

# **LAPORAN PRAKTIKUM PEMROGRAMAN DASAR**

**Array**



**NAMA: Firdaus Bintang Maulana Jesa**

**NIM: 25104410034**

**PERIODE: SEMESTER GANJIL 2024/2025**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ISLAM BALITAR**

## NO.1 MATRIKS PENJUMLAHAN

### 1. Deskripsi Tugas

Tugas ini bertujuan untuk membuat program menggunakan bahasa pemrograman C++ yang dapat melakukan penjumlahan dua buah matriks berukuran  $3 \times 3$  menggunakan array dua dimensi. Program menerima input elemen matriks dari pengguna, menampilkan hasil penjumlahan, serta menggunakan fungsi `sizeof()` untuk mengetahui ukuran dan jumlah elemen matriks.

### 2. Algoritma Penyelesaian

- 1) Mulai program.
- 2) Deklarasikan tiga buah matriks  $3 \times 3$ .
- 3) Input elemen matriks A dan B.
- 4) Lakukan penjumlahan matriks.
- 5) Hitung ukuran dan jumlah elemen matriks menggunakan `sizeof()`.
- 6) Tampilkan hasil penjumlahan dan informasi matriks.
- 7) Selesai.

### 3. Penjelasan Fungsi Dan Code Program

#### A. Deklarasi Matriks

```
int main() {  
    int A[3][3], B[3][3], C[3][3];
```

Kode ini berfungsi untuk mendeklarasikan tiga buah array dua dimensi berukuran  $3 \times 3$ . Matriks A dan B digunakan sebagai input, sedangkan matriks C digunakan untuk menyimpan hasil penjumlahan.

#### B. Input Elemen Matriks

```
cout << "Masukkan elemen matriks A (3x3):" << "\n";  
for (int i = 0; i < 3; i++) {  
    for (int j = 0; j < 3; j++) {  
        cin >> A[i][j];  
    }  
}
```

Potongan kode ini digunakan untuk memasukkan nilai ke dalam matriks A menggunakan perulangan bersarang. Perulangan pertama mengatur baris, sedangkan perulangan kedua mengatur kolom.

### C. Proses Penjumlahan Matriks

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

Kode ini merupakan inti dari program penjumlahan matriks. Setiap elemen matriks A dijumlahkan dengan elemen matriks B yang memiliki indeks yang sama, lalu hasilnya disimpan ke dalam matriks C.

### D. Penggunaan Fungsi Sizeof

```
int ukuranMatriks = sizeof(A);  
int ukuranElemen = sizeof(A[0][0]);  
int jumlahElemen = ukuranMatriks / ukuranElemen;
```

Kode ini digunakan untuk mengetahui ukuran total memori matriks dan jumlah elemen matriks. Fungsi `sizeof(A)` menghitung seluruh ukuran matriks, sedangkan `sizeof(A[0][0])` menghitung ukuran satu elemen. Hasil pembagian menunjukkan bahwa matriks  $3 \times 3$  memiliki 9 elemen.

## 4. Source Code Program

```
Matrks_Penjumlahan  
1  #include <iostream>  
2  using namespace std;  
3  
4  int main() {  
5      int A[3][3], B[3][3], C[3][3];  
6  
7      // Input matriks A  
8      cout << "Masukkan elemen matriks A (3x3):" << "\n";  
9      for (int i = 0; i < 3; i++) {  
10         for (int j = 0; j < 3; j++) {  
11             cin >> A[i][j];  
12         }  
13     }  
14  
15     // Input matriks B  
16     cout << "Masukkan elemen matriks B (3x3):" << "\n";  
17     for (int i = 0; i < 3; i++) {  
18         for (int j = 0; j < 3; j++) {  
19             cin >> B[i][j];  
20         }  
21     }  
22  
23     // Proses penjumlahan matriks  
24     for (int i = 0; i < 3; i++) {  
25         for (int j = 0; j < 3; j++) {  
26             C[i][j] = A[i][j] + B[i][j];  
27         }  
28     }  
29  
30     // Menghitung ukuran matriks menggunakan sizeof()  
31     int ukuranMatriks = sizeof(A);  
32     int ukuranElemen = sizeof(A[0][0]);  
33     int jumlahElemen = ukuranMatriks / ukuranElemen;  
34  
35     // Output hasil penjumlahan  
36     cout << "\nHasil penjumlahan matriks A dan B:" << "\n";  
37     for (int i = 0; i < 3; i++) {  
38         for (int j = 0; j < 3; j++) {  
39             cout << C[i][j] << " ";  
40         }  
41         cout << "\n";  
42     }  
43  
44     // Output informasi sizeof  
45     cout << "\nUkuran total matriks (byte) : " << ukuranMatriks << "\n";  
46     cout << "Jumlah elemen matriks : " << jumlahElemen << "\n";  
47  
48     return 0;  
49 }  
50
```

