

LAPORAN PRAKTIKUM
PEMROGRAMAN DASAR
IMPLEMENTASI ARRAY UNTUK MENGHITUNG
PENJUMLAHAN, PERKALIAN MATRIX DAN SIZEOF ARRAY
DALAM BAHASA PEMROGRAMAN C++



NATHANAEL KEVIN NUGRAHA
NIM: 25104410060
PERIODE: SEMESTER GANJIL 2025/2025

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ISLAM BALITAR

DAFTAR ISI

DAFTAR ISI.....	1
BAB 1 PENDAHULUAN	2
1.1 LATAR BELAKANG	2
1.2 TUJUAN PRAKTIKUM	2
BAB 2 TINJAUAN PUSTAKA	2
2.1 PERSIAPAN PRAKTIKUM.....	2
2.2 DASAR TEORI	3
BAB 3 KETENTUAN DAN SOAL PRAKTIKUM	4
BAB 4 PEMBAHASAN.....	5
BAB 5 PENUTUP	11
KESIMPULAN.....	11
SARAN.....	11

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Pemrograman dasar merupakan fondasi utama dalam mempelajari ilmu komputer. Salah satu konsep penting dalam pemrograman adalah array, yang digunakan untuk menyimpan sekumpulan data bertipe sama dalam satu variabel secara terstruktur dan berurutan. Penggunaan array sangat membantu dalam pengolahan data yang berjumlah banyak agar program menjadi lebih efisien dan mudah dikelola.

Dalam praktikum ini, mahasiswa mempelajari penggunaan array satu dimensi dan array multidimensi dalam bahasa pemrograman C++. Selain itu, mahasiswa juga mempelajari penerapan array dalam penyelesaian kasus sederhana seperti operasi matriks serta penggunaan fungsi `sizeof()` untuk mengetahui panjang suatu array. Dengan adanya praktikum ini, diharapkan mahasiswa mampu memahami konsep array serta mengimplementasikannya dalam pembuatan program yang terstruktur dan efektif.

1.2 TUJUAN PRAKTIKUM

Tujuan dari praktikum ini adalah:

1. Memahami konsep dasar array dalam pemrograman.
2. Mengetahui penggunaan array satu dimensi dan array multidimensi.
3. Mampu mengimplementasikan array untuk menyimpan dan mengolah data dalam program.
4. Mampu menggunakan array untuk menyelesaikan permasalahan sederhana, seperti operasi matriks dan penggunaan fungsi `sizeof()` untuk mengetahui panjang array.

BAB 2

TINJAUAN PUSTAKA

2.1 PERSIAPAN PRAKTIKUM

Saya menggunakan Perangkat Keras dan Perangkat Lunak dalam praktikum ini. Perangkat Keras saya menggunakan 1 buah laptop dengan spesifikasi RAM 8 GB, Prosessor Intel I3 Generasi 12. Dan Perangkat Lunak yang saya gunakan adalah Visual Studio 2022 dengan menggunakan Bahasa Pemrograman C++.

2.2 DASAR TEORI

Array merupakan salah satu struktur data dasar dalam pemrograman yang digunakan untuk menyimpan sekumpulan data yang memiliki tipe yang sama dalam satu variabel. Data-data tersebut disimpan secara berurutan di dalam memori dan dapat diakses menggunakan indeks tertentu. Penggunaan array bertujuan untuk mempermudah pengolahan data dalam jumlah banyak agar program menjadi lebih rapi, efisien, dan mudah dipahami.

Dalam bahasa pemrograman C++, array dideklarasikan dengan menentukan tipe data, nama array, serta jumlah elemen yang akan disimpan. Setiap elemen dalam array memiliki alamat memori yang saling berurutan (kontigu), sehingga proses akses data dapat dilakukan dengan cepat. Indeks pada array dimulai dari angka nol, sehingga elemen pertama berada pada indeks ke-0.

Berdasarkan jumlah dimensinya, array dibedakan menjadi dua jenis, yaitu array satu dimensi dan array multidimensi. Array satu dimensi merupakan array yang hanya memiliki satu baris data dan biasanya digunakan untuk menyimpan daftar nilai. Sementara itu, array multidimensi merupakan array yang memiliki lebih dari satu dimensi, seperti array dua dimensi yang sering digunakan untuk merepresentasikan tabel atau matriks yang terdiri dari baris dan kolom.

