

LAPORAN TUGAS BESAR

IF2110 Algoritma dan Struktur Data

Burung Biru



Dipersiapkan oleh kelompok IF2110_TB_A:

Yusuf Ardian Sandi 13522015

Kristo Anugrah 13522024

Dhidit Abdi Aziz 13522040

Farel Winalda 13522047

Erdianti Wiga Putri Andini 13522053

Andhita Naura Hariyanto 13522060

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 40132

Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
	IF2110-TB-A-01		
	Revisi	1	24/11/2023

Daftar Isi

1 Ringkasan.....	3
2 Penjelasan Tambahan Spesifikasi Tugas.....	5
2.1 Nomor HP Unlimited.....	5
2.2 FYB (For Your BurBir).....	5
2.3 PROFIL dan PENGGUNA.....	5
2.4 COMMAND.....	5
3 Struktur Data (ADT).....	7
3.1 ADT Sederhana.....	7
3.2 ADT Boolean.....	7
3.3 ADT Mesin Karakter.....	7
3.4 ADT Mesin Kata.....	8
3.5 ADT Mesin Konfigurasi.....	8
3.6 ADT Time.....	8
3.7 ADT Datetime.....	9
3.8 ADT Stack.....	9
3.9 ADT Matriks.....	11
3.10 ADT Graf dengan Representasi Adjacency Matriks.....	12
3.11 ADT List dengan Struktur Data Array Dinamik.....	13
3.12 ADT List dengan Struktur Data Berkait.....	15
3.13 ADT List dengan Struktur Data Array Statik.....	16
3.14 ADT Priority Queue dengan Struktur Data Berkait.....	18
3.15 ADT Tree.....	19
4 Program Utama.....	21
5 Algoritma-Algoritma Menarik.....	22
5.1 Depth First Search (DFS).....	22
6 Data Test.....	23
6.1 Inisialisasi.....	23
6.2 Pengguna.....	23
6.3 Profil.....	25
6.4 Teman.....	27
6.5 Permintaan Pertemanan.....	29
6.6 Kicauan.....	30
6.7 Balasan.....	33
6.8 Draf Kicauan.....	34

6.9 Utas.....	36
6.10 Simpan dan Muat.....	41
7 Test Script.....	44
8 Pembagian Kerja dalam Kelompok.....	56
9 Lampiran.....	57
9.1 Deskripsi Tugas Besar 1.....	57
9.2 Notulen Rapat.....	58
9.3 Log Activity Anggota Kelompok.....	61
9.4 Kendala.....	63
9.5 Milestone.....	63
9.6 Form Asistensi.....	68

1 Ringkasan

BurBir merupakan program buatan kelompok A dari K01 IF2110 Algoritma dan Struktur Data untuk menyelesaikan permasalahan pembuatan simulator aplikasi sosial media berbasis *command-line interface (CLI)* yang memanfaatkan ADT dengan menggunakan bahasa C. Program ini sebagai bentuk simulasi pengoperasian suatu sosial media. Pengimplementasian ADT primitif menjadi fitur aplikasi memerlukan beberapa aksi modifikasi untuk menyesuaikan kebutuhan fitur. Fitur-fitur yang terdapat pada program ini sesuai dengan spesifikasi antara lain adalah pengguna, profil, pertemanan, permintaan pertemanan, kicauan, balasan, draf kicauan, utas, serta simpan dan muat.

Laporan Tugas Besar IF2110 Algoritma dan Struktur Data ini berisi penjelasan spesifikasi, ADT, program, serta bukti implementasi dan pengetesan program Burbir. Penjelasan spesifikasi meliputi spesifikasi wajib dengan informasi lebih detail sesuai dengan program kami. ADT berisi struktur-struktur data primitif yang kami gunakan. ADT tersebut juga kami modifikasi sedemikian rupa agar dapat digunakan pada program ini. Terakhir, Algoritma-Algoritma menarik adalah untuk mencatat algoritma-algoritma yang sekiranya kami anggap menarik.

Secara umum, kami telah menerapkan konsep-konsep yang diajarkan dalam mata kuliah Algoritma dan Struktur Data selama satu semester ke dalam pembuatan sebuah permainan. Program ini dapat dijalankan dengan menggunakan sejumlah perintah yang telah ditentukan sebelumnya. Kami meyakini bahwa penyelesaian proyek besar ini telah memberikan kontribusi signifikan dalam pemahaman dan implementasi berbagai ADT yang telah kami pelajari, mengubah pengetahuan teoritis menjadi pengalaman praktis.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Nomor HP Unlimited

Program kami menyimpan data nomor HP pengguna dengan menggunakan ADT List Dinamis. Untuk melakukan penyimpanan, kami akan menghapus semua nomor yang tersimpan dengan fungsi *deletelast* lalu menjalankan fungsi *inserilast* untuk memasukkan nomor yang baru. Fungsi *insertlast* dan *deletelast* ini akan otomatis menyesuaikan kapasitas yang dibutuhkan sehingga bisa memuat nomor yang unlimited.

2.2 FYB (For Your BurBir)

Kami berhasil membuat fitur FYB, tetapi tidak menggunakan ADT Max Heap. Kami menggunakan konsep sorting biasa dengan algoritma *bubble sorting*. Walaupun demikian, fitur FYB ini dapat berjalan dengan lancar untuk menampilkan kicauan dengan *like* terbanyak.

2.3 PROFIL dan PENGGUNA

Pada seluruh perintah yang terdapat pada program profil dan pengguna, ditambahkan pemberitahuan kepada pengguna jika nama atau password yang dimasukkan melebihi 20 karakter. Selain itu, untuk semua perintah pada program profil, akan divalidasi apakah user sudah melakukan login atau belum dan jika belum akan diberikan sebuah pesan untuk melakukan login terlebih dahulu. Pada spesifikasi LIHAT_PROFIL, tidak terdapat kasus dimana nama user mengandung spasi sehingga kami membuat parsing saat pembacaan perintah LIHAT_PROFIL <nama> untuk memeriksa validitas masukkan *command* dengan menganggap semua *char* setelah spasi ialah variabel nama.

2.4 COMMAND

Program dapat memeriksa validitas jumlah kata yang dimiliki untuk setiap perintah yang masuk. Perintah-perintah dikategorikan terlebih dahulu berdasarkan jumlah kata yang seharusnya dimiliki oleh perintah tersebut. Apabila jumlah kata yang dimasukkan tidak sesuai dengan jumlah kata yang seharusnya, program akan mengembalikan pesan error dan meminta masukan

STEI- ITB	IF2110-TB-A-01	Halaman 5 dari 68 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

ulang dari pengguna hingga jumlah kata yang dimasukkan oleh pengguna sesuai dengan keperluan setiap perintah.

STEI- ITB	IF2110-TB-A-01	Halaman 6 dari 68 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

3 Struktur Data (ADT)

3.1 ADT Sederhana

Hampir seluruh fungsi dalam Program Burbir ini menggunakan ADT Sederhana baik itu secara langsung ataupun setelah melalui modifikasi. Sebagai tipe bentukan paling kecil, ADT Sederhana bermanfaat untuk menyimpan data-data yang sifatnya statis dan eksak. Pengimplementasian ADT Sederhana terdapat pada:

- a. Boolean
- b. Time
- c. Datetime
- d. Char (Mesin Karakter)
- e. Word (Mesin Kata)
- f. ID Pengguna (ADT Pengguna)
- g. Weton (ADT Pengguna)
- h. ID Follow (ADT Priority Queue)
- i. dan lain-lain

3.2 ADT Boolean

Nama file header: boolean.h

Tidak terdapat tipe data bawaan bahasa C yang merepresentasikan nilai kebenaran secara intuitif. Oleh karena itu, didefinisikan nilai benar (1) sebagai “true”, dan nilai salah (0) sebagai “false”. ADT ini sangat bermanfaat untuk setiap fungsi pengecekan dengan ciri berawalan is-XXX, misalkan isEmpty, isFriend, dan lain-lain.

3.3 ADT Mesin Karakter

Nama file header: charmachine.h

Nama file realisasi: charmachine.c

ADT mesin karakter adalah ADT yang dibuat sebagai basis dalam menerima input dari pengguna. ADT ini digunakan untuk membaca suatu input dari terminal. ADT ini berkorelasi dengan ADT mesin kata dan diimplementasikan dalam mesin kata. Kami menggunakan ADT ini sebagai basis pemrosesan input pada program kami.

STEI- ITB	IF2110-TB-A-01	Halaman 7 dari 68 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

3.4 ADT Mesin Kata

Nama file header: wordmachine.h

Nama file realisasi: wordmachine.c

ADT mesin kata adalah bentuk pengimplementasian ADT mesin karakter dalam pembacaan input dari pengguna. Mesin kata yang kami gunakan adalah mesin kata model akuisisi versi I. Kami menggunakan versi ini karena kami men-*define* suatu MARK yaitu ; sebagai akhir dari proses input.

3.5 ADT Mesin Konfigurasi

Nama file header: configmachine.h

Nama file realisasi: configmachine.c

ADT mesin konfigurasi juga merupakan bentuk pengimplementasian dalam pembacaan input. Namun bedanya ADT ini membaca input dari suatu file, bukan dari input pengguna.

3.6 ADT Time

Nama file header: time.h

Nama file realisasi: time.c

ADT Time

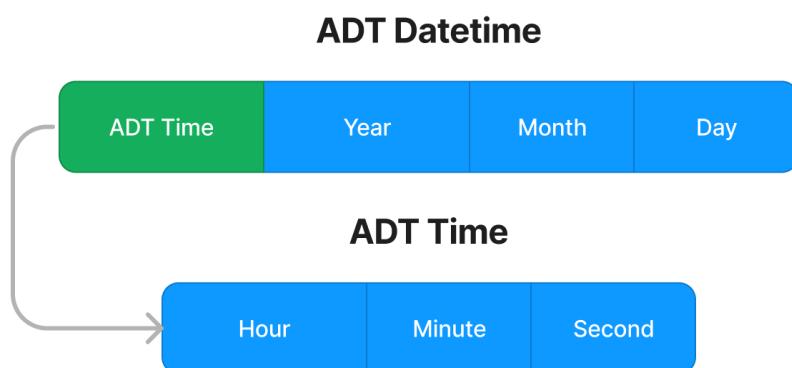


ADT time berfungsi untuk merepresentasikan waktu dalam simulator serta memproses suatu struktur waktu yang terdiri dari jam, menit, dan detik. Program-program pada file ini berkaitan dengan pembacaan waktu, penulisan waktu, dll. Kami menggunakan ADT ini pada fitur-fitur yang perlu menampilkan waktu.

3.7 ADT Datetime

Nama file header: datetime.h

Nama file realisasi: datetime.c



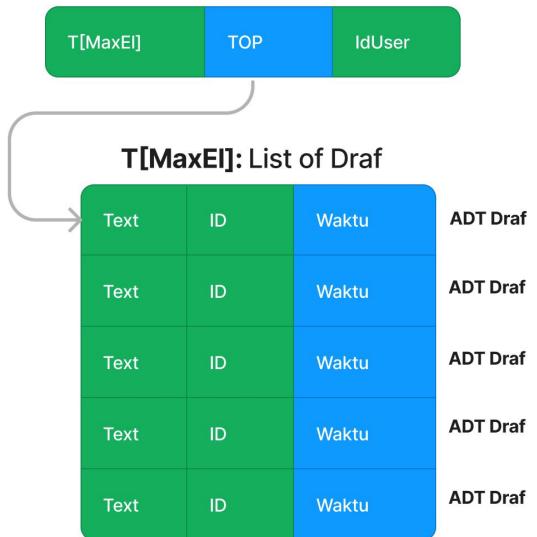
ADT datetime merupakan implementasi lanjutan dari ADT time. Pada ADT ini, elemen waktu diperbanyak menjadi tanggal, bulan, tahun, jam, menit, dan detik. Program-program pada file ini berkaitan dengan pembacaan waktu, penulisan waktu, dll. Kami menggunakan ADT ini pada fitur-fitur yang perlu menampilkan waktu.

