

# TUGAS PENDAHULUAN

## MODUL XIII

### LINKED LIST

Versi A.1

Prepared by: Vivi Novia - Muhammad Ikhlashul Amal / 27 Februari 2016

#### Pengenalan

#### *Pengertian Linked List*

Linked list adalah salah satu struktur data bentukan. Untuk mempermudah membayangkan, kita menggunakan analogi, kereta api. Sebuah rangkaian kereta api terdiri dari sebuah lokomotif dan beberapa gerbong (*coach*). Awal dari sebuah rangkaian kereta api pasti sebuah lokomotif, sedangkan akhirnya adalah sebuah gerbong. Semuanya dihubungkan dengan alat bernama *coupler*.



Gambar 1 Ilustrasi Kereta Api

Setiap gerbong yang ditarik oleh lokomotif tersebut mewakili satu buah data. Lokomotif sendiri adalah sebuah penanda dari awal rangkaian atau linked list. Pada struktur linked list, “lokomotif”-nya adalah pointer penanda awal linked list, sedangkan “gerbong”-nya adalah data. Setiap data memiliki penunjuk ke data berikutnya, data terakhir tidak menunjuk ke mana pun. Data dan penunjuk ke data berikutnya disebut “node”.

#### Tugas 1 : Membuat dan mencetak linked list

#### *Definisi Masalah*

Buatlah program utama untuk membuat dan mencetak linked list masukan user ke layar. Program menerima informasi berupa nilai elemen berupa bilangan bulat. Program akan meminta pengguna memasukkan nilai elemen hingga 3 buah inputan. Kemudian program mencetak *linked list* dengan nilai elemen secara berurutan sesuai masukan user. Asumsikan nilai elemen yang dimasukkan pengguna selalu valid.

#### Contoh Input dan Output

Input ke STDIN

3  
5  
4

Output ke STDOUT

Masukkan nilai pada linked list :  
Masukkan nilai pada linked list :  
Masukkan nilai pada linked list :  
Linked list yang dimasukkan :



3 5 4

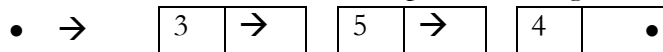
## Deliverable

Simpan program utama problem 1 dengan nama `problem1.c`. Jangan lupa memberikan identitas (*header file*) di awal file ini.

### Tugas 2 : Menggabungkan list

#### *Definisi Masalah*

Program penggabungan list yang akan dibuat adalah menggabungkan list kedua ke akhir list pertama. Misalkan kita memiliki list pertama sebagai berikut:



Gambar 2 Sebelum digabungkan

Kemudian setelah kita gabungkan dengan list kedua berisi data 6 dan 5, maka list yang kita punya akan menjadi :



Gambar 3 Setelah digabungkan

Tugas anda adalah membuat program yang menerima inputan untuk dua buah list. List pertama berisi 3 node dan list kedua berisi 4 node. Kemudian menggabungkan kedua list tersebut dan mencetaknya pada layar.

#### Contoh Input dan Output

Input ke STDIN

3  
5  
4  
6  
5  
7  
8

Output ke STDOUT

```
Input A :  
Input A :  
Input A :  
Input B :  
Input B :  
Input B :  
Input B :  
Linked list gabungan A dan B yang dimasukkan :  
3 5 4 6 5 7 8
```

## Deliverable

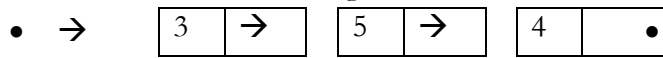
Simpan program utama problem 2 dengan nama `problem2.c`. Jangan lupa memberikan identitas (*header file*) di awal file ini.



### Tugas 3 : Menambahkan node pada list

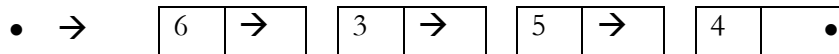
#### *Definisi Masalah*

Program penambahan node yang akan dibuat adalah menambahkan node pada awal list. Misalkan kita memiliki list sebagai berikut:



Gambar 4 Sebelum ditambahkan node

Kemudian setelah menambahkan node bernilai 6 pada awal list, maka list yang kita punya akan menjadi :



Gambar 5 Setelah ditambahkan node

Tugas anda adalah membuat program yang menerima inputan untuk membuat list berisi 3 node, mencetak ke layar, menerima inputan node yang akan ditambahkan pada awal list, kemudian menambahkan pada list tersebut dan mencetaknya pada layar.

