

LAPORAN TUGAS BESAR
EL2008 PEMECAHAN MASALAH DENGAN PEMROGRAMAN
MANAJEMEN JADWAL JAGA DOKTER RUMAH SAKIT
SEMOGA SEHAT SELALU



Disusun Oleh:
Kelompok 03

Deven Zie Manuela Sitanggang (13223091)
Muhammad Luthfii Alghazali (13223097)
Sarah Beatrix Sitorus (13223100)
Naufal Ridho Wicaksono (13223102)
Abdul Hakim Yafi (13223107)

Program Studi Teknik Elektro
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2024

DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR GAMBAR.....	3
DAFTAR TABEL.....	6
BAB I	
PENDAHULUAN.....	7
1.1 Latar Belakang.....	7
1.2 Rumusan Masalah.....	7
1.3 Tujuan.....	8
1.4 Manfaat.....	8
1.5 Sistematika Penulisan Laporan.....	9
BAB II	
DEKOMPOSISSI DAN	
SOLUSI MASALAH.....	11
2.1 Deskripsi Masalah Awal.....	11
2.2 Dekomposisi Masalah.....	11
2.3 Pendekatan Solusi Untuk Setiap Modul.....	12
2.4 Fitur Bonus dan Perbaikan.....	18
BAB III	
PERANCANGAN SISTEM.....	20
3.1 Flowchart.....	20
3.1.1. Program utama.....	20
3.1.2. Pengelolaan data dokter.....	26
3.1.3. Penentuan Jadwal Otomatis.....	47
3.1.4. Penampilan Informasi dan Jadwal.....	56
3.2 Data Flow Diagram.....	74
3.2.1 Program Utama.....	74
3.2.2 Pengelolaan Data Dokter.....	77
3.2.3 Penentuan Jadwal Otomatis.....	81
3.2.4 Penampilan Informasi Jadwal dan Laporan.....	83
BAB IV	
IMPLEMENTASI SISTEM.....	88
4.1. Struktur file program.....	88
4.2. Program Utama.....	89
4.2.1. Penjelasan logika.....	89
4.3. Pengelolaan Data Dokter.....	89
4.3.1. Penjelasan logika.....	89
4.3.2. Struktur data.....	91
4.4. Penentuan Jadwal Otomatis.....	92

4.4.1. Penjelasan logika.....	92
4.4.2. Struktur data.....	94
4.5. Penampilan Informasi Jadwal dan Laporan.....	98
4.5.1. Penjelasan logika.....	98
4.5.2. Struktur Data.....	98
BAB V	
STUDI KASUS.....	103
BAB VI	
ANALISIS HASIL DAN	
JUSTIFIKASI.....	105
6.1. Analisis hasil studi kasus.....	113
6.2. Output yang dihasilkan.....	115
6.3. Justifikasi pendekatan yang digunakan.....	117
6.4. Kesimpulan dan saran.....	118
6.4.1. Kesimpulan.....	118
6.4.2. Saran untuk pengembangan lebih lanjut.....	118
BAB VII	
PEMBAGIAN TUGAS.....	119
DAFTAR PUSTAKA.....	121
LAMPIRAN.....	122

DAFTAR GAMBAR

Gambar 1. Flowchart main.c.....	20
Gambar 2. Flowchart fungsi menu_manajemen_dokter.....	21
Gambar 3. Flowchart fungsi menu_penjadwalan_dokter.....	22
Gambar 4. Flowchart fungsi menu_lihat_jadwal.....	23
Gambar 5. Flowchart fungsi menu_laporan_kinerja.....	24
Gambar 6. Flowchart dokter.c.....	26
Gambar 7. Flowchart desain struktur data Dokter.c.....	27
Gambar 8. Flowchart fungsi load_data_daricsv().....	28
Gambar 9. Flowchart fungsi tambah_dokter_manual().....	30
Gambar 10. Flowchart fungsi hapus_dokter().....	32
Gambar 11. Flowchart fungsi cari_dokter_nama(), cari_dokter_bidang(), cari_dokter_tingkat().....	34
Gambar 12. Flowchart fungsi cari_dokter_menu().....	35
Gambar 13. Flowchart fungsi tampilkan_hasil(Dokter* hasil).....	37
Gambar 14. Flowchart tampilkan_semua().....	38
Gambar 15. Flowchart fungsi tambah_aktivitas(AksiType tipe, Dokter* data).....	39
Gambar 16. Flowchart fungsi tampilkan_log().....	40
Gambar 17. Flowchart fungsi save_data_to_csv(const char *nama_file).....	42
Gambar 18. Flowchart fungsi clean_newline(char* str).....	43
Gambar 19. Flowchart fungsi contains_keyword(const char* src, const char* keyword).....	44
Gambar 20. Flowchart fungsi salin_dokter(Dokter* src).....	45
Gambar 21. Flowchart fungsi free_memory().....	46
Gambar 22. Flowchart penjadwalan.c.....	47
Gambar 23. Flowchart fungsi load_doctors_from_csv().....	48
Gambar 24. Flowchart fungsi generate_schedule().....	49
Gambar 25. Flowchart fungsi calculate_score().....	50
Gambar 26. Flowchart fungsi save_schedule().....	51
Gambar 27. Flowchart fungsi parse_shift().....	52
Gambar 28. Flowchart fungsi is_doctor_already_assigned().....	53
Gambar 29. Flowchart fungsi is_doctor_available().....	54
Gambar 30. Flowchart fungsi assign_shift().....	55
Gambar 31. Flowchart Laporan1.c.....	56
Gambar 32. Flowchart jalankan_Opsi1().....	57
Gambar 33. Flowchart jalankan_Opsi2().....	58
Gambar 34. Flowchart jalankan_Opsi3.....	59
Gambar 35. Laporan..c.....	60
Gambar 36. Flowchart fungsi load_shift_data.....	61

Gambar 37. Flowchart fungsi load_doctor_data().....	63
Gambar 38. Flowchart fungsi proses_data().....	65
Gambar 39. Flowchart fungsi jumlah_shift().....	67
Gambar 40. Flowchart fungsi lihat_pelanggaran().....	69
Gambar 41. Flowchart fungsi lihat_pelanggaran_dokter().....	71
Gambar 42. Flowchart fungsi lihat_jumlah_pelanggaran_per_dokter().....	73
Gambar 43. DFD level 0 program utama main.c.....	74
Gambar 44. DFD level 1 program utama main.c.....	75
Gambar 45. DFD Level 0 Pengelolaan Data Dokter.....	77
Gambar 46. DFD Level 1 Pengelolaan Data Dokter.....	78
Gambar 47. DFD Level 0 Penentuan Jadwal Otomatis.....	81
Gambar 48. DFD Level 1 Penentuan Jadwal Otomatis.....	82
Gambar 49. DFD level 0 Informasi Jadwal.....	83
Gambar 50. DFD Level 1 Informasi Jadwal.....	84
Gambar 51. DFD Level 0 Pelaporan.....	85
Gambar 52. DFD Level 1 Pelaporan.....	86
Gambar 53. Kondisi akhir daftar_dokter.csv.....	115
Gambar 54. Hasil file jadwal_dokter.csv.....	116
Gambar 55. Hasil file laporan_dokter.csv.....	116

DAFTAR TABEL

Tabel 1. Struktur Data Doctor.....	95
Tabel 2. Struktur Data Jadwal.....	96
Tabel 3. Struktur Data Enum.....	96
Tabel 4. Struktur Data Konstanta.....	97
Tabel 5. Struktur Data Array Pendukung.....	98
Tabel 6. Hasil studi kasus operasi program.....	105

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penjadwalan waktu kerja tenaga medis, terlebihnya dokter, merupakan salah satu aspek penting dalam operasional rumah sakit. Rumah Sakit Semoga Sehat Selalu menghadapi tantangan dalam penyusunan jadwal jaga dokter untuk periode 30 hari, yang terdiri dari tiga shift jaga per hari (pagi, siang, malam). Penjadwalan ini harus mempertimbangkan sejumlah parameter seperti ketersediaan dokter, batas maksimum shift per minggu, serta preferensi shift dan waktu masing-masing dokter. Ketidaksesuaian dalam penjadwalan dapat menyebabkan kelebihan beban kerja, pelanggaran preferensi pribadi, hingga terjadinya penurunan kualitas layanan.

Sebelumnya proses penjadwalan ini dilakukan secara manual, bukan hanya memakan waktu tetapi juga rentan terhadap pelanggaran preferensi dokter saat pembagian shift jaga. Oleh karena itu, diperlukan sebuah sistem otomatis yang dapat menyusun jadwal dengan mempertimbangkan berbagai parameter tersebut. Sistem ini diharapkan dapat meningkatkan efisiensi operasional, distribusi shift, serta kepuasan kerja dokter.

Melalui Tugas Besar EL2008 Pemecahan Masalah dengan Pemrograman ini, akan dikembangkan sebuah aplikasi penjadwalan dokter otomatis berbasis bahasa pemrograman C. Sistem ini akan memanfaatkan pendekatan algoritmik, yaitu algoritma greedy, untuk penjadwalan. Program ini akan memanfaatkan GTK sebagai sarana antarmuka yang ramah pengguna dan terminal *powershell*.

1.2 Rumusan Masalah

Berdasarkan latar belakang, terdapat beberapa masalah utama yang perlu dipecahkan dalam tugas besar ini, sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan sistem pengelolaan data dokter yang mencakup fungsi tambah, hapus, tampil, dan baca dari file eksternal (CSV)?
2. Bagaimana merancang algoritma penjadwalan otomatis yang dapat mengalokasikan dokter secara optimal ke dalam 90 shift dalam 30 hari, memperhatikan kebutuhan dokter, batas beban kerja, dan preferensi individu?
3. Bagaimana menampilkan hasil penjadwalan dalam format harian, mingguan, dan bulanan serta menyediakan laporan jumlah shift per dokter dan pelanggaran preferensi dokter?
4. Bagaimana merancang program dengan sistem modular berbasis Bahasa C yang dapat dengan mudah diintegrasikan ke dalam GUI berbasis GTK?

1.3 Tujuan

Berdasarkan rumusan masalah di atas, tujuan dari dokumen ini adalah untuk memberikan solusi yang terstruktur dan efektif dalam perancangan serta implementasi sistem penjadwalan shift dokter secara otomatis. Adapun tujuan-tujuan spesifik yang ingin dicapai adalah sebagai berikut:

1. Merancang sistem pengelolaan data dokter yang mampu melakukan proses tambah, hapus, tampil, dan baca data dari file eksternal (CSV) secara efisien.
2. Mengembangkan algoritma penjadwalan otomatis yang mempertimbangkan jumlah shift, batas kerja maksimal per minggu, dan preferensi dokter untuk menghasilkan jadwal yang optimal.
3. Menyediakan fitur visualisasi dan pelaporan jadwal jaga dalam berbagai format (harian, mingguan, bulanan), termasuk informasi pelanggaran preferensi dan distribusi beban kerja.
4. Membangun sistem yang modular dan mudah diintegrasikan dengan antarmuka melalui terminal *powershell* dan GUI berbasis GTK guna meningkatkan kemudahan penggunaan dan pengalaman pengguna.

1.4 Manfaat

Adapun manfaat yang diharapkan dari perancangan dan implementasi program ini adalah sebagai berikut:

1. Bagi Rumah Sakit Semoga Sehat Selalu
 - a) Meningkatkan efisiensi dan akurasi dalam proses penjadwalan dokter, mengurangi kesalahan akibat penjadwalan manual.
 - b) Memastikan ketersediaan dokter yang memadai di setiap shift, sehingga pelayanan medis tetap optimal.
 - c) Mengurangi potensi konflik internal dan ketidakpuasan dokter terkait distribusi shift.
 - d) Memudahkan akses informasi jadwal bagi manajemen dan dokter.
2. Bagi Dokter
 - a) Memastikan beban kerja yang adil dan sesuai dengan batasan maksimal shift per minggu.
 - b) Preferensi shift lebih diperhatikan, meningkatkan performa dan kepuasan kerja.
 - c) Informasi jadwal yang jelas dan mudah diakses.
3. Bagi Pengembang (Mahasiswa)
 - a) Menerapkan konsep pemecahan masalah dengan pemrograman dalam kasus nyata.
 - b) Meningkatkan pemahaman dan keterampilan dalam perancangan algoritma kompleks.
 - c) Mendapatkan pengalaman dalam pengembangan aplikasi modular menggunakan Bahasa C.
 - d) Mempelajari integrasi dengan menggunakan antarmuka grafis (GUI) menggunakan GTK.

1.5 Sistematika Penulisan Laporan

Laporan ini disusun dengan sistematika sebagai berikut:

1. BAB I PENDAHULUAN: Menjelaskan latar belakang, rumusan masalah, tujuan, manfaat proyek, dan sistematika laporan.
2. BAB II DEKOMPOSISI DAN SOLUSI MASALAH: Memecah masalah ke dalam sub-masalah (manajemen dokter, penjadwalan) dan menjelaskan solusi umum yang digunakan.
3. BAB III PERANCANGAN SISTEM (Flowchart dan DFD): Menyajikan diagram alur (flowchart) dan Data Flow Diagram (DFD) secara tekstual untuk menggambarkan alur kerja sistem.
4. BAB IV IMPLEMENTASI PROGRAM (Penjelasan Logika dan Struktur Data): Menjelaskan algoritma penjadwalan (greedy), struktur data (linked list, array, enum), logika pemrosesan data, dan implementasi GUI dengan GTK.
5. BAB V STUDI KASUS: Menyajikan contoh input (data dokter) dan output (jadwal, laporan) berdasarkan file CSV yang ada.
6. BAB VI ANALISIS HASIL DAN JUSTIFIKASI: Menganalisis hasil penjadwalan, pelanggaran preferensi, dan justifikasi terhadap pendekatan greedy serta struktur modular.
7. BAB VII PEMBAGIAN TUGAS: Menjelaskan kontribusi substansial masing-masing anggota kelompok.

BAB II

DEKOMPOSISSI DAN

SOLUSI MASALAH

2.1 Deskripsi Masalah Awal

Rumah Sakit Semoga Sehat Selalu membutuhkan sistem otomatis untuk mengelola jadwal jaga dokter selama 30 hari, dengan tiga shift harian (pagi, siang, malam). Sistem dapat memastikan setiap shift diisi minimal dengan satu dokter (diimplementasikan dengan dua dokter per shift), mematuhi batasan maksimal shift per minggu, dan meminimalkan pelanggaran preferensi. Selain itu, sistem harus mengelola data dokter, menampilkan jadwal (harian, mingguan, bulanan), menghitung jumlah shift, mendeteksi pelanggaran, dan menyimpan hasil ke file CSV. Untuk meningkatkan pengalaman pengguna, sistem dirancang secara modular.

2.2 Dekomposisi Masalah

Masalah utama didekomposisi menjadi tiga modul fungsional:

1. Pengelolaan Data Dokter: Mengelola data dokter (nama, bidang, tingkat, maksimal shift per minggu, preferensi shift, preferensi waktu), termasuk operasi tambah, hapus, cari, dan tampilkan, dengan data disimpan di `daftar_dokter.csv`.
2. Penentuan Jadwal Otomatis: Menyusun jadwal jaga untuk 30 hari (90 shift), memastikan dua dokter per shift, mematuhi batasan shift mingguan, meminimalkan pelanggaran preferensi, dan menyimpan jadwal ke `jadwal_dokter.csv`.
3. Penampilan Informasi Jadwal dan Laporan: Menampilkan jadwal (harian, mingguan, bulanan), menghitung jumlah shift per dokter, mendeteksi pelanggaran preferensi (shift, waktu, batasan shift), dan menyimpan laporan ke `laporan_dokter.csv`.

2.3 Pendekatan Solusi Untuk Setiap Modul

A. Pengelolaan Data Dokter

Modul ini digunakan untuk mengelola data dokter, termasuk penambahan, penghapusan, dan pencarian dokter, serta menyimpan dan memuat data dari file CSV. Pada modul ini juga terdapat juga fitur undo yang dapat membatalkan tindakan terakhir yang dilakukan oleh user saat sesi itu berlangsung.

Untuk mengatur struktur data yang ada pada modul ini digunakan linked list dengan struktur dokter yang berisi Nama, bidang, tingkat dokter, preferensi shift dan waktu, batas shift maksimal per minggu, dan pointer ke node berikutnya. Fungsi `free_memory()` pada modul ini digunakan untuk membersihkan seluruh alokasi sebelum program berakhir. Fungsi `salin_dokter()` digunakan untuk membuat salinan data dokter ketika diperlukan pada fungsi lain. Linked list digunakan pada modul ini untuk memudahkan dalam penambahan dan penghapusan data, efisien dalam penggunaan memori dan juga dalam operasi undo.

1. Operasi tambah dokter

Fungsi `tambah_dokter_manual()` digunakan untuk meminta input dari pengguna lalu menambahkan node baru ke head linked list dan menyimpannya ke file CSV setelah penambahan dilakukan

2. Operasi hapus dokter

Fungsi `hapus_dokter()` meminta input nama dokter yang ingin di hapus dari pengguna lalu mengonfirmasi kepada pengguna jika yakin untuk dihapus. Setelah itu, melakukan pencarian berdasarkan nama dari pengguna menggunakan traversal linked list lalu melakuakan penghapusan node dan pembebasan memori dan menyimpannya lagi ke file CSV setelah penghapusan.

3. Operasi tampilan data

Fungsi `tampilkan_semua()` digunakan untuk menampilkan semua data dokter dan juga terdapat fungsi pencarian dengan filter (nama, bidang tingkat).

4. Fitur Undo

Pada modul ini juga tersedia fitur UNDO. Fitur ini menggunakan stack (linked list) untuk menyimpan aktivitas. Terdapat dua jenis aktivitas yaitu `AKSI_TAMBAH` dan `AKSI_HAPUS`. Pada fitur ini juga terdapat fungsi `tambah_aktivitas` untuk mencatat setiap operasi yang pernah dilakukan dan menyalin salinan datanya. Lalu fungsi `hapus_aktivitas_terakhir()` digunakan untuk menghapus aktivitas terakhir yang dilakukan oleh pengguna. Pada fitur ini juga disediakan log aktivitas untuk melihat aktivitas yang dilakukan oleh pengguna melalui `tampilan_log()`.

5. Fungsi pencarian

Terdapat fungsi `cari_dokter_nama()`, `cari_dokter_bidang()`, dan `cari_dokter_tingkat()`. Ketiga fungsi ini menggunakan case-insensitive dengan memanfaatkan fungsi `strcontains_keyword()`, yang memungkinkan pencarian tanpa memperhatikan perbedaan huruf besar/kecil. Hasil pencarian kemudian disalin ke dalam linked list yang terpisah untuk memudahkan proses penampilan hasil. Proses pencarian ini dilakukan menggunakan algoritma linear search dengan menelusuri seluruh node dalam linked list utama. Untuk setiap nodenya, dilakukan pembandingan string dengan mengkonversi kedua string ke lowercase, lalu menggunakan fungsi `strstr()` untuk memeriksa apakah keyword pencarian yang digunakan terdapat sebagai substring dalam data dokter. Lalu terdapat juga fungsi `tampilkan_hasil()` untuk menampilkan seluruh data dokter secara terstruktur. Lalu membersihkan memori hasil pencarian setelah data dokter ditampilkan untuk mencegah terjadinya memory leak.

6. Fungsi load data

Load data pada modul ini digunakan melalui fungsi `load_data_dari_csv()`. Fungsi ini membaca data dari file CSV. Jika file atau direktori tidak ada maka fungsi ini akan membuat file/direktori yang baru. Proses parsing data digunakan dengan fungsi `strtok()` untuk memisahkan field - field dalam format csv, dan dilakukan konversi tipe data yang diperlukan menggunakan `atoi()` untuk mengubah string menjadi integer.

7. Fungsi save data

Untuk menyimpan data digunakan melalui fungsi `save_data_to_csv()`. Fungsi ini digunakan untuk menulis seluruh linked list ke dalam file dan mencakup juga header kolom. Fungsi ini juga dapat melakukan penanganan error dengan membuat direktori otomatis jika diperlukan.

B. Penentuan Jadwal Otomatis

Modul ini menyusun jadwal jaga untuk 30 hari (90 shift) dengan dua dokter per shift, mematuhi batasan maksimal shift per minggu, dan meminimalkan pelanggaran preferensi shift (pagi, siang, malam) dan waktu (awal bulan: hari 1-15, akhir bulan: hari 16-30). Solusi menggunakan algoritma greedy sebagai pendekatan utama, yang diterapkan dalam fungsi `generate_schedule()`.

1. Apa itu Greedy Algorithm?

Algoritma greedy adalah metode pemecahan masalah yang membuat keputusan optimal secara lokal pada setiap langkah, dengan harapan menghasilkan solusi keseluruhan yang baik [1]. Dalam konteks ini, greedy memilih dokter dengan skor tertinggi untuk setiap shift berdasarkan preferensi dan ketersediaan, tanpa kembali mengevaluasi keputusan sebelumnya.

2. Penggunaan Greedy

Sistem mengiterasi setiap shift (90 total) dan menghitung skor untuk setiap dokter menggunakan fungsi sederhana yang mempertimbangkan:

- Kesesuaian preferensi shift (misalnya, +15 untuk "Pagi" di shift pagi, -5 jika tidak sesuai).
- Kesesuaian preferensi waktu (misalnya, +10 untuk "AwalBulan" di hari 1-15).
- Prioritas untuk dokter dengan shift mingguan atau total lebih sedikit.
- Bobot tambahan berdasarkan tingkat dokter.

Dua dokter akan dipilih dengan skor tertinggi per shift, dengan pengecekan ketersediaan untuk memastikan tidak melampaui batas shift mingguan. Jika tidak ada dokter yang memenuhi, sistem mengizinkan pelanggaran agar jadwal tetap terisi.

3. Mengapa Memilih Greedy?

- Efisien: Memberikan solusi cepat untuk jumlah dokter dan shift yang terbatas (maksimal 1000 dokter, 90 shift), cocok untuk kebutuhan rumah sakit.
- Sederhana: Tidak memerlukan perhitungan kompleks seperti algoritma eksak, sehingga mudah diimplementasikan.
- Efektif: Meminimalkan pelanggaran preferensi dengan pendekatan skor
- Meskipun tidak menjamin solusi optimal global, greedy cukup untuk kebutuhan praktis tugas ini. Jadwal disimpan ke jadwal_dokter.csv dan dapat diakses via menu_penjadwalan_otomatis() dari main.c.

C. Penampilan Informasi Jadwal dan Laporan

Modul ini memberikan laporan informasi mengenai jadwal jaga tiap dokter (harian, mingguan, maupun bulanan), informasi jumlah shift tiap dokter, informasi jumlah pelanggaran shift tiap dokter (jika ada), lalu melakukan penulisan ke file csv.

Maka dari itu, tiap masalah diatas dibuat solusi-solusi sebagai berikut :

1. Laporan jumlah shift

Untuk melihat jumlah shift tiap dokter, user pertama perlu menginput terlebih dahulu dokter mana yang ingin dicek jumlah shiftnya, lalu program akan mengecek di file laporan_dokter.csv, program akan melakukan iterasi (while loop), lalu mencari nama dokter yang sesuai dengan input user, jika ketemu lalu akan ditampilkan jumlah shiftnya, jika tidak, program akan mengeluarkan pesan “Dokter dengan nama [nama dokter] tidak ditemukan.”

2. Laporan pelanggaran

Di bagian ini, pertama-tama program akan mengecek input dari file jadwal_dokter.csv, program melakukan iterasi sebanyak jumlah shift yang ada di file input, lalu di setiap shift akan dilakukan pengecekan sebagai berikut :

- Mengecek preferensi shift

Apakah di shift tersebut, dokter yang melakukan jaga sudah sesuai dengan preferensi shiftnya, misalnya dokter A mempunyai preferensi shift pagi, akan dicek apakah dokter A sudah tepat berjaga di pagi atau tidak, jika tidak akan dianggap pelanggaran dan akan ditampilkan.

- Mengecek preferensi waktu (awal atau akhir bulan)

Akan dilakukan pengecekan terhadap preferensi dokter terkait waktu jaga di awal bulan atau akhir bulan, dokter dengan preferensi waktu awal bulan(tanggal 1-15) atau akhir bulan (16-31, 16-28 atau 29 jika bulan februari) akan dicek apakah sudah sesuai atau belum, jika tidak akan dianggap sebuah pelanggaran dan akan ditampilkan pelanggarannya.

- Mengecek kelebihan shift mingguan

Setiap dokter mempunyai batas maksimal jumlah shift setiap minggu, setiap iterasi akan dilakukan perhitungan jumlah shift dokter tersebut dalam minggu tersebut. Misalnya iterasi sudah berada di hari ke 5 di pagi

hari, dokter yang berjaga akan dicek apakah di minggu tersebut jumlah shiftnya mencapai batas atau belum, jika sudah akan dianggap pelanggaran dan ditampilkan.

3. Menulis laporan ke file

Program akan menulis laporan ke file laporan_dokter.csv, terdapat dua file lainnya yang digunakan sebagai input agar datanya diproses terlebih dahulu sebelum ditulis, dua file tersebut adalah jadwal_dokter.csv dan daftar_dokter.csv

File akan menulis informasi sebagai berikut

- Nama dokter,
Diambil dari file daftar_dokter.csv
- Bidang dokter,
Diambil dari file daftar_dokter.csv
- Tingkat dokter,
Diambil dari file daftar_dokter.csv
- Total shift,
Diambil dari file jadwal_dokter.csv, akan dilakukan terlebih dahulu berapa total shift dari tiap dokter.
- Maksimal shift perminggu,
Diambil dari file daftar_dokter.csv
- Preferensi shift,
Diambil dari file daftar_dokter.csv
- Preferensi waktu,
Diambil dari file daftar_dokter.csv
- pelanggaran preferensi tiap dokter
Diambil dari file daftar_dokter dan jadwal_dokter.csv, dilakukan terlebih dahulu pelanggaran preferensi tiap dokter.
- Jumlah shift di setiap minggu,
Di ambil dari file jadwal_dokter.csv, menampilkan jumlah shift di tiap minggu, misal minggu ke 1 berapa total shift dari dokter tersebut.

- Dan jumlah shift di setiap jadwal jaga (pagi, siang dan malam).
Di ambil dari file jadwal_dokter.csv, menampilkan jumlah shift di tiap jadwal jaga, misal di pagi berapa total shift dari dokter tersebut.

2.4 Fitur Bonus dan Perbaikan

a. Fitur Bonus

- Fitur Undo untuk Manajemen Dokter: Modul dokter.c mendukung pembatalan operasi tambah/hapus dokter melalui struktur Aktivitas dan fungsi hapus_aktivitas_terakhir(). Ini memungkinkan pengguna untuk mengembalikan perubahan terakhir, meningkatkan fleksibilitas dan keamanan manajemen data.
- Statistik Dokter : Fungsi statistik() di dokter.c menyediakan analisis tambahan, seperti rata-rata shift maksimal, distribusi preferensi shift (pagi, siang, malam), preferensi waktu (awal bulan, akhir bulan, campur), dan frekuensi bidang/tingkat dokter, yang membantu manajemen rumah sakit memahami komposisi tim dokter.
- Pencarian Case-Insensitive: Fungsi pencarian (cari_dokter_nama(), cari_dokter_bidang(), cari_dokter_tingkat()) mendukung pencocokan parsial tanpa mempedulikan huruf besar/kecil, membuat pencarian lebih fleksibel.
- Kalender Interaktif: Fungsi jalankanOpsi1() dan jalankanOpsi2() di laporan1.c menampilkan kalender sederhana sebelum memilih tanggal/minggu, membantu pengguna memvisualisasikan periode jadwal.
- Rencana Integrasi GUI dengan GTK: Sistem dirancang modular untuk mendukung pengembangan GUI menggunakan GTK.

b. Perbaikan

- Penanganan Error File: Modul dokter.c dan penjadwalan.c menyertakan penanganan error seperti pembuatan direktori data/ jika tidak ada
- Pembersihan Buffer Input: Fungsi clear_input_buffer() di main.c mencegah error input dari scanf, memastikan antarmuka pengguna lebih stabil.

- Validasi Input: Fungsi seperti tambah_dokter_manual() memeriksa input kosong atau tidak valid
- Modularitas Kode: Pemisahan fungsi ke dalam modul terpisah (dokter.c, penjadwalan.c, laporan.c, laporan1.c) dengan header masing-masing meningkatkan keterbacaan, kemudahan debugging.

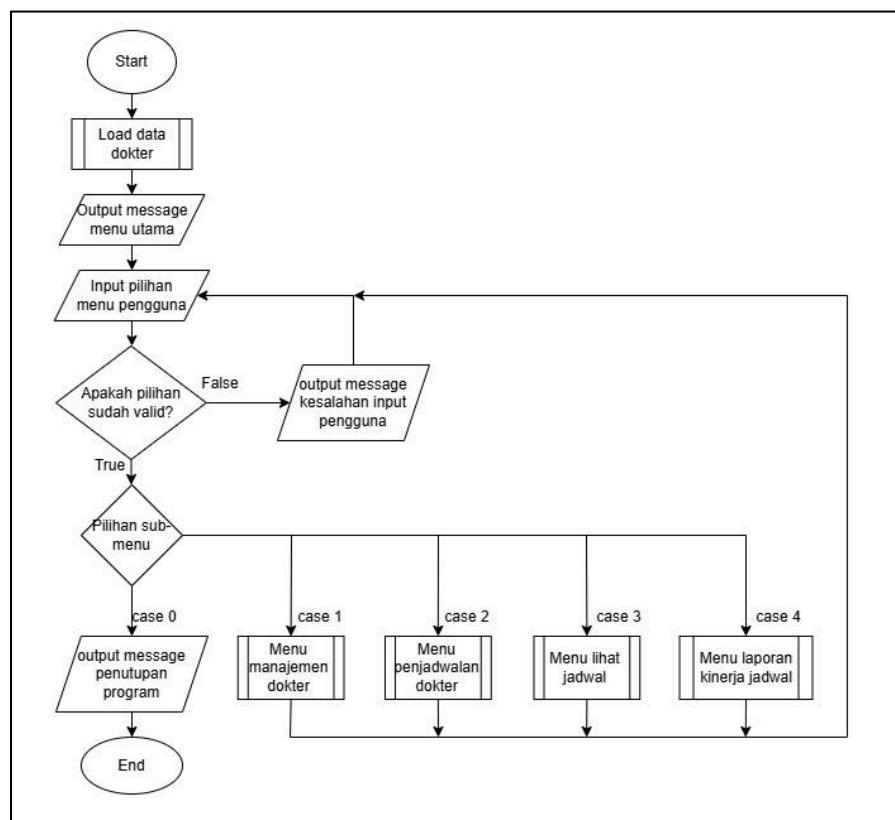
BAB III

PERANCANGAN SISTEM

3.1 Flowchart

3.1.1. Program utama

a. Flowchart main.c

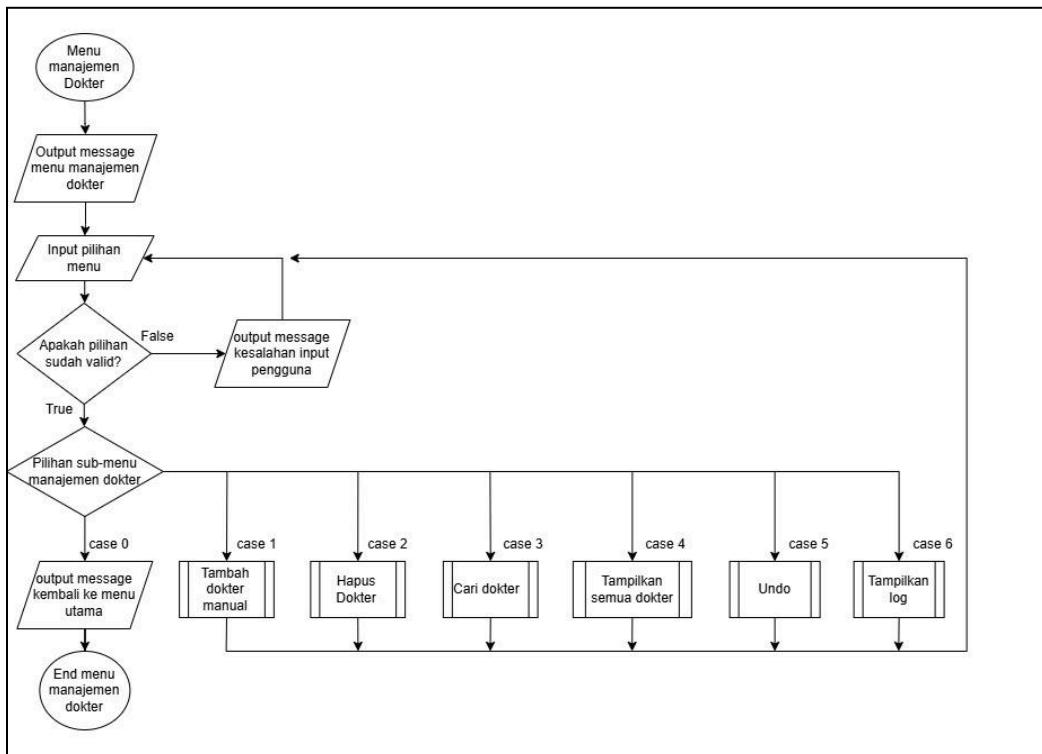


Gambar 1. Flowchart main.c

Flowchart ini menjelaskan alur kerja dari program utama main.c. Program ini bekerja sebagai antarmuka pengguna dengan program. Pertama program akan load daftar_dokter.csv dan memberikan message yang menampilkan pilihan menu utama. Lalu, pengguna akan input menu yang ingin dipilih. Jika pilihannya valid (ada dalam daftar fitur, direpresentasikan dengan pilihan 1-4) maka program akan masuk switch case. Masing-masing case membawa program ke menu fungsi

masing-masing. Jika pengguna memilih untuk menutup program, maka program akan memberi kalimat penutup dan program berhenti.

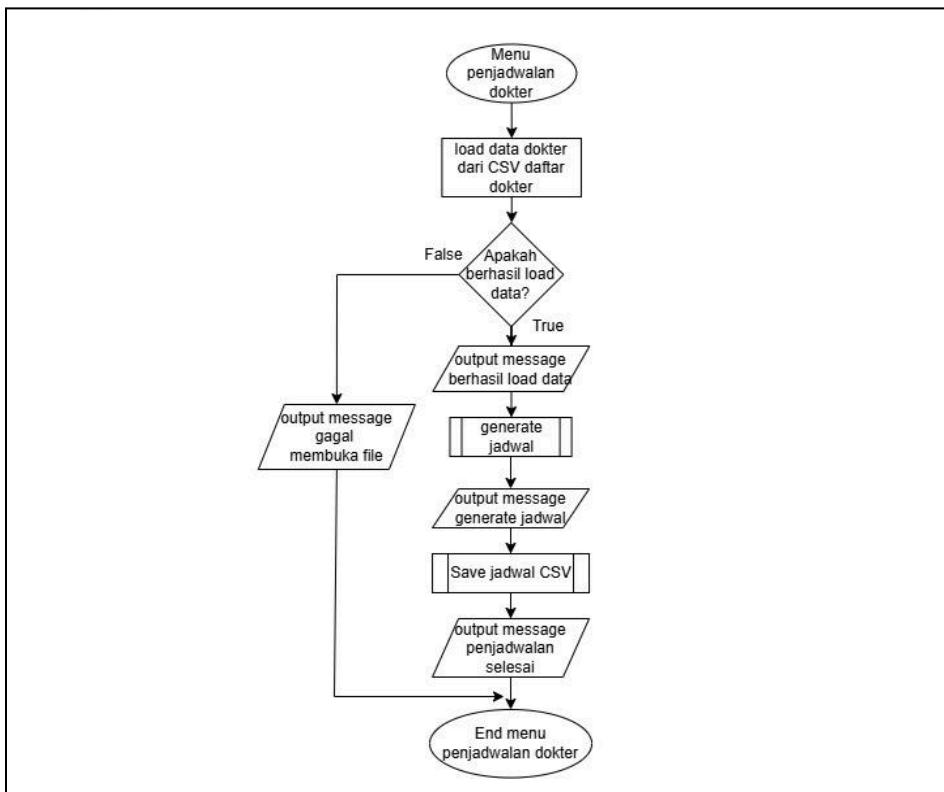
b. Fungsi menu_manajemen_dokter



Gambar 2. Flowchart fungsi menu_manajemen_dokter

Flowchart ini menjelaskan alur kerja menu manajemen dokter. Menu ini berfungsi sebagai antarmuka pengguna dengan pilihan-pilihan fungsi yang disediakan. Ketika pengguna memilih menu manajemen dokter, program akan memberikan message pilihan fungsi yang bisa digunakan. Kemudian pengguna akan memberikan masukan fungsi yang ingin dipilih. Lalu, jika pilihan pengguna valid (ada dalam daftar pilihan, direpresentasikan dalam case 1-6), program akan membawa pengguna ke fungsi masing-masing dan masuk ke sub program pengelolaan data dokter. Jika pengguna memilih untuk keluar ke menu utama, maka alur menu ini selesai dan kembali ke menu utama.