3.8 ADT Stack

Nama file header: draf.h

Nama file realisasi: draf.c

DrafStack: Stack of ADT Draf (Terurut)



Seorang pengguna dapat menulis kicauan tanpa segera dikirimkan yaitu dengan fitur draf. Apabila terdapat lebih dari satu draf, maka draf yang dapat diedit, dihapus, dan dirilis hanyalah draf yang paling terbaru dibuat. Dengan demikian, dapat disimpulkan bahwa draf memiliki sifat Last In First Out (LIFO). Untuk merealisasikan draf, dibutuhkan struktur data yang juga memiliki sifat LIFO, yaitu stack. Dari sekian banyaknya tipe representasi stack, dipilih metode representasi dengan List Linier. Alasan pemilihan:

- a. Draf merupakan fungsi yang tidak digunakan sesering berkicau dan membalas, sehingga tidak memerlukan representasi yang memiliki ukuran tak terbatas
- b. Draf dibatasi jumlahnya agar pengguna tidak terus menerus membuat draf tanpa merilisnya sehingga ketika kapasitas penyimpanan draf sudah penuh, pengguna dipaksa untuk merilis drafnya sebelum menambahkan draf kembali

ADT Drafstack tersebut tergabung ke dalam ADT Pengguna, sehingga tiap pengguna memiliki Stack of Drafnya masing masing. ADT Drafstack terdiri atas:

- a. T[MaxEl]: Tabel penyimpanan draft dengan batasan kapasitas sebanyak MaxEl
- b. TOP: Address (berupa indeks) menuju draf paling atas
- c. IdUser: Id User yang membuat draf

T[MaxEl] terdiri atas:

- a. Text (ADT Word)
- b. ID Draf (Eltype)
- c. Waktu (ADT Datetime)

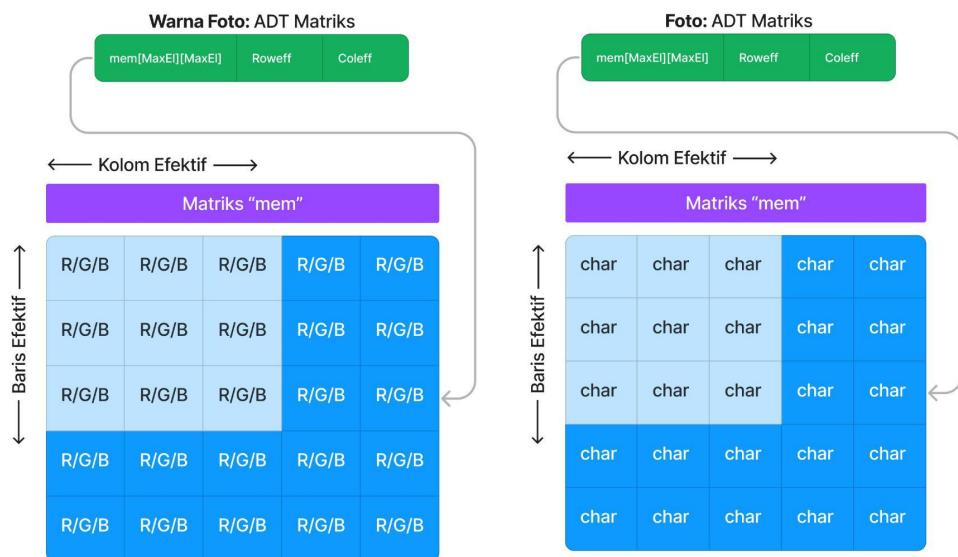
Struktur data ini bermanfaat untuk menyelesaikan segala permasalahan yang terkait dengan menulis draf, yaitu:

- a. Membuat Draf
- b. Melihat Draf
- c. Merilis Draf

3.9 ADT Matriks

Nama file header: matrix.h

Nama file realisasi: matrix.c



Pada spesifikasi tugas besar, secara eksplisit tertulis bahwa foto profil beserta warnanya memiliki tipe matriks. ADT Matriks tersebut terdiri atas:

- a. Mem[MaxEl][MaxEl] dengan tipe ADT Sederhana (karakter)
- b. Panjang baris efektif
- c. Panjang kolom efektif

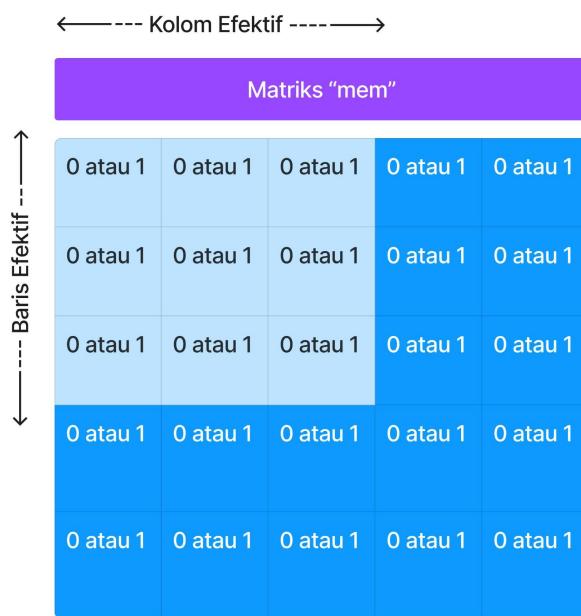
Tipe data ini bermanfaat untuk menyelesaikan permasalahan seperti:

- Memasukkan foto profil
- Mengubah foto profil
- Melihat foto profil

3.10 ADT Graf dengan Representasi Adjacency Matriks

Nama file header: friendmatrix.h

Nama file realisasi: friendmatrix.c



Matriks pertemanan sejatinya merupakan ADT GRAF dengan representasi Adjacency Matrix untuk menyimpan informasi pertemanan. ADT ini terdiri atas:

- Matriks yang berisi elemen integer bernilai 0 atau 1, dengan maksimal ukuran 20x20
- Jumlah baris efektif
- Jumlah kolom efektif

ADT Graf dengan representasi Adjacency Matrix dipilih karena struktur ini memudahkan pengecekan jumlah teman yang dimiliki seorang user. Hal ini memungkinkan apabila kita menganggap indeks baris dan kolom dari matriks tersebut merupakan spesifik informasi

pertemanan dari suatu user dengan id = indeks + 1 (indeks matriks dimulai dari 0 sedangkan id pengguna dimulai dari 1. Untuk menghitung jumlah teman, kita tutup menjumlahkan nilai dari suatu baris dengan mengabaikan satu kolom diagonal. Hal serupa juga berlaku ketika kita ingin mengakses id-id user mana sajakah yang berteman dengan kita.

Penggunaan Struktur ADT Graf sangat bermanfaat dalam menyelesaikan segala persoalan yang berkaitan dengan pertemanan, sebagai contoh:

- Melihat daftar teman yang dimiliki seseorang
- Menghapus teman
- Menambahkan teman
- Menjadi parameter atas terlihat atau tidaknya kicauan seseorang di beranda

3.11 ADT List dengan Struktur Data Array Dinamik

Nama file header: listdinkicauan.h, listdin.h

Nama file realisasi: listdinkicauan.c, listdin.c



Modul ADT List Dinamis Kicauan merupakan modifikasi ADT List Dinamis yang digunakan untuk menyimpan informasi data kicauan serta melakukan operasi terhadap list dinamis kicauan.

List Dinamis Kicauan menyimpan struktur data bentukan Kicauan yang terdiri atas :

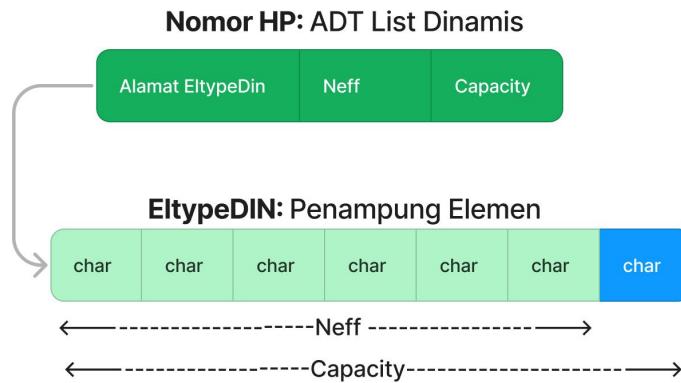
- a. ID Kicauan (ADT Sederhana)
- b. Teks Kicauan (ADT Sederhana)
- c. Like (ADT Sederhana)
- d. ID Pengguna (ADT Sederhana)
- e. Waktu (ADT Datetime)
- f. Address Utas (ADT Linked List Utas)
- g. Address Balasan (ADT Tree Balasan)

Pada dasarnya, seorang pengguna dapat menuliskan kicauan sebanyak mungkin (tidak terbatas). Struktur data tersebut sebaiknya juga memiliki slot kosong yang dapat menjadi jaminan bahwa akan selalu tersedia ruangan untuk menambahkan kicauan. Maka dari itu, kami memilih ADT List Dinamis sebagai struktur data karena memiliki karakteristik yang sesuai untuk kebutuhan tersebut. ADT List Dinamis ini akan selalu ditambahkan kapasitasnya setiap kicauan yang sudah ada memenuhi suatu batasan tertentu.

List Dinamis Kicauan sangat bermanfaat untuk menyelesaikan persoalan-persoalan seperti:

- a. Menulis kicauan
- b. Menyukai Kicauan
- c. Memulai utas
- d. Membalas kicauan

ADT List Dinamis juga digunakan dalam merepresentasikan nomor HP pengguna. Berikut ini ilustrasinya:



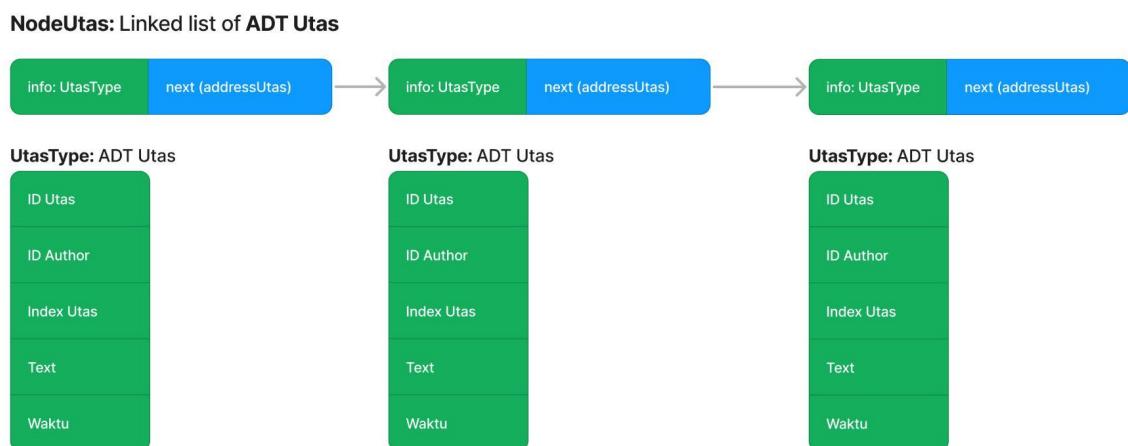
Nomor HP terdiri atas:

- a. EltypeDin
Merupakan alamat ke penampung elemen. Setiap elemennya bertipe karakter. Tiap digit dari nomor HP yang diinput pengguna akan dianggap sebagai karakter.
- b. Jumlah elemen efektif
- c. Kapasitas maksimal penampung elemen

3.12 ADT List dengan Struktur Data Berkait

Nama file header: listlinierutas.h

Nama file realisasi: listlinierutas.c



List Linier Utas merupakan List Berkait antar ADT Utas, yang terdiri atas info (ADT Utas), dan alamat ke list linier selanjutnya. Struktur berkait dipilih karena utas sebelum ke utas sesudahnya harus saling berhubungan sehingga diperlukan struktur yang dapat mengakomodasi perpindahan dengan cepat. ADT Utas terdiri atas:

- a. ID Utas (Eltype)
- b. ID Authr (Eltype)
- c. Index Utas (Eltype)
- d. Text (Word)
- e. Waktu (ADT Datetime)

Struktur data ini bermanfaat ketika menyelesaikan permasalahan:

- a. Memulai utas baru
- b. Menyambung utas sebelumnya
- c. Menghapus utas
- d. Mencetak utas

3.13 ADT List dengan Struktur Data Array Statik

Nama file header: liststatikuser.h

Nama file realisasi: liststatikuser.c

List Pengguna: List of ADT Pengguna (Max 20)											
ADT Pengguna	ID Pengguna/Index	Nama	Sandi	Bio	Nomor HP	Weton	Tipe Akun	Foto Profil	Warna foto	Request Follow	Draf
ADT Pengguna	ID Pengguna/Index	Nama	Sandi	Bio	Nomor HP	Weton	Tipe Akun	Foto Profil	Warna foto	Request Follow	Draf
ADT Pengguna	ID Pengguna/Index	Nama	Sandi	Bio	Nomor HP	Weton	Tipe Akun	Foto Profil	Warna foto	Request Follow	Draf
ADT Pengguna	ID Pengguna/Index	Nama	Sandi	Bio	Nomor HP	Weton	Tipe Akun	Foto Profil	Warna foto	Request Follow	Draf
ADT Pengguna	ID Pengguna/Index	Nama	Sandi	Bio	Nomor HP	Weton	Tipe Akun	Foto Profil	Warna foto	Request Follow	Draf
ADT Pengguna	ID Pengguna/Index	Nama	Sandi	Bio	Nomor HP	Weton	Tipe Akun	Foto Profil	Warna foto	Request Follow	Draf

Untuk merepresentasikan setiap akun pengguna, kami menggunakan struktur data bentukan ADT sederhana bernama Pengguna yang terdiri atas:

- a. ID Pengguna (ADT Sederhana)

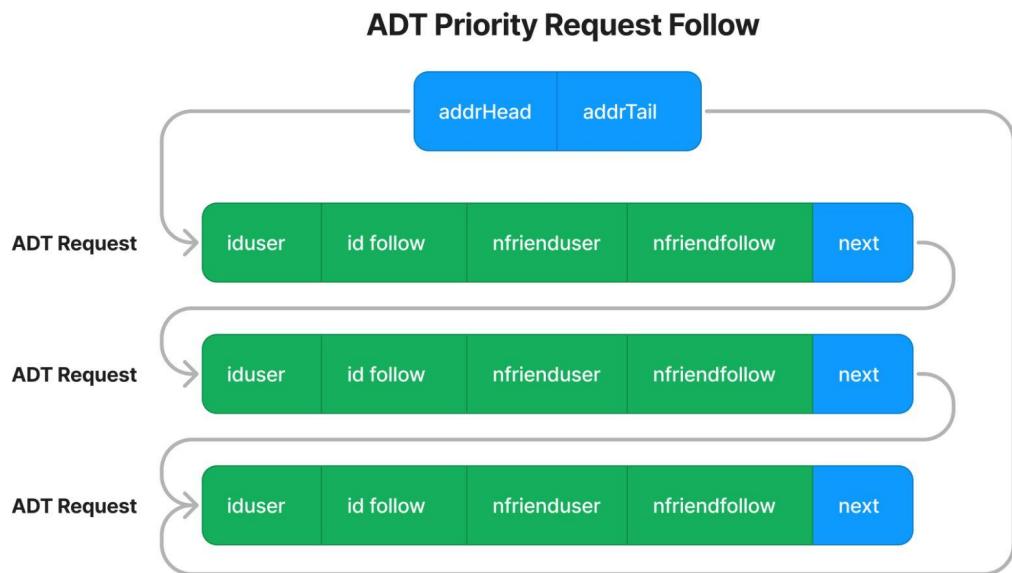
- b. Nama (ADT Sederhana)
- c. Sandi (ADT Sederhana)
- d. Bio (ADT Sederhana)
- e. Nomor HP (ADT List Dinamis)
- f. Weton (ADT Sederhana)
- g. Tipe Akun (ADT Sederhana)
- h. Foto Profil (ADT Matrix)
- i. Warna Foto (ADT Matrix)
- j. Request Follow (ADT Priority Queue)
- k. Draft (ADT Stack)

ADT Pengguna ini disusun menjadi sebuah List Statis bernama ADT List Statis Pengguna. Berdasarkan spesifikasi tugas besar, tertulis bahwa jumlah pengguna maksimal adalah 20 pengguna. Dengan demikian, diperlukan suatu struktur data yang memiliki jumlah tetap, dan mudah diakses dengan cepat, yaitu List Statis.

