

LAPORAN TUGAS BESAR

IF1210 DASAR PEMROGRAMAN

Kelas Mahasiswa (K-5) / Kelompok 02

Dosen Pengampu:

Dr. Yusuf Kurniawan, ST,MT

Disusun Oleh:

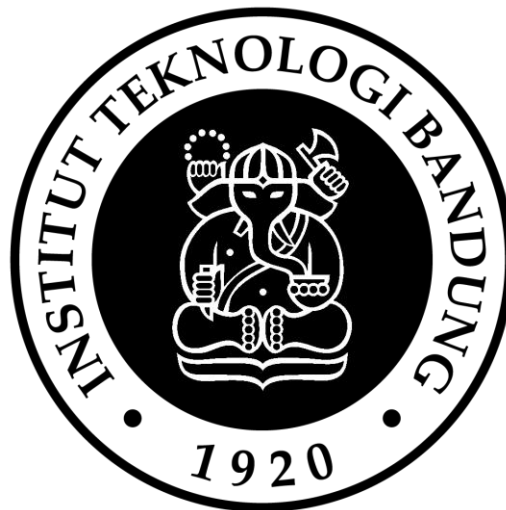
Yanuar Sano Nur Rasyid – 16521176

Firda Sabrina Lutfiyah – 16521239

Mohammad Rifqi Farhansyah – 16521329

Asisten:

Reyhan Emyr Arrosyid – 13519167



SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2022

HALAMAN PERNYATAAN

Yanuar Sano Nur Rasyid – 16521176

“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.”

Firda Sabrina Lutfiyah – 16521239

“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.”

Mohammad Rifqi Farhansyah – 16522329

“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.”

DAFTAR ISI

HALAMAN PERNYATAAN	i
DAFTAR ISI.....	ii
DAFTAR TABEL.....	iii
DAFTAR GAMBAR	iv
DESKRIPSI PERSOALAN	1
DAFTAR PEMBAGIAN KERJA ANGGOTA KELOMPOK.....	4
DAFTAR CHECKLIST HASIL RANCANGAN	6
DESAIN COMMAND SETIAP PRIMITIF	7
DESAIN KAMUS DATA	19
DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL PROGRAM	25
Dekomposisi Algoritmik Program	25
Dekomposisi Fungsional Program	30
SPESIFIKASI MODUL/PROSEDUR/FUNGSI	36
SCREENSHOT HASIL PENGUJIAN PROGRAM	40

DAFTAR TABEL

Tabel 1. Tabel Pembagian Kerja Anggota Kelompok	4
Tabel 2. Tabel Checklist Hasil Rancangan	6

DAFTAR GAMBAR

Gambar 1 Dekomposisi Fungsional Program	30
Gambar 2 Dekomposisi Fungsional F05	31
Gambar 3 Dekomposisi Fungsional F06	31
Gambar 4 Dekomposisi Fungsional F07	32
Gambar 5 Dekomposisi Fungsional F08	32
Gambar 6 Dekomposisi Fungsional F09	32
Gambar 7 Dekomposisi Fungsional F10	33
Gambar 8 Dekomposisi Fungsional F11	33
Gambar 9 Dekomposisi Fungsional F13	33
Gambar 10 Dekomposisi Fungsional F15	34
Gambar 11 Dekomposisi Fungsional F16	34
Gambar 12 Dekomposisi Fungsional F1	35
Gambar 13 Dekomposisi Fungsional B02l	35
Gambar 14 Eksekusi F01 Argumen Folder Tidak Diberikan	41
Gambar 15 Eksekusi F01 Argumen Folder Tidak Ditemukan	41
Gambar 16 Eksekusi F01 Tampilan Antarmuka	41
Gambar 17 Eksekusi F02 User Terdaftar	41
Gambar 18 Eksekusi F02 Username Telah Terpakai	41
Gambar 19 Eksekusi F03 Password atau Username Salah	42
Gambar 20 Eksekusi F03 Login Berhasil	42
Gambar 21 Eksekusi F04 Mengembalikan Pesan Error	42
Gambar 22 Eksekusi F04 Mengembalikan Pesan Sukses	42
Gambar 23 Eksekusi F05 ID Tidak Boleh Kosong	43
Gambar 24 Eksekusi F05 ID Tidak Ada	43
Gambar 25 Eksekusi F05 Mengembalikan Pesan Sukses	43
Gambar 26 Eksekusi F06 ID Tidak Ada	44
Gambar 27 Eksekusi F06 Stok Game Kurang	44
Gambar 28 Eksekusi F06 Stok Game Berhasil Ditambah	44
Gambar 29 Eksekusi F07 Sorting Tahun Ascending	44
Gambar 30 Eksekusi F07 Sorting Harga Descending	45
Gambar 31 Eksekusi F07 Sorting ID	45
Gambar 32 Eksekusi F08 Game Telah Dimiliki	45
Gambar 33 Eksekusi F08 Stok Game Sedang Habis	45
Gambar 34 Eksekusi F08 Game Berhasil Dibeli	46
Gambar 35 Eksekusi F09 Mengembalikan List Game User	46
Gambar 36 Eksekusi F10 Game Tidak Ditemukan	46
Gambar 37 Eksekusi F10 Game Berhasil Ditemukan	46
Gambar 38 Eksekusi F11 Daftar Game Sesuai Kriteria	47
Gambar 39 Eksekusi F11 Game Tidak Ditemukan	47
Gambar 40 Eksekusi F12 Username Tidak Ditemukan	47
Gambar 41 Eksekusi F12 Masukkan Saldo Tidak Valid	48
Gambar 42 Eksekusi F12 Top Up berhasil	48
Gambar 43 Riwayat Pembelian User	48

Gambar 44 Eksekusi F14 untuk role admin	48
Gambar 45 Eksekusi F14 untuk role user.....	49
Gambar 46 Eksekusi F16 Keberhasilan Proses Penyimpanan.....	49
Gambar 47 Eksekusi F17 Input Tidak Valid	49
Gambar 48 Eksekusi F17 tanpa Penyimpanan.....	49
Gambar 49 Eksekusi F12 dengan Penyimpanan	49
Gambar 50 Eksekusi B02 Program Kerang Ajaib.....	50
Gambar 51 Eksekusi B03 Masukkan Tidak Valid.....	50
Gambar 52 Eksekusi B03 Hasil Permainan Imbang.....	51
Gambar 53 Eksekusi B03 Hasil Akhir Permainan	51

DESKRIPSI PERSOALAN

Program ini merupakan perbaikan dari sistem inventarisasi & toko game BNMO yang telah mengalami kerusakan. Program dibuat menggunakan bahasa pemrograman Python. Sistem inventarisasi dan toko game ini memiliki fitur yang sangat canggih karena pengguna dapat melakukan berbagai macam perintah dengan cara mengetik *command* yang telah disediakan sesuai dengan kehendak mereka. Fitur-fitur yang disediakan oleh program perbaikan ini, antara lain:

1. Register

Bagi pengguna yang baru pertama kali mengakses BNMO, pengguna perlu melakukan registrasi melalui admin. Admin akan memasukkan username, nama, dan password pengguna. Data ini akan disimpan di dalam database sebagai *user*. Data ini kemudian akan digunakan untuk melakukan login pada sistem.

2. Login

Untuk mengakses fitur-fitur pada BNMO, pengguna wajib melakukan login. Pengguna akan memasukkan username dan password yang akan dicocokkan dengan data pada database. Apabila cocok, pengguna dapat mengakses fitur-fitur pada BNMO.

3. Menambah Game ke Toko Game

Admin dapat menambahkan game ke toko game dengan memasukkan informasi game berupa nama game, kategori game, tahun rilis, harga, dan stok game. Semua informasi tersebut harus terisi. Jika ada informasi yang kosong, program akan menanyakan kembali informasi game hingga seluruh informasi terisi. Selanjutnya, informasi ini akan disimpan di database *game*.

4. Mengubah Game pada Toko Game

Admin dapat menggunakan fitur ini untuk mengubah atribut dari suatu game di toko, kecuali id game dan stok. Input dari fungsi ini tidak harus diisi semuanya yang penting hanya ide game yang diinput. Jika pada pengisian atribut dikosongkan maka data game yang sedang diubah akan sama dengan data pada keadaan sebelumnya. Setelah proses selesai maka data akan disimpan kembali ke variabel pada program.

5. Mengubah Stok Game di Toko

Admin dapat menggunakan fitur ini untuk menambah atau mengurangi stok yang ada di toko. ini sendiri mempengaruhi kemampuan user untuk membeli sebuah game. Admin dapat memasukkan angka positif untuk menambah stok dan angka negatif untuk mengurangi stok. Akan tetapi, pada pengurangan stok, jika input yang dimasukkan lebih besar daripada stok yang tersedia maka program tidak akan mengubah data stok tersebut. Setelah selesai diubah data stok ini akan disimpan kembali di variabel program.

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

Pengguna atau admin dapat mengakses list game dalam toko yang telah di-*sorting* berdasarkan harga dan tahun rilisnya. Pengguna atau admin dapat memberikan masukan berupa skema *sorting* dari daftar game yang ingin dilihat ke dalam aplikasi. Terdapat 2 skema *sorting*, yaitu berdasarkan tahun rilis atau harga, dengan variasi *ascending* maupun *descending*. Jika pengguna atau admin tidak memberi masukan skema *sorting*, maka list game akan di-*sorting* berdasarkan ID *ascending*. Apabila pengguna memasukkan skema *sorting* diluar lingkup yang telah disebutkan, maka akan muncul pesan “skema *sorting* tidak valid!”. Melalui fitur ini, pengguna

dapat menyesuaikan kebutuhannya dalam mencari game yang *up-to-date*, klasik, ataupun game dengan budget yang minim.

