

PROGRAM TOKO GAME

“BINOMO”

Dibuat Sebagai Tugas Besar

IF1210

Dasar Pemrograman



K-07

Kelompok 5

Intan Putri Maharani P	16521222
Reyhan Ghifari Tanjung	16521366
Raden Dizi Assyafadi P	16521456
Mahardika Wisnu Aji	16521474

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2022

HALAMAN 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 IF210 Dasar Pemrograman Semester 2 2021/2022.”

Intan Putri Maharani P	16521222
Reyhan Ghifari Tanjung	16521366
Raden Dizi Assyafadi P	16521456
Mahardika Wisnu Aji	16521474

Daftar isi

Daftar Tabel	3
Deskripsi Persoalan	5
Daftar Pembagian Tugas	8
Checklist Hasil Rancangan	10
Desain Command Untuk Setiap Primitif	11
Desain Kamus Data	15
Desain Dekomposisi Algoritmik dan Fungsional Program	19
Spesifikasi untuk tiap Modul/Prosedur/Fungsi	22
Hasil Output dari Program	47
Lampiran : MoM Asistensi	58
Daftar Pustaka	60

Daftar Tabel

	Halaman
Tabel 1: Pembagian tugas program utama	8
Tabel 2: Pembagian tugas program bonus	9
Tabel 2: Pembagian tugas laporan	9
Tabel 3: Daftar Checklist hasil rancangan Program Utama	9
Tabel 4: Daftar Checklist hasil rancangan Program Bonus	10

Daftar Gambar

	Halaman
Gambar 1: Hasil output F02 - Register	31
Gambar 2: Hasil output F03 - Load	31
Gambar 3: Hasil output F04 - Tambah Game	32
Gambar 4: Hasil output F05 - Ubah Game	32
Gambar 5: Hasil output F06 - Ubah Stok	33
Gambar 6: Hasil output F07 - Listing Game Toko	33
Gambar 7: Hasil output F08 - Buy Games	34
Gambar 8: Hasil output F09 - List game User	34
Gambar 9: Hasil output F10 - Search Game User	35
Gambar 10: Hasil output F11 - Search Game at Store	35
Gambar 11: Hasil output F12 - Top Up	36
Gambar 12: Hasil output F13 - Riwayat	36
Gambar 13: Hasil output F14 - Help	37
Gambar 14: Hasil output F15 - Load	37
Gambar 15: Hasil output F16 - Save	37
Gambar 16: Hasil output F17 - Exit	38
Gambar 17: Hasil output B02 - Magic Conch Shell	38
Gambar 18: Hasil output B03 - Tic Tac Toe	38
Gambar 19: MoM Asistensi 1	39
Gambar 20: MoM Asistensi 2	40

Deskripsi Persoalan

Dalam tugas besar ini, mahasiswa Institut Teknologi Bandung program TPB Fakultas Sekolah Teknik Elektro dan Informatika kelas Dasar Pemrograman diminta membuat suatu program toko game. Pengakses dari program ini dibedakan menjadi dua jenis, yaitu admin dan user. Program ini memiliki 16 fungsionalitas utama, yaitu:

F02 - Register

Fungsi Register adalah program yang dapat diakses oleh admin dan akan melakukan registrasi akun. Fungsi register menerima input nama, *username*, dan *password*. Password yang diinputkan harus menyesuaikan dengan ketentuan yang ada pada program. Akun yang berhasil diregistrasi akan mendapatkan role user.

F03 - Login

Fungsi Login menerima input username dan password, bila password dan username yang diinputkan cocok maka pengguna berhasil login. Fungsi ini dapat diakses oleh user dan admin.

F04 - Menambah Game ke Toko Game

Fungsi Menambah Game ke Toko Game hanya dapat diakses oleh admin. Fungsi ini menerima input nama game, kategori, tahun rilis, harga, dan stok. Inputan dari admin lalu akan divalidasi oleh program sesuai ketentuan pada program dan jika berhasil maka akan dimasukkan kedalam toko game.

F05 - Mengubah Game pada Toko Game

Fungsi ini hanya dapat diakses oleh admin yang berfungsi mengubah game yang telah ada pada toko game. Fungsi ini menerima input ID game, nama game, tahun rilis, dan harga. Inputan dari admin lalu akan divalidasi oleh program dan jika berhasil atribut dari game yang diinputkan akan diganti sesuai inputan admin.

F06 - Mengubah Stok Game di Toko

Fungsi ini hanya dapat diakses oleh admin dan berfungsi untuk mengubah stok game pada toko. Fungsi ini menerima input ID game dan jumlah stok yang akan diubah. Program akan

melakukan validasi dari input dan ketika valid maka stok pada game yang diinputkan akan berubah sesuai dengan inputan dari admin.

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

Fungsi ini dapat diakses oleh user dan admin. Fungsi ini berfungsi untuk menampilkan daftar game yang terdapat pada toko game berdasarkan tahun atau harga dan *ascending* atau *descending* sesuai dengan inputan pengguna.

F08 - Membeli Game

Fungsi Membeli Game digunakan untuk membeli game. Fungsi ini hanya dapat diakses oleh user dan akan menerima input ID game, lalu program akan melakukan validasi. Jika game berhasil dibeli, riwayat pembelian dan kepemilikan akan disimpan.

F09 - Melihat Game yang Dimiliki

Fungsi ini dapat diakses oleh user dan akan memberikan output game yang dimiliki. Jika user tidak mempunyai game maka akan dikeluarkan pesan kepada user untuk membeli game.

F10 - Mencari Game yang Dimiliki dari ID dan Tahun Rilis

Fungsi ini dapat diakses oleh user yang digunakan untuk mencari game yang dimiliki dari ID dan tahun rilis. Fungsi ini mempunyai parameter ID game dan tahun rilis yang tidak wajib diisi. Ketika parameter kosong maka program akan memberikan output game yang dimiliki dan jika parameter fungsi diisi maka program akan memberikan output sesuai dengan inputan user.

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

Fungsi ini dapat diakses oleh user dan admin. Fungsi ini mempunyai parameter ID game, nama game, harga, kategori, dan tahun rilis. Parameter pada fungsi ini tidak wajib diisi. Program akan melakukan validasi dan mencari game yang sesuai dengan inputan pengguna dan akan memberikan output sesuai dengan inputan user.

F12 - Top Up Saldo

Fungsi Top Up Saldo hanya dapat digunakan oleh admin dan akan menerima input *username* dan *password*. Program akan melakukan validasi dan mengubah saldo dari user.

F13 - Melihat Riwayat Pembelian

Fungsi ini digunakan untuk melihat riwayat pembelian yang hanya dapat diakses oleh user. Fungsi ini tidak menerima inputan apapun dan akan memberikan input riwayat game yang dibeli oleh user, dan jika user tidak mempunyai game maka program akan memberikan pesan kepada user untuk membeli game.

F14 - Help

Fungsi Help dapat diakses oleh user dan admin yang berfungsi memberikan output fungsi - fungsi yang dapat diakses oleh pengguna sesuai dengan role. Fungsi ini dapat diakses ketika belum melakukan login.

F15 - Load

Prosedur Load digunakan untuk melakukan loading data ke dalam sistem. Prosedur ini otomatis dijalankan ketika sistem pertama kali menerima input nama folder yang berisi file penyimpanan. Semua file penyimpanan dijamin ada, tetapi untuk folder harus dilakukan validasi terlebih dahulu.

F16 - Save

Prosedur Save digunakan untuk melakukan penyimpanan data ke dalam file setelah dilakukan perubahan. Prosedur ini dapat diakses oleh admin dan user. Untuk nama folder, bila belum ada, maka akan dibuat folder baru dengan file csv di dalamnya. Jika sudah ada, tidak perlu menghapus folder lama. Untuk nama file, jika belum ada, maka akan dibuat file baru. Jika sudah ada, ganti file lama dengan file baru.

F17 - Exit

Merupakan fungsi yang digunakan untuk keluar dari program/aplikasi.

Selain 16 fungsionalitas utama di atas, ada juga tiga program bonus yang dapat dibuat, yaitu:

B01 - Cipher

Prosedur Cipher merupakan prosedur yang bertujuan untuk menyamarkan password dari pengguna pada file csv.

B02 - Magic Conch Shell

Merupakan program yang dapat menerima input pertanyaan dan mengeluarkan output jawaban secara random/acak.

B03 - Game Tic Tac Toe

Program yang men-digitalisasi permainan papan “Tic Tac Toe” yang pada prakteknya terdiri dari simbol X dan O.

Seluruh program di atas dibuat dengan mengimplementasikan Modular Programming, yaitu teknik pemrograman untuk memecah/dekomposisi suatu program yang besar menjadi beberapa bagian program yang lebih kecil.

