Institut Teknologi Telkom Purwokerto

Teknik Informatika - Fakultas Informatika

Pertemuan 5 – Linked List

Author: **Wahyu Andi Saputra [WAA]**

Co-Author: **Condro Kartiko [CKO]**

Institut Teknologi Telkom Purwokerto

OUTLINE

- Linked List vs Array
- Single Linked List
- Double Linked List
- Circular Linked List

By: Wahyu Andi Saputra [WAA]

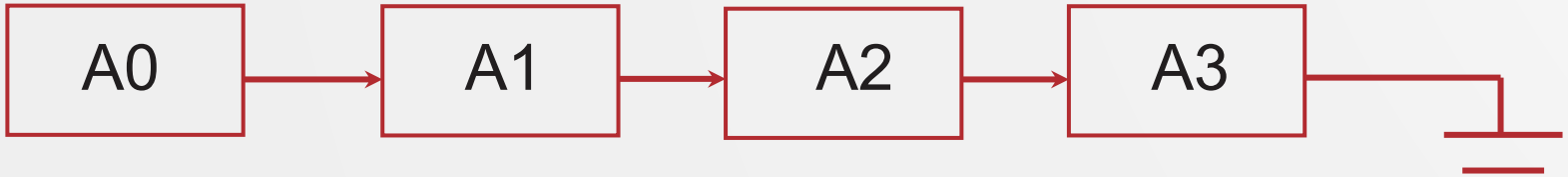
Institut Teknologi
Telkom
Purwokerto

Array vs Linked List

By: Wahyu Andi Saputra [WAA]

Linked List

Institut Teknologi Telkom Purwokerto



- **Menyimpan koleksi elemen secara non-contiguously.**
  - Elemen dapat terletak pada lokasi memory yang saling berjauhan. Bandingkan dengan array dimana tiap-tiap elemen akan terletak pada lokasi memory yang berurutan.
- **Mengizinkan operasi penambahan atau penghapusan elemen ditengah-tengah koleksi dengan hanya membutuhkan jumlah perpindahan elemen yang konstan.**
  - Bandingkan dengan array. Berapa banyak elemen yang harus dipindahkan bila akan menyisipi elemen ditengah- tengah array?

By: Wahyu Andi Saputra [WAA]

Linked List

Institut Teknologi Telkom Purwokerto

- Sebuah list merupakan rantai dari object bertipe `ListNode` yang berisikan data dan referensi (pointer) kepada `ListNode` selanjutnya dalam list.
- Harus diketahui dimana letak elemen pertama!

By: Wahyu Andi Saputra [WAA]

Linked List

Institut Teknologi Telkom Purwokerto

- Array
  - Statis
  - Terbatas dalam menambah/menghapus data
  - Random Access
  - Penghapusan data tidak mungkin
- Linked List
  - Dinamis
  - Tidak terbatas dalam menambah/menghapus
  - Sequential Access
  - Penghapusan data mudah

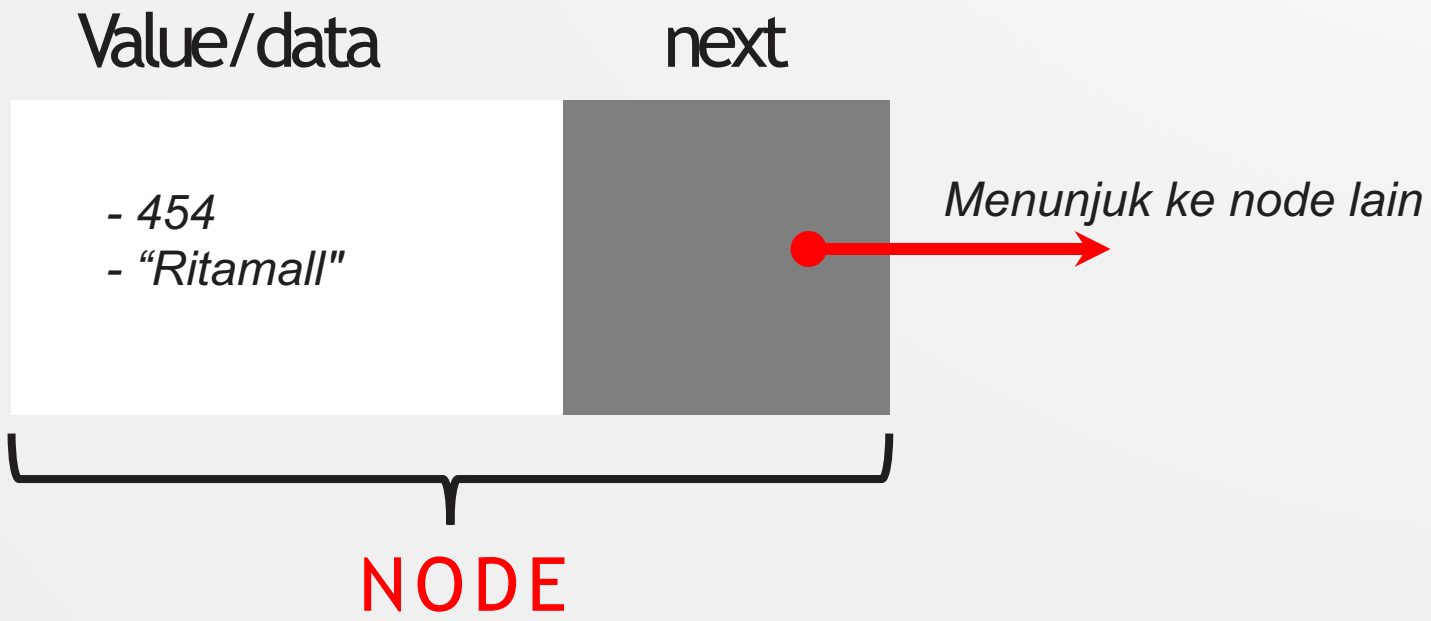
By: Wahyu Andi Saputra [WAA]

Linked List

Institut Teknologi Telkom Purwokerto

- Yang disimpan dalam **`ListNode`** adalah reference dari object-nya, BUKAN object-nya itu sendiri atau salinan dari object-nya !!!

```
struct namaTeman
{
    string nama;
    namaTeman *next;
}
```



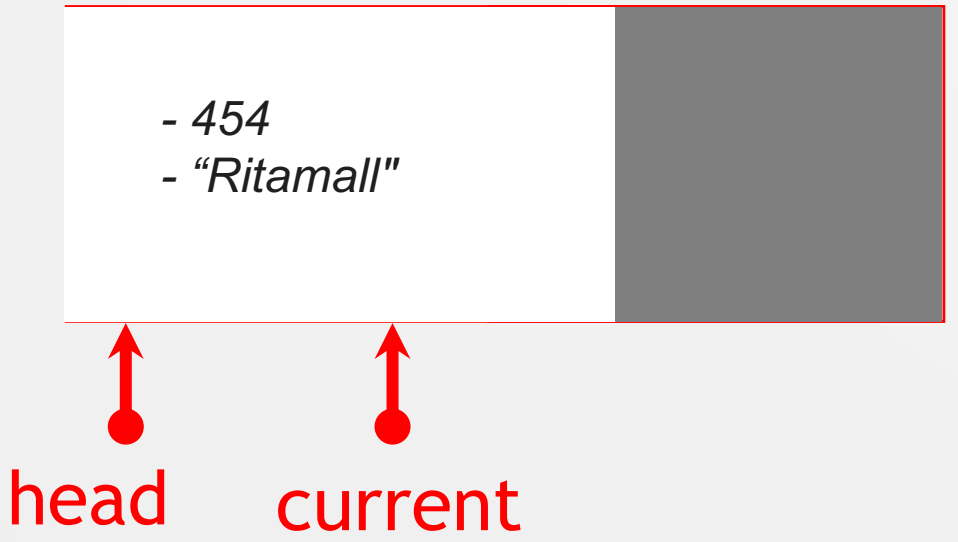
By: Wahyu Andi Saputra [WAA]