7. Membeli Game

Pengguna memiliki akses untuk membeli sebuah game yang ada di toko dengan memasukkan ID Game yang hendak dibeli. Game yang telah dibeli akan masuk ke list Game yang dimiliki oleh pengguna. Data pembelian game oleh pengguna akan tercatat pada data storage/CSV hanya jika dilakukan perintah save. Keberhasilan pembelian game oleh pengguna memperhatikan beberapa syarat, antara lain: Game hanya dapat dibeli user yang sama sebanyak satu kali, saldo yang dimiliki oleh pengguna harus mencukupi, serta stok game yang hendak dibeli harus tersedia.

8. Melihat Game yang dimiliki

Pengguna dapat melihat daftar game yang dimilikinya. Daftar game yang dimiliki pengguna dapat dilihat melalui riwayat transaksi yang tercatat oleh sistem. Ketika pengguna belum memiliki / belum melakukan pembelian sebuah game, maka akan muncul pesan “Maaf, kamu belum membeli game. Ketik perintah `beli_game` untuk beli.”

9. Mencari Game yang dimiliki dari ID dan tahun rilis

Pengguna program dapat mencari game dengan parameter ID game dan/atau tahun rilis dari game yang sudah dimiliki. Fitur ini akan menerima dua input tersebut dan mencari game yang sesuai dengan input yang sudah dimasukkan. Input ini bersifat opsional. Setelah selesai bekerja akan terlihat daftar game yang memenuhi parameter di layar.

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

Selain mencari game yang sudah dimiliki, pengguna program juga dapat mencari game yang ada di toko sampai dengan lima parameter yaitu ID, nama, harga, kategori, dan tahun. Sama seperti program sebelumnya, input program ini bersifat opsional. Setelah memasukkan input, program akan mencari data game di toko yang sesuai dengan input pengguna dan menampilkannya di layar.

11. Top Up Saldo

Admin dapat menambahkan saldo pada user. Admin cukup memasukkan username dan jumlah saldo, lalu saldo akan ditambahkan atau dikurangi dari saldo awal pengguna. Saldo dapat bernilai positif (saldo bertambah) ataupun negatif (saldo berkurang). Apabila saldo bernilai negatif, dilakukan validasi untuk memastikan bahwa saldo yang dikurangi tidak melebihi nominal saldo awal.

12. Melihat Riwayat Pembelian

Pengguna atau user dapat melihat riwayat pembelian game yang telah dimiliki pengguna. Namun, ketika pengguna belum melakukan transaksi pembelian game, maka akan muncul pesan “Maaf, kamu tidak ada riwayat pembelian game. Ketik perintah `beli_game` untuk membeli.”

13. Help

Pengguna ataupun admin dapat mengakses fitur help. Fitur ini memberikan informasi untuk membantu pengguna ataupun admin dalam menjalankan fitur-fitur pada program. Pengguna ataupun admin tidak perlu login untuk mengakses fitur ini, tetapi jika mereka telah login, fitur help yang ditampilkan akan disesuaikan dengan role pada saat login.

14. Load

Saat menjalankan program, fitur load ini otomatis dijalankan. Fitur ini berfungsi sebagai pembaca data dari file .csv yang tersimpan di komputer agar dapat dibaca dan diolah oleh program. Fitur ini membutuhkan input di terminal yaitu folder yang menyimpan file .csv yang akan diolah. Data .csv akan dibaca dan disimpan menjadi sebuah array of array of string yang dapat dibaca oleh program.

15. Save

Untuk menyimpan data yang sudah diolah oleh program, user dapat menggunakan fitur ini. Saat dijalankan, fitur ini akan menerima sebuah input folder yang akan berfungsi sebagai tempat folder penyimpanan data dari program. Langkahnya program akan membuat folder jika belum ada kemudian akan menuliskan kembali data yang sudah diubah oleh load kembali menjadi data .csv

16. Exit

Fitur ini digunakan untuk keluar dari sistem. Sebelum keluar, program akan menampilkan pilihan apakah perubahan yang dilakukan user ataupun admin ingin disimpan atau tidak.

DAFTAR PEMBAGIAN KERJA ANGGOTA KELOMPOK

Tabel 1. Tabel Pembagian Kerja Anggota Kelompok

Fitur	Implementasi	NIM Desiner	NIM Coder	NIM Tester
F02 – Register	procedure register	16521239	16521239	16521176
F03 – Login	function login	16521239	16521239	16521239
F04 – Menambah Game ke Toko Game	procedure tambah_game	16521239	16521239	16521176
F05 – Mengubah Game pada Toko Game	procedure len procedure ubah_game	16521176	16521176	16521176
F06 – Mengubah Stok Game di Toko Game	procedure len procedure ubah_stok	16521176	16521176	16521176
F07 – Listing Game di Toko Berdasarkan ID, Tahun Rilis, dan Harga	procedure list_game_toko function filter_tahun_plus function filter_tahun_minus function filter_harga_plus function filter_harga_minus	16521329	16521329	16521329
F08 – Membeli Game	procedure buy_game	16521329	16521329	16521329 16521176
F09 – Melihat Game yang Dimiliki	function melihat_game_user	16521329	16521329	16521329
F10 – Mencari Game yang Dimiliki dari ID dan Tahun Rilis	procedure len procedure search_in procedure no_duplicate procedure pop	16521176	16521176	16521176

	procedure remove_not_ma tch procedure search_my_gam e			
F11 – Mencari Game di Toko dari ID, Nama Game, Harga, Kategori, dan Tahun Rilis	procedure len procedure search_cat procedure pop procedure remove_not_ma tch procedure search_my_gam e	16521176	16521176	16521176
F12 – Top Up Saldo	procedure topup	16521239	16521239	16521176
F13 – Melihat Riwayat Pembelian	procedure lihat_riwayat	16521329	16521329	16521329
F14 – Help	procedure help	16521239	16521239	16521239
F15 – Load	procedure len procedure read_csv procedure inside procedure load	16521176	16521176	16521176
F16 – Save	procedure len procedure write_csv procedure save procedure save_data	16521176	16521176	16521176
F17 – Exit	procedure exit function save	16521239	16521239	16521239
B02 – Magic Conch Shell	procedure lcg procedure kerang_ajaib	16521176	16521176	16521176
B03 – Game Tic-Tac-Toe	procedure tictactoe	16521329	16521329	16521329

DAFTAR CHECKLIST HASIL RANCANGAN

Tabel 2. Tabel Checklist Hasil Rancangan

Fitur	Desain	Implementasi	Testing
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	✓	✓	✓
B01 - Cipher	x	x	x
B02 – Magic Conch Shell	✓	✓	✓
B03 – Game Tic-Tac-Toe	✓	✓	✓

DESAIN COMMAND SETIAP PRIMITIF

F02 - Register

Nama command: register

Input:

Masukkan nama:

Masukkan username:

Masukkan password:

Kemungkinan 1:

Output:

<tidak ada output, data register tersimpan di database>

Kemungkinan 2:

Output:

Maaf! Username sudah dipakai.

Kemungkinan 3:

Output:

Username harus terdiri atas alfabet, angka, _, dan -

F03 - Login

Nama command: login

Input:

Masukkan username:

Masukkan password:

Kemungkinan 1:

Output:

Halo <nama user>! Selamat datang di "Binomo"

Kemungkinan 2:

Output:

Password atau username salah atau tidak ditemukan.

F04 - Menambahkan Game ke Toko Game

Nama command: tambah_game

Input:

Masukkan nama game:

Masukkan kategori:

Masukkan tahun rilis:

Masukkan harga:

Masukkan stok awal:

Kemungkinan 1:

Output:

Selamat! Berhasil menambahkan game <nama game>

Kemungkinan 2:

Output:

Mohon masukkan semua informasi mengenai game agar dapat disimpan di BNMO

F05 - Mengubah Game pada Toko Game

Nama command : ubah_game

Input :

Masukkan ID game:

Masukkan nama game:

Masukkan kategori game:

Masukkan tahun rilis game:

Masukkan harga game:

Kemungkinan 1

Output:

Game berhasil diubah

Kemungkinan 2

Output :

Tidak ada id tersebut.

Kemungkinan

Output :

ID tidak boleh kosong.

F06 - Mengubah Stok Game di Toko

Nama command : ubah_stok

Input :

Masukkan ID game:

Kemungkinan 1

Input:

Masukkan stok:

Kemungkinan 1.1

Output:

Stok harus berupa angka

Kemungkinan 1.2

Output :

Stok tidak boleh kosong!

Kemungkinan 2

Kemungkinan 2.1

Output:

Stok game <game> berhasil ditambah. Stok sekarang: <stok>

Kemungkinan 2.2

Output :

Stok game <game> berhasil dikurang. Stok sekarang: <stok>

Kemungkinan 2.3

Output :

Stok game <game> gagal dikurang karena stok kurang. Stok sekarang: <stok>

Kemungkinan 2.3

Output :

Tidak ada id tersebut.

Kemungkinan 3

Output :

ID tidak boleh kosong.

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

Nama command : list_game_toko

Input:

Skema sorting:

Kemungkinan 1:

Output:

```
{ID} | {NAMA} | {HARGA} | {Kategori} | {Tahun rilis} | {Stok}
```

Kemungkinan 2:

Output:

Skema sorting tidak valid!

F08 – Membeli Game

Nama command : buy_game

Input:

Masukkan ID Game:

Kemungkinan 1:

Output:

Game "{NAMA}" berhasil dibeli!

