

# **TUGAS BESAR SIMULASI INVENTARISASI GADGET**

**IF1210 DASAR PEMROGRAMAN  
KELAS 10**



Oleh  
**Kelompok 11**

16520070 Diffa' Shada 'Aqila  
16520090 Bostang Palaguna  
16520170 Muhammad Daris Nurhakim  
16520230 Joshi Ryu Setiady

**Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
2021**

## **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 IF 1210 Dasar Pemrograman Semester 2 2020/2021”

## **DAFTAR ISI**

HALAMAN PERNYATAAN KELOMPOK

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

1. DESKRIPSI PERSOALAN
2. PEMBAGIAN KERJA ANGGOTA KELOMPOK
3. CHECKLIST HASIL RANCANGAN, IMPLEMENTASI, DAN TESTING
4. DESAIN COMMAND
5. DESAIN KAMUS DATA
6. DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL
7. SPESIFIKASI UNTUK TIAP MODUL/PROSEDUR/FUNGSI
8. SCREENSHOT HASIL PENGUJIAN PROGRAM

LAMPIRAN FORM ASISTENSI

## **DAFTAR TABEL**

Tabel 1. Tabel Pembagian Kerja Anggota Kelompok

Tabel 2. Tabel Checklist Hasil Rancangan, Implementasi, dan Testing

## **DAFTAR GAMBAR**

- Gambar 1. Tampilan uji coba fungsi F14
- Gambar 2. Tampilan uji coba fungsi F02
- Gambar 3. Tampilan uji coba fungsi F01
- Gambar 4. Tampilan uji coba fungsi F03
- Gambar 5. Tampilan uji coba fungsi F04
- Gambar 6. Tampilan uji coba fungsi F05
- Gambar 8. Tampilan uji coba fungsi F06
- Gambar 9. Tampilan uji coba fungsi F07
- Gambar 10. Tampilan uji coba fungsi F08
- Gambar 11. Tampilan uji coba fungsi F09
- Gambar 12. Tampilan uji coba fungsi F10
- Gambar 13. Tampilan uji coba fungsi F11
- Gambar 14. Tampilan uji coba fungsi F12
- Gambar 15. Tampilan uji coba fungsi F13

### **1. DESKRIPSI PERSOALAN**

Tugas Besar mata kuliah Dasar Pemrograman kali ini mendeskripsikan sebuah persoalan, yaitu sebuah simulasi inventarisasi gadget. Sistem ini menggunakan metode simulasi, yang mana diasumsikan pengguna menginput informasi sebebas-bebasnya. Informasi yang diinputkan oleh pengguna disimpan dalam suatu folder tertentu dan file berformat CSV (Comma Separated Values) untuk memisahkan input pengguna sesuai spesifikasi database simulator. Dengan spesifikasi dan ketentuan sebagai berikut,

1. Prosedur login, Prosedur untuk meminta input dari pengguna sesuai dengan data yang diinputkan ke dalam sistem
2. Prosedur register, Prosedur untuk mendaftarkan keanggotaan pengguna ke dalam simulator
3. Prosedur cari\_rarity, Prosedur untuk mencari kelangkaan sebuah item gadget dalam database simulator
4. Prosedur cari\_tahun, Prosedur untuk mencari item gadget sesuai tahun yang diinputkan, tahun dapat diinputkan sesuai maupun tahun setelahnya dan tahun sebelumnya
5. Prosedur tambah\_item, Prosedur untuk memasukkan item gadget baru ke dalam database simulator
6. Prosedur hapus\_item, Prosedur untuk menghapus data item gadget di dalam database simulator
7. Prosedur ubah\_jumlah, Prosedur untuk mengubah jumlah item gadget yang tersedia dalam database simulator

8. Prosedur pinjam\_gadget, Prosedur untuk meminjam item gadget dengan jumlah tertentu dan menginput tanggal peminjaman dari dalam database simulator
9. Prosedur kembalikan\_gadget, Prosedur untuk mengembalikan item gadget dengan jumlah tertentu dan memasukkan tanggal pengembalian ke dalam database simulator
10. Prosedur minta\_consumable, Prosedur untuk meminta item consumable dengan jumlah tertentu dan memasukkan tanggal permintaan ke dalam database simulator
11. Prosedur riwayat\_pinjam, Prosedur untuk menampilkan data rekam jejak peminjaman item gadget dari dalam database simulator
12. Prosedur riwayat\_kembali, Prosedur untuk menampilkan data rekam jejak pengembalian item gadget ke dalam database simulator
13. Prosedur riwayat\_ambil, Prosedur untuk menampilkan data rekam jejak permintaan item consumable dari dalam database simulator
14. Prosedur load\_data, Prosedur untuk menampilkan loading data ke dalam nama folder yang diinputkan untuk menjalankan program pertama kali
15. Prosedur save\_data, Prosedur untuk menyimpan semua data input dari pengguna ke dalam masing-masing file bertipe CSV
16. Prosedur help, Prosedur untuk menampilkan perintah command bantuan untuk mengakses setiap fungsi dalam simulator
17. Prosedur exit, Prosedur untuk menghentikan seluruh aktivitas dalam sistem inventarisasi gadget sesuai input pengguna

## 2. PEMBAGIAN KERJA ANGGOTA KELOMPOK

Tabel 1. Tabel Pembagian Kerja Anggota Kelompok

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F01	Procedure register	16520070 16520230	16520070 16520230	16520090 16520170
F02	Procedure login	16520070 16520230	16520070 16520230	16520090 16520170
F03	Procedure cari_rarity	16520070 16520170	16520070 16520170	16520090 16520230
F04	Procedure cari_tahun	16520070 16520170	16520070 16520170	16520090 16520230
F05	Procedure tambah_item	16520090 16520170	16520090 16520170	16520070 16520230
F06	Procedure hapus_item	16520090 16520170	16520090 16520170	16520070 16520230

F07	Procedure ubah_jumlah	16520090 16520170	16520090 16520170	16520070 16520230
F08	Procedure pinjam_gadget	16520070 16520090	16520070 16520090	16520170 16520230
F09	Procedure kembalikan_gadget	16520070 16520090	16520070 16520090	16520170 16520230
F10	Procedure minta_consumable	16520070 16520230	16520070 16520230	16520090 16520170
F11	Procedure riwayat_pinjam	16520170 16520230	16520170 16520230	16520070 16520090
F12	Procedure riwayat_kembali	16520170 16520230	16520170 16520230	16520070 16520090
F13	Procedure riwayat_ambil	16520170 16520230	16520170 16520230	16520070 16520090
F14	Procedure load_data	16520090 16520230	16520090 16520230	16520070 16520170
F15	Procedure save_data	16520090 16520230	16520090 16520230	16520070 16520170
F16	Procedure help	16520090 16520230	16520090 16520230	16520070 16520170
F17	Procedure exit	16520090 16520230	16520090 16520230	16520070 16520170

### 3. CHECKLIST HASIL RANCANGAN, IMPLEMENTASI, DAN TESTING

Tabel 2. Tabel Checklist Hasil Rancangan, Implementasi, dan Testing

Fitur	Desain	Implementasi	Testing
F01	✓	✓	✓
F02	✓	✓	✓
F03	✓	✓	✓
F04	✓	✓	✓
F05	✓	✓	✓

F06	✓	✓	✓
F07	✓	✓	✓
F08	✓	✓	✓
F09	✓	✓	✓
F10	✓	✓	✓
F11	✓	✓	✓
F12	✓	✓	✓
F13	✓	✓	✓
F14			
F15			
F16	✓	✓	✓
F17	✓	✓	✓

#### 4. DESAIN COMMAND

1. \$ register
  - > Masukkan nama:
  - > Masukkan username:
  - > Masukkan password:
  - > Masukkan alamat:
  - > User {username} telah berhasil register ke dalam Kantong Ajaib.
2. \$ login
  - > Masukkan username:
  - > Masukkan password:
  - > Halo {username}! Selamat datang di Kantong Ajaib.
3. \$ cari\_rarity
  - > Masukkan rarity:
  - > Hasil pencarian:
4. \$ cari\_tahun
  - > Masukkan tahun:
  - > Masukkan kategori:
  - > Hasil pencarian:
5. \$ tambah\_item
  - > Masukkan ID:



- > Masukkan Nama:
  - > Masukkan Deskripsi:
  - > Masukkan Jumlah:
  - > Masukkan Rarity:
  - > Masukkan Tahun Ditemukan:
6. \$ hapus\_item
- > Masukkan ID item:
  - > Apakah Anda yakin ingin menghapus {nama item}? (y/n)
7. \$ ubah\_jumlah
- > Masukkan ID:
  - > Masukkan Jumlah:
  - > {jumlah item} {nama item} berhasil {ditambahkan/dibuang}. Stok sekarang: {jumlah item yang baru}
8. \$ pinjam\_gadget
- > Masukkan ID item:
  - > Tanggal peminjaman:
  - > Jumlah peminjaman:
  - > Item {nama gadget} (× {jumlah peminjaman}) berhasil dipinjam!
9. \$ kembalikan\_gadget
- > {list nama gadget yang dipinjam}
  - > Masukkan nomor peminjaman:
  - > Tanggal pengembalian:
  - > Item {nama gadget} (× {jumlah peminjaman}) telah dikembalikan!
10. \$ minta\_consumable
- > Masukkan ID item:
  - > Jumlah:
  - > Tanggal permintaan:
  - > Item {nama gadget} (× {jumlah peminjaman}) telah berhasil diambil!
11. \$ riwayat\_pinjam
- > Riwayat peminjaman:
  - > Apakah Anda mau ke halaman selanjutnya? (y/n)
  - > Semua riwayat sudah ditampilkan
12. \$ riwayat\_kembali
- > Riwayat pengembalian:
  - > Apakah Anda mau ke halaman selanjutnya? (y/n)
  - > Semua riwayat sudah ditampilkan
13. \$ riwayat\_ambil
- > Riwayat pengambilan:
  - > Apakah Anda mau ke halaman selanjutnya? (y/n)

- > Semua riwayat sudah ditampilkan
- 14. \$ load\_data
- 15. \$ save\_data
  - > Masukkan nama folder penyimpanan:
  - > Saving...
  - > Data telah disimpan pada folder {nama folder}!
- 16. \$ help
  - > {Penjelasan mengenai prosedur yang ada}
- 17. \$ exit
  - > Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n)

## 5. DESAIN KAMUS DATA

- Desain Kamus Data Lokal untuk fungsi utama dalam program:
  - function clear\_screen
  - cls: string\_os -- untuk membersihkan tampilan di program
  - function splitList
  - database: array\_of\_string
  - splitList: array\_of\_string -- digunakan pemisah delimiter ;
  - function convert\_line\_to\_data
  - array\_from\_data: array\_of\_string
  - array\_from\_data: splitList array\_of\_string
  - function convert\_array\_to\_real\_value
  - array\_of\_data: array\_of\_string
  - array:data: integer, array\_of\_string
  - function overwrite\_database
  - datas\_as\_string: string
  - array\_database from csv
- Desain Kamus Data Tambahan untuk tiap fungsi utama:
  - Array\_of\_data:array\_of\_string
  - g: textIOwrapper
  - c: textIOwrapper
  - id\_remove: string
  - kondisiLanjut: boolean
  - ArrayJumlahPinjam: array\_of\_integer
  - ArrayBarangPinjam: array\_of\_string)
  - (idUbah: string
  - idPinjam: string
  - idPengembalian: string
  - idConsumable: string
  - ubahJumlah: integer

jumlahPengembalian: integer  
tanggalPengembalian: string  
tanggalPinjam: string  
raw\_lines\_g: array\_of\_character,  
raw\_lines\_c: array\_of\_character,  
raw\_lines\_u: array\_of\_character,  
raw\_lines\_gb: array\_of\_characte)  
\_gadget: array\_database from file csv  
\_consumable: array\_database from file csv  
\_user: array\_database from file csv  
lines\_g: array\_of\_character  
lines\_c : array\_of\_character

