

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



Sistem Pengelolaan Rumah Sakit Berbasis C dan CLI

Disusun oleh:
Kelompok G/K-03

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

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

PERNYATAAN KEJUJURAN

Dengan ini, kami:

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

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

DAFTAR ISI

PERNYATAAN KEJUJURAN.....	2
DAFTAR ISI.....	3
DAFTAR TABEL.....	4
DAFTAR GAMBAR.....	5
BAB I.....	6
PENDAHULUAN.....	6
1.1. Deskripsi Permasalahan.....	6
1.2. Tujuan.....	6
BAB II.....	7
PERANCANGAN PROGRAM.....	7
2.1. Rencana Implementasi Fitur.....	7
2.2. Desain Program.....	10
BAB III.....	19
IMPLEMENTASI DAN PENGUJIAN.....	19
3.1. Checklist Progres Fitur.....	19
3.2. Pembagian Tugas Anggota.....	20
3.3. Pengujian.....	21
LAMPIRAN.....	21

DAFTAR TABEL

Tabel 2.1. Rencana Implementasi.....	7
Tabel 2.2. Rincian Command.....	10
Tabel 3.1. Progres fitur.....	19
Tabel 3.2. Pembagian tugas.....	20

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Deskripsi Permasalahan

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

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

1.2 Tujuan

Program ini dikembangkan dengan tujuan untuk:

1. Membuat fitur *login* dan registrasi dengan validasi yang benar.
2. Mengelola data *user* dengan struktur data yang sesuai (Set, List, Queue, Stack).
3. Mengimplementasikan fitur pencarian dan pengurutan data.
4. Mengelola alur kontrol berdasarkan *role* pengguna.
5. Menyusun sistem antrean dan pengobatan.

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

BAB II

PERANCANGAN PROGRAM

2.1. Rencana Implementasi Fitur

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

Tabel 2.1. Rencana Implementasi

Implementasi ADT	Fitur	Deskripsi Implementasi	Alasan Implementasi
ADT List (array of User)	F01 - Login	Fungsi login menggunakan array of User untuk mencocokkan username dan password dari manajer, dokter, atau pasien.	karena array of User menyimpan atribut penting (username, password, role) yang dibutuhkan saat autentikasi, penggunaan linked list untuk akses mudah dan add/remove yang mudah
ADT List (array of User)	F02 - Register Pasien	Data pasien baru disimpan ke dalam array User [], yang merupakan daftar seluruh pengguna bertipe pasien. Setiap elemen memiliki atribut username, password, dan role.	karena Struktur User memudahkan penyimpanan dan akses data pengguna secara terstruktur. Dengan array, pencarian dan pengelolaan data pasien menjadi lebih sistematis.

ADT Set		Set digunakan untuk menyimpan daftar username yang telah terdaftar. Sebelum pasien baru ditambahkan, sistem memeriksa apakah username sudah ada di Set.	karena ADT Set menjamin bahwa username yang digunakan bersifat unik, dan pencarian menggunakan searchSet lebih efisien daripada mencari manual dalam array User[].
ADT List (array of User)	F03 - Logout	Fungsi logout menghapus status login dengan mengubah isLoggedIn menjadi 0 dan mengosongkan data pada currentUser.	Perlu menyimpan status pengguna yang sedang aktif, sehingga logout dapat membatalkan sesi.
ADT List (array of User)	F04 - Lupa Password	Menghasilkan kode unik berbasis username menggunakan algoritma Run Length Encoding (RLE), yang bekerja di atas ADT User.	Kode unik mempermudah akan proses validasi pemulihan akun secara sederhana.
ADT List (User), ADT Linked List	F05 - Menu & Help	List untuk username dan password user, Linked list untuk opsi yang dapat dipilih buat opsi hep	Untuk efisiensi opsi yang dapat digunakan oleh user dan agar saling terhubung
ADT List (list statik berisi Ruangan)	F06 - Denah Rumah Sakit	Menyimpan daftar ruangan dalam array bertipe Ruangan dan menampilkan isi ruangan berdasarkan indeks (kapasitas, dokter, pasien)	karena jumlah ruangan tetap dan dapat diakses via indeks. Tipe Ruangan juga memudahkan pengelompokkan data