Kemungkinan 2:

Output:

Anda sudah memiliki Game tersebut!

Kemungkinan 3:

Output:

Saldo anda tidak cukup untuk membeli Game tersebut!

Kemungkinan 4:

Output:

Stok Game tersebut sedang habis!

F09 – Melihat Game yang dimiliki

Nama command : list_game

Kemungkinan 1:

Output:

Daftar game:

```
{ID} | {Nama game} | {Kategori} | {Tahun Rilis} | {Harga}
```

Kemungkinan 2:

Output:

Maaf, kamu belum membeli game. Ketik perintah beli_game untuk beli.

F10 - Mencari Game yang dimiliki dari ID dan tahun rilis

Nama command : search_my_game

Kemungkinan 1

Input :

Masukkan ID game:

Masukkan tahun rilis game:

Kemungkinan 1.1

Output:

Game tidak ditemukan

Kemungkinan 1.2

Output:

```
<id> , <nama>, <kategori>, <tahun_rilis>,<harga>,<stok>
```

Kemungkinan 2

Input :

Masukkan ID game:

Kemungkinan 2.1

Output:

Game tidak ditemukan

Kemungkinan 2.2

Output:

<id> , <nama>, <kategori>, <tahun_rilis>,<harga>,<stok>

Kemungkinan 3

Input :

Masukkan tahun rilis game:

Kemungkinan 3.1

Output:

Game tidak ditemukan

Kemungkinan 3.2

Output:

<id> , <nama>, <kategori>, <tahun_rilis>,<harga>,<stok>

Kemungkinan 4

Output:

Tidak ada input yang dimasukkan.

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

Nama command : search_game_at_store

Input :

Masukkan ID game:

Masukkan nama game:

Masukkan kategori game:

Masukkan tahun rilis game:

Masukkan harga game:

Kemungkinan 1**Output:**

Game tidak ditemukan

Kemungkinan 2**Kemungkinan 2.1****Output:**

Game tidak ditemukan

Kemungkinan 2.2**Output:**

<id> , <nama>, <kategori>, <tahun_rilis>,<harga>,<stok>

F12 - Top Up Saldo

Nama command: topup

Input:

Masukkan username:

Masukkan saldo:

Kemungkinan 1:**Output:**

Selamat. Berhasil menambahkan saldo.

Kemungkinan 2:

Output:

Masukan tidak valid. Saldo Anda tidak mencukupi.

Kemungkinan 3:

Output:

Username <input username> tidak ditemukan.

F13 – Melihat Riwayat Pembelian

Nama command : riwayat

Kemungkinan 1:

Output:

Daftar game:

{ID} | {Nama game} | {Harga} | {Tahun Beli}

Kemungkinan 2:

Output:

Maaf, kamu tidak ada riwayat pembelian game. Ketik perintah beli_game untuk membeli.

F14 - Help

Nama command: help

Kemungkinan 1:

Output:

1. register - untuk melakukan registrasi user baru.

2. login - untuk melakukan login ke sistem.
3. tambah_game - untuk menambahkan game ke toko
4. ubah_game - untuk mengubah game pada toko.
5. ubah_stok - untuk mengubah stok game di toko
6. list_game_toko - untuk melakukan sorting daftar game sesuai kategori tertentu.
7. search_game_at_store - untuk mencari game di toko.
8. topup - untuk menambahkan saldo user.
9. save - untuk melakukan penyimpanan pada file yang diubah.

Kemungkinan 2:

Output:

1. login - untuk melakukan login ke sistem.
2. list_game_toko - untuk melakukan sorting pada game di toko sesuai kategori tertentu.
3. buy_game - untuk membeli game di toko.
4. list_game - untuk melihat game yang dimiliki user.
5. search_my_game - untuk mencari game yang dimiliki berdasarkan id dan tahun rilis.
6. search_game_at_store - untuk mencari game di toko berdasarkan kategori tertentu.
7. riwayat - untuk melihat riwayat pembelian.
8. save - untuk menyimpan perubahan pada file.

F15 – Load

Nama command : load

Kemungkinan 1

Output:

Tidak ada nama folder yang diberikan

Kemungkinan 2

Input :

Masukkan nama folder:

Kemungkinan 2.1

Output:

Folder <folder> tidak ditemukan

Kemungkinan 2.2

Output:

Folder <folder> ditemukan

F16 – Save

Nama command : save_data

Kemungkinan 1

Input:

Masukkan nama folder penyimpanan

Kemungkinan 1.1

Output:

Saving ...

Data berhasil disimpan di folder <folder>

Kemungkinan 2

Input :

Tidak ada input yang dimasukkan. Kembali ke menu

F17 - Exit

Nama command: exit

Input:

Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n):

Kemungkinan 1:

Output:

saving...
Sampai jumpa!

Kemungkinan 2:

Output:

Sampai jumpa

Kemungkinan 3:

Output:

Input tidak valid

B02 - Magic Conch Shell

Nama command : kerang_ajaib

Input:

Apa pertanyaanmu?

Output:

<wisdom>

DESAIN KAMUS DATA

Kamus Global

{Deklarasi Variabel}

data_game , data_user, data_riwayat, data_kepemilikan : array [1..Nmax] of array [1..Idmax] of string

user_id , role : string

menu : string

Kamus Lokal Tiap Fitur

F02 – register

{deklarasi variabel}

name, user_name, pword: string

ID: integer

F03 - login

{deklarasi variabel}

user_name, pword: string

F04 – Menambah Game ke Toko Game

{deklarasi variabel}

nama_game, kategori, tahun rilis, harga, stok: string

F05 - Mengubah Game pada Toko Game

{Deklarasi Variabel}

i : integer

data : array [1..Nmax] of array [1..Idmax] of string

id , nama, kategori, tahun_rilis, harga : string

found : boolean

F06 - Mengubah Stok Game di Toko

{Deklarasi Variabel}

i : integer

data : array [1..Nmax] of array [1..Idmax] of string

id , stok : string

found : boolean

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

{Deklarasi Variabel}

filterlist : string

banyakbaris : int

F08 – Membeli Game

{Deklarasi Variabel}

id_game : string

F09 – Melihat Game yang Dimiliki

{Deklarasi Variabel}

properti_game : array

data_game_user : matrix

F10 - Mencari Game yang dimiliki dari ID dan tahun rilis

{Deklarasi Variabel}

i : integer

data : array [1..Nmax] of array [1..Idmax] of string

hasil : array [1..Nmax] of array [1..Idmax] of string

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

{Deklarasi Variabel}

i : integer
data : array [1..Nmax] of array [1..Idmax] of string
id , nama, kategori, tahun_rilis, harga : string
data_nama, data_kategori, data_tahun, data_harga : array [1..Nmax] of array [1..Idmax] of string
total, hasil, akhir : array [1..Nmax] of array [1..Idmax] of string

F12 – Top Up Saldo

{deklarasi variabel}

user_name: string

saldo: integer

F13 - Melihat Riwayat Pembelian

{Deklarasi Variabel}

user_id : string

panjang_riwayat : int

cek : bool

F14 – Help

{deklarasi variable}

role_user = string

F15 – Load

{Deklarasi Tipe}

type parser : < folder : string >

{Deklarasi Variabel}

args : parser

list_folder : array of string

found : boolean

i : integer

F16 – Save

{Deklarasi Variabel}

folder : string

F17 - Exit

{deklarasi variabel}

x = character

B02 - Magic Conch Shell

{Deklarasi Variabel}

wisdom : array of string

m : integer

num : integer

B03 – TicTacToe

{Deklarasi Variabel}

T : array [1..3] of array [1..3] of char

cek : boolean

baris, kolom : int

Kamus lokal yang digunakan pada beberapa fungsi dan prosedur lainnya

{Fungsi length_manual}

list : list

length : int

{Fungsi append_manual}

list_awal : list

item_baru : str, int, float, list

length_list_awal, indeks_terakhir : int

list_baru : list

{Fungsi split_manual}

```

    list_awal : list
    pemisah : char
{Fungsi remove_manual}

    list_baru : list
    index : int
{Fungsi max_length}

    kolom, max : int
{Prosedur print_matrix}

    baris, kolom : int
    maxlencolumn : array [NMin_of_kolom..NMax_of_kolom] of int
{Fungsi len}
count, i : integer
    data : array
{Fungsi inside}

    i : integer
    list : array of string
{Fungsi pop}
i : integer
    data : array of string
    temp : array of string
{Fungsi no_duplicate}

    i : integer
    list1 : array [1..Nmax] of array [1..Idmax] of string
    list2 : array [1..Nmax] of array [1..Idmax] of string
    temp : array [1..Nmax] of array [1..Idmax] of string

{Fungsi search_cat}

    i : integer
    data : array [1..Nmax] of array [1..Idmax] of string
    hasil : array [1..Nmax] of array [1..Idmax] of string

```

{Fungsi search_in}

i : integer

data : array [1..Nmax] of array [1..Idmax] of string

hasil : array [1..Nmax] of array [1..Idmax] of string

{Fungsi remove_not_match}

i : integer

data : array [1..Nmax] of array [1..Idmax] of string

{Fungsi read_csv}

seperator : string

f : seqfile of

(line) : array [1..Nmax] of array [1..Idmax] of string

(1) : mark

batas, i : integer

result : array of string

word : array of char

count : integer

data : array [1..Nmax] of array [1..Idmax] of string

{Prosedur write_csv}

f : seqfile of

(line) : array [1..Nmax] of array [1..Idmax] of string

(1) : mark

i, j : integer

DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL PROGRAM

Dekomposisi Algoritmik Program

F02 - Register

{Procedure register}

{I.S. : program berjalan dan pengakses adalah admin}

Masukkan nama, username, dan password akun yang ingin dibuat.

