



- Dasar Pemrograman – Pertemuan 11

Tim Bahan Ajar Dasar Pemrograman
Teknik Informatika - S1
Fakultas Ilmu Komputer

Capaian Pembelajaran

- Menjelaskan, menggunakan, membuat, dan mempraktikkan larik atau array satu dan lebih dari satu dimensi pada pemrograman prosedural.



Array



Tipe Skalar

- Tipe skalar membuat setiap variabel hanya memiliki satu nilai.
- Contoh program dengan objek tipe skalar

Program nilai mata kuliah

Kamus

```
nilaiMhs1, nilaiMhs2, nilaiMhs3, nilaiMhs4, nilaiMhs5 : float  
rata2 : float
```

Algoritma

```
input(nilaiMhs1, nilaiMhs2, nilaiMhs3, nilaiMhs4, nilaiMhs5)  
rata2 ← (nilaiMhs1+nilaiMhs2+nilaiMhs3+nilaiMhs4+nilaiMhs5)/5  
output(rata2)
```

Apa kelemahan program diatas?
Cuma 5 mhs tidak masalah, kalau 1000?
Diketik semua? 1 variabel 1 nilai mhs?

Solusi: Array

- 1 variabel 1 nilai mahasiswa:

Mhs1



Mhs2



Mhs3



Mhs4



Mhs5



- Ilustrasi Array seperti tabel:

1 variabel menyimpan 5 nilai mahasiswa

Atau, 1 variabel array memiliki 5 elemen nilai

Mhs



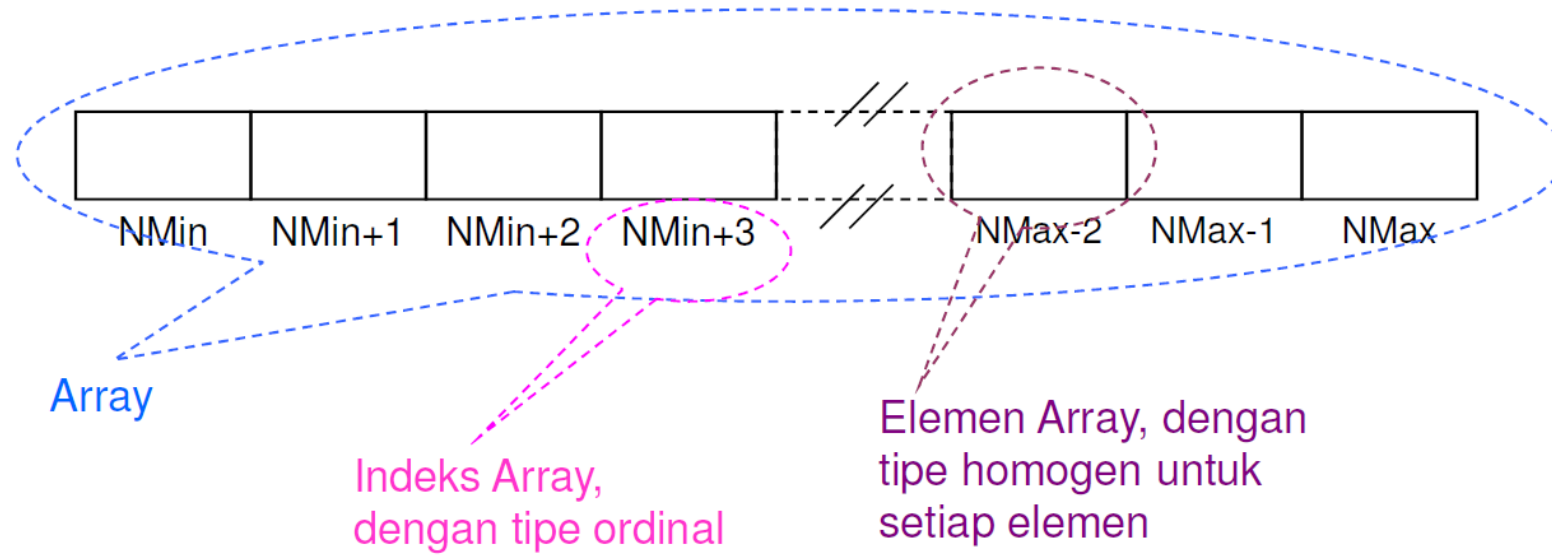
- Identik dengan simbol []

