

TUGAS BESAR DASAR PEMROGRAMAN

Pembuatan Sistem Inventarisasi dan Toko Game pada Robot Game BNMO



Oleh :

K-12

KELOMPOK 1

Muhammad Rifky Fachrizain	(16521510)
Ahmad Nadil	(16521516)
Haikal Ardzi Shofiyyurrohman	(16521524)
Athira Dhyanissa Takfir	(16521548)
Dhafin Gholib Luqman Hakim	(16521546)

Dosen Pengampu :

Farrell Yodihartomo, S.T., M.S.Sc.

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2021 / 2022

Pernyataan Kejujuran

"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 2020/2021.

- 1. Nama** : Muhammad Rifky Fachrizain
NIM : 16521510
- 2. Nama** : Ahmad Nadil
NIM : 16521516
- 3. Nama** : Haikal Ardzi Shofiyyurrohman
NIM : 16521524
- 4. Nama** : Dhafin Gholib Luqman Hakim
NIM : 16521546
- 5. Nama** : Athira Dhyanissa Takfir
NIM : 16521548

DAFTAR ISI

Pernyataan Kejujuran	1
Daftar ISI	2
Daftar TABEL	2
DAFTAR GAMBAR	3
DESKRPSI PERSOALAN	4
DAFTAR PEMBAGIAN TUGAS	5
CHECKLIST HASIL RANCANGAN	7
DESAIN COMMAND UNTUK SETIAP PRIMITIF	8
DESAIN KAMUS DATA	11
DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL PROGRAM	17
SPESIFIKASI MODUL YANG DIBUAT	39
HASIL PENGUJIAN PROGRAM	44
LAMPIRAN	55

DAFTAR TABEL

Pembagian Tugas	5
Checklist Rancangan	7

Daftar Gambar

Menu Utama Admin	44
Menu Utama User	44
F02 - Register	44
F03 - Login	45
F04 - Menambah Game ke Toko Game	45
F05 - Mengubah Game pada Toko Game	46
F06 - Mengubah Stok Game di Toko	46
F07 - Listing Game di Toko Berdasarkan ID, Tahun Rilis, dan Harga	46
F08 - Membeli dan Menjual Game	48
F09 - Melihat Game yang Dimiliki	49
F10 - Mencari Game yang Dimiliki dari ID dan Tahun Rilis	49
F11 - Mencari Game di Toko dari ID, Nama Game, Harga, Kategori dan Tahun Rilis	50
F12 - Top Up Saldo	50
F13 - Melihat Riwayat Pembelian	51
F14 - Help	51
F15 - Load	52
F16 - Save	53
F17 - Exit	53
B01 - Cipher	53
B02 - Magic Conch Shell	53
B03 - TicTacToe	54

DESKRPSI PERSOALAN

Dalam tugas besar ini, kami sebagai mahasiswa Institut Teknologi Bandung Fakultas STEI peserta kelas 12 dasar pemrograman diberikan sebuah tugas untuk memperbaiki Sistem Inventarisasi dan Toko Game pada sebuah robot bernama BNMO. BNMO adalah sebuah robot milik Indra dan Doni, yang mereka gunakan untuk bermain game dan gacha. Doni dan Indra telah menggunakan robot ini selama mereka berkuliah, namun suatu saat Indra marah karena kalah gacha dan membanting BNMO tanpa belas kasih. BNMO pun rusak, namun Indra dan Doni tidak memiliki kemampuan coding yang mumpuni. Alhasil Doni dan Indra meminta tolong kepada kami untuk memperbaiki BNMO. Sebagai mahasiswa ITB yang telah mempelajari dasar dasar pemrograman secara menyeluruh, kami yakin bisa memperbaiki BNMO. Hal tersebut dikarenakan seluruh materi mulai dari *algorithmic and computational thinking*, pemrograman prosedural, dan pemrograman modular telah kami kuasa dengan baik.

Hal yang perlu dilakukan untuk memperbaiki BNMO adalah menyusun ulang program BNMO beserta fungsi-fungsi di dalamnya. Kami juga harus membuat database yang berisi user, nama-nama game yang terdaftar, riwayat pembelian dan kepemilikan game. Berikut kami lampirkan list program yang kami buat:

1. Register
2. Login
3. Menambah Game ke Toko Game
4. Mengubah Game pada Toko Game
5. Mengubah Stok Game di Toko
6. Listing Game di Toko Berdasarkan ID, Tahun Rilis dan Harga
7. Membeli Game
8. Melihat Game yang dimiliki
9. Mencari Game yang dimiliki dari ID dan tahun rilis
10. Mencari Game di Toko dari ID, Nama Game, Harga, Kategori dan Tahun Rilis
11. Top Up Saldo
12. Melihat Riwayat Pembelian
13. Help
14. Load
15. Save
16. Exit

Sebagai bonus, empat program berikut juga dapat dibuat:

1. Password Cipher
2. Kerang Ajaib
3. Game TicTacToe

Sehubungan dengan banyaknya program yang harus kami rancang, tugas diusahakan dibagi secara merata ke seluruh anggota kelompok. Dalam Tugas Besar kali ini, semua program akan dibuat, disimulasikan, serta dilaporkan untuk melengkapi kewajiban kami di kelas dasar pemrograman tahun 2021.

DAFTAR PEMBAGIAN TUGAS

Tabel 1. Pembagian Tugas

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F02 - Register	Function register	16521510	16521510	All
F03 - Login	Procedure login	16521510	16521510	All
F04 - Menambah Game ke Toko Game	Function tambah_game	16521510	16521510	All
F05 - Mengubah Game pada Toko Game	Function ubah_game	16521546	16521546	All
F06 - Mengubah Stok Game di Toko	Function ubah_stok	16521516	16521516	All
F07 - Listing Game di Toko Berdasarkan ID, Tahun Rilis dan Harga	Function list_game_toko	16521516	16521516	All
F08 - Membeli dan Menjual Game	Function buy_game	16521516	16521516	All
F09 - Melihat Game yang dimiliki	Procedure list_game	16521516	16521516	All
F10 - Mencari Game yang dimiliki dari ID dan tahun rilis	Procedure search_my_game	16521524	16521524	All
F11 - Mencari Game di Toko dari ID, Nama Game, Harga, Kategori dan Tahun Rilis	Procedure search_game_at_store	16521524	16521524	All

F12 - Top Up Saldo	Function topup	16521524	16521524	All
F13 - Melihat Riwayat Pembelian	Procedure riwayat	16521546	16521546	All
F14 - Help	Procedure help	16521546	16521546	All
F15 - Load	Function load	16521548	16521548	All
F16 - Save	Function save	16521548	16521548	All
F17 - Exit	Procedure exit	16521548	16521548	All
B01 - Cipher	Function Cipher, Function deCipher	16521524	16521524	All
B02 - Magic Conch Shell	Procedure kerangajaib	16521516	16521516	All
B03 - TicTacToe	Procedure TicTacToe	16521516	16521516	All

CHECKLIST HASIL RANCANGAN

Link GitHub Program : <https://github.com/IceTeaXXD/Tubes-Daspro-Kelompok-1>

Tabel 2. Checklist Rancangan

Fitur	Checklist
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 dan Menjual 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	✓
B02 - Magic Conch Shell	✓
B03 - TicTacToe	✓

DESAIN COMMAND UNTUK SETIAP PRIMITIF

F02 - Register

Nama command : register (userData,role)

Masukan : nama, username, password

Keluaran : Registrasi Berhasil!

F03 - Login

Nama command : login (userData)

Masukan : username, password

Keluaran : Halo {username} selamat datang di BNMO!

F04 - Menambah Game ke Toko Game

Nama command : tambah_game (gameData,role)

Masukan : namaGame, kategori, tahunRilis, harga, stokAwal, jumlah_game

Keluaran : Selamat! Berhasil menambahkan game {namaGame}

F05 - Mengubah Game pada Toko Game

Nama command : ubah_game (gameData, role)

Masukan : namaGame, kategori, tahunRilis, harga

Keluaran : Selamat! Berhasil mengubah game {idGame}

F06 - Mengubah Stok Game di Toko

Nama command : ubah_stok (gameData,role)

Masukan : idGame

Keluaran : Stok game {nama game} berhasil ditambahkan. Stok sekarang: {stok game sekarang}

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

Nama command : list_game_toko (gameData)

Masukan : skema

Keluaran : {Tabel sorted sesuai skema}

F08 - Membeli dan Menjual Game

Nama command :

buy_game(userData,gameData,kepemilikanData,riwayatData,userid,role,saldo)

Masukan : idGame

Keluaran : Selamat! Anda telah membeli game {nama game}

F09 - Melihat Game yang Dimiliki

Nama command : list_game (gameData, riwayatData, userid, role)

Masukan : {data}

Keluaran : {Tabel game yang dimiliki}

F10 - Mencari Game yang Dimiliki dari ID dan Tahun Rilis

Nama command : search_my_game (gameData, riwayatData, userid, role)

Masukan : idGame, tahunRilis

Keluaran : {Tabel game yang dimiliki sesuai filter yang dimasukkan}

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

Nama command : search_game_at_store (gameData)

Masukan : idGame, namaGame, hargaGame, kategoriGame, tahunRilis

Keluaran : 'Daftar game pada toko yang memenuhi kriteria :'
{Tabel game pada toko}

F12 - Top Up Saldo

Nama command : topup (userData, role)

Masukan : username

Keluaran : Top up berhasil! Saldo {username} menjadi {saldo sekarang}

F13 - Melihat Riwayat Pembelian

Nama command : riwayat (gameData, riwayatData, userid, role)

Masukan : {Data}

Keluaran : 'Riwayat Transaksi :'
{Tabel riwayat transaksi}

F14 - Help

Nama command : help (role)

Masukan : role

Keluaran : {List command sesuai role pengguna}

F15 - Load

Nama command : load

Masukan : Menerima input nama folder

Keluaran : userData, gameData, kepemilikanData, riwayatData

F16 - Save

Nama command : save (userData, gameData, kepemilikanData, riwayatData)

Masukan : Nama folder penyimpanan

Keluaran : Data csv dalam folder yang diinput

F17 - Exit

Nama command : exit

Masukan : -

Keluaran : Terima kasih telah menggunakan aplikasi ini, sampai jumpa lagi!

