

TUGAS BESAR

Mata Kuliah : IF1210 Algoritma dan Pemrograman 1

Milestone 1



Disusun oleh Kelompok L Kelas 01:

Junior Natra Situmorang	13524055
Kalyca Nathania B. Manullang	13524071
Yavie Azka Putra Araly	13524077
Muhammad Faiz Alfada Dharma	13524097
Rainaldi Pratama F. Sembiring	13524117
Reysa Syafitri Mulya Ramadhan	13524137

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2025

HALAMAN PERNYATAAN

Kelompok K01-L – IF1210 Algoritma dan Pemrograman 1

Semester 2 Tahun Akademik 2024/2025

Sekolah Teknik Elektro dan Informatika – ITB

Kami, kelompok K01-L yang beranggotakan:

1. 13524055 – Junior Natra Situmorang
2. 13524071 – Kalyca Nathania B. Manullang
3. 13524077 - Yavie Azka Putra Araly
4. 13524097 – Muhammad Faiz Alfada Dharma
5. 13524117 – Rainaldi Pratama F. Sembiring
6. 13524137 – Reysa Syafitri Mulya Ramadhan

dengan ini menyatakan hal berikut:

“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.”

Demikian pernyataan ini dibuat dengan sebenar-benarnya.

DAFTAR ISI

Halaman Pernyataan.....	1
Daftar Isi.....	2
Daftar Tabel.....	3
Daftar Gambar.....	4
Bab I Deskripsi Persoalan.....	5
Bab II Rencana Implementasi.....	6
Bab III Daftar Pembagian Kerja Anggota Kelompok.....	7
Bab IV <i>Checklist</i> Hasil Rancangan, Implementasi, dan <i>Testing</i>	12
Bab V Desain <i>Command</i> untuk Setiap Primitif.....	14
Bab VI Desain Kamus Data.....	15
Bab VII Desain Dekomposisi Algoritmik dan Fungsional Program.....	16
Bab VIII Spesifikasi untuk Tiap Modul/Prosedur/Fungsi yang Dibuat (dalam Notasi Algoritmik).....	17
Bab IX Tangkapan Layar Hasil Pengujian Program Berdasarkan Fitur-fitur pada Spesifikasi.....	21
Bab X Lampiran.....	22

DAFTAR TABEL

Tabel 2.1 Rencana Implementasi.....	6
Tabel 3.1 Daftar Pembagian Kerja Anggota Kelompok.....	7
Tabel 4.1 <i>Checklist</i> Hasil Rancangan, Implementasi, dan <i>Testing</i>	12

DAFTAR GAMBAR

Gambar 10.1 Asistensi 1.....	23
------------------------------	----

BAB I

DESKRIPSI PERSOALAN

Tugas besar *milestone* 1 meminta penulis untuk membangun sebuah program berbasis terminal (*Command Line Interface/CLI*) yang mampu menangani kebutuhan manajemen data pengguna, seperti proses registrasi, *login*, pencarian pengguna, hingga tampilan informasi dalam bentuk visualisasi data. Program yang dibangun bersifat modular dan dikembangkan dalam kelompok sehingga perlu merancang alur program yang terstruktur, fleksibel, dan mudah untuk dikembangkan secara kolaboratif.

Secara umum, program akan berfungsi sebagai sistem manajemen pengguna. Pengguna dapat mendaftar sebagai *user* baru (melalui fitur *register*), masuk ke sistem (*login*), serta mengakses berbagai informasi melalui fitur-fitur seperti lihat *user*, cari *user*, dan lain-lain. Pada menu utama (*help*), pengguna akan diarahkan ke fitur-fitur ini.

Tugas besar sekarang menekankan pentingnya kolaborasi tim dalam mengembangkan modul-modul secara terpisah, tetapi terintegrasi. Untuk itu, dibutuhkan sistem manajemen *state* program yang rapi agar setiap modul dapat saling berkomunikasi dan berbagi data dengan baik. Salah satu pendekatan yang disarankan adalah menggunakan struct global yang berisi semua variabel penting, yang kemudian dapat diparsing ke dalam fungsi-fungsi sesuai kebutuhan, sehingga meminimalisir penggunaan variabel global sembarangan dan meningkatkan keterbacaan serta pengujian program.

