STRUKTUR DATA

LIST REPRESENTASI STATIS

Rosa Ariani Sukamto

ROSA ARIANI SUKAMTO

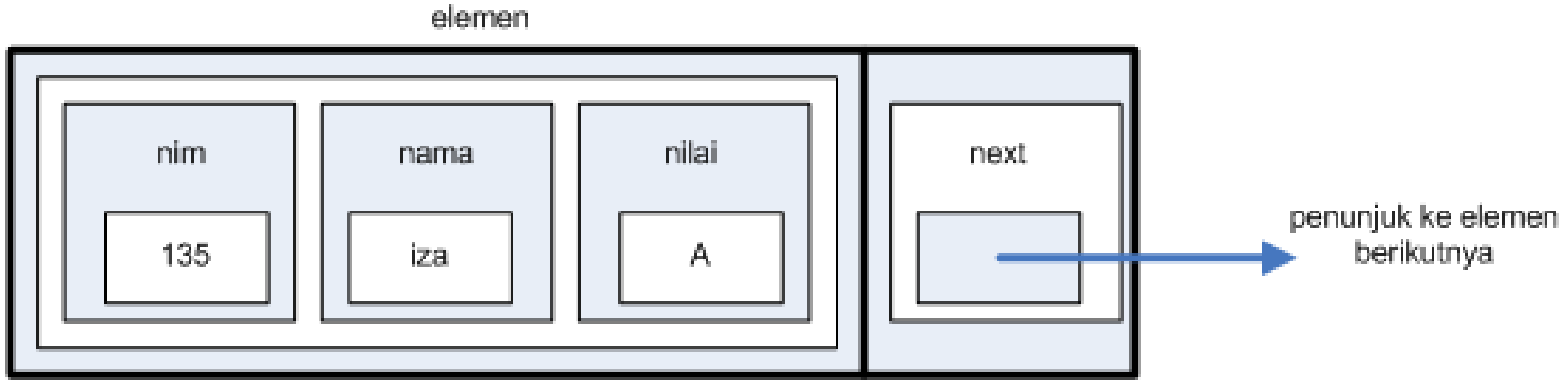
Blog: http://hariiniadalahhadiah.wordpress.com

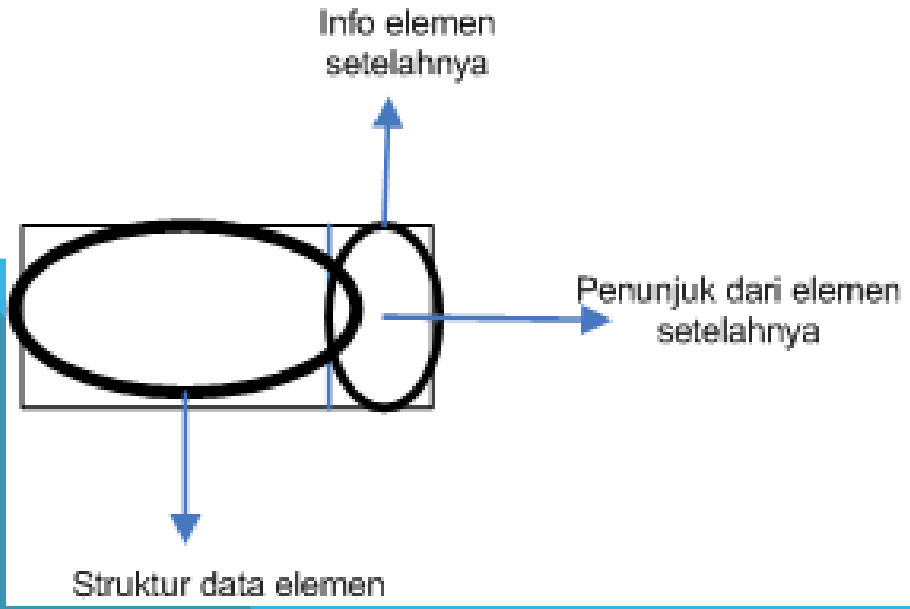
Facebook: https://www.facebook.com/rosa.ariani.sukamto

