

## **TUGAS BESAR**

**Mata Kuliah : IF1210 Algoritma dan Pemrograman 1**

*Milestone 2*



**Disusun oleh Kelompok L Kelas 01:**

Junior Natra Situmorang	13524055
Kalyca Nathania B. Manullang	13524071
Yavie Azka Putra Araly	13524077
Muhammad Faiz Alfada Dharma	13524097
Rainaldi Pratama F. Sembiring	13524117
Reysha Syafitri Mulya Ramadhan	13524137

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2025**

## **HALAMAN PERNYATAAN**

**Kelompok K01-L – IF1210 Algoritma dan Pemrograman 1**

**Semester 2 Tahun Akademik 2024/2025**

**Sekolah Teknik Elektro dan Informatika – ITB**

Kami, kelompok K01-L yang beranggotakan:

1. 13524055 – Junior Natra Situmorang
2. 13524071 – Kalyca Nathania B. Manullang
3. 13524077 - Yavie Azka Putra Araly
4. 13524097 – Muhammad Faiz Alfada Dharma
5. 13524117 – Rainaldi Pratama F. Sembiring
6. 13524137 – Reysha Syafitri Mulya Ramadhan

dengan ini menyatakan hal berikut:

“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.”

Demikian pernyataan ini dibuat dengan sebenar-benarnya.

## **DAFTAR ISI**

Halaman Pernyataan.....	1
Daftar Isi.....	2
Daftar Tabel.....	3
Daftar Gambar.....	4
Bab I Deskripsi Persoalan.....	5
Bab II Rencana Implementasi.....	6
Bab III Daftar Pembagian Kerja Anggota Kelompok.....	7
Bab IV <i>Checklist</i> Hasil Rancangan, Implementasi, dan <i>Testing</i> .....	12
Bab V Desain <i>Command</i> untuk Setiap Primitif.....	14
Bab VI Desain Kamus Data.....	15
Bab VII Desain Dekomposisi Algoritmik dan Fungsional Program.....	16
Bab VIII Spesifikasi untuk Tiap Modul/Prosedur/Fungsi yang Dibuat (dalam Notasi Algoritmik).....	17
Bab IX Tangkapan Layar Hasil Pengujian Program Berdasarkan Fitur-fitur pada Spesifikasi.....	21
Bab X Lampiran.....	22

## **DAFTAR TABEL**

<b>Tabel 2.1</b> Rencana Implementasi.....	6
<b>Tabel 3.1</b> Daftar Pembagian Kerja Anggota Kelompok.....	7
<b>Tabel 4.1</b> Checklist Hasil Rancangan, Implementasi dan Testing.....	14

## **DAFTAR GAMBAR**

Gambar 10.1 Asistensi 1.....	208
Gambar 10.2 Asistensi 2.....	211

## **BAB I**

### **DESKRIPSI PERSOALAN**

Tugas besar *milestone* 1 meminta penulis untuk membangun sebuah program berbasis terminal (*Command Line Interface/CLI*) yang mampu menangani kebutuhan manajemen data pengguna, seperti proses registrasi, *login*, pencarian pengguna, hingga tampilan informasi dalam bentuk visualisasi data. Program yang dibangun bersifat modular dan dikembangkan dalam kelompok sehingga perlu merancang alur program yang terstruktur, fleksibel, dan mudah untuk dikembangkan secara kolaboratif.

Secara umum, program akan berfungsi sebagai sistem manajemen pengguna. Pengguna dapat mendaftar sebagai *user* baru (melalui fitur *register*), masuk ke sistem (*login*), serta mengakses berbagai informasi melalui fitur-fitur seperti lihat *user*, cari *user*, dan lain-lain. Pada menu utama (*help*), pengguna akan diarahkan ke fitur-fitur ini.

Tugas besar sekarang menekankan pentingnya kolaborasi tim dalam mengembangkan modul-modul secara terpisah, tetapi terintegrasi. Untuk itu, dibutuhkan sistem manajemen *state* program yang rapi agar setiap modul dapat saling berkomunikasi dan berbagi data dengan baik. Salah satu pendekatan yang disarankan adalah menggunakan struct global yang berisi semua variabel penting, yang kemudian dapat diparsing ke dalam fungsi-fungsi sesuai kebutuhan, sehingga meminimalisir penggunaan variabel global sembarangan dan meningkatkan keterbacaan serta pengujian program.

Program dikembangkan di lingkungan Linux (disarankan menggunakan WSL untuk pengguna Windows) agar lebih kompatibel dan sesuai dengan konteks penggunaan bahasa C yang digunakan dalam tugas ini. Beberapa fungsi dalam bahasa C dapat memiliki perbedaan perilaku antara Windows dan Linux sehingga penting untuk melakukan pengujian langsung di sistem yang menyerupai target lingkungan program.

Dengan tugas besar, penulis tidak hanya diminta untuk menerapkan konsep dasar pemrograman dan algoritma, tetapi juga dilatih untuk bekerja dalam tim, memahami

praktik pengembangan perangkat lunak skala kecil hingga menengah, serta merancang program yang bersifat modular, dapat di-*merge*, dan teruji secara fungsional.

## BAB II

### RENCANA IMPLEMENTASI

**Tabel 2.1** Rencana Implementasi

<b>Implementasi ADT</b>	<b>FITUR</b>	<b>Deskripsi Implementasi</b>	<b>Alasan Implementasi</b>
ADT Sederhana	F01 - Login F02 - Register F03 - Logout F04 - Lupa Password F05 - Menu & Help F06 / D01 - Denah Rumah Sakit F07 - Lihat User F08 - Cari User F09 / D02 - Lihat Antrian F10 - Tambah Dokter F11 - Diagnosis F12 - Ngobatin F13 - Aku boleh pulang ga, dok 😊 ? F14 - Daftar Check-Up F15 - Antrian Saya! F16 - Minum Obat F17 - Minum Penawar D03 - Load D04 - Save B02 - Denah Dinamis	Digunakan untuk struktur data user struktur data ruangan..	karena dapat mengumpulkan atribut-atribut user ke dalam satu adt saja dan menyimpan atribut-atribut ruangan ke dalam satu adt.

ADT Set	F02 - Register F10 - Tambah Dokter	Digunakan untuk menyimpan username sehingga tidak memiliki username yang sama (mencari username unik)..	Karena set tidak bisa memiliki nilai yang sama
ADT Queue	F12 - Ngobatin F14 - Daftar Check-Up F15 - Antrian Saya! D03 - Load D04 - Save	Digunakan untuk menyimpan data antrian pasien di dalam ruangan	Karena Queue menerapkan prinsip antrian dalam dunia nyata.
ADT Matrix	F06 / D01 - Denah Rumah Sakit F09 - Lihat Antrian F10 - Tambah Dokter F11 - Diagnosis F12 - Ngobatin F13 - Aku boleh pulang ga, dok 😊 ? F14 - Daftar Check-Up F15 - Antrian Saya! D03 - Load D04 - Save B02 - Denah Dinamis	Digunakan untuk menyimpan ruanganList matrix.	Karena ruanganList memiliki panjang dan lebar yang dapat disimpan dalam matrix
ADT Stack	F13 - Aku boleh pulang ga, dok 😊 ? F16 - Minum Obat	Digunakan untuk menyimpan isi pasien.	Karena Stack adalah yang LIFO (last in, first out) seperti saat kita

	F17 - Minum Penawar D03 - Load D04 - Save		memasukkan makanan ke dalam perut maka yang terakhir masuk, pertama kali keluar
ADT Map	F12 - Ngobatin	Digunakan untuk menyimpan obat sesuai urutan..	Karena Map dapat menentukan urutan minum obat.
ADT Linked List	F12 - Ngobatin F14 - Daftar Check-Up F15 - Antrian Saya! D03 - Load D04 - Save	Digunakan untuk Queue yang pada bagian antrian menggunakan prinsip linked list (variasi linked list)	Karena dapat membantu pasien.

### **BAB III**

### **DAFTAR PEMBAGIAN KERJA ANGGOTA KELOMPOK**

**Tabel 3.1** Daftar Pembagian Kerja Anggota Kelompok

<b>Fitur</b>	<b>Implementasi *)</b>	<b>NIM Desainer **)</b>	<b>NIM Coder **)</b>	<b>NIM Tester **)</b>

F01 - Login	<pre> function login function getUserIndex function getUserData function getTrombosit function getKadarKolesterolLDL function getKadarKolesterol function getTinggiBadan function getSaturasiOksigen function getKadarGulaDarah function getBeratBadan function getSuhuTubuh function getTekananDarahSistolik function getTekananDarahDiastolik function getDetakJantung function stringToInt function stringToFloat  procedure getUsername procedure getPassword procedure getRole </pre>	13524077	13524077	13524077
-------------	--	----------	----------	----------

	procedure getRiwayatPenyakit function getId procedure getFieldAt			
F02 Register Pasien	Procedure registerUser, Procedure toLowerCase, Function isUsernameUnique, Procedure appendUserToFile function searchMaxID	13524055	13524055	13524055
F03 - Logout	Function logout	13524055	13524055	13524055
F04 - Lupa Password	Function generateUniqueCode Function validasiCode procedure lupaPassword	13524117	13524117	13524117
F05 - Menu & Help	function help procedure print_helpMessage_pasi en procedure print_helpMessage_dokt er procedure print_helpMessage_man ager	13524077	13524077	13524077
F06 / D01	void lihatRuang	13524055	13524055	13524055

Denah Rumah Sakit	void gambarDenah			
F07 - Lihat User	procedure urutanSort procedure sortDescID procedure sortAscID procedure sortDescNama procedure sortAscNama procedure lihatUser procedure lihatPasien procedure lihatDokter procedure interfaceUser	13524097	13524097	13524097
F08 - Cari User	procedure printHorizontalLine function toLowerChar procedure strToLower function compareUsername function binarySearchUsername procedure printHeader procedure printUser_general procedure printPasien procedure printPilihan procedure printDokter procedure cariPasien procedure cariDokter procedure cariUser	13524077	13524077	13524077

F09	Lihat Antrian /D02 -	procedure lihatSemuaAntrian	13524077	13524077	13524077
F10	Tambah Dokter	procedure tambahDokter procedure assignDokter procedure tampilanDokterDanRuang	13524077	13524077	13524077 13524055
F11	Diagnosis	procedure searchRuang function cekSuhu function cekTekananSistolik function cekTekananDiastolik function cekDetakJantung function cekSaturasiOksigen function cekKadarGulaDarah function cekBeratBadan function cekTinggiBadan function cekKadarKolesterol function cekTrombosit procedure cekPenyakit	13524055	13524055	13524055 13524077

	procedure diagnosis			
F12 Ngobatin	procedure ngobatin	13524071	13524071 13524077	13524071 13524077
F13 - Aku boleh pulang ga, dok 😊?	function checkDiagnosis function getIdFromPenyakit function getJumlahObat function checkUrutanObat function urutanHarapan function cariRuanganPasien function cekAntrianPasien procedure akubolehpulang	13524097	13524097 13524077	13524097 13524055 13524077
F14 - Daftar Check-Up	procedure daftarCheckUp	13524071	13524071 13524055 13524077	13524071 13524055 13524077
F15 Antrian Saya!	procedure antrianSaya	13524071	13524071	13524071 13524077
F16 - Minum Obat	procedure minumObat	13524071	13524071 13524055 13524077	13524071 13524055 13524077
F17 - Minum Penawar	Procedure minumPenawar	13524097	13524097	13524097 13524055

				13524077
F18 - Exit	procedure exit_program()	13524137	13524137	13524137
D03 - Load	procedure load_all_data procedure getUserIndex procedure getObatData procedure getObatPenyakitData procedure getPenyakitData procedure getUserData procedure loadConfig	13524077	13524077	13524055 13524077
D04 - Save	function folder_exists function is_folder_empty procedure save_config procedure save_data	13524137	13524137	13524137 13524055 13524077
B02 - Denah Dinamis	procedure ubahDenah procedure pindahDokter	13524055	13524055	13524055

## BAB IV

### **CHECKLIST HASIL RANCANGAN, IMPLEMENTASI, DAN *TESTING***

**Tabel 4.1** Checklist Hasil Rancangan, Implementasi dan *Testing*

<b>Fitur</b>	<b>Desain</b>	<b>Implementasi</b>	<b>Testing</b>
F01 - Login	V	V	V
F02 - Register	V	V	V
F03 - Logout	V	V	V
F04 - Lupa Password	V	V	V
F05 - Menu & Help	V	V	V
F06 / D01 - Denah Rumah Sakit	V	V	V
F07 - Lihat User	V	V	V
F08 - Cari User	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 - Aku boleh pulang ga, 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
D02 - Lihat Antrian	V	V	V
D03 - Load	V	V	V
D04 - Save	V	V	V
B02 - Denah Dinamis	V	V	V

## **BAB V**

### **DESAIN *COMMAND* UNTUK SETIAP PRIMITIF**

#### 1. F1 - Login

*# Kasus 1 : Pasien sudah terdaftar dan password valid*

>>> **LOGIN**

Username: furinadefontaine

Password: fufurinafafa

Login Berhasil! Selamat datang pasien furinadefontaine.

*# Kasus 2: Pasien sudah terdaftar namun password salah*

>>> **LOGIN**

Username: **furinadefontaine**

Password: **watashiwapasswordoforugetto**

Password salah untuk user furinadefontaine.

Lupa password? Ketik [LUPA\_PASSWORD] untuk mengganti password.

*# Kasus 3 : Pasien belum terdaftar*

>>> **LOGIN**

Username: **furinadefontaine**

Password: **watashiwapasswordoforugetto**

Pengguna dengan username "furinadefontaine" belum terdaftar.

Silakan lakukan REGISTER dengan mengetik [REGISTER]

atau melakukan LOGIN dengan akun yang sudah terdaftar.

#### 2. F2 - Register

*# Kasus 1 : Nama pasien dengan format yang benar dan nama unik*

>>> **REGISTER**.

**Username : Jeje**  
**Password : Pass123**

Pasien Jeje berhasil ditambahkan!  
silahkan melakukan login dengan mengetik [LOGIN].

# Kasus 2 : Nama pasien tidak unik

**>>> REGISTER.**

**Username : Jeje**  
**Password : Pass123**

Registrasi gagal! Pasien dengan nama Jeje sudah terdaftar.

#Kasus 3 : Nama pasien tidak sesuai format

**Username : Jeje123**

Masukkan username hanya dengan huruf A-Z, a-z!!

**Username : Jeje**  
**Password : Pass123**

Pasien Jeje berhasil ditambahkan!  
silahkan melakukan login dengan mengetik [LOGIN].

### 3. F3 - Logout

# Kasus 1 : User dalam status USER\_LOGGED\_OFF

**>>> LOGOUT**

Logout gagal!

Anda belum login, silakan login terlebih dahulu sebelum logout.

# Kasus 2 : User dalam status USER\_LOGGED\_ON dan User seorang Pasien

**>>> LOGOUT**

Sampai jumpa, Pasien Jeje!

# Kasus 3 : User dalam status *USER\_LOGGED\_ON* dan User seorang Dokter

>>> **LOGOUT**

Sampai jumpa, Dokter Jeje!

# Kasus 4 : User dalam status *USER\_LOGGED\_ON* dan User seorang Manager

>>> **LOGOUT**

Sampai jumpa, Manager Jeje!

4. F04 - Lupa Password

F04-LUPA\_PASSWORD

>>> LUPA\_PASSWORD

==== Lupa Password ===

Username: soon

Kode Unik: s2on

Password Baru: skibidi

Password berhasil diubah untuk user soon.

Silakan melakukan login kembali dengan mengetik [LOGIN].

5. F05 - Menu & Help

# Kasus 1 : user login sebagai pasien

>>> **HELP**

HELP

=====

=====

Halo pasien furina, berikut adalah seluruh fungsi yang dapat kamu akses:

1. LOGOUT

2. DAFTAR\_CHECKUP

3. ANTRIAN\_SAYA
4. MINUM\_OBAT
5. MINUM\_PENAWAR
6. DOK\_AKU\_MAU\_PULANG
7. LIHAT\_DENAH
8. LIHAT\_RUANGAN
9. LIHAT\_SEMUA\_ANTRIAN
10. EXIT

Footnote:

---

---

1. Untuk menggunakan aplikasi, silahkan masukkan perintah yang terdaftar.
2. Jangan lupa untuk memasukkan input yang valid.

# Kasus 2 : user login sebagai dokter

>>> **HELP**

---

---

HELP

---

---

Halo dokter neronimo, berikut adalah seluruh fungsi yang dapat kamu akses:

1. LOGOUT
2. LIHAT\_ANTRIAN
3. DIAGNOSIS
4. NGOBATIN
5. LIHAT\_DENAH
6. LIHAT\_RUANGAN
7. LIHAT\_SEMUA\_ANTRIAN
8. EXIT

Footnote:

---

---

1. Untuk menggunakan aplikasi, silahkan masukkan perintah yang terdaftar.
2. Jangan lupa untuk memasukkan input yang valid.

# Kasus 3 : user login sebagai manajer

>>> **HELP**

HELP

---

---

Halo manager zedu, berikut adalah seluruh fungsi yang dapat kamu akses:

1. LOGOUT
2. TAMBAH\_DOKTER
3. ASSIGN\_DOKTER
4. LIHAT\_USER
5. LIHAT\_PASIEN
6. LIHAT\_DOKTER
7. CARI\_USER
8. CARI\_PASIEN
9. CARI\_DOKTER
10. LIHAT\_DENAH
11. LIHAT\_RUANGAN
12. LIHAT\_SEMUA\_ANTRIAN
13. UBAH\_DENAH
14. PINDAH\_DOKTER
15. EXIT

Footnote:

---

---

1. Untuk menggunakan aplikasi, silahkan masukkan perintah yang terdaftar.
2. Jangan lupa untuk memasukkan input yang valid.

6. F06/D01 - Denah Rumah Sakit

#Kasus 1 : User ingin melihat denah rumah sakit

>>> **LIHAT\_DENAH**

1	2	3
A   A1   A2   A3		
B   B1   B2   B3		

# Kasus 2 : Ruangan berisi dokter dan pasien

>>> **LIHAT\_RUANGAN**

Masukkan ruangan yang ingin dilihat: **A1**

--- Detail Ruangan A1 ---

Kapasitas : 3

Dokter : Dr. neronimo

Pasien di dalam ruangan:

1. gro
  2. kebin
  3. stewart
- 

#Kasus 3 : Ruangan berisi dokter tanpa pasien

>>> **LIHAT\_RUANGAN**

Masukkan ruangan yang ingin dilihat: **B3**

--- Detail Ruangan B3 ---

Kapasitas : 3

Dokter : Dr. risol

Pasien di dalam ruangan:

1. -

---

# Kasus 4 : Ruangan kosong

>>> **LIHAT\_RUANGAN**

Masukkan ruangan yang ingin dilihat: **B2**

--- Detail Ruangan B2 ---

Kapasitas : 3

Dokter : -

Pasien di dalam ruangan:

Tidak ada pasien di dalam ruangan saat ini.

---

## 7. F07 - Lihat User

# Kasus 1: Melihat data user (user bisa berarti dokter atau pasien) urut berdasarkan ID dan di-sort secara ascending

>>> **LIHAT\_USER**

Urutkan berdasarkan? (1/2)

1. ID

2. Nama

>>> 1

Urutkan Sort? (1/2)

1. ASC (A-Z)

2. DESC (Z-A)

>>> 1

===== DATA USER =====

ID	Nama	Role	Penyakit	
1	stewart	pasien		
2	gro	pasien	COVID-19	

3   kebin		pasien		
+---+-----+-----+-----+				
4   pop		pasien		Diabetes Mellitus
+---+-----+-----+-----+				
5   opor		pasien		
+---+-----+-----+-----+				
6   nikeb		pasien		
+---+-----+-----+-----+				
7   tuart		pasien		
+---+-----+-----+-----+				
8   minonette		pasien		
+---+-----+-----+-----+				
9   tobo		pasien		
+---+-----+-----+-----+				
10   ropik		pasien		
+---+-----+-----+-----+				
10   neronimo		dokter		
+---+-----+-----+-----+				
11   ciciko		dokter		
+---+-----+-----+-----+				
12   cacako		dokter		
+---+-----+-----+-----+				
13   kroket		dokter		
+---+-----+-----+-----+				
15   risol		dokter		
+---+-----+-----+-----+				
16   tobokan		pasien		
+---+-----+-----+-----+				
20   popokan		pasien		
+---+-----+-----+-----+				

88   zeru	manager		
+-----+	+-----+	+-----+	+-----+
100   pendatang	pasien		
+-----+	+-----+	+-----+	+-----+
111   furina	pasien	Diabetes Mellitus	
+-----+	+-----+	+-----+	+-----+

# Kasus 2: Melihat data user (user bisa berarti dokter atau pasien) urut berdasarkan ID dan di-sort secara descending

### >>> LIHAT\_USER

Urutkan berdasarkan? (1/2)

1. ID

2. Nama

>>> 1

Urutkan Sort? (1/2)

1. ASC (A-Z)

2. DESC (Z-A)

>>> 2

===== DATA USER =====

ID	Nama	Role	Penyakit	
+-----+	+-----+	+-----+	+-----+	+-----+
111   furina	pasien	Diabetes Mellitus		
+-----+	+-----+	+-----+	+-----+	+-----+
100   pendatang	pasien			
+-----+	+-----+	+-----+	+-----+	+-----+
88   zeru	manager			
+-----+	+-----+	+-----+	+-----+	+-----+
20   popokan	pasien			
+-----+	+-----+	+-----+	+-----+	+-----+
16   tobokan	pasien			

15   risol	dokter			
+---+-----+-----+-----+				
13   kroket	dokter			
+---+-----+-----+-----+				
12   cacako	dokter			
+---+-----+-----+-----+				
11   ciciko	dokter			
+---+-----+-----+-----+				
10   ropik	pasien			
+---+-----+-----+-----+				
10   neronimo	dokter			
+---+-----+-----+-----+				
9   tobo	pasien			
+---+-----+-----+-----+				
8   minonette	pasien			
+---+-----+-----+-----+				
7   tuart	pasien			
+---+-----+-----+-----+				
6   nikeb	pasien			
+---+-----+-----+-----+				
5   opor	pasien			
+---+-----+-----+-----+				
4   pop	pasien	Diabetes Mellitus		
+---+-----+-----+-----+				
3   kebin	pasien			
+---+-----+-----+-----+				
2   gro	pasien	COVID-19		
+---+-----+-----+-----+				
1   stewart	pasien			
+---+-----+-----+-----+				

# Kasus 3: Melihat data user (user bisa berarti dokter atau pasien) urut berdasarkan Nama dan di-sort secara ascending

>>> **LIHAT\_USER**

Urutkan berdasarkan? (1/2)

1. ID

2. Nama

>>> 2

Urutkan Sort? (1/2)

1. ASC (A-Z)

2. DESC (Z-A)

>>> 1

===== DATA USER =====			
ID   Nama   Role   Penyakit			
12   cacako   dokter			
11   ciciko   dokter			
111   furina   pasien   Diabetes Mellitus			
2   gro   pasien   COVID-19			
3   kebin   pasien			
13   kroket   dokter			
8   minonette   pasien			
10   neronimo   dokter			

6	nikelb	pasien		
+-----+	+-----+	+-----+	+-----+	+-----+
5	opor	pasien		
+-----+	+-----+	+-----+	+-----+	+-----+
100	pendatang	pasien		
+-----+	+-----+	+-----+	+-----+	+-----+
4	pop	pasien	Diabetes Mellitus	
+-----+	+-----+	+-----+	+-----+	+-----+
20	popokan	pasien		
+-----+	+-----+	+-----+	+-----+	+-----+
15	risol	dokter		
+-----+	+-----+	+-----+	+-----+	+-----+
10	ropik	pasien		
+-----+	+-----+	+-----+	+-----+	+-----+
1	stewart	pasien		
+-----+	+-----+	+-----+	+-----+	+-----+
9	tobo	pasien		
+-----+	+-----+	+-----+	+-----+	+-----+
16	tobokan	pasien		
+-----+	+-----+	+-----+	+-----+	+-----+
7	tuart	pasien		
+-----+	+-----+	+-----+	+-----+	+-----+
88	zeru	manager		
+-----+	+-----+	+-----+	+-----+	+-----+

# Kasus 4: Melihat data user (user bisa berarti dokter atau pasien) urut berdasarkan Nama dan di-sort secara descending

>>> **LIHAT\_USER**

Urutkan berdasarkan? (1/2)

1. ID

2. Nama

>>> 2

Urutkan Sort? (1/2)

1. ASC (A-Z)

2. DESC (Z-A)

>>> 2

===== DATA USER =====

ID   Nama	Role	Penyakit
88   zeru	manager	
7   tuart	pasien	
16   tobokan	pasien	
9   tobo	pasien	
1   stewart	pasien	
10   ropik	pasien	
15   risol	dokter	
20   popokan	pasien	
4   pop	pasien	Diabetes Mellitus
100   pendatang	pasien	
5   opor	pasien	

6	nikeb	pasien		
+---+	-----+	-----+	-----+	-----+
10	neronimo	dokter		
+---+	-----+	-----+	-----+	-----+
8	minonette	pasien		
+---+	-----+	-----+	-----+	-----+
13	kroket	dokter		
+---+	-----+	-----+	-----+	-----+
3	kebin	pasien		
+---+	-----+	-----+	-----+	-----+
2	gro	pasien	COVID-19	
+---+	-----+	-----+	-----+	-----+
111	furina	pasien	Diabetes Mellitus	
+---+	-----+	-----+	-----+	-----+
11	ciciko	dokter		
+---+	-----+	-----+	-----+	-----+
12	cacako	dokter		
+---+	-----+	-----+	-----+	-----+

# Kasus 5: Melihat data pasien berdasarkan ID dan di-sort secara ascending

>>> **LIHAT\_PASIEN**

Urutkan berdasarkan? (1/2)

1. ID

2. Nama

>>> 1

Urutkan Sort? (1/2)

1. ASC (A-Z)

2. DESC (Z-A)

>>> 1

===== DATA PASIEN =====

+---+-----+-----+

ID	Nama	Penyakit
1	stewart	
2	gro	COVID-19
3	kebin	
4	pop	Diabetes Mellitus
5	opor	
6	nikeb	
7	tuart	
8	minonette	
9	tobo	
10	ropik	
16	tobokan	
20	popokan	
100	pendatang	
111	furina	Diabetes Mellitus

# Kasus 6: Melihat data pasien berdasarkan ID dan di-sort secara descending

>>> **LIHAT PASIEN**

Urutkan berdasarkan? (1/2)

1. ID

2. Nama

>>> 1

Urutkan Sort? (1/2)

1. ASC (A-Z)

2. DESC (Z-A)

>>> 2

===== DATA PASIEN =====

ID	Nama	Penyakit
111	furina	Diabetes Mellitus
100	pendatang	
20	popokan	
16	tobokan	
10	ropik	
9	tobo	
8	minonette	
7	tuart	

6	nikeb			
+---+-----+				-----+
5	opor			
+---+-----+				-----+
4	pop		Diabetes Mellitus	
+---+-----+				-----+
3	kebin			
+---+-----+				-----+
2	gro		COVID-19	
+---+-----+				-----+
1	stewart			
+---+-----+				-----+

# Kasus 7: Melihat data pasien berdasarkan Nama dan di-sort secara ascending

>>> **LIHAT\_PASIEN**

Urutkan berdasarkan? (1/2)

1. ID
2. Nama

>>> 2

Urutkan Sort? (1/2)

1. ASC (A-Z)
2. DESC (Z-A)

>>> 1

===== DATA PASIEN =====

ID	Nama	Penyakit	
+---+-----+			-----+
111	furina	Diabetes Mellitus	
+---+-----+			-----+
2	gro	COVID-19	
+---+-----+			-----+

3   kebin			
+-----+	+-----+	+-----+	
8   minonette			
+-----+	+-----+	+-----+	
6   nikeb			
+-----+	+-----+	+-----+	
5   opor			
+-----+	+-----+	+-----+	
100   pendatang			
+-----+	+-----+	+-----+	
4   pop		Diabetes Mellitus	
+-----+	+-----+	+-----+	
20   popokan			
+-----+	+-----+	+-----+	
10   ropik			
+-----+	+-----+	+-----+	
1   stewart			
+-----+	+-----+	+-----+	
9   tobo			
+-----+	+-----+	+-----+	
16   tobokan			
+-----+	+-----+	+-----+	
7   tuart			
+-----+	+-----+	+-----+	

# Kasus 8: Melihat data pasien berdasarkan Nama dan di-sort secara descending

>>> **LIHAT\_PASIEN**

Urutkan berdasarkan? (1/2)

1. ID

2. Nama

>>> 2

Urutkan Sort? (1/2)

1. ASC (A-Z)

2. DESC (Z-A)

>>> 2

===== DATA PASIEN =====

ID	Nama	Penyakit
7	tuart	
16	tobokan	
9	tobo	
1	stewart	
10	ropik	
20	popokan	
4	pop	Diabetes Mellitus
100	pendatang	
5	opor	
6	nikeb	
8	minonette	

3   kebin		
+-----+	-----+	-----+
2   gro	COVID-19	
+-----+	-----+	-----+
111   furina	Diabetes Mellitus	
+-----+	-----+	-----+

# Kasus 9: Melihat data dokter berdasarkan ID dan di-sort secara ascending

### >>> LIHAT\_DOKTER

Urutkan berdasarkan? (1/2)

1. ID

2. Nama

>>> 1

Urutkan Sort? (1/2)

1. ASC (A-Z)

2. DESC (Z-A)

>>> 1

### ===== DATA DOKTER =====

+-----+	-----+
ID   Nama	
+-----+	-----+
10   neronimo	
+-----+	-----+
11   ciciko	
+-----+	-----+
12   cacako	
+-----+	-----+
13   kroket	
+-----+	-----+

15   risol	
+-----+	+-----+

# Kasus 10: Melihat data dokter berdasarkan ID dan di-sort secara descending

>>> **LIHAT\_DOKTER**

Urutkan berdasarkan? (1/2)

1. ID

2. Nama

>>> 1

Urutkan Sort? (1/2)

1. ASC (A-Z)

2. DESC (Z-A)

>>> 2

===== DATA DOKTER =====

+-----+	+-----+
ID   Nama	
+-----+	+-----+
15   risol	
+-----+	+-----+
13   kroket	
+-----+	+-----+
12   cacako	
+-----+	+-----+
11   ciciko	
+-----+	+-----+
10   neronimo	
+-----+	+-----+

# Kasus 11: Melihat data dokter berdasarkan Nama dan di-sort secara ascending

>>> **LIHAT\_DOKTER**

Urutkan berdasarkan? (1/2)

1. ID

2. Nama

>>> 2

Urutkan Sort? (1/2)

1. ASC (A-Z)

2. DESC (Z-A)

>>> 1

===== DATA DOKTER =====

+-----+
ID   Nama
+-----+
12   cacako
+-----+
11   ciciko
+-----+
13   kroket
+-----+
10   neronimo
+-----+
15   risol
+-----+

# Kasus 12: Melihat data dokter berdasarkan Nama dan di-sort secara descending

>>> **LIHAT\_DOKTER**

Urutkan berdasarkan? (1/2)

1. ID

2. Nama

>>> 2

Urutkan Sort? (1/2)

```

1. ASC (A-Z)
2. DESC (Z-A)

>>> 2

===== DATA DOKTER =====

+-----+
| ID | Nama      |
+-----+
| 15 | risol      |
+-----+
| 10 | neronimo   |
+-----+
| 13 | kroket     |
+-----+
| 11 | ciciko     |
+-----+
| 12 | cacako     |
+-----+

```

# Kasus 13: input tidak valid

## 8. F08 - Cari User

# Kasus 1 : Cari user berdasarkan ID dan pasien sudah terdaftar di database

>>> **CARI\_USER**

Cari berdasarkan:

1. ID
2. Nama

>>> 1

Cari ID: 2

Pengguna dengan ID 2:

```
+-----+-----+-----+
```

ID	Nama	Role	Penyakit
2	gro	pasien	COVID-19

# Kasus 2 : Cari user berdasarkan nama dan user sudah terdaftar di database

>>> **CARI\_USER**

Cari berdasarkan:

1. ID
2. Nama

>>> 2

Cari Username: grok

User dengan username 'grok' tidak ditemukan.

# Kasus 3 : Cari user berdasarkan ID dan user belum terdaftar

>>> **CARI\_USER**

Cari berdasarkan:

1. ID
2. Nama

>>> 1

Cari ID: 99

ID user tidak ditemukan!

# Kasus 4 : Cari user berdasarkan nama dan user belum terdaftar

>>> **CARI\_USER**

Cari berdasarkan:

1. ID
2. Nama

>>> 2

Cari Username: prikeun

User dengan username 'prikeun' tidak ditemukan.

# Kasus 5 : Cari pasien dan pasien sudah terdaftar di database

>>> **CARI\_PASIEN**

Cari berdasarkan:

1. ID
2. Nama
3. Penyakit

>>> 1

Cari ID: 1

Pengguna dengan ID 1:

ID	Nama	Role	Penyakit
1	stewart	pasien	

# Kasus 6 : Cari user berdasarkan penyakit

>>> **CARI\_PASIEN**

Cari berdasarkan:

1. ID
2. Nama
3. Penyakit

>>> 3

Cari penyakit: Diabetes Mellitus

ID	Nama	Penyakit
4	pop	Diabetes Mellitus
111	furina	Diabetes Mellitus

# Kasus 7: Cari dokter berdasarkan nama

>>> **CARI\_DOKTER**

Cari berdasarkan:

1. ID
2. Nama

Masukkan angka pilihan:

>>> 2

Cari Username: neronimo

Menampilkan data Dokter dengan nama neronimo:

+-----+	+-----+
ID   Nama	
+-----+	+-----+
10   neronimo	
+-----+	+-----+

# Kasus 7 : Cari dokter yang belum terdaftar

>>> **CARI\_DOKTER**

Cari berdasarkan:

1. ID
2. Nama

Masukkan angka pilihan:

>>> 2

Cari Username: dok

Dokter dengan username 'dok' tidak ditemukan.

