

# Praktikum 6

---

## Single Linked List (2)

---

### A. TUJUAN PEMBELAJARAN

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

1. Memahami konsep membangun *single linked list*
2. Memahami konsep operasi menghapus simpul tertentu
3. Memahami konsep operasi menyisipkan setelah simpul tertentu
4. Memahami konsep operasi menyisipkan sebelum simpul tertentu
5. Mengimplementasikan semua operasi *single linked list* dalam pemrograman
6. Mengidentifikasi permasalahan-permasalahan pemrograman yang harus diselesaikan dengan menggunakan *linked list* dan menyelesaikannya.

### B. DASAR TEORI

#### B.1 Operasi Pada *Linked list*

Terdapat beberapa Operasi yang penting pada *linked list*, yaitu:

1. Menyisipkan sebagai simpul ujung(awal) dari *linked list*.
2. Membaca atau menampilkan
3. Mencari sebuah simpul tertentu
4. Menyisipkan sebagai simpul terakhir
5. Menghapus simpul tertentu
6. Menyisipkan setelah simpul tertentu
7. Menyisipkan sebelum simpul tertentu

#### B.1.1 Menghapus Simpul Tertentu

Langkah-langkah untuk menghapus simpul tertentu dari *linked list* adalah sebagai berikut:

1. Inisialisasi sebuah variabel bertipe `struct simpul* (hapus)` dengan ujung
2. Lakukan langkah 3 dan 4 selama data pada simpul hapus tidak sama dengan data yang dicari
3. Arahkan variabel bertipe `struct simpul* (sbl)` dengan hapus
4. Arahkan `hapus` ke `hapus->next`
5. Arahkan `sbl->next` ke `hapus->next`
6. Bebaskan simpul `hapus`

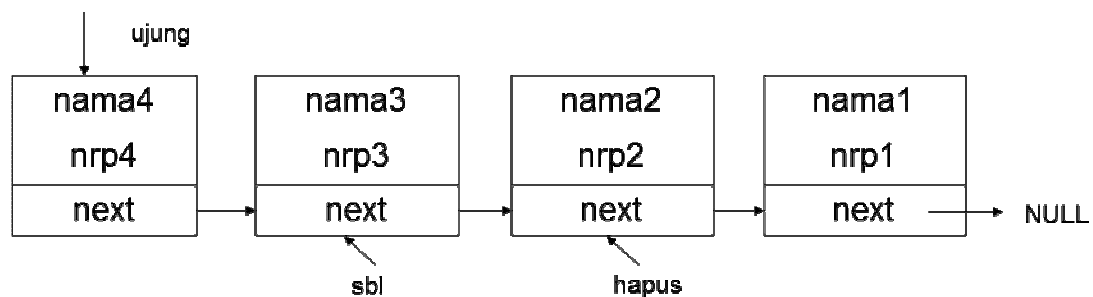
Berikut ini adalah perintah untuk menyisipkan data baru sebagai simpul terakhir pada *single linked list*

```

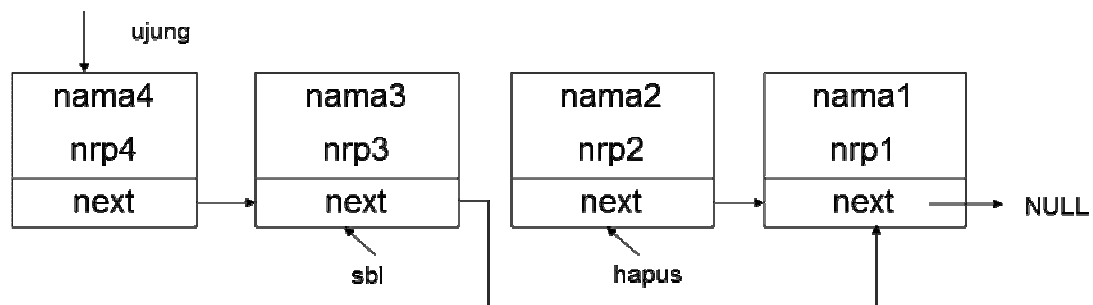
1. hapus = ujung;
2. while (hapus->nama != nama2)
3. {
4.     sbl = hapus;
5.     hapus=hapus->next;
6. }
7. sbl->next=hapus->next;
8. free(hapus);