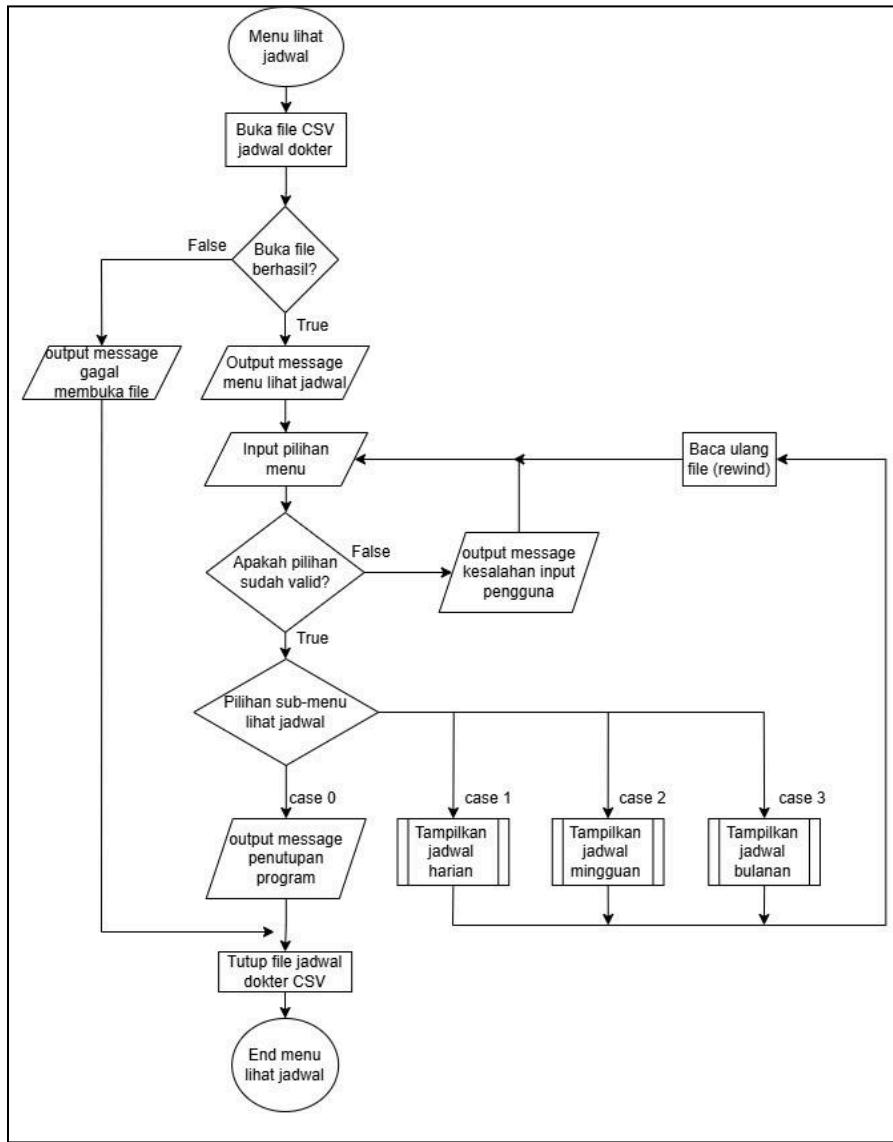
c. Fungsi menu_penjadwalan_otomatis



Gambar 3. Flowchart fungsi menu_penjadwalan_dokter

Flowchart ini menjelaskan alur kerja menu penjadwalan dokter. Fungsi ini berfungsi untuk generate jadwal dokter yang baru dengan mempertimbangkan preferensi dokter-dokter yang ada pada data dokter. Pertama, program akan load data dokter dari daftar_dokter.csv. Lalu, jika load file berhasil, maka program akan memberikan message load data berhasil dan menjalankan fungsi untuk generasi jadwal dokter. Jika berhasil maka akan memberikan message. Lalu dilanjutkan dengan save jadwal tersebut menjadi file jadwal_dokter.csv. Setelah itu, message penjadwalan berhasil diberikan dan alur menu ini selesai.

d. Fungsi menu_lihat_jadwal

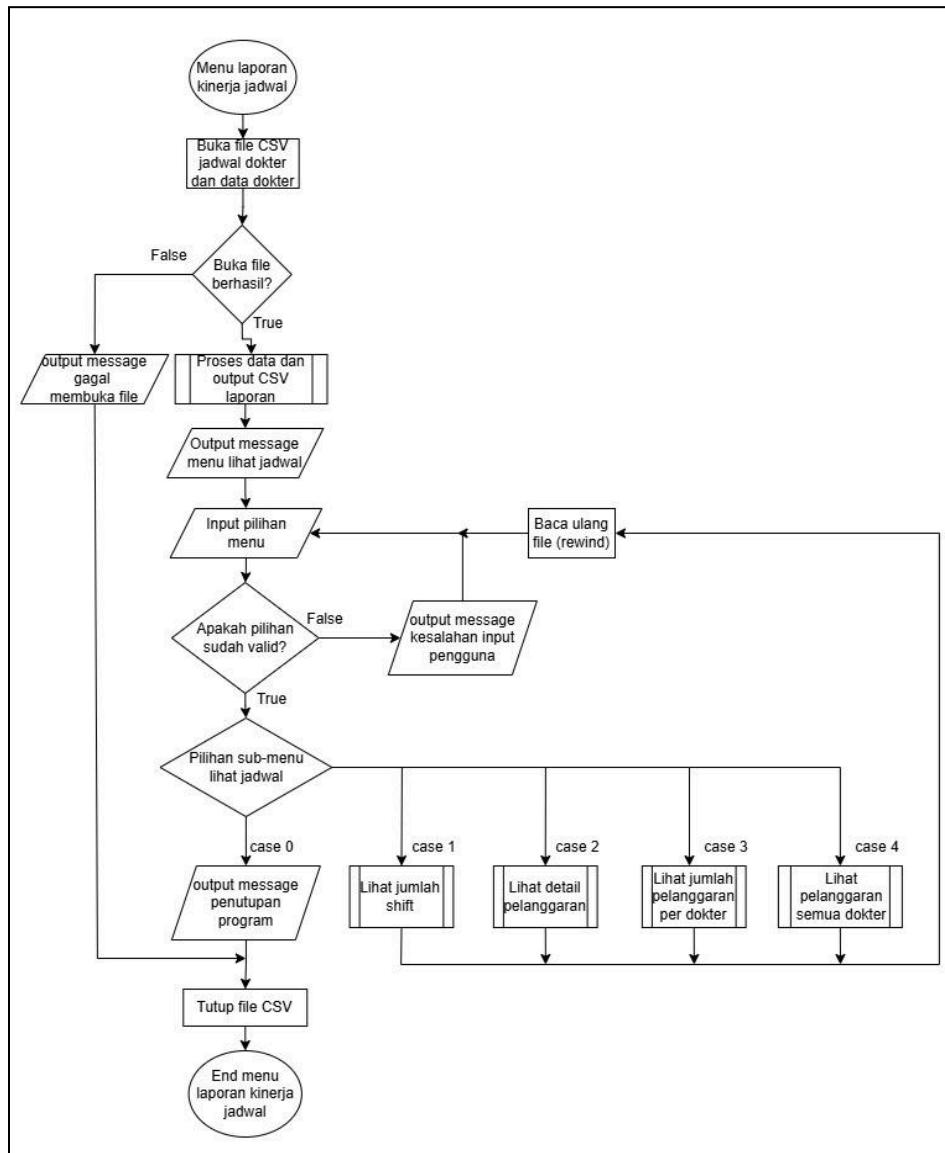


Gambar 4. Flowchart fungsi menu_lihat_jadwal

Flowchart ini menjelaskan alur menu lihat jadwal. Menu ini berfungsi sebagai antarmuka pengguna dengan fungsi yang disediakan untuk melihat jadwal yang sudah ada atau sudah dibuat. Jadwal dapat dilihat secara harian, mingguan, dan bulanan. Pertama, program akan membuka file jadwal_dokter.csv yang dibuat. Jika berhasil, program akan memberikan message pilihan fungsi yang disediakan. Pengguna lalu memasukkan perintah yang ingin dilakukan. Jika pilihan fungsinya

valid, maka program akan membawa pengguna ke fungsi masing-masing dan masuk ke subprogram penampilan informasi jadwal. Jika pengguna memilih untuk keluar ke menu utama, maka alur menu ini selesai dan kembali ke menu utama.

e. Fungsi menu_laporan_kinerja

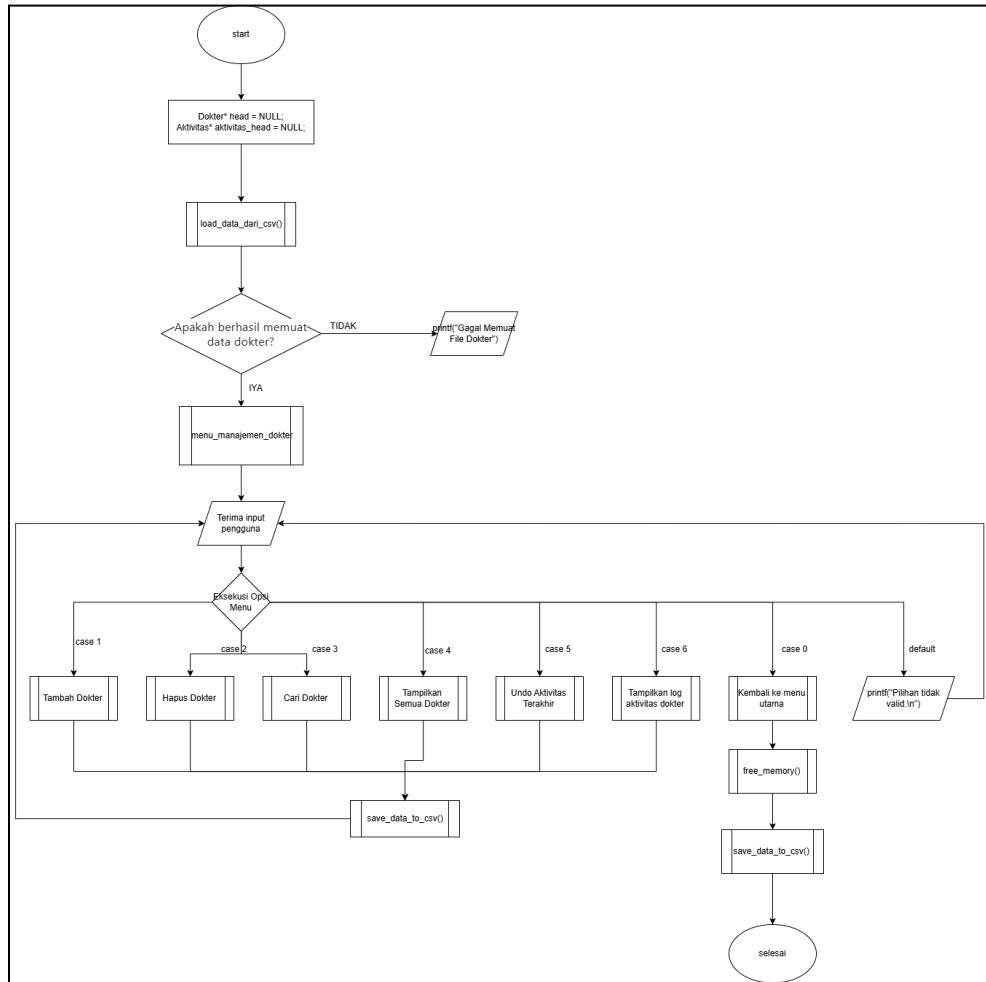


Gambar 5. Flowchart fungsi menu_laporan_kinerja

Flowchart ini menjelaskan alur kerja menu laporan kinerja jadwal dokter. Menu ini berfungsi sebagai antarmuka pengguna dengan fungsi yang disediakan untuk melihat kinerja laporan dokter yang dibuat program. Kinerja yang dapat dilihat adalah total shift dokter, preferensi dokter (maksimal shift, preferensi tanggal, dan preferensi waktu), pelanggaran terhadap preferensi, dan jumlah shift yang didapatkan tiap dokter pada parameter tertentu. Pertama, program akan membaca file jadwal_dokter.csv dan data_dokter.csv. Jika berhasil, program akan membuat laporan dan menyimpannya dalam laporan_dokter.csv. Kemudian program akan memberikan message pilihan fungsi yang bisa dilakukan untuk melihat kinerja jadwal. Jika pengguna memilih suatu fungsi, maka program akan membawa pengguna ke fungsi tersebut dan masuk ke sub program penampilan informasi. Jika pengguna memilih untuk keluar dari program, maka pengguna akan dibawa kembali ke menu utama.

3.1.2. Pengelolaan data dokter

a. Flowchart dokter.c

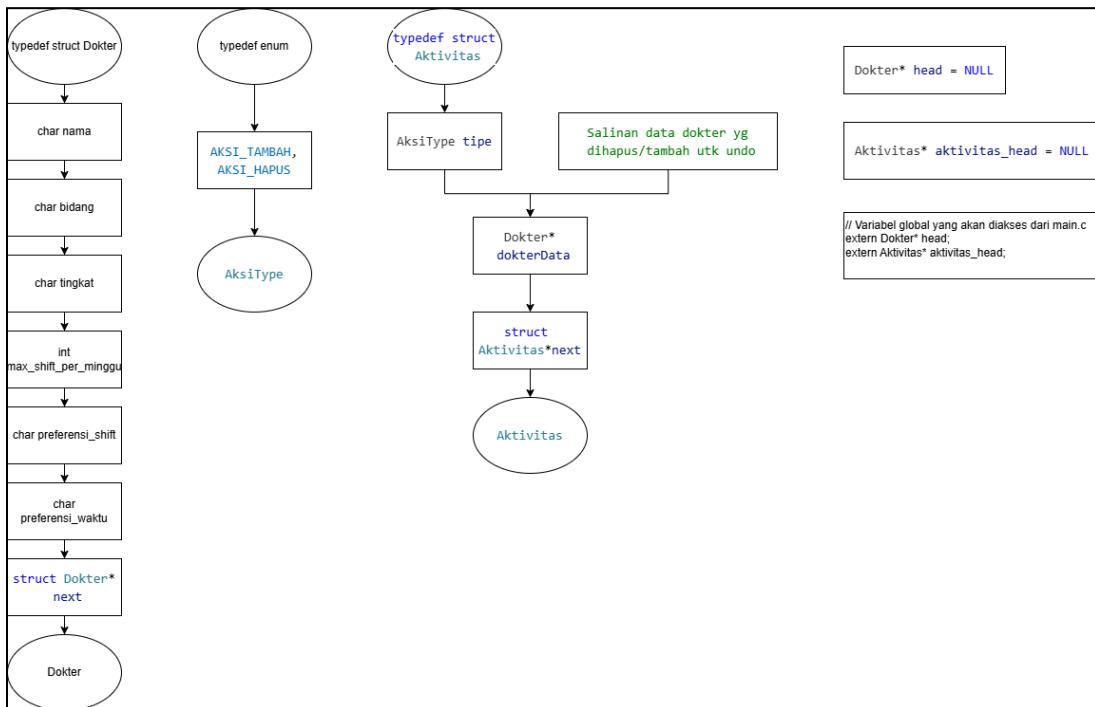


Gambar 6. Flowchart dokter.c

Program dimulai dengan menginisialisasi sistem dengan me-reset variabel global seperti `head` dan `aktivitas_head` dan memuat data dari file CSV menggunakan fungsi `load_data_dar_csv()`. Jika proses memuat data gagal, maka program akan menampilkan pesan error. Namun jika berhasil, maka program akan melanjutkan ke tampilan menu manajemen dokter yang menampilkan daftar opsi kepada pengguna, seperti tambah dokter, hapus dokter, dan cari dokter, tampilkan semua dokter, undo aktivitas terakhir dokter, tampilkan log aktivitas terakhir dokter, atau keluar dari main manajemen dokter. Setelah pengguna memilih salah satu opsi, program

akan mengesekusi fungsi yang sesuai dengan pilihan pengguna. Semua opsi menu dapat diakses secara berulang hingga pengguna memilih untuk kembali ke menu utama. Jika pengguna memilih kembali ke menu utama, maka program akan membebaskan semua memori yang dialokasikan dan menyimpan data akhir ke file CSV.

b. Desain Struktur Data

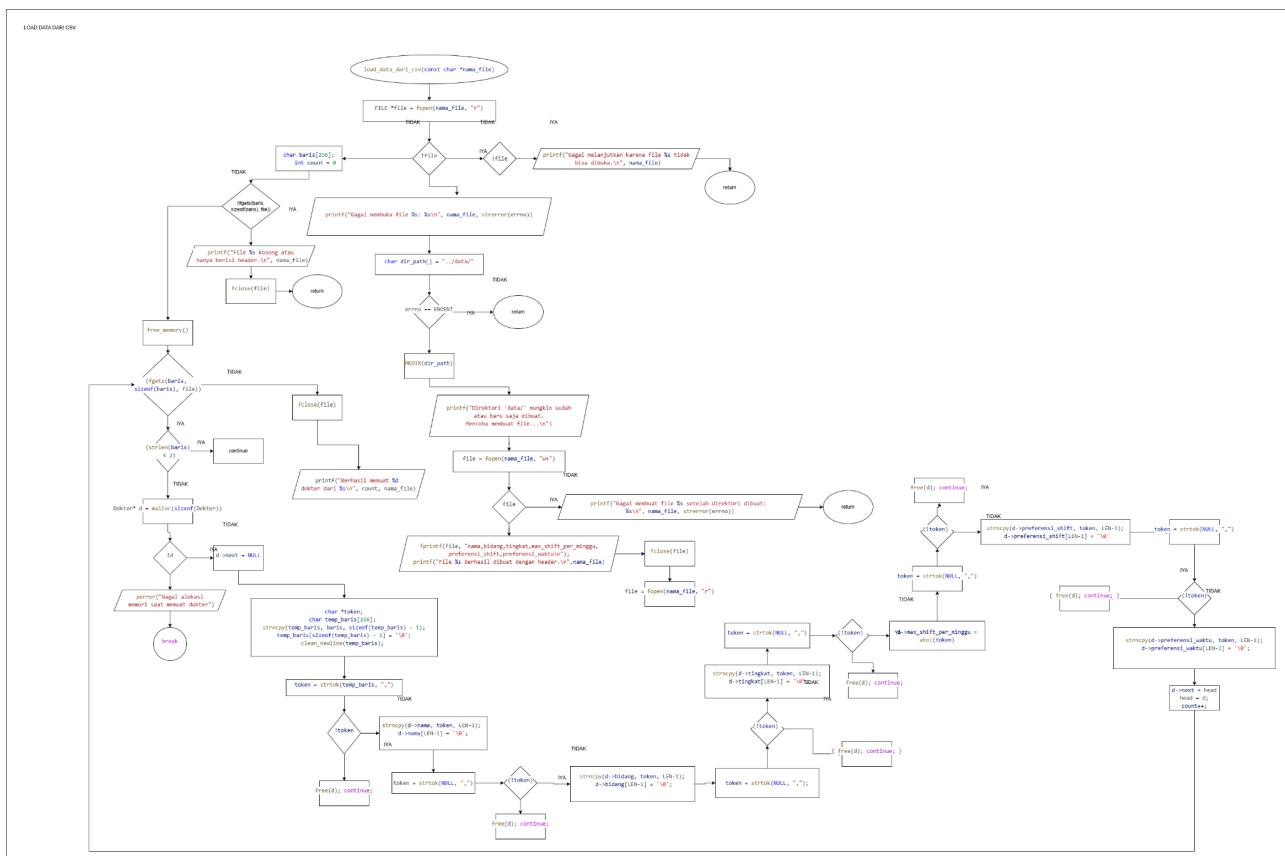


Gambar 7. Flowchart desain struktur data Dokter.c

Modul manajemen dokter menggunakan struktur data yang dirancang untuk menyimpan informasi dokter dan melacak aktivitas tambah/hapus dokter. Struktur utama dalam program ini adalah Dokter, yang digunakan untuk menyimpan data setiap dokter, termasuk nama, bidang, tingkat, jumlah maksimal shift per minggu, preferensi shift, dan preferensi waktu. Setiap elemen pada struktur tersebut dihubungkan dalam sebuah linked list melalui pointer next. Selain itu, modul ini juga menggunakan sebuah enumerasi bernama AksiType untuk merepresentasikan jenis aktivitas yang dilakukan pada data dokter, yaitu AKSI_TAMBAH untuk penambahan dokter dan AKSI_HAPUS untuk penghapusan dokter. Untuk melacak riwayat aktivitas tersebut, digunakan struktur Aktivitas. Struktur ini memiliki field tipe yang menunjukkan jenis aksi,

dokterData yang menyimpan salinan data dokter saat aktivitas terjadi, dan pointer next untuk menghubungkan aktivitas aktivitas lainnya dalam bentuk linked list. Sehingga fitur undo dapat diterapkan untuk membatalkan aktivitas terakhir sesuai dengan log aktivitas. Modul ini juga menggunakan dua variabel global, yaitu head dan aktivitas_head. Variabel head berfungsi untuk menunjuk ke awal linked list dokter, sedangkan aktivitas_head digunakan untuk menunjuk ke awal linked list aktivitas. Kedua variabel ini adalah pusat pengelolaan data dalam modul ini. Melalui kombinasi struktur - struktur data tersebut, modul ini dapat menyimpan, mengelola, dan melacak perubahan data dokter secara efisien.

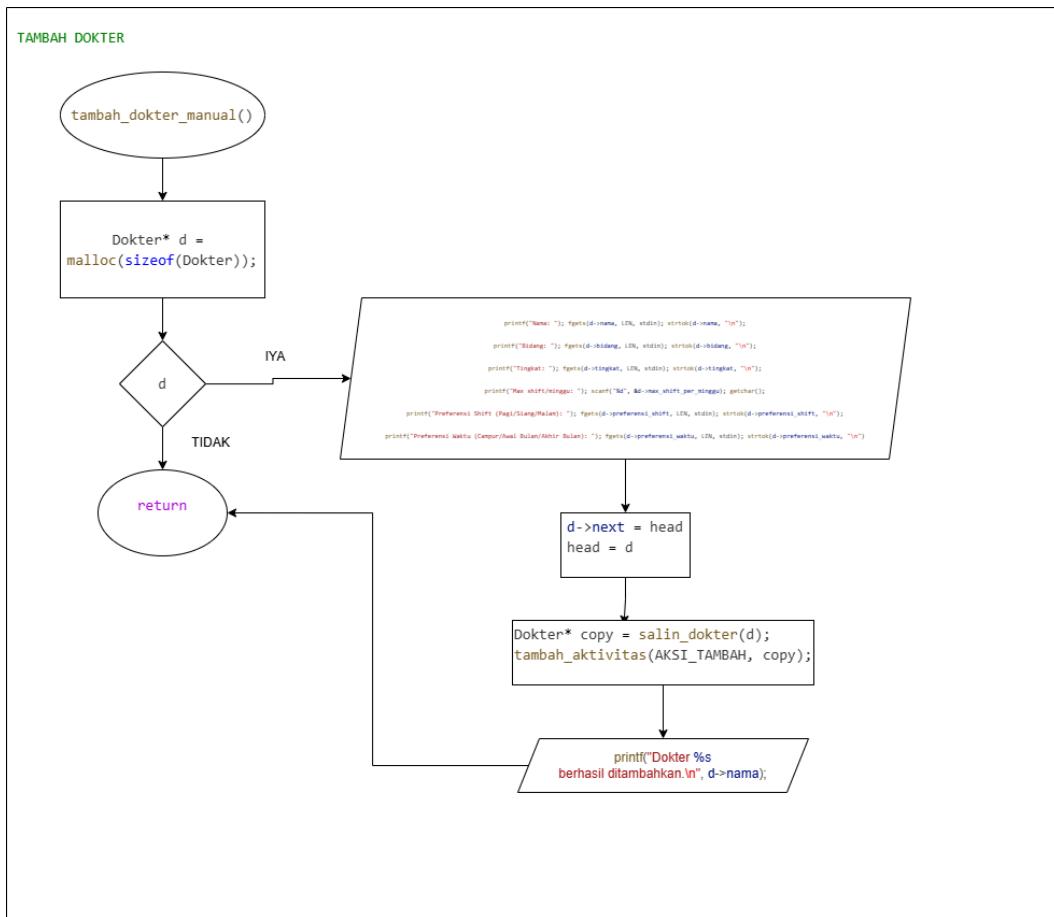
c. Fungsi load data dari csv



Gambar 8. Flowchart fungsi `load_data_daricsv()`

Fungsi `load_data_dari_csv()` ini dimulai dengan membuka file CSV sesuai dengan nama file. Jika file gagal dibuka sistem akan menampilkan pesan error dan deskripsi kesalahan menggunakan `strerror(errno)`. Setelah ini dilakukan pemeriksaan apakah error tersebut ENOENT(file atau direktori tidak ditemukan). Jika ya, maka program akan membuat direktori `../data/` menggunakan macro `MKDIR`. Setelah itu, sistem akan membuat file baru dengan mode "w+" agar bisa dibaca dan ditulis. Pada pembuatan file ini, header `nama,bidang,tingkat,max_shift_per_minggu,preferensi_shift,preferensi_waktu` akan langsung ditulis kedalam file untuk memastikan format data sudah benar. Setelah file berhasil dibuat, maka ditutup dan dibuka kembali menggunakan mode "r", agar program siap untuk proses pemrosesan selanjutnya. Jika file masih tetap tidak bisa dibuka fungsi akan langsung dihentikan. Jika file berhasil dibuka, langkah selanjutnya adalah membaca baris pertama sebagai header CSV dengan fungsi `fgets`. Jika header tidak ditemukan atau file kosong, maka sistem akan menampilkan pesan bahwa file kosong atau hanya berisi header, lalu menutup file dan mengakhiri fungsi. Setelah header dibaca, program akan membersihkan struktur data sebelumnya dengan memanggil fungsi `free_memory()` untuk memastikan linked list head benar - benar kosong sebelum memuat data baru. Setelah itu, dilakukan pembacaan baris demi baris dari file CSV menggunakan loop `while(fgets(...))`. Setiap baris yang dibaca akan dicek apakah baris tersebut adalah baris kosong dengan memeriksa panjang karakternya. Jika kurang dari 2 baris, maka baris tersebut akan dilewati. Untuk setiap baris yang valid, alokasi memori dilakukan untuk mode baru bertipe Dokter. Jika alokasi gagal, maka akan muncul pesan error dan proses dihentikan. Selanjutnya, salinan dari baris tersebut disimpan kedalam buffer `temp_baris` agar akan digunakan oleh fungsi `strok`. Fungsi `clean_newline` kemudian dipanggil untuk membersihkan karakter newline dari baris tersebut agar tidak mengganggu parsing. Baris kemudian diparse menjadi token - token menggunakan fungsi `strok`, dengan pemisah tanda koma. Setiap token akan dimasukkan kedalam field yang sesuai pada struct `Dokter`. Setiap kali token tidak ditemukan mana node yang dialokasikan akan dibatalkan penggunaannya dengan `free(D)` dan baris tersebut dilewati. Setelah semua field dokter terisi dengan benar maka node tersebut akan ditambahkan ke awal linked list (`head`) sehingga data baru muncul di posisi paling depan. Proses akan terus berulang sampai semua baris file selesai dibaca. Setelah selesai, file akan ditutup dan data dokter berhasil dimuat dari file CSV, serta pesan sukses ditampilkan.

d. Fungsi tambah dokter

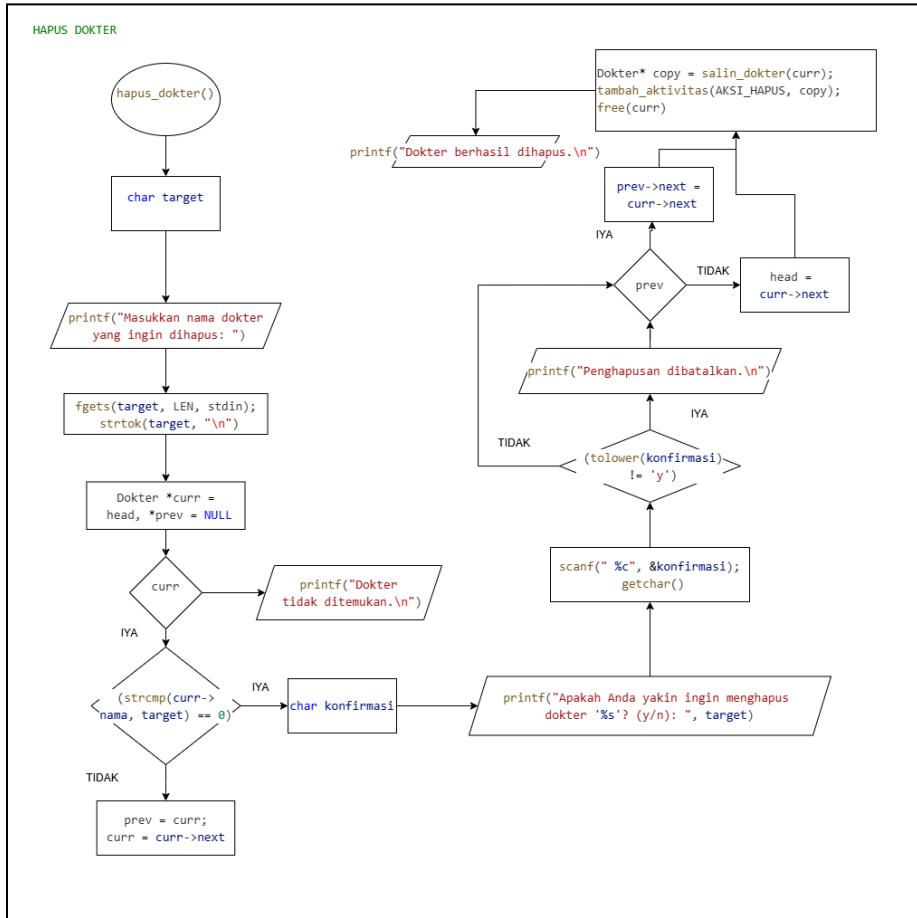


Gambar 9. Flowchart fungsi tambah_dokter_manual()

Fungsi tambah_dokter_manual ini dimulai dengan alokasi memori untuk struktur bertipe Dokter menggunakan malloc. Jika alokasi memori gagal maka program akan menampilkan pesan error menggunakan perror dan langsung keluar dari fungsi. Jika alokasi berhasil, maka sistem meminta pengguna untuk mengisi data dokter secara manual. Langkah pertama, program meminta nama dokter dan membaca input menggunakan fgets ke field d->nama. Setelah itu, karakter new line dihapus menggunakan fungsi clean_newline. Jika hasilnya kosong maka sistem akan menampilkan pesan bahwa nama tidak boleh kosong, membebaskan memori node tersebut dengan free(d), lalu fungsi dihentikan. Selanjutnya langkah serupa dilakukan untuk field bidang, tingkat, preferensi_shift, dan preferensi_waktu. Masing - masing input dibaca menggunakan fgets, dibersihkan newline nya, dan diperiksa apakah isinya kosong. Jika salah satu dari input ini

kosong, sistem akan menampilkan pesan kesalahan yang sesuai, membebaskan memori, dan mengakhiri fungsi. Untuk field numerik max_shift_per_minggu, program menampilkan prompt "Max shift/minggu: ", lalu membaca input menggunakan scanf("%d", &d->max_shift_per_minggu). Jika pembacaan gagal maka sistem akan menampilkan pesan bahwa input tidak valid. Lalu sistem akan membersihkan buffer dengan while(getchar()!="\n"). Hal ini untuk mencegah loop yang tak berujung, membebaskan memori, dan mengakhiri fungsi. Setelah itu, karakter newline tersisa dikonsumsi dengan getchar() agar tidak mengganggu input berikutnya. Setelah semua field dokter berhasil diisi dengan valid, node baru ini ditambahkan ke linked list sebagai head baru. Dokter baru ditempatkan di awal daftar linked list (d->next = head; head = d;). Sebelum dokter benar-benar disimpan ke dalam file CSV, fungsi membuat salinan dari dokter ini menggunakan salin_dokter(d) agar bisa digunakan oleh fitur undo. Jika salinan berhasil dibuat, aktivitas penambahan dokter dicatat ke dalam stack aktivitas melalui fungsi tambah_aktivitas(AKSI_TAMBAH, copy_for_undo). Jika gagal membuat salinan, sistem tetap melanjutkan proses namun menampilkan peringatan bahwa fitur undo tidak bisa menyimpan aksi ini. Setelah selesai memproses penambahan dokter ke linked list dan mencatat aktivitas undo, fungsi langsung menyimpan seluruh data dokter ke file CSV menggunakan save_data_to_csv(OUTPUT_FILE_DOKTER) agar perubahan tersimpan secara permanen. Setelah itu program menampilkan pesan konfirmasi bahwa dokter dengan nama tertentu berhasil ditambahkan, dan fungsi selesai dijalankan.

e. Fungsi hapus dokter

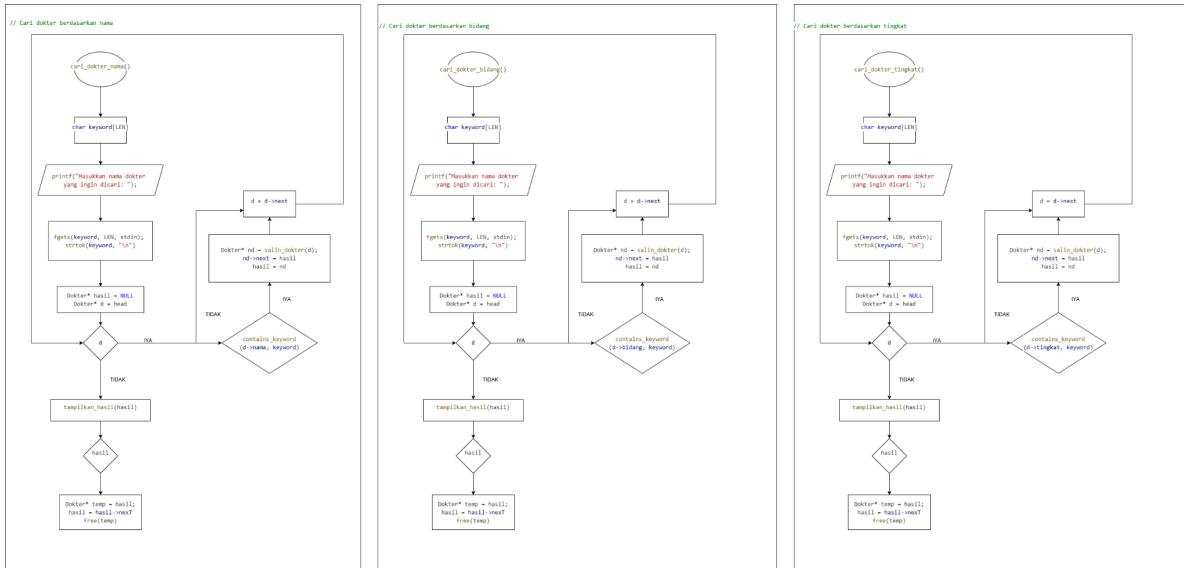


Gambar 10. Flowchart fungsi hapus_dokter()

Fungsi dimulai dengan deklarasi variabel target[LEN] untuk menyimpan nama dokter yang ingin dihapus. Program menampilkan prompt "Masukkan nama dokter yang ingin dihapus: " dan membaca input pengguna menggunakan fgets. Setelah itu, karakter newline pada akhir input dibersihkan dengan fungsi clean_newline(target). Jika hasilnya kosong (panjang string 0), sistem menampilkan pesan bahwa nama dokter tidak boleh kosong, lalu fungsi langsung dihentikan. Jika nama valid, program memulai pencarian dokter dalam linked list dengan menggunakan dua pointer: curr sebagai pointer utama yang berjalan sepanjang list, dan prev yang selalu mengikuti satu langkah di belakang curr untuk memudahkan operasi penghapusan node di tengah atau akhir list. Dengan loop while (curr), setiap node dalam linked list dicek apakah sesuai dengan nama dokter yang dicari menggunakan strcmp(curr->nama, target). Saat dokter ditemukan (strcmp ==

0), program akan meminta konfirmasi kepada pengguna melalui prompt "Apakah Anda yakin ingin menghapus dokter 'X'? (y/n): ". Input jawaban dibaca dengan `scanf(" %c", &konfirmasi)`. Jika input tidak valid (misalnya bukan karakter), sistem menampilkan pesan kesalahan, membersihkan buffer input dengan `while (getchar() != '\n')`, dan mengakhiri fungsi. Selain itu, karakter newline tersisa dikonsumsi dengan `getchar()` agar tidak mengganggu input berikutnya. Jika jawaban bukan 'y' atau 'Y', maka penghapusan dibatalkan dan fungsi langsung diakhiri. Namun jika pengguna menyetujui penghapusan, sistem melakukan pemutusan koneksi node dari linked list: jika dokter berada di tengah/akhir list, `prev->next` diarahkan ke `curr->next`; sedangkan jika dokter berada di head (awal list), maka `head` diubah menjadi `curr->next`. Sebelum membebaskan memori node dokter yang akan dihapus, program membuat salinan data dokter tersebut menggunakan fungsi `salin_dokter(curr)`. Salinan ini digunakan untuk fitur undo. Jika berhasil, aktivitas penghapusan dicatat ke dalam stack aktivitas melalui `tambah_aktivitas(AKSI_HAPUS, copy_for_undo)`. Jika gagal membuat salinan, sistem tetap melanjutkan proses namun menampilkan peringatan bahwa fitur undo tidak bisa menyimpan aksi ini. Setelah salinan disimpan, memori node dokter asli dibebaskan dengan `free(curr)`. Lalu, sistem menampilkan pesan "Dokter berhasil dihapus.", dan langsung menyimpan perubahan ke file CSV menggunakan `save_data_to_csv(OUTPUT_FILE_DOKTER)` agar data terakhir tersimpan secara permanen. Jika sampai akhir loop while dokter tidak ditemukan, sistem menampilkan pesan "Dokter tidak ditemukan." dan fungsi selesai tanpa melakukan penghapusan.

f. Fungsi cari doter berdasarkan nama, bidang, dan tingkat

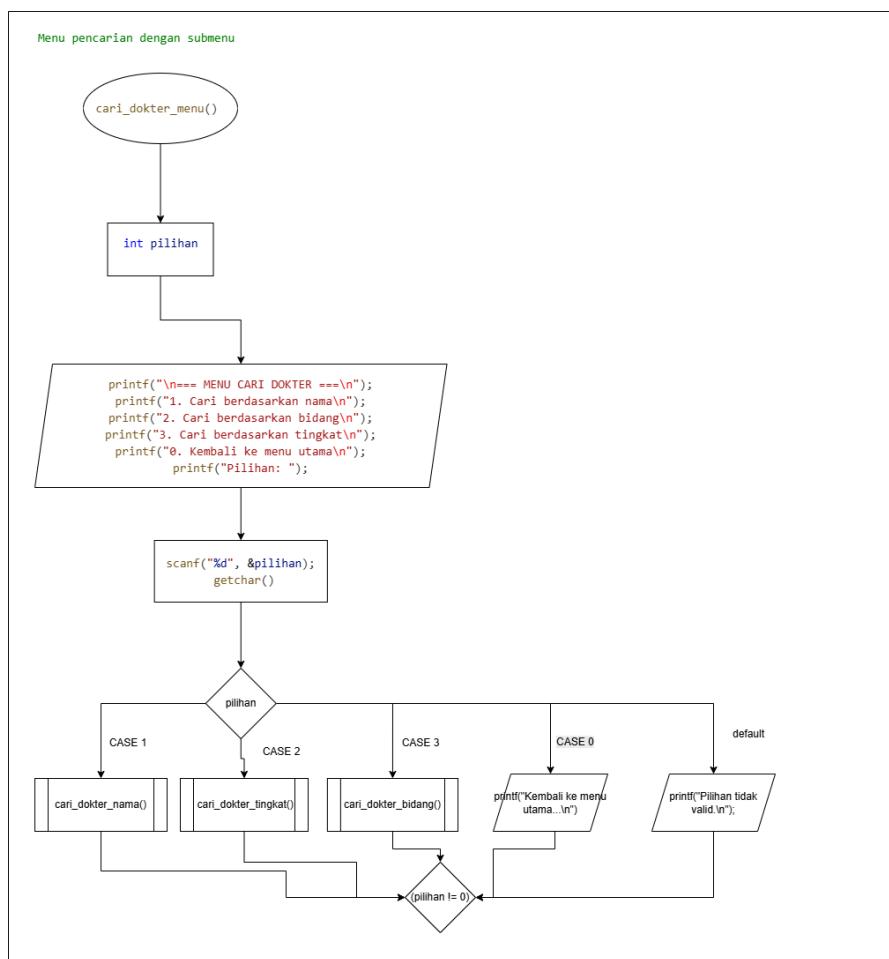


Gambar 11. Flowchart fungsi `cari_dokter_nama()`, `cari_dokter_bidang()`, `cari_dokter_tingkat()`