9. 9a. F10 - Tambah Dokter

# Kasus 1 : Dokter belum terdaftar

>>> **TAMBAH\_DOKTER**

Masukkan username dokter baru: doktermagang

Masukkan password dokter baru: admin123

# Kasus 2 : Dokter sudah terdaftar

>>> **TAMBAH\_DOKTER**

Masukkan username dokter baru: doktermagang

Username sudah digunakan. Silakan pilih username lain.

#### 9b. Assign Dokter

# Kasus 1 : Dokter sudah terdaftar dan ruangan yang ingin diisi valid dan kosong

>>> **ASSIGN\_DOKTER**

Masukkan username dokter yang akan diassign: doktermagang

Masukkan kode ruangan (misal A1, B3, dst): B2

Dokter doktermagang berhasil diassign ke ruangan B2.

# Kasus 1 : Dokter sudah terdaftar dan sudah diassign

>>> **ASSIGN\_DOKTER**

Masukkan username dokter yang akan diassign: doktermagang

Dokter sudah diassign ke ruangan B2.

# Kasus 1 : Dokter belum terdaftar

>>> **ASSIGN\_DOKTER**

Masukkan username dokter yang akan diassign: dokteraja

Dokter dengan username dokteraja tidak ditemukan.

# Kasus 1 : Dokter sudah terdaftar dan ruangan yang ingin diisi sudah ditempati dokter lain

>>> **ASSIGN\_DOKTER**

Masukkan username dokter yang akan diassign: dokteretkod

Masukkan kode ruangan (misal A1, B3, dst): B2

Ruangan B2 sudah diisi oleh dokter doktermagang.

# Kasus 1 : Dokter sudah terdaftar dan ruangan yang ingin diisi tidak valid

>>> **ASSIGN\_DOKTER**

Masukkan username dokter yang akan diassign: dokteretkod

Masukkan kode ruangan (misal A1, B3, dst): B4  
Kolom tidak valid. Harus antara 1 hingga 3.

10. F11 - Diagnosis

# Kasus 1 : Dokter memiliki pasien yang perlu diperiksa

>>> **DIAGNOSIS**

Budi terdiagnosa penyakit Maag!

# Kasus 2 : Antrian pasien sudah kosong dan tidak ada pasien yang perlu diperiksa

>>> **DIAGNOSIS**

Tidak ada pasien untuk diperiksa!

# Kasus 3: Pasien tidak terjangkit penyakit apapun setelah diperiksa. Jika terdapat kasus ini maka pasien diperbolehkan pulang.

>>> **DIAGNOSIS**

Budi tidak terdiagnosa penyakit apapun!

# Kasus 4: Pasien telah didiagnosis sebelumnya

>>> **DIAGNOSIS**

Pasian telah terdiagnosa, arahkan pasien menuju ke Ngobatin.

11. F12 - Ngobatin

# Kasus 1: Dokter belum ditempatkan di ruangan mana pun

>>> **NGOBATIN**

Dokternya belum ditempatkan di ruangan mana pun. Tolong assign dulu ya!

# Kasus 2: Dokter tidak memiliki pasien dalam antrian

>>> **NGOBATIN**

Belum ada pasien nih di antrian ruangan dokter 'risol'. Santai dulu dok!

# Kasus 3: Pasien belum punya diagnosis penyakit

>>> **NGOBATIN**

Pasien ini belum ada diagnosis penyakit, dok. Kasih diagnosis dulu ya biar bisa dikasih obat yang tepat.

# Kasus 4: Berhasil memberikan obat ke pasien

>>> **NGOBATIN**

Pasien 'pop' terdiagnosis menderita Diabetes Mellitus.

Berikut daftar obat yang harus diberikan, urutannya jangan sampai kebalik ya!

>> Obat ke-1: Metformin

>> Obat ke-2: Lisinopril

>> Obat ke-3: Remdesivir

>> Obat ke-4: Vitamin C

Pastikan obat diberikan satu per satu sesuai urutan di atas ya. Semangat menyembuhkan pasien!

12. F13 - Aku boleh pulang ga, Dok?

# Kasus 1: Pasien belum diberikan diagnosa penyakit oleh dokter

>>> **DOK\_AKU\_MAU\_PULANG**

Kamu belum menerima diagnosis apapun dari dokter, jangan buru-buru pulang!

T^T

# Kasus 2: Pasien belum menghabiskan seluruh obat yang diberikan kepadanya

>>> **DOK\_AKU\_MAU\_PULANG**

Dokter sedang memeriksa keadaanmu...

Masih ada obat yang belum kamu habiskan, minum semuanya dulu yukk! :3

# Kasus 3: Pasien sudah menghabiskan obat, namun terdapat urutan yang salah dalam konsumsinya

>>> **DOK\_AKU\_MAU\_PULANG**

Dokter sedang memeriksa keadaanmu...

Maaf, tapi kamu masih belum bisa pulang T^T !

Urutan MINUM\_OBAT yang diharapkan:

Metformin -> Lisinopril -> Remdesivir -> Vitamin C

Urutan MINUM\_OBAT yang kamu minum:

Lisinopril-> Metformin -> Vitamin C -> Remdesivir

Silahkan kunjungi dokter untuk meminta penawar (MINUM\_PENAWAR) yang sesuai!

# Kasus 4: Pasien sudah menghabiskan obat, dan semuanya valid.

>>> **DOK\_AKU\_MAU\_PULANG**

Dokter sedang memeriksa keadaanmu...

SELAMAT!! :D

Kamu sudah dinyatakan sembuh oleh dokter. Silahkan pulang dan semoga sehat selalu! :3

# Kasus 5: Pasien belum ada di antrian paling depan

>>> **DOK\_AKU\_MAU\_PULANG**

Kamu belum berada di urutan terdepan antrian mana pun. Sabar dulu ya! :(

# Kasus 6: Pasien tidak ditemukan di antrian / ruangan mana pun

>>> **DOK\_AKU\_MAU\_PULANG**

Pasien tidak ditemukan. Silahkan melakukan registrasi terlebih dahulu :)

# Kasus 7: Pasien yang telah terdiagnosis “Sehat” / tidak ada penyakit apa pun

>>> **DOK\_AKU\_MAU\_PULANG**

Kamu tidak ada penyakit apapun. Silahkan pulang dan semoga sehat selalu! :3

13. F14 - Daftar Check-Up

# Kasus 1: Pasien sudah terdaftar di antrian

>>> **DAFTAR\_CHECKUP**

Eh, kamu udah daftar buat check-up nih!

Yuk, tunggu giliran dulu. Nanti aja daftarnya lagi.

# Kasus 2: Berhasil mendaftar dan memilih dokter

>>> **DAFTAR\_CHECKUP**

--- FORMULIR CHECK-UP PASIEN ---

Masukin data kesehatan kamu dulu ya. Jangan asal, ini penting!

Suhu Tubuh (°C): **37.5**

Tekanan Darah (sistol dan diastol): **80 120**

Detak Jantung (bpm): **60**

Saturasi Oksigen (%): **5**

Gula Darah (mg/dL): **100**

Berat Badan (kg): **55**

Tinggi Badan (cm): **165**

Kolesterol Total (mg/dL): **5**

Trombosit (ribu/µL): **5**

--- DOKTER YANG SIAP NANGGEPIN KAMU ---

1. Dr. neronimo - Ruangan A1 (Antrian: 5 pasien)

2. Dr. ciciko - Ruangan A2 (Antrian: 2 pasien)

3. Dr. cacako - Ruangan A3 (Antrian: 1 pasien)

4. Dr. kroket - Ruangan B1 (Antrian: 2 pasien)

5. Dr. risol - Ruangan B3 (Antrian: 0 pasien)

Pilih dokter (1 - 5): **3**

Mantap! Check-up kamu udah masuk daftar.

Antrian kamu sekarang di Dr. cacako - Ruangan A3

Nomor urut kamu: 2

14. F15 - Antrian Saya!

# Kasus 1: Pasien belum ada di antrian mana pun

>>> **ANTRIAN\_SAYA**

Eits, kamu belum ada di antrian mana pun nih!

Yuk daftar dulu buat check-up lewat command: DAFTAR\_CHECKUP

# Kasus 2: Pasien sedang menunggu di luar ruangan

>>> **ANTRIAN\_SAYA**

==== STATUS ANTRIAN KAMU ====

Ketemu! Kamu lagi nunggu di ruangan A1 bareng Dokter neronimo.

Posisimu sekarang di antrian: 6 dari 6 pasien.

Kamu masih DI LUAR ruangan. Tunggu 3 pasien lagi buat masuk, ya!

# Kasus 3: Pasien sudah di dalam ruangan

>>> **ANTRIAN\_SAYA**

==== STATUS ANTRIAN KAMU ====

Ketemu! Kamu lagi nunggu di ruangan A3 bareng Dokter cacako.

Posisimu sekarang di antrian: 2 dari 2 pasien.

Kamu udah DI DALAM ruangan. Siap-siap ya, giliranmu bentar lagi!

15. F16 - Minum Obat

# Kasus 1: Inventory obat pasien kosong

>>> **MINUM\_OBAT**

Eh, obatmu kosong loh! Gak ada yang bisa diminum sekarang :)

# Kasus 2: Pasien memilih nomor obat yang tidak valid

>>> **MINUM\_OBAT**

===== DAFTAR OBAT =====

1. Remdesivir

Pilih nomor obat yang mau kamu minum (1-1): **2**

Waduh! Nomor yang kamu pilih gak ada di daftar!

# Kasus 3: Pasien berhasil minum obat

>>> **MINUM\_OBAT**

===== DAFTAR OBAT =====

1. Remdesivir

Pilih nomor obat yang mau kamu minum (1-2): **1**

\*GLEKGLEKGLEK\*... Remdesivir berhasil diminum! Rasanya... unik ya^^

Inventory diperbarui. Obat Remdesivir udah keluar dari kantong ajaib kamu!

16. F17 - Minum Penawar

# Kasus 1: Obat terakhir yang diminum adalah Lisinopril

>>> **MINUM\_PENAWAR**

Uwekkk :O !!!

Inventory diperbarui. Obat Lisinopril keluar dan kembali ke kantong ajaib kamu!

# Kasus 2: Belum ada obat yang diminum

>>> **MINUM\_PENAWAR**

Perut kosong!! Belum ada obat yang dimakan. >:)

# Kasus 3: Pasien melakukan MINUM\_PENAWAR tetapi belum terdiagnosis / belum mendapatkan obat

>>> **MINUM\_PENAWAR**

Kamu belum menerima obat dari dokter. Tidak boleh meminta obat penawar!! >:)

17. F18 - Exit

# Kasus 1 : input valid pertama

>>> **EXIT**

Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): **ya**  
(melanjutkan spesifikasi save)

# Kasus 2 : input valid kedua

>>> **EXIT**

Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): **tidak**

# Kasus 3 : input tidak valid

>>> **EXIT**

Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak):  
**PGPGPGPGPGPGPG**

Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.

>>> **EXIT**

Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak):

**AA**  
**AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA**  
**AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA**  
**AAAAAAAAAAAAAA**

Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.

Segmentation fault (core dumped)

18. D02 - Lihat Semua Antrian

>>> **LIHAT\_SEMUA\_ANTRIAN**

1 2 3

+----+----+----+

A | A1 | A2 | A3 |

+-----+-----+-----+

B | B1 | B2 | B3 |

+-----+-----+-----+

===== A1 =====

Kapasitas : 3

Dokter : Dr. neronimo

Pasien di dalam ruangan:

1. gro
2. kebin
3. stewart

Pasien di antrian:

1. tobokan
2. popokan

===== A2 =====

Kapasitas : 3

Dokter : Dr. ciciko

Pasien di dalam ruangan:

1. pop
2. opor

Pasien di antrian:

Tidak ada pasien di antrian saat ini.

===== A3 =====

Kapasitas : 3

Dokter : Dr. cacako

Pasien di dalam ruangan:

1. nikeb

Pasien di antrian:

Tidak ada pasien di antrian saat ini.

===== B1 =====

Kapasitas : 3

Dokter : Dr. kroket

Pasien di dalam ruangan:

1. minonette

2. tuart

Pasien di antrian:

Tidak ada pasien di antrian saat ini.

===== B2 =====

Kapasitas : 3

Dokter : Dr. doktermagang

Pasien di dalam ruangan:

Tidak ada pasien di dalam ruangan saat ini.

Pasien di antrian:

Tidak ada pasien di antrian saat ini.

===== B3 =====

Kapasitas : 3

Dokter : Dr. risol

Pasien di dalam ruangan:

1. -

Pasien di antrian:

Tidak ada pasien di antrian saat ini.

## 19. D03 - Load

# Setelah file dicompile menggunakan makefile, program dapat dijalankan dengan meload file data terlebih dahulu dengan format ./<nama program.exe><namafolderdata>

# Kasus 1 : folder ditemukan dan data berhasil dibaca

>./main.exe data

Loading...

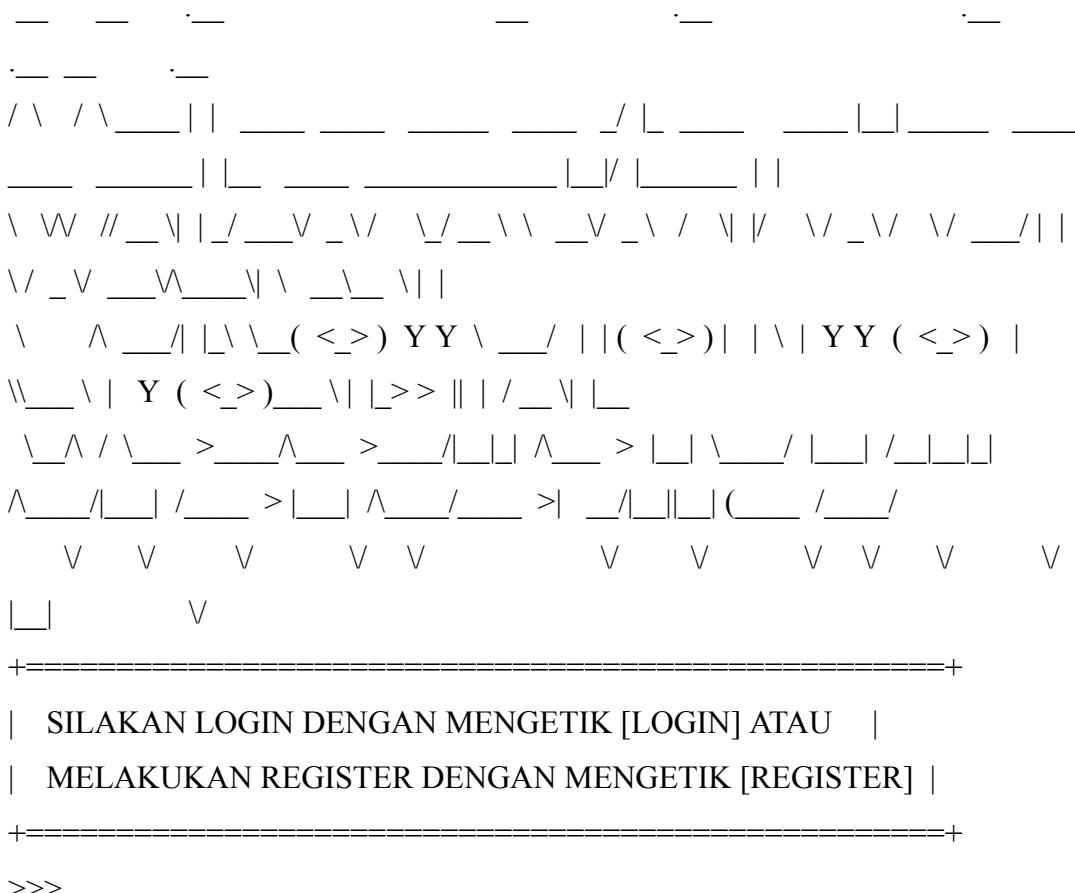
Memasukkan obat 4 ke perut pasien pop

Memasukkan obat 5 ke perut pasien pop

Konfigurasi berhasil dimuat dari data/config.txt

Data berhasil diload dari folder: data

Data berhasil dimuat dari folder: data



# Kasus 2 : pengguna tidak memasukkan nama folder

\$./main.exe

Usage: main.exe <nama folder>

# Kasus 3 : folder data tidak ada/tidak dapat dibaca

\$ ./main.exe folder

Loading...

Fatal error: Gagal membuka file folder/user.csv

20. D04 - Save

# Kasus 1 : Folder belum ada

>>> EXIT

Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): ya

Masukkan nama folder untuk menyimpan data: **09-03-2024**

Folder "09-03-2024" belum ada. Membuat folder baru...

Berhasil menyimpan data di folder 09-03-2024.

Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.

# Kasus 2 : Folder sudah ada, baik ada isinya atau tidak (kosong)

>>> EXIT

Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): ya

Masukkan nama folder untuk menyimpan data: **09-03-2024**

Berhasil menyimpan data di folder 09-03-2024.

Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.

# Kasus 3 : Nama folder sangat panjang

>>> EXIT

Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): ya

Masukkan nama folder untuk menyimpan data:

AA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAA\

Berhasil menyimpan data di folder

AAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAA

AAAAAAA.

Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.

# Kasus 4 : Nama folder terdapat spasi

>>> **EXIT**

Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): **ya**

Masukkan nama folder untuk menyimpan data: **INI SPASI**

Folder "INI SPASI" belum ada. Membuat folder baru...

Berhasil menyimpan data di folder INI SPASI.

Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.

## 21. B02 - Denah Dinamis

### 21.1. Ubah Denah

# Kasus 1 : Ukuran denah yang baru valid

>>> **UBAH\_DENAH**

Masukkan ukuran denah yang baru: **3 3**

Denah rumah sakit berhasil diubah menjadi 3 baris dan 3 kolom.

# Kasus 2 : Ukuran denah yang baru tidak valid

>>> **UBAH\_DENAH**

Masukkan ukuran denah yang baru: **2 2**

Tidak dapat mengubah ukuran denah. Ruangan A3 masih ditempati oleh Dr. cacako. Silakan pindahkan dokter terlebih dahulu.

## 21.2. Pindah Dokter

# Kasus 1 : Ruangan Tujuan dokter pindah kosong

>>> **PINDAH\_DOKTER**

Masukkan ruangan asal dan tujuan (misal: A1 B2): **A1 B2**

Dr. neronimo berhasil dipindahkan dari ruangan A1 ke ruangan B2.

# Kasus 2 : Ruangan tujuan dokter pindah sudah ditempati dokter lain

>>> **PINDAH\_DOKTER**

Masukkan ruangan asal dan tujuan (misal: A1 B2): **A1 A2**

Pemindahan gagal. Ruangan A2 Sudah ditempati.

# Kasus 3 : Ruangan awal dokter pindah kosong

>>> **PINDAH\_DOKTER**

Masukkan ruangan asal dan tujuan (misal: A1 B2): **B2 A1**

Pemindahan gagal. Ruangan B2 Kosong.

# Kasus 4 : Ruangan tidak terdefinisi (tidak ada)

>>> **PINDAH\_DOKTER**

Masukkan ruangan asal dan tujuan (misal: A1 B2): **A3 B4**

Pemindahan gagal. Ruangan B4 Tidak Ada.



## BAB VI

### DESAIN KAMUS DATA

#### 1. ADT Sederhana

##### 1.1. ADT User

Kamus

```
ruanganList: array [0..panjang_denah-1,
0..lebar_denah-1] of Ruangan
{ Matriks denah rumah sakit }

panjang_denah: integer
{ Banyak baris denah rumah sakit }

lebar_denah: integer
{ Banyak kolom denah rumah sakit }

constant MAX_USERNAME : integer = 50
constant MAX_PASSWORD : integer = 50
constant MAX_ROLE : integer = 10
constant MAX_RIWAYAT : integer = 100
constant MAX_USERS : integer = 100

type User : <
    id: integer;
    {ID unik pengguna}
    username: string;
    {Nama pengguna, huruf saja, max 50 karakter}
    password: string;
    {Kata sandi akun, max 50 karakter}
    role: string;
    {Peran pengguna: "manager", "dokter", atau "pasien"}
    riwayat_penyakit: string;
    {Nama penyakit yang terakhir terdiagnosis, max 100
karakter}
```

```

suhu_tubuh: real;
{ Suhu tubuh (°C) }

tekanan_darah_sistolik: integer;
{Tekanan darah bagian atas (mmHg) }

tekanan_darah_diastolik: integer;
{Tekanan darah bagian bawah (mmHg) }

detak_jantung: integer;
{Detak jantung (bpm) }

saturasi_oksigen: real;
{Kadar oksigen dalam darah (%) }

kadar_gula_darah: integer;
{Gula darah (mg/dL) }

berat_badan: real;
{Berat badan (kg) }

tinggi_badan: integer;
{Tinggi badan (cm) }

kadar_kolesterol: integer;
{Kolesterol total (mg/dL) }

trombosit: integer;
{Jumlah trombosit (ribu/µL) }

perut: Stack;
{Riwayat makanan yang dimakan (tipe stack)}>

```

#### Kamus Global

```

users: array [0..MAX_USERS-1] of User
{ Daftar seluruh user terdaftar }

userCount: integer
{ Banyaknya user yang sudah terdaftar }

```

#### 1.2. ADT Ruangan

```
constant MAX_PASIEN : integer = 10
```

```
constant MAX_RUANGAN : integer = 100
```

### Kamus

```
ruanganList: array [0..panjang_denah-1] of
array [0..lebar_denah-1] of Ruangan
{ Matriks denah rumah sakit }
panjang_denah: integer
{ Banyak baris denah rumah sakit }
lebar_denah: integer
{ Banyak kolom denah rumah sakit }

type Ruangan : <
    idDokter: integer;
    { ID dokter yang bertugas di ruangan }
    usernameDokter: string;
    { Username dokter yang bertugas }
    idPasien: array [0..MAX_PASIEN-1] of integer;
    { ID pasien yang sedang berada di ruangan }
    usernamePasien: array [0..MAX_PASIEN-1] of
string;
    { Username pasien yang sedang berada di ruangan }
    totalPasien: integer;
    { Jumlah pasien dalam ruangan }
    antrianPasien: Queue;
    { Antrian pasien yang menunggu }>
```

### 1.3. ADT Inventory

```
constant MAX_INVENTORY : integer = 100
```

```

type InventoryPasien : <
    pasien_id: integer;
        { ID dari pasien pemilik inventory }
    obat_id: array [0..MAX_INVENTORY-1] of
    integer;
        { Daftar ID obat yang dimiliki }
    count: integer;
        { Banyaknya item obat yang dimiliki }>

```

Kamus Global

```

    inventoryList: array [0..MAX_USERS-1] of
    InventoryPasien
        { Daftar inventory untuk semua pasien }

```

**procedure** initInventory()

{I.S. inventoryPasien mungkin belum berisi data yang valid.}

{F.S.

- inventoryPasien.count bernilai 0
- Semua elemen obat\_id dianggap kosong
- 
- pasien\_id diinisialisasi dengan nilai default (misalnya -1) }

**procedure** addInventory(input id\_obat : integer)

{I.S.

- inventoryPasien telah terdefinisi
- count menyatakan jumlah item saat ini}

{F.S.

- `id_obat` ditambahkan ke indeks `obat_id[count]`
- `count` bertambah 1}

```
procedure removeInventory(input index : integer)
{I.S.
  • inventoryPasien berisi daftar obat
  • index adalah posisi obat yang akan dihapus (0
    ≤ index < count)}
{F.S.
  • Obat di indeks index dihapus
  • Elemen setelah index digeser ke kiri
  • count berkurang 1}
```

```
procedure printInventory()
{I.S. inventoryPasien berisi daftar obat yang
dimiliki pasien}
{F.S. Menampilkan isi obat_id sebanyak count ke
layar}
```

```
function getInventoryByUser(user : User) ->
InventoryPasien
{I.S.
  • user telah login
  • Sistem memiliki daftar semua inventory (jika
ada)}
{F.S.
  • Mengembalikan pointer ke InventoryPasien milik
user tersebut}
```

- (jika tidak ada, bisa return NULL atau pointer baru tergantung implementasi) }

```
procedure removeInventoryByUser(input/ouput inv :  
InventoryPasien, input index : integer)  
{I.S.  
    • inv menunjuk ke data inventory milik seorang  
        pasien  
    • index adalah indeks obat yang akan dihapus}  
  
{F.S.  
    • Obat di indeks index dihapus dari inv->obat_id  
    • inv->count berkurang 1, elemen setelahnya  
        digeser ke kiri}
```

```
procedure searchJumlahInventory(input/output  
jumlahInventory : integer)  
{I.S. inventoryPasien.count telah terdefinisi}  
{F.S. jumlahInventory diisi dengan jumlah item  
dalam inventory (yaitu count)}
```

#### 1.4. ADT penyakit

```
constant MAX_NAMA_PENYAKIT : integer = 50  
constant MAX_PENYAKIT : integer = 50
```

```
type Penyakit : <  
    id : integer ;  
    { ID unik penyakit }  
    name_penyakit : string ;
```

```

{ Nama penyakit (maks 50 karakter) }

suhu_tubuh_min: real;
suhu_tubuh_max: real;
{ Rentang suhu tubuh normal (°C)
tekanan_darah_sistolik_min : integer;
tekanan_darah_sistolik_max : integer;
{ Tekanan darah atas (mmHg)
tekanan_darah_diastolik_min : integer;
tekanan_darah_diastolik_max : integer;
{ Tekanan darah bawah (mmHg)
detak_jantung_min: integer;
detak_jantung_max: integer;
{ Detak jantung (bpm)
saturasi_oksigen_min : real;
saturasi_oksigen_max : real;
{ Saturasi oksigen (%)
kadar_gula_darah_min : integer;
kadar_gula_darah_max : integer;
{ Gula darah (mg/dL)
berat_badan_min : real;
berat_badan_max : real;
{ Berat badan (kg)
tinggi_badan_min : integer;
tinggi_badan_max : integer;
{ Tinggi badan (cm)
kadar_kolesterol_min : integer;
kadar_kolesterol_max : integer;
{ Kolesterol total (mg/dL)
trombosit_min : integer;
trombosit_max : integer; { 
```

```
Trombosit (ribu/ $\mu$ L) }>
```

#### KAMUS GLOBAL

```
penyakitList : array [0..MAX_PENYAKIT-1] of  
Penyakit  
{ Daftar seluruh penyakit yang ada pada sistem }  
penyakitCount: integer  
{ Jumlah data penyakit yang dibaca dari file eksternal  
}
```

#### 1.5. ADT Obat Penyakit

```
constant MAX_OBAT_PER_PENYAKIT : integer = 20  
{ Maksimum jumlah obat berbeda yang bisa diresepkan untuk 1  
penyakit }
```

```
type ObatPenyakit : <  
    obat_id          : integer  
    { ID dari obat (unik di list obat) }  
    penyakit_id      : integer  
    { ID penyakit yang membutuhkan obat ini }  
    urutan_minum     : integer  
    { Urutan konsumsi obat terhadap penyakit terkait }>
```

#### KAMUS GLOBAL

```
obatPenyakitList : array [0..MAX_OBAT_PER_PENYAKIT  
* MAX_PENYAKIT - 1] of ObatPenyakit  
{ Daftar relasi antara obat dan penyakit }
```

```
obatPenyakitCount : integer  
{ Jumlah data relasi obat-penyakit yang dimuat dari  
file }
```

## 2. ADT Set

```
constant MAX_SET_SIZE: integer = 1000
constant MAX_STRING_LENGTH: integer = 512

type Set:
    <elements: array [0..MAX_SET_SIZE] of array
        [0..MAX_STRING_LENGTH] of string;
    { Menyimpan elemen-elemen set, berupa array string }
    size: integer
    { Banyak elemen dalam set (0 ≤ size ≤ MAX_SET_SIZE) }>

procedure createEmptySet(input/output s:set)
{I.S. Set s terdefinisi, str adalah string input}
{F.S. Mengembalikan 1 jika str ada dalam
s.elements, 0 jika tidak ada}

function contains(input s:set, input str:string) ->
integer
{I.S. Set s mungkin sudah berisi beberapa elemen
str adalah string yang ingin dimasukkan}
{F.S. Jika str belum ada di s.elements, str
ditambahkan dan s.size bertambah 1 Jika sudah ada,
tidak ada perubahan}

procedure insert(input/output s: set, str: string)
{I.S.
```

- s adalah set yang terdefinisi dan mungkin sudah berisi beberapa elemen unik.
- str adalah string yang akan dimasukkan ke dalam set.
- s->size menyatakan jumlah elemen saat ini dalam set.}

{F.S.

Jika str belum ada di dalam s, maka:

- str disalin ke indeks ke-s->size dalam s->elements,
- s->size bertambah 1.

Jika str sudah ada di dalam s, maka:

- Tidak ada perubahan pada s.}

### 3. ADT Queue

```
constant MAX_USERNAME : integer = 50

type Node : <
    idPasien: integer;
    { ID pasien dalam antrian }
    usernamePasien: string;
    { Username pasien dalam antrian }
    next: ^Node
    { Pointer ke node berikutnya dalam antrian}>

type Queue : <
    front: ^Node;
    { Pointer ke elemen paling depan dari antrian }
    rear: ^Node;
```

```

    { Pointer ke elemen paling belakang dari antrian }

length: integer;

{ Banyaknya elemen dalam antrian }>

```

### Kamus Global

```

q: Queue

{ Variabel queue yang sedang digunakan }

idPasien: integer

{ ID pasien yang akan dimasukkan / dikeluarkan dari
antrian }

usernamePasien: string

{ Nama pasien yang akan dimasukkan / dikeluarkan }

```

```

procedure initQueue(input/output q : Queue}

{I.S. Queue q belum tentu kosong atau terdefinisi
benar.}

{F.S. Queue q diinisialisasi menjadi kosong (front
= NULL, rear = NULL, length = 0).}

```

```

procedure enqueue(input/output q : Queue, input
idPasien : integer , constant usernamePasien :
string)

{I.S. Queue q terdefinisi dan mungkin berisi
elemen.}

{F.S. Node baru berisi idPasien dan
usernamePasien ditambahkan ke belakang antrian.
length bertambah 1.}

```

```

function dequeue(q : Queue, idOut : integer,
usernameOut : string) -> boolean
{I.S. Queue q mungkin kosong atau tidak kosong.}
{F.S.

    • Jika antrian tidak kosong, elemen terdepan
      dihapus.idOut dan usernameOut diisi dengan
      data pasien tersebut, dan length berkurang 1
      Mengembalikan true.

    • Jika kosong, false dikembalikan dan data tidak
      diubah.}

```

```

function peekFront(q : Queue, idOut : integer,
usernameOut : string) -> boolean
{I.S. Queue q mungkin kosong atau tidak kosong.}
{F.S.