ADT List (array of User, DokterInfo, Ruangan)	F10 - Tambah Dokter	Menambahkan akun dokter ke array User dan DokterInfo jika username belum terpakai.	karena data statik mudah diakses via indeks
ADT Set		Mengassign dokter ke ruangan tertentu jika belum ditempati dokter lain	memastikan username dokter tidak duplikat melalui <code>searchSet()</code> dan <code>insertSet() 0'</code>
ADT Map	F-06	Untuk memasukkan denah dari rumah sakit agar mudah diakses	Karena cocok untuk membuat denah dan mudah diakses
ADT Stack, ADT List	F-17, F-16, F-13	Untuk memasukkan obat diminum ke perut dan obat sebelumnya ke dalam inventory	Karena LIFO sehingga pada implementasinya mudah untuk organize obat, list untuk mengakses data pasien
ADT Queue & ADT Map	F-14, F-09	Queue dengan linked list untuk mengetahui nomor antrian serta lokasi mengantri dan ruangan berikutnya	Menghindari terjadinya ruangan sama antrian sama dan lebih mendetil
ADT List (pasien, dokter, manager)	F-07, F-08	Untuk lebih mudah dalam mengakses sejumlah data dalam sekaligus	Menghindar terjadinya data switch-up dan kesesuaian dengan data asli pasien
ADT Linked List (penyakit)	F-11	Menggunakan linked list untuk mencari penyakit dengan keterangan sesuai diagnosa	Agar mencari dari urutan terdepan, dan memetakan penyakit dengan <code>match>1</code> kepada match pertama

2.2. Desain Program

Program ini dirancang menggunakan pendekatan modular dengan membagi fungsionalitas ke dalam beberapa file header dan implementasi. setiap modul memiliki tanggung jawab spesifik, seperti autentikasi pengguna, manajemen data user, validasi username, dan pengelolaan proses keluar program. Program ini memiliki komponen utama yaitu:

- auth-management : menangani login, registrasi, logout, dan lupa password.
- set-management : digunakan untuk validasi keunikan username secara case-insensitive menggunakan ADT set.
- exit : mengatur proses keluar program dengan opsi penyimpanan data.
- file-io : dipersiapkan untuk menyimpan data ke file eksternal.
- user-management : menyimpan struktur data dan definisi pengguna.

2.2.1. Desain Command

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

Tabel 2.2. Rincian *Command*

Fitur	Nama <i>command</i>	Masukan	Keluaran
F01 - Login	Login	username, password	status login berhasil
F02 - Register Pasien	Register_Pasien	username, password, konfirmasi password	status pendaftaran berhasil
F03 - Logout	Logout	-	status logout berhasil
F04 - Lupa Password	Lupa_Password	username, kode unik	-
F05 - Menu & Help	help	HELP	Menampilkan daftar fitur (perintah) yang bisa digunakan oleh user berdasarkan role (Manager,

			Dokter, atau Pasien). Jika belum login, menampilkan fitur untuk user belum login.
	displayHelpMenu	role	Menampilkan daftar fitur sesuai dengan role dalam bentuk paginasi (per halaman).
F06 - Denah Rumah Sakit	LIHAT_DENAH	jumlahRuang	visualisasi kotak denah rumah sakit
	LIHAT_RUANGAN	indeks	informasi kapasitas, dokter, dan pasien di ruangan yang dimaksud
F10 - Tambah Dokter	TAMBAH_DOKTER	username, password	status berhasil/gagal menambah dokter
	ASSIGN_RUANGAN	username, dokter, nomor ruangan	status berhasil/gagal proses assign dokter sesuai ruangnya
F18 - Exit	exit_program	void	print menyimpan data / exit dari program