Linked List

Institut Teknologi Telkom Purwokerto

```
struct namaTeman
{
    string nama;
    namaTeman *next;
} *head

head = new namaTeman;
current = head;
```

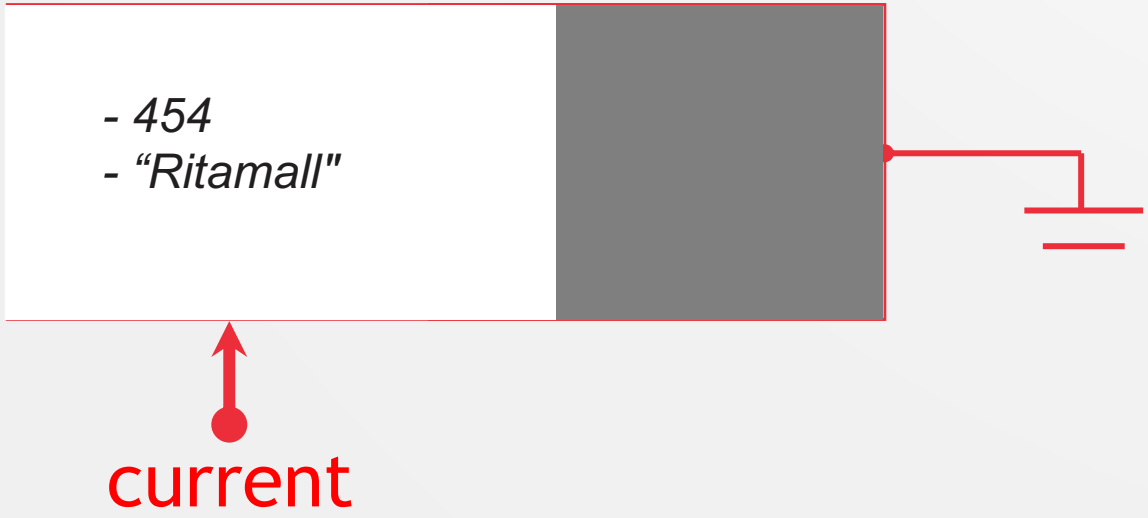


By: Wahyu Andi Saputra [WAA]

Linked List

Institut Teknologi Telkom Purwokerto

```
current->next = NULL;
```

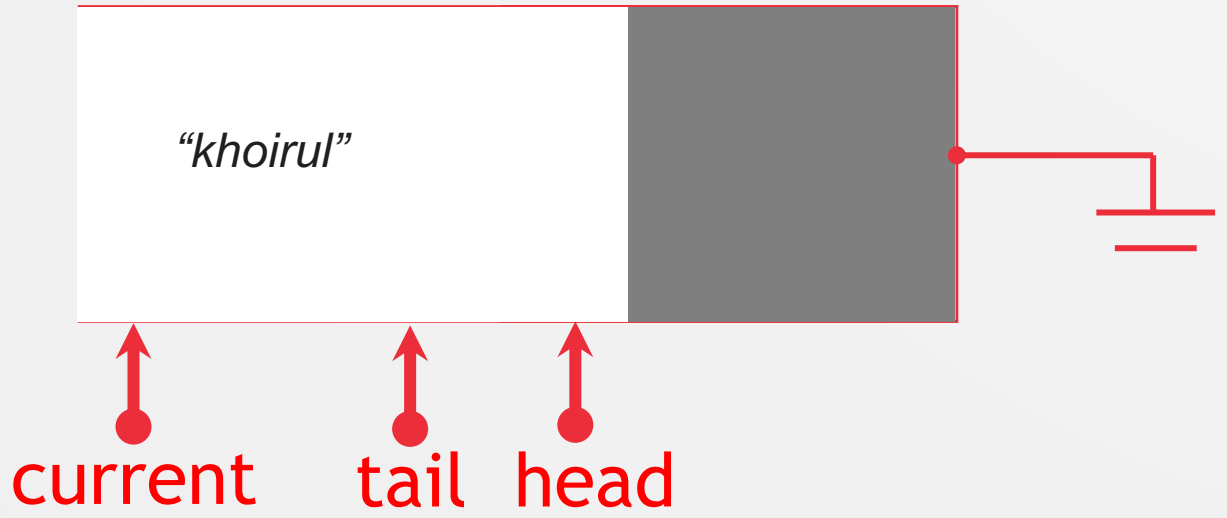


By: Wahyu Andi Saputra [WAA]

Linked List

Institut Teknologi Telkom Purwokerto

```
current = new namaTeman;
current->nama = "khoirul";
tail = current;
head = current;
tail->next=NULL;
```



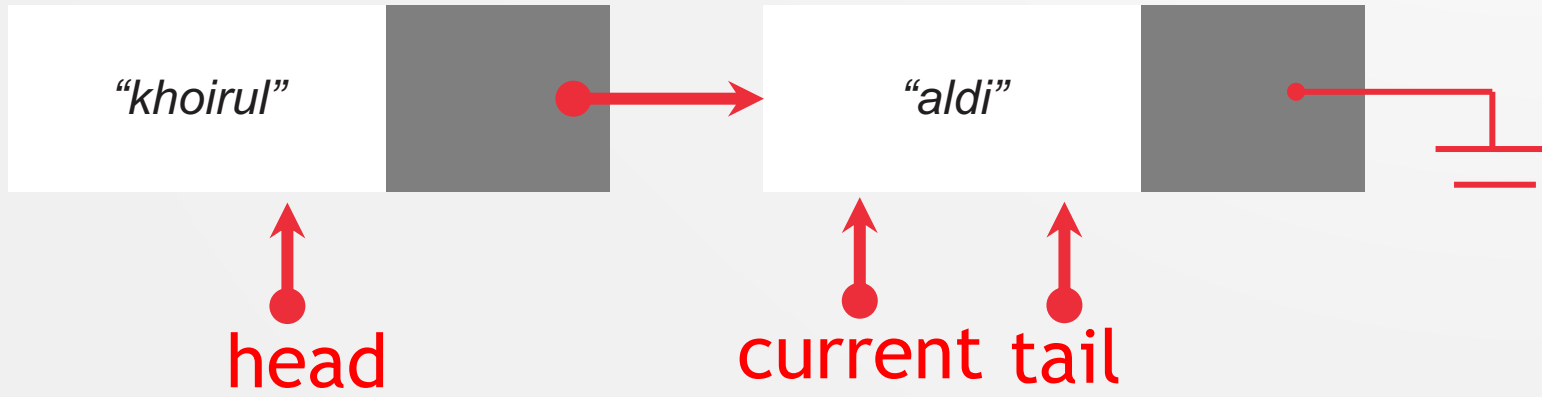
By: Wahyu Andi Saputra [WAA]

Linked List

Institut Teknologi Telkom Purwokerto

```
current = new namaTeman;
current->nama = "khoirul";
tail = current;
head = current;
tail->next=NULL;

current = new namaTeman;
current->nama = "aldi";
tail->next = current;
tail = tail->next;
tail->next=NULL;
```



By: Wahyu Andi Saputra [WAA]

