— incase kalo readme gabisa dibaca, soalnya aku liat sangat ga terbaca 😀

```
[TUTS - Struktur Data]

Identitas Pengumpul

Nama: Abidah Fatimatuzzahrah
NIM: 103122400004
Kelas: SE08-01

1. Kode Program

Berikut adalah kode program untuk Aplikasi my TelU menyimpan data
postingan.

---------------- subprogram.cpp
#include "header.h" //sesuaikan dengan file header yang dibuat


address createElement (infotype data){
    address p = allocate(data);
    return p;
} //fungsi untuk membuat elemen baru

address allocate(infotype data){
    address p = new element; //alokasi memori
    info(p) = data; //mengisi info elemen dengan data
    next(p) = NIL; //inisialisasi next elemen dengan NIL
    prev(p) = NIL; //inisialisasi prev elemen dengan NIL
    return p; //mengembalikan alamat elemen yang telah dialokasikan
} //fungsi untuk mengalokasikan memori elemen baru

void createList(List &L) {
    first(L) = NIL; //inisialisasi list kosong
    last(L) = NIL; //inisialisasi list kosong
} //fungsi untuk membuat list kosong


List createNewList(){
    List a; //membuat variabel list baru
    createList(a); //menginisialisasi list baru tersebut
    return a; //mengembalikan list baru
} //fungsi untuk membuat list baru dan mengembalikan list tersebut
```

```
bool isEmpty (List a){
   return first(a) == NIL && last(a) == NIL; //mengembalikan true jika
list kosong
} //fungsi untuk mengecek apakah list kosong

//silahkan uncomment untuk NIM ganjil
void insertFirst (List &a, address p){
    if (isEmpty(a)){
        first(a) = p; //jika list kosong, first dan last sama dengan p
        last(a) = p; //karena hanya ada 1 elemen
    }else{
        next(p) = first(a); //jika list tidak kosong, next p menunjuk
ke first
        prev(first(a)) = p; //prev first menunjuk ke p
        first(a) = p; //first diupdate menjadi p
    }
} //fungsi untuk memasukkan elemen di awal list

void insertAfter(List &a, infotype x, address p){
    address Q = findElement(a, x); //mencari elemen dengan info x
    if (Q != NIL){ //jika elemen ditemukan
        if (Q == last(a)){ //jika elemen adalah last
            insertLast(a, p); //p dimasukkan di akhir list
        }else{
            next(p) = next(Q); //next p menunjuk ke next Q
            prev(next(Q)) = p; //prev next Q menunjuk ke p
            next(Q) = p; //next Q diupdate menjadi p
            prev(p) = Q; //prev p diupdate menjadi Q
        }
    } else {
        cout << "elemen dengan info " << x.id << " tidak ditemukan" <<
endl;
    } //jika elemen tidak ditemukan
} //fungsi untuk memasukkan elemen setelah elemen dengan info x


void insertLast(List &a, address p){
    if (isEmpty(a)){ //jika list kosong
        first(a) = p;
        last(a) = p; //first dan last sama dengan p
    }else{
```

```
        prev(p) = last(a); //jika list tidak kosong, prev p menunjuk ke
last
        next(last(a)) = p; //next last menunjuk ke p
        last(a) = p; //last diupdate menjadi p
    }
} //fungsi untuk memasukkan elemen di akhir list

void insertSorted(List &L, address P) {
    if (isEmpty(L)) { //jika list kosong
        insertFirst(L, P); //masukkan di awal
    } else if (info(P).id < info(first(L)).id) { //jika id P lebih
kecil dari first
        insertFirst(L, P); //masukkan di awal
    } else if (info(P).id > info(last(L)).id) { //jika id P lebih besar
dari last
        insertLast(L, P); //masukkan di akhir
    } else {
        address Q = first(L); //mulai dari first
        while (Q != NIL && info(P).id > info(Q).id) { //cari posisi
yang tepat
            Q = next(Q); //lanjut ke next
        }
        address before = prev(Q); //elemen sebelum Q
        next(before) = P; //sisipkan P di antara before dan Q
        prev(P) = before; //atur prev P
        next(P) = Q; //atur next P
        prev(Q) = P; //atur prev Q
    }
} //fungsi untuk memasukkan elemen secara terurut berdasarkan id

void showByUsername(List L, string uname) {
    address P = first(L); //mulai dari first
    bool found = false; //flag untuk menandai apakah ada postingan
ditemukan
    while (P != NIL) {
        if (info(P).username == uname) { //jika username cocok
            cout << "ID: " << info(P).id  //tampilkan info postingan
                 << ", Konten: " << info(P).konten
                 << ", Like: " << info(P).like
                 << ", User: " << info(P).username << endl;
            found = true; //tandai bahwa ada postingan ditemukan
        } //jika username tidak cocok
        P = next(P); //lanjut ke next
```

```
    }
    if (!found) {
        cout << "Tidak ada postingan dari " << uname << endl;
    } //jika tidak ada postingan ditemukan
} //fungsi untuk menampilkan semua post berdasarkan username

void showMostLiked(List L) {
    if (isEmpty(L)) { //jika list kosong
        cout << "List kosong" << endl; //tampilkan pesan
        return;
    } //jika list tidak kosong
    address P = first(L);  //mulai dari first
    address maxP = P; //inisialisasi maxP dengan first
    while (P != NIL) {
        if (info(P).like > info(maxP).like) { //jika like P lebih besar
dari maxP
            maxP = P; //update maxP
        }
        P = next(P); //lanjut ke next
    }
    cout << "Post Terpopuler:" << endl; //tampilkan info post dengan
like terbanyak
    cout << "ID: " << info(maxP).id
         << ", Konten: " << info(maxP).konten
         << ", Like: " << info(maxP).like
         << ", User: " << info(maxP).username << endl;
} //fungsi untuk menampilkan post dengan like terbanyak

void updateLike(List &L, int id, bool like) { //like = true untuk like,
false untuk unlike
    address P = first(L);
    while (P != NIL && info(P).id != id) { //cari elemen dengan id
        P = next(P); //lanjut ke next
    }
    if (P != NIL) { //jika elemen ditemukan
        if (like) info(P).like++; //jika like true, tambahkan like
        else info(P).like--; //jika like false, kurangi like
    } else {
        cout << "Post ID " << id << " tidak ditemukan" << endl; //jika
elemen tidak ditemukan
    }
} //fungsi untuk mengupdate like atau unlike pada post dengan id
tertentu
```

```
void deleteFirst(List &a, address p){
    if (isEmpty(a)) { //jika list kosong
