

LAPORAN TUGAS BESAR
PROGRAM SIMULASI SISTEM INFORMASI RUMAH SAKIT NIMONS



ALGORITMA DAN PEMROGRAMAN 1
IF1210

Disusun Oleh:

Kloce Paul William Saragih	13524040
Suryani Mulia Utami	13524042
Narendra Dharma Wistara M.	13524044
Bernhard Aprillio Pramana	13524074
Dylan Gregory Tondang	13524118

KELOMPOK D K02
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2025

HALAMAN PERNYATAAN KELOMPOK

Kami di bawah ini menyatakan bahwa kami mengerjakan tugas besar ini dengan sejujurnya, 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.

Tertanda,

- Kloce Paul William Saragih (13524040)
- Suryani Mulia Utami (13524042)
- Narendra Dharma Wistara (13524044)
- Bernhard Aprillio Pramana (13524074)
- Dylan Gregory Tondang (13524118)

DAFTAR ISI

HALAMAN PERNYATAAN KELOMPOK	1
DAFTAR ISI.....	1
DAFTAR TABEL	1
1. Deskripsi Persoalan	1
2. Rencana Implementasi	1
3. Deskripsi Implementasi Fitur	11
3.1 F01 – Login	11
3.2 F02 – Register Pasien.....	11
3.3 F03 – Logout	11
3.4 F04 – Lupa Password	11
3.5 F05 – Menu & Help.....	11
3.6 F06 - Denah Rumah Sakit	12
3.7 F07 – Lihat User	12
3.8 F08 – Cari User.....	12
3.9 F09 – Lihat Antrian.....	12
3.10 F10 – Tambah Dokter	12
3.11 F11 – Diagnosis	13
3.12 F12 – Ngobatin	13
3.13 F13 – Aku boleh pulang ga, dok?	13
3.14 F14 – Daftar Check-Up.....	13
3.15 F15 – Antrian Saya!.....	14
3.16 F16 – Minum Obat.....	14
3.17 F17 – Minum Penawar	14
3.18 F18 – Exit & D04 – Save	15
3.19 D03 – Load.....	15
3.20 B02 – Denah Dinamis	15
3.21 B03 – Aura! & B04 – Banarich!!!	15
3.22 B06 – Mainin Antrian	16
4. Pembagian Kerja Anggota Kelompok	16
5. Checklist Fitur dan Implementasi.....	18

6.	Desain Command.....	19
7.	Kamus Data	23
8.	Kamus Makro.....	26
9.	Dekomposisi Algoritmik dan Fungsional Program	29
9.1	F01 – Login	29
9.2	F02 – Register Pasien.....	30
9.3	F03 – Logout	31
9.4	F04 – Lupa Password	32
9.5	F05 – Menu & Help	33
9.6	F06 – Denah Rumah Sakit.....	34
9.7	F07 – Lihat User	35
9.8	F08 – Cari User.....	37
9.9	F09 – Lihat Antrian.....	39
9.10	F10 – Tambah Dokter	40
9.11	F11 – Diagnosis	41
9.12	F12 – Ngobatin	44
9.13	F13 – Aku boleh pulang ga, dok?	46
9.14	F14 – Daftar Check-Up.....	47
9.15	F15 – Antrian Saya!.....	48
9.16	F16 – Minum Obat	49
9.17	F17 – Minum Penawar	50
9.18	F18 – Exit & D04 – Save	51
9.19	D03 – Load.....	53
9.20	B02 – Denah Dinamis	54
9.21	B06 – Mainin Antrian	59
10.	Spesifikasi Modul/Prosedur/Fungsi	61
10.1	F01 – Login	61
10.2	F02 – Register.....	62
10.3	F03 – Logout	62
10.4	F04 – Lupa Password	63
10.5	F05 – Menu & Help	64
10.6	F06 – Denah Rumah Sakit.....	66

10.7	F07 – Lihat User	67
10.8	F08 – Cari User.....	68
10.9	F09 – Lihat Antrian.....	69
10.10	F10 – Tambah Dokter.....	71
10.11	F11 – Diagnosis	72
10.12	F12 – Ngobatin	76
10.13	F13 – Aku boleh pulang ga, dok?	78
10.14	F14 – Daftar Checkup	82
10.15	F15 – Antrian Saya!	87
10.16	F16 – Minum Obat.....	88
10.17	F17 – Minum Penawar	90
10.18	F18 – Exit & D04 – Save	91
10.19	D03 – Load	94
10.20	B02 - Denah Dinamis.....	95
10.21	B04 - Banarich.....	97
10.22	B06 - Mainin Antrian.....	99
11.	Hasil Pengujian	113
11.1	F01 – Login	113
11.2	F02 - Register Pasien	115
11.3	F03 – Logout	117
11.4	F04 - Lupa Password.....	118
11.5	F05 - Menu & Help	120
11.6	F06 - Denah Rumah Sakit	123
11.7	F07 - Lihat User.....	125
11.8	F08 - Cari User	127
11.9	F09 - Lihat Antrian	131
11.10	F10 - Tambah Dokter	132
11.11	F11 – Diagnosis	137
11.12	F12 – Ngobatin	138
11.13	F13 - Aku boleh pulang ga, dok?	140
11.14	F14 - Daftar Check-Up	142
11.15	F15 - Antrian Saya!	144

11.16	F16 - Minum Obat	146
11.17	F17 - Minum Penawar	148
11.18	F18 – Exit & D04 – Save	149
11.19	D03 – Load.....	152
11.20	B02 – Denah Dinamis	154
11.21	B03 – Aura!	157
11.22	B04 – Banarich!!!!.....	158
11.23	B06 - Mainin Antrian.....	161
11.24	Perintah Tidak Valid	164
LAMPIRAN.....		166

DAFTAR TABEL

Tabel 1 Rencana Implementasi ADT	10
Tabel 2 Pembagian Kerja Kelompok.....	18
Tabel 3 Checklist Fitur dan Implementasi	19
Tabel 4 Design Command	23
Tabel 5 Definisi Aksesori	28
Tabel 6 Spesifikasi Modul/Prosedur/Fungsi	112
Tabel 7 Hasil Pengujian F01.....	114
Tabel 8 Hasil Pengujian F02.....	116
Tabel 9 Hasil Pengujian F03.....	117
Table 10 Hasil Pengujian F04	119
Tabel 11 Hasil Pengujian F05	122
Tabel 12 Hasil Pengujian F06	124
Tabel 13 Hasil Pengujian F07	127
Tabel 14 Hasil Pengujian F08	130
Tabel 15 Hasil Pengujian F09	131
Tabel 16 Hasil Pengujian F10	136
Tabel 17 Hasil Pengujian F11	137
Tabel 18 Hasil Pengujian F12	139
Tabel 19 Hasil Pengujian F13	141
Tabel 20 Hasil Pengujian F14	143
Tabel 21 Hasil Pengujian F15	145
Tabel 22 Hasil Pengujian F16	147
Tabel 23 Hasil Pengujian F17	148
Tabel 24 Hasil Pengujian F18 & D04.....	151
Table 25 Hasil Pengujian D03.....	153
Tabel 26 Hasil Pengujian B02.....	156
Tabel 27 Hasil Pengujian B03.....	157
Table 28 Hasil Pengujian B04.....	160
Tabel 29 Hasil Pengujian B06.....	163
Tabel 30 Hasil Pengujian Perintah Tidak Valid	165

DAFTAR GAMBAR

Gambar 1 Flowchart Login.....	29
Gambar 2 Flowchart Register	30
Gambar 3 Flowchart Logout.....	31
Gambar 4 Flowchart PasswordUpdate.....	32
Gambar 5 Flowchart Help	33
Gambar 6 Flowchart Denah.....	34
Gambar 7 Flowchart PilihanLihat	35
Gambar 8 Flowchart MenuLihat.....	36
Gambar 9 Flowchart PilihanCari	37
Gambar 10 Flowchart CariUser	38
Gambar 11 Flowchart LihatAntrian	39
Gambar 12 Flowchart TambahDokter.....	40
Gambar 13 Flowchart Diagnosis 1	41
Gambar 14 Flowchart Diagnosis 2	42
Gambar 15 Flowchart Diagnosis 3	43
Gambar 16 Flowchart CariDokter	44
Gambar 17 Flowchart Ngobatin	45
Gambar 18 Flowchart PulangDok	46
Gambar 19 Flowchart DaftarCheckup	47
Gambar 20 Flowchart CekAntrian	48
Gambar 21 Flowchart MinumObat	49
Gambar 22 Flowchart MinumPenawar	50
Gambar 23 Flowchart Exit	51
Gambar 24 Flowchart Save.....	52
Gambar 25 Flowchart Load	53
Gambar 26 Flowchart UbahDenah	54
Gambar 27 Flowchart PindahDokter	55
Gambar 28 Flowchart LihatDompot	56
Gambar 29 Flowchart LihatFinansial	57
Gambar 30 Flowchart Gacha	58
Gambar 31 Flowchart SkipAntrian	59

Gambar 32 Flowchart KeluarAntrian	60
Gambar 34 Hasil Pengujian Login: Kasus 1.....	113
Gambar 35 Hasil Pengujian Login: Kasus 2.....	113
Gambar 36 Hasil Pengujian Login: Kasus 3.....	114
Gambar 37 Hasil Pengujian Login: Kasus 4.....	114
Gambar 38 Hasil Pengujian Register: Kasus 1	115
Gambar 39 Hasil Pengujian Register: Kasus 2	115
Gambar 40 Hasil Pengujian Register: Kasus 3	116
Gambar 41 Hasil Pengujian Logout: Kasus 1.....	117
Gambar 42 Hasil Pengujian Logout: Kasus 2.....	117
Gambar 43 Hasil Pengujian LupaPassword: Kasus 1.....	118
Gambar 44 Hasil Pengujian LupaPassword: Kasus 2.....	119
Gambar 45 Hasil Pengujian LupaPassword: Kasus 3.....	119
Gambar 46 Hasil Pengujian Help: Kasus 1	120
Gambar 47 Hasil Pengujian Help: Kasus 2	120
Gambar 48 Hasil Pengujian Help: Kasus 3	121
Gambar 49 Hasil Pengujian Help: Kasus 4	121
Gambar 50 Hasil Pengujian Help: Kasus 5	122
Gambar 51 Hasil Pengujian LihatDenah	123
Gambar 52 Hasil Pengujian LihatRuangan: Kasus 1	123
Gambar 53 Hasil Pengujian LihatRuangan: Kasus 2	124
Gambar 54 Hasil Pengujian LihatRuangan: Kasus 3	124
Gambar 55 Hasil Pengujian LihatUser: Kasus 1.....	125
Gambar 56 Hasil Pengujian LihatUser: Kasus 2.....	126
Gambar 57 Hasil Pengujian LihatPasien	126
Gambar 58 Hasil Pengujian LihatDokter	127
Gambar 59 Hasil Pengujian CariUser: Kasus 1.....	128
Gambar 60 Hasil Pengujian CariUser: Kasus 2.....	128
Gambar 61 Hasil Pengujian CariUser: Kasus 3.....	129
Gambar 62 Hasil Pengujian CariPasien	129
Gambar 63 Hasil Pengujian CariDokter.....	130
Gambar 64 Hasil Pengujian LihatSemuaAntrian	131

Gambar 65 Hasil Pengujian TambahDokter: Kasus 1	132
Gambar 66 Hasil Pengujian TambahDokter: Kasus 2	133
Gambar 67 Hasil Pengujian AssignDokter: Kasus 1.....	133
Gambar 68 Hasil Pengujian AssignDokter: Kasus 2.....	134
Gambar 69 Hasil Pengujian AssignDokter: Kasus 3.....	134
Gambar 70 Hasil Pengujian AssignDokter: Kasus 4.....	135
Gambar 71 Hasil Pengujian AssignDokter: Kasus 5.....	136
Gambar 72 Hasil Pengujian Diagnosis: Kasus 1	137
Gambar 73 Hasil Pengujian Diagnosis: Kasus 2	137
Gambar 74 Hasil Pengujian Ngobatin: Kasus 1.....	138
Gambar 75 Hasil Pengujian Ngobatin: Kasus 2.....	138
Gambar 76 Hasil Pengujian Ngobatin: Kasus 3.....	139
Gambar 77 Hasil Pengujian PulangDok: Kasus 1	140
Gambar 78 Hasil Pengujian PulangDok: Kasus 2	140
Gambar 79 Hasil Pengujian PulangDok: Kasus 3	141
Gambar 80 Hasil Pengujian PulangDok: Kasus 4	141
Gambar 81 Hasil Pengujian DaftarCheckup: Kasus 1.....	142
Gambar 82 Hasil Pengujian DaftarCheckup: Kasus 2.....	142
Gambar 83 Hasil Pengujian DaftarCheckup: Kasus 3.....	143
Gambar 84 Hasil Pengujian Antrian: Kasus 1.....	144
Gambar 85 Hasil Pengujian Antrian: Kasus 2.....	144
Gambar 86 Hasil Pengujian Antrian: Kasus 3.....	145
Gambar 87 Hasil Pengujian MinumObat: Kasus 1	146
Gambar 88 Hasil Pengujian MinumObat: Kasus 2	147
Gambar 89 Hasil Pengujian MinumObat: Kasus 3	147
Gambar 90 Hasil Pengujian MinumPenawar: Kasus 1.....	148
Gambar 91 Hasil Pengujian MinumPenawar: Kasus 2.....	148
Gambar 92 Hasil Pengujian Exit: Kasus 1	149
Gambar 93 Hasil Pengujian Exit: Kasus 2	150
Gambar 94 Hasil Pengujian Exit: Kasus 3	150
Gambar 95 Hasil Pengujian Save.....	151
Gambar 96 Hasil Pengujian Load : Kasus 1.....	152

Gambar 97 Hasil Pengujian Load : Kasus 2.....	152
Gambar 98 Hasil Pengujian Load : Kasus 3.....	153
Gambar 99 Hasil Pengujian UbahDenah: Kasus 1.....	154
Gambar 100 Hasil Pengujian UbahDenah: Kasus 2	154
Gambar 101 Hasil Pengujian PindahDokter: Kasus 1	155
Gambar 102 Hasil Pengujian PindahDokter: Kasus 2	155
Gambar 103 Hasil Pengujian PindahDokter: Kasus 3	156
Gambar 104 Hasil Pengujian Aura: Kasus 1 (DaftarCheckup).....	157
Gambar 105 Hasil Pengujian Aura: Kasus 2 (LihatDokter)	157
Gambar 106 Hasil Pengujian Banarich: Kasus 1 (DaftarCheckup)	158
Gambar 107 Hasil Pengujian Banarich: Kasus 2 (DaftarCheckup)	158
Gambar 108 Hasil Pengujian Banarich: Kasus 3 (LihatDompel)	159
Gambar 109 Hasil Pengujian Banarich: Kasus 4 (LihatFinansial)	159
Gambar 110 Hasil Pengujian Banarich: Kasus 5 (Gacha)	160
Gambar 111 Hasil Pengujian SkipAntrian: Kasus 1	161
Gambar 112 Hasil Pengujian SkipAntrian: Kasus 2.....	161
Gambar 113 Hasil Pengujian SkipAntrian: Kasus 3.....	162
Gambar 114 Hasil Pengujian CancelAntrian: Kasus 1	162
Gambar 115 Hasil Pengujian CancelAntrian: Kasus 2.....	161
Gambar 116 Hasil Pengujian Input NonValid: Kasus 1	164
Gambar 117 Hasil Pengujian Input NonValid: Kasus 2	164
Gambar 118 Hasil Pengujian Input NonValid: Kasus 3	165
Gambar 119 Hasil Pengujian Input NonValid: Kasus 4	165

1. Deskripsi Persoalan

Tugas besar ini meminta kami untuk mengembangkan sebuah program simulasi sistem informasi rumah sakit yang dirancang untuk mendukung berbagai kegiatan operasional rumah sakit. Tujuan dari program ini adalah untuk mengatur semua layanan medis di Rumah Sakit Nimons dengan berbagai fitur yang sudah ditentukan. Program ini dibentuk sebagai sistem digital yang mampu menangani data pengguna, beserta jadwal check-up, antrian, denah ruangan, dan banyak lagi. Untuk menyelesaikan persoalan ini dibutuhkan pemanfaatan dari berbagai konsep ADT, seperti List, Set, Map, Queue.

Program ini harus mampu mengelola data pengguna dengan berbagai peran, seperti manajer, dokter, dan pasien. Fitur-fitur yang ada di program ini juga disesuaikan dengan peran yang dimiliki oleh pengguna. Sehingga, seorang pasien tidak dapat mengorganisir ruangan pada rumah sakit, dan hal-hal yang tidak diinginkan lainnya. Sistem harus mendukung berbagai interaksi berbasis perintah, memungkinkan pengguna untuk mengakses informasi terkait rumah sakit, dan memproses data sesuai dengan prosedur yang telah ditentukan. Terdapat banyak fitur yang diimplementasikan pada program ini, seperti login, registrasi, melihat denah rumah sakit, diagnosis otomatis, minum obat, dan banyak lagi.

Persoalan ini dirancang untuk diselesaikan menggunakan materi-materi yang telah diajarkan pada Mata Kuliah Algoritma dan Pemrograman 1 (IF1210) Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung. Materi yang digunakan untuk menyelesaikan persoalan ini antara lain adalah skema percabangan, skema perulangan, modularitas program (fungsi dan prosedur), skema pemrosesan array, skema pemrosesan file, ADT Sederhana, ADT List, ADT Matrix, ADT Queue, ADT Stack, ADT Set, ADT Map, dan ADT Linked List. Persoalan ini diselesaikan secara berkelompok dengan satu kelompok berjumlah lima orang. Program dirancang menggunakan bahasa C, disusun menggunakan GCC dan Makefile, dan dirancang untuk dapat dijalankan dalam sebuah lingkungan virtual unix.

2. Rencana Implementasi

Fitur	Implementasi ADT	Deskripsi Implementasi	Alasan Implementasi
F01 – Login	File external, ADT sederhana	Membaca data dari user.csv dan mencocokkan input username dan	Menggunakan file eksternal memudahkan penyimpanan data

		password dengan isinya.	akun secara permanen. ADT sederhana mempermudah strukturisasi tiap atribut pengguna. Fungsi dan prosedur menjaga agar logika login tetap modular dan mudah diuji.
F02 – Register Pasien	File external, ADT sederhana, Fungsi&Prosedur	Menyimpan data pasien baru ke file user.csv menggunakan prosedur AddUser, serta memeriksa apakah username sudah terdaftar sebelumnya menggunakan fungsi IdxUser.	Menggunakan file eksternal memudahkan penyimpanan data pasien baru secara permanen. ADT sederhana memastikan setiap data pasien tertata dalam format yang sama. Fungsi dan prosedur menjaga agar pengecekan dan penambahan data terpisah dan mudah dikembangkan.
F03 – Logout	-	-	Tidak membutuhkan struktur data tambahan karena hanya mengatur alur kontrol program tanpa mengubah data.

F04 – Lupa Password	File external, Fungsi & Prosedur	Meng-update file user.csv dengan password baru menggunakan prosedur UpdateFile, dan fungsi RunLengthEncoding untuk validasi kode unik.	Membaca&Menulis ke file user.csv terkait role, username, dan password. Fungsi runLengthEncoding untuk validasi kode unik.
F05 – Menu & Help	Fungsi & Prosedur, typedef enum	Mencetak opsi fitur yang dapat digunakan sesuai dengan typedef enum LoggedIn dan pengisian command fitur dengan prosedur askQuestion agar pertanyaan dapat terus ditanyakan	Prosedur digunakan agar pertanyaan untuk pemanggilan fitur dapat diulang dan penggunaan typedef enum berfungsi untuk menyimpan role saat login.
F06 (D01, B02) - Denah Rumah Sakit	File external, ADT matriks, ADT list, Fungsi & Prosedur	Mencetak denah rumah sakit dengan prosedur PrintDenah, mencetak detail ruangan sesuai inputan dari user dengan prosedur PrintRuang, dan mengubah ukuran denah berdasarkan inputan user dengan prosedur UbahDenah.	Menggunakan file external memudahkan penyimpanan data denah permanen. ADT matriks dan ADT list memudahkan penyimpanan data sementara, pengubahan data, dan penggunaan ulang data.

F07 – Lihat User	ADT List	Menampilkan sebuah list berisikan data seluruh pengguna (dokter dan pasien), atau secara spesifik hanya pasien atau hanya dokter. Manager dapat memilih metode pengurutan berdasarkan: ID (Numerik) Nama (Leksikografis, Case-Insensitive).	UserList (ADT List) dipakai karena seluruh data user disimpan dalam array dinamis yang memungkinkan traversal dan pengurutan.
F08 – Cari User	ADT List	Menampilkan sebuah list berisikan data pengguna secara spesifik berdasarkan ID atau Nama melalui perintah	UserList (ADT List) tetap digunakan karena seluruh data user sudah berada dalam list saat program berjalan, memudahkan pencarian. Pendekatan list memungkinkan tampilan hasil pencarian lebih fleksibel dan integrasi mudah dengan fitur tampilan (lihat user).

F09 (D02) – Lihat Antrian	Map, Queue, List, Denah, prosedur	User (dokter atau pasien) dapat melihat detail informasi suatu ruangan berdasarkan namanya. Informasi yang ditampilkan mencakup kapasitas ruangan, nama dokter yang bertugas, daftar pasien yang sedang berada di dalam ruangan, dan daftar pasien yang masih menunggu dalam antrian.	ADT Map digunakan untuk mencocokkan nama ruangan dengan ID dokter yang bertugas di ruangan tersebut, melalui struktur RuangtoDokter. ADT Queue digunakan untuk menyimpan daftar pasien dalam ruangan dan pasien yang masih mengantre. ADT List digunakan melalui Ulist, untuk mengambil informasi akun user.
F10 – Tambah Dokter	ADT set, file external, fungsi & prosedur	User manager dapat menambahkan dokter. Dokter baru bisa login dengan username dan password yang unik. Manager juga dapat melakukan assign ruangan ke dokter tertentu yang belum memiliki ruangan.	ADT Set digunakan untuk menyimpan dan memastikan tidak ada duplikasi data dokter, juga mempermudah proses pencarian dokter berdasarkan

			ID atau username saat melakukan <i>assign</i> ruangan. Data dokter yang telah dimasukkan juga akan disimpan ke dalam file eksternal,
F11 – Diagnosis	ADT list, queue, file external, fungsi & prosedur	Menentukan jenis penyakit pada pasien dengan membandingkan kondisi pasien dengan batasan kondisi	Membaca data kondisi pasien dan menyesuaikannya dengan batasan kondisi yang ada pada file penyakit.csv.
F12 – Ngobatin	ADT list, linked list, queue, file external, fungsi & prosedur	Memberikan obat kepada pasien yang sudah terdiagnosis dengan file obatpenyakit	Membaca data penyakit dan menyesuaikan pemberian obat kepada pasien
F13 – Aku boleh pulang ga, dok?	Stack, Queue, List, Set, prosedur dan fungsi	Pasien dapat melakukan pemeriksaan terakhir untuk melihat apakah mereka sudah boleh pulang. Sistem akan mengecek apakah	ADT Set dan List digunakan untuk menyimpan dan mengakses data pasien, dokter, penyakit, dan obat dari struktur data

		<p>pasien sudah mendapatkan diagnosis, sudah meminum obat secara lengkap dan dalam urutan yang benar. Jika semua syarat terpenuhi, pasien dianggap sembuh dan dipulangkan dari ruangan dokter.</p>	<p>global seperti Ulist, Olist, Plist, dan OPlist.</p> <p>Fungsi riwayat() digunakan untuk mengambil nama penyakit terakhir pasien.</p> <p>ADT Queue dipakai untuk mengeluarkan pasien dari antrian dokter ketika pasien berhasil pulang.</p> <p>ADT Stack digunakan untuk menyimpan urutan obat yang telah diminum pasien (perut). Stack ini kemudian dibandingkan dengan urutan obat seharusnya yang diambil dari file penyakit-obat (OPlist).</p> <p>Proses verifikasi kesembuhan dilakukan dengan mencocokkan urutan ideal (arr1) dan urutan yang dikonsumsi pasien</p>
--	--	--	---

			(arr2). Jika berbeda, pasien tidak boleh pulang dan akan diberi informasi urutan seharusnya serta urutan yang dikonsumsi. Untuk kasus pasien sembuh
F14 – Daftar Check-Up	Queue, List, Set, prosedur & fungsi	Pasien dapat mendaftar untuk pemeriksaan kesehatan (check-up) dengan memilih dokter yang tersedia. Sebelum mendaftar, pasien diwajibkan mengisi data check-up seperti suhu tubuh, tekanan darah, detak jantung, dll. Setelah berhasil mendaftar, pasien akan dimasukkan ke antrian dokter.	ADT queue untuk menyimpan urutan pasien yang telah mendaftar ke dokter tertentu. List untuk menyimpan dan mengakses daftar user dan dokter yang ada di sistem. Set digunakan dalam penyimpanan data user karena setiap user memiliki username dan ID yang unik.
F15 – Antrian Saya!	Queue, Linkedlist, List, Map	Memungkinkan pasien yang telah mendaftar check-up melihat status antrian mereka saat ini. Validasi dilakukan dengan mencocokkan ID	ADT Queue menyimpan urutan pasien yang menunggu check-up di dokter. PasienList, DokterList, Userlist digunakan untuk

		pasien terhadap elemen dalam antrean dokter untuk menentukan status dan posisinya.	mengakses informasi. Sistem akan mencari ID pasien di antrean dokter yang sesuai melalui struktur data Map. Queue diimplementasikan melalui linkedlist
F16 – Minum Obat	ADT stack, list linked list.	Menampilkan ke layar daftar obat di inventoryObat pasien, kemudian memasukkan obat yang dipilih ke stack Perut pasien dan menghapusnya dari inventoryObat	Membaca daftar id obat dari linkedlist inventoryObat, membaca nama obat dari listObat dan mencocokkannya dengan id obat. Push id obat yang diminum ke Stack.
F17 – Minum Penawar	ADT stack, list, linked list.	Menghapus obat terakhir yang diminum pasien dari stack Perut, kemudian menambahkannya kembali ke inventoryObat	Membaca nama obat dari listObat, Pop top element dari Stack, dan memasukkannya ke linkedlist inventoryObat kembali.
F18 – Exit	-	Program terus berlanjut kecuali saat prompt userexit, akan diminra validasi dan program berhenti.	-
D03 – Load	File Eksternal, ADT List	Memasukkan data dari sebuah file	ADT bentukan yang ada di dalam

		eksternal yang berada pada suatu folder ke ADT List yang sudah dibentuk.	program berbentuk list, sehingga menggunakan ADT List memudahkan untuk melakukan penyimpanan ke file
D04 – Save	File Eksternal, ADT List	Menyimpan data dari ADT yang ada di dalam program ke dalam sebuah file eksternal	ADT bentukan yang ada di dalam program berbentuk list, sehingga menggunakan ADT List memudahkan untuk melakukan penyimpanan ke file.
B01 – Git Best Practice	-	-	-
B03 – Aura!	ADT List	Variabel aura disimpan pada list User	Menyesuaikan ADT bentukan untuk User
B04 – Banarich!!!	ADT List	Variabel Banarich disimpan pada list User	Menyesuaikan ADT bentukan untuk User
B06 – Mainin Antrian	ADT Queue	Pasien dapat melakukan <i>skip</i> terhadap antrian sehingga langsung maju ke antrian nomor 1 atau membatalkan antrian (keluar dari antrian)	Menyesuaikan ADT Queue yang digunakan pada F09 dan D02 tentang antrian.

Tabel 1 - Rencana Implementasi ADT

3. Deskripsi Implementasi Fitur

3.1 F01 – Login

Fitur ini memungkinkan pengguna (manager, dokter, atau pasien) masuk ke sistem dengan mencocokkan username dan password yang dimasukkan terhadap data yang tersedia. Sistem memverifikasi peran pengguna dan mengatur status sesi login.

3.2 F02 – Register Pasien

Fitur ini memungkinkan pasien untuk membuat akun baru dengan memasukkan username dan password. Sistem akan memeriksa apakah username sudah digunakan. Jika belum, data pasien ditambahkan ke sistem. Hanya pasien yang bisa melakukan register.

3.3 F03 – Logout

Fitur ini memungkinkan pengguna yang sedang login (manager, dokter, atau pasien) untuk keluar dari akun mereka. Setelah logout, status login direset dan sistem tidak mengizinkan akses fitur lain sampai login kembali dilakukan. Fitur ini hanya bisa digunakan jika ada pengguna yang sedang aktif (logged in).

3.4 F04 – Lupa Password

Fitur ini memungkinkan pengguna (manager, dokter, atau pasien) untuk mengatur ulang password jika lupa. Verifikasi identitas dilakukan dengan mencocokkan kode unik yang merupakan hasil Run-Length Encoding dari username. Jika valid, pengguna dapat memasukkan password baru yang akan menggantikan yang lama.

3.5 F05 – Menu & Help

Fitur ini menampilkan daftar command yang tersedia sesuai peran pengguna (manager, dokter, pasien, atau belum login). Command HELP dapat diakses oleh siapa saja dan berfungsi sebagai panduan penggunaan sistem. Informasi yang ditampilkan disesuaikan dengan hak akses pengguna. Pesan tambahan (footnote) juga ditampilkan untuk mencegah penyalahgunaan fitur.

3.6 F06 (D01) – Denah Rumah Sakit

Fitur ini memungkinkan semua pengguna (manager, dokter, pasien) untuk melihat denah rumah sakit beserta detail informasi setiap ruangan. Ukuran dan konfigurasi denah dibaca dari file konfigurasi. Informasi detail meliputi ID ruangan, jenis ruangan, status, dan data terkait lainnya.

3.7 F07 – Lihat User

Fitur ini hanya dapat digunakan oleh peran manager. Fitur ini memungkinkan manager untuk menampilkan list yang berisi seluruh pengguna yang terdaftar dalam sistem. Perintah ini dapat melakukan filter terhadap list yang ditampilkan, sehingga hanya menampilkan pasien saja atau dokter saja. List yang ditampilkan juga dapat diurutkan berdasarkan ID atau nama dan mengurut naik ataupun mengurut turun.

3.8 F08 – Cari User

Cari User diimplementasikan untuk memberikan akses bagi manager dalam mencari data pengguna secara spesifik. Sistem menyediakan dua metode pencarian utama, yaitu berdasarkan ID menggunakan algoritma binary search (karena data diasumsikan sudah diurutkan berdasarkan ID), dan berdasarkan Nama menggunakan sequential search.

3.9 F09 (D02) – Lihat Antrian

Lihat Antrian diimplementasikan untuk memungkinkan manager melihat seluruh ruangan yang saat ini aktif dengan mencetak denah rumah sakit secara visual, lalu dilanjutkan dengan menampilkan rincian tiap ruangan yang tidak kosong. Fitur mencetak denah rumah sakit secara visual, lalu dilanjutkan dengan menampilkan rincian tiap ruangan yang tidak kosong.

3.10 F10 – Tambah Dokter

Fitur ini memungkinkan manager untuk menambahkan akun dokter baru dengan username dan password yang unik. Validasi keunikan username dilakukan menggunakan struktur data set. Selain itu, manager dapat langsung menetapkan ruangan ke dokter yang belum memiliki ruangan.

3.11 F11 – Diagnosis

Fitur ini digunakan untuk dokter mendiagnosis pasien pertama dalam antriannya secara otomatis. Implementasi dilakukan dengan membaca data kondisi medis pasien seperti suhu tubuh, tekanan darah, kadar gula, dan parameter lainnya dari struktur `UserList`, lalu mencocokkannya dengan kriteria batasan penyakit yang disimpan dalam `penyakit.csv` dan dimuat ke dalam `PenyakitList`. Pencocokan dilakukan secara berurutan, dan jika pasien memenuhi lebih dari satu kriteria penyakit, hanya penyakit pertama yang cocok berdasarkan urutan file yang akan digunakan sebagai hasil diagnosis. Hasil diagnosis kemudian disimpan ke field riwayat pasien, dan pasien dikeluarkan dari antrian dokter menggunakan `ADT Queue`.

3.12 F12 – Ngobatin

Fitur ini digunakan untuk dokter memberikan obat kepada pasien pertama dalam antreannya secara otomatis. Sistem ini bekerja dengan membaca diagnosis pasien dari field riwayat, kemudian mencocokkannya dengan data relasi penyakit dan obat dalam `obat_penyakit.csv`, yang telah dimuat ke dalam struktur `ObatPenyakitList`. Setiap penyakit memiliki daftar obat yang harus diminum secara berurutan, dan obat-obat ini dimasukkan ke `inventoryObat` pasien menggunakan struktur data `LinkedList`.

3.13 F13 – Aku boleh pulang ga, dok?

Fitur ini digunakan untuk pasien yang telah didiagnosis dan menjalani pengobatan berkonsultasi kepada dokter terkait kepulangannya. Fitur ini memeriksa apakah seluruh obat dalam `inventoryObat` pasien telah dikonsumsi sesuai urutan yang benar. Validasi dilakukan dengan membandingkan urutan peminuman obat yang disimpan dalam `stack` perut pasien terhadap urutan obat yang seharusnya. Jika urutan benar dan semua obat telah diminum, pasien dianggap sembuh dan diperbolehkan pulang, sehingga data medis pasien direset seperti saat pertama kali mendaftar (kecuali `username` dan `password`), dan pasien dikeluarkan dari antrian dokter. Jika terjadi kesalahan urutan, pasien dapat menggunakan fitur penawar (F17) terlebih dahulu.

3.14 F14 – Daftar Check-Up

Fitur ini digunakan untuk memungkinkan pasien yang telah login mendaftar pemeriksaan kesehatan (check-up) dengan dokter pilihan. Dalam prosesnya, pasien

terlebih dahulu mengisi data medis dasar seperti tekanan darah (sistol dan diastol), berat badan, dan tinggi badan yang disimpan dalam field terkait pada User. Setelah itu, pasien memilih dokter dari daftar dokter yang tersedia. Sistem kemudian akan menambahkan ID pasien ke dalam antrian dokter menggunakan struktur data Map untuk memetakan ID dokter ke Queue antrian pasien, di mana Queue diimplementasikan sebagai linked list untuk mendukung operasi dinamis secara efisien. Setelah berhasil mendaftar, pasien dapat melihat posisi mereka dalam antrian. Fitur ini memastikan bahwa proses pendaftaran dan manajemen antrian berjalan terstruktur dan efisien dengan dukungan ADT Map dan Queue.

