

LAPORAN TUGAS BESAR
IF1210 ALGORITMA DAN PEMROGRAMAN 1
SEMESTER II TAHUN 2024/2025



Disusun oleh Kelompok N K-01

Anggota:

| | |
|---------------------------------|-----------------|
| Muhammad Ashkar | 13524007 |
| Muhammad Haris Putra S. | 13524053 |
| Aurelia Jennifer Gunawan | 13524089 |
| Nathanael Shane Bennet | 13524119 |
| Yuhan Fanzuri Nizar | 13524123 |
| Fazri Arrashyi Putra | 13524127 |

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2025

Halaman Pernyataan Kelompok

1. Nama : Muhammad Ashkar
NIM : 13524007

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

2. Nama : Muhammad Haris Putra S.
NIM : 13524053

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

3. Nama : Aurelia Jennifer Gunawan
NIM : 13524089

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

4. Nama : Nathanael Shane Bennet
NIM : 13524119

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

5. Nama : Yuhan Fanzuri Nizar
NIM : 13524123

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

6. Nama : Fazri Arrashyi Putra
NIM : 13524127

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

Daftar Isi

| | |
|--|----|
| Halaman Pernyataan Kelompok..... | 1 |
| Daftar Isi..... | 3 |
| Daftar Tabel..... | 4 |
| Daftar Gambar..... | 5 |
| A. Deskripsi Persoalan..... | 6 |
| B. Rencana Implementasi..... | 6 |
| C. Daftar Pembagian Kerja Kelompok..... | 11 |
| D. Checklist Hasil Rancangan..... | 12 |
| E. Desain Command Primitif..... | 13 |
| F. Desain Kamus Data..... | 14 |
| G. Desain Dekomposisi Algoritmik dan Fungsional Program..... | 14 |
| H. Spesifikasi Modul, Prosedur, dan Fungsi..... | 14 |
| I. Hasil Pengujian Program..... | 14 |
| Lampiran..... | 15 |

Daftar Tabel

- Tabel 1. Rencana Implementasi
- Tabel 2. Daftar Pembagian Kerja Kelompok
- Tabel 3. Checklist Hasil Rancangan
- Tabel 4. Desain Command Primitif

Daftar Gambar

- Gambar 1. Flowchart Dekomposisi Keseluruhan Program Awal–Akhir
- Gambar 2. Flowchart dekomposisi fitur F01 Login
- Gambar 3. Flowchart dekomposisi fitur F02 Register Pasien
- Gambar 4. Flowchart dekomposisi F03 Logout
- Gambar 5. Flowchart dekomposisi fitur F04 Lupa Password
- Gambar 6. Flowchart dekomposisi fitur F05 Menu & Help
- Gambar 7. Flowchart dekomposisi fitur F06/D01 Denah Rumah Sakit
- Gambar 8. Flowchart dekomposisi fitur F07 Lihat User
- Gambar 9. Flowchart dekomposisi fitur F08 Cari User
- Gambar 10. Flowchart dekomposisi fitur F09/D02 Lihat Antrian
- Gambar 11. Flowchart dekomposisi fitur F10 Tambah Dokter
- Gambar 12. Flowchart dekomposisi fitur F11 Diagnosis
- Gambar 13. Flowchart dekomposisi fitur F12 Ngobatin
- Gambar 14. Flowchart dekomposisi fitur F13 Aku boleh pulang ga dok?
- Gambar 15. Flowchart dekomposisi fitur F14 Daftar Check-up
- Gambar 16. Flowchart dekomposisi fitur F15 Antrian Saya!
- Gambar 17. Flowchart dekomposisi fitur F16 Minum Obat
- Gambar 18. Flowchart dekomposisi fitur F17 Minum Penawar
- Gambar 19. Flowchart dekomposisi fitur F18 Exit
- Gambar 20. Flowchart dekomposisi fitur D03 Load
- Gambar 21. Flowchart dekomposisi fitur D04 Save
- Gambar 22. Load, awal folder
- Gambar 23. Load, tidak ada nama folder yang diinput
- Gambar 24. Load, tidak ada folder yang sesuai
- Gambar 25. HalamanUtama, awal masuk
- Gambar 26. HalamanUtama, input salah
- Gambar 27. login , awal masuk
- Gambar 28. login , tidak ada usn
- Gambar 29. login , valid, pasien
- Gambar 30. register, usn sudah dipakai
- Gambar 31. register, format usn salah
- Gambar 32. register, valid, usn benar
- Gambar 33. LupaPassword, usn tidak terdaftar

Gambar 34. LupaPassword, usn benar, kode unik salah
Gambar 35. LupaPassword, usn benar, kode unik benar, diminta password
Gambar 36. Exit, input 4
Gambar 37. Exit, input salah
Gambar 38. Exit, input y, mau save
Gambar 39. Save, Folder buatan save
Gambar 40. Save, user tidak mau save
Gambar 41. Pesan keluar dari sistem
Gambar 42. Pasien, awal
Gambar 43. Pasien, input tidak sesuai
Gambar 44. Logout, validasi
Gambar 45. Logout, y, kembali ke halaman utama
Gambar 46. Logout, n
Gambar 47. Logout, n, kembali ke laman pasien
Gambar 48. Antrian Saya, pasien sudah terdaftar
Gambar 49. Daftar Check-up, pasien sudah terdaftar
Gambar 50. Daftar check-up, list pilihan dokter
Gambar 51. Daftar Check-up, pendaftaran berhasil
Gambar 52. Diagnosis, pasien berhasil didiagnosis penyakit
Gambar 53. Minum obat, pasien meminum obat
Gambar 54. PulangDok, pasien salah urutan minum obat
Gambar 55. Minum Penawar, pasien minum penawar
Gambar 56. Minum Obat, pilihan tidak valid
Gambar 57. PulangDok, pasien sudah boleh pulang
Gambar 58. Antrian saya, pasien sudah berada di dalam ruangan
Gambar 59. Antrian Saya, pasien berada dalam antrian
Gambar 60. Diagnosis, dokter neronimo mendiagnosis pasien gro
Gambar 61. Diagnosis, antrian kosong
Gambar 62. Ngobatin, pasien sudah pernah diobati
Gambar 63. Ngobatin, pasien sedang diobati dokter
Gambar 64. Lihat User, urutkan berdasarkan ID, ascending
Gambar 65. Lihat User, urutkan berdasarkan nama, descending
Gambar 66. Cari user, melalui interface lihat user
Gambar 67. Cari pasien, melalui interface lihat user, ketika input username bukan pasien
Gambar 68. Cari pasien, melalui interface lihat user, ketika input benar
Gambar 69. Cari dokter, melalui interface lihat user
Gambar 70. Cari dokter, melalui interface cari user
Gambar 71. Tambah dokter, ketika dokter sudah terdaftar
Gambar 72. Tambah dokter, ketika ruangan sudah ditempati
Gambar 73. Tambah dokter, dokter berhasil ditambahkan dan diassign ke ruangan

Gambar 74. Lihat Antrian, untuk dokter baru dr. tuntung

Gambar 75. Daftar check-up, dokter baru sudah ditambahkan ke opsi

Gambar 76. History commit repo github yang tidak tercantum di github classroom

Gambar 77. Hasil pindai MoM form asistensi 1

Gambar 78. Hasil Pindai MoM Asistensi 2

Deskripsi Persoalan

Tugas besar ini meminta kami untuk membangun sebuah sistem informasi rumah sakit digital yang dapat digunakan oleh tiga peran utama: pasien, dokter, dan manajer. Sistem ini ditujukan untuk menangani permasalahan administratif dan medis yang muncul akibat kekacauan operasional rumah sakit milik Gro, seorang mantan penjahat yang kini mengurus ratusan makhluk kuning bernama *Nimon*.

Permasalahan yang dihadapi cukup kompleks, mulai dari pendaftaran pasien, pengelolaan antrian, diagnosis otomatis, hingga manajemen data pengguna dan ruangan. Sistem ini juga perlu mampu menyimpan dan memuat data dari file eksternal agar informasi tidak hilang saat program ditutup.

Dalam program ini, terdapat beragam fungsionalitas yang harus diimplementasikan, seperti:

- Login dan registrasi untuk mengatur akses pengguna
- Pendaftaran check-up dan antrian pasien agar proses berobat lebih terstruktur
- Diagnosa penyakit otomatis dan pemberian resep obat oleh dokter
- Pengecekan kondisi pasien dan validasi kesembuhan sebelum diperbolehkan pulang
- Manajemen dokter, ruangan, serta statistik rumah sakit oleh manajer
- Fitur penyimpanan dan pemuatan data ke/dari file CSV untuk menjaga data antar sesi

Kami juga diwajibkan menggunakan berbagai jenis struktur data dan ADT (Abstract Data Type), seperti List, Stack, Queue, Set, Map, Matrix, dan File Eksternal, serta menerapkan konsep searching, sorting, filtering, dan modularisasi dengan fungsi/prosedur.

Dengan kata lain, tugas besar ini membangun sebuah sistem manajemen rumah sakit yang otomatis, terstruktur, dan efisien, agar Nimons-nimons yang ceroboh bisa mendapatkan penanganan medis yang layak dan tidak menambah kekacauan di rumah sakit Gro.

A. Rencana Implementasi

Tabel 1. Rencana Implementasi

| Implementasi ADT | Fitur | Deskripsi Implementasi | Alasan Implementasi |
|--|-------------|--|---|
| ADT Sederhana, List, Procedure, Array Search | F01 – Login | Digunakan untuk masuk ke dalam sistem bagi user yang belum melakukan login | ADT List dinamis agar memudahkan penyesuaian ukuran array dari users jika |

| | | | |
|-----------------------------|-----------------------------|--|--|
| | | | <p>melebihi kapasitas.</p> <p>Array Search untuk autentifikasi username dan password.</p> <p>ADT Sederhana definisi tipe Identitas, KondisiPasien, User.</p> <p>Procedure dipanggil untuk validasi format username..</p> |
| ADT Set | F02 – Register Pasien | Digunakan untuk registrasi, khusus pasien. | ADT Set karena menyimpan data username user yang bersifat unik (<i>case-insensitive</i>). |
| ADT Sederhana | F03 – Logout | Digunakan untuk keluar dari akun yang sedang dipakai, hanya dapat dipanggil ketika sedang ada akun yang <i>logged in</i> . | <p>ADT Sederhana definisi tipe Identitas, KondisiPasien, User.</p> <p>Free pointer dari user, dan memberikan nilai NULL.</p> |
| ADT Sederhana, Array Search | F04 – Lupa Password | Digunakan untuk memulihkan/mengupdate password apabila user lupa password | <p>Array Searching autentifikasi kode unik dari user dengan password pada array users.</p> <p>ADT Sederhana definisi tipe Identitas, KondisiPasien, User.</p> |
| ADT Sederhana, Prosedur | F05 – Menu & Help | Digunakan untuk menunjukkan apa saja yang bisa user lakukan di rumah sakit, | <p>ADT Sederhana definisi tipe Identitas, KondisiPasien, User Pilihan, PilihanPasien.</p> <p>Prosedur dipanggil pada laman menu sesuai pilihan user.</p> |
| ADT Matriks | F06/D01 – Denah Rumah Sakit | Digunakan untuk melihat denah dan detail ruangan di rumah sakit | ADT Matriks sangat tepat karena denah memiliki komponen |

| | | | |
|--|------------------|---|--|
| | | | koordinat X, Y yang dapat ditampilkan dengan baik oleh matriks. |
| ADT List, Sederhana, Array Sort, Array Filter, Array Search(binary), fungsi & prosedur | F07 – Lihat User | Digunakan untuk melihat data baik secara spesifik maupun keseluruhan user, khusus manager | <p>ADT List untuk users yaitu array of user.</p> <p>ADT Sederhana definisi tipe Identitas, Kondisi Pasien, User.</p> <p>Array Filter untuk manager agar dapat memilih untuk melihat data baik secara keseluruhan atau spesifik (implementasi fungsi & prosedur) difilter role.</p> <p>Array Search untuk autentifikasi nama user dengan data users.</p> <p>Array Sort untuk sorting array of user alias users berdasarkan jenis urutan yang diinginkan.</p> <p>Fungsi & prosedur untuk sorting mempersingkat kode yang berulang.</p> |
| ADT List, fungsi & prosedur, Array Search(binary), Array sort, Array filter. | F08 – Cari User | Digunakan untuk mencari data pengguna | <p>ADT List untuk users yaitu array of user.</p> <p>Array Filter untuk manager agar dapat memilih untuk mencari data user baik secara umum atau spesifik (implementasi fungsi & prosedur) difilter role.</p> <p>Array Search untuk autentifikasi nama user dengan data users.</p> |

| | | | |
|----------------------------------|-----------------------------------|--|---|
| | | | <p>Array Sort untuk sorting array of user alias users berdasarkan jenis urutan yang diinginkan.</p> <p>Fungsi & prosedur untuk sorting mempersingkat kode yang berulang.</p> |
| ADT List | F09/D02 – Lihat Antrian | Digunakan untuk melihat rincian di seluruh ruangan saat ini, ruangan kosong tidak perlu ditampilkan | ADT List untuk users yaitu array of user dan untuk array antrian. |
| ADT Set | F10 – Tambah Dokter | Digunakan oleh manager untuk menambahkan dokter baru | ADT Set digunakan untuk validasi keunikan username |
| ADT List, Map, Fungsi & Prosedur | F11 – Diagnosis | Digunakan untuk mengecek penyakit pasien dengan menyesuaikan data kondisi pasien dan batasan kondisi | <p>ADT List untuk array of user, penyakit, dan kondisi.</p> <p>ADT Map untuk menghubungkan diagnosis berdasarkan data kondisi dan penyakit dengan data kondisi.</p> <p>Fungsi & prosedur untuk memanggil kondisi pasien dan mempersingkat kode yang berulang.</p> |
| ADT Map | F12 – Ngobatin | Digunakan untuk memberikan obat kepada pasien secara terurut | ADT Map untuk menyimpan data obat yang cocok untuk pasien yang mengidap penyakit tertentu <penyakit, list obat> |
| ADT Stack, List | F13 – Aku boleh pulang ga, 🥺 dok? | Digunakan untuk melakukan validasi untuk memastikan bahwa pasien sudah siap untuk pulang | <p>ADT Stack definisi urutan stack obat pada perut pasien.</p> <p>ADT List menyimpan urutan obat yang seharusnya</p> |

| | | | |
|------------------------------|-----------------------|--|---|
| | | | <p>diminum oleh pasien</p> <p>Disini akan dilakukan validasi kecocokan urutan pada stack perut dan list urutan obat yang benar.</p> |
| ADT Map, Queue (Linked List) | F14 – Daftar Check-Up | Digunakan oleh pasien untuk mendaftar melakukan check-up dengan dokter | <p>ADT Map definisi <dokter, antrian pasien></p> <p>ADT queue (linked list) definisi antrian pasien, elemen dalam bentuk linked list yang menyimpan nama pasien dan nama dokter..</p> |
| ADT Queue, Map | F15 – Antrian Saya! | Digunakan oleh pasien untuk melihat status antriannya | <p>ADT Queue definisi antrian pengguna pasien.</p> <p>ADT Map untuk cek posisi dalam antrian berdasarkan nama pengguna.</p> |
| ADT Stack, List | F16 – Minum Obat | Digunakan oleh pasien untuk mengambil obat dari inventory dan memasukkannya ke perut. | <p>ADT List definisi inventory pasien.</p> <p>ADT Stack definisi perut, dimana obat akan ditumpuk ke dalam stack setiap kali pasien minum obat yang telah diambil dari inventory.</p> |
| ADT Stack, List | F17 – Minum Penawar | Digunakan oleh pasien yang salah minum obat untuk minum penawar, sehingga obat terakhir yang sebelumnya diminum langsung dikeluarkan dari perut pasien | <p>ADT List definisi inventory, untuk menyimpan obat yang telah dikeluarkan dari perut</p> <p>ADT Stack definisi perut pasien, pop obat terakhir dari perut</p> |

| | | | |
|--------------------------------------|------------|--|---|
| File Eksternal, fungsi & prosedur | F18 – Exit | Digunakan oleh user untuk keluar dari sistem, dan memutuskan apakah user masih ingin melakukan save terlebih dahulu | Save ke file data format csv, bisa memilih untuk save data atau tidak. |
| File Eksternal, fungsi & prosedur | D03 – Load | Digunakan ketika user memulai program dengan memberi nama folder yang berisi file penyimpanan dan melakukan validasi | Load dari folder dan inisialisasi semua struktur data |
| File Eksternal, fungsi & prosedur | D04 – Save | Digunakan untuk menyimpan data ke dalam file. | Menyimpan data ke csv bisa memilih untuk save ke folder pilihan pengguna berdasarkan inputan. |

