

Modul 1. Array

[Jump to bottom](#)

berliana-amelia edited this page on Apr 4 · 11 revisions

Pembahasan Materi :

1. Definisi Array
2. Mengapa Menggunakan Array?
3. Kekurangan dan Kelebihan menggunakan array
4. Pendeklarasian Array
 - Array 1 Dimensi
 - Array 2 Dimensi
 - Array Multidimensi dan Array of String
5. Mengubah nilai dalam Array

Apa itu Array?

40	55	63	17	22	68	89	97	89
0	1	2	3	4	5	6	7	8

<- Array Indices

Array Length = 9

First Index = 0

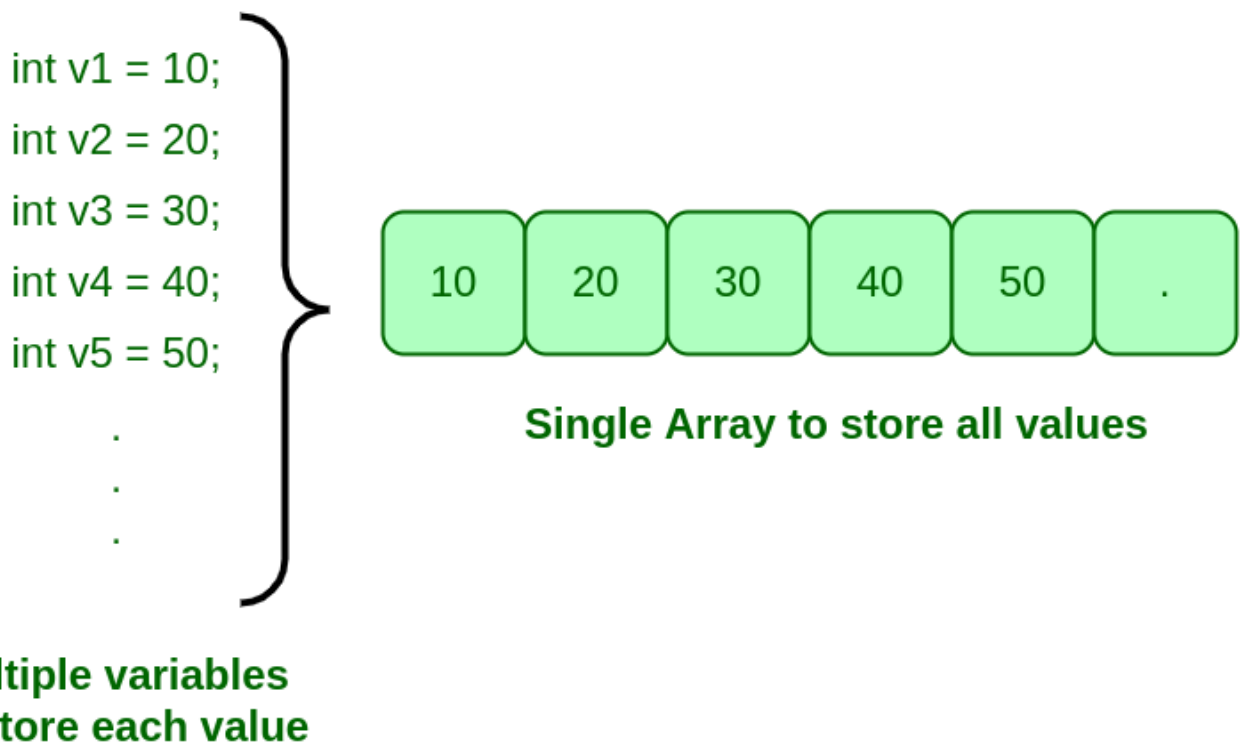
Last Index = 8

Array adalah variabel yang dapat menyimpan beberapa nilai. Array merupakan kumpulan item data serupa yang disimpan di lokasi memori yang berdekatan dan elemen dapat diakses secara acak menggunakan indeks array. Array dapat digunakan untuk menyimpan koleksi tipe data primitif seperti int, float, double, char, dll dari jenis tertentu. Array dapat dibayangkan seperti gedung yang mempunyai no ruangan masing-masing. Array biasanya digunakan untuk mengatasi penggunaan variabel banyak dengan tipe data yang sama. Misalnya, jika Anda ingin menyimpan 100 bilangan bulat, Anda dapat membuat array untuk itu.

