# TUGAS BESAR

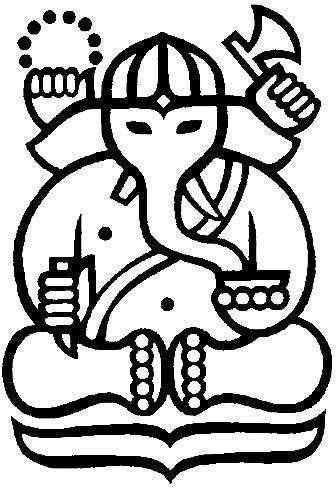
**IF1210 DASAR PEMROGRAMAN KANTONG DORAEMON**

Oleh

**Kelompok 01**

**Kelas 03**

|  |  |
| --- | --- |
| **Usman Zakaria** | **16520003** |
| **Muhammad Garebaldhie** | **16520273** |
| **Yosafat Raditya** | **16520293** |
| **Nicholas Glenn** | **16520343** |



**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA INSTITUT TEKNOLOGI BANDUNG**

**2021**

# PEMBUKAAN

Puji dan syukur kami panjatkan kepada Tuhan Yang Mahas Esa atas rahmat dan karunia yang telah diberikan sehingga penulis diberi kemudahan dalam penulisan laporan ini. Tanpa pertolongan-Nya penulis tidak akan dapat menyelesaikan laporan ini dengan baik. Laporan ini berisikan hasil pengerjaan kami dalam mata kuliah Dasar Pemrograman. Tujuan dari pembuatan laporan ini sebagai salah satu syarat untuk memenuhi kriteria penilaian mata kuliah Dasar Pemrograman juga untuk menjadi bahan evaluasi kami terhadap proses-proses yang telah dilakukan terhadap modul-modul spesifikasi yang telah diberikan. Harapan kami laporan ini dapat bermanfaat untuk masyarakat terutama mahasiswa ITB. proses dalam penulisan laporan ini dimulai dengan memahami spesifikasi yang telah diberikan dalam modul, kemudian mencoba untuk membuat kerangka algoritmanya dan dilanjutkan dengan membuat kode agar pemrograman dapat berjalan sesuai dengan apa yang sudah tertera dalam modul spesifikasi.

Kendala yang terdapat selama mengerjakan tugas besar ini adalah terdapat banyak ujian serta tugas besar lain yang harus dilalui oleh penulis sehingga pembagian waktu untuk penyelesaian tugas besar dasar pemrograman sekaligus pembuatan laporan ini cukup sulit. Tidak lupa penulis menyampaikan terima kasih yang sebesar-besarnya kepada pihak yang telah membantu dalam penulisan laporan ini, Bu Fariska Zakhralativa Ruskanda, S.T.,M.T. selaku dosen kelas kami pada Dasar Pemrograman, kak Felicia Gojali selaku asistensi. Penulis tentu menyadari bahwa laporan ini masih jauh dari kata sempurna dan masih terdapat banyak kekurangan. Untuk itu, penulis mengharapkan saran serta kritik dari pembaca. Apabila terdapat banyak kesalahan pada laporan ini penulis mohon maaf yang sebesar-besarnya.

Bandung, 25 April 2021 Kelompok 01/K03

# PERNYATAAN KELOMPOK

“*Saya Usman Zakaria, NIM 16520003. 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 konsekuensianya, yaitu mendapatkan nilai E pada mata kuliah IF1210 Dasar Pemrograman Semester 2 2020/2021”.*

“*Saya Muhammad Garebaldhie, NIM 16520273 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 konsekuensianya, yaitu mendapatkan nilai E pada mata kuliah IF1210 Dasar Pemrograman Semester 2 2020/2021”.*

“*Saya Yosafat Raditya , NIM 16520293. 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 konsekuensianya, yaitu mendapatkan nilai E pada mata kuliah IF1210 Dasar Pemrograman Semester 2 2020/2021”.*

“*Saya Nicholas Glenn , NIM 16520343. 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 konsekuensianya, yaitu mendapatkan nilai E pada mata kuliah IF1210 Dasar Pemrograman Semester 2 2020/2021”.*

**DAFTAR ISI**

[TUGAS BESAR 1](#_Toc70575682)

[PEMBUKAAN 1](#_Toc70575683)

[PERNYATAAN KELOMPOK 2](#_Toc70575684)

[2](#_Toc70575685)

[DAFTAR TABEL 7](#_Toc70575686)

[DAFTAR GAMBAR 8](#_Toc70575687)

[PEMBAHASAN 9](#_Toc70575688)

[A. Deskripsi Persoalan 9](#_Toc70575689)

[B. Daftar Pembagian Kerja Anggota Kelompok 10](#_Toc70575690)

[Tabel 01. Pembagian Kerja 10](#_Toc70575691)

[C. Checklist Hasil Rancangan, Implementasi, dan Testing 11](#_Toc70575692)

[Tabel 02. Checklist 11](#_Toc70575693)

[A. Desain Command 12](#_Toc70575694)

[2. login 12](#_Toc70575695)

[3. cariraryti 12](#_Toc70575696)

[4. caritahun 13](#_Toc70575697)

[5. tambahitem 13](#_Toc70575705)

[6. hapusitem 13](#_Toc70575712)

[7. ubahitem 13](#_Toc70575715)

[8. pinjam 13](#_Toc70575719)

[9. kembalikan 14](#_Toc70575723)

[10. minta 14](#_Toc70575727)

[11. riwayatpinjam 14](#_Toc70575731)

[12. riwayatkembali 14](#_Toc70575737)

[13. riwayatambil 14](#_Toc70575743)

[14. save 15](#_Toc70575749)

[15. exit 15](#_Toc70575751)

[B. Desain Kamus Data 15](#_Toc70575753)

[C. Desain Dekomposisi Algoritmik dan Fungsional Program 16](#_Toc70575754)

[D. Spesifikasi 17](#_Toc70575755)

[unit unit\_login\_register; 18](#_Toc70575756)

[unit unit\_login; 19](#_Toc70575757)

[unit unit\_pencarian; 20](#_Toc70575758)

[unit unit\_pencarian; 22](#_Toc70575759)

[unit unit\_pelaporan; 24](#_Toc70575760)

[unit unit\_pelaporan; 26](#_Toc70575761)

[unit unit\_pelaporan; 28](#_Toc70575762)

[unit unit\_pelaporan; 29](#_Toc70575763)

[unit unit\_penambahan\_buku; 30](#_Toc70575764)

[unit unit\_penambahan\_buku; 30](#_Toc70575765)

[unit unit\_pencarian; 31](#_Toc70575766)

[unit unit\_pencarian; 32](#_Toc70575767)

[unit unit\_load\_save\_exit; 35](#_Toc70575768)

[unit unit\_load\_save\_exit; 36](#_Toc70575769)

[unit unit\_pencarian; 37](#_Toc70575770)

[unit unit\_load\_save\_exit; 39](#_Toc70575771)

[E. Hasil Pengujian 41](#_Toc70575772)

[LAMPIRAN 46](#_Toc70575773)

# DAFTAR TABEL

**DAFTAR TABEL Halaman**

Tabel 01 : Pembagian Kerja… 7

Tabel 02 : Checklist… 9

Tabel 03 : Hasil Pengujian… 46

# DAFTAR GAMBAR

**DAFTAR GAMBAR Halaman**

Gambar 1-17 : Hasil Screenshot Pengujian… 46

# PEMBAHASAN

## 1. Deskripsi Persoalan



Seiring dengan modernisasi dan perkembangan zaman, seorang pelajar dituntut untuk bisa belajar dari berbagaimacam sumber. Sebagai seorang pelajar, Mobita ingin lulus dengan nilai yang memuaskan, Doraemonangis ingin membantunya dengan memberikan dia alat-alat/gadget, namun suatu hari Doraemonangis memiliki kerusakan sistem inventarisasi gadget. Dalam “Spesifikasi Tugas Besar IF1210 Dasar Pemrograman”, persoalan tersebut secara lebih rinci diterjemahkan menjadi kisah Mobita. Mahasiswa STEI sebagai calon *programmer* berniat untuk membantu Doraemonangis, untuk menyelesaikan persoalan tersebut.

Persoalan ini dapat diatasi dengan membuat sistem inventarisasi gadget yang dibuat dengan bahasa pemrograman *Python*. Fitur-fitur yang tersedia dalam sistem inventarisasi gadget tersebut antara lain: (1) Register (2) Login (3) Pencarian gadget berdasarkan rarity (4) Pencarian gadget berdasarkan tahun ditemukan (5) Menambah Item (6) Menghapus Gadget atau Consumable (7) Mengubah Jumlah Gadget atau Consumable pada Inventory (8) Meminjam Gadget (9) Mengembalikan gadget (10) Meminta Consumable (11) Melihat Riwayat Peminjaman Gadget (12) Melihat Riwayat Pengembalian Gadget (13) Melihat Riwayat Pengembalian Gadget (14) *Load* *Data* (15) *Save Data* (16) Help dan (17) Exit

1. F01 – Register

Pada bagian register, hanya admin yang bisa mengaksesnya, admin bisa mendaftarkan pengguna baru yang memiliki role user yang akan di masukan ke dalam database. Admin hanya bisa mendaftarkan role user namun tidak bisa mendaftarkan dengan role admin, untuk membuat role admin pengguna bisa mengedit csv files yang ada

1. F02 – Login

Pada bagian login akan diminta input username dan password, apabila username tidak sesuai dengan yang data base maka akan muncul input username Kembali, begitu pula dengan password. Pengguna dengan role user dan admin memiliki fitur yang berbeda.

