

**LAPORAN TUGAS BESAR IF1210
ALGORITMA DAN PEMROGRAMAN 1**



Sistem Pengelolaan Rumah Sakit Berbasis C dan CLI

Disusun oleh:
Kelompok G/K-03

Naomi Azzahra	18224017
Muthia Ariesta Anggraeni	18224045
Nathan Pasha Athallah	18224053
Riantama Putra	18224061
Stephanie Mae	18224117

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

PERNYATAAN KEJUJURAN

Dengan ini, kami:

Nama dan NIM	: 1. Naomi Azzahra (18224017) 2. Muthia Ariesta Anggraeni (18224045) 3. Nathan Pasha Athallah (18224053) 4. Riantama Putra (18224061) 5. Stephanie Mae (18224117)
Fakultas/Sekolah	: Sekolah Teknik Elektro dan Informatika
Kode Mata Kuliah	: IF1210
Nama Mata Kuliah	: Algoritma dan Pemrograman 1
Tema Proyek	: Sistem Pengelolaan Rumah Sakit Berbasis C

menyatakan bahwa kami mengerjakan tugas besar ini dengan sejujur-jujurnya, tanpa menggunakan cara yang tidak dibenarkan. Apabila di kemudian hari diketahui kami mengerjakan tugas besar ini dengan cara yang tidak jujur, kami bersedia mendapatkan konsekuensinya, yaitu mendapatkan nilai E pada mata kuliah IF1210 Algoritma dan Pemrograman 1 Semester 2 2024/2025.

DAFTAR ISI

PERNYATAAN KEJUJURAN.....	2
DAFTAR ISI.....	3
DAFTAR TABEL.....	4
DAFTAR GAMBAR.....	5
BAB I.....	6
PENDAHULUAN.....	6
1.1. Deskripsi Permasalahan.....	6
1.2. Tujuan.....	6
BAB II.....	7
PERANCANGAN PROGRAM.....	7
2.1. Rencana Implementasi Fitur.....	7
2.2. Desain Program.....	8
BAB III.....	90
IMPLEMENTASI DAN PENGUJIAN.....	90
3.1. Checklist Progres Fitur.....	90
3.2. Pembagian Tugas Anggota.....	91
LAMPIRAN.....	97

DAFTAR TABEL

Tabel 2.1.1 Rencana Implementasi.....	7
Tabel 2.2.1.1 Rincian Command.....	9
Tabel 3.1.1 Progres fitur.....	90
Tabel 3.2.1 Pembagian tugas.....	91

DAFTAR GAMBAR

Gambar 2.2.2.1 Flowchart Login.....	14
Gambar 2.2.2.2 Flowchart Register.....	17
Gambar 2.2.2.3 Flowchart Logout.....	20
Gambar 2.2.2.4 Flowchart Lupa Password.....	21
Gambar 2.2.2.5 Flowchart Menu Help.....	24
Gambar 2.2.2.6 Flowchart Lihat Denah.....	28
Gambar 2.2.2.7 Flowchart Lihat Ruangan.....	29
Gambar 2.2.2.8 Flowchart Lihat User.....	35
Gambar 2.2.2.9 Flowchart Cari User.....	44
Gambar 2.2.2.10 Flowchart Tambah Dokter.....	56
Gambar 2.2.2.11 Flowchart Assign Dokter.....	56
Gambar 2.2.2.12 Flowchart Diagnosis.....	61
Gambar 2.2.2.13 Flowchart Ngobatin.....	62
Gambar 2.2.2.14 Flowchart Aku Boleh Pulang Gak.....	64
Gambar 2.2.2.15 Flowchart Daftar Checkup.....	69
Gambar 2.2.2.16 Flowchart Status Antrian Saya.....	76
Gambar 2.2.2.17 Flowchart Minum Obat.....	79
Gambar 2.2.2.18 Flowchart Minum Penawar.....	80
Gambar 2.2.2.19 Flowchart Dead or Alive.....	81
Gambar 2.2.2.20 Flowchart Mainin Antrian.....	87
Gambar 2.2.2.21 Flowchart Exit.....	89

BAB I

PENDAHULUAN

1.1. Deskripsi Permasalahan

Dalam mendukung kelancaran proses penanganan pasien, sistem pelayanan rumah sakit merupakan hal yang harus diperhatikan. Sayangnya, pada kenyataannya, masih banyak rumah sakit, khususnya yang belum terdigitalisasi sepenuhnya, mengalami sejumlah rintangan, seperti antrian yang tidak terstruktur, pencatatan pengobatan yang berantakan, serta pengelolaan data pengguna yang kurang baik.

Pada tugas besar ini, kami diminta untuk membuat sistem rumah sakit digital sederhana yang bisa dijalankan lewat *Command Line Interface (CLI)*. Meskipun tampilannya hanya teks, sistem ini harus bisa menjawab kebutuhan yang ada, seperti login dan logout pengguna, manajemen data pasien dan dokter, antrean pemeriksaan, hingga diagnosis dan pengobatan. Sistem ini nantinya akan digunakan oleh tiga jenis *user*, yakni manajer, dokter, dan pasien dengan hak aksesnya masing-masing

1.2 Tujuan

Program ini dikembangkan dengan tujuan fungsional untuk:

1. Manajemen User: Sistem memungkinkan manager untuk menambah dokter dan pasien ke dalam sistem. Setiap dokter dan pasien dapat melakukan login dengan akun yang telah terdaftar sebelumnya. Manager juga dapat mencari data pasien atau dokter menggunakan pencarian berdasarkan ID, nama, atau penyakit.
2. Pendaftaran dan Antrian Pasien: Pasien yang telah login dapat mendaftar check-up dengan memilih dokter yang tersedia. Kemudian, pasien dapat melihat posisi mereka dalam antrian.
3. Diagnosis dan Pengobatan: Sistem ini memudahkan dokter dalam pemberian pengobatan yang sesuai dengan diagnosis penyakit pasien.
4. Manajemen Ruangan: Fitur ini memungkinkan manager untuk mengassign dokter ke ruangan yang tersedia di rumah sakit. Manager dapat memastikan bahwa ruangan yang dipilih belum terisi oleh dokter lain sehingga alokasi ruangan tetap terorganisir.

Dengan demikian, program ini diharapkan menjadi solusi perbaikan alur interaksi antara pihak rumah sakit dengan pasien kedepannya.

BAB II

PERANCANGAN PROGRAM

2.1. Rencana Implementasi Fitur

Sebelum proses pengembangan dimulai, diperlukan perencanaan yang matang terkait bagaimana setiap fitur akan diimplementasikan. Rencana ini mencakup pemetaan fitur terhadap abstraksi data yang digunakan, strategi implementasi, serta alasan di balik pemilihan pendekatan tersebut. Tabel berikut menyajikan ringkasan rencana implementasi untuk masing-masing fitur dalam sistem.

Tabel 2.1.1 Rencana Implementasi

Implementasi ADT	Fitur	Deskripsi Implementasi	Alasan Implementasi
ADT Sederhana	F01–F02, F06–F18	Menggunakan struct sederhana (user, dokterInfo, antrianNode, dokterQueue, checkUpData) dan variabel global seperti currentUser, dokterQueues	Cocok untuk menyimpan data entitas sederhana tanpa relasi kompleks
ADT List	F01, F02, F06, F09–F15	Daftar user, dokter, dan antrian pasien disimpan dalam array statik dan linked list untuk antrian. Semua operasi seperti pencarian, penambahan, dan penghapusan dilakukan dengan indeks	Efisien untuk jumlah data terbatas dan struktur linear. Mudah digunakan dan diakses via indeks
ADT Linked List	F05, F09, F14, F15	Menggunakan FiturHelp* sebagai node untuk fitur per role, antrian pasien pada tiap dokter diimplementasikan dengan linked list (AntrianNode*)	Dinamis untuk pagination, memudahkan manipulasi antrian dengan jumlah pasien yang berubah-ubah
ADT Matrix	F06, F09	Representasi denah secara implisit dengan indeks baris dan kolom	Struktur ruang cocok direpresentasikan sebagai matriks/grid
ADT Set	F10, F13	Username dokter disimpan dalam set (simulasi array), untuk validasi keunikan. Fungsi insertSet() dan searchSet() menjaga tidak ada duplikasi	Set memastikan tidak ada duplikasi data seperti username dokter

ADT Map	F01, F13	Konsep username -> user digunakan saat login, meski teknisnya pakai pencarian manual di array. Digunakan juga dalam menyimpan pemetaan penyakit terhadap resep obat untuk verifikasi penyembuhan	Simulasi map mempercepat pencarian user berdasarkan key unik
ADT Stack	F13, F16, F17	Stack digunakan untuk menyimpan obat yang diminum dalam struktur StackObat perut. Fungsi popObat() mengeluarkan obat terakhir untuk dikembalikan ke inventory.	Stack cocok untuk menyimpan data yang perlu diakses secara mundur (LIFO)
ADT Queue	F09, F14–F15	Tiap dokter punya dokterQueue untuk antrian pasien. Pasien baru di-enqueue, pasien dilayani di-dequeue. Implementasi pakai linked lis	FIFO cocok untuk antrian layanan seperti pasien ke dokter, termasuk fitur check-up
File External	F01, F06, F12–F15	File data.c digunakan untuk menyimpan data secara permanen	Dibutuhkan untuk persistensi data antar sesi program
Fungsi, Prosedur	F01–F15, F18	Setiap fitur diimplementasi sebagai fungsi/prosedur modular (daftarCheckUp(), antrianSaya(), login(), tambah_pasien(), dst). Input, validasi, dan logika dipisah per prosedu	Modularisasi mempermudah debugging, pengujian, dan pengembangan fitur lanjutan
Array Search, Sort, Filter	F01, F02, F07–F15	Linear search dipakai untuk cari user saat login, cek username unik saat register, identifikasi antrian pasien pada dokter, dan validasi input check-up. Sort bubble berdasarkan ID/nama/aura, cari via binary & sequential search	Untuk mencocokkan akun saat login, mencakup sorting dinamis dan pencarian user berdasarkan kriteria

2.2. Desain Program

Pada bab ini, dijelaskan desain program untuk aplikasi manajemen rumah sakit yang dibangun menggunakan bahasa C. Program ini terdiri dari berbagai fitur yang

diimplementasikan dalam struktur modular dengan menggunakan command-line interface (CLI). Setiap fitur dipisahkan dalam file sumber C dan header, yang memudahkan pengelolaan kode dan pemeliharaan sistem.

Program ini mencakup berbagai fungsi seperti manajemen pengguna (dokter dan pasien), pendaftaran check-up, diagnosis, pengobatan, dan manajemen antrian. Desain sistem juga menggunakan struktur data seperti Set untuk validasi username, Map untuk penyakit dan obat, serta Queue untuk antrian pasien.

Berikut adalah daftar command yang ada di sistem beserta penjelasan rinciannya dan notasi algoritmik masing-masing fitur yang telah diimplementasikan.

2.2.1. Desain *Command*

Setiap fitur dalam sistem ini diakses melalui perintah khusus yang dijalankan oleh pengguna melalui antarmuka berbasis teks. Oleh karena itu, perancangan command menjadi bagian penting dalam memastikan interaksi pengguna berjalan lancar dan sesuai ekspektasi. Tabel berikut merinci nama *command*, masukan yang diperlukan, serta keluaran yang dihasilkan untuk masing-masing fitur.

Tabel 2.2.1.1 Rincian *Command*

Fitur	Nama <i>command</i>	Masukan	Keluaran
F01 - Login	LOGIN	Data manager, list dokter, dan pasien	<ul style="list-style-type: none"> - Jika berhasil login, <code>currentUser</code> dan <code>isLoggedIn</code> diperbarui - Jika gagal, tampilkan pesan gagal login
F02 - Register Pasien	REGISTER	<code>pasienList</code> , <code>pasienCount</code> , dan <code>usernameSet</code> terdefinisi	<ul style="list-style-type: none"> - Jika username unik, akun baru pasien ditambahkan ke <code>pasienList</code>. - Jika username sudah ada, tampilkan pesan gagal register
F03 - Logout	LOGOUT	User dalam keadaan login (<code>isLoggedIn = 1</code> dan <code>currentUser</code> terisi)	User berhasil logout, <code>currentUser</code> dikosongkan, dan <code>isLoggedIn = 0</code>

F04 - Lupa Password	LUPA_PASSWORD	Data manager, list dokter, dan pasien	Jika username ditemukan, tampilkan kode unik hasil encoding. Jika tidak, tampilkan pesan bahwa username tidak ditemukan
F05 - Menu & Help	HELP	isLoggedIn dan currentUser terdefinisi (isLoggedIn = 1)	Menampilkan daftar fitur (perintah) yang bisa digunakan oleh user berdasarkan role (Manager, Dokter, atau Pasien). Jika belum login, menampilkan fitur untuk user belum login.
	displayHelpMenu	role merupakan salah satu dari manager, dokter, atau pasien	Menampilkan daftar fitur sesuai dengan role dalam bentuk paginasi (per halaman).
F06 - Denah Rumah Sakit	LIHAT_DENAH	jumlahRuang terdefinisi	Menampilkan denah RS ke layar berupa kotak berisi nomor ruangan
	LIHAT_RUANGAN	indeks dan ruanganList terdefinisi	Menampilkan detail ruangan ke layar: kapasitas, dokter dan daftar pasien (jika ada)
F07 - Lihat User	LIHAT_USER	Telah login, currentUser terdefinisi, pasienList dan dokterList sudah berisi data pasien dan dokter	Menampilkan seluruh user ke layar secara teratur. Hanya bisa diakses manager
	LIHAT_PASIEN	Telah login, currentUser terdefinisi,	Menampilkan seluruh pasien ke layar secara teratur.

		pasienList sudah berisi data dokter	Hanya bisa diakses manager
	LIHAT_DOKTER	Telah login, currentUser terdefinisi, dokterList sudah berisi data dokter	Menampilkan seluruh dokter ke layar secara terurut. Hanya bisa diakses manager
F08 - Cari User	CARI_USER	Semua data user (manager, dokter, pasien) sudah terdefinisi dan terurut berdasarkan ID	Menampilkan data user (Manager, Dokter, atau Pasien) sesuai dengan input pencarian berdasarkan ID atau Nama
	CARI_DOKTER	dokterList terdefinisi dan tidak kosong jika dokterCount > 0	Menampilkan informasi dokter berdasarkan pencarian ID atau Nama
	CARI_PASIEN	pasienList telah terdefinisi dan terurut berdasarkan ID	Menampilkan pasien berdasarkan ID, Nama, atau Penyakit
F09 - Lihat Antrian	LIHAT_ANTRIAN	Seluruh ruangan dan data antrian terdefinisi, user sudah login	Menampilkan denah ruangan dan detail informasi tiap ruangan termasuk pasien dan antriannya
F10 - Tambah Dokter	TAMBAH_DOKTER	Data dokterList, dokterInfoList, usernameSet, dan dokterCount terdefinisi	Jika username belum terdaftar, dokter baru ditambahkan ke semua struktur terkait
	ASSIGN_DOKTER	Data dokterList, dokterInfoList, dan ruanganList terdefinisi. Manager menginput username	Jika nama dokter valid dan ruangan belum terpakai, dokter berhasil diassign ke ruangan

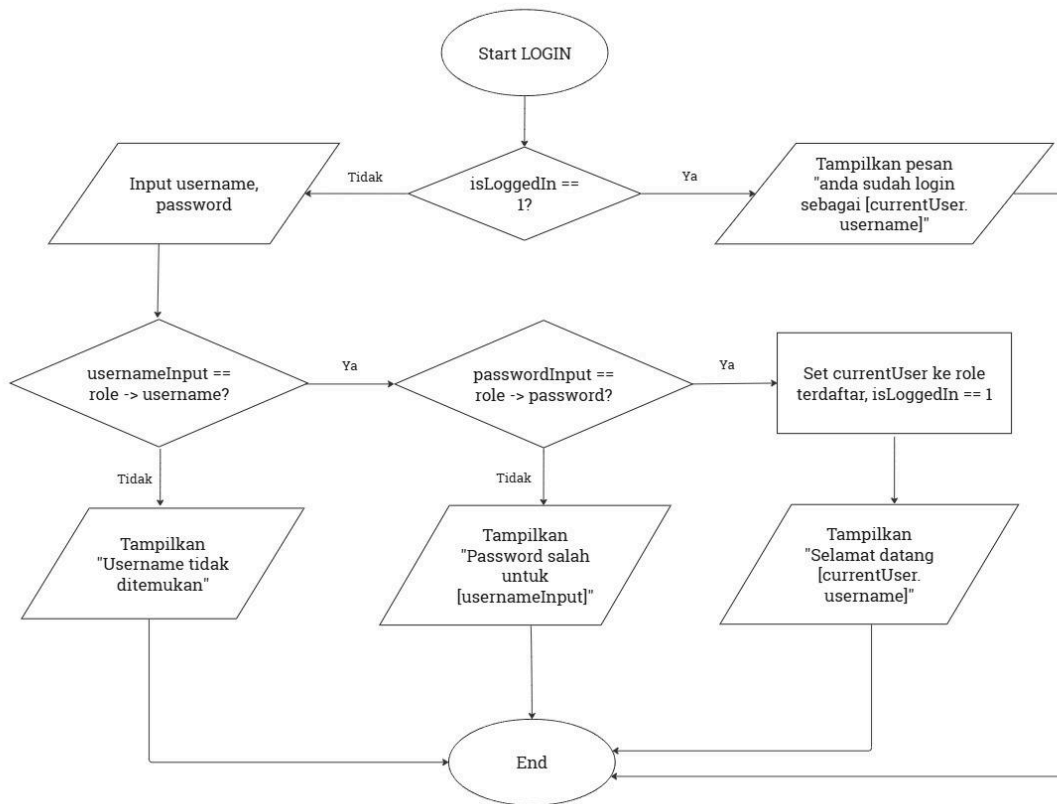
		dokter dan nomor ruangan yang ingin dituju	
F11 - Diagnosis	DIAGNOSIS	list penyakit telah terisi dan data pasien telah lengkap	Menampilkan diagnosis penyakit yang cocok (jika ada)
F12 - Ngobatin	NGOBATIN	Data pasien telah terdefinisi, termasuk nama penyakit jika sudah terdiagnosis	Menampilkan nama penyakit dan daftar obat dari map penyakit yang sesuai
F13 - Aku Boleh Pulang Ga, Dok	PULANG_DOK	dataPasien, userloggedIn sudah terisi jika pasien pernah didiagnosis	<ul style="list-style-type: none"> - Jika obat yang diminum sesuai resep, pasien dianggap sembuh dan keluar dari sistem - Jika tidak sesuai, nyawa berkurang. Jika nyawa habis, data pasien dihapus dari sistem
F14 - Daftar Check-Up	DAFTAR_CHECKUP	Pengguna sudah login sebagai pasien dan data dokterInfoList, dokterCount, dokterlistmain, doktercountmain terdefinisi. Struktur data antrian dokter sudah diinisialisasi dengan initializeDokterQueues	Pasien berhasil mendaftar ke salah satu antrian dokter yang tersedia beserta data chek-up lengkap. Antrian dokter bertambah
F15 - Antrian Saya	ANTRIAN_SAYA	array of Ruangan dan jumlahRuangan terdefinisi, DokterQueues terinisialisasi, currentUser	<ul style="list-style-type: none"> - Jika belum login atau role bukan pasien, akan ditampilkan pesan error - Jika pasien sedang

		terdefinisi	berada dalam ruangan, maka akan muncul notifikasi keberadaan pasien di ruangan. Jika pasien ada dalam antrian, maka akan ditampilkan status antrian: dokter, ruangan, posisi. Jika tidak ditemukan, sistem akan menampilkan pesan bahwa pasien belum mendaftar check-up.
F16 - Minum Obat	MINUM_OBAT	Data pasien, stack perut (obat yang sedang diminum), dan inventory (daftar obat pasien) sudah terdefinisi dan valid. Inventory berisi obat yang tersedia untuk diminum (bisa kosong atau tidak)	Jika inventory kosong, tampil pesan pemberitahuan. Jika tidak, tampilkan daftar obat, pengguna memilih satu obat yang kemudian dipindahkan ke perut, muncul pesan keberhasilan, obat tersebut dihapus dari inventory, dan daftar inventory diperbarui atau diberi pesan jika kosong.
F17 - Minum Penawar	PENAWAR	Stack perut berisi obat yang sedang diminum (bisa kosong atau tidak). Inventory berisi daftar obat pasien (list obat)	Jika perut kosong, tampil pesan bahwa tidak ada obat yang bisa dikeluarkan. Jika tidak kosong, satu obat diambil dari perut dan dikembalikan ke inventory, serta muncul pesan keberhasilan pengembalian obat.

F18 - Exit	EXIT	Program sedang berjalan dan menerima perintah keluar	Program berhenti dan keluar dari aplikasi dengan menampilkan pesan perpisahan
------------	------	--	---

2.2.2. Desain Kamus Data, Dekomposisi, dan Spesifikasi Modul

• F01 - Login



Gambar 2.2.2.1 Flowchart Login

1. Procedure login(managerData: pointer to User, dokterListParam: array of User, dokterCountParam: integer, pasienListParam: array of User, pasienCountParam: integer)
2. { I.S. Data manager, dokter, dan pasien sudah terdefinisi; isLoggedIn dan currentUser global; user belum login atau sudah login. }
3. { F.S. Status login diperbarui (isLoggedIn = 1 jika login berhasil, currentUser di-update), dan pesan login ditampilkan sesuai hasil verifikasi. }
- 4.
5. KAMUS
6. usernameInput : string

```

7.     passwordInput : string
8.     i : integer
9.     nextChar : char
10.
11. ALGORITMA
12.     if isLoggedIn = 1 then
13.         output "Anda sudah login sebagai ", currentUser.username
14.         return
15.     end if
16.
17.     output "Username: "
18.     input usernameInput
19.     if input gagal then
20.         output "Input username gagal, Coba lagi."
21.         panggil clearInputBuffer
22.         return
23.     end if
24.
25.     if panjang usernameInput ≥ MAX_USERNAME_LEN then
26.         baca nextChar
27.         if nextChar ≠ newline or EOF or spasi or tab then
28.             output "Username terlalu panjang!"
29.             kembalikan nextChar ke input buffer
30.             panggil clearInputBuffer
31.             return
32.         end if
33.         if nextChar ≠ EOF then kembalikan ke input buffer
34.     end if
35.     panggil clearInputBuffer
36.
37.     output "Password: "
38.     input passwordInput
39.     if input gagal then
40.         output "Input password gagal. Coba lagi."
41.         panggil clearInputBuffer
42.         return
43.     end if
44.
45.     if panjang passwordInput ≥ MAX_PASSWORD_LEN then
46.         baca nextChar
47.         if nextChar ≠ newline, EOF, spasi, atau tab then
48.             output "Password terlalu panjang!"
49.             kembalikan nextChar ke input buffer
50.             panggil clearInputBuffer
51.             return
52.         end if
53.         if nextChar ≠ EOF then kembalikan ke input buffer
54.     end if
55.     panggil clearInputBuffer
56.
57.     if usernameInput = managerData.username then
58.         if passwordInput = managerData.password then
59.             currentUser ← *managerData
60.             isLoggedIn ← 1

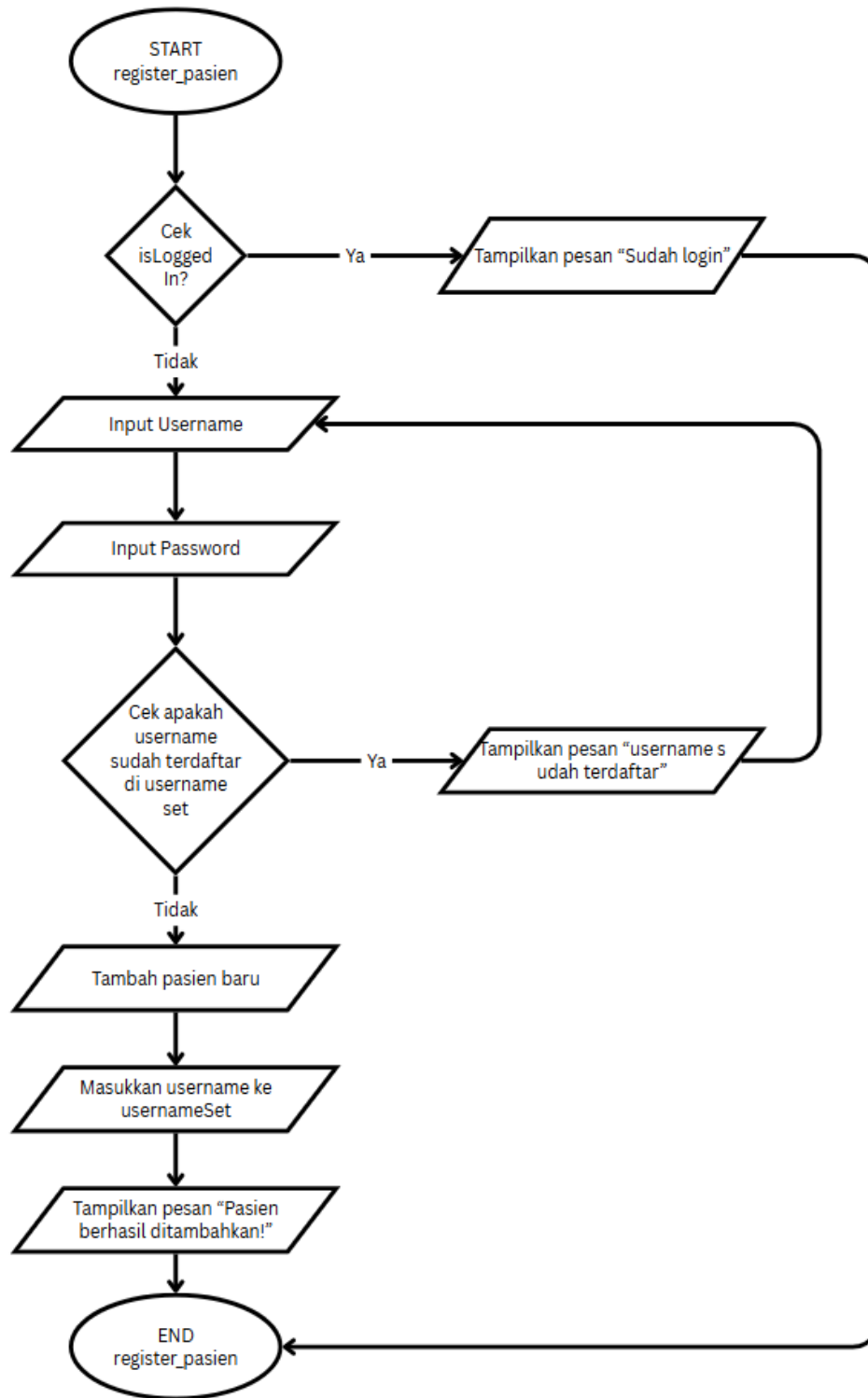
```

```

61.         output "Selamat datang Manager ", currentUser.username
62.     else
63.         output "Password salah untuk pengguna bernama ",
64.             usernameInput
65.     end if
66.     return
67. end if
68.
69. for i ← 0 to dokterCountParam - 1 do
70.     if usernameInput = dokterListParam[i].username then
71.         if passwordInput = dokterListParam[i].password then
72.             currentUser ← dokterListParam[i]
73.             isLoggedIn ← 1
74.             output "Selamat pagi Dokter ", currentUser.username
75.         else
76.             output "Password salah untuk pengguna bernama ",
77.                 usernameInput
78.         end if
79.         return
80.     end if
81. end for
82.
83. for i ← 0 to pasienCountParam - 1 do
84.     if usernameInput = pasienListParam[i].username then
85.         if passwordInput = pasienListParam[i].password then
86.             currentUser ← pasienListParam[i]
87.             isLoggedIn ← 1
88.             output "Selamat pagi ", currentUser.username, "! Ada
89.                 keluhan apa?"
90.         else
91.             output "Username atau password salah untuk pengguna
92.                 bernama ", usernameInput
93.         end if
94.         return
95.     end if
96. end for
97.
98. output "Tidak ada Manager, Dokter, atau pun Pasien yang bernama ",
99.     usernameInput