Fungsi-fungsi (`cari_dokter_nama`, `cari_dokter_bidang`, dan `cari_dokter_tingkat`) memiliki struktur alur logika yang sama, hanya berbeda pada field yang dicari (nama, bidang, atau tingkat). Setiap fungsi dimulai dengan deklarasi variabel `keyword[LEN]` untuk menyimpan kata kunci pencarian. Program menampilkan prompt sesuai dengan jenis pencarian, seperti "Masukkan nama dokter yang ingin dicari: ", "Masukkan bidang dokter yang ingin dicari: ", atau "Masukkan tingkat dokter yang ingin dicari: ". Input pengguna dibaca menggunakan `fgets`, lalu karakter newline di akhir baris dibersihkan dengan fungsi `clean_newline(keyword)`. Jika panjang string `keyword` menjadi nol setelah dibersihkan, sistem akan menampilkan pesan bahwa `keyword` tidak boleh kosong dan langsung mengakhiri fungsi. Jika input valid, program melanjutkan ke langkah berikutnya. Inisialisasi linked list hasil pencarian (`Dokter* hasil = NULL`) dan pointer `d` yang digunakan untuk menjelajahi linked list utama (`head`). Dengan loop while (`d`), setiap node dalam linked list utama dicek apakah field yang bersesuaian (nama, bidang, atau tingkat) mengandung kata kunci pencarian. Pemeriksaan dilakukan dengan menggunakan fungsi `contains_keyword(field, keyword)`, yang memeriksa apakah `keyword` merupakan bagian dari nilai field tersebut secara *case-insensitive* atau *case-sensitive*. Jika kondisi cocok, maka sistem membuat salinan dari node dokter tersebut menggunakan fungsi `salin_dokter(d)`. Jika alokasi

berhasil, salinan dokter ditambahkan ke awal linked list hasil agar hasil pencarian tersusun secara terbalik. Setelah seluruh node dalam linked list utama diproses, fungsi tampilkan_hasil(hasil) dipanggil untuk menampilkan semua dokter yang ditemukan kepada pengguna.. Setelah ditampilkan, program langsung membebaskan seluruh memori yang dialokasikan untuk linked list hasil menggunakan loop while yang melepaskan setiap node satu per satu dengan free(temp).

g. Fungsi pencarian dengan submenu

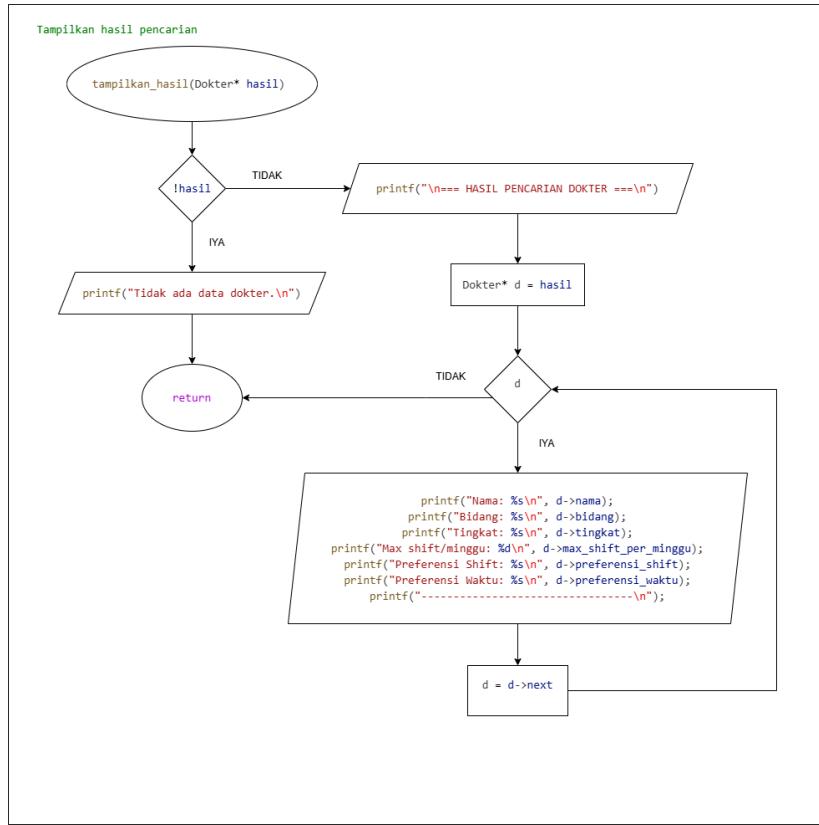


Gambar 12. Flowchart fungsi cari_dokter_menu()

Fungsi dimulai dengan menampilkan antarmuka menu pencarian dokter kepada pengguna. Menu ini berisi tiga opsi yaitu berdasarkan nama , bidang , atau tingkat , serta satu opsi untuk kembali

ke menu utama (0). Tampilan menu ini dilakukan di dalam loop do-while. Setelah menampilkan daftar pilihan, program akan menunggu input dari pengguna menggunakan `scanf("%d", &pilihan)` untuk membaca angka sebagai pilihan menu. Jika input tidak valid maka sistem akan menampilkan pesan "Input tidak valid. Harap masukkan angka." Setelah itu membersihkan buffer input dengan perintah `while(getchar()!='\n')`. Setelah itu, loop akan langsung melanjutkan ke iterasi berikutnya tanpa menjalankan block switch. Jika input berhasil dibaca secara angka, maka karakter newliine yang tersisa di buffer input dikonsumsi dengan `getchar()` agar tidak mengganggu operasi berikutnya. Selanjutnya program akan masuk ke blok switch untuk mengevaluasi nilai pilihan. Jika pengguna memilih 1 , maka fungsi `cari_dokter_nama()` dipanggil untuk memulai proses pencarian dokter berdasarkan nama. Jika pengguna memilih 2 , maka fungsi `cari_dokter_bidang()` dipanggil untuk mencari dokter berdasarkan bidang. Jika pengguna memilih 3 , maka fungsi `cari_dokter_tingkat()` dipanggil untuk mencari dokter berdasarkan tingkat. Jika pengguna memilih 0 , sistem menampilkan pesan "Kembali ke menu manajemen dokter...", dan keluar dari loop do-while, sehingga eksekusi fungsi `cari_dokter_menu()` selesai dan kontrol kembali ke fungsi pemanggil. Jika pengguna memasukkan angka lain selain 0–3, sistem menampilkan pesan "Pilihan tidak valid." dan loop akan berlanjut untuk menampilkan ulang menu. Setiap kali salah satu fungsi pencarian (`cari_dokter_nama`, `cari_dokter_bidang`, atau `cari_dokter_tingkat`) dipanggil, alur eksekusi akan beralih ke fungsi tersebut. Fungsi-fungsi tersebut akan meminta pengguna memasukkan keyword pencarian, melakukan pencarian dalam linked list dokter, menampilkan hasil pencarian, lalu kembali ke menu pencarian setelah selesai. Setelah pengguna memilih untuk kembali ke menu utama, fungsi `cari_dokter_menu()` selesai dijalankan.

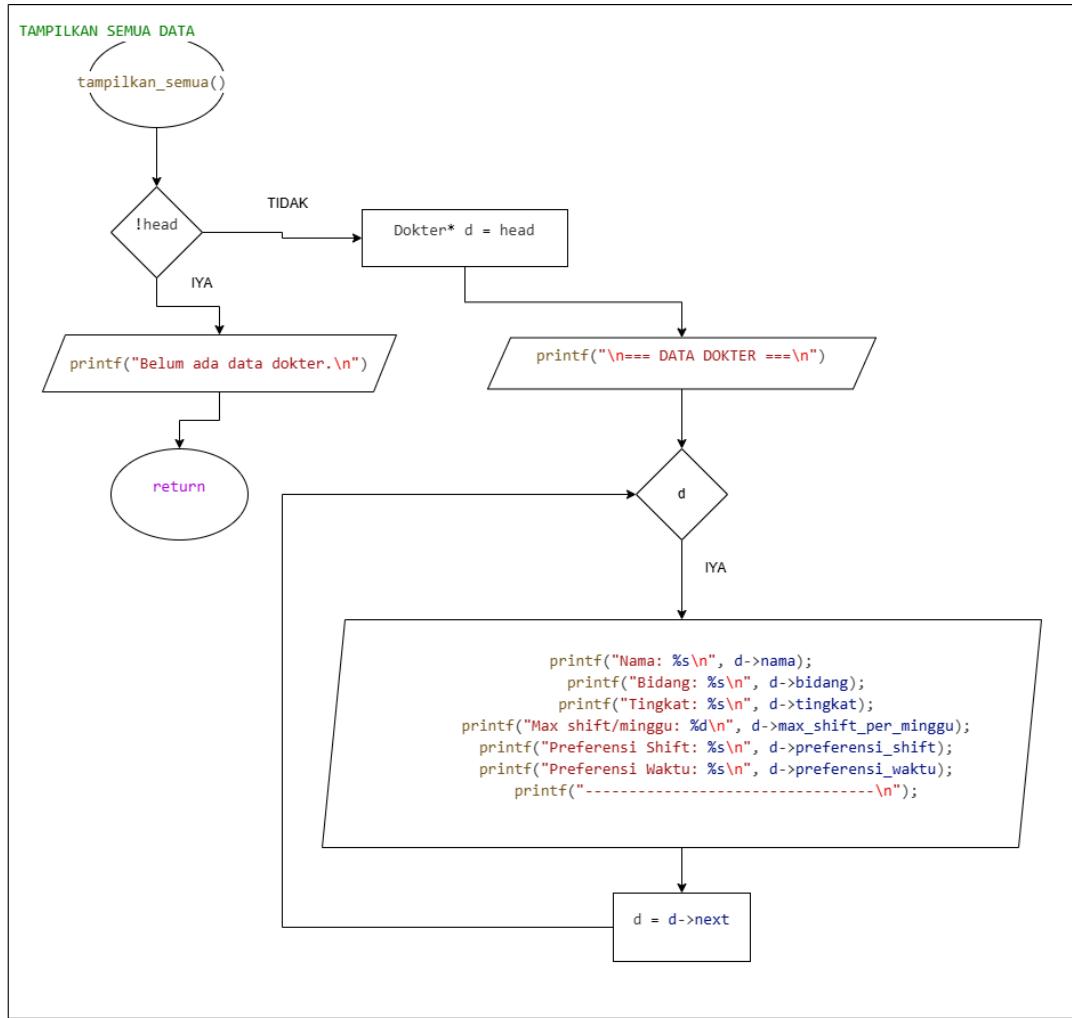
h. Fungsi menampilkan hasil pencarian



Gambar 13. Flowchart fungsi tampilan_hasil(Dokter* hasil)

Fungsi tampilan_hasil(Dokter* hasil) bertugas untuk menampilkan daftar dokter yang ditemukan setelah proses pencarian. Jika pointer hasil bernilai NULL, artinya tidak ada dokter yang cocok, dan sistem langsung menampilkan pesan "Tidak ada data dokter yang cocok dengan kriteria pencarian.". Namun jika ada hasil, fungsi akan menampilkan header "==== HASIL PENCARIAN DOKTER ===" lalu melakukan iterasi menggunakan pointer d untuk mencetak setiap field dokter (nama, bidang, tingkat, max_shift_per_minggu, preferensi_shift, dan preferensi_waktu) ke layar secara terurut. Setiap dokter dipisahkan dengan garis pemisah "-----". Proses berlanjut hingga seluruh node dalam linked list hasil selesai ditampilkan.

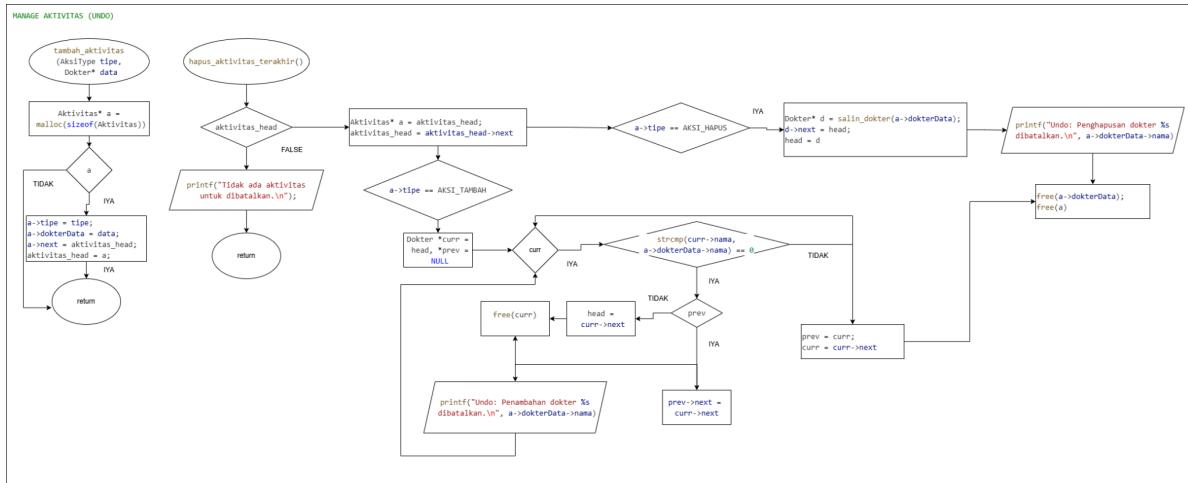
i. Fungsi menampilkan semua data



Gambar 14. Flowchart `tampilkan_semua()`

Fungsi `tampilkan_semua()` digunakan untuk menampilkan seluruh data dokter yang tersimpan dalam linked list. Jika `head` bernilai `NULL`, artinya belum ada data dokter, dan sistem akan menampilkan pesan "Belum ada data dokter.". Jika ada data, fungsi menampilkan header "==== DATA DOKTER ===" lalu melakukan iterasi dari awal linked list menggunakan pointer `d`. Setiap dokter ditampilkan dengan nomor urut, disertai detail lengkap field-nya: nama, bidang, tingkat, `max_shift_per_minggu`, `preferensi_shift`, dan `preferensi_waktu`. Setiap entri dipisahkan dengan garis pemisah untuk memudahkan pembacaan. Proses ini berlanjut hingga semua dokter dalam linked list selesai ditampilkan.

j. Fungsi manage aktivitas (UNDO)

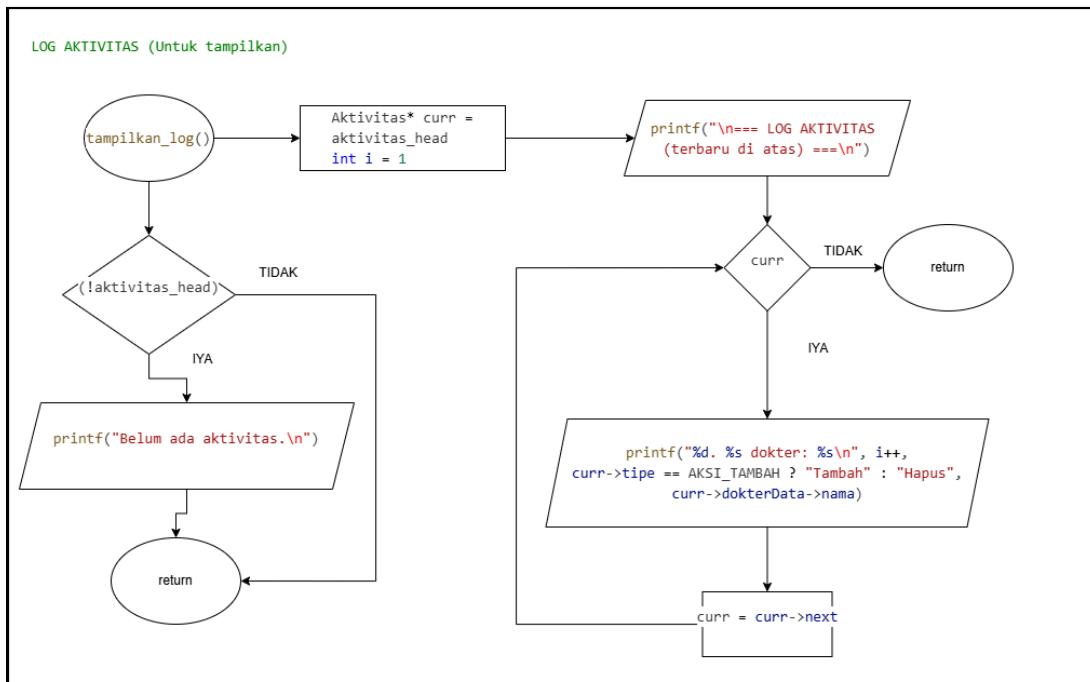


Gambar 15. Flowchart fungsi tambah_aktivitas(AksiType tipe, Dokter* data)

Fungsi tambah_aktivitas(AksiType tipe, Dokter* data) digunakan untuk menambahkan aktivitas baru ke dalam struktur stack aktivitas (undo). Dengan fungsi ini pengguna dapat membatalkan aksi sebelumnya. Proses ini dimulai dengan mengalokasikan memori pada elemen 'Aktivitas' lalu menggunakan 'malloc'. Jika alokasi gagal (pointer 'a' bernilai NULL), maka fungsi langsung membebaskan pointer 'data' (yang merupakan salinan dari dokter asli) dan keluar dari fungsi. Jika alokasi berhasil, tipe aktivitas ('tipe') disimpan ke dalam node tersebut, lalu pointer 'dokterData' pada node tersebut diarahkan ke 'data' (salinan data dokter). Setelah itu dilakukan pemeriksaan ulang apakah 'dokterData' valid atau tidak, jika tidak valid, maka node 'a' dibebaskan dan fungsi dihentikan. Jika semua langkah berhasil, node 'a' ditambahkan ke awal linked list 'aktivitas_head', lalu menjadi head baru dari stack aktivitas. Fungsi 'hapus_aktivitas_terakhir()' berfungsi sebagai mekanisme "Undo" dengan menghapus aktivitas terakhir yang tersimpan dalam stack aktivitas ('aktivitas_head'). Langkah pertama adalah memeriksa apakah 'aktivitas_head' kosong. Jika iya, maka cetak pesan bahwa tidak ada aktivitas yang bisa dibatalkan, lalu fungsi berakhir. Jika ada aktivitas, node 'aktivitas_head' saat ini disimpan ke variabel lokal 'a', lalu 'aktivitas_head' dipindahkan ke node berikutnya ('next'). Hal ini dilakukan untuk menghapus node paling atas dari stack. Selanjutnya, diperiksa jenis aktivitas ('a->tipe'). Jika tipe aktivitas adalah 'AKSI_TAMBAH' maka program mencari dokter dalam daftar utama ('head') yang memiliki nama dan bidang sama dengan 'a->dokterData' (salinan

dokter yang ditambahkan). Dalam melakukan pencarian ini digunakan algoritma traversal linked list dengan pointer 'curr' dan 'prev' untuk mempertahankan referensi ke node sebelumnya. Jika dokter ditemukan, maka ia dihapus dari linked list dengan memperbarui koneksi 'prev->next'. Jika dokter berada di kepala list, maka 'head' dipindahkan ke 'curr->next'. Setelah dokter dihapus dari linked list, memori untuk node dokter tersebut dibebaskan menggunakan 'free(curr)'. Lalu pesan konfirmasi pembuatan penambahan dokter ditampilkan kepada pengguna. Jika dokter tidak ditemukan, pesan error ditampilkan. Jika tipe aktivitas adalah 'AKSI_HAPUS', maka dokter yang disimpan dalam 'a->dokterData' akan dikembalikan ke daftar dokter aktif ('head') dengan menyambungkan 'dokterData->next' ke 'head' saat ini, lalu 'head' diatur ke 'a->dokterData'. Setelah itu, pesan konfirmasi pembatalan penghapusan dokter ditampilkan kepada pengguna. Setelah aktivitas selesai diproses, node 'a' sendiri dibebaskan menggunakan 'free(a)' untuk mencegah memory leak. Terakhir, perubahan pada daftar dokter disimpan ke file CSV menggunakan fungsi 'save_data_to_csv(OUTPUT_FILE_DOKTER)'.

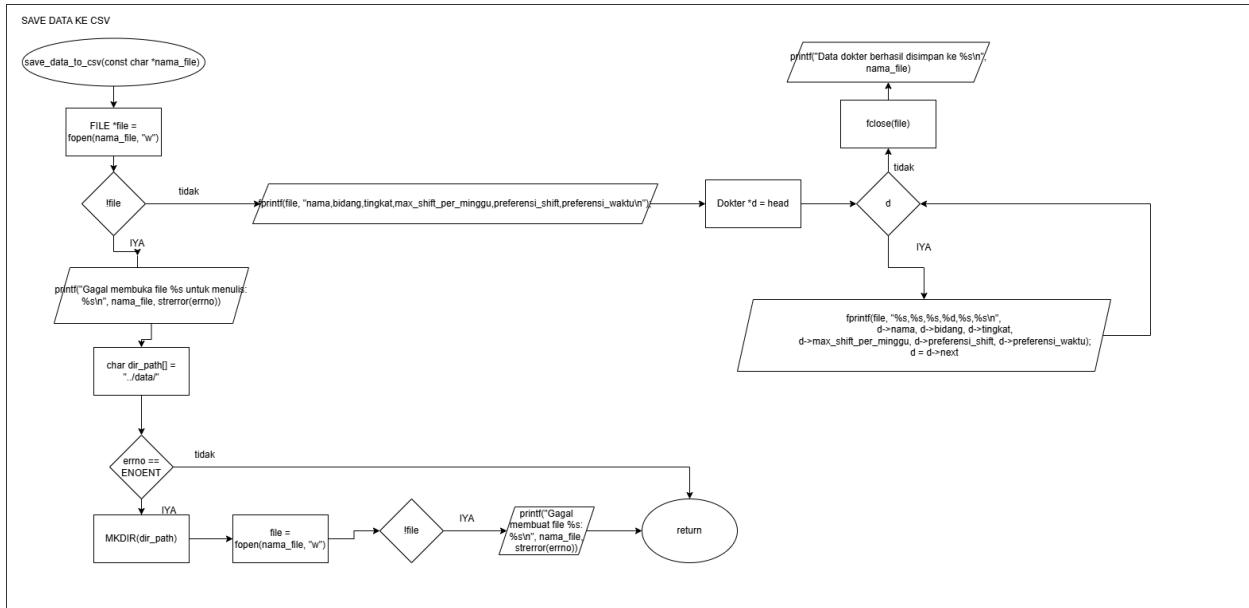
k. Fungsi log aktivitas



Gambar 16. Flowchart fungsi *tampilkan_log()*

Fungsi tampilkan_log() bertujuan untuk menampilkan riwayat aktivitas (penambahan atau penghapusan dokter) yang telah dilakukan selama sesi program berlangsung. Fungsi ini bekerja dengan menggunakan struktur linked list aktivitas_head yang menyimpan setiap aktivitas dalam bentuk stack (dengan urutan terbaru di depan). Pada awal eksekusi, fungsi memeriksa apakah aktivitas_head bernilai NULL. Jika iya, artinya belum ada log aktivitas yang tersimpan, dan sistem akan langsung menampilkan pesan "Belum ada aktivitas." serta mengakhiri fungsi. Namun jika log aktivitas tersedia, fungsi menampilkan header "==== LOG AKTIVITAS (terbaru di atas) ===". Selanjutnya, sebuah pointer bernama curr digunakan untuk menjelajahi linked list aktivitas dari awal (aktivitas_head). Dengan bantuan loop while (curr), setiap node aktivitas diperiksa secara berurutan. Pada tiap iterasi, sistem memastikan bahwa field dokterData pada node aktivitas tidak bernilai NULL. Jika data dokter valid, fungsi menampilkan nomor urut aktivitas, jenis aksi (apakah itu "Tambah" atau "Hapus", bergantung pada nilai curr->tipe), serta nama dokter yang terlibat. Jika sebaliknya, yaitu dokterData bernilai NULL, maka sistem tetap menampilkan entri log dengan pesan alternatif "Aktivitas tidak diketahui (data dokter hilang)". Setelah semua node aktivitas ditampilkan, pointer curr berpindah ke node berikutnya (curr = curr->next) hingga seluruh log selesai diproses. Hasil akhirnya adalah daftar aktivitas lengkap yang dapat digunakan oleh pengguna untuk melacak perubahan pada data dokter.

1. Fungsi save data ke csv



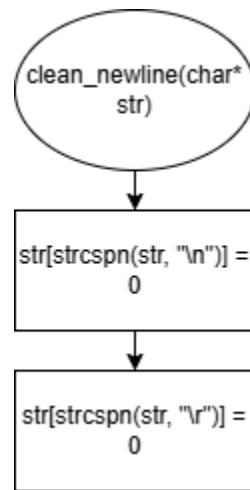
Gambar 17. Flowchart fungsi save_data_to_csv(const char *nama_file)

Fungsi ini bertujuan untuk menyimpan seluruh data dokter yang tersimpan dalam linked list (head) ke dalam sebuah file CSV dengan nama sesuai parameter nama_file. Proses dimulai dengan membuka file menggunakan fopen(nama_file, "w"), di mana mode "w" akan membuat file baru atau menimpa file lama jika sudah ada. Jika pembukaan file gagal, sistem menampilkan pesan error beserta deskripsi kesalahan menggunakan strerror(errno). Selanjutnya dilakukan pemeriksaan jenis error: jika error tersebut adalah ENOENT (file atau direktori tidak ditemukan), maka program mencoba membuat direktori/data/ menggunakan macro MKDIR(dir_path). Setelah direktori dibuat, sistem kembali mencoba membuka file dalam mode "w". Jika pembuatan file tetap gagal setelah langkah ini, sistem menampilkan pesan kesalahan dan mengakhiri fungsi. Namun jika berhasil, proses penyimpanan data dapat dilanjutkan. Langkah berikutnya adalah menulis header CSV ke dalam file menggunakan fprintf(file, "nama,bidang,tingkat,...\n") agar file memiliki struktur yang benar. Setelah itu, fungsi mulai menjelajahi linked list dokter dari awal (Dokter *d = head) hingga akhir menggunakan loop while (d). Pada setiap iterasi, field-field dokter (nama, bidang, tingkat, max_shift_per_minggu, preferensi_shift, dan preferensi_waktu) ditulis ke dalam file CSV dan dipisahkan dengan tanda koma (,), diikuti dengan karakter newline \n untuk memisahkan tiap baris data. Pointer d

kemudian berpindah ke node berikutnya ($d = d->next$) hingga seluruh dokter dalam linked list selesai ditulis. Setelah semua data selesai ditulis, file ditutup dengan `fclose(file)`. Terakhir, fungsi menampilkan pesan "Data dokter berhasil disimpan ke [nama_file]" sebagai konfirmasi bahwa operasi penyimpanan telah berhasil dilakukan.

m. Fungsi tambahan

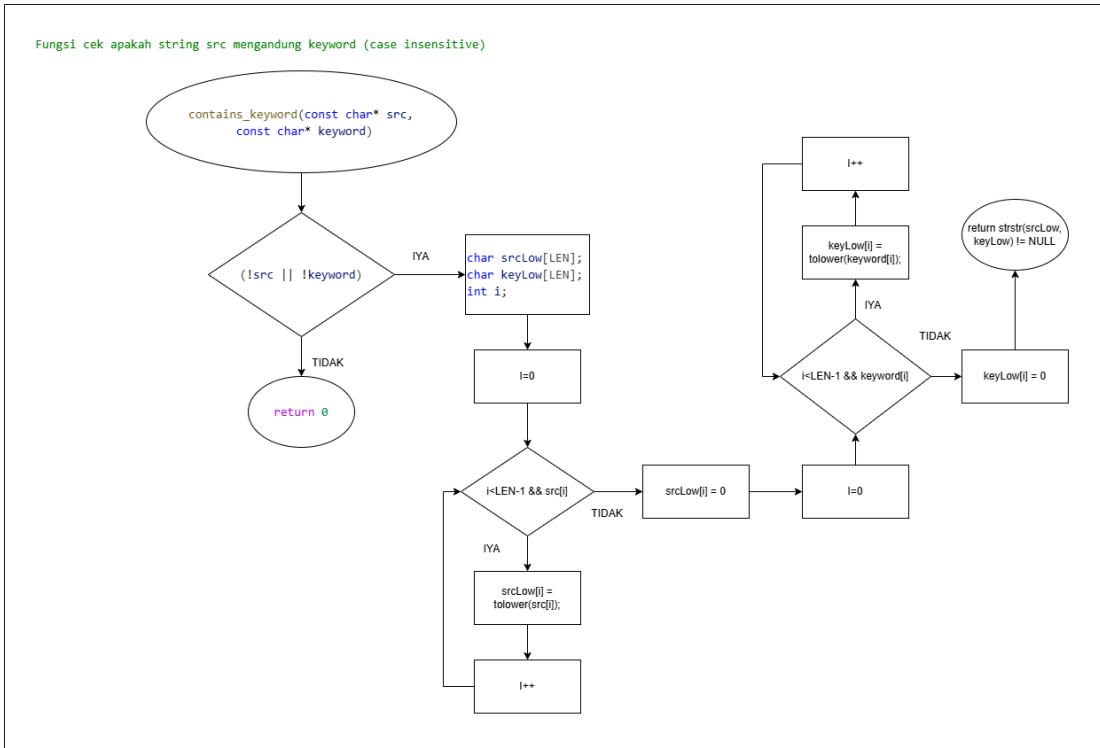
- Fungsi untuk membersihkan newline dari fgets



Gambar 18. Flowchart fungsi `clean_newline(char* str)`

Fungsi `clean_newline(char* str)` digunakan untuk membersihkan karakter newline (\n) dan carriage return (\r) dari akhir sebuah string. Fungsi ini bekerja dengan mencari posisi karakter \n dalam string menggunakan `strcspn(str, "\n")`, lalu menggantinya dengan null terminator (\0) untuk memotong string di titik tersebut. Langkah yang sama dilakukan untuk karakter \r. Dengan demikian, string menjadi lebih bersih dan siap untuk diproses lebih lanjut.

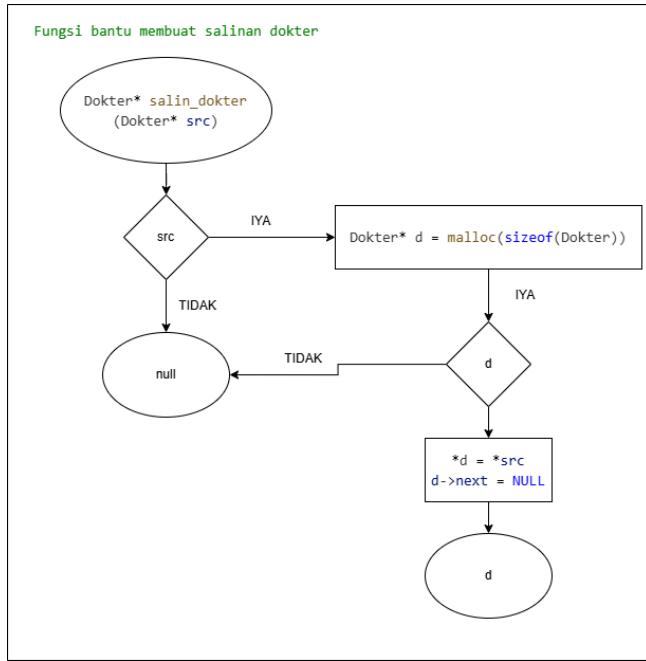
- Fungsi untuk cek apakah string src mengandung keyword(case insensitive)



Gambar 19. Flowchart fungsi `contains_keyword(const char* src, const char* keyword)`

Fungsi `contains_keyword(const char* src, const char* keyword)` digunakan untuk memeriksa apakah sebuah string (src) mengandung kata kunci tertentu (keyword), dengan pencarian yang tidak sensitif terhadap huruf besar/kecil . Pertama, fungsi melakukan pengecekan apakah kedua pointer src dan keyword tidak bernilai NULL. Jika salah satunya NULL, fungsi langsung mengembalikan nilai 0 (false). Selanjutnya, dua buffer lokal (srcLow dan keyLow) dibuat untuk menyimpan salinan dari src dan keyword. Panjang salinan dibatasi hingga LEN - 1 karakter agar aman dari buffer overflow, lalu null terminator (\0) ditambahkan di akhir masing-masing buffer. Setelah itu, semua karakter dalam srcLow dan keyLow diubah menjadi huruf kecil menggunakan fungsi `tolower()` agar pencarian bersifat case-insensitive. Fungsi `strstr()` kemudian digunakan untuk memeriksa apakah keyLow ditemukan sebagai substring dari srcLow. Jika ditemukan, fungsi mengembalikan nilai 1 (true); jika tidak, maka mengembalikan 0 (false).

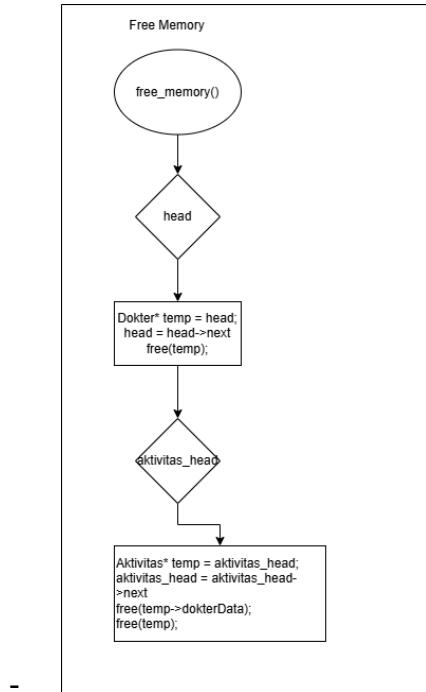
- Fungsi membantu membuat salinan dokter



Gambar 20. Flowchart fungsi salin_dokter(Dokter* src)

Fungsi `salin_dokter(Dokter* src)` bertujuan untuk membuat salinan dari sebuah node dokter (`Dokter*`). Pertama, fungsi memeriksa apakah pointer `src` bernilai `NULL`. Jika ya, fungsi langsung mengembalikan nilai `NULL`. Jika `src` valid, alokasi memori dilakukan untuk node baru menggunakan `malloc`. Jika alokasi gagal (misalnya karena tidak cukup memori), fungsi juga mengembalikan `NULL`. Jika alokasi berhasil, seluruh isi `src` disalin ke node baru menggunakan `memcpy`, sehingga semua field seperti nama, bidang, tingkat, `max_shift_per_minggu`, dan lainnya ikut tersalin. Pointer `next` dari node baru diatur menjadi `NULL` agar tidak menunjuk ke alamat sembarang. Terakhir, fungsi mengembalikan pointer ke node baru yang sudah disalin siap digunakan, misalnya untuk fitur undo atau pencarian.

- Fungsi free memory

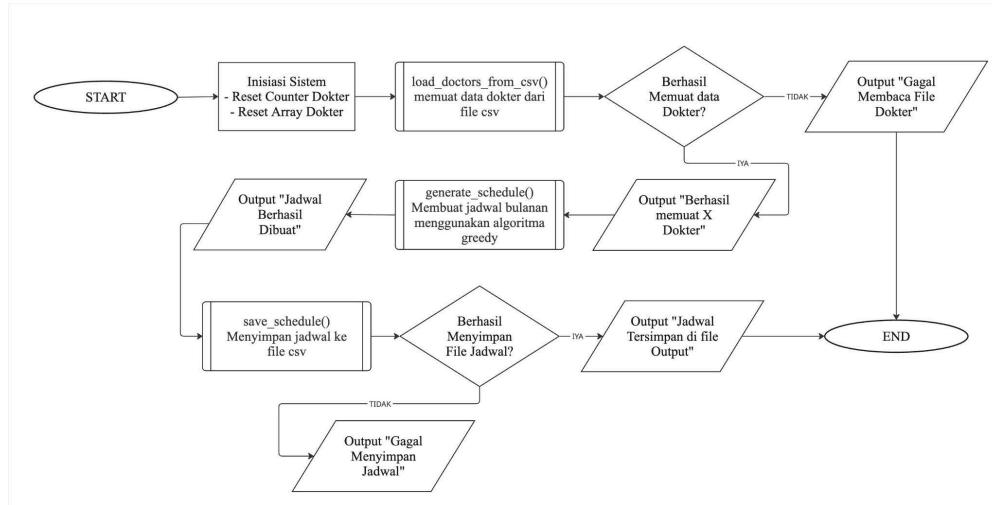


Gambar 21. Flowchart fungsi free_memory()

Fungsi `free_memory()` digunakan untuk membebaskan seluruh memori yang dialokasikan secara dinamis dalam program, baik itu untuk linked list dokter (`head`) maupun stack aktivitas (`aktivitas_head`). Pertama, fungsi melepaskan semua node dalam linked list dokter dengan loop while (`head`). Setiap node disimpan ke dalam variabel sementara `temp`, lalu pointer `head` dipindahkan ke node berikutnya dan `temp` dibebaskan menggunakan `free(temp)`. Selanjutnya, fungsi membersihkan stack aktivitas dengan cara yang sama. Setiap node aktivitas disimpan dalam `temp`, pointer `aktivitas_head` dipindahkan ke node berikutnya, lalu data dokter dalam aktivitas (`temp->dokterData`) dibebaskan terlebih dahulu sebelum node aktivitas itu sendiri dilepaskan dengan `free(temp)`.

3.1.3. Penentuan Jadwal Otomatis

a. Flowchart Penjadwalan.c



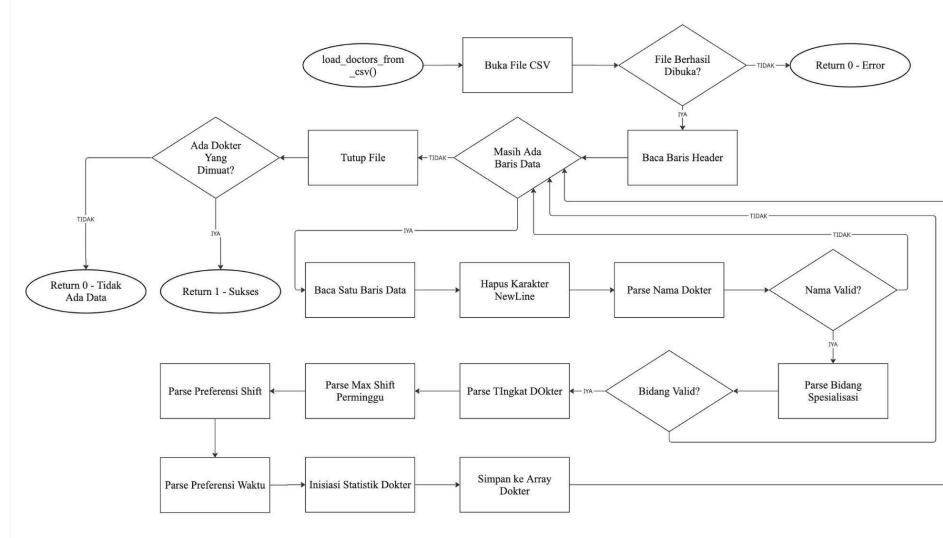
Gambar 22. Flowchart penjadwalan.c

Flowchart ini menguraikan proses spesifik pembuatan jadwal dokter di sebuah sistem kesehatan, dimulai dengan inisiasi sistem, yang meliputi reset counter dokter dan array dokter untuk memastikan data bersih sebelum memproses. Selanjutnya, sistem memuat data dokter dari file CSV bernama "daftar_dokter.csv". Jika proses pemuatan berhasil, sistem menggunakan algoritma greedy untuk menghasilkan jadwal bulanan dokter, misalnya untuk bulan Juli 2025, dan hasilnya ditampilkan sebagai "Jadwal Berhasil Dibuat" dengan detail jadwal spesifik per dokter.

