")

```

Procedure topup {F12}

```

global admin
global tuser
global loginn
if loginn then
    if admin then
        input(username)
        input(saldo)
        a <- 0
        total <- 0
        i transversal [0..(len(user))]

```

```

if (useri.username = username) then
    total <- int(useri.saldo) + saldo
    a<-i
    break
else
    output ("Username", username , "tidak ditemukan")
    return false

if total<0 then
    output ("Masukan tidak valid")
else
    output ("Top up berhasil. Saldo" username "menjadi" total")
    useri.saldo <- str(total)
else
    output ("Bukan admin")
else
    output ("Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain \"login\"")

```

Procedure riwayat {F13}

```

global tpengguna
global triwayat
global admin
global loginn
if loginn then
    if admin = False then
        cek <- 0
        a <- 1
        i transversal [1..(len(riwayat))]
        if riwayati.user_id = tpengguna[0] then
            if cek=0 then

```

```

output ("Daftar game :")

output ("{a}.", riwayati.game_id, "|", riwayati.nama, "|", riwayati.harga, "|"
riwayati.tahun_beli)

a <- a+1

cek <- cek+1

else

if cek=0 then

output ("Maaf kamu tidak punya riwayat pembelian game.")

else

output ("Hanya user")

else

output ("Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain login")

```

Procedure Help {F14}

```

login <- False

admin <- True

if login=False then

output ("===== HELP =====")

output ("1. login - Untuk melakukan login ke dalam sistem")

output ("2. list_game_toko - Untuk melihat list game yang dijual pada toko")

output ("3. buy_game - Untuk melakukan pembelian game")

output ("4. list_game - Untuk memberikan daftar game yang dimiliki pengguna")

output ("5. search_my_game - Untuk mendapatkan informasi game sesuai dengan query
yang diminta")

output ("6. search_game_at_store - Untuk melakukan pencarian game berdasarkan
parameter")

output ("7. riwayat - Untuk melihat riwayat pembelian game")

```

```

output ("8. save - melakukan penyimpanan data ke dalam file setelah dilakukan
perubahan")

else: {login=True}

if admin=True then

    output ("===== HELP =====")

    output ("1. register - Untuk melakukan registrasi user baru")

    output ("2. tambah_game - Untuk menambah game yang dijual pada toko")

    output ("3. ubah_game - Untuk mengubah informasi game yang dijual pada toko")

    output ("4. ubah_stok - Untuk mengubah stok game yang dijual pada toko")

    output ("5. list_game_toko - Untuk melihat list game yang dijual pada toko")

    output ("6. search_game_at_store - Untuk melakukan pencarian game berdasarkan
parameter")

    output ("7. topup - Untuk menambahkan saldo pada user")

    output ("8. save - melakukan penyimpanan data ke dalam file setelah dilakukan
perubahan")

else {admin=False}

    output ("1. list_game_toko - Untuk melihat list game yang dijual pada toko")

    output ("2. buy_game - Untuk melakukan pembelian game")

    output ("3. list_game - Untuk memberikan daftar game yang dimiliki pengguna")

    output ("4. search_my_game - Untuk mendapatkan informasi game sesuai dengan query
yang diminta")

    output ("5. search_game_at_store - Untuk melakukan pencarian game berdasarkan
parameter")

    output ("6. riwayat - Untuk melihat riwayat pembelian game")

    output ("7. save - melakukan penyimpanan data ke dalam file setelah dilakukan
perubahan")

```

Fungsi Load {F15}

Global filegame

Global filekepemilikan

Global fileriwayat

Global fileusers

Parser <-argparse.ArgumentParser

Parser.add_argument("folder_csv")

Args <- parser.parse_args()

Cwd <- os.getcwd()

Filecsv <- os.path.join(cwd,args.folder_csv)

Filegame <- filecsv+r"\game.csv"

Filekepemilikan <- filecsv+r"\kepemilikan.csv"

Fileriwayat <- filecsv+r"\riwayat.csv"

Fileusers <- filecsv+r"\users.csv"

Ada <- False

Roots,dirs,files transversal[os.walk(format(cwd))]

Dirr transversal [dirrs]

If args.folder_csv=dir then

ada <-True

break

If ada then

output ("Loading")

output ("Selamat datang di antarmuka Binomo")

else