1. F03 – Pencarian gadget berdasarkan rarity

Pada system inventarisasi, terdapat gadget yang memiliki rarity yang berbeda – beda (C, B, A, S). Input rarity sudah dipastikan valid sesuai dengan 4 jenis tersebut. Output dari spesifikasi ini berupa Gadget yang sesuai dengan rarity yang sebelumnya diinput

1. F04 – Pencarian gadget berdasarkan tahun

Pada program ini, pengguna menginput tahun dan operator, operator berupa comparison operator (< , ≤, =, > ≥) Output dari fungsi ini berupa gadget dengan tahun ditemukan sesuai dengan input tahun dan operator yang diberikan

1. F05 – Menambah item

Program ini hanya bisa dilakukan dengan pengguna yang memiliki role admin. Program dimulai dengan menerima input berupa ID item yang ingin ditambah, apabila awalan ID tersebut G, maka akan ditambahkan kedalam Gadget namun jika ID tersebut awalannya memiliki huruf C maka akan ditambahkan kedalam Consumables setelah menerima ID program akan meminta input berupa hal hal yang diperlukan sesuai dengan type id yang diinput. Fungsi tidak akan berjalan jika input ID tidak sesuai

1. F06 – Menghapus Gadget atau consumable

Program ini hanya bisa dijalankan dengan pengguna yang memiliki role admin. Program dimulai dengan menerima input berupa ID item yang ingin dihapus, apabila awalan ID berupa huruf G maka akan dihapus item dengan ID tersebut pada file gadget dan jika awalan ID berupa huruf C maka akan dihapus item dengan ID tersebut pada file consumables. Program tidak akan jalan jika ID yang diinput tidak valid

1. F07 – Mengubah jumlah gadget atau consumable pada inventory

Program ini hanya bisa dijalankan dengan pengguna yang memilki role admin. Program dimulai dengan menerima input berupa ID item yang ingin dirubah julahnya. Apabila awalan ID berupa huruf G maka akan dirubah jumlah item dengan ID tersebut pada file gadget dan apabila awalan ID berupa huruf C maka akan diubah jumlah item dengan ID tersebut pada file consumables. Program tidak berjalan jika ID yang diinput tidak valid

1. F08 – Meminjam Gadget

Program ini hanya bisa dijalankan dengan pengguna yang memiliki role user. Input yang diterima oleh program berupa ID Gadget yang akan dipinjam serta tanggal peminjaman. User tidak bisa meminjam barang yang sama Ketika barang yang dipinjam belum dikembalikan

1. F09 – Mengembalikan Gadget

Program ini hanya bisa dijalankan dengan pengguna yang memiliki role user. Setiap user memiliki entry peminjaman yang berbeda beda, setelah ditampilkan entry peminjaman maka program menerima input berupa nomor barang mana yang akan dikembalikan lalu program menerima input berupa tanggal pengemblaian

1. F10 - Meminta Consumables

Program ini hanya bisa dijalaankan dengan pengguna yang memiliki role user. Program akan meminta input berupa ID barang, tanggal dan jumlah barang yang diminta, apabila barang yang diminta lebih besar dibandingkan dengan jumlah barang yang dimiliki saat ini makan program tidak akan berjalan

1. F11 – Melihat Riwayat Peminjaman Gadget

Program ini hanya bisa dijalankan dengan pengguna yang memiliki role admin. Program akan menampilkan list entry peminjaman dari masing masing user entry yang ditampilkan hanya 5 saja, namun apabila entry lebih dari 5 maka program akan meminta input berupa apakah akan dilanjutkan menampilkan entry, jika ya maka ditampilkan 5 entry lagi jika tidak program berhenti. Program menampilkan entry yang sudah terurut berdasarkan tanggal.

1. F12 – Melihat Riwayat Pengembalian Gadget

Program ini hanya bisa dijalankan dengan pengguna yang memilki role admin. Program akan menampilkan list entry / Riwayat pengembalian yang ada dan yang ditampilkan hanya lima buah saja, apabila pengguna ingin melihat entry yang lain maka program akan meminta input 5 entry yang baru begitu pula selanjutnya. Program akan menampilakn entry yang sudah terurut berdasarkan tanggal.

1. F13 – Melihat Riwayat pengembalian consumable

Program ini hanya bisa dijalankan dengan pengguna yang memiliki role admin. Program akan meanmpilkan list entry / Riwayat pengembalian consumable yang telah dipinjam oleh user. Hanya 5 entry yang akan ditampilkan, lalu program akan meminta input berupa apakah akan dilanjutkan dengan menampilkan 5 entry selanjutnya dan begitu seterusnya. Entry yang ditampilkan berupa daftar yang sudah terurut berdasarkan tanggal yang paling baru

1. F14 – Load Data

Program ini berfungsi untuk merubah csv file yang ada kedalam array global yang sudah tersedia, input dari program ini berupa nama folder yang akan di load datanya

1. F15 – Save data

Program ini berfungsi untuk membuat csv dari array global yang ada di main program. Program menerima input berupa nama folder yang akan di save.

1. F16 – Help

Program ini berfungsi untuk melihat command atau perintah yang tersedia. Command yang tersedia sesuai dengan role yang dimiliki oleh pengguna.

1. F17 – Exit

Program ini akan memberhentikan main program dan akan menerima input berupa apakah pengguna akan menjalankan prosedur save atau tidak, apabila pengguna tidak ingin menjalankan save maka program berhenti namun jika pengguna ingin melaksanakan prosedur save maka program akan menjalankan prosedur save lalu akan memberhentikan program

1. FB02 – Mengembalikan Gadget Secara Parsial

Program ini merupakan tambahan dari F09, program ini befungsi untuk mengembalikan gadget namun tidak seutuhnya

## Daftar Pembagian Kerja Anggota Kelompok

Daftar pembagian kerja anggota kelompok 01 dari kelas 03 adalah sebagai berikut:

## Tabel 01. Pembagian Kerja

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fitur** | **Implementasi** | **NIM**  **Desainer** | **NIM**  **Coder** | **NIM**  **Tester** |
| F01-Register | **Procedure** register  **Procedure** makeList  **Function** checkFunction | 16520003 | 16520003 | 16520003 |
|  |  |  |  |  |
| F02-Login | **Procedure** login  **Procedure** readUser | 16520003 | 16520003 | 16520003 |
|  |  |  |  |  |
| F03-Pencarian gadget berdasarkan rarity | **Procedure** cariRarity  **Procedure** formatGadget  **Procedure** GadgetCsvParser  **Procedure** lineParserGadget | 16520343 | 16520343 | 16520343 |
| F04 – Pencarian Gadget berdasarkan tahun | **Procedure** cariTahun  **Procedure** formatGadget  **Procedure** GadgetCsvParser  **Procedure** lineParserGadget | 16520343 | 16520343 | 16520343 |
|  |  |  |  |  |
| F05-Menambah Item | **Procedure** tambahItem  **Procedure** tambahGadget  **Procedure** tambahConsumables | 16520343 | 16520343 | 16520343 |
|  |  |  |  |  |
| F06-Menghapus Gadget atau Consumable | **Procedure** hapusItem  **Procedure** hapusGadget  **Procedure**  hapusConsumables | 16520293 | 16520293 | 16520293 |
| F07- Mengubah Jumlah Gadget atau Consumable pada Inventory | **Procedure** ubahItem  **Procedure** ubahGadget  **Procedure** ubahConsumables | 16520293 | 16520293 | 16520293 |
| F08-Meminjam Gadget | **Procedure** pinjamGadget | 16520273 | 16520273 | 16520273 |
| F09-Mengembalikan gadget | **Procedure** mengembalikanGadget  **Function** returnIDGadget  **Function** returnIndexGadget | 16520273 | 16520273 | 16520273 |
| F10-Meminta Consumable | **Procedure** memintaConsumables | 16520273 | 16520273 | 16520273 |
|  |  |  |  |  |
| F11-Melihat Ri wayat Peminjaman Gadget | **Procedure** riwayatPinjamGadget  **Function**  returnIndexBorrowGadgetDate  **Procedure**  formatPinjamGadget | 16520273 | 16520273 | 16520273 |
| F12-Melihat Riwayat Pengembalian Gadget | **Procedure** riwayatReturnGadget  **Function**  returnIndexReturnGadgetDate  **procedure** formatReturnGadget | 16520273 | 16520273 | 16520273 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| F13-Melihat Riwayat Pengembalian Gadget | **Procedure** RiwayatAmbilConsumables  **Function** returnIndexAmbilConsumablesDate  **Procedure** formatReturnConsumables | 16520273 | 16520273 | 16520273 |
| F14-Load Data | **Procedure** foundFolder  **Procedure** findFolder  **Procedure** readUser  **Procedure** readGadget  **Procedure** readGadgetBorrow  **Procedure** readGadgetReturn  **Procedure** readConsumables  **Procedure** readConsumablesHistory | 16520273 | 16520273 | 16520273 |
| F15-Save Data | **Prosedure** save  **Procedure** toCsv  **Procedure** makeDir | 16520273 | 16520273 | 16520273 |
| F16-Help | **Procedure** help | 16520273 | 16520273 | 16520273 |
| F17-Exit | **Procedure** exit | 16520293 | 16520293 | 16520293 |

