

LAPORAN TUGAS BESAR

IF2110 Algoritma dan Pemrograman 2

Groddit

Dipersiapkan oleh:

K03-A

Nathanael Shane Bennet - 13524119

Billy Ontoseno Irawan - 13524121

Nathaniel Christian - 13524122

Fazri Arrashyi Putra - 13524127

Safira Berlianti - 13524128

Muh Hartawan Haidir - 13524147

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen	Halaman
		<i>IF2110-TB-A-K03</i>	45
		Revisi	1

Daftar Isi

1	Ringkasan	3
2	Penjelasan Tambahan Spesifikasi Tugas	4
2.1	Save	5
2.2	Load	5
2.3	Advanced (Linear) Search	6
2.4	Following Recommendation	6
2.5	Content Moderation	7
3	Struktur Data (ADT)	7
3.1	ADT Sederhana	7
3.2	ADT List Berkait	9
3.3	Mesin Kata dan Mesin Karakter	10
3.4	ADT Graph	10
3.5	ADT Tree	11
4	Program Utama	12
4.1	Implementasi Mesin Karakter dan Mesin Kata	12
4.2	Implementasi Struktur Data Inti	12
4.2.1	Struktur User	12
4.2.2	Struktur Posting	13
4.2.3	Struktur Komentar	13
4.2.4	Struktur Subgroddit dan Linked List Post	14
4.2.5	ADT Graph	14
4.3	Implementasi Database dan Manajemen Data	15
4.3.1	Loading Data	16
4.3.2	Saving Data	16
4.4	Implementasi Command Processor	16
4.5	Implementasi Fitur Utama Groddit	17
4.5.1	Register User	17
4.5.2	Login dan Logout	17
4.5.3	Membuat Postingan	18
4.5.4	Menampilkan dan Menghapus postingan	18
4.5.5	Komentar pada Post	18
4.5.6	Voting	18
4.5.7	Follow	19
4.5.8	Subgroddit	19
4.5.9	Fitur Bantuan dan Validasi Umum	19
4.6	Error Handling dan Validasi	20
4.7	Implementasi Fitur Tambahan Groddit	20
4.7.1	Friend Recommendation	20
4.7.2	Advanced Search	21

4.7.3 Moderation System	21
4.8 Ringkasan Implementasi	22
5 Data Test	23
5.1 user.csv	23
5.2 subgroddit.csv	24
5.3 post.csv	24
5.4 comment.csv	25
5.5 voting.csv	26
5.6 social.csv	28
5.7 Blacklist.csv	28
6 Test Script	29
7 Pembagian Kerja dalam Kelompok	43
8 Lampiran	43
8.1 Deskripsi Tugas Besar 1	43
8.2 Notulen Rapat	44
8.3 Log Activity Anggota Kelompok	45

1 Ringkasan

Groddit adalah aplikasi media sosial berbasis *Command Line Interface* (CLI) yang memungkinkan pengguna untuk berinteraksi dalam subgroddit, membuat postingan, memberikan komentar berjenjang, serta membangun hubungan atau jejaring sosial melalui fitur *follow*. Proyek ini dibangun menggunakan bahasa Pemrograman C dengan pendekatan modular, memisahkan antara Tipe Data Abstrak (ADT), entitas data (entities), serta logika program. Dalam implementasinya, Groddit menggunakan beberapa struktur data penting, seperti *Graph* untuk jaringan pertemanan, *Tree* sebagai representasi komentar berjenjang, *Linked List*, serta Mesin kata sebagai tokenizer input. Seluruh fitur diintegrasikan melalui *command processor* yang menjalankan perintah pengguna. Selain fitur wajib, aplikasi ini juga dilengkapi dengan beberapa fitur “bonus” yaitu fitur Suggested Friends, fitur Advanced Search, dan Content Moderation. Laporan ini disusun untuk membahas mengenai hasil dari proses perancangan dan implementasi aplikasi Groddit. Laporan ini akan merangkum mengenai :

1. Deskripsi umum permasalahan dan tujuan implementasi.
2. Penggunaan ADT untuk menyelesaikan kebutuhan program.
3. Alasan pemilihan ADT.
4. Arsitektur perangkat lunak dan alur logika utama aplikasi.
5. Implementasi fitur sesuai spesifikasi tugas besar.
6. Kesimpulan terkait pencapaian dan hasil akhir.

Secara keseluruhan, aplikasi Groddit berhasil untuk diimplementasikan oleh struktur data yang tepat dan dibangun sesuai spesifikasi. Program dapat memuat konfigurasi awal, menjalankan berbagai perintah, dan menyimpan kembali seluruh data ke dalam folder konfigurasi.

2 Penjelasan Tambahan Spesifikasi Tugas

Pada proses pengembangan aplikasi Groddit, terdapat beberapa asumsi, batasan teknik, serta detail implementasi yang tidak dijelaskan secara eksplisit dalam dokumen deskripsi tugas

besar. Bagian ini memberikan penjelasan tambahan terkait bagaimana aplikasi diimplementasikan agar tetap konsisten dengan spesifikasi dan dapat berjalan secara stabil.

2.1 Save

Save adalah perintah untuk menyimpan seluruh data yang ada dalam sistem Groddit ke dalam berkas csv di folder tertentu. Setiap jenis data seperti pengguna, post, komentar, subgroddit, vote, dan hubungan sosial disimpan ke berkas terpisah dengan format yang terstruktur. Perintah ini dapat dijalankan kapan saja, baik saat pengguna sedang masuk (login) maupun saat belum masuk ke dalam akun.

Mekanisme penyimpanan bekerja dengan cara:

1. Membuka atau membuat berkas csv untuk setiap entitas (misal: users.csv, posts.csv, dll).
2. Menulis header kolom sesuai skema data.
3. Mengiterasi setiap data yang tersimpan di memori (dalam Database) dan menuliskannya ke berkas dalam format baris per record.
4. Untuk field yang mengandung koma atau karakter spesial (seperti judul dan konten post), data akan dibungkus dengan tanda kutip ganda ("") agar format csv tetap valid

2.2 Load

Load adalah perintah yang otomatis dijalankan (atau diwajibkan untuk dijalankan) saat program di-run untuk memuat data Groddit dari kumpulan berkas csv yang tersimpan di suatu folder ke dalam memori sistem. Setiap jenis data (pengguna, post, komentar, subgroddit, vote, dan hubungan sosial) dimuat dari berkas csv terpisah dan direkonstruksi ke dalam program. Perintah ini hanya dapat dijalankan saat pengguna belum login, untuk mencegah konflik atau ketidakakonsistennan data.

Mekanisme pemuatannya bekerja dengan cara:

1. Membuka berkas csv untuk setiap entitas (misal: users.csv, posts.csv, dll).
2. Membaca dan melewati baris header (jika ada).
3. Mengurai setiap baris data sesuai format kolom yang telah ditentukan.

4. Untuk field yang diapit tanda kutip ganda (seperti judul atau konten), sistem akan menangani karakter koma dan tanda kutip secara khusus agar tidak mengganggu pemisahan kolom.
5. Data yang berhasil dibaca akan dibuat menjadi objek internal (User, Post, Comment, dll.) dan ditambahkan ke struktur Database.
6. Hubungan antar data juga dibangun kembali, misalnya:
 - a. Post dikaitkan dengan subgroddit yang sesuai.
 - b. Vote dikaitkan dengan pengguna dan targetnya (post/komentar).
 - c. Graf sosial direkonstruksi berdasarkan data hubungan follow.

2.3 Advanced (*Linear*) Search

Fitur ini memungkinkan pengguna melakukan pencarian terhadap username, judul postingan, konten postingan, maupun nama Subgroddit secara sekaligus dalam satu perintah. (Fitur yang kami implementasikan disini tidak menggunakan tree). Proses pencarian dilakukan dengan cara:

1. Sistem menelusuri seluruh database (Subgroddits, Users, dan Posts) untuk mencari kecocokan substring dengan query yang diberikan.
2. Untuk setiap kategori, sistem melakukan iterasi dan membandingkan query dengan: Nama Subgroddit (subgroddits[i].name), Username pengguna (users[i].username), Judul post (posts[i].title) dan konten post (posts[i].content)
3. Hasil pencarian ditampilkan dalam tiga bagian terpisah: Subgroddits Found, Users Found, dan Posts Found.

