

TUGAS BESAR
SISTEM INVENTARISASI GADGET

Disusun untuk memenuhi tugas mata kuliah IF1201 Dasar Pemrograman.

Dosen: Prof. Ir. Dwi Hendratmo Widyantoro, M.Sc., Ph.D.

Fariska Zakhralativa Ruskanda, S.T., M.T

Asisten pembimbing: 13518114 Mario Gunawan

Oleh

16520193 Zhillan Attarizal Rezyarifin

16520203 Steven

16520353 Tubagus Baraka Kautsar S.

16520413 Jonathan



SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2021

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.”

Anggota kelompok,

1. 16520193 Zhillan Attarizal Rezyarifin
2. 16520203 Steven
3. 16520353 Tubagus Baraka Kautsar S.
4. 16520413 Jonathan

DAFTAR ISI

PERNYATAAN KELOMPOK	i
DAFTAR ISI.....	ii
DAFTAR TABEL.....	iii
DAFTAR GAMBAR	iv
BAB 1 Daftar Persoalan.....	1
BAB 2 Pembagian Kerja.....	2
BAB 3 Desain Command.....	7
BAB 4 Desain Kamus Data, Dekomposisi, Spesifikasi, Pengujian Per Fungsi	12
LAMPIRAN.....	63

DAFTAR TABEL

Tabel 2.1 - Tabel Pembagian Kerja	2
Tabel 2.2 - Tabel Checklist Hasil Rancangan, Implementasi dan Testing Setiap Primitif.....	5

DAFTAR GAMBAR

Gambar 4.1 – Gambar testing register	14
Gambar 4.2 – Gambar testing login.....	15
Gambar 4.3 – Gambar testing cari rarity	16
Gambar 4.4 – Gambar testing cari tahun	18
Gambar 4.5 – Gambar testing tambahitem	23
Gambar 4.6.1 – Hapus item apabila dipanggil oleh non-Admin	25
Gambar 4.6.2 – Hapus item apabila ID salah atau ID tidak ada dalam database	25
Gambar 4.6.3 – Hapus item consumable apabila masukan ‘N’ atau ‘Y’ pada konfirmasi	25
Gambar 4.6.4 – Hapus item gadget apabila diberimasukan ‘N’ atau ‘Y’ pada konfirmasi	26
Gambar 4.7 – Gambar testing ubahjumlah	29
Gambar 4.8 – Gambar pinjam saat inputan salah dan benar	33
Gambar 4.9 – Gambar testing riwayatpinjam	40
Gambar 4.10 – Gambar fungsi minta apabila ID salah atau jumlah salah	43
Gambar 4.11 – Gambar testing riwayatpinjam	45
Gambar 4.12 – Gambar riwayat pengembalian gadget.....	48
Gambar 4.13 – Gambar riwayatambil ketika barang yang diambil dihapus dari csv	53
Gambar 4.14.1 – Gambar apabila diload folder yang tidak terdapat pada database.....	54
Gambar 4.14.2 – Gambar apabila diload folder yang terdapat pada database.....	55
Gambar 4.15 – Gambar pengujian save	57
Gambar 4.16 – Gambar output help	59
Gambar 4.17.1 – Gambar exit ketika n	60
Gambar 4.17.2 – Gambar exit ketika y	60
Gambar Form Asistensi Pertama	63
Gambar Form Asistensi kedua.....	64

BAB 1

DESKRIPSI PERSOALAN

Program ini dibuat untuk membantu Doremonangis dan Mobita dalam menyelesaikan masalah mereka yaitu masalah migrasi sistem kantong ajaib Doremonangis ke dalam komputer Mobita. Sistem yang dibuat ini nantinya akan membantu untuk membuat Mobita lulus tepat waktu sebagai pelajar di ITB.

Dalam program ini kami diminta untuk menyelesaikan 17 fungsi wajib dan 3 fungsi tambahan. Fungsi tambahan atau fungsi bonus ini tidak wajib, hanya untuk membantu penggunaan Mobita agar lebih nyaman saja. Dalam pengerjaan fungsi-fungsi wajib dan fungsi tambahan, ada request dari Mobita, yakni tidak menggunakan `.split()` karena laptop Mobita tidak memiliki itu sehingga kami yang ingin menggunakan `.split()` harus membuatnya sendiri lalu, request dari Doremonangis adalah hanya boleh menggunakan Python 3.8+, `import os, sys, math, time, argparse`, dan `time`.

BAB 2 PEMBAGIAN KERJA

Tabel 2.1 - Tabel Pembagian Kerja

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F01 – Register	Procedure register	16520193	16520193	16520193 16520203 16520353 16520413
F02 – Login	procedure login	16520203	16520203	16520193 16520203 16520353 16520413
F03 – Pencarian gadget berdasar <i>rarity</i>	function carirarity	16520353	16520353	16520193 16520203 16520353 16520413
F04 – Pencarian gadget berdasar tahun ditemukan	function mencari item berdasarkan tahunnya	16520413	16520413	16520193 16520203 16520353 16520413
F05 – Menambah item	Procedure tambahitem	16520193	16520193	16520193 16520203 16520353 16520413

F06 - Menghapus gadget atau consumable	function hapusitem	16520203	16520203	16520193 16520203 16520353 16520413
F07 – Mengubah jumlah gadget atau consumable pada inventory	Function ubahjumlah	16520353	16520353	16520193 16520203 16520353 16520413
F08 – Meminjam gadget	function meminjam gadget	16520413	16520413	16520193 16520203 16520353 16520413
F09 – Mengembalikan gadget	Procedure kembalikan	16520193	16520193	16520193 16520203 16520353 16520413
F10 – Meminta consumable	procedure minta	16520203	16520203	16520193 16520203 16520353 16520413
F11 – Melihat riwayat peminjaman gadget	function riwayatpinjam	16520353	16520353	16520193 16520203 16520353 16520413

F12 – Melihat riwayat pengembalian gadget	function melihat riwayat pengembalian gadget	16520413	16520413	16520193 16520203 16520353 16520413
F13 – Melihat riwayat pengembalian consumable	Procedure riwayatambil	16520193	16520193	16520193 16520203 16520353 16520413
F14 – Load data	function load	16520203	16520203	16520193 16520203 16520353 16520413
F15 – Save data	function save	16520353	16520353	16520193 16520203 16520353 16520413
F16 – Help	function help	16520413	16520413	16520193 16520203 16520353 16520413
F17 - Exit	Procedure exit	16520193	16520193	16520193 16520203 16520353 16520413

FB01 - Hashing	function encrypt	16520203	16520203	16520193 16520203 16520353 16520413
FB02 – Mengembalikan gadget secara parsial	Fungsi mengembalikan gadget secara parsial	16520193	16520193	16520193 16520203 16520353 16520413
FB03 – Meningkatkan rarity consumables	-	-	-	-

Checklist Hasil Rancangan, Implementasi dan Testing Setiap Primitif

Tabel 2.2 - Tabel Checklist Hasil Rancangan, Implementasi dan Testing Setiap Primitif

Fitur	Desain	Implementasi	Testing
F01 – Register	V	V	V
F02 – Login	V	V	V
F03 – Pencarian gadget berdasar <i>rarity</i>	V	V	V
F04 – Pencarian gadget berdasar tahun ditemukan	V	V	V
F05 – Menambah item	V	V	V
F06 - Menghapus gadget atau consumable	V	V	V
F07 – Mengubah jumlah gadget atau consumable pada inventory	V	V	V
F08 – Meminjam gadget	V	V	V

F09 – Mengembalikan gadget	V	V	V
F10 – Meminta consumable	V	V	V
F11 – Melihat riwayat peminjaman gadget	V	V	V
F12 – Melihat Riwayat pengembalian gadget	V	V	V
F13 – Melihat Riwayat pengembalian consumable	V	V	V
F14 – Load data	V	V	V
F15 – Save data	V	V	V
F16 – Help	V	V	V
F17 - Exit	V	V	V
FB01 - Hashing	V	V	V
FB02 – Mengembalikan gadget secara parsial	V	V	V
FB03 – Meningkatkan rarity consumables	-	-	-

BAB 3

DESAIN COMMAND

>>> register

- Akses: Admin
- Menerima masukan berupa username terlogin dan array user.csv
- Apabila username terlogin bukan admin, menampilkan pesan blokir dan program selesai.
- Apabila username terlogin adalah admin, menerima username baru.
- Apabila username baru sudah ada di data, ditolak dan loop hingga username baru tidak ada di data.
- Menerima password baru yang kemudian di-encrypt dengan fungsi hash
- Menerima alamat
- Menyimpan data baru ke dalam array dan mengembalikan array itu

>>> login

- Akses: Admin dan User
- Menerima masukan berupa username
- Menerima masukan berupa password, dan kemudian password tersebut akan dihashing sebelum dicocokkan dengan password yang ada di database
- Apabila masukan yang diterima tidak terdapat pada database, maka akan dikeluarkan "Username dan/atau Password Anda Mungkin Salah!"
- Apabila masukan yang diterima terdapat pada database, akan dikeluarkan "Halo username! Selamat datang di Kantong Ajaib."

>>> carirarity

- Akses: Admin dan User
- Menerima masukan berupa jenis rarity gadget yang ingin dicari
- Apabila masukan valid maka akan menampilkan nama, deskripsi, jumlah, rarity, dan tahun ditemukan semua gadget yang memiliki rarity sesuai
- Apabila masukan tidak valid atau tidak ditemukan gadget dengan rarity tersebut, maka akan dikeluarkan "Tidak ditemukan gadget dengan rarity rarity."

>>> caritahun

- Akses: Admin dan User
- Menerima masukan berupa tahun ditemukan dan kategorinya.
- Apabila masukan valid akan menampilkan semua nama, deskripsi, jumlah, rarity, dan tahun ditemukan yang berada dalam jangkauan tahunnya
- Apabila masukan tidak valid akan diminta menginput ulang

- Apabila tidak ada yang ditemukan gadget yang berada di daerah tahun yang dicari akan dikeluarkan “Gadget tidak ditemukan”

>>> tambahitem

- Akses: Admin
- Menerima array dari gadget.csv, consumable.csv, user.csv; menerima username terlogin
- Mengecek jika username terlogin adalah admin. Jika bukan, output tuple “null” dan print notifikasi penolakan
- Jika admin, input id
- Jika id dimulai dari “G” dan belum ada, fungsi penambahan gadget dan return hasilnya. Jika “C” dan belum ada, fungsi penambahan consumable dan return hasilnya. Selain itu, gagalkan dan return tuple null.
- Untuk fungsi gadget, menerima id yang telah dimasukkan, nama gadget, deskripsi, rarity (yang akan loop sampai tervalidasi), jumlah (yang akan loop sampai tervalidasi), tahun (yang akan loop sampai tervalidasi).
- Untuk fungsi gadget, menerima id yang telah dimasukkan, nama gadget, deskripsi, rarity (yang akan loop sampai tervalidasi), jumlah (yang akan loop sampai tervalidasi).

>>> hapusitem

- Akses: Admin
- Menerima masukan berupa ID item
- Apabila ID item tidak terdapat pada database, maka akan dikeluarkan “Tidak ada item dengan ID tersebut.”
- Apabila ID item terdapat pada database, maka akan keluar pesan konfirmasi “Apakah anda yakin ingin menghapus item (Y/N)?”
- Apabila konfirmasi diisi Y, maka akan dikeluarkan “Item telah berhasil dihapus dari database.”
- Apabila konfirmasi diisi N, maka akan dikeluarkan “Item tidak jadi dihapus dari database.”
- Apabila konfirmasi diisi dengan char selain ‘Y’ dan ‘N’, maka akan dikeluarkan “Input tidak valid”

>>> ubahjumlah

- Akses: Admin
- Menerima masukan berupa ID item
- Apabila ID item tidak ditemukan pada database, maka akan dikeluarkan “Tidak ada item dengan ID tersebut.”