        cout << "list kosong" << endl; //tampilkan pesan
    } else if (first(a) != last(a)){ //jika lebih dari 1 elemen
        p = first(a); //simpan alamat first ke p
        first(a) = next(p); //update first ke next p
        prev(first(a)) = NIL; //menghapus link prev first
        next(p) = NIL; //menghapus link next p
    } else { //tinggal 1 elemen
        p = first(a); //simpan alamat first ke p
        first(a) = NIL; //update first dan last ke NIL
        last(a) = NIL;
        next(p) = NIL;
        prev(p) = NIL;
    }
} //fungsi untuk menghapus elemen pertama list


void deleteLast(List &a, address p){
    if (isEmpty(a)) {
        cout << "list kosong" << endl; //jika list kosong
    } else if (next(first(a))==NIL){ //jika tinggal 1 elemen
        p = first(a); //simpan alamat first ke p
        first(a) = NIL; //update first dan last ke NIL
        last(a) = NIL;
        next(p) = NIL;
        prev(p) = NIL;
    }else{ //jika lebih dari 1 elemen
        p = last(a); //simpan alamat last ke p
        last(a) = prev(p); //update last ke prev p
        prev(p) = NIL; //menghapus link prev p
        next(last(a)) = NIL; //menghapus link next last
    }

} //fungsi untuk menghapus elemen terakhir list

int length(List a) {
    int jml = 1; //inisialisasi jumlah elemen
    address p = first(a); //mulai dari first
    if(p == NIL){ //jika list kosong
```

```
       return 0;
    }else {
       while(p != last(a)){ //iterasi sampai last
           jml++;
           p = next(p);
       }
       return jml; //mengembalikan jumlah elemen
    }
}


address findElement(List a, infotype x) { //mencari elemen dengan info
x
    if (isEmpty(a)) { //jika list kosong
        return NIL;
    } else { //jika list tidak kosong
        address Q = first(a); //mulai dari first
        while (Q != NIL && info(Q).id != x.id) { //cari elemen dengan
info x
            Q = next(Q); //lanjut ke next
        }
        return Q;
    }
} //fungsi untuk mencari elemen dengan info x


void printList(List a) {
    address p = first(a); //mulai dari first
    if (isEmpty(a)) { //jika list kosong
        cout << "List kosong" << endl;
    } else { //jika list tidak kosong
        cout << "Daftar Postingan:" << endl;
        while (p != NIL) { //iterasi sampai akhir list
            cout << "ID: " << info(p).id
                 << ", Konten: " << info(p).konten
                 << ", Like: " << info(p).like
                 << ", Username: " << info(p).username
                 << endl;
            p = next(p);
        }
    }
} //fungsi untuk menampilkan semua elemen dalam list
```

```
-------------------------- main.cpp
#include "header.h"
#include <iostream>
using namespace std;
//103122400004 Abidah Fatimatuzzahrah

void garis() {
    cout << "=======================================" << endl;
}

int main() {
    List L;
    createList(L);

    int pilihan;
    do {
        garis();
        cout << "        Aplikasi My Tel-U Timeline      " << endl;
        garis();
        cout << "1. Tambah Postingan Baru (terurut)" << endl;
        cout << "2. Tampilkan Semua Postingan" << endl;
        cout << "3. Tampilkan Postingan Berdasarkan Username" << endl;
        cout << "4. Tampilkan Postingan Terpopuler" << endl;
        cout << "5. Update Like (Like / Unlike)" << endl;
        cout << "0. Keluar" << endl;
        garis();
        cout << "Pilih menu: ";
        cin >> pilihan;
        cout << endl;

        if (pilihan == 1) {
            infotype data;
            cout << "Masukkan ID Post     : ";
            cin >> data.id;
            cin.ignore();
            cout << "Masukkan Konten Post : ";
            getline(cin, data.konten);
            cout << "Masukkan Username    : ";
            getline(cin, data.username);
            data.like = 0;

            address P = allocate(data);
```

```
            insertSorted(L, P);
            cout << "Postingan berhasil ditambahkan!" << endl;

        } else if (pilihan == 2) {
            printList(L);

        } else if (pilihan == 3) {
            string uname;
            cout << "Masukkan username yang dicari: ";
            cin.ignore();
            getline(cin, uname);
            showByUsername(L, uname);

        } else if (pilihan == 4) {
            showMostLiked(L);

        } else if (pilihan == 5) {
            int id;
            char aksi;
            cout << "Masukkan ID Post: ";
            cin >> id;
            cout << "Like (+) atau Unlike (-): ";
            cin >> aksi;

            if (aksi == '+') updateLike(L, id, true);
            else if (aksi == '-') updateLike(L, id, false);
            else cout << "Input tidak valid." << endl;

        } else if (pilihan == 0) {
            cout << "Keluar dari aplikasi..." << endl;

        } else {
            cout << "Pilihan tidak valid!" << endl;
        }

        cout << endl;
    } while (pilihan != 0);

    return 0;
}


----------------------------------header.cpp
```

```
#ifndef HEADER_H
#define HEADER_H
#include <iostream>
#include <string>
using namespace std;

#define first(L) (L).first
#define last(L) (L).last
#define next(P) (P)->next
#define prev(P) (P)->prev
#define info(P) (P)->info
#define NIL NULL

struct element;
typedef element* address;

struct infotype {
    int id;
    string konten;
    int like;
    string username;
};

struct element {
    infotype info;
    address next;
    address prev;
};

struct List {
    address first;
    address last;
};

void createList(List &L);
List createNewList();
address allocate(infotype data);
bool isEmpty(List a);
void insertFirst(List &a, address p);
void insertLast(List &a, address p);
void deleteFirst(List &a, address p);
void deleteLast(List &a, address p);
address findElement(List a, infotype x);
```

```
void printList(List a);
int length(List a);

void insertSorted(List &a, address P); //masukin terurut berdasarkan id
void showByUsername(List a, string uname); //tampilkan semua post
berdasarkan username
void showMostLiked(List a); //tampilkan post dengan like terbanyak
void updateLike(List &a, int id, bool like); //like = true untuk like,
false untuk unlike

#endif
```