Email: rosa_if_itb_01@yahoo.com

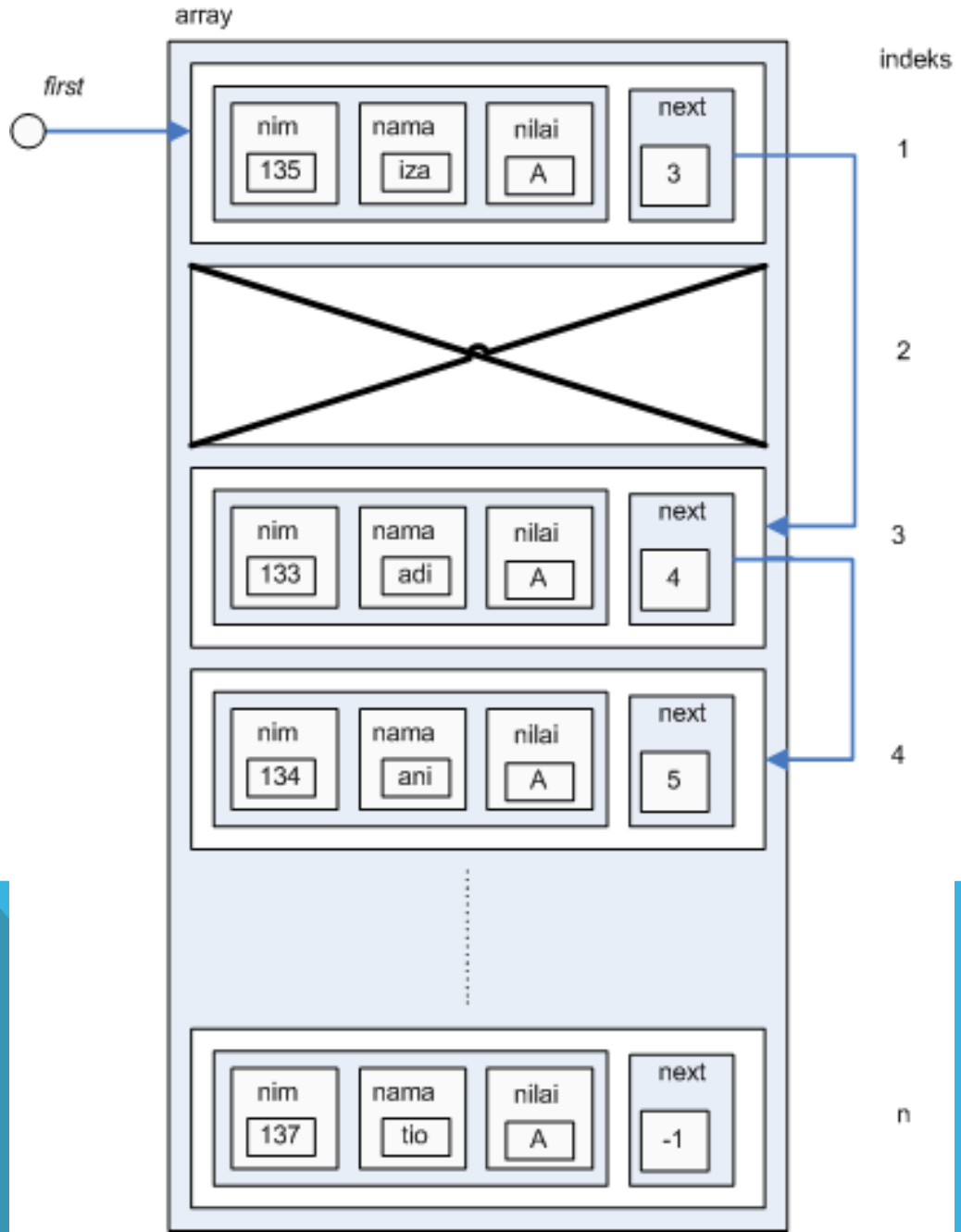
Jangan Lupa untuk Tawadhu

KONTAINER ELEMEN





