

MAKALAH TENTANG ARRAY



Disusun oleh :

Nama	: L Hafid Alkhair
NIM	: 2023903430060
Kelas	: TRKJ 1.C
Jurusan	: Teknologi Informasi dan Komputer
Program Studi	: Teknologi Rekayasa Komputer Jaringan
Dosen Pengajar	: Indrawati, SST. MT



**JURUSAN TEKNOLOGI INFORMASI KOMPUTER
PRODI TEKNOLOGI REKAYASA KOMPUTER JARINGAN
POLITEKNIK NEGERI LHOKSEUMAWE
TAHUN AJARAN 2023/2024**

DAFTAR ISI

BAB 1.....	3
PEMBAHASAN	3
PENGERTIAN ARRAY.....	3
A. Pengenalan ke Array Satu Dimensi	4
1. Mengapa Array Satu Dimensi Penting	5
2. Penggunaan Array Satu Dimensi.....	7
3. Operasi pada Array Satu Dimensi.	9
4. Kendala dan Pertimbangan.....	10
5. Kesimpulan Array 1 dimensi	11
B. Pengenalan ke Array Dua Dimensi.....	11
1. Mengapa Array Dua Dimensi Penting.....	12
2. Penggunaan Array Dua Dimensi	13
3. Operasi pada Array Dua Dimensi.....	15
4. Kendala dan Pertimbangan.....	16
5. Kesimpulan Array 2 Dimensi	17
C. Pengenalan ke Array Tiga Dimensi.....	18
1. Mengapa Array Tiga Dimensi Penting.....	19
2. Penggunaan Array Tiga Dimensi	20
3. Operasi pada Array Tiga Dimensi.....	21
4. Kendala dan Pertimbangan.....	23
5. Kesimpulan Array 3 Dimensi	24
D. PROGRAM ARRAY	25
1. Program Array 1 Dimensi	25
2. Program Array 2 Dimensi	28
3. Program Array 3 dimensi	29
DAFTAR PUSTAKA	34

BAB 1

PEMBAHASAN

PENGERTIAN ARRAY

Array adalah struktur data yang digunakan dalam pemrograman untuk menyimpan kumpulan nilai atau elemen-elemen data sejenis dalam satu variabel. Elemen-elemen ini disusun dalam urutan tertentu dan diidentifikasi oleh indeks atau posisi relatif mereka dalam array.

Konsep Array adalah bagian integral dari bahasa pemrograman C yang dikembangkan oleh Dennis Ritchie di Laboratorium Bell pada tahun 1972. Array digunakan untuk mengatur dan mengelola data dalam berbagai program C, dan ini adalah salah satu fitur dasar dalam bahasa tersebut. Jadi, tidak ada satu penemu tunggal untuk konsep array dalam bahasa C.

Beberapa karakteristik utama dari array adalah:

1. Tipe Data Seragam:

Semua elemen dalam array harus memiliki tipe data yang sama, misalnya, integer, string, atau float.

2. Indeks

Elemen-elemen dalam array diakses melalui indeks numerik atau posisi relatif mereka dalam array. Indeks dimulai dari 0 (nol) untuk elemen pertama, 1 untuk elemen kedua, dan seterusnya.

3. Ukuran Tetap

Saat Anda mendefinisikan sebuah array, Anda harus menentukan ukuran atau panjangnya. Ini berarti jumlah elemen dalam array tetap selama program berjalan, kecuali jika Anda membuat array dinamis dalam bahasa tertentu.

4. Penyimpanan Terkompak

Array menyimpan elemen-elemen secara terkompak dalam memori karena mereka memiliki tipe data seragam.

5. Contoh Penggunaan

Berikut adalah contoh pendefinisian dan penggunaan array dalam bahasa C:

```
1  int angka[5]; // Membuat array integer dengan panjang 5
2
3  angka[0] = 10;
4  angka[1] = 20;
5  angka[2] = 30;
6  angka[3] = 40;
7  angka[4] = 50;
8
9  // Mengakses elemen-elemen array
10 int nilai = angka[2]; // Memperoleh nilai 30 dari indeks ke-2
```

Gambar 1.1 Contoh penggunaan Array

Dalam contoh di atas, angka adalah array yang menyimpan lima nilai integer dan elemennya diakses melalui indeks.

Array digunakan secara luas dalam bahasa C untuk mengelola dan memproses data dalam berbagai program, dan mereka merupakan salah satu struktur data dasar yang sangat penting dalam pemrograman.

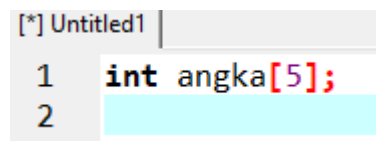
A. Pengenalan ke Array Satu Dimensi

Array adalah salah satu konsep paling fundamental dalam pemrograman komputer. Mereka adalah struktur data yang digunakan untuk menyimpan kumpulan elemen data serupa, seperti bilangan bulat, desimal, karakter, atau bahkan objek yang lebih

kompleks. Array digunakan untuk mengorganisasi dan mengakses data dengan cara yang efisien dan terstruktur.

Array satu dimensi adalah salah satu jenis array yang paling sederhana dan paling umum. Ini digunakan untuk menyimpan data dalam satu dimensi atau satu baris. Setiap elemen dalam array memiliki indeks numerik yang mengidentifikasikan posisinya dalam array. Indeks dimulai dari 0 untuk elemen pertama dan berlanjut secara berurutan hingga elemen terakhir.

Misalnya, pertimbangkan array berikut dalam bahasa C:

A screenshot of a code editor window titled "[*] Untitled1". The editor shows two lines of C code. Line 1 is "1 int angka[5];" where "1" is the line number, "int" is in blue, "angka" is in black, and "[5];" is in red. Line 2 is "2" followed by a light blue selection bar, indicating it is selected. The line numbers "1" and "2" are in a light gray margin on the left.

