

---

# DOUBLE LINKED LIST (MENGHAPUS SIMPUL)

Dosen : Sulistyowati, ST., M.Kom.

# MENGHAPUS SIMPUL DI DEPAN

- Operasi ini berguna untuk menghapus data/simpul pada posisi pertama/di awal double linked list.
- Ada 3 keadaan yang mungkin terjadi ketika akan melakukan proses penghapusan simpul di depan, yaitu :

- Kondisi double linked list masih kosong

Jika ini terjadi, maka proses penghapusan data/simpul tidak bisa dilakukan karena double linked list masih kosong / tidak ada data/simpul

# MENGHAPUS SIMPUL DI DEPAN

## Algoritma :

```
if (awal==NULL)
```

```
    cout<<"PENGHAPUSAN DATA TIDAK BISA DILAKUKAN,  
    KARENA LIST KOSONG";
```

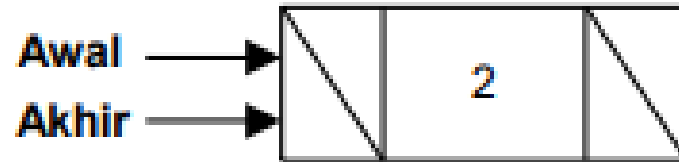
# MENGHAPUS SIMPUL DI DEPAN

- Kondisi double linked list hanya memiliki 1 data/simpul

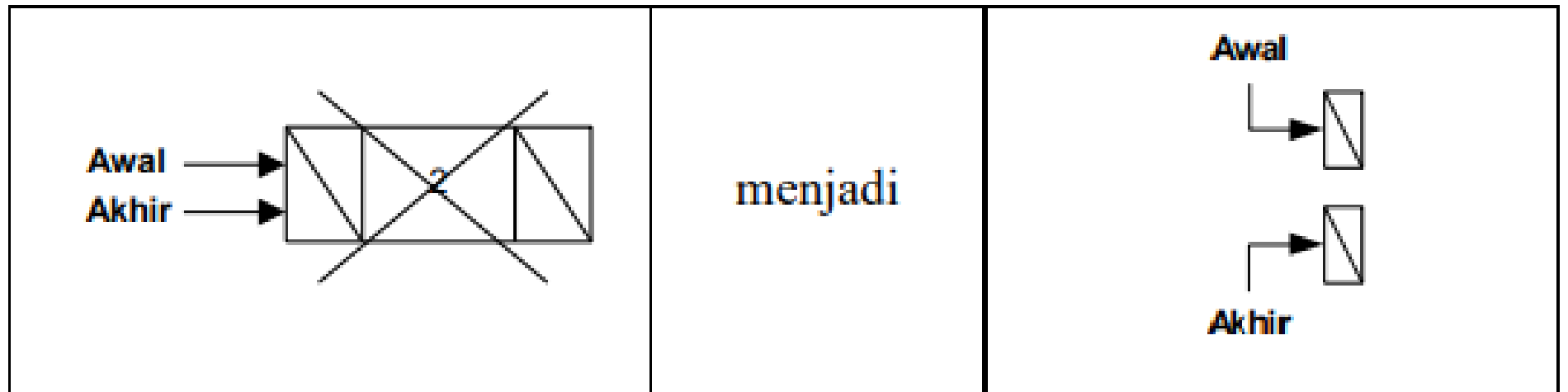
Langkah yang perlu dilakukan ketika ingin melakukan proses penghapusan double linked list yang memiliki hanya 1 data/simpul adalah dengan langsung menghapus data/simpul dan kemudian pointer awal dan akhir diberi nilai NULL.