```

- **F02 - Register Pasien**



Gambar 2.2.2.2 *Flowchart Register*

```

1. Procedure registerPasien(listPasien: array of User, jumlahPasienAktif:
   pointer to integer, usernameSetGlobal: pointer to Set)
2. { I.S. User belum terdaftar dan belum login. }
3. { F.S. Pasien baru berhasil terdaftar jika tidak ada error. }
4.
5.   KAMUS
6.     tempUsername : string
7.     tempPassword : string
8.     i : integer
9.     idxBaru : integer
10.
11.
12.   ALGORITMA
13.     if isLoggedIn = 1 then
14.       output "Anda sudah login sebagai ", currentUser.username, ".
15.       Silakan logout dulu untuk register akun baru."
16.       return
17.     end if
18.
19.     if *jumlahPasienAktif ≥ MAX_JUMLAH_USER then
20.       output "Registrasi gagal! Kapasitas penyimpanan pasien di sistem
21.       sudah penuh."
22.       return
23.     end if
24.
25.     output "Username: "
26.     input tempUsername
27.     if input gagal then
28.       output "Input username gagal."
29.       clearInputBuffer
30.       return
31.     end if
32.     if strlen(tempUsername) ≥ MAX_USERNAME_LEN then
33.       output "Registrasi gagal! Username terlalu panjang."
34.       clearInputBuffer
35.       return
36.     end if
37.     clearInputBuffer
38.
39.     for i ← 0 to strlen(tempUsername) - 1 do
40.       if not ((tempUsername[i] ≥ 'a' && tempUsername[i] ≤ 'z') ||
41.         (tempUsername[i] ≥ 'A' && tempUsername[i] ≤ 'Z')) then
42.         output "Registrasi gagal! Username hanya boleh terdiri dari
43.         huruf."
44.         return
45.       end if
46.     end for
47.
48.     output "Password: "
49.     input tempPassword
50.     if input gagal then
51.       output "Input password gagal."
52.       clearInputBuffer
53.       return

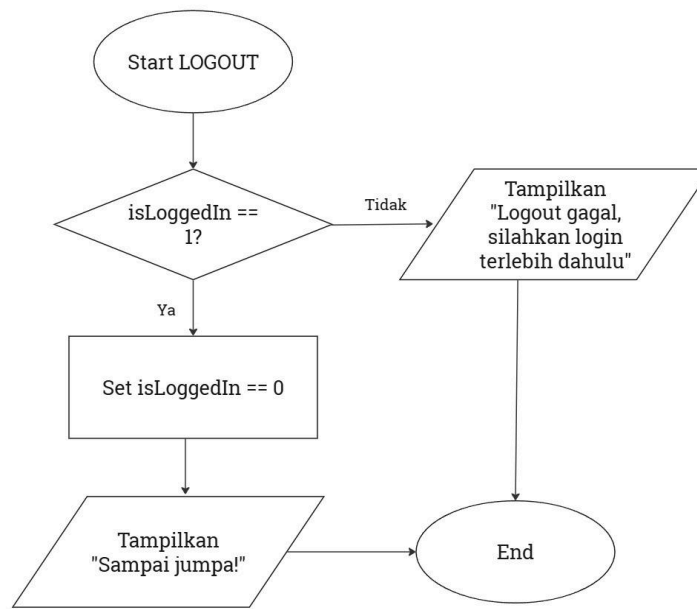
```

```

54.     end if
55.     if strlen(tempPassword) ≥ MAX_PASSWORD_LEN then
56.         output "Registrasi gagal! Password terlalu panjang."
57.         clearInputBuffer
58.         return
59.     end if
60.     clearInputBuffer
61.
62.     if searchSet(*usernameSetGlobal, tempUsername) then
63.         output "Registrasi gagal! Pasien dengan nama ", tempUsername, "
64.             sudah terdaftar."
65.         return
66.     end if
67.
68.     idpenggunasebelum ← idpenggunasebelum + 1
69.     idxBaru ← *jumlahPasienAktif
70.     listPasien[idxBaru].id ← idpenggunasebelum
71.     strncpy(listPasien[idxBaru].username, tempUsername, MAX_USERNAME_LEN
72.         - 1)
73.     strncpy(listPasien[idxBaru].password, tempPassword, MAX_PASSWORD_LEN
74.         - 1)
75.     strcpy(listPasien[idxBaru].role, "Pasien")
76.
77.     listPasien[idxBaru].namaPenyakit ← ""
78.     listPasien[idxBaru].nyawa ← 3
79.     listPasien[idxBaru].dokterPerawat ← ""
80.     listPasien[idxBaru].aura ← 0
81.     listPasien[idxBaru].kodeRuangan ← ""
82.     listPasien[idxBaru].suhuTubuh ← 0.0f
83.     listPasien[idxBaru].tekananDarahSistolik ← 0
84.     listPasien[idxBaru].tekananDarahDiastolik ← 0
85.     listPasien[idxBaru].detakJantung ← 0
86.     listPasien[idxBaru].saturasiOksigen ← 0.0f
87.     listPasien[idxBaru].kadarGulaDarah ← 0
88.     listPasien[idxBaru].beratBadan ← 0.0f
89.     listPasien[idxBaru].tinggiBadan ← 0.0f
90.     listPasien[idxBaru].kadarKolesterol ← 0
91.     listPasien[idxBaru].trombosit ← 0
92.
93.     (*jumlahPasienAktif) ← *jumlahPasienAktif + 1
94.     insertSet(usernameSetGlobal, tempUsername)
95.     output "Pasien ", tempUsername, " berhasil ditambahkan!"

```

● F03 - Logout



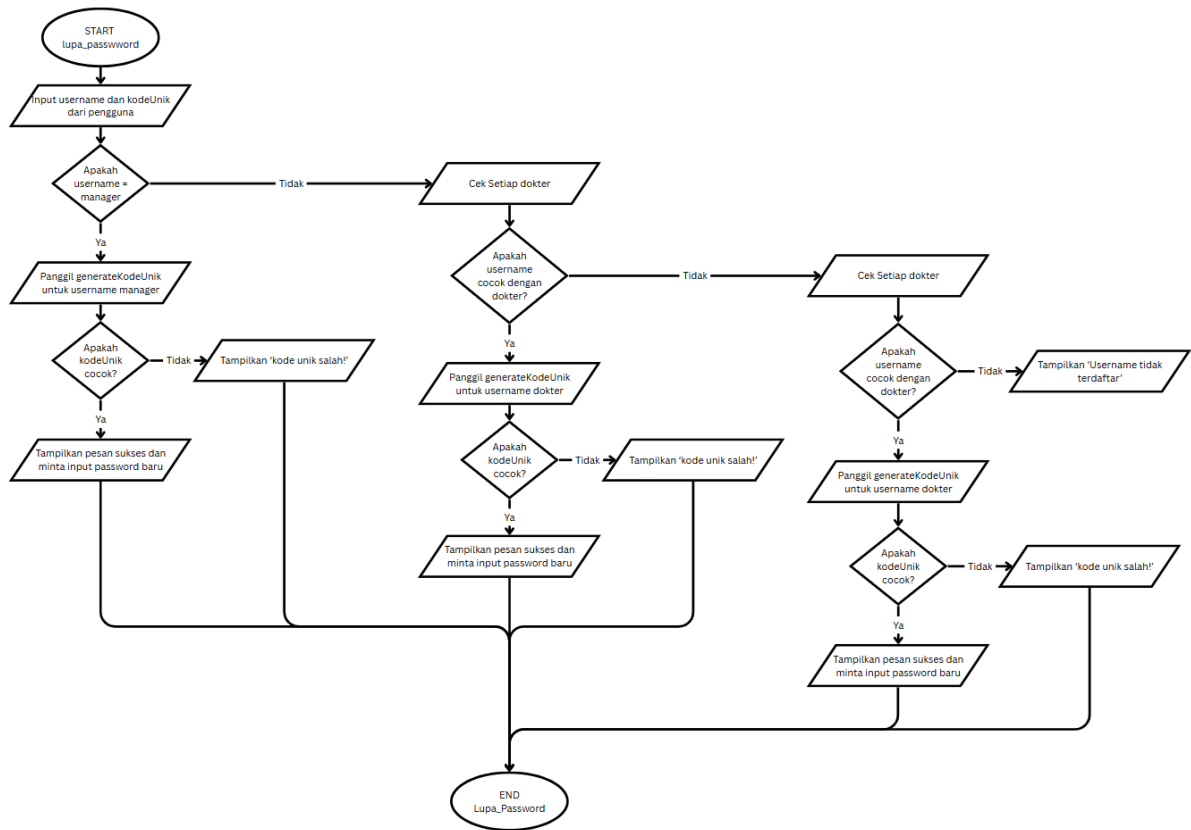
Gambar 2.2.2.3 *Flowchart Logout*

```

1. Procedure logout()
2. { I.S. Pengguna sudah login atau belum }
3. { F.S. Pengguna berhasil logout dan status isLoggedIn diubah menjadi 0 }
4.
5. KAMUS
6.   isLoggedIn: integer
7.
8. ALGORITMA
9.   if isLoggedIn = 0 then
10.     output "Logout gagal! Anda belum login, silahkan login terlebih
11.     dahulu sebelum melakukan logout."
12.     return
13.   end if
14.
15.   output "Sampai jumpa!"
16.   isLoggedIn ← 0
17.   memset(&currentUser, 0, sizeof(currentUser))
18.

```

- **F04 - Lupa Password**



Gambar 2.2.2.4 Flowchart Lupa Password

```

1.  Procedure lupaPassword(managerData: pointer to User, dokterListParam:
    array of User, dokterCountParam: integer, pasienListParam: array of User,
    pasienCountParam: integer)
2.  {
3.      I.S. User ingin mereset password.
4.      F.S. Password berhasil diubah jika username dan kode unik sesuai.
5.
6.      KAMUS
7.          username : string
8.          kodeUnikInput : string
9.          expectedKodeUnik : string
10.         newPassword : string
11.
12.     ALGORITMA
13.         output "Username: "
14.         input username
15.         if input gagal then
16.             clearInputBuffer
17.             return
18.         end if
19.         if strlen(username) > MAX_USERNAME_LEN then
20.             output "Invalid. Username terlalu panjang!"
21.             clearInputBuffer

```

```

22.         return
23.     end if
24.     clearInputBuffer
25.
26.     output "Kode Unik: "
27.     input kodeUnikInput
28.     clearInputBuffer
29.
30.     generateKodeUnik(username, expectedKodeUnik)
31.
32.     if kodeUnikInput ≠ expectedKodeUnik then
33.         output "Kode unik salah!"
34.         return
35.     end if
36.
37.     if username = managerData.username then
38.         output "Halo Manager ", username, ", silakan daftarkan ulang
39.         password anda!"
40.         output "Password Baru: "
41.         input newPassword
42.         if input gagal then
43.             clearInputBuffer
44.             return
45.         end if
46.         if strlen(newPassword) ≥ MAX_PASSWORD_LEN then
47.             output "Password terlalu panjang!"
48.             clearInputBuffer
49.             return
50.         end if
51.         strncpy(managerData.password, newPassword, MAX_PASSWORD_LEN - 1)
52.         managerData.password[MAX_PASSWORD_LEN - 1] ← '\0'
53.         output "Password berhasil diubah!"
54.         return
55.     end if
56.
57.     for i ← 0 to dokterCountParam - 1 do
58.         if username = dokterListParam[i].username then
59.             output "Halo Dokter ", username, ", silakan daftarkan ulang
60.             password anda!"
61.             output "Password Baru: "
62.             input newPassword
63.             if input gagal then
64.                 clearInputBuffer
65.                 return
66.             end if
67.             if strlen(newPassword) ≥ MAX_PASSWORD_LEN then
68.                 output "Password terlalu panjang!"
69.                 clearInputBuffer
70.                 return
71.             end if
72.             strncpy(dokterListParam[i].password, newPassword,
73.             MAX_PASSWORD_LEN - 1)
74.             dokterListParam[i].password[MAX_PASSWORD_LEN - 1] ← '\0'
75.             output "Password berhasil diubah!"

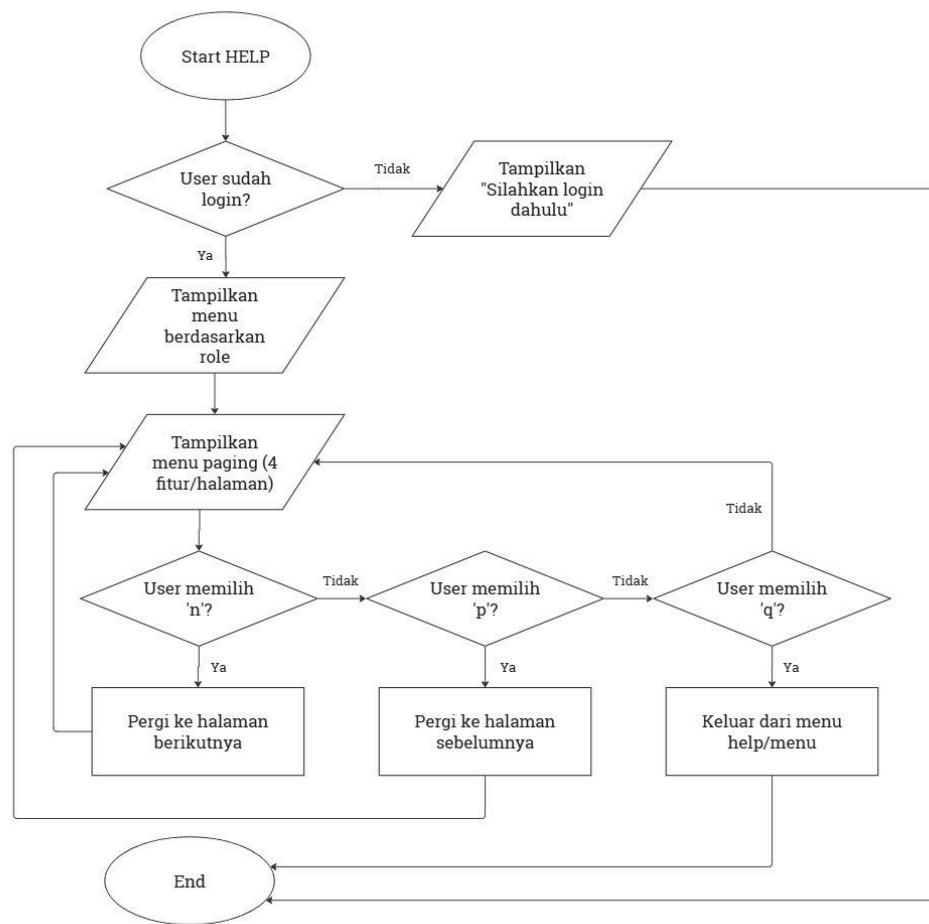
```

```

76.         return
77.     end if
78. end for
79.
80. for i ← 0 to pasienCountParam - 1 do
81.     if username = pasienListParam[i].username then
82.         output "Halo ", username, ", silakan daftarkan ulang password
83.         anda!"
84.         output "Password Baru: "
85.         input newPassword
86.         if input gagal then
87.             clearInputBuffer
88.             return
89.         end if
90.         if strlen(newPassword) ≥ MAX_PASSWORD_LEN then
91.             output "Password terlalu panjang!"
92.             clearInputBuffer
93.             return
94.         end if
95.         strncpy(pasienListParam[i].password, newPassword,
96.         MAX_PASSWORD_LEN - 1)
97.         pasienListParam[i].password[MAX_PASSWORD_LEN - 1] ← '\0'
98.         output "Password berhasil diubah!"
99.         return
100.    end if
101. end for
102.
103.    output "Username tidak terdaftar!"

```

- **F05 - Help and Menu**



Gambar 2.2.2.5 *Flowchart Menu Help*

```

1. Procedure help()
2. { I.S. Pengguna belum login atau sudah login. }
3. { F.S. Menampilkan menu bantuan sesuai dengan status login pengguna. }
4.
5. KAMUS
6.   input : string
7.
8. ALGORITMA
9.   output "\n===== HELP =====\n\n"
10.
11.   if isLoggedIn = 0 then
12.     output "Kamu belum login sebagai role apapun. Silahkan login
13.     terlebih dahulu.\n\n"
14.     output "1. LOGIN: Masuk ke dalam akun yang sudah terdaftar\n"
15.     output "2. REGISTER: Membuat akun baru\n"
16.   else
17.     tampilkanMenuHelp(currentUser.role)
18.   end if

```



```

19.
20.     output "\nFootnote:\n"
21.     output "1. Untuk menggunakan aplikasi, silahkan masukkan nama
22.     fungsi yang terdaftar\n"
23.     output "2. Jangan lupa untuk memasukkan input yang valid\n"

24. Procedure createLinkedList(features: array of string, featureCount:
25. integer)
26. { I.S. Daftar fitur yang ingin dibuatkan linked list. }
27. { F.S. Menghasilkan linked list yang berisi fitur-fitur. }
28.
29.     KAMUS
30.         head : FiturHelp
31.         tail : FiturHelp
32.         newNode : FiturHelp
33.         i : integer
34.
35.     ALGORITMA
36.         head ← NULL
37.         tail ← NULL
38.
39.         for i ← 0 to featureCount - 1 do
40.             newNode ← alokasiMemori(FiturHelp)
41.             strcpy(newNode.feature, features[i])
42.             newNode.next ← NULL
43.
44.             if head = NULL then
45.                 head ← newNode
46.                 tail ← newNode
47.             else
48.                 tail.next ← newNode
49.                 tail ← newNode
50.             end if
51.         end for
52.
53.         return head

54. Procedure displayLinkedList(head: FiturHelp)
55. { I.S. Linked list fitur sudah tersedia. }
56. { F.S. Menampilkan daftar fitur dalam bentuk halaman. }
57.
58.     KAMUS
59.         itemsPerPage : integer
60.         totalItems : integer
61.         totalPages : integer
62.         currentPage : integer
63.         current : FiturHelp
64.         nomor : integer
65.         input : string
66.
67.     ALGORITMA
68.         itemsPerPage ← 4

```

```

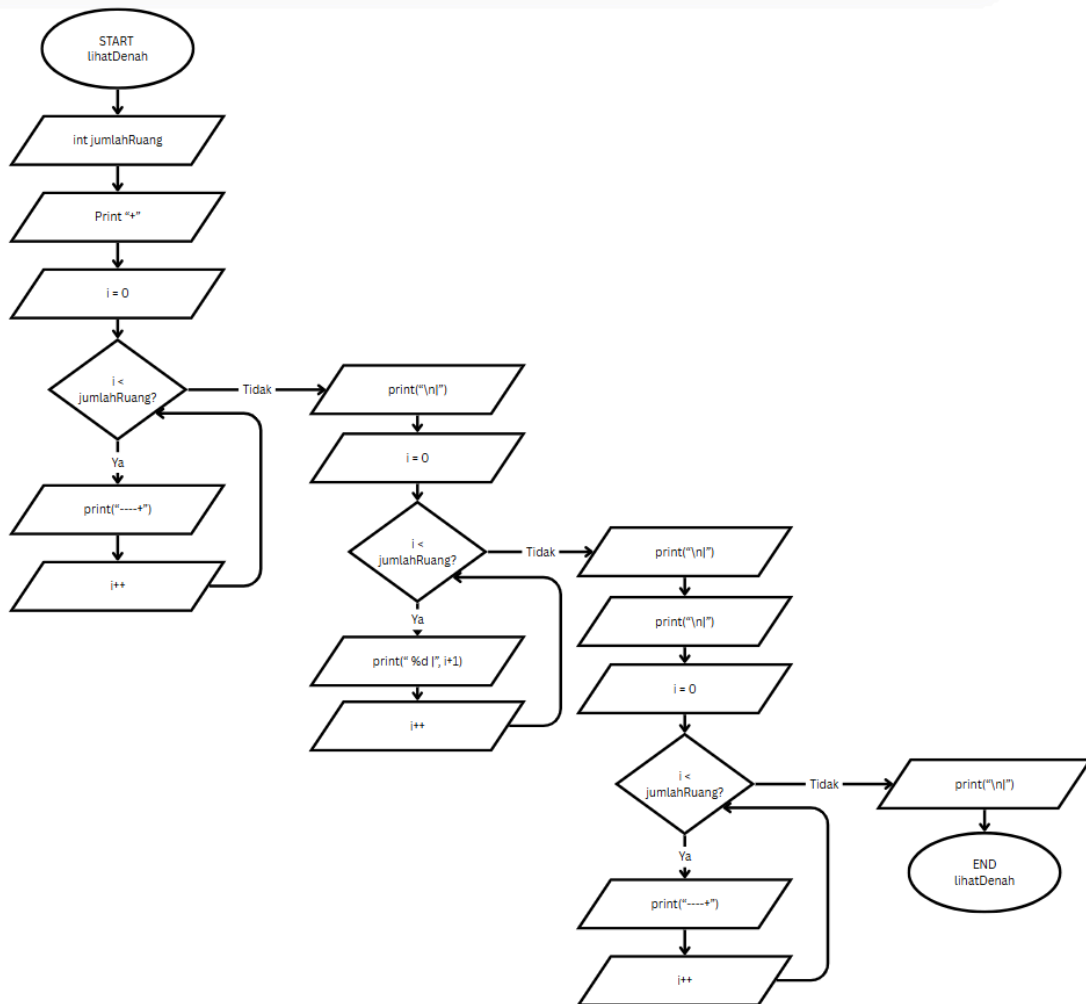
69.     totalItems ← 0
70.     current ← head
71.
72.     while current ≠ NULL do
73.         totalItems ← totalItems + 1
74.         current ← current.next
75.     end while
76.
77.     totalPages ← (totalItems + itemsPerPage - 1) / itemsPerPage
78.     currentPage ← 0
79.     input ← ""
80.
81.     do
82.         #clear screen
83.
84.         output "===== MENU (Page ", currentPage + 1, " of ",
85.             totalPages, ") =====\n\n"
86.
87.         current ← head
88.         nomor ← 1 + currentPage * itemsPerPage
89.
90.         for i ← 0 to currentPage * itemsPerPage - 1 do
91.             current ← current.next
92.         end for
93.
94.         for i ← 0 to itemsPerPage - 1 do
95.             if current ≠ NULL then
96.                 output nomor, ". ", current.feature, "\n"
97.                 nomor ← nomor + 1
98.                 current ← current.next
99.             end if
100.        end for
101.
102.        output "\n[n] Next page | [p] Previous page | [q] Quit
103.        help\n"
104.        output ">>> "
105.        input ← getInput()
106.
107.        if input = "n" and currentPage < totalPages - 1 then
108.            currentPage ← currentPage + 1
109.        else if input = "p" and currentPage > 0 then
110.            currentPage ← currentPage - 1
111.        else if input ≠ "q" and input ≠ "n" and input ≠ "p" then
112.            output "Perintah tidak dikenal. Gunakan 'n', 'p', atau
113.            'q'.\n"
114.        end if
115.    while input ≠ "q"

116. Procedure deleteLinkedList(head: FiturHelp)
117. { I.S. Linked list fitur yang sudah ditampilkan. }
118. { F.S. Linked list dihapus dari memori. }
119.
120.     KAMUS

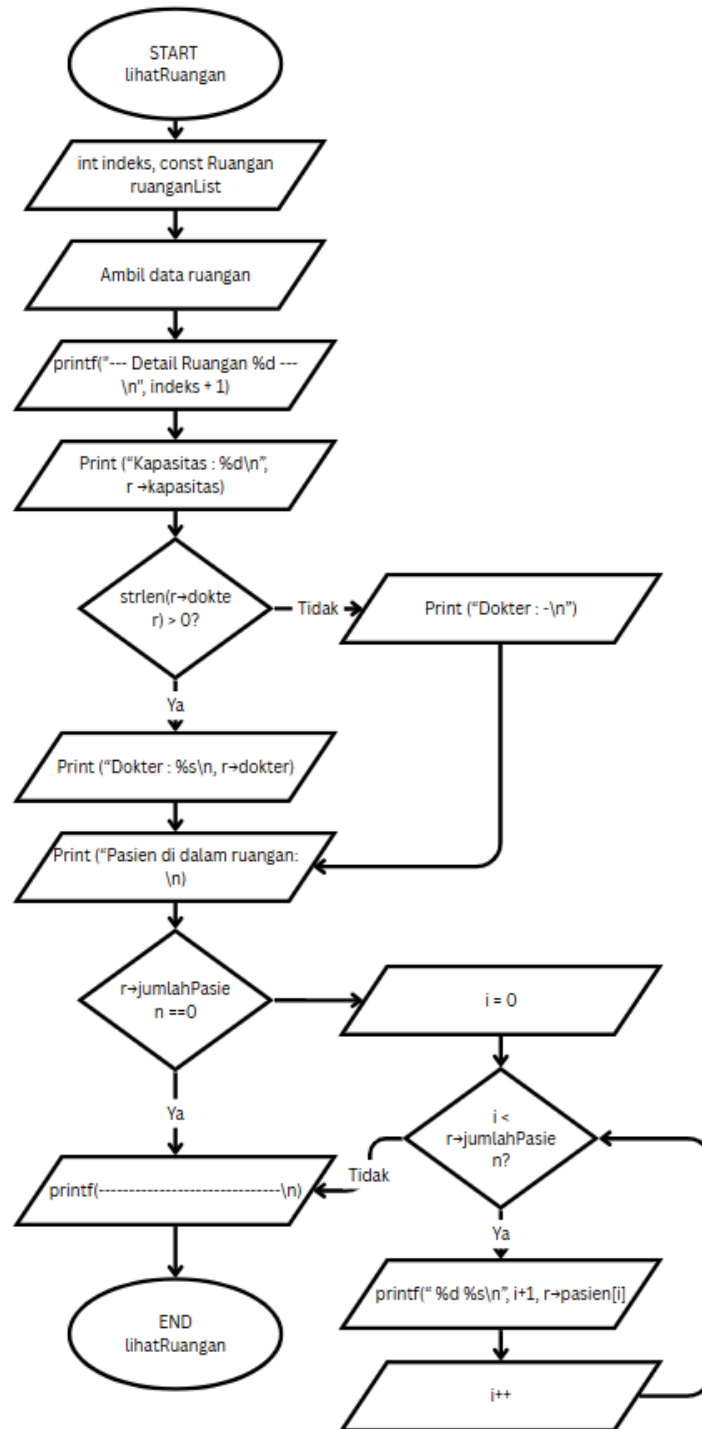
```

```
121.     current : FiturHelp
122.     toDelete : FiturHelp
123.
124.     ALGORITMA
125.         current ← head
126.         while current ≠ NULL do
127.             toDelete ← current
128.             current ← current.next
129.             free(toDelete)
130.         end while
```

- **F06 - Denah Rumah Sakit**



Gambar 2.2.2.6 *Flowchart* Lihat Denah



Gambar 2.2.2.7 Flowchart Lihat Ruangan

1. Procedure lihatDenah(jumlahRuanganIgnored: integer, barisDenahAktual: integer, kolomDenahAktual: integer)
2. { I.S. Dimensi denah rumah sakit (jumlah baris dan kolom) sudah terdefinisi. }

```

3.  { F.S. Menampilkan denah rumah sakit sesuai dengan jumlah baris dan
4.    kolom yang diberikan. }
5.
6.  KAMUS
7.    j : integer
8.    i : integer
9.
10. ALGORITMA
11.   if barisDenahAktual ≤ 0 or kolomDenahAktual ≤ 0 then
12.     output "Dimensi denah tidak valid untuk ditampilkan."
13.     return
14.   end if
15.
16.   output "      "
17.   for j ← 0 to kolomDenahAktual - 1 do
18.     output j + 1, "      "
19.   end for
20.   output "\n"
21.
22.   output "    +"
23.   for j ← 0 to kolomDenahAktual - 1 do
24.     output "-----+"
25.   end for
26.   output "\n"
27.
28.   for i ← 0 to barisDenahAktual - 1 do
29.     output " ", 'A' + i, " |"
30.     for j ← 0 to kolomDenahAktual - 1 do
31.       output " ", 'A' + i, j + 1, " |"
32.     end for
33.     output "\n"
34.
35.     output "    +"
36.     for j ← 0 to kolomDenahAktual - 1 do
37.       output "-----+"
38.     end for
39.     output "\n"
40.   end for

```

```

41. Procedure lihatRuangan(indeks: integer, ruanganListFromParam: array of
    Ruangan, denahKolomAktual: integer)
42. { I.S. Denah ruangan sudah tersedia dan indeks ruangan sudah diberikan. }
43. { F.S. Menampilkan detail ruangan berdasarkan indeks yang diberikan. }
44.
45. KAMUS
46.   kodeRuangStr : string
47.   r : Ruangan
48.
49. ALGORITMA
50.   if denahKolomAktual ≤ 0 then
51.     output "Informasi kolom denah tidak valid untuk menampilkan
52.       detail ruangan."
53.     return

```

```

54.     end if
55.
56.     if ruanganListFromParam = NULL or indeks < 0 or indeks ≥ (MAX_BARIS *
57. MAX_KOLOM) then
58.         output "Indeks ruangan tidak valid atau denah belum
59.         diinisialisasi."
60.         return
61.     end if
62.
63.     r ← ruanganListFromParam[indeks]
64.     snprintf(kodeRuangStr, sizeof(kodeRuangStr), "%c%d", 'A' + (indeks /
65. denahKolomAktual), (indeks % denahKolomAktual) + 1)
66.
67.     output "\n--- Detail Ruangan ", kodeRuangStr, " ---\n"
68.     output "Kapasitas   : ", r.kapasitas
69.     output "Dokter      : ", if(strlen(r.dokter) > 0) then r.dokter else
70.     "-"
71.     output "Pasien di dalam ruangan:"
72.     if r.jumlahPasien = 0 then
73.         output "   Tidak ada pasien di dalam ruangan saat ini."
74.     else
75.         for i ← 0 to r.jumlahPasien - 1 do
76.             if r.pasien[i][0] ≠ '\0' then
77.                 output "   ", i + 1, ". ", r.pasien[i]
78.             end if
79.         end for
80.     end if
81.     output "-----\n"

```

```

82. Procedure ubahDenah(ptrJumlahRuangTotal: pointer to integer,
    ptrToRuanganListGlobal: pointer to array of Ruangan, barisBaru: integer,
    kolomBaru: integer, ptrDenahBarisMain: pointer to integer,
    ptrDenahKolomMain: pointer to integer)
83. { I.S. Ukuran denah yang baru (baris dan kolom) diberikan. }
84. { F.S. Ukuran denah diubah, memori ruangan diubah sesuai jumlah baris dan
85.   kolom baru. }
86.
87. KAMUS
88.   jumlahRuangBaru : integer
89.   jumlahRuangLama : integer
90.   kolomLama : integer
91.   ruanganListBaru : pointer to array of Ruangan
92.   i : integer
93.   j : integer
94.   kodeRuangHilang : string
95.
96. ALGORITMA
97.   if barisBaru ≤ 0 or kolomBaru ≤ 0 or barisBaru > MAX_BARIS or
98.   kolomBaru > MAX_KOLOM then
99.       output "Ukuran denah baru tidak valid!"
100.      return FALSE
101.   end if
102.