{F.S/I.S : nama, username, dan password user dimasukkan, role dibuat default user, saldo default 0}

Mengecek apakah username mengandung character yang dilarang

{F.S/I.S : username sudah dicek, apabila tidak ada character terlarang, username berlanjut di program }

Mengecek jumlah baris pada *user*

{F.S/I.S : jumlah baris dihitung dan disimpan sebagai ID}

Mengecek apakah username sudah ada di *user*

{F.S/I.S : username sudah dicek keberadaannya, apabila belum ada, siap untuk ditambahkan ke *user*}

Tambahkan ID, nama, username, password, role, dan saldo ke *user*

{F.S. : ID, nama, username, password ditambahkan ke *user*}

F03 - Login

{Function login}

{I.S. : program berjalan, pengguna belum melakukan login}

Masukkan username dan password

{F.S./I.S. : username dan password telah ditentukan, program mengecek username yang dimasukkan}

Mencari dan mencocokkan input username dan password pada *user*

{F.S. : jika ditemukan, login berhasil. Jika tidak ditemukan, login gagal}

F04 - Menambahkan Game ke Toko Game

{procedure tambah_game}

{I.S. : program berjalan dan diakses oleh admin}

Masukkan nama game, kategori, tahun rilis, harga, dan stok awal game

{F.S./I.S. : masukkan dicek apakah terisi semua atau ada data kosong}

{F.S./I.S. : program telah dicek dan semua data terisi, data disimpan dalam list bersiap untuk ditambahkan ke *game*}

{F.S. : nama game, kategori, tahun rilis, harga, stok awal game ditambahkan ke *game*}

F05 - Mengubah Game pada Toko Game

{Fungsi ubah_game}

{I.S. : program berjalan (sudah load data) dan pengakses adalah admin}

Men-input id, nama, kategori, tahun rilis, harga

{F.S./I.S. : id tidak kosong dan terdapat di data_game. Jika tidak berhenti}

Mengubah data nama, kategori, tahun rilis, dan harga jika ada. Jika tidak tetap

{F.S. : data disimpan lagi ke variabel}

F06 - Mengubah Stok Game di Toko

{Fungsi ubah_stok}

{I.S. : program berjalan (sudah load data) dan pengakses adalah admin}

Memasukkan id dan stok

{F.S./I.S. : id dan stok tidak kosong dan stok berupa angka. Jika tidak berhenti}

{F.S./I.S. : data stok ditambah dengan stok input. Jika hasil akhir negatif berhenti}

Mengubah data stok variabel dengan stok yang baru

{F.S. : data disimpan lagi ke variabel}

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

{prosedur list_game_toko}

{I.S. : program berjalan (sudah load data) dan sudah login}

Memasukkan skema sorting

{F.S./I.S. melakukan sorting sesuai dengan masukkan skema sorting dari pengguna}

{F.S. mencetak ke layar list game di toko sesuai dengan skema sorting hasil masukan user}

F08 – Membeli Game

{prosedur buy_game}

{I.S. : program berjalan (sudah load data) dan sudah login}

Memasukkan id_game

{F.S. menghasilkan keluaran sesuai kondisi yang dipenuhi oleh user}

F09 – Melihat Game yang dimiliki

{prosedur list_game}

{I.S. : program berjalan (sudah load data) dan sudah login}

{F.S/I.S. melakukan pengecekan terkait data dalam game yang sesuai dengan user id dari pengguna yang melakukan login}

{F.S. mencetak ke layar informasi list game yang dimiliki oleh pengguna}

F10 - Mencari Game yang dimiliki dari ID dan tahun rilis

{Prosedur search_my_game }

{I.S. : program berjalan (sudah load data) dan sudah login}

Input id game dan tahun rilis

{F.S/I.S : menerima id_user dan mencari daftar game yang dimiliki}

{F.S/I.S : mencari id game dan tahun yang sama dari daftar game yang dicari sebelumnya.

Jika tidak ditemukan yang sama berhenti}

{F.S. : Output hasil pencarian}

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

{Prosedur search_game_at_store}

{I.S. : program berjalan (sudah load data) dan sudah login}

Input id, nama, kategori, tahun rilis, harga

{F.S/I.S : search setiap parameter jika ada inputnya. Jika tidak deklarasi array kosong.}

{F.S/I.S : Jumlahkan semua hasil pencarian sebelumnya kemudian gunakan no_duplicate agar tidak muncul lebih dari sekali. Jika jumlahnya kosong berhenti}

{F.S/I.S : Tergantung dari banyaknya parameter yang digunakan lakukan operasi fungsi `remove_not_match` agar sesuai dengan semua parameter yang dipakai.}

{F.S. : Output hasil `remove_not_match` terakhir.}

F12 - Top Up Saldo

{procedure `topup`}

{I.S. : program berjalan dan diakses ole admin}

Masukkan username dan saldo

{F.S./I.S. : program mengecek apakah username tersimpan dalam *user*}

{F.S./I.S. : username ditemukan}

input saldo ditambahkan ke saldo awal

{F.S./I.S. : saldo awal diganti dengan saldo akhir }

{F.S./I.S. : program membaca baris yang terdapat username seperti masukan}

{F.S. : ID, nama, username, password, role, dan saldo akhir ditambahkan ke *user*}

F13 - Melihat Riwayat Pembelian

{prosedur riwayat}

{I.S. : program berjalan (sudah load data) dan sudah login}

{F.S/I.S : program akan melakukan pengecekan terkait riwayat pembelian yang sesuai dengan user id dari pengguna yang login}

{F.S. : program akan mencetak daftar riwayat pembelian pengguna ke layar}

F14 - help

procedure `help`}

{I.S. : program berjalan, pengakses adalah user atau admin}

cek role pengakses

{F.S./I.S. : role pengakses ditemukan}

cetak command `help`

{F.S. : program berhasil dijalankan, tampilan menyesuaikan role pengakses (admin/user)}

F15 – Load

{Fungsi load}

{I.S. : Jalankan program dan masukan argumen folder}

{F.S/I.S : Jika argumen kosong berhenti. Jika tidak cari foldernya di lokasi programnya.}

{F.S/I.S : Jika tidak ditemukan berhenti. Lakukan read_csv terhadap file csv yang ada di dalam folder.}

{F.S. : Simpan hasil baca ke dalam variabel.}

F16 – Save

{Prosedur save_data}

{I.S. : program berjalan (sudah load data) dan sudah login}

Input folder

{F.S/I.S : mencari folder di lokasi source code. Jika tidak ada buat folder.}

{F.S/I.S : Lakukan write_csv terhadap setiap data yang dipakai}

{F.S. : Terdapat file csv di folder.}

F17 – Exit

{procedure exit_program}

{I.S. : program berjalan}

Masukkan input (y/n) apakah perubahan ingin disimpan

{F.S. : Jika input adalah ‘y’ atau ‘Y’, perubahan disimpan. Jika input adalah ‘n’ atau ‘N’, program segera berhenti.}

B02 - Magic Conch Shell

{Prosedur kerang_ajaib}

{I.S. : program berjalan (sudah load data)}

Input pertanyaan

{F.S/I.S : mendapatkan angka n dari lcg.}

{F.S. : Output wisdom[n] .}

B03 – TicTacToe

{prosedur TicTacToe}

{I.S. : program berjalan (sudah load data)}

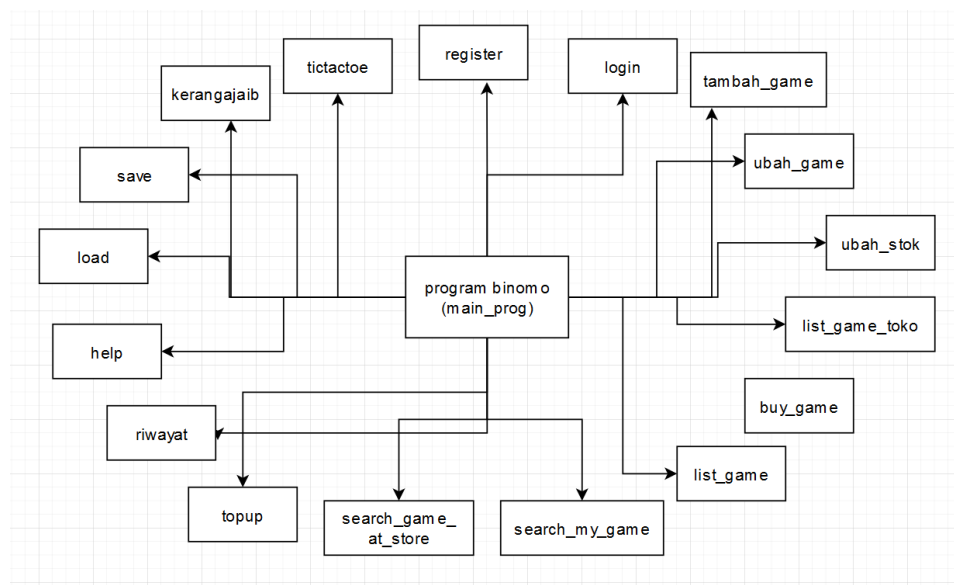
{F.S./I.S. : program diinisialisasi dengan matriks berukuran 3x3 yang masih kosong}

