

STRUKTUR DATA -> ANALOGI LEMAR

nah kan banyak tuh dia ada yang di hanger di laci dll

terus kalo misalnya kita mau ambil atasan ambil dr atas dll  
kan beda2 nah itu yang kita pelajari

misal kalo berangkat kerja kan ambil baju, itu berarti remove baju

running time itu nanti ada Big Oh

struktur data itu banyak bgt jenisnya (yg buat sda lumayan berat)



# CSGE602040 • Struktur Data & Algoritma: Pengantar

di setiap list nanti beda2 juga aturannya

**Fakultas Ilmu Komputer • Universitas Indonesia**

*Slide acknowledgments:*

Suryana Setiawan, Ade Azurat, Denny, Ruli Manurung, Tisha Melia



# Tujuan Mata Kuliah

- Mempelajari dasar-dasar ilmu komputer agar dapat melakukan
  - perancangan dan pemilihan struktur data yang sesuai,
  - implementasi, dan
  - melakukan analisis secara umum pada algoritma yang dibuat.
- Meningkatkan ketrampilan pemrograman
  - Skala lebih besar, lebih efisien dan elegan.
  - *“Programming to an interface”*
  - Prinsip-prinsip dasar RPL: abstraksi, modularitas, dst.



# Arti kata (Webster)

- **da•ta (n.pl.)**

1. facts or figures to be processed; evidence, records, statistics, etc. from which conclusions can be inferred; information

- **struc•ture (n.)**

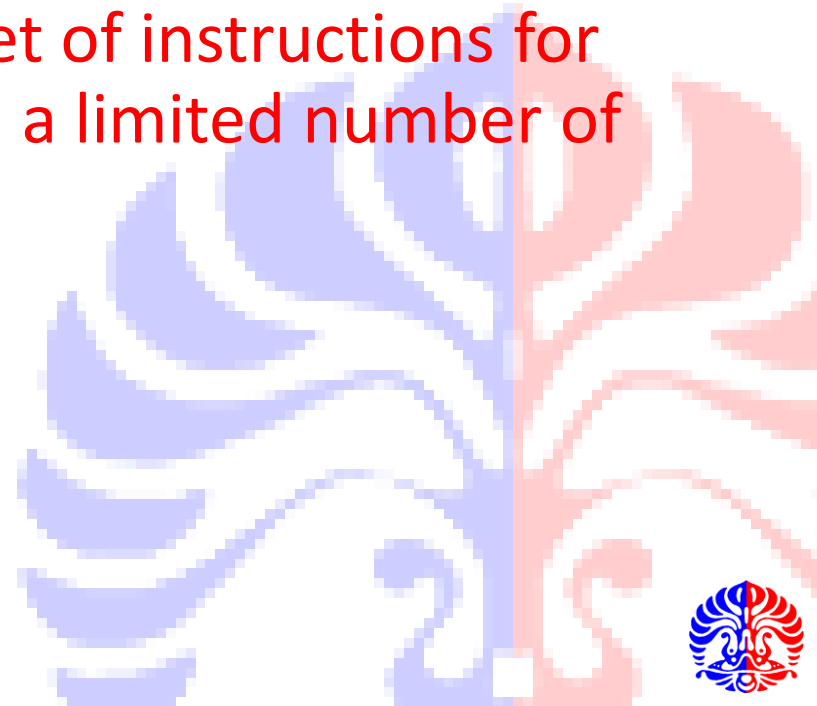
1. manner of building, constructing, or organizing
2. something built or constructed, as a building or dam
3. the arrangement or interrelation of all the parts of a whole; manner of organization or construction [**the structure of the atom, the structure of society**]
4. something composed of interrelated parts forming an organism or an organization



# Arti kata (Webster)

- **al•go•rithm (n.)**

1. *Math.* a) any systematic method of solving a certain kind of problem b) the repetitive calculations used in finding the greatest common divisor of two numbers (called in full Euclidean algorithm)
2. *Comput.* a predetermined set of instructions for solving a specific problem in a limited number of steps



# Struktur Data

- Semua program berurusan dengan data
  - Sistem informasi: informasi, laporan, user, ...
  - Game: posisi & status pemain, musuh, skor, ...
  - Search engine: URL, isi, hyperlink, bobot, ...
- Mengapa data itu disimpan?
  - Supaya bisa diakses/diproses di kemudian waktu
- Mengapa dalam penyimpanan data diperlukan sebuah struktur?
  - Supaya lebih mudah/efisien dalam pengaksesan/pemrosesan data tersebut



# Fungsi Struktur Data

- Sebagai Look-up Table: struktur dibentuk untuk memungkinkan pencarian data dengan tingkat efisiensi yang baik
  - data disusun secara hirarkis dalam B+Tree, tapi data itu sendiri tidak memiliki hubungan semantic yang hirarkis tersebut.
- Sebagai Representasi Semantik dari Data: struktur dibentuk sesuai dengan keterkaitan satu data dengan data lainnya.
  - silsilah keluarga dalam tree karena hubungan semantik hirarkis antar data.
  - Peta jalan dalam graph karena hubungan semantik  $n$  ke  $n$  antar kota.