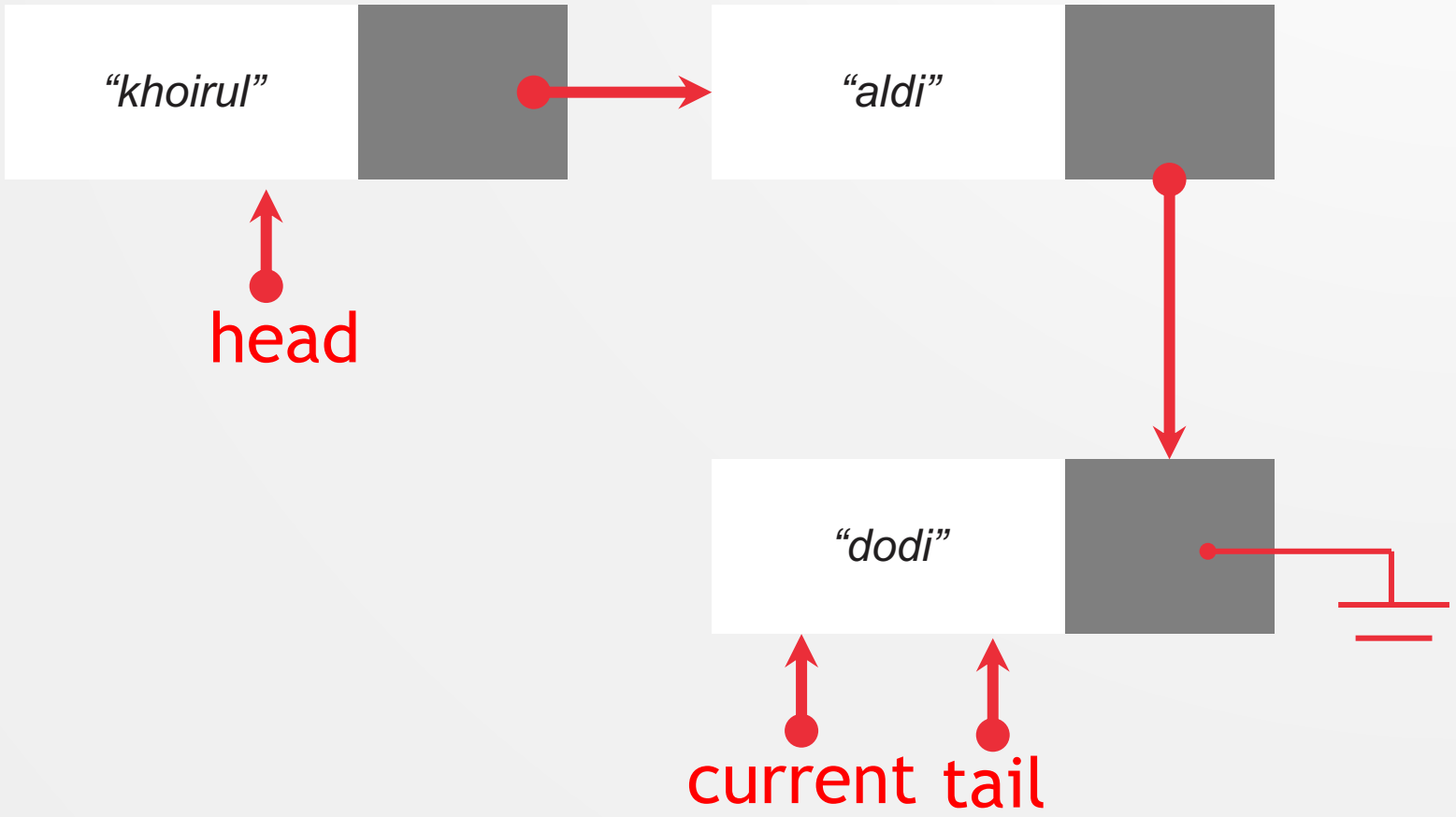
Linked List

Institut Teknologi Telkom Purwokerto

```
current = new namaTeman;
current->nama = "khoirul";
tail = current;
head = current;
tail->next=NULL;

current = new namaTeman;
current->nama = "aldi";
tail->next = current;
tail = tail->next;
tail->next=NULL;

current = new namaTeman;
current->nama = "dodi";
tail->next = current;
tail = tail->next;
tail->next=NULL;
```

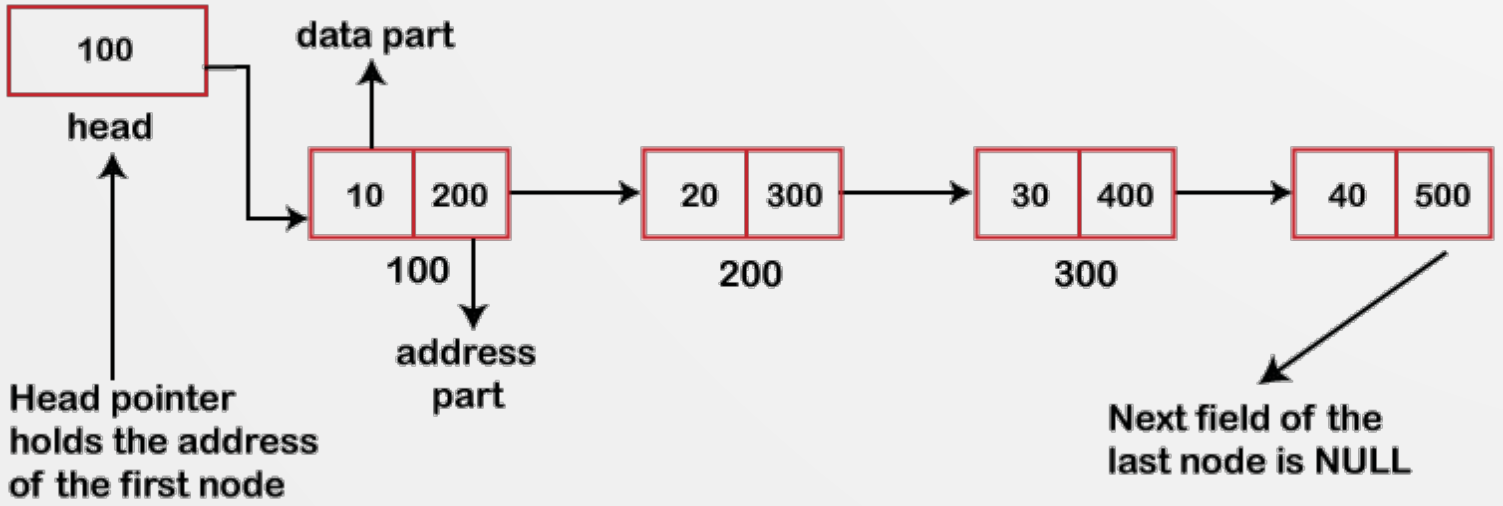


By: Wahyu Andi Saputra [WAA]

Single Linked List

Institut Teknologi Telkom Purwokerto

- *Single linked list* atau linked list
- Tiap elemen terdiri dari dua bagian, yaitu sebuah data dan sebuah pointer/link yang disebut dengan link *next*.

