

# LINKED LIST

DAFTAR BERTAUT/SENARAI BERANTAI/SENARAI BERTAUT\*

# KELOMPOK 3

FAHMI EKA P.

SITI JUBAEDAH

ARKYANA USMAN DWI B.

# LINKED LIST

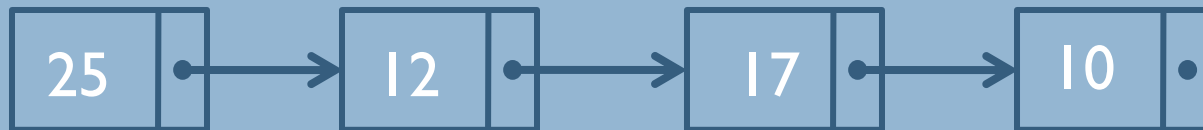
Sejumlah objek yang *dilink* atau dihubungkan satu dengan yang lainnya sehingga membentuk suatu list

## LALU, APA ITU OBJEK?

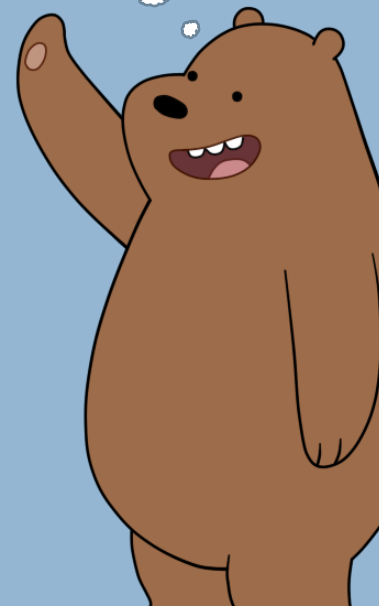
Gabungan beberapa elemen data (variabel) yang dijadikan satu kelompok atau satu structure atau record yang terbentuk dengan perintah struct

# LINKED LIST

Linked list dapat diilustrasikan sebagai berikut :

