

HANDS-ON LABS 1

STRUKTUR DATA



Oleh:

Martogi Jekson C. Siagian 2105541054

PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS UDAYANA

2023

Hands-On Labs 1

Tugas

1. Buatlah program menggunakan konsep ADT dan Array untuk menginputkan data hasil ujian mahasiswa, dengan data mahasiswa terdiri dari: NIM, Nama, Nilai. Ketentuan program:

- Program menampilkan menu pilihan: input data nilai mahasiswa, lihat seluruh data mahasiswa dan rekap data mahasiswa.
- Pada pilihan menu: input data nilai mahasiswa maka pengguna dapat menginputkan berapa jumlah mahasiswa yang akan diinputkan nilainya dan menginputkan detail nilai mahasiswa
- Pada pilihan menu: lihat seluruh data mahasiswa maka pengguna dapat menampilkan seluruh data mahasiswa beserta nilainya yang telah diinputkan sebelumnya
- Pada pilihan menu: rekap data mahasiswa maka pengguna dapat menampilkan data mahasiswa dengan nilai terbesar dan terkecil serta nilai rata-rata ujian mahasiswa yang telah diinputkan.

Jawab :

Berikut Merupakan Source Code yang saya buat untuk membuat Program Tersebut:

```
tugas1.cpp X
1  #include <iostream>
2  #include <vector>
3  #include <string>
4  #include <algorithm>
5
6  using namespace std;
7
8  // Struktur data untuk menyimpan informasi mahasiswa
9  struct Mahasiswa {
10     string nim;
11     string nama;
12     float nilai;
13 };
14
15 // Fungsi untuk menginputkan data mahasiswa
16 vector<Mahasiswa> input_data() {
17     vector<Mahasiswa> data_mahasiswa;
18     int jumlah_mahasiswa;
19
20     cout << "\nMasukkan jumlah mahasiswa: ";
21     cin >> jumlah_mahasiswa;
22     cout << "-----";
23
24     for (int i = 0; i < jumlah_mahasiswa; i++) {
25         Mahasiswa mahasiswa;
26         cout << "\nMasukkan NIM mahasiswa ke-" << i + 1 << ": ";
27         cin >> mahasiswa.nim;
28         cout << "Masukkan Nama mahasiswa ke-" << i + 1 << ": ";
29         cin.ignore(); // Untuk membersihkan newline karakter
30         getline(cin, mahasiswa.nama);
31         cout << "Masukkan Nilai mahasiswa ke-" << i + 1 << ": ";
32         cin >> mahasiswa.nilai;
33         data_mahasiswa.push_back(mahasiswa);
34         cout << "Data mahasiswa berhasil dimasukkan.\n";
35         cout << "-----";
36     }
37
38     return data_mahasiswa;
39 }
40
```

```

tugas1.cpp X
41 // Fungsi untuk melihat seluruh data mahasiswa
42 void lihat_data(const vector<Mahasiswa>& data_mahasiswa) {
43     cout << "\nData Mahasiswa:\n";
44     if (data_mahasiswa.empty()) {
45         cout << "Data mahasiswa kosong. Silakan masukkan data terlebih dahulu.\n";
46     } else {
47         for (size_t i = 0; i < data_mahasiswa.size(); i++) {
48             const Mahasiswa& mahasiswa = data_mahasiswa[i];
49             cout << "-----\n";
50             cout << "Mahasiswa ke-" << i + 1 << ":\n";
51             cout << "NIM : " << mahasiswa.nim << "\n";
52             cout << "Nama : " << mahasiswa.nama << "\n";
53             cout << "Nilai : " << mahasiswa.nilai << endl;
54         }
55     }
56 }
57
58 // Fungsi untuk merekap data mahasiswa
59 void rekap_data(const vector<Mahasiswa>& data_mahasiswa) {
60     if (data_mahasiswa.empty()) {
61         cout << "Data mahasiswa kosong. Silakan masukkan data terlebih dahulu.\n";
62         return;
63     }
64
65     auto max_it = max_element(data_mahasiswa.begin(), data_mahasiswa.end(), [](const Mahasiswa& a, const Mahasiswa& b) {
66         return a.nilai < b.nilai;
67     });
68
69     auto min_it = min_element(data_mahasiswa.begin(), data_mahasiswa.end(), [](const Mahasiswa& a, const Mahasiswa& b) {
70         return a.nilai < b.nilai;
71     });
72
73     float total_nilai = 0;
74     for (const Mahasiswa& mahasiswa : data_mahasiswa) {
75         total_nilai += mahasiswa.nilai;
76     }
77     float rata_rata = total_nilai / data_mahasiswa.size();
78