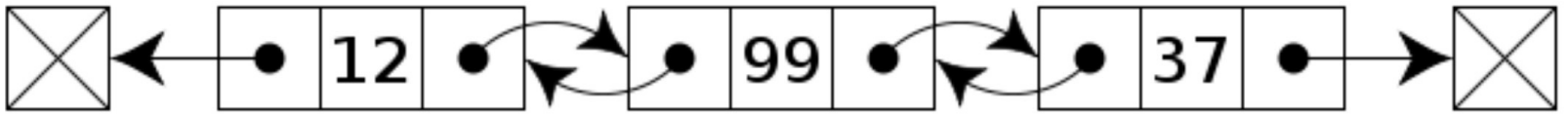


By: Wahyu Andi Saputra [WAA]

Double Linked List

Institut Teknologi Telkom Purwokerto

- Struktur data atas tiap-tiap node memiliki rujukan pada node sebelum dan berikutnya. Sebagian algoritma membutuhkan taut ganda, contohnya sorting dan reverse traversing.



Senarai ganda dengan tiap-tiap node yang terdiri atas tiga elemen, data integer, dan dua elemen rujukan ke node sebelum serta berikutnya

By: Wahyu Andi Saputra [WAA]

Circular Linked List

Institut Teknologi Telkom Purwokerto

- Pada dua jenis senarai sebelumnya, node terakhir dalam senarai tersebut merujuk pada null yang artinya akhir dari sebuah senarai, begitu pula null sebagai rujukan node sebelumnya pada node pertama bila senarai yang dimaksudkan adalah senarai ganda.

- Pada senarai sirkular, informasi rujukan pada node terakhir akan merujuk pada node pertama, dan rujukan pada node pertama akan merujuk pada node terakhir bila yang digunakan sebagai dasar implementasi adalah senarai ganda.

By: Wahyu Andi Saputra [WAA]

Circular Linked List

Institut Teknologi Telkom Purwokerto

Senarai sirkular dengan menggunakan model implementasi senarai tungal. Node terakhir menyimpan rujukan pada node pertama

By: Wahyu Andi Saputra [WAA]

Institut Teknologi
Telkom
Purwokerto

Single Linked List

By: Wahyu Andi Saputra [WAA]

Single Linked List

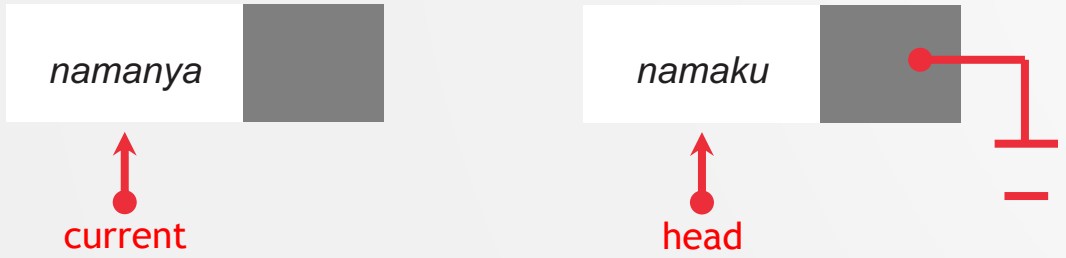
Institut Teknologi Telkom Purwokerto

- **INSERT**
  - insert sebagai node awal (head) dari linked list
  - insert sebagai node akhir (tail) dari linked list
  - insert setelah node tertentu
  - insert sebelum node tertentu

By: Wahyu Andi Saputra [WAA]

Single Linked List

insert sebagai node **awal (head)** dari linked list




```
void tambahDiDepan(string namanya)
{
    current = new namaTeman;
    current->nama = namanya;
    current->next = head;
    head = current;
}
```

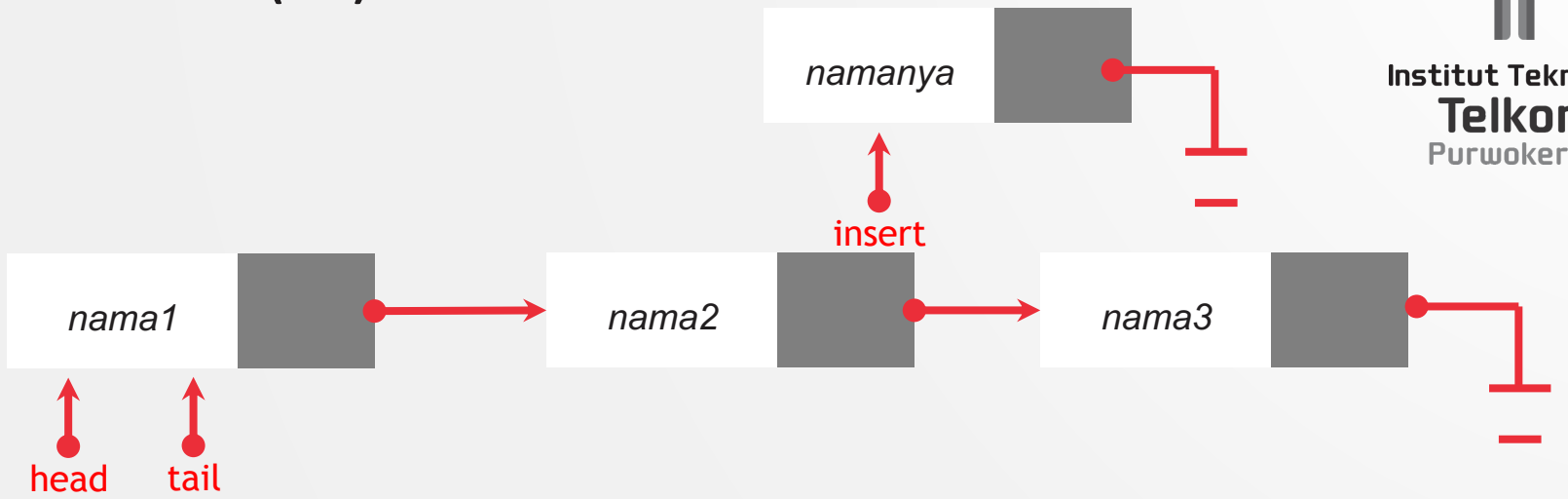
By: Wahyu Andi Saputra [WAA]

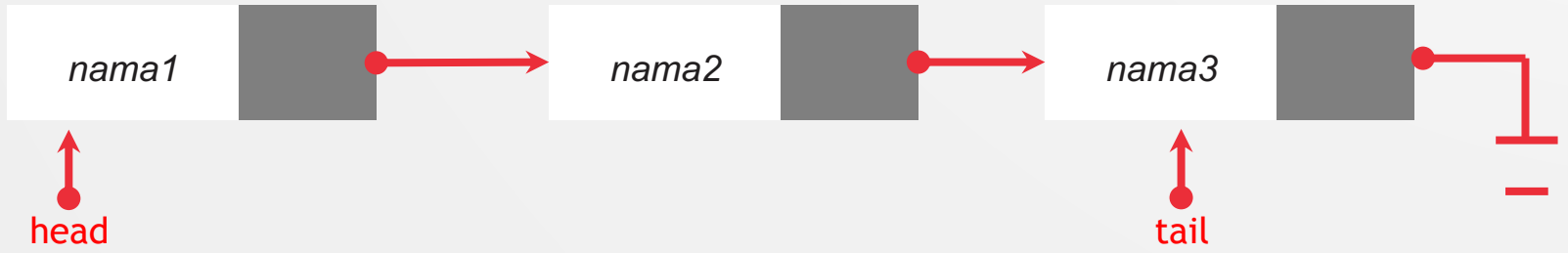
Single Linked List

insert sebagai node **akhir (tail)** dari linked list

Institut Teknologi Telkom Purwokerto

```
void tambahDiBelakang(string namanya)
{
    insert = new namaTeman();
    insert->nama = namanya;
    insert->next = NULL;
    tail = head;
    do
        tail = tail->next;
    while (tail->next != NULL);
    tail->next = insert;
    tail = tail->next;
    tail->next = NULL;
}
```