2.4 Following Recommendation

Fitur ini digunakan untuk memberikan rekomendasi teman baru kepada pengguna berdasarkan hubungan pertemanan yang sudah ada. Sistem menggunakan pendekatan Breadth-First Search (BFS) pada graf relasi pertemanan untuk menemukan pengguna-pengguna yang berpotensi menjadi teman baru. Proses ini dilakukan dengan cara:

1. Program melakukan traversal BFS dari pengguna yang sedang login dengan depth maksimum 2-3 level.

2. Program mengumpulkan daftar pengguna yang belum di-follow oleh current_user, tetapi memiliki hubungan friends of friends (note: A teman B berarti A follow B dan B follow A).
3. Setiap pengguna dari hasil rekomendasi di-sort berdasarkan jumlah mutual connections yang dimiliki dengan current_user.
4. Program menampilkan hasil rekomendasi.

2.5 Content Moderation

Fitur ini digunakan untuk menjaga kualitas konten agar tetap sesuai dengan pedoman komunitas Groddit. Sistem akan melakukan penyaringan otomatis terhadap setiap post dan komentar yang akan dipublikasikan oleh pengguna. Daftar kata terlarang (blacklist words) disimpan dalam file konfigurasi blacklist.csv yang dimuat saat program dijalankan. Content moderation berlaku untuk dua jenis konten, yaitu post dan comment.

Proses validasi dilakukan dengan cara:

1. Program akan memeriksa seluruh isi konten post atau komentar menggunakan mesin kata.
2. Jika ditemukan satu atau lebih kata yang termasuk dalam daftar blacklist, maka Post atau komentar akan ditolak dan sistem akan menampilkan pesan bahwa konten mengandung kata terlarang.
3. Jika tidak ditemukan kata terlarang, maka konten diperbolehkan untuk disimpan dan ditampilkan.

3 Struktur Data (ADT)

Pada bagian ini, akan dijelaskan mengenai struktur data-struktur data (ADT) yang digunakan untuk membangun aplikasi Groddit, beserta dengan alasan mengenai pemilihan ADT tersebut sehingga dapat berhubungan dengan fitur-fitur yang ada.

3.1 ADT Sederhana

ADT sederhana direpresentasikan menggunakan struct C untuk entitas yang memiliki properti tetap dan tidak memerlukan dinamika kompleks. Contohnya :

- User (username, password, karma)

- Post (judul, isi, penulis, subgroddit, vote)
- Comment (id, parent, isi, penulis)
- Subgroddit (nama, list postingan)

Berikut adalah struktur data untuk,

File: voting/, user/, post/ ('/' artinya folder)

Struktur data:

```
typedef struct {
    char user_id[MAX_USER_ID_LEN];
    char username[MAX_USERNAME_LEN];
    char password[MAX_PASSWORD_LEN];
    int karma;
    char created_at[MAX_DATETIME_LEN];
} User;

typedef struct {
    char post_id[MAX_POST_ID_LEN];
    char subgroddit_id[MAX_POST_ID_LEN];
    char author_id[MAX_POST_ID_LEN];
    char title[MAX_TITLE_LEN];
    char content[MAX_CONTENT_LEN];
    char created_at[MAX_DATETIME_LEN];
    int upvotes;
    int downvotes;
} Post;

typedef struct {
    char user_id[MAX_USER_ID_LEN];
    TargetType target_type;
    char target_id[MAX_TARGET_ID_LEN];
    VoteType vote_type;
} Vote;
```

Persoalan yang diselesaikan:

- user/: menyimpan identitas pengguna.
- post/: menyimpan konten ‘post’ yang diberikan oleh user Groddit, ada tracking popularitas.
- voting/: mencatat vote user terhadap post dan comment.

Alasan pemilihan:

- user/: sebagai sebuah ‘entitas’, strukturnya sederhana, tidak ada hubungan kompleks.
- post/: sebagai sebuah ‘entitas’ yang berdiri sendiri, hanya reference pakai id.
- voting/: strukturnya sederhana.

3.2 ADT List Berkait

ADT List berkait digunakan untuk menyimpan data linier yang bersifat dinamis. Pada implementasi Groddit, list ini terutama digunakan untuk menyimpan daftar postingan pada setiap Subgroddit.

File : **linkedlist.h, linkedlist.c**

Struktur data :

```
Typedef struct ListNode *Address;
typedef struct ListNode {
    Int data;
    Address next;
}ListNode;
typedef Address List;
```

Persoalan yang diselesaikan :

1. Mengimplementasikan fitur subgroddit (subgroddit.h, subgroddit.c) yang bernama list post.

Alasan pemilihan :

1. Jumlah postingan pada sebuah Subgroddit tidak tetap
2. Insert / delete jauh lebih efisien dibanding array statis.
3. Struktur ringan dan mudah di transverse

3.3 Mesin Kata dan Mesin Karakter

File: mesinkar.c, mesinkar.h, mesinkata.c, mesinkata.h

Struktur data:

- Mesin Karakter:
 - CC (Current Character): karakter yang sedang dibaca
 - EOP (End of Process): penanda akhir input
- Mesin Kata:
 - CKata (Current Kata): struct berisi TabKata[NMax] dan Length
 - EndKata: penanda akhir kata
 - MARK = ';' sebagai delimiter

Persoalan yang diselesaikan:

- Membaca input pengguna dari CLI.
- Parsing load dan save file.

Alasan pemilihan:

- Sesuai spesifikasi tugas besar.

3.4 ADT Graph

Groddit menggunakan Graph dengan representasi adjacency matrix untuk menyimpan hubungan follow antar pengguna.

File: graph.c, graph.h

Struktur data:

```
typedef struct {  
    int numVertices;  
    User *vertices[MAX_GRAPH_VERTICES];  
    boolean  
    adjMatrix[MAX_GRAPH_VERTICES][MAX_GRAPH_VERTICES];  
    int vertexIndexMap[MAX_GRAPH_VERTICES];  
    int vertexCount;  
} Graph;
```

Persoalan yang diselesaikan:

- Mengimplementasikan fitur social (social.c, social.h) seperti follow, unfollow, followers, following.

Alasan pemilihan:

- Ada hubungan antar dua node, misalnya A follow B. Kemudian relasi A follow B tidak sama dengan B follow A (directed).
- Pada spek bonus ada penggunaan BFS untuk follow recommendation, sehingga dapat mengimplementasikan bonus tersebut.

3.5 ADT Tree

Groddit menggunakan struktur First Child - Next Sibling untuk merepresentasikan komentar dan balasan (reply) pada sebuah postingan.

File : tree.c, tree.h

Struktur data :

```
Typedef struct TreeNode *AddressTree;
typedef struct TreeNode{
    Comment *comment;
    AddressTree firstChild;
    AddressTree nextSibling;
    Int depth;
} TreeNode;
```

Persoalan yang diselesaikan :

- Mengimplementasikan fitur comment (comment.h, comment.c)

Alasan Pemilihan :

- Komentar media sosial bersifat hierarkis dengan kedalaman tidak terbatas.
- Model First Child - Next Sibling paling efisien untuk general tree.
- Memudahkan untuk Traversal komentar (preorder), Cascade deletion, Indentasi berdasarkan depth

4 Program Utama

Implementasi dilakukan menggunakan bahasa C dengan pendekatan modular, di mana setiap komponen (user, posting, komentar, subgroddit, graf sosial, mesin kata) diorganisasikan ke dalam direktori dan berkas yang terpisah untuk meningkatkan keterbacaan serta memudahkan proses pemeliharaan.

Program utama terdiri dari 3 lapisan utama :

1. Data load dan input: menggunakan Mesin Karakter dan Mesin Kata.
2. ‘Penyimpanan data’ dan ADT: mewakili seluruh entitas seperti User, Post, Comment, Subgroddit, Voting, Social, dan ADT yang terkait.
3. Command input dan algoritma program: interaksi pengguna melalui CLI dan menjalankan operasi sesuai perintah.