Gambar 1.2 Array angka

Array ini memiliki nama angka dan panjangnya adalah 5. Ini berarti array angka dapat menyimpan 5 elemen data bertipe integer. Elemen-elemen tersebut dapat diakses menggunakan indeks, seperti `angka[0]`, `angka[1]`, `angka[2]`, `angka[3]`, dan `angka[4]`. Setiap elemen dapat berisi nilai integer, dan Anda dapat melakukan berbagai operasi, seperti pengisian, pengambilan, atau pemutakhiran nilai pada elemen-elemen tersebut.

1. Mengapa Array Satu Dimensi Penting

Array satu dimensi memiliki peran yang sangat penting dalam pengembangan perangkat lunak dan komputasi. Berikut beberapa alasan mengapa array satu dimensi sangat penting:

a) Struktur Data yang Terorganisir

Array memungkinkan Anda untuk menyimpan data dalam struktur yang terorganisir. Ini membantu dalam mengelola data dengan cara yang lebih terstruktur daripada hanya menggunakan variabel individu untuk setiap elemen.

Misalnya, jika Anda perlu menyimpan skor dari 100 siswa, alih-alih membuat 100 variabel individu, Anda dapat menggunakan array satu dimensi dengan 100 elemen. Ini membuat kode Anda lebih terstruktur dan mudah dipahami.

b) Pengaksesan Data yang Efisien.

Array menyediakan cara yang efisien untuk mengakses elemen data. Karena setiap elemen memiliki indeks numerik, Anda dapat dengan mudah mengakses elemen tertentu dengan merujuk ke indeksnya. Ini lebih efisien daripada mencari elemen data dalam daftar yang panjang, yang dapat menghabiskan banyak waktu dan sumber daya komputasi.

c) Pemrosesan Data Massal

Array memungkinkan Anda untuk melakukan operasi yang sama pada banyak elemen data dalam satu kali loop atau iterasi. Ini sangat bermanfaat ketika Anda perlu menghitung rata-rata, mencari nilai maksimum atau minimum, atau melakukan operasi matematika lainnya pada sekumpulan data.

d) Implementasi Struktur Data Lain

Array adalah bagian integral dalam implementasi struktur data lain, seperti tumpukan, antrian, dan daftar tautan. Mereka digunakan sebagai dasar untuk mengimplementasikan struktur data yang lebih kompleks dalam pemrograman.

e) Efisiensi Penggunaan Memori

Array menggunakan memori secara efisien karena mereka menyimpan elemen-elemen dalam urutan yang kontigu. Ini

membuat penggunaan memori lebih efisien daripada memiliki variabel individu untuk setiap elemen.

Misalnya, jika Anda memiliki 10 variabel integer, itu akan menggunakan lebih banyak memori daripada memiliki array dengan 10 elemen integer.

2. Penggunaan Array Satu Dimensi

Array satu dimensi digunakan dalam berbagai konteks dan aplikasi dalam pemrograman komputer. Berikut beberapa contoh penggunaan umum:

a) Penyimpanan Data Sederhana

Anda dapat menggunakan array satu dimensi untuk menyimpan kumpulan data sederhana, seperti daftar nilai, skor, atau nama.

```
int nilai[10]; // Array untuk menyimpan 10 nilai
char nama[50]; // Array untuk menyimpan nama dengan panjang maksimal 50 karakter
```

Gambar 1.3 Penyimpan data Sederhana

b) Pengolahan Data

Array digunakan untuk mengolah data dalam berbagai cara. Anda dapat menghitung rata-rata, mencari nilai maksimum atau minimum, mengurutkan data, atau melakukan operasi matematika lainnya pada elemen-elemen array.

```
// Menghitung rata-rata
float total = 0;
for (int i = 0; i < 10; i++) {
    total += nilai[i];
}
float rata_rata = total / 10;
```

Gambar 1.4 Array Pengelolaan data

c) Merepresentasikan Teks

Dalam bahasa pemrograman, string sering diimplementasikan sebagai array satu dimensi karakter. Ini memungkinkan Anda untuk memanipulasi dan mengakses karakter dalam teks.

```
char teks[] = "Halo, dunia!";
```

Gambar 1.5 Mepresentasikan Teks

d) Pemrograman Berulang

Array digunakan dalam pemrograman berulang, seperti pengulangan for atau while, untuk mengakses elemen-elemen data dalam urutan.

```
for (int i = 0; i < 5; i++) {
    printf("Elemen ke-%d: %d\n", i, angka[i]);
}
```

Gambar 1.6 Pemrograman Berulang

e) Matriks

Array satu dimensi juga digunakan untuk mewakili matriks dalam matematika. Ini berguna dalam berbagai aplikasi, termasuk pengolahan gambar, grafika komputer, dan simulasi fisika.

```
int matriks[3][3]; // Array dua dimensi untuk mewakili matriks 3x3
```

Gambar 1.7 Matriks

3. Operasi pada Array Satu Dimensi.

Ketika bekerja dengan array satu dimensi, ada beberapa operasi umum yang dapat Anda lakukan:

a) Inisialisasi

Anda dapat menginisialisasi array dengan nilai awal saat Anda mendeklarasikannya, atau Anda dapat mengisi array setelahnya.

```
int angka[5] = {10, 20, 30, 40, 50}; // Inisialisasi saat mendeklarasikan
```

Gambar 1.8 Inisialisai

b) Mengakses Elemen

Anda dapat mengakses elemen-elemen array dengan merujuk ke indeksnya. Indeks dimulai dari 0.

```
int nilai = angka[2]; // Mengakses elemen ke-2
```