```

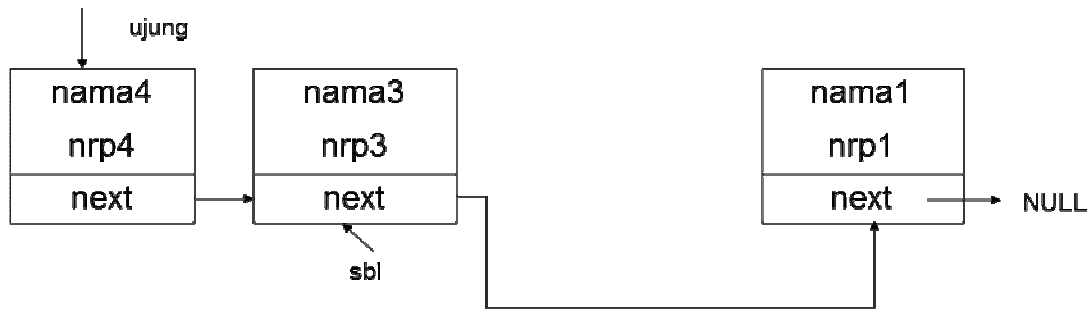
Setelah proses pencarian simpul yang dihapus didapat (Perintah baris 1-4)



Setelah perintah baris ke-11



Setelah perintah baris ke-12



### B.1.2 Menyisipkan Setelah Simpul Tertentu

Langkah-langkah untuk menyisipkan simpul baru setelah simpul tertentu pada *linked list* yang sudah terbentuk di atas adalah sebagai berikut:

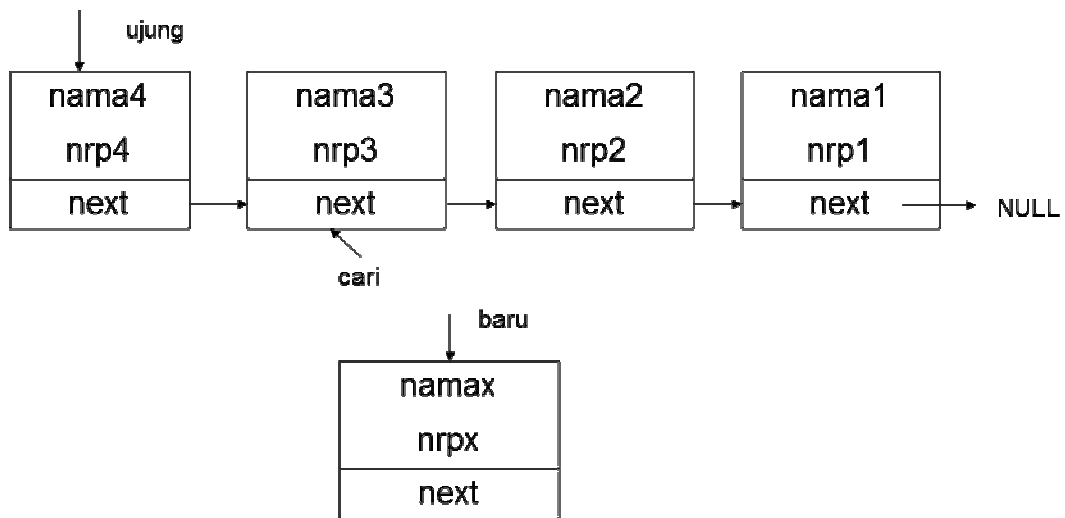
1. Alokasikan memori untuk simpul baru yang akan disisipkan
2. Inisialisasi sebuah variabel bertipe `struct simpul* (cari)` dengan `ujung`
3. Lakukan proses pencarian sampai data pada simpul `cari` sama dengan data yang setelahnya akan disisipkan simpul baru
4. Hubungkan `cari->next` ke simpul baru

Berikut ini adalah perintah untuk menyisipkan data baru setelah simpul tertentu pada *single linked list*