B01 - Cipher

Nama command : Cipher & deCipher

Masukan : Menerima argumen teks yang akan diubah bentuknya

Keluaran : Teks yang telah di enkripsi / dekripsi

B02 - Magic Conch Shell

Nama command : kerangajaib

Masukan : Menerima pertanyaan

Keluaran : Mengeluarkan jawaban random antara Ya,Tidak, Bisa Jadi, Mungkin, Tentunya, Tidak mungkin

B03 - TicTacToe

Nama command : TicTacToe

Masukan : Input pilihan letak X atau O

Keluaran : Gambar papan tictactoe dan pemenangnya jika telah selesai

DESAIN KAMUS DATA

F02 - Register

```
##### KAMUS Argument #####  
  
# userData : 2D Matrix of String  
# role : String  
  
##### Kamus Lokal #####  
  
# username : String  
# password : String  
# nama : String  
# validasi_user : Boolean  
# count : Integer  
# char : String  
# i : Integer
```

F03 - Login

```
##### KAMUS ARGUMEN #####  
  
# userData : 2D Matrix of String  
  
##### KAMUS LOKAL #####  
  
# userid,username,nama,password,role,saldo : String  
# indeks : Integer
```

F04 - Menambah Game ke Toko Game

```
##### KAMUS ARGUMEN #####  
  
# gameData : 2D Matrix of String  
# role : String  
  
##### KAMUS LOKAL #####  
  
# idGame,namaGame,kategori,tahunRilis,harga,stokAwal : String  
# jumlah_game : Integer
```

F05 - Mengubah Game pada Toko Game

```
##### KAMUS ARGUMEN #####
```

```
# gameData : 2D Matrix of String
```

```
# role : String
```

```
##### KAMUS LOKAL #####
```

```
# sudahUbah : Boolean
```

```
# idGame, namaGame, kategori, tahunRilis, harga : String
```

```
# indeks : Integer
```

F06 - Mengubah Stok Game di Toko

```
##### KAMUS ARGUMEN #####
```

```
# gameData : 2D Matrix of String
```

```
# role : String
```

```
##### KAMUS LOKAL #####
```

```
# idGame, namaGame, kategori, tahunRilis, harga, stokAwal : String
```

```
# stok, stokNow, indeks : Integer
```

```
# sudahUbah : Boolean
```

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

```
##### KAMUS ARGUMEN #####
```

```
# gameData : 2D Matrix of String
```

```
##### KAMUS LOKAL #####
```

```
# skema : String
```

```
# list_sorted : 2D Matrix of String
```

F08 - Membeli dan Menjual Game

```
##### KAMUS ARGUMENT #####
```

```
# userData : 2D Matrix of String
```

```
# gameData : 2D Matrix of String
```

```
# kepemilikanData : 2D Matrix of String
```

```
# riwayatData : 2D Matrix of String
```

```

# userid : String
# role : String
# saldo : String

##### KAMUS LOKAL #####
# saldo_ : Integer
# indeks : Integer
# found : Boolean
# idGame : String
# hargaGame : Integer
# stokGame : Integer
# namaGame : String
# tahun : String
# temp_kepemilikan : 2D Matrix of String

```

F09 - Melihat Game yang Dimiliki

```

##### KAMUS ARGUMENT #####
# gameData : 2D Matrix of String
# riwayatData : 2D Matrix of String
# userid : String
# role : String

```

F10 - Mencari Game yang Dimiliki dari ID dan Tahun Rilis

```

##### KAMUS ARGUMENT #####
# gameData : 2D Matrix of String
# riwayatData : 2D Matrix of String
# userid : String
# role : String

```

```

##### KAMUS LOKAL #####
# idGame : String
# tahunRilis : String
# arr_search : 2D Matrix of String
# arr_idGame : 2D Matrix of String

```

```
# arr_tahunRilis : 2D Matrix of String
# arr_my_game : 2D Matrix of String
```

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

```
##### KAMUS ARGUMENT #####
# gameData : 2D Matrix of String

##### KAMUS LOKAL #####
# idGame : String
# namaGame : String
# hargaGame : String
# kategoriGame : String
# tahunRilis : String
# arr_search : 2D Matrix of String
# arr_idGame : 1D Matrix of String
# arr_namaGame : 1D Matrix of String
# arr_hargaGame : 1D Matrix of String
# arr_kategoriGame : 1D Matrix of String
# arr_tahunRilis : 1D Matrix of String
```

F12 - Top Up Saldo

```
##### KAMUS ARGUMENT #####
# userData : 2D Matrix of String
# role : String

##### KAMUS LOKAL #####
# userName : String
# indeks : Integer
# saldo : Integer
```

F13 - Melihat Riwayat Pembelian

```
##### KAMUS ARGUMENT #####
# gameData : 2D Matrix of String
```

```
# riwayatData : 2D Matrix of String
# userid : String
# role : String

##### KAMUS LOKAL #####
# game_dimiliki : 2D Matrix of String
```

F14 - Help

```
##### KAMUS ARGUMENT #####
# role : String
```

F15 - Load

```
##### KAMUS LOKAL #####
# userData : 2D Matrix of String
# gameData : 2D Matrix of String
# kepemilikanData : 2D Matrix of String
# riwayatData : 2D Matrix of String
```

F16 - Save

```
##### KAMUS ARGUMENT #####
# userData : 2D Matrix of String
# gameData : 2D Matrix of String
# kepemilikanData : 2D Matrix of String
# riwayatData : 2D Matrix of String
```

```
##### KAMUS LOKAL #####
# path : String
# userData : 2D Matrix of String
# gameData : 2D Matrix of String
# kepemilikanData : 2D Matrix of String
# riwayatData : 2D Matrix of String
```

F17 - Exit


```
##### KAMUS ARGUMENT #####
```

```
# userData : 2D Matrix of String
```

```
# gameData : 2D Matrix of String
```

```
# kepemilikanData : 2D Matrix of String
```

```
# riwayatData : 2D Matrix of String
```

```
# saved : Boolean
```

B01 - Cipher

```
##### KAMUS ARGUMENT #####
```

```
# chars_arr : Array of String
```

```
##### KAMUS LOKAL #####
```

```
# a, b, m : Integer
```

```
# cipheredText, text : String
```

B02 - Magic Conch Shell

```
##### KAMUS LOKAL #####
```

```
# tanya = String
```

```
# x = Integer
```

```
# a = Integer
```

```
# c = Integer
```

```
# m = Integer
```

```
# LCG = Integer
```

```
# n = Integer
```

```
# jawaban = Array of String
```

B03 - TicTacToe

```
##### KAMUS LOKAL #####
```

```
# arr = Array of String
```

```
# turn = Integer
```

```
# pos = Integer
```

```
# char = String
```

DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL PROGRAM

F02 - Register

```
from .Function import *
from .B01_Cipher import *

function register(userdata, role : string) -> string
    if role == admin then
        output('Register')
        input(nama('Masukkan nama: '))

        validasi_user <- False
        while not (validasi_user) do
            input(username('Masykan username: '))
            count <- 0
            char : ['-','.'['_']

            repeat (lenght_of_obj(username)) times
                if username[i].isupper() or
username[i].islower() or (username[i] in char) or
username[i].isdigit() then
                    count <- count + 1

            if count == lenght_of_obj(username) then
                repeat (lenght_of_obj(username)) times
                    if username == userData[i][1] then
                        print('Username sudah terpakai,
silakan masukkan username lain.')
                        validasi_user <- False
                    else if i == lenght_of_obj(userData) - 1 then
                        validasi_user <- True
                else
                    output('Username tidak valid, silahkan ulangi
input!')

        depend-on (password)
        input(password('Masukan password: '))
```

```

        (length_of_obj(password) < 8) : output('Password
minimal terdiri atas 8 karakter, silakan ulangi input!')
        not any(char.isdigit() for char in password) :
output('Password harus memiliki minimal 1 angka, silakan ulangi
input!')
        not any(char.isupper() for char in password) :
output('Password harus memiliki minimal 1 huruf besar, silakan
ulangi input!')
        not any(char.islower() for char in password) :
output('Password harus memiliki minimal 1 huruf kecil, silakan
ulangi input!')
        ' ' in password : output('Password tidak boleh
memiliki spasi, silakan ulangi input!')
        else
            break

        password <- Cipher(password)
        data_user <- [str(length_of_obj(userData) + 1),
username, nama, password, 'user', '0']
        userData <- userData + 1
        output('Registrasi Berhasil!')
    else
        output('Maaf, anda tidak memiliki izin untuk menjalankan
perintah berikut. Mintalah ke administrator untuk melakukan hal
tersebut.')