Gambar 1.9 Mengakses Elemen

c) Mengubah Nilai Elemen

Anda dapat mengubah nilai elemen array dengan mengassign nilai baru ke indeks yang sesuai.

```
angka[2] = 35; // Mengubah nilai elemen ke-2 menjadi 35
```

Gambar 1.10 Nilai Elemen

d) Iterasi Melalui Array

Anda dapat menggunakan loop, seperti for atau while, untuk mengiterasi melalui semua elemen dalam array.

```
for (int i = 0; i < 5; i++) {  
    printf("Elemen ke-%d: %d\n", i, angka[i]);  
}
```

Gambar 1.11 Iterasi Melalui Array

e) Menghitung Panjang Array

Untuk mengetahui panjang (jumlah elemen) dari array, Anda dapat menggunakan operator sizeof atau dengan menghitung panjang saat mendeklarasikan array.

```
int panjang = sizeof(angka) / sizeof(angka[0]); // Menghitung panjang array
```

Gambar 1.12 Menghitung Panjang Array

4. Kendala dan Pertimbangan

Meskipun array satu dimensi adalah alat yang sangat berguna dalam pemrograman, ada beberapa kendala dan pertimbangan yang perlu diperhatikan:

a) Ukuran Tetap

Panjang array harus ditentukan saat mendeklarasikannya, dan itu tidak dapat diubah selama runtime. Jadi, Anda harus memilih ukuran yang sesuai dengan kebutuhan Anda.

b) Pengelolaan Memori

Array menggunakan sejumlah memori yang tetap tergantung pada jumlah elemennya. Ini bisa menjadi masalah jika Anda bekerja

dengan data yang sangat besar atau jika Anda perlu menghemat memori.

c) Indeks Out-of-Bounds

Jika Anda mencoba mengakses elemen di luar rentang indeks yang valid, Anda dapat menyebabkan kesalahan atau perilaku yang tidak terduga dalam program Anda. Oleh karena itu, Anda harus selalu memeriksa batasan indeks saat mengakses elemen array.

5. Kesimpulan Array 1 dimensi

Array satu dimensi adalah salah satu konsep paling penting dalam pemrograman komputer. Mereka digunakan untuk mengorganisasi, menyimpan, dan mengakses data dalam berbagai aplikasi. Array menyediakan cara yang efisien dan terstruktur untuk mengelola kumpulan data, dan mereka digunakan dalam berbagai konteks, mulai dari pengolahan data sederhana hingga pemrograman berulang yang kompleks. Dengan memahami konsep array satu dimensi, Anda dapat memperluas kemampuan pemrograman Anda dan membuat kode yang lebih efisien dan terstruktur.

B. Pengenalan ke Array Dua Dimensi

Array dua dimensi adalah salah satu konsep paling penting dalam pemrograman komputer. Ini adalah jenis array yang digunakan untuk menyimpan data dalam dua dimensi atau sebagai matriks. Array ini memiliki baris dan kolom, dan setiap elemen dalam array dapat diidentifikasi oleh dua indeks: satu untuk baris dan satu lagi untuk kolom.

Misalnya, pertimbangkan array dua dimensi berikut dalam bahasa C:

```
int matriks[3][3];
```

Gambar 2.1 Array Matriks

Array ini memiliki nama matriks dan memiliki tiga baris dan tiga kolom. Ini berarti ada total 9 elemen dalam array. Elemen-elemen ini dapat diakses menggunakan dua indeks, seperti matriks[0][0], matriks[0][1], matriks[0][2], dan seterusnya.

Array dua dimensi sangat berguna ketika Anda perlu mengorganisasi data dalam struktur yang lebih kompleks daripada array satu dimensi. Mereka digunakan untuk merepresentasikan matriks, tabel, atau kumpulan data dua dimensi lainnya. Array ini digunakan dalam berbagai aplikasi, termasuk pengolahan gambar, pemodelan matematika, pengembangan permainan, dan analisis data.

1. Mengapa Array Dua Dimensi Penting

Array dua dimensi memiliki peran yang sangat penting dalam pengembangan perangkat lunak dan komputasi. Berikut beberapa alasan mengapa array dua dimensi sangat penting:

a) Merepresentasikan Data Dua Dimensi

Array dua dimensi adalah alat yang kuat untuk merepresentasikan data yang memiliki struktur dua dimensi, seperti matriks atau tabel. Mereka memungkinkan Anda untuk mengorganisasi data dalam bentuk yang terstruktur dan mudah dipahami.

b) Pengaksesan Data yang Terstruktur

Karena setiap elemen dalam array dua dimensi memiliki dua indeks, Anda dapat mengakses data dengan cara yang sangat terstruktur.

Misalnya, jika Anda ingin mengambil elemen pada baris ke-2 dan kolom ke-3 dari matriks, Anda dapat melakukan ini dengan menggunakan indeks matriks[1][2].

c) Pemrosesan Data Dua Dimensi

Array dua dimensi memungkinkan Anda untuk melakukan operasi yang melibatkan data dua dimensi dengan mudah. Anda dapat menghitung jumlah baris atau kolom, mengalikan dua matriks, atau melakukan berbagai operasi matematika lainnya pada elemen-elemen matriks.

d) Pemrograman Grafis dan Pengolahan Gambar

Dalam pemrograman grafis dan pengolahan gambar, array dua dimensi digunakan untuk merepresentasikan gambar, piksel, dan citra. Setiap elemen dalam matriks dua dimensi mewakili nilai warna atau intensitas piksel di lokasi tertentu dalam gambar.

e) Pengembangan Permainan

Dalam pengembangan permainan, array dua dimensi digunakan untuk merepresentasikan peta permainan, papan permainan, dan banyak elemen permainan lainnya. Mereka digunakan untuk mengatur posisi objek dalam permainan dan untuk mendeteksi tabrakan dan interaksi lainnya.

f) Analisis Data

Dalam analisis data, array dua dimensi digunakan untuk menyimpan dan mengelola data yang berhubungan dengan dua variabel, seperti data waktu dan harga saham, data iklim dari berbagai lokasi, atau data lain yang memiliki dimensi waktu dan ruang.