Struktur data ini bermanfaat dalam menyelesaikan persoalan yang berkaitan dengan data pengguna dan akun:

- a. Pendaftaran akun
- b. Keluar/masuk akun
- c. Mengganti foto profil
- d. Mengganti bio akun
- e. Mengganti nomor HP
- f. Mengganti Weton
- g. Melihat profil seseorang
- h. Mengatur privasi akun

3.14 ADT Priority Queue dengan Struktur Data Berkait



Nama file header: prioreqfollinked.h

Nama file realisasi: prioreqfollinked.c

ADT Priority Queue Linked terdiri atas alamat head dan alamat tail. Kedua elemen tersebut merupakan alamat kepada struktur ADT Request. ADT Request terdiri atas:

- a. iduser
Id akun yang melakukan permintaan pertemanan
- b. idfollow
Id akun yang diminta iduser untuk berteman
- c. nfrienduser
Jumlah teman akun dengan id iduser
- d. nfriendfollow
Jumlah teman akun dengan id idfollow
- e. next
Alamat ke ADT Request selanjutnya

ADT Priority Queue Linked sebenarnya merupakan ADT Queue tetapi dengan beberapa modifikasi. Modifikasi pertama adalah diterapkannya penerapan skala prioritas dalam mengurutkan elemen pada queue. Pada kasus ini, request dari akun dengan jumlah teman paling banyak (paling populer) akan berada pada paling atas atau head. Begitupun sebaliknya, permintaan pertemanan dari akun yang jumlah temannya paling sedikit (paling tidak populer), akan berada pada paling bawah atau tail. Modifikasi kedua adalah diterapkannya struktur berkait antara satu request kepada request lainnya. Modifikasi ini memudahkan implementasi skala prioritas yang telah dijelaskan sebelumnya, yaitu ketika menyisipkan suatu request pada tengah ataupun akhir Queue. Dengan struktur berkait, melakukan penyisipan elemen baik itu di depan, tengah, ataupun belakang sangatlah mudah, tanpa perlu melakukan pergeseran elemen-elemen lainnya.

Struktur data Priority Queue sangat bermanfaat untuk menyelesaikan permasalahan seperti:

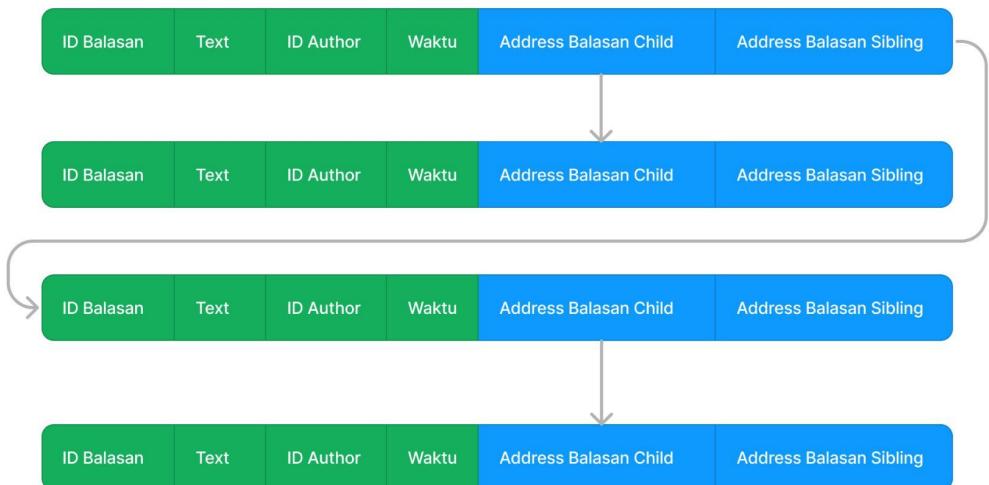
- a. Menambahkan teman
- b. Melihat daftar permintaan pertemanan orang lain kepada kita
- c. Menyetujui permintaan pertemanan orang lain kepada kita

3.15 ADT Tree

Nama file header: treebalasan.h

Nama file realisasi: treebalasan.c

ADT Tree Balasan



Struktur balasan terbilang cukup rumit. Suatu kicauan dapat memiliki lebih dari satu balasan, sedangkan suatu balasan juga dapat memiliki satu atau lebih balasan di saat yang bersamaan. Untuk mengakomodasi kebutuhan ini, dipilihlah struktur Binary Tree dengan sedikit modifikasi. Simpul kiri diganti menjadi balasan dari balasan (child), sedangkan simpul kanan diganti menjadi balasan yang setara (sibling). Struktur Tree Balasan terdiri atas:

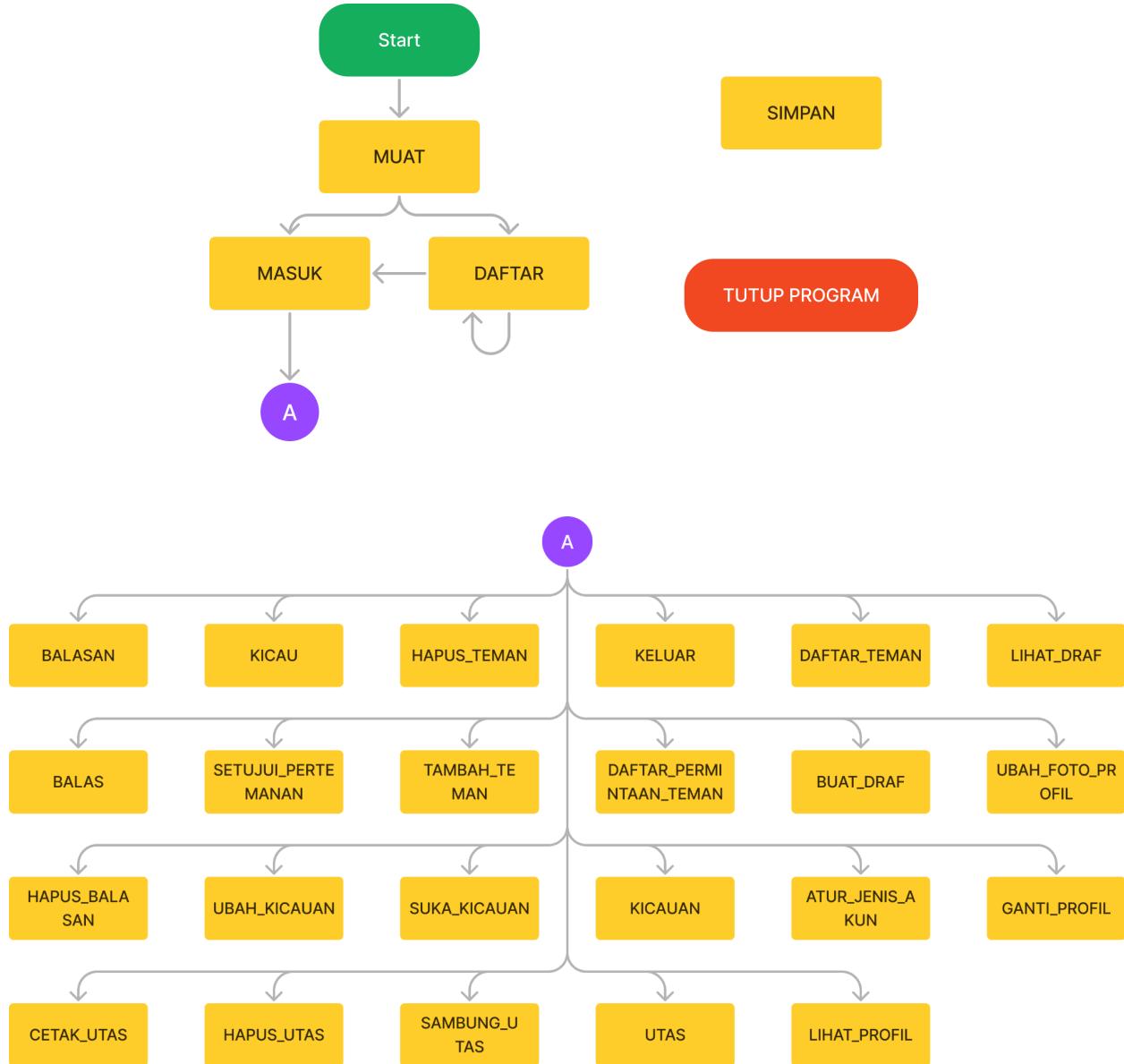
- ID Balasan
- Text (ADT Word)
- ID Author
- Waktu (ADT Datetime)
- Address Balasan Child(Address to ADT Balasan)
- Address Balasan Child(Address to ADT Balasan)

Struktur data Tree Balasan ini sangat bermanfaat untuk menyelesaikan persoalan:

- Membalas kicauan
- Melihat daftar balasan pada suatu kicau
- Menghapus balasan

4 Program Utama

Berikut alur dari program utama kami pada main.c:



Program utama kami dimulai dengan pemuatan file config yang dipilih oleh pengguna dan selanjutnya pengguna dapat menjalankan perintah DAFTAR ataupun MASUK. Namun, semua fitur BurBir hanya dapat diakses apabila pengguna telah memanggil perintah MASUK. Fitur

BurBir yang kami sediakan lengkap mulai dari program Pengguna, Profil, Teman, Permintaan Pertemanan, Kicauan, Balasan, Draf Kicauan, Utas, hingga Simpan dan Muat. Pada gambar yang tertera diatas, terlihat bahwa SIMPAN dan TUTUP_PROGRAM berdiri sendiri tanpa garis penunjuk. Hal ini dimaksudkan karena perintah SIMPAN dan TUTUP_PROGRAM ini dapat diakses dalam kondisi kapanpun setelah pengguna melakukan Muat pertama kali.

5 Algoritma-Algoritma Menarik

5.1 Depth First Search (DFS)

Algoritma ini adalah algoritma pencarian berbasis rekursi yang banyak digunakan di Abstract Data Type Tree dan Graph. Pada tugas besar ini, algoritma DFS ini digunakan pada ADT Tree di fitur balasan untuk mencari balasan yang ingin dibalas. Algoritma ini menarik karena dapat melakukan pencarian pada struktur data berkait (seperti linked-list, tree, dan graph), secara efisien dengan kompleksitas waktu $O(n)$.

6 Data Test

6.1 Inisialisasi

Fitur (dalam bentuk perintah)	Screenshot	Hasil yang diharapkan atau Keterangan lainnya
Fitur Inisialisasi	 <p>Selamat datang di BurBir. Aplikasi untuk studi kualitatif mengenai perilaku manusia dengan menggunakan metode (pengambilan data berupa) Focused Group Discussion kedua di zamannya. Silahkan masukan folder konfigurasi untuk dimuat: config-1; File konfigurasi berhasil dimuat! Selamat berkicau!</p>	Melakukan pemuatan sesuai dengan file config yang dimasukkan pengguna.

6.2 Pengguna

Fitur (dalam bentuk perintah)	Screenshot	Hasil yang diharapkan atau Keterangan lainnya
DAFTAR;	<pre>>> DAFTAR; Masukkan nama: Yati; Masukkan kata sandi: Aku suka Waluh Kukus; Pengguna telah berhasil terdaftar. Masuk untuk menikmati fitur-fitur BurBir. >> MASUK; Masukkan nama: Yati; Masukkan kata sandi: Aku suka Waluh Kukus; Anda telah berhasil masuk dengan nama pengguna Yati. Mari menjelajahi BurBir bersama Ande-Ande Lumut!</pre>	Program berhasil menyimpan data nama dan kata sandi yang baru sehingga pengguna bisa melakukan MASUK dengan nama dan kata sandi tersebut

DAFTAR;	<pre>>> DAFTAR; Masukkan nama: Tuan Hak; Wah, sayang sekali nama tersebut telah diambil.</pre> <p style="text-align: center;">Gambar 6.2.2</p>	Akan ditampilkan pesan nama telah diambil jika terdapat akun lain yang memiliki nama akun yang sama dengan nama masukkan.
MASUK;	<pre>>> MASUK; Masukkan nama: Tuan hak; Wah, nama yang Anda cari tidak ada. Masukkan nama lain! Masukkan nama: Tuan Hak; Masukkan kata sandi: salah password; Wah, kata sandi yang Anda masukkan belum tepat. Periksa kembali kata sandi Anda ! Masukkan kata sandi: passTuanHak; Anda telah berhasil masuk dengan nama pengguna Tuan Hak. Mari menjelajahi BurBir bersama Ande-Ande Lumut! >> MASUK; Wah Anda sudah masuk. Keluar dulu yuk! >> █</pre> <p style="text-align: center;">Gambar 6.2.3</p>	Pengguna dapat berhasil melakukan MASUK dengan syarat nama dan kata sandi tepat dan sesuai dengan data yang tersimpan. Jika tidak, akan diberikan pesan kesalahan
KELUAR;	<pre>>> KELUAR; Anda belum login! Masuk terlebih dahulu untuk menikmati layanan Burbir. >> MASUK; Masukkan nama: Tuan Hak; Masukkan kata sandi: passTuanHak; Anda telah berhasil masuk dengan nama pengguna Tuan Hak. Mari menjelajahi BurBir bersama Ande-Ande Lumut! >> KELUAR; Anda berhasil logout. Sampai jumpa di pertemuan berikutnya!</pre> <p style="text-align: center;">Gambar 6.2.4</p>	Pengguna dapat melakukan KELUAR dengan syarat pengguna sudah pernah melakukan MASUK
TUTUP_PROGRAM;	<pre>>> TUTUP_PROGRAM; Anda telah keluar dari program Burbir. Sampai jumpa di pertemuan berikutnya!</pre> <p style="text-align: center;">Gambar 6.2.5</p>	Pengguna bisa keluar dari program dengan perintah TUTUP_PROGRAM lalu program berhenti melakukan looping untuk meminta masukkan command

6.3 Profil

Fitur (dalam bentuk perintah)	Screenshot	Hasil yang diharapkan atau Keterangan lainnya
GANTI_PROFIL;	<pre>>> GANTI_PROFIL; Nama : Tuan Hak Bio Akun : Ini bio lho No HP : 081 Weton : Pahing Masukkan Bio Akun: Aku cinta tubes <3; Masukkan No HP: 08a a-; No HP tidak valid. Masukkan lagi yuk! Masukkan No HP: 0811111111; Masukkan Weton: Pahig; Weton yang dimasukkan tidak valid! Masukkan Weton: ; Profil Anda sudah berhasil diperbarui!</pre>	Saat pengguna melakukan GANTI_PROFIL, masukkan bio akun akan divalidasi dengan panjang kurang dari 136, masukkan No HP divalidasi hanya berupa angka, dan masukkan weton divalidasi antara Pahing, Kliwon, Legi, Pon, Wage, atau kosong (<i>case insensitive</i>).