```

```

tugas1.cpp X
79     cout << "\nRekap Data Mahasiswa:\n";
80     cout << "-----\n";
81     cout << "Nilai Terbesar:\n";
82     cout << "NIM : " << max_it->nim << "\n";
83     cout << "Nama : " << max_it->nama << "\n";
84     cout << "Nilai : " << max_it->nilai << "\n";
85
86     cout << "-----\n";
87     cout << "Nilai Terkecil:\n";
88     cout << "NIM : " << min_it->nim << "\n";
89     cout << "Nama : " << min_it->nama << "\n";
90     cout << "Nilai : " << min_it->nilai << "\n";
91
92     cout << "-----\n";
93     cout << "Nilai Rata-rata: " << rata_rata << "\n";
94 }
95
96 void cetak_garis_pemisah() {
97     cout << "-----\n";
98 }
99
100 int main() {
101     vector<Mahasiswa> data_mahasiswa;
102
103     cout << "Selamat datang di Program Manajemen Data Mahasiswa!\n";
104     cout << "By : Martogi Jekson C.Siagian\n";
105
106     while (true) {
107         cout << "\nMenu Pilihan:\n";
108         cout << "1. Input Data Nilai Mahasiswa\n";
109         cout << "2. Lihat Seluruh Data Mahasiswa\n";
110         cout << "3. Rekap Data Mahasiswa\n";
111         cout << "4. Keluar\n";
112
113         int pilihan;
114         cout << "Pilih menu (1/2/3/4): ";
115         cin >> pilihan;
116
117         switch (pilihan) {
118             case 1:
119                 data_mahasiswa = input_data();
120                 break;

```

```

121             case 2:
122                 lihat_data(data_mahasiswa);
123                 break;
124             case 3:
125                 rekap_data(data_mahasiswa);
126                 break;
127             case 4:
128                 cout << "Terima kasih telah menggunakan program ini. Sampai jumpa!\n";
129                 return 0; // Keluar dari program
130             default:
131                 cout << "Pilihan tidak valid. Silakan pilih menu yang benar.\n";
132         }
133     }
134
135     return 0;
136 }
137

```

1. Include Libraries

Pertama-tama, kode ini mengimpor beberapa pustaka (libraries) standar C++ yang diperlukan untuk program ini:

```
1  #include <iostream>
2  #include <vector>
3  #include <string>
4  #include <algorithm>
```

- **iostream:** Library ini digunakan untuk input dan output standar, sehingga kita dapat berinteraksi dengan pengguna melalui terminal.
- **vector:** Digunakan untuk menyimpan data mahasiswa dalam bentuk vektor (koleksi dinamis).
- **string:** Untuk bekerja dengan data teks, seperti nama mahasiswa.
- **algorithm:** Digunakan untuk menggunakan fungsi `max_element` dan `min_element` yang membantu mencari nilai maksimum dan minimum dalam vektor.

2. using namespace std;

Baris ini adalah perintah yang memberi tahu kompilator bahwa Anda akan menggunakan semua anggota (class, fungsi, dan objek) yang didefinisikan dalam namespace `std` (yang merupakan namespace standar C++ Standard Library) dalam program Anda tanpa harus menyebutkan `std::` sebelumnya.