- Apabila ID ditemukan, fungsi akan menerima masukan berupa integer jumlah yang ingin ditambah atau dikurang dari jumlah yang sudah ada.
- Apabila jumlah akhir valid (positif) maka akan dikeluarkan “nama item berhasil ditambahkan/dibuang. Stok sekarang:...”
- Apabila jumlah akhir tidak valid (negatif) maka akan dikeluarkan “nama item gagal dibuang. Stok sekarang:...”

>>> pinjam

- Akses: User
- Menerima masukan berupa ID item, tanggal peminjaman dan jumlah peminjaman
- Apabila semua masukan valid, maka akan dikeluarkan “Item (item yang dipinjam + jumlah yang dipinjam) berhasil dipinjam.”
- Apabila ID item tidak valid atau tidak ditemukan, maka akan dikeluarkan “ID tidak valid.”. Apabila item sedang dipinjam, maka akan dikeluarkan “Gadget sedang dipinjam, silahkan pinjam gadget yang lain.”
- Apabila tanggal tidak valid, maka akan dikeluarkan “Tanggal tidak valid.”
- Apabila jumlah peminjaman melebihi jumlah gadget yang tersedia, maka akan dikeluarkan “Jumlah tidak mencukupi.”

>>> kembalikan

- Akses: User
- Menerima masukan berupa nomor peminjaman dan tanggal pengembalian
- Tanggal pengembalian tidak perlu lebih besar dari tanggal peminjaman
- Entry peminjaman harus ada untuk mengembalikan gadget

>>> minta

- Akses: User
- Menerima masukan berupa ID item
- Menerima masukan berupa jumlah item yang diinginkan
- Menerima tanggal permintaan dari device pengguna
- Apabila ID item tidak terdapat pada database, maka akan dikeluarkan “Tidak ada item dengan ID tersebut.”
- Apabila masukan jumlah item yang diinginkan melebihi jumlah item pada database, akan dikeluarkan “Item tidak mencukupi permintaan”
- Syarat jumlah yang diminta adalah jumlah harus lebih dari nol.
- Apabila semua input telah benar, maka akan dikeluarkan “Item itemName (x itemQuantity) telah berhasil diambil!”

>>> riwayatpinjam

- Akses: Admin
- Mengeluarkan riwayat peminjaman gadget sebanyak 5 buah dari peminjaman terbaru hingga terlama.
- Apabila masih ada entry, pengguna dapat memilih untuk mengeluarkan 5 entry selanjutnya.

>>> riwayatkembali

- Akses: Admin
- Mengeluarkan riwayat pengembalian gadget sebanyak 5 buah dari pengembalian terbaru dan tersusun menurun menurut tanggalnya
- Apabila masih ada riwayat yang belum dikeluarkan, bisa mengeluarkan tambahan riwayat sebanyak 5 buah
- Apabila riwayat sudah ditampilkan semua, akan dikeluarkan “Riwayat sudah ditampilkan semua.”

>>> riwayatambil

- Akses: Admin
- Mengecek apakah user terlogin adalah admin. Jika bukan, hentikan program dan print notifikasi.
- Mengecek apakah riwayat ada. Jika kasus kosong, print notifikasi dan selesai. Jika tidak, sort berdasar tanggal dan lanjut.
- Mengecek apakah jumlah riwayat < 5. Jika ya, display semuanya secara berurutan.
- Jika > 5, tampilkan 5 terbaru, tampilkan pesan apakah ingin melihat lebih. Jika ya, loop langkah di atas dan (jika masih > 5) langkah ini.
- Jika habis, tampilkan notifikasi dan selesai.

>>> load

- Dijalankan bersamaan ketika file utama dijalankan
- Format penulisannya pada terminal adalah: “python kantongajaib.py nama_folder_database_yang_ingin_diakses”

>>> save

- Akses: Admin dan User
- Menerima nama folder yang ingin digunakan untuk penyimpanan file.
- Apabila folder sudah ada maka program akan mengecek isi folder dan menggantikan file lama dengan file yang baru.
- Apabila folder belum ada maka program akan membuat folder baru dan

mengisinya dengan file yang sesuai

- Apabila berhasil akan dikeluarkan “Data berhasil disimpan di folder nama folder!”

>>> help

- Akses: Admin dan User
- Mengeluarkan panduan penggunaan sistem

>>> exit

- Akses: -
- Menerima masukan yang bertujuan untuk menyimpan file atau tidak

BAB 4

DESAIN KAMUS DATA, DEKOMPOSISI, SPESIFIKASI, PENGUJIAN PER FUNGSI

F01

PROGRAM REGISTER

```
{ Spesifikasi: Mendaftarkan data user baru dengan username
unik. Khusus admin. }
{ Asumsi : semua array, kecuali diatur, telah memiliki banyak
elemen yang benar, fungsi bawaan Python sudah ada di notal }
```

KAMUS

```
{ Deklarasi variabel }
```

```
    type userline :
        < id : integer;
          username : string;
          nama : string;
          alamat : string;
          password : string;
          role : string >
```

```
{ Deklarasi fungsi dan prosedur }
```

```
function encrypt (input password : string) -> integer
{ Spek : hasing password, dengan prosedur dijelaskan di FB01 }
{ I.S. password }
{ F.S. deretan integer hasil hash }
```

```
function isAdmin(logged_user : string, user_data : array of
userline) -> boolean
{ SPEK : Menentukan apakah user terlogin admin atau bukan,
sebagaimana f99 }
{ I.S. username terlogin }
{ F.S. sebuah boolean }
```

```
procedure register (input user_data : array of userline, input
logged_user : string, output return : integer atau array of
userline)
{ Spek : sesuai spek modul }
{ I.S. user terlogin, array data user.csv, inputan user }
{ F.S. data user.csv baru atau string "notAdmin", berbagai
print out }
```

KAMUS LOKAL

```
    userline : userline
```

ALGORITMA

```

    if isAdmin(logged_user, user_data) then
        output("Masukkan nama: ")
        input(userline.nama)
        output("Masukkan username: ")
        Input(userline.username)

        while usernameNotUnique(userline.username, user_data)
do
            output("Username sudah dipakai. Mohon masukkan
username unik.")
            input(userline.username)
            { username is unique }

            encrypt(output("Masukkan password: "))
            input(userline.usersame)
            output("Masukkan alamat: ")
            input(userline[3])
            Userline.id <- (user_data[len(user_data) - 1][0]) + 1
            output("User " + userline.username + "berhasil resister
sementara ke dalam Kantong Ajaib.")
            user_data.append(userline)
            return <- user_data {F.S.}
        else
            Output("Ditolak! Fungsi terbatas admin.")
            return <- "notAdmin" {F.S.}

function usernameNotUnique (input username : string, input
user_data : array of userline) -> boolean
{ Mengecek apakah username sudah ada di user_data }
{ I.S. username yang dimasukkan }
{ F.S. boolean sudah ada atau tidak ada }
KAMUS LOKAL
    notUnique : boolean
    i : integer
ALGORITMA
    notUnique <- false
    i transversal [1..(len(user_data) - 1)]
        if username = user_data[i][1] then
            NotUnique <- true
    -> notUnique

```

Pengujian:

Gambar 4.1 – Gambar testing register

```
>>> register
Masukkan nama: darkness
Masukkan username: darkness
Username sudah dipakai. Mohon masukkan username unik.
Masukkan username: namabaru
Masukkan password: test
Masukkan alamat: inialamat
User namabaru berhasil register sementara ke dalam Kantong Ajaib.
```

F02

KAMUS

```
file : array of array of string
terdaftar : boolean
username : string
password : string
pesan : string
```

procedure login (input file: array of array of string, input username: string, input password: string)
 {Menerima suatu array of array of string (array of user) bernama file, lalu mengoutputkan suatu variabel misalkan "pesan" yang bertipe string sesuai dengan data yang terolah}

function encrypt (key: string) -> string
 {menerima input bertipe string misalnya sandi, lalu mengoutput sandi yang telah dienkripsi}

KAMUS LOKAL

```
row : integer {indeks yang menyatakan baris}
```

Algoritma

```
terdaftar <- false

while terdaftar = false do
  password = encrypt(password)

  row travelsal file
    if (username = row[1]) and (password = row[4]) then
      pesan <- ("Halo " + username + "! " + "Selamat
datang di Kantong Ajaib."
      output(pesan)
      terdaftar <- true
    if (terdaftar = false) then
      pesan <- ("Username dan/atau Password Anda
Mungkin Salah!")
      output(pesan)
```

Pengujian:

Gambar 4.2 – Gambar testing login

```
PROGRAM KANTONG AJAIB IF1201 K-03 Kel. 6.  
2021. All rights not reserved.  
  
Masukan username: steven  
Masukan password: password salah  
Username dan/atau Password Anda Mungkin Salah!  
  
Masukan username: username salah  
Masukan password: 16520203  
Username dan/atau Password Anda Mungkin Salah!  
  
Masukan username: steven  
Masukan password: 16520203  
Halo steven! Selamat datang di Kantong Ajaib.  
  
Ada yang bisa dibantu, steven?  
>>>
```

F03

PROGRAM PENCARIAN BERDASARKAN RARITY

{ Spesifikasi : menampilkan semua gadget dengan rarity yang dimasukkan pengguna }

KAMUS

```
{ Deklarasi variabel }  
  type gadget :  
    < id : integer;  
      nama : string;  
      desc : string;  
      jumlah : integer;  
      rarity : string;  
      tahun : string >  
{ Deklarasi variabel }
```

Procedure carirarity (input gadget_data: array of gadget)

{ Spesifikasi : menampilkan semua gadget dengan rarity yang dimasukkan pengguna }

{ I.S. Array dari gadget.csv }

{ F.S. Print out informasi gadget dengan rarity sesuai input }

KAMUS LOKAL

```
  rarity : str  
  found : bool
```

ALGORITMA

```
  Output ("Masukkan rarity: ")  
  Input ( rarity)  
  Found <- False {inisiasi}  
  I traversal [0..length(gadget data)]
```

```

    If gadget_data[i][4]=rarity then
        Output ("Nama      : "+gadget_data[i][1])
        Output ("Deskripsi : "+gadget_data[i][2])
        Output ("Jumlah    : "+gadget_data[i][3])
        Output ("Rarity    : "+gadget_data[i][4])
        Output ("Tahun ditemukan: "+gadget_data[i][5])
        Output ()
        found <- True
    if found = False then
        output ("Tidak ditemukan gadget dengan rarity " +
rarity)

```

Pengujian:

Gambar 4.3 – Gambar testing cari rarity

```

>>> carirarity
Masukkan rarity: A

Hasil Pencarian:

Nama      : Art Code Generator
Deskripsi : Menghasilkan deretan 6 angka dengan cara men-scan cultured art apapun
Jumlah    : 416
Rarity    : A
Tahun ditemukan : 2072

Nama      : Spoiler Sinker
Deskripsi : Menenggelamkan pembaca manga tukang spoiler dalam pasir hisap hitam
Jumlah    : 415
Rarity    : A
Tahun ditemukan : 2073

```

F04

PROGRAM PENCARIAN GADGET BERDASARKAN TAHUN DITEMUKAN

```

{ Spesifikasi : Untuk admin dan user, mencari gadget dari
inventori berdasarkan tahun ditemukannya:
(1) tahun berupa 'yyyy'
(2) kategori pencarian berupa {=,<,>,>=,<=}
    • = artinya gadget yang terbit pada tahun yyyy;
    • < artinya gadget yang terbit setelah tahun yyyy;
    • > artinya gadget yang terbit sebelum tahun yyyy;
    • >= artinya gadget yang terbit pada atau sebelum tahun
      yyyy;
    • <= artinya gadget yang terbit pada atau setelah tahun
      yyyy.)
{ Asumsi : format tanggal dan kategori yang diinput pasti valid
}
{ I.S. : menerima 2 buah input berupa tahun dan kategori, dan
array dari gadget.csv}
{ F.S. : mengeluarkan semua nama, deskripsi, jumlah, rarity,
dan tahun ditemukan gadget yang berada dalam jangkauan kategori
tahunnya }

```

procedure cari_gadget_tahun

Kamus lokal

Arr : string

thn : string

kat : string

Kategori() : procedure

Algoritma

arr = x

Output ('Masukkan tahun: ')

Input (thn)

Output ('Masukkan kategori: ')

Input (kat)

Kategori(arr, thn, kat)

Procedure kategori (input arr : array of gadget, input kat : string, input thn : string)

{ menyaring gadget yang berada di dalam jangkauan kategori tahun }

Kamus lokal

A : integer

N : integer

Output() : procedure

Algoritma

A <- 0

N <- [0... length(arr)]

Output('')

Output('Hasil pencarian:')

I traversal [arr]

If kat = '=' then

If arr[I][5] = thn then

Output(arr,I)

I = I + 1

A = A + 1

Else

I = I + 1

If kat = '>' then

If arr[I][5] > thn then

Output(arr,I)

I = I + 1

```

        A = A + 1
    Else
        I = I + 1
    If kat = '<' then
        If arr[I][5] < thn then
            Output(arr,I)
            I = I + 1
            A = A + 1
        Else
            I = I + 1
    If kat = '>=' then
        If arr[I][5] >= thn then
            Output(arr,I)
            I = I + 1
            A = A + 1
        Else
            I = I + 1
    If kat = '<=' then
        If arr[I][5] <= thn then
            Output(arr,I)
            I = I + 1
            A = A + 1
        Else
            I = I + 1
    If A <= 0 then
        Output('Gadget tidak ditemukan.')

