

TUGAS BESAR
IF1210 DASAR PEMROGRAMAN
KELAS 09



Disusun Oleh:

Kelompok 11

Farisa Aliya	16520129
Gerald Abraham Sianturi	16520149
Malik Akbar Hashemi Rafsanjani	16520299
Hughie Raymonelika Manggala	16520329

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 IF1210 Dasar Pemrograman Semester 2 2020/2021.”

Farisa Aliya	16520129
Gerald Abraham Sianturi	16520149
Malik Akbar Hashemi Rafsanjani	16520299
Hughie Raymonelika Manggala	16520329

DAFTAR ISI

HALAMAN PERNYATAAN KELOMPOK	2
DAFTAR ISI	3
DAFTAR TABEL	4
DAFTAR GAMBAR	5
DESKRIPSI PERSOALAN	6
DAFTAR PEMBAGIAN TUGAS	9
CHECKLIST HASIL RANCANGAN, IMPLEMENTASI, DAN UJI COBA	11
DESAIN PERINTAH	12
DESAIN KAMUS DATA	16
DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL PROGRAM	24
SPESIFIKASI	32
HASIL PENGUJIAN PROGRAM	66
CATATAN DAN LAMPIRAN	73

DAFTAR TABEL

Tabel 01: Daftar Pembagian Tugas Pada Source Code	9
Tabel 02: Daftar Pembagian Tugas Pada Laporan	10
Tabel 03: Daftar Checklist Hasil Rancangan, Implementasi dan Testing Setiap Primitif	11
Tabel 04: Hasil Pengujian Program	66

DAFTAR GAMBAR

Gambar 01 : Desain Dekomposisi F01 Register	24
Gambar 02 : Desain Dekomposisi F02 Login	24
Gambar 03 : Desain Dekomposisi F03 Pencarian Gadget Berdasarkan Rarity	25
Gambar 04 : Desain Dekomposisi F04 Pencarian Gadget Berdasarkan Tahun	25
Gambar 05 : Desain Dekomposisi F05 Menambah Item	25
Gambar 06 : Desain Dekomposisi F06 Menghapus Gadget atau Consumable	26
Gambar 07 : Desain Dekomposisi F07 Mengubah Jumlah Gadget	26
Gambar 08 : Desain Dekomposisi F08 Meminjam Gadget	27
Gambar 09 : Desain Dekomposisi F09 + FB 02 Mengembalikan Gadget	27
Gambar 10 : Desain Dekomposisi F10 Meminta Consumable	28
Gambar 11 : Desain Dekomposisi F11 Melihat Riwayat Peminjaman Gadget	28
Gambar 12 : Desain Dekomposisi F12 Melihat Riwayat Pengembalian Gadget	28
Gambar 13 : Desain Dekomposisi F13 Melihat Riwayat Pengambilan Consumable	29
Gambar 14 : Desain Dekomposisi F14 Load Data	29
Gambar 15 : Desain Dekomposisi F15 Save Data	30
Gambar 16 : Desain Dekomposisi F16 Help	30
Gambar 17 : Desain Dekomposisi F17 Exit	30
Gambar 18 : Desain Dekomposisi FB01 Hashing	31
Gambar 19 : Desain Dekomposisi FB03 Meningkatkan Rarity Consumable	31
Gambar 20 : Asistensi 1	73
Gambar 21 : Asistensi 2	74

I. DESKRIPSI PERSOALAN

Persoalan yang kelompok kami selesaikan melalui pengerjaan Tugas Besar IF1210 Dasar Pemrograman adalah terkait kebutuhan beberapa pihak pada suatu sistem inventarisasi. Dalam sistem inventarisasi ini, terdapat dua jenis kebutuhan yang harus diinventarisasi, yakni gadget dan *consumables*. Baik gadget maupun *consumables* memiliki spesifikasi masing-masing. Oleh sebab itu, untuk pengelolaan dua jenis benda tersebut, dibutuhkan sistem atau program yang mampu, di antaranya, memberikan kesempatan bagi user atau pengguna mendaftarkan sebagai *user* valid atau yang dapat mengakses program, mencari gadget atau *consumables* beserta spesifikasinya, melakukan pencatatan terhadap penambahan atau pengurangan gadget baru, dll. Untuk pengelolaan tersebut, perlu diciptakan beberapa sub-program atau fungsi yang dapat menjawab kebutuhan inventarisasi yang akan dijabarkan sebagai berikut

F01 - Register

Program register digunakan untuk menerima masukan berupa nama, username, password, dan alamat sebagai syarat pengguna dapat menggunakan program ini. Validasi username dibuat unik sehingga satu username hanya bisa digunakan oleh satu pengguna saja.

F02 - Login

Program login digunakan pengguna untuk masuk ke dalam menu program. Fungsi login menerima masukan berupa username dan password yang telah didaftarkan pengguna pada fungsi register. Pada masukan login, password telah di *hash* sehingga pengguna tidak dapat melihat password dari file csv (FB01).

F03 - Pencarian Gadget Berdasarkan Rarity

Program ini digunakan oleh admin dan user untuk mencari gadget berdasarkan rarity tertentu. Fungsi cariRarity menerima masukan berupa jenis rarity. Validasi dari rarity tersebut yaitu masukan harus valid (C, B, A, S).

F04 - Pencarian Gadget Berdasarkan Tahun Ditemukan

Program ini digunakan oleh admin untuk mencari gadget berdasarkan tahun ditemukan. Fungsi cariTahun menerima masukan berupa integer dan kategori. Validasi dari masukan tersebut yaitu format tanggal (YYYY) dan format masukan kategori (=, >, <, >=, <=).

F05 - Menambah Item

Program menambah item digunakan oleh admin untuk menambahkan item ke dalam inventori. Fungsi tambahItem menerima masukan yang nantinya akan divalidasi berdasarkan rarity gadget (C, B, A, S) atau rarity consumable.

F06 - Menghapus Gadget atau Consumable

Program ini digunakan oleh admin untuk menghapus gadget atau consumable. Fungsi hapusItem menerima masukan berupa ID Item yang akan dihapus dari database file csv.

F07 - Mengubah Jumlah Gadget

Program ini digunakan oleh admin untuk mengubah jumlah gadget atau consumable. Fungsi ubahJumlah menerima masukan berupa ID Item dan jumlah yang akan diubah. Validasi jumlah digunakan ketika jumlah yang akan diubah lebih banyak dari stok sekarang.

F08 - Meminjam Gadget

Program ini digunakan oleh user untuk melakukan peminjaman gadget. Fungsi pinjam menerima masukan berupa ID Item, tanggal peminjaman, dan jumlah yang akan divalidasi dengan database file csv.

F09 - Mengembalikan Gadget

Program ini digunakan oleh user untuk melakukan pengembalian gadget. Fungsi ini menerima masukan berupa nomor peminjaman dan tanggal pengembalian yang telah divalidasi sebelumnya. Gadget yang telah dikembalikan akan bertambah ke dalam database gadget.csv

F10 - Meminta Consumable

Program ini digunakan oleh user untuk meminta consumable. Fungsi mintaConsumable menerima masukan berupa ID Item, jumlah, dan tanggal permintaan. Masukan akan divalidasi sebelum diteruskan ke dalam database.

F11 - Melihat Riwayat Peminjaman Gadget

Program ini digunakan oleh admin untuk melihat riwayat peminjaman gadget. Fungsi riwayatPinjam akan menampilkan data 5 entry paling baru dan 5 entry tambahan apabila diinginkan dan *sorted descending* berdasarkan tanggal.

F12 - Melihat Riwayat Pengembalian Gadget

Program ini digunakan oleh admin untuk melihat riwayat pengembalian gadget. Fungsi riwayatKembali akan menampilkan 5 data entry paling baru dan 5 data entry tambahan apabila diinginkan dengan format *sorted descending* berdasarkan tanggal.

F13 - Melihat Riwayat Pengembalian Consumable

Program ini digunakan oleh admin untuk melihat riwayat pengembalian consumable. Fungsi riwayatConsumable akan menampilkan 5 data entry paling baru dan 5 data entry tambahan apabila diinginkan dengan format *sorted descending* berdasarkan tanggal.

F14 - Load Data

Program load data digunakan untuk melakukan loading data ke dalam sistem. Data yang dimaksud tersebut yaitu berupa file database csv meliputi user, gadget, consumable, consumable history, gadget borrow history, gadget return history, dan inventory user.

F15 - Save Data

Fungsi save data digunakan untuk melakukan penyimpanan data dari file database csv yang telah dimodifikasi. Data yang dimaksud sama seperti data loading pada F14.

F16 - Help

Fungsi help digunakan untuk memberikan panduan baik kepada admin maupun user. Fungsi ini dapat dipanggil tanpa melakukan login sebelumnya. Terdapat perbedaan keluaran fungsi help sebagai admin atau user.

F17 - Exit

Fungsi exit digunakan untuk mengakhiri program atau keluar dari aplikasi. Fungsi exit menerima konfirmasi masukan (Y/N) untuk menjalankan prosedur save.

FB01 - Hashing

Fungsi ini akan melakukan hashing pada password user. Dengan melakukan hashing, password user akan aman meskipun user membuka database csv. Pada tugas besar kali ini kami menggunakan metode Polynomial rolling hash. Metode ini mengubah string password menjadi tipe integer. Algoritma dari metode hashing ini ialah pertama inisialisasikan sebuah integer, misal p (biasanya bilangan prima, kami memilih 101) dan sebuah integer besar, misal m (bernilai cukup besar, kami memilih $10^9 + 1$) dan juga sebuah integer power_of_p (polinomial dari p , tetapi di mod dengan m) serta integer hashed (bernilai 0, dan akan dijumlahkan selama berjalannya fungsi hash, sebagai hasil dari hashing). Lalu per karakter dari string password dijadikan kode bilangan unicode dikalikan dengan power_of_p dan di mod dengan m lalu ditambahkan ke dalam variabel hashed, juga setiap berganti karakter, power_of_p dikalikan dengan p (polinomial dari p dengan pangkat urutan karakter) dan di-mod dengan m . Hal tersebut dilakukan dan dijumlahkan terus menerus sampai karakter terakhir dari string password. Hasil dari algoritma tersebut, berupa variabel hashed, dikembalikan sebagai hasil dari fungsi hashing ini.

FB02 - Mengembalikan Gadget secara Parsial

Fungsi ini berhubungan dengan F09 yang memungkinkan user untuk melakukan pengembalian gadget secara parsial.

FB03 - Meningkatkan Rarity Consumables

Fungsi ini memungkinkan pengguna untuk meningkatkan rarity consumables dengan menggabungkan beberapa consumables menjadi sebuah consumables baru.

II. DAFTAR PEMBAGIAN TUGAS

Tabel 01: Daftar Pembagian Tugas Pada Source Code

Fungsionalitas	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F01 - Register	Procedure register	16520129, 16520299	16520129, 16520299	16520129 16520299 16520149 16520329
F02 - Login	Procedure login	16520299	16520299	
F03 - Pencarian Gadget Berdasarkan Rarity	Procedure carirarity	16520299	16520299	
F04 - Pencarian Gadget Berdasarkan Tahun	Procedure caritahun	16520149	16520149	
F05 - Menambah Item	Procedure tambahItem	16520299	16520299	
F06 - Menghapus Gadget atau Consumable	Procedure hapusItem	16520299	16520299	
F07 - Mengubah Jumlah Gadget	Procedure ubahjumlah	16520149	16520149	
F08 - Meminjam Gadget	Procedure pinjam	16520149	16520149	
F09 - Mengembalikan Gadget	Procedure kembalikan	16520149	16520149	
F10 - Meminta Consumable	Procedure mintaconsumable	16520329 16520299 16520149	16520329 16520299 16520149	
F11 - Melihat Riwayat Peminjaman Gadget	Procedure riwayatPinjam	16520299	16520299	
F12 - Melihat Riwayat Pengembalian Gadget	Procedure riwayatKembali	16520329	16520329	
F13 - Melihat Riwayat Pengembalian Consumable	Procedure riwayatConsumable	16520329	16520329	
F14 - Load Data	Procedure load	16520129, 16520299	16520129, 16520299	

F15 - Save Data	Procedure save	16520299	16520299	
F16 - Help	Procedure help	16520129, 16520299	16520129, 16520299	
F17 - Exit	Procedure exit	16520129, 16520299	16520129	
FB01 - Hashing	Function hashing	16520299	16520299	
FB02 - Mengembalikan Gadget secara Parsial	Procedure kembalikan	16520149	16520149	
FB03 - Meningkatkan Rarity Consumables	Procedure gacha	16520299	16520299	
Main Program	Main Program	16520299	16520299	

Tabel 02: Daftar Pembagian Tugas Pada Laporan

No	Bagian Laporan	NIM
1.	Halaman Cover	16520129, 16520329
2.	Daftar Isi	16520129, 16520329
3.	Daftar Tabel	16520129, 16520329
4.	Daftar Gambar	16520129, 16520149, 16520299
5.	Deskripsi Persoalan	16520329, 16520149, 16520299
6.	Daftar Pembagian Tugas	16520129, 16520299, 16520149, 16520329
7.	Checklist Hasil Rancangan, Implementasi, dan Uji Coba	16520329
8.	Desain Perintah	16520149
9.	Desain Kamus Data	16520129, 16520299
10.	Desain Dekomposisi Algoritmik dan Fungsional Program	16520129, 16520149, 16520299

11.	Spesifikasi	16520129, 16520149, 16520299
12.	Hasil Pengujian Program	16520329
13.	Catatan dan Lampiran	16520129

III. CHECKLIST HASIL RANCANGAN, IMPLEMENTASI, DAN UJI COBA

Tabel 03: Daftar Checklist Hasil Rancangan, Implementasi dan Testing Setiap Primitif

Fungsionalitas	Desain	Implementasi	Testing
F01 - Register	✓	✓	✓
F02 - Login	✓	✓	✓
F03 - Pencarian Gadget Berdasarkan Rarity	✓	✓	✓
F04 - Pencarian Gadget Berdasarkan Tahun	✓	✓	✓
F05 - Menambah Item	✓	✓	✓
F06 - Menghapus Gadget atau Consumable	✓	✓	✓
F07 - Mengubah Jumlah Gadget	✓	✓	✓
F08 - Meminjam Gadget	✓	✓	✓
F09 - Mengembalikan Gadget	✓	✓	✓
F10 - Meminta Consumable	✓	✓	✓
F11 - Melihat Riwayat Peminjaman Gadget	✓	✓	✓
F12 - Melihat Riwayat Pengembalian Gadget	✓	✓	✓
F13 - Melihat Riwayat Pengembalian Consumable	✓	✓	✓
F14 - Load Data	✓	✓	✓
F15 - Save Data	✓	✓	✓
F16 - Help	✓	✓	✓
F17 - Exit	✓	✓	✓
FB01 - Hashing	✓	✓	✓

FB02 - Mengembalikan Gadget secara Parsial	✓	✓	✓
FB03 - Meningkatkan Rarity Consumables	✓	✓	✓

IV. DESAIN PERINTAH

1. >>> register

input:

nama → pemakai akan di-*prompt* untuk mengisi nama

username → pemakai akan di-*prompt* untuk mengisi username

password → pemakai akan di-*prompt* untuk mengisi password

alamat → pemakai akan di-*prompt* untuk mengisi alamat

output:

data akan disimpan ke *database*, akan muncul juga pesan informasi bahwa user telah berhasil register ke dalam Kantong Ajaib

2. >>> login

input:

username → pemakai akan di-*prompt* untuk mengisi username yang telah diregister

password → pemakai akan di-*prompt* untuk mengisi password sesuai data registrasi

output:

akan ditampilkan pesan bahwa pengguna telah berhasil login dan akan mampu membuat perintah sesuai syarat role (admin atau user)

