

**LAPORAN TUGAS BESAR
RUMAH SAKIT NIMONS
IF1210 ALGORITMA DAN PEMROGRAMAN 1**



Disusun oleh Kelompok K01-E :

Samuelson Dharmawan Tanuraharja (13524001)
Muhammad Rafi Akbar (13524125)
Natanael I. Manurung (13524021)
Rhenaldy Cahyadi Putra (13524039)
Miguel Rangga Deardo Sinaga (13524069)

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2025**

LEMBAR PERNYATAAN

“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 Algoritma dan Pemrograman 1 Semester 2 2024/2025.”

Samuelson Dharmawan Tanuraharja/13524001;

Muhammad Rafi Akbar/13524125;

Natanael I. Manurung/13524021;

Rhenaldy Cahyadi Putra/13524039;

Miguel Rangga Deardo Sinaga/13524069.

DAFTAR ISI

LEMBAR PERNYATAAN.....	1
DAFTAR ISI.....	2
DAFTAR TABEL.....	3
DAFTAR GAMBAR.....	4
BAB I.....	5
DESKRIPSI MASALAH.....	5
1.1 Latar Belakang.....	5
1.2 Spesifikasi Program.....	5
BAB II.....	7
2.1 Rencana Implementasi.....	7
2.2 Pembagian Tugas.....	10
2.3 Progress Implementasi.....	13
BAB III.....	15
3.1 Desain Perintah Primitif.....	15
3.2 Desain Kamus Data.....	17
3.3 Desain Dekomposisi dan Algoritmik Program.....	33
3.4 Spesifikasi Modul.....	49
3.5 Pengujian Fitur.....	88
BAB IV.....	90
Lampiran 1. MoM Asistensi 1.....	90
Lampiran 2. MoM Asistensi 2:.....	90

DAFTAR TABEL

Tabel 1. Rencana Implementasi.....	10
Tabel 2. Pembagian Tugas.....	13
Tabel 3. Progress Implementasi Fitur.....	14
Tabel 4. Desain Perintah Primitif.....	17

DAFTAR GAMBAR

Gambar 1. Flowchart Fitur Login.....	34
Gambar 2. Flowchart Fitur Register.....	35
Gambar 3. Flowchart Fitur Logout.....	36
Gambar 4. Flowchart Fitur lupaPassword dan runLengthEncoding.....	36
Gambar 5. Flowchart Fitur.....	37
Gambar 6. Flowchart Fitur printDenah dan lihatRuang.....	38
Gambar 7. Flowchart Fitur lihatUser.....	38
Gambar 8. Flowchart Fitur Cari User, Pasien, dan Dokter.....	39
Gambar 9. Flowchart Fitur lihatAntrian.....	39
Gambar 10. Flowchart Fitur tambahDokter.....	40
Gambar 11. Flowchart Fitur Diagnosis.....	41
Gambar 12. Flowchart Fitur Ngobatin.....	42
Gambar 13. Flowchart Fitur bolehPulang.....	43
Gambar 14. Flowchart Fitur CheckUp.....	44
Gambar 15. Flowchart Fitur antrianSaya.....	45
Gambar 16. Flowchart Fitur minumObat.....	45
Gambar 17. Flowchart Fitur minumPenawar.....	46
Gambar 18. Flowchart exitProgram.....	47
Gambar 19. Flowchart load_data.....	48
Gambar 20. Flowchart save_data.....	49

BAB I

DESKRIPSI MASALAH

1.1 Latar Belakang

Sejak Gro bergabung dengan Anti-Villain League dan meninggalkan dunia kejahatan, kehidupannya berubah drastis. Bersama istrinya, yang bernama Luiy, ia kini menghadapi tantangan baru, yaitu menjaga ketertiban rumah tangga di tengah kekacauan ratusan Nimon—makhluk kecil yang setia, namun Insiden medis ringan yang terus-menerus menjadi masalah yang serius.

Dr. Neroifa, ilmuwan jenius di balik teknologi Gro, menyadari perlunya sistem perawatan kesehatan yang layak. Ia pun menggagas pembangunan rumah sakit khusus Nimon. Namun, masalah muncul ketika proses pengelolaan pasien, dokter, antrian, hingga obat-obatan tidak berjalan teratur. Sistem manual yang diterapkan tidak efektif, dan para Nimon justru memperparah kekacauan.

Untuk itu, diperlukan sebuah sistem manajemen rumah sakit yang terstruktur dan efisien. Sebagai mahasiswa IF1210, kami dipercaya untuk merancang solusi untuk membantu Dr. Neroifa mewujudkan rumah sakit yang fungsional, aman, dan tertib bagi para Nimon melalui Tugas Besar Algoritma dan Pemrograman I.

1.2 Spesifikasi Program

Untuk menciptakan suatu program yang dapat digunakan untuk memudahkan pekerjaan Dr Neroifa di rumah sakit, kami membuat fitur-fitur sebagai berikut:

1. Fitur Utama

a. Manajemen Pengguna:

- Login dan Registrasi (Pasien, Dokter, Manager).
- Manajemen permission user berdasarkan tingkatan (hierarki)
- Reset password (Run-Length Encoding).
- Log-out

b. Manajemen Rumah Sakit:

- Visualisasi dan operasi denah rumah sakit

- Manajemen ruangan (Assignment dokter dan pasien)
 - Diagnosis penyakit (dokter) & resep obat.
- c. Pasien:
- Daftar check-up , minum obat , pulang.
 - Kondisi kesehatan (Tekanan darah, suhu, dsb)

2. Fitur Tambahan

- a. Manajemen data user, penyakit dan obatnya, beserta kondisi fisik rumah sakit melalui file eksternal.
- b. Denah rumah sakit dinamis.
- c. Visualisasi antrian pasien.

BAB II

RENCANA DAN PROGRESS IMPLEMENTASI

2.1 Rencana Implementasi

Untuk memastikan program manajemen rumah sakit dapat dibangun secara terstruktur dan efisien, diperlukan sebuah *rencana implementasi* yang merincikan setiap tipe data yang akan digunakan dalam masing-masing fitur. Rencana ini bertujuan untuk memetakan kebutuhan struktur data secara sistematis agar proses pengembangan program berjalan lebih terarah dan optimal.

Implementasi ADT	Fitur Bersangkutan	Alasan Implementasi
ADT Sederhana	User	Semua fitur yang berhubungan dengan user (F01–F18) Mewakili informasi pengguna seperti ID, nama, role, dan password
	ListUser	Semua fitur yang berhubungan dengan user (F01-F18) Sebagai container data user pada user.csv
	Room	Semua fitur yang berhubungan dengan denah (config.txt) (F06, F11, F12, F14, F15) Mewakili informasi terkait data ID dokter dan ID pasien pada ruangan
	Obat	F11-Diagnosis, F12-Ngobatin, F16-Minum Obat Mewakili informasi obat seperti id dan nama obat
	ListObat	F11-Diagnosis, F12-Ngobatin, F16-Minum Obat Sebagai container data obat pada obat.csv
	Penyakit	F11-Diagnosis, F12-Ngobatin, Mewakili informasi terkait penyakit seperti id, nama, dan data data kesehatan yang mendukung diagnosa
	ListPenyakit	F11-Diagnosis, F12-Ngobatin, Sebagai container data penyakit

		pada penyakit.csv
Formula	F11-Diagnosis, F12-Ngobatin	Mewakili informasi terkait id obat untuk penyakit apa dan dikonsumsi di urutan berapa
ListFormula	F11-Diagnosis, F12-Ngobatin	Sebagai container data formula pada obat_penyakit.csv
Ruangan	F06 - Denah Rumah Sakit, F09 - Lihat Antrian, F10 - Tambah Dokter, F11 - Diagnosis, F12 - Ngobatin, F14 - Daftar Check-Up, F15 - Antrian Saya!	Mewakili informasi terkait id_dokter dan antrian pasien pada ruangan tersebut
RumahSakit	F06 - Denah, F09 - Lihat Antrian ,F10 - Tambah Dokter, F11 - Diagnosis, F12 - Ngobatin, F14 - Daftar Check Up, F15 - Antrian Saya	Sebagai container data terkait kondisi fisik rumah sakit dari config.txt yang berisikan dimensi rumah sakit(2d), array berisikan ADT ruangan, serta batasan untuk kapasitas ruangan dan kapasitas barisannya
Pasien	F06 - Denah, F14 - Daftar Check Up,F15 - Antrian Saya, F16 - Minum Obat,	Mewakili informasi terkait pasien yang memiliki obat, beserta dengan obat yang sudah/belum dikonsumsinya
Inventory	F12 - Ngobatin, F16 Minum Obat	Sebagai container data terkait pasien yang memiliki obat dari config.txt
ADT List	F13- Aku boleh pulang ga dok?	Sebagai substitusi array of

		integer
ADT Linked List	F14-Daftar Check Up	
ADT Matrix	ADT rumahsakit	Sebagai kontainer variabel ruangan di ADT rumahsakit
ADT Set	F01-Login, F02-Register	Untuk mewakili dataset yang tidak boleh diulangi seperti
ADT Penyakit dan obat	F11-Diagnosis, F12-Ngobatin, F16-Minum Obat	
ADT Map	F12-Ngobatin	Digunakan untuk assignment terhadap sebuah string seperti opsi input yang dipilih sehingga dapat ditranslate menjadi sebuah integer yang lebih mudah untuk diproses
ADT Queue	Semua fungsi yang menggunakan ADT Room	Digunakan dalam merepresentasikan sebuah barisan seperti barisan pada rumah sakit.
ADT Stack	F16-Minum Obat	Digunakan dalam merepresentasi data yang ditumpuk seperti urutan konsumsi obat pada pasien
File External	Semua fungsi yang menggunakan file external, sejauh ini:	Membaca dan menyimpan data dari/ke file .csv dan .txt

	(F01, F02, F04, F06)	
--	----------------------	--

Tabel 1. Rencana Implementasi

2.2 Pembagian Tugas

Agar pembuatan program rumah sakit dapat diselesaikan dengan efisien dibutuhkan pembagian tugas yang merata berdasarkan fitur, implementasi, desainer, coder, dan tester.

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F00 - Rencana Implementasi	<ul style="list-style-type: none"> - ADT User, ListUser, List, Linked List, Matrix, Set, Map, Stack, Queue, Parser. - File External (user.csv, penyakit.csv, obat.csv, obat_penyakit.csv, config.txt). (Buka & simpan) - Array Search, Sort, Filter. - Building main.c dan makefile, serta perancangan UI. 	13524021	13524021, 13524001	13524021, 13524001

F01 - Login	<u>procedure</u> Validasi user, ADT List User	13524125	13524125	13524125
F02 - Register Pasien	ADT List, <u>procedure</u> registrasiUser	13524125	13524125	13524125
F03 - Logout	<u>procedure</u> logout	13524039	13524039	13524039
F04 - Lupa Password	<u>procedure</u> lupaPassword, runLengthEncoding	13524039	13524039	13524039
F05 - Menu & Help	<u>procedure</u> help	13524069	13524069	13524069
F06 - Denah Rumah Sakit	<u>procedure</u> printDenah, <u>procedure</u> lihatRuang	13524001	13524001	13524001
F07 - Lihat User	<u>procedure</u> LihatUser	13524001	13524001	13524001
F08 - Cari User	<u>procedure</u>	13524001	13524001	13524001
F09 - Lihat Antrian	<u>procedure</u> lihatSemuaAntrian	13524001	13524001	13524001
F10 - Tambah Dokter	<u>procedure</u> tambahDokter, <u>procedure</u> assignDokter	13524001	13524001	13524001

F11 - Diagnosis	<u>procedure</u> Diagnosis <u>function</u> CariRuanganDokter	13524125, 13524001	13524125	13524125
F12 - Ngobatin	<u>procedure</u> , TambahObatKeInventory,Ngobatin, <u>function</u> CariRuanganDokter	13524125, 13524001	13524125	13524125
F13 - Pulang Dok	<u>procedure</u> bolehPulang	13524039	13524039	13524039
F14 - Daftar Check Up	<u>procedure</u> tampilkanDokterTersedia, DaftarCheckUp	13524125	13524125	13524125
F15 - Antrian Saya	<u>procedure</u> antrianSaya	13524039	13524039	13524039
F16- Minum Obat	<u>procedure</u> minumObat	13524039	13524039	13524039
F17- Minum Penawar	<u>procedure</u> minumPenawar	13524039	13524039	13524039
F18 - Exit	<u>procedure</u> exit, saveData	13524069	13524069	13524069

D01 - Denah Rumah Sakit	<u>procedure</u> printDenah, <u>procedure</u> lihatRuang	13524001	13524001	13524001
D02 - Lihat Antrian	<u>procedure</u> lihatSemuaAntrian	13524001	13524001	13524001
D03 - Load	<u>function</u> validate_folder, <u>procedure</u> check_manager_account, <u>procedure</u> load data.	13524069	13524069	13524069
D04 - Save	<u>function</u> create_directory_recursive, <u>procedure</u> save_data	13524069	13524069	13524069

Tabel 2. Pembagian Tugas

2.3 Progress Implementasi

Terdapat juga progress rancangan, implementasi, dan testing setiap fitur sebagai berikut.

Fitur	Desain	Implementasi	Testing
F01 - Login	V	V	V
F02 - Register Pasien	V	V	V
F03 - Logout	V	V	V
F04 - Lupa Password	V	V	V

F05 - Menu & Help	V	V	V
F06 - Denah Rumah Sakit	V	V	V
F07 - Lihat User	V	V	V
F08 - Cari User	V	V	V
F09 - Lihat Antrian	V	V	V
F10 - Tambah Dokter	V	V	V
F11 - Diagnosis	V	V	V
F12 - Ngobatin	V	V	V
F13 - Pulang Dok	V	V	V
F14 - Daftar Check Up	V	V	V
F15 - Antrian Saya	V	V	V
F16 - Minum Obat	V	V	V
F17 - Minum Penawar	V	V	V
F18 - Exit	V	V	V

Tabel 3. Progress Implementasi Fitur

BAB III

DESAIN PROGRAM

3.1 Desain Perintah Primitif

Command	Input	Output
LOGIN	accounts	Login user jika username dan password benar
REGISTER	Accounts, username, password	Menambahkan akun baru ke list user
LOGOUT	current_user	Logout dari sistem dan kembali ke state belum login
LUPA_PASSWORD	accounts	Memvalidasi user berdasarkan kode unik RLE dari username. Mengatur ulang password jika valid.
HELP	current_user	Menampilkan fitur apa saja yang bisa diakses oleh user
LIHAT_DENAH	ADT RumahSakit	Tampilan denah Rumah Sakit berbentuk matriks 2 dimensi.
LIHAT_RUANGAN	ADT RumahSakit, accounts, dan kode ruangan (A1, B3, dll)	Melihat isi ruangan dengan kode tertentu dan menampilkan dokter di dalamnya dan pasien di dalam ruangan.
LIHAT_USER	accounts, mode	Menampilkan seluruh data user dengan nama/id terurut menaik/menurun
LIHAT_DOKTER	accounts, mode	Menampilkan seluruh data dokter dengan nama/id terurut menaik/menurun
LIHAT_PASIEN	accounts, mode	Menampilkan seluruh data pasien dengan nama/id terurut menaik/menurun
CARI_USER	accounts	Mencari data user berdasarkan nama/id

CARI_DOKTER	accounts	Mencari data user berdasarkan nama/id
CARI_PASIEN	accounts	Mencari data user berdasarkan nama/id/penyakit
LIHAT_SEMUA_ANTRIAN	ADT RumahSakit, accounts	Melihat seluruh ruangan yang memiliki dokter di dalamnya
TAMBAH_DOKTER	accounts	Menambahkan seorang user dengan role dokter yang memiliki username dan password baru.
ASSIGN_DOKTER	ADT RumahSakit, accounts	Menugaskan user dengan role dokter ke salah satu ruangan di Rumah Sakit Nimons.
DIAGNOSIS	current_user (dokter), ADT RumahSakit, accounts, penyakit	Mendiagnosis pasien terdepan di antrian berdasarkan data medis
NGOBATIN	current_user (dokter), ADT RumahSakit, accounts, penyakit, obat, formula	Memberi obat ke pasien berdasarkan diagnosis dan formula, menambah ke inventory
PULANGDOK	current_user (pasien), inventory, ADT RumahSakit	Memeriksa apakah pasien sudah memenuhi syarat untuk pulang: telah di diagnosis, semua obat dikonsumsi, urutan obat benar.
DAFTAR_CHECKUP	Accounts, ADT RumahSakit, username	Memasukkan data medis pasien dan mendaftarkannya ke antrian ruangan dokter
ANTRIAN	current_user (pasien), ADT RumahSakit	Menampilkan status antrian pasien terhadap dokter yang dituju: posisi dan total antrian.
MINUM_OBAT	current_user (pasien), inventory, ADT RumahSakit	Memindahkan obat dari inventory ke "perut" (stack), memperhatikan urutan pemakaian obat.
PENAWAR	current_user (pasien),	Mengeluarkan obat terakhir

	inventory	dari perut dan mengembalikannya ke inventory.
EXIT	Accounts, ADT RumahSakit	
SAVE	Accounts, ADT RumahSakit	Melakukan save data ke csv atau txt

Tabel 4. Desain Perintah Primitif

3.2 Desain Kamus Data

```

{ KAMUS ADT User dengan Alokasi Statik}
{ Berisi definisi dan semua primitif pemrosesan data User}

constant MAX_FIELD: integer = 128

type User : <
    id: integer,
    username: string,
    password: string,
    role: string,
    riwayat_penyakit: string,
    suhu_tubuh: real,
    tekanan_darah_sistolik: integer,
    tekanan_darah_diastolik: integer,
    detak_jantung: integer,
    saturasi_oksigen: real,
    kadar_gula_darah: integer,
    berat_badan: real,
    tinggi_badan: integer,
    kadar_kolesterol: integer,
    kadar_kolesterol_ldl: integer,
    trombosit: integer
>

type ListUser : <
    users : array [0..MAX_FIELD - 1] of User,
    jumlahuser : integer
>

{ ***** PRIMITIF PRIMITIF ***** }

{ ***** KONSTRUKTOR ***** }

procedure CreateUser(input/output U: User)
{ I.S. sembarang }

```

```

{ F.S. Terbentuk User default dengan semua field bernilai awal
}

procedure InitializeListUser(input/output L: ListUser)
{ I.S. sembarang }
{ F.S. Terbentuk ListUser kosong, jumlahuser = 0 }

{***** SELEKTOR*****}

function GetUser (L: ListUser, id: integer) → User
{ Mencari User dengan ID tertentu }
{ Jika tidak ditemukan, mengembalikan User dengan id = -1 }

{***** MUTATOR*****}

procedure InsUser(input/output L: ListUser, input U: User)
{ Menyisipkan User U ke ListUser L secara urut menaik
berdasarkan id }
{ Jika ID sudah ada, tidak disisipkan }

procedure DelUser(input/output L: ListUser, input id: integer,
output U: User)
{ Menghapus user dengan ID tertentu dan menyimpan salinannya ke
U }

{***** OUTPUT *****}

procedure TampilkanUser(input L: ListUser)
{ Menampilkan semua informasi user pada list ke layar secara
berurutan }

```

```

{ MODUL LIST PENYAKIT DAN OBAT DENGAN ALOKASI STATIK }
constant MAX_OBAT : integer = 100
constant MAX_PENYAKIT: integer = 100
constant MAX_FORMULA: integer = 100
constant MAX_NAMA: integer = 100
constant MAX_FIELD: integer = 128

type Obat : <
    id : integer,
    nama : string
>

```

```

type ListObat : <
    obats : array [IdxMinObat..IdxMaxObat] of Obat,
    jumlahobat : integer
>

{ **** KONSTRUKTOR **** }
procedure CreateObat(input/output o: Obat)
{I.S Sembarang}
{ F.S terbentuk obat kosong }

procedure InitializeListObat(input/output l: ListObat)
{I.S Sembarang}
{F.S Membentuk list obat kosong }

{ **** OPERASI **** }
function GetObat(l: ListObat, id: integer) → Obat
{ Mencari obat berdasarkan ID }

procedure DelObat(input/output l: ListObat, input id: integer,
output buffer: Obat)
{ Menghapus obat berdasarkan ID }

procedure InsObat(input/output l: ListObat, input buffer: Obat)
{ Menyisipkan obat ke list }

procedure tampilan_obat(input l: ListObat)
{ Menampilkan semua obat }

{TYPE PENYAKIT}

type Penyakit : <
    id: integer,
    nama: string,
    suhu_min, suhu_max: real,
    sys_min, sys_max: integer,
    dias_min, dias_max: integer,
    detak_min, detak_max: integer,
    saturasi_min, saturasi_max: real,
    gula_min, gula_max: real,
    berat_min, berat_max: real,
    tinggi_min, tinggi_max: integer,
    kolesterol_min, kolesterol_max: integer,
    trombosit_min, trombosit_max: integer
>

```

```

type ListPenyakit : <
    penyakits : array [0..MAX_PENYAKIT] of Penyakit,
    jumlahpenyakit : integer
>

{***** KONSTRUKTOR *****}
procedure CreatePenyakit(input/output p: Penyakit)
{I.S Sembarang}
{ Membentuk penyakit kosong }

procedure InitializeListPenyakit(input/output l: ListPenyakit)
{I.S Sembarang}
{ Membentuk list penyakit kosong }

{***** OPERASI *****}
function GetPenyakit(l: ListPenyakit, id: integer) → Penyakit
{ Mencari penyakit berdasarkan ID }

procedure DelPenyakit(input/output l: ListPenyakit, input id:
integer, output buffer: Penyakit)
{ Menghapus penyakit berdasarkan ID }

procedure InsPenyakit(input/output l: ListPenyakit, input
buffer: Penyakit)
{ Menyisipkan penyakit ke list }

procedure tampilan_penyakit(input/output l: ListPenyakit)
{ Menampilkan semua penyakit }

{TYPE PENYAKIT}

type Formula : <
    obat_id: integer,
    penyakit_id: integer,
    urutan: integer
>

type ListFormula : <
    formulas : array [0..MAX_FORMULA - 1] of Formula,
    jumlahformula : integer
>

{***** KONSTRUKTOR *****}
procedure CreateFormula(input/output f: Formula)

```

```

{ Membentuk formula kosong }

procedure InitializeListFormula(input/output l: ListFormula)
{ Membentuk list formula kosong }

{***** OPERASI *****}
function GetFormula(l: ListFormula, obat_id: integer,
penyakit_id: integer) → Formula
{ Mengambil formula berdasarkan pasangan id }

procedure DelFormula(input/output l: ListFormula, input
obat_id: integer, input penyakit_id: integer, output buffer:
Formula)
{ Menghapus formula berdasarkan pasangan id }

procedure InsFormula(input/output l: ListFormula, input buffer:
Formula)
{ Menambahkan formula ke list }

procedure tampilan_formula(input/output l: ListFormula)
{ Menampilkan semua formula }

```

```

{ MODUL RUMAH SAKIT, RUANGAN, PASIEN, DAN INVENTORY }
{ Berisi definisi dan primitif untuk pengolahan data rumah
sakit dari config.txt }

constant MAX_OBAT: integer = 100
constant MAX_KOLOM_RUANGAN: integer = 26
constant MAX_BARIS_RUANGAN: integer = 26

{ ***** TIPE BENTUKAN ***** }
type Ruangan : <
    dokterId : integer, { 0 jika tidak ada dokter },
    antrianPasienIds : Queue { antrian pasien pada
ruangan }
>

type RumahSakit : <
    rows : integer,
    cols : integer,
    data : array[0..MAX_BARIS_RUANGAN-1] of array
0..MAX_KOLOM_RUANGAN-1] of Ruangan,
    kapasitasPerRuang : integer,

```

```

    kapasitasBaris : integer
>

type Pasien : <
    id : integer,
    jumlahobat : integer,
    obat : array [0..MAX_OBAT-1] of integer,
    perut : Stack
>

type Inventory : <
    jumlahPasienwObat : integer,
    data : array [0..CAPACIRY_QUEUE-1] of Pasien
>

{***** KONSTRUKTOR *****}

procedure InitializeRumahSakit(input/output rs: RumahSakit)
{ I.S. sembarang }
{ F.S. RumahSakit rs terbentuk kosong, semua ruangan belum memiliki dokter maupun pasien }

procedure InitializePasien(input/output P: Pasien)
{ I.S. sembarang }
{ F.S. Pasien P terbentuk dengan id 0, tidak memiliki obat, dan perut kosong }

procedure InitializeInventory(input/output I: Inventory)
{ I.S. sembarang }
{ F.S. Inventory I terbentuk kosong, jumlah pasien dengan obat = 0 }