```

Procedure Output (input x : array, input I : integer)
 {mengeluarkan output berupa nama, deskripsi, jumlah, rarity, tahun ditemukan}

Algoritma

```

    Output('')
    Output('Nama:', x[I][1])
    Output('Deskripsi:', x[I][2])
    Output('Jumlah:', x[I][3], 'buah')
    Output('Rarity:', x[I][4])
    Output('Tahun Ditemukan', x[I][5])

```

Pengujian:

Gambar 4.4 – Gambar testing cari tahun

```

Ada yang bisa dibantu, steven?
>>> caritahun
Masukkan tahun: 1990
Masukkan kategori: <

Hasil pencarian:
Gadget tidak ditemukan.

Ada yang bisa dibantu, steven?
>>> caritahun
Masukkan tahun: 2069
Masukkan kategori: =

Hasil pencarian:

Nama: Auto Absen
Deskripsi: Robot yang memantengi laptop dan mengabsenkan dalam format apapun
Jumlah: 420 buah
Rarity: B
Tahun Ditemukan: 2069

```

F05

PROGRAM MENAMBAH ITEM

```

{ Spesifikasi : Hanya untuk admin, menambahkan ke inventory:
(1) gadget, dengan awalan id "G",
(2) consumable, dengan awalan id "C" }
{ Asumsi : semua array, kecuali diatur, telah memiliki banyak
elemen yang benar, fungsi bawaan Python sudah ada di notal }
{ I.S. Array dari gadget.csv, consumable.csv, user.csv,
username terlogin }
{ F.S. Array gadget.csv, consummable.csv yang baru }

```

KAMUS

```

{ Deklarasi variabel }
    type gadget :
        < id : integer;
          nama : string;
          desc : string;
          jumlah : integer;
          rarity : string;
          tahun : string >
    type consumable :
        < id : integer;
          nama : string;
          desc : string;
          jumlah : integer;
          rarity : string >
    type userline :

```



```

        < id : integer;
        username : string;
        nama : string;
        alamat : string;
        password : string;
        role : string >

{ Deklarasi fungsi dan prosedur }

function isAdmin(logged_user : string, user_data : array of
userline) -> boolean
{ SPEK : Menentukan apakah user terlogin admin atau bukan,
sebagaimana f99 }
{ I.S. username terlogin }
{ F.S. sebuah boolean }

function charAreInt(text : string) -> boolean
{ SPEK : Mengecek apakah inputan string bisa jadi integer,
fungsi dari f99 }
{ I.S. text yang akan diperiksa }
{ F.S. boolean }

procedure tambahitem (
    input gadget_data : array of gadget;
    input consumable_data : array of consumable;
    input user_dataa : array of userline;
    input logged_user : string,
    output return : (return_arr : array of gadget or "null";
return_arr : array of gadget or "null" )
{ Spesifikasi : Hanya untuk admin, menambahkan ke inventory:
(1) gadget, dengan awalan id "G",
(2) consumable, dengan awalan id "C" }
{ I.S. Array dari gadget.csv, consumable.csv, user.csv,
username terlogin }
{ F.S. Array gadget.csv, consumable.csv yang baru, berbagai
print out }
KAMUS LOKAL
    id, id_status : string
ALGORITMA
    if isAdmin(logged_user, user_data) then
        output("Masukkan ID: ")
        input(id)
        id_status <- chechId(id, gedget_dataa, consumable_data)
        if id_status = idIsGadget then
            return_arr <- tambahGadget(id, gadget_data)
            return <- (return_arr, "null")
        else if id_status = "idIsConsum" then

```

```

        return_arr = tambahConsum(is, consumable_data)
        return <- return_arr
    else if id_status = "idAlreadyExist then
        output("Gagal menambahkan item karena ID sudah ada)
        return <- ("null", "null")
    else
        output("Gagal menambahkan item karena ID tidak
valid)
        return <- ("null", "null")
    else
        output("Halt! Fungsi terbatas admin.")
        return <- ("null", "null")

```

function checkId(id : array of character, gadget_data : array of gadget, consumable_data : array of consumable) -> string
{ SPEK: Mengecek kevalidan ID }
{ I.S. Id inputan, array gadget dan consumable }
{ F.S. string status kevalidan }

KAMUS LOKAL

```

    return_str : string
    row : gadget

```

ALGORITMA

```

    return_str <- idNotValid
    if id[0] = "G" then
        return_str <- "idIsGadget"
        row transversal gadget_data
        if row[0] = id then
            return_str <- "idAlreadyExist"
    else if id[0] = "C" then
        return_str <- "idIsConsum"
        row transversal consumable_data
        if row[0] = id then
            return_str <- "idAlreadyExist"
    -> return_str

```

function tambahGadget(id : string, gadget_data : array of gadget) -> array of gadget
{ SPEK : Menginput data, mengecek kevalidan, menyimpan }
{ I.S. id, array gadget, inputan user }
{ F.S. array gadget yang baru }

KAMUS LOKAL

```

    gadget : gadget

```

ALGORITMA

```

    gadget.id <- id
    output("Masukkan Nama Gadget :")
    input(gadget.nama)

```

```

output("Masukkkkan Deskripsi :")
input(gadget.desc)

output("Masukkkkan Jumlah :")
input(gadget.jumlah)
while not charAreInt(gadget.jumlah]) do
  output("Input jumlah tidak valid!")
  output("Masukkkkan Jumlah :")
  input(gadget.jumlah)
{ jumlah adalah integer }

output("Masukkkkan Rarity :")
input(gadget.rarity)
while not rarityValid(gadget.rarity]) do
  output("Input rarity tidak valid!")
  output("Masukkkkan Jumlah :")
  input(gadget.rarity)
{ rarity valid }

output("Masukkkkan Tahun Ditemukan:")
input(gadget.tahun)
while not isTahun(gadget.tahun]) do
  output("Input tahun tidak valid!")
  output("Masukkkkan Tahun Ditemukan:")
  input(gadget.tahun)
{ jumlah adalah integer }

output(gadget.nama + "(x" + gadget.jumlah + ") ditambahkan
ke database sementara")
gadget_data.append(gadget)
-> gadget_data

```

```

function tambahGadget(id : string, consumable_data : array of
gadget) -> array of consumable
{ SPEK : Menginput data, mengecek kevalidan, menyimpan }
{ I.S. id, array cosumaable, inputan user }
{ F.S. array consumable yang baru }

```

KAMUS LOKAL

```
Consumable : consumable
```

ALGORITMA

```

Consumable.id <- id
output("Masukkkkan Nama Consumable: ")
input(consumable.nama)

output("Masukkkkan Deskripsi: ")

```

```

input(consumable.desc)

output("Masukkkkan Jumlah: ")
input(consumable.)
while not charAreInt(consumable.jumlah]) do
    output("Input jumlah tidak valid!")
    output("Masukkkkan Jumlah: ")
    input(consumable.jumlah)
{ jumlah adalah integer }

output("Masukkkkan Rarity :")
input(consumable.rarity)
while not rarityValid(consumable.rarity]) do
    output("Input rarity tidak valid!")
    output("Masukkkkan Jumlah :")
    input(consumable.rarity)
{ rarity valid }

output(consumable.nama + "(x" + consumable.jumlah + ")
ditambahkan ke database sementara")
consumable _data.append(gadget)
-> consumable _data

function rarityValid(rarity : string) -> boolean
{ SPEK : Mengecek apakah rarity valid }
{ I.S. inputan rarity }
{ F.S. boolean }
ALGORITMA
    if rarity = "C" or text = "B" or text = "A" or text = "S"
do
    -> true
else
    -> false

```

Pengujian:

Gambar 4.5 – Gambar testing tambahitem

```

Perintah aku, master steven~!
>>> tambahitem
Masukkan ID: C1
Gagal menambahkan item karena ID sudah ada.

Perintah aku, master steven~!
>>> tambahitem
Masukkan ID: C20
Masukkan Nama Consumable: permen kelar tubes
Masukkan Deskripsi: permen untuk menyelesaikan tubes secara instan
Masukkan Jumlah: 20
Masukkan Rarity: 5
permen kelar tubes(x20) ditambahkan ke database sementara.

```

F06

KAMUS

ada_id_item: boolean
keputusan: character
array_yang_digunakan: array of array of string

procedure hapusitem (input id_item: string, input array_of_gadget: array of array of string, input array_of_consumable: array of array of string)

{Initial State:

Menerima suatu input id_item dan menerima 2 buah array, yaitu array_of_gadget dan array_of_consumable yang dimana akan diproses sesuai dengan id_item yang telah diinput}

{Final State:

Menghapus data yang memiliki id_item yang diinputkan apabila id_item tersebut terdapat dalam database dan penghapusan telah disetujui oleh pengguna}

KAMUS LOKAL

i: integer {indeks looping}
j: integer {indeks looping}
row_matriks: integer {indeks looping}

ALGORITMA

ada_id_item <- false
keputusan <- ''

{Mengecek apakah id_item yang diinput ada dalam data base}

if id_item[0] = "G" then
 array_yang_digunakan <- array_of_gadget
 i traversal [0..length(array_yang_digunakan)]
 if id_item = array_yang_digunakan[i][0] then
 ada_id_item <- true
if id_item[0] = "C" then
 array_yang_digunakan <- array_of_consumable
 j traversal [0..length(array_yang_digunakan)]
 if id_item = array_yang_digunakan[j][0] then
 ada_id_item <- true
if ada_id_item = false then
 output("Tidak ada item dengan ID tersebut")

{Meminta konfirmasi dari pengguna}

if ada_id_item = true then
 row_matriks traversal [0..length(array_yang_digunakan)]

```

        if id_item = row_matriks[0] then
            keputusan <- input("Apakah anda yakin ingin
menghapus " + row_matriks[1] + " (Y/N)?")
            if keputusan = 'Y' then
                new_arr <- write_new_array {Memanggil fungsi untuk
membuat array baru hasil perubahan yang telah dilakukan}
                array_to_csv(new_arr) {Memanggil fungsi bantuan
untuk membuat array menjadi csv dengan delimiter ";" }
                output("Item telah berhasil dihapus dari
database.")
            if keputusan = 'N' then
                output("Item tidak jadi dihapus dari database.")
            if (keputusan <> 'Y') or (keputusan <> 'N') then
                output("Input tidak valid")

```

Pengujian:

Gambar 4.6.1 - Hapus item apabila dipanggil oleh non-Admin

```

Ada yang bisa dibantu, userbiasa?
>>> hapusitem
Anda Bukan Admin!
Akses hapusitem ditolak!