Jika jadwal berhasil dibuat, sistem mencoba menyimpan hasil ke file "jadwal_dokter.csv". Jika penyimpanan sukses, outputnya adalah "Jadwal Tersimpan di File" dengan konfirmasi lokasi penyimpanan, dan proses berakhir. Namun, jika pemuatan data dari "dokter_data.csv" gagal, sistem langsung mengeluarkan pesan "Gagal Membaca File Dokter" dan berhenti. Jika pembuatan jadwal gagal atau penyimpanan ke file gagal, masing-masing menghasilkan output "Jadwal Berhasil Dibuat" (dengan catatan kegagalan) atau "Gagal

"Menyimpan Jadwal", memastikan pengguna tahu titik masalahnya pada tanggal dan waktu tersebut.

b. Flowchart fungsi `load_doctors_from_csv()`



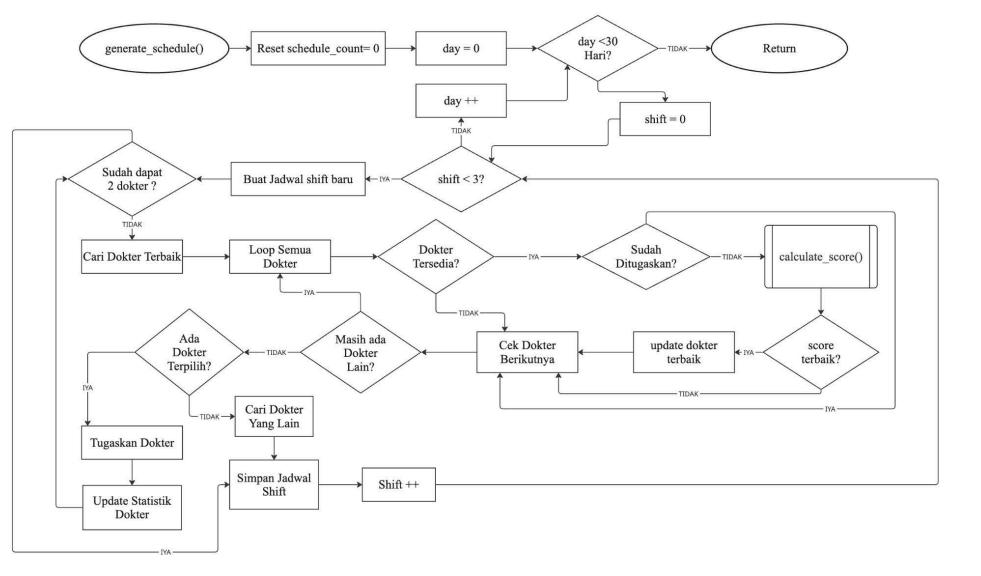
Gambar 23. Flowchart fungsi `load_doctors_from_csv()`

Flowchart ini menggambarkan proses pengolahan data dokter dari file CSV, dimulai dengan memuat data dari file "daftar_dokter.csv". Sistem membuka file CSV, dan jika gagal dibuka, proses langsung mengembalikan kode error 0. Jika berhasil dibuka, file ditutup, dan sistem memeriksa apakah ada dokter yang dimuat. Jika tidak ada data dokter, hasilnya adalah "Return 0 - Tidak Ada Data"; jika ada, proses melanjutkan dengan membaca baris data satu per satu. Setiap baris data diproses dengan menghapus karakter newline, lalu mem-parsing nama dokter untuk memastikan validitasnya.

Jika nama dokter valid, proses melanjutkan dengan mem-parsing preferensi shift, batas maksimum shift per minggu, tingkat dokter, preferensi waktu, dan bidang spesialisasi. Setiap langkah validasi, seperti "Nama Valid?" dan "Bidang Valid?", memastikan data konsisten sebelum disimpan. Data yang lolos validasi diinisiasi untuk statistik dokter dan disimpan ke array dokter, dengan hasil akhir "Return 1 -

"Sukses" jika semua berjalan lancar. Jika ada kegagalan pada tahap validasi bidang atau data lainnya, proses berhenti dan mengembalikan error sesuai kondisi spesifik.

c. Flowchart Fungsi generate_schedule()

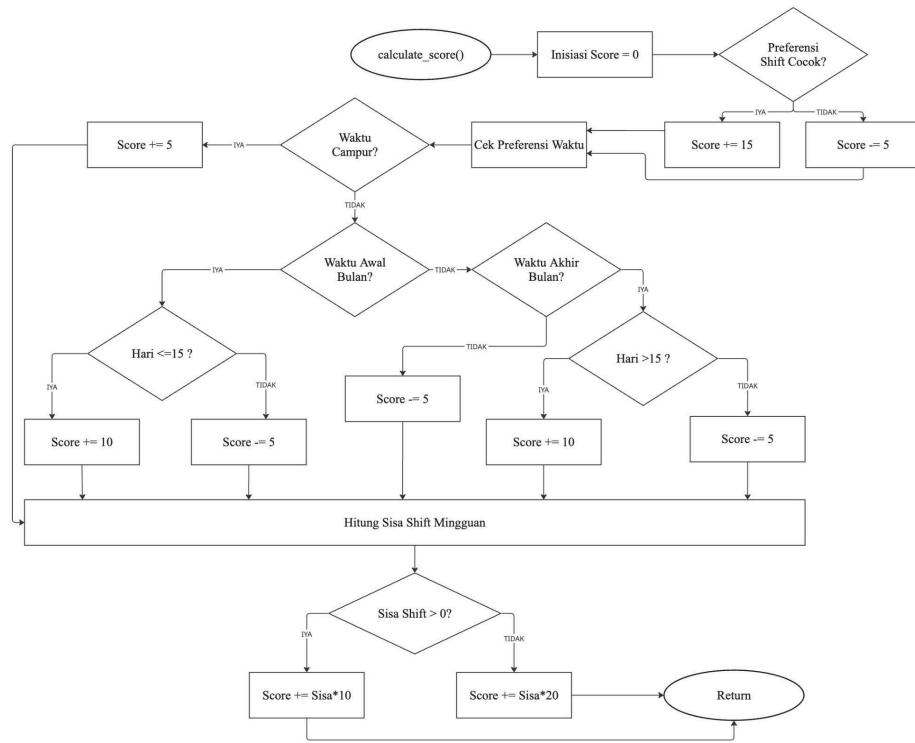


Gambar 24. Flowchart fungsi generate_schedule()

Flowchart ini menggambarkan proses pembuatan jadwal dokter melalui fungsi generate_schedule(), yang dimulai dengan mengatur schedule count ke 0 dan inisialisasi hari ke 0. Proses berjalan dalam loop harian selama 30 hari, dan untuk setiap hari, sistem membuat jadwal shift baru jika belum ada dua dokter yang ditugaskan. Jika shift kurang dari 3, sistem memeriksa semua dokter untuk menemukan yang tersedia, dan jika ada dokter yang bisa ditugaskan, mereka dipilih berdasarkan skor terbaik yang dihitung melalui calculate_score(). Jika skor terbaik ditemukan, dokter tersebut diupdate dan ditugaskan, lalu statistik dokter diperbarui.

Jika tidak ada dokter tersedia atau jadwal tidak memenuhi kriteria, sistem mencari dokter lain atau melanjutkan ke dokter berikutnya untuk dicek. Setelah tugas selesai, jadwal shift disimpan, dan jumlah shift ditambah satu. Jika proses mencapai hari ke-30 atau gagal menemukan kombinasi dokter yang sesuai, fungsi mengembalikan hasil, dengan "Return" menandakan penyelesaian atau kegagalan berdasarkan kondisi loop dan validasi.

d. Flowchart Fungsi calculate_score()



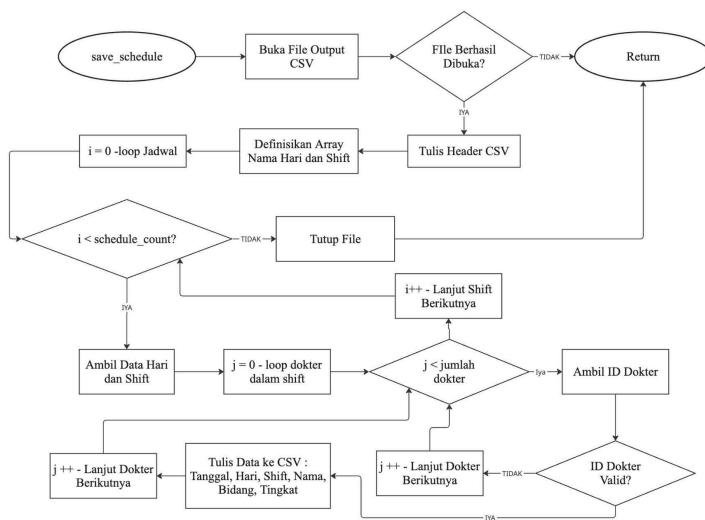
Gambar 25. Flowchart fungsi calculate_score()

Flowchart ini menggambarkan proses perhitungan skor dalam fungsi calculate_score() untuk mengevaluasi preferensi dokter, dimulai dengan inisiasi skor awal ke 0. Skor awal ditambah 5 jika ada preferensi waktu yang sesuai, lalu

diperiksa apakah waktu tersebut adalah awal atau akhir bulan. Jika waktu awal bulan, skor dikurangi 5, sedangkan jika akhir bulan, skor ditambah 10. Selanjutnya, sistem memeriksa hari relatif terhadap tanggal 15; jika hari sama dengan 15, skor ditambah 10, dan jika lebih dari 15, skor ditambah 5, tetapi jika kurang dari 15, skor dikurangi 5.

Setelah itu, proses melanjutkan ke tahap menghitung sisa shift mingguan, di mana skor ditambah 10 jika sisa shift lebih dari 0, atau ditambah 20 jika tidak ada sisa shift. Terakhir, skor diperiksa lagi untuk kecocokan shift; jika cocok, skor ditambah 15, tetapi jika tidak, skor dikurangi 5. Proses berakhir dengan mengembalikan nilai skor akhir yang mencerminkan semua kondisi yang dievaluasi.

e. Flowchart Fungsi save_schedule()



Gambar 26. Flowchart fungsi save_schedule()

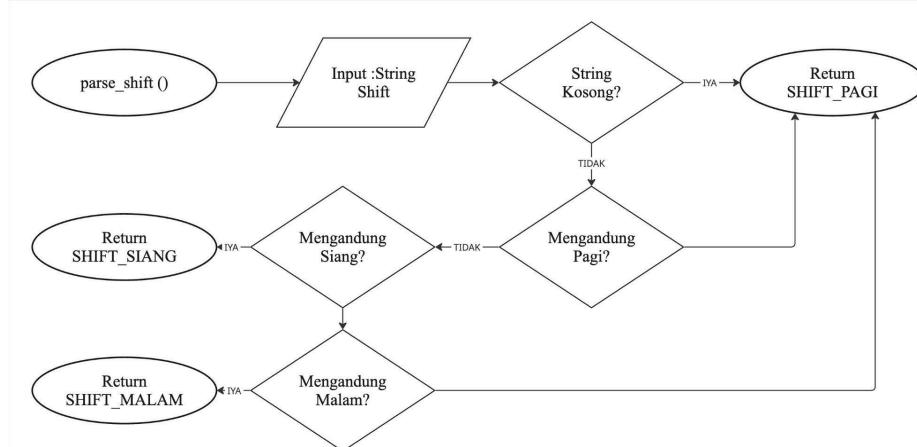
Flowchart ini menggambarkan proses penyimpanan jadwal (save_schedule) ke file CSV, dimulai dengan membuka file output CSV. Jika file gagal dibuka, proses langsung mengembalikan hasil tanpa melanjutkan. Jika berhasil dibuka, sistem

mendefinisikan array berisi nama hari dan shift, lalu menulis header CSV. Proses berlanjut dalam loop berdasarkan jumlah jadwal (schedule_count), di mana untuk setiap iterasi, data hari dan shift diambil dan ditulis ke CSV bersama informasi seperti tanggal, nama, bidang, dan tingkat dokter.

Dalam loop dokter untuk setiap shift, sistem memeriksa jumlah dokter yang tersedia. Jika ada dokter yang valid (ID dokter valid), ID dokter tersebut ditambahkan, dan data ditulis ke CSV. Jika tidak ada dokter atau ID tidak valid, proses melanjutkan ke dokter berikutnya. Setelah semua data shift tertulis, file ditutup, dan iterasi shift berikutnya dilakukan hingga semua jadwal selesai, lalu proses berakhir dengan pengembalian hasil.

f. Flowchart Fungsi-Fungsi Pembantu

- `parse_shift()`



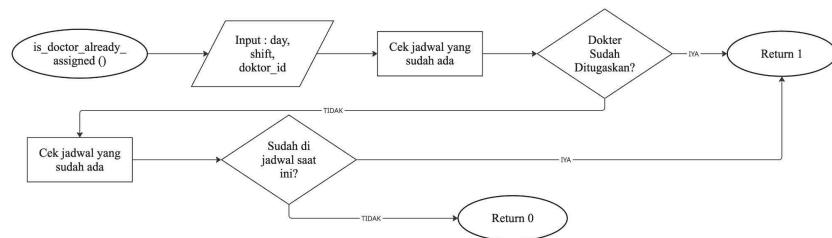
Gambar 27. Flowchart fungsi `parse_shift()`

Flowchart ini menggambarkan proses `parse_shift()` yang mengolah input string untuk menentukan jenis shift, dimulai dengan menerima input string shift. Sistem pertama-tama memeriksa apakah string tersebut kosong; jika

ya, fungsi mengembalikan "SHIFT_PAGI" sebagai default. Jika tidak kosong, proses melanjutkan dengan memeriksa apakah string menandakan "Siang"; jika iya, fungsi mengembalikan "SHIFT_SIANG". Jika tidak, sistem memeriksa apakah string menandakan "Pagi"; jika iya, fungsi mengembalikan "SHIFT_PAGI".

Jika tidak sesuai dengan "Pagi", proses melanjutkan dengan memeriksa apakah string menandakan "Malam"; jika iya, fungsi mengembalikan "SHIFT_MALAM". Jika semua kondisi sebelumnya tidak terpenuhi, proses berhenti tanpa pengembalian nilai tambahan, menandakan input tidak valid atau tidak sesuai dengan kategori shift yang dikenali.

- `is_doctor_already_assigned ()`

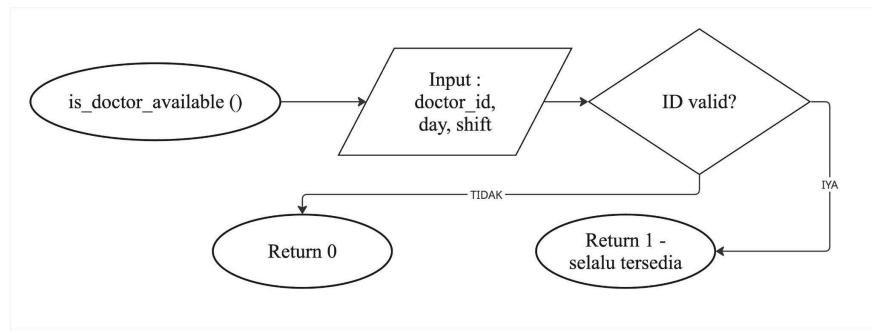


Gambar 28. Flowchart fungsi `is_doctor_already_assigned()`

Flowchart ini menggambarkan proses `is_doctor_already_assigned()` yang memeriksa apakah dokter sudah ditugaskan pada shift tertentu, dimulai dengan menerima input berupa hari, shift, dan ID dokter. Sistem pertama-tama memeriksa jadwal yang sudah ada untuk dokter tersebut. Jika dokter sudah ditugaskan pada jadwal tersebut, fungsi mengembalikan nilai 1. Jika tidak, proses melanjutkan dengan memeriksa apakah jadwal saat ini sudah diisi.

Jika jadwal saat ini belum diisi, fungsi mengembalikan nilai 0. Jika jadwal sudah diisi, sistem kembali memeriksa jadwal yang sudah ada untuk memastikan konsistensi, dan jika dokter tidak ditemukan dalam jadwal yang ada, fungsi tetap mengembalikan 0, menandakan dokter tersebut belum ditugaskan.

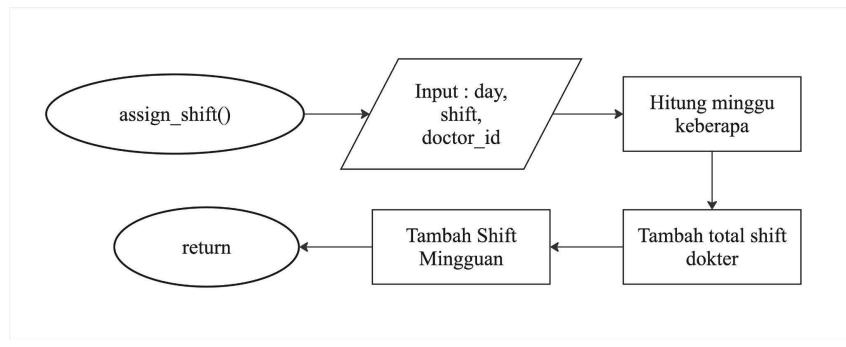
- `is_doctor_available ()`



Gambar 29. Flowchart fungsi `is_doctor_available()`

Flowchart ini menggambarkan proses `is_doctor_available()` yang memeriksa ketersediaan dokter berdasarkan input ID dokter, hari, dan shift. Proses dimulai dengan menerima input tersebut, lalu memvalidasi apakah ID dokter valid. Jika ID tidak valid, fungsi langsung mengembalikan nilai 0. Jika ID valid, fungsi mengembalikan nilai 1, menandakan dokter selalu tersedia selama ID-nya sah.

- assign_shift()

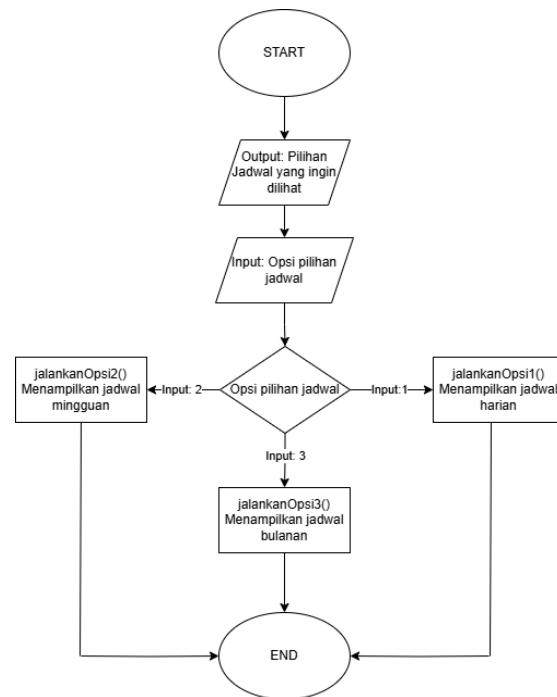


Gambar 30. Flowchart fungsi assign_shift()

Flowchart ini menggambarkan proses `assign_shift()` yang digunakan untuk menugaskan dokter ke shift tertentu, dimulai dengan menerima input berupa hari, shift, dan ID dokter. Sistem pertama-tama menghitung minggu keberapa berdasarkan hari tersebut, lalu menambahkan shift ke jadwal mingguan yang sesuai. Selanjutnya, proses menambahkan total shift untuk dokter tersebut, dan fungsi berakhir dengan mengembalikan nilai untuk menyelesaikan tugas.

3.1.4. Penampilan Informasi dan Jadwal

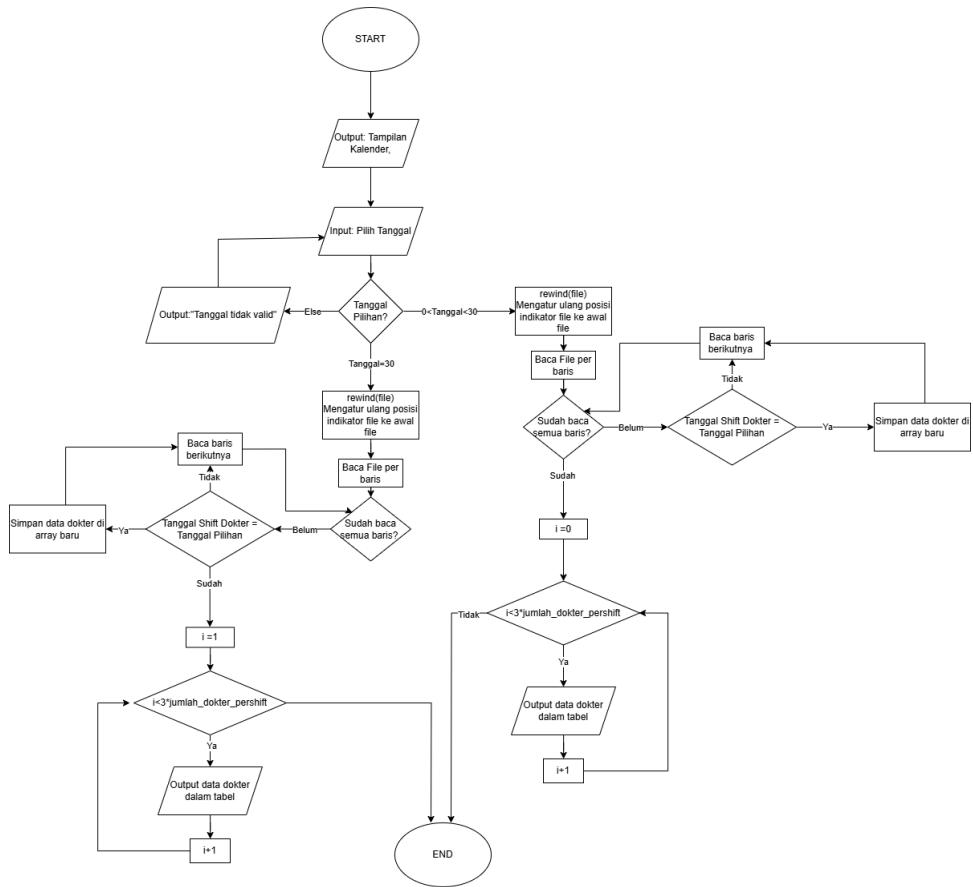
a. Flowchart Laporan1.c



Gambar 31. Flowchart Laporan1.c

Flowchart ini memvisualisasikan proses pada laporan1.c yang memiliki fungsi untuk menampilkan jadwal shift dokter yang terbagi dalam 3 pilihan yaitu menampilkan jadwal harian, menampilkan jadwal mingguan dan menampilkan jadwal bulanan. Sistem pertama-tama akan menampilkan output pilihan jadwal yang ingin dilihat (Harian/Mingguan/Bulanan), kemudian sistem akan mengarahkan setiap pilihan ke fungsi nya masing masing, seperti pilihan 1(harian) akan diarahkan ke fungsi `jalankan_Opsi1()` yang merupakan fungsi untuk menampilkan jadwal harian begitu juga untuk pilihan 2 dan 3.

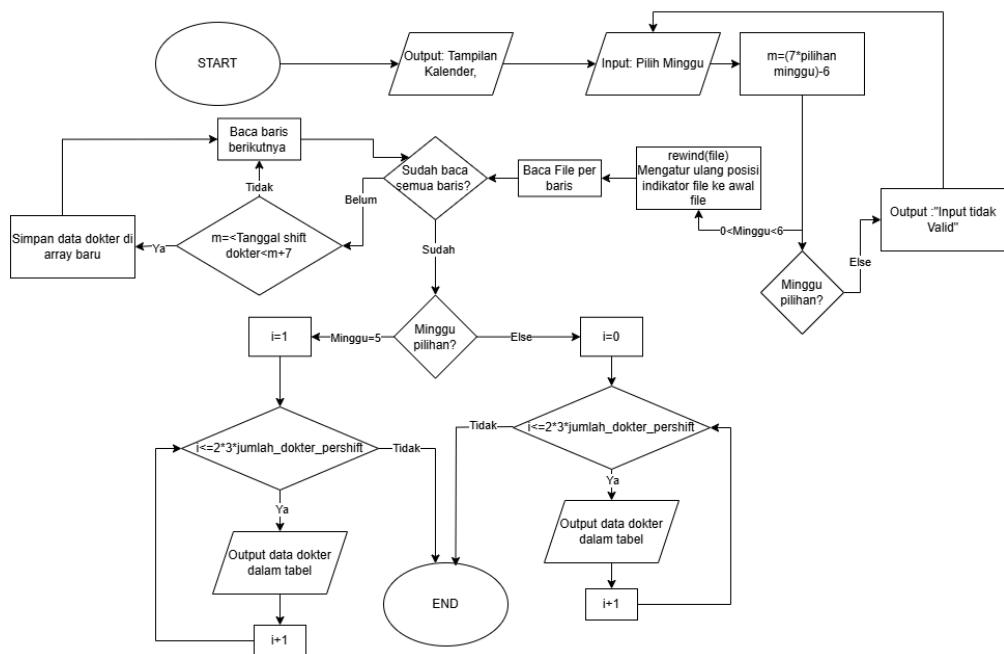
b. Flowchart jalankan_Opsi1()



Gambar 32. Flowchart jalankan_Opsi1()

Flowchart ini memvisualisasikan proses yang terjadi pada fungsi jalankan_Opsi1(). Fungsi ini berfungsi untuk menampilkan jadwal harian dari shift dokter dari setiap tanggal pada satu bulan. Pertama sistem akan mengatur ulang posisi indikator file ke awal file kemudian menampilkan tampilan output kalender untuk memudahkan user memilih tanggal, kemudian sistem menerima input tanggal, dan akan membaca file kemudian menyimpan data dokter yang dibaca dari file per baris ke array baru jika tanggal shift dokter sama dengan tanggal yang dipilih user. Setelah itu sistem akan mengeluarkan output data dokter dari array baru dalam bentuk tabel.

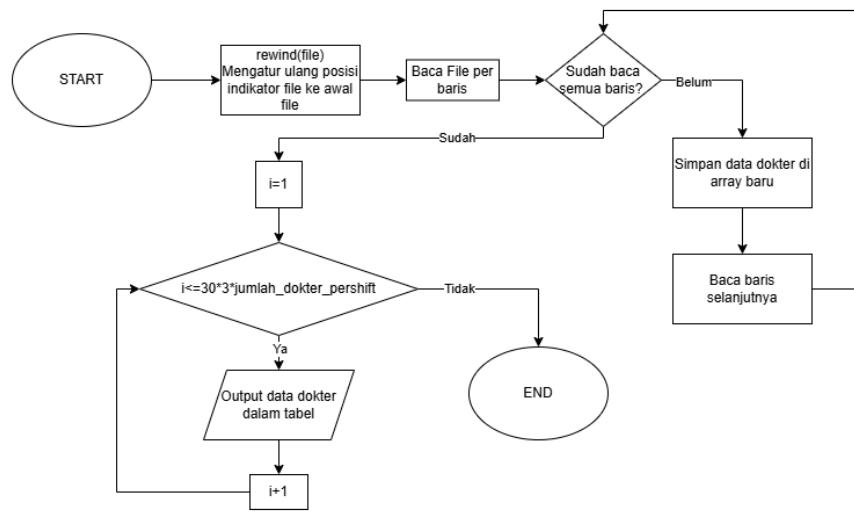
c. Flowchart jalankan_Opsi2()



Gambar 33. Flowchart jalankan_Opsi2()

Flowchart ini memvisualisasikan proses yang terjadi pada fungsi jalankan_Opsi2(). Fungsi ini berfungsi untuk menampilkan jadwal mingguan dari shift dokter dari setiap minggu pada satu bulan. Pertama sistem akan mengatur ulang posisi indikator file ke awal file kemudian menampilkan tampilan output kalender untuk memudahkan user memilih minggu, kemudian sistem menerima input minggu dengan rentang minggu 1 - minggu 5, sistem ini menggunakan 5 minggu dikarenakan asumsi jumlah hari dalam 1 bulan selalu 30 hari sehingga jika hanya menggunakan 4 minggu maka akan ada 2 hari yang tidak dapat dilihat jadwalnya(tanggal 29 dan 30). Kemudian sistem akan membaca file dan menyimpan data dokter yang dibaca dari file per baris ke array baru jika tanggal shift dokter lebih besar sama dengan variabel integer m (7x pilihan minggu user -6) dan lebih kecil dari $m+7$. Setelah itu sistem akan mengeluarkan output data dokter dari array baru dalam bentuk tabel.

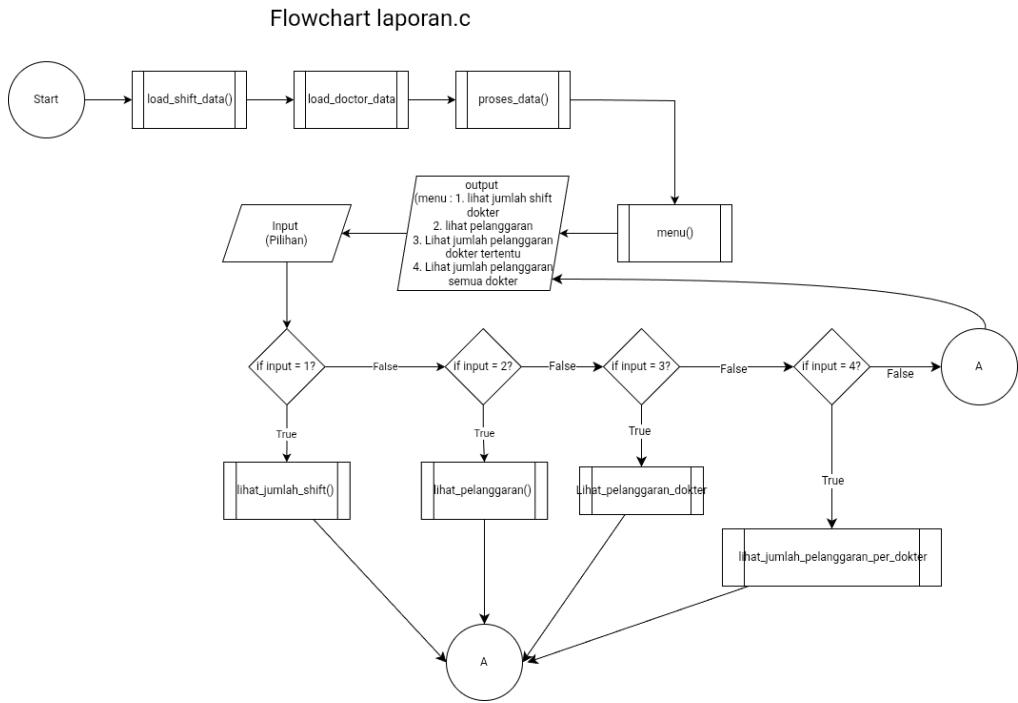
d. Flowchart jalankanOpsi3



Gambar 34. Flowchart jalankan_Opsi3

Flowchart ini memvisualisasikan proses yang terjadi pada fungsi jalankan_Opsi3(). Fungsi ini berfungsi untuk menampilkan jadwal bulanan shift dokter. Pertama sistem akan mengatur ulang posisi indikator file ke awal file kemudian sistem akan membaca file dan menyimpan data dokter yang dibaca dari file perbaris ke array baru. Setelah itu sistem akan mengeluarkan output data dokter dari array baru dalam bentuk tabel.

e. Flowchart laporan.c



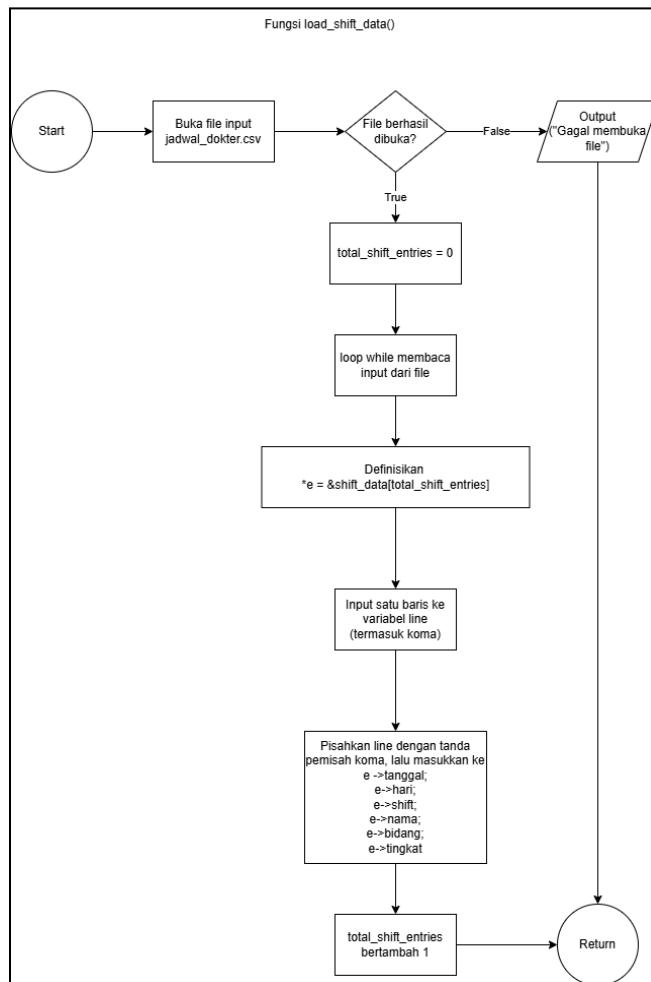
Gambar 35. Laporan..c

Flowchart ini menjelaskan urutan pengerjaan di file laporan.c, hal yang pertama dilakukan adalah menjalankan fungsi `load_shift_data`, dimana fungsi tersebut mengambil data dari file `jadwal_dokter.csv` lalu dimasukkan ke dalam sebuah array baru.

Lalu program akan menjalankan fungsi `load_doctor_data`, dimana program akan mengambil data dari file `daftar_dokter.csv` lalu dimasukkan ke dalam sebuah variabel array, setelah itu program akan menjalankan fungsi `proses_data`, dimana data dari dua input tadi akan diproses.

Lalu akan muncul sebuah menu dan user akan diminta menginput pilihannya, terdapat dua pilihan yaitu melihat jumlah shift dokter dan melihat pelanggaran yang terjadi.

F. flowchart load_shift_data



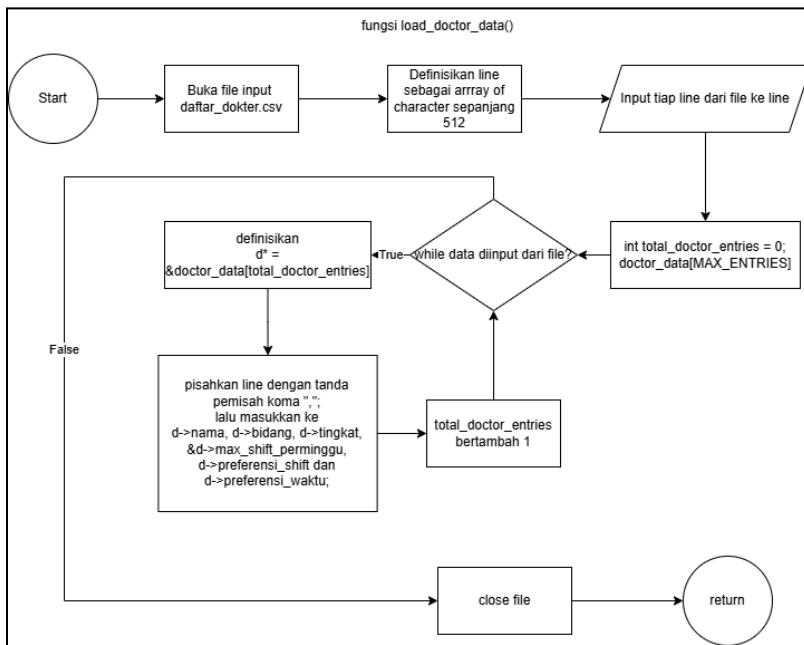
Gambar 36. Flowchart fungsi load_shift_data

Fungsi `load_shift_data()` dalam file `laporan.c` berfungsi untuk membaca data jadwal dokter dari file CSV eksternal dan menyimpannya ke dalam array `shift_data`. Proses diawali dengan membuka file `data/jadwal_dokter.csv` menggunakan fungsi `fopen` dalam mode baca ("r"). Jika file gagal dibuka, program langsung menghentikan eksekusi dengan menampilkan pesan kesalahan, karena data shift merupakan komponen krusial dalam analisis jadwal dan pelanggaran. Setelah file berhasil dibuka, baris pertama yang biasanya berisi header kolom dibaca dan diabaikan karena tidak mengandung data dokter yang relevan.

Selanjutnya, fungsi memasuki sebuah perulangan while yang membaca setiap baris dari file CSV. Setiap baris yang dibaca merepresentasikan satu entri shift dokter, yang kemudian diuraikan menggunakan sscanf() ke dalam enam elemen, yaitu tanggal, hari, shift, nama dokter, bidang spesialisasi, dan tingkat (seperti spesialis atau residen). Data ini disimpan ke dalam elemen berikutnya dari array shift_data, menggunakan indeks total_shift_entries sebagai penanda posisi. Setelah data satu entri selesai diproses dan disimpan, variabel penghitung total_shift_entries dinaikkan satu untuk mencatat bahwa satu entri baru telah ditambahkan.

Setelah semua baris dalam file dibaca dan diproses, fungsi menutup file menggunakan fclose. Dengan demikian, fungsi load_shift_data() memastikan bahwa seluruh informasi shift dokter tersedia dalam memori program untuk dianalisis lebih lanjut, misalnya saat menghitung pelanggaran jadwal atau membuat laporan. Fungsi ini menjadi tahapan awal dan sangat penting dalam alur kerja sistem pelaporan karena dari sinilah data mentah jadwal diimpor dan disiapkan untuk diproses lebih lanjut oleh fungsi lain seperti proses_data() dan lihat_pelanggaran().

G. flowchart load_doctor_data()



Gambar 37. Flowchart fungsi load_doctor_data()

Fungsi `load_doctor_data()` bertugas untuk membaca data preferensi masing-masing dokter dari file eksternal data/`daftar_dokter.csv`, kemudian menyimpannya ke dalam array `doctor_data`. Fungsi ini sangat penting karena data preferensi tersebut menjadi dasar untuk mengevaluasi apakah jadwal shift yang diberikan kepada dokter sudah sesuai dengan preferensi shift, waktu kerja, dan batas maksimal jumlah shift mingguan yang diinginkan atau diperbolehkan.

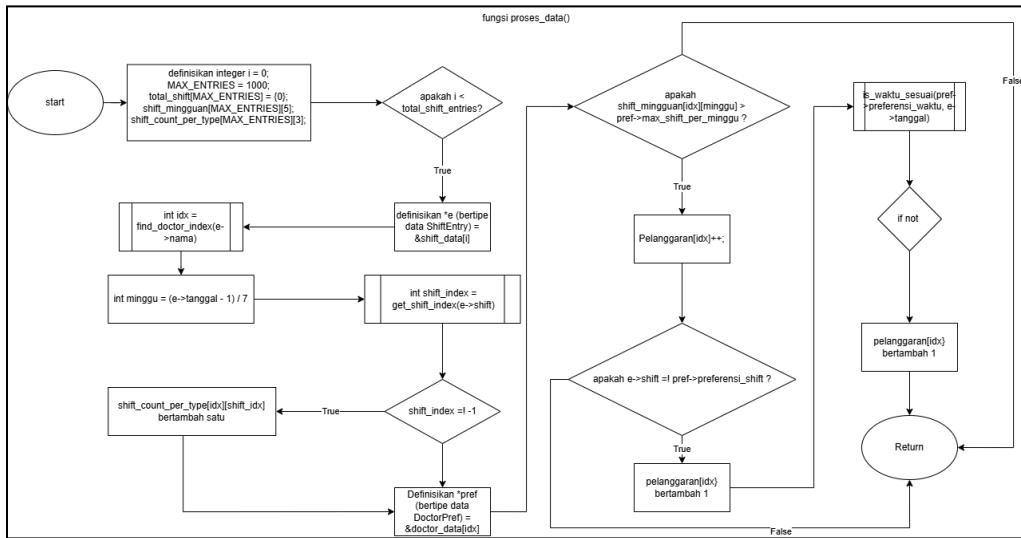
Proses dimulai dengan membuka file menggunakan `fopen` dalam mode baca. Sama seperti pada fungsi pembacaan data shift, apabila file tidak berhasil dibuka (misalnya karena file tidak ditemukan atau tidak memiliki izin baca), maka program akan mencetak pesan kesalahan dan menghentikan eksekusi dengan `exit(1)`. Ini menunjukkan bahwa data preferensi dokter bersifat esensial dan program tidak dapat berjalan tanpa memuatnya terlebih dahulu. Setelah berhasil membuka file, fungsi

membaca baris pertama sebagai header dan mengabaikannya karena hanya berisi penjelasan nama-nama kolom.