Definisi Array

- **Array** merupakan tipe **non-skalar** yang mengacu pada **sebuah/sekumpulan elemen tipe** dan diakses melalui **indeks**.
- Elemen yang ada didalam array harus mempunyai **tipe yang sama**.
- Disebut juga tabel, vektor, atau larik
- Penyimpanan tipe ini dilakukan **secara kontingu dalam memory komputer**

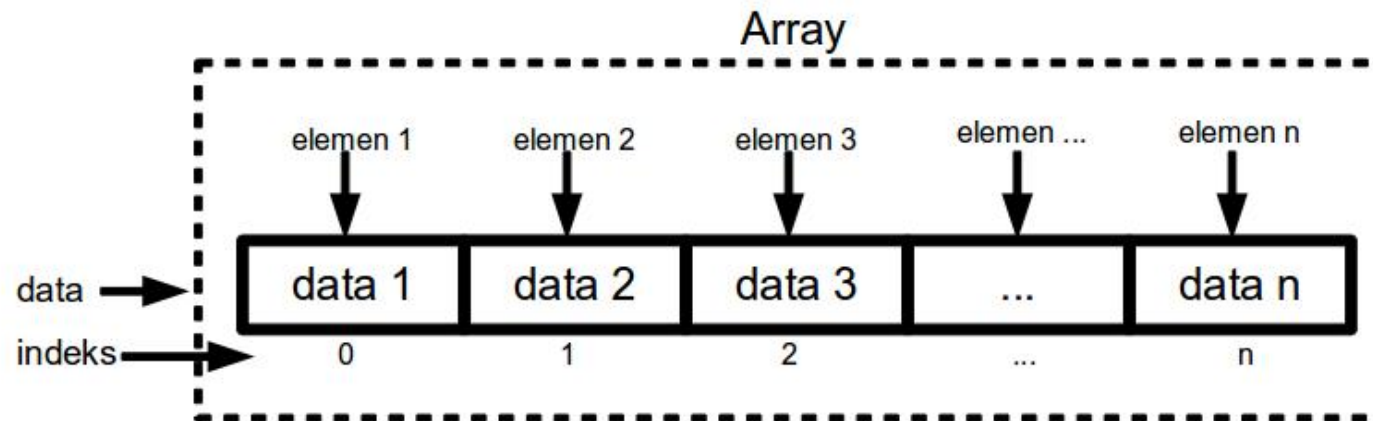
34	82	78	69	43
----	----	----	----	----

Visualisasi Array



Indeks Array

- Elemen suatu array dapat di akses melalui index yang telah terdefinisi: **array[index]**
- Elemen di awali dari **0** adalah **elemen pertama** (beberapa bahasa pemrograman yang lain mungkin memulai indeks dari 1)



Pengaksesan

- Pengaksesan elemen array dapat dilakukan berurutan atau acak berdasarkan indeks tertentu secara langsung.

nama_var_array[indeks]

- Contoh : **nilai[3]** bernilai 6

0	1	2	3	4	5	Indeks
8	7	5	6	4	3	Data
1d2	1d4	1d6	1d8	1da	2dc	Alamat

Array di C++

- Di C++, cara membangkitkan array adalah :
 - Deklarasi: `<tipe> <nama>[<ukuran>];`
 - Deklarasi & Inisialisasi langsung: `<tipe> <nama>[<ukuran>]={<val1>,...,<val-n>}`
- Akses setiap nilai:
 - `<nama>[<indeks>]`