2. Penggunaan Array Dua Dimensi

Array dua dimensi digunakan dalam berbagai konteks dan aplikasi dalam pemrograman komputer. Berikut beberapa contoh penggunaan umum:

a) Merepresentasikan Matriks atau Tabel

Array dua dimensi digunakan untuk merepresentasikan matriks matematika atau tabel data. Ini sangat berguna dalam pemodelan matematika, analisis data, atau simulasi fisika.

```
int matriks[3][3]; // Matriks 3x3 dengan integer
```

Gambar 2.2 Matriks atau Tabel

b) Pengolahan Gambar

Dalam pengolahan gambar, citra sering diwakili sebagai array dua dimensi, dengan setiap elemen matriks mewakili nilai piksel pada lokasi tertentu dalam gambar.

```
int piksel[800][600]; // Citra dengan resolusi 800x600 piksel
```

Gamabar 2.3 Pengelolaan Gambar

c) Pengembangan Permainan

Dalam pengembangan permainan, array dua dimensi digunakan untuk merepresentasikan peta permainan, papan permainan, dan lokasi objek dalam permainan.

```
char peta[10][10]; // Peta permainan dengan ukuran 10x10
```

Gambar 2.4 Pengembangan Permainan

d) Analisis Data

Dalam analisis data, array dua dimensi digunakan untuk menyimpan data yang berkaitan dengan dua variabel, seperti data waktu dan harga saham.

```
float data_harga[365][2]; // Data harga saham untuk satu tahun
```

Gambar 2.5 Analisa Data

3. Operasi pada Array Dua Dimensi

Ketika bekerja dengan array dua dimensi, ada beberapa operasi umum yang dapat Anda lakukan:

e) Inisialisasi

Anda dapat menginisialisasi array dua dimensi dengan nilai awal saat Anda mendeklarasikannya, atau Anda dapat mengisi array setelahnya.

```
int matriks[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
// Inisialisasi saat mendeklarasikan
```

Gambar 2.6 Inisialisasi

f) Mengakses Elemen

Anda dapat mengakses elemen-elemen array dua dimensi dengan menggunakan dua indeks, satu untuk baris dan satu lagi untuk kolom.

```
int elemen = matriks[1][2];  
// Mengakses elemen pada baris ke-1 dan kolom ke-2
```

Gambar 2.7 Mengakses Elemen

g) Mengubah Nilai Elemen.

Anda dapat mengubah nilai elemen array dua dimensi dengan mengassign nilai baru ke indeks yang sesuai.

```
matriks[0][1] = 10;  
// Mengubah nilai elemen pada baris ke-0 dan kolom ke-1 menjadi 10
```

Gambar 2.8 Mengubah Nilai Elemen

h) Iterasi Melalui Array

Anda dapat menggunakan loop bersarang, seperti for dalam for, untuk mengiterasi melalui semua elemen dalam array dua dimensi.

```
titled1  
for (int i = 0; i < 3; i++) {  
    for (int j = 0; j < 3; j++) {  
        printf("Elemen pada baris %d dan kolom %d: %d\n", i, j, matriks[i][j]);  
    }  
}
```

Gamabar 2.9 iterasi Array

i) Menghitung Panjang Baris dan Kolom

Untuk mengetahui panjang baris dan kolom dari array dua dimensi, Anda dapat menggunakan operator sizeof atau menghitung panjang saat mendeklarasikan array.

```
int panjang_baris = sizeof(matriks) / sizeof(matriks[0]); // Menghitung panjang baris  
int panjang_kolom = sizeof(matriks[0]) / sizeof(matriks[0][0]); // Menghitung panjang kolom
```

Gamabar 2.10 Menghitung Panjang Dan Kolam

4. Kendala dan Pertimbangan

Meskipun array dua dimensi adalah alat yang sangat berguna dalam pemrograman, ada beberapa kendala dan pertimbangan yang perlu diperhatikan:

a) Ukuran Tetap

Panjang baris dan kolom array dua dimensi harus ditentukan saat mendeklarasikannya, dan itu tidak dapat diubah selama runtime. Jadi, Anda harus memilih ukuran yang sesuai dengan kebutuhan Anda.

b) Pengelolaan Memori

Array dua dimensi menggunakan sejumlah memori yang tetap tergantung pada jumlah baris dan kolomnya. Ini bisa menjadi masalah jika Anda bekerja dengan data yang sangat besar atau jika Anda perlu menghemat memori.

c) Indeks Out-of-Bounds

Seperti dalam array satu dimensi, jika Anda mencoba mengakses elemen di luar rentang indeks yang valid, Anda dapat menyebabkan kesalahan atau perilaku yang tidak terduga dalam program Anda. Oleh karena itu, Anda harus selalu memeriksa batasan indeks saat mengakses elemen array dua dimensi.

5. Kesimpulan Array 2 Dimensi

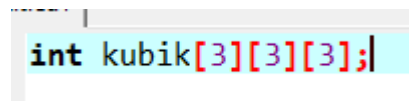
Array dua dimensi adalah salah satu konsep paling penting dalam pemrograman komputer. Mereka digunakan untuk merepresentasikan data dua dimensi, seperti matriks atau tabel, dan menyediakan cara yang terstruktur dan efisien untuk mengakses dan mengelola data. Array ini digunakan dalam berbagai konteks, termasuk pengolahan gambar, pengembangan permainan, analisis data, dan pemodelan matematika. Dengan memahami konsep array

dua dimensi, Anda dapat memperluas kemampuan pemrograman Anda dan membuat kode yang lebih efisien dan terstruktur.