```
if args.folder_csv="" then
    output ("Tidak ada nama folder yang diberikan")
else
    output ("Folder {args.folder_csv} tidak ditemukan")
    read (exit())
```

Procedure save {F16}

Global tuser

Global tgame

Global triwayat

Global tkepemilikan

input (directory)

Parent_dir <- os.getcwd()

Path <- os.path.join(parent_dir,directory)

Os.makedirs(path,exist_ok=True)

Pathusers <- path+"\users.csv"

Pathgame <- path+"\game.csv"

Pathriwayat <- path+"\riwayat.csv"

Pathkepemilikan <- path+"\kepemilikan.csv"

if os.path.exists(pathuser) then

Os.remove(pathuser)

if os.path.exists(pathgame) then

Os.remove(pathgame)

if os.path.exists(pathriwayat) then

```

Os.remove(pathriwayat)

if os.path.exists(pathkepemilikan) then

    Os.remove(pathkepemilikan)

    Fgame<-assign(pathuser,"w")

    j transversal [0..len(game)]

        i transversal [0...6]

            if I !=5 then

                write(fuser,str(tuser[j][i]))

                write(fuser,";")

            else

                write(fuser,str(tuser[j][i]))

        Fgame <- assign(pathgame,"w")

        j transversal[0..len(game)]

            i transversal[0...6]

                If i!=5 then

                    write(fgame,str(tgame[j][i]))

                    write(fgame,";")

                else

                    write(fgame,str(tgame[j][i]))

    Fkepemilikan <- assign(pathkepemilikan,"w")

    j transversal[0..len(kepemilikan)]

        i transversal[0...6]

            if i!=5 then

                write(fkepemilikan,str(tkepemilikan[j][i]))

```

```

write(fkepemilikan,";")

else

write(fkepemilikan,str(tkepemilikan[j][i])

Fgame <- assign(pathriwayat,"w")

j transversal[0..len(riwayat)]

i transversal[0...6]

If i!=5 then

    write(friwayat,str(triwayat[j][I])

    write(friwayat,";")

else

    write(friwayat,str(triwayat[j][I])

output ("Data telah disimpan pada folder", directory)

```

Procedure exit {F17}

```

input(keluar)

while keluar!="n" and keluar!="N" and keluar!="y" and keluar!="Y" do

    input(keluar)

    if keluar=="n" or keluar=="N":

        clear()

        exit()

    else:

        save()

        exit()

```

I. Screenshot Hasil Pengujian Program Berdasarkan Fitur – Fitur pada Spesifikasi

```
PS D:\tubesdaspro> python main.py folder_csv
Loading...
Selamat datang di antarmuka "Binomo"
```

Gambar I.1 Screenshot Hasil Pengujian Load

```
Masukkan nama : Jason Rivalino
Masukkan username: jasonrivalino
Masukkan password: tubes
Username jasonrivalino telah berhasil register ke dalam Binomo!
```

Gambar I.2 Screenshot Hasil Pengujian Register

```
Masukkan nama : Jason Rivalino
Masukkan username: jasonrivalino
Masukkan password: tubes
Username jasonrivalino sudah terpakai, silahkan menggunakan username yang lain
```

Gambar I.3 Screenshot Hasil Pengujian Register Apabila Username Sudah Terpakai

```
Masukkan username: fariz123
Masukkan password: farizz
Username atau password tidak ditemukan atau salah
```

Gambar I.4 Screenshot Hasil Pengujian Login Salah

```
Masukkan username: fariz123
Masukkan password: tubes
Halo fariz selamat datang di Binomo!
```

Gambar I.5 Screenshot Hasil Pengujian Login Benar

```
Masukkan Nama Game:
Masukkan Kategori: test
Masukkan Tahun Rilis: 2022
Masukkan Harga Game:
Masukkan Stok Awal:
Mohon masukkan semua informasi mengenai game agar disimpan di BNMO
Masukkan Nama Game:
```

Gambar I.6 Screenshot Hasil Pengujian Menambah Game ke Toko Game Apabila Data Tidak Lengkap (Gagal)