```

- Jika tidak kosong, \*idOut dan \*usernameOut
 diisi dengan data pasien di depan antrian,
 antrian tidak berubah. Mengembalikan true.
- Jika kosong, false dikembalikan.}

```

function isEmptyQueue(q : Queue) -> boolean
{I.S. Queue q terdefinisi.}
{F.S. Mengembalikan true jika queue kosong (front
== NULL), false jika tidak.}

```

```

procedure printQueue(input q : Queue)
{I.S. Queue q terdefinisi dan berisi 0 atau lebih
node.}

```

{F.S. Menampilkan seluruh usernamePasien dalam antrian dari depan ke belakang. Jika kosong, tampilkan pesan bahwa antrian kosong.}

#### 4. ADT Stack

```
type Stack : <  
    data : array [0..MAX_STACK-1] of integer  
    { Menyimpan nilai yang dimasukkan ke stack };  
    top : integer  
    { indeks elemen teratas, -1 jika kosong} >  
  
procedure initStack(s : Stack)  
{ I.S. Stack s mungkin berisi data tidak valid atau acak }  
{ F.S. Stack s menjadi kosong (s.top ← -1) }  
  
function isStackEmpty(s : Stack) → boolean  
{ I.S. Stack s terdefinisi }  
{ F.S. Mengembalikan true jika s.top = -1 (stack kosong), false jika tidak }  
  
function isStackFull(s : Stack) → boolean  
{ I.S. Stack s terdefinisi }  
{ F.S. Mengembalikan true jika s.top = MAX_STACK - 1 (stack penuh), false jika tidak }  
  
procedure push(s : Stack, value : integer)
```

```

{ I.S. Stack s terdefinisi dan belum penuh
(isStackFull(s) = false) }

{ F.S. Menambahkan value ke elemen paling atas
stack (s.data[s.top+1]), dan s.top bertambah 1 }

function pop(s : Stack) → integer

{ I.S. Stack s terdefinisi dan tidak kosong
(isStackEmpty(s) = false) }

{ F.S. Menghapus dan mengembalikan elemen paling
atas stack, s.top berkurang 1 }

{ Jika kosong, mengembalikan -1 }

function peek(s : Stack) → integer

{ I.S. Stack s terdefinisi dan tidak kosong
(isStackEmpty(s) = false) }

{ F.S. Mengembalikan nilai elemen paling atas stack
tanpa mengubah nilai s.top }

{ Jika kosong, mengembalikan -1 }

```

## 5. ADT Map

```

type ObatUntukPenyakit : <
    urutan    : integer
    { Menyatakan urutan konsumsi obat terhadap suatu penyakit }

    id_obat   : integer
    { ID obat yang harus dikonsumsi dalam urutan tersebut }>

```

## 6. ADT Matrix

```

ruanganList: array [0..panjang_denah-1] of array
[0..lebar_denah-1] of Ruangan

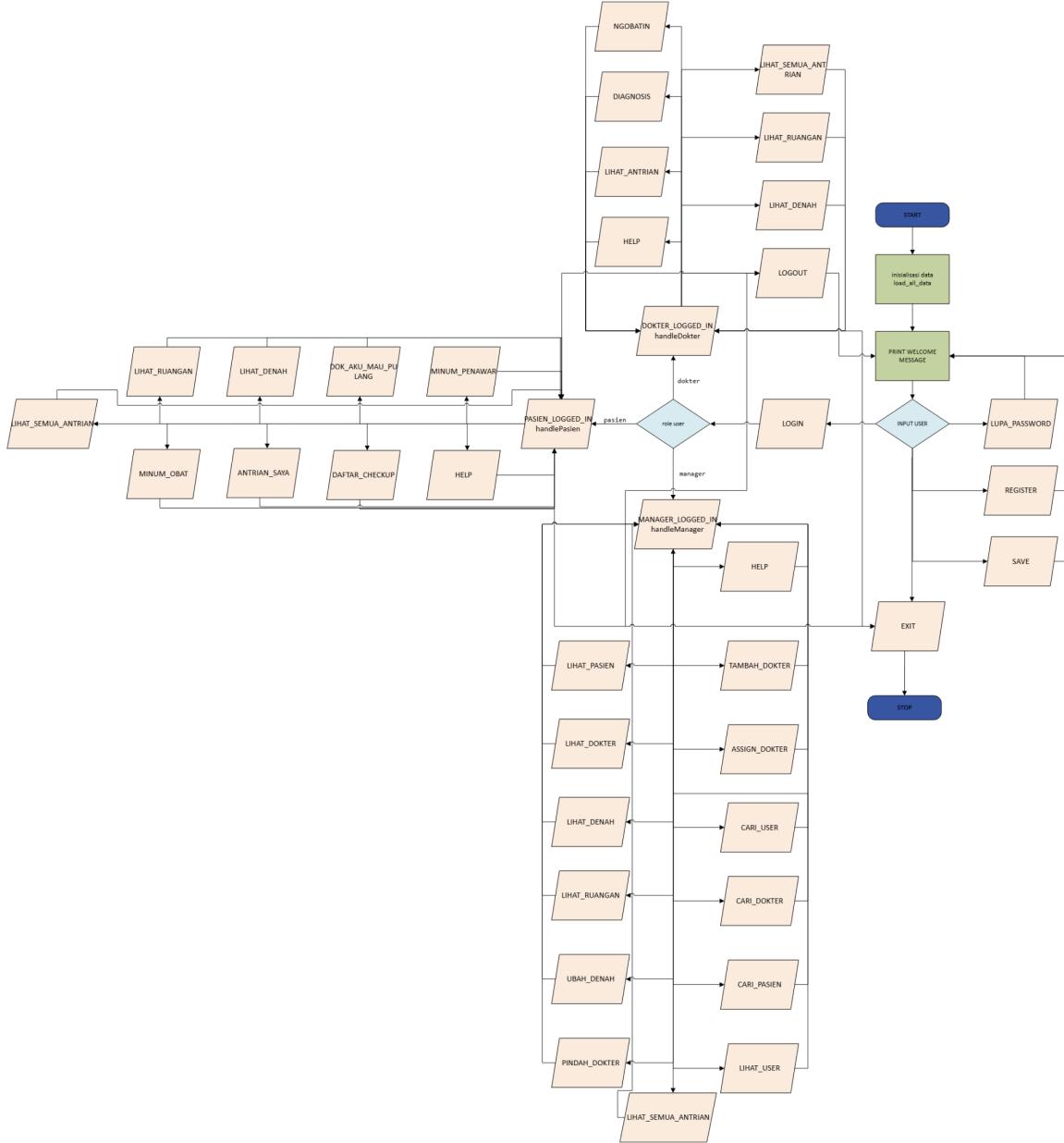
```



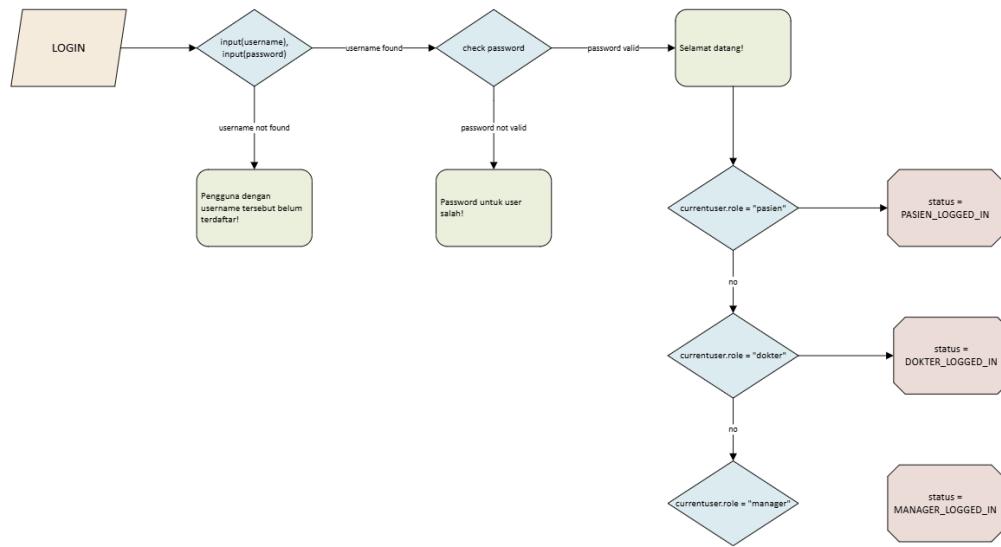
# BAB VII

## DESAIN DEKOMPOSISSI ALGORITMIK DAN FUNGSIONAL PROGRAM

### 1. Main Program

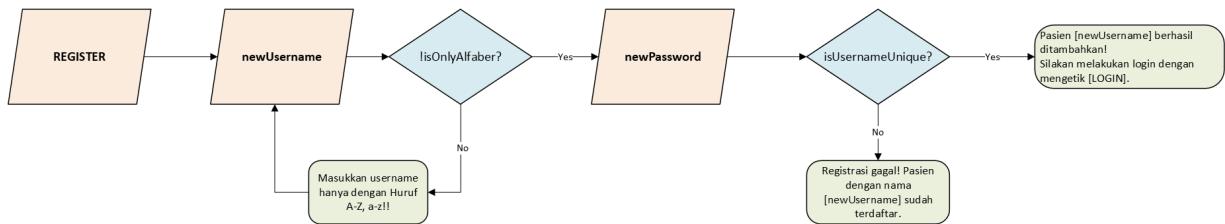


### 2. F01 - Login



### 3. F02 - Register

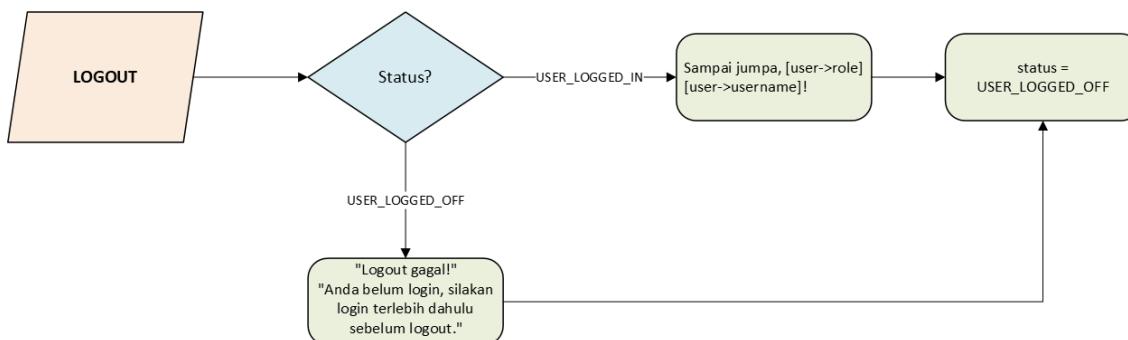
**REGISTER** Pengguna yang belum login dapat melakukan registrasi menjadi seorang pasien.



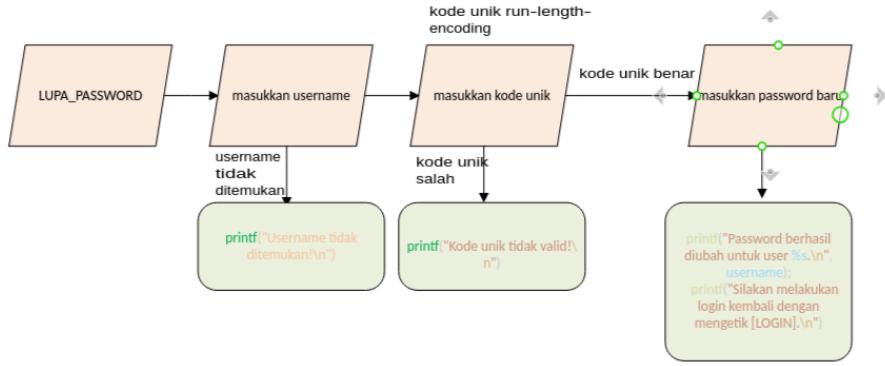
### 4. F03 - Logout

**Logout**

Pengguna melakukan logout untuk mengubah status kondisinya.

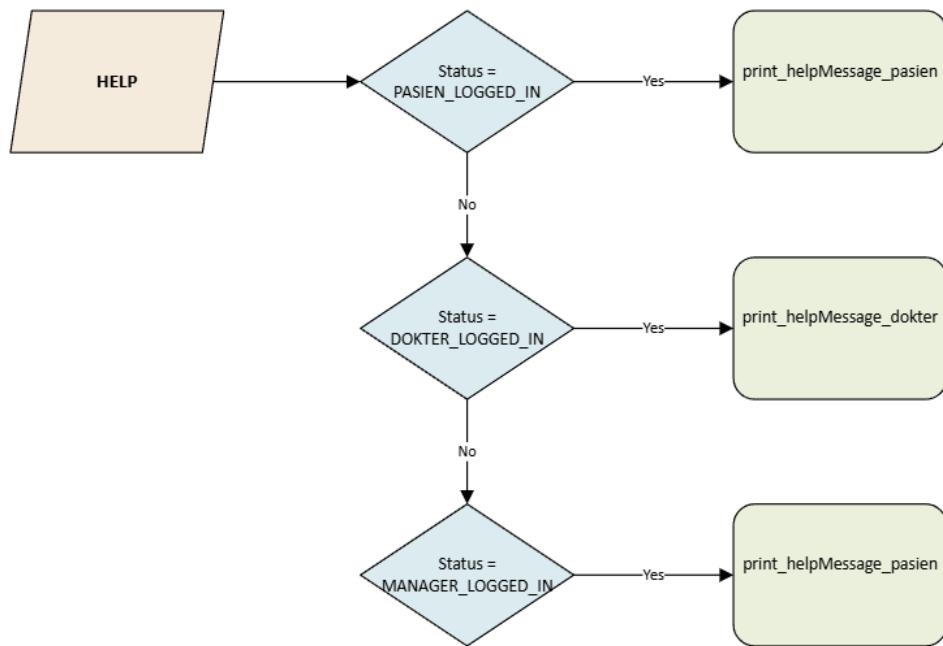


### 5. F04 - Lupa Password



## 6. F05 - Help

### Help



## 7. F06/D01 - Denah Rumah Sakit

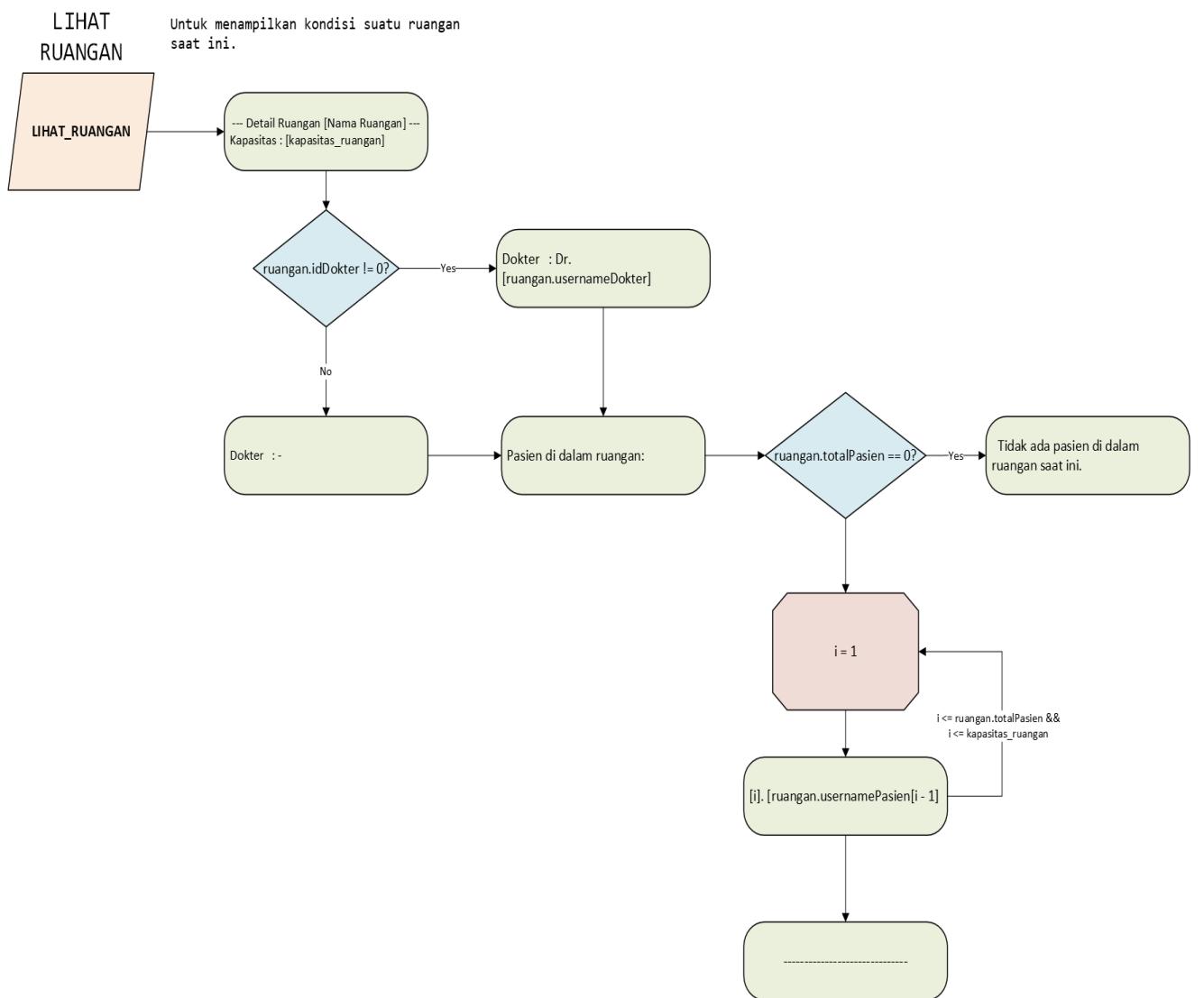
### 7.1. Lihat Denah

**LIHAT DENAH**

Untuk menampilkan gambar denah dengan ukuran saat ini.



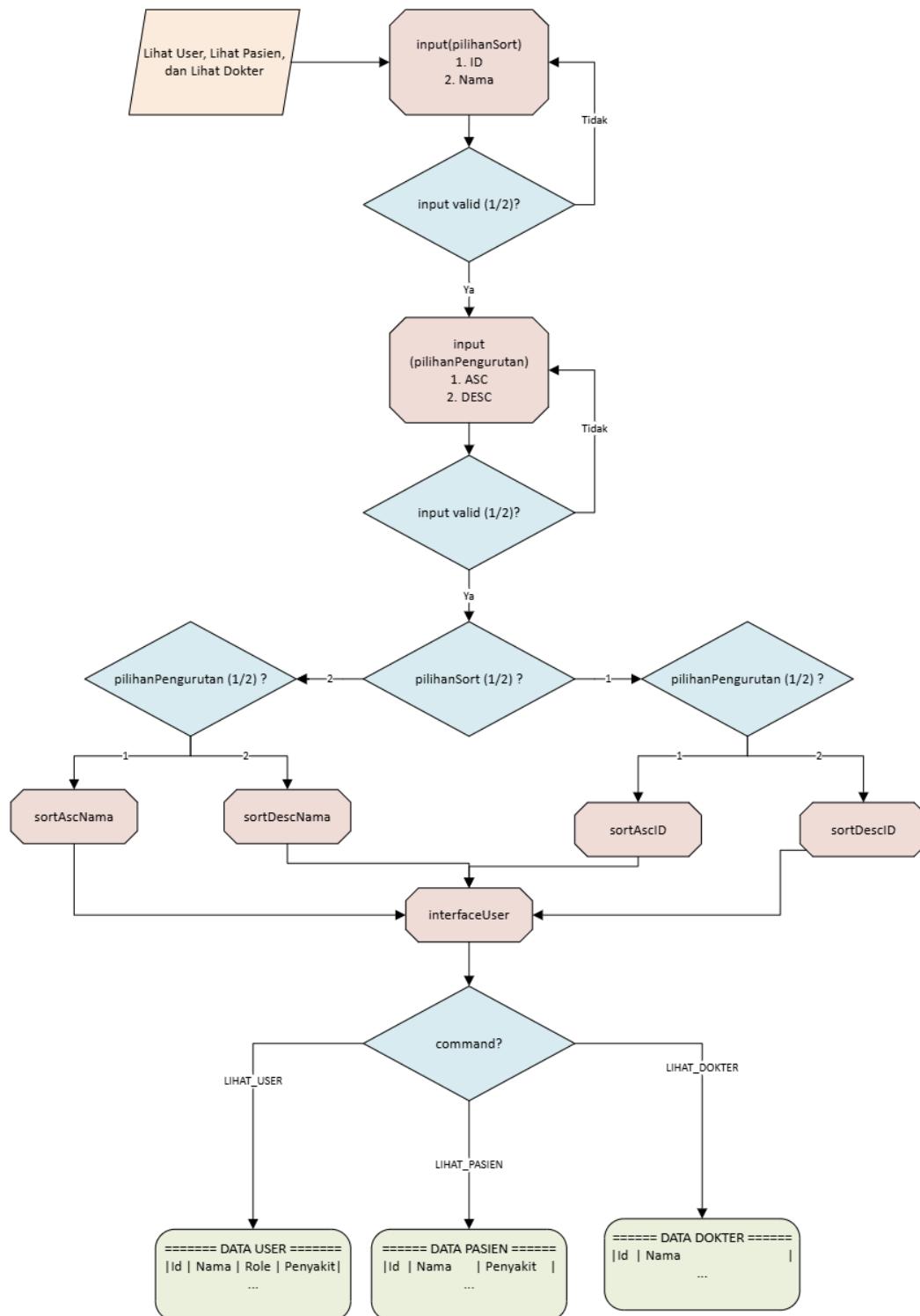
### 7.2. Lihat Ruangan



#### 8. F07 - Lihat User

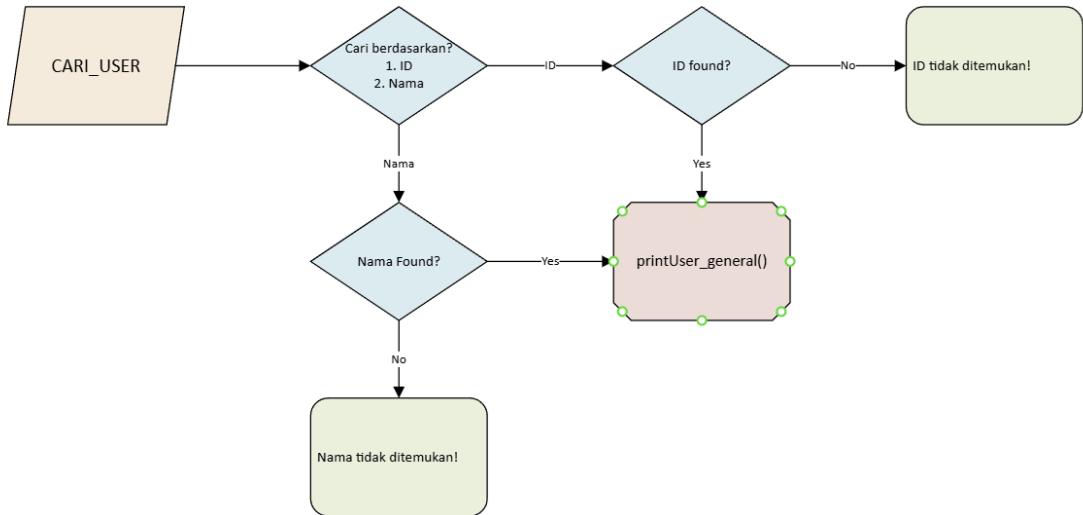
### Lihat User, Lihat Pasien, Lihat Dokter

Untuk menampilkan Data User (termasuk dokter dan pasien) sesuai pilihan sorting yang diinginkan (ID/Nama dan Ascending/Descending)