```

```

103.     jumlahRuangBaru ← barisBaru * kolomBaru
104.     jumlahRuangLama ← *ptrJumlahRuangTotal
105.     kolomLama ← *ptrDenahKolomMain
106.
107.     if jumlahRuangBaru < jumlahRuangLama then
108.         for i ← jumlahRuangBaru to jumlahRuangLama - 1 do
109.             if strlen((*ptrToRuanganListGlobal)[i].dokter) > 0 then
110.                 snprintf(kodeRuangHilang, sizeof(kodeRuangHilang),
111.                     "%c%d", 'A' + (i / kolomLama), (i % kolomLama) + 1)
112.                 output "Tidak dapat mengubah ukuran denah. Ruangan ",
113.                     kodeRuangHilang, " masih ditempati oleh Dr. ",
114.                     (*ptrToRuanganListGlobal)[i].dokter, ". Silakan
115.                     pindahkan dokter terlebih dahulu."
116.                 return FALSE
117.             end if
118.         end for
119.     end if
120.
121.     ruanganListBaru ← realloc(*ptrToRuanganListGlobal,
122.         sizeof(Ruangan) * jumlahRuangBaru)
123.     if ruanganListBaru = NULL and jumlahRuangBaru > 0 then
124.         output "Gagal mengubah ukuran denah (realloc gagal)!"
125.         return FALSE
126.     end if
127.
128.     *ptrToRuanganListGlobal ← ruanganListBaru
129.
130.     if jumlahRuangBaru > jumlahRuangLama then
131.         for i ← jumlahRuangLama to jumlahRuangBaru - 1 do
132.             if *ptrToRuanganListGlobal ≠ NULL then
133.                 (*ptrToRuanganListGlobal)[i].kapasitas ←
134.                     MAX_PASIEN_RUANGAN
135.                 (*ptrToRuanganListGlobal)[i].dokter[0] ← '\0'
136.                 (*ptrToRuanganListGlobal)[i].jumlahPasien ← 0
137.                 for j ← 0 to MAX_PASIEN_RUANGAN - 1 do
138.                     (*ptrToRuanganListGlobal)[i].pasien[j][0] ← '\0'
139.                 end for
140.             end if
141.         end for
142.     end if
143.
144.     *ptrJumlahRuangTotal ← jumlahRuangBaru
145.     *ptrDenahBarisMain ← barisBaru
146.     *ptrDenahKolomMain ← kolomBaru
147.
148.     output "Denah rumah sakit berhasil diubah menjadi ", barisBaru, "
149.     baris dan ", kolomBaru, " kolom."
150.     return TRUE

```



```

151.     Procedure pindahDokter(kodeRuanganAsal: string, kodeRuanganTujuan:
        string, ruanganListFromParam: array of Ruangan, denahBarisAktual:
        integer, denahKolomAktual: integer)
152.     { I.S. Kode ruangan asal dan tujuan serta denah ruangan sudah

```



```

153.     terdefinisi. }
154. { F.S. Memindahkan dokter dari satu ruangan ke ruangan lain. }
155.
156. KAMUS
157.     asalBarisIdx : integer
158.     asalKolomAngka : integer
159.     asalKolomIdx : integer
160.     tujuanBarisIdx : integer
161.     tujuanKolomAngka : integer
162.     tujuanKolomIdx : integer
163.     asalIdx : integer
164.     tujuanIdx : integer
165.     namaDokterYangPindah : string
166.
167. ALGORITMA
168.     if kodeRuanganAsal = NULL or kodeRuanganTujuan = NULL or
169.     strlen(kodeRuanganAsal) < 2 or strlen(kodeRuanganTujuan) < 2 then
170.         output "Format kode ruangan asal atau tujuan tidak valid."
171.         return FALSE
172.     end if
173.
174.     asalBarisIdx ← kodeRuanganAsal[0] - 'A'
175.     asalKolomAngka ← atoi(kodeRuanganAsal[1])
176.     asalKolomIdx ← asalKolomAngka - 1
177.
178.     tujuanBarisIdx ← kodeRuanganTujuan[0] - 'A'
179.     tujuanKolomAngka ← atoi(kodeRuanganTujuan[1])
180.     tujuanKolomIdx ← tujuanKolomAngka - 1
181.
182.     if (asalBarisIdx < 0 or asalBarisIdx ≥ denahBarisAktual or
183.     asalKolomAngka < 1 or asalKolomAngka > denahKolomAktual) or
184.     (tujuanBarisIdx < 0 or tujuanBarisIdx ≥ denahBarisAktual or
185.     tujuanKolomAngka < 1 or tujuanKolomAngka > denahKolomAktual) then
186.         output "Kode ruangan di luar batas denah."
187.         return FALSE
188.     end if
189.
190.     asalIdx ← asalBarisIdx * denahKolomAktual + asalKolomIdx
191.     tujuanIdx ← tujuanBarisIdx * denahKolomAktual + tujuanKolomIdx
192.
193.     if ruanganListFromParam = NULL or asalIdx < 0 or asalIdx ≥
194.     (denahBarisAktual * denahKolomAktual) or
195.     tujuanIdx < 0 or tujuanIdx ≥ (denahBarisAktual *
196.     denahKolomAktual) then
197.         output "Indeks ruangan tidak valid atau denah belum
198.         diinisialisasi!"
199.         return FALSE
200.     end if
201.
202.     if strlen(ruanganListFromParam[asalIdx].dokter) = 0 then
203.         output "Pemindahan gagal. Ruangan ", kodeRuanganAsal, "
204.         kosong."
205.         return FALSE
206.     end if

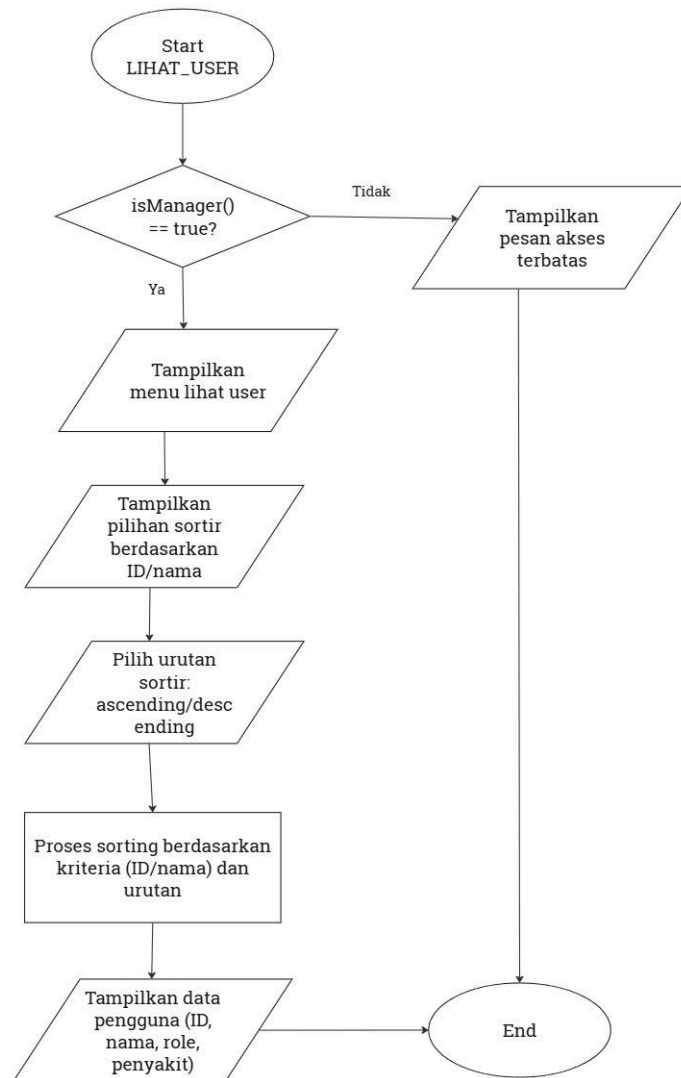
```

```

207.
208.     strncpy(namaDokterYangPindah,
209.     ruanganListFromParam[asalIdx].dokter, MAX_USERNAME_LEN - 1)
210.
211.     if strlen(ruanganListFromParam[tujuanIdx].dokter) > 0 then
212.         if strcmp(ruanganListFromParam[tujuanIdx].dokter,
213.         namaDokterYangPindah) = 0 then
214.             output "Dokter ", namaDokterYangPindah, " sudah berada di
215.             ruangan ", kodeRuanganTujuan, ". Tidak ada pemindahan."
216.             return TRUE
217.         end if
218.         output "Pemindahan gagal. Ruangan ", kodeRuanganTujuan, "
219.         sudah ditempati."
220.         return FALSE
221.     end if
222.
223.     ruanganListFromParam[asalIdx].dokter[0] ← '\0'
224.     strncpy(ruanganListFromParam[tujuanIdx].dokter,
225.     namaDokterYangPindah, MAX_USERNAME_LEN - 1)
226.     ruanganListFromParam[tujuanIdx].dokter[MAX_USERNAME_LEN - 1] ←
227.     '\0'
228.
229.     output "Dr. ", namaDokterYangPindah, " berhasil dipindahkan dari
230.     ruangan ", kodeRuanganAsal, " ke ruangan ", kodeRuanganTujuan,
231.     "."
232.     return TRUE

```

- **F07 - Lihat User**



Gambar 2.2.2.8 *Flowchart* Lihat User

```

1. Procedure isCurrentUserAManager()
2. { I.S. Status login dan role pengguna sudah terdefinisi. }
3. { F.S. Mengembalikan TRUE jika pengguna yang sedang login adalah Manager,
4.   FALSE jika bukan. }
5.
6. KAMUS
7.   result : boolean
8.
9. ALGORITMA
10.  if isLoggedIn = TRUE and strcmp(currentUser.role, "Manager") = 0 then
11.    result ← TRUE
12.  else
13.    result ← FALSE
14.  end if
  
```

```
15.     return result
```

```
1. Procedure kriteriasort(var method: SortMethod, var order: SortOrder,
   aurasort: boolean)
2. { I.S. method dan order belum terdefinisi; aurasort adalah apakah "aura"
   perlu ditampilkan }
3. { F.S. method dan order terisi sesuai input pengguna }
4.
5. KAMUS
6.     methodinput: integer
7.     orderinput: integer
8.
9.
10. ALGORITMA
11.     output "Urutkan berdasarkan?"
12.     output "1. ID"
13.     output "2. Nama"
14.     if aurasort then
15.         output "3. Aura"
16.     end if
17.     output ">>> Pilihan: "
18.     input methodinput
19.
20.     if aurasort and methodinput = 3 then
21.         method ← SORT_BY_AURA
22.     else if methodinput = 2 then
23.         method ← SORT_BY_NAME
24.     else
25.         method ← SORT_BY_ID
26.     end if
27.
28.     output newline + "Urutan sort?"
29.     output "1. Ascending (A to Z atau terkecil)"
30.     output "2. Descending (Z to A atau terbesar)"
31.     output ">>> Pilihan: "
32.     input orderinput
33.
34.     if orderinput = 2 then
35.         order ← DESC
36.     else
37.         order ← ASC
38.     end if
```

```
1. Procedure swapUsers(a: pointer to User, b: pointer to User)
2. { I.S. Dua pengguna yang ingin ditukar posisi sudah terdefinisi. }
3. { F.S. Posisi dua pengguna di dalam array telah ditukar. }
4.
5. KAMUS
6.     temp : User
7.
8. ALGORITMA
9.     temp ← *a
```

```

10.     *a ← *b
11.     *b ← temp

```

```

1. Procedure sortUsers(arr: array of User, n: integer, method: SortMethod,
   order: SortOrder)
2. { I.S. Array pengguna dan metode serta urutan pengurutan sudah
3.   terdefinisi. }
4. { F.S. Array pengguna terurut sesuai dengan kriteria yang diberikan. }
5.
6.   KAMUS
7.     i, j : integer
8.     harusSwap : boolean
9.
10.  ALGORITMA
11.    for i ← 0 to n - 2 do
12.      for j ← 0 to n - i - 2 do
13.        harusSwap ← FALSE
14.
15.        if order = ASC then
16.          if method = SORT_BY_ID and arr[j].id > arr[j+1].id then
17.            harusSwap ← TRUE
18.          else if method = SORT_BY_NAME and
19.            strcmp(arr[j].username, arr[j+1].username) > 0 then
20.            harusSwap ← TRUE
21.          else if method = SORT_BY_AURA and arr[j].aura >
22.            arr[j+1].aura then
23.            harusSwap ← TRUE
24.          end if
25.        else
26.          if method = SORT_BY_ID and arr[j].id < arr[j+1].id then
27.            harusSwap ← TRUE
28.          else if method = SORT_BY_NAME and
29.            strcmp(arr[j].username, arr[j+1].username) < 0 then
30.            harusSwap ← TRUE
31.          else if method = SORT_BY_AURA and arr[j].aura <
32.            arr[j+1].aura then
33.            harusSwap ← TRUE
34.          end if
35.        end if
36.
37.        if harusSwap = TRUE then
38.          swapUsers(&arr[j], &arr[j+1])
39.        end if
40.      end for
41.    end for

```

```

42. Procedure mintaKriteriaSort(method: pointer to SortMethod, order: pointer
   to SortOrder, bisaSortByAura: boolean)
43. { I.S. Pengguna ingin memilih kriteria pengurutan dan urutan. }

```

```

44. { F.S. Menentukan metode pengurutan dan urutan berdasarkan input
45.   pengguna. }
46.
47.   KAMUS
48.     methodInput, orderInput : integer
49.
50.   ALGORITMA
51.     output "Urutkan berdasarkan?\n1. ID\n2. Nama\n"
52.     if bisaSortByAura = TRUE then
53.       output "3. Aura\n"
54.     end if
55.     output ">>> Pilihan: "
56.     if scanf("%d", &methodInput) ≠ 1 or (methodInput < 1 or
57.       (bisaSortByAura = TRUE and methodInput > 2)) then
58.       output "Pilihan invalid!\n"
59.       return FALSE
60.     end if
61.
62.     if methodInput = 2 then
63.       *method ← SORT_BY_NAME
64.     else
65.       *method ← SORT_BY_ID
66.     end if
67.
68.     output "\nUrutan sort?\n1. ASC (A-Z / Terkecil)\n2. DESC (Z-A /
69.     Terbesar)\n>>> Pilihan: "
70.     if scanf("%d", &orderInput) ≠ 1 or (orderInput ≠ 1 and orderInput ≠
71.     2) then
72.       output "Pilihan invalid!\n"
73.       return FALSE
74.     end if
75.
76.     if orderInput = 2 then
77.       *order ← DESC
78.     else
79.       *order ← ASC
80.     end if
81.     return TRUE

```

```

82. Procedure lihatsemuaUser(managerData: User, dokterListGlobal: array of
    User, dokterCountGlobal: integer, pasienListGlobal: array of User,
    pasienCountGlobal: integer)
83. { I.S. Manager sudah login dan data pengguna tersedia. }
84. { F.S. Menampilkan semua pengguna (Manager, Dokter, Pasien) yang terurut
85.   berdasarkan kriteria. }
86.
87.   KAMUS
88.     method : SortMethod
89.     order : SortOrder
90.     totalUser : integer
91.     semuaUserListTemp : array of User
92.     i : integer

```

```

93.
94.  ALGORITMA
95.    if not isCurrentUserAManager() then
96.      output "Fitur ini hanya untuk Manager."
97.      return
98.    end if
99.
100.    if not mintaKriteriaSort(&method, &order, FALSE) then
101.      return
102.    end if
103.
104.    totalUser ← 1 + dokterCountGlobal + pasienCountGlobal
105.
106.    if totalUser > MAX_JUMLAH_USER then
107.      output "Jumlah pengguna (%d) melebihi kapasitas array
108.      sementara (%d).\n", totalUser, MAX_JUMLAH_USER
109.      return
110.    end if
111.
112.    semuaUserListTemp[0] ← managerData
113.    for i ← 0 to dokterCountGlobal - 1 do
114.      semuaUserListTemp[i + 1] ← dokterListGlobal[i]
115.    end for
116.    for i ← 0 to pasienCountGlobal - 1 do
117.      semuaUserListTemp[i + 1 + dokterCountGlobal] ←
118.      pasienListGlobal[i]
119.    end for
120.
121.    sortUsers(semuaUserListTemp, totalUser, method, order)
122.
123.    output "\nMenampilkan semua pengguna dengan ", if(method =
124.    SORT_BY_NAME) then "nama" else "ID", " terurut ", if(order =
125.    DESC) then "descending" else "ascending", "... \n"
126.
127.    output "ID      | Nama          | Role      | Penyakit\n"
128.    output "-----\n"
129.
130.    for i ← 0 to totalUser - 1 do
131.      output "%-4d | %-10s | %-8s | %s\n", semuaUserListTemp[i].id,
132.      semuaUserListTemp[i].username, semuaUserListTemp[i].role,
133.      if(strcmp(semuaUserListTemp[i].role, "Pasien") = 0 and
134.      strlen(semuaUserListTemp[i].namaPenyakit) > 0) then
135.      semuaUserListTemp[i].namaPenyakit else "-"
136.    end for

```

```

137.  Procedure lihatPasien(pasienListGlobal: array of User,
138.  pasienCountGlobal: integer)
139.  { I.S. Data pasien tersedia. }
140.  { F.S. Menampilkan daftar pasien yang terdaftar. }
141.
142.  KAMUS

```

```

143.     method : SortMethod
144.     order : SortOrder
145.
146.     ALGORITMA
147.         if not isCurrentUserAManager() then
148.             output "Fitur ini hanya untuk Manager."
149.             return
150.         end if
151.
152.         if pasienCountGlobal = 0 then
153.             output "Tidak ada pasien terdaftar."
154.             return
155.         end if
156.
157.         if not mintaKriteriaSort(&method, &order, FALSE) then
158.             return
159.         end if
160.
161.         if pasienCountGlobal > MAX_JUMLAH_USER then
162.             return
163.         end if
164.
165.         User tempPasienList[MAX_JUMLAH_USER]
166.         for i ← 0 to pasienCountGlobal - 1 do
167.             tempPasienList[i] ← pasienListGlobal[i]
168.         end for
169.
170.         sortUsers(tempPasienList, pasienCountGlobal, method, order)
171.
172.         output "\nMenampilkan pasien dengan ", if(method = SORT_BY_NAME)
173.         then "nama" else "ID", " terurut ", if(order = DESC) then
174.         "descending" else "ascending", "... \n"
175.
176.         output "ID      | Nama          | Penyakit\n"
177.         output "-----\n"
178.
179.         for i ← 0 to pasienCountGlobal - 1 do
180.             output "%-4d | %-10s | %s\n", tempPasienList[i].id,
181.             tempPasienList[i].username,
182.             if(strlen(tempPasienList[i].namaPenyakit) > 0) then
183.             tempPasienList[i].namaPenyakit else "-"
184.         end for

```

```

185.     Procedure lihatDokter(dokterListGlobal: array of User,
186.     dokterCountGlobal: integer)
187.     { I.S. Data dokter tersedia. }
188.     { F.S. Menampilkan daftar dokter yang terdaftar. }
189.
190.     KAMUS
191.         method : SortMethod
192.         order : SortOrder

```

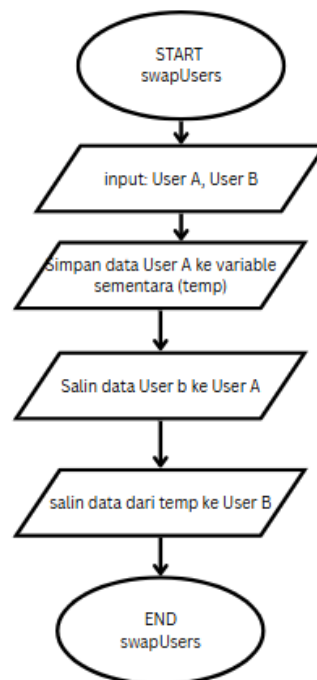
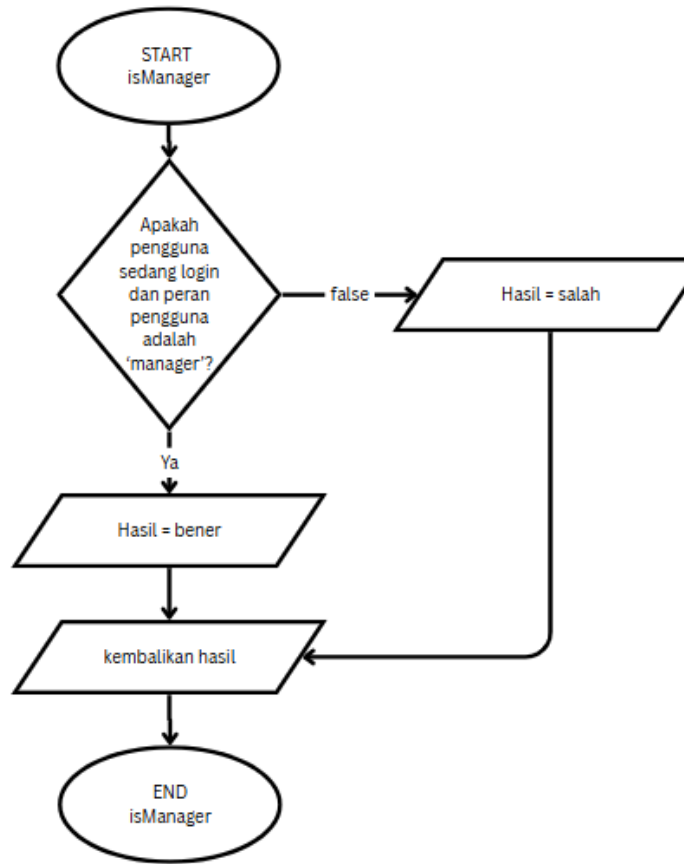