2. Penjelasan Kode

a. Struktur Program
* header.h: berisi struct infotype, element, dan List, makro pointer, serta deklarasi semua fungsi.
* subprogram.cpp: berisi implementasi fungsi untuk membuat list, tambah, hapus, cari, tampil, dan update data.
* main.cpp: program utama dengan menu interaktif untuk menjalankan semua fitur.

b. Fungsi Utama
* createList: inisialisasi list kosong.
* allocate: buat node baru.
* insertSorted: tambah data terurut berdasarkan ID.
* showByUsername: tampilkan postingan sesuai username.
* showMostLiked: tampilkan postingan dengan like terbanyak.
* updateLike: tambah atau kurangi jumlah like.
* printList: tampilkan semua isi list.

c. Alur Program
   User memilih menu.
   Program membuat list kosong.
   User bisa menambah posting baru yang disimpan terurut.
   Data bisa dilihat semua, difilter berdasarkan username, atau ditampilkan yang paling populer.
   Jumlah like bisa diubah sesuai input user.

```
```

3. Output Program

Berikut adalah hasil eksekusi program (output) ketika dijalankan.
```



```
PS D:\StrukturData\103122400004_AbidahF_Week9_UTS\TEMPLATE\DLL> ./main
==
        Aplikasi My Tel-U Timeline
==
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
==
Pilih menu: 1

Masukkan ID Post   : 123
Masukkan Konten Post : halo
Masukkan Username   : haloHere
Postingan berhasil ditambahkan!