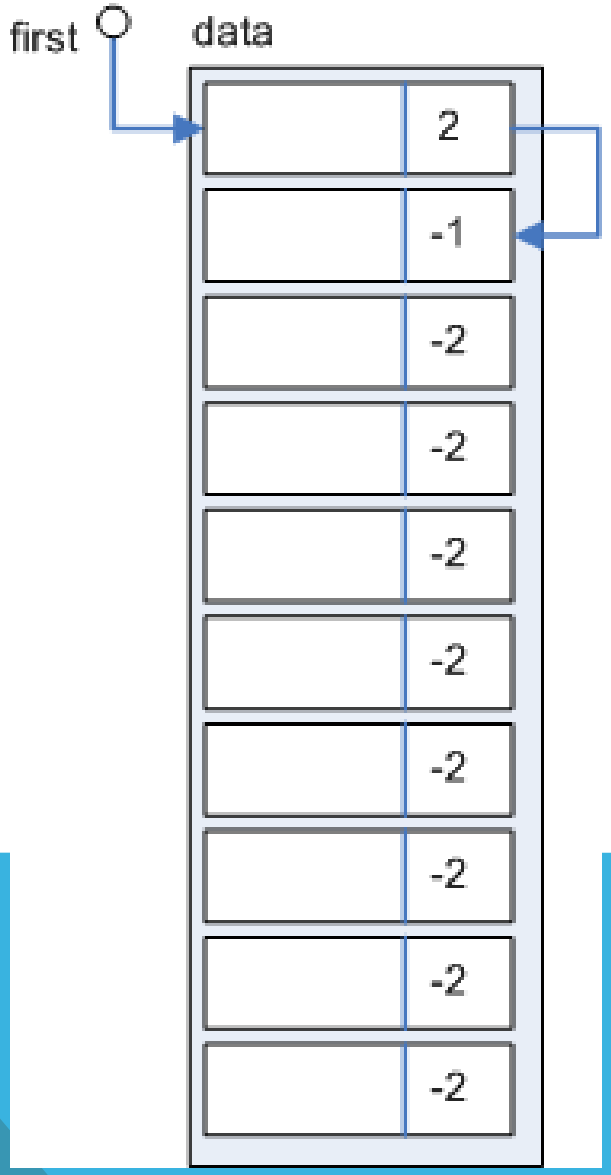
LIST REPRESENTASI STATIS (1)



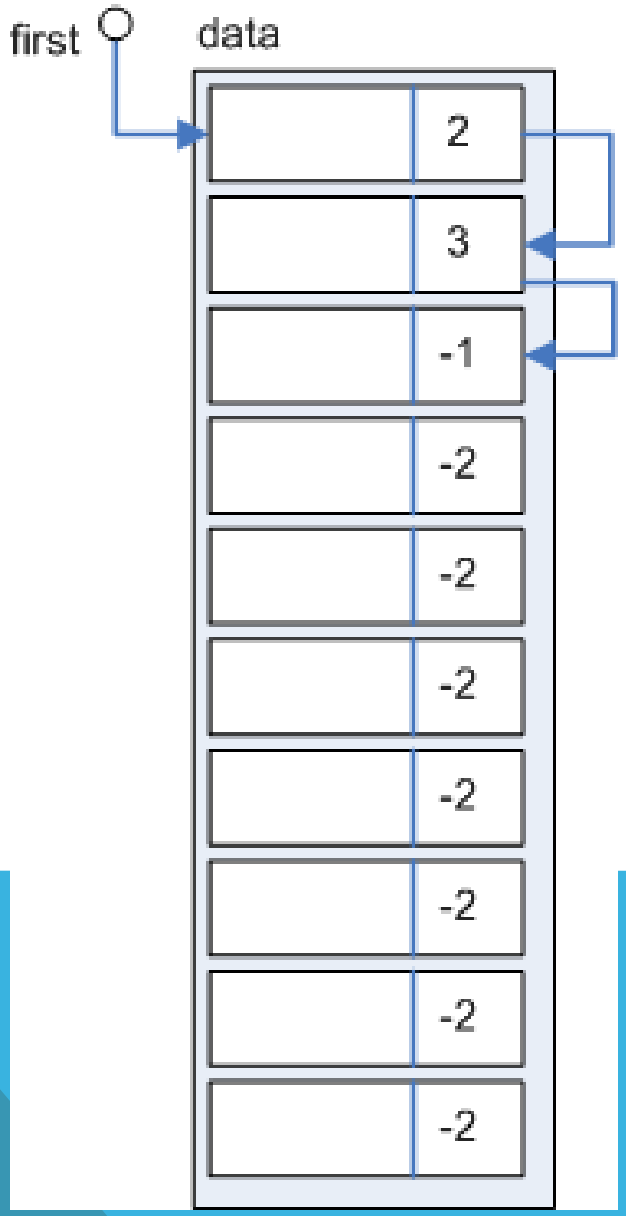
LIST REPRESENTASI STATIS (2) - CREATE LIST