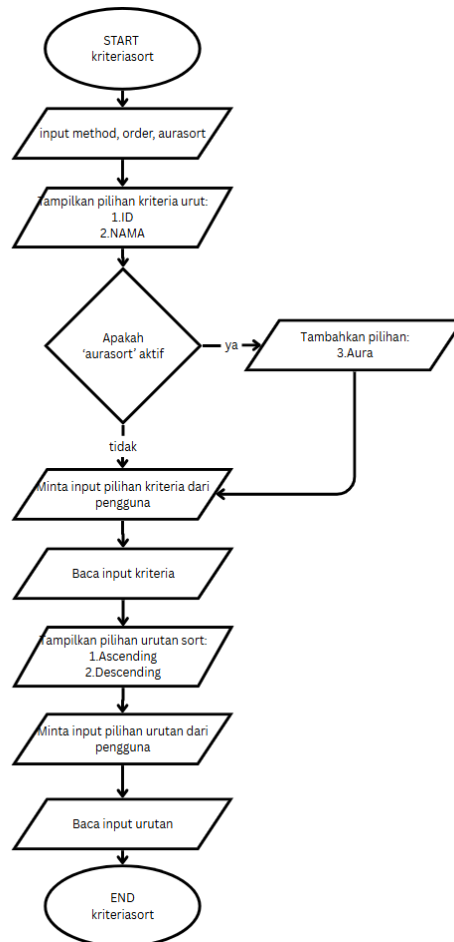
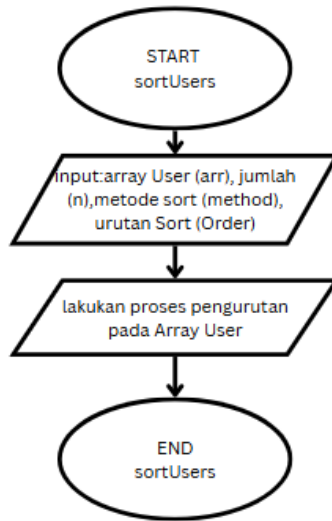
```

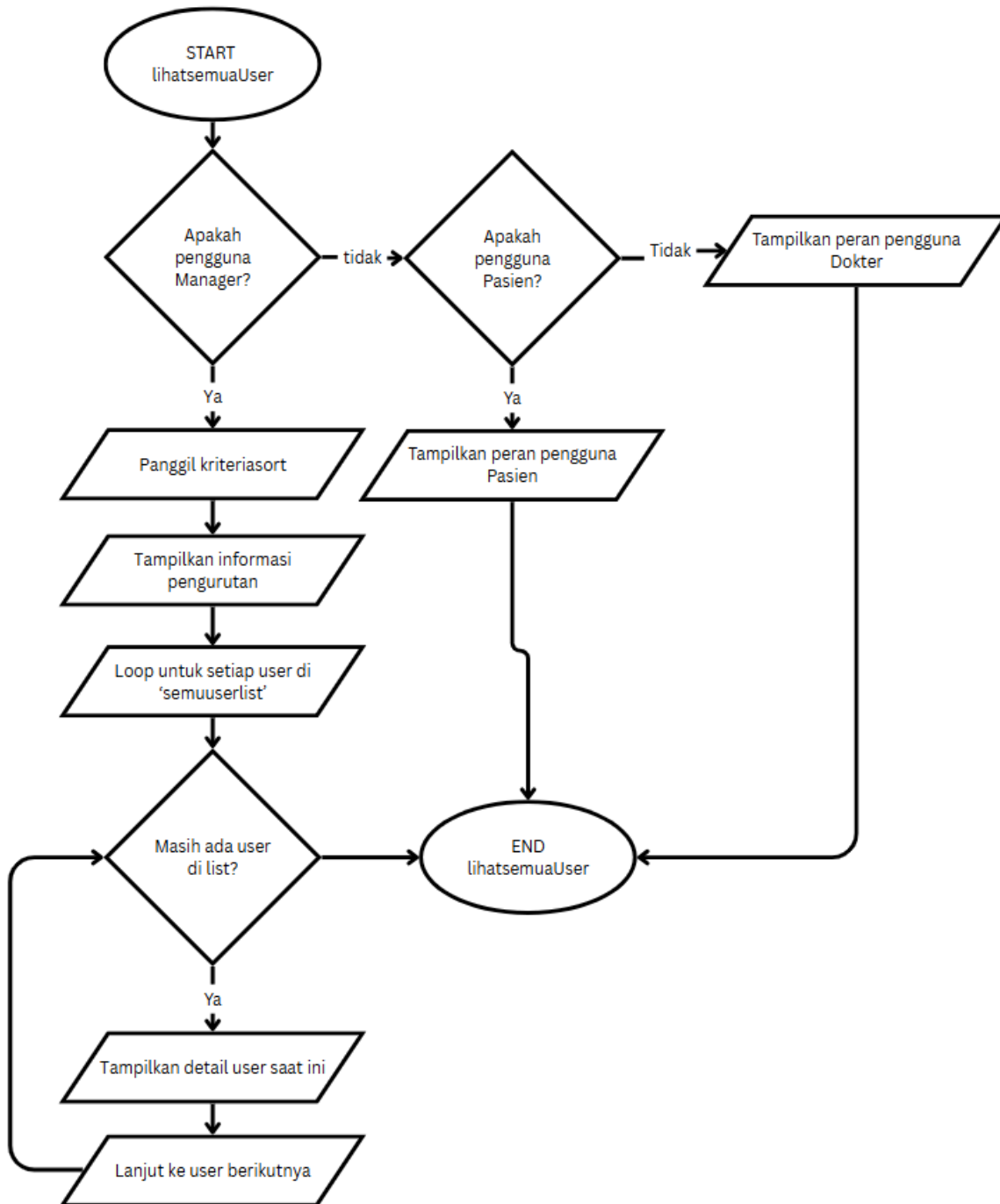
193.  ALGORITMA
194.      if not isCurrentUserAManager() then
195.          output "Fitur ini hanya untuk Manager."
196.          return
197.      end if
198.
199.      if dokterCountGlobal = 0 then
200.          output "Tidak ada dokter terdaftar."
201.          return
202.      end if
203.
204.      if not mintaKriteriaSort(&method, &order, TRUE) then
205.          return
206.      end if
207.
208.      if dokterCountGlobal > MAX_JUMLAH_USER then
209.          return
210.      end if
211.
212.      User tempDokterList[MAX_JUMLAH_USER]
213.      for i ← 0 to dokterCountGlobal - 1 do
214.          tempDokterList[i] ← dokterListGlobal[i]
215.      end for
216.
217.      sortUsers(tempDokterList, dokterCountGlobal, method, order)
218.
219.      output "\nMenampilkan dokter dengan ", if(method = SORT_BY_AURA)
220.      then "aura" else if(method = SORT_BY_NAME) then "nama" else "ID",
221.      " terurut ", if(order = DESC) then "descending" else "ascending",
222.      "... \n"
223.
224.      output "ID      | Nama          | Aura\n"
225.      output "-----\n"
226.
227.      for i ← 0 to dokterCountGlobal - 1 do
228.          output "%-4d | %-10s | %d\n", tempDokterList[i].id,
229.          tempDokterList[i].username, tempDokterList[i].aura
230.      end for

```

- **F08 - Cari User**







Gambar 2.2.2.9 Flowchart Cari User

```

1. Procedure clearInputBufferCariUser()
2. { I.S. Input buffer berisi karakter ekstra. }
3. { F.S. Input buffer dibersihkan dari karakter hingga newline atau EOF. }
4.

```

```

5.   KAMUS
6.     c : integer
7.
8.   ALGORITMA
9.     while (c ← getchar()) ≠ '\n' and c ≠ EOF do
10.      # Mengabaikan karakter hingga menemui newline atau EOF
11.    end while
12.
13.
14. Procedure printHeaderUserTable()
15. { I.S. Header untuk tabel pengguna. }
16. { F.S. Menampilkan header tabel pengguna. }
17.
18. ALGORITMA
19.   output "ID      | Username      | Role      | Penyakit\n"
20.   output "-----\n"
21.
22.
23. Procedure printUserDetail(user: User)
24. { I.S. Data pengguna sudah terdefinisi. }
25. { F.S. Menampilkan detail pengguna dalam format tabel. }
26.
27. ALGORITMA
28.   output "%-4d | %-10s | %-8s | %s\n", user.id, user.username,
29.   user.role,
30.   if(strcmp(user.role, "Pasien") = 0 and
31.   strlen(user.namaPenyakit) > 0) then user.namaPenyakit else "-"
32.
33.
34. Procedure printHeaderDokterTable()
35. { I.S. Header untuk tabel dokter. }
36. { F.S. Menampilkan header tabel dokter. }
37.
38. ALGORITMA
39.   output "\nID      | Username      | Aura\n"
40.   output "-----\n"
41. }
42.
43. Procedure printDokterDetail(dokter: User)
44. { I.S. Data dokter sudah terdefinisi. }
45. { F.S. Menampilkan detail dokter dalam format tabel. }
46.
47. ALGORITMA
48.   output "%-4d | %-10s | %d\n", dokter.id, dokter.username, dokter.aura
49.
50.
51. Procedure printHeaderPasienTable()
52. { I.S. Header untuk tabel pasien. }
53. { F.S. Menampilkan header tabel pasien. }
54.
55. ALGORITMA
56.   output "\nID      | Username      | Penyakit      | Nyawa | Dokter
57.   Perawat\n"
58.   output

```

```

59. "-----\n"
60.
61.
62. Procedure printPasienDetail(pasien: User)
63. { I.S. Data pasien sudah terdefinisi. }
64. { F.S. Menampilkan detail pasien dalam format tabel. }
65.
66.   ALGORITMA
67.     output "%-4d | %-10s | %-19s | %-5d | %s\n", pasien.id,
68.     pasien.username,
69.       if(strlen(pasien.namaPenyakit) > 0) then pasien.namaPenyakit
70.       else "-",
71.     pasien.nyawa,
72.       if(strlen(pasien.dokterPerawat) > 0) then pasien.dokterPerawat
73.       else "-"
74.
75.
76. Procedure binarySearchByID(sortedList: array of User, size: integer, id:
    integer)
77. { I.S. Daftar pengguna yang sudah terurut berdasarkan ID.}
78. { F.S. Mengembalikan indeks pengguna dengan ID yang dicari, atau -1 jika
79.   tidak ditemukan.}
80.
81.   KAMUS
82.     left, right, mid : integer
83.
84.   ALGORITMA
85.     left ← 0
86.     right ← size - 1
87.
88.     while left ≤ right do
89.       mid ← left + (right - left) / 2
90.       if sortedList[mid].id = id then
91.         return mid
92.       else if sortedList[mid].id < id then
93.         left ← mid + 1
94.       else
95.         right ← mid - 1
96.       end if
97.     end while
98.
99.     return -1
100.  }
101.
102. Procedure sequentialSearchByName(list: array of User, count: integer,
103. name: string)
104. { I.S. Daftar pengguna yang belum terurut. }
105. { F.S. Mengembalikan indeks pengguna dengan nama yang dicari, atau -1
106.   jika tidak ditemukan. }
107.
108.   KAMUS
109.     i : integer
110.
111.   ALGORITMA

```

```

112.     for i ← 0 to count - 1 do
113.         if strcmp(list[i].username, name) = 0 then
114.             return i
115.         end if
116.     end for
117.
118.     return -1
119. }
120.
121. Procedure cariUser(managerData: pointer to User, dokterListGlobal:
    array of User, dokterCountGlobal: integer, pasienListGlobal: array of
    User, pasienCountGlobal: integer)
122. { I.S. Data pengguna (Manager, Dokter, Pasien) sudah tersedia. }
123. { F.S. Menampilkan pengguna berdasarkan ID atau nama sesuai pilihan. }
124. }
125.
126. KAMUS
127.     pilihan, idInput : integer
128.     namaInput : string
129.     ditemukan : boolean
130.
131. ALGORITMA
132.     if strcmp(currentUser.role, "Manager") ≠ 0 then
133.         output "Perintah CARI_USER hanya bisa digunakan oleh
            Manager."
134.         return
135.     end if
136.
137.     output "Cari berdasarkan?\n1. ID\n2. Nama\n>>> Pilihan: "
138.     if scanf("%d", &pilihan) ≠ 1 then
139.         output "Input pilihan tidak valid. Masukkan angka (1 atau
            2). "
140.         clearInputBufferCariUser()
141.         return
142.     end if
143.     clearInputBufferCariUser()
144.
145.     if pilihan = 1 then
146.         output "\n>>> Masukkan nomor ID user: "
147.         if scanf("%d", &idInput) ≠ 1 then
148.             output "Input ID tidak valid. Masukkan angka."
149.             clearInputBufferCariUser()
150.             return
151.         end if
152.         clearInputBufferCariUser()
153.
154.         output "\nMenampilkan pengguna dengan nomor ID ", idInput,
            "... \n"
155.         if managerData.id = idInput then
156.             printHeaderUserTable()
157.             printUserDetail(managerData)
158.             ditemukan ← TRUE
159.         end if
160.

```

```

161.         if !ditemukan then
162.             index ← binarySearchByID(dokterListGlobal,
            dokterCountGlobal, idInput)
163.             if index ≠ -1 then
164.                 printHeaderUserTable()
165.                 printUserDetail(dokterListGlobal[index])
166.                 ditemukan ← TRUE
167.             end if
168.         end if
169.
170.         if !ditemukan then
171.             User tempPasienList[MAX_JUMLAH_USER]
172.             for i ← 0 to pasienCountGlobal - 1 do
173.                 tempPasienList[i] ← pasienListGlobal[i]
174.             end for
175.             sortUsers(tempPasienList, pasienCountGlobal, SORT_BY_ID,
            ASC)
176.             index ← binarySearchByID(tempPasienList,
            pasienCountGlobal, idInput)
177.             if index ≠ -1 then
178.                 printHeaderUserTable()
179.                 printUserDetail(tempPasienList[index])
180.                 ditemukan ← TRUE
181.             end if
182.         end if
183.         else if pilihan = 2 then
184.             output "\n>>> Masukkan nama user: "
185.             if scanf("%19s", namaInput) ≠ 1 then
186.                 output "Input nama gagal."
187.                 clearInputBufferCariUser()
188.                 return
189.             end if
190.             if strlen(namaInput) ≥ MAX_USERNAME_LEN then
191.                 output "Nama user terlalu panjang! Maksimum ",
            MAX_USERNAME_LEN - 1, " karakter."
192.                 clearInputBufferCariUser()
193.                 return
194.             end if
195.             clearInputBufferCariUser()
196.
197.             userDitemukanDiCetak ← FALSE
198.             if strcmp(managerData.username, namaInput) = 0 then
199.                 if !userDitemukanDiCetak then
200.                     printf("\nMenampilkan pengguna dengan nama %s...\n",
            namaInput)
201.                     printHeaderUserTable()
202.                     userDitemukanDiCetak ← TRUE
203.                 end if
204.                 printUserDetail(managerData)
205.                 ditemukan ← TRUE
206.             end if
207.
208.             if !ditemukan then

```



```

209.         index ← sequentialSearchByName(dokterListGlobal,
        dokterCountGlobal, namaInput)
210.         if index ≠ -1 then
211.             if !userDitemukanDiCetak then
212.                 printf("\nMenampilkan pengguna dengan nama
        %s...\n", namaInput)
213.                 printHeaderUserTable()
214.                 userDitemukanDiCetak ← TRUE
215.             end if
216.             printUserDetail(dokterListGlobal[index])
217.             ditemukan ← TRUE
218.         end if
219.     end if
220.
221.     if !ditemukan then
222.         index ← sequentialSearchByName(pasienListGlobal,
        pasienCountGlobal, namaInput)
223.         if index ≠ -1 then
224.             if !userDitemukanDiCetak then
225.                 printf("\nMenampilkan pengguna dengan nama
        %s...\n", namaInput)
226.                 printHeaderUserTable()
227.                 userDitemukanDiCetak ← TRUE
228.             end if
229.             printUserDetail(pasienListGlobal[index])
230.             ditemukan ← TRUE
231.         end if
232.     end if
233. else
234.     output "Pilihan tidak valid!"
235.     return
236. end if
237.
238. if !ditemukan then
239.     if pilihan = 1 then
240.         output "\nTidak ditemukan pengguna dengan ID tersebut."
241.     else if pilihan = 2 then
242.         output "\nTidak ditemukan pengguna dengan nama ",
        namaInput, "!"
243.     end if
244. end if
245. }
246.
247. Procedure cariDokter(dokterListGlobal: array of User,
        dokterCountGlobal: integer)
248. { I.S. Data dokter sudah tersedia. }
249. { F.S. Menampilkan dokter berdasarkan ID atau nama sesuai pilihan.}
250.
251.     KAMUS
252.         pilihan : integer
253.         namaInput : string
254.         foundIndex : integer
255.
256.     ALGORITMA

```

```

257.     if strcmp(currentUser.role, "Manager") ≠ 0 then
258.         output "Perintah CARI_DOKTER hanya untuk Manager."
259.         return
260.     end if
261.
262.     if dokterCountGlobal = 0 then
263.         output "Belum ada dokter terdaftar."
264.         return
265.     end if
266.
267.     output "Cari dokter berdasarkan?\n1. ID\n2. Nama\n>>> Pilihan: "
268.     if scanf("%d", &pilihan) ≠ 1 then
269.         output "Input pilihan tidak valid."
270.         clearInputBufferCariUser()
271.         return
272.     end if
273.     clearInputBufferCariUser()
274.
275.     if pilihan = 1 then
276.         output "\n>>> Masukkan ID dokter: "
277.         if scanf("%d", &idInput) ≠ 1 then
278.             output "Input ID tidak valid."
279.             clearInputBufferCariUser()
280.             return
281.         end if
282.         clearInputBufferCariUser()
283.         output "\nMenampilkan dokter dengan ID ", idInput, "... \n"
284.         foundIndex ← binarySearchByID(dokterListGlobal,
            dokterCountGlobal, idInput)
285.     else if pilihan = 2 then
286.         output "\n>>> Masukkan nama dokter: "
287.         if scanf("%19s", namaInput) ≠ 1 then
288.             output "Input nama gagal."
289.             clearInputBufferCariUser()
290.             return
291.         end if
292.         if strlen(namaInput) ≥ MAX_USERNAME_LEN then
293.             output "Nama dokter terlalu panjang!"
294.             clearInputBufferCariUser()
295.             return
296.         end if
297.         clearInputBufferCariUser()
298.         output "\nMenampilkan dokter dengan nama ", namaInput,
            "... \n"
299.         foundIndex ← sequentialSearchByName(dokterListGlobal,
            dokterCountGlobal, namaInput)
300.     else
301.         output "Pilihan tidak valid!"
302.         return
303.     end if
304.
305.     if foundIndex ≠ -1 then
306.         printHeaderDokterTable()
307.         printDokterDetail(dokterListGlobal[foundIndex])

```

```

308.         else
309.             if pilihan = 1 then
310.                 output "\nTidak ditemukan dokter dengan ID tersebut."
311.             else if pilihan = 2 then
312.                 output "\nTidak ditemukan dokter dengan nama ",
namaInput, "."
313.             end if
314.         end if
315.     }
316.
317. Procedure cariPasien(pasienListGlobal: array of User,
pasienCountGlobal: integer)
318. { I.S. Data pasien sudah tersedia. }
319. { F.S. Menampilkan pasien berdasarkan ID, nama, atau penyakit sesuai
320.     pilihan. }
321.
322. KAMUS
323.     pilihan : integer
324.     namaInput : string
325.     penyakitInput : string
326.     ditemukan : boolean
327.
328. ALGORITMA
329.     if strcmp(currentUser.role, "Manager") ≠ 0 then
330.         output "Perintah CARI_PASIEN hanya untuk Manager."
331.         return
332.     end if
333.
334.     if pasienCountGlobal = 0 then
335.         output "Belum ada pasien terdaftar."
336.         return
337.     end if
338.
339.     output "Cari pasien berdasarkan?\n1. ID\n2. Nama\n3.
Penyakit\nn>>> Pilihan: "
340.     if scanf("%d", &pilihan) ≠ 1 then
341.         output "Input pilihan tidak valid."
342.         clearInputBufferCariUser()
343.         return
344.     end if
345.     clearInputBufferCariUser()
346.
347.     if pilihan = 1 then
348.         output "\nn>>> Masukkan ID pasien: "
349.         if scanf("%d", &idInput) ≠ 1 then
350.             output "Input ID tidak valid."
351.             clearInputBufferCariUser()
352.             return
353.         end if
354.         clearInputBufferCariUser()
355.
356.         if pasienCountGlobal > 0 then
357.             User tempPasienList[MAX_JUMLAH_USER]
358.             for i ← 0 to pasienCountGlobal - 1 do

```

```

359.         tempPasienList[i] ← pasienListGlobal[i]
360.     end for
361.     sortUsers(tempPasienList, pasienCountGlobal, SORT_BY_ID,
    ASC)
362.     index ← binarySearchByID(tempPasienList,
    pasienCountGlobal, idInput)
363.     if index ≠ -1 then
364.         printf("\nMenampilkan pasien dengan ID %d...\n",
    idInput)
365.         printHeaderPasienTable()
366.         printPasienDetail(tempPasienList[index])
367.         ditemukan ← TRUE
368.     end if
369. end if
370. else if pilihan = 2 then
371.     output "\n>>> Masukkan nama pasien: "
372.     if scanf("%19s", namaInput) ≠ 1 then
373.         output "Input nama gagal."
374.         clearInputBufferCariUser()
375.         return
376.     end if
377.     if strlen(namaInput) ≥ MAX_USERNAME_LEN then
378.         output "Nama pasien terlalu panjang!"
379.         clearInputBufferCariUser()
380.         return
381.     end if
382.     clearInputBufferCariUser()
383.
384.     index ← sequentialSearchByName(pasienListGlobal,
    pasienCountGlobal, namaInput)
385.     if index ≠ -1 then
386.         printf("\nMenampilkan pasien dengan nama %s...\n",
    namaInput)
387.         printHeaderPasienTable()
388.         printPasienDetail(pasienListGlobal[index])
389.         ditemukan ← TRUE
390.     end if
391. else if pilihan = 3 then
392.     output "\n>>> Masukkan nama penyakit: "
393.     if scanf(" %29[^\n]", penyakitInput) ≠ 1 then
394.         output "Input nama penyakit gagal."
395.         clearInputBufferCariUser()
396.         return
397.     end if
398.     clearInputBufferCariUser()
399.
400.     User hasilPencarian[MAX_JUMLAH_USER]
401.     jumlahHasil ← 0
402.     for i ← 0 to pasienCountGlobal - 1 do
403.         if strcmp(pasienListGlobal[i].namaPenyakit,
    penyakitInput) = 0 then
404.             if jumlahHasil < MAX_JUMLAH_USER then
405.                 hasilPencarian[jumlahHasil++] ←
    pasienListGlobal[i]

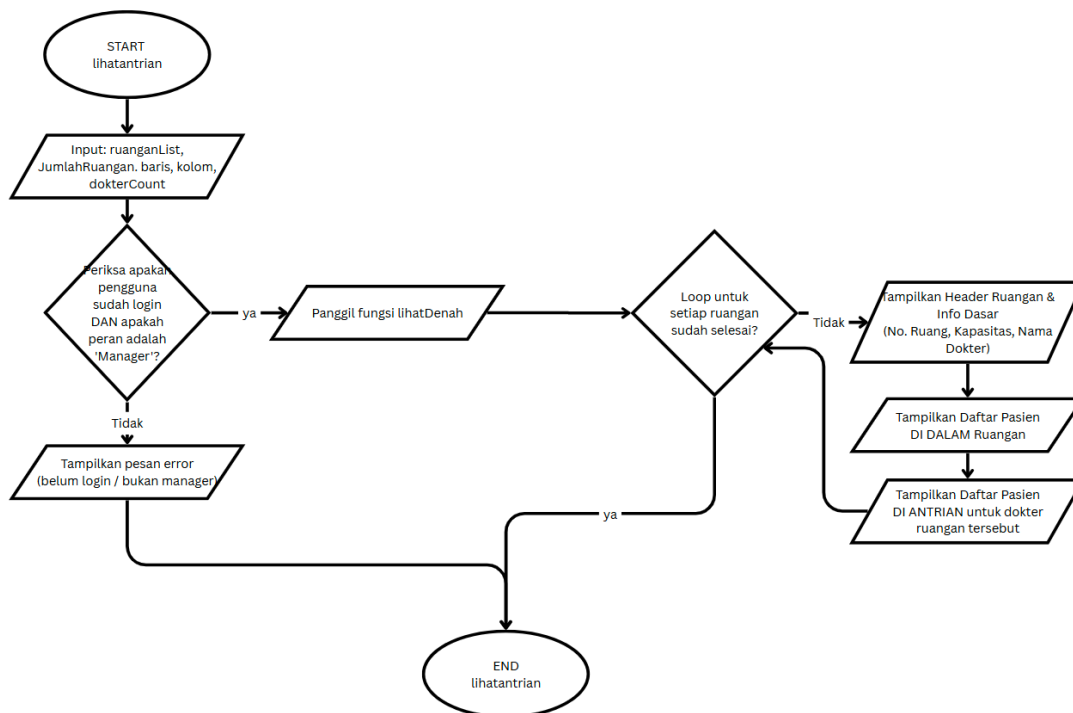
```

```

406.             end if
407.         end if
408.     end for
409.
410.     if jumlahHasil > 0 then
411.         ditemukan ← TRUE
412.         SortMethod method
413.         SortOrder order
414.         printf("Urutkan berdasarkan?\n1. ID\n2. Nama\n>>>
Pilihan: ")
415.         if scanf("%d", &methodInput) ≠ 1 then
416.             clearInputBufferCariUser()
417.             return
418.         end if
419.         clearInputBufferCariUser()
420.         method ← (methodInput = 2)

