

Struktur Data

Saniati,S.ST.,M.T.

EPISODE 4A

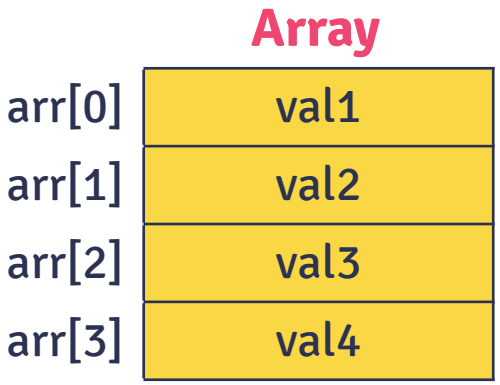
Single Linked List

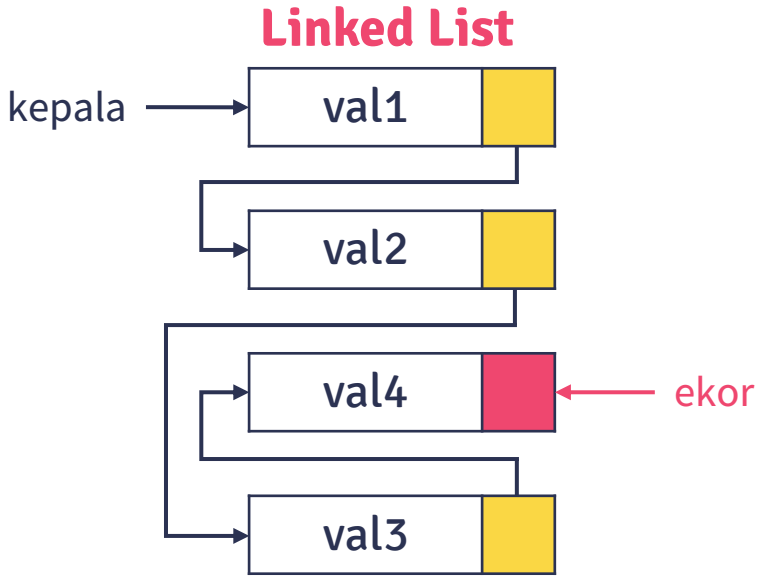
Linked List ?

- Linked List adalah elemen yang berurutan yang dihubungkan dengan pointer.
- Elemen terakhir menunjuk ke NULL (untuk Linked List non Circular).
- Elemen pada Single Linked List dapat bertambah atau berkurang (dinamis) selama program dijalankan.
- Dapat dibuat selama diperlukan (hingga memori sistem habis).
- Linked List tidak membuang ruang memori (tetapi membutuhkan beberapa memori ekstra untuk pointer).

Array vs Linked List ?

- Array memiliki ruang atau aksesbilitas yang terbatas. Sedangkan Linked bisa meng-Alokasi memori secara dinamis.



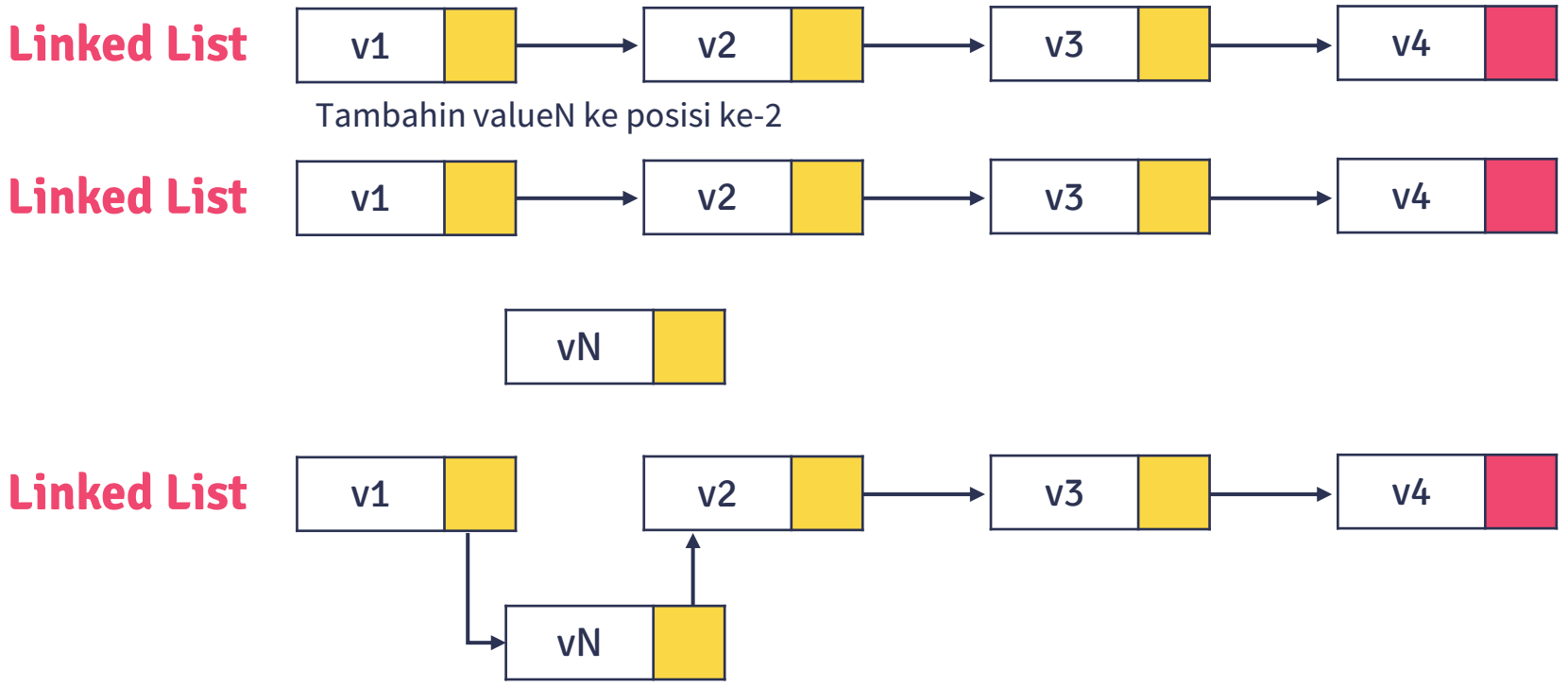


Array vs Linked List ?

