

**TUGAS BESAR STRUKTUR DATA**  
**“DATA ORGANISASI - MAHASISWA - PRESTASI**  
**DENGAN MULTI LINKED LIST M ke N”**



**oleh kelompok 2:**

Gede Bagus Krishnanditya Merta      1301223088

Raka Aditya Waluya      1301220192

**PROGRAM STUDI S-1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**UNIVERSITAS TELKOM**

**2023**

## DAFTAR ISI

<b>ABSTRAK.....</b>	<b>3</b>
<b>PENDAHULUAN.....</b>	<b>3</b>
<b>DATA.....</b>	<b>3</b>
<b>METODE.....</b>	<b>5</b>
<b>VISUALISASI.....</b>	<b>5</b>
<b>SIMULASI.....</b>	<b>6</b>
A. Tampilan awal.....	7
B. Tambah data mahasiswa.....	7
C. Tambah data organisasi.....	8
D. Tampil data mahasiswa.....	8
1. Sub-menu tampil data mahasiswa.....	8
2. Tampil list mahasiswa.....	9
3. Tampil organisasi yang diikuti mahasiswa tertentu.....	9
4. Tampil semua data mahasiswa beserta organisasi yang diikuti.....	10
E. Tampil data organisasi.....	10
F. Hapus data mahasiswa.....	11
G. Hapus data organisasi.....	11
H. Cari data mahasiswa.....	12
I. Cari data organisasi.....	12
J. Connect Mahasiswa - Organisasi.....	13
K. Disconnect Mahasiswa - Organisasi.....	13
L. Hitung Elemen.....	14
1. Sub-menu hitung elemen.....	14
2. Hitung Total Mahasiswa.....	14
3. Hitung Total Organisasi.....	15
4. Hitung total organisasi pada mahasiswa tertentu.....	15
<b>KONTRIBUSI.....</b>	<b>16</b>
<b>KESIMPULAN.....</b>	<b>17</b>

## **ABSTRAK**

Laporan ini membahas pengembangan sebuah program yang bertujuan untuk mendaftarkan organisasi dan prestasi mahasiswa dengan menggunakan representasi Multi Linked List M ke N. Sistem ini dirancang untuk memberikan dukungan yang optimal dalam pencatatan serta pemrosesan data terkait keterlibatan mahasiswa dalam organisasi dan prestasi yang dimilikinya.

Implementasi Multi Linked List M ke N pada sistem ini memungkinkan penanganan relasi kompleks antara data mahasiswa, organisasi, dan prestasi, memberikan fleksibilitas dalam analisis dan pengambilan informasi terkait organisasi serta prestasi yang dimiliki oleh mahasiswa.

Melalui laporan ini, akan dibahas bagaimana cara kerja program beserta visualisasi fitur-fitur yang ada pada program yang kami buat. Kami akan menguraikan bagaimana pengguna dapat menampilkan, menambahkan, menghapus, memperbarui, mencari, hingga menghitung data mahasiswa, organisasi dan prestasinya.

## **PENDAHULUAN**

Dalam era pendidikan tinggi yang semakin berkembang, penting bagi institusi untuk memiliki sistem yang mendukung pencatatan dan pengelolaan data keanggotaan mahasiswa dalam organisasi serta pencapaian prestasi mereka.

Ketidakmudahan dalam pencatatan, pemrosesan data, dan analisis hubungan antara keanggotaan mahasiswa dalam organisasi dan pencapaian prestasi menjadi sorotan masalah yang ingin kami atasi. Program yang kami kembangkan bertujuan untuk memberikan solusi mudah dalam hal ini. Menggunakan konsep Multi Linked List M ke N, program ini dirancang untuk memberikan kemudahan dalam mencatat hubungan antara data mahasiswa, organisasi, dan prestasi.