```

Gambar 4.6.2- Hapus item apabila ID salah atau ID tidak ada dalam database

```

>>> hapusitem
Masukan ID item: Z1
Tidak ada item dengan ID tersebut
Masukan ID item: C1000000
Tidak ada item dengan ID tersebut

```

Gambar 4.6.3 - Hapus item consumable apabila diberi masukan 'N' atau 'Y' pada konfirmasi

```

>>> hapusitem
Masukan ID item: C10
Apakah anda yakin ingin menghapus Obat Bahagia (Y/N)? N
Item tidak jadi dihapus dari database.
Masukan ID item: C10
Apakah anda yakin ingin menghapus Obat Bahagia (Y/N)? Y
Item telah berhasil dihapus dari database.

```

Gambar 4.6.4 - Hapus item gadget apabila diberimasukan 'N' atau 'Y' pada konfirmasi

```

>>> hapusitem
Masukan ID item: G8
Apakah anda yakin ingin menghapus Kalkulator Wolfram (Y/N)? N
Item tidak jadi dihapus dari database.
Masukan ID item: G8
Apakah anda yakin ingin menghapus Kalkulator Wolfram (Y/N)? Y
Item telah berhasil dihapus dari database.

```

F07

PROGRAM MENGUBAH JUMLAH GADGET ATAU CONSUMABLE PADA INVENTORY

{ Spesifikasi : Mengubah jumlah gadget atau consumable pada inventory sistem (akses: Admin)}

KAMUS

```

{ Deklarasi variabel }
    type gadget :
        < id : integer;
          nama : string;
          desc : string;
          jumlah : integer;
          rarity : string;
          tahun : string >
    type consumable :
        < id : integer;
          nama : string;
          desc : string;
          jumlah : integer;
          rarity : string >
    type status :
        < jenis : string;
          Idx : integer >
{ Deklarasi fungsi dan procedure }

```

Procedure ubahjumlah (

```

    input gadget_data : array of gadget;
    input consumable_data : array of consumable;
    input user_data : array of userline;
    input logged_user : string)
{ Spesifikasi : Menerima input id item dan melanjutkan proses sesuai jenis dan status validasi id)
{ I.S. Array dari gadget.csv, consumable.csv, user.csv; string user name logged user }
{ F.S. Array gadget.csv, consumable.csv yang baru }

```

KAMUS LOKAL

```

    id : str
    id_status : status

```

ALGORITMA

```

    if isAdmin(logged user, user data) then

```

```

        Output ("Masukkan id: ")
        Input (id)
        Id_status <- Checkid(id, gadget_data, consumable_data)
        If id_status[0]="Gadget" then
        Else if id_status[0]="Consumable" then
        Else
            Output ("Tidak ada item dengan id tersebut!")
    Else
        Output ("Halt! Fungsi terbatas admin.")

```

Function Checkid(id:string, gadget_data,consumable_data: arr of arr) -> status: arr
 {Spesifikasi: Memeriksa jenis dan kevalidan id, kemudian memeriksa keberadaan nya pada inventory dan memberikan integer berupa indeks}

KAMUS LOKAL

```

    Id: string
    Gadget_data, consumable_data : arr of arr of string
    Return_arr: status

```

ALGORITMA

```

    Return_arr <- ["Not Valid", 0] {inisiasi}
    If id[0]="G"
        i traversal [0..length(gadget_data)]
            if id[0]=gadget_data[i][0] then
                return_arr <- ["Gadget",i]
    else If id[0]="C"
        i traversal [0..length(consumable_data)]
            if id[0]=consumable_data[i][0] then
                return_arr <- ["Consumable",i]

    output(return_arr)

```

procedure ubahjumlahGadget (input id_status : status, gadget_data: array of gadget)
 {Spesifikasi: Menerima perubahan jumlah, mengecek kevalidan pengubahan, merubah variabel data bila valid}
 { I.S. Array dari gadget.csv }
 { F.S. Array gadget.csv yang baru}

KAMUS LOKAL

```

    Idx, ubah, jumlah_i,jumlah_f : integer
    Gadget_data : arr of gadget
    Id_status: status

```

ALGORITMA

```

    Output("Masukkan jumlah: ")
    Input(ubah)
    Idx <- id status[1]

```



```

Jumlah_i <- gadget_data[idx][3]{jumlah inisial}
Jumlah_f <- Jumlah_i + ubah {jumlah final}
If (jumlah_f > 0) then
    Gadget_data[idx][3]<-string(jumlah_f)
    If (ubah>0) then
        Output(ubah+gadget_data[idx][1]+"berhasil ditambah,
Stok Sekarang:"+Jumlah_f)
    Else if (ubah<0) then
        Output(ubah+gadget_data[idx][1]+"berhasil dibuang,
Stok Sekarang:"+Jumlah_f)
    Else
        Output(ubah+gadget_data[idx][1]+"gagal dibuang, Stok
Sekarang:"+Jumlah_i)

```

procedure ubahjumlahConsumable (input id_status : status,
consumable_data: array of consumable)
{Spesifikasi: Menerima perubahan jumlah, mengecek kevalidan
pengubahan, merubah variabel data bila valid}
{ I.S. Array dari consumable.csv}
{ F.S. Array consumable.csv yang baru}

KAMUS LOKAL

```

Idx, ubah, jumlah_i, jumlah_f : integer
consumable_data : arr of consumable
Id_status: status

```

ALGORITMA

```

Output("Masukkan jumlah: ")
Input(ubah)
Idx <- id_status[1]
Jumlah_i <- consumable_data[idx][3]{jumlah inisial}
Jumlah_f <- Jumlah_i + ubah {jumlah final}
If (jumlah_f > 0) then
    consumable_data[idx][3]<-string(jumlah_f)
    If (ubah>0) then
        Output(ubah+ consumable_data[idx][1]+"berhasil
ditambah, Stok Sekarang:"+Jumlah_f)
    Else if (ubah<0) then
        Output(ubah+ consumable_data[idx][1]+"berhasil
dibuang, Stok Sekarang:"+Jumlah_f)
    Else
        Output(ubah+ consumable_data[idx][1]+"gagal dibuang,
Stok Sekarang:"+Jumlah_i)

```

Pengujian:

Gambar 4.7 – Gambar testing ubahjumlah

```

Perintah aku, daddy raka~!
>>> ubahJumlah
Masukkan ID: G1
Masukkan Jumlah: 10
10 Auto Absen berhasil ditambahkan. Stok sekarang: 430

```

F08

PROGRAM PINJAM GADGET

(Spesifikasi : Hanya untuk user, meminjam gadget dari inventori:

1. peminjaman gadget akan menambahkan entry pada file history peminjaman gadget bila di-save
2. bila gadget menjadi 0, tidak perlu dihapus dari sistem
3. bila gadget sedang dipinjam, gadget tidak bisa dipinjam lagi.)

{ I.S. : array dari gadget.csv dan gadget_borrow_history.csv, menerima 3 buah input berupa ID, tanggal dan jumlah, serta pilihan untuk menyimpan pinjaman kedalam history atau tidak }
 { F.S. : mengeluarkan array gadget.csv dan history_borrow_history.csv yang baru }

Procedure pinjamgadget (input arr1 : array of gadget, input arr2 : array of gadget_borrow_history)

Kamus

ID : string
 Tgl : string
 Jml : string
 Cek_ID() : procedure
 Cek_Pinjam() : procedure
 Cek_Tgl() : procedure
 Cek_Jml() : procedure
 i : integer
 N : integer
 s : string
 arr_baru() : procedure

Algoritma

```

Output('Masukkan ID item: ')
Input(ID)
while True do
    if Cek_ID(arr1, ID) == True then
        if Cek_Pinjam(ID, arr2) == True then
            Output('Gadget sedang dipinjam, silahkan pinjam

```

```

gadget yang lain.')
    Output('Masukkan ID item: ')
    Input(ID)
    else
        break
    else
        Output('ID tidak valid.')
        Output('Masukkan ID item: ')
        Input(ID)
Output('Tanggal peminjaman: ')
Input(Tgl)
while True do
    if Cek_Tgl(arr1, Tgl) == True then
        break
    else
        output('Tanggal tidak valid.')
        output('Tanggal peminjaman: ')
        input(Tgl)
output('Jumlah peminjaman: ')
input(Jml)
while True do
    if Cek_Jml(arr1, ID, Jml) != True then
        output('Jumlah tidak mencukupi.')
        output('Jumlah peminjaman: ')
        input(Jml)
    else
        i <- 0
        N <- [0... length(arr1)]
        i traversal N
        if ID != arr1[i][0] then
            i = i+1
        elif ID == arr1[i][0] then
            output('Item ' + arr1[i][1] + ' (x' + Jml + ')
berhasil dipinjam!')
            break

    output('Apakah pinjaman gadget ingin di simpan?')
    input(s)
    if s == 'YA' then
        arr_baru(arr2, ID, Tgl, Jml)
        return arr2

```

procedure Cek_ID

```
{ mengecek apakah ID ada atau tidak di inventori }
```

Kamus lokal

i = integer

N = integer

arr = array of array

Algoritma

i <- 0

N <- [0... length(arr)

i traversal N

if ID != arr[i][0] then

 i = i+1

else if ID == arr[i][0] then

return True

procedure Cek_Tgl

```
{ mengecek input tanggal valid atau tidak }
```

Kamus lokal

arr_d_28 = array of string

arr_d_29 = array of string

arr_d_30 = array of string

arr_d_31 = array of string

arr_m_30 = array of string

arr_m_31 = array of string

d = string

m = string

y = integer

isKabisat() = procedure

Algoritma

arr_d_28 <- ["01", "02", "03", "04", "05", "06", "07", "08",
"09", "10", "11", "12", "13",

 "14", "15", "16", "17", "18", "19", "20", "21", "22",
"23", "24", "25", "26", "27", "28"]

arr_d_29 <- ["01", "02", "03", "04", "05", "06", "07", "08",
"09", "10", "11", "12", "13",

 "14", "15", "16", "17", "18", "19", "20", "21", "22",
"23", "24", "25", "26", "27", "28", "29"]

arr_d_30 <- ["01", "02", "03", "04", "05", "06", "07", "08",
"09", "10", "11", "12", "13", "14",

 "15", "16", "17", "18", "19", "20", "21", "22", "23",

```

"24", "25", "26", "27", "28", "29", "30"]
  arr_d_31 <- ["01", "02", "03", "04", "05", "06", "07", "08",
"09", "10", "11", "12", "13", "14",
  "15", "16", "17", "18", "19", "20", "21", "22", "23",
"24", "25", "26", "27", "28", "29", "30", "31"]
  arr_m_30 <- ["04", "06", "09", "11"]
  arr_m_31 <- ["01", "03", "05", "07", "08", "10", "12"]
  if [0... length(Tgl)] == 10 then
    d <- Tgl[0:2]
    m <- Tgl[3:5]
    y <- int(Tgl[6:10])
    if isKabisat(y) and m == '02' and arr_d_29.count(d) > 0
then
    return True
  else if isKabisat(y) == False and m == '02' and
arr_d_28.count(d) > 0 then
    return True
  else if arr_m_30.count(m) > 0 and arr_d_30.count(d) > 0
then
    return True
  else if arr_m_31.count(m) > 0 and arr_d_31.count(d) > 0
then
    return True
  Else
    return False

```

procedure isKabisat

{ mengecek apakah tahun kabisat atau tidak }

Kamus lokal

tahun = integer

Algoritma

```

  if tahun % 4 == 0 then
    if tahun % 100 == 0 then
      if tahun % 400 == 0 then
        return True
      else
        return False
    else
      return False
  else
    return False