- Array memiliki ruang atau aksesbilitas yang terbatas. Sedangkan Linked bisa meng-Alokasi memori secara dinamis.

Array vs Linked List ?

- Array memiliki ruang atau aksesbilitas yang terbatas. Sedangkan Linked bisa meng-Alokasi memori secara dinamis.



Tipe Linked List ?

- Single Linked List (Singly Linked List).
- Double Linked List (Doubly Linked List).
- Circular Linked List.
- Multiple Linked List.

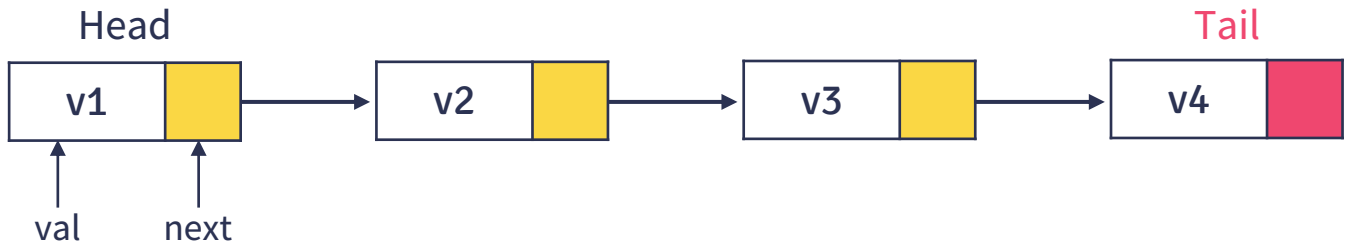
Single Linked List ?

- Single Linked List merupakan suatu linked list yang hanya memiliki satu variabel pointer saja. Dimana pointer tersebut menunjuk ke node selanjutnya dan pointer pada tail menunjuk ke NULL.

- Navigasi item maju saja.

- Single Linked List terdiri dari sejumlah elemen (node) dimana setiap node memiliki penunjuk berikutnya ke elemen (node) berikutnya.

- Penunjuk node terakhir adalah NULL, yang menunjukkan akhir dari Single Linked List.



Deklarasi & Inisialisasi ?

```
/*
  struct LinkListName{
    // komponen / member
    dataTypeData1 dataName1;
    . . .
    LinkListName *next;
  };
*/
```

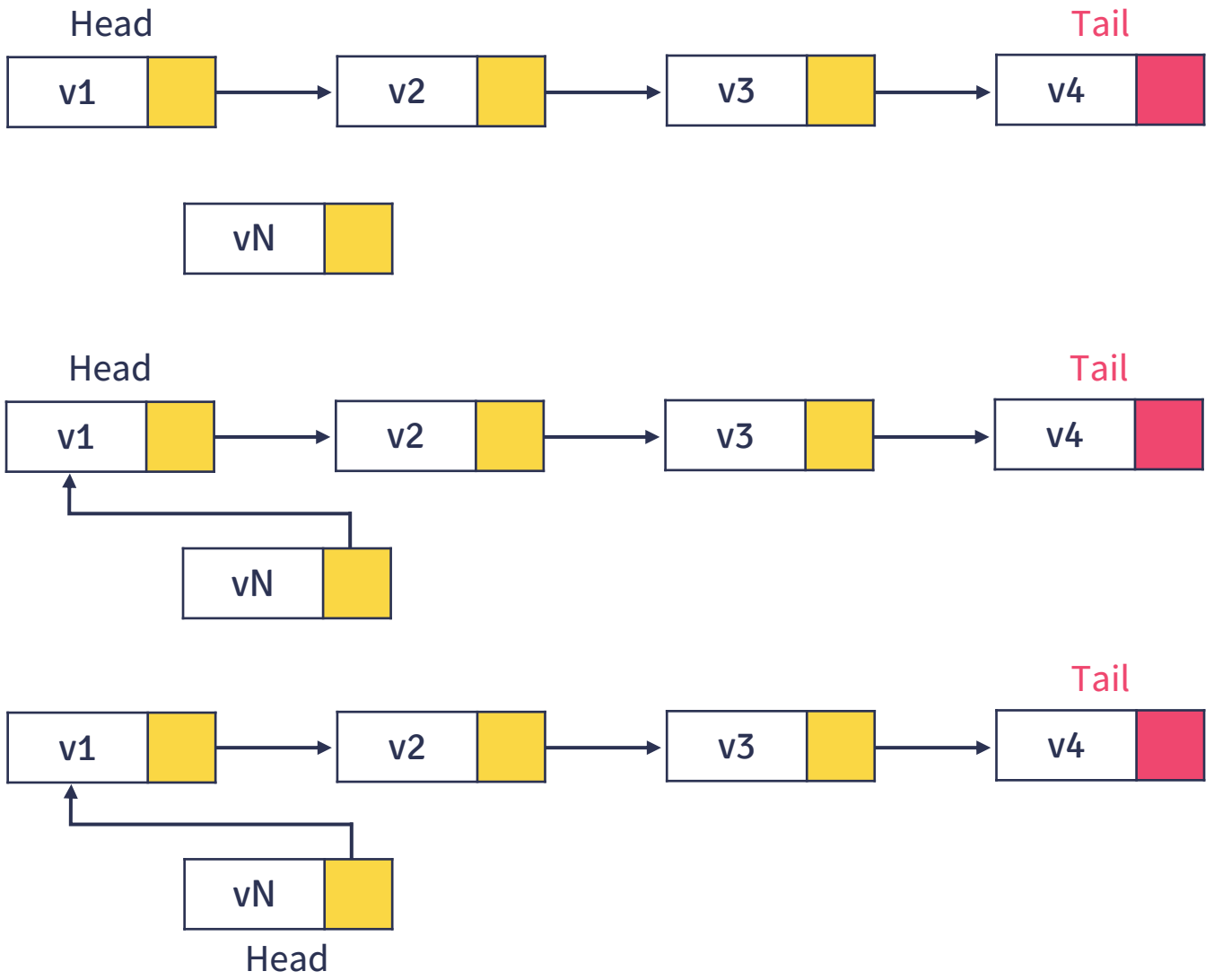
```
int main()
{
  /*
    LinkListName *node1, *nodeN;
    node1 = (LinkListName*) malloc(sizeof(LinkListName));
    nodeN = new LinkListName();
  */
  /*
    node1->dataName1 = valData1;
    . . .
    node1->next = nodeN;

    nodeN->dataNameN = valDataN;
    . . .
    nodeN->next = NULL;
  */
}
```