Daftar Pembagian Tugas

Tabel 1. Pembagian Tugas Program Utama

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F02 - Register	register	16521456	16521456	16521222
F03 - Login	login	16521456	16521456	16521222
F04- Menambah Game ke Toko Game	tambah_game	16521222	16521222	16521222
F05 - Mengubah Game di Toko	ubah_game	16521474	16521474	16521222
F06 - Mengubah Stok Game di Toko	ubah_stok	16521366	16521366	16521456
F07 - Listing Game di Toko	list_game_toko	16521474	16521474	16521456
F08 - Membeli Game	buy_game	16521366	16521366	16521456
F09 - Melihat Game yang dimiliki	list_game	16521366	16521366	16521456
F10 - Mencari Game yang dimiliki dari ID dan Tahun Rilis	search_my_game	16521474	16521474	16521366
F11 - Mencari Game di Toko dari ID, Nama Game, Harga, Kategori, dan Tahun Rilis	search_game_at_store	16521456	16521456	16521366

F12 - Top Up Saldo	topup	16521366	16521366	16521366
F13 - Melihat Riwayat Pembelian	history	16521474	16521474	16521366
F14 - Help	Help	16521456	16521456	16521474
F15 - Load	load	16521222	16521222	16521474
F16 - Save	save	16521222	16521222	16521474
F17 - Exit	exit_program	16521222	16521222	16521474

Tabel 2. Pembagian Tugas Program Bonus:

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
B01 - Cipher	encrypt decrypt outputing	16521366	16521366	16521456
B02 - Magic Conch Shell	kerangajaib	16521456	16521456	16521222
B03 - Game Tic Tac Toe	tictactoe	16521456	16521456	16521366

Tabel 3. Pembagian Tugas Laporan:

Bagian Laporan	NIM
Halaman Cover	16521366, 16521222
Halaman Pernyataan Kelompok	16521222
Daftar Isi	16521366
Daftar Tabel	16521222
Daftar Gambar	16521222
Deskripsi Persoalan	16521366, 16521222
Daftar Pembagian Tugas	16521366, 16521474
Checklist Hasil Rancangan	16521474
Desain Command Untuk Setiap Primitif	16521474
Desain Kamus Data	16521456
Desain Dekomposisi Algoritmik dan Fungsional	16521456

Program	
Spesifikasi Untuk Tiap Modul/Prosedur/Fungsi	16521456
Hasil Output dari Program	16521456
Lampiran : MoM Asistensi	16521474
Editing video	16521474

Checklist Hasil Rancangan

Tabel 4. daftar Checklist hasil rancangan Program Utama

Fitur	Desain	Implementasi	Testing
F02 - Register	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F03 - Login	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F04- Menambah Game ke Toko Game	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F05 - Mengubah Game di Toko	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F06 - Mengubah Stok Game di Toko	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F07 - Listing Game di Toko	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F08 - Membeli Game	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F09 - Melihat Game yang dimiliki	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F10 - Mencari Game yang dimiliki dari ID dan Tahun Rilis	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F11 - Mencari Game di Toko dari ID, Nama Game, Harga, Kategori, dan Tahun Rilis	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F12 - Top Up Saldo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F13 - Melihat Riwayat	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Pembelian			
F14 - Help	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F15 - Load	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F16 - Save	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F17 - Exit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Program Utama 16/16

Tabel 5. daftar Checklist hasil rancangan Program Bonus

Fitur	Desain	Implementasi	Testing
B01 - Cipher	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
B02 - Magic Conch Shell	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
B03 - Game Tic Tac Toe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Program Bonus 3/3

Desain Command Untuk Setiap Primitif

1. register (database)

Input:

username → Pemakai diminta meng-input username yang diinginkan

nama → Pemakai diminta meng-input nama yang diinginkan

password → Pemakai diminta menginput password yang diinginkan tetapi sesuai dengan ketentuan

Output:

Data akan disimpan di *database*, akan muncul juga pesan registrasi berhasil apabila registrasi berhasil dan pesan untuk memberitahu user untuk mengulangi apabila variabel yang diminta tidak valid atau tidak sesuai ketentuan.

2. login (database)

Input:

username → Pemakai diminta meng-input username yang telah di register

password → Pemakai diminta meng-input password yang telah terdaftar bersama username sesuai registrasi

Output:

Ditampilkan pesan *login* telah berhasil dan juga selamat datang kepada nama yang teregistrasi bersama username.

3. tambah_game (database)

Input:

nama game → Pemakai diminta meng-input nama game yang ingin ditambahkan ke toko

kategori → Pemakai diminta meng-input kategori game yang ingin ditambahkan ke toko

tahun rilis → Pemakai diminta meng-input tahun rilis game yang ingin ditambahkan ke toko

harga → Pemakai diminta meng-input harga game yang ingin ditambahkan ke toko

stok → Pemakai diminta meng-input stok game yang ingin ditambahkan ke toko

Output:

Data game baru akan dimasukkan kedalam database dan ditampilkan pesan game berhasil ditambahkan

4. ubah_game (database)

Input:

nama game → Pemakai diminta meng-input nama game yang ingin diubah

kategori → Pemakai diminta meng-input kategori game yang ingin diubah

tahun rilis → Pemakai diminta meng-input tahun rilis game yang ingin diubah

harga → Pemakai diminta meng-input harga game yang ingin ditambahkan diubah

stok → Pemakai diminta meng-input stok game yang ingin diubah

Output:

Data game yang ingin diubah akan mengubah data yang ada di dalam database dan diberikan pesan game berhasil diubah. Tidak bisa mengubah stok game

5. ubah_stok (database)

Input:

id → Pemakai diminta meng-input id game yang ingin penambahan atau perubahan stok

stok → Pemakai diminta meng-input berapa stok yang ingin ditambahkan atau dikurangkan (+ 'stok' , - 'stok')

Output:

Data stok game akan mengubah data yang ada didalam database dan ditampilkan pesan stok berhasil diubah

6. list_game_toko (database)

Output:

Ditampilkan data tabel dari semua nama beserta informasi id, harga, kategori, tahun rilis, dan stok dari game yang ada di toko (database) serta diberikan juga fitur sort untuk mengurutkan berdasarkan harga dan tahun yang ada.

7. buy_game (database)

Input:

id → Pemakai diminta meng-input id game yang ingin dibeli

Output:

Diberikan nama dari game yang dibeli serta ditampilkan pesan bahwa game tersebut sudah berhasil terbeli. Kemudian data dari game tersebut dimasukkan ke dalam database kepemilikan serta dicatat pembelian ke dalam database riwayat.

8. list_game (database)

Output:

Diberikan data berupa tabel game apa saja yang sudah dibeli beserta informasi id, harga, kategori, dan tahun rilis game-game tersebut.

9. search_my_game (database)

Input:

id → Pemakai diminta meng-input id game yang ingin dicari

tahun rilis → Pemakai diminta meng-input tahun rilis game yang ingin dicari

Output:

Diberikan data dari game yang dicari bisa dari salah satu parameter atau dua-duanya. Apabila dua-duanya dikosongkan maka diberikan seluruh data game yang dimiliki.

10. search_game_at_store (database)

Input:

id → Pemakai diminta meng-input id game yang ingin dicari

nama game → Pemakai diminta meng-input nama game yang ingin dicari

kategori → Pemakai diminta meng-input kategori game yang ingin dicari

tahun rilis → Pemakai diminta meng-input tahun rilis game yang ingin dicari

harga → Pemakai diminta meng-input harga game yang ingin dicari

Output:

Diberikan data dari game yang dicari bisa dari salah satu parameter atau semuanya. Apabila semuanya dikosongkan maka diberikan seluruh data game yang dimiliki toko.

11. topup (database)

Input:

username → Pemakai diminta meng-input username yang ingin ditambah saldonya

saldo → Pemakai diminta meng-input berapa saldo yang ingin ditambahkan atau dikurangkan (+'saldo', -'saldo')

Output:

Data penambahan atau pengurangan saldo akan dimasukkan ke dalam database dan diberikan pesan topup berhasil

12. history (database)

Output:

Diberikan data pembelian username yang terlogin seperti game yang dibeli beserta informasi id, harga, dan tahun dibelinya game tersebut.

13. help (database)

Output:

Ditampilkan panduan penggunaan sistem. panduan dari semua prosedur akan ditampilkan juga kegunaan setiap prosedur tersebut.

14. load (database)

Input:

nama_folder → pemakai akan di-prompt untuk mengisi nama_folder sesuai dengan nama folder penyimpanan data .csv

Output:

Menge-load data dari database kedalam sistem serta menampilkan pesan loading berhasil dan meminta pemakai untuk memencet tombol apa saja untuk melanjutkan

15. save (database, folderdatabase)

Input:

nama folder penyimpanan → pemakai akan di-prompt untuk mengisi nama folder untuk menyimpan data-data yang baru saja dibuat atau dijalankan

Output:

jika input nama folder belum tersedia, akan ada folder baru dengan nama sesuai input, tetapi jika sudah ada, maka file database akan dihapus dan diganti dengan yang baru (replaced)

16. exit_program (database, folderdatabase)

Input:

perintah → pemakai akan di-prompt untuk mengisi Y atau N (Y, y, N, n)

Output:

user akan keluar dari program. Jika user memasukan input Y sebelumnya, maka akan dilakukan penyimpanan file yang sudah diubah. Sebaliknya, jika N, tidak dilakukan penyimpanan file yang sudah diubah

17. encrypt(password)

Input:

password → password yang dimasukan pemakai akan di enkripsi

Output:

password yang telah terenkripsi akan direturn sesuai aturan

18. kerangajaib(question)

Input:

perintah → pemakai akan di-prompt untuk mengisi pertanyaan

Output:

Program akan memberikan jawaban random berdasarkan randomizer LCG

19. tictactoe(X,Y)

Input:

perintah → pemakai akan di-prompt untuk mengisi absis dan ordinat dari papan tictactoe

Output:

Program akan menjalankan operasi dan menampilkan board untuk tictactoe

Desain Kamus Data

GLOBAL VARIABLE

user : 2D array of string { Array yang mengandung data dari user.csv }
game: 2D array of string { Array yang mengandung data dari game.csv }
riwayat: 2D array of string { Array yang mengandung data dari riwayat.csv }
kepemilikan: 2D array of string {Array yang mengandung data dari kepemilikan.csv }
current_id : string { ID user yang login }
current_username : string {Username dari user yang login }
isLoggedIn : boolean { State apakah user login atau tidak }

F02 - register

KAMUS LOKAL

id : string { id user baru }
username : string { username user baru }
name : string { nama user baru }
password : string { password user baru }