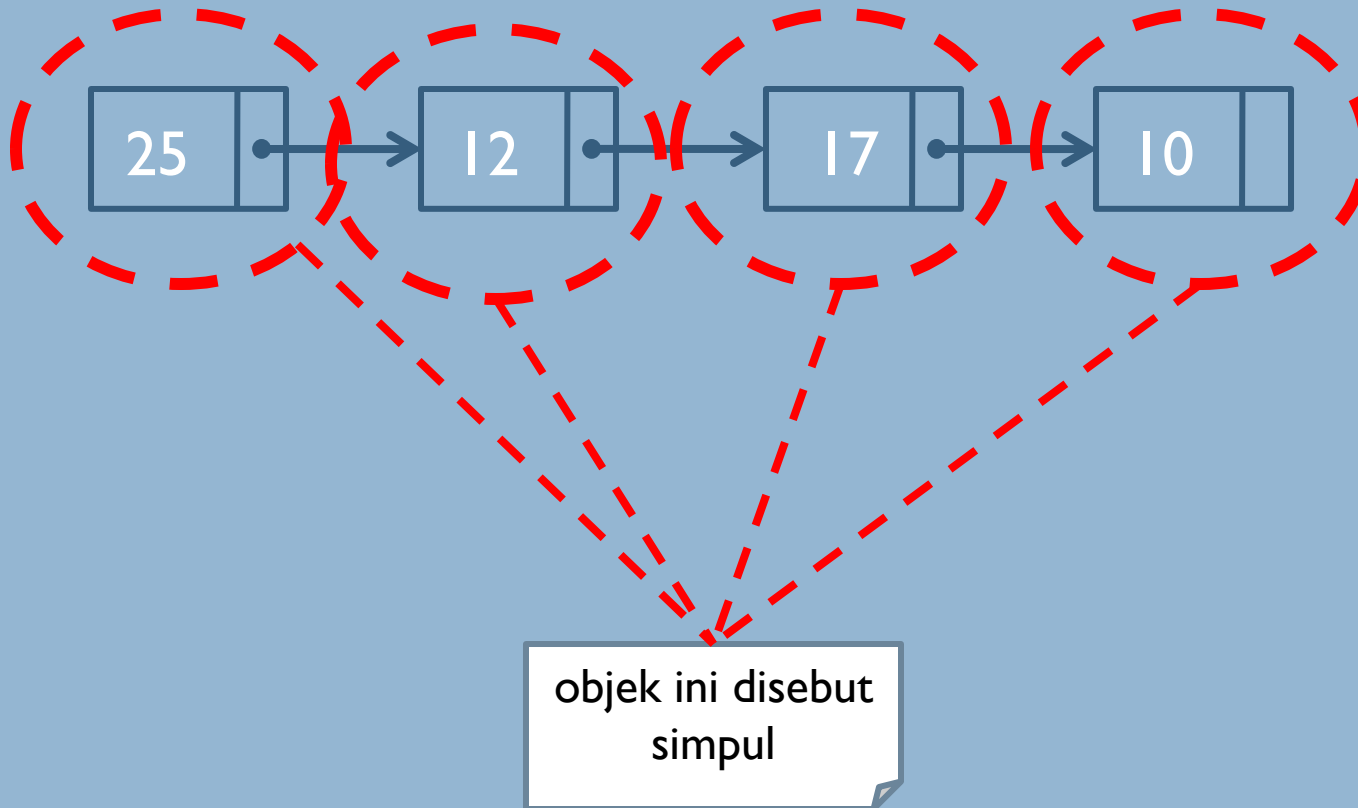


*Ini apa  
ya???*



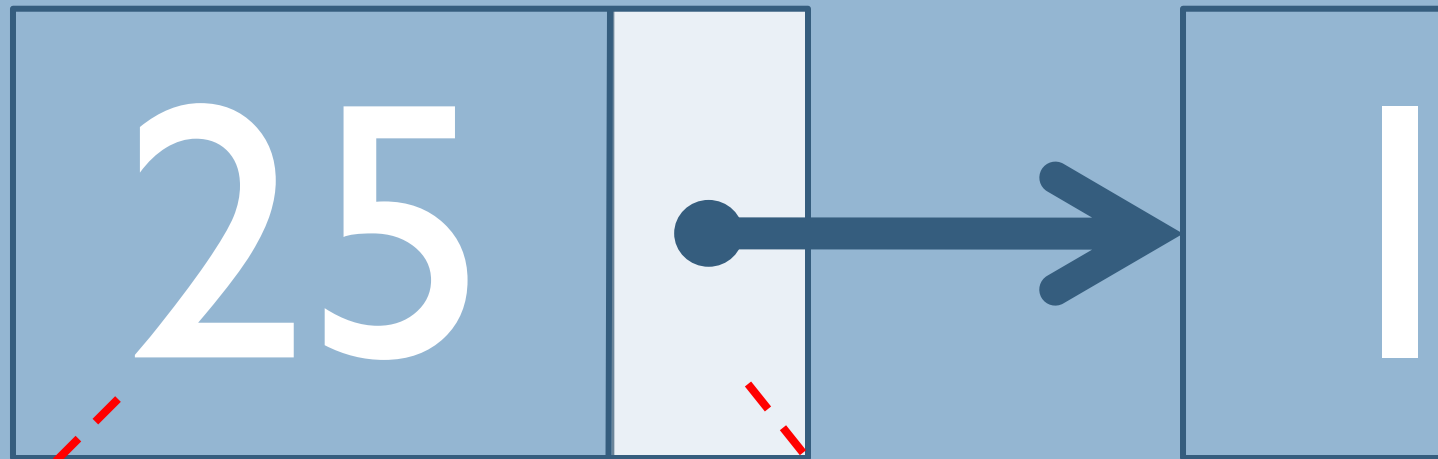
# LINKED LIST

Okee, mari kita perjelas.....



# LINKED LIST

di dalam simpul terdapat 2 elemen

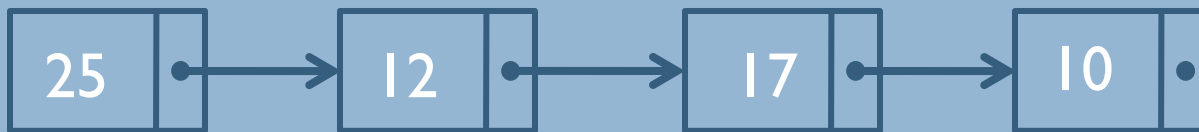


Elemen pertama berisi data yg dicontohkan dengan nilai numerik

Elemen kedua yang bertipe pointer

# LINKED LIST

Linked list dapat diilustrasikan sebagai berikut :



Yang artinya :

Ada 4 buah simpul (objek) yang ter-link atau terhubung

Dalam Penggunaan Linked List, terdapat 4 macam proses dasar, yaitu :