3. >>> carirarity

input:

rarity → pemakai akan di-*prompt* untuk mengisi rarity gadget ('C', 'B', 'A', atau 'S')

output:

akan ditampilkan semua gadget yang memiliki rarity yang sesuai dengan masukannya

4. >>> caritahun

input:

tahun → pemakai akan di-*prompt* untuk mengisi tahun gadget ditemukan

kategori → pemakai akan di-*prompt* untuk mengisi kategori operasi pembandingan (=, <, >, <=, atau >=)

output:

akan ditampilkan semua gadget yang ditemukan saat, sebelum, setelah, saat dan sebelum, atau saat dan setelah tahun inputan user

5. >>> tambahitem

input:

id_item → pemakai akan di-*prompt* untuk mengisi id gadget atau consumable baru

nama → pemakai akan di-*prompt* untuk mengisi nama gadget atau consumable baru

deskripsi → pemakai akan di-*prompt* untuk mengisi deskripsi gadget atau consumable

baru

jumlah → pemakai akan di-*prompt* untuk mengisi jumlah gadget atau consumable

rarity → pemakai akan di-*prompt* untuk mengisi rarity dari gadget atau consumable (tahun ditemukan) → jika id_item yang berhasil dimasukan dimulai dengan karakter ‘ G’ atau gadget, pemakai akan di-*prompt* untuk mengisi tahun ditemukan

output:

jika semua data sudah terisi dengan benar, akan muncul pesan bahwa item berhasil dimasukan ke database

6. >>> hapusitem

input:

id_item → pemakai akan di-*prompt* untuk mengisi id item dan akan divalidasi apakah item tersebut ada di database

validasi dari pengguna → pemakai akan di-*prompt* untuk mengisi Y atau N untuk memastikan apakah pengguna benar-benar akan menghapus item tersebut. Jika Y maka item akan terhapus dari database

output:

jika id_item sebelumnya teregistrasi dan pengguna memasukkan Y maka akan muncul pesan bahwa item telah berhasil dihapus dari database

7. >>> ubahjumlah

input:

id_item → pemakai akan di-*prompt* untuk mengisi id_item dan akan divalidasi keberadaannya di database

jumlah → pemakai akan di-*prompt* untuk mengisi jumlah, (+) untuk penambahan item, (-) untuk mengurangi item

output:

akan ditampilkan pesan bahwa item telah ditambahkan atau dikurangi dan informasi stok terbaru setelah ditambah atau dikurangi

8. >>> pinjam

input:

id_item → pemakai akan di-*prompt* untuk mengisi id gadget yang ingin dipinjam. Akan divalidasi apakah pengguna pernah meminjam barang yang sama tapi tidak pernah dikembalikan. Jika pernah meminjam dan belum mengembalikan secara utuh, maka pengguna tidak dapat meminjam barang tersebut.

tanggal → pemakai akan di-*prompt* untuk mengisi tanggal peminjaman dengan format dd/mm/yyyy

jumlah_peminjaman → pemakai akan di-*prompt* untuk mengisi jumlah barang yang dipinjam dan harus kurang dari jumlah data yang ada dan harus lebih besar dari 0.

output:

akan dimunculkan pesan bahwa gadget berhasil dipinjam

9. >>> kembalikan

output:

akan dimunculkan setiap item yang pernah dipinjam beserta penomoran setiap item

input:

nomor_pilihan → pemakai akan di-*prompt* untuk mengisi nomor opsi sesuai nomor item yang dimunculkan sebelumnya

tanggal → pemakai akan di-*prompt* untuk mengisi tanggal pengembalian dengan format dd/mm/yyyy

jumlah_pengembalian → pemakai akan di-*prompt* untuk mengisi jumlah yang ingin dikembalikan dengan minimal 1 dan maksimalnya adalah jumlah yang ada di inventory pengguna

output:

akan muncul pesan bahwa item tersebut telah dikembalikan

10. >>> minta

input:

id_item → pemakai akan di-*prompt* untuk mengisi id_item dan akan divalidasi apakah id tersebut ada di dalam database

jumlah_permintaan → pemakai akan di-*prompt* untuk mengisi jumlah yang diminta dengan minimal 0 dan maksimal jumlah consumable yang terdapat pada database

tanggal_permintaan → pemakai akan di-*prompt* untuk mengisi tanggal dan akan divalidasi pengisian apakah benar dan sesuai format

output:

akan muncul pesan bahwa item telah berhasil diambil

11. >>> riwayatpinjam

output:

akan ditampilkan daftar riwayat pinjam dengan maksimal 5 entri terbaru (sorted descending berdasarkan tanggal). Jika pengguna meminta lebih dari 5 entri, pengguna dapat melihat lagi tambahan 5 entri sebelumnya

12. >>> riwayatkembali

output:

akan ditampilkan daftar riwayat kembali dengan maksimal 5 entri terbaru (sorted descending berdasarkan tanggal). Jika pengguna meminta lebih dari 5 entri, pengguna dapat melihat lagi tambahan 5 entri sebelumnya

13. >>> riwayatambil

output:

akan ditampilkan daftar riwayat ambil dengan maksimal 5 entri terbaru (sorted descending berdasarkan tanggal). Jika pengguna meminta lebih dari 5 entri, pengguna dapat melihat lagi tambahan 5 entri sebelumnya

14. >>> load data: python kantongajaib.py <nama_folder>

input:

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

output:

akan muncul pesan Loading dan data akan terload. Akan muncul juga pesan selamat datang di “Kantong Ajaib!”

15. >>> save

input:

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

output:

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

16. >>> help

output:

akan muncul setiap perintah yang dapat dilakukan beserta dengan keterangan kegunaan perintah tersebut. Akan dimunculkan juga eligibilitas dari perintah sesuai role.

17. >>> exit

input:

perintah → pemakai akan di-*prompt* untuk mengisi Y atau N

output:

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

V. DESAIN KAMUS DATA

MAIN PROGRAM

KAMUS GLOBAL

```
type data_user : <id: string, username: string, nama: string, alamat: string,  
                password: integer, role: string>  
type data_gadget : <id:string, nama: string; deskripsi: string; jumlah: integer;  
                   rarity: character; tahun_ditemukan: integer>  
type data_consumable : <id: string; nama: string; deskripsi: string; jumlah: integer; rarity: character>  
type data_gadget_borrow_history : <id: string; id_peminjam: string; id_gadget: string;  
                                   tanggal_peminjaman: string; jumlah: integer; is_returned: boolean>  
type data_gadget_return_history : <id: string; id_peminjaman: string; tanggal_pengembalian: string;  
                                   jumlah_pengembalian: integer; applicable_or_not: string>  
type data_consumable_history : <id: string; id_pengambil: string; id_consumable: string;  
                               tanggal_pengambilan: string; jumlah: integer>  
type data_inventory_user : <id_user: string; id_consumable: string; jumlah_yang_dipunyai: integer>  
  
user : array of data_user  
gadget : array of data_gadget  
consumable : array of data_consumable  
consumable_history : array of data_consumable_history  
gadget_borrow_history : array of data_gadget_borrow_history  
gadget_return_history : array of data_gadget_return_history  
inventory_user : array of data_inventory_user  
idUser, directory, perintah : string  
random : integer  
lstChance : array of float  
lstPerintah : array of string  
program, isAdmin, hasLogin : boolean
```

F01

procedure register()

KAMUS LOKAL

```
nama, username, password, alamat, idUser : string  
notUnik : boolean  
i, count : integer  
register : array of data_user  
{ Variable global }  
global user  
{ function / procedure }  
function hashing(password : string) -> integer
```

F02

procedure login()

KAMUS LOKAL

username, password : string
i : integer
rolling : boolean
{ Variable global }
global hasLogin
global isAdmin
global idUser
{ function / procedure }
function hashing(password : string) -> integer
function Bold(text : string) -> string

F03

procedure cariRarity()

KAMUS LOKAL

rarity : string
i : integer
ditemukan : boolean

F04

procedure caritahun()

KAMUS LOKAL

tahun, i : integer
kategori : string
found : boolean

F05

procedure tambahItem()

KAMUS LOKAL

lanjut, isNumber : boolean
nama, deskripsi : string
jumlah, tahun : integer
rarity : character
arrTambahItem : any of data_gadget or data_consumable
{ function / procedure }
function IDItemAda(data : array of any of data_user or data_gadget or data_consumable or
data_gadget_return_history or data_gadget_borrow_histroy or
data_consumable_history or data_inventory_user,
ID : string) -> boolean

F06

procedure hapusItem()

KAMUS LOKAL

ID, jawaban : string

count : integer
 returnSort : data_gadget_return_histroy
 namaUser, namaGadget, id_user, id_gadget, nextInp : string
function validasiYN(jawaban : string) → boolean

F13

procedure riwayatConsumable()

KAMUS LOKAL

rolling, berikutnya : boolean
 count : integer
 consumableSort : data_consumable_history
 namaUser, namaConsumable : string
 { function / procedure }
function validasiYN(jawaban : string) → boolean

F14

procedure load(folder)

KAMUS LOKAL

user : array of data_user
 gadget : array of data_gadget
 consumable : array of data_consumable
 consumable_history : array of data_consumable_history
 gadget_borrow_history : array of data_gadget_borrow_history
 gadget_return_history : array of data_gadget_return_history
 inventory_user : array of data_inventory_user

{ Variable global }

global user

global gadget

global consumable

global consumable_history

global gadget_borrow_history

global gadget_return_history

global inventory_user

{ function / procedure }

function load_data(file : csv) → array of array of string

function tryChange(data : array of array of string) -> array of any of data_user or data_gadget or
 data_consumable or data_gadget_return_history or
 data_gadget_borrow_histroy or data_consumable_history or
 data_inventory_user

function load_data(file : csv) → array of array of string

KAMUS LOKAL

f : SEQFILE OF
 (*) raw_lines : array of string
 (1) mark : None
 lst : array of string
 data : string
 lstAll : array of array of string

function tryChange(data : array of array of string) → array of any of data_user or data_gadget or data_consumable or data_gadget_return_history or data_gadget_borrow_histroy or data_consumable_history or data_inventory_user

KAMUS LOKAL

i, j : integer

function loading() → string

KAMUS LOKAL

parser : ArgumentParser
 directory, parent, path, files : string
 lstFile : array of string
 { function / procedure }
 function fileExist(files : csv) → boolean

function fileExist(files : csv) → boolean

KAMUS LOKAL

-

F15

procedure save()

KAMUS LOKAL

parent, directory, path : string
 { function / procedure }
procedure save_data (file : csv, data : array of any of data_user or data_gadget or data_consumable or data_gadget_return_history or data_gadget_borrow_histroy or data_consumable_history or data_inventory_user)

procedure save_data (file : csv, data : array of any of data_user or data_gadget or data_consumable or data_gadget_return_history or data_gadget_borrow_histroy or data_consumable_history or data_inventory_user)

KAMUS LOKAL

data : string
 f : SEQFILES OF
 (*) array of string

(1) mark : None

function convert_datas_to_string (data : array of array of string) → string
string_data : string
arr_data : array of string and integer and boolean
arr_data_all_string : array of string

function convert_datas_to_string (data : array of array of string) -> string

KAMUS LOKAL

string_data : string
arr_data : array of string and integer and boolean
arr_data_all_string : array of string

F16

procedure help()

KAMUS LOKAL

-

F17

procedure exit()

KAMUS LOKAL

isSave : string
{ global variable }
global program
{ function / procedure }
function validasiYN(jawaban : string) → boolean
procedure save()

FB01

function hashing(str) → integer

KAMUS LOKAL

P, m, powerOfP, hashed, i : integer

FB02 + F09

procedure kembalikan()

KAMUS LOKAL

syaratnya : boolean
option, indeksnya, markernya, max_returned, total_amount_returned, total_amount_returned_updated,
amount_returned, z, n, an : integer
id_returned_gadget, id_pengembalian, tanggal : string
personal_borrow_not_returned, updated_unique_personal_borrow_not_returned : array of string
unique_personal_borrow_not_returned : set of string

FB03

procedure gacha()

KAMUS LOKAL

count, urutan, digunakan, urutanInventory, jumlah : integer
IDInventory : array of string
digunakanBenar, jumlahBenar, finished : boolean
perintah, rarity : character
tambah_con_history : data_consumable_history
tambah_inventory : data_inventory_user
perintah : string
{ Variable global }
global lstChance
global inventory_user
global consumable_history
global consumable
{ function / procedure }
function cariData(data : array of any of data_user or data_gadget or data_consumable or data_gadget_return_history or data_gadget_borrow_histroy or data_consumable_history or data_inventory_user, dicari : string, index : integer) -> integer
function cariID(data : array of any of data_user or data_gadget or data_consumable or data_gadget_return_history or data_gadget_borrow_histroy or data_consumable_history or data_inventory_user, ID : string) -> integer
function Bold(text : string) → string
function chance(lstRarity : array of float, rarity : character) → array of float
function validasiYN(jawaban : string) → boolean
function hasilGacha(lstChance : array of float) → character

function chance(lstRarity : array of float, rarity : character) → array of float

KAMUS LOKAL

sumN : float

function hasilGacha(lstChance) → character

KAMUS LOKAL

random, i : integer
{ function / procedure }
procedure seed()
function rand() → integer

procedure seed()

KAMUS LOKAL

random : integer

function rand()

KAMUS LOKAL

a, c, m : integer
{ Variable global }

global random

FUNGSI TAMBAHAN

function Bold(string : string) → string

KAMUS LOKAL

hasil : string

function cariID(data : array of any of data_user or data_gadget or data_consumable or data_gadget_return_history or data_gadget_borrow_histroy or data_consumable_history or data_inventory_user, ID : string) → integer

KAMUS LOKAL

i : integer

function IDItemAda(data : array of any of data_user or data_gadget or data_consumable or data_gadget_return_history or data_gadget_borrow_histroy or data_consumable_history or data_inventory_user, ID : string) → boolean

KAMUS LOKAL

Ada : boolean

i : integer

function cariData(data : array of any of data_user or data_gadget or data_consumable or data_gadget_return_history or data_gadget_borrow_histroy or data_consumable_history or data_inventory_user, dicari : string, index : integer) → integer