## Checklist Hasil Rancangan, Implementasi, dan Testing

*Checklist* hasil rancangan, implementasi, dan testing dari setiap primitif adalah sebagai berikut:

## Tabel 02. Checklist

|  |  |  |  |
| --- | --- | --- | --- |
| **Fitur** | **Desain** | **Implementasi** | **Testing** |
| F01-Register | ✓ | ✓ | ✓ |
| F02-Login | ✓ | ✓ | ✓ |
| F03-Pencarian gadget berdasarkan rarity | ✓ | ✓ | ✓ |
| F04-Pencarian gadget berdasarkan tahun ditemukan | ✓ | ✓ | ✓ |
| F05-Menambah item | ✓ | ✓ | ✓ |
| F06-Menghapus Gadget atau Consumable | ✓ | ✓ | ✓ |
| F07-Mengubah Jumlah Gadget atau Consumable pada Inventory | ✓ | ✓ | ✓ |
| F08-Meminjam Gadget | ✓ | ✓ | ✓ |
| F09-Mengembalikan gadget | ✓ | ✓ | ✓ |
| F10-Meminta Consumable | ✓ | ✓ | ✓ |
| F11-Melihat Riwayat Peminjaman Gadget | ✓ | ✓ | ✓ |
| F12-Melihat Riwayat Pengembalian Gadget | ✓ | ✓ | ✓ |
| F13-Melihat Riwayat Pengembalian Gadget | ✓ | ✓ | ✓ |
| F14-Load Data | ✓ | ✓ | ✓ |
| F15-Save Data | ✓ | ✓ | ✓ |
| F16-Help | ✓ | ✓ | ✓ |
| F17-Exit | ✓ | ✓ | ✓ |

## Desain Command

* 1. **Register**

input

Masukan nama: ….

Masukan username: ….

Masukan password: ….

Masukan alamat: ….

Output Valid

User {username} telah berhasil register kedalam kantong ajaib

Output Tidak Valid

Username sudah ada, silahkan gunakan username yang lain!

## Login

input

Masukan username: ….

Masukan password: ….

Output Valid

Halo {username}! Selamat dating di kantong ajaib

Output Tidak Valid

Username atau password salah!

## carirarity

input {input rarity sudah dipastikan valid}

Masukan Rarity: ….

Nama : ….

Deskripsi : ….

Jumlah : ….

Rarity : ….

Tahun Ditemukan :

Output Valid

Nama : {namaGadget}

Deskripsi : {deskripsiGadget}

Jumlah : {jumlahGadget}

Rarity : {rarityGadget}

Tahun Ditemukan ; {Tahun ditemukan}

## caritahun

## input {input kategori dan tahun pasti valid}

## Masukan Tahun: ….

## Masukan kategori: ….

## Nama : ….

## Deskripsi : ….

## Jumlah : ….

## Rarity : ….

## Tahun Ditemukan : ….

## Output Valid

Nama : {namaGadget}

Deskripsi : {deskripsiGadget}

Jumlah : {jumlahGadget}

Rarity : {rarityGadget}

Tahun Ditemukan ; {Tahun ditemukan}

## tambahitem

## input

## Masukan ID: ….

## Masukan Nama: ….

## Masukan Deskripsi: ….

## Masukan Jumlah: ….

## Masukan rarity: ….

## Masukan tahun ditemukan: ….

## Output Valid

## item telah berhasil ditambahkan

## Output Tidak Valid

## ID tidak valid {Ketika id tidak valid bukan G atau C}

## Gagal menambahkan item karena ID sudah ada {Ketika id sudah ada}

## Hanya admin yang bisa menambah item {Ketika id selain admin menjalankan program}

## hapusitem

## input

## Masukan ID item: ….

## Apakah anda yakin ingin menghapus Buku (Y/N)?

## Output Valid

## item telah berhasil dihapus dari database

## Output Tidak Valid

## id tidak valid {id tidak memiliki komponen G atau C} Tidak ada item dengan ID tersebut. {Ketika item tidak ada dengan ID}

## Hanya admin yang bisa menghapus item {Ketika id selain admin menjalankan program}

## Ubahitem

## input

## Masukan ID item: ….

## Masukan jumlah: ….

## Stok sekarang: ….

## Output Valid

## {jumlah} {namaItem} berhasil ditambahkan. Stok sekarang: {jumlahBaru}

## Output Tidak Valid

## Id tidak valid {Ketika ID tidak ada komponen G dan C}

## Tidak ada item dengan ID tersebut {Ketika item tidak ada dengan id tersebut}

## Hanya admin yang bisa mengubah item {Ketika id selain admin menjalankan program}

## Pinjam

## Input

## Masukan ID item: ….

## Tanggal peminjaman: ….

## Jumlah peminjaman: ….

## Output Valid

## Item {namaItem} (x{jumlah}) berhasil dipinjam!

## Output Tidak Valid

## Stok barang yang dipinjam kurang {Ketika barang yang dipinjam melebihi stok}

## Tidak ada item dengan ID tersebut {Ketika item tidak ada dengan id tersebut}

## Hanya user yang bisa meminjam gadget {Ketika id selain user menjalankan program}

## kembalikan

## input

## Masukan nomor peminjaman: ….

## Tanggal pengembalian: ….

## Masukan jumlah pengembalian: ….

## Output Valid

## Item {namaItem} (x{banyak item} telah dikembalikan)

## Output Tidak Valid

## Kembalikan item sesuai dengan jumlahnya {ketika user mengembalikan item melebih yang dipinjam}

## Tidak ada barang yang dipinjam {Ketika user tidak meminjam barang apapun}

## Nomor tidak ada {Ketika user memasukan nomor pinjam yang tidak ada}

## Hanya user yang bisa mengembalikan gadget {Ketika id selain user menjalankan program}

## minta

## input

## Masukan ID item: ….

## Jumlah: ….

## Tanggal permintaan: ….

## Output Valid

## Item {namaItem} (x{jumlah}) telah berhasil di ambil!

## Output Tidak Valid

## Tidak ada item dengan ID tersebut {Ketika id tidak ada pada daftar}

## Gagal diambil karena stok kurang! {Ketika jumlah lebih dari stok yang ada}

## riwayatpinjam

## Output Valid

## ID Peminjaman : {idPeminjaman}

## Nama Pengambil : {nama peminjam

## Nama Gadget : {deskripsiGadget}

## Tanggal Peminjaman : {tanggalPeminjaman}

## Jumlah : {jumlah}

## riwayatkembali

## Output Valid

## ID Pengembalian : {idPengembalian}

## Nama Pengambil : {namaPengambil}

## Nama Gadget : {namaGadget

## Tanggal Pengembalian : {tanggal}

## Jumlah peminjaman : {jumlah}

## riwayatambil

## Output Valid

## ID Pengambilan : {idPengambilan}

## Nama Pengambil : {namaPengambil}

## Nama Consumable : {namaConsumables}

## Tanggal Pengembalian : {tanggalPengembalian}

## Jumlah : {jumlah}

## Load

## input

## kantongajaib.py {namafolder}

## Output Valid

## Selamat datang di kantong ajaib!

## save

## input

## Masukan nama Folder penyimpanan: ….

## Output Valid

## saving…

## Data telah disimpan pada folder {foldername}

## help

## input

## output

## {apabila pengguna memiliki role admin}

## ========= HELP =========

## login - untuk melakukan login ke dalam sistem

## register - untuk melakukan registrasi user baru

## carirarity - untuk mencari gadget dengan rarity tertentu

## caritahun - melakukan pencarian gadget berdasarkan tahun ditemukan

## tambahitem - untuk melakukan penambahan item

## hapusitem - untuk menghapus suatu item pada database

## ubahitem - digunakan untuk mengubah jumlah gadget dan consumable

## riwayatpinjam - untuk melihat riwayat pinjam

## riwayatkembali - untuk melihat riwayat pengembalian

## riwayatambil - untuk melihat riwayat ambil

## help - untuk menampilkan list command

## save - untuk mengesave program

## exit - untuk keluar dari program

## {apabila pengguna memiliki role user}

## ========= HELP =========

## carirarity - untuk mencari gadget dengan rarity tertentu

## caritahun - melakukan pencarian gadget berdasarkan tahun ditemukan

## pinjam - digunakan untuk memnijam gadget

## kembalikan - untuk mengembalikan gadget

## minta - untuk meminta consumable yang tersedia

## help - untuk menampilkan list command

## save - untuk mengesave program

## exit - untuk keluar dari program

## exit

## input {prosedur exit hanya melakukan save pada folder yang telah ada}

## Apakah anda mau melakukan penyimpaanan file yang sudah diubah? (y/n)

## Output Valid

## saving…

## Data telah disimpan pada folder {foldername}

## Desain Kamus Data

* 1. F01-Register

|  |
| --- |
| **KAMUS**  id, nama, username, alamat, role : string  listUser : array of user  Type user : < id : string; username : string; nama : string; alamat : string; password : string; role : string >  function makeList (path : string) → array of username |

