

STRUKTUR DATA

JULIO ADISANTOSO
Departemen Ilmu Komputer IPB

Pertemuan 1 : 20 Juni 2016

Permasalahan Struktur Data

- Suatu sistem pengolahan data kependudukan di Indonesia meliputi data pribadi, pendidikan, pekerjaan, dsb. Bagaimana struktur datanya?
- Suatu sistem aplikasi pengolahan data dengan representasi data berbentuk matrik sparse. Bagaimana struktur datanya?
- Suatu sistem mengintegrasikan beberapa sistem lainnya. Bagaimana struktur datanya?
- Isu struktur data semakin penting pada saat:
 - Ukuran data sangat besar
 - Data sangat kompleks
 - Kebutuhan akses data sangat cepat
 - Adanya keterbatasan sumberdaya

Identitas Mata Kuliah

Nama Mata Kuliah	:	Struktur Data
Kode Mata Kuliah	:	KOM207
Koordinator	:	Julio Adisantoso (JAS)
Semester	:	Pendek Genap 2015/2016
Periode Perkuliahan	:	20 Juni 2016
Pengajar Materi UTS	:	Julio Adisantoso (JAS)
Pengajar Materi UAS	:	Auzi Asfarian (AAS)
Pengajar Praktikum	:	HKH dan VDE
Jadwal Kuliah	:	Senin dan Selasa, 19:00-20:40
Jadwal Praktikum	:	Sabtu, 13:00-17:00

Penentuan Nilai Akhir

- Praktikum : 10%
- Tugas dan Kuis : 5%
- Keaktifan : 5%
- UTS dan UAS Tertulis : 60%
- UTS dan UAS Praktikum : 20%
- Catatan: **Tidak ada ujian perbaikan**

Materi UTS

Minggu	Kompetensi Dasar	Materi Ajar	%
1	Mahasiswa dapat menjelaskan abstraksi data dan representasi struktur data	Pendahuluan, ADT, dan Konsep Struktur Data	5
2-5	Mahasiswa dapat mengimplementasikan struktur data dinamis	Linked List dan aplikasi struktur data linier	15
6-7	Mahasiswa dapat mengimplementasikan struktur data sekuensial	Struktur data sekuensial	10
Total Bobot Nilai UTS			30

Materi UAS

Minggu	Kompetensi Dasar	Materi Ajar	%
08-11	Mahasiswa dapat memahami dan menjelaskan akan dapat menguasai struktur data berhirarki	Pengenalan struktur data pohon dan turunan-turunannya serta metode penelusuran pohon	15
12-13	Mahasiswa dapat menjelaskan struktur data graf dan aplikasinya	Struktur data graf	10
14	Mahasiswa dapat mengimplementasikan konsep hashing	Hashing	5
Total Bobot Nilai UAS			30

Materi Praktikum

Minggu	Topik	Bahan Kajian
1	Set	Pendahuluan, Konsep Struktur Data
2	Struct, Array, Vektor	Linked List, struktur data linier
3-5	List, Sort	Rekursifitas dan Metode Pengurutan Data Lanjut.
6	Stack dan Queue	Struktur data sekuensial
7	UTSP	Ujian Tengah Semester Praktikum
08-10	Tree	Traversal, priority queue, BST, AVL, B-Tree, Trie
11-12	Graf	Representasi graf, BFS, DFS, shortest path, MST
13	Hash Map	Hashing
14	UASP	Ujian Akhir Semester Praktikum

Perangkat Perkuliahan

- Peserta: Mahasiswa ILKOM Alih Jenis
- Bahan Materi: Data Structures and Algorithm Analysis in C - Mark Allen Weiss (Addison Wesley)
- Situs latihan dan praktikum: <http://apps.cs.ipb.ac.id/lx>
- Site Material Elektronik (resources) lms.ipb.ac.id (KOM207-AJ)
- Bahasa Pemrograman : C++ (OOP Based)

Tata Tertib

- Kehadiran Paling lambat 15 menit setelah dosen masuk kelas/lab
- Berpakaian sesuai ketentuan TaTib IPB
- Minimum kehadiran 70% masing-masing untuk kuliah dan praktikum (syarat untuk UAS)
- Handphone silahkan dinonaktifkan (*silent*), no chatting/BBM/FB/etc saat kuliah maupun praktikum

Kejujuran Akademik

- Setiap **KECURANGAN** akan diberikan imbalan nilai 0 pada mata kuliah ini
 - Menyontek ataupun bekerja sama pada saat ujian
 - Menyalin tugas hasil pekerjaan pihak lain
 - Titip tanda tangan kehadiran
- Imbalan (sanksi) akan diberikan untuk si pelaku maupun yang memberikan kesempatan

ADT and Data Structure

ADT : Abstract Data Type

- ADT adalah definisi dari **TYPE** dan sekumpulan operasi dasar (**PRIMITIF**) dari TYPE tersebut
- Definisi **TYPE** dari sebuah ADT dapat mengandung definisi ADT lainnya. Contoh:
 - ADT **WAKTU** terdiri atas ADT **JAM** dan ADT **DATE**
 - ADT **GARIS** memiliki dua buah **TITIK**
- **TYPE** diterjemahkan menjadi *data type* yang terdefinisi sesuai bahasa pemrograman, misalnya `struct` dalam C, `record` dalam Pascal, `class` dalam C++/Java
- **PRIMITIF**, dalam konteks prosedural, diterjemahkan sebagai fungsi atau prosedur