Masukkan input baris dan kolom untuk tiap giliran pemain

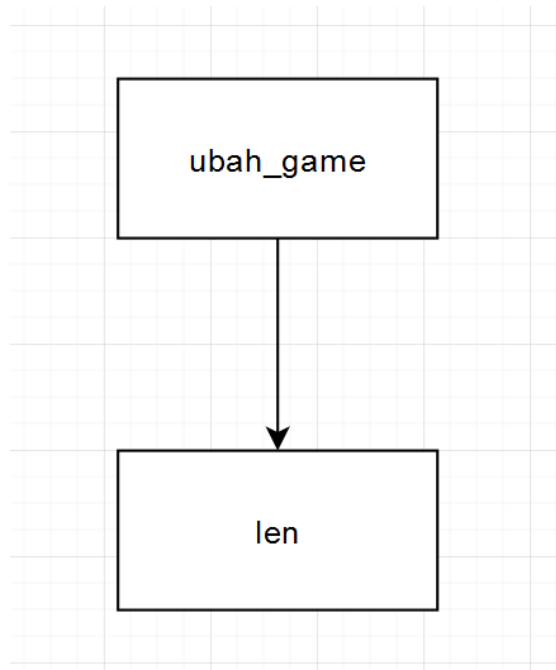
{F.S./I.S. : program akan memperhitungkan pemain mana yang terlebih dahulu dapat membuat sebuah garis lurus atau diagonal dalam matriks}