* 1. F02-Login

|  |
| --- |
| **KAMUS**  I, banyakUser : integer  Username, password, checkPassword, idUser, roleUser : string  Type user : < id : string; username : string; nama : string; alamat : string; password : string; role : string >  listUser : array of user  function readUser(path : string) → array of user |

* 1. F03-Pencarian gadget berdasarkan rarity

|  |
| --- |
| **KAMUS**  i, banyak\_gadget : integer  rarity : string  array\_gadget : array\_of\_gadget |

* 1. F04-Pencarian gadget berdasarkan tahun

|  |
| --- |
| **KAMUS**  i, tahun, banyak\_gadget : integer  Kategori : string  Found : Boolean  Array\_gadget : array\_of\_gadget |

* 1. F05-Menambah Item

|  |
| --- |
| **KAMUS**  type Gadget: < id : string; nama : string; deskripsi : string; jumlah : integer; rarity : string; tahun\_ditemukan : integer >  type Consumable: < id : string; deskripsi : string; jumlah : integer; rarity : string >  procedure tambahGadget(input : array of gadget, input : integer) {menambahkan dictionary of gadget kedalam array global}  procedure tambahConsumables(input : array of consumable, input :  integer)  {menambahkan dictionary of consumable kedalam array global}  Id : string |

* 1. F06-Menghapus Gadget atau Consumable

|  |
| --- |
| **KAMUS**  **Function** append  { Fungsi bawaan Python untuk menambahkan data ke list }  { I.S. List yang sudah terisi maupun kosong }  { F.S. List ditambahkan data baru }  **Function** remove  { Fungsi bawaan Python untuk menghilangkan data dari list }  { I.S. List yang sudah terisi maupun kosong }  { F.S. Data yang dipilih dihilangkan dari list }  **Function** upper  { Fungsi bawaan Python untuk mengkapitalkan string }  { I.S. String yang berhuruf kecil maupun kapital }  { F.S. String yang berhuruf kecil akan berubah menjadi kapital dan yang kapital tetap kapital }  listID : array[]  found : **boolean** = **False** |

* 1. F07- Mengubah Jumlah Gadget atau Consumable pada Inventory

|  |
| --- |
| **KAMUS LOKAL**  **Function** append  { Fungsi bawaan Python untuk menambahkan data ke list }  { I.S. List yang sudah terisi maupun kosong }  { F.S. List ditambahkan data baru }  found : boolean = False  listID : array[]  Jumlah : integer |

* 1. F08-Meminjam Gadget

|  |
| --- |
| **KAMUS LOKAL**  i, banyak\_gadget,banyak\_pinjam, banyak\_id, Jumlah : integer  found, flag, canContinue :Boolean  isPinjam, isReturned, idPinjam, id, idUser, tanggal : string  array\_gadget : array\_of\_gadget  array\_pinjam\_gadget : array\_of\_pinjam\_gadget  array\_id : array\_of |

* 1. F09-Mengembalikan gadget

|  |
| --- |
| **KAMUS LOKAL**  newId, Id, roleUser, tanggalPermintaan: string  Found : boolean  listId : array of string  I, nomor pinjam, jumlah sementara, tanggal pengembalian,jumlah pengembalian: integer  tempItem : consumableReturn  **type** Gadget**:** <id : string; deskripsi : string; jumlah : integer; rarity : string >  **type** GadgetReturn : < id : string, id\_peminjam : string ; tanggal\_pengambilan : string; jumlah peminjaman'  : integer >  Gadget : array of Gadget  ReturnGadgetHistory : array of ReturnGadget |

* 1. F10-Meminta Consumables

|  |
| --- |
| **KAMUS LOKAL**  newId, Id, roleUser, tanggalPermintaan: string  Found : boolean  listId : array of string  I, jumlah, banyakConsuables, banyakConsumableHistory, banyakId : integer  tempItem : consumableReturn  **type** Consumable**:** <id : string; deskripsi : string; jumlah : integer; rarity : string >  **type** ConsumableReturn : < id : string, id\_pengambil : string; id\_consumable : string; tanggal\_pengambilan : string; jumlah : integer >  consumables : array of consumable  consumablesHistory : array of consumableReturn |

* 1. F11-Melihat Riwayat Peminjaman Gadget

|  |
| --- |
| **KAMUS LOKAL**  Index : array of integer  banyakIndex : length(index)  idPeminjam, namaPengambil, idGadget, idPinjam, namaGadget, tanggal, jumlah, tempAll: array of string  I, banyakPinjam, banyakUser, banyakIdGadget, banyakGadget : integer  listTempAll : array of array of string  procedure returnIndexBorrowGadgetDate(input listBorrowGadget: array of borrow gadget) {mengembalikan index yang sudah di sort berdasarkan tanggal}  procedure sortDateTimeGadget(input list : array of string)  {melakukan sort berdasarkan tanggal}  Procedure formatPinjamGadget(input list : array of array of string)  {mengoutput sesuai dengan format yang diberikan}  type user : < id : string; username : string; nama : string; alamat : string; password : string; role : string >  typeGadget**:** <id : string; nama : string; deskripsi : string; jumlah : integer; rarity : string; tahun\_ditemukan : integer >  type GadgetBorrow : < id : string; id\_peminjam : string; id\_gadget : string; tanggal\_peminjaman : string; jumlah : string; is\_returned : boolean > |

* 1. F12-Melihat Riwayat Pengembalian Gadget

|  |
| --- |
| **KAMUS LOKAL**  array\_tempAll : array [] of string  array\_temp : array [] of string  array\_idpeminjaman : array [] of integer  array\_namapengambil : array [] of string  array\_idgadget : array [] of string  array\_idpengembalian : array [] of integer  array\_namagadget : array [] of string  array\_tanggal : array [] of string  array\_jumlah : array [] of integer  array\_idUser ; array [] of integer  i, j : integer  banyak\_array\_borror\_gadget : integer  banyak\_array\_user : integer  banyak\_array\_gadget : integer  banyak\_index : integer  banyak\_array\_idpeminjaman : integer  banyak\_array\_idpengembalian : integer  banyak\_array\_idUser : integer  banyak\_array\_idgadget : integer  {proses-proses}  **function** returnIndexreturnGadgetDate  **function** formatReturnGadget |

* 1. F13-Melihat Riwayat pengambilan Gadget

|  |
| --- |
| **KAMUS LOKAL**  **Function** append  { Fungsi bawaan Python untuk menambahkan data ke list }  { I.S. List yang sudah terisi maupun kosong }  { F.S. List ditambahkan data baru }  listTempAll : array[] |

* 1. F14-Load Data

|  |
| --- |
| **ALGORITMA**  Import os  Import argparse  Parser <- argparse.ArgumentParser(description <- ‘Masukan data yang akan di load’)  Parser.add\_argument(‘folder’, type <- str)  Args <- parser.parse\_args() |

* 1. F15-Save Data

|  |
| --- |
| **KAMUS LOKAL**  {listFile sudah terdefinisi sebelumnya}  {modul os sudah terdefinisi}  folderName : string  listFile : array of filename  Found : Boolean  I : integer  banyakFile : len(listFile)  banyakArray : len(list)  **Procedure** checkPath(input foldername : string) {mengecek apakah sudah ada filenya apa belum }  **Procedure** makeDir(input foldername : string) {membuat folder baru bila belum ada dan mengoverwrite file jika sudah ada}  **Procedure** toCSV(input path: string; input list: array of dictionary)  {memparse array of dictionary baris per baris kedalam bentuk csv} |