Mengapa menggunakan Array?

Kita dapat menggunakan variabel normal (var1, var2, var3, ..) ketika kita menggunakan data/objek yang kecil, tetapi jika kita ingin menggunakan data dengan jumlah yang banyak akan menjadi sulit untuk mengelolanya dengan variabel normal. Ide array adalah untuk mewakili banyak contoh dalam satu variabel. Misal membuat satu array berisi nilai integer bilang 1 - 100 lebih mudah dikelola daripada mengelola var1 - var100.

Kelebihan dan Kekurangan Penggunaan Array



Kelebihan

1. Code Optimization: kita dapat mengambil atau mengurutkan data secara efisien.

Note : Pengambilan data pada index berapapun di dalam array hanya membutuhkan kompleksitas waktu $O(1)$.

2. Easy to Access: Kita bisa mendapatkan data apa pun yang terletak di posisi indeks.

Kekurangan

1. Size Limit : Kita hanya dapat menyimpan ukuran elemen tetap dalam array. Ukuran itu tidak bisa bertambah saat runtime*.

*Compile-time adalah waktu di mana kode diubah menjadi kode yang dapat dieksekusi sedangkan runtime adalah waktu di mana kode yang dapat dieksekusi mulai berjalan.

Mendeklarasikan Array

Array 1 Dimensi

Array satu dimensi adalah array yang mempunyai satu nilai indeks. Setiap indeks digunakan untuk mengakses isi array tersebut. Bentuk umum array satu dimensi sebagai berikut :

```
dataType arrayName[arraySize];
```

misalnya :

```
float iniArray[10];
```

keterangan :

1. dataType disesuaikan dengan isi array
2. arrayName sesuai aturan penamaan variabel
3. arraySize merupakan ukuran array harus bernilai integer
4. Indeks array dimulai dari 0

Namun ada beberapa cara lainnya untuk mendeklarasikan array :

```
#include <stdio.h>

int main(void)
{
    // 1. Deklarasi Array dengan menspesifikasikan ukuran array
    int arr1[10];
    arr1[0] = 1;
    arr1[1] = 2;
    arr1[2] = 3;
    // Deklarasi ukuran array dengan menggunakan variable
    int n = 10;
    int arr2[n];

    //2. Deklarasi Array dengan menginisialisasi elemen
    int arr3[] = { 10, 20, 30, 40};
    //Compiler akan membuat array berukuran 4, dengan cara ini ukuran dan elemen array telah ditentukan

    //3. Deklarasi array dengan menentukan ukuran dan menginisialisasi elemen
    int arr4[6] = { 10, 20, 30, 40 };
    //Compiler akan membuat array berukuran 6, dan akan mengisi indeks 0 - 3 dengan elemen yang telah ditentukan
```

```
//Index 4 - 5 akan di deklarasikan sebagai 0
// seperti "int arr[] = {10, 20, 30, 40, 0, 0}"

printf("Nilai arr4 pada %i", arr4[4]);
return (0);
}
```

Jangan lupa index array dimulai dari 0

Hasil dari kode di atas akan menampilkan hasil sebagai berikut :

Nilai arr4 pada index 4 adalah 0

Array 2 Dimensi

	Column 0	Column 1	Column 2
Row 0	x[0][0]	x[0][1]	x[0][2]
Row 1	x[1][0]	x[1][1]	x[1][2]
Row 2	x[2][0]	x[2][1]	x[2][2]

Array 2 dimensi adalah array yang memiliki dua nilai indeks. Bentuk umum untuk array dua dimensi adalah:

```
dataType arrayName[x][y];
```

