

Struktur Data Statis

(Array / Larik)



Akmal, S.Si, MT

Mata Kuliah : Struktur Data

Tujuan

- ❑ Mahasiswa dapat menjelaskan pengertian array dan bisa menerangkan operasi dasar menggunakan array dengan benar
- ❑ Mahasiswa dapat : Mengoperasikan dan membuat program dari semua algoritma array dengan benar menggunakan model program modular.

Pokok Bahasan

- Array 1 dimensi
- Array 2 dimensi (Matriks)

Array / Larik

- ❑ Array adalah suatu tipe data bentukan yang menyimpan sekumpulan elemen data yang bertipe sama, dan memiliki indeks.
- ❑ Indeks array harus tipe data yang menyatakan keterurutan, misalnya *integer* atau karakter
- ❑ Penyimpanan di memori secara kontigu.

Array 1 Dimensi

- ❑ Setiap elemen larik ditulis dengan notasi :

n-1

$\Sigma A[i] = A[0], A[1], A[2], A[3], A[4], \dots, A[n-1]$

i=0

Angka di dalam kurung siku menyatakan indeks array yang dimulai dari 0 sampai dengan (n-1) yang diinginkan.

Deklarasi

Deklarasi Sebagai variabel

<tipe> variabel[ukuran];

Contoh :

int A[10];

- Untuk bisa membuat program modular dengan melewati (passing) array baik by value atau by referensi maka disarankan untuk membuat cara sebagai berikut:
 - Definisikan suatu tipe alias dari data bertipe array
 - Variabel array kemudian mengacu ke tipe alias tersebut
- Contoh :

typedef int larik[10];

larik A;

a merupakan sebuah variabel yang bertipe array of integer dengan ukuran maksimal 10 elemen dengan index 0 .. 9

```

typedef int larik[10]; //pendefinisian alias array int

void banyakData(int& n);
void isiLarik(larik& a, int n);
void printLarik(larik a, int n);
void cariRata(larik a, int n, float& rata);

main() {
    larik x;          // variabel array x bertipe alias larik
    float rata2;
    int n;
    banyakData(n)
    isiLarik(x,n);
    printLarik(x,n);
    cariRata(x,n,rata2);
    cout<<"Rata-rata = "<<rata2<<endl;
}

void banyakData(int& n){          // Input banyak data
    cout<<"Banyak data : "; cin>>n;
}

void isiLarik(larik& a, int n){    // Input data larik
    for (int i=0;i<n;i++) {
        cout<<"Masukkan data ke- "<<(i+1)<<" : "; cin>>a[i];
    }
}

void printLarik(larik a, int n){    //Mencetak data larik
    cout <<"Data yang sudah dimasukkan" <<endl;
    cout <<"-----"<<endl;
    for(int i=0;i<n;i++) {
        cout <<"Data ke- "<<(i+1)<<" = "<< a[i] <<endl;
    }
}

void cariRata(larik a, int n, float& rata){
    float jumlah=0;                // Ubah jadi fungsi ?
    for (int i=0;i<n;i++) {
        jumlah=jumlah+a[i];
    }
    rata=jumlah/n;
}

```

Array 2 Dimensi (Matriks)

Deklarasi Sebagai variabel

```
<tipe> variabel[indBaris][indKolom];
```

Contoh :

```
int x[10][10];
```

Cara Deklarasi Matriks untuk membuat Program modular

- Definisikan suatu tipe alias dari data bertipe matriks
- Variabel matriks kemudian mengacu ke tipe alias tersebut

```
typedef int matriks[10][10];
```

```
matriks x;
```

Contoh matriks

```
typedef int  matriks[10][10];

main() {
    matriks x;          // variabel array x bertipe matriks
    int nBaris, nKolom;

    banyakData(nBaris, nKolom)
    isiMatriks(x,nBaris, nKolom);
    cetakMatriks(x,nBaris, nKolom);
}
```


Contoh matriks

```
void banyakData(int& nBaris, int& nKolom){
    cout<<"Banyak baris matriks : "; cin>>nBaris;
    cout<<"Banyak kolom matriks : "; cin>>nKolom;
}

void isiMatriks(matriks& x,int nBaris, int nKolom){
    cout<<"Inputan untuk matriks"<<endl;
    for (int i=0; i<nBaris; i++){
        for (int j=0; j<nKolom; j++){
            cout<<"Masukkan data ke-" <<i<<","<<j<<" : "; cin>>x[i][j];
        }
    }
}

void cetakMatriks(matriks x,int nBaris, int nKolom){
    cout<<"Pencetakan matriks"<<endl;
    for (int i=0; i<nBaris; i++){
        for (int j=0; j<nKolom; j++){
            cout<<x[i][j]<<" ";
        }
        cout<<endl;
    }
}
```