#### Contoh Input dan Output

Input ke STDIN

```
3
5
4
6
```

Output ke STDOUT

```
Masukkan nilai pada list :
Masukkan nilai pada list :
Masukkan nilai pada list :
Linked list yang dimasukkan :
3 5 4
Masukkan nilai yang akan ditambahkan :
Linked list setelah ditambahkan :
6 3 5 4
```

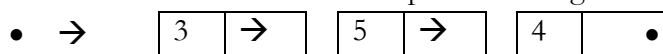
#### Deliverable

Simpan program utama problem 3 dengan nama problem3.c. Jangan lupa memberikan identitas (*header file*) di awal file ini.

### Tugas 4 : Menghapus node pada list

#### *Definisi Masalah*

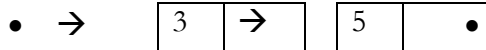
Program menghapus node yang akan dibuat adalah menghapus node pada urutan yang dipilih pada list. Misalkan kita memiliki list pertama sebagai berikut:



Gambar 6 Sebelum dihapus

Setelah node ketiga dihapus, list yang kita punya menjadi :





Gambar 7 Setelah dihapus

Tugas anda adalah membuat program yang menerima inputan untuk membuat list berisi 3 node, mencetak ke layar, menerima inputan node ke berapa yang akan dihapus pada list dengan asumsi input benar, kemudian menambahkan pada list tersebut dan mencetaknya pada layar.

### Contoh Input dan Output

Input ke STDIN

```
3
5
4
3
```

Output ke STDOUT

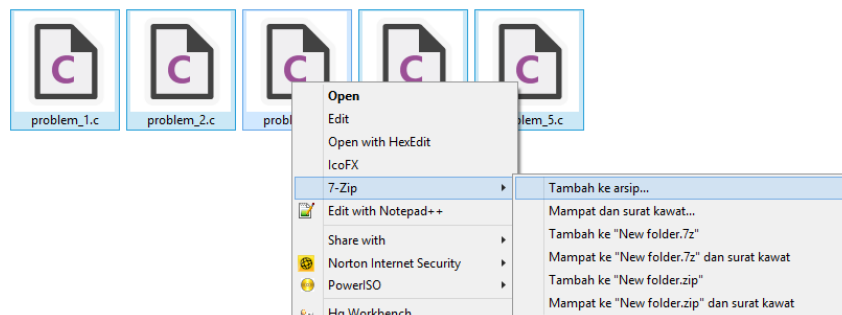
```
Masukkan nilai pada list :
Masukkan nilai pada list :
Masukkan nilai pada list :
Linked list yang dimasukkan :
3 5 4
Masukkan urutan node yang akan dihapus :
Linked list setelah dihapus :
3 5
```

### Deliverable

Simpan program utama problem 4 dengan nama problem4.c. Jangan lupa memberikan identitas (*header file*) di awal file ini.

### Petunjuk Penyerahan Tugas Pendahuluan Modul XIII

Simpan file problem1.c, problem2.c, problem3.c, dan problem4.c dalam satu folder. Gunakan program 7-zip untuk mengkompresi menjadi arsip TAR (.tar). Penamaan file TAR bebas (disarankan menggunakan NIM). File TAR ini yang akan di-submit ke server MIKU saat pengumpulan tugas pendahuluan saat memasuki laboratorium. Hanya file kode saja yang dimasukkan ke dalam arsip TAR. File *executable* tidak perlu dimasukkan.



Tambah ke arsip

Arsip: 13211007tar

Format arsip: tar

Level kompresi: Simpan

Metode kompresi:

Ukuran kamus:

Ukuran kata:

Ukuran blok padat:

Jumlah CPU: / 4

Pemakaian memori untuk Pemampatan: 1 MB

Pemakaian memori untuk Pengawamampatan: 1 MB

Belah ke volume, bita:

Parameter:

Ragam perbaru: Tambah dan ganti berkas

Pilihan

☐ Buat arsip SFX

☐ Kompres berkas bersama

Enkripsi

Masukkan kata sandi:

Konfirmasi kata sandi:

☐ Perlihatkan kata sandi

Metode enkripsi:

Oke Batal Bantuan

Selesai