## 5. Hasil Output Program

```
Microsoft Visual Studio Debu... X + v
Masukkan elemen matriks A (3x3):
1 2 3
1 2 3
1 2 3
Masukkan elemen matriks B (3x3):
1 2 3
1 2 3
1 2 3

Hasil penjumlahan matriks A dan B:
2 4 6
2 4 6
2 4 6

Ukuran total matriks (byte) : 36
Jumlah elemen matriks      : 9

C:\Users\LENOVO\OneDrive\Documents\Tuga...
(process 24404) exited with code 0 (0)
```

## 6. Kesimpulan

Program penjumlahan matriks  $3 \times 3$  berhasil dibuat menggunakan array dua dimensi dan perulangan bersarang. Setiap elemen matriks dapat diakses berdasarkan indeks baris dan kolom sehingga proses penjumlahan berjalan dengan benar. Penggunaan fungsi `sizeof()` membantu mengetahui ukuran dan jumlah elemen matriks, sehingga menambah pemahaman tentang struktur array dalam bahasa pemrograman C++.

## NO.2 MATRIKS PERKALIAN

### 1. Deskripsi Tugas

Tugas ini bertujuan untuk membuat program C++ yang dapat melakukan perkalian dua buah matriks berukuran  $3 \times 3$  menggunakan array dua dimensi dan perulangan bersarang. Program juga menggunakan fungsi `sizeof()` untuk mengetahui ukuran dan jumlah elemen matriks.

### 2. Algoritma Penyelesaian

- 1) Mulai program.
- 2) Deklarasikan matriks A, B, dan C berukuran  $3 \times 3$ .
- 3) Input elemen matriks A dan B.
- 4) Inisialisasi matriks hasil dengan nol.
- 5) Lakukan perkalian matriks.
- 6) Hitung ukuran dan jumlah elemen matriks menggunakan `sizeof()`.
- 7) Tampilkan hasil perkalian.
- 8) Selesai.

### 3. Penjelasan Fungsi Dan Code Program

#### A. Inisialisasi Matriks Hasil

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

Kode ini digunakan untuk menginisialisasi seluruh elemen matriks C dengan nilai nol sebelum proses perkalian dilakukan agar hasil perhitungan tidak salah.

#### B. Proses Perkalian Matriks

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

Kode ini merupakan inti dari proses perkalian matriks. Setiap elemen matriks C diperoleh dari penjumlahan hasil kali elemen baris matriks A dengan elemen kolom matriks B.

### C. Penggunaan Fungsi sizeof()

```
int ukuranMatriks = sizeof(A);  
int ukuranElemen = sizeof(A[0][0]);  
int jumlahElemen = ukuranMatriks / ukuranElemen;
```

Kode ini digunakan untuk mengetahui ukuran total matriks dan jumlah elemen matriks.