KAMUS RETURNED VARIABLE

user : 2D array of string { Array yang sudah diupdate}

F03 - login

KAMUS LOKAL

username : string { username user }
password : string { password user }

KAMUS RETURNED VARIABLE

isLoggedIn : bool {login state dari program}
current_id : string {id dari akun yang login}
current_user : string {username dari akun yang login}

F04 - tambah_game

KAMUS LOKAL

id : integer { id game baru }
nama : string { nama game }
kategori : string { kategori game }
tahun_rilis : string { tahun rilis game }
harga : string { harga game }
stok : string { stok }

KAMUS RETURNED VARIABLE

game : 2D array of string { Array yang sudah diupdate}

F05 - ubah_game

KAMUS LOKAL

id : integer { id game baru }
nama : string { nama game }
kategori : string { kategori game }
tahun_rilis : string { tahun rilis game }
harga : string { harga game }

KAMUS RETURNED VARIABLE

game : 2D array of string { Array yang sudah diupdate}

F06 - ubah_stok

KAMUS LOKAL

id : integer { id game baru }
stok : string { stok }

KAMUS RETURNED VARIABLE

game : 2D array of string { Array yang sudah diupdate}

F07 - list_game_toko

KAMUS LOKAL

sorter : string { metode sorter }
res : array of string { stok }

F08 - buy_games

KAMUS LOKAL

id : integer { id game yang dicari }
user_ : string { id user }
user_idx : integer { index user }
game : string { id game }
game_idx : integer { index game }

KAMUS RETURNED VARIABLE

game : 2D array of string { Array yang sudah diupdate}
user : 2D array of string { Array yang sudah diupdate}
kepemilikan : 2D array of string { Array yang sudah diupdate}
riwayat : 2D array of string { Array yang sudah diupdate}

F09 - list_game

KAMUS LOKAL

sorted : 2D Array of string { array game yang sudah di sort }
res : array of string { array yang sudah di filter }

F10 - search_my_game

KAMUS LOKAL

id: string { id dari game}
name: name { nama dari game}
res : array of string { array yang sudah di filter }

F11 - search_game_at_store

KAMUS LOKAL

id : integer { id game }
name : string { nama game }
category : string { kategori game }
year : string { tahun rilis game }
price : string { harga game }
res : array of string
parameter : 2D array of string

F12 - topup

KAMUS RETURNED VARIABLE

user : 2D array of string { Array yang sudah diupdate}

F13 - riwayat

KAMUS LOKAL

res : array of string { temporary variable }

F14 - help

KAMUS LOKAL

role : string { role dari user atau admin }

F15 - load

KAMUS LOKAL

user : 2D array of string
game : 2D array of string
kepemilikan : 2D array of string
riwayat : 2D array of string
folder : string
f : SEQFILE OF
 (*) raw_lines : array of string
 (1) mark : None

KAMUS RETURNED VARIABLE

user : 2D array of string { Array yang mengandung data dari user.csv }
game: 2D array of string { Array yang mengandung data dari game.csv }
riwayat: 2D array of string { Array yang mengandung data dari riwayat.csv }
kepemilikan: 2D array of string {Array yang mengandung data dari kepemilikan.csv }

F16 - save

KAMUS LOKAL

user : 2D array of string
game : 2D array of string
kepemilikan : 2D array of string
riwayat : 2D array of string
folder : string

KAMUS RETURNED VARIABLE

f : SEQFILE OF
 (*) raw_lines : array of string
 (1) mark : None
user : csv { user.csv }
game: csv { game.csv }
riwayat: csv { riwayat.csv }
kepemilikan: csv { kepemilikan.csv }

F17 - exit

KAMUS LOKAL

var : array of char
user_input : string

B01- Cipher

KAMUS LOKAL

lower_bound : integer
upper_bound : integer
dev : integer
daftar_ord : array of integer
check : integer

KAMUS RETURNED VARIABLE

encrypted : string

B02- Magic Conch Shell

KAMUS LOKAL

q : string
lcg : integer
ans : array of string

B03- Tictactoe

KAMUS LOKAL

board : array of array of char
player : char
turn : char
won : boolean
X,Y : integer

Desain Dekomposisi Algoritmik dan Fungsional Program

F02 - register

INPUT user_attribute
VALIDATE user_attribute
WRITE user_array
OUTPUT ("Register success")

F03 - login

INPUT username_and_password
READ user_array
VALIDATE username_and_password
OUTPUT ("Login success")

F04 - tambah_game

INPUT game_attribute
VALIDATE game_attribute
READ last_game_id
WRITE new_game_to_game_array
OUTPUT ("Game Berhasil ditambahkan")

F05 - ubah_game

INPUT game_attribute
VALIDATE game_id
READ game_array
REWRITE game_array
OUTPUT ("Game Berhasil diubah")

F06 - ubah_stok

INPUT game_id_and_game_stock
VALIDATE game_id
READ game_array
REWRITE game_stock
OUTPUT ("Stock game berhasil diubah")

F07 - list_game_toko

INPUT sorter
VALIDATE sorter
READ game_array
OUTPUT sorted_array

F08 - buy_games

INPUT game_id_or_game_name
VALIDATE game_id_or_game_name

READ game_array__user_array_kepemilikan_array_and_riwayat_array
REWRITE all_array
OUTPUT (“Berhasil membeli game”)

F09 - list_game

VALIDATE user_array_kepemilikan_array_and_game_array
READ all_array
OUTPUT sorted_array

F10 - search_my_game

INPUT game_attribute
VALIDATE game_attribute
READ game_array
OUTPUT searched_game

F11 - search_game_at_store

INPUT game_attribute
VALIDATE game_attribute
READ game_array
OUTPUT searched_game

F12 - topup

INPUT user_id_and_saldo
VALIDATE user_id_and_saldo
READ user_array
OUTPUT (“Saldo berhasil diubah”)

F13 - riwayat

INPUT user_id_and_saldo
VALIDATE user_id_and_saldo
READ user_array
OUTPUT (“Saldo berhasil diubah”)

F14 - help

VALIDATE role
OUTPUT help_list

F15 - load

INPUT foldername
READ csv_inside_folder
REWRITE csv_to_array
OUTPUT (“Loading Berhasil”)

F16 - save

INPUT foldername
VALIDATE folder_exist
WRITE array_to_csv
OUTPUT (“Save Berhasil”)

F17 - exit

INPUT user_choice
VALIDATE user_choice
RUN save_or_not_save
OUTPUT (“Terima kasih sudah bermain!”)

B01- Cipher

INPUT password
ENCRYPT password
OUTPUT Encrypted_Password

B02- Magic Conch Shell

INPUT user input
RANDOMIZE answer_index
OUTPUT answer_at_answer_index

B03- Tictactoe

INPUT user_input
VALIDATE user_input
OUTPUT board
VALIDATE board_and_winner
OUTPUT winner

Spesifikasi untuk tiap Modul/Prosedur/Fungsi

F02 - register

```
procedure register(input:id,username,name,password)
{ Prosedur untuk melakukan register ke BNMO (Hanya bisa dilakukan admin) }
{ I.S. Input bisa saja kosong atau tidak valid }
{ F.S. Menambahkan data user baru ke dalam matriks user }
```

ALGORITMA

```
new_user <- [] -- Assign Empty array
-- Input data from admin
input(id)
input(username)
input(name)
input(password)
-- Validate the data
validateUser
validatePassword
if (length(id) = 0 || # if the input is empty
    length (username) = 0 ||
    length(name) = 0||
    length(password) = 0) then
    input(id)
    input(username)
    input(name)
    input(password)
else
    new_user <- [id,username,name,password] # Reassign the array to new array
    containing input data
    addObj(user, new_user)
    output("Registrasi sudah berhasil ^_^")
```

F03 - login

```
procedure login(input:username,password)
{ Prosedur untuk melakukan login ke BNMO}
{ I.S. Input bisa saja kosong atau tidak valid }
{ F.S. Mengembalikan state login, username dan password user yang login ke global}
```

ALGORITMA

```
while True do
    -- Input data from admin
    input(username)
    input(password)
```

```

-- Validate the data
iterate
    if username /= user[i][1] and password /= user[i][3]
        output("Yeyy, kamu berhasil login. Selamat Datang {username}!")
        return True,user[i][0]{id},user[i][1]
    else
        output("Username anda tidak ditemukan atau password anda salah")
until (return)

```

F04 - tambah_game

```

procedure tambah_game(input:nama, kategori, tahun_rilis, harga,stok)
{ Prosedur untuk melakukan menambahkan game ke BNMO }
{ I.S. Input bisa saja kosong atau tidak valid }
{ F.S. Menambahkan data game baru ke dalam matriks game }

```

ALGORITMA

```

while True do
    -- Input data from admin
    input(name)
    input(kategori)
    input(tahun_rilis)
    input(harga)
    input(stok)

    iterate
        if (name = "" || kategori = "" || tahun_rilis = "" || harga = "" || stok = "") then
            output("Mohon masukkan semua informasi mengenai
                    game dan tidak boleh ada ; pada tiap informasi")
        else
            id <- int(game[length(game)-1][0])+1
            game_data <- [id, nama, kategori, tahun_rilis, harga, stok]
            addObj(game, game_data)
            output("Game berhasil ditambahkan!")
            return
    until (return)

```

F05 - ubah_game

```

procedure ubah_game(input:nama, kategori, tahun_rilis, harga)
{ Prosedur untuk melakukan mengubah detail game di BNMO }
{ I.S. Input bisa saja kosong atau tidak valid }
{ F.S. Mengubah detail attribute dari game yang diubah }