2.2.2. Desain Kamus Data, Dekomposisi, dan Spesifikasi Modul

● F01 - Login

```

1. {I.S. Data manajer, list dokter, dan pasien}
2. {F.S. Jika login berhasil, currentUser dan isLoggedIn diperbarui; jika
   gagal, tampilkan pesan gagal login}
3.
4. KAMUS
5.   manajer: User
6.   dokterList: array of User
7.   dokterCount: integer
8.   pasienList: array of User
9.   pasienCount: integer
10.  username: string
11.  password: string
12.  i: integer
13.

```

```

14. ALGORITMA
15.  output "Masukkan username: "
16.  input username
17.
18.  output "Masukkan password: "
19.  input password
20.
21.  if username = manager.username and password = manager.password then
22.      currentUser ← manager
23.      isLoggedIn ← 1
24.      output "Login berhasil sebagai Manajer"
25.      return
26.  end if
27.
28.  for i ← 0 to dokterCount -1
29.      if username = dokterList[i].username and password =
        dokterList[i].password then
30.          currentUser ← dokterList[i]
31.          isLoggedIn ← 1
32.          output "Login berhasil sebagai Dokter"
33.          return
34.      end if
35.  end for
36.
37.  for i ← 0 to pasienCount -1
38.      if username = pasienList[i].username and password =
        pasienList[i].password then
39.          currentUser ← pasienList[i]
40.          isLoggedIn ← 1
41.          output "Login berhasil sebagai Pasien"
42.          return
43.      end if
44.  end for

```

- F02 - Register Pasien

```

1. { I.S. pasienList, pasienCount, dan usernameSet terdefinisi }
2. { F.S. Jika username belum terdaftar, pasien baru ditambahkan ke
  pasienList dan username}
3.
4. KAMUS
5.  pasienList: array of User
6.  pasienCount: integer
7.  usernameSet: Set
8.  newUser: User
9.  tempUsername: string
10. tempPassword: string
11. isUnique: boolean
12.
13. ALGORITMA
14.  output "Masukkan username baru: "
15.  input tempUsername
16.
17.  isUnique ← (searchSet(usernameSet, tempUsername) = -1)

```

```

18.
19.   if not isUnique then
20.       output "Registrasi gagal: username sudah digunakan"
21.       return
22.   end if
23.
24.   output "Masukkan password: "
25.   input tempPassword
26.
27.   newUser.username ← tempUsername
28.   newUser.password ← tempPassword
29.   newUser.role ← "pasien"
30.
31.   pasienList[pasienCount] ← newUser
32.   pasienCount ← pasienCount + 1
33.
34.   insertSet(usernameSet, tempUsername)
35.
36.   output "Registrasi berhasil sebagai Pasien"

```

● F03 - Logout

```

1. { I.S. User dalam keadaan login (isLoggedIn = 1 dan currentUser terisi) }
2. { F.S. User berhasil logout, currentUser dikosongkan dan isLoggedIn = 0 }
3.
4. KAMUS
5.   (Tidak ada variabel lokal tambahan; menggunakan variabel global:)
6.   currentUser: User
7.   isLoggedIn: integer
8.
9. ALGORITMA
10.  if isLoggedIn = 1 then
11.      isLoggedIn ← 0
12.      currentUser.username ← " "
13.      currentUser.password ← " "
14.      currentUser.role ← " "
15.      output "Logout berhasil"
16.  end if
17.
18.  else
19.      output "Tidak ada pengguna yang sedang login"
20.  end else

```

● F04 - Lupa Password

```

1. { I.S. Data semua pengguna (manajer, dokter, pasien) sudah tersedia }
2. { F.S. Jika username ditemukan, ditampilkan kode unik hasil encoding;
   jika tidak, tampilkan pesan tidak ditemukan }
3.
4. KAMUS
5.   username: string
6.   i: integer
7.   ditemukan: boolean