1. Inisialisasi, yaitu persiapan pembuatan linked list.
2. Membuat simpul awal
3. Membuat simpul baru dan menambahkannya (insert) kedalam linked list.
4. Menghapus (mendelete) simpul dari linked list



Dilihat dari fungsi-fungsi yang digunakan untuk proses, maka keempat proses dasar sebelumnya, dapat dibagi lagi menjadi 9 macam proses

## 9 MACAM PROSES BERDASARKAN FUNGSINYA

1. Inisialisasi linked list  
Yaitu membuat kondisi awal yang menyatakan linked list belum ada dengan cara mengisi pointer FIRST dengan NULL
2. Pembutan sebuah simpul
3. Pembuatan simpul awal
4. Insert kanan (insert akhir)  
Yaitu menginsert sebuah simpul baru di ujung paling kanan, atau di akhir
5. Insert kiri (Insert awal)  
Yaitu menginsert sebuah simpul baru di ujung paling kiri, atau di awal

## 9 MACAM PROSES BERDASARKAN FUNGSINYA

### 6. Insert Tengah

Yaitu menginsert sebuah simpul baru antara dua buah simpul yang sudah ada

### 7. Delete Kiri

Yaitu menghapus simpul yang ada di ujung paling kiri atau simpul awal

### 8. Delete Kanan

Yaitu menghapus simpul yang ada ujung paling kanan atau simpul akhir

### 9. Delete Tengah

Yaitu menghapus sebuah simpul yang berada diantara dua buah simpul lain

# **4 MACAM STRUKTUR LINKED LIST\***

**I. LINEAR SINGLY-LINKED LIST**

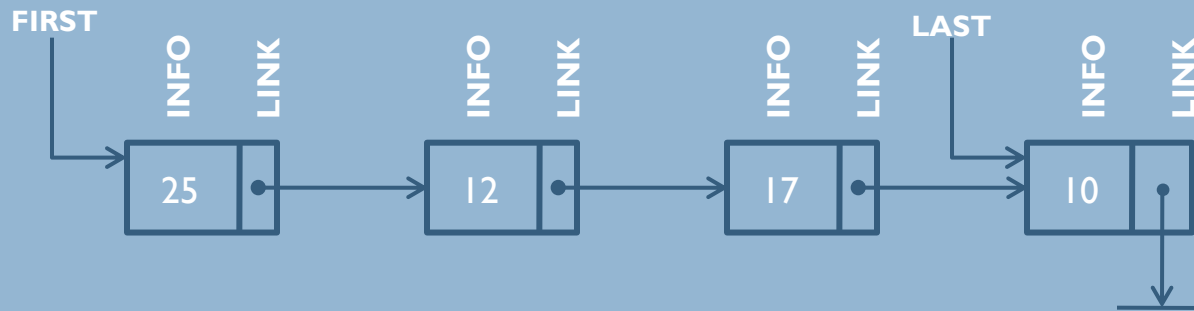
**II. LINEAR DOUBLY-LINKED LIST**

**III. CIRCULAR SINGLY-LINKED LIST**

**IV. CIRCULAR DOUBLY-LINKED LIST**

# STRUKTUR LINKED LIST

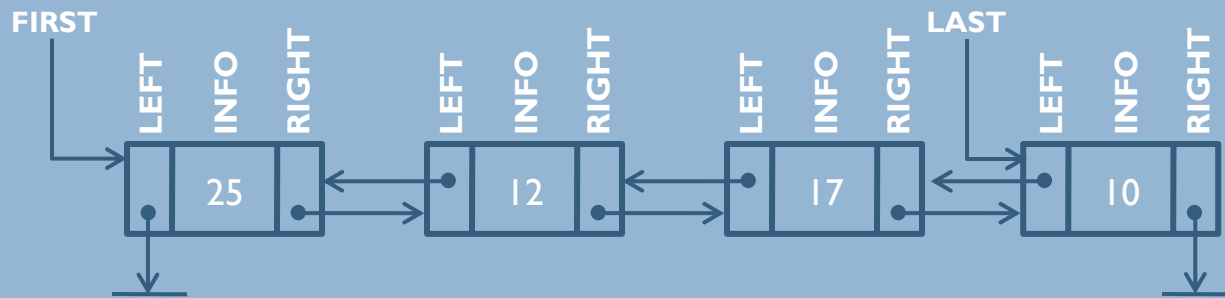
## LINEAR SINGLY-LINKED LIST



- Linear Singly-Linked List maksudnya adalah Linked List Lurus dengan pointer tunggal.
- Jadi dalam satu struktur simpul hanya ada satu elemen atau field atau variabel yang bertipe pointer yang isinya adalah alamat simpul berikutnya atau next node