Fungsi kemudian melakukan perulangan while untuk membaca setiap baris data dokter satu per satu. Setiap baris CSV akan dipecah menggunakan sscanf() menjadi enam elemen data, yaitu nama dokter, bidang spesialisasi, tingkat (misalnya spesialis atau residen), jumlah maksimum shift per minggu yang diperbolehkan (max_shift_per_minggu), preferensi shift (pagi, siang, atau malam), dan preferensi waktu (awal bulan, akhir bulan, atau campur). Data yang telah diparsing ini akan disimpan dalam elemen berikutnya pada array doctor_data, berdasarkan indeks total_doctor_entries. Setelah penyimpanan data selesai untuk satu dokter, indeks tersebut dinaikkan satu untuk mempersiapkan tempat bagi entri berikutnya.

Setelah semua baris dalam file daftar_dokter.csv selesai dibaca dan data dokter tersimpan dengan rapi dalam array, file ditutup menggunakan fclose. Dengan demikian, fungsi load_doctor_data() mempersiapkan seluruh informasi preferensi dokter ke dalam memori program. Informasi ini kemudian akan digunakan oleh fungsi lain seperti proses_data() dan lihat_pelanggaran() untuk membandingkan antara jadwal yang diberikan dengan preferensi dokter, mendeteksi adanya pelanggaran, dan menyusun laporan akhir. Fungsi ini menjadi komponen krusial dalam keseluruhan sistem evaluasi penjadwalan dokter.

H. fungsi proses_data()



Gambar 38. Flowchart fungsi proses_data()

Fungsi proses_data() dalam program ini berfungsi sebagai inti dari proses evaluasi terhadap kesesuaian jadwal shift dokter berdasarkan data preferensi yang telah dimuat sebelumnya. Fungsi ini membaca data dari array shift_data (berisi jadwal shift dokter) dan doctor_data (berisi preferensi masing-masing dokter), kemudian menghitung total shift per dokter, jumlah shift per minggu, jenis shift yang dijalani, dan mendeteksi pelanggaran-pelanggaran terhadap preferensi atau batasan yang ditetapkan.

Fungsi ini bekerja dengan menggunakan perulangan for yang menelusuri seluruh entri shift dari indeks 0 hingga total_shift_entries. Untuk setiap entri, ia mengambil pointer ShiftEntry *e yang menunjuk ke data shift pada indeks ke-i. Selanjutnya, nama dokter pada entri shift tersebut dicocokkan terhadap daftar preferensi dokter menggunakan fungsi find_doctor_index(). Jika nama dokter tidak ditemukan dalam daftar preferensi (hasil pencarian bernilai -1), maka entri tersebut diabaikan dengan continue, dan iterasi berikutnya langsung dijalankan. Hal ini mencegah program memproses data yang tidak lengkap atau tidak valid.

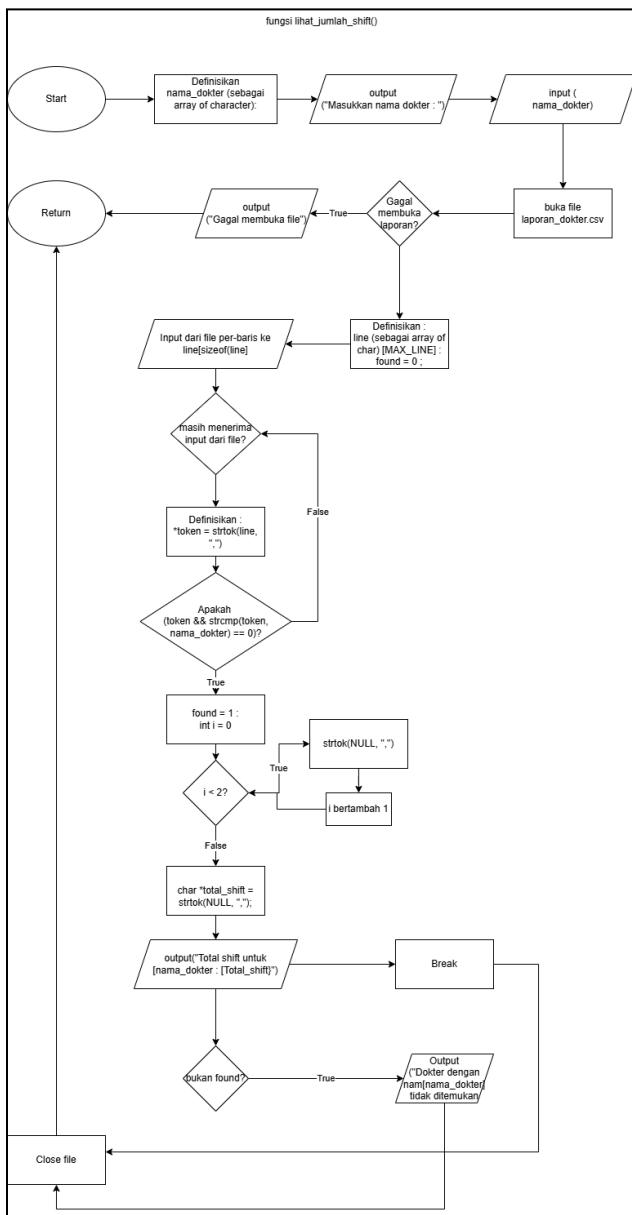
Jika nama dokter ditemukan, maka berbagai penghitungan dimulai. Pertama, minggu keberapa shift itu terjadi dihitung berdasarkan tanggal: $(e->tanggal - 1) / 7$. Ini akan membagi bulan menjadi lima minggu maksimum. Kemudian, jenis shift (pagi, siang, malam) diterjemahkan ke dalam indeks numerik (0, 1, atau 2) dengan menggunakan fungsi `get_shift_index()`.

Data statistik kemudian mulai dikumpulkan: total shift dokter ditambahkan satu, jumlah shift pada minggu tersebut juga ditambahkan satu, dan jika jenis shift valid (bukan nilai -1), maka penghitung shift untuk jenis tersebut juga diperbarui. Dengan cara ini, sistem menyimpan semua informasi penting yang nantinya digunakan dalam laporan akhir.

Setelah proses akumulasi selesai, fungsi kemudian mulai melakukan evaluasi terhadap potensi pelanggaran. Pelanggaran pertama yang dicek adalah jika jumlah shift mingguan dokter melebihi batas maksimum (`pref->max_shift_per_minggu`). Jika ya, penghitung pelanggaran[idx] dinaikkan satu. Lalu dicek apakah shift yang diberikan sama dengan preferensi shift dokter, dan jika tidak sama, pelanggaran kembali dicatat. Terakhir, dicek apakah tanggal shift tersebut sesuai dengan preferensi waktu kerja dokter menggunakan fungsi `is_waktu_sesuai()`. Jika tidak sesuai, maka kembali ditandai sebagai pelanggaran.

Dengan cara ini, fungsi `proses_data()` tidak hanya mengumpulkan statistik penting tentang beban kerja tiap dokter, tapi juga mendekripsi dan mencatat semua penyimpangan dari preferensi atau ketentuan yang berlaku. Hasil dari fungsi ini kemudian menjadi dasar untuk pembuatan laporan yang informatif melalui fungsi `tulis_laporan()` dan untuk ditampilkan dalam bentuk notifikasi atau analisis dalam fungsi lain seperti `lihat_pelanggaran()`.

I. fungsi lihat_jumlah_shift()



Gambar 39. Flowchart fungsi jumlah_shift()

Fungsi lihat_jumlah_shift() bertanggung jawab untuk menampilkan jumlah total shift yang telah dijalani oleh seorang dokter tertentu berdasarkan data yang tersimpan dalam file laporan (laporan_dokter.csv). Fungsi ini bersifat interaktif karena meminta input dari pengguna secara langsung melalui terminal atau konsol.

Proses dimulai dengan mendeklarasikan sebuah variabel `nama_dokter` bertipe array karakter untuk menampung nama dokter yang akan dicari. Program kemudian meminta pengguna untuk memasukkan nama dokter yang ingin dicek jumlah shiftnya dengan fungsi `fgets`. Setelah nama dimasukkan, newline (`\n`) di akhir input dihapus menggunakan `strcspn`, agar nama tersebut bisa dibandingkan dengan benar terhadap data yang ada dalam file CSV.

Setelah input diterima, fungsi mencoba membuka file laporan (`laporan_dokter.csv`) dalam mode baca. Jika file gagal dibuka—misalnya karena belum dibuat oleh fungsi `tulis_laporan()` atau karena file rusak—maka program akan mencetak pesan kesalahan dan keluar dari fungsi, tanpa melanjutkan proses.

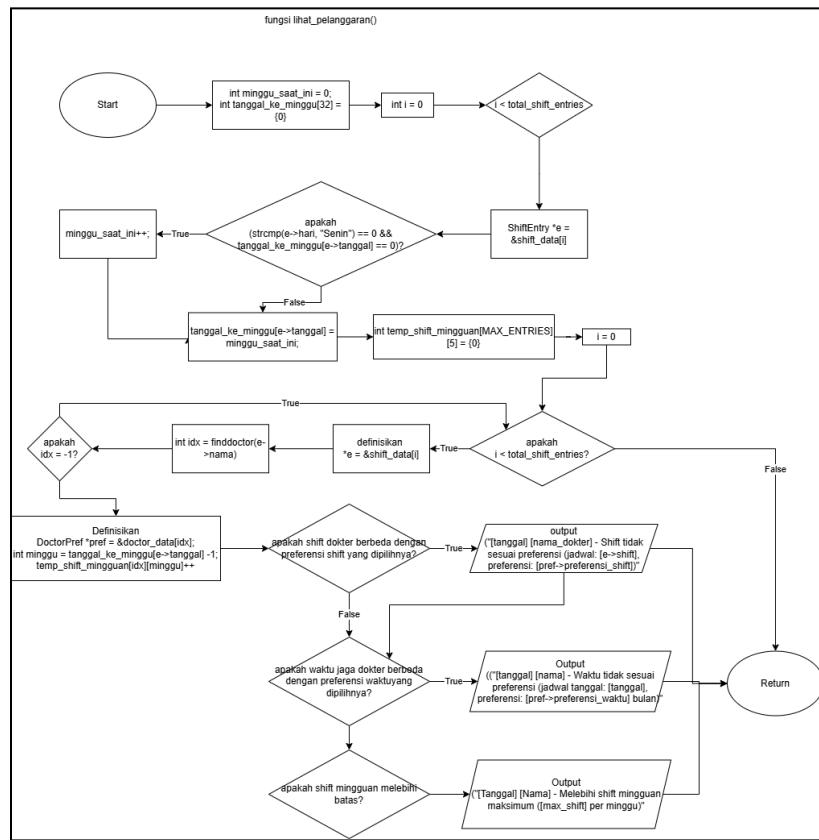
Jika file berhasil dibuka, maka baris pertama (header kolom) dilewati dengan `fgets`, karena hanya berisi label kolom dan tidak perlu diproses. Setelah itu, program memasuki perulangan `while` yang membaca baris-baris selanjutnya satu per satu menggunakan `fgets`. Setiap baris kemudian diparsing menggunakan `strtok`, dimulai dari token pertama yang merupakan nama dokter. Nama ini dibandingkan dengan nama yang dimasukkan oleh pengguna.

Jika ditemukan kecocokan, maka program menetapkan bahwa dokter telah ditemukan (`found = 1`) dan mengambil token keempat dari baris tersebut, yaitu `total_shift`, setelah melewati dua token tambahan (bidang dan tingkat). Nilai `total shift` tersebut kemudian ditampilkan ke layar, dan perulangan dihentikan dengan `break`.

Jika perulangan selesai tanpa menemukan nama yang dicari, maka program akan memberi tahu pengguna bahwa dokter dengan nama tersebut tidak ditemukan dalam laporan. Terakhir, file laporan ditutup menggunakan `fclose` untuk membebaskan sumber daya sistem.

Secara keseluruhan, fungsi lihat_jumlah_shift() memberikan cara cepat bagi pengguna untuk memverifikasi total beban kerja seorang dokter berdasarkan laporan akhir, tanpa harus membuka file CSV secara manual. Fungsi ini juga menunjukkan bagaimana file CSV bisa digunakan sebagai media pertukaran data antar bagian program yang bersifat modular.

J. fungsi lihat_pelanggaran()



Gambar 40. Flowchart fungsi lihat_pelanggaran()

Fungsi lihat_pelanggaran() dalam program ini bertujuan untuk menampilkan secara langsung pelanggaran-pelanggaran jadwal shift yang dilakukan oleh dokter berdasarkan data preferensi yang dimuat sebelumnya. Tidak seperti fungsi proses_data() yang menghitung pelanggaran secara diam-diam untuk dicatat ke dalam laporan, fungsi ini menampilkan pelanggaran secara eksplisit di layar, lengkap dengan

tanggal dan alasan pelanggarannya. Fungsi ini sangat berguna sebagai alat bantu pemeriksaan manual dan analisis langsung oleh pengguna.

Fungsi dimulai dengan membuat array tanggal_ke_minggu[32], yang digunakan untuk memetakan tanggal ke nomor minggu dalam bulan. Inisialisasi minggu saat ini dimulai dari nol dengan variabel minggu_saat_ini. Perulangan pertama membaca setiap entri shift dari shift_data, dan setiap kali ditemukan entri dengan hari "Senin" yang belum dikaitkan ke minggu manapun (nilai mapping-nya masih nol), maka minggu_saat_ini dinaikkan satu. Ini digunakan sebagai penanda dimulainya minggu baru, sehingga setiap tanggal bisa dipetakan ke minggu tertentu dalam bulan. Nilai mapping ini disimpan dalam array tanggal_ke_minggu dengan indeks berdasarkan tanggal shift.

Setelah mapping minggu selesai, array temp_shift_minggu[MAX_ENTRIES][5] diinisialisasi untuk menghitung jumlah shift minggu sementara setiap dokter. Fungsi kemudian memasuki perulangan kedua untuk memproses entri shift satu per satu. Setiap entri diambil sebagai pointer ShiftEntry *e. Nama dokter dari entri tersebut dicocokkan ke dalam data preferensi menggunakan find_doctor_index(). Jika nama tidak ditemukan, maka entri ini dilewati dengan continue.

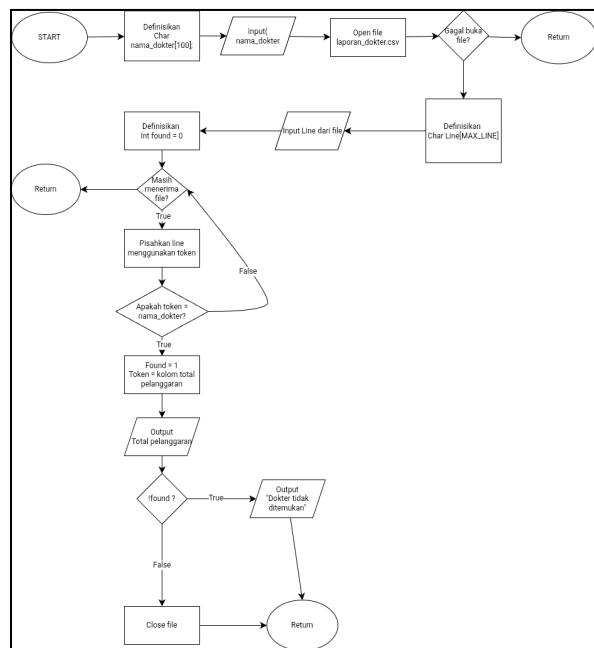
Jika nama dokter valid, maka program mencari minggu berapa shift itu terjadi menggunakan hasil dari tanggal_ke_minggu. Nilai temp_shift_minggu[idx][minggu] untuk dokter tersebut ditambah satu, untuk mencatat bahwa dokter tersebut memiliki satu shift tambahan pada minggu itu.

Setelah itu, fungsi mulai memeriksa pelanggaran berdasarkan tiga kriteria. Pertama, jika shift yang dijadwalkan (e->shift) tidak sesuai dengan preferensi shift dokter (pref->preferensi_shift), maka ditampilkan pesan

bawa shift tidak sesuai. Kedua, jika tanggal shift tidak sesuai dengan preferensi waktu dokter—misalnya dokter hanya bersedia bekerja di awal bulan, tetapi jadwal jatuh pada akhir bulan—maka dicetak peringatan bahwa waktu tidak sesuai preferensi. Ketiga, jika jumlah shift mingguan melebihi batas maksimum ($\text{pref} \rightarrow \text{max_shift_per_minggu}$), maka dicetak pesan bahwa dokter melebihi jumlah shift mingguan.

Setiap pelanggaran ini dicetak ke layar dengan menyebutkan tanggal, nama dokter, jenis pelanggaran, dan nilai-nilai yang tidak sesuai. Dengan demikian, fungsi `lihat_pelanggaran()` memungkinkan pengguna untuk secara langsung melihat penyimpangan dari jadwal ideal tanpa perlu membuka laporan yang sudah disimpan ke file, dan cocok digunakan untuk debugging, verifikasi cepat, atau inspeksi manual data sebelum laporan difinalisasi.

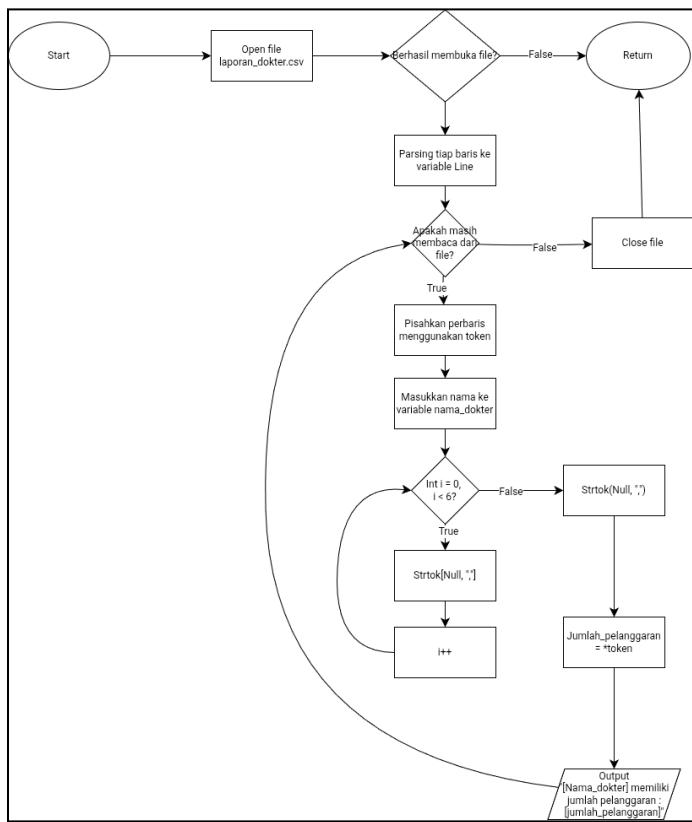
K. Flowchart fungsi lihat jumlah pelanggaran dokter tertentu



Gambar 41. Flowchart fungsi `lihat_pelanggaran_dokter()`

Fungsi `lihat_pelanggaran_dokter()` digunakan untuk menampilkan jumlah pelanggaran sistem yang menimpa seorang dokter, berdasarkan data dalam file CSV bernama `laporan_dokter.csv`. Program pertama-tama meminta pengguna memasukkan nama dokter, kemudian menghapus karakter newline (`\n`) dari input agar cocok saat dibandingkan. Setelah itu, file CSV dibuka dalam mode baca, dan jika gagal dibuka, program akan menampilkan pesan kesalahan. Baris pertama dari file, yang kemungkinan merupakan header, dilewati. Setiap baris berikutnya diproses untuk memisahkan kolom dengan fungsi `strtok`. Jika nama dokter yang ditemukan di kolom pertama cocok dengan input, program akan melewati enam kolom pertama dan mengambil kolom ketujuh yang diasumsikan berisi total pelanggaran sistem terhadap dokter tersebut. Nilai ini lalu ditampilkan. Jika hingga akhir file tidak ditemukan nama dokter yang dimaksud, maka program akan memberitahu bahwa dokter tersebut tidak ditemukan dalam data. Setelah proses selesai, file ditutup.

L. Flowchart fungsi jumlah pelanggaran semua dokter



Gambar 42. Flowchart fungsi *lihat_jumlah_pelanggaran_per_dokter()*

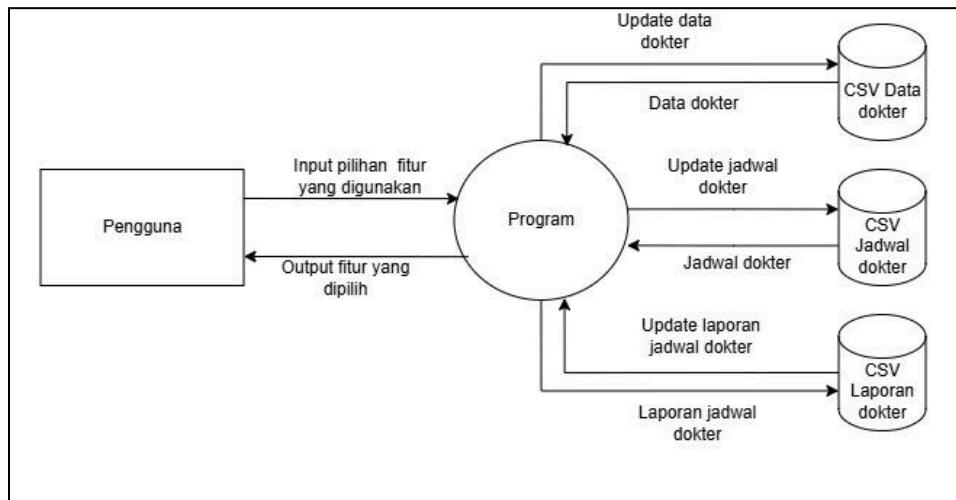
digunakan untuk menampilkan total pelanggaran sistem terhadap setiap dokter yang tercatat dalam file `laporan_dokter.csv`. Program membuka file tersebut dalam mode baca, dan jika gagal, akan menampilkan pesan kesalahan. Baris pertama dari file (kemungkinan berisi header) dilewati. Kemudian, setiap baris dibaca dan diproses menggunakan strtok untuk memisahkan kolom berdasarkan koma. Kolom pertama diambil sebagai nama dokter dan disimpan dalam variabel `nama_dokter`. Setelah itu, enam kolom berikutnya dilewati karena dianggap bukan informasi yang diperlukan. Kolom ketujuh, yang diasumsikan berisi total pelanggaran sistem terhadap dokter tersebut, diambil dan dicetak bersama dengan nama dokter. Nilai pelanggaran hanya diambil satu karakter karena disimpan

sebagai char, bukan string atau angka penuh, sehingga ini bisa menjadi kelemahan jika nilainya lebih dari satu digit. Setelah semua baris diproses, file ditutup.

3.2 Data Flow Diagram

3.2.1 Program Utama

1. DFD Level 0



Gambar 43. DFD level 0 program utama main.c

Komponen utama:

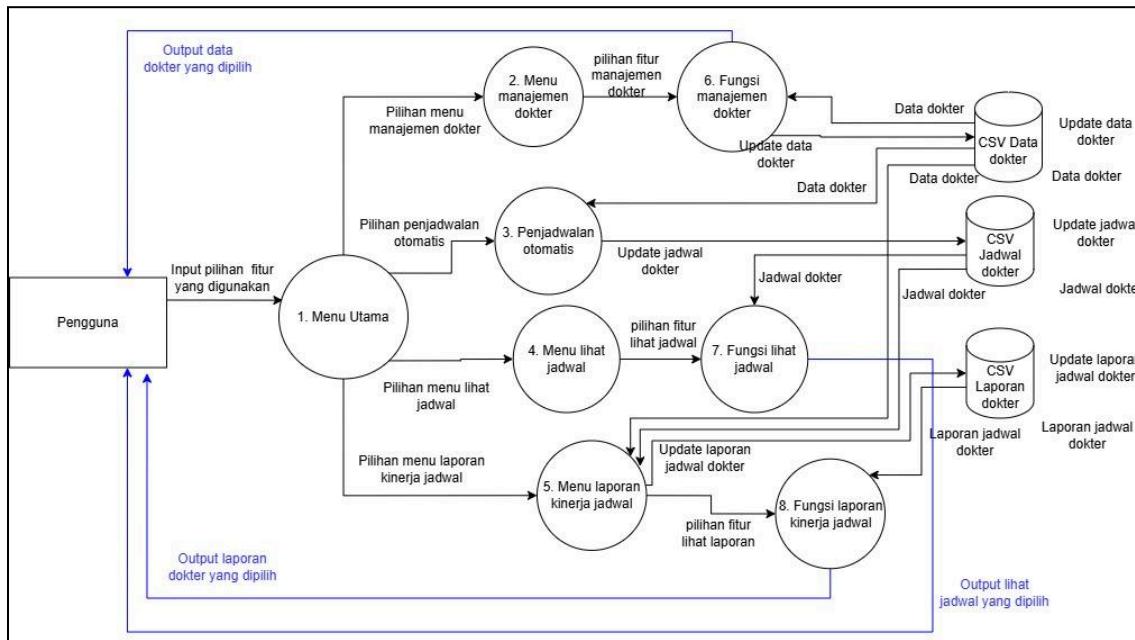
- Pengguna: Entitas yang menjadi pengguna program
- Program: Program utama (algoritma)
- CSV Data dokter: file penyimpanan data dokter beserta preferensinya
- CSV Jadwal dokter: file penyimpanan data jadwal dokter yang sudah dibuat
- CSV Laporan dokter: file penyimpanan laporan dokter yang dibuat dari perbandingan data dokter dengan jadwal dokter. Menghasilkan informasi jumlah shift per dokter dan total pelanggaran terhadap preferensi tiap dokter.

Alur data:

- Pengguna akan memberikan input pilihan fungsi yang ingin dioperasikan oleh program.

- Program akan menerima data dari semua CSV dan mulai melakukan fungsi yang dipilih oleh pengguna.
- Program mengolah data yang diberikan oleh tiap CSV sesuai kebutuhan dan perintah dari pengguna.
- Program akan memberikan output tampilan dari fungsi yang diminta oleh pengguna.

2. DFD Level 1



Gambar 44. DFD level 1 program utama main.c

Komponen:

- 1. Menu Utama

Input: input pilihan fitur oleh pengguna

Output: submenu yang dipilih

Fungsi: antarmuka pengguna untuk memilih submenu program yang ingin digunakan.

- 2. Menu Manajemen dokter:

Input: pilihan submenu manajemen dokter oleh pengguna

- Output: fitur/fungsi manajemen dokter yang dipilih pengguna
Fungsi: antarmuka pengguna untuk memilih fungsi manajemen dokter yang ingin digunakan
- 3. Penjadwalan otomatis
Input: pilihan submenu penjadwalan otomatis oleh pengguna dan CSV data dokter
Output: update jadwal dokter baru
Fungsi: membuat jadwal dokter baru berdasarkan informasi CSV data dokter
- 4. Menu Lihat jadwal
Input: pilihan submenu lihat jadwal
Output: fitur/fungsi lihat jadwal yang dipilih pengguna
Fungsi: antarmuka pengguna untuk memilih fungsi lihat jadwal yang diinginkan
- 5. Menu Laporan kinerja jadwal
Input: pilihan submenu laporan kinerja jadwal, CSV data dokter, dan CSV jadwal dokter
Output: fitur/fungsi lihat laporan kinerja jadwal yang dipilih pengguna dan CSV laporan kinerja jadwal
Fungsi: antarmuka pengguna untuk memilih fungsi lihat laporan kinerja jadwal yang diinginkan
- 6. Fungsi Manajemen dokter
Input: fitur/fungsi manajemen dokter yang dipilih dan CSV data dokter
Output: tampilan dari fitur/fungsi manajemen dokter yang dipilih dan update CSV data dokter
Fungsi: memanipulasi data dokter dan menampilkannya ke pengguna
- 7. Fungsi lihat jadwal
Input: fitur/fungsi lihat jadwal yang dipilih dan CSV jadwal dokter
Output: tampilan dari fitur/fungsi lihat jadwal yang dipilih
Fungsi: menampilkan jadwal dokter ke pengguna
- 8. Fungsi laporan kinerja jadwal

Input: fitur/fungsi laporan kinerja jadwal yang dipilih dan CSV laporan kinerja jadwal

Output: tampilan dari fitur/fungsi laporan kinerja jadwal yang dipilih

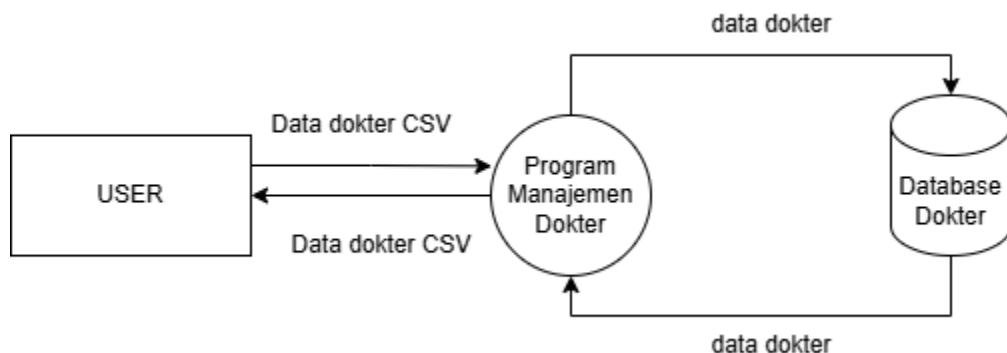
Fungsi: menampilkan laporan kinerja jadwal ke pengguna.

Alur data:

Pengguna memilih submenu → pengguna memilih fitur/fungsi pada submenu → fitur/fungsi melakukan proses terhadap data CSV yang didapatkan berdasarkan input yang diberikan → fitur/fungsi menampilkan hasil prosesnya ke pengguna dan/atau update CSV dengan data yang sudah diolah.

3.2.2 Pengelolaan Data Dokter

a. DFD Level 0



Gambar 45. DFD Level 0 Pengelolaan Data Dokter

Komponen Utama:

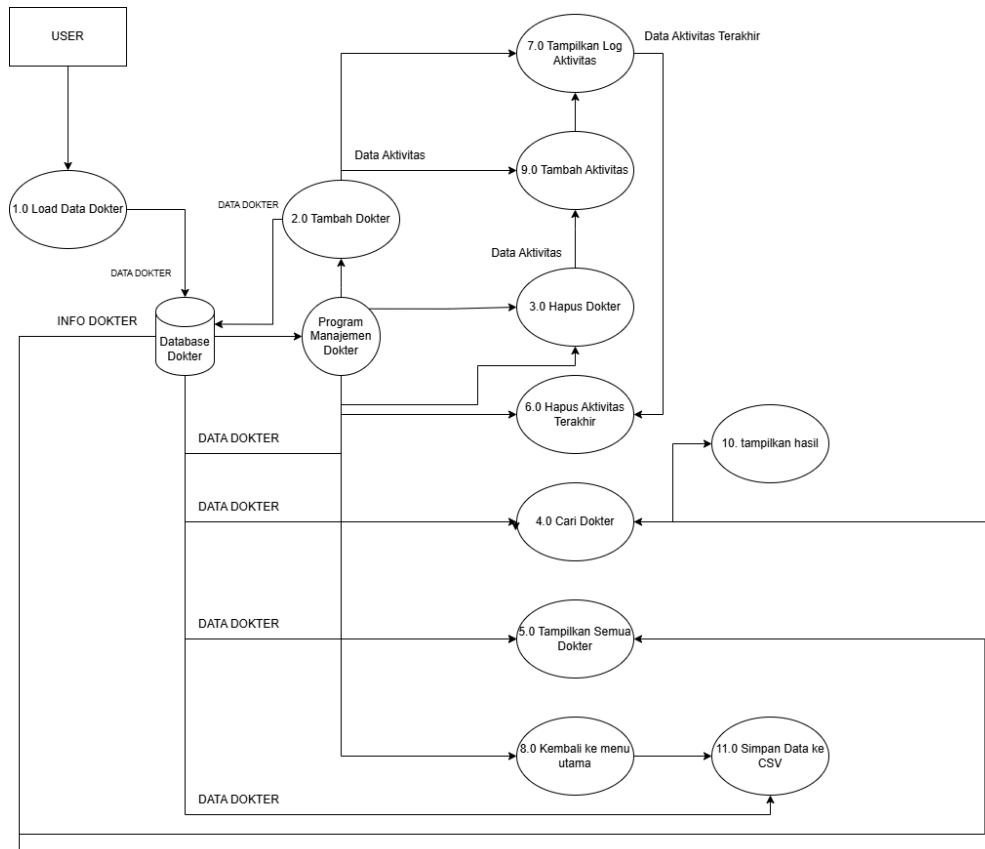
- User: Pengguna sistem yang berinteraksi dengan program.
- Program Manajemen Dokter: Proses inti yang mengelola data dokter.
- Database Dokter (CSV): Tempat menyimpan data dokter dalam format file CSV.

- Aktivitas Log: Tempat menyimpan log aktivitas tambah/hapus dokter

Alur Data:

- User mengirim data dokter (CSV) ke sistem
- Sistem memvalidasi data dan menyimpannya ke Database Dokter
- Sistem mengambil info dokter dari database untuk berbagai fungsi
- Sistem menyimpan data dokter ke dalam format CSV

b. DFD Level 1



Gambar 46. DFD Level 1 Pengelolaan Data Dokter

1.0 Load Data Dokter:

- Input: Data dokter CSV dari user.
- Output: Data dokter valid ke database dokter.
- Fungsi: Memuat dan memvalidasi data dokter dari file CSV.

2.0 Tambah Dokter

- Input: Data dokter dari user
- Output: Data dokter yang valid ke database dokter
- Fungsi: Menambahkan dokter baru ke linked list dan menyimpannya ke file CSV.

3.0 Hapus Dokter

- Input: Nama Dokter
- Output: Dokter yang dihapus dari database dokter
- Fungsi: Menghapus dokter dari linked list dan memperbarui file CSV.

4.0 Cari Dokter

- Input: Keyword untuk pencarian (Nama/Bidang/Tingkat)
- Output: Hasil pencarian dokter
- Fungsi: Proses pencarian berdasarkan kata kunci dan menampilkan hasilnya

5.0 Tampilkan Semua Dokter

- Input: Permintaan dari user
- Output: Menampilkan daftar semua dokter
- Fungsi: Mencetak semua data dokter yang ada

6.0 Hapus Aktivitas Terakhir

- Input: Permintaan Undo dari User
- Output: Aktivitas Terakhir Dihapus dari Log
- Fungsi: Menghapus aktivitas terakhir yang dicatat

7.0 Tampilkan Log Aktivitas

- Input: Permintaan dari User
- Output: Menampilkan log aktivitas kepada pengguna
- Fungsi: Mencetak setiap aktivitas yang dilakukan

8.0 Kembali ke Menu utama

- Input : Permintaan dari User
- Output : Tampilan kembali beralih ke menu utama program
- Fungsi : Mengizinkan pengguna untuk keluar dari submenu dan kembali ke menu utama

9.0 Tambah Aktivitas

- Input : Aksi (Tambah/Hapus) dan Data Dokter
- Output: Aktivitas Tercatat ke Log Aktivitas
- Fungsi: Mencatat aktivitas yang dilakukan untuk undo.

10.0 Tampilkan Hasil

- Input : Linked list dokter hasil pencarian
- Output : Daftar dokter sesuai dengan hasil pencarian
- Fungsi : Menampilkan dokter yang sesuai dengan kriteria pencarian

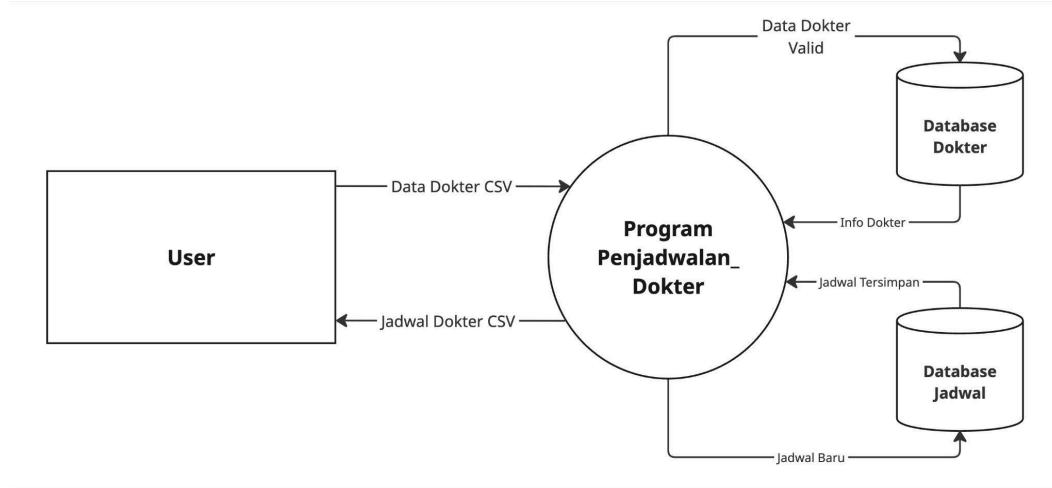
11.0 Simpan Data ke CSV

- Input: Data Dokter dari linked list
- Output: Data Dokter dalam format CSV
- Fungsi: Menyimpan data dokter ke file CSV

Alur kerja keseluruhan: Sistem dimulai dengan memuat data dokter dari file CSV. Pengguna dapat menambah, menghapus, dan mencari dokter, serta menampilkan semua dokter yang ada. Setiap aktivitas dicatat untuk memungkinkan undo, dan data dokter disimpan kembali dalam format CSV. Pengguna juga dapat melihat log aktivitas untuk melacak perubahan yang telah dilakukan. Dengan Program ini, sistem dapat diberikan kemudahan dalam manajemen data dokter secara efisien dan menyimpan data dokter kembali kedalam CSV.