```

F03 - Login

```

from .Function import *
from .B01_Cipher import *

function login (userData -> string) -> string
    output('Login')
    loginValid <- False
    while not loginValid do
        input(username('Masukan username: '))
        input(password('Masukan password: '))

```

```

repeat lenght_of_obj(userData) times
  if username == userData[i][1] and password ==
deCipher(userData[i][3]) then
    output('Halo ', username, 'selamat datang di
BNMO!')

    loginValid <- True
    indeks <- i
    break
  else
    output('username atau password salah,
silahkan ulangi input!')

userid <- userData[indeks][0]
nama <- userData[indeks][2]
role <- userData[indeks][4]
saldo <- userData[indeks][5]
-> userid,username,nama,password,role,saldo

```

F04 - Menambah Game ke Toko Game

```

from .Function import *

function tambah_game(gameData, rike -> string) -> string
  if role == 'admin' then
    depend-on (jumlah_game,namaGame)
    input(namaGame('Masukan nama game: '))
    input(kategori('Masukan kategori: '))
    input(tahunRilis('Masukan tahun rilis: '))
    input(harga('Masukan harga game:'))
    input(stokAwal('Masukan stok awal: '))
    jumlah_game <- hitungGame(gameData)

    jumlah_game < 9 : idGame -> 'GAME00' +
string(jumlah_game + 1)
    jumlah_game < 99 : idGame -> 'GAME0' +
string(jumlah_game + 1)
    jumlah_game >= 99 : idGame -> 'GAME' +
string(jumlah_game + 1)

```

```

        namaGame != ' ' and kategori != ' ' and tahunRilis
!= ' ' and harga != ' ' and stokAwal != ' ' : data -> [idGame,
namaGame, tahunRilis, harga, stokAwal]

        gameData -> gameData + [data]

        output(f 'Selamat! berhasil menambahkan game ',
namaGame)

    else
        output('Mohon masukan semua informasi mengenai
game agar dapat disimpan BNMO')
    else
        output('Maaf, anda tidak memiliki izin untuk menjalankan
perintah berikut. Mintalah ke administrator untuk melakukan hal
tersebut.
```

F05 - Mengubah Game pada Toko Game

```
from .Functions import*
```

```

procedure ubah_game (input gameData: 2D Matrix of Strings, input role:
String)
    if length_of_object(gameData) = 0 then
        output('Maaf, tidak ada data game yang tersimpan.')
    else
        If role = 'admin' then
            sudahUbah ← False
            While not (sudahUbah) do
                idGame ← input(idGame)
                i traversal [0..range(length_of_object(gameData))]
                if idGame = gameData[i][0] then
                    indeks ← i
                    sudahUbah = True
                    break

            if sudahUbah then
                namaGame ← input(namaGame)
                kategori ← input(kategori)
```

```

        tahunRilis ← input(tahunRilis)
        harga ← input(harga)
        depend on (namaGame, kategori, tahunRilis, harga)
            namaGame != '' : gameData[indeks][1] ← namaGame
            kategori != '' : gameData[indeks][2] ← kategori
            tahunRilis != '' : gameData[indeks][3] ←
tahunRilisinput
            harga != '' : gameData[indeks][4] ← harga
            output(f'Selamat! Berhasil mengubah game' + idGame)
            break

        else
            output('Maaf, id game tidak ditemukan. Ulangi input
idGame!')
        else
            output('Maaf, anda tidak memiliki izin untuk menjalankan perintah
berikut. Mintalah ke administrator untuk melakukan hal tersebut.')

```

F06 - Mengubah Stok Game di Toko

```

From .Functions import*
procedure ubah_stok (input gameData : 2D Matrix of String, input role :
String)
    if (length_of_obj(gameData) == 0) then
        output(Maaf, tidak ada data game yang tersimpan.)

    else then
        if (role == 'admin') then
            sudahUbah <- False
            while (sudahUbah == False) do
                input(idGame)
                i traversal [0..length_of_obj(gameData)]
                if (idGame == gameData [i][0]) then
                    indeks <- i
                    sudahUbah <- True
                    break

            if (sudahUbah == True) then
                input(stok)
                if (stok >= 0) then
                    stokNow <- gameData [indeks][5]
                    stokNow <- stokNow + stok

```

```

        gameData[indkes][5] <- stokNow
        output(Stok game {gameData[indeks][1]} berhasil
ditambahkan. Stok sekarang: {gameData[i][5]})
        break

    else if (stok < 0 and int(gameData[i][5]) >= abs(stok))
then
        stokNow <- gameData[indeks][5]
        stokNow <- stokNow + stok
        gameData[indeks][5] <- str(stokNow)
        output(Stok game {gameData[indeks][1]} berhasil
dikurangi. Stok sekarang: {gameData[i][5]})
        break
    else then
        output(Stok game {gameData[indeks][1]} gagal
dikurangi karena stok kurang. Stok sekarang: {gameData[i][5]})
        break
    else then
        output(Tidak ada game dengan ID tersebut! Ulangi input
idGame!)
    else then
        output(Maaf, anda tidak memiliki izin untuk menjalankan perintah
berikut. Mintalah ke administrator untuk melakukan hal tersebut.)

```

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

```

from .Functions import*
procedure list_game_toko (input gameData : 2D Matrix of String)
    input(skema)
    if (skema == 'tahun-') then
        list_sorted <- quicksort(gameData,3,'descending')
        bikinTabel(list_sorted,'game')
    else if (skema == 'harga-') then
        list_sorted <- quicksort(gameData,4,'descending')
        bikinTabel(list_sorted,'game')
    else if (skema == 'tahun+') then
        list_sorted <- quicksort(gameData,3,'ascending')
        bikinTabel(list_sorted,'game')
    else if (skema == 'harga+') then
        list_sorted <- quicksort(gameData,4,'ascending')
        bikinTabel(list_sorted,'game')
    else if (skema == '') then
        list_sorted <- quicksort(gameData,0,'ascending')
        bikinTabel(list_sorted,'game')
    else then
        output('Skema sorting tidak valid')
        quicksort(gameData,0,'ascending')

```

F08 - Membeli dan Menjual Game

```
from .Functions import*
procedure buy_game(input userData : 2D Matrix of String, input gameData : 2D
Matrix of String,
                    input kepemilikanData : 2D Matrix of String, input
riwayatData : 2D Matrix of String,
                    input userid : String , input role : String, input saldo :
integer)

    if (role == "user") then
        indeks <- 0
        found <- False
        input(idGame)
        i traversal [0..length_of_obj(gameData)]
            if (idGame == gameData[i][0])
                indeks <- i
                found <- True
                break

        hargaGame <- gameData[indeks][4]
        stokGame <- gameData[indeks][5]
        namaGame <- gameData[indeks][1]

        if (isOwned(kepemilikanData,idGame,userid)) then
            output('Anda sudah memiliki game tersebut!')

        else if (found == False) then
            output('Maaf, id game tidak ditemukan. Ulangi input idGame!')

        else if (found and saldo >= hargaGame and stokGame > 0) then
            gameData[indeks][5] <- gameData[indeks][5]) - 1

            saldo <- saldo - hargaGame
            i traversal [0..length_of_obj(userData)]
                if (userData[i][0] == userid) then
                    userData[i][5] <- saldo
                    break

            temp_kepemilikan <- [idGame, userid]
            kepemilikanData <- kepemilikanData + [temp_kepemilikan]

            tahun <- (date.today()).year
            riwayatData <- riwayatData +
[[idGame,namaGame,str(hargaGame),userid,tahun]]

            otuput(Selamat! Anda telah membeli game {gameData[indeks][1]})

        else then
            output('Maaf, stok game tidak mencukupi atau saldo anda tidak
mencukupi untuk membeli game tersebut!')

    else then
        otuput('Akses hanya untuk role user!')
```


F09 - Melihat Game yang Dimiliki

```
from .Functions import*

procedure list_game (input gameData : 2D Matrix of String, input riwayatData
: 2D Matrix of String
                    input userid : String, input role : String)
  if (role == 'user') then
    list_game_arr <- list_game_dimiliki(gameData, riwayatData, userid)
    if (length_of_obj(list_game_arr) /= 0) then
      bikinTabel(list_game_arr, 'owned')
    else then
      output(Anda belum memiliki game!)
  else then
    output(Akses hanya untuk role user!)
```

F10 - Mencari Game yang Dimiliki dari ID dan Tahun Rilis

```
from .Functions import*
{F10 search_my_game}
procedure search_my_game (input gameData : 2D matrix of strings,
  input riwayatData : 2D matrix of strings, input userid : 2D
  matrix of strings, input role : 2D matrix of strings)
{Mencari game yang dimiliki oleh user}
  if role = "user then
    output("Masukan id game: ")
    input(idGame)
    output("Masukan tahun rilis: ")
    input(tahunRilis)
    arr_search <- [] {empty array}
    arr_idGame <- [] {empty array}
    arr_tahunRilis <- [] {empty array}

    arr_my_game <- function list_game_dimiliki (gameData, riwayatData,
userid)
      if idGame = "" and tahunRilis = "" then
        output("Maaf, id game dan tahun rilis tidak boleh kosong!")

      if idGame ≠ "" then
        arr_idGame += [idGame]
      else
        repeat length_of_obj(gameData) times
          arr_idGame += [gameData[i][0]]

      if tahunRilis ≠ "" then
        arr_tahunRilis += [tahunRilis]
      else
        repeat length_of_obj(gameData) times
          arr_tahunRilis += [gameData[i][3]]

    repeat length_of_obj(arr_my_game) times
```

```

        if arr_my_game[i][0] in arr_idGame and arr_my_game[i][3] in
            arr_tahunRilis
            Arr_search += [arr_my_game[i]]

    if length_of_obj(arr_search) ≠ 0 then
        bikinTabel(arr_search,"owned")
    else
        output("Daftar game pada toko yang memenuhi kriteria:")
        output("Tidak ada game pada toko yang memenuhi
            kriteria")

```

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

```

from .Functions import*
{F11 search_game_at_store}
procedure search_game_at_store(input gameData : 2D matrix of strings)
{Mencari game yang tersedia pada toko}
    output("Masukan id game: ")
    input(idGame)
    output("Masukan nama game: ")
    input(namaGame)
    output("Masukan harga game: ")
    input(hargaGame)
    output("Masukan kategori game: ")
    input(kategoriGame)
    output("Masukan tahun rilis: ")
    input(tahunRilis)

    arr_search ← [] {empty array}
    arr_idGame ← [] {empty array}
    arr_namaGame ← [] {empty array}
    arr_hargaGame ← [] {empty array}
    arr_kategoriGame ← [] {empty array}
    arr_tahunRilis ← [] {empty array}
    if idGame = "" and namaGame = "" and hargaGame = "" and
        kategoriGame = "" and tahunRilis = "" then
        output('Maaf, id game, nama game, harga game, kategori game,
            dan tahun rilis tidak boleh kosong!')

    if length_of_obj(idGame) > 0 then
        arr_idGame += [idGame]
    else
        repeat length_of_obj(gameData) times
            arr_idGame += [gameData[i][0]]

    if length_of_obj(namaGame) > 0 then
        arr_namaGame += [namaGame]
    else
        repeat length_of_obj(gameData) times
            arr_namaGame += [gameData[i][1]]
    if length_of_obj(hargaGame) > 0 then