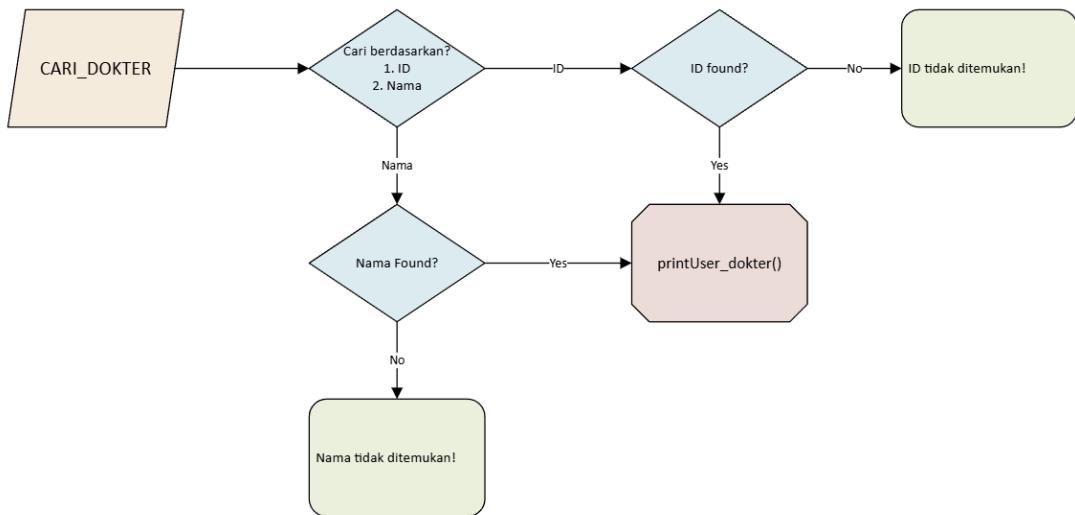


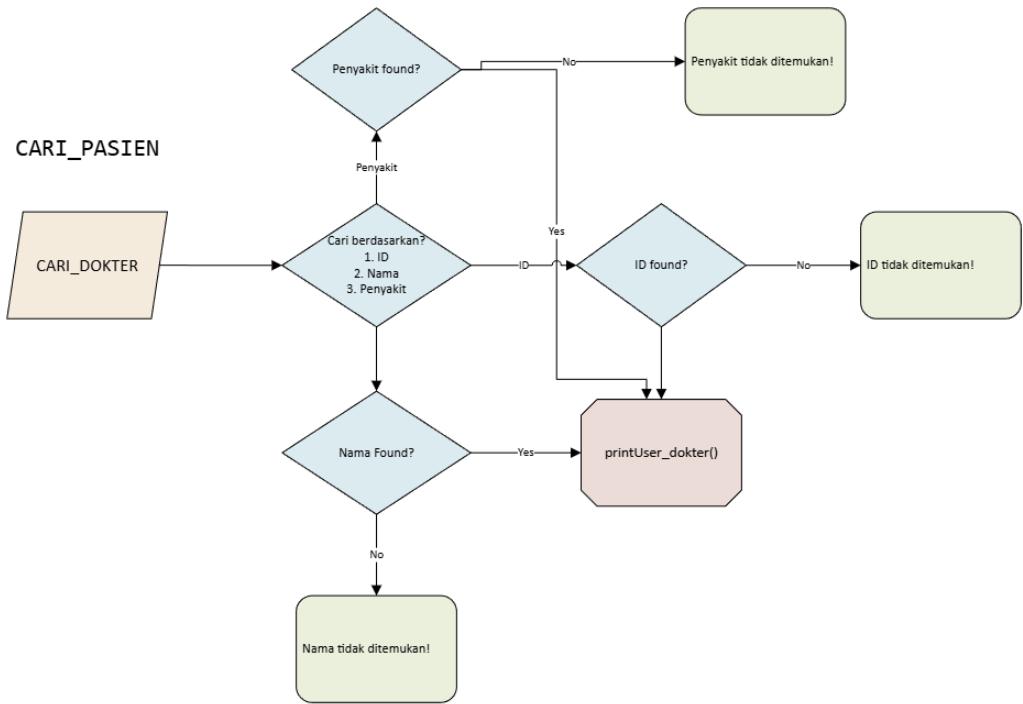
## 9. F08 - Cari User

CART\_USER

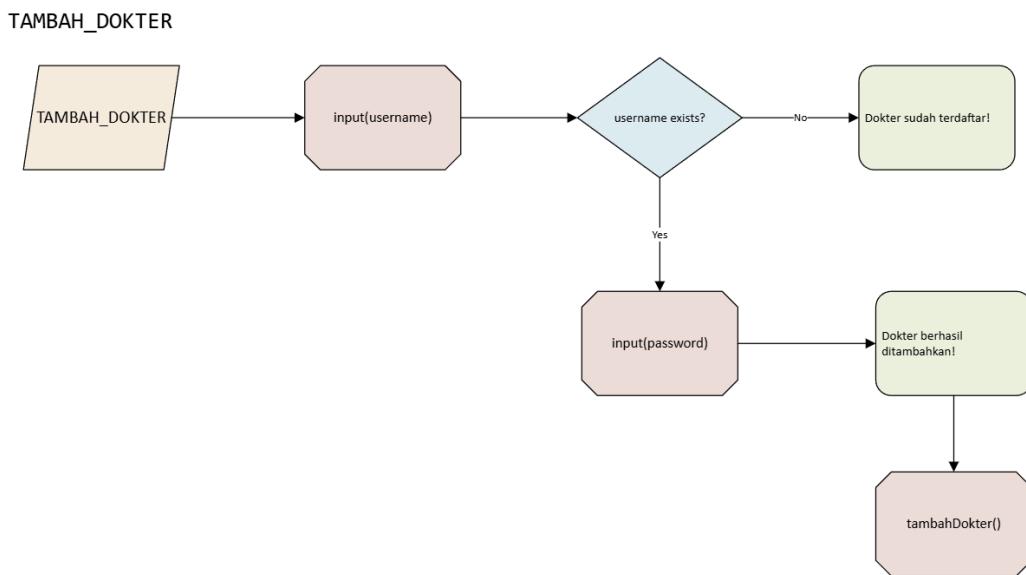


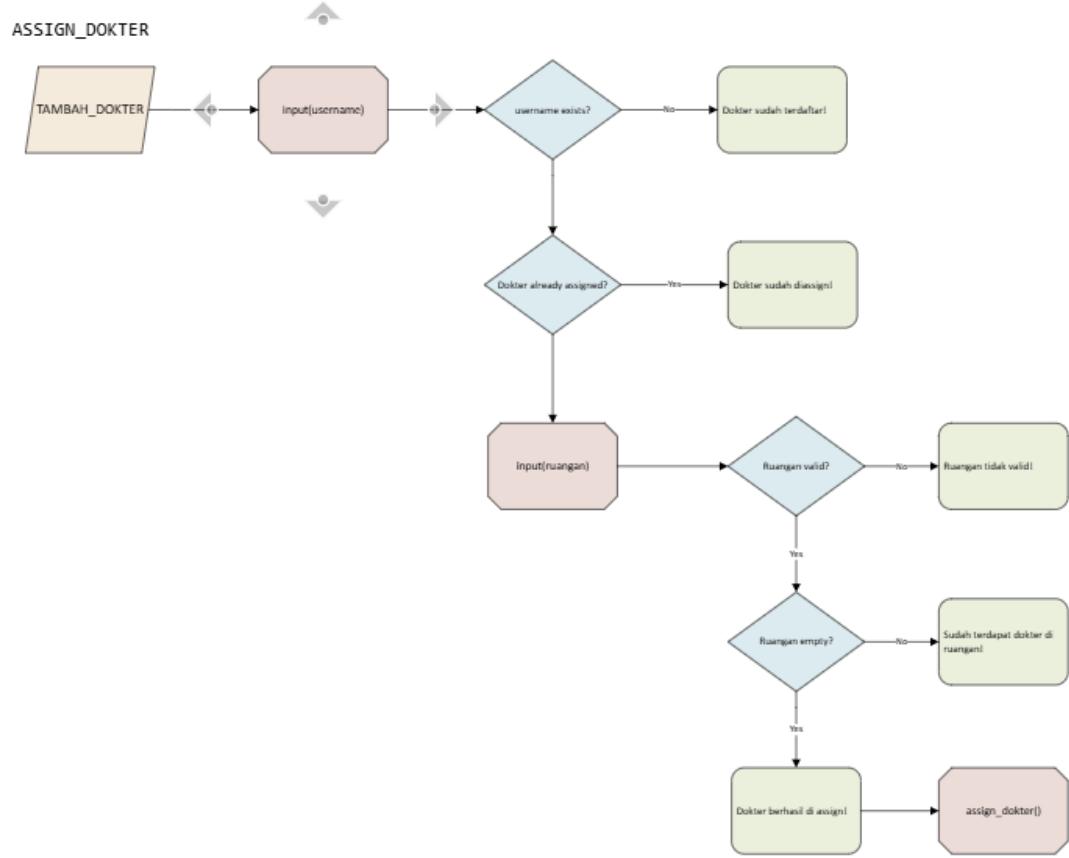
CARI\_DOKTER





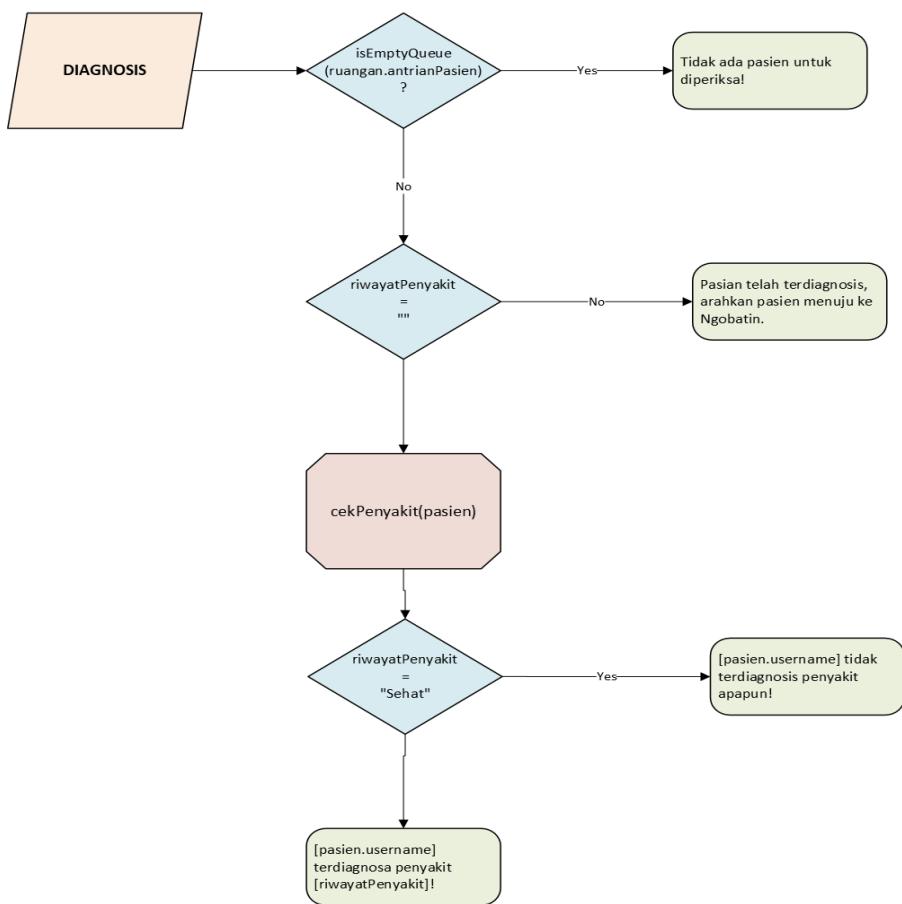
## 10. F10 - Tambah Dokter



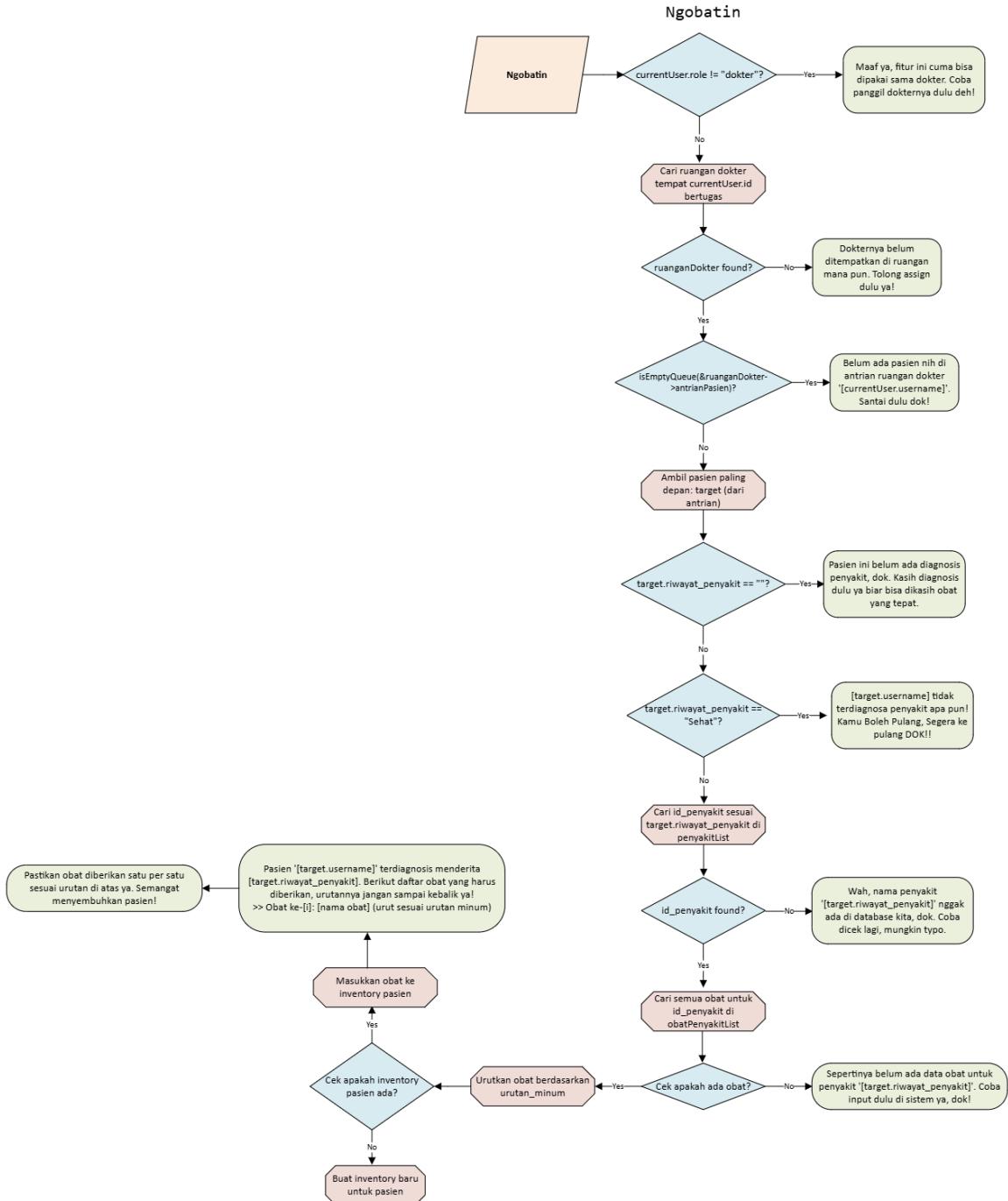


### 11. F11 - Diagnosis

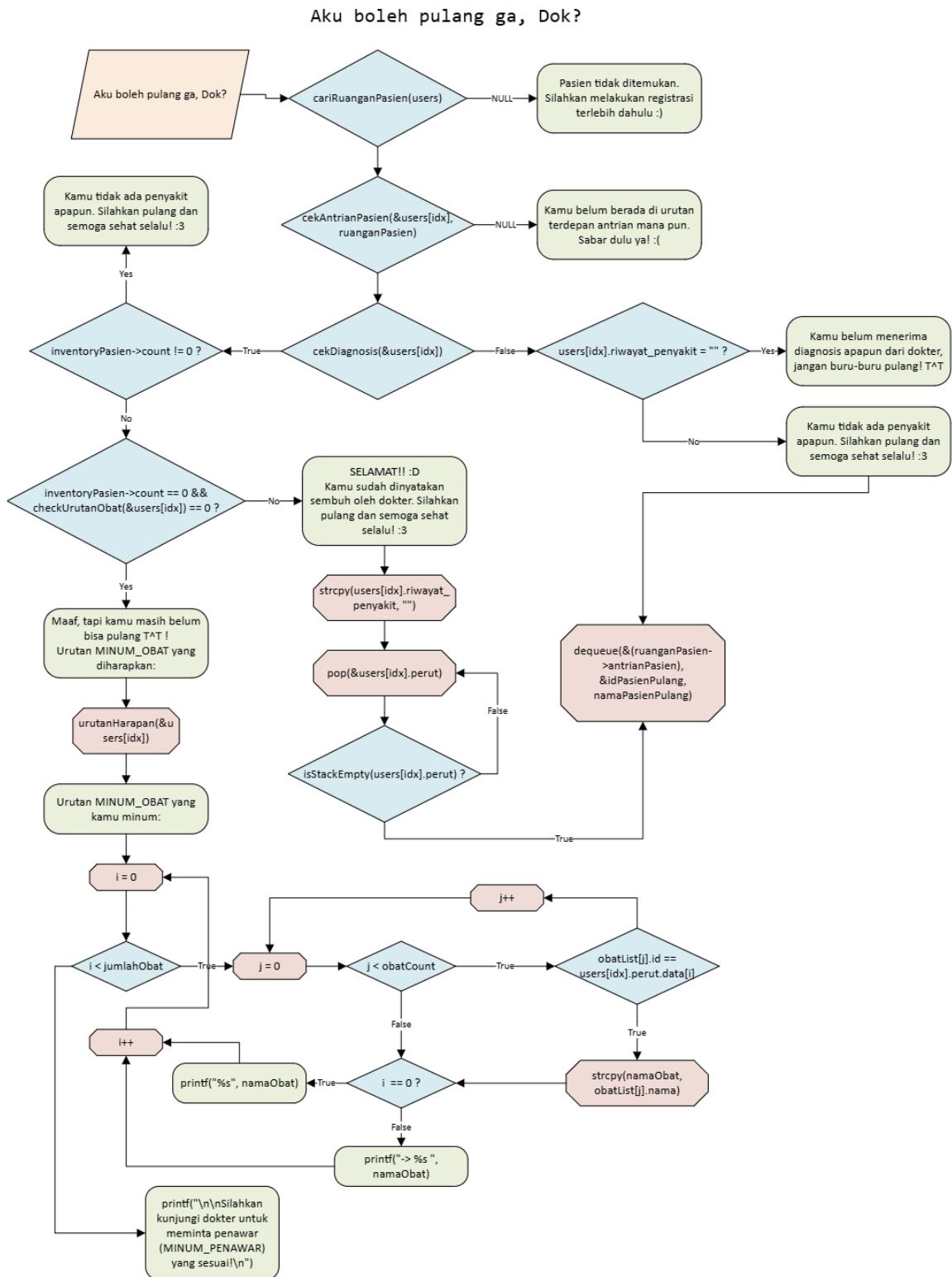
**Diagnosis** dokter dapat mendiagnosis penyakit pasien



## 12. F12 - Ngobatin

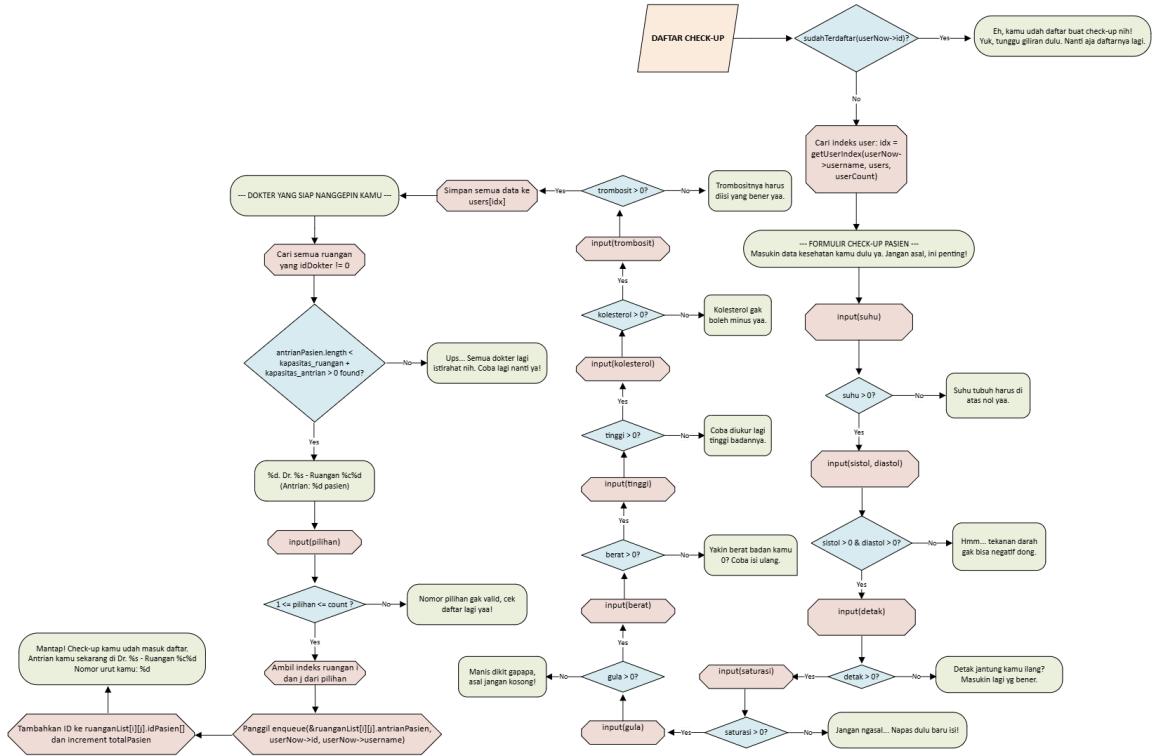


### 13. F13 - Aku boleh pulang, dok?



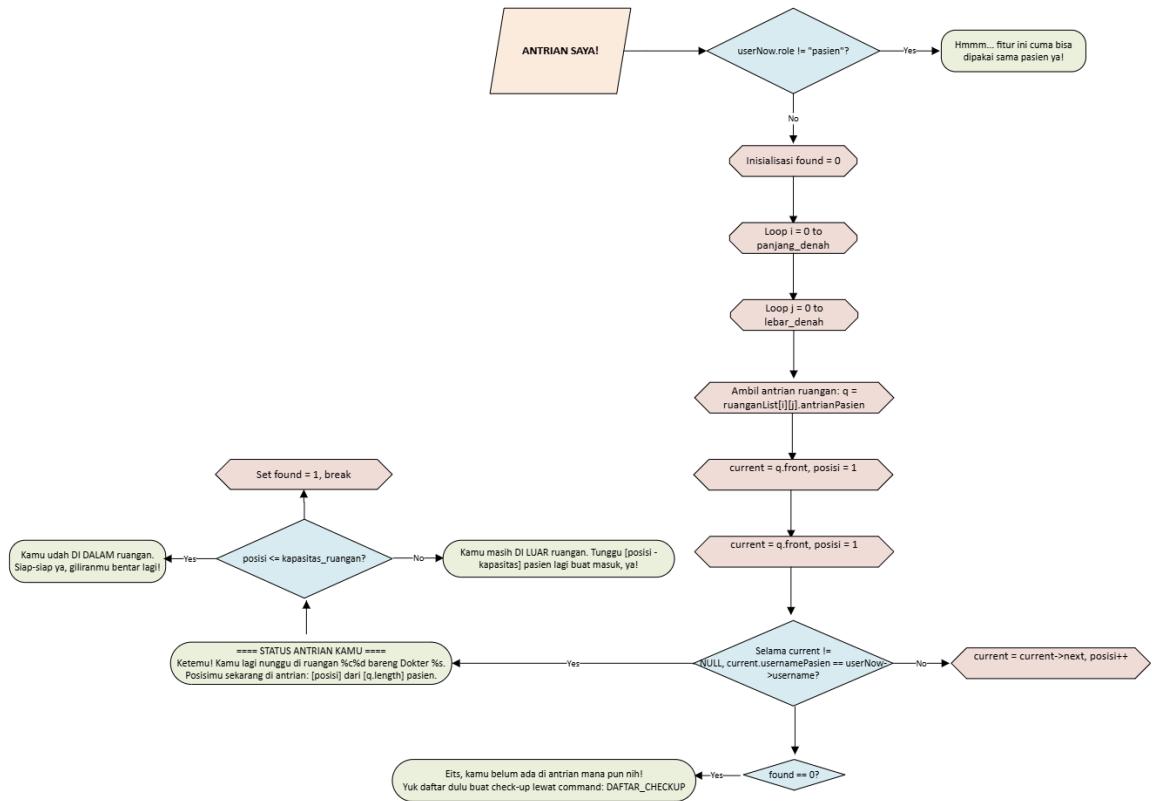
#### 14. F14 - Daftar Check-Up

### Daftar Check-Up

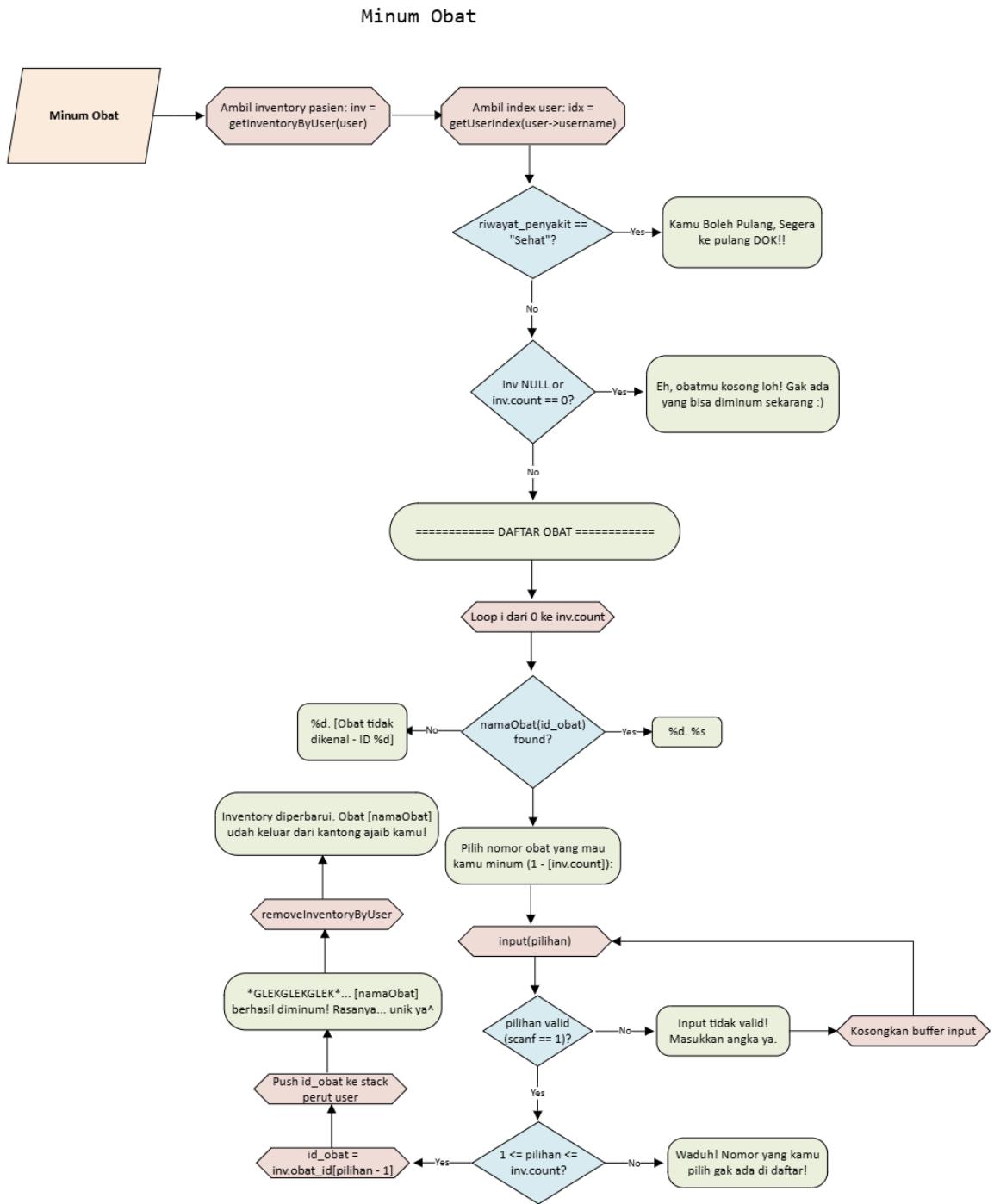


## 15. F15 - Antrian Saya

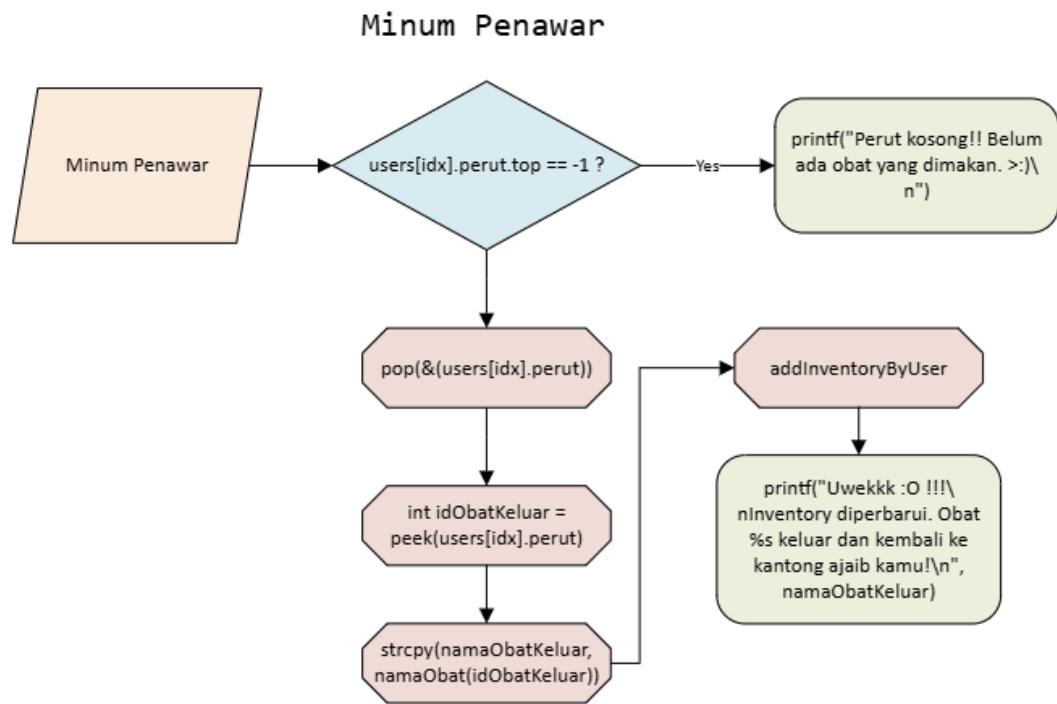
### Antrian Saya!



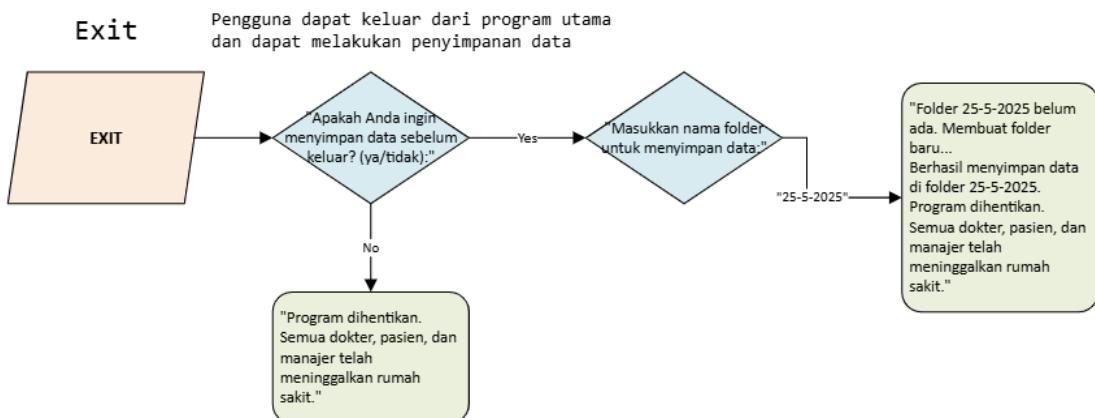
### 16. F16 - Minum Obat



17. F17 - Minum Penawar

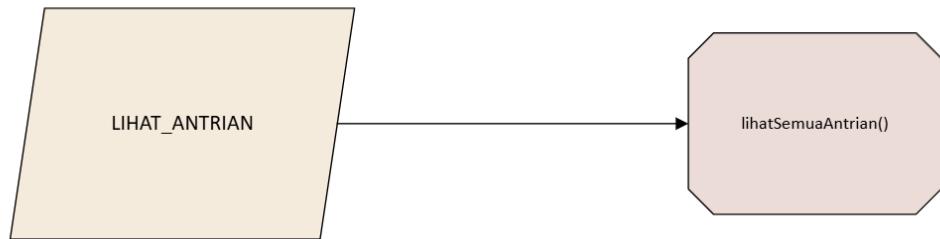


## 18. F18 - EXIT

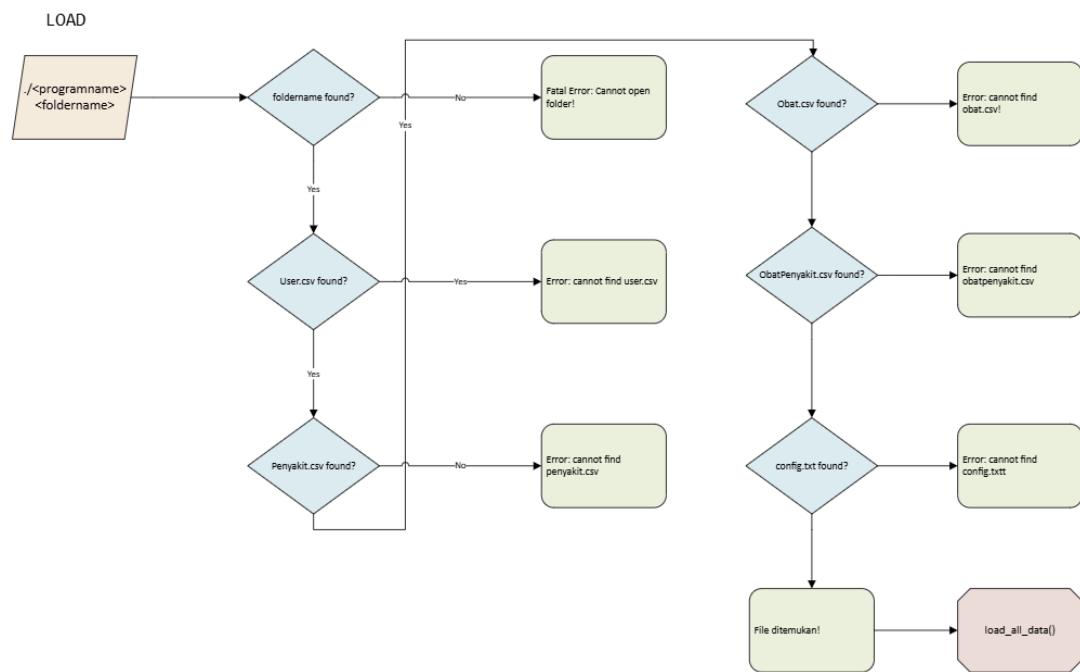


## 19. F09 / D02 - Lihat Antrian

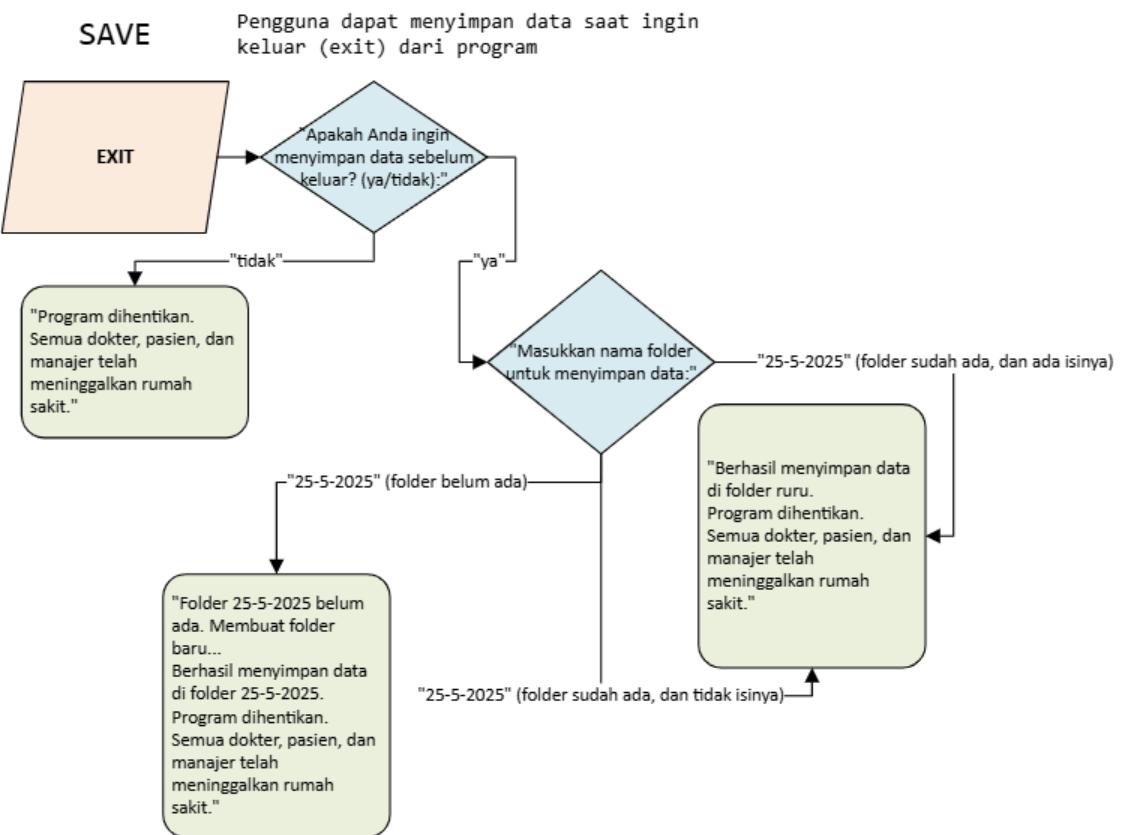
## LIHAT\_ANTRIAN



## 20. D03 - Load

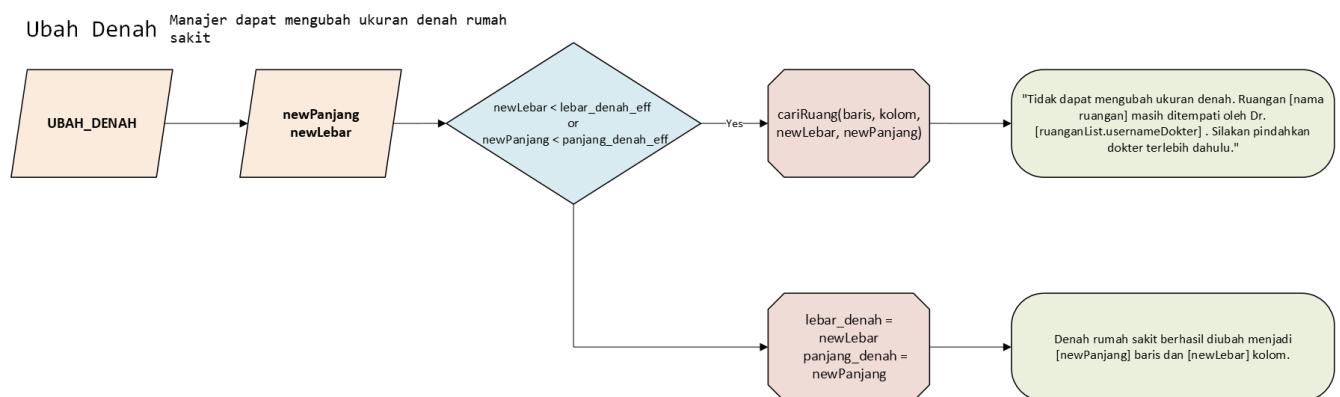


## 21. D04 - SAVE

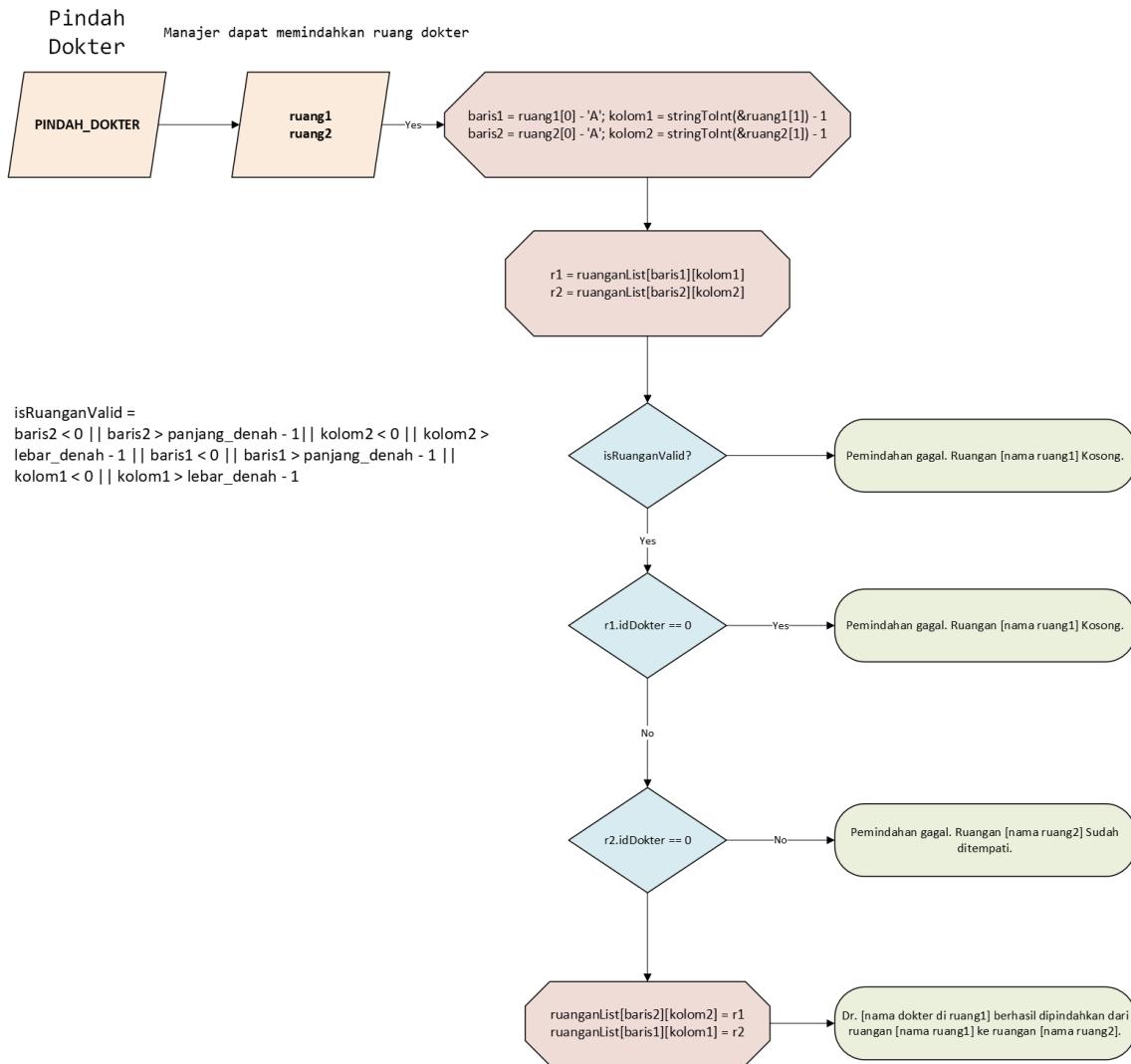


## 22. B02 - Denah Dinamis

### 22.1. Ubah Denah



## 22.2. Pindah Dokter



## BAB VIII

### SPESIFIKASI UNTUK TIAP MODUL/PROSEDUR/FUNGSI YANG DIBUAT (DALAM NOTASI ALGORITMIK)

#### 1. F01 - Login

```
function login (output user_now: User) -> User
{ I.S.: user_now adalah variabel User yang akan diisi
dengan data pengguna yang berhasil login atau data guest
jika gagal. Pengguna diminta memasukkan username dan
password. Array global users dan userCount terdefinisi
dan berisi data pengguna yang terdaftar. }
{ F.S.: user_now berisi data pengguna yang sesuai dengan
username dan password yang dimasukkan jika login
berhasil. Jika username tidak ditemukan atau password
salah, user_now.id akan diatur ke 9999 (ID guest) dan
pesan kesalahan yang sesuai ditampilkan ke layar. Pesan
keberhasilan login juga ditampilkan jika login berhasil.
Fungsi mengembalikan salinan dari user_now. }
```

#### **KAMUS LOKAL**

```
username: string[100]
password: string[100]
index: integer
```

#### **ALGORITMA**

```
user_now.id <- 9999 { ID default = guest (jika gagal
login) }
```

```

output ("Username: ")

str_toLower(username) { opsional: biar case-insensitive
}
output ("Password: ")

index <- getUserIndex(username, users, userCount)
if (index != -1) then
    if (users[index].password = password) then
        user_now <- users[index]
        output ("Login Berhasil! Selamat datang " +
        user_now.role + " " + user_now.username + ".")
    else
        output ("Password salah untuk user " +
        users[index].username + ".")
        output ("Lupa password? Ketik [LUPA_PASSWORD]
untuk mengganti password.")
else
    output ("Pengguna dengan username '" + username + "'"
belum terdaftar.")
    output ("Silakan lakukan REGISTER dengan mengetik
[REGISTER]")
    output ("atau melakukan LOGIN dengan akun yang sudah
terdaftar.")

-> user_now { Mengembalikan nilai user_now }

```

## 2. F02 - Register

function searchMaxID() -> integer

```

{I.S.: Terdefinisi array users[0..userCount-1].}
{F.S.: Mengembalikan id terbesar dari semua user yang ada.}

function isUsernameUnique(username : string, users : set, user_count : integer) -> integer
{I.S.: Terdefinisi username, userset berisi daftar username, dan jumlah user user_count.}
{F.S.: Mengembalikan 0 jika username sudah ada di userset, -1 jika unik.}

function onlyAlfabet(input str : string) -> boolean
{I.S.: Terdefinisi string username.}
{F.S.: Mengembalikan true jika username hanya terdiri dari huruf A-Z dan a-z, false jika mengandung karakter lain.}

procedure registerUser()
{I.S.: Terdefinisi array users[0..userCount-1], dan jumlah userCount.}
{F.S.:


- Jika username tidak valid (mengandung selain huruf), maka diminta ulang.
- Jika username sudah ada, tampilkan pesan gagal.
- Jika valid dan unik, tambahkan user baru dengan role "pasien", id unik, dan password yang dimasukkan, lalu naikkan userCount.}


}

```

#### **KAMUS LOKAL**

setUsername : Set

```

newUsername      : string
newPassword      : string
lowerUsername    : string
i                : integer
maxID           : integer

ALGORITMA

createEmptySet(setUsername)

i traversal [0..userCount - 1]
    insert(setUsername, users[i].username)

output("Username: ")
input(newUsername)
while (not(onlyAlfabet(newUsername))) do
    output("Masukkan username hanya dengan huruf
A-Z, a-z!!")
    output("Username: ")
    input(newUsername)

output("Password: ")
input(newPassword)

usernameToLowerCase(newUsername, lowerUsername)

if (not(isUsernameUnique(lowerUsername,
setUsername, userCount)) then
    output("Registrasi gagal! Pasien dengan nama
", newUsername, " sudah terdaftar.")
else

```

```

        users [userCount] .username <- newUsername
        users [userCount] .password <- newPassword
        users [userCount] .role <- "pasien"

        maxID <- searchMaxID ()
        users [userCount] .id <- maxID + 1
        userCount <- userCount + 1
        output ("Pasien ", newUsername, " berhasil
        ditambahkan!")
        output ("Silakan melakukan login dengan
        mengetik [LOGIN].")
    
```

**procedure** usernameToLowerCase(input username : string,
output lowerUsername)

{I.S.: Terdefinisi username sebagai string input.