```


ALGORITMA

```
while True do
    -- Input data from admin
    input(name)
    input(kategori)
    input(tahun_rilis)
    input(harga)
    input(stok)

    while (length(id) = 0) do
        if (id = "") then
            output("ID game tidak boleh kosong!")

        iterate [0, length(game)]
            if game[i][0] == id then
                if length(nama) > 0 then
                    game[i][1] = nama
                if(length(kategori) > 0) then
                    game[i][2] = kategori
                if(length(tahun_rilis) > 0) then
                    game[i][3] = tahun_rilis
                if(length(harga) > 0) then
                    game[i][4] = harga
                output("Game berhasil diupdate!")
            return
    until (return)
```

F06 - ubah_stok

```
procedure ubah_stok(input:id,stok)
{ Prosedur untuk melakukan menambah atau mengurangi stok game di BNMO}
{ I.S. Input bisa saja kosong atau tidak valid }
{ F.S. Mengubah stok dari game yang diubah }
```

ALGORITMA

```
while True do
    -- Input id from admin
    input(id)

    -- Validate id
    iterate
        if (id = "") then
            output("ID game tidak boleh kosong!")
```

```

stop (length(id) > 0)

-- Input stock
input(stok)

game_, idx <- getGamebyId(game, id)
if length(game_) = 0:
    input("Game dengan ID tersebut tidak ditemukan...")
else:
    res <- int(game_[5])+sum
    if res < 0:
        output("Maaf, stok game {game[idx][1]} gagal dikurangi karena stok
                kurang. Stok sekarang adalah {game[idx][5]} ")
    else:
        game[idx][5] = str(res)
        input("Stok game {game[idx][1]} berhasil diubah. Stok sekarang
                adalah {game[idx][5]}")
    return
until (return)

```

F07 - list_game_toko

```

procedure list_game_toko(input:sorter)
{ Prosedur untuk menampilkan list game toko BNMO}
{ I.S. Input akan divalidasi sampai didapatkan input yang sesuai aturan }
{ F.S. Menampilkan list game toko ke layar }

```

ALGORITMA

```

while True do
    -- input id from client
    input(sorter)

    - validate the input
    if length(sorter) = 0 then
        idx <- 0 # assign index
        res <- removeFirstElmt(game) # remove first element(header)
        res <- sortMatrix(res, idx, '+') # sorting the matrix by index
        printGame(res, 1) # print list of games
        return
    if (sorter = 'tahun+') then
        idx <- 3
        res <- removeFirstElmt(game)
        res <- sortMatrix(res, idx, '+')
        printGame(res, 1)

```

```

        return
    if (sorter = 'tahun-') then
        idx <- 3
        res <- removeFirstElmt(game)
        res <- sortMatrix(res, idx, '-')
        printGame(res, 1)
        return
    if (sorter = 'harga+') then
        idx <- 4
        res <- removeFirstElmt(game)
        res <- sortMatrix(res, idx, '+')
        printGame(res, 1)
        return
    if (sorter = 'harga-') then
        idx <- 4
        res <- removeFirstElmt(game)
        res <- sortMatrix(res, idx, '-')
        printGame(res, 1)
        return
    else:
        input("Skema sorting tidak valid")

```

F08 - buy_games

```

procedure buy_games(input:user_id)
{ Prosedur untuk membeli game di BNMO oleh user}
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi}
{ F.S. Menambahkan game ke inventory (kepemilikan user) sekaligus mengubah
parameter-parameter yang berkaitan }

```

ALGORITMA

```

while True do
    -- Input data from user
    input(id)

    user_, user_idx <- getUserById(user, user_id)
    game_, game_idx <- getGamebyId(game, game_id)

    -- Validate and add to database
    saldo <- int(getValue(user, user_id))
    harga <- 0
    iterate (i<- 0, length(game))
        if game_idx = game[i][0]:
            harga += int(game[i][4])

```

```

    if(isOnLibrary(user_, game_, kepemilikan)) then
        output("Anda sudah memiliki game ini!")
    else if(saldo < harga) then
        output("Saldo anda tidak cukup untuk membeli game ini!")
    else if(int(user[user_idx][5]) <= 0) then
        output("Stok game tersebut sedang tidak ada")
    else:
        user[user_idx][4] <- str(saldo - harga)
        game[game_idx][5] <- str(int(game[game_idx][5])-1)
        write_riwayat(user[user_idx], game[game_idx], riwayat)
        output(f"Game {game_[1]} telah berhasil dibeli!")
    return
until(return)

```

```

function isOnLibrary(input: user_data)->boolean
{ fungsi untuk memvalidasi apakah game sudah dimiliki oleh user }

```

```

function getUserById(input:user_id) -> array
{ fungsi untuk mendapatkan data lengkap user }

```

```

function getGameById(input:game_id) -> array
{ fungsi untuk mendapatkan data lengkap game }

```

```

function write_history() -> array
{ fungsi untuk menuliskan riwayat pembelian }

```

F09 - list_game

```

procedure list_game(input:user_id)
{ Prosedur untuk menampilkan list game user yang login ke BNMO}
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi }
{ F.S. Menampilkan list game user ke layar}

```

ALGORITMA

```

while True do
    -- Input data from user
    sorted <- sortMatrix(game, 0, '+')
    game <- removeFirstElmt(game)
    game_owned <- ownedByUser(user_id, kepemilikan)
    res <- getOwnedGames(sorted, game_owned)
    if res = [] then
        output("Maaf kamu belum membeli game ini. Ketik perintah 7 di main menu
            untuk membeli game")
    else:
        printGame(res, 2)

```

```

        return
until(return)

```

F10 - search_my_game

```

procedure search_my_game(input:game_id,name)
{ Prosedur untuk menampilkan game yang dicari dan dimiliki oleh user BNMO}
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi }
{ F.S. Menampilkan list game yang dicari user ke layar}

```

ALGORITMA

```

while True do
    -- Input data from user
    input(id)
    input(name)

    -- Validate Data
    while id = " && name = " do
        output("Parameter tidak boleh kosong. Silahkan isi minimal 1 parameter untuk
            menjalankan operasi ini")
        input(id)
        input(name)
    game <- removeFirstElmt(game)
    game_owned <- ownedByUser(user_id, kepemilikan)
    owned_games <- getOwnedGames(sortMatrix(game, 0, '+'), game_owned)
    res <- []

    iterate (i <- 0, length(owned_games))
        if owned_games[i][0] = id || owned_games[i][1] = name then
            res += owned_games[i]

        if (length(res) /= 0):
            printGame(res, 1)
            return
        else:
            output("Tidak ada game yang memenuhi kriteria")
            return
until(return)

```

F11 - search_game_at_store

```

procedure search_my_game(input:id,name,price,category,year)
{ Prosedur untuk menampilkan game yang dicari oleh user di database game toko BNMO}
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi }
{ F.S. Menampilkan list game toko yang dicari user ke layar}

```

ALGORITMA

```
while True do
  -- Input data from user
  input(id)
  input(name)
  input(price)
  input(category)
  input(year)

  -- Validate data
  while id == " " || name == " " do
    output("Parameter tidak boleh kosong. Silahkan isi minimal 1 parameter untuk
           menjalankan operasi ini")
    input(id)
    input(name)
    input(price)
    input(category)
    input(year)
  -- create empty array for variable parameter
  parameter <- [[], [], [], [], []]
  res <- []

  -- cases for empty input and non-empty input
  if length(id) > 0 then
    addObj(parameter[0], id)
  else:
    iterate (i <- 0, length(game))
      addObj(parameter[0], game[i][0])
  if length(name) > 0 then
    addObj(parameter[1], name)
  else:
    iterate (i <- 0, length(game))
      addObj(parameter[1], game[i][1])

  if length(price) > 0 then
    addObj(parameter[2], id)
  else:
    iterate (i <- 0, length(game))
      addObj(parameter[4], game[i][4])

  if length(category) > 0 then
    addObj(parameter[3], id)
  else:
```

```

        iterate (i <-0, length(game))
            addObj(parameter[2], game[i][2])

    if length(year) > 0 then
        addObj(parameter[4], id)
    else:
        iterate (i <-0, length(game))
            addObj(parameter[3], game[i][3])

    -- searching game array for searched game
    iterate (i <-0, length(game))
        if (game[i][0] in parameter[0] &&
            game[i][1] in parameter[1] &&
            game[i][2] in parameter[2] &&
            game[i][3] in parameter[3] &&
            game[i][4] in parameter[4]) then
            addObj(res, game[i])

    if (length(res) > 0) then
        printGame(res, 1)
        return
    else
        print("Tidak ada game yang memenuhi kriteria")
        return
until(return)

```

F12 - topup

```

procedure topup(input:username,saldo)
{ Prosedur untuk mengubah saldo dari user BNMO}
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi }
{ F.S. Mengubah saldo dari user berdasarkan input dari admin}

```

ALGORITMA

```

while True do
    -- Input data from user
    input(validateUser(username))
    input(saldo)

    -- Cases for success and fail
    iterate (i <- 0, length(user)):
        if user[i][1] = username then
            if int(user[i][5]) + saldo < 0:
                print("Masukkan tidak valid")

```

```

        return
    else:
        user[i][5] <- str(int(user[i][5]) + saldo)
        output("Topup berhasil...")
        return
until(return)

```

F13 - riwayat

```

procedure riwayat(input:user_id)
{ Prosedur untuk mencetak riwayat pembelian user}
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi }
{ F.S. Menampilkan riwayat pembelian game user}

```

ALGORITMA

```

if length(riwayat) = 0 then
    input("Kamu tidak memiliki riwayat pembelian. Ketik 7 pada menu utama untuk
        membeli game.")
else
    riwayat <- removeFirstElmt(riwayat)
    res <- getUserHistory(user_id, riwayat)
    print_riwayat(res)

```

```

function print_riwayat(input matrix) -> array
{ fungsi untuk mencetak riwayat pembelian }