# MENGHAPUS SIMPUL DI DEPAN

- Kondisi double linked list sebelum dihapus :



- Kondisi double linked list setelah dihapus :



# MENGHAPUS SIMPUL DI DEPAN

## Algoritma :

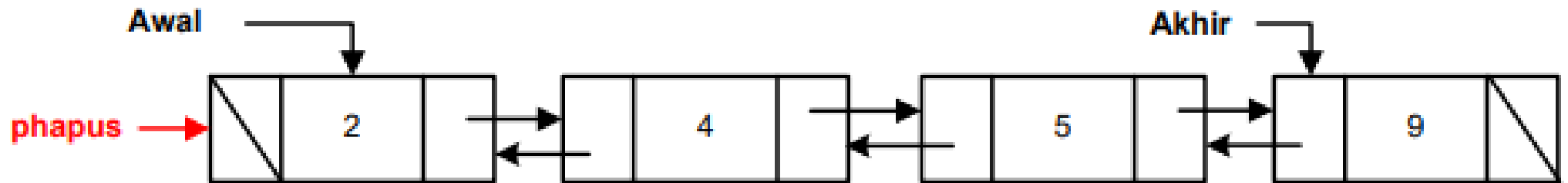
```
if (awal==akhir)
{
    delete awal;
    awal=akhir=NULL;
}
```

# MENGHAPUS SIMPUL DI DEPAN

- Kondisi double linked list memiliki lebih dari 1 data/simpul

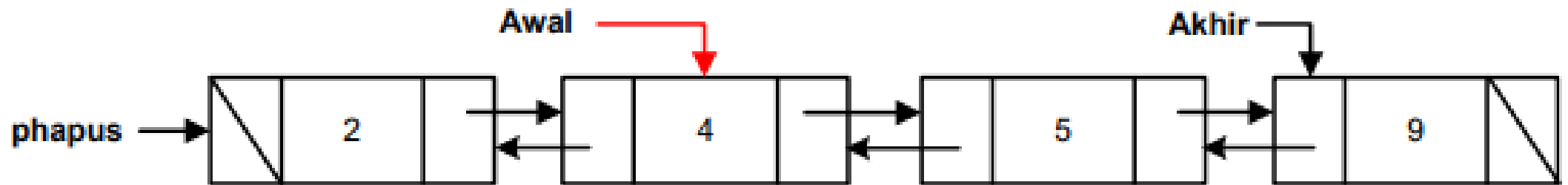
Langkahnya :

1. Tempatkan pointer bantuan (**phapus**) ke simpul yang ditunjuk oleh pointer awal

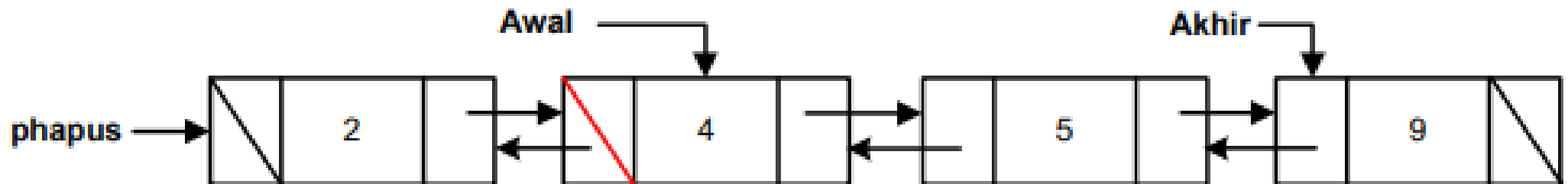


# MENGHAPUS SIMPUL DI DEPAN

2. Pindahkan pointer awal ke simpul tetangga kanannya



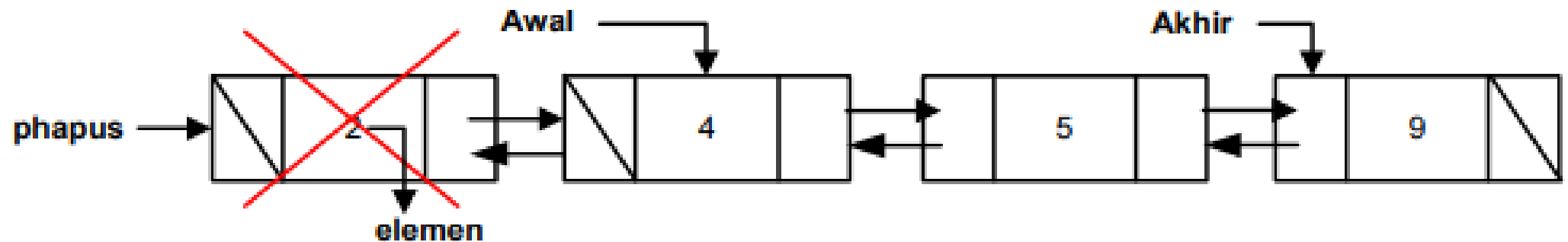
3. Penyambung kiri dari simpul yang ditunjuk oleh pointer awal diberi nilai NULL



