

Pertemuan 02

Array

Learning Outcomes

Pada akhir pertemuan ini, diharapkan mahasiswa akan mampu :

- mendemonstrasikan Tipe Data Array.
- melakukan perhitungan dg AMF.
- menerapkan Array pada program aplikasi komputer.

Materi

- Karakteristik
- Dimensi
- Akses data
- Operasi
- AMF
- Passing Parameter

Karakteristik

KARAKTERISTIK dari Array :

- ♦ Hubungan antar elemen adalah linier.
- ♦ Penempatan di memory secara PHYSICAL & LOGICAL serupa.
- ♦ Terdiri dari Indeks & Komponen.
- ♦ Hubungan Indeks & Komponen *one-to-one*.
- ♦ Semua komponen mempunyai tipe sama (HOMOGENEOUS).

Dimensi

DIMENSI

- ♦ 1-Dimensi : List, Vektor.
- ♦ 2-Dimensi : Tabel, Matriks.
- ♦ Multidimensional : secara teoritis jumlah dimensi tidak terbatas, hanya dibatasi oleh besarnya memory.

Akses Data

PENCAPAIAN / Akses DATA

- ♦ Positional Access : akses berdasarkan nilai indexnya
- ♦ Associative Access : akses berdasarkan nilai / isi komponennya

Akses data pada array dilakukan secara RANDOM dengan menggunakan indeks.

- Array 1d : perlu 1 indeks
- Array 2d : perlu 2 indeks
- Array 3d : perlu 3 indeks
- d.s.t.

Operasi

Retrieve (s,k,i)

mengambil nilai yang ke-i dari array s dan di assign ke var k.

Pada bahasa C $\rightarrow k = s[i];$

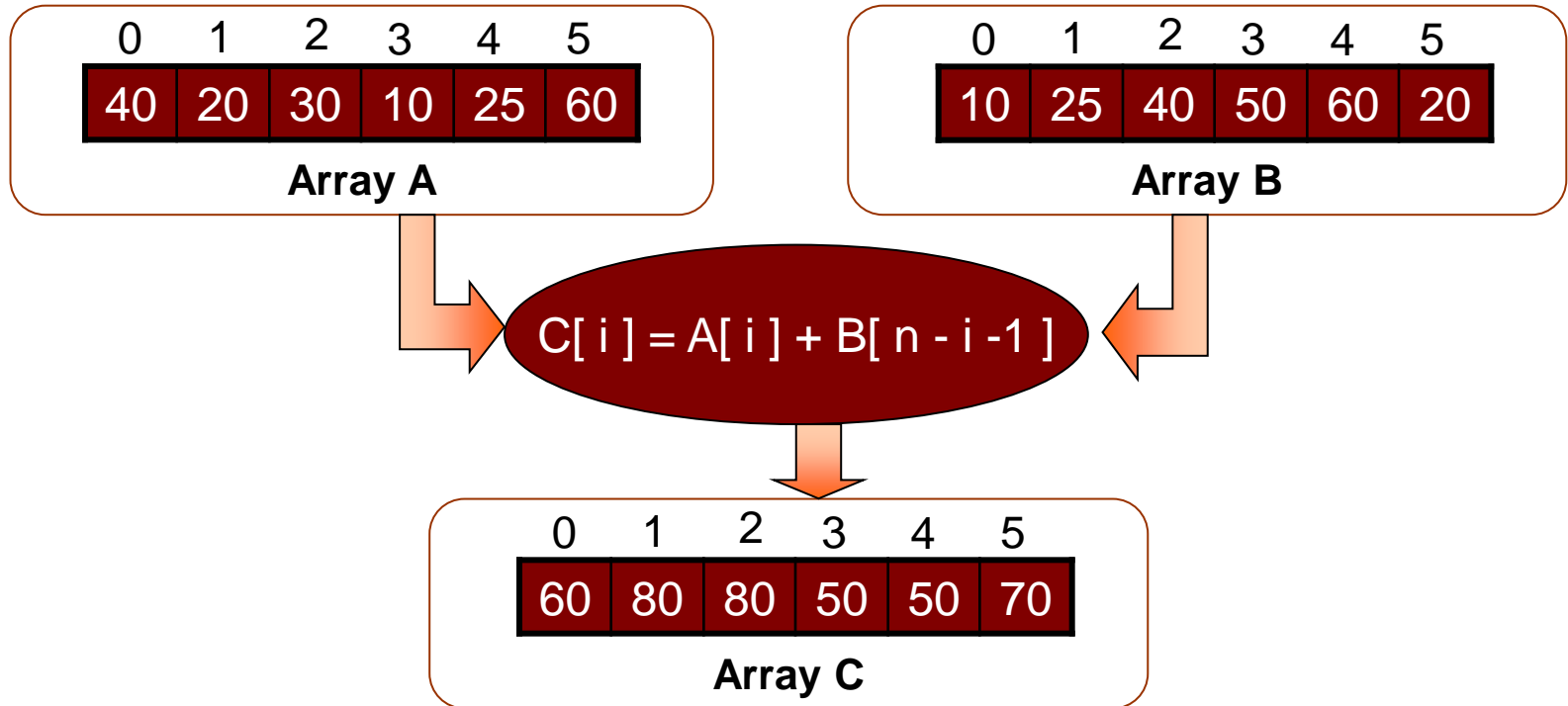
Update (s,k,i)

mengubah nilai yang ke-i dari array s dengan nilai dari var k.