Gambar 6.3.1

LIHAT_PROFIL;	<pre>>> LIHAT_PROFIL Tuan Vin; Nama : Tuan Vin Bio Akun : No HP : 08 Weton : Foto profil akun Tuan Vin ***** ***** ***** ***** *****</pre> <pre>>> LIHAT_PROFIL Tuan Bus; Wah, akun Tuan Bus diprivat nih. Ikuti dulu yuk untuk bisa melihat profil Tuan Bus!</pre> <pre>>> LIHAT_PROFIL Tuan Man; Nama : Tuan Man Bio Akun : AFFA IYAH No HP : 080129391223018777777737123208173012371273721377263640324 Weton : Foto profil akun Tuan Man ***** ***** ***** *****</pre>	Pengguna dapat melihat profil dari nama yang dituju jika nama yang dituju tersebut memiliki tipe akun publik. Namun, jika nama yang dituju memiliki tipe akun privat, maka akan ada pesan bahwa akun tersebut tidaklah publik
ATUR_JENIS_AKUN;	<pre>>> ATUR_JENIS_AKUN; Nama : Tuan Hak Bio Akun : Ini bio lho No HP : 081 Weton : Pahing Saat ini, akun Anda adalah akun Privat. Ingin mengubah ke akun Publik? (YA/TIDAK) YA; Akun anda sudah diubah menjadi akun Publik.</pre>	Pengguna dapat mengubah tipe akun dari publik ke privat dan sebaliknya

UBAH_FOTO_PR OFIL;	<pre>>> UBAH_FOTO_PROFIL; Foto profil Anda saat ini adalah ***** *@@@* *@@@* *@@@* ***** Masukkan foto profil yang baru R * R * R * R * R * R * G * G * G * R * R * G * B * G * R * R * G * G * G * R * R * R * R * R * R *; Foto profil Anda sudah berhasil diganti! >> LIHAT_PROFIL Tuan Hak; Nama : Tuan Hak Bio Akun : Ini bio lho No HP : 081 Weton : Pahing Foto profil akun Tuan Hak ***** *****</pre>	Pengguna dapat mengubah foto profil dan masukkan foto profil yang baru dapat tersimpan sehingga jika pengguna memanggil LIHAT_PROFIL, yang ditampilkan ialah foto profil yang baru
-------------------------------	--	--

6.4 Teman

Fitur (dalam bentuk perintah)	Screenshot	Hasil yang diharapkan atau Keterangan lainnya
DAFTAR_TEMA N;	<pre>>> DAFTAR_TEMAN; Tuan Hak memiliki 3 teman Daftar teman Tuan Hak Tuan Bri Tuan Bus Tuan Man</pre>	Program menampilkan daftar teman yang dimiliki user
DAFTAR_TEMA N;	<pre>>> DAFTAR_TEMAN; Tuan Vin belum mempunyai teman >> </pre>	Program menampilkan pesan bahwa user belum memiliki teman

DAFTAR_TEMA N;	<pre>>> DAFTAR TEMAN; Anda belum login! Masuk terlebih dahulu untuk menikmati layanan Burbir. >> </pre> <p style="text-align: center;">Gambar 6.4.3</p>	Jika belum login, maka akan ditampilkan pesan bahwa pengguna belum login
HAPUS_TEMAN	<pre>>> HAPUS TEMAN; Masukkan nama pengguna: Tuan Hak; Apakah anda yakin ingin menghapus Tuan Hak dari daftar teman anda? (YA/TIDAK) YA; Tuan Hak berhasil dihapus dari daftar teman anda</pre> <p style="text-align: center;">Gambar 6.4.4</p>	Jika masukkan nama ada dalam daftar teman dan pengguna mengkonfirmasi untuk menghapus, maka nama teman tersebut akan terhapus dari daftar teman
HAPUS_TEMAN	<pre>>> HAPUS TEMAN; Masukkan nama pengguna: Tuan Hak; Apakah anda yakin ingin menghapus Tuan Hak dari daftar teman anda? (YA/TIDAK) TIDAK; Tuan Hak tidak jadi dihapus dari daftar teman anda</pre> <p style="text-align: center;">Gambar 6.4.5</p>	Jika masukkan nama ada dalam daftar teman dan pengguna membatalkan untuk menghapus, maka nama teman tersebut batal terhapus dari daftar teman
HAPUS_TEMAN	<pre>>> HAPUS TEMAN; Masukkan nama pengguna: Tuan Vin; Tuan Vin bukan teman Anda.</pre> <p style="text-align: center;">Gambar 6.4.6</p>	Jika masukkan nama tidak ada pada daftar teman, maka akan ditampilkan pesan bahwa nama tersebut bukanlah teman dari pengguna
TAMBAH_TEMA N	<pre>>> TAMBAH TEMAN; Terdekat permintaan pertemanan yang belum Anda setujui. Silakan kosongkan daftar permintaan pertemanan untuk Anda terlebih dahulu.</pre> <p style="text-align: center;">Gambar 6.4.7</p>	Apabila ingin menambah teman padahal masih terdapat permintaan pertemanan kepada user yang belum disetujui, maka user akan diminta untuk mengosongkan permintaan pertemanan tersebut terlebih dahulu.

TAMBAH_TEMA N	<pre>>> TAMBAH TEMAN;</pre> <p>Masukkan nama pengguna: Hihanghoheng;</p> <p>Pengguna bernama Hihanghoheng tidak ditemukan.</p>	<p>Gambar 6.4.8</p> <p>Apabila menambahkan teman dengan nama yang tidak valid, maka akan dikeluarkan pesan bahwa pengguna tersebut tidak ditemukan. Kemudian, user akan diminta untuk mengisi ulang nama teman yang ingin ditambahkan.</p>
TAMBAH_TEMA N	<pre>>> TAMBAH TEMAN;</pre> <p>Masukkan nama pengguna: Tuan Man;</p> <p>Permintaan pertemanan kepada Tuan Man telah dikirim. Tunggu beberapa saat hingga permintaan anda disetujui.</p>	<p>Gambar 6.4.9</p> <p>Apabila menginput nama teman yang valid dan berhasil melakukan pengiriman permintaan pertemanan maka akan dituliskan pesan berhasil.</p>
TAMBAH_TEMA N	<pre>>> TAMBAH TEMAN;</pre> <p>Masukkan nama pengguna: Tuan Man;</p> <p>Anda sudah mengirimkan permintaan pertemanan kepada Tuan Man. Silakan tunggu hingga permintaan Anda disetujui.</p>	<p>Gambar 6.4.10</p> <p>Apabila menambahkan teman yang sudah direquest maka akan dikeluarkan pesan</p>

6.5 Permintaan Pertemanan

Fitur (dalam bentuk perintah)	Screenshot	Hasil yang diharapkan atau Keterangan lainnya
DAFTAR_PERMI NTAAN_PERTE MANAN;	<pre>>> DAFTAR_PERMINTAAN_PERTEMANAN;</pre> <p>Terdapat 2 permintaan pertemanan untuk Anda.</p> <ul style="list-style-type: none"> Tuan Bus <ul style="list-style-type: none"> Jumlah teman: 1 Tuan Vin <ul style="list-style-type: none"> Jumlah teman: 0 	<p>Pengguna saat ini: Tuan Hak. Akan ditampilkan daftar teman yang meminta kepada akun yang saat ini sedang login untuk berteman</p>

DAFTAR_PERMI NTAAN_PERTE MANAN;	<pre>>> DAFTAR_PERMINTAAN_PERTEMANAN;</pre> <p>Tidak ada permintaan pertemanan.</p> <p style="text-align: center;">Gambar 6.5.2</p>	Pengguna saat ini : Tuan Bri. Akan ditampilkan pesan bahwa akun saat ini tidak memiliki permintaan pertemanan jika belum ada akun lain yang meminta akun saat ini untuk berteman
SETUJUI_PERTE MANAN;	<pre>>> SETUJUI_PERTEMANAN;</pre> <p>Permintaan pertemanan teratas dari Tuan Vin Tuan Vin Jumlah teman: 0</p> <p>Apakah Anda ingin menyetujui permintaan pertemanan ini? (YA/TIDAK) YA;</p> <p>Permintaan pertemanan dari Tuan Vin telah disetujui. Selamat! Anda telah berteman dengan Tuan Vin.</p> <p style="text-align: center;">Gambar 6.5.3</p>	Pengguna saat ini: Tuan Hak. Akan ditampilkan permintaan pertemanan teratas dan jika pengguna memasukkan YA, maka permintaan akan disetujui
SETUJUI_PERTE MANAN;	<pre>>> SETUJUI_PERTEMANAN;</pre> <p>Permintaan pertemanan teratas dari Tuan Bus Tuan Bus Jumlah teman: 1</p> <p>Apakah Anda ingin menyetujui permintaan pertemanan ini? (YA/TIDAK) TIDAK;</p> <p>Permintaan pertemanan dari Tuan Bus berhasil ditolak</p> <p style="text-align: center;">Gambar 6.5.4</p>	Pengguna saat ini: Tuan Hak. Akan ditampilkan permintaan pertemanan teratas dan jika pengguna memasukkan TIDAK, maka permintaan akan ditolak

6.6 Kicauan

Fitur (dalam bentuk perintah)	Screenshot	Hasil yang diharapkan atau Keterangan lainnya

KICAU;	<pre>>> KICAU; Masukkan kicauan : Hello, Burbir!; Selamat! kicauan telah diterbitkan! Detil kicauan: ID = 6 Tuan Hak 24/11/2023 16:44:06 Hello, Burbir! Disukai: 0</pre>	<p>Menerima masukan bertipe bentukan Word dan menampilkan masukan teks kicauan bersamaan dengan data lainnya, yaitu id kicauan, nama pembuat kicauan, waktu dibuatnya kicauan, dan jumlah like kicauan tersebut. Kicauan berhasil ditampilkan dengan panjang teks kicau < 280 karakter.</p>
KICAU;	<pre>>> KICAU; Masukkan kicauan : ; Kicauan tidak boleh hanya berisi spasi!</pre>	<p>Kicauan gagal dibuat karena masukan pengguna hanya berupa spasi tanpa karakter apapun sebelum MARK.</p>
KICAU;	<pre>>> KICAU; Masukkan kicauan : Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. Selamat! Kicauan telah diterbitkan! Detil kicauan: ID = 7 Tuan Hak 24/11/2023 16:57:01 Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. Disukai: 0</pre>	<p>Kicauan berhasil ditampilkan dengan memotong inputan teks kicau akibat batasan teks kicau sepanjang 280 karakter dan inputan pengguna melebihi batasan karakter teks kicau.</p>
KICAUAN;		<p>Menampilkan seluruh kicauan milik pengguna dan teman pengguna terurut dengan kicauan terbaru ditampilkan</p>

	<pre>>> KICAUAN; ID = 9 Tuan Hak 24/11/2023 17:25:55 takut Disukai: 0 ID = 8 Tuan Bri 24/11/2023 17:25:33 tes teman Disukai: 0 ID = 7 Tuan Bri 24/11/2023 17:25:16 Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five cent Disukai: 0 ID = 6 Tuan Hak 24/11/2023 17:24:53 Hello, Burbir! Disukai: 0 ID = 5 Tuan Bri 13/10/2023 11:10:01 wjhheebfrbfh Disukai: 101</pre>	<p>teratas. Pada kondisi ini, pengguna adalah Tuan Hak dan Tuan Hak berteman dengan Tuan Bri.</p>
	Gambar 6.6.4	
SUKA_KICAUAN <IDKicau>;	<pre>>> SUKA_KICAUAN 1000; Tidak ditemukan kicauan dengan ID = 1000</pre>	<p>Program mengembalikan pesan gagal apabila pengguna memasukkan ID Kicauan yang tidak valid.</p>
SUKA_KICAUAN <IDKicau>;	<pre>>> SUKA_KICAUAN 2; ID = 2 Tuan Man 14/10/2023 11:09:01 Ini Tuan Bus Disukai: 16</pre>	<p>Pengguna dapat menyukai kicauan milik pengguna dengan tipe akun publik meskipun pengguna tidak berteman dengan akun tersebut. Kemudian, program akan menampilkan kicauan dengan jumlah like yang sudah bertambah.</p>
	Gambar 6.6.5	
	Gambar 6.6.6	

SUKA_KICAUAN <IDKicau>;	>> SUKA_KICAUAN 1; Wah, kicauan tersebut dibuat oleh akun privat! Ikuti akun itu dulu ya Gambar 6.6.7	Pengguna tidak dapat menyukai kicauan milik pengguna dengan tipe akun privat sebelum berteman dengan akun tersebut.
UBAH_KICAUAN <IDKicau>;	>> UBAH_KICAUAN 2; Kicauan dengan ID = 2 bukan milikmu! Gambar 6.6.8	Pengguna tidak dapat mengubah teks kicauan yang bukan milik pengguna tersebut.
UBAH_KICAUAN <IDKicau>;	>> UBAH_KICAUAN 3; Masukkan kicauan baru : halo saya berhasil mengubah kicauan ini; ID = 3 Tuan Hak 20/11/2023 15:31:01 halo saya berhasil mengubah kicauan ini Disukai: 2381 Gambar 6.6.9	Pengguna dapat mengubah teks kicauan dari kicauan yang dimiliki oleh pengguna tersebut. Kemudian, program akan menampilkan kicauan dengan teks baru sesuai masukan perubahan.