```

```

{ MODUL MAP DENGAN ALOKASI STATIK }
{ Berisi definisi dan semua primitif pemrosesan data pasangan key-value }
{ Elemen disimpan secara rapat kiri }

{ ***** KONSTANTA ***** }
constant CAPACITY_MAP: integer = 100

{ ***** TIPE BENTUKAN ***** }

type KeyType : array [0..49] of string
type ValueType : integer

```

```

type MapElType : <
    key : KeyType,
    value : ValueType
>

type Map : <
    elements : array [0..CAPACITY_MAP-1] of MapElType,
    count : integer
>

{ ***** KONSTRUKTOR ***** }

procedure CreateEmptyMap(input/output m: Map)
{ I.S. sembarang }
{ F.S. terbentuk map kosong dengan count = 0 }

{ ***** PREDIKAT ***** }

function IsMapEmpty(m: Map) → boolean
{ Mengirimkan true jika map kosong (count = 0) }

function IsMapFull(m: Map) → boolean
{ Mengirimkan true jika map penuh (count = CAPACITY_MAP) }

function IsKeyMember(m: Map, k: KeyType) → boolean
{ Mengirimkan true jika k adalah key dari salah satu elemen dalam map }

{ ***** OPERASI DASAR ***** }

procedure InsertMap(input/output m: Map, input k: KeyType,
input v: ValueType)
{ I.S. Map m terdefinisi, tidak penuh }
{ F.S. Jika key k belum ada, maka pasangan (k, v) ditambahkan ke dalam map }

procedure DeleteMap(input/output m: Map, input k: KeyType)
{ I.S. Map m terdefinisi, dan k terdapat dalam map }
{ F.S. Elemen dengan key = k dihapus dari map, elemen di belakangnya bergeser ke kiri }

function GetValue(m: Map, k: KeyType) → ValueType
{ Mengembalikan value yang terkait dengan key k }
{ Prekondisi: k terdapat dalam map }

procedure PrintMap(input/output m: Map)

```

```

{ I.S. Map m terdefinisi }
{ F.S. Menampilkan seluruh pasangan key-value dalam map ke
layar }

```

```

{ MODUL LIST DENGAN ALOKASI STATIK }
{ Berisi definisi dan primitif dalam pengolahan array of
integer }

{ ***** KONSTANTA ***** }
type ElType : integer
type IdxType : integer

constant integer CAPACITY_LIST = 100
constant integer IDX_MIN = 0
constant integer IDX_UNDEF = -1
constant integer MARK = -9999

{ ***** TIPE BENTUKAN ***** }
type ListStatik :
    < contents : array[0..CAPACITY_LIST-1] of ElType >

{ ***** KONSTRUKTOR ***** }
procedure CreateListStatik(output l : ListStatik)
{I. S. l terdefinisi}
{F. S. l terisi dengan mark}

{ ***** PREDIKAT ***** }
function listLength(l : ListStatik) → integer
{Mengembalikan panjang l}

function getListFirstIdx(l : ListStatik) → IdxType
{Mengembalikan index pertama l}

function getListLastIdx(l : ListStatik) → IdxType
{Mengembalikan index terakhir l}

function isIdxValidList(l : ListStatik, i : IdxType) → boolean
{Mengembalikan true jika i tidak < 0 dan > CAPACITY_LIST}

function isIdxEffList(l : ListStatik, i : IdxType) → boolean

```

```

{Mengembalikan true jika l[i] memiliki nilai}

function isEmpty(l : ListStatik) → boolean
{Mengembalikan true jika l kosong}

function isListFull(l : ListStatik) → boolean
{Mengembalikan true jika l penuh}

{ ***** OPERASI ***** }
procedure readList(output l : ListStatik)
{I. S. l terdefinisi}
{F. S. l diisi dengan membaca input user}

procedure printList(l : ListStatik)
{I. S. l terdefinisi}
{F. S. l dicetak ke layar}

function plusMinusList(l1 : ListStatik, l2 : ListStatik, plus : boolean) → ListStatik
{Mengembalikan list dengan setiap element di n = l1[n] + l2[n]}

function isEqual(l1 : ListStatik, l2 : ListStatik) → boolean
{Mengembalikan true jika l1 = l2}

function indexOfList(l : ListStatik, val : ElType) → integer
{Mengembalikan index elemen pertama yang bernilai val di l}

procedure extremeValuesList(input l : ListStatik, output max : ElType, output min : ElType)
{I. S. l, max, min terdefinisi}
{F. S. max diisi dengan nilai terbesar di l dan terkecil diisi di min}

procedure insertFirstList(input/output l : ListStatik, val : ElType)
{I. S. l dan val terdefinisi}
{F. S. val dimasukkan ke awal l}

procedure insertAtList(input/output l : ListStatik, val : ElType, idx : IdxType)
{I. S. l dan val terdefinisi}
{F. S. val dimasukkan ke l di index idx}

procedure insertLastList(input/output l : ListStatik, val : ElType)

```

```

{I. S. l dan val terdefinisi}
{F. S. val dimasukkan ke akhir l}

procedure deleteFirstList(input/output l : ListStatik, output
val : ElType)
{I. S. l dan val terdefinisi}
{F. S. nilai index awal l dihapus dan disimpan di val}

procedure deleteAtList(input/output l : ListStatik, output val
: ElType, idx : IdxType)
{I. S. l, idx, val terdefinisi}
{F. S. nilai index idx di l dihapus dan disimpan di val}

procedure deleteLastList(input/output l : ListStatik, output
val : ElType)
{I. S. l dan val terdefinisi}
{F. S. nilai index awal l dihapus dan disimpan di val}

procedure sortList(input/output l : ListStatik, asc : boolean)
{I. S. l dan asc terdefinisi}
{F. S. l diurut dari kecil ke besar jika asc = true, dan
sebaliknya}

```

```

{ MODUL STACK DENGAN ALOKASI STATIK }
{ Berisi definisi dan primitif dalam pengolahan stack of
integer menggunakan array statik }

{ ***** KONSTANTA ***** }
type ElTypeStack : integer
type IdxTypeStack : integer

constant integer CAPACITY_STACK = 100
constant integer IDX_UNDEF_STACK = -1

{ ***** TIPE BENTUKAN ***** }
type Stack :
< T : array [0..CAPACITY_STACK-1] of ElTypeStack,
TOP : IdxTypeStack >

{ ***** SELEKTOR ***** }
function Top(S : Stack) → IdxTypeStack

```

```

{Mengembalikan indeks elemen paling atas dari stack S}

function InfoTop(S : Stack) → ElTypeStack
{Mengembalikan nilai elemen paling atas dari stack S}

{ ***** KONSTRUKTOR ***** }
procedure CreateEmptyStack(output S : Stack)
{I. S. S terdefinisi}
{F. S. S merupakan stack kosong dengan TOP = IDX_UNDEF_STACK}

{ ***** PREDIKAT ***** }
function isEmptyStack(S : Stack) → boolean
{Mengembalikan true jika stack S kosong}

function isFullStack(S : Stack) → boolean
{Mengembalikan true jika stack S penuh}

{ ***** OPERASI ***** }
procedure Push(input/output S : Stack, val : ElTypeStack)
{I. S. S mungkin kosong, tetapi tidak penuh}
{F. S. val ditambahkan menjadi elemen paling atas dari S}

procedure Pop(input/output S : Stack, output val : ElTypeStack)
{I. S. S tidak kosong}
{F. S. elemen paling atas S dihapus dan disimpan ke val}

```

```

{ MODUL SET DENGAN ALOKASI STATIK }
{ Berisi definisi dan primitif dalam pengolahan set of integer
menggunakan array statik dengan elemen unik }

{ ***** KONSTANTA ***** }
type ElTypeSet : integer
type IdxTypeSet : integer

constant integer CAPACITY_SET = 100

{ ***** TIPE BENTUKAN ***** }
type Set :
< elements : array [0..CAPACITY_SET-1] of ElTypeSet,
  count : integer >

{ ***** SELEKTOR ***** }
function Count(s : Set) → integer
{Mengembalikan jumlah elemen dalam set s}

```

```

function Elmt(s : Set, i : IdxTypeSet) → ElTypeSet
{Mengembalikan elemen ke-i dari set s}

{ ***** KONSTRUKTOR ***** }
procedure CreateEmptySet(output s : Set)
{I. S. s sembarang}
{F. S. s merupakan set kosong, count bernilai 0}

{ ***** PREDIKAT ***** }
function isSetEmpty(s : Set) → boolean
{Mengembalikan true jika set kosong (count = 0)}

function isSetFull(s : Set) → boolean
{Mengembalikan true jika set penuh (count = CAPACITY_SET)}

function isMember(s : Set, val : ElTypeSet) → boolean
{Mengembalikan true jika val terdapat dalam set s}

{ ***** OPERASI ***** }
procedure insertSet(input/output s : Set, val : ElTypeSet)
{I. S. s terdefinisi dan tidak penuh}
{F. S. val ditambahkan ke set jika belum ada di dalamnya}

procedure deleteSet(input/output s : Set, val : ElTypeSet)
{I. S. s terdefinisi}
{F. S. val dihapus dari set jika ada}

function setUnion(s1 : Set, s2 : Set) → Set
{Mengembalikan hasil union dari s1 dan s2}

function setIntersection(s1 : Set, s2 : Set) → Set
{Mengembalikan hasil intersection dari s1 dan s2}

function setDifference(s1 : Set, s2 : Set) → Set
{Mengembalikan hasil difference dari s1 terhadap s2}

procedure printSet(s : Set)
{I. S. s terdefinisi}
{F. S. isi set ditampilkan dalam format {a,b,c,...}}

```

```

{ MODUL QUEUE DENGAN ALOKASI STATIK }
{ Berisi definisi dan primitif dalam pengolahan Queue (antrian)
menggunakan array statik secara eksplisit dan buffer melingkar
}

```

```

{ ***** KONSTANTA ***** }
type ElTypeQueue : integer
type IdxTypeQueue : integer

constant integer CAPACITY_QUEUE = 100
constant integer IDX_UNDEF = -1

{ ***** TIPE BENTUKAN ***** }
type Queue :
< buffer : array [0..CAPACITY_QUEUE-1] of ElTypeQueue,
  idxHead : IdxTypeQueue,
  idxTail : IdxTypeQueue >

{ ***** SELEKTOR ***** }
function IdxHead(q : Queue) → IdxTypeQueue
{Mengembalikan indeks elemen head dari q}

function IdxTail(q : Queue) → IdxTypeQueue
{Mengembalikan indeks elemen tail dari q}

function Head(q : Queue) → ElTypeQueue
{Mengembalikan elemen di head dari q}

function Tail(q : Queue) → ElTypeQueue
{Mengembalikan elemen di tail dari q}

{ ***** KONSTRUKTOR ***** }
procedure CreateQueue(output q : Queue)
{I. S. q sembarang}
{F. S. Sebuah queue kosong terbentuk dengan idxHead = IDX_UNDEF
dan idxTail = IDX_UNDEF}

{ ***** PREDIKAT ***** }
function isEmptyQueue(q : Queue) → boolean
{Mengembalikan true jika queue kosong}

function isFullQueue(q : Queue) → boolean
{Mengembalikan true jika queue penuh (buffer melingkar) }

function lengthQueue(q : Queue) → integer
{Mengembalikan banyaknya elemen dalam queue}

{ ***** OPERASI ***** }
procedure enqueue(input/output q : Queue, val : ElTypeQueue)
{I. S. q terdefinisi, mungkin kosong, mungkin penuh}
{F. S. val menjadi elemen baru di tail q jika tidak penuh}

```

```

procedure dequeue(input/output q : Queue, output val :
ElTypeQueue)
{I. S. q terdefinisi, mungkin kosong}
{F. S. elemen di head dihapus dan disimpan ke val jika tidak
kosong}

procedure displayQueue(q : Queue)
{I. S. q terdefinisi}
{F. S. isi queue dicetak dalam format [a,b,c,...] tanpa spasi,
diakhiri newline}

```

```

{ MODUL MATRIX }
{ Berisi definisi dan primitif dalam pengolahan matriks integer
dengan alokasi statik maksimum ROW_CAP x COL_CAP }

{ ***** KONSTANTA ***** }
type ElTypeMatrix : integer
constant integer ROW_CAP = 100
constant integer COL_CAP = 100
constant integer IDX_MIN = 0
constant integer IDX_UNDEF = -1

{ ***** TIPE BENTUKAN ***** }
type Matrix :
< mem : array [0..ROW_CAP-1, 0..COL_CAP-1] of ElTypeMatrix,
  rowEff : integer, { banyak baris efektif, 0..ROW_CAP }
  colEff : integer { banyak kolom efektif, 0..COL_CAP } >

{ ***** SELEKTOR ***** }
function RowEff(m : Matrix) → integer
{Mengembalikan banyak baris efektif matrix m}

function ColEff(m : Matrix) → integer
{Mengembalikan banyak kolom efektif matrix m}

function ElmtMtrx(m : Matrix, i : integer, j : integer) →
ElTypeMatrix
{Mengembalikan elemen baris i kolom j matrix m, i dan j indeks
efektif}

{ ***** KONSTRUKTOR ***** }
procedure CreateMatrix(nRows : integer, nCols : integer, output
m : Matrix)
{I.S. nRows dan nCols valid (0 < nRows ≤ ROW_CAP, 0 < nCols ≤

```

```

COL_CAP) }

{F.S. Matrix m dengan ukuran nRows x nCols terbentuk, nilai
elemen tidak terdefinisi}

{ ***** VALIDASI INDEKS ***** }
function IsIdxValidMatrix(i : integer, j : integer) → boolean
{Mengembalikan true jika i dan j dalam rentang indeks
0..ROW_CAP-1 dan 0..COL_CAP-1}

function IsMatrixIdxEff(m : Matrix, i : integer, j : integer) →
boolean
{Mengembalikan true jika i dan j merupakan indeks efektif
matrix m ( $0 \leq i < \text{rowEff}$  dan  $0 \leq j < \text{colEff}$ )}

{ ***** OPERASI ***** }
procedure ReadMatrix(output m : Matrix, nRows : integer, nCols
: integer)
{I.S. nRows dan nCols valid}
{F.S. Elemen matrix m dengan ukuran nRows x nCols diisi dari
}

procedure DisplayMatrix(m : Matrix)
{I.S. m terdefinisi}
{F.S. Matrix m dicetak dalam format baris per baris dengan
spasi antar elemen}

function AddMatrix(m1 : Matrix, m2 : Matrix) → Matrix
{Mengembalikan matrix hasil penjumlahan m1 dan m2, dengan
syarat ukuran sama}

function SubtractMatrix(m1 : Matrix, m2 : Matrix) → Matrix
{Mengembalikan matrix hasil pengurangan m1 dan m2, dengan
syarat ukuran sama}

function MultiplyMatrix(m1 : Matrix, m2 : Matrix) → Matrix
{Mengembalikan matrix hasil perkalian m1 dan m2, dengan syarat
colEff m1 = rowEff m2}

function MultiplyByConst(m : Matrix, x : ElTypeMatrix) → Matrix
{Mengembalikan matrix hasil perkalian setiap elemen m dengan
konstanta x}

function IsSquare(m : Matrix) → boolean
{Mengembalikan true jika matrix m berbentuk persegi (rowEff =
colEff)}

```

```

function Transpose(m : Matrix) → Matrix
{Mengembalikan matrix transpose dari m}

function Determinant2x2(m : Matrix) → ElTypeMatrix
{Mengembalikan determinan matrix 2x2 m (asumsi m berukuran
2x2) }

```

```

{ MODUL LINKEDLIST }
{ Berisi definisi dan primitif pengolahan linked list integer }

{ ***** TIPE BENTUKAN ***** }
type Node :
    record
        data : integer
        next : pointer to Node
    endrecord

type LinkedList :
    record
        head : pointer to Node
    endrecord

{ ***** KONSTRUKTOR ***** }
procedure InitList(output l : LinkedList)
{I.S. sembarang}
{F.S. linked list kosong dengan head = null}

function CreateNode(val : integer) → pointer to Node
{Mengembalikan alamat node baru dengan data = val dan next =
null}

{ ***** OPERASI ***** }
procedure InsertAtHead(input/output l : LinkedList, val :
integer)
{I.S. l terdefinisi}
{F.S. node baru dengan data val menjadi elemen pertama linked
list}

procedure InsertAtTail(input/output l : LinkedList, val :
integer)
{I.S. l terdefinisi}
{F.S. node baru dengan data val menjadi elemen terakhir linked
list}

```

```

function DeleteNode(input/output l : LinkedList, val : integer)
→ boolean
{Menghapus node pertama dengan data = val dari linked list l}
{Mengembalikan true jika node ditemukan dan dihapus, false jika
tidak}

function SearchNode(l : LinkedList, val : integer) → pointer to
Node
{Mengembalikan alamat node dengan data = val jika ada, null
jika tidak}

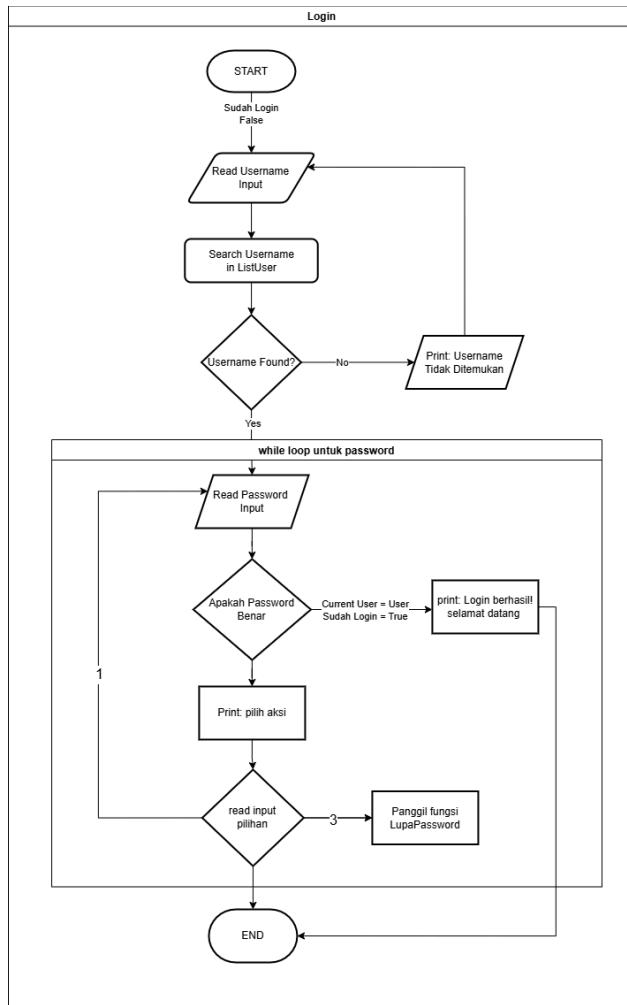
procedure PrintLinkedList(l : LinkedList)
{Menampilkan seluruh elemen linked list l dari head ke tail}

procedure FreeList(input/output l : LinkedList)
{Menghapus seluruh node dalam linked list dan membuat list
menjadi kosong}

```

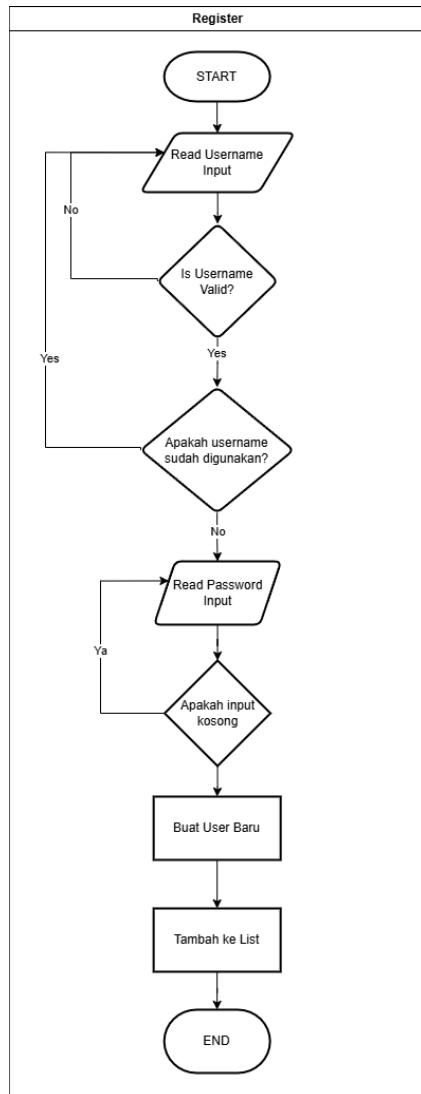
3.3 Desain Dekomposisi dan Algoritmik Program