3.15 F15 – Antrian Saya!

Fitur ini diimplementasikan untuk memungkinkan pasien yang telah mendaftar check-up melihat status antrian mereka saat ini. Sistem akan mencari ID pasien di antrian dokter yang sesuai melalui struktur data Map yang memetakan ID dokter ke Queue pasien, dengan Queue diimplementasikan menggunakan linked list. Validasi dilakukan dengan mencocokkan ID pasien terhadap elemen dalam antrian dokter untuk menentukan status dan posisinya.

3.16 F16 – Minum Obat

Fitur ini diimplementasikan untuk memungkinkan pasien meminum obat dari inventory yang telah diberikan oleh dokter. Obat-obatan pasien disimpan dalam struktur data LinkedList sebagai inventory (inventoryObat), dan saat pasien memilih obat untuk diminum, obat tersebut akan dipindahkan ke struktur data Stack perut. Fitur ini memastikan urutan peminuman obat terekam dengan benar dan dapat divalidasi nantinya saat pasien ingin pulang (F13).

3.17 F17 – Minum Penawar

Fitur ini digunakan untuk memungkinkan pasien mengeluarkan obat terakhir yang telah diminum apabila terjadi kesalahan konsumsi. Obat-obatan yang telah diminum disimpan dalam struktur data Stack bernama perut, dan saat fitur ini dijalankan, sistem akan memeriksa apakah stack kosong. Jika kosong, sistem akan menampilkan pesan bahwa perut pasien masih kosong. Jika tidak, sistem akan melakukan operasi pop untuk

mengeluarkan obat paling atas (terakhir diminum) dari Stack, kemudian obat tersebut dimasukkan kembali ke inventoryObat pasien yang disimpan dalam struktur data LinkedList. Fitur ini berguna untuk memperbaiki kesalahan urutan konsumsi obat sebelum pasien meminta izin pulang, serta menjaga akurasi riwayat pengobatan pasien.

3.18 F18 – Exit & D04 – Save

Fitur ini digunakan untuk keluar dari sistem dan mengakhiri sesi program. Sebelum sistem ditutup, pengguna diberikan opsi untuk menyimpan data terlebih dahulu agar riwayat aktivitas tidak hilang. Fitur ini tersedia untuk semua peran (manager, dokter, pasien) dan memastikan data penting tetap tersimpan sebelum program dihentikan.

3.19 D03 – Load

Fitur ini digunakan untuk memasukkan data saat program dijalankan. Sebelum program dijalankan, pengguna harus memasukkan folder data untuk memasukkan data dari sebuah file eksternal ke ADT-ADT yang terdapat di dalam program untuk mengembalikan data hasil simpanan eksekusi jalan sebelumnya.

3.20 B02 – Denah Dinamis

Fitur ini mengubah ukuran denah rumah sakit sesuai dengan keinginan manager. Fitur ini mengimplementasikan ADT Matrix. Selain mengubah ukuran denah, fitur ini juga bisa memindahkan dokter dari suatu ruangan ke ruangan lainnya.

3.21 B03 – Aura! & B04 – Banarich!!!

Fitur ini menambahkan dua atribut baru pada pengguna. Aura untuk dokter dan Banarich untuk dokter dan pasien. Aura merupakan atribut keberhasilan seorang dokter dalam menyembuhkan pasiennya. Sedangkan, banarich merupakan sistem keuangan pada rumah sakit ini yang dimiliki oleh dokter dan pasien. Rumah sakit juga menggunakan Banarich untuk biaya operasionalnya, sehingga rumah sakit juga memiliki banarich. Kedua fitur ini menggunakan implementasi dari ADT List yang diasosiasikan oleh masing-masing pengguna.

3.22 B06 – Mainin Antrian

Fitur ini memungkinkan pengguna yang memiliki peran sebagai pasien untuk maju ke posisi antrian pertama ataupun membatalkan antriannya. Fitur ini menggunakan implementasi ADT Queue.

4. Pembagian Kerja Anggota Kelompok

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F00 - ADT	ADT list, linked list, matrix, stack, queue, node, set, map, role	13524040	13524040	13524040
F01 - Login	Membuat file user.csv, parsing file user.csv, fungsi login	13524074	13524074	13524074
F02 - Register Pasien	Prosedur AddUser, fungsi IdxUser, fungsi utama register	13524040	13524040	13524040
F03 - Logout	Seluruh implementasi terkait	13524040	13524040	13524040
F04 - Lupa Password	Fungsi RLE, fungsi utama passwordUpdate	13524042	13524042	13524042
F05 - Menu & Help	Prosedur help	13524074	13524074	13524074
F06 (D01) - Denah Rumah Sakit	Fungsi print_denah, print_ruangan,	13524118 13524040	13524118 13524040	13524118 13524040
F07 – Lihat User	Fungsi ToLower, ToUpper, SortList;	13524044	13524044	13524044

	Prosedur PilihanLihat, MenuLihat, LihatUser, LihatPasien, LihatDokter			
F08 – Cari User	Prosedur BinarySearchUser, PilihanCari, MenuCari, CariUser, CariPasien, CariDokter	13524044	13524044	13524044
F09 (D02) – Lihat Antrian	Seluruh implementasi terkait	13524040	13524040	13524040
F10 - Tambah Dokter	Seluruh implementasi terkait	13524044	13524044	13524044
F11 - Diagnosis	Implementasi adt dan fungsi lain untuk prosedur diagnosis	13524074	13524074	13524040
F12 - Ngobatin	Implementasi adt dan fungsi lain untuk prosedur ngobatin	13524074	13524074	13524040
F13 - Aku boleh pulang ga, dok?	Seluruh implementasi terkait	13524118	13524118	13524040
F14 – Daftar Checkup	Seluruh implementasi terkait	13524040	13524040	13524040
F15 – Antrian Saya!	Seluruh implementasi terkait	13524040	13524040	13524040

F16 – Minum Obat	Fungsi minumObat, implementasi perut	13524042	13524042	13524040
F17 – Minum Penawar	Fungsi minumPenawar, implementasi perut	13524042	13524042	13524040
F18 - Exit	Seluruh implementasi terkait	13524044	13524044	13524044
D03 - Load	Seluruh implementasi terkait	13524040	13524040	13524040 13524044
D04 - Save	Seluruh implementasi terkait	13524040	13524040	13524040 13524044
B02 – Denah Dinamis	Fungsi UbahRuangan, PindahDokter	13524040	13524040	13524040
B03 – Aura	Seluruh implementasi terkait	13524040	13524040	13524040
B04 - Banarich	Seluruh implementasi terkait	13524040	13524040	13524040
B06 - Mainin Antrian	Seluruh implementasi terkait	13524040	13524040	13524040
Laporan	Milestone 1 & MoM	13524042		
	Milestone 2	Semua anggota		

Tabel 2 - Pembagian Kerja Kelompok

5. Checklist Fitur dan Implementasi

Fitur	Desain	Implementasi	Testing
F01 - Login	√	√	√
F02 - Register Pasien	√	√	√
F03 - Logout	√	√	√
F04 - Lupa Password	√	√	√
F05 - Menu & Help	√	√	√
F06 (D01) - Denah Rumah Sakit	√	√	√

F07 – Lihat User	√	√	√
F08 – Cari User	√	√	√
F09 (D02) – Lihat Antrian	√	√	√
F10 - Tambah Dokter	√	√	√
F11 – Diagnosis	√	√	√
F12 – Ngobatin	√	√	√
F13 – Aku boleh pulang ga, dok?	√	√	√
F14 – Daftar Checkup	√	√	√
F15 – Antrian Saya!	√	√	√
F16 – Minum Obat	√	√	√
F17 – Minum Penawar	√	√	√
F18 - Exit	√	√	√
D03 – Load	√	√	√
D04 - Save	√	√	√
B01 – Git Best Practice	√	√	√
B02 – Denah Dinamis	√	√	√
B03 - Aura	√	√	√
B04 - Banarich,	√	√	√
B05 - Dead or Alive?!	√	√	√
B06 - Mainin Antrian	√	√	√

Tabel 3 - Checklist Fitur dan Implementasi

Keterangan:

- √ : Sudah Selesai
- × : Belum Selesai
- - : Belum Dikerjakan.

6. Desain Command

Fitur	Command	Masukan	Keluaran
F01	LOGIN	Username, password	Melakukan sequential searching user.csv dan membandingkan username dan password dengan input

F02	REGISTER	Username, password	Menambahkan user baru pada user.csv
F03	LOGOUT	-	Kembali ke menu login bila user dalam kondisi loggedIn, dan set loggedIn menjadi 0.
F04	LUPA_PASSWORD	Username, kode unik, password baru	Mengupdate password baru pada username.csv
F05	- (Menu)	Masukan string perintah	Membandingkan masukan dengan strcmp() dan memanggil fungsi fitur jika role benar
	HELP	-	Mencetak perintah- perintah yang dapat digunakan oleh pengguna berdasarkan peran yang dimiliki
F06 (D01)	LIHAT_DENAH	-	Mencetak denah rumah sakit
	LIHAT_RUANGAN	-	Mencetak kapasitas pasien, dokter di ruangan, dan pasien-pasien di ruangan
F07	LIHAT_USER	Pilihan urutan, Pilihan sort	Memanggil PilihanLihat(Ulist, prompt) dalam keadaan user sudah login.
	LIHAT_PASIEN		
	LIHAT_DOKTER		
F08	CARI_USER	Pilihan pencarian, nomor id/nama user/nama	Memanggil PilihanCari(Ulist,
	CARI_PASIEN		

	CARI_DOKTER	penyakit (untuk pasien)	prompt dalam keadaan user sudah login.
F09 (D02)	LIHAT_SEMUA_ANTRIAN	-	Menampilkan seluruh antrian pasien di semua ruang dalam denah rumah sakit
F10	TAMBAH_DOKTER	username dan password dokter	Menambahkan usn dan pass dokter bar uke dokterlist
	ASSIGN_DOKTER	username dokter dan no. ruangan	Assign dokter ke ruangan yang dipilih
F11	DIAGNOSIS	-	Menampilkan penyakit yang diderita user dengan memanggil prosedur DIAGNOSIS()
F12	NGOBATIN	-	Menampilkan penyakit yang diderita user dan menuliskan urutan obat yang perlu diminum dengan memanggil prosedur NGOBATIN()
F13	PULANGDOK	-	Melakukan pengecekan terhadap diagnosis penyakit dan kesesuaian obat yang diminum (bila ada)
F14	DAFTAR_CHECKUP	Data medis dasar dan memilih dokter yang tersedia	Memanggil DaftarCheckup() Memasukkan user ke daftar antrian
F15	ANTRIAN	-	Memanggil CekAntrian(masterID)

			Menampilkan status antrian
F16	MINUM_OBAT	Pilihan obat yang diminum	Memanggil minumObat(User, Olist) Menampilkan daftar obat, menghapus obat yang diminum dari inventory, dan memasukkannya ke stack perut.
F17	MINUM_PENAWAR	-	Memanggil minumPenawar(User, Olist) Meng-pop stack teratas dan menambahkannya kembali ke inventory
F18	EXIT	validasi pilihan (y/n)	Memanggil exitProgram(&exit) Program selesai dan keluar
D03	LOAD	-\$./main <<nama_folder>>	Memuat data sesuai dengan isi pada file external saat memulai program
D04	SAVE	folderName	Meminta masukkan nama folder, validasi, dan menyimpan semua perubahan ke file external
B02	UBAH_DENAH	luas	Mengubah Rows dan Cols denah sesuai dengan luas
	PINDAH_DOKTER	nama dokter, ruangan sebelum, ruangan setelah	Memindahkan dokter dari ruangan sebelum ke ruangan setelah.
B06	SKIP_ANTRIAN	-	Maju ke paling depan dari antrian jika pasien sedang

			berada dalam antrian dan belum menjadi yang terdepan
	CANCEL_ANTRIAN	-	Keluar dari antrian jika pasien sedang berada dalam antrian
OTHER			Menuliskan ke layar, "Perintah tidak dikenali. Silakan coba lagi. Ketik HELP untuk bantuan tentang perintah yang ada."

Tabel 4 - Design Command

7. Kamus Data

KONSTANTA

```

MAX_FIELDS ← 20
MAX_FIELD_LENGTH ← 100
MAX_LINE_LENGTH ← 256
MAX_ROWS ← 100
MAX_COLS ← 100
MAX_PASIEN ← 20
CAPACITY ← 100
CONFIG_FILE ← "config.txt"
USER_FILE ← "user.csv"

extern Ulist : UserList
extern Olist : ObatList
extern OPlist : ObatPenyakitList
extern Plist : PenyakitList
extern masterID : integer
extern banarich: array[0..MAX_USER] of integer

type fields : array[0..MAX_FIELD_LENGTH-1] of char

type LoggedIn : enum < NotLoggedIn = 0

```

```

Manager      = 1,
Dokter       = 2,
Pasien       = 3 >

```

```

type User : <
    field          : array[0..15] of fields,
    inventoryObat  : LinkedList,
    perut          : Stack
>

type UserList : <
    contents : array[0..MAX_USER-1] of User,
    len      : integer
>

```

```

type Obat : <
    field : array[0..1] of fields
>

type ObatList : <
    contents : array[0..MAX_USER-1] of Obat,
    len      : integer
>

type ObatPenyakit : <
    field : array[0..2] of fields
>

type ObatPenyakitList : <
    contents : array[0..MAX_USER-1] of ObatPenyakit,
    len      : integer
>

```

```

type List : < nEff      : integer,
               contents: array [0..CAPACITY-1] of integer>

```

```

type LinkedList : < head: ^Node,
                    tail: ^Node,
                    size: integer >
type Node : < data: integer,
              next: ^Node >
type Queue : < front: ^Node,
              tail: ^Node,
              size: integer >
type Stack : < top: ^Node,
              size: integer >
type NodeMap: < key   : integer,
                prior : integer,
                data  : Queue,
                l, r   : ^NodeMap  >
type Map    = ^NodeMap
type pMap   = ^Map
type NodeSet : < key   : integer,
                prior : long,
                l, r   : ^NodeSet  >
type Set    = ^NodeSet
type pSet   = ^Set
type Matrix : <
    rows      : integer,
    cols      : integer,
    maxP      : integer,
    contents  : array[0..MAX_ROWS-1, 0..MAX_COLS-1,
0..MAX_PASIEN-1] of integer
>

```

```

type Manager : < id: integer >
type Dokter: < id: integer,
              ruangKerja: string[2],
              spesialisasi: string[99],

```

```

        antrian: ^Queue,
        queue_size: integer >
type Pasien : < id: integer,
                idDokter : integer >
type ManagerList : < contents : array[0..MAX_USER-1] of
Manager,
                    neff : integer >
type DokterList : < contents : array[0..MAX_USER-1] of Dokter,
                    neff : integer >
type PasienList : < contents : array[0..MAX_USER-1] of Pasien,
                    neff : integer >

```

```

type Denah: < M: Matrix,
               L: List,
               maxPasien: integer,
               maxPerRoom: integer >
type Ruangan : < idDokter : integer,
                  antrean : Queue >
type ListRuangan : < ruangan : array[0..MAX_USER-1] of Ruangan
>
RuangtoDokter: Map

```

8. Kamus Makro

User	
id(x)	(x).fields[0]
username(x)	(x).fields[1]
password(x)	(x).fields[2]
role(x)	(x).fields[3]
riwayat(x)	(x).fields[4]
suhu(x)	(x).fields[5]
sistol(x)	(x).fields[6]
diastol(x)	(x).fields[7]
detak(x)	(x).fields[8]

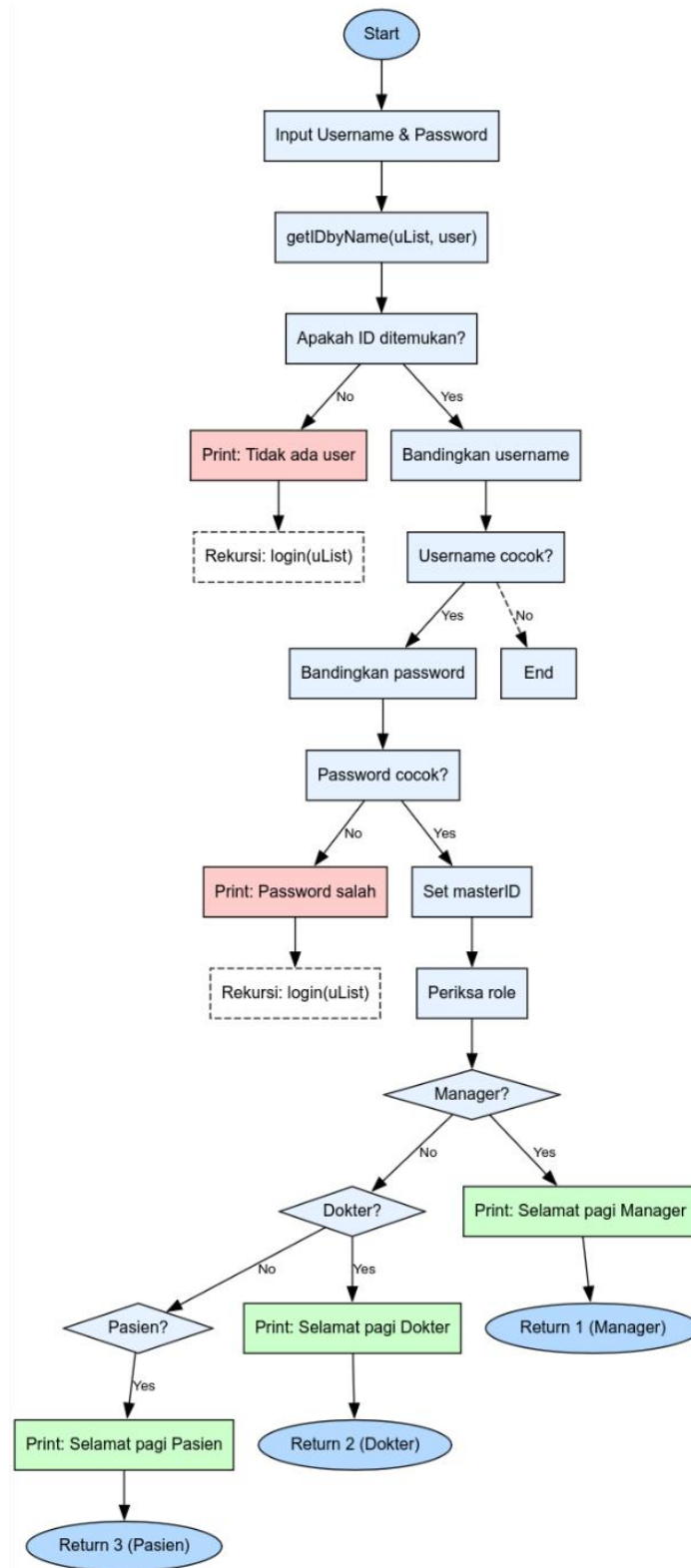
saturasi(x)	(x).fields[9]
gula(x)	(x).fields[10]
berat(x)	(x).fields[11]
tinggi(x)	(x).fields[12]
kolesterol(x)	(x).fields[13]
trombosit(x)	(x).fields[14]
USER(l, x)	(l).contents[x]
Penyakit	
id(x)	(x).field[0]
penyakit(x)	(x).field[1]
suhu_min(x)	(x).field[2]
suhu_max(x)	(x).field[3]
sistol_min(x)	(x).field[4]
sistol_max(x)	(x).field[5]
diastol_min(x)	(x).field[6]
diastol_max(x)	(x).field[7]
detak_min(x)	(x).field[8]
detak_max(x)	(x).field[9]
saturasi_min(x)	(x).field[10]
saturasi_max(x)	(x).field[11]
gula_min(x)	(x).field[12]
gula_max(x)	(x).field[13]
berat_min(x)	(x).field[14]
berat_max(x)	(x).field[15]
tinggi_min(x)	(x).field[16]
tinggi_max(x)	(x).field[17]
trombosit_min(x)	(x).field[18]
trombosit_max(x)	(x).field[19]
PENYAKIT(l, x)	(l).contents[x]
Obat	
id(x)	(x).fields[0]
nama_obat(x)	(x).fields[1]
OBAT(l, x)	(l).contents[x]

Obat Penyakit	
<code>id(x)</code>	<code>(x).fields[0]</code>
<code>penyakit_id(x)</code>	<code>(x).fields[1]</code>
<code>urutan(x)</code>	<code>(x).fields[2]</code>
<code>OBATPENYAKIT(l, x)</code>	<code>(l).contents[x]</code>
Role	
<code>DOKTER(i)</code>	<code>dokterList.contents[i]</code>
<code>PASIEN(i)</code>	<code>pasienList.contents[i]</code>
Hospital	
<code>RUANGAN(x)</code>	<code>ListRuangan.ruangan[x]</code>
Denah	
<code>MAT(d)</code>	<code>(d).M</code>
<code>LIST(d)</code>	<code>(d).l</code>
Matrix	
<code>ROWS(M)</code>	<code>(M).rows</code>
<code>COLS(M)</code>	<code>(M).cols</code>
<code>MAXP(M)</code>	<code>(M).maxP</code>
<code>ELMT(M, I, j, k)</code>	<code>(M).contents[i][j][k]</code>

Tabel 5 - Definisi Aksesori

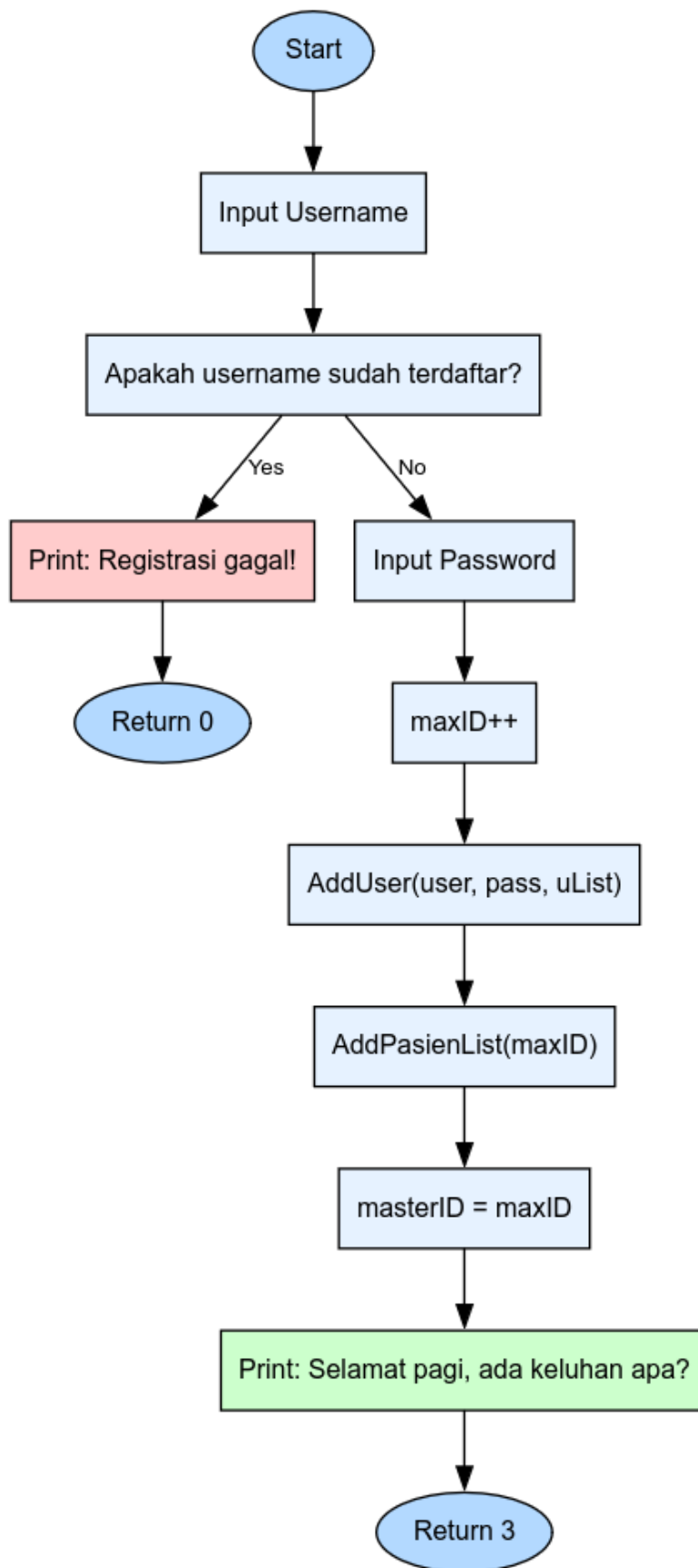
9. Dekomposisi Algoritmik dan Fungsional Program