6.7 Balasan

Fitur (dalam bentuk perintah)	Screenshot	Hasil yang diharapkan atau Keterangan lainnya
BALAS <IDKicau> <IDBalasan>;	>> BALAS 5 3; 5 3 Masukkan balasan: INI BALASAN DARI BALASAN ID 3; Selamat! balasan telah diterbitkan! Detil balasan: 5 Tuan Hak 24/11/2023 18:47:49 INI BALASAN DARI BALASAN ID 3	Dengan perintah BALASAN 5 3, pengguna ingin membalas balasan dengan id 3 di kicauan dengan id 5. Karena semua syarat terpenuhi, balasan berhasil terbit.

	Gambar 6.7.1	
BALASAN <IDKicau>;	<pre>>> BALASAN 5; 1 Tuan Bri 14/10/2023 11:09:18 Ini Balasan dari Node Utama, yaitu Kicauan ke-5 2 Tuan Man 14/10/2023 11:09:12 Ini Balasan dari Balasan ID ke-1 4 Tuan Man 14/10/2023 11:09:12 Ini Balasan dari Balasan ID ke-2 3 Tuan Man 14/10/2023 11:09:12 Ini Balasan dari Balasan ID ke-1 5 Tuan Hak 24/11/2023 18:47:49 INI BALASAN DARI BALASAN ID 3</pre>	Setelah menambah balasan di balasan dengan id ke-3, maka tampilan BALASAN sesuai apa yang diharapkan
	Gambar 6.7.2	
HAPUS_BALASA N <IDKicau> <IDBalasan>;	<pre>>> HAPUS_BALASAN 5 5; BALASAN BERHASIL DIHAPUS >> BALASAN 5; 1 Tuan Bri 14/10/2023 11:09:18 Ini Balasan dari Node Utama, yaitu Kicauan ke-5 2 Tuan Man 14/10/2023 11:09:12 Ini Balasan dari Balasan ID ke-1 4 Tuan Man 14/10/2023 11:09:12 Ini Balasan dari Balasan ID ke-2 3 Tuan Man 14/10/2023 11:09:12 Ini Balasan dari Balasan ID ke-1</pre>	Setelah membalas di balasan ke 3, perintah HAPUS_BALASAN 5 5 akan menghapus balasan tersebut. Perintah BALASAN di bawah menunjukkan bahwa balasan berhasil dihapus
	Gambar 6.7.3	

6.8 Draf Kicauan

Fitur (dalam bentuk perintah)	Screenshot	Hasil yang diharapkan atau Keterangan lainnya
BUAT_DRAF;	<pre>>> BUAT_DRAF; Masukkan draf : halo ini buat draf; Apakah anda ingin menghapus, menyimpan, atau menerbitkan draf ini? SIMPAN; Draf telah berhasil disimpan!</pre>	<p>Pengguna berhasil membuat draf dan menyimpan teks yang dimasukkan pengguna sebagai draf kicauan ketika pengguna memasukkan perintah SIMPAN.</p>
BUAT_DRAF;	<pre>>> BUAT_DRAF; Masukkan draf : halo ini simpan draf; Apakah anda ingin menghapus, menyimpan, atau menerbitkan draf ini? HAPUS; Draf telah berhasil dihapus!</pre>	<p>Pengguna berhasil menghapus draf yang dimasukkan pengguna ketika pengguna memasukkan perintah HAPUS.</p>
BUAT_DRAF;	<pre>>> BUAT_DRAF; Masukkan draf : halo ini terbit draf; Apakah anda ingin menghapus, menyimpan, atau menerbitkan draf ini? TERBIT; Selamat! Draf kicauan telah diterbitkan! ID = 6 Tuan Hak 24/11/2023 19:01:35 halo ini terbit draf Disukai: 0</pre>	<p>Pengguna berhasil membuat draf dan menerbitkan masukan teks draf kicauan menjadi kicauan ketika pengguna memasukkan perintah TERBIT.</p>
LIHAT_DRAF;	<pre>>> LIHAT_DRAF; Ini draf terakhir anda: halo ini buat draf 24/11/2023 18:59:12 Apakah anda ingin mengubah, menghapus, atau menerbitkan draf ini? (KEMBALI jika ingin kembali) UBAH; Masukkan draf yang baru: halo ini mengubah draf; Apakah anda ingin menghapus, menyimpan, atau menerbitkan draf ini? TERBIT; Selamat! Draf kicauan telah diterbitkan! ID = 7 Tuan Hak 24/11/2023 19:02:57 halo ini mengubah draf Disukai: 0</pre>	<p>Pengguna dapat melihat draf terakhir yang pengguna buat, melakukan pengubahan teks kicauan pada draf teratas dengan perintah UBAH, dan menerbitkan draf kicauan teratas menjadi kicauan saat pengguna memasukkan perintah TERBIT.</p>

LIHAT_DRAF;	<pre>>> LIHAT_DRAF; Ini draf terakhir anda: Hebo 3 14/10/2023 11:09:18 Apakah anda ingin mengubah, menghapus, atau menerbitkan draf ini? (KEMBALI jika ingin kembali) HAPUS; Draf telah berhasil dihapus!</pre>	Gambar 6.8.5	Pengguna dapat melihat draf terakhir yang pengguna buat dan menghapus draf kicauan teratas saat pengguna memasukkan perintah HAPUS.
LIHAT_DRAF;	<pre>>> LIHAT_DRAF; Ini draf terakhir anda: Hebo 2 14/10/2023 11:09:17 Apakah anda ingin mengubah, menghapus, atau menerbitkan draf ini? (KEMBALI jika ingin kembali) TERBIT; Selamat! Draf kicauan telah diterbitkan! ID = 8 Tuan Hak 24/11/2023 19:04:08 Hebo 2 Disukai: 0</pre>	Gambar 6.8.6	Pengguna dapat melihat draf terakhir yang pengguna buat dan menerbitkan draf kicauan teratas menjadi kicauan saat pengguna memasukkan perintah TERBIT.

6.9 Utas

Fitur (dalam bentuk perintah)	<i>Screenshot</i>	Hasil yang diharapkan atau Keterangan lainnya
UTAS <IDKicau>;	<pre>>> UTAS 3; Kicauan ini sudah dijadikan utas</pre>	Pengguna yang sedang login : Tuan Hak Kicauan dengan ID 3 sudah dijadikan utas oleh Tuan Hak, sehingga tidak dapat membuat utas pada ID Kicau tersebut.
UTAS <IDKicau>;	<pre>>> UTAS 10; Kicauan tidak ditemukan!</pre>	Pengguna yang sedang login : Tuan Hak Kicauan dengan ID 10 belum tersedia sehingga tidak dapat dijadikan utas.

UTAS <IDKicau>;	<pre>>> KICAU; Masukkan kicauan : Halo ini Tuan Hak; Selamat! kicauan telah diterbitkan! Detil kicauan: ID = 6 Tuan Hak 24/11/2023 16:03:53 Halo ini Tuan Hak Disukai: 0 >> UTAS 6; Utas berhasil dibuat! Masukkan kicauan: Selamat pagi selamat bekerja; Apakah Anda ingin melanjutkan utas ini? (YA/TIDAK) YA; Masukkan kicauan: Bersiaplah nikmati hari; Apakah Anda ingin melanjutkan utas ini? (YA/TIDAK) YA; Masukkan kicauan: Itu tadi Sherina; Apakah Anda ingin melanjutkan utas ini? (YA/TIDAK) TIDAK; Utas selesai!</pre>	<p>Pengguna yang sedang login : Tuan Hak</p> <p>Setelah Tuan Hak membuat kicauan dengan ID Kicau 6, kicauan tersebut dapat dijadikan utas karena baru saja dibuat. Input utas dapat dilakukan berkali-kali ketika mengetikkan command ‘YA’ lalu akan berhenti saat mengetikkan command ‘TIDAK’. Setelah mengetikkan command ‘TIDAK’, akan muncul keterangan ‘Utas selesai!’</p>
UTAS <IDKicau>;	<pre>>> UTAS 1; Utas ini bukan milik anda!</pre>	<p>Pengguna yang sedang login : Tuan Hak</p> <p>Tuan Hak tidak dapat menjadikan kicauan dengan ID Kicau 1 sebagai utas karena kicauan tersebut bukan miliknya, melainkan milik Tuan Bus.</p>

CETAK_UTAS <IDUtas>;	<pre>>> CETAK_UTAS 3; ID = 6 Tuan Hak 24/11/2023 16:19:15 Hai ini Tuan Hak Disukai: 0 INDEX = 1 Tuan Hak 24/11/2023 16:19:43 Selamat pagi selamat bekerja INDEX = 2 Tuan Hak 24/11/2023 16:19:43 Bersiaplah nikmati hari INDEX = 3 Tuan Hak 24/11/2023 16:19:43 Itu tadi Sherina</pre>	Pengguna yang sedang login : Tuan Hak Kicauan dengan ID 6 yang telah dijadikan utas memiliki ID Utas 3. Saat command cetak akan menampilkan kicauan utama beserta semua utasnya, lengkap dengan informasi index, author yang membuat utas, datetime pembuatan utas, dan isi utas tersebut.
CETAK_UTAS <IDUtas>;	<pre>>> CETAK_UTAS 5; Utas tidak ditemukan!</pre>	Pengguna yang sedang login : Tuan Hak Utas dengan ID Utas 5 tidak tersedia sehingga tidak dapat dicetak
CETAK_UTAS <IDUtas>;	<pre>>> CETAK_UTAS 4; Akun yang membuat utas ini adalah akun privat! Ikuti dahulu akun ini untuk melihat utasnya!</pre>	Pengguna yang sedang login : Tuan Hak Utas dengan ID Utas 4 tidak dapat dicetak karena authornya adalah akun privat yaitu Tuan Bus.
CETAK_UTAS <IDUtas>;	<pre>>> CETAK_UTAS 3; ID = 6 Tuan Hak 24/11/2023 16:19:15 Hai ini Tuan Hak Disukai: 0</pre>	Pengguna yang sedang login : Tuan Hak Utas dengan ID Utas 3 hanya tersisa kicauan utama karena semua utasnya telah dihapus.

	Gambar 6.9.8	
CETAK_UTAS <IDUtas>;	<pre>>> CETAK_UTAS 4; ID = 9 Tuan Vin 24/11/2023 17:06:17 Hai ini Vin Disukai: 0 INDEX = 1 Tuan Vin 24/11/2023 17:06:27 Oke guys INDEX = 2 Tuan Vin 24/11/2023 17:06:27 Semangat ya nubesnya INDEX = 3 Tuan Vin 24/11/2023 17:06:27 Jangan koid dulu</pre>	Pengguna yang sedang login : Tuan Hak Utas dengan ID 4 dapat ditampilkan walaupun bukan milik sendiri. Utas tersebut adalah milik Tuan Vin yang akunnya publik sehingga dapat ditampilkan.
	Gambar 6.9.9	
HAPUS_UTAS <IDUtas> <index>;	<pre>>> HAPUS_UTAS 3 1; Kicauan sambungan berhasil dihapus!</pre>	Pengguna yang sedang login : Tuan Hak Utas dengan ID Utas 3 dan index 1 berhasil dihapus oleh Tuan Hak karena ia lah author utas tersebut.
HAPUS_UTAS <IDUtas> <index>;	<pre>>> HAPUS_UTAS 3 4; Kicauan sambungan dengan index 4 tidak ditemukan pada utas!</pre>	Pengguna yang sedang login : Tuan Hak Tidak dapat index 4 pada ID Utas 3 sehingga tidak ada utas yang dihapus.
	Gambar 6.9.11	

HAPUS_UTAS <IDUtas> <index>;	<pre>>> HAPUS_UTAS 7 1; Utas tidak ditemukan!</pre> <p style="text-align: center;">Gambar 6.9.12</p>	Pengguna yang sedang login : Tuan Hak Tidak terdapat utas dengan ID 7 sehingga tidak ada utas yang dihapus.
HAPUS_UTAS <IDUtas> <index>;	<pre>>> HAPUS_UTAS 4 1; Anda tidak bisa menghapus kicauan dalam utas ini!</pre> <p style="text-align: center;">Gambar 6.9.13</p>	Pengguna yang sedang login : Tuan Hak Tuan Hak tidak dapat menghapus utas dengan ID Utas 4 karena utas tersebut bukan miliknya, melainkan milik Tuan Bus.
HAPUS_UTAS <IDUtas> <index>;	<pre>>> HAPUS_UTAS 3 0; Anda tidak bisa menghapus kicauan utama!</pre> <p style="text-align: center;">Gambar 6.9.14</p>	Pengguna yang sedang login : Tuan Hak Index 0 menandakan kicauan utama sehingga tidak dapat dihapus (sesuai dengan ketentuan pada kicauan).
SAMBUNG_UTAS <IDUtas> <index>;	<pre>>> SAMBUNG_UTAS 4 1; Index terlalu rendah!</pre> <p style="text-align: center;">Gambar 6.9.15</p>	Pengguna yang sedang login : Tuan Vin Sesuai dengan Gambar 6.9.9, Tuan Vin memiliki utas dengan ID 4. Saat ini ia sedang memiliki index utas 1, 2, 3. Maka ketika ia ingin menyambung utas pada index 1 menjadi tidak bisa karena index 1 sudah terisi.