F.S.: lowerUsername berisi salinan username dalam huruf kecil.}

### 3. F03 - Logout

```

procedure logout(input user : User, input/output status
: Status)

{I.S. Terdefinisi user dan status. status bernilai
USER_LOGGED_IN atau USER_LOGGED_OFF.}

{F.S.:


- Jika status == USER_LOGGED_OFF, tampilkan pesan gagal logout.
- Jika status == USER_LOGGED_IN, maka:


}
    
```

```
Tampilkan pesan perpisahan dengan user.  
Status diubah menjadi USER_LOGGED_OFF.  
Seluruh informasi dalam user di-reset (kosong).}
```

#### **KAMUS LOKAL**

-

#### **ALGORITMA**

```
if (status = USER_LOGGED_OFF) then  
    output("Logout gagal!")  
    output("Anda belum login, silakan login  
terlebih dahulu sebelum logout.")  
else  
    output("Sampai jumpa, ", user.role, " ",  
user.username, "!")  
    status ← USER_LOGGED_OFF  
    user.id ← -1  
    user.username ← ""  
    user.password ← ""  
    user.role ← ""
```

#### 4. F04 - Lupa Password

```
function generateUniqueCode(input username: string) ->  
string  
{ I.S.: username adalah string valid yang telah  
terdefinisi. }  
{ F.S.: Mengembalikan string kode unik hasil kompresi  
run-length dari username. }
```

#### **KAMUS LOKAL**

```
len : integer  
code : string
```

```
    index, count, i : integer
```

**ALGORITMA**

```
    len ← length(username)
    code ← alokasi memori sepanjang (2 * len + 1)
    index ← 0
    count ← 1
    for i ← 1 to len do
        if (i < len) and (username[i] = username[i-1]) then
            count ← count + 1
        else
            if (count > 1) then
                index ← index + sprintf(code + index,

```

```
"%d", count)
```

```
                code[index] ← username[i-1]
```

```
                index ← index + 1
```

```
                count ← 1
```

```
                code[index] ← '\0'
```

```
    return code
```

```
function validasiCode(input username: string, input
inputCode: string) -> boolean
```

```
{ I.S.: username dan inputCode adalah string valid yang
telah terdefinisi. }
```

```
{ F.S.: Mengembalikan true jika inputCode sesuai dengan
hasil generateUniqueCode(username), false jika tidak. }
```

**KAMUS LOKAL**

```
    trueCode : string
```

```
    isValid : boolean
```

**ALGORITMA**

```
trueCode ← generateUniqueCode(username)
isValid ← (trueCode = inputCode)
free(trueCode)
return isValid
```

```
procedure lupaPassword()
{ I.S.: Belum ada input username, kode unik, dan
password baru. Data user tersimpan dalam array users
sebanyak userCount. }
{ F.S.: Jika username dan kode unik valid, maka password
user yang sesuai diperbarui. Jika tidak valid, proses
dibatalkan dengan pesan kesalahan. }
```

#### **KAMUS LOKAL**

```
username : string[MAX_USERNAME]
kodeUnik : string[200]
newPassword : string[MAX_PASSWORD]
userIndex : integer
```

#### **ALGORITMA**

```
output ("==== Lupa Password ===")
output ("Username: ")
input (username)

userIndex ← -1
for i ← 0 to userCount-1 do
    if (users[i].username = username) then
        userIndex ← i
    break
```

```

if (userIndex = -1) then
    output ("Username tidak ditemukan!")
    return

output ("Kode Unik: ")
input (kodeUnik)

if not validasiCode(username, kodeUnik) then
    output ("Kode unik tidak valid!")
    return

output ("Password Baru: ")
input (newPassword)

users [userIndex] .password ← newPassword
output ("Password berhasil diubah untuk user ",
username, ".")
output ("Silakan melakukan login kembali dengan
mengetik [LOGIN].")

```

## 5. F05 - Help

```

procedure print_helpMessage_pasien (input usernow: User)
{ I.S.: usernow adalah record User yang valid dengan
role "pasien" dan username terdefinisi. }
{ F.S.: Pesan bantuan spesifik untuk peran pasien,
termasuk nama username pasien, ditampilkan ke layar. }

```

Pesan berisi daftar perintah yang tersedia untuk pasien.  
}

**procedure** print\_helpMessage\_dokter (input usernow: User)  
{ I.S.: usernow adalah record User yang valid dengan  
role "dokter" dan username terdefinisi. }  
{ F.S.: Pesan bantuan spesifik untuk peran dokter,  
termasuk nama username dokter, ditampilkan ke layar.  
Pesan berisi daftar perintah yang tersedia untuk dokter.  
}

**procedure** print\_helpMessage\_manager (input usernow:  
User)  
{ I.S.: usernow adalah record User yang valid dengan  
role "manager" dan username terdefinisi. }  
{ F.S.: Pesan bantuan spesifik untuk peran manager,  
termasuk nama username manager, ditampilkan ke layar.  
Pesan berisi daftar perintah yang tersedia untuk  
manager. }

**procedure** help (input userNow: User)  
{ I.S.: userNow adalah record User yang valid dengan  
atribut role dan username terdefinisi. }  
{ F.S.: Memanggil prosedur print\_helpMessage yang sesuai  
berdasarkan role dari userNow (manager, dokter, atau  
pasien). Jika role tidak valid, pesan "Role tidak  
valid." ditampilkan ke layar. }

## **ALGORITMA**

```

if (userNow.role = "manager") then
    print_helpMessage_manager(userNow)
else if (userNow.role = "dokter") then
    print_helpMessage_dokter(userNow)
else if (userNow.role = "pasien") then
    print_helpMessage_pasien(userNow)
else
    output("Role tidak valid.")

```

## 6. F06/D01 - Denah Rumah Sakit

**procedure** gambarDenah()  
 {I.S. Nilai panjang\_denah dan lebar\_denah sudah ditentukan.}

{F.S. Denah rumah sakit ditampilkan dalam bentuk grid baris-kolom dengan label posisi ruangan (misal A1, B2, ...).}

### KAMUS LOKAL

i, j, k: integer

### ALGORITMA

```

output("      ")
k traversal [1..lebar_denah]
    output(" ", k, " ")

i traversal [1..panjang_denah]
    output("      ")
    k traversal [1..lebar_denah]
        output("-----")
    output("+")
    output(" ", chr(ord('A') + i - 1), " ")
        j traversal [1..lebar_denah]
            output("| ", chr(ord('A') + i - 1), j, " ")
        output("| ")

```

```

        output("      ")
        k traversal [1..lebar_denah]
            output("-----")
            output("+")
    
```

**procedure** lihatRuang(input str : string)  
{I.S.Terdefinisi str sebagai nama ruangan valid (misal:  
"A1") dan data ruangan (ruanganList) telah dimuat.}  
{F.S. Ditampilkan informasi lengkap mengenai ruangan  
tersebut: kapasitas, nama dokter, serta daftar pasien di  
dalam ruangan.}

#### KAMUS LOKAL

```

str: string
x, y: integer
i: integer

```

#### ALGORITMA

```

x ← ord(str[0]) - ord('A')
y ← stringToInt(substring(str, 1)) - 1
i <- 1
output("--- Detail Ruangan " + str + " ---")
output("Kapasitas : ", kapasitas_ruangan)

if (ruanganList[x][y].idDokter ≠ 0) then
    output("Dokter : Dr. " +
    ruanganList[x][y].usernameDokter)
else
    output("Dokter : -")

output("Pasien di dalam ruangan:")
if (ruanganList[x][y].totalPasien = 0) then
    output(" Tidak ada pasien di dalam ruangan
saat ini.")
else
    while (i < ruanganList[x][y].totalPasien and i
    < kapasitas_ruangan) do
        output(" , i, ". ,
        ruanganList[x][y].usernamePasien[i - 1])
output("-----")

```

7. F07 - Lihat user

```
procedure urutanSort(input users : User, userCount :  
integer)  
{I.S : users sudah terdefinisi dan userCount sudah  
terdefinisi sebagai variabel global}  
{F.S : Prosedur untuk melakukan pemilihan pengurutan (ID  
/ Nama dan Ascending / Descending}  
KAMUS LOKAL  
pilihanSort : integer  
pilihanPengurutan : integer  
ALGORITMA  
output ("Urutkan berdasarkan? (1/2)\n")  
output ("    1. ID\n")  
output ("    2. Nama\n")  
iterate  
    output (">>>")  
    input (pilihanSort)  
stop(pilihanSort = 1 or pilihanSort = 2)  
    output ("Input tidak Valid. Silahkan ketik ulang\n")  
output ("Urutkan Sort? (1/2)\n")  
output ("    1. ASC(A-Z) \n")  
output ("    2. DESC(Z-A) \n")  
iterate  
    output (">>>")  
    input (pilihanPengurutan)  
stop(pilihanPengurutan = 1 or pilihanPengurutan = 2)  
    output ("Input tidak Valid. Silahkan ketik ulang\n")
```

```

depend on : pilihanSort, pilihanPengurutan

    pilihanSort = 1 and pilihanPengurutan = 1 :
        sortAscID(users, userCount)

    pilihanSort = 2 and pilihanPengurutan = 1 :
        sortAscNama(users, userCount)

    pilihanSort = 1 and pilihanPengurutan = 2 :
        sortDescID(users, userCount)

    pilihanSort = 2 and pilihanPengurutan = 2 :
        sortDescNama(users, userCount)

procedure sortDescID(input users : User, userCount :
integer)

{I.S : users sudah terdefinisi dan userCount sudah
terdefinisi sebagai variabel global}

{F.S : Prosedur untuk melakukan pengurutan descending
berdasarkan ID}

procedure sortAscID(input users : User, userCount :
integer)

{I.S : users sudah terdefinisi dan userCount sudah
terdefinisi sebagai variabel global}

{F.S : Prosedur untuk melakukan pengurutan ascending
berdasarkan ID}

procedure sortDescNama(input users : User, userCount :
integer)

{I.S : users sudah terdefinisi dan userCount sudah
terdefinisi sebagai variabel global}

```

{F.S : Prosedur untuk melakukan pengurutan descending berdasarkan nama}

```
procedure sortAscNama(input users : User, userCount : integer)
```

```
{I.S : users sudah terdefinisi dan userCount sudah  
terdefinisi sebagai variabel global}
```

{F.S : Prosedur untuk melakukan pengurutan ascending berdasarkan nama }

**procedure** interfaceUser(input users: User, input command

: char, input useCount : integer)

{I.S : users dan userCount sudah terdefinisi. userCount sudah terdefinisi sebagai variabel global}

{F.S : Prosedur untuk menampilkan user interface dan menampilkan data yang ingin dilihat sesuai dengan command (LIHAT USER / LIHAT PASIEN / LIHAT DOKTER)}

## KAMUS LOKAL

1

## ALGORITMA

depend on : command

```
command = "LIHAT USER" :
```

output ("===== DATA

USER =====\n")

-----+-----+ \n")

```
output ("| ID | Nama | Role | Penyakit |")
```

```

output ("+-----+-----+
-----+-----+\n")
i traversal[0..userCount - 1]
output ("|", users[i].id, "|", users[i].
username, "|", users[i].role,"|",
users[i].riwayat_penyakit, "|")

command = "LIHAT_PASIEN" :
output ("===== DATA
PASIEN =====\n")
output ("+-----+-----+
-----+-----+\n")
output ("| ID | Nama | Penyakit|")

output ("+-----+-----+
-----+-----+\n")
i traversal[0..userCount - 1]
output ("|", users[i].id, "|", users[i].
username, "|", users[i].riwayat_penyakit,
"|")

command = "LIHAT_DOKTER" :
output ("===== DATA
DOKTER =====\n")
output ("+-----+-----+
-----+-----+\n")
output ("| ID | Nama | ")

output ("+-----+-----+
-----+-----+\n")

```

```
i traversal[0..userCount - 1]
    output("|", users[i].id, "|", users[i].
    username, " | ")
```

**procedure** lihatUser(input command : char)  
{I.S : users sudah terdefinisi dan userCount sudah  
terdefinisi sebagai variabel global}  
{F.S : Prosedur untuk menampilkan Data User (termasuk  
pasien dan dokter) sesuai dengan pilihan sorting yang  
dipilih}

**KAMUS LOKAL**

-  
**ALGORITMA**

```
urutanSort(users, userCount)
interfaceUser(users, command, userCount)
```

**procedure** lihatPasien(input command : char)  
{I.S : users sudah terdefinisi dan userCount sudah  
terdefinisi sebagai variabel global}  
{F.S : Prosedur untuk menampilkan Data Pasien sesuai  
dengan pilihan sorting yang dipilih}

**KAMUS LOKAL**

-  
**ALGORITMA**

```
lihatUser(command)
```

**procedure** lihatDokter(input command : char)  
{I.S : users sudah terdefinisi dan userCount sudah  
terdefinisi sebagai variabel global}

{F.S : Prosedur untuk menampilkan Data Dokter sesuai dengan pilihan sorting yang dipilih}

**KAMUS LOKAL**

-

**ALGORITMA**

lihatUser(command)

8. F08 - Cari User

```
procedure cariPasien ()  
{ I.S.: Array global users dan userCount terdefinisi dan  
berisi data pengguna. Pengguna diminta memilih kriteria  
pencarian (ID, Nama, atau Penyakit) dan memasukkan nilai  
yang dicari. }  
{ F.S.: Hasil pencarian pasien ditampilkan ke layar.  
Jika pasien ditemukan, detail pasien ditampilkan. Jika  
tidak ditemukan, pesan yang sesuai ditampilkan. Jika  
pilihan tidak valid, pesan kesalahan ditampilkan. Memori  
sementara yang dialokasikan (jika ada) akan dibebaskan.  
}
```

**KAMUS LOKAL**

userChoice: integer

c: character { Untuk membersihkan buffer input }

usersTemp: pointer to array of User { Array sementara  
untuk sorting jika mencari berdasarkan ID }

i: integer

targetID: integer

index: integer

targetUsername: string[128]

```

found: boolean
penyakit: string[32]
countPenyakit: integer

ALGORITMA
output("Cari pasien berdasarkan:")
output("1. ID")
output("2. Nama")
output("3. Penyakit")
output("Masukkan angka pilihan: ")
output(">>> ") { Menampilkan prompt tanpa baris baru }
input(userChoice)
read_remaining_line(c) { Membersihkan buffer input }

usersTemp <- NULL
if (userChoice = 1) then
if (userCount > 0) then
    usersTemp <- allocate_memory_array_user(userCount)
    if (usersTemp = NULL) then
        output("Gagal mengalokasikan memori untuk
usersTemp")
    i traversal [0..userCount-1]
    usersTemp[i] <- users[i]

if (userChoice = 1) then { Pencarian berdasarkan ID
(Binary Search) }
output("Cari ID: ")
input(targetID)
read_remaining_line(c) { Membersihkan buffer }

```

```

if (userCount > 0) then

    qsort(usersTemp, userCount, size_of_user, compareID)
    index <- binarySearchID(usersTemp, userCount,
targetID)

else

    index <- -1 { Tidak ada user untuk dicari }

if (index != -1) and (usersTemp[index].role = "pasien")
then

    output("Pasien dengan ID " +
integer_to_string(targetID) + " ditemukan:")
    printUser_general(usersTemp[index])

else

    output("Pasien dengan ID '" +
integer_to_string(targetID) + "' tidak ditemukan
atau bukan pasien.")

else if (userChoice = 2) then { Pencarian berdasarkan
Username (Sequential Search) }

    output("Cari Username: ")
    input(targetUsername, 128) { Membaca baris termasuk
spasi }

    remove_newline(targetUsername) { Menghapus newline
dari input_line }

        found <- false
        output("Mencari pasien dengan username '" +
targetUsername + "':")
        i traversal [0..userCount-1]

```

```

    if (users[i].username = targetUsername) and
(users[i].role = "pasien") then

        printUser_general(users[i])
        found <- true

{ break opsional jika hanya ingin satu hasil }

if (not found) then

    output("Pasien dengan username '" +
targetUsername + "' tidak ditemukan.")

else if (userChoice = 3) then { Pencarian berdasarkan
Riwayat Penyakit (Sequential Search) }

    output("Cari penyakit: ")
    input_line(penyakit, 32)
    remove_newline(penyakit)

countPenyakit <- 0
i traversal [0..userCount-1]

    if (users[i].riwayat_penyakit = penyakit) and
(users[i].role = "pasien") then

        if (countPenyakit = 0) then

            output( "-----+-----+-----+
-----+")
            output( " | ID | Nama | "
Penyakit | ")

output( "-----+-----+-----+
-----+")

printPasien(users[i]) { Menggunakan printPasien
untuk format tabel }

```

```

output ("-----+-----+
+")
countPenyakit <- countPenyakit + 1

if (countPenyakit = 0) then
    output ("Pasien dengan penyakit '" + penyakit + "'"
tidak ditemukan.")
else
    output("Pilihan tidak valid. Mohon masukkan pilihan yang
benar!.")  
  

if (usersTemp != NULL) then free_memory(usersTemp)

procedure cariDokter ()
{ I.S.: Array global users dan userCount terdefinisi dan
berisi data pengguna. Pengguna diminta memilih kriteria
pencarian (ID atau Nama) dan memasukkan nilai yang
dicari. }
{ F.S.: Hasil pencarian dokter ditampilkan ke layar.
Jika dokter ditemukan, detail dokter ditampilkan. Jika
tidak ditemukan, pesan yang sesuai ditampilkan. Jika
pilihan tidak valid, pesan kesalahan ditampilkan. Memori
sementara yang dialokasikan (jika ada) akan dibebaskan.
}

```

#### **KAMUS LOKAL**

userChoice: integer  
c: character { Untuk membersihkan buffer input }

```
usersTemp: pointer to array of User { Array sementara  
untuk sorting jika mencari berdasarkan ID }  
i: integer  
targetID: integer  
index: integer  
targetUsername: string[128]  
found: boolean
```

#### ALGORITMA

```
output("Cari dokter berdasarkan:")  
output("1. ID")  
output("2. Nama")  
output("Masukkan angka pilihan: ")  
output(">>> ")  
input(userChoice)  
read_remaining_line(c)  
  
usersTemp <- NULL  
if (userChoice = 1) then  
    if (userCount > 0) then  
        usersTemp <-  
        allocate_memory_array_user(userCount)  
        if (usersTemp = NULL) then  
            output("Gagal mengalokasikan memori untuk  
usersTemp")  
i traversal [0..userCount-1]  
usersTemp[i] <- users[i]
```

```

if (userChoice = 1) then { Pencarian berdasarkan ID
(Binary Search) }

    output("Cari ID: ")
    input(targetID)
    read_remaining_line(c)

if (userCount > 0) then

    qsort(usersTemp, userCount, size_of_user,
compareID)

    index <- binarySearchID(usersTemp, userCount,
targetID)

else

    index <- -1

if (index != -1) and (usersTemp[index].role = "dokter")
then

    output("Dokter dengan ID " +
integer_to_string(targetID) + " ditemukan:")
    printDokter(usersTemp[index]) { atau printUser_general
}

else

    output("Dokter dengan ID '" +
integer_to_string(targetID) + "' tidak ditemukan atau
bukan dokter.")

else if (userChoice = 2) then { Pencarian berdasarkan
Username (Sequential Search) }

    output("Cari Username: ")
    input(targetUsername, 128)
    remove_newline(targetUsername)

```

```

found <- false

output("Mencari dokter dengan username '" +
targetUsername + "' :")
i traversal [0..userCount-1]
if (users[i].username = targetUsername) and
(users[i].role = "dokter") then
    printDokter(users[i]) { atau printUser_general }
    found <- true
    { break opsional }

if (not found) then
    output("Dokter dengan username '" +
targetUsername + "' tidak ditemukan.")

else
    output("Pilihan tidak valid. Mohon masukkan
pilihan yang benar! .")

if (usersTemp != NULL) then
    free(usersTemp)

procedure cariUser ()
{ I.S.: Array global users dan userCount terdefinisi dan
berisi data pengguna. Pengguna diminta memilih kriteria
pencarian (ID atau Nama) melalui prosedur printPilihan
dan memasukkan nilai yang dicari. }
{ F.S.: Hasil pencarian pengguna (apapun rolenya)
ditampilkan ke layar. Jika pengguna ditemukan, detail
pengguna ditampilkan. Jika tidak ditemukan, pesan yang

```

sesuai ditampilkan. Jika pilihan tidak valid, pesan kesalahan ditampilkan. Memori sementara yang dialokasikan (jika ada) akan dibebaskan. }

#### **KAMUS LOKAL**

userChoice: integer  
c: character { Untuk membersihkan buffer input }  
usersTemp: pointer to array of User { Array sementara untuk sorting jika mencari berdasarkan ID }  
i: integer  
targetID: integer  
index: integer  
targetUsername: string[128]  
found: boolean

#### **ALGORITMA**

```
printPilihan() { Mencetak opsi: 1. ID, 2. Nama }
(userChoice)
read_remaining_line(c)

usersTemp <- NULL
if (userChoice = 1) then
    if (userCount > 0) then
        usersTemp <-
allocate_memory_array_user(userCount)
        if (usersTemp = NULL) then
            output("Gagal mengalokasikan memori untuk
usersTemp")
i traversal [0..userCount-1]
```

```

usersTemp[i] <- users[i]

if (userChoice = 1) then { Pencarian berdasarkan ID
(Binary Search) }

output("Cari ID: ")
input(targetID)
read_remaining_line(c)

if (userCount > 0) then
    qsort(usersTemp, userCount, size_of_user,
compareID)
    index <- binarySearchID(usersTemp, userCount,
targetID)

else
    index <- -1
    if (index != -1) then
        output("Pengguna dengan ID " +
integer_to_string(targetID) + " ditemukan:")
        printUser_general(usersTemp[index])
    else
        output("User dengan ID '" +
integer_to_string(targetID) + "' tidak
ditemukan.")

else if (userChoice = 2) then { Pencarian berdasarkan
Username (Sequential Search) }

output("Cari Username: ")
input(targetUsername, 128)
remove_newline(targetUsername)
found <- false

```

```

output("Mencari user dengan username '" +
targetUsername + "' :")
    i traversal [0..userCount-1]
        if (users[i].username = targetUsername) then
            output("Menampilkan " + users[i].role + " "
dengan nama " + users[i].username + ":")
            printUser_general(users[i])
            found <- true
        { break opsional }

        if (not found) then
            output("User dengan username '" +
targetUsername + "' tidak ditemukan.")
        else
            output("Pilihan tidak valid. Mohon masukkan pilihan
yang benar!..")

        if (usersTemp != NULL) then
            free(usersTemp)

```

## 9. F10 - Tambah Dokter

```

function isValidUsername (input username: string) ->
integer
{ I.S.: username adalah string yang akan divalidasi. }
{ F.S.: Mengembalikan 1 jika username hanya terdiri dari
huruf (A-Z, a-z). Mengembalikan 0 jika mengandung
karakter selain huruf atau kosong. }

```

```

function getNextUserId () -> integer

```

```
{ I.S.: Array global users dan userCount terdefinisi.  
userCount menyatakan jumlah pengguna saat ini. }  
{ F.S.: Mengembalikan ID unik berikutnya yang dapat  
digunakan untuk pengguna baru, yaitu ID maksimum yang  
ada saat ini ditambah satu. Jika tidak ada pengguna,  
mengembalikan 1. }
```

```
procedure tambahDokter ()  
{ I.S.: Array global users dan userCount terdefinisi.  
usernameSet global belum terinisialisasi atau berisi  
data lama. Pengguna mungkin memasukkan username dan  
password. }  
{ F.S.: Jika kapasitas pengguna belum penuh, username  
valid dan belum digunakan, maka seorang pengguna baru  
dengan peran "dokter" ditambahkan ke array users.  
userCount bertambah satu. Username baru dimasukkan ke  
usernameSet. Pesan keberhasilan atau kegagalan  
ditampilkan ke layar. Jika kapasitas penuh, username  
tidak valid, atau username sudah ada, dokter tidak  
ditambahkan dan pesan error yang sesuai ditampilkan. }
```

KAMUS LOKAL

```
i: integer  
username, temp, lowerUsername : string [MAX_USERNAME]  
password: string [MAX_PASSWORD]  
newDokter: User
```

ALGORITMA

```
createEmptySet (usernameSet)  
i traversal [0..userCount-1]
```

```

        toLowerCase(users[i].username, temp)
        insert(usernameSet, temp)

if (userCount >= MAX_USERS) then
    output("Kapasitas pengguna penuh!")

output("Masukkan username dokter baru: ")
input(username)

if (not isValidUsername(username)) then
    output("Username hanya boleh terdiri dari huruf!")

toLowerCase(username, lowerUsername)

if (contains(usernameSet, lowerUsername)) then
    output("Username sudah digunakan. Silakan pilih
username lain.")

output("Masukkan password dokter baru: ")
input(password)

newDokter.id <- getNextUserId()
newDokter.username <- username
newDokter.password <- password
newDokter.role <- "dokter"
newDokter.riwayat_penyakit <- ""
newDokter.suhu_tubuh <- 0.0
newDokter.tekanan_darah_sistolik <- 0

```

```

newDokter.tekanan_darah_diastolik <- 0
newDokter.detak_jantung <- 0
newDokter.saturasi_oksigen <- 0
newDokter.kadar_gula_darah <- 0
newDokter.berat_badan <- 0.0
newDokter.tinggi_badan <- 0.0
newDokter.kadar_kolesterol <- 0
newDokter.trombosit <- 0

users[userCount] <- newDokter
userCount <- userCount + 1
insert(usernameSet, username)

output("Dokter " + username + " berhasil ditambahkan
dengan ID " + integer_to_string(newDokter.id) + ".")  
  

procedure assignDokter ()
{
  I.S.: Array global users, userCount, ruanganList,
  panjang_denah, dan lebar_denah terdefinisi. Pengguna
  diminta memasukkan username dokter dan kode ruangan. }
{ F.S.: Jika dokter dengan username yang diberikan ada,
  belum diassign ke ruangan lain, dan kode ruangan valid
  serta belum terisi, maka dokter tersebut diassign ke
  ruangan yang ditentukan dalam ruanganList. ID dokter dan
  username dokter pada ruangan tersebut diperbarui,
  totalPasien direset ke 0, dan antrianPasien
  diinisialisasi. Pesan keberhasilan atau kegagalan
  (dokter tidak ditemukan, sudah diassign, format ruangan
}

```

```
salah, baris/kolom tidak valid, ruangan sudah terisi)  
ditampilkan ke layar. }
```

#### KAMUS LOKAL

```
username: string[MAX_USERNAME]  
dokterId: integer  
i, j: integer  
input_ruangan: string[10]  
rowChar: character  
row, col: integer
```

#### ALGORITMA

```
output("Masukkan username dokter yang akan diassign: ")  
input(username)
```

```
dokterId <- -1  
i traversal [0..userCount-1]  
    if (users[i].username = username) and  
        (users[i].role = "dokter") then  
            dokterId <- users[i].id  
            break { Keluar dari loop setelah dokter ditemukan }
```

```
if (dokterId = -1) then  
    output("Dokter dengan username " + username + "  
tidak ditemukan.")
```

```
{ Cek apakah sudah diassign ke suatu ruangan }  
i traversal [0..panjang_denah-1]  
    j traversal [0..lebar_denah-1]
```

```

    if (ruanganList[i] [j].idDokter = dokterId)
then

        output ("Dokter sudah diassign ke ruangan
        " + character('A' + i) + integer_to_string(j +
        1) + ".")  

output ("Masukkan kode ruangan (misal A1, B3, dst): ")
input (input_ruangan)  

if (length(input_ruangan) < 2) then
    output ("Format ruangan tidak valid.")  

rowChar <- input_ruangan[0] { Mengambil karakter pertama
}
if (rowChar < 'A') or (rowChar >= character('A' +
panjang_denah)) then
    output ("Baris tidak valid. Maksimum baris adalah "
+ character('A' + panjang_denah - 1) + ".")  

row <- integer(rowChar) - integer('A')
col <- string_to_integer(substring(input_ruangan, 1,
length(input_ruangan)-1)) - 1 { Mengambil sisa string
dan konversi ke integer }  

if (col < 0) or (col >= lebar_denah) then
    output ("Kolom tidak valid. Harus antara 1 hingga "
+ integer_to_string(lebar_denah) + ".")  

if (ruanganList[row] [col].idDokter != 0) then

```

```

output("Ruangan " + character('A' + row) +
integer_to_string(col + 1) + " sudah diisi oleh
dokter " + ruanganList[row][col].usernameDokter +
".")

ruanganList[row][col].idDokter <- dokterId
ruanganList[row][col].usernameDokter <- username
ruanganList[row][col].totalPasien <- 0
initQueue(ruanganList[row][col].antrianPasien)

output("Dokter " + username + " berhasil diassign ke
ruangan " + character('A' + row) + integer_to_string(col
+ 1) + ".")
```

**procedure** tampilanDokterDanRuangan () { I.S.: Array  
global users, userCount, ruanganList, panjang\_denah, dan  
lebar\_denah terdefinisi. } { F.S.: Menampilkan daftar  
semua pengguna dengan peran "dokter" beserta ID,  
username, dan ruangan tempat mereka diassign (jika ada).  
Jika tidak ada dokter yang terdaftar, pesan yang sesuai  
akan ditampilkan. Format tampilan adalah tabel. }

## 10. F11 - Diagnosis

```

procedure cekPenyakit (input/output user : User)
{I.S.: Terdefinisi pointer pasien dengan data-data  

medis, dan daftar penyakit (penyakitList).}
{F.S.: Field riwayat_penyakit pasien diisi dengan nama  

penyakit yang cocok dengan data medis pasien, atau  

"Sehat" jika tidak ada yang cocok.}
```

```

procedure searchRuangan(input user : User, input/output
ruangan : Ruangan)
{I.S.: Terdefinisi user (dokter) dan ruanganList berisi
daftar ruangan.}
{F.S.: Variabel *ruangan berisi ruangan tempat dokter
dengan id == user.id, jika ada.}

function cekSuhu(user : User, penyakit : Penyakit) ->
boolean
{I.S.: Terdefinisi data suhu tubuh user dan rentang suhu
penyakit.}
{F.S.: Menghasilkan nilai true jika suhu tubuh user
berada dalam rentang suhu penyakit.}

function cekTekananSistolik(user : User, penyait : Penyakit) -> boolean
{I.S.: Terdefinisi tekanan sistolik user dan batas nilai
pada penyakit.}
{F.S.: Menghasilkan true jika tekanan sistolik user
sesuai batas penyakit.}

function cekTekananDiastolik(user : User, penyakit : Penyakit) -> boolean
{I.S.: Terdefinisi tekanan diastolik user dan batas pada
penyakit.}
{F.S.: Menghasilkan true jika tekanan sesuai dengan
rentang penyakit.}

function cekDetakJantung(user : User, penyakit : Penyakit) -> boolean
{I.S.: Terdefinisi detak jantung user dan rentang pada
penyakit.}
{F.S.: Menghasilkan true jika detak jantung user sesuai
rentang penyakit.}

```

```
function cekSaturasiOksigen(user : User, penyakit : Penyakit) -> boolean
{I.S.: Terdefinisi saturasi oksigen user dan batasan pada penyakit.}
{F.S.: Menghasilkan true jika saturasi oksigen dalam rentang penyakit.}
```

```
function cekKadarGulaDarah(user : User, penyakit : Penyakit) -> boolean
{I.S.: Terdefinisi kadar gula darah user dan rentang penyakit.}
{F.S.: Menghasilkan true jika kadar gula darah dalam rentang penyakit.}
```

```
function cekBeratBadan(user : User, penyakit : Penyakit)
-> boolean
{I.S.: Terdefinisi berat badan user dan rentang penyakit.}
{F.S.: Menghasilkan true jika berat badan sesuai batas penyakit.}
```

```
function cekTinggiBadan(user : User, penyakit : Penyakit) -> boolean
{I.S.: Terdefinisi tinggi badan user dan batasan penyakit.}
{F.S.: Menghasilkan true jika tinggi badan berada dalam batas yang sesuai.}
```

```
function cekKadarKolesterol(user : User, penyakit : Penyakit) -> boolean
```

{I.S.: Terdefinisi kadar kolesterol user dan rentang penyakit.}

{F.S.: Menghasilkan true jika kadar kolesterol sesuai dengan rentang penyakit.}

**function** cekKadarKolesterol(user : User, penyakit : Penyakit) -> boolean

{I.S.: Terdefinisi jumlah trombosit user dan rentang pada penyakit.}

{F.S.: Menghasilkan true jika trombosit dalam batas penyakit.}

**procedure** diagnosis(input user : User)

{I.S.: Terdefinisi user sebagai dokter, ruangan telah memiliki antrian pasien.

