

Array Pada Pemrograman C++

Pengertian Array

Array adalah tumpukan variabel yang sejenis dan memiliki nama yang sama. Konsep suatu array menjadikan tumpukan variabel tersebut mampu menampung nilai yang berbeda asalkan tetap dengan tipe yang sama dengan tipe array. Sintaks penulisan array adalah sebagai berikut:

```
1 tipe nama_array[elemen];
```

Tipe array merupakan tipe data tumpukan variabel tersebut, sedangkan nama_array adalah nama tumpukan variabel. Untuk elemen atau indeks, merupakan pembeda yang membedakan tumpukan variabel yang satu dengan yang lainnya dalam satu array, karena di awal sudah Mimin jelaskan bahwa array memiliki nama dan jenis yang sama namun pembendanya terletak di indeks atau elemen itu sendiri.

Pendeklarasian Array

Berikut mimin contohkan cara mendeklarasikan suatu array:

```
1 int tampungan[5];
```

Pada contoh diatas, Mimin membuat array dengan nama tampungan bertipe data integer dimana array ini dapat menampung nilai sebanyak 5 buah. Masing-masing nilai yang dimasukkan akan diberikan nomor elemen atau indeks dari elemen '0' hingga '4'. Yang perlu kamu ingat bahwa, penomoran elemen pada array dimulai dari angka '0'.

Pemberian Nilai Pada Array

Untuk memberikan nilai pada suatu array, kamu dapat memasukkan nilai satu persatu pada masing-masing elemen atau dengan memasukkan dengan cara bersamaan. Untuk memasukkan satu-persatu pada masing-masing elemen kamu dapat lakukan dengan kode program sebagai berikut:

```
1
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     int tampungan[5];
7     tampungan[0] = 4;
8     tampungan[1] = 3;
9     tampungan[2] = 7;
10    tampungan[3] = 8;
11    tampungan[4] = 9;
12    return 0;
13 }
```

Untuk memasukkan nilai secara serentak dapat kamu coba dengan kode di bawah ini:

```

1  #include <iostream>
2  using namespace std;
3
4  int main(){
5  int tampungan[5] = {4,3,7,8,9};
6  return 0;
7  }

```

Kedua contoh program diatas memiliki fungsi dan tujuan yang sama yaitu memasukkan kelima nilai di atas ke dalam array “tampungan”, hanya saja dengan cara yang berbeda.

Mengakses Nilai Pada Array

Setelah kamu bisa mendeklarasikan dan memberikan nilai pada suatu array, maka langkah selanjutnya adalah mengakses nilai yang berada pada suatu array. Untuk programnya Mimin contohkan sebagai berikut:

```

1
2  #include <iostream>
3  using namespace std;
4
5  int main(){
6  int tampungan[5] = {4,3,7,8,9};
7  cout<<"Nilai pada array tampungan elemen ke "<<0<<" adalah :"<<tampungan[0]<<endl;
8  cout<<"Nilai pada array tampungan elemen ke "<<1<<" adalah :"<<tampungan[1]<<endl;
9  cout<<"Nilai pada array tampungan elemen ke "<<2<<" adalah :"<<tampungan[2]<<endl;
10 cout<<"Nilai pada array tampungan elemen ke "<<3<<" adalah :"<<tampungan[3]<<endl;
11 cout<<"Nilai pada array tampungan elemen ke "<<4<<" adalah :"<<tampungan[4]<<endl;
12 return 0;
13 }

```

Berikut adalah hasil output programnya:

```

Nilai pada array tampungan elemen ke 0 adalah :4
Nilai pada array tampungan elemen ke 1 adalah :3
Nilai pada array tampungan elemen ke 2 adalah :7
Nilai pada array tampungan elemen ke 3 adalah :8
Nilai pada array tampungan elemen ke 4 adalah :9
Process returned 0 (0x0) execution time : 0.015 s
Press any key to continue.

```

Kamu juga bisa memanfaatkan loop untuk mengakses nilai pada array karena lebih cepat dan simple. Berikut Mimin contohkan kode programnya:

```

1  #include <iostream>
2  using namespace std;
3
4  int main(){
5  int tampungan[5] = {4,3,7,8,9};
6  for(int i=0; i<=5; i++){
7  cout<<"Nilai pada array tampungan elemen ke "<<i<<" adalah :"<<tampungan[i]<<endl;
8  }

```

```

8     return 0;
9     }
10

```

Terlihat bahwa program untuk mengakses array dengan memanfaatkan loop lebih simple untuk menghasilkan output yang sama sebagai berikut:

```

Nilai pada array tampungan elemen ke 0 adalah :4
Nilai pada array tampungan elemen ke 1 adalah :3
Nilai pada array tampungan elemen ke 2 adalah :7
Nilai pada array tampungan elemen ke 3 adalah :8
Nilai pada array tampungan elemen ke 4 adalah :9
Nilai pada array tampungan elemen ke 5 adalah :5

Process returned 0 (0x0)   execution time : 0.020 s
Press any key to continue.