3.2.3 Penentuan Jadwal Otomatis

a. DFD Level 0



Gambar 47. DFD Level 0 Penentuan Jadwal Otomatis

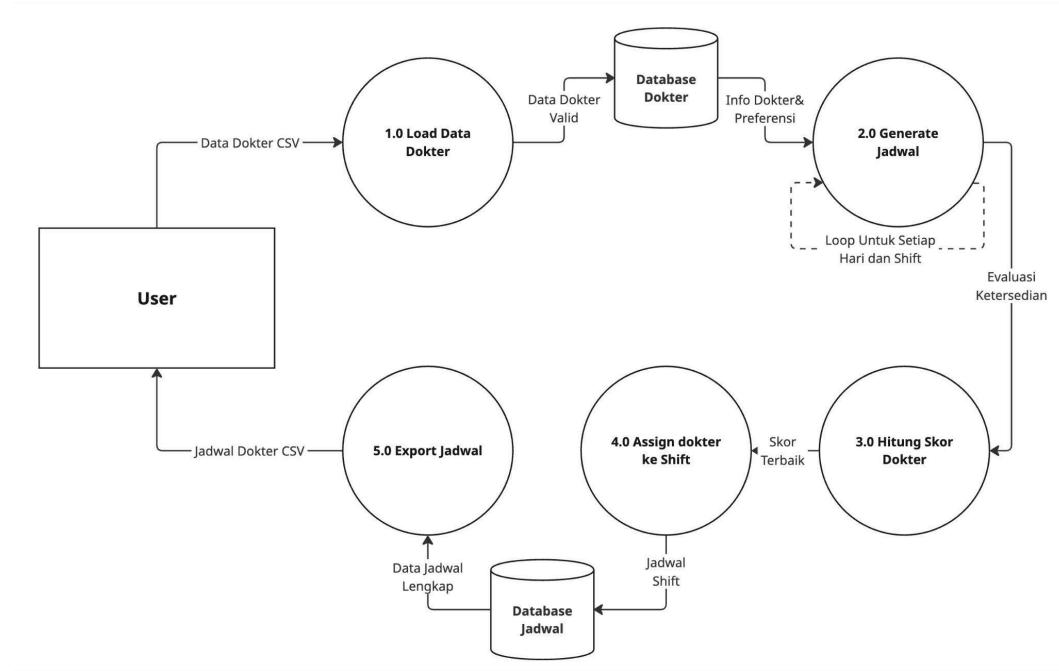
Komponen Utama:

- User: Pengguna sistem
- Program Penjadwalan Dokter: Sistem utama (proses inti)
- Database Dokter: Tempat menyimpan data dokter
- Database Jadwal: Tempat menyimpan data jadwal

Alur Data:

- User mengirim Data Dokter CSV ke sistem
- Sistem memvalidasi data dan menyimpan ke Database Dokter
- Sistem mengambil Info Dokter dari database untuk proses penjadwalan
- Sistem menyimpan Jadwal Tersimpan ke Database Jadwal
- Sistem membuat Jadwal Baru dan mengirim kembali Jadwal Dokter CSV ke User

b. DFD Level 1



Gambar 48. DFD Level 1 Penentuan Jadwal Otomatis

1.0 Load Data Dokter

- Input: Data Dokter CSV dari User
- Output: Data Dokter Valid ke Database Dokter
- Fungsi: Memuat dan memvalidasi data dokter dari file CSV

2.0 Generate Jadwal

- Input: Info Dokter & Preferensi dari Database Dokter
- Output: Masuk ke loop untuk setiap hari dan shift
- Fungsi: Membuat jadwal berdasarkan data dokter dan preferensi mereka

3.0 Hitung Skor Dokter

- Input: Evaluasi Ketersediaan dari proses Generate Jadwal
- Output: Skor Terbaik ke proses Assign Dokter
- Fungsi: Menghitung skor setiap dokter untuk menentukan yang paling cocok

4.0 Assign Dokter ke Shift

- Input: Skor Terbaik dari proses Hitung Skor
- Output: Jadwal Shift ke Database Jadwal
- Fungsi: Menugaskan dokter terbaik ke shift tertentu

5.0 Export Jadwal

- Input: Data Jadwal Lengkap dari Database Jadwal
- Output: Jadwal Dokter CSV ke User
- Fungsi: Mengekspor jadwal final dalam format CSV untuk User

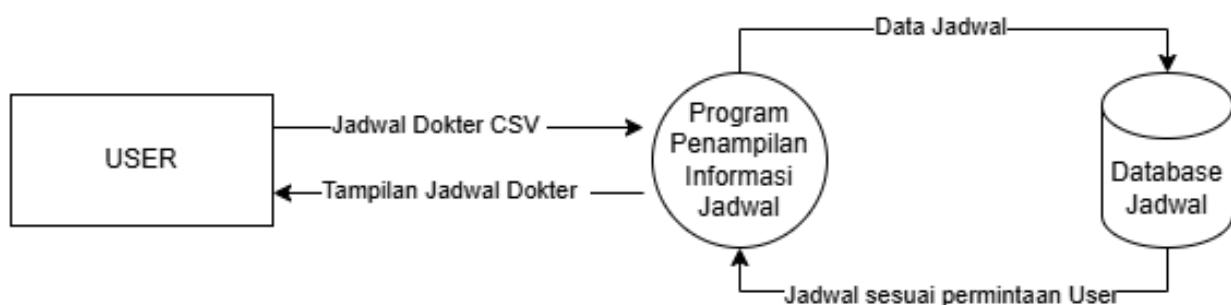
Alur Kerja Keseluruhan:

1. Sistem memuat data dokter → 2. Generate jadwal → 3. Hitung skor optimal →
4. Assign dokter ke shift → 5. Export hasil ke User

Dengan program ini, sistem dapat membuat penjadwalan otomatis yang membantu mengatur jadwal kerja dokter berdasarkan ketersediaan dan preferensi mereka, kemudian memberikan hasil optimal kembali kepada pengguna.

3.2.4 Penampilan Informasi Jadwal dan Laporan

a. DFD Level 0 Informasi Jadwal



Gambar 49. DFD level 0 Informasi Jadwal

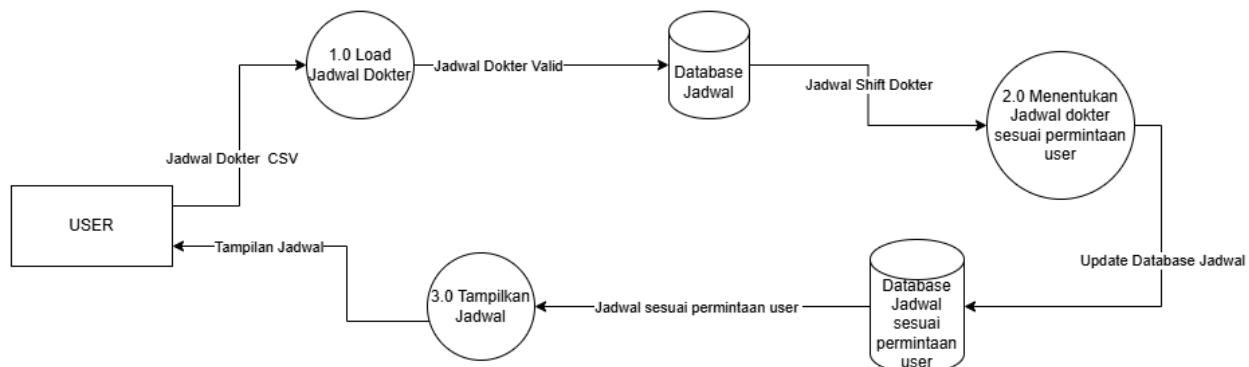
Komponen Utama:

- User: Pengguna Sistem
- Program Penampilan Informasi Jadwal: Sistem Utama (proses inti)
- Database Jadwal: Tempat menyimpan data jadwal dokter per hari/minggu/bulan.

Alur Data:

- User mengirim Data Jadwal CSV ke sistem
- Sistem memvalidasi data dan menyimpan ke Database Jadwal
- Sistem mengambil jadwal dokter dari database untuk ditentukan sesuai pilihan user
- Sistem menampilkan jadwal sesuai pilihan user

b. DFD Level 1 Informasi Jadwal



Gambar 50. DFD Level 1 Informasi Jadwal

1.0 Load Jadwal Dokter

- Input : Jadwal Dokter CSV dari user
- Output : Jadwal Dokter Valid ke database jadwal
- Fungsi : Memuat dan memvalidasi jadwal dokter dari file csv

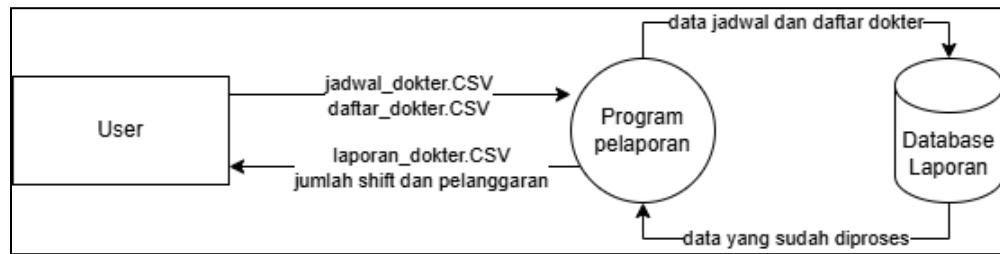
2.0 Menentukan jadwal dokter sesuai permintaan user

- Input : Jadwal shift dokter
- Output : Jadwal dokter sesuai permintaan user
- Fungsi : Mengupdate database jadwal dengan jadwal yang diinginkan user

3.0 Tampilkan Output

- Input : Database jadwal yang sudah di update
- Output : Tampilan jadwal shift dokter ke user
- Fungsi : Menampilkan jadwal shift ke user

c. DFD Level 0 Pelaporan



Gambar 51. DFD Level 0 Pelaporan

Komponen utama :

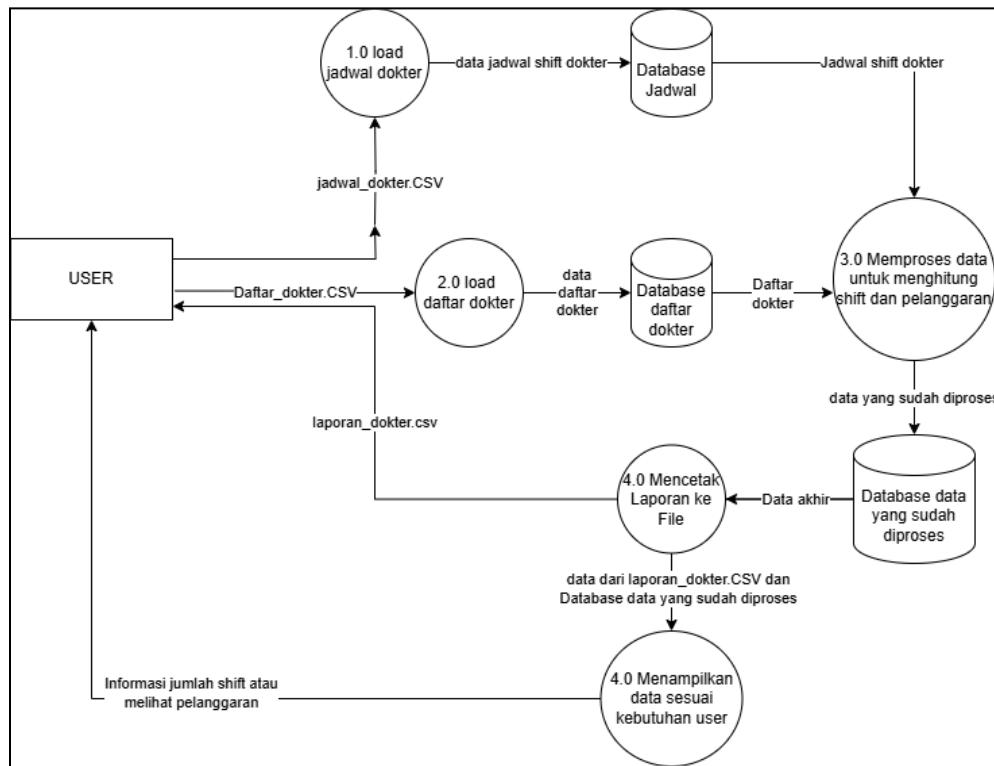
- User : Pengguna program
- Program pelaporan : sistem utama
- Database laporan : tempat penyimpanan data

Alur Data :

- Program menerima data dari file jadwal_dokter.CSV dan daftar_dokter.CSV
- Program memasukkan data tersebut ke database
- Program mengolah data yang diambil dari database

- Data yang sudah jadi akan dikirimkan kembali ke user dalam bentuk file laporan_dokter.CSV dan informasi jumlah shift serta pelanggaran.

d. DFD Level 1 Pelaporan



Gambar 52. DFD Level 1 Pelaporan

1.0. Load jadwal dokter

- Input : file jadwal_dokter.CSV
- Output : Jadwal dokter
- Fungsi : Memparsing data dari file ke program

2.0. Load daftar dokter

- Input : file daftar_dokter.CSV

- Output : daftar dokter
- Fungsi : Memparsing data dari file ke program

3.0. Memproses data

- Input : Jadwal dokter dan daftar dokter
- Output : gabungan data serta jumlah shift dan pelanggaran
- Fungsi : mengolah data dari dua file agar siap disajikan

4.0 Mencetak laporan ke file

- Input : Data akhir
- Output : file laporan_dokter.csv
- Fungsi : menyajikan laporan kepada user dalam bentuk file berformat csv

5.0. Menampilkan jumlah shift atau melihat pelanggaran

- Input : data akhir dan laporan_dokter.csv
- Output : jumlah shift per dokter atau jumlah pelanggaran
- Fungsi : menampilkan jumlah shift per dokter atau jumlah pelanggaran tergantung kebutuhan user

BAB IV

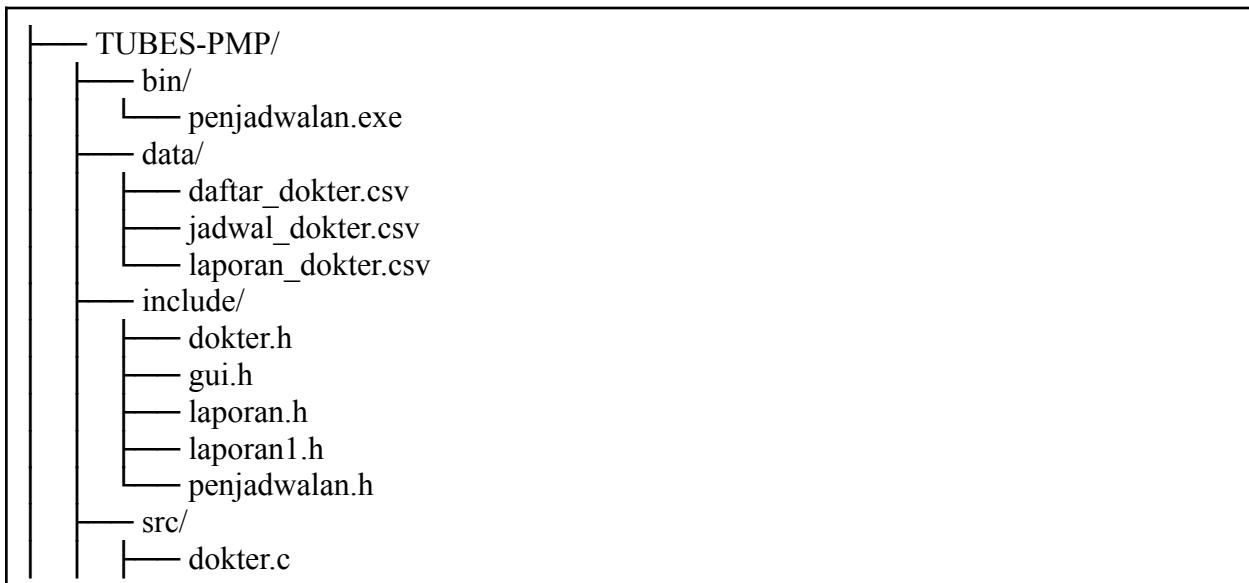
IMPLEMENTASI SISTEM

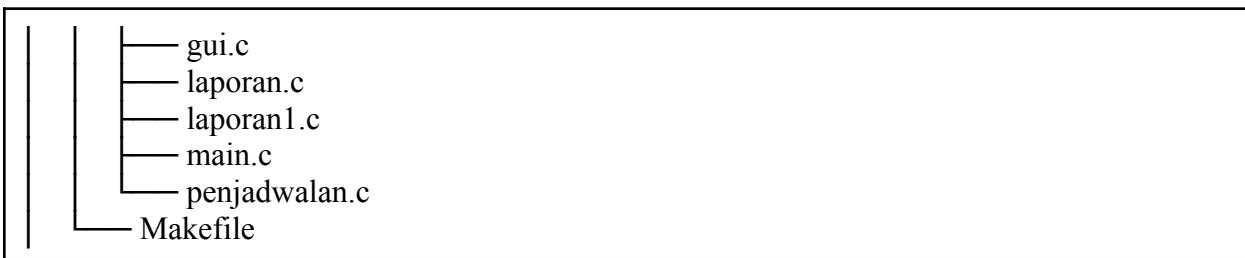
Source code implementasi sistem tercantum pada repository GitHub terlampir pada bagian lampiran di akhir dokumen ini. Bagian ini akan menjelaskan logika cara kerja dan struktur data yang digunakan (jika ada) pada program dan sub program telah dirancang dan diaplikasikan dalam program bahasa C pada repository tersebut.

4.1. Struktur file program

Struktur file program menggunakan struktur yang umum digunakan pada program berbasis C. program memiliki 4 folder dan 1 file yaitu bin, data, include, src, dan Makefile:

- a. bin: berfungsi sebagai tempat penyimpanan program yang sudah dicompile
- b. data: berfungsi sebagai tempat penyimpanan database, dalam kasus ini file berformat .csv
- c. include: berfungsi sebagai tempat penyimpanan file header untuk di-include sebagai library, berisikan file berformat .h
- d. src: berfungsi sebagai tempat penyimpanan source code untuk header dan program utama, berisikan file berformat .c
- e. Makefile: berfungsi sebagai preset perintah yang digunakan untuk compile seluruh file dengan mudah dan agar proses compile hanya dilakukan pada file yang diubah saja.





4.2. Program Utama

4.2.1. Penjelasan logika

Program utama menggunakan tampilan dengan konsep logika selayaknya struktur tree. Cabang dari tree ini adalah submenu dari subprogram yang ada pada program.

- Fungsi main dari program utama diawali dengan membaca file data_dokter.csv
- Program memberikan menu berisi pilihan subprogram yang bisa digunakan pengguna. Pengguna memilih dengan klik menu yang ada di window.
- Setelah pengguna memilih, program akan masuk ke submenu berikutnya.
- Fungsi menu_manajemen_dokter(), menu_lihat_jadwal(), dan menu_laporan_kinerja() masing-masing memberikan menu baru untuk subprogram dengan nama yang sama. Ketika fungsi dalam tiap subprogram dipilih oleh pengguna, program akan langsung mengeksekusi dan menampilkan output dari fungsi yang dipilih.
- menu_penjadwalan_otomatis() berfungsi untuk menjalankan algoritma untuk membuat jadwal berdasarkan data dokter yang tersimpan di data_dokter.csv
- Ketika pengguna memilih menu untuk keluar dari program, sistem akan menyimpan semua perubahan pada file data_dokter.csv dan menutupnya. Lalu semua memori akan dibebaskan dan program pun selesai.

4.3. Pengelolaan Data Dokter

4.3.1. Penjelasan logika

- Program memuat data dari file CSV saat pertama kali dijalankan
- Jika file tidak ada, program akan membuat file baru dengan header
- Setiap penambahan atau penghapusan dokter akan:

- Memodifikasi linked list dokter
 - Mencatat aktivitas ke dalam log
 - Menyimpan perubahan ke file CSV
- Sistem dapat membatalkan operasi terakhir dengan memanfaatkan log aktivitas
 - Operasi undo akan mengembalikan sistem ke keadaan sebelumnya dan menyimpan perubahan.
 - Pencarian dilakukan dengan membuat salinan sementara dari data yang sesuai kriteria
 - Hasil pencarian ditampilkan dan kemudian dibersihkan dari memori
 - Saat program ditutup, semua alokasi memori dibersihkan
 - Perubahan terakhir disimpan ke file CSV

Program ini merupakan sistem manajemen data dokter untuk rumah sakit yang mencakup fitur-fitur berikut:

1. Manajemen Data Dokter:
 - a. Menambah dokter baru secara manual
 - b. Menghapus dokter dari sistem
 - c. Mencari dokter berdasarkan berbagai kriteria (nama, bidang, tingkat)
 - d. Menampilkan semua data dokter
2. Sistem Undo:
 - a. Mencatat semua aktivitas penambahan dan penghapusan dokter
 - b. Memungkinkan pembatalan operasi terakhir (undo)
 - c. Menyimpan log aktivitas
3. Penyimpanan Data:
 - a. Menyimpan data dalam format CSV
 - b. Membuat direktori dan file secara otomatis jika belum ada
 - c. Memuat data dari file saat program dimulai

4.3.2. Struktur data

- Struktur Utama

1. Struktur Dokter (Dokter) :

- a. nama[LEN]: Nama dokter (string)
- b. bidang[LEN]: Bidang spesialisasi dokter (string)
- c. tingkat[LEN]: Tingkat dokter (Koass/Residen/Spesialis/Konsulen)
- d. max_shift_per_minggu: Batas maksimum shift per minggu (integer)
- e. preferensi_shift[LEN]: Preferensi shift (Pagi/Siang/Malam)
- f. preferensi_waktu[LEN]: Preferensi waktu (Campur/Awal Bulan/Ahir Bulan)
- g. next: Pointer ke dokter berikutnya dalam linked list

2. Struktur Aktivitas (Aktivitas) :

- a. tipe: Jenis aksi (AKSI_TAMBAH/AKSI_HAPUS)
- b. dokterData: Pointer ke data dokter yang terkait dengan aktivitas
- c. next: Pointer ke aktivitas berikutnya dalam linked list

3. Enum:

- a. AksiType: Menyimpan jenis aksi yang dilakukan (penambahan atau penghapusan dokter)

4. Konstanta:

- a. LEN: Panjang maksimum string untuk berbagai field
- b. OUTPUT_FILE_DOKTER: Nama file untuk menyimpan data dokter-
- c. AKSI_TAMBAH dan AKSI_HAPUS: Untuk menandai jenis aktivitas (penambahan atau penghapusan dokter).

5. Variabel Global:

- a. head: Pointer ke kepala linked list dokter
- b. aktivitas_head: Pointer ke kepala linked list aktivitas

- Fungsi Utama

1. Manajemen Dokter:

- a. tambah_dokter_manual(): Menambahkan dokter baru dengan input manual

- b. hapus_dokter(): Menghapus dokter berdasarkan nama
 - c. cari_dokter_nama(), cari_dokter_bidang(), cari_dokter_tingkat(): Fungsi pencarian dengan kriteria berbeda
 - d. tampilkan_semua(): Menampilkan semua data dokter
 - e. cari_dokter_menu(): Menampilkan submenu untuk memilih metode pencarian dokter (berdasarkan nama, bidang, atau tingkat).
 - f. tampilkan_hasil(Dokter* hasil) : Menampilkan hasil pencarian dokter berdasarkan kriteria tertentu.
2. Sistem Undo:
- a. tambah_aktivitas(): Mencatat aktivitas ke dalam log
 - b. hapus_aktivitas_terakhir(): Membatalkan operasi terakhir (undo)
 - c. tampilkan_log(): Menampilkan log aktivitas
3. Manajemen File:
- a. load_data_dari_csv(): Memuat data dari file CSV
 - b. save_data_to_csv(): Menyimpan data ke file CSV
 - c. free_memory(): Membersihkan semua alokasi memori
4. Fungsi Pembantu:
- a. salin_dokter(): Membuat salinan data dokter
 - b. clean_newline(): Membersihkan karakter newline dari string
 - c. contains_keyword(): Pencarian string case-insensitive

4.4. Penentuan Jadwal Otomatis

4.4.1. Penjelasan logika

Kode ini membuat sistem penjadwalan dokter untuk rumah sakit dengan menggunakan **algoritma greedy**. Tujuannya adalah untuk menghasilkan jadwal bulanan dokter dengan mempertimbangkan preferensi shift, tingkat dokter, dan batasan shift mingguan. Berikut adalah logika utama dari kode:

4.4.1.1. Memuat Data Dokter dari File CSV (load_doctors_from_csv):

- a. Fungsi ini akan membaca data dokter dari file CSV, dengan setiap baris berisi informasi seperti nama, bidang spesialisasi, tingkat dokter (Koass, Residen, Spesialis, Konsulen), jumlah maksimum shift per minggu, preferensi shift (Pagi, Siang, Malam), dan preferensi waktu (awal bulan, akhir bulan, atau campur).
- b. Data disimpan ke dalam array global doctors dengan batas maksimum dokter (MAX_DOCTORS).
- c. Pemisahan/ Parsing dilakukan dengan memisahkan setiap baris CSV menggunakan koma sebagai pemisah, dan string diubah menjadi tipe enum yang sesuai (ShiftType, TingkatDokter, PreferensiWaktu).
- d. Jika file tidak valid atau tidak ada dokter yang berhasil dimuat, fungsi mengembalikan 0 (gagal).

4.4.1.2. Menghasilkan Jadwal Bulanan (Algoritma Greedy):

- a. Fungsi ini menggunakan pendekatan algoritma greedy untuk menentukan jadwal dokter untuk setiap hari (30 hari dalam sebulan) dan setiap shift (Pagi, Siang, Malam).
- b. Untuk setiap hari dan shift, fungsi mencoba menugaskan dua dokter (dijelaskan oleh DOCTORS_PER_SHIFT).
- c. Proses pemilihan dokter:
 - i. **Skoring:** Setiap dokter yang tersedia dievaluasi berdasarkan skor yang dihitung oleh fungsi calculate_score. Skor ini mempertimbangkan:
 1. Kesesuaian dengan preferensi shift (Pagi, Siang, Malam).
 2. Kesesuaian dengan preferensi waktu (awal/akhir bulan).
 3. Jumlah shift yang masih tersedia dalam minggu tersebut (untuk menghindari kelebihan shift).
 4. Tingkat dokter (Koass, Residen, Spesialis, Konsulen) untuk memberikan prioritas.
 5. Jumlah total shift yang sudah diambil dokter (untuk distribusi yang merata).

- ii. **Pemilihan Dokter:** Dokter dengan skor tertinggi dipilih, asalkan belum dijadwalkan pada shift yang sama dan masih memenuhi batas shift mingguan.
- iii. **Penugasan:** Dokter yang dipilih ditambahkan ke jadwal, dan statistik shift mereka (total shift dan shift mingguan) diperbarui.
- d. Jadwal disimpan dalam array global schedule.

4.4.1.3. Menyimpan Jadwal ke File CSV (save_schedule):

- a. Jadwal yang dihasilkan disimpan ke file CSV dengan nama yang ditentukan oleh konstanta OUTPUT_FILE_PENJADWALAN.
- b. Format output mencakup kolom: Tanggal, Hari, Shift, Nama Dokter, Bidang, dan Tingkat Dokter.
- c. Data ditulis untuk setiap dokter yang dijadwalkan pada setiap shift.

4.4.1.4. Fungsi Pembantu:

- a. get_week_number: Menghitung minggu ke berapa dari hari tertentu (dihitung sebagai day / 7).
- b. is_early_month: Mengecek apakah hari berada di awal bulan (hari ≤ 15).
- c. is_doctor_available: Selalu mengembalikan 1 (dokter dianggap tersedia, tetapi batasan shift ditangani di scoring).
- d. is_doctor_already_assigned: Memastikan dokter belum dijadwalkan pada hari dan shift yang sama.
- e. calculate_score: Menghitung skor dokter berdasarkan preferensi shift, preferensi waktu, sisa shift mingguan, total shift, dan tingkat dokter.
- f. assign_shift: Memperbarui statistik dokter (total shift dan shift mingguan) setelah penugasan.
- g. parse_shift, parse_tingkat, parse_waktu: Mengonversi string dari CSV ke enum untuk shift, tingkat dokter, dan preferensi waktu.

4.4.2. Struktur data

Kode ini menggunakan beberapa struktur data untuk menyimpan dan mengelola informasi dokter dan jadwal. Struktur data utama adalah sebagai berikut:

4.4.2.1. Struktur Dokter (Doctor)

Tabel 1. Struktur Data Doctor

Atribut	Tipe Data	Deskripsi
name	String (maks. MAX_NAME_LEN)	Nama dokter.
bidang	String (maks. MAX_NAME_LEN)	Bidang spesialisasi dokter.
tingkat	Enum TingkatDokter	Tingkat dokter (Koass, Residen, Spesialis, Konsulen)
max_shifts_per_weeks	Integer	Batas maksimum shift per minggu.
preferred_shift	Enum ShiftType	Shift yang diinginkan (Pagi, Siang, Malam)
preferred_time	Enum PreferensiWaktu	Preferensi waktu (Awal Bulan, Akhir Bulan, Campur).
total_shifts_assigned	Integer	Jumlah total shift yang sudah diambil dokter.
weekly_shifts	Array Integer	Jumlah shift per minggu (diasumsikan 5 minggu dalam sebulan)
weekly_shifts	Array global untuk menyimpan semua dokter (maks. MAX_DOCTORS).	Array global untuk menyimpan semua dokter (maks. MAX_DOCTORS).

4.4.2.2. Struktur Jadwal (Schedule)

Tabel 2. Struktur Data Jadwal

Atribut	Tipe Data	Deskripsi
day	Integer	Hari dalam bulan (0 hingga 29, mewakili 30 hari).
shift	Integer	Jenis shift (0 = Pagi, 1 = Siang, 2 = Malam).
num_doctors	Integer	Jumlah dokter yang dijadwalkan pada shift tersebut.
doctor_ids	Array Integer (MAX_DOCTORS_PER_SHIFT)	ID dokter yang dijadwalkan (indeks ke array doctors).
Penyimpanan	Array schedule[DAY_IN_MONTH * SHIFTS_PER_DAY]	Array global untuk menyimpan semua jadwal (30 hari × 3 shift = 90 entri).

4.4.2.3. Enum:

Tabel 3. Struktur Data Enum

Enum	Nilai
ShiftType	SHIFT_PAGI, SHIFT_SIANG, SHIFT_MALAM

TingkatDokter	TINGKAT_KOASS, TINGKAT_RESIDEN, TINGKAT_SPESIALIS, TINGKAT_KONSULEN
PreferensiWaktu	WAKTU_AWAL_BULAN, WAKTU_AKHIR_BULAN, WAKTU_CAMPUR

4.4.2.4. Konstanta:

Tabel 4. Struktur Data Konstanta

Konstanta	Nilai	Deskripsi
MAX_DOCTORS	-	Batas maksimum jumlah dokter.
DAYS_IN_MONTH	30	Jumlah hari dalam sebulan.
SHIFTS_PER_DAY	3	Jumlah shift per hari (Pagi, Siang, Malam).
DOCTORS_PER_SHIFT	2	Jumlah dokter per shift.
MAX_DOCTORS_PER_SHIFT	(sama dengan DOCTORS_PER_SHIFT)	Batas maksimum dokter per shift.
LINE_BUFFER_SIZE	-	Ukuran buffer untuk membaca baris dari file CSV.
OUTPUT_FILE_PENJADWALAN	-	Nama file output untuk menyimpan jadwal

4.4.2.5. Array Pendukung:

Tabel 5. Struktur Data Array Pendukung

Array	Tipe Data	Deskripsi
hari	Array String	Nama hari (Senin, Selasa, dst.).
shift_names	Array String	Nama shift (Pagi, Siang, Malam).
tingkat_str	Array String	Nama tingkat dokter (Koass, Residen, Spesialis, Konsulen).

4.5. Penampilan Informasi Jadwal dan Laporan

4.5.1. Penjelasan logika

Program Laporan : Program ini bertujuan untuk memuat data jadwal dan preferensi dokter, kemudian memproses dan memeriksa kesesuaian antara jadwal dengan preferensi, serta menghasilkan laporan pelanggaran dan jumlah shift dokter. Program dibangun dalam bentuk modular, artinya fungsi-fungsinya terpisah sesuai tanggung jawab masing-masing, namun saling terhubung melalui struktur data global.

Program Penampilan Informasi Jadwal : Program ini bertujuan untuk membuat jadwal yang sudah ditentukan kemudian menentukan jadwal dokter yang sesuai dengan permintaan user dan menampilkan jadwal tersebut ke user. Jadwal dokter akan ditentukan dengan menggunakan algoritma brute force yaitu membandingkan satu-satu semua tanggal shift dokter pada Jadwal Dokter CSV dengan tanggal/minggu yang ingin dilihat user.

4.5.2. Struktur Data

4.5.2.1. Struktur data ShiftEntry

- a. Didefinisikan di header laporan.h
- b. Atribut
 - Tanggal : Integer untuk infomasi tanggal

- Hari[20] : array of character, berisi informasi mengenai hari (Senin, selasa..dst)
- Shift[10] : array of character, berisi informasi mengenai shift (pagi, siang, malam)
- Nama[] : array of character, berisi informasi nama dokter
- Bidang[] : array of character, berisi informasi bidang dari dokter tertentu
- Tingkat[] : array of character, berisi informasi mengenai tingkat dari dokter tertentu

4.5.2.2.Struktur data DoctorPref

- a. Didefinisikan di header laporan.h
- b. atribut
 - nama[MAX_NAME] : array of character berisi informasi nama dokter
 - Bidang[100] ; array of character berisi informasi mengenai bidang dari dokter tertentu
 - Tingkat[20] : array of character berisi informasi mengenai tingaktan dokter
 - Max_shift_per_minggu : int berupa informasi maksimal shift per minggu dari tiap dokter
 - Preferensi_shift[10] : array of character berisi informasi preferensi shifr dokter missalnya pagi siang atau malam
 - Preferensi_waktu[20] : array of character berisi informasi preferensi waktu dari tiap dokter, seperti awal atau akhir bulan.

4.5.2.3.Enum

Tidak terdapat enum

4.5.2.4.Konstanta

- a. MAX_ENTRIES : jumlah maksimal entri data yang dapat ditampung dalam array, baik untuk shift maupun data dokter (maksimum 1000 entri).
- b. MAX_NAME : maksimal panjang nama dokter yang akan disimpan (100 karakter).
- c. MAX_LINE : panjang maksimal satu baris teks yang dibaca dari file CSV (1024 karakter).

- d. INPUT_FILE_JADWAL : nama file input untuk data jadwal shift dokter ("data/jadwal_dokter.csv").
- e. INPUT_FILE_DOKTER : nama file input untuk data preferensi dokter ("data/daftar_dokter.csv").
- f. INPUT_FILE_LAPORAN : nama file input untuk membaca laporan yang sudah disimpan ("data/laporan_dokter.csv"), dipakai saat ingin melihat data seperti total shift.
- g. OUTPUT_FILE_LAPORAN : nama file output untuk menyimpan laporan yang dihasilkan ("data/laporan_dokter.csv"), sama dengan input laporan karena file yang sama dipakai untuk dibaca dan ditulis.

4.5.2.5.Array pendukung

- a. total_shift[MAX_ENTRIES]
 - Menyimpan jumlah total shift yang dimiliki oleh masing-masing dokter.
 - Index ke-i sesuai dengan dokter ke-i dalam doctor_data.
- b. shift_mingguan[MAX_ENTRIES][5]
 - Menyimpan jumlah shift tiap dokter per minggu (maksimal 5 minggu).
 - shift_mingguan[i][j] adalah jumlah shift dokter ke-i pada minggu ke-j.
- c. shift_count_per_type[MAX_ENTRIES][3]
 - Menyimpan jumlah shift berdasarkan jenis (Pagi, Siang, Malam) untuk setiap dokter.
 - Index ke-0 = Pagi, 1 = Siang, 2 = Malam.
- d. pelanggaran[MAX_ENTRIES]
 - Menyimpan jumlah total pelanggaran untuk masing-masing dokter.
 - Digunakan untuk akumulasi pelanggaran saat fungsi proses_data() dipanggil.
- e. tanggal_ke_minggu[32]
 - Array bantu berukuran 32 (tanggal 1–31) untuk memetakan tanggal ke minggu keberapa.
 - Digunakan dalam fungsi lihat_pelanggaran() untuk mengetahui minggu dari sebuah tanggal.

- f. temp_shift_mingguan[MAX_ENTRIES][5]
 - Array lokal dalam fungsi lihat_pelanggaran()
 - Menyimpan jumlah shift per minggu sementara saat memeriksa pelanggaran.
- g. hari[100]
 - Array berukuran 100 untuk membaca hari shift dokter dari Jadwal Dokter CSV
- h. Shift[100]
 - Array berukuran 100 untuk membaca jam shift dokter dari Jadwal Dokter CSV
- i. Nama_dokter[100]
 - Array berukuran 100 untuk membaca nama dokter dari Jadwal Dokter CSV
- j. Bidang[100]
 - Array berukuran 100 untuk membaca bidang dokter dari Jadwal Dokter CSV
- k. Tingkat[100]
 - Array berukuran 100 untuk membaca tingkat dokter dari Jadwal Dokter CSV
- l. Hari_next[100]
 - Array berukuran 100 sebagai indikator perulangan, agar jika array hari sama dengan hari_next maka array hari pada kondisi tersebut tidak akan ditampilkan di output
- m. Shift_next[100]
 - Array berukuran 100 sebagai indikator perulangan, agar jika array shift sama dengan shift_next maka array hari pada kondisi tersebut tidak akan ditampilkan di output
- n. Nama_dokter2[1000][100]
 - Array berukuran 1000x100 ini berfungsi untuk menyimpan nama dokter yang sesuai dengan jadwal permintaan user
- o. Bidang2[1000][100]

- Array berukuran 1000x100 ini berfungsi untuk menyimpan bidang dokter yang sesuai dengan jadwal permintaan user
- p. Tingkat2[1000][100]
- Array berukuran 1000x100 ini berfungsi untuk menyimpan tingkat dokter yang sesuai dengan jadwal permintaan user
- q. Hari2[1000][100]
- Array berukuran 1000x100 ini berfungsi untuk menyimpan hari shift dokter yang sesuai dengan jadwal permintaan user
- r. Shift2[1000][100]
- Array berukuran 1000x100 ini berfungsi untuk menyimpan jam shift dokter yang sesuai dengan jadwal permintaan user

BAB V

STUDI KASUS

Untuk menguji kemampuan dari program, dibuat studi kasus sebagai berikut:

Kondisi awal:

- Pengguna adalah seorang pegawai tata usaha Rumah Sakit Semoga Sehat Selalu. Ia bertugas sebagai manajer penjadwalan shift jaga dokter di rumah sakit tersebut.
- Program sudah di-compile dan sudah siap dipakai.
- Sudah ada data_dokter.csv berisikan daftar dokter yang diisi sebelumnya.
- Belum ada file jadwal_dokter.csv dan file laporan_dokter.csv.

Langkah-langkah studi kasus:

1. Pengguna ingin membuka menu manajemen dokter untuk menambahkan dokter baru:

Nama	:	Mazaya Bari Nafiah
Bidang	:	Bedah Toraks
Tingkat	:	Konsulen
Maks. shift per minggu	:	6
Preferensi shift	:	Siang
Preferensi waktu	:	Awal bulan

2. Pengguna kemudian ingin melihat semua data dokter.
3. Lalu, untuk memastikan, pengguna ingin melihat apakah Mazaya sudah berhasil ditambahkan atau belum.
4. Ketika melihat data semua dokter, pengguna ingin menghapus dokter bernama Rafif Farras.
5. Setelah itu, pengguna lupa perubahan apa saja yang telah Ia lakukan. Sehingga pengguna ingin melihat log aktivitas.
6. Pengguna teringat bahwa Rafif Farras belum seharusnya dikeluarkan dari daftar dokter di RS tersebut di bulan ini. Pengguna kemudian ingin membatalkan aksi sebelumnya.