{F.S.: Pasien terdepan pada antrian didiagnosis penyakitnya jika belum, dan hasil diagnosis ditampilkan ke layar.}

#### **KAMUS LOKAL**

user: User

ruangan: Ruangan

i: integer

pasien: pointer to User

#### **ALGORITMA**

searchRuang(user, ruangan)

if (isEmptyQueue(ruangan.antrianPasien)) then

output("Tidak ada pasien untuk diperiksa!")

else

    i ← 0

    pasien ← NIL

while (i < userCount) and (pasien = null) do

if(ruangan.antrianPasien.front.idPasien = users[i].id) then

            pasien ← &users[i]

```

else
    i ← i + 1

if (pasien.riwayat_penyakit = "") then
    cekPenyakit(pasien)
    if pasien.riwayat_penyakit = "Sehat" then
        output(pasien.username, " tidak
        terdiagnosis penyakit apapun!")
    else
        output(pasien.username, "
        terdiagnosa penyakit ",
        pasien.riwayat_penyakit, "!")
else
    output("Pasien telah terdiagnosis,
    arahkan pasien menuju ke Ngobatin.")

```

## 11. F12 - Ngobatin

```

procedure ngobatin(input currentUser : User)
{ I.S.: currentUser terdefinisi dan merupakan user
dengan role "dokter". Data user, antrian, dan relasi
penyakit-obat tersedia. }

{ F.S.: Jika dokter memiliki pasien dengan riwayat
penyakit, sistem akan mencari obat yang sesuai dan
memberikan daftar obat ke inventory pasien. Jika tidak
ditemukan obat, tidak ada perubahan. }

```

### **KAMUS LOKAL**

```

ruanganDokter: Ruangan
i, j, k: integer
idPasien: integer
usernamePasien: string
target: User
id_penyakit: integer
list_obat: array[0..49] of record
count_obat: integer
id_obat: integer

```

```

urutan: integer
foundObat: boolean
inv: InventoryPasien

ALGORITMA

if currentUser.role ≠ "dokter" then
    output("Maaf ya, fitur ini cuma bisa dipakai
    sama dokter. Coba panggil dokternya dulu deh!")
- >

ruanganDokter ← null
for i ← 0 to panjang_denah - 1 do
    for j ← 0 to lebar_denah - 1 do
        if ruanganList[i][j].idDokter =
        currentUser.id then
            ruanganDokter ← ruanganList[i][j]
            break

if ruanganDokter = null then
    output("Dokternya belum ditempatkan di
ruangan mana pun. Tolong assign dulu ya!")
- >

if isEmptyQueue(ruanganDokter.antrianPasien) then
    output("Belum ada pasien nih di antrian
ruangan dokter '", currentUser.username, "'.
Santai dulu dok!")
- >

peekFront(ruanganDokter.antrianPasien, idPasien,
usernamePasien)

target ← null
for i ← 0 to userCount - 1 do
    if users[i].id = idPasien then
        target ← users[i]
        break

if target = null then

```

```

output("Hmm... pasien di antrian nggak ketemu
di database. Ada yang janggal nih, dok.")
->

if target.riwayat_penyakit = "" then
    output("Pasien belum ada diagnosis penyakit,
dok. Kasih diagnosis dulu ya biar bisa dikasih
obat yang tepat.")
->

if target.riwayat_penyakit = "Sehat" then
    output("Kamu Boleh Pulang, Segera ke pulang
DOK!!")
->

id_penyakit <- -1
for i <- 0 to penyakitCount - 1 do
    if penyakitList[i].name_penyakit = target.riwayat_penyakit then
        id_penyakit <- penyakitList[i].id
        break

if id_penyakit = -1 then
    output("Wah,      nama      penyakit      '',
target.riwayat_penyakit, '' nggak ada di database
kita, dok. Coba dicek lagi, mungkin typo.")
->

count_obat <- 0
for i <- 0 to obatPenyakitCount - 1 do
    if obatPenyakitList[i].penyakit_id = id_penyakit then
        list_obat[count_obat].id_obat <- obatPenyakitList[i].obat_id
        list_obat[count_obat].urutan <- obatPenyakitList[i].urutan_minum
        count_obat <- count_obat + 1

```

```

if count_obat = 0 then
    output("Sepertinya belum ada data obat untuk
penyakit '", target.riwayat_penyakit, "'. Coba
input dulu di sistem ya, dok!")
->

{ bubble sort list_obat berdasarkan urutan }
for i < 0 to count_obat - 2 do
    for j < 0 to count_obat - i - 2 do
        if list_obat[j].urutan >
list_obat[j+1].urutan then
            tukar(list_obat[j], list_obat[j+1])

inv < null
for i < 0 to jumlah_inventory - 1 do
    if daftar_inventory[i].pasien_id = target.id
then
    inv < daftar_inventory[i]
    break

if inv = null then
    inv < daftar_inventory[jumlah_inventory]
    inv.pasien_id < target.id
    inv.count < 0
    jumlah_inventory < jumlah_inventory + 1

output("")
output("Pasien      '", target.username, "'"
terdiagnosis           menderita      '',
target.riwayat_penyakit, ".")
output("Berikut daftar obat yang harus diberikan,
urutannya jangan sampai kebalik ya!")
output("")

for i < 0 to count_obat - 1 do
    id_obat < list_obat[i].id_obat
    inv.obat_id[inv.count] < id_obat
    inv.count < inv.count + 1

```

```

foundObat <= false
for k < 0 to obatCount - 1 do
    if obatList[k].id = id_obat then
        output(">> Obat ke-" + (i+1) + ":" + obatList[k].nama)
        foundObat <= true
        break

    if not(foundObat) then
        output(">> Obat ke-", i+1, ":[Data obat dengan ID ", id_obat, " nggak ditemukan, ada yang jangggal nih!]")

    output("")
    output("Pastikan obat diberikan satu per satu sesuai urutan di atas ya. Semangat menyembuhkan pasien!")

```

## 12. F13 - Aku boleh pulang ga, Dok?

```

function checkDiagnosis (pasien : User) -> boolean
{I.S : pasien terdefinisi}

{F.S : Mengembalikan false jika pasien belum terdiagnosis atau pasien sudah berstatus "Sehat". Mengembalikan true jika pasien terdiagnosis sakit (memiliki riwayat_penyakit)}

function getIdFromPenyakit (pasien : User) -> integer
{I.S : pasien sudah terdefinisi, variabel penyakitCount dan penyakitList sudah terdefinisi sebagai variabel global}

{F.S : Mengembalikan id penyakit yang dialami pasien berdasarkan riwayat penyakit pasien}

```

```
function getJumlahObat (pasien : User) -> integer
{I.S : pasien terdefinisi, obatPenyakitCount dan
obatPenyakitList sudah terdefinisi sebagai variabel
global}
{F.S : Mengembalikan jumlah obat yang harus diminum oleh
pasien}
```

```
function checkUrutanObat (pasien : User) -> integer
{I.S : pasien terdefinisi, obatPenyakitCount dan
obatPenyakitList sudah terdefinisi sebagai variabel
global}
{F.S : Melakukan pengecekan apakah obat yang diminum
oleh pasien sudah sesuai urutan atau belum.
Mengembalikan 0 apabila urutan obat yang diminum belum
sesuai urutan. Mengembalikan 1 apabila obat yang diminum
sudah sesuai urutan}
```

```
procedure urutanHarapan (input pasien : User)
{I.S : pasien sudah terdefinisi, obatPenyakitCount,
ObatPenyakitList, obatCount, dan obatList sudah
terdefinisi sebagai variabel global}
{F.S : Menampilkan ke layar urutan obat dan nama obat
yang seharusnya diminum oleh pasien}
```

```
function cariRuanganPasien(pasien : User) -> Ruangan
{I.S : pasien sudah terdefinisi, panjang_denah,
lebar_denah, dan variabel bertipe Ruangan sudah
terdefinisi sebagai variabel global}
```

{F.S : Mengembalikan Ruangan dimana pasien itu berada}

**function** cariAntrian(pasien : User, ruanganPasien : Ruangan) -> Ruangan  
{I.S : pasien sudah terdefinisi, panjang\_denah, lebar\_denah, dan variabel bertipe Ruangan sudah terdefinisi sebagai variabel global}  
{F.S : Mengembalikan urutan antrian pasien pada suatu ruangan}

**procedure** akuBolehPulang (input pasien : User)  
{I.S : pasien sudah terdefinisi variabel bertipe Ruangan, obatCount, dan obatList sudah terdefinisi sebagai variabel global}  
{F.S : Melakukan pengecekan apakah pasien sudah diperbolehkan untuk pulang. Pasien diperbolehkan untuk pulang dengan syarat pasien harus ada di salah satu ruangan dan berada di antrian paling depan, pasien sudah terdiagnosis. Jika pasien "Sehat" maka diperbolehkan untuk pulang. Jika pasien terdiagnosis penyakit dan sudah minum obat sesuai dengan urutan minum, maka pasien diperbolehkan untuk pulang}

#### **KAMUS LOKAL**

idPasienPulang : integer  
namaPasienPulang : string  
idx : integer  
ruanganPasien : Ruangan  
inventoryPasien : InventoryPasien

#### **ALGORITMA**

```

ruanganPasien ← cariRuanganPasien(users[idx])
inventoryPasien ← getInventoryByUser(user[idx])
if(ruanganPasien = NULL) then
    output("Pasien tidak ditemukan. Silahkan melakukan
    registrasi terlebih dahulu :)\n")
else
    if(cekAntrianPasien(users[idx], ruanganPasien) =
    NULL) then
        output("Kamu belum berada di urutan terdepan
        antrian mana pun. Sabar dulu ya! :(\n")
    else
        if(checkDiagnosis(users[idx]) = false) then
            if(users[idx].riwayat_penyakit) = ""
            then
                output("Kamu belum menerima
                diagnosis apapun dari dokter, jangan
                buru-buru pulang! T^T\n")
            if(users[idx].riwayat_penyakit) =
            "Sehat") then
                output("Kamu tidak ada penyakit
                apapun. Silahkan pulang dan semoga
                sehat selalu! :3\n")
            dequeue(ruanganPasien.antrianPasien,
            idPasienPulang, namaPasienPulang)
        else
            output("Dokter sedang memeriksa
            keadaanmu... \n")
        if (inventoryPasien.count ≠ 0) then
            output("Masih ada obat yang belum
            kamu habiskan, minum semuanya dulu
            yukk! :3\n")

```

```

if(inventoryPasien.count      =      0      and
checkUrutanObat(users[idx])  =  0) then

    output("Maaf, tapi kamu masih belum
bisa pulang T^T !\n\n")
    output("Urutan      MINUM_OBAT      yang
diharapkan:\n")
    urutanHarapan(users[idx])
    output("Urutan MINUM_OBAT yang kamu
minum:\n")

    jumlahObat           ←
    getJumlahObat(users[idx])

    i traversal[0..jumlahObat-1]
    namaObat ← ""
    i traversal[0..obatCount]
        if(obatList[j].id      =
users[idx].perut.data[i])t
        hen
            namaObat           ←
            obatList[j].nama
        if(namaObat = "") then
            break
        if(i = 0)then
            output(namaObat)
        else
            output("->, namaObat)
    output("\\n\\Silahkan kunjungi dokter
untuk      meminta      penawar
(MINUM_PENAWAR) yang sesuai!\\n")
    else

```

```

output("SELAMAT!! :D\n")
output("Kamu sudah dinyatakan sembuh
oleh dokter. Silahkan pulang dan
semoga sehat selalu! :3\n")

users[idx].riwayat_penyakit ← ""
while(not
isStackEmpty(users[idx].perut))do
    pop(users[idx].perut)

dequeue(ruanganPasien.antrianPasien,
idPasienPulang, namaPasienPulang)

```

### 13. F14 - Daftar Checkup

```

procedure daftarCheckUp(input/output userNow: User)
{ I.S.: userNow adalah pasien yang sedang login dan
telah valid. Data ruangan, dokter, dan antrian tersedia.
}
{ F.S.: Pasien dimasukkan ke dalam antrian ruangan
berdasarkan pilihan. Sistem menampilkan informasi lokasi
ruangan dan posisi pasien dalam antrian. }

```

#### **KAMUS LOKAL**

```

pasienID : integer
idx : integer
suhu, saturasi, berat : real
sistol, diastol, detak, gula, tinggi, kolesterol,
trombosit : integer
opsi : array [1..100, 1..2] of integer
count, pilihan, i, j : integer

```

#### **ALGORITMA**

```

if ((sudahTerdaftar(userNow.id)) then
    output("Eh, kamu udah daftar buat check-up
nih!")

```

```

        output("Yuk, tunggu giliran dulu. Nanti aja
daftarnya lagi.")
    else
        idx <- getUserIndex(userNow.username, users,
userCount)
        output("--- FORMULIR CHECK-UP PASIEN ---")
        output("Masukin data kesehatan kamu dulu ya.
Jangan asal, ini penting!")
        input(suhu)
        while (suhu ≤ 0) do
            output("Suhu tubuh harus di atas nol
yaa.")
            input(suhu)
        input(sistol, diastol)
        while (sistol ≤ 0 or diastol ≤ 0) do
            output("Hmm... tekanan darah gak bisa
negatif dong!")
            input(sistol, diastol)
        input(detak)
        while (detak ≤ 0) do
            output("Detak jantung kamu ilang? Masukin
lagi yg benar.")
            input(detak)
        input(saturasi)
        while (saturasi ≤ 0) do
            output("Jangan ngasal... Napas dulu baru
isi!")
            input(saturasi)
        input(gula)
        while (gula ≤ 0) do
            output("Manis dikit gapapa, asal jangan
kosong!")
            input(gula)
        input(berat)
        while (berat ≤ 0) do
            output("Yakin berat badan kamu 0? Coba
isi ulang.")
            input(berat)

```

```

input(tinggi)
while (tinggi ≤ 0) do
    output("Coba      diukur      lagi      tinggi
badannya .")
    input(tinggi)
input(kolesterol)
while (kolesterol ≤ 0) do
    output("Kolesterol gak boleh minus yaa .")
    input(kolesterol)
input(trombosit)
while (trombosit ≤ 0) do
    output("Trombositnya harus diisi yang
bener yaa .")
    input(trombosit)
{ Simpan data }
users[idx].suhu_tubuh ← suhu
users[idx].tekanan_darah_sistolik ← sistol
users[idx].tekanan_darah_diastolik ← diastol
users[idx].detak_jantung ← detak
users[idx].saturasi_oksigen ← saturasi
users[idx].kadar_gula_darah ← gula
users[idx].berat_badan ← berat
users[idx].tinggi_badan ← tinggi
users[idx].kadar_kolesterol ← kolesterol
users[idx].trombosit ← trombosit
output("--- DOKTER YANG SIAP NANGGEPIN KAMU
---")
count ← 0
for i ← 0 to panjang_denah - 1 do
    for j ← 0 to lebar_denah - 1 do
        if (ruanganList[i][j].idDokter ≠ 0
and ruanganList[i][j].antrianPasien.length <
kapasitas_ruangan + kapasitas_antrian) then
            count ← count + 1
            opsi[count][1] ← i
            opsi[count][2] ← j
            output(count, ". Dr. ", ruanganList[i][j].usernameDokte

```

```

r, " - Ruangan ", (char) ('A' +
i), j + 1, " (Antrian: ",
ruanganList[i][j].antrianPasien
.length, " pasien") )

if (count = 0) then
    output ("Ups... Semua dokter lagi
istirahat nih. Coba lagi nanti ya!")
    ->
input (pilihan)
while (pilihan < 1 or pilihan > count) do
    output ("Nomor pilihan gak valid, cek
daftar lagi yaa!")
    input (pilihan)
i ← opsi[pilihan][1]
j ← opsi[pilihan][2]
enqueue(ruanganList[i][j].antrianPasien,
userNow.id, userNow.username)
ruanganList[i][j].idPasien[ruanganList[i][j].t
otalPasien] ← userNow.id
ruanganList[i][j].totalPasien ←
ruanganList[i][j].totalPasien + 1
output ("Mantap! Check-up kamu udah masuk
daftar.")
output ("Antrian kamu sekarang di Dr. ",
ruanganList[i][j].usernameDokter, " - Ruangan
", (char) ('A' + i), j + 1)
output ("Nomor urut kamu: ",
ruanganList[i][j].antrianPasien.length)

```

#### 14. F15 - Antrian Saya!

```

procedure antrianSaya(input userNow: User)
{ I.S.: userNow adalah user dengan role "pasien".
Terdapat data ruangan dan antrian yang aktif. }
{ F.S.: Menampilkan posisi userNow dalam antrian pada
ruangan tertentu (jika ada) dan statusnya apakah sudah
di dalam atau masih di luar ruangan. }

```

#### **KAMUS LOKAL**

```

found : integer { 0 atau 1 → sebagai boolean }
i, j : integer { indeks baris dan kolom denah }
q : Queue { antrian di ruangan }
current : ^Node { pointer ke pasien di antrian }
posisi : integer
kapasitas : integer { kapasitas ruangan }
antrianLuar : integer

```

### **ALGORITMA**

```

if (userNow.role ≠ "pasien") then
    output("Hmmm... fitur ini cuma bisa dipakai
sama pasien ya!")
    ->
found ← 0
for i ← 0 to panjang_denah - 1 do
    for j ← 0 to lebar_denah - 1 do
        q ← ruanganList[i][j].antrianPasien
        current ← q.front
        posisi ← 1
        while (current ≠ NULL) do
            if (current.usernamePasien      =
userNow.username) then
                output("===== STATUS ANTRIAN
KAMU =====")
                output("Ketemu! Kamu lagi
nunggu di ruangan ", (char)('A' + i), j + 1, " bareng
Dokter ", ruanganList[i][j].usernameDokter)
                output("Posisimu sekarang di
antrian: ", posisi, " dari ", q.length, " pasien")
                kapasitas ← kapasitas_ruangan
                if (posisi ≤ kapasitas) then
                    output("Kamu udah DI DALAM
ruangan. Siap-siap ya, giliranmu bentar lagi!")
                else
                    antrianLuar ← posisi -
                    kapasitas
                    output("Kamu masih DI LUAR
ruangan. Tunggu ", antrianLuar, " pasien lagi buat
masuk, ya!")

```

```

        found <- 1
        break
        current <- current.next
        posisi <- posisi + 1
        if (found = 1) then
            ->
        if (found = 0) then
            output("Eits, kamu belum ada di antrian mana
pun nih!")
            output("Yuk daftar dulu buat check-up lewat
command: DAFTAR_CHECKUP")

```

## 15. F16 - Minum Obat

```

procedure minumObat(input/output user: User)
{ I.S.: user adalah pasien yang memiliki inventory obat.
Stack perut telah terdefinisi. }

{ F.S.: Pasien memilih satu obat dari daftar yang
dimiliki dan meminumnya (masuk ke stack perut). Obat
dihapus dari inventory setelah diminum. }

```

### KAMUS LOKAL

```

inv : InventoryPasien
idx : integer { index pasien di array users }
i : integer
id : integer
pilihan : integer { nomor obat yang dipilih }
id_obat : integer

```

### ALGORITMA

```

inv <- getInventoryByUser(user)
idx <- getUserIndex(user.username, users, userCount)
if (users[idx].riwayat_penyakit = "Sehat") then
    output("Kamu Boleh Pulang, Segera ke pulang
DOK!!")
    ->
if (inv = NULL or inv.count = 0) then

```

```

        output("Eh, obatmu kosong loh! Gak ada yang
bisa diminum sekarang :)")
    ->
output("===== DAFTAR OBAT =====")
for i ← 0 to inv.count - 1 do
    id ← inv.obat_id[i]
    if (namaObat(id) ≠ "") then
        output(i + 1, ". ", namaObat(id))
    else
        output(i + 1, ". [Obat tidak dikenal - ID
", id, "]")
    output("Pilih nomor obat yang mau kamu minum (1-",
inv.count, "): ")
    input(pilihan)
    if (pilihan bukan angka) then
        output("Input tidak valid! Masukkan angka
ya.")
        kosongkan buffer input
    ->
    if (pilihan < 1 or pilihan > inv.count) then
        output("Waduh! Nomor yang kamu pilih gak ada
di daftar!")
    ->
    id_obat ← inv.obat_id[pilihan - 1]
    push(users[idx].perut, id_obat)
    output("GLEKGLEKGLEK... ", namaObat(id_obat), "
berhasil diminum! Rasanya... unik ya^^")
    removeInventoryByUser(inv, pilihan - 1)
    output("Inventory diperbarui. Obat      ",
namaObat(id_obat), " udah keluar dari kantong ajaib
kamu!")

```

#### 16. F17 - Minum Penawar

```

procedure addInventoryByuser (input/output inv :
InventoryPasien, input id : integer)
{I.S : inv sudah terdefinisi}
{F.S : Menambahkan id obat baru ke dalam inventory dan
menambahkan kapasitas inventory}

```

```
procedure minumPenawar(input users : User)
{I.S : users terdefinisi}
{F.S : Melakukan pengecekan apakah Stack perut users
kosong. Jika tidak kosong, maka dikeluarkan obat yang
terakhir kali diminum oleh users dan memasukkan kembali
obat ke dalam inventory}
```

#### **KAMUS LOKAL**

```
inv : InventoryPasien
idx : integer
namaObatKeluar : string
idObatKeluar : integer
```

#### **ALGORITMA**

```
inv ← getInventoryByUser(user)
idx ← getUserIndex(user.username, users, userCount)
if(users[idx].riwayat_penyakit = "Sehat") then
    output("Kamu Boleh Pulang, Segera ke pulang
DOK!!\n")
if(inv = NULL) then
    output("Kamu belum menerima obat dari dokter. Tidak
boleh meminta obat penawar!! >:\n")
else
    if(users[idx].perut.top = -1) then
        output("Perut kosong!! Belum ada obat yang
dimakan. >:\n")
    else
        idObatKeluar ← peek(users[idx].perut)
        pop(users[idx].perut)
        namaObatKeluar ← namaObat(idObatKeluar)
```

```

output("Uwekkk :O !!!\nInventory diperbarui.
Obat", namaObatKeluar, "keluar dan kembali ke
kantong ajaib kamu!\n")
addInventoryByUser(inv, idObatKeluar)

```

### 17. F18 - Exit

```

procedure exit_program()
{I.S.: Program sedang berjalan, user belum keluar.}
{F.S.: Program dihentikan, data disimpan jika pengguna
memilih untuk menyimpannya.}

```

#### **KAMUS LOKAL**

```

pilihan      : string
folder_name   : string
len          : integer

```

#### **ALGORITMA**

repeat

```

    output("Apakah Anda ingin menyimpan data sebelum
keluar? (ya/tidak): ")
    input(pilihan)
until (pilihan = "ya" or pilihan = "tidak")

```

if (pilihan = "ya") then

```

    output("Masukkan nama folder untuk menyimpan
data: ")
    read(folder_name)

```

len ← panjang(folder\_name)

if (len > 0 and folder\_name[len - 1] = '\n')

then

folder\_name[len - 1] ← '\0'

save\_data(folder\_name)

```

    output("Program dihentikan. Semua dokter, pasien,
dan manajer telah meninggalkan rumah sakit.")

```

## 18. D02 - Lihat Antrian

```
procedure lihatSemuaAntrian ()  
{ I.S.: Variabel global panjang_denah, lebar_denah,  
ruanganList, dan kapasitas_ruangan terdefinisi dan  
berisi data yang valid. Fungsi gambarDenah diasumsikan  
ada dan menampilkan denah ruangan. }  
{ F.S.: Denah ruangan ditampilkan ke layar. Untuk setiap  
ruangan yang memiliki dokter (idDokter bukan 0),  
informasi ruangan tersebut ditampilkan, meliputi nama  
ruangan, kapasitas, nama dokter, daftar pasien di dalam  
ruangan (hingga kapasitas ruangan), dan daftar pasien  
yang sedang mengantri di luar ruangan tersebut. Jika  
tidak ada pasien di dalam ruangan atau di antrian, pesan  
yang sesuai akan ditampilkan untuk ruangan tersebut.  
}Baik, saya akan menyesuaikan format notasi algoritma  
untuk fungsi lihatSemuaAntrian dan fungsi-fungsi  
berikutnya dengan menggunakan indentasi untuk menandakan  
blok program dan tanpa menggunakanendif atau endfor.
```

KAMUS GLOBAL (relevan untuk modul ini)

```
panjang_denah: integer  
lebar_denah: integer  
kapasitas_ruangan: integer  
ruanganList: array [0..MAX_RUANGAN-1] of array  
[0..MAX_RUANGAN-1] of Ruangan
```

```

Ruangan: type { (tipe data Ruangan diasumsikan sudah
terdefinisi, minimal memiliki atribut idDokter,
usernameDokter, antrianPasien (sebuah Queue)) }

Queue: type { (tipe data Queue untuk antrianPasien,
minimal memiliki atribut front (pointer ke Node
pertama)) }

Node: type { (tipe data Node untuk elemen dalam Queue,
minimal memiliki atribut usernamePasien dan next
(pointer ke Node berikutnya)) }

procedure lihatSemuaAntrian ()
{ I.S.: Variabel global panjang_denah, lebar_denah,
ruanganList, dan kapasitas_ruangan terdefinisi dan
berisi data yang valid. Fungsi gambarDenah diasumsikan
ada dan menampilkan denah ruangan. }

{ F.S.: Denah ruangan ditampilkan ke layar. Untuk setiap
ruangan yang memiliki dokter (idDokter bukan 0),
informasi ruangan tersebut ditampilkan, meliputi nama
ruangan, kapasitas, nama dokter, daftar pasien di dalam
ruangan (hingga kapasitas ruangan), dan daftar pasien
yang sedang mengantri di luar ruangan tersebut. Jika
tidak ada pasien di dalam ruangan atau di antrian, pesan
yang sesuai akan ditampilkan untuk ruangan tersebut. }

```

#### KAMUS LOKAL

```

i, j: integer

r: Ruangan

namaRuang: string[4] { Ukuran disesuaikan dengan
kebutuhan, misal "A1", "B10" }

```

```

cur: pointer to Node
index: integer
adaPasienDalam: boolean
adaPasienAntrian: boolean
antriIndex: integer

ALGORITMA
gambarDenah() { Menampilkan grid denah ruangan }

i traversal [0..panjang_denah-1]
    j traversal [0..lebar_denah-1]
    r <- ruanganList[if (r.idDokter != 0) then
        namaRuang <- character('A' + i) +
        integer_to_string(j + 1) { Menggabungkan karakter baris
        dan nomor kolom }

output("") { Baris baru sebelum header ruangan
}
output("===== " + namaRuang + "
=====")
output("Kapasitas : " +
integer_to_string(kapasitas_ruangan))
output("Dokter      : Dr. " + r.usernameDokter)

output("Pasien di dalam ruangan:")

cur <- r.antrianPasien.front
index <- 0
adaPasienDalam <- false

```

```

{ Pasien dalam ruangan: index <
kapasitas_ruangan }

while (cur != NULL) and (index <
kapasitas_ruangan) do

    output(" " + integer_to_string(index + 1) +
". " + cur.usernamePasien)

    adaPasienDalam <- true

    cur <- cur.next

    index <- index + 1


if (not adaPasienDalam) then

    output(" Tidak ada pasien di dalam ruangan
saat ini.")

    output("Pasien di antrian:")

    adaPasienAntrian <- false

    antriIndex <- 1


{ Sisanya dianggap sebagai pasien di antrian }

while (cur != NULL) do

    output(" " + integer_to_string(antriIndex) +
". " + cur.usernamePasien)

    adaPasienAntrian <- true

    cur <- cur.next

    antriIndex <- antriIndex + 1


if (not adaPasienAntrian) then

```

```
output(" Tidak ada pasien di antrian saat  
ini.")i] [j]
```

## 19. D03 - Load

```
procedure trim_newline (input/output str: string)  
  
{ I.S.: str adalah sebuah string yang mungkin memiliki  
karakter newline ('\n') atau carriage return ('\r') di  
akhirnya. }  
  
{ F.S.: Karakter newline atau carriage return di akhir  
string str telah dihapus. }  
  
function getObatData (input filename: string, output  
count: integer) -> pointer to array of Obat  
  
{ I.S.: filename adalah nama file CSV yang valid berisi  
data obat. count adalah integer. }  
  
{ F.S.: Mengembalikan pointer ke array of Obat yang  
berisi data dari file. count berisi jumlah obat yang  
berhasil dibaca. Jika file tidak dapat dibuka atau  
terjadi error alokasi memori, mengembalikan NULL dan  
count tidak terdefinisi (atau 0). }  
  
function getPenyakitData (input filename: string, output  
count: integer) -> pointer to array of Penyakit
```

{ I.S.: filename adalah nama file CSV yang valid berisi data penyakit. count adalah integer. }

{ F.S.: Mengembalikan pointer ke array of Penyakit yang berisi data dari file. count berisi jumlah penyakit yang berhasil dibaca. Jika file tidak dapat dibuka atau terjadi error alokasi memori, mengembalikan NULL dan count tidak terdefinisi (atau 0). }

**function** getObatPenyakitData (input filename: string,  
output count: integer) -> pointer to array of  
ObatPenyakit

{ I.S.: filename adalah nama file CSV yang valid berisi data mapping obat dan penyakit. count adalah integer. }

{ F.S.: Mengembalikan pointer ke array of ObatPenyakit yang berisi data dari file. count berisi jumlah mapping yang berhasil dibaca. Jika file tidak dapat dibuka atau terjadi error alokasi memori, mengembalikan NULL dan count diinisialisasi ke 0. }

**function** getId (input line: string) -> integer

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field pertamanya merupakan representasi integer dari ID. }

{ F.S.: Mengembalikan nilai integer dari ID yang terdapat pada field pertama string line. }

**procedure** getUsername (input line: string, output username: string)

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field keduanya merupakan username. username adalah variabel string. }

{ F.S.: username berisi string username yang terdapat pada field kedua string line. }

**procedure** getPassword (input line: string, output password: string)

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field ketiganya merupakan password. password adalah variabel string. }

{ F.S.: password berisi string password yang terdapat pada field ketiga string line. }

**procedure** getRole (input line: string, output role: string)

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field keempatnya merupakan role. role adalah variabel string. }

{ F.S.: role berisi string role yang terdapat pada field keempat string line. }

**procedure** getRiwayatPenyakit (input line: string, output riwayat\_penyakit: string)

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field kelimanya merupakan riwayat penyakit. riwayat\_penyakit adalah variabel string. }

{ F.S.: riwayat\_penyakit berisi string riwayat penyakit yang terdapat pada field kelima string line. }

**function** getSuhuTubuh (input line: string) -> real

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field keenamnya merupakan representasi float dari suhu tubuh. }

{ F.S.: Mengembalikan nilai real dari suhu tubuh yang terdapat pada field keenam string line. }

**function** getTekananDarahSistolik (input line: string) -> integer

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field ketujuhnya merupakan representasi integer dari tekanan darah sistolik. }

{ F.S.: Mengembalikan nilai integer dari tekanan darah sistolik yang terdapat pada field ketujuh string line. }

**function** getTekananDarahDiastolik (input line: string) -> integer

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field kedelapannya merupakan representasi integer dari tekanan darah diastolik. }

{ F.S.: Mengembalikan nilai integer dari tekanan darah diastolik yang terdapat pada field kedelapan string line. }

**function** getDetakJantung (input line: string) -> integer

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field kesembilannya merupakan representasi integer dari detak jantung. }

{ F.S.: Mengembalikan nilai integer dari detak jantung yang terdapat pada field kesembilan string line. }

**function** getSaturasiOksigen (input line: string) -> real

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field kesepuluhnya merupakan representasi float dari saturasi oksigen. }

{ F.S.: Mengembalikan nilai real dari saturasi oksigen yang terdapat pada field kesepuluh string line. }

**function** getKadarGulaDarah (input line: string) -> integer

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field kesebelasnya merupakan representasi integer dari kadar gula darah. }

{ F.S.: Mengembalikan nilai integer dari kadar gula darah yang terdapat pada field kesebelas string line. }

**function** getBeratBadan (input line: string) -> real

{ I.S.: line adalah sebuah string (baris dari file CSV) yang field keduabelasnya merupakan representasi float dari berat badan. }

```
{ F.S.: Mengembalikan nilai real dari berat badan yang  
terdapat pada field keduabelas string line. }  
  
function getTinggiBadan (input line: string) -> real  
  
{ I.S.: line adalah sebuah string (baris dari file CSV)  
yang field ketigabelasnya merupakan representasi float  
dari tinggi badan. }  
  
{ F.S.: Mengembalikan nilai real dari tinggi badan yang  
terdapat pada field ketigabelas string line. }  
  
function getKadarKolesterol (input line: string) ->  
integer  
  
{ I.S.: line adalah sebuah string (baris dari file CSV)  
yang field keempatbelasnya merupakan representasi  
integer dari kadar kolesterol. }  
  
{ F.S.: Mengembalikan nilai integer dari kadar  
kolesterol yang terdapat pada field keempatbelas string  
line. }  
  
function getTrombosit (input line: string) -> integer  
  
{ I.S.: line adalah sebuah string (baris dari file CSV)  
yang field kelimabelasnya merupakan representasi integer  
dari trombosit. }  
  
{ F.S.: Mengembalikan nilai integer dari trombosit yang  
terdapat pada field kelimabelas string line. }
```

```

function getUserIndex (input username: string, input
users_arr: array of User, input user_arr_count: integer)
-> integer

{ I.S.: username adalah string yang dicari. users_arr
adalah array global users yang berisi data pengguna.
user_arr_count adalah userCount global yang menyatakan
jumlah pengguna dalam array. }

{ F.S.: Mengembalikan indeks dari pengguna dengan
username yang cocok (case-insensitive) dalam array
users_arr. Mengembalikan -1 jika tidak ditemukan. }

procedure str_toLower (input/output str: string)

{ I.S.: str adalah sebuah string. }

{ F.S.: Semua karakter huruf kapital dalam str diubah
menjadi huruf kecil. }

procedure getUserData (input filename: string)

{ I.S.: filename adalah nama file CSV yang valid berisi
data pengguna. Variabel global users dan userCount belum
terisi atau berisi data lama. }

{ F.S.: Variabel global users terisi dengan data
pengguna dari file, dan userCount diperbarui dengan
jumlah pengguna yang berhasil dibaca. Jika file tidak
bisa dibuka, pesan error ditampilkan. }

function getUsernameById (input id: integer) -> string

```

```

{ I.S.: id adalah integer. Variabel global users dan
userCount sudah terisi. }

{ F.S.: Mengembalikan string username dari pengguna
dengan id yang cocok. Mengembalikan "-" jika tidak
ditemukan. }

function readIntsFromLine (input line: string, output
arr_output: array of integer, input maxInts: integer) ->
integer

{ I.S.: line adalah sebuah string yang berisi
angka-angka yang dipisahkan oleh karakter non-numerik.
arr_output adalah array untuk menyimpan integer yang
dibaca. maxInts adalah ukuran maksimum arr output. }

{ F.S.: arr output berisi integer yang berhasil dibaca
dari line. Mengembalikan jumlah integer yang berhasil
dibaca dan disimpan dalam arr output. }

function getUserById (input id: integer) -> pointer to
User

{ I.S.: id adalah integer. Variabel global users dan
userCount sudah terisi. }

{ F.S.: Mengembalikan pointer ke objek User dengan id
yang cocok dalam array global users. Mengembalikan NULL
jika tidak ditemukan. }

procedure loadConfig (input folder: string)

```

```
{ I.S.: folder adalah nama direktori yang berisi file "config.txt". Variabel global seperti panjang_denah, lebar_denah, kapasitas_ruangan, kapasitas_antrian, jumlah_ruangan, ruanganList, daftar_inventory, jumlah_inventory, dan stack perut pada users belum terkonfigurasi atau berisi data lama. }
```

```
{ F.S.: Variabel-variabel global konfigurasi (panjang_denah, lebar_denah, dst.) terisi berdasarkan data dari file "config.txt". Data ruangan, antrian pasien, inventory pasien, dan stack perut pengguna dimuat. Jika file tidak dapat dibuka, pesan error ditampilkan. }
```

```
procedure load_all_data (input folder: string)  
  
{ I.S.: folder adalah nama direktori yang berisi semua file data (user.csv, obat.csv, penyakit.csv, obat_penyakit.csv, config.txt). Variabel-variabel global data (users, obatList, penyakitList, obatPenyakitList, dan variabel konfigurasi) belum terisi atau berisi data lama. }  
  
{ F.S.: Semua data dari file-file dalam direktori folder dimuat ke dalam variabel-variabel global yang sesuai. Jika ada file yang gagal dibuka, program akan keluar dengan status error. Pesan keberhasilan ditampilkan jika semua data berhasil dimuat. }
```

Baik, saya akan membuat notasi algoritma untuk fungsi `load_all_data` sesuai dengan panduan yang telah Anda berikan sebelumnya, mencakup judul, kamus (jika ada yang spesifik dan belum tercakup di global), dan algoritma.