```

● F09 - Lihat Antrian



Gambar 2.2.2.10 Flowchart Lihat Antrian

```

1. Procedure lihatantrian(ruanganListDilempar: array of Ruangan,
   jumlahRuanganDilempar: integer, barisUntukDenah: integer,
   kolomUntukDenah: integer, dokterCountDilempar: integer)
2.
3. { I.S. Daftar ruangan, jumlah ruangan, baris, kolom denah, dan jumlah
   dokter sudah terdefinisi. }
4. { F.S. Menampilkan denah ruangan dan antrian pasien yang ada di setiap
   ruangan. }
5.
6.   KAMUS
7.     adaRuanganDitampilkan: boolean
8.     r: pointer to Ruangan
9.     kodeBaris: char
10.    nomorKolom: integer
11.    dq: pointer to AntrianDokter
12.    node: pointer to AntrianNode
13.    nomor: integer
14.
15.   ALGORITMA
16.     if !isLoggedIn then
17.       output "Harap melakukan login terlebih dahulu."
18.       return
19.     end if
20.
21.     if strcmp(currentUser.role, "Manager") ≠ 0 then
22.       output "Perintah LIHAT_ANTRIAN hanya dapat diakses oleh Manager."
23.       return
24.     end if
25.
26.     panggil lihatDenah(jumlahRuanganDilempar, barisUntukDenah,
   kolomUntukDenah)
27.     output "\n"
28.
29.     adaRuanganDitampilkan ← FALSE
30.
31.     for i ← 0 to jumlahRuanganDilempar - 1 do
32.       r ← &ruanganListDilempar[i]
33.
34.       if strlen(r.dokter) = 0 then
35.         continue
36.       end if
37.
38.       adaRuanganDitampilkan ← TRUE
39.       kodeBaris ← 'A' + (i / kolomUntukDenah)
40.       nomorKolom ← (i % kolomUntukDenah) + 1
41.
42.       output "===== Ruangan ", kodeBaris, nomorKolom, "
   =====\n"
43.       output "Kapasitas   : ", r.kapasitas
44.       output "Dokter      : Dr. ", r.dokter
45.
46.       output "Pasien di dalam ruangan:\n"
47.       if r.jumlahPasien = 0 then
48.         output "   Tidak ada pasien di dalam ruangan saat ini."
49.       else
50.         for j ← 0 to r.jumlahPasien - 1 do
51.           output "   ", j + 1, ". ", r.pasien[j]
52.         end for
53.       end if
54.
55.       output "Pasien di antrian:\n"
56.       dq ← NULL
57.       for j ← 0 to jumlahDokter - 1 do
58.         if dokterQueues[j].dokter ≠ NULL and
   strcmp(dokterQueues[j].dokter->username, r.dokter) = 0 then
59.           dq ← &dokterQueues[j]

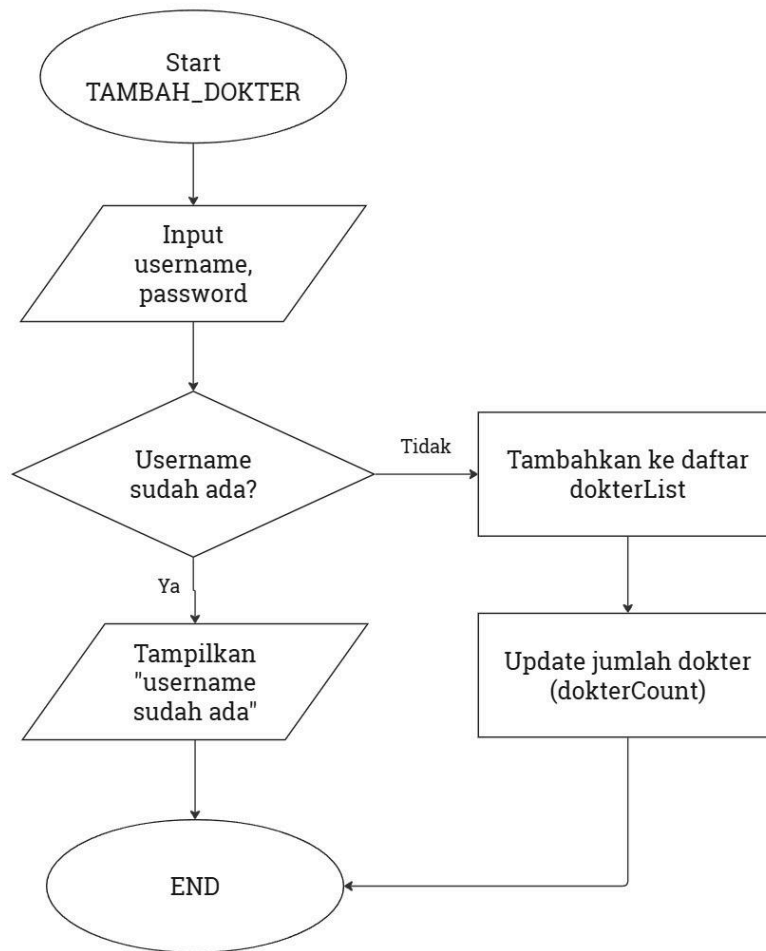
```

```

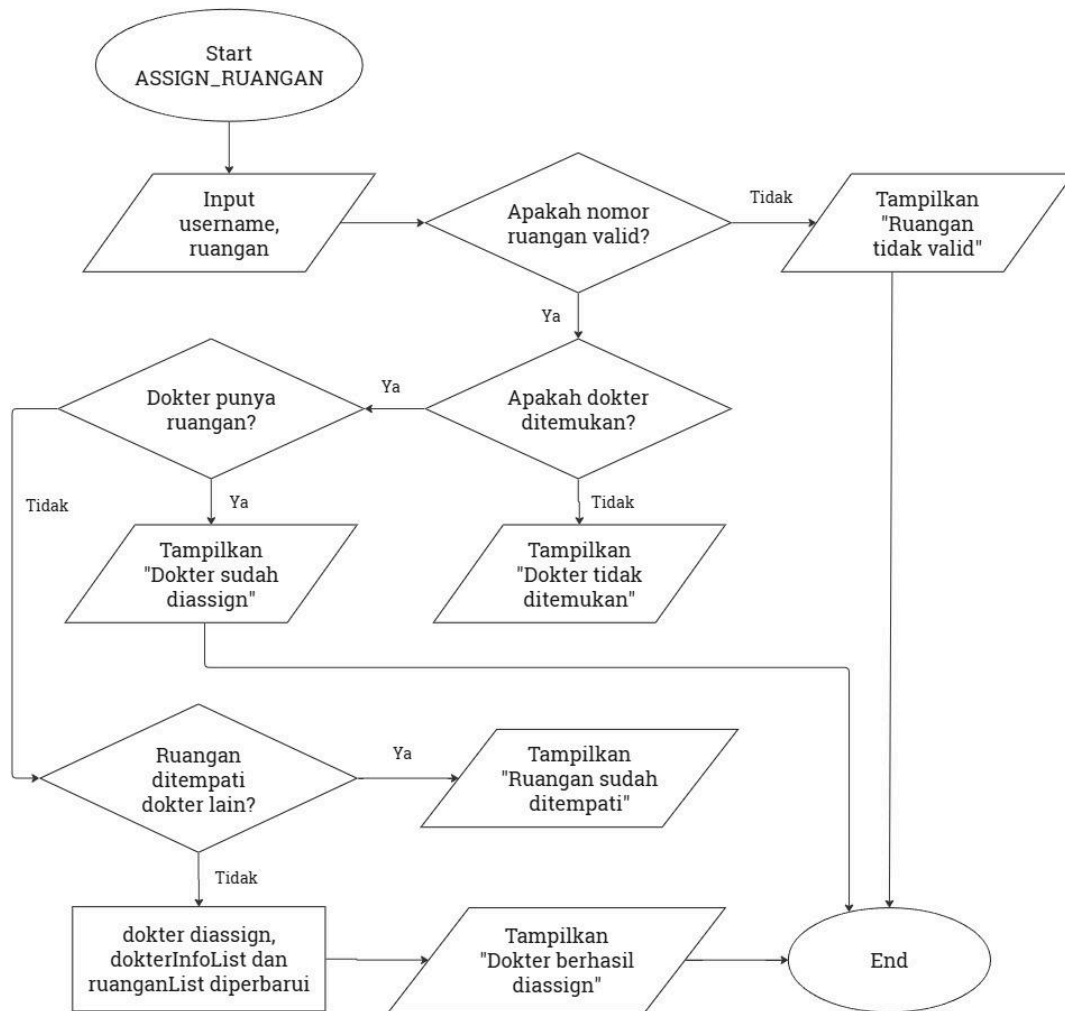
60.         break
61.     end if
62. end for
63.
64. if dq ≠ NULL and dq.head ≠ NULL then
65.     node ← dq.head
66.     nomor ← 1
67.     while node ≠ NULL do
68.         output " ", nomor, ". ", node.pasien_data.username
69.         node ← node.next
70.         nomor ← nomor + 1
71.     end while
72. else
73.     output " Tidak ada pasien di antrian saat ini."
74. end if
75. output "\n"
76. end for
77.
78. if !adaRuanganDitampilkan then
79.     output "Saat ini tidak ada ruangan yang memiliki dokter bertugas
atau tidak ada antrian."
80. end if

```

- **F10 - Tambah Dokter**



Gambar 2.2.2.10 *Flowchart* Tambah Dokter



Gambar 2.2.2.11 *Flowchart* Assign Dokter

```

1. Procedure clearInputBufferDokterMgmt()
2. { I.S. Input buffer berisi karakter ekstra. }
3. { F.S. Input buffer dibersihkan dari karakter hingga newline atau EOF. }
4.
5. KAMUS
6.   c : integer
7.
8. ALGORITMA
9.   while (c ← getchar()) ≠ '\n' and c ≠ EOF do
10.    # Mengabaikan karakter hingga menemui newline atau EOF
11.   end while
12. }
13.

```



```

14. Procedure tambahDokter(listDokter: array of User, currentTotalDokter:
    pointer to integer, usernameSetGlobal: pointer to Set)
15. { I.S. Kapasitas dokter belum tercapai dan data dokter masih kosong. }
16. { F.S. Dokter baru berhasil ditambahkan ke dalam sistem. }
17.
18. KAMUS
19.     tempUsername: string
20.     tempPassword: string
21.     idxBaru: integer
22.
23. ALGORITMA
24.     if *currentTotalDokter ≥ KAPASITAS_MAKSIMUM_DOKTER_DI_ARRAY then
25.         output "Kapasitas penyimpanan untuk dokter sudah penuh!"
26.         return
27.     end if
28.
29.     output "Username: "
30.     if scanf("%20s", tempUsername) ≠ 1 then
31.         output "Input username gagal. Harap coba lagi."
32.         clearInputBufferDokterMgmt()
33.         return
34.     end if
35.     if strlen(tempUsername) ≥ MAX_USERNAME_LEN then
36.         output "Username terlalu panjang!"
37.         clearInputBufferDokterMgmt()
38.         return
39.     end if
40.
41.     output "Password: "
42.     if scanf("%20s", tempPassword) ≠ 1 then
43.         output "Input password gagal. Harap coba lagi."
44.         clearInputBufferDokterMgmt()
45.         return
46.     end if
47.     if strlen(tempPassword) ≥ MAX_PASSWORD_LEN then
48.         output "Password terlalu panjang!"
49.         clearInputBufferDokterMgmt()
50.         return
51.     end if
52.
53.     if searchSet(*usernameSetGlobal, tempUsername) then
54.         output "Sudah ada Dokter bernama ", tempUsername
55.         return
56.     end if
57.
58.     idpenggunasebelum ← idpenggunasebelum + 1
59.     idxBaru ← *currentTotalDokter
60.
61.     listDokter[idxBaru].id ← idpenggunasebelum
62.     strncpy(listDokter[idxBaru].username, tempUsername, MAX_USERNAME_LEN
        - 1)
63.     strncpy(listDokter[idxBaru].password, tempPassword, MAX_PASSWORD_LEN
        - 1)
64.     strcpy(listDokter[idxBaru].role, "Dokter")

```

```

65.
66.     listDokter[idxBaru].namaPenyakit[0] ← '\0'
67.     listDokter[idxBaru].aura ← 0
68.     listDokter[idxBaru].kodeRuangan[0] ← '\0'
69.     listDokter[idxBaru].dokterPerawat[0] ← '\0'
70.     listDokter[idxBaru].nyawa ← 0
71.     listDokter[idxBaru].suhuTubuh ← 0.0f
72.     listDokter[idxBaru].tekananDarahSistolik ← 0
73.     listDokter[idxBaru].tekananDarahDiastolik ← 0
74.     listDokter[idxBaru].detakJantung ← 0
75.     listDokter[idxBaru].saturasiOksigen ← 0.0f
76.     listDokter[idxBaru].kadarGulaDarah ← 0
77.     listDokter[idxBaru].beratBadan ← 0.0f
78.     listDokter[idxBaru].tinggiBadan ← 0.0f
79.     listDokter[idxBaru].kadarKolesterol ← 0
80.     listDokter[idxBaru].trombosit ← 0
81.
82.     insertSet(usernameSetGlobal, tempUsername)
83.     *currentTotalDokter ← *currentTotalDokter + 1
84.
85.     output "Dokter ", tempUsername, " berhasil ditambahkan ke sistem!"
86. }
87.
88. Procedure assignRuangan(listDokter: array of User, currentTotalDokter:
    integer, listRuangan: array of Ruangan, jumlahRuangGlobal: integer,
    denahBarisGlobal: integer, denahKolomGlobal: integer)
89. { I.S. Data dokter, ruangan, dan kapasitas ruangan sudah tersedia. }
90. { F.S. Dokter berhasil diassign ke ruangan yang valid. }
91.
92. KAMUS
93.     usernameDokter: string
94.     kodeRuangInput: string
95.     idxDokterDitemukan: integer
96.     barisRuangIdx, kolomRuangIdx, idxRuanganTarget: integer
97.     dokterSudahPunyaRuangan, ruanganTargetSudahDitempati: boolean
98.     namaDokterDiTarget, kodeRuanganDokterSaatIni: string
99.
100. ALGORITMA
101.     output "Username: "
102.     if scanf("%20s", usernameDokter) ≠ 1 then
103.         return
104.     end if
105.     if strlen(usernameDokter) ≥ MAX_USERNAME_LEN then
106.         return
107.     end if
108.     clearInputBufferDokterMgmt()
109.
110.     output "Ruangan: "
111.     if scanf("%3s", kodeRuangInput) ≠ 1 then
112.         return
113.     end if
114.     clearInputBufferDokterMgmt()
115.
116.     idxDokterDitemukan ← -1

```

```

117.     for i ← 0 to currentTotalDokter - 1 do
118.         if strcmp(listDokter[i].username, usernameDokter) = 0 then
119.             idxDokterDitemukan ← i
120.             break
121.         end if
122.     end for
123.     if idxDokterDitemukan = -1 then
124.         output "\nDokter dengan username '", usernameDokter, "' tidak
            ditemukan!"
125.         return
126.     end if
127.
128.     if !(kodeRuangInput[0] ≥ 'A' and kodeRuangInput[0] < 'A' +
        denahBarisGlobal) or !(kodeRuangInput[1] ≥ '1' and kodeRuangInput[1] <
        '1' + denahKolomGlobal) or kodeRuangInput[2] ≠ '\0' then
129.         output "\nFormat kode ruangan '", kodeRuangInput, "' salah
            atau di luar batas denah (A-", 'A' + denahBarisGlobal - 1, ", 1-",
            denahKolomGlobal, ")!"
130.         return
131.     end if
132.
133.     barisRuangIdx ← kodeRuangInput[0] - 'A'
134.     kolomRuangIdx ← kodeRuangInput[1] - '1'
135.     idxRuanganTarget ← barisRuangIdx * denahKolomGlobal +
        kolomRuangIdx
136.
137.     if idxRuanganTarget < 0 or idxRuanganTarget ≥ jumlahRuangGlobal
        then
138.         output "\nKode ruangan '", kodeRuangInput, "' menghasilkan
            indeks di luar batas yang valid!"
139.         return
140.     end if
141.
142.     dokterSudahPunyaRuangan ←
        (strlen(listDokter[idxDokterDitemukan].kodeRuang) > 0)
143.     ruanganTargetSudahDitempati ←
        (strlen(listRuangan[idxRuanganTarget].dokter) > 0)
144.     namaDokterDiTarget ← listRuangan[idxRuanganTarget].dokter
145.     kodeRuanganDokterSaatIni ←
        listDokter[idxDokterDitemukan].kodeRuang
146.
147.     if dokterSudahPunyaRuangan and ruanganTargetSudahDitempati then
148.         if strcmp(namaDokterDiTarget, usernameDokter) ≠ 0 then
149.             output "\nDokter ", usernameDokter, " sudah menempati
                ruangan ", kodeRuanganDokterSaatIni
150.             output "Ruangan ", kodeRuangInput, " juga sudah ditempati
                dokter ", namaDokterDiTarget
151.         else
152.             output "\nError: Dokter ", usernameDokter, " sudah di
                ruangan ", kodeRuangInput, ", namun juga tercatat punya ruangan ",
                kodeRuanganDokterSaatIni
153.         end if
154.         return
155.     end if

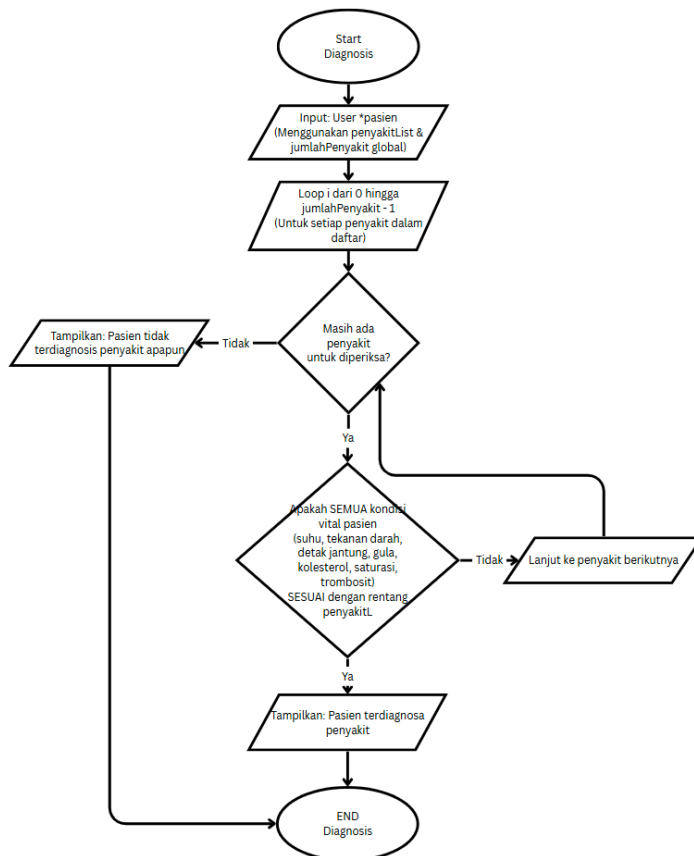
```

```

156.     if dokterSudahPunyaRuangan then
157.         output "\nDokter ", usernameDokter, " sudah diassign ke
           ruangan ", kodeRuanganDokterSaatIni
158.         return
159.     end if
160.     if ruanganTargetSudahDitempati then
161.         output "\nDokter ", namaDokterDiTarget, " sudah menempati
           ruangan ", kodeRuanganInput
162.         output "Silakan cari ruangan lain untuk dokter ",
           usernameDokter
163.         return
164.     end if
165.
166.     strncpy(listDokter[idxDokterDitemukan].kodeRuangan,
           kodeRuanganInput, 3)
167.     listDokter[idxDokterDitemukan].kodeRuangan[3] ← '\0'
168.
169.     strncpy(listRuangan[idxRuanganTarget].dokter, usernameDokter,
           MAX_USERNAME_LEN - 1)
170.     listRuangan[idxRuanganTarget].dokter[MAX_USERNAME_LEN - 1] ← '\0'
171.
172.     output "\nDokter ", usernameDokter, " berhasil diassign ke
           ruangan ", kodeRuanganInput

```

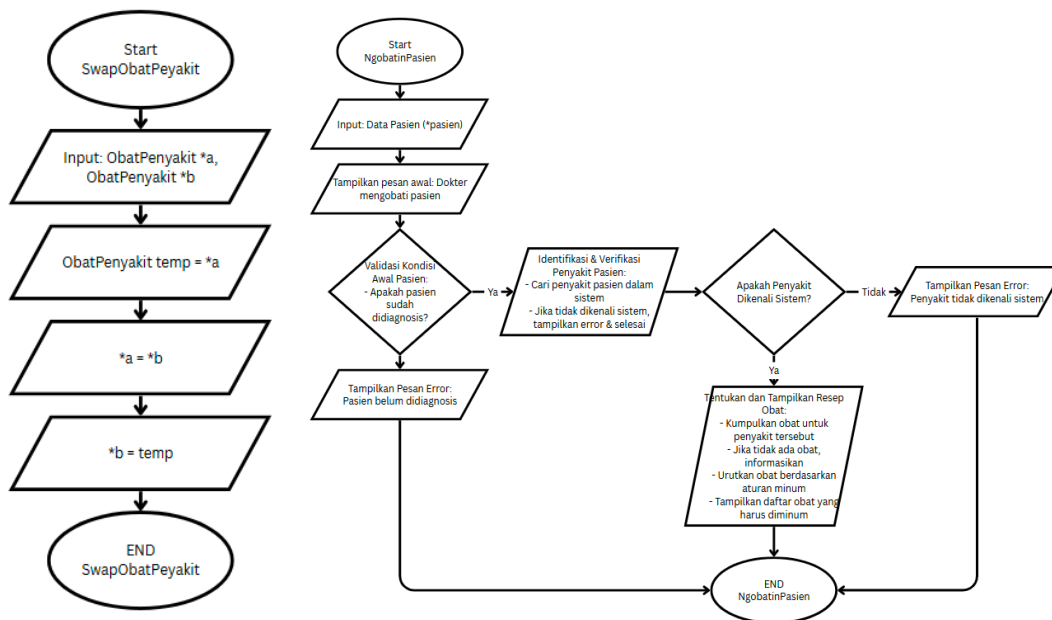
- F11 - Diagnosis



Gambar 2.2.2.12 *Flowchart* Diagnosis

```
1. Procedure diagnosisPasien(pasien: pointer to User)
2. { I.S. Data pasien yang mencakup gejala seperti suhu tubuh, tekanan
  darah, detak jantung, kadar gula darah, kadar kolesterol, saturasi
  oksigen, dan trombosit sudah terdefinisi. }
3. { F.S. Pasien terdiagnosis dengan penyakit yang sesuai jika kondisinya
  cocok dengan salah satu penyakit yang terdaftar, atau tidak terdiagnosis
  penyakit apapun. }
4.
5.   KAMUS
6.     i: integer
7.
8.   ALGORITMA
9.     for i ← 0 to jumlahPenyakit - 1 do
10.       if pasien.suhuTubuh ≥ penyakitList[i].suhuTubuhMin and
pasien.suhuTubuh ≤ penyakitList[i].suhuTubuhMax and
11.         pasien.tekananDarahSistolik ≥
pasakitList[i].tekananDarahSistolikMin and pasien.tekananDarahSistolik ≤
pasakitList[i].tekananDarahSistolikMax and
12.         pasien.tekananDarahDiastolik ≥
pasakitList[i].tekananDarahDiastolikMin and pasien.tekananDarahDiastolik
≤ pasakitList[i].tekananDarahDiastolikMax and
13.         pasien.detakJantung ≥ penyakitList[i].detakJantungMin and
pasien.detakJantung ≤ penyakitList[i].detakJantungMax and
14.         pasien.kadarGulaDarah ≥ penyakitList[i].kadarGulaDarahMin and
pasien.kadarGulaDarah ≤ penyakitList[i].kadarGulaDarahMax and
15.         pasien.kadarKolesterol ≥ penyakitList[i].kadarKolesterolMin
and pasien.kadarKolesterol ≤ penyakitList[i].kadarKolesterolMax and
16.         pasien.saturasiOksigen ≥ penyakitList[i].saturasiOksigenMin
and pasien.saturasiOksigen ≤ penyakitList[i].saturasiOksigenMax and
17.         pasien.trombosit ≥ penyakitList[i].trombositMin and
pasien.trombosit ≤ penyakitList[i].trombositMax then
18.           output pasien.username, " terdiagnosa penyakit ",
pasakitList[i].namaPenyakit, "!"
19.           pasien.namaPenyakit ← penyakitList[i].namaPenyakit
20.           return
21.         end if
22.       end for
23.
24.     output pasien.username, " tidak terdiagnosis penyakit apapun!"
25.     pasien.namaPenyakit[0] ← '\0'
```

- **F12 - Ngobatin**



Gambar 2.2.2.13 Flowchart Ngobatin

```

1. Procedure swapObatPenyakit(a: pointer to ObatPenyakit, b: pointer to
   ObatPenyakit)
2. { I.S. ObatPenyakit a dan b telah terdefinisi. }
3. { F.S. Posisi a dan b ditukar. }
4.
5.   KAMUS
6.     temp: ObatPenyakit
7.
8.   ALGORITMA
9.     temp ← *a
10.    *a ← *b
11.    *b ← temp
12. }
13.
14. Procedure ngobatinPasien(pasien: pointer to User)
15. { I.S. Pasien yang memiliki data penyakit dan gejala terkait sudah
    terdiagnosis. }
16. { F.S. Pasien diberikan resep obat sesuai dengan penyakit yang
    didiagnosis. }
17.
18.   KAMUS
19.     ditemukan: integer
20.     resepUntukPasien: array of ObatPenyakit
21.     countobat: integer
22.     namaObatDitemukan: string
23.
24.   ALGORITMA
25.     output "Dokter sedang mengobati pasien!"
26.
27.     if strlen(pasien->namaPenyakit) = 0 then

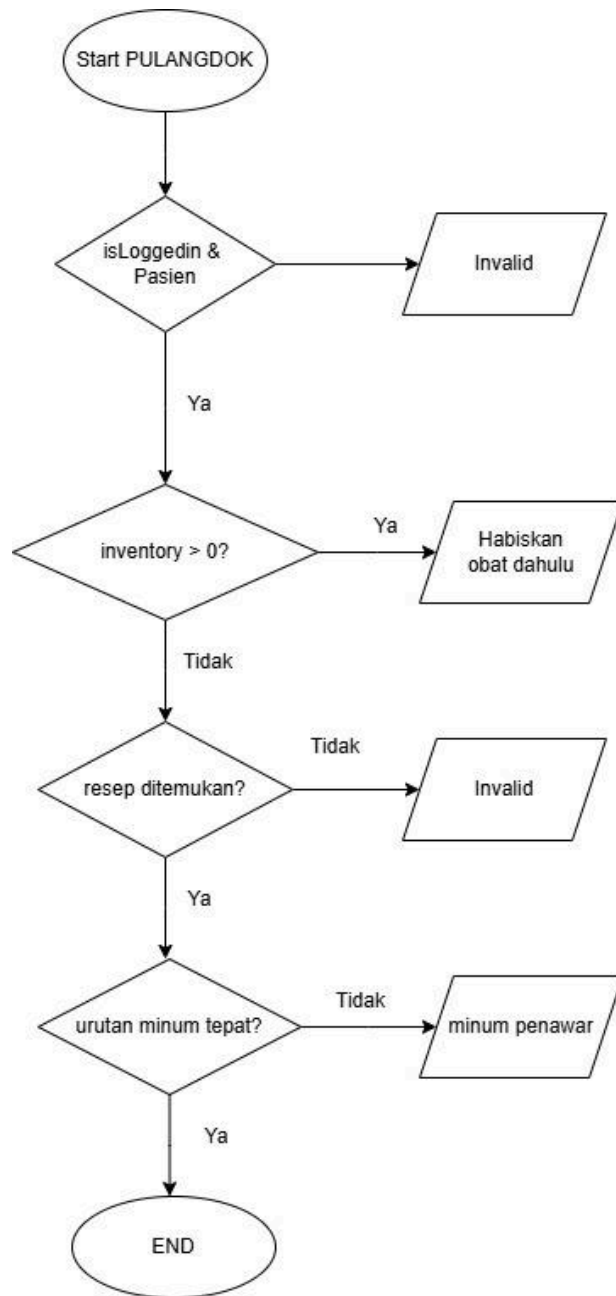
```

```

28.         output "Pasien tidak memiliki penyakit karena belum di
diagnosis!"
29.         output "Gunakan DIAGNOSIS terlebih dahulu."
30.         return
31.     end if
32.
33.     ditemukan ← -1
34.     for i ← 0 to jumlahPenyakit - 1 do
35.         if strcmp(pasien->namaPenyakit, penyakitList[i].namaPenyakit) = 0
then
36.             ditemukan ← penyakitList[i].idPenyakit
37.             break
38.         end if
39.     end for
40.
41.     if ditemukan ≠ -1 then
42.         output "Pasien memiliki penyakit ", pasien->namaPenyakit, ". Obat
yang harus diberikan:"
43.         countobat ← 0
44.
45.         for j ← 0 to jumlahObatPenyakit - 1 do
46.             if obatPenyakitList[j].idPenyakit = ditemukan then
47.                 if countobat < MAX_OBAT then
48.                     resepUntukPasien[countobat] ← obatPenyakitList[j]
49.                     countobat ← countobat + 1
50.                 end if
51.             end if
52.         end for
53.
54.         if countobat = 0 then
55.             output "Belum ada obat untuk penyakit ",
pasien->namaPenyakit, " di database."
56.             end if
57.
58.             for i ← 0 to countobat - 2 do
59.                 for j ← 0 to countobat - i - 2 do
60.                     if resepUntukPasien[j].urutanMinum > resepUntukPasien[j +
1].urutanMinum then
61.                         swapObatPenyakit(&resepUntukPasien[j],
&resepUntukPasien[j + 1])
62.                     end if
63.                 end for
64.             end for
65.
66.             for i ← 0 to countobat - 1 do
67.                 namaObatDitemukan ← "Obat Tidak Dikenal"
68.                 for k ← 0 to jumlahObatPenyakit - 1 do
69.                     if obatList[k].idObat = resepUntukPasien[i].idObat then
70.                         strcpy(namaObatDitemukan, obatList[k].namaObat)
71.                         break
72.                     end if
73.                 end for
74.                 output resepUntukPasien[i].urutanMinum, ". ",
namaObatDitemukan
75.             end for
76.         else
77.             output "Penyakit ", pasien->namaPenyakit, " tidak dikenali oleh
sistem."
78.         end if

```

- **F13 - Aku Boleh Pulang Gak**



Gambar 2.2.2.14 *Flowchart* Aku Boleh Pulang Gak

```

1. Procedure swapObatPenyakitEntri(a: pointer to ObatPenyakit, b: pointer to
   ObatPenyakit)
2. { I.S. Dua entri dalam daftar ObatPenyakit telah terdefinisi. }
3. { F.S. Posisi kedua entri tersebut telah ditukar. }
4.
5.   KAMUS
6.     temp: ObatPenyakit
7.
8.   ALGORITMA
9.     temp ← *a
10.    *a ← *b
  