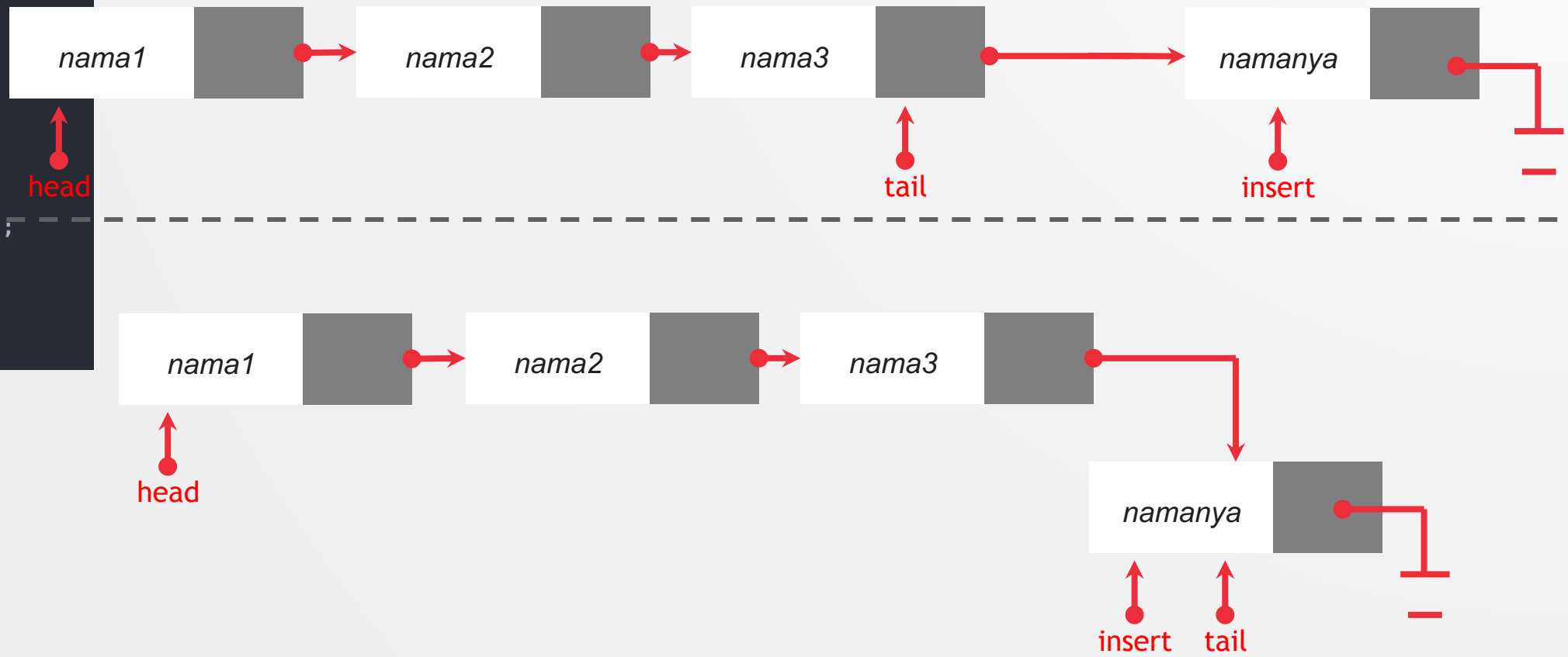
By: Wahyu Andi Saputra [WAA]

Single Linked List

insert sebagai node **akhir (tail)** dari linked list

Institut Teknologi Telkom Purwokerto

```
void tambahDiBelakang(string namanya)
{
    insert = new namaTeman();
    insert->nama = namanya;
    insert->next = NULL;
    tail = head;
    do
        tail = tail->next;
    while (tail->next != NULL);
    tail->next = insert;
    tail = tail->next;
    tail->next = NULL;
}
```



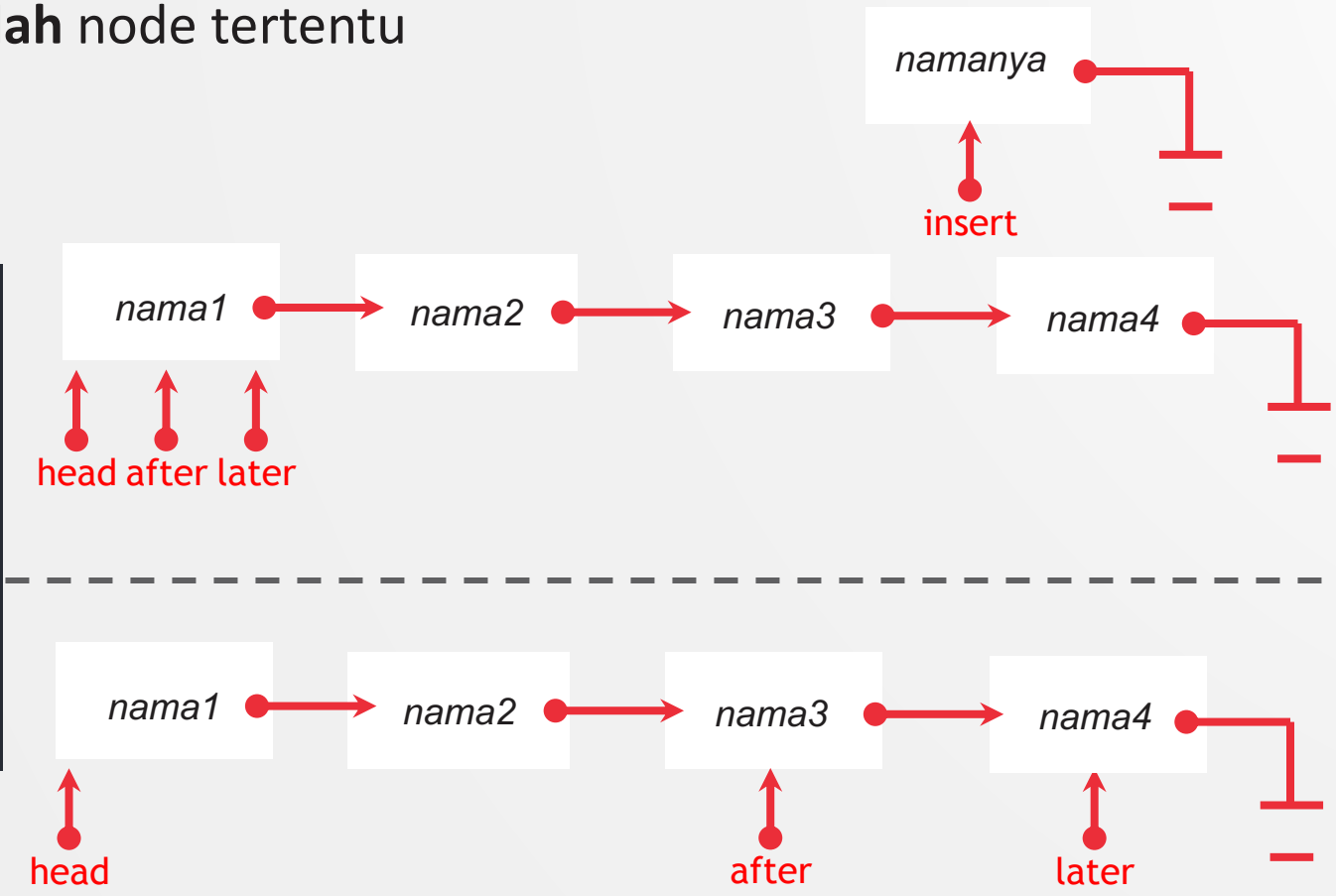
By: Wahyu Andi Saputra [WAA]