4.1 Implementasi Mesin Karakter dan Mesin Kata

Mesin karakter (mesinkar) dan mesin kata (mesinkata) bertanggung jawab untuk membaca input pengguna dan memecahnya menjadi token terstruktur. Implementasi mengikuti spesifikasi modul di spesifikasi, tetapi dengan beberapa penyesuaian:

Modifikasi utama

- Dengan Quoted String pada input berupa “teks dengan spasi”.
- Mark ditetapkan sebagai ; sehingga setiap instruksi CLI wajib diakhiri tanda titik koma.
- **STARTKATAFILE** dan **ADVKATAFILE** untuk proses pemuatan csv konfigurasi.

Dengan mekanisme ini, format perintah seperti :

```
CREATE_POST "Judul Contoh" "Isi konten yang panjang";
```

dapat diproses menjadi token yang benar tanpa kehilangan spasi.

4.2 Implementasi Struktur Data Inti

4.2.1 Struktur User

Struktur user menyimpan identitas pengguna, informasi akun, dan nilai karma:

```
Typedef struct {  
    char user_id[MAX_USER_ID_LEN];  
    char username[MAX_USERNAME_LEN];  
    char password[MAX_PASSWORD_LEN];  
    int karma;  
    char created_at[MAX_DATETIME_LEN];  
} User;
```

User disimpan dalam array statis pada Database, mendukung operasi seperti registrasi, autentikasi, dan perhitungan popularitas.

4.2.2 Struktur Posting

```
Typedef struct {  
    char post_id[MAX_POST_ID_LEN];  
    char subgroddit_id[MAX_POST_ID_LEN];  
    char author_id[MAX_POST_ID_LEN];  
    char title[MAX_TITLE_LEN];  
    char content[MAX_CONTENT_LEN];  
    char created_at[MAX_DATETIME_LEN];  
    int upvotes;  
    int downvotes;  
} Post;
```

Posting disimpan dalam array statis dan dihubungkan dengan Subgroddit melalui linked list berisi indeks posting.

4.2.3 Struktur Komentar

Komentar direpresentasikan dengan :

```
typedef struct {
```

```

int comment_id;
char post_id[MAX_COMMENT_ID_LEN];
char author_id[MAX_COMMENT_ID_LEN];
int parent_comment_id;
char content[MAX_CONTENT_LEN];
int upvotes;
int downvotes;
} Comment;

```

Struktur komentar dapat diintegrasikan dengan ADT Tree untuk menampilkan komentar berjenjang

4.2.4 Struktur Subgroddit dan Linked List Post

Subgroddit memiliki struktur :

```

typedef struct {
    char subgroddit_id[MAX_SUBGRODDIT_ID_LEN];
    char name[MAX_SUBGRODDIT_NAME_LEN];
    List post_list;
} Subgroddit;

```

Alasan Pemilihan ADT Linked List

1. Penambahan posting sangat sering → insertLast
2. Ukuran daftar tidak tetap dan dapat bertambah dinamis
3. Tidak membutuhkan random access

List ini hanya menyimpan indeks posting pada database, sehingga memori efisien dan integrasi antar modul terjaga.

4.2.5 ADT Graph

Graph digunakan untuk merepresentasikan hubungan “follow” antar pengguna.

Implementasi menggunakan *adjacency matrix* :

```
boolean adjMatrix[MAX][MAX];
```

Fitur Sosial yang diimplementasikan :

- FOLLOW, UNFOLLOW
- *Mutual friends*
- *Suggested friends* (berbasis eksplorasi Friend-of-a-Friend)
- Path antar pengguna menggunakan BFS/DFS
- *Most Popular users* berdasarkan in-degree

Adjacency matrix dipilih karena :

- Akses O(1) untuk mengecek relasi follow
- Jumlah user maksimum = 1000 (cukup efisien untuk matrix 1000 x 1000)

4.3 **Implementasi Database dan Manajemen Data**

Database utama menyatukan seluruh data pada program:

```
typedef struct {
    User users[MAX_USERS];
    int user_count;

    Post posts[MAX_POSTS];
    int post_count;

    Comment comments[MAX_COMMENTS];
    int comment_count;

    Subgroddit subgroddits[MAX_SUBGRODDITS];
    int subgroddit_count;

    Vote votes[MAX_VOTES];
    int vote_count;

    Social socials[MAX_SOCIALS];
    int social_count;

    Graph social_graph;
```

```
    char current_user_id[MAX_USER_ID_LEN];  
} Database;
```

4.3.1 Loading Data

Saat aplikasi dimulai :

- user.csv → memuat seluruh user
- post.csv → memuat posting
- comment.csv → memuat komentar
- subgorddit.csv → memuat subgorddit dan post_list
- social.csv → membentuk graf follow
- vote.csv → memuat riwayat voting

File dibaca menggunakan mesin kata untuk memastikan format konsisten dan aman dari error parsing.

4.3.2 Saving Data

Sebelum keluar (QUIT;), seluruh data ditulis kembali ke folder konfigurasi menggunakan format csv.

4.4 Implementasi Command Processor

Program menggunakan pola REPL (Read-Eval-Print Loop) :

1. READ → Mesin kata membaca instruksi sampai MARK (;)
2. EVAL → Command Processor mencocokkan input menggunakan IsKataEqual()
3. EXECUTE → Fungsi logika dipanggil dari modul terkait
4. PRINT → Output hasil operasi ditampilkan

Contoh validasi command:

```
(IsKataEqual(cmd, StringToKata("login"))){
```

```
cmdLogin(db);  
lse if (IsKataEqual(cmd, StringToKata("logout"))) {  
    cmdLogout(db);  
lse if (IsKataEqual(cmd, StringToKata("register"))) {  
    cmdRegister(db);  
...  
}
```

4.5 Implementasi Fitur Utama Groddit

Pada bagian ini akan dijelaskan mengenai setiap fitur utama dan tambahan yang terdapat pada aplikasi Groddit. Seluruh fitur dibangun dengan pendekatan modular menggunakan ADT dan modul yang telah dibuat.

4.5.1 Register User

Fitur Register memungkinkan pengguna baru untuk melakukan pendaftaran akun pada sistem, implementasi dilakukan pada fungsi `cmdRegister()`.

Ketika perintah ini dipanggil, maka mesin kata akan membaca parameter username dan password, kemudian melakukan validasi terhadap username yang sudah terdaftar pada database. Apabila belum, modul user akan membuat objek user baru dan menambahkannya ke dalam daftar user, serta menginisialisasi struktur data terkait seperti daftar post, daftar following, followers, dan riwayat voting. Validasi dilakukan untuk menangani kasus :

- Username kosong atau mengandung karakter tidak valid
- Username sudah digunakan

Validasi tersebut akan menjamin setiap pengguna yang terdaftar sehingga memiliki struktur data yang konsisten

4.5.2 Login dan Logout

Fitur login (`cmdLogin`) dan logout(`cmdLogout`) berfungsi mengelola konteks pengguna yang sedang aktif. Fitur login akan melakukan pencarian username dan password di database. Apabila cocok, maka pengguna bisa memasuki aplikasi tersebut. Akan tetapi, apabila tidak ditemukan username dan password apda database, maka pengguna tidak akan bisa memasuki aplikasi tersebut.

