

# Jenis struktur data & ADT umum

IF1210 – Algoritma dan Pemrograman 1  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung

# Jenis struktur data umum

## ***record*** atau ***tuple***

- Agregasi beberapa nilai dengan jumlah yang tetap.
- Setiap nilai dapat berbeda tipe.
- Elemennya diakses dari namanya.

## ***array***

- Sejumlah elemen yang diletakkan di memori secara kontigu.
- Setiap elemen bertipe data sama.
- Elemennya diakses melalui indeks.

## ***node-based*** atau ***linked*** (struktur berkait)

- Sejumlah elemen yang saling terhubung melalui *pointer*.
- Setiap elemen bertipe data sama.
- Elemen diacu oleh elemen sebelumnya.

# Record/tuple

Struktur data yang digunakan dalam ADT Time alt-1 adalah contoh **tuple**.

- Merupakan agregasi dari 3 nilai: **jam**, **menit**, dan **detik**.
- Masing-masing nilai dapat berasal dari tipe logik yang berbeda (**jam** hanya 0-23, sedangkan **menit** dan **detik** 0-59).
- Diakses berdasarkan namanya (misal: **t.hours**).

Dituliskan dengan tanda ‘<’ dan ‘>’, contoh:

**Time** dapat dituliskan sebagai **tuple** **<hours, minutes, seconds>**

# Array

*Array* harus dialokasikan di memori dengan **ukuran yang sudah ditentukan**.

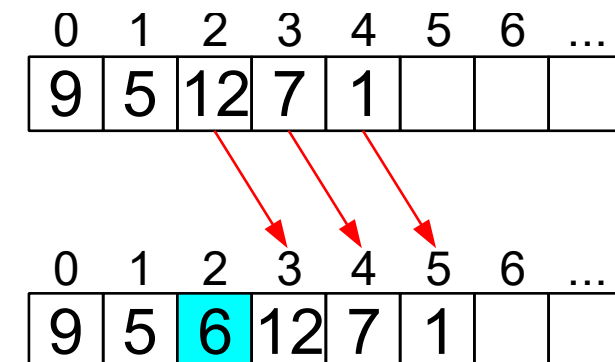
Misal: *array* 10 elemen bertipe integer → dialokasikan memori sebesar  $10 \times 4$  byte.

Jika perlu mengubah ukuran *array*:

1. buat *array* baru dengan ukuran yang dikehendaki,
2. salin isi *array* lama ke *array* baru,
3. dealokasi *array* lama.

Akses ke setiap elemen *array* dilakukan melalui **indeks** (biasanya dimulai dari 0).

Menambah/menghapus elemen di tengah *array* mengakibatkan elemen-elemen setelah posisi tersebut harus  **digeser satu per satu**.



# Struktur berkait

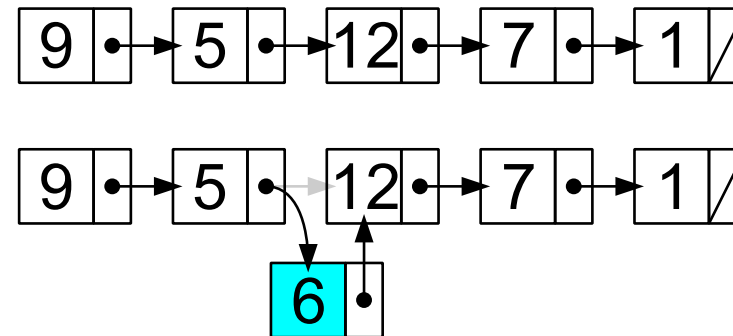
Sebuah *node* adalah **tuple** dengan dua elemen:

1. **Nilai** yang hendak disimpan dalam *node*,
2. **Pointer** ke *node* berikutnya.

Di memori, lokasi (alamat) *node* satu dan *node* berikutnya tidak harus bersebelahan.

Akses ke suatu *node* harus **melalui node-node sebelumnya**.

Menambah/menghapus elemen di tengah struktur berbasis *node* cukup dengan “**rerouting**” pointer-pointer.



# Pakai yang mana?

Beberapa jenis data dapat distrukturkan dengan lebih dari satu cara. Contoh:

- Time bisa saja distrukturkan dalam **array** berukuran 3, sehingga bagian jam diakses dengan  $t[0]$ , menit dengan  $t[1]$ , dan detik dengan  $t[2]$ .  
→ Konsekuensi: ketiga nilai tersebut kini bertipe sama (e.g., integer).
- Sebuah antrian (elemen yang baru datang harus “berdiri” di belakang, dan elemen yang boleh “dilayani” hanya elemen yang paling depan) dapat distrukturkan dalam bentuk **array** maupun **node-based**.

Pemilihan struktur data berakibat pada *trade-off* **efisiensi algoritma** pada berbagai operasi.

Diperlukan **analisis algoritma** untuk memilih struktur yang lebih efisien.

# ADT umum yang akan dibahas

List/sequence

Matriks

Stack

Queue

Set,  
Multiset

Map

Tree,  
binary tree,  
binary search tree

Graph