## DATA

Program ini mengimplementasikan struktur data Multi Linked List jenis M ke N yang terdiri dari 2 list utama yaitu Mahasiswa dan Organisasi. List mahasiswa berperan sebagai parent yang merupakan list yang berjenis Circular Single Linked List, sedangkan list organisasi berperan sebagai child yang merupakan list berjenis Double Linked List. Kedua list tersebut kemudian dihubungkan dengan suatu elemen relasi menggunakan pointer.

Berikut merupakan penjabaran dari abstract data type tiap list

### 1. Mahasiswa (Parent) :

```
type infotype_mhs <
  nama, nim : string
  umur, semester : integer
  ipk : float
  prestasi : array of string
>

type adr_mhs pointer to elmlist_mhs

type elmlist_mhs <
  info : infotype_mhs
  firstRelasi : adr_relasi
  next : adr_mhs
>

type List_mhs <
  first : adr_mhs
>
```

### 2. Organisasi (Child) :

```
type infotype_org <
  nama, jenis, : string
  tahun_terbit, jumlah_anggota : integer
>

type adr_org pointer to elmlist_org

type elmlist_org <
  info : infotype_org
  next : adr_org
  prev : adr_org
>

type List_org <
  first : adr_org
  last : adr_org
>
```

### 3. Relasi (Elemen Penghubung)

```
type adr_relasi pointer to elmlist_relasi

type elmlist_relasi <
  nextRelasi : adr_relasi
  prevRelasi : adr_relasi
  nextChild : adr_org
>
```

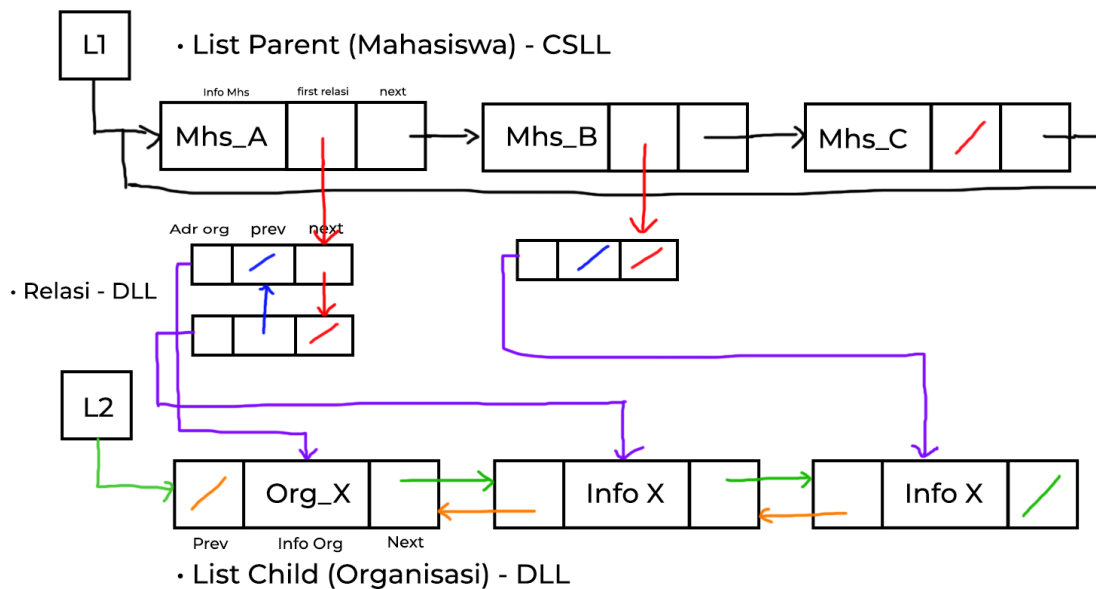
Melalui elemen relasi, setiap elemen di list mahasiswa dapat terhubung dengan sejumlah elemen di list organisasi, dan sebaliknya. Dengan adanya hubungan ini, program dapat dengan mudah mengakses dan memanipulasi data terkait keanggotaan mahasiswa dalam organisasi. Langkah-langkah implementasi program ini akan dijelaskan secara lebih rinci dalam laporan berikut.