3. Struktur Data Mahasiswa

```
8  // Struktur data untuk menyimpan informasi mahasiswa
9  struct Mahasiswa {
10     string nim;
11     string nama;
12     float nilai;
13 };
```

ini adalah struktur data yang digunakan untuk menyimpan informasi tentang mahasiswa. Setiap Mahasiswa memiliki tiga atribut: NIM, nama, dan nilai.

- **nim (tipe string):** Ini adalah anggota yang bertipe string dan digunakan untuk menyimpan Nomor Induk Mahasiswa (NIM) dari

seorang mahasiswa. NIM biasanya berupa teks atau karakter yang mengidentifikasi unik setiap mahasiswa dalam sebuah program atau institusi pendidikan.

- **nama (tipe string):** Ini adalah anggota yang juga bertipe string dan digunakan untuk menyimpan nama lengkap seorang mahasiswa. Nama biasanya adalah teks atau karakter yang mewakili nama mahasiswa.
- **nilai (tipe float):** Ini adalah anggota yang bertipe float dan digunakan untuk menyimpan nilai numerik yang berkaitan dengan prestasi akademik seorang mahasiswa. Dalam konteks ini, nilai bisa berupa nilai dalam bentuk angka desimal, seperti nilai rata-rata akademik atau nilai dalam suatu ujian.

4. Fungsi input_data()

```
15 // Fungsi untuk menginputkan data mahasiswa
16 vector<Mahasiswa> input_data() {
17     vector<Mahasiswa> data_mahasiswa;
18     int jumlah_mahasiswa;
19
20     cout << "\nMasukkan jumlah mahasiswa: ";
21     cin >> jumlah_mahasiswa;
22     cout << "-----";
23
24     for (int i = 0; i < jumlah_mahasiswa; i++) {
25         Mahasiswa mahasiswa;
26         cout << "\nMasukkan NIM mahasiswa ke-" << i + 1 << ": ";
27         cin >> mahasiswa.nim;
28         cout << "Masukkan Nama mahasiswa ke-" << i + 1 << ": ";
29         cin.ignore(); // Untuk membersihkan newline karakter
30         getline(cin, mahasiswa.nama);
31         cout << "Masukkan Nilai mahasiswa ke-" << i + 1 << ": ";
32         cin >> mahasiswa.nilai;
33         data_mahasiswa.push_back(mahasiswa);
34         cout << "Data mahasiswa berhasil dimasukkan.\n";
35         cout << "-----";
36     }
37
38     return data_mahasiswa;
39 }
```

Fungsi ini digunakan untuk menginputkan data mahasiswa. Ini bekerja dengan cara sebagai berikut:

- **vector<Mahasiswa> input_data() :** Ini adalah deklarasi fungsi input_data yang mengembalikan vektor (vector) dari objek-objek tipe Mahasiswa. Mahasiswa kemungkinan adalah sebuah struktur atau kelas yang telah didefinisikan sebelumnya untuk

menyimpan informasi tentang mahasiswa, seperti NIM, nama, dan nilai.

- **vector<Mahasiswa> data_mahasiswa;** : Ini adalah deklarasi dan inisialisasi vektor **data_mahasiswa** yang akan digunakan untuk menyimpan data mahasiswa yang diinputkan.
- **int jumlah_mahasiswa;** : Ini adalah deklarasi variabel **jumlah_mahasiswa** yang akan digunakan untuk menyimpan jumlah mahasiswa yang akan diinputkan.
- Selanjutnya, program akan meminta pengguna untuk memasukkan jumlah mahasiswa dengan perintah **cout** dan akan membaca jumlah mahasiswa yang dimasukkan oleh pengguna menggunakan **cin**.
- Kemudian, program akan memulai loop **for** untuk menginputkan data mahasiswa sebanyak **jumlah_mahasiswa** kali.
- Pada setiap iterasi loop, program akan membuat objek **Mahasiswa** baru yang akan digunakan untuk menyimpan data mahasiswa yang akan diinputkan.
- Program akan meminta pengguna untuk memasukkan NIM, nama, dan nilai mahasiswa menggunakan **cin** dan **getline** untuk menghindari masalah dengan newline karakter setelah memasukkan jumlah mahasiswa.
- Data mahasiswa yang telah diinputkan kemudian dimasukkan ke dalam vektor **data_mahasiswa** menggunakan **push_back**.
- Setelah data mahasiswa berhasil dimasukkan, pesan "Data mahasiswa berhasil dimasukkan." akan ditampilkan, dan program akan mempersiapkan diri untuk menginputkan data mahasiswa berikutnya.
- Loop akan terus berlanjut hingga semua data mahasiswa telah diinputkan sesuai dengan **jumlah_mahasiswa** yang dimasukkan oleh pengguna.

