ARTIKEL STRUKTUR DATA, STRUKTUR DATA

Muhamad Alwi Alfagih

175100047

Universitas Mitra Indonesia, Sistem Informasi

muhamadalwi.students@umitra.ac.id

ABSTRAK

Pemrograman dalam struktur data ada beberapa macam. Salah satunya adalah pemrograman C++. Dalam pemrograman ini biasanya menggunakan variable Array, Struktur dan Linked List

Makalah ini membahas tentang 3 variabel tersebut dimana ketiga variable mempunyai ciri dan umum yang berbeda sesuai dengan tipe file yang di gunakan pembaca. Seperti array yang menggunakan satu dimensi dan dua dimensi serta 3 dimensi dimana sangat berbeda dengan struktur yang menggunakan tingkatan prosedur.

Pemrograman ini merupakan pemrograman yang berbeda dari pemrograman lainnya misalnya VB, Delphi atau Pascal namun perbedaan juga tidak begitu signifikan pada pemrograman pascal.

1.2 Rumusan Masalah

- Pengertian Array dan contoh-contoh programnya pada penggunaan program C++
- Pengertian Struktur dan contoh-contoh programnya pada penggunaan program C++.
- Pengertian Linked List dan contoh-contoh programnya pada penggunaan program C++

1.3 Tujuan

- Untuk memenuhi tugas dari dosen pada pada kuliah Struktur Data
- Bisa menggunakan/menjalankan program C++ terutama pada Array, Struktur dan Linked List.

1.4 Manfaat

- Supaya kita mengerti atau mengetahui apa yang di maksud dengan Array dan beberapa contoh programnya, sehingga kita dapat menjalankan programnya dengan baik dan benar.
- Supaya kita mengerti atau mengetahui apa yang di maksud dengan Struktur dan beberapa contoh programnya, sehingga kita dapat menjalankan programnya dengan baik dan benar.
- Supaya kita mengerti atau mengetahui apa yang di maksud dengan Linked List dan beberapa contoh programnya, sehingga kita dapat menjalankan programnya

dengan baik dan benar.

Kata Kunci : PENGERTIAN STRUKTUR DATA

PEMBAHASAN

A. PENGERTIAN STRUKTUR DATA

Dalam istilah ilmu komputer, sebuah **struktur data** adalah cara penyimpanan, penyusunan dan pengaturan data di dalam media penyimpanan komputer sehingga data tersebut dapat digunakan secara efisien. Sedangkan **Data** adalah representasi dari fakta dunia nyata. Fakta atau keterangan tentang kenyataan yang disimpan, direkam atau direpresentasikan dalam bentuk tulisan, suara, gambar, sinyal atau simbol.

Konstanta digunakan untuk menyatakan nilai tetap sedangkan variable digunakan dalam program untuk menyatakan nilai yang dapat berubah-ubah selang eksekusi berlangsung.

Ada empat istilah data, yaitu:

1. Tipe data adalah jenis atau macam data di dalam suatu variable dalam bahasa pemrograman.
2. Objek data mengacu kumpulan elemen, D (domain).
3. Representasi data : Suatu mapping dari struktur data 'd' ke suatu set ke struktur data 'e' ($d=e$) misal boolean di representasikan dalam 0 dan 1.
4. Struktur data biasa dipakai untuk mengelompokkan beberapa informasi yang terkait menjadi sebuah kesatuan.

Dalam teknik pemrograman, struktur data berarti tata letak data yang berisi kolom-kolom data, baik itu kolom yang tampak oleh pengguna (*user*) atau pun kolom yang hanya digunakan untuk keperluan pemrograman yang tidak tampak oleh pengguna. Setiap baris dari kumpulan kolom-kolom tersebut dinamakan catatan (*record*). Lebar kolom untuk data dapat berubah dan bervariasi. Ada kolom yang lebarnya berubah secara dinamis sesuai masukan dari pengguna, dan juga ada kolom yang lebarnya tetap. Dengan sifatnya ini, sebuah struktur data dapat diterapkan untuk pengolahan database (misalnya untuk keperluan data keuangan) atau untuk pengolah kata (*word processor*) yang kolomnya berubah secara dinamis. Contoh struktur data dapat dilihat pada berkas-berkas lembar-sebar (*spreadsheet*), pangkal-data (*database*), pengolahan kata, citra yang dipampat (dikompres), juga pemampatan berkas dengan teknik tertentu yang memanfaatkan struktur data.

Secara garis besar type data dapat dikategorikan menjadi:

Type data sederhana.

Type data sederhana tunggal, misalnya Integer, real, boolean dan karakter.

Type data sederhana majemuk, misalnya String

Struktur Data, meliputi:

Struktur data sederhana, misalnya array dan record.

Struktur data majemuk, yang terdiri dari:

- a) Linier : Stack, Queue, serta List dan Multilist

b) Non Linier : Pohon Biner dan Graph

Pemakaian struktur data yang tepat didalam proses pemrograman akan menghasilkan algoritma yang lebih jelas dan tepat, sehingga menjadikan program secara keseluruhan lebih efisien dan sederhana.