Keterangan : dataType: Tipe data yang akan disimpan dalam array arrayName: nama Array x: Jumlah baris y: Jumlah kolom misalnya :

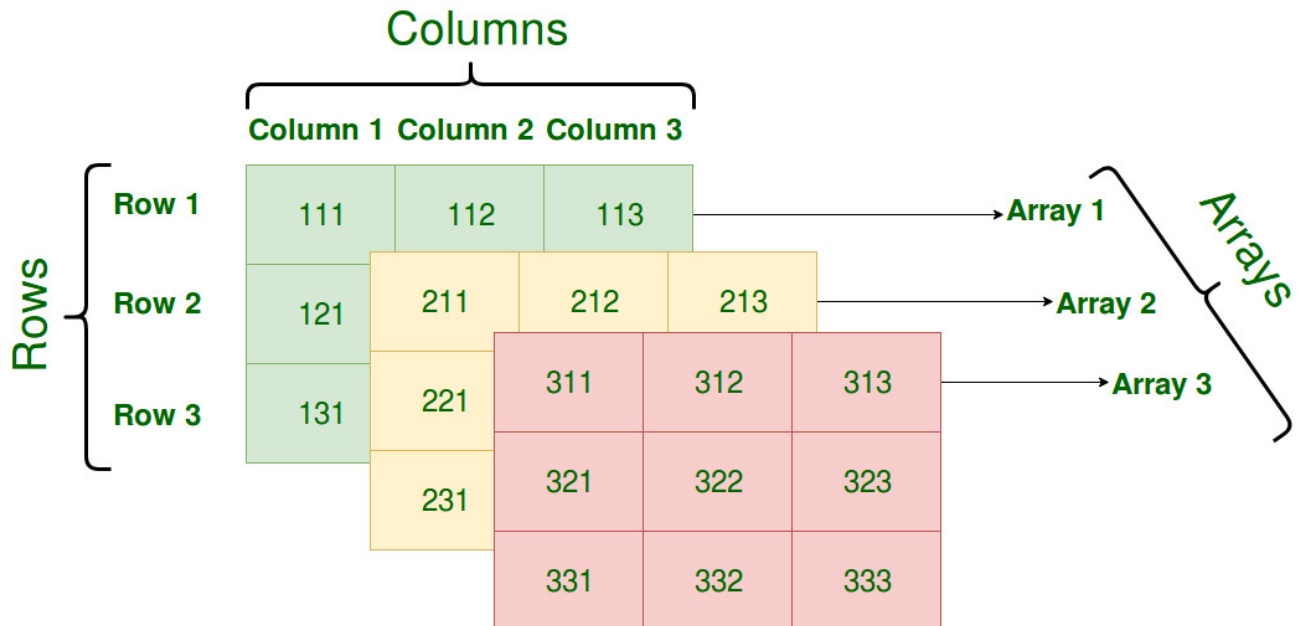
```
int iniArray[3][4] = {{0,1,2,3}, {4,5,6,7}, {8,9,10,11}};

printf("Nilai ini array index 2,1 adalah : %i", iniArray[2][1]);
```

Jika kode di atas di jalankan maka kita akan mendapatkan nilai sebagai berikut :

Nilai ini array index 2,1 adalah : 9

Array Multidimensi dan Array of String



Array multi-dimensi dapat disebut sebagai array yang menyimpan data homogen dalam bentuk tabel. Data dalam array multidimensi umumnya disimpan dalam urutan baris-mayor dalam memori. Bentuk umum array multidimensi adalah :

```
dataType arrayName[size1][size2]...[sizeN];
```

Keterangan : dataType: Tipe data yang akan disimpan dalam array arrayName: nama Array
size1,size2,...: Ukuran array setiap dimensi

Ukuran array multidimensi

Jumlah total elemen yang dapat disimpan dalam array multidimensi dapat dihitung dengan mengalikan ukuran semua dimensi. Misalnya:

1. Array int x[10][20] dapat menyimpan total $(10 * 20) = 200$ elemen.
2. array int x[5][10][20] dapat menyimpan total $(5 * 10 * 20) = 1000$ elemen.