B. Daftar Pembagian Kerja Kelompok

Tabel 2. Daftar Pembagian Kerja Kelompok

| Fitur | Implementasi | NIM Desainer | NIM Coder | NIM Tester |
|--------------------------|---|--------------|-----------|------------|
| F01 – Login | Procedure cekFormatUsn, ADT Sederhana, List, Procedure, Array Search Procedure clearScreen | 13524053 | 13524053 | 13524053 |
| F02 – Register Pasien | Procedure cekFormatUsn, ADT Set | 13524053 | 13524053 | 13524053 |
| F03 – Logout | Menerapkan NULL pada pointer user, ADT Sederhana | 13524053 | 13524053 | 13524053 |
| F04 – Lupa Password | Procedure cekFormatUsn, ADT Sederhana, Array Search | 13524053 | 13524053 | 13524053 |

| | | | | |
|-----------------------------|---|----------------------|----------------------|----------------------|
| F05 – Menu & Help | Help diberikan sesuai role, ADT Sederhana | 13524089 | 13524053 13524089 | 13524053 13524089 |
| F06/D01 – Denah Rumah Sakit | ADT Matriks, procedure muatDataRumahSakit, procedure tampilkanDenahRumahSakit, procedure tampilkanDetailRuang, procedure tampilkanDenahDanDetail, procedure tampilkanInventarisObat | 13524127 | 13524127 13524053 | 13524127 13524053 |
| F07 – Lihat User | procedure lihatUser, ADT List, binary search dan array sort. | 13524127 | 13524127 | 13524127 13524053 |
| F08 – Cari User | ADT List, binary search, array sort, procedure queryAndPrintUserTable, procedure cariUser | 13524127 | 13524127 | 13524127 13524053 |
| F09/D02 – Lihat Antrian | procedure tampilkanSemuaAntrianUntukManager, ADT List | 13524007 13524053 | 13524007 13524053 | 13524007 13524053 |
| F10 – Tambah Dokter | Procedure assignDokter(User *dokterUser), procedure lamanTambahDokter(), fungsi ruanganStringToInt(const char *kode, int cols) . | 13524127 13524053 | 13524127 13524053 | 13524127 13524053 |
| F11 – Diagnosis | procedure lamanDiagnosis, procedure printQueuePositions, ADT List, ADT Map | 13524053 | 13524053 | 13524053 |
| F12 – Ngobatin | Procedure lamanNgobatin(), procedure | 13524127 13524053 | 13524127 13524053 | 13524127 13524053 |

| | | | | |
|-----------------------------------|---|----------------------------------|----------------------------------|----------------------------------|
| | tambahObatInventory. | | | |
| F13 – Aku boleh pulang ga, 🥺 dok? | procedure resetKondisiPasi en, procedure cekUrutanObat, procedure pulangDok, procedure compareObatPenyakit , fungsi getPenyakitIdByPeny akitName | 13524119 13524053 | 13524119 13524053 | 13524119 13524053 |
| F14 – Daftar Check-Up | procedure pasien_terdaftar, ADT Queue (dengan Linked List), ADT Map | 13524089 13524127 13524053 | 13524089 13524127 13524053 | 13524089 13524127 13524053 |
| F15 – Antrian Saya! | fungsi lamanLihatAntrianSa ya, ADT Queue, ADT Map | 13524089 13524053 | 13524089 13524053 | 13524089 13524053 |
| F16 – Minum Obat | procedure lamanMinumObat, procedure cariObatById, procedure tambahObatDalamPer ut, ADT Stack, ADT List | 13524089 13524053 | 13524089 13524053 | 13524089 13524053 |
| F17 – Minum Penawar | procedure lamanMinumPenawar , procedure tambahObatInventory, ADT Stack, ADT List | 13524053 | 13524053 | 13524053 |
| F18 – Exit | file eksternal, pemanggilan procedure save | 13524053 | 13524053 | 13524053 |
| D03 – Load | Prosedur CSVtoArr, Memuat folder sesuai input user, File Eksternal, fungsi & prosedur | 13524053 213524089 | 13524053 13524089 | 13524053 |
| D04 – Save | Prosedur ArrtoCSV, Menyimpan data ke | 13524053 13524089 | 13524053 13524089 | 13524053 |

| | | | | |
|--|---|--|--|--|
| | folder sesuai input user, File Eksternal, fungsi & prosedur | | | |
|--|---|--|--|--|

C. Checklist Hasil Rancangan

Tabel 3. Checklist Hasil Rancangan

| Fitur | Desain | Implementasi | Testing |
|-----------------------------------|--------|--------------|---------|
| F01 – Login | V | V | V |
| F02 – Register Pasien | V | V | V |
| F03 – Logout | V | V | V |
| F04 – Lupa Password | V | V | V |
| F05 – Menu & Help | V | V | V |
| F06/D01 – Denah Rumah Sakit | V | V | V |
| F07 – Lihat User | V | V | V |
| F08 – Cari User | V | V | V |
| F09/D02 – Lihat Antrian | V | V | V |
| F10 – Tambah Dokter | V | V | V |
| F11 – Diagnosis | V | V | V |
| F12 – Ngobatin | V | V | V |
| F13 – Aku boleh pulang ga, 🥰 dok? | V | V | V |
| F14 – Daftar Check-Up | V | V | V |
| F15 – Antrian Saya! | V | V | V |
| F16 – Minum Obat | V | V | V |
| F17 – Minum Penawar | V | V | V |

| | | | |
|------------|---|---|---|
| F18 – Exit | V | V | V |
| D03 – Load | V | V | V |
| D04 – Save | V | V | V |

Keterangan:

- V: sudah selesai dikerjakan,
- X: dikerjakan, tapi belum selesai,
- -: tidak dikerjakan sama sekali.

D. Desain Command Primitif

Tabel 4. Desain Command Primitif

| Nama Command | Masukan | Keluaran |
|---------------------|--|---|
| CSVtoArr | Path file | array User[], totalUser |
| ArrtoCSV | path file, array User[], totalUser | file .csv yang tersimpan |
| cekFormatUsn | Inputan username dari user | Skema validasi 1, input hanya boleh huruf |
| createUser | username, password, role, identitas, kondisi | struct User |
| appendUser | list users[], totalUser, userBaru | list baru dengan userBaru ditambahkan |
| isUsernameAvailable | array User[], totalUser, username | boolean (true jika username belum ada) |
| cariUserLogin | username, password | indeks User dalam array / -1 jika tidak ada |
| login | username, password | boolean, true jika berhasil login, false jika gagal |
| logout | - | User *user = NULL, kembali ke menu |
| registerPasien | username, password | boolean, true jika berhasil regis, false jika gagal, user baru dengan role pasien terdaftar |
| lupaPassword | username, kodeUnik, | Mengganti password user jika |

| | | |
|-----------------------------------|---|--|
| | newPassword | kode unik benar. |
| help | role, username | list command yang dapat dipanggil oleh user |
| lamanLihatAntrianSaya | rumahSakit, map, users, userCount | detail status antrian diagnosis/ngobatan pasien. |
| lamanCariUser | by, selectOpt, selectTerm | Memunculkan user yang dicari berdasarkan ID atau username yg diinput. |
| daftar_checkup | user.kondisiPasien, pilihan | user, map, pasien, penyakit User pasien mempunyai atribut kondisiPasien baru. |
| tampilkanDenahRumahSakit | rumahSakit | layout matrix denah rumahSakit |
| tampilkanDetailRuangan | rumahSakit, kodeRuangan | list detail ruangan (kapasitas, dokter, pasien dalam ruangan) |
| tampilkanSemuaAntrianUntukManajer | rumahSakit, users, userCount | tampilan layout denah rumah sakit, semua antrian dan ruangan yang tidak kosong |
| lamanDiagnosis | user, map, pasien, penyakit | User pasien mempunyai atribut riwayat_penyakit baru(jika terdiagnosa). |
| lamanLihatUser | users, jumlah_user | integer, tampilan seluruh data pengguna dalam format tabel |
| lamanMinumObat | obatTerpilih | Push obat pada stack perut user. |
| minum_penawar | perut, user | Pop obat pada stack perut user. |
| lamanNgobatan | user, jumlah_penyakit, dokter, pasien, obatpenyakit | Memberi obat pada pasien yang terdiagnosa. |
| pulangDok | urutanMinumObat, perut, obatExpected, obatMinum | User boleh pulang/tidak bergantung kondisi perut dan urutan minumObat, pesan apakah pasien sudah boleh pulang atau belum |
| lamanTambahDokter | username, password | integer, 1 jika berhasil mendaftarkan dokter, 0 jika tidak, menambah dokter pada |

| | | |
|--------------|---|--|
| | | array users. |
| assignDokter | username, ,kodeRuangan | Dokter baru menempati ruangan baru di rumahSakit |
| exit | Character (y/n yes/no), dan masukan nama folder | file eksternal yang menyimpan data rumahSakit, membuat folder baru jika nama folder belum ada. |

E. Desain Kamus Data

MAX_USER : integer = 100
 MAX_LINE_LEN : integer = 1024
 MAX_FIELD : integer = 50
 MAX_COL_LEN : integer = 100
 MAX_OBAT : integer = 50

type Identitas : <
 id : integer,
 username : string,
 password : string,
 role : string {"Admin", "Dokter", "Pasien"} >

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

type ObatPenyakit: <
 obat_id : integer,
 penyakit_id : integer,
 urutan_minum: integer >

type StackPerut: <
 items : Obat,
 top : integer,
 capacity: integer>

type KondisiPasien : <
 riwayat_penyakit : string

suhu_tubuh : float (dalam °C) ,
 tekanan_darah_sistolik : integer (mmHg) ,
 tekanan_darah_diastolik : integer (mmHg) ,
 detak_jantung : integer (bpm) ,
 saturasi_oksigen : float (%) ,
 kadar_gula_darah : integer (mg/dL) ,
 berat_badan : float (kg) ,
 tinggi_badan : integer (cm) ,
 kadar_kolesterol : integer (mg/dL) ,
 trombosit : integer (/μL),
 inventory[MAX_OBAT] : Obat,
 jumlahObat : integer,
 perut : StackPerut,
 sudahDiagnosis, sudahObati: integer >

type User : <

identitas: Identitas,
 kondisi: KondisiPasien >

type Penyakit: <

id : integer
 nama : string
 suhu_tubuh_min : real,
 suhu_tubuh_max: real,
 tekanan_darah_sistolik_min : integer,
 tekanan_darah_sistolik_max: integer,
 tekanan_darah_diastolik_min : integer,
 tekanan_darah_diastolik_max: integer,
 detak_jantung_min : integer,
 detak_jantung_max: integer,
 saturasi_oksigen_min : real,
 saturasi_oksigen_max: real,
 kadar_gula_darah_min : integer,
 kadar_gula_darah_max: integer,
 berat_badan_min : real,
 berat_badan_max: real,
 tinggi_badan_min : integer,
 tinggi_badan_max: integer,
 kadar_kolesterol_min : integer,
 kadar_kolesterol_max: integer,

trombosit_min : integer,
trombosit_max: integer>

type

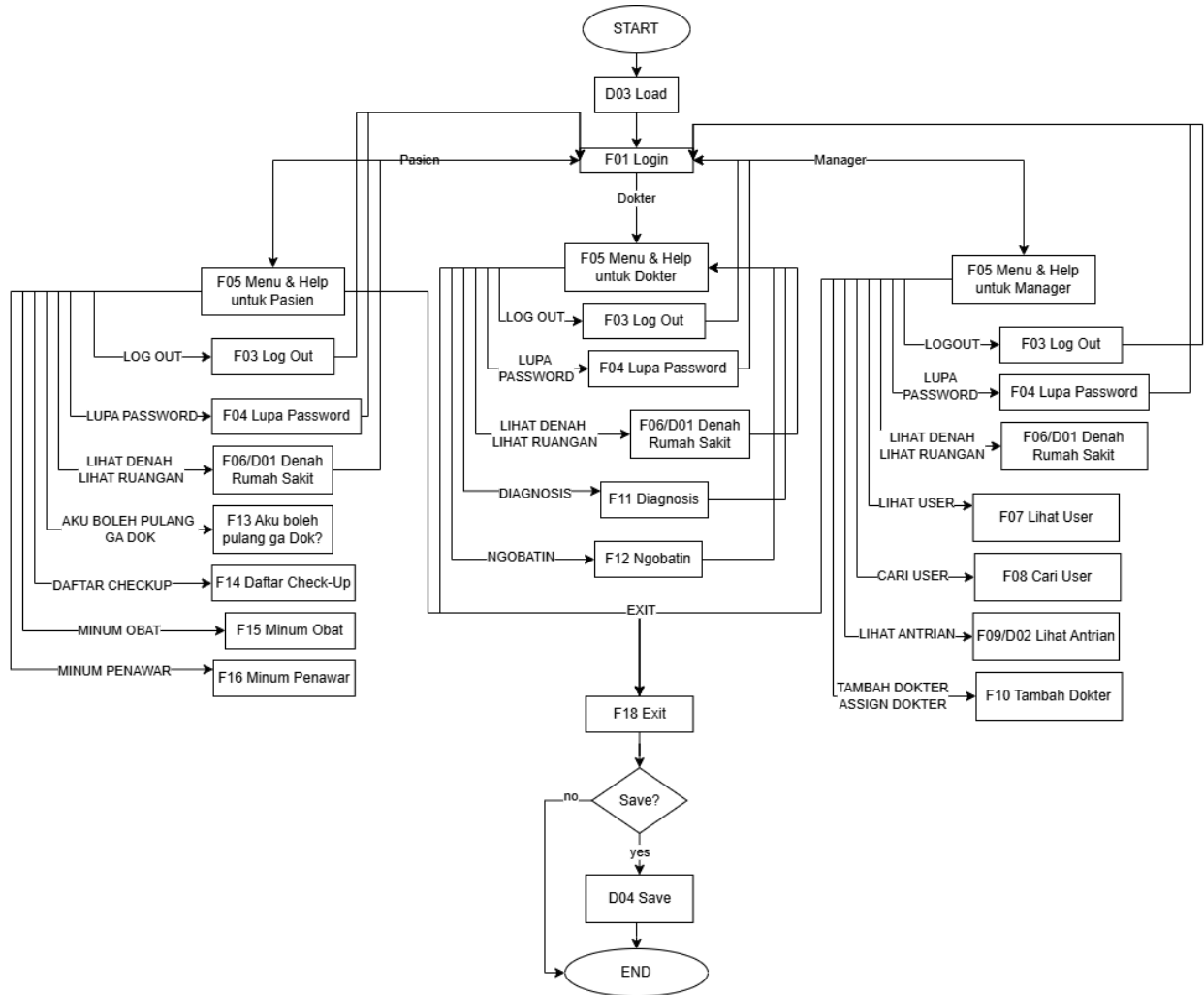
Pilihan = (LOGIN = 1, REGISTER, LUPAPASSWORD, EXIT)

PilihanPasien = (
 DAFTARCHECKUP = 1,
 ANTRIAN SAYA,
 MINUMOBAT,
 MINUMPENAWAR,
 DENAHRUMAHSAKIT,
 PULANGDOK,
 LOGOUTP
)

PilihanManager = (
 DENAHRUMAHSAKITMANAGER = 1,
 LIHATUSER,
 CARIUSER,
 TAMBAHDOKTER,
 LOGOUTM
)

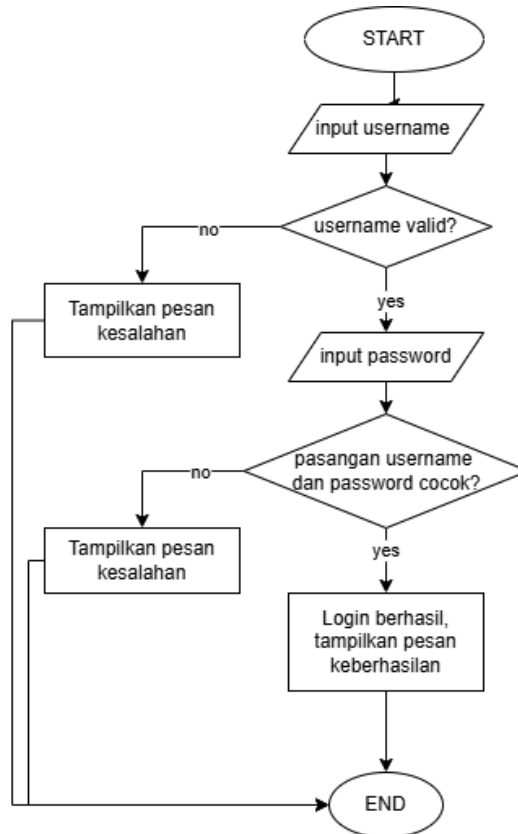
PilihanDokter = (
 DIAGNOSIS = 1,
 NGOBATIN,
 LOGOUTD
)

F. Desain Dekomposisi Algoritmik dan Fungsional Program



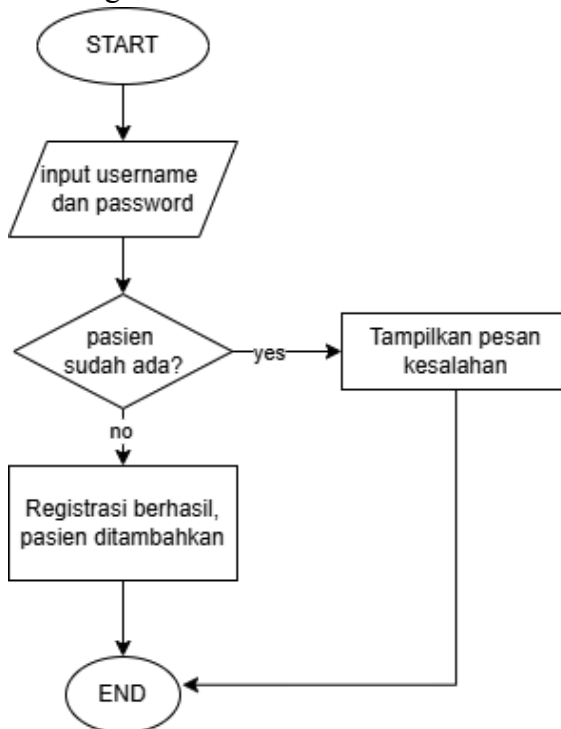
Gambar 1. Flowchart Dekomposisi Keseluruhan Program Awal–Akhir

1. Dekomposisi fitur F01 Login



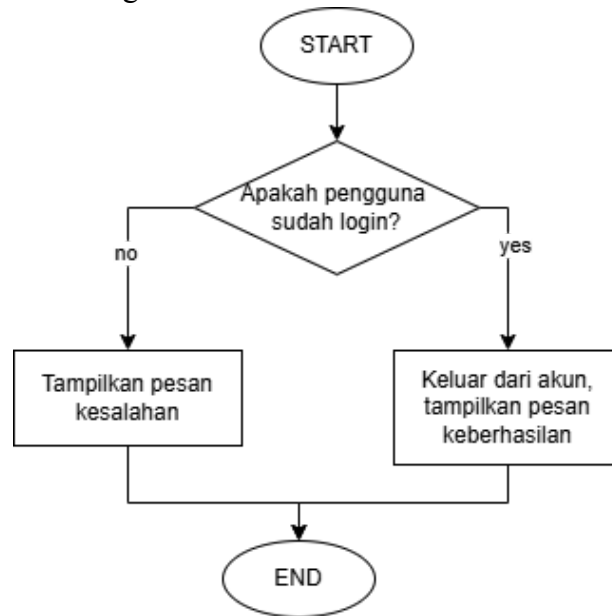
Gambar 2. Flowchart dekomposisi fitur F01 Login