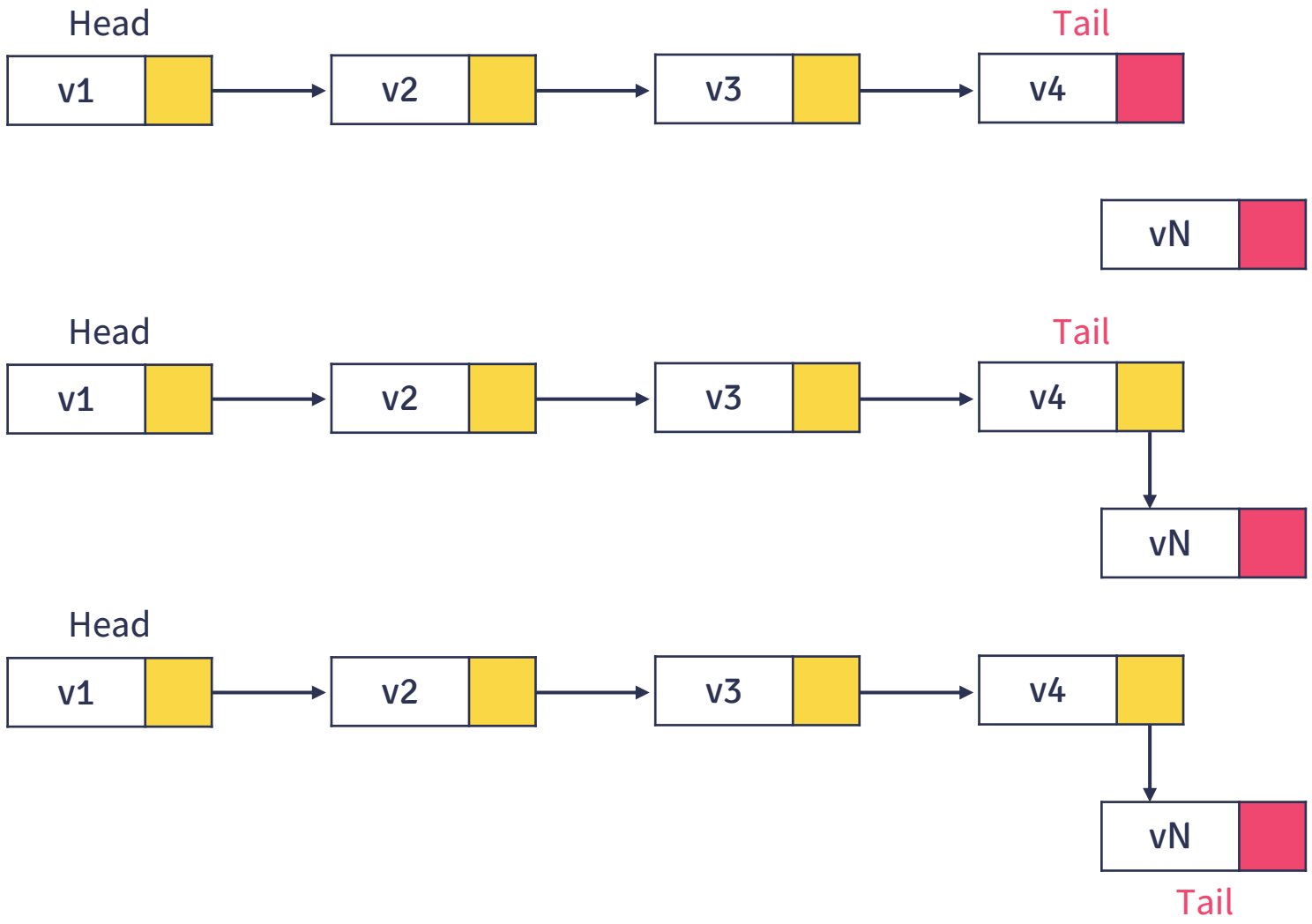
Print Single Linked List ?

```
/*
  LinkListName *cur;
  cur = node1;
  while( cur != NULL ){
    // print
    cur = cur->next;
  }
*/
```