* 1. F16-Help
  2. F17-Exit

|  |
| --- |
| **KAMUS LOKAL**  Filename, list, folderName, text = string |

## Desain Dekomposisi Algoritmik dan Fungsional Program

## Program Utama

|  |
| --- |
| **Program** kantongajaib  {spesifikasi : program menerima input dari pengguna dan mengolah data sesuai dengan hasil input pengguna } |
| **KAMUS LOKAL**  listfile : array [‘user.csv’, ‘gadget.csv’, ‘gadget\_return\_history.csv’, ‘gadget\_borrow\_history.csv’,’consuamble.csv’,’consumable\_history.csv’]  zipfile : array [] of string  Userfile : integer  roleUser : string  idUser : integer  text : string  {proses-proses}  **from** Load **import** \*  **from** Register **import** \*  **from** Help **import** \*  **from** Login **import** \*  **from** Exit **import** \*  **from** Save **import** \*  **from** Gadget **import** \*  **from** consumables **import** \*  **import** sys |
| **ALGORITMA**  user <- buat\_array\_user  gadget <- buat\_array\_gadget  consumables <- buat\_array\_consumables  gadgetBorrow <- buat\_array\_gadgetBorrow  gadgetReturn <- buat\_array\_gadgetReturn  consumablesHistory <- buat\_array\_consumableHistory  output()  output(‘Loading…’)  output()  Folder <- foundFolder()  UserFile <- ‘’  idUser <- ‘’  roleUser <- ‘’  if Folder then  Folder <- findFolder()  user <- readUser(Folder)  gadget <- readGadget(Folder)  gadgetBorrow <- readGadgetBorrow(Folder)  gadgetReturn <- readGadgerReturn(Folder)  consumables <- readConsumables(Folder)  consumablesHistory <- readConsumablesHistory(Folder)  else  output(‘Tidak ada nama folder yang ditemukan!’)  output(‘Usage: python kantongajaib.py <nama\_folder>’)  sys.exit()  zipFile <- [user, gadget, gadgetReturn, gadgetBorrow, consumables, consumableHistory]  while **True** do  output()  text <- input().lower()  output()  depend on (text)  text == ‘login’ :  userFile <- login(Folder)  idUser <- str(userFile[0])  roleUser <- userFile[1]  text == ‘register’ :  if roleUser == ‘admin’ then  register(user)  else  output(‘Hanya admin yang bisa register’)    text == ‘carirarity’ :  searchRarity <- input()  output()  cariRarity(gadget, searchRarity)  text == ‘caritahun’ :  tahun <- input()  kategori <- input  ouput()  cariTahun(tahun, kategori, gadget)  text == ‘tambahitem’ :  if roleUser == ‘admin’ then  id <- input()  depend on id[0]  id[0] == ‘G’ :  tambahItem(gadget, id)  id[0] == ‘C’ :  tambahItem(consumables, id)  else  output(‘Hanya Admin yang bisa menambah item’)    text == ‘hapusitem’ :  if roleUser == ‘admin’ then  id <- input()  depend on id[0]  id[0] == ‘G’ :  hapusItem(gadget, id)  id[0] == ‘C’ :  hapusItem(consumables, id)  else  output(‘Hanya Admin yang bisa manghapus item’)    text == ‘ubahitem’ :  if roleUser == ‘admin’ then  id <- input()  depend on id[0]  id[0] == ‘G’ :  ubahItem(gadget, id)  id[0] == ‘C’ :  ubahItem(consumables, id)  else  output(‘Hanya Admin yang bisa mengubah item’)  text == ‘pinjam’ :  if roleUser == ‘user’ then  id <- input()  pinjamGadget(gadget, gadgetBorrow, id, idUser)  else  output(‘Hanya user yang bisa meminjam gadget’)    text == ‘kembalikan’ :  if roleUser == ‘user’ then  mengembalikanGadget(gadget, gadgetBorrow, gadgetReturn, idUser)  else  output(‘Hanya user yang bisa meminjam gadget’)  text == ‘minta’ :  if roleUser == ‘user’ then  id <- input()  memintaConsumables(consumables, consumableHistory, id, idUser)  else  output(‘hanya user yang bisa meminjam gadget’)  text == ‘riwayatpinjam’ :  if roleUser == ‘admin’ then  RiwayatPinjamGadget(gadgetBorrow, user, gadget)  else  output(‘Hanya admin yang bisa melihat riwayat pinjam’)  text == ‘riwayatkembali’ :  if roleUser == ‘admin’ then  RiwayatReturnGadget(gadgetReturn, gadgetBorrow, user, gadget)  else  output(‘Hanya admin yang bisa melihat riwayat kembali’)  text == ‘riwayatambil’ :  if roleUser == ‘admin’ then  RiwayatAmbilConsumables(consumableHistory, user, conasumables)  else  output(‘Hanya admin yang bisa melihat Riwayat ambil’)  text == ‘help’ :  if roleUser == ‘admin’ then  help(1)  else  help(2)  text == ‘save’ :  folderName <- input()  makeDir(folderName)  Csv, CsvContent traversal zip(listFile, zipFile)  Ssve(Csv, CsvContent, folderName)  output(‘Saving…’)  output(‘Data telah disimpan pada folder {folderName}!)  text == ‘exit’ :  output(‘Apakah anda mau melakukan penyimpanan file yang sudah diubah? (y/n)’)    if text == ‘n’ then  sys.exit()    folderName <- returnFolderName()  makeDir(folderName)    Csv, CsvContent traversal zip(listFile, zipFile)  exit(Csv, CsvContent, FolderName, text)    sys.exit() |

## Spesifikasi

* 1. F01-Register

## unit unit\_login\_register;

|  |
| --- |
| **Procedure** register (input/output folder : input)  {spesifikasi : procedure yang memasukkan data registrasi yang dilakukan oleh admin terhadap pengunjung ke dalam user.csv} |
| **KAMUS LOKAL**  id, nama, username, alamat, role : string  listUser : array of user  Type user : < id : string; username : string; nama : string; alamat : string; password : string; role : string >  function makeList (path : string) → array of username |
| **ALGORITMA**  listUsername <- makeList(listUser, "username")  input(nama)  input(username)  while checkFunction(listUsername, username) do  output()  output("Username sudah ada, silahkan gunakan username yang lain!")  input(username)  input(password )  input(alamat)  id <- str(len(listUser))  role <- "user"  tempDict <- {  'id' : id,  'username' : username,  'nama' : nama,  'alamat' : alamat,  'password' : password,  'role' : role  }  output ()  output(f"User {username} telah berhasil register kedalam Kantong Ajaib.")  listUser.append(tempDict)  -> listUser  {content = lineParser(id, username, nama, alamat, password, role}  {appendFile(filename, content} |

* 1. F02-Login

## unit unit\_login;

|  |
| --- |
| **Procedure** login (input/output folder : input)  {spesifikasi : procedure menerima input tulisan berupa login, username, dan password} |
| **KAMUS LOKAL**  I, banyakUser : integer  Username, password, checkPassword, idUser, roleUser : string  Type user : < id : string; username : string; nama : string; alamat : string; password : string; role : string >  listUser : array of user  function readUser(path : string) → array of user |
| **ALGORITMA**  listUser <- readUser {mengubah file csv menjadi array of user}  banyakUser <- length(listUser)  Input(username)  Input(password)  i <- 1  traversal i [1..banyakUser]  valueUsername <- listUser[i].username  if username = valueUsername then  checkPassword <- listUser[i].password  idUser <- ‘i'  roleUser <- listUser[i].role  if checkPassword = password then  output(“\n”)  output(“Halo {username}! Selamat datang di kantong ajaib”)  else:  output(“Username atau password salah!”)  output(“\n”)    while checkPassword != password do  input(username)  input(password)    traversal i [1..banyakUser]  if username = valueUsername then  checkPassword <- listUser[i].password  idUser <- ‘i'  roleUser <- listUser[i].role  if checkPassword != password then  output(“Username atau password salah!”)  output(“\n”)  output(“Halo {username}! Selamat dating di kantong ajiab”)  → (idUser, roleUser) |

* 1. F03-Pencarian gadget berdasarkan rarity

|  |
| --- |
| **Procedure** carirarity  spesifikasi : prosedur menerima input berupa rarity dan akan mencari gadget yang memiliki rarity yang diterima} |
| **KAMUS LOKAL**  i, banyak\_gadget : integer  rarity : string  array\_gadget : array\_of\_gadget |
| **ALGORITMA**  array\_gadget <- buat\_array\_gadget(tabel); {membuat array gadget dari table}  banyak\_gadget <- length(array\_gadget); {menentukan banyak gadget}  input(rarity)  i <- 1  traversal i [banyak\_gadget]  if array\_gadget[i].rarity == rarity then  output(array\_gadget[i].nama)  output(array\_gadget[i].deskripsi)  output(array\_gadget[i].jumlah)  output(array\_gadget[i].rarity)  output(array\_gadget[i].tahun\_ditemukan) |