7. Setelah mengatur data dokter di RS tersebut, Pengguna ingin membuat jadwal secara otomatis dengan mempertimbangkan preferensi para dokter.
8. Setelah jadwal berhasil dibuat, Pengguna ingin melihat jadwal yang telah dibuat. Ia ingin melihat jadwal bulanan, minggu ke-2, dan hari ke-4.
9. Setelah berhasil membuat jadwal, Pengguna ingin memeriksa seberapa sesuai jadwal yang dibuat dengan preferensi para dokter.
10. Setelah laporan berhasil dibuat, Pengguna ingin melihat total shift jaga yang didapatkan oleh dokter bernama Abdul Hakim Yafi.
11. Kemudian, pengguna ingin melihat jumlah pelanggaran preferensi yang dialami semua dokter dan juga yang dialami dokter Abdul Hakim Yafi.
12. Terakhir, Pengguna ingin melihat pelanggaran apa saja yang terjadi terhadap preferensi para dokter.
13. Setelah itu, Pengguna ingin mengetahui apa yang terjadi jika memasukkan input yang salah ke dalam program. Pengguna pun mencoba memasukkan nama dokter, tanggal, dan minggu yang tidak ada pada data.
14. Setelah melakukan semua hal di atas, telah menyelesaikan tugasnya untuk hari itu. Studi kasus pun selesai.

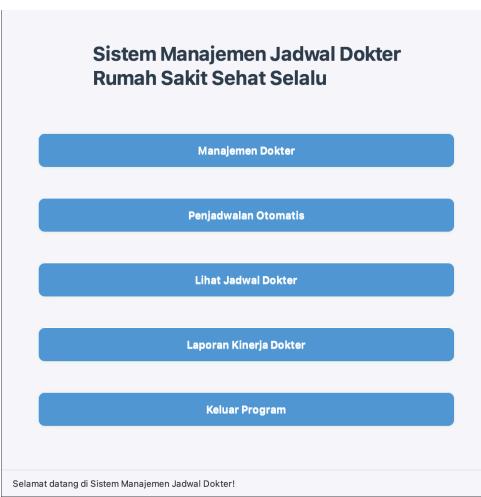
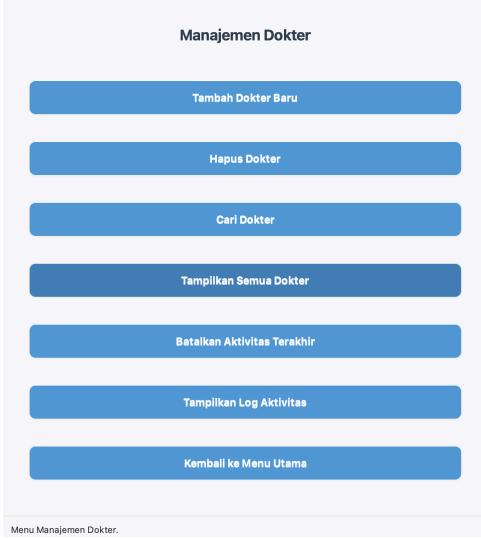
Output yang diharapkan dari studi kasus di atas adalah:

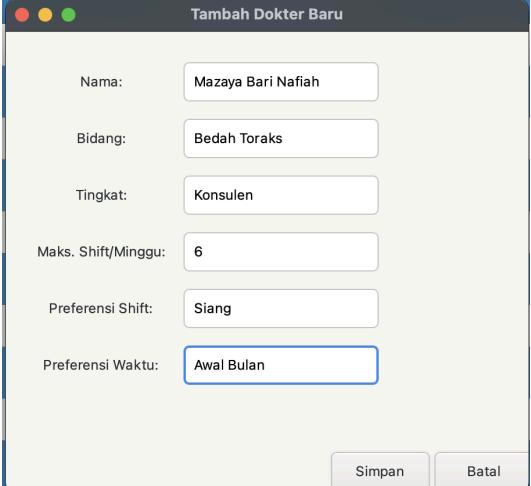
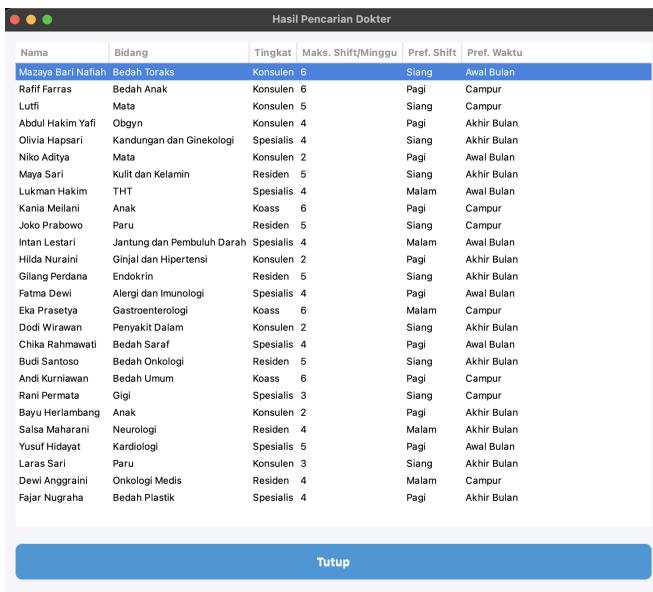
- Tangkapan layar tampilan yang menunjukkan program berhasil melakukan fungsi-fungsi sesuai alur studi kasus di atas.
- File `jadwal_dokter.csv` yang berisikan jadwal dokter lengkap dari tanggal 1-30
- File `laporan_dokter.csv` yang berisikan laporan dokter lengkap dengan data dokter, preferensi, jumlah jadwal, dan pelanggaran yang terjadi.

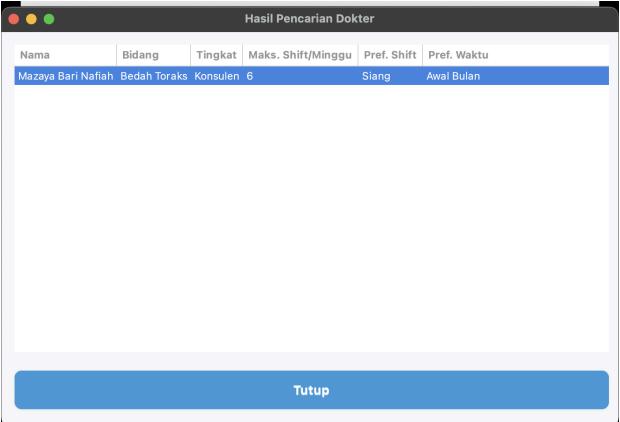
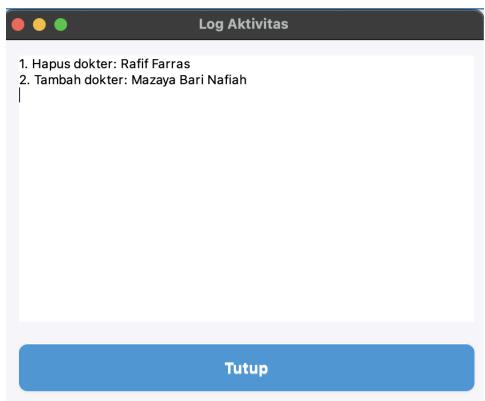
BAB VI

ANALISIS HASIL DAN JUSTIFIKASI

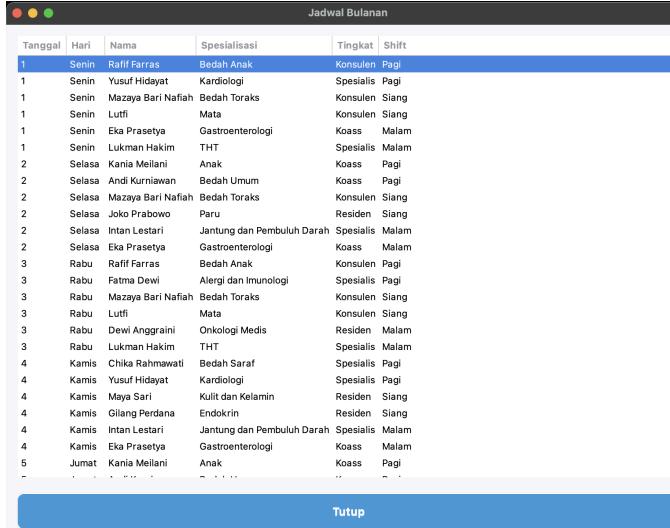
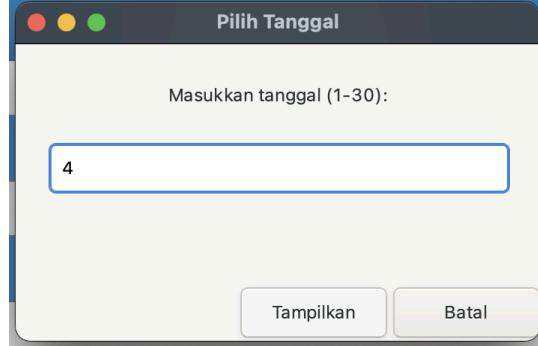
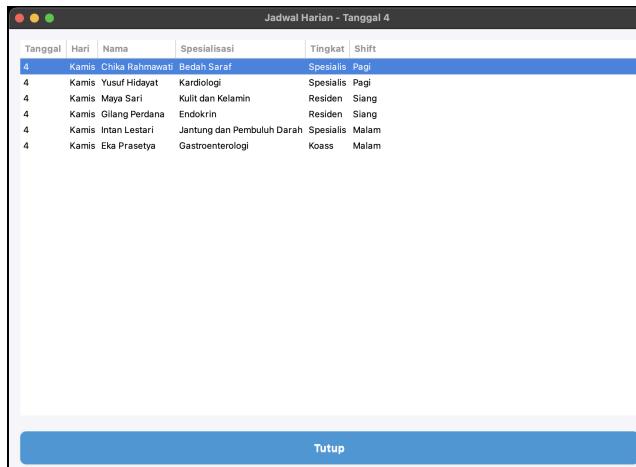
Tabel 6. Hasil studi kasus operasi program

No	Tampilan UI	Penjelasan
1	 <p>Sistem Manajemen Jadwal Dokter Rumah Sakit Sehat Selalu</p> <ul style="list-style-type: none"> Manajemen Dokter Penjadwalan Otomatis Lihat Jadwal Dokter Laporan Kinerja Dokter Keluar Program <p>Selamat datang di Sistem Manajemen Jadwal Dokter!</p>	Program utama
2	 <p>Manajemen Dokter</p> <ul style="list-style-type: none"> Tambah Dokter Baru Hapus Dokter Caril Dokter Tampilkan Semua Dokter Batalkan Aktivitas Terakhir Tampilkan Log Aktivitas Kembali ke Menu Utama <p>Menu Manajemen Dokter.</p>	Menu Manajemen Dokter

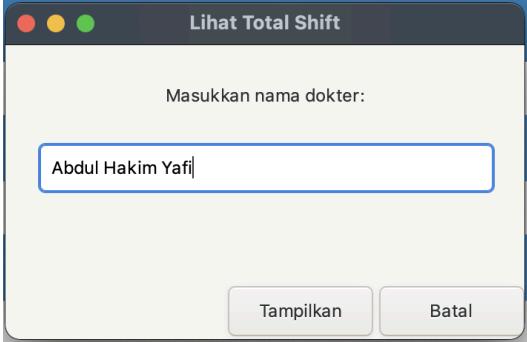
3	 <p>Tambah Dokter Baru</p> <p>Nama: Mazaya Bari Nafiah</p> <p>Bidang: Bedah Toraks</p> <p>Tingkat: Konsulen</p> <p>Maks. Shift/Minggu: 6</p> <p>Preferensi Shift: Siang</p> <p>Preferensi Waktu: Awal Bulan</p> <p>Simpan Batal</p>	Menu Tambah Dokter Baru																																																																																																																																																																		
4	 <p>Hasil Pencarian Dokter</p> <table border="1"> <thead> <tr> <th>Nama</th> <th>Bidang</th> <th>Tingkat</th> <th>Maks. Shift/Minggu</th> <th>Pref. Shift</th> <th>Pref. Waktu</th> </tr> </thead> <tbody> <tr><td>Mazaya Bari Nafiah</td><td>Bedah Toraks</td><td>Konsulen 6</td><td>Siang</td><td>Awal Bulan</td><td></td></tr> <tr><td>Rafif Farras</td><td>Bedah Anak</td><td>Konsulen 6</td><td>Pagi</td><td>Campur</td><td></td></tr> <tr><td>Lutfi</td><td>Mata</td><td>Konsulen 5</td><td>Siang</td><td>Campur</td><td></td></tr> <tr><td>Abdul Hakim Yafi</td><td>Obgyn</td><td>Konsulen 4</td><td>Pagi</td><td>Akhir Bulan</td><td></td></tr> <tr><td>Olivia Hapsari</td><td>Kandungan dan Ginekologi</td><td>Spesialis 4</td><td>Siang</td><td>Akhir Bulan</td><td></td></tr> <tr><td>Niko Aditya</td><td>Mata</td><td>Konsulen 2</td><td>Pagi</td><td>Awal Bulan</td><td></td></tr> <tr><td>Maya Sari</td><td>Kulit dan Kelamin</td><td>Residen 5</td><td>Siang</td><td>Akhir Bulan</td><td></td></tr> <tr><td>Lukman Hakim</td><td>THT</td><td>Spesialis 4</td><td>Malam</td><td>Awal Bulan</td><td></td></tr> <tr><td>Kania Melani</td><td>Anak</td><td>Koass 6</td><td>Pagi</td><td>Campur</td><td></td></tr> <tr><td>Joko Prabowo</td><td>Paru</td><td>Residen 5</td><td>Siang</td><td>Campur</td><td></td></tr> <tr><td>Intan Lestari</td><td>Jantung dan Pembuluh Darah</td><td>Spesialis 4</td><td>Malam</td><td>Awal Bulan</td><td></td></tr> <tr><td>Hilda Nuraini</td><td>Ginjal dan Hipertensi</td><td>Konsulen 2</td><td>Pagi</td><td>Akhir Bulan</td><td></td></tr> <tr><td>Gilang Perdana</td><td>Endokrin</td><td>Residen 5</td><td>Siang</td><td>Akhir Bulan</td><td></td></tr> <tr><td>Fatma Dewi</td><td>Alergi dan Imunologi</td><td>Spesialis 4</td><td>Pagi</td><td>Awal Bulan</td><td></td></tr> <tr><td>Eka Prasetya</td><td>Gastroenterologi</td><td>Koass 6</td><td>Malam</td><td>Campur</td><td></td></tr> <tr><td>Dodi Wirawa</td><td>Penyakit Dalam</td><td>Konsulen 2</td><td>Siang</td><td>Akhir Bulan</td><td></td></tr> <tr><td>Chika Rahmawati</td><td>Bedah Saraf</td><td>Spesialis 4</td><td>Pagi</td><td>Awal Bulan</td><td></td></tr> <tr><td>Budi Santoso</td><td>Bedah Onkologi</td><td>Residen 5</td><td>Siang</td><td>Akhir Bulan</td><td></td></tr> <tr><td>Andi Kurniawan</td><td>Bedah Umum</td><td>Koass 6</td><td>Pagi</td><td>Campur</td><td></td></tr> <tr><td>Rani Permatasari</td><td>Gigi</td><td>Spesialis 3</td><td>Siang</td><td>Campur</td><td></td></tr> <tr><td>Bayu Herlambang</td><td>Anak</td><td>Konsulen 2</td><td>Pagi</td><td>Akhir Bulan</td><td></td></tr> <tr><td>Salsa Maharani</td><td>Neurologi</td><td>Residen 4</td><td>Malam</td><td>Akhir Bulan</td><td></td></tr> <tr><td>Yusuf Hidayat</td><td>Kardiologi</td><td>Spesialis 5</td><td>Pagi</td><td>Awal Bulan</td><td></td></tr> <tr><td>Laras Sari</td><td>Paru</td><td>Konsulen 3</td><td>Siang</td><td>Akhir Bulan</td><td></td></tr> <tr><td>Dewi Anggraini</td><td>Onkologi Medis</td><td>Residen 4</td><td>Malam</td><td>Campur</td><td></td></tr> <tr><td>Fajar Nugraha</td><td>Bedah Plastik</td><td>Spesialis 4</td><td>Pagi</td><td>Akhir Bulan</td><td></td></tr> </tbody> </table> <p>Tutup</p>	Nama	Bidang	Tingkat	Maks. Shift/Minggu	Pref. Shift	Pref. Waktu	Mazaya Bari Nafiah	Bedah Toraks	Konsulen 6	Siang	Awal Bulan		Rafif Farras	Bedah Anak	Konsulen 6	Pagi	Campur		Lutfi	Mata	Konsulen 5	Siang	Campur		Abdul Hakim Yafi	Obgyn	Konsulen 4	Pagi	Akhir Bulan		Olivia Hapsari	Kandungan dan Ginekologi	Spesialis 4	Siang	Akhir Bulan		Niko Aditya	Mata	Konsulen 2	Pagi	Awal Bulan		Maya Sari	Kulit dan Kelamin	Residen 5	Siang	Akhir Bulan		Lukman Hakim	THT	Spesialis 4	Malam	Awal Bulan		Kania Melani	Anak	Koass 6	Pagi	Campur		Joko Prabowo	Paru	Residen 5	Siang	Campur		Intan Lestari	Jantung dan Pembuluh Darah	Spesialis 4	Malam	Awal Bulan		Hilda Nuraini	Ginjal dan Hipertensi	Konsulen 2	Pagi	Akhir Bulan		Gilang Perdana	Endokrin	Residen 5	Siang	Akhir Bulan		Fatma Dewi	Alergi dan Imunologi	Spesialis 4	Pagi	Awal Bulan		Eka Prasetya	Gastroenterologi	Koass 6	Malam	Campur		Dodi Wirawa	Penyakit Dalam	Konsulen 2	Siang	Akhir Bulan		Chika Rahmawati	Bedah Saraf	Spesialis 4	Pagi	Awal Bulan		Budi Santoso	Bedah Onkologi	Residen 5	Siang	Akhir Bulan		Andi Kurniawan	Bedah Umum	Koass 6	Pagi	Campur		Rani Permatasari	Gigi	Spesialis 3	Siang	Campur		Bayu Herlambang	Anak	Konsulen 2	Pagi	Akhir Bulan		Salsa Maharani	Neurologi	Residen 4	Malam	Akhir Bulan		Yusuf Hidayat	Kardiologi	Spesialis 5	Pagi	Awal Bulan		Laras Sari	Paru	Konsulen 3	Siang	Akhir Bulan		Dewi Anggraini	Onkologi Medis	Residen 4	Malam	Campur		Fajar Nugraha	Bedah Plastik	Spesialis 4	Pagi	Akhir Bulan		Menu Tampilkan Semua Dokter
Nama	Bidang	Tingkat	Maks. Shift/Minggu	Pref. Shift	Pref. Waktu																																																																																																																																																															
Mazaya Bari Nafiah	Bedah Toraks	Konsulen 6	Siang	Awal Bulan																																																																																																																																																																
Rafif Farras	Bedah Anak	Konsulen 6	Pagi	Campur																																																																																																																																																																
Lutfi	Mata	Konsulen 5	Siang	Campur																																																																																																																																																																
Abdul Hakim Yafi	Obgyn	Konsulen 4	Pagi	Akhir Bulan																																																																																																																																																																
Olivia Hapsari	Kandungan dan Ginekologi	Spesialis 4	Siang	Akhir Bulan																																																																																																																																																																
Niko Aditya	Mata	Konsulen 2	Pagi	Awal Bulan																																																																																																																																																																
Maya Sari	Kulit dan Kelamin	Residen 5	Siang	Akhir Bulan																																																																																																																																																																
Lukman Hakim	THT	Spesialis 4	Malam	Awal Bulan																																																																																																																																																																
Kania Melani	Anak	Koass 6	Pagi	Campur																																																																																																																																																																
Joko Prabowo	Paru	Residen 5	Siang	Campur																																																																																																																																																																
Intan Lestari	Jantung dan Pembuluh Darah	Spesialis 4	Malam	Awal Bulan																																																																																																																																																																
Hilda Nuraini	Ginjal dan Hipertensi	Konsulen 2	Pagi	Akhir Bulan																																																																																																																																																																
Gilang Perdana	Endokrin	Residen 5	Siang	Akhir Bulan																																																																																																																																																																
Fatma Dewi	Alergi dan Imunologi	Spesialis 4	Pagi	Awal Bulan																																																																																																																																																																
Eka Prasetya	Gastroenterologi	Koass 6	Malam	Campur																																																																																																																																																																
Dodi Wirawa	Penyakit Dalam	Konsulen 2	Siang	Akhir Bulan																																																																																																																																																																
Chika Rahmawati	Bedah Saraf	Spesialis 4	Pagi	Awal Bulan																																																																																																																																																																
Budi Santoso	Bedah Onkologi	Residen 5	Siang	Akhir Bulan																																																																																																																																																																
Andi Kurniawan	Bedah Umum	Koass 6	Pagi	Campur																																																																																																																																																																
Rani Permatasari	Gigi	Spesialis 3	Siang	Campur																																																																																																																																																																
Bayu Herlambang	Anak	Konsulen 2	Pagi	Akhir Bulan																																																																																																																																																																
Salsa Maharani	Neurologi	Residen 4	Malam	Akhir Bulan																																																																																																																																																																
Yusuf Hidayat	Kardiologi	Spesialis 5	Pagi	Awal Bulan																																																																																																																																																																
Laras Sari	Paru	Konsulen 3	Siang	Akhir Bulan																																																																																																																																																																
Dewi Anggraini	Onkologi Medis	Residen 4	Malam	Campur																																																																																																																																																																
Fajar Nugraha	Bedah Plastik	Spesialis 4	Pagi	Akhir Bulan																																																																																																																																																																
5	 <p>Cari Dokter</p> <p>Masukkan kata kunci pencarian:</p> <p>Mazaya Bari Nafiah</p> <p>Pilih kriteria pencarian:</p> <p>Nama</p> <p>Cari Batal</p>	Menu Cari Dokter																																																																																																																																																																		

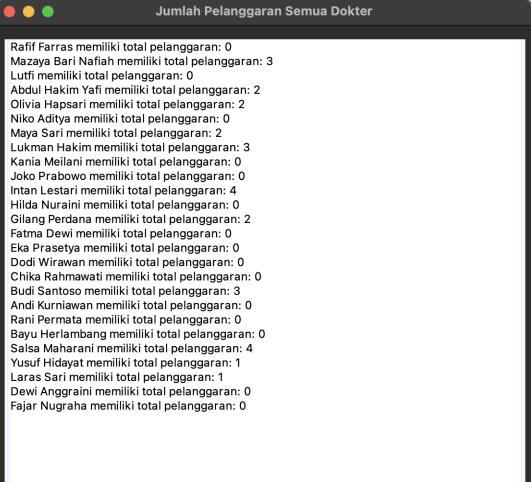
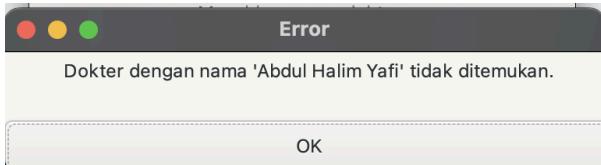
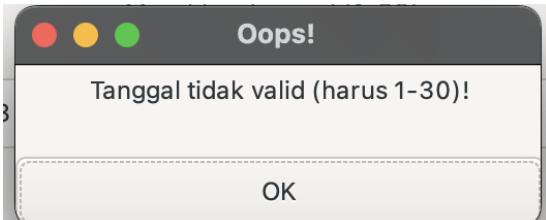
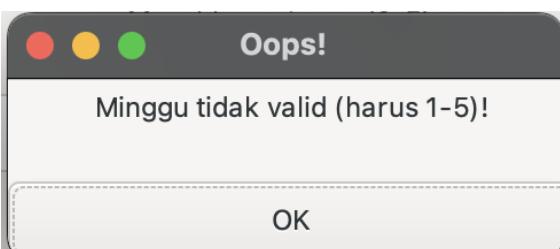
6	 <p>Hasil Pencarian Dokter</p> <table border="1"> <thead> <tr> <th>Nama</th><th>Bidang</th><th>Tingkat</th><th>Maks. Shift/Minggu</th><th>Pref. Shift</th><th>Pref. Waktu</th></tr> </thead> <tbody> <tr> <td>Mazaya Bari Nafiah</td><td>Bedah Toraks</td><td>Konsulen: 6</td><td>Siang</td><td></td><td>Awal Bulan</td></tr> </tbody> </table> <p>Tutup</p>	Nama	Bidang	Tingkat	Maks. Shift/Minggu	Pref. Shift	Pref. Waktu	Mazaya Bari Nafiah	Bedah Toraks	Konsulen: 6	Siang		Awal Bulan	Hasil Pencarian Dokter
Nama	Bidang	Tingkat	Maks. Shift/Minggu	Pref. Shift	Pref. Waktu									
Mazaya Bari Nafiah	Bedah Toraks	Konsulen: 6	Siang		Awal Bulan									
7	 <p>Hapus Dokter</p> <p>Masukkan nama dokter yang ingin dihapus:</p> <input type="text" value="Rafif Farris"/> <p>Hapus Batal</p>	Menu Hapus Dokter												
8	 <p>Log Aktivitas</p> <p>1. Hapus dokter: Rafif Farris 2. Tambah dokter: Mazaya Bari Nafiah</p> <p>Tutup</p>	Menu Log Aktivitas												
9	 <p>Batalkan Aktivitas Terakhir</p> <p>Batalkan penghapusan dokter: Mazaya Bari Nafiah?</p> <p>Batalkan Kembali</p>	Menu Batalkan Aktivitas Terakhir												

10	<p style="text-align: center;">Penjadwalan Otomatis</p> <p>Klik tombol di bawah untuk menghasilkan jadwal otomatis berdasarkan data dokter.</p> <p style="text-align: center;">Generate Jadwal Bulanan</p> <p style="text-align: center;">Kembali ke Menu Utama</p> <p style="text-align: center;">Menu Penjadwalan Otomatis.</p>	
11	 <p style="text-align: center;">Sukses</p> <p>Penjadwalan berhasil dibuat dan disimpan!</p> <p style="text-align: center;">OK</p>	Output Generate Jadwal Bulanan
12	<p style="text-align: center;">Lihat Jadwal</p> <p style="text-align: center;">Tampilkan Jadwal Harian</p> <p style="text-align: center;">Tampilkan Jadwal Mingguan</p> <p style="text-align: center;">Tampilkan Jadwal Bulanan</p> <p style="text-align: center;">Kembali ke Menu Utama</p> <p style="text-align: center;">Menampilkan jadwal bulanan.</p>	Menu Lihat Jadwal

13	 <p>Jadwal Bulanan</p> <table border="1"> <thead> <tr> <th>Tanggal</th> <th>Hari</th> <th>Nama</th> <th>Spesialisasi</th> <th>Tingkat</th> <th>Shift</th> </tr> </thead> <tbody> <tr><td>1</td><td>Senin</td><td>Raffi Farras</td><td>Bedah Anak</td><td>Konsulen</td><td>Pagi</td></tr> <tr><td>1</td><td>Senin</td><td>Yusuf Hidayat</td><td>Kardiologi</td><td>Spesialis</td><td>Pagi</td></tr> <tr><td>1</td><td>Senin</td><td>Mazaya Bari Nafiah</td><td>Bedah Toraks</td><td>Konsulen</td><td>Siang</td></tr> <tr><td>1</td><td>Senin</td><td>Lutfi</td><td>Mata</td><td>Konsulen</td><td>Siang</td></tr> <tr><td>1</td><td>Senin</td><td>Eka Prasetya</td><td>Gastroenterologi</td><td>Koass</td><td>Malam</td></tr> <tr><td>1</td><td>Senin</td><td>Lukman Hakim</td><td>THT</td><td>Spesialis</td><td>Malam</td></tr> <tr><td>2</td><td>Selasa</td><td>Kania Meilani</td><td>Anak</td><td>Koass</td><td>Pagi</td></tr> <tr><td>2</td><td>Selasa</td><td>Andi Kurniawan</td><td>Bedah Umum</td><td>Koass</td><td>Pagi</td></tr> <tr><td>2</td><td>Selasa</td><td>Mazaya Bari Nafiah</td><td>Bedah Toraks</td><td>Konsulen</td><td>Siang</td></tr> <tr><td>2</td><td>Selasa</td><td>Joko Prabowo</td><td>Paru</td><td>Residen</td><td>Siang</td></tr> <tr><td>2</td><td>Selasa</td><td>Intan Lestari</td><td>Jantung dan Pembuluh Darah</td><td>Spesialis</td><td>Malam</td></tr> <tr><td>2</td><td>Selasa</td><td>Eka Prasetya</td><td>Gastroenterologi</td><td>Koass</td><td>Malam</td></tr> <tr><td>3</td><td>Rabu</td><td>Raffi Farras</td><td>Bedah Anak</td><td>Konsulen</td><td>Pagi</td></tr> <tr><td>3</td><td>Rabu</td><td>Fatma Dewi</td><td>Alergi dan Imunologi</td><td>Spesialis</td><td>Pagi</td></tr> <tr><td>3</td><td>Rabu</td><td>Mazaya Bari Nafiah</td><td>Bedah Toraks</td><td>Konsulen</td><td>Siang</td></tr> <tr><td>3</td><td>Rabu</td><td>Lutfi</td><td>Mata</td><td>Konsulen</td><td>Siang</td></tr> <tr><td>3</td><td>Rabu</td><td>Dewi Anggraini</td><td>Oncokologi Medis</td><td>Residen</td><td>Malam</td></tr> <tr><td>3</td><td>Rabu</td><td>Lukman Hakim</td><td>THT</td><td>Spesialis</td><td>Malam</td></tr> <tr><td>4</td><td>Kamis</td><td>Chika Rahmawati</td><td>Bedah Saraf</td><td>Spesialis</td><td>Pagi</td></tr> <tr><td>4</td><td>Kamis</td><td>Yusuf Hidayat</td><td>Kardiologi</td><td>Spesialis</td><td>Pagi</td></tr> <tr><td>4</td><td>Kamis</td><td>Maya Sari</td><td>Kulit dan Kelamin</td><td>Residen</td><td>Siang</td></tr> <tr><td>4</td><td>Kamis</td><td>Gilang Perdana</td><td>Endokrin</td><td>Residen</td><td>Siang</td></tr> <tr><td>4</td><td>Kamis</td><td>Intan Lestari</td><td>Jantung dan Pembuluh Darah</td><td>Spesialis</td><td>Malam</td></tr> <tr><td>4</td><td>Kamis</td><td>Eka Prasetya</td><td>Gastroenterologi</td><td>Koass</td><td>Malam</td></tr> <tr><td>5</td><td>Jumat</td><td>Kania Meilani</td><td>Anak</td><td>Koass</td><td>Pagi</td></tr> </tbody> </table> <p style="text-align: center;">Tutup</p>	Tanggal	Hari	Nama	Spesialisasi	Tingkat	Shift	1	Senin	Raffi Farras	Bedah Anak	Konsulen	Pagi	1	Senin	Yusuf Hidayat	Kardiologi	Spesialis	Pagi	1	Senin	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang	1	Senin	Lutfi	Mata	Konsulen	Siang	1	Senin	Eka Prasetya	Gastroenterologi	Koass	Malam	1	Senin	Lukman Hakim	THT	Spesialis	Malam	2	Selasa	Kania Meilani	Anak	Koass	Pagi	2	Selasa	Andi Kurniawan	Bedah Umum	Koass	Pagi	2	Selasa	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang	2	Selasa	Joko Prabowo	Paru	Residen	Siang	2	Selasa	Intan Lestari	Jantung dan Pembuluh Darah	Spesialis	Malam	2	Selasa	Eka Prasetya	Gastroenterologi	Koass	Malam	3	Rabu	Raffi Farras	Bedah Anak	Konsulen	Pagi	3	Rabu	Fatma Dewi	Alergi dan Imunologi	Spesialis	Pagi	3	Rabu	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang	3	Rabu	Lutfi	Mata	Konsulen	Siang	3	Rabu	Dewi Anggraini	Oncokologi Medis	Residen	Malam	3	Rabu	Lukman Hakim	THT	Spesialis	Malam	4	Kamis	Chika Rahmawati	Bedah Saraf	Spesialis	Pagi	4	Kamis	Yusuf Hidayat	Kardiologi	Spesialis	Pagi	4	Kamis	Maya Sari	Kulit dan Kelamin	Residen	Siang	4	Kamis	Gilang Perdana	Endokrin	Residen	Siang	4	Kamis	Intan Lestari	Jantung dan Pembuluh Darah	Spesialis	Malam	4	Kamis	Eka Prasetya	Gastroenterologi	Koass	Malam	5	Jumat	Kania Meilani	Anak	Koass	Pagi	Lihat Jadwal Bulanan
Tanggal	Hari	Nama	Spesialisasi	Tingkat	Shift																																																																																																																																																									
1	Senin	Raffi Farras	Bedah Anak	Konsulen	Pagi																																																																																																																																																									
1	Senin	Yusuf Hidayat	Kardiologi	Spesialis	Pagi																																																																																																																																																									
1	Senin	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang																																																																																																																																																									
1	Senin	Lutfi	Mata	Konsulen	Siang																																																																																																																																																									
1	Senin	Eka Prasetya	Gastroenterologi	Koass	Malam																																																																																																																																																									
1	Senin	Lukman Hakim	THT	Spesialis	Malam																																																																																																																																																									
2	Selasa	Kania Meilani	Anak	Koass	Pagi																																																																																																																																																									
2	Selasa	Andi Kurniawan	Bedah Umum	Koass	Pagi																																																																																																																																																									
2	Selasa	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang																																																																																																																																																									
2	Selasa	Joko Prabowo	Paru	Residen	Siang																																																																																																																																																									
2	Selasa	Intan Lestari	Jantung dan Pembuluh Darah	Spesialis	Malam																																																																																																																																																									
2	Selasa	Eka Prasetya	Gastroenterologi	Koass	Malam																																																																																																																																																									
3	Rabu	Raffi Farras	Bedah Anak	Konsulen	Pagi																																																																																																																																																									
3	Rabu	Fatma Dewi	Alergi dan Imunologi	Spesialis	Pagi																																																																																																																																																									
3	Rabu	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang																																																																																																																																																									
3	Rabu	Lutfi	Mata	Konsulen	Siang																																																																																																																																																									
3	Rabu	Dewi Anggraini	Oncokologi Medis	Residen	Malam																																																																																																																																																									
3	Rabu	Lukman Hakim	THT	Spesialis	Malam																																																																																																																																																									
4	Kamis	Chika Rahmawati	Bedah Saraf	Spesialis	Pagi																																																																																																																																																									
4	Kamis	Yusuf Hidayat	Kardiologi	Spesialis	Pagi																																																																																																																																																									
4	Kamis	Maya Sari	Kulit dan Kelamin	Residen	Siang																																																																																																																																																									
4	Kamis	Gilang Perdana	Endokrin	Residen	Siang																																																																																																																																																									
4	Kamis	Intan Lestari	Jantung dan Pembuluh Darah	Spesialis	Malam																																																																																																																																																									
4	Kamis	Eka Prasetya	Gastroenterologi	Koass	Malam																																																																																																																																																									
5	Jumat	Kania Meilani	Anak	Koass	Pagi																																																																																																																																																									
14	 <p>Pilih Tanggal</p> <p>Masukkan tanggal (1-30):</p> <p>4</p> <p style="text-align: center;">Tampilkan Batal</p>	Menu Lihat jadwal Harian																																																																																																																																																												
15	 <p>Jadwal Harian - Tanggal 4</p> <table border="1"> <thead> <tr> <th>Tanggal</th> <th>Hari</th> <th>Nama</th> <th>Spesialisasi</th> <th>Tingkat</th> <th>Shift</th> </tr> </thead> <tbody> <tr><td>4</td><td>Kamis</td><td>Chika Rahmawati</td><td>Bedah Saraf</td><td>Spesialis</td><td>Pagi</td></tr> <tr><td>4</td><td>Kamis</td><td>Yusuf Hidayat</td><td>Kardiologi</td><td>Spesialis</td><td>Pagi</td></tr> <tr><td>4</td><td>Kamis</td><td>Maya Sari</td><td>Kulit dan Kelamin</td><td>Residen</td><td>Siang</td></tr> <tr><td>4</td><td>Kamis</td><td>Gilang Perdana</td><td>Endokrin</td><td>Residen</td><td>Siang</td></tr> <tr><td>4</td><td>Kamis</td><td>Intan Lestari</td><td>Jantung dan Pembuluh Darah</td><td>Spesialis</td><td>Malam</td></tr> <tr><td>4</td><td>Kamis</td><td>Eka Prasetya</td><td>Gastroenterologi</td><td>Koass</td><td>Malam</td></tr> </tbody> </table> <p style="text-align: center;">Tutup</p>	Tanggal	Hari	Nama	Spesialisasi	Tingkat	Shift	4	Kamis	Chika Rahmawati	Bedah Saraf	Spesialis	Pagi	4	Kamis	Yusuf Hidayat	Kardiologi	Spesialis	Pagi	4	Kamis	Maya Sari	Kulit dan Kelamin	Residen	Siang	4	Kamis	Gilang Perdana	Endokrin	Residen	Siang	4	Kamis	Intan Lestari	Jantung dan Pembuluh Darah	Spesialis	Malam	4	Kamis	Eka Prasetya	Gastroenterologi	Koass	Malam	Output Lihat Jadwal Harian																																																																																																																		
Tanggal	Hari	Nama	Spesialisasi	Tingkat	Shift																																																																																																																																																									
4	Kamis	Chika Rahmawati	Bedah Saraf	Spesialis	Pagi																																																																																																																																																									
4	Kamis	Yusuf Hidayat	Kardiologi	Spesialis	Pagi																																																																																																																																																									
4	Kamis	Maya Sari	Kulit dan Kelamin	Residen	Siang																																																																																																																																																									
4	Kamis	Gilang Perdana	Endokrin	Residen	Siang																																																																																																																																																									
4	Kamis	Intan Lestari	Jantung dan Pembuluh Darah	Spesialis	Malam																																																																																																																																																									
4	Kamis	Eka Prasetya	Gastroenterologi	Koass	Malam																																																																																																																																																									