```



```

11.     *b ← temp
12.
13.
14. Procedure akubolehpulanggaDok(
15.     userLoggedInInstance: pointer to User,
16.     perutPasien: pointer to StackObat,
17.     inventoryPasien: pointer to ListObat,
18.     pasienListGlobal: array of User,
19.     jumlahPasienGlobalPtr: pointer to integer,
20.     usernameSetGlobal: pointer to Set,
21.     statusLoginGlobalPtr: pointer to integer,
22.     dokterListGlobal: array of User,
23.     dokterCountGlobal: integer)
24. { I.S. Pasien yang sedang login, sudah terdiagnosis dan memiliki status
  pengobatan. }
25. { F.S. Pasien diizinkan pulang jika telah memenuhi syarat pengobatan,
  atau diberitahukan jika masih ada masalah. }
26.
27. KAMUS
28.     idPenyakitPasien: integer
29.     resepMentahDinamis: array of ObatPenyakit
30.     countResepMentah: integer
31.     resepBenar>NamaObat: array of string
32.     jumlahObatDiResepBenar: integer
33.     jumlahObatYangDiminum: integer
34.     apakahSalahMinum: boolean
35.     namaObatDitemukan: string
36.
37. ALGORITMA
38.     if statusLoginGlobalPtr = 0 or userLoggedInInstance.role ≠ "Pasien"
  then
39.         output "Perintah ini hanya dapat diakses oleh Pasien yang sudah
  login."
40.         return
41.     end if
42.
43.     if userLoggedInInstance.namaPenyakit = "" then
44.         output "Kamu belum menerima diagnosis apapun dari dokter, jangan
  buru-buru pulang!"
45.         return
46.     end if
47.
48.     output "Dokter sedang memeriksa keadaanmu..."
49.
50.     if inventoryPasien.size > 0 then
51.         output "Masih ada obat di inventory (%d obat) yang belum kamu
  habiskan, minum semuanya dulu yukk!", inventoryPasien.size
52.         return
53.     end if
54.
55.     idPenyakitPasien ← -1
56.     for i ← 0 to jumlahPenyakit - 1 do
57.         if strcmp(userLoggedInInstance.namaPenyakit,
  penyakitList[i].namaPenyakit) = 0 then
58.             idPenyakitPasien ← penyakitList[i].idPenyakit
59.             break
60.         end if
61.     end for
62.
63.     if idPenyakitPasien = -1 then
64.         output "Penyakit '%s' yang terdiagnosis tidak dikenali dalam
  sistem resep. Hubungi admin.", userLoggedInInstance.namaPenyakit
65.         return
66.     end if
67.
68.     resepMentahDinamis ← array of ObatPenyakit

```

```

69.     countResepMentah ← 0
70.     for i ← 0 to jumlahObatPenyakit - 1 do
71.         if obatPenyakitList[i].idPenyakit = idPenyakitPasien then
72.             if countResepMentah < MAX_OBAT then
73.                 resepMentahDinamis[countResepMentah] ←
obatPenyakitList[i]
74.                 countResepMentah ← countResepMentah + 1
75.             end if
76.         end if
77.     end for
78.
79.     if countResepMentah = 0 and perutPasien.top = -1 then
80.         output "Selamat! Kamu tidak memerlukan pengobatan khusus untuk
'%s' atau sudah sembuh. Silahkan pulang dan jaga kesehatan!",
userLoggedInInstance.namaPenyakit
81.         prosesPasienSelesaiAtauDed(userLoggedInInstance.username,
userLoggedInInstance, FALSE, pasienListGlobal, jumlahPasienGlobalPtr,
usernameSetGlobal, perutPasien, inventoryPasien)
82.
83.         if userLoggedInInstance.dokterPerawat ≠ "" then
84.             for i ← 0 to dokterCountGlobal - 1 do
85.                 if strcmp(dokterListGlobal[i].username,
userLoggedInInstance.dokterPerawat) = 0 then
86.                     dokterListGlobal[i].aura ← dokterListGlobal[i].aura +
1
87.                     output "Dokter %s mendapatkan +1 Aura! (Total: %d)",
dokterListGlobal[i].username, dokterListGlobal[i].aura
88.                     break
89.                 end if
90.             end for
91.         end if
92.         return
93.     end if
94.
95.     for i ← 0 to countResepMentah - 1 do
96.         for j ← 0 to countResepMentah - i - 2 do
97.             if resepMentahDinamis[j].urutanMinum > resepMentahDinamis[j +
1].urutanMinum then
98.                 swapObatPenyakitEntri(&resepMentahDinamis[j],
&resepMentahDinamis[j + 1])
99.             end if
100.        end for
101.    end for
102.
103.    resepBenar_NamaObat ← array of string
104.    jumlahObatDiResepBenar ← countResepMentah
105.    for i ← 0 to jumlahObatDiResepBenar - 1 do
106.        idObatCari ← resepMentahDinamis[i].idObat
107.        ketemuObatDiList ← 0
108.        for k ← 0 to jumlahObat - 1 do
109.            if obatList[k].idObat = idObatCari then
110.                strncpy(resepBenar_NamaObat[i], obatList[k].namaObat,
MAX_NAMA_OBAT_LEN - 1)
111.                resepBenar_NamaObat[i][MAX_NAMA_OBAT_LEN - 1] ← '\0'
112.                ketemuObatDiList ← 1
113.                break
114.            end if
115.        end for
116.        if ketemuObatDiList = 0 then
117.            strcpy(resepBenar_NamaObat[i], "ObatTidakDikenal")
118.        end if
119.    end for
120.
121.    jumlahObatYangDiminum ← perutPasien.top + 1
122.    apakahSalahMinum ← FALSE
123.    if jumlahObatYangDiminum ≠ jumlahObatDiResepBenar then

```

```

124.         apakahSalahMinum ← TRUE
125.         if jumlahObatDiResepBenar = 0 and jumlahObatYangDiminum > 0
126.         then
127.             output "Kamu seharusnya tidak minum obat apapun untuk
128.             penyakit ini!"
129.         end if
130.         else if jumlahObatDiResepBenar = 0 and jumlahObatYangDiminum = 0
131.         then
132.             apakahSalahMinum ← FALSE
133.         else
134.             for i ← 0 to jumlahObatDiResepBenar - 1 do
135.                 if strcmp(perutPasien.data[i], resepBenar_NamaObat[i]) ≠
136.                 0 then
137.                     apakahSalahMinum ← TRUE
138.                     break
139.                 end if
140.             end for
141.         end if
142.         if apakahSalahMinum then
143.             output "Maaf, tapi kamu masih belum bisa pulang!"
144.             if jumlahObatDiResepBenar > 0 then
145.                 output "Urutan peminuman obat yang diharapkan untuk %s:",
146.                 userLoggedInInstance.namaPenyakit
147.                 for i ← 0 to jumlahObatDiResepBenar - 1 do
148.                     output resepBenar_NamaObat[i], (i =
149.                     jumlahObatDiResepBenar - 1) ? "" : " -> "
150.                 end for
151.                 output "\n"
152.             else
153.                 output "Seharusnya kamu tidak minum obat apapun."
154.             end if
155.             output "Urutan obat yang kamu minum:"
156.             if jumlahObatYangDiminum = 0 then
157.                 output "(Perut kosong)"
158.             else
159.                 for i ← 0 to jumlahObatYangDiminum - 1 do
160.                     output perutPasien.data[i], (i =
161.                     jumlahObatYangDiminum - 1) ? "" : " -> "
162.                 end for
163.                 output "\n"
164.             end if
165.             if userLoggedInInstance.nyawa > 0 then
166.                 userLoggedInInstance.nyawa ← userLoggedInInstance.nyawa -
167.                 1
168.             end if
169.             for i ← 0 to *jumlahPasienGlobalPtr - 1 do
170.                 if pasienListGlobal[i].id = userLoggedInInstance.id then
171.                     pasienListGlobal[i].nyawa ←
172.                     userLoggedInInstance.nyawa
173.                     break
174.                 end if
175.             end for
176.             sisanyawanimon(userLoggedInInstance.nyawa)
177.             if userLoggedInInstance.nyawa ≤ 0 then
178.                 output "Kondisi pasien menjadi fatal, kamu dinyatakan
179.                 'ded'!"
180.                 prosesPasienSelesaiAtauDed(userLoggedInInstance.username,
181.                 userLoggedInInstance, TRUE, pasienListGlobal, jumlahPasienGlobalPtr,
182.                 usernameSetGlobal, perutPasien, inventoryPasien)
183.                 *statusLoginGlobalPtr ← 0

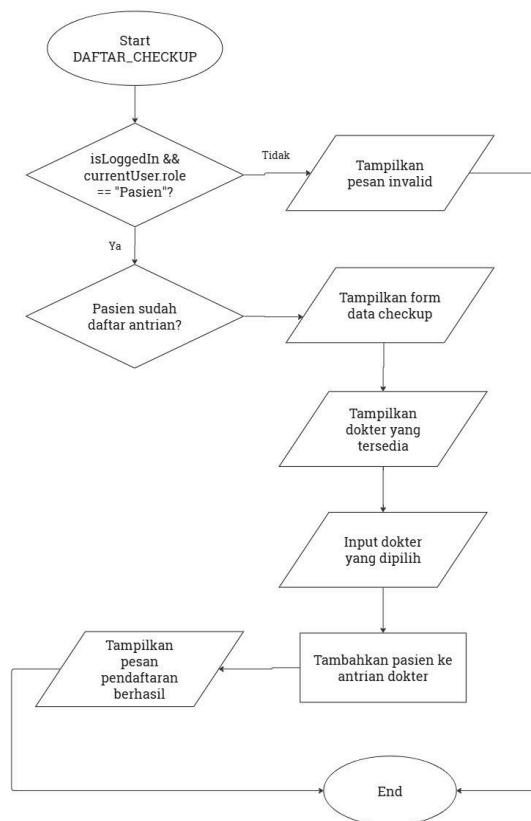
```

```

178.         else
179.             output "Silahkan gunakan PENAWAR atau konsultasi lagi!"
180.         end if
181.     else
182.         output "Selamat! Kamu sudah dinyatakan sembuh oleh dokter.
Silahkan pulang dan semoga sehat selalu!"
183.
184.         if strlen(userLoggedInInstance.dokterPerawat) > 0 then
185.             for i ← 0 to dokterCountGlobal - 1 do
186.                 if strcmp(dokterListGlobal[i].username,
userLoggedInInstance.dokterPerawat) = 0 then
187.                     dokterListGlobal[i].aura ←
dokterListGlobal[i].aura + 1
188.                     output "Dokter %s mendapatkan +1 Aura! (Total
Aura: %d)", dokterListGlobal[i].username, dokterListGlobal[i].aura
189.                     break
190.                 end if
191.             end for
192.         end if
193.
194.         prosesPasienSelesaiAtauDed(userLoggedInInstance.username,
userLoggedInInstance, FALSE, pasienListGlobal, jumlahPasienGlobalPtr,
usernameSetGlobal, perutPasien, inventoryPasien)
195.         output "# Pasien %s kembali ke menu utama dan meninggalkan
ruangan.", userLoggedInInstance.username
196.     end if

```

- **F14 - Daftar Checkup**



Gambar 2.2.2.15 *Flowchart* Daftar Checkup

```
1. Procedure swapObatPenyakitEntri(a: pointer to
   ObatPenyakit, b: pointer to ObatPenyakit)
2. { I.S. Dua entri dalam daftar ObatPenyakit telah
   terdefinisi. }
3. { F.S. Posisi kedua entri tersebut telah ditukar. }
4.
5.   KAMUS
6.     temp: ObatPenyakit
7.
8.   ALGORITMA
9.     temp ← *a
10.    *a ← *b
11.    *b ← temp
12. }
13.
14. Procedure daftarCheckUp(
15.   paraDokter: array of User,
16.   jumlahDokterAktif: integer,
17.   semuaPasien: array of User,
18.   jumlahPasienSekarang: integer)
19. { I.S. Pasien yang sedang login, belum terdaftar dalam
   antrian check-up. }
20. { F.S. Pasien didaftarkan dalam antrian check-up atau
   diberikan peringatan jika sudah terdaftar. }
21.
22.   KAMUS
23.     sudahAdaDiAntrian: boolean
24.     dataMedisInputLokal: CheckUpData
25.     pilihanUserNomor: integer
26.     dokterYangBisaDipilih: array of User
27.     indeksAsliDokter: array of integer
28.     countDokterBisaDipilih: integer
29.     antrianDokterIni: pointer to AntrianDokter
30.     jumlahAntrianSkrng: integer
31.
32.   ALGORITMA
33.     if not isLoggedIn or currentUser.role ≠ "Pasien" then
34.       output "Perintah DAFTAR_CHECKUP hanya untuk Pasien
   yang sudah login!"
35.       return
36.     end if
37.
38.     sudahAdaDiAntrian ← FALSE
39.     for i ← 0 to jumlahDokterAktif - 1 do
40.       if dokterQueues[i].dokter = NULL then continue end
   if
41.         cekNode ← dokterQueues[i].head
42.         while cekNode ≠ NULL do
43.           if strcmp(cekNode.pasien_data.username,
   currentUser.username) = 0 then
44.             sudahAdaDiAntrian ← TRUE
45.             break
```

```

46.         end if
47.         cekNode ← cekNode.next
48.     end while
49.     if sudahAdaDiAntrian then break end if
50. end for
51.
52. if sudahAdaDiAntrian then
53.     output "Anda sudah terdaftar dalam antrian
    check-up!"
54.     output "Silakan selesaikan check-up yang sudah
    terdaftar terlebih dahulu."
55.     return
56. end if
57.
58. input dataMedisInputLokal
59. do until valid suhu_tubuh do
60.     input suhu_tubuh
61.     if suhu_tubuh < 35.0 or suhu_tubuh > 42.0 then
62.         output "Suhu tubuh invalid (35°C-42°C)!
        Masukkan lagi."
63.     end if
64. end do
65.
66. do until valid tekanan_darah_sistol and
    tekanan_darah_diastol do
67.     input tekanan_darah_sistol, tekanan_darah_diastol
68.     if tekanan_darah_sistol < 90 or
    tekanan_darah_sistol > 180 or
69.     tekanan_darah_diastol < 60 or
    tekanan_darah_diastol > 120 then
70.         output "Tekanan darah invalid (Sistol 90-180,
        Diastol 60-120)! Masukkan lagi."
71.     end if
72. end do
73.
74. do until valid detak_jantung do
75.     input detak_jantung
76.     if detak_jantung < 50 or detak_jantung > 220 then
77.         output "Detak jantung invalid (50-220 bpm)!
        Masukkan lagi."
78.     end if
79. end do
80.
81. do until valid saturasi_oksigen do
82.     input saturasi_oksigen
83.     if saturasi_oksigen < 90.0 or saturasi_oksigen >
    100.0 then
84.         output "Saturasi oksigen invalid (90-100%)!
        Masukkan lagi."
85.     end if
86. end do
87.
88. do until valid kadar_gula do
89.     input kadar_gula

```

```

90.         if kadar_gula < 50 or kadar_gula > 250 then
91.             output "Kadar gula darah invalid (50-250
           mg/dL)! Masukkan lagi."
92.         end if
93.     end do
94.
95.     do until valid berat_badan do
96.         input berat_badan
97.         if berat_badan <= 0 or berat_badan > 200 then
98.             output "Berat badan invalid (>0, max 200kg)!
           Masukkan lagi."
99.         end if
100.    end do
101.
102.    do until valid tinggi_badan do
103.        input tinggi_badan
104.        if tinggi_badan <= 50 or tinggi_badan > 220
           then
105.            output "Tinggi badan invalid (50-220 cm)!
           Masukkan lagi."
106.        end if
107.    end do
108.
109.    do until valid kadar_kolesterol do
110.        input kadar_kolesterol
111.        if kadar_kolesterol < 100 or kadar_kolesterol >
           300 then
112.            output "Kadar kolesterol invalid (100-300
           mg/dL)! Masukkan lagi."
113.        end if
114.    end do
115.
116.    do until valid trombosit do
117.        input trombosit
118.        if trombosit < 150000 or trombosit > 450000
           then
119.            output "Trombosit invalid (150000-450000
           per uL)! Masukkan lagi."
120.        end if
121.    end do
122.
123.    output "Berikut adalah daftar dokter yang tersedia
           (yang memiliki ruangan):"
124.    dokterYangBisaDipilih ← array of User
125.    indeksAsliDokter ← array of integer
126.    countDokterBisaDipilih ← 0
127.    for i ← 0 to jumlahDokterAktif - 1 do
128.        if paraDokter[i].kodeRuangan ≠ "" then
129.            dokterYangBisaDipilih[countDokterBisaDipilih] ←
           paraDokter[i]
130.            indeksAsliDokter[countDokterBisaDipilih] ←
           i

```

```

131.         countDokterBisaDipilih ←
countDokterBisaDipilih + 1
132.         end if
133.     end for
134.
135.     if countDokterBisaDipilih = 0 then
136.         output "Mohon maaf, tidak ada dokter yang
praktek di ruangan saat ini."
137.         return
138.     end if
139.
140.     for i ← 0 to countDokterBisaDipilih - 1 do
141.         antrianDokterIni ← NULL
142.         for k ← 0 to jumlahDokterAktif - 1 do
143.             if dokterQueues[k].dokter ≠ NULL and
dokterQueues[k].dokter.id = dokterYangBisaDipilih[i].id
then
144.                 antrianDokterIni ← dokterQueues[k]
145.                 break
146.             end if
147.         end for
148.         jumlahAntrianSkrng ← antrianDokterIni ≠ NULL ?
antrianDokterIni.jumlah_antrian : 0
149.         output "%d. Dr. %s - Ruangan %s (Antrian: %d
orang) - Aura: %d",
150.             i + 1,
dokterYangBisaDipilih[i].username,
dokterYangBisaDipilih[i].kodeRuangan, jumlahAntrianSkrng,
dokterYangBisaDipilih[i].aura
151.     end for
152.
153.     input pilihanUserNomor
154.     if pilihanUserNomor < 1 or pilihanUserNomor >
countDokterBisaDipilih then
155.         output "Pilihan dokter tidak tersedia!"
156.         return
157.     end if
158.
159.     idxAsliDokterYgDipilih ←
indeksAsliDokter[pilihanUserNomor - 1]
160.
161.     newNode ← malloc(sizeof(AntrianNode))
162.     if newNode = NULL then
163.         output "Error Kritis: Gagal alokasi memori
untuk membuat antrian baru."
164.         return
165.     end if
166.
167.     newNode.pasien_data ← currentUser
168.     newNode.data_medis ← dataMedisInputLokal
169.     newNode.status ← 0
170.     newNode.next ← NULL
171.
172.     selectedQueue ← NULL

```



```

173.         for i ← 0 to jumlahDokterAktif - 1 do
174.             if dokterQueues[i].dokter ≠ NULL and
               dokterQueues[i].dokter.id =
               paraDokter[idxAsliDokterYgDipilih].id then
175.                 selectedQueue ← dokterQueues[i]
176.                 break
177.             end if
178.         end for
179.         if selectedQueue = NULL or selectedQueue.dokter =
           NULL then
180.             output "Error Internal: Gagal menemukan antrian
               untuk dokter yang dipilih."
181.             free(newNode)
182.             return
183.         end if
184.
185.         if selectedQueue.head = NULL then
186.             selectedQueue.head ← newNode
187.             selectedQueue.tail ← newNode
188.         else
189.             selectedQueue.tail.next ← newNode
190.             selectedQueue.tail ← newNode
191.         end if
192.         selectedQueue.jumlah_antrian ←
           selectedQueue.jumlah_antrian + 1
193.
194.         strncpy(currentUser.dokterPerawat,
           selectedQueue.dokter.username, MAX_USERNAME_LEN - 1)
195.         currentUser.dokterPerawat[MAX_USERNAME_LEN - 1] ←
           '\0'
196.
197.         currentUser.suhuTubuh ←
           dataMedisInputLokal.suhu_tubuh
198.         currentUser.tekananDarahSistolik ←
           dataMedisInputLokal.tekanan_darah_sistol
199.         currentUser.tekananDarahDiastolik ←
           dataMedisInputLokal.tekanan_darah_diastol
200.         currentUser.detakJantung ←
           dataMedisInputLokal.detak_jantung
201.         currentUser.saturasiOksigen ←
           dataMedisInputLokal.saturasi_oksigen
202.         currentUser.kadarGulaDarah ←
           (int) dataMedisInputLokal.kadar_gula
203.         currentUser.beratBadan ←
           dataMedisInputLokal.berat_badan
204.         currentUser.tinggiBadan ←
           (int) dataMedisInputLokal.tinggi_badan
205.         currentUser.kadarKolesterol ←
           (int) dataMedisInputLokal.kadar_kolesterol
206.         currentUser.trombosit ←
           dataMedisInputLokal.trombosit
207.
208.         for i ← 0 to jumlahPasienSekarang - 1 do
209.             if semuaPasien[i].id = currentUser.id then

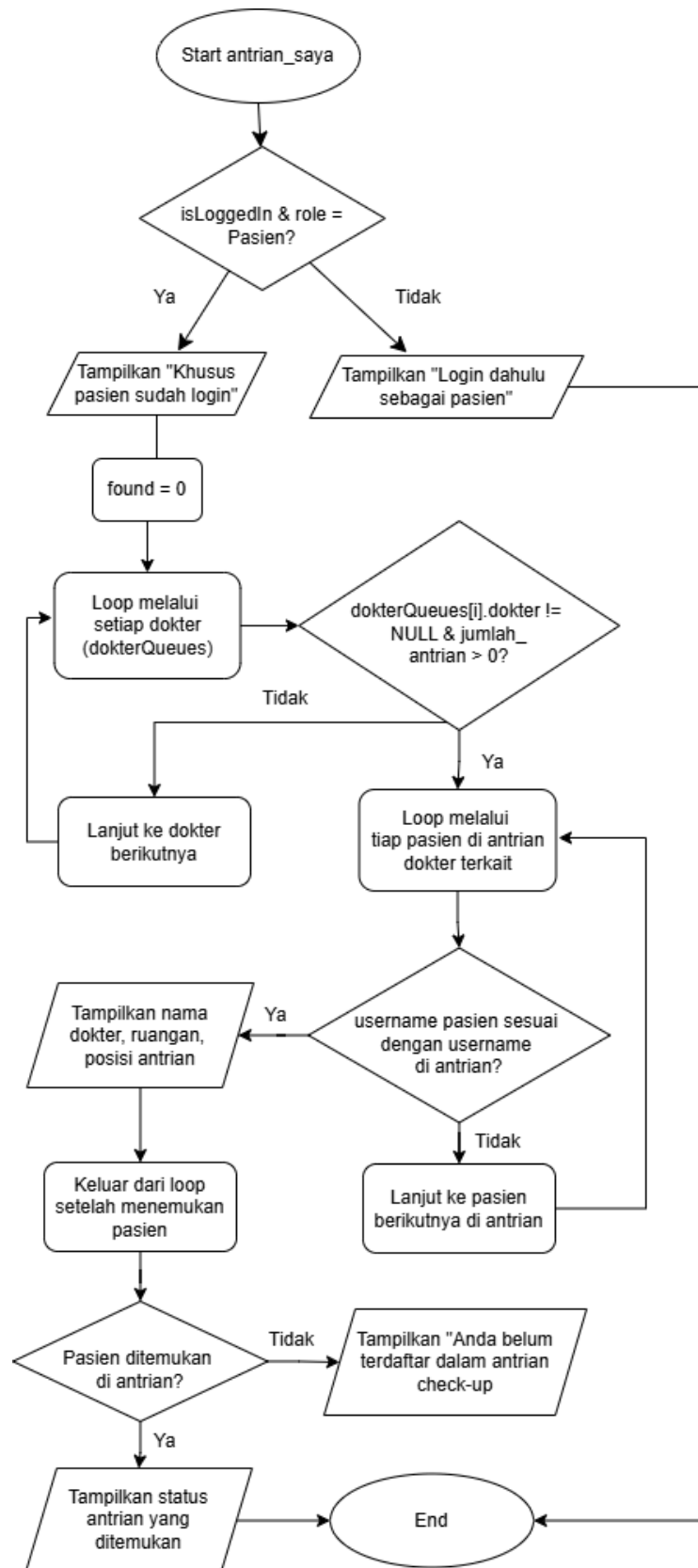
```

```

210.             strncpy(semuaPasien[i].dokterPerawat,
selectedQueue.dokter.username, MAX_USERNAME_LEN - 1)
211.             semuaPasien[i].dokterPerawat[MAX_USERNAME_LEN - 1] ← '\0'
212.
213.             semuaPasien[i].suhuTubuh ←
dataMedisInputLokal.suhu_tubuh
214.             semuaPasien[i].tekananDarahSistolik ←
dataMedisInputLokal.tekanan_darah_sistol
215.             semuaPasien[i].tekananDarahDiastolik ←
dataMedisInputLokal.tekanan_darah_diastol
216.             semuaPasien[i].detakJantung ←
dataMedisInputLokal.detak_jantung
217.             semuaPasien[i].saturasiOksigen ←
dataMedisInputLokal.saturasi_oksigen
218.             semuaPasien[i].kadarGulaDarah ←
(int)dataMedisInputLokal.kadar_gula
219.             semuaPasien[i].beratBadan ←
dataMedisInputLokal.berat_badan
220.             semuaPasien[i].tinggiBadan ←
(int)dataMedisInputLokal.tinggi_badan
221.             semuaPasien[i].kadarKolesterol ←
(int)dataMedisInputLokal.kadar_kolesterol
222.             semuaPasien[i].trombosit ←
dataMedisInputLokal.trombosit
223.             break
224.         end if
225.     end for
226.
227.     output "Pendaftaran check-up berhasil!"
228.     output "Anda terdaftar pada antrian Dr. %s di
ruangan %s.", selectedQueue.dokter.username,
selectedQueue.dokter.kodeRuangan
229.     output "Posisi antrian Anda: %d dari %d",
selectedQueue.jumlah_antrian, selectedQueue.jumlah_antrian