KAMUS LOKAL

i : integer

function colorRed(text : string) → string

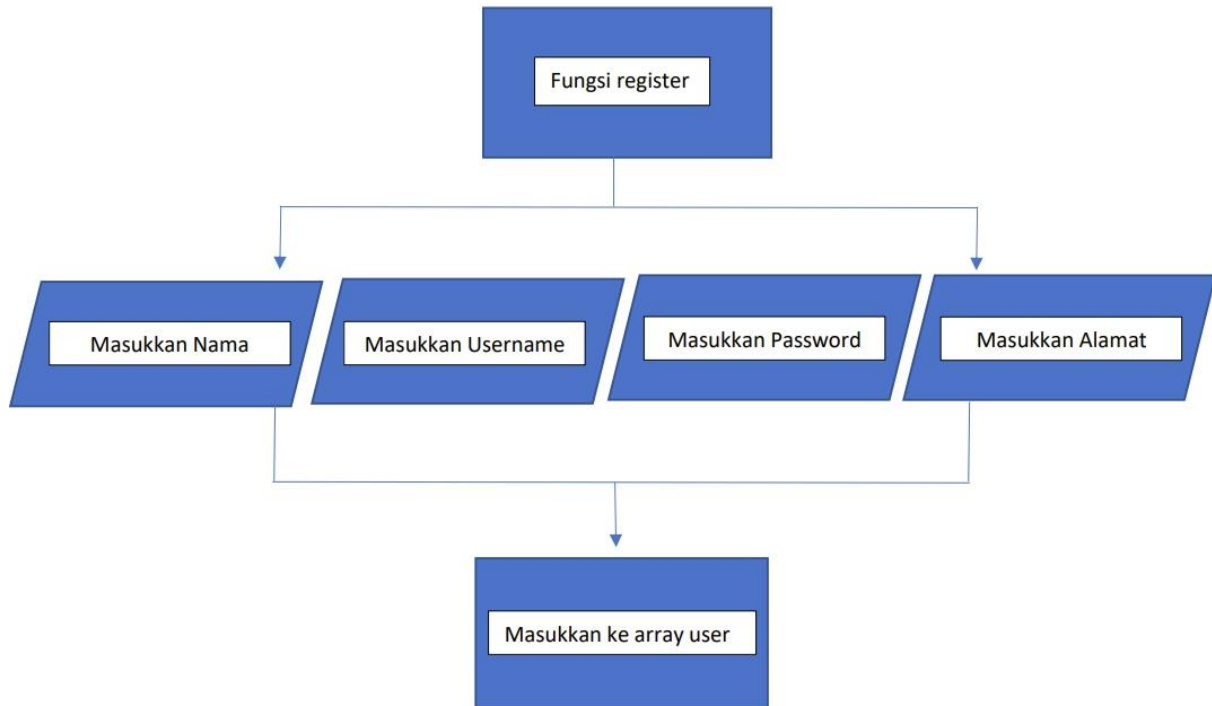
KAMUS LOKAL

function validasiYN(string : string) → boolean

KAMUS LOKAL

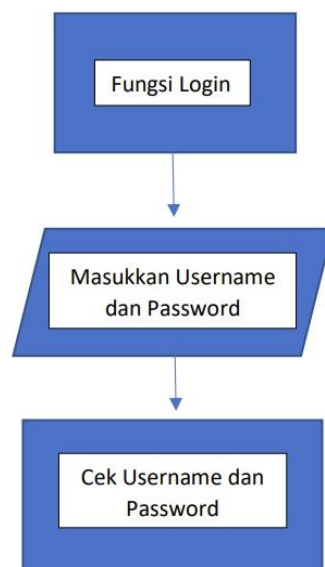
VI. DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL PROGRAM

F01 - Register



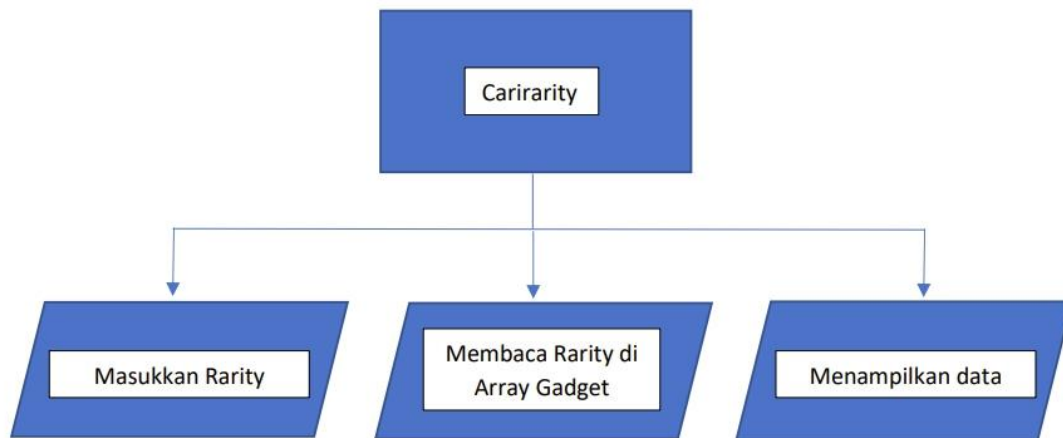
Gambar 01 : Desain Dekomposisi F01 Register

F02 - Login



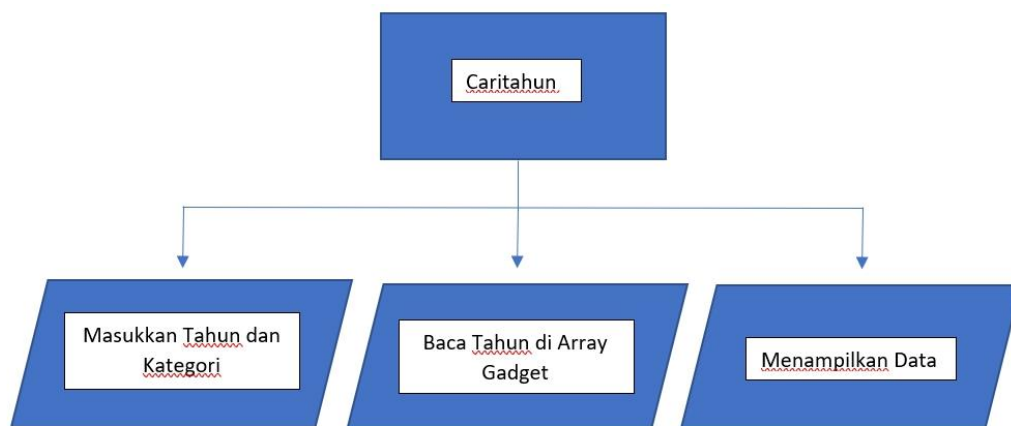
Gambar 02 : Desain Dekomposisi F02 Login

F03 - Pencarian Gadget Berdasarkan Rarity



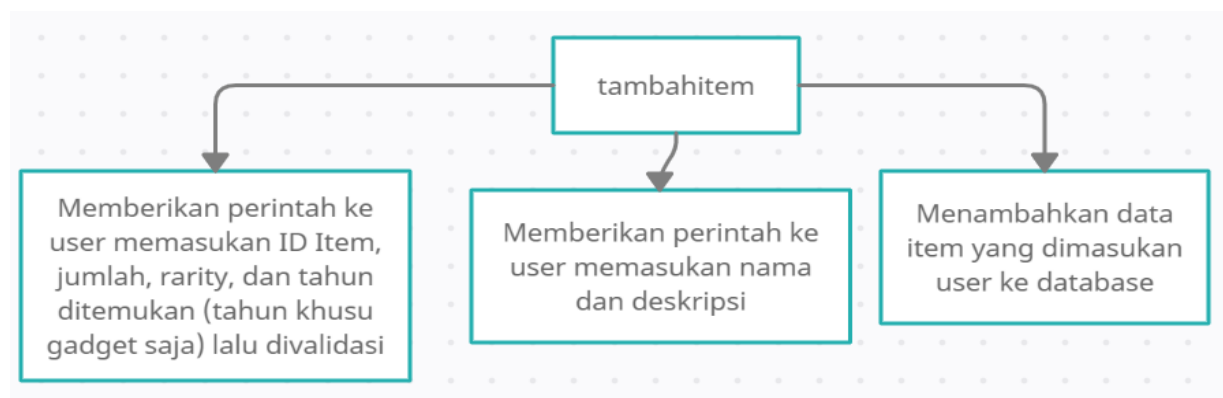
Gambar 03 : Desain Dekomposisi F03 Pencarian Gadget Berdasarkan Rarity

F04 - Pencarian Gadget Berdasarkan Tahun Ditemukan



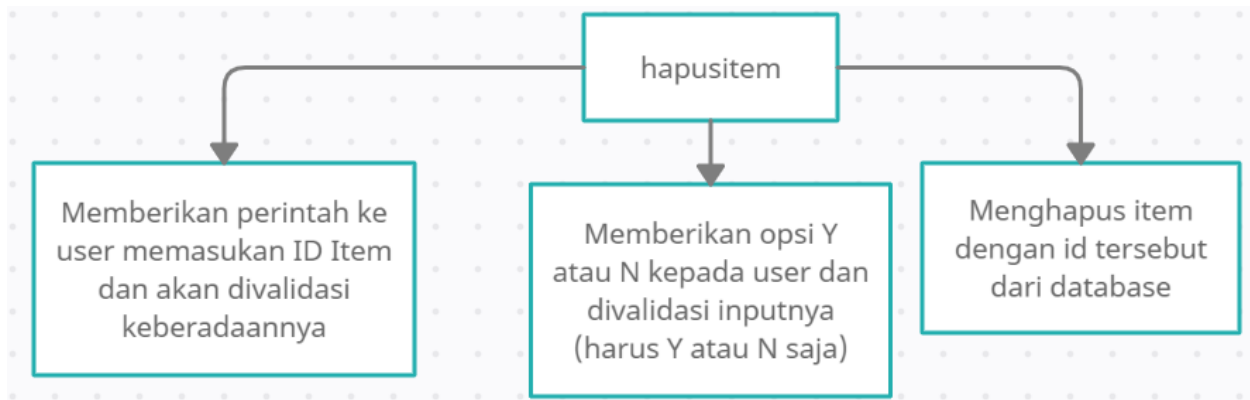
Gambar 04 : Desain Dekomposisi F04 Pencarian Gadget Berdasarkan Tahun Ditemukan

F05 - Menambah Item



Gambar 05 : Desain Dekomposisi F05 Menambah Item

F06 - Menghapus Gadget atau Consumable



Gambar 06 : Desain Dekomposisi F06 Menghapus Gadget atau Consumable

F07 - Mengubah Jumlah Gadget atau Consumable pada Inventory



Gambar 07 : Desain Dekomposisi F07 Mengubah Jumlah Gadget atau Consumable pada Inventory

F08 - Meminjam Gadget



Gambar 08 : Desain Dekomposisi F08 Meminjam Gadget

F09 + FB02 - Mengembalikan Gadget



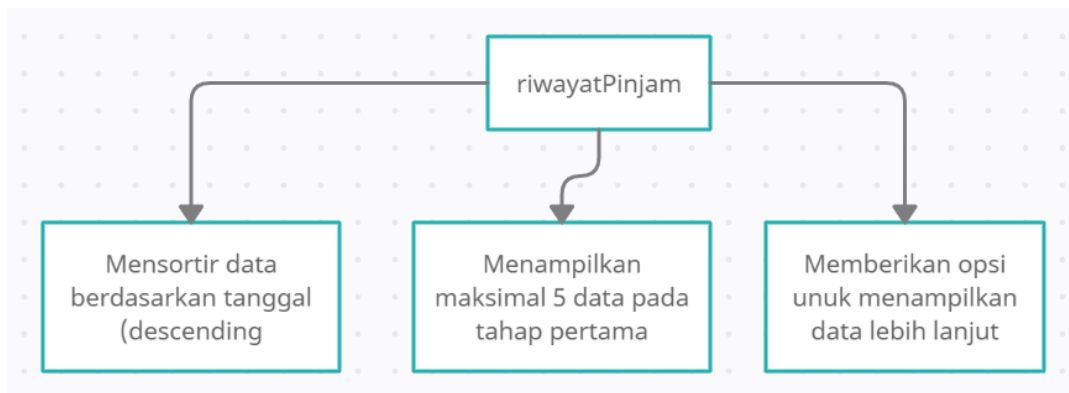
Gambar 09 : Desain Dekomposisi F09 + FB02 Mengembalikan Gadget

F10 - Meminta Consumable



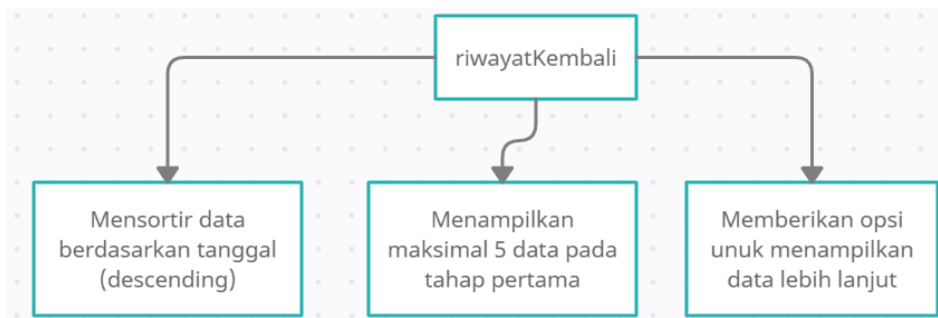
Gambar 10 : Desain Dekomposisi F10 Meminta Consumable

F11 - Melihat Riwayat Peminjaman Gadget



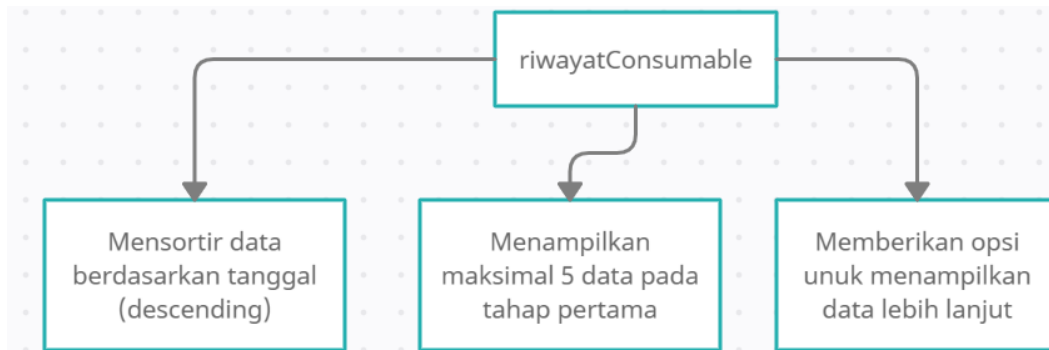
Gambar 11 : Desain Dekomposisi Melihat Riwayat Peminjaman Gadget

F12 - Melihat Riwayat Pengembalian Gadget



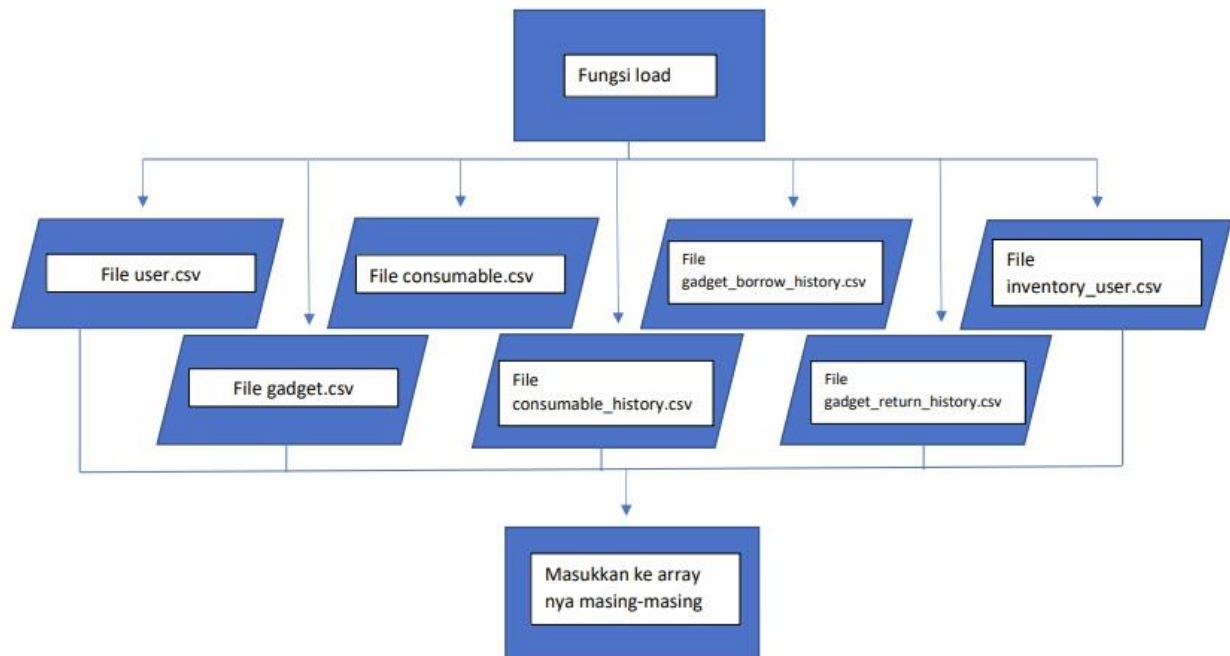
Gambar 12 : Desain Dekomposisi F12 Melihat Riwayat Pengembalian Gadget