## 4. Source Code Program

```
Matrks_perkalian  
1 #include <iostream>  
2 using namespace std;  
3  
4 int main() {  
5     int A[3][3], B[3][3], C[3][3];  
6  
7     // Input matriks A  
8     cout << "Masukkan elemen matriks A (3x3):" << "\n";  
9     for (int i = 0; i < 3; i++) {  
10         for (int j = 0; j < 3; j++) {  
11             cin >> A[i][j];  
12         }  
13     }  
14  
15     // Input matriks B  
16     cout << "Masukkan elemen matriks B (3x3):" << "\n";  
17     for (int i = 0; i < 3; i++) {  
18         for (int j = 0; j < 3; j++) {  
19             cin >> B[i][j];  
20         }  
21     }  
22  
23     // Inisialisasi matriks hasil  
24     for (int i = 0; i < 3; i++) {  
25         for (int j = 0; j < 3; j++) {  
26             C[i][j] = 0;  
27         }  
28     }  
29  
30     // Proses perkalian matriks  
31     for (int i = 0; i < 3; i++) {  
32         for (int j = 0; j < 3; j++) {  
33             for (int k = 0; k < 3; k++) {  
34                 C[i][j] += A[i][k] * B[k][j];  
35             }  
36         }  
37     }  
38  
39     // Menghitung ukuran matriks menggunakan sizeof  
40     int ukuranMatriks = sizeof(A);  
41     int ukuranElemen = sizeof(A[0][0]);  
42     int jumlahElemen = ukuranMatriks / ukuranElemen;  
43  
44     // Output hasil perkalian  
45     cout << "Hasil perkalian matriks A dan B:" << "\n";  
46     for (int i = 0; i < 3; i++) {  
47         for (int j = 0; j < 3; j++) {  
48             cout << C[i][j] << " ";  
49         }  
50         cout << "\n";  
51     }  
52  
53     // Output informasi sizeof  
54     cout << "Ukuran total matriks (byte) : " << ukuranMatriks << "\n";  
55     cout << "Jumlah elemen matriks : " << jumlahElemen << "\n";  
56  
57     return 0;  
58 }
```

## 5. Hasil Output Program

```
Microsoft Visual Studio Debu  
Masukkan elemen matriks A (3x3):  
1 2 3  
1 2 3  
1 2 3  
Masukkan elemen matriks B (3x3):  
1 2 3  
1 2 3  
1 2 3  
Hasil perkalian matriks A dan B:  
6 12 18  
6 12 18  
6 12 18  
Ukuran total matriks (byte) : 36  
Jumlah elemen matriks : 9  
C:\Users\LENOVO\OneDrive\Documents\T
```

## **6. Kesimpulan**

Program perkalian matriks  $3 \times 3$  dapat dijalankan dengan baik menggunakan array dua dimensi dan tiga perulangan bersarang. Proses perkalian dilakukan sesuai dengan aturan matematika matriks dan menghasilkan output yang tepat. Penerapan fungsi `sizeof()` memberikan pemahaman tambahan mengenai ukuran memori dan jumlah elemen matriks yang digunakan dalam program.

## NO. 3 PENGGUNAAN SIZEOF

### 1. Deskripsi Tugas

Soal nomor 3 bertujuan untuk memahami penggunaan fungsi sizeof() dalam menentukan ukuran dan jumlah elemen array. Pada praktikum ini, fungsi sizeof() telah diterapkan langsung pada program penjumlahan dan perkalian matriks.

### 2. Algoritma Penyelesaian

- 1) Gunakan array dua dimensi.
- 2) Hitung ukuran total array menggunakan sizeof().
- 3) Hitung ukuran satu elemen array.
- 4) Tentukan jumlah elemen array.
- 5) Tampilkan hasilnya

### 3. Penjelasan Fungsi Sizeof

```
int ukuranMatriks = sizeof(A);  
int ukuranElemen = sizeof(A[0][0]);  
int jumlahElemen = ukuranMatriks / ukuranElemen;
```

```
int ukuranMatriks = sizeof(A);  
int ukuranElemen = sizeof(A[0][0]);  
int jumlahElemen = ukuranMatriks / ukuranElemen;
```

Fungsi sizeof() digunakan untuk mengetahui ukuran memori array yang dialokasikan. Dengan membagi ukuran total array dengan ukuran satu elemen, jumlah elemen array dapat diketahui.

### 4. Kesimpulan

Fungsi sizeof() digunakan untuk mengetahui ukuran memori array yang dialokasikan dan jumlah elemennya, bukan jumlah data yang diinput. Dengan mengintegrasikan sizeof() pada program penjumlahan dan perkalian matriks, pemahaman mengenai cara kerja array di dalam memori pada bahasa pemrograman C++ menjadi lebih jelas.