## unit unit\_pencarian

* 1. F04-Pencarian gadget berdasarkan tahun

## unit unit\_pencarian

|  |
| --- |
| **procedure** caritahun  {spesifikasi : prosedur menerima input berupa tahun dan kategori dan akan melakukan pencarian gadget berdasarkan tahun yang dimasukkan, lalu akan menampilkannya ke layar} |
| **KAMUS LOKAL**  i, tahun, banyak\_gadget : integer  Kategori : string  Found : Boolean  Array\_gadget : array\_of\_gadget |
| **ALGORITMA**  array\_gadget <- buat\_array\_gadget(tabel); {membuat array gadget dari table}  banyak\_gadget <- length(array\_gadget); {menentukan banyak gadget}  key\_tahun <- array\_gadget[i].tahun\_ditemukan  valueTahun <- keyTahun  found <- false  input(tahun)  input(kategori)  i <- 1  traversal i [1, banyak\_gadget]  depend on (kategori)  kategori == ‘=’  if valueTahun == (tahun) then  found <- True  output(array\_gadget[i].nama)  output(array\_gadget[i].deskripsi)  output(array\_gadget[i].jumlah)  output(array\_gadget[i].rarity)  output(array\_gadget[i].tahun\_ditemukan)  kategori == ‘>’  if valueTahun > (tahun) then  found <- True  output(array\_gadget[i].nama)  output(array\_gadget[i].deskripsi)  output(array\_gadget[i].jumlah)  output(array\_gadget[i].rarity)  output(array\_gadget[i].tahun\_ditemukan)  kategori == ‘>’  if valueTahun >= (tahun) then  found <- True  output(array\_gadget[i].nama)  output(array\_gadget[i].deskripsi)  output(array\_gadget[i].jumlah)  output(array\_gadget[i].rarity)  output(array\_gadget[i].tahun\_ditemukan)  kategori == ‘<’  if valueTahun < (tahun) then  found <- True  output(array\_gadget[i].nama)  output(array\_gadget[i].deskripsi)  output(array\_gadget[i].jumlah)  output(array\_gadget[i].rarity)  output(array\_gadget[i].tahun\_ditemukan)  kategori == ‘<=’  if valueTahun <= (tahun) then  found <- True  output(array\_gadget[i].nama)  output(array\_gadget[i].deskripsi)  output(array\_gadget[i].jumlah)  output(array\_gadget[i].rarity)  output(array\_gadget[i].tahun\_ditemukan)  if not found then  output(‘Tidak ada gadget yang ditemukan’) |

* 1. F05-Menambah Item

## unit unit\_tambahItem

|  |
| --- |
| **Procedure** tambahItem(input list: array of gadget or consumables, input id : string)  {Admin dapat menambahkan item ke inventori, input berupa id item yang akan ditambahkan, nama item, deskripsi item, jumlah, rarity, tahun ditemukan} |
| **KAMUS LOKAL**  type Gadget: < id : string; nama : string; deskripsi : string; jumlah : integer; rarity : string; tahun\_ditemukan : integer >  type Consumable: < id : string; deskripsi : string; jumlah : integer; rarity : string >  procedure tambahGadget(input : array of gadget, input : integer) {menambahkan dictionary of gadget kedalam array global}  procedure tambahConsumables(input : array of consumable, input :  integer)  {menambahkan dictionary of consumable kedalam array global}  Id : string |
| **ALGORITMA**  {gadget dan consumables sudah terdefinisi saat di load}  Input(id)  depend on (id):  id¬0 = ‘G’  tambahGadget(gadget, id)  id0 = ‘C’  tambahConsumables(consumables, id)  else  output(“Gagal menghapus item karena Id tidak valid”) |

* 1. F06-Menghapus Gadget atau Consumable

## unit unit\_hapusGadget

|  |
| --- |
| **procedure** hapusGadget  {Admin dapat menghapus Gadget atau Consumable dalam melalui perintah} |
| **KAMUS LOKAL**  **Function** append  { Fungsi bawaan Python untuk menambahkan data ke list }  { I.S. List yang sudah terisi maupun kosong }  { F.S. List ditambahkan data baru }  **Function** remove  { Fungsi bawaan Python untuk menghilangkan data dari list }  { I.S. List yang sudah terisi maupun kosong }  { F.S. Data yang dipilih dihilangkan dari list }  **Function** upper  { Fungsi bawaan Python untuk mengkapitalkan string }  { I.S. String yang berhuruf kecil maupun kapital }  { F.S. String yang berhuruf kecil akan berubah menjadi kapital dan yang kapital tetap kapital }  listID : array[]  found : **boolean** = **False** |
| **ALGORITMA**  traversal i (len(listGadget))  listID.**append**(listGadget[i]['id'])  traversal i(len(listID))  if id = listID[i] then  found 🡨 **True**  output(f"Apakah anda yakin ingin menghapus {listGadget[i]['nama']} (Y/N)?)  text 🡨 input().**upper**()  if text = 'Y' then  listGadget.**remove**(listGadget[i])  output()  output("Item telah berhasil dihapus dari database.")  else  pass  if not found then  output("Tidak ada item dengan ID tersebut.") |

* 1. F07-Mengubah Jumlah Gadget atau Consumable pada Inventory

## unit unit\_ubahGadget

|  |
| --- |
| **procedure** ubahGadget (input : jumlah (string) , output : jumlah (string)  {Spesifikasi: Admin dapat mengubahan jumlah Gedget atau Consumable pada inventory melalui procedure ini .} |
| **KAMUS LOKAL**  **Function** append  { Fungsi bawaan Python untuk menambahkan data ke list }  { I.S. List yang sudah terisi maupun kosong }  { F.S. List ditambahkan data baru }  found : boolean = False  listID : array[]  Jumlah : integer |
| **ALGORITMA**  traversal i (len(listGadget))  listID.append(listGadget[i]['id'])    traversal i(1, len(listID))  if id = listID[i] then  found <- **True**  input(Jumlah)  if Jumlah + int(listGadget[i]['jumlah']) >= 0 then  listGadget[i]['jumlah'] 🡨 str(int(listGadget[i]['jumlah']) + Jumlah)  if Jumlah >= 0 then  output(f"{Jumlah} {listGadget[i]['nama']} berhasil ditambahkan. Stok sekarang: {listGadget[i]['jumlah']}")  else  output(f"{Jumlah} {listGadget[i]['nama']} berhasil dibuang Stok sekarang: {listGadget[i]['jumlah']}")  **break**  else  output(f"{-Jumlah}  {listGadget[i]['nama']} gagal dibuang karena stok kurang. Stok sekarang: {listGadget[i]['jumlah']} (< {-Jumlah})")  if not found then  output("Tidak ada item dengan ID tersebut.") |

* 1. F08-Meminjam Gadget

## unit unit\_pinjamGadget

|  |
| --- |
| **procedure** pinjamGadget  {spesifikasi: prosedur menerima input berupa ID gadget dan IDUser dan akan melakukan peminjaman gadget kepada User} |
| **KAMUS LOKAL**  i, banyak\_gadget,banyak\_pinjam, banyak\_id, Jumlah : integer  found, flag, canContinue :Boolean  isPinjam, isReturned, idPinjam, id, idUser, tanggal : string  array\_gadget : array\_of\_gadget  array\_pinjam\_gadget : array\_of\_pinjam\_gadget  array\_id : array\_of |
| **ALGORITMA**  array\_gadget <- buat\_array\_gadget  banyak\_gadget <- length(array\_gadget)  array\_pinjam\_gadget <- buat\_array\_pinjam\_gadget  banyak\_pinjam <- length(array\_pinjam\_gadget)  array\_id <- buat\_array\_id  banyak\_id <- length(array\_id)  found <- False  flag <- False  idPinjam <- str(length(list\_pinjam\_gadget))  isPinjam <- ‘False’  canContinue <- False  isReturned <- ‘’  input(id)  input(idUser)  i <- 1  traversal i [banyak\_gadget]  array\_id <- array\_gadget[i].id  i <- 1  traversal i [banyak\_pinjam]  if array\_pinjam\_gadget[i].id\_gadget == id then  if array\_pinjam\_gadget[i].id\_peminjam == idUser then  flag <- True  depend on (array\_pinjam\_gadget[i].is\_returned)  array\_pinjam\_gadget[i].is\_returned == ‘False’  isReturned <- False  array\_pinjam\_gadget[i].is\_returned == ‘True’  isReturned <- True  depend on (isReturned)  isReturned == True  canContinue <- True  isReturned == False  canContinue <- False  isReturned == ‘’  canContinue <- True  break  if not flag then  canContinue <- True  if canContinue then  i <- 1  traversal i [1, banyak\_id]  if array\_id[i] == id then  found <- True  input(‘tanggal’)  input(‘Jumlah’)  if Jumlah > array\_gadget[i].jumlah then  output(‘Stok barang yang dipinjam kurang!’)  break  else  array\_gadget[i].jumlah <- array\_gadget[i].jumlah – Jumlah  output(‘’)  output(‘Item’, array\_gadget[i].nama,’x’,jumlah,’berhasil dipinjam!’)  dict <- {  ‘id’ : idPinjam,  ‘id\_peminjam’ : idUser,  ‘id\_gadget’ : id,  ‘tanggal\_peminjaman’ : tanggal  ‘jumlah’ : str(Jumlah)  ‘is\_returned’ : isPinjam  }  array\_pinjam\_gadget <- dict  if not found then  output(‘Tidak ada item dengan ID tersebut’)  else  output(‘Tidak bisa meminjam, kembalikan dulu barang-nya!) |