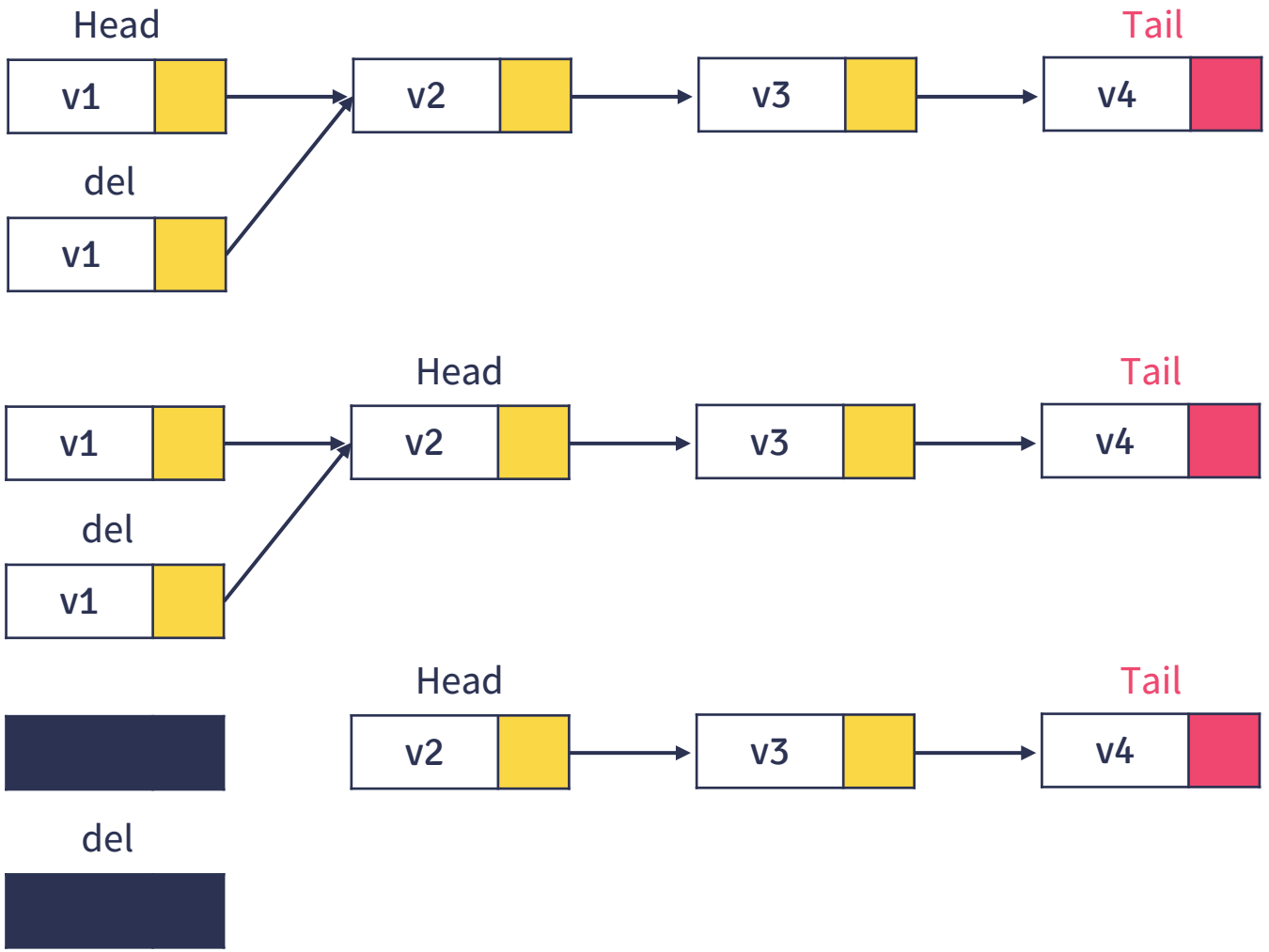
Added at Beginning Node ?



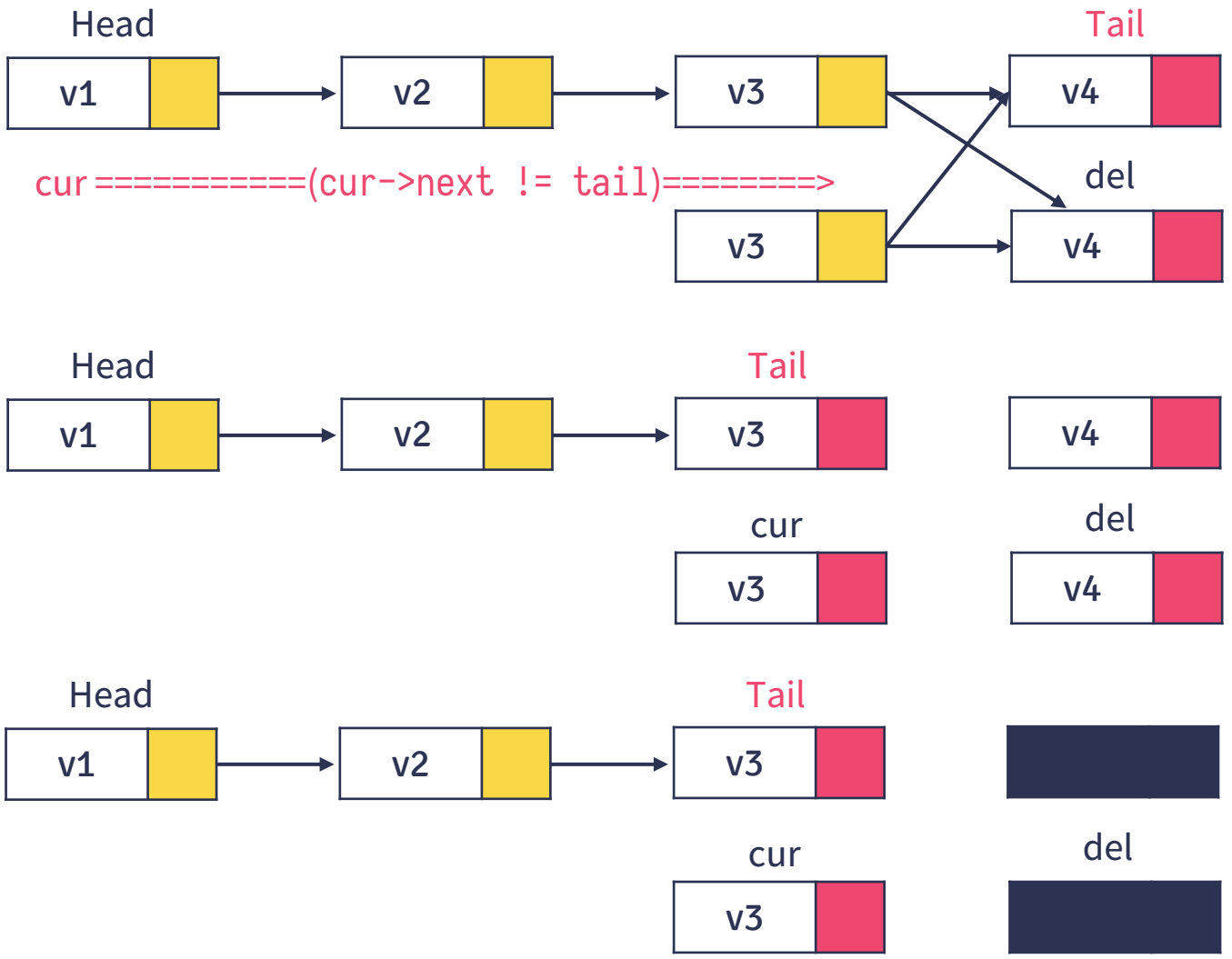
Added at Last Node ?