## METODE

Pendekatan struktur data yang kami gunakan pada tugas besar kami adalah konsep Multi Linked List jenis M ke N. Dibandingkan dengan 1 ke N, penggunaan metode ini memiliki lebih banyak keuntungan dalam mengatur spesifikasi data yang kami pilih. Keuntungan pertama adalah struktur data ini memungkinkan hubungan yang kompleks antara mahasiswa dan organisasi dan memungkinkan setiap mahasiswa terhubung dengan beberapa organisasi dan sebaliknya.

## VISUALISASI

Berikut adalah visualisasi dari Multi Linked List M ke N sesuai dengan spesifikasi yang kami miliki :



Berdasarkan gambar di atas, terlihat terdapat dua jenis list, yaitu list parent dan list child. Sesuai dengan spesifikasi, list parent disebut CSLL (Circular Singly Linked List) dan mewakili entitas mahasiswa, sedangkan list child disebut DLL (Doubly Linked List) dan mewakili entitas organisasi. Selanjutnya, terdapat elemen relasi (DLL) yang berfungsi sebagai penghubung antara kedua list tersebut.

List parent merupakan kumpulan elemen yang menyimpan informasi mengenai mahasiswa. Setiap elemen dalam list parent memiliki beberapa atribut, termasuk info mahasiswa yang mencakup Nama (String), NIM (String), Umur (Integer), Semester (Integer), dan Prestasi (Array of String). Selain itu, setiap elemen juga memiliki alamat pointer ke first relasi (yang mengarah ke elemen relasi) dan alamat pointer ke elemen mahasiswa berikutnya dalam list.

List child adalah kumpulan elemen yang menyimpan informasi mengenai organisasi. Setiap elemen dalam list child memiliki atribut seperti Nama (String), Jenis (String), Tahun Terbit (Integer), serta Jumlah Anggota (Integer). Setiap elemen memiliki alamat pointer ke elemen organisasi berikutnya (Next) dan alamat pointer ke elemen organisasi sebelumnya (Prev).

Elemen relasi, yang disimpan di dalam elemen mahasiswa, memiliki atribut Next relasi (pointer ke elemen relasi berikutnya), Prev relasi (pointer ke elemen relasi sebelumnya), dan Next child (pointer ke elemen organisasi dalam list child). Pointer Next child berfungsi untuk menghubungkan elemen list mahasiswa dengan elemen list organisasi, membentuk relasi antara keduanya.

# SIMULASI

## A. Tampilan awal

```
+-----+
|          *MENU UTAMA*          |
+-----+
| 1 | Insert Data Mahasiswa      |
+-----+
| 2 | Insert Data Organisasi     |
+-----+
| 3 | Show Data Mahasiswa       |
+-----+
| 4 | Show Data Organisasi       |
+-----+
| 5 | Hapus Data Mahasiswa       |
+-----+
| 6 | Hapus Data Organisasi      |
+-----+
| 7 | Cari Data Mahasiswa        |
+-----+
| 8 | Cari Data Organisasi       |
+-----+
| 9 | Connect Mahasiswa-Organisasi |
+-----+
|10| Disconnect Mahasiswa-Organisasi |
+-----+
|11| Count Element              |
+-----+
| 0 | Exit*                     |
+-----+
PILIHAN ANDA : |
```

Pada tampilan awal ini kami menggunakan while loop, yang mana program akan berhenti hingga user memilih 0

## B. Tambah data mahasiswa

```
PILIHAN ANDA : 1

+-----+
|          *TAMBAH MAHASISWA*          |
+-----+
Nama           : Raka
NIM            : 1301220192
Umur           : 20
Semester       : 3
IPK            : 4
+-----+
Prestasi(y/n)  : y
Banyak prestasi : 2
Prestasi 1     : Lomba_edit_video
Prestasi 2     : Hackfest_2024
+-----+
|-----*SUKSES*-----|
```

Ketika user memilih pilihan 1 pada menu utama, maka program akan meminta user untuk memasukkan data mahasiswa yang kemudian akan ditambahkan ke dalam list mahasiswa.