* 1. F09-Mengambalikan Gadget

## unit unit\_Mengembalikan \_Gadget

|  |
| --- |
| **procedure** mengembalikanGadget (input Gadget: Gadget, input BorrowGadget : array of BorrowGadgetHistory, input Return Gadget: array of ReturnGadgetHistory, input : idUser)  **{s**pesifikasi : program mengembalikan data kedalam gadget.csv**}** |
| **KAMUS LOKAL**  newId, Id, roleUser, tanggalPermintaan: string  Found : boolean  listId : array of string  I, nomor pinjam, jumlah sementara, tanggal pengembalian,jumlah pengembalian: integer  tempItem : consumableReturn  **type** Gadget**:** <id : string; deskripsi : string; jumlah : integer; rarity : string >  **type** GadgetReturn : < id : string, id\_peminjam : string ; tanggal\_pengambilan : string; jumlah peminjaman'  : integer >  Gadget : array of Gadget  ReturnGadgetHistory : array of ReturnGadget |
| **ALGORITMA**  temp <- returnIDGadget(listBorrowGadget, idUser)  idGadget <- temp[0]  listIndexBorrowGadget <- temp[1]  listIndexGadget <-  []    traversal i  (len(idGadget)):  index <- returnIndexGadget(listGadget, idGadget[i])  listIndexGadget.append(index)  output(f"{i+1}. {listGadget[index]['nama']}")  output ()  nomorPinjam <- **int**(input("Masukan nomor peminjaman: "))  jumlahSementara <- listBorrowGadget[int(listIndexBorrowGadget[nomorPinjam - 1])]['jumlah']  tanggalPengembalian <- input("Tanggal pengembalian: ")  jumlahPengembalian <- **int**(input("Masukan jumlah pengembalian: "))  if jumlahPengembalian <= **int**(jumlahSementara) then  listGadget[int(listIndexGadget[nomorPinjam - 1])]['jumlah'] <- str(int(listGadget[int(listIndexGadget[nomorPinjam - 1])]['jumlah']) + jumlahPengembalian)  listBorrowGadget[listIndexBorrowGadget[nomorPinjam - 1]]['jumlah'] <- str(int(listBorrowGadget[listIndexBorrowGadget[nomorPinjam - 1]]['jumlah']) - jumlahPengembalian)  if **int**(jumlahSementara) - jumlahPengembalian == 0 then  listBorrowGadget[listIndexBorrowGadget[nomorPinjam - 1]]['is\_returned'] <- **'True'**  id <- len(listReturnGadget)  idPeminjaman <- listBorrowGadget[int(listIndexBorrowGadget[nomorPinjam - 1])]['id']  tempDict <- {  'id' : str(id),  'id\_peminjaman' : idPeminjaman,  'tanggal\_pengembalian' : tanggalPengembalian,  'jumlah\_peminjaman' : str(jumlahPengembalian)  }    listReturnGadget.**append**(tempDict)  else  output("Kembalikan item sesuai jumlahnya!") |

* 1. F10-Meminta Consumable

## unit unit\_meminta Consumable

|  |
| --- |
| **Procedure** memintaConsomables (input consumbles: array of consumables, input consumableHistory : array of consumablesHistory, input id : string, input idUser : string) |

|  |
| --- |
| **KAMUS LOKAL**  newId, Id, roleUser, tanggalPermintaan: string  Found : boolean  listId : array of string  I, jumlah, banyakConsuables, banyakConsumableHistory, banyakId : integer  tempItem : consumableReturn  **type** Consumable**:** <id : string; deskripsi : string; jumlah : integer; rarity : string >  **type** ConsumableReturn : < id : string, id\_pengambil : string; id\_consumable : string; tanggal\_pengambilan : string; jumlah : integer >  consumables : array of consumable  consumablesHistory : array of consumableReturn |
| **ALGORITMA**  banyakId <- length(listId)  banyakConsumables <- length(array of consumable)  banyakConsumableHIstory <- length(array of consumableReturn)  input(id)  found <- **false**  traversal I [0…banyakConsumables]  listId[i] <- consumables[i].id  traversal I [0 … banyakId]  if id = listId[i] then  found <- **true**    input(jumlah)  if jumlah > Consumables[i].jumlah then  output(“Gagal diambil karena stok kurang!”)  else  input(tanggalPermintaan)  output(“item {Consumables[i].nama} (x {jumlah}) telah berhasil diambil)  consumables[i].jumlah <- consumables[i].jumlah – jumlah  newId <- length(consumableReturn)    tempItem.id <- newId  tmpeItem.id\_pengambil <- idUser  tempItem.id\_consumable <- id  tempItem.tanggal\_peminjaman <- tanggalPermintaan  tempItem.jumlah <- jumlah  consumableHistory <- tempItem  if not found then  output(“Tidak ada item dengan ID tersebut.”) |

* 1. F11-Melihat Riwayat Peminjaman Gadget

## unit unit\_Riwayat Peminjaman Gadget

|  |
| --- |
| **Procedure** riwayatPinjamGadget (input gadgetBorrow : array of GadgetBorrow, input user : array of user, input gadget: array of Gadget) |
| **KAMUS LOKAL**  Index : array of integer  banyakIndex : length(index)  idPeminjam, namaPengambil, idGadget, idPinjam, namaGadget, tanggal, jumlah, tempAll: array of string  I, banyakPinjam, banyakUser, banyakIdGadget, banyakGadget : integer  listTempAll : array of array of string  procedure returnIndexBorrowGadgetDate(input listBorrowGadget: array of borrow gadget) {mengembalikan index yang sudah di sort berdasarkan tanggal}  procedure sortDateTimeGadget(input list : array of string)  {melakukan sort berdasarkan tanggal}  Procedure formatPinjamGadget(input list : array of array of string)  {mengoutput sesuai dengan format yang diberikan}  type user : < id : string; username : string; nama : string; alamat : string; password : string; role : string >  typeGadget**:** <id : string; nama : string; deskripsi : string; jumlah : integer; rarity : string; tahun\_ditemukan : integer >  type GadgetBorrow : < id : string; id\_peminjam : string; id\_gadget : string; tanggal\_peminjaman : string; jumlah : string; is\_returned : boolean > |
| **ALGORITMA**  banyakPinjam <- length(idpeminjam)  banyakUser <- length(listUser)  banyakIdGadget <- length(idGadget)  banyakGadget <- length(listGadget)  Index <- returnIndexBorrowGadgetDate(gadgetBorrow)  traversal I [0 … banyakIndex]  idPeminjam[i] <- gadgetBorrow[index[i]].id\_peminjam  idGadget[i] <- gadgetBorrow[index[i]].id\_gadget  idPinjam[i] <- gadgetBorrow[index[i]].id  tanggal[i] <- gadgetBorrow[index[i]].tanggal\_peminjaman  jumlah[i] <- gadgetBorrow[index[i]].jumlah  traversal I [0 … banyakPinjam]  traversal j [1 … banyakUser]  If idPeminjam[i] = user[j].id then  namaPengambil[i] <- user[j].nama  traversal I [0 … banyakIdGadget]  traversal j [1 … banyakGadget]  If idGadget[i] = gadget[j].id then  namaGadget[i] <- gadget[j].nama  traversal I [0 … banyakPinjam]  tempAll <- []  tempAll.append(idPinjam[i])  tempAll.append(namaPengambil[i])  tempAll.append(namaGadget[i])  tempAll.append(tanggal[i])  tempAll.append(jumlah[i])  listTempAll.append(tempAll)    formatPinjamGadget(listTempAll |