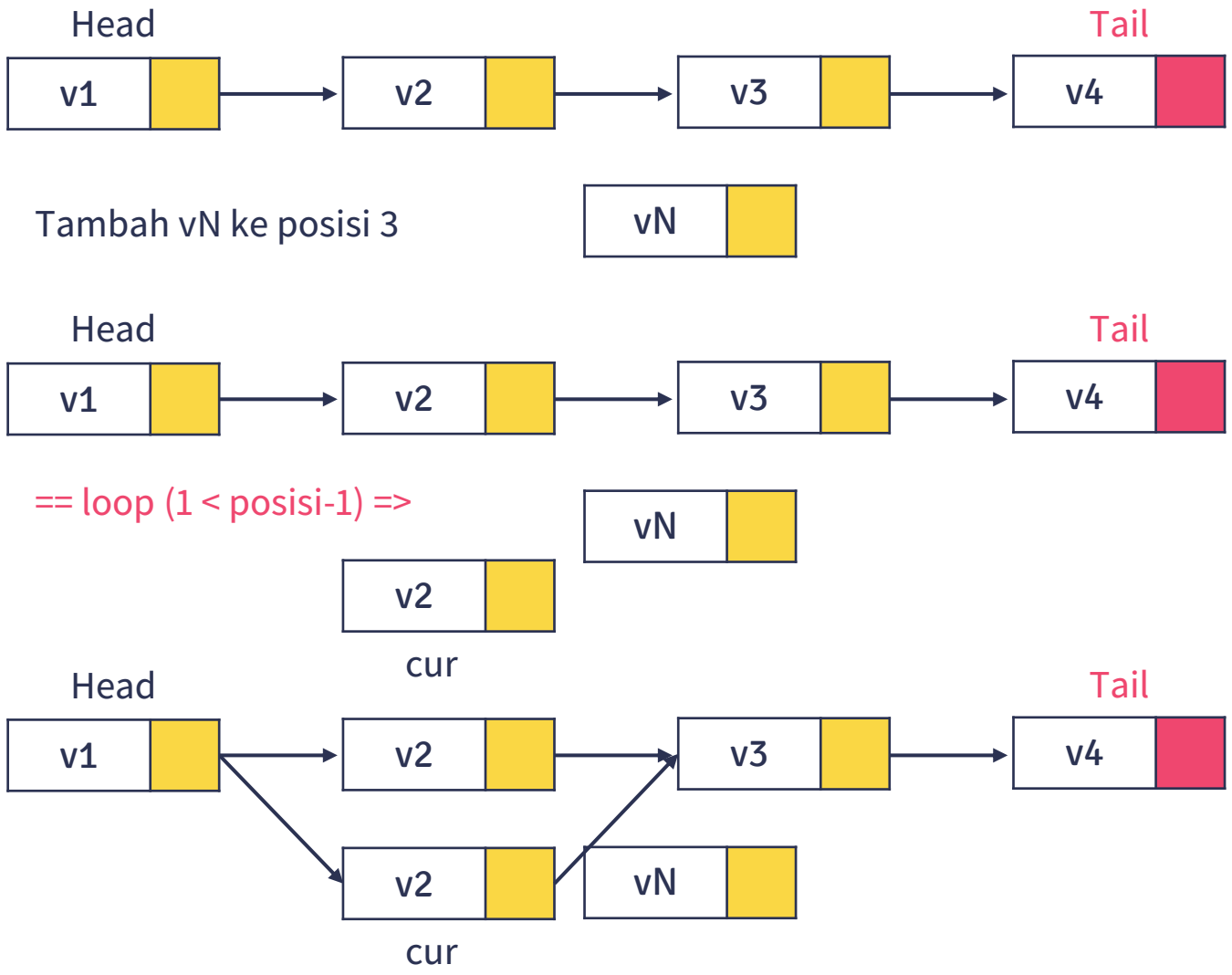
Delete the First Node ?