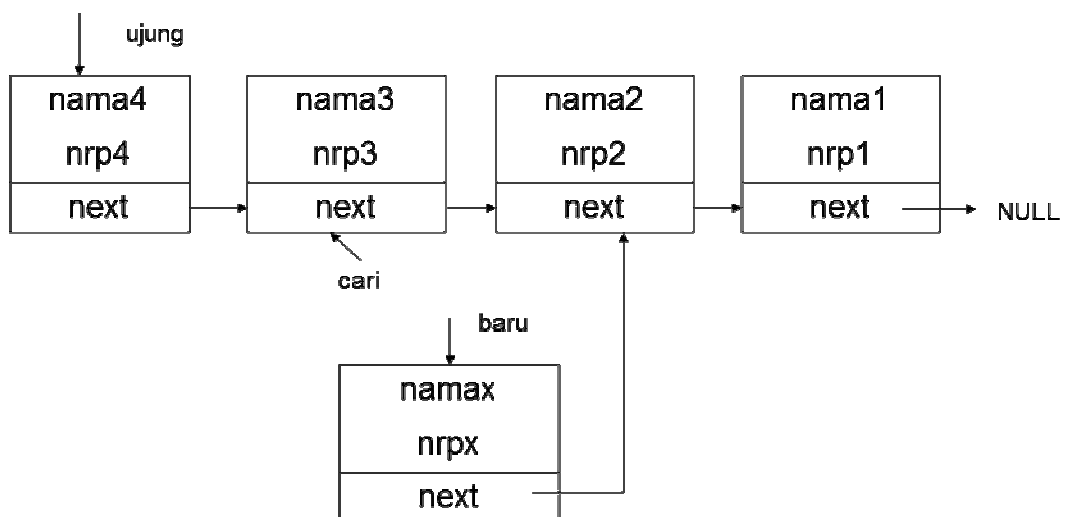
```

1. baru=alokasi_simpul();
2. cari = ujung;
3. while (cari->next !=nama3)
4.     cari = cari->next;
5. baru->next = cari->next;
6. cari->next = baru;
  
```

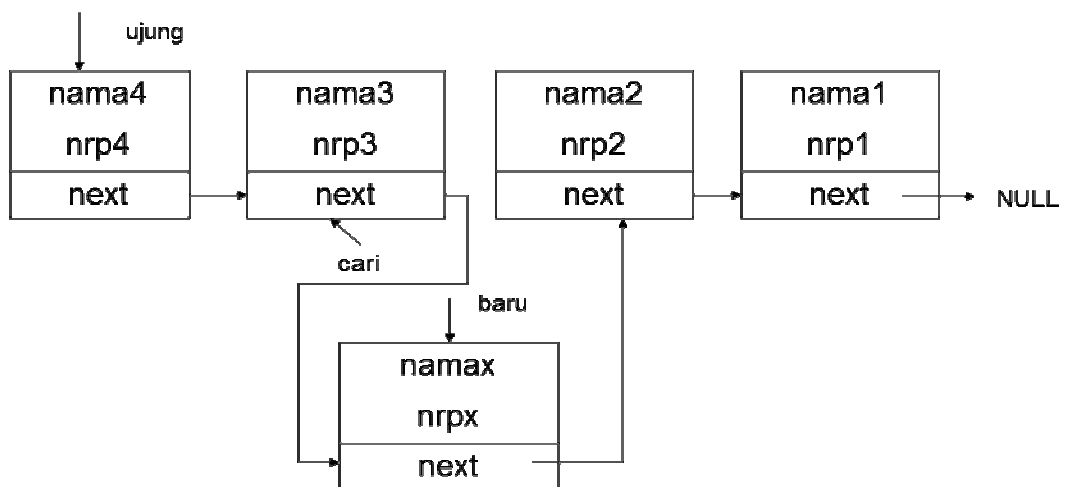
Setelah perintah baris 1-4



Setelah perintah baris ke-5



Setelah perintah baris ke-6



### 5.1.1. Menyisipkan Sebelum Simpul Tertentu

Langkah-langkah untuk menyisipkan simpul baru sebelum simpul tertentu pada *linked list* yang sudah terbentuk di atas adalah sebagai berikut:

1. Alokasikan memori untuk simpul baru yang akan disisipkan
2. Inisialisasi sebuah variabel bertipe `struct simpul* (cari)` dengan ujung
3. Lakukan langkah 4 dan 5 selama data pada simpul `hapus` tidak sama dengan data yang dicari
4. Arahkan variabel bertipe `struct simpul* (stl)` dengan `cari`
5. Arahkan `cari` ke `cari->next`
6. Arahkan `baru->next` ke `cari`
7. Arahkan `stl->next` ke `baru`