9.1 F01 – Login



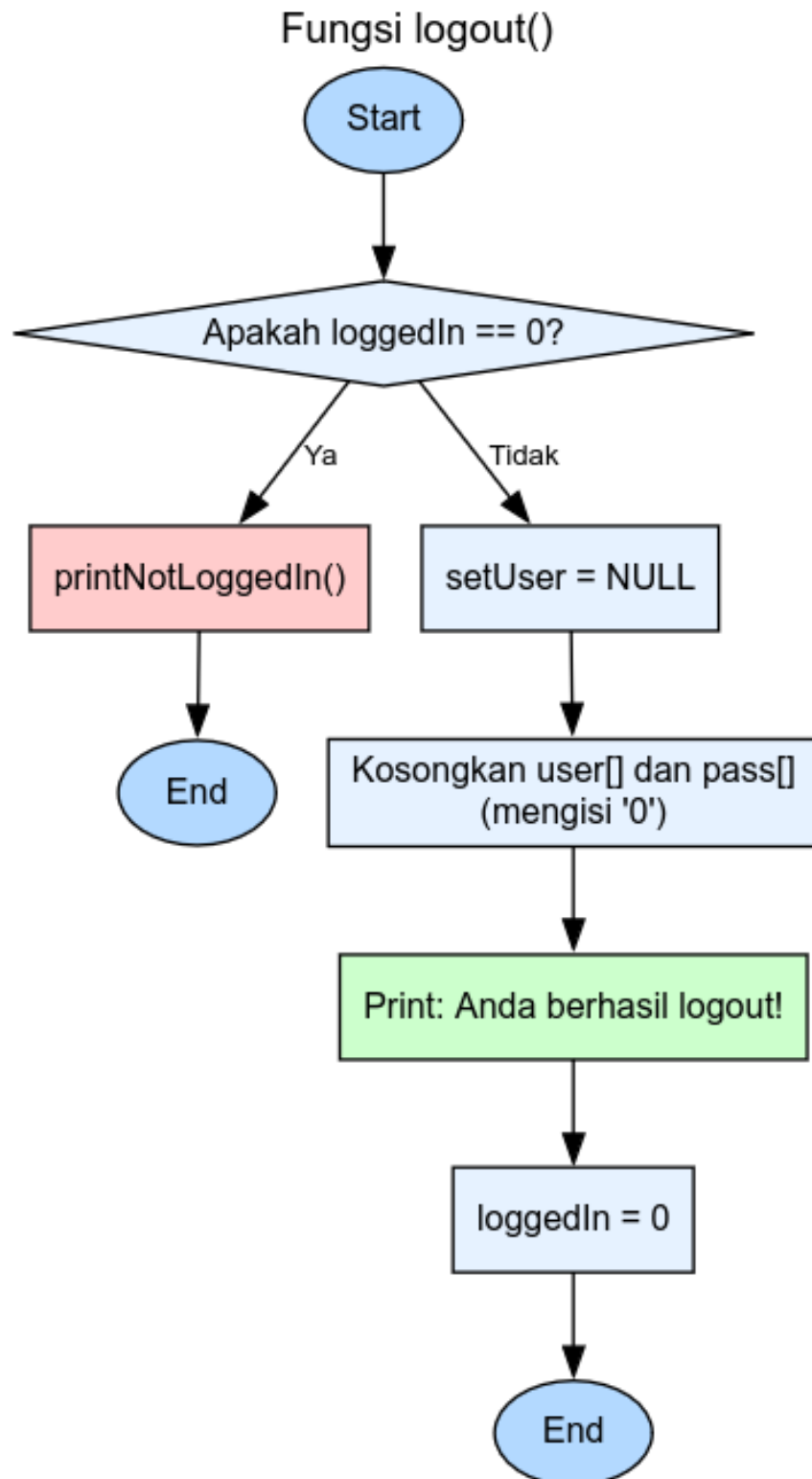
Gambar 1 - Flowchart Login

9.2 F02 – Register Pasien



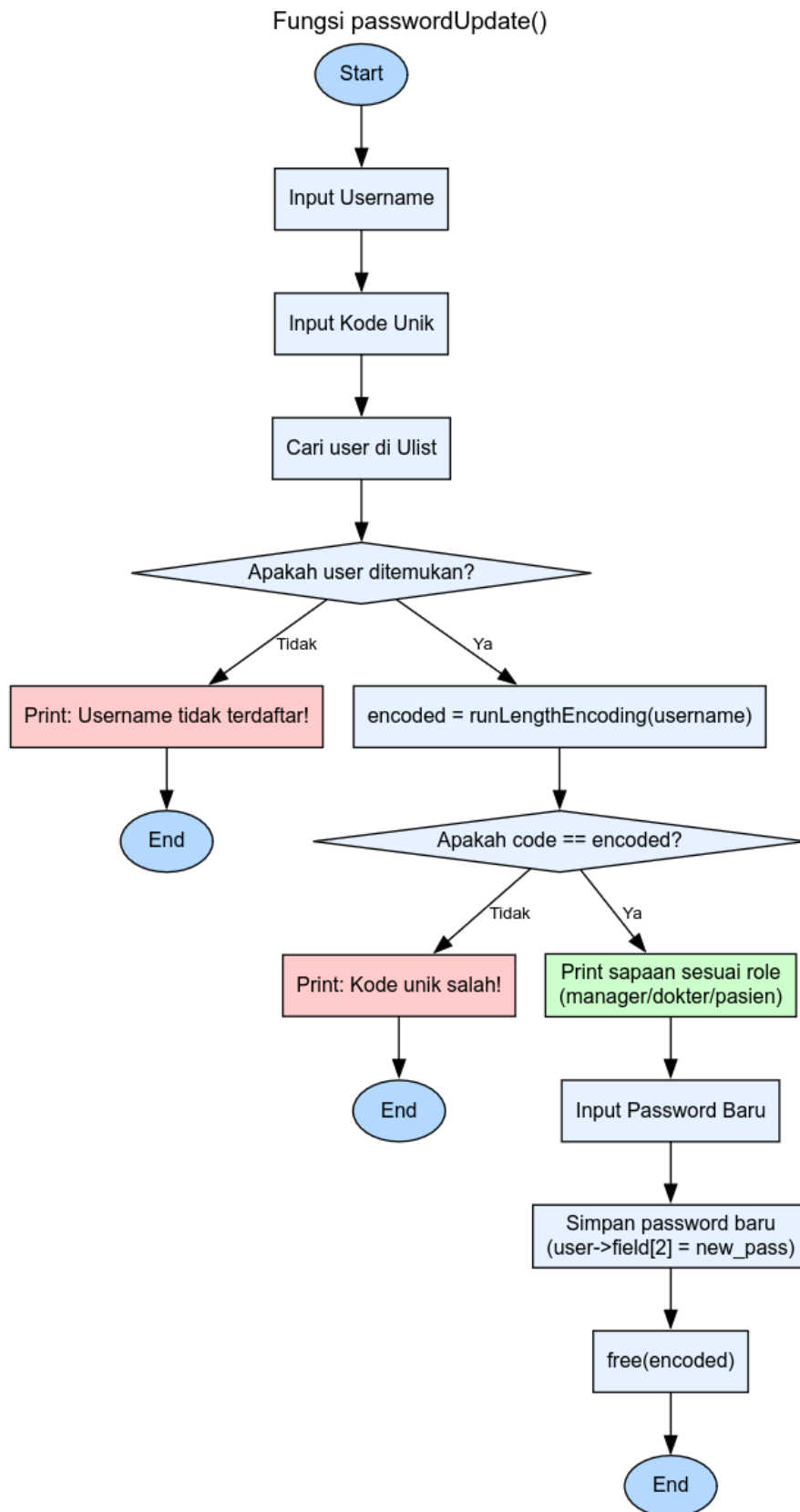
Gambar 2 - Flowchart Register

9.3 F03 – Logout



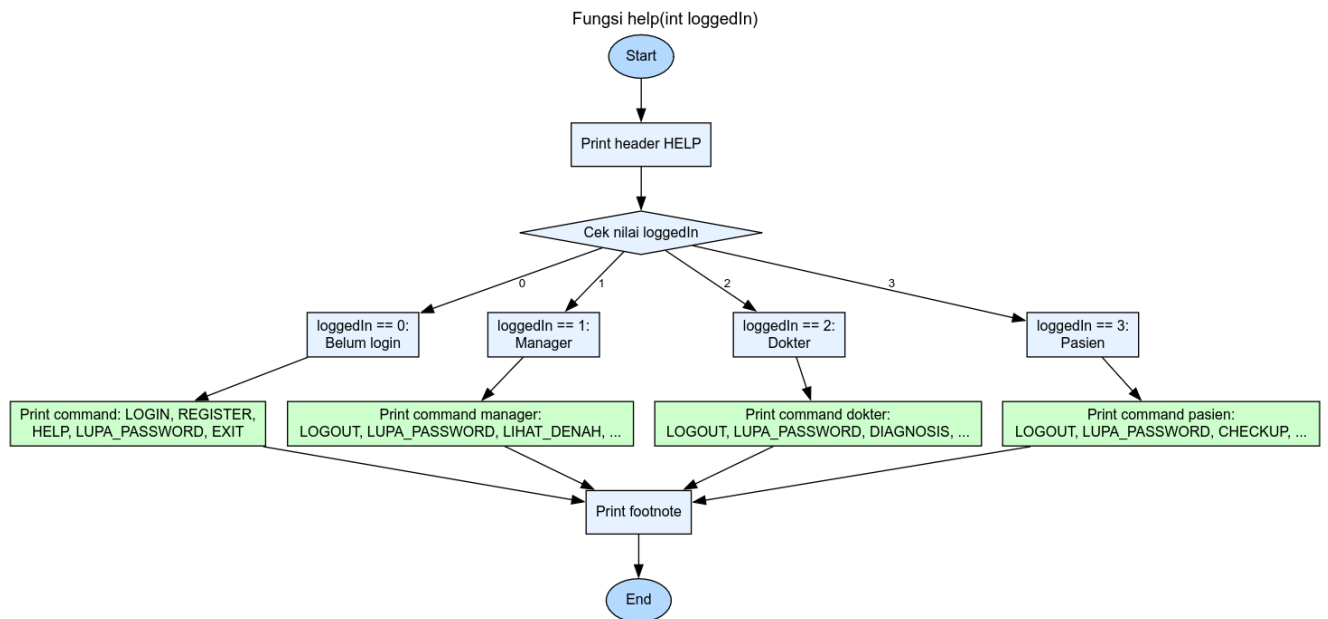
Gambar 3 - Flowchart Logout

9.4 F04 – Lupa Password



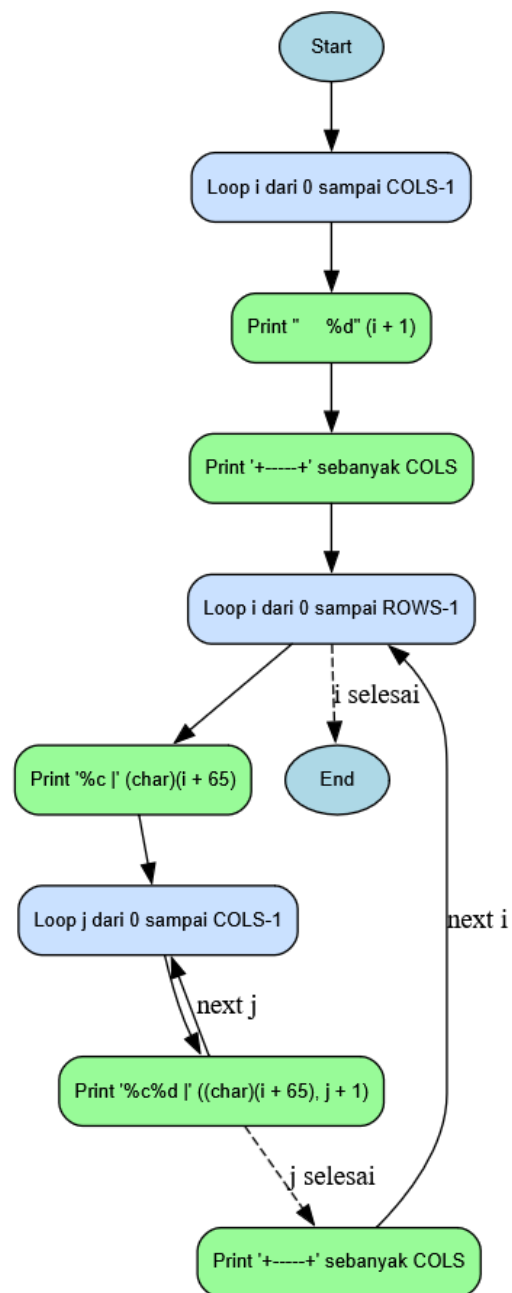
Gambar 4 - Flowchart PasswordUpdate

9.5 F05 – Menu & Help



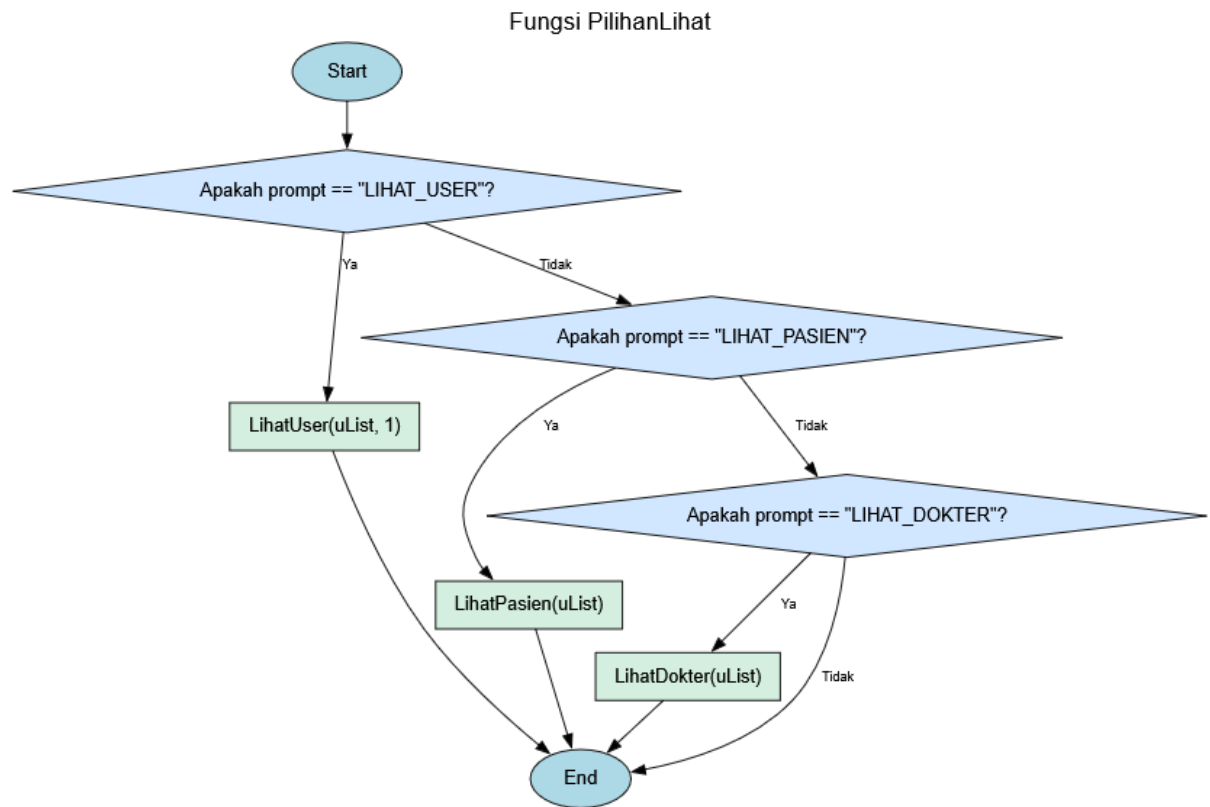
Gambar 5 - Flowchart Help

9.6 F06 (D01) – Denah Rumah Sakit



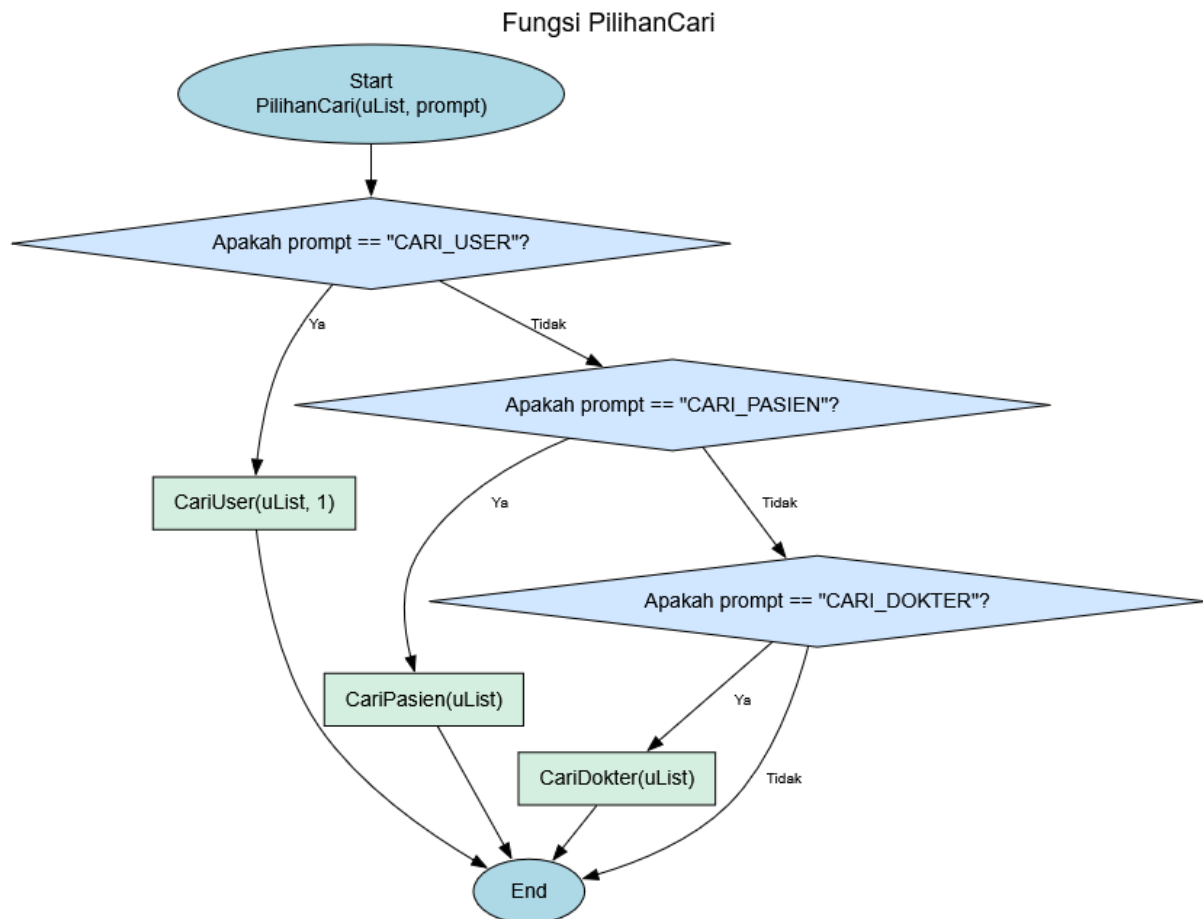
Gambar 6 - Flowchart Denah

9.7 F07 – Lihat User

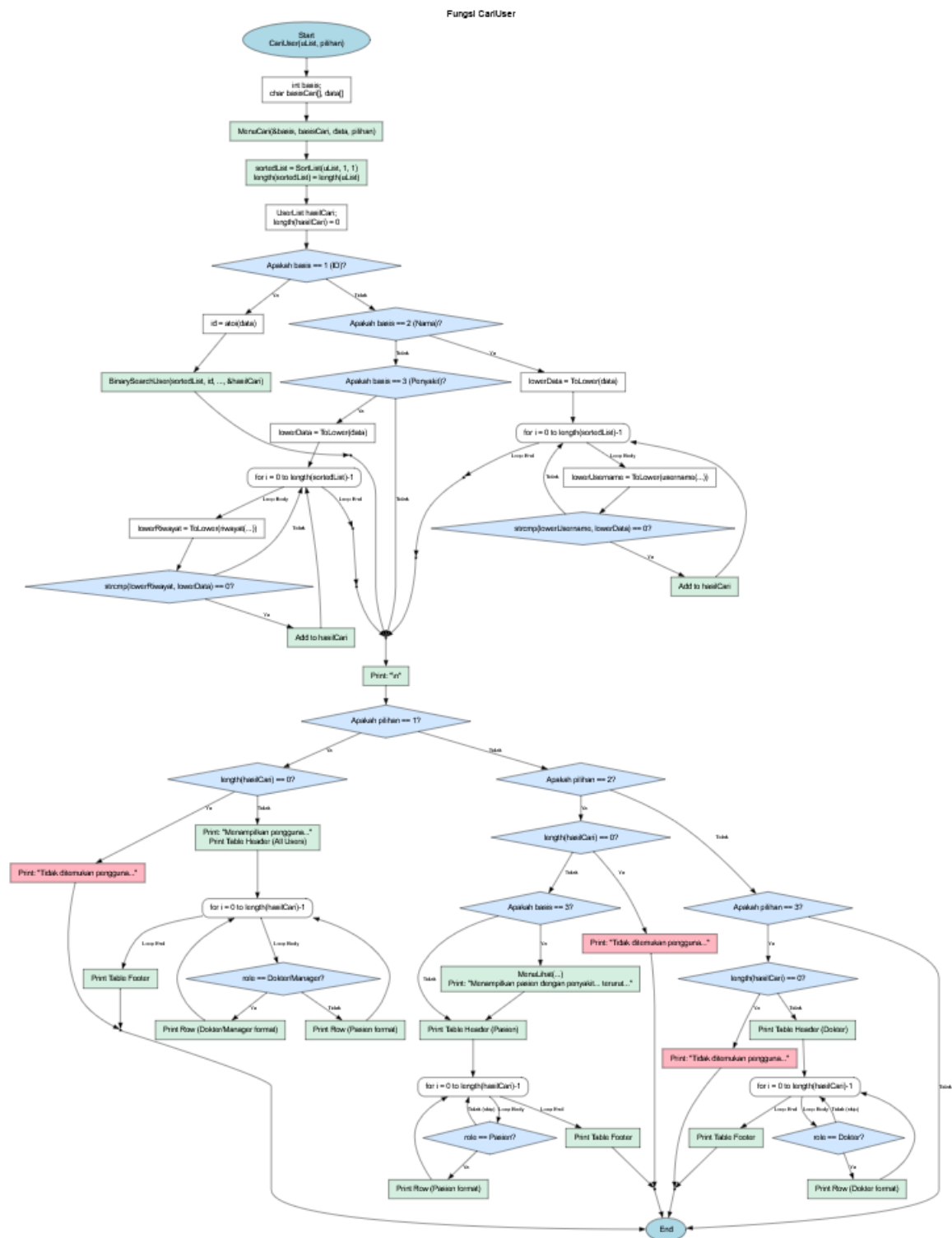


Gambar 7 - Flowchart PilihanLihat

9.8 F08 – Cari User

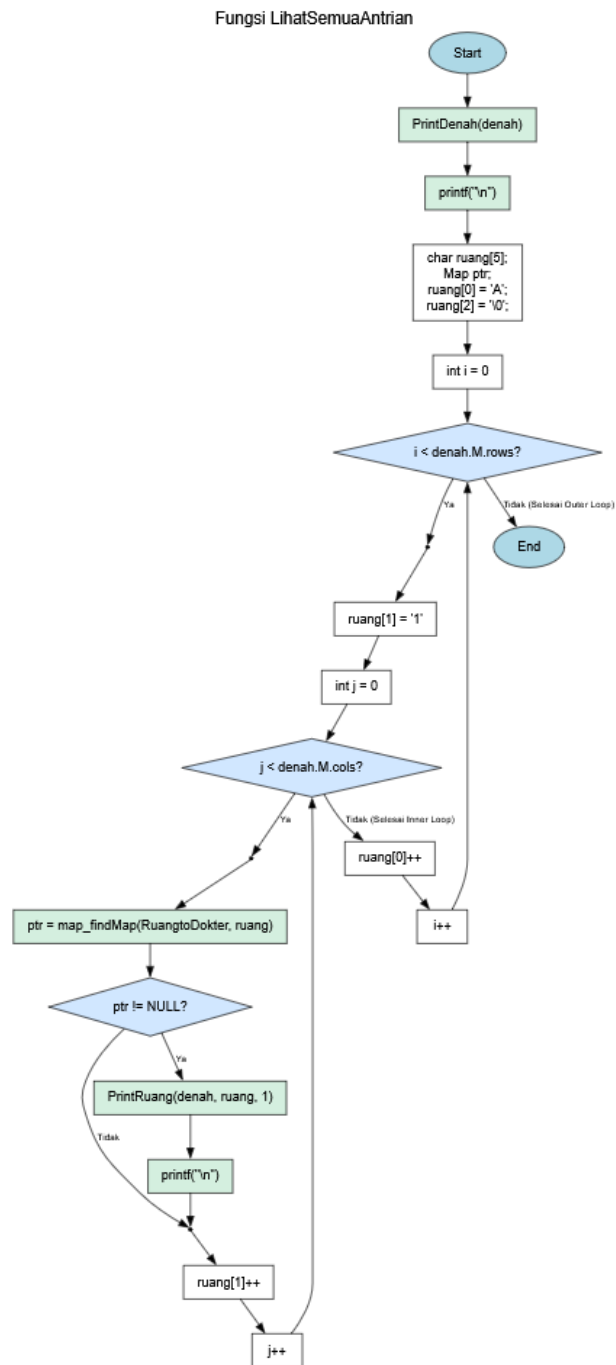


Gambar 9 - Flowchart PilihanCari



Gambar 10 - Flowchart CariUser

9.9 F09 (D02) – Lihat Antrian



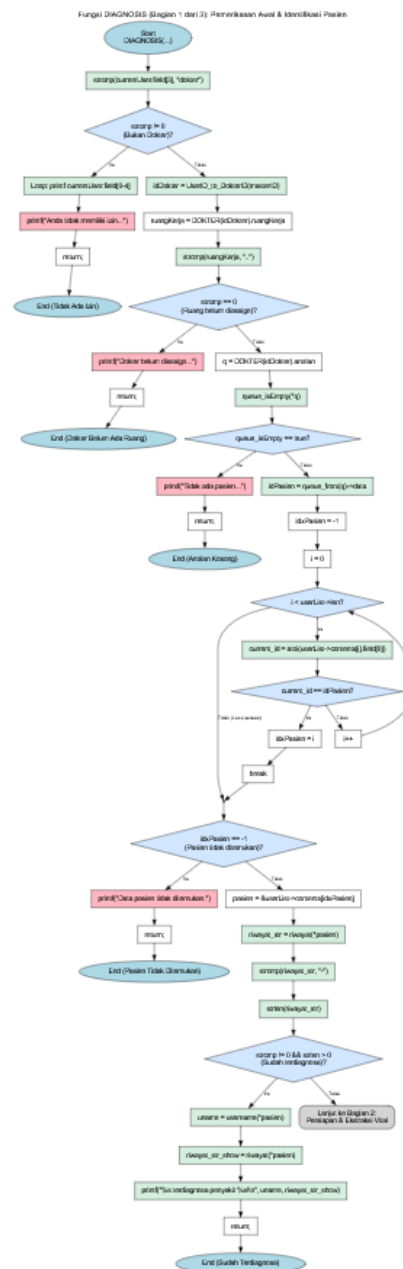
Gambar 11 - Flowchart LihatAntrian

9.10 F10 – Tambah Dokter



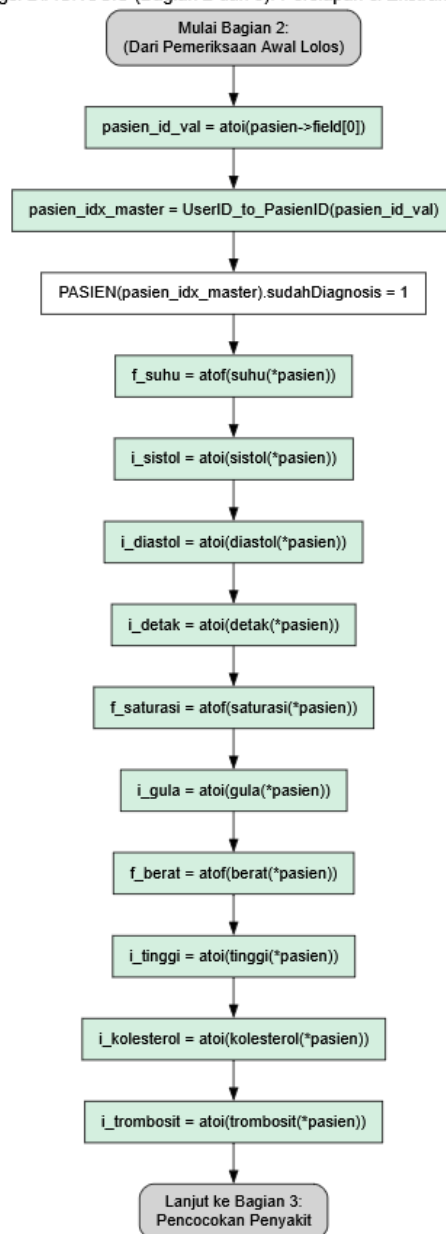
Gambar 12 - Flowchart TambahDokter

Sumber Diagram



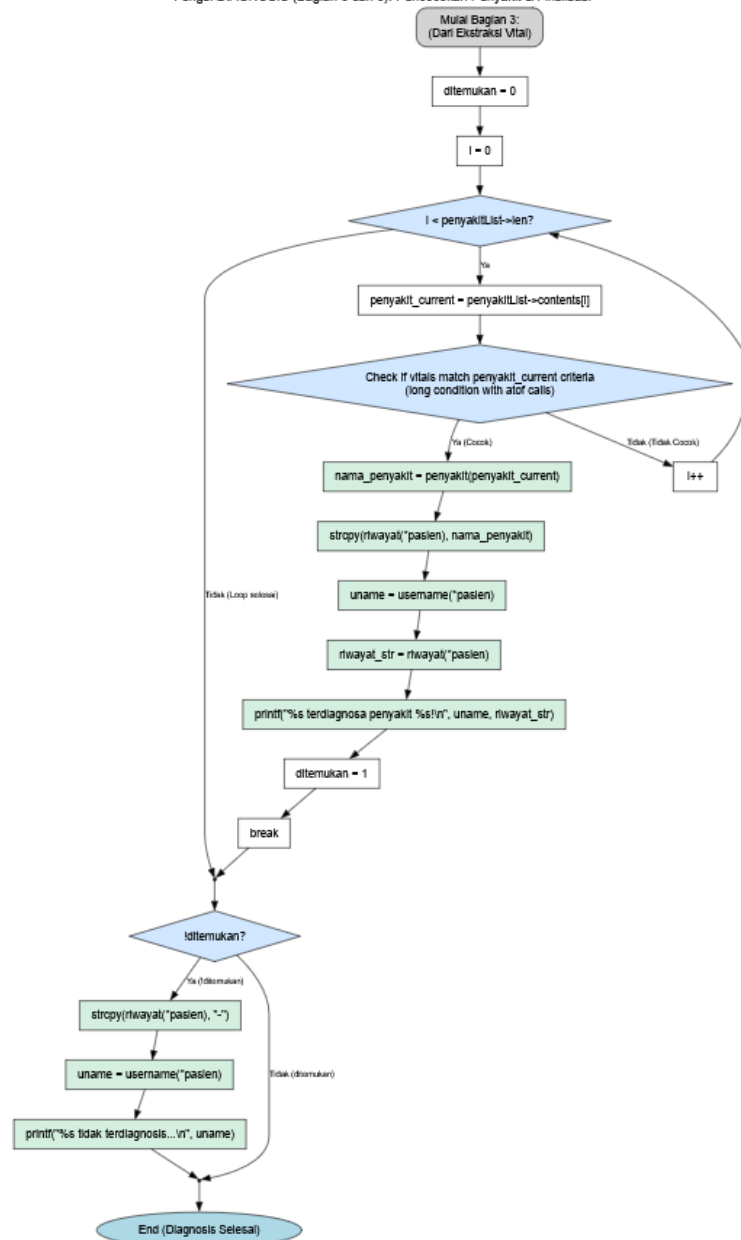
Gambar 13 - Flowchart Diagnosis 1

Fungsi DIAGNOSIS (Bagian 2 dari 3): Persiapan & Ekstraksi Vital



Gambar 14 - Flowchart Diagnosis 2

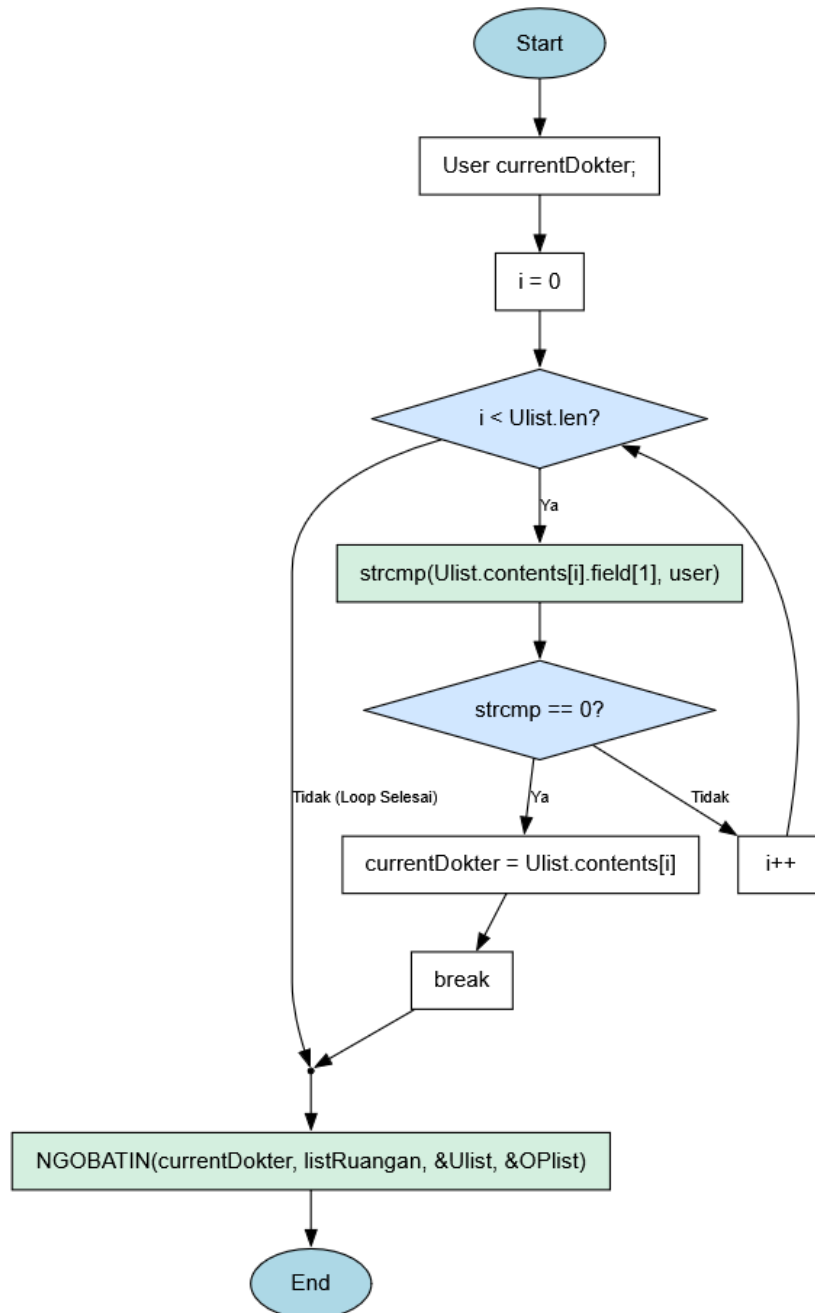
Fungsi DIAGNOSIS (Bagian 3 dari 3): Pencocokan Penyakit & Finalisasi