```

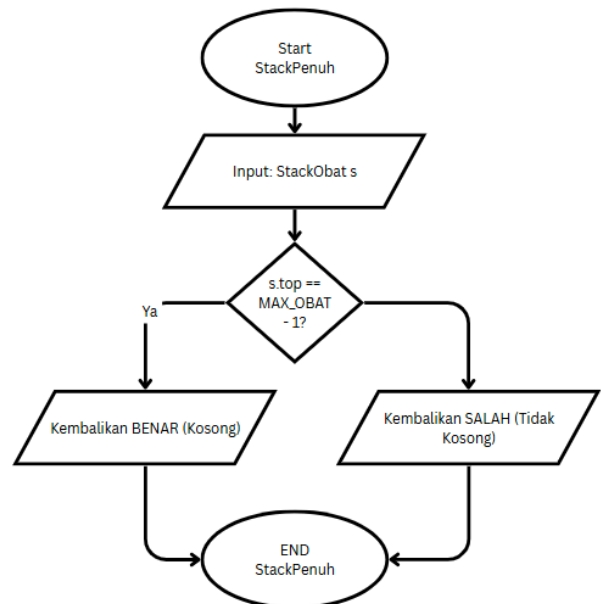
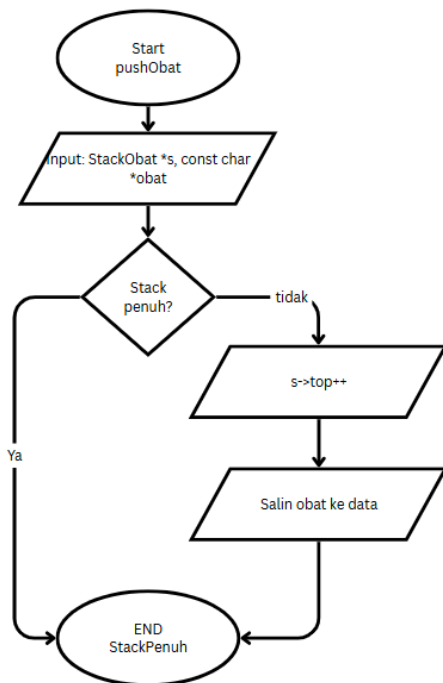
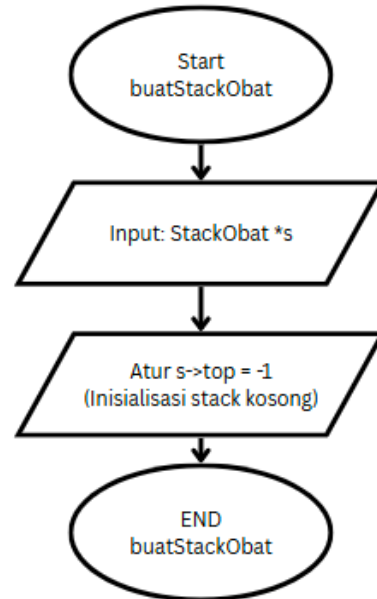
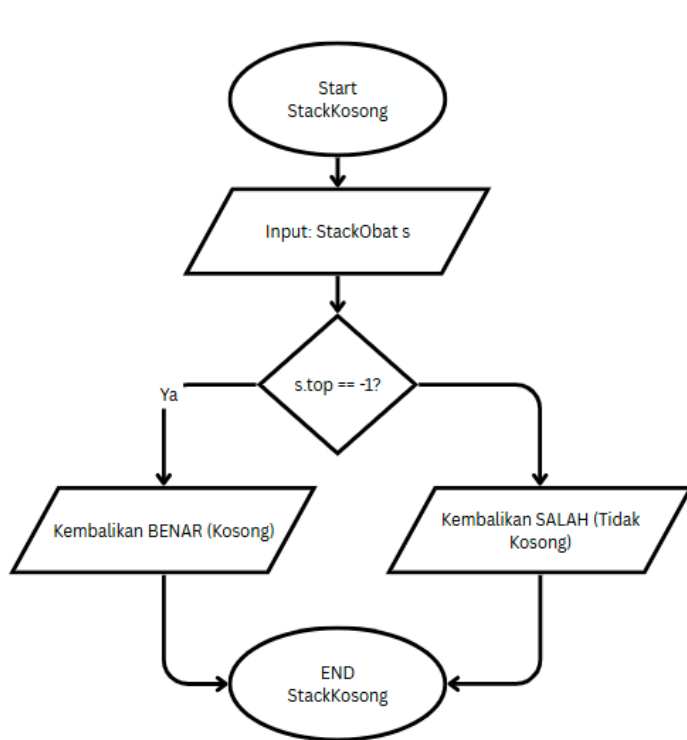
- **F15 - Status Antrian Saya**

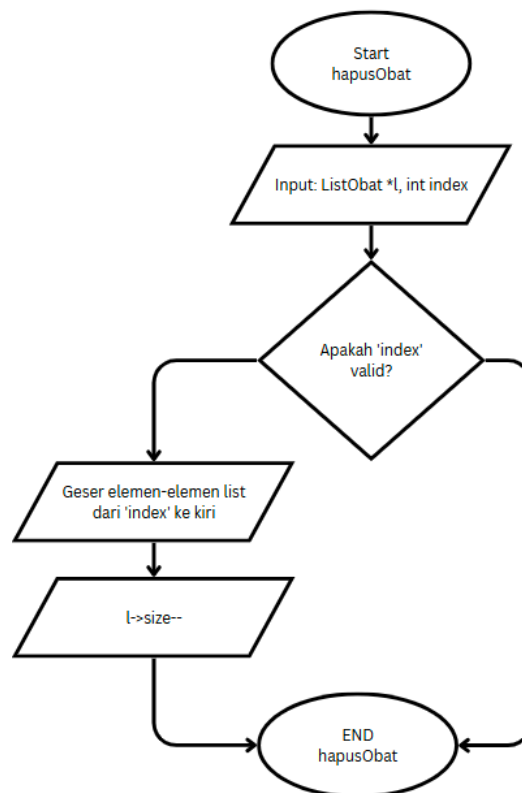
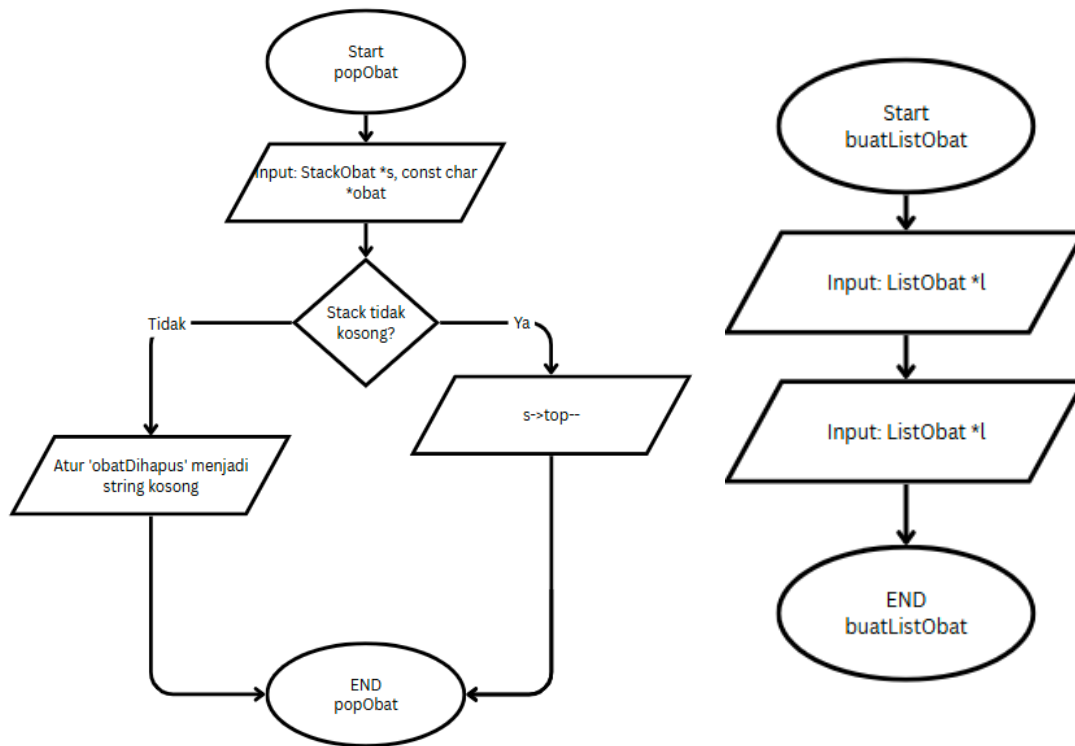


Gambar 2.2.2.16 *Flowchart* Status Antrian Saya

```
1. Procedure antrianSaya(dokterCount: integer)
2. { I.S. Pasien yang sedang login, dan sistem antrean dokter telah
  diinisialisasi. }
3. { F.S. Menampilkan status antrian pasien atau pesan bahwa pasien belum
  terdaftar. }
4.
5.   KAMUS
6.     found: boolean
7.     current: pointer to AntrianNode
8.     posisi: integer
9.
10.  ALGORITMA
11.    found ← FALSE
12.    for i ← 0 to MAX_JUMLAH_USER - 1 do
13.      if dokterQueues[i].dokter = NULL or
dokterQueues[i].jumlah_antrian = 0 then
14.        continue
15.      end if
16.
17.      current ← dokterQueues[i].head
18.      posisi ← 1
19.      while current ≠ NULL do
20.        if strcmp(current.pasien_data.username, currentUser.username)
= 0 then
21.          output "\nStatus antrian Anda:"
22.          output "Dokter: Dr. ", dokterQueues[i].dokter.username
23.          output "Ruangan: ", dokterQueues[i].dokter.kodeRuangan
24.          output "Posisi antrian: ", posisi, " dari ",
dokterQueues[i].jumlah_antrian
25.          found ← TRUE
26.          break
27.        end if
28.        current ← current.next
29.        posisi ← posisi + 1
30.      end while
31.      if found then break end if
32.    end for
33.
34.    if not found then
35.      output "\nAnda belum terdaftar dalam antrian check-up!"
36.      output "Silakan daftar terlebih dahulu dengan command
DAFTAR_CHECKUP."
37.    end if
```

- **F16 - Minum Obat**

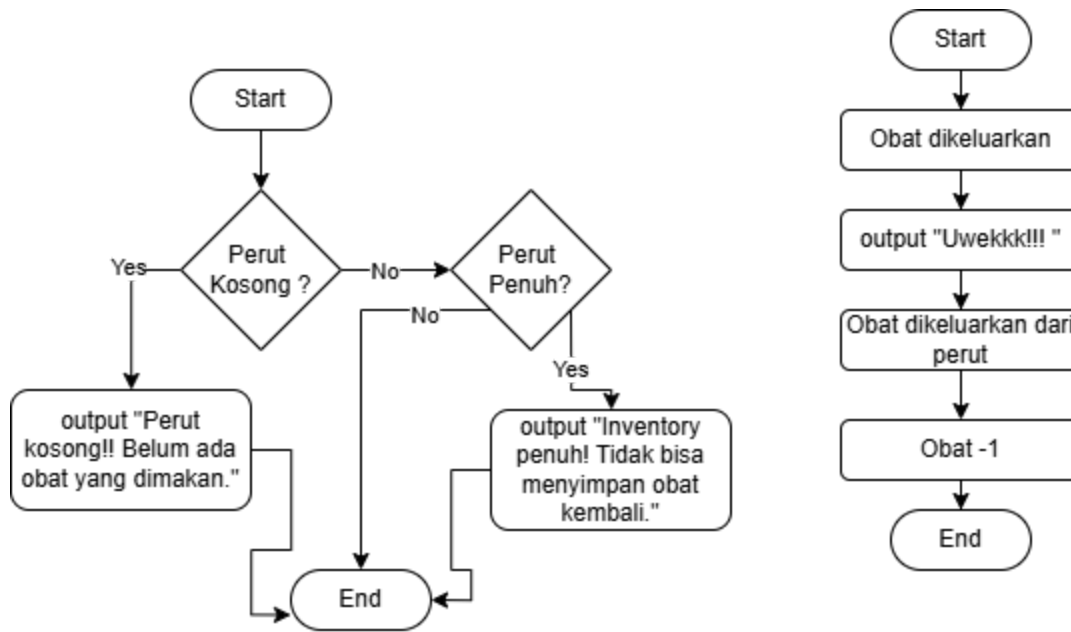




Gambar 2.2.2.17 *Flowchart* Minum Obat

```
1. Procedure minumObat(pasien: User, perut: StackObat, inventory: ListObat)
2. { I.S. Pasien telah login, inventory obat tersedia, dan pasien memiliki
  obat yang dapat diminum. }
3. ( F.S. Obat dipilih dan diminum, dan inventory obat diperbarui. )
4.
5.   KAMUS
6.     pilihan: integer
7.     c: integer
8.
9.   ALGORITMA
10.    if inventory.size = 0 then
11.      output "Inventory obat kosong! Tidak ada obat yang bisa diminum."
12.      return
13.    end if
14.
15.    tampilkanListObat(inventory)
16.    output "Pilih obat untuk diminum (nomor): "
17.    if scanf("%d", pilihan) ≠ 1 then
18.      output "Input tidak valid! Masukkan nomor obat."
19.      clearInputBuffer()
20.      return
21.    end if
22.
23.    if pilihan < 1 or pilihan > inventory.size then
24.      output "Pilihan nomor tidak tersedia!"
25.      return
26.    end if
27.
28.    pushObat(perut, inventory.data[pilihan - 1])
29.    output "GLEKGLEKGLEK... ", inventory.data[pilihan - 1], " berhasil
    diminum!!!"
30.
31.    hapusObat(inventory, pilihan - 1)
32.
33.    if inventory.size > 0 then
34.      tampilkanListObat(inventory)
35.    else
36.      output "Inventory sekarang kosong."
37.    end if
```

- **F17 - Minum Penawar**

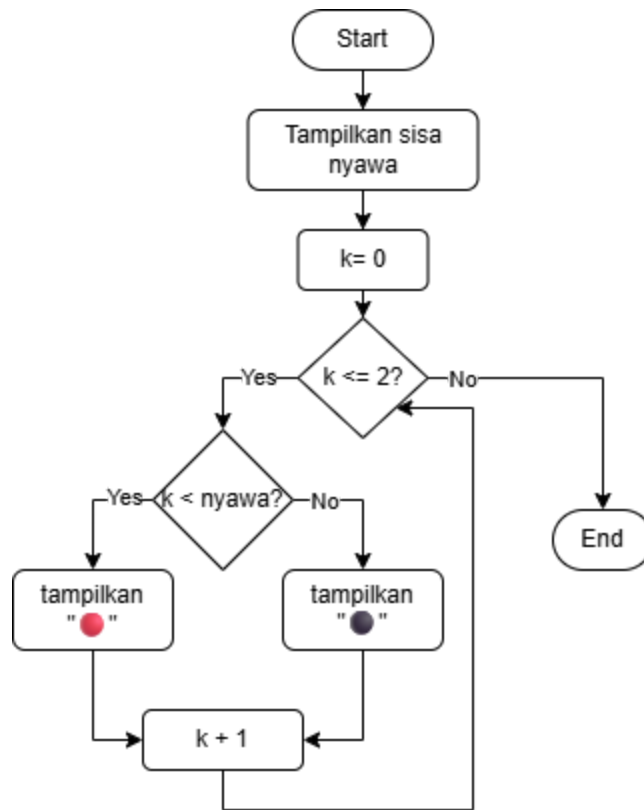


Gambar 2.2.2.18 Flowchart Minum Penawar

```

1. Procedure minumPenawar(perut: StackObat, inventory: ListObat)
2. { I.S. Perut berisi obat yang telah diminum, inventory belum penuh. }
3. { F.S. Obat dikeluarkan dari perut dan dikembalikan ke dalam inventory. }
4.
5.   KAMUS
6.     obatDikeluarkan: string
7.
8.   ALGORITMA
9.     if stackKosong(perut) then
10.       output "Perut kosong!! Belum ada obat yang dimakan."
11.       return
12.     end if
13.
14.     if inventory.size >= MAX_OBAT then
15.       output "Inventory penuh! Tidak bisa menyimpan obat kembali."
16.       return
17.     end if
18.
19.     popObat(perut, obatDikeluarkan)
20.     strcpy(inventory.data[inventory.size], obatDikeluarkan)
21.     inventory.size++
22.
23.     output "Uwekkk!!! ", obatDikeluarkan, " keluar dan kembali ke
     inventory"
  
```

- **B05 - DEAD or ALIVE??**



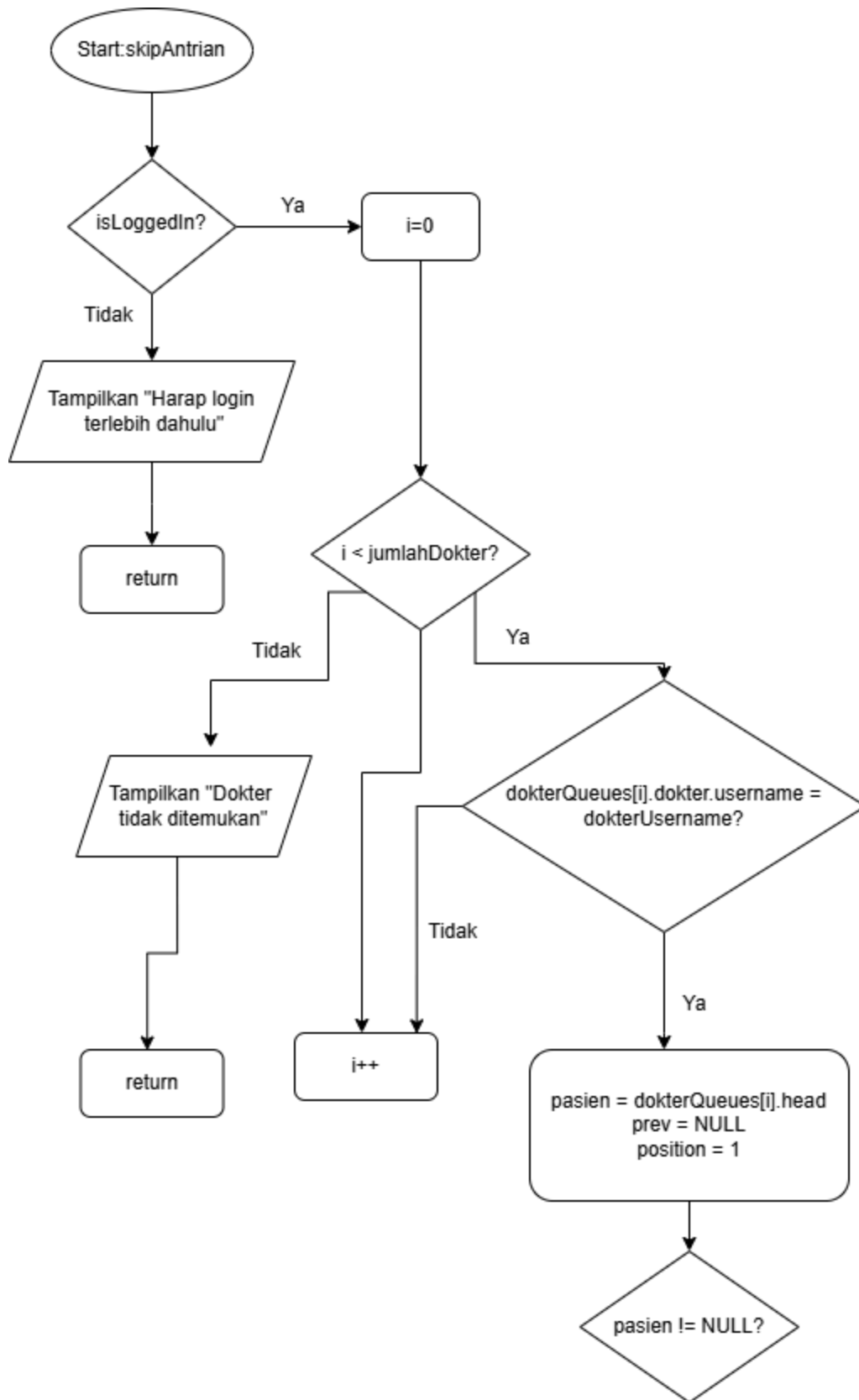
Gambar 2.2.2.19 Flowchart Dead or Alive

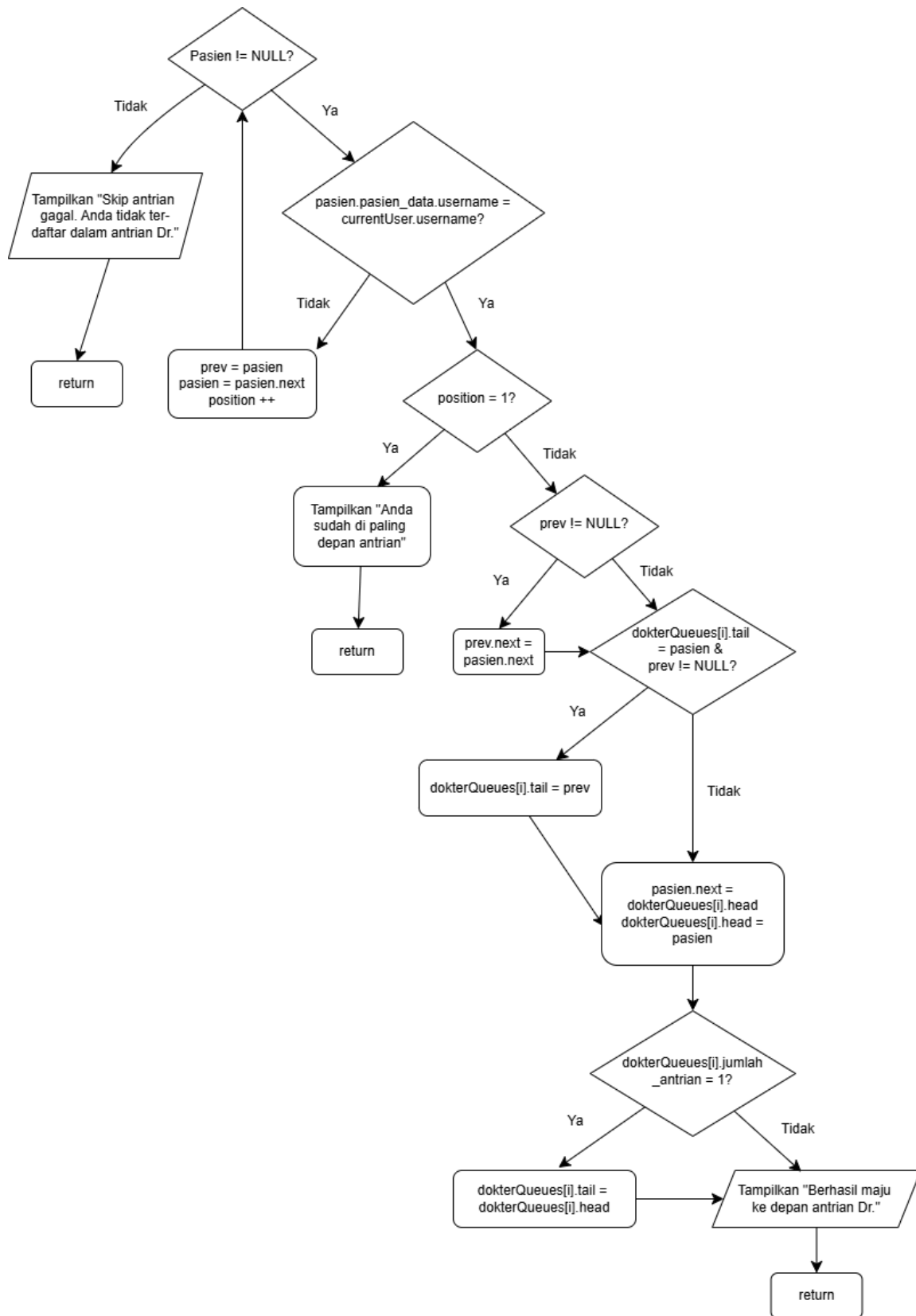
```

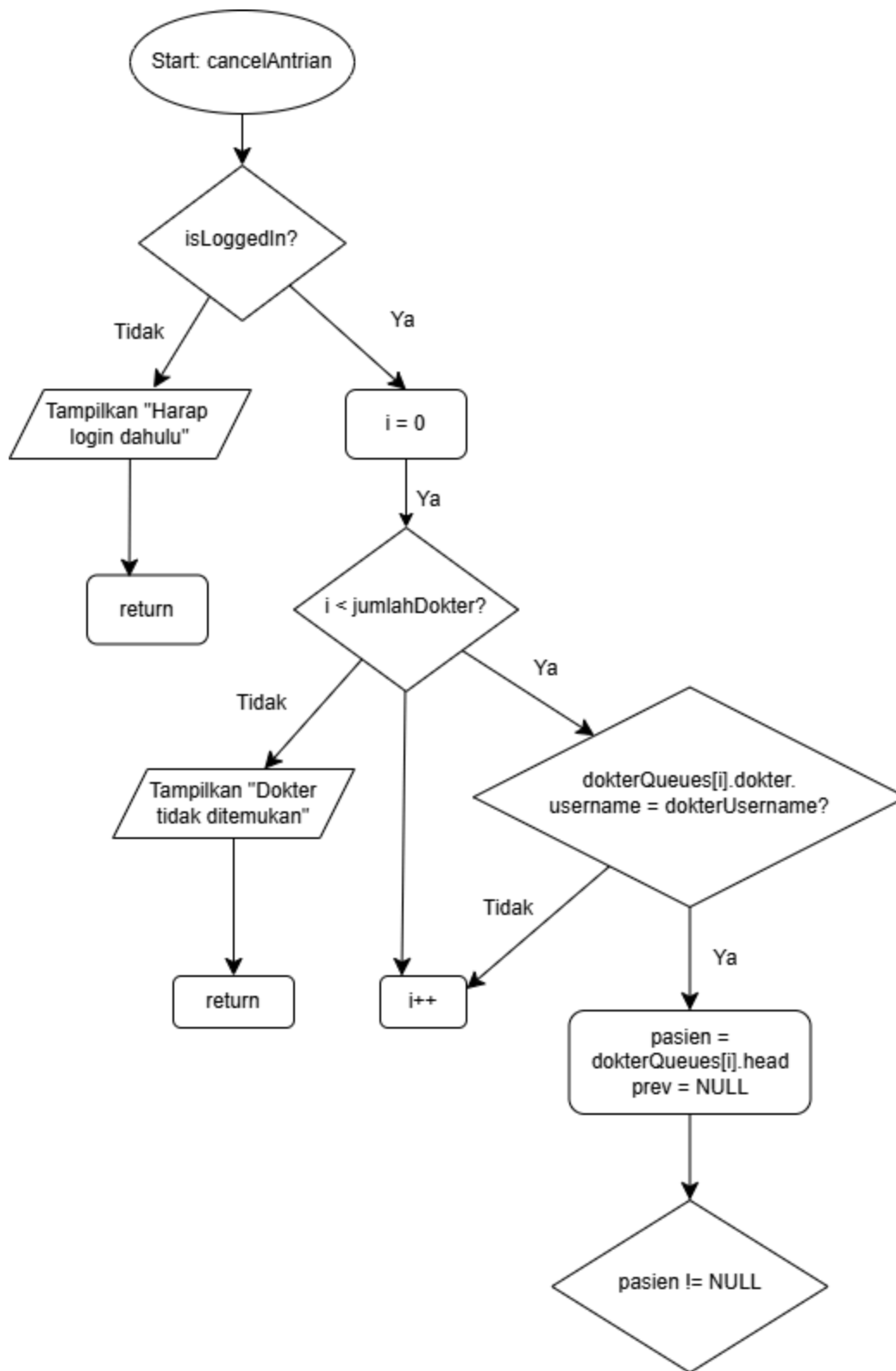
1. Procedure sisanyawanimon(nyawa: integer)
2. { I.S. nyawa bernilai 0..3 }
3. { F.S. Menampilkan nyawa dalam bentuk ● dan ● sesuai jumlah nyawa }
4.
5. KAMUS
6.   nyawa, k: integer
7.
8. ALGORITMA
9.   output "Sisa nyawa Anda: "
10.  for k ← 0 to 2 do
11.    if k < nyawa then
12.      output "●"
13.    else
14.      output "●"
15.    end if
16.  end for
17.  output " (", nyawa, " nyawa)"

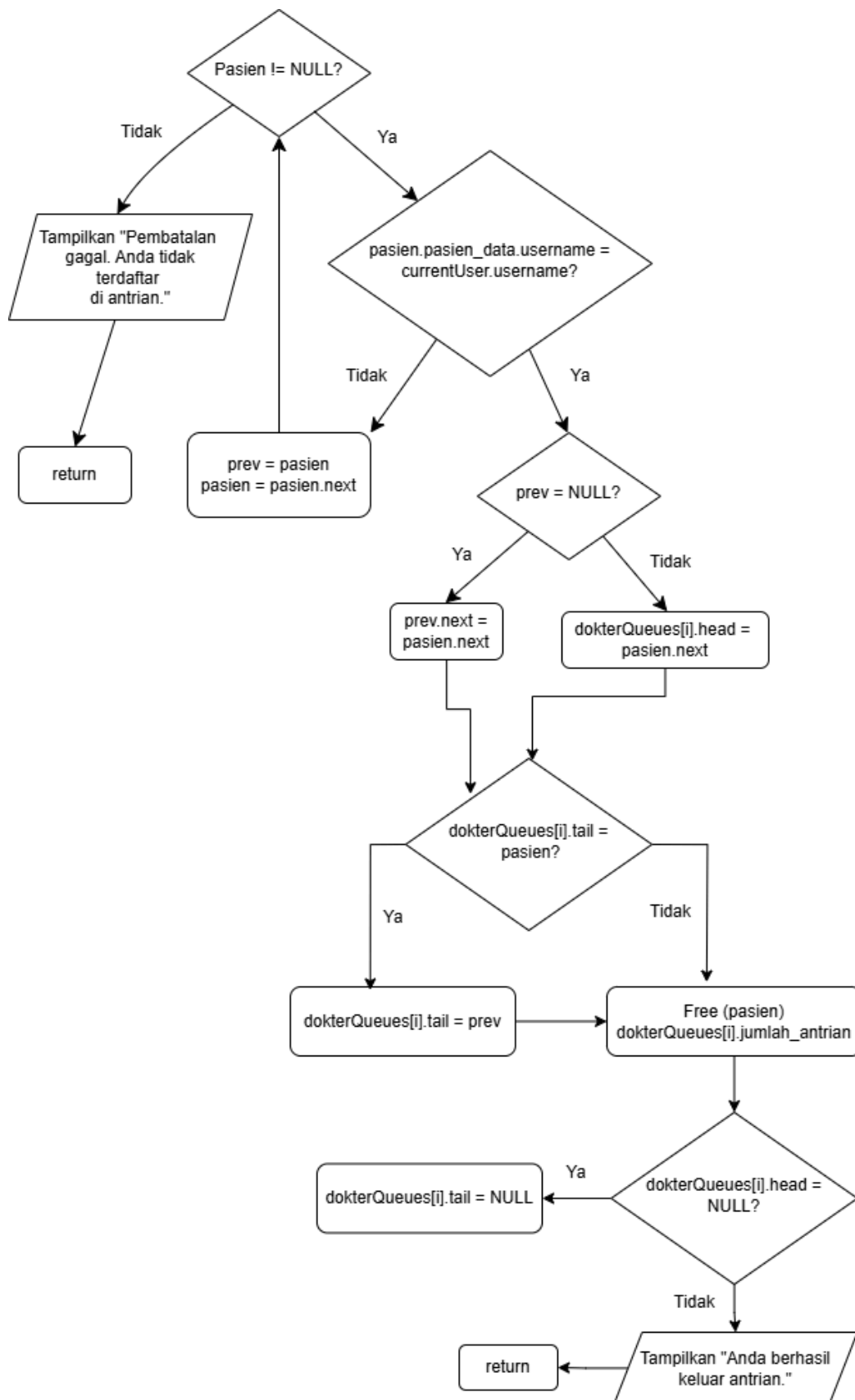
```

