

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



**Disusun oleh Kelompok N K-01**

**Anggota:**

<b>Muhammad Ashkar</b>	<b>13524007</b>
<b>Muhammad Haris Putra S.</b>	<b>13524053</b>
<b>Aurelia Jennifer Gunawan</b>	<b>13524089</b>
<b>Nathanael Shane Bennet</b>	<b>13524119</b>
<b>Yuhan Fanzuri Nizar</b>	<b>13524123</b>
<b>Fazri Arrashyi Putra</b>	<b>13524127</b>

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

## Halaman Pernyataan Kelompok

1. Nama : Muhammad Ashkar  
NIM : 13524007

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

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

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

3. Nama : Aurelia Jennifer Gunawan  
NIM : 13524089

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

4. Nama : Nathanael Shane Bennet  
NIM : 13524119

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

5. Nama : Yuhan Fanzuri Nizar  
NIM : 13524123

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

6. Nama : Fazri Arrashyi Putra  
NIM : 13524127

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

## Daftar Isi

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

## **Daftar Tabel**

Tabel 1. Rencana Implementasi

Tabel 2. Daftar Pembagian Kerja Kelompok

Tabel 3. Checklist Hasil Rancangan

Tabel 4. Desain Command Primitif

## **Daftar Gambar**

Gambar 1. Load, awal folder

Gambar 2. Load, tidak ada nama folder yang diinput

Gambar 3. Load, tidak ada folder yang sesuai

Gambar 4. HalamanUtama, awal masuk

Gambar 5. HalamanUtama, input salah

Gambar 6. login , awal masuk

Gambar 7. login , tidak ada usn

Gambar 8. login , valid, pasien

Gambar 9. register, usn sudah dipakai

Gambar 10. register, format usn salah

Gambar 11. register, valid, usn benar

Gambar 12. LupaPassword, usn tidak terdaftar

Gambar 13. LupaPassword, usn benar, kode unik salah

Gambar 14. LupaPassword, usn benar, kode unik benar, diminta password

Gambar 15. Exit, input 4

Gambar 16. Exit, input salah

Gambar 17. Exit, input y, mau save

Gambar 18. Save, Folder buatan save

Gambar 19. Save, user tidak mau save

Gambar 20. Pesan keluar dari sistem

Gambar 21. Pasien, awal

Gambar 22. Pasien, input tidak sesuai

Gambar 23. Logout, validasi

Gambar 24. Logout, y, kembali ke halaman utama

Gambar 25. Logout, n

Gambar 26. Logout, n, kembali ke laman pasien

Gambar 27. History commit repo github yang tidak tercantum di github classroom

Gambar 28. Hasil pindai MoM form asistensi 1

## A. Deskripsi Persoalan

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

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

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

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

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

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

## B. Rencana Implementasi

Tabel 1. Rencana Implementasi

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

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



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

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

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

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

### C. Daftar Pembagian Kerja Kelompok

Tabel 2. Daftar Pembagian Kerja Kelompok

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

F05 – Menu & Help	Help diberikan sesuai role, ADT Sederhana	13524053 13524089	13524053 13524089	13524053 13524089
F06/D01 – Denah Rumah Sakit				
F07 – Lihat User				
F08 – Cari User				
F09/D02 – Lihat Antrian				
F10 – Tambah Dokter				
F11 – Diagnosis		13524053	13524053	13524053
F12 – Ngobatin				
F13 – Aku boleh pulang ga, 🥺 dok?				
F14 – Daftar Check-Up				
F15 – Antrian Saya!		13524089	13524089	13524089
F16 – Minum Obat		13524089	13524089	13524089
F17 – Minum Penawar		13524053	13524053	13524053
F18 – Exit		13524053	13524053	13524053
D03 – Load	Prosedur CSVtoArr, Memuat folder sesuai input user, File Eksternal, fungsi & prosedur	13524053 13524089	13524053 13524089	13524053
D04 – Save	Prosedur ArrtoCSV, Menyimpan data ke folder sesuai input user, File Eksternal, fungsi & prosedur	13524053 13524089	13524053 13524089	13524053