Setiap elemen array dapat diakses dan diolah menggunakan perulangan (looping), seperti perulangan for, sehingga memudahkan dalam proses pengisian maupun penampilan data. Pada array multidimensi, biasanya digunakan perulangan bersarang (nested loop) untuk mengakses setiap elemen berdasarkan baris dan kolomnya.

Selain itu, bahasa pemrograman C++ menyediakan fungsi `sizeof()` yang dapat digunakan untuk mengetahui ukuran array dalam satuan byte. Dengan membagi ukuran total array dengan ukuran satu elemen, maka dapat diketahui jumlah atau panjang elemen dalam array tersebut. Hal ini sangat berguna ketika jumlah elemen array tidak ditentukan secara langsung.

Dengan memahami konsep array dan cara penggunaannya, programmer dapat mengolah data secara lebih terstruktur dan efektif. Penerapan array dalam praktikum ini membantu mahasiswa dalam menyelesaikan permasalahan sederhana, seperti operasi matriks dan pengolahan data numerik, sehingga program yang dihasilkan menjadi lebih efisien dan mudah dikembangkan.

BAB 3

KETENTUAN DAN SOAL PRAKTIKUM

SOAL :

1. Buatlah suatu program untuk menghitung operasi penjumlahan pada suatu matriks 3x3.
2. Buatlah suatu program untuk menghitung operasi perkalian pada suatu matriks 3x3.
3. Berikan contoh program yang mana menggunakan `sizeof()` untuk mengetahui jumlah dari panjang array.

BAB 4

PEMBAHASAN

SOAL 1 : Praktikum-pemrograman-dasar-2-penjumlahan.cpp

SOURCE CODE :

```
Praktikum-pe...jumlahan.cpp
Praktikum-pemrograman-dasar-2-penjumlahan
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int A[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
6      int B[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
7      int C[3][3];
8
9      for (int i = 0; i < 3; i++) {
10         for (int j = 0; j < 3; j++) {
11             C[i][j] = A[i][j] + B[i][j];
12         }
13     }
14
15     cout << "\nHasil penjumlahan matriks:\n";
16     for (int i = 0; i < 3; i++) {
17         for (int j = 0; j < 3; j++) {
18             cout << C[i][j] << " ";
19         }
20         cout << endl;
21     }
22
23     return 0;
24 }
25
```

PENJELASAN CODE:

```
int main() {
    int A[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
    int B[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
    int C[3][3];
}
```

Program ini menggunakan array dua dimensi untuk melakukan operasi penjumlahan matriks berukuran 3×3 . Pada program ini, array A dan array B dideklarasikan sebagai matriks 3×3 yang masing-masing telah diinisialisasi dengan nilai tertentu. Array C dideklarasikan sebagai array dua dimensi yang berfungsi untuk menyimpan hasil penjumlahan dari setiap elemen matriks A dan matriks B. Penggunaan array dua dimensi memungkinkan data disusun dalam bentuk baris dan kolom sehingga memudahkan pengolahan data matriks.

```
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        C[i][j] = A[i][j] + B[i][j];
    }
}
```

Proses penjumlahan matriks dilakukan menggunakan perulangan bersarang (nested loop). Perulangan pertama digunakan untuk mengakses baris matriks, sedangkan perulangan kedua

digunakan untuk mengakses kolom matriks. Pada setiap iterasi, elemen matriks A dan B yang memiliki indeks baris dan kolom yang sama dijumlahkan, kemudian hasilnya disimpan ke dalam array C dengan pernyataan $C[i][j] = A[i][j] + B[i][j]$. Dengan cara ini, seluruh elemen matriks dapat dijumlahkan secara sistematis.

```
cout << "\nHasil penjumlahan matriks:\n";
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        cout << C[i][j] << " ";
    }
    cout << endl;
}

return 0;
}
```

Setelah proses penjumlahan selesai, hasil penjumlahan matriks ditampilkan ke layar menggunakan perulangan bersarang. Setiap elemen dari array C dicetak berdasarkan indeks baris dan kolomnya, sehingga hasil yang ditampilkan berbentuk matriks. Perintah `cout << endl;` digunakan untuk memindahkan baris setelah satu baris matriks dicetak, agar tampilan output lebih rapi dan mudah dibaca.