Single Linked List

insert **setelah** node tertentu

```
void tambahSetelah(string yangDicari, string namanya)
{
    insert = new namaTeman();
    insert->nama = namanya;
    after = head;
    later = head;
    do
    {
        after = later;
        later = later->next;
    } while (after->nama != yangDicari);
    after->next = insert;
    insert->next = later;
}
```

Misal, setelah "nama3"



Institut Teknologi Telkom Purwokerto

By: Wahyu Andi Saputra [WAA]

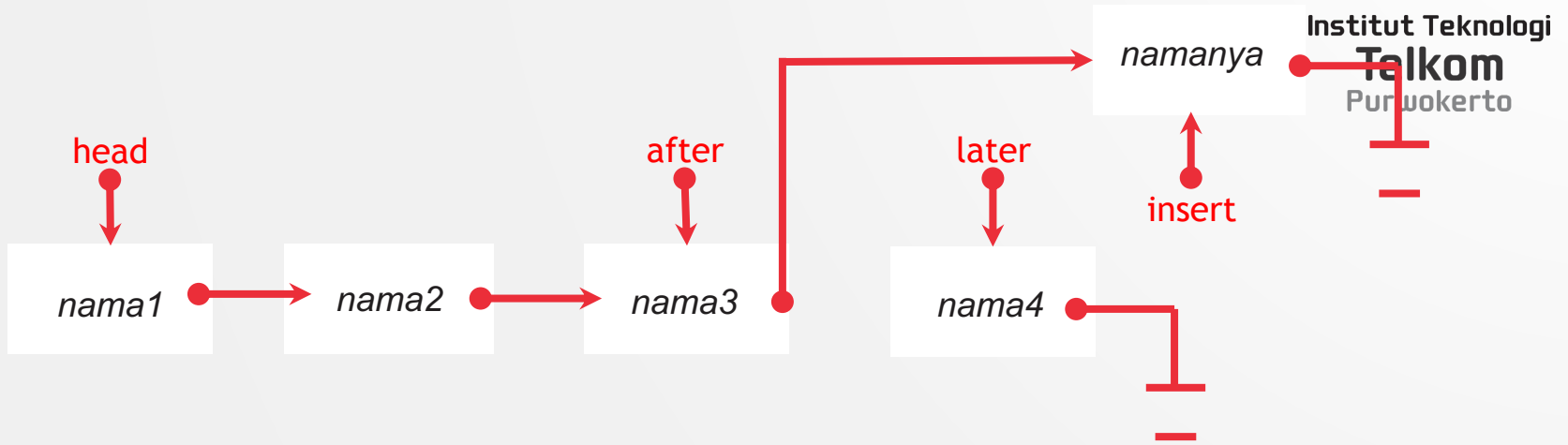
Single Linked List

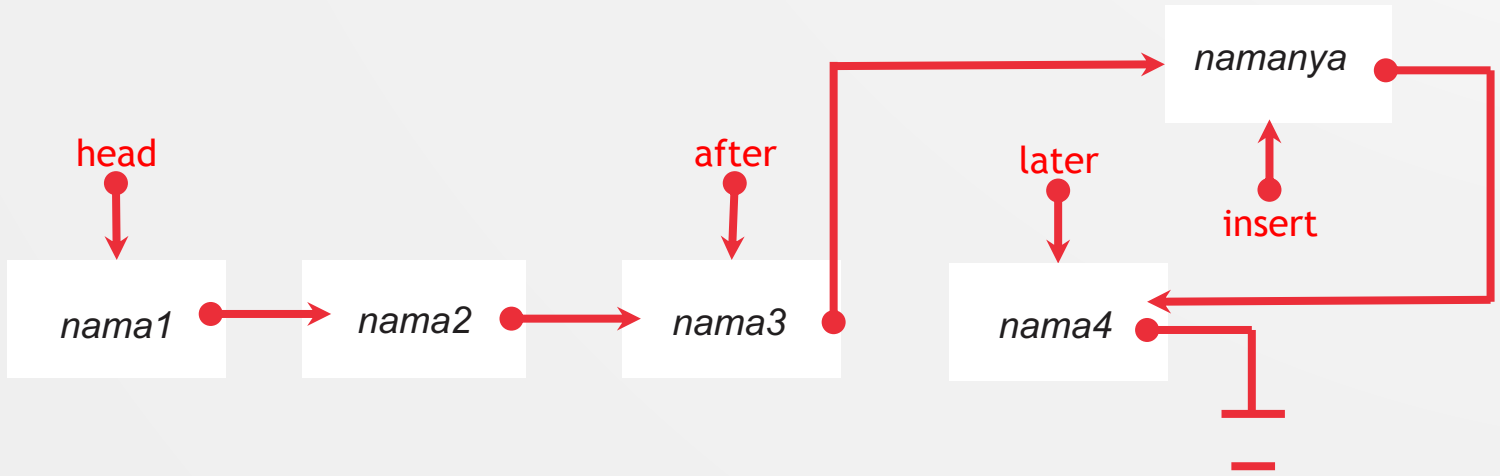
insert **setelah** node tertentu

Institut Teknologi Telkom Purwokerto

```
void tambahSetelah(string yangDicari, string namanya)
{
    insert = new namaTeman();
    insert->nama = namanya;
    after = head;
    later = head;
    do
    {
        after = later;
        later = later->next;
    } while (after->nama != yangDicari);
    after->next = insert;
    insert->next = later;
}
```

Misal, setelah "nama3"





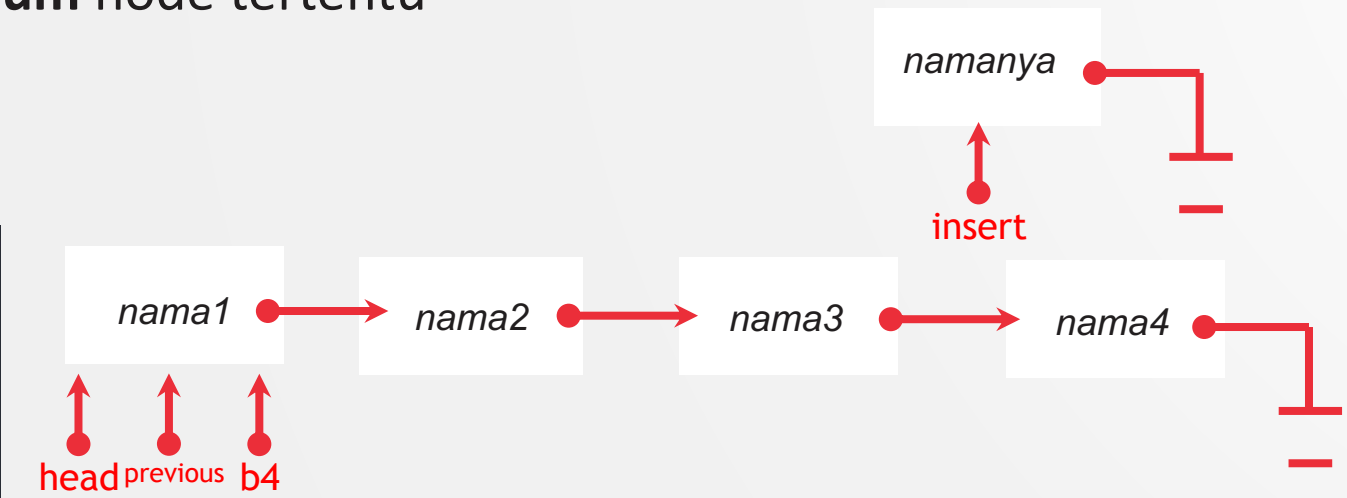
By: Wahyu Andi Saputra [WAA]