```

procedure Cek_pinjam

```
{ mengecek gadget sedang dipinjam atau tidak }
```

Kamus lokal

i = integer

N = integer

arr = array of array

Algoritma

```
i <- 0
N <- [0... length(arr)]
i traversal N
  if ID != arr[i][2] then
    i = i+1
  else if ID == arr[i][2]
    return True
```

procedure arr_baru

```
{ membuat array baru untuk entry riwayat peminjaman gadget }
```

Kamus lokal

arr2 = array of array

ID = string

Tgl = string

Jml = string

X = array

Algoritma

```
N <- [0... length(arr2)]
X <- [str(N+1), 'User', str(ID), str(Tgl), str(Jml),
'False']
arr2.append(X)
```

Pengujian:

Gambar 4.8 – Gambar pinjam saat inputan salah dan benar

```
Ada yang bisa dibantu, steven?
>>> pinjam
Masukkan ID item: ho
ID tidak valid.
Masukkan ID item: G1
Tanggal peminjaman: 29/02/1999
Tanggal tidak valid.
Tanggal peminjaman: 11/01/2000
Jumlah peminjaman: 10000
Jumlah tidak mencukupi.
Jumlah peminjaman: 100
Item Auto Absen(x100) berhasil dipinjam!
```

F09

PROGRAM MENGEMBALIKAN GADGET

{ SPEK : Untuk admin dan user. Mengembalikan gadget dengan jumlah tertentu sesuai data peminjaaman }
{ Asumsi : semua array, kecuali diatur, telah memiliki banyak elemen yang benar, fungsi bawaan Python sudah ada di notal }

KAMUS

```
{ Deklarasi variabel }
    type gadget :
        < id : integer;
          nama : string;
          desc : string;
          jumlah : integer;
          rarity : string;
          tahun : string >
    type arr_disp :
        < disp_n : integer;
          id_gadget : integer;
          nama_gadget : string;
          jumlah_pinjam : integer;
          tanggal : string;
          id_borrow : integer >
    type disp_tup :
        < arr_disp : arr_disp;
          n_eff : integer >
    type pinjam :
        < id : integer;
          id_peminjam : integer;
          id_gadget : integer;
          tanggal : string;
          jumlah : integer;
          is_returned : boolean >
    type kembali :
        < id : integer;
          id_peminjaman : integer;
          tanggal : string;
          jumlah : integer; >
    type userline :
        < id : integer;
          username : string;
          nama : string;
          alamat : string;
          password : string;
          role : string >
```

```

{ Deklarasi fungsi dan prosedur }

function kembalikan(
  user_data : array[0..] of userline;
  gadget_borrow_history_data : array of pinjam,
  gadget_return_history_data : array of kembali,
  logged_user : string
) -> < array of kembali;
      array of pinjam;
      array of gadget > or string
{ SPEK : sesuai spek modul }
{ I.S. array2 dari berbaagaai csv, inputan user }
{ F.S. array2 baaru, berbagai print out }
KAMUS LOKAL
  borrow_data : array[0..] of pinjam
  return_data : array[0..] of kembali
  disp_tup : disp_tup
  i, n_eff, no_pinjam : integer
  id_user : string
ALGORITMA
  id_user <- findIdFromUsername(user_data, logged_user)
  borrow_data <- gadget_borrow_history_data
  return_data <- gadget_return_history_data
  disp_tup <- fillArrDisp(borrow_data, gadget_data, id_user)
  n_eff <- disp_tup.n_eff
  if not isMeminjam(id_user, borrow_data) do
    print(logged_user + " sedang tidaak meminjam gadget!")
    -> "NULL"
  else
    output("Wah, kamu " + logged_user + " belum
mengembalikan:")

    i transversal[0..n_eff]
    disp_n <- disp_tup[i].arr_disp.disp_n
    nama_gadget <- disp_tup[i].arr_disp.nama_gadget
    jumlah_pinjam <- disp_tup[i].arr_disp.jumlah_pinjam
    output(disp_n + ". " + nama_gadget + ", sebanyak "
+ jumlah_pinjam + " buah")

    kembali.id <- return_data[len(return_data) - 1].id + 1

    output("Masukkan nomor yang ingin dikembalikan: ")
    input(no_pinjam)
    isNoPinjam <- noPinjamValidity(no_pinjam, n_eff)

    idx <- no_pinjam - 1

```



```

        nama_gadget <- disp_tup[idx].arr_disp.nama_gadget
        kembali.id <- disp_tup[idx].arr_disp.id_borrow

        output("Masukkan tanggal pengembalian: ")
        input(kembali.tanggal)
        while not tanggalValid(kembali.tanggal) do
            output("Tanggal tidak valid! Masukkan dengan
formaat DD/MM/YYYY")
            output("Masukkan tanggal pengembalian: ")
            input(kembali.tanggal)

        output("Masukkan banyak " + nama_gadget + " yang
dikembalikan: ")
        input(kembali.jumlah)
        while not jumlahCukup(jumlah_pinjam, kembali.jumlah)
do
            output("Masukkan banyak " + nama_gadget + " yang
dikembalikan: ")
            input(kembali.jumlah)

        return_data.append(kembali)
        borrow_data <-
plusGlobalGadget(disp_tup[idx].arr_disp.id_gadget,
kembali.jumlah, gadget_data)
        gadget_data <- minGlobalBorrow(kembali.id_peminjaman,
kembali.jumlah, borrow_data)

        output("Item " + nama_gadget + "(x" + kembali.jumlah +
") dikembalikan.")
        output("Perubahan disimpan di database sementara.")
        -> (return_data, borrow_data, gadget_data)

function charAreInt(text : string) -> boolean
{ SPEK : Mengecek apakah inputan string bisa jadi integer,
fungsi dari f99 }
{ I.S. text yang akan diperiksa }
{ F.S. boolean }

function noPinjamValidity(no_pinjam : integer, n_eff : integer)
-> boolean
{ SPEK : Cek kevalidan no. Pinjam }

KAMUS LOKAL
    isNoPinjam : boolean

ALGORITMA
    if charAreInt(no_pinjam) then
        if int(no_pinjam) <= n_eff then
            isNoPinjam <- true

```

```

-> isNoPinjam

function findIdFromUsername(user_data : array[0..] of userline,
logged_user : string) -> string
{ SPEK : Mencari id dari data array user, dijamin ada }
{ I.S. array userline dan username terlogin }
{ F.S. id user terlogin }
KAMUS LOKAL
    id_user : integer
    row : useerline
ALGORITMA
    row transversal user_data
    if row.username = logged_user then
        id_user <- row.id
    -> id_user

function isMeminjam(id_user : integer, borrow_data : array[0..]
of pinjam) -> boolean
{ SPEK : Mencari apakah user terlogin sedang meminjam gadget }
{ I.S. data peminjaman, id username terlogin }
KAMUS LOKAL
    i : integer
    isMeminjam : boolean
ALGORITMA
    isMeminjam <- False
    i transversal [1..(len(borrow_data) - 1)]
    if borrow_data[i].id_peminjam = id_user and
borrow_data[i].is_returned = false then
        isMeminjam <- true
    -> isMeminjam

function fillArrDisp(borrow_data : array[0..] of pinjam,
gadget_data : array[0..] of gadget, id_user : integer) ->
<arr_disp, integer>
{ SPEK : Membuat array berisi data2 yang bermanfaat }
{ I.S. data peminjaman dan gadget, username terlogin }
{ F.S. disp_tup yang terisi }
KAMUS LOKAL
    i, disp_n, count : integer
ALGORITMA
    count <- 0
    i transversal [1..(len(borrow_data))]
    if borrow_data[i].id_peminjam = id_user and
borrow_data[i].is_returned = false then
        arr_disp[count].disp_n <- count + 1
        arr_disp[count].id_gadget <-
borrow_data[i].id_gadget

```

```

        arr_disp[count].nama_gadget <-
gadgetName(borrow_data[i].id_gadget)
        arr_disp[count].jumlah_pinjam <-
borrow_data[i].jumlah
        arr_disp[count].tanggal <- borrow_data[i].tanggal
        arr_disp[count].id_borrow <- borrow_data[i].id
        id_user <- borrow_data[I].id_peminjam
        count <- count + 1
    ->(arr_disp, count)

```

```

function gadgetName(id_gadget, gadget_data)
{ SPEK : Mencari nama gadget dengan id terspesifikasi }

```

KAMUS LOKAL

```

    nama_gadget : string
    i : integer

```

ALGORITMA

```

    i transversal [1..(len(gadget_data))]
    if gadget_data[i].id
        nama_gadget <- gadget_data[i].nama
    -> nama_gadget

```

```

function tanggalValid(tanggal):

```

```

{ SPEK : Mengecek apakah tanggal valid dan berformat benar }

```

KAMUS LOKAL

```

    arr_d_28, arr_d_29, arr_d_30, arr_d_31, arr_m_30, arr_m_31
: arr of string
    d, m, y, tanggal : string
    isValid : boolean

```

ALGORITMA

```

    isValid <- false
    arr_d_28 = ["01", "02", "03", "04", "05", "06", "07", "08",
"09", "10", "11", "12", "13", "14", "15", "16", "17", "18",
"19", "20", "21", "22", "23", "24", "25", "26", "27", "28"]
    arr_d_29 = ["01", "02", "03", "04", "05", "06", "07", "08",
"09", "10", "11", "12", "13", "14", "15", "16", "17", "18",
"19", "20", "21", "22", "23", "24", "25", "26", "27", "28",
"29"]
    arr_d_30 = ["01", "02", "03", "04", "05", "06", "07", "08",
"09", "10", "11", "12", "13", "14", "15", "16", "17", "18",
"19", "20", "21", "22", "23", "24", "25", "26", "27", "28",
"29", "30"]
    arr_d_31 = ["01", "02", "03", "04", "05", "06", "07", "08",
"09", "10", "11", "12", "13", "14", "15", "16", "17", "18",
"19", "20", "21", "22", "23", "24", "25", "26", "27", "28",
"29", "30", "31"]
    arr m 31 = ["01", "02", "03", "04", "05", "06", "07", "08",

```

```

"09", "10", "11", "12"]
arr_m_30 = ["04", "06", "09", "11"]
arr_m_31 = ["01", "03", "05", "07", "08", "10", "12"]

if len(tanggal) = 28 then
    d <- tanggal[0..1]
    m <- tanggal[3..4]
    y <- tanggal[6..9]
    if isTahun(y) then
        if arr_d_31.count(d) > 0 and arr_m_31.count(m) > 0
then
            isValid <- true
        else if arr_d_30.count(d) > 0 and arr_m_30.count(m)
> 0 then
            isValid <- true
        else if arr_d_29.count(d) > 0 and isKabisat(y) then
            isValid <- true
        else if arr_d_28.count(d) > 0 and not isKabisat(y)
then
            isValid <- true
        -> isValid

function isTahun(tahun : string) -> bool
{ SPEK : Mencari apakah input tahun valid, fungsi dari f99 }

function isKabisat(tahun : string) -> bool
{ SPEK : Mencari apakah inputan tahun adalah kabisat }
ALGORITMA
    if tahun mod 4 = 0 then
        if tahun mod 100 = 0 then
            if tahun mod 400 = 0 then
                -> true
            else
                -> false
        else
            -> false
    else
        -> false

function jumlahCukup(jumlah_pinjam : array[0..] of pinjam,
jumlah_input : integer) -> boolean
{ SPEK : Mencari apakah jumlah gadget yang dikembalikan > yang
dipinjam }
KAMUS LOKAL
    isCukup : boolean
ALGORITMA