Program dikembangkan di lingkungan Linux (disarankan menggunakan WSL untuk pengguna Windows) agar lebih kompatibel dan sesuai dengan konteks penggunaan bahasa C yang digunakan dalam tugas ini. Beberapa fungsi dalam bahasa C dapat memiliki perbedaan perilaku antara Windows dan Linux sehingga penting untuk melakukan pengujian langsung di sistem yang menyerupai target lingkungan program.

Dengan tugas besar, penulis tidak hanya diminta untuk menerapkan konsep dasar pemrograman dan algoritma, tetapi juga dilatih untuk bekerja dalam tim, memahami praktik pengembangan perangkat lunak skala kecil hingga menengah, serta merancang program yang bersifat modular, dapat di-*merge*, dan teruji secara fungsional.

BAB II

RENCANA IMPLEMENTASI

Tabel 2.1 Rencana Implementasi

Implementasi ADT	FITUR	Deskripsi Implementasi	Alasan Implementasi
ADT Sederhana	F01 - Login F02 - Register F04 - Lupa Password F05 - Menu & Help F07 - Lihat User	Digunakan untuk struktur data user.	karena pengen
ADT Set	F02 - Register	Digunakan untuk mengumpulkan data username sehingga tidak memiliki username yang sama.	Karena set tidak bisa memiliki nilai yang sama.

BAB III

DAFTAR PEMBAGIAN KERJA ANGGOTA KELOMPOK

Tabel 3.1 Daftar Pembagian Kerja Anggota Kelompok

Fitur	Implementasi *)	NIM Desainer **)	NIM <i>Coder</i> **)	NIM <i>Tester</i> **)

F01 - Login	function login function getUserIndex function getUserData function getTrombosit function getKadarKolester olLDL function getKadarKolester ol function getTinggiBadan function getSaturasiOksige n function getKadarGulaDar ah function getBeratBadan function getSuhuTubuh function getTekananDarah Sistolik	13524077	13524077	
-------------	---	----------	----------	--

	function getTekananDarah Diastolik function getDetakJantung function stringToInt function stringToFloat procedure getUsername procedure getPassword procedure getRole procedure getRiwayatPenyakit function getId procedure getFieldAt			
--	---	--	--	--

F02 - Register	Procedure registerUser, Procedure toLowerCase, Function isUsernameUnique, Procedure appendUserToFile	13524055	13524055	13524055
F03 - Logout	Function logout	13524055	13524055	13524055
F04 - Lupa Password	Function generateUniqueCode Function validasiCode procedure overwriteUserFile procedure gantiPassword procedure lupaPassword	13524117	13524117	
F05 - Menu & Help	function help procedure print_helpMessage_pasien procedure print_helpMessage_dokter procedure	13524077	13524077	

	print_helpMessage_manager			
F06 - Denah Rumah Sakit				
F07 - Lihat User	procedure urutanSort procedure sortDescID procedure sortAscID procedure sortDescNama procedure sortAscNama procedure lihatUser procedure lihatPasien procedure lihatDokter procedure interfaceUser procedure output	13524097	13524097	
F08 - Cari User	procedure printHorizontalLine function toLowerChar procedure	13524077	13524077	

	strToLower function compareUsername function binarySearchUsername procedure printHeader procedure printUser_general procedure printPasien procedure printPilihan procedure printDokter procedure cariPasien procedure cariDokter procedure cariUser			
F09 - Tambah Dokter				
F10 - Diagnosis				
F11 - Ngobatin				
F12 - Aku boleh				

pulang ga, dok 🙄?				
F13 - Daftar Check-Up				
F14 - Antrian Saya!	procedure daftarPasien procedure cekAntrian	13524071	13524071	13524071
F15 - Minum Obat	procedure daftarObat procedure minumObat	13524071	13524071	13524071
F16 - Minum Penawar				
F17 - Exit	Procedure save procedure cariuser	13524137	13524137	
B05 - Dead or Alive?! 💀				