SAMBUNG_UTAS <IDUtas> <index>;	<pre>>> SAMBUNG_UTAS 4 6; Index terlalu tinggi!</pre> <p style="text-align: center;">Gambar 6.9.16</p>	Pengguna yang sedang login : Tuan Vin Sesuai dengan Gambar 6.9.9, Tuan Vin memiliki utas dengan ID 4. Saat ini ia sedang memiliki index utas 1, 2, 3. Maka ketika ia ingin menyambung utas pada index 6 menjadi tidak bisa karena index tersebut terlalu jauh dan utas tidak boleh memiliki index yang loncat.
SAMBUNG_UTAS <IDUtas> <index>;	<pre>>> SAMBUNG_UTAS 4 4; Masukkan kicauan: Pagiku cerahku matahari bersinar;</pre> <p style="text-align: center;">Gambar 6.9.17</p>	Pengguna yang sedang login : Tuan Vin Sesuai dengan Gambar 6.9.9, Tuan Vin memiliki utas dengan ID 4. Saat ini ia sedang memiliki index utas 1, 2, 3. Maka ketika ia ingin menyambung utas pada index 4 menjadi berhasil dan akan meminta input utas pada index 4.

6.10 Simpan dan Muat

Fitur (dalam bentuk perintah)	Screenshot	Hasil yang diharapkan atau Keterangan lainnya

SIMPAN;	<pre>>> SIMPAN; Masukkan nama folder yang hendak dimuat. config-2; Belum terdapat config-2. Akan dilakukan pembuatan config-2 terlebih dahulu. Mohon tunggu... 1... 2... 3... config-2 sudah berhasil dibuat. Anda akan melakukan penyimpanan di config-2 Mohon tunggu... 1... 2... 3... Penyimpanan telah berhasil dilakukan! >> SIMPAN; Masukkan nama folder yang hendak dimuat. config-2; Anda akan melakukan penyimpanan di config-2 Mohon tunggu... 1... 2... 3... Penyimpanan telah berhasil dilakukan!</pre>	<p>Folder config yang di load berasal dari config-1.</p> <p>Akan disimpan dalam folder config-2, sehingga akan dibuat folder baru bernama config-2 yang berisi file-file config</p> <p>Jika disimpan lagi ke folder config-2, maka tidak terjadi pembuatan folder melainkan hanya mengganti isi file-file config dari folder tersebut.</p>
MUAT;	<pre>>> MUAT; Masukkan nama folder yang hendak dimuat. config-2; Anda akan melakukan pemuatan dari config-2. Mohon tunggu... 1... 2... 3... Pemuatan selesai ! >> </pre>	<p>Folder config-2 yang telah dibuat di fitur SIMPAN; digunakan dan dimuat di fitur MUAT dengan kondisi user belum login</p>
MUAT;	<pre>Masukkan nama folder yang hendak dimuat. config-3; Tidak ada folder yang dimaksud. >> </pre>	<p>Folder config-3 belum pernah dibuat sehingga akan muncul pesan kesalahan jika dimuat config-3</p>
MUAT;	<pre>>> MUAT; Anda harus keluar terlebih dahulu untuk melakukan pemuatan. >> </pre>	<p>MUAT tidak bisa diakses dikarenakan user masih dalam posisi login, harus melakukan</p>

		KELUAR terlebih dahulu
--	--	------------------------

STEI- ITB	IF2110-TB-A-01	Halaman 43 dari 68 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	INISIALI SASI	Memastikan program dapat melakukan pemuatan file config	Mengetikkan nama file konfigurasi yang akan di- <i>load</i>	Nama file konfigurasi, yaitu config-1	Program dapat memuat data-data yang ada pada file config yang dipilih oleh user	Gambar 6.1.1
2	DAFTAR	Menguji penyimpanan data nama dan kata sandi pada program	Mengetikkan command DAFTAR, kemudian memasukkan nama dan password yang akan digunakan.	DAFTAR, nama user, password user	Program berhasil menyimpan data nama dan kata sandi	Gambar 6.2.1
3	DAFTAR	Menguji kondisi dimana nama yang ingin didaftarkan sudah pernah diambil	Mengetikkan perintah DAFTAR, kemudian memasukkan nama.	DAFTAR dan nama user	Program berhasil memberikan pemberitahuan bahwa nama tersebut telah diambil	Gambar 6.2.2
4	MASUK	Memeriksa kondisi dimana nama yang dimasukkan belum pernah didaftarkan	Mengetikkan command MASUK dan nama user	MASUK dan nama user	Program berhasil menampilkan pesan bahwa nama yang dicari tidak ada	Gambar 6.2.3
5	MASUK	Memeriksa kondisi dimana kata sandi yang dimasukkan tidak tepat	Mengetikkan command MASUK, nama user, dan kata sandi	MASUK, nama user, dan kata sandi	Program berhasil menampilkan pesan kesalahan password	Gambar 6.2.3

No.	Fitur yang Dtes	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
6	MASUK	Memeriksa kondisi dimana kata sandi yang dimasukkan telah sesuai	Mengetikkan command MASUK, nama user, dan kata sandi password user	MASUK,	Program berhasil masuk pada status dimana pengguna telah login	Gambar 6.2.3
7	MASUK	Memeriksa kondisi dimana pengguna memanggil perintah MASUK saat sudah pernah melakukan MASUK sebelumnya	Mengetikkan command MASUK	MASUK	Menampilkan pesan bahwa pengguna sudah melakukan MASUK	Gambar 6.2.3
8	KELUAR	Memeriksa kondisi dimana pengguna belum MASUK	Mengetikkan command KELUAR	KELUAR	Menampilkan pesan bahwa pengguna belum login	Gambar 6.2.4
9	KELUAR	Memeriksa kondisi dimana pengguna sudah MASUK dan ingin KELUAR	Melakukan MASUK lalu memasukkan nama dan kata sandi. Setelah itu, mengetikkan command KELUAR	MASUK, nama user, kata sandi, KELUAR	Program masuk ke dalam status tidak login	Gambar 6.2.4
10	TUTUP_PROGRAM	Memeriksa kondisi saat pengguna ingin keluar dari program	Mengetikkan command TUTUP_PROGRAM	TUTUP_PRO GRAM	Berhasil keluar dari program	Gambar 6.2.5
11	GANTI_PROFIL	Memeriksa kondisi dimana pengguna	Mengetikkan command GANTI_PROFIL,	GANTI_PRO FIL, Bio akun, No HP	Menampilkan pesan kesalahan bahwa no HP tidak valid	Gambar 6.3.1

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		memasukkan No HP yang tidak invalid	memasukkan bio akun, dan no HP			
12	GANTI_PROFIL	Memeriksa kondisi dimana pengguna memasukkan weton invalid	Mengetikkan command GANTI_PROFIL, memasukkan bio akun, no HP, dan weton	GANTI_PROFIL, bio akun, no HP, dan weton	Menampilkan pesan kesalahan bahwa weton tak valid	Gambar 6.3.1
13	LIHAT_PROFIL [NAMA]	Memeriksa kondisi akun publik	Mengetikkan command LIHAT_PROFIL diikuti nama user	LIHAT_PROFIL, nama user	Menampilkan profil dari nama user yang dirujuk	Gambar 6.3.2
14	LIHAT_PROFIL [NAMA]	Memeriksa kondisi akun privat	Mengetikkan command LIHAT_PROFIL diikuti nama user	LIHAT_PROFIL, nama user	Menampilkan pesan bahwa akun tersebut ialah privat	Gambar 6.3.2
15	ATUR_JENIS_AKUN	Memeriksa proses perubahan jenis akun pada suatu akun	Mengetikkan command ATUR_JENIS_AKUN dan masukkan YA/TIDAK	ATUR_JENIS_AKUN, masukkan YA/TIDAK	Menampilkan data akun dan pertanyaan apakah ingin merubah jenis akun	Gambar 6.3.3
16	UBAH_FOTO_PROFIL	Menguji program untuk menyimpan data perubahan foto profil	Mengetikkan command UBAH_FOTO_PROFIL dan masukkan foto profil	UBAH_FOTO_PROFIL, foto profil	Program berhasil menyimpan perubahan	Gambar 6.3.4
17	DAFTAR_TEMAN	Memeriksa kondisi dimana pengguna memiliki teman	Memasukkan command DAFTAR_TEMAN	DAFTAR_TEMAN	Menampilkan daftar teman yang dimiliki	Gambar 6.4.1

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
18	DAFTAR _TEMAN	Memeriksa kondisi dimana pengguna tidak memiliki teman	Memasukkan command DAFTAR_TEMAN	DAFTAR_TE MAN	Menampilkan pesan bahwa pengguna belum memiliki teman	Gambar 6.4.2
19	DAFTAR _TEMAN	Memeriksa kondisi dimana pengguna belum login	Memasukkan command DAFTAR_TEMAN	DAFTAR_TE MAN	Menampilkan pesan bahwa pengguna belum login	6.4.3
20	HAPUS_ TEMAN	Memeriksa kondisi dimana nama teman dari pengguna terdapat pada DAFTAR_TEMA N dan dihapus	Memasukkan command HAPUS_TEMAN, nama user, serta pilihan jawaban YA	HAPUS_TE MAN, nama user, YA	Program berhasil menghapus teman dari DAFTAR_TEMAN	6.4.4
21	HAPUS_ TEMAN	Memeriksa kondisi dimana nama teman dari pengguna terdapat pada DAFTAR_TEMA N dan tidak dihapus	Memasukkan command HAPUS_TEMAN, nama user, serta pilihan jawaban TIDAK	HAPUS_TE MAN, nama user, TIDAK	Program berhasil membatalkan penghapusan teman	Gambar 6.4.5
22	HAPUS_ TEMAN	Memeriksa kondisi dimana nama teman dari pengguna tidak terdapat pada DAFTAR_TEMA N	Memasukkan command HAPUS_TEMAN dan nama user	HAPUS_TE MAN dan nama user	Program menampilkan pesan bahwa nama tersebut bukanlah teman dari pengguna	Gambar 6.4.6

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
23	TAMBAH _TEMAN	Memeriksa kondisi dimana nama calon teman ditemukan dan daftar permintaan pertemanan pengguna telah kosong	Memasukkan command TAMBAH_TEMAN dan nama calon teman	TAMBAH_T EMAN dan nama	Program berhasil menambahkan data permintaan pertemanan kepada calon teman	Gambar 6.4.9
24	TAMBAH _TEMAN	Memeriksa kondisi dimana nama calon teman tak ditemukan dan daftar permintaan pertemanan pengguna telah kosong	Memasukkan command TAMBAH_TEMAN dan nama calon teman	TAMBAH_T EMAN dan nama	Menampilkan pesan bahwa nama calon teman tidak ditemukan	Gambar 6.4.8
25	TAMBAH _TEMAN	Memeriksa kondisi dimana daftar permintaan pertemanan pengguna belum kosong	Memasukkan command TAMBAH_TEMAN	TAMBAH_T EMAN	Menampilkan pesan bahwa masih terdapat permintaan pertemanan yang belum disetujui	Gambar 6.4.7
26	TAMBAH _TEMAN	Memeriksa kondisi dimana nama calon teman ditemukan dan nama tersebut telah dimintai untuk berteman	Memasukkan command TAMBAH_TEMAN dan nama calon teman	TAMBAH_T EMAN dan nama	Menampilkan pesan bahwa pengguna telah mengirimkan permintaan pertemanan ke calon teman tersebut	Gambar 6.4.10
27	DAFTAR _PERMI	Memeriksa kondisi dimana	Memasukkan command DAFTAR_PERMI	DAFTAR_PERMI	Menampilkan daftar permintaan	Gambar 6.5.1

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
	NTAAN_PERTEMANAN	pengguna memiliki permintaan pertemanan	DAFTAR_PERMINTAAN_PERTEMANAN	PERTEMANAN	pertemanan yang dimiliki pengguna	
28	DAFTAR_PERMI_NTAAN_PERTEMANAN	Memeriksa kondisi dimana pengguna tidak memiliki permintaan pertemanan	Memasukkan command DAFTAR_PERMINTAAN_PERTEMANAN	DAFTAR_PERMINTAAN_PERTEMANAN	Menampilkan pesan bahwa daftar permintaan pertemanan kosong	Gambar 6.5.1
29	SETUJUI_PERTEMANAN	Menguji program untuk menerima permintaan pertemanan	Memasukkan command SETUJUI_PERTEMANAN dan YA	SETUJUI_PERTEMANAN dan YA	Menampilkan permintaan teratas dan berhasil menyetujui pertemanan	Gambar 6.5.3
30	SETUJUI_PERTEMANAN	Menguji program untuk menolak permintaan pertemanan	Memasukkan command SETUJUI_PERTEMANAN dan TIDAK	SETUJUI_PERTEMANAN dan TIDAK	Menampilkan permintaan teratas dan berhasil menolak pertemanan	Gambar 6.5.4
36	KICAU	Memeriksa keberjalanannya dengan memastikan pengguna dapat menuliskan dan menerbitkan kicauan diikuti penampilan hasil kicauan.	Mengetikkan command KICAU saat pengguna terdaftar sudah masuk ke program, kemudian pengguna memberikan input masukan teks kicauan.	KICAU	Menampilkan kicauan dengan format id kicauan, nama pembuat kicauan, waktu dibuatnya kicauan, teks kicauan, dan jumlah like kicauan tersebut.	Gambar 6.6.1, 6.6.2, dan 6.6.3
37	KICAUAN	Memeriksa keberjalanannya dengan	Mengetikkan command KICAUAN saat pengguna	KICAUAN	Menampilkan kicauan pengguna dan teman-teman	Gambar 6.6.4