F13 - Melihat Riwayat Pengambilan Consumable



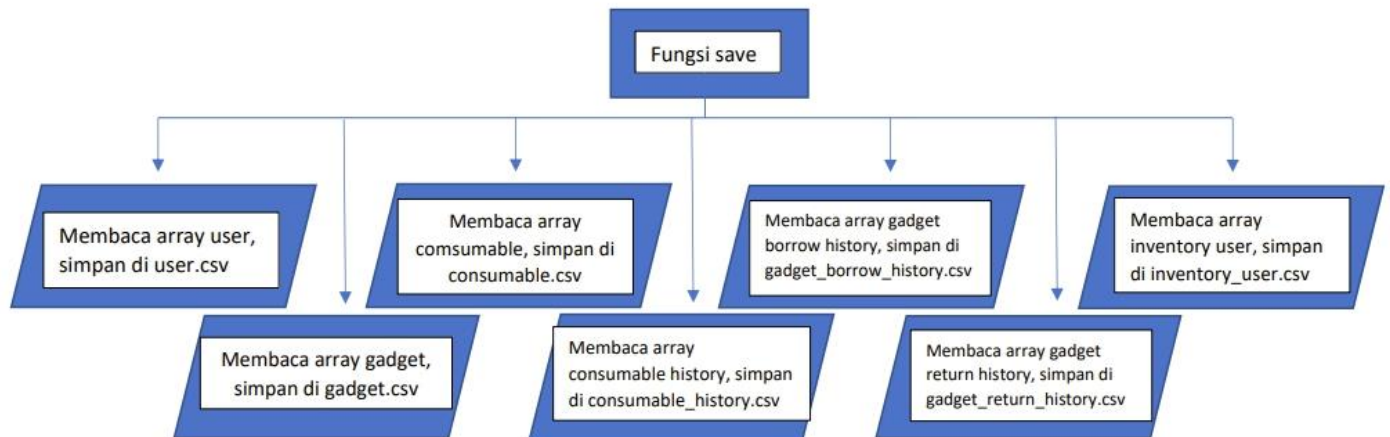
Gambar 13 : Desain Dekomposisi F13 Melihat Riwayat Pengembalian Consumable

F14 - Load Data



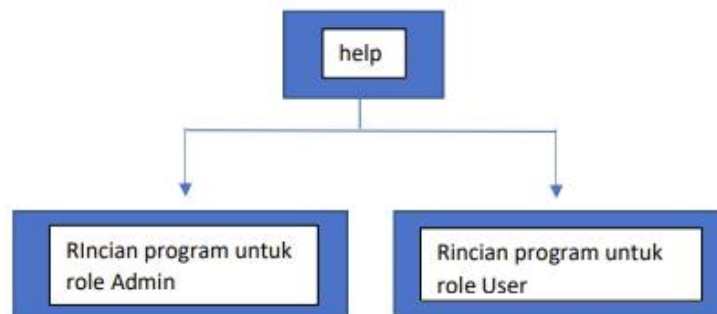
Gambar 14 : Desain Dekomposisi F14 Load Data

F15 - Save Data



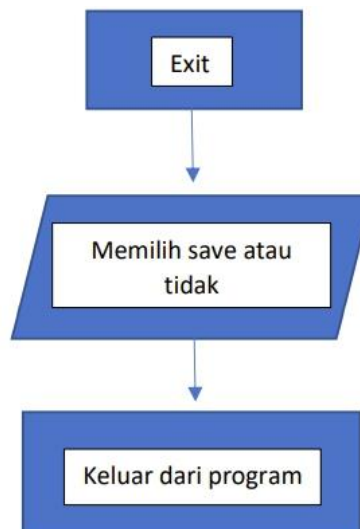
Gambar 15 : Desain Dekomposisi F15 Save Data

F16 - Help



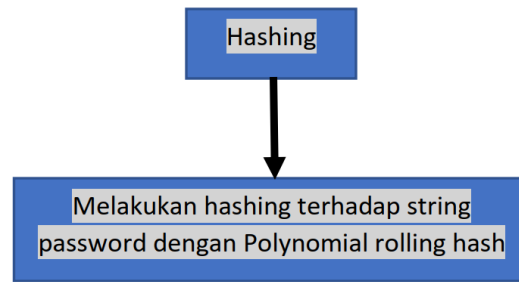
Gambar 16 : Desain Dekomposisi F16 Help

F17 - Exit



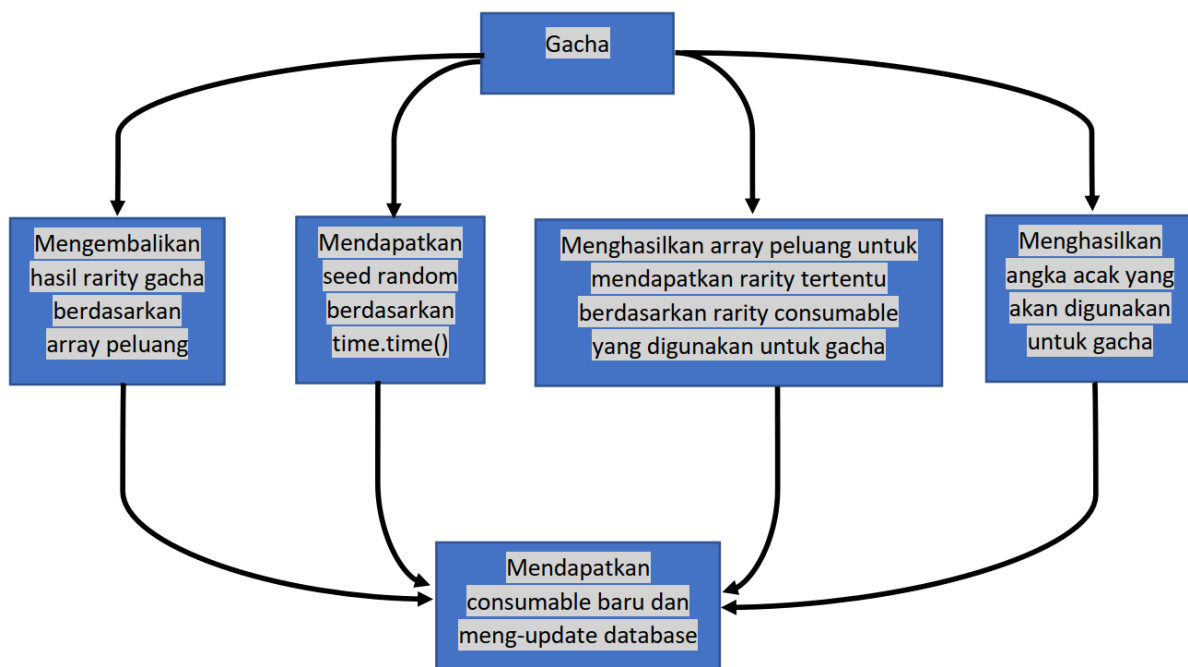
Gambar 17 : Desain Dekomposisi F17 Exit

FB01 - Hashing



Gambar 18 : Desain Dekomposisi FB01 - Hashing

FB03 - Meningkatkan Rarity Consumable



Gambar 19 : Desain Dekomposisi FB03 - Meningkatkan Rarity Consumable

VII. SPESIFIKASI

1. MAIN PROGRAM

KAMUS GLOBAL

```
type data_user : <id: string, username: string, nama: string, alamat: string,  
                password: integer, role:string>  
type data_gadget : <id:string, nama: string; deskripsi: string; jumlah: integer;  
                  rarity: character; tahun_ditemukan: integer>  
type data_consumable : <id: string; nama: string; deskripsi: string;  
                      jumlah: integer; rarity: character>  
type data_gadget_borrow_history : <id: string; id_peminjam: string; id_gadget: string;  
                                tanggal_peminjaman: string; jumlah: integer; is_returned: boolean>  
type data_gadget_return_history : <id: string; id_peminjaman: string; tanggal_pengembalian:  
                                string; jumlah_pengembalian: integer; applicable_or_not: string>  
type data_consumable_history : <id: string; id_pengambil: string; id_consumable: string;  
                              tanggal_pengambilan: string; jumlah: integer>  
type data_inventory_user : <id_user: string; id_consumable: string;  
                           jumlah_yang_dipunyai: integer>
```

```
user : array of data_user  
gadget : array of data_gadget  
consumable : array of data_consumable  
consumable_history : array of data_consumable_history  
gadget_borrow_history : array of data_gadget_borrow_history  
gadget_return_history : array of data_gadget_return_history  
inventory_user : array of data_inventory_user  
idUser, directory, perintah : string  
random : integer  
lstChance : array of float  
lstPerintah : array of string  
program, isAdmin, hasLogin : boolean
```

```
{ INISIALISASI }
```

```
user ← []; gadget ← []; consumable ← []; consumable_history ← []; gadget_borrow_history ←  
[]; gadget_return_history ← []; inventory_user ← []  
idUser ← ""; random ← 0; lstChance ← [0.0,0.0,0.0,0.0]  
lstPerintah ← ['register', 'login', 'carirarity', 'caritahun', 'tambahitem', 'hapusitem', 'ubahjumlah',  
              'pinjam', 'kembalikan', 'minta', 'riwayatpinjam', 'riwayatkembali', 'riwayatambil',  
              'save', 'help', 'gacha']
```

```
program ← true  
hasLogin ← false  
isAdmin ← false
```

```
{ Pemanggilan procedure loading() }  
directory ← loading()
```



```

\033[36m      .:#####. .:q.. :#####. \033[0m
\033[36m      #####  :  :  :  :  :  :  :  : \033[0m
\033[36m      #####..:.. :#####. :.:.:q. \033[0m
\033[36m      #####:..:.. :#####. :.:.:q. \033[0m
\033[36m      :.:.:q. :#####. :.:.:q. \033[0m
\033[36m      :.:.:q. :#####. :.:.:q. \033[0m
\033[36m      :.:.:q. :#####. :.:.:q. \033[0m
\033[36m      :.:.:q. :#####. :.:.:q. \033[0m
\033[36m      :.:.:q. :#####. :.:.:q. \033[0m
\033[36m      :.:.:q. :#####. :.:.:q. \033[0m

```

```

        """)
    elif perintah = "exit" then
        { Asumsi exit tidak perlu login }
        exit()
    else
        if hasLogin then
            if perintah = "register" then
                if isAdmin then
                    register()
                else
                    output("Maaf, hanya boleh diakses oleh admin ^_^")
                    output()
            elif perintah = "carirarity" then
                cariRarity()
            elif perintah = "caritahun" then
                caritahun()
            elif perintah = "tambahitem" then
                if isAdmin then
                    tambahItem()
                else
                    output("Maaf, hanya boleh diakses oleh admin ^_^")
                    output()
            elif perintah = "hapusitem" then
                if isAdmin then
                    hapusItem()
                else
                    output("Maaf, hanya boleh diakses oleh admin ^_^")
                    output()
            elif perintah = "ubahjumlah" then
                if isAdmin then
                    ubahjumlah()
                else
                    output("Maaf, hanya boleh diakses oleh admin ^_^")
                    output()

```

```

elif perintah = "pinjam" then
    if not isAdmin then
        pinjam()
    else
        output("Maaf, hanya boleh diakses oleh user ^_^")
        output()
elif perintah = "kembalikan" then
    if not isAdmin then
        kembalikan()
    else
        output("Maaf, hanya boleh diakses oleh user ^_^")
        output()
elif perintah = "minta" then
    if not isAdmin then
        mintaConsumable()
    else
        output("Maaf, hanya boleh diakses oleh user ^_^")
        output()
elif perintah = "riwayatpinjam" then
    if isAdmin then
        riwayatPinjam()
    else
        output("Maaf, hanya boleh diakses oleh admin ^_^")
        output()
elif perintah = "riwayatkembali" then
    if isAdmin then
        riwayatKembali()
    else
        output("Maaf, hanya boleh diakses oleh admin ^_^")
        output()
elif perintah = "riwayatambil" then
    if isAdmin then
        riwayatConsumable()
    else
        output("Maaf, hanya boleh diakses oleh admin ^_^")
        output()
elif perintah = "save" then
    save()
elif perintah = "gacha" then
    if not isAdmin then
        gacha()
    else
        output("Maaf, hanya boleh diakses oleh user ^_^")
        output()

```

```

    else
        { Masukan salah, tidak sesuai keyword yang valid, sudah login }
        output(colorRed("Input anda tidak valid"))
        output("Berikut merupakan input yang valid")
        help()
    elif perintah in IstPerintah then
        { Masukan benar, tetapi belum login }
        output(colorRed("Anda harus login terlebih dahulu"))
        output()
    else
        { Masukan salah, tidak sesuai keyword yang valid, belum login }
        output(colorRed("Input yang diberikan tidak tersedia"))
        output()

```

2. F01 - Register

```

{ Menambahkan data user baru ke dalam database
I.S. matriks data user terdefinisi
F.S. matriks data user ditambahkan data user baru }

```

KAMUS LOKAL

```

nama, username, password, alamat, idUser : string
notUnik : boolean
i, count : integer
register : array of data_user

```

ALGORITMA

```

output("Masukkan nama: "); input(nama)
output("Masukkan username: "); input(username)

{ Validasi username unik }
notUnik ← true
while notUnik do
    notUnik ← false
    i traversal [0..len(user)-1]
    if user[i][1] = username then
        notUnik ← true
        output()
        output("Username telah digunakan oleh user lain")
        output("Silakan input username yang berbeda")
        output()
        output("Masukkan username: "); input(username)
{ notUnik is false }

output("Masukkan password: "); input(password)

```

output("Masukkan alamat: "); input(alamat)

{ Pembuatan idUSer }

count \leftarrow 0

i traversal [0..len(user)-1]

if user[i][0][0] = 'U' then

 count \leftarrow count + 1

id_user \leftarrow "U" + str(count + 1)

{ Menambahkan data user baru ke dalam matriks data user }

register \leftarrow [[id_user,username,nama,alamat,hashing(password),"User"]]

user \leftarrow user + register

output("User", username, "telah berhasil register ke dalam Kantong Ajaib.")