## D. Checklist Hasil Rancangan

Tabel 3. Checklist Hasil Rancangan

Fitur	Desain	Implementasi	Testing
F01 – Login	V	V	V
F02 – Register Pasien	V	V	V
F03 – Logout	V	V	V
F04 – Lupa Password	V	V	V
F05 – Menu & Help	V	V	V
F06/D01 – Denah Rumah Sakit	V	V	V
F07 – Lihat User	-	-	-
F08 – Cari User	-	-	-
F09/D02 – Lihat Antrian	-	-	-
F10 – Tambah Dokter	-	-	-
F11 – Diagnosis	-	-	-
F12 – Ngobatin	-	-	-
F13 – Aku boleh pulang ga, 🙄 dok?	-	-	-
F14 – Daftar Check-Up	-	-	-
F15 – Antrian Saya!	-	-	-
F16 – Minum Obat	-	-	-
F17 – Minum Penawar	-	-	-
F18 – Exit	V	V	V
D03 – Load	V	V	V
D04 – Save	V	V	V

**Keterangan:**

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

## E. Desain Command Primitif

Tabel 4. Desain Command Primitif

Nama Command	Masukan	Keluaran
CSVtoArr	Path file	array User[], totalUser
ArrtoCSV	path file, array User[], totalUser	file .csv yang tersimpan
cekFormatUsn	Inputan username dari user	Skema validasi 1, input hanya boleh huruf
createUser	username, password, role, identitas, kondisi	struct User
appendUser	list users[], totalUser, userBaru	list baru dengan userBaru ditambahkan
isUsernameAvailable	array User[], totalUser, username	boolean (true jika username belum ada)
cariUserLogin	username, password	indeks User dalam array / -1 jika tidak ada

## F. Desain Kamus Data

Identitas : <

id : integer,  
 username : string ,  
 password : string ,  
 role : string {"Admin", "Dokter", "Pasien"} ,  
 riwayat\_penyakit : string >

KondisiPasien : <

suhu\_tubuh : float (dalam °C) ,  
 tekanan\_darah\_sistolik : integer (mmHg) ,  
 tekanan\_darah\_diastolik : integer (mmHg) ,  
 detak\_jantung : integer (bpm) ,

saturasi\_oksigen : float(%) ,  
kadar\_gula\_darah : integer(mg/dL) ,  
berat\_badan : float(kg) ,  
tinggi\_badan : integer(cm) ,  
kadar\_kolesterol : integer(mg/dL) ,  
kadar\_kolesterol\_ldl : integer(mg/dL) ,  
trombosit : integer(/μL) >

User : < identitas: Identitas, kondisi: KondisiPasien >

## G.Desain Dekomposisi Algoritmik dan Fungsional Program

Struktur fungsi utama :

1. login(users, totalUser, &userAktif)
  - ↳ validasi username & password, set userAktif
2. logout()
  - ↳ reset status login
3. registerpasien(&users, &totalUser)
  - ↳ input data, cek duplikat username, simpan ke array
4. lupapassword(user, users[], jumlah user)
  - ↳ memberikan opsi untuk mengganti password, dengan syarat kode unik run length code
5. CSVtoArr(path, &users, &totalUser)
  - ↳ baca file CSV dan parsing ke array User
6. ArrtoCSV(path, users, totalUser)
  - ↳ simpan array User ke file CSV
7. load(folder\_name)
  - ↳ memuat data dari folder sesuai masukan
8. save(folder\_name)
  - ↳ simpan data ke folder sesuai masukan
9. help(user.role)
  - ↳ memberikan bantuan penjelasan sesuai role user, jika role tidak ada, berarti belum login, dan akan diberikan bantuan penjelasan untuk login



## H. Spesifikasi Modul, Prosedur, dan Fungsi

### ALGORITMA login

#### KAMUS

user, users : array of User (extern)

jumlah\_user : integer (extern)

valid, format, i : integer

nama, role : string

#### ALGORITMA

clearScreen()

cekFormatUsn(format, user)

nama ← user.identitas.username

output "Password: "

input user.identitas.password

valid ← 0

i traversal [0..jumlah\_user-1]