```
Masukkan Nama Game: haskell
Masukkan Kategori: adventure
Masukkan Tahun Rilis: 2010
Masukkan Harga Game: 2000
Masukkan Stok Awal: 100
Selamat berhasil menambahkan game haskell
[]
```

Gambar I.7 Screenshot Hasil Pengujian Menambah Game ke Toko Game Berhasil

```
Masukkan ID Game:
Masukkan Nama Game: python
Masukkan Kategori: game
Masukkan Tahun Rilis: 2022
Masukkan Harga: 5000
Game ID tidak ditemukan!
[]
```

Gambar I.8 Screenshot Hasil Pengujian Mengubah Game pada Toko Game saat ID Tidak Dimasukkan

```
Masukkan ID Game: GAME001
Masukkan Nama Game: hello world
Masukkan Kategori: simulation
Masukkan Tahun Rilis: 2009
Masukkan Harga: 10000
[]
```

Gambar I.9 Screenshot Hasil Pengujian Mengubah Game pada Toko Game Berhasil

```
Masukkan ID Game: GAME001
Masukkan Jumlah: -20
Stok game hello world gagal dikurangi karena stok kurang. Stok sekarang 9
[]
```

Gambar I.10 Screenshot Hasil Pengujian Mengubah Stok Game di Toko Apabila Stok Kurang

```
Masukkan ID Game: GAME010
Masukkan Jumlah: 10
Tidak ada game dengan ID tersebut!
[]
```

Gambar I.11 Screenshot Hasil Pengujian Mengubah Stok Game di Toko Apabila Tidak Ada Game dengan ID Tersebut

```
Masukkan ID Game: GAME002
Masukkan Jumlah: -3
Stok game python berhasil dikurangi. Stok sekarang 7
[]
```

Gambar I.12 Screenshot Hasil Pengujian Mengubah Stok Game di Toko Berhasil Dikurangi

```
Masukkan ID Game: GAME003
Masukkan Jumlah: 4
Stok game tubess berhasil ditambah. Stok sekarang 14
```

Gambar I.13 Screenshot Hasil Pengujian Mengubah Stok Game di Toko Berhasil Ditambah

```
Skema sorting: test+
Skema sorting tidak valid!
```

Gambar I.14 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Tidak Valid

```
Skema sorting:
1. | GAME001 | binomo | 1000 | action | 2022 | 9
2. | GAME002 | python | 199 | adventure | 2010 | 10
3. | GAME003 | tubess | 10000 | action | 2000 | 9
4. | GAME004 | line | 1000 | action | 2000 | 20
5. | GAME005 | asd | 23 | 123 | 123 | 1
6. | GAME006 | Ass | 2000 | action | 2012 | 100
7. | GAME007 | accc | 3000 | action | 2011 | 999
```

Gambar I.15 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Kosong

```
Skema sorting: harga+
1. | GAME005 | asd | 23 | 123 | 123 | 1
2. | GAME002 | python | 199 | adventure | 2010 | 7
3. | GAME004 | line | 1000 | action | 2000 | 20
4. | GAME006 | test | 2000 | edpenter | 2022 | 2
5. | GAME008 | catur | 10000 | simulation | 2015 | 15
6. | GAME001 | hello world | 10000 | simulation | 2009 | 9
7. | GAME003 | tubess | 10000 | action | 2000 | 14
8. | GAME007 | pesta | 20000 | party | 2019 | 23
```

Gambar I.16 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Harga dari Terkecil

Skema sorting: harga-						
1.	GAME007	pesta	20000	party	2019	23
2.	GAME003	tubess	10000	action	2000	14
3.	GAME001	hello world	10000	simulation	2009	9
4.	GAME008	catur	10000	simulation	2015	15
5.	GAME006	test	2000	edpenter	2022	2
6.	GAME004	line	1000	action	2000	20
7.	GAME002	python	199	adventure	2010	7
8.	GAME005	asd	23	123	123	1

Gambar I.17 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Harga dari Terbesar

Skema sorting: tahun+						
1.	GAME005	asd	23	123	123	1
2.	GAME003	tubess	10000	action	2000	14
3.	GAME004	line	1000	action	2000	20
4.	GAME001	hello world	10000	simulation	2009	9
5.	GAME002	python	199	adventure	2010	7
6.	GAME008	catur	10000	simulation	2015	15
7.	GAME007	pesta	20000	party	2019	23
8.	GAME006	test	2000	edpenter	2022	2

Gambar I.18 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Tahun dari Terkecil

Skema sorting: tahun-						
1.	GAME006	test	2000	edpenter	2022	2
2.	GAME007	pesta	20000	party	2019	23
3.	GAME008	catur	10000	simulation	2015	15
4.	GAME002	python	199	adventure	2010	7
5.	GAME001	hello world	10000	simulation	2009	9
6.	GAME004	line	1000	action	2000	20
7.	GAME003	tubess	10000	action	2000	14
8.	GAME005	asd	23	123	123	1

Gambar I.19 Screenshot Hasil Pengujian Listing Game di Toko Skema Sorting Tahun dari Terbesar