- Setelah loop selesai, vektor **data_mahasiswa** yang berisi data mahasiswa akan dikembalikan sebagai hasil dari fungsi **input_data**.

5. Fungsi lihat_data()

```

41 // Fungsi untuk melihat seluruh data mahasiswa
42 void lihat_data(const vector<Mahasiswa>& data_mahasiswa) {
43     cout << "\nData Mahasiswa:\n";
44     if (data_mahasiswa.empty()) {
45         cout << "Data mahasiswa kosong. Silakan masukkan data terlebih dahulu.\n";
46     } else {
47         for (size_t i = 0; i < data_mahasiswa.size(); i++) {
48             const Mahasiswa& mahasiswa = data_mahasiswa[i];
49             cout << "-----\n";
50             cout << "Mahasiswa ke-" << i + 1 << ":\n";
51             cout << "NIM    Maharishi" << mahasiswa.nim << "\n";
52             cout << "Nama  : " << mahasiswa.nama << "\n";
53             cout << "Nilai : " << mahasiswa.nilai << endl;
54         }
55     }
56 }

```

Fungsi ini digunakan untuk menampilkan seluruh data mahasiswa yang telah dimasukkan. Berikut merupakan bagaimana kode ini bekerja:

- **void lihat_data(const vector<Mahasiswa>& data_mahasiswa)**: Ini adalah deklarasi fungsi **lihat_data** yang tidak mengembalikan nilai (void) dan menerima satu argumen, yaitu vektor **data_mahasiswa** yang berisi data mahasiswa. Argumen ini diberi kata kunci **const**, yang berarti bahwa vektor tersebut tidak akan diubah dalam fungsi ini.
- **cout << "\nData Mahasiswa:\n";**: Ini adalah pernyataan untuk mencetak judul "Data Mahasiswa" di layar, memberi tahu pengguna bahwa data mahasiswa akan ditampilkan.
- **if (data_mahasiswa.empty()) { ... }**: Ini adalah struktur kondisional **if** yang digunakan untuk memeriksa apakah vektor **data_mahasiswa** kosong atau tidak. Jika vektor tersebut kosong (tidak berisi data mahasiswa), maka program akan mencetak pesan "Data mahasiswa kosong. Silakan masukkan data terlebih dahulu."
- Jika vektor **data_mahasiswa** tidak kosong, program akan menjalankan blok **else**. Dalam blok **else**, program akan menggunakan loop **for** untuk menampilkan setiap data mahasiswa yang ada dalam vektor.

- `for (size_t i = 0; i < data_mahasiswa.size(); i++) { ... }`: Ini adalah loop `for` yang akan berjalan sebanyak jumlah elemen dalam vektor `data_mahasiswa`. Variabel `i` digunakan sebagai indeks untuk mengakses setiap elemen vektor.
- `const Mahasiswa& mahasiswa = data_mahasiswa[i];`: Ini adalah pernyataan yang digunakan untuk mengakses elemen vektor `data_mahasiswa` pada indeks `i` dan mengikatnya ke referensi `mahasiswa` yang diberi kata kunci `const`. Ini memungkinkan kita untuk mengakses data mahasiswa tanpa mengubahnya.
- Kemudian, program akan mencetak informasi tentang setiap mahasiswa, termasuk NIM, nama, dan nilai menggunakan perintah `cout`.
- Setelah satu mahasiswa selesai ditampilkan, program akan mencetak garis pemisah `"-----"` untuk membedakan antara setiap mahasiswa.
- Loop akan berlanjut hingga semua data mahasiswa telah ditampilkan.