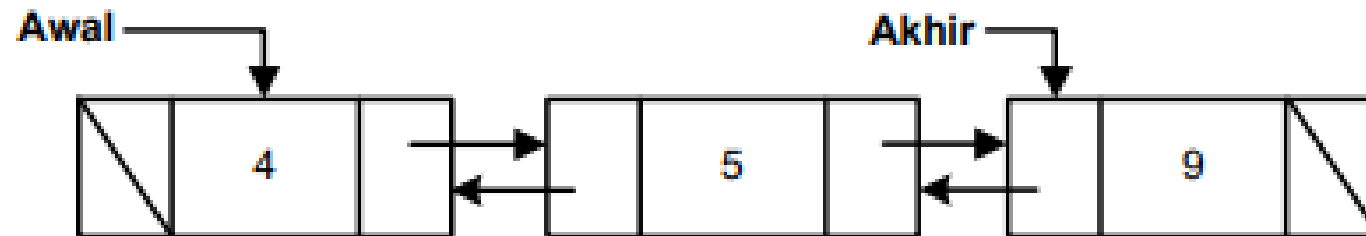


# MENGHAPUS SIMPUL DI DEPAN

4. Hapus simpul yang ditunjuk oleh pointer phapus



5. Setelah dihapus. maka kondisi akhir double linked list adalah seperti berikut :



# MENGHAPUS SIMPUL DI DEPAN

## Algoritma :

phapus=awal;

awal=awal->kanan;

awal->kiri=NULL;

delete phapus;

# MENGHAPUS SIMPUL DI TENGAH

- Operasi ini berguna untuk menghapus data/simpul pada posisi tengah double linked list.
- Untuk melakukan penghapusan data/simpul di tengah double linked list, maka harus diketahui terlebih dahulu data/simpul apa yang akan dihapus
- Ada 4 keadaan yang mungkin terjadi ketika akan melakukan proses penghapusan simpul di tengah, yaitu :
  - Kondisi double linked list masih kosong

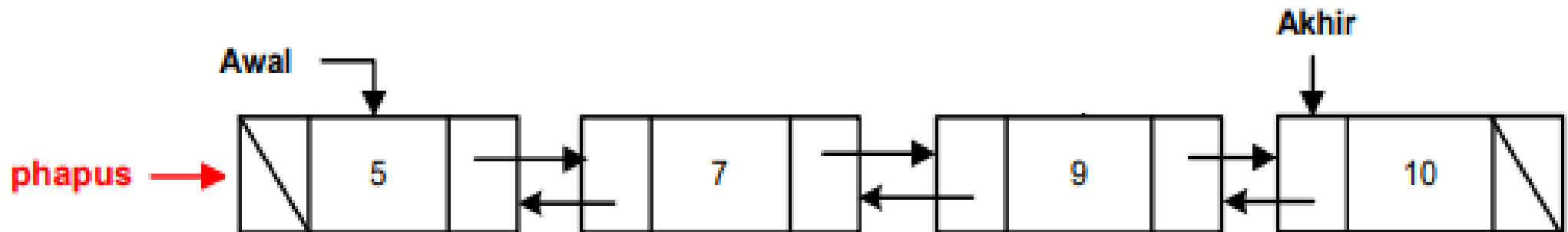
Jika ini terjadi, maka proses penghapusan data/simpul tidak bisa dilakukan karena double linked list masih kosong / tidak ada data/simpul