```

KAMUS LOKAL

i : integer

ALGORITMA

```

id_len <- (length("ID"), getLongestStrLen(matrix, 0))[
    getLongestStrLen(matrix, 0) > length("ID")]
game_len <- (length("Nama Game"), getLongestStrLen(matrix, 1))[
    getLongestStrLen(matrix, 1) > length("ID")]
price_len <- (length("Harga"), getLongestStrLen(matrix, 4))[
    getLongestStrLen(matrix, 4) > length("Harga")]
year_len <- (length("Tahun Pembelian"), getLongestStrLen(matrix, 3))[
    getLongestStrLen(matrix, 3) > length("Tahun Pembelian")]
output("ID", " "*(id_len-length("ID")), "|",
    "Nama Game", " "*(game_len-length("Nama Game")), "|",
    "Harga", " "*(price_len - length("Harga")), "|",
    "Tahun Beli", " " * (year_len-length("Tahun Beli")))
output("="*(id_len+game_len+price_len+year_len))
traversal (i <- 0, length(matrix))
    output(matrix[i][0], " "*whole_numerate(int(id_len)-length(matrix[i][0])), "|",
        matrix[i][1], " " *

```



```

whole_numerate(int(game_len)-length(matrix[i][1])), "|",
matrix[i][2], " " *
whole_numerate(int(price_len)-length(matrix[i][2])), "|",
matrix[i][4], " " * whole_numerate(int(year_len)-length(matrix[i][4])),)

```

```

function getUserHistory(input:user_id) -> array
{ fungsi untuk mendapatkan riwayat pembelian user }

```

KAMUS LOKAL

i : integer

ALGORITMA

```

res = []
iterate(i <- 0, length(riwayat))
  if riwayat[i][3] = user_id then
    addObj(res, riwayat[i])
-> res

```

F14 - help

procedure help

```

{ Prosedur untuk mencetak help }
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi }
{ F.S. Menampilkan list help sesuai role user ke layar }

```

ALGORITMA

```

output('=====Need Help?=====')
if role = 'admin' then
  output('1. Login - Untuk login ke akun')
  output('2. Register - Untuk mendaftarkan akun')
  output('3. Tambah Game - Untuk menambah game yang dijual oleh toko')
  output('4. Ubah Game - Untuk mengubah detail game')
  output('5. List Game Toko - Untuk melihat list game yang dijual toko')
  output('6. Ubah Stok - Untuk mengubah stok game toko')
  output('7. List Game - Untuk menampilkan list game yang dijual toko')
  output('8. Cari Game Toko - Untuk mencari game di Toko')
  output('9. TopUp - Untuk menambahkan saldo User')
  output('10. Help - Untuk melihat list command')
  output('11. Save - Menyimpan Data')
  output('12. Exit - Mengakhiri Program')
else:
  output('1. Login - Untuk login ke akun')
  output('2. List Game Toko - Untuk melihat list game yang dijual toko')
  output('3. Beli Game - Untuk membeli game di Toko')
  output('4. List Game User - Untuk melihat list game yang dimiliki')
  output('5. Search Game - Untuk mencari game yang dimiliki')
  output('6. Cari Game Toko - Untuk mencari game di Toko')

```

```

output('7. History - Melihat Riwayat Pembelian')
output('8. Help - Untuk melihat list command')
output('9. Save - Menyimpan Data')
output('10. Exit - Mengakhiri Program')

```

F15 - load

```

procedure load
{ Prosedur untuk mengload data dari csv ke program }
{ I.S. folder bisa saja tidak ada }
{ F.S. Mengassign file csv ke masing-masing array nya}

```

ALGORITMA

```

folder <- argParser
user <- csv_reader(folder, "user")
game <- csv_reader(folder, "game")
kepemilikan <- csv_reader(folder, "kepemilikan")
riwayat <- csv_reader(folder, "riwayat")
output('Loading berhasil...')
return user, game, kepemilikan, riwayat

```

```

function argParser()->string
{fungsi untuk membaca argumen di command line }

```

KAMUS LOKAL

folder : string

ALGORITMA

```

parser <- argparse.ArgumentParser(description='argument')
parser.add_argument('foldername', type=str)
args <- parser.parse_args()
folder <- args.foldername
-> folder

```

```

function csv_reader(input:folder, filename)->array
{fungsi untuk membaca csv dan menerjemahkannya ke dalam bentuk matriks 2D}

```

```

OPEN({filename}.csv, f)
result = []
traversal each in f:
    row <-[]
    tmp <-""
    delimiter <-";"
    traversal ch in each:
        if ch = delimiter || ch = "\n" then
            row <- addObj(row, tmp)
            tmp <- ""

```

```

else:
    tmp <-tmp + ch
result <-addObj(result, row)
-> result

```

F16 - save

procedure save

```

{ Prosedur untuk mengsave data ke csv }
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi }
{ F.S. Menuliskan array-array data menjadi bentuk csv kembali }

```

ALGORITMA

```

folder <- strip_str(input("Masukkan nama folder penyimpanan: "))
create_folder(folder)
write_csv(folder, user, "user")
write_csv(folder, game, "game")
write_csv(folder, kepemilikan, "kepemilikan")
write_csv(folder, riwayat, "riwayat")
output("Data berhasil tersimpan...")

```

```

function write_csv(input:folder, matriks, filename)->array
{fungsi untuk menuliskan csv dari matriks yang berkaitan }

```

KAMUS LOKAL

res : string

ALGORITMA

```

OPEN({filename}.csv, f)
res <- ""
    traversal [0, length(data)]
        for j in range(length(data[i]))
            if j = length(data[i]) - 1 then
                res <- res + str(data[i][j])
            else:
                res <- res + "{data[i][j]};"
        res <- + "\n"
WRITE(f, res)
CLOSE (f)

```

```

procedure create_folder(input:folder)
{prosedur untuk membuat folder jika folder tidak ada di path }

```

KAMUS LOKAL

path : string

ALGORITMA

```
path <- "/"
os.chdir(path)
if !(os.path.exists(folder)) then
  os.mkdir(folder)
```

F17 - exit

procedure exit

```
{ Prosedur untuk keluar dari program }
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi }
{ F.S. Mengsave data (atau tidak) dan keluar dari program }
```

ALGORITMA

```
input(user_input)
var <- ['y', 'Y', 'n', 'N']
while True do
  if user_input in var then
    break
  else
    output("Tolong berikan input yang benar atau nanti BNMO marah >:(")
    output("=====")
    input(user_input)

if user_input.lower() = 'y' then
  save(user, game, kepemilikan, riwayat)
  output("Terima kasih sudah bermain :D")
```

B01 - cipher

procedure cipher(input : password)

```
{ Prosedur untuk mengenkripsi password }
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi }
{ F.S. Mengsave data (atau tidak) dan keluar dari program }
```

ALGORITMA

```
encrypt(password)
outputting(encrypted)
function encrypt(input:string)->array
{fungsi untuk mengenkripsi setiap char di password menjadi char lain dengan sebuah key }
```

ALGORITMA

```
daftar_ord <- []
dev <- 7
```

```

iterate (i <- 0 , length(string))
  daftar_ord <- daftar_ord + [ord(string[i])]
iterate (j <-0, length(daftar_ord))
  check <- 0
  check <- daftar_ord[j] + dev
  if check /= 59:
    if lowerbound <= check <= upperbound then
      pass
    else:
      check <- check - ( upperbound - lowerbound + 1)
      daftar_ord[j] <- check
  else
    daftar_ord[j] <- 126
  dev <- dev + 1
until ( j >= length(daftar_ord))

-> daftar_ord

```

```

function outputting (input : array) -> string
{ fungsi untuk mengubah array enkripsi menjadi string }

```

ALGORITMA

```

encrypted <- ""
iterate (i <- 0, length(array))
  encrypted <- encrypted + chr(array[i])
-> encrypted

```

B02 - kerangajaib

```

procedure kerangajaib(input : q)
{ Prosedur untuk memainkan magic conch shell }
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi }
{ F.S. Menampilkan jawaban random berdasarkan LCG }

```

ALGORITMA

```

while True do
  try:
    q <- strip_str(input(q)) # User Input
    lcg <- LCG() # Randomize using Linear Congruential Generator

    if q = " then
      output("ok")
      -> None
    else:
      ans <- ['Y', 'G', 'Mayhaps', 'g dl y ges y', 'yntkts', # Random Answer

```

```

        'Mungkin', 'Oh jelas', 'p mkst?', 'dih, apaan dah ☹_☹']
    output(ans[lcg mod 8])
except:
    output("Input mu ndak valid mas e")
until(return)

```

B03 - tictactoe

```

procedure tictactoe(input : x,y)
{ Prosedur untuk memainkan tictactoe }
{ I.S. Input bisa saja kosong atau tidak valid dan harus divalidasi }
{ F.S. Menampilkan board tictactoe dan berhenti saat ada yang menang }

```

ALGORITMA

```

turn <- 'X'
board <- [[' ', ' ', ' '], [' ', ' ', ' '], [' ', ' ', ' ']]
won <- False
while won /= True do
    os.system("cls")
    output("Player {turn} turn!")
    printBoard(board)
    while True do
        try:
            X = int(input(X))-1 # X Axis
            Y = int(input(Y))-1 # Y Axis

            if (X < 0 or X > 2) || (Y < 0 or Y > 2) then
                output("Inputnya yang benar ya sayang ya")

            if checkboard(board, X, Y, turn) = False then
                pass
            else
                break
        except ValueError:
            output("Inputnya yang benar ya sayang ya")
            continue

    board <- modBoard(board, X, Y, turn)
    printBoard(board)
    won <- check(board, turn)
    if turn = 'X' then
        turn <- 'O'
    elif turn = 'O' then
        turn <- 'X'
    if won = True then