Gambar 15 - Flowchart Diagnosis 3

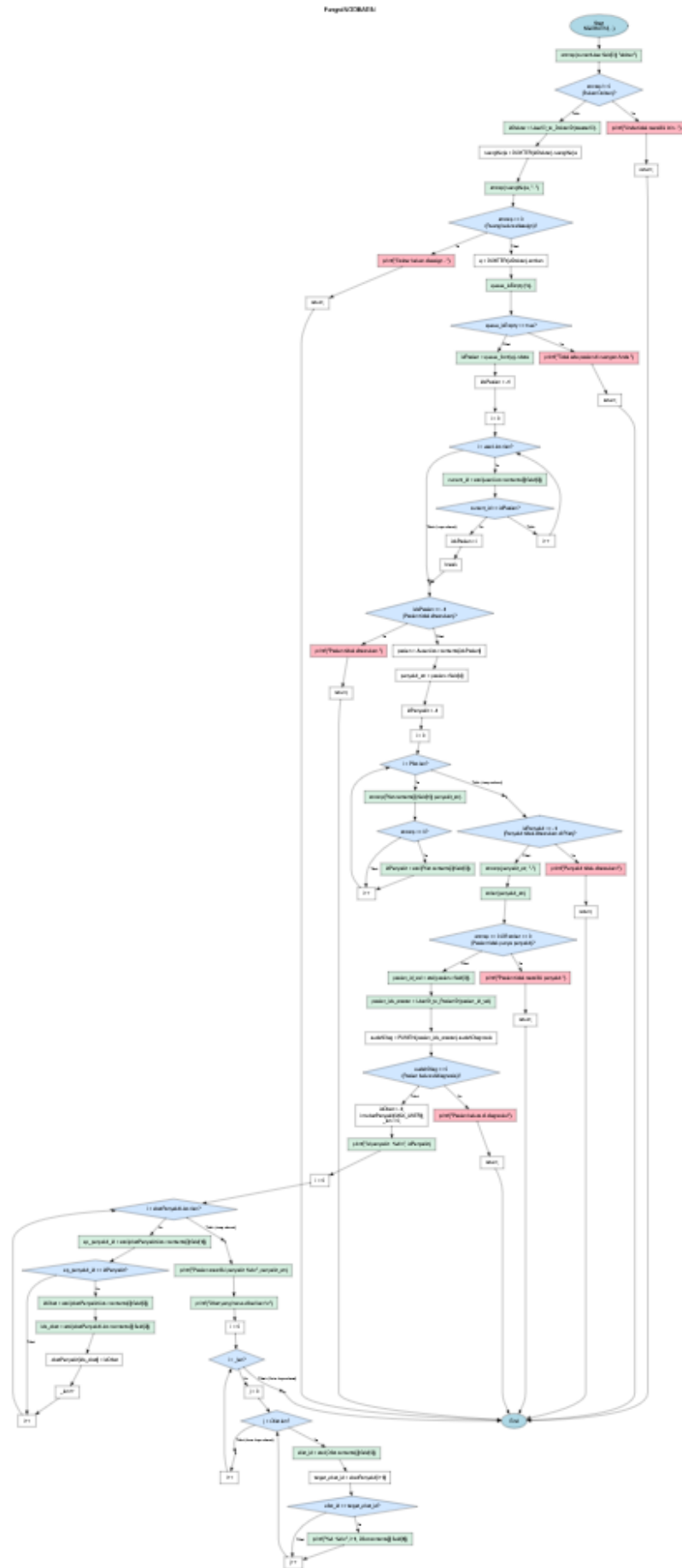
9.12 F12 – Ngobatin

Snippet: Cari Dokter & Panggil NGOBATIN



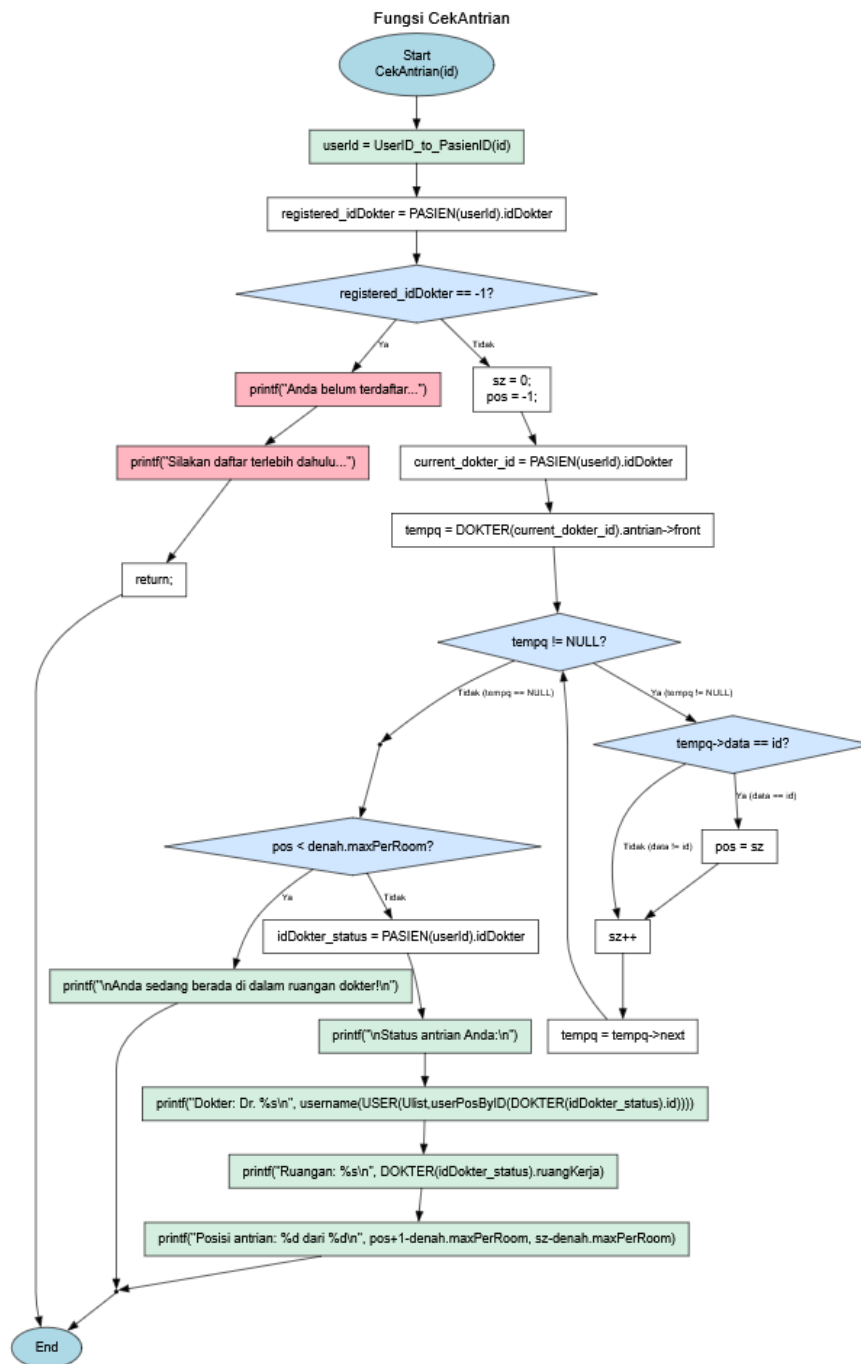
Gambar 16 - Flowchart CariDokter

Summer Diagram



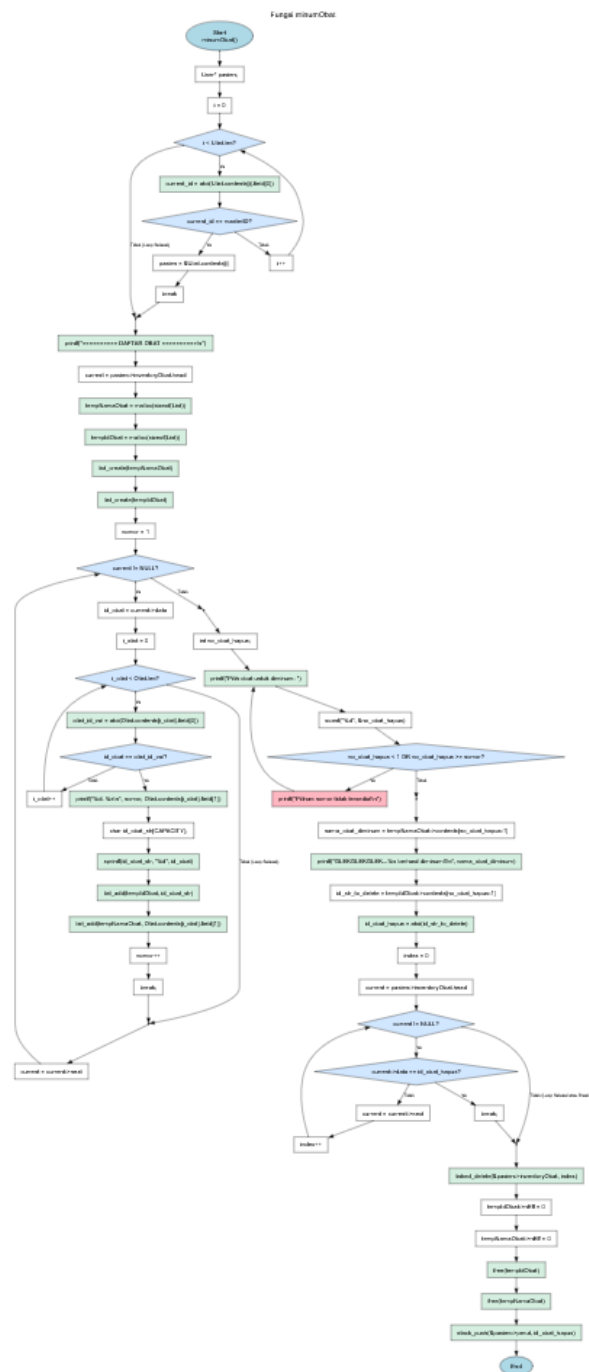
Gambar 17 - Flowchart Ngobatin

9.15 F15 – Antrian Saya!



Gambar 20 - Flowchart CekAntrian

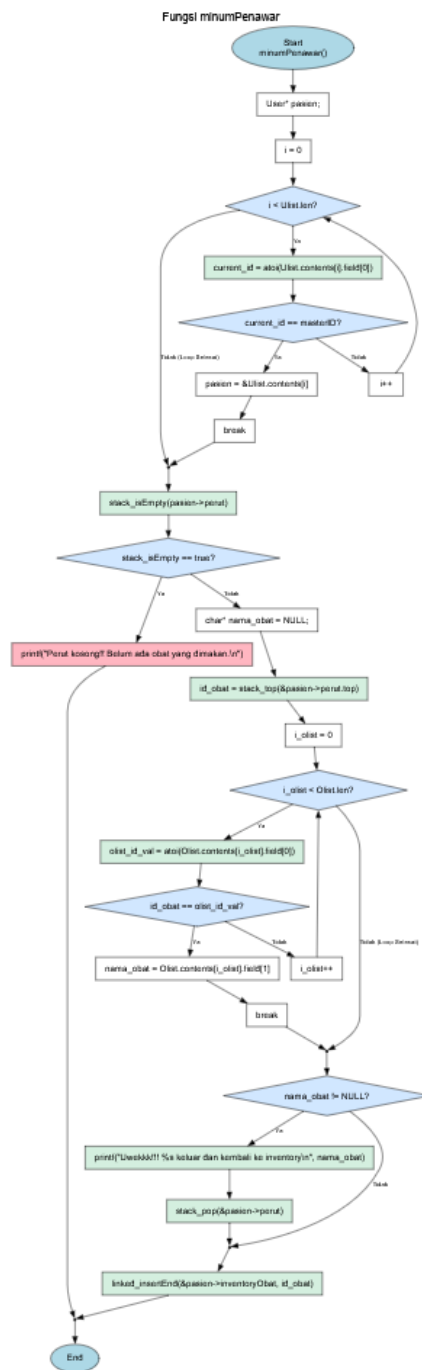
Sumber Diagram



Gambar 21 - Flowchart MinumObat

9.17 F17 – Minum Penawar

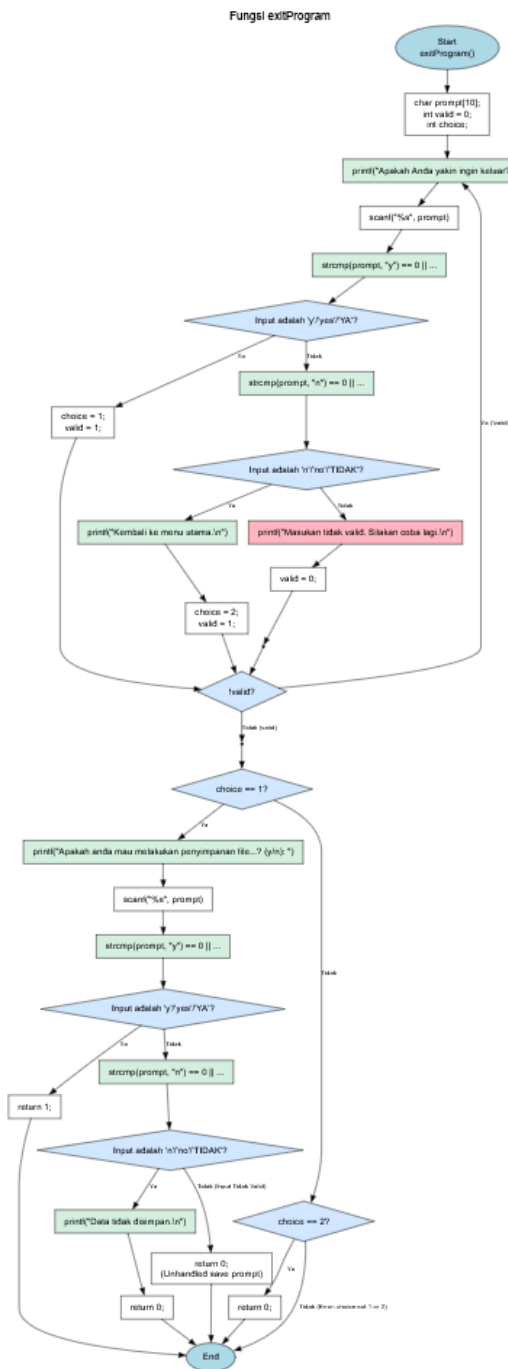
Sumber Diagram



Gambar 22 - Flowchart MinumPenawar

9.18 F18 – Exit & D04 – Save

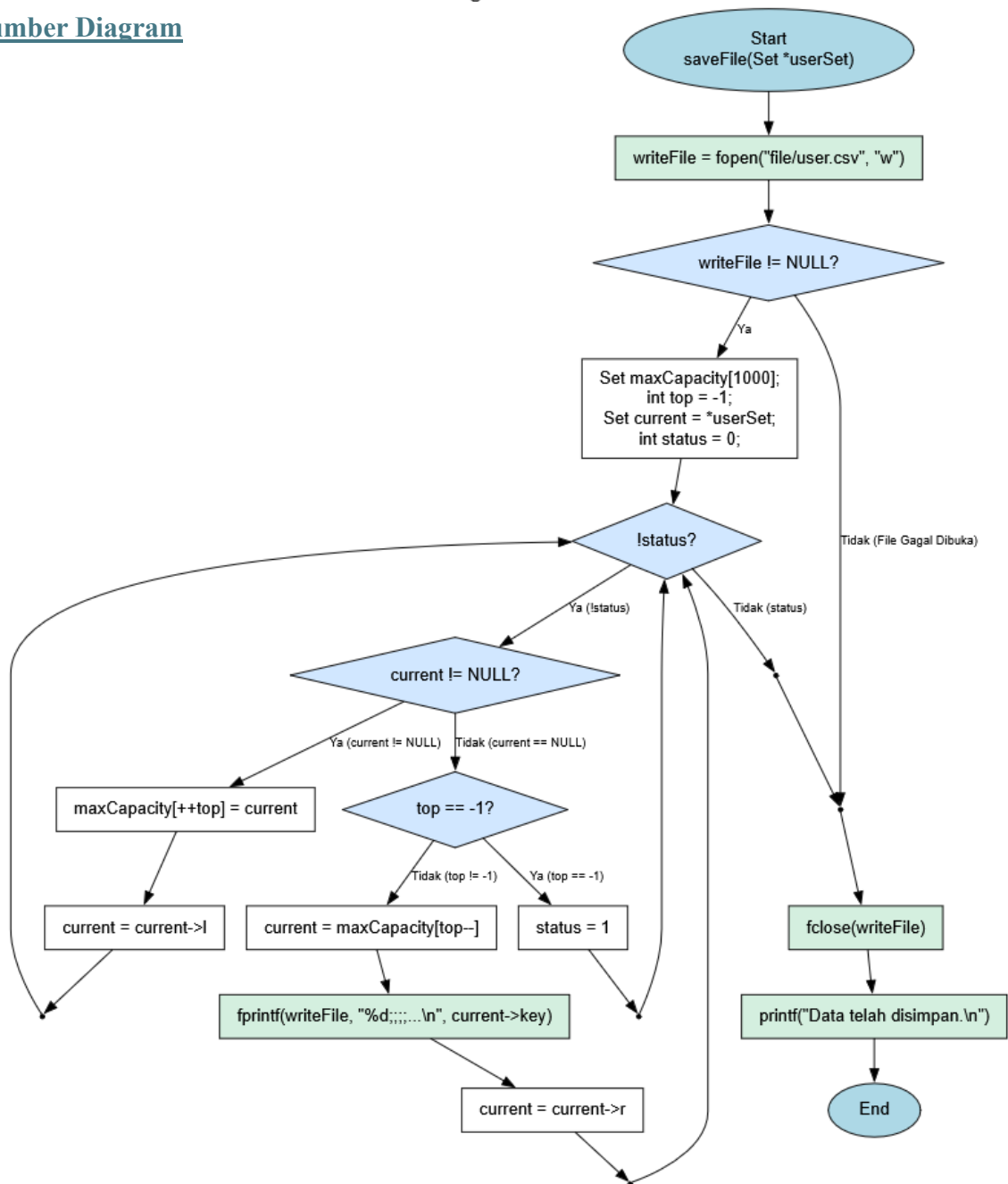
Sumber Diagram



Gambar 23 - Flowchart Exit

Fungsi saveFile

Sumber Diagram



Gambar 24 - Flowchart Save

9.19 D03 – Load

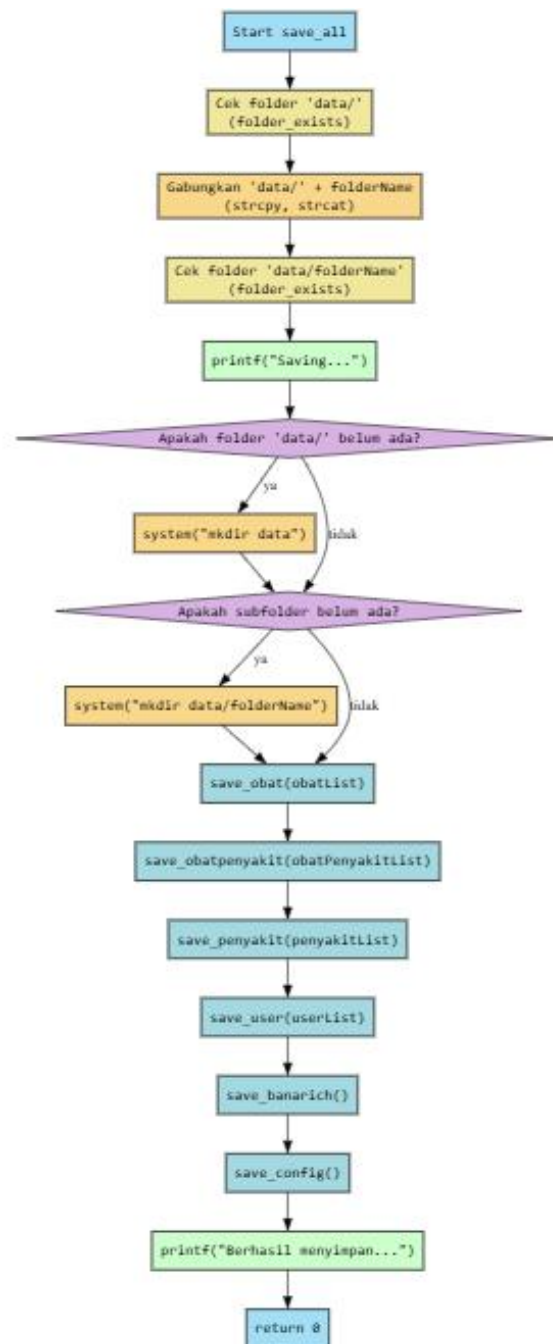
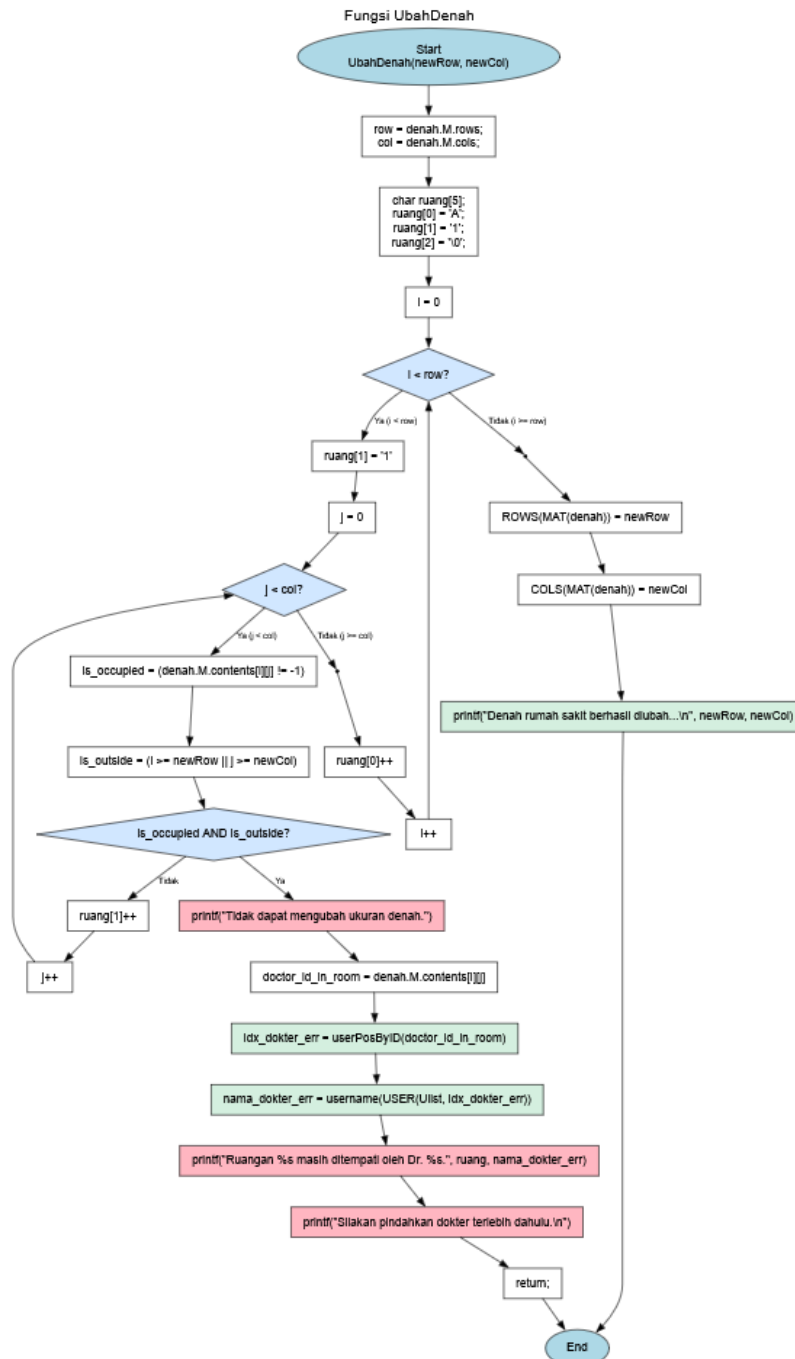


Diagram source

Gambar 25 - Flowchart Load

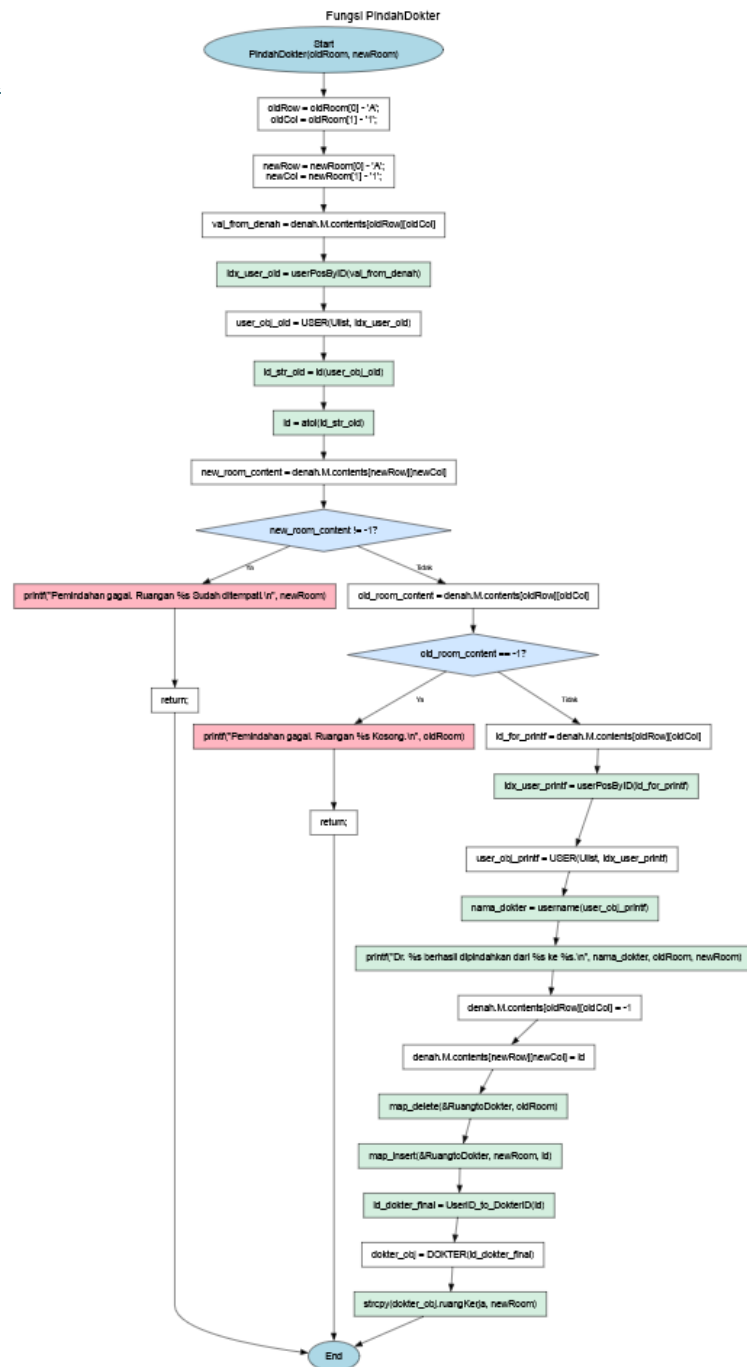
9.20 B02 – Denah Dinamis

Sumber Diagram

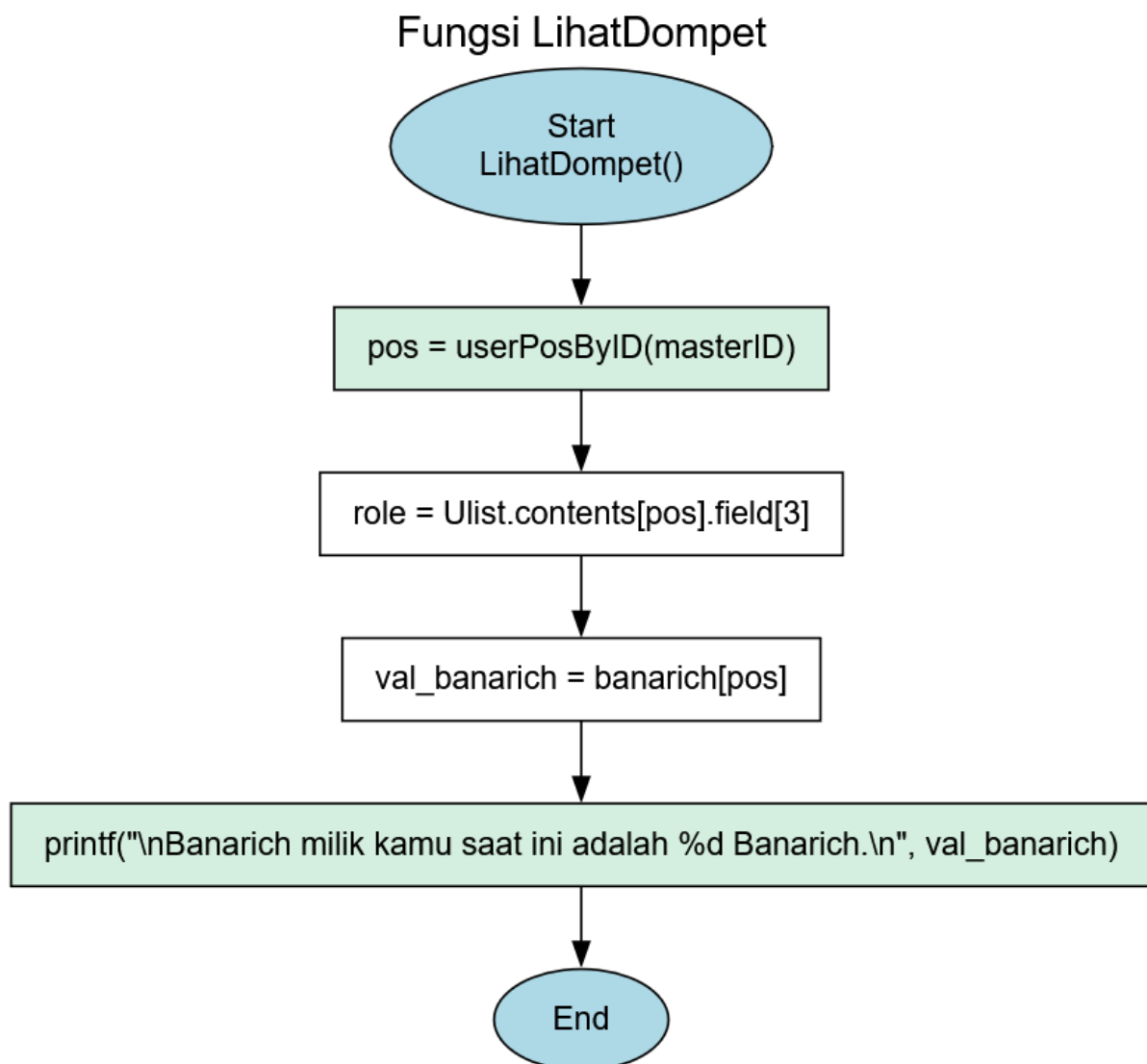


Gambar 26 - Flowchart UbahDenah

Sumber Diagram

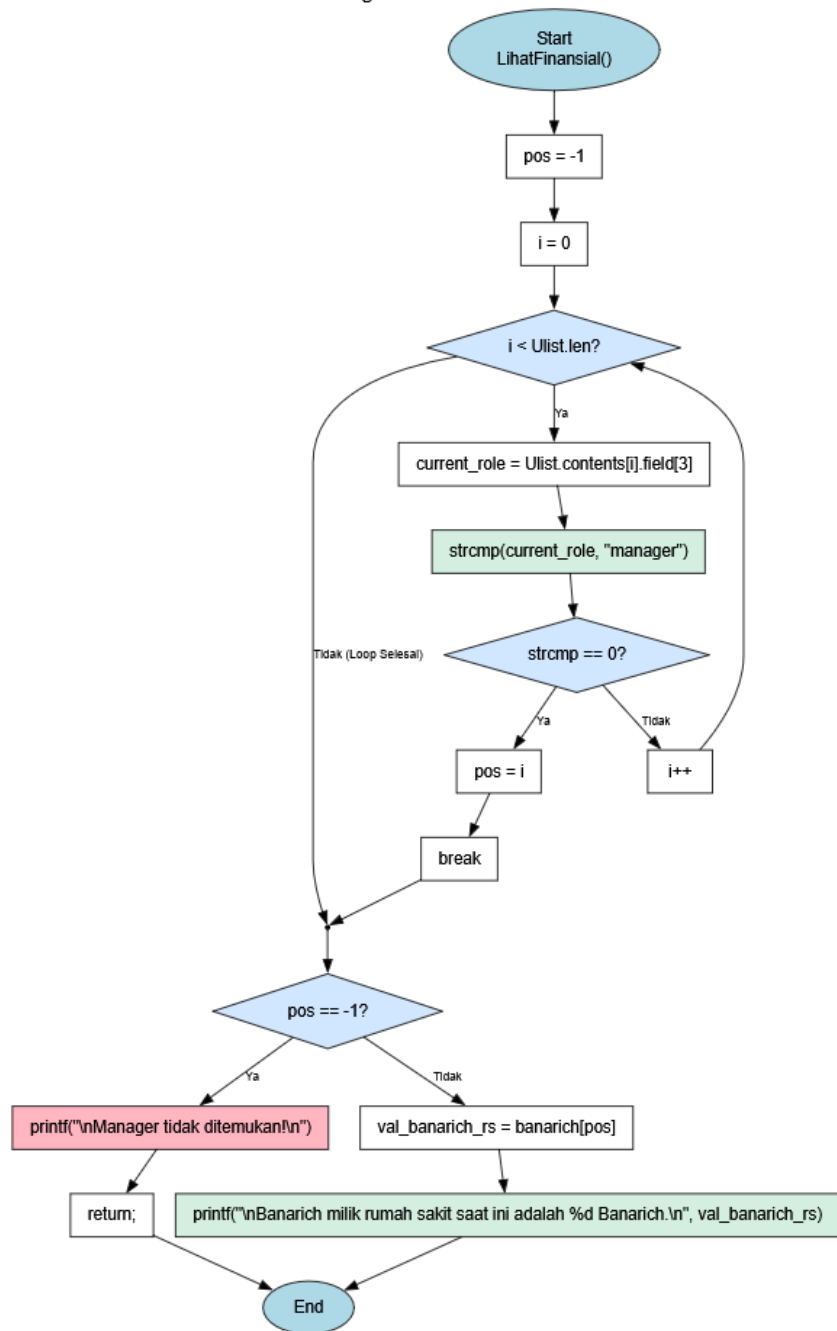


Gambar 27 - Flowchart PindahDokter



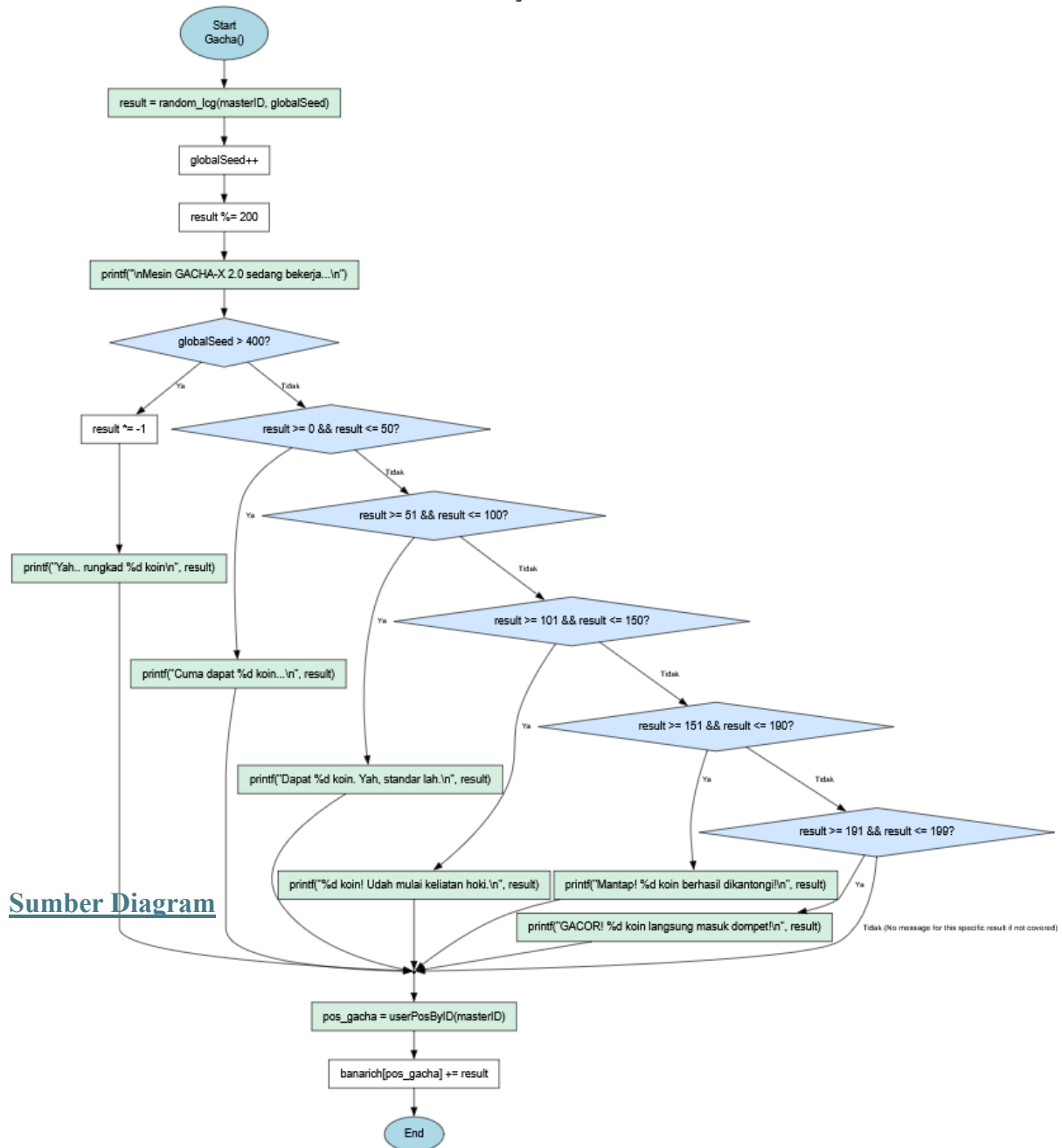
Gambar 28 - Flowchart LihatDomp

Fungsi LihatFinansial



Gambar 29 - Flowchart LihatFinansial

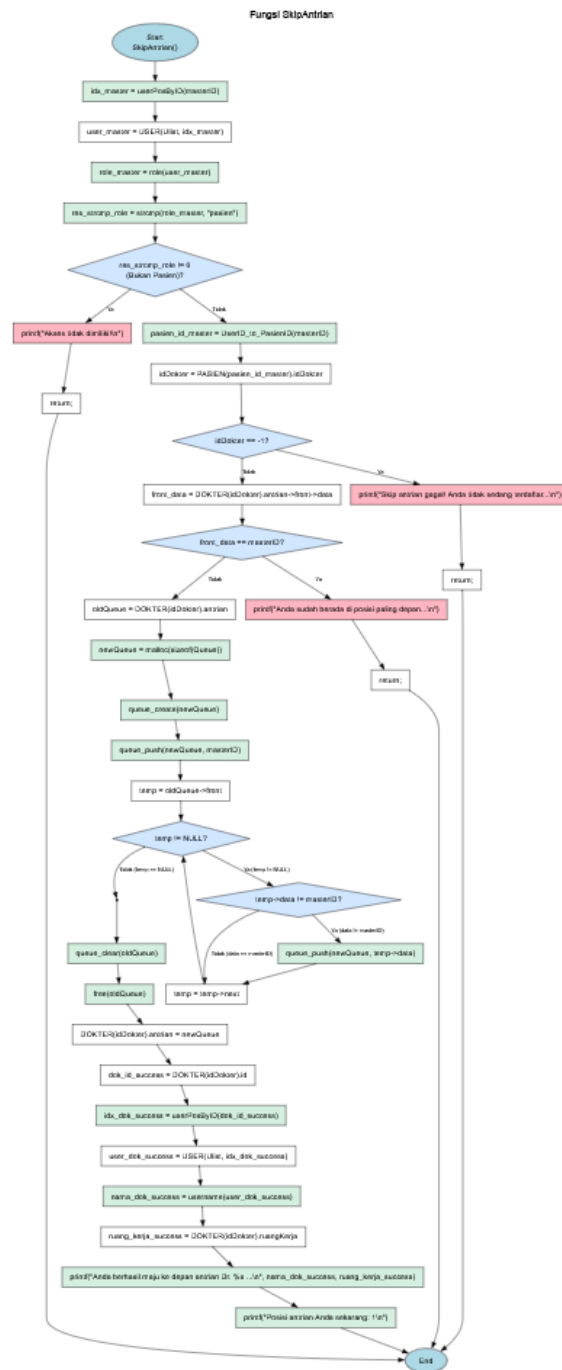
Fungsi Gacha



Sumber Diagram

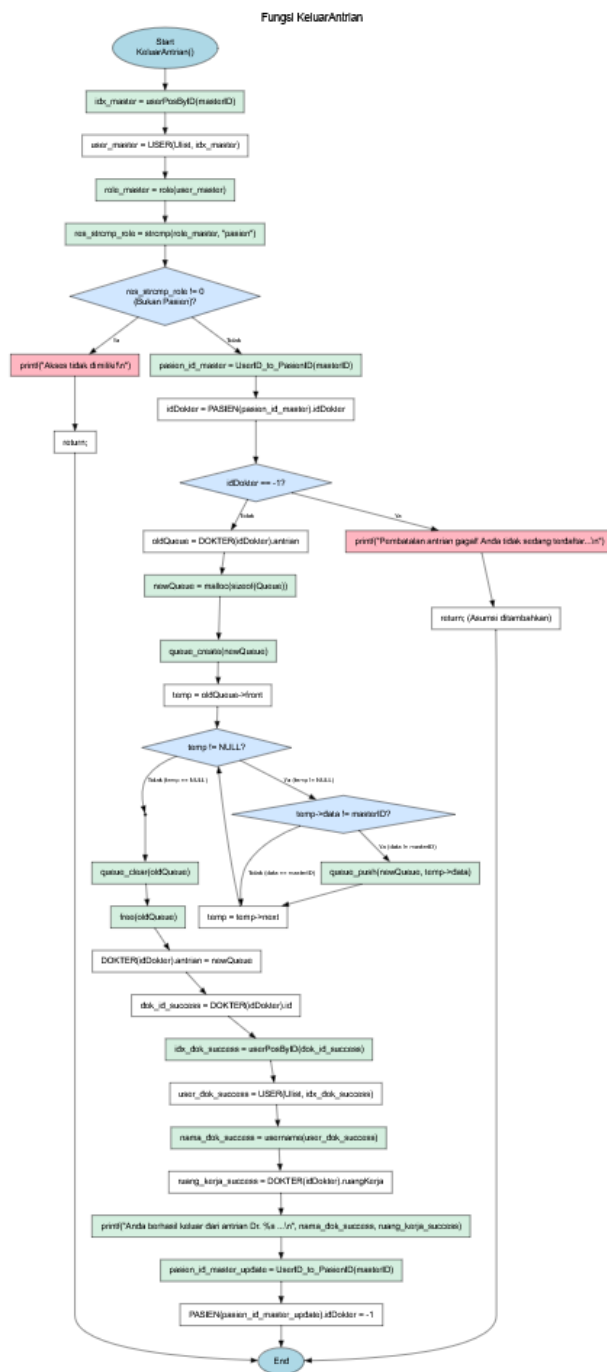
Gambar 30 - Flowchart Gacha

Sumber Diagram



Gambar 31 - Flowchart SkipAntrian

Sumber Diagram



Gambar 32 - Flowchart KeluarAntrian

10. Spesifikasi Modul/Prosedur/Fungsi

10.1 F01 – Login

function login (uList: UserList)