1. F01 - Login



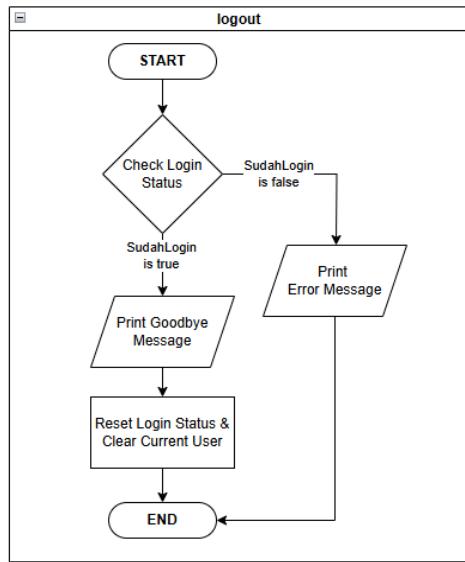
Gambar 1. *Flowchart* Fitur Login

2. F02 - Register Pasien



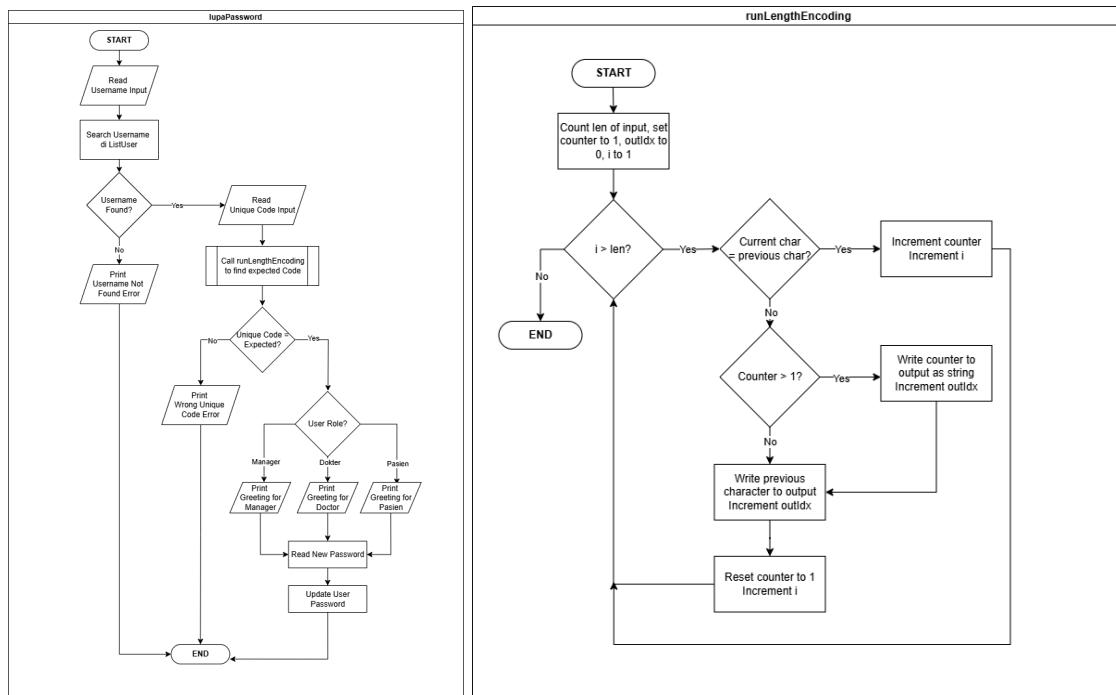
Gambar 2. Flowchart Fitur Register

3. F03 - Logout



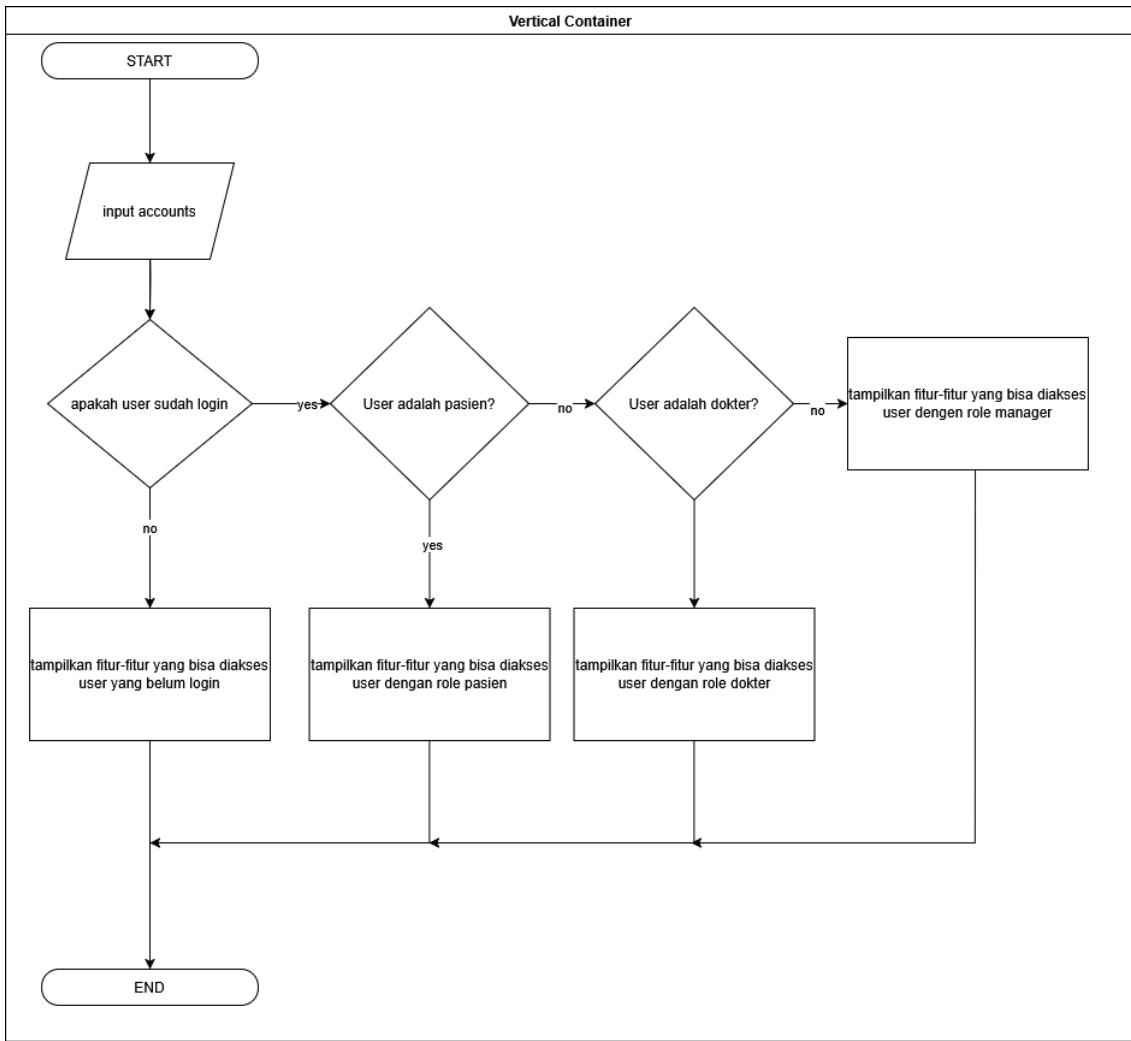
Gambar 3. Flowchart Fitur Logout

4. F04 - Lupa Password



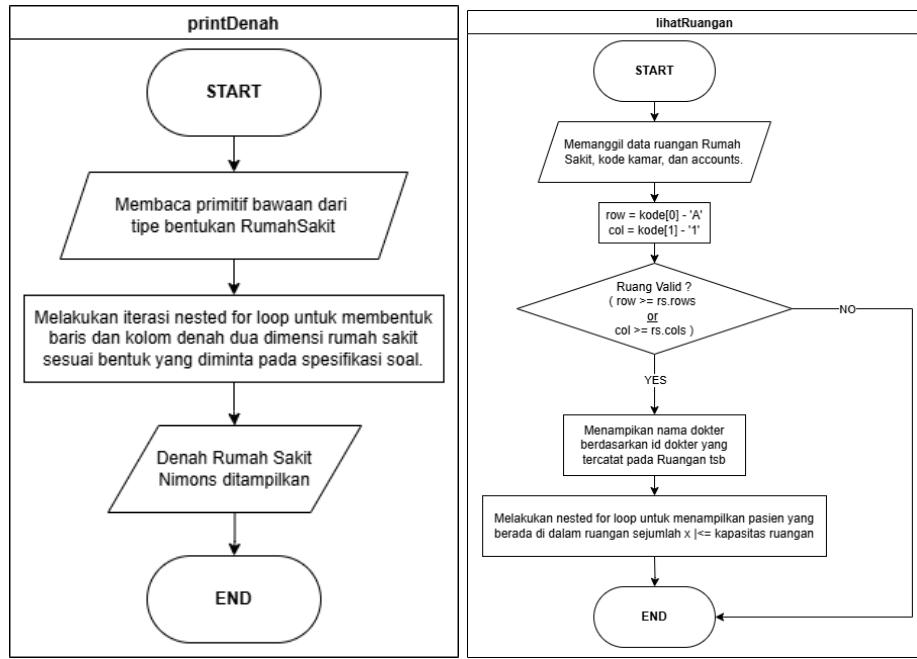
Gambar 4. Flowchart Fitur lupaPassword dan runLengthEncoding

5. F05 - Menu & Help



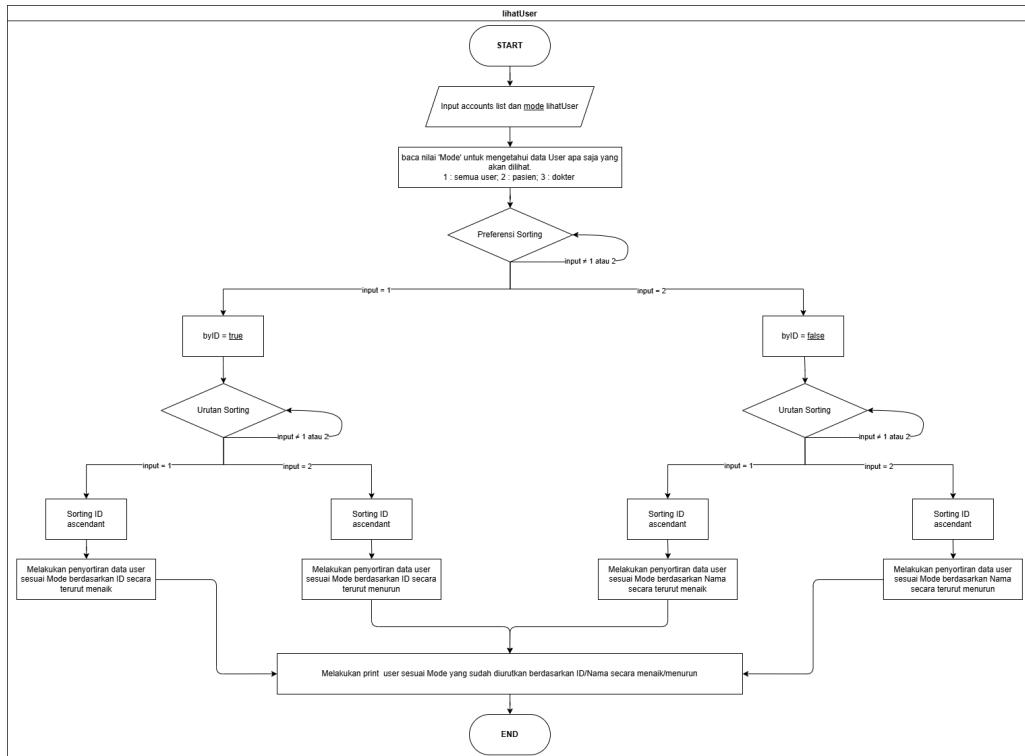
Gambar 5. *Flowchart* Fitur

6. F06 - Denah Rumah Sakit



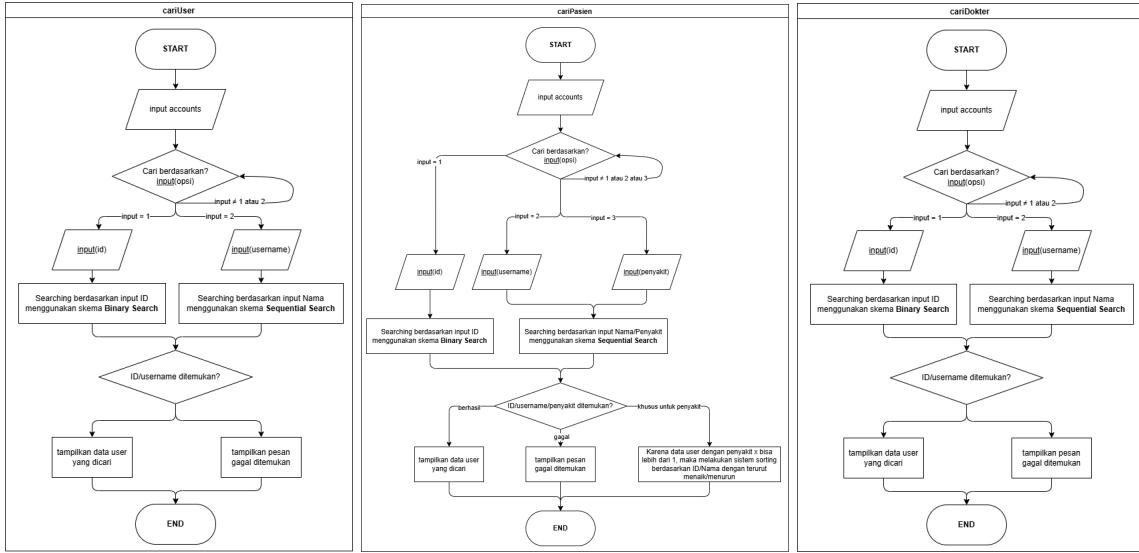
Gambar 6. *Flowchart* Fitur printDenah dan lihatRuangan

7. F07 - Lihat User



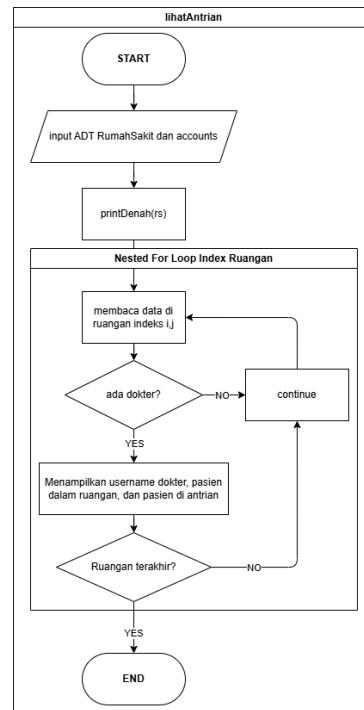
Gambar 7. *Flowchart* Fitur lihatUser

8. F08 - Cari User



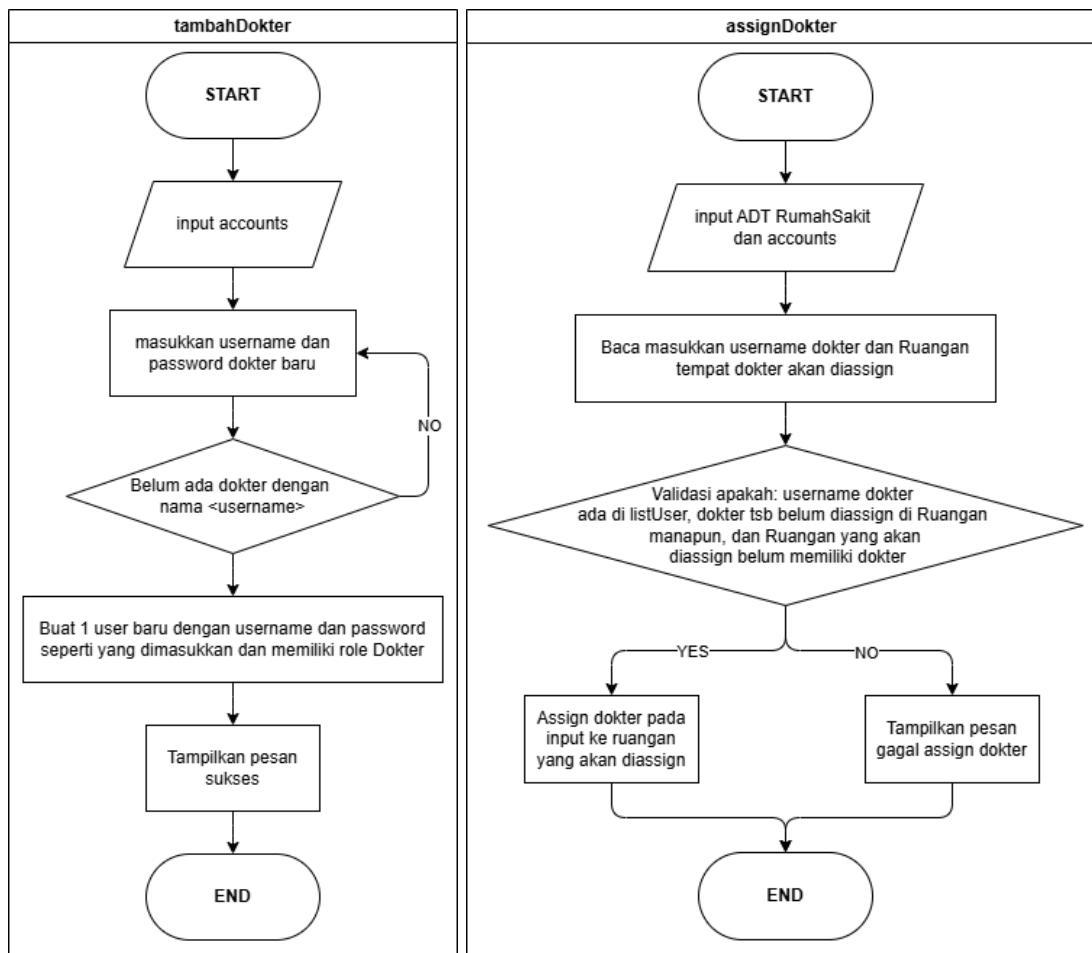
Gambar 8. Flowchart Fitur Cari User, Pasien, dan Dokter

9. F09 - Lihat Antrian



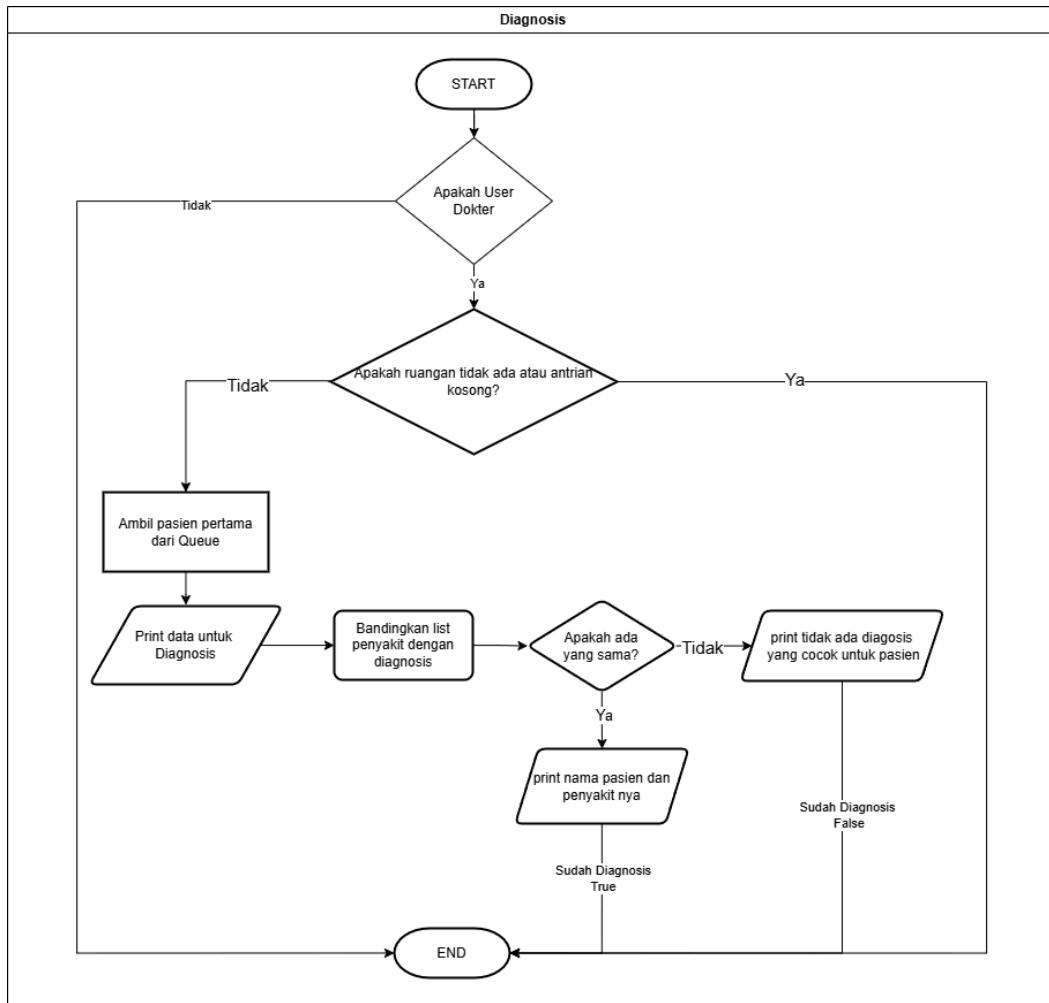
Gambar 9. Flowchart Fitur lihatAntrian

10. F10 - Tambah Dokter



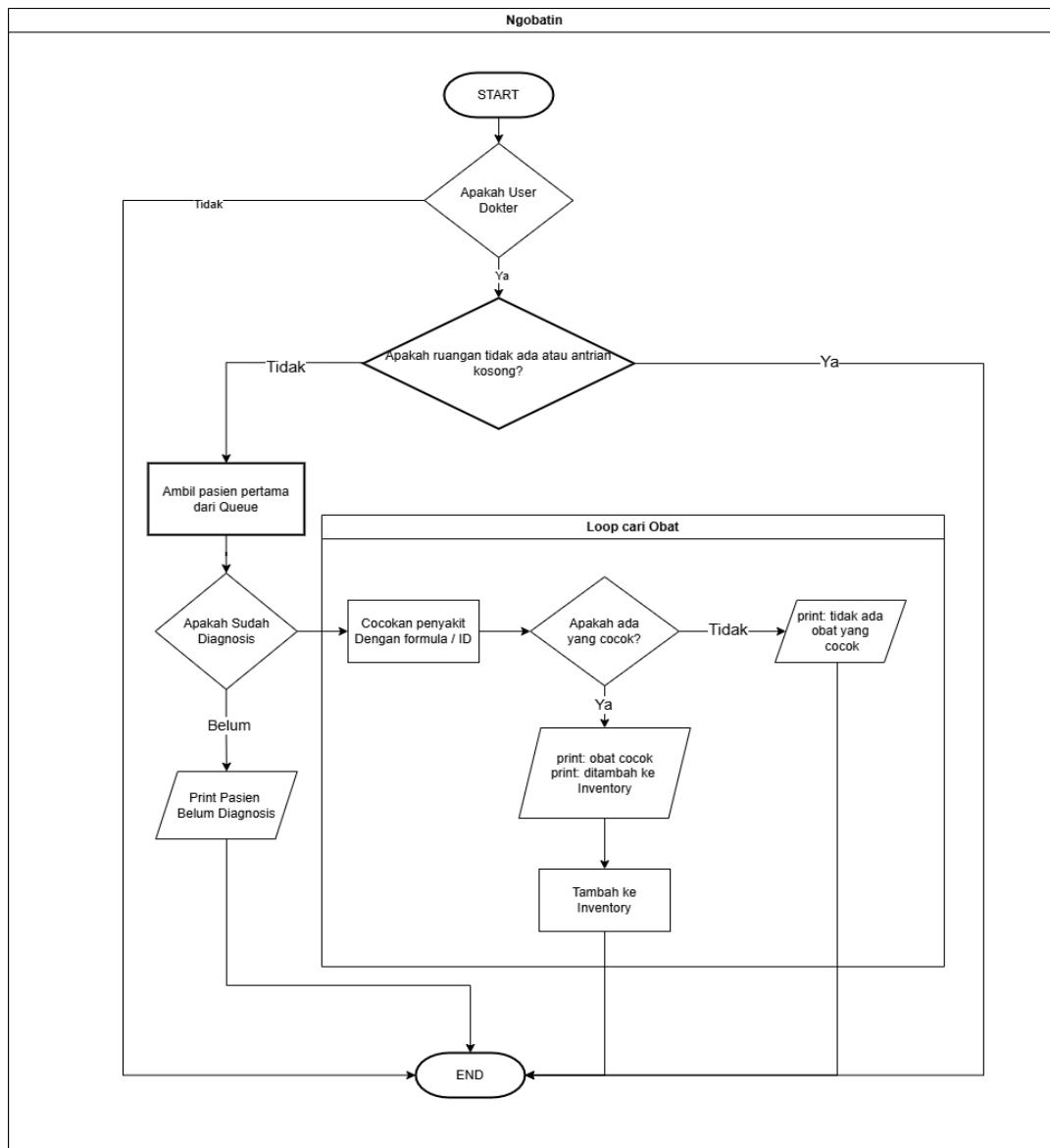
Gambar 10. *Flowchart* Fitur tambahDokter