```

```

isCukup <- false
if not charAreInt(jumlah_input) then
    output("Inputan tidak valid! Mohon masukkan lagi.")
else if jumlah_pinjam < jumlah_input then
    output("Inputan melebihi tunggakan! Mohon masukkan
lagi.")
else:
    isCukup <- true
-> isCukup

function minGlobalBorrow(id_peminjaman : integer, jumlah_input
: integer, borrow_data array[0..] of pinjam) ->
{ SPEK : Megurangi jumlah tunggakan di file borrow, mengubah
is_returned jika perlu }
KAMUS LOKAL
    i : int
ALGORITMA
    i transversal [1..(len(borrow_data) - 1)]:
        if borrow_data[i].id == id_peminjaman then
            borrow_data[i][4] = str(int(borrow_data[i][4]) -
int(jumlah_input))
            if borrow_data[i].jumlah <= "0":
                borrow_data[i].is_returned = "true"

    -> borrow_data

function plusGlobalGadget(id_gadget : string, jumlah_input :
integer, gadget_data : array[0..] of gadget)
{ SPEK: Menambahkan barang yang sudaah dikembalikan ke file
global }
KAMUS LOKAL
    gadget_data : array[0..] of gadget
ALGORITMA
    i transversal[1..(len(gadget_data) - 1)]
        if gadget_data[i].id = id_gadget then
            gadget_data[i].jumlah <- gadget_data[i].jumlah +
jumlah_input
    -> gadget_data

```

Pengujian:

Gambar 4.9 – Gambar testing riwayatpinjam

```

Perintah aku, daddy raka~!
>>> riwayatpinjam

ID Peminjaman      :4
Nama Pengambil      :darkness
Nama Gadget         :Spoiler Sinker
Tanggal Peminjaman  :02/02/2121
Jumlah              :45

ID Peminjaman      :6
Nama Pengambil      :raka
Nama Gadget         :Auto Absen
Tanggal Peminjaman  :21/06/2071
Jumlah              :44

ID Peminjaman      :2
Nama Pengambil      :thousandface
Nama Gadget         :Spoiler Sinker
Tanggal Peminjaman  :02/02/2021
Jumlah              :45

ID Peminjaman      :1
Nama Pengambil      :raka
Nama Gadget         :Art Code Generator
Tanggal Peminjaman  :21/06/2002
Jumlah              :44

ID Peminjaman      :3
Nama Pengambil      :aquasama
Nama Gadget         :Kilometer Catnip
Tanggal Peminjaman  :04/02/2000
Jumlah              :47

Lihat entry selanjutnya? (Y/N) :

```

F10

KAMUS

ambil: boolean
 ada_di_database: boolean
 new_consumableArr: array
 new_arr: array

procedure minta (input id_item: string, input jumlah: integer, input tanggal_minta: string, input array_of_consumable: array of array of string)

{Initial State:

Menerima input berupa id_item, jumlah item, tanggal minta, dan array_of_consumable yang dimana array_of_consumable tersebut akan diproses sesuai dengan input dari pengguna}

{Final State:

Membuat array_of_consumable yang baru sesuai dengan modifikasi jumlah yang telah dilakukan user. Syarat: permintaan user tidak boleh melebihi apa yang ada pada database.}

{Asumsi command "length" dan "append" terdapat dalam notasi

```
algoritmik}
```

KAMUS LOKAL

```
i: integer {indeks looping}  
j: integer {indeks looping}  
k: integer {indeks looping}  
l: integer {indeks looping}
```

ALGORITMA

```
ambil <- false  
ada_di_database <- false  
new_consumableArr <- []  
  
i traversal [0..length(array_of_consumable)]  
  if id_item = i[0] then  
    ada_di_database <- true  
  
  if ada_di_database = false then  
    output("Tidak ada item dengan ID tersebut")  
  else {ada_di_database = true}  
    j traversal [0..length(array_of_consumable)]  
      new_consumableArr.append(array_of_consumable[j][0])  
    k traversal [0..length(array_of_consumable)]  
      if id_item = array_of_consumable[k][0] then  
        if array_of_consumable[k][3]-jumlah >= 0 then  
          array_of_consumable[k][3] =  
array_of_consumable[k][3] - jumlah  
          ambil <- true  
        else {array_of_consumable[k][3]-jumlah < 0}  
          output("Item tidak mencukupi permintaan")  
          ambil <- false  
      new_arr <- write_new_array {Memanggil fungsi untuk  
membuat array baru hasil perubahan yang telah dilakukan}  
      array_to_csv(new_arr) {Memanggil fungsi bantuan untuk  
membuat array menjadi csv dengan delimiter ";"}  
  
    if ambil = true then  
      l traversal [0..length(array_of_consumable)]  
      if id_item = array_of_consumable[l][0] then  
        output("Item " + array_of_consumable[l][1] + "  
(x" + jumlah + ") " + "telah berhasil diambil!")
```

Pengujian:

Gambar 4.10 – Gambar fungsi minta apabila ID salah atau jumlah salah

```

Ada yang bisa dibantu, steven?
>>> minta
Masukan ID item: C10
Jumlah: 1
Tanggal permintaan: 2/05/2021
Item Obat Bahagia (x1) telah berhasil diambil!

Ada yang bisa dibantu, steven?
>>> minta
Masukan ID item: G1
Jumlah: -10
Tanggal permintaan: 2/05/2021
Jumlah permintaan minimal adalah 1
Tidak ada item dengan ID tersebut

Ada yang bisa dibantu, steven?
>>> minta
Masukan ID item: c8
Jumlah: 1
Tanggal permintaan: 2/05/2021
Tidak ada item dengan ID tersebut

Ada yang bisa dibantu, steven?
>>> minta
Masukan ID item: C10
Jumlah: takhingga
Input jumlah harus integer

```

F11

PROGRAM MELIHAT RIWAYAT PEMINJAMAN GADGET

{Spesifikasi: Melihat riwayat peminjaman gadget yang telah terurut berdasarkan tanggal dari peminjaman yang paling baru}

KAMUS

{Deklarasi Variabel}

type gadget_borrow_history:

```

    <   id           : str
        id_peminjam  : str
        Id_gadget    : str
        tanggal     : str
        jumlah       : str
        is_returned  : str >

```

type gadget :

```

    <   id      : str
        nama    : str
        desc    : str
        jumlah  : str
        rarity   : str
        tahun   : str >

```

type user :

```

    <   id      : str

```



```

        username : str
        nama      : str
        alamat    : str
        password  : str
        role      : str >
{Deklarasi Fungsi dan Prosedur}
Procedure riwayatpinjam (
    Input gadget_borrow_history_data : array of gadget borrow
history;
    Input gadget_data : array of gadget;
    Input User_data : array of user;
    Input logged_user : string)
{Spesifikasi: Melihat riwayat peminjaman gadget yang telah
terurut berdasarkan tanggal dari peminjaman yang paling baru}
{I.S. Array dari gadget_borrow_history.csv dan gadget.csv}
{F.S. Print out informasi tentang riwayat peminjaman}
KAMUS LOKAL
    Header : array of string
    Lanjut: string
    Idx : integer
    Finish : bool
    Sorted_tanggal : array of gadget_borrow_history
ALGORITMA
    if isAdmin (logged_user, user_data) then
        header <- gadget_borrow_history_data.pop(0)
        sorted_tanggal <-
sorted(gadget_borrow_history_data, key=lambda date:
datetime.strptime(date[3], '%d/%m/%Y'), reverse=True)
        gadget_borrow_history_data.insert(0, header)
        Finish <- False {inisiasi}
        Idx <- -1 {inisiasi}
        Repeat
            I traversal [0..4]
                Idx <- idx + 1
                If idx >= length(sorted_tanggal) then
                    Finish <- True
                Else
                    output("\nIDPeminjaman\t\t:" + sorted_tanggal[idx][0])
                    output("Nama
Pengambil\t\t:" + carinama(sorted_tanggal[idx][1], user_data))
                    output("Nama
Gadget\t\t:" + carigadget(sorted_tanggal[idx][2], gadget_data))
                    output("Tanggal
Peminjaman\t\t:" + sorted_tanggal[idx][3])

                    output("Jumlah\t\t\t\t:" + sorted_tanggal[idx][4])
                until finish= True

```

```
function carinama (id: string, user_data: array of user) ->
string
```

```
{Spesifikasi: Mencari nama user dengan id sesuai dengan input}
```

KAMUS LOKAL

```
Nama, id : string
```

```
User_data : array of user
```

ALGORITMA

```
I Traversal [0..length(user_data)]
```

```
  If id = user_data[i][0] then
```

```
    Nama <- user_data[i][0]
```

```
    Output (nama)
```

```
function carigadget (id: string, gadget_data: array of gadget)
-> string
```

```
{Spesifikasi: Mencari nama gadget dengan id sesuai dengan
input}
```

KAMUS LOKAL

```
Nama, id : string
```

```
gadgetUser_data : array of gadget
```

ALGORITMA

```
I Traversal [0..length(gadget_data)]
```

```
  If id = gadget_data[i][0] then
```

```
    Nama <- gadget_data[i][0]
```

```
    Output (nama)
```

Pengujian:

Gambar 4.11 – Gambar testing riwayatpinjam

```
>>> riwayatpinjam

ID Peminjaman      :3
Nama Pengambil      :hiyapeko
Nama Gadget         :Notasi Converter
Tanggal Peminjaman :05/05/2021
Jumlah              :11

ID Peminjaman      :2
Nama Pengambil      :hiyapeko
Nama Gadget         :Auto Absen
Tanggal Peminjaman :05/02/2021
Jumlah              :10
```

F12

PROGRAM RIWAYAT PENGEMBALIAN GADGET

```
( Spesifikasi : Hanya untuk admin, menampilkan riwayat
pengembalian gadget:
```

1. bila terdapat lebih dari 5 entry, menampilkan 5 entry

paling baru

2. dapat mengeluarkan 5 entry tambahan bila diinginkan

3. keluaran harus *sorted descending* berdasarkan tanggal)

{ I.S. : array dari gadget_return_history.csv,
gadget_borrow_history.csv, dan gadget.csv }

{ F.S. : mengeluarkan riwayat pengembalian gadget }

procedure riwayat_pengembalian_gadget (input
gadget_return_history_data : array of gadget_return_history,
input gadget_borrow_history_data : array of
gadget_borrow_history, input gadget_data : array of gadget)

Kamus

arr1 : array of array

arr2 : array of array

arr3 : array of array

riwayat : procedure

x : string

Algoritma

arr1 <- gadget_return_history_data

arr2 <- gadget_borrow_history_data

arr3 <- gadget_data

riwayat(arr1, arr2, arr3)

del arr1[[0... length(arr1)] : [0... length(arr1) -6] : -1]

if [0... length(arr1)] >= 2 then

Output('Apakah ingin menampilkan 5 riwayat lagi?')