4.5.3 Membuat Postingan

Fitur CREATE_POST diimplementasikan melalui cmdCreatePost. Pengguna yang sedang login dapat membuat postingan dengan memasukkan judul dan isi posting. Data posting disimpan dalam ADT post dan ditambahkan ke daftar posting pengguna maupun daftar global database. Setiap postingan menyimpan ID Unik, Judul dan isi, pointer pemilik post, daftar komentar, dan informasi vote

4.5.4 Menampilkan dan Menghapus postingan

cmdViewPost menampilkan posting berdasarkan ID sehingga program dapat menampilkan isi post, jumlah vote, serta komentar dalam bentuk tree. Selain itu, cmdDeletePost memungkinkan pengguna menghapus posting miliknya. Proses penghapusan memastikan :

- Post dihapus dari daftar global
- Post dihapus dari daftar milik user
- Seluruh komentar dan subtree-nya ikut terhapus
- Riwayat voting terkait post dibersihkan

4.5.5 Komentar pada Post

Fitur komentar (cmdCreateComment) dan penghapusan komentar (cmdDeleteComment) menggunakan ADT Tree. Setiap komentar dapat direpresentasikan sebagai node tree, kemudian komentar balasan menjadi child dari komentar sebelumnya. Selain itu, ketika ditampilkan, tree ditelusuri untuk mencetak struktur komentar berjenjang. Di saat itu, algoritma selalu memastikan bahwa komentar hanya dapat dibuat oleh pengguna yang login, hanya dapat dihapus oleh pemilik, dan subtree komentar juga ikut terhapus jika induknya dihapus

4.5.6 Voting

Voting diterapkan pada post dan komentar melalui enam fungsi, yaitu cmdUpvotePost, cmdDownvotePost, cmdUndoVotePos, cmdUpvoteComment,

`cmdDownvoteComment`, `cmdUndoVoteComment`. Setiap entitas pada vote menyimpan status vote per pengguna. Validasi yang dilakukan oleh algoritma meliputi :

- Pengguna tidak dapat memberikan vote bertentangan tanpa melakukan undo
- Tidak boleh vote dua kali dengan jenis yang sama
- Perubahan vote langsung memperbarui skor post/komentar
-

4.5.7 *Follow*

Pada fitur follow, relasi sosial antar pengguna akan diimplementasikan melalui ADT graph. Pengguna dapat menambah follower atau pertemanan melalui fungsi `cmdFollow`, menghapus follower tersebut melalui fitur `cmdUnfollow`, melihat siapa saja kah yang berhubungan dengan kita melalui fitur `cmdFollowers`, dan fitur yang lain menyangkut tentang follow. Untuk mengoptimalkan fitur tersebut, algoritma akan disetting untuk melakukan validasi, yaitu tidak dapat follow diri sendiri, tidak dapat follow dua kali, dan tidak dapat unfollow jika belum follow

4.5.8 *Subgroddit*

Fitur Subgroddit direpresentasikan sebagai semacam “forum” atau “kategori”. Fitur tersebut mempunyai dua fungsi utama, yaitu :

- `cmdCreateSubgroddit` — membuat subgroddit baru
- `cmdViewSubgroddit` — menampilkan daftar subgroddit yang tersedia

Setiap post dapat terhubung ke subgroddit tertentu (Berdasarkan spesifikasi TB)

4.5.9 *Fitur Bantuan dan Validasi Umum*

Perintah `cmdHelp` menampilkan seluruh perintah yang didukung sistem.

Fitur ini terintegrasi dengan REPL dan hanya memanggil fungsi pencetakan teks.

Selain itu, seluruh fitur program dilengkapi validasi standar:

- Command harus diakhiri “;”
- Parameter minimal harus terpenuhi
- User harus login untuk fitur tertentu
- ID harus valid

4.6 Error Handling dan Validasi

Aplikasi memeriksa :

1. Login sebelum melakukan aksi tertentu
2. Posting hanya dibuat di subgroddit valid
3. Komentar hanya untuk posting yang ada
4. User tidak bisa follow diri sendiri
5. Nama subgroddit tervalidasi melalui `isValidSubgrodditName()`

4.7 Implementasi Fitur Tambahan Groddit

4.7.1 Friend Recommendation

Fitur Friend Recommendation adalah fitur tambahan dari sistem aplikasi Groddit yang memanfaatkan struktur graph dalam database. Tujuan dari fitur ini adalah memberikan rekomendasi teman kepada pengguna berdasarkan hubungan pertemanan tidak langsung atau tidak kedekatan dalam graph. Implementasi rekomendasi dilakukan sebagai berikut :

1. Representasi graph

Setiap user direpresentasikan sebagai node dan hubungan follow direpresentasikan sebagai edge terarah.

2. Identifikasi kandidat rekomendasi

Untuk pengguna aktif, program menelusuri

- Node-node yang di-follow oleh teman
- Node-node yang mem-follow teman
- Node-node yang terhubung pada kedalaman tertentu

3. Eksklusi

Pengguna berikut tidak dimasukkan ke dalam rekomendasi :

- Diri sendiri
- User yang sudah di-follow
- Duplikasi Hasil

4. Output

Sistem menampilkan daftar kandidat rekomendasi terurut berdasarkan skor atau kemunculan

4.7.2 Advanced Search

Fitur pencarian dasar memungkinkan pengguna mencari posting berdasarkan kata kunci. Namun pada implementasi bonus ini, fungsi diperluas untuk mendukung pencarian yang lebih fleksibel dan akurat. Perintah cmdSearch melakukan pencarian kata kunci pada judul maupun isi posting. Implementasinya mencakup :

- Menelusuri seluruh daftar post, user, komentar, subgroddit
- Menggunakan pencocokan string sederhana
- Menampilkan seluruh posting yang relevan

4.7.3 Moderation System

Fitur *Moderation* memungkinkan pencipta subgroddit atau pengguna berkepentingan untuk menjaga kualitas komunitas dengan menghapus posting atau komentar tertentu. Implementasi fitur moderasi meliputi:

1. Hak Moderasi

Hanya pengguna yang memiliki izin (misalnya pemilik subgroddit atau moderator tertentu) dapat melakukan moderasi. Sistem mengecek apakah currentUser berada dalam daftar moderator subgroddit tersebut.

2. Perintah Moderasi

Beberapa aksi moderasi yang didukung sistem:

- Menghapus posting dari subgroddit
- Menghapus komentar tertentu
- Menghapus atau menonaktifkan pengguna (jika diizinkan spesifikasi)
- Menampilkan daftar laporan atau post yang dilaporkan

3. Integrasi dengan ADT

Ketika moderator menghapus post:

- Node post dihapus dari daftar subgroddit
- Post dihapus dari list global database
- Tree komentar ikut dihapus

- Riwayat vote ikut dibersihkan

4. Validasi dan keamanan

Sistem memastikan :

- Moderator tidak bisa menghapus posting di subgroddit lain
- Moderator tidak bisa menghapus posting atau user yang tidak berada di ruang lingkup moderasinya
- Error message ditampilkan jika aksi moderasi tidak sah

Fitur moderasi diperlukan untuk menopang sistem komunitas seperti Groddit, sehingga menjaga interaksi tetap positif dan teratur.

4.8 Ringkasan Implementasi

Melalui implementasi modular menggunakan ADT buatan sendiri, Groddit berhasil menyediakan fitur media sosial lengkap berupa :

- Manajemen akun
- Jejaring sosial berbasis graf
- Posting dan komentar berjenjang
- Subgroddit dan datar posting dinamis
- Sistem voting
- Penyimpanan (load/save) ke/dari csv

Implementasi ini menegaskan pemanfaatan efektif struktur data Graph, Linked List, Tree, Mesin Kata, dan array statis untuk membangun aplikasi yang stabil, modular, dan sesuai spesifikasi.

5 Data Test

Laporan ini menggunakan data sample dari asisten.

