ARTIKEL STRUKTUR DATA, PENGERTIAN ARRAY

ALdesion Denagi Zenda

175100033

Universitas Mitra Indonesia, Sistem Informasi

aldesiondenagi.students@umitra.ac.

ABSTRAK

Array atau larik di definisikan sebagai pemesanan alokasi memory berurutan. definisi ini kurang tepat, karena terjadi kerancuan antara struktur data dan representasinya. Memang benar array hampir selalu di implementasikan menggunakan memory berurutan tapi tidak selalu demikian.

KATA KUNCI : PENGERTIAN DAN PENJELASAN ARRAY

ARRAY

1. PENGERTIAN

Array atau larik di definisikan sebagai pemesanan alokasi memory berurutan. definisi ini kurang tepat, karena terjadi kerancuan antara struktur data dan representasinya. Memang benar array hampir selalu di implementasikan menggunakan memory berurutan tapi tidak selalu demikian.

Semua elemem array bertipe sama. Array cocok untuk organisasi kumpulan data homogen yang ukuran atau jumlah elemen maksimumnya telah diketahui dari awal.

Homogen adalah bahwa setiap elemen dari sebuah array tertentu haruslah mempunyai tipe data yang sama.

2. KARAKTERISTIK ARRAY

- a) Mempunyai batasan dari pemesanan alokasi memori (bersifat statis)
- b) Mempunyai tipe data sama (bersifat homogen)
- c) Dapat diakses secara acak.

2.1. DEKLARASI ARRAY

Ada tiga hal yang harus di ketahui dalam mendeklarasikan array, yaitu :

- a) Type data array
- b) Nama variable array
- c) Subkrip / index array.

Contoh deklarasi dari array adalah sebagai berikut :

`int A[5]` ; artinya variabel A adalah kumpulan data sebanyak 5 bilangan bertipe integer.

2.2. JENIS ARRAY

1. ARRAY DIMENSI SATU

Deklarasi : Type_Data Nama_Variabel [index]

Rumus untuk menentukan jumlah elemen dalam array adalah :

$$\begin{array}{l} n \\ \pi \text{ (Index Array)} \\ i = 1 \end{array}$$

π = Perkalian dari index sebelumnya (untuk arraybdimensi dua dan tiga).

PEMETAAN (MAPPING) ARRAY DIMENSI SATU KE STORAGE

Rumus : $@A[i] = B + (i - 1) * L$

Dimana : $@A[i]$: Posisi array yang dicari

B : Posisi awal index di memori computer

i : Subkrip atau index array yang di cari

L : Ukuran atau besar memori suatu tipe data

2. ARRAY DIMENSI DUA

Deklarasi : Type_Data Nama_Variabel [index1] [index2]

Menentukan jumlah elemen dalam array dimensi dua :

$$\begin{array}{l} n \\ \pi \text{ (Index Array)} \\ i = 1 \end{array}$$

π = Perkalian dari statemen sebelumnya

PEMETAAN (MAPPING) ARRAY DIMENSI DUA KE STORAGE

Terbagi dua cara pandang (representasi) yang berbeda :

- Secara kolom per kolom (coloumn major order / CMO)

$$@M[i][j] = M[0][0] + \{(j - 1) * K + (i - 1)\} * L$$

- Secara baris per baris (row major order / RMO)

$$@M[i][j] = M[0][0] + \{(i - 1) * N + (j - 1)\} * L$$

Keterangan :

@M[i][j] = Posisi array yang di cari, M[0][0] = Posisi alamat awal index array,
i = Baris, j = Kolom, L = Ukuran memory type data, K = Banyaknya elemen
per kolom, N = Banyaknya elemen per baris.

3. ARRAY DIMENSI TIGA

Deklarasi : type_Data Nama_Variabel [index1][index2][index3]

Menentukan jumlah elemen dalam array dimensi tiga :

$$\sum_{i=1}^n \pi \text{ (Index Array)}$$

π = Perkalian dari statemen sebelumnya

PEMETAAN (MAPPING) ARRAY DIMENSI TIGA KE STORAGE

$$\text{Rumus : } @M[n][m][p] = M[0][0][0] + \{((n - 1) * (\text{index1})) + ((m - 1) * (\text{index2})) + ((p - 1) * (\text{index3}))\} * L$$

TRIANGULAR ARRAY (ARRAY SEGI TIGA)

Triangular array dapat merupakan Upper Triangular (seluruh elemen di bawah diagonal utama = 0), ataupun Lower Triangular (seluruh elemen di atas diagonal utama = 0).

Dalam array Lower Triangular dengan N baris, jumlah maksimum elemen ≤ 0 , tidak lebih dari

$$\sum_{i=1}^N I = N(N+1)/2$$

SPERSE ARRAY (ARRAY JARANG)

Suatu array yang sangat banyak elemen nol-nya.

2.3. OPERASI DASAR PADA ARRAY

Operasi terhadap elemen di array dilakukan dengan pengaksesan langsung. Nilai di masing-masing posisi elemen dapat diambil dan nilai dapat disimpan tanpa melewati posisi-posisi lain.