# STRUKTUR LINKED LIST

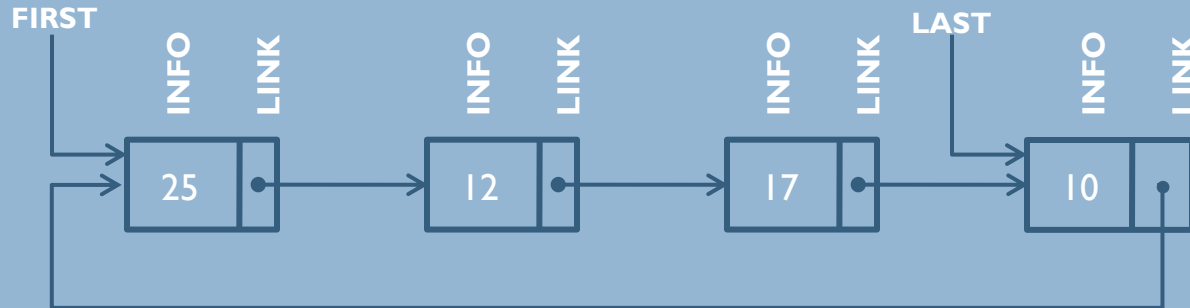
## LINEAR DOUBLY-LINKED LIST



- Linear Doubly-Linked List maksudnya adalah Linked List Lurus dengan pointer ganda, yaitu ada dua buah pointer.
- Jadi dalam struktur simpul ada dua elemen atau field atau variabel yang bertipe pointer.
- Yang pertama menunjuk atau berisi alamat sebelumnya atau previous node.
- Yang kedua menunjuk simpul berikutnya atau next node.

# STRUKTUR LINKED LIST

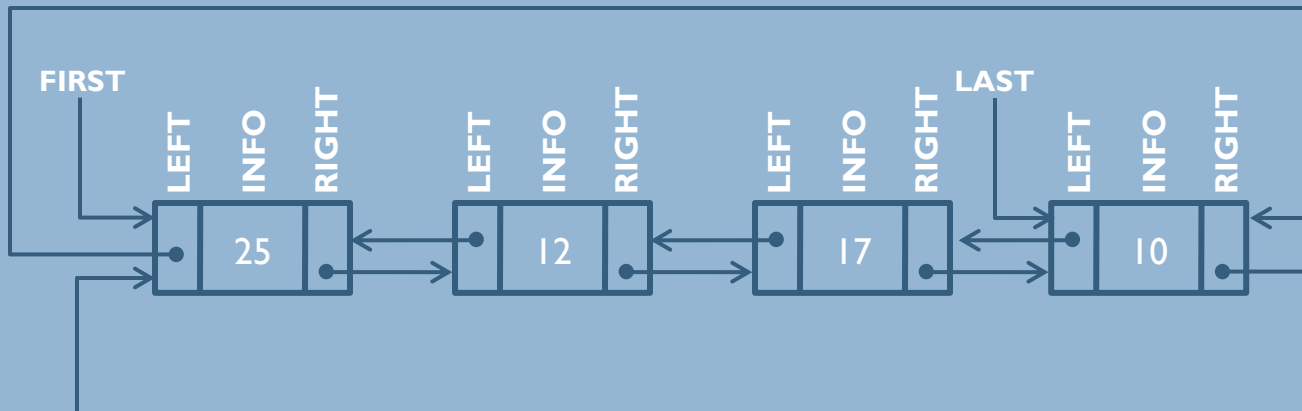
## CIRCULAR SINGLY-LINKED LIST



- Circular Singly Linked List adalah Singly Linked List dimana link simpul terakhir bukan diisi dengan null,
- Melainkan diisi dengan alamat simpul pertama yaitu simpul yang ditunjuk oleh pointer FIRST
- Sehingga menciptakan efek melingkar 'sesuai arah jarum jam'

# STRUKTUR LINKED LIST

## CIRCULAR DOUBLY-LINKED LIST



- Circular Doubly Linked List adalah adalah Doubly Linked List dimana pointer RIGHT simpul paling kanan berisi alamat simpul paling kiri,
- Pointer LIFT simpul paling kiri berisi alamat paling kanan,
- Sehingga menciptakan efek melingkar baik menurut 'arah jarum jam' maupun 'arah kebalikannya'.



TERIMA KASIH