# MENGHAPUS SIMPUL DI TENGAH

- Kondisi jika data/simpul yang akan dihapus adalah data/simpul pertama  
Ketika kondisi ini terjadi, maka proses yang dilakukan adalah proses penghapusan data/simpul di depan
- Kondisi jika data/simpul yang akan dihapus adalah data/simpul yang terakhir  
Ketika kondisi ini terjadi, maka proses yang dilakukan adalah proses penghapusan data/simpul di belakang

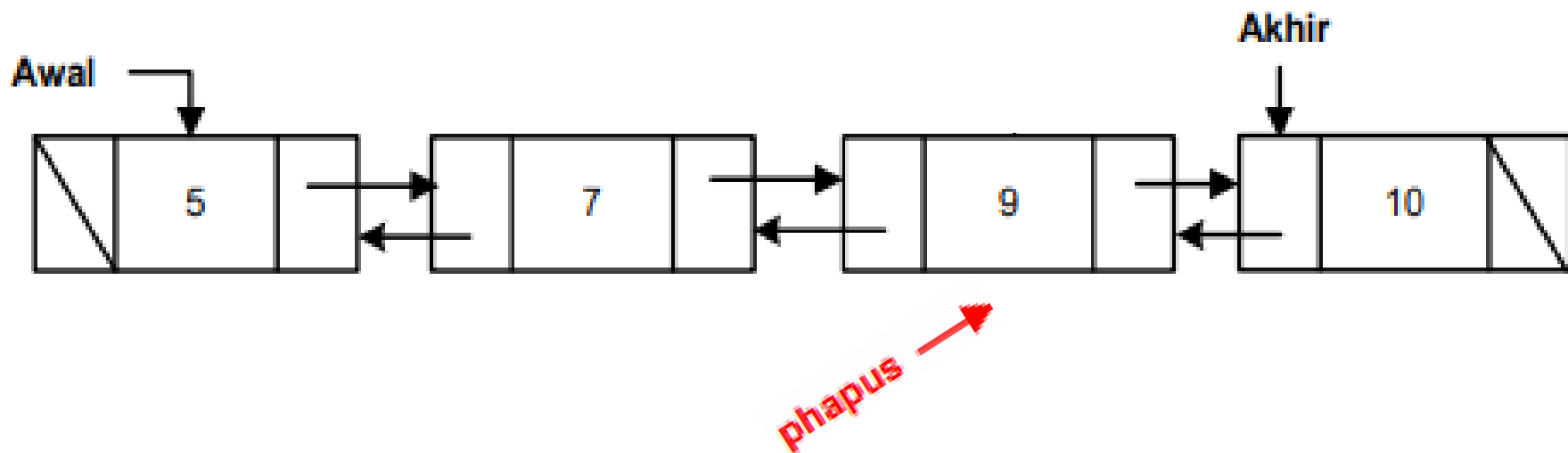
# MENGHAPUS SIMPUL DI TENGAH

- Kondisi jika data/simpul yang akan dihapus adalah data/simpul yang di tengah  
Misal yang akan dihapus adalah simpul yang datanya=9, maka langkahnya :
  1. Tempatkan pointer bantuan (**phapus**) ke simpul yang ditunjuk oleh pointer awal