C. Pengenalan ke Array Tiga Dimensi

Array tiga dimensi adalah salah satu konsep yang lebih kompleks dalam pemrograman komputer. Ini adalah jenis array yang digunakan untuk menyimpan data dalam tiga dimensi atau sebagai kubus tiga dimensi. Array ini memiliki tiga dimensi: baris, kolom, dan kedalaman. Dalam hal ini, setiap elemen dalam array diidentifikasi oleh tiga indeks: satu untuk baris, satu lagi untuk kolom, dan satu lagi untuk kedalaman.

Misalnya, pertimbangkan array tiga dimensi berikut dalam bahasa C:



```
int kubik[3][3][3];
```

Gambar 3.1 Array 3 dimensi

Array ini memiliki nama kubik dan memiliki tiga baris, tiga kolom, dan tiga lapisan. Ini berarti ada total 27 elemen dalam array. Elemen-elemen ini dapat diakses menggunakan tiga indeks, seperti `kubik[0][0][0]`, `kubik[0][0][1]`, `kubik[0][0][2]`, dan seterusnya.

Array tiga dimensi adalah alat yang sangat kuat dan fleksibel yang digunakan dalam berbagai aplikasi, termasuk pemodelan 3D, pengolahan citra tiga dimensi, pemodelan data tiga dimensi dalam ilmu fisika, kimia, dan rekayasa, serta dalam pengembangan permainan dan grafika komputer.

1. Mengapa Array Tiga Dimensi Penting

Array tiga dimensi memiliki peran yang sangat penting dalam pemrograman komputer untuk alasan berikut:

1. Merepresentasikan Data Tiga Dimensi

Array tiga dimensi adalah alat yang kuat untuk merepresentasikan data yang memiliki struktur tiga dimensi, seperti objek 3D, volume data medis, atau citra tiga dimensi.

2. Pengaksesan Data yang Terstruktur

Dengan tiga indeks, Anda dapat mengakses data dengan cara yang sangat terstruktur. Anda dapat dengan mudah mengambil elemen pada baris, kolom, dan kedalaman yang tepat dalam array.

3. Pemrosesan Data Tiga Dimensi

Array tiga dimensi memungkinkan Anda untuk melakukan operasi yang melibatkan data tiga dimensi dengan mudah. Anda dapat melakukan transformasi 3D, analisis volume data, atau berbagai operasi matematika lainnya pada elemen-elemen dalam array.

4. Pemodelan 3D

Array tiga dimensi digunakan untuk memodelkan objek 3D dalam pemrograman grafis dan pengembangan permainan. Mereka digunakan untuk merepresentasikan bentuk objek, tekstur, dan posisi dalam dunia virtual.

5. Pengolahan Citra 3D

Dalam pengolahan citra 3D, array tiga dimensi digunakan untuk merepresentasikan citra medis, citra pencitraan satelit, atau citra tiga dimensi lainnya. Mereka memungkinkan analisis dan manipulasi data citra yang kompleks.

2. Penggunaan Array Tiga Dimensi

Array tiga dimensi digunakan dalam berbagai konteks dan aplikasi dalam pemrograman komputer. Berikut beberapa contoh penggunaan umum:

1. Pemodelan 3D

Array tiga dimensi digunakan untuk memodelkan objek 3D dalam pemrograman grafis dan pengembangan permainan. Mereka digunakan untuk merepresentasikan bentuk objek, tekstur, dan posisi dalam dunia virtual.

```
float objek_3d[100][100][100]; // Objek 3D dengan volume 100x100x100
```

Gambar 3.2 Pemodelan 3D

2. Pengolahan Citra 3D

Dalam pengolahan citra 3D, array tiga dimensi digunakan untuk merepresentasikan citra medis, citra pencitraan satelit, atau citra tiga dimensi lainnya. Mereka memungkinkan analisis dan manipulasi data citra yang kompleks.

```
int citra_medis[256][256][256];  
// Citra medis 3D dengan resolusi 256x256x256
```

Gambar 3.3 Pengolah Citra 3D

3. Pemodelan Data Fisika

Array tiga dimensi digunakan dalam ilmu fisika dan rekayasa untuk memodelkan data tiga dimensi, seperti medan magnet, distribusi suhu, atau aliran fluida dalam suatu sistem.

```
double medan_magnet[50][50][50];  
// Medan magnet dalam sistem 3D dengan ukuran 50x50x50
```

Gambar 3.4 Pemodelan data Fisika

4. Analisis Volume Data

Dalam analisis volume data, seperti pemrosesan data medis atau simulasi numerik, array tiga dimensi digunakan untuk menyimpan dan mengelola volume data yang besar.

```
float volume_data[128][128][128];  
// Volume data numerik dengan resolusi 128x128x128
```

Gambar 3.5 Analisa Volume Data

3. Operasi pada Array Tiga Dimensi

Ketika bekerja dengan array tiga dimensi, ada beberapa operasi umum yang dapat Anda lakukan:

1. Inisialisasi

Anda dapat menginisialisasi array tiga dimensi dengan nilai awal saat Anda mendeklarasikannya, atau Anda dapat mengisi array setelahnya.

```
int kubik[3][3][3] = {{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}},
                      {{10, 11, 12}, {13, 14, 15}, {16, 17, 18}},
                      {{19, 20, 21}, {22, 23, 24}, {25, 26, 27}}};
// Inisialisasi saat mendeklarasikan
```