```

```

        arr_hargaGame += [hargaGame]
    else
        repeat length_of_obj(gameData) times
            arr_hargaGame += [gameData[i][4]]
    if length_of_obj(kategoriGame) > 0 then
        arr_kategoriGame += [kategoriGame]
    else
        repeat length_of_obj(gameData) times
            arr_kategoriGame += [gameData[i][2]]

    if length_of_obj(tahunRilis) > 0 then
        arr_tahunRilis += [tahunRilis]
    else
        repeat length_of_obj(gameData) times
            arr_tahunRilis += [gameData[i][3]]

    repeat length_of_obj(gameData) times
        if gameData[i][0] in arr_idGame and gameData[i][1] in
            arr_namaGame and gameData[i][4] in arr_hargaGame
            and gameData[i][2] in arr_kategoriGame and
            gameData[i][3] in arr_tahunRilis

            arr_search += [gameData[i]]

    if length_of_obj(arr_search) ≠ 0 then
        output("Daftar game pada toko yang memenuhi kriteria: ")
        bikinTabel(arr_search, "game")

    else:
        output("Daftar game pada toko yang memenuhi kriteria: ")
        output("Tidak ada game pada toko yang memenuhi
kriteria")

```

F12 - Top Up Saldo

```

from .Functions import*
{F12 Topup}
procedure topup (input userData : 2D matrix of strings, input role
: 2D matrix of strings)
{Mentopup sejumlah uang yang diinginkan user ke saldo user}
    if role = "admin" then
        output("Masukan username : ")
        input(userName)
        repeat length_of_obj(userData) times
            if userData [i][1] = userName then
                indeks ← i
                stop
        if indeks ≠ 0 then

```

```

        output("Masukan jumlah saldo yang ingin
ditambahkan:")
        input(saldo : integer)
        if saldo + integer(userData[i][5]) >= 0 then
            userData[indeks][5]←
                string(integer(userData[indeks][5])
                +saldo)
            output(f"Top up berhasil! Saldo
            {username}
            menjadi {userData[indeks][5]}")
        else
            output("Maaf, jumlah saldo tidak
            valid!")
        else
            output("Maaf, username tidak ditemukan!")
    else
        output("Akses hanya untuk role admin")

```

F13 - Melihat Riwayat Pembelian

```

from .Functions import*

```

```

procedure riwayat(input gameData: 2D Matrix of Strings, input riwayatData: 2D
Matrix of String, input userid: 2D Matrix of String, input role: String)
    if role = 'user' then
        game_dimiliki ← list_game_dimiliki(gameData,riwayatData,userid)

        if length_of_obj(game_dimiliki) != 0 then
            output('Riwayat transaksi:')
            bikinTabel(game_dimiliki,'riwayat')

        else
            output('Riwayat transaksi:')
            output('Maaf, kamu tidak ada riwayat pembelian game. Ketik
perintah beli_game untuk membeli.')

        else
            output('Akses hanya untuk role user!')

```

F14 - Help

```

from .Functions import*

```

```

procedure help(input role: String)
    output('=====MENU=====')
    depend on (role)

```

```

role = 'user':
    output('1. Login')
    output('2. Listing Game Toko')
    output('3. Beli Game')
    output('4. Lihat Game Saya')
    output('5. Cari Game Saya')
    output('6. Cari Game Toko')
    output('7. Riwayat Pembelian')
    output('8. Help')
    output('9. Save')
    output('10. Exit')
    output('=====BONUS=====')
    output('11. Magic Conch Shell')
    output('12. Tic Tac Toe')

role = 'admin' :
    output('1. Register')
    output('2. Login')
    output('3. Tambah Game')
    output('4. Ubah Game')
    output('5. Ubah Stok')
    output('6. Listing Game Toko')
    output('7. Cari Game Toko')
    output('8. Top Up')
    output('9. Help')
    output('10. Save')
    output('11. Exit')
    output('=====BONUS=====')
    output('12. Magic Conch Shell')
    output('13. Tic Tac Toe')

else
    output('1. Login')
    output('2. Help')

```

F15 - Load

```

from .Functions import *

procedure load(input namaFolder : string)
namaFolder ← (namaFolder)
depend on (namaFolder)

```

```

if namaFolder = user then
  rewrite userData to csv_to_arr
  open userData
  break

if namaFolder = game then
  rewrite gameData to csv_to_arr
  open gameData
  break

if namaFolder = kepemilikanData then
  rewrite kepemilikanData to csv_to_arr
  open kepemilikanData
  break

if namaFolder = riwayatData
  rewrite riwayatData to csv_to_arr
  open riwayatData
  break

```

F16 - Save

```

from .Functions import *

procedure save( input userData:2D matrix of strings, input gameData: 2D
matrix of strings, input riwayatData: 2D matrix of strings, input
namaFolder:string)
if path = "./" then
  namaFolder ← (namaFolder)
  open namaFolder
  rewrite userData to ['id','username','nama','password','role','saldo']] +
userData
  rewrite gameData to ['id','nama','kategori','tahun_rilis','harga','stok']] +
gameData

  rewrite kepemilikanData to[['game_id','user_id']] + kepemilikanData
  rewrite riwayatData to['game_id','nama','harga','user_id','tahun_beli']] +
riwayatData
  saving(userData)
  saving(gameData)
  saving(kepemilikanData)
  saving(riwayatData)

```

F17 - Exit

```

from .Functions import *

procedure exit ()
output ('Terimakasih telah menggunakan aplikasi ini')
output ('Sampai jumpa lagi')

```

B01 - Cipher

```
from .Functions import*

chars_arr <- arr of nums and alphabet

function searchIndex(input char : String)
  i traversal [0..length_of_obj(chars_arr)]
    if (chars_arr[i] == char)
      -> (i+1)

function Cipher (input text : String)
  cipheredText <- ''
  a <- 5
  b <- 1
  m <- length_of_obj(chars_arr)
  i traversal [0..length_of_obj(text)]
    cipheredText <- cipheredText + chars_arr[(searchIndex(text[i]) * a +
b) % m -1]
  -> (cipheredText)

function deCipher (input text : String)
  cipheredText = ""
  a <- inv(5,length_of_obj(chars_arr))
  b <- 1
  m <- length_of_obj(chars_arr)
  i traversal [0..(length_of_obj(text))]
    cipheredText = cipheredText +
chars_arr[round(a*((searchIndex(text[i])-b))%m-1)]
  -> (cipheredText)

function inv (input a : integer, input m : integer)
  y <- 1
  while ((a*y) % m /= 1) do
    y <- y + 1
  -> (y)
```

B02 - Magic Conch Shell

```
from .Functions import*

procedure kerangajaib
  while True do
    input (tanya)
    if tanya == '':
      break
    else:
      x <- int(time.time())
      a <- x % 10
      c <- int(time.process_time_ns())
      m <- 19
      LCG <- ((a*x)+c) % m
      n <- abs((LCG % 6))

      jawaban <- ['Ya', 'Tidak', 'Bisa Jadi', 'Mungkin', 'Tentunya', 'Tidak
mungkin']
```

```
output(jawaban[n])
```

B03 - TicTacToe

```
from .Functions import*
```

```
proceudre gambarPapan(input arr : Array of String)
    output(arr[0] + '|' + arr[1] + '|' + arr[2])
    output(kaliString('-',3) + '+' + kaliString('-',3) + '+' +
kaliString('-',3))
    output(arr[3] + '|' + arr[4] + '|' + arr[5])
    output(kaliString('-',3) + '+' + kaliString('-',3) + '+' +
kaliString('-',3))
    output(arr[6] + '|' + arr[7] + '|' + arr[8])
```

```
procedure TicTacToe
    arr<- [' ' i traversal [0..9]]
    turn <- 0
    while True do
        clear()
        gambarPapan(arr)
        turn = turn + 1

        if (arr[0] == arr[1] == arr[2] != ' ') then
            output('Pemenangnya adalah ' + arr[0])
            break
        else if (arr[3] == arr[4] == arr[5] != ' ') then
            output('Pemenangnya adalah ' + arr[3])
            break
        else if (arr[6] == arr[7] == arr[8] != ' ') then
            output('Pemenangnya adalah ' + arr[6])
            break
        else if (arr[0] == arr[3] == arr[6] != ' ') then
            output('Pemenangnya adalah ' + arr[0])
            break
        else if (arr[1] == arr[4] == arr[7] != ' ') then
            output('Pemenangnya adalah ' + arr[1])
            break
        else if (arr[2] == arr[5] == arr[8] != ' ') then
            output('Pemenangnya adalah ' + arr[2])
            break
```



```

else if (arr[0] == arr[4] == arr[8] != ' ') then
    output('Pemenangnya adalah ' + arr[0])
    break
else if (arr[2] == arr[4] == arr[6] != ' ') then
    output('Pemenangnya adalah ' + arr[2])
    break
else if (turn == 10) then
    output('Tidak ada pemenang')
    break

if (turn % 2 == 0) then
    output('Giliran ' + 'X')
    char = ' X '
else then
    output('Giliran ' + 'O')
    char = ' O '

while True do
    try:
        input(pos)
        if (arr[pos-1] == ' ') then
            arr[pos-1] <- char
            break
        else if (pos == 1 or pos == 2 or pos == 3 or pos == 4 or pos
== 5 or pos == 6 or pos == 7 or pos == 8 or pos == 9) then
            output('Posisi sudah terisi')
        else then
            output('Input tidak valid!')
    except:
        output('Input tidak valid')