```

```

8.   outputKode: string
9.
10. ALGORITMA
11.   output "Masukkan username yang lupa password: "
12.   input username
13.   ditemukan ← false
14.
15.   if username = manager.username then
16.     generateKodeUnik(username, outputKode)
17.     output "Kode unik Anda: ", outputKode
18.     ditemukan ← true
19.   end if
20.
21.   for i ← 0 to dokterCount - 1 do
22.     if username = dokterList[i].username then
23.       generateKodeUnik(username, outputKode)
24.       output "Kode unik Anda: ", outputKode
25.       ditemukan ← true
26.     end if
27.   end for
28.
29.   for i ← 0 to pasienCount - 1 do
30.     if username = pasienList[i].username then
31.       generateKodeUnik(username, outputKode)
32.       output "Kode unik Anda: ", outputKode
33.       ditemukan ← true
34.     end if
35.   end for
36.
37.   if not ditemukan then
38.     output "Username tidak ditemukan"
39.   end if
40.
41.

```

- F05 - Help and Menu

```

#include <stdio.h>
1. #include <string.h>
2. #include <stdlib.h>
3. #include "auth-management.h"
4. #include "help.h"
5.
6. #define MAX_PER_PAGE 4
7.
8. void waitAndClear() {
9.   printf("\nTekan ENTER untuk melanjutkan...");
10.  getchar();
11.  getchar(); // Untuk menangkap newline dari input sebelumnya
12. }
13.
14. void displayPaginated(const char* role, const char* fitur[], int
    jumlahFitur) {
15.   int currentPage = 0;

```

```

16.     int totalPages = (jumlahFitur + MAX_PER_PAGE - 1) / MAX_PER_PAGE;
17.     char command;
18.
19.     do {
20.         system("clear || cls"); // Untuk Windows/Mac/Linux
21.         printf("\n===== HELP (%s - Page %d/%d) =====\n\n",
role, currentPage + 1, totalPages);
22.         int start = currentPage * MAX_PER_PAGE;
23.         int end = (start + MAX_PER_PAGE < jumlahFitur) ? (start +
MAX_PER_PAGE) : jumlahFitur;
24.
25.         for (int i = start; i < end; i++) {
26.             printf("%s\n", fitur[i]);
27.         }
28.
29.         printf("\n(n: next, p: previous, q: quit): ");
30.         scanf(" %c", &command);
31.
32.         if (command == 'n' && currentPage < totalPages - 1)
currentPage++;
33.         else if (command == 'p' && currentPage > 0) currentPage--;
34.     } while (command != 'q');
35. }
36.
37. void displayHelpMenu(const char* role) {
38.     const char* fiturManager[] = {
39.         "LOGOUT: Keluar dari akun yang sedang digunakan",
40.         "LUPA_PASSWORD: Membuat password baru",
41.         "DENAH: Lihat ruangan di RS serta detailnya",
42.         "LIHAT_USER: Melihat semua pengguna (Dokter/Pasien)",
43.         "LIHAT_PASIEN: Melihat data semua pasien",
44.         "LIHAT_DOKTER: Melihat data semua dokter",
45.         "CARI_USER: Mencari data spesifik",
46.         "CARI_PASIEN: Mencari data pasien spesifik",
47.         "CARI_DOKTER: Mencari data dokter spesifik",
48.         "TAMBAH_DOKTER : Mendaftarkan dokter baru ke sistem",
49.         "ASSIGN_DOKTER : Menempatkan dokter ke dalam ruangan tertentu",
50.         "LIHAT_SEMUA_ANTRIAN : Melihat denah dan antrian ruangan",
51.         "LIHAT_FINANSIAL : Melihat kondisi keuangan rumah sakit",
52.         "UBAH_DENAH : Mengubah ukuran denah rumah sakit",
53.         "PINDAH_DOKTER : Memindahkan dokter antar ruangan"
54.     };
55.
56.     const char* fiturDokter[] = {
57.         "LOGOUT: Keluar dari akun yang sedang digunakan",
58.         "LUPA_PASSWORD: Membuat password baru",
59.         "DENAH: Lihat ruangan di RS serta detailnya",
60.         "DIAGNOSIS: Melakukan diagnosis penyakit",
61.         "NGOBATIN: Memberikan obat kepada pasien",
62.         "LIHAT_RUANGAN: Melihat detail ruangan",
63.         "LIHAT_DOMPET: Melihat jumlah Banarich"
64.     };
65.