{I.S : program akan mencetak ke layar terkait informasi pemenang permainan}

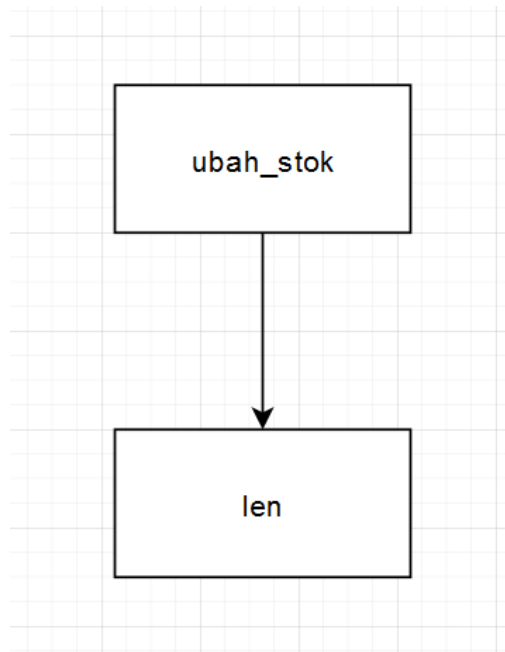
Dekomposisi Fungsional Program



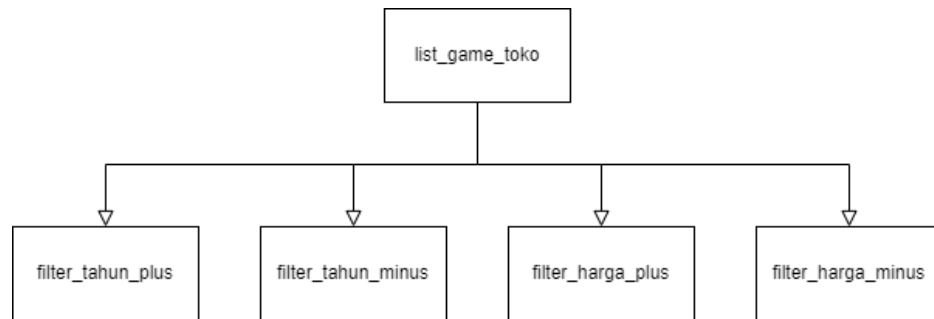
Gambar 1 Dekomposisi Fungsional Program



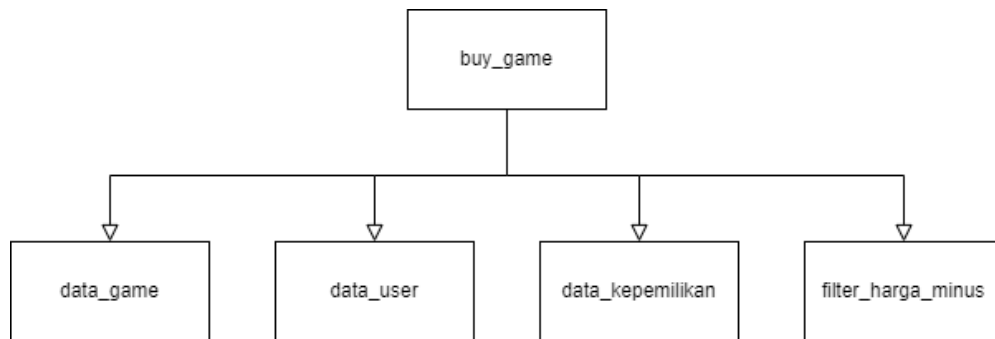
Gambar 2 Dekomposisi Fungsional F05



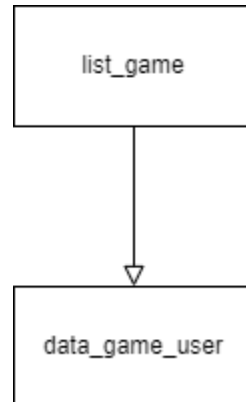
Gambar 3 Dekomposisi Fungsional F06



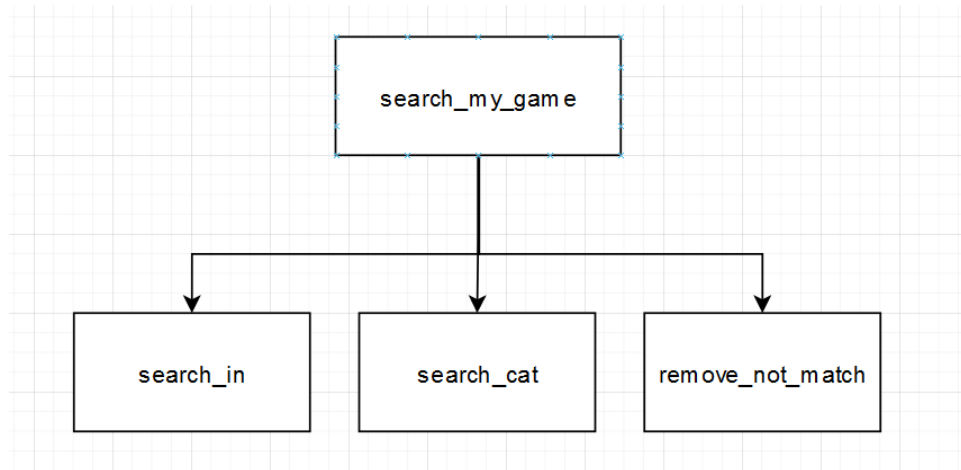
Gambar 4 Dekomposisi Fungsional F07



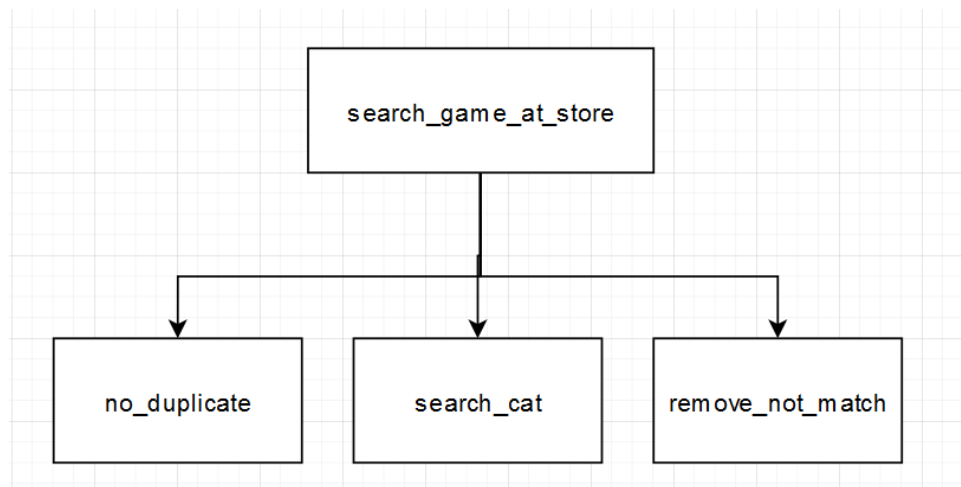
Gambar 5 Dekomposisi Fungsional F08



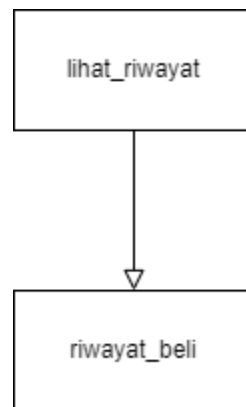
Gambar 6 Dekomposisi Fungsional F09



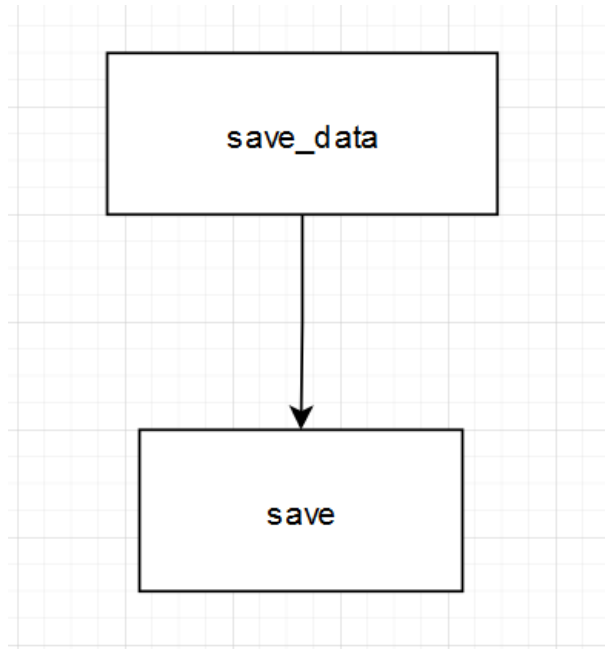
Gambar 7 Dekomposisi Fungsional F10



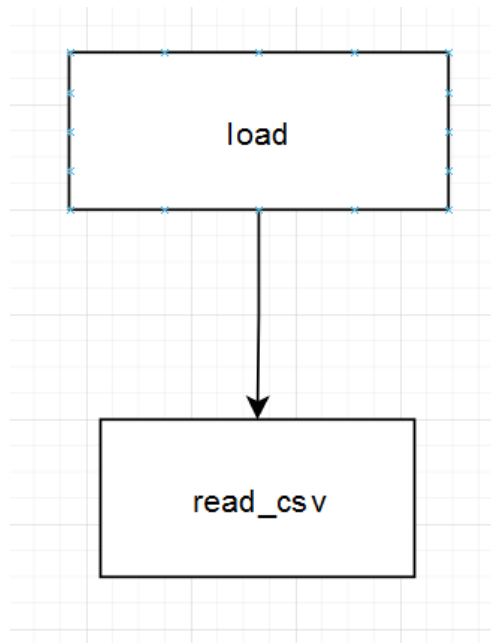
Gambar 8 Dekomposisi Fungsional F11



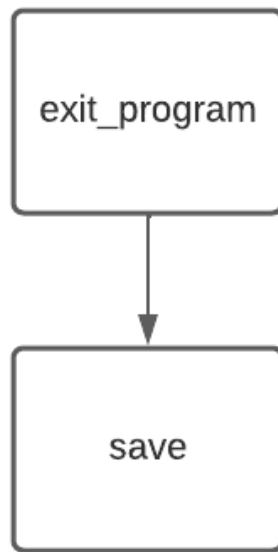
Gambar 9 Dekomposisi Fungsional F13



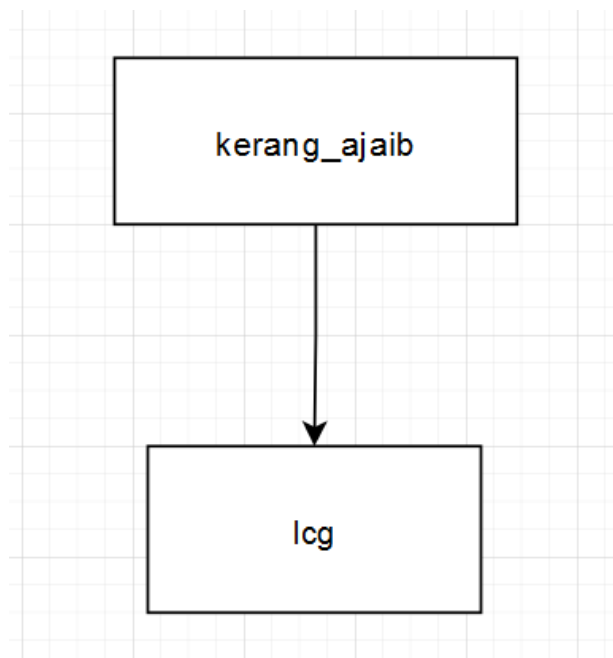
Gambar 10 Dekomposisi Fungsional F15



Gambar 11 Dekomposisi Fungsional F16



Gambar 12 Dekomposisi Fungsional F1



Gambar 13 Dekomposisi Fungsional B02I

SPEKIFIKASI MODUL/PROSEDUR/FUNGSI

procedure register (input/output data_register : array [1...Nmax] of data_user)

{Memasukkan user_name ke database untuk memastikan username unik}

{Mengambil ID dari banyak baris data_user}

{I.S. : program berjalan dan pengakses adalah admin}

{F.S. : ID, nama, username, password, role 'user', dan saldo '0' ditambahkan ke data_user}

function login (user_name : array [1..Nmax] of data_user) -> (integer, array [1..Nmax] of string)

{fungsi mengembalikan nilai user_id yang sedang menjalankan program}

{I.S. : pengguna belum login, program berjalan}

{F.S. : fungsi mengembalikan nilai user_id}

procedure tambah_game (input/output data_tambah_game : array [1..Nmax] of data_game)

{memasukkan nama game, kategori, tahun rilis, harga, dan stok awal game ke list untuk dicek apakah semua informasi terisi}

{I.S. : program berjalan dan diakses oleh admin}

{F.S. : nama game, kategori, tahun rilis, harga, stok awal game ditambahkan ke data_game}

function ubah_game (data : array [1..Nmax] of array [1..Idmax] of string) -> array [1..Nmax] of array [1..Idmax] of string

{Spesifikasi program : mengubah atribut dari data game kecuali id game dan stok}

{I.S. : program berjalan (sudah load data) dan pengakses adalah admin}

{F.S. : data disimpan lagi ke variabel}

function ubah_stok (data : array [1..Nmax] of array [1..Idmax] of string) -> array [1..Nmax] of array [1..Idmax] of string

{Spesifikasi program : mengubah stok game yang diinginkan}

{I.S. : program berjalan (sudah load data) dan pengakses adalah admin}

{F.S. : data disimpan lagi ke variabel}

procedure search_my_game (input data : array [1..Nmax] of array [1..Idmax] of string)

{Spesifikasi program : mencari game yang dimiliki oleh user dari input yang dimasukkan dan men-
outputkan ke layar}

{I.S. : program berjalan (sudah load data) dan sudah login}

{F.S. : Output hasil pencarian}

procedure search_game_at_store (input data : array [1..Nmax] of array [1..Idmax] of string)

{ Spesifikasi program : mencari game di toko sesuai dengan input parameter dan men-outputkan ke layar}

{I.S. : program berjalan (sudah load data) dan sudah login}

{F.S. : Output hasil remove_not_match terakhir.}

procedure topup (input/output user_name : array [1..Nmax] of data_user)

{prosedur untuk menambah atau mengurangi saldo pada data_user}

{I.S. : user_name terdefinisi pada data_user, program diakses oleh admin }

{F.S. : saldo ditambah atau dikurangi sesuai masukan dari array [1..Nmax] of <saldo> sesuai username
yang dimasukkan.}

procedure help (input role : string)

{prosedur untuk menampilkan fitur help sesuai dengan role pengakses}

{I.S. : program berjalan dengan pengakses adalah admin atau user}

{F.S. : cetak command help sesuai role pengakses}

function load (filename : string) - > array [1..Nmax] of array [1..Idmax] of string)

{ Spesifikasi program : membaca filename.csv dan mengubahnya menjadi array of array of string }

{I.S. : Jalankan program dan masukan argumen folder}

{F.S. : Simpan hasil baca ke dalam variabel.}

procedure save_data (input data_game : array [1..Nmax] of array [1..Idmax] of string, input data_user :
array [1..Nmax] of array [1..Idmax] of string, input data_rwayat : array [1..Nmax] of array [1..Idmax] of
string, input data_kepemilikan : array [1..Nmax] of array [1..Idmax] of string)

{ Spesifikasi program : menulis data_game, data_user, data_rwayat, data_kepemilikan ke filename.csv }

{I.S. : program berjalan (sudah load data) dan sudah login}

{F.S. : Terdapat file csv di folder.}

procedure exit_program (input data_seluruh : data)

{prosedur ini mengeluarkan pengguna dari program. sebelumnya, pengguna diberikan opsi untuk menyimpan perubahan}

{I.S. : program berjalan, data_seluruh terdefinisi}

{F.S. : jika pengguna menginput 'y' atau 'Y', jalankan prosedur save. jika pengguna menginput 'n' atau 'N', program berhenti}

procedure kerang_ajaib ()

{Spesifikasi program : menghasilkan string yang indeks nya ditentukan oleh lcg()}

{I.S. : program berjalan (sudah load data)}

{F.S. : Output wisdom[n] .}

function len (data : array) -> integer

{Spesifikasi program : menghitung panjang suatu array}

{I.S. : array terdefinisi}

{F.S. : menghasilkan panjang array (integer).}

function inside (element: char, list : array) -> boolean

{ Spesifikasi program : mengecek apakah sebuah data ada di dalam array }

{ I.S. : array dan element terdefinisi }

{ F.S. : menghasilkan true jika ada, false jika tidak ada }

function pop (data : array , index : integer) -> array

{ Spesifikasi program : menghapus sebuah data[index] dari array }

{ I.S. : array terdefinisi dan index terdefinisi }

{ F.S. : menghasilkan array yang telah dihapus data[index] }

```
function no_duplicate (list1 : array [1..Nmax] of array [1..Idmax] of string , list2 : array [1..Nmax] of array [1..Idmax] of string, index : integer) -> array [1..Nmax] of array [1..Idmax] of string
```

```
{ Spesifikasi program : menghasilkan array of array of string yang tidak sama }
```

```
{ I.S. : array terdefinisi dan index terdefinisi }
```

```
{ F.S. : menghasilkan array of array of string yang tidak sama }
```

```
function search_cat ( category : string , name : string , data : array [1..Nmax] of array [1..Idmax] of string) -> array [1..Nmax] of array [1..Idmax] of string
```

```
{ Spesifikasi program : mencari data yang memiliki nilai yang sama pada kategori tertentu }
```

```
{ I.S. : category, name, data terdefinisi }
```

```
{ F.S. : menghasilkan array yang berisi data yang memiliki nilai yang sama pada kategori tertentu }
```

```
function search_in ( index : integer , name : string , data : array [1..Nmax] of array [1..Idmax] of string) -> array [1..Nmax] of array [1..Idmax] of string
```

```
{ Spesifikasi program : mencari data yang memiliki nilai yang sama pada index tertentu }
```

```
{ I.S. : index, name, data terdefinisi }
```

```
{ F.S. : menghasilkan array yang berisi data yang memiliki nilai yang sama pada kategori tertentu }
```

```
function remove_not_match (data: array [1..Nmax] of array [1..Idmax] of string , element : string , index : integer ) -> array [1..Nmax] of array [1..Idmax] of string
```

```
{ Spesifikasi program : menghapus data yang tidak sesuai dengan kriteria (element) }
```

```
{ I.S. : array, element, index terdefinisi }
```

```
{ F.S. : menghasilkan array yang berisi data yang sesuai dengan kriteria }
```

```
function read_csv (filename : array) -> array [1..Nmax] of array [1..Idmax] of string
```

```
{ Spesifikasi program : membaca file csv dan mengubahnya menjadi array of array of string }
```

```
{ I.S. : filename terdefinisi }
```

```
{ F.S. : menghasilkan array of array of string yang berisi data dari file csv }
```

```
procedure write_csv (input filename : string, input data : array [1..Nmax] of array [1..Idmax] of string)
```

```
{ Spesifikasi program : menulis data ke file csv }
```

