

Tugas Pendahuluan Modul 6
STRUKTUR DATA – Ganjil 2024/2025
"DLL"

A. Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara **Individu**.
2. TP ini bersifat **WAJIB**, tidak mengerjakan = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
3. Hanya **MENGUMPULKAN** tetapi **TIDAK MENERJAKAN** = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
4. Deadline pengumpulan TP Modul 6 adalah **Senin, 21 Oktober 2024 pukul 06.00 WIB**.
5. **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENERJAKAN**.
6. **DILARANG PLAGIAT (PLAGIAT = E)**.
7. Kerjakan TP dengan jelas agar dapat dimengerti.
8. File diupload di LMS menggunakan format **PDF** dengan ketentuan: **TP_MOD_[XX]_NIM_NAMA.pdf**
9. **SOAL TEORI WAJIB DIKERJAKAN TULIS TANGAN, TIDAK BOLEH DIKETIK!**

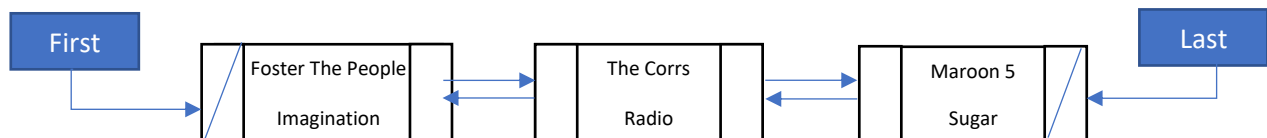
CP (WA):

- Raihan (089638482851)
- Kayyisa (085105303555)
- Abiya (082127180662)
- Rio (081210978384)

SELAMAT MENERJAKAN^^

TUGAS PENDAHULUAN MODUL 6

Diberikan list yang setiap info berisi Nama Band dan Judul Lagunya
Berikut ilustrasinya:



Bagian I: Buatlah ADT Double linked list diatas dengan nama DLL.h
Definisi tipe data untuk list tersebut sebagai berikut:

```

type infotype: integer
type address: pointer to elmtList
type elmtList: <info: infotype, prev: address, next: address>
type list: <first, last: address>

function isEmpty (L: list) →boolean
{Fungsi mengirimkan true jika list kosong, dan false jika tidak kosong}
procedure createList (output L: list)
{I.S. –
F.S. terdefinisi L, list kosong}

function createNewElmt (input X: infotype, output P: address)
{I.S. X adalah info yang akan ditempatkan pada elemen baru yang dialokasi
F.S. terdefinisi elemen list dengan address P, dimana info dari P adalah X, atau
dikembalikan NULL jika alokasi elemen baru gagal}

procedure insertFirst (input/output L: list, input P: address)
{I.S List L mungkin kosong. Telah terdefinisi P untuk disisipkan ke dalam L; next(P) dan
Prev(P)= NULL
F.S. P menjadi elemen pertama dari list L}

procedure insertAfter (input/output L: list, input/output Prec, P: address)
{I.S. Prec tidak NULL dan merupakan elemen list L. P akan disisipkan setelah Prec
F.S. P sudah disisipkan ke dalam L dan terletak setelah Prec}

procedure insertLast (input/output L: list, input P: address)
{I.S List L mungkin kosong. Telah terdefinisi P untuk disisipkan ke dalam L; next(P) dan
prev(P)= NULL
F.S. P menjadi elemen terakhir pada dari list L}

procedure deleteFirst (input/output L: list, output P: address)
{I.S. L mempunyai satu atau lebih elemen
F.S. Elemen pertama sudah dihapus dan dicatat pada P. List L mungkin menjadi kosong}

```

procedure deleteAfter (input/output L: list, Prec: address, output P: address)
{I.S. Prec adalah elemen list L dan next(prec) tidak NULL. Next(prec) mungkin menunjuk ke elemen terakhir
F.S. Elemen setelah Prec sudah dihapus dari L dan diingat pada P}

procedure deleteLast (input/output L: list, output P: address)
{I.S. L mempunyai satu atau lebih elemen
F.S. Elemen terakhir sudah dihapus dan dicatat pada P. List L mungkin menjadi kosong}

procedure concat (input L1, L2: list, output L3: list)
{I.S. L1 dan L2 masing-masing mempunyai satu atau lebih elemen
F.S. L3 berisi semua elemen L1 digabung dengan semua elemen L2}

function findLagu (Judul: String, L: List) → address
{Mencari lagu dengan judul tertentu.
IS: Terdefinisi List L dan lagu yang akan dicari. List mungkin kosong.
FS: Return alamat dimana lagu ditemukan, return nil jika tidak ditemukan.}

procedure removeLagu (In Judul: String, In/Out L: List)
{IS: List mungkin kosong. Jika tidak kosong, minimal memiliki 1 elemen.
FS: Lagu dengan Judul tsb di-dealokasi dari List L.
Notes:
a. Gunakan function findLagu untuk mencari judul lagu.
b. Lagu yang dihapus mungkin merupakan elemen pertama, tengah, atau terakhir.}

BAGIAN 2: Buat implementasi double linked list di atas dalam “DLL.cpp”.

BAGIAN 3: panggil semua fungsi dan procedure di program utama “main.cpp” untuk membuktikan fungsi dan prosedur memberikan hasil yang benar.