Pada bahasa C $\rightarrow s[i] = k;$

Contoh Operasi

Contoh operasi Retrieve dan Update pada Array



Operasi Retrieve ?
Operasi Update ?
Programming ?

Array Mapping Function

Parameter

- Base Address (b) : alamat (byte pertama) dr array yg di assign pd saat binding time. Binding time adalah waktu dimana array diassign pd suatu lokasi/address di memory, bisa pada saat compile, execute, dll.
- Component Length (L):panjangnya memory utk menyimpan 1 komponen. L tergantung dari tipe komponen dan bahasa pemrograman misalnya tipe integer :

Bahasa Pemrograman	L
Turbo Pascal 7.0	2
Turbo C 2.0	2
Visual C++ 5.0	4

Array Mapping Function(2)

- Lower Bound (lk) : nilai index yang terkecil.
- Upper Bound (uk) : nilai index yang terbesar.

Contoh :

int S[10]; maka lk = 0 uk = 9

- ♦ Dimension (d) : besarnya dimensi dari suatu array.

Contoh :

int S[10][20]; \Rightarrow maka d = 2

int S[9]; \Rightarrow maka d = 1

Array Mapping Function(3)

Fungsi untuk memetakan nilai index (i) ke alamat (*address*) dari komponen.

Fungsi mapping / pemetaan :

- Row-major Order : Cara penyimpanan array di dalam bahasa pemrograman.
- Virtual Origin atau Virtual Base : Nilai constan c_0 (lokasi dari komponen dengan indeks 0) dalam AMF.

Rumus dalam AMF :

$$\text{addr}(s[i_1][i_2] \dots [i_d]) = c_0 + c_1 \times i_1 + c_2 \times i_2 + \dots + c_d \times i_d$$

dimana : $c_d = L$

$$c_{t-1} = (u_t - l_t + 1) \times c_t \text{ dan } 1 < t \leq d$$

$$c_0 = b - (c_1 \times l_1) - \dots - (c_d \times l_d)$$

Rumus perhitungan besarnya memory (M) :

$$M = L \times (u_1 - l_1 + 1) \times \dots \times (u_d - l_d + 1)$$

Array Mapping Function (4)

Array Spesial

- Array Segitiga (*Triangular Array*)

- Lower Triangular Array : Array 2d berbentuk bujur sangkar ($u_1 = u_2$), dimana semua komponen di atas diagonal berisi 0.
- Upper Triangular Array : Perbedaan dengan Lower Triangular Array, adalah semua komponen di bawah diagonal yang berisi 0.

Memerlukan lebih sedikit memory, krn angka 0 tdk perlu disimpan.

Rumus AMF untuk Triangular Array

$\text{addr}(s[i,j]) = c_0 + c_1 \times (i \times (i - 1)) + c_2 \times j$, dimana : $c_0 = b$, $c_1 = L/2$, $c_2 = L$

Jumlah elemen = $(u \times (u+1))/2$

Jumlah Memory = $L \times \text{Jumlah elemen}$

- Array Jarang (*Sparse Array*)

Array yg kebanyakan komponennya mempunyai satu nilai yg sama, misalnya nilai 0, hanya sebagian kecil yg mempunyai nilai yg tdk sama dg 0.

Passing Parameter

Cntoh potongan program :

```
...
char LeftMost(char S[])
{
    return(S[0]);
};

char RightMost(char S[])
{
    return(S[Len(S)-1]);
};

void main()
{
    char MyString[MaxLen];
    ...
    printf("Leftmost elemen      :%c",LeftMost(MyString));
    printf("Rightmost elemen     :%c",RightMost(MyString));
    ...
};
```

The diagram illustrates the passing of the 'MyString' parameter from the 'main' function to the 'LeftMost' and 'RightMost' functions. Red dashed circles highlight the 'S[]' parameter in the function definitions and the 'MyString' argument in the function calls. Two red dashed curved arrows originate from the 'MyString' argument in the 'main' function and point to the 'S[]' parameters in the 'LeftMost' and 'RightMost' functions, respectively.

CONTOH PROGRAM SEDERHANA UNTUK ARRAY

// PROGRAM ISI DAN CETAK ARRAY

int i,j,k;

**int A[10]; // Array A dengan 10
elemen**

int Isi_data()

{ for(i=0; i<10; i++) A[i] = i*3;

return 10;

}

```
int Cetak_data()  
{    printf("Data Elemen Array\n\n");  
        for(i=0; i<10; i++) printf("  %5d  
        ,A[i]);  
        printf("\n\n");    return 1;  
}  
  
int main()  
{    Isi_data();  
        Cetak_data();  
        return 5;  
}
```

Selesai