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

if lowercase(users[i].identitas.password) = lowercase(user.identitas.password) then

valid ← 1

role ← users[i].identitas.role

user.identitas.role ← role

break

else

output "Username or password salah untuk pengguna yang bernama ", nama

valid  $\leftarrow$  -1

if valid = 0 then

output "Tidak ada Manager, Dokter, or pun Pasien yang bernama ", nama

else if valid = 1 then

output "Selamat pagi ", role, " ", nama

### **ALGORITMA logout**

#### **KAMUS**

user : pointer to User (extern)

#### **ALGORITMA**

if user  $\neq$  NULL then

dealokasi user

user  $\leftarrow$  NULL

### **ALGORITMA registerpasien**

## KAMUS

user, users : array of User (extern)

jumlah\_user, i, format : integer

valid : boolean

## ALGORITMA

clearScreen()

cekFormatUsn(format, user)

valid  $\leftarrow$  true

i traversal [0..jumlah\_user-1]

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

valid  $\leftarrow$  false

if not valid then

output "Registrasi gagal! Pasien dengan nama ", user.identitas.username, " sudah terdaftar."

dealokasi user

user  $\leftarrow$  NULL

return

else

realokasi users menjadi (jumlah\_user + 1)

output "Password: "

```

 user.identitas.password

salin data user ke users[jumlah_user]

users[jumlah_user].identitas.role ← "pasien"

jumlah_user ← jumlah_user + 1

output "Pasien ", user.identitas.username, " berhasil ditambahkan!"

```

### ALGORITMA lupaPassword

#### KAMUS

```

user, users : array of User (extern)

jumlah_user, idx, format, i, p, j, r, k : integer

valid : boolean

kodeUnik, pass, count : string

c : char

result : integer

```

#### ALGORITMA

```

cekFormatUsn(format, user)

valid ← false

i traversal [0..jumlah_user-1]

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

    valid ← true

    idx ← i

```

if not valid then

output "Username tidak terdaftar!"

else

output "Kode Unik: "

input kodeUnik

$i \leftarrow 0$  ;  $p \leftarrow 0$

while kodeUnik[i]  $\neq$  null do

$c \leftarrow$  kodeUnik[i]

if c adalah huruf then

$i \leftarrow i + 1$

count  $\leftarrow$  ""

$k \leftarrow 0$

while kodeUnik[i] adalah digit do

count  $\leftarrow$  count + kodeUnik[i]

$i \leftarrow i + 1$

result  $\leftarrow$  konversi count ke integer

i traversal [1..result]

pass  $\leftarrow$  pass + c

$p \leftarrow p + 1$

else

$i \leftarrow i + 1$

$pass[p] \leftarrow null$

if lowercase(user.identitas.password) = lowercase(pass) then

output "Halo Dokter ", user.identitas.username, ", silakan daftarkan ulang password anda!"

output "Password baru: "

input users[idx].identitas.password

output "Berhasil!"

else

output "Kode unik salah!"

## **ALGORITMA ArrtoCSV**

### **KAMUS**

filename : string

handler : CSVRowHandler

file : SEQFILE of

(\*) string

### **ALGORITMA**

Assign (file, "file")

If gagal buka, keluar dengan error

handler(NULL, -1, file)

close (file)

## ALGORITMA CSVtoArr

### KAMUS

filename : string

handler : CSVRowHandler

file : SEQFILE of

(\*) string

line : string

fields : array of string

fieldIdx, charIdx, i : integer

c : char

### ALGORITMA

Assign (filename,"file")

If gagal buka, keluar dengan error

read (file,header)

read (file,baris)

fieldIdx  $\leftarrow$  0, charIdx  $\leftarrow$  0

untuk setiap karakter dalam baris

```

If c = ';' or '\n' or null then
    akhiri string fields[fieldIdx]
    fieldIdx++
    charIdx ← 0
    If c = '\n' or null then break
else
    fields[fieldIdx] <- c
    charIdx++

handler(fields, fieldIdx, target)

close (file)