Dengan demikian, fungsi `lihat_data` akan menampilkan seluruh data mahasiswa yang telah dimasukkan ke layar, atau memberikan pesan bahwa data mahasiswa kosong jika vektor `data_mahasiswa` kosong.

6. Fungsi `rekap_data()`

```

58 // Fungsi untuk merekap data mahasiswa
59 void rekap_data(const vector<Mahasiswa>& data_mahasiswa) {
60     if (data_mahasiswa.empty()) {
61         cout << "Data mahasiswa kosong. Silakan masukkan data terlebih dahulu.\n";
62         return;
63     }
64     auto max_it = max_element(data_mahasiswa.begin(), data_mahasiswa.end(), [](const Mahasiswa& a, const Mahasiswa& b) {
65         return a.nilai < b.nilai;
66     });
67     auto min_it = min_element(data_mahasiswa.begin(), data_mahasiswa.end(), [](const Mahasiswa& a, const Mahasiswa& b) {
68         return a.nilai < b.nilai;
69     });
70     float total_nilai = 0;
71     for (const Mahasiswa& mahasiswa : data_mahasiswa) {
72         total_nilai += mahasiswa.nilai;
73     }
74     float rata_rata = total_nilai / data_mahasiswa.size();
75
76     cout << "\nRekap Data Mahasiswa:\n";
77     cout << "-----\n";
78     cout << "Nilai Terbesar:\n";
79     cout << "NIM : " << max_it->nim << "\n";
80     cout << "Nama : " << max_it->nama << "\n";
81     cout << "Nilai : " << max_it->nilai << "\n";
82
83     cout << "-----\n";
84     cout << "Nilai Terkecil:\n";
85     cout << "NIM : " << min_it->nim << "\n";
86     cout << "Nama : " << min_it->nama << "\n";
87     cout << "Nilai : " << min_it->nilai << "\n";
88
89     cout << "-----\n";
90     cout << "Nilai Rata-rata: " << rata_rata << "\n";
91 }

```


Fungsi ini digunakan untuk merekap data mahasiswa. Ini mencari nilai maksimum dan minimum dalam vektor `data_mahasiswa` menggunakan fungsi `max_element` dan `min_element`. Selain itu, ia menghitung nilai rata-rata dari semua mahasiswa. Hasil rekap ini ditampilkan di layar. Berikut penjelasan kode ini bekerja:

- **`void rekap_data(const vector<Mahasiswa>& data_mahasiswa)`** : Ini adalah deklarasi fungsi **`rekap_data`** yang tidak mengembalikan nilai (**`void`**) dan menerima satu argumen, yaitu vektor **`data_mahasiswa`** yang berisi data mahasiswa. Argumen ini diberi kata kunci **`const`**, yang berarti bahwa vektor tersebut tidak akan diubah dalam fungsi ini.
- **`if (data_mahasiswa.empty()) { ... }`** : Ini adalah struktur kondisional **`if`** yang digunakan untuk memeriksa apakah vektor **`data_mahasiswa`** kosong atau tidak. Jika vektor tersebut kosong (tidak berisi data mahasiswa), maka program akan mencetak pesan "Data mahasiswa kosong. Silakan masukkan data terlebih dahulu." dan keluar dari fungsi dengan pernyataan **`return`**.
- Selanjutnya, program menggunakan fungsi **`max_element`** dan **`min_element`** dari C++ STL (Standard Template Library) untuk menemukan nilai terbesar dan nilai terkecil dalam vektor **`data_mahasiswa`** dengan membandingkan nilai-nilai mahasiswa menggunakan lambda function yang ditentukan di dalam fungsi ini.
- Program juga menghitung nilai rata-rata dari nilai-nilai mahasiswa dengan menjumlahkan semua nilai dan membaginya dengan jumlah mahasiswa.
- Setelah menghitung semua statistik yang diperlukan, program mencetak rekap data mahasiswa ke layar, termasuk:
Nilai terbesar (NIM, nama, nilai)
Nilai terkecil (NIM, nama, nilai)
Nilai rata-rata

- Setiap informasi mahasiswa (nilai terbesar dan terkecil) dicetak dengan garis pemisah "-----" untuk membedakan antara setiap informasi.