# Apa Ruang Lingkup Kuliah Struktur Data?

- Struktur Data Linier
  - Array (unsorted, sorted), Linked-list, Stack, Queue, Hash Tables
- Struktur Data Hirarkis
  - Tree, Binary Tree, Binary Search Tree, B<sup>+</sup>Tree
- Struktur Data Graph
  - Graph dengan varian-varian: Edge-list, vertex-list, adjacent list, adjacent matrix.



# Mengapa kuliah ini penting?

Apakah kuliah DDP saja tidak cukup?

Perhatikan program memeriksa apakah data X ada dalam array C yang isinya tidak teratur (sequential searching):

```
Int C[] = {65, 84, 90, 43, 34, 19, 23}; // data tak teratur
boolean cariVersi1(int X) {
    for (int i = 0; i < 7; i++)
        if (C[i] == X) return true;
    return false;
}
```

- Jika X ada dalam C, program dieksekusi dalam jumlah iterasi 1 sd. 7 (rata-rata 4 iterasi).
- Jika X tidak ada dalam C, program dieksekusi selalu dalam jumlah 7 iterasi.





# Mengapa kuliah ini penting? (2)

Perhatikan program mencari apakah data X ada dalam array C yang isinya terurut (sequential searching):

```
Int C[] = {19, 23, 34, 43, 65, 84, 90}; // data terurut
boolean cariVersi2(int X) {
    for (int i = 0; i < 7; i++)
        if (C[i] == X) return true;
        else if (C[i] > X) return false;
    return false;
}
```

- Jika X ada dalam C, jumlah kemungkinan iterasi 1 sd. 7 (rata-rata 4 iterasi).
- Jika X tidak ada dalam C, jumlah kemungkinan iterasi 1 sd. 7 (rata-rata 4 iterasi).



# Mengapa kuliah ini penting? (3)

- Untuk 7 data perbedaan tampak tidak berarti, untuk 1 juta data perbedaan 1 juta iterasi terhadap  $\frac{1}{2}$  juta iterasi.
- Lebih berarti lagi jika pencarian dilakukan dengan algoritma Binary Searching dalam array C yang terurut.
- Ide Memeriksa X pada ruas array C[left] s.d. C[right]:
  - prekondisi:  $\text{left} < \text{right}$
  - periksa X terhadap C[mid] dengan  $\text{mid} = (\text{left} + \text{right}) / 2$ ,
  - if  $X == C[\text{mid}]$  return true;
  - else if  $X < C[\text{mid}]$  periksa X thd ruas array C[left] s.d. C[mid-1];
  - else periksa X thd ruas array C[mid+1] sampai dengan C[right];
- Untuk ukuran data 1 juta, iterasi hingga diketahui X terdapat dalam C atau tidak, paling banyak 20 kali.