```

- F06 - Denah Rumah Sakit

```
1. Procedure lihatDenah(jumlahRuang: integer)
2. { I.S. jumlahRuang terdefinisi sebagai jumlah ruangan rumah sakit }
3. { F.S. Menampilkan denah rumah sakit ke layar berupa kotak-kotak berisi
   nomor ruangan }
4.
5. KAMUS
6.   jumlahRuang: integer
7.   i: integer
8.
9. ALGORITMA
10.  output "+"
11.  for i ← 0 to jumlahRuang - 1 do
12.    output "-----+"
13.  end for
14.
15.  output newline + "|"
16.  for i ← 0 to jumlahRuang - 1 do
17.    output " ", i + 1, " |"
18.  end for
19.
20.  output newline + "+"
21.  for i ← 0 to jumlahRuang - 1 do
22.    output "-----+"
23.  end for
24.
25.  output newline

26. Procedure lihatRuangan(indeks: integer, ruanganList: array of Ruangan)
27. { I.S. indeks dan ruanganList terdefinisi }
28. { F.S. Menampilkan detail ruangan ke layar: kapasitas, dokter (jika ada),
   dan daftar pasien (jika ada) }
29.
30. KAMUS
31.  indeks, i: integer
32.  r: Ruangan
33.  kapasitas: integer
34.  dokter: string
35.  pasien: array [1..N] of string
36.  jumlahPasien: integer
37.
38. ALGORITMA
39.  r ← ruanganList[indeks]
40.
41.  output "--- Detail Ruangan ", indeks + 1, " ---"
42.  output "Kapasitas : ", r.kapasitas
43.
44.  if panjang(r.dokter) > 0 then
45.    output "Dokter : ", r.dokter
46.  else
47.    output "Dokter : -"
48.  end if
49.
```



```

50. output "Pasien di dalam ruangan:"
51. if r.jumlahPasien = 0 then
52.   output " Tidak ada pasien di dalam ruangan saat ini."
53. else
54.   for i ← 0 to r.jumlahPasien - 1 do
55.     output " ", i + 1, ". ", r.pasien[i]
56.   end for
57. end if
58.
59. output "-----"

```

● F10 - Tambah Dokter

```

1. Procedure tambahDokter(dokterList: array of User,
2.                         dokterInfoList: array of DokterInfo,
3.                         dokterCount: byref integer,
4.                         usernameSet: byref Set)
5. { I.S. Data dokterList, dokterInfoList, usernameSet, dan dokterCount
  terdefinisi }
6. { F.S. Jika username belum terdaftar, dokter baru ditambahkan ke semua
  struktur terkait }
7.
8. KAMUS
9.   username, password, role, ruangan: string
10.
11. ALGORITMA
12.   input username
13.   if searchSet(usernameSet, username) then
14.     output "Sudah ada Dokter bernama", username
15.     return
16.   end if
17.
18.   input password
19.
20.   dokterList[dokterCount].username ← username
21.   dokterList[dokterCount].password ← password
22.   dokterList[dokterCount].role ← "Dokter"
23.
24.   dokterInfoList[dokterCount].username ← username
25.   dokterInfoList[dokterCount].ruangan ← ""
26.
27.   dokterCount ← dokterCount + 1
28.   insertSet(usernameSet, username)
29.
30.   output "Dokter", username, "berhasil ditambahkan!"