{ I.S. : filename dan data terdefinisi }

{ F.S. : filename yang berisi data terdapat dalam disk }

SCREENSHOT HASIL PENGUJIAN PROGRAM

F15 – Load

```
PS C:\Personal\Homework\Tugas\TPB\Semester 2\Daspro\Tubes\TBIF1210-05-02> python .\main_prog.py
usage: main_prog.py [-h] folder
main_prog.py: error: the following arguments are required: folder
Tidak ada nama folder yang diberikan!
PS C:\Personal\Homework\Tugas\TPB\Semester 2\Daspro\Tubes\TBIF1210-05-02>
```

Gambar 14 Eksekusi F01 Argumen Folder Tidak Diberikan

```
PS C:\Personal\Homework\Tugas\TPB\Semester 2\Daspro\Tubes\TBIF1210-05-02> python .\main_prog.py asdgsd
Folder asdgsd tidak ditemukan.
```

Gambar 15 Eksekusi F01 Argumen Folder Tidak Ditemukan

```
PS C:\Personal\Homework\Tugas\TPB\Semester 2\Daspro\Tubes\TBIF1210-05-02> python .\main_prog.py database
loading ...
Selamat datang di antarmuka "Binomo"!
>>>
```

Gambar 16 Eksekusi F01 Tampilan Antarmuka

F02 – Register

```
>>> register
Masukkan nama: qiqi
Masukkan username: qiqi
Masukkan password: 123
User terdaftar!
>>>
```

Gambar 17 Eksekusi F02 User Terdaftar

```
>>> register
Masukkan nama: qiqi
Masukkan username: Qiqi
Masukkan password: 123
Maaf! Username sudah dipakai.
```

Gambar 18 Eksekusi F02 Username Telah Terpakai

F03 – Login

```
>>> login
Masukkan username: qiqi
Masukkan password: 123
Password atau username salah atau tidak ditemukan.
>>> █
```

Gambar 19 Eksekusi F03 Password atau Username Salah

```
>>> login
Masukkan username: Qiqi
Masukkan password: 123
Halo Rifqi! Selamat datang di "Binomo"
>>> █
```

Gambar 20 Eksekusi F03 Login Berhasil

F04 – Menambah Game ke Toko Game

```
>>> tambah_game
Masukkan nama game:
Masukkan kategori: adventure
Masukkan tahun rilis: 2012
Masukkan harga: 50000
Masukkan stok awal: 100
Mohon masukkan semua informasi mengenai game agar dapat disimpan di BNMO
Masukkan nama game: █
```

Gambar 21 Eksekusi F04 Mengembalikan Pesan Error

```
Masukkan nama game: FF
Masukkan kategori: rpg
Masukkan tahun rilis: 2012
Masukkan harga: 100000
Masukkan stok awal: 100
Selamat! Berhasil menambahkan gameFF
>>> █
```

Gambar 22 Eksekusi F04 Mengembalikan Pesan Sukses

F05 – Mengubah Game pada Toko Game


```
>>> ubah_game
Masukkan ID game:
Masukkan nama game: FF
Masukkan kategori game: adv
Masukkan tahun rilis game: 2010
Masukkan harga game: 99000
ID tidak boleh kosong.
>>> █
```

Gambar 23 Eksekusi F05 ID Tidak Boleh Kosong

```
Masukkan ID game: G099
Masukkan nama game: FF
Masukkan kategori game: adv
Masukkan tahun rilis game: 2010
Masukkan harga game: 99000
Tidak ada id tersebut.
>>> █
```

Gambar 2424 Eksekusi F05 ID Tidak Ada

```
>>> ubah_game
Masukkan ID game: G011
Masukkan nama game: FF
Masukkan kategori game: adv
Masukkan tahun rilis game: 2010
Masukkan harga game: 99000
Game berhasil diubah
>>> █
```

Gambar 2525 Eksekusi F05 Mengembalikan Pesan Sukses

F06 – Mengubah Stok Game di Toko

```
>>> ubah_stok
Masukkan ID game: G099
Masukkan jumlah: 100
Tidak ada id tersebut.
>>> █
```

Gambar 26 Eksekusi F06 ID Tidak Ada

```
>>> ubah_stok
Masukkan ID game: G001
Masukkan jumlah: -100
Stok game Tetris gagal dikurang karena stok kurang. Stok sekarang: 4 ( < 100 )
>>> █
```

Gambar 27 Eksekusi F06 Stok Game Kurang

```
>>> ubah_stok
Masukkan ID game: G001
Masukkan jumlah: 10
Stok game Tetris berhasil ditambah. Stok sekarang: 14
>>> █
```

Gambar 28 Eksekusi F06 Stok Game Berhasil Ditambah

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

```
>>> list_game_toko
Skema sorting: tahun+
1. G014 | Frogger | Puzzle | 1981 | 21000 | 34
2. G036 | Donkey Kong | Race | 1981 | 8000 | 67
3. G037 | Galaga | Adventure | 1981 | 36000 | 98
4. G038 | Pac Man | Adventure | 1982 | 6500 | 48
5. G008 | Duck Hunt | Adventure | 1984 | 28000 | 15
6. G001 | Tetris | Puzzle | 1984 | 170000 | 14
7. G002 | Mario Kart Wii | Race | 1985 | 48000 | 5
8. G039 | Contra | Battle | 1987 | 8200 | 86
9. G040 | Zelda | Adventure | 1987 | 4700 | 35
```

Gambar 29 Eksekusi F07 Sorting Tahun Ascending

```
>>> list_game_toko
Skema sorting: harga-
1. G003 | Minecraft | Adventure | 2011 | 176000 | 6
2. G001 | Tetris | Puzzle | 1984 | 170000 | 14
3. G011 | FF | adv | 2010 | 99000 | 91
4. G032 | Death Rally | Race | 2014 | 76400 | 21
5. G034 | Farm Clan | Adventure | 2009 | 53000 | 26
6. G030 | Looney Tunes Dash | Race | 2000 | 50101 | 51
7. G021 | Pou | Adventure | 2005 | 50000 | 43
8. G002 | Mario Kart Wii | Race | 1985 | 48000 | 5
9. G006 | Pokemon | Adventure | 1996 | 47000 | 11
```

Gambar 30 Eksekusi F07 Sorting Harga Descending

```
>>> list_game_toko
Skema sorting:
1. G001 | Tetris | Puzzle | 1984 | 170000 | 4
2. G002 | Mario Kart Wii | Race | 1985 | 48000 | 5
3. G003 | Minecraft | Adventure | 2011 | 176000 | 6
4. G005 | Pubg | Battle | 2017 | 5000 | 8
5. G006 | Pokemon | Adventure | 1996 | 47000 | 11
6. G007 | Skyrim | Race | 2011 | 30000 | 13
7. G008 | Duck Hunt | Adventure | 1984 | 28000 | 15
8. G009 | Terraria | Puzzle | 2011 | 27000 | 17
9. G010 | Fifa 18 | Sports | 2017 | 24000 | 18
```

Gambar 31 Eksekusi F07 Sorting ID

F08 – Membeli Game

```
>>> buy_game
Masukkan ID Game: G001
Anda sudah memiliki Game tersebut!
>>> █
```

Gambar 32 Eksekusi F08 Game Telah Dimiliki

```
>>> buy_game
Masukkan ID Game: G002
Stok Game tersebut sedang habis!
>>> █
```

Gambar 33 Eksekusi F08 Stok Game Sedang Habis

```
>>> buy_game
Masukkan ID Game: G002
Game Mario Kart Wii berhasil dibeli!
>>>
```

Gambar 3434 Eksekusi F08 Game Berhasil Dibeli

F09 – Melihat Game yang dimiliki

```
>>> list_game
1. G001 | Tetris | Puzzle | 1984 | 170000
2. G011 | Sonic | Race | 1991 | 2400
3. G021 | Pou | Adventure | 2005 | 50000
4. G031 | Cut The Rope | Puzzle | 2003 | 21800
5. G041 | Mega Man | Battle | 1990 | 3900
>>>
```

Gambar 35 Eksekusi F09 Mengembalikan List Game User

F10 – Mencari Game yang dimiliki dari ID dan tahun rilis

```
>>> search_my_game
Masukkan ID Game: G001
Masukkan Tahun Rilis Game: 1983
Game tidak ditemukan
```