```

Function Tambahan

```

{ Melakukan Import Library }
import os
import argparse
import sys
import math
import time
from datetime import date

function csv_to_arr
    (input folder : String, input filename : String)
    (output new_arr : 2D Matriks of String)

```

```

{I.S. Menerima input nama folder dan nama file}
{F.S. Menghasilkan bentuk Matriks 2D of String dari sebuah file csv}

{Melakukan inisiasi variabel kolom sesuai dengan jenis file}
if (filename == 'user' or filename == 'game') then
    kolom <- 6
else if (filename == 'kepemilikan') then
    kolom <- 2
else if (filename == 'riwayat') then
    kolom <- 5

arr <- []
count <- 0
open(./folder/filename.csv , csv)

{Looping untuk melakukan parsing file csv menjadi sebuah array}
row traversal csv
    parser <- ';'
    tmp <- ''
    if (count != 0) then
        c traversal row
            if (c == parser or c == '\n') then
                arr <- arr + [tmp]
                tmp <- ''
            else then
                tmp <- tmp + c
        count <- count + 1

    new_arr <- [[' traversal [0..kolom]] traversal
[0..(length_of_obj(arr)//kolom)]]
    count <- 0

{Looping untuk mengubah array yang terbentuk menjadi sebuah matriks
sesuai jumlah baris pada csv}
i traversal [0..length_of_obj(arr)//kolom]
    j traversal [0..kolom]
        new_arr[i][j] <- arr[count]
        count <- count + 1

    if (count == length_of_obj(arr)) then
        break

-> (new_arr) {return value}

function length_of_obj
    (input obj : (String or Array or 2D Matriks of String))
    (output counter : integer)

{I.S. Menerima input sebuah matriks atau string atau array}
{F.S. Menghasilkan panjang dari inputan dalam integer}

counter <- 0 {inisiasi variabel}

{Looping i terhadap seluruh karakter pada obj}
i traversal obj
    counter <- counter + 1

```

```

-> (counter) {return value}

procedure saving
  (input data : 2D Matriks of String, input namaFolder : String, input
  fileName : String)

  {I.S. Menerima input data, nama folder, serta nama file}
  {F.S. Melakukan writing data pada sebuah file .csv}

  open('./namaFolder/filename.csv', f)

  i traversal [0..length_of_obj(data)]
    j traversal [0..length_of_obj(data[i])]
      write(f, data[i][j])
      if (j == length_of_obj(data[i])-1) then
        write(f, '\n')
      else then
        write(f, ';')

function quicksort
  (input arr : 2D Matrix of String, input indeks : integer, input sortType
  : String)
  (output arr : 2D Matrikx of String)

  {I.S. Menerima input sebuah array, indeks, dan jenis sorting}
  {F.S. Menghasilkan array yang sudah dilakukan sorting}

  if (indeks == 0) then
    i traversal [0..length_of_obj(arr)]
      j traversal [i+1..length_of_obj(arr)]
        if (arr[i][indeks] > arr[j][indeks])
          arr[i], arr[j] <- arr[j], arr[i]

  else if (sortType == 'ascending') then
    i traversal [0..length_of_obj(arr)]
      j traversal [i+1..length_of_obj(arr)]
        if int(arr[i][indeks]) > int(arr[j][indeks]) then
          arr[i], arr[j] <- arr[j], arr[i]

  else if (sortType == 'descending') then
    i traversal [0..length_of_obj(arr)]
      j traversal [i+1..length_of_obj(arr)]
        if int(arr[i][indeks]) < int(arr[j][indeks]) then
          arr[i], arr[j] <- arr[j], arr[i]

  -> (arr) {return value}

function kaliString
  (input string : String, input kali : integer)
  (output hasil : string)

  {I.S. Menerima input sebuah string dan indeks perkalian}
  {F.S. Menghasilkan string sebanyak indeks perkalian}

  hasil <- ''
  i traversal [0..kali]

```

```

    hasil <- hasil + string

-> (hasil) {return value}

procedure bikinTabel
  (input list : 2D Matrix of String, input jenisTabel : String)

  {I.S. Menerima input sebuah 2D Matrix of String dan jenis nya}
  {F.S. Menghasilkan output sebuah tabel}

  if (jenisTabel == 'game') then
    kolom0 <- ' ID Game '
    kolom1 <- ' Nama Game '
    kolom2 <- ' Kategori '
    kolom3 <- ' Tahun Rilis '
    kolom4 <- ' Harga '
    kolom5 <- ' Stok '

    longest0 <- length_of_obj(kolom0)
    longest1 <- length_of_obj(kolom1)
    longest2 <- length_of_obj(kolom2)
    longest3 <- length_of_obj(kolom3)
    longest4 <- length_of_obj(kolom4)
    longest5 <- length_of_obj(kolom5)

    item traversal list
      if (length_of_obj(str(item[0])) > longest0) then
        longest0 <- length_of_obj(str(item[0]))

    item traversal list
      if (length_of_obj(str(item[1])) > longest1) then
        longest1 <- length_of_obj(str(item[1]))

    item traversal list
      if (length_of_obj(str(item[2])) > longest2) then
        longest2 <- length_of_obj(str(item[2]))

    item traversal list
      if (length_of_obj(str(item[3])) > longest3) then
        longest3 <- length_of_obj(str(item[3]))

    item traversal list
      if (length_of_obj(str(item[4])) > longest4) then
        longest4 <- length_of_obj(str(item[4]))

    item traversal list
      if (length_of_obj(str(item[5])) > longest5) then
        longest5 <- length_of_obj(str(item[5]))

  output (kaliString('=', longest0+longest1+longest2+longest3+longest4+longest5+2
5))

  output ('|', kolom0, kaliString('
', (longest0-length_of_obj(kolom0))), '|', kolom1, kaliString('
', (longest1-length_of_obj(kolom1)+1))+ '|', kolom2, kaliString('
', (longest2-length_of_obj(kolom2)+1))+ '|', kolom3, kaliString('
', (longest3-length_of_obj(kolom3)+1))+ '|', kolom4, kaliString('

```

```
' , (longest4-length_of_obj(kolom4)+1))+'|' , kolom5, kaliString('
', (longest5-length_of_obj(kolom5)+1))+'|')
```

```
output(kaliString('=' , longest0+longest1+longest2+longest3+longest4+longest5+2
5))
```

```
    item traversal list
        output('|' , item[0], kaliString('
', (longest0-length_of_obj(str(item[0])))) , '|' , item[1], kaliString('
', (longest1-length_of_obj(str(item[1])))) , '|' , item[2], kaliString('
', (longest2-length_of_obj(str(item[2])))) , '|' , item[3], kaliString('
', (longest3-length_of_obj(str(item[3])))) , '|' , item[4], kaliString('
', (longest4-length_of_obj(str(str(item[4])))) , '|' , item[5], kaliString('
', (longest5-length_of_obj(str(item[5])))) , '|')
```

```
output(kaliString('=' , longest0+longest1+longest2+longest3+longest4+longest5+2
5))
```

```
else if (jenisTabel == 'owned') then
```

```
    kolom0 <- ' ID Game '
    kolom1 <- ' Nama Game '
    kolom2 <- ' Kategori '
    kolom3 <- ' Tahun Rilis '
    kolom4 <- ' Harga '
```

```
    longest0 <- length_of_obj(kolom0)
    longest1 <- length_of_obj(kolom1)
    longest2 <- length_of_obj(kolom2)
    longest3 <- length_of_obj(kolom3)
    longest4 <- length_of_obj(kolom4)
```

```
    item traversal list
```

```
        if length_of_obj(str(item[0])) > longest0 then
            longest0 <- length_of_obj(str(item[0]))
```

```
    item traversal list
```

```
        if (length_of_obj(str(item[1])) > longest1) then
            longest1 <- length_of_obj(str(item[1]))
```

```
    item traversal list
```

```
        if (length_of_obj(str(item[2])) > longest2) then
            longest2 <- length_of_obj(str(item[2]))
```

```
    item traversal list
```

```
        if (length_of_obj(str(item[3])) > longest3) then
            longest3 <- length_of_obj(str(item[3]))
```

```
    item traversal list
```

```
        if (length_of_obj(str(item[4])) > longest4) then
            longest4 <- length_of_obj(str(item[4]))
```

```
output(kaliString('=' , longest0+longest1+longest2+longest3+longest4+25))
```

```
    output('|' , kolom0, kaliString('
', (longest0-length_of_obj(kolom0))) , '|' , kolom1, kaliString('
', (longest1-length_of_obj(kolom1)+1))+'|' , kolom2, kaliString('
', (longest2-length_of_obj(kolom2)+1))+'|' , kolom3, kaliString('
', (longest3-length_of_obj(kolom3)+1))+'|' , kolom4, kaliString('
```

```

', (longest4-length_of_obj(kolom4)+1))+'|')

output(kaliString('=', longest0+longest1+longest2+longest3+longest4+25))
  item traversal list
    output('|', item[0], kaliString('
', (longest0-length_of_obj(str(item[0]))), '|', item[1], kaliString('
', (longest1-length_of_obj(str(item[1]))), '|', item[2], kaliString('
', (longest2-length_of_obj(str(item[2]))), '|', item[3], kaliString('
', (longest3-length_of_obj(str(item[3]))), '|', item[4], kaliString('
', (longest4-length_of_obj(str(item[4]))), '|')

output(kaliString('=', longest0+longest1+longest2+longest3+longest4+25))

else if (jenisTabel == 'riwayat') then
  kolom0 <- ' ID Game '
  kolom1 <- ' Nama Game '
  kolom2 <- ' Harga Game '
  kolom3 <- ' Tahun Beli '

  longest0 <- length_of_obj(kolom0)
  longest1 <- length_of_obj(kolom1)
  longest2 <- length_of_obj(kolom2)
  longest3 <- length_of_obj(kolom3)

  item traversal list
    if (length_of_obj(str(item[0])) > longest0) then
      longest0 <- length_of_obj(str(item[0]))

  item traversal list
    if (length_of_obj(str(item[1])) > longest1) then
      longest1 <- length_of_obj(str(item[1]))

  item traversal list
    if (length_of_obj(str(item[4])) > longest2) then
      longest2 <- length_of_obj(str(item[2]))

  item traversal list
    if (length_of_obj(str(item[5])) > longest3) then
      longest3 <- length_of_obj(str(item[5]))

  output(kaliString('=', longest0+longest1+longest2+longest3+17))
  output('|', kolom0, kaliString('
', (longest0-length_of_obj(kolom0)), '|', kolom1, kaliString('
', (longest1-length_of_obj(kolom1)+1))+'|', kolom2, kaliString('
', (longest2-length_of_obj(kolom2)+1))+'|', kolom3, kaliString('
', (longest3-length_of_obj(kolom3)+1))+'|')
  output(kaliString('=', longest0+longest1+longest2+longest3+17))

  item traversal list
    output('|', item[0], kaliString('
', (longest0-length_of_obj(str(item[0]))), '|', item[1], kaliString('
', (longest1-length_of_obj(str(item[1]))), '|', item[4], kaliString('
', (longest2-length_of_obj(item[4])), '|', item[5], kaliString('
', (longest3-length_of_obj(str(item[5]))), '|')
    output(kaliString('=', longest0+longest1+longest2+longest3+17))

function hitungGame