```
Masukkan ID Game: GAME001
Game "hello world" berhasil dibeli!
```

Gambar I.20 Screenshot Hasil Pengujian Membeli Game

```
Masukkan ID Game: GAME006
Anda sudah memiliki game tersebut!
```

Gambar I.21 Screenshot Hasil Pengujian Membeli Game Apabila Game Sudah Dimiliki

```
Masukkan ID Game: GAME008  
Saldo anda tidak cukup untuk membeli game tersebut
```

Gambar I.22 Screenshot Hasil Pengujian Membeli Game Apabila Saldo Tidak Cukup

```
Masukkan ID Game: GAME009  
Stok tersebut sedang habis!
```

Gambar I.23 Screenshot Hasil Pengujian Membeli Game Apabila Stok Game Habis

```
Daftar game:  
Maaf, kamu belum memiliki game.
```

Gambar I.24 Screenshot Hasil Pengujian Melihat Game yang Dimiliki Apabila Tidak Memiliki Game

```
Masukkan ID Game:  
Masukkan Tahun Rilis Game:  
Daftar game pada inventory yang memenuhi kriteria:  
GAME006 | test | 2000 | edpenter | 2022  
GAME002 | python | 199 | adventure | 2010  
GAME007 | pesta | 20000 | party | 2019  
GAME003 | tubess | 10000 | action | 2000
```

Gambar I.25 Screenshot Hasil Pengujian Melihat Game yang Dimiliki Apabila Tidak Ada Parameter yang di Input

```
Masukkan ID Game: GAME002  
Masukkan Tahun Rilis Game:  
Daftar game pada inventory yang memenuhi kriteria:  
GAME002 | python | 199 | adventure | 2010
```

Gambar I.26 Screenshot Hasil Pengujian Mencari Game yang Dimiliki Apabila Salah Satu Parameter di Input

```
Masukkan ID Game: GAME010  
Masukkan Tahun Rilis Game: 2100  
Daftar game pada inventory yang memenuhi kriteria:  
Tidak ada game yang memenuhi kriteria!
```

Gambar I.27 Screenshot Hasil Pengujian Mencari Game yang Dimiliki Apabila Tidak Ada yang Memenuhi Kriteria

```
Masukkan ID Game:  
Masukkan Nama Game:  
Masukkan Harga Game: 10000  
Masukkan Kategori Game:  
Masukkan Tahun Rilis Game:  
1. | GAME001 | hello world | 10000 | simulation | 2009 | 9  
2. | GAME003 | tubess | 10000 | action | 2000 | 13  
3. | GAME008 | catur | 10000 | simulation | 2015 | 15
```

Gambar I.28 Screenshot Hasil Pengujian Mencari Game di Toko Jika Hanya Salah Satu Parameter di Input

```
Masukkan ID Game: GAME002  
Masukkan Nama Game: python  
Masukkan Harga Game: 199  
Masukkan Kategori Game: adventure  
Masukkan Tahun Rilis Game: 2010  
1. | GAME002 | python | 199 | adventure | 2010 | 7
```

Gambar I.29 Screenshot Hasil Pengujian Mencari Game di Toko Jika Semua Parameter di Input