### C. Tambah data organisasi

```
PILIHAN ANDA : 2

+-----+
|          *TAMBAH ORGANISASI*          |
+-----+
Nama           : BPH
Jenis          : Ormawa
Tahun Terbit   : 2016
Jumlah Anggota : 125
+-----+
|-----*SUKSES*-----|
+-----+
```

Ketika user memilih pilihan 2 pada menu utama, maka program akan meminta user untuk memasukkan data organisasi yang kemudian akan ditambahkan ke dalam list organisasi

### D. Tampil data mahasiswa

#### 1. Sub-menu tampil data mahasiswa

```
PILIHAN ANDA : 3

+-----+
|          *SUB-MENU*          |
+-----+
|          *SHOW DATA MAHASISWA*          |
+-----+
| 1 | Show List Mahasiswa      |
+-----+
| 2 | Show Relasi Mahasiswa 'x'|
+-----+
| 3 | Show All Data Mahasiswa  |
+-----+
| 0 | Kembali*                |
+-----+
PILIHAN ANDA :
```

Pada pilihan nomor 3 di menu utama, akan ditampilkan sub-menu yang meminta user untuk memilih kembali apa data yang ingin ditampilkan. Kami menggunakan while loop yang hanya akan keluar sub-menu jika user memilih 0



## 2. Tampil list mahasiswa

```
PILIHAN ANDA : 1

+-----+
|      *DATA MAHASISWA 1*      |
+-----+
| Nama      : Raka              |
| NIM       : 1301220192        |
| Umur      : 20                |
| Semester  : 3                |
| IPK       : 4                |
+-----+
| Prestasi 1 : Lomba_edit_video |
| Prestasi 2 : Hackfest_2024    |
+-----+
|      *DATA MAHASISWA 2*      |
+-----+
| Nama      : Bagus            |
| NIM       : 13012203038       |
| Umur      : 20                |
| Semester  : 3                |
| IPK       : 4                |
+-----+
| Prestasi 1 : UI_UX_Design     |
| Prestasi 2 : Hackfest_2024    |
+-----+
~Press any key
```

Ketika user memilih 1 pada sub-menu tampil data mahasiswa, maka program akan menampilkan seluruh data yang ada pada list mahasiswa.

## 3. Tampil organisasi yang diikuti mahasiswa tertentu

```
PILIHAN ANDA : 2

+-----+
| *MASUKKAN NAMA MAHASISWA*    |
+-----+
JAWABAN ANDA: Raka

+-----+
| Organisasi 1 : BPH            |
| Organisasi 2 : IEEE_Telkom_Univ |
+-----+
~Press any key
```

Ketika user memilih 2 pada sub-menu tampil data mahasiswa, maka program akan meminta masukkan user terkait nama mahasiswa yang ingin ditampilkan organisasinya.

#### 4. Tampil semua data mahasiswa beserta organisasi yang diikuti

```
PILIHAN ANDA : 3

+-----+
| *DATA MAHASISWA 1* |
+-----+
| Nama      : Raka    |
| NIM       : 1301220192 |
| Umur      : 20      |
| Semester  : 3       |
| IPK       : 4       |
+-----+
| Prestasi 1 : Lomba_edit_video |
| Prestasi 2 : Hackfest_2024   |
+-----+
| Organisasi 1 : BPH          |
| Organisasi 2 : IEEE_Telkom_Univ |
+-----+
| *DATA MAHASISWA 2* |
+-----+
| Nama      : Bagus    |
| NIM       : 13012203038 |
| Umur      : 20      |
| Semester  : 3       |
| IPK       : 4       |
+-----+
| Prestasi 1 : UI_UX_Design |
| Prestasi 2 : Hackfest_2024 |
+-----+
| Organisasi 1 : BPM        |
+-----+
~Press any key
```

Ketika user memilih 3 pada sub-menu tampil data mahasiswa, maka program akan menampilkan seluruh data mahasiswa beserta organisasinya.