2. Dekomposisi fitur F02 Register Pasien



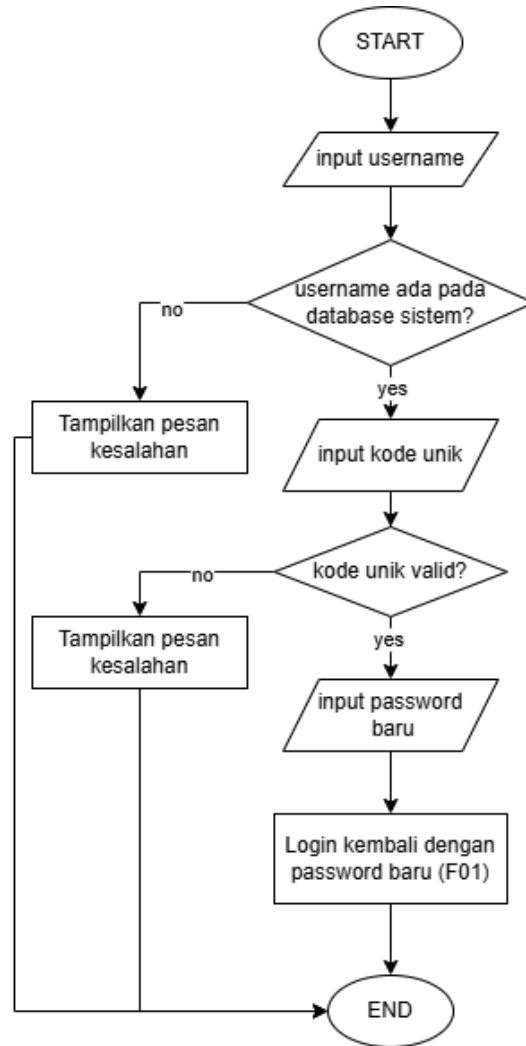
Gambar 3. Flowchart dekomposisi fitur F02 Register Pasien

3. Dekomposisi fitur F03 Logout



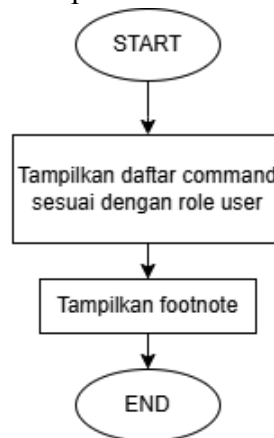
Gambar 4. Flowchart dekomposisi F03 Logout

4. Dekomposisi fitur F04 Lupa Password



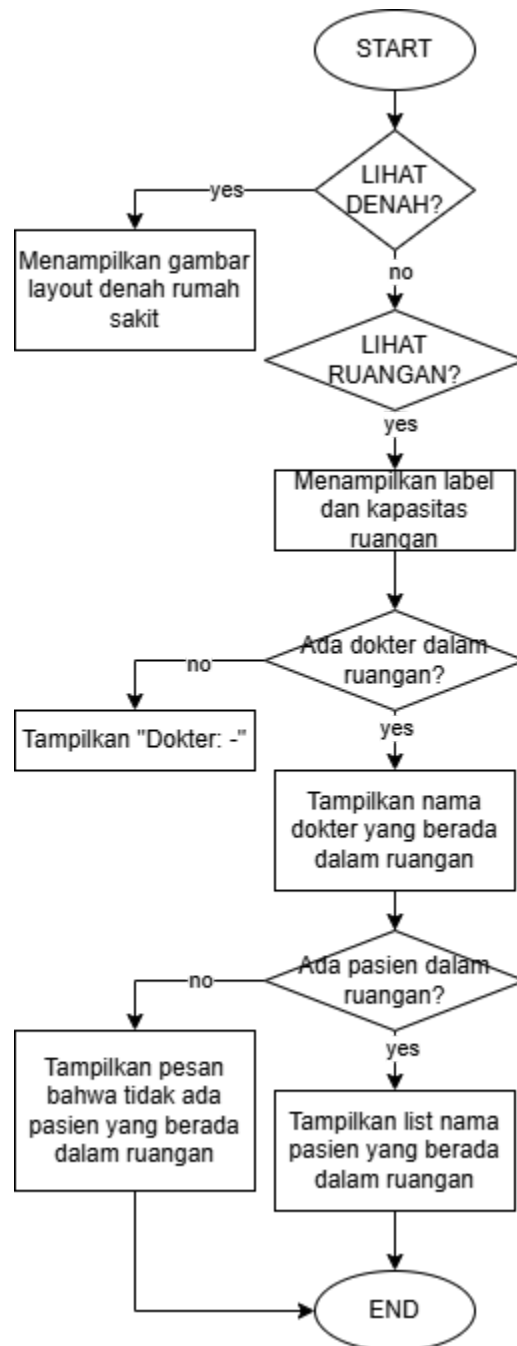
Gambar 5. Flowchart dekomposisi fitur F04 Lupa Password

5. Dekomposisi fitur F05 Menu & Help



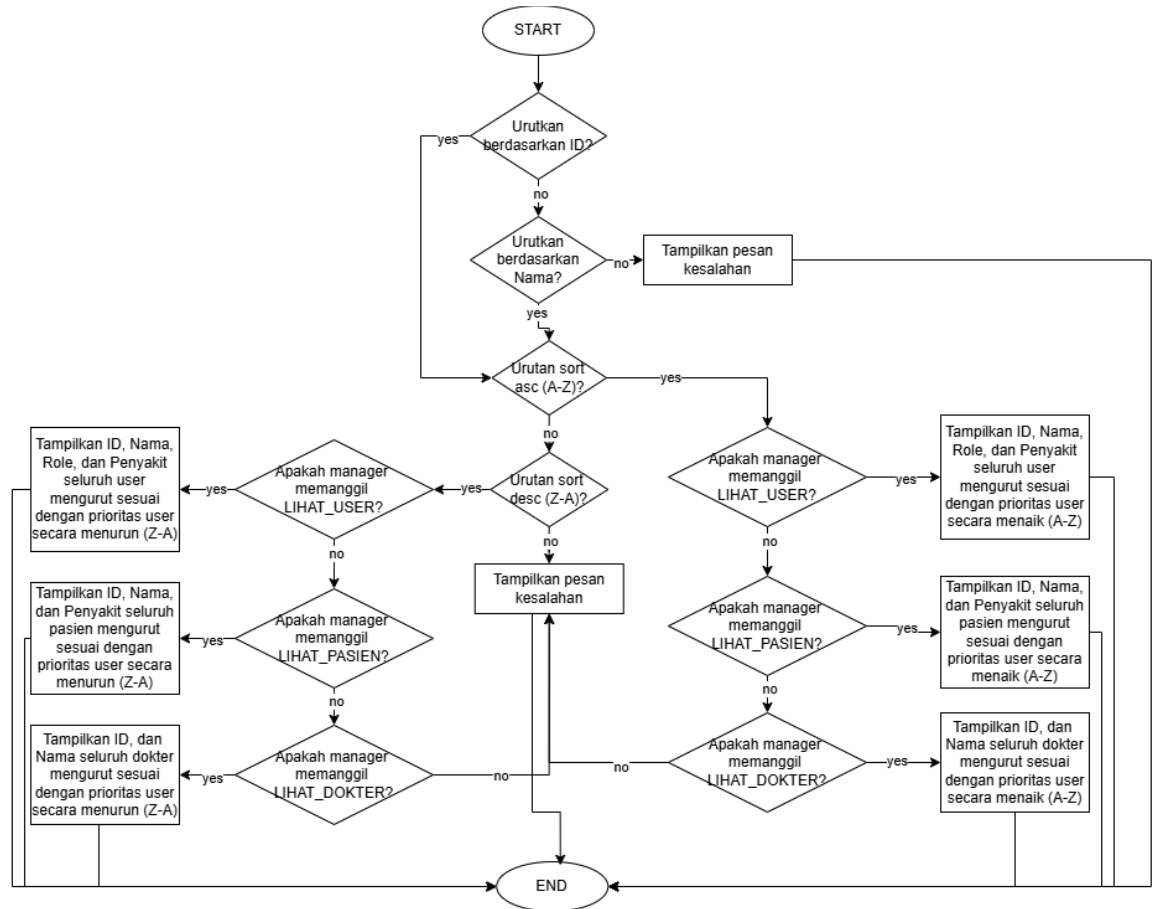
Gambar 6. Flowchart dekomposisi fitur F05 Menu & Help

6. Dekomposisi fitur F06/D01 Denah Rumah Sakit



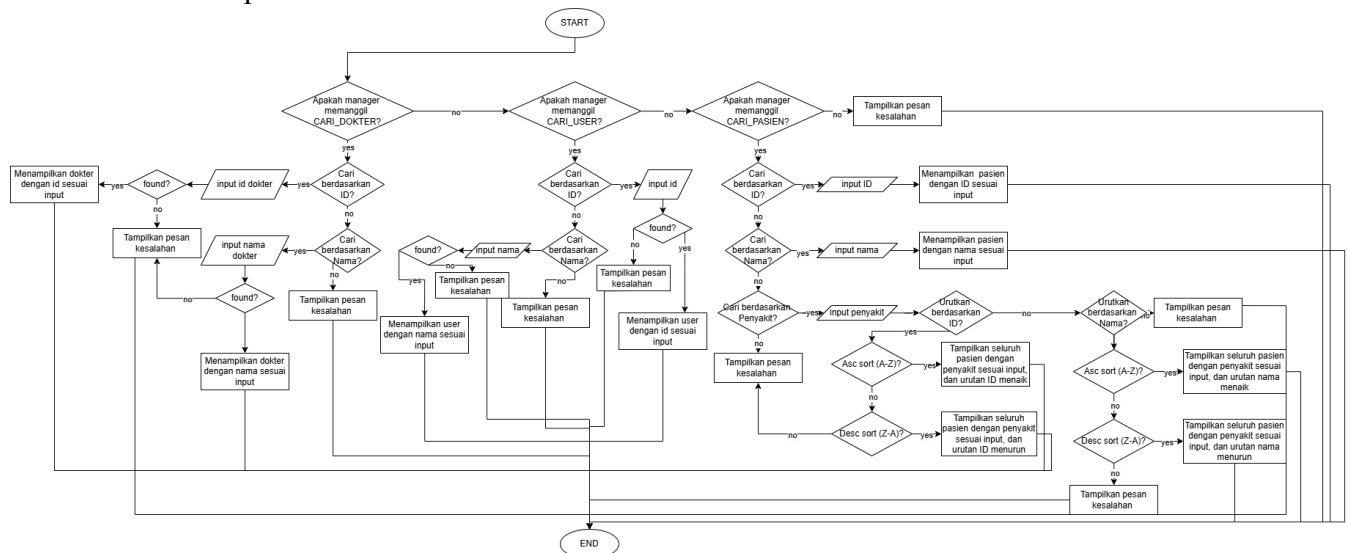
Gambar 7. Flowchart dekomposisi fitur F06/D01 Denah Rumah Sakit

7. Dekomposisi fitur F07 Lihat User



Gambar 8. Flowchart dekomposisi fitur F07 Lihat User

8. Dekomposisi fitur F08 Cari User



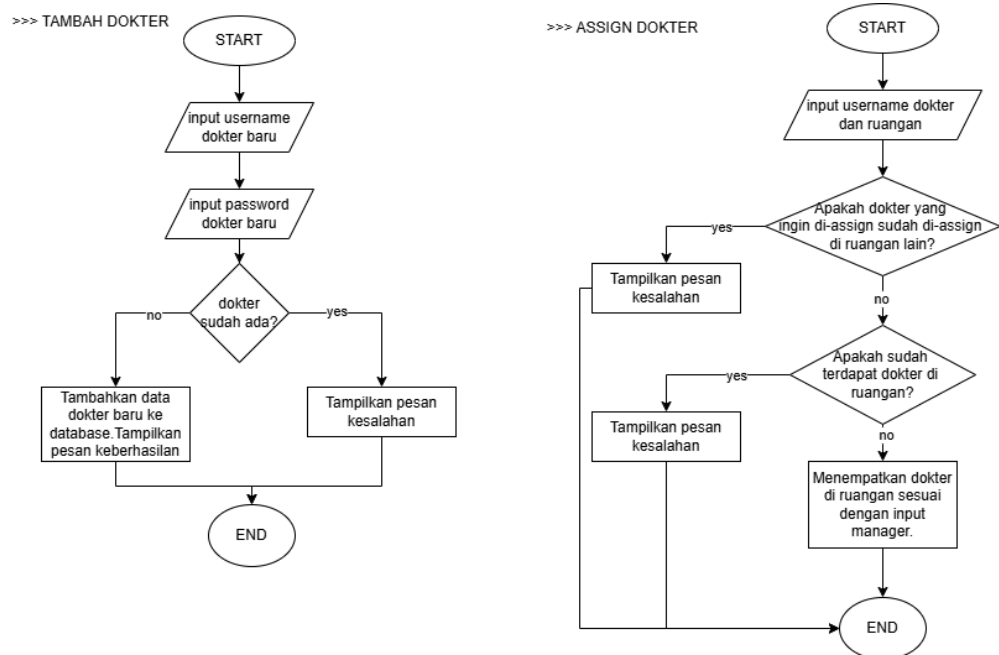
Gambar 9. Flowchart dekomposisi fitur F08 Cari User

9. Dekomposisi fitur F09/D02 Lihat Antrian



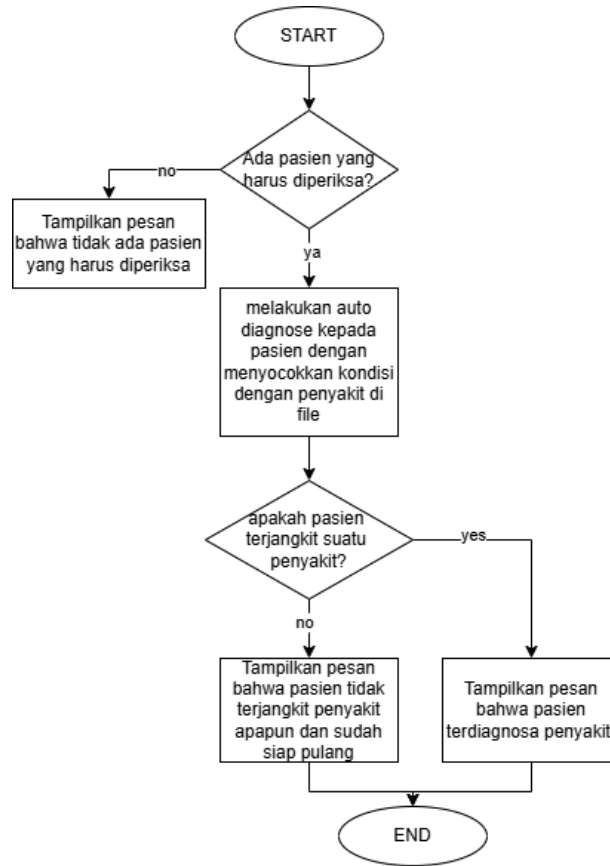
Gambar 10. Flowchart dekomposisi fitur F09/D02 Lihat Antrian

10. Dekomposisi fitur F10 Tambah Dokter



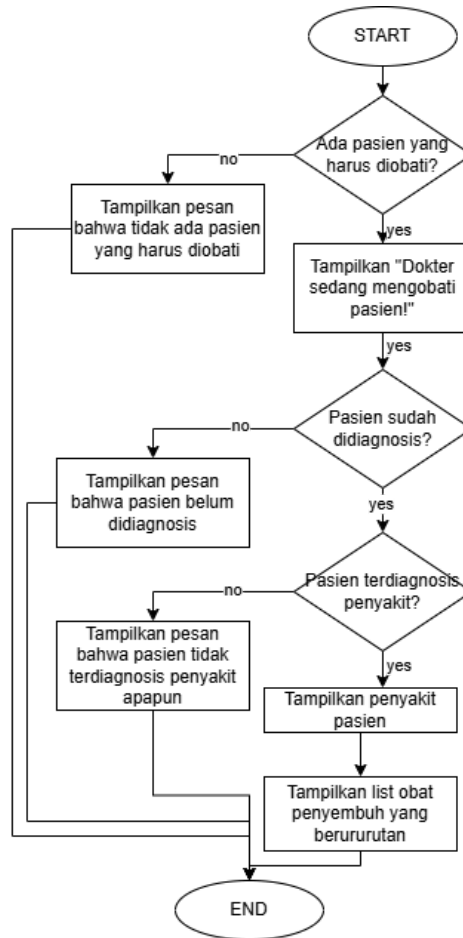
Gambar 11. Flowchart dekomposisi fitur F10 Tambah Dokter

11. Dekomposisi fitur F11 Diagnosis



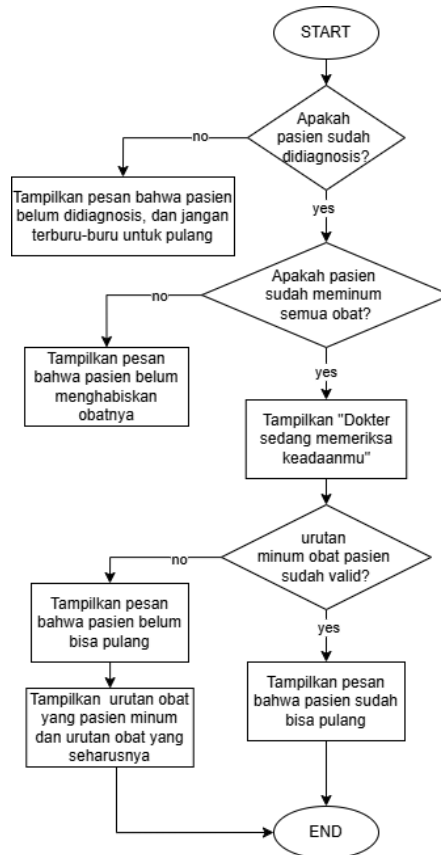
Gambar 12. Flowchart dekomposisi fitur F11 Diagnosis

12. Dekomposisi fitur F12 Ngobatin



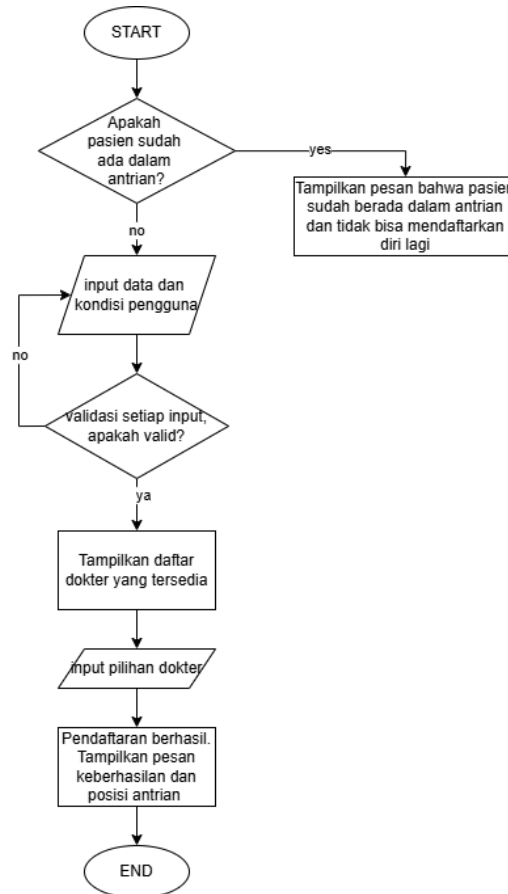
Gambar 13. Flowchart dekomposisi fitur F12 Ngobatin

13. Dekomposisi fitur F13 Aku boleh pulang ga dok?



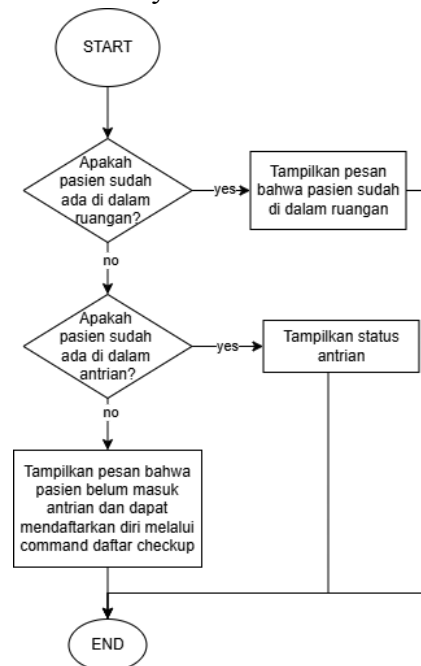
Gambar 14. Flowchart dekomposisi fitur F13 Aku boleh pulang ga dok?