```

```

1. Procedure assignRuangan(dokterList: array of User,
2.                         dokterInfoList: array of DokterInfo,
3.                         dokterCount: integer,
4.                         ruanganList: array of Ruangan,
5.                         jumlahRuangan: integer)
6. { I.S. Semua data array terdefinisi. User menginput username dokter dan
  nomor ruangan }

```

```

7. { F.S. Jika valid dan belum terpakai, dokter diassign ke ruangan dan
   sebaliknya }
8.
9. KAMUS
10. username: string
11. nomorRuangan, idxDokter, idxRuangan: integer
12.
13.     kapasitas: integer
14.     dokter: string
15.     pasien: array [1..N] of string
16.     jumlahPasien: integer
17.
18. ALGORITMA
19.     input username
20.     input nomorRuangan
21.
22.     if nomorRuangan < 1 or nomorRuangan > jumlahRuangan then
23.         output "Nomor ruangan tidak valid!"
24.         return
25.     end if
26.
27.     idxDokter ← -1
28.     for i ← 0 to dokterCount - 1 do
29.         if dokterList[i].username = username then
30.             idxDokter ← i
31.             break
32.         end if
33.     end for
34.
35.     if idxDokter = -1 then
36.         output "Dokter", username, "tidak ditemukan!"
37.         return
38.     end if
39.
40.     if dokterInfoList[idxDokter].ruangan ≠ "" then
41.         output "Dokter", username, "sudah diassign ke ruangan",
           dokterInfoList[idxDokter].ruangan
42.         return
43.     end if
44.
45.     idxRuangan ← nomorRuangan - 1
46.     if ruanganList[idxRuangan].dokter ≠ "" then
47.         output "Ruangan", nomorRuangan, "sudah ditempati oleh dokter",
           ruanganList[idxRuangan].dokter
48.         return
49.     end if
50.
51.     dokterInfoList[idxDokter].ruangan ← username
52.     ruanganList[idxRuangan].dokter ← username
53.
54.     output "Dokter", username, "berhasil diassign ke ruangan", nomorRuangan

```

BAB III

IMPLEMENTASI DAN PENGUJIAN

3.1. Checklist Progres Fitur

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

Tabel 3.1. Progres fitur

Fitur	Desain	Implementasi	Testing
F01- Login	X	V	X
F02 - Register Pasien	X	V	X
F03 - Logout	X	V	X
F04 - Lupa Password	X	V	X
F05 - Menu & Help	V	V	V
F06 - Denah Rumah Sakit	V	V	X
F10 - Tambah Dokter	X	V	X
F18 - Exit	V	V	X

3.2. Pembagian Tugas Anggota

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

Tabel 3.2. Pembagian tugas

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F01- Login		18224053	18224053	18224061
F02 - Register Pasien		18224053	18224053	18224061
F03 - Logout		18224053	18224053	18224061
F04 - Lupa Password		18224053	18224053	18224061
F05 - Menu & Help		18224017	18224017	18224061
F06 - Denah Rumah Sakit		18224045	18224061	18224061
F10 - Tambah Dokter		18224045	18224045	18224061
F18 - Exit		18224117	18224117	18224061

3.3. Pengujian

LAMPIRAN

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

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

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

Catatan Asistensi:

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

Dokumentasi

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

Elbert Challes (Presenting)

The diagram is a hand-drawn flowchart on a whiteboard. It starts with 'Foo' on the left, which branches into 'Subject/object' and 'field'. 'Subject/object' leads to a box containing '1. type struct' and 'Diver'. 'field' leads to a box containing 'Diver' and 'Diver'. These boxes are connected to a central box labeled 'modu -> modular'. From there, the flow goes to 'FOU - FID', then to a box labeled 'robot', and finally to a box labeled 'save' and 'test'. There are also some handwritten notes like 'Bench Arrival' and 'Bench'.

Elbert Challes

18224117 Ste...

18224061 Ria...

18224053 Na...

18224017 Na...

18224045 M...

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

21:42 Tuesday 06/05/2023