```

Array Multidimensi

Array multidimensi merupakan array yang memiliki indeks lebih dari 1. Indeks pada array merepresentasikan dimensi dari array. Sebagai contoh array dengan indeks berjumlah dua, maka dapat dikatakan bahwa array tersebut merupakan array dua dimensi dan begitu juga untuk dimensi yang lainnya. Pada contoh kasus ini, Mimin mencontohkan array dua dimensi dalam C++ sebagai berikut:

```

1     int arrayMulti[2][3];

```

Mimin sengaja mencontohkan array dua dimensi pada kasus ini karena array ini dapat dipahami sebagai sebuah sel dalam tabel. Dimensi pertama array adalah *baris*, sedangkan dimensi array kedua adalah *kolom*. Visualisasi array dua dimensi, Mimin gambarkan sebagai berikut:

	0	1	2
0	int arrayMulti[0][0]	int arrayMulti[0][1]	int arrayMulti[0][2]
1	int arrayMulti[1][0]	int arrayMulti[1][1]	int arrayMulti[1][2]

Dari tabel diatas kamu dapat melihat bahwa, jumlah elemen dari arrayMulti adalah 6, yang didapat dari perkalian indeks pertama dan kedua $2 \times 3 = 6$. Untuk memberikan nilai pada array dua dimensi kamu dapat menggunakan sintaks sebagai berikut:

```

1     int arrayMulti[2][3] = {{1,2,3},{4,5,6}};

```

Pada contoh diatas, dimensi pertama adalah abstrak atau bisa dikatakan hanya sebagai kelompok-kelompok bilangan (kurung kurawal ({})) didalam kurung kurawal utama atau besar). Dimensi kedua merupakan anggota dari kelompok-kelompok tersebut (isi dari kurung kurawal). Secara mudahnya bahwa dimensi kedua merupakan isi dari dimensi pertama. Berikut mimin contohkan aplikasinya dalam program C++:

```

1     #include <iostream>
        using namespace std;

```

```

2
3     int main(){
4     int arrayMulti[2][3] = {{1,2,3},{4,5,6}};
5     cout<<"Nilai arrayMulti[0][0] adalah "<<arrayMulti[0][0]<<endl;
6     cout<<"Nilai arrayMulti[0][0] adalah "<<arrayMulti[0][1]<<endl;
7     cout<<"Nilai arrayMulti[0][0] adalah "<<arrayMulti[0][2]<<endl;
8     cout<<"Nilai arrayMulti[0][0] adalah "<<arrayMulti[1][0]<<endl;
9     cout<<"Nilai arrayMulti[0][0] adalah "<<arrayMulti[1][1]<<endl;
10    cout<<"Nilai arrayMulti[0][0] adalah "<<arrayMulti[1][2]<<endl;
11    return 0;
12    }
13

```

Hasil output programnya adalah sebagai berikut:

```

Nilai arrayMulti[0][0] adalah 1
Nilai arrayMulti[0][0] adalah 2
Nilai arrayMulti[0][0] adalah 3
Nilai arrayMulti[0][0] adalah 4
Nilai arrayMulti[0][0] adalah 5
Nilai arrayMulti[0][0] adalah 6

Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.

```

Kamu juga bisa menggunakan looping for untuk mengakses nilai dari array multidimensi guna mempersingkat pemrograman sebagai berikut:

```

1
2     #include <iostream>
3     using namespace std;
4
5     int main(){
6     int arrayMulti[2][3] = {{1,2,3},{4,5,6}};
7     cout<<"Akses Array Multidimensi Dengan Looping For"<<endl;
8     for(int x=0; x<=1; x++){
9         for(int y=0; y<=2; y++){
10            cout<<"Nilai arrayMulti["<<x<<"]["<<y<<"] adalah "<<arrayMulti[x][y]<<endl;
11        }
12    }
13    return 0;
14    }

```

Hasil output programnya adalah sebagai berikut:

```
Akses Array Multidimensi Dengan Looping For
Nilai arrayMulti[0][0] adalah 1
Nilai arrayMulti[0][1] adalah 2
Nilai arrayMulti[0][2] adalah 3
Nilai arrayMulti[1][0] adalah 4
Nilai arrayMulti[1][1] adalah 5
Nilai arrayMulti[1][2] adalah 6

Process returned 0 (0x0)   execution time : 0.024 s
Press any key to continue.
```

Array Sebagai Parameter Suatu Fungsi

Ternyata sobat, array dapat juga berperan sebagai parameter suatu fungsi. Untuk menjadikan array sebagai parameter fungsi cukup dengan menambahkan kurung kotak ([]) pada akhir nama parameter. Yang harus kamu perhatikan, bahwa parameter pada array tidak perlu penambahan indeks. Berikut Mimin contohkan sintaks sederhananya:

```
1 void cetakArray(int arrayInput[],int indeks);
```

Mimin hendak membuat fungsi dengan nama *cetakArray* yang bertindak untuk mencetak nilai pada array, dimana array ini diinputkan user melalui parameter kedalam fungsi. Nah untuk memudahkan pemahaman kamu, Mimin contohkan program sederhananya sebagai berikut:

```
1
2 #include <iostream>
3 using namespace std;
4
5 void cetakArray(int arrayInput[], int indeks);
6 int main(){
7     int arrayMimin[4] = {1,2,3,4};
8     cetakArray(arrayMimin,4);
9     return 0;
10 }
11
12 void cetakArray(int arrayInput[], int indeks){
13     for (int x=0; x<indeks; x++){
14         cout<<"Nilai array indeks "<<x<<" adalah = "<<arrayInput[x]<<endl;
15     }
16 }
```

Hasil output programnya adalah sebagai berikut:

```
Nilai array indeks 0 adalah = 1
Nilai array indeks 1 adalah = 2
Nilai array indeks 2 adalah = 3
Nilai array indeks 3 adalah = 4

Process returned 0 (0x0)   execution time : 0.022 s
Press any key to continue.
```

Array Sebagai Kumpulan Karakter

Sedikit mengulang mengenai *string*. *String* merupakan tumpukan karakter, untaian karakter atau disebut dengan *array of character*. Secara mudahnya, *string* merupakan kumpulan dari beberapa karakter yang disusun menurut aturan tertentu, sehingga menghasilkan suatu kata atau kalimat.

String yang terdiri dari kata “sinauArduino”, jika ditulis dalam sebuah array dalam C++ dapat dideklarasikan dengan menggunakan sintaks sebagai berikut:

```
1 char nama[13] = "sinauArduino";
```

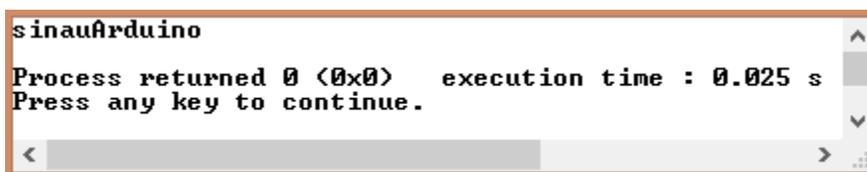
Berdasarkan elemen-elemen array yang sudah kamu pelajari, kamu dapat mengakses huruf “A” dari elemen ke-enam array tersebut dengan sintaks:

```
1 char nama[5];
```

Yang perlu kamu ingat adalah, bahwa *string* merupakan *null-terminated array of character*, dengan artian bahwa sebuah string selalu diakhiri dengan karakter null (`'\0'`). Dengan keistimewaan inilah kamu tidak perlu menghitung panjang string untuk batas akhir suatu looping, namun cukup apabila nilai elemen dalam suatu string sama dengan null (`'\0'`), maka bisa dipastikan bahwa elemen tersebut adalah akhir dari sebuah string. Untuk lebih memudahkan pemahamanmu, berikut Mimin contohkan program sederhananya:

```
1
2 #include <iostream>
3 using namespace std;
4
5 int main(){
6     char nama[13] = "sinauArduino";
7     for (int i=0; nama[i]!='\0'; i++){
8         cout<<nama[i];
9     }
10    cout<<endl;
11    return 0;
12 }
```

Hasil output programnya sebagai berikut:



```
sinauArduino
Process returned 0 (0x0) execution time : 0.025 s
Press any key to continue.
```

Program di atas berfungsi untuk mencetak semua nilai yang terdapat pada array karakter “nama” dengan bantuan looping for, dimana looping ini akan terus bekerja mencetak isi dalam setiap elemen array hingga ditemukan *null terminated character* `'\0'`.