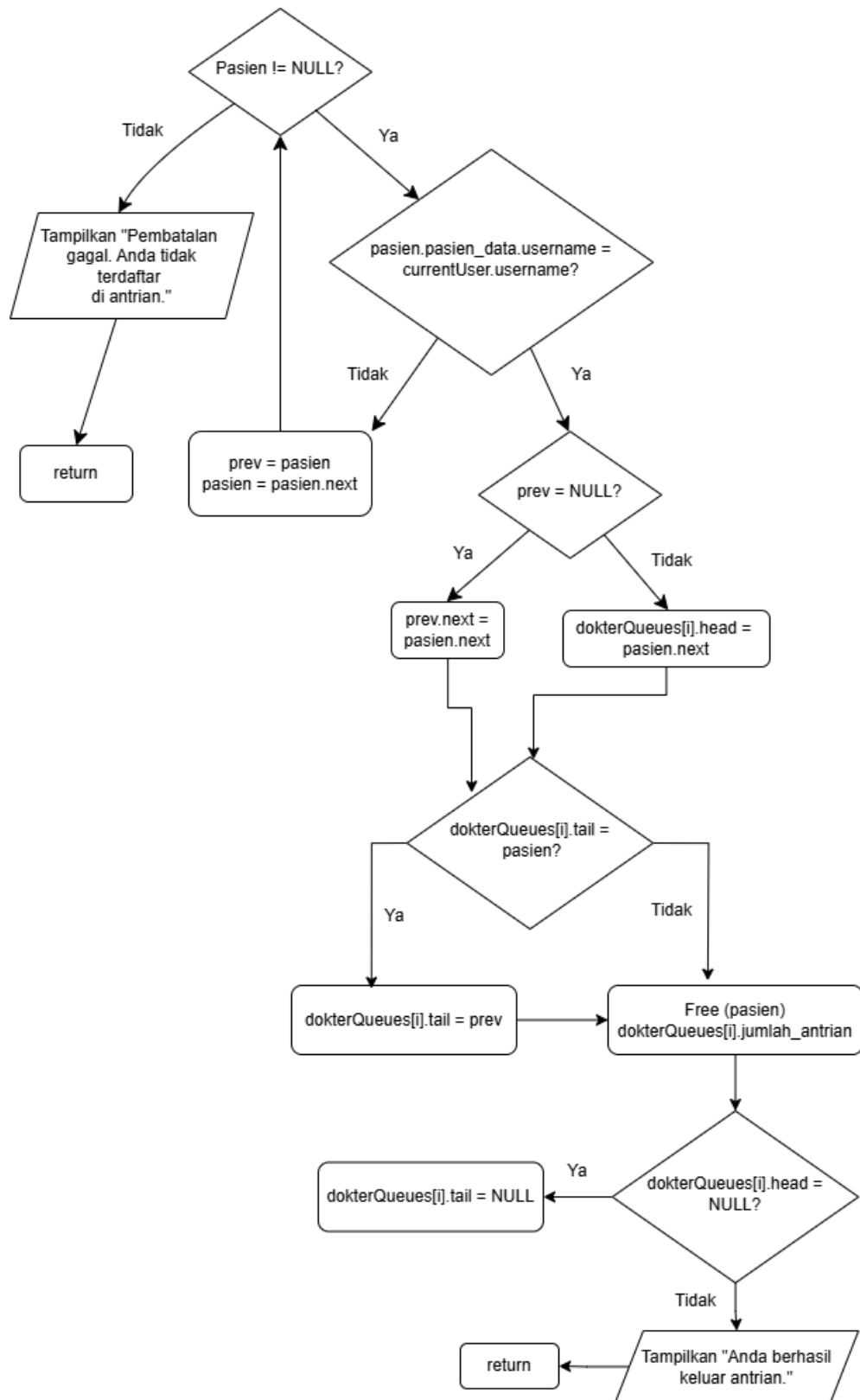
- **B06 - Mainin Antrian**











Gambar 2.2.2.20 *Flowchart* Mainin Antrian

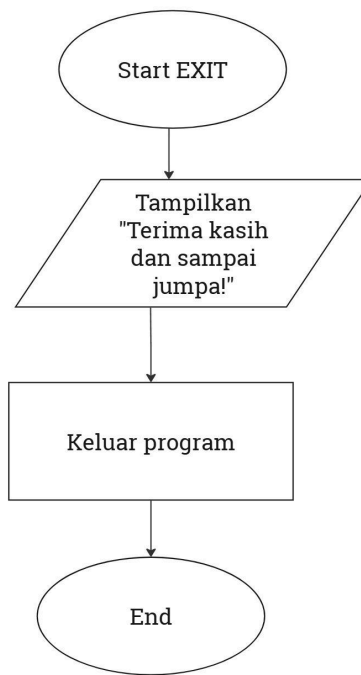
```
1. Procedure skipAntrian(dokterUsername: string)
2. { I.S. Pasien sudah login dan ingin memajukan posisi mereka dalam
  antrian. }
3. { F.S. Pasien berhasil maju ke depan antrian atau menunjukkan pesan
  kesalahan. }
4.
5.   KAMUS
6.     dokterUsername: string
7.     pasien: AntrianNode*
8.     prev: AntrianNode*
9.     position: integer
10.    ditemukan: boolean
11.
12.   ALGORITMA
13.     if not isLoggedIn or currentUser.role is not "Pasien" then
14.       print "Harap login terlebih dahulu!"
15.       return
16.
17.     for i = 0 to jumlahDokter - 1 do
18.       if dokterQueues[i].dokter.username equals dokterUsername then
19.         pasien = dokterQueues[i].head
20.         prev = NULL
21.         position = 1
22.         while pasien is not NULL do
23.           if pasien.pasien_data.username equals
24.             currentUser.username then
25.               if position equals 1 then
26.                 print "Anda sudah berada di posisi paling depan
27.                 antrian!"
28.                 return
29.               end if
30.               if prev is not NULL then
31.                 prev.next = pasien.next
32.               end if
33.               if dokterQueues[i].tail equals pasien then
34.                 dokterQueues[i].tail = prev
35.               end if
36.               pasien.next = dokterQueues[i].head
37.               dokterQueues[i].head = pasien
38.               if dokterQueues[i].jumlah_antrian equals 1 then
39.                 dokterQueues[i].tail = dokterQueues[i].head
40.               end if
41.               print "Anda berhasil maju ke depan antrian Dr. " +
42.                 dokterQueues[i].dokter.username
43.               return
44.             end if
45.             prev = pasien
46.             pasien = pasien.next
47.             position = position + 1
48.           end while
49.           print "Skip antrian gagal! Anda tidak sedang terdaftar dalam
50.           antrian Dr. " + dokterUsername
51.           return
52.         end if
53.       end for
54.       print "Dokter " + dokterUsername + " tidak ditemukan."
55. }
```

```

55. { F.S. Pasien berhasil keluar dari antrian atau menunjukkan pesan
    kesalahan.}
56.
57.   KAMUS
58.     dokterUsername: string
59.     pasien: AntrianNode*
60.     prev: AntrianNode*
61.
62.   ALGORITMA
63.     if not isLoggedIn then
64.       print "Harap login terlebih dahulu!"
65.       return
66.
67.     for i = 0 to jumlahDokter - 1 do
68.       if dokterQueues[i].dokter.username equals dokterUsername then
69.         pasien = dokterQueues[i].head
70.         prev = NULL
71.         while pasien is not NULL do
72.           if pasien.pasien_data.username equals
currentUser.username then
73.             if prev is NULL then
74.               dokterQueues[i].head = pasien.next
75.             else
76.               prev.next = pasien.next
77.             end if
78.             if dokterQueues[i].tail equals pasien then
79.               dokterQueues[i].tail = prev
80.             end if
81.             free(pasien)
82.             dokterQueues[i].jumlah_antrian =
dokterQueues[i].jumlah_antrian - 1
83.             if dokterQueues[i].head is NULL then
84.               dokterQueues[i].tail = NULL
85.             end if
86.             print "Anda berhasil keluar dari antrian Dr. " +
dokterQueues[i].dokter.username + " di ruangan " +
dokterQueues[i].dokter.kodeRuangan
87.             return
88.           end if
89.           prev = pasien
90.           pasien = pasien.next
91.         end while
92.         print "Pembatalan antrian gagal! Anda tidak sedang terdaftar
dalam antrian manapun."
93.         return
94.       end if
95.     end for
96.     print "Dokter " + dokterUsername + " tidak ditemukan."

```

- **F18 - Exit**



Gambar 2.2.2.21 *Flowchart* Exit

```
1. Procedure exit_program()
2. { I.S. Pengguna memutuskan untuk keluar dari program }
3. { F.S. Program akan berhenti dan menampilkan pesan perpisahan }
4.
5. KAMUS
6.   Tidak ada variabel lokal
7.
8. ALGORITMA
9.   output "Terima kasih dan sampai jumpa!"
10.  exit(0)
11. end Procedure
```

BAB III

IMPLEMENTASI DAN PENGUJIAN

3.1. Checklist Progres Fitur

Setelah merancang dan mengimplementasikan berbagai fitur utama dalam sistem, tahap selanjutnya adalah melakukan pencatatan dan evaluasi terhadap progres pengerjaan masing-masing fitur. Tabel berikut menunjukkan status terkini dari tiap fitur yang telah dikembangkan, mencakup tahapan desain, implementasi, serta pengujian.

Tabel 3.1.1 Progres fitur


Fitur	Desain	Implementasi	Testing
F01- Login	V	V	V
F02 - Register Pasien	V	V	V
F03 - Logout	V	V	V
F04 - Lupa Password	V	V	V
F05 - Menu & Help	V	V	V
F06 - Denah Rumah Sakit	V	V	V
F07 - Lihat User	V	V	V
F08 - Cari User	V	V	V
F09 - Lihat Antrian	V	V	V
F10 - Tambah Dokter	V	V	V
F11 - Diagnosis	V	V	V
F12 - Ngobatin	V	V	V

F13 - 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

3.2. Pembagian Tugas Anggota

Untuk mempermudah penyelesaian tugas besar ini, setiap anggota tim diberikan tanggung jawab sesuai kompetensinya masing-masing. Rincian pembagian tugas tersebut disajikan pada tabel berikut.

Tabel 3.2.1 Pembagian tugas

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
Welcome Page		18224117	18224117	18224117
F01- Login	<pre>>>> LOGIN Username: zeru Password: pass77 Selamat datang Manager zeru! >>> LOGIN Username: neronimo Password: pass1010 Selamat pagi Dokter neronimo!</pre>	18224053	18224053	18224061, 18224117

	<pre>>>> LOGIN Username: gro Password: pass22 Selamat pagi gro! Ada keluhan apa ? >>> LOGIN Username: zeru Password: bukanpasszeru Password salah untuk pengguna yang bernama zeru!</pre>			
F02 - Register Pasien	<pre>>>> REGISTER Username: jini Password: ayamgoreng Pasien jini berhasil ditambahkan!</pre>	18224053	18224053	18224061, 18224117
F03 - Logout	<pre>>>> LOGOUT Sampai jumpa!</pre>	18224053	18224053	18224061, 18224117
F04 - Lupa Password	<pre>>>> LUPA_PASSWORD Username: zeru Kode Unik: zeru Halo Manager zeru, silakan daftarkan ulang password anda! Password Baru: pass777</pre>	18224053	18224053	18224061, 18224117
F05 - Menu & Help	<pre>===== MENU (Page 1 of 4) ===== 1. LOGOUT: Keluar dari akun yang sedang digunakan 2. LUPA_PASSWORD: Membuat password baru 3. LIHAT_DENAH: Lihat ruangan di RS serta detailnya 4. LIHAT_USER: Melihat semua pengguna (Dokter/Pasien) [n] Next page [p] Previous page [q] Quit help >>> ===== MENU (Page 2 of 4) ===== 5. LIHAT_PASIEN: Melihat data semua pasien 6. LIHAT_DOKTER: Melihat data semua dokter 7. CARI_USER: Mencari data spesifik 8. CARI_PASIEN: Mencari data pasien spesifik [n] Next page [p] Previous page [q] Quit help >>> ===== MENU (Page 3 of 4) ===== 9. CARI_DOKTER: Mencari data dokter spesifik 10. TAMBAH_DOKTER: Mendaftarkan dokter baru ke sistem 11. ASSIGN_DOKTER: Menempatkan dokter ke dalam ruangan tertentu 12. LIHAT_SEMUA_ANTRIAN: Melihat denah dan antrian ruangan [n] Next page [p] Previous page [q] Quit help >>> ===== MENU (Page 4 of 4) ===== 13. LIHAT_FINANSIAL: Melihat kondisi keuangan rumah sakit 14. UBAH_DENAH: Mengubah ukuran denah rumah sakit 15. PINDAH_DOKTER: Memindahkan dokter antar ruangan [n] Next page [p] Previous page [q] Quit help >>> >>> q Footnote: 1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar 2. Jangan lupa untuk memasukkan input yang valid</pre>	18224017	18224017	18224017, 18224117
F06 - Denah Rumah Sakit	<pre>>>> 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 +---+---+---+---+---+ E E1 E2 E3 E4 E5 +---+---+---+---+---+</pre>	18224045	18224045, 18224117	18224061, 18224117

F07 - Lihat User

```
>>> LIHAT_USER
Urutkan berdasarkan?
1. ID
2. Nama
>>> Pilihan: 1

Urutan sort?
1. Ascending (A to Z atau terkecil)
2. Descending (Z to A atau terbesar)
>>> Pilihan: 1

Menampilkan pasien dengan ID terurut ascending...
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 | neronimo | Dokter | -
11 | ciciko | Dokter | -
12 | cacako | Dokter | -
13 | krocket | Dokter | -
15 | risol | Dokter | -
16 | tobokan | Pasien | -
20 | popokan | Pasien | -
24 | ropik | Pasien | -
88 | zeru | Manager | -
100 | pendatang | Pasien | -
```

```
>>> LIHAT_USER
Urutkan berdasarkan?
1. ID
2. Nama
>>> Pilihan: 2

Urutan sort?
1. Ascending (A to Z atau terkecil)
2. Descending (Z to A atau terbesar)
>>> Pilihan: 2

Menampilkan pasien dengan nama terurut descending...
ID | Nama | Role | Penyakit
-----
88 | zeru | Manager | -
7 | tuart | Pasien | -
16 | tobokan | Pasien | -
9 | tobo | Pasien | -
1 | stewart | Pasien | -
24 | 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 | krocket | Dokter | -
3 | kebin | Pasien | -
2 | gro | Pasien | COVID-19
11 | ciciko | Dokter | -
12 | cacako | Dokter | -
```


18224017

18224117,
18224017

18224061,
18224117

	<pre> >>> LIHAT_USER Urutkan berdasarkan? 1. ID 2. Nama >>> Pilihan: 1 Urutan sort? 1. Ascending (A to Z atau terkecil) 2. Descending (Z to A atau terbesar) >>> Pilihan: 2 Menampilkan pasien dengan ID terurut descending... ID Nama Role Penyakit ----- 100 pendatang Pasien - 88 zeru Manager - 24 ropik Pasien - 20 popokan Pasien - 16 tobokan Pasien - 15 risol Dokter - 13 kroket Dokter - 12 cacako Dokter - 11 ciciko Dokter - 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 - >>> LIHAT_USER Urutkan berdasarkan? 1. ID 2. Nama >>> Pilihan: 2 Urutan sort? 1. Ascending (A to Z atau terkecil) 2. Descending (Z to A atau terbesar) >>> Pilihan: 1 Menampilkan pasien dengan nama terurut ascending... ID Nama Role Penyakit ----- 12 cacako Dokter - 11 ciciko Dokter - 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 Diabetes Mellitus 20 popokan Pasien - 15 risol Dokter - 24 ropik Pasien - 1 stewart Pasien - 9 tobo Pasien - 16 tobokan Pasien - 7 tuart Pasien - 88 zeru Manager - </pre>			
F08 - Cari User	<pre> >>> CARI_USER Cari berdasarkan? 1. ID 2. Nama >>> Pilihan: 1 >>> Masukkan nomor ID user: 1 ID Nama Role Penyakit ----- 1 stewart Pasien >>> CARI_USER Cari berdasarkan? 1. ID 2. Nama >>> Pilihan: 2 >>> Masukkan nama user: risol Mencari pengguna dengan nama risol... ID Nama Role Penyakit ----- 15 risol Dokter - </pre>	18224017	18224117	18224061, 18224117

F09 - Lihat Antrian	<pre> >>> LIHAT_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 +-----+ E E1 E2 E3 E4 E5 +-----+ ===== Ruangan A2 ===== Kapasitas : 3 Dokter : Dr. Neronimo Pasien di dalam ruangan: Tidak ada pasien di dalam ruangan saat ini. Pasien di antrian: Tidak ada pasien di antrian saat ini. </pre>	18224017	18224061, 18224117	18224061, 18224117
F10 - Tambah Dokter	<pre> >>> ASSIGN_DOKTER Username: risol Kode Ruangan (contoh: A1): A3 Dokter risol berhasil diassign ke ruangan A3! >>> TAMBAH_DOKTER Username: simba Password: pass90 Dokter simba berhasil ditambahkan! </pre>	18224045, 18224117	18224045, 18224117	18224061, 18224117
F11 - Diagnosis	<pre> >>> DIAGNOSIS Memeriksa pasien terdepan: Stewart Stewart terdiagnosa penyakit COVID-19! </pre>	18224061	18224061, 18224117	18224061, 18224117
F12 - Ngobatin	<pre> >>> NGOBATIN Dokter sedang mengobati pasien! Pasien memiliki penyakit COVID-19. Obat yang harus diberikan: 1. Remdesivir </pre>	18224061	18224061, 18224117	18224061, 18224117
F13 - Aku Boleh Pulang Ga, Dok	<pre> >>> PULANGDOK Dokter sedang memeriksa keadaanmu... Maaf, tapi kamu masih belum bisa pulang! Urutan minuman obat yang diharapkan: Remdesivir Urutan obat yang kamu minum: Sisa nyawa Anda: ●●○ (2 nyawa) Silahkan kunjungi dokter untuk meminta penawar yang sesuai ! </pre>	18224117, 18224061	18224117, 18224061	18224061, 18224117
F14 - Daftar Check-Up	<pre> >>> DAFTAR_CHECKUP Silakan masukkan data check-up Anda: Suhu Tubuh (Celcius): 36.1 Tekanan Darah (sistol/diastol, contoh 120 80): 92 77 Detak Jantung (bpm): 66 Saturasi Oksigen (%): 93.7 Kadar Gula Darah (mg/dL): 127 Berat Badan (kg): 52.4 Tinggi Badan (cm): 177 Kadar Kolesterol (mg/dL): 193 Trombosit (ribu/uL, contoh 150): 328 Berikut adalah daftar dokter yang tersedia: 1. Dr. Neronimo - Ruangan A2 (Antrian: 0 orang) - Aura: 0 2. Dr. Ciciko - Ruangan N/A (Antrian: 0 orang) - Aura: 0 3. Dr. Caciko - Ruangan N/A (Antrian: 0 orang) - Aura: 0 4. Dr. Kroket - Ruangan N/A (Antrian: 0 orang) - Aura: 0 5. Dr. Risol - Ruangan N/A (Antrian: 0 orang) - Aura: 0 Pilih dokter (1-5): 1 Pendaftaran check-up berhasil! Anda terdaftar pada antrian Dr. Neronimo di ruangan A2. Posisi antrian Anda: 1 </pre>	18224117	18224117	18224061, 18224117
F15 - Antrian Saya	<pre> >>> ANTRIAN Anda belum terdaftar dalam antrian check-up! Silakan daftar terlebih dahulu dengan command DAFTAR_CHECKUP. </pre>	18224117	18224117	18224061, 18224117

F16 - Minum Obat	<pre>>>> MINUM_OBAT Inventory obat kosong! Tidak ada obat yang bisa diminum.</pre>	18224061, 18224117	18224061	18224061, 18224117
F17 - Minum Penawar	<pre>>>> PENAWAR Perut kosong!! Belum ada obat yang dimakan.</pre>	18224061, 18224117	18224061, 18224117	18224061, 18224117
F18 - Exit		18224117		18224117
B02 - Denah Dinamis	<pre>>>> ubah_denah Masukkan jumlah baris denah baru: 3 Masukkan jumlah kolom denah baru: 3 Denah rumah sakit berhasil diubah menjadi 3 baris dan 3 kolom. >>> lihat_denah 1 2 3 +-----+-----+ A A1 A2 A3 +-----+-----+ B B1 B2 B3 +-----+-----+ C C1 C2 C3 +-----+-----+ >>> ubah_denah Masukkan jumlah baris denah baru: 4 Masukkan jumlah kolom denah baru: 4 Denah rumah sakit berhasil diubah menjadi 4 baris dan 4 kolom. >>> lihat_denah 1 2 3 4 +-----+-----+-----+ A A1 A2 A3 A4 +-----+-----+-----+ B B1 B2 B3 B4 +-----+-----+-----+ C C1 C2 C3 C4 +-----+-----+-----+ D D1 D2 D3 D4 +-----+-----+-----+</pre>	18224117	18224061, 18224117	18224061, 18224117
B03 - Aura		18224117	18224061, 18224117	18224061, 18224117

LAMPIRAN

Form MoM Asistensi Tugas Besar
IF1210/Algoritma dan Pemrograman 1
Sem. 2 2024/2025

Nomor Asistensi : 1
No. Kelompok/Kelas : G / K-03
Tanggal asistensi : 6 Mei 2025

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
1	18224017 / Naomi Azzahra
2	18224045 / Muthia Ariesta Anggraeni
3	18224053 / Nathan Pasha Athallah
4	18224061 / Riantama Putra
5	18224117 / Stephanie Mae
Asisten pembimbing	NIM / Nama
	13522045 / Elbert Chailes

Catatan Asistensi:

Rangkuman Diskusi
<ul style="list-style-type: none">- Membahas <i>command</i> atau <i>workflow</i> Github, seperti <i>commit</i>, <i>add</i>, <i>clone</i>, <i>pull</i>, dan <i>push</i>.- Memberi saran perbaikan laporan, khususnya bagian perencanaan implementasi. <p>QnA:</p> <ol style="list-style-type: none">1. Q: Apa itu <i>release tag</i>? A: <i>Release tag</i> seperti checkpoint versi, penanda versi stabil dalam repo.2. Q: Apakah butuh file eksternal untuk disimpan saat exit? A: Tidak perlu. Sistem tidak menyimpan dan memuat file, exit langsung berhenti.3. Q: Apakah format demo harus persis seperti contoh? A: Sebaiknya ikuti contoh, tapi boleh fleksibel selama sesuai fungsionalitas. Kode unik untuk demo akan dihasilkan sistem, bukan ditentukan user.
Tindak Lanjut
Diskusi kelompok perencanaan implementasi, dekomposisi, dan pembagian tugas yang lebih terstruktur

Dokumentasi

meet.google.com/kek-owof-pfu?pli=1&authuser=1

Elbert Challes (Presenting)

The diagram is a hand-drawn flowchart on a whiteboard. It starts with 'Foo' on the left, which points to 'Subject/object'. Below 'Subject/object' is '1. type struct'. An arrow points from 'Subject/object' to 'field'. From 'field', an arrow points to 'modu -> modular'. Below 'modu -> modular' is '1. type struct' and 'robot'. An arrow points from 'modu -> modular' to 'FOU - FID'. From 'FOU - FID', an arrow points to 'robot'. Below 'robot' is 'save' and 'test'. An arrow points from 'robot' to 'save' and 'test'. The diagram is titled 'Bench (Arrival)' at the top left and 'Bar' at the top right.

Elbert Challes

18224117 Ste...

18224061 Ria...

18224053 Na...

18224017 Na...

18224045 M...

9:42 PM | Milestone I - Tugas Besar Algoritma...

21:42 Tuesday 06/05/2023

Form MoM Asistensi Tugas Besar
IF1210/Algoritma dan Pemrograman 1
Sem. 2 2024/2025

Nomor Asistensi : 2
No. Kelompok/Kelas : G / K-03
Tanggal asistensi : 24 Mei 2025

Anggota kelompok		NIM / Nama (Hanya yang Hadir)
1		18224017 / Naomi Azzahra
2		18224045 / Muthia Ariesta Anggraeni
3		18224053 / Nathan Pasha Athallah
4		18224061 / Riantama Putra
5		18224117 / Stephanie Mae
Asisten pembimbing		NIM / Nama
		13522045 / Elbert Chailes

Catatan Asistensi:

Rangkuman Diskusi
<ul style="list-style-type: none"> - <i>Rename branch</i>: Menggunakan <i>feat</i>, <i>refer</i> ke dokumen - Bersihkan <i>commit</i>: <i>commit</i> harus jelas (apa, dimana, dan bagaimana) - <i>Pull request</i>: Jangan langsung digabung kodenya, bisa kasih saran satu sama lain - Memberikan gambaran pelaksanaan demo - login dan logout disatukan di user-management.c
Tindak Lanjut
Melanjutkan progres sesuai ketentuan dan membenarkan <i>commit</i>
Dokumentasi

meet.google.com/zzc-iign-rjx?pli=1&authuser=1

Elbert Challes (Presenting)

8:41 PM | [IF1210] Asistensi_2_K03-G

20:41 Saturday 24/05/2025