11. F11 - Diagnosis



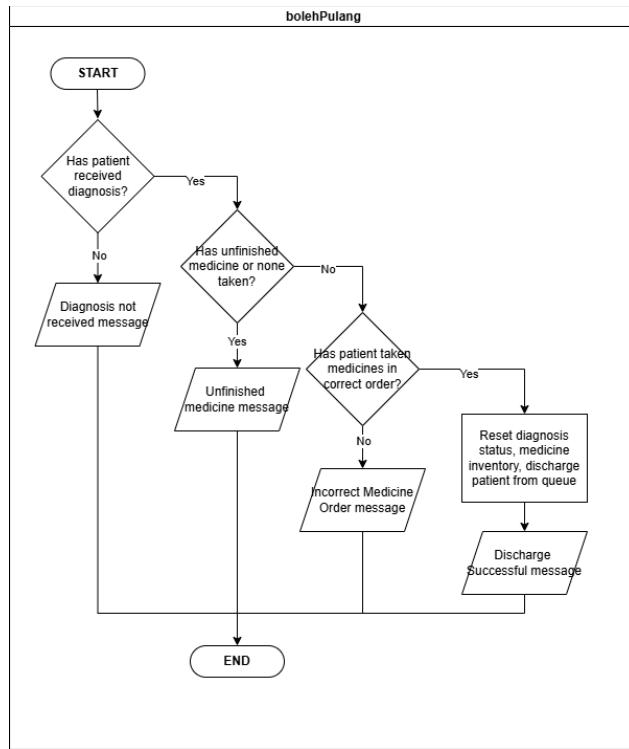
Gambar 11. *Flowchart* Fitur Diagnosis

12. F12 - Ngobatin



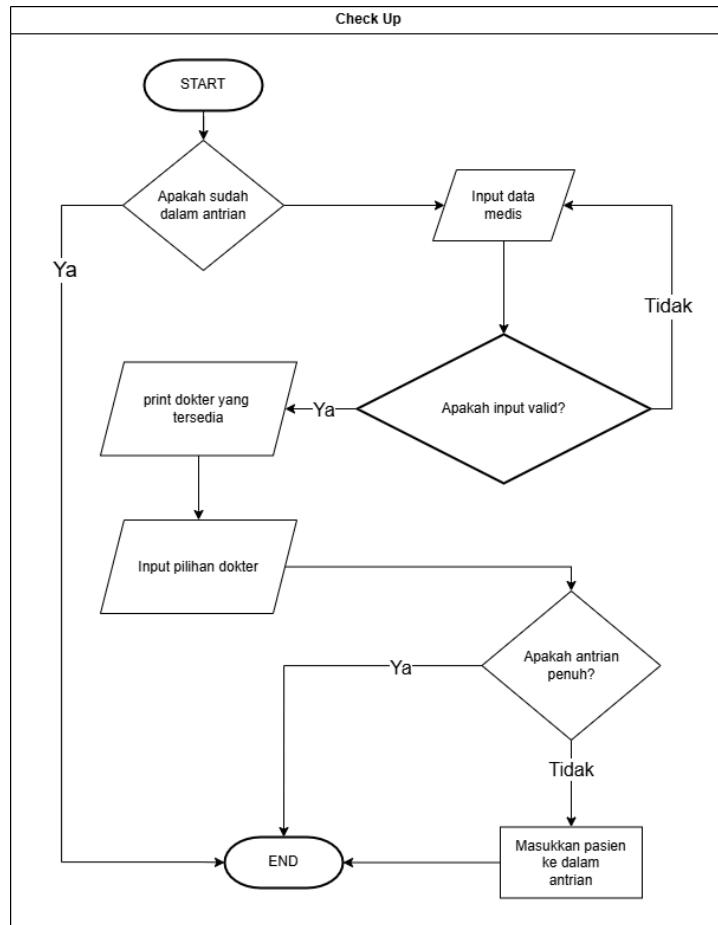
Gambar 12. *Flowchart* Fitur Ngobatin

13. F13 - Pulang Dok



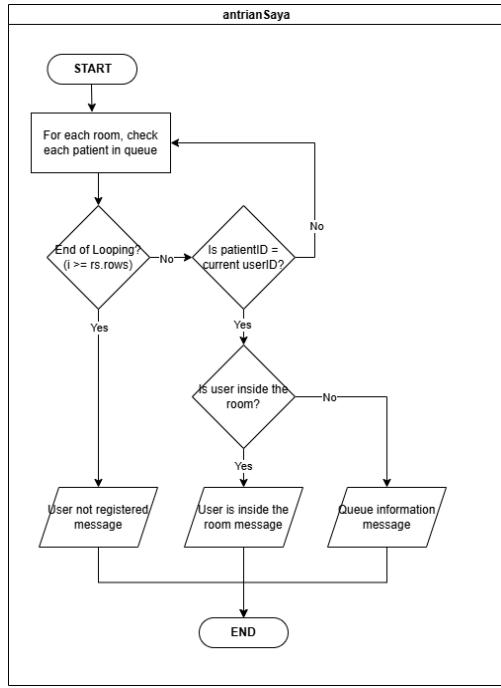
Gambar 13. *Flowchart* Fitur bolehPulang

14. F14 - Daftar Check Up



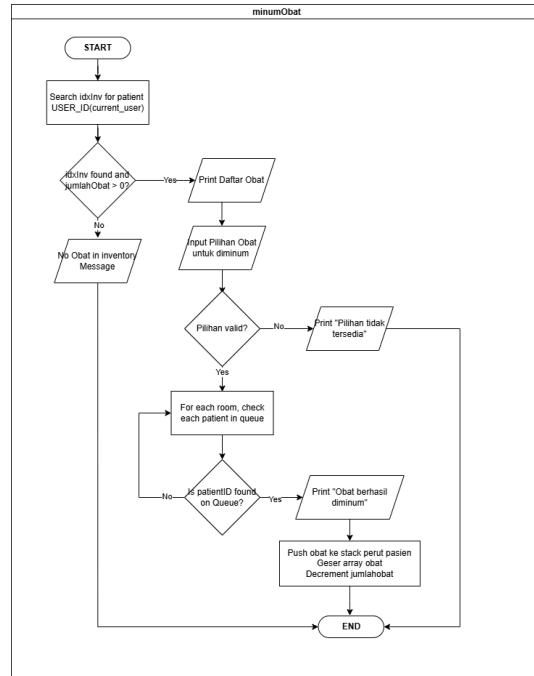
Gambar 14. *Flowchart* Fitur CheckUp

15. F15 - Antrian Saya



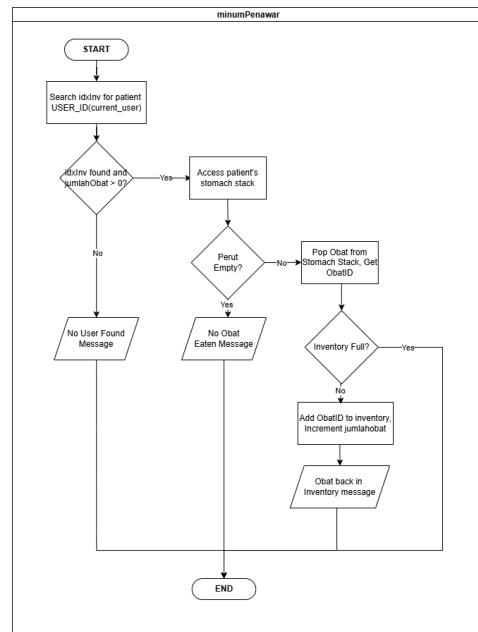
Gambar 15. *Flowchart* Fitur antrianSaya

16. F16 - Minum Obat



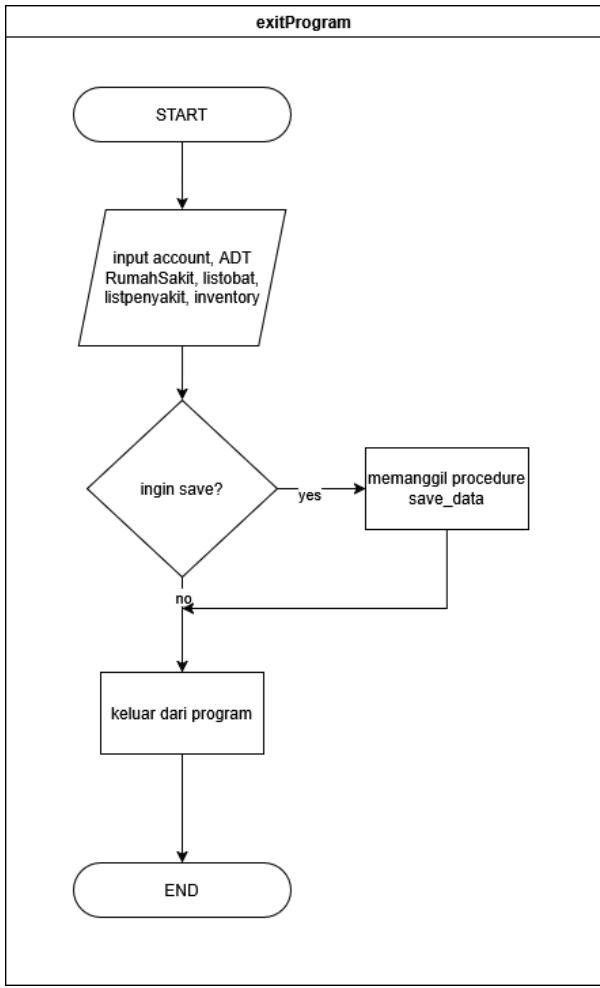
Gambar 16. *Flowchart* Fitur minumObat

17. F17 - Minum Penawar



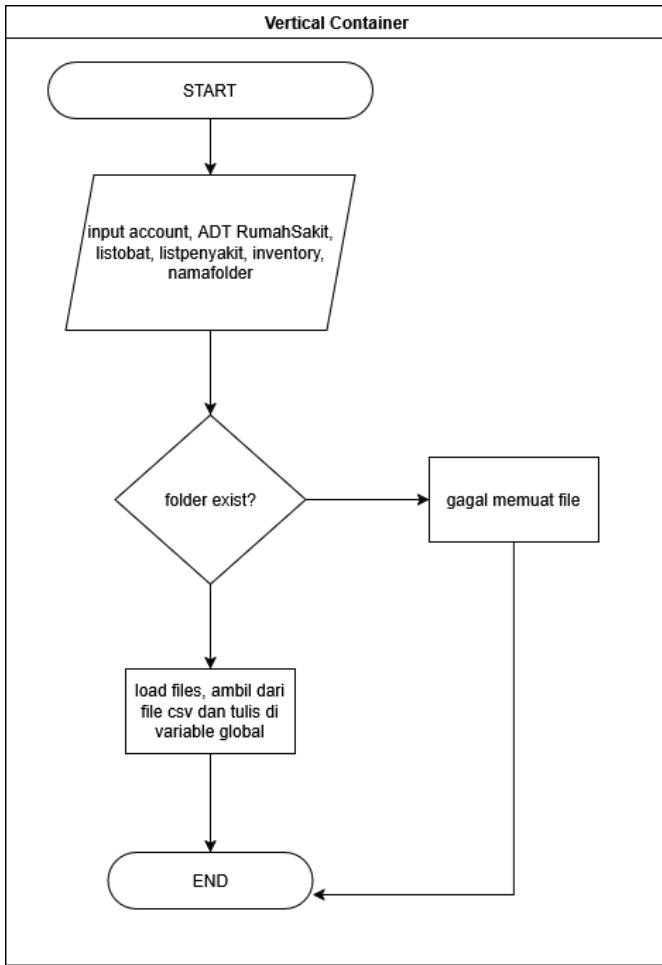
Gambar 17. *Flowchart* Fitur minumPenawar

18. F18 - Exit



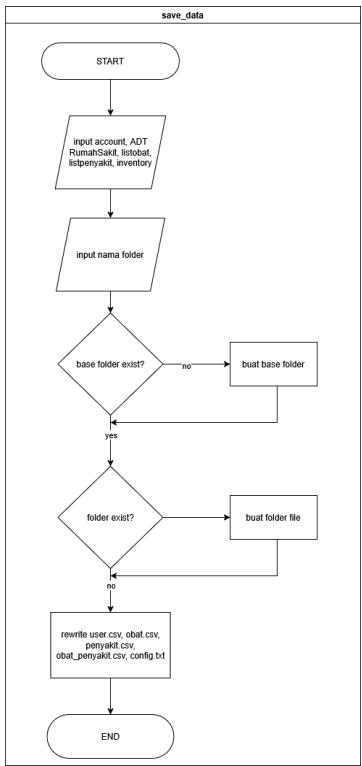
Gambar 18. *Flowchart exitProgram*

19. D03 - Load



Gambar 19. *Flowchart load_data*

20. D04 - Save



Gambar 20. Flowchart save_data

3.4 Spesifikasi Modul

1. Main Program



2. F01 - Login

```

procedure login (input users: ListUser, output current_user: User, output SudahLogin: boolean)
procedure login (input users: ListUser, output current_user: User, output SudahLogin: boolean)
{ I.S. users terdefinisi, bisa kosong atau berisi daftar pengguna. }
{ F.S. Jika login berhasil, current_user berisi data pengguna yang login dan SudahLogin bernilai true. Jika gagal atau pengguna memilih keluar, SudahLogin bernilai false. Pesan yang sesuai tercetak ke layar. }


```

KAMUS LOKAL

```

usernameInput, passwordInput: string
i, pilihan, idxUserDitemukan: integer
userDitemukan: boolean

```

```

prosesLoginBerlanjut: boolean

ALGORITMA
    SudahLogin ← false
    prosesLoginBerlanjut ← true

    output("Username: ")
    input(usernameInput)

    userDitemukan ← false
    idxUserDitemukan ← -1    i ← 0

    while (i < users.jumlahuser) and (not userDitemukan) do
        if (users.users[i].username = usernameInput) then
            userDitemukan ← true
            idxUserDitemukan ← i
        endif
        i ← i + 1

    if (not userDitemukan) then
        output("username tidak ditemukan!")
        prosesLoginBerlanjut ← false

    while (prosesLoginBerlanjut) and (not SudahLogin) do
        output("Password: ")
        input(passwordInput)

        if (users.users[idxUserDitemukan].password =
passwordInput) then
            current_user ← users.users[idxUserDitemukan]
            SudahLogin ← true
            output("Login berhasil! Selamat datang, ",
current_user.username, ".")
        else
            output("Password salah!")
            output("Pilih aksi:")
            output("1. Coba lagi")
            output("2. Keluar dari login", EOL)
            output("3. Lupa password", EOL)
            output("Masukkan pilihan (1/2/3): ")
            input(pilihan)

            if (pilihan = 1) then
                continue
            else if (pilihan = 2) then
                prosesLoginBerlanjut ← false
            else if (pilihan = 3) then
                lupaPassword(users)

        prosesLoginBerlanjut ← false

```

```

        output("Pilihan tidak valid. Keluar dari login.")
        prosesLoginBerlanjut ← false

    endif
endif
endwhile

```

3. F02 - Register Pasien

```

procedure register_pasien (input/output lUser: ListUser, output
new_user_data: User)

KAMUS LOKAL
usernameInput, passwordInput: string
i: integer
usernameSudahDigunakan: boolean
inputSukses: boolean

ALGORITMA
inputSukses ← input(usernameInput)
if (not inputSukses) then
    // Prosedur berakhir
else
    if (not is_username_valid(usernameInput)) then
        output("Format username tidak valid.")
    else
        inputSukses ← input(passwordInput)
        if (not inputSukses) then
        else
            usernameSudahDigunakan ← false
            i ← 0
            while (i < lUser.jumlahuser) and (not
usernameSudahDigunakan) do
                if compare_case_insensitive(lUser.users[i].username,
usernameInput) then
                    usernameSudahDigunakan ← true
                endif
                i ← i + 1
            endwhile

            if (usernameSudahDigunakan) then
                output("Username sudah digunakan.")
            else
                CreateUser(new_user_data)
                if (lUser.jumlahuser = 0) then
                    new_user_data.id ← 1
                else
                    new_user_data.id ← lUser.users[lUser.jumlahuser -

```

```

        1].id + 1
    endif
    new_user_data.username ← usernameInput
    new_user_data.password ← passwordInput
    new_user_data.role ← "pasien"

    InsUser(lUser, new_user_data)

    endif
endif
endif
endif
endif

```

4. F03 - Logout

```

procedure logout(input/output SudahLogin: boolean, input/output
current_user: User)
{ I.S. : SudahLogin dan current_user terdefinisi. }
{ F.S. : Jika SudahLogin bernilai true, maka SudahLogin
menjadi false dan current_user di-reset. Jika tidak, tampilkan
pesan kesalahan. }

KAMUS LOKAL
-
ALGORITMA
if (SudahLogin = true) then
    output("Sampai jumpa!")
    SudahLogin ← false
    CreateUser(current_user)
else
    output("Logout gagal!")
    output("Anda belum login, silahkan login terlebih dahulu
sebelum melakukan logout")

```

5. F04 - Lupa Password

```

procedure runLengthEncode(input/output: string)
{ I.S. : input terdefinisi, merupakan string karakter }
{ F.S. : output berisi representasi run-length encoding dari
input }

KAMUS LOKAL
len, count, outIdx, i: integer

```

ALGORITMA

```
len ← panjang(input)
count ← 1
outIdx ← 0

i traversal [1..len]
if (input[i] = input[i - 1]) then
    count ← count + 1
else
    if (count > 1) then
        output ← output + konversi(count)
        output ← output + input[i - 1]
        count ← 1
```

procedure lupaPassword(input/output LU: ListUser)

```
{ I.S. : LU berisi kumpulan user, termasuk username dan
password mereka }
{ F.S. : Jika username dan kode unik valid, password user
diminta ulang dan diperbarui }
```

KAMUS LOKAL

```
username, kodeUnik, expectedKode, newPassword: string
userIndex, i: integer
```

ALGORITMA

```
output("Username: ")
input(username)

userIndex ← -1
i ← 0
iterate
    if (LU.users[i].username = username) then
        userIndex ← i
    stop (LU.users[i].username ≠ username and i < LU.jumlahuser
- 1)
    i ← i + 1
{ i = userIndex or i = LU.jumlahuser }

if (userIndex = -1) then
    output("Username tidak terdaftar!")
else
    output("Kode Unik: ")
```

```

input(kodeUnik)
runLengthEncode(username, expectedKode)

if (kodeUnik ≠ expectedKode) then
    output("Kode unik salah!")
else
    output("Halo ")
    if (LU.users[userIndex].role = "Dokter") then
        output("Dokter ")
    else if (LU.users[userIndex].role = "Manager") then
        output("Manager ")
    else
        output("Pasien ")

    output(LU.users[userIndex].username, ", silakan
daftarkan ulang password anda!")
    output("Password Baru: ")
    input(newPassword)
    LU.users[userIndex].password ← newPassword

```

6. F05 - Menu & Help

```

procedure helpManager(input u: User)
{ Menampilkan bantuan untuk pengguna dengan role manager }

KAMUS LOKAL
ALGORITMA
    output("Halo manager ", u.username, ".")
    output("Kenapa kamu memanggil command HELP? Kan kamu
manager, tapi yasudahlah kamu pasti sedang kebingungan.")
    output("Berikut adalah hal-hal yang dapat kamu lakukan
sekarang:")
    output("1. LOGOUT : Keluar dari akun yang sedang
digunakan")
    output("2. LUPA_PASSWORD : Melakukan update password")
    output("3. LIHAT_DENAH : Melihat denah ruangan")
    output("4. LIHAT_RUANGAN <nama_ruang> : Melihat ruangan.
nama_ruang : A1, A2, dst")
    output("5. LIHAT_USER : Melihat data seluruh pengguna")
    output("6. CARI_USER : Mencari data pengguna secara
spesifik")
    output("7. LIHAT_SEMUA_ANTRIAN : Melihat seluruh rincian
di seluruh ruangan")
    output("8. TAMBAH_DOKTER : Menambahkan dokter baru")
    output("9. EXIT : Keluar dari program")

procedure helpPasien(input u: User)

```

```

{ Menampilkan bantuan untuk pengguna dengan role pasien }

KAMUS LOKAL
ALGORITMA
    output("Selamat datang, ", u.username, ". Kamu memanggil
command HELP. Kamu pasti sedang kebingungan.")
    output("Berikut adalah hal-hal yang dapat kamu lakukan
sekarang:")
        output("1. LOGOUT : Keluar dari akun yang sedang
digunakan")
        output("2. LUPA_PASSWORD : Melakukan update password")
        output("3. LIHAT_DENAH : Melihat denah ruangan")
        output("4. LIHAT_RUANGAN <nama_ruang> : Melihat ruangan.
nama_ruang : A1, A2, dst")
        output("5. PULANGDOK : Akyu boleh pulang ga, dok? >///<")
        output("6. DAFTAR_CHECKUP : Mendaftar untuk melakukan
check-up dengan dokter")
        output("7. ANTRIAN : Melihat status antrian")
        output("8. MINUM_OBAT : Minum obat")
        output("9. PENAWAR : Penawar khusus jika kamu salah minum
obat")
        output("10. EXIT : Keluar dari program")

procedure helpDokter(input u: User)
{ Menampilkan bantuan untuk pengguna dengan role dokter }

KAMUS LOKAL
ALGORITMA
    output("Halo Dokter ", u.username, ". Kamu memanggil
command HELP. Kamu pasti sedang kebingungan.")
    output("Berikut adalah hal-hal yang dapat kamu lakukan
sekarang:")
        output("1. LOGOUT : Keluar dari akun yang sedang
digunakan")
        output("2. LUPA_PASSWORD : Melakukan update password")
        output("3. LIHAT_DENAH : Melihat denah ruangan")
        output("4. LIHAT_RUANGAN <nama_ruang> : Melihat ruangan.
nama_ruang : A1, A2, dst")
        output("5. DIAGNOSIS : Auto-Diagnose-2.0 ; Diagnosis
secara otomatis!")
        output("6. NGOBATIN : Auto-Prescription-2.0 ; Sistem
peresepan obat")
        output("7. EXIT : Keluar dari program")

procedure notLogin()
{ Menampilkan bantuan ketika belum login }

KAMUS LOKAL
ALGORITMA
    output("Kamu belum login sebagai role apapun. Silahkan
login terlebih dahulu.")
    output("1. LOGIN : Masuk ke dalam akun yang sudah
terdaftar")

```

```

output("2. REGISTER : Membuat akun baru")

procedure templateHelp(input u: User)
{ Menampilkan bantuan sesuai role user }

KAMUS LOKAL
ALGORITMA
    output("===== HELP =====")
    output("")

    if u.role = "" then
        notLogin()
    else if u.role = "pasien" then
        helpPasien(u)
    else if u.role = "dokter" then
        helpDokter(u)
    else if u.role = "manager" then
        helpManager(u)

    output("Footnote:")
    output("1. Untuk menggunakan aplikasi, silahkan masukkan
nama fungsi yang terdaftar")
    output("2. Jangan lupa untuk memasukkan input yang valid")

```

7. F06 - Denah Rumah Sakit

```

procedure printDenah (input rs: RumahSakit)
{ I.S. rs terdefinisi }
{ F.S. Denah rumah sakit tercetak ke layar sesuai dengan
format yang ditentukan }

KAMUS LOKAL
    i, j: integer
    karakterBaris: character

ALGORITMA
    output("      ")
    j traversal [0..rs.cols-1]
        output(" ", j + 1, " ")
        output(EOL)

    i traversal [0..rs.rows-1]
        output(" +")
        j traversal [0..rs.cols-1]
            output("---+")
            output(EOL)

    karakterBaris ← KonversiKeKarakter('A' + i)

```

```

output(karakterBaris, " |")
j traversal [0..rs.cols-1]
    output(" ", karakterBaris, j + 1, " |")
output(EOL)

output(" +")
j traversal [0..rs.cols-1]
    output("---+")
output(EOL)

procedure lihatRuangan (input rs: RumahSakit, input kode:
string, input lUser: ListUser)
{ I.S. rs, kode, dan lUser terdefinisi. }
{ F.S. Detail ruangan yang sesuai dengan kode tercetak ke
layar. Jika kode tidak valid, pesan kesalahan tercetak. }

KAMUS LOKAL
row, col: integer
r: Ruangan
dokterName: string
i, j: integer
pasienName: string
len, printedCount, cap, idx, pasienId: integer

ALGORITMA
row ← KarakterKeinteger(kode[0]) - KarakterKeinteger('A')
col ← KarakterKeinteger(kode[1]) - KarakterKeinteger('1')

if (row >= rs.rows) or (col >= rs.cols) then
    output("Ruangan tidak valid.", EOL)
else
    r ← rs.data[row][col]
    output("--- Detail Ruangan ", kode, " ---", EOL)
    output("Kapasitas : ", rs.kapasitasPerRuang, EOL)

    // Tampilkan nama dokter
    if (r.dokterId = 0) then
        output("Dokter : -", EOL)
    else
        dokterName ← "-"
        i ← 0
        while (i < lUser.jumlahUser) and (dokterName = "-") do
            if (lUser.users[i].id = r.dokterId) then
                dokterName ← lUser.users[i].username
            endif
            i ← i + 1
            output("Dokter : Dr. ", dokterName, EOL)
        endif

    // Tampilkan pasien
    len ← lengthQueue(r.antrianPasienIds)

```

```

if (len = 0) then
    output("Pasien di dalam ruangan:", EOL)
    output(" Tidak ada pasien di dalam ruangan saat ini.", EOL)
else
    output("Pasien di dalam ruangan:", EOL)
    printedCount ← 0
    cap ← rs.kapasitasPerRuangan

    i ← 0
    while (i < len) and (printedCount < cap) do
        idx ← (r.antrianPasienIds.idxHead + i) mod
CAPACITY_QUEUE
        pasienId ← r.antrianPasienIds.buffer[idx]
        pasienName ← "-"
        j ← 0
        while (j < lUser.jumlahuser) and (pasienName = "-") do
            if (lUser.users[j].id = pasienId) then
                pasienName ← lUser.users[j].username
            endif
            j ← j + 1
        output(" ", printedCount + 1, ". ", pasienName, EOL)
        printedCount ← printedCount + 1
        i ← i + 1

        if (printedCount = 0) then
            output(" Tidak ada pasien di dalam ruangan saat ini.", EOL)
        endif
    endif

output("-----", EOL)
endif