```

```
break
output("Lanjut...")
```

```
function modBoard (input : array,x,y,turn) -> array
{ fungsi untuk mengubah board }
```

ALGORITMA

```
board[X][Y] <- turn
-> board
```

```
function printBoard (input : array) -> None
{ fungsi untuk menampilkan board }
```

ALGORITMA

```
output("1 2 3")
iterate(i<-0, length(board))
  iterate(j<-0, length(board[i])):
    if j = 2 then
      output(board[i][j])
    else
      output(board[i][j],"|")
output("-+-+--")
```

```
function checkBoard (input : array,x,y,turn) -> boolean
{ fungsi untuk mengvalidasi board }
```

ALGORITMA

```
if board[X][Y] = "X" && turn == "O" then
  output("Kolomnya udah ada yang punya mas/mbak e :(")
  -> False
else if board[X][Y] = "O" && turn = "X" then
  output("Kolomnya udah ada yang punya mas/mbak e :(")
  -> False
else if board[X][Y] /= " ":
  print("Kolomnya udah ada di isi mas/mbak e :(")
  -> False
else:
  -> True
```

```
function filled (input : array) -> bool
{ fungsi untuk melihat apakah board sudah terisi full }
```

ALGORITMA

```
board[X][Y] <- turn
-> board
```

```
function check(input : array,player) -> bool
{ fungsi untuk melihat apakah board sudah terisi full }
```

ALGORITMA

```
if filled(board) then
    output("It's a tieeeeeeee!!!!!!")
    -> True
if (board[0][0] = board[0][1] = board[0][2] = 'X') || (board[0][0] = board[0][1] =
board[0][2] = 'O') then
    output("Player {player} Wins.")
    -> True
else if (board[1][0] = board[1][1] = board[1][2] = 'X') || (board[1][0] = board[1][1] =
board[1][2] = 'O') then
    output("Player {player} Wins.")
    -> True
else if (board[2][0] = board[2][1] = board[2][2] = 'X') || (board[2][0] = board[2][1] =
board[2][2] = 'O') then
    output("Player {player} Wins.")
    -> True
else if (board[0][0] = board[1][0] = board[2][0] = 'X') || (board[0][0] = board[1][0] =
board[2][0] = 'O') then
    output("Player {player} Wins.")
    -> True
else if (board[0][1] = board[1][1] = board[2][1] = 'X') || (board[0][1] = board[1][1] =
board[2][1] = 'O') then
    output("Player {player} Wins.")
    -> True
else if (board[0][2] = board[1][2] = board[2][2] = 'X') || (board[0][2] = board[1][2] =
board[2][2] = 'O') then
    output("Player {player} Wins.")
    -> True
elif (board[0][0] = board[1][1] = board[2][2] = 'X') || (board[0][0] = board[1][1] =
board[2][2] = 'O'):
    output("Player {player} Wins.")
    -> True
elif (board[2][0] = board[1][1] = board[0][2] = 'X') || (board[2][0] = board[1][1] =
board[0][2] = 'O') then
    output("Player {player} Wins.")
    -> True
else
    -> False
```


base

function strip_str() -> string

{ Menghilangkan trailing spaces di depan dan dibelakang input }

KAMUS LOKAL

res : string

begin,end : integer

i : integer

ALGORITMA

if(length(s) = 0) then

return ""

res <- ""

begin <- 0

end <- length(s)-1

while s[begin] = " " do

begin = begin + 1

while s[end] = " " do

end <- end - 1

end <- end + 1

traversal [i <- begin, end]:

res <- res + s[i]

-> res

function length(input : array) -> integer

{ Menghitung panjang dari string dan juga array }

KAMUS LOKAL

element : integer

ALGORITMA

element = 0

iterate:

element += 1

stop(EOP(array))

-> element

function validateUser() -> string

KAMUS LOKAL

username_ : string

ALGORITMA

{ Menerima input username dan memvalidasinya }

while True do

try:

username_ <- strip_str(input("Masukkan Username: "))

```

if not((conUpper(username_) || conLower(username_))
      && conSpec(username_)
      && conDigit(username_)) then
  print("Username tidak sesuai dengan ketentuan. Silahkan coba kembali!")
else
  iterate (i <- 0, length(user)) then
    if(user[i][1] = username_) then
      output("Username ini sudah diambil. Gunakan username lain!")
    else:
      -> username_
except ValueError:
  print("Input anda tidak valid. Silahkan ulangi input")
until ( return )

```

```

function validatePassword() -> string
{ Menerima input password dan memvalidasinya }

```

KAMUS LOKAL

element : integer

ALGORITMA

```

while True do
  try:
    output(
      ""
      Password harus :
      - lebih dari 7 kata
      - mengandung huruf besar atau kecil
      - mengandung digit angka
      ""
    )
    password_ <- strip_str(input("Masukkan Password: "))
    if not((conUpper(password_) or conLower(password_))
          and length(password_) > 7
          and conDigit(password_)) then
      output("Password tidak sesuai dengan ketentuan. Silahkan coba kembali!")
    else:
      -> password_
  except:
    output("Input anda tidak valid. Silahkan ulangi input")
until(return)

```

```

function addObj(input : array) -> array
{ Menambahkan objek ke dalam array }

```

KAMUS LOKAL

ALGORITMA

```
list_par <- list_par + [param]  
-> list_par
```

```
function conUpper(input : string) -> boolean  
{ Mengecek apakah string mengandung huruf kapital }
```

KAMUS LOKAL

```
res : boolean  
i : integer
```

ALGORITMA

```
res <- False  
traversal(i <- 0, length(string))  
  if (string[i] = string[i].upper()) then  
    res <- True  
    break  
-> res
```

```
function conLower(input : string) -> boolean  
{ Mengecek apakah string mengandung huruf kecil }
```

KAMUS LOKAL

```
res : boolean  
i : integer
```

ALGORITMA

```
res <- False  
traversal(i <- 0, length(string))  
  if (string[i] = string[i].lower()) then  
    res <-  
    break  
-> res
```

```
function conDigit(input : string) -> boolean  
{ Mengecek apakah string mengandung angka }
```

KAMUS LOKAL

```
res : boolean  
constant num : array of character { ["1", "2" .. "9"] }  
i, j : integer
```

ALGORITMA

```
res <- False  
traversal(i <- 0, length(string))
```

```

    traversal(j <-0, length(num))
    if (string[i] = num[j]) then
        res <- True
        break
-> res

```

```

function conSpec(input : string) -> boolean
{ Mengecek apakah string mengandung simbol spesial _ atau - }

```

KAMUS LOKAL

```

res : boolean
constant spec: array of character { [ "-", "_" ] }
i, j : integer

```

ALGORITMA

```

res <- False
traversal(i <-0, length(string))
    traversal (j <-0, length(num))
        if (string[i] = num[j]) then
            res <- True
            break
-> res

```

```

function absolute(input : integer) -> integer
{ Fungsi nilai mutlak}

```

KAMUS LOKAL

ALGORITMA

```

if (num < 0) then
    num <- (-num)
-> num

```

```

function removeFirstElmt(input : array) -> array
{ Menghilangkan elemen paling pertama dalam array }

```

KAMUS LOKAL

```

res : array
i : integer

```

ALGORITMA

```

res <- []
traversal(i <- 0, length(arr))

```

```

    if i = 0 then
        continue
    res <- res + [arr[i]]
-> res

```

function ownedByUser(input:user_id,kepemilikan) -> array of string
 { Mencari game yang dimiliki oleh user dari kepemilikan berdasarkan user id }

KAMUS LOKAL

res : array
 i : integer

ALGORITMA

```

res <- []
traversal(i <- 0, length(kepemilikan))
    if (kepemilikan[i][1] = user_id) then
        res <- res + kepemilikan[i][0]
-> res

```

function getOwnedGames(input:sorted,game_owned) -> array of string
 { Mencari game yang dimiliki oleh user untuk ditampilkan ke output }

KAMUS LOKAL

res : array
 i : integer

ALGORITMA

```

res <- []
traversal(i <- 0, length(arr))
    if (arr[i][0] = param) then
        res <- res + arr[i]
-> res

```

function sortMatrix(input:array)->array
 { Mengurutkan matrix berdasarkan sorter ascending atau descending tergantung input }

KAMUS LOKAL

i, j: integer

ALGORITMA

```

if order = "+" then
    traversal (i <-0, length(matrix))
        traversal (j <- i+1, length(matrix))
            if int(matrix[i][param]) > int(matrix[j][param]) then

```

```

        matrix[i], matrix[j] <- matrix[j], matrix[i]
else if order = "-" then
    traversal (i <-0, length(matrix))
    traversal (j <- i+1, length(matrix))
    if int(matrix[i][param]) < int(matrix[j][param]) then
        matrix[i], matrix[j] <- matrix[j], matrix[i]
-> matrix

```

function printGame(input: array)-> none

{ Menampilkan list dari array yang sudah tersusun rapih (untuk F07,F09,F10, dan F11)}