Struktur data yang standar yang biasanya digunakan dibidang informatika adalah:

- List linier (Linked List) dan variasinya
- Multilist
- Stack (Tumpukan)
- Queue (Antrian)
- Tree (Pohon)
- Graph (Graf)

B. PEMBUATAN STRUKTUR DATA

Untuk membuat menjadi struktur data, kita harus melakukan dulu aktivitas terhadap objek data, yaitu :

- ❖ Mendeskripsikan kumpulan operasi sah yang diterapkan ke elemen-elemen objek data.
- ❖ Menunjukkan mekanisme kerja operasi-operasi.

Objek data integer ditambah operasi (+, -, *, /, mod, cell, floor, <, >) dan operasi-operasi lain yang memanipulasi objek data integer menyatakan struktur data. Struktur data = Objek data + { Operasi manipulasi }.

Tahap pembuatan struktur data adalah :

- Tahap pertama : Spesifikasi
Pendeskripsian / spesifikasi struktur data menyatakan apa yang dapat dilakukan struktur data, bukan cara penerapannya.

Spesifikasi dapat dilakukan dengan dua cara, yaitu :

- Spesifikasi secara formal
- Spesifikasi secara informal

➤ Tahap kedua : Implementasi

Implementasi menyatakan cara penerapan struktur data dengan struktur data yang telah ada. Implementasi struktur data adalah proses pendefinisian tipe data abstrak sehingga semua operasi dapat dieksekusi computer. Implementasi struktur penyimpanan item-item data serta algoritma-algoritma untuk implementasi operasi-operasi sehingga menjamin terpenuhinya karakteristik struktur data, relasi item-item data atau invariant pada struktur data itu.

➤ Tahap ketiga : Pemrograman

Pemrograman terstruktur adalah penerjemahan menjadi pernyataan di bahasa pemrograman tertentu. Prosesnya terdiri dari :

- Deklarasi yang mendefinisikan objek-objek data dan hubungannya...
- Pembuatan prosedur / rutin untuk operasi-operasi dasar yang menjaga invariant pada struktur data itu .

Sesuai dengan relasi yang didefinisikan di spesifikasi perancangan harus memilih tipe-tipe data yang telah ada untuk merepresentasikan struktur data.

Struktur data di bangun menggunakan fasilitas pembentukan atau pembuatan struktur data yang disediakan bahasa seperti array, record, dan sebagainya atau yang telah di buat seperti stack, queue, atau himpunan menggunakan linked list.

Pembuatan struktur data adalah pembentukan tipe data lengkap yang mempunyai empat property berikut :

1. Nama : Identifier tipe data
2. Domain : Domain / himpunan semesta nilai di tipe data
3. Konstanta (penyebutan anggota-anggotanya) : Cara penyebutan anggota-anggota tipe data
4. Operasi-operasi terhadap tipe data itu (operator) : Daftar operasi terhadap anggota tipe data sehingga kelakuan objek data sesuai spesifikasi.

C. ARRAY

2.1 Pengertian Array

Array atau larik didefinisikan 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.

2.2 Karakteristik Array

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

2.3 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.

2.4 Jenis Array

1. Array Dimensi Satu

Deklarasi : Type_Data Nama_Variabel [index]

Rumus untuk menentukan jumlah elemen dalam array adalah :

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]

n
 π (Index
 Array)
 i = 1

Menentukan jumlah elemen dalam array dimensi dua :

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]

n
 π (Index
 Array)
 i = 1

Menentukan jumlah elemen dalam array dimensi tiga :

π = Perkalian dari statemen sebelumnya

Pemetaan (Mapping) Array Dimensi Tiga Ke Storage

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

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).

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

Dalam array Lower Triangular dengan N baris, jumlah maksimum elemen $\neq 0$, tidak lebih dari

SPERSE ARRAY (ARRAY JARANG)

Suatu array yang sangat banyak elemen nol-nya.

2.5 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

2.6 Penciptaan Dan Penghancuran

Operasi penciptaan biasa disebut inisialisasi. Operasi ini untuk mempersiapkan struktur data untuk operasi-operasi berikutnya. Operasi penghancuran menyatakan ketidak berlakuan 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

2.7 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

2.8 Pemrosesan Transversal

Operasi pemrosesan transversal adalah pemrosesan mengolah seluruh elemen secara sistematis.

2.9 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 sekuensial (sequential searching), yaitu:
  - a. Tanpa Boolean, terbagi:
    - Tanpa sentinen
    - Dengan sentinen
  - b. Menggunakan boolean
- 2) Pencarian secara biner / dikotom (binary = dichotomy searching).

2.10 Pengurutan Array

Pengurutan atau sorting adalah proses yang paling sering dilakukan dalam pengolahan data. Pengurutan dapat dibedakan menjadi dua, yaitu :

- a. Pengurutan internal

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

- b. Pengurutan eksternal

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

2.11 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
  - Menyia-nyiakan 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.

PENUTUP

3.1 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 batasnya maka kita menggunakan Array (type data statis), namun apabila datanya belum kita ketahui batasnya maka gunakan pointer (type data dinamis).

REFERENSI

<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"