```

```

(input gameData : 2D Matrix of String)
(output jumlahGame : integer)

{I.S. Menerima input sebuah data game}
{F.S. Menghasilkan output jumlah game yang ada}

jumlahGame <- 0

i traversal [0..length_of_obj(gameData)]
    jumlahGame <- jumlahGame + 1

-> (jumlahGame)

function isOwned
    (input kepemilikanData : 2D Matrix of String, input idGame : String,
input userid : string) -> Boolean
    i traversal [0..length_of_obj(kepemilikan)]
        if (kepemilikanData[i][0] == idGame and kepemilikanData[i][1] ==
userid) then
            -> (True)
        -> (False)

function list_game_dimiliki
    (input gameData : 2D Matriks of String, input riwayatData : 2D Matrix of
String, input userid : String)
    (output list_game_dimiliki : 2D Matriks of String)

    {I.S. Menerima data game, riwayat, serta userid}
    {F.S. Menghasilkan data game yang dimiliki}

    list_game_dimiliki <- []
    i traversal [0..length_of_obj(riwayatData)]
        if (riwayatData[i][3] == userid) then
            idGame <- riwayatData[i][0]
            j traversal [0..length_of_obj(gameData)]
                if (idGame == gameData[j][0]) then
                    namaGame <- gameData[j][1]
                    kategoriGame <- gameData[j][2]
                    tahunRilis <- gameData[j][3]
                    hargaGame <- gameData[j][4]
                    tahunBeli <- riwayatData[i][4]
                    list_game_dimiliki <- list_game_dimiliki + [[idGame, namaGame,
kategoriGame, tahunRilis, hargaGame, tahunBeli]]
            -> (list_game_dimiliki)

function clear
    os.system('cls')

function batas
    output('=====')

```

SPESIFIKASI MODUL YANG DIBUAT

F02 - Register

Modul untuk melakukan registrasi user baru.

I.S. : Menerima argumen userData sebagai data seluruh user dan role untuk melakukan validasi apakah user tersebut admin. Lalu, akan diterima input data user baru.

F.S. : Data user baru akan ditambahkan ke dalam database seluruh user

F03 - Login

Modul untuk melakukan login user

I.S. : Menerima argumen userData sebagai data seluruh user. Akan diminta input username dan password.

F.S. : Jika username dan password tervalidasi, user akan berhasil login.

F04 - Menambah Game ke Toko Game

Modul untuk melakukan penambahan data game pada database

I.S. : Menerima argumen database game dan role, lalu akan menerima input data game baru.

F.S. : Data game baru akan digabungkan ke dalam database game.

F05 - Mengubah Game pada Toko Game

Modul untuk melakukan penambahan data game pada database

I.S. : Menerima argumen database game dan role, lalu akan menerima input data game.

F.S. : Data game yang sesuai dengan masukan idGame akan berhasil diubah.

F06 - Mengubah Stok Game di Toko

Modul untuk melakukan pengubahan stok game pada database game yang ada

I.S. : Menerima argumen database game dan role, lalu akan menerima input idGame dan stok game baru.

F.S. : Stok game akan bertambah / berkurang sesuai input.

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

Modul untuk melakukan pencarian game berdasarkan kategori dan disajikan dalam bentuk tabel

I.S. : Menerima argumen gameData dan skema sorting

F.S. : Output berupa tabel gameData yang sesuai dengan skema sorting

F08 - Membeli dan Menjual Game

Modul untuk melakukan pembelian game

I.S. : Menerima argumen database dan role. Akan menerima input idGame untuk dibeli.

F.S. : Penambahan game baru ke dalam database riwayat serta kepemilikan, lalu saldo user akan dikurangi sesuai harga game.

F09 - Melihat Game yang Dimiliki

Modul untuk melakukan pencarian game berdasarkan kategori dan disajikan dalam bentuk tabel

I.S. : Menerima argumen database

F.S. : Output berupa tabel dari list game yang hanya dimiliki oleh user

F10 - Mencari Game yang Dimiliki dari ID dan Tahun Rilis

Modul untuk melakukan pencarian game yang dimiliki berdasarkan id dan tahun dan disajikan dalam bentuk tabel.

I.S. : Menerima argumen database, lalu menerima input parameter pencarian yaitu idGame dan tahunRilis

F.S. : Output berupa tabel dari list game yang dimiliki sesuai kriteria pencarian.

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

Modul untuk melakukan pencarian game pada toko sesuai parameter yang diberikan

I.S. : Menerima argumen database, lalu akan menerima input parameter untuk melakukan filter pencarian.

F.S. : Output berupa tabel dari game pada toko sesuai kriteria pencarian

F12 - Top Up Saldo

Modul untuk melakukan top up / penambahan saldo pada user

I.S. : Menerima argumen userData dan role. Lalu, input username dan input nominal saldo yang ingin ditambahkan.

F.S. : Saldo user pada database user berhasil ditambahkan

F13 - Melihat Riwayat Pembelian

Modul untuk mencari riwayat pembelian game

I.S. : Menerima argumen database.

F.S. : Output berupa tabel dari riwayat pembelian game

F14 - Help

Modul untuk memberi penjelasan atas menu yang ada

I.S. : Menerima argumen role

F.S. : Penjelasan mengenai setiap command yang ada sesuai role

F15 - Load

Modul untuk melakukan load data dari csv menjadi sebuah array matrix

I.S. : Menerima argumen nama folder yang berisi database game dalam format csv

F.S. : Database game dalam bentuk array matrix

F16 - Save

Modul untuk menyimpan data ke dalam folder csv

I.S. : Menerima argumen database dan menerima input naam folder penyimpanan

F.S. : Database game yang semula berbentuk array matrix akan diubah menjadi data csv

F17 - Exit

Modul untuk keluar dari program

I.S. : Menerima input konfirmasi keluar

F.S. : Program diterminasi

B01 - Cipher

Modul bonus untuk melakukan enkripsi dan dekripsi kata sandi. Diimplementasikan pada fitur register.

I.S. : Menerima sebuah string yaitu password user.

F.S. : Password user akan dienkripsi menggunakan metode Affine Cipher.

B02 - Magic Conch Shell

Modul bonus randomizer jawaban sebuah pertanyaan

I.S. : Menerima input pertanyaan

F.S. : Output berupa jawaban pertanyaan yang di-random

B03 - TicTacToe

Modul bonus game tic tac toe

I.S. : Menerima input posisi user untuk permainan

F.S. : Papan akan digambar sesuai input dan jika telah memenuhi kondisi akan ditemukan pemenang dari game.

Fungsi Tambahan

csv_to_arr

Fungsi untuk merubah data csv menjadi sebuah array

I.S. : Menerima input path file dari csv

F.S. : Output berupa array matrix dari data csv

Length_of_obj

Fungsi untuk melakukan perhitungan banyaknya char dari sebuah elemen. Merupakan implementasi dari fungsi len()

I.S. : Menerima argumen sebuah elemen
(string/integer/array/matrix)

F.S. : Output berupa integer panjang dari elemen

Saving

Fungsi untuk melakukan write sebuah data ke dalam file csv

I.S. : Menerima argumen sebuah data, nama folder, dan nama file

F.S. : Sebuah file csv yang berisi array yang telah diubah

Quicksort

Fungsi untuk melakukan sorting sebuah array matrix berdasarkan sebuah parameter

I.S. : Menerima sebuah array matrix dan parameter sorting

F.S. : Array yang telah di-sort sesuai parameter yang diinput.

kaliString

Fungsi melakukan perkalian string.