- Desain Kamus Data Lokal untuk setiap fungsi:

**F01:**

- function add\_item\_to\_database  
data\_baru: list\_of\_string  
\_user:array\_of\_string
- function user\_list  
nama\_user: list\_of\_string,array\_of\_string  
output: boolean\_of\_array
- function register  
username, nama,password,alamat: string  
output: string

**F02:**

- function user\_list  
nama\_user: list\_of\_string,array\_of\_string  
output: boolean\_of\_array
- function password\_list  
password: list\_of\_string,array\_of\_string  
output: boolean\_of\_array
- function id\_user  
id\_user: list\_of\_string,array\_of\_string  
output: boolean\_of\_array
- function login  
username,password:string  
output:string

**F03:**

- function isRarityinDatabase  
element:array\_of\_string  
output: boolean\_of\_array

- function infoBarang  
arrayProcess:array\_of\_string  
output: baris:array\_of\_string
- function cariRarity  
rarityitem:string  
output: string

**F04:**

- function IsTahuninDatabase  
element: array\_of\_string  
output: boolean\_of\_array
- function infoBarang  
arrayProcess:array\_of\_string  
output: baris:array\_of\_string
- procedure cariTahun  
tahun,kategori:string  
output: list\_of\_string

**F05:**

- function isIDinDatabase  
element:array\_of\_string  
output: boolean\_of\_array
- function add\_item\_to\_database  
data\_baru: list\_of\_string  
\_user:array\_of\_string
- function tambah\_item:  
id,nama,deskripsi,jumlah,rarity,tahuntemu:string  
output: string

**F06:**

- function isIDinDatabase  
element:array\_of\_string  
output: boolean\_of\_array
- function infoBarang  
arrayProcess:array\_of\_string  
baris:array\_of\_string
- function hapusItem  
konfirmasi: string  
output: string

**F07:**

- function isIDinDatabase  
element:array\_of\_string  
output: boolean\_of\_array

- function convert\_to\_real

array\_salinan: array of array of integer,sing; kolom\_jumlah: integer;baris:

array\_of\_string,integer

output: array\_of\_string

- function change\_quantity\_in\_database

arrayProcess: array\_of\_array\_of\_integer,string; baris: array\_of\_integer.string

arrayProcess: array\_of\_string

- function infoBarang

arrayProcess:array\_of\_string

baris:array\_of\_string

- procedure ubahJumlah

idUbah, jumlahUbah: string; hasilUbah: array\_of\_string

output: string

#### **F08:**

- function modify\_datas

output: array\_of\_data

- function isIDinDatabase

element:array\_of\_string

output: boolean\_of\_array

- function isUserinDatabase

element:array\_of\_string

output: boolean\_of\_array

- function infoBarang

arrayProcess:array\_of\_string

baris:array\_of\_string

- function isLocIDinArray

k: integer

k: integer, string

- function infoUser

baris: array\_of\_string

baris: array\_of\_string,string

- function infoPinjam

arrayProcess: array\_of\_string

baris: array\_of\_string dari arrayProcess

- function pinjamGadget:

idPinjam, tanggalPinjam,JumlahPinjam: string, character

output: string

#### **F09:**

- function modify\_datas

output: array\_of\_data

- function isIDinDatabase  
 element:array\_of\_string  
 output: boolean\_of\_array
- function isUserinDatabase  
 element:array\_of\_string  
 output: boolean\_of\_array
- function infoBarang  
 arrayProcess:array\_of\_string  
 baris:array\_of\_string
- function isLocIDinArray  
 k: integer  
 k: integer, string
- function infoUser  
 baris: array\_of\_string  
 baris: array\_of\_string,string
- function inventory  
 riwayat: array\_of\_string  
 k: integer
- function infoPinjam  
 arrayProcess: array\_of\_string  
 baris: array\_of\_string dari arrayProcess
- function infoKembali  
 arrayProcess: array\_of\_string  
 baris: array\_of\_string dari arrayProcess
- function kembalikanGadget  
 dataKembali: string\_to\_database\_csv  
 output: array\_of\_string\_from\_database

**F10:**

- function modify\_datas  
 output: array\_of\_data
- function isIDinDatabase  
 element:array\_of\_string  
 output: boolean\_of\_array
- function isUserinDatabase  
 element:array\_of\_string  
 output: boolean\_of\_array
- function infoBarang  
 arrayProcess:array\_of\_string  
 baris:array\_of\_string
- function isLocIDinArray

k: integer  
k: integer, string  
- function infoUser  
baris: array\_of\_string  
baris: array\_of\_string,string  
- function mintaConsum  
idPeminta: string  
jumlahPermintaan: integer  
output : string

**F11:**

- function isIDinDatabase  
element:array\_of\_string  
output: boolean\_of\_array  
- function isUserinDatabase  
element:array\_of\_string  
output: boolean\_of\_array  
- function infoBarang  
arrayProcess:array\_of\_string  
baris:array\_of\_string  
- function isLocIDinArray  
k: integer  
k: integer, string  
- function infoUser  
baris: array\_of\_string  
baris: array\_of\_string,string  
- function infoPinjam  
arrayProcess: array\_of\_string  
baris: array\_of\_string dari arrayProcess  
procedure riwayatPinjam  
lanjut: string  
output: list\_of\_string, array\_of\_string

**F12:**

- function isIDinDatabase  
element:array\_of\_string  
output: boolean\_of\_array  
- function isUserinDatabase  
element:array\_of\_string  
output: boolean\_of\_array  
- function infoBarang  
arrayProcess:array\_of\_string

baris:array\_of\_string  
- function infoUser  
baris: array\_of\_string  
baris: array\_of\_string,string  
- function infoPinjam  
arrayProcess: array\_of\_string  
baris: array\_of\_string dari arrayProcess  
- function isPinjamInDatabase  
element: array\_of\_string  
output: boolean\_of\_array  
- function infoKembali  
arrayProcess: array\_of\_string  
baris: array\_of\_string dari arrayProcess  
procedure riwayatKembali  
lanjut: string  
output: list\_of\_string, array\_of\_string

**F13:**

- function isIDinDatabase  
element:array\_of\_string  
output: boolean\_of\_array  
- function isUserinDatabase  
element:array\_of\_string  
output: boolean\_of\_array  
- function infoBarang  
arrayProcess:array\_of\_string  
Baris:array\_of\_string  
- function infoUser  
baris: array\_of\_string  
baris: array\_of\_string,string  
- function infoAmbil  
arrayProcess: array\_of\_string  
baris: array\_of\_string dari arrayProcess  
- function riwayatAmbil  
lanjut: string  
output: list\_of\_string, array\_of\_string

**F14:**

**F15:**

- function simpan  
folder : string



type folder = c; string,ch: string,gb: string, g: string, gr: string, u: string

Folder:

SEQFILE of :

- (1) consumable.csv
- (2) gadget.csv
- (3) consumable\_history.csv
- (4) gadget\_borrow\_history.csv
- (5) gadget\_return\_history.csv
- (6) user.csv

**F16:**

- function isUserinDatabase

element:array\_of\_string

output: boolean\_of\_array

- function infoUser

baris: array\_of\_string

baris: array\_of\_string,string

- function help

output: string

**F17:**

- function keluar

save : string

output: string

## 6. DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL

- Notasi Algoritmik fungsi utama tiap program:

**Basic Procedure:**

```
import os
```

```
function clear_screen():
```

```
    # Membuat efek clear screen pada Python
```

```
    os.system('cls' depend on os.name = 'nt' else 'clear')
```

```
function splitList(list):
```

```
# menulis data per baris dalam csv ke dalam bentuk array dengan delimiter ";"
```

```
# KAMUS LOKAL
```

```
# Variabel
```

```
# database : array of string { array sementara [masih kotor oleh \n] }
```

```
# aplit_list : array of string { array hasil berisi data-data dari csv}
```

```
# ALGORITMA
```

```
sentence <- ".join(list)
```

```
split_list = []
```

```
tmp = "
```

```
iterate s in sentence:
```

```
if s = ';':
```

```
split_list.append(tmp)
```

```
    tmp <- "
```

```
else:
```

```
    tmp += s
```

```
if tmp:
```

```
split_list.append(tmp)
```

```
    Stop:
```

```
return split_list
```

```
function convert_line_to_data(line):
```

```
# Mengubah baris menjadi data
```

```
raw_array_from_data = splitList(line)
```

```
array_from_data = [data.strip() for data in raw_array_from_data]
```

```
return array_from_data
```

```
function convert_array_to_real_value(array_data):
```

```
# Mengubah array data ke dalam nilai asli
```

```
arr_cpy = array_data[:]
```

```
for i in range(6):
```

```
depend on (i <- 0):
```

```
    arr_cpy[i] <- int(arr_cpy[i])
```

```
else:
```

```
    arr_cpy[i] <- str(arr_cpy[i])
```

```
return arr_cpy
```

- Notasi Algoritmik untuk tiap-tiap fungsi:

1. F01:

```
procedure login ( input : string, output:string)
```

```
-- mencocokkan input pengguna dengan yang ada di dalam array database
```

```
{I.S. pengguna menginput string username dan password}
```

```
{F.S. program mencetak output keluaran sesuai masukan program}
```

```
import os
```

```
function clear_screen():
```

```
# Membuat efek clear screen pada Python
```

```
os.system('cls' depend on os.name = 'nt' else 'clear')
```

```
function splitList(list):
```

```
# menulis data per baris dalam csv ke dalam bentuk array dengan delimiter ";"
```

```
# KAMUS LOKAL
```

```
# Variabel
```

```
# database : array of string { array sementara [masih kotor oleh \n] }
```

```
# aplit_list : array of string { array hasil berisi data-data dari csv}
```

```

# ALGORITMA

sentence <- ".join(list)

split_list = []

tmp = ""

iterate s in sentence:

    if s = ';':

        split_list.append(tmp)

        tmp <- ""

    else:

        tmp += s

if tmp:

    split_list.append(tmp)

stop:

return split_list

function convert_line_to_data(line):

    # Mengubah baris menjadi data

    raw_array_from_data = splitList(line)

    array_from_data <- [data.strip() for data in raw_array_from_data]

    return array_from_data

function convert_array_to_real_value(array_data):

    # Mengubah array data ke dalam nilai asli

    arr_cpy = array_data[:]

    for i in range(6):

        depend on (i <- 0):

            arr_cpy[i] <- int(arr_cpy[i])

```

```
else:
    arr_cpy[i] <- str(arr_cpy[i])
return arr_cpy
```

```
import os
```

```
from Basic_Procedure import *
```

```
# Procedure registeimpr (input: username,nama,alamat,password berupa string,
output: string_data)
```

```
# I.S : User memasukkan input nama, username, pasword, dan alamat lalu
menyimpan hasil input ke dalam database
```

```
# F.S. : Program mengeluarkan output dengan perintah
```

```
#KAMUS
```

```
#username, password, nama, alamat, : string
```

```
#output : string
```

```
#FUMGSI-FUNGSI TAMBAHAN
```

```
function
```

```
add_item_to_database(register_indeks,username,nama,alamat,password,_user):
```

```
    data_baru <- [register_indeks,username,nama,alamat,password,'user']
```

```
    _user <- _user.append(data_baru)
```

```
function user_list(nama_user,_user):
```

```
    # Cek data username
```

```

iterate element in _user:
    depend on nama_user = element[1]:
        return True
return False

```

#### #ALGORITMA

```

function register(_user):
    nama <- input("Masukkan nama:").title()
    username <- input('Masukkan username:')
    depend on not user_list(username,_user): # Cek username yang sama
        password <- input('Masukkan password:')
        alamat <- input('Masukkan alamat:')
        register_indeks <- str(int(_user[-1][0]) + 1)
        add_item_to_database(register_indeks,username,nama,alamat,password,_user)
        output('User',username, "telah berhasil register ke dalam Kantong Ajaib!")#user
        telah berhasil register
        os.system('pause')
    else:
        output('Username sama. Gagal register!')