KAMUS LOKAL

id_len, name_len, price_len, cat_len, year_len, stock_len : integer
i : integer

ALGORITMA

```

id_len <- getLongestStrLen(game_mat, 0)
name_len <- getLongestStrLen(game_mat, 1)
price_len <- getLongestStrLen(game_mat, 2)
cat_len <- getLongestStrLen(game_mat, 3)
year_len <- getLongestStrLen(game_mat, 4)
stock_len <- getLongestStrLen(game_mat, 5)

if list_type = 1 then
    output("ID", " "*abs(id_len-length("ID")), "||",
        "Nama Game", " "*abs(name_len-length("Nama Game")), "||",
        "Harga", " "*abs(price_len - length("Harga")), "||",
        "Kategori", " "*abs(cat_len-length("Kategori")), "||",
        "Tahun Rilis", " " * abs(year_len-length("Tahun Rilis")), "||",
        "Stok", " "*abs(stock_len-length("Stok")))
    output("="*(id_len+name_len+price_len+cat_len+year_len+stock_len+25))
    for i in range(0, length(game_mat)) then
        print(game_mat[i][0], " "*abs(id_len-length(game_mat[i][0])), "||",
            game_mat[i][1], " " *
            abs(name_len-length(game_mat[i][1])), "||",
            game_mat[i][4], " " *
            abs(price_len - length(game_mat[i][4])), "||",
            game_mat[i][2], " " *
            abs(cat_len-length(game_mat[i][2])), "||",
            game_mat[i][3], " " *
            abs(year_len-length(game_mat[i][3])), "||",
            game_mat[i][5], " "*abs(stock_len-length(game_mat[i][5])))

```

```

if list_type = 2 then
  output("ID", " "*abs(id_len-length("ID")), "||",
    "Nama Game", " "*abs(name_len-length("Nama Game")), "||",
    "Kategori", " "*abs(cat_len-length("Kategori")), "||",
    "Tahun Rilis", " " *
    abs(year_len-length("Tahun Rilis")), "||",
    "Harga", " "*abs(price_len - length("Harga")), "||",)
  output("="*(name_len+price_len+cat_len+year_len+stock_len+20))
  for i in range(1, length(game_mat)) then
    output(game_mat[i][0], " "*abs(id_len-length(game_mat[i][0])), "||",
      game_mat[i][1], " " *
      abs(name_len-length(game_mat[i][1])), "||",
      game_mat[i][2], " " *
      abs(cat_len-length(game_mat[i][2])), "||",
      game_mat[i][3], " " *
      abs(year_len-length(game_mat[i][3])), "||",
      game_mat[i][4], " "*abs(price_len - length(game_mat[i][4])))

```

Hasil Output dari Program

F02 - Register

```
Selamat datang di interface BNMO \(*^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
register
Masukkan Username:
Username tidak sesuai dengan ketentuan. Silahkan coba kembali!
Masukkan Username: dahlhcapek002_
Masukkan Nama: Hehe Boah

        Password harus :
        - Lebih dari 7 kata
        - mengandung huruf besar atau kecil
        - mengandung digit angka

Masukkan Password:
Password tidak sesuai dengan ketentuan. Silahkan coba kembali!

        Password harus :
        - Lebih dari 7 kata
        - mengandung huruf besar atau kecil
        - mengandung digit angka

Masukkan Password: apaanya01
Registrasi sudah berhasil ^_^|
```

Gambar 1. Hasil output F02 - Register

F03 - Login

```
Selamat datang di interface BNMO \(*^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
login
Masukkan Username:
Masukkan Password:
Username anda tidak ditemukan atau password anda salah
Masukkan Username: hehehabsd_
Masukkan Password: 1992ohyeah
Username anda tidak ditemukan atau password anda salah
Masukkan Username: admin01
Masukkan Password: dizi23_
Username anda tidak ditemukan atau password anda salah
Masukkan Username: dizi23_
Masukkan Password: admin01
Yeyy, kamu berhasil login. Selamat Datang Dizi!|
```

Gambar 2. Hasil output F03 - Login

F04 - Tambah Game

```
Selamat datang di interface BNMO \ (^o^*) /
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
tambah_game
Masukkan nama game: Death on the Nile
Masukkan kategori: Mystery
Masukkan tahun rilis:2021
Masukkan harga: 199999
Masukkan stok awal: 727
Game berhasil ditambahkan!
```

```
Selamat datang di interface BNMO \ (^o^*) /
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
>>> tambah_game
Masukkan nama game:
Masukkan kategori:
Masukkan tahun rilis:
Masukkan harga:
Masukkan stok awal:
Mohon masukkan semua informasi mengenai game dan tidak boleh ada ; pada tiap informasi
Masukkan nama game: |
```

Gambar 3. Hasil output F04 - Tambah Game

F05 - Ubah Game

```
Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
ubah_game
Masukkan ID game:
Masukkan nama game: Death on the Nile
Masukkan kategori: Mystery
Masukkan tahun rilis:2020
Masukkan harga: 699999
ID game tidak boleh kosong!
Masukkan ID game: 10
Game berhasil diupdate!
```

Gambar 4. Hasil output F05 - Ubah Game

F06 - Ubah Stok

```
Selamat datang di interface BNMO \ (^o^*) /
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
ubah_stok
Masukkan ID game: 10
Masukkan jumlah: 999
Stok game Death on the Nile berhasil diubah. Stok sekarang adalah 1726|

Selamat datang di interface BNMO \ (^o^*) /
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
>>> ubah_stok
Masukkan ID game: 9
Masukkan jumlah: -99999
Maaf, stok game Doki Doki Literature Club gagal dikurangi karena stok kurang. Stok sekarang adalah 67 |
```

Gambar 5. Hasil output F06 - Ubah Stok

F07 - List Game Toko

```
Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
list_game_toko

    tahun+ || tahun-
    harga+ || harga-
    Masukkan Skema Pencarian:

ID | Nama Game | Harga | Kategori | Tahun Rilis | Stok
=====
1 | Detroit Become Human | 899999 | Action | 2018 | 100
2 | Sekiro : Shadow Die Twice | 699999 | Adventure | 2020 | 13
3 | Subnautica | 299999 | Survival | 2017 | 16
4 | Forza 5 Horizon | 699999 | Action | 2021 | 86
5 | Cyberbug 2077 | 799999 | Adventure | 2021 | 800
6 | GOW : Ragnarok | 1099999 | Adventure | 2022 | 78
7 | GTA 6 | 6999999 | Adventure | 2029 | 0
8 | RDR 3 | 899999 | Adventure | 2027 | 28
9 | Doki Doki Literature Club | 99999 | Psychological Horror | 2018 | 69
10 | Death on the Nile | 699999 | Mystery | 2020 | 1726
Press any key to continue...|
```

Gambar 6. Hasil output F07 - List Game Toko

F08 - Buy Games

```
Selamat datang di interface BNMO \ (^o^*) /
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
buy_game
Masukkan ID Game: 9
Game Doki Doki Literature Club telah berhasil dibeli!
```

```
Selamat datang di interface BNMO \ (^o^*) /
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
>>> buy_game
Masukkan ID Game: 19
ID tidak valid. Silahkan ulangi
Masukkan ID Game: |
```

Gambar 7. Hasil output F08 - Buy Games

F09 - List Game User

```
Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
list_game
ID | Nama Game | Harga | Kategori | Tahun Rilis
-----
2 | Sekiro : Shadow Die Twice | 699999 | Adventure | 2020
5 | Cyberbug 2077 | 799999 | Adventure | 2021
Press any key to continue...|
```

Gambar 8. Hasil output F09 - List Game User

F10 - Search Game User

```
Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
search_my_game
Masukan ID Game: 2
Masukan Nama Game:
ID | Nama Game | Harga | Kategori | Tahun Rilis | Stok
-----
2 | Sekiro : Shadow Die Twice | 699999 | Adventure | 2020 | 13
Press any key to continue...|
```

```
Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
>>> search_my_game
Masukan ID Game: 19
Masukan Tahun Rilis Game:
Tidak ada game yang memenuhi kriteria
Press any key to continue...|
```

Gambar 9. Hasil output F10 - Search Game User

F11 - Search Game at Store

```
Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit

=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe

Silahkan pilih menu yang diinginkan:
search_game_at_store
Masukan ID Game: 10
Masukan Nama Game:
Masukan Harga Game:
Masukan Kategori Game:
Masukan Tahun rilis:
ID | Nama Game | Harga | Kategori | Tahun Rilis | Stok
=====
10 | Death on the Nile | 699999 | Mystery | 2020 | 727
Press any key to continue...

Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit

=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe

Silahkan pilih menu yang diinginkan:
>>> search_game_at_store
Masukan ID Game: 29
Masukan Nama Game:
Masukan Harga Game:
Masukan Kategori Game:
Masukan Tahun rilis:
Tidak ada game yang memenuhi kriteria
Press any key to continue...
```

Gambar 10. Hasil output F11 - Search Game at Store

F12 - Top up

```
Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit

=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe

Silahkan pilih menu yang diinginkan:
topup
Masukkan Username: tanjungwibu02_
Masukkan saldo: -999
Topup berhasil...

Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit

=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe

Silahkan pilih menu yang diinginkan:
>>> topup
Masukkan Username: intan_222
Masukkan saldo: -999999999999999
Masukkan tidak valid
```

Gambar 11. Hasil output F12 - Top up

F13 - Riwayat

```
Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
riwayat
ID | Nama Game | Harga | Tahun Beli
=====
5 | Cyberbug 2077 | 699999 | 2022
2 | Sekiro : Shadow Die Twice | 699999 | 2022
9 | Doki Doki Literature Club | 99999 | 2022
Press any key to continue...|
```

Gambar 12. Hasil output F13 - Riwayat

F14 - Help

```
Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
help
=====Need Help?=====
1. Login - Untuk login ke akun
2. Register - Untuk mendaftarkan akun
3. Tambah Game - Untuk menambah game yang dijual oleh toko
4. Ubah Game - Untuk mengubah detail game
5. List Game Toko - Untuk melihat list game yang dijual toko
6. Ubah Stok - Untuk mengubah stok game toko
7. List Game - Untuk menampilkan list game yang dijual toko
8. Cari Game Toko - Untuk mencari game di Toko
9. TopUp - Untuk menambahkan saldo User
10. Help - Untuk melihat list command
11. Save - Menyimpan Data
12. Exit - Mengakhiri Program
Press any key to continue...|

Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
>>> help
=====Need Help?=====
1. Login - Untuk login ke akun
2. List Game Toko - Untuk melihat list game yang dijual toko
3. Beli Game - Untuk membeli game di Toko
4. List Game User - Untuk melihat list game yang dimiliki
5. Search Game - Untuk mencari game yang dimiliki
6. Cari Game Toko - Untuk mencari game di Toko
7. History - Melihat Riwayat Pembelian
8. Help - Untuk melihat list command
9. Save - Menyimpan Data
10. Exit - Mengakhiri Program
Press any key to continue...|
```