```

8. F07 - Lihat User

```

procedure lihatUser (input lUser: ListUser, input mode: integer)
{ I.S. lUser terdefinisi dan berisi daftar pengguna, mode terdefinisi (0: semua, 1: pasien, 2: dokter). }
{ F.S. Daftar pengguna yang difilter dan diurutkan berdasarkan pilihan pengguna tercetak ke layar. }

```

KAMUS LOKAL

pilihanUrutById, pilihanOrder: integer

```

isById, isAscending: boolean
temp: array [0..MAX_FIELD-1] of User
count, i: integer
judulMode, judulUrut, judulOrder: string

ALGORITMA
output("Urutkan berdasarkan?", EOL)
output("1. ID", EOL, "2. Nama", EOL, ">> Pilihan: ")
while (pilihanUrutById ≠ 1) and (pilihanUrutById ≠ 2) do
    output("Masukan salah, pilih antara 1/2!", EOL, "Urutkan
berdasarkan?", EOL, "1. ID", EOL, "2. Nama", EOL, ">> Pilihan:
")
    if (pilihanUrutById = 1) then
        isById ← true
    else
        isById ← false
    endif

    output("Urutan sort?", EOL)
    output("1. ASC (A-Z)", EOL, "2. DESC (Z-A)", EOL, ">>
Pilihan: ")
    while (pilihanOrder ≠ 1) and (pilihanOrder ≠ 2) do
        output("Masukan salah, pilih antara 1/2!", EOL, "Urutan
sort?", EOL, "1. ASC (A-Z)", EOL, "2. DESC (Z-A)", EOL, ">>
Pilihan: ")
        if (pilihanOrder = 1) then
            isAscending ← true
        else
            isAscending ← false
        endif

        count ← 0
        i traversal [0..lUser.jumlahuser-1]
        if (mode = 0) or ((mode = 1) and (lUser.users[i].role =
"pasien")) or ((mode = 2) and (lUser.users[i].role =
"dokter")) then
            temp[count] ← lUser.users[i]
            count ← count + 1
        endif

        sortUsers(temp, count, isById, isAscending)

        if (mode = 0) then
            judulMode ← " semua pengguna"

```

```

else if (mode = 1) then
    judulMode ← "pasien"
else { mode = 2 }
    judulMode ← "dokter"
endif

if (isById) then
    judulUrut ← "dengan ID terurut "
else
    judulUrut ← "dengan nama terurut "
endif

if (isAscending) then
    judulOrder ← "ascending"
else
    judulOrder ← "descending"
endif
output("Menampilkan ", judulMode, " ", judulUrut,
judulOrder, "...", EOL)

displayHeader(mode)
i traversal [0..count-1]
displayUser(temp[i], mode)

```

9. F08 - Cari User

```

procedure cariUser (input lu: ListUser)
{ I.S. lu terdefinisi dan mungkin berisi daftar pengguna. }
{ F.S. Hasil pencarian pengguna berdasarkan ID atau nama
tercetak ke layar. Jika pengguna tidak ditemukan, pesan
kesalahan tercetak. }

KAMUS LOKAL
choice: integer
id: integer
idx: integer
name: string

ALGORITMA
output("Cari berdasarkan?", EOL)
output("1. ID", EOL, "2. Nama", EOL, ">>> Pilihan: ")
input(choice)

while (choice ≠ 1) and (choice ≠ 2) do
    output(EOL, "Masukan salah, pilih antara 1/2!", EOL, "Cari
berdasarkan?", EOL, "1. ID", EOL, "2. Nama", EOL, ">> Pilihan:
")
    input(choice)

```

```

if (choice = 1) then
    output(EOL, ">>> Masukkan nomor ID user: ")
    input(id)
    idx ← binarySearchById(lu.users, lu.jumlahuser, id)
{ Asumsikan binarySearchById mengembalikan indeks atau -1 jika
tidak ditemukan }

    if (idx = -1) then
        output(EOL, "Tidak ditemukan pengguna dengan ID ", id,
"!", EOL)
    else
        output(EOL, "Menampilkan pengguna dengan nomor ID ", id,
"...", EOL)
        displayHeader(0)
        displayUser(lu.users[idx], 0)
    endif
else { choice = 2 }
    output(EOL, ">>> Masukkan nama user: ")
    input(name)
    idx ← sequentialSearchByName(lu.users, lu.jumlahuser,
name)
{ Asumsikan sequentialSearchByName mengembalikan indeks atau
-1 jika tidak ditemukan }

    if (idx = -1) then
        output(EOL, "Tidak ditemukan pengguna dengan nama ",
name, "!", EOL)
    else
        output(EOL, "Menampilkan pengguna dengan nama ", name,
"...", EOL)
        displayHeader(0)
        displayUser(lu.users[idx], 0)
    endif
endif

```

```

procedure cariPasien (input lu: ListUser)
{ I.S. lu terdefinisi dan mungkin berisi daftar pengguna. }
{ F.S. Hasil pencarian pasien berdasarkan ID, nama, atau
penyakit tercetak ke layar. Jika pasien tidak ditemukan, pesan
kesalahan tercetak. }

```

KAMUS LOKAL

```

choice: integer
id: integer
idx: integer
disease: string
name: string
results: array [0..MAX_FIELD-1] of integer
temp: array [0..MAX_FIELD-1] of User

```

```

count, i: integer
pilihanSortBy, pilihanOrder: integer
isSortById, isSortAscending: boolean
judulUrut, judulOrder: string

ALGORITMA
output("Cari berdasarkan?", EOL)
output("1. ID", EOL, "2. Nama", EOL, "3. Penyakit", EOL,
">>> Pilihan: ")
input(choice)

while (choice ≠ 1) and (choice ≠ 2) and (choice ≠ 3) do
    output(EOL, "Masukan salah, pilih antara 1/2/3!", EOL,
"Cari berdasarkan?", EOL, "1. ID", EOL, "2. Nama", EOL, "3.
Penyakit", EOL, ">> Pilihan: ")
    input(choice)

if (choice = 3) then
    output(EOL, ">>> Masukkan nama penyakit: ")
    input(disease)
    count ← searchPatientsByDisease(lu.users, lu.jumlahuser,
disease, results)
{ Asumsikan fungsi ini mengisi array 'results' dengan indeks
dan mengembalikan jumlahnya }

if (count = 0) then
    output(EOL, "Tidak ditemukan pasien dengan penyakit ",
disease, "!", EOL)
else
    output(EOL, "Urutkan berdasarkan?", EOL, "1. ID", EOL,
"2. Nama", EOL, ">>> Pilihan: ")
    input(pilihanSortBy)

while (pilihanSortBy ≠ 1) and (pilihanSortBy ≠ 2) do
    output("Masukan salah, pilih antara 1/2!", EOL,
"Urutkan berdasarkan?", EOL, "1. ID", EOL, "2. Nama", EOL,
">>> Pilihan: ")
    input(pilihanSortBy)

output(EOL, "Urutan sort?", EOL, "1. ASC (A-Z)", EOL,
"2. DESC (Z-A)", EOL, ">>> Pilihan: ")
input(pilihanOrder)

while (pilihanOrder ≠ 1) and (pilihanOrder ≠ 2) do
    output("Masukan salah, pilih antara 1/2!", EOL,
"Urutan sort?", EOL, "1. ASC (A-Z)", EOL, "2. DESC (Z-A)",
EOL, ">>> Pilihan: ")
    input(pilihanOrder)

if (pilihanSortBy = 1) then

```

```

        isSortById ← true
else
        isSortById ← false
endif
if (pilihanOrder = 1) then
        isSortAscending ← true
else
        isSortAscending ← false
endif

        i traversal [0..count-1]
        temp[i] ← lu.users[results[i]]

        sortUsers(temp, count, isSortById, isSortAscending)

        if (isSortById) then
            judulUrut ← "dengan ID terurut "
        else
            judulUrut ← "dengan nama terurut "
        endif
        if (isSortAscending) then
            judulOrder ← "ascending"
        else
            judulOrder ← "descending"
        endif
        output(EOL, "Menampilkan pasien dengan penyakit ", disease, " ", judulUrut, judulOrder, "...", EOL)

        displayHeader(1)
        i traversal [0..count-1]
        displayUser(temp[i], 1)
endif

else if (choice = 1) then
    output(EOL, ">>> Masukkan nomor ID pasien: ")
    input(id)
    idx ← binarySearchById(lu.users, lu.jumlahuser, id)

    if (idx = -1) or (lu.users[idx].role ≠ "pasien") then
        output(EOL, "Tidak ditemukan pasien dengan ID ", id, "!", EOL)
    else
        output(EOL, "Menampilkan pasien dengan ID ", id, "...", EOL)
        displayHeader(1)
        displayUser(lu.users[idx], 1)
endif

else if (choice = 2) then
    output(EOL, ">>> Masukkan nama pasien: ")

```

```

input(name)
idx ← sequentialSearchByName(lu.users, lu.jumlahuser,
name)

if (idx = -1) or (lu.users[idx].role ≠ "pasien") then {
Perbandingan string case-sensitive }
    output(EOL, "Tidak ditemukan pasien dengan nama ", name,
"!", EOL)
else
    output(EOL, "Menampilkan pasien dengan nama ", name,
"...", EOL)
    displayHeader(1)
    displayUser(lu.users[idx], 1)
endif
endif

```

procedure cariDokter (input lu: ListUser)
{ I.S. lu terdefinisi dan mungkin berisi daftar pengguna. }
{ F.S. Hasil pencarian dokter berdasarkan ID atau nama
tercetak ke layar. Jika dokter tidak ditemukan atau pengguna
yang ditemukan bukan dokter, pesan kesalahan tercetak. }

KAMUS LOKAL

```

choice: integer
id: integer
idx: integer
name: string

```

ALGORITMA

```

output("Cari berdasarkan?", EOL)
output("1. ID", EOL, "2. Nama", EOL, ">>> Pilihan: ")
input(choice)

while (choice ≠ 1) and (choice ≠ 2) do
    output(EOL, "Masukan salah, pilih antara 1/2!", EOL, "Cari
berdasarkan?", EOL, "1. ID", EOL, "2. Nama", EOL, ">> Pilihan:
")
    input(choice)

if (choice = 1) then
    output(EOL, ">>> Masukkan nomor ID dokter: ")
    input(id)
    idx ← binarySearchById(lu.users, lu.jumlahuser, id)
    if (idx = -1) or (lu.users[idx].role ≠ "dokter") then
        output(EOL, "Tidak ditemukan dokter dengan ID ", id,
"!", EOL)
    else
        output(EOL, "Menampilkan dokter dengan ID ", id, "...",
EOL)
        displayHeader(2)

```

```

        displayUser(lu.users[idx], 2)
    endif
else if (choice = 2) then
    output(EOL, ">>> Masukkan nama dokter: ")
    input(name)
    idx ← sequentialSearchByName(lu.users, lu.jumlahuser,
name)

    if (idx = -1) or (lu.users[idx].role ≠ "dokter") then
        Perbandingan string case-sensitive. Kode C asli menggunakan
        strcasecmp untuk case-insensitive
        output(EOL, "Tidak ditemukan dokter dengan nama ", name,
        "!", EOL)
    else
        output(EOL, "Menampilkan dokter dengan nama ", name,
        "...", EOL)
        displayHeader(2)
        displayUser(lu.users[idx], 2)
    endif
endif

```

10. F09 - Lihat Antrian

```

procedure lihatSemuaAntrian (input rs: RumahSakit, input lUser:
ListUser)
{ I.S. rs dan lUser terdefinisi. }
{ F.S. Denah rumah sakit dan detail antrian untuk setiap
ruangan yang memiliki dokter tercetak ke layar. }

KAMUS LOKAL
i, j, u, k, l: integer
r: Ruangan
namaDokter, namaPasien: string
len, dalamRuangan, dalamAntrian, index, idPasien: integer
karakterBaris: character

ALGORITMA
printDenah(rs)

i traversal [0..rs.rows-1]
j traversal [0..rs.cols-1]
r ← rs.data[i][j]
if (r.dokterId ≠ 0) then
    karakterBaris ← KonversiKeKarakter('A' + i)
    output(EOL, "===== ", karakterBaris, j + 1, "
=====", EOL)
    output("Kapasitas : ", rs.kapasitasPerRuangan, EOL)

```

```

namaDokter ← "-"
u ← 0
while (u < lUser.jumlahuser) and (namaDokter = "-") do
    if (lUser.users[u].id = r.dokterId) then
        namaDokter ← lUser.users[u].username
    endif
    u ← u + 1
    output("Dokter      : Dr. ", namaDokter, EOL)

len ← lengthQueue(r.antrianPasienIds)
if (len < rs.kapasitasPerRuangan) then
    dalamRuangan ← len
else
    dalamRuangan ← rs.kapasitasPerRuangan
endif
dalamAntrian ← len - dalamRuangan

output("Pasien di dalam ruangan:", EOL)
if (dalamRuangan = 0) then
    output(" Tidak ada pasien di dalam ruangan saat
ini.", EOL)
else
    index ← IDX_HEAD(r.antrianPasienIds)
    k traversal [0..dalamRuangan-1]
    idPasien ← r.antrianPasienIds.buffer[index]
    namaPasien ← "-"
    l ← 0
    while(l < lUser.jumlahuser)and(namaPasien = "-")do
        if (lUser.users[l].id = idPasien) then
            namaPasien ← lUser.users[l].username
        endif
        l ← l + 1
    output(" ", k + 1, ". ", namaPasien, EOL)
    index ← (index + 1) mod CAPACITY_QUEUE
endif

output("Pasien di antrian:", EOL)
if (dalamAntrian = 0) then
    output(" Tidak ada pasien di antrian saat ini.",
EOL)
else
    index ← (IDX_HEAD(r.antrianPasienIds) +
dalamRuangan) mod CAPACITY_QUEUE
    k traversal [0..dalamAntrian-1]
    idPasien ← r.antrianPasienIds.buffer[index]
    namaPasien ← "-"
    l ← 0
    while(l < lUser.jumlahuser)and(namaPasien = "-")do
        if (lUser.users[l].id = idPasien) then
            namaPasien ← lUser.users[l].username

```

```

        endif
        l ← l + 1
        output(" ", k + 1, ". ", namaPasien, EOL)
        index ← (index + 1) mod CAPACITY_QUEUE
    endif
endif
endfor {j}
endfor {i}

```

11. F10 - Tambah Dokter

```

procedure tambahDokter (input/output lUser: ListUser)
{ I.S. lUser terdefinisi, bisa kosong atau sudah berisi
pengguna. }
{ F.S. Jika username belum digunakan oleh dokter lain dan
ListUser belum penuh, seorang dokter baru ditambahkan ke lUser
dan jumlahuser bertambah 1. Pesan keberhasilan atau kegagalan
(jika username sudah ada) tercetak ke layar. }

KAMUS LOKAL
username, password: string
i: integer
usernameSudahAda: boolean
newUser: User

ALGORITMA
output("Username: ")
input(username)
output("Password: ")
input(password)

usernameSudahAda ← false
i ← 0
while (i < lUser.jumlahuser) and ((not)usernameSudahAda) do
    if (lUser.users[i].role = "dokter") and
(lUser.users[i].username = username) then
        usernameSudahAda ← true
    endif
    i ← i + 1

if (usernameSudahAda) then
    output("Sudah ada Dokter bernama ", username, "!", EOL)
else
    lUser.jumlahuser ← lUser.jumlahuser + 1
    CreateUser(newUser)
    newUser.id ← lUser.jumlahuser
    newUser.username ← username
    newUser.password ← password

```

```

newUser.role ← "dokter"

lUser.users[lUser.jumlahuser-1] ← newUser

output("Dokter ", username, " berhasil ditambahkan!", EOL)
endif

```

```

procedure assignDokter (input/output rs: RumahSakit, input
lUser: ListUser)
{ I.S. rs dan lUser terdefinisi. }
{ F.S. Dokter dengan username yang diinput diassign ke ruangan
yang diinput jika kondisi terpenuhi. Pesan yang sesuai
tercetak. }

```

KAMUS LOKAL

```

username: string
ruanganInput: string
dokterId: integer
i, j, k: integer
row, col: integer
dokterDitemukanDiSistem: boolean
dokterSudahAdaDiRuanganLain: boolean
ruanganTargetSudahTerisi: boolean
pesanError: string
karakterBaris: character
namaDokterPenghuni: string
idxRuanganDokterLama: Point

```

ALGORITMA

```

output("Username: ")
input(username)
output("Ruangan: ")
input(ruanganInput)

dokterId ← -1
dokterDitemukanDiSistem ← false
i ← 0
while (i < lUser.jumlahuser) and ((not)
dokterDitemukanDiSistem) do
    if (lUser.users[i].username = username) and
(lUser.users[i].role = "dokter") then
        dokterId ← lUser.users[i].id
        dokterDitemukanDiSistem ← true
    endif
    i ← i + 1

if ((not) dokterDitemukanDiSistem) then
    output("Dokter dengan username ", username, " tidak
ditemukan.", EOL)
else

```

```

row ← KarakterKeinteger(ruanganInput[0]) -
KarakterKeinteger('A')
col ← StringKeinteger(Substring(ruanganInput, 1,
PanjangString(ruanganInput)-1)) - 1

if (row < 0) or (row >= rs.rows) or (col < 0) or (col >=
rs.cols) then
    output("Ruangan tidak valid.", EOL)
else
    dokterSudahAdaDiRuanganLain ← false
    i ← 0
    while (i < rs.rows) and (not
dokterSudahAdaDiRuanganLain) do
        j ← 0
        while (j < rs.cols) and (not
dokterSudahAdaDiRuanganLain) do
            if (rs.data[i][j].dokterId = dokterId) then
                dokterSudahAdaDiRuanganLain ← true
                idxRuanganDokterLama.baris ← i
                idxRuanganDokterLama.kolom ← j
            endif
            j ← j + 1
        endwhile
        i ← i + 1
    endwhile

    if (dokterSudahAdaDiRuanganLain) then
        karakterBaris ← KonversiKeKarakter('A' +
idxRuanganDokterLama.baris)
        if (row = idxRuanganDokterLama.baris) and (col =
idxRuanganDokterLama.kolom) then
            output("Dokter ", username, " sudah berada di
ruangan ", karakterBaris, idxRuanganDokterLama.kolom + 1, ".", ,
EOL)
        else { Dokter ada di ruangan lain, dan ruangan target
berbeda }
            output("Dokter ", username, " sudah diassign ke
ruangan ", karakterBaris, idxRuanganDokterLama.kolom + 1, "!", ,
EOL)
        if (rs.data[row][col].dokterId ≠ 0) then
            namaDokterPenghuni ← "-"
            k ← 0
            while (k < lUser.jumlahuser) and
(namaDokterPenghuni = "-") do
                if (lUser.users[k].id =
rs.data[row][col].dokterId) then
                    namaDokterPenghuni ←
lUser.users[k].username
                endif
            k ← k + 1
        
```

```

        karakterBaris ← KonversiKeKarakter('A' + row)
        output("Ruangan ", karakterBaris, col + 1, "
sudah ditempati oleh Dokter ", namaDokterPenghuni, ".", EOL)
        endif
        output("Silakan kosongkan ruangan dokter ",
username, " sebelumnya atau pilih ruangan lain.", EOL)
        endif
else
    if (rs.data[row][col].dokterId ≠ 0) then
        namaDokterPenghuni ← "-"
        k ← 0
        while (k < lUser.jumlahuser) and (namaDokterPenghuni
= "-") do
            if (lUser.users[k].id =
rs.data[row][col].dokterId) then
                namaDokterPenghuni ← lUser.users[k].username
            endif
            k ← k + 1
            karakterBaris ← KonversiKeKarakter('A' + row)
            output("Ruangan ", karakterBaris, col + 1, " sudah
ditempati oleh Dokter ", namaDokterPenghuni, "!", EOL)
            output("Silakan pilih ruangan lain untuk Dokter ",
username, ".", EOL)
            else
                rs.data[row][col].dokterId ← dokterId
                karakterBaris ← KonversiKeKarakter('A' + row)
                output("Dokter ", username, " berhasil diassign ke
ruangan ", karakterBaris, col + 1, "!", EOL)
            endif
        endif
    endif
endif

```

12. F11 - Diagnosis

```

function cariRuanganDokter (input rs: RumahSakit, input
dokterIdCari: integer, output ruanganHasil: Ruangan) ->
boolean
KAMUS LOKAL
    i, j: integer
ALGORITMA
    i ← 0
    while (i < rs.rows) do
        j ← 0
        while (j < rs.cols) do
            if (rs.data[i][j].dokterId = dokterIdCari) then
                ruanganHasil ← rs.data[i][j]
            -> true

```

```

        endif
        j ← j + 1
    endwhile
    i ← i + 1
endwhile
-> false

procedure diagnosis (input currentUser: User, input rs:
RumahSakit, input/output lUser: ListUser, input lPenyakit:
ListPenyakit)
KAMUS LOKAL
    i, j, k: integer
    pasienIdCari: integer
    ruangDokter: Ruangan
    pasienSaatIni: User
    penyakitDiperiksa: Penyakit
    isRuanganDitemukan: boolean
    isPasienDitemukan: boolean
    isDiagnosisCocok: boolean
ALGORITMA
    if (currentUser.role ≠ "Dokter") then
        // Prosedur berakhir
    else
        isRuanganDitemukan ← cariRuanganDokter(rs, currentUser.id,
ruangDokter)
        if (not isRuanganDitemukan) or
(isEmptyQueue(ruangDokter.antrianPasienIds)) then
            // Prosedur berakhir
        else
            pasienIdCari ←
ruangDokter.antrianPasienIds.buffer[ruangDokter.antrianPasienI
ds.idxHead]

            isPasienDitemukan ← false
            i ← 0
            while (i < lUser.jumlahuser) and (not isPasienDitemukan)
do
            if (lUser.users[i].id = pasienIdCari) then
                pasienSaatIni ← lUser.users[i]
                isPasienDitemukan ← true
            endif
            i ← i + 1
        endwhile

        if (not isPasienDitemukan) then
            // Prosedur berakhir
        else
            isDiagnosisCocok ← false
            k ← 0
            while (k < lPenyakit.jumlahPenyakit) and (not

```

```

isDiagnosisCocok) do
    penyakitDiperiksa ← lPenyakit.listData[k]
    if (pasienSaatIni.suhuTubuh >=
penyakitDiperiksa.suhuMin) and (pasienSaatIni.suhuTubuh <=
penyakitDiperiksa.suhuMax) and
        (pasienSaatIni.tekananSistolik >=
penyakitDiperiksa.sysMin) and (pasienSaatIni.tekananSistolik
<= penyakitDiperiksa.sysMax) and
            (pasienSaatIni.tekananDiastolik >=
penyakitDiperiksa.diasMin) and (pasienSaatIni.tekananDiastolik
<= penyakitDiperiksa.diasMax) and
                (pasienSaatIni.detakJantung >=
penyakitDiperiksa.detakMin) and (pasienSaatIni.detakJantung <=
penyakitDiperiksa.detakMax) and
                    (pasienSaatIni.oksigen >=
penyakitDiperiksa.saturasiMin) and (pasienSaatIni.oksigen <=
penyakitDiperiksa.saturasiMax) and
                        (pasienSaatIni.gulaDarah >=
penyakitDiperiksa.gulaMin) and (pasienSaatIni.gulaDarah <=
penyakitDiperiksa.gulaMax) and
                            (pasienSaatIni.beratBadan >=
penyakitDiperiksa.beratMin) and (pasienSaatIni.beratBadan <=
penyakitDiperiksa.beratMax) and
                                (pasienSaatIni.tinggiBadan >=
penyakitDiperiksa.tinggiMin) and (pasienSaatIni.tinggiBadan <=
penyakitDiperiksa.tinggiMax) and
                                    (pasienSaatIni.kolesterol >=
penyakitDiperiksa.kolesterolMin) and (pasienSaatIni.kolesterol
<= penyakitDiperiksa.kolesterolMax) and
                                        (pasienSaatIni.trombosit >=
penyakitDiperiksa.trombositMin) and (pasienSaatIni.trombosit
<= penyakitDiperiksa.trombositMax) then

    lUser.users[indeksPasien]
        j ← 0
    while (j < lUser.jumlahuser) do
        if (lUser.users[j].id = pasienIdCari) then
            lUser.users[j].riwayatPenyakit ←
penyakitDiperiksa.namaPenyakit
                j ← lUser.jumlahuser
            endif
            j ← j + 1
    endwhile

    SudahDiagnosis[pasienIdCari] ← true
    output("Pasien ", pasienSaatIni.username, "
di-diagnosiskan dengan penyakit: ",
penyakitDiperiksa.namaPenyakit)
        isDiagnosisCocok ← true
    endif

