

# **LAPORAN PRAKTIKUM**

## **Algoritma Dan Struktur Data**

### **Jobsheet - 9 : Linked List**



Nama : Ghoffar Abdul Ja'far

NIM : 41720035

Kelas : 1E

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023/2024**

## Praktikum 1

Hasil:

```
Linked List Kosong!
Isi Linked List:      890
Isi Linked List:      890      760
Isi Linked List:      700      890      760
Isi Linked List:      700      999      890      760
Isi Linked List:      700      999      890      833      760
PS E:\!Kuliah\P_Algoritma_Struktur_Data>
```

### Pernyataan

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?  
= Karena pada print pertama pointer head belum nenujuk kepada head menjadikan list belum ada atau masih kosong.
2. Jelaskan kegunaan variable temp secara umum pada setiap method!  
= Temp merupakan variable sementara yang digunakan untuk mempermudah pembacaan kode dalam menunjuk alamat yang dituju
3. Perhatikan class SingleLinkedList, pada method insertAt Jelaskan kegunaan kode berikut  

```
if(temp.next.next==null) tail=temp.next;
```

  
= Digunakan untuk memperbarui referensi tail dari linked list jika elemen baru yang dimasukkan berada pada posisi terakhir dari list/tail

## Praktium 2

Hasil:

```
Linked List Kosong!
Isi Linked List:      890
Isi Linked List:      890      760
Isi Linked List:      700      890      760
Isi Linked List:      700      999      890      760
Isi Linked List:      700      999      890      833      760
Data pada indeks ke-1 = 999
Data 3 berada pada indeks ke-4
Isi Linked List:      700      890      833      760
Isi Linked List:      890      833      760
Isi Linked List:      833      760
Isi Linked List:      833
PS E:\!Kuliah\P_Algoritma_Struktur_Data>
```

### Pertanyaan

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!  
= Digunakan untuk menghentikan iterasi loop alamat yang ditunjuk telah dihapus
2. Jelaskan kegunaan kode dibawah pada method remove  

```
else if (temp.next.data == key) {
    temp.next = temp.next.next;
```

  - **temp.next.data == key:** Ini adalah kondisi untuk memeriksa apakah data pada node yang berada tepat setelah node yang saat ini ditunjuk oleh `temp` (yaitu `temp.next`) sama dengan `key`. Jika kondisi ini benar, artinya node yang berada setelah `temp` adalah node yang harus dihapus.

- **temp.next = temp.next.next;** Jika kondisi di atas terpenuhi, baris ini akan dijalankan. Ini mengubah referensi `next` dari node `temp` ke node yang berada setelah node `temp.next`. Dengan kata lain, ini "melewatkan" node yang saat ini `temp.next` dan langsung menghubungkan `temp` ke node setelahnya. Ini efektif menghapus node `temp.next` dari linked list karena tidak ada lagi referensi yang mengarah ke node tersebut, sehingga akan dihapus oleh garbage collector di Java.