5.1 user.csv

user_id	username	password	karma	created_at
USER001	kebinganteng	pass_kb_123	48	2025-11-01 21:10:00
USER002	bob	pass_bob_123	7	2025-11-02 21:10:00
USER003	drnefario	pass_nefario_123	99	2025-11-02 21:10:00
USER004	minion_dave	pass_dave_123	0	2025-11-03 21:10:00
USER005	minion_aved	pass_aved_123	0	2025-11-03 21:10:00
USER006	minion_veda	pass_veda_123	0	2025-11-03 21:10:00
USER007	alice	pass_alice_123	1250	2025-10-31 21:10:00

Fitur yang dites:

- LOAD data pengguna
- LOGIN
- REGISTER (validasi duplikasi)
- VIEW_PROFILE
- Perhitungan karma user

Skenario uji dan hasil yang diharapkan:

- Load user dari file → semua user berhasil dimuat
- Login user valid → login berhasil
- Login user salah password → login gagal

- Register username yang sudah ada → register ditolak
- Lihat profil user → profil menampilkan karma user

5.2 *subgroddit.csv*

subgroddit_id	name
S001	r/algorithms
S002	r/programming
S003	r/askreddit
S004	r/random

Fitur yang dites:

- LOAD Subgroddit
- VIEW_SUBGRODDIT
- CREATE_SUBGRODDIT (validasi duplikasi & prefix r/)

Skenario uji dan hasil yang diharapkan:

- Load subgroddit → semua subgroddit berhasil dimuat
- View subgroddit → post dari subgroddit ditampilkan
- Create subgroddit valid → subgroddit berhasil dibuat
- Create subgroddit dengan name yang sudah ada → ditolak (tidak unik)
- Create tanpa r/ → ditolak (format salah)

5.3 *post.csv*

post_id	subgroddit_id	author_id	title	content	created_at	upvotes	down votes
P001	S001	USER001	Perbedaan BFS dan DFS	Ada yang bisa jelaskan perbedaan dan kasus penggunaan	2025-11-02 21:10:10	45	3

				BFS serta DFS?			
P002	S002	USER007	Cara kerja Topological Sort	Konten untuk post P002...	2025-11-02 21:10:10	28	1
P003	S003	USER003	Optimasi Dijkstra vs A*	Konten untuk post P003...	2025-11-03 21:10:10	87	2
P004	S004	USER007	Best C textbooks ?	Konten untuk post P004...	2025-10-31 21:10:10	18	0

Fitur yang dites:

- LOAD Post
- VIEW_POST
- DELETE_POST
- UPVOTE_POST / DOWNVOTE_POST
- Hot Sorting

Skenario uji dan hasil yang diharapkan:

- Load post → semua post berhasil dimuat
- View post → post ditampilkan
- Delete post valid → post dihapus
- Delete post bukan milik sendiri → ditolak
- View hot post → urutan post berdasarkan vote
- Vote memengaruhi karma

5.4 comment.csv

comment_id	post_id	author_id	parent_comment_id	content	upvotes	downvotes
1	P001	USER007	-1	Wah, tubesnya	10	0

				asik banget!		
2	P001	USER002	1	Setuju, bagus untuk eksplorasi nih.	7	1
3	P001	USER003	2	Dih, Gaje.	12	0

Fitur yang dites:

- LOAD Comment
- VIEW_COMMENT
- Nested Comment (parent_comment_id)
- DELETE_COMMENT
- UPVOTE_COMMENT / DOWNVOTE_COMMENT

Skenario uji dan hasil yang diharapkan:

- Load komentar → semua komentar berhasil dimuat
- View komentar → komentar ditampilkan bertingkat
- Delete comment valid → komentar dihapus
- Delete post → semua komentar dari post yang dihapus ikut terhapus
- Upvote/downvote komentar → upvotes/downvotes komentar bertambah

5.5 voting.csv

user_id	target_type	target_id	vote_type
USER001	POST	P002	UPVOTE
USER002	POST	P001	UPVOTE
USER003	POST	P001	DOWNVOTE
USER004	POST	P002	UPVOTE
USER005	POST	P002	DOWNVOTE
USER006	COMMENT	1	UPVOTE

USER007	COMMENT	2	UPVOTE
USER001	COMMENT	3	UPVOTE
USER002	COMMENT	3	DOWNVOTE
USER003	COMMENT	2	UPVOTE
USER004	POST	P003	UPVOTE
USER005	POST	P003	UPVOTE
USER006	POST	P003	UPVOTE
USER007	POST	P003	DOWNVOTE
USER001	COMMENT	1	UPVOTE
USER002	COMMENT	1	DOWNVOTE
USER003	COMMENT	1	DOWNVOTE
USER004	POST	P004	UPVOTE
USER005	POST	P004	UPVOTE
USER006	COMMENT	3	DOWNVOTE

Fitur yang dites:

- LOAD Voting
- UPVOTE
- DOWNVOTE
- UNDO
- Perhitungan karma

Skenario uji dan hasil yang diharapkan:

- Load voting → semua vote berhasil dimuat
- Self-voting → ditolak
- Vote ganda → ditolak
- Undo → upvote/downvote dibatalkan

5.6 *social.csv*

user_id	following_id
USER003	USER007
USER001	USER007
USER007	USER001
USER007	USER004

Fitur yang dites:

- LOAD Graph Sosial
- FOLLOW
- UNFOLLOW
- FOLLOWERS & FOLLOWING

Skenario uji dan hasil yang diharapkan:

- Load relasi sosial → semua relasi sosial berhasil dimuat, graph terbentuk
- View followers/following → followers/following user ditampilkan
- Follow diri sendiri → ditolak
- Unfollow user yang belum difollow → ditolak

5.7 *Blacklist.csv*

[table redacted]

File ini berisi kata-kata terlarang dan tidak pantas. File ini digunakan untuk fitur content moderation.

Fitur yang dites:

- Content Moderation
- COMMENT
- POST

Skenario uji dan hasil yang diharapkan:

- Menambahkan komentar atau post yang mengandung kata-kata terlarang → ditolak

6 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Register	Mengecek apakah user dapat registrasi dengan username yang belum ada di csv	1. Load folder konfigurasi 2. Register dengan username yang belum terdaftar	>> register; Masukkan username: nimon; Masukkan kata sandi: 12345678;	Akun terdaftar, muncul pesan registrasi sukses	Akun terdaftar, muncul pesan registrasi sukses
2	Register	Mengecek apakah user dapat registrasi dengan username yang sudah ada di csv	1. Load folder konfigurasi 2. Register dengan username yang sudah terdaftar di csv	>> register; Masukkan username: kebinganteng; Masukkan kata sandi: 12345678;	Registrasi gagal, muncul pesan username telah terdaftar, pilih username lain	Registrasi gagal, muncul pesan username telah terdaftar, pilih username lain

3	Register	Mengecek apakah user dapat register apabila sudah login	1. Load folder konfigurasi 2. Login dengan username dan password yang valid 3. Register	>> login; Masukkan username: kebinganteng; Masukkan kata sandi: pass_kb_123; >> register; Masukkan username: minion; Masukkan kata sandi: password;	Registrasi gagal, muncul pesan bahwa tidak dapat registrasi dalam keadaan telah login	Registrasi gagal, muncul pesan bahwa tidak dapat registrasi dalam keadaan telah login
4	Login	Mengecek apakah user dapat login dengan username dan password valid	1. Load folder konfigurasi 2. Login dengan username dan password valid	>> login; Masukkan username: kebinganteng; Masukkan kata sandi: pass_kb_123;	Berhasil login sebagai kebinganteng, muncul pesan login sukses	Berhasil login sebagai kebinganteng, muncul pesan login sukses
5	Login	Mengecek apakah user dapat login dengan username atau password tidak valid	1. Load folder konfigurasi 2. Login dengan password yang salah	>> login; Masukkan username: kebinganteng; Masukkan kata sandi: 12345678	Login gagal, muncul pesan username atau password tidak valid	Login gagal, muncul pesan username atau password tidak valid