No.	Fitur yang Dtes	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		memastikan program dapat menampilkan kicauan pengguna dan teman-teman pengguna.	terdaftar sudah masuk ke program.		pengguna terurut dengan kicauan terbaru berada pada urutan teratas.	
38	SUKA_K ICAUAN [IDKicau]	Memeriksa kondisi di mana ID Kicauan tidak valid.	Mengetikkan command SUKA_KICAUAN diikuti dengan ID Kicau yang tidak valid.	SUKA_KICA UAN, ID Kicau	Program mengembalikan pesan gagal karena ID Kicau tidak valid.	Gambar 6.6.5
39	SUKA_K ICAUAN [IDKicau]	Memeriksa keberjalanannya program dengan memastikan pengguna dapat menyukai kicauan yang menyebabkan jumlah like suatu kicauan bertambah. Kemudian, kicauan yang disukai akan ditampilkan.	Mengetikkan command SUKA_KICAUAN diikuti dengan ID kicauan yang ingin disukai.	SUKA_KICA UAN, ID Kicau	Jumlah like pada kicauan yang dituju bertambah 1 dan program menampilkan kicauan yang telah bertambah sejumlah likenya.	Gambar 6.6.6
40	SUKA_K ICAUAN [IDKicau]	Memeriksa kondisi program jika pengguna mencoba melakukan	Mengetikkan command SUKA_KICAUAN diikuti dengan ID Kicau.	SUKA_KICA UAN, ID Kicau	Program mengembalikan pesan yang menyatakan bahwa	Gambar 6.6.7

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		melakukan perintah SUKA_KICAUA N pada kicauan dengan pembuat bertipe akun privat dan pengguna tidak berteman dengan			akun pembuat kicauan privat.	
41	UBAH_KICAUAN [IDKicau]	Memeriksa kondisi program saat menerima masukan ID Kicau yang bukan milik pengguna pada perintah UBAH_KICAUA N.	Mengetikkan command UBAH_KICAUAN diikuti dengan ID Kicauan yang ingin diubah pengguna serta memasukan teks kicauan baru untuk mengganti teks pada ID Kicau yang dituju.	UBAH_KICAUAN, ID Kicau	Program mengembalikan pesan gagal apabila ID Kicau yang dimasukkan bukan milik pengguna.	Gambar 6.6.8
42	UBAH_KICAUAN [IDKicau]	Memeriksa keberjalanan program dengan memastikan program dapat menampilkan kicauan pengguna dan teman-teman pengguna.	Mengetikkan command UBAH_KICAUAN diikuti dengan ID Kicauan yang ingin diubah pengguna serta memasukan teks kicauan baru untuk mengganti teks pada ID Kicau yang dituju.	UBAH_KICAUAN, ID Kicau	Teks kicauan pada kicauan dengan ID terpilih akan berubah dan program menampilkan kicauan dengan teks yang sudah berubah.	Gambar 6.6.9
43	BALAS [IDKicau]	Memeriksa keberjalanan program dengan	Mengetikkan command BALAS diikuti dengan ID Balasan	BALAS, ID Kicau, ID Balasan	Menambah balasan dengan id = ID Balasan pada kicauan	Gambar 6.7.1

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
	[IDBalasan]	memastikan program dapat menampilkan keterangan bahwa sebuah kicauan dengan id kicau dapat dibalas pada posisi sesuai id balasan yang diinput serta menampilkan balasan yang dibuat.	Kicauan yang ingin dibalas serta ID Balasan sebagai posisi pengguna ingin membalas			
44	BALASA N [IDKicau]	Memeriksa keberjalanannya program dengan memastikan program dapat menampilkan daftar balasan dari suatu kicauan	Mengetikkan command BALASAN diikuti dengan ID Kicauan yang ingin dilihat semua daftar balasannya	BALASAN, ID Kicau	Menampilkan seluruh balasan dalam satu kicauan pada ID Kicauan yang dipilih.	Gambar 6.7.2
45	HAPUS_ BALASA N [IDKicau] [IDBalasan]	Memeriksa keberjalanannya program dengan memastikan program dapat menampilkan keterangan bahwa sebuah balasan dengan id balasan pada id kicau	Mengetikkan command HAPUS_BALASAN diikuti dengan ID Kicauan serta ID Balasan dimana balasan tersebut berada	HAPUS_BAL ASAN, ID Kicau, ID Balasan	Menampilkan keterangan apakah sebuah balasan berhasil dihapus atau tidak.	Gambar 6.7.3

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		tertentu dapat dibalas atau tidak				
46	BUAT_DRAF	Memeriksa keberjalanannya program dengan memastikan program dapat membuat, menghapus, menerbitkan, dan menampilkan draf kicauan pengguna.	Mengetikkan command BUAT_DRAF	BUAT_DRAF	Masukan draf kicauan yang masih ingin disimpan dalam draf akan dimasukkan ke dalam draf, masukan draf kicauan yang ingin diterbitkan akan diterbitkan sebagai kicauan baru, masukan draf kicauan yang ingin dihapus tidak akan masuk ke dalam draf kicauan.	Gambar 6.8.1, 6.8.2, dan 6.8.3
47	LIHAT_DRAF	Memeriksa keberjalanannya program dengan memastikan program dapat menampilkan, menghapus, mengubah, dan menerbitkan draf kicauan pengguna.	Mengetikkan command LIHAT_DRAF	LIHAT_DRAF	Menampilkan isi draf, mengubah isi draf teratas jika ingin diubah, menghapus isi draf teratas jika ingin dihapus, serta menerbitkan isi draf kicauan teratas menjadi kicauan jika ingin diterbitkan.	Gambar 6.8.4, 6.8.5, dan 6.8.6
48	UTAS [IDKicau]	Memeriksa keberjalanannya program dengan memastikan program dapat menampilkan	Mengetikkan command UTAS dengan diikuti ID Kicauan yang ingin diubah tipenya menjadi utas	UTAS, ID Kicau	Menampilkan keterangan apakah sebuah utas berhasil dibuat atau tidak	Gambar 6.9.1, 6.9.2, 6.9.3, dan 6.9.4

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		keterangan bahwa sebuah utas dapat dibuat dari id kicauan yang diinput.				
49	SAMBU NG_UTA S [IDUtas] [index]	Memeriksa keberjalanannya program dengan memastikan program dapat menampilkan keterangan bahwa sebuah utas dapat dilanjutkan berdasarkan id utas dan indeks utas pada id tersebut.	Mengetikkan command SAMBUNG_UTAS dengan diikuti ID Utas yang ingin dilanjutkan dan diikuti index dimana posisi utas tersebut ingin dilanjutkan	SAMBUNG_UTAS, ID Utas, index	Menampilkan keterangan apakah suatu utas berhasil disambung atau tidak, bergantung pada kesesuaian idUtas dan index	Gambar 6.9.15, 6.9.16, dan 6.9.17
50	HAPUS_UTAS [IDUtas] [index]	Memeriksa keberjalanannya program dengan memastikan program dapat menampilkan keterangan bahwa sebuah utas dengan indeks dan id utas tertentu dapat dihapus atau tidak.	Mengetikkan command HAPUS_UTAS dengan diikuti ID Utas serta index utas yang ingin dihapus	HAPUS_UTA S, ID Utas, index	Menampilkan keterangan apakah sebuah utas berhasil dihapus atau tidak	Gambar 6.9.10, 6.9.11, 6.9.12, 6.9.13, dan 6.9.14

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
51	CETAK_UTAS [IDUtas]	Memeriksa keberjalanannya program dengan memastikan program dapat menampilkan kicauan utama dan utas-utasnya.	Mengetikkan command CETAK_UTAS dengan diikuti ID Utas yang ingin ditampilkan daftar utasnya.	CETAK_UTAS, ID Utas	Menampilkan kicauan utama beserta seluruh utasnya dengan idUtas yang sama dan berurutan berdasarkan index utasnya	Gambar 6.9.5, 6.9.6, 6.9.7, 6.9.8, dan 6.9.9
52	SIMPAN	Memeriksa apakah ada perubahan data dan menuliskan di file penyimpanan ke dalam sebuah folder dan mengisi folder tersebut dengan 5 macam file “data.config”	Mengetikkan command SIMPAN	SIMPAN	Meminta masukan berupa nama folder yang ingin dijadikan tempat penyimpanan dan akan menampilkan jika berhasil disimpan atau tidak	6.10.1
53	MUAT	Memeriksa apakah suatu “data.config” ada dan akan disimpan ke sebuah struktur data utama yang akan digunakan dalam berbagai fitur	Mengetikkan command MUAT	MUAT	Meminta masukan nama folder yang akan dimuat yang berisi data-data config dan akan menampilkan jika berhasil dimuat atau tidak	6.10.2, 6.10.3, 6.10.4

8 Pembagian Kerja dalam Kelompok

NIM	Tugas
13522015	<ol style="list-style-type: none">1. Membuat struktur folder github2. Membuat ADT List Statik User3. Membuat program PROFIL dan PENGGUNA/USER beserta drivernya dan mengintegrasikannya dengan main.c4. Membuat semua driver dari ADT yang ada5. Membuat laporan bagian penjelasan tambahan spesifikasi tugas, data test, test script, lampiran
13522024	<ol style="list-style-type: none">1. Membuat ADT Tree Balasan2. Membuat program BALASAN beserta drivernya dan mengintegrasikannya dengan main.c3. Membuat config....4. Membuat program SIMPAN dan MUAT5. Membuat laporan
13522040	<ol style="list-style-type: none">1. Membuat perencanaan dan sketsa seluruh ADT yang dipakai pada Program Burbir2. Membuat ADT Graf dengan Representasi Adjacency Matrix (matriks pertemanan) dan ADT Priority Queue dengan Struktur Berkait (antrean permintaan pertemanan)3. Membuat heade beserta realisasi dari ADT Matriks Pertemanan dan Permintaan Pertemanan4. Membuat driver permintaan pertemanan5. Mengintegrasikan ADT Matriks Pertemanan dan Permintaan Pertemanan ke main.c6. Membuat laporan

13522047	<ol style="list-style-type: none"> 1. Membuat ADT Stack Draf Kicauan 2. Membuat program DRAF beserta drivernya dan mengintegrasikannya dengan main.c 3. Membuat baca config / Load 4. Membuat program SIMPAN dan MUAT 5. Membuat laporan
13522053	<ol style="list-style-type: none"> 1. Membuat struktur folder github 2. Convert ADT-ADT bawaan 3. Membuat dan memodifikasi ADT Word Machine 4. Membuat ADT Linked List Utas 5. Membuat program UTAS beserta drivernya dan mengintegrasikannya dengan main.c 6. Notulen asistensi dan meet kelompok 7. Membuat laporan bagian ringkasan, penjelasan tambahan spesifikasi tugas, struktur data, data test, test script
13522060	<ol style="list-style-type: none"> 1. Membuat ADT List Dinamis Kicauan 2. Membuat program KICAUAN beserta drivernya dan mengintegrasikannya dengan main.c 3. Memodifikasi ADT Word Machine 4. Notulen asistensi dan meet kelompok 5. Membuat laporan bagian ringkasan, struktur data, data test, test script

9 Lampiran

9.1 Deskripsi Tugas Besar 1

Buatlah sebuah program simulasi berbasis CLI (command-line interface). Program ini dibuat dalam bahasa C dengan menggunakan struktur data yang sudah kalian pelajari di mata kuliah ini.

Kalian boleh menggunakan struktur data yang sudah kalian buat untuk praktikum pada tugas besar ini. Daftar ADT dapat dilihat pada bagian Daftar ADT. Kompilasi program wajib menggunakan Makefile dan wajib dapat dikompilasi dan dijalankan pada sistem operasi berbasis UNIX. Jika sistem operasi Anda adalah Windows, gunakan WSL untuk memiliki environment seperti UNIX. Contoh Makefile dapat dilihat pada Olympia. Perhatikan bahwa Makefile yang diberikan merupakan contoh, Anda dapat membuat Makefile yang lebih sesuai dengan kebutuhan Anda.

9.2 Notulen Rapat

NOTULENSI MEET 1 TUBES ALSTRUKDAT

26/10/2023

Attendance List :

1. Yusuf Ardian S.
2. Dhidit Abdi Aziz
3. Erdianti Wiga
4. Andhita Naura
5. Farel W.
6. Kristo Anugrah

Meeting Result :

A. Penggunaan ADT

- Inisialisasi :
 - Perlu kemampuan cari directory file
 - Ga pake ADT apapun
- Perintah :
 - Pakai ADT Mesin Kata
 - Perintah lebih ke main drivernya

- Pengguna :
 - Pakai ADT List/ADT Matrix untuk simpan username dan kata sandi

- Profil :
 - ADT Matrix
 - Foto profil :
 - ADT Matrix

- Teman :
 - ADT Graf

- Permintaan pertemanan :
 - ADT Graf

*Teman dan Permintaan Pertemanan dikerjain satu orang yang sama

- Kicauan :
 - ADT Datetime
 - ADT Priority Queue karena ditampilkannya berurut berdasarkan tanggal
 - ADT Queue

- Balasan :
 - ADT Matrix untuk simpan balasan
 - ADT Linked List

- Draf Kicauan :
 - ADT Queue

- Utas :
 - ADT Linked List

- Simpan dan Muat :

- Write file
- Save file
- Load file

B. Configuration

- Perannya sebagai pusat data, segala update akan masuk ke config
- Prioritas atas karena perlu config untuk mulai penggerjaan