```

#### 2. F02:

```

{Program menerima input masukan registrasi user ke dalam mesin simulator}
function user_list(string -> array of string) -> array of string
-- menampilkan informasi username dalam array string
procedure register (input:string,output:string)
-- memasukkan informasi pengguna ke dalam program

```

{I.S. pengguna memasukkan informasi sesuai permintaan program}  
{F.S. output keluaran sebagai tanda user telah berhasil mendaftar dalam simulator}

```
import os
```

```
from Basic_Procedure import *
```

```
function user_list(nama_user,_user):
```

```
# Cek data username
```

```
depend on element in _user:
```

```
    else if nama_user = element[1]:
```

```
        return True
```

```
return False
```

```
function password_list(password,_user):
```

```
# Cek data password
```

```
iterate element in _user:
```

```
    if password = element[4]:
```

```
        return True
```

```
    stop:
```

```
return False
```

```
function id_user(nama_user,_user):
```

```
# Cek data username
```

```
for element in _user:
```

```
    if nama_user = element[1]:
```

```
        return element
```

```

# Procedure login

# User memasukkan input berupa username dan password kemudian diproses
dalam database

# Apabila username dan password sesuai maka output login berhasil

function login(_user):

  Loop = True

  while loop:

    username <- input('Masukkan username:')

    password <- input('Masukkan password:')

    depend on user_list(username,_user) == True and password_list(password,_user)
== True: # Cek username dan password dalam database

    output('Halo', username,'! Selamat datang di Kantong Ajaib.')

    user_aktif <- id_user(username,_user)

    loop <- False

    return user_aktif

  else:

    output('Username atau password anda salah. Gagal login!')

```

### 3. F03:

{Program membaca file csv dalam database dan mengeluarkan output berupa list string sesuai input pengguna}

function isRarity\_in\_database(string->array\_of\_string)->boolean\_of\_array

-- menerima validasi input pengguna berdasarkan array dalam database

function infoBarang(string->array\_of\_string)->array\_of\_string

-- melakukan validasi input pengguna sesuai ketentuan string (S,A,B,C) dan mencocokkan item rarity dalam database}



```
procedure cari_rarity(input:string,output:list_of_string)
-- mencari item rarity sesuai dengan input pengguna
{I.S. pengguna memasukkan input dengan key command sesuai fungsi di atas}
(F.S. output berupa list string dengan tampilan sesuai rarity yang diinputkan}
```

#### # REALISASI FUNGSI/PROSEDUR

```
function isRarityinDatabase(rarity,database):
    # mengecek apakah rarity yang diinginkan ada di database atau tidak
    # KAMUS LOKAL
        # element : array of string { baris pada database }
    # ALGORITMA
        iterate element in database:
            depend on rarity = element[4]: # element[4] berisi rarity yang ada di
            database
                return True
            return False

function infoBarang(rarity,spek):
    # mendapatkan informasi mengenai barang yang ada di database
    # KAMUS LOKAL
        # Variabel
            # arrayProcess : array of array of string { array tempat item berada }
            # baris : baris pada arrayProcess { untuk skema pencarian }
    # ALGORITMA
        depend on (rarity == 'S') or (rarity == 'A') or (rarity == 'B') or (rarity == 'C'): #
        jika rarity valid
            arrayProcess <- _gadget
        else:
            arrayProcess <- []

    # mengecek apakah rarity yang diinginkan ada di database atau tidak
    if isRarityinDatabase(rarity,arrayProcess):
        # mengecek baris per baris
        i = 0
        while arrayProcess[i] do:
            if spek == 'id':
                return baris[0]
            elif spek == 'nama':
                return baris[1]
            elif spek == 'deskripsi':
```

```

        return baris[2]
    elif spek == 'jumlah':
        return baris[3]
    elif spek == 'rarity':
        return baris[4]
    elif spek == 'tahun_ditemukan':
        return baris[5]
    i += 1
else:
    return "\nbarang tidak ditemukan di database"

```

# ALGORITMA UTAMA

# membaca file database .csv

g = open("gadget.csv", "r")

raw\_lines\_g = g.readlines()

g.close()

lines\_g = [raw\_line.replace("\n", "") for raw\_line in raw\_lines\_g]

# mengakses database dalam bentuk array

\_gadget = []

for line in lines\_g:

array\_of\_data = splitList(line)

\_gadget.append(array\_of\_data)

# melakukan skema pencarian berdasarkan rarity

clear\_screen()

database = \_gadget

output(">>> Pencarian Gadget berdasarkan rarity\n")

rarityItem <- input("Masukkan rarity: ").upper()

# jika inputan salah, maka akan diulang

while (rarityItem != 'S') and (rarityItem != 'A') and (rarityItem != 'B') and  
(rarityItem != 'C') do:

output("\nMasukan Anda salah!\n")

rarityItem <- input("Masukkan rarity: ").upper()

output("\nHasil pencarian:")

# mengecek apakah rarity ada di database atau tidak

depend on isRarityinDatabase(rarityItem, database):

iterate baris in database:

```

        if baris[4] == rarityItem: # baris[4] berisi data tentang rarity
            output(f"\nNama      : {infoBarang(rarityItem,'nama')}")
            output(f"Deskripsi   : {infoBarang(rarityItem,'deskripsi')}")
            output(f"Jumlah      : {infoBarang(rarityItem,'jumlah')}")
            output(f"Rarity      : {infoBarang(rarityItem,'rarity')}")
            output(f"Tahun ditemukan : {infoBarang(rarityItem,'tahun_ditemukan')}")

        output("\nSemua data telah ditampilkan")
    else:
        output("\nTidak ada")

    output("")
    os.system('pause')

```

#### 4. F04:

{Program membaca file csv dalam database dan mengeluarkan output list berupa string tahun dan informasi barang sesuai input pengguna}

function

isTahuninDatabase(string->list\_of\_string->array\_of\_string)->boolean\_of\_array

-- menerima validasi input pengguna sesuai dengan tahun dalam database

function infoBarang(string->array\_of\_string)->array\_of\_string

-- melakukan validasi input pengguna sesuai ketentuan string (S,A,B,C) dan mencocokkan item rarity dalam database}

procedure isTahuninDatabase(input:string,output:list\_of\_string)

-- mencocokkan informasi dalam database sesuai dengan input dari pengguna

{I.S. user memasukkan input berupa string(<,>,<=>,>=<=>) sesuai ketentuan program}

{F.S. output mengeluarkan list string dengan informasi sesuai input dari pengguna}

#REALISASI FUNGSI

clear\_screen()

database <- \_gadget

output(">>> Pencarian Gadget berdasarkan rarity\n")

```
# untuk memastikan bahwa tahun yang di input dalam integer, jika tidak bertipe integer maka akan meminta input ulang
```

```
while True:
```

```
    try:
```

```
        tahun = int(input("Masukkan tahun: "))
```

```
        break
```

```
    except ValueError:
```

```
        print("\nMasukan tahun dalam bentuk angka!\n")
```

```
kategori = input("Masukkan kategori: ")
```

```
# mengecek kategori, jika tidak valid maka akan diminta input ulang
```

```
while (kategori != '=') and (kategori != '>') and (kategori != '<') and (kategori != '>=') and (kategori != '<='):
```

```
    output("Kategori salah!")
```

```
    kategori <- input("Masukkan kategori: ")
```

```
output("\nHasil pencarian:")
```

```
# mengecek apakah tahun ada di database atau tidak, sesuai kategori
```

```
depend on isTahuninDatabase(tahun,kategori,database):
```

```
    for baris in database[1:]: # database dimulai pada baris ke 1(header tidak dihitung)
```

```
        depend on(kategori == '='):
```

```
depend on int(baris[5]) == tahun: # baris[5] berisi data tentang
tahun_ditemukan
```

```
output(f'\nNama      : {infoBarang(tahun,kategori,'nama')}')
```

```
ouput(f'Deskripsi   : {infoBarang(tahun,kategori,'deskripsi')}')
```

```
output(f'Jumlah     : {infoBarang(tahun,kategori,'jumlah')}')
```

```
output(f'Rarity     : {infoBarang(tahun,kategori,'rarity')}')
```

```
output(f'Tahun ditemukan :
{infoBarang(tahun,kategori,'tahun_ditemukan')}')
```

```
elif (kategori == '>'):
```

```
depend on int(baris[5]) >= tahun+1: # baris[5] berisi data tentang
tahun_ditemukan
```

```
output(f'\nNama      : {infoBarang(tahun,kategori,'nama')}')
```

```
output(f'Deskripsi   : {infoBarang(tahun,kategori,'deskripsi')}')
```

```
output(f'Jumlah     : {infoBarang(tahun,kategori,'jumlah')}')
```

```
output(f'Rarity     : {infoBarang(tahun,kategori,'rarity')}')
```

```
output(f'Tahun ditemukan :
{infoBarang(tahun,kategori,'tahun_ditemukan')}')
```

```
else if (kategori == '<'):
```

```
depend on int(baris[5]) <= tahun-1: # baris[5] berisi data tentang
tahun_ditemukan
```

```
output(f'\nNama      : {infoBarang(tahun,kategori,'nama')}')
```

```
output(f'Deskripsi   : {infoBarang(tahun,kategori,'deskripsi')}')
```

```
output(f'Jumlah     : {infoBarang(tahun,kategori,'jumlah')}')
```

```
output(f'Rarity     : {infoBarang(tahun,kategori,'rarity')}')
```

```

        output(f'Tahun ditemukan :
{infoBarang(tahun,kategori,'tahun_ditemukan')}')

    elif (kategori == '>='):

        if int(baris[5]) >= tahun: # baris[5] berisi data tentang tahun_ditemukan

            output(f'\nNama      : {infoBarang(tahun,kategori,'nama')}')

            output(f'Deskripsi   : {infoBarang(tahun,kategori,'deskripsi')}')

            output(f'Jumlah      : {infoBarang(tahun,kategori,'jumlah')}')

            output(f'Rarity      : {infoBarang(tahun,kategori,'rarity')}')

            output(f'Tahun ditemukan :
{infoBarang(tahun,kategori,'tahun_ditemukan')}')

        elif (kategori == '<='):

            if int(baris[5]) <= tahun: # baris[5] berisi data tentang tahun_ditemukan

                output(f'\nNama      : {infoBarang(tahun,kategori,'nama')}')

                output(f'Deskripsi   : {infoBarang(tahun,kategori,'deskripsi')}')

                output(f'Jumlah      : {infoBarang(tahun,kategori,'jumlah')}')

                output(f'Rarity      : {infoBarang(tahun,kategori,'rarity')}')

                output(f'Tahun ditemukan :
{infoBarang(tahun,kategori,'tahun_ditemukan')}')

    output("\nSemua data telah ditampilkan")

else:

    output("\nTidak ada gadget yang ditemukan")

```

```
output("")
```

```
os.system('pause')
```

5. F05:

{Program membaca file csv dan menerima input pengguna item gadget ke dalam database array}

function isIDindatabase(string->array\_of\_string)->boolean\_of\_array

-- menerima validasi input ID pengguna dalam database

procedure add\_item\_to\_database(input:string,output: string)

-- menerima input dari user berupa penambahan jumlah item gadget dan memberi informasi sesuai masukan

{I.S. input berupa string sesuai ketentuan yang diminta oleh program}

{F.S. output keluaran berupa string informasi masukan dari user}

#ALGORITMA

```
clear_screen()
```

```
kondisiLanjut <- True
```

```
print(">>> Menambah Item")
```

```
id <- input("Masukan ID: ")
```

```
if id[0] == 'C':
```

```
    tempatData <- _consumable
```

```
elif id[0] == 'G':
```

```
    tempatData <- _gadget
```

```
else:
```

```
    output("\nGagal menambahkan item karena ID tidak valid")
```

```
    kondisiLanjut <- False
```

if kondisiLanjut:

if not isIDinDatabase(id,tempatData):

nama <- input("Masukan Nama: ")

deskripsi <- input("Masukan Deskripsi: ")

i = 0

while True:

try:

jumlah <- int(input("Masukan Jumlah: "))

i <- i + 1

except ValueError:

output("\nMasukan jumlah dalam bentuk angka!\n")

rarity <- input("Masukan Rarity: ").upper()

while (rarity != 'C') and (rarity != 'B') and (rarity != 'A') and (rarity != 'S'):

output("\nInput rarity tidak valid!\n")

rarity = input("Masukan Rarity: ")

else:

if id[0] == 'G':

while True:

try:

tahuntemu <- int(input("Masukan tahun ditemukan: "))

break

except ValueError:

output("\nMasukan tahun dalam bentuk angka!\n")

else:



```
tahuntemu <- 0 # nilai dummy  
add_item_to_database(id,nama,deskripsi,jumlah,rarity,tahuntemu)  
output("\nItem telah berhasil ditambahkan ke database")
```

else:

```
output("\nGagal menambahkan item karena ID sudah ada")
```

# ALGORITMA UTAMA

```
# membaca file database.csv  
g <- open("gadget.csv","r")  
c <- open("consumable.csv","r")  
raw_lines_g = g.readlines()  
raw_lines_c = c.readlines()  
g.close()  
c.close()  
lines_g <- [raw_line.replace("\n", "") for raw_line in raw_lines_g]  
lines_c <- [raw_line.replace("\n", "") for raw_line in raw_lines_c]  
  
# mengakses database dalam bentuk array  
_gadget <- []  
_consumable <- []  
  
for line in lines_g:  
    array_of_data <- splitList(line)  
    _gadget.append(array_of_data)
```

```

for line in lines_c:

    array_of_data <- splitList(line)

    _consumable.append(array_of_data)

# menambahkan gadget baru

tambahItem()

output("")

os.system('pause')

```

6. F06:

{Program membaca file csv dan menerima input dari pengguna untuk menghapus item gadget dari dalam database}

function isIDindatabase(string->array\_of\_string)->boolean\_of\_array

-- menerima validasi input ID pengguna dalam database

function infoBarang(string->array\_of\_string)->array\_of\_string

-- melakukan validasi input pengguna sesuai ketentuan string (S,A,B,C) dan mencocokkan item rarity dalam database}

procedure hapus\_item(input:string,output:string)

-- menerima input dari user sesuai permintaan program dan menghapus data item gadget dalam database program

{I.S. user memasukkan input berupa string sesuai dengan ketentuan program dan memvalidasi program untuk menghapus item gadget}

{F.S. output mengeluarkan pesan string sebagai informasi item gadget telah dihapus dari dalam database}

function hapusItem():

# menerima input id barang yang ingin dihapuskan dan melakukan validasi

# KAMUS LOKAL

# Variabel

# database : array of array of string

```

# ALGORITMA

# memeriksa huruf pertama
if id_remove[0] == 'G':

    database <- _gadget

elif id_remove[0] == 'C':

    database = _consumable

else:

    database = []

kondisiLanjut = True

else if (isIDinDatabase(id_remove,database)):

    konfirmasi <- input(f"Apakah Anda yakin ingin menghapus
{infoBarang(id_remove,'nama')}(Y/N)? ")

    while (konfirmasi != 'Y') and (konfirmasi != 'y') and (konfirmasi != 'N') and
(konfirmasi != 'n'):

        output("\nHarap masukan dengan benar!\n")

    else:

        if (konfirmasi == 'Y') or (konfirmasi == 'y'):

            output("\nItem telah berhasil dihapus dari database")

        else:

            output("\nItem tidak dihapus dari database")

            kondisiLanjut = False

else: # barang tidak ditemukan di database

    output("\nTidak ada item dengan id tersebut")

    kondisiLanjut <- False

```

if kondisiLanjut:

    overwrite\_database(id\_remove[0])

7. F07:

{Program membaca file csv dan menerima input dari user untuk mengubah jumlah item gadget dalam database}

function infoBarang(string->array\_of\_string)->array\_of\_string

-- melakukan validasi input pengguna sesuai ketentuan string (S,A,B,C) dan mencocokkan item rarity dalam database}

function change\_quantity\_in\_database(string->string->string)->array\_of\_string

-- melakukan skema perubahan jumlah item gadget dalam database array

procedure ubah\_jumlah(input:string,output:string)

-- menerima input dari user dan mengubah jumlah item gadget dalam database dan memberi informasi sesuai input pengguna

{I.S. user menginput masukan sesuai permintaan program dan menghapus program sesuai inputan user}

{F.S. output mengeluarkan informasi kondisi jumlah item gadget terbaru setelah diubah dalam database array}

clear\_screen()

output(">>> Mengubah Jumlah Gadget atau Consumable")

idUbah <- input("Masukkan ID: ")

if idUbah[0] == 'C':

    arrayProcess = \_consumable

elif idUbah[0] == 'G':

    arrayProcess = \_gadget

else:

    arrayProcess = []

    # melakukan pemeriksaan validitas input

if (isIDinDatabase(idUbah,arrayProcess)):

```

jumlahUbah <- int(input("Masukkan jumlah: "))

if (jumlahUbah + int(infoBarang(idUbah,'jumlah')) >= 0):

    kondisiLanjut <- True

else:

    output(f"\n{(-1)*jumlahUbah} {infoBarang(idUbah,'nama')} gagal dibuang
    karena stok kurang. Stok sekarang: {infoBarang(idUbah,'jumlah')}
    (<{(-1)*jumlahUbah})")

    kondisiLanjut = False

else:

    output("\nTidak ada item dengan ID tersebut!")

    kondisiLanjut <- False

# melakukan overwrite perubahan data ke file csv

if kondisiLanjut:

    hasilUbah <- change_quantity_in_database(idUbah,jumlahUbah,idUbah[0])

    overwrite_database(idUbah[0])

    if (jumlahUbah >= 0):

        output(f"\n{jumlahUbah} {infoBarang(idUbah,'nama')} berhasil
        ditambahkan. Stok sekarang: {infoBarang(idUbah,'jumlah')}")

    else:

        output(f"\n{(-1)*jumlahUbah} {infoBarang(idUbah,'nama')} berhasil
        dibuang. Stok sekarang: {infoBarang(idUbah,'jumlah')}")

    output("")

    os.system('pause')

```

#### 8. F08:

{Program membaca file csv dan menerima input elemen peminjaman berdasarkan tanggal dan jumlah item gadget dan mengeluarkan informasi kondisi gadget terkini}

```

function user_list(string -> array of string) -> array of string
-- menampilkan informasi username dalam array string
function infoBarang(string->array_of_string)->array_of_string
-- melakukan validasi input pengguna sesuai ketentuan string (S,A,B,C) dan
mencocokkan item rarity dalam database}
procedure pinjam_gadget(input:string,output:string)
-- menerima input sesuai permintaan program dan meminjam item gadget dengan
jumlah tertentu dalam database
{I.S. input dari user berupa string dengan skema tanggal dan jumlah tertentu ke
dalam database}
{F.S. output berupa string informasi dari jumlah yang dipinjam oleh user dari
dalam database}

```

#REALISASI

```
clear_screen()
```

```
# melakukan proses peminjaman barang
```

```
output(">>> Meminjam Gadget")
```

```
kondisiLanjut <- True
```

```
idPinjam = input("Masukkan ID item: ")
```

```
for baris in (_gadgetBorrowHistory):
```

```
    i = 0
```

```
id_peminjam <- infoUser(userAktif,'id')
```

```
    if (id_peminjam <- infoPinjam('id_peminjam')) and (idPinjam ==
infoPinjam('id_gadget')) and (infoPinjam('is_returned') == 'False'):
```

```
        print("User tidak bisa meminjam lagi barang tersebut!")
```

```
        kondisiLanjut <- False
```

```
        i = i + 1
```

```
if kondisiLanjut:
```

```
    if isIDinDatabase(idPinjam, _gadget): # peminjaman barang HANYA DARI
gadget.csv
```

```

tanggal <- str(datetime.datetime.now().strftime("%d/%m/%Y"))
tanggalPinjam <- print("Tanggal peminjaman:",tanggal)
jumlahPinjam <- int(input("Jumlah peminjaman: "))
if (jumlahPinjam <= int(infoBarang(idPinjam,'jumlah'))):
    namaBarangPinjam <- infoBarang(idPinjam,"nama")
    output(f"Item {namaBarangPinjam} (x {jumlahPinjam}) berhasil
dipinjam!")
elif (jumlahPinjam <= 0):
    output("Peminjaman gagal! (jumlah pinjam harus > 0)")
    kondisiLanjut <- False
else:
    print("Peminjaman Gagal! jumlah barang di database kurang")
    kondisiLanjut <- False
else:
    print("Peminjaman Gagal! barang tidak ditemukan di database")
    kondisiLanjut <- False

if kondisiLanjut:
    pinjamGadget()

```

#### 9. F09:

{Program membaca file csv dan menerima pengembalian gadget yang telah dipinjam dari dalam database}

function infoBarang(string->array\_of\_string)->array\_of\_string

-- melakukan validasi input pengguna sesuai ketentuan string (S,A,B,C) dan mencocokkan item rarity dalam database}

function user\_list(string -> array of string) -> array of string

-- menampilkan informasi username dalam array string

function inventory(string)->string

-- menampilkan data item gadget yang dipinjam dalam array database  
 procedure kembalikanGadget (input:string: output:string)  
 -- menerima input dari user sesuai dengan permintaan program dan mengeluarkan informasi berdasarkan input dari user  
 {I.S. user menginput data nomor pengembalian dan memasukkan data item gadget serta tanggal pengembalian ke dalam sistem data}  
 (F.S. output berupa string memberi informasi validasi antara peminjaman dan pengembalian gadget}

function kembalikanGadget():

    # prosedur mengembalikan gadget ke gadget.csv dan mendaftarkan riwayatnya ke gadget\_return\_history.csv

    # KAMUS LOKAL

        # Variabel

        # dataKembali : string { string data pengembalian [ yang akan ditulis ke gadget\_return\_history.csv ]}

    # ALGORITMA

        # menambahkan riwayat pengembalian ke gadget\_return\_history.csv

        id\_peminjam <- infoUser(userAktif,'id')

        database <- \_gadgetReturnHistory

        id\_transaksi <- int(database[-1][0]) + 1

        idPeminjaman <- infoPinjam(id\_peminjam,'id')

        tanggalPengembalian <- datetime.datetime.now().strftime("%d/%m/%Y")

        id\_gadget <- infoPinjam(id\_peminjam,'id\_gadget')

        idPeminjaman <- infoPinjam(id\_peminjam,'id')

        if (infoKembali(idPeminjaman,'sisasisa') == infoPinjam(id\_peminjam,'jumlah\_pinjam')):

            sisasisa <- int(infoPinjam(id\_peminjam,'jumlah\_pinjam')) - jumlahPengembalian

        else:



```

    sisa = int(infoKembali(idPeminjaman,'sisa')) - jumlahPengembalian

    if (sisa < 0):

        print("Barang yang dikembalikan melebihi barang yang dipinjam")

        kondisiLanjut <- False

    else:

        dataKembali <-
        f"{id_transaksi};{idPeminjaman};{tanggalPengembalian};{jumlahPengembalian}
        ;{sisa}\n"

        gh <- open("gadget_return_history.csv","a")

        gh.write(dataKembali)

        gh.close()

        namaBarangPinjam <- infoBarang(id_gadget,"nama")

        output(f"\nItem {namaBarangPinjam} (x {jumlahPengembalian}) telah
        dikembalikan!")

        # mengubah jumlah barang yang tersedia [bertambah setelah
        pengembalian] pada gadget.csv

        modify_datas(_gadget,locIDinArray(_gadget,idPengembalian),3,int(infoBarang(id
        Pengembalian,'jumlah'))+jumlahPengembalian)

        overwrite_database('G')

```

#### # ALGORITMA UTAMA

```

userAktif = "bostang123" # user yang aktif akan disesuaikan dengan
fungsionalitas login [ F02 ]

arrayBarangPinjam = [] # array berisi barang yang dipinjam dan jumlah barang
yang dipinjam

arrayJumlahPinjam = [] # array berisi JUMLAH barang yang dipinjam

```

# perhatikan bahwa elemen arrayBarangPinjam dan arrayJumlahPinjam saling berkorespondensi

# mengakses database dalam bentuk array

g = open("gadget.csv","r")

u = open("user.csv","r")

gb = open("gadget\_borrow\_history.csv","r")

gr = open("gadget\_return\_history.csv","r")

raw\_lines\_g = g.readlines()

raw\_lines\_u = u.readlines()

raw\_lines\_gb = gb.readlines()

raw\_lines\_gr = gr.readlines()

g.close()

u.close()

gb.close()

gr.close()

lines\_g <- [raw\_line.replace("\n", "") for raw\_line in raw\_lines\_g]

lines\_u <- [raw\_line.replace("\n", "") for raw\_line in raw\_lines\_u]

lines\_gb <- [raw\_line.replace("\n", "") for raw\_line in raw\_lines\_gb]

lines\_gr <- [raw\_line.replace("\n", "") for raw\_line in raw\_lines\_gr]

\_gadget <- []

\_user <- []

```

_gadgetBorrowHistory <- []
_gadgetReturnHistory <- []

for line in lines_g:
    array_of_data <- splitList(line)
    _gadget.append(array_of_data)
for line in lines_u:
    array_of_data <- splitList(line)
    _user.append(array_of_data)
for line in lines_gb:
    array_of_data = splitList(line)
    _gadgetBorrowHistory.append(array_of_data)
for line in lines_gr:
    array_of_data <- splitList(line)
    _gadgetReturnHistory.append(array_of_data)

# skema pengembalian barang
output(">>> Mengembalikan Gadget\n")

# menampilkan inventori [ daftar barang yang telah dipinjam ]
Inventory(userAktif)
output("")

# validasi input pengembalian
kondisiLanjut <- False

```

```

nomorPengembalian <- int(input("Masukan nomor peminjaman: "))
if nomorPengembalian > 0 and nomorPengembalian <= len(arrayBarangPinjam):
    idPengembalian <- arrayBarangPinjam[nomorPengembalian -1]
    tanggal <- str(datetime.datetime.now().strftime("%d/%m/%Y"))
    tanggalPengembalian <- print("Tanggal pengembalian:",tanggal)
    jumlahPengembalian = int(input(f"Masukkan jumlah
    {infoBarang(idPengembalian,'nama')} yang mau dikembalikan: "))
    if jumlahPengembalian > arrayJumlahPinjam[nomorPengembalian-1]: # jangan
    lupa aritmatika indeks -1
        output("Pengembalian item gagal! Jumlah barang yang ingin dikembalikan
        lebih dari yang dipinjam")
    elif jumlahPengembalian <= 0:
        output("Pengembalian item gagal! Jumlah barang yang ingin dikembalikan
        tidak valid!")
    else: # input valid
        kondisiLanjut <- True
else:
    output("Nomor barang yang ingin dikembalikan tidak sesuai!")

if kondisiLanjut:
    kembalikanGadget()

```

#### 10. F10:

```

{Program membaca file csv dan meminta input dari pengguna untuk mengakses
item consumable dari dalam database}
function infoBarang(string->array_of_string)->array_of_string
-- melakukan validasi input pengguna sesuai ketentuan string (S,A,B,C) dan
mencocokkan item rarity dalam database}
function user_list(string -> array of string) -> array of string
-- menampilkan informasi username dalam array string

```

```

procedure minta_consum(input:string,output:string)
-- menerima input dari username sesuai ketentuan program dan mengakses
permintaan item consumable dari dalam database array
{I.S. user menginput jumlah permintaan dan tanggal permintaan ke dalam
database}
{F.S. output keluaran berupa string informasi jumlah permintaan item consumable
dari dalam array database}

```

```

function mintaConsum():

    # Prosedur untuk menuliskan informasi permintaan consumable ke data
    consumable_history.csv

    # proses ini dijalankan bila sudah divalidasi bahwa consumable BISA diminta

    # KAMUS LOKAL

    # ALGORITMA

    id_peminta = infoUser(userAktif,'id')

    database <- _consumableHistory

    id_minta <- int(database[-1][0]) + 1

    tanggalPeminta <- str(datetime.datetime.now().strftime("%d/%m/%Y"))

    # menambahkan riwayat peminjaman ke gadget_borrow_history.csv

    dataMinta <-
        f"{id_minta};{id_peminta};{idPeminta};{tanggalPeminta};{jumla
        hPermintaan}\n"

    c <- open("consumable_history.csv","a")

    c.write(dataMinta)

    c.close()

    # mengubah jumlah barang yang tersedia [berkurang setelah diminta] pada
    consumable.csv

```

```
modify_datas(_consumable,locIDinArray(_consumable,idPeminta),  
3,int(infoBarang(idPeminta,'jumlah'))-jumlahPermintaan)
```

```
overwrite_database('C')
```

```
# ALGORITMA UTAMA
```

```
userAktif <- "bostang123" # user yang sedang aktif [butuh penyesuaian dengan  
F02 ]
```

```
# asumsikan ini adalah user yang telah login
```

```
# membaca file csv database
```

```
u <- open("user.csv","r")
```

```
c <- open('consumable.csv','r')
```

```
ch <- open('consumable_history.csv','r')
```

```
raw_lines_u <- u.readlines()
```

```
raw_lines_c <- c.readlines()
```

```
raw_lines_ch <- ch.readlines()
```

```
u.close()
```

```
c.close()
```

```
ch.close()
```

```
lines_u <- [raw_line.replace("\n", "") for raw_line in raw_lines_u]
```

```
lines_c <- [raw_line.replace("\n", "") for raw_line in raw_lines_c]
```

```
lines_ch <- [raw_line.replace("\n", "") for raw_line in raw_lines_ch]
```

```
# mengakses database dalam bentuk array
```

```
_user <- []
```

```
_consumable <- []
```

```
_consumableHistory <- []
```

```
for line in lines_u:
```

```
    array_of_data <- splitList(line)
```

```
    _user.append(array_of_data)
```

```
for line in lines_c:
```

```
    array_of_data <- splitList(line)
```

```
    _consumable.append(array_of_data)
```

```
for line in lines_ch:
```

```
    array_of_data <- splitList(line)
```

```
    _consumableHistory.append(array_of_data)
```

```
clear_screen()
```

```
# melakukan proses peminjaman barang
```

```
output(">>> Meminta Consumable")
```

```
kondisiLanjut <- True
```

```
idPeminta <- input("Masukkan ID item: ")
```

```

if isIDinDatabase(idPeminta, _consumable): # permintaan barang HANYA DARI
    consumable.csv

    jumlahPermintaan <- int(input("Jumlah: "))

    tanggal <- str(datetime.datetime.now().strftime("%d/%m/%Y"))

    tanggalPeminta <- print("Tanggal permintaan:", tanggal)

    if (jumlahPermintaan <= int(infoBarang(idPeminta, 'jumlah'))):

        namaBarangPermintaan <- infoBarang(idPeminta, "nama")

        print(f"Item {namaBarangPermintaan} (x {jumlahPermintaan}) telah berhasil
            diambil!")

    elif (jumlahPermintaan <= 0):

        output("Permintaan gagal! (jumlah pengambilan harus > 0)")

        kondisiLanjut <- False

    else:

        output("Permintaan Gagal! jumlah barang di database kurang")

        kondisiLanjut <- False

    else:

        output("Permintaan Gagal! barang tidak ditemukan di database")

        kondisiLanjut <- False

if kondisiLanjut:

    mintaConsum()

```

#### 11. F11:

```

{Program membaca file csv dan user menginput perintah sesuai ketentuan
program dan menampilkan riwayat peminjaman item gadget dalam database}
function infoBarang(string->array_of_string)->array_of_string
-- melakukan validasi input pengguna sesuai ketentuan string (S,A,B,C) dan
mencocokkan item rarity dalam database}
function infoPinjam(string)->array_of_string

```



```
-- mencari informasi terkait dengan item gadget yang dipinjam dari dalam
database
procedure riwayat_pinjam(output:list_of_string)
-- menampilkan list data rekam jejak peminjaman item gadget dari database
{I.S. user me-run program TampilRiwayat}
{F.S. output ditampilkan berupa list_of_string data riwayat peminjaman suatu
item dalam database}
```

#### #REALISASI FUNGSI

```
output(">>> Riwayat Kembali")

database <- _gadgetBorrowHistory

else if (len(database) < 6):

    m <- 1

else:

    m <- len(database) - 5

n <- len(database)

iterate baris in reversed(database[m:n]):

    output(f"\nID Peminjaman      : {infoPinjam('id')}")

    idPeminjam <- infoPinjam('id_peminjam')

    if isUserinDatabase(idPeminjam,_user) and infoUser(idPeminjam,'id'):

        output(f>Nama Pengambil      : {infoUser(idPeminjam,'nama')}")

    idGadget <- infoPinjam('id_gadget')

    if isIDinDatabase(idGadget,_gadget) and infoBarang(idGadget,'id'):

        output(f>Nama Gadget          : {infoBarang(idGadget,'nama')}")

    output(f">Tanggal Peminjaman : {infoPinjam('tanggal_peminjaman')}")

    output(f">Jumlah                : {infoPinjam('jumlah_pinjam')}")

i = 0

while (m > 1):
```

```

lanjut <- input("\nApakah Anda mau ke halaman selanjutnya?(Y/N)")
while (lanjut != 'Y') and (lanjut != 'y') and (lanjut != 'N') and (lanjut != 'n'):
    output("Masukan Anda salah!")
    lanjut <- input("\nApakah Anda mau ke halaman selanjutnya?(Y/N)")
if (lanjut == 'Y') or (lanjut == 'y'):
    if (m > 6):
        m -= 5
    else:
        m = 1
    n -= 5

clear_screen()

for baris in reversed(database[m:n]):
    output(">>> Riwayat Kembali")
    output(f"\nID Peminjaman      : {infoPinjam('id')}")
    idPeminjam <- infoPinjam('id_peminjam')
    if isUserinDatabase(idPeminjam,_user) and infoUser(idPeminjam,'id'):
        output(f>Nama Pengambil      : {infoUser(idPeminjam,'nama')}")
    idGadget <- infoPinjam('id_gadget')
    if isIDinDatabase(idGadget,_gadget) and infoBarang(idGadget,'id'):
        output(f>Nama Gadget          : {infoBarang(idGadget,'nama')}")
    output(f>Tanggal Peminjaman : {infoPinjam('tanggal_peminjaman')}")
    output(f>Jumlah                : {infoPinjam('jumlah_pinjam')}")
else:

```

```
i = i + 1
```

```
else:
```

```
output("\nSemua riwayat sudah ditampilkan")
```

```
output("")
```

```
os.system('pause')
```

## 12. F12:

{Program membaca file csv dan menampilkan data rekam jejak pengembalian item gadget yang telah dipinjam}

function infoBarang(string->array\_of\_string)->array\_of\_string

-- melakukan validasi input pengguna sesuai ketentuan string (S,A,B,C) dan mencocokkan item rarity dalam database}

function infoKembali(string)->array\_of\_string

-- mencari informasi terkait dengan item gadget yang dikembalikan ke dalam database

procedure tampil\_riwayat\_kembali(output:lst\_of\_string)

-- menampilkan riwayat berupa list data pengembalian gadget ke dalam database

{I.S. user me-run program TampilKembali}

{F.S. output ditampilkan berupa list\_of\_integer data riwayat pengembalian gadget dari dalam database}

#REALISASI FUNGSI

```
output(">>> Riwayat Kembali")
```

```
database <- _gadgetBorrowHistory
```

```
else if (len(database) < 6):
```

```
    m <- 1
```

```
else:
```

```
    m <- len(database) - 5
```

```
n <- len(database)
```

```
iterate baris in reversed(database[m:n]):
```

```
    output(f"\nID Peminjaman      : {infoPinjam('id')}")
```

```

idPeminjam <- infoPinjam('id_peminjam')
if isUserinDatabase(idPeminjam,_user) and infoUser(idPeminjam,'id'):
    output(f>Nama Pengambil      : {infoUser(idPeminjam,'nama')})")
idGadget <- infoPinjam('id_gadget')
if isIDinDatabase(idGadget,_gadget) and infoBarang(idGadget,'id'):
    output(f>Nama Gadget        : {infoBarang(idGadget,'nama')})")
output(f>Tanggal Peminjaman   : {infoPinjam('tanggal_peminjaman')})")
output(f>Jumlah                : {infoPinjam('jumlah_pinjam')})")
i = 0
while (m > 1):
    lanjut <- input("\nApakah Anda mau ke halaman selanjutnya?(Y/N)")
    while (lanjut != 'Y') and (lanjut != 'y') and (lanjut != 'N') and (lanjut != 'n'):
        output("Masukan Anda salah!")
        lanjut <- input("\nApakah Anda mau ke halaman selanjutnya?(Y/N)")
    if (lanjut == 'Y') or (lanjut == 'y'):
        if (m > 6):
            m -= 5
        else:
            m = 1
        n -= 5

clear_screen()

for baris in reversed(database[m:n]):
    output(">>> Riwayat Kembali")

```

```

        output(f"\nID Peminjaman      : {infoPinjam('id')}")
idPeminjam <- infoPinjam('id_peminjam')
if isUserinDatabase(idPeminjam,_user) and infoUser(idPeminjam,'id'):
    output(f"Nama Pengambil      : {infoUser(idPeminjam,'nama')}")
idGadget <- infoPinjam('id_gadget')
if isIDinDatabase(idGadget,_gadget) and infoBarang(idGadget,'id'):
    output(f"Nama Gadget          : {infoBarang(idGadget,'nama')}")
    output(f"Tanggal Peminjaman : {infoPinjam('tanggal_peminjaman')}")
    output(f"Jumlah              : {infoPinjam('jumlah_pinjam')}")
else:
    i = i + 1
else:
    output("\nSemua riwayat sudah ditampilkan")
output("")
os.system('pause')

output(">>> Riwayat Kembali")
database <- _gadgetReturnHistory
if (len(database) < 6):
    m <- 1
else:
    m <- len(database) - 5
n <- len(database)
for baris in reversed(database[m:n]):
    output(f"\nID Pengembalian      : {infoKembali('id')}")

```

```

idPeminjaman <- infoKembali('id_peminjaman')

if isPinjaminDatabase(idPeminjaman,_gadgetBorrowHistory) and
infoPinjam(idPeminjaman,'id_peminjam'):

    idPeminjam <- infoPinjam(idPeminjaman,'id_peminjam')

    if isUserinDatabase(idPeminjam,_user) and infoUser(idPeminjam,'id'):

        output(f>Nama Pengambil      : {infoUser(idPeminjam,'nama')}")

    if isPinjaminDatabase(idPeminjaman,_gadgetBorrowHistory) and
infoPinjam(idPeminjaman,'id_gadget'):

        idGadget <- infoPinjam(idPeminjaman,'id_gadget')

        if isIDinDatabase(idGadget,_gadget) and infoBarang(idGadget,'id'):

            output(f>Nama Gadget      : {infoBarang(idGadget,'nama')}")

        output(f>Tanggal Pengembalian : {infoKembali('tanggal_pengembalian')}")

while (m > 1):

    lanjut <- input("\nApakah Anda mau ke halaman selanjutnya?(Y/N)")

    while (lanjut != 'Y') and (lanjut != 'y') and (lanjut != 'N') and (lanjut != 'n') do:

        output("Masukan Anda salah!")

        lanjut <- input("\nApakah Anda mau ke halaman selanjutnya?(Y/N)")

    if (m > 6):

        m -= 5

    else:

        m <- 1

    n -= 5

clear_screen()

```

```

for baris in reversed(database[m:n]):

    output(f"\nID Pengembalian      : {infoKembali('id')}")

    idPeminjaman <- infoKembali('id_peminjaman')

    if isPinjaminDatabase(idPeminjaman,_gadgetBorrowHistory) and
infoPinjam(idPeminjaman,'id_peminjam'):

        idPeminjam <- infoPinjam(idPeminjaman,'id_peminjam')

        if isUserinDatabase(idPeminjam,_user) and infoUser(idPeminjam,'id'):

            output(f>Nama Pengambil      : {infoUser(idPeminjam,'nama')}")

        if isPinjaminDatabase(idPeminjaman,_gadgetBorrowHistory) and
infoPinjam(idPeminjaman,'id_gadget'):

            idGadget <- infoPinjam(idPeminjaman,'id_gadget')

            if isIDinDatabase(idGadget,_gadget) and infoBarang(idGadget,'id'):

                output(f>Nama Gadget      : {infoBarang(idGadget,'nama')}")

            output(f>Tanggal Pengembalian :
{infoKembali('tanggal_pengembalian')}")

        else:

            i <- i + 1

    else:

        output("\nSemua riwayat sudah ditampilkan")

output("")

os.system('pause')

```

### 13. F13:

{Program membaca file csv dan menampilkan data rekam jejak pengambilan item consumable dari dalam database}

```

function infoBarang(string->array_of_string)->array_of_string
-- melakukan validasi input pengguna sesuai ketentuan string (S,A,B,C) dan
mencocokkan item rarity dalam database}
function infoAmbil(string)->array_of_string

```

```
-- mencari informasi terkait dengan item consumable yang diminta dari dalam database
Procedure tampil_consumable(output:list_of_string)
-- menampilkan data riwayat permintaan item consumable dari dalam database
{I.S. user me-run program TampilConsumable}
{F.S. output ditampilkan berupa list_of_integer data riwayat permintaan item consumable dari dalam database}
```

```
output(">>> Riwayat Ambil")
```

```
database <- _consumableHistory
```

```
if (len(database) < 6):
```

```
    m <- 1
```

```
else:
```

```
    m <- len(database) - 5
```

```
n = len(database)
```

```
for baris in reversed(database[m:n]):
```

```
    output(f"\nID Pengambilan    : {infoAmbil('id')}")
```

```
    idPengambil <- infoAmbil('id_pengambil')
```

```
    if isUserinDatabase(idPengambil,_user) and infoUser(idPengambil,'id'):
```

```
        output(f>Nama Pengambil      : {infoUser(idPengambil,'nama')}")
```

```
    idConsumable <- infoAmbil('id_consumable')
```

```
    if isIDinDatabase(idConsumable,_consumable) and
infoBarang(idConsumable,'id'):
```

```
        output(f>Nama Consumable    : {infoBarang(idConsumable,'nama')}")
```

```
    output(f"Tanggal Peminjaman : {infoAmbil('tanggal_pengambilan')}")
```

```
    output(f"Jumlah              : {infoAmbil('jumlah')}")
```

```
while (m > 1):
```

```
    lanjut <- input("\nApakah Anda mau ke halaman selanjutnya?(Y/N)")
```



```

while lanjut not in 'YyNn':

    output("Masukan Anda salah!")

    lanjut <- input("\nApakah Anda mau ke halaman selanjutnya?(Y/N)")

if (lanjut == 'Y') or (lanjut == 'y'):

    if (m > 6):

        m -= 5

    else:

        m <- 1

    n -= 5

clear_screen()

for baris in reversed(database[m:n]):

    output(f"\nID Pengambilan    : {infoAmbil('id')}")

    idPengambil <- infoAmbil('id_pengambil')

    if isUserinDatabase(idPengambil,_user) and infoUser(idPengambil,'id'):

        print(f>Nama Pengambil    : {infoUser(idPengambil,'nama')}")

        idConsumable <- infoAmbil('id_consumable')

        if isIDinDatabase(idConsumable,_consumable) and
infoBarang(idConsumable,'id'):

            output(f>Nama Consumable    : {infoBarang(idConsumable,'nama')}")

            output(f>Tanggal Pengambilan : {infoAmbil('tanggal_pengambilan')}")

            output(f>Jumlah              : {infoAmbil('jumlah')}")

        else:

            I <- I + 1

    else:

```

```

        output("\nSemua riwayat sudah ditampilkan")

print("")

os.system('pause')

```

#### 14. F14:

{Program untuk menjalankan pertama kali dan me-*load* folder dalam sistem simulator}

#REALISASI FUNGSI

function load():

# Untuk mencari file di folder tersebut

```

parser <- argparse.ArgumentParser(usage='python kantongajaib.py
<folder_name>')

```

```

parser.add_argument('folder_name', help="Masukkan nama folder tempat
penyimpanan", default=None)

```

```

args <- parser.parse_args()

```

if args.folder\_name is not None:

try:

```

if __name__ == "__main__":

```

```

    for (root,dirs,files) in os.walk(args.folder_name, topdown=True):

```

```

        path <- root

```

```

        nama_file <- files

```

except SyntaxError:

```

    output('Tidak ada nama folder yang diberikan!')

```

```

    output("Usage : python kantongajaib.py <nama_folder>")

```

else:

```

    output('Tidak ada nama folder yang diberikan!') # In case kepace (harusnya
    engga)

```

```

    output("Usage : python kantongajaib.py <nama_folder>")

```

# Untuk mengubah file CSV menjadi array yg bisa dipakai di fungsi lain

try:

```
c <- open(path+"/"+nama_file[0],"r")
ch <- open(path+"/"+nama_file[1],"r")
g <- open(path+"/"+nama_file[2],"r")
gb <- open(path+"/"+nama_file[3],"r")
gr <- open(path+"/"+nama_file[4],"r")
u <- open(path+"/"+nama_file[5],"r")

raw_lines_c <- c.readlines()
raw_lines_ch <- ch.readlines()
raw_lines_g <- g.readlines()
raw_lines_gb <- gb.readlines()
raw_lines_gr <- gr.readlines()
raw_lines_u <- u.readlines()

c.close()
ch.close()
g.close()
gb.close()
gr.close()
u.close()

lines_c <- [raw_line.replace("\n", "") for raw_line in raw_lines_c]
lines_ch <- [raw_line.replace("\n", "") for raw_line in raw_lines_ch]
lines_g <- [raw_line.replace("\n", "") for raw_line in raw_lines_g]
lines_gb <- [raw_line.replace("\n", "") for raw_line in raw_lines_gb]
```

```

lines_gr <- [raw_line.replace("\n", "") for raw_line in raw_lines_gr]
lines_u <- [raw_line.replace("\n", "") for raw_line in raw_lines_u]

_consumable <- []
_consumableHistory <- []
_gadget <- []
_gadgetBorrow <- []
_gadgetReturn <- []
_user <- []

for line in lines_c:
    array_of_data <- splitList(line)
    _consumable.append(array_of_data)

for line in lines_ch:
    array_of_data <- splitList(line)
    _consumableHistory.append(array_of_data)

for line in lines_g:
    array_of_data <- splitList(line)
    _gadget.append(array_of_data)

for line in lines_gb:
    array_of_data <- splitList(line)
    _gadgetBorrow.append(array_of_data)

for line in lines_gr:
    array_of_data <- splitList(line)
    _gadgetReturn.append(array_of_data)

for line in lines_u:
    array_of_data <- splitList(line)

```

```

        _user.append(array_of_data)

# Menyimpan array dari file csv untuk dipakai di fungsi lain
except UnboundLocalError:

    output('Tidak ada nama folder yang diberikan!')

    output("Usage : python kantongajaib.py <nama_folder>")

```

#### 15. F15:

{ Program menyimpan semua data input dari user dan menyimpan dalam file berformat csv }

function simpan(input: array\_of\_string, output: write: file.csv)  
 -- menyimpan data input dari pengguna ke dalam file csv

```

function
simpan(_consumable,_consumableHistory,_gadget,_gadgetBorrow,_gadgetReturn
,_user):

```

```

    folder <- str(input("Masukkan nama folder penyimpanan : "))

```

```

    parent_dir <- os.getcwd()

```

```

    print(parent_dir)

```

```

    path <- os.path.join(parent_dir, folder)

```

```

    try:

```

```

        os.mkdir(path)

```

```

        c <- open(os.path.join(path, "consumable.csv"),"w+")

```

```

        ch <- open(os.path.join(path, "consumable_history.csv"),"w+")

```

```

        g <- open(os.path.join(path, "gadget.csv"),"w+")

```

```

        gb <- open(os.path.join(path, "gadget_borrow_history.csv"),"w+")

```

```

        gr <- open(os.path.join(path, "gadget_return_history.csv"),"w+")

```

```
u <- open(os.path.join(path, "user.csv"), "w+")
c.write(convert_data_to_string(_consumable))
ch.write(convert_data_to_string(_consumableHistory))
g.write(convert_data_to_string(_gadget))
gb.write(convert_data_to_string(_gadgetBorrow))
gr.write(convert_data_to_string(_gadgetReturn))
u.write(convert_data_to_string(_user))
c.close()
ch.close()
g.close()
gb.close()
gr.close()
u.close()
```

except FileExistsError:

```
c <- open(os.path.join(path, "consumable.csv"), "w")
ch <- open(os.path.join(path, "consumable_history.csv"), "w")
g <- open(os.path.join(path, "gadget.csv"), "w")
gb <- open(os.path.join(path, "gadget_borrow_history.csv"), "w")
gr <- open(os.path.join(path, "gadget_return_history.csv"), "w")
u <- open(os.path.join(path, "user.csv"), "w")
c.write(convert_data_to_string(_consumable))
ch.write(convert_data_to_string(_consumableHistory))
g.write(convert_data_to_string(_gadget))
gb.write(convert_data_to_string(_gadgetBorrow))
gr.write(convert_data_to_string(_gadgetReturn))
```

```

u.write(convert_data_to_string(_user))

c.close()

ch.close()

g.close()

gb.close()

gr.close()

u.close()

```

#### 16. F16:

```

{Program menampilkan command bantuan untuk mengakses program dalam
simulator}
procedure help(output:string)
-- menampilkan teks perintah untuk mengakses tiap program dalam sistem
simulator
{I.S. user me-run program Help}
{F.S. output ditampilkan berupa text string perintah dan spesifikasinya}

```

function help(role):

# KAMUS LOKAL

# ALGORITMA

#Akses : admin

if role == 'admin':

output('COMMAND TEXT HELP UNTUK AKSES ADMIN')

output('===== HELP =====')#tampilan help untuk akses  
admin

output('register - untuk melakukan registrasi user baru')

output('login - untuk melakukan login ke dalam sistem')

output('tambahitem - untuk melakukan penambahan item')

```

        output('carirarity - untuk mencari item berdasarkan tingkat rarity suatu item')

        output('caritahun - untuk mencari item berdasarkan input tahun yang
diberikan')

        output('hapusitem - untuk menghapus suatu item dari database')

        output('ubahjumlah - untuk mengubah jumlah suatu item dalam database')

        output('riwayatpinjam - untuk menampilkan riwayat peminjaman suatu item
dalam gadget')

        output('riwayatkembali - untuk menampilkan riwayat pemngembalian suatu
item dalam gadget')

        print('riwayatambil - untuk menampilkan riwayat pengambilan consumable
suatu item dalam database')

        output('save - untuk menyimpan data ke dalam suatu file setelah diubah')

        output('help - untuk menampilkan command sesuai kebutuhan user/admin')

```

#Akses : user

```
elif role == 'user':
```

```

        output('COMMAND TEXT HELP UNTUK AKSES USER')

        output('===== HELP =====')#tampilan help untuk akses
user

        output('login - untuk melakukan login ke dalam sistem')

        output('carirarity - untuk mencari item berdasarkan tingkat rarity suatu item')

        output('caritahun - untuk mencari item berdasarkan input tahun yang
diberikan')

        output('pinjam - untuk meminjam suatu item gadget dalam database')

        output('kembalikan - untuk mengembalikan suatu item gadget yang
dipinjam')

        output('minta - untuk memingingta consumable yang tersedia dalam database')

        output('save - untuk menyimpan data ke dalam suatu file setelah diubah')

```



```
output('help - untuk menampilkan command sesuai kebutuhan user/admin')
```

```
# ALGORITMA UTAMA
```

```
user_aktif <- input() #### harus disesuaikan dengan pengguna aktif di F02
```

```
# mengakses database user
```

```
u <- open("user.csv","r")
```

```
raw_lines_u <- u.readlines()
```

```
u.close()
```

```
lines_u <- [raw_line.replace("\n", "") for raw_line in raw_lines_u]
```

```
_user <- []
```

```
for line in lines_u:
```

```
    array_of_data <- convert_line_to_data(line)
```

```
    _user.append(array_of_data)
```

```
help(infoUser(user_aktif,'role'))
```

17. F17:

```
{Program mengkonfirmasi masukan pengguna untuk keluar dari program}
```

```
function keluar(array_of_string -> array_of_string -> array_of_string->
```

```
array_of-string -> array_of_string) -> string
```

```
-- menampilkan validasi input dari pengguna untuk mengakhiri aktivitas di sistem  
inventarisasi gadget simulator
```

```
{I.S. user memasukkan input untuk memvalidasi exit}
```

```
{F.S. output menampilkan informasi terkait masukan user dan aktivitas simulator  
dihentikan}
```

```
function
```

```
keluar(_consumable,_consumableHistory,_gadget,_gadgetBorrow,_gadgetRetur  
n,_user):
```

```

loop <- True

while loop do:

    clear_screen()

    save <- str(input("Apakah Anda mau melakukan penyimpanan file yang
sudah diubah? (y/n) "))

    if (save == 'y') or (save == 'Y'):

        simpan(_consumable,_consumableHistory,_gadget,_gadgetBorrow,_gadgetRetur
n,_user)

        exit()

    elif (save == 'n') or (save == 'N'):

        exit()

    else:

        output("Masukkan salah. Mohon ulangi")

        os.system('pause')

```

## 7. SPESIFIKASI UNTUK TIAP MODUL/PROSEDUR/FUNGSI

- Spesifikasi prosedur/modul/ fungsi utama program
  1. Fungsi clear\_screen
    - {Fungsi untuk menghapus tampilan text di program untuk merapikan kode ketika di\_run}
  2. Fungsi split\_list
    - {Fungsi untuk memisahkan kalimat di dalam array dengan delimiter (;)}
  3. Fungsi convert\_line\_to\_data
    - {Fungsi untuk mengubah baris menjadi sebuah data array yang dapat diakses}
  4. Fungsi convert\_array\_to\_real\_value
    - {Fungsi untuk mengubah string di array menjadi data real sesuai type data}
- Spesifikasi prosedur/modul/program untuk fungsi program lanjutan
  1. Fungsi add\_item\_to\_database
    - {Fungsi untuk memasukkan input pengguna ke dalam array database yang dapat diakses}

2. Fungsi user\_list  
{Fungsi untuk mengakses data user di dalam suatu array database}
3. Fungsi register  
{Fungsi untuk menerima input informasi dari pengguna ke dalam database array}
4. Fungsi password\_list  
{Fungsi untuk mengakses data password pengguna dalam suatu array}
5. Fungsi id\_user  
[Fungsi untuk menampilkan id user dalam database array]
6. Fungsi login  
{Fungsi utama untuk masuk ke dalam sistem inventarisasi gadget}
7. Fungsi IsRarityInDatabase  
{Fungsi untuk memvalidasi elemen rarity di dalam array database}
8. Fungsi infoBarang  
{Fungsi untuk menampilkan elemen-elemen yang ada dalam suatu array database}
9. Procedure cariRarity  
{Program untuk menampilkan list data dalam array sesuai dengan input rarity dari user}
10. Fungsi IsTahunInDatabase  
{Fungsi untuk memvalidasi dan menampilkan tahun sesuai dengan input pengguna}
11. Procedure cariTahun  
{Program untuk menampilkan list data item gadget sesuai input tahun dari pengguna}
12. Fungsi IsIDInDatabase  
[Fungsi untuk memvalidasi id dari pengguna di dalam array database]
13. Fungsi tambahItem  
{Fungsi untuk menambah item gadget ke dalam suatu database array}
14. Fungsi hapusItem  
{Fungsi untuk menghapus data item gadget dari dalam database}
15. Fungsi ubahJumlah:  
{Fungsi untuk mengubah jumlah item gadget dalam array database}
16. Fungsi pinjamGadget  
{Fungsi untuk meminjam jumlah gadget dalam array database}
17. Procedure KembalikanGadget  
{Fungsi untuk mengembalikan item gadget dalam jumlah tertentu}
18. Fungsi mintaConsum  
{Fungsi untuk meminta item consumable dalam array database}
19. Fungsi riwayatPinjam

- {Fungsi untuk menampilkan data riwayat peminjaman gadget dalam database array}
- 20. Fungsi riwayatKembali  
{Fungsi untuk menampilkan riwayat pengembalian gadget dalam array database}
- 21. Fungsi riwayatAmbil  
{Fungsi untuk menampilkan riwayat pengambilan consumable dalam array database}
- 22. Fungsi Load  
{Fungsi untuk menampilkan loading folder sebagai akses sistem pertama kali}
- 23. Fungsi save  
{Fungsi untuk menyimpan semua input dari user ke dalam database array}
- 24. Fungsi Help  
{Fungsi untuk menampilkan perintah bantuan sebagai akses fungsi di simulator}
- 25. Fungsi exit  
[Fungsi untuk mengakhiri aktivitas di simulator]

## 8. SCREENSHOT HASIL PENGUJIAN PROGRAM

- F14:

```
PS G:\Kuliah~\SEMESTER 2\DasPro (10)\Coding\Tugas Besar\testboi> python kantongajaib.py CSV
loading...
Selamat Datang di 'Kantong Ajaib'!
Press any key to continue . . .
```

- F02;

```
Masukkan username:ryujoshi
Masukkan password:joshi456
Halo ryujoshi ! Selamat datang di Kantong Ajaib.
Press any key to continue . . .
```

- F01:

```
Masukkan nama:Budi
Masukkan username:BudiDoremi
Masukkan password:password123
Masukkan alamat:Jl. Gamma no 9
User BudiDoremi telah berhasil register ke dalam Kantong Ajaib!
Press any key to continue . . .
```

- F03:

```
Masukkan rarity: B

Hasil pencarian:

Nama          : Kalkulator
Deskripsi     : Alat yang dapat menolongmu ketika ujian Kimia
Jumlah        : 100
Rarity        : B
Tahun ditemukan : 1879

Nama          : Tissue Basah
Deskripsi     : Dapat melembabkan tubuh
Jumlah        : 44
Rarity        : B
Tahun ditemukan : 1990

Semua data telah ditampilkan

Press any key to continue . . .
```

- F04:

```
>>> Pencarian Gadget berdasarkan rarity

Masukkan tahun: 2020
Masukkan kategori: >

Hasil pencarian:

Nama          : Pintu ke ITB
Deskripsi     : Siapapun yang masuk ke dalam pintu ini akan diteleport ke kolam Intel
Jumlah        : 12
Rarity        : A
Tahun ditemukan : 2112

Semua data telah ditampilkan

Press any key to continue . . .
```

- F05:

```
>>> Menambah Item
Masukan ID: G10
Masukan Nama: handphone emas
Masukan Deskripsi: handphone yang sangat mahal untuk flexing
Masukan Jumlah: 2
Masukan Rarity: s
Masukan tahun ditemukan: 2020

Item telah berhasil ditambahkan ke database

Press any key to continue . . . █
```

- F06;

```
>>> Menghapus Item
Masukkan ID item: g1
Apakah Anda yakin ingin menghapus Pintu ke ITB(Y/N)? y

Item telah berhasil dihapus dari database
Press any key to continue . . . █
```

- F07:

```
>>> Mengubah Jumlah Gadget atau Consumable
Masukkan ID: g1
Masukkan jumlah: 25

25 Pintu ke ITB berhasil ditambahkan. Stok sekarang: 37

Press any key to continue . . . █
```

- F08:

```
>>> Meminjam Gadget

Masukkan ID item: G2
Tanggal peminjaman: 03/05/2021
Jumlah peminjaman: 10
Item Durian montok (x10) berhasil dipinjam!

Press any key to continue . . . █
```

- F09:

```
>>> Mengembalikan Gadget

1. Tissue Basah          || jumlah : 2
2. Durian montok         || jumlah : 3

Masukan nomor peminjaman: 1
Tanggal pengembalian: 03/05/2021
Masukkan jumlah Tissue Basah yang mau dikembalikan: 2

Item Tissue Basah (x2) telah dikembalikan. Sisa di inventory berjumlah 0.
Press any key to continue . . .
```

- F10:

```
>>> Meminta Consumable

Masukkan ID item: C3
Jumlah: 8
Tanggal permintaan: 03/05/2021

Item Air Super (x8) telah berhasil diambil!

Press any key to continue . . .
```

- F11:

>>> Riwayat Kembali

ID Peminjaman : 6  
Nama Pengambil : Bostang  
Nama Gadget : Pintu ke ITB  
Tanggal Peminjaman : 01/05/2021  
Jumlah : 3

ID Peminjaman : 5  
Nama Pengambil : Joshi  
Nama Gadget : Kalkulator  
Tanggal Peminjaman : 26/04/2021  
Jumlah : 8

ID Peminjaman : 4  
Nama Pengambil : Daris  
Nama Gadget : Tissue Basah  
Tanggal Peminjaman : 26/04/2021  
Jumlah : 5

ID Peminjaman : 3  
Nama Pengambil : Diffa' Shada  
Nama Gadget : Kalkulator  
Tanggal Peminjaman : 26/04/2021  
Jumlah : 10

ID Peminjaman : 2  
Nama Pengambil : Joshi  
Nama Gadget : Pintu ke ITB  
Tanggal Peminjaman : 25/04/2021  
Jumlah : 5

Apakah Anda mau ke halaman selanjutnya?(Y/N) ☐

- F12;



```
>>> Riwayat Kembali

ID Pengembalian      : 2
Nama Pengambil       : Bostang
Nama Gadget          : Durian montok
Tanggal Pengembalian : G2

ID Pengembalian      : 1
Nama Pengambil       : Bostang
Nama Gadget          : Durian montok
Tanggal Pengembalian : G2

Semua riwayat sudah ditampilkan

Press any key to continue . . . █
```

- F13:

```
>>> Riwayat Ambil

ID Pengambilan      : 4
Nama Pengambil      : Bostang
Nama Consumable     : Permen Pintar
Tanggal Peminjaman  : 01/05/2021
Jumlah              : 1

ID Pengambilan      : 3
Nama Pengambil      : Diffa' Shada
Nama Consumable     : Air Super
Tanggal Peminjaman  : 26/04/2021
Jumlah              : 104

ID Pengambilan      : 2
Nama Pengambil      : Joshi
Nama Consumable     : Permen Pintar
Tanggal Peminjaman  : 25/04/2021
Jumlah              : 5

ID Pengambilan      : 1
Nama Pengambil      : Bostang
Nama Consumable     : Pen Tablet
Tanggal Peminjaman  : 17/04/2021
Jumlah              : 69

Semua riwayat sudah ditampilkan

Press any key to continue . . . █
```

## LAMPIRAN FORM ASISTENSI

- Form Asistensi:

Nomor Asistensi : 1

.....

No. Kelompok/Kelas : Kelompok 11/ Kelas 10

.....

---

Tanggal asistensi : 10 April 2021

---

Anggota kelompok

NIM / Nama (Hanya yang Hadir)	
1	16520070/ Diffa' Shada 'Aqila
2	16520090/Bostang Palaguna
3	16520170/ Muhammad Daris Nurhakim
4	16520230/ Joshi Ryu Setiadi
5	
6	

Asisten pembimbing

NIM / Nama

13518016/ Indra Febrio Nugroho

Catatan Asistensi: Rekomendasi run program menggunakan git version control system

Rangkuman Diskusi

1. Module argparse: modul untuk mengambil nama file dari command line file python
2. Validasi input fungsi consumable dan rarity
3. Input nama user harus memperhatikan kapitalisasi  
Seperti : rOjaP SupRiyadi, yang disimpan: Rojap Supriyadi
4. Interface pyqT batal digunakan
5. Deskripsi nama dan kapitalisasi boleh mirip,tetapi id harus berbeda

Tindak Lanjut

1. Mempelajari python library, seperti: os,sys,math,time,argparse
2. Memperhatikan kapitalisasi input nama
3. Validasi input consumable dan rarity:

Misal: input consumable dan raritynya berbeda (tidak ada dengan kategori) , harus dicegah

Nomor Asistensi : 2

No. Kelompok/Kelas : Kelompok 11/ Kelas 10

.....

Tanggal asistensi : 28 April 2021

.....

Anggota kelompok

NIM / Nama (Hanya yang Hadir)	
1	16520070/ Diffa' Shada 'Aqila
2	16520090/Bostang Palaguna
3	16520170/ Muhammad Daris Nurhakim
4	16520230/ Joshi Ryu Setiadi
5	
6	

Asisten pembimbing

NIM / Nama

13518016/ Indra Febrio Nugroho

Catatan Asistensi: Fungsi import boleh digunakan antar tiap fungsi internal

Rangkuman Diskusi
-------------------

1. Menyelesaikan Fungsi Load, save dan exit
2. Fungsi load disimpan di satu file directory tertentu, boleh satu folder dengan source code
3. Fungsi save disimpan di dalam array, jangan di overwrite langsung ke file csv
4. Data riwayat diurutkan dari bawah sebagai asumsi data terbaru yang diinput user

Tindak Lanjut

1. Pelajari lagi module library argparse untuk me-load folder
2. Fungsi login di cek karena crucial, diperiksa untuk role user dan admin
3. Fungsi is\_returned tetap dipakai untuk memvalidasi pengembalian item oleh user