6	Login	Mengecek apakah user dapat login apabila sudah login	1. Load folder konfigurasi 2. Login sebagai kebinganteng 3. Login sebagai alice	>> login; Masukkan username: kebinganteng; Masukkan kata sandi: pass_kb_123; >> login; Masukkan username: alice; Masukkan kata sandi: pass_alice_123;	Login gagal, muncul pesan “You are not logged in.”	Login gagal, muncul pesan “You are not logged in.”
7	Logout	Mengecek apakah user dapat logout apabila sudah login	1. Load folder konfigurasi 2. Login sebagai kebinganteng 3. Logout	>> login; Masukkan username: kebinganteng; Masukkan kata sandi: pass_kb_123; >> logout;	Berhasil logout dari akun kebinganteng, muncul pesan logout sukses	Berhasil logout dari akun kebinganteng, muncul pesan logout sukses
8	Logout	Mengecek apakah user dapat logout apabila belum login	1. Load folder konfigurasi 2. Logout	>> logout;	Muncul pesan “You are not logged in.”	Muncul pesan “You are not logged in.”
9	Quit	Mengecek apakah fitur quit berfungsi	1. Load folder konfigurasi 2. Quit program	>> quit;	Keluar dari program	Keluar dari program
10	View Profile	Mengecek apakah user dapat	1. Load folder konfigurasi 2. Login	>> view_profile kebinganteng;	Muncul profil dari user kebinganteng	Muncul profil dari user kebinganteng

		melihat profil pengguna yang ada	3. Lihat profil kebinganteng		dengan data yang sesuai	dengan data yang sesuai
11	View Profile	Mengecek apakah user dapat melihat profil pengguna yang tidak ada	1. Load folder konfigurasi 2. Login 3. Lihat profil nimon (user tidak ada)	>> view_profile nimon;	Muncul pesan user tidak ditemukan	Muncul pesan user tidak ditemukan
12	Post	Mengecek apakah user dapat melakukan post pada subgreddit yang sudah ada	1. Load folder konfigurasi 2. Login 3. Ketik command “post;” 4. Masukkan nama subgreddit yang sudah ada 5. Masukkan judul post 6. Masukkan konten post	>> post; Masukkan nama Subgreddit: r/random; Masukkan nama subgreddit yang sudah ada Masukkan judul post: wleowle; Masukkan konten post: tes;	Post berhasil dibuat, muncul pesan post sukses	Post berhasil dibuat, muncul pesan post sukses
13	Post	Mengecek apakah user dapat melakukan post pada subgreddit yang belum ada	1. Load folder konfigurasi 2. Login 3. Ketik command “post;” 4. Masukkan nama subgreddit yang belum ada	>> post; Masukkan nama Subgreddit: r/alpro;	Post gagal, muncul pesan subgreddit tidak ditemukan, buat subgreddit terlebih dahulu	Post gagal, muncul pesan subgreddit tidak ditemukan, buat subgreddit terlebih dahulu
14	View Post	Mengecek apakah user dapat melihat postingan	1. Load folder konfigurasi 2. Login 3. Lihat post yang sudah ada	>> view_post P001;	Muncul post dengan formatting yang sesuai	Muncul post dengan formatting yang sesuai

		yang sudah ada				
15	View Post	Mengecek apakah user dapat melihat postingan yang belum ada	1. Load folder konfigurasi 2. Login 3. Lihat post yang belum ada	>> view_post P999;	Muncul pesan post tidak ditemukan	Muncul pesan post tidak ditemukan
16	Delete Post	Mengecek apakah user dapat menghapus postingan miliknya sendiri	1. Load folder konfigurasi 2. Login sebagai kebinganteng 3. Delete post dengan ID P001 4. Mengetik Y; setelah pesan konfirmasi muncul	>> delete_post P001; >> Y;	Muncul pesan delete sukses, post terhapus	Muncul pesan delete sukses, post terhapus
17	Delete Post	Mengecek apakah user dapat menghapus postingan yang bukan miliknya	1. Load folder konfigurasi 2. Login sebagai kebinganteng 3. Delete post dengan ID P002 (post milik alice)	>> delete_post P002;	Muncul pesan hanya pembuat post yang dapat menghapus postingan, post tidak terhapus	Muncul pesan hanya pembuat post yang dapat menghapus postingan, post tidak terhapus
18	Delete Post	Mengecek apakah user dapat menghapus postingan yang tidak ada	1. Load folder konfigurasi 2. Login sebagai kebinganteng 3. Delete post dengan ID P999 (post tidak ada)	>> delete_post P999;	Muncul pesan post tidak ditemukan	Muncul pesan post tidak ditemukan
19	Create Subgroddit	Mengecek apakah user dapat membuat	1. Load folder konfigurasi 2. Login	>> create_subgroddit r/tubes;	Subgroddit berhasil dibuat, muncul pesan sukses	Subgroddit berhasil dibuat, muncul pesan sukses

		subgroddit baru (valid)	3. Create subgroddit baru (penulisan valid)			
20	Create Subgroddit	Mengecek apakah user dapat membuat subgroddit dengan format yang tidak valid	1. Load folder konfigurasi 2. Login 3. Create subgroddit dengan penulisan tidak valid (tanpa diawali r/)	>> create_subgroddit nimons;	Muncul pesan nama subgroddit tidak valid, subgroddit gagal dibuat	Muncul pesan nama subgroddit tidak valid, subgroddit gagal dibuat
21	Create Subgroddit	Mengecek apakah user dapat membuat subgroddit yang sudah dibuat sebelumnya	1. Load folder konfigurasi 2. Login 3. Create subgroddit r/random (sudah dibuat sebelumnya)	>> create_subgroddit r/random;	Muncul pesan subgroddit sudah ada, buat subgroddit dengan nama lain	Muncul pesan subgroddit sudah ada, buat subgroddit dengan nama lain
22	View Subgroddit	Mengecek apakah fitur view subgroddit berfungsi	1. Load folder konfigurasi 2. Login 3. Lihat subgroddit yang sudah ada	>> view_subgrod dit r/algorithms HOT DECR;	Muncul subgroddit dengan mode dan urutan yang sesuai	Muncul subgroddit dengan data dan urutan yang sesuai
23	View Subgroddit	Mengecek apakah user dapat melihat subgroddit yang belum ada	1. Load folder konfigurasi 2. Login 3. Lihat subgroddit yang belum ada	>> view_subgrod dit r/itb NEW INCR;	Muncul pesan subgroddit tidak ditemukan	Muncul pesan subgroddit tidak ditemukan
24	View Subgroddit	Mengecek apakah user dapat melihat subgroddit dengan mode atau	1. Load folder konfigurasi 2. Login 3. Lihat subgroddit menggunakan mode yang tidak valid	>> view_subgrod dit r/algorithms COOL DECR;	Muncul pesan mode tidak dikenali	Muncul pesan mode tidak dikenali

		urutan yang tidak valid				
25	Comment	Mengecek apakah user dapat menambahkan komentar pada post	1. Load folder konfigurasi 2. Login 3. Tambahkan komentar ke postingan	>> comment P001 -1; Masukkan isi komentar: aku sih gatau ya;	Komentar berhasil ditambahkan ke postingan	Komentar berhasil ditambahkan ke postingan
26	Comment	Mengecek apakah user dapat menambahkan komentar pada komentar lain	1. Load folder konfigurasi 2. Login 3. Tambahkan komentar ke komentar lain	>> comment P001 3; Masukkan isi komentar: lu yang gaje;	Komentar berhasil ditambahkan ke komentar lain	Komentar berhasil ditambahkan ke komentar lain
27	Delete Comment	Mengecek apakah user dapat menghapus komentar miliknya sendiri	1. Load folder konfigurasi 2. Login sebagai bob 3. Hapus komentar milik bob	>> delete_comment P001 2;	Komentar beserta semua reply di bawahnya berhasil dihapus	Komentar beserta semua reply di bawahnya berhasil dihapus
28	Delete Comment	Mengecek apakah user dapat menghapus komentar milik user lain	1. Load folder konfigurasi 2. Login sebagai bob 3. Hapus komentar milik alice	>> delete_comment P001 1;	Komentar tidak terhapus, muncul pesan tidak bisa menghapus komentar milik orang lain	Komentar tidak terhapus, muncul pesan tidak bisa menghapus komentar milik orang lain
29	Delete Comment	Mengecek apakah user dapat menghapus komentar	1. Load folder konfigurasi 2. Login 3. Hapus komentar yang tidak ada	>> delete_comment P001 7;	Muncul pesan komentar tidak ditemukan	Muncul pesan komentar tidak ditemukan