Array of String

Dari awal kita menggunakan array dengan tipe data berupa angka numerik. Pada array of String sendiri kita dapat membuat sebuah array untuk character dengan cara seperti :

```
char arr[3][10] = {"Hallo", "Gojo", "Satoru"};
// karena di dalam bahasa C tidak terdapat tipe data String maka digunakan tipe data char
// dengan begitu batas character dalam suatu array adalah 10
for (int i = 0; i < 3; i++)
{
    printf("%s\n", arr[i]);
}
```

Apabila kita menjalankan kode di atas maka di dapatkan hasil :

```
Hallo
Gojo
Satoru
```

Mengubah nilai dalam array

1. Menambahkan nilai dengan perulangan

Biasanya kita menginisialisasi nilai dalam array satu persatu itu, namun ketika kita ingin memasukkan data dalam jumlah besar ke dalam array biasanya perulangan dapat digunakan untuk mengefisienkan kode kita. misal membuat array satu dimensi yang berisi nilai 1 - 100 maka dapat digunakan perulangan. Di bawah ini adalah contoh penggunaan perulangan untuk mengisi array 2 dimensi :

```
int x[3][4];

for(int i = 0; i < 3; i++){
    for(int j = 0; j < 4; j++){
        x[i][j] = i + j;
    }
}

for(int i = 0; i < 3; i++){
    for(int j = 0; j < 4; j++){
        printf("Nilai array x[%i][%i] adalah : %i \n",i,j,x[i][j]);
    }
}
```

Kode di atas akan menghasilkan :

```
Nilai array x[0][0] adalah : 0
Nilai array x[0][1] adalah : 1
```

Nilai array x[0][2] adalah : 2
Nilai array x[0][3] adalah : 3
Nilai array x[1][0] adalah : 1
Nilai array x[1][1] adalah : 2
Nilai array x[1][2] adalah : 3
Nilai array x[1][3] adalah : 4
Nilai array x[2][0] adalah : 2
Nilai array x[2][1] adalah : 3
Nilai array x[2][2] adalah : 4
Nilai array x[2][3] adalah : 5

2. Mengupdate nilai suatu array

contoh :

```
int x[5] = {6, 8, 10, 12, 15};
printf("Nilai array x : ");
for(int i = 0; i < 5; i++){
    printf("%i ", x[i]);
}
int cari, new;
printf("\nMasukkan elemen array yang ingin dirubah : ");
scanf("%i", &cari);
//misal 15
printf("\nMasukkan nilai elemen yang baru : ");
scanf("%i", &new);
//misal 12
for(int i = 0; i < 5; i++){
    if(x[i] == cari){
        x[i] = new;
    }
}
printf("\nNilai array x yang telah di update adalah : ");
for(int i = 0; i < 5; i++){
    printf("%i ", x[i]);
}
```

Maka hasilnya adalah :

Nilai array x : 6 , 8 , 10 , 12 , 15 ,
Masukkan elemen array yang ingin dirubah : 15

Masukkan nilai elemen yang baru : 14

Nilai array x yang telah di update adalah : 6 , 8 , 10 , 12 , 14 ,

Referensi

- [Programiz](#)
- [GeeksforGeeks](#)

Asisten Praktikum Algoritma Struktur Data Informatika 2023

► Pages 8

Home

- [Modul 0. Pendahuluan](#)
- [Modul 1. Array](#)
- [Modul 2. Pointer dan Fungsi](#)
- [Modul 3. ADT dan Struct](#)
- [Modul 4. Stack](#)
- [Modul 5. Queue](#)
- [Modul 6. Single Linked List](#)

Clone this wiki locally

https://github.com/fzl-22/Algoritma-dan-Struktur-Data_Informatika.wiki.git