```

## **ALGORITMA load**

### **KAMUS**

```

userpath, penyakitpath, folderpath : string
st : struct stat
pt : ParseTarget
pp : ParsePenyakit
temp_users : pointer to User

```

### **ALGORITMA**

```

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

```



```
folderpath ← "data/folder_name"
```

If folder tidak ada, tampilkan pesan and keluar

```
jumlah_user <- 0
```

```
jumlah_penyakit <- 0
```

inisialisasi pt and pp

```
CSVtoArr(userpath, handleUserRow, pt)
```

```
CSVtoArr(penyakitpath, handlePenyakitRow, pp)
```

realokasi users ke jumlah\_user

If gagal, keluarkan error and keluar

```
users ← temp_users
```

```
output "Loading..."
```

## **ALGORITMA save**

### **KAMUS**

base\_path, full\_path, test\_file, original\_path : string

userfile : SEQFILE of

(\*) string

test : SEQFILE of

(\*) string

Jumlah\_user : integer (extern)

## ALGORITMA

full\_path ← "data/folder\_name"

test\_file ← "data/folder\_name/user.csv"

Assign (testfile,"test")

If test = NULL then

output "Membuat folder..."

    make\_folder(full\_path)

output "Saving..."

else

close test

output "Saving..."

rewrite(userfile)

If berhasil buka then

write(userfile,header)

    I traversal [0..jumlah\_user-1]

write(userfile,data user dalam format CSV)

close userfile

output "Berhasil menyimpan data di folder!"

else

output "Gagal menyimpan file"

## **ALGORITMA help**

### **KAMUS**

role, username : string

### **ALGORITMA**

output "===== HELP ====="

If role = "" then

tampilkan perintah umum: LOGIN and REGISTER

else If role = "Dokter" then

tampilkan daftar perintah untuk dokter

else If role = "Pasien" then

tampilkan daftar perintah untuk pasien

else If role = "Manager" then

tampilkan daftar perintah untuk manager

tampilkan footnote

[IF1210 Spesifikasi Tugas Besar - Google Docs](#)

## I. Hasil Pengujian Program

Load folder sesuai keinginan user, jika valid:

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

Gambar 1. Load, awal folder

Jika tidak ada nama folder::

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

Gambar 2. Load, tidak ada nama folder yang diinput

Jika tidak ditemukan:

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

Gambar 3. Load, tidak ada folder yang sesuai

Halaman utama

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

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

Gambar 4. HalamanUtama, awal masuk

Jika Input tidak valid:

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

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

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

Gambar 5. HalamanUtama, input salah

Jika input 1, masuk ke login

```
Username: █
```

Gambar 6. login , awal masuk

Jika username/password salah

```
Username: f
Password: g

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

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

Gambar 7. login , tidak ada usn

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

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

Selamat pagi a! Ada keluhan apa?

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

>>> Masukkan pilihan (1-5): █
```