Gambar 36 Eksekusi F10 Game Tidak Ditemukan

```
>>> search_my_game
Masukkan ID Game: G001
Masukkan Tahun Rilis Game: 1984
1 . G001 | Tetris | Puzzle | 1984 | 170000
```

Gambar 37 Eksekusi F10 Game Berhasil Ditemukan

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

```
>>> search_game_at_store
Masukkan ID Game:
Masukkan Nama Game:
Masukkan Harga Game:
Masukkan Kategori Game:
Masukkan Tahun Rilis Game: 1984
Daftar game pada inventory yang memenuhi kriteria:
1 . G001 | Tetris | Puzzle | 1984 | 170000
2 . G008 | Duck Hunt | Adventure | 1984 | 28000
>>> █
```

Gambar 38 Eksekusi F11 Daftar Game Sesuai Kriteria

```
>>> search_game_at_store
Masukkan ID Game: G008
Masukkan Nama Game: Tetris
Masukkan Harga Game:
Masukkan Kategori Game:
Masukkan Tahun Rilis Game:
Daftar game pada inventory yang memenuhi kriteria:
Game tidak ditemukan.
>>> █
```

Gambar 39 Eksekusi F11 Game Tidak Ditemukan

F12 – Top Up Saldo

```
>>> topup
Masukkan username: qiqi
Masukkan saldo: 100000
Username qiqi tidak ditemukan.
>>> █
```

Gambar 40 Eksekusi F12 Username Tidak Ditemukan

```
>>> topup
Masukkan username: Qiqi
Masukkan saldo: -1000000
Masukan tidak valid. Saldo Anda tidak mencukupi.
>>> █
```

Gambar 4141 Eksekusi F12 Masukkan Saldo Tidak Valid

```
>>> topup
Masukkan username: Qiqi
Masukkan saldo: 99999
Top up berhasil. Saldo Rifqi bertambah menjadi 99999
>>> █
```

Gambar 42 Eksekusi F12 Top Up berhasil

F13 – Melihat Riwayat Pembelian

```
>>> riwayat
1. G001 | Tetris | 170000 | 1990
2. G011 | Sonic | 2400 | 2000
3. G021 | Pou | 50000 | 2008
4. G031 | Cut The Rope | 21800 | 2007
5. G041 | Mega Man | 3900 | 1194
>>> █
```

Gambar 43 Riwayat Pembelian User

F14 - Help

```
>>> help
===== HELP =====
1. register - untuk melakukan registrasi user baru.
2. login - untuk melakukan login ke sistem.
3. tambah_game - untuk menambahkan game ke toko
4. ubah_game - untuk mengubah game pada toko.
5. ubah_stok - untuk mengubah stok game di toko
6. list_game_toko - untuk melakukan sorting daftar game sesuai kategori tertentu.
7. search_game_at_store - untuk mencari game di toko.
8. topup - untuk menambahkan saldo user.
9. save - untuk melakukan penyimpanan pada file yang diubah.
10. kerang_ajaib - untuk bertanya kepada kerang ajaib.
11. tictactoe - untuk bermain tic tac toe.
.....
>>> █
```

Gambar 44 Eksekusi F14 untuk role admin

```
>>> help
===== HELP =====
1. login - untuk melakukan login ke sistem.
2. list_game_toko - untuk melakukan sorting pada game di toko sesuai kategori tertentu.
4. buy_game - untuk membeli game di toko.
5. list_game - untuk melihat game yang dimiliki user.
6. search_my_game - untuk mencari game yang dimiliki berdasarkan id dan tahun rilis.
7. search_game_at_store - untuk mencari game di toko berdasarkan kategori tertentu.
8. riwayat - untuk melihat riwayat pembelian.
9. save - untuk menyimpan perubahan pada file.
10. kerang_ajaib - untuk bertanya kepada kerang ajaib.
11. tictactoe - untuk bermain tic tac toe.
.....
>>> 
```

Gambar 4545 Eksekusi F14 untuk role user

F16 - Save

```
>>> save
Masukkan nama folder penyimpanan: 2022
Saving ...
Data berhasil disimpan di folder 2022
>>> 
```

Gambar 46 Eksekusi F16 Keberhasilan Proses Penyimpanan

F17 – Exit

```
>>> exit
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n)
>>> a
Input tidak valid.
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n)
>>> 
```

Gambar 47 Eksekusi F17 Input Tidak Valid

```
>>> exit
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n)
>>> n
Sampai jumpa!
```

Gambar 4848 Eksekusi F17 tanpa Penyimpanan

```
>>> exit
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n)
>>> y
Masukkan nama folder penyimpanan: save
Saving ...
Data berhasil disimpan di folder save
Sampai jumpa!
```

Gambar 49 Eksekusi F12 dengan Penyimpanan

B02 – Magic Conch Shell

```
>>> kerangajaib
Apa pertanyaanmu? apa?

Tentunya

>>> kerangajaib
Apa pertanyaanmu? apa?

Tidak ada

>>> █
```

Gambar 5050 Eksekusi B02 Program Kerang Ajaib

B03 – Game Tic-Tac-Toe

```
Status Papan
O - -
- X -
- - O

Giliran pemain X
Baris: 4
Kolom: 4

Kotak tidak valid.
```

Gambar 51 Eksekusi B03 Masukkan Tidak Valid


```

Status Papan
X O X
- O O
O X -

Giliran pemain X
Baris: 2
Kolom: 1

Status Papan
X O X
X O O
O X -

Giliran pemain O
Baris: 3
Kolom: 3

Status Papan
X O X
X O O
O X O
Hasil Permainan Imbang
>>>

```

Gambar 52 Eksekusi B03 Hasil Permainan Imbang

```

Status Papan
O - X
- O -
- - -

Giliran pemain X
Baris: 2
Kolom: 1

Status Papan
O - X
X O -
- - -

Giliran pemain O
Baris: 3
Kolom: 3

Status Papan
O - X
X O -
- - O

Pemain "O" menang
>>>

```

Gambar 53 Eksekusi B03 Hasil Akhir Permainan

LAMPIRAN

Form Asistensi

Form MoM Asistensi 1, 14 April 2022

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

Nomor Asistensi : 1
No. Kelompok/Kelas : 02 / K05
Tanggal asistensi : 14 April 2022

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	16521176 / Yanuar Sano Nur Rasyid
	2	16521239 / Firda Sabrina Lutfiyah
	3	16521329 / Mohammad Rifqi Farhansyah
	4	
	5	
	6	
Asisten pembimbing	NIM / Nama	
	Reyhan Emyr Arrosyid \ 13519167	

Catatan Asistensi:

Pastiin programnya memenuhi spesifikasi dan batasan yang ada karena tujuan tugas ini buat belajar.

Rangkuman Diskusi
<ul style="list-style-type: none">• Output dari program tidak harus sama dengan spesifikasi yang penting memiliki fungsi yang setara• File .csv yang akan di-load lokasinya bebas, boleh berada di folder yang sama dengan file .py atau boleh memakai path tertentu• File .csv dibaca dengan <code>with open</code> dan menggunakan fungsi <code>.split()</code> yang ditulis sendiri.• Desain command primitif pada deliverables berisi apa input yang dimasukkan dan apa output yang diberikan.• Efisiensi program tidak memengaruhi nilai selama program yang dibuat 'normal-normal saja.'
Tindak Lanjut
Mengumpulkan tugas tepat waktu dan kerjakan sesuai spesifikasi yang diberikan, contohnya membuat fitur <code>split()</code> dan <code>append()</code> sendiri. Jika ada pertanyaan, silakan hubungi kakak asistennya

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

Nomor Asistensi : 2
 No. Kelompok/Kelas : 02 / K05
 Tanggal asistensi : 22 April 2022

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	16521176 / Yanuar Sano Nur Rasyid
	2	16521239 / Firda Sabrina Lutfiyah
	3	16521329 / Mohammad Rifqi Farhansyah
	4	
	5	
	6	
Asisten pembimbing	NIM / Nama	
	Reyhan Emyr Arroseyid \ 13519167	

Catatan Asistensi:

Program jangan kepisah-pisah (jadiin 1 program) dan jangan simpan variabel di temporary csv. Buat satu kesatuan, yang nerima input perintah terus sampai ke exit. Simpan temporarynya di array pada satu variabel. Intinya yang modular fungsinya aja, buat program jalanin 1 aja. Operasi 'if something in array' tidak boleh, harus bikin fungsi search sendiri. Jangan pakai dictionary. Untuk load filenya relative ke satu folder yang setara sama program aja. Untuk F01, yang diminta bukan membuat diagram atau slie, tetapi modular dalam pembuatan programnya langsung.

Rangkuman Diskusi

Untuk menambahkan string, boleh pakai '+', tetapi untuk fungsi split, harus bikin sendiri.
 Fungsi dari import os boleh dipakai semua.
 Data ID game unik untuk tiap game sehingga jika di-search menggunakan ID game, hanya akan keluar 1 game. Intinya, ID game sifatnya identik, satu ID untuk 1 game.
 Search pakai data yang sudah dimasuki di array/data yang di-update. Untuk notasi algoritmik, boleh dijelasin dalam komentar.
 Untuk fungsi .strip(), harus diimplementasikan sendiri.
 Save temporary juga berlaku untuk register, artinya data register baru tersimpan setelah dilakukan perintah save.
 Format laporan tidak ada format khusus, yang penting sesuai spesifikasi dan rapi.
 Format video boleh langsung dimulai dengan program tanpa harus membuat power point.

Tindak Lanjut

Menyatukan subprogram, mengecek kembali fungsi yang dipakai, membuat laporan dan membuat video demo.