Dengan demikian, fungsi `rekap_data` digunakan untuk menghasilkan rekapitulasi statistik dari data mahasiswa yang telah dimasukkan sebelumnya dalam program Anda, seperti nilai terbesar, nilai terkecil, dan nilai rata-rata. Jika vektor `data_mahasiswa` kosong, maka fungsi ini akan memberikan pesan bahwa data mahasiswa kosong.

7. Fungsi `cetak_garis_pemisah()`

```

96 void cetak_garis_pemisah() {
97     cout << "-----\n";
98 }
```

Fungsi ini digunakan untuk mencetak garis pemisah horizontal. Ini digunakan untuk memisahkan output yang berbeda dalam tampilan program.

8. Fungsi `main()`

```

100 int main() {
101     vector<Mahasiswa> data_mahasiswa;
102
103     cout << "Selamat datang di Program Manajemen Data Mahasiswa!\n";
104     cout << "By : Martogi Jekson C.Siagian\n";
105
106     while (true) {
107         cout << "\nMenu Pilihan:\n";
108         cout << "1. Input Data Nilai Mahasiswa\n";
109         cout << "2. Lihat Seluruh Data Mahasiswa\n";
110         cout << "3. Rekap Data Mahasiswa\n";
111         cout << "4. Keluar\n";
112
113         int pilihan;
114         cout << "Pilih menu (1/2/3/4): ";
115         cin >> pilihan;
116
117         switch (pilihan) {
118             case 1:
119                 data_mahasiswa = input_data();
120                 break;
121             case 2:
122                 lihat_data(data_mahasiswa);
123                 break;
124             case 3:
125                 rekap_data(data_mahasiswa);
126                 break;
127             case 4:
128                 cout << "Terima kasih telah menggunakan program ini. Sampai jumpa!\n";
129                 return 0; // Keluar dari program
130             default:
131                 cout << "Pilihan tidak valid. Silakan pilih menu yang benar.\n";
132         }
133     }
134
135     return 0;
136 }
137
```

Program ini memberikan beberapa opsi kepada pengguna melalui menu, seperti input data nilai mahasiswa, melihat seluruh data mahasiswa, rekap

data mahasiswa, dan keluar dari program. Di bawah ini adalah penjelasan lebih rinci tentang kode tersebut:

- **vector<Mahasiswa> data_mahasiswa;;** Ini adalah deklarasi vector **data_mahasiswa** yang akan digunakan untuk menyimpan data **mahasiswa**. Mahasiswa adalah jenis data yang mungkin sudah didefinisikan di kode sebelumnya atau merupakan bagian dari kode yang tidak ditampilkan.
- Pesan selamat datang: Program mencetak pesan selamat datang dan nama penulis program ke layar.
- **while (true) { ... }** Ini adalah loop tak terbatas yang akan menjalankan program secara terus-menerus sampai pengguna memilih untuk keluar.
- Menu pilihan: Program mencetak menu pilihan ke layar, termasuk opsi untuk input data, melihat data, rekap data, dan keluar.
- **int pilihan;;** Mendeklarasikan variabel pilihan untuk menyimpan pilihan menu yang dimasukkan oleh pengguna.
- **cout << "Pilih menu (1/2/3/4) : " ;** Program meminta pengguna untuk memasukkan pilihan menu.
- **cin >> pilihan;;** Program membaca pilihan menu yang dimasukkan oleh pengguna.
- **switch (pilihan) { ... }** Ini adalah sebuah switch statement yang digunakan untuk mengeksekusi tindakan yang sesuai dengan pilihan menu yang dimasukkan oleh pengguna.
 - case 1:** Jika pengguna memilih menu 1, program akan memanggil fungsi `input_data()` yang kemungkinan telah didefinisikan di tempat lain untuk mengambil data nilai mahasiswa dan menyimpannya ke dalam vector `data_mahasiswa`.
 - case 2:** Jika pengguna memilih menu 2, program akan memanggil fungsi `lihat_data(data_mahasiswa)` untuk menampilkan seluruh data mahasiswa yang telah dimasukkan.