```
Masukkan ID Game:  
Masukkan Nama Game:  
Masukkan Harga Game:  
Masukkan Kategori Game:  
Masukkan Tahun Rilis Game:  
1. | GAME001 | hello world | 10000 | simulation | 2009 | 9  
2. | GAME002 | python | 199 | adventure | 2010 | 6  
3. | GAME003 | tubess | 10000 | action | 2000 | 13  
4. | GAME004 | line | 1000 | action | 2000 | 20  
5. | GAME005 | asd | 23 | 123 | 123 | 1  
6. | GAME006 | test | 2000 | edpenter | 2022 | 1  
7. | GAME007 | pesta | 20000 | party | 2019 | 22  
8. | GAME008 | catur | 10000 | simulation | 2015 | 15  
9. | GAME009 | komputer simulasi | 200000 | simulation | 2013 | 0
```

Gambar I.30 Screenshot Hasil Pengujian Mencari Game di Toko Jika Tidak Ada Parameter di Input

```
Masukkan ID Game: GAME011  
Masukkan Nama Game: yes  
Masukkan Harga Game: 1000000  
Masukkan Kategori Game: open world  
Masukkan Tahun Rilis Game: 2023  
Tidak ada game yang memenuhi kriteria
```

Gambar I.31 Screenshot Hasil Pengujian Mencari Game di Toko Jika Tidak Ada Game yang Memenuhi Kriteria

```
Masukkan username: farizuser  
Masukkan saldo: 1500000  
Top up berhasil. Saldo farizuser menjadi 11478000
```

Gambar I.32 Screenshot Hasil Pengujian Top Up Saldo

```
Daftar game :  
1. GAME006 | test | edpenter | 2022  
2. GAME002 | python | adventure | 2022  
3. GAME007 | pesta | party | 2022  
4. GAME003 | tubess | action | 2022
```

Gambar I.33 Screenshot Hasil Pengujian Melihat Riwayat Pembelian Apabila Sudah Pernah Membeli Game

```
Maaf kamu tidak punya riwayat pembelian game.
```

Gambar I.34 Screenshot Hasil Pengujian Melihat Riwayat Pembelian Apabila Tidak Memiliki Riwayat Pembelian Game

1. login - untuk melakukan login ke dalam sistem
2. list_game_toko - untuk melihat list game yang dijual pada toko
3. buy_game - untuk membeli game yang tersedia di toko
4. search_my_game - untuk mencari game yang dimiliki user
5. search_game_at_store - mencari game pada toko berdasarkan kriteria tertentu
6. riwayat - melihat riwayat tahun pembelian game user
7. help - memberi panduan terhadap penggunaan program

Gambar I.35 Screenshot Hasil Pengujian Help Untuk User

1. register - untuk melakukan registrasi user baru
2. login - untuk melakukan login ke dalam sistem
3. tambah_game - untuk menambah game yang dijual pada toko
4. list_game_toko - untuk melihat list game yang dijual pada toko
5. ubah_game - mengubah isi game
6. search_game_at_store - mencari game pada toko berdasarkan kriteria tertentu
7. topup - Menambahkan atau mengurangi saldo kepada user
8. help - memberi panduan terhadap penggunaan program

Gambar I.36 Screenshot Hasil Pengujian Help Untuk Admin

```
Maaf, anda harus login terlebih dahulu untuk mengirim perintah selain "login"
```

Gambar I.37 Screenshot Hasil Pengujian Help Apabila Login Tidak Dilakukan

```
Masukkan nama folder penyimpanan: folder_csv  
Data telah disimpan pada folder folder_csv
```

Gambar I.38 Screenshot Hasil Pengujian Save

```
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) y  
Masukkan nama folder penyimpanan: folder_csv  
Data telah disimpan pada folder folder_csv  
PS D:\tubesdaspro> █
```

Gambar I.39 Screenshot Hasil Pengujian Exit Apabila Setuju Save