Gambar 3.6 Inisialisasi

2. Mengakses Elemen

Anda dapat mengakses elemen-elemen array tiga dimensi dengan menggunakan tiga indeks, satu untuk baris, satu lagi untuk kolom, dan satu lagi untuk kedalaman.

```
int elemen = kubik[1][2][0];
// Mengakses elemen pada baris ke-1, kolom ke-2, dan kedalaman ke-0
```

Gambar 3.7 Mengakses Elemen

3. Mengubah Nilai Elemen

Anda dapat mengubah nilai elemen array tiga dimensi dengan mengassign nilai baru ke indeks yang sesuai.

```
kubik[0][1][2] = 42;
// Mengubah nilai elemen pada baris ke-0, kolom ke-1, dan kedalaman ke-2 menjadi 42
```

Gambar 3.8 Mengubah Nilai Elemen

4. Iterasi Melalui Array

Anda dapat menggunakan tiga loop bersarang, seperti for dalam for dalam for, untuk mengiterasi melalui semua elemen dalam array tiga dimensi.

```

for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        for (int k = 0; k < 3; k++) {
            printf("Elemen pada baris %d, kolom %d, dan kedalaman %d: %d\n", i, j, k, kubik[i][j][k]);
        }
    }
}

```

Gambar 3.8 Iterasi melalui Array

5. Menghitung Panjang Baris, Kolom, dan Kedalaman

Untuk mengetahui panjang baris, kolom, dan kedalaman dari array tiga dimensi, Anda dapat menggunakan operator `sizeof` atau menghitung panjang saat mendeklarasikan array.

```

int panjang_baris = sizeof(kubik) / sizeof(kubik[0]); // Menghitung panjang baris
int panjang_kolom = sizeof(kubik[0]) / sizeof(kubik[0][0]); // Menghitung panjang kolom
int panjang_kedalaman = sizeof(kubik[0][0]) / sizeof(kubik[0][0][0]); // Menghitung panjang kedalaman

```

Gamabar 3.9 Menghitung Panjang, Baris, dan Kedalaman

4. Kendala dan Pertimbangan.

Meskipun array tiga dimensi adalah alat yang sangat kuat dalam pemrograman, ada beberapa kendala dan pertimbangan yang perlu diperhatikan:

1. Kompleksitas dan Ukuran Memori

Array tiga dimensi dapat menjadi sangat besar dan kompleks, terutama jika digunakan untuk menyimpan data tiga dimensi yang besar. Ini memerlukan alokasi memori yang signifikan dan dapat mempengaruhi kinerja program Anda.

2. Pengelolaan Memori

Karena ukuran memori yang lebih besar, Anda perlu mempertimbangkan pengelolaan memori dengan cermat. Ini termasuk alokasi memori dinamis, penghapusan, dan pembebasan memori ketika tidak lagi diperlukan.

3. Kompleksitas Kode

Ketika bekerja dengan array tiga dimensi, kode Anda dapat menjadi lebih kompleks karena Anda perlu mengatur tiga tingkat indeks. Ini dapat membuat kode sulit dibaca dan dipahami jika tidak diorganisir dengan baik.

4. Pengujian dan Validasi

Pengujian dan validasi menjadi lebih penting saat Anda bekerja dengan array tiga dimensi karena ada lebih banyak kemungkinan interaksi antara elemen-elemen dalam array.

5. Kesimpulan Array 3 Dimensi

Array tiga dimensi adalah salah satu konsep yang kuat dalam pemrograman komputer. Mereka digunakan untuk merepresentasikan data tiga dimensi, seperti objek 3D, citra medis, atau volume data, dan menyediakan cara yang terstruktur dan efisien untuk mengakses dan mengelola data. Array ini digunakan dalam berbagai aplikasi, termasuk pemodelan 3D, pengolahan citra tiga dimensi, pemodelan data tiga dimensi dalam ilmu fisika, kimia, dan rekayasa, serta dalam pengembangan permainan dan grafika komputer. Dengan memahami konsep array tiga dimensi, Anda dapat memperluas kemampuan pemrograman Anda dan membuat kode yang lebih efisien dan terstruktur.

D. PROGRAM ARRAY

1. Program Array 1 Dimensi

```
1  #include <stdio.h>
2
3  int main() {
4      // Deklarasi dan inisialisasi array satu
5      int angka[5] = {10, 25, 7, 42, 5};
6
7      // Menampilkan isi array
8      printf("Isi array: ");
9      for (int i = 0; i < 5; i++) {
10         printf("%d ", angka[i]);
11     }
12     printf("\n");
13
14     // Menghitung nilai maksimum dan minimum
15     int maksimum = angka[0];
16     int minimum = angka[0];
17
18     for (int i = 1; i < 5; i++) {
19         if (angka[i] > maksimum) {
20             maksimum = angka[i];
21         }
22         if (angka[i] < minimum) {
23             minimum = angka[i];
24         }
25     }
26
27     // Menghitung rata-rata
28     int total = 0;
29     for (int i = 0; i < 5; i++) {
30         total += angka[i];
31     }
32     double rataRata = (double)total / 5;
33
34     // Menampilkan hasil
35     printf("Nilai maksimum: %d\n", maksimum);
36     printf("Nilai minimum: %d\n", minimum);
37     printf("Rata-rata: %.2lf\n", rataRata);
38
39     // Mencari elemen tertentu dalam array
40     int cari = 42;
41     int ditemukan = 0; // 0 menandakan elemen tidak ditemukan
42
43     for (int i = 0; i < 5; i++) {
44         if (angka[i] == cari) {
45             ditemukan = 1; // 1 menandakan elemen ditemukan
46             break; // Keluar dari loop jika elemen ditemukan
47         }
48     }
49
50     if (ditemukan) {
51         printf("%d ditemukan dalam array.\n", cari);
52     } else {
53         printf("%d tidak ditemukan dalam array.\n", cari);
54     }
55
56     return 0;
57 }
```