Input(x)

Output('')

if x == 'YA' then

riwayat_pengembalian_gadget(arr1, arr2, arr3)

else

Output('Riwayat sudah ditampilkan semua.')

procedure riwayat

{ mengambil 5 riwayat pengembalian gadget yang paling baru }

Kamus lokal

N : integer

X : array

arr1 : array of array

arr2 : array of array

arr3 : array of array

```

sort_tgl() : procedure
indeks : integer
output : procedure

```

Algoritma

```

N <- [0... length(arr1)]
X <- []
k traversal (N, N-5, -1)
  X = X+[arr1[N-1]]
  N = N-1
  if N == 1 then
    break
M = sort_tgl(X)
k traversal [0... length(M)]
  indeks = M.index(max(M))
  output(X, arr2, arr3, indeks)
  M[indeks] = -999

```

procedure sort_tgl

```
{ membuat array yang berisikan tanggal pengembaliannya }
```

Kamus lokal

```

i : integer
j : array
N : integer
l : string
d : string
m : string
y : string
z : integer

```

Algoritma

```

i <- 0
j <- []
N <- [0... length(X)]
i traversal N
  l <- X[i][1]
  d <- l[0:2]
  m <- l[3:5]
  y <- l[6:10]
  z <- int(y+m+d)
  i = i+1
  j.append(z)
return j

```

procedure output

{ mengeluarkan output }

Kamus lokal

X : array

arr1 : array of array

arr2 : array of array

arr3 : array of array

j : integer

index : integer

indeks : integer

Algoritma

```
j traversal [0... length(arr2)]
  if X[i][0] == arr2[j][0] then
    index <- j
k traversal [0... length(arr3)]
  if arr2[index][2] == arr3[k][0] then
    indeks <- k
  Output('ID Pengembalian           :',X[i][0])
  Output('Nama Pengambil           :',arr2[index][1])
  Output('Nama Gadget              :',arr3[indeks][1])
  Output('Tanggal Pengembalian:',X[i][2])
  Output('')
```

Pengujian:

Gambar 4.12 - Gambar riwayat pengembalian gadget

```
Perintah aku, master steven~!
>>> riwayatkembali
ID Pengembalian      : 1
Nama Pengambil       : id_peminjam
Nama Gadget          : nama
Tanggal Pengembalian : 05/05/2066

Riwayat sudah ditampilkan semua.
```

F13

PROGRAM MELIHAT RIWAYAT PENGAMBILAN CONSUMABLE

```
{ SPEK : Membaca dan mengout-putkan file riwayat pengambilan, 5
riwayat terakhir dan bisa diperpanjang }
{ Asumsi : semua array, kecuali diatur, telah memiliki banyak
elemen yang benar, fungsi bawaan Python sudah ada di notal }
{ I.S. array consumable_history_data }
{ F.S. output data ke terminal, sesuai format }
```

KAMUS

```
{ Deklarasi Variabel }
  type consumable_history :
    < id : integer,
      id_pengambil : integer,
      id_consumable : integer,
      tanggal : string,
      jumlah : integer >
  type userline :
    < id : integer
      username : string
      nama : string
      alamat : string
      password : integer
      role : string >
  type consumable :
    < id : string
      nama : string
      desc : string
      jumlah : string
      rarity : string >
```

```
{ Deklarasi fungsi dan prosedur }
```

```
procedure riwayatambil(
  input user_data : array[0..] of userline;
  input consumable_data : array[0..] of consumable;
  input consumable_history_data : array[0..] of
consumable_history;
  input logged_user : string)
{ SPEK : Sesuai modul }
```

KAMUS

```
length, i : integer
command : string
```

ALGORITMA

```
  if f99.isAdmin(logged_user) then
    length <- len(consumable_history_data)
    if length > 1 then
      consumable_history_data <-
sortByDate(consumable_history_data, length)

    if length = 0 then
      output("Data kosong! Tidak ada riwayat.")
    else if 0 < length <= 5 then
```

```

        output"Data " + length + "x pengambilan terakhir:")
        i transversal[(length - 1)..0]
        printOneRow(i, user_data, consumable_data,
consumable_history_data)
    else { length > 5 }
        while length > 1 do
            if length < 5 then
                output("Data " + str(length) + "x
pengambilan terakhir:")
            else:
                output("Data 5x pengambilan terakhir:")

        length <- printFiveRow(length, user_data,
consumable_data, consumable_history_data)

        while length > 1 do
            output("Ingin melihat 5 entry sebelumnya?
(y/n): ")
            input(command)
            if command.strip().lower() = "n" then
                length = 0
                break
            else if command.strip().lower() == "y" then
                break
            else
                output("Inputan tidak valid! Silakan
masukkan lagi.\n")
        else
            output("Halt! Fungsi terbatas admin.")

```

```

procedure printOneRow(input idx : integer, input user_data :
array[0..] of userline, input consumable_data : array[0..] of
consumable, consumable_history_data : array[0..] of
consumable_history )
{ SPEK : Print 1 buah row }

```

ALGORITMA

```

    output("ID Pengambilan: " +
consumable_history_data[idx].id)
    output("Nama pengambil: " +
getUsername(consumable_history_data[idx].id_pengambil,
user data))

```

```

        output("Nama Consumable: " +
getConsumName(consumable_history_data[idx].id_consumable,
consumable_data))
        output"Tanggal Pengambilan: " +
consumable_history_data[idx].tanggal_pengembalian)
        output("Jumlah: " + consumable_history_data[idx].jumlah)
        output("")

```

```

function getUsername(id : integer, user_data : array of
userline)-> string
{ SPEK : Mendapatkan username dari user_data dengan id user }

```

ALGORITMA

```

    row transversal user_data
        if row.id = id then
            username = row.nama
-> username

```

```

function printFiveRow(length : integer, user_data : array of
userline, consumable_data : array of consumable,
consumable_history_data : array of consumable_history) ->
integer
{ SPEK : Print 5 buah row, lalu return length - 5 }

```

ALGORITMA

```

    i <- 0
    iterate
        printOneRow((length - 1), user_data, consumable_data,
consumable_history_data)
        length <- length - 1
    stop (length <= 1 or i = 5) { EOP }
        i <- i + 1
    if length <= 0 then
        output("Data selesai.")
-> length

```

```

function sortByDate(consumable_history_data: array[0..] of
consumable_history, length : integer)-> array[0..] of
consumable_history
{ SPEK : Mengurutkan list dari date terawal ke date terakhir }

```

KAMUS LOKAL


```

i, j, idx_max, t, t_max, m, m_max, ddmax : integer
row_temp : consumable_history
ALGORITMA
{ 1. Year sort }
i transversal[1..length - 1]
  { Find max idx }
  idx_max <- i
  t_max <- consumable_history_data[i][6..9].tanggal
  j transversal[i + 1..length - 1]
  t <- consumable_history_data[j][6..9].tanggal
  if t > t_max then
    idx_max <- j

  { Swap }
  row_temp <- consumable_history_data[idx_max]
  consumable_history_data[idx_max] <-
consumable_history_data[i]
  consumable_history_data[idx_max] <- row_temp

{ 2. Month sort }
i transversal[1..length - 1]
  { Find max idx }
  idx_max <- i
  m_max = consumable_history_data[i][3..4].tanggal
  j transversal[i + 1..length - 1]
  m = consumable_history_data[j][3..4].tanggal
  if m > m_max then
    idx_max <- j

  { Swap }
  row_temp = consumable_history_data[idx_max]
  consumable_history_data[idx_max] <-
consumable_history_data[i]
  consumable_history_data[idx_max] <- row_temp

{ # 3. Day sort }
i transversal[1..length - 1]
  { Find max idx }
  idx_max <- i
  m_max = consumable_history_data[i][0..1].tanggal
  j transversal[i + 1..length - 1]
  d = consumable_history_data[j][3][0:2]
  if d > d_max then

```

```

        idx_max <- j

        { Swap }
        row_temp <- consumable_history_data[idx_max]
        consumable_history_data[idx_max] <-
consumable_history_data[i]
        consumable_history_data[idx_max] <- row_temp

        { Sorted based on date }
        -> consumable_history_data

function getConsumName(id : integer, consumable_data : array of
consumable) -> string
{ SPEK : Mendapatkan nama consumable dari consumable_data
dengan id consumable }
KAMUS LOKAL
    name : string
    row : arr of consumable
ALGORITMA
    for row in consumable_data:
        if row.id = id then
            name <- row.nama
    -> name

```

Pengujian:

Gambar 4.13 – Gambar riwayatambil ketika barang yang diambil dihapus dari csv

```

>>> riwayatambil
Traceback (most recent call last):
  File "kantongajaib.py", line 210, in <module>
    displayOptions(command.lower().strip(" "))
  File "kantongajaib.py", line 114, in displayOptions
    f13.riwayatambil(user_data, consumable_data, consumable_history_data, logged_user)
  File "C:\Users\Asus\Desktop\TUBES_Kel6_K03\f13.py", line 47, in riwayatambil
    if f99.isAdmin(logged_user):
TypeError: isAdmin() missing 1 required positional argument: 'user_data'

```

F14

KAMUS

array: array

function load(nama_folder_csv: string, nama_file: string) ->
array of array of string
{Initial State:
Menerima masukan nama folder database csv dan nama file csv}
{Final State:

Mengkonversikan file csv tersebut menjadi array of array of string}

function csv_to_arr(nama_file: string) -> array of array of string

{Initial State:

Menerima masukan nama file csv}

{Final State:

Mengkonversi file csv tersebut menjadi array of array of string}

KAMUS LOKAL

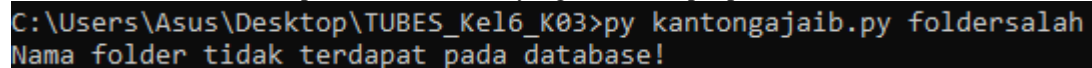
{-}

ALGORITMA

```
    depend on (nama_file)
        user_csv: array <- csv_to_arr(nama_folder_csv +
"/user.csv")
        gadget_csv: array <- csv_to_arr(nama_folder_csv +
"/gadget.csv")
        consumable_csv: array <- csv_to_arr(nama_folder_csv +
"/consumable.csv")
        consumable_history_csv: array <-
csv_to_arr(nama_folder_csv + "/consumable_history.csv")
        gadget_borrow_history_csv: array <-
csv_to_arr(nama_folder_csv + "/gadget_borrow_history.csv")
        gadget_return_history_csv: array <-
csv_to_arr(nama_folder_csv + "/gadget_return_history.csv")
    else:
        -> "Nama File yang Anda Masukkan Tidak Ada Dalam Folder
CSV"
    -> array {Me-return array}
```

Pengujian:

Gambar 4.14.1 – Gambar apabila diload folder yang tidak terdapat pada database



```
C:\Users\Asus\Desktop\TUBES_Kel6_K03>py kantongajaib.py foldersalah
Nama folder tidak terdapat pada database!
```

Gambar 4.14.2 – Gambar apabila diload folder yang terdapat pada database

```

C:\Users\Asus\Desktop\TUBES_Kel6_K03>py kantongajaib.py CSV
Loading...
Loading...
Loading...
Folder CSV telah berhasil di load

PROGRAM KANTONG AJAIB IF1201 K-03 Kel. 6.
2021. All rights not reserved.

Masukan username:

```

F15

PROGRAM MENYIMPAN DATA

{Spesifikasi: Penyimpanan semua file ke folder yang diinput}

KAMUS

{Deklarasi Variabel}

type gadget_borrow_history:

```

    < id : str
      id_peminjam : str
      Id_gadget : str
      tanggal : str
      jumlah : str
      is_returned : str >

```

type gadget_return_history:

```

    < id : str
      id_peminjam : str
      tanggal : str >

```

type consumable_history:

```

    < id : str
      id_peminjam : str
      Id_gadget : str
      tanggal : str
      jumlah : str >

```

type gadget :

```

    < id : str
      nama : str
      desc : str
      jumlah : str
      rarity : str
      tahun : str >

```

type user :

```

    < id : str
      username : str
      nama : str
      alamat : str
      password : str
      role : str >

```

type consumable :

```

    <   id : str
        nama : str
        desc : str
        jumlah : str
        rarity : str >
{Deklarasi Fungsi}
procedure save (
    input user_data : array of user;
    input gadget_data : array of gadget;
    input consumable_data : array of consumable;
    input gadget_borrow_history_data : array of
gadget_borrow_history;
    input gadget_return_history_data : array of
gadget_return_history;
    input consumable_history_data : array of
consumable_history)
{Spesifikasi : Menerima input nama folder dan menyimpan semua
variabel menjadi file csv di dalam folder tersebut)
{I.S Array of array dari user.csv, gadget.csv, consumable.csv,
Gadget_borrow_history.csv, Gadget_return_history.csv,
consumable_history.csv}
{F.S Folder dengan semua file bernama sesuai input}