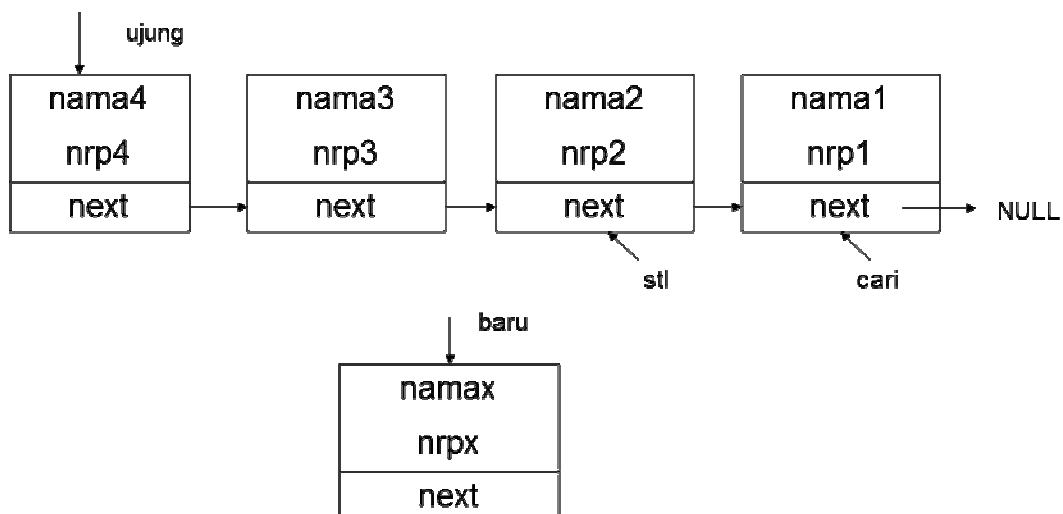
Berikut ini adalah perintah untuk menyisipkan data baru sebelum simpul tertentu pada *single linked list*

```

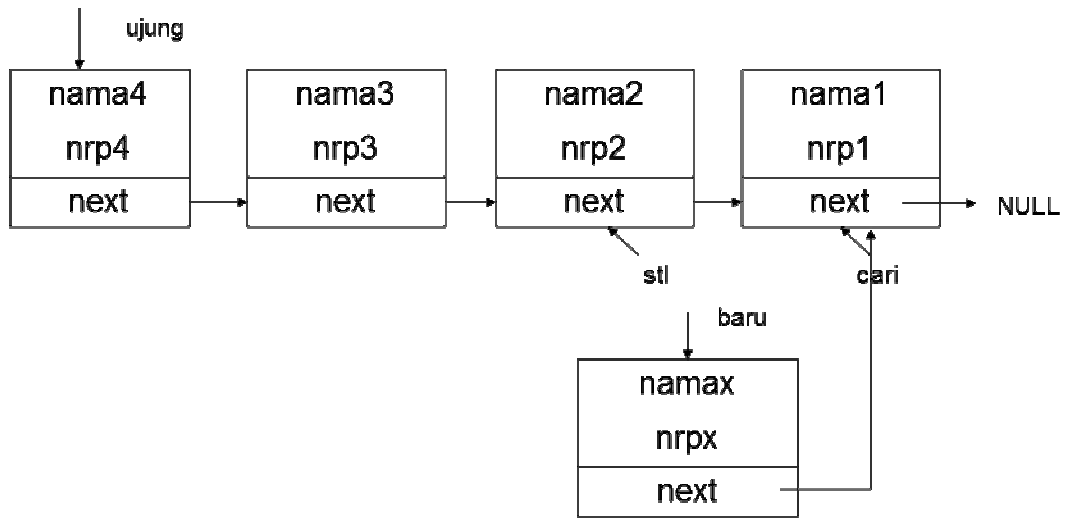
1. cari = ujung;
2. while (cari->nama!=nama1)
3. {
4.     stl=cari;
5.     cari=cari->next;
6. }
7. baru->next = cari;
8. stl->next = baru;