14. Dekomposisi fitur F14 Daftar Check-Up



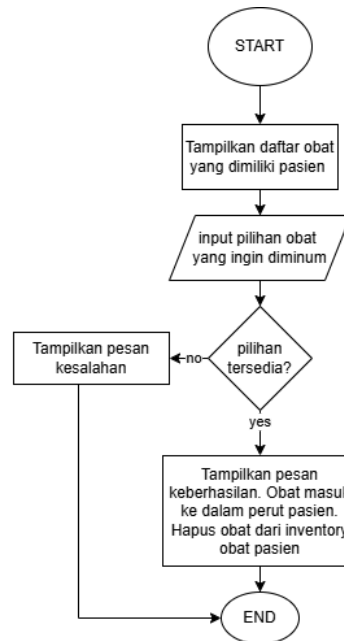
Gambar 15. Flowchart dekomposisi fitur F14 Daftar Check-up

15. Dekomposisi fitur F15 Antrian Saya!



Gambar 16. Flowchart dekomposisi fitur F15 Antrian Saya!

16. Dekomposisi fitur F16 Minum Obat



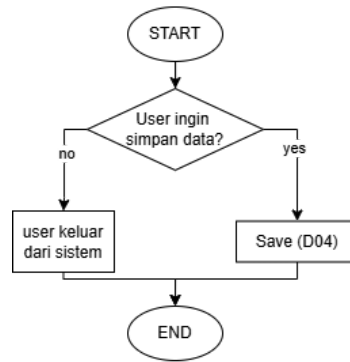
Gambar 17. Flowchart dekomposisi fitur F16 Minum Obat

17. Dekomposisi fitur F17 Minum Penawar



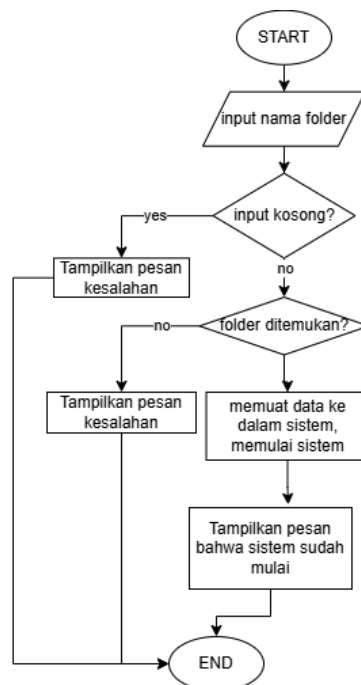
Gambar 18. Flowchart dekomposisi fitur F17 Minum Penawar

18. Dekomposisi fitur F18 Exit



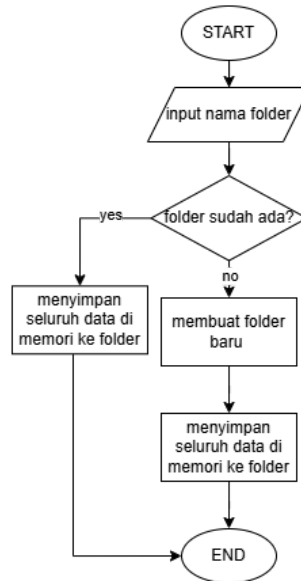
Gambar 19. Flowchart dekomposisi fitur F18 Exit

19. Dekomposisi fitur D03 Load



Gambar 20. Flowchart dekomposisi fitur D03 Load

20. Dekomposisi fitur D04 Save



Gambar 21. Flowchart dekomposisi fitur D04 Save

G. Spesifikasi Modul, Prosedur, dan Fungsi

1. Spesifikasi fitur F01 Login

function login () → boolean

{Menerima masukan username, melalui prosedur cekFormatUsn(input user.identitas.username : string), dan password pengguna}

{Memberikan output 1 jika pasangan username dan password sesuai dengan di database sistem dan pengguna berhasil login, output 0 jika tidak sesuai}

KAMUS LOKAL

users : array of User[0..jumlah_user] (extern)

user : User (extern)

jumlah_user : integer (extern)

i, format : integer

nama, role : string

valid : boolean

procedure clearScreen()

{Proses : Membersihkan tampilan layar}

{I.S. Layar menampilkan teks atau output sebelumnya}

{F.S. Layar menjadi kosong/bersih dari teks sebelumnya}

procedure waitForEnter()

{Proses: Menunggu pengguna menekan tombol ENTER}

{I.S. Program selesai menampilkan informasi sebelumnya}

{F.S. Program berhenti sejenak sampai pengguna menekan ENTER}

procedure cekFormatUsn(input format : string, input user: User)

{Proses: Meminta input username dari pengguna dan mengecek apakah sesuai format yang diinginkan sistem}

{I.S. user.identitas.username belum terisi}

```

{F.S. Username tersimpan pada user.identitas.username}
procedure lowercase(input/output s : string)
{Proses: Mengubah seluruh karakter dalam string menjadi huruf kecil}
{I.S. Sebuah input berupa string, mungkin mengandung huruf kapital}
{F.S. String yang telah diubah seluruhnya menjadi huruf kecil/lowercase}

```

ALGORITMA

```

clearScreen
cekFormatUsn(format, user)
nama ← user.identitas.username
output ("Password: ")
false
i traversal [0..jumlah_user - 1]
    if lowercase(users[i].identitas.username) =
lowercase(user.identitas.password) then
        if lowercase(users[i].identitas.password) =
lowercase(user.identitas.password) then
            valid ← true
            role ← users[i].identitas.role
            user.identitas.role ← role
            output ("Selamat pagi ", role, " ", nama)
            output ("Sedang memuat laman ", role, " ...")
            → valid
        else
            clearScreen
            output ("Password salah untuk pengguna yang
bernama", nama)
            waitForEnter
            → valid
    else
        clearScreen
        output("Tidak ada pengguna yang bernama ", nama)
        waitForEnter
        → valid

```

2. Spesifikasi fitur F02 Register Pasien

```

function registerPasien() → boolean
{Menerima input username pengguna melalui prosedur cekFormatUsn, lalu melakukan
validasi apakah pasien sudah terdaftar atau belum. Memberikan output true jika registrasi
berhasil, output false jika pasien sudah terdaftar}

```

KAMUS LOKAL

```

users : array of User (extern)
user : User (extern)

```

```

jumlah_user : integer (extern)
valid : boolean
format : integer
procedure clearScreen()
{Proses : Membersihkan tampilan layar}
{I.S. Layar menampilkan teks atau output sebelumnya}
{F.S. Layar menjadi kosong/bersih dari teks sebelumnya}
procedure cekFormatUsn(input format : string, input user: User)
{Proses: Meminta input username dari pengguna dan mengecek apakah sesuai
format yang diinginkan sistem}
{I.S. user.identitas.username belum terisi}
{F.S. Username tersimpan pada user.identitas.username}
procedure lowercase(input/output s : string)
{Proses: Mengubah seluruh karakter dalam string menjadi huruf kecil}
{I.S. Sebuah input berupa string, mungkin mengandung huruf kapital}
{F.S. String yang telah diubah seluruhnya menjadi huruf kecil/lowercase}

```

ALGORITMA

```

clearScreen
cekFormatUsn(format, user)
valid ← true
i traversal [0..jumlah_user - 1]
    if lowercase(users[i].identitas.username) =
lowercase(user.identitas.username) then
        valid ← false
if not(valid) then
    clearScreen
    output("Registrasi gagal! Pasien dengan nama ", user.identitas.username,
"sudah terdaftar.")
    waitForEnter
    dealokasi(user)
    user ← NULL
    → false
else
    realokasi(users, jumlah_user + 1)
    output("Password: ")
    input(user.identitas.password)
    users[jumlah_user] ← user
    users[jumlah_user].identitas.role ← "pasien"
    jumlah_user ← jumlah_user + 1
    output("Pasien ", user.identitas.username, " berhasil ditambahkan!")
    → true

```

3. Spesifikasi fitur F03 Logout

```

procedure logout(input/output user : pointer)
{Proses: Menghapus data pengguna aktif dari memori dan mengatur ulang pointer user}

```

{I.S. Pointer user menunjuk ke data pengguna yang aktif/sedang login}
 {F.S. Memori yang ditunjuk oleh pointer user telah dibebaskan dan pointer user menjadi nil}

KAMUS LOKAL

user: pointer to User (extern)

ALGORITMA

if user \neq NULL then
 dealokasi (user)
 user \leftarrow NULL

4. Spesifikasi fitur F04 Lupa Password

procedure lupaPassword(input user: User)

{Proses: Meminta username pengguna, mencocokkan dengan daftar pengguna, lalu mengubah password jika ditemukan}

{I.S. Array users berisi daftar semua pengguna yang terdaftar, username pengguna mungkin ada}

{F.S.

- Jika username ditemukan: password pengguna pada array users telah diubah ke nilai baru.
- Jika tidak ditemukan: tidak ada perubahan pada array users.}

KAMUS LOKAL

users: array of User (extern)

user: User

jumlah_user : integer (extern)

i, format : integer

found: boolean

kodeUnik : string

procedure clearScreen()

{Proses : Membersihkan tampilan layar}

{I.S. Layar menampilkan teks atau output sebelumnya}

{F.S. Layar menjadi kosong/bersih dari teks sebelumnya}

procedure cekFormatUsn(input format : string, input user: User)

{Proses: Meminta input username dari pengguna dan mengecek apakah sesuai format yang diinginkan sistem}

{I.S. user.identitas.username belum terisi}

{F.S. Username tersimpan pada user.identitas.username}

procedure lowercase(input/output s : string)

{Proses: Mengubah seluruh karakter dalam string menjadi huruf kecil}

{I.S. Sebuah input berupa string, mungkin mengandung huruf kapital}

{F.S. String yang telah diubah seluruhnya menjadi huruf kecil/lowercase}

procedure runLengthEncode(output output : string)

{Proses: Mengubah string input menjadi representasi run-length encoding dan menyimpannya ke suatu variabel output}

{I.S. input berupa string (tidak kosong), output dalam keadaan kosong}

{F.S. output berisi hasil encoding dari input}

ALGORITMA

```
clearScreen
cekFormatUsn(format,user)
runLengthEncode(username, kodeUnik)
output("Kode Unik: ", kodeUnik)
found ← false
i ← 0
while(i < jumlah_user and not(found))
    runLengthEncode(users[i].identitas.username, validKodeUnik)
    if lowercase(users[i].identitas.username) = lowercase(username) then
        found ← true
    i ← i + 1
{i = jumlah_user or found}
if (found) then
    output("Username ditemukan.")
    output("Masukkan password baru: ")
    input(newPassword)
    users[i].identitas.password ← newPassword
    output("Password berhasil diubah untuk ", username)
else
    output("Tidak ada pengguna yang bernama ", username)
```

5. Spesifikasi fitur F05 Menu & Help

```
procedure help()
{Proses: Menampilkan list command yang bisa dipanggil oleh pengguna
berdasarkan rolenya}
{I.S. sembarang}
{F.S. ditampilkan list command yang dapat dipanggil oleh pengguna}
```

KAMUS LOKAL

role, username : string

ALGORITMA

```
output ("===== HELP =====")
if (role = "") then
    output("Kamu belum login sebagai role apapun. Silahkan login
    terlebih dahulu")
    output("1. LOGIN: Masuk ke dalam akun yang sudah terdaftar")
    output("2. REGISTER: Membuat akun baru")
else if (role = "dokter") then
    output("Halo Dokter ", username, " Kamu memanggil command
    HELP. Kamu pasti sedang kebingungan. Berikut adalah
    hal-hal yang dapat kamu lakukan sekarang : ")
    output("1. LOGOUT: Keluar dari akun yang sedang digunakan")
    output("2. DIAGNOSIS: Melakukan diagnosis penyakit pasien")
```

```

        berdasarkan kondisi tubuh pasien")
    output("3. NGOBATIN: Memberi obat kepada pasien secara
        teratur")
    else if (role = "pasien") then
        output("Selamat datang,", username, "Kamu memanggil
            command HELP. Kamu pasti sedang kebingungan.
            Berikut adalah hal-hal yang dapat kamu lakukan
            sekarang:")
        output("1. LOGOUT: Keluar dari akun yang sedang digunakan")
        output("2. DAFTAR_CHECKUP: Mendaftarkan diri untuk
            pemeriksaan dokter")
        output("3. AKU_BOLEH_PULANG_GA_DOK: Melakukan
            pemeriksaan ulang kepada dokter untuk memastikan
            apakah pasien benar-benar sudah siap untuk pulang")
        output("4. ANTRIAN_SAYA: Melihat status antrian")
        output("5. MINUM_OBAT: Meminum obat yang telah diberikan
            oleh dokter")
        output("6. MINUM_PENAWAR: Mengeluarkan obat terakhir
            yang diminum dan mengembalikannya ke dalam
            inventory obat")
    else if (role = "manager") then
        output("Halo Manager ", username, " Kenapa kamu memanggil
            command HELP? Kan kamu manager, tapi ya sudahlah
            kamu pasti sedang kebingungan. Berikut adalah hal-hal
            yang dapat kamu lakukan sekarang:")
        output("1. LOGOUT: Keluar dari akun yang sedang digunakan")
        output("2. TAMBAH_DOKTER: Mendaftarkan dokter baru ke
            sistem")
        output("3. LIHAT_USER: Melihat data seluruh pengguna
            (dokter dan pasien), atau secara spesifik hanya pasien
            atau hanya dokter")
        output("4. CARI_USER: Mencari data pengguna secara spesifik
            berdasarkan ID atau Nama melalui perintah
            (CARI_USER), maupun secara terfokus melalui
            (CARI_PASIEN) dan (CARI_DOKTER)")
        output("5. LIHAT_ANTRIAN: Melihat rincian di seluruh
            ruangan saat ini. Ruangan yang perlu ditampilkan adalah
            ruangan yang tidak kosong")
    output ("Footnote:")
    output ("1. Untuk menggunakan aplikasi, silahkan masukkan nama
        fungsi yang terdaftar")
    output ("2. Jangan lupa untuk memasukkan input yang valid")

```

6. Spesifikasi fitur F06/D01 Denah Rumah Sakit

procedure tampilkanDenahRumahSakit (input rumahSakit: Hospital)
 {Menampilkan layout denah rumah sakit berdasarkan jumlah baris dan kolom}
 {I.S. ukuran denah terdefinisi}
 {F.S. tampilan layout denah rumah sakit}

KAMUS LOKAL

i, j, index : integer
 labelRuangan : string
function dapatkanLabelRuangan(index, cols: integer) → string
 {Mengubah indeks ruangan menjadi label seperti “A1”, “B2”, dll.}

ALGORITMA

```

j traversal [0..rumahSakit.cols - 1]
  output(j:7)

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

i traversal [0..rumahSakit.rows - 1]
  j traversal [0..rumahSakit.cols - 1]
    index ← i * rumahSakit.cols + j
    labelRuangan ← dapatkanLabelRuangan(index,
    rumahSakit.cols)
    output("| ", labelRuangan:5, " ")
  output("|")
  j traversal [0..rumahSakit.cols - 1]
    output("+-----")
  output("+")
  
```

procedure tampilkanDetailRuangan(input kodeRuangan: string, input rumahSakit: Hospital, input users: array of User, input userCount : integer)
 {Menampilkan detail isi ruangan tertentu, termasuk ID dokter dan pasien}
 {I.S. rumahSakit, users, dan kodeRuangan terdefinisi}
 {F.S. tampilan detail isi ruangan kodeRuangan}

KAMUS LOKAL

i, j, targetIndex: integer
 labelRuangan: string
 ruangan: Room
 namaDokter, namaPasien: string
 adaPasien: boolean
function dapatkanLabelRuangan(index, cols: integer) → string
 {Mengubah indeks ruangan menjadi label seperti “A1”, “B2”, dll.}
function dapatkanUsername(users: array of User; userCount, id: integer; role: string) → string
 {Mengembalikan username pengguna dengan id dan role tertentu. Jika tidak ditemukan, mengembalikan string status}

ALGORITMA

```
targetIndex ← - 1
i traversal [0..rumahSakit.roomCount - 1]
  labelRuangan ←
  dapatkanLabelRuangan(rumahSakit.rooms[i].roomIndex, rumahSakit.cols)
  output("Ruangan ", labelRuangan, ": ")
  if rumahSakit.rooms[i].doctorId = 0 then
    output("Tidak ada dokter, ")
  else
    output("Dokter ID ", rumahSakit.rooms[i].doctorId, ", ")
  adaPasien ← false
  j traversal [0..rumahSakit.rooms[i].patientCount - 1]
    if rumahSakit.rooms[i].patients[j] > 0 then
      adaPasien ← true
      break
  if not(adaPasien) then
    output("Tidak ada pasien")
  else
    output("Pasien: ")
    j traversal [0..rumahSakit.rooms[i].patientCount - 1]
      if rumahSakit.rooms[i].patients[j] > 0 then
        if j > 0 then
          output(", ")
          output(rumahSakit.rooms[i].patients[j])
        if labelRuangan = kodeRuangan then
          targetIndex ← i

if targetIndex = -1 then
  output("Ruangan ", kodeRuangan, " tidak ditemukan.")

ruangan ← rumahSakit.rooms[targetIndex]
output("--- Detail Ruangan ", kodeRuangan, " ---")
output("Kapasitas: ", rumahSakit.maxPatientsPerRoom)

namaDokter ← dapatkanUsername(users, userCount, ruangan.doctorId,
"dokter")

output("Dokter : Dr. ", namaDokter)
output("Pasien di dalam ruangan:")
if ruangan.patientCount = 0 then
  output("Tidak ada pasien di dalam ruangan saat ini.")
else
  i traversal [0..ruangan.patientCount - 1]
    namaPasien ← dapatkanUsername(users, userCount,
ruangan.patients[i], "pasien")
```

```

                                output (i + 1, “. ”, namaPasien)
output(“-----”)