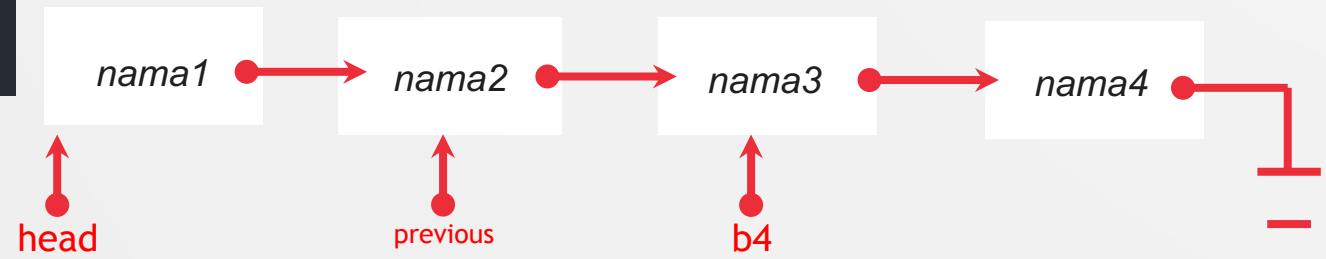
Single Linked List

insert **sebelum** node tertentu

```
void tambahSebelum(string yangDicari, string namanya)
{
    insert = new namaTeman();
    insert->nama = namanya;
    previous = head;
    b4 = head;
    do
    {
        previous = b4;
        b4 = b4->next;
    } while (b4->nama != yangDicari);
    previous->next = insert;
    insert->next = b4;
}
```

Misal, sebelum "nama3"





Institut Teknologi Telkom Purwokerto

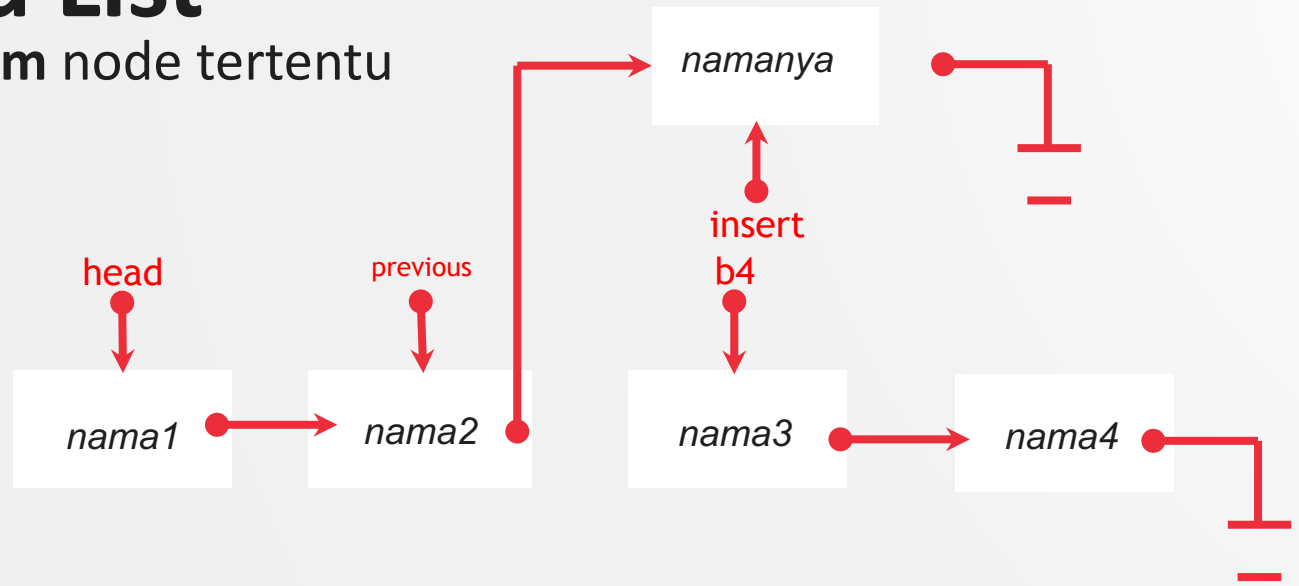
By: Wahyu Andi Saputra [WAA]

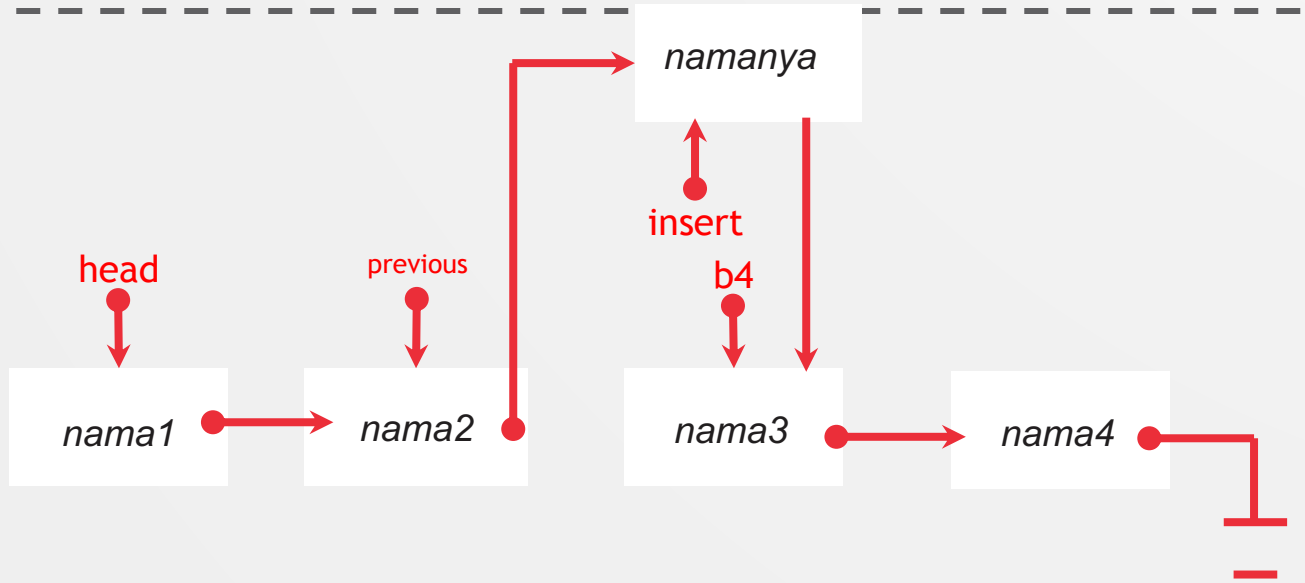
Single Linked List

insert **sebelum** node tertentu

```
void tambahSebelum(string yangDicari, string namanya)
{
    insert = new namaTeman();
    insert->nama = namanya;
    previous = head;
    b4 = head;
    do
    {
        previous = b4;
        b4 = b4->next;
    } while (b4->nama != yangDicari);
    previous->next = insert;
    insert->next = b4;
}
```