Deklarasi Array

`tipe_data nama_var_array [ukuran_indeks]`

- `tipe_data` : menyatakan jenis tipe data elemen (int, char, float, dll)
- `nama_var_array` : menyatakan nama variabel yang dipakai.
- `ukuran` : menunjukkan jumlah indeks maksimal elemen array.
- Contoh : `int nilai[10];`

Lanjutan Array

❑ Inisialisasi array

```
int b[4] = {1, 2, 3, 4};
```

❑ Array n-dimensi

```
int b[<s1>][<s2>]...[<sn>]  si Harus konstanta
```

❑ Indeks array di hitung dari 0 sampai (s_i-1)

❑ C/C++ tidak menangani "out-of-range exception" artinya akan memberikan nilai yang tidak jelas

```
int b[4] = {1, 2, 3, 4};  
for (int i = 0; i < 4; i++)  
{  
    printf("%d\n", b[i]);  
}  
printf("%d\n", b[10]);
```

b[10] = ?

Array Assignment

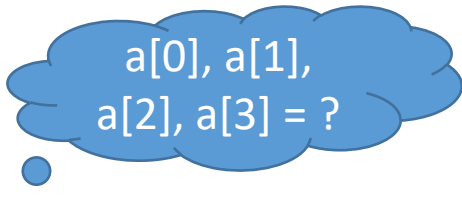
```
int a[4] = {1, 2, 3, 4};
```

```
int a[4] = {1};
```

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

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

```
int a[4];  
memset(a, 0, 4*sizeof(int));
```



a[0], a[1],
a[2], a[3] = ?



Pemrosesan Sekuensial pada Array



Skema pemrosesan sekuensial pada Array

- Merupakan pemrosesan sekuensial **tanpa mark** dimana elemen terakhir mengandung info bahwa elemen tersebut adalah elemen terakhir
- Dimungkinkan adanya akses langsung **jika indeks terdefinisi**
 - First-Elmt adalah elemen array dengan indeks terkecil
 - Next-Elmt dicapai melalui suksesor indeks (indeks berikutnya)
- Model akses sekuensial tanpa mark
 - kondisi berhenti adalah jika indeks sudah mencapai harga indeks yang terbesar yang telah terdefinisi
- Array tidak mungkin “kosong”
 - jika kita mendefinisikan tabel, maka minimal mengandung sebuah elemen

Skema umum

KAMUS UMUM PEMROSESAN ARRAY

constant NMin: integer $\leftarrow 1$ {batas bawah}
constant NMax: integer $\leftarrow 100$ {batas atas}
type ElType : ... {suatu type terdefinisi, eg integer/float}
i : integer [NMin..NMax]
T : array [NMin..NMax] of ElType {tabel dengan elemen bertipe ElType}
prosedur Inisialisasi {persiapan yang harus dilakukan sebelum pemrosesan}
prosedur Proses (input X : ElType) {proses thd Current-Elmt tabel T}
prosedur Terminasi { "penutupan" setelah pemrosesan selesai}

SKEMA Pemrosesan Array T untuk indeks dari NMin sampai Nmax {transversal Array T untuk indeks yang bernilai NMin sampai NMax}

Inisialisasi

i transversal [NMin...NMax]

Proses (T_i) Terminasi

Pemrosesan sekuensial pada Array

Kasus mengisi dan mengoutputkan elemen array dengan jumlah elemen yang diketahui

Program InputOutputArray

Kamus

constant NMin: integer $\leftarrow 1$ {batas bawah}
constant NMax: integer $\leftarrow 100$ {batas atas}
i : integer [NMin..NMax]
T : array [NMin..NMax] of integer
N: integer

Algoritma

{inisialisasi}
repeat
 input(N)
until (NMin <= N <= NMax)
i transversal [NMin...NMax]
 input (T_i)
i transversal [NMin...NMax]
 output (T_i)