Terdapat dua tipe operasi, yaitu :

1. Operasi terhadap satu elemen / posisi dari array
2. Operasi terhadap array sebagai keseluruhan

Dua operasi paling dasar terhadap satu elemen / posisi adalah

1. Penyimpanan nilai elemen ke posisi tertentu di array
2. Pengambilan nilai elemen dari posisi tertentu di array

Operasi-operasi dasar terhadap array secara keseluruhan adalah :

1. Operasi penciptaan
2. Operasi penghancuran
3. Operasi pemrosesan traversal
4. Operasi pencarian (table look-up)
5. Operasi sorting

2.3.1. PENCIPTAAN DAN PENGHANCURAN

Operasi penciptaan biasa disebut inisialisasi.

Operasi ini untuk mempersiapkan struktur data untuk operasi-operasi berikutnya. Operasi penghancuran menyatakan ketidak berlaku struktur data atau membebaskan memory, menyerahkan memory ke manajemen memory agar dapat di pergunakan keperluan lain.

Operasi penghancuran penting terutama bila struktur data di implementasikan secara dinamis menggunakan pointer

2.3.2. PENYIMPANAN DAN PENGAMBILAN NILAI

Biasanya bahasa pemrograman menyediakan sintaks tertentu untuk penyimpanan dan pengambilan nilai elemen pada posisi tertentu di array.

Contoh :

$A[10] = 78$, berarti penyimpanan nilai 78 ke posisi ke-10 dari array A

$C = A[10]$, berarti pengambilan nilai elemen posisi ke-10 dari array A

2.3.3. PEMROSESAN TRANSVERSAL

Operasi pemrosesan transversal adalah pemrosesan mengolah seluruh elemen secara sistematis.

2.3.4. PENCARIAN DI ARRAY (table look-up)

Pencarian di array (table look-up) adalah proses pencarian suatu nilai di array. Klasifikasi pencarian di array adalah :

1) Pencarian sekuen (sequential searching), yaitu:

i. Tanpa Boolean, terbagi:

- Tanpa sentinen
- Dengan sentinen

ii. Menggunakan boolean

2) Pencarian secara biner / dikotom (binary = dichotomy searching).

2.4. PENGURUTAN ARRAY

Pengurutan atau sorting adalah proses yang paling sering dilakukan dalam pengolahan data. pengurutan di bedakan menjadi dua, yaitu :

a. Pengurutan internal

Pengurutan dilakukan terhadap sekumpulan data di media memory internal komputer dimana data dapat di akses elemennya secara langsung.

b. Pengurutan eksternal

Pengurutan data di memory sekunder. Biasanya data bervolume besar sehingga tidak mampu dimuat semuanya di memori utama.

3. KEUNGGULAN DAN KELEMAHAN ARRAY

Keunggulan array adalah sebagai berikut :

1. Array sangat cocok untuk pengaksesan acak. Sembarang elemen di array dapat diacu secara langsung tanpa melalui elemen-elemen lain.
2. Jika berada di suatu lokasi elemen, maka sangat mudah menelusuri ke elemen-
elemen tetangga, baik elemen pendahulu atau elemen penerus
3. Jika elemen-elemen array adalah nilai-nilai independen dan seluruhnya harus terjaga,
maka penggunaan penyimpanannya sangat efisien.

Kelemahan array adalah sebagai berikut :

Array mempunyai fleksibilitas rendah, sehingga tidak cocok untuk berbagai aplikasi karena array mempunyai batasan sebagai berikut :

1. Array harus bertipe homogen. Kita tidak dapat mempunyai array dimana satu elemen
adalah karakter, elemen lain bilangan, dan elemen lain adalah tipe-tipe lain
2. Kebanyakan bahasa pemrograman mengimplementasikan array statik yang sulit
diubah ukurannya di waktu eksekusi. Bila penambahan dan pengurangan terjadi
terus-menerus, maka representasi statis

- Tidak efisien dalam penggunaan memori
- Menyiakan banyak waktu komputasi
- Pada suatu aplikasi, representasi statis tidak dimungkinkan

Bila penambahan dan pengurangan terjadi terus menerus, maka representasi statis (array):

1. Tidak efisien dalam penggunaan memory
2. Menyiakan banyak waktu komputasi
3. Pada suatu aplikasi, representasi statis tidak di mungkinkan.

KESIMPULAN

Struktur data merupakan salah satu bahan dasar pembuatan program. Pemakaian struktur data yang tepat di dalam proses pemrograman, akan menghasilkan algoritma yang jelas dan tepat sehingga menjadikan program secara keseluruhan lebih sederhana. Array merupakan bagian dari struktur data yaitu termasuk kedalam struktur data sederhana yang dapat di definisikan sebagai pemesanan alokasi memory sementara pada komputer. Apabila kita membuat program dengan data yang sudah kita ketahui batasnyamaka kita menggunakan Array (type data statis), namun apabila datanya belum kita ketahui batasnya maka gunakan pointer (type data dinamis).

DAFTAR PUSTAKA

<http://dyananyun.blogspot.com/2010/02/struktur-data-array.html>

<http://kintung05.wordpress.com/2008/11/23/pengertian-struktur-data/>

<http://www.nusinau.com/pengertian-struktur-data-2/>

<http://www.nusinau.com/pengenalan-array-dan-string/>

Hariyanto, Bambang. 2000. *struktur data*. Bandung : CV. Informatika.

Modul dasar struktur data “BSI”