Misal, sebelum "nama3"





Institut Teknologi Telkom Purwokerto

By: Wahyu Andi Saputra [WAA]

Single Linked List

Institut Teknologi Telkom Purwokerto

- **DELETE**
  - delete sebagai simpul pertama(head) dari linked list
  - delete simpul terakhir
  - delete setelah simpul tertentu

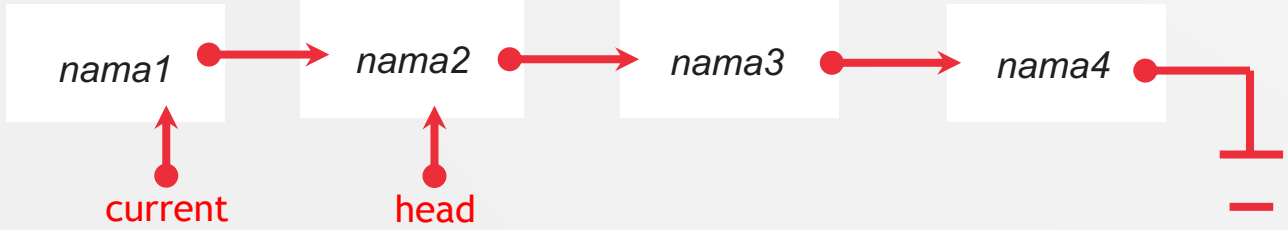
By: Wahyu Andi Saputra [WAA]

Single Linked List

delete sebagai **simpul pertama(head)** dari linked list

Institut Teknologi Telkom Purwokerto

```
void hapusAwal()
{
    current = head;
    head = current->next;
}
```



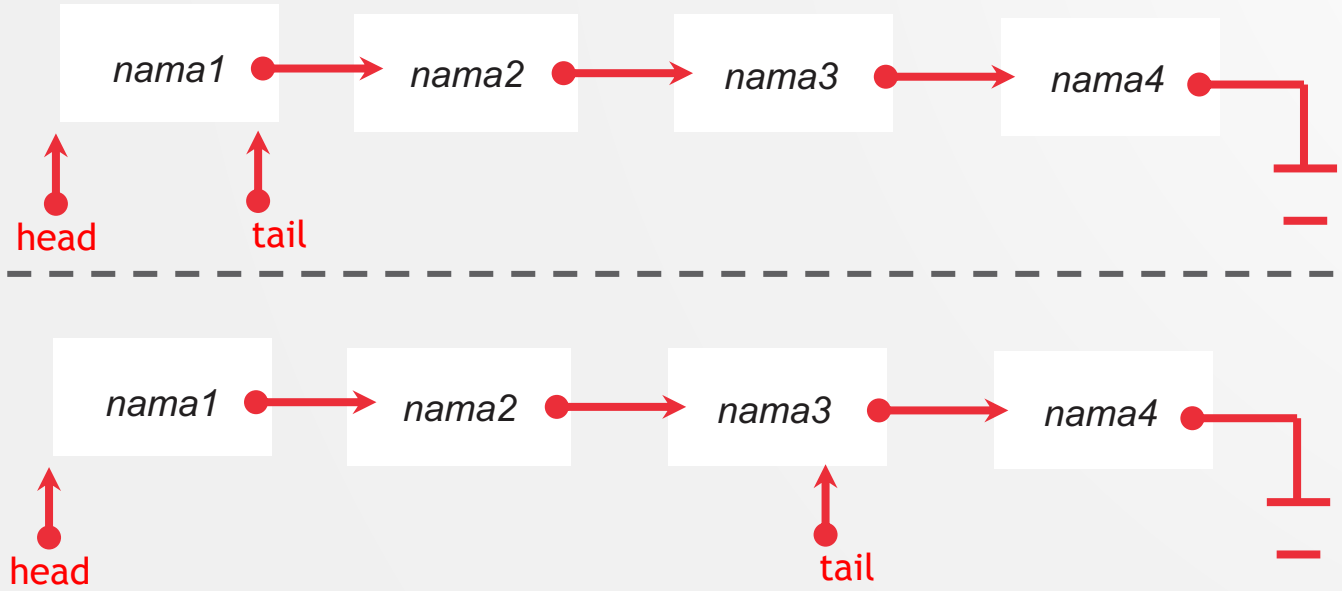
By: Wahyu Andi Saputra [WAA]

Single Linked List

delete sebagai **simpul akhir (tail)** dari linked list

Institut Teknologi Telkom Purwokerto

```
void hapusAkhir2()
{
    tail = head;
    while (tail->next->next != NULL)
        tail = tail->next;
    tail->next = NULL;
}
```

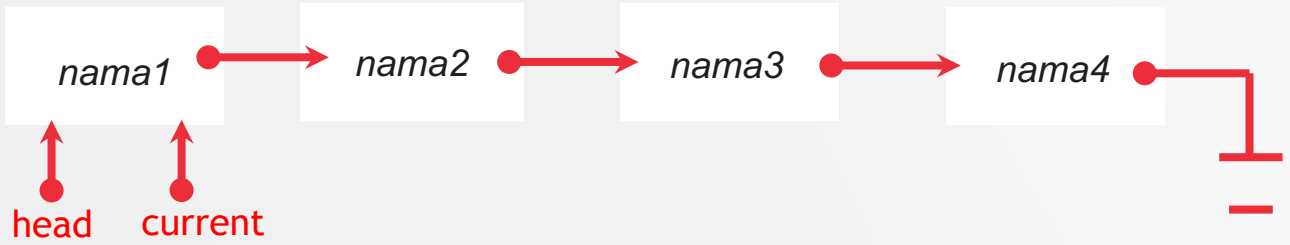


By: Wahyu Andi Saputra [WAA]

Single Linked List

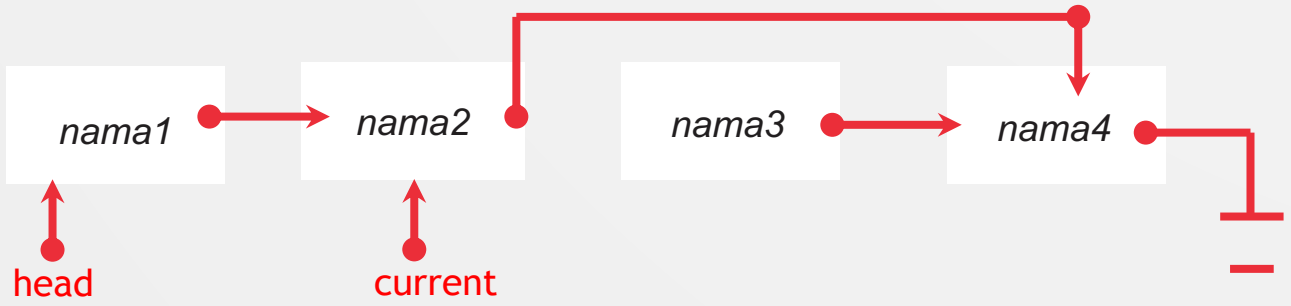
delete **simpul tertentu**

Institut Teknologi Telkom Purwokerto




```
void hapusTertentu(string yangDicari)
{
    current = head;
    while (current->next->nama != yangDicari)
    {
        current = current->next;
    }
    current->next = current->next->next;
}
```

Misal, menghapus "nama3"



By: Wahyu Andi Saputra [WAA]

Institut Teknologi
Telkom
Purwokerto

TERIMA KASIH

Angkat tangan apabila ada pertanyaan

Institut Teknologi
Telkom
Purwokerto