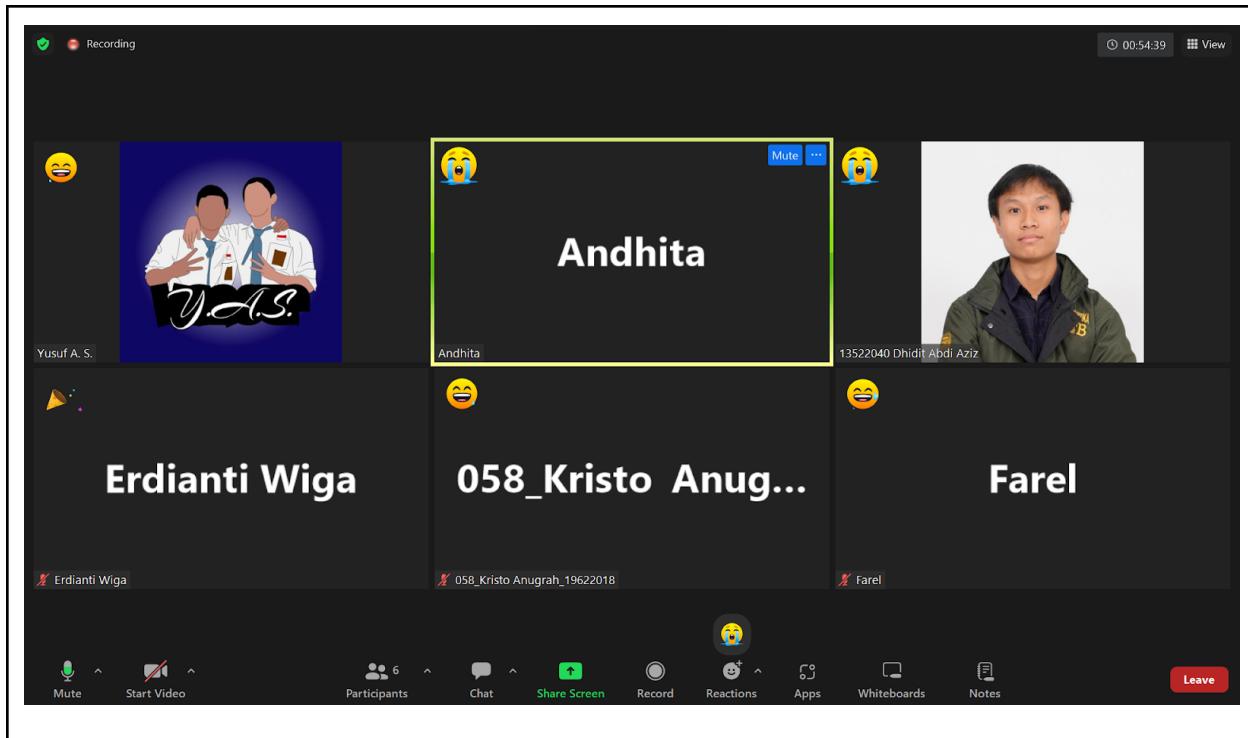
C. Asistensi

- Konfirmasi pemahaman kita ke asistennya
- Dipolling yaah tanggalnya

D. To Do List

1. Pahamin struktur programnya
2. Pahamin config
3. Polling tanggal asistensi
4. Buat ADT

Dokumentasi :



9.3 Log Activity Anggota Kelompok

NIM	W10	W11	W12	W13	W14
13522015	Struktur folder Github dan perancangan rencana pembuatan	ADT List Statik User	Program profil dan pengguna	Program profil dan pengguna	Driver dari semua ADT, integrasi profil dan pengguna ke main, pembuatan laporan
13522024	Merancang	Mencari tahu	Mencari tahu	Mengimpleme	Mengimpleme

	struktur data Tree di fitur balasan	detail open dan write file di bahasa C	signifikansi perbedaan CRLF dan LF di Windows dan Linux	ntasikan ADT Tree untuk fitur balasan	ntasikan fitur simpan.
13522040	Merancang dan membuat sketsa struktur data Program.	Membuat header dan realisasi ADT Matriks Pertemanan	Membuat header dan realisasi ADT Pertemanan	Mengintegrasikan ADT Matriks Pertemanan dan Permintaan Pertemanan ke dalam main.c	Membuat driver testing dari permintaan pertemanan dan membuat laporan.
13522047	Merancang struktur data untuk Draf	Membuat ADT Draf	Membuat pembacaan config ke variabel struktur data	Membuat beberapa program pembacaan config dan menginisialisasi main	Memasukkan ke main draf, fitur pertemanan, permintaan pertemanan, dan muat
13522053	Struktur folder Github dan perancangan rencana pembuatan	Convert ADT bawaan, modifikasi ADT Word Machine	ADT Linked List Utas	Program Utas	Integrasi program utas ke main, pembuatan laporan

13522060	Visualisasi perencanaan alur program	ADT List Dinamis Kicauan	Program List Dinamis Kicauan	Driver List Dinamis Kicauan dan Draf Kicauan	Integrasi program Kicauan ke main, pembuatan laporan
----------	--------------------------------------	--------------------------------	------------------------------------	--	--

9.4 Kendala

Seluruh anggota kelompok menggunakan laptop dengan Operating System Windows dalam proses pengembangan program sehingga pembuatan program, pembacaan config, hingga testing pada awalnya hanya dilakukan di atas Windows. Algoritma pada Windows memungkinkan penambahan /r diikuti dengan /n (Format pembacaan file CRLF) pada setiap akhir baris di file txt yang menjadi patokan utama dalam melakukan proses simpan dan muat konfigurasi. Akan tetapi, ketika program Burbir dijalankan pada OS Ubuntu, /r hilang sehingga tersisa /n (Format pembacaan file LF) yang mengakibatkan error apabila pemrosesan file konfigurasi dilakukan dengan program yang terlanjur dibuat di Windows. Setelah melakukan modifikasi dan sedikit perubahan, pada akhirnya kendala ini berhasil diselesaikan.

9.5 Milestone

Tanggal			16/11/2023
No	Fitur	Progress (0-100%)	Keterangan (Komponen fitur yang telah selesai ataupun kendala)
1.	Inisialisasi	60%	Selesai : - Inisialisasi ListPengguna dan ListKicauan

			<ul style="list-style-type: none"> - Inisialisasi if else sesuai command <p>Belum selesai :</p> <ul style="list-style-type: none"> - Variabel global yang lain
2.	Perintah	100%	<p>Selesai:</p> <p>ADT wordmachine yang sudah dimodifikasi menyesuaikan dengan spek tubes sehingga bisa menerima perintah.</p>
3.	Pengguna	80%	<p>Selesai :</p> <ul style="list-style-type: none"> - Fitur DAFTAR - Fitur MASUK - Fitur KELUAR - Fitur TUTUP_PROGRAM <p>Belum selesai :</p> <ul style="list-style-type: none"> - Belum bisa melihat realisasi karena belum diintegrasikan ke main - Belum diintegrasikan dengan variabel global
4.	Profil	90%	<p>Selesai:</p> <ul style="list-style-type: none"> - Profil.c - user.c <p>Belum selesai:</p> <ul style="list-style-type: none"> - Profil_driver.c - profil.c dan user.c belum di test

			<ul style="list-style-type: none"> - Menyelesaikan conflict liststatikuser.h, liststatikuser.c, dan user.c
6.	Teman	70%	<p>Selesai:</p> <ul style="list-style-type: none"> - friendmatrix.c - friendmatrix.h <p>Belum selesai:</p> <ul style="list-style-type: none"> - Driver untuk matrix pertemanan
7.	Permintaan Pertemanan	70%	<p>Selesai:</p> <ul style="list-style-type: none"> - prioreqfollinked.h - prioreqfollinked.h <p>Belum selesai:</p> <ul style="list-style-type: none"> - Driver untuk permintaan pertemanan - Menambahkan fungsi permintaan pertemanan kepada read config
8.	Kicauan	85%	<p>Selesai :</p> <ul style="list-style-type: none"> - ADT list dinamis khusus kicauan - Fitur KICAU done - Fitur KICAUAN hampir selesai, tinggal melihat kembali aplikasinya jika

			<p>dikaitkan dengan penggunaan FriendMatrix</p> <ul style="list-style-type: none"> - Fitur SUKA_KICAUAN hampir selesai, tinggal melihat kembali aplikasinya jika dikaitkan dengan penggunaan FriendMatrix - Fitur UBAH_KICAUAN done - Driver kicauan <p>Belum selesai :</p> <ul style="list-style-type: none"> - Belum bisa connect sama utas - Belum bisa melihat realisasi jika ada keterlibatan pengguna lain/belum bisa melihat penggunaan berbagai fitur kicauan dengan lebih dari satu user
9.	Balasan	95%	<p>Selesai:</p> <ul style="list-style-type: none"> - Perintah HAPUS_BALASAN - Perintah BALAS - Perintah BALASAN <p>Belum selesai:</p> <ul style="list-style-type: none"> - ListDinKicauan masih harus difinalisasi (masih conflict dengan ADT lain)

			<ul style="list-style-type: none"> - Helper untuk perintah simpan
10.	Draf Kicauan	80%	<p>Selesai :</p> <ul style="list-style-type: none"> - ADT stack untuk draf kicauan - Driver Draf - Fitur BUAT_DRAF - Fitur LIHAT_DRAF <p>Belum selesai :</p> <ul style="list-style-type: none"> - Belum bisa melihat realisasi penggunaan drive karena belum diintegrasikan ke main - Belum bisa melihat realisasi penerbitan draf menjadi kicauan
11.	Utas	80%	<p>Selesai :</p> <ul style="list-style-type: none"> - Fitur UTAS - Fitur SAMBUNG_UTAS - Fitur HAPUS_UTAS - Fitur CETAK_UTAS - Driver utas <p>Belum selesai :</p> <ul style="list-style-type: none"> - Belum bisa connect ke list global kicauan - Belum bisa melihat realisasi karena belum diintegrasikan ke main

12.	Simpan dan Muat	40%	<p>Selesai :</p> <ul style="list-style-type: none"> - Baca config pengguna - Baca config kicauan <p>Belum selesai :</p> <ul style="list-style-type: none"> - Baca config balasan nunggu yang lain - Baca config utas nunggu yang lain - Baca config draf - Simpan kicauan menunggu list matrix pertemanan, finalisasi matrix foto di pengguna.h - Simpan balasan hampir selesai (75%) - Simpan utas - Simpan draf
13.	Laporan	0%	Bahkan belum membuat docs nya hehe

9.6 Form Asistensi

terlampir

Form Asistensi Tugas Besar
IF2110/Algoritma dan Struktur Data
Sem. 1 2023/2024

No. Kelompok/Kelas : A / K 01
 Nama Kelompok :
 Anggota Kelompok (Nama/NIM) :
 1. Yusuf Ardian Sandi / 13522015
 2. Kristo Anugrah / 13522024
 3. Dhidit Abdi Aziz / 13522040
 4. Farel Winalda / 13522047
 5. Erdianti Wiga Putri Andini / 13522053
 6. Andhita Naura Haryanto / 13522060

Asisten Pembimbing : Primanda Adyatma Hafiz

Asistensi I

Tanggal: Kamis, 2 NOVEMBER 2023 Tempat: Upnormal Sumur Bandung Kehadiran Anggota Kelompok: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">No</th><th style="text-align: center;">NIM</th><th style="text-align: center;">Tanda tangan</th></tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td><td style="text-align: center;">13522015</td><td style="text-align: center;"></td></tr> <tr> <td style="text-align: center;">2</td><td style="text-align: center;">13522024</td><td style="text-align: center;"></td></tr> <tr> <td style="text-align: center;">3</td><td style="text-align: center;">13522040</td><td style="text-align: center;"></td></tr> <tr> <td style="text-align: center;">4</td><td style="text-align: center;">13522047</td><td style="text-align: center;"></td></tr> <tr> <td style="text-align: center;">5</td><td style="text-align: center;">13522053</td><td style="text-align: center;"></td></tr> <tr> <td style="text-align: center;">6</td><td style="text-align: center;">13522060</td><td style="text-align: center;"></td></tr> </tbody> </table>	No	NIM	Tanda tangan	1	13522015		2	13522024		3	13522040		4	13522047		5	13522053		6	13522060		Catatan Asistensi: <ul style="list-style-type: none"> 1) Di ADT Kicauan, list of address balaran buat pake ADT Tree 2) ID Balasan ga balik ke i, tp increment terus 3) Yang direbut tree cuma list of address balaran 4) Waktu saat ubah kicauan tetep 5) config hrs pake word machine 6) Utas sm kicauan beda ranah 7) No hp tipenya boleh word 8) ID user akies indeks di list of user ($index + 1$) 9) Tidak ada batasan jumlah karakter no hp dibaca pakai mesin karakter
No	NIM	Tanda tangan																				
1	13522015																					
2	13522024																					
3	13522040																					
4	13522047																					
5	13522053																					
6	13522060																					
*	Tanda Tangan Asisten: 																					

Form Asistensi Tugas Besar
IF2110/Algoritma dan Struktur Data
Sem. 1 2023/2024

No. Kelompok/Kelas : A / K1
 Nama Kelompok :
 Anggota Kelompok (Nama/NIM) :
 1. Yusuf Ardian Sandi / 13522015
 2. Dhidit Abdi Aziz / 13522040
 3. Erdianti Wiga Putri A / 13522053
 4. Andhita Naura Haryanto / 13522060
 5. Farzl Winilda / 13522017
 6.

Asisten Pembimbing : Primanda Adyatma Hafiz

Asistensi II

Tanggal: Kamis, 16 November 2023 Tempat: Lab Pemrograman Kehadiran Anggota Kelompok: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th><th>NIM</th><th>Tanda tangan</th></tr> </thead> <tbody> <tr> <td>1</td><td>13522015</td><td></td></tr> <tr> <td>2</td><td>13522040</td><td></td></tr> <tr> <td>3</td><td>13522053</td><td></td></tr> <tr> <td>4</td><td>13522060</td><td></td></tr> <tr> <td>5</td><td>13522017</td><td></td></tr> <tr> <td>6</td><td></td><td></td></tr> </tbody> </table>	No	NIM	Tanda tangan	1	13522015		2	13522040		3	13522053		4	13522060		5	13522017		6			Catatan Asistensi: <ol style="list-style-type: none"> 1. Membuat request follow di baca config 2. Membuat fungsi untuk debug hasil config
No	NIM	Tanda tangan																				
1	13522015																					
2	13522040																					
3	13522053																					
4	13522060																					
5	13522017																					
6																						
	Tanda Tangan Asisten: 																					