16	<p>Pilih Minggu</p> <p>Masukkan minggu (1-5):</p> <p>Tampilkan Batal</p>	Menu Lihat Jadwal Mingguan																																																																																																																																																
17	<p>Jadwal Mingguan - Minggu 2</p> <table border="1"> <thead> <tr> <th>Tanggal</th> <th>Hari</th> <th>Nama</th> <th>Spesialisasi</th> <th>Tingkat</th> <th>Shift</th> </tr> </thead> <tbody> <tr><td>8</td><td>Senin</td><td>Rafif Farris</td><td>Bedah Anak</td><td>Konsulen</td><td>Pagi</td></tr> <tr><td>8</td><td>Senin</td><td>Yusuf Hidayat</td><td>Kardiolog</td><td>Spesialis</td><td>Pagi</td></tr> <tr><td>8</td><td>Senin</td><td>Mazaya Bari Nafiah</td><td>Bedah Toraks</td><td>Konsulen</td><td>Siang</td></tr> <tr><td>8</td><td>Senin</td><td>Lutfi</td><td>Mata</td><td>Konsulen</td><td>Siang</td></tr> <tr><td>8</td><td>Senin</td><td>Eka Prasetya</td><td>Gastroenterologi</td><td>Koassis</td><td>Malam</td></tr> <tr><td>8</td><td>Senin</td><td>Lukman Hakim</td><td>THT</td><td>Spesialis</td><td>Malam</td></tr> <tr><td>9</td><td>Selasa</td><td>Kania Melani</td><td>Anak</td><td>Koassis</td><td>Pagi</td></tr> <tr><td>9</td><td>Selasa</td><td>Andi Kurniawan</td><td>Bedah Umum</td><td>Koassis</td><td>Pagi</td></tr> <tr><td>9</td><td>Selasa</td><td>Mazaya Bari Nafiah</td><td>Bedah Toraks</td><td>Konsulen</td><td>Siang</td></tr> <tr><td>9</td><td>Selasa</td><td>Joko Prabowo</td><td>Paru</td><td>Residen</td><td>Siang</td></tr> <tr><td>9</td><td>Selasa</td><td>Intan Lestari</td><td>Jantung dan Pembuluh Darah</td><td>Spesialis</td><td>Malam</td></tr> <tr><td>9</td><td>Selasa</td><td>Eka Prasetya</td><td>Gastroenterologi</td><td>Koassis</td><td>Malam</td></tr> <tr><td>10</td><td>Rabu</td><td>Rafif Farris</td><td>Bedah Anak</td><td>Konsulen</td><td>Pagi</td></tr> <tr><td>10</td><td>Rabu</td><td>Fatma Dewi</td><td>Alergi dan Imunologi</td><td>Spesialis</td><td>Pagi</td></tr> <tr><td>10</td><td>Rabu</td><td>Mazaya Bari Nafiah</td><td>Bedah Toraks</td><td>Konsulen</td><td>Siang</td></tr> <tr><td>10</td><td>Rabu</td><td>Lutfi</td><td>Mata</td><td>Konsulen</td><td>Siang</td></tr> <tr><td>10</td><td>Rabu</td><td>Dewi Anggraini</td><td>Onkologi Medis</td><td>Residen</td><td>Malam</td></tr> <tr><td>10</td><td>Rabu</td><td>Lukman Hakim</td><td>THT</td><td>Spesialis</td><td>Malam</td></tr> <tr><td>11</td><td>Kamis</td><td>Chika Rahmawati</td><td>Bedah Saraf</td><td>Spesialis</td><td>Pagi</td></tr> <tr><td>11</td><td>Kamis</td><td>Yusuf Hidayat</td><td>Kardiolog</td><td>Spesialis</td><td>Pagi</td></tr> <tr><td>11</td><td>Kamis</td><td>Maya Sari</td><td>Kulit dan Kelamin</td><td>Residen</td><td>Siang</td></tr> <tr><td>11</td><td>Kamis</td><td>Gilang Perdana</td><td>Endokrin</td><td>Residen</td><td>Siang</td></tr> <tr><td>11</td><td>Kamis</td><td>Intan Lestari</td><td>Jantung dan Pembuluh Darah</td><td>Spesialis</td><td>Malam</td></tr> </tbody> </table> <p>Tutup</p>	Tanggal	Hari	Nama	Spesialisasi	Tingkat	Shift	8	Senin	Rafif Farris	Bedah Anak	Konsulen	Pagi	8	Senin	Yusuf Hidayat	Kardiolog	Spesialis	Pagi	8	Senin	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang	8	Senin	Lutfi	Mata	Konsulen	Siang	8	Senin	Eka Prasetya	Gastroenterologi	Koassis	Malam	8	Senin	Lukman Hakim	THT	Spesialis	Malam	9	Selasa	Kania Melani	Anak	Koassis	Pagi	9	Selasa	Andi Kurniawan	Bedah Umum	Koassis	Pagi	9	Selasa	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang	9	Selasa	Joko Prabowo	Paru	Residen	Siang	9	Selasa	Intan Lestari	Jantung dan Pembuluh Darah	Spesialis	Malam	9	Selasa	Eka Prasetya	Gastroenterologi	Koassis	Malam	10	Rabu	Rafif Farris	Bedah Anak	Konsulen	Pagi	10	Rabu	Fatma Dewi	Alergi dan Imunologi	Spesialis	Pagi	10	Rabu	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang	10	Rabu	Lutfi	Mata	Konsulen	Siang	10	Rabu	Dewi Anggraini	Onkologi Medis	Residen	Malam	10	Rabu	Lukman Hakim	THT	Spesialis	Malam	11	Kamis	Chika Rahmawati	Bedah Saraf	Spesialis	Pagi	11	Kamis	Yusuf Hidayat	Kardiolog	Spesialis	Pagi	11	Kamis	Maya Sari	Kulit dan Kelamin	Residen	Siang	11	Kamis	Gilang Perdana	Endokrin	Residen	Siang	11	Kamis	Intan Lestari	Jantung dan Pembuluh Darah	Spesialis	Malam	Output Lihat Jadwal Bulanan
Tanggal	Hari	Nama	Spesialisasi	Tingkat	Shift																																																																																																																																													
8	Senin	Rafif Farris	Bedah Anak	Konsulen	Pagi																																																																																																																																													
8	Senin	Yusuf Hidayat	Kardiolog	Spesialis	Pagi																																																																																																																																													
8	Senin	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang																																																																																																																																													
8	Senin	Lutfi	Mata	Konsulen	Siang																																																																																																																																													
8	Senin	Eka Prasetya	Gastroenterologi	Koassis	Malam																																																																																																																																													
8	Senin	Lukman Hakim	THT	Spesialis	Malam																																																																																																																																													
9	Selasa	Kania Melani	Anak	Koassis	Pagi																																																																																																																																													
9	Selasa	Andi Kurniawan	Bedah Umum	Koassis	Pagi																																																																																																																																													
9	Selasa	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang																																																																																																																																													
9	Selasa	Joko Prabowo	Paru	Residen	Siang																																																																																																																																													
9	Selasa	Intan Lestari	Jantung dan Pembuluh Darah	Spesialis	Malam																																																																																																																																													
9	Selasa	Eka Prasetya	Gastroenterologi	Koassis	Malam																																																																																																																																													
10	Rabu	Rafif Farris	Bedah Anak	Konsulen	Pagi																																																																																																																																													
10	Rabu	Fatma Dewi	Alergi dan Imunologi	Spesialis	Pagi																																																																																																																																													
10	Rabu	Mazaya Bari Nafiah	Bedah Toraks	Konsulen	Siang																																																																																																																																													
10	Rabu	Lutfi	Mata	Konsulen	Siang																																																																																																																																													
10	Rabu	Dewi Anggraini	Onkologi Medis	Residen	Malam																																																																																																																																													
10	Rabu	Lukman Hakim	THT	Spesialis	Malam																																																																																																																																													
11	Kamis	Chika Rahmawati	Bedah Saraf	Spesialis	Pagi																																																																																																																																													
11	Kamis	Yusuf Hidayat	Kardiolog	Spesialis	Pagi																																																																																																																																													
11	Kamis	Maya Sari	Kulit dan Kelamin	Residen	Siang																																																																																																																																													
11	Kamis	Gilang Perdana	Endokrin	Residen	Siang																																																																																																																																													
11	Kamis	Intan Lestari	Jantung dan Pembuluh Darah	Spesialis	Malam																																																																																																																																													
18	<p>Rumah Sakit Sehat Selalu</p> <p>Laporan Kinerja Dokter</p> <p>Generate Laporan Kinerja</p> <p>Lihat Total Shift Dokter</p> <p>Lihat Pelanggaran Jadwal</p> <p>Lihat Total Pelanggaran Semua Dokter</p> <p>Lihat Pelanggaran Per Dokter</p> <p>Kembali ke Menu Utama</p> <p>Menampilkan total pelanggaran dokter.</p>	Menu Laporan Kinerja Dokter																																																																																																																																																

19		Generate Laporan Kinerja Dokter
20		Lihat Total Shift Dokter
21		Output Total Shift Dokter
22	 <p>[4] Maya Sari - Waktu tidak sesuai preferensi (jadwal tanggal: 4, preferensi: Akhir bulan) [4] Intan Lestari - Waktu tidak sesuai preferensi (jadwal tanggal: 4, preferensi: Akhir bulan) [5] Budi Santoso - Waktu tidak sesuai preferensi (jadwal tanggal: 5, preferensi: Akhir bulan) [5] Salsa Maharani - Waktu tidak sesuai preferensi (jadwal tanggal: 5, preferensi: Akhir bulan) [7] Abdul Hakim Yafi - Waktu tidak sesuai preferensi (jadwal tanggal: 7, preferensi: Akhir bulan) [7] Olivia Hapsari - Waktu tidak sesuai preferensi (jadwal tanggal: 7, preferensi: Akhir bulan) [7] Salsa Maharani - Waktu tidak sesuai preferensi (jadwal tanggal: 7, preferensi: Akhir bulan) [10] Mazaya Bari Nafiah - Melebihi shift mingguan maksimum (6 per minggu) [10] Intan Lestari - Melebihi shift mingguan maksimum (6 per minggu) [11] Maya Sari - Waktu tidak sesuai preferensi (jadwal tanggal: 11, preferensi: Akhir bulan) [11] Gilang Perdana - Waktu tidak sesuai preferensi (jadwal tanggal: 11, preferensi: Akhir bulan) [11] Intan Lestari - Melebihi shift mingguan maksimum (4 per minggu) [11] Eka Prasetya - Melebihi shift mingguan maksimum (6 per minggu) [12] Budi Santoso - Waktu tidak sesuai preferensi (jadwal tanggal: 12, preferensi: Akhir bulan) [12] Salsa Maharani - Waktu tidak sesuai preferensi (jadwal tanggal: 12, preferensi: Akhir bulan) [13] Abdul Hakim Yafi - Waktu tidak sesuai preferensi (jadwal tanggal: 13, preferensi: Akhir bulan) [13] Mazaya Bari Nafiah - Melebihi shift mingguan maksimum (6 per minggu) [13] Olivia Hapsari - Waktu tidak sesuai preferensi (jadwal tanggal: 13, preferensi: Akhir bulan) [13] Lukman Hakim - Melebihi shift mingguan maksimum (4 per minggu) [13] Intan Lestari - Melebihi shift mingguan maksimum (4 per minggu) [14] Lutfi - Melebihi shift mingguan maksimum (5 per minggu) [14] Eka Prasetya - Melebihi shift mingguan maksimum (6 per minggu) [14] Salsa Maharani - Waktu tidak sesuai preferensi (jadwal tanggal: 14, preferensi: Akhir bulan)</p>	Output Lihat Pelanggaran Jadwal

23	 <pre>Rafif Farris memiliki total pelanggaran: 0 Mazaya Bari Nafiah memiliki total pelanggaran: 3 Lutfi memiliki total pelanggaran: 0 Abdul Hakim Yafi memiliki total pelanggaran: 2 Olivia Hapsari memiliki total pelanggaran: 2 Niko Aditya memiliki total pelanggaran: 0 Maya Sari memiliki total pelanggaran: 2 Lukman Hakim memiliki total pelanggaran: 3 Kania Melani memiliki total pelanggaran: 0 Joko Prabowo memiliki total pelanggaran: 0 Intan Lestari memiliki total pelanggaran: 4 Hilda Nuraini memiliki total pelanggaran: 0 Gilang Perdana memiliki total pelanggaran: 2 Fatma Dewi memiliki total pelanggaran: 0 Eka Prasetya memiliki total pelanggaran: 0 Dodi Wirawan memiliki total pelanggaran: 0 Chika Rahmawati memiliki total pelanggaran: 0 Budi Santoso memiliki total pelanggaran: 3 Andi Kurniawan memiliki total pelanggaran: 0 Rani Permata memiliki total pelanggaran: 0 Bayu Herlambang memiliki total pelanggaran: 0 Salsa Maharani memiliki total pelanggaran: 4 Yusuf Hidayat memiliki total pelanggaran: 1 Laras Sari memiliki total pelanggaran: 1 Dewi Anggraini memiliki total pelanggaran: 0 Fajar Nugraha memiliki total pelanggaran: 0</pre>	Output Total Pelanggaran Tiap Dokter
24	 <p>Total pelanggaran untuk Abdul Hakim Yafi: 2</p> <p>OK</p>	Output Pelanggaran 1 Dokter
25	 <p>Dokter dengan nama 'Abdul Halim Yafi' tidak ditemukan.</p> <p>OK</p>	Output Pelanggaran 1 Dokter apabila nama tidak valid
26	 <p>Tanggal tidak valid (harus 1-30)!</p> <p>OK</p>	Input Cek Jadwal Harian Tidak Valid
27	 <p>Minggu tidak valid (harus 1-5)!</p> <p>OK</p>	Input Cek Jadwal Mingguan Tidak Valid

28		Input Pencarian Dokter tidak Valid
----	--	---------------------------------------

Berdasarkan Tabel 1, kasus-kasus di bagian sebelumnya telah berhasil diselesaikan oleh program. Tabel 1 menampilkan hasil tangkapan layar operasi program untuk menyelesaikan studi kasus secara berurutan. Berikutnya akan ditinjau analisis studi kasus sesuai dengan poin-poin kasus yang dideskripsikan sebelumnya. Gambar pertama adalah tampilan awal Sistem Manajemen Jadwal Dokter Rumah Sakit Sehat Selalu.

6.1. Analisis hasil studi kasus

1. Penambahan dokter baru: Pengguna memilih menu manajemen dokter lalu menu tambah dokter. ketika pengguna memilih fungsi tersebut, pengguna menginputkan data secara manual ke textbox pada window yang disediakan. Penambahan dokter baru berhasil dilakukan dan ditunjukkan oleh nomor 1-3.
2. Menampilkan data semua dokter: setelah menambah dokter, pengguna memilih menu tampilkan semua dokter. Window baru kemudian muncul menampilkan tabel data semua dokter yang ada. Hal ini ditunjukkan oleh nomor 4.
3. Mencari dokter berdasarkan nama: Pengguna memilih menu cari dokter dan memilih kriteria pencarian yang ingin dilakukan. Pada kasus ini pengguna memilih kriteria nama. Pengguna kemudian menuliskan nama dokter yang ingin dicari pada window yang muncul (nomor 5). Setelah klik cari, daftar dokter yang memenuhi kriteria muncul pada window yang baru (nomor 6).
4. Menghapus dokter: Pengguna memilih menu hapus dokter dan memasukkan nama dokter terkait ke dalam textbox pada window yang muncul. Setelah klik hapus, dokter berhasil dihapus (nomor 7).

5. Cek log aktivitas: Pengguna memilih menu tampilkan log aktivitas. Window baru akan muncul menampilkan daftar aktivitas dan deskripsi aktivitas yang dilakukan di sistem (nomor 8).
6. Membatalkan aksi sebelumnya (undo): pengguna memilih menu batalkan aktivitas terakhir dan setelah mengonfirmasi pembatalan, maka aktivitas sebelumnya (penghapusan Rafif Farras) berhasil dibatalkan (nomor 9).
7. Generate jadwal: pengguna kembali ke menu utama dan memilih menu penjadwalan otomatis dan memilih menu generate jadwal bulanan (nomor 10-11). Jadwal pun berhasil dibuat.
8. Lihat jadwal bulanan, minggu ke-2, dan hari ke-4: Pengguna kembali ke menu utama dan memilih menu lihat jadwal. Pengguna kemudian memilih untuk melihat berdasarkan hari, bulan, atau minggu. Pengguna harus menginputkan hari atau minggu yang diinginkan pada window yang muncul setelahnya. Setelah memilih “tampilkan”, jadwal dokter dengan kriteria tersebut ditampilkan di window baru (nomor 12-17).
9. Generate laporan kinerja jadwal: Pengguna kembali ke menu utama lalu memilih menu laporan kinerja dokter. Kemudian pengguna memilih menu generate laporan kinerja (nomor 18). Laporan kinerja lalu selesai dibuat (nomor 19).
10. Lihat total shift dokter: Pengguna memilih menu lihat total shift dokter dan menginput nama dokter yang ingin dilihat total shiftnya dalam sebulan pada window yang muncul (nomor 20). Setelah memilih “tampilkan” window baru muncul dan menampilkan total shift untuk dokter yang dicari (nomor 21).
11. Lihat jumlah pelanggaran preferensi: Pengguna menggunakan fitur lihat jumlah pelanggaran jadwal semua dokter sehingga ditampilkan daftar semua dokter dan pelanggaran yang dialami (nomor 23). Kemudian pengguna menggunakan fitur Lihat Jumlah Pelanggaran Jadwal dokter (berdasarkan nama). Window baru akan muncul. Lalu pengguna memasukkan nama dokter Abdul Hakim Yafi ke dalam textbox. Ketika klik tampilkan, nama dokter Abdul Hakim Yafi dan jumlah pelanggaran yang ia alami ditampilkan pada window baru (nomor 24).
12. Lihat detail pelanggaran jadwal: pengguna memilih menu lihat pelanggaran jadwal. Lalu window baru akan muncul menampilkan semua daftar pelanggaran yang terjadi dan kepada dokter siapa pelanggaran terjadi (nomor 22).

13. Validasi input: Pengguna memasukkan nama yang tidak ada dalam database ketika fungsi yang digunakan memintanya. Ketika dieksekusi, program memberikan pesan bahwa dokter dengan nama tersebut tidak ditemukan (nomor 25 dan 28). Ketika memasukkan minggu dan tanggal yang tidak valid, program juga memberikan pesan bahwa minggu dan tanggal tidak valid (nomor 26 dan 27).
14. Menutup program: Pengguna kembali ke menu utama dan memilih menu keluar program.

6.2. Output yang dihasilkan

- Output tangkapan layar yang dihasilkan ditampilkan pada Tabel 1.
- File daftar_dokter.csv memiliki keadaan akhir sebagai berikut:

D: > Luthfii > School 2.0 > Sem 4 > PMP > transit > TUBES-PMP > data > daftar_dokter.csv > data
1 nama,bidang,tingkat,max_shift_per_minggu,preferensi_shift,preferensi_waktu
2 Mazaya Bari Nafiah,Bedah Toraks,Konsulen, <u>6</u> ,Siang,Awal Bulan
3 Fajar Nugraha,Bedah Plastik,Spesialis, <u>4</u> ,Pagi,Akhir Bulan
4 Dewi Anggraini,Onkologi Medis,Residen, <u>4</u> ,Malam,Campur
5 Laras Sari,Paru,Konsulen, <u>3</u> ,Siang,Akhir Bulan
6 Yusuf Hidayat,Kardiologi,Spesialis, <u>5</u> ,Pagi,Awal Bulan
7 Salsa Maharanji,Neurologi,Residen, <u>4</u> ,Malam,Akhir Bulan
8 Bayu Herlambang,Anak,Konsulen, <u>2</u> ,Pagi,Akhir Bulan
9 Rani Permata,gigi,Spesialis, <u>3</u> ,Siang,Campur
10 Andi Kurniawan,Bedah Umum,Koass, <u>6</u> ,Pagi,Campur
11 Budi Santoso,Bedah Onkologi,Residen, <u>5</u> ,Siang,Akhir Bulan
12 Chika Rahmawati,Bedah Saraf,Spesialis, <u>4</u> ,Pagi,Awal Bulan
13 Dodi Wirawan,Penyakit Dalam,Konsulen, <u>2</u> ,Siang,Akhir Bulan
14 Eka Prasetya,Gastroenterologi,Koass, <u>6</u> ,Malam,Campur
15 Fatma Dewi,Alergi dan Imunologi,Spesialis, <u>4</u> ,Pagi,Awal Bulan
16 Gilang Perdana,Endokrin,Residen, <u>5</u> ,Siang,Akhir Bulan
17 Hilda Nuraini,Ginjal dan Hipertensi,Konsulen, <u>2</u> ,Pagi,Akhir Bulan
18 Intan Lestari,Jantung dan Pembuluh Darah,Spesialis, <u>4</u> ,Malam,Awal Bulan
19 Joko Prabowo,Paru,Residen, <u>5</u> ,Siang,Campur
20 Kania Meilani,Anak,Koass, <u>6</u> ,Pagi,Campur
21 Lukman Hakim,IHT,Spesialis, <u>4</u> ,Malam,Awal Bulan
22 Maya Sari,Kulit dan Kelamin,Residen, <u>5</u> ,Siang,Akhir Bulan
23 Niko Aditya,Mata,Konsulen, <u>2</u> ,Pagi,Awal Bulan
24 Olivia Hapsari,Kandungan dan Ginekologi,Spesialis, <u>4</u> ,Siang,Akhir Bulan
25 Abdul Hakim Yafi,Obgyn,Konsulen, <u>4</u> ,Pagi,Akhir Bulan
26 Lutfi,Mata,Konsulen, <u>5</u> ,Siang,Campur
27 Rafif Farris,Bedah Anak,Konsulen, <u>6</u> ,Pagi,Campur
28

Gambar 53. Kondisi akhir daftar_dokter.csv

- File jadwal_dokter.csv memiliki hasil akhir sebagai berikut:

1	Janggal,Hari,Shift,Nama Dokter,Bidang,Tingkat
2	1,Senin,Pagi,Rafif Farras,Bedah Anak,Konsulen
3	1,Senin,Pagi,Yusuf Hidayat,Kardiologi,Spesialis
4	1,Senin,Siang,Mazaya Bari Nafiah,Bedah Toraks,Konsulen
5	1,Senin,Siang,Lutfi,Mata,Konsulen
6	1,Senin,Malam,Eka Prasetya,Gastroenterologi,Koass
7	1,Senin,Malam,Lukman Hakim,THT,Spesialis
8	2,Selasa,Pagi,Kania Meilani,Anak,Koass
9	2,Selasa,Pagi,Andi Kurniawan,Bedah Umum,Koass
10	2,Selasa,Siang,Mazaya Bari Nafiah,Bedah Toraks,Konsulen
11	2,Selasa,Siang,Joko Prabowo,Paru,Residen
12	2,Selasa,Malam,Intan Lestari,Jantung dan Pembuluh Darah,Spesialis
13	2,Selasa,Malam,Eka Prasetya,Gastroenterologi,Koass
14	3,Rabu,Pagi,Rafif Farras,Bedah Anak,Konsulen
15	3,Rabu,Pagi,Fatma Dewi,Alergi dan Imunologi,spesialis
16	3,Rabu,Siang,Mazaya Bari Nafiah,Bedah Toraks,Konsulen
17	3,Rabu,Siang,Lutfi,Mata,Konsulen
18	3,Rabu,Malam,Dewi Anggraini,Onkologi Medis,Residen
19	3,Rabu,Malam,Lukman Hakim,THT,Spesialis
20	4,Kamis,Pagi,Chika Rahmawati,Bedah Saraf,Spesialis
21	4,Kamis,Pagi,Yusuf Hidayat,Kardiologi,Spesialis
22	4,Kamis,Siang,Maya Sari,Kulit dan Kelamin,Residen
23	4,Kamis,Siang,Gilang Perdana,Endokrin,Residen
24	4,Kamis,Malam,Intan Lestari,Jantung dan Pembuluh Darah,Spesialis
25	4,Kamis,Malam,Eka Prasetya,Gastroenterologi,Koass
26	5,Jumat,Pagi,Kania Meilani,Anak,Koass
27	5,Jumat,Pagi,Andi Kurniawan,Bedah Umum,Koass
28	5,Jumat,Siang,Budi Santoso,Bedah Onkologi,Residen
29	5,Jumat,Siang,Joko Prabowo,Paru,Residen
30	5,Jumat,Malam,Salsa Maharani,Neurologi,Residen
31	5,Jumat,Malam,Dewi Anggraini,Onkologi Medis,Residen
32	6,Sabtu,Pagi,Rafif Farras,Bedah Anak,Konsulen
33	6,Sabtu,Pagi,Fatma Dewi,Alergi dan Imunologi,Spesialis

Gambar 54. Hasil file jadwal_dokter.csv

- File laporan_dokter.csv memiliki hasil akhir sebagai berikut:

1	Nama,Bidang,Tingkat,Total_Shift,Max_Per_Minggu,Pref_Shift,Pref_Waktu,Pelanggaran,Minggu_1,Minggu_2,Minggu_3,Minggu_4,Minggu_5,Shift_Pagi,Shift_Siang
2	Rafif Farras,Bedah Anak,Konsulen,13,6,Pagi,Campur,0,3,3,3,1,13,0,0
3	Mazaya Bari Nafiah,Bedah Toraks,Konsulen,12,6,Siang,Awal,3,4,4,2,2,0,0,12,0
4	Lutfi,Mata,Konsulen,11,5,Siang,Campur,0,3,3,2,2,1,0,11,0
5	Abdul Hakim Yafi,Obgyn,Konsulen,7,4,Pagi,Akhir,2,1,1,2,2,1,7,0,0
6	Olivia Hapsari,Kandungan dan Ginekologi,Spesialis,6,4,Siang,Akhir,2,1,1,2,2,0,0,6,0
7	Niko Aditya,Mata,Konsulen,0,2,Pagi,Awal,0,0,0,0,0,0,0,0,0
8	Maya Sari,Kulit dan Kelamin,Residen,7,5,Siang,Akhir,2,1,1,2,2,1,0,7,0
9	Lukman Hakim,THT,Spesialis,10,4,Malam,Awal,3,3,3,2,2,0,0,0,10
10	Kania Meilani,Anak,Koass,10,6,Pagi,Campur,0,2,2,3,2,1,10,0,0
11	Joko Prabowo,Paru,Residen,8,5,Siang,Campur,0,2,2,2,0,0,8,0
12	Intan Lestari,Jantung dan Pembuluh Darah,Spesialis,10,4,Malam,Awal,4,3,3,2,2,0,0,0,10
13	Hilda Nuraini,Ginjal dan Hipertensi,Konsulen,2,2,Pagi,Akhir,0,0,0,1,1,0,2,0,0
14	Gilang Perdana,Endokrin,Residen,7,5,Siang,Akhir,2,1,1,2,2,1,0,7,0
15	Fatma Dewi,Alergi dan Imunologi,Spesialis,4,4,Pagi,Awal,0,2,2,0,0,0,4,0,0
16	Eka Prasetya,Gastroenterologi,Koass,18,6,Malam,Campur,0,4,4,4,4,2,0,0,18
17	Dodi Wirawan,Penyakit Dalam,Konsulen,0,2,Siang,Akhir,0,0,0,0,0,0,0,0,0
18	Chika Rahmawati,Bedah Saraf,Spesialis,4,4,Pagi,Awal,0,2,2,0,0,0,4,0,0
19	Budi Santoso,Bedah Onkologi,Residen,7,5,Siang,Akhir,3,1,1,2,2,1,0,6,1
20	Andi Kurniawan,Bedah Umum,Koass,0,6,Pagi,Campur,0,2,2,2,2,1,0,9,0
21	Rani Permatasari,Gigi,Spesialis,2,3,Siang,Campur,0,1,1,0,0,0,0,2,0
22	Bayu Herlambang,Anak,Konsulen,1,2,Pagi,Akhir,0,0,0,0,1,0,1,0,0
23	Salsa Maharani,Neurologi,Residen,11,4,Malam,Akhir,4,2,2,3,3,1,0,0,11
24	Yusuf Hidayat,Kardiologi,Spesialis,6,5,Pagi,Awal,1,2,2,1,1,0,6,0,0
25	Laras Sari,Paru,Konsulen,2,3,Siang,Akhir,1,0,0,1,1,0,0,1,1
26	Dewi Anggraini,Onkologi Medis,Residen,9,4,Malam,Campur,0,2,2,2,2,1,0,0,9
27	Fajar Nugraha,Bedah Plastik,Spesialis,4,4,Pagi,Akhir,0,0,0,2,2,0,4,0,0

Gambar 55. Hasil file laporan_dokter.csv

6.3.Justifikasi pendekatan yang digunakan

- Antarmuka

Antarmuka berbasis modular dengan konsep tree yang cabangnya adalah menu-menu dari subprogram-subprogram yang dipilih menjadi pendekatan yang diambil karena penggunaannya yang cukup intuitif dan tampilannya yang lebih rapih serta enak dipandang. Hal ini karena konsep tree seperti ini sering digunakan oleh masyarakat modern saat ini. Contoh penggunaannya adalah pada file explorer di alat-alat elektronik, software-software multifungsi, dan perangkat digital lainnya.

- Format CSV

Format CSV untuk data-data yang digunakan sudah memenuhi dan bahkan melengkapi spesifikasi yang diminta pada dokumen arahan tugas besar. Tambahan pada format data yang digunakan dibandingkan spesifikasi adalah preferensi waktu (awal bulan, akhir bulan, atau campur), bidang kedokteran, tingkat dokter, dan pada laporan ditambahkan juga jumlah-jumlah shift per parameter nya (pada minggu kesekian, pada shift tertentu). Dengan adanya tambahan parameter ini, membuat program ini dapat menampilkan data dokter dengan lebih lengkap. Selain itu, dengan ditambahnya parameter preferensi dokter, akan membuat performa dokter menjadi lebih baik.

- Performa jadwal dan algoritma

Melihat dari file laporan_dokter.csv, performa program sudah cukup baik dan jumlah pelanggaran relatif minim. Algoritma greedy yang digunakan dalam fungsi pembuatan jadwal sudah cukup baik untuk mendapatkan hasil dengan pelanggaran yang minim. Meski belum berarti mendapatkan hasil yang global best, algoritma ini sudah mengaplikasikan optimasi dari algoritma pembuatan jadwal sehingga menghasilkan algoritma yang cepat dan optimal. Berikutnya, performa algoritma bisa diatur (tuning) lebih baik dengan mengubah pembobotan dari skor yang digunakan dalam menentukan penugasan dokter ke shift-shift yang tersedia.

6.4.Kesimpulan dan saran

6.4.1. Kesimpulan

1. Perancangan sistem pengelolaan data dokter dengan fitur tambah, hapus, tampil, dan baca data dari file eksternal (CSV) telah berhasil dilakukan.
2. Pengembangan algoritma penjadwalan otomatis dengan mempertimbangkan batasan kerja maksimal dan preferensi dokter telah berhasil dibuat dengan optimal menggunakan algoritma greedy.
3. Fitur visualisasi jadwal dan pelaporan jadwal shift dokter telah berhasil dibuat dengan visualisasi berbagai format dan pelaporan informasi pelanggaran preferensi dan distribusi beban kerja.
4. Sistem modular berhasil dibuat dengan antarmuka yang bisa digunakan melalui terminal powershell maupun dengan GUI berbasis GTK.

6.4.2. Saran untuk pengembangan lebih lanjut

1. Untuk aplikasi pada rumah sakit yang berbeda atau penggunaan yang berbeda, format file CSV dapat disesuaikan untuk memenuhi kebutuhan masing-masing penggunaan.
2. Pengembang berikutnya dapat mengeksplorasi pendekatan lainnya dalam membuat algoritma penjadwalan otomatis. Perlu dibandingkan dengan algoritma greedy yang dibuat pada program ini. Salah satu pendekatan yang dapat dicoba adalah backtracking, pendekatan matematis untuk kasus spesifik, dan pendekatan prioritas.
3. Pengembang berikutnya dapat melakukan tuning terhadap pembobotan skor pada algoritma penentu penugasan dokter pada suatu shift. Tuning ini harus disesuaikan dengan kebutuhan rumah sakit saat itu.
4. Untuk kemudahan pembacaan jadwal, pengembang berikutnya dapat membuat tampilan GUI untuk menampilkan jadwal dalam bentuk kalender yang menyesuaikan hari dan tanggal pada bulan saat itu. Bentuk tampilan seperti ini lebih intuitif dan mudah dicerna lebih banyak kalangan.

BAB VII

PEMBAGIAN TUGAS

Nama	NIM	Tugas Substansional
Deven Zie Manuela Sitanggang	13223091	<ul style="list-style-type: none">- Membuat subprogram laporan1.c- Membuat PPT presentasi- Membuat laporan bagian penampilan informasi jadwal
Muhammad Luthfii Alghazali	13223097	<ul style="list-style-type: none">- Membuat program utama main.c- Membantu membuat program penjadwalan.c- Bug fix dan revisi dokter.c (sub program pengelolaan data dokter)- Melakukan analisis hasil studi kasus, kesimpulan, dan saran (operasi program).- Membuat laporan (semua bagian program utama, BAB V, dan BAB VI)- Revisi instruksi dasar- Membuat salindia (bagian analisis hasil dan logika program utama)
Sarah Beatrix Sitorus	13223100	<ul style="list-style-type: none">- Membuat sub pprogram dokter.c- Membutat PPT presentasi- Membuat laporan Tugas Besar-
Naufal Ridho Wicaksono	13223102	<ul style="list-style-type: none">- Membuat subprogram laporan.c- Membuat laporan bagian informasi dan laporan dokter- Revisi dan tambahan fungsi

		<p>di laporan.c</p> <ul style="list-style-type: none"> - Menyusun ppt
Abdul Hakim Yafi	13223107	<ul style="list-style-type: none"> - Membuat repository github - Membuat sub program penjadwalan.c - Membantu mengintegrasikan program main.c - Mengimplementasikan kode main ke GUI menggunakan GTK - Membuat Drive Terpusat - Membuat MasterSheet utama - Membuat Instruksi Dasar - Membuat PPT presentasi (pendahuluan dan logika struktu penjadwalan.c serta implementasi paga GUI) - Membuat Laporan Tugas Besar (Semua yang berhubungan dengan penjadwalan.c dan bab 1, bab 2 secara garis besar) - Selalu berinisiatif dan mengingatkan tugas di grup internal

DAFTAR PUSTAKA

- [1]A. Cormen, *Introduction to algorithms*. Cambridge, Mass.: Mit Press, 2003.
- [2]Greedy Method - Introduction. www.youtube.com.
https://www.youtube.com/watch?v=ARvQcqJ_-NY (accessed 2022-09-13).
- [3]GeeksforGeeks. Greedy Algorithms. GeeksforGeeks.
<https://www.geeksforgeeks.org/dsa/greedy-algorithms/>.
- [4] Simple Option Menu (C)
<https://stackoverflow.com/questions/33929671/simple-option-menu-c>

LAMPIRAN

Github: [Repository GitHub TUBES PMP Kelompok 3](#)

The screenshot shows the GitHub repository page for 'TUBES-PMP'. The repository is private and has 1 branch and 0 tags. The main file listed is 'main'. The repository description is 'Aplikasi Penjadwalan Dokter Otomatis untuk Rumah Sakit'. The README file contains the following text:

Aplikasi Penjadwalan Dokter Otomatis Rumah Sakit

Ini adalah proyek Tugas Besar Pemecahan Masalah dengan Pemrograman (PMP) untuk membuat sistem penjadwalan dokter otomatis di RS Semoga Sehat Selalu.

Deskripsi Proyek

Aplikasi ini bertujuan untuk mengotomatisasi proses penjadwalan jaga dokter di rumah sakit selama periode 30 hari. Setiap hari dibagi menjadi tiga shift: pagi, siang, dan malam. Setiap shift harus diisi oleh minimal satu dokter. Sistem ini mempertimbangkan batasan jumlah shift maksimal per minggu untuk setiap dokter serta preferensi shift (misalnya hanya ingin pagi, atau tidak ingin malam).

Tujuan utama adalah menghasilkan jadwal yang:

- Efisien dan adil.
- Menghindari beban kerja berlebih.
- Mengakomodasi preferensi pribadi dokter sebanyak mungkin.

Drive Utama: [TUBES PMP](#)

The screenshot shows the Google Drive interface for the 'TUBES PMP' folder. The folder contains three files:

- Instruksi Dasar (PDF)
- tubes pmp (Screenshot of a software interface showing a grid of data)
- Tugas-besar_PMP... (PDF)

Mastersheet Utama : [tubes pmp](#)

tubes pmp

File Edit View Insert Format Data Tools Extensions Help

B13 C D E F G H I J K L M N O P

TUBES PMP

an input kode 2	# output kode 3	#output kode 2
jadwal csv	jadwal_dokter.csv	laporan_dokter.csv
jadwal harian	Tanggal	Nama
jadwal mingguan	Hari	Bidang
jadwal bulanan	Shift	Tingkat
jumlah shift	Nama	Total Shift,Max_Per_Minggu
(nalam)	pelanggaran shift dokter	Bidang
, akhir bulan, Campur)	Tingkat	Pref_Waktu
		Pelanggaran
		Minggu_1
		Minggu_2
		Minggu_3
		Minggu_4
		Minggu_5
		Shift_Pagi
		Shift_Siang
		Shift_Malam

TIMELINE

Senin	Selasa	Rabu	Kamis	Jumat	Sabtu	Minggu
26	27	28	29	30	23	24
25					21	22

MAY

Senin	Selasa	Rabu	Kamis	Jumat	Sabtu	Minggu
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24					

JUNI

Senin	Selasa	Rabu	Kamis	Jumat	Sabtu	Minggu
1						

Milestone

- Perencanaan Kode dan implementasi
- Finalisasi dan Tampilkan
- pembuatan proposal dan finalisasi
- rencana integrasi
- blajar.c, h

Instruksi Dasar: Instruksi Dasar

Instruksi Dasar

File Edit View Insert Format Tools Extensions Help

Heading 3 Arial 13px

Panduan Awal Proyek TUBES PMP: Setup Lingkungan & Alur Kerja Git

Halo Tim!

Untuk memastikan pengerjaan *Tugas Besar Pemecahan Masalah dengan Pemrograman (PMP)* kita berjalan lancar, terorganisir, dan efisien, kita akan menggunakan Git (melalui GitHub) untuk manajemen kode dan **VS Code** sebagai editor utama kita.

Panduan ini akan menjelaskan langkah-langkah setup awal dan alur kerja Git dasar yang perlu kita pahami bersama.

1. Struktur Proyek Kita

Proyek kita akan mengikuti struktur modular untuk memisahkan logika dan memudahkan kolaborasi. Ini adalah gambaran *folder* dan *file* yang akan kita gunakan:

```

TUBES-PMP/
├── src/          # Folder untuk semua file kode sumber (.c)
│   ├── main.c    # File utama program & menu
│   ├── dokter.c  # Implementasi fungsi pengelolaan data dokter
│   ├── penjadwalan.c # Implementasi logika penjadwalan otomatis
│   ├── laporan.c # Implementasi fungsi laporan jadwal
│   └── utils.c   # Implementasi fungsi utilitas umum
├── include/      # Folder untuk semua file header (.h)
│   ├── dokter.h  # Deklarasi struk Dokter & fungsi dokter
│   ├── penjadwalan.h # Deklarasi struk jadwal & fungsi penjadwalan
│   ├── laporan.h # Deklarasi fungsi laporan
│   └── utils.h   # Deklarasi fungsi utilitas
└── data/         # Folder untuk file input/output data (CSV)
    ├── daftar_dokter.csv # Input data dokter awal [cite: 7, 25]
    └── jadwal.csv # Output jadwal yang dihasilkan
    ├── Makefile # Skrip untuk mengkomplasi program kita
    ├── .gitignore # Memberitahu Git file apa saja yang harus diabaikan (tidak di-commit)
    └── README.md # Deskripsi proyek kita
    ├── .vscode/ # Konfigurasi spesifik VS Code (jangan dihapus!)
        ├── tasks.json # Menjalankan perintah 'make' dari VS Code
        └── launch.json # Konfigurasi untuk debugging di VS Code

```

Dokumentasi saat pengeroaan:

