



- Dasar Pemrograman – Pertemuan 11&12

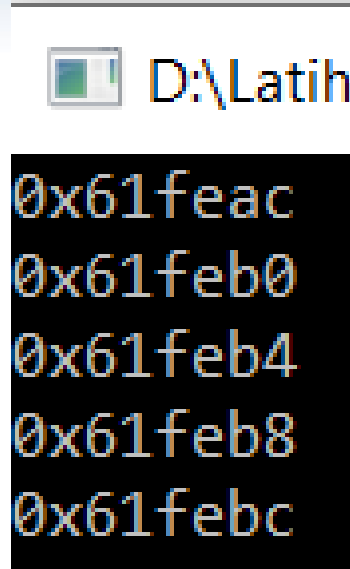
Tim Bahan Ajar Dasar Pemrograman

Teknik Informatika - S1

Fakultas Ilmu Komputer

Memahami dimana array disimpan

- Penting untuk melihat bahwa array disimpan pada suatu memory di komputer ini dengan “per byte”.
- Simbol & ketika mengotuputkan akan mengembalikan referensi lokasi variabel tersebut disimpan di memory.
- Karena integer membutuhkan alokasi 4 byte. Sehingga array i dimana bertipe array of integer dengan panjang 5 memiliki alokasi sebesar 4x5 yaitu 20. **(bukti array kontingu)**
- Lihat urutan angkanya, setiap blok ada selisih 4. (alamat ditulis dengan hexadesimal 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f)



```
#include <iostream>

using namespace std;

int main()
{
    //kamus
        int i[5];
    //algoritma
        cout << &i[0] <<endl;
        cout << &i[1] <<endl;
        cout << &i[2] <<endl;
        cout << &i[3] <<endl;
        cout << &i[4] <<endl;

        return 0;
}
```

Pertanyaan

1. Dari kode slide sebelumnya, Apa yang terjadi jika memanggil: **cout << &i[5] <<endl;** setelah perintah **cout << &i[4] <<endl;** ?
2. Dari kode slide sebelumnya, Berapa alokasi memory yang dibutuhkan jika variabel i bertipe **short**?
3. Diberikan kode sebagai berikut: →
4. apa outputnya?
5. Kenapa output i[0] dan i[3] pasti 100 dan 15?
6. Kenapa selain i[0] dan i[3] outputnya angka tidak jelas?

```
1 | #include <iostream>
2 |
3 | using namespace std;
4 |
5 | int main()
6 | {
7 |     //kamus
8 |     int i[5];
9 |     i[0] = 100;
10 |    i[3] = 15;
11 |    //algoritma
12 |    cout << i[0] <<endl;
13 |    cout << i[1] <<endl;
14 |    cout << i[2] <<endl;
15 |    cout << i[3] <<endl;
16 |    cout << i[4] <<endl;
17 |
18 |    return 0;
19 | }
```

Jawaban

1. Outputnya alamat memory juga dengan alamat selanjutnya. (tidak error)
2. 10 karena setiap blok memory kapasitasnya 2 byte dan panjang arraynya ada 5 sehingga $2 \times 5 = 10$
3. Silahkan di coba sendiri ^-^ biar tau outputnya
4. Karena $i[0]$ dan $i[3]$ sudah di assign nilai 100 dan 15
5. Karena selain $i[0]$ dan $i[3]$ belum di assign sesuatu nilai, jadi outputnya angka tidak jelas

Memset

- Dengan memberikan pustaka `string.h` atau `cstring`.
- Bisa memberikan nilai default pada setiap elemen array atau tipe string yang kita miliki.
- Contoh kode yang memberikan nilai default 0 pada semua elemen array i.

```
#include <iostream>
#include <cstring>

using namespace std;

int main()
{
    //kamus
    int i[5];
    //algoritma
    cout << sizeof(i) << endl;
    memset(i, 0, sizeof(i));
    cout << i[0] << endl;
    cout << i[1] << endl;
    cout << i[2] << endl;
    cout << i[5] << endl; // akan muncul angka tidak jelas, knp?

    return 0;
}
```

Pencarian elemen pada Array

- Diketahui array dari bilangan bulat yang terdefinisi: `int arrNilai[10]={12,3,4,1,2,3,4,20,10,11};`
- Buat variabel K dimana variabel ini akan diassign oleh inputan integer dari user.
- Jika k merupakan elemen dari `arrNilai` maka outputkan “k ketemu”
- Jika tidak outputkan “k tidak ketemu”

```
#include <iostream>

using namespace std;

int main()
{
    //kamus
    int arrNilai[10]={12,3,4,1,2,3,4,20,10,11};
    int k;
    char ket;
    //algoritma
    cin >> k;
    ket = 't';
    for(int i=0;i<10;i++)
    {
        if(k==arrNilai[i])
        {
            ket = 'y';
            break;
        }
    }
    if(ket == 'y')
    {
        cout << k << "ketemu";
    }
    else
    {
        cout << k << "tidak ketemu";
    }
    return 0;
}
```

Pencarian elemen terbesar atau terkecil

- Buat variabel penampung nilai terbesar atau terkecil
- Assign variabel sebelumnya tadi dengan elemen pertama dari array
- Lakukan penelusuran setiap elemen dari array, lakukan:
- Bandingkan elemen array yang sekarang di cari dengan variabel penampung.
- Jika memenuhi maka assign element array sekarang ke variabel penampung
- Setelah penelusuran selesai maka variabel penampung akan menyimpan nilai elemen terbesar atau terkecil dan outputkan



Array Multidimensi



Array Multidimensi

- Array 2D dapat direpresentasikan seperti matriks
- Array Lebih dari 2D atau multidimensi dapat direpresentasikan seperti tensor

Array 2D

	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 2D Lanjutan

- Terdiri dari dua batasan nilai, untuk panjang baris dan panjang kolom.
- Sehingga perlu untuk membuat kurung kotak sebanyak 2. [][]
- Indeks di kurung kotak pertama merepresentasikan elemen baris.
- Indeks di kurung kotak kedua merepresentasikan elemen kolom.
- Contoh data natural/nyata dari array 2D: gambar hitam putih atau gambar grayscale

Pemahaman Array 2D

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

	Colom ke – 0	Colom ke – 1	Colom ke – 2	Colom ke – 3
Baris ke – 0	0	1	2	3
Baris ke – 1	4	5	6	7
Baris ke – 2	8	9	10	11

- Berapakah nilai `x[0][1]`?
 - Jawaban: 1

Array 2D Assignment

```
int a[3][2];  
a[0][0] = 1;  
a[0][1] = 2;  
a[1][0] = 3;  
a[1][1] = 4;  
a[2][0] = 5;  
a[2][1] = 6;
```

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

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

```
int a[3][2];  
memset(a, 0, 6*sizeof(int));
```

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

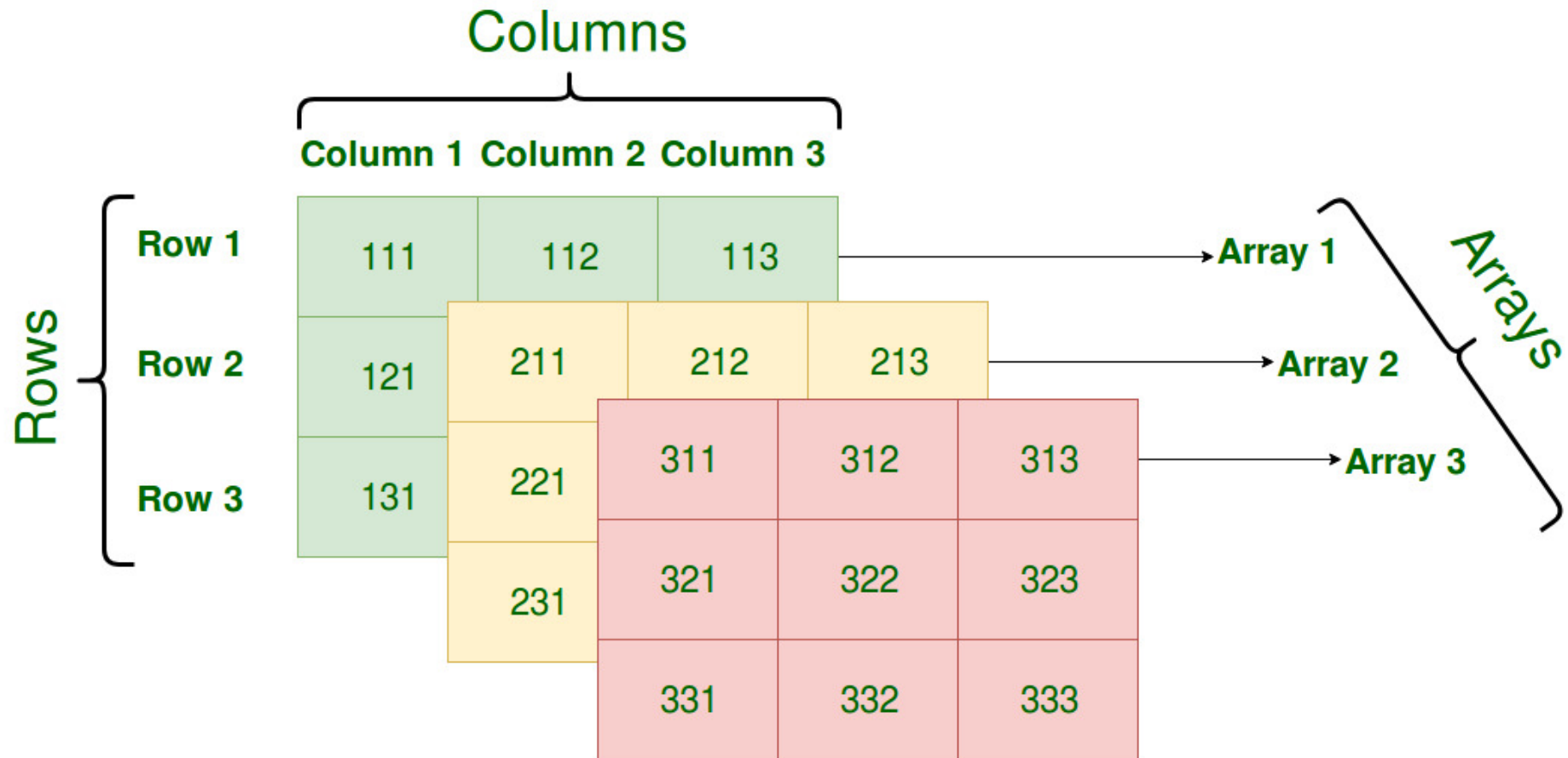
Array 2D Skema Pemrosesan Sekuensial

```
#include <iostream>

using namespace std;

int main()
{
    //kamus
    int arrNilai[2][3];
    //algoritma
    //input
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<3;j++)
        {
            cin>>arrNilai[i][j];
        }
    }
    //output
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<3;j++)
        {
            cout<<"i:"<<i<<"j:"<<j<<"nilai:"<<arrNilai[i][j]<<endl;
        }
    }
    return 0;
}
```

Array 3D



Array 3D Lanjutan

- Perbedaannya dengan Array 2D adanya tambahan kedalaman disini.
- Dalam hal ini terdiri dari tiga kotak [<dimensi baris>][<dimensi kolom>][<dimensi kedalaman>]
- Contoh data natural/nyata dari array 2D: gambar berwarna
- Dari gambar sebelumnya, kita bisa mendeklarasikan array 3D-nya seperti berikut:

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


Pemahaman Array 3D

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

	Colom ke – 0	Colom ke – 1	Colom ke – 2
Baris ke – 0	0	2	4
Baris ke – 1	6	8	10

d = 0

	Colom ke – 0	Colom ke – 1	Colom ke – 2
Baris ke – 0	1	3	5
Baris ke – 1	7	9	11

d = 1

- *d merupakan depth/kedalaman*

- Berapakah nilai $x[0][1][1]$?
 - Jawaban: 3
- Berapakah nilai $x[1][2][0]$?
 - Jawaban: 10

Array 3D Skema Pemrosesan Sekuensial

```
#include <iostream>

using namespace std;

int main()
{
    //kamus
    int arrNilai[2][3][4];
    //algoritma
    //input
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<3;j++)
        {
            for(int k=0;k<3;k++)
            {
                cin>>arrNilai[i][j][k];
            }
        }
    }
    //output
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<3;j++)
        {
            for(int k=0;k<3;k++)
            {
                cout<<"i:"<<i<<"j:"<<j<<"k:"<<k<<" nilai:"<<arrNilai[i][j][k]<<endl;
            }
        }
    }
    return 0;
}
```

Referensi



Utama :

1. Liem, Inggriani. Diktat Pemrograman Prosedural Informatika ITB. IF-ITB. 2007
2. Bjarne Stroustrup, 2014, Programming: Principles and Practice Using C++ (Second Edition), Addison-Wesley Professional

Pendukung :

1. Introduction to Computer Science and Programming in Python, MIT
<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016>
2. Introduction to Computer Science and Programming, MIT <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/index.htm>