3. F02 - Login

{ Melakukan prosedur login ke program dengan mengecek apakah data yang diinputkan sudah terdaftar di database

I.S. matriks data user, variable hasLogin, isAdmin, dan idUser terdefinisi

F.S. mengubah variable hasLogin jika username dan password sesuai dengan data dan isAdmin jika rolenya adalah admin }

KAMUS LOKAL

username, password : string

i : integer

rolling : boolean

{ Variable global }

global hasLogin

global isAdmin

global idUser

{ Function / Procedure }

function hashing(password : string) \rightarrow integer

{ Meng-hash password user menggunakan metode Polynomial Rolling Hash

I.S. password yang belum di hash terdefinisi

F.S. password ter-hash }

function Bold(text : string) \rightarrow string

{ Mengubah text menjadi terlihat bold jika di-print

I.S. text terdefinisi

F.S. text diberi 'kode' yang jika di-print text menjadi terlihat bold }

ALGORITMA

rolling \leftarrow true

while rolling do

output("Masukkan username: "); input(username)

```

output("Masukkan password: "); input(password)
output()
i traversal [0..len(user)-1]
    { Mengecek apakah data yang diinputkan telah terdaftar di database }
    if (username = user[i][1]) and (str(hashing(password)) = str(user[i][4])) then
        hasLogin ← true
        idUser ← user[i][0]
        output("Selamat datang " + Bold(user[i][2]) + " ^_^")

        { Mengecek apakah rolenya Admin }
        if user[i][5] is "Admin" then
            isAdmin = true
            break
    if not hasLogin then
        output("Username atau password Anda tidak cocok")
        output("Silakan masukkan kembali username dan password")
        output()
    else
        rolling ← false
{ rolling = false }

```

4. F03 - Pencarian gadget berdasarkan rarity

{ Mencari gadget yang memiliki rarity sesuai yang diinputkan
I.S. matriks data gadget terdefinisi
F.S. tercetak ke layar data gadget yang memiliki rarity sesuai yang diinputkan }

KAMUS LOKAL

rarity : string
i : integer
ditemukan : boolean

ALGORITMA

```

rarity ← input("Masukkan rarity: ")
output()
output("Hasil pencarian: ")
output()

ditemukan ← false
i traversal [0..len(user)-1]
    if gadget[i][4] = rarity then
        output("Nama      :", gadget[i][1])
        output("Deskripsi   :", gadget[i][2])
        output("Jumlah      :", gadget[i][3])
        output("Rarity      :", gadget[i][4])

```

```

    output("Tahun ditemukan :", gadget[i][5])
    output()
    ditemukan ← true
  if not ditemukan then
    output("Tidak ada gadget yang memiliki rarity", rarity)

```

5. F04 - Pencarian gadget berdasarkan tahun ditemukan

{ Mencari gadget berdasarkan tahun ditemukan dan kategorinya

I.S. : matriks data gadget terdefinisi

F.S. : tercetak ke layar data gadget sesuai input tahun ditemukan dan kategorinya }

KAMUS LOKAL

tahun, i : integer

kategori : string

found : boolean

ALGORITMA

```

repeat
  try
    input(tahun)
    input(kategori)
    found ← false
    output("")
    output("Hasil pencarian: ")
    output()
    depend on(kategori)
    kategori is '=' :
      repeat (len(gadget)) times
        if gadget[i][5] is tahun then
          output("Nama" + gadget[i][1])
          output("Deskripsi" + gadget[i][2] )
          output("Jumlah" + gadget[i][3] )
          output("Rarity" + gadget[i][4] )
          output("Tahun ditemukan" + gadget[i][5] )
          found ← true
    kategori is '<':
      repeat (len(gadget)) times
        if gadget[i][5] less than tahun then
          output("Nama" + gadget[i][1])
          output("Deskripsi" + gadget[i][2] )
          output("Jumlah" + gadget[i][3] )
          output("Rarity" + gadget[i][4] )
          output("Tahun ditemukan" + gadget[i][5] )
          found ← true

```

kategori is '>':

```
repeat (len(gadget)) times  
    if gadget[i][5] more than tahun then  
        output("Nama" + gadget[i][1])  
        output("Deskripsi" + gadget[i][2] )  
        output("Jumlah" + gadget[i][3] )  
        output("Rarity" + gadget[i][4] )  
        output("Tahun ditemukan" + gadget[i][5] )
```

kategori is '<=':

```
repeat (len(gadget)) times  
    if gadget[i][5] less or equal than tahun then  
        output("Nama" + gadget[i][1])  
        output("Deskripsi" + gadget[i][2] )  
        output("Jumlah" + gadget[i][3] )  
        output("Rarity" + gadget[i][4] )  
        output("Tahun ditemukan" + gadget[i][5] )
```

kategori is '>=':

```
repeat (len(gadget)) times  
    if gadget[i][5] more or equal than tahun then  
        output("Nama" + gadget[i][1])  
        output("Deskripsi" + gadget[i][2] )  
        output("Jumlah" + gadget[i][3] )  
        output("Rarity" + gadget[i][4] )  
        output("Tahun ditemukan" + gadget[i][5] )  
  
    if found is false then  
        output("Tidak ada gadget yang ditemukan")  
        output("")
```

except ValueError:

```
    output("Tahun yang diinputkan harus berupa bilangan bulat")  
    output("")
```

else

break

6. F05 - Menambah item

{ Menambahkan data item baru ke database

I.S. : matriks data gadget dan consumable telah terdefinisi

F.S. : data item baru dimasukkan ke dalam database }

KAMUS LOKAL

lanjut, isNumber : boolean

nama, deskripsi : string

jumlah, tahun : integer

rarity : character

arrTambahItem : any of data_gadget or data_consumable


```
{ function / procedure }
function IDItemAda(data : array of any of data_user or data_gadget or data_consumable or
data_gadget_return_history or data_gadget_borrow_histroy or
data_consumable_history or data_inventory_user,
ID : string) -> boolean
```

ALGORITMA

```
{ Validasi ID }
lanjut ← false
while not lanjut do
  ID ← input("Masukkan ID: ")
  if (ID[0] = 'G') then
    if IDItemAda(gadget,ID) then
      output("Gagal menambahkan item karena ID sudah ada.")
    else:
      lanjut ← true
  elif (ID[0] = 'C') then
    if IDItemAda(consumable,ID) then
      output("Gagal menambahkan item karena ID sudah ada.")
    else
      lanjut ← true
  else
    { asumsi ID diawali huruf besar (kapitalisasi benar) }
    output("Gagal menambahkan item karena ID tidak valid.")

output("Masukkan Nama: "); input(nama)
output("Masukkan Deskripsi: "); input(deskripsi)

{ Validasi jumlah }
isNumber ← false
while not isNumber do
  try
    output("Masukkan Jumlah: "); input(jumlah)
    if jumlah <= 0 then
      output("Jumlah harus bernilai positif")
    else
      isNumber = true
  except
    ValueError
    output("Jumlah harus berupa bilangan integer, silakan masukkan kembali")
    output()

{ Validasi rarity }
isRarity ← false
```

```

while not isRarity do
    output("Masukkan Rarity: "); input(rarity)
    if rarity in "CBAS" then
        isRarity ← true
    else
        output("Rarity harus berupa karakter C, B, A, atau S")
        output()

arrTambahItem ← [ID,nama,deskripsi,jumlah,rarity]
if ID[0] = 'G' then

    { Validasi tahun }
    isTahun ← false
    while not isTahun do
        try:
            output("Masukkan tahun ditemukan: ", tahun); input(tahun)
            break
        except:
            ValueError
            output("Tahun harus berupa bilangan integer, silakan masukkan kembali")
            output()
    { isTahun = true }

    arrTambahItem.append(tahun)
    gadget.append(arrTambahItem)
else
    consumable.append(arrTambahItem)

output("Item telah berhasil ditambahkan ke database.")

```

7. F06 - Menghapus gadget atau consumable

{ Menghapus gadget dari database
 I.S. matriks data gadget terdefinisi
 F.S. data yang diinputkan dihapus dari data gadget }

KAMUS LOKAL

ID, jawaban : string
 urutan : integer
 rolling : boolean
 { function / procedure }
 function validasiYN(jawaban : string) → boolean
 function IDItemAda(data : array of any of data_user or data_gadget or data_consumable or
 data_gadget_return_history or data_gadget_borrow_histroy or
 data_consumable_history or data_inventory_user,

ID : string) -> boolean

{ Function / Procedure }

function validasiYN(jawaban : string) → boolean

{ Memvalidasi input dari user, harus 'Y' atau 'N'

I.S. string terdefinisi

F.S. mengembalikan True jika string adalah 'Y' atau 'N' dan False jika sebaliknya }

function IDItemAda(data : array of any of data_user or data_gadget or data_consumable or data_gadget_return_history or data_gadget_borrow_histroy or data_consumable_history or data_inventory_user, ID : string) -> boolean

{ Mengecek apakah ID ada pada data

I.S. data dan ID terdefinisi

F.S. Mengembalikan True jika ID item ada di data dan False jika sebaliknya }

ALGORITMA

{ Validasi ID }

rolling ← true

while rolling do

output()

output("Masukkan ID item: "); input(ID)

if ID[0] = 'G' then

if IDItemAda(gadget,ID) then

urutan ← cariID(gadget,ID)

output("Apakah anda yakin ingin menghapus " + gadget[urutan][1] + " (Y/N)? ")

input(jawaban)

{ Validasi jawaban }

while not validasiYN(jawaban) do

output("Apakah anda yakin ingin menghapus " + gadget[urutan][1] + " (Y/N)? ")

input(jawaban)

if jawaban = 'Y' then

gadget.pop(urutan)

output()

output("Item telah berhasil dihapus dari database.")

else

output("Item tidak jadi dihapus dari database")

rolling ← false

else

output("Tidak ada item dengan ID tersebut.")

elif ID[0] = 'C' then

if IDItemAda(consumable,ID) then

urutan ← cariID(consumable,ID)

```

    output("Apakah anda yakin ingin menghapus " + consumable[urutan][1] + " (Y/N)? ")
    input(jawaban)

    { Validasi jawaban }
    while not validasiYN(jawaban) do
        output("Apakah anda yakin ingin menghapus " + gadget[urutan][1] + " (Y/N)? ")
        input(jawaban)

    if jawaban = 'Y' then
        consumable.pop(urutan)
        output()
        output("Item telah berhasil dihapus dari database.")
    else
        output("Item tidak jadi dihapus dari database")
        rolling ← false
    else
        output("Tidak ada item dengan ID tersebut.")
    else
        output("ID yang anda masukkan invalid, ID harus diawali dengan huruf C atau G")
{ rolling = false }

```

8. F07 - Mengubah jumlah gadget atau consumable pada inventory

{ Mengubah jumlah gadget dan consumable yang ada pada database
 I.S. matriks data gadget dan consumable terdefinisi
 F.S. jumlah item pada database berubah }

KAMUS LOKAL

id_item : string
 before, change, indeks_found: integer
 isInteger, found : boolean

ALGORITMA

```

input(id_item)

isInteger ← false
while not isInteger do
    try
        input(change)
        isInteger ← true
    except ValueError
        output("silahkan masukan kembali angka dengan benar")
        output("")

```

```

found ← false
indeks_found ← None

depend on(id_item[0])
  id_item[0] is 'G':
    repeat
      if gadget[i][0] = id_item then
        found ← true
        indeks_found ← i
    until found is true

    if indeks_found is None then
      output("Tidak ada item dengan ID tersebut!")
    else { indeks_found ketemu }
      before ← gadget[indeks_found][3]
      if before + change is less than zero then
        output("{change} {gadget[indeks_found]} gagal dibuang karena stok kurang. Stok
sekarang: {before} (< {change})")
      else { before + change is equal or more than zero }
        gadget[indeks_found][3] ← gadget[indeks_found][3] + change
        if change is equal or more than zero then
          output("Barang berhasil ditambahkan beserta jumlah stok sekarang")
        else { change less than zero }
          output("Barang berhasil dibuang beserta jumlah stok sekarang")
  id_item[0] is 'C':
    repeat
      if consumable[i][0] = id_item then
        found ← true
        indeks_found ← i
    until found is true

    if indeks_found is None then
      output("Tidak ada item dengan ID tersebut!")
    else { indeks_found ketemu }
      before ← consumable[indeks_found][3]
      if before + change is less than zero then
        output("{change} {consumable[indeks_found]} gagal dibuang karena stok kurang.
Stok sekarang: {before} (< {change})")
      else { before + change is equal or more than zero }
        consumable[indeks_found][3] ← consumable[indeks_found][3] + change
        if change is equal or more than zero then
          output("Barang berhasil ditambahkan beserta jumlah stok sekarang")
        else { change less than zero }
          output("Barang berhasil dibuang beserta jumlah stok sekarang")

```

else
 output("Tidak ada item dengan ID tersebut!")

9. F08 - Meminjam gadget

{ Meminjam gadget sesuai id_item yang dimasukan dan akan mengurangi jumlah pada gadget dan menambahkan entri pada gadget_borrow_history
I.S. matriks data gadget dan gadget_borrow_history terdefinisi
F.S. jumlah gadget pada data gadget berkurang dan terdapat entri baru pada gadget_borrow_history }

KAMUS LOKAL

id_item, id_peminjaman, data_string : string
condition, found, syarat_terpenuhi_1, terms : boolean
indeks, current_amount, amount, a : integer
personal_borrow : array of string

ALGORITMA

```
{ Validasi ID_Item }  
condition ← true  
while condition is true do  
    try  
        id_item ← input("Masukan ID item: ")  
        found ← false  
        i traversal [0..len(gadget) -1]  
        if gadget[i][0] is id_item:  
            found ← true  
            condition ← false  
            indeks ← i  
        if found is false then  
            output("Tidak ada item dengan ID tersebut. Silahkan masukan kembali ID item yang sesuai")  
            output("")  
        except ValueError then  
            output("")  
  
{ Membuat array dari setiap gadget yang user pernah pinjam }  
personal_borrow ← []  
a traversal [0..len(gadget_borrow_history) -1]  
    if gadget_borrow_history[a][1] is equal to idUser then  
        personal_borrow.append(gadget_borrow_history[a])  
  
{ Mengecek apakah user pernah meminjam dan belum mengembalikan gadget yang sama atau user belum pernah meminjam sama sekali dari gadget dengan id_item  
Syarat 1: User sudah mengembalikan gadget dengan id_item yang dimasukan secara utuh (tidak
```

```

sebagian) }
syarat_terpenuhi_1 ← false
i traversal [len(personal_borrow)-1..0]
  if personal_borrow[i][2] is equal to id_item then
    if personal_borrow[i][5] is true then
      syarat_terpenuhi_1 ← true
      break
{ Syarat 2: User belum pernah sama sekali meminjam gadget dengan id_item inputan }
check ← None
i traversal [0..len(personal_borrow) -1]
  if personal_borrow[i][2] is equal to id_item then
    check ← 'Checked'

{ Jika user sudah pernah mengembalikan secara lengkap gadget dengan id = id_item (atau)
  gadget dengan id = id item belum pernah ia pinjam sama sekali }
if syarat_terpenuhi_1 is True or check is None then
  { Validasi Tanggal }
  kondisi ← true
  while kondisi is true do
    format ← "%d/%m/%Y"
    output("Tanggal peminjaman: "); input(date_string)

    cond ← false
    if len(date_string) is equal to 10 then
      cond ← true

    if cond is false then
      while true do
        output("Masukan tanggal dengan benar, yakni 2 digit tanggal, 2 digit bulan, dan 4 digit
          tahun dan format DD/MM/YYYY")
        output("Tanggal peminjaman: "); input(date_string)
        if len(date_string) is equal to 10 then
          cond ← true
          break

      try
        datetime.datetime.strptime(date_string, format)
      break
    except ValueError
      output("Tanggal yang anda masukan salah. Silahkan masukan kembali tanggal dengan
        format DD/MM/YYYY")

  { Validasi jumlah }
  current_amount ← gadget[indeks][3]
  terms ← true
  while terms is true do

```

```

try
    output("Jumlah peminjaman: "); input(amount)

    if (amount <= current_amount) and (amount > 0) then
        gadget[indeks][3] ← current_amount - amount
        output(f"Item {gadget[indeks][1]} (x{amount}) berhasil dipinjam!")
        output("")
        terms ← false
    else { amount is not valid }
        output(f"Jumlah yang anda ingin pinjam melebihi yang ada dalam stok penyimpanan
            atau anda memasukan angka di bawah 1. Silahkan masukan kembali jumlah
            yang ingin dipinjam dengan maksimal meminjam {current_amount}")
except ValueError
    output("Silahkan masukan kembali jumlah dengan angka yang benar")

{ Memasukan ke data gadget_borrow_history }
id_peminjaman ← 'GBH' + str(len(gadget_borrow_history))
gadget_borrow_history.append([id_peminjaman, idUser, id_item, date_string, amount, false])

{ Kondisi jika user pernah meminjam gadget dengan id = id_item, namun belum
mengembalikannya }
else
    output("Maaf, anda pernah meminjam gadget yang sama dan belum mengembalikannya, anda
        harus mengembalikan secara keseluruhan gadget yang baru saja anda ingin pinjam")
    output("")