```

```

        k ← k + 1
    endwhile

    if (not isDiagnosisCocok) then
        j ← 0
        while (j < lUser.jumlahuser) do
            if (lUser.users[j].id = pasienIdCari) then
                lUser.users[j].riwayatPenyakit ← "-"
                j ← lUser.jumlahuser
            endif
            j ← j + 1
        endwhile

        SudahDiagnosis[pasienIdCari] ← false
        output("Tidak ada diagnosis yang cocok untuk
pasien!")
    endif
endif
endif
endif

```

13. F12 - Ngobatin

```

procedure TambahObatKeInventory (input/output inv: Inventory,
input pasienId: integer, input id_obat: integer)
{ I.S. inv, pasienId, id_obat terdefinisi.
  F.S. id_obat ditambahkan ke daftar obat pasienId dalam inv.
  Jika pasien belum ada di inv, entri baru dibuat.
  Jika daftar obat pasien sudah penuh (MAX_OBAT), obat
  tidak ditambahkan. }

KAMUS LOKAL
i: integer
pasienDitemukan: boolean

ALGORITMA
    pasienDitemukan ← false
    i ← 0
    while (i < inv.jumlahPasienwObat) and (not pasienDitemukan)
do
    if (inv.data[i].id = pasienId) then
        pasienDitemukan ← true
        if (inv.data[i].jumlahobat < MAX_OBAT) then
            inv.data[i].obat[inv.data[i].jumlahobat] ← id_obat
            inv.data[i].jumlahobat ← inv.data[i].jumlahobat + 1
        endif
    endif
    i ← i + 1

```

```

if (not pasienDitemukan) then
    if (inv.jumlahPasienwObat < KAPASITAS_INVENTORY_PASIEN)
then
    inv.data[inv.jumlahPasienwObat].id ← pasienId
    inv.data[inv.jumlahPasienwObat].jumlahobat ← 1
    inv.data[inv.jumlahPasienwObat].obat[0] ← id_obat
    inv.jumlahPasienwObat ← inv.jumlahPasienwObat + 1
endif
endif'

procedure ngobatin (input current_user: User, input rs:
RumahSakit, input lUser: ListUser, input lObat: ListObat,
input lFormula: ListFormula, input lPenyakit: ListPenyakit,
 inv: Inventory)
{ I.S. Semua parameter input terdefinisi. Array global
SudahDiagnosis terdefinisi.
    F.S. Jika dokter valid, ada pasien, dan sudah didiagnosis,
maka obat yang sesuai diberikan
        dan ditambahkan ke inventory. Pesan yang relevan
tercetak. }

KAMUS LOKAL
hasilCariRuang: HasilCariRuangan
ruangDokter: Ruangan
pasienId: integer
pasienSaatIni: User
count: integer
i, j: integer
f: Formula
p: Penyakit
o: Obat

ALGORITMA
if (current_user.role ≠ "dokter") then
else
    hasilCariRuang ← cariRuanganDokter(rs, current_user.id)
    if (not hasilCariRuang.ditemukan) then
        endif

    else
        ruangDokter ←
rs.data[hasilCariRuang.baris][hasilCariRuang.kolom]
        if (isEmptyQueue(ruangDokter.antrianPasienIds))
            endif
        else
            pasienId ←
ruangDokter.antrianPasienIds.buffer[ruangDokter.antrianPasienI
ds.idxHead]
            pasienSaatIni ← GetUser(lUser, pasienId)

```

```

        output("Dokter sedang mengobati pasien!", EOL)

        if (SudahDiagnosis[current_user.id] = false) then
        else
            count ← 1
            i traversal [0..lFormula.jumlah_formula-1]
            f ← lFormula.formulas[i]
            j traversal [0..lPenyakit.jumlah_penyakit-1]
            p ← lPenyakit.penyakit[j]
            if (p.id = f.penyakit_id) and (p.nama =
pasienSaatIni.riwayat_penyakit) then
                o ← GetObat(lObat, f.obat_id)
                TambahObatKeInventory(inv, pasienId,
f.obat_id)
                count ← count + 1
            endif
            endfor {j}
            endfor {i}
        endif
        endif
    endif
endif

```

14. F13 - Aku boleh pulang ga, dok?

```

procedure bolehPulang(input current_user: User, input/output
inv: Inventory, input lObat: ListObat, input lFormula:
ListFormula, input lPenyakit: ListPenyakit, input/output rs:
RumahSakit)
{ I.S. : current_user, inv, rs, dan seluruh data master
(lObat, lFormula, lPenyakit) terdefinisi. }
{ F.S. : Jika syarat terpenuhi, maka pasien keluar dari sistem
dan status di-reset; jika tidak, tampilkan pesan alasan tidak
bisa pulang. }

```

KAMUS LOKAL

```

pasienId, idxInv, i, j, k, top, kapasitas, len, idx: integer
penyakitPasien: string
obatSeharusnya: ListStatik
urutanBenar: boolean
o: Obat
f: Formula
p: Penyakit
salinanPerut, tampilPerut: Stack
q: Queue

```

ALGORITMA

```

pasienId ← USER_ID(current_user)

```

```

if (SudahDiagnosis[pasienId] = false) then
    output("Kamu belum menerima diagnosis apapun dari dokter,
jangan buru-buru pulang!")
else
    idxInv ← -1
    i ← 0
    iterate
        if (inv.data[i].id = pasienId) then
            idxInv ← i
        stop (inv.data[i].id ≠ pasienId and i <
inv.jumlahPasienwObat - 1)
        i ← i + 1
    { idxInv = i or i = inv.jumlahPasienwObat }

    if (idxInv = -1) then
        output("Dokter sedang memeriksa keadaanmu...")
        output("Masih ada obat yang belum kamu habiskan, minum
semuanya dulu yukk!")
    else
        if (inv.data[idxInv].jumlahobat > 0 or
IsEmptyStack(inv.data[idxInv].perut)) then
            output("Dokter sedang memeriksa keadaanmu...")
            output("Masih ada obat yang belum kamu habiskan, minum
semuanya dulu yukk!")
        else
            penyakitPasien ← RIWAYAT_PENYAKIT(current_user)
            CreateListStatik(obatSeharusnya)

            i traversal [0..JUMLAH_FORMULA(lFormula) - 1]
            f ← FORMULA_LIST(lFormula, i)

            j ← 0
            iterate
                p ← PENYAKIT_LIST(lPenyakit, j)
                if (ID_PENYAKIT(p) = f.penyakit_id and
NAMA_PENYAKIT(p) = penyakitPasien) then
                    insertLastList(obatSeharusnya, f.obat_id)
                stop (not (ID_PENYAKIT(p) = f.penyakit_id and
NAMA_PENYAKIT(p) = penyakitPasien) and j <
JUMLAH_PENYAKIT(lPenyakit) - 1)
                j ← j + 1

                urutanBenar ← true
                salinanPerut ← inv.data[idxInv].perut

                if (listLength(obatSeharusnya) ≠ salinanPerut.TOP + 1)
then
                urutanBenar ← false
            else

```

```

i ← listLength(obatSeharusnya) - 1
while (i ≥ 0 and urutanBenar) do
    Pop(salinanPerut, top)
    if (top ≠ ELMT_LIST(obatSeharusnya, i)) then
        urutanBenar ← false
    i ← i - 1
    { i < 0 or urutanBenar = false }

if (not urutanBenar) then
    output("Dokter sedang memeriksa keadaanmu...")
    output("Maaf, tapi kamu masih belum bisa pulang!")
    output("Urutan peminuman obat yang diharapkan:")

i traversal [0..listLength(obatSeharusnya) - 1]
o ← GetObat(lObat, ELMT_LIST(obatSeharusnya, i))
output(NAMA_OBAT(o))
if (i ≠ listLength(obatSeharusnya) - 1) then
    output(" -> ")

output("")
output("Urutan obat yang kamu minum:")
tampilPerut ← inv.data[idxInv].perut

while (not IsEmptyStack(tampilPerut)) do
    Pop(tampilPerut, top)
    o ← GetObat(lObat, top)
    output(NAMA_OBAT(o))
    if (not IsEmptyStack(tampilPerut)) then
        output(" -> ")
    output("")

output("Silahkan kunjungi dokter untuk meminta
penawar yang sesuai!")
else
    output("Dokter sedang memeriksa keadaanmu...")
    output("Selamat! Kamu sudah dinyatakan sembuh oleh
dokter. Silahkan pulang dan semoga sehat selalu!")

SudahDiagnosis[pasienId] ← false
inv.data[idxInv].jumlahobat ← 0
CreateEmptyStack(inv.data[idxInv].perut)

i traversal [0..rs.rows - 1]
j traversal [0..rs.cols - 1]
q ← rs.data[i][j].antrianPasienIds
kapasitas ← rs.kapasitasPerRuang
len ← lengthQueue(q)

k ← 0
while (k < kapasitas and k < len) do

```

```

    idx ← (q.idxHead + k) mod CAPACITY_QUEUE
    if (q.buffer[idx] = pasienId and k = 0) then
        dequeue(q, top)
    k ← k + 1

```

15. F14 - Daftar Check-Up

```

procedure DaftarCheckUp(input/output listUser: ListUser,
input/output rs: RumahSakit, input username: string,
input/output pasienCheckUpList: ListPasienCheckUp)
{ I.S. listUser, rs, username, pasienCheckUpList terdefinisi.
}
{ F.S. Pasien dengan 'username' mendaftar untuk check-up. Data
medis pasien diisi/diperbarui.
    Pasien dimasukkan ke antrian dokter yang dipilih jika
valid dan belum antri.
    pasienCheckUpList diperbarui. Pesan yang sesuai
tercetak. }

KAMUS LOKAL
idxPasien, i, j, k: integer
pasienDitemukan: boolean
pilihan, current, dokterIdDipilih, rowRuangDipilih,
colRuangDipilih: integer
selesaiCariDokter: boolean
antrianTujuan: Queue

ALGORITMA
    pasienDitemukan ← false
    idxPasien ← -1 { Atau indeks yang menandakan tidak ditemukan
}
    i ← 0
    while (i < listUser.jumlahuser) and (not pasienDitemukan) do
        if (listUser.users[i].username = username) then
            idxPasien ← i
            pasienDitemukan ← true
        endif
        i ← i + 1
    endwhile

    if (not pasienDitemukan) then
        output("Pasien dengan username ", username, " tidak
ditemukan.")
    else
        { Input Data Medis - idxPasien merujuk ke pasien yang
datanya akan diisi }
        output("Masukkan suhu tubuh (32.0 - 42.0): ")
        repeat
            input(listUser.users[idxPasien].suhu_tubuh)

```

```

        if (listUser.users[idxPasien].suhu_tubuh < 32.0) or
(listUser.users[idxPasien].suhu_tubuh > 42.0) then
            output("Suhu tidak valid. Masukkan ulang: ")
        endif
        until (listUser.users[idxPasien].suhu_tubuh >= 32.0) and
(listUser.users[idxPasien].suhu_tubuh <= 42.0)

        output("Masukkan tekanan darah (sistolik diastolik): ")
        repeat
            input(listUser.users[idxPasien].tekanan_darah_sistolik,
listUser.users[idxPasien].tekanan_darah_diastolik)
            if (listUser.users[idxPasien].tekanan_darah_sistolik <=
80) or (listUser.users[idxPasien].tekanan_darah_diastolik <=
50) or (listUser.users[idxPasien].tekanan_darah_sistolik <
listUser.users[idxPasien].tekanan_darah_diastolik) then
                output("Tekanan darah tidak valid. Masukkan ulang
(sistolik diastolik): ")
            endif
        until
not((listUser.users[idxPasien].tekanan_darah_sistolik <= 80)
or (listUser.users[idxPasien].tekanan_darah_diastolik <= 50)
or (listUser.users[idxPasien].tekanan_darah_sistolik <
listUser.users[idxPasien].tekanan_darah_diastolik))

        output("Masukkan detak jantung (61 - 159): ")
        repeat
            input(listUser.users[idxPasien].detak_jantung)
            if (listUser.users[idxPasien].detak_jantung <= 60) or
(listUser.users[idxPasien].detak_jantung >= 160) then
                output("Detak jantung tidak valid. Masukkan ulang: ")
            endif
        until (listUser.users[idxPasien].detak_jantung > 60) and
(listUser.users[idxPasien].detak_jantung < 160)

        output("Masukkan saturasi oksigen (80.0 - 100.0): ")
        repeat
            input(listUser.users[idxPasien].saturasi_oksigen)
            if (listUser.users[idxPasien].saturasi_oksigen < 80.0)
or (listUser.users[idxPasien].saturasi_oksigen > 100.0) then
                output("Saturasi oksigen tidak valid. Masukkan ulang:
")
            endif
        until (listUser.users[idxPasien].saturasi_oksigen >= 80.0)
and (listUser.users[idxPasien].saturasi_oksigen <= 100.0)

        output("Masukkan kadar gula darah (71 - 139): ")
        repeat
            input(listUser.users[idxPasien].kadar_gula_darah)
            if (listUser.users[idxPasien].kadar_gula_darah <= 70) or
(listUser.users[idxPasien].kadar_gula_darah >= 140) then

```

```

        output("Kadar gula tidak valid. Masukkan ulang: ")
    endif
    until (listUser.users[idxPasien].kadar_gula_darah > 70)
and (listUser.users[idxPasien].kadar_gula_darah < 140)

    output("Masukkan berat badan (11.0 - 199.0): ")
repeat
    input(listUser.users[idxPasien].berat_badan)
    if (listUser.users[idxPasien].berat_badan <= 10.0) or
(listUser.users[idxPasien].berat_badan >= 200.0) then
        output("Berat badan tidak valid. Masukkan ulang: ")
    endif
    until (listUser.users[idxPasien].berat_badan > 10.0) and
(listUser.users[idxPasien].berat_badan < 200.0)

    output("Masukkan tinggi badan (51 - 249): ")
repeat
    input(listUser.users[idxPasien].tinggi_badan)
    if (listUser.users[idxPasien].tinggi_badan <= 50) or
(listUser.users[idxPasien].tinggi_badan >= 250) then
        output("Tinggi badan tidak valid. Masukkan ulang: ")
    endif
    until (listUser.users[idxPasien].tinggi_badan > 50) and
(listUser.users[idxPasien].tinggi_badan < 250)

    output("Masukkan kadar kolesterol (101 - 299): ")
repeat
    input(listUser.users[idxPasien].kadar_kolesterol)
    if (listUser.users[idxPasien].kadar_kolesterol <= 100)
or (listUser.users[idxPasien].kadar_kolesterol >= 300) then
        output("Kolesterol tidak valid. Masukkan ulang: ")
    endif
    until (listUser.users[idxPasien].kadar_kolesterol > 100)
and (listUser.users[idxPasien].kadar_kolesterol < 300)

    output("Masukkan kadar trombosit (101 - 499): ")
repeat
    input(listUser.users[idxPasien].trombosit)
    if (listUser.users[idxPasien].trombosit <= 100) or
(listUser.users[idxPasien].trombosit >= 500) then
        output("Trombosit tidak valid. Masukkan ulang: ")
    endif
    until (listUser.users[idxPasien].trombosit > 100) and
(listUser.users[idxPasien].trombosit < 500)

    tampilanDokterTersedia(listUser, rs)
    output("pilih dokter yang tersedia: ")
    input(pilihan)

    current ← 1

```

```

dokterIdDipilih ← -1
rowRuanganDipilih ← -1
colRuanganDipilih ← -1
selesaiCariDokter ← false
i ← 0
while (i < rs.rows) and (not selesaiCariDokter) do
    j ← 0
    while (j < rs.cols) and (not selesaiCariDokter) do
        if (rs.data[i][j].dokterId ≠ 0) then { Hanya hitung
ruangan yang ada dokternya }
            if (current = pilihan) then
                dokterIdDipilih ← rs.data[i][j].dokterId
                rowRuanganDipilih ← i
                colRuanganDipilih ← j
                selesaiCariDokter ← true
            endif
            current ← current + 1
        endif
        j ← j + 1
    endwhile
    i ← i + 1
endwhile

if (dokterIdDipilih ≠ -1) then { Artinya dokter dengan
pilihan tersebut ditemukan }
    antrianTujuan ←
    rs.data[rowRuanganDipilih][colRuanganDipilih].antrianPasienIds
    if (isPasienSudahAntri(antrianTujuan,
listUser.users[idxPasien].id)) then
        output("Anda sudah terdaftar dalam antrian check-up!
Silakan selesaikan check-up yang sudah terdaftar terlebih
dahulu.")
    else

enqueue(rs.data[rowRuanganDipilih][colRuanganDipilih].antrianP
asienIds, listUser.users[idxPasien].id)
        insertAtTail(pasienCheckUpList,
listUser.users[idxPasien].id)
        output("Pasien berhasil masuk ke antrian check up.")
    endif
else
    output("pilihan anda tidak valid")
endif
endif

```

16. F15 - Antrian Saya!

```
procedure antrianSaya(input current_user: User, input rs:
```

```

RumahSakit)
{ I.S. : current_user dan rs terdefinisi }
{ F.S. : Menampilkan informasi posisi pasien dalam antrian
jika ditemukan, atau pesan jika belum terdaftar }

KAMUS LOKAL
i, j, k, len, idx, pasienId: integer
ruang: Ruangan
q: Queue
found: boolean

ALGORITMA
found ← false

i ← 0
while (i < rs.rows and not found) do
    j ← 0
    while (j < rs.cols and not found) do
        ruang ← rs.data[i][j]
        q ← ruang.antrianPasienIds
        len ← lengthQueue(q)

        k ← 0
        while (k < len and not found) do
            idx ← (q.idxHead + k) mod CAPACITY_QUEUE
            pasienId ← q.buffer[idx]

            if (pasienId = current_user.id) then
                found ← true
                if (k < rs.kapasitasPerRuang) then
                    output("Anda sedang berada di dalam ruangan
dokter!")
                else
                    output("Status antrian Anda:")
                    output("Dokter: ID ", ruang.dokterId)
                    output("Ruang: ", chr('A' + i), j + 1)
                    output("Posisi antrian: ", k + 1, " dari ", len)
            else
                k ← k + 1
            j ← j + 1
        i ← i + 1

        if (not found) then
            output("Anda belum terdaftar dalam antrian check-up!")
            output("Silakan daftar terlebih dahulu dengan command
DAFTAR_CHECKUP.")

```

```

procedure minumObat(input current_user: User, input/output inv:
Inventory, input rs: RumahSakit, input lObat: ListObat)
{ I.S. : current_user, inv, rs, dan lObat terdefinisi }
{ F.S. : Jika pasien memiliki obat dan memilih dengan benar,
maka obat diminum, masuk ke stack perut, dan dihapus dari
inventory }

KAMUS LOKAL
pasienId, idxInv, i, j, k, x: integer
obatId, pilihan: integer
o: Obat
q: Queue
idx: integer

ALGORITMA
pasienId ← USER_ID(current_user)
idxInv ← -1

i ← 0
iterate
  if (inv.data[i].id = pasienId) then
    idxInv ← i
  stop (inv.data[i].id ≠ pasienId and i <
inv.jumlahPasienwObat - 1)
  i ← i + 1
  { i = idxInv or i = inv.jumlahPasienwObat }

  if (idxInv = -1 or inv.data[idxInv].jumlahobat = 0) then
    output("Tidak ada obat dalam inventory!")
  else
    output("===== DAFTAR OBAT =====")
    i traversal [0..inv.data[idxInv].jumlahobat - 1]
    obatId ← inv.data[idxInv].obat[i]
    o ← GetObat(lObat, obatId)
    output(i + 1, ". ", NAMA_OBAT(o))

    output("Pilih obat untuk diminum: ")
    input(pilihan)

    if (pilihan < 1 or pilihan > inv.data[idxInv].jumlahobat)
then
    output("Pilihan nomor tidak tersedia!")
  else
    obatId ← inv.data[idxInv].obat[pilihan - 1]
    o ← GetObat(lObat, obatId)

    i ← 0
    iterate
      j ← 0
      iterate

```

```

q ← rs.data[i][j].antrianPasienIds
k ← 0
iterate
    idx ← (q.idxHead + k) mod CAPACITY_QUEUE
    if (q.buffer[idx] = pasienId) then
        Push(inv.data[idxInv].perut, obatId)
        output("GLEKGLEKGLEK...", NAMA_OBAT(o), "
berhasil diminum!!!")

    x traversal [pilihan -
1..inv.data[idxInv].jumlahhobat - 2]
        inv.data[idxInv].obat[x] ←
inv.data[idxInv].obat[x + 1]
        inv.data[idxInv].jumlahhobat ←
inv.data[idxInv].jumlahhobat - 1
        stop (q.buffer[idx] ≠ pasienId and k <
lengthQueue(q) - 1)
        k ← k + 1
        stop (j < rs.cols - 1)
        j ← j + 1
        stop (i < rs.rows - 1)
        i ← i + 1

```

18. F17 - Minum Penawar

```

procedure minumPenawar(input current_user: User, input/output
inv: Inventory, input lObat: ListObat)
{ I.S. : current_user terdefinisi, inv dan lObat terdefinisi }
{ F.S. : Obat paling atas dikeluarkan dari perut pasien (jika ada) dan dikembalikan ke inventory }

KAMUS LOKAL
    pasienId, idxInv, i, obatId: integer
    perut: Stack
    o: Obat

ALGORITMA
    pasienId ← USER_ID(current_user)
    idxInv ← -1

    i ← 0
    iterate
        if (inv.data[i].id = pasienId) then
            idxInv ← i
        stop (inv.data[i].id ≠ pasienId and i <
inv.jumlahPasienwObat - 1)
            i ← i + 1
        { i = idxInv or i = inv.jumlahPasienwObat }

```

```

if (idxInv = -1) then
    output("Pasien tidak ditemukan.")
else
    perut ← inv.data[idxInv].perut

if (IsEmptyStack(perut)) then
    output("Perut kosong!! Belum ada obat yang dimakan.")
else
    Pop(perut, obatId)
    o ← GetObat(lObat, obatId)

if (inv.data[idxInv].jumlahobat < MAX_OBAT) then
    inv.data[idxInv].obat[inv.data[idxInv].jumlahobat] ←
    obatId
    inv.data[idxInv].jumlahobat ←
    inv.data[idxInv].jumlahobat + 1

output("Uwekk!!! " , NAMA_OBAT(o) , " keluar dan
kembali ke inventory")

```

19. F18 - Exit

```

procedure exitProgram(input listuser: ListUser, input listobat:
ListObat, input listpenyakit: ListPenyakit, input listformula:
ListFormula, input rs: RumahSakit, input inventory: Inventory)
{ I.S. Data sudah termuat di dalam program, mungkin telah
diubah }
{ F.S. Program berhenti, data disimpan ke file jika pengguna
memilih "y" }

KAMUS LOKAL
jawaban: string

ALGORITMA
    output("Apakah Anda mau melakukan penyimpanan file yang
    sudah diubah? (y/n) ")
    input(jawaban)

    if (jawaban = "y") or (jawaban = "Y") then
        save_data(listuser, listobat, listpenyakit,
        listformula, rs, inventory)

    exit(0)

```

20. D03 - Load

```

function load_data(folder: string, listuser: ListUser,
listobat: ListObat, listpenyakit: ListPenyakit, listformula:

```

```
ListFormula, rs: RumahSakit, inventory: Inventory) → integer
{ Meload seluruh data dari folder eksternal ke dalam program }
{ Mengembalikan 1 jika sukses, 0 jika gagal }
```

KAMUS LOKAL

```
user_csv_path, obat_csv_path, penyakit_csv_path: string
obatpenyakit_csv_path, config_path: string
i: integer
hasManager: boolean
maxAntrianLuar: integer
```

ALGORITMA

```
if not folder_exists(folder) then
    output("Folder \"", folder, "\" tidak ditemukan.")
    →0

    output("Loading...")

    user_csv_path ← folder + SEPARATOR + "user.csv"
    obat_csv_path ← folder + SEPARATOR + "obat.csv"
    penyakit_csv_path ← folder + SEPARATOR + "penyakit.csv"
    obatpenyakit_csv_path ← folder + SEPARATOR +
    "obat_penyakit.csv"
    config_path ← folder + SEPARATOR + "config.txt"

    if not (file_exists(user_csv_path) and
    file_exists(obat_csv_path) and
        file_exists(penyakit_csv_path) and
    file_exists(obatpenyakit_csv_path) and
        file_exists(config_path)) then
        output("File yang diperlukan tidak lengkap dalam
folder \"", folder, "\").")
        →0
    parse_user_csv(user_csv_path, listuser)
    parse_obat_csv(obat_csv_path, listobat)
    parse_penyakit_csv(penyakit_csv_path, listpenyakit)
    parse_obatpenyakit_csv(obatpenyakit_csv_path, listformula)

    if not loadConfig(config_path, rs, inventory,
maxAntrianLuar) then
        output("Gagal membaca config.txt")
        return 0

    hasManager ← false
    for i ← 0 to listuser.jumlahuser -1 do
        if listuser.users[i].role = "manager" then
            hasManager ← true
            exit loop

    if not hasManager then
```

```

    output("File user.csv harus memiliki setidaknya satu
akun manager.")
    return 0

    output("Selamat datang kembali di rumah sakit Nimons !")
    return 1

```

21. D04 - Save

```

procedure save_data(input listuser: ListUser, input listobat:
ListObat, input listpenyakit: ListPenyakit, input listformula:
ListFormula, input rs: RumahSakit, input inventory: Inventory
)
{ Menyimpan seluruh data ke file eksternal }

KAMUS LOKAL
    folder_name: string
    base_path, full_path: string
    user_csv, obat_csv, penyakit_csv, obatpenyakit_csv,
config_txt: SEQFILE of text
    mark: character

ALGORITMA
    output("Masukkan nama folder: ")
    input(folder_name)

    base_path ← "data"
    full_path ← gabungPath(base_path, folder_name) // fungsi
bantu menggabungkan path
    output("Saving...")
    if not folderExists(base_path) then
        buatFolder(base_path)
        output("Membuat folder ", base_path)

    if not folderExists(full_path) then
        buatFolder(full_path)
        output("Membuat folder ", full_path)

    // ===== Simpan user.csv =====
    assign(user_csv, gabungPath(full_path, "user.csv"))
    rewrite(user_csv)
    tulisUserKeFile(user_csv, listuser)
    write(user_csv, mark)
    close(user_csv)

    // ===== Simpan obat.csv =====
    assign(obat_csv, gabungPath(full_path, "obat.csv"))
    rewrite(obat_csv)

```

```

tulisObatKeFile(obat_csv, listobat)
write(obat_csv, mark)
close(obat_csv)

// ===== Simpan penyakit.csv =====
assign(penyakit_csv, gabungPath(full_path,
"penyakit.csv"))
rewrite(penyakit_csv)
tulisPenyakitKeFile(penyakit_csv, listpenyakit)
write(penyakit_csv, mark)
close(penyakit_csv)

// ===== Simpan obat_penyakit.csv =====
assign(obatpenyakit_csv, gabungPath(full_path,
"obat_penyakit.csv"))
rewrite(obatpenyakit_csv)
tulisFormulaKeFile(obatpenyakit_csv, listformula)
write(obatpenyakit_csv, mark)
close(obatpenyakit_csv)

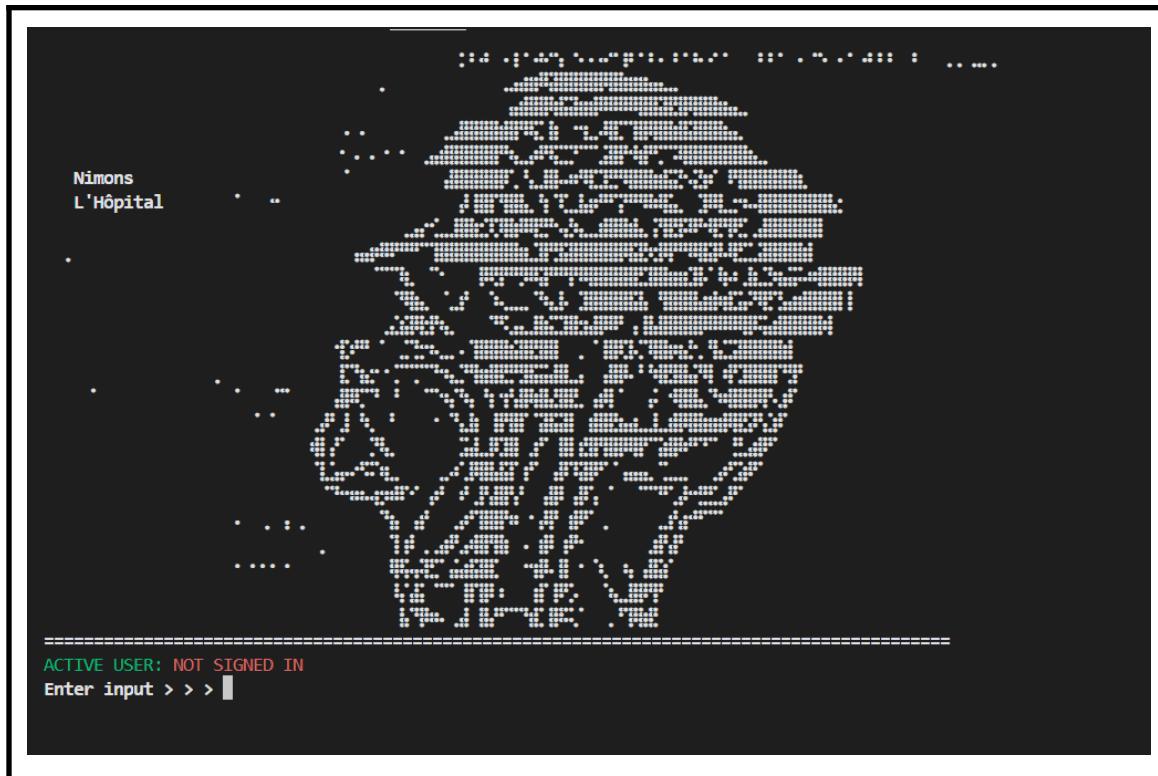
// ===== Simpan config.txt =====
assign(config_txt, gabungPath(full_path, "config.txt"))
rewrite(config_txt)
simpanConfig(config_txt, rs, inventory)
write(config_txt, mark)
close(config_txt)

output("Berhasil menyimpan data di folder ", full_path)

```

3.5 Pengujian Fitur

1. Main Program



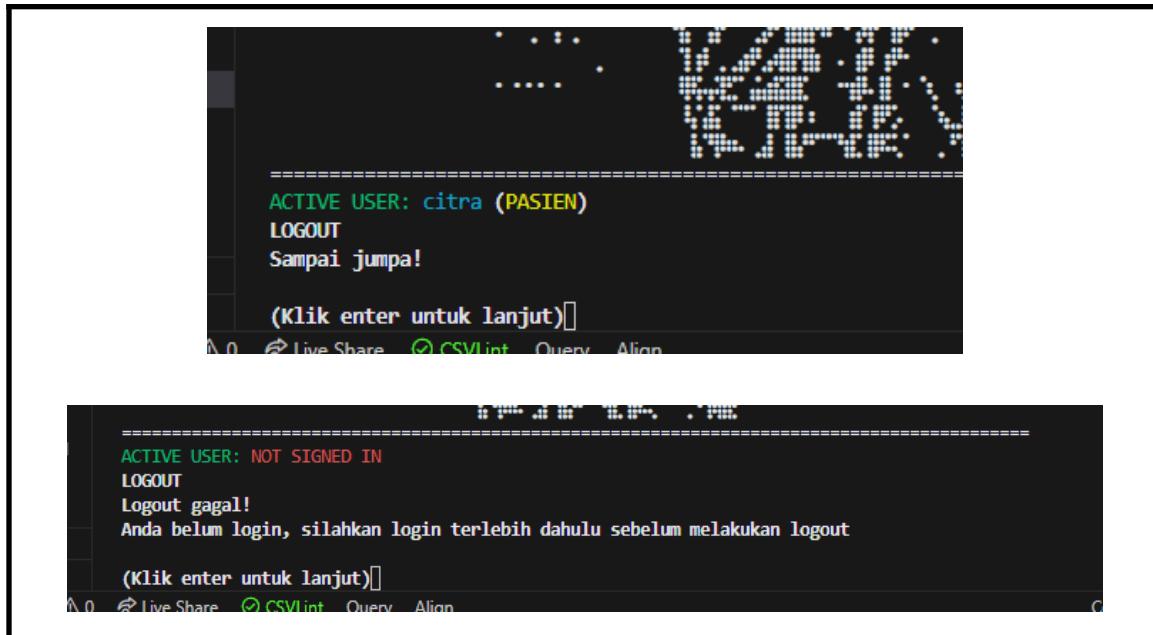
2. F01 - Login



3. F02 - Register Pasien

```
=====
ACTIVE USER:
NOT SIGNED IN
REGISTER
masukkan username:
Fibar
masukkan password:
Fibar123
Data user berhasil ditambahkan:
Username: Fibar
Password: Fibar123
Role: pasien
ID: 6
```

4. F03 - Logout



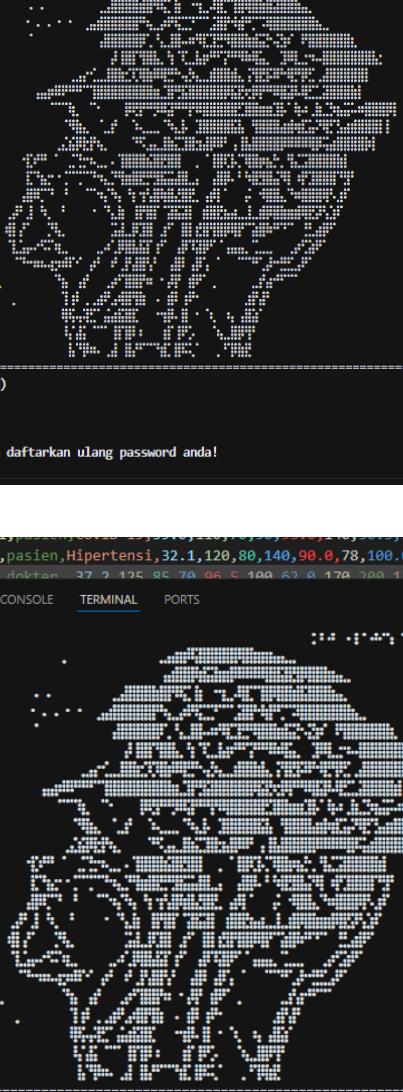
The screenshot shows two terminal windows side-by-side. The left window displays a successful registration message for a user named 'Fibar'. The right window shows the logout process. It starts with the user 'citra (PASIEN)' logging out successfully. A message says 'Sampai jumpa!' (See you later!). Below it, a prompt '(Klik enter untuk lanjut)' (Press enter to continue) is visible. The bottom of the window shows standard terminal navigation keys like arrow keys, 'Live Share', 'CSV Int', 'Query', and 'Align'.

```
=====
ACTIVE USER: citra (PASIEN)
LOGOUT
Sampai jumpa!
(Klik enter untuk lanjut)
A 0 Live Share CSV Int Query Align
```



```
=====
ACTIVE USER: NOT SIGNED IN
LOGOUT
Logout gagal!
Anda belum login, silahkan login terlebih dahulu sebelum melakukan logout
(Klik enter untuk lanjut)
```

5. F04 - Lupa Password



```
4 3,citra,citrалove,pasien,Hipertensi,36.9,115,75,65,98.0,95,48.3,158,180,110,230
5 4 dian diantekan dokter 37.2,125,85,70,06.5,100,60,0,170,200,130,210

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Nimons
L'Hôpital

ACTIVE USER: citra (PASIEN)
LUPA_PASSWORD
Username: citra
Kode Unik: citra
Halo Pasien citra, silakan daftarkan ulang password anda!
Password Baru: citrahate[]

4 3,citra,citrahate,pasien,Hipertensi,32.1,120,80,140,90.0,78,100.0,100,140,110,140
5 4 dian diantekan dokter 37.2,125,85,70,06.5,100,60,0,170,200,130,210

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

(Klik enter untuk lanjut)

ACTIVE USER: citra (PASIEN)
Enter input > > []
```

6. F05 - Menu & Help

Ubuntu Windows PowerShell

Enter input > > HELP

Nimons
L'Hôpital

ACTIVE USER:
NOT SIGNED IN
HELP

Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.

1. LOGIN : Masuk ke dalam akun yang sudah terdaftar
2. REGISTER: Membuat akun baru

Ubuntu Windows PowerShell

alex (PASTEN)

HELP

Selamat datang, alexa. Kamu memanggil command HELP. Kamu pasti sedang kebingungan. Berikut adalah hal-hal yang dapat kamu lakukan sekarang:

1. LOGOUT : Keluar dari akun yang sedang digunakan
2. LIUPA_PASSWORD : Melakukan update password
3. LIHAT DENAH : Melihat denah ruangan
4. LIHAT RUANGAN <nama_ruang> : Melihat ruangan. nama_ruang : A1 , A2 dst
5. PULANGDOK : Akyu boleh pulang ga, dok? >/<
6. DAFTAR_CHECKUP : Mendaftar untuk melakukan check-up dengan dokter
7. ANTRIAN : Melihat status antrian
8. MINUM OBAT : Minum obatttt
9. PENAWAR : Penawar khusus jika kamu salah minum obat
9. EXIT:

The image shows two screenshots of a Windows PowerShell window running on an Ubuntu desktop. The background features a dark anime-style illustration of a person's face and a book cover for 'THE C PROGRAMMING LANGUAGE' by Brian W. Kernighan and Dennis M. Ritchie.

Screenshot 1 (User: dian):

```
Ubuntu x Windows PowerShell x + v
Nimons
L'Hôpital
ACTIVE USER:
dian (DOKTER)
HELP
Halo Dokter dian. Kamu memanggil command HELP. Kamu pasti sedang kebingungan. Berikut
adalah hal-hal yang dapat kamu lakukan sekarang:
1. LOGOUT : Keluar dari akun yang sedang digunakan
2. LUPA_PASSWORD : Melakukan update password
3. LIHAT_DENAH : Melihat denah ruangan
4. LIHAT_RUANGAN <nama_ruang> : Melihat ruangan. nama_ruang : A1 , A2 dst
5. DIAGNOSIS : Auto-Diagnose-2.0 ; Diagnosis secara otomatis!
6. NGOBATIN : Auto-Prescription-2.0 ; Sistem peresepan obat
7. EXIT :
```

Screenshot 2 (User: elvin):

```
Ubuntu x Windows PowerShell x + v
ACTIVE USER:
elvin (MANAGER)
HELP
Halo manager elvin. Kenapa kamu memanggil command HELP? Kan kamu manager, tapi
yasudahlah kamu pasti sedang kebingungan. Berikut adalah hal-hal yang dapat kamu lakukan
sekarang:
1. LOGOUT : Keluar dari akun yang sedang digunakan
2. LUPA_PASSWORD : Melakukan update password
3. LIHAT_DENAH : Melihat denah ruangan
4. LIHAT_RUANGAN <nama_ruang> : Melihat ruangan. nama_ruang : A1 , A2 dst
5. LIHAT_USER : Melihat data seluruh pengguna
6. CARI_USER : Mencari data pengguna secara spesifik
7. LIHAT_SEMUA_ANTRIAN : Melihat seluruh rincian di seluruh ruangan
8. TAMBAH_DOKTER : Menambahkan dokter baru
9. EXIT : Keluar dari program
```

7. F06 - Denah Rumah Sakit

```
=====
ACTIVE USER: elvin (MANAGER)
LIHAT_DENAH
    1   2   3
+---+---+---+
A | A1| A2| A3|
+---+---+---+
B | B1| B2| B3|
+---+---+---+
(Klik enter untuk lanjut)█
```

8. F07 - Lihat User

```
=====
ACTIVE USER: elvin (MANAGER)
LIHAT_USER
Urutkan berdasarkan?
1. ID
2. Nama
>> Pilihan: 1
Urutan sort?
1. ASC (A-Z)
2. DESC (Z-A)
>> Pilihan: 1
Menampilkan pengguna dengan ID terurut ascending...
ID | Nama      | Role     | Penyakit
-----+-----+-----+-----+
1 | alexa     | pasien   | Influenza
2 | budi      | pasien   | COVID-19
3 | citra     | pasien   | Hipertensi
4 | dian      | dokter   | -
5 | elvin     | manager  | -
6 | fajar     | pasien   | COVID-19
7 | gina      | pasien   | Diabetes Mellitus
8 | hari      | pasien   | Hipertensi
9 | irene     | pasien   | Influenza
10 | joko     | dokter   | -
11 | kiki     | pasien   | COVID-19
12 | lina     | dokter   | -
13 | maya     | dokter   | -
14 | nino     | pasien   | Diabetes Mellitus
15 | oscar    | dokter   | -
16 | miguh    | dokter   | -
17 | fazaboba | dokter   | -
```

```

ACTIVE USER: elvin (MANAGER)
LIHAT_USER
Urutkan berdasarkan?
1. ID
2. Nama
>> Pilihan: miguh
Masukan salah, mohon input hanya 1 atau 2!
Urutkan berdasarkan?
1. ID
2. Nama
>> Pilihan: 1
Urutan sort?
1. ASC (A-Z)
2. DESC (Z-A)
>> Pilihan: miguh
Masukan salah, mohon input hanya 1 atau 2!
Urutan sort?
1. ASC (A-Z)
2. DESC (Z-A)
>> Pilihan: 2
Menampilkan pengguna dengan ID terurut descending...
ID | Nama      | Role    | Penyakit
---+---+---+---+
17 | fazaboba  | dokter  | -
16 | miguh     | dokter  | -
15 | oscar     | dokter  | -
14 | nino      | pasien   | Diabetes Mellitus
13 | maya      | dokter  | -
12 | lina      | dokter  | -
11 | kiki      | pasien   | COVID-19
10 | joko      | dokter  | -
9  | irene     | pasien   | Influenza
8  | hari      | pasien   | Hipertensi
7  | gina      | pasien   | Diabetes Mellitus
6  | fajar     | pasien   | COVID-19
5  | elvin     | manager  | -
4  | dian      | dokter  | -
3  | citra     | pasien   | Hipertensi
2  | budi      | pasien   | COVID-19
1  | alexa     | pasien   | Influenza

```

(Klik enter untuk lanjut) █

```

ACTIVE USER: elvin (MANAGER)
LIHAT_USER
Urutkan berdasarkan?
1. ID
2. Nama
>> Pilihan: 2
Urutan sort?
1. ASC (A-Z)
2. DESC (Z-A)
>> Pilihan: 1
Menampilkan pengguna dengan nama terurut ascending...
ID | Nama      | Role    | Penyakit
---+---+---+---+
1 | alexa     | pasien  | Influenza
2 | budi       | pasien  | COVID-19
3 | citra     | pasien  | Hipertensi
4 | dian       | dokter  | -
5 | elvin     | manager | -
6 | fajar      | pasien  | COVID-19
17 | fazaboba  | dokter  | -
7 | gina       | pasien  | Diabetes Mellitus
8 | hari       | pasien  | Hipertensi
9 | irene      | pasien  | Influenza
10 | joko      | dokter  | -
11 | kiki       | pasien  | COVID-19
12 | lina       | dokter  | -
13 | maya      | dokter  | -
16 | miguh     | dokter  | -
14 | nino       | pasien  | Diabetes Mellitus
15 | oscar      | dokter  | -

```

(Klik enter untuk lanjut)█

```

ACTIVE USER: elvin (MANAGER)
LIHAT_DOKTER
Urutkan berdasarkan?
1. ID
2. Nama
>> Pilihan: 1
Urutan sort?
1. ASC (A-Z)
2. DESC (Z-A)
>> Pilihan: 2
Menampilkan dokter dengan ID terurut descending...
ID | Nama
---+---
17 | fazaboba
16 | miguh
15 | oscar
13 | maya
12 | lina
10 | joko
4 | dian

```

(Klik enter untuk lanjut)█

```
ACTIVE USER: elvin (MANAGER)
LIHAT_DOKTER
Urutkan berdasarkan?
1. ID
2. Nama
>> Pilihan: 2
Urutan sort?
1. ASC (A-Z)
2. DESC (Z-A)
>> Pilihan: 1
Menampilkan dokter dengan nama terurut ascending...
ID | Nama
-----
```

4	dian
17	fazaboba
10	joko
12	lina
13	maya
16	miguh
15	oscar

(Klik enter untuk lanjut)█

```
ACTIVE USER: elvin (MANAGER)
LIHAT_PASIEN
Urutkan berdasarkan?
1. ID
2. Nama
>> Pilihan: 1
Urutan sort?
1. ASC (A-Z)
2. DESC (Z-A)
>> Pilihan: 2
Menampilkan pasien dengan ID terurut descending...
ID | Nama | Penyakit
-----
```

14	nino	Diabetes Mellitus
11	kiki	COVID-19
9	irene	Influenza
8	hari	Hipertensi
7	gina	Diabetes Mellitus
6	fajar	COVID-19
3	citra	Hipertensi
2	budi	COVID-19
1	alexa	Influenza

(Klik enter untuk lanjut)█

```

=====
ACTIVE USER: elvin (MANAGER)
LIHAT_PASIEN
Urutkan berdasarkan?
1. ID
2. Nama
>> Pilihan: 2
Urutan sort?
1. ASC (A-Z)
2. DESC (Z-A)
>> Pilihan: miguh
Masukan salah, mohon input hanya integer dari 1 sampai 2!
Urutan sort?
1. ASC (A-Z)
2. DESC (Z-A)
>> Pilihan: 1
Menampilkan pasien dengan nama terurut ascending...
ID | Nama      | Penyakit
+-----+
1 | alexa     | Influenza
2 | budi       | COVID-19
3 | citra     | Hipertensi
6 | fajar      | COVID-19
7 | gina       | Diabetes Mellitus
8 | hari       | Hipertensi
9 | irene      | Influenza
11 | kiki      | COVID-19
14 | nino      | Diabetes Mellitus

(Klik enter untuk lanjut)█

```

9. F08 - Cari User

```

=====
ACTIVE USER: elvin (MANAGER)
CARI_USER
Cari berdasarkan?
1. ID
2. Nama
>>> Pilihan: 1

>>> Masukkan nomor ID user: miguh
Masukan salah, mohon input hanya integer dari 1 sampai 17!

>>> Masukkan nomor ID user: 18
Masukan salah, mohon input hanya integer dari 1 sampai 17!

>>> Masukkan nomor ID user: 13

Menampilkan pengguna dengan nomor ID 13...
ID | Nama      | Role    | Penyakit
+-----+
13 | maya      | dokter  | -