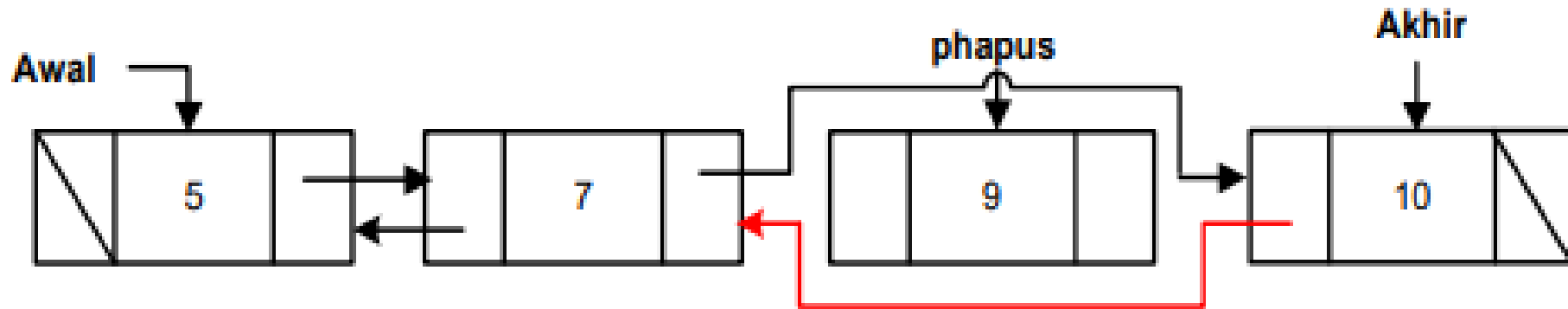
# MENGHAPUS SIMPUL DI TENGAH

2. Secara berulang gerakkan pointer phapus ke simpul dikanannya dan akan berhenti jika data dari simpul yang ditunjuk oleh pointer phapus sudah sama dengan data yang akan dihapus.



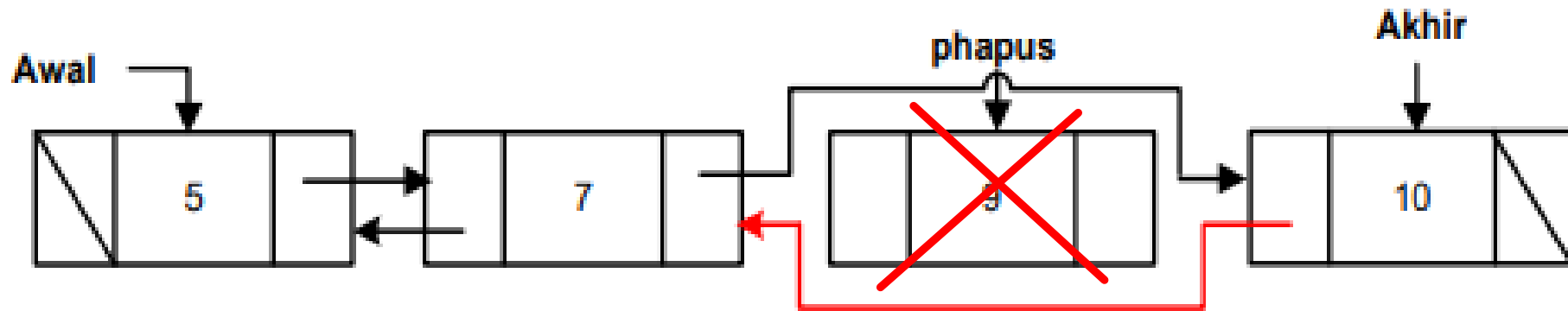
# MENGHAPUS SIMPUL DI TENGAH

3. Kemudian hubungkan penyambung kiri dari simpul yang berada di kanan phapus ke simpul yang berada di kiri phapus. Dan hubungkan pula penyambung kanan dari simpul yang berada di kiri phapus ke simpul yang berada di kanan phapus.



# MENGHAPUS SIMPUL DI TENGAH

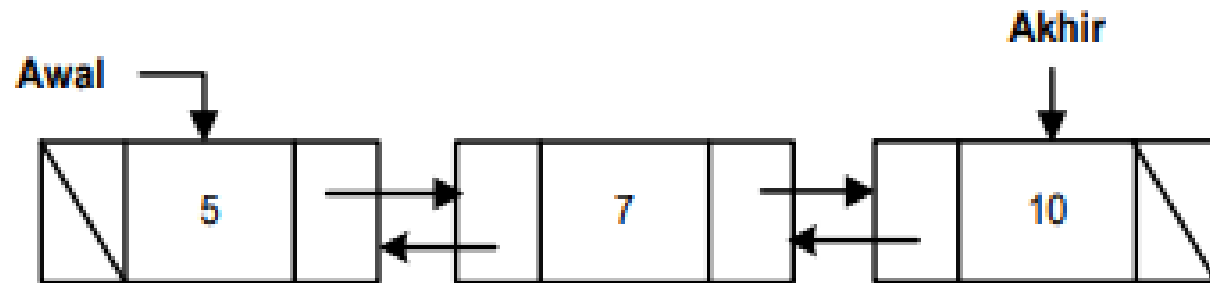
4. Kemudian hapus simpul yang ditunjuk oleh pointer phapus