Hasil OUTPUT :

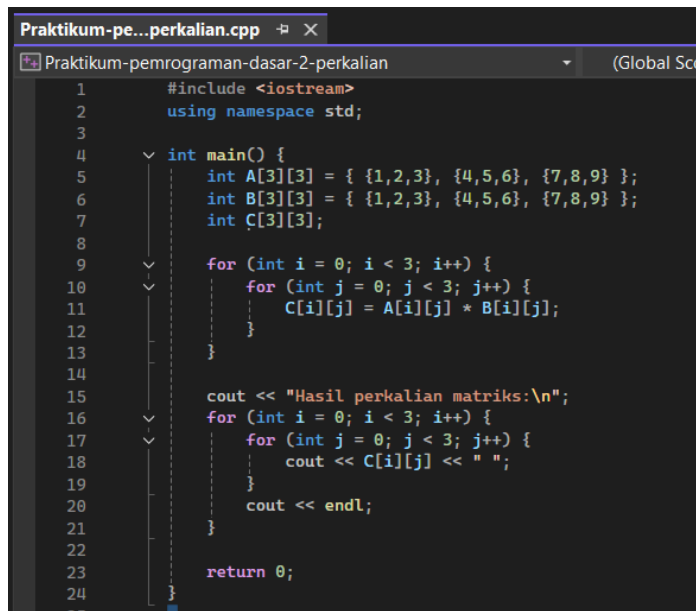
```
Hasil penjumlahan matriks:
2 4 6
8 10 12
14 16 18
D:\Praktikum-pemrograman-dasar-2-penjumlahan\
```

PENJELASAN OUTPUT :

Output yang dihasilkan dari program ini adalah matriks hasil penjumlahan dari matriks A dan matriks B. Baris pertama menampilkan nilai 2 4 6, yang diperoleh dari hasil penjumlahan elemen baris pertama pada kedua matriks. Baris kedua menampilkan nilai 8 10 12, dan baris ketiga menampilkan nilai 14 16 18. Hasil tersebut menunjukkan bahwa setiap elemen matriks dijumlahkan dengan elemen lain yang berada pada posisi baris dan kolom yang sama.

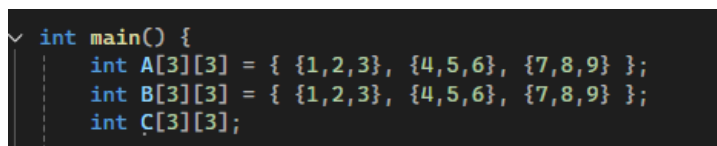
SOAL 2 : Praktikum-pemrograman-dasar-2-perkalian.cpp

SOURCE CODE :



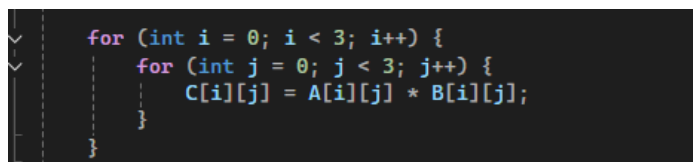
```
1      #include <iostream>
2      using namespace std;
3
4      int main() {
5          int A[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
6          int B[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
7          int C[3][3];
8
9          for (int i = 0; i < 3; i++) {
10             for (int j = 0; j < 3; j++) {
11                 C[i][j] = A[i][j] * B[i][j];
12             }
13         }
14
15         cout << "Hasil perkalian matriks:\n";
16         for (int i = 0; i < 3; i++) {
17             for (int j = 0; j < 3; j++) {
18                 cout << C[i][j] << " ";
19             }
20             cout << endl;
21         }
22
23         return 0;
24     }
```

PENJELASAN CODE:



```
int main() {
    int A[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
    int B[3][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
    int C[3][3];
}
```

Program ini menggunakan array dua dimensi untuk melakukan operasi perkalian matriks berukuran 3×3 . Pada program ini, array A dan array B dideklarasikan sebagai matriks 3×3 yang masing-masing telah diinisialisasi dengan nilai tertentu. Array C digunakan untuk menyimpan hasil perkalian dari elemen-elemen yang bersesuaian pada matriks A dan matriks B. Dengan menggunakan array dua dimensi, data dapat disusun dalam bentuk baris dan kolom sehingga mudah untuk diolah sebagai matriks.



```
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        C[i][j] = A[i][j] * B[i][j];
    }
}
```