```
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) save  
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) █
```

Gambar I.40 Screenshot Hasil Pengujian Exit Apabila Input di Luar Y/N

Lampiran

Form MoM Asistensi Tugas Besar
IF1210/Dasar Pemrograman
Sem. 2 2021/2022

Nomor Asistensi	:	1
No. Kelompok/Kelas	:	07/K11
Tanggal asistensi	:	12 April 2022

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
1	16521499 / M. Andiananto Bismaramadhan
2	16521505 / Fariz Putra Hanggara
3	16521511 / Nabilah Amanda Putri
4	16521541 / Jason Rivalino
5	
6	

Asisten pembimbing	NIM / Nama
	13519179 / Akifa Nabil

Catatan Asistensi:

Rangkuman Diskusi
<p>1. Bagaimana ide untuk sorting?</p> <ul style="list-style-type: none"> - Untuk bagian sorting pengurutan dilakukan dengan menggunakan skema standar yang sudah dipelajari di kuliah dengan beberapa cara sorting yang ada seperti seperti selection sort, counting sort, dll. Pilih salah satu untuk diimplementasikan. <p>2. Apakah fungsi help sudah benar dalam menentukan tampilannya?</p> <ul style="list-style-type: none"> - Untuk fungsi help sebelum login diasumsikan sama seperti help untuk role user, tampilan help baru dibedakan berdasarkan role (user atau admin) sesudah login <p>3. Apakah sudah kebayang untuk fungsi load seperti apa?</p> <ul style="list-style-type: none"> - Membuat fungsi load, menyimpan data ke array, toko, game, riwayat memiliki arraynya sendiri. Edit hanya mengedit data array-nya. Fungsi save harus digunakan untuk menyimpan data. Mencatat role yang melakukan login (admin atau user) <p>4. Apakah tugas pertama F01, hanya menuliskan step by step saja atau ada yang harus dijelasinkan?</p> <ul style="list-style-type: none"> - Dari F02 sampai terakhir, fungsi dekomposisi, abstraksi, generalisasi diterapkan. Fungsi-fungsi dipecah menjadi lebih kecil. Di fungsi memiliki parameter data apa yang ingin dibaca. Program diharapkan tidak menumpuk semua di program utama. Coba pelajari dekomposisi lagi Apabila fungsi mirip dibuat satu fungsi saja. Lakukan dekomposisi dengan modular programming. <p>5. Sistem masukin csv nya sudah sesuai atau belum?</p> <ul style="list-style-type: none"> - Untuk csvnya itu ditampung di array terlebih dahulu, setelah itu gunakan fungsi save buat menyimpannya dalam csv. Jika fungsi save tidak diinisiasi maka csv tidak akan mengalami penambahan file.

Tindak Lanjut
<ol style="list-style-type: none">1. Melanjutkan berbagai macam fungsi lain yang masih belum siap2. Mengimplementasikan dekomposisi untuk membuat program yang lebih terstruktur dan tidak menumpuk3. Mengimplementasikan ide skema standar sorting ke dalam program yang akan dibuat4. Melakukan perubahan pada bagian role fungsi help5. Mengatur sistem penambahan data ke dalam csv berdasarkan adanya kondisi fungsi save atau tidaknya

Form MoM Asistensi Tugas Besar

IF1210/Dasar Pemrograman

Sem. 2 2021/2022

Nomor Asistensi	:	
No. Kelompok/Kelas	:	07 / K11
Tanggal asistensi	:	19 April 2022

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	16521499 / M. Andiananto Bismaramadhan
	2	16521505 / Fariz Putra Hanggara
	3	16521511 / Nabilah Amanda Putri
	4	16521541 / Jason Rivalino
Asisten pembimbing	NIM / Nama	
		13519179 / Akifa Nabil

Catatan Asistensi:

Rangkuman Diskusi
<ul style="list-style-type: none">- Array pengguna masih belum diperbaiki- Program belum bisa mencari game berdasarkan kepemilikan- Save menerima nama folder- Pada dekomposisi diminta untuk menunjukkan program sudah terdekomposisi dengan benar atau belum- F01 meminta untuk terdekomposisi dengan file terpisah, misal fungsi searching di file tersendiri, jadi dipanggil ke file utama, main.py diimport dari library yang dibuat sendiri, untuk memanggil semua fungsi yang ada di file utama- Implementasi sorting tidak menggunakan fungsi bawaan atau algoritma bawaan- Masukkan csv ke array lalu csv yang lama ditimpas csv yang baru
Tindak Lanjut
<ul style="list-style-type: none">- Memperbaiki array pengguna- Memisahkan fungsi ke file berbeda- Melanjutkan pengerojaan fungsi lain yang belum diselesaikan- Memperbaiki kekurang dari fungsi yang telah dibuat- Menyelesaikan laporan tugas besar