```

7. Spesifikasi fitur F07 Lihat User

function lamanLihatUser (users: array of User; jumlah_user: integer) → integer
 {Menampilkan seluruh data pengguna dari array users dalam format tabel.}

KAMUS LOKAL

i, j, by, order, selectSearchUser: integer
 copy: array of User
 temp: User
 compare: integer
function getValidIntInput(min, max : integer, prompt: string) → integer
 {Meminta input angka bulat dan memastikan nilainya berada dalam rentang [min..max]. Jika tidak valid, akan diminta ulang}

ALGORITMA

```

if jumlah_user = 0 then
    output(“Tidak ada user yang terdaftar”)
    exit

    output (“Urutkan berdasarkan?”)
    output(“1. ID”)
    output(“2. Nama”)
    by ← getValidIntInput(1,2, “>>> Pilihan: ”)

    if by = 1 then
        output(“Urutan sort?”)
        output(“1. ASC (1-100)”)
        output(“2. DESC (100-1)”)
    else
        output(“Urutan sort?”)
        output(“1. ASC (A-Z)”)
        output(“2. DESC (Z-A)”)
    order ← getValidIntInput(1,2, “>>> Pilihan: ”)
    setLength(copy, jumlah_user)
    i traversal [0..jumlah_user - 1]
        copy[i] ← users[i]

    i traversal [0..jumlah_user - 2]
        j traversal [i + 1..jumlah_user - 1]
            if by = 1 then
                compare ← copy[i].identitas.id - copy[j].identitas.id
            else
                compare ← CompareText(copy[i].identitas.username,
                    copy[j].identitas.username)

```

```

    if ((order = 1) and (compare > 0)) or ((order = 2) and (compare <
0)) then
        temp ← copy[i]
        copy[i] ← copy[j]
        copy[j] ← temp
    output("Menampilkan semua pengguna dengan ", IfThen(by = 1, "ID", "Nama"),
    " terurut ", IfThen(order = 1, "ascending", "descending", "..."))
    output("+-----+-----+-----+-----+")
    output("| ID          | Nama          | Role      | Penyakit      |")
    output("+-----+-----+-----+-----+")

i traversal [0..jumlah_user - 1]
    if (uppercase(copy[i].identitas.role) = "PASIEN") and
(length(copy[i].kondisi.riwayat_penyakit) > 0) then
        output("| ", copy[i].identitas.id:10, " | ",
        copy[i].identitas.username:19, " | ",
        copy[i].identitas.role:10, " | ",
        copy[i].kondisi.riwayat_penyakit:18, " |")
    else
        output("| ", copy[i].identitas.id:10, " | ",
        copy[i].identitas.username:19, " | ",
        copy[i].identitas.role:10, " | ", "-:18, " |")
    output("+-----+-----+-----+-----+")

    output("Apakah anda ingin mencari user?")
    output("1. Iya")
    output("2. Tidak")
    selectSearchUser ← getValidIntInput(1, 2, ">>> Pilihan: ")
    if selectSearchUser = 1 then
        → (lamanCariUser(copy, jumlah_user, by))
    else
        exit

```

8. Spesifikasi fitur F08 Cari User

function lamanCariUser(by : integer) → integer

{Menyediakan antarmuka pencarian user berdasarkan ID, nama, atau riwayat penyakit, dan memfilter berdasarkan role. Mengembalikan 1 jika berhasil, 0 jika gagal}

KAMUS LOKAL

selectOpt: integer

searchTerm: string

procedure queryAndPrintUserTable(input by : integer, input searchTerm: string,
input users: array User, input role : string)

{Melakukan pencarian data user dalam array dan mencetak dalam format tabel

berdasarkan kriteria.}

function getValidIntInput(min, max : integer, prompt: string) → integer
{Meminta input angka bulat dan memastikan nilainya berada dalam rentang [min..max]. Jika tidak valid, akan diminta ulang}

ALGORITMA

```
if by = -1 then
    output("Cari berdasarkan?")
    output("1. ID")
    output("2. Nama")
    output("3. Riwayat Penyakit")
    by ← getValidIntInput(1, 3, ">>> Pilihan: ")
```

```
output(">>> Pilih apa yang mau dicari")
output("1. Cari User (Pasien dan Dokter)")
output("2. Cari Pasien")
output("3. Cari Dokter")
selectOpt ← getValidIntInput(1, 3, ">>> Pilihan: ")
```

```
depend on (selectOpt)
selectOpt = 1 :
    output(">>> Masukkan", ifThen(by = 1, "ID", "Nama"), " User: ")
    input(searchTerm)
    queryAndPrintUserTable(by, searchTerm, users, "ALL")
selectOpt = 2 :
    output(">>> Masukkan", ifThen(by = 1, "ID", "Nama"), " Pasien: ")
    input(searchTerm)
    queryAndPrintUserTable(by, searchTerm, users, "Pasien")
selectOpt = 3 :
    output(">>> Masukkan ", ifThen(by = 1, "ID", "Nama"),
        Dokter:")
    input(searchTerm)
    queryAndPrintUserTable(by, searchTerm, users, "Dokter")
else
    output("Pilihan tidak valid!")
    → 0
→ 1
```

9. Spesifikasi fitur F09/D02 Lihat Antrian

procedure tampilkanSemuaAntrianUntukManajer(input rumahSakit: Hospital, input map : Map, input users: array of User, input userCount : integer)
{Proses: Menampilkan seluruh antrian pasien pada setiap ruangan rumah sakit yang memiliki dokter. Prosedur ini menampilkan denah rumah sakit, kemudian untuk setiap ruangan yang berisi dokter, ditampilkan label ruangan dan detail isi ruangan (termasuk

```

antrian pasien))}
{I.S. data rumah sakit, map dokter, dan user terdefinisi.}
{F.S. Di layar ditampilkan:
- Denah rumah sakit.
- Untuk setiap ruangan yang memiliki dokter:
    - Label ruangan (misalnya A1, B2).
    - ID dokter (jika ditemukan di map).
    - Detail ruangan, termasuk antrian pasien di ruangan tersebut.
- Informasi "[SELESAI] Semua ruangan telah ditampilkan." muncul di akhir
  tampilan.}

```

KAMUS LOKAL

```

i: integer
room: Room
labelRuangan: string
dokter: Dokter
procedure getDoctor(input map : Map, input id : integer)
{Proses: Mengembalikan pointer ke data dokter dari peta berdasarkan ID.}
{I.S. Map sudah terisi data dokter.}
{F.S. Jika ditemukan, pointer ke dokter dikembalikan. Jika tidak, nil
dikembalikan.}
procedure tampilkanDenahRumahSakit (input rumahSakit: Hospital)
{Menampilkan layout denah rumah sakit berdasarkan jumlah baris dan kolom}
{I.S. ukuran denah terdefinisi}
{F.S. tampilan layout denah rumah sakit}
procedure tampilkanDetailRuangan(input kodeRuangan: string, input
rumahSakit: Hospital, input users: array of User, input userCount : integer)
{Menampilkan detail isi ruangan tertentu, termasuk ID dokter dan pasien}
{I.S. rumahSakit, users, dan kodeRuangan terdefinisi}
{F.S. tampilan detail isi ruangan kodeRuangan}

```

ALGORITMA

```

output("==== SEMUA ANTRIAN DI RUMAH SAKIT =====")
tampilkanDenahRumahSakit(rumahSakit)

i traversal [0..rumahSakit.roomCount - 1]
    room ← rumahSakit.rooms[i]
    if room.doctorId = -1 then
        continue
    labelRuangan ← dapatkanLabelRuangan(room.roomIndex,
        rumahSakit.cols)
    dokter ← getDoctor(map, room.doctorId)

    if dokter = NULL then
        output("=== Ruangan “, labelRuangan, “ (Dokter tidak ada)
            ===")

```

```

        tampilkanDetailRuangan(rumahSakit, map, labelRuangan, users,
                                userCount)
        continue
        output("== Ruangan ", labelRuangan, " (Dokter ID: ', dokter.id, ")
        ==")
        tampilkanDetailRuangan(rumahSakit, map, labelRuangan, users,
                                userCount)
        output("[SELESAI] Semua ruangan telah ditampilkan.")

```

10. Spesifikasi fitur F10 Tambah Dokter

function lamanTambahDokter()

{Menambahkan akun dokter baru ke dalam sistem jika username belum pernah digunakan dan memenuhi format}

KAMUS LOKAL

temp_user: User

i, format, idBaru: integer

valid: boolean

passwordInput: string

ALGORITMA

valid ← true

cekFormatUsn(format, temp_user)

i traversal [0..jumlah_user - 1]

if lowercase(users[i].identitas.username) =

lowercase(temp_user.identitas.username) then

valid ← false

break

if not(valid)then

output("Registrasi gagal! dokter dengan nama ",

temp_user.identitas.username, " sudah terdaftar.")

→ 0

output("Password: ")

input(passwordInput)

temp_user.identitas.password ← passwordInput

temp_user.identitas.role ← "dokter"

idBaru ← idTertinggi() + 1

temp_user.identitas.id ← idBaru

jumlah_user ← jumlah_user + 1

users[jumlah_user] ← temp_user

output("Dokter ", temp_user.identitas.username, " berhasil ditambahkan!")

→ 1

11. Spesifikasi fitur F11 Diagnosis

function lamanDiagnosis() → integer

{Mengakses variabel global (user, map, dokter, pasien, penyakit) dan memberikan output 1 jika proses berhasil dijalankan, 0 jika gagal karena syarat tidak terpenuhi.}

{Output lain:

- Menampilkan daftar pasien pada antrian dokter ke layar.
- Mencetak hasil diagnosis.
- Menghapus pasien dari antrian jika terdiagnosa suatu penyakit.
- Mengurangi jumlah pasien dalam/luar ruangan dokter sesuai kondisi.

}

KAMUS LOKAL

dokter : Dokter

pasien : Pasien

current : Node

found : boolean

i, selectPatient: integer

procedure printQueuePositions(input q : Queue)

{Proses: Menampilkan seluruh pasien dalam queue antrian dokter beserta urutan posisinya.}

{I.S. Queue mungkin kosong atau berisi pasien.}

{F.S. Jika tidak kosong, seluruh isi queue ditampilkan dalam format terurut.}

procedure getDoctor(input map : Map, input id : integer)

{Proses: Mengembalikan pointer ke data dokter dari peta berdasarkan ID.}

{I.S. Map sudah terisi data dokter.}

{F.S. Jika ditemukan, pointer ke dokter dikembalikan. Jika tidak, nil dikembalikan.}

function getValidIntInput(minValue, maxValue : integer, prompt : string)

→ integer

{Meminta input dari pengguna dalam rentang tertentu, dan mengembalikan input valid}

procedure dequeue(input/output q : Queue)

{Proses: Menghapus elemen pertama dari queue}

{I.S. Queue terdefinisi, mungkin kosong}

{F.S. Pasien paling depan dihapus dan queue diperbaharui (tidak melakukan apapun jika queue kosong)}

ALGORITMA

if (user = NULL) or (user.identitas.role = “”) then

output(“Anda belum login!”)

exit

if (user.identitas.role ≠ “dokter”) then

output(“Hanya dokter yang dapat mengakses laman ini!”)

exit

if (penyakit = NULL) or (jumlah_penyakit = 0) then


```

        output("Data penyakit tidak ditemukan!")
        exit
    output("Pilih pasien yang ingin didiagnosis: ")
    dokter ← getDoctor(map, user.identitas.id)
    if (dokter = NULL) then
        output("Dokter tidak ditemukan!")
        exit

printQueuePositions(dokter.queue)

output("Pilih pasien (1-", dokter.queue.size, "): ")
selectPatient ← getValidInput(1, dokter.queue.size, ">>> Pilihan: ")

current ← dokter.queue.front
i traversal [0..selectPatient - 1]
    current ← current↑.next
pasien ← current↑.patient
output("Diagnosa untuk pasien ", pasien.nama, ":")
output("[AUTO DIAGNOSE 2.0] Mendiagnosis...")
output("Hasil diagnosa: ")

found ← false
i traversal [0..jumlah_penyakit-1] do
    if (pasien.suhu_tubuh ≥ penyakit[i].suhu_tubuh_min) and
        (pasien.suhu_tubuh ≤ penyakit[i].suhu_tubuh_max) and
        (pasien.tekanan_darah_sistolik ≥
        penyakit[i].tekanan_darah_sistolik_min) and
        (pasien.tekanan_darah_sistolik ≤
        penyakit[i].tekanan_darah_sistolik_max) and
        (pasien.tekanan_darah_diastolik ≥
        penyakit[i].tekanan_darah_diastolik_min) and
        (pasien.tekanan_darah_diastolik ≤
        penyakit[i].tekanan_darah_diastolik_max) and
        (pasien.detak_jantung ≥ penyakit[i].detak_jantung_min) and
        (pasien.detak_jantung ≤ penyakit[i].detak_jantung_max) and
        (pasien.saturasi_oksigen ≥ penyakit[i].saturasi_oksigen_min) and
        (pasien.saturasi_oksigen ≤ penyakit[i].saturasi_oksigen_max) and
        (pasien.kadar_gula_darah ≥ penyakit[i].kadar_gula_darah_min) and
        (pasien.kadar_gula_darah ≤ penyakit[i].kadar_gula_darah_max) and
        (pasien.berat_badan ≥ penyakit[i].berat_badan_min) and
        (pasien.berat_badan ≤ penyakit[i].berat_badan_max) and
        (pasien.tinggi_badan ≥ penyakit[i].tinggi_badan_min) and
        (pasien.tinggi_badan ≤ penyakit[i].tinggi_badan_max) and
        (pasien.kadar_kolesterol ≥ penyakit[i].kadar_kolesterol_min) and

```

```

        (pasien.kadar_kolesterol ≤ penyakit[i].kadar_kolesterol_max) and
        (pasien.trombosit ≥ penyakit[i].trombosit_min) and
        (pasien.trombosit ≤ penyakit[i].trombosit_max) then
            output(pasien.nama, “ terdiagnosa penyakit: ”,
        penyakit[i].nama)
            found ← true
            dequeue(dokter.queue)
            dokter.queueLength ← dokter.queueLength - 1

            if (dokter.jumlahPasienDalamRuangan > 0) then
                dokter.jumlahPasienDalamRuangan ←
        dokter.jumlahPasienDalamRuangan - 1
            else
                dokter.jumlahPasienLuarRuangan ←
        dokter.jumlahPasienLuarRuangan - 1

            output(“Pasien ”, pasien.nama, “ telah keluar dari antrian.”)

if not(found) then
    output(“Tidak ada penyakit yang terdiagnosa untuk pasien ”,
        pasien.nama)
→ 1

```

12. Spesifikasi fitur F12 Ngobatin

function lamanNgobatin() → integer

{Menjalankan proses pengobatan otomatis terhadap pasien pertama di antrian dokter.
Fitur khusus dokter. Fungsi akan mencetak nama pasien dan menginformasikan jika ada riwayat penyakit untuk kemudian digunakan dalam mencocokkan obat yang tepat.
Memberikan output 1 jika proses berhasil dijalankan, output 0 jika gagal(karena belum login, bukan dokter, atau data tidak ditemukan)}

KAMUS LOKAL

user : User (extern)

jumlah_obatpenyakit : integer (extern)

obatpenyakit : string (extern)

dokter: Dokter

pasien: Pasien

procedure getDoctor(input map : Map, input id : integer)

{Proses: Mengembalikan pointer ke data dokter dari peta berdasarkan ID.}

{I.S. Map sudah terisi data dokter.}

{F.S. Jika ditemukan, pointer ke dokter dikembalikan. Jika tidak, nil dikembalikan.}

ALGORITMA

if(user = NULL) or (user.identitas.role = “”) then

output(“Anda belum login!”)

```

    → 0
    if (user.identitas.role ≠ “dokter”) then
        output(“Hanya dokter yang dapat mengakses laman ini!”)
        → 0
    if(obatpenyakit = NULL) or (jumlah_obatpenyakit = 0) then
        output(“Data obat-penyakit tidak ditemukan!”)
        → 0
    dokter ← getDoctor(map, user.identitas.id)
    if (dokter = NULL) then
        output(“Dokter tidak ditemukan!”)
        → 0

    pasien ← dokter.queue.front.patient
    output(“Dokter sedang mengobati pasien ”, pasien.nama, “.”)
    output(“[Auto-Prescription-2.0] Mengobati...”)

    if pasien.riwayat_penyakit = “” then
        output(“Pasien ”, pasien.nama, “ tidak memiliki riwayat penyakit.”)
        → 0

    i traversal [0..jumlah_obatpenyakit - 1]
        if (obatpenyakit[i].penyakit = pasien.riwayat_penyakit) then
            output(“Obat yang harus diberikan:”)
            j ← 0
            while(obatpenyakit[i].obat[j] ≠ “”) do
                output(“j+1, “. ”, obatpenyakit[i].obat[j]”)
                j ← j + 1
            {obatpenyakit[i].obat[j] = “”}
        output(“Obat dari penyakit ”, pasien.riwayat_penyakit, “ tidak ditemukan.”)
        → 1

```

13. Spesifikasi fitur F13 Aku boleh pulang ga dok

procedure pulangdok()

{Melakukan pengecekan terhadap diagnosis dan kesesuaian urutan peminuman obat.}

KAMUS LOKAL

i, j, : integer

equal : boolean = true

current : Queue

dokterName : string

labelRuangan: string

urutanMinumObat: StackObat

obatExpected, obatMinum : Obat

ALGORITMA

```

if map = NULL then
    output("Map tidak ditemukan!")
    exit
i traversal [0..map.size - 1]
    if map.dokter[i].queue ≠ NULL then
        current ← map.dokter[i].queue.front
        while current ≠ nil do
            if current.patient.id = user.identitas.id then
                user.kondisi.riwayat_penyakit ←
                    current.patient.riwayat_penyakit
                break
            current ← current↑.next
if length(user.kondisi.riwayat_penyakit) = 0 then
    output("Kamu belum menerima diagnosis apapun dari dokter, jangan
    buru-buru pulang!")
    exit
output("Dokter sedang memeriksa keadaanmu...")
urutanMinumObat ← getUrutanObat(user.kondisi.riwayat_penyakit)

if urutanMinumObat.top > user.kondisi.StackPerut.top then
    output("Masih ada obat yang belum kamu habiskan, minum semuanya
    dulu yukk!")
    exit

if urutanMinumObat.top ≠ user.kondisi.StackPerut.top then
    equal ← false

i traversal[0..user.kondisi.StackPerut.top]
    obatExpected ← urutanMinumObat.items[i]
    obatMinum ← user^.kondisi.StackPerut.items[i]
    if (obatExpected.id ≠ obatMinum.id) then
        equal ← false
        break
if not(equal) then
    output("Maaf, tapi kamu masih belum bisa pulang!")
    output("Urutan peminuman obat yang diharapkan:")
    i traversal [0..urutanMinumObat.top]
        output(urutanMinumObat.items[i].nama)
        if i < urutanMinumObat.top then
            output(" → ")

    output("Urutan obat yang kamu minum:")
    i traversal [0..user.kondisi.StackPerut.top]
        if urutanMinumObat.items[i].id ≠

```

```

        user.kondisi.StackPerut.items[i].id then
            output('*')
        output(user^.kondisi.StackPerut.items[i].nama)
        if i < user.kondisi.StackPerut.top then
            output(" → ")
        output("Silahkan kunjungi dokter untuk meminta penawar yang sesuai")
        exit
    output("Selamat! Kamu sudah dinyatakan sembuh oleh dokter. Silahkan pulang
    dan semoga sehat selalu!")
    kosongkanKondisiUser(user)

```

14. Spesifikasi fitur F14 Daftar Check-Up

procedure daftarCheckup()

{Proses: Mendaftarkan check-up pasien ke dokter, jika berhasil maka data kondisi pasien akan disimpan dan pasien dimasukkan ke dalam antrian dokter yang dipilih. Data ruangan juga diperbarui.}

{I.S. user sudah login sebagai pasien, mungkin sudah berada dalam antrian, denah rumah sakit, data user aktif, data semua user, dan data rumah sakit sudah terdefinisi}

{F.S.