#### E. Tampil data organisasi

```
PILIHAN ANDA : 4

+-----+
| *DATA ORGANISASI 1* |
+-----+
| Nama      : BPM      |
| Jenis     : Ormawa    |
| Tahun Terbit : 2017   |
| Anggota   : 130      |
+-----+
| *DATA ORGANISASI 2* |
+-----+
| Nama      : BPH      |
| Jenis     : Ormawa    |
| Tahun Terbit : 2016   |
| Anggota   : 125      |
+-----+
~Press any key
```

Ketika user memilih 4 pada pilihan menu utama, maka program akan menampilkan seluruh data organisasi yang ada di dalam list

## F. Hapus data mahasiswa

```
PILIHAN ANDA : 5

+-----+
| *NAMA MAHASISWA YANG AKAN DIHAPUS* |
+-----+
JAWABAN ANDA : tes
+-----+
|      *DATA BERHASIL DIHAPUS*      |
+-----+

~Press any key
|
```

Ketika user memilih 5 pada pilihan menu utama, maka program akan meminta masukkan nama mahasiswa yang ingin dihapus oleh user, yang kemudian akan dilakukan pencarian berdasarkan nama mahasiswa dan jika data tersebut ditemukan, maka data akan dihapus, sebaliknya jika tidak ditemukan maka akan muncul notifikasi “Data tidak ditemukan”. Hal ini juga secara otomatis menghapus semua organisasi yang diikuti mahasiswa tersebut.

## G. Hapus data organisasi

```
PILIHAN ANDA : 6

+-----+
| *NAMA ORGANISASI YANG AKAN DIHAPUS* |
+-----+
JAWABAN ANDA : tes_hapus
+-----+
|      *DATA BERHASIL DIHAPUS*      |
+-----+

~Press any key
```

Ketika user memilih 6 pada pilihan menu utama, maka program akan meminta masukkan nama organisasi yang ingin dihapus oleh user, yang kemudian akan dilakukan pencarian berdasarkan nama mahasiswa dan jika data tersebut ditemukan, maka data akan dihapus, sebaliknya jika tidak ditemukan maka akan muncul notifikasi “Data tidak ditemukan”. Ketika hal ini dilakukan, secara otomatis program juga akan menghapus relasi dari mahasiswa yang memiliki data organisasi yang baru saja dihapus.

## H. Cari data mahasiswa

```
PILIHAN ANDA : 7

+-----+
| *NIM MAHASISWA YANG DICARI* |
+-----+
JAWABAN ANDA: 1301220192

+-----+
| *DATA DITEMUKAN* |
+-----+

+-----+
| *DATA MAHASISWA* |
+-----+
| Nama      : Raka |
| NIM       : 1301220192 |
| Umur      : 20 |
| Semester  : 3 |
| IPK       : 4 |
+-----+
| Prestasi 1 : Lomba_edit_video |
| Prestasi 2 : Hackfest_2024 |
+-----+
~Press any key
```

Jika user memilih 7 pada pilihan menu utama, maka program akan meminta user untuk memasukkan NIM mahasiswa yang dicari, kemudian akan dilakukan proses pencarian mahasiswa berdasarkan NIM nya. Jika data ditemukan maka program akan menampilkan data mahasiswa, sebaliknya jika data tidak ditemukan maka program akan menampilkan notifikasi “Data tidak ditemukan”

## I. Cari data organisasi

```
PILIHAN ANDA : 8

+-----+
| *NAMA ORGANISASI YANG DICARI* |
+-----+
JAWABAN ANDA: BPM

+-----+
| *DATA DITEMUKAN* |
+-----+

+-----+
| *DATA ORGANISASI* |
+-----+
| Nama      : BPM |
| Jenis     : Ormawa |
| Tahun Terbit : 2017 |
| Anggota   : 130 |
+-----+
~Press any key
```