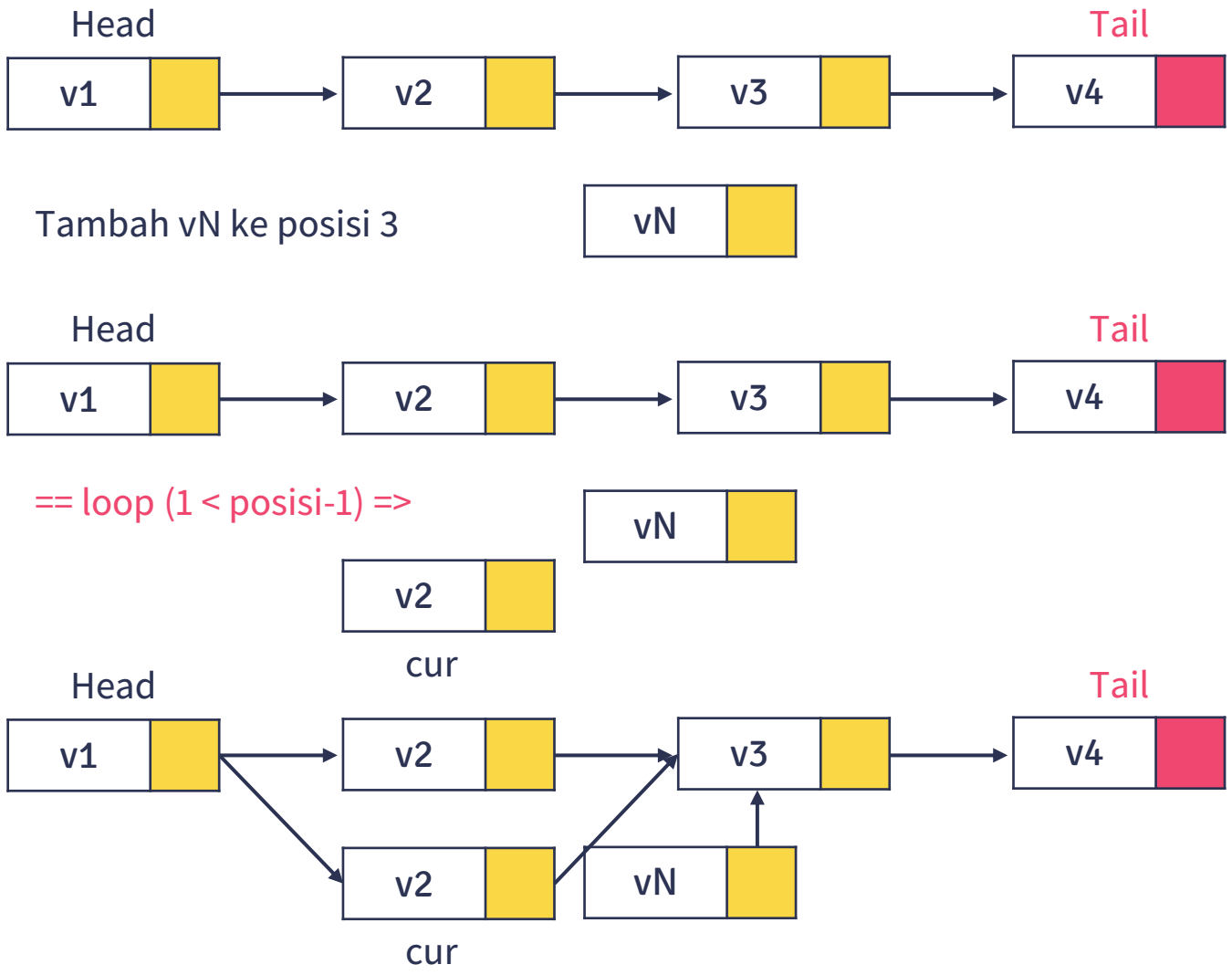
Delete the Last Node ?



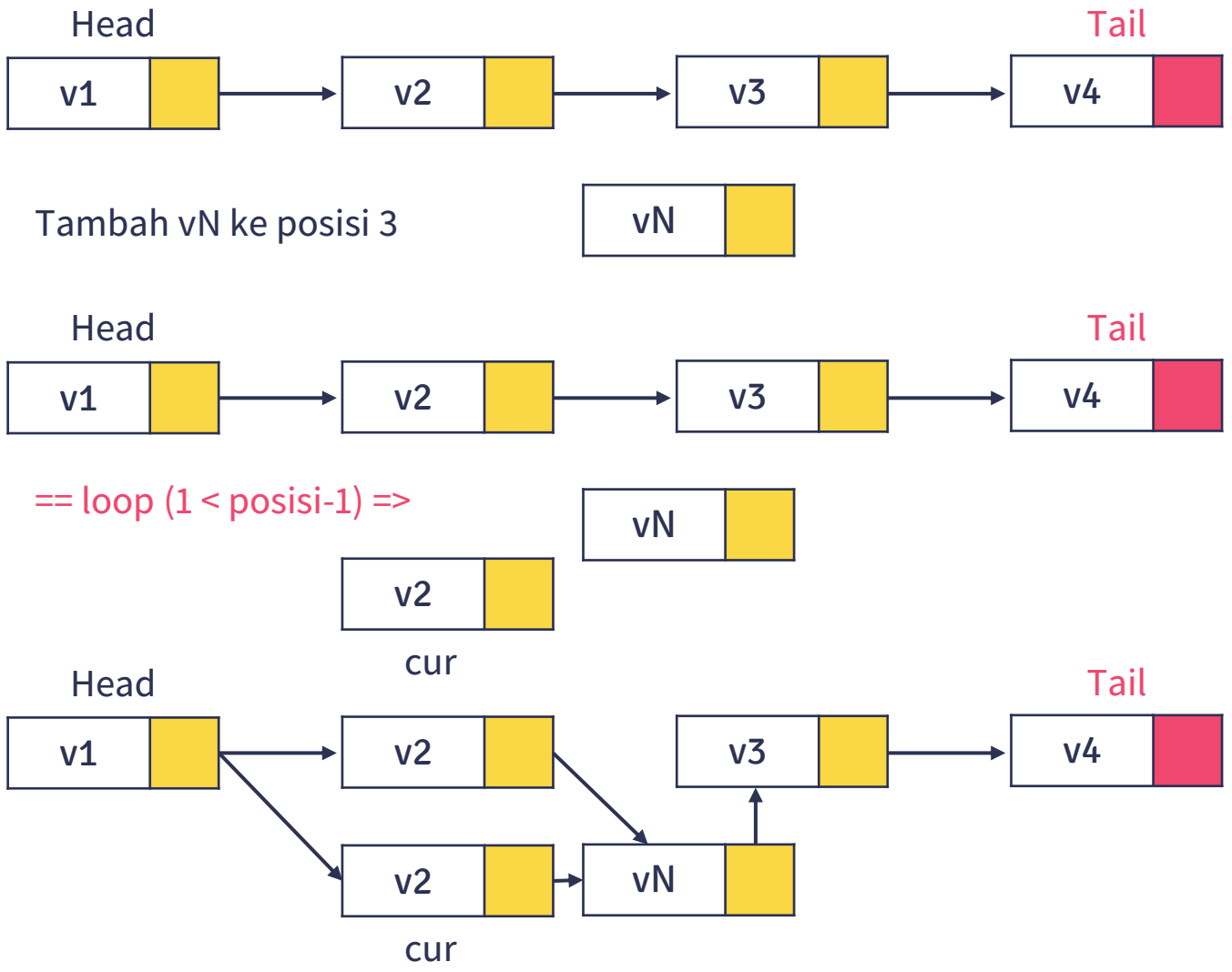
Added at Middle Node ?