{ I.S. : UserList terdefinisi }

{ F.S. : Jika username dan password terdapat pada UserList, login berhasil }

KAMUS

validUser, validPass, check, masterID: integer

line, fields: string

getIDbyName(uList: UserList, name: string) -> integer

findID(uList: UserList, name: string) -> integer

ALGORITMA

validUser <- 0; validPass <- 0

output("Username: ")

input(user)

output("Password: ")

input(pass)

check <- getIDbyName(uList, user)

if (check = -1) then

output("Tidak ada Manager, Dokter, atau pun Pasien yang bernama" + user "!")

login(uList)

if (strcmp(username(USER(Ulist, check), user) = 0) then

masterID <- findID(uList, user)

if (strcmp(role(USER(Ulist, check), "manager" == 0)

then

-> 1

if (strcmp(role(USER(Ulist, check), "dokter" == 0)

then

-> 2

if (strcmp(role(USER(Ulist, check), "pasien" == 0)

then

-> 3


```

else
    output ("Password salah")
    login (uList)

```

10.2 F02 – Register

function Register(uList: Userlist) → integer
 {I.S. : UserList terdefinisi}
 {F.S. : Data user baru ditambahkan ke UserList}

KAMUS

ALGORITMA

```

    Output ("Username: ")
    Input (user)
    If (getIDbyName(*uList,user) != -1) then
        Output ("Registrasi gagal! Pasien dengan nama" +
user + "sudah terdaftar.")
        → 0
    Output ("Pass: ")
    Input (pass)
    MaxID ← MaxID + 1
    AddUser (user,pass, uList)
    AddPasienList (maxID)
    masterID → maxID
    Output ("Selamat pagi" + user + "! Ada keluhan
apa?\n")
    → 3;

```

10.3 F03 – Logout

procedure logout()
 { I.S. User mungkin sedang login(loggedIn = 1) atau tidak(loggedIn = 0).
 F.S. Jika sebelumnya login, maka user berhasil logout dan data login dikosongkan. Jika belum, maka ditampilkan pesan bahwa user belum login.}

KAMUS

i : integer

ALGORITMA

```
if loggedIn = 0 then  
    printNotLoggedIn()  
else  
    setUser ← NULL  
    i traversal[0...1000]  
        user[i] ← '\0'  
        pass[i] ← '\0'  
    output("Anda berhasil logout!")  
    loggedIn ← 0
```

10.4 F04 – Lupa Password

function passwordUpdate() → integer

{I.S. : File username.csv terdefinisi

F.S. : Password user yang sesuai diperbarui di file
user.csv}

KAMUS

username, new_pass, code: string
line, fields, encoded: char
user: User
i, validuser: integer

ALGORITMA

```
Output ("Username: ")  
Input (username)  
Output ("Kode Unik: ")  
Input (code)  
User = NULL  
i traversal[0.. Ulist.len]  
    if(strcmp(Ulist.contents[i].field[1],username)= 0)  
        user ← &Ulist.contents[i]  
        Break  
validuser = 0
```

```

    if (user != NULL) then
        encoded ← runLengthEncoding(user.field[1])
        if(strcmp(code, encoded)= 0)
            if (strcmp(user.field[3],"manager")= 0) then
                Output("Halo manager" + username + ",
silakan daftarkan ulang password anda!")
            else if (strcmp(user.field[3],"dokter") = 0)
then
                Output("Halo dokter" + username + ",
silakan daftarkan ulang password anda!")
            else if (strcmp(user.field[3],"pasien") = 0)
then
                Output ("Halo pasien + username + ",
silakan daftarkan ulang password anda!")
                Output ("Password Baru: ")
                Input (new_pass)
                strcpy(user->field[2],new_pass)
                free(encoded)
            Else
                Output("Kode unik salah!")
        Else
            Output("Username tidak terdaftar!")
    → 0

```

10.5 F05 – Menu & Help

```

procedure HELP(loggedIn: integer)
{ I.S. Nilai `loggedIn` diberikan sesuai status login saat
ini: 0 (belum login), 1 (manager), dll.
  F.S. Ditampilkan daftar perintah yang dapat digunakan
sesuai status login user saat ini.}

KAMUS

ALGORITMA
    output("\n")

```

```

output("=====
=====  HELP
=====\\n
")

    switch loggedIn do
        case 0:
            output("Kamu belum login sebagai role apapun.
Silahkan login terlebih dahulu.\\n\\n")
            output("    1. LOGIN
: Masuk ke dalam akun yang sudah terdaftar")
            output("    2. REGISTER
: Membuat akun baru")
            output("    3. HELP
: Menampilkan daftar perintah yang tersedia")
            output("    4. LUPA_PASSWORD
: Mengubah password akun")
            output("    5. EXIT
: Keluar dari aplikasi\\n")

        case 1:
            output("Halo Manager " + user + ". Kenapa kamu
memanggil command HELP? Kan kamu manager, tapi yasudahlah
kamu pasti sedang kebingungan. Berikut adalah hal-hal yang
dapat kamu lakukan sekarang:")
            u("    1. LOGOUT
: Keluar dari akun yang sedang digunakan")
            output("    2. LUPA_PASSWORD
: Mengubah password akun")
            output("    3. LIHAT_DENAH
: Menampilkan denah ruangan rumah sakit")
            (tambahkan perintah lain sesuai implementasi)

```

```

        (dst., case untuk role lainnya)
    end switch
    output("Footnote:")
    output("    1. Untuk menggunakan aplikasi, silahkan
masukkan perintah yang terdaftar (bukan memasukkan nomor
perintah)")
    output("    2. Jangan lupa untuk memasukkan input yang
valid\n\n")

```

10.6 F06 (D01) – Denah Rumah Sakit

procedure PrintDenah(input denah: Denah)

{I.S. denah sudah terdefinisi
F.S. denah rumah sakit tercetak}

KAMUS

i, j : integer

ALGORITMA

```

i traversal[0..COLS(MAT(denah))]
    Output "    " + (i + 1))
Output("    +")
i traversal[0..COLS(MAT(denah))]
    Output("-----+")
Output()
i traversal[0..ROWS(MAT(denah))]
    Output (" " + (char)(i + 65) + " |", )
    j traversal[0..COLS(MAT(denah))]
        Output(" " + (char)(i + 65), j + 1)+ " |")
    Output()
    Output("    +")
    j traversal[0..COLS(MAT(denah))]
        Output ("-----+")
        Output()

```

10.7 F07 – Lihat User

procedure PilihanLihat(input uList: UserList, input prompt: string)

```
{ I.S. prompt valid, uList terdefinisi }  
{ F.S. Mengarahkan ke prosedur lihat (LihatUser,  
LihatDokter, atau LihatPasien) }
```

KAMUS

pasienList, dokterList: UserList

ALGORITMA

```
if (strcmp(prompt, "LIHAT_USER") = 0) then  
    LihatUser(uList, 1);  
if (strcmp(prompt, "LIHAT_PASIEN") = 0) then  
    CreateList(pasienList)  
    CopyList(uList, pasienList)  
    LihatUser(pasienList, 2)  
if (strcmp(prompt, "LIHAT_DOKTER") = 0) then  
    CreateList(dokterList)  
    CopyList(uList, dokterList)  
    LihatUser(dokterList, 3)
```

procedure LihatUser(input uList: UserList, input pilihan: int)

```
{ I.S. uList terdefinisi dan valid, pilihan valid }  
{ F.S. Menampilkan list pengguna berdasarkan prompt pengguna }
```

KAMUS

urutan, sort: integer
urutanLihat, sortLihat: string
sortedList: uList

MenuLihat(input/output urutan, sort: integer, input/output urutanLihat, sortLihat: string, input pilihan: integer)

SortList(uList: UserList, urutan, sort: integer) -> UserList

ALGORITMA

```
MenuLihat(urutan, sort, urutanList, sortList,
pilihan)
    if (strcmp(sortLihat, "Aura") = 0) then
        return
    sortedList <- SortList(uList, urutan, sort)
    output("...") { Menampilkan hasil sesuai dengan hasil
prompt di Menu Lihat }
```

10.8 F08 – Cari User

```
procedure PilihanCari(input uList: UserList, input prompt:
string)
{ I.S. prompt valid, uList terdefinisi }
{ F.S. Mengarahkan ke prosedur cari (CariUser, CariDokter,
atau CariPasien) }
```

KAMUS

pasienList, dokterList: UserList

ALGORITMA

```
if (strcmp(prompt, "CARI_USER") = 0) then
    CariUser(uList, 1);
if (strcmp(prompt, "CARI_PASIEN") = 0) then
    CreateList(pasienList)
    CopyList(uList, pasienList)
    CariUser(pasienList, 2)
if (strcmp(prompt, "CARI_DOKTER") = 0) then
    CreateList(dokterList)
    CopyList(uList, dokterList)
    CariUser(dokterList, 3)
```

```
procedure CariUser(input uList: UserList, input pilihan:
int)
{ I.S. uList terdefinisi dan valid, pilihan valid }
{ F.S. Menampilkan list pengguna berdasarkan prompt
pengguna }
```

KAMUS

```

basis, id: integer
basisCari, data: string
sortedList, hasilCari: uList
MenuCari(input/output basis: integer, input/output
basisCari, data: string, input pilihan: integer)
SortList(uList: UserList, urutan, sort: integer) ->
UserList
BinarySearchUser(input uList: UserList, input id,
first, last: integer, input/output hasilCari: UserList)

```

ALGORITMA

```

MenuLihat(basis, basisCari, data, pilihan)
sortedList <- SortList(uList, urutan, sort)
if (basis = 1) then
    id = atoi(data)
    BinarySearchUser(sortedList, id, 0,
length(uList-1), hasilCari)
if ((basis = 2) or (basis = 3)) then
    sortListString(sortedList, hasilCari)
    output("...") { Menampilkan hasil pada hasilCari
sesuai dengan hasil prompt di MenuCari }

```

10.9 F09 (D02) – Lihat Antrian

```

procedure PrintRuang(input ruang: string, input denah:
Denah)
{I.S. ruang dan denah sudah terdefinisi
F.S. kapasitas pasien tercetak dokter di ruangan tercetak
pasien-pasien di ruangan tercetak}

```

KAMUS

```

denah : Denah
ruang : string
diluar, cur : integer
temp : Map

```

ALGORITMA


```

    if (diluar = 0) then
        Output("--- Detail Ruangan " + ruang + " ---")
        Output("Kapasitas : " + denah.maxPerRoom)
    else
        Output("===== " + ruang + " =====")
    temp ← map_findMap(RuangtoDokter, ruang)
    if (temp = NULL) then
        Output("Dokter      : -")
        Output("Pasien di dalam ruangan:")
        Output("  Tidak ada pasien di dalam ruangan saat
ini.")

        Output("Pasien di antrian:")
        Output("  Tidak ada pasien di antrian saat ini.")
        return

    Output("Dokter      : ")
    if (temp ≠ NULL) then
        Output("Dr. " + username(USER(Ulist,
userPosByID(temp.value))))
    else
        Output(" -")
    tempq ←
DOKTER(UserID_to_DokterID(temp.value)).antrian.front
    cur ← 1

    Output("Pasien di dalam ruangan:")
    if (tempq = NULL) then
        Output("  Tidak ada pasien di dalam ruangan saat
ini.")
    else
        while tempq ≠ NULL and cur ≤ denah.maxPerRoom do
            Output(" " + cur + ". " + username(USER(Ulist,
userPosByID(tempq.data))))
            cur ← cur + 1
            tempq ← tempq.next

```

```

Output()
if diluar = 0 then
    return
Output("Pasien di antrian:")
if tempq = NULL then
    Output(" Tidak ada pasien di antrian saat ini.")
else
    while tempq ≠ NULL do
        Output(" " + cur + ". " + username(USER(Ulist,
userPosByID(tempq.data))))
        cur ← cur + 1
        tempq ← tempq.next

```

10.10 F10 – Tambah Dokter

prosedur AddDokterList(input id: integer)

{I.S. :

- List dokter (dokterList) telah terdefinisi dan mungkin sudah berisi beberapa dokter.
- ID dokter (id) sudah merupakan ID dari user bertipe dokter.

F.S. :

- Dokter baru ditambahkan ke dokterList dengan:
- ID diisi dengan `id`
- Ruangan diset ke ".." (belum di-assign)
- Spesialisasi default "Umum"
- Queue antrian pasien dikosongkan dan diinisialisasi
- Size antrian diset ke 0
- Jumlah elemen efektif (neff) dokterList bertambah 1.}

KAMUS

```

user, pass : string
len        : integer
temp       : string
newUser    : pointer ke User

```

ALGORITMA

```
Output("Username: ")
Input(user)
if set_contains(setUser, user) = 1 then
    Output("Sudah ada Dokter bernama " + user + "!")
    → 0
Output("Pass: ")
Input(pass)
maxID ← maxID + 1
len ← uList.len
temp ← to_string(maxID)
newUser ← alamat uList.contents[len]
set_insertData(setUser, user, len)
strcpy(newUser.field[0], to_string(maxID))
strcpy(newUser.field[1], user)
strcpy(newUser.field[2], pass)
strcpy(newUser.field[3], "dokter")
banarich[len] ← 0
uList.len ← uList.len + 1
AddDokterList(maxID)
Output("Dokter " + user + " berhasil ditambahkan!")
→ 1
```

10.11 F11 – Diagnosis

procedure DIAGNOSIS(currentUser: User, ruangList: ListRuangan, userList: UserList, penyakitList: PenyakitList)
{I.S. currentUser merupakan dokter yang sedang login.
RuangList memuat informasi ruangan beserta daftar pasien.
UserList berisi data user (dokter dan pasien).
PenyakitList berisi daftar penyakit dari file penyakit.csv.

F.S. Pasien pertama di ruangan yang ditugaskan ke dokter berhasil didiagnosis, dan data penyakit pasien diperbarui sesuai hasil diagnosis.}

KAMUS

idDokter, idPasien, idxPasien, f_suhu, f_saturasi, f_berat
: float i_sistol, i_diastol, i_detak, i_gula, i_tinggi,
i_kolesterol, i_trombosit, ditemukan, i: integer
q : pointer ke Queue pasien : pointer ke User
penyakit : Penyakit

ALGORITMA

```
if currentUser.field[3] ≠ "dokter" then
    i traversal[0..4]
        Output(currentUser.field[i])
    Output("Anda tidak memiliki izin untuk melakukan
diagnosis.")
    return

idDokter ← UserID_to_DokterID(masterID)

if DOKTER(idDokter).ruangKerja = ".." then
    Output("Dokter belum diassign ke ruangan
manapun.")
    return

q ← DOKTER(idDokter).antrian

if queue_isEmpty(q) then
    Output("Tidak ada pasien untuk diperiksa!")
    return

idPasien ← queue_front(q).data
idxPasien ← -1

i traversal[0..userList.len - 1]
```

```

        if atoi(userList.contents[i].field[0]) = idPasien
then
            idxPasien ← i
            break

    if idxPasien = -1 then
        Output("Data pasien tidak ditemukan.")
        return

    pasien ← alamat userList.contents[idxPasien]

    if riwayat(pasien) ≠ "-" dan panjang(riwayat(pasien))
> 0 then
        Output(username(pasien) + " terdiagnosa penyakit "
+ riwayat(pasien) + "!")
        return
    PASIEN(UserID_to_PasienID(atoi(pasien.field[0]))).sudahDia
gnosis ← 1

    f_suhu      ← atof(suhu(pasien))
    i_sistol    ← atoi(sistol(pasien))
    i_diastol   ← atoi(diastol(pasien))
    i_detak     ← atoi(detak(pasien))
    f_saturasi  ← atof(saturasi(pasien))
    i_gula      ← atoi(gula(pasien))
    f_berat     ← atof(berat(pasien))
    i_tinggi    ← atoi(tinggi(pasien))
    i_kolesterol ← atoi(kolesterol(pasien))
    i_trombosit ← atoi(trombosit(pasien))

    ditemukan ← 0

    i traversal[0..penyakitList.len - 1]
        penyakit ← penyakitList.contents[i]

```

```

    if
        f_suhu ≥ atof(suhu_min(penyakit)) AND
f_suhu ≤ atof(suhu_max(penyakit)) AND
        i_sistol ≥ atoi(sistol_min(penyakit)) AND
i_sistol ≤ atoi(sistol_max(penyakit)) AND
        i_diastol ≥ atoi(diastol_min(penyakit)) AND
i_diastol ≤ atoi(diastol_max(penyakit)) AND
        i_detak ≥ atoi(detak_min(penyakit)) AND
i_detak ≤ atoi(detak_max(penyakit)) AND
        f_saturasi ≥ atof(saturasi_min(penyakit)) AND
f_saturasi ≤ atof(saturasi_max(penyakit)) AND
        i_gula ≥ atoi(gula_min(penyakit)) AND
i_gula ≤ atoi(gula_max(penyakit)) AND
        f_berat ≥ atof(berat_min(penyakit)) AND
f_berat ≤ atof(berat_max(penyakit)) AND
        i_tinggi ≥ atoi(tinggi_min(penyakit)) AND
i_tinggi ≤ atoi(tinggi_max(penyakit))
        // Tambahan kondisi bisa diaktifkan bila
        diperlukan:
        // AND i_kolesterol ≥
        atoi(kolesterol_min(penyakit)) AND i_kolesterol ≤
        atoi(kolesterol_max(penyakit))
        // AND i_trombosit ≥
        atoi(trombosit_min(penyakit)) AND i_trombosit ≤
        atoi(trombosit_max(penyakit))
    then
        riwayat(pasien) ← penyakit(penyakit)
        Output(username(pasien) + " terdiagnosa penyakit "
+ riwayat(pasien) + "!")
        ditemukan ← 1
        break

    if (ditemukan = 0) then

```

```

        riwayat(pasien) ← "-"
        Output(username(pasien) + " tidak terdiagnosis
penyakit apapun!")

```

10.12 F12 – Ngobatin

```

procedure NGOBATIN(currentUser: User, ruangList:
ListRuang, userList: UserList, obatPenyakitList:
ObatPenyakitList)
{I.S. currentUser adalah dokter yang sedang login,
ruangList berisi seluruh ruangan dan pasien, userList
berisi semua user, dan obatPenyakitList memuat relasi
penyakit dan obat.
F.S. Obat yang sesuai dengan penyakit pasien diberikan
oleh dokter jika tersedia, dan dipindahkan dari inventory
dokter ke pasien.}

```

KAMUS

```

        idDokter, idPasien, idxPasien, _len, i, j,
idPenyakit, idObat      : integer
        q: pointer to Queue
        pasien: pointer to User
        penyakit: string
        obatPenyakit[MAX_USER]: array of integer

```

ALGORITMA

```

        if currentUser.field[3] ≠ "dokter" then
            Output("Anda tidak memiliki izin untuk memberikan
obat.")
            Return
        idDokter ← UserID_to_DokterID(masterID)
        if DOKTER(idDokter).ruangKerja = ".." then
            Output("Dokter belum diassign ke ruangan
manapun.")
            Return
        q ← DOKTER(idDokter).antrian
        if queue_isEmpty(q) then

```

```

        Output("Tidak ada pasien di ruangan Anda.")
        Return
idPasien ← queue_front(q).data
idxPasien ← -1
i traversal[0..userList.len - 1]
    if atoi(userList.contents[i].field[0]) = idPasien
then
        idxPasien ← i
        Break
if idxPasien = -1 then
    Output("Pasien tidak ditemukan.")
    Return
pasien ← alamat userList.contents[idxPasien]
penyakit ← pasien.field[4]

if penyakit = "-" or panjang(penyakit) = 0 then
    Output("Pasien tidak memiliki penyakit.")
    Return
If
PASIEN(UserID_to_PasienID(atoi(pasien.field[0]))).sudahDia
gnosis = 0 then
    Output("Pasien belum di diagnosis!")
    Return
idPenyakit ← -1
i traversal[0..Plist.len - 1]
    if Plist.contents[i].field[1] = penyakit then
        idPenyakit ← atoi(Plist.contents[i].field[0])
if idPenyakit = -1 then
    Output("Penyakit tidak ditemukan!")
    Return
idObat ← -1 _len ← 0
Output("id penyakit : " + idPenyakit)

i traversal[0..obatPenyakitList.len - 1]

```



```

        if atoi(obatPenyakitList.contents[i].field[1]) =
idPenyakit then
            idObat ←
atoi(obatPenyakitList.contents[i].field[0])

obatPenyakit[atoi(obatPenyakitList.contents[i].field[2])]
← idObat

        _len ← _len + 1
        Output("Pasien memiliki penyakit " + penyakit)
        Output("Obat yang harus diberikan:")

        i traversal[0.._len - 1]
            j traversal[0..Olist.len - 1]
                if atoi(Olist.contents[j].field[0]) =
obatPenyakit[i + 1] then
                    Output((i + 1) + ". " +
Olist.contents[j].field[1])

```

10.13 F13 – Aku boleh pulang ga, dok?

```

function pulangdok()-> integer
{I.S. }
{F.S. Menampilkan status pasien (apakah dapat dipulangkan)
}

```

KAMUS LOKAL

```

Pasien      : pointer to User
penyakit    : string
i, idx      : integer
idPenyakit  : integer
len         : integer
arr1, arr2  : array of integer
perut       : Stack
beda        : integer
idDokter    : integer

```

ALGORITMA

```
{Cari pasien berdasarkan masterID}
i traversal [0 ... Ulist.len - 1]
    if atoi(Ulist.contents[i].field[0]) = masterID
then
    Pasien ← &Ulist.contents[i]
    break

penyakit ← riwayat(*Pasien)

    if PASIEN(UserID_to_PasienID(atoi(Pasien-
>field[0]))).sudahDiagnosis = 0 then
        output("Kamu belum menerima diagnosis apapun dari
dokter, jangan buru-buru pulang!")
        return 0

    output("Dokter sedang memeriksa keadaanmu...")

    if (riwayat(*Pasien) = "-" or len(riwayat(*Pasien)) =
0) then
        output("Selamat! Kamu sudah dinyatakan sembuh oleh
dokter. Silahkan pulang dan semoga sehat selalu!")
        return 1

    if Pasien->inventoryObat.size ≠ 0 then
        output("Masih ada obat yang belum kamu habiskan,
minum semuanya dulu yukk!")
        return 0

{Cari ID penyakit dari nama penyakit}
idPenyakit ← -1
i traversal [0 ... Plist.len - 1]
    if PENYAKIT(Plist, i).field[1] = penyakit then
        idPenyakit ← atoi(PENYAKIT(Plist, i).field[0])
```

```

    {Buat list obat yang terkait dengan penyakit tersebut}
    idx ← 0
    NewOPlist.len ← 0
    i traversal [0 ... OPlist.len - 1]
        if atoi(OPlist.contents[i].field[1]) = idPenyakit
then
            NewOPlist.contents[idx] ← OPlist.contents[i]
            NewOPlist.len ← NewOPlist.len + 1
            idx ← idx + 1

{ Pindahkan id obat sesuai urutan }
len ← NewOPlist.len
i traversal [0 ... len - 1]
    j ← 0
    while atoi(NewOPlist.contents[j].field[2]) ≠ i + 1
do
        j ← j + 1
        arr1[i] ← atoi(NewOPlist.contents[j].field[0])

{Ambil isi perut pasien ke arr2}
perut ← Pasien->perut
i traversal [len - 1 ... 0] dengan langkah -1
    arr2[i] ← perut.top->data
    stack_pop(perut)

{Bandingkan arr1 dan arr2}
beda ← 0
i traversal [0 ... len - 1]
    if arr1[i] ≠ arr2[i] then
        beda ← 1
        break

if beda = 1 then

```

```

        output("Maaf, tapi kamu masih belum bisa pulang!")
        output("Urutan peminuman obat yang diharapkan:")
        i traversal [0 ... len - 1]
            j ← 0
            while atoi(Olist.contents[j].field[0]) ≠
arr1[i] do
                j ← j + 1
                output(Olist.contents[j].field[1])
                if i ≠ len - 1 then output(" -> ")
                output baris baru

        output("Urutan obat yang kamu minum:")
        i traversal [0 ... len - 1]
            j ← 0
            while atoi(Olist.contents[j].field[0]) ≠
arr2[i] do
                j ← j + 1
                output(Olist.contents[j].field[1])
                if i ≠ len - 1 then output(" -> ")
                output baris baru

        output("Silahkan kunjungi dokter untuk meminta
penawar yang sesuai!")
        return 0

    {Pasien berhasil sembuh}
    idDokter ←
PASIEN(UserID_to_PasienID(atoi(Pasien.field[0]))).idDokter
    queue_pop(DOKTER(idDokter).antrian)
    DOKTER(idDokter).aura ← DOKTER(idDokter).aura + 1

PASIEN(UserID_to_PasienID(atoi(Pasien.field[0]))).idDokter
← -1

```

```

    {Reset field pasien di user.csv}
    i traversal [4 ... 15]
        Pasien->field[i] ← ""

    output("Selamat! Kamu sudah dinyatakan sembuh oleh
dokter. Silahkan pulang dan semoga sehat selalu!")
    return 1

```

10.14 F14 – Daftar Checkup

Procedure DaftarCheckup()

I.S. :

- User dengan ID `masterID` merupakan pasien dan telah login.
- Pasien belum terdaftar dalam antrian check-up (jika sudah, prosedur akan langsung berhenti).
- Data dokter, user, dan antrian dokter telah terdefinisi.
- Field check-up (suhu, tekanan darah, dsb.) pada user bisa kosong atau belum valid. F.S. :
- Jika input valid dan pasien belum terdaftar:
- Field check-up pada pasien diisi berdasarkan input dari pengguna.
- Pasien ditambahkan ke antrian dokter pilihan.
- Informasi pendaftaran dan posisi antrian ditampilkan.
- Jika pasien sudah terdaftar:
- Menampilkan pesan bahwa pasien sudah berada dalam antrian, tanpa melakukan perubahan data.

Kamus

```

user: pointer ke User
cnt, i, pick, biaya: integer
temp: array of integer [0..MAX_USER-1]
aura: integer
ruang, namaDokter, spesialisasi: string

```

Algoritma

```

        if PASIEN(UserID_to_PasienID(masterID)).idDokter ≠ -1
then
        output("Anda sudah terdaftar dalam antrian check-
up!")

        output("Silakan selesaikan check-up yang sudah
terdaftar terlebih dahulu.")
        return
        output("Silakan masukkan data check-up Anda: ")

user ← alamat USER(Ulist, userPosByID(masterID))

output("Suhu Tubuh (Celcius): ")
input(suhu(user))
while atof(suhu(user)) < 0 do
        output("Suhu tubuh harus berupa angka positif!")
        output("Suhu Tubuh (Celcius): ")
        input(suhu(user))

output("Tekanan Darah (sistol/diastol, contoh 120 8):
")
input(sistol(user), diastol(user))
while atoi(sistol(user)) < 0 V atoi(diastol(user)) < 0
do
        output("Tekanan darah harus berupa angka
positif!")
        output("Tekanan Darah (sistol/diastol, contoh 120
8): ")
        input(sistol(user), diastol(user))

output("Detak Jantung (bpm): ")
input(detak(user))
while atoi(detak(user)) < 0 do
        output("Detak Jantung harus berupa angka
positif!")

```

```

        output("Detak Jantung (bpm): ")
        input(detak(user))

    output("Saturasi Oksigen (%): ")
    input(saturasi(user))
    while atof(saturasi(user)) < 0 V atof(saturasi(user))
> 100 do
        output("Saturasi oksigen harus berada dalam
rentang 0 sampai 100!")
        output("Saturasi Oksigen (%): ")
        input(saturasi(user))

    output("Kadar Gula Darah (mg/dL): ")
    input(gula(user))
    while atoi(gula(user)) < 0 do
        output("Kadar gula darah harus berupa angka
positif!")
        output("Kadar Gula Darah (mg/dL): ")
        input(gula(user))

    output("Berat Badan (kg): ")
    input(berat(user))
    while atof(berat(user)) ≤ 0 do
        output("Berat badan harus berupa angka positif!")
        output("Berat Badan (kg): ")
        input(berat(user))

    output("Tinggi Badan (cm): ")
    input(tinggi(user))
    while atoi(tinggi(user)) ≤ 0 do
        output("Tinggi badan harus berupa angka positif!")
        output("Tinggi Badan (cm): ")
        input(tinggi(user))

```

```

    output("Kadar Kolestrol (mg/dL): ")
    input(kolesterol(user))
    while atoi(kolesterol(user)) < 0 do
        output("Kolesterol badan harus berupa angka
positif!")
        output("Kolesterol Badan (mg/dL): ")
        input(kolesterol(user))

    output("Trombosit (ribu/ $\mu$ L): ")
    input(trombosit(user))
    while atoi(trombosit(user)) < 0 do
        output("Trombosit harus berupa angka positif!")
        output("Trombosit (ribu/ $\mu$ L): ")
        input(trombosit(user))

    cnt  $\leftarrow$  0
    i traversal[0..dokterList.neff - 1]
        if DOKTER(i).ruangKerja  $\neq$  ".." then
            temp[cnt]  $\leftarrow$  i
            cnt  $\leftarrow$  cnt + 1
            output(cnt + ". Dr. " + username(USER(Ulist,
userPosByID(DOKTER(i).id))) +
                " - Spesialisasi " + DOKTER(i).spesialisasi
+
                " - Ruangan " + DOKTER(i).ruangKerja +
                " (Antrian: " + DOKTER(i).queue_size + "
orang)" +
                " - Aura " + DOKTER(i).aura +
                " - Biaya " + ((DOKTER(i).aura + 1)  $\times$  120))

    if cnt = 0 then
        output("Tidak ada dokter yang dapat dipilih!")
        return

```



```

    output("Pilih dokter (1.." + cnt + "): ")
    input(pick)
    while pick ≤ 0 V pick > cnt do
        output("Pilihan tidak valid!")
        output("Pilih dokter (1.." + cnt + "): ")
        input(pick)

    biaya ← 120 × (DOKTER(temp[pick - 1]).aura + 1)

    if banarich[userPosByID(masterID)] < biaya then
        output("Maaf Banarich kamu tidak cukup!")
        output("Silakan pilih dokter lain atau mainkan
mesin gacha!")
        return

    banarich[userPosByID(masterID)] ←
banarich[userPosByID(masterID)] - biaya
    banarich[userPosByID(DOKTER(temp[pick - 1]).id)] ←
banarich[userPosByID(DOKTER(temp[pick - 1]).id)] + biaya ×
0.8
    UpdateKasRumahSakit(biaya × 0.2)

    output("Pendaftaran check-up berhasil!")
    AddPasien_to_Dokter(masterID, DOKTER(temp[pick -
1]).id)
    output("Anda terdaftar pada antrian Dr. " +
username(USER(Ulist, userPosByID(DOKTER(temp[pick -
1]).id))) +
        " di ruangan " + DOKTER(temp[pick - 1]).ruangKerja
+ ".")
    output("Posisi antrian Anda: " + DOKTER(temp[pick -
1]).queue_size)

```

10.15 F15 – Antrian Saya!

Procedure CekAntrian(id: integer)

Kamus:

UserId, idDokter, sz, pos : integer

tempq : pointer to Node

Algoritma

```
sz ← 1; pos ← 0
UserId ← UserID_to_PasienID(id)
if PASIEN(userId).idDokter = -1 then
    output("Anda belum terdaftar dalam antrian check-
up!")
    output("Silakan daftar terlebih dahulu dengan
command DAFTAR_CHECKUP.")
    return

tempq ← DOKTER(PASIEN(userId).idDokter).antrian.front
while tempq ≠ NULL do
    if tempq.data = id then
        pos ← sz
        sz ← sz + 1
        tempq ← tempq.next

if pos < denah.maxPerRoom then
    output("Anda sedang berada di dalam ruangan
dokter!")
else
    idDokter ← PASIEN(userId).idDokter
    output("Status antrian Anda:")
    output("Dokter: Dr. " + username(USER(Ulist,
userPosByID(DOKTER(idDokter).id))))
    output("Ruangan: " + DOKTER(idDokter).ruangKerja)
    output("Posisi antrian: " + (pos + 1 -
denah.maxPerRoom) +
        " dari " + (sz - denah.maxPerRoom))
```

10.16 F16 – Minum Obat

procedure minumObat()

{I.S. inventoryObat terdefinisi. Stack Perut terdefinisi (mungkin kosong)}.