Jika user memilih 8 pada pilihan menu utama, maka program akan meminta user untuk memasukkan nama organisasi yang dicari, kemudian akan dilakukan proses pencarian organisasi berdasarkan namanya. Jika data ditemukan maka program akan menampilkan data organisasi, sebaliknya jika data tidak ditemukan maka program akan menampilkan notifikasi “Data tidak ditemukan”

#### J. Connect Mahasiswa - Organisasi

```
PILIHAN ANDA : 9

Masukkan NAMA Mahasiswa : Raka
Masukkan NAMA Organisasi : BPH
+-----+
|  *DATA BERHASIL TERKONEKSI*  |
+-----+
~Press any key
```

Jika user memilih 9 pada pilihan menu utama, maka program akan meminta 2 masukkan user, yaitu nama mahasiswa dan organisasi yang ingin dihubungkan. Setelah itu akan dibuat sebuah address relasi yang berfungsi untuk menghubungkan kedua elemen mahasiswa dan organisasi. Program juga otomatis akan menambahkan data jumlah\_anggota yang dimiliki oleh suatu organisasi ketika terdapat koneksi baru yang dibuat antara mahasiswa dan organisasi yang bersangkutan

#### K. Disconnect Mahasiswa - Organisasi

```
PILIHAN ANDA : 10

Masukkan NAMA Mahasiswa : Raka
Masukkan NAMA Organisasi : BPH
+-----+
|  *DATA BERHASIL TERDISCONNECT*  |
+-----+
~Press any key
```

Jika user memilih 10 pada pilihan menu utama, maka program akan meminta 2 masukkan user, yaitu nama mahasiswa dan organisasi yang ingin dihapus. Setelah itu akan address relasi yang menghubungkan kedua elemen mahasiswa dan organisasi akan dihapus. Program juga otomatis akan mengurangi data jumlah\_anggota yang dimiliki oleh suatu organisasi ketika terdapat relasi antara mahasiswa dan organisasi yang terkait dihapus

## L. Hitung Elemen

### 1. Sub-menu hitung elemen

```
PILIHAN ANDA : 11

+-----+
|               *SUB-MENU*               |
+-----+
|               *COUNT ELEMENT*          |
+-----+
| 1 | Total Mahasiswa |
+-----+
| 2 | Total Organisasi |
+-----+
| 3 | Jumlah Organisasi pada Mhs 'x' |
+-----+
| 0 | Kembali*        |
+-----+
PILIHAN ANDA :
```

Ketika user memilih 11, maka program akan menampilkan sub-menu hitung elemen yang kemudian user akan diminta untuk memilih kembali, kami menggunakan while loop yang mana user hanya akan keluar sub-menu jika memilih 0.

### 2. Hitung Total Mahasiswa

```
+-----+
|               *SUB-MENU*               |
+-----+
|               *COUNT ELEMENT*          |
+-----+
| 1 | Total Mahasiswa |
+-----+
| 2 | Total Organisasi |
+-----+
| 3 | Jumlah Organisasi pada Mhs 'x' |
+-----+
| 0 | Kembali*        |
+-----+
PILIHAN ANDA : 1
Total Mahasiswa: 2
```

Ketika user memilih 1 pada sub-menu hitung elemen, maka program akan menampilkan total mahasiswa yang ada di dalam list mahasiswa

### 3. Hitung Total Organisasi

```
+-----+
|               *SUB-MENU*               |
+-----+
|               *COUNT ELEMENT*          |
+-----+
| 1 | Total Mahasiswa |
+---+-----+
| 2 | Total Organisasi |
+---+-----+
| 3 | Jumlah Organisasi pada Mhs 'x' |
+---+-----+
| 0 | Kembali*        |
+-----+
PILIHAN ANDA : 2

Total Organisasi: 3
```