- Data kondisi pasien tersimpan di data user dan diperbarui di array `users`.
- Pasien masuk ke antrian dokter yang dipilih (dalam atau luar ruangan tergantung ketersediaan).
- Data ruangan pada `rumahSakit` diperbarui untuk mencatat ID pasien.
- Informasi keberhasilan pendaftaran ditampilkan ke layar.

}

KAMUS LOKAL

suhu, oksigen, berat, tinggi : real

sistol, diastol, bpm, gula, kolesterol, trombosit: integer

i, j, pilihan, dokterIdTerpilih, pasienId: integer

dokterName, labelRuangan: string

dokterValid : User

dokterList : Map

room : Room

function isSudahNgantri(map: Map; user: User) → integer

{Mengecek apakah pasien sudah berada dalam antrian manapun}

procedure validateData(input head, desc: string, output data: integer)

{Meminta input angka bulat positif hingga nilai valid dimasukkan.}

{I.S. data sembarang, pengguna diminta menginput angka bulat positif}

{F.S. data diisi dengan angka bulat positif valid dari pengguna}

procedure validatefData(input head, desc: string, output data: real)

{Meminta input angka riil positif hingga nilai valid dimasukkan.}

{I.S. data sembarang}

{F.S. data diisi dengan angka riil valid dan positif}

procedure validate2Data(input head, desc: string, output data1, data2: integer)

{Meminta input dua buah bilangan bulat positif dari pengguna dalam satu baris, dipisahkan dengan spasi. Contoh: `120 80`.}

{I.S. data1 dan data2 sembarang}

{F.S.data1 dan data2 terisi dengan dua bilangan bulat positif yang valid dari pengguna. }

function getAvailableDokter(map: Map) → Map

{Mengembalikan daftar dokter yang masih memiliki slot antrian}

procedure dapatkanLabelRuangan(input index, cols: integer, output label: string)

{Mengonversi indeks ruangan (dalam bentuk integer) ke label ruangan berdasarkan jumlah kolom map.}

{I.S. label sembarang}

{F.S. label terisi dengan nama ruangan berbentuk huruf dan angka}

procedure enqueue(input/output queue: Queue, input pasien: User)

{Proses: Menambahkan elemen pasien ke queue}

{I.S. pasien dan queue terdefinisi}

{F.S. Pasien ditambahkan ke paling belakang antrian}

procedure lowercase(input/output s : string)

{Proses: Mengubah seluruh karakter dalam string menjadi huruf kecil}

{I.S. Sebuah input berupa string, mungkin mengandung huruf kapital}

{F.S. String yang telah diubah seluruhnya menjadi huruf kecil/lowercase}

ALGORITMA

if isSudahNgantri(map, user) = 1 then

exit

output (“Silahkan masukkan data check-up Anda: ”)

validatefData(“Suhu Tubuh”, “Celcius”, suhu)

validate2Data(“Tekanan Darah”, “sistol/diastol, contoh 120 80”, sistol, diastol)

validateData(“Detak Jantung”, “bpm”, bpm)

validatefData(“Saturasi Oksigen”, “%”, oksigen)

validateData(“Kadar Gula Darah”, “mg/dL”, gula)

validatefData(“Berat Badan”, “kg”, berat)

validatefData(“Tinggi Badan”, “cm”, tinggi)

validateData(“Kadar Kolesterol”, “mg/dL”, kolesterol)

validateData(“Trombosit”, “ribu/mikroL”, trombosit)

output(“Berikut adalah daftar dokter yang tersedia: ”)

dokterList ← getAvailableDokter(map)

i traversal [0..dokterList.size - 1]

dokterIdTerpilih ← dokterList.dokter[i].id

dokterName ← “DOKTER TIDAK DITEMUKAN”

j traversal [0..jumlah_user - 1]

if lowercase(users[j].identitas.role) = “dokter” and

users[j].identitas.id = dokterIdTerpilih then

```

        dokterName ← users[j].identitas.username
        dapatkanLabelRuangan(dokterList.dokter[i].ruangan, map.cols,
                               labelRuangan)
        output(i + 1, “. dr. ”, dokterName, “ - Ruangan ”, labelRuangan,
               “(Antrian: “, dokterList.dokter[i].jumlahPasienDalamRuangan +
               dokterList.dokter[i].jumlahPasienLuarRuangan, “ orang)”)

    repeat
        output(“Pilih dokter (1-”, dokterList.size, “): “)
        input(pilihan)
        if(pilihan < 1) or (pilihan > dokterList.size) then
            output(“Pilihan di luar rentang!”)
    until (pilihan ≥ 1) and (pilihan ≤ dokterList.size)

    dokterIdTerpilih ← dokterList.dokter[pilihan - 1].id
    dokterValid ← NULL

    j traversal [0..jumlah_user - 1]
        if lowercase(users[j].identitas.roles) = “dokter” and users[j].identitas.id =
        dokterIdTerpilih then
            dokterValid ← users[j]
            break
    if dokterValid = NULL then
        output(“Dokter tidak ditemukan di sistem. Pendaftaran dibatalkan.”)
        exit

    user.kondisi.suhu_tubuh ← suhu
    user.kondisi.tekanan_darah_sistolik ← sistol
    user.kondisi.tekanan_darah_diastolik ← diastol
    user.kondisi.detak_jantung ← bpm
    user.kondisi.saturasi_oksigen ← oksigen
    user.kondisi.kadar_gula_darah ← gula
    user.kondisi.berat_badan ← berat
    user.kondisi.tinggi_badan ← tinggi
    user.kondisi.kadar_kolesterol ← kolestrol
    user.kondisi.trombosit ← trombosit

    i traversal [0..jumlah_user - 1]
        if users[i].identitas.id = user.identitas.id then
            users[i] ← user
            break
    dokterTerpilih ← dokterList.dokter[pilihan - 1]
    dokterTerpilih.queueLength ← dokterTerpilih.queueLength + 1

```

```

if dokterTerpilih.jumlahPasienDalamRuangan < map.maxPasienDalamRuangan
then
    dokterTerpilih.jumlahPasienDalamRuangan ←
dokterTerpilih.jumlahPasienDalamRuangan + 1
else
    dokterTerpilih.jumlahPasienLuarRuangan ←
dokterTerpilih.jumlahPasienLuarRuangan + 1

enqueue(dokterTerpilih.queue, user)
pasienId ← user.identitas.id

i traversal [0..rumahSakit.roomCount - 1]
    if rumahSakit.rooms[i].roomIndex = dokterTerpilih.ruangan then
        room ← rumahSakit.rooms[i]
        if room.patientCount < MAX_PATIENTS then
            room.patients[room.patientCount] ← pasienId
            room.patientCount ← room.patientCount + 1
        else
            output("Ruangan penuh, tidak bisa update Hospital.")
            break
    dapatkanLabelRuangan(dokterTerpilih.ruangan, map.cols, labelRuangan)
    output("Pendaftaran check-up berhasil!")
    output("Anda terdaftar pada antrian dokter: ", dokterValid.identitas.username, ",
di ruangan: ", labelRuangan)
    output("Posisi antrian Anda: ", dokterTerpilih.queue.size)

```

15. Spesifikasi fitur F15 Antrian Saya!

function lamanLihatAntrianSaya() → integer
{Menampilkan status check-up user aktif}

KAMUS LOKAL

i, j, k : integer
labelRuangan: string
dokterName: string
found : boolean
current: Queue

ALGORITMA

```

found ← false
i traversal [0..rumahSakit.roomCount - 1]
    j traversal [0..rumahSakit.rooms[i].patientCount - 1]
        if rumahSakit.rooms[i].patients[j] = user.identitas.id then
            dapatkanLabelRuangan(rumahSakit.rooms[i].roomIndex,
rumahSakit.cols, labelRuangan)
            dokterName ← "DOKTER TIDAK DITEMUKAN"

            if rumahSakit.rooms[i].doctorId ≠ 0 then
                k traversal [0..Length(users) - 1]

```



```

        if (users[k].identitas.id =
rumahSakit.rooms[i].doctorId) and lowercase(users[k].identitas.role) =
"dokter" then
            dokterName ← users[k].identitas.username
            break
        output("Anda sedang berada di dalam ruangan
            dokter!")
        output("Ruangan: ", labelRuangan)
        output("Dokter: ", dokterName)
        → 1
i traversal [0..map.size - 1]
    if map.dokter[i].queue ≠ NULL then
        current ← map.dokter[i].queue.front
        while current ≠ NULL do
            if current.patient.identitas.id = user.identitas.id then
                dokterName ← "DOKTER TIDAK
                    DITEMUKAN"
                j traversal [0..length(users) - 1]
                if lowercase(users[j].identitas.role)
                    = "dokter" and users[j].identitas.id
                    = map.dokter[i].id then
                    dokterName ←
                        users[j].identitas.username
                    break
                dapatkanLabelRuangan(
                    map.dokter[i].ruangan, map.cols,
                    labelRuangan)
                output("STATUS ANTRIAN ANDA:")
                output("NAMA: ", user.identitas.username)
                output("DOKTER: ", dokterName)
                output("RUANGAN: ", labelRuangan)
                output("POSISI ANTRIAN: ",
                    getQueuePosition(map.dokter[i].queue,
                    user.identitas.id), " dari ",
                    map.dokter[i].queue.rear.patient.identit
                    as.id)
                found ← true
                break
            current ← current↑.next

    if not(found) then
        output("Anda belum terdaftar dalam antrian check-up!")
        output("Silakan Daftar Check-Up terlebih dahulu.")
    → 1

```

16. Spesifikasi fitur F16 Minum Obat

procedure lamanMinumObat()

{Proses: Menampilkan daftar obat dalam inventory user, menerima input pilihan obat yang ingin diminum, memindahkannya ke stack perut, dan menghapus obat tersebut dari inventory user}

KAMUS LOKAL

user : User (extern)

obatTerpilih : string

pilihan, i, idx : integer

procedure tampilkanDaftarObat(input inventory: array of Obat, input jumlahObat: integer)

{Proses: Menampilkan daftar obat yang ada dalam inventory ke layar secara terformat.}

{I.S. Inventory obat berisi sejumlah obat yang valid.}

{F.S. Seluruh nama obat dalam array inventory ditampilkan ke layar.}

procedure tambahObatDalamPerut(output perut: StackPerut)

{Proses: Menambahkan obat ke struktur stack "perut" pasien sesuai ID. Jika belum ada stack, maka dialokasikan.}

{I.S. Stack perut user bisa kosong atau terisi.}

{F.S. Obat 'obatTerpilih' berada di atas stack perut user. Stack diperluas bila perlu.}

ALGORITMA

if (user.kondisi.jumlahObat = 0) then

output("Inventory obat kamu kosong!")

exit

tampilkanDaftarObat(user.kondisi.inventory, user.kondisi.jumlahObat)

output("Pilih obat untuk diminum ")

i traversal [0..user.kondisi.jumlahObat - 1]

output(i+1, ". ", user.kondisi.inventory[i].nama)

input(pilihan)

if(pilihan < 1) or (pilihan > user.kondisi.jumlahObat) then

output("Pilihan nomor tidak tersedia!")

exit

idx ← pilihan - 1

obatTerpilih ← user.kondisi.inventory[idx]

if not(tambahObatDalamPerut(user.id, obatTerpilih)) then

output("Gagal menyimpan obat ke perut!")

exit

i ← idx

while(i < user.kondisi.jumlahObat - 1)

user.kondisi.inventory[i] ← user.kondisi.inventory[i+1]

i ← i + 1

{i = user.kondisi.jumlahObat - 1}

user.kondisi.jumlahObat ← user.kondisi.jumlahObat - 1

output("GLEKGLEKGLEK...", obatTerpilih.nama, " berhasil diminum!!!")

if(user.kondisi.jumlahObat > 0) then

output("Sisa obat di inventory: ")

tampilkanDaftarObat(user.kondisi.inventory, user.kondisi.jumlahObat)

else

output("Inventory obat kamu sekarang kosong!")

17. Spesifikasi fitur F17 Minum Penawar

procedure minumPenawar(input currentUser : User)

{ Proses: Mengeluarkan obat terakhir yang telah dikonsumsi pasien dan mengembalikannya ke `inventory` pasien jika masih ada ruang. Hanya dapat dilakukan oleh pengguna dengan role "pasien". }

{ I.S.

- currentUser sudah terdefinisi.

- Jika currentUser.role = "pasien", maka mungkin terdapat obat pada stack `perut`.

- `inventory` pasien mungkin sudah penuh (jumlahObat = MAX_OBAT) atau masih tersedia ruang. }

{ F.S.

- Jika currentUser bukan pasien:

- Tidak ada perubahan pada perut maupun inventory.

- Pesan "Fitur ini hanya tersedia untuk pasien!" ditampilkan.

- Jika perut kosong:

- Tidak ada perubahan pada inventory.

- Pesan "Perut kosong!! Belum ada obat yang dimakan." ditampilkan.

- Jika ada obat di perut:

- Obat terakhir (top stack) dikeluarkan dari perut.

- Jika inventory belum penuh:

- Obat ditambahkan ke inventory.

- jumlahObat bertambah 1.

- Pesan "Uwekkk!!! <nama obat> keluar dan kembali ke inventory" ditampilkan. }

- Jika inventory penuh:

- Obat dikembalikan lagi ke stack perut (tidak jadi keluar).

- Stack perut kembali ke keadaan semula.