I.S. : Menerima sebuah string dan jumlah perkalian

F.S. : String sepanjang jumlah perkalian

bikinTabel

Fungsi untuk membuat sebuah tabel

I.S. : Menerima argumen sebuah database

F.S. : Tabel dari database tersebut

hitungGame

Fungsi untuk menghitung banyak game yang ada pada database

I.S. : Menerima argumen database game

F.S. : Jumlah game yang ada pada database dalam bentuk integer

isOwned

Fungsi mencari apakah sebuah game telah dimiliki atau tidak

I.S. : Menerima argumen database game dan id dari game

F.S. : data Boolean tentang kepemilikan sebuah game

List_game_dimiliki

Fungsi mengetahui game apa yang telah dimiliki oleh user

I.S. : Database game

F.S. : Array data game yang dimiliki oleh user

Clear

Fungsi melakukan clearing dari terminal

Batas

Fungsi sebagai pembatas

HASIL PENGUJIAN PROGRAM

Menu Utama Admin

```
=====MENU=====
1. Register
2. Login
3. Tambah Game
4. Ubah Game
5. Ubah Stok
6. Listing Game Toko
7. Cari Game Toko
8. Top Up
9. Help
10. Save
11. Exit
=====BONUS=====
12. Magic Conch Shell
13. Tic Tac Toe
=====
Pilih nomor menu: █
```

Menu Utama User

```
=====MENU=====
1. Login
2. Listing Game Toko
3. Beli Game
4. Lihat Game Saya
5. Cari Game Saya
6. Cari Game Toko
7. Riwayat Pembelian
8. Help
9. Save
10. Exit
=====BONUS=====
11. Magic Conch Shell
12. Tic Tac Toe
=====
Pilih nomor menu: █
```

F02 - Register

Tampilan jika username telah dipakai

```
Masukan username: 16521516
Username sudah terpakai, silakan masukkan username lain.
█
```

Tampilan jika password tidak sesuai syarat

```
Masukan password: password
Password harus memiliki minimal 1 angka, silakan ulangi input!
```

F03 - Login

Tampilan login user

```
Login
Masukan username: admin
Masukan password: admin
```

Tampilan login user jika salah

```
Login
Masukan username: admin
Masukan password: admi
Username atau password salah, silakan ulangi input!
Masukan username:
```

Tampilan login user jika benar

```
Masukan username: admin
Masukan password: admin
Halo admin selamat datang di BNMO!
=====
Press Enter To Continue >>>
```

F04 - Menambah Game ke Toko Game

Tampilan jika data masukan kosong

```
Masukan nama game:
Masukan kategori:
Masukan tahun rilis:
Masukan harga game:
Masukan stok awal:
Mohon masukkan semua informasi mengenai game agar dapat disimpan BNMO.
```

Tampilan jika data masukan valid

```
Masukan nama game: Valorant
Masukan kategori: Action
Masukan tahun rilis: 2020
Masukan harga game: 0
Masukan stok awal: 50
Selamat! Berhasil menambahkan game Valorant
=====
Press Enter To Continue >>>
```

F05 - Mengubah Game pada Toko Game

Tampilan jika data id game tidak ditemukan

```
Masukan id game yang akan diubah: GAME0100
Maaf, id game tidak ditemukan. Ulangi input idGame!
Masukan id game yang akan diubah: █
```

Tampilan jika berhasil

```
Masukan id game yang akan diubah: GAME010
Masukan nama game: Valorant : Global Shooter
Masukan kategori:
Masukan tahun rilis:
Masukan harga game:
Selamat! Berhasil mengubah game GAME010
=====
Press Enter To Continue >>> █
```

F06 - Mengubah Stok Game di Toko

Tampilan Salah

```
Masukan id game yang akan diubah: GAME0199
Tidak ada game dengan ID tersebut! Ulangi input idGame!
Masukan id game yang akan diubah: █
```

```
Masukan id game yang akan diubah: GAME010
Masukan stok baru: -1000
Stok game VALORANT gagal dikurangi karena stok kurang. Stok sekarang: 100
=====
Press Enter To Continue >>> █
```

Tampilan Berhasil

```
Masukan id game yang akan diubah: GAME010
Masukan stok baru: 100
Stok game VALORANT berhasil ditambahkan. Stok sekarang: 150
=====
Press Enter To Continue >>> █
```

```
Masukan id game yang akan diubah: GAME010
Masukan stok baru: -50
Stok game VALORANT berhasil dikurangi. Stok sekarang: 100
=====
Press Enter To Continue >>> █
```

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

Tampilan jika skema sorting salah

Masukan skema pencarian (tahun-/harga-/tahun+/harga+) : skema+
Skema sorting tidak valid

Press Enter To Continue >>>

Tampilan jika skema sorting benar

Masukan skema pencarian (tahun-/harga-/tahun+/harga+) : tahun-

ID Game	Nama Game	Kategori	Tahun Rilis	Harga	Stok
GAME001	Elden Ring	Adventure	2022	700000	8
GAME007	Fisika Dasar	Horror	2022	10000000	8
GAME009	Dasar Pemrograman : Semester 2	Education	2022	10	548
GAME002	Valhalla	Adventure	2021	40000	6
GAME006	FIFA 22	Sports	2021	250000	9
GAME003	COD MW	Action	2020	500000	8
GAME010	VALORANT	Action	2020	0	100
GAME008	Read Dead Redemption 2	Action	2018	300000	58
GAME005	GTA V	Simulation	2015	150000	8
GAME004	Zelda	Classic	2002	100000	9

Press Enter To Continue >>>

Masukan skema pencarian (tahun-/harga-/tahun+/harga+) : harga-

ID Game	Nama Game	Kategori	Tahun Rilis	Harga	Stok
GAME007	Fisika Dasar	Horror	2022	10000000	8
GAME001	Elden Ring	Adventure	2022	700000	8
GAME003	COD MW	Action	2020	500000	8
GAME008	Read Dead Redemption 2	Action	2018	300000	58
GAME006	FIFA 22	Sports	2021	250000	9
GAME005	GTA V	Simulation	2015	150000	8
GAME004	Zelda	Classic	2002	100000	9
GAME002	Valhalla	Adventure	2021	40000	6
GAME009	Dasar Pemrograman : Semester 2	Education	2022	10	548
GAME010	VALORANT	Action	2020	0	100

Press Enter To Continue >>>

Masukan skema pencarian (tahun-/harga-/tahun+/harga+) : tahun+

ID Game	Nama Game	Kategori	Tahun Rilis	Harga	Stok
GAME004	Zelda	Classic	2002	100000	9
GAME005	GTAV	Simulation	2015	150000	8
GAME008	Read Dead Redemption 2	Action	2018	300000	58
GAME003	COD MW	Action	2020	500000	8
GAME010	VALORANT	Action	2020	0	100
GAME002	Valhalla	Adventure	2021	40000	6
GAME006	FIFA 22	Sports	2021	250000	9
GAME001	Elden Ring	Adventure	2022	700000	8
GAME009	Dasar Pemrograman : Semester 2	Education	2022	10	548
GAME007	Fisika Dasar	Horror	2022	1000000	8

Press Enter To Continue >>>

Masukan skema pencarian (tahun-/harga-/tahun+/harga+) : harga+

ID Game	Nama Game	Kategori	Tahun Rilis	Harga	Stok
GAME010	VALORANT	Action	2020	0	100
GAME009	Dasar Pemrograman : Semester 2	Education	2022	10	548
GAME002	Valhalla	Adventure	2021	40000	6
GAME004	Zelda	Classic	2002	100000	9
GAME005	GTAV	Simulation	2015	150000	8
GAME006	FIFA 22	Sports	2021	250000	9
GAME008	Read Dead Redemption 2	Action	2018	300000	58
GAME003	COD MW	Action	2020	500000	8
GAME001	Elden Ring	Adventure	2022	700000	8
GAME007	Fisika Dasar	Horror	2022	1000000	8

Press Enter To Continue >>>

F08 - Membeli dan Menjual Game

Tampilan jika game telah dimiliki

```
Masukan id game yang ingin dibeli: GAME006
Anda sudah memiliki game tersebut!
=====
Press Enter To Continue >>>
```

Tampilan jika stok atau saldo tidak cukup

```
Masukan id game yang ingin dibeli: GAME007
Maaf, stok game tidak mencukupi atau saldo anda tidak mencukupi untuk membeli game tersebut!
=====
Press Enter To Continue >>>
```

Tampilan Berhasil

```
Masukan id game yang ingin dibeli: GAME010
Selamat! Anda telah membeli game VALORANT
```

```
=====
Press Enter To Continue >>>
```

F09 - Melihat Game yang Dimiliki

Tampilan jika tidak memiliki game

```
Anda belum memiliki game!
```

```
=====
Press Enter To Continue >>>
```

Tampilan jika memiliki game

```
=====
| ID Game | Nama Game | Kategori | Tahun Rilis | Harga |
=====
| GAME010 | VALORANT | Action   | 2020         | 0     |
| GAME006 | FIFA 22  | Sports   | 2021         | 250000|
=====
```

```
Press Enter To Continue >>>
```

F10 - Mencari Game yang Dimiliki dari ID dan Tahun Rilis

Tampilan pencarian jika tidak sesuai

```
Masukan id game: GAME077
Masukan tahun rilis:
Daftar game pada toko yang memenuhi kriteria:
Tidak ada game pada toko yang memenuhi kriteria
```

```
=====
Press Enter To Continue >>>
```

Tampilan Pencarian jika filter kosong

```
Masukan id game:
Masukan tahun rilis:
```

```
=====
| ID Game | Nama Game | Kategori | Tahun Rilis | Harga |
=====
| GAME001 | Elden Ring | Adventure | 2022         | 700000 |
| GAME003 | COD MW     | Action   | 2020         | 500000 |
| GAME005 | GTAV       | Simulation | 2015         | 150000 |
| GAME007 | Fisika Dasar | Horror   | 2022         | 10000000 |
| GAME009 | Dasar Pemrograman : Semester 2 | Education | 2022         | 10     |
| GAME002 | Valhalla   | Adventure | 2021         | 40000  |
=====
```

```
Press Enter To Continue >>>
```

Tampilan Pencarian sesuai filter

```
Masukan id game: GAME001
Masukan tahun rilis:
=====
| ID Game | Nama Game | Kategori | Tahun Rilis | Harga |
=====
| GAME001 | Elden Ring | Adventure | 2022 | 700000 |
=====
Press Enter To Continue >>> █
```

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

Tampilan filter salah

```
Masukan id game:
Masukan nama game: Hahahates
Masukan harga game:
Masukan kategori game:
Masukan tahun rilis:
Daftar game pada toko yang memenuhi kriteria:
Tidak ada game pada toko yang memenuhi kriteria
=====
Press Enter To Continue >>> █
```