Primitif

PRIMITIF dikelompokkan menjadi:

- *Constructor/Creator*, pembentuk nilai awal (inisialisasi). Biasanya diawali dengan **make**.
- *Selector/Accessor*, untuk mengakses komponen ADT. Biasanya diawali dengan **get**.
- *Mutator*, prosedur pengubah nilai komponen. Biasanya diawali dengan **set**.
- *Validator*, penguji apakah nilai type sesuai batasan.
- *Destructor/Deallocator*, untuk menghapus nilai obyek, sekaligus lokasi memori penyimpanannya
- ... *goto next slide*

Primitif

PRIMITIF dikelompokkan menjadi (lanjutan):

- *Read/Write*, untuk antar-muka dengan *input/output devices*.
- *Relational Operator*, untuk mendefinisikan hubungan relasi lebih besar, lebih kecil, dsb.
- *Arithmetic*, karena biasanya operasi aritmatika hanya terdefinisi untuk bilangan numerik, bukan obyek sesuai ADT yang ada.
- *Convert*, untuk konversi ke tipe data dasar, atau sebaliknya

Implementasi ADT

- ADT biasanya diimplementasikan menjadi dua modul:
 - Definisi/Spesifikasi dari TYPE dan PRIMITIF
 - Spesifikasi TYPE sesuai bahasa
 - Spesifikasi PRIMITIF sesuai konteks (fungsi ataukah prosedur)
 - Body, berupa kode program
- Supaya ADT dapat diuji tuntas, maka dalam kuliah ini harus dilengkapi dengan program utama yang mengandung pemakaian (*call*) terhadap setiap PRIMITIF dalam ADT. Disebut sebagai **DRIVER**.

Realisasi ADT

Realisasi ADT dalam beberapa bahasa pemrograman

BAHASA	SPESIFIKASI	BODY
Pascal	Unit <i>interface</i>	Implementasi
C	File <i>header</i> *.h	File kode program (*.c)
C++	File <i>header</i> *.h	File kode program (*.cpp)
Java	Class	Public Class

Standard Kuliah untuk Implementasi ADT

- 1 Setiap ADT harus dibuat menjadi spesifikasi, body, dan driver
- 2 Dalam bahasa C++, modul spesifikasi dan body dapat dibuat dengan cara:
 - di-*include* dari file *header* yang ada
 - di-*encapsulate* dalam *class*
- 3 Driver digunakan untuk menguji ADT dan sebagai jawaban atas persoalan yang diberikan

Contoh Implementasi ADT

```
// SPESIFIKASI dan BODY
#include <iostream>
using namespace std;
class Stack {
    .....
    .....
}

// DRIVER
int main() {
    .....
    .....
}
```

Struktur Data

- Sering terjadi salah pengertian antara ADT dan Struktur Data.
- Struktur data lebih kongkret, merupakan teknik atau strategi untuk mengimplementasikan sebuah ADT (ADT lebih merupakan deskripsi logika)
- Struktur data merupakan cara membentuk, mengkonstruksi, mengaransemen, mengkomposisikan ataupun mengorganisasikan data (ADT)
- ADT: stack, queue, priority queue, dictionary, sequence, set
- Struktur Data: array, linked list, hash table (open, closed, circular hashing), trees (binary search trees, heaps, AVL trees, 2-3 trees, tries, red/black trees, B-trees)

Contoh Kasus

Diketahui kumpulan bilangan bulat (1 s/d 1000) sebanyak maksimum 10^7 bilangan. Selanjutnya, buat program untuk mencari apakah suatu nilai bilangan ada ataukah tidak ada di dalam kumpulan data tersebut.

Contoh input

```
5 9 8 7 6 8 7 6 2 3 4 5 1 -9  
4 7 9 10 3
```

Contoh output

```
7 ada  
9 ada  
10 tidak ada  
3 ada
```

Solusi (p01.cpp)

- Alternatif 1 (indeks array terlalu besar, error)
 - 1 Baca semua data dan masukkan ke dalam array
 - 2 Sort
 - 3 Search
- Alternatif 2 (optimum)
 - 1 Karena nilai data bulat 1-1000, maka buat indeks array sebanyak 1000 (0-999)
 - 2 Baca data, dan isi nilai 1 ke array dengan $\text{indeks} = \text{data} - 1$
- Bagaimana kalau bilangan yang diketahui adalah riil (bukan bulat)? Tidak dapat menggunakan Alternatif 2.

Solusi (p01b.cpp)

- Pada kasus ini, duplikasi bilangan tidak berpengaruh, sehingga sangat baik kalau kumpulan bilangan tersebut direpresentasikan sebagai suatu himpunan (*set*).
- Oleh karena itu, dapat menggunakan ADT **set**