Gambar 13. Hasil output F14 - Help

F15 - Load

```
Loading berhasil...

Press any key to continue...
|
```

Gambar 14. Hasil output F15 - Load

F16 - Save

```
Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
save
Masukkan nama folder penyimpanan: database
Data berhasil tersimpan...
```

Database	
game.csv	M
kepemilikan.csv	M
riwayat.csv	M
user.csv	M

Gambar 15. Hasil output F16 - Save

F17 - Exit

```
Selamat datang di interface BNMO \(^o^*)/
=====Main Menu=====
-> Register
-> Login
-> Tambah Game
-> Ubah Game
-> Ubah Stok
-> List Game Toko
-> Buy Game
-> List Game User
-> Search My Game
-> Search Game At Store
-> Top Up
-> Riwayat
-> Help
-> Save
-> Exit
=====Mini Games=====
-> Magic Conch Shell
-> Tic Tac Toe
=====
Silahkan pilih menu yang diinginkan:
exit
Apakah anda ingin melakukan penyimpanan file?(y/n) y
Masukkan nama folder penyimpanan: database
Data berhasil tersimpan...
Terima kasih sudah bermain :D
```

Gambar 16. Hasil output F17 - Exit

B02 - Magic Conch Shell

```
Apa pertanyaanmu? Apakah saya wibu?
Mayhaps
Apa pertanyaanmu? Apakah daspro susah?
G
Apa pertanyaanmu? ah yang bener?
Mayhaps
Apa pertanyaanmu? yang bener dong, iya apa kagak?
p mkst?
Apa pertanyaanmu? lah marah
p mkst?
Apa pertanyaanmu? iya iya maaf
yntkts
Apa pertanyaanmu? PARAH
G
Apa pertanyaanmu?
ok|
```

Gambar 17. Hasil output B02 - Magic Conch Shell

B03 - Tic Tac Toe

```
Player X turn!
1 2 3
| |
--+-
| |
--+-
| |
--+-
X-Axis: 3
Y-Axis: 4
Inputnya yang benar ya sayang ya
X-Axis: 3
Y-Axis: 3
1 2 3
| |
--+-
| |
--+-
| |X
--+-
Lanjut ...|

Player O turn!
1 2 3
| |
--+-
| |
--+-
| |X
--+-
X-Axis: 2
Y-Axis: 2
1 2 3
| |
--+-
|O|
--+-
| |X
--+-
Lanjut ...|

Player O turn!
1 2 3
X| |O
--+-
|O|
--+-
|X|X
--+-
X-Axis: 3
Y-Axis: 1
1 2 3
X| |O
--+-
|O|
--+-
O|X|X
--+-
Player O Wins.
Press any key to continue...|
```

Gambar 18. Hasil output B03 - Tic Tac Toe

Lampiran : MoM Asistensi

Asistensi Pertama (14 April 2022)

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

Nomor Asistensi : 1
No. Kelompok/Kelas : 5 / K07
Tanggal asistensi : 14 April 2022

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
1	16521222 / Intan Putri Maharani P
2	16521366 / Reyhan Ghifari Tanjung
3	16521456 / Raden Dizi Assyafadi Putra
4	16521474 / Mahardika Wisnu Aji
5	
6	
Asisten pembimbing	NIM / Nama
	13519043 / Reihan Andhika Putra

Catatan Asistensi:

Rangkuman Diskusi
<ul style="list-style-type: none">• Penekanan pada bagian batasan spesifikasi:<ol style="list-style-type: none">1. Menggunakan Bahasa Python 3.8 atau 3.9, kalau bisa jangan Python 3.10. setiap anggota harus sama Python-nya2. Perhatikan modul-modul yang boleh di import3. Tidak boleh menginstall dan mengimport library dan modul eksternal selain yang diperbolehkan4. Tidak boleh menggunakan beberapa built-in function, kecuali yang memang diperbolehkan5. Fungsi bawaan yang boleh dipakai adalah yang sudah diajarkan di kelas• Penjelasan dan penekanan pada bagian deliverables:<ol style="list-style-type: none">1. Source code<ul style="list-style-type: none">- Menggunakan nama variable yang berarti- Clean, hanya berisi codingan final dan terpakai, tidak boleh ada codingan yang hanya bertujuan untuk meng-test program- Well-commented, minimal berisi input, output, dan proses dari suatu fungsi2. Laporan<ul style="list-style-type: none">- Perhatikan template dan ketentuan laporan yang tertera- Jika ada perubahan pembagian kerja, cantumkan table pembagian kerja yang terbaru- Desain kamus data untuk variable yang global saja- Desain dekomposisi algoritmik dan fungsional program mengikuti seperti yang telah diajarkan di kelas- Membuat notasi algoritmik untuk semua program yang telah ditulis. Untuk code yang belum pernah dijelaskan penulisan notasi algoritmiknya, boleh menuliskan codingan aslinya saja.• Penjelasan dan penekanan pada code of conduct:<ul style="list-style-type: none">- Perhatikan etika dalam pembuatan code- Tidak boleh plagiat dari internet maupun kelompok lain- Tidak boleh bekerja sama dengan kelompok lain• Q&A:<ol style="list-style-type: none">1. Apakah notasi algoritmik harus mengikuti seperti apa yang diajarkan di kuliah, atau boleh menggunakan versi lain? → Memakai notasi algoritmik sesuai yang diajarkan di mata kuliah Dasar Pemrograman2. Apakah jika ada algoritma yang mirip langsung dianggap sebagai plagiasi? → Keputusan ada tidaknya plagiasi mempertimbangkan berbagai aspek, tidak sekadar kesamaan algoritma saja.

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

<p>3. Apakah kita boleh membuat asumsi sendiri jika tidak ada peraturan yang jelas mengenai suatu hal, seperti mengurutkan ID pada F04 dan menambahkan ketentuan untuk password? → Boleh</p> <p>4. Apakah untuk interface boleh dikreasikan? → Boleh, asalkan tidak menggunakan library atau modul di luar batasan spesifikasi</p> <p>5. Apakah untuk output Magic Conch Shell boleh dikreasikan? → Boleh, asal tidak ada unsur SARA maupun menjatuhkan suatu pihak</p>
<p>Tindak Lanjut</p> <ul style="list-style-type: none"> • Mengoreksi kembali code yang telah dibuat • Memulai pengerjaan laporan

Gambar 19. MoM Asistensi 1

Asistensi Kedua (23 April 2022)

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

Nomor Asistensi	:	2
No. Kelompok/Kelas	:	5 / K07
Tanggal asistensi	:	23 April 2022

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
	1 16521222 / Intan Putri Maharani P
	2 16521366 / Reyhan Ghifari Tanjung
	3 16521456 / Raden Dizi Assyafadi Putra
	4 16521474 / Mahardika Wisnu Aji
	5
	6

Asisten pembimbing	NIM / Nama
	13519043 / Reihan Andhika Putra

Catatan Asistensi:

<p>Rangkuman Diskusi</p> <p>Penjelasan dan penekanan mengenai <i>deliverables</i>:</p> <ul style="list-style-type: none"> - Semua fungsional yang ada di tugas besar ini harus dijalankan sebagai satu kesatuan. Artinya, harus dibuatkan satu main program yang mengatur flow masukan pengguna dari <i>load</i> sampai <i>exit</i>. - Tidak boleh ada <i>temporary CSV</i>. CSV yang diperbolehkan hanya pada saat programnya di-load dan saat program di-save. - Di dalam program tidak boleh ada <i>dictionary</i>, <i>searching</i> - Untuk fungsionalitas <i>load</i>, diasumsikan folder yang ingin di-load ada di dalam satu <i>directory</i> - Untuk notasi algoritmik, yang wajib dibuat hanyalah fungsionalitas utama. Fungsi-fungsi pendukung hanya dibuat definisi dan spesifikasinya saja. - Perhatikan <i>deadline deliverables</i>! - Durasi maksimal video demonstrasi 15 menit, setiap anggota kelompok harus menjelaskan bagian kerjanya masing-masing.
<p>Tindak Lanjut</p> <ul style="list-style-type: none"> • Mengoreksi kembali code yang telah dibuat • Finishing laporan dan video demonstrasi

Gambar 20. MoM Asistensi 2

Daftar Pustaka

Referensi

<https://i.imgur.com/P1m8SZT.png>

<https://www.askpython.com/python/examples/in-and-not-in-operators-in-python>

<https://www.geeksforgeeks.org/bubble-sort/>

<https://www.geeksforgeeks.org/ord-function-python/>

<https://www.geeksforgeeks.org/python-string-strip/>

https://www.w3schools.com/python/python_try_except.asp

<https://www.neverstopbuilding.com/blog/minimax>

Liem, Inggiani. 2007. Draft Diktat Kuliah Dasar Pemrograman (Bagian Pemrograman Prosedural). Bandung: ITB.

Link video tubes

https://drive.google.com/drive/folders/1gn6t6cZpZ7rHC_0hTrUqhIpPABGOyTOS?usp=sharing

Link Github

<https://github.com/DiziASP/Tubes-Daspro---BNMO.git>