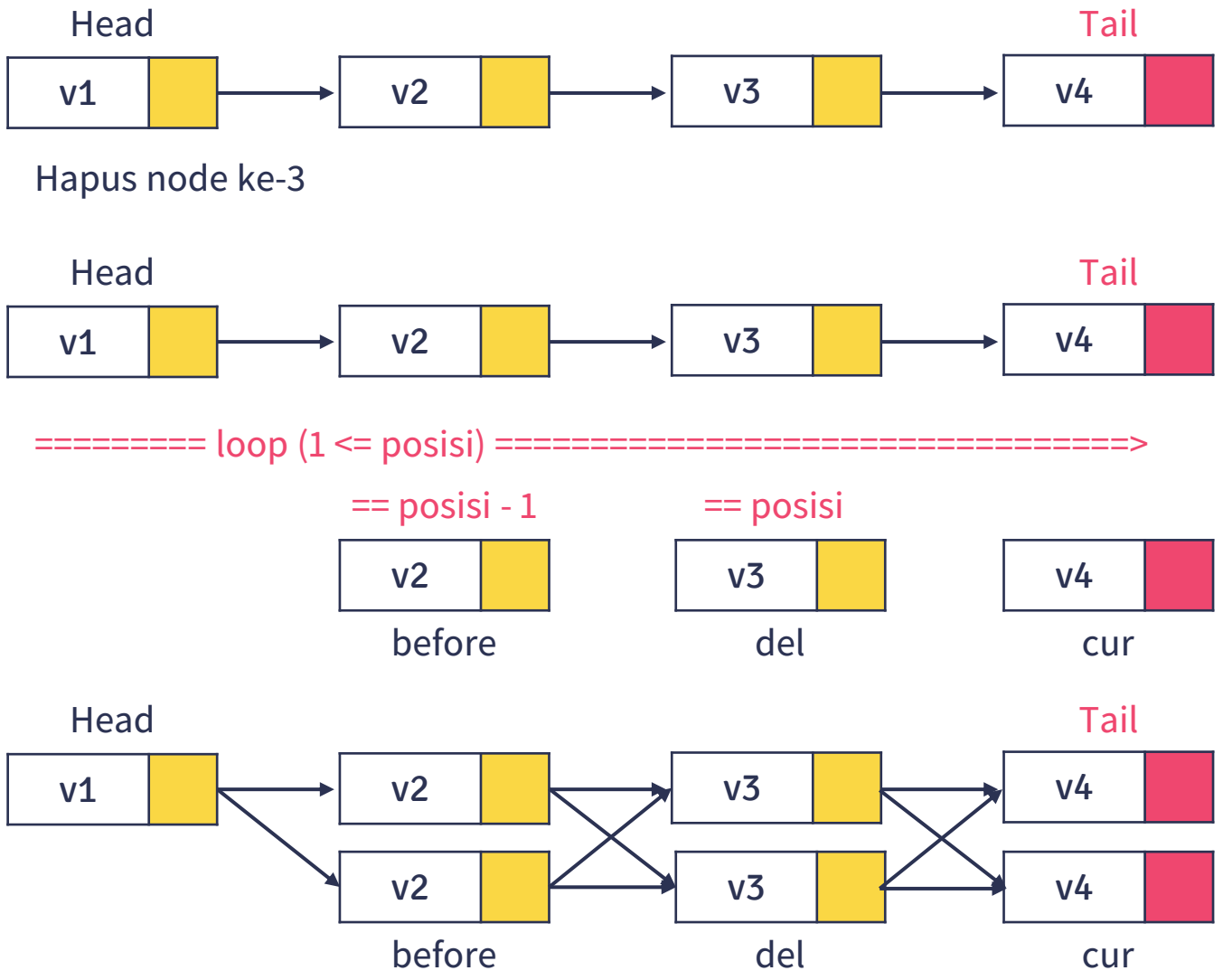
Added at Middle Node ?