Ketika user memilih 2 pada sub-menu hitung elemen, maka program akan menampilkan total organisasi yang ada di dalam list organisasi

### 4. Hitung total organisasi pada mahasiswa tertentu

```
+-----+
|               *SUB-MENU*               |
+-----+
|               *COUNT ELEMENT*          |
+-----+
| 1 | Total Mahasiswa |
+---+-----+
| 2 | Total Organisasi |
+---+-----+
| 3 | Jumlah Organisasi pada Mhs 'x' |
+---+-----+
| 0 | Kembali*        |
+-----+
PILIHAN ANDA : 3

Masukkan nama mahasiswa: Bagus
Total Organisasi yang diikuti Bagus: 1
```

Ketika user memilih 3 pada sub-menu hitung elemen, maka program akan menampilkan total organisasi yang diikuti oleh mahasiswa tertentu sesuai dengan masukkan user.

## KONTRIBUSI

Dalam proses pembuatan program untuk menyelesaikan masalah pencatatan dan pengelolaan data keanggotaan mahasiswa dalam organisasi serta pencapaian prestasi, kami mengimplementasikan konsep-konsep yang telah kami pelajari melalui mata kuliah Struktur Data. Melalui materi yang diajarkan di kelas, tugas-tugas, praktikum serta ujian, kami memperoleh pemahaman yang baik untuk akhirnya dapat kami terapkan pada program ini.

Berikut adalah kontribusi masing-masing dari kami dalam pengerjaan tugas besar :

No	Jobdesk	Jenis	Bagus	Raka
1	Membuat Header dan ADT file Parent.h	Header File Program	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	Membuat Header dan ADT file Child.h		<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	Membuat Header dan ADT file Relasi.h		<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Membuat Source (function dan procedure) file Parent.cpp	Source File Program	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Membuat Source (function dan procedure) file Child.cpp		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Membuat Source (function dan procedure) file Relasicpp		<input type="checkbox"/>	<input checked="" type="checkbox"/>
7	Membuat menu utama pada main.cpp	File Utama (main.cpp)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	Membuat sub menu untuk setiap pilihan pada menu utama		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Mengimplementasikan function dan procedure yang telah dibuat pada file source ke dalam masing-masing sub-menu		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Melakukan debugging v1	debugging program	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	Melakukan debugging v2		<input type="checkbox"/>	<input checked="" type="checkbox"/>
12	Finishing Program		<input type="checkbox"/>	<input checked="" type="checkbox"/>
13	Penyusunan PPT	Laporan Akhir	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	Penyusunan Laporan		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	Mengecek, memperbaiki dan menggabungkan semua hal yang telah dikerjakan.	Finishing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>



## KESIMPULAN

Program yang kami rancang untuk pencatatan dan pengelolaan data keanggotaan mahasiswa dalam organisasi menggunakan Multi Linked List M ke N berjalan lancar dan memenuhi ketentuan yang telah ditetapkan. Implementasi Circular Linked List untuk list mahasiswa dan Double Linked List untuk list organisasi, yang dihubungkan melalui elemen relasi, memberikan solusi efektif dan efisien dalam mengatasi kompleksitas hubungan antar elemen.

Program ini memberikan kemudahan dalam pengelolaan informasi, memungkinkan pengguna untuk dengan mudah menambah, menghapus, dan memperbarui data keanggotaan mahasiswa dalam organisasi serta mencatat prestasi yang dimilikinya. Antarmuka yang mudah dipahami membuat program ini dapat dioperasikan dengan baik oleh semua user

Dengan demikian, dapat disimpulkan bahwa program ini berhasil mencapai tujuan yang diinginkan. kami telah menerapkan ilmu yang kami peroleh selama satu semester mempelajari struktur data untuk kemudian diimplementasikan ke program pencatatan dan pengelolaan data mahasiswa, organisasi dan prestasi.