KAMUS GLOBAL (diasumsikan sudah terdefinisi berdasarkan konteks sebelumnya)

```
users: array [0..MAX_USERS-1] of User  
userCount: integer  
  
obatList: pointer to array of Obat  
  
obatCount: integer  
  
penyakitList: pointer to array of Penyakit  
  
penyakitCount: integer  
  
obatPenyakitList: pointer to array of ObatPenyakit  
  
obatPenyakitCount: integer  
  
(Variabel-variabel global lain yang relevan seperti yang diisi oleh loadConfig juga diasumsikan ada)  
  
EXIT_FAILURE: constant integer { nilai konstanta untuk  
status keluar program karena error }
```

## PROSEDUR/FUNGSI

```
procedure load_all_data (input folder: string)

{ I.S.: folder adalah string yang merepresentasikan nama
direktori yang berisi semua file data yang diperlukan
(user.csv, obat.csv, penyakit.csv, obat_penyakit.csv,
config.txt). Variabel-variabel global data (seperti
users, userCount, obatList, obatCount, penyakitList,
penyakitCount, obatPenyakitList, obatPenyakitCount, dan
variabel konfigurasi lainnya) belum terisi atau berisi
data lama. Prosedur getUserData, fungsi getObatData,
getPenyakitData, getObatPenyakitData, dan prosedur
loadConfig diasumsikan sudah terdefinisi dan dapat
diakses. }

{ F.S.: Semua data dari file-file yang berada dalam
direktori folder dimuat ke dalam variabel-variabel
global yang sesuai. Ini meliputi data pengguna, data
obat, data penyakit, data relasi obat-penyakit, dan
konfigurasi sistem. Jika salah satu file penting gagal
dibuka, program akan menampilkan pesan error fatal ke
piranti error standar dan berhenti dengan status
EXIT_FAILURE. Jika semua data berhasil dimuat, pesan
konfirmasi " Data berhasil diload dari folder: [nama
folder]" akan ditampilkan ke layar. }
```

## KAMUS LOKAL

```
path: string { variabel untuk menyimpan path lengkap ke  
file }
```

```
test: FilePointer { variabel untuk menguji keberadaan  
file }
```

#### ALGORITMA

```
{ USER }
```

```
path <- folder + "/user.csv"
```

```
test <- open_for_read(path) { Membuka file untuk dibaca  
}
```

```
if (test = NULL) then { Jika file tidak bisa dibuka }
```

```
    output("Fatal error: Gagal membuka file " + path) {  
        Menulis ke stderr }
```

```
    exit(EXIT_FAILURE) { Menghentikan program }
```

```
close(test) { Menutup file }
```

```
getUserData(path) { Memuat data pengguna dari file yang  
path-nya valid }
```

```
{ OBAT }
```

```
path <- folder + "/obat.csv"
```

```
test <- open_for_read(path)
```

```
if (test = NULL) then
```

```

output("Fatal error: Gagal membuka file " + path)

exit(EXIT_FAILURE)

close(test)

obatList <- getObatData(path, obatCount) { Memuat data
obat dan mengupdate obatCount }

{ PENYAKIT }

path <- folder + "/penyakit.csv"

test <- open_for_read(path)

if (test = NULL) then

output("Fatal error: Gagal membuka file " + path)

exit(EXIT_FAILURE)

close(test)

penyakitList <- getPenyakitData(path, penyakitCount) {
Memuat data penyakit dan mengupdate penyakitCount }

{ OBAT_PENYAKIT }

path <- folder + "/obat_penyakit.csv"

test <- open_for_read(path)

if (test = NULL) then

```

```

output("Fatal error: Gagal membuka file " + path)

exit(EXIT_FAILURE)

close(test)

obatPenyakitList <- getObatPenyakitData(path,
obatPenyakitCount) { Memuat data relasi obat-penyakit
dan mengupdate obatPenyakitCount }

{ CONFIG }

path <- folder + "/config.txt"

test <- open_for_read(path)

if (test = NULL) then

    output("Fatal error: Gagal membuka file " + path)

    exit(EXIT_FAILURE)

close(test)

loadConfig(folder) { Memuat data konfigurasi dari
direktori folder }

output(" Data berhasil diload dari folder: " + folder)

```

## 20. D04 - Save

```

function folder_exists (string folder_name) → integer
{ I.S.: Diberikan nama folder. }

```

```

{F.S.: Mengembalikan 1 jika folder ada, 0 jika tidak ada.}

function is_folder_empty (string folder_name) → integer
{I.S.: Diberikan nama folder.}
{F.S.: Mengembalikan 1 jika folder kosong, 0 jika tidak kosong.}

procedure save_config (input folder_name : string)
{I.S.: Terdapat konfigurasi denah, ruangan, inventory pasien, dan stack perut di memori.}
{F.S.: Data konfigurasi disimpan dalam file config.txt di folder folder_name.}

procedure save_data(input folder_name : string)
{I.S.: Terdefinisi data pengguna, obat, penyakit, relasi obat-penyakit, konfigurasi ruangan, inventory pasien, dan data stack perut di memori.}
{F.S.: Semua data disimpan ke dalam folder folder_name dalam bentuk file .csv dan .txt. Folder dibuat jika belum ada.}

```

#### KAMUS LOKAL

folder_name	:	string
path	:	string
f_user	:	<u>FILE of</u> string
f_obat	:	<u>FILE of</u> string
f_penyakit	:	<u>FILE of</u> string
f_op	:	<u>FILE of</u> string
i, j	:	<u>integer</u>

#### ALGORITMA

```

if (not folder_exists(folder_name)) then
    mkdir(folder_name, 0777)
    output("Folder " + folder_name + "belum ada.
Membuat folder baru...")
if (is_folder_empty(folder_name)) then

```

```

output("Folder " + folder_name + "ditemukan dan
kosong.")

output("Berhasil menyimpan data di folder " +
folder_name + ".")  
  

{ Menyimpan user.csv }
path <- folder_name + "/user.csv"
assign(f_user, path)
rewrite(f_user)
write(f_user,
"id;username;password;role;riwayat_penyakit;suhu_tubuh;.
..;trombosit")  
  

i transversal [0..userCount - 1]
    write(f_user, users[i].id + ";" +
users[i].username + ";" + users[i].password + ";" + ...)
    close(f_user)  
  

{ Menyimpan obat.csv }
path <- folder_name + "/obat.csv"
assign(f_obat, path)
rewrite(f_obat)
write(f_obat, "obat_id;nama_obat")
i <- 0
i transversal [0..obatCount - 1]
    write(f_obat, obatList[i].id + ";" +
obatList[i].nama)
    close(f_obat)  
  

{ Menyimpan penyakit.csv }
path <- folder_name + "/penyakit.csv"
assign(f_penyakit, path)
rewrite(f_penyakit)
write(f_penyakit,
"id;nama_penyakit;suhu_tubuh_min;...;trombosit_max")
i <- 0
i transversal [0..penyakitCount - 1]

```

```

write(f_penyakit, penyakitList[i].id + ";" +
penyakitList[i].name_penyakit + ";" + ...)
close(f_penyakit)

{ Menyimpan obat_penyakit.csv }
path ← folder_name + "/obat_penyakit.csv"
assign(f_op, path)
rewrite(f_op)
write(f_op, "obat_id;penyakit_id;urutan_minum")
i ← 0
i transversal [0..obatPenyakitCount - 1]
    write(f_op, obatPenyakitList[i].obat_id + ";" +
obatPenyakitList[i].penyakit_id + ";" + ...)
close(f_op)

{ Menyimpan config.txt }
save_config(folder_name)

```

## 21. B02 - Denah Dinamis

procedure ubahDenah()

{I.S. panjang\_denah\_eff dan lebar\_denah\_eff dihitung dari posisi dokter dalam ruanganList.}

{F.S.Jika ukuran baru lebih kecil dari batas efektif, tampilkan pesan gagal karena ada dokter di luar area. Jika ukuran valid, panjang\_denah dan lebar\_denah diperbarui dengan nilai baru dan tampilkan pesan sukses.}

### KAMUS LOKAL

newPanjang, newLebar: integer  
i, j: integer

### ALGORTIMA

```

i traversal [0..panjang_denah - 1]
j traversal [0..lebar_denah - 1]
    if ruanganList[i][j].idDokter ≠ 0 then
        if (i + 1 > panjang_denah_eff) then

```

```

            panjang_denah_eff ← i + 1
        if (j + 1 > lebar_denah_eff) then
            lebar_denah_eff ← j + 1

output ("Masukkan ukuran denah yang baru: ")
input (newPanjang, newLebar)

if (newLebar < lebar_denah_eff or newPanjang <
panjang_denah_eff) then
    cariRuang(baris, kolom, newLebar, newPanjang)
    output("Tidak dapat mengubah ukuran denah.
Ruangan ", chr(baris + 'A'), kolom + 1, " masih
ditempati oleh Dr. ",
ruanganList[baris][kolom].usernameDokter)
else
    panjang_denah ← newPanjang
    lebar_denah ← newLebar
    output("Denah rumah sakit berhasil diubah
menjadi ", newPanjang, " baris dan ",
newLebar, " kolom.")

```

```

procedure      cariRuang(output baris : integer, output
kolom : integer, input newLebar : integer, input
newPanjang : integer)
{I.S. Diberikan ukuran denah baru (newLebar,
newPanjang).}
{F.S. Jika terdapat ruangan di luar ukuran baru yang
masih ditempati dokter, maka *baris dan *kolom akan
berisi koordinat ruangan tersebut.}

```

```

procedure pindahDokter()
{I.S. Diberikan dua kode ruangan (ruang1, ruang2) yang
valid.}
{F.S. Jika ruangan asal kosong, tampilkan pesan gagal.
Jika ruangan tujuan kosong, data dokter dipindahkan dari

```

```
asal ke tujuan. Jika ruangan tujuan sudah terisi,  
tampilkan pesan gagal.}
```

#### KAMUS LOKAL

```
ruang1, ruang2: string  
baris1, kolom1, baris2, kolom2: integer  
r1, r2: Ruangan
```

#### ALGORITMA

```
output ("Masukkan ruangan asal dan tujuan (misal: A1  
B2) : ")
```

```
input (ruang1, ruang2)
```

```
baris1 ← ord(ruang1[0]) - ord('A')  
kolom1 ← stringToInt(ruang1[1..]) - 1  
baris2 ← ord(ruang2[0]) - ord('A')  
kolom2 ← stringToInt(ruang2[1..]) - 1
```

```
r1 ← ruanganList [baris1] [kolom1]  
r2 ← ruanganList [baris2] [kolom2]
```

```
if (baris1 < 0 or baris1 > panjang_denah-1 or  
kolom1 < 0 or kolom1 > lebar_denah-1) then  
    output ("Pemindahan gagal. Ruangan ",  
            chr(baris1 + 'A'), kolom1 + 1, " Tidak Ada.")  
else if (baris2 < 0 or baris2 > panjang_denah-1 or  
kolom2 < 0 or kolom2 > lebar_denah-1) then  
    output ("Pemindahan gagal. Ruangan ",  
            chr(baris2 + 'A'), kolom2 + 1, " Tidak Ada.")  
else if (r1.idDokter = 0) then
```

```
output("Pemindahan gagal. Ruangan " ,  
chr(baris1 + 'A') , kolom1 + 1 , " Kosong .")  
else if (r2.idDokter ≠ 0) then  
output("Pemindahan gagal. Ruangan " ,  
chr(baris2 + 'A') , kolom2 + 1 , " Sudah  
ditempati .")  
else  
ruanganList [baris2] [kolom2] ← r1  
ruanganList [baris1] [kolom1] ← r2  
output("Dr. " , r1.usernameDokter , " berhasil  
dipindahkan dari ruangan " ,  
chr(baris1 + 'A') , kolom1 + 1 , " ke ruangan " ,  
chr(baris2 + 'A') , kolom2 + 1 , ".")
```

## BAB IX

### TANGKAPAN LAYAR HASIL PENGUJIAN PROGRAM BERDASARKAN FITUR-FITUR PADA SPESIFIKASI

#### F01 - LOGIN

# Kasus 1 : User sudah terdaftar dan password valid

```
>>> LOGIN
Username: furina
Password: fufurinafafa
Login Berhasil! Selamat datang pasien furina.
-----
Masukkan perintah (Ketik [HELP] untuk melihat daftar perintah):
>>> 
```

# Kasus 2 : User sudah terdaftar namun password salah

```
>>> LOGIN
Username: furina
Password: wooooeddan
Password salah untuk user furina.
Lupa password? Ketik [LUPA_PASSWORD] untuk mengganti password.
```

# Kasus 3 : User belum terdaftar

```
>>> LOGIN
Username: hutao
Password: unyah
Pengguna dengan username "hutao" belum terdaftar.
Silakan lakukan REGISTER dengan mengetik [REGISTER]
atau melakukan LOGIN dengan akun yang sudah terdaftar.
```

#### F02 - REGISTER

# Kasus 1 : Nama pasien dengan format yang benar dan nama unik

```
>>> REGISTER
Username: Jerannn
Password: lakersInFive
Pasien Jerannn berhasil ditambahkan!
Silakan melakukan login dengan mengetik [LOGIN].
```

# Kasus 2 : Nama pasien tidak unik

```
>>> REGISTER
Username: jerannn
Password: bestAnimationStudio
Registrasi gagal! Pasien dengan nama jerannn sudah terdaftar.
```

#Kasus 3 : Nama pasien tidak sesuai format

```
>>> REGISTER
Username: kobe24
Masukkan username hanya dengan Huruf A-Z, a-z!!
Username: kobeBryant
Password: THEGOAT
Pasien kobeBryant berhasil ditambahkan!
Silakan melakukan login dengan mengetik [LOGIN].
```

#### F03 - LOGOUT

# Kasus 1 : User dalam status USER\_LOGGED\_OFF

```
>>> LOGOUT
Logout gagal!
Anda belum login, silakan login terlebih dahulu sebelum logout.
```

# Kasus 2 : User dalam status USER\_LOGGED\_ON dan User seorang Pasien

```
>>> LOGOUT
Sampai jumpa, pasien pop!
```

# Kasus 3 : User dalam status USER\_LOGGED\_ON dan User seorang Dokter

```
>>> LOGOUT
Sampai jumpa, dokter cacako!
```

# Kasus 4 : User dalam status USER\_LOGGED\_ON dan User seorang Manajer

```
>>> LOGOUT
Sampai jumpa, manager zera!
```

#### F04 - Lupa Password

#kasus 1 : user tidak terdaftar

```
>>> LUPA_PASSWORD
== Lupa Password ==
Username: mrskibidimeow
| Username tidak ditemukan!
```

#kasus 2 : user terdaftar dan kode unik salah

```
== Lupa Password ==
Username: furina
Kode Unik: neuvillette
| Kode unik tidak valid!
```

#kasus 3 : user terdaftar dan kode unik benar dan memasukkan password baru

```
>>> LUPA_PASSWORD
== Lupa Password ==
Username: furina
Kode Unik: furina
Password Baru: neuvillette
| Password berhasil diubah untuk user furina.
Silakan melakukan login kembali dengan mengetik [LOGIN].
```

## F05 - Help

# Kasus 1 : Help pada pasien

```
Masukkan perintah (Ketik [HELP] untuk melihat daftar perintah):
>>> HELP
HELP =====
Halo pasien furina, berikut adalah seluruh fungsi yang dapat kamu akses:
1. LOGOUT
2. DAFTAR_CHECKUP
3. ANTRIAN_SAYA
4. MINUM_OBAT
5. MINUM_PENAWAR
6. DOK_AKU_MAU_PULANG
7. LIHAT_DENAH
8. LIHAT_RUANGAN
9. LIHAT_SEMUA_ANTRIAN
10. EXIT
Footnote: =====
1. Untuk menggunakan aplikasi, silahkan masukkan perintah yang terdaftar.
2. Jangan lupa untuk memasukkan input yang valid.
-----
Masukkan perintah (Ketik [HELP] untuk melihat daftar perintah):
>>> 3

Eits, kamu belum ada di antrian mana pun nih!
Yuk daftar dulu buat check-up lewat command: DAFTAR_CHECKUP
```

# Kasus 2: Interface Help pada dokter

```
>>> HELP
HELP =====
Halo dokter neronimo, berikut adalah seluruh fungsi yang dapat kamu akses:
1. LOGOUT
2. LIHAT_ANTRIAN
3. DIAGNOSIS
4. NGOBATIN
5. LIHAT_DENAH
6. LIHAT_RUANGAN
7. LIHAT_SEMUA_ANTRIAN
8. EXIT
Footnote: =====
1. Untuk menggunakan aplikasi, silahkan masukkan perintah yang terdaftar.
2. Jangan lupa untuk memasukkan input yang valid.
```

# Kasus 3: Interface Help pada manager

```
>>> HELP
HELP =====
Halo manager zero, berikut adalah seluruh fungsi yang dapat kamu akses:
1. LOGOUT
2. TAMBAH_DOKTER
3. ASSIGN_DOKTER
4. LIHAT_USER
5. LIHAT_PASIEN
6. LIHAT_DOKTER
7. CARI_USER
8. CARI_PASIEN
9. CARI_DOKTER
10. LIHAT_DENAH
11. LIHAT_RUANGAN
12. LIHAT_SEMUA_ANTRIAN
13. UBAH_DENAH
14. PINDAH_DOKTER
15. EXIT
Footnote: =====
1. Untuk menggunakan aplikasi, silahkan masukkan perintah yang terdaftar.
2. Jangan lupa untuk memasukkan input yang valid.
```

## F06 / D01 - Denah Rumah Sakit

#Kasus 1 : User ingin melihat denah rumah sakit

```
>>> LIHAT_DENAH
      1   2   3
+---+---+---+
A | A1 | A2 | A3 |
+---+---+---+
B | B1 | B2 | B3 |
+---+---+---+
```

# Kasus 2 : Ruangan berisi dokter dan pasien

```
>>> LIHAT_RUANGAN
Masukkan ruangan yang ingin dilihat: A1
--- Detail Ruangan A1 ---
Kapasitas : 3
Dokter    : Dr. neronimo
Pasien di dalam ruangan:
1. gro
2. kebin
3. stewart
-----
```

#Kasus 3 : Ruangan berisi dokter tanpa pasien

```
>>> LIHAT_RUANGAN
Masukkan ruangan yang ingin dilihat: A1
--- Detail Ruangan A1 ---
Kapasitas : 3
Dokter    : Dr. neronimo
Pasien di dalam ruangan:
1. gro
2. kebin
3. stewart
-----
```

# Kasus 4 : Ruangan kosong

```
>>> LIHAT_RUANGAN
Masukkan ruangan yang ingin dilihat: B2
--- Detail Ruangan B2 ---
Kapasitas : 3
Dokter    : -
Pasien di dalam ruangan:
Tidak ada pasien di dalam ruangan saat ini.
-----
```

# Kasus 1: Melihat data user (user bisa berarti dokter atau pasien) urut berdasarkan ID dan di-sort secara ascending

```
>>> LIHAT_USER
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 1
Urutkan Sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 1
===== DATA USER =====
+---+-----+-----+-----+
| ID | Nama      | Role     | Penyakit |
+---+-----+-----+-----+
| 1  | stewart   | pasien   |           |
+---+-----+-----+-----+
| 2  | gro       | pasien   | COVID-19  |
+---+-----+-----+-----+
| 3  | kebin     | pasien   |           |
+---+-----+-----+-----+
| 4  | pop       | pasien   |           |
+---+-----+-----+-----+
| 5  | opor      | pasien   |           |
+---+-----+-----+-----+
| 6  | nikeb     | pasien   |           |
+---+-----+-----+-----+
| 7  | tuart     | pasien   |           |
+---+-----+-----+-----+
| 8  | minonette | pasien   |           |
+---+-----+-----+-----+
| 9  | tobo      | pasien   |           |
+---+-----+-----+-----+
| 10 | neronimo  | dokter   |           |
+---+-----+-----+-----+
| 10 | ropik     | pasien   |           |
+---+-----+-----+-----+
| 11 | ciciko    | dokter   |           |
+---+-----+-----+-----+
```

```
+---+-----+-----+-----+
| 12 | cacako   | dokter   |           |
+---+-----+-----+-----+
| 13 | kroket   | dokter   |           |
+---+-----+-----+-----+
| 15 | risol     | dokter   |           |
+---+-----+-----+-----+
| 16 | tobokan  | pasien   |           |
+---+-----+-----+-----+
| 20 | popokan  | pasien   |           |
+---+-----+-----+-----+
| 88 | zeru     | manager  |           |
+---+-----+-----+-----+
| 100 | pendatang | pasien   |           |
+---+-----+-----+-----+
| 111 | furina   | pasien   | Diabetes Mellitus |
+---+-----+-----+-----+
```

# Kasus 2: Melihat data user (user bisa berarti dokter atau pasien) urut berdasarkan ID dan di-sort secara descending

```

>>> LIHAT_USER
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 1
Urutkan Sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 2
===== DATA USER =====
+---+-----+-----+
| ID | Nama        | Role      | Penyakit   |
+---+-----+-----+
| 111 | furina     | pasien    | Diabetes Mellitus |
+---+-----+-----+
| 100 | pendatang  | pasien    |             |
+---+-----+-----+
| 88  | zero        | manager   |             |
+---+-----+-----+
| 20  | popokan    | pasien    |             |
+---+-----+-----+
| 16  | tobokan    | pasien    |             |
+---+-----+-----+
| 15  | risol       | dokter    |             |
+---+-----+-----+
| 13  | kroket      | dokter    |             |
+---+-----+-----+
| 12  | cacako     | dokter    |             |
+---+-----+-----+
| 11  | ciciko      | dokter    |             |
+---+-----+-----+
| 10  | neronimo   | dokter    |             |
+---+-----+-----+
+---+-----+-----+
| 10 | ropik       | pasien    |             |
+---+-----+-----+
| 9  | tobo        | pasien    |             |
+---+-----+-----+
| 8  | minonette   | pasien    |             |
+---+-----+-----+
| 7  | tuart       | pasien    |             |
+---+-----+-----+
| 6  | nikeb       | pasien    |             |
+---+-----+-----+
| 5  | opor         | pasien    |             |
+---+-----+-----+
| 4  | pop          | pasien    |             |
+---+-----+-----+
| 3  | kebin       | pasien    |             |
+---+-----+-----+
| 2  | gro          | pasien    | COVID-19   |
+---+-----+-----+
| 1  | stewart     | pasien    |             |
+---+-----+-----+

```

# Kasus 3: Melihat data user (user bisa berarti dokter atau pasien) urut berdasarkan Nama dan di-sort secara ascending

```

>>> LIHAT_USER
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 2
Urutkan Sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 1
===== DATA USER =====
+---+-----+-----+-----+
| ID | Nama          | Role   | Penyakit |
+---+-----+-----+-----+
| 12 | cacako       | dokter |           |
+---+-----+-----+-----+
| 11 | ciciko        | dokter |           |
+---+-----+-----+-----+
| 111| furina        | pasien | Diabetes Mellitus |
+---+-----+-----+-----+
| 2  | gro            | pasien | COVID-19   |
+---+-----+-----+-----+
| 3  | kebin          | pasien |           |
+---+-----+-----+-----+
| 13 | kroket         | dokter |           |
+---+-----+-----+-----+
| 8  | minonette      | pasien |           |
+---+-----+-----+-----+
| 10 | neronimo       | dokter |           |
+---+-----+-----+-----+
| 6  | nikeb          | pasien |           |
+---+-----+-----+-----+
| 5  | opor            | pasien |           |
+---+-----+-----+-----+
| 100| pendatang      | pasien |           |
+---+-----+-----+-----+
+---+-----+-----+-----+
| 4  | pop            | pasien |           |
+---+-----+-----+-----+
| 20 | popokan        | pasien |           |
+---+-----+-----+-----+
| 15 | risol           | dokter |           |
+---+-----+-----+-----+
| 10 | ropik           | pasien |           |
+---+-----+-----+-----+
| 1  | stewart         | pasien |           |
+---+-----+-----+-----+
| 9  | tobo            | pasien |           |
+---+-----+-----+-----+
| 16 | tobokan         | pasien |           |
+---+-----+-----+-----+
| 7  | tuart           | pasien |           |
+---+-----+-----+-----+
| 88 | zeru            | manager |           |
+---+-----+-----+-----+

```

# Kasus 4: Melihat data user (user bisa berarti dokter atau pasien) urut berdasarkan Nama dan di-sort secara descending

```

>>> LIHAT_USER
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 2
Urutkan Sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 2
===== DATA USER =====
+---+-----+-----+-----+
| ID | Nama        | Role      | Penyakit   |
+---+-----+-----+-----+
| 88 | zero        | manager   |            |
+---+-----+-----+-----+
| 7  | tuart       | pasien    |            |
+---+-----+-----+-----+
| 16 | tobokan     | pasien    |            |
+---+-----+-----+-----+
| 9  | tobo        | pasien    |            |
+---+-----+-----+-----+
| 1  | stewart     | pasien    |            |
+---+-----+-----+-----+
| 10 | ropik        | pasien    |            |
+---+-----+-----+-----+
| 15 | risol        | dokter    |            |
+---+-----+-----+-----+
| 20 | popokan     | pasien    |            |
+---+-----+-----+-----+
| 4  | pop          | pasien    |            |
+---+-----+-----+-----+
| 100| pendaratng  | pasien    |            |
+---+-----+-----+-----+
| 5  | opor         | pasien    |            |
+---+-----+-----+-----+
+---+-----+-----+-----+
| 6  | nukeb        | pasien    |            |
+---+-----+-----+-----+
| 10 | neronimo    | dokter    |            |
+---+-----+-----+-----+
| 8  | minonette   | pasien    |            |
+---+-----+-----+-----+
| 13 | kroket       | dokter    |            |
+---+-----+-----+-----+
| 3  | kebin        | pasien    |            |
+---+-----+-----+-----+
| 2  | gro          | pasien    | COVID-19   |
+---+-----+-----+-----+
| 111| furina       | pasien    | Diabetes Mellitus |
+---+-----+-----+-----+
| 11 | ciciko       | dokter    |            |
+---+-----+-----+-----+
| 12 | cacako       | dokter    |            |
+---+-----+-----+-----+

```

# Kasus 5: Melihat data pasien berdasarkan ID dan di-sort secara ascending

```

>>> LIHAT_PASIEN
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 1
Urutkan sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 1
===== DATA PASIEN =====
+---+-----+-----+
| ID | Nama           | Penyakit      |
+---+-----+-----+
| 1  | stewart       |               |
+---+-----+-----+
| 2  | gro            | COVID-19      |
+---+-----+-----+
| 3  | kebin          |               |
+---+-----+-----+
| 4  | pop            |               |
+---+-----+-----+
| 5  | opor           |               |
+---+-----+-----+
| 6  | nikeb          |               |
+---+-----+-----+
+---+-----+-----+
| 7  | tuart          |               |
+---+-----+-----+
| 8  | minonette      |               |
+---+-----+-----+
| 9  | tobo           |               |
+---+-----+-----+
| 10 | ropik          |               |
+---+-----+-----+
| 16 | tobokan        |               |
+---+-----+-----+
| 20 | popokan        |               |
+---+-----+-----+
| 100 | pendatang     |               |
+---+-----+-----+
| 111 | furina         | Diabetes Mellitus |
+---+-----+-----+

```

# Kasus 6: Melihat data pasien berdasarkan ID dan di-sort secara descending

```

>>> LIHAT_PASIEN
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 1
Urutkan Sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 2
===== DATA PASIEN =====
+-----+-----+
| ID | Nama           | Penyakit      |
+-----+-----+
| 111 | furina        | diabetes Mellitus |
+-----+
| 100 | pendatang     |                   |
+-----+
| 20  | popokan        |                   |
+-----+
| 16  | tobokan        |                   |
+-----+
| 10  | ropik           |                   |
+-----+
| 9   | tobo            |                   |
+-----+
| 8   | minonette       |                   |
+-----+
| 7   | tuart           |                   |
+-----+
+-----+-----+
| 6   | nikeb          |                   |
+-----+
| 5   | opor            |                   |
+-----+
| 4   | pop              |                   |
+-----+
| 3   | kebin           |                   |
+-----+
| 2   | gro              | COVID-19        |
+-----+
| 1   | stewart         |                   |
+-----+

```

# Kasus 7: Melihat data pasien berdasarkan Nama dan di-sort secara ascending

```

>>> LIHAT_PASIEN
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 2
Urutkan Sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 1
===== DATA PASIEN =====
+---+-----+
| ID | Nama           | Penyakit      |
+---+-----+
| 111 | furina        | Diabetes Mellitus |
+---+-----+
| 2   | gro            | COVID-19       |
+---+-----+
| 3   | kebin          |               |
+---+-----+
| 8   | minonette      |               |
+---+-----+
| 6   | nikeb          |               |
+---+-----+
| 5   | opor           |               |
+---+-----+
| 100 | pendatang     |               |
+---+-----+
| 4   | pop            |               |
+---+-----+
+---+-----+
| 20  | popokan        |               |
+---+-----+
| 10  | ropik          |               |
+---+-----+
| 1   | stewart        |               |
+---+-----+
| 9   | tobo           |               |
+---+-----+
| 16  | tobokan        |               |
+---+-----+
| 7   | tuart          |               |
+---+-----+

```

# Kasus 8: Melihat data pasien berdasarkan Nama dan di-sort secara descending

```

>>> LIHAT_PASIEN
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 2
Urutkan Sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 2
===== DATA PASIEN =====
+---+-----+-----+
| ID | Nama           | Penyakit      |
+---+-----+-----+
| 7  | tuart          |              |
+---+-----+-----+
| 16 | tobokan        |              |
+---+-----+-----+
| 9  | tobo           |              |
+---+-----+-----+
| 1   | stewart         |              |
+---+-----+-----+
| 10 | ropik           |              |
+---+-----+-----+
| 20 | popokan         |              |
+---+-----+-----+
| 4   | pop             |              |
+---+-----+-----+
| 100 | pendatang       |              |
+---+-----+-----+
| 5   | opor            |              |
+---+-----+-----+
+---+-----+-----+
| 6   | nikeb           |              |
+---+-----+-----+
| 8   | minonette        |              |
+---+-----+-----+
| 3   | kebin            |              |
+---+-----+-----+
| 2   | gro              | COVID-19      |
+---+-----+-----+
| 111 | furina          | Diabetes Mellitus |
+---+-----+-----+

```

# Kasus 9: Melihat data dokter berdasarkan ID dan di-sort secara ascending

```
>>> LIHAT_DOKTER
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 1
Urutkan Sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 1
===== DATA DOKTER =====
+---+-----+
| ID | Nama |
+---+-----+
| 10 | neronimo |
+---+-----+
| 11 | ciciko |
+---+-----+
| 12 | cacako |
+---+-----+
| 13 | kroket |
+---+-----+
| 15 | risol |
+---+-----+
```

# Kasus 10: Melihat data dokter berdasarkan ID dan di-sort secara descending

```
>>> LIHAT_DOKTER
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 1
Urutkan Sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 2
===== DATA DOKTER =====
+---+-----+
| ID | Nama |
+---+-----+
| 15 | risol |
+---+-----+
| 13 | kroket |
+---+-----+
| 12 | cacako |
+---+-----+
| 11 | ciciko |
+---+-----+
| 10 | neronimo |
+---+-----+
```