```

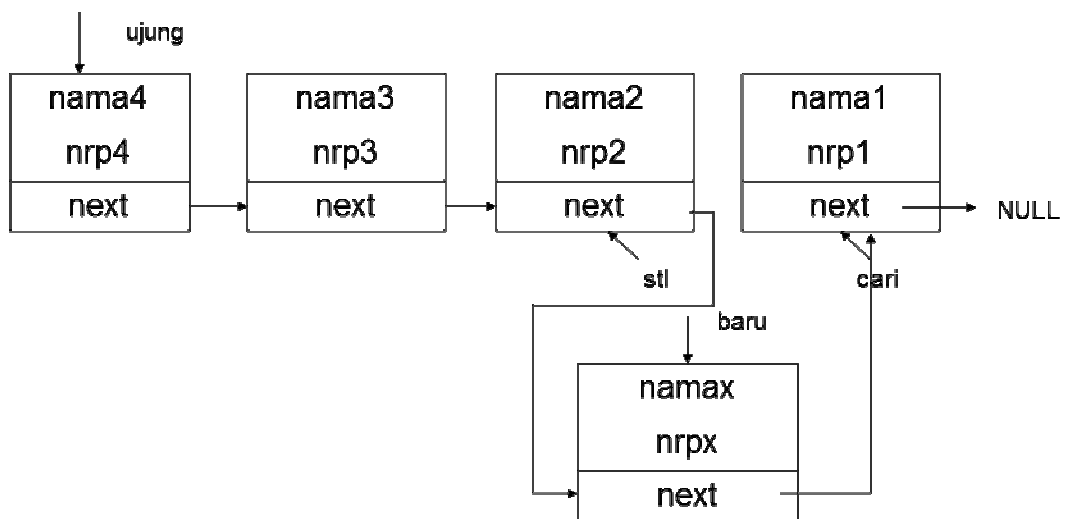
Setelah perintah baris 1-6



Setelah perintah baris ke-7



Setelah perintah baris ke-8



### C. TUGAS PENDAHULUAN

Untuk semua operasi dasar *single linked list* persoalan di bawah ini, desainlah algoritma dan flowchartnya :

1. Menghapus simpul tertentu
2. Menyisipkan setelah simpul tertentu
3. Menyisipkan sebelum simpul tertentu

#### D. PERCOBAAN

1. Implementasikan operasi dasar *Single linked list* : Menghapus simpul tertentu. Tambahkan kondisi jika yang dihapus adalah data yang paling depan atau data yang paling terakhir.
2. Implementasikan operasi dasar *Single linked list* : Menyisipkan setelah simpul tertentu. Tambahkan kondisi jika data yang disisipkan setelahnya adalah data terakhir.
3. Implementasikan operasi dasar *Single linked list* : Menyisipkan sebelum simpul tertentu. Tambahkan kondisi jika data yang disisipkan setelahnya adalah data terakhir.
4. Gabungkan semua operasi di atas dalam sebuah Menu Pilihan.

#### E. LATIHAN

1. Bangunlah sebuah *single linked list* dengan prinsip FIFO (First In First Out)
2. Merepresentasikan sebuah bilangan polinomial dengan *single linked list*

Masalah aritmatika polinom adalah membuat sekumpulan subrutin manipulasi terhadap polinom simbolis (symbolic Polynomial).

$$\text{Misalnya: } P1 = 6x^8 + 8x^7 + 5x^5 + x^3 + 15$$

$$P2 = 3x^9 + 4x^7 + 3x^4 + 2x^3 + 2x^2 + 10$$

Representasikan bilangan polinom dengan menggunakan linked list dan buatlah prosedur-prosedur untuk :

- Menyisipkan simpul di awal jika pangkat yang dimasukkan lebih dari pangkat tertinggi dari bilangan polinomial.
- Menyisipkan simpul di tengah jika pangkat dari bilangan yang kita sisipkan berada di tengah.
- Menyisipkan simpul di akhir jika pangkat dari bilangan yang disisipkan adalah 0.
- Menghapus simpul, baik di awal, di tengah, ataupun di akhir.

#### F. LAPORAN RESMI

1. Kerjakan hasil percobaan(D) dan latihan(E) di atas dan tambahkan analisa.
2. Tuliskan kesimpulan dari percobaan dan latihan yang telah anda lakukan.