Gambar 8. login , valid, pasien

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

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

Gambar 9. register, usn sudah dipakai

Jika input usn tidak sesuai format maka

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

Gambar 10. register, format usn salah

Jika input usn sesuai

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

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

Gambar 11. register, valid, usn benar

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

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

Gambar 12. LupaPassword, usn tidak terdaftar

Jika usn benar, kode unik salah

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

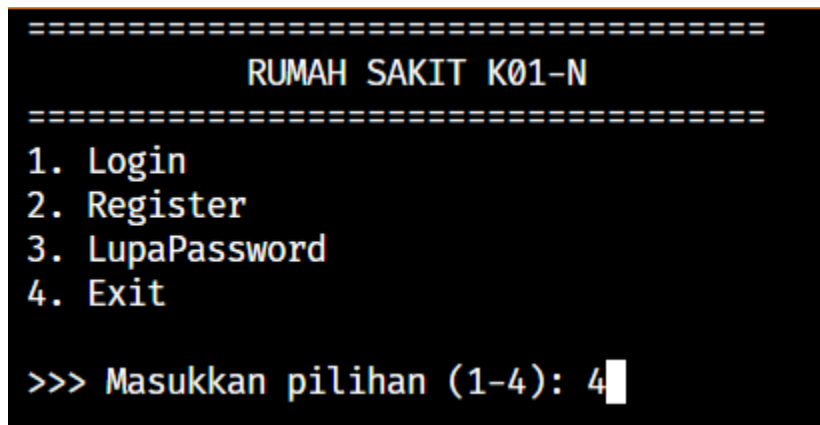
Gambar 13. LupaPassword, usn benar, kode unik salah

Jika keduanya benar, maka valid

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

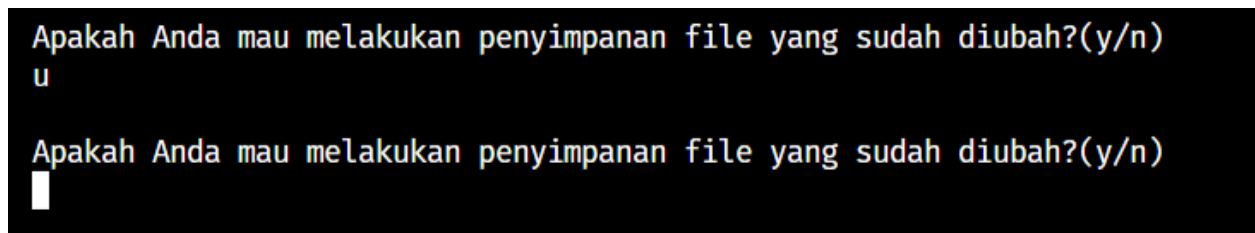
Gambar 14. LupaPassword, usn benar, kode unik benar, diminta password

Jika input 4, mau exit



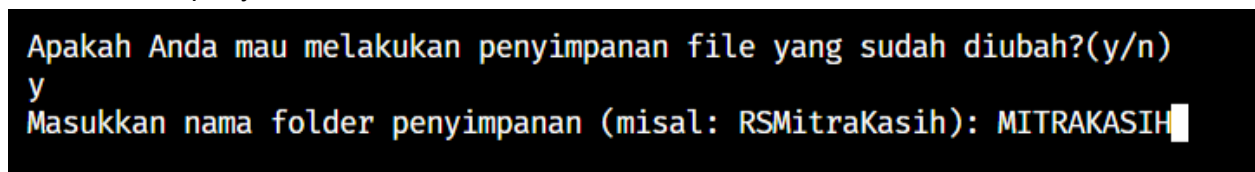
Gambar 15. Exit, input 4

Jika mau save file, dan input salah



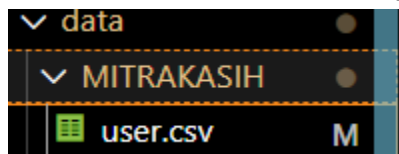
Gambar 16. Exit, input salah

Jika input y maka mau save file



Gambar 17. Exit, input y, mau save

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



Gambar 18. Save, Folder buatan save

Jika sudah ada foldernya, akan overwrite folder yang sudah ada



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

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

Gambar 19. Save, user tidak mau save

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

Gambar 20. Pesan keluar dari sistem

#### LAMAN PASIEN

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

Selamat pagi a! Ada keluhan apa?

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

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

Gambar 21. Pasien, awal

Jika input tidak sesuai

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

Selamat pagi a! Ada keluhan apa?

