

Representasi Fisik List Linier: Struktur Berkait dengan Array

IF2110/IF2111 – Algoritma dan Struktur Data
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Kekurangan struktur berkait

Pada pembahasan implementasi ADT List menggunakan struktur berkait, setiap Node dialokasikan satu demi satu.

- Persoalan: alokasi & dealokasi memori adalah operasi yang “mahal” pada sistem operasi.
- Akan lebih efisien jika dapat dilakukan alokasi beberapa Node sekaligus.

Persoalan lain: bahasa pemrograman yang digunakan mungkin tidak mendukung pointer.

Alternatif: array of Node

Banyak Node dialokasi dengan satu kali pemanggilan ke sistem operasi, dalam bentuk array.

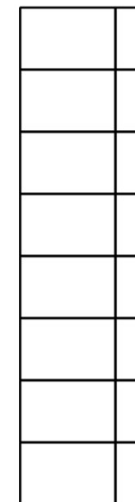
Bagian Next dari Node kini bukan mengacu pada alamat fisik memori melainkan indeks array.

Array of Node dapat dideklarasikan secara global untuk digunakan oleh beberapa List sekaligus.

Node



array of Node



Array of Node

Saat inisialisasi, bagian Next setiap Node diisi dengan indeks elemen array berikutnya (`nodeArray[i].next = i+1`).

Untuk Node terakhir, diisi dengan indeks yang tidak valid (konstanta, misal -1).

Diperlukan sebuah pencatat Node pertama yang kosong.

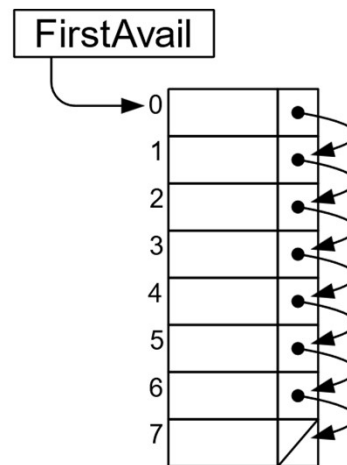
Saat inisialisasi, diisi dengan indeks pertama array (0).

Ilustrasi: setelah inisialisasi

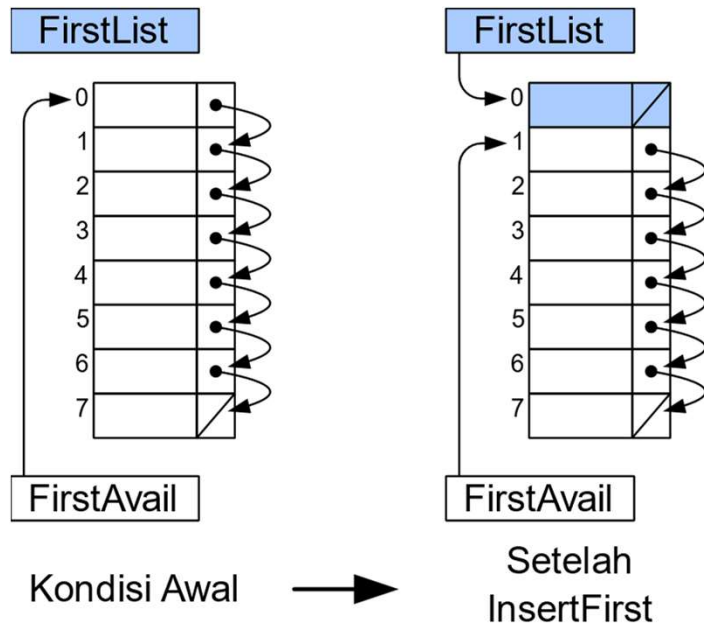
FirstAvail = 0

| | | |
|---|--|----|
| 0 | | 1 |
| 1 | | 2 |
| 2 | | 3 |
| 3 | | 4 |
| 4 | | 5 |
| 5 | | 6 |
| 6 | | 7 |
| 7 | | -1 |