Gambar 5.1 Program Arraay 1 Dimensi

Analisis Program:

1. Program di atas menggunakan array satu dimensi (angka) untuk menyimpan lima bilangan bulat.
2. Program menampilkan isi array menggunakan loop for dan mencetak setiap elemen array.
3. Kemudian, program mencari nilai maksimum dan minimum dalam array. Ini dilakukan dengan menggunakan dua variabel (maksimum dan minimum) yang awalnya diatur ke elemen pertama dalam array. Program kemudian membandingkan setiap elemen array dengan nilai maksimum dan minimum, dan jika ditemukan elemen yang lebih besar atau lebih kecil, nilai maksimum dan minimum diperbarui.
4. Program menghitung rata-rata elemen dalam array dengan menjumlahkan semua elemen dan membaginya dengan jumlah elemen. Hasilnya disimpan dalam variabel rataRata.
5. Hasil operasi (nilai maksimum, minimum, dan rata-rata) kemudian dicetak ke layar.
6. Selanjutnya, program mencari elemen tertentu dalam array (dalam contoh ini, mencari angka 42). Ini dilakukan dengan menggunakan loop for yang mencocokkan setiap elemen dengan angka yang dicari. Jika elemen ditemukan, variabel ditemukan diatur ke 1, jika tidak, diatur ke 0. Ini digunakan untuk menampilkan pesan apakah elemen tersebut ditemukan atau tidak.

Contoh Eksekusi Program:

Misalnya, jika array angka berisi {10, 25, 7, 42, 5}, maka program akan menghasilkan output berikut:

```
Isi array: 10 25 7 42 5
Nilai maksimum: 42
Nilai minimum: 5
Rata-rata: 17.80
42 ditemukan dalam array.
```

Kesimpulan:

Program di atas adalah contoh sederhana penggunaan array satu dimensi dalam bahasa C. Program ini menunjukkan cara mendeklarasikan, mengisi, dan melakukan operasi dasar seperti mencari nilai maksimum, minimum, rata-rata, dan mencari elemen tertentu dalam array. Penggunaan array memungkinkan kita untuk mengorganisasi dan mengelola kumpulan data dengan lebih efisien dalam pemrograman.

2. Program Array 2 Dimensi

```
1  #include <stdio.h>
2
3  int main() {
4      // Deklarasi array dua dimensi untuk menyimpan matriks
5      const int baris = 3;    // Jumlah baris matriks
6      const int kolom = 3;    // Jumlah kolom matriks
7      int matriks[baris][kolom];
8
9      // Input elemen-elemen matriks dari pengguna
10     printf("Masukkan elemen-elemen matriks %d x %d:\n", baris, kolom);
11     for (int i = 0; i < baris; i++) {
12         for (int j = 0; j < kolom; j++) {
13             printf("Baris %d, Kolom %d: ", i + 1, j + 1);
14             scanf("%d", &matriks[i][j]);
15         }
16     }
17
18     // Menampilkan matriks
19     printf("Matriks yang dimasukkan:\n");
20     for (int i = 0; i < baris; i++) {
21         for (int j = 0; j < kolom; j++) {
22             printf("%d\t", matriks[i][j]);
23         }
24         printf("\n");
25     }
26
27     // Menghitung total elemen matriks
28     int total = 0;
29     for (int i = 0; i < baris; i++) {
30         for (int j = 0; j < kolom; j++) {
31             total += matriks[i][j];
32         }
33     }
34
35     // Menampilkan total elemen matriks
36     printf("Total elemen matriks: %d\n", total);
37
38     return 0;
39 }
```

Gamabar 5.2 Program Array 2 Dimensi

Contoh Eksekusi Program:

Misalnya, jika pengguna memasukkan elemen-elemen matriks berikut:

```
Baris 1, Kolom 1: 2
Baris 1, Kolom 2: 4
Baris 1, Kolom 3: 6
Baris 2, Kolom 1: 1
Baris 2, Kolom 2: 3
Baris 2, Kolom 3: 5
Baris 3, Kolom 1: 7
Baris 3, Kolom 2: 8
Baris 3, Kolom 3: 9
```

Maka program akan menghasilkan output berikut:

```

Matriks yang dimasukkan:
2       4       6
1       3       5
7       8       9
Total elemen matriks: 45
-----

```

Kesimpulan:

Program di atas adalah contoh penggunaan array dua dimensi dalam bahasa C untuk menyimpan dan mengelola data matriks. Program ini memungkinkan pengguna untuk memasukkan elemen-elemen matriks, menampilkan matriks, menghitung total elemen matriks, dan menampilkan hasilnya. Ini adalah salah satu contoh sederhana penggunaan array dua dimensi dalam pemrograman C.

3. Program Array 3 dimensi

```

1  #include <stdio.h>
2
3  int main() {
4      // Deklarasi dan inisialisasi array tiga dimensi
5      const int panjang = 3;
6      const int lebar = 3;
7      const int tinggi = 3;
8      int kubus[panjang][lebar][tinggi];
9
10     // Mengisi array dengan data volume kubus
11     for (int i = 0; i < panjang; i++) {
12         for (int j = 0; j < lebar; j++) {
13             for (int k = 0; k < tinggi; k++) {
14                 kubus[i][j][k] = (i + 1) * (j + 1) * (k + 1); // Volume kubus
15             }
16         }
17     }
18
19     // Menghitung volume total
20     int totalVolume = 0;
21     for (int i = 0; i < panjang; i++) {
22         for (int j = 0; j < lebar; j++) {
23             for (int k = 0; k < tinggi; k++) {
24                 totalVolume += kubus[i][j][k];
25             }
26         }
27     }
28
29     // Menampilkan hasil
30     printf("Data Volume Kubus:\n");
31     for (int i = 0; i < panjang; i++) {
32         for (int j = 0; j < lebar; j++) {
33             for (int k = 0; k < tinggi; k++) {
34                 printf("Kubus[%d][%d][%d] = %d\n", i, j, k, kubus[i][j][k]);
35             }
36         }
37     }
38     printf("Total Volume Kubus: %d\n", totalVolume);
39
40     return 0;
41 }

```

Gamabar 5.3 Program Array 3 Dimensi

Analisis Program:

1. Program di atas menggunakan array tiga dimensi (kubus) untuk menyimpan volume kubus dalam tiga dimensi (panjang, lebar, dan tinggi).
2. Program mengisi array dengan data volume kubus menggunakan tiga loop bersarang (loop untuk panjang, lebar, dan tinggi). Volume kubus dihitung sebagai hasil perkalian panjang, lebar, dan tinggi.
3. Setelah semua data volume kubus dimasukkan ke dalam array, program menghitung volume total dengan menggunakan tiga loop bersarang yang lain.
4. Hasil operasi (data volume kubus dan total volume) kemudian dicetak ke layar.

Contoh Eksekusi Program:

Hasil dari program ini adalah data volume kubus dan total volume yang dihasilkan oleh volume kubus yang diisi dalam array tiga dimensi.

```
Kubus[0][0][2] = 3
Kubus[0][1][0] = 2
Kubus[0][1][1] = 4
Kubus[0][1][2] = 6
Kubus[0][2][0] = 3
Kubus[0][2][1] = 6
Kubus[0][2][2] = 9
Kubus[1][0][0] = 2
Kubus[1][0][1] = 4
Kubus[1][0][2] = 6
Kubus[1][1][0] = 4
Kubus[1][1][1] = 8
Kubus[1][1][2] = 12
Kubus[1][2][0] = 6
Kubus[1][2][1] = 12
Kubus[1][2][2] = 18
Kubus[2][0][0] = 3
Kubus[2][0][1] = 6
Kubus[2][0][2] = 9
Kubus[2][1][0] = 6
Kubus[2][1][1] = 12
Kubus[2][1][2] = 18
Kubus[2][2][0] = 9
Kubus[2][2][1] = 18
Kubus[2][2][2] = 27
Total Volume Kubus: 216
```

Kesimpulan:

Program di atas adalah contoh penggunaan array tiga dimensi dalam bahasa C. Dalam program ini, array tiga dimensi digunakan untuk menyimpan data volume kubus dalam tiga dimensi. Program kemudian menghitung total volume dari semua kubus yang disimpan dalam array. Ini adalah salah satu contoh penggunaan array.

BAB II

PENUTUP

KESIMPULAN

Kesimpulan mengenai array satu, dua, dan tiga dimensi adalah sebagai berikut:

1. Array Satu Dimensi:

- Array satu dimensi adalah struktur data yang digunakan untuk menyimpan sejumlah elemen dengan tipe data yang sama.
- Setiap elemen dalam array dapat diakses menggunakan indeks yang dimulai dari 0.
- Array satu dimensi berguna untuk menyimpan data berurutan seperti daftar angka, nama, atau hasil pengukuran.
- Operasi umum pada array satu dimensi meliputi inisialisasi, pengisian, pencarian, pengurutan, dan penghitungan statistik seperti rata-rata, maksimum, dan minimum.

2. Array Dua Dimensi:

- Array dua dimensi adalah struktur data dengan dua indeks yang digunakan untuk menyimpan data dalam bentuk matriks (baris dan kolom).
- Setiap elemen dalam array dua dimensi memiliki dua indeks: baris dan kolom.
- Array dua dimensi sering digunakan untuk representasi data tabel atau grid, seperti data spreadsheet atau citra piksel.
- Operasi umum pada array dua dimensi meliputi inisialisasi, pengisian, pencarian, manipulasi elemen, dan operasi matriks seperti perkalian dan penjumlahan matriks.

3. Array Tiga Dimensi:

- Array tiga dimensi adalah struktur data dengan tiga indeks yang digunakan untuk menyimpan data dalam tiga dimensi (panjang, lebar, dan kedalaman).
- Setiap elemen dalam array tiga dimensi memiliki tiga indeks yang mewakili koordinat dalam ruang tiga dimensi.
- Array tiga dimensi digunakan untuk merepresentasikan data tiga dimensi seperti volume objek 3D, citra medis 3D, atau model data kompleks.
- Operasi umum pada array tiga dimensi meliputi inisialisasi, pengisian, pencarian, dan manipulasi elemen dalam ruang tiga dimensi.

Kesimpulannya, array dengan dimensi yang berbeda-beda digunakan untuk menyimpan dan mengorganisasi data dalam pemrograman. Masing-masing dimensi array menyediakan tingkat kompleksitas yang berbeda dalam merepresentasikan data. Pemahaman tentang penggunaan array satu, dua, atau tiga dimensi sangat penting dalam pemrograman karena memungkinkan pengelolaan dan akses data yang efisien dan terstruktur sesuai dengan kebutuhan proyek.

DAFTAR PUSTAKA

<https://algorit.ma/blog/array-adalah-2022/>
<https://itbox.id/blog/array-adalah/>
<https://course-net.com/blog/array-adalah/>
<https://fikti.umsu.ac.id/struktur-data-array-pengertian-fungsi-dan-contoh-program/>
<https://codingstudio.id/blog/fungsi-array-pengertian-jenis/>
https://staffnew.uny.ac.id/upload/132310817/penelitian/Bab8.Array_.pdf