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

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

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

Gambar 22. Pasien, input tidak sesuai

Jika input 5, mau logout validasi

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

Gambar 23. Logout, validasi

Jika y, maka mau, akan kembali ke halaman utama

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

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

>>> Masukkan pilihan (1-4): █
```

Gambar 24. Logout, y, kembali ke halaman utama

Jika n, maka tidak mau, kembali ke laman pasien

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

Selamat pagi a! Ada keluhan apa?

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

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

Gambar 25. Logout, n

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

Selamat pagi a! Ada keluhan apa?

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

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

Gambar 26. Logout, n, kembali ke laman pasien

## Lampiran

<b>feat: add lupa password</b> MHarisPutraS committed 3 days ago	0f30767		
Commits on May 6, 2025			
<b>feat: add logout, add verifikasi save file setelah exit, add menu pasien, fix: struktur variabel yang berantakan</b> MHarisPutraS committed 5 days ago	f088d9e		
Commits on May 3, 2025			
<b>feat: add registerpasien, sudah bisa menulis data user arr ke CSV</b> MHarisPutraS committed last week	7595f49		
<b>feat: add UI menu, clear terminal saat pilih opsi menu, add login(tinggal final touch), sinkronisasi data sudah bisa</b> MHarisPutraS committed last week	bd795fd		
Commits on Apr 30, 2025			
<b>feat: UI menu, login (belum selesai), tinggal sinkronisasi dengan data</b> MHarisPutraS committed last week	f338dc2		
<b>docs: perubahan struktur folder sesuai spek tubes</b> MHarisPutraS committed last week	6e89934		
Commits on Apr 29, 2025			
<b>test: Mengerjakan ParserCSV.c(belum selesai), dan user.c(belum selesai)</b> MHarisPutraS committed 2 weeks ago	0c9c22a		
<b>chore: Inisialisasi Parsing CSV, Input Data, Inisialisasi ADT User</b> MHarisPutraS committed 2 weeks ago	16d5761		
<b>Initial commit</b> MHarisPutraS authored 2 weeks ago	0ea6637		

[history commit di repo sebelum bisa masuk ke github classroom.jpg]

Gambar 27. History commit repo github yang tidak tercantum di github classroom

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

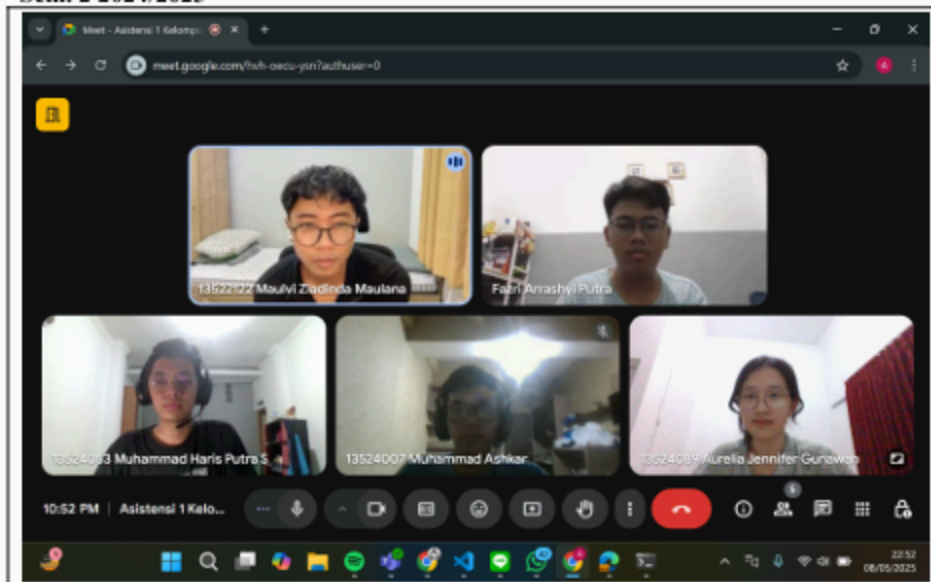
Nomor Asistensi : 01  
 No. Kelompok/Kelas : N/01  
 Tanggal asistensi : Kamis, 8 Mei 2025

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

**Catatan Asistensi:**

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

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



A handwritten signature in black ink, appearing to read "Maulvi Ziadinda Maulana".

**Maulvi Ziadinda Maulana  
13522122**

Gambar 28. Hasil pindai MoM form asistensi 1

**Note: Laporan ini 90% dikerjakan hanya oleh 13524053**