# MENGHAPUS SIMPUL DI TENGAH

5. Setelah terjadi penghapusan di tengah, maka kondisi akhir double linked list adalah sebagai berikut :



# MENGHAPUS SIMPUL DI TENGAH

## Algoritma :

```
phapus=awal;  
while (phapus->data != hapus)  
    phapus=phapus->next;  
phapus->kiri->kanan=phapus->kanan;  
phapus->kanan->kiri=phapus->kiri;  
delete phapus;
```

# MENGHAPUS SIMPUL DI BELAKANG

- Operasi ini berguna untuk menghapus data/simpul pada posisi belakang double linked list.
- Ada 3 keadaan yang mungkin terjadi ketika akan melakukan proses penghapusan simpul di belakang, yaitu :

- ➔ Kondisi double linked list masih kosong

Jika ini terjadi, maka proses penghapusan data/simpul tidak bisa dilakukan karena double linked list masih kosong / tidak ada data/simpul

# MENGHAPUS SIMPUL DI BELAKANG

- ➔ Kondisi double linked list hanya memiliki 1 data/simpul

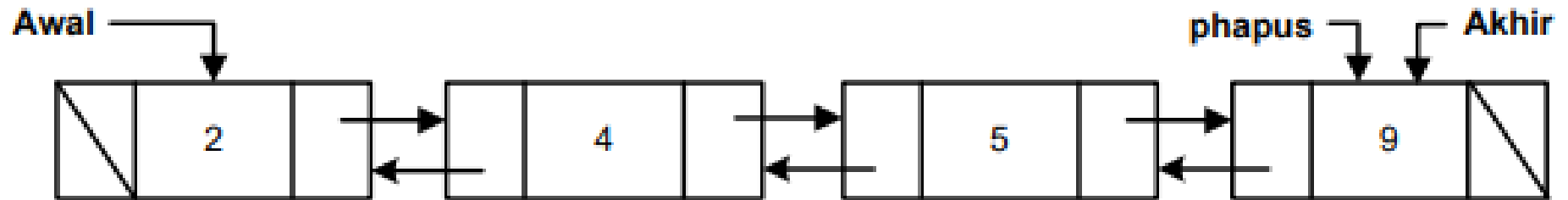
Penghapusan di belakang, prosesnya sama seperti penghapusan di depan

# MENGHAPUS SIMPUL DI BELAKANG

- Kondisi double linked list memiliki lebih dari 1 data/simpul

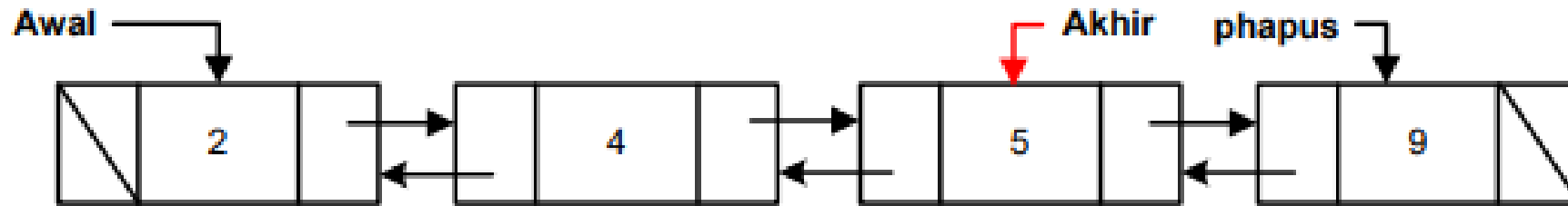
Langkahnya :