Tampilan berhasil

```
Masukan id game:
Masukan nama game:
Masukan harga game:
Masukan kategori game:
Masukan tahun rilis:
Daftar game pada toko yang memenuhi kriteria:
=====
| ID Game | Nama Game | Kategori | Tahun Rilis | Harga | Stok |
=====
| GAME001 | Elden Ring | Adventure | 2022 | 700000 | 8 |
| GAME002 | Valhalla | Adventure | 2021 | 40000 | 6 |
| GAME003 | COD MW | Action | 2020 | 500000 | 8 |
| GAME004 | Zelda | Classic | 2002 | 100000 | 9 |
| GAME005 | GTAV | Simulation | 2015 | 150000 | 8 |
| GAME006 | FIFA 22 | Sports | 2021 | 250000 | 8 |
| GAME007 | Fisika Dasar | Horror | 2022 | 10000000 | 8 |
| GAME008 | Read Dead Redemption 2 | Action | 2018 | 300000 | 58 |
| GAME009 | Dasar Pemrograman : Semester 2 | Education | 2022 | 10 | 548 |
| GAME010 | VALORANT | Action | 2020 | 0 | 99 |
=====
Press Enter To Continue >>> █
```

F12 - Top Up Saldo

Penambahan Saldo

```

Masukan username: 16521510
Masukan jumlah saldo yang ingin ditambahkan: 50000
Top up berhasil! Saldo 16521510 menjadi 1050000
=====
Press Enter To Continue >>> █

```

Pengurangan Saldo

```

Masukan username: 16521510
Masukan jumlah saldo yang ingin ditambahkan: -700000
Top up berhasil! Saldo 16521510 menjadi 350000
=====
Press Enter To Continue >>> █

```

Username Salah

```

Masukan username: username
Maaf, username tidak ditemukan!
=====
Press Enter To Continue >>> █

```

F13 - Melihat Riwayat Pembelian

```

Riwayat transaksi:
=====
| ID Game | Nama Game | Harga Game | Tahun Beli |
=====
| GAME001 | Elden Ring | 700000 | 2022 |
| GAME003 | COD MW | 500000 | 2022 |
| GAME005 | GTAV | 150000 | 2022 |
| GAME007 | Fisika Dasar | 10000000 | 2022 |
| GAME009 | Dasar Pemrograman : Semester 2 | 10 | 2022 |
| GAME002 | Valhalla | 40000 | 2022 |
=====
Press Enter To Continue >>> █

```

F14 - Help

Tampilan help untuk admin

```

=====H E L P=====
1. Register - Untuk mendaftar akun
2. Login - Untuk login ke akun
3. Tambah Game - Untuk menambah game ke dalam toko
4. Ubah Game - Untuk mengubah game yang telah ditambahkan
5. Ubah Stok - Untuk mengubah stok game yang telah ditambahkan
6. Listing Game Toko - Untuk menampilkan daftar game yang tersedia di toko
7. Cari Game Toko - Untuk mencari game yang tersedia di toko
8. Top Up - Untuk menambah saldo
9. Help - Untuk melihat daftar menu
10. Save - Untuk menyimpan data ke dalam file
11. Exit - Untuk keluar dari program
12. Magic Conch Shell - Minigame yang dapat menjawab pertanyaan kita
13. Tic Tac Toe - Minigame yang Tic Tac Toe
=====
Press Enter To Continue >>> █

```

Tampilan help untuk user

```

=====H E L P=====
1.Login - Untuk login ke akun
2. Listing Game Toko 0 - Untuk menampilkan daftar game yang tersedia di toko
3. Beli Game - Untuk membeli game
4. Lihat Game Saya - Untuk melihat daftar game yang telah dibeli
5. Cari Game Saya - Untuk mencari game yang telah dibeli
6. Cari Game Toko - Untuk mencari game yang tersedia di toko
7. Riwayat Pembelian - Untuk melihat daftar game yang telah dibeli
8. Help - Untuk melihat daftar menu
9. Save - Untuk menyimpan data ke dalam file
10. Exit - Untuk keluar dari program
11. Magic Conch Shell - Minigame yang dapat menjawab pertanyaan kita
12. Tic Tac Toe - Minigame yang Tic Tac Toe
=====
Press Enter To Continue >>> █

```

F15 - Load

Tampilan jika tidak diberikan nama folder

```

Tidak ada nama folder yang diberikan!
usage: main.py [-h] NamaFolder
main.py: error: the following arguments are required: NamaFolder
PS C:\File\Nadil\STEI ITB\Semester 2\Daspro\Program\Tubes-Daspro-Kelompok-1> █

```

Tampilan jika nama folder yang diberikan tidak ditemukan

```

Folder asdas tidak ditemukan!
PS C:\File\Nadil\STEI ITB\Semester 2\Daspro\Program\Tubes-Daspro-Kelompok-1> █

```

Tampilan jika nama folder benar dan berhasil dilakukan load data

```
=====MENU=====
```

1. Login
2. Help

```
=====
```

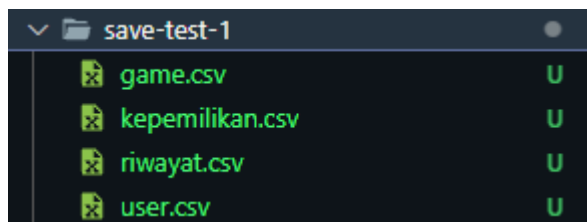
Pilih nomor menu: █

F16 - Save

```
Masukan nama folder penyimpanan: save-test-1
```

```
=====
```

Press Enter To Continue >>> █



F17 - Exit

```
Terima kasih telah menggunakan aplikasi ini.  
Sampai jumpa lagi!
```

```
PS C:\File\Nadil\STEI ITB\Semester 2\Daspro\Program\Tubes-Daspro-Kelompok-1> █
```

B01 - Cipher

Bentuk password sudah di-cipher pada file csv

```
id;username;nama;password;role;saldo  
1;admin;admin;zExd3;admin;100000000000000000  
2;16521516;Ahmad Nadil;7zRR6UPA6P6U;user;1000000  
3;16521546;Gholib;7zRR6UPA6PKU;user;1000000  
4;16521510;Zain;7zRR6UPA6P6o;user;350000  
5;16521524;Haikal;7zRR6UPA6PAK;user;1000000  
6;16521548;Athira;7zRR6UPA6PKe;user;750000
```

B02 - Magic Conch Shell

```
Apa pertanyaanmu? Apakah daspro mudah?
```

```
Ya
```

```
Apa pertanyaanmu? Apakah tubes daspro kurang banyak?
```

```
Tidak
```

```
Apa pertanyaanmu? █
```

B03 - TicTacToe

```
0 | X | O
---+---+---
X |   |
---+---+---
   |   |
Giliran O
Pilih posisi: 4
Posisi sudah terisi
```

```
0 | X | O
---+---+---
   |   |
---+---+---
   |   |
Giliran X
Pilih posisi: 4
```

```
0 | X |
---+---+---
   |   |
---+---+---
   |   |
Giliran O
Pilih posisi: 3
```

```
0 | X | O
---+---+---
X | O | X
---+---+---
O |   |
Pemenangnya adalah  O
=====
Press Enter To Continue >>>
```

LAMPIRAN

Nomor Asistensi	:	1
No. Kelompok/Kelas	:	1 / K-12
Tanggal asistensi	:	13 April 2022

Anggota kelompok

	NIM / Nama (Hanya yang Hadir)
1	16521516 / Ahmad Nadil
2	16521510 / Muhammad Rifky Fachrizain
3	16521524 / Haikal Ardzi Shofiyyurrohman
4	16521546 / Dhafin Ghalib Luqman Hakim
5	16521548 / Athira Dhyannis Tafkir
6	

Asisten pembimbing

NIM / Nama
13519179 / Akifa Nabil Ufairah

Catatan Asistensi:

Rangkuman Diskusi

Fungsi yang dibuat harus dipisah satu-satu atau gimana? Fungsi tidak harus dipisah satu-satu dan beberapa fungsi bisa dijadiin satu modul, tetapi dibebaskan kalian dekomposisinya gimana

Fungsi help langsung di print tanpa dipanggil, karena digunakan sebagai menu? Boleh-boleh saja

Notal dibikinnya gimana? Dibikin di template di screenshot ke laporan

Untuk soal bonus no 2, apa ada yang perlu diubah atau diperbaiki? Udah aman

Tindak Lanjut

Menyusun laporan akhir

Nomor Asistensi	:	2
No. Kelompok/Kelas	:	1 / K-12
Tanggal asistensi	:	20 April 2022

Anggota kelompok

	NIM / Nama (Hanya yang Hadir)
1	16521516 / Ahmad Nadil
2	16521510 / Muhammad Rifky Fachrizain
3	16521524 / Haikal Ardzi Shofiyyurrohman
4	16521546 / Dhafin Ghalib Luqman Hakim
5	16521548 / Athira Dhyannis Tafkir
6	

Asisten pembimbing

NIM / Nama
13519179 / Akifa Nabil Ufairah

Catatan Asistensi:

Rangkuman Diskusi

1. Jika ada bentuk python yang tidak diajarkan bentuk notasi algoritmiknya bagaimana ya?

Dapat menggunakan syntax pythonnya saja untuk notasi algoritmiknya, karena notal tidak terlalu strict aturannya
2. Bentuk laporan Spesifikasi, Kamus, dan Dekomposisi bagaimana ya?

Bentuk nya dipisah perbagian fungsi, terpisah di sub bab yang berbeda dalam laporannya.
3. Untuk main menu pada source code, apakah boleh hanya menampilkan menu yang hanya bisa diakses oleh role user yang sedang login?

Boleh, disesuaikan saja kalian enak nya bagaimana,

Tindak Lanjut

Menyelesaikan Ilaporan akhir