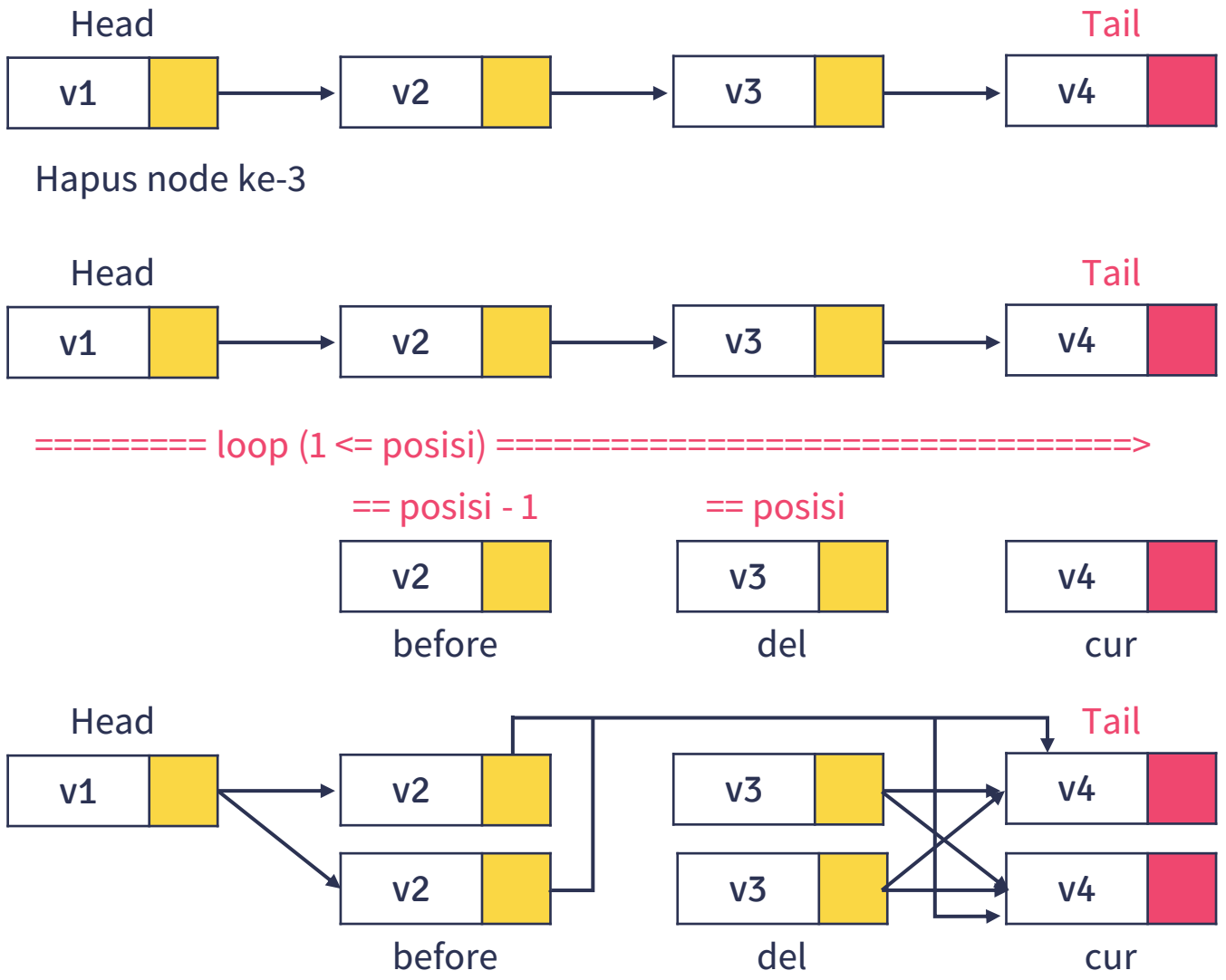
Added at Middle Node ?



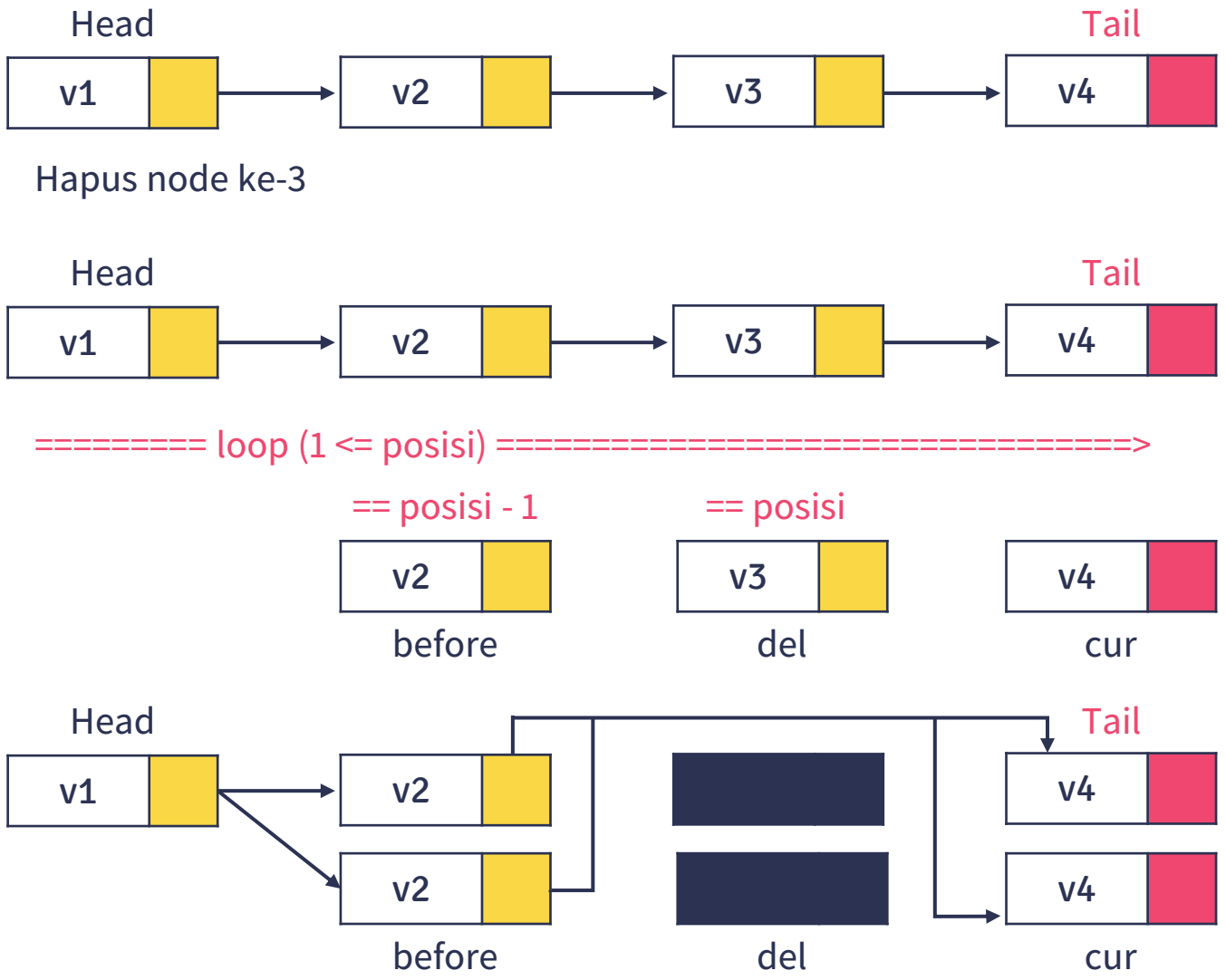
Delete at Middle Node ?



Delete at Middle Node ?



Delete at Middle Node ?



Video Selanjutnya

Double Linked List

Thank you

#KEEPLEARNING
#KEEPSPIRITS