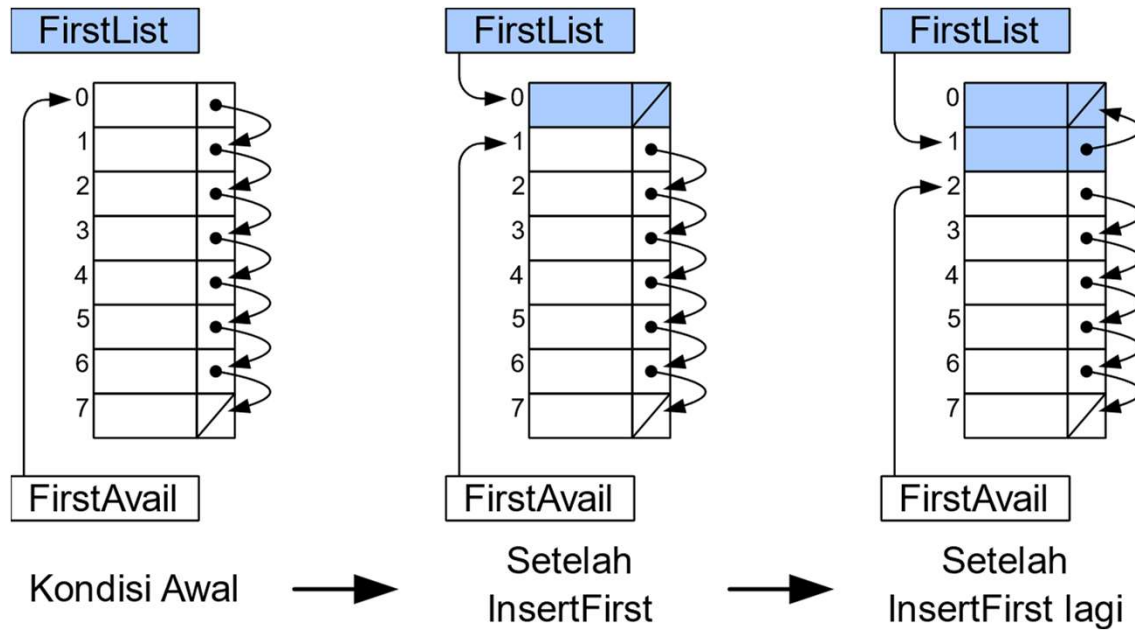
atau



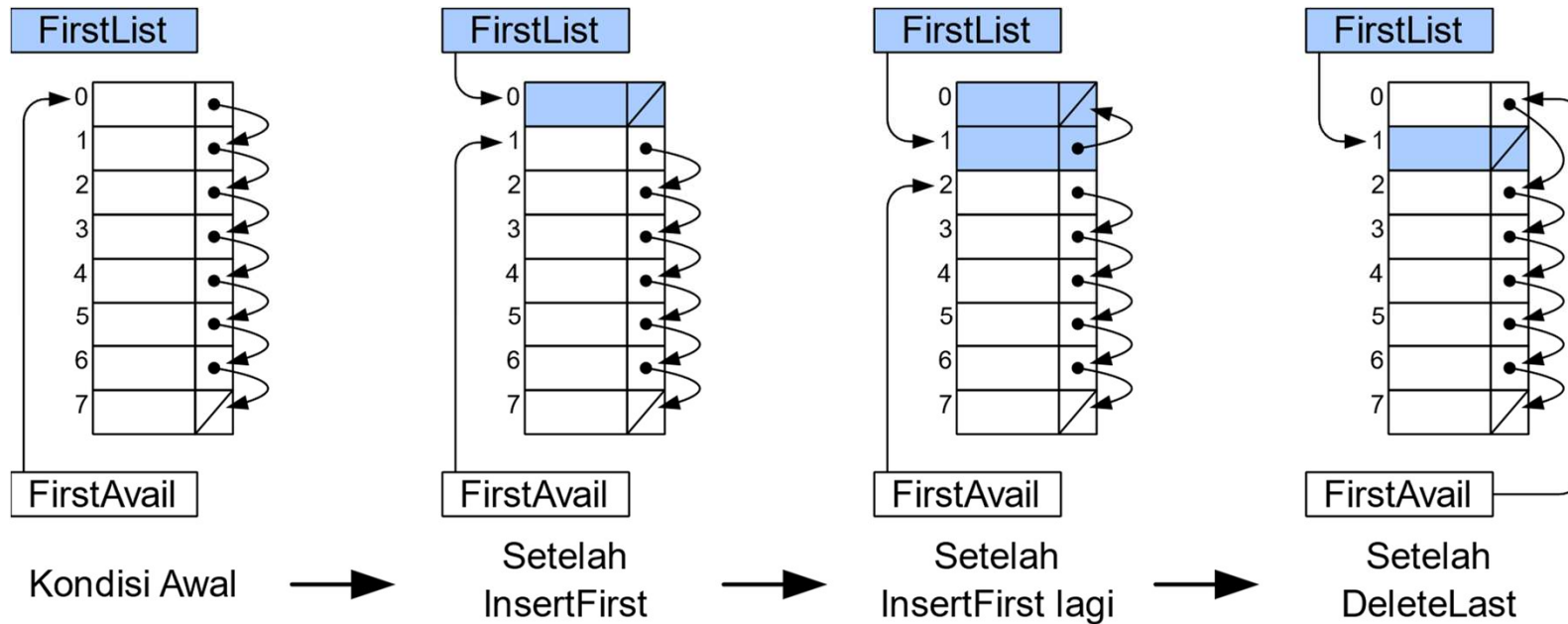
Ilustrasi: pemakaian memori list



Ilustrasi: pemakaian memori list



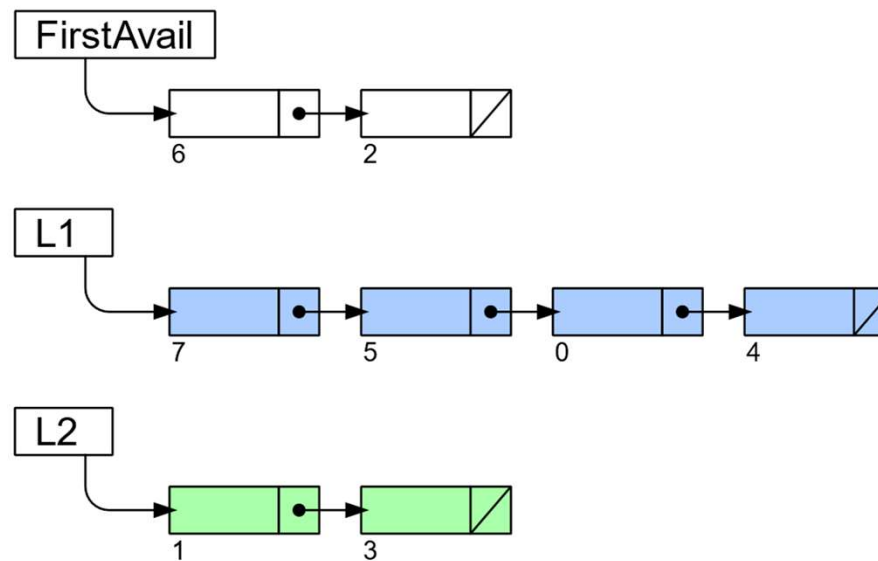
Ilustrasi: pemakaian memori list



Ilustrasi: array digunakan oleh dua list

FirstAvail = 6
First L1 = 7
First L2 = 1

| | | |
|---|--|---|
| 0 | | 4 |
| 1 | | 3 |
| 2 | | / |
| 3 | | / |
| 4 | | / |
| 5 | | 0 |
| 6 | | 2 |
| 7 | | 5 |



Indeks yang kosong membentuk sebuah Stack!