BAB IV

CHECKLIST HASIL RANCANGAN, IMPLEMENTASI, DAN ***TESTING***

Tabel 4.1 *Checklist* Hasil Rancangan, Implementasi dan *Testing*

Fitur	Desain	Implementasi	<i>Testing</i>
F01 - Login	V	V	X
F02 - Register	V	V	X
F03 - Logout	V	V	X
F04 - Lupa Password	V	V	X
F05 - Menu & Help	V	V	
F06 - Denah Rumah Sakit			
F07 - Lihat User	V	V	X
F08 - Cari User	V	V	
F09 - Tambah Dokter			
F10 - Diagnosis			
F11 - Ngobatin			
F12 - Aku boleh pulang ga, dok			

🙄?			
F13 - Daftar Check-Up			
F14 - Antrian Saya!	V	V	V
F15 - Minum Obat	V	V	V
F16 - Minum Penawar			
F17 - Exit	V	V	X
B05 - Dead or Alive?! 💀			
F09 - Tambah Dokter			

BAB V
DESAIN *COMMAND* UNTUK SETIAP PRIMITIF

BAB VI

DESAIN KAMUS DATA

BAB VII
DESAIN DEKOMPOSISI ALGORITMIK DAN FUNGSIONAL
PROGRAM

BAB VIII

SPESIFIKASI UNTUK TIAP MODUL/PROSEDUR/FUNGSI YANG DIBUAT (DALAM NOTASI ALGORITMIK)

1. F00 - Rencana Implementasi

1.1. ADT Sederhana

```
type User:<id: integer,  
           username      : string,  
           password      : string,  
           role          : string,  
           suhu_tubuh    : float,  
           tekanan_darah_sistolik : integer,  
           tekanan_darah_diastolik: integer,  
           detak_jantung : integer,  
           saturasi_oksigen: integer,  
           kadar_gula_darah: integer,  
           berat_badan   : float,  
           tinggi_badan  : float,  
           kadar_kolesterol: integer,  
           kadar_kolesterol_ldl: integer,  
           trombosit     : integer>
```

1.2. ADT Set

```
constant MAX_SET_SIZE: integer = 1000  
constant MAX_STRING_LENGTH: integer = 512  
  
type Set:  
    <elements: array [0..MAX_SET_SIZE] of  
    array [0..MAX_STRING_LENGTH] of string,  
    size: integer>  
  
procedure createEmptySet(input/output s:set)  
  
function contains(input s:set, input str:string)  
-> integer  
  
procedure insert(input/output s: set, str:  
string)
```

2. F01 - Login

```
procedure login(input user_now: User)  
function getUserIndex(username: string, users: User  
array, user_count: integer) -> integer
```

```

function getUserData(filename: string, user_count:
pointer to integer) -> User
function getTrombosit(line: string) -> integer
function getKadarKolesterolLDL(line: string) ->
integer
function getKadarKolesterol(line: string) -> integer
function getTinggiBadan(line: string) -> float
function getSaturasiOksigen(line: string) -> integer
function getKadarGulaDarah(line: string) -> integer
function getBeratBadan(line: string) -> float
function getSuhuTubuh(line: string) -> float
function getTekananDarahSistolik(line: string) ->
integer
function getTekananDarahDiastolik(line: string) ->
integer
function getDetakJantung(line: string) -> integer
function stringToInt(str: string) -> integer
function stringToFloat(str: string) -> float
function getUsername(line: string, username: string)
function getPassword(line: string, password: string)
function getRole(line: string, role: string)
function getRiwayatPenyakit(line: string,
riwayat_penyakit: string)
function getId(line: string) -> integer
function getFieldAt(line: string, field_number:
integer, output: string, max_length: integer)