==
        Aplikasi My Tel-U Timeline
==
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
==
Pilih menu: 1

Masukkan ID Post   : 124
Masukkan Konten Post : hi
Masukkan Username   : hiHere
Postingan berhasil ditambahkan!

==
        Aplikasi My Tel-U Timeline
```
```

```
========================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 1

Masukkan ID Post     : 125
Masukkan Konten Post : piw
Masukkan Username    : piwHere
Postingan berhasil ditambahkan!

========================================
        Aplikasi My Tel-U Timeline
========================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 2

Daftar Postingan:
ID: 123, Konten: halo, Like: 0, Username: haloHere
ID: 124, Konten: hi, Like: 0, Username: hiHere
ID: 125, Konten: piw, Like: 0, Username: piwHere

========================================
        Aplikasi My Tel-U Timeline
========================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 3
```

```
Masukkan username yang dicari: hiHere
ID: 124, Konten: hi, Like: 0, User: hiHere

========================================
        Aplikasi My Tel-U Timeline
========================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 4

Post Terpopuler:
ID: 123, Konten: halo, Like: 0, User: haloHere

========================================
        Aplikasi My Tel-U Timeline
========================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 5

Masukkan ID Post: 124
Like (+) atau Unlike (-): +

========================================
        Aplikasi My Tel-U Timeline
========================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
```

```
========================================
Pilih menu: 5

Masukkan ID Post: 124
Like (+) atau Unlike (-): +

========================================
        Aplikasi My Tel-U Timeline
========================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 4

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 4

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere

========================================
        Aplikasi My Tel-U Timeline
========================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu:
```

```
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
=======================================
Pilih menu: 4

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere

=======================================
        Aplikasi My Tel-U Timeline
=======================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
=======================================
Pilih menu:




4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
=======================================
Pilih menu: 4

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere

=======================================
        Aplikasi My Tel-U Timeline
=======================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
```

```
0. Keluar
========================================
Pilih menu:
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 4

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere

========================================
        Aplikasi My Tel-U Timeline
========================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 4

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere

========================================
        Aplikasi My Tel-U Timeline
========================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 4

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere
```

```
========================================
        Aplikasi My Tel-U Timeline
========================================
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 4

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere

========================================
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 4

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere

5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu: 4

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere
========================================
Pilih menu: 4

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere

Post Terpopuler:
ID: 124, Konten: hi, Like: 2, User: hiHere
ID: 124, Konten: hi, Like: 2, User: hiHere



========================================
        Aplikasi My Tel-U Timeline
        Aplikasi My Tel-U Timeline
```

```
========================================
========================================
1. Tambah Postingan Baru (terurut)
2. Tampilkan Semua Postingan
3. Tampilkan Postingan Berdasarkan Username
4. Tampilkan Postingan Terpopuler
5. Update Like (Like / Unlike)
5. Update Like (Like / Unlike)
0. Keluar
5. Update Like (Like / Unlike)
5. Update Like (Like / Unlike)
0. Keluar
========================================
Pilih menu:
```

4. Penjelasan Lanjutan (Analisis Output)

Sesuai permintaan Anda ("Penjelasan Lagi"), bagian ini menganalisis mengapa output yang dihasilkan sudah benar:

* Pada Output 1: User menambah beberapa posting dengan ID berbeda. Program menampilkan data secara urut berdasarkan ID karena fungsi insertSorted menyisipkan node sesuai posisi ID terkecil ke terbesar.
* Pada Output 2: Saat user menampilkan postingan berdasarkan username, program hanya mencetak data dengan username yang sama, menandakan fungsi showByUsername berjalan benar.
* Pada Output 3: Saat user memilih tampilkan postingan terpopuler, program mencari node dengan nilai like tertinggi dan menampilkan datanya. Ini membuktikan fungsi showMostLiked bekerja sesuai logika perbandingan like.
* Pada Output 4: Saat user melakukan like atau unlike, jumlah like pada ID yang sesuai bertambah atau berkurang. Fungsi updateLike berhasil memperbarui data langsung di node yang ditemukan.

5. Kesimpulan

Berdasarkan implementasi dan pengujian kode di atas, dapat disimpulkan bahwa:

Program My Tel-U Timeline berhasil dijalankan dengan struktur Doubly Linked List.

Semua fungsi seperti tambah posting terurut, tampilkan posting berdasarkan username, tampilkan posting terpopuler, dan update like bekerja dengan benar.
Data tersimpan dan ditampilkan sesuai urutan ID, serta perubahan like langsung terlihat pada output.

Program telah berhasil memenuhi spesifikasi tugas.#