LIST REPRESENTASI STATIS (3) - ADD FIRST

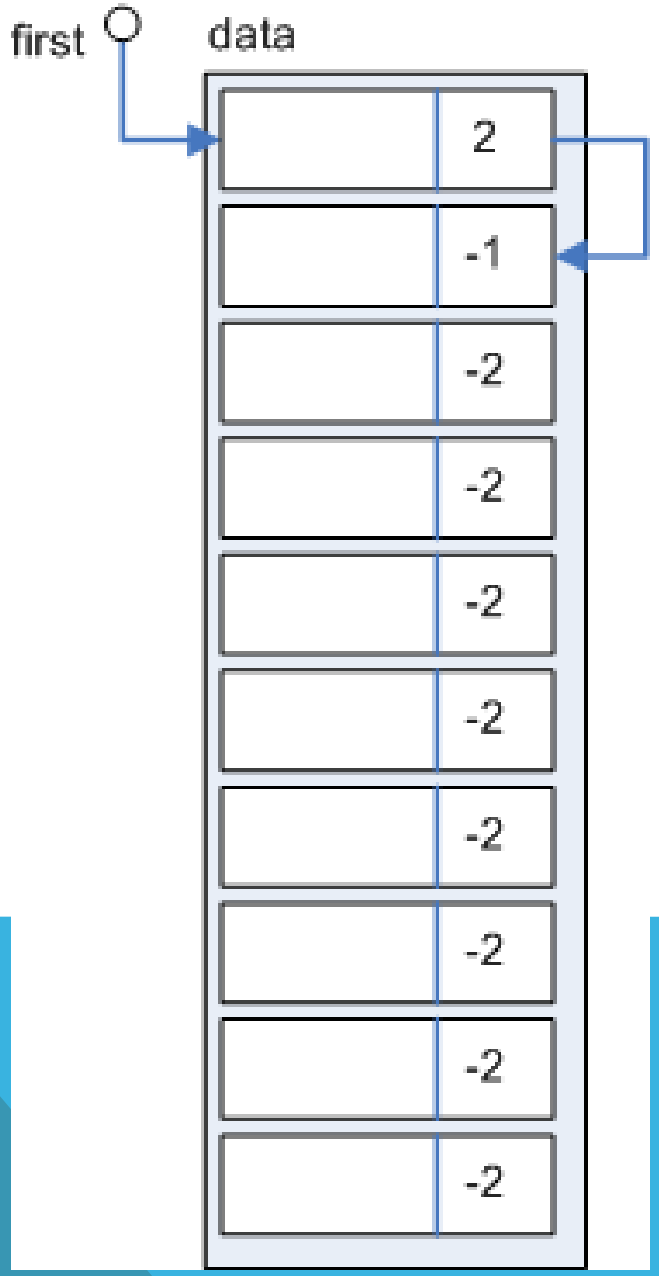
LIST REPRESENTASI STATIS (4) - ADD AFTER



LIST REPRESENTASI STATIS (5) - ADD LAST



LIST REPRESENTASI STATIS (6) - DEL LAST



LIST REPRESENTASI STATIS (7) - DEL AFTER

LIST REPRESENTASI STATIS (8) - DEL FIRST

DEKLARASI ELEMEN

```
#include <stdio.h>
#include <string.h>

typedef struct{
  char  nim[10];
  char nama[50];
  char nilai[2];
}nilaiMatKul;

typedef struct{
  nilaiMatKul elmt;
  int next;
}elemen;

typedef struct{
  int first;
  elemen data[10];
}list;
```

CREATE LIST

```
void createList(list *L){

  (*L).first = -1;
  int i;

  for(i=0;i<10;i++){
    /*proses menginisialisasi isi array*/
    (*L).data[i].next = -2;
  }
}
```

COUNT ELEMENT

```
int countElement(list L){
  int hasil = 0;
  if(L.first != -1){
    /*list tidak kosong*/
    int elmt;

    /*inisialisasi*/
    elmt = L.first;

    while(elmt != -1){
      /*proses*/
      hasil = hasil + 1;

      /*iterasi*/
      elmt = L.data[elmt].next;
    }
  }
```

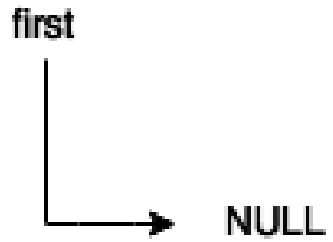
```
    return hasil;
}
```

EMPTY ELEMENT