- Pesan "Inventory penuh! tidak bisa mengeluarkan obat" ditampilkan. }

KAMUS LOKAL

keluar: Obat

procedure lowercase(input/output s : string)

{Proses: Mengubah seluruh karakter dalam string menjadi huruf kecil}

{I.S. Sebuah input berupa string, mungkin mengandung huruf kapital}

{F.S. String yang telah diubah seluruhnya menjadi huruf kecil/lowercase}

ALGORITMA

```

if lowercase(currentUser.identitas.role) ≠ “pasien” then
    output(“Fitur ini hanya tersedia untuk pasien!”)
    exit
if currentUser.kondisi.perut.top = -1 then
    output(“Perut kosong!! Belum ada obat yang dimakan.”)
    exit

keluar ← currentUser.kondisi.perut.items[currentUser.kondisi.perut.top]
currentUser.kondisi.perut.top ← currentUser.kondisi.perut.top - 1

if currentUser.kondisi.jumlahObat < MAX_OBAT then
    currentUser.kondisi.inventory[currentUser.kondisi.jumlahObat] ← keluar
    currentUser.kondisi.jumlahObat ← currentUser.kondisi.jumlahObat - 1
    output('Uwekkk!!! ', keluar.nama, ' keluar dan kembali ke inventory')
else
    output(“Inventory penuh! tidak bisa mengeluarkan obat”)
    currentUser.kondisi.perut.top ← currentUser.kondisi.perut.top + 1
    currentUser.kondisi.perut.items[currentUser.kondisi.perut.top] ← keluar

```

18. Spesifikasi fitur F18 Exit

```

procedure keluarSistem()
{Keluar dari sistem, memberi opsi apakah user ingin menyimpan data atau tidak}
{I.S. user masih di dalam sistem, sistem masih berjalan}
{F.S. sistem dihentikan, user keluar dari sistem}

```

KAMUS LOKAL

```

c : char
folder_name : string
procedure save()
{Proses: Menyimpan data rumah sakit secara permanen}
{I.S. data masih disimpan di memori}
{F.S.
    - Jika folder sudah ada : data disimpan di dalam folder
    - Jika folder belum ada : folder baru dibuat dan data disimpan di dalamnya}

```

ALGORITMA

```

output(“Apakah Anda mau melakukan penyimpanan file yang sudah diubah?
(y/n)”)
input(c)
if(c = ‘y’) or (c = ‘Y’) then
    output(“Masukkan nama folder penyimpanan : ”)
    input(folder_name)
    save(folder_name, folder)
output(“Terima kasih telah menggunakan sistem! <3”)
if user then

```

```

        free(user)
    if users then
        free(users)
    if penyakits then
        free(penyakits)
    if obats then
        free(obats)
    if map then
        free(map)
    if rumahSakit then
        free(rumahSakit)

```

19. Spesifikasi fitur D03 Load

procedure load()

{Proses: Memuat data yang diperlukan ke dalam sistem}

{I.S. Data belum dimuat ke dalam sistem}

{F.S. Data berhasil dimuat ke dalam sistem}

KAMUS LOKAL

userpath, penyakitpath, folderpath : string

jumlah_user, jumlah_penyakit, jumlah_obat : integer (extern)

st : struct stat

pt : ParseTarget

pp : ParsePenyakit

po : ParseObat

tempUser : pointer to User

tempPenyakit : pointer to Penyakit

tempObat : pointer to Obat

procedure CSVtoArr(input filename : string, input/output target : void)

{Proses: Membaca file CSV berformat baris dan kolom terpisah `;`, lalu memanggil handler untuk setiap baris data (tanpa header) dan meneruskan hasil parsing ke handler tersebut.}

{I.S. File CSV dengan format field terpisah `;` tersedia dan dapat dibuka, handler dan target terdefinisi.}

{F.S. Semua baris setelah header diproses dengan handler(fields, field_count, target). Target dapat dimodifikasi oleh handler.}

procedure loadConfig()

{Proses: Membaca file config sistem rumah sakit dari file teks berformat baris per baris ke struktur data global.}

{I.S. File config tersedia dan dapat dibuka, struktur data global telah dialokasikan.}

{ F.S. Struktur data global berhasil dimuat sesuai isi file.}

function muatDataRumahSakit(namaFile : file, rumahSakit : Hospital) →

boolean

{Menerima masukan dari file, memberikan output false jika data rumah sakit gagal dimuat, true jika berhasil}

ALGORITMA

```
userpath ← "data/folder_name/user.csv"
penyakitpath ← "data/folder_name/penyakit.csv"
folderpath ← "data/folder_name"

{cek apakah folder ada}
if stat(folderpath, st) ≠ 0 or not (isDirectory(st.mode)) then
    output("Folder ", folder_name, " tidak ditemukan di dalam folder data")
    exit
jumlah_user ← 0
jumlah_penyakit ← 0
pt.data_user ← users
pt.jumlah_user ← jumlah_user
pp.data_penyakit ← penyakits
pp.jumlah_penyakit ← jumlah_penyakit
po.obats ← obats
po.jumlah_obat ← jumlah_obat

CSVtoArr(userpath, handleUserRow, pt)
CSVtoArr(penyakitpath, handlePenyakitRow, pp)
CSVtoArr(obatpath, handleObatRow, po)

if (jumlah_user > 0 )then
    realokasi (tempUser, jumlah_user)
    if(tempUsers) then
        users ← tempUser
if (jumlah_penyakit > 0 )then
    realokasi (tempPenyakit, jumlah_Penyakit)
    if(tempPenyakit) then
        penyakit ← tempPenyakit
if (jumlah_obat > 0 )then
    realokasi (tempObat, jumlah_obat)
    if(tempObat) then
        obats ← tempObat
loadConfig(configpath)
muatDataRumahSakit(configpath, rumahSakit)
```

20. Spesifikasi fitur D04 Save

procedure save()

{Proses: Menyimpan data rumah sakit secara permanen}

{I.S. data masih disimpan di memori}

{F.S.

- Jika folder sudah ada : data disimpan di dalam folder
- Jika folder belum ada : folder baru dibuat dan data disimpan di dalamnya}

KAMUS LOKAL

base_path, full_path, test_file : string
userfile : SEQFILE of (*)string
test: SEQFILE of (*)string
jumlah_user : integer (extern)

ALGORITMA

```
base_path ← "data"
full_path ← "data/folder_name"
test_file ← "data/folder_name/user.csv"
assign (testfile,"test")
if (test = NULL) then
    output ("Membuat folder...")
    make_folder(full_path)
else
    close test
    output ("Saving...")

    rewrite(userfile)
    if (userfile ≠ NULL) then
        write(userfile,
            users[i].identitas.username,
            users[i].identitas.password,
            users[i].identitas.role,
            users[i].kondisi.riwayat_penyakit,
            users[i].kondisi.suhu_tubuh,
            users[i].kondisi.tekanan_darah_sistolik,
            users[i].kondisi.tekanan_darah_diastolik,
            users[i].kondisi.detak_jantung,
            users[i].kondisi.saturasi_oksigen,
            users[i].kondisi.kadar_gula_darah,
            users[i].kondisi.berat_badan,
            users[i].kondisi.tinggi_badan,
            users[i].kondisi.kadar_kolesterol,
            users[i].kondisi.trombosit, i + 1)
        i traversal [0..jumlah_user-1]
        write(userfile,data user dalam format CSV)
        close userfile
        output ("Berhasil menyimpan data di folder ", full_path, "!")
    else
        output ("Gagal menyimpan file di ", full_path)
```

H. Hasil Pengujian Program

Load folder sesuai keinginan user, jika valid:

```
nullmhps@null:/mnt/c/Users/Wahyu Sulastianto/OneDrive/Documents/BELAJAR CODING/if1210-tubes-2025-k01-n$ make run ARGS=original
```

Gambar 22. Load, awal folder

Jika tidak ada nama folder::

```
nullmhps@null:/mnt/c/Users/Wahyu Sulastianto/OneDrive/Documents/BELAJAR CODING/if1210-tubes-2025-k01-n$ make run  
./main  
Tidak ada nama folder yang diberikan!  
Usage : ./main <<nama_folder>>  
make: *** [makefile:29: run] Error 1
```

Gambar 23. Load, tidak ada nama folder yang diinput

Jika tidak ditemukan:

```
nullmhps@null:/mnt/c/Users/Wahyu Sulastianto/OneDrive/Documents/BELAJAR CODING/if1210-tubes-2025-k01-n$ make run ARGS=origina  
./main origina  
Folder "origina" tidak ditemukan di dalam folder data/.  
make: *** [makefile:29: run] Error 1
```

Gambar 24. Load, tidak ada folder yang sesuai

Halaman utama

```
=====
                        RUMAH SAKIT K01-N
=====
1. Login
2. Register
3. LupaPassword
4. Exit

>>> Masukkan pilihan (1-4): 1
```

Gambar 25. HalamanUtama, awal masuk

Jika Input tidak valid:


```
=====
                RUMAH SAKIT K01-N
=====
1. Login
2. Register
3. LupaPassword
4. Exit

>>> Masukkan pilihan (1-4): f

Input tidak valid! Masukkan angka antara 1-4.
Silahkan enter untuk mengulang!
█
```

Gambar 26. HalamanUtama, input salah

Jika input 1, masuk ke login

```
Username: █
```

Gambar 27. login , awal masuk

Jika username/password salah

```
Username: f
Password: g

Tidak ada Manager, Dokter, atau pun Pasien yang bernama f!

Tekan Enter untuk kembali ke menu...█
```

Gambar 28. login , tidak ada usn

Jika valid, usn pass benar, akan masuk ke laman sesuai role:

```
=====
                pasien a
=====

Selamat pagi a! Ada keluhan apa?

1. Daftar Check-Up
2. Antrian Saya
3. Minum Obat
4. Minum Penawar
5. Logout

>>> Masukkan pilihan (1-5): █
```

Gambar 29. login , valid, pasien

Jika input 2, masuk ke register pasien, jika username yang diinput sudah ada maka

```
Username: a
Registrasi gagal! Pasien dengan nama a sudah terdaftar.
Tekan Enter untuk kembali ke menu...█
```

Gambar 30. register, usn sudah dipakai

Jika input usn tidak sesuai format maka

```
Username: 2
Format username hanya boleh huruf! Ulangi
Username: █
```

Gambar 31. register, format usn salah

Jika input usn sesuai

```
Username: halo
Password: halo
Pasien halo berhasil ditambahkan!

Tekan Enter untuk kembali ke menu...█
```

Gambar 32. register, valid, usn benar

Jika input 3, masuk ke lupa password, jika usn salah

```
>>> LUPA PASSWORD  
  
Username: ko  
Username tidak terdaftar!  
  
Tekan Enter untuk kembali ke menu...
```

Gambar 33. LupaPassword, usn tidak terdaftar

Jika usn benar, kode unik salah

```
Username: a  
Kode Unik: l  
Kode unik salah!  
  
Tekan Enter untuk kembali ke menu...
```

Gambar 34. LupaPassword, usn benar, kode unik salah

Jika keduanya benar, maka valid

```
Username: a  
Kode Unik: a  
Halo Dokter a, silakan daftarkan ulang password anda!  
Password baru: loh  
Berhasil!  
  
Tekan Enter untuk kembali ke menu...
```

Gambar 35. LupaPassword, usn benar, kode unik benar, diminta password

Jika input 4, mau exit

```
=====
                RUMAH SAKIT K01-N
=====
1. Login
2. Register
3. LupaPassword
4. Exit

>>> Masukkan pilihan (1-4): 4
```

Gambar 36. Exit, input 4

Jika mau save file, dan input salah

```
Apakah Anda mau melakukan penyimpanan file yang sudah diubah?(y/n)
u

Apakah Anda mau melakukan penyimpanan file yang sudah diubah?(y/n)

```

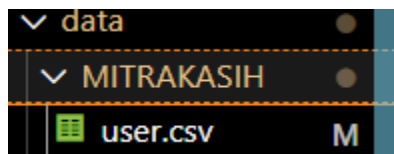
Gambar 37. Exit, input salah

Jika input y maka mau save file

```
Apakah Anda mau melakukan penyimpanan file yang sudah diubah?(y/n)
y
Masukkan nama folder penyimpanan (misal: RSMitraKasih): MITRAKASIH
```

Gambar 38. Exit, input y, mau save

Jika belum ada foldernya, akan terbentuk folder MITRAKASIH di dalam folder data



Gambar 39. Save, Folder buatan save

Jika sudah ada foldernya, akan overwrite folder yang sudah ada

Jika input n, tidak mau save, maka langsung ke pesan keluar

```
Apakah Anda mau melakukan penyimpanan file yang sudah diubah?(y/n)
n
```

Gambar 40. Save, user tidak mau save

```
Terima kasih telah menggunakan sistem! <3
```

Gambar 41. Pesan keluar dari sistem

LAMAN PASIEN

```
=====
                pasien a
=====

Selamat pagi a! Ada keluhan apa?

1. Daftar Check-Up
2. Antrian Saya
3. Minum Obat
4. Minum Penawar
5. Logout

>>> Masukkan pilihan (1-5):
```

Gambar 42. Pasien, awal

Jika input tidak sesuai

```
=====
                pasien a
=====

Selamat pagi a! Ada keluhan apa?

1. Daftar Check-Up
2. Antrian Saya
3. Minum Obat
4. Minum Penawar
5. Logout

>>> Masukkan pilihan (1-5): f

Input tidak valid! Masukkan angka antara 1-5.
Silahkan enter untuk mengulang!
█
```

Gambar 43. Pasien, input tidak sesuai

Jika input 5, mau logout validasi

```
>>> Masukkan pilihan (1-5): 5
Yakin mau logout? (y/n)
y█
```

Gambar 44. Logout, validasi

Jika y, maka mau, akan kembali ke halaman utama

```
=====
                RUMAH SAKIT K01-N
=====

1. Login
2. Register
3. LupaPassword
4. Exit

>>> Masukkan pilihan (1-4): █
```

Gambar 45. Logout, y, kembali ke halaman utama

Jika n, maka tidak mau, kembali ke laman pasien

```
=====
                pasien a
=====

Selamat pagi a! Ada keluhan apa?

1. Daftar Check-Up
2. Antrian Saya
3. Minum Obat
4. Minum Penawar
5. Logout

>>> Masukkan pilihan (1-5): 5
Yakin mau logout? (y/n)
n
```

Gambar 46. Logout, n

```
=====
                pasien a
=====

Selamat pagi a! Ada keluhan apa?

1. Daftar Check-Up
2. Antrian Saya
3. Minum Obat
4. Minum Penawar
5. Logout

>>> Masukkan pilihan (1-5):
```

Gambar 47. Logout, n, kembali ke laman pasien

Ketika pasien memilih opsi ANTRIAN SAYA, dan pasien sudah terdaftar di dalam status antrian

```
>>> ANTRIAN SAYA

STATUS ANTRIAN ANDA:
NAMA: gro
DOKTER: Fajar
RUANGAN: B1
POSISI ANTRIAN: 3 dari 3

Tekan Enter untuk kembali ke menu...
```

Gambar 48. Antrian Saya, pasien sudah terdaftar

Hasil testing ketika pasien sudah terdaftar dalam antrian

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

>>> DAFTAR CHECKUP

Anda sudah terdaftar dalam antrian!

Tekan Enter untuk kembali ke menu...
```

Gambar 49. Daftar Check-up, pasien sudah terdaftar

Hasil testing daftar check-up ketika pasien belum terdaftar dalam antrian dan hendak memilih dokter

```
Berikut adalah daftar dokter yang tersedia:
1. dr. Asep - Ruangan A1 (Antrian: 1 orang)
2. dr. Gunawan - Ruangan A2 (Antrian: 0 orang)
3. dr. Aulia - Ruangan A3 (Antrian: 0 orang)
4. dr. Fajar - Ruangan B1 (Antrian: 0 orang)
5. dr. Agus - Ruangan B3 (Antrian: 0 orang)
Pilih dokter (1-5): 2

Pendaftaran check-up berhasil!
```

Gambar 50. Daftar check-up, list pilihan dokter

Setelah memilih dokter, pasien berhasil mendaftarkan diri untuk check-up


```
Pendaftaran check-up berhasil!  
Anda terdaftar pada antrian dokter: Gunawan, di ruangan: A2  
Posisi antrian Anda: 3  
  
Tekan Enter untuk kembali ke menu...
```

Gambar 51. Daftar Check-up, pendaftaran berhasil

Hasil testing fitur diagnosis

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
  
>>> Pilihan: 1  
Diagnosa untuk pasien stewart:  
[AUTO DIAGNOSE 2.0] Mendiagnosis...  
Hasil diagnosa:  
  
stewart terdiagnosa penyakit: COVID-19  
  
Pasien stewart telah keluar dari antrian.  
  
Tekan Enter untuk kembali ke menu...
```

Gambar 52. Diagnosis, pasien berhasil didiagnosis penyakit

Hasil testing minum obat,

```

>>> MINUM OBAT

Inventory obat kamu:
===== DAFTAR OBAT =====
1. Remdesivir
2. Vitamin C
=====

Pilih obat untuk diminum (1 s.d 2, 0 untuk batal): 1

GLEKGLEKGLEK... Remdesivir berhasil diminum!!!
Inventory obat kamu:
===== DAFTAR OBAT =====
1. Vitamin C
=====

Pilih obat untuk diminum (1 s.d 1, 0 untuk batal): 1

GLEKGLEKGLEK... Vitamin C berhasil diminum!!!

Inventory obat kamu sekarang kosong!

Tekan Enter untuk kembali ke menu...

```

Gambar 53. Minum obat, pasien meminum obat

Hasil testing pulangdok ketika obat yang diminum tidak sesuai urutan

```

>>> PULANG DOK

Dokter sedang memeriksa keadaanmu...

Maaf, tapi kamu masih belum bisa pulang!

Urutan peminuman obat yang diharapkan:
Metformin -> Lisinopril -> Remdesivir -> Vitamin C

Urutan obat yang kamu minum:
Lisinopril -> Metformin -> Remdesivir -> Vitamin C

Silahkan kunjungi dokter untuk meminta penawar yang sesuai !

Tekan Enter untuk kembali ke menu...

```

Gambar 54. PulangDok, pasien salah urutan minum obat

Hal ini berarti pasien harus minum penawar sebanyak 4 kali

```
>>> MINUM PENAWAR

Uwekkk!!! Lisinopril keluar dan kembali ke inventory

Tekan Enter untuk kembali ke menu...[]
```

Gambar 55. Minum Penawar, pasien minum penawar

Kemudian dia akan minum obat lagi sesuai dengan urutan yang seharusnya. Berikut adalah tampilan ketika pasien memilih pilihan yang tidak valid

```
Inventory obat kamu:
===== DAFTAR OBAT =====
1. Vitamin C
2. Remdesivir
3. Lisinopril
=====

Pilih obat untuk diminum (1 s.d 3, 0 untuk batal): 4
Pilihan nomor tidak tersedia!
```

Gambar 56. Minum Obat, pilihan tidak valid

Karena pop sudah meminum obat sesuai urutan, maka ketika dokter memeriksa keadaannya ia sudah diperbolehkan pulang

```
>>> PULANG DOK

Dokter sedang memeriksa keadaanmu...

Selamat! Kamu sudah dinyatakan sembuh oleh dokter. Silahkan pulang dan semoga sehat selaluu!

Tekan Enter untuk kembali ke menu...[]
```

Gambar 57. PulangDok, pasien sudah boleh pulang

Tampilan antrian saya ketika pasien sudah berada di dalam ruangan dokter

```
>>> ANTRIAN SAYA

Anda sedang berada di dalam ruangan dokter!

Tekan Enter untuk kembali ke menu...[]
```

Gambar 58. Antrian saya, pasien sudah berada di dalam ruangan

Tampilan antrian saya ketika pasien masih berada di dalam antrian

```
>>> ANTRIAN SAYA

Status antrian Anda:
Jenis antrian: Diagnosis
Dokter: neronimo
Ruangan: A1
Posisi antrian: 1 dari 2

Tekan Enter untuk kembali ke menu...
```

Gambar 59. Antrian Saya, pasien berada dalam antrian
Hasil testing diagnosis ketika masih ada pasien dalam antrian

```
>>> DIAGNOSIS

Diagnosa untuk pasien gro:
[AUTO DIAGNOSE 2.0] Mendiagnosis...
Hasil diagnosa:

gro terdiagnosa penyakit: COVID-19

Pasien gro telah keluar dari antrian.

Tekan Enter untuk kembali ke menu...
```

Gambar 60. Diagnosis, dokter neronimo mendiagnosis pasien gro

Hasil testing diagnosis ketika antrian sudah kosong

```
>>> DIAGNOSIS

Antrian pasien kosong. Kembali ke menu.

Tekan Enter untuk kembali ke menu...
```

Gambar 61. Diagnosis, antrian kosong

Hasil testing ngobatin ketika pasien sudah pernah diobati oleh dokter

```
>>> Masukkan pilihan (1-3): 2
Dokter sedang mengobati pasien gro!
Pasien gro sudah pernah diobati!

Tekan Enter untuk kembali ke menu...
```