```

3. F02 - Register

```

procedure RegisterUser()
procedure toLowerCase(input username: string, output
lowerUsername: string)
procedure appendUserToFile(input filename: string,
input/output user: User)
function isUsernameUnique(username : string, users:
Set,user_count: integer) -> integer

```

4. F03 - Logout

```

type LogType: integer
function logout(status : LogType) -> LogType

```

5. F04 - Lupa Password

```

function generateUniqeCode(username : string) ->
string
function validasiCode(username : string, inputCode :
string) -> string

```

- ```

procedure overWriteUserFile(users : User, count :
integer, filename : string) -> string
procedure gantiPassword (username : string,
newPassword : string) -> string
procedure lupaPassword()

```
6. F05 - Help
- ```

function help(userNow: User) -> string
procedure print_helpMessage_pasien(input usernow:
User)
procedure print_helpMessage_dokter(input usernow:
User)
procedure print_helpMessage_manager(input usernow:
User)

```
7. F07 - Lihat user
- ```

procedure urutanSort(input users : User, user_count :
integer)
procedure sortDescID(input users : User, user_count :
integer)
procedure sortAscID(input users : User, user_count :
integer)
procedure sortDescNama(input users : User, user_count
: integer)
procedure sortAscNama(input users : User, user_count :
integer)
procedure lihatUser(input users : User, usernow :
User)
procedure lihatPasien(input users : User, usernow :
User)
procedure lihatDokter(input users : User, usernow :
User)
procedure interfaceUser(usernow : User)
procedure output(usernow : User)

```
8. F08 - Cari User
- ```

procedure printHorizontalLine()
function toLowerChar(c: char) -> char
function strToLower(src: string, dest: string)
function compareUsername(a: string, b: string) ->
integer
function binarySearchUsername(users: User array,
userCount: integer, target: string) -> integer
procedure printHeader()
procedure printUser_general(U: User)

```

```
procedure printPasien(U: User)
procedure printPilihan()
procedure printDokter(U: User)
procedure cariPasien()
procedure cariDokter()
procedure cariUser()
```

9. F14 - Antrian Saya!

```
procedure daftarPasien(input nama: string, input
dokter: string, input ruangan: string)
procedure cekAntrian(input nama: string)
```

10. F15 - Minum Obat

```
procedure daftarObat(input list: Obat array, input
jumlah: integer)
procedure minumObat(input list: Obat array, input
nomorObat: integer)
```

BAB IX

TANGKAPAN LAYAR HASIL PENGUJIAN PROGRAM

BERDASARKAN FITUR-FITUR PADA SPESIFIKASI

1. data masukan
2. data keluaran untuk input yang valid, tidak valid jika terdapat validasi atau kasus-kasus lainnya.

BAB X

LAMPIRAN

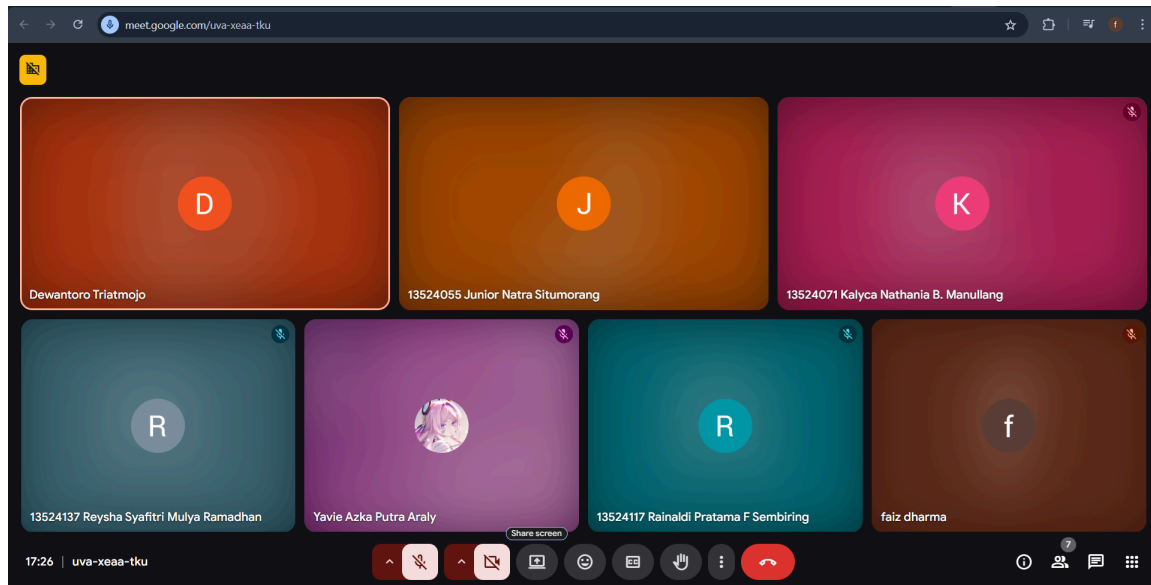
Nomor Asistensi : 1
 No. Kelompok/Kelas : L/01
 Tanggal asistensi : Jumat, 9 Mei 2025