Sama seperti penjumlahan proses perkalian dilakukan menggunakan perulangan bersarang (nested loop). Perulangan pertama digunakan untuk mengakses baris matriks, sedangkan perulangan kedua digunakan untuk mengakses kolom matriks. Pada setiap iterasi, elemen pada matriks A dikalikan dengan elemen pada matriks B yang memiliki indeks baris dan kolom yang

sama. Hasil perkalian tersebut kemudian disimpan ke dalam array C dengan pernyataan $C[i][j] = A[i][j] * B[i][j]$. Proses ini dilakukan hingga seluruh elemen matriks selesai dikalikan.

```
cout << "Hasil perkalian matriks:\n";
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        cout << C[i][j] << " ";
    }
    cout << endl;
}

return 0;
```

Setelah proses perkalian selesai, hasil perkalian matriks ditampilkan ke layar menggunakan perulangan bersarang. Setiap elemen dari array C dicetak secara berurutan sesuai dengan baris dan kolomnya. Perintah `cout << endl;` digunakan untuk memindahkan baris setelah satu baris matriks dicetak, sedangkan tambahan `endl` berikutnya digunakan untuk memberi jarak antarbaris agar tampilan output lebih jelas dan mudah dibaca.

Hasil OUTPUT :

```
Hasil perkalian matriks:
1 4 9
16 25 36
49 64 81

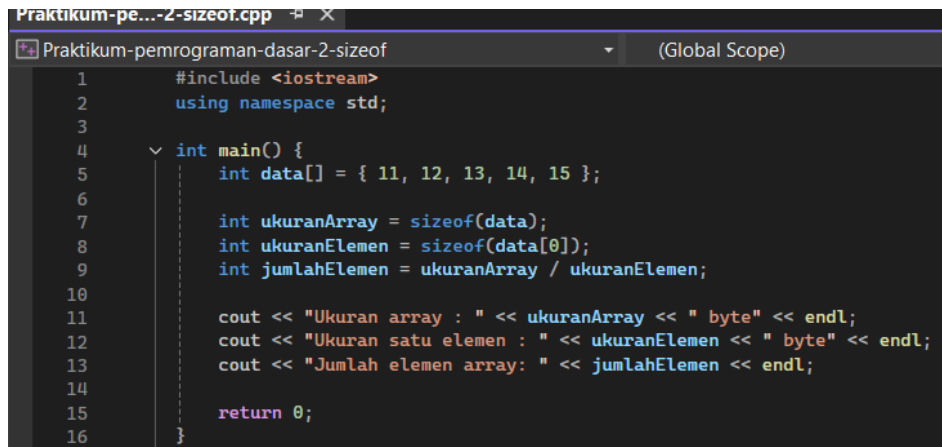
D:\Praktikum-pemrograman-dasar-2-perkalian\
```

PENJELASAN OUTPUT :

Output yang dihasilkan dari program ini adalah matriks hasil perkalian elemen antara matriks A dan matriks B. Baris pertama menampilkan nilai 1 4 9, yang diperoleh dari hasil perkalian elemen baris pertama pada kedua matriks. Baris kedua menampilkan nilai 16 25 36, dan baris ketiga menampilkan nilai 49 64 81. Hasil tersebut menunjukkan bahwa setiap elemen matriks dikalikan dengan elemen lain yang berada pada posisi baris dan kolom yang sama.

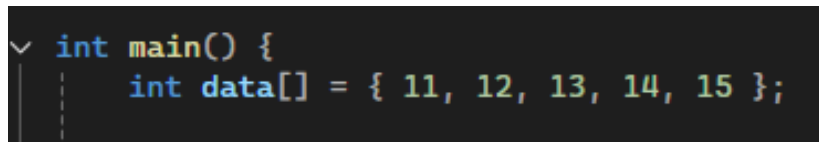
SOAL 3 : Praktikum-pemrograman-dasar-2-sizeof.cpp

SOURCE CODE :



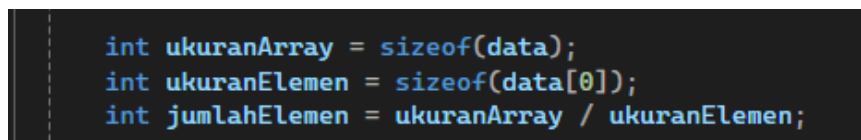
```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int data[] = { 11, 12, 13, 14, 15 };
6
7      int ukuranArray = sizeof(data);
8      int ukuranElemen = sizeof(data[0]);
9      int jumlahElemen = ukuranArray / ukuranElemen;
10
11     cout << "Ukuran array : " << ukuranArray << " byte" << endl;
12     cout << "Ukuran satu elemen : " << ukuranElemen << " byte" << endl;
13     cout << "Jumlah elemen array: " << jumlahElemen << endl;
14
15     return 0;
16 }
```