```

(Klik enter untuk lanjut)█

```
=====  
ACTIVE USER: elvin (MANAGER)  
CARI_USER  
Cari berdasarkan?  
1. ID  
2. Nama  
>>> Pilihan: 2  
  
>>> Masukkan nama user: miguh  
  
Menampilkan pengguna dengan nama miguh...  
ID | Nama | Role | Penyakit  
---+-----+-----+  
16 | miguh | dokter | -  
  
(Klik enter untuk lanjut)█
```

```
=====  
ACTIVE USER: elvin (MANAGER)  
CARI_USER  
Cari berdasarkan?  
1. ID  
2. Nama  
>>> Pilihan: 2  
  
>>> Masukkan nama user: samson  
  
Tidak ditemukan pengguna dengan nama samson!  
  
(Klik enter untuk lanjut)█
```

```
=====  
ACTIVE USER: elvin (MANAGER)  
CARI_DOKTER  
Cari berdasarkan?  
1. ID  
2. Nama  
>>> Pilihan: 1  
  
>>> Masukkan nomor ID dokter: MIGHU  
Masukan salah, mohon input hanya integer dari 1 sampai 17!  
  
>>> Masukkan nomor ID dokter: 15  
  
Menampilkan dokter dengan ID 15...  
ID | Nama  
---+-----  
15 | oscar  
  
(Klik enter untuk lanjut)█
```

```
=====  
ACTIVE USER: elvin (MANAGER)  
CARI_DOKTER  
Cari berdasarkan?  
1. ID  
2. Nama  
>>> Pilihan: 1  
  
>>> Masukkan nomor ID dokter: 1  
  
Tidak ditemukan dokter dengan ID 1!  
  
(Klik enter untuk lanjut)█
```

```
=====  
ACTIVE USER: elvin (MANAGER)  
CARI_DOKTER  
Cari berdasarkan?  
1. ID  
2. Nama  
>>> Pilihan: 2  
  
>>> Masukkan nama dokter: fazaboba  
  
Menampilkan dokter dengan nama fazaboba...  
ID | Nama  
---+---  
17 | fazaboba  
  
(Klik enter untuk lanjut)█
```

```
=====  
ACTIVE USER: elvin (MANAGER)  
CARI_PASIEN  
Cari berdasarkan?  
1. ID  
2. Nama  
3. Penyakit  
>>> Pilihan: 1  
  
>>> Masukkan nomor ID pasien: 17  
  
Tidak ditemukan pasien dengan ID 17!  
  
(Klik enter untuk lanjut)█
```

```
=====  
ACTIVE USER: elvin (MANAGER)  
CARI_PASIEN  
Cari berdasarkan?  
1. ID  
2. Nama  
3. Penyakit  
>>> Pilihan: 1  
  
>>> Masukkan nomor ID pasien: 1  
  
Menampilkan pasien dengan ID 1...  
ID | Nama | Penyakit  
---+---+  
1 | alexa | Influenza  
  
(Klik enter untuk lanjut)█
```

```
ACTIVE USER: elvin (MANAGER)
CARI_PASIEN
Cari berdasarkan?
1. ID
2. Nama
3. Penyakit
>>> Pilihan: 2

>>> Masukkan nama pasien: citra

Menampilkan pasien dengan nama citra...
ID | Nama      | Penyakit
---+---+-----+
3  | citra     | Hipertensi

(Klik enter untuk lanjut)█
```

```
ACTIVE USER: elvin (MANAGER)
CARI_PASIEN
Cari berdasarkan?
1. ID
2. Nama
3. Penyakit
>>> Pilihan: 2

>>> Masukkan nama pasien: samson

Tidak ditemukan pasien dengan nama samson!

(Klik enter untuk lanjut)█
```

```
ACTIVE USER: elvin (MANAGER)
CARI_PASIEN
Cari berdasarkan?
1. ID
2. Nama
3. Penyakit
>>> Pilihan: 3

>>> Masukkan nama penyakit: samson

Tidak ditemukan pasien dengan penyakit samson!

(Klik enter untuk lanjut)█
```

```
=====  
ACTIVE USER: elvin (MANAGER)  
CARI_PASIEN  
Cari berdasarkan?  
1. ID  
2. Nama  
3. Penyakit  
>>> Pilihan: 3  
  
>>> Masukkan nama penyakit: Influenza  
  
Urutkan berdasarkan?  
1. ID  
2. Nama  
>>> Pilihan: 1  
  
Urutan sort?  
1. ASC (A-Z)  
2. DESC (Z-A)  
>>> Pilihan: 2  
  
Menampilkan pasien dengan penyakit Influenza dengan ID terurut descending...  
ID | Nama | Penyakit  
---+-----+  
 9 | irene | Influenza  
 1 | alexa | Influenza  
  
(Klik enter untuk lanjut)■
```

```
=====  
ACTIVE USER: elvin (MANAGER)  
CARI_PASIE3N  
Cari berdasarkan?  
1. ID  
2. Nama  
3. Penyakit  
>>> Pilihan:  
  
>>> Masukkan nama penyakit: COVID-19  
  
Urutkan berdasarkan?  
1. ID  
2. Nama  
>>> Pilihan: 2  
  
Urutan sort?  
1. ASC (A-Z)  
2. DESC (Z-A)  
>>> Pilihan: 1  
  
Menampilkan pasien dengan penyakit COVID-19 dengan ID terurut ascending...  
ID | Nama | Penyakit  
---+-----+  
 2 | budi | COVID-19  
 6 | fajar | COVID-19  
11 | kiki | COVID-19  
  
(Klik enter untuk lanjut)■
```

10. F09 - Lihat Antrian

```

=====
ACTIVE USER: elvin (MANAGER)
LIHAT_SEMUA_ANTRIAN
 1 2 3
+---+---+
A | A1| A2| A3|
+---+---+
B | B1| B2| B3|
+---+---+

===== A1 =====
Kapasitas : 3
Dokter    : Dr. joko
Pasien di dalam ruangan:
 1. budi
 2. citra
 3. alexa
Pasien di antrian:
 1. miguh

===== A2 =====
Kapasitas : 3
Dokter    : Dr. dian
Pasien di dalam ruangan:
 1. kiki
Pasien di antrian:
 Tidak ada pasien di antrian saat ini.

===== A3 =====
Kapasitas : 3
Dokter    : Dr. lina
Pasien di dalam ruangan:
 1. fajar
Pasien di antrian:
 Tidak ada pasien di antrian saat ini.

===== B1 =====
Kapasitas : 3
Dokter    : Dr. maya
Pasien di dalam ruangan:
 1. hari
 2. gina
Pasien di antrian:
 Tidak ada pasien di antrian saat ini.

===== B3 =====
Kapasitas : 3
Dokter    : Dr. oscar
Pasien di dalam ruangan:
 Tidak ada pasien di dalam ruangan saat ini.
Pasien di antrian:
 Tidak ada pasien di antrian saat ini.

(klik enter untuk lanjut)■

```

11. F10 - Tambah Dokter

```
ACTIVE USER: elvin (MANAGER)
TAMBAH_DOKTER
Username: Samson
Password: samsonganteng
Dokter Samson berhasil ditambahkan!
```

(Klik enter untuk lanjut) █

```
ACTIVE USER: elvin (MANAGER)
TAMBAH_DOKTER
Username: miguh
Sudah ada Dokter bernama miguh!
```

(Klik enter untuk lanjut) █

```
ACTIVE USER: elvin (MANAGER)
ASSIGN_DOKTER
Username: miguh
Ruangan: Z10
Ruangan tidak valid.
```

(Klik enter untuk lanjut) █

```
ACTIVE USER: elvin (MANAGER)
ASSIGN_DOKTER
Username: samson
Dokter dengan username samson tidak ditemukan.
```

(Klik enter untuk lanjut) █

12. F11 - Diagnosis

```
ACTIVE USER: oscar (DOKTER)
DIAGNOSIS
Memulai proses diagnosis...
Mencari ruangan untuk dokter dengan ID: 15
Mengambil pasien dengan ID: 16 dari antrian.
Melakukan pemeriksaan terhadap pasien: fibar
Mengecek kemungkinan penyakit: Influenza
Kriteria tidak cocok untuk penyakit: Influenza
Mengecek kemungkinan penyakit: COVID-19
Diagnosis berhasil! Pasien fibar didiagnosis dengan: COVID-19
```

13. F12 - Ngobatin

ACTIVE USER: joko (DOKTER)
NGOBATIN
Dokter sedang mengobati pasien!
-> obat cocok, dan akan ditambahkan ke inventory pasien.
1. Remdesivir

(Klik enter untuk lanjut)■

14. F13 - Aku boleh pulang ga, dok?

ACTIVE USER: budi (PASIEN)
PULANGDOK
Dokter sedang memeriksa keadaanmu...
Masih ada obat yang belum kamu habiskan, minum semuanya dulu yukk!

(Klik enter untuk lanjut)■

15. F14 - Daftar Check-Up

ACTIVE USER: fibar (PASIEN)
DAFTAR_CHECKUP
1. Dr. joko (Ruang A1)
2. Dr. dian (Ruang A2)
3. Dr. lina (Ruang A3)
4. Dr. maya (Ruang B1)
5. Dr. oscar (Ruang B3)
Masukkan suhu tubuh (32.0 - 42.0): ■

```
Masukkan suhu tubuh (32.0 - 42.0): 38.2
Masukkan tekanan darah (sistolik diastolik): 120
80
Masukkan detak jantung (61 - 159): 71
Masukkan saturasi oksigen (80.0 - 100.0): 90
Masukkan kadar gula darah (71 - 139): 75
Masukkan berat badan (11.0 - 199.0): 110
Masukkan tinggi badan (51 - 249): 165
Masukkan kadar kolesterol (101 - 299): 175
Masukkan kadar trombosit (101 - 499): 200
```

1. Dr. joko (Ruangan A1)
2. Dr. dian (Ruangan A2)
3. Dr. lina (Ruangan A3)
4. Dr. maya (Ruangan B1)
5. Dr. oscar (Ruangan B3)

pilih dokter yang tersedia:

Pilih dokter yang tersedia (1 - 5): 5

Pasien berhasil masuk ke antrian check up.

Dokter: Dr. oscar

Nomor antrian Anda: 1

Daftar antrian saat ini:

1. fibar

(Klik enter untuk lanjut)

16. F15 - Antrian Saya!

ACTIVE USER: budi (PASIEN)

ANTRIAN

Anda sedang berada di dalam ruangan dokter!

(Klik enter untuk lanjut)

17. F16 - Minum Obat

```
=====  
ACTIVE USER: budi (PASIEN)  
MINUM_OBAT  
Tidak ada obat dalam inventory!  
  
(Klik enter untuk lanjut)[]  
0 ↗ Live Share
```



```
=====  
ACTIVE USER: budi (PASIEN)  
MINUM_OBAT  
===== DAFTAR OBAT =====  
1. Remdesivir  
  
Pilih obat untuk diminum: 1[]  
↗ Live Share
```

18. F17 - Minum Penawar

```
=====  
ACTIVE USER: budi (PASIEN)  
PENAWAR  
Uwekkk!!! Remdesivir keluar dan kembali ke inventory  
  
(Klik enter untuk lanjut)[]  
↗ Live Share
```



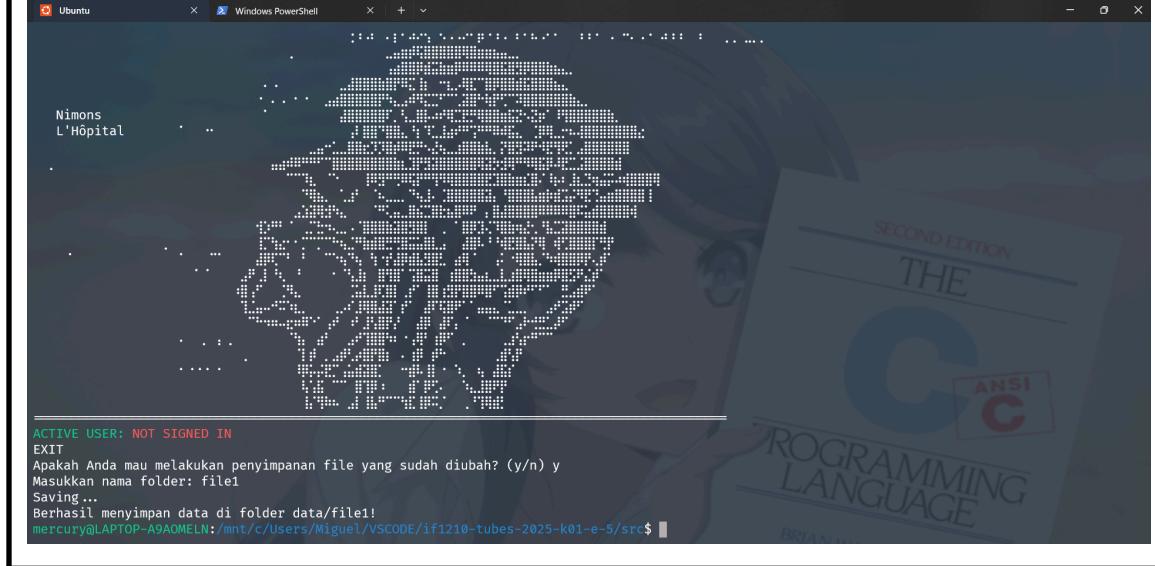
```
=====  
ACTIVE USER: budi (PASIEN)  
PENAWAR  
Perut kosong!! Belum ada obat yang dimakan.  
  
(Klik enter untuk lanjut)[]
```

19. F18 - Exit



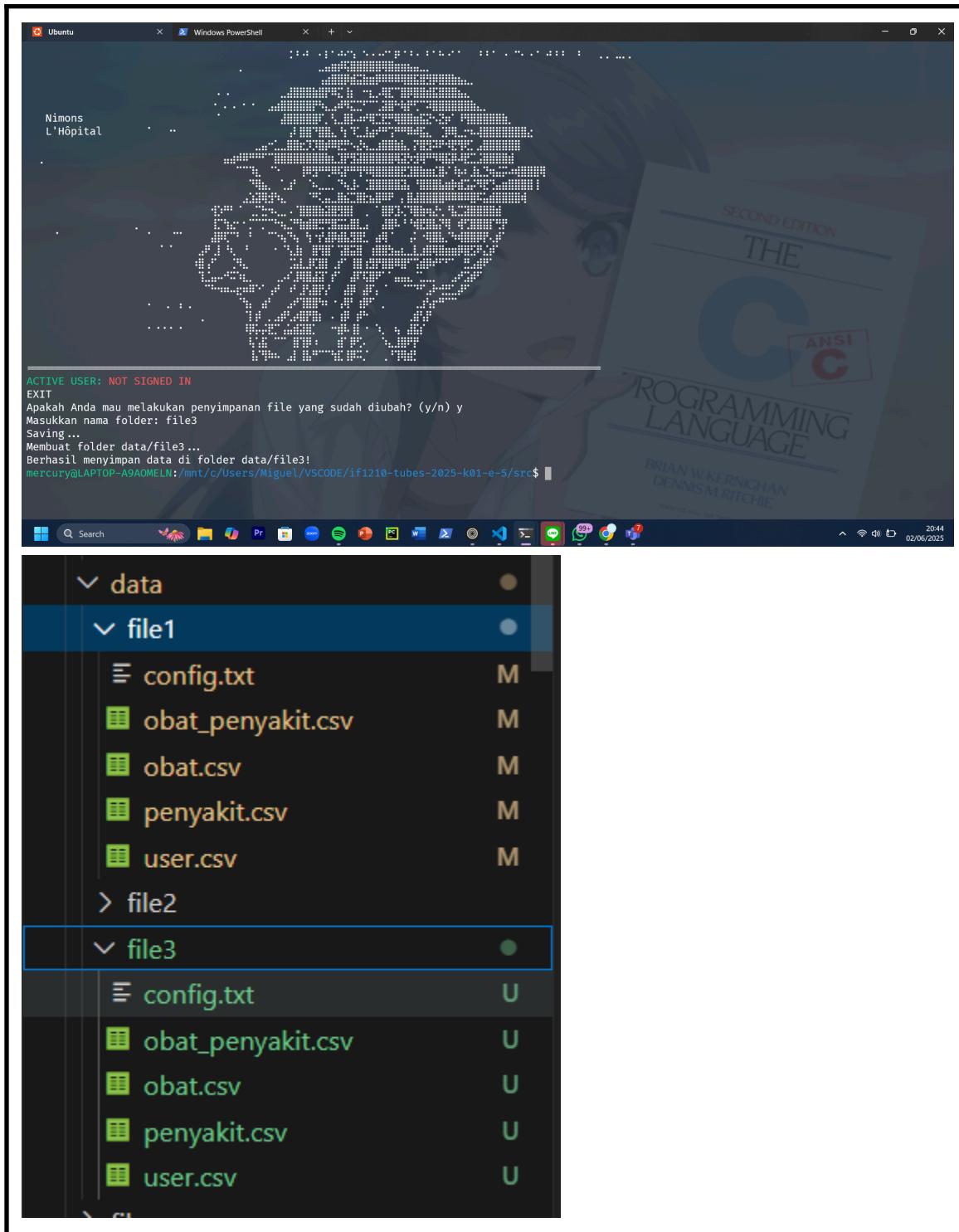
Nimons
L'Hôpital

```
ACTIVE USER: NOT SIGNED IN
EXIT
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) n
mercury@LAPTOP-A9AOMELN:/mnt/c/Users/Miguel/VSCODE/if1210-tubes-2025-k01-e-5/src$
```

Nimons
L'Hôpital

```
ACTIVE USER: NOT SIGNED IN
EXIT
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) y
Masukkan nama folder: file1
Saving...
Berhasil menyimpan data di folder data/file1!
mercury@LAPTOP-A9AOMELN:/mnt/c/Users/Miguel/VSCODE/if1210-tubes-2025-k01-e-5/src$
```



20. D03 - Load

```

mercury@LAPTOP-A9AOMELN:/mnt/c/Users/Miguel/VS CODE/if1210-tubes-2025-k01-e-3/src$ ..//build/main file1
0123456789101112131415012345678910111213141501234567891011121314150123456789101112131415012345678910
1112131415012345678910111213141501234567891011121314150123456789101112131415012345678910111213141501
23456789101112131415012345678910111213141501234567891011121314150123456789101112131415012345678910111213141501
0123456789101112131415012345678910111213141501234567891011121314150123456789101112131415012345678910111213141501Selamat datang kembali di rumah sa
kit Nimons !

```

ACTIVE USER:

```

mercury@LAPTOP-A9AOMELN:/mnt/c/Users/Miguel/VS CODE/if1210-tubes-2025-k01-e-3/src$ ..//build/main folderpalsu
0123456789101112131415012345678910111213141501234567891011121314150123456789101112131415012345678910
1112131415012345678910111213141501234567891011121314150123456789101112131415012345678910111213141501
23456789101112131415012345678910111213141501234567891011121314150123456789101112131415012345678910111213141501
mercury@LAPTOP-A9AOMELN:/mnt/c/Users/Miguel/VS CODE/if1210-tubes-2025-k01-e-3/src$ 
```

```

mercury@LAPTOP-A9AOMELN:/mnt/c/Users/Miguel/VS CODE/if1210-tubes-2025-k01-e-3/src$ ..//build/main file2
0123456789101112131415012345678910111213141501234567891011121314150123456789101112131415012345678910
1112131415012345678910111213141501234567891011121314150123456789101112131415012345678910111213141501
23456789101112131415012345678910111213141501234567891011121314150123456789101112131415012345678910111213141501
mercury@LAPTOP-A9AOMELN:/mnt/c/Users/Miguel/VS CODE/if1210-tubes-2025-k01-e-3/src$ 
```

▼ data

- ▼ file1
 - Ξ config.txt
 - ▀ obat_penyakit.csv
 - ▀ obat.csv
 - ▀ penyakit.csv
 - ▀ user.csv
- ▼ file2
 - ▀ obat_penyakit.csv
 - ▀ obat.csv
 - ▀ penyakit.csv
 - ▀ user.csv

21. D04 - Save

1 id,username,password,role,riwayat_penyakit,suhu_tubuh,tekanan_darah_sistolik,tekanan_darah_diastolik,detak_jantung,saturasi_oksigen,kadar
 2 1,alex,alexasecure1,pasien,Influenza,38,2,120,80,75,97,5,110,55,0,165,190,120,240
 3 2,budi,budikeren21,pasien,COVID-19,39,0,110,70,90,95,0,140,50,5,160,220,150,180
 4 3,citra,citralore,pasien,Hipertensi,36,9,115,75,65,98,0,95,48,3,158,180,110,230
 5 4,dian,dianDokter,dokter,,37,2,125,85,70,96,5,100,62,0,170,200,130,210
 6 5,elvin,elvman123,manager,,36,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
 7 6,fajar,fajar321,pasien,COVID-19,38,4,118,78,88,96,2,135,53,1,168,210,140,200
 8 7,gina,ginagirl,pasien,Diabetes Mellitus,37,8,112,76,70,97,0,160,60,2,172,230,145,220
 9 8,hari,harihari,pasien,Hipertensi,36,7,130,90,72,95,5,105,66,0,180,190,125,250
 10 9,irene,irene123,pasien,Influenza,38,0,122,82,74,98,5,100,54,0,163,195,115,235
 11 10,joko,jokodoc,dokter,,36,6,120,80,69,97,5,98,60,0,175,185,110,215
 12 11,kiki,kikipasien,pasien,COVID-19,39,1,115,77,85,94,0,145,58,3,166,205,130,190
 13 12,lina,linadoc,dokter,,37,0,125,85,71,96,8,102,63,0,178,200,135,205
 14 13,maya,mayamedic,dokter,,36,8,118,80,68,97,3,99,59,0,172,190,120,198
 15 14,nino,ninopasien,pasien,Diabetes Mellitus,37,9,110,75,80,96,0,155,57,2,170,225,150,210
 16 15,oscar,oscardoctor,dokter,,37,3,123,83,73,97,6,101,61,5,174,195,125,212
 17
 18

Ubuntu x Windows PowerShell x + v

Nimons L'Hôpital

ACTIVE USER: elvin (MANAGER)

SAVE

Masukkan nama folder: file1

Saving ...

Berhasil menyimpan data di folder data/file1!

(Klik enter untuk lanjut)■

main.c user.csv M load.c

src > data > file1 > user.csv > data

1 id,username,password,role,riwayat_penyakit,suhu_tubuh,tekanan_darah_sistolik,tekanan_darah_diastolik,detak_jantung,saturasi_oksigen,kadar
 2 1,alex,alexasecure1,pasien,Influenza,38,2,120,80,75,97,5,110,55,0,165,190,120,240
 3 2,budi,budikeren21,pasien,COVID-19,39,0,110,70,90,95,0,140,50,5,160,220,150,180
 4 3,citra,citralore,pasien,Hipertensi,36,9,115,75,65,98,0,95,48,3,158,180,110,230
 5 4,dian,dianDokter,dokter,,37,2,125,85,70,96,5,100,62,0,170,200,130,210
 6 5,elvin,elvman123,manager,,36,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
 7 6,fajar,fajar321,pasien,COVID-19,38,4,118,78,88,96,2,135,53,1,168,210,140,200
 8 7,gina,ginagirl,pasien,Diabetes Mellitus,37,8,112,76,70,97,0,160,60,2,172,230,145,220
 9 8,hari,harihari,pasien,Hipertensi,36,7,130,90,72,95,5,105,66,0,180,190,125,250
 10 9,irene,irene123,pasien,Influenza,38,0,122,82,74,98,5,100,54,0,163,195,115,235
 11 10,joko,jokodoc,dokter,,36,6,120,80,69,97,5,98,60,0,175,185,110,215
 12 11,kiki,kikipasien,pasien,COVID-19,39,1,115,77,85,94,0,145,58,3,166,205,130,190
 13 12,lina,linadoc,dokter,,37,0,125,85,71,96,8,102,63,0,178,200,135,205
 14 13,maya,mayamedic,dokter,,36,8,118,80,68,97,3,99,59,0,172,190,120,198
 15 14,nino,ninopasien,pasien,Diabetes Mellitus,37,9,110,75,80,96,0,155,57,2,170,225,150,210
 16 15,oscar,oscardoctor,dokter,,37,3,123,83,73,97,6,101,61,5,174,195,125,212
 17
 18

BAB IV

LAMPIRAN

Lampiran 1. MoM Asistensi 1

<https://docs.google.com/document/d/1otoGIZBhSmqQqkQJ9Ln5x1NBHWeqsUej/edit?usp=sharing&oid=104094379305910453571&rtpof=true&sd=true>

 IF1210_FormAsistensiTB_1_K1-E.docx.pdf

Lampiran 2. MoM Asistensi 2:

https://docs.google.com/document/d/1Bb5qYmCljUpGhZzqi3P_S9NZDSt9cLUz/edit?usp=sharing&oid=104094379305910453571&rtpof=true&sd=true

 IF1210_FormAsistensiTB_2_K1-E.docx.pdf