		yang tidak ada				
30	Upvote Post	Mengecek apakah user tidak dapat memberikan vote pada postingan miliknya sendiri	1. Load folder konfigurasi 2. Login sebagai kebinganteng 3. Upvote post dengan ID P001 (milik kebinganteng)	>> upvote_post P001;	Muncul pesan “Anda tidak dapat memberikan vote pada post Anda sendiri!”	Muncul pesan “Anda tidak dapat memberikan vote pada post Anda sendiri!”
31	Upvote Post	Mengecek apakah user dapat memberikan vote pada postingan user lain yang belum divote	1. Load folder konfigurasi 2. Login sebagai kebinganteng 3. Upvote post dengan ID P004 (milik alice)	>> upvote_post P004;	Upvote berhasil, muncul pesan sukses, karma alice bertambah 1	Upvote berhasil, muncul pesan sukses, karma alice bertambah 1
32	Upvote Post	Mengecek apakah user dapat memberikan vote pada postingan user yang sudah diupvote	1. Load folder konfigurasi 2. Login sebagai kebinganteng 3. Upvote post dengan ID P002 (sudah diupvote sebelumnya)	>> upvote_post P002;	Muncul pesan post sudah diupvote, upvote dan karma tidak bertambah	Muncul pesan post sudah diupvote, upvote dan karma tidak bertambah
33	Upvote Post	Mengecek apakah user dapat memberikan vote pada postingan user yang sebelumnya di downvote	1. Load folder konfigurasi 2. Login sebagai drnefario 3. Upvote post dengan ID P001 (sudah didownvote sebelumnya)	>> upvote_post P001;	Downvote diubah menjadi upvote, mucul pesan sukses	Downvote diubah menjadi upvote, mucul pesan sukses

34	Undo vote post	Mengecek apakah user dapat membatalkan vote pada post yang belum diberi vote	1. Load folder konfigurasi 2. Login sebagai alice 3. Undo vote	>> undo_vote_post P001;	Muncul pesan user belum memberikan vote pada post	Muncul pesan user belum memberikan vote pada post
35	Undo vote post	Mengecek apakah user dapat membatalkan vote pada post yang telah diupvote	1. Load folder konfigurasi 2. Login sebagai alice 3. Upvote post dengan ID P002 4. Undo vote	>> upvote_post P001; >> undo_vote_post P001;	Upvote dihilangkan dari post, muncul pesan vote berhasil dibatalkan	Upvote dihilangkan dari post, muncul pesan vote berhasil dibatalkan
36	Undo vote post	Mengecek apakah user dapat membatalkan vote pada post yang telah didownvote	1. Load folder konfigurasi 2. Login sebagai alice 3. Downvote post dengan ID P002 4. Undo vote	>> downvote_post P001; >> undo_vote_post P001;	Downvote dihilangkan dari post, muncul pesan vote berhasil dibatalkan	Downvote dihilangkan dari post, muncul pesan vote berhasil dibatalkan
37	Downvote post	Mengecek apakah user tidak dapat memberikan vote pada postingan miliknya sendiri	1. Load folder konfigurasi 2. Login sebagai kebinganteng 3. Upvote post dengan ID P001 (milik kebinganteng)	>> downvote_post P001;	Muncul pesan “Anda tidak dapat memberikan vote pada post Anda sendiri!”	Muncul pesan “Anda tidak dapat memberikan vote pada post Anda sendiri!”
38	Downvote Post	Mengecek apakah user dapat memberikan vote pada	1. Load folder konfigurasi 2. Login sebagai bob 3. Downvote post dengan ID P001	>> downvote_post P001;	Upvote diubah menjadi downvote, muncul pesan sukses	Upvote diubah menjadi downvote, muncul pesan sukses

		postingan yang sebelumnya di upvote	(sudah diupvote sebelumnya)			
39	Upvote comment	Mengecek apakah user tidak dapat memberikan vote pada comment miliknya sendiri	1. Load folder konfigurasi 2. Login sebagai bob 3. Upvote comment 2 di post P001	>> upvote_comment P001 2;	Muncul pesan “Anda tidak dapat memberikan vote pada komentar Anda sendiri!”	Muncul pesan “Anda tidak dapat memberikan vote pada komentar Anda sendiri!”
40	Upvote comment	Mengecek apakah user dapat memberikan vote pada comment user lain	1. Load folder konfigurasi 2. Login sebagai minion_aved 3. Upvote comment 1 di post P001	>> upvote_comment P001 1;	Upvote berhasil, muncul pesan sukses, karma alice bertambah 1	Upvote berhasil, muncul pesan sukses, karma alice bertambah 1
41	Upvote comment	Mengecek apakah user dapat memberikan vote pada comment yang sudah diupvote	1. Load folder konfigurasi 2. Login sebagai minion_veda 3. Upvote comment 1 di post P001	>> upvote_comment P001 1;	Muncul pesan comment sudah diupvote, upvote dan karma tidak bertambah	Muncul pesan comment sudah diupvote, upvote dan karma tidak bertambah
42	Upvote comment	Mengecek apakah user dapat memberikan vote pada comment user yang sebelumnya didownvote	1. Load folder konfigurasi 2. Login sebagai minion_veda 3. Upvote comment 3 di post P001	>> upvote_comment P001 3;	Muncul pesan vote berhasil diubah menjadi upvote, upvote dan karma bertambah 2	Muncul pesan vote berhasil diubah menjadi upvote, upvote dan karma bertambah 2

43	Downvote comment	Mengecek apakah user tidak dapat memberikan vote pada comment miliknya sendiri	1. Load folder konfigurasi 2. Login sebagai bob 3. Downvote comment 2 di post P001	>> downvote_comment P001 2;	Muncul pesan "Anda tidak dapat memberikan vote pada komentar Anda sendiri!"	Muncul pesan "Anda tidak dapat memberikan vote pada komentar Anda sendiri!"
44	Downvote comment	Mengecek apakah user dapat memberikan vote pada comment yang sebelumnya di upvote	1. Load folder konfigurasi 2. Login sebagai minion_veda 3. Downvote comment 1 di post P001	>> downvote_comment P001 1;	Muncul pesan "Vote Anda pada komentar #1 di post [P001] berhasil diubah menjadi downvote."	Muncul pesan "Vote Anda pada komentar #1 di post [P001] berhasil diubah menjadi downvote."
45	Downvote comment	Mengecek apakah user dapat memberikan vote pada comment yang sudah didownvote	1. Load folder konfigurasi 2. Login sebagai bob 3. Downvote comment 1 di post P001	>> downvote_comment P001 1;	Muncul pesan "Anda sudah memberikan downvote pada komentar #1 di post [P001]."	Muncul pesan "Anda sudah memberikan downvote pada komentar #1 di post [P001]."
46	Undo vote comment	Mengecek apakah undo vote berhasil jika sebelumnya sudah upvote	1. Load folder konfigurasi 2. Login sebagai minion_veda 3. Undo vote comment 1 di post P001	>> undo_vote_comment P001 1;	Muncul pesan "Vote Anda pada komentar #1 di post [P001] berhasil dibatalkan."	Muncul pesan "Vote Anda pada komentar #1 di post [P001] berhasil dibatalkan."
47	Undo vote comment	Mengecek apakah undo vote berhasil jika sebelumnya	1. Load folder konfigurasi 2. Login sebagai minion_veda	>> undo_vote_comment P001 3;	Muncul pesan "Vote Anda pada komentar #3 di post [P001]	Muncul pesan "Vote Anda pada komentar #3 di post [P001]