1. Tempatkan pointer bantuan (**phapus**) ke simpul yang ditunjuk oleh pointer akhir

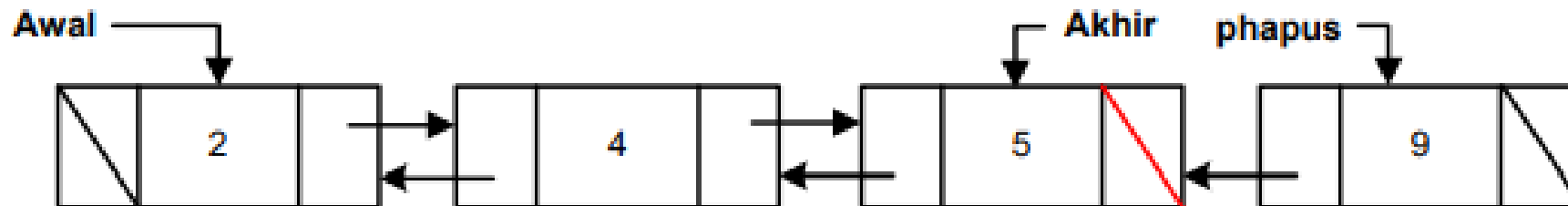


# MENGHAPUS SIMPUL DI BELAKANG

2. Pindahkan pointer akhir ke simpul sebelumnya

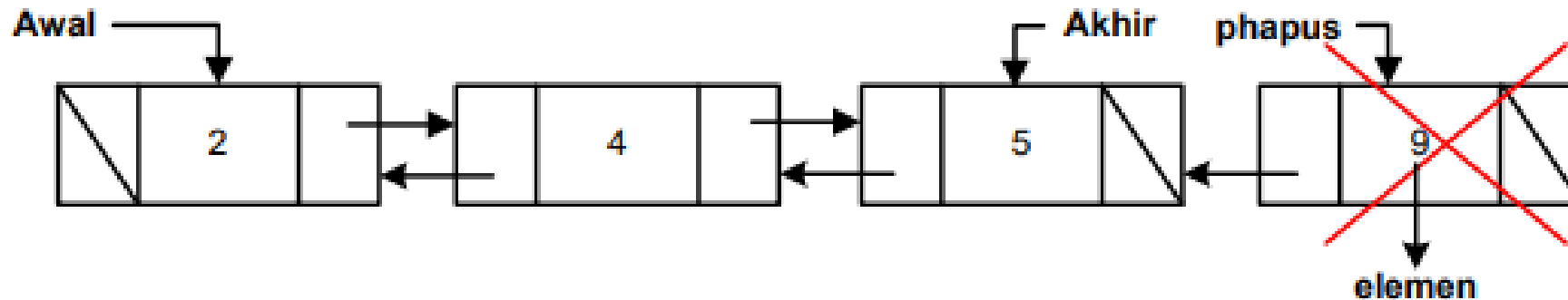


3. Penyambung kanan dari pointer akhir diberi nilai NULL

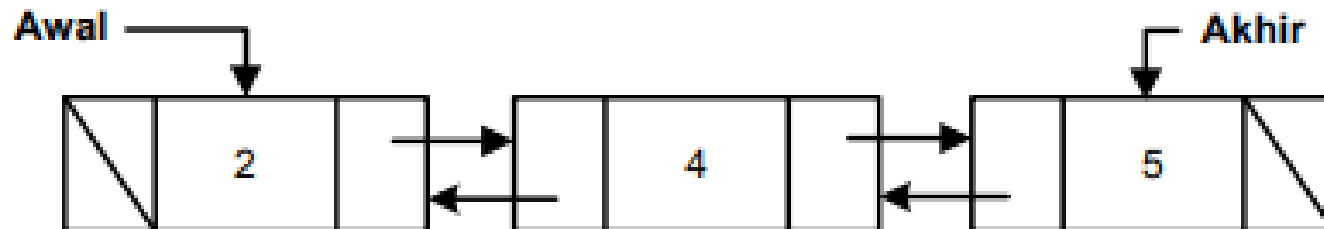


# MENGHAPUS SIMPUL DI BELAKANG

4. Hapus simpul yang ditunjuk oleh pointer phapus



5. Setelah simpul dihapus, maka kondisi double linked list adalah sebagai berikut :



# MENGHAPUS SIMPUL DI BELAKANG

## Algoritma :

phapus=akhir;

akhir=akhir->kiri;

akhir->kanan=NULL;

delete phapus;