```

KAMUS LOKAL

```

    All_str, All_file : array of strings
    Folder : string

```

ALGORITMA

```

    all_str <- [data_str(user_data), data_str(gadget_data),
data_str(consumable_data),
data_str(gadget_borrow_history_data),
data_str(gadget_return_history_data),
data_str(consumable_history_data)]
    all_file <- ["user.csv", "gadget.csv", "consumable.csv",
"gadget_borrow_history.csv", "gadget_return_history.csv",
"consumable_history.csv"]
    output("masukkan nama folder penyimpanan: ")
    input(folder)
    if folder not in os.listdir() then
        os.mkdir(folder) {pembuatan folder bila diperlukan}
    I traversal [length(all_file)]
        Savefile(all_file[i], folder, all_str[i])
    Output("Saving...")
    Output("Data telah disimpan pada folder" + folder)

```

```

Procedure savefile(input file: string, input folder: string,
input string: string)
{Spesifikasi: menyimpan file berisi string di dalam folder,
apabila folder sudah memiliki file tersebut maka program akan

```

```
menghapus dan menggantikan file tersebut}
{I.S. string nama file, folder, dan isi string}
{F.S. file berisi srtring tersimpan dalam folder}
```

KAMUS LOKAL

```
File, folder, string: string
f : SEQ file of type
```

ALGORITMA

```
If file in os.listdir(folder) then
    Os.remove(folder+"/"+file){penghapusan file dari folder
untuk digantikan}
Assign (f, file)
Rewrtite (f)
Write (f,string)
Close (f)
```

Function data_str(data: array of array of string) -> string
{Spesifikasi : Mengubah array of array of string menjadi string dengan pemisah ";" sesuai dengan format csv}

KAMUS LOKAL

```
String : string
Data : array of array of string
```

ALGORITMA

```
String <- "" {inisiasi}
i traversal [length(data)]
    string <- string + ":".join(data[i])
    string <- string + "\n"
output(string)
```

Pengujian:

Gambar 4.15 – Gambar pengujian save

10 Auto Absen Berhasil ditambahkan. Stok sekarang: 430

```
Perintah aku, daddy raka~!
>>> save
Masukkan nama folder penyimpanan: test
Saving...
Data telah disimpan pada folder test!
```

F16

PROGRAM HELP

```
{ Spesifikasi : memberikan panduan penggunaan sistem, dan tidak
perlu melakukan login terlebih dahulu untuk menggunakan menu
ini.}
{ I.S. : menerima input tulisan "HELP" }
{ F.S. : mengeluarkan panduan penggunaan sistem }
```

procedure help

Algoritma

```
Output('===== HELP =====')
Output('register - untuk melakukan registrasi user baru')
Output('login - untuk masuk ke dalam akun user yang telah
teregristasi')
Output('carirarity - untuk mencari gadget berdasarkan
raritynya')
Output('caritahun - untuk mencari gadget berdasarkan tahun
ditemukannya')
Output('tambahitem - untuk menambah item ke dalam
inventori')
Output('hapusitem - untuk menghapus item dari inventori')
Output('ubahjumlah - untuk mengubah jumlah gadget atau
consumable yang ada di inventori')
Output('pinjam - untuk meminjam gadget dari inventori')
Output('kembalikan - untuk mengembalikan gadget yang telah
dipinjam kembali ke inventori')
Output('minta - untuk meminta consumable dari inventori')
Output('riwayatpinjam - untuk melihat riwayat peminjaman
gadget')
Output('riwayatkembali - untuk melihat riwayat pengembalian
gadget')
Output('riwayatambil - untuk melihat riwayat pengambilan
consumable')
Output('save - untuk menyimpan perubahan data pada file')
Output('help - untuk memberikan panduan penggunaan sistem')
Output('exit - untuk keluar dari aplikasi')
```

Pengujian:

Gambar 4.16 - Gambar output help

```

Ada yang bisa dibantu, steven?
>>> help
===== HELP =====
register - untuk melakukan registrasi user baru
login - untuk masuk ke dalam akun user yang telah teregritasi
carirarity - untuk mencari gadget berdasarkan raritynya
caritahun - untuk mencari gadget berdasarkan tahun ditemukannya
tambahitem - untuk menambah item ke dalam inventori
hapusitem - untuk menghapus item dari inventori
ubahjumlah - untuk mengubah jumlah gadget atau consumable yang ada di inventori
pinjam - untuk meminjam gadget dari inventori
kembalikan - untuk mengembalikan gadget yang telah dipinjam kembali ke inventori
minta - untuk meminta consumable dari inventori
riwayatpinjam - untuk melihat riwayat peminjaman gadget
riwayatkembali - untuk melihat riwayat pengembalian gadget
riwayatambil - untuk melihat riwayat pengambilan consumable
save - untuk menyimpan perubahan data pada file
help - untuk memberikan panduan penggunaan sistem
exit - untuk keluar dari aplikasi

```

F17

PROGRAM Exit

```

{ Spesifikasi : Exit program secara umum dengan pilihan save }
{ import fungsi dari f15 as f15 }

```

KAMUS

```

{ Deklarasi fungsi dan prosedur }

```

```

function exit(isRunning : boolean) -> boolean
{ SPEK : Sesuai spek modul }

```

KAMUS LOKAL

```

char : character

```

ALGORITMA

```

Iterate
    output("Apakah Anda mau menyimpan semua perubahan? (y/n):")
)
    if char.upper().strip() = "Y" then
        f15.save()
        output("Semua perubahan berhasil disimpan.")
        isRunning <- False
        output("Terima kasih telah menggunakan Kantong Ajaib.
~Byeonara!\n")
        stop (true)

    else if char.upper().strip() = "N" then
        output("Semua perubahan *tidak* disimpan, dan telah

```



```

dihapus.")
    isRunning <- False
    output("Terima kasih telah menggunakan Kantung Ajaib.
~Byeonara!\n")
    stop (true)

    else { Not y, Y, n, or N }
        output("Input tidak valid! Silakan masukkan ulang.")

-> isRunning

```

Pengujian:

Gambar 4.17.1 – Gambar exit ketika n

```

>>> exit
Apakah Anda mau menyimpan semua perubahan? (y/n): n
Semua perubahan *tidak* disimpan, dan telah dihapus.
Terima kasih telah menggunakan Kantung Ajaib. ~Byeonara!

```

Gambar 4.17.2 – Gambar exit ketika y

```

>>> exit
Apakah Anda mau menyimpan semua perubahan? (y/n): y
Masukkan nama folder penyimpanan:

```

FB01

KAMUS

```

hasil: array of integer
samaran: string
nilai: integer

```

function encrypt (key: string) -> string
{menerima input bertipe string misalnya sandi, lalu mengoutput sandi yang telah dienkripsi}
{asumsikan ord, length, dan append dalam python terdapat pada notasi algoritmik}

KAMUS LOKAL

```

i: integer {indeks}
j: integer {indeks}

```

ALGORITMA

```

hasil <- []
samaran <- ""
nilai <- 0

i traversal [0..length(key)]
    nilai <- ord(key[i])
    hasil.append(nilai)

```

```

        nilai <- ord(key[i]) - 1
        hasil.append(nilai)
        if ord(key[i]) mod 2 = 0 then
            nilai <- ord(key[i]) mod 10
            hasil.append(nilai)
        else {ord(key[i] mod 2 <> 0}
            nilai <- ord(key[i] div 10
            hasil.append(nilai)

    j travelsal [0..length(hasil)]
    samaran <- samaran & hasil[j]

-> samaran {me-return samaran, yaitu decrypted password}

```

FB02

Telah terintegrasi dengan F09. Lihat F09.

F99

PROGRAM KUMPULAN FUNGSI

{ Spesifikasi: Modul tambahan, kumpulan berbagai fungsi yang bermanfaat. Sudah disetujui oleh asisten. }

KAMUS

{ Deklarasi fungsi dan prosedur }

function charAreInt(text : array[0..] of character) -> boolean
 { Spesifikasi : Mengecek apakah inputan : str bisa dijadikan int }

KAMUS LOKAL

char : character
 areInt : boolean

ALGORITMA

```

areInt <- true
char transversal text
    if not (48 <= ord(char) <= 57) then
        areInt <- false
return areInt

```

function isTahun(tahun : array[0..] of character) -> boolean
 { Spesifikasi : Mengecek apakah tahun int berformat YYYY }

ALGORITMA

```

if charAreInt(tahun) then
    if 1000 <= tahun <= 9999 then
        -> true
    else

```

```

        -> false
    else:
        -> false

function isAdmin(logged_user : string, user_data : array of
userline) -> boolean
{ Spesifikasi : Mengecek apakah pengguna adalah admin }

KAMUS LOKAL
    # idx, i : int
    # userIsAdmin : bool

ALGORITMA
    userIsAdmin <- False
    idx <- 0
    i transversal[1..(len(user_data) - 1)]: # Pengecekan
username
        if logged_user = user_data[i].username then
            idx <- i
        if user_data[idx].role.lower() = "admin" then
            userIsAdmin <- True

    -> userIsAdmin

```

LAMPIRAN

Gambar Form Asistensi Pertama

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

Nomor Asistensi : 1
No. Kelompok/Kelas : 6/K03
Tanggal asistensi : 9 April 2021

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
1	16520193 / Zhillan Attarizal <u>Rezvarifin</u>
2	16520203 / Steven
3	16520353 / Tubagus Baraka Kautsar S.
4	16520413 / Jonathan
5	
6	
Asisten pembimbing	NIM / Nama
	13518114 / Mario Gunawan

Rangkuman Diskusi
<ol style="list-style-type: none"> 1. Interface? Cuma command line di Python aja. 2. Tanggal descending? Dari yang paling bar uke yang paling lama. 3. Csv gimana ngolahnya? Pernama. Ipake fungsi buat baca datanya, terus dijadiin variable, terus bisa diubahz, terus kalo udah selesai, save. Tidak perlu import buat baca dll, ada built in, ada tutorialnya yang di-share Michelle. 4. Design? Design sama kode biasanya sama sih. Design itu buikun alur pikir. Misal Zhillan mikir "stepnya gijz", terus Steven yang nulis, berarti Zhillan designer dan Steven coder. 5. Dekomposisi: kayak ngejelasin tiap fungsinya. 6. Spesifikasi: kode yang udah dibikin tulis ulang dalam notasi algoritmik. 7. Sharez lewat gdrive, 1 file 1 modul 8. 17 file, terus nanti import aja
Tindak Lanjut
-

Gambar Form Asistensi Kedua

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

Nomor Asistensi : 2
No. Kelompok/Kelas : 6/Ko3
Tanggal asistensi : 28 April 2021

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	16520193 / Zhillan Attarizal <u>Rezvarifin</u>
	2	16520203 / Steven
	3	16520353 / Tubagus Baraka Kautsar S.
	4	16520413 / Jonathan
	5	
Asisten pembimbing	NIM / Nama	
	13518144	

Catatan Asistensi:

Rangkuman Diskusi	
1.	Notasi algoritmik? Kalo udah jelas, <u>gausah</u> kasih <u>komen</u> lrbh. Bagian <u>dkomp</u> itu <u>emang</u> alur <u>utamanya</u> aja. Flowchart juga <u>sabji</u> .
2.	Testing beda sama yang <u>biikin</u>
3.	append fungsinya sama seperti assign <u>variabel</u> baru, sehingga yang <u>direturn</u> adalah variabel lama yang sudah dimodifikasi (<u>user_data</u>), bukan prosesnya (<u>user_data.append(userline)</u>)
4.	Current working bermasalah, kenapa? Dalam Vscode, current working directory ditentukan oleh cara pengguna membuka file/folder tertentu maka hal tersebut perlu diperhatikan.
5.	Notal, ganti tipe gimana caranya? <u>Gausah</u> , <u>asumsiin</u> di <u>file</u> si <u>type</u> <u>nya</u> <u>usah</u> betul
6.	Masalah yang berhubungan sama pop. Mending pake arr kosong
7.	pop list tidak ke loop, kenapa ya? index nya tidak berubah
Tindak Lanjut	
-	