Pemrosesan sekuensial pada Array

Kasus mengisi dan mengoutputkan elemen array dengan jumlah elemen tidak diketahui

Program InputOutputArray

Kamus

constant NMin: integer $\leftarrow 1$ {batas bawah}
constant NMax: integer $\leftarrow 100$ {batas atas}
i : integer [NMin..NMax]
T : array [NMin..NMax] of integer
x: integer {nilai yang akan di baca sebagai elemen array}

Algoritma

$i \leftarrow \text{Nmin}$ {inisialisasi}
input(x) {first element}
while x $\neq 9999$ and $i < \text{NMax}$ do
 $T[i] \leftarrow x$
 $i \leftarrow i + 1$
 input(x)
 {x = 9999 or $i > \text{NMax}$ }
if $i > \text{Nmax}$ then
 output("array penuh")
i transversal [Nmin-1...i+1]
 output(T[j])

Skema Pemrosesan Sekuensial

- Looping adalah jawabannya!

```
#include <iostream>

using namespace std;

int main()
{
    //kamus
    int arr[5]; // deklarasi
    //algoritma
    // input
    for(int i=0;i<5;i++)
    {
        cin >> arr[i];
    }
    // output
    for(int i=0;i<5;i++)
    {
        cout << arr[i] << endl;
    }

    return 0;
}
```

```
#include <iostream>

using namespace std;

int main()
{
    //kamus
    int arr[5]; // deklarasi
    //algoritma
    // input dan output sekaligus
    for(int i=0;i<5;i++)
    {
        cin >> arr[i];
        cout << arr[i] << endl;
    }

    return 0;
}
```

Menyiapkan panjang array dengan input user

- Pastikan bahwa batasan bertipe integer atau konstanta.

```
#include <iostream>

using namespace std;

int main()
{
    //kamus
    int N;
    cin >> N;
    int arr[N]; // deklarasi
    //algoritma
    // input dan output sekaligus
    for(int i=0;i<N;i++)
    {
        cin >> arr[i];
        cout << arr[i] << endl;
    }

    return 0;
}
```

Latihan

- Buat program dengan array untuk menyimpan nama hari dalam 1 minggu!

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

using namespace std;

int main()
{
    //kamus
    string arrNamaHari[7];
    int i;
    //algoritma
    i = 0;
    // inputkan hari
    while(i < 7)
    {
        cin >> arrNamaHari[i];
        i++;
    }
    i = 0;
    // outputkan hari
    while(i < 7)
    {
        cout << arrNamaHari[i] << endl;
        i++;
    }

    return 0;
}
```

Program NamaHari

Kamus

arrNamaHari: array [1...7] of string

i : integer

Algoritma

$i \leftarrow 0$

{inputkan hari}

while $i < 7$ do

 input(arrNamaHari_i)

$i \leftarrow i + 1$

$i \leftarrow 0$

{outputkan hari}

while $i < 7$ do

 output(arrNamaHari_i)

$i \leftarrow i + 1$

Latihan

- Hitung sejumlah nilai float didalam suatu array yang setiap elemennya di inputkan oleh user!

```
main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      //kamus
8      int N,i;
9      float sum;
10     cin >> N;
11     float arrNilai[N];
12     //algoritma
13     i =0;
14     // inputkan nilai
15     while(i<N)
16     {
17         cin >> arrNilai[i];
18         i++;
19     }
20     i =0;
21     sum = 0.0f;
22     // proses penjumlahan
23     while(i<N)
24     {
25         sum = sum + arrNilai[i];
26         i++;
27     }
28
29     cout << sum <<endl;
30     return 0;
31 }
```

Dari soal sebelumnya hitung rata-rata?

- Di baris kode ke-29 diganti `cout << sum/N << endl;`

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>
3. <https://www.geeksforgeeks.org/multidimensional-arrays-c-cpp/>