→ dia udh berurut gitu



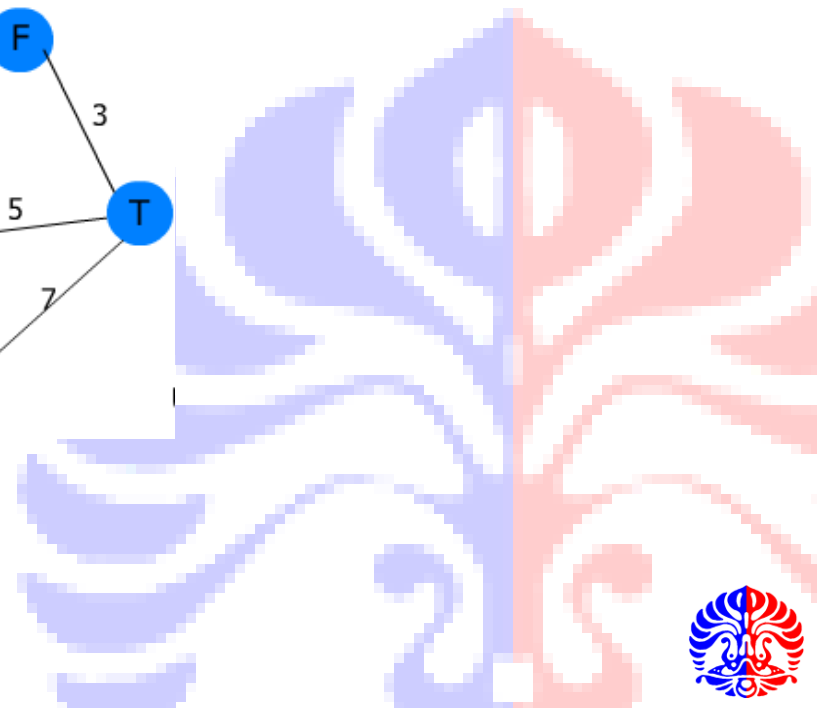
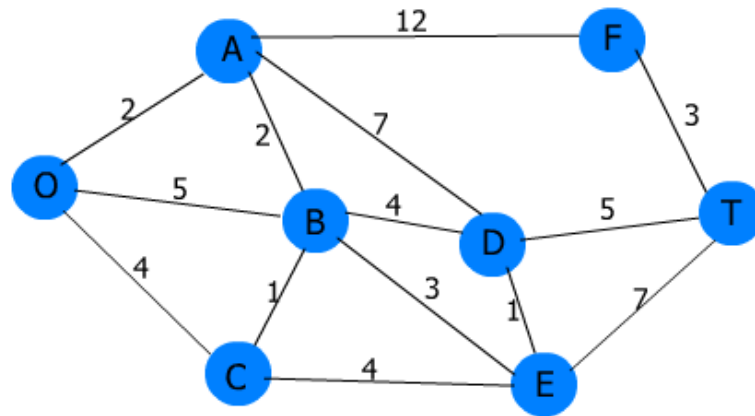
# Mengapa kuliah ini penting? (4)

- Perbandingan ketiga solusi (untuk N data):
  - Dengan struktur data **array tak terurut** iterasi sebanyak N kali.
  - Dengan struktur data **array terurut** tapi menggunakan **sequential searching** iterasi dilakukan sebanyak N/2 kali.
  - Dengan struktur data **array terurut** tapi menggunakan **binary searching** iterasi dilakukan sebanyak  $\log_2(N)$  kali.
- Moral of the story:
  - Pemilihan **struktur data** maupun **algoritma** yang tepat dapat membuat program lebih efisien



# Mengapa kuliah ini penting? (5)

- Kegunaan nyata:
  - Google Map, Waze, Gojek, Grab, dll, salah satu fungsinya mencari rute terpendek antara posisi. Bagaimana data itu disimpan? Struktur data Graph



# Mengapa kuliah ini penting? (6)

- Contoh Aplikasi lain:
  - Sistem basis data (Oracle, SQL Server, dll)
    - Engine menggunakan struktur data B+Tree, Hashtable
  - Menghitung ekspresi:  $(5 + 2) * 7$ 
    - Interpreter menggunakan struktur data Stack/Tree saat melakukan parsing
  - Aplikasi permainan (arcade game)
    - Engine menyimpan dalam graph untuk mengelola “dunia” virtualnya



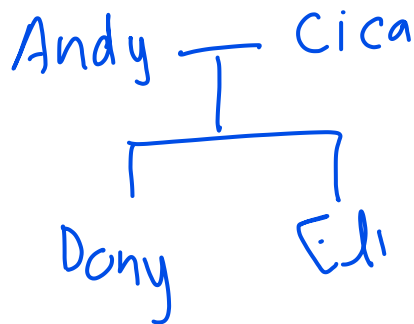
# Mengapa Belajar *Implementasi* Struktur Data?

- Mengetahui kelebihan dan kekurangan dari masing-masing struktur data.
- Cara yang terbaik untuk benar-benar dapat memahami masing-masing struktur data adalah membuatnya.
- Menyesuaikan struktur data yang ada untuk problem baru (*augmented data structure*)
- Dalam industri, bahasa yang digunakan tidaklah selalu Java. Mungkin saja di bahasa tersebut tidak terdapat library untuk struktur data.
- Melatih berpikir tentang efisiensi



# Struktur Data & Rekayasa Perangkat Lunak

- Struktur data memudahkan kita untuk *component reuse*.
  - Sekali kita implementasi, dapat digunakan berkali-kali dalam aplikasi yang berbeda



→ pake tipe tree aja,  
misal pake list biasa  
dia kan ga keruan



# Summary

- Struktur data + Algoritma = Program
- Pemilihan struktur data dan algoritma yang tepat dapat membuat program lebih efisien, mudah, dan elegan