case 3: Jika pengguna memilih menu 3, program akan memanggil fungsi `rekap_data(data_mahasiswa)` untuk memberikan rekapitulasi data mahasiswa.

case 4: Jika pengguna memilih menu 4, program akan mencetak pesan terima kasih dan kemudian keluar dari program dengan pernyataan **`return 0;`**.

default: Jika pengguna memasukkan pilihan selain 1, 2, 3, atau 4, program akan mencetak pesan bahwa pilihan tidak valid.

- Loop akan kembali ke awal setelah pengguna selesai melakukan operasi di salah satu pilihan menu.
- Program akan terus berjalan hingga pengguna memilih menu "Keluar" (pilihan 4), pada saat itu program akan keluar dari loop dan mengakhiri eksekusi dengan pernyataan **`return 0;`**.

Dan Berikut merupakan Hasil Output Program nya :

1. Menu Tampilan Input Data Mahasiswa

```
"D:\Semester 5\Struktur Data" x + v
Selamat datang di Program Manajemen Data Mahasiswa!
By : Martogi Jekson C.Siagian

Menu Pilihan:
1. Input Data Nilai Mahasiswa
2. Lihat Seluruh Data Mahasiswa
3. Rekap Data Mahasiswa
4. Keluar
Pilih menu (1/2/3/4): 1

Masukkan jumlah mahasiswa: 3
-----
Masukkan NIM mahasiswa ke-1: 1
Masukkan Nama mahasiswa ke-1: tio
Masukkan Nilai mahasiswa ke-1: 88
Data mahasiswa berhasil dimasukkan.
-----
Masukkan NIM mahasiswa ke-2: 2
Masukkan Nama mahasiswa ke-2: rio
Masukkan Nilai mahasiswa ke-2: 89
Data mahasiswa berhasil dimasukkan.
-----
Masukkan NIM mahasiswa ke-3: 3
Masukkan Nama mahasiswa ke-3: rey
Masukkan Nilai mahasiswa ke-3: 90
Data mahasiswa berhasil dimasukkan.
-----
```

2. Menu Tampilan Lihat Seluruh Data Mahasiswa

```
"D:\Semester 5\Struktur Data" x + v
Menu Pilihan:
1. Input Data Nilai Mahasiswa
2. Lihat Seluruh Data Mahasiswa
3. Rekap Data Mahasiswa
4. Keluar
Pilih menu (1/2/3/4): 2

Data Mahasiswa:
-----
Mahasiswa ke-1:
NIM : 1
Nama : tio
Nilai : 88
-----
Mahasiswa ke-2:
NIM : 2
Nama : rio
Nilai : 89
-----
Mahasiswa ke-3:
NIM : 3
Nama : rey
Nilai : 90
```

3. Menu Tampilan Rekap data Mahasiswa & Menu Keluar

```
"D:\Semester 5\Struktur Data" x + v
Menu Pilihan:
1. Input Data Nilai Mahasiswa
2. Lihat Seluruh Data Mahasiswa
3. Rekap Data Mahasiswa
4. Keluar
Pilih menu (1/2/3/4): 3

Rekap Data Mahasiswa:
-----
Nilai Terbesar:
NIM : 3
Nama : rey
Nilai : 90
-----
Nilai Terkecil:
NIM : 1
Nama : tio
Nilai : 88
-----
Nilai Rata-rata: 89
```

4. Tampilan Menu Keluar

```
Menu Pilihan:
1. Input Data Nilai Mahasiswa
2. Lihat Seluruh Data Mahasiswa
3. Rekap Data Mahasiswa
4. Keluar
Pilih menu (1/2/3/4): 4
Terima kasih telah menggunakan program ini. Sampai jumpa!
```