* 1. F12-Melihat Riwayat Pengembalian Gadget

## unit unit\_RiwayatReturnGadget

|  |
| --- |
| **procedure** RiwayatReturnGadget  {prosedur menerima input perintah kembalian gadget} |
| **KAMUS LOKAL**  array\_tempAll : array [] of string  array\_temp : array [] of string  array\_idpeminjaman : array [] of integer  array\_namapengambil : array [] of string  array\_idgadget : array [] of string  array\_idpengembalian : array [] of integer  array\_namagadget : array [] of string  array\_tanggal : array [] of string  array\_jumlah : array [] of integer  array\_idUser ; array [] of integer  i, j : integer  banyak\_array\_borror\_gadget : integer  banyak\_array\_user : integer  banyak\_array\_gadget : integer  banyak\_index : integer  banyak\_array\_idpeminjaman : integer  banyak\_array\_idpengembalian : integer  banyak\_array\_idUser : integer  banyak\_array\_idgadget : integer  {proses-proses}  **function** returnIndexreturnGadgetDate  **function** formatReturnGadget |
| **ALGORITMA**  array\_return\_gadget <- buat\_array\_return\_gadget  array\_borrow\_gadget <- buat\_array\_borrow\_gadget  array\_user <- buat\_array\_user  array\_gadget <- buat\_array\_gadget  banyak\_array\_borrow\_gadget <- length(array\_borrow\_gadget)  banyak\_array\_user <- length(array\_user)  banyak\_array\_gadget <- length(array\_gadget)  index <- returnIndexreturnGadgetDate(array\_return\_gadget)  banyak\_index <- length(index)  array\_idpeminjaman <- buat\_array\_idpeminjaman  array\_namapengambil <- buat\_array\_namapengambil  array\_idgadget <- buat\_array\_idgadget  array\_idpengembalian <- buat\_array\_idpengembalian  array\_namagadget <- buat\_array\_namagadget  array\_tanggal <- buat\_array\_tanggal  array\_jumlah <- buat\_array\_jumlah  array\_idUser <- buat\_array\_idUser  banyak\_array\_idpeminjaman <- length(array\_idpeminjaman)  banyak\_array\_idpengembalian <- length(array\_idpengembalian)  banyak\_array\_idUser <- length(array\_idUser)  banyak\_array\_idgadget <- length(array\_idgadget)  i <- 1  traversal i [banyak\_index]  array\_idpeminjaman <- array\_return\_gadget[index[i]].id\_peminjaman  array\_idpengembalian <- array\_return\_gadget[index[i]].id  array\_tanggal <- array\_return\_gadget[index[i]].tanggal\_pengembalian  array\_jumlah <- array\_return\_gadget[index[i]].jumlah\_peminjaman  i <- 1  traversal i [banyak\_array\_idpeminjaman]  j <- 2  traversal j [1, banyak\_array\_borrow\_gadget]  if array\_idpeminjaman[i] == array\_borrow\_gadget[j].id then  array\_idgadget <- array\_borrow\_gadget[j].id\_gadget  array\_idUser <- array\_borrow\_gadget[j].id\_peminjam  i <- 1  traversal i [banyak\_array\_idUser]  j <- 2  traversal j [1, banyak\_array\_user]  if array\_idUser[i] == array\_user[j].id then  array\_namapengambil <- array\_user[j].nama  i <- 1  traversal i [banyak\_array\_idgadget]  j <- 2  traversal j [1, banyak\_array\_gadget]  if array\_idgadget[i] == array\_gadget[j].id then  array\_namagadget <- array\_gadget[j].nama    array\_tempAll <- buat\_array\_tempAll  i <- 1  traversal i [banyak\_array\_idpengembalian]  array\_temp <- buat\_array\_temp  array\_temp <- array\_idpengembalian[i]  array\_temp <- array\_namapengambil[i]  array\_temp <- array\_namagadget[i]  array\_temp <- array\_tanggal[i]  array\_temp <- array\_jumlah[i]  array\_tempAll <- array\_temp  formatReturnGadget(array\_tempAll) |

* 1. F13-Melihat Riwayat pengambilan Consumable

## unit unit\_riwayat pengambilan Consumable

|  |
| --- |
| **procedure** RiwayatAmbilConsumables (output: string ) |
| **KAMUS LOKAL**  **Function** append  { Fungsi bawaan Python untuk menambahkan data ke list }  { I.S. List yang sudah terisi maupun kosong }  { F.S. List ditambahkan data baru }  listTempAll : array[] |
| **ALGORITMA**  index 🡨 returnIndexAmbilConsumablesDate(listConsumableHistory)  idpengambil 🡨 []  idConsumable 🡨 []  idambil 🡨 []  namaConsumables 🡨 []  tanggal 🡨 []  jumlah 🡨 []  namaPengambil 🡨 []  i traversal(len(index))  idpengambil.append(listConsumableHistory[index[i]]['id\_pengambil'])  idConsumable.append(listConsumableHistory[index[i]]['id\_consumable'])  idambil.append(listConsumableHistory[index[i]]['id'])  tanggal.append(listConsumableHistory[index[i]]['tanggal\_pengambilan'])  jumlah.append(listConsumableHistory[index[i]]['jumlah'])    traversal i (len(idpengambil))  traversal j in range(1, len(listUser))  if idpengambil[i] = listUser[j]['id'] then  namaPengambil.append(listUser[j]['nama'])    traversal i (len(idConsumable))  traversal j in range(1, len(listConsumable))  if idConsumable[i] = listConsumable[j]['id'] then  namaConsumables.append(listConsumable[j]['nama'])  listTempAll = []  traversal i(len(idambil))  tempAll 🡨 []  tempAll.append(idambil[i])  tempAll.append(namaPengambil[i])  tempAll.append(namaConsumables[i])  tempAll.append(tanggal[i])  tempAll.append(jumlah[i])  listTempAll.append(tempAll)  formatReturnConsumables(listTempAll) |

* 1. F14-Load Data

## unit unit\_load\_save\_exit;

|  |
| --- |
| **procedure procedure** load |
| **KAMUS LOKAL**  {Proses-proses}  **function** returnFolderName  **function** loadFolder  **function** founFolder  **function** findFolder |
| **ALGORITMA**  Import os  Import argparse  Parser <- argparse.ArgumentParser(description <- ‘Masukan data yang akan di load’)  Parser.add\_argument(‘folder’, type <- str)  Args <- parser.parse\_args() |

* 1. F15-Save Data

## unit unit\_save data

|  |
| --- |
| **Procedure** save (input filename : string, input list : array of dictionary, input foldername : string) |
| **KAMUS LOKAL**  {listFile sudah terdefinisi sebelumnya}  {modul os sudah terdefinisi}  folderName : string  listFile : array of filename  Found : Boolean  I : integer  banyakFile : len(listFile)  banyakArray : len(list)    **Procedure** checkPath(input foldername : string) {mengecek apakah sudah ada filenya apa belum }  **Procedure** makeDir(input foldername : string) {membuat folder baru bila belum ada dan mengoverwrite file jika sudah ada}  **Procedure** toCSV(input path: string; input list: array of dictionary)  {memparse array of dictionary baris per baris kedalam bentuk csv} |
| **ALGORITMA**  input(folderName)  **makeDir**(folderName)  Path <- os.getcwd()  Path += “\” + folderName + “\” + filename  Traversal I [0 … banyakFile]  Traversal I [0 … banyakArray]  **toCSV**(path, list[i])  output(“saving”)  output(“Data telah disimpan pada folder {folderName}” |

* 1. F16-Help

## unit unit\_help :

|  |
| --- |
| **procedure** help (**output** : string) |
| **KAMUS LOKAL**  idx |
| **ALGORITMA**  output(f'========= HELP =========')  if idx == 1 then  output (f'login - untuk melakukan login ke dalam sistem')  output (f'register - untuk melakukan registrasi user baru')  output (f'carirarity - untuk mencari gadget dengan rarity tertentu')  output(f'caritahun - melakukan pencarian gadget berdasarkan tahun ditemukan')  output (f'tambahitem - untuk melakukan penambahan item')  output (f'hapusitem - untuk menghapus suatu item pada database')  output(f'ubahjumlah - digunakan untuk mengubah jumlah gadget dan consumable')  output (f'riwayatpinjam - untuk melihat riwayat pinjam')  output (f'riwayatkembali - untuk melihat riwayat pengembalian')  output (f'riwayatambil - untuk melihat riwayat ambil')    else  (f'carirarity - untuk mencari gadget dengan rarity tertentu')  (f'caritahun - melakukan pencarian gadget berdasarkan tahun ditemukan)  (f'pinjam - digunakan untuk memnijam gadget')  (f'kembalikan - untuk mengemebalikan gadget')  (f'minta - untuk meminta consumable yang tersedia')        output (f'help - untuk menampilkan list command')      output (f'save - untuk mengesave program')      output (f'exit - untuk keluar dari program')      output () |

* 1. F17-Exit

## unit unit\_exit:

|  |
| --- |
| **procedure** exit (filename, list, folderName, text) |
| **KAMUS LOKAL**  Filename, list, folderName, text = string |
| **ALGORITMA**  import sys     if text == 'y' then          save(filename, list, folderName)      else          sys.exit() |

## Hasil Pengujian

**Tabel 03. Hasil Pengujian**

|  |  |
| --- | --- |
| **Fitur** | **Hasil Screenshot** |
| F01-Register |  |
| F02-Login |  |
| F03-Pencarian gadget berdasarkan rarity |  |
| F04-Pencarian gadget berdasarkan tahun ditemukan |  |
| F05-Menambah Item |  |

|  |  |
| --- | --- |
| F06-Menghapus Gadget atau Consumable |  |

|  |  |
| --- | --- |
| F07-Mengubah Jumlah Gadget atau  Consumable pada Inventory |  |
| F08-Meminjam Gadget |  |
| F09-Mengambalikan Gadget |  |
| F10-Meminta Consumable |  |

|  |  |
| --- | --- |
| F11-Melihat Riwayat Peminjaman Gadget |  |

|  |  |
| --- | --- |
| F12-Melihat Riwayat Pengembalian Gadget |  |
| F13-Melihat Riwayat pengambilan Consumable |  |
| F14-Load Data |  |

|  |  |
| --- | --- |
| F15-Save Data |  |

|  |  |
| --- | --- |
| F16-Help | Help untuk user :    Help untuk admin : |
| F17-Exit |  |

# 

# LAMPIRAN

# 