PENJELASAN CODE:



```
int main() {
    int data[] = { 11, 12, 13, 14, 15 };
```

Program ini digunakan untuk mengetahui ukuran dan jumlah elemen pada sebuah array dengan memanfaatkan fungsi `sizeof()` dalam bahasa pemrograman C++. Pada program ini, array data dideklarasikan sebagai array satu dimensi yang berisi lima elemen bertipe `int`, yaitu 11, 12, 13, 14, dan 15. Array tersebut digunakan sebagai contoh untuk menunjukkan cara menghitung ukuran array dan jumlah elemennya secara otomatis.



```
int ukuranArray = sizeof(data);
int ukuranElemen = sizeof(data[0]);
int jumlahElemen = ukuranArray / ukuranElemen;
```

Untuk mengetahui ukuran total array, digunakan fungsi `sizeof(data)`. Fungsi ini mengembalikan ukuran keseluruhan array dalam satuan byte. Karena array data terdiri dari lima elemen bertipe `int`, dan setiap tipe data `int` umumnya memiliki ukuran 4 byte, maka ukuran total array adalah hasil perkalian antara jumlah elemen dan ukuran satu elemen.

Selanjutnya, ukuran satu elemen array diperoleh dengan menggunakan `sizeof(data[0])`. Pernyataan ini digunakan untuk mengetahui ukuran satu elemen pertama dalam array, yang bertipe `int`. Nilai ini digunakan sebagai pembagi untuk menentukan jumlah elemen dalam array, sehingga program dapat menghitung panjang array tanpa harus menuliskannya secara manual. Selanjutnya jumlah elemen array dihitung dengan membagi ukuran total array dengan ukuran satu elemen, yaitu `jumlahElemen = ukuranArray / ukuranElemen`.

```
cout << "Ukuran array : " << ukuranArray << " byte" << endl;  
cout << "Ukuran satu elemen : " << ukuranElemen << " byte" << endl;  
cout << "Jumlah elemen array: " << jumlahElemen << endl;  
  
return 0;  
}
```

Hasil perhitungan tersebut kemudian ditampilkan ke layar menggunakan perintah cout. Dengan cara ini, jumlah elemen dalam array dapat diketahui secara otomatis dan lebih fleksibel jika isi array diubah.

Hasil OUTPUT :

```
Ukuran array : 20 byte  
Ukuran satu elemen : 4 byte  
Jumlah elemen array: 5  
  
D:\Praktikum-pemrograman-dasar-2-sizeof\
```

PENJELASAN OUTPUT :

Output yang dihasilkan dari program ini menunjukkan bahwa ukuran total array adalah 20 byte, ukuran satu elemen adalah 4 byte, dan jumlah elemen dalam array adalah 5. Nilai tersebut diperoleh karena array terdiri dari lima elemen bertipe int, di mana setiap elemen memiliki ukuran 4 byte. Hal ini membuktikan bahwa fungsi sizeof() dapat digunakan untuk mengetahui panjang array secara efektif.

BAB 5

PENUTUP

KESIMPULANa

Setelah melakukan praktikum ini, saya dapat memahami konsep penggunaan array dalam pemrograman C++. Saya memahami bahwa array digunakan untuk menyimpan sekumpulan data yang memiliki tipe yang sama dalam satu variabel sehingga pengolahan data menjadi lebih terstruktur dan efisien. Saya juga dapat membedakan penggunaan array satu dimensi dan array multidimensi serta cara mengakses elemen array menggunakan indeks. Selain itu, dari tugas yang diberikan, saya memahami bagaimana array digunakan untuk menyelesaikan permasalahan sederhana seperti penjumlahan dan perkalian matriks serta penggunaan fungsi `sizeof()` untuk mengetahui panjang atau jumlah elemen dalam array.

SARAN

Saya memiliki saran agar pada pengembangan program selanjutnya, pengisian dan pengolahan data array tidak hanya menggunakan nilai yang telah ditentukan di dalam program, tetapi dapat menggunakan input dari pengguna. Dengan demikian, program menjadi lebih fleksibel dan dapat digunakan untuk berbagai data yang berbeda.

Link Github : https://github.com/natankev/PRAKTIKUM_Pemrograman-Dasar-2-UNISBA