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	13524055 / Junior Natra Situmorang
	2	13524071 / Kalyca Nathania Benedicta Manullang
	3	13524077 / Yavie Azka Putra Araly
	4	13524097 / Muhammad Faiz Alfada Dharma
	5	13524117 / Rainaldi Pratama F. Sembiring
	6	13524137 / Reysha Syafitri Mulya Ramadhan
Asisten pembimbing	NIM / Nama	
	13522011 / Dewantoro Triatmojo	

Catatan Asistensi:

Rangkuman Diskusi
<ol style="list-style-type: none"> 1. Penulis disarankan untuk melakukan <i>merge</i> modul atau kode yang sudah selesai secara berkala agar tidak kesulitan saat menyatukan program utama di tahap akhir. 2. Saat ini pengembangan masih dilakukan di Windows. Penulis disarankan menggunakan WSL (Windows Subsystem for Linux) karena beberapa fungsi di C memiliki perbedaan implementasi di Linux dan Windows. WSL digunakan sebagai lingkungan pengembangan yang mendekati sistem target (Linux) sehingga lebih aman untuk <i>compile</i> dan menjalankan program. 3. Untuk struktur program, saat ini fitur-fitur masih berdiri sendiri dan belum disatukan dalam program utama. Penulis ditekankan pentingnya manajemen <i>state</i> program agar lebih terstruktur dan dapat dikembangkan bersama secara efisien. 4. Penulis disarankan menggunakan pendekatan <i>pass by reference</i> ke fungsi dengan variabel global atau <i>struct</i> sebagai media penyimpanan <i>state</i>. Untuk manajemen <i>state</i> program, penggunaan <i>global state</i> memudahkan proses pengujian dan pengembangan karena lebih terprediksi. 5. Untuk strategi pengelolaan variabel, penulis disarankan membuat satu <i>struct</i> global berisi seluruh variabel penting (misalnya daftar <i>user</i>, dokter, dan lain-lain). <i>Struct</i> ini dapat diparsing ke fungsi-fungsi yang membutuhkan tanpa harus memanggil banyak variabel satu per satu. Alternatif lain adalah menggunakan pendekatan <i>extern</i>, tetapi hal ini memerlukan kehati-hatian dalam penulisan dan implementasi.
Tindak Lanjut
<ol style="list-style-type: none"> 1. Mulai menggunakan WSL untuk proses pengembangan dan kompilasi program. 2. Segera menyatukan fitur-fitur ke dalam program utama secara bertahap dan rutin melakukan <i>merge</i>. 3. Membuat satu <i>struct</i> global yang berisi semua variabel penting (<i>global state</i>). 4. Menerapkan manajemen <i>state</i> menggunakan pendekatan <i>pass by reference</i> terhadap <i>struct</i> tersebut ke fungsi-fungsi terkait. 5. Membagi tugas pengembangan agar terkoordinasi dan terintegrasi dengan manajemen <i>state</i> yang jelas.

Dokumentasi



Gambar 10.1 Asistensi 1