		sudah downvote	3. Undo vote comment 3 di post P001		berhasil dibatalkan."	[P001] berhasil dibatalkan."
48	Undo vote comment	Mengecek apakah user tidak dapat undo vote jika belum pernah memberikan vote	1. Load folder konfigurasi 2. Login sebagai minion_dave 3. Undo vote comment 2 di post P001	>> undo_vote_comment P001 2;	Muncul pesan "Anda belum memberikan vote pada komentar #2 di post [P001]."	Muncul pesan "Anda belum memberikan vote pada komentar #2 di post [P001]."
49	Follow	Mengecek apakah user dapat follow user yang sebelumnya belum difollow	1. Load file konfigurasi 2. Login sebagai kebinganteng 3. Follow bob (belum difollow sebelumnya)	>> follow bob;	User berhasil difollow, muncul pesan sukses	User berhasil difollow, muncul pesan sukses
50	Follow	Mengecek apakah user dapat follow user yang sudah difollow	4. Load file konfigurasi 5. Login sebagai kebinganteng 6. Follow alice (sudah difollow sebelumnya)	>> follow alice;	Muncul pesan "Anda sudah mengikuti user ini"	Muncul pesan "Anda sudah mengikuti user ini"
51	Follow	Mengecek apakah user dapat follow dirinya sendiri	1. Load file konfigurasi 2. Login sebagai kebinganteng 3. Follow kebinganteng	>> follow kebinganteng;	Muncul pesan "Anda tidak dapat mengikuti diri sendiri."	Muncul pesan "Anda tidak dapat mengikuti diri sendiri."
52	Unfollow	Mengecek apakah user dapat unfollow user yang telah difollow	1. Load file konfigurasi 2. Login sebagai kebinganteng 3. Unfollow alice	>> unfollow alice	Berhasil unfollow user, muncul pesan sukses	Berhasil unfollow user, muncul pesan sukses
53	Unfollow	Mengecek apakah user	1. Load file konfigurasi	>> unfollow minion_dave	Muncul pesan "Anda belum	Muncul pesan "Anda belum

		dapat unfollow user yang belum difollow	2. Login sebagai kebinganteng 3. Unfollow minion_dave (belum difollow)		mengikuti user”	mengikuti user”
54	Followers	Mengecek apakah user dapat melihat followers	1. Load file konfigurasi 2. Login 3. Lihat followers user yang terdaftar	>> followers kebinganteng	Muncul daftar followers user	Muncul daftar followers user
55	Following	Mengecek apakah user dapat melihat following	1. Load file konfigurasi 2. Login 3. Lihat followers user yang terdaftar	>> following kebinganteng	Muncul daftar following user	Muncul daftar following user
56	Save	Mengecek apakah fitur save berfungsi	1. Load file konfigurasi 2. Lakukan apapun 3. Save	>> save <directory>;	Berhasil save data terbaru	Berhasil save data terbaru
57	Load	Mengecek apakah fitur load berfungsi	1. Load file konfigurasi	Masukan nama directory file konfigurasi	File konfigurasi berhasil dimuat	File konfigurasi berhasil dimuat
58	Following Recommentation	Mengecek apakah fitur friend recommendation berfungsi	1. Load file konfigurasi 2. Login 3. Cari friend recommendation	>> friend_recom mendation	Muncul daftar rekomendasi teman	Muncul daftar rekomendasi teman
59	Advanced Search	Mengecek apakah fitur advanced search berfungsi	1. Load file konfigurasi 2. Login 3. Search apapun	>> search keb;	Muncul subgroddit, user, dan/atau post yang awalan katanya sama dengan yang dicari	Muncul subgroddit, user, dan/atau post yang awalan katanya sama dengan yang dicari

60	Content Moderation	Mengecek apakah fitur content moderation berfungsi di post	1. Load file konfigurasi 2. Login 3. Buat post yang mengandung kata-kata tidak pantas	>> post; Masukkan nama subgreddit: r/random; Masukkan judul post: tes; Masukkan konten post: pemerintah sampah;	Post tidak berhasil dibuat, muncul pesan gagal	Post tidak berhasil dibuat, muncul pesan gagal
61	Content Moderation	Mengecek apakah fitur content moderation berfungsi di komentar	1. Load file konfigurasi 2. Login 3. Buat komentar yang mengandung kata-kata tidak pantas	>> comment P001 -1; Masukkan isi komentar: lu tolol;	Komentar tidak berhasil ditambahkan, muncul pesan gagal	Komentar tidak berhasil ditambahkan, muncul pesan gagal

7 Pembagian Kerja dalam Kelompok

No.	NIM	Pembagian Tugas	Status
1	13524119	Minor bug fix	Completed
2	13524121	Post, laporan	Completed
3	13524122	ADT (linked list - all drivers), Init, Load, Register, Subgreddit, Voting, Friend recommendation	Completed
4	13524127	ADT (graph, tree, mesinkata/mesinkar, sederhana), Save, Login/Logout, Perintah, Comment/reply	Completed
5	13524128	Social, content moderation, laporan	Completed
6	13524147	Laporan, advanced search	Completed

8 Lampiran

8.1 Deskripsi Tugas Besar 1

Buatlah sebuah program simulasi yang berjalan melalui CLI (command-line interface). Program harus ditulis dalam bahasa C dan memanfaatkan ADT yang telah dipelajari pada mata kuliah ini. Kalian diperbolehkan memakai ADT yang sebelumnya telah dibuat untuk praktikum sebagai bagian dari tugas besar ini. Daftar ADT yang dapat digunakan tercantum pada bagian *Daftar ADT*.

Program wajib dikompilasi menggunakan Makefile, dan hasilnya harus bisa dikompilasi serta dijalankan pada sistem operasi berbasis UNIX. Jika kalian menggunakan Windows, gunakan WSL agar mendapatkan lingkungan yang sesuai dengan UNIX.

Contoh Makefile tersedia di Template GitHub Classroom. Perlu diingat bahwa Makefile tersebut hanya contoh; kalian boleh memodifikasinya atau membuat Makefile baru yang lebih cocok dengan kebutuhan program kalian.

8.2 Notulen Rapat

Notulen Asistensi 1:

- ADT sederhana itu adt kayak point, user
- ADT di spek bagian list paling atas semuanya harus diimplementasikan semua, kalo ada yg ga di implementasi nilainya minus
- Batasan library ada di spek
- Format output ga harus persis di spek, yang penting inti/datanya sama dan jelas
- Program di simpan ke memori dulu, ga langsung ke csv, csv boleh diakses pas load/save doang
- Kalo ada yg mau ditanyain langsung chat kakaknya aja
- Disarankan struktur foldernya kaya disini
https://github.com/ValentinoTriadi/TugasBesar_IF2110_2023_A/tree/main/src/ADT
- Load (inisialisasi) harus minta input dari folder mana
- Disarankan pake branching

Notulen Asistensi 2:

- Kalo pencet enter ngaruh ke post, kemungkinan masalah di mesin kata, kalo terpaksa gpp tp itu miss dari spek.
- Disaranin kerjain voting dulu baru comment.
- Kalo emang fitur yg blm ga selesai jgn dipaksain masukin ke main.
- Laporan ga harus terlalu detail, tp kalo misal kakaknya ada pertanyaan/sesuatu yang ga jelas bisa langsung diliat di laporan.
- Pembagian kerja ditulis apa adanya, sesuai dengan kondisi state terakhir.

8.3 Log Activity Anggota Kelompok

Tanggal	NIM	Aktivitas	Fitur Terkait
12/11/25	13524122	Implementasi linkedlist	ADT list berkait
14/11/25	13524127	Implementasi ADT biasa, mesin kata, command awal (barebone)	ADT mesin karakter/kata Fitur 2: Perintah

15/11/25	13524127	Implementasi profil, save and load	Fitur 4: Profile, Fitur 10: Save dan Load
26/11/25	13524122	Implementasi inisialisasi	Fitur 1: Inisialisasi
28/11/25	13524122	Implementasi subgroddit	Fitur 6: Subgroddit
28/11/25	13524121	Implementasi post	Fitur 5: Post
28/11/25	13524122	Implementasi pengguna	Fitur 3: Pengguna
2/12/25	13524128	Implementasi social	Fitur 9: Social
3/12/25	13524127	Implementasi comment and reply	Fitur 7: Comment dan Reply
3/12/25	13524122	Implementasi voting	Fitur 8: Voting
6/12/25	13524128	Implementasi bonus: Content moderation	Fitur bonus: Content moderation
6/12/25	13524147	Implementasi bonus: Advanced search	Fitur bonus: Advanced search
6/12/25	13524122	Implementasi bonus: Friend Recommendation	Fitur bonus: Friend Recommendation
6/12/25	13524121, 13524122, 13524128, 13524147	Laporan	Laporan