# Kasus 11: Melihat data dokter berdasarkan Nama dan di-sort secara ascending

```
>>> LIHAT_DOKTER
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 2
Urutkan Sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 1
===== DATA DOKTER =====
+---+-----+
| ID | Nama |
+---+-----+
| 12 | cacako |
+---+-----+
| 11 | ciciko |
+---+-----+
| 13 | kroket |
+---+-----+
| 10 | neronimo |
+---+-----+
| 15 | risol |
+---+-----+
```

# Kasus 12: Melihat data dokter berdasarkan Nama dan di-sort secara

```

>>> LIHAT_DOKTER
Urutkan berdasarkan? (1/2)
    1. ID
    2. Nama
>>> 2
Urutkan Sort? (1/2)
    1. ASC (A-Z)
    2. DESC (Z-A)
>>> 2
===== DATA DOKTER =====
+---+-----+
| ID | Nama |
+---+-----+
| 15 | risol |
+---+-----+
| 10 | neronimo |
+---+-----+
| 13 | kroket |
+---+-----+
| 11 | ciciko |
+---+-----+
| 12 | cacako |
+---+-----+

```

#### F08 - Cari User

# Kasus 1 : Cari user berdasarkan ID dan user ditemukan

```

>>> CARI_USER
Cari berdasarkan:
1. ID
2. Nama
Masukkan angka pilihan:
>>> 1
Cari ID: 111
Pengguna dengan ID 111:
+---+-----+-----+-----+
| ID | Nama       | Role      | Penyakit   |
+---+-----+-----+-----+
| 111 | furina     | pasien    | Diabetes Mellitus |
+---+-----+-----+-----+

```

```
>>> CARI_USER  
Cari berdasarkan:  
1. ID  
2. Nama  
Masukkan angka pilihan:  
>>> 1  
Cari ID: 101  
ID user tidak ditemukan!
```

# Kasus 2 : Cari user berdasarkan nama dan user ditemukan

```
Masukkan perintah (Ketik [HELP] untuk melihat daftar perintah):  
>>> CARI_USER  
Cari berdasarkan:  
1. ID  
2. Nama  
Masukkan angka pilihan:  
>>> 2  
Cari Username: furina  
Menampilkan pasien dengan nama furina:  
+-----+-----+-----+-----+  
| ID | Nama | Role | Penyakit |  
+-----+-----+-----+-----+  
| 111 | furina | pasien | Diabetes Mellitus |  
+-----+-----+-----+-----+
```

# Kasus 3 : Cari user berdasarkan ID dan user ditemukan

```
Masukkan perintah (Ketik [HELP] untuk melihat daftar perintah):  
>>> CARI_USER  
Cari berdasarkan:  
1. ID  
2. Nama  
Masukkan angka pilihan:  
>>> 2  
Cari Username: masfuad  
User dengan username 'masfuad' tidak ditemukan.
```

# Kasus 4 : Cari user berdasarkan ID namun user tidak ditemukan

```
>>> CARI_USER  
Cari berdasarkan:  
1. ID  
2. Nama  
Masukkan angka pilihan:  
>>> 1  
Cari ID: 101  
User dengan ID '101' tidak ditemukan.
```

# Kasus 5 : Cari pasien, namun input pilihan tidak valid (berlaku untuk seluruh kasus pencarian)

```
>>> CARI_PASIEN  
Cari berdasarkan:  
1. ID  
2. Nama  
3. Penyakit  
Masukkan angka pilihan:  
>>> 8  
Pilihan tidak valid. Mohon masukkan pilihan yang benar!.
```

# Kasus 6 : Cari pasien berdasarkan ID dan pasien ditemukan

```
>>> CARI_PASIEN  
Cari berdasarkan:  
1. ID  
2. Nama  
3. Penyakit  
Masukkan angka pilihan:  
>>> 1  
Cari ID: 111  
Pengguna dengan ID 111:  
+-----+-----+-----+  
| ID   | Nama        | Role      | Penyakit    |  
+-----+-----+-----+  
| 111  | furina     | pasien    | Diabetes Mellitus |  
+-----+-----+-----+
```

# Kasus 7 : Cari pasien berdasarkan, namun pasien tidak ditemukan (bisa tidak terdaftar atau user dengan ID tersebut bukan seorang pasien. Dalam kasus ini adalah dokter)

```
>>> CARI_PASIEN
Cari pasien berdasarkan:
1. ID
2. Nama
3. Penyakit
Masukkan angka pilihan:
>>> 1
Cari ID: 10
Pasien dengan ID '10' tidak ditemukan atau bukan pasien.
```

# Kasus 8 : Cari pasien berdasarkan penyakit dan pasien ditemukan

```
>>> CARI_PASIEN
Cari berdasarkan:
1. ID
2. Nama
3. Penyakit
Masukkan angka pilihan:
>>> 3
Cari penyakit: Diabetes Mellitus
+-----+
| ID | Nama           | Penyakit      |
+-----+
| 4  | pop            | Diabetes Mellitus |
+-----+
| 111 | furina        | Diabetes Mellitus |
+-----+
```

# Kasus 9 : Cari pasien berdasarkan penyakit namun penyakit tidak ditemukan

```
>>> CARI_PASIEN
Cari berdasarkan:
1. ID
2. Nama
3. Penyakit
Masukkan angka pilihan:
>>> 3
Cari penyakit: Demam Berdarah
Pasien dengan penyakit Demam Berdarah tidak ditemukan
```

# Kasus 10 : Cari pasien berdasarkan nama dan pasien tidak ditemukan

```
>>> CARI_PASIEN
Cari pasien berdasarkan:
1. ID
2. Nama
3. Penyakit
Masukkan angka pilihan:
>>> 2
Cari Username: kita ikuyo
Mencari pasien dengan username 'kita ikuyo':
Pasien dengan username 'kita ikuyo' tidak ditemukan.
```

# Kasus 11 : Cari dokter berdasarkan ID dan dokter ditemukan

```
>>> CARI_DOKTER
Cari dokter berdasarkan:
1. ID
2. Nama
Masukkan angka pilihan:
>>> 1
Cari ID: 10
Dokter dengan ID 10 ditemukan:
+---+-----+
| ID | Nama      |
+---+-----+
| 10 | neronimo |
+---+-----+
```

# Kasus 12 : Cari dokter berdasarkan ID dan dokter tidak ditemukan

```
Masukkan perintah (Ketik [HELP] untuk melihat daftar perintah):
>>> CARI_DOKTER
Cari dokter berdasarkan:
1. ID
2. Nama
Masukkan angka pilihan:
>>> 1
Cari ID: 111
Dokter dengan ID '111' tidak ditemukan atau bukan dokter.
```

# Kasus 13 : Cari dokter berdasarkan nama dan dokter ditemukan

```
>>> 2
Cari Username: risol
Mencari dokter dengan username 'risol':
+---+-----+
| ID | Nama      |
+---+-----+
| 15 | risol      |
+---+-----+
```

# Kasus 14 : Cari dokter berdasarkan nama dan dokter tidak ditemukan

```
>>> CARI_DOKTER
Cari dokter berdasarkan:
1. ID
2. Nama
Masukkan angka pilihan:
>>> 2
Cari Username: DOKTER
Mencari dokter dengan username 'DOKTER':
Dokter dengan username 'DOKTER' tidak ditemukan.
```

F09 / D02 - Lihat Antrian

# Kasus 1: Lihat semua antrian

```

>>> LIHAT_SEMUA_ANTRIAN
      1   2   3   4   5
+---+---+---+---+---+
A | A1 | A2 | A3 | A4 | A5 |
+---+---+---+---+---+
B | B1 | B2 | B3 | B4 | B5 |
+---+---+---+---+---+
C | C1 | C2 | C3 | C4 | C5 |
+---+---+---+---+---+
D | D1 | D2 | D3 | D4 | D5 |
+---+---+---+---+---+
===== A1 =====
Kapasitas : 3
Dokter   : Dr. neronimo
Pasien di dalam ruangan:
 1. gro
 2. kebin
 3. stewart
Pasien di antrian:
 1. tobokan
 2. popokan

===== A2 =====
Kapasitas : 3
Dokter   : Dr. ciciko
Pasien di dalam ruangan:
 1. pop
 2. opor
Pasien di antrian:
 Tidak ada pasien di antrian saat ini.

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

===== B1 =====
Kapasitas : 3
Dokter   : Dr. kroket
Pasien di dalam ruangan:
 1. minonette
 2. tuart
Pasien di antrian:
 Tidak ada pasien di antrian saat ini.

===== B2 =====
Kapasitas : 3
Dokter   : Dr. doktzyheker
Pasien di dalam ruangan:
 Tidak ada pasien di dalam ruangan saat ini.
Pasien di antrian:
 Tidak ada pasien di antrian saat ini.

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

===== B4 =====
Kapasitas : 3
Dokter   : Dr. tuandokter
Pasien di dalam ruangan:
 Tidak ada pasien di dalam ruangan saat ini.
Pasien di antrian:
 Tidak ada pasien di antrian saat ini.

```

```

===== B1 =====
Kapasitas : 3
Dokter   : Dr. kroket
Pasien di dalam ruangan:
 1. minonette
 2. tuart
Pasien di antrian:
 Tidak ada pasien di antrian saat ini.

===== B2 =====
Kapasitas : 3
Dokter   : Dr. doktzyheker
Pasien di dalam ruangan:
 Tidak ada pasien di dalam ruangan saat ini.
Pasien di antrian:
 Tidak ada pasien di antrian saat ini.

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

===== B4 =====
Kapasitas : 3
Dokter   : Dr. tuandokter
Pasien di dalam ruangan:
 Tidak ada pasien di dalam ruangan saat ini.
Pasien di antrian:
 Tidak ada pasien di antrian saat ini.

```

# Kasus 1 : Tambah dokter dengan username yang tidak valid

```
>>> TAMBAH_DOKTER
Masukkan username dokter baru: doktzyhengker123
Username hanya boleh terdiri dari huruf!
```

# Kasus 2 : Tambah dokter dengan username yang valid

```
Masukkan perintah (Ketik [HELP] untuk melihat daftar perintah):
>>> TAMBAH_DOKTER
Masukkan username dokter baru: doktzyheker
Masukkan password dokter baru: doktzyheker123
Dokter doktzyheker berhasil ditambahkan dengan ID 112.
```

```
>>> LOGIN
Username: doktzyheker
Password: doktzyheker123
Login Berhasil! Selamat datang dokter doktzyheker.
```

# Kasus 3 : Assign Dokter ke ruangan yang sudah terdapat dokter

```
>>> ASSIGN_DOKTER
Masukkan username dokter yang akan diassign: doktzyheker
Masukkan kode ruangan (misal A1, B3, dst): A1
Ruang A1 sudah diisi oleh dokter neronimo.
```

# Kasus 4 : Assign Dokter ke ruangan yang kosong

```
>>> ASSIGN_DOKTER
Masukkan username dokter yang akan diassign: doktzyheker
Masukkan kode ruangan (misal A1, B3, dst): B2
Dokter doktzyheker berhasil diassign ke ruangan B2.
```

# Kasus 5 : Assign Dokter yang sudah memiliki ruangan

```
>>> ASSIGN_DOKTER
```

Masukkan username dokter yang akan diassign: doktzyheker  
Dokter sudah diassign ke ruangan B2.

# Kasus 6 : Assign Dokter ke ruangan yang tidak valid

```
>>> ASSIGN_DOKTER
```

Masukkan username dokter yang akan diassign: tuandokter  
Masukkan kode ruangan (misal A1, B3, dst): B4  
Kolom tidak valid. Harus antara 1 hingga 3.

# Kasus 7 : Assign Dokter ke denah yang telah diubah (sebelumnya ruangan tidak valid, namun setelah diubah ruangan menjadi valid)

Masukkan perintah (Ketik [HELP] untuk melihat daftar perintah):

```
>>> UBAH_DENAH
```

Masukkan ukuran denah yang baru: 4 5

Denah rumah sakit berhasil diubah menjadi 4 baris dan 5 kolom.

Masukkan perintah (Ketik [HELP] untuk melihat daftar perintah):

```
>>> LIHAT_DENAH
```

	1	2	3	4	5
A	A1   A2   A3   A4   A5				
B	B1   B2   B3   B4   B5				
C	C1   C2   C3   C4   C5				
D	D1   D2   D3   D4   D5				

Masukkan perintah (Ketik [HELP] untuk melihat daftar perintah):

```
>>> ASSIGN_DOKTER
```

Masukkan username dokter yang akan diassign: tuandokter

Masukkan kode ruangan (misal A1, B3, dst): B4

Dokter tuandokter berhasil diassign ke ruangan B4.

## F11 - Diagnosis

# Kasus 1 : Dokter memiliki pasien yang perlu diperiksa

```
>>> DIAGNOSIS
```

pendatang terdiagnosa penyakit Influenza!

# Kasus 2 : Antrian pasien sudah kosong dan tidak ada pasien yang perlu diperiksa

>>> DIAGNOSIS

Tidak ada pasien untuk diperiksa!

# Kasus 3: Pasien tidak terjangkit penyakit apapun setelah diperiksa. Jika terdapat kasus ini maka pasien diperbolehkan pulang.

>>> DIAGNOSIS

pendatang tidak terdiagnosis penyakit apapun!

# Kasus 4: Pasien telah didiagnosis sebelumnya

>>> DIAGNOSIS

Pasien telah terdiagnosa, arahkan pasien menuju ke Ngobatin.

## F12 - Ngobatin

# Kasus 1: Dokter belum ditempatkan di ruangan mana pun

>>> NGOBATIN

Dokternya belum ditempatkan di ruangan mana pun. Tolong assign dulu ya!

# Kasus 2: Dokter tidak memiliki pasien dalam antrian

>>> NGOBATIN

Belum ada pasien nih di antrian ruangan dokter 'risol'. Santai dulu dok!

# Kasus 3: Pasien belum punya diagnosis penyakit

>>> NGOBATIN

Pasien ini belum ada diagnosis penyakit, dok. Kasih diagnosis dulu ya biar bisa dikasih obat yang tepat.

# Kasus 4: Berhasil memberikan obat ke pasien

>>> NGOBATIN

Pasien 'pop' terdiagnosa menderita Diabetes Mellitus.

Berikut daftar obat yang harus diberikan, urutannya jangan sampai kebalik ya!

>> Obat ke-1: Metformin  
>> Obat ke-2: Lisinopril  
>> Obat ke-3: Remdesivir  
>> Obat ke-4: Vitamin C

Pastikan obat diberikan satu per satu sesuai urutan di atas ya. Semangat menyembuhkan pasien!

## F13 - Aku Boleh Pulang ga, Dok?

# Kasus 1: Pasien belum diberikan diagnosa penyakit oleh dokter

>>> DOK\_AKU\_MAU\_PULANG

Kamu belum menerima diagnosis apapun dari dokter, jangan buru-buru pulang! T^T

# Kasus 2: Pasien belum menghabiskan seluruh obat yang diberikan kepadanya

>>> DOK\_AKU\_MAU\_PULANG

Dokter sedang memeriksa keadaanmu...

Masih ada obat yang belum kamu habiskan, minum semuanya dulu yukk! :3

# Kasus 3: Pasien sudah menghabiskan obat, namun terdapat urutan yang salah dalam konsumsinya

>>> DOK\_AKU\_MAU\_PULANG

Dokter sedang memeriksa keadaanmu...

Maaf, tapi kamu masih belum bisa pulang T^T !

Urutan MINUM\_OBAT yang diharapkan:

Metformin -> Lisinopril -> Remdesivir -> Vitamin C

Urutan MINUM\_OBAT yang kamu minum:

Lisinopril-> Metformin -> Remdesivir -> Vitamin C

Silahkan kunjungi dokter untuk meminta penawar (MINUM\_PENAWAR) yang sesuai!

# Kasus 4: Pasien sudah menghabiskan obat, dan semuanya valid.

>>> DOK\_AKU\_MAU\_PULANG

Dokter sedang memeriksa keadaanmu...

SELAMAT!! :D

Kamu sudah dinyatakan sembuh oleh dokter. Silahkan pulang dan semoga sehat selalu! :3

# Kasus 5: Pasien belum ada di antrian paling depan

>>> DOK\_AKU\_MAU\_PULANG

Kamu belum berada di urutan terdepan antrian mana pun. Sabar dulu ya! :(

# Kasus 6: Pasien tidak ditemukan di antrian / ruangan mana pun

>>> DOK\_AKU\_MAU\_PULANG

Pasien tidak ditemukan. Silahkan melakukan registrasi terlebih dahulu :)

# Kasus 7: Pasien yang telah terdiagnosis “Sehat” / tidak ada penyakit apa pun

>>> DOK\_AKU\_MAU\_PULANG

Kamu tidak ada penyakit apapun. Silahkan pulang dan semoga sehat selalu! :3

## F14 - Daftar Check-Up

# Kasus 1: Pasien sudah terdaftar di antrian

```
>>> 2
```

Eh, kamu udah daftar buat check-up nih!  
Yuk, tunggu giliran dulu. Nanti aja daftarnya lagi.

# Kasus 2: Berhasil mendaftar dan memilih dokter

```
>>> 2
```

--- FORMULIR CHECK-UP PASIEN ---

Masukin data kesehatan kamu dulu ya. Jangan asal, ini penting!

Suhu Tubuh (°C): 37.5

Tekanan Darah (sistol dan diastol): 80 120

Detak Jantung (bpm): 60

Saturasi Oksigen (%): 5

Gula Darah (mg/dL): 100

Berat Badan (kg): 55

Tinggi Badan (cm): 165

Kolesterol Total (mg/dL): 5

Trombosit (ribu/µL): 5

--- DOKTER YANG SIAP NANGGEPIN KAMU ---

1. Dr. neronimo - Ruangan A1 (Antrian: 5 pasien)

2. Dr. ciciko - Ruangan A2 (Antrian: 2 pasien)

3. Dr. cacako - Ruangan A3 (Antrian: 1 pasien)

4. Dr. kroket - Ruangan B1 (Antrian: 2 pasien)

5. Dr. risol - Ruangan B3 (Antrian: 0 pasien)

Pilih dokter (1 - 5): 3

Mantap! Check-up kamu udah masuk daftar.

Antrian kamu sekarang di Dr. cacako - Ruangan A3

Nomor urut kamu: 2

## F15 - Antrian Saya!

# Kasus 1: Pasien belum ada di antrian mana pun

```
>>> ANTRIAN_SAYA
```

Eits, kamu belum ada di antrian mana pun nih!  
Yuk daftar dulu buat check-up lewat command: DAFTAR\_CHECKUP

# Kasus 2: Pasien sedang menunggu di luar ruangan

```
>>> ANTRIAN_SAYA
```

==== STATUS ANTRIAN KAMU ====

Ketemu! Kamu lagi nunggu di ruangan A1 bareng Dokter neronimo.  
Posisimu sekarang di antrian: 6 dari 6 pasien.  
Kamu masih DI LUAR ruangan. Tunggu 3 pasien lagi buat masuk, ya!

# Kasus 3: Pasien sudah di dalam ruangan

```
>>> ANTRIAN_SAYA
```

==== STATUS ANTRIAN KAMU ====

Ketemu! Kamu lagi nunggu di ruangan A3 bareng Dokter cacako.  
Posisimu sekarang di antrian: 2 dari 2 pasien.  
Kamu udah DI DALAM ruangan. Siap-siap ya, giliranmu bentar lagi!

## F16 - Minum Obat

# Kasus 1: Inventory obat pasien kosong

```
>>> MINUM_OBAT
```

Eh, obatmu kosong loh! Gak ada yang bisa diminum sekarang :)

# Kasus 2: Pasien memilih nomor obat yang tidak valid

```
>>> MINUM_OBAT
```

===== DAFTAR OBAT =====

1. Remdesivir

Pilih nomor obat yang mau kamu minum (1-1): 2

Waduh! Nomor yang kamu pilih gak ada di daftar!

# Kasus 3: Pasien berhasil minum obat

```
>>> 4
===== DAFTAR OBAT =====
1. Remdesivir

Pilih nomor obat yang mau kamu minum (1-1): 1
*GLEKGLEKGLEK*... Remdesivir berhasil diminum! Rasanya... unik ya^^
Inventory diperbarui. Obat Remdesivir udah keluar dari kantong ajaib kamu!
```

## F17 - Minum Penawar

# Kasus 1: Obat terakhir yang diminum adalah Lisinopril

```
>>> MINUM_PENAWAR
Uwekk :o !!!
Inventory diperbarui. Obat Vitamin C keluar dan kembali ke kantong ajaib kamu!
```

# Kasus 2: Belum ada obat yang diminum

```
>>> MINUM_PENAWAR
Perut kosong!! Belum ada obat yang dimakan. >:)
```

# Kasus 3: Pasien melakukan MINUM\_PENAWAR tetapi belum terdiagnosis / belum mendapatkan obat

```
>>> MINUM_PENAWAR
Kamu belum menerima obat dari dokter. Tidak boleh meminta obat penawar!! >:)
```

## F18 - EXIT:

1. #Kasus input valid

```
[rerre@LAPTOP-MH6CM0J4] - [~/if1210-tubes-2025-k01-1]
$ ./tes
Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): ya
Masukkan nama folder untuk menyimpan data: Ruru
Folder "Ruru" belum ada. Membuat folder baru...
Berhasil menyimpan data di folder Ruru.
Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.
```

```
[rerre@LAPTOP-MH6CM0J4] - [~/if1210-tubes-2025-k01-1]
$ ./tes
Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): tidak
Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.
```

2. #Kasus input tidak valid

```
[rerre@LAPTOP-MH6CM0J4] - [~/if1210-tubes-2025-k01-1]
$ ./tes
Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): PGPGPGPGPGP
Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.
```

```
(erre@LAPTOP-MH6CM034) [~/if1210-tubes-2025-k01-1]
$ ./tes
Konfigurasi berhasil dimuat dari data/config.txt
Data berhasil diload dari folder: data
Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): AAAAAAAAAAAAAA
Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.
Segmentation fault (core dumped)
```

## D03 - Load

# Kasus 1 : Mulai menjalankan program tanpa nama folder database

```
/if1210-tubes-2025-k01-1$ ./main
Usage: ./main <nama_folder>
```

# Kasus 2 : Mulai menjalankan program dengan nama folder database yang valid

```
/if1210-tubes-2025-k01-1$ ./main data
Loading...
Konfigurasi berhasil dimuat dari data/config.txt
✓ Data berhasil diload dari folder: data
Data berhasil dimuat dari folder: data

+=====+
| SILAKAN LOGIN DENGAN MENGETIK [LOGIN] ATAU |
| MELAKUKAN REGISTER DENGAN MENGETIK [REGISTER] |
+=====+
```

# Kasus 3 : Mulai menjalankan program namun nama folder tidak terbaca/corrupt/tidak terdapat di direktori

```
/if1210-tubes-2025-k01-1$ ./main foldertest
Loading...
Fatal error: Gagal membuka file foldertest/user.csv
```

## D04 - SAVE:

#Kasus 1 folder belum tersedia

```
(erre@LAPTOP-MH6CM034) [~/if1210-tubes-2025-k01-1]
$ ./tes
Konfigurasi berhasil dimuat dari data/config.txt
Data berhasil diload dari folder: data
Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): ya
Masukkan nama folder untuk menyimpan data: ruru
Folder "ruru" belum ada. Membuat folder baru...
Berhasil menyimpan data di folder ruru.
Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.
```

Folder input (data file csv semula):

Folder output (nama folder baru: ruru)

#Kasus 2 folder sudah tersedia, baik ada isinya atau tidak (kosong)

Folder output (nama folder lama: ruru):

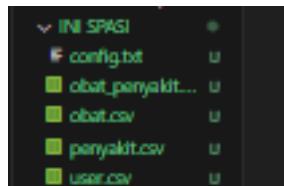
```
[rerre@LAPTOP-MH6CMQJ4] - [~/if1210-tubes-2025-k01-1]
$ ./tes
Konfigurasi berhasil dimuat dari data/config.txt
Data berhasil diload dari folder: data
Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): ya
Masukkan nama folder untuk menyimpan data: ruru
Berhasil menyimpan data di folder ruru.
Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.
```

#Kasus 3 nama folder sangat panjang

```
[revri@LAPTOP-84C9D1A ~]$ ./vfi210-tubes-2025-kel1  
Konfigurasi berhasil diambil dari data/config.txt  
Data berhasil diambil dari folder: data  
Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): ya  
Folder "AAAAAAA" berhasil dibuat.  
Folder "AAAAAAA" berhasil dibuat.  
Berhasil menyimpan data di folder AAAAAAA.  
Program dihentikan. Semoga dokter, pasien, danawejer tetap sehat dan aman.
```

#Kasus 4 nama folder terdapat spasi

```
(rerre@LAPTOP-MH6CM034)-[~/if1210-tubes-2025-k01-1]
$ ./tes
Konfigurasi berhasil dimuat dari data/config.txt
Data berhasil diload dari folder: data
Apakah Anda ingin menyimpan data sebelum keluar? (ya/tidak): ya
Masukkan nama folder untuk menyimpan data:INI SPASI
Folder "INI SPASI" belum ada. Membuat folder baru...
Berhasil menyimpan data di folder INI SPASI.
Program dihentikan. Semua dokter, pasien, dan manajer telah meninggalkan rumah sakit.
```



B02 - Denah Dinamis

# Kasus 1 : Ukuran denah yang baru valid

```
>>> UBAH_DENAH  
Masukkan ukuran denah yang baru: 3 3  
Denah rumah sakit berhasil diubah menjadi 3 baris dan 3 kolom.
```

# Kasus 2 : Ukuran denah yang baru tidak valid

```
>>> UBAH_DENAH  
Masukkan ukuran denah yang baru: 2 2  
Tidak dapat mengubah ukuran denah. Ruangan A3 masih ditempati oleh Dr. cacako. Silakan pindahkan dokter terlebih dahulu.
```

# Kasus 1 : Ruangan Tujuan dokter pindah kosong

```
>>> PINDAH_DOKTER  
Masukkan ruangan asal dan tujuan (misal: A1 B2): A1 B2  
Dr. neronimo berhasil dipindahkan dari ruangan A1 ke ruangan B2.
```

# Kasus 2 : Ruangan tujuan dokter pindah sudah ditempati dokter lain

```
>>> PINDAH_DOKTER  
Masukkan ruangan asal dan tujuan (misal: A1 B2): A1 A2  
Pemindahan gagal. Ruangan A2 Sudah ditempati.
```

# Kasus 3 : Ruangan awal dokter pindah kosong

```
>>> PINDAH_DOKTER  
Masukkan ruangan asal dan tujuan (misal: A1 B2): B2 A1  
Pemindahan gagal. Ruangan B2 Kosong.
```

# Kasus 4 : Ruangan tidak terdefinisi (tidak ada)

```
>>> PINDAH_DOKTER  
Masukkan ruangan asal dan tujuan (misal: A1 B2): A3 B4  
Pemindahan gagal. Ruangan B4 Tidak Ada.
```

## BAB X

### LAMPIRAN

Nomor Asistensi	:	1
No. Kelompok/Kelas	:	L/01
Tanggal asistensi	:	Jumat, 9 Mei 2025

Anggota kelompok		NIM / Nama (Hanya yang Hadir)
1		13524055 / Junior Natra Situmorang
2		13524071 / Kalyca Nathania Benedicta Manullang
3		13524077 / Yavie Azka Putra Araly
4		13524097 / Muhammad Faiz Alfada Dharmo
5		13524117 / Rainaldi Pratama F. Sembiring
6		13524137 / Reysha Syafitri Mulya Ramadhan
Asisten pembimbing		NIM / Nama
		13522011 / Dewantoro Triatmojo

Catatan Asistensi:

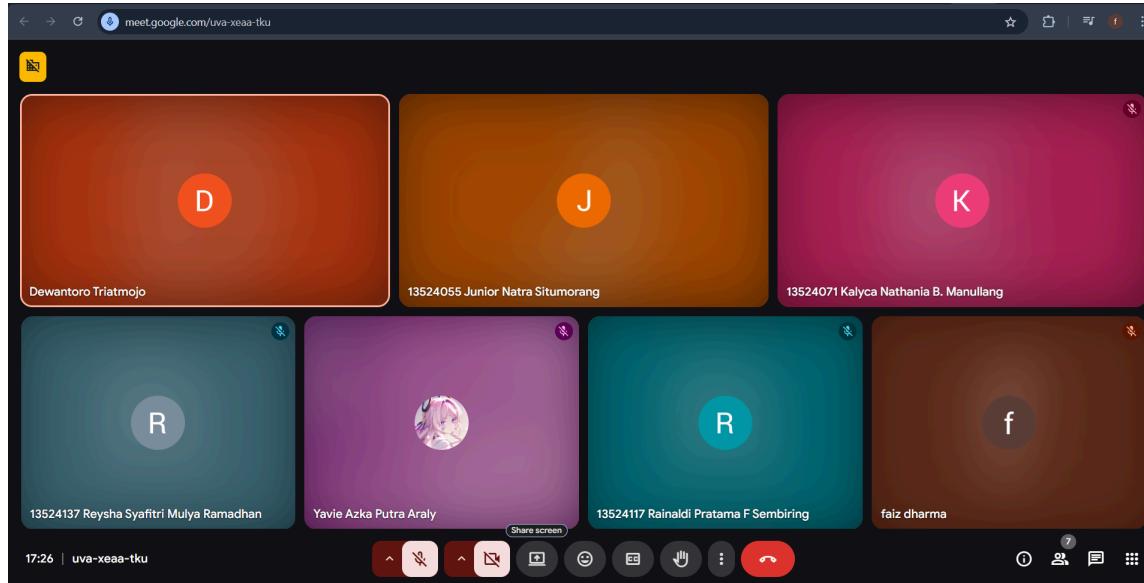
Rangkuman Diskusi
<ol style="list-style-type: none"><li>Penulis disarankan untuk melakukan <i>merge</i> modul atau kode yang sudah selesai secara berkala agar tidak kesulitan saat menyatukan program utama di tahap akhir.</li><li>Saat ini pengembangan masih dilakukan di Windows. Penulis disarankan menggunakan WSL (Windows Subsystem for Linux) karena beberapa fungsi di C memiliki perbedaan implementasi di Linux dan Windows. WSL digunakan sebagai lingkungan pengembangan yang mendekati sistem target (Linux) sehingga lebih aman untuk <i>compile</i> dan menjalankan program.</li><li>Untuk struktur program, saat ini fitur-fitur masih berdiri sendiri dan belum disatukan dalam program utama. Penulis ditekankan pentingnya manajemen <i>state</i> program agar lebih terstruktur dan dapat dikembangkan bersama secara efisien.</li><li>Penulis disarankan menggunakan pendekatan <i>pass by reference</i> ke fungsi dengan variabel global atau <i>struct</i> sebagai media penyimpanan <i>state</i>. Untuk manajemen <i>state</i> program penggunaan <i>global state</i> memudahkan proses pengujian dan pengembangan karena lebih terprediksi.</li></ol>

5. Untuk strategi pengelolaan variabel, penulis disarankan membuat satu *struct* global berisikan seluruh variabel penting (misalnya daftar *user*, dokter, dan lain-lain). *Struct* ini dapat diparsing ke fungsi-fungsi yang membutuhkan tanpa harus memanggil banyak variabel satu per satu. Alternatif lain adalah menggunakan pendekatan *extern*, tetapi hal ini memerlukan kehati-hatian dalam penulisan dan implementasi.

#### Tindak Lanjut

1. Mulai menggunakan WSL untuk proses pengembangan dan kompilasi program.
2. Segera menyatukan fitur-fitur ke dalam program utama secara bertahap dan rutin melakukan *merge*.
3. Membuat satu *struct* global yang berisi semua variabel penting (*global state*).
4. Menerapkan manajemen *state* menggunakan pendekatan *pass by reference* terhadap *struct* tersebut ke fungsi-fungsi terkait.
5. Membagi tugas pengembangan agar terkoordinasi dan terintegrasi dengan manajemen *state* yang jelas.

#### Dokumentasi



Gambar 10.1 Asistensi 1

<b>Nomor Asistensi</b>	:	<b>2</b>
<b>No. Kelompok/Kelas</b>	:	<b>L/01</b>
<b>Tanggal asistensi</b>	:	<b>Jumat, 23 Mei 2025</b>

<b>Anggota kelompok</b>	<b>NIM / Nama (Hanya yang Hadir)</b>
1	<b>13524055 / Junior Natra Situmorang</b>
2	<b>13524071 / Kalyca Nathania Benedicta Manullang</b>
3	<b>13524077 / Yavie Azka Putra Araly</b>
4	<b>13524097 / Muhammad Faiz Alfada Dharma</b>
5	<b>13524137 / Reysha Syafitri Mulya Ramadhan</b>
<b>Asisten pembimbing</b>	<b>NIM / Nama</b>
	<b>13522011 / Dewantoro Triatmojo</b>

**Catatan Asistensi:**

<b>Rangkuman Diskusi</b>
<ul style="list-style-type: none"> <li>- Data dari file hanya akan dibaca saat proses load dan disimpan kembali saat save. Semua modifikasi terhadap data dilakukan langsung pada struktur data yang berada di memori.</li> <li>- Asisten menanyakan perkembangan pengerajan tugas besar. Fitur F01 hingga F10 telah berhasil diintegrasikan ke dalam program utama.</li> <li>- Program sudah dijalankan melalui WSL (Windows Subsystem for Linux), bukan lagi melalui terminal bawaan Windows.</li> <li>- Desain Kamus Data pada laporan harus menggunakan notasi algoritmik, bukan sintaks bahasa C.</li> </ul>

- Desain dekomposisi algoritmik dan fungsional harus disajikan dalam bentuk flowchart pada laporan.
- Disarankan untuk tidak menunda pekerjaan mendekati tenggat waktu karena hal-hal tak terduga sering terjadi di akhir masa penggeraan.

#### Tindak Lanjut

- Menyelesaikan dan memastikan seluruh fitur berjalan sesuai spesifikasi sebelum tanggal 31 Mei 2025.
- Merapikan dan melengkapi laporan.
- Melakukan pengujian menyeluruh di lingkungan WSL agar tidak ada kendala saat demo.
- Memastikan tidak ada perubahan kode setelah tahap finalisasi agar siap untuk proses demo.

#### Dokumentasi



Gambar 10.2 Asistensi 2