F.S. Satu obat dipindahkan dari inventoryObat ke stack Perut.}

KAMUS LOKAL

pasien	: pointer to User
i	: integer
current	: Address
id_obat	: integer
id_obat_hapus	: integer
no_obat_hapus	: integer
nomor	: integer
index	: integer
tempNamaObat	: pointer to List
tempIdObat	: pointer to List
id_obat_str	: array of char

ALGORITMA

```
{ Cari pasien dengan masterID }  
i traversal [0..Ulist.len - 1]  
    if atoi(Ulist.contents[i].field[0]) = masterID  
then  
    pasien ← &Ulist.contents[i]  
    break  
  
output("===== DAFTAR OBAT =====")  
  
current ← pasien→inventoryObat.head  
alokasi(tempNamaObat, sizeof(List))
```

```

list_create(tempNamaObat)
alokasi(tempIdObat, sizeof(List))
list_create(tempIdObat)
nomor ← 1
{ Simpan nama dan id obat ke array sementara dan
tampilkan }
while current ≠ NIL do
    id_obat ← current↑.data
    i traversal[0..Olist.len - 1]
    if id_obat = atoi(Olist.contents[i].field[0])
then
        output(nomor + ". " +
Olist.contents[i].field[1])
        id_obat_str ← konversi id_obat ke string
        list_add(tempIdObat, id_obat_str)
        list_add(tempNamaObat,
Olist.contents[i].field[1])
        nomor ← nomor + 1
        break

    current ← current↑.next

{ Input pilihan user }
output("Pilih obat untuk diminum: ")
input(no_obat_hapus)
while (no_obat_hapus < 1) or (no_obat_hapus ≥ nomor)
do
    output("Pilihan nomor tidak tersedia!")
    input(no_obat_hapus)

    output("GLEKGLEKGLEK... " + tempNamaObat[no_obat_hapus
- 1] + " berhasil diminum!!")

    id_obat_hapus ← atoi(tempIdObat[no_obat_hapus - 1])

```

```

    { Cari indeks node yang menyimpan id_obat_hapus }
    index ← 0
    current ← pasien→inventoryObat.head
    while (current ≠ NIL) and (current↑.data ≠
id_obat_hapus) do
        current ← current↑.next
        index ← index + 1

    linked_delete(pasien→inventoryObat, index)
    dealokasi(tempIdObat)
    dealokasi(tempNamaObat)

    { Tambahkan id obat yang diminum ke stack perut }
    stack_push(pasien→perut, id_obat_hapus)

```

10.17 F17 – Minum Penawar

```

void minumPenawar()
{ I.S. Stack Perut terdefinisi (mungkin kosong)
  F.S. Obat terakhir yang diminum dipindahkan kembali ke
inventoryObat.}

```

KAMUS LOKAL

```

pasien      : pointer to User
i, id_obat  : integer
nama_obat   : string

```

ALGORITMA

```

{ Cari pasien dengan masterID }
  i traversal [0 ... Ulist.len - 1]
    if atoi(Ulist.contents[i].field[0]) = masterID
then
        pasien ← &Ulist.contents[i]
        break

  if stack_isEmpty(pasien→perut) then

```

```

        output("Perut kosong!! Belum ada obat yang
dimakan.")
    else
        id_obat ← stack_top(pasien->perut.top)

        { Cari nama obat berdasarkan id_obat di Olist }
        i traversal [0 ... Olist.len - 1]
            if id_obat = atoi(Olist.contents[i].field[0])
then
                nama_obat ← Olist.contents[i].field[1]
                break

        if nama_obat ≠ NULL then
            output("Uwekkk!!! " + nama_obat + " keluar dan
kembali ke inventory")
            stack_pop(pasien.perut)

        { Tambahkan id obat kembali ke inventory }
        linked_insertEnd(pasien.inventoryObat, id_obat)

```

10.18 F18 – Exit & D04 – Save

```

procedure exitProgram()
{I.S. Program sedang berjalan.
  F.S. Jika pengguna memilih keluar, maka program berhenti
(dan dapat menyimpan file jika diinginkan).
  Jika tidak, program kembali ke menu utama.}

```

KAMUS LOKAL

```

prompt   : array of char
valid    : integer
choice   : integer

```

ALGORITMA

```

valid ← 0

```

```

    { Loop hingga pengguna memberikan input valid }
    while valid = 0 do
        output("Apakah Anda yakin ingin keluar? (y/n): ")
        input(prompt)

        if prompt ∈ {"y", "Y", "yes", "YES", "YA", "ya"}
then
            choice ← 1 // Pilihan 1 untuk keluar
            valid ← 1
        else if prompt ∈ {"n", "N", "no", "NO", "TIDAK",
"tidak"} then
            output("Kembali ke menu utama.")
            choice ← 2 // Pilihan 2 untuk kembali ke menu
utama
            valid ← 1
        else
            output("Masukan tidak valid. Silakan coba
lagi.")
            valid ← 0

{ User yang ingin keluar ditanya apakah ingin simpan file}
    if choice = 1 then
        output("Apakah anda mau melakukan penyimpanan file
yang sudah diubah? (y/n): ")
        input(prompt)

        if prompt ∈ {"y", "Y", "yes", "YES", "YA", "ya"}
then
            saveFile()
        else if prompt ∈ {"n", "N", "no", "NO", "TIDAK",
"tidak"} then
            output("Data tidak disimpan.")
            return

```

```
else if choice = 2 then  
    return
```

```
function save_all(folderName: string, obatList: pointer to  
ObatList, obatPenyakitList: pointer to ObatPenyakitList,  
penyakitList: pointer to PenyakitList, userList: pointer  
to UserList) -> integer  
{ I.S. folderName terdefinisi. Data pada obatList,  
obatPenyakitList, penyakitList, dan userList terdefinisi  
dan vali. }  
{ F.S. Membuat folder 'data/folderName' jika belum ada.  
Menyimpan semua data dari struktur data ke dalam file di  
dalam folder 'data/folderName'. Mengembalikan 0 sebagai  
tanda proses selesai }
```

KAMUS LOKAL

```
dataExists : integer  
subfolderExists : integer  
fullFolderPath : string  
command : string
```

ALGORITMA

```
dataExists <- folder_exists("data")  
strcpy(fullFolderPath, "data/");  
strcat(fullFolderPath, folderName);  
subfolderExists <- folder_exists(fullFolderPath)  
output("\nSaving...")  
if (dataExists = 0) then  
    output("\nMembuat folder data...\n")  
    system("mkdir data")  
if (subfolderExists = 0) then  
    output("Membuat folder " + fullFolderPath)  
    command <- "mkdir " + fullFolderPath  
    strcpy(command, "mkdir ");  
    strcat(command, fullFolderPath);  
    system(command);
```



```

    save_obat (obatList)
    save_obatpenyakit (obatPenyakitList)
    save_penyakit (penyakitList) save_user (userList)
    save_banarich()
    output("Berhasil menyimpan data di folder " +
fullFolderPath + "!")
    -> 0

```

10.19 D03 – Load

function load_all(folder: string, denah: pointer to Denah, uList: pointer to UserList, Olist: pointer to ObatList, Oplist: pointer to ObatPenyakitList, Plist: pointer to PenyakitList) -> integer

{ I.S. Data pada tiap ADT terdefinisi dan valid }
{ F.S. Memuat semua data dari folder eksternal ke dalam struktur data masing-masing. Mengembalikan 1 jika berhasil, 0 jika gagal. }

KAMUS LOKAL

fp: SEQFILE of
(*) string
(1) MARK
filepath: string
procedure path (input/output dest: string, input dir, name: string)

ALGORITMA

```

path(filePath, folder, "config.txt"
assign(fp, filePath)
read(fp)
if (fp = NULL) then
    output("Folder " + folder " tidak ditemukan")
    -> 1
output("Loading..")
load_obat(folder, Olist)
load_user(folder, Ulist)

```

```

    load_obatpenyakit(folder, OPlist)
    load_penyakit(folder, Plist)
    load_config(folder, denah, Ulist)
    load_banarich(folder);
    output("Selamat datang kembali di rumah sakit
Nimons!")
    -> 0

```

10.20 B02 - Denah Dinamis

procedure UbahDenah(newRow: integer, newCol: integer)
{ Mengubah ukuran denah rumah sakit jika tidak ada dokter
di luar batas denah baru }

KAMUS

row, col: integer
ruang: array [0..4] of char
i, j: integer

ALGORITMA

```

row ← denah.M.rows
col ← denah.M.cols

ruang[0] ← 'A'
ruang[1] ← '1'
ruang[2] ← '\0'

i traversal [0..row - 1]
    ruang[1] ← '1'
    j traversal [0..col - 1]
        if denah.M.contents[i][j] ≠ -1 and (i ≥ newRow
or j ≥ newCol) then
            output("Tidak dapat mengubah ukuran
denah.")

```

```

        output("Ruangan " + ruang + " masih
ditempati oleh Dr. " +
        username(USER(Ulist,
userPosByID(denah.M.contents[i][j]))))
        output("Silakan pindahkan dokter terlebih
dahulu.")
        return

        ruang[1] ← ruang[1] + 1

        ruang[0] ← ruang[0] + 1

        ROWS(MAT(denah)) ← newRow
        COLS(MAT(denah)) ← newCol
        output("Denah rumah sakit berhasil diubah menjadi " +
newRow + " baris dan " + newCol + " kolom")

```

procedure PindahDokter(oldRoom: string, newRoom: string)
{ Memindahkan dokter dari ruangan lama ke ruangan baru
jika memungkinkan }

KAMUS

```

oldRow, oldCol: integer
newRow, newCol: integer
id: integer

```

ALGORITMA

```

oldRow ← ord(oldRoom[0]) - ord('A')
oldCol ← ord(oldRoom[1]) - ord('1')
newRow ← ord(newRoom[0]) - ord('A')
newCol ← ord(newRoom[1]) - ord('1')

id ← atoi(id(USER(Ulist,
userPosByID(denah.M.contents[oldRow][oldCol]))))

```

```

    if denah.M.contents[newRow][newCol] ≠ -1 then
        output("Pemindahan gagal. Ruangan " + newRoom + "
sudah ditempati.")
        return

    if denah.M.contents[oldRow][oldCol] = -1 then
        output("Pemindahan gagal. Ruangan " + oldRoom + "
kosong.")
        return

    output("Dr. " + username(USER(Ulist,
userPosByID(denah.M.contents[oldRow][oldCol]))) +
        " berhasil dipindahkan dari ruangan " + oldRoom
+ " ke ruangan " + newRoom + ".")

    denah.M.contents[oldRow][oldCol] ← -1
    denah.M.contents[newRow][newCol] ← id

    map_delete(RuangtoDokter, oldRoom)
    map_insert(RuangtoDokter, newRoom, id)

    id ← UserID_to_DokterID(id)
    DOKTER(id).ruangKerja ← newRoom

```

10.21 B04 - Banarich

```

procedure LihatDompel()
{ Mencetak banyaknya Banarich milik user }

```

KAMUS LOKAL

```

    pos: integer
    role: string

```

ALGORITMA

```

    pos ← userPosByID(masterID);
    role ← Ulist.contents[pos].field[3]
    output("")

```

```

    output("Banarich milik kamu saat ini adalah" +
banarich[pos] + " Banarich.")
    output("")

```

procedure LihatFinansial()

```
{ Mencetak banyaknya Banarich milik rumah sakit }
```

KAMUS LOKAL

```
pos, i: integer
```

ALGORITMA

```

pos <- -1
i traversal [0..Ulist.len-1]
    if Ulist.contents[i].field[3] = "manager" then
        pos <- i
        i <- Ulist.len - 1
{ i = Ulist.len }
if pos = -1 then
    output("")
    output("Manager tidak ditemukan")
    return
output("")
    output("Banarich milik rumah sakit saat ini adalah "
+ banarich[pos] + " Banarich.")
    output("")

```

procedure Gacha()

```
{ Simulasi mesin Gacha yang menghasilkan koin secara acak
dan memperbarui saldo user }
```

KAMUS LOKAL

```

random_lcg(n: integer, seed: integer) -> integer
result: integer

```

ALGORITMA

```

result <- random_lcg(MasterID, globalSeed)
globalSeed <- globalSeed + 1
result <- result mod 200

output("")
    output("Mesin GACHA-X 2.0 sedang bekerja...")
    output("")

if globalSeed > 400 then

```

```

        result <- result * -1
        output("Yah.. runtkad " + result + " koin")
        output("")
    else { globalSeed ≤ 400 }
        depend on result
            (0 ≤ result ≤ 50): output("Cuma dapat " +
result + " koin...")
                                output("")
            (51 ≤ result ≤ 100): output("Dapat " + result
+ " koin. Yah, standar lah.")
                                output("")
            (101 ≤ result ≤ 150): output(result + "koin!
Udah mulai keliatan hoki.")
                                output("")
            (151 ≤ result ≤ 190): output("Mantap! " +
result + " koin berhasil dikantongi")
                                output("")
            (191 ≤ result ≤ 199): output("GACOR! " +
result + " koin langsung masuk ke dompet!")
                                output("")

    banarich[userPosByID(masterID)] <-
banarich[userPosByID(masterID)] + 1

```

10.22 B06 - Mainin Antrian

procedure SkipAntrian()
{ Memindahkan pasien ke posisi terdepan dalam antrian
dokter jika memenuhi syarat }

KAMUS

idDokter: integer
oldQueue, newQueue: Queue pointer
temp: Address

ALGORITMA

```

    if role(USER(Ulist, userPosByID(masterID))) ≠ "pasien"
then
        output("Akses tidak dimiliki!")
        return

```

```

    idDokter ←
PASIEN(UserID_to_PasienID(masterID)).idDokter

    if idDokter = -1 then
        output("Skip antrian gagal! Anda tidak sedang
terdaftar dalam antrian manapun!")
        return

    if DOKTER(idDokter).antrian->front->data = masterID
then
        output("Anda sudah berada di posisi paling depan
antrian! Tidak bisa skip lagi!")
        return

    oldQueue ← DOKTER(idDokter).antrian
    alokasi(newQueue) // alokasi memori untuk Queue baru
    queue_create(newQueue)

    // Tambah masterID di depan
    queue_push(newQueue, masterID)

    // Salin antrean lama, kecuali masterID
    temp ← oldQueue->front
    while temp ≠ NULL do
        if temp->data ≠ masterID then
            queue_push(newQueue, temp->data)

        temp ← temp->next

    // Hapus antrean lama dan ganti dengan antrean baru
    queue_clear(oldQueue)
    dealokasi(oldQueue)
    DOKTER(idDokter).antrian ← newQueue

```

```

    output("Anda berhasil maju ke depan antrian Dr. " +
          username(USER(Ulist,
userPosByID(DOKTER(idDokter).id))) +
          " di ruangan " + DOKTER(idDokter).ruangKerja +
"!")
    output("Posisi antrian Anda sekarang: 1")

```

procedure KeluarAntrian()

{ Mengeluarkan pasien dari antrian dokter jika pasien sedang mengantri }

KAMUS

idDokter: integer
oldQueue, newQueue: Queue pointer
temp: Address

ALGORITMA

```

    if role(USER(Ulist, userPosByID(masterID))) ≠ "pasien"
then
        output("Akses tidak dimiliki!")
        return

```

```

    idDokter ←
PASIEN(UserID_to_PasienID(masterID)).idDokter

```

```

    if idDokter = -1 then
        output("Pembatalan antrian gagal! Anda tidak
sedang terdaftar dalam antrian manapun!")
        return

```

```

    oldQueue ← DOKTER(idDokter).antrian
    alokasi(newQueue) {alokasi memori untuk Queue baru}
    queue_create(newQueue)

```



```

temp ← oldQueue->front
while temp ≠ NULL do
    if temp.data ≠ masterID then
        queue_push(newQueue, temp->data)

    temp ← temp->next

queue_clear(oldQueue)
dealokasi(oldQueue)
DOKTER(idDokter).antrian ← newQueue

output("Anda berhasil keluar dari antrian Dr. " +
        username(USER(Ulist,
userPosByID(DOKTER(idDokter).id))) +
        " di ruangan " + DOKTER(idDokter).ruangKerja +
        ".")

PASIEN(UserID_to_PasienID(masterID)).idDokter ← -1

```

Bagian	Modul/Prosedur/Fungsi	Spesifikasi
F04 – Lupa Passwo rd	<u>function</u> runLengthEncoding(na me: <u>string</u>) → <u>string</u>	Mengembalikan username setelah RLE
F06 (D01) – Denah Rumah Sakit	procedure CreateDenah(output denah: Denah)	I.S. denah belum terdefinisi F.S. denah terdefinisi dari file config.txt
	<u>function</u> ToLower(str: ^char) → ^char	Mengubah seluruh karakter dalam string menjadi huruf kecil

F07 – Lihat User	function ToUpper(str: ^char) → ^char	Mengubah seluruh karakter dalam string menjadi huruf kapital
	function sortList(uList: UserList, urutan: integer, sort: integer) → UserList	Mengembalikan UserList yang telah disortir berdasarkan urutan dan mode (ascending/descending)
	procedure MenuLihat(var urutan: integer, var sort: integer, var sortLihat: string, var urutanLihat: string)	Menampilkan dan menerima input pilihan urutan dan metode sortir
	procedure LihatUser(uList: UserList, pilihan: integer)	Menampilkan seluruh data user dari UserList berdasarkan pilihan urutan
	procedure LihatPasien(uList: UserList)	Menampilkan data user yang berperan sebagai pasien
	procedure LihatDokter(uList: UserList)	Menampilkan data user yang berperan sebagai dokter
F08 – Cari User	procedure BinarySearchUser(uList: UserList, id: integer, first: integer, last: integer, var hasilCari: UserList)	Melakukan binary search pada UserList berdasarkan ID dan menyimpan hasil ke hasilCari

	<pre>procedure MenuCari(var basis: integer, var basisCari: string, var data: string, pilihan: integer)</pre>	Menampilkan dan menerima input parameter pencarian
	<pre>procedure CariUser(uList: UserList, pilihan: integer)</pre>	Menampilkan hasil pencarian user berdasarkan basis pilihan
	<pre>procedure CariPasien(uList: UserList)</pre>	Menampilkan hasil pencarian user yang berperan sebagai pasien
	<pre>procedure CariDokter(uList: UserList)</pre>	Menampilkan hasil pencarian user yang berperan sebagai dokter
	<pre>function load_obat(folder: string, l: ^ObatList) → integer</pre>	I.S. Data obat terdefinisi dan valid F.S. Memuat data dari obat.csv dan menyimpannya ke dalam ObatList.
	<pre>function load_penyakit(folder : string, l: ^PenyakitList) → integer</pre>	I.S. Data penyakit terdefinisi dan valid F.S. Memuat data dari penyakit.csv dan menyimpannya ke dalam PenyakitList.
	<pre>function load_user(folder: string, l: ^UserList) → integer</pre>	I.S. Data user terdefinisi dan valid F.S. Memuat data dari user.csv dan menyimpannya ke dalam UserList.
	<pre>function load_obatpenyakit(fo lder: string, l: ^ObatPenyakitList) → integer</pre>	I.S. Data obat penyakit terdefinisi dan valid F.S. Memuat data dari obat-penyakit.csv dan menyimpannya ke dalam ObatPenyakitList.

	<pre>function load(folder: string, listUser: ^UserList, listPenyakit: ^PenyakitList, listObat: ^ObatList, listObatPenyakit: ^ObatPenyakitList) → integer</pre>	<p>I.S. Data dari seluruh data dari file eksternal terdefinisi dan valid</p> <p>F.S. Memuat seluruh data dari folder eksternal (user, penyakit, obat, dan relasi obat-penyakit) ke dalam struktur data masing-masing.</p> <p>Return 1 jika semua berhasil dimuat, 0 jika ada yang gagal.</p>
D04 - Save	<pre>function save_user(l: ^UserList) → integer</pre>	<p>I.S. Data dari UserList terdefinisi dan valid</p> <p>F.S. Menyimpan data UserList ke file eksternal. Mengembalikan 1 jika berhasil, 0 jika gagal.</p>
	<pre>function save_penyakit(l: ^PenyakitList) → integer</pre>	<p>I.S. Data dari PenyakitList terdefinisi dan valid</p> <p>F.S. Menyimpan data PenyakitList ke file eksternal.</p>
	<pre>function save_obat(l: ^ObatList) → integer</pre>	Menyimpan data ObatList ke file eksternal.
	<pre>function save_obatpenyakit(l: ^ObatPenyakitList) → integer</pre>	Menyimpan data ObatPenyakitList ke file eksternal.
User	<pre>function IdxUser(name: string) → integer</pre>	<p>I.S. Nama terdefinisi dan valid</p> <p>F.S. Mengembalikan ID User yang terdapat pada user.csv</p> <p>Mengembalikan -1 jika user dengan nama "name" tidak ditemukan</p>
	<pre>procedure AddUser(input name:</pre>	<p>I.S. : File username.csv terbaca</p> <p>F.S. : Data user baru ditambahkan</p>

	string, input pass: string)	
	function ParseData(input line: string, input/output i: integer, input token: character) → string	<ul style="list-style-type: none"> - i adalah indeks awal data yang ingin di parse - token adalah char yang berfungsi sebagai MARK (untuk '\0' dan '\n' tidak perlu dibuat sebagai token) - line adalah string char di mana data berada Mengembalikan data yang sudah di parse
	function NamaUser(input idx: integer) → string	Prakondisi: idx adalah ID yang terdapat pada user.csv Mengembalikan Username yang terdapat pada user.csv
List	procedure CreateList(input/output l: List)	I.S. : l belum terdefinisi F.S. : l terdefinisi; l.nEff = 0
Linked list	function linked_create() → ^LinkedList	Membuat linked list kosong
	procedure linked_insertEnd(inp ut/output head: ^LinkedList, input data: integer)	I.S. : Linked list mungkin kosong F.S. : Data ditambahkan sebagai elemen terakhir
	procedure linked_print(input head: ^LinkedList)	I.S. : Linked list mungkin kosong atau berisi F.S. : Isi list ditampilkan ke layar

	<pre> procedure linked_free(input/output head: ^LinkedList) </pre>	<p>I.S. : Linked list berisi sejumlah node</p> <p>F.S. : Semua node didealokasi</p>
	<pre> function linked_isEmpty(head: ^LinkedList) → boolean </pre>	<p>Mengembalikan 1 jika stack kosong</p> <p>Mengembalikan 0 jika stack memiliki setidaknya satu elemen</p>
	<pre> procedure linked_delete(input/output head: ^LinkedList, input idx: integer) </pre>	<p>I.S. : Linked list memiliki elemen, dan idx valid</p> <p>F.S. : Elemen ke-idx dihapus dari list</p>
Node	<pre> function createNode(data: integer) → ^Node </pre>	Membuat node baru
Queue	<pre> function queue_create() → ^Queue </pre>	Membuat node baru yang berisikan data dari parameter
	<pre> procedure queue_push(input/output q: ^Queue, input data: integer) </pre>	<p>I.S. : Queue mungkin kosong</p> <p>F.S. : Data baru ditambahkan ke posisi akhir queue</p>
	<pre> function queue_front(q: ^Queue) → ^Node </pre>	<p>Mengembalikan pointer ke posisi terdepan pada queue</p> <p>Mengembalikan NULL jika queue kosong</p>

	<pre>procedure queue_pop(input/output q: ^Queue)</pre>	<p>I.S. : Queue tidak kosong</p> <p>F.S. : Elemen paling depan dihapus</p>
	<pre>procedure queue_clear(input/output q: ^Queue)</pre>	<p>I.S. : Queue mungkin berisi elemen</p> <p>F.S. : Queue kosong</p>
	<pre>function queue_isEmpty(q: Queue) → boolean</pre>	<p>Mengembalikan 1 jika queue kosong</p> <p>Mengembalikan 0 jika queue memiliki setidaknya satu elemen</p>
Stack	<pre>stack_create</pre>	Membuat node baru yang berisikan data dari parameter
	<pre>function stack_create() → ^Stack</pre>	Menambahkan node baru di stack
	<pre>function stack_top(s: ^Stack) → ^Node</pre>	<p>Mengembalikan pointer ke posisi teratas pada stack</p> <p>Mengembalikan NULL jika stack kosong</p>
	<pre>procedure stack_pop(input/output s: ^Stack)</pre>	<p>I.S. : Stack tidak kosong</p> <p>F.S. : Elemen paling atas dihapus</p>
	<pre>procedure stack_clears(input/output s: ^Stack)</pre>	<p>I.S. : Stack berisi elemen</p> <p>F.S. : Stack menjadi kosong</p>
	<pre>function stack_isEmpty(s: Stack) → boolean</pre>	<p>Mengembalikan 1 jika stack kosong</p> <p>Mengembalikan 0 jika stack memiliki setidaknya satu elemen</p>

	<pre> procedure stack_push(input/output s: ^Stack, input data: integer) </pre>	<p>I.S. : Stack mungkin kosong</p> <p>F.S. : Data ditambahkan di atas stack</p>
Map	<pre> procedure map_create(input/output t: Map, input key: integer) </pre>	<p>I.S. : t belum terdefinisi</p> <p>F.S. : Terbentuk map baru dengan key sesuai parameter</p>
	<pre> function map_findMap(input t: Map, input key: integer) → Map </pre>	<p>Mengembalikan pointer ke node map dengan key sesuai parameter.</p> <p>Jika tree kosong atau node dengan key yang sesuai tidak ditemukan, maka akan dikembalikan NULL.</p>
	<pre> procedure map_split(input t: Map, input key: integer, output l: Map, output r: Map) </pre>	<p>I.S. : t adalah node map yang valid, key adalah id yang valid</p> <p>F.S. : Terbentuk tree l yang berisi node dengan $key \leq$ parameter, dan tree r yang berisi node dengan $key >$ parameter</p>
	<pre> procedure map_insertNewMap(input/output t: Map, input it: Map) </pre>	<p>I.S. : t terinisialisasi (bisa NULL), it adalah node map yang valid</p> <p>F.S. : ADT map berisi node baru *it disisipkan sesuai prioritas dan key</p>
	<pre> procedure map_insertData(input/output t: Map, input idDokter: integer, input idPasien: integer) </pre>	<p>I.S. : t terinisialisasi, idDokter valid di dalam map</p> <p>F.S. : Data idPasien ditambahkan ke antrian pada node idDokter di akhir antrian</p>
	<pre> procedure map_popData(input t: </pre>	<p>I.S. : t terinisialisasi, antrian di node idDokter tidak kosong</p>

	Map, input idDokter: integer)	F.S. : Elemen pasien di posisi terdepan antrian node idDokter dikeluarkan
	function random_lcg(input n: integer, input seed: integer) → integer	Memberikan angka random berdasarkan n dan seed
Set	procedure set_create(input/output t: Set, input key: integer)	Membuat set baru dengan key sesuai parameter I.S. : t belum terdefinisi F.S. : Terbentuk set baru (tree) dengan root berkey = key
	function set_findSet(input t: Set, input key: integer) → Set	Mengembalikan pointer ke node set dengan key sesuai parameter. Jika tree kosong atau node dengan key yang sesuai tidak ditemukan, maka akan dikembalikan NULL.
	procedure set_split(input t: Set, input key: integer, output l: Set, output r: Set)	Melakukan split pada tree t, yang menghasilkan: - Tree l yang berisikan node dengan key ≤ key parameter - Tree r yang berisikan node dengan key > key parameter I.S. t adalah node yang valid, key adalah id yang valid F.S. Terbentuk tree l dan r sesuai kriteria diatas
	procedure set_insertNewSet(input/output t: Set, input it: Set)	Menyisipkan node set baru pada ADT set. I.S. Set terinisialisasi F.S. ADT Set berisi node baru, yakni node *it
	procedure set_insertData(input/output t: Set,	I.S. idDokter valid F.S. antrian idDokter mendapatkan data baru di akhir antrian

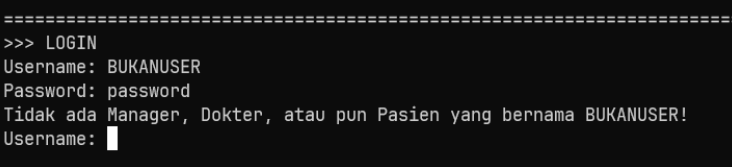
	input idDokter: integer, input idPasien: integer)	
	function random_lcg(input n: integer, input seed: integer) → integer	Memberikan angka random berdasarkan n dan seed
Matrix	procedure CreateMatrix(input rows: integer, input cols: integer, output M: Matrix)	I.S. : $1 \leq \text{rows} \leq \text{MAX_ROWS}$, $1 \leq \text{cols} \leq \text{MAX_COLS}$ F.S. : $M.\text{rows} = \text{rows}$; $M.\text{cols} = \text{cols}$;
Role	procedure AddPasienList(int id)	I.S. PasienList mungkin kosong atau berisi beberapa data pasien. F.S. Data pasien baru dengan id ditambahkan ke PasienList. neff bertambah 1.
	procedure AddPasien_to_Dokter(int idPasien,int idDokter)	I.S. Terdapat data pasien dan dokter di masing-masing list. F.S. Pasien dengan idPasien ditambahkan ke antrian dokter dengan idDokter. Queue dokter bertambah satu elemen.
	function DokterList_NametoID(char* name)	Mengembalikan ID dokter yang memiliki nama ruang kerja (ruangKerja) sesuai name. Jika tidak ditemukan, bisa mengembalikan -1.
	function UserID_to_DokterID(i nt id)	Mengembalikan idDokter dari pasien yang memiliki id pengguna = id. Jika tidak ditemukan, bisa mengembalikan -1.
	procedure AddManagerList(int id)	I.S. ManagerList berisi sejumlah data manager. F.S. Manager baru dengan id ditambahkan ke ManagerList. neff bertambah 1.

File-utilities	procedure path(dest: ^string, dir: string, name: string)	I.S. dir dan name merupakan string direktori dan nama file. F.S. dest berisi path lengkap dalam format: ../file/dir/name
	procedure add_user(Ulist: ^List, Uset: ^Set)	I.S. Ulist dan Uset berisi sejumlah user. F.S. Menambahkan user baru ke Ulist dan Uset.
Random	function random_lcg(n: integer, seed: long) → long	Menghasilkan bilangan acak dalam rentang 0..n-1 menggunakan metode Linear Congruential Generator (LCG) dengan seed sebagai nilai awal.

Tabel 6 - Spesifikasi Modul/Prosedur/Fungsi

11. Hasil Pengujian

11.1 F01 – Login

LOGIN	
KASUS 1	KETERANGAN
 <p><i>Gambar 3428 - Hasil Pengujian Login: Kasus 1</i></p>	<p>I: Username dan password valid</p> <p>O: Masuk ke menu</p>
KASUS 2	KETERANGAN
 <p><i>Gambar 29 35 - Hasil Pengujian Login: Kasus 2</i></p>	<p>I: Username tidak ditemukan</p> <p>O: Username diprompt ulang</p>

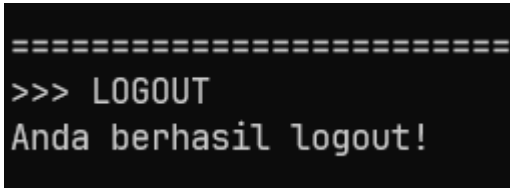

KASUS 3	KETERANGAN
 <p><i>Gambar 3630 - Hasil Pengujian Login: Kasus 3</i></p>	<p>I: Username valid, Password tidak valid O: Username diprompt ulang</p>
KASUS 4	KETERANGAN
 <p><i>Gambar 3731 - Hasil Pengujian Login: Kasus 4</i></p>	<p>I: Kondisi sudah login O: Kembali ke menu</p>

Tabel 7 - Hasil Pengujian F01

11.2 F02 - Register Pasien

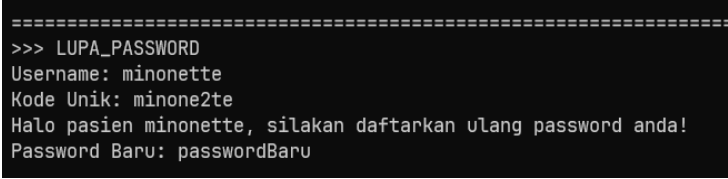
REGISTER	
KASUS 1	KETERANGAN
 <p><i>Gambar 3832 - Hasil Pengujian Register: Kasus 1</i></p>	<p>I: Username dan password valid belum ada di sistem</p> <p>O: Username dan password terdaftar ke dalam sistem sebagai pasien, otomatis melakukan login</p>
KASUS 2	KETERANGAN
 <p><i>Gambar 3933 - Hasil Pengujian Register: Kasus 2</i></p>	<p>I: Username telah ada di dalam sistem</p> <p>O: Pesan error, kembali ke menu</p>

11.3 F03 – Logout

LOGOUT	
KASUS 1	KETERANGAN
 <p><i>Gambar 35 41- Hasil Pengujian Logout: Kasus 1</i></p>	<p>I: Sudah login O: Berhasil Logout, Masuk ke menu awal</p>
KASUS 2	KETERANGAN
 <p><i>Gambar 4236 - Hasil Pengujian Logout: Kasus 2</i></p>	<p>I: Belum login O: Pesan error, kembali ke menu awal</p>

Tabel 9 - Hasil Pengujian F03

11.4 F04 - Lupa Password

LUPA_PASSWORD	
KASUS 1	KETERANGAN
 <p><i>Gambar 4337 - Hasil Pengujian LupaPassword: Kasus 1</i></p>	<p>I: Username dan password valid</p> <p>O: Prompt password baru, password baru terupdate, otomatis login</p>
KASUS 2	KETERANGAN

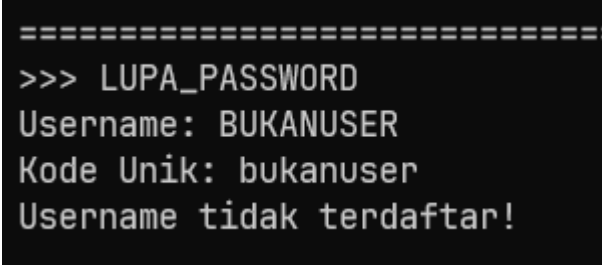
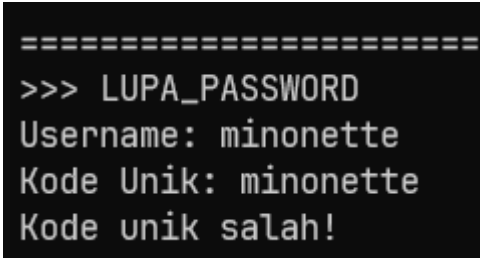

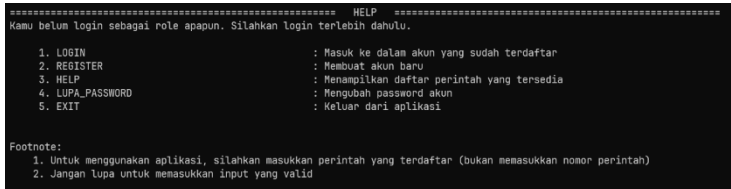
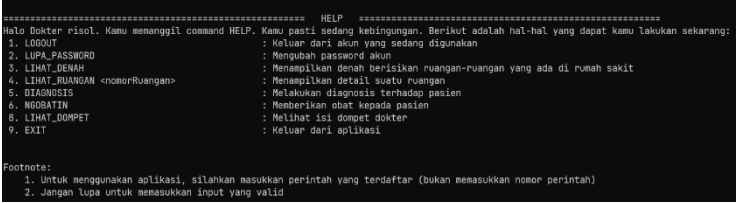
 <p><i>Gambar 4438 - Hasil Pengujian LupaPassword: Kasus 2</i></p>	<p>I: Username tidak ditemukan O: Pesan error, kembali ke menu</p>
KASUS 3	KETERANGAN
 <p><i>Gambar 39 45- Hasil Pengujian LupaPassword: Kasus 3</i></p>	<p>I: Username valid, kode unik tidak valid O: Pesan error, kembali ke menu</p>

Table 10 - Hasil Pengujian F04

11.5 F05 - Menu & Help

HELP	
KASUS 1	KETERANGAN
 <p><i>Gambar 4640 - Hasil Pengujian Help: Kasus 1</i></p>	<p>I: Program berhasil diinisiasi</p> <p>O: Menu untuk sebelum login dan setelah login</p>
KASUS 2	KETERANGAN
 <p><i>Gambar 41 47 - Hasil Pengujian Help: Kasus 2</i></p>	<p>I: Perintah HELP dengan kondisi belum login</p> <p>O: Menampilkan list perintah untuk pengguna yang belum login</p>

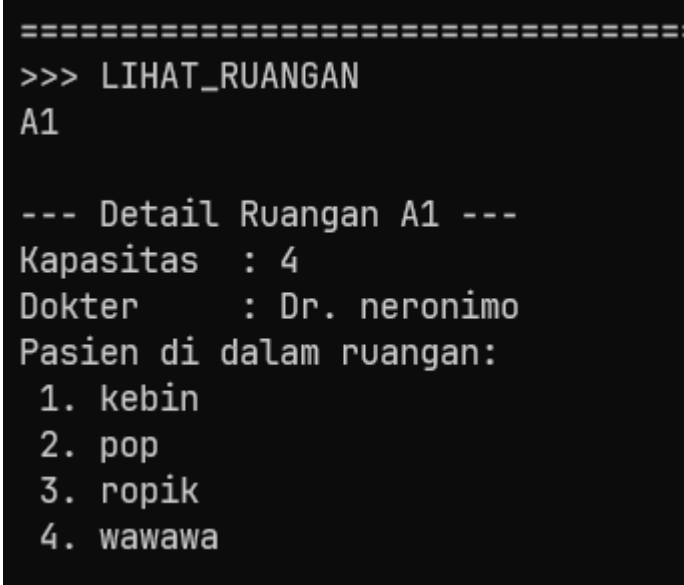
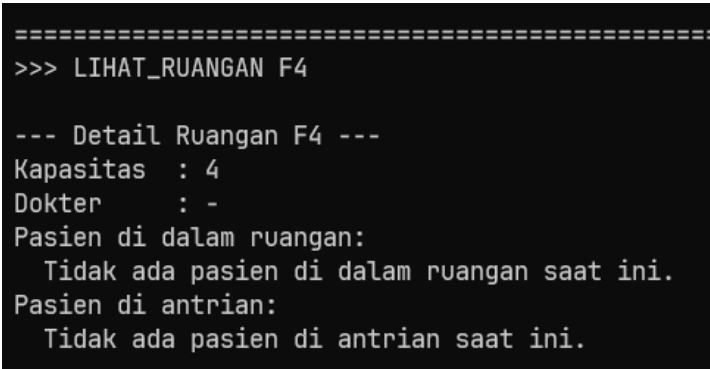
KASUS 3	KETERANGAN
<div><pre>***** HELP ***** Basis Manager perlu. Kenapa kamu memanggil command HELP? Kamu pasti sedang kebingungan. Berikut adalah hal-hal yang dapat kamu lakukan sekarang: 1. LOGOUT : Keluar dari akun yang sedang digunakan 2. LUPA_PASSWORD : Mengubah password akun 3. LIHAT_DEKAM : Menampilkan detail ruangan-ruangan yang ada di rumah sakit 4. LIHAT_RUMAHAN <nomorRuangan> : Menampilkan detail suatu ruangan 5. UMUM_DEDAH <daris> <tolup> : Mengubah tampilan rumah sakit 6. LIHAT_DEKAM : Melihat daftar user yang terdaftar 7. LIHAT_DOKTER : Melihat daftar dokter yang terdaftar 8. LIHAT_PASIR : Melihat daftar pasien yang terdaftar 9. CARI_DEKAM : Mencari user berdasarkan ID atau nama 10. CARI_DOKTER : Mencari dokter berdasarkan ID, nama, atau penyakit 11. LIHAT_DEKAM : Melihat semua antrian yang ada di rumah sakit 12. LIHAT_DEKAM : Melihat semua antrian yang ada di rumah sakit 13. PERUBAH_DOKTER : Menambahkan dokter baru ke dalam sistem 14. ASILAN_DOKTER : Menampilkan dokter ke ruangan tertentu 15. PERUBAH_DOKTER <ruangan> <ruangan> : Menambahkan dokter dari satu ruangan ke ruangan lain 16. LIHAT_PASIR : Melihat tampilan rumah sakit 17. EXIT : Keluar dari aplikasi Footnote: 1. Untuk menggunakan aplikasi, silakan masukkan perintah yang terdaftar (bukan memasukkan nomor perintah) 2. Jangan lupa untuk memasukkan input yang valid</pre></div> <p>Gambar 48 42 - Hasil Pengujian Help: Kasus 3</p>	<p>I: Perintah HELP dengan pengguna manager</p> <p>O: Menampilkan list perintah untuk manager</p>
KASUS 4	KETERANGAN
<div><pre>***** HELP ***** Selamat datang, pop. Kamu memanggil command HELP. Kamu pasti sedang kebingungan. Berikut adalah hal-hal yang dapat kamu lakukan sekarang: 1. LOGOUT : Keluar dari akun yang sedang digunakan 2. LUPA_PASSWORD : Mengubah password akun 3. LIHAT_DEKAM : Menampilkan detail ruangan-ruangan yang ada di rumah sakit 4. LIHAT_RUMAHAN <nomorRuangan> : Menampilkan detail suatu ruangan 5. PULANGDOOR : Mengecek status kepulangan pasien (Apakah sudah boleh pulang atau belum?) 6. DAFTAR_CHECKUP : Mendaftar diri untuk melakukan checkup 7. ANTRIAN : Melihat posisi antrian pasien saat ini 8. MINUM_OBAT : Minum obat yang harus diberikan oleh dokter 9. PENBAR : Minum obat untuk mengeluarkan obat terakir yang dininun dan memasukkannya ke inventory 10. SKIP_ANTRIAN : Melewati antrian dan langsung mendapatkan pelayanan 11. CANCEL_ANTRIAN : Membatalkan antrian yang sudah didaftarkan 12. BACA_GAMING : Melakukan game untuk mendapatkan banarich tambahan 13. LIHAT_DOKTER : Melihat isi dompet pasien 14. EXIT : Keluar dari aplikasi Footnote: 1. Untuk menggunakan aplikasi, silakan masukkan perintah yang terdaftar (bukan memasukkan nomor perintah) 2. Jangan lupa untuk memasukkan input yang valid</pre></div> <p>Gambar 49 43 - Hasil Pengujian Help: Kasus 4</p>	<p>I: Perintah HELP dengan pengguna pasien</p> <p>O: Menampilkan list perintah untuk pasien</p>

KASUS 5	KETERANGAN
 <p><i>Gambar 50 44 - Hasil Pengujian Help: Kasus 5</i></p>	<p>I: Perintah HELP dengan pengguna dokter</p> <p>O: Menampilkan list perintah untuk dokter</p>

Tabel 11- Hasil Pengujian F05

11.6 F06 (D01) - Denah Rumah Sakit

LIHAT_DENAH & LIHAT_RUANGAN <<nama ruangan>>	
KASUS 1	KETERANGAN
 <pre> ===== >>> LIHAT_DENAH 1 2 3 +-----+-----+-----+ A A1 A2 A3 +-----+-----+-----+ B B1 B2 B3 +-----+-----+-----+ </pre> <p><i>Gambar 45 51 - Hasil Pengujian LihatDenah</i></p>	<p>I: LIHAT_DENAH O: Denah ditampilkan</p>
KASUS 2	KETERANGAN
 <pre> ===== >>> LIHAT_RUANGAN A1 --- Detail Ruangan A1 --- Kapasitas : 4 Dokter : Dr. neronimo Pasien di dalam ruangan: 1. kebin 2. pop 3. ropik 4. wawawa </pre> <p><i>Gambar 5246 - Hasil Pengujian LihatRuangan: Kasus 1</i></p>	<p>I: LIHAT_RUANGAN dengan ruangan valid O: Status dan detail ruangan ditampilkan</p>

KASUS 3	KETERANGAN
 <p><i>Gambar 5347 - Hasil Pengujian LihatRuangan: Kasus 2</i></p>	<p>I: LIHAT_RUANGAN tidak memasukkan ruangan</p> <p>O: Menunggu input ruangan. Jika valid, status dan detail ruangan ditampilkan</p>
KASUS 4	KETERANGAN
 <p><i>Gambar 48 54 - Hasil Pengujian LihatRuangan: Kasus 3</i></p>	<p>I: LIHAT_RUANGAN dengan ruangan tidak terdapat dokter</p> <p>O: Menampilkan kapasitas ruangan normal berdasarkan file konfigurasi. Menampilkan status dan detail ruangan kosong.</p>

Tabel 12 - Hasil Pengujian F06

11.7 F07 - Lihat User

LIHAT_USER & LIHAT_PASIEN & LIHAT_DOKTER	
KASUS 1	KETERANGAN
<pre> ===== >>> LIHAT_USER Basis pengurutan: 1. ID 2. Nama >>> Pilihan: 1 Metode pengurutan: 1. ASC (A-Z) 2. DESC (Z-A) >>> Pilihan: 1 Menampilkan semua pengguna dengan ID terurut ascending... ----- ID Nama Role Penyakit ----- 01 pop pasien Diabetes Mellitus 02 nikeb pasien 03 kebin pasien 04 minionette pasien 05 opor pasien 06 zeru manager 07 stewart pasien 08 tuant pasien 09 gro pasien COVID-19 10 neronimo dokter 11 ciciko dokter 12 cacako dokter 13 krocket dokter 14 tobo pasien 15 risol dokter 16 ropik pasien 17 tobokan pasien 18 popokan pasien 19 pendatang pasien 20 wawawa pasien ----- </pre> <p><i>Gambar 5549 - Hasil Pengujian LihatUser: Kasus 1</i></p>	<p>I: LIHAT_USER, pilihan valid</p> <p>O: List ditampilkan dengan urutan sesuai keinginan user</p>
KASUS 2	KETERANGAN


```

=====
>>> LIHAT_USER
Basis pengurutan:
1. ID
2. Nama
>>> Pilihan: 3
Pilihan tidak valid
>>> Pilihan: 1

Metode pengurutan:
1. ASC (A-Z)
2. DESC (Z-A)
>>> Pilihan: 4
Pilihan tidak valid
>>> Pilihan: 2

Menampilkan semua pengguna dengan ID terurut descending...
-----
|ID|-----Nama-----|---Role---|----Penyakit---|
-----
| 20 | wawawa | pasien | |
| 19 | pendatang | pasien | |
| 18 | popokan | pasien | |
| 17 | tobokan | pasien | |
| 16 | ropik | pasien | |
| 15 | risol | dokter | |
| 14 | tobo | pasien | |
| 13 | kroket | dokter | |
| 12 | cacako | dokter | |
| 11 | ciciko | dokter | |
| 10 | neronimo | dokter | |
| 09 | gro | pasien | COVID-19
| 08 | tuart | pasien | |
| 07 | stewart | pasien | |
| 06 | zeru | manager | |
| 05 | opor | pasien | |
| 04 | minonette | pasien | |
| 03 | kebin | pasien | |
| 02 | nikeb | pasien | |
| 01 | pop | pasien | Diabetes Mellitus |
-----

```

Gambar 5650 - Hasil Pengujian LihatUser: Kasus 2

I: LIHAT_USER, pilihan tidak valid
O: Pilihan diminta ulang sampai valid. Jika sudah valid, list ditampilkan dengan urutan sesuai keinginan user

KASUS 3

KETERANGAN

```

=====
>>> LIHAT_PASIEN
Basis pengurutan:
1. ID
2. Nama
>>> Pilihan: 2

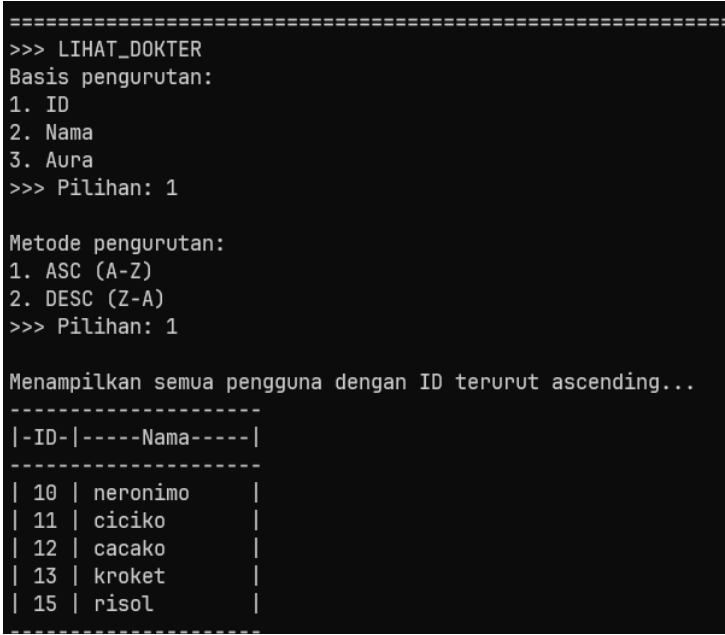
Metode pengurutan:
1. ASC (A-Z)
2. DESC (Z-A)
>>> Pilihan: 1

Menampilkan semua pengguna dengan Nama terurut ascending...
-----
|ID|-----Nama-----|----Penyakit---|
-----
| 09 | gro | COVID-19 |
| 03 | kebin | |
| 04 | minonette | |
| 02 | nikeb | |
| 05 | opor | |
| 19 | pendatang | |
| 01 | pop | Diabetes Mellitus |
| 18 | popokan | |
| 16 | ropik | |
| 07 | stewart | |
| 14 | tobo | |
| 17 | tobokan | |
| 08 | tuart | |
-----

```

Gambar 5751 - Hasil Pengujian LihatPasien

I: LIHAT_PASIEN, pilihan valid
O: List ditampilkan berisikan hanya pasien dengan urutan sesuai keinginan user

KASUS 4	KETERANGAN
 <pre> ===== >>> LIHAT_DOKTER Basis pengurutan: 1. ID 2. Nama 3. Aura >>> Pilihan: 1 Metode pengurutan: 1. ASC (A-Z) 2. DESC (Z-A) >>> Pilihan: 1 Menampilkan semua pengguna dengan ID terurut ascending... ----- -ID- -----Nama----- ----- 10 neronimo 11 ciciko 12 cacako 13 kroket 15 risol ----- </pre> <p><i>Gambar 5852 - Hasil Pengujian LihatDokter</i></p>	<p>I: LIHAT_DOKTER, pilihan valid</p> <p>O: List ditampilkan berisikan hanya dokter dengan urutan sesuai keinginan user</p>

Tabel 13 - Hasil Pengujian F07

11.8 F08 - Cari User

CARI_USER & CARI_PASIEN & CARI_DOKTER	
KASUS 1	KETERANGAN

<pre>===== >>> CARI_USER Basis Pencarian: 1. ID 2. Nama >>> Pilihan: 1 >>> Masukkan nomor ID user: 1 Menampilkan pengguna dengan nomor ID 1... ----- -ID- ----Nama---- ---Role--- ----Penyakit---- ----- 01 pop pasien Diabetes Mellitus -----</pre> <p><i>Gambar 5953 - Hasil Pengujian CariUser: Kasus 1</i></p>	<p>I: CARI_USER, pilihan valid</p> <p>O: Menampilkan list berisikan pengguna yang sesuai dengan kriteria yang diminta</p>
KASUS 2	KETERANGAN
<pre>===== >>> CARI_USER Basis Pencarian: 1. ID 2. Nama >>> Pilihan: 3 Pilihan tidak valid >>> Pilihan: 1 >>> Masukkan nomor ID user: 1 Menampilkan pengguna dengan nomor ID 1... ----- -ID- ----Nama---- ---Role--- ----Penyakit---- ----- 01 pop pasien Diabetes Mellitus -----</pre> <p><i>Gambar 6054 - Hasil Pengujian CariUser: Kasus 2</i></p>	<p>I: CARI_USER, pilihan tidak valid</p> <p>O: Pilihan diminta ulang sampai valid. Jika sudah valid, program menampilkan list berisikan pengguna yang sesuai dengan kriteria yang diminta</p>
KASUS 3	KETERANGAN

<pre> ===== >>> CARI_USER Basis Pencarian: 1. ID 2. Nama >>> Pilihan: 2 >>> Masukkan nama user: BUKANUSER Tidak ditemukan pengguna dengan nama BUKANUSER </pre> <p><i>Gambar 6155 - Hasil Pengujian CariUser: Kasus 3</i></p>	<p>I: CARI_USER, pengguna tidak ditemukan</p> <p>O: Menampilkan pesan</p>
KASUS 4	KETERANGAN
<pre> ===== >>> CARI_PASIEN Basis Pencarian: 1. ID 2. Nama 3. Penyakit >>> Pilihan: 3 >>> Masukkan nama penyakit: COVID-19 Basis pengurutan: 1. ID 2. Nama >>> Pilihan: 1 Metode pengurutan: 1. ASC (A-Z) 2. DESC (Z-A) >>> Pilihan: 1 Menampilkan pasien dengan penyakit COVID-19 dengan ID terurut ascending... ----- -ID- ----Nama----- ----Penyakit--- ----- 09 gro COVID-19 ----- </pre> <p><i>Gambar 6256 - Hasil Pengujian CariPasien</i></p>	<p>I: CARI_PASIEN, input valid, pencarian menggunakan penyakit</p> <p>O: List ditampilkan berisikan hanya pasien dengan urutan sesuai keinginan pengguna</p>
KASUS 5	KETERANGAN

```
=====
>>> CARI_DOKTER
Basis Pencarian:
1. ID
2. Nama
>>> Pilihan: neronimo
>>> Masukkan nama user:
-----
|-ID-|-----Nama-----|
-----
| 10 | neronimo          |
-----
```

Gambar 6357 - Hasil Pengujian CariDokter

I: CARI_DOKTER, input valid

O: List ditampilkan berisikan hanya dokter yang sesuai dengan kriteria yang diminta

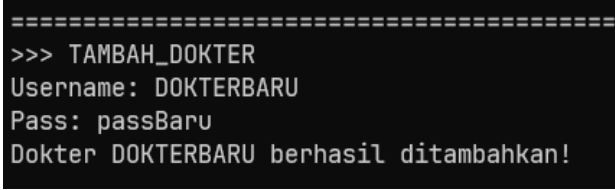
Tabel 14 - Hasil Pengujian F08


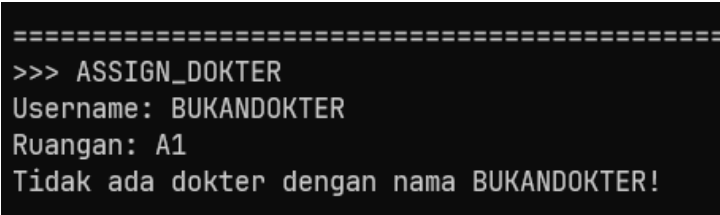
11.9 F09 (D02) - Lihat Antrian

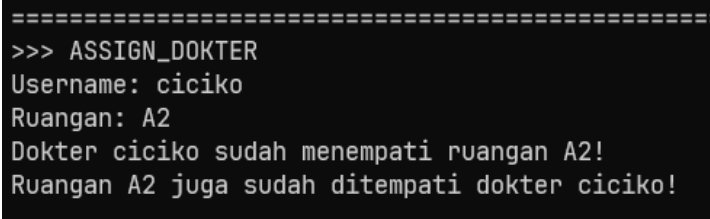
LIHAT_SEMUA_ANTRIAN	
KASUS 1	KETERANGAN
<pre> ===== >>> LIHAT_SEMUA_ANTRIAN 1 2 3 +-----+-----+ A A1 A2 A3 +-----+-----+ B B1 B2 B3 +-----+-----+ ===== A1 ===== Dokter : Dr. neronimo Pasien di dalam ruangan: 1. kebin 2. pop 3. ropik 4. wawawa Pasien di antrian: Tidak ada pasien di antrian saat ini. ===== A2 ===== Dokter : Dr. ciciko Pasien di dalam ruangan: 1. minonette 2. opor Pasien di antrian: Tidak ada pasien di antrian saat ini. ===== A3 ===== Dokter : Dr. cacako Pasien di dalam ruangan: 1. zeru Pasien di antrian: Tidak ada pasien di antrian saat ini. </pre> <p><i>Gambar 6458 - Hasil Pengujian LihatSemuaAntrian</i></p>	<p>I: LIHAT_SEMUA_ANTRIAN</p> <p>O: Menampilkan denah dan semua ruangan yang berisi dokter di dalamnya (tidak terdapat pasien tidak masalah)</p>

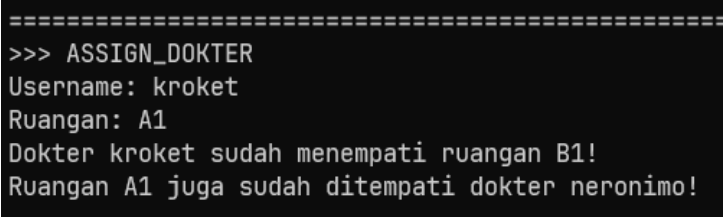
Tabel 15 - Hasil Pengujian F09

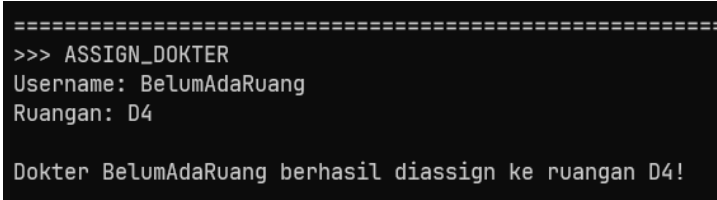
11.10 F10 - Tambah Dokter

TAMBAH_DOKTER & ASSIGN_DOKTER	
KASUS 1	KETERANGAN
 <p><i>Gambar 6559 - Hasil Pengujian TambahDokter: Kasus 1</i></p>	<p>I: TAMBAH_DOKTER, Dokter dengan nama tersebut belum ada</p> <p>O: Dokter ditambahkan di dalam sistem dan dapat melakukan login menggunakan username dan password yang telah didaftarkan</p>
KASUS 2	KETERANGAN

 <p><i>Gambar 6660 - Hasil Pengujian TambahDokter: Kasus 2</i></p>	<p>I: TAMBAH_DOKTER, Dokter dengan nama tersebut sudah ada O: Menampilkan pesan error</p>
KASUS 3	KETERANGAN
 <p><i>Gambar 61 67- Hasil Pengujian AssignDokter: Kasus 1</i></p>	<p>I: ASSIGN_DOKTER, dokter tidak terdaftar, ruangan valid O: Menampilkan pesan error</p>
KASUS 4	KETERANGAN

 <pre> ===== >>> ASSIGN_DOKTER Username: ciciko Ruangan: A2 Dokter ciciko sudah menempati ruangan A2! Ruangan A2 juga sudah ditempati dokter ciciko! </pre> <p><i>Gambar 6862 - Hasil Pengujian AssignDokter: Kasus 2</i></p>	<p>I: ASSIGN_DOKTER, dokter terdaftar, ruangan yang dimasukkan sudah diisi dokter tersebut</p> <p>O: Menampilkan pesan error</p>
KASUS 5	KETERANGAN
 <pre> ===== >>> ASSIGN_DOKTER Username: BelumAdaRuang Ruangan: A1 Dokter neronimo sudah menempati ruangan A1! Silakan cari ruangan lain untuk dokter BelumAdaRuang. </pre> <p><i>Gambar 6963 - Hasil Pengujian AssignDokter: Kasus 3</i></p>	<p>I: ASSIGN_DOKTER, dokter terdaftar, ruangan yang dimasukkan sudah diisi oleh dokter lain</p> <p>O: Menampilkan pesan error</p>
KASUS 6	KETERANGAN

 <pre>===== >>> ASSIGN_DOKTER Username: kroket Ruangan: A1 Dokter kroket sudah menempati ruangan B1! Ruangan A1 juga sudah ditempati dokter neronimo!</pre> <p><i>Gambar 7064 - Hasil Pengujian AssignDokter: Kasus 4</i></p>	<p>I: ASSIGN_DOKTER, input valid, dokter sudah memenuhi suatu ruangan, ruangan sudah diisi oleh dokter yang ingin diassign</p> <p>O: Menampilkan pesan error</p>
<p>KASUS 7</p>	<p>KETERANGAN</p>

 <p><i>Gambar 7165 - Hasil Pengujian AssignDokter: Kasus 5</i></p>	<p>I: ASSIGN_DOKTER, input valid, dokter belum memenuhi suatu ruangan, ruangan masih kosong</p> <p>O: Dokter berhasil ditugaskan di ruangan tersebut</p>
--	--

Tabel 16 - Hasil Pengujian F10

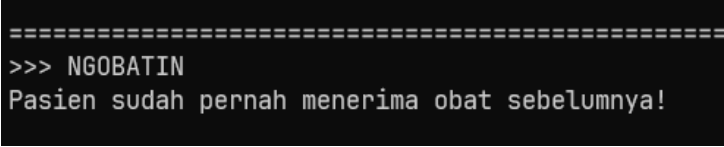
11.11 F11 – Diagnosis

DIAGNOSIS	
KASUS 1	KETERANGAN
<pre>===== >>> DIAGNOSIS minonette terdiagnosa penyakit Diabetes Mellitus!</pre> <p><i>Gambar 7266 - Hasil Pengujian Diagnosis: Kasus 1</i></p>	<p>I: Pasien belum pernah didiagnosis</p> <p>O: Hasil diagnosa pasien ditampilkan</p>
KASUS 2	KETERANGAN
<pre>===== >>> DIAGNOSIS Pasien sudah pernah didiagnosis penyakit Diabetes Mellitus! Silakan gunakan perintah 'NGOBATIN' untuk memberikan obat kepada pasien.</pre> <p><i>Gambar 67 73- Hasil Pengujian Diagnosis: Kasus 2</i></p>	<p>I: Pasien sudah pernah didiagnosis</p> <p>O: Menampilkan pesan error</p>

Tabel 17 - Hasil Pengujian F11

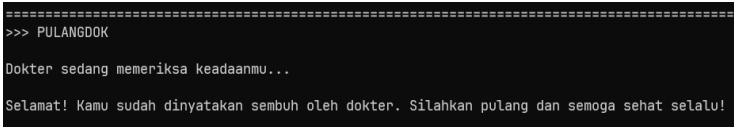
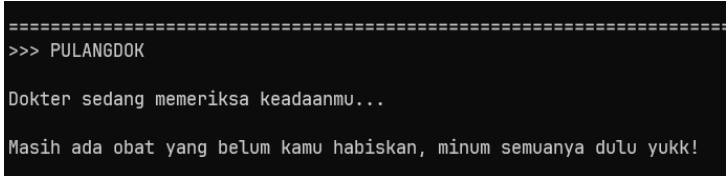
11.12 F12 – Ngobatin

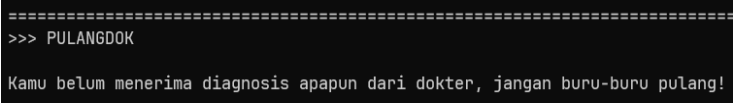
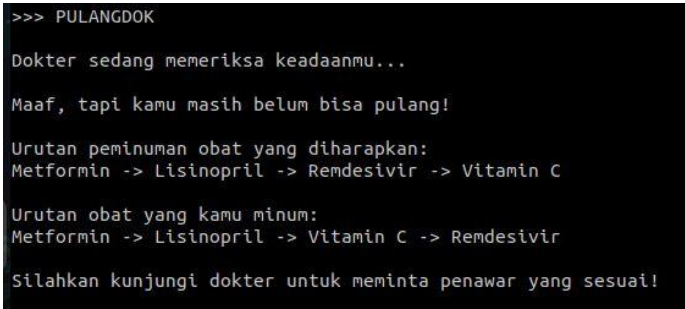
NGOBATIN	
KASUS 1	KETERANGAN
 <p>===== >>> NGOBATIN Pasien memiliki penyakit COVID-19 Obat yang harus diberikan: 1. Remdesivir</p> <p><i>Gambar 7468 - Hasil Pengujian Ngobatin: Kasus 1</i></p>	<p>I: Pasien belum pernah diobati O: Obat yang diberikan ke pasien ditampilkan</p>
KASUS 2	KETERANGAN
 <p>===== >>> NGOBATIN Pasien belum didiagnosis! Lakukan DIAGNOSIS terlebih dahulu</p> <p><i>Gambar 7569- Hasil Pengujian Ngobatin: Kasus 2</i></p>	<p>I: Pasien belum didiagnosis O: Menampilkan pesan error</p>

KASUS 3	KETERANGAN
 <p><i>Gambar 7670 - Hasil Pengujian Ngobatin: Kasus 3</i></p>	<p>I: Pasien sudah pernah diobati</p> <p>O: Menampilkan pesan error</p>

Tabel 18 - Hasil Pengujian F12

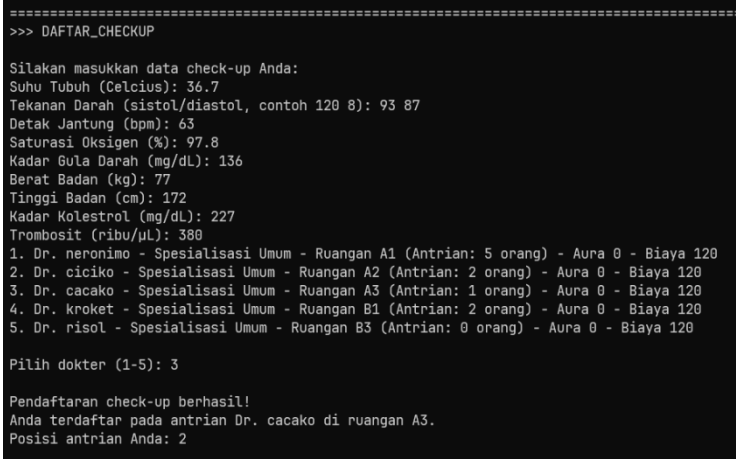
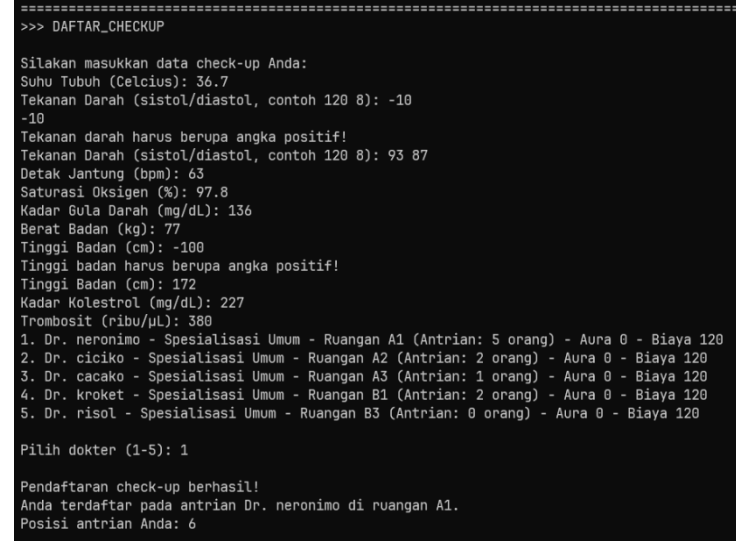
11.13 F13 - Aku boleh pulang ga, dok?

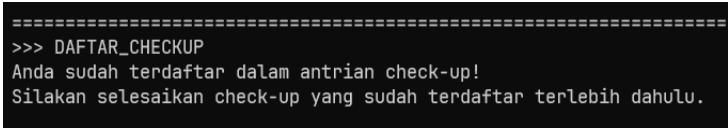
PULANGDOK	
KASUS 1	KETERANGAN
 <p><i>Gambar 71 77- Hasil Pengujian PulangDok: Kasus 1</i></p>	<p>I: Pasien sudah minum obat, penyakit sudah hilang O: Riwayat penyakit pasien hilang</p>
KASUS 2	KETERANGAN
 <p><i>Gambar 7872 - Hasil Pengujian PulangDok: Kasus 2</i></p>	<p>I: Pasien belum minum obat, penyakit masih ada O: Menampilkan pesan error</p>

KASUS 3	KETERANGAN
 <p><i>Gambar 7973 - Hasil Pengujian PulangDok: Kasus 3</i></p>	<p>I: Pasien belum didiagnosis O: Menampilkan pesan error</p>
KASUS 4	KETERANGAN
 <p><i>Gambar 80 74 - Hasil Pengujian PulangDok: Kasus 4</i></p>	<p>I: Pasien meminum obat dengan tidak sesuai O: Menampilkan pesan error</p>

Tabel 19 - Hasil Pengujian F13

11.14 F14 - Daftar Check-Up

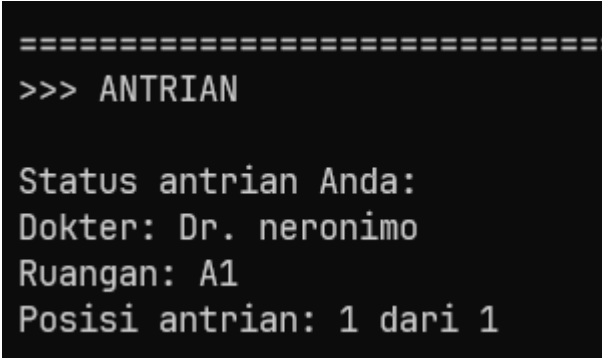
DAFTAR_CHECKUP	
KASUS 1	KETERANGAN
 <pre> ===== >>> DAFTAR_CHECKUP Silakan masukkan data check-up Anda: Suhu Tubuh (Celcius): 36.7 Tekanan Darah (sistol/diastol, contoh 120 8): 93 87 Detak Jantung (bpm): 63 Saturasi Oksigen (%): 97.8 Kadar Gula Darah (mg/dL): 136 Berat Badan (kg): 77 Tinggi Badan (cm): 172 Kadar Kolesterol (mg/dL): 227 Trombosit (ribu/µL): 388 1. Dr. neronimo - Spesialisasi Umum - Ruangan A1 (Antrian: 5 orang) - Aura 0 - Biaya 120 2. Dr. ciciko - Spesialisasi Umum - Ruangan A2 (Antrian: 2 orang) - Aura 0 - Biaya 120 3. Dr. cacako - Spesialisasi Umum - Ruangan A3 (Antrian: 1 orang) - Aura 0 - Biaya 120 4. Dr. kroket - Spesialisasi Umum - Ruangan B1 (Antrian: 2 orang) - Aura 0 - Biaya 120 5. Dr. risol - Spesialisasi Umum - Ruangan B3 (Antrian: 0 orang) - Aura 0 - Biaya 120 Pilih dokter (1-5): 3 Pendaftaran check-up berhasil! Anda terdaftar pada antrian Dr. cacako di ruangan A3. Posisi antrian Anda: 2 </pre> <p><i>Gambar 8175 - Hasil Pengujian DaftarCheckup: Kasus 1</i></p>	<p>I: Pasien belum terdaftar checkup</p> <p>O: Program meminta data medis dasar pasien, Pasien memilih dokter, mendaftarkan dalam antrian dokter</p>
KASUS 2	KETERANGAN
 <pre> ===== >>> DAFTAR_CHECKUP Silakan masukkan data check-up Anda: Suhu Tubuh (Celcius): 36.7 Tekanan Darah (sistol/diastol, contoh 120 8): -10 -10 Tekanan darah harus berupa angka positif! Tekanan Darah (sistol/diastol, contoh 120 8): 93 87 Detak Jantung (bpm): 63 Saturasi Oksigen (%): 97.8 Kadar Gula Darah (mg/dL): 136 Berat Badan (kg): 77 Tinggi Badan (cm): -100 Tinggi badan harus berupa angka positif! Tinggi Badan (cm): 172 Kadar Kolesterol (mg/dL): 227 Trombosit (ribu/µL): 388 1. Dr. neronimo - Spesialisasi Umum - Ruangan A1 (Antrian: 5 orang) - Aura 0 - Biaya 120 2. Dr. ciciko - Spesialisasi Umum - Ruangan A2 (Antrian: 2 orang) - Aura 0 - Biaya 120 3. Dr. cacako - Spesialisasi Umum - Ruangan A3 (Antrian: 1 orang) - Aura 0 - Biaya 120 4. Dr. kroket - Spesialisasi Umum - Ruangan B1 (Antrian: 2 orang) - Aura 0 - Biaya 120 5. Dr. risol - Spesialisasi Umum - Ruangan B3 (Antrian: 0 orang) - Aura 0 - Biaya 120 Pilih dokter (1-5): 1 Pendaftaran check-up berhasil! Anda terdaftar pada antrian Dr. neronimo di ruangan A1. Posisi antrian Anda: 6 </pre> <p><i>Gambar 76 82- Hasil Pengujian DaftarCheckup: Kasus 2</i></p>	<p>I: Pasien belum terdaftar checkup, input data medis tidak valid (negatif)</p> <p>O: Menampilkan pesan error, mengulang prompt untuk data medis</p>

KASUS 3	KETERANGAN
 <p><i>Gambar 8377 - Hasil Pengujian DaftarCheckup: Kasus 3</i></p>	<p>I: Pasien sudah terdaftar checkup</p> <p>O: Menampilkan pesan error</p>

Tabel 20 - Hasil Pengujian F14


11.15 F15 - Antrian Saya!

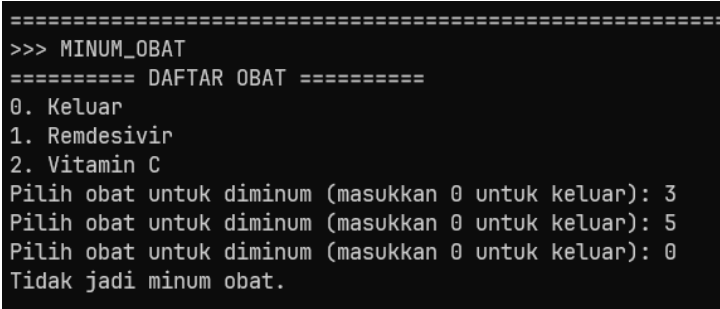
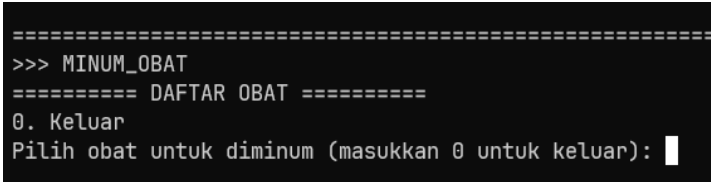
ANTRIAN	
KASUS 1	KETERANGAN
 <pre>===== >>> ANTRIAN Anda sedang berada di dalam ruangan dokter!</pre> <p><i>Gambar 8478 - Hasil Pengujian Antrian: Kasus 1</i></p>	<p>I: Pasien tidak berada dalam antrian, sudah masuk ruangan dokter</p> <p>O: Menampilkan pesan error</p>
KASUS 2	KETERANGAN
 <pre>===== >>> ANTRIAN Anda belum terdaftar dalam antrian check-up! Silakan daftar terlebih dahulu dengan command DAFTAR_CHECKUP.</pre> <p><i>Gambar 8579 - Hasil Pengujian Antrian: Kasus 2</i></p>	<p>I: Pasien belum terdaftar checkup, tidak berada dalam antrian</p> <p>O: Menampilkan pesan error</p>

KASUS 3	KETERANGAN
 <p><i>Gambar 8680 - Hasil Pengujian Antrian: Kasus 3</i></p>	<p>I: Pasien sudah terdaftar checkup</p> <p>O: Menampilkan pesan error</p>

Tabel 21 - Hasil Pengujian F15

11.16 F16 - Minum Obat

ANTRIAN	
KASUS 1	KETERANGAN
 <p><i>Gambar 8781 - Hasil Pengujian MinumObat: Kasus 1</i></p>	<p>I: Pasien sudah di-NGOBATIN oleh dokter, sudah memiliki obat di dalam inventorynya, input valid</p> <p>O: Obat hilang dari inventory dan pindah ke perut</p>
KASUS 2	KETERANGAN

 <p><i>Gambar 8882 - Hasil Pengujian MinumObat: Kasus 2</i></p>	<p>I: Pasien sudah di-NGOBATIN oleh dokter, sudah memiliki obat di dalam inventorynya, input tidak valid</p> <p>O: Input akan terus diprompt hingga valid, jika input = 0, maka kembali ke menu</p>
KASUS 3	KETERANGAN
 <p><i>Gambar 8983 - Hasil Pengujian MinumObat: Kasus 3</i></p>	<p>I: Jika pasien belum di-NGOBATIN, maka inventory kosong</p> <p>O: Memunculkan hanya pilihan untuk keluar dari menu MINUM_OBAT</p>

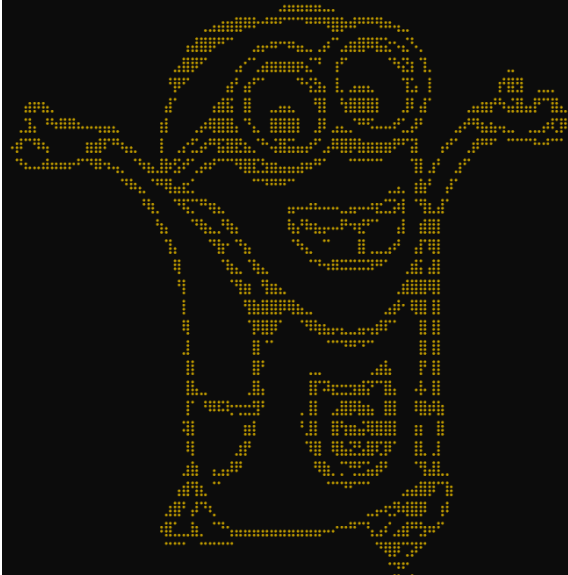
Tabel 22 - Hasil Pengujian F16

11.17 F17 - Minum Penawar

PENAWAR	
KASUS 1	KETERANGAN
 <p><i>Gambar 9084 - Hasil Pengujian MinumPenawar: Kasus 1</i></p>	<p>I: Pasien sudah meminum obat</p> <p>O: Obat keluar dari perut dan masuk ke dalam stack inventory</p>
KASUS 2	KETERANGAN
 <p><i>Gambar 85 91- Hasil Pengujian MinumPenawar: Kasus 2</i></p>	<p>I: Pasien belum meminum obat (perut kosong)</p> <p>O: Menampilkan pesan error, kembali ke menu</p>

Tabel 23 - Hasil Pengujian F17

11.18 F18 – Exit & D04 – Save

EXIT & SAVE	
KASUS 1	KETERANGAN
<pre>>>> EXIT Apakah Anda yakin ingin keluar? (y/n): y Apakah anda mau melakukan penyimpanan file yang sudah diubah? (y/n): n Data tidak disimpan.</pre>  <pre>Sampai jumpa lagi di aplikasi Rumah Sakit Nimons! Terima kasih telah menggunakan layanan kami.</pre> <p><i>Gambar 9286 - Hasil Pengujian Exit: Kasus 1</i></p>	<p>I: Input valid O: Program berakhir</p>
KASUS 2	KETERANGAN


```

=====
>>> EXIT
Apakah Anda yakin ingin keluar? (y/n): a
Masukan tidak valid. Silakan coba lagi.
Apakah Anda yakin ingin keluar? (y/n): y
Apakah anda mau melakukan penyimpanan file yang sudah diubah? (y/n): a
Masukan tidak valid. Silakan coba lagi.
Apakah anda mau melakukan penyimpanan file yang sudah diubah? (y/n): n
Data tidak disimpan.



Sampai jumpa lagi di aplikasi Rumah Sakit Nimons!
Terima kasih telah menggunakan layanan kami.

```

Gambar 9387 - Hasil Pengujian Exit: Kasus 2

I: Input tidak valid
O: Pdsesan error, input kembali diminta hingga tepat.

I: File tidak ingin disimpan
O: File tidak disimpan, keluar dari program

KASUS 3

KETERANGAN

```

=====
>>> EXIT
Apakah Anda yakin ingin keluar? (y/n): y
Apakah anda mau melakukan penyimpanan file yang sudah diubah? (y/n): y

Masukkan nama folder: ray

Saving...
Berhasil menyimpan data di folder data/ray!

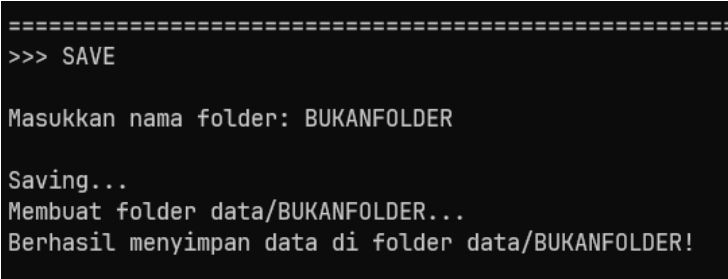


Sampai jumpa lagi di aplikasi Rumah Sakit Nimons!
Terima kasih telah menggunakan layanan kami.

```


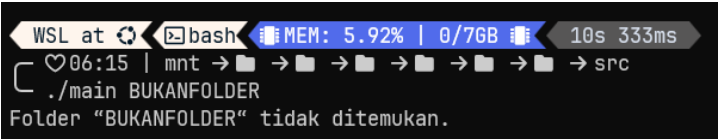
Gambar 9488 - Hasil Pengujian Exit: Kasus 3

I: File ingin disimpan
O: Folder diprompt. File disimpan ke folder yang diinput

KASUS 4	KETERANGAN
 <p style="text-align: center;"><i>Gambar 9589 - Hasil Pengujian Save</i></p>	<p>I: Langsung meminta prompt SAVE</p> <p>O: Folder diprompt. Tidak ada folder dengan nama tersebut. Membuat folder baru. Output sama dengan Kasus 1 sampai Kasus 4</p>

Tabel 24 - Hasil Pengujian F18 & D04

11.19D03 – Load

-\$./main <<nama folder>>	
KASUS 1	KETERANGAN
 <p><i>Gambar 9690 - Hasil Pengujian Load : Kasus 1</i></p>	<p>I: Folder ditemukan O: Program dimulai</p>
KASUS 2	KETERANGAN
 <p><i>Gambar 9791 - Hasil Pengujian Load : Kasus 2</i></p>	<p>I: Folder tidak ditemukan O: Pesan error, program tidak dimulai</p>

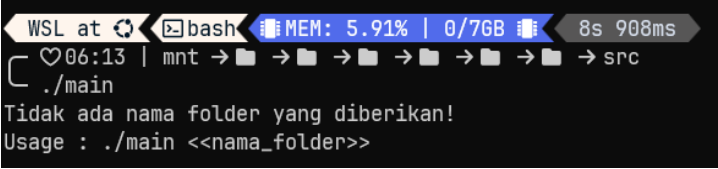
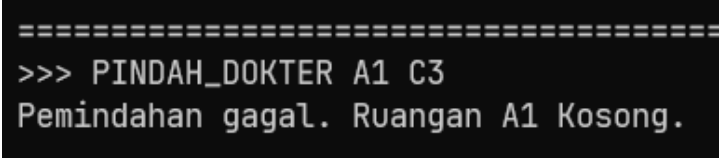
KASUS 3	KETERANGAN
<div><p><i>Gambar 9892 - Hasil Pengujian Load : Kasus 3</i></p></div>	<p>I: Tidak memasukkan folder</p> <p>O: Pesan error, program tidak dimulai</p>

Table 25 - Hasil Pengujian D03

11.20 B02 – Denah Dinamis

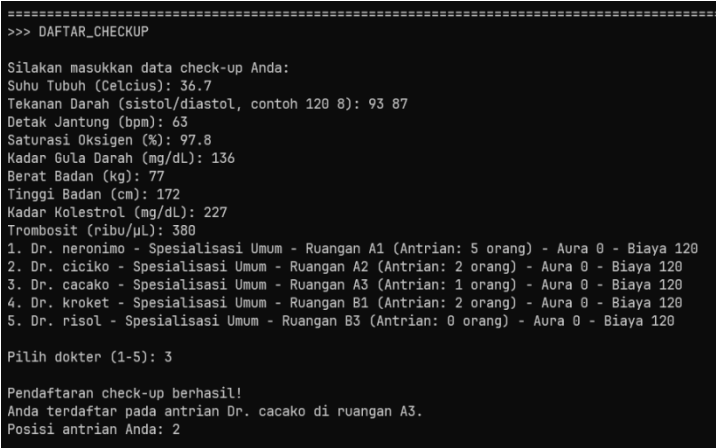
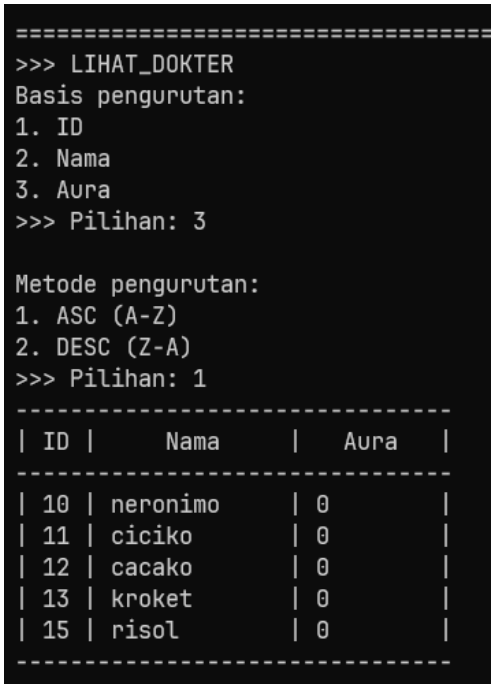
UBAH_DENAH <r> <c> & PINDAH_DOKTER <rAwal> <rAkhir>	
KASUS 1	KETERANGAN
 <p><i>Gambar 9993 - Hasil Pengujian UbahDenah: Kasus 1</i></p>	<p>I: Denah terdefinisi, input UBAH_DENAH valid, tidak ada dokter yang menghalangi</p> <p>O: Denah berubah sesuai spesifikasi baru</p>
KASUS 2	KETERANGAN
 <p><i>Gambar 10094 - Hasil Pengujian UbahDenah: Kasus 2</i></p>	<p>I: Denah terdefinisi, input UBAH_DENAH valid, ada dokter yang menghalangi perubahan denah</p> <p>O: Pesan error, denah tidak berubah</p>

KASUS 3	KETERANGAN
<pre data-bbox="296 517 1023 611">===== >>> PINDAH_DOKTER A1 C1 Dr. neronimo berhasil dipindahkan dari ruangan A1 ke ruangan C1.</pre> <p data-bbox="379 633 938 663"><i>Gambar 10195 - Hasil Pengujian PindahDokter: Kasus 1</i></p>	<p data-bbox="1082 573 1393 663">I: Dokter, ruangan awal, dan ruangan akhir valid</p> <p data-bbox="1118 685 1356 775">O: Dokter berhasil dipindahkan</p>
KASUS 4	KETERANGAN
<pre data-bbox="309 1411 1023 1536">===== >>> PINDAH_DOKTER C1 A2 Pemindahan gagal. Ruangan A2 Sudah ditempati.</pre> <p data-bbox="383 1559 948 1588"><i>Gambar 96 102 - Hasil Pengujian PindahDokter: Kasus 2</i></p>	<p data-bbox="1062 1447 1414 1592">I: Dokter dan ruangan awal valid, ruangan akhir sudah ditempati</p> <p data-bbox="1086 1615 1390 1704">O: Menampilkan pesan error, kembali ke menu</p>

KASUS 5	KETERANGAN
 <p><i>Gambar 10397 - Hasil Pengujian PindahDokter: Kasus 3</i></p>	<p>I: Dokter dan ruangan akhir valid, ruangan awal tidak valid</p> <p>O: Menampilkan pesan error, kembali ke menu</p>

Tabel 26 - Hasil Pengujian B02

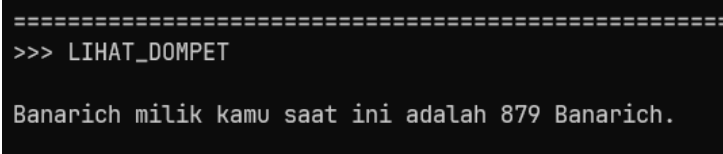
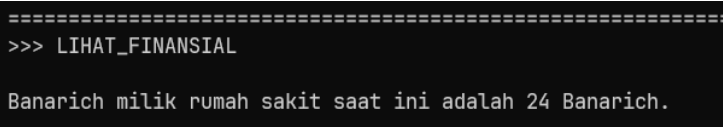
11.21 B03 – Aura!

KASUS 1	KETERANGAN
 <p>=====</p> <pre>>>> DAFTAR_CHECKUP Silakan masukkan data check-up Anda: Suhu Tubuh (Celcius): 36.7 Tekanan Darah (sistol/diastol, contoh 120 8): 93 87 Detak Jantung (bpm): 63 Saturasi Oksigen (%): 97.8 Kadar Gula Darah (mg/dL): 136 Berat Badan (kg): 77 Tinggi Badan (cm): 172 Kadar Kolesterol (mg/dL): 227 Trombosit (ribu/µL): 380 1. Dr. neronimo - Spesialisasi Umum - Ruangan A1 (Antrian: 5 orang) - Aura 0 - Biaya 120 2. Dr. ciciko - Spesialisasi Umum - Ruangan A2 (Antrian: 2 orang) - Aura 0 - Biaya 120 3. Dr. cacako - Spesialisasi Umum - Ruangan A3 (Antrian: 1 orang) - Aura 0 - Biaya 120 4. Dr. kroket - Spesialisasi Umum - Ruangan B1 (Antrian: 2 orang) - Aura 0 - Biaya 120 5. Dr. risol - Spesialisasi Umum - Ruangan B3 (Antrian: 0 orang) - Aura 0 - Biaya 120 Pilih dokter (1-5): 3 Pendaftaran check-up berhasil! Anda terdaftar pada antrian Dr. cacako di ruangan A3. Posisi antrian Anda: 2</pre> <p><i>Gambar 10498 - Hasil Pengujian Aura: Kasus 1 (DaftarCheckup)</i></p>	<p>I: DAFTAR_CHECKUP</p> <p>O: Memunculkan atribut aura dari dokter</p>
KASUS 2	KETERANGAN
 <pre>===== >>> LIHAT_DOKTER Basis pengurutan: 1. ID 2. Nama 3. Aura >>> Pilihan: 3 Metode pengurutan: 1. ASC (A-Z) 2. DESC (Z-A) >>> Pilihan: 1 ----- ID Nama Aura ----- 10 neronimo 0 11 ciciko 0 12 cacako 0 13 kroket 0 15 risol 0 -----</pre> <p><i>Gambar 105 99 - Hasil Pengujian Aura: Kasus 2 (LihatDokter)</i></p>	<p>I: LIHAT_DOKTER, Memilih basis pengurutan Aura</p> <p>O: Memunculkan list dokter diurutkan berdasarkan Aura</p>

Tabel 27 - Hasil Pengujian B03

11.22 B04 – Banarich!!!!

DAFTAR_CHECKUP & LIHAT_DOMPET & LIHAT_FINANSIAL & GACHA_GAMING	
KASUS 1	KETERANGAN
<pre> ===== >>> DAFTAR_CHECKUP Silakan masukkan data check-up Anda: Suhu Tubuh (Celcius): 36.7 Tekanan Darah (sistol/diastol, contoh 120 8): 93 87 Detak Jantung (bpm): 63 Saturasi Oksigen (%): 97.8 Kadar Gula Darah (mg/dL): 136 Berat Badan (kg): 77 Tinggi Badan (cm): 172 Kadar Kolesterol (mg/dL): 227 Trombosit (ribu/μL): 380 1. Dr. neronimo - Spesialisasi Umum - Ruangan A1 (Antrian: 5 orang) - Aura 0 - Biaya 120 2. Dr. ciciko - Spesialisasi Umum - Ruangan A2 (Antrian: 2 orang) - Aura 0 - Biaya 120 3. Dr. cacako - Spesialisasi Umum - Ruangan A3 (Antrian: 1 orang) - Aura 0 - Biaya 120 4. Dr. kroket - Spesialisasi Umum - Ruangan B1 (Antrian: 2 orang) - Aura 0 - Biaya 120 5. Dr. risol - Spesialisasi Umum - Ruangan B3 (Antrian: 0 orang) - Aura 0 - Biaya 120 Pilih dokter (1-5): 3 Pendaftaran check-up berhasil! Anda terdaftar pada antrian Dr. cacako di ruangan A3. Posisi antrian Anda: 2 </pre> <p><i>Gambar 100 106- Hasil Pengujian Banarich: Kasus 1 (DaftarCheckup)</i></p>	<p>I: DAFTAR_CHECKUP, Banarich mencukupi untuk suatu dokter</p> <p>O: Banarich berkurang, Pasien masuk antrian</p>
KASUS 2	KETERANGAN
<pre> ===== >>> DAFTAR_CHECKUP Silakan masukkan data check-up Anda: Suhu Tubuh (Celcius): 36.7 Tekanan Darah (sistol/diastol, contoh 120 8): -5 -10 Tekanan darah harus berupa angka positif! Tekanan Darah (sistol/diastol, contoh 120 8): 93 87 Detak Jantung (bpm): 63 Saturasi Oksigen (%): 97.8 Kadar Gula Darah (mg/dL): 136 Berat Badan (kg): 77 Tinggi Badan (cm): 1722 Kadar Kolesterol (mg/dL): 227 Trombosit (ribu/μL): 380 1. Dr. neronimo - Spesialisasi Umum - Ruangan A1 (Antrian: 5 orang) - Aura 0 - Biaya 120 2. Dr. ciciko - Spesialisasi Umum - Ruangan A2 (Antrian: 2 orang) - Aura 0 - Biaya 120 3. Dr. cacako - Spesialisasi Umum - Ruangan A3 (Antrian: 1 orang) - Aura 0 - Biaya 120 4. Dr. kroket - Spesialisasi Umum - Ruangan B1 (Antrian: 2 orang) - Aura 0 - Biaya 120 5. Dr. risol - Spesialisasi Umum - Ruangan B3 (Antrian: 0 orang) - Aura 0 - Biaya 120 Pilih dokter (1-5): 1 Maaf Banarich kamu tidak cukup! Silakan pilih dokter lain atau mainkan mesin gacha! </pre> <p><i>Gambar 107101 - Hasil Pengujian Banarich: Kasus 2 (DaftarCheckup)</i></p>	<p>I: DAFTAR_CHECKUP, Banarich tidak mencukupi untuk suatu dokter</p> <p>O: Banarich tidak berkurang, pasien tidak masuk antrian</p>

KASUS 3	KETERANGAN
 <p><i>Gambar 108102 - Hasil Pengujian Banarich: Kasus 3 (LihatDompel)</i></p>	<p>I: LIHAT_DOMPET O: Menunjukkan Banarich yang dimiliki pengguna (pasien atau dokter)</p>
KASUS 4	KETERANGAN
 <p><i>Gambar 109103 - Hasil Pengujian Banarich: Kasus 4 (LihatFinansial)</i></p>	<p>I: LIHAT_FINANSIAL O: Menunjukkan Banarich yang dimiliki rumah sakit</p>

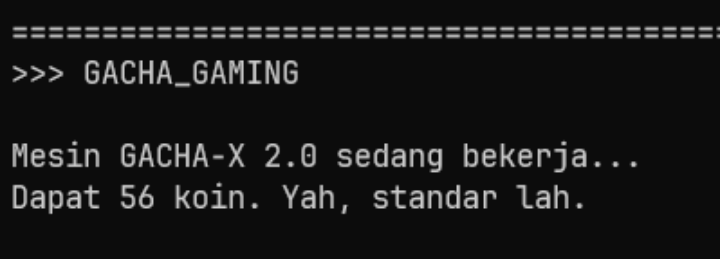
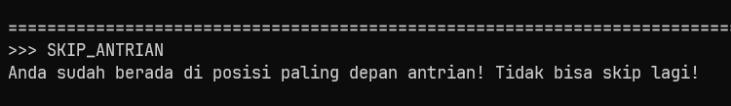
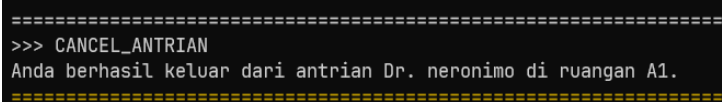
KASUS 5	KETERANGAN
 <p><i>Gambar 110104 - Hasil Pengujian Banarich: Kasus 5 (Gacha)</i></p>	<p>I: GACHA_GAMING</p> <p>O: Menunjukkan hasil gacha banarich. Banarich berkurang atau bertambah sesuai hasil gacha.</p>

Table 28 - Hasil Pengujian B04

11.23 B06 - Mainin Antrian



SKIP_ANTRIAN & CANCEL_ANTRIAN	
KASUS 1	KETERANGAN
<div><div>>>> SKIP_ANTRIAN Skip antrian gagal! Anda tidak sedang terdaftar dalam antrian manapun!</div><div>Gambar 111105 - Hasil Pengujian SkipAntrian: Kasus 1</div></div>	I: Pasien belum terdaftar antrian O: Menampilkan pesan error
KASUS 2	KETERANGAN
<div><div>Pendaftaran check-up berhasil! Anda terdaftar pada antrian Dr. neronimo di ruangan A1. Posisi antrian Anda: 6 =====</div><div>RUMAH SAKIT NIMONS!</div><div>Selamat datang di aplikasi Rumah Sakit Nimons! Silahkan masukkan perintah yang diinginkan. Ketik 'HELP' untuk melihat daftar perintah yang tersedia.</div><div>>>> SKIP_ANTRIAN Anda berhasil maju ke depan antrian Dr. neronimo di ruangan A1! Posisi antrian Anda sekarang: 1</div></div> <div>Gambar 112106 - Hasil Pengujian SkipAntrian: Kasus 2</div>	I: Pasien sudah berada di dalam antrian O: Memajukan pasien menjadi antrian pertama



KASUS 3	KETERANGAN
 <p><i>Gambar 113107 - Hasil Pengujian SkipAntrian: Kasus 3</i></p>	<p>I: Pasien sudah berada di antrian paling depan</p> <p>O: Menunjukkan pesan error</p>
KASUS 4	KETERANGAN
 <p><i>Gambar 114108 - Hasil Pengujian CancelAntrian: Kasus 1</i></p>	<p>I: Pasien berada di urutan mana saja di dalam antrian</p> <p>O: Pasien keluar dari antrian</p>

KASUS 5	KETERANGAN
 <p><i>Gambar 115109 - Hasil Pengujian CancelAntrian: Kasus 2</i></p>	<p>I: Pasien tidak berada di dalam antrian</p> <p>O: Menunjukkan pesan error</p>

Tabel 29 - Hasil Pengujian B06

11.24 Perintah Tidak Valid

EXIT & SAVE	
KASUS 1	KETERANGAN
 <p><i>Gambar 116110 - Hasil Pengujian Input NonValid: Kasus 1</i></p>	<p>I: Perintah tidak sesuai peran</p> <p>O: Pesan error, kembali ke menu</p>
KASUS 2	KETERANGAN
 <p><i>Gambar 117111 - Hasil Pengujian Input NonValid: Kasus 2</i></p>	<p>I: Perintah tidak dikenali</p> <p>O: Pesan error, kembali ke menu</p>

KASUS 3	KETERANGAN
 <p><i>Gambar 118112 - Hasil Pengujian Input NonValid: Kasus 3</i></p>	<p>I: Perintah yang membutuhkan pengguna login dipanggil ketika belum login</p> <p>O: Pesan error, kembali ke menu</p>
KASUS 4	KETERANGAN
 <p><i>Gambar 119113 - Hasil Pengujian Input NonValid: Kasus 4</i></p>	<p>I: Perintah yang membutuhkan pengguna tidak dalam keadaan login dipanggil ketika sudah login</p> <p>O: Pesan error, kembali ke menu</p>

Tabel 30 - Hasil Pengujian Perintah Tidak Valid

LAMPIRAN

IF1210_FormAsistensiTB_1_K02-D.pdf -

<https://drive.google.com/file/d/1JbKIDpttNvJhnMbYx-L6DlAqhGJp958L/view?usp=sharing>

IF1210_FormAsistensiTB_2_K02-D.pdf

https://drive.google.com/file/d/1sftsilwCps482lZrZsArpONuQ3cuLJUx/view?usp=drive_link