Gambar 62. Ngobatin, pasien sudah pernah diobati

Hasil testing ngobatin ketika pasien belum diobati

```
Dokter sedang mengobati pasien kebin!  
Pasien memiliki penyakit COVID-19  
Obat yang harus diberikan:  
1. Remdesivir  
Obat Remdesivir berhasil ditambahkan ke inventory pasien kebin  
Tekan Enter untuk kembali ke menu...
```

Gambar 63. Ngobatin, pasien sedang diobati dokter

Hasil testing lihat user

```

>>> LIHAT USER

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

Urutan sort?
1. ASC (1-100)
2. DESC (100-1)
>>> Pilihan: 1

Menampilkan semua pengguna dengan ID terurut ascending...
+-----+-----+-----+-----+
| ID      | Nama           | Role    | Penyakit    |
+-----+-----+-----+-----+
| 1       | stewart        | pasien  | COVID-19    |
| 2       | gro            | pasien  | COVID-19    |
| 3       | kebin          | pasien  | COVID-19    |
| 4       | pop            | pasien  | -           |
| 5       | opor           | pasien  | -           |
| 6       | nikeb          | pasien  | -           |
| 7       | tuart          | pasien  | -           |
| 8       | minionette     | pasien  | -           |
| 9       | tobo           | pasien  | -           |
| 10      | neronimo       | dokter  | -           |
| 11      | ciciko         | dokter  | -           |
| 12      | cacako         | dokter  | -           |
| 13      | kroket         | dokter  | -           |
| 15      | risol          | dokter  | -           |
| 16      | tobokan        | pasien  | COVID-19    |
| 20      | popokan        | pasien  | COVID-19    |
| 24      | ropik          | pasien  | -           |
| 88      | zeru           | manager | -           |
| 100     | pendatang      | pasien  | -           |
+-----+-----+-----+-----+

Apakah anda ingin mencari user?
1. Iya
2. Tidak
>>> Pilihan: 

```

Gambar 64. Lihat User, urutkan berdasarkan ID, ascending

```

>>> LIHAT USER

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

Urutan sort?
1. ASC (A-Z)
2. DESC (Z-A)
>>> Pilihan: 2

Menampilkan semua pengguna dengan Nama terurut descending...
+-----+-----+-----+-----+
| ID      | Nama          | Role    | Penyakit |
+-----+-----+-----+-----+
| 88      | zeru          | manager | -        |
| 7       | tuart         | pasien  | -        |
| 16      | tobokan       | pasien  | COVID-19 |
| 9       | tobo         | pasien  | -        |
| 1       | stewart       | pasien  | COVID-19 |
| 24      | ropik         | pasien  | -        |
| 15      | risol         | dokter  | -        |
| 20      | popokan       | pasien  | COVID-19 |
| 4       | pop           | pasien  | -        |
| 100     | pendatang     | pasien  | -        |
| 5       | opor          | pasien  | -        |
| 6       | nikeb         | pasien  | -        |
| 10      | neronimo      | dokter  | -        |
| 8       | minionette    | pasien  | -        |
| 13      | kroket        | dokter  | -        |
| 3       | kebin         | pasien  | COVID-19 |
| 2       | gro           | pasien  | COVID-19 |
| 11      | ciciko        | dokter  | -        |
| 12      | cacako        | dokter  | -        |
+-----+-----+-----+-----+

Apakah anda ingin mencari user?
1. Iya
2. Tidak
>>> Pilihan: █

```

Gambar 65. Lihat User, urutkan berdasarkan nama, descending

```

>>> Pilih apa yang mau dicari
1. Cari User (Pasien dan Dokter)
2. Cari Pasien
3. Cari Dokter
>>> Pilihan: 1
>>> Masukkan Nama User: cacako
Menampilkan pengguna dengan nama cacako:
+-----+-----+-----+-----+
| ID      | Nama          | Role    | Penyakit |
+-----+-----+-----+-----+
| 12      | cacako        | dokter  | -        |
+-----+-----+-----+-----+

Tekan Enter untuk kembali ke menu...

```

Gambar 66. Cari user, melalui interface lihat user

```

>>> Pilih apa yang mau dicari
1. Cari User (Pasien dan Dokter)
2. Cari Pasien
3. Cari Dokter
>>> Pilihan: 2
>>> Masukkan Nama Pasien:ciciko
User dengan Nama ciciko dan role Pasien tidak ditemukan.

Tekan Enter untuk kembali ke menu...

```

Gambar 67. Cari pasien, melalui interface lihat user, ketika input username bukan pasien

```

>>> Pilih apa yang mau dicari
1. Cari User (Pasien dan Dokter)
2. Cari Pasien
3. Cari Dokter
>>> Pilihan: 2
>>> Masukkan Nama Pasien:pop
Menampilkan pengguna dengan nama pop:
+-----+-----+-----+-----+
| ID      | Nama          | Role    | Penyakit    |
+-----+-----+-----+-----+
| 4       | pop           | pasien  | -           |
+-----+-----+-----+-----+

Tekan Enter untuk kembali ke menu...

```

Gambar 68. Cari pasien, melalui interface lihat user, ketika input benar

```

>>> Pilih apa yang mau dicari
1. Cari User (Pasien dan Dokter)
2. Cari Pasien
3. Cari Dokter
>>> Pilihan: 3
>>> Masukkan Nama Dokter:neronimo
Menampilkan pengguna dengan nama neronimo:
+-----+-----+-----+-----+
| ID      | Nama          | Role    | Penyakit    |
+-----+-----+-----+-----+
| 10      | neronimo      | dokter  | -           |
+-----+-----+-----+-----+

Tekan Enter untuk kembali ke menu...

```

Gambar 69. Cari dokter, melalui interface lihat user


```

>>> CARI USER

Cari berdasarkan?
1. ID
2. Nama
3. Riwayat Penyakit
>>> Pilihan: 1
>>> Pilih apa yang mau dicari
1. Cari User (Pasien dan Dokter)
2. Cari Pasien
3. Cari Dokter
>>> Pilihan: 3
>>> Masukkan ID Dokter:13
Menampilkan pengguna dengan ID 13:
+-----+-----+-----+-----+
| ID      | Nama      | Role    | Penyakit |
+-----+-----+-----+-----+
| 13      | kroket    | dokter  | -        |
+-----+-----+-----+-----+

Tekan Enter untuk kembali ke menu...

```

Gambar 70. Cari dokter, melalui interface cari user

```

>>> TAMBAH DOKTER

Username: kroket
Registrasi gagal! Dokter dengan nama kroket sudah terdaftar.

Tekan Enter untuk kembali ke menu...

```

Gambar 71. Tambah dokter, ketika dokter sudah terdaftar

```

>>> TAMBAH DOKTER

Username: tungtung
Password: tungtungtungsahur
Ruangan: A1
Dokter neronimo sudah menempati ruangan A1!
Silakan cari ruangan lain untuk dokter tungtung.
Gagal assign ruangan. Dokter tungtung tidak ditambahkan ke sistem.

Tekan Enter untuk kembali ke menu...

```

Gambar 72. Tambah dokter, ketika ruangan sudah ditempati

```

>>> TAMBAH DOKTER

Username: tuntung
Password: tungsahur
Ruangan: B2
Dokter tuntung berhasil diassign ke ruangan B2!
Dokter tuntung berhasil ditambahkan!

Tekan Enter untuk kembali ke menu...

```

Gambar 73. Tambah dokter, dokter berhasil ditambahkan dan diassign ke ruangan

```
=== Ruangan B2 (Dokter ID: 101) ===

--- Detail Ruangan B2 ---
Kapasitas : 3
Dokter    : Dr. tuntung
Pasien di dalam ruangan:
Tidak ada pasien di dalam ruangan saat ini.
-----
```

Gambar 74. Lihat Antrian, untuk dokter baru dr. tuntung

Hasil testing daftar check-up ketika ditambahkan dokter baru

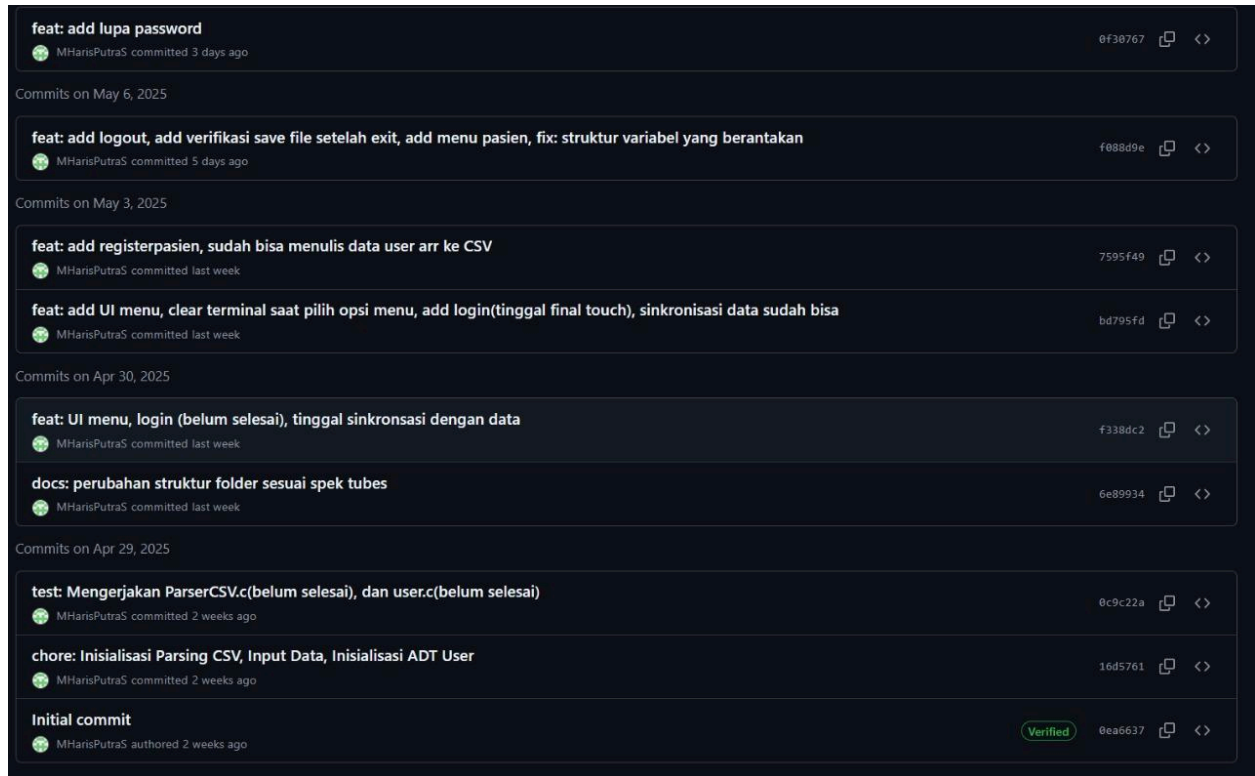
```
>>> DAFTAR CHECKUP

Silakan masukkan data check-up Anda:
Suhu Tubuh (Celcius): 36.8
Tekanan Darah (sistol/diastol, contoh 120 80): 114 74
Detak Jantung (bpm): 62
Saturasi Oksigen (%): 94.6
Kadar Gula Darah (mg/dL): 100
Berat Badan (kg): 80.6
Tinggi Badan (cm): 173
Kadar Kolesterol (mg/dL): 152
Trombosit (ribu/mikroL): 386

Berikut adalah daftar dokter yang tersedia:
1. dr. neronimo - Ruangan A1 (Antrian: 0 orang)
2. dr. ciciko - Ruangan A2 (Antrian: 2 orang)
3. dr. cacako - Ruangan A3 (Antrian: 1 orang)
4. dr. kroket - Ruangan B1 (Antrian: 2 orang)
5. dr. risol - Ruangan B3 (Antrian: 0 orang)
6. dr. tuntung - Ruangan B2 (Antrian: 0 orang)
Pilih dokter (1-6): █
```

Gambar 75. Daftar check-up, dokter baru sudah ditambahkan ke opsi

Lampiran



[history commit di repo sebelum bisa masuk ke github classroom.jpg]
Gambar 76. History commit repo github yang tidak tercantum di github classroom

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

Nomor Asistensi : 01
 No. Kelompok/Kelas : N/01
 Tanggal asistensi : Kamis, 8 Mei 2025

| | | |
|--------------------|----------------------------------|-----------------------------------|
| Anggota kelompok | NIM / Nama (Hanya yang Hadir) | |
| | 1 | 13524089/Aurelia Jennifer Gunawan |
| | 2 | 13524053/Muhammad Haris Putra S. |
| | 3 | 13524007/Muhammad Ashkar |
| | 4 | 13524127/Fazri Arrashyi Putra |
| | 5 | |
| | 6 | |
| Asisten pembimbing | NIM / Nama | |
| | 13522122/Maulvi Ziadinda Maulana | |

Catatan Asistensi:

| | |
|---|--|
| Rangkuman Diskusi | |
| <p>Pada asistensi I ini, kami mendiskusikan format laporan, beberapa spesifikasi fitur, masalah yang ditemui, dan hal-hal yang harus ditindaklanjuti. Berikut ini adalah rangkuman dari hasil diskusi yang telah dilakukan.</p> <ol style="list-style-type: none"> 1. Harus segera masuk repo github labpro22, dan memindahkan hasil pekerjaan dari repo personal ke repo labpro22 2. Penting : <ol style="list-style-type: none"> a. download WSL, b. menggunakan makefile, untuk mengotomatisasi dan memudahkan proses compile program c. membuat fitur load dan save 3. Q: Apa perbedaan spesifikasi untuk tiap modul/prosedur/fungsi yang dibuat (dalam notasi algoritmik) dengan desain command untuk setiap primitif? A: Pada spesifikasi, dilakukan penulisan setiap modul fungsi prosedur yang digunakan dalam implementasi dalam notasi algoritmik, sedangkan pada desain command dilakukan penulisan untuk spek/fitur tugas besar 4. Q : Apakah desain dekomposisi algoritmik dan fungsional program dalam bentuk flowchart atau dalam bentuk lain? A : Flowchart 5. Q : Untuk apa fitur D03-Load digunakan? A : Untuk mengload semua data yang diperlukan ke dalam sistem, seperti file penyakit.csv, dll. | |
| Tindak Lanjut | |
| <p>To-do:</p> <ol style="list-style-type: none"> 1. Join github labpro22 bagi yang belum bergabung 2. Memindahkan hasil pekerjaan yang sebelumnya dilakukan pada repo pribadi ke repo pada github labpro22 3. Download WSL 4. Gunakan makefile untuk memudahkan compile program 5. Buat fitur load & save 6. Lanjutkan mengerjakan fitur-fitur yang belum terimplementasi | |
| Dokumentasi | |

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



A handwritten signature in black ink, appearing to read 'Maulvi'.

**Maulvi Ziadinda Maulana
13522122**

Gambar 77. Hasil pindai MoM form asistensi 1

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

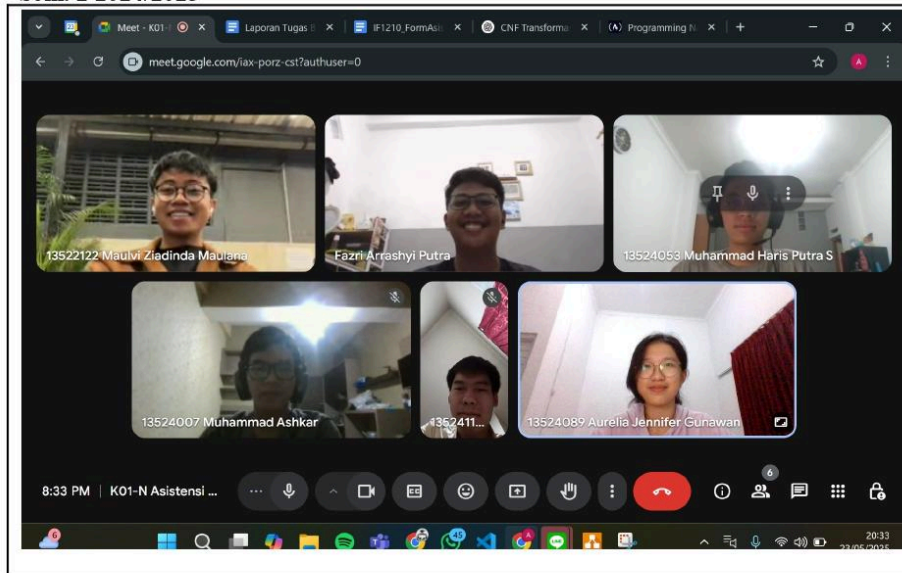
Nomor Asistensi : 02
 No. Kelompok/Kelas : N/01
 Tanggal asistensi : Jumat, 23 Mei 2025

| | | |
|--------------------|-------------------------------|-----------------------------------|
| Anggota kelompok | NIM / Nama (Hanya yang Hadir) | |
| | 1 | 13524007 Muhammad Ashkar |
| | 2 | 13524053 Muhammad Haris Putra S |
| | 3 | 13524089 Aurelia Jennifer Gunawan |
| | 4 | 13524119 Nathanael Shane Bennet |
| | 5 | 13524127 Fazri Arrashyi Putra |
| Asisten pembimbing | 6 | |
| | NIM / Nama | |
| | | 13522122 Maulvi Ziadinda Maulana |

Catatan Asistensi:

| |
|---|
| Rangkuman Diskusi |
| <ul style="list-style-type: none"> • Load dan save disatukan • Map, Queue, Set, dan ADT Lainnya disimpan dalam satu folder khusus ADT • Antrian baru hanya temporary di memori, kecuali user melakukan penyimpanan eksplisit (seperti pada fitur "exit" dan "save") ke config.txt • Cara melakukan testing terhadap antrian adalah dengan login sebagai dokter lalu exit dan login kembali sebagai pasien dan melakukan testing. Namun, testing juga dapat dilakukan dengan menggunakan debugger. • Pada fitur tambahdokter, masih perlu ditambahkan command assign dokter • Penulisan disarankan untuk diseragamkan • Fitur save perlu direvisi |
| Tindak Lanjut |
| <ul style="list-style-type: none"> - Revisi fitur save - Revisi fitur denah - Penambahan assign dokter pada fitur tambah dokter - Melanjutkan pengerjaan fitur yang belum diimplementasi - Memindahkan fitur load dan save ke satu lokasi - Menyatukan seluruh modul ADT ke folder khusus ADT |
| Dokumentasi |

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



A handwritten signature in black ink, appearing to read 'Maulvi Ziadinda M.'.

Maulvi Ziadinda M
13522122

Gambar 78. Hasil Pindai MoM Asistensi 2