```

10. F09 + FB02 - Mengembalikan gadget

{ Mengembalikan gadget yang pernah dipinjam baik sebagian maupun keseluruhan
I.S. matriks data gadget, gadget_borrow_history, dan gadget_return_history terdefinisi
F.S. jumlah gadget pada data gadget berkurang dan terdapat entri baru pada
gadget_borrow_history }

KAMUS LOKAL

id_returned_gadget, id_pengembalian, tanggal : string
syaratnya : boolean
option, indeksnya, markernya, max_returned, total_amount_returned,
total_amount_returned_updated, amount_returned, z, n, an : integer
personal_borrow_not_return, updated_unique_personal_borrow_not_returned : array of string
unique_personal_borrow_not_returned : set of string

ALGORITMA

{ Menampilkan ke user gadget yang pernah ia pinjam }

personal_borrow_not_returned ← []


```

repeat (len(gadget_borrow_history)) times
    if gadget_borrow_history[a][1] is idUser and
        gadget_borrow_history[a][5] is False then
        personal_borrow_not_returned.append(gadget_borrow_history[a][2])

unique_personal_borrow_not_returned ← set(personal_borrow_not_returned)

updated_unique_personal_borrow_not_returned ← list(unique_personal_borrow_not_returned)

{ Kondisi jika user pernah meminjam barang sebelumnya }

if len(updated_unique_personal_borrow_not_returned) more than zero then
    { Menampilkan setiap gadget yang pernah dipinjam oleh user }
    i traversal [0..(len(updated_unique_personal_borrow_not_returned))-1]
        j traversal [0..(len(gadget))-1]
            if updated_unique_personal_borrow_not_returned[i] = gadget[j][0] then
                output(nomor dan nama gadget yang pernah dipinjam)
    { Meminta user memilih opsi item sesuai nomor }
    banyak ← len(updated_unique_personal_borrow_not_returned)
    syaratnya ← true
    while syaratnya is true:
        try
            output(nomor peminjaman); input(option)
            if option more than 0 and option is equal or less than banyak then
                syaratnya ← false
        except ValueError
            output("Silahkan masukan kembali nomor dengan benar")
    { Meminta user memasukan tanggal dan memvalidasinya }
    format ← "%d/%m/%Y"
    while true:
        output("Tanggal pengembalian: ").strip(); input(tanggal)
        try
            datetime.datetime.strptime(tanggal, format)
            break
        except ValueError
            output("Tanggal yang anda masukan salah. Silahkan masukan kembali
                tanggal dengan format DD/MM/YYYY")
            output("")
    { Membuat string dari gadget yang user ingin kembalikan }
    id_returned_gadget ← updated_unique_personal_borrow_not_returned[option - 1]
    { Menelusuri entri gadget yang ingin user kembalikan pada data
        gadget_borrow_history }
    repeat (len(gadget_borrow_history) time from the last element

```

```

    if gadget_borrow_history[z][2] is equal to id_returned_gadget and
    gadget_borrow_history[z][1] is idUser then
        indeksnya ← z
        break
    { Menelusuri data gadget untuk mendapatkan sebagian informasi dari gadget yang
    ingin dikembalikan }
    repeat (len(gadget)) times
        if gadget[n][0] is id_returned_gadget then
            markernya ← n
            break
    { Menghitung jumlah semua yang pernah dikembalikan sebelumnya ditambahkan
    (jika ada) }
    total_amount_returned = 0
    repeat (len(gadget_return_history)) times
        if gadget_return_history[an][1] is equal gadget_borrow_history[indeksnya][0]
        and gadget_return_history[an][4] is 'applicable' then
            total_amount_returned ← total_amount_returned +
            gadget_return_history[an][3]
    { Prompting user memasukan jumlah barang yang ingin ia kembalikan (baik sebagian
    atau keseluruhan) }
    max_returned ← gadget_borrow_history[indeksnya][4] - total_amount_returned
    while true:
        try:
            output(f"Berapa jumlah {gadget[markernya][1]} yang ingin anda kembalikan
            (maksimal {max_returned}): "); input(amount_returned)
            if amount_returned more than 0 and amount_returned equal or less
            than max_returned then
                break
            else
                output("Jumlah tidak sesuai")
                output("")
        except ValueError
            output("Masukkan kembali jumlah dengan benar harus bilangan bulat")
    { Menambahkan entri gadget_return_history }
    id_pengembalian ← 'GRH' + str(len(gadget_return_history))
    gadget_return_history.append([id_pengembalian,
    gadget_borrow_history[indeksnya][0], tanggal, amount_returned, 'applicable'])
    { Total keseluruhan yang pernah dikembalikan sebelumnya ditambah dengan yang
    baru saja hendak dikembalikan }
    total_amount_returned_updated ← total_amount_returned + amount_returned
    { Menambah jumlah pada data gadget sesuai jumlah yang dikembalikan }
    gadget[markernya][3] ← gadget[markernya][3] + amount_returned
    { Mengubah kolom isReturned pada gadget_borrow_history menjadi True jika
    jumlah yang dipinjam sudah dikembalikan utuh }

```

```

        if gadget_borrow_history[indeksnya][4] - total_amount_returned_updated is 0 then
            gadget_borrow_history[indeksnya][5] ← true
            repeat (len(gadget_return_history)) times
                if gadget_return_history[m][1] is equal to
                    gadget_borrow_history[indeksnya][0] then
                        gadget_return_history[m][4] ← 'not applicable'
            { Menampilkan nama gadget dan jumlah yang pernah ia pinjam secara keseluruhan }
            output(item telah dikembalikan)

{ Kondisi jika user belum pernah meminjam barang }

else { Tidak ada barang yang belum dikembalikan atau user belum pernah meminjam sama
sekali }
    output("Anda belum pernah meminjam gadget sama sekali")

```

11. F10 - Meminta consumable

```

{ Meminta consumable yang tersedia pada database
I.S. matriks data consumable, inventory_user, dan consumable_history terdefinisi
F.S. consumable terpinjam dan data consumable, inventory_user, consumable_history telah
diubah }

```

KAMUS LOKAL

```

ID, date_string, id_history : string
kondisinya, ketemu, syrt : boolean
indeks, amount_asked : integer

```

ALGORITMA

```

{ Validasi ID ada }
kondisinya ← true
while kondisinya do
    try
        output("Masukkan ID item: "); input(ID)
        ketemu ← false

        j traversal [1..len(consumable)-1]
            if consumable[j][0] = ID then
                ketemu ← true
                kondisinya ← false
                indeks ← j
                break
        if ketemu = false then
            output("ID item tidak tersedia, mohon inputkan ID yang benar")
    except ValueError:
        output()

```

```

{ Validasi jumlah }
syrt ← true
while syrt do
    try
        output("Jumlah: "); input(amount_asked)
        if amount_asked > 0 and amount_asked <= consumable[indeks][3] then
            syrt ← false
        if syrt = true then
            output(f"Silahkan masukan jumlah dengan benar, minimal 1 dan maksimal
                {consumable[indeks][3]}")
    except ValueError
        output("Masukan jumlah dengan benar dengan bilangan bulat")

{ Validasi tanggal }
kondisi ← true
while(kondisi) do
    format ← "%d/%m/%Y"
    output("Tanggal permintaan: "); input(date_string)

    cond ← false
    if len(date_string) = 10 then
        cond ← true

    if cond = false then
        while(true) do
            output("Masukan tanggal dengan benar, yakni 2 digit tanggal, 2 digit bulan, dan 4 digit
                tahun dan format DD/MM/YYYY")
            output("Tanggal permintaan: "); input(date_string)
            if len(date_string) = 10 then
                cond ← true
                break

    try
        datetime.datetime.strptime(date_string, format)
        break
    except ValueError
        output("Tanggal yang anda masukan salah. Silahkan masukan kembali tanggal dengan
            format DD/MM/YYYY")

{ Mengubah data terbaru }
id_history ← 'CH' + str(len(consumable_history))
consumable_history.append([id_history, idUser, ID, date_string, amount_asked])

consumable [indeks][3] ← consumable[indeks][3] - amount_asked

```

```

pernah ← false
al traversal [0..len(inventoryr_user)-1]
    if ID = inventory_user[al][1] and idUser = inventory_user[al][0] then
        pernah ← true
        final_indeks ← al
        break

if pernah then
    inventory_user[final_indeks][2] ← inventory_user[final_indeks][2] + amount_asked
else { belum pernah }
    inventory_user.append([idUser, ID, amount_asked])

{ Menampilkan ke user bahwa item berhasil diminta }
output()
output(f"Item {consumable[indeks][1]} (x{amount_asked}) telah berhasil diambil!")

```

12. F11 - Melihat riwayat peminjaman gadget

```

{ Menampilkan daftar peminjaman gadget yang telah dilakukan para user ke layar
I.S. matriks data user, gadget, gadget_borrow_history terdefinisi
F.S. tercetak ke layar riwayat peminjaman user }

```

KAMUS LOKAL

```

rolling, bisaLanjut : boolean
count : integer
borrowSort : data_gadget_borrow_history
namaUser, namaGadget, lanjut : string

```

{ Function / Procedure }

function validasiYN(jawaban : string) → boolean

{ Memvalidasi input dari user, harus 'Y' atau 'N'

I.S. string terdefinisi

F.S. mengembalikan True jika string adalah 'Y' atau 'N' dan False jika sebaliknya }

ALGORITMA

```

rolling ← true
count ← 0
while rolling do
    { Mensortir data berdasarkan tanggal, secara descending }
    borrowSort ← sorted(gadget_borrow_history[count+1:], key = lambda date:
                        datetime.datetime.strptime(date[3], '%d/%m/%Y'), reverse=True)
    bisaLanjut ← true
    i traversal [0..4]
        try
            namaUser ← user[cariID(user,borrowSort[i][1))][2]

```

```

namaGadget ← gadget[cariID(gadget,borrowSort[i][2))][1]
output()
output("ID Peminjam      : ", borrowSort[i][1])
output("Nama Pengambil    : ", namaUser)
output("Nama Gadget       : ", namaGadget)
output("Tanggal Peminjaman : ", borrowSort[i][3])
output("Jumlah           : ", borrowSort[i][4])
except
  { Ketika data habis maka akan terjadi IndexError }
  IndexError
  output()
  output("Data sudah habis")
  bisaLanjut ← false
  break

if bisaLanjut and len(borrowSort) != 5 u
  output()
  output("Apakah mau ditampilkan data lebih lanjut? (Y/N) "); input(lanjut)
  { Validasi input }
  while not validasiYN(lanjut) do
    output("Apakah mau ditampilkan data lebih lanjut? (Y/N) "); input(lanjut)
  if lanjut = 'Y' then
    count = count + 5
  else
    rolling ← false
  else
    rolling ← false

```

13. F12 - Melihat riwayat pengembalian gadget

{ Menampilkan daftar pengembalian gadget yang telah dilakukan para user ke layar
 I.S. matriks data user, gadget, gadget_borrow_history terdefinisi
 F.S. tercetak ke layar riwayat peminjaman user }

KAMUS LOKAL

rolling, lanjutkan : boolean
 count : integer
 returnSort : data_gadget_return_histroy
 namaUser, namaGadget, id_user, id_gadget, nextInp : string

{ Function / Procedure }

function validasiYN(jawaban : string) → boolean

{ Memvalidasi input dari user, harus 'Y' atau 'N'

I.S. string terdefinisi

F.S. mengembalikan True jika string adalah 'Y' atau 'N' dan False jika sebaliknya }

ALGORITMA

```
rolling ← true
count ← 0
while rolling do
    { Mensortir data berdasarkan tanggal, secara descending }
    returnSort ← sorted(gadget_return_history[count+1:], key = lambda date:
                        datetime.datetime.strptime(date[2], '%d/%m/%Y'),reverse=True)
    lanjutkan ← true
    i traversal [0..4]
        try
            { cari id gadget dan id user }
            line traversal [0..len(gadget_borrow_history) - 1]
            if returnSort[i][1] = gadget_borrow_history[line][0] then
                id_gadget ← gadget_borrow_history[line][2]
                id_user ← gadget_borrow_history[line][1]

            namaUser ← user[cariID(user,id_user)][2]
            namaGadget ← gadget[cariID(gadget,id_gadget)][1]

            output()
            output("ID Pengembalian : ", returnSort[i][0])
            output("Nama Pengambil : ", namaUser)
            output("Nama Gadget : ", namaGadget)
            output("Tanggal Pengembalian : ", returnSort[i][2])
            output("Jumlah : ", returnSort[i][3])
        except
            { Ketika data habis maka akan terjadi IndexError }
            IndexError
            output()
            output("Data sudah habis")
            lanjutkan ← false
            break
    if lanjutkan and len(returnSort) != 5 then
        output()
        output("Apakah mau ditampilkan data lebih lanjut? (Y/N) "); input(nextInp)
        { Validasi input }
        while not validasiYN(nextInp) do
            output("Apakah mau ditampilkan data lebih lanjut? (Y/N) "); input(nextInp)
        if nextInp = 'Y' then
            count ← count + 5
        else
            rolling ← false
    else
```

rolling ← false

14. F13 - Melihat riwayat pengambilan consumable

{ Menampilkan daftar pengambilan consumable yang telah dilakukan para user ke layar
I.S. matriks data user, gadget, gadget_borrow_history terdefinisi
F.S. tercetak ke layar riwayat peminjaman user }

KAMUS LOKAL

rolling, berikutnya : boolean
count : integer
consumableSort : data_consumable_history
namaUser, namaConsumable : string

{ Function / Procedure }

function validasiYN(jawaban : string) → boolean

{ Memvalidasi input dari user, harus 'Y' atau 'N'

I.S. string terdefinisi

F.S. mengembalikan True jika string adalah 'Y' atau 'N' dan False jika sebaliknya }

ALGORITMA

rolling ← true

count ← 0

while rolling do

{ Mensortir data berdasarkan tanggal, secara descending }

consumableSort ← sorted(consumable_history[count+1:], key = lambda date:
datetime.datetime.strptime(date[3], '%d/%m/%Y'),reverse=True)

berikutnya ← true

i traversal [0..4]

try

namaUser ← user[cariID(user,consumableSort[i][1])[2]

namaConsumable ← consumable[cariID(consumable,consumableSort[i][2])[1]

output()

output("ID Pengambilan : ", consumableSort[i][1])

output("Nama Pengambil : ", namaUser)

output("Nama Consumable : ", namaConsumable)

output("Tanggal Pengambilan : ", consumableSort[i][3])

output("Jumlah : ", consumableSort[i][4])

except

{ Ketika data habis maka akan terjadi IndexError }

output(i)

IndexError

output()

output("Data sudah habis")

berikutnya ← false


```

        break
    if berikutnya and len(consumableSort) != 5 then
        output()
        output("Apakah mau ditampilkan data lebih lanjut? (Y/N) "); input(nextInp)
        { Validasi input }
        while not validasiYN(nextInp) do
            output("Apakah mau ditampilkan data lebih lanjut? (Y/N) "); input(lanjut)
            if nextInp = 'Y' then
                count = count + 5
            else
                rolling ← false
        else
            rolling ← false

```

15. F14 - Load data

```

{ Membaca file-file csv pada folder yang diinputkan
  I.S. pada folder terdapat file-file csv yang dibutuhkan
  F.S. didapatkan matriks data sesuai dengan file-file csv yang ada }

```

KAMUS LOKAL

```

user : array of data_user
gadget : array of data_gadget
consumable : array of data_consumable
consumable_history : array of data_consumable_history
gadget_borrow_history : array of data_gadget_borrow_history
gadget_return_history : array of data_gadget_return_history
inventory_user : array of data_inventory_user

```

```

{ Variable global }

```

```

global user
global gadget
global consumable
global consumable_history
global gadget_borrow_history
global gadget_return_history
global inventory_user

```

```

{ Function / Procedure }

```

```

function load_data(file : csv) → array of array of string
{ Membaca file csv dan mengembalikan matriks data sesuai data csv
  I.S. file terdefinisi
  F.S. dikembalikan matriks data sesuai file }

```

```

function tryChange(data : array of array of string) → any of data_user or data_gadget or

```

data_consumable or data_gadget_return_history or
data_gadget_borrow_histroy or
data_consumable_history or data_inventory_user

{ Mengubah tipe data yang ada pada data menjadi boolean atau integer jika dimungkinkan
I.S. data terdefinisi
F.S. dikembalikan data dimana elemen yang dapat diubah, diubah menjadi boolean atau integer
telah diubah }

ALGORITMA

os.chdir('./' + str(folder))

```
user ← tryChange(load_data("user.csv"))
gadget ← tryChange(load_data("gadget.csv"))
consumable ← tryChange(load_data("consumable.csv"))
consumable_history ← tryChange(load_data("consumable_history.csv"))
gadget_borrow_history ← tryChange(load_data("gadget_borrow_history.csv"))
gadget_return_history ← tryChange(load_data("gadget_return_history.csv"))
inventory_user ← tryChange(load_data("inventory_user.csv"))
os.chdir('./')
```

16. F15 - Save data

{ Menyimpan data ke dalam file di folder yang diinputkan
I.S. user, gadget, consumable, gadget_borrow_history, gadget_return_history,
consumable_history, inventory_user terdefinisi
F.S. file user, gadget, consumable, gadget_borrow_history, gadget_return_history,
consumable_history, inventory_user tersimpan }

KAMUS LOKAL

parent, directory, path : string

{ Function / Procedure }

procedure save_data (file : csv, data : any of data_user or data_gadget or data_consumable or
data_gadget_return_history or data_gadget_borrow_histroy or
data_consumable_history or data_inventory_user)

{ Menulis data ke dalam file csv

I.S. data terdefinisi

F.S. jika sebelumnya file sudah ada, maka di file diperbarui, dan jika belum ada, maka tertulis
file baru }

ALGORITMA

parent ← os.getcwd()

{ Validasi directory }

checkedDir ← false

```

while not checkedDir do
    output("Masukkan nama folder penyimpanan : "); input(directory)
    checkedDir ← true
    i traversal [0..len(directory)-1]
        if directory[i] in ("\:*?"<>|') then
            checkedDir ← false
            output('nama folder anda mengandung \:*?"<>|')
            output()

path ← os.path.join(parent, directory)

try
    { Membuat folder baru jika folder belum ada }
    os.mkdir(path)
except
    FileExistsError

    { Berpindah directory ke dalam folder csv }
    os.chdir('./' + directory)
    output()
    output("Saving...")
    time.sleep(2)

    { Menyimpan file-file csv }
    save_data("user.csv",user)
    save_data("gadget.csv",gadget)
    save_data("consumable.csv",consumable)
    save_data("gadget_borrow_history.csv",gadget_borrow_history)
    save_data("gadget_return_history.csv",gadget_return_history)
    save_data("consumable_history.csv",consumable_history)
    save_data("inventory_user.csv",inventory_user)

    output("Data telah disimpan pada folder " + Bold(directory))
    { Berpindah directory ke folder parent }
    os.chdir('./')

```

17. F16 - Help

```

{ Menampilkan keyword-keyword yang tersedia ke layar
  I.S. sembarang
  F.S. tercetak list keyword ke layar }

```

KAMUS LOKAL

-

ALGORITMA

output(""

===== HELP

=====

Berikut merupakan keyword yang dapat digunakan beserta fungsi dan aksesnya

Ketikkan keyword di bawah ini untuk melakukan fungsi yang diinginkan

```
> register => melakukan registrasi user baru (Akses: Admin)
> login => melakukan login ke dalam program (Akses: Admin/User)
> carirarity => mencari gadget berdasarkan rarity yang diinputkan (Akses: Admin/User)
> caritahun => mencari gadget berdasarkan tahun ditemukan (Akses: Admin/User)
> tambahitem => menambahkan item (gadget/consumable) ke dalam database (Akses: Admin)
> hapusitem => menghapus item (gadget/consumable) dari database (Akses: Admin)
> ubahjumlah => mengubah jumlah gadget/consumable pada database (Akses: Admin)
> pinjam => meminjam gadget dari database dan memasukkan ke dalam inventar (Akses: User)
> kembalikan => mengembalikan gadget yang dipinjam (Akses: User)
> minta=> meminta consumable dari database dan memasukkan ke dalam inventory (Akses:User)
> riwayatpinjam => melihat record peminjaman gadget yang tersortir berdasar tanggal
(Akses: Admin)
> riwayatkembali => melihat record pengembalian gadget yang tersortir berdasar tanggal
(Akses: Admin)
> riwayatambil => melihat record pengambilan consumable yang tersortir berdasar tanggal
(Akses: Admin)
> save => menyimpan data setelah dilakukan perubahan (Akses: Admin/User)
> gacha => menggacha consumable yang ada di inventory untuk mendapatkan consumable baru
(Akses: User)
> help => memberikan panduan penggunaan sistem (Tidak perlu login)
> exit => keluar dari program (Tidak perlu login)
""")
```

18. F17 - Exit

```
{ Menutup dan keluar dari program
I.S. program sedang berjalan
F.S. program ditutup dan selesai }
```

KAMUS LOKAL

```
isSave : string
{ global variable }
global program
{ function / procedure }
function validasiYN(jawaban : string) -> boolean
procedure save()
```

```
{ Function / Procedure }
```

function validasiYN(jawaban : string) → boolean
 { Memvalidasi input dari user, harus 'Y' atau 'N'
 I.S. string terdefinisi
 F.S. mengembalikan True jika string adalah 'Y' atau 'N' dan False jika sebaliknya }

procedure save()
 { Menyimpan data ke dalam file di folder yang diinputkan
 I.S. user, gadget, consumable, gadget_borrow_history, gadget_return_history, consumable_history, inventory_user terdefinisi
 F.S. file user, gadget, consumable, gadget_borrow_history, gadget_return_history, consumable_history, inventory_user tersimpan }

ALGORITMA

if hasLogin then
 output("Apa Anda mau menyimpan file yang sudah diubah?(Y/N)"); input(isSave)
 while not validasiYN(isSave) do
 output("Apa Anda mau menyimpan file yang sudah diubah? (Y/N) "); input(isSave)
 if isSave = "Y" then
 save()
 output()
 output("Terima kasih telah menggunakan kantong ajaib ^_^")
 { Menghentikan program }
 program ← false

19. FB1 - Hashing

{ Melakukan hashing pada password dengan metode RollingHash yang bersifat satu arah
 I.S. string password terdefinisi
 F.S. string password telah dilakukan hashing }

KAMUS LOKAL

P, m, powerOfP, hashed, i : integer

ALGORITMA

P ← 101
 m ← 1e9 + 1
 powerOfP ← 1
 hashed ← 0
i traversal [0..len(str)-1]
 hashed ← ((hashed + (ord(str[i]) - ord('!') + 1) * powerOfP) mod m)
 powerOfP ← (powerOfP * P) mod m
 return int(hashed)

20. FB03 - Meningkatkan rarity consumable

{ Mendapatkan consumable baru yang rarity-nya mungkin lebih tinggi / lebih rendah

I.S. list lstChance, dan data inventory_user, consumable_history, consumable terdefinisi
F.S. User mendapatkan consumable baru }

KAMUS LOKAL

count, urutan, digunakan, urutanInventory, jumlah : integer

IDInventory : array of string

digunakanBenar, jumlahBenar, finished : boolean

perintah, rarity : character

tambah_con_history : data_consumable_history

tambah_inventory : data_inventory_user

perintah : string

{ Variable global }

global lstChance

global inventory_user

global consumable_history

global consumable

{ Function / Procedure }

function cariData(data: any of data_user or data_gadget or data_consumable or

data_gadget_return_history or data_gadget_borrow_histroy or

data_consumable_history or data_inventory_user, ID : string, index : integer) → integer

{ Mencari "dicari" di dalam data berdasarkan index kolomnya

I.S. data, dicari, dan index terdefinisi

F.S. dikembalikan index baris dimana "dicari" berada pada data }

function cariID(data : any of data_user or data_gadget or data_consumable or

data_gadget_return_history or data_gadget_borrow_histroy or

data_consumable_history or data_inventory_user, ID : string) -> integer

{ Mencari index dimana ID adaa pada data

I.S. data dan ID terdefinisi

F.S. dikembalikan index dimana ID berada pada data }

function Bold(text : string) → string

{ Mengubah text menjadi terlihat bold jika di-print

I.S. text terdefinisi

F.S. text diberi 'kode' yang jika di-print text menjadi terlihat bold }

function chance(lstRarity : array of float, rarity : character) → array of float

{ Menghasilkan array peluang untuk mendapatkan rarity tertentu berdasarkan rarity consumable yang digunakan untuk gacha

I.S. lstRarity dan rarity terdefinisi

F.S. dikembalikan lstRarity yang nilainya telah diubah }

```

function validasiYN(jawaban : string) → boolean
{ Memvalidasi input dari user, harus 'Y' atau 'N'
  I.S. string terdefinisi
  F.S. mengembalikan True jika string adalah 'Y' atau 'N' dan False jika sebaliknya }

```

```

function hasilGacha(lstChance : array of float) → character
{ Mengembalikan rarity berdasarkan array peluang
  I.S. lstChance terdefinisi
  F.S. dikembalikan rarity yang didapatkan secara acak berdasarkan array peluang }

```

ALGORITMA

```

output()
output("=====INVENTORY=====")
count ← 0
IDInventory ← []
i traversal [0..len(inventory_user)-1]
  if inventory_user[i][0] = idUser then
    count ← count + 1
    urutan ← cariID(consumable,inventory_user[i][1])
    output(str(count) + ". " + consumable[urutan][1] + "(Rarity " + consumable[urutan][4] + " )
      (" + str(inventory_user[i][2]) + " )")
    IDInventory.append(inventory_user[i][1])
if count = 0 then
  output("Anda tidak mempunyai consumable di inventory Anda")
  output("=====")
  output()
  return
output("=====")
output()

{ Validasi consumable yang digunakan }
digunakanBenar ← false
while not digunakanBenar do
  try
    output("Pilih consumable yang mau digunakan: "); input(digunakan)
    if digunakan > count then
      output("Input anda tidak valid")
      output()
    else
      digunakanBenar ← true
  except
    ValueError
    output("Pilihan harus berupa bilangan bulat")
    output()

```

```

urutanInventory ← cariData(inventory_user, IDInventory[digunakan-1], 1)

{ Validasi jumlah consumable yang digunakan }
jumlahBenar ← false
while not jumlahBenar do
  try
    output("Jumlah yang digunakan: "); input(jumlah)
    if jumlah > inventory_user[urutanInventory][2] then
      output("Jumlah yang diinputkan terlalu banyak")
      output()
    else
      jumlahBenar ← true
  except
    ValueError
    output("Pilihan harus berupa bilangan bulat")
    output()

{ Mengubah data consumable, dan inventory_user }
urutan ← cariID(consumable, IDInventory[digunakan - 1])
consumable[urutan][3] ← consumable[urutan][3] + jumlah
inventory_user[urutanInventory][2] ← inventory_user[urutanInventory][2] - jumlah
if inventory_user[urutanInventory][2] = 0 then
  inventory_user.pop(urutanInventory)
output(Bold(consumable[urutan][1]) + " (" + Bold("x" + str(jumlah)) + ") ditambahkan!")

{ Menampilkan peluang per rarity ke layar }
lstChance ← chance(lstChance, consumable[urutan][4])
output("Chance mendapatkan Rarity ", end="")
if lstChance[0] != 0 then
  output(Bold('C') + " (" + Bold("{:.2f}".format(lstChance[0])) + "%) ", end="")
if lstChance[1] != 0 then
  output(Bold('B') + " (" + Bold("{:.2f}".format(lstChance[1])) + "%) ", end="")
if lstChance[2] != 0 then
  output(Bold('A') + " (" + Bold("{:.2f}".format(lstChance[2])) + "%) ", end="")
if lstChance[3] != 0 then
  output(Bold('S') + " (" + Bold("{:.2f}".format(lstChance[3])) + "%)", end="")
output()

output("Tambahkan item lagi? (Y/N): "); input(perintah)
{ Validasi perintah }
while not validasiYN(perintah) do
  output("Tambahkan item lagi? (Y/N): "); input(perintah)

if perintah = 'Y' then

```



```

    gacha()
else
    output()
    output("Rolling...")
    time.sleep(3)
    output()
    rarity ← hasilGacha(1stChance)
    finished ← false

while not finished do
    i traversal [0..(len(consumable))-1]
    if consumable[i][4] = rarity then
        output("Selamat Anda mendapatkan " + Bold(consumable[i][1]) + " (Rarity " + rarity +
            ") sebanyak x" + Bold(str(consumable[i][3])) + "!")
        tambah_con_history ← ["CH" + str(len(consumable_history)), idUser,
            consumable[i][0], datetime.date.today().strftime("%d/%m/%Y"),
            consumable[i][3]]
        consumable_history.append(tambah_con_history)
        tambah_inventory ← [idUser, consumable[i][0], consumable[i][3]]
        inventory_user.append(tambah_inventory)
        consumable[i][3] ← 0
        1stChance ← [0,0,0,0]
        finished ← true
    return
    { Bila tidak ada data consumable yang memiliki rarity seperti yang didapatkan }
    rarity ← hasilGacha(1stChance)

```

VIII. HASIL PENGUJIAN PROGRAM

Tabel 04: Hasil Pengujian Program

Fitur	Hasil Pengujian
F01 - Register	<pre> >>> register Masukkan nama: suneo Masukkan username: nobita Username telah digunakan oleh user lain Silakan input username yang berbeda Masukkan username: suneyoww Masukkan password: suneo123 Masukkan alamat: Jalan Ganesha User suneyoww telah berhasil register ke dalam Kantong Ajaib. >>> </pre>
F02 - Login	<pre> Selamat datang di "Kantong Ajaib!" >>> login Masukkan username: nobita Masukkan password: nobita123 Username atau password Anda tidak cocok Silakan masukkan kembali username dan password Masukkan username: nobita Masukkan password: akunobi Selamat datang Nobi Nobita ^_^ >>> </pre>
F03 - Pencarian Gadget Berdasarkan Rarity	<pre> >>> carirarity Masukkan rarity: X Hasil pencarian: Tidak ada gadget yang memiliki rarity X >>> carirarity Masukkan rarity: S Hasil pencarian: Nama : Anywhere Door Deskripsi : Pintu kemana saja Jumlah : 1 Rarity : S Tahun ditemukan : 2121 Nama : 4D Pocket Deskripsi : Kantong tapi 4D hehe Jumlah : 2 Rarity : S Tahun ditemukan : 2210 </pre>

F04 - Pencarian Gadget
Berdasarkan Tahun
Ditemukan

```
>>> caritahun
Masukkan tahun: 2000
Masukkan kategori: =
```

Hasil pencarian:

```
Nama: Small Light
Deskripsi: Buat ngecilin barang
Jumlah: 10
Rarity: A
Tahun Ditemukan: 2000
```

```
>>> caritahun
Masukkan tahun: 2210
Masukkan kategori: <
```

Hasil pencarian:

```
Nama: ini gadget
Deskripsi: no desc
Jumlah: 5
Rarity: C
Tahun Ditemukan: 2002
```

```
>>> caritahun
Masukkan tahun: 2100
Masukkan kategori: >
```

Hasil pencarian:

```
Nama: Anywhere Door
Deskripsi: Pintu kemana saja
Jumlah: 1
Rarity: S
Tahun Ditemukan: 2121
```

```
Nama: 4D Pocket
Deskripsi: Kantong tapi 4D hehe
Jumlah: 2
Rarity: S
Tahun Ditemukan: 2210
```

```
Nama: Time Cloth
Deskripsi: Buat baju
Jumlah: 6
Rarity: C
Tahun Ditemukan: 2122
```

```
Nama: Big Light
Deskripsi: Buat gedein barang
Jumlah: 3
Rarity: B
Tahun Ditemukan: 2222
```

```
Nama: Anywhere Door
Deskripsi: Pintu kemana saja
Jumlah: 1
Rarity: S
Tahun Ditemukan: 2121
```

```
Nama: Translation Jelly
Deskripsi: Jelly buat translate
Jumlah: 10
Rarity: A
Tahun Ditemukan: 2100
```

```
Nama: Takecopter
Deskripsi: Buat terbang ngab
Jumlah: 5
Rarity: B
Tahun Ditemukan: 2001
```

```
Nama: Time Cloth
Deskripsi: Buat baju
Jumlah: 6
Rarity: C
Tahun Ditemukan: 2122
```

```
Nama: Small Light
Deskripsi: Buat ngecilin barang
Jumlah: 10
Rarity: A
Tahun Ditemukan: 2000
```

F05 - Menambah Item	<pre> >>> tambahitem Masukkan ID: C6 Gagal menambahkan item karena ID sudah ada. Masukkan ID: C7 Gagal menambahkan item karena ID sudah ada. Masukkan ID: C10 Masukkan Nama: ayam Masukkan Deskripsi: ayam goreng Masukkan Jumlah: 1 Masukkan Rarity: 5 Item telah berhasil ditambahkan ke database. </pre>
F06 - Menghapus Gadget atau Consumable	<pre> >>> hapusitem Masukkan ID item: C10 Apakah anda yakin ingin menghapus ayam (Y/N)? Y Item telah berhasil dihapus dari database. </pre>
F07 - Mengubah Jumlah Gadget	<pre> >>> ubahjumlah Masukan ID: G2 Masukkan Jumlah: 150 150 Anywhere Door berhasil ditambahkan. Stok sekarang: 240 >>> ubahjumlah Masukan ID: -200 Masukkan Jumlah: -200 Tidak ada item dengan ID tersebut! >>> ubahjumlah Masukan ID: G2 Masukkan Jumlah: -200 200 Anywhere Door berhasil dibuang. Stok sekarang: 40 >>> ubahjumlah Masukan ID: G2 Masukkan Jumlah: -200 -200 Anywhere Door gagal dibuang karena stok kurang. Stok sekarang: 40 (< -200) </pre>
F08 - Meminjam Gadget	<pre> >>> pinjam Masukan ID item: G1 Tanggal peminjaman: 02/05/2021 Jumlah peminjaman: 10 Jumlah yang anda ingin pinjam melebihi yang ada dalam stok penyimpanan atau anda memasukan angka di bawah 1. Silahkan masukan kembali jumlah yang ingin dipinjam dengan maksimal meminjam 5 Jumlah peminjaman: 2 Item ini gadget (x2) berhasil dipinjam! </pre>
F09 - Mengembalikan Gadget	<pre> >>> kembalikan 1. Anywhere Door Masukan nomor peminjaman: 1 Tanggal pengembalian: 02/05/2021 Berapa jumlah Anywhere Door yang ingin anda kembalikan (maksimal 1): 1 Item Anywhere Door (x1) telah dikembalikan </pre>

F10 - Meminta Consumable	<pre> >>> minta Masukkan ID item: C10 ID item tidak tersedia, mohon inputkan ID yang benar Masukkan ID item: C5 Jumlah: 1000000 Silahkan masukan jumlah dengan benar, minimal 1 dan maksimal 34 Jumlah: 34 Tanggal permintaan: 40/05/2021 Tanggal yang anda masukan salah. Silahkan masukan kembali tanggal dengan format DD/MM/YYYY Tanggal permintaan: 04/05/2021 Item Sukiyaki (x34) telah berhasil diambil! </pre>
F11 - Melihat Riwayat Peminjaman Gadget	<pre> >>> riwayatpinjam ID Peminjam : U2 Nama Pengambil : Sutetsu Nama Gadget : Small Light Tanggal Peminjaman : 15/04/2021 Jumlah : 10 ID Peminjam : U2 Nama Pengambil : Sutetsu Nama Gadget : ini gadget Tanggal Peminjaman : 15/04/2021 Jumlah : 5 ID Peminjam : U1 Nama Pengambil : Willy Wonka Nama Gadget : Time Cloth Tanggal Peminjaman : 14/04/2021 Jumlah : 6 Data sudah habis </pre>
F12 - Melihat Riwayat Pengembalian Gadget	<pre> >>> riwayatkembali ID Pengembalian : GRH2 Nama Pengambil : Sutetsu Nama Gadget : Small Light Tanggal Pengembalian : 16/04/2021 Jumlah : 5 ID Pengembalian : GRH1 Nama Pengambil : Willy Wonka Nama Gadget : Time Cloth Tanggal Pengembalian : 15/04/2021 Jumlah : 6 Data sudah habis </pre>

<p>F13 - Melihat Riwayat Pengembalian Consumable</p>	<pre> >>> riwayatambil Anda harus login terlebih dahulu >>> login Masukkan username: malikakbar Masukkan password: mabar Selamat datang Malik Akbar ^_^ >>> riwayatambil ID Pengambilan : U2 Nama Pengambil : Sutetsu Nama Consumable : Japanese Soba Tanggal Pengambilan : 30/04/2021 Jumlah : 6 ID Pengambilan : U2 Nama Pengambil : Sutetsu Nama Consumable : Ramen Tanggal Pengambilan : 15/04/2021 Jumlah : 50 ID Pengambilan : U2 Nama Pengambil : Sutetsu Nama Consumable : Japanese Soba Tanggal Pengambilan : 15/04/2021 Jumlah : 4 ID Pengambilan : U3 Nama Pengambil : Rere Roro Nama Consumable : permen Tanggal Pengambilan : 14/04/2021 Jumlah : 1000 4 Data sudah habis >>> </pre>
<p>F14 - Load Data</p>	<pre> PS R:\TPB STEI SMT 2 2020\Daspro\Tubes Daspro\TBIF1210-09-11> python main.py -f csvBaruku Loading... Kantong Ajaib Selamat datang di "Kantong Ajaib!" </pre>
<p>F15 - Save Data</p>	<pre> >>> save Masukkan nama folder penyimpanan : csvBaruku Saving... Data telah disimpan pada folder csvBaruku >>> </pre>

F16 - Help	<pre>>>> help ===== HELP ===== Berikut merupakan keyword yang dapat digunakan beserta fungsi dan aksesnya Ketikkan keyword di bawah ini untuk melakukan fungsi yang diinginkan > register => melakukan registrasi user baru (Akses: Admin) > login => melakukan login ke dalam program (Akses: Admin/User) > carirarity => mencari gadget berdasarkan rarity yang diinputkan (Akses: Admin/User) > caritahun => mencari gadget berdasarkan tahun ditemukan (Akses: Admin/User) > tambahitem => menambahkan item (gadget/consumable) ke dalam database (Akses: Admin) > hapusitem => menghapus item (gadget/consumable) dari database (Akses: Admin) > ubahjumlah => mengubah jumlah gadget/consumable pada database (Akses: Admin) > pinjam => meminjam gadget dari database dan memasukkan ke dalam inventory (Akses: User) > kembalikan => mengembalikan gadget yang dipinjam (Akses: User) > minta => meminta consumable dari database dan memasukkan ke dalam inventory (Akses: User) > riwayatpinjam => melihat record peminjaman gadget yang tersortir berdasar tanggal (Akses: Admin) > riwayatkembali => melihat record pengembalian gadget yang tersortir berdasar tanggal (Akses: Admin) > riwayatambil => melihat record pengambilan consumable yang tersortir berdasar tanggal (Akses: Admin) > save => menyimpan data setelah dilakukan perubahan (Akses: Admin/User) > gacha => menggacha consumable yang ada di inventory untuk mendapatkan consumable baru (Akses: User) > help => memberikan panduan penggunaan sistem (Tidak perlu login) > exit => keluar dari program</pre>
F17 - Exit	<pre>>>> exit Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (Y/N) █</pre>
FB01 - Hashing	<pre>Masukkan password: mabar Password yang telah di-hash adalah 600602347</pre>
FB02 - Mengembalikan Gadget secara Parsial	<pre>>>> kembalikan 1. ini gadget 2. Anywhere Door Masukan nomor peminjaman: 1 Tanggal pengembalian: 02/05/2021 Berapa jumlah ini gadget yang ingin anda kembalikan (maksimal 2): 1 Item ini gadget (x1) telah dikembalikan >>> kembalikan 1. ini gadget 2. Anywhere Door Masukan nomor peminjaman: 1 Tanggal pengembalian: 03/05/2021 Berapa jumlah ini gadget yang ingin anda kembalikan (maksimal 1): 1 Item ini gadget (x1) telah dikembalikan >>> █</pre>

FB03 - Meningkatkan
Rarity Consumables

```
>>> gacha

=====INVENTORY=====
1. permen(Rarity C) (1)
2. Sukiyaki(Rarity C) (34)
3. Ramen(Rarity B) (100)
=====

Pilih consumable yang mau digunakan: 2
Jumlah yang digunakan: 30
Sukiyaki (x30) ditambahkan!
Chance mendapatkan Rarity C (90.00%) B (10.00%)
Tambahkan item lagi? (Y/N): Y

=====INVENTORY=====
1. permen(Rarity C) (1)
2. Sukiyaki(Rarity C) (4)
3. Ramen(Rarity B) (100)
=====

Pilih consumable yang mau digunakan: 3
Jumlah yang digunakan: 50
Ramen (x50) ditambahkan!
Chance mendapatkan Rarity C (50.00%) B (45.00%) A (5.00%)
Tambahkan item lagi? (Y/N): Y

=====INVENTORY=====
1. permen(Rarity C) (1)
2. Sukiyaki(Rarity C) (4)
3. Ramen(Rarity B) (100)
=====

Pilih consumable yang mau digunakan: 1
Jumlah yang digunakan: 1
permen (x1) ditambahkan!
Chance mendapatkan Rarity C (70.00%) B (27.50%) A (2.50%)
Tambahkan item lagi? (Y/N): N

Rolling...

Selamat Anda mendapatkan Ramen (Rarity B) sebanyak x50!
```


IX. CATATAN DAN LAMPIRAN

Asistensi 1

**Form MoM Asistensi Tugas Besar
IF1210/Dasar Pemrograman
Sem. 2 2020/2021**

Nomor Asistensi : 1
No. Kelompok/Kelas : 11/K09
Tanggal asistensi : 11 April 2021

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
1	16520129/Farisa Aliya
2	16520149/Gerald Abraham Sianturi
3	16520299/Malik Akbar Hashemi Rafsanjani
4	16520329/Hughie Raymonetika Manggala
5	
6	
Asisten pembimbing	NIM / Nama
	13518137/Michel Fang

Catatan Asistensi:

Rangkuman Diskusi
<p>Untuk F01, jika username sudah ada sebelumnya, maka tidak akan diterima lagi. Untuk lowercase dan uppercase nya terserah saja. Bisa saat disimpan di compress menjadi lowercase semua atau tidak, bisa juga diasumsikan dari awal.</p> <p>Untuk F09, jika entry dihapus, artinya datanya di kita sudah tidak ada tapi di orang lain masih ada. Table history adalah tempat menyimpan id nya, jadi saat mengembalikan, yang sudah dihapus dari database dapat diganti dengan id yang terdapat di table history. Yang artinya dimunculkan kembali.</p> <p>Untuk F14, saat load program python variabelnya kosong. Salah satu caranya adalah kita mencari folder csv nya dimana. Bisa dengan input dari program pythonnya dulu, foldernya dimana, kemudian kita dapat load file csv nya dengan suatu cara. Python spasi nama file spasi nama folder adalah cara singkatnya, tidak perlu input karena sudah terinput saat menjalankan program. Maka, Argument parsing atau argparse, akan men extract informasi yang kita berikan saat menjalankan program. Karena csv nya banyak, kita tidak mungkin memberikannya satu per satu, hanya perlu memberikan lokasi folder, dan semua csv harus sudah ada di file tersebut.</p> <p>Untuk F15, jika folder yang disimpan csv nya sudah ada sebelumnya, akan ditimpa. Namun jika belum ada, akan jadi csv baru.</p> <p>Untuk permintaan input kepada user, penggunaan tanda ">>>" adalah untuk menentukan itu input. Jadi bebas ingin menggunakannya atau tidak. Kemudian aturan kapitalisasinya dibebaskan. Bisa asumsikan harus kapital, bisa juga divalidasi dan diubah dulu ke lowercase.</p> <p>Tidak ada spesifikasi untuk keharusan validasi keberadaan bagi item yang ingin dipinjam atau diubah. Tetapi lebih baik dilakukan validasi karena jika terjadi kesalahan, seperti kesalahan penulisan, saat dicari akan terjadi out of index. Jangan terlalu fokus divalidasi, fokus ke fungsionalitasnya terlebih dahulu, karena saat testing akan terlihat jika ada yang kurang validasi.</p>
Boleh menggunakan append, tidak boleh menggunakan split.
Terkait laporan poin L, spesifikasi setiap fungsi di tulis dalam notasi algoritmik langsung ditulis di laporan. Namun, jika ingin ditambahkan di python sebagai comment juga boleh, di copy paste dan dirubah sedikit.
Tindak Lanjut
Penyelesaian program Pembagian tugas

Asistensi 2

Form MoM Asistensi Tugas Besar IF1210/Dasar Pemrograman Sem. 2 2020/2021

Nomor Asistensi : 2
No. Kelompok/Kelas : 11/Kog
Tanggal asistensi : 28 April 2021

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
1	16520129/Farisa Aliya
2	16520149/Gerald Abraham Sianturi
3	16520299/Malik Akbar Hashemi Rafsanjani
4	16520329/Hughie Raymonelika Manggala
5	
6	
Asisten pembimbing	NIM / Nama
	13518137/Michel Fang

Catatan Asistensi:

Rangkuman Diskusi
Untuk F12, F13, dan F14 fungsinya hampir sama, namun lebih baik fungsinya dipisah saja. Karena jika dijadikan 1 fungsi yang general akan banyak if else nya sehingga kurang rapih.
Untuk F14, bagian nama folder yang argparse itu nama folder csv bukan nama file python.
Untuk F16, jika pada aksesnya tidak tercantum admin dan user, berarti semua bisa mengaksesnya.
Untuk folder penyimpanan, bisa diberikan asumsi satu parent directory (dibawah folder yang sama dengan codenya). Tetapi dengan argparse juga bisa pindah folder.
Untuk notasi algoritmik cukup definisi fungsi yang diajarkan di kelas saja.
Untuk bagian spesifikasi di laporan, realisasi perlu dibuat juga. Namun tidak usah semua fungsi, fungsi-fungsi utama saja yang sekiranya perlu dijelaskan.
Tabel baru hanya diperbolehkan untuk soal bonus.
Link video youtube dan github dicantumkan di laporan dahulu, jika ada link pengumpulan yang lebih spesifik, bisa dikumpulkan lagi.
Tindak Lanjut
Menyelesaikan program Membuat video Menyusun laporan

Link Video Demo :

<https://youtu.be/swAJODnvOLk>

Link Github :

<https://github.com/malikrafsan/TubesDaspro>

Referensi :

<https://www.kite.com/python/answers/how-to-validate-a-date-string-format-in-python>

<https://www.geeksforgeeks.org/string-hashing-using-polynomial-rolling-hash-function/>

https://en.wikipedia.org/wiki/Rolling_hash

<https://13518114.medium.com/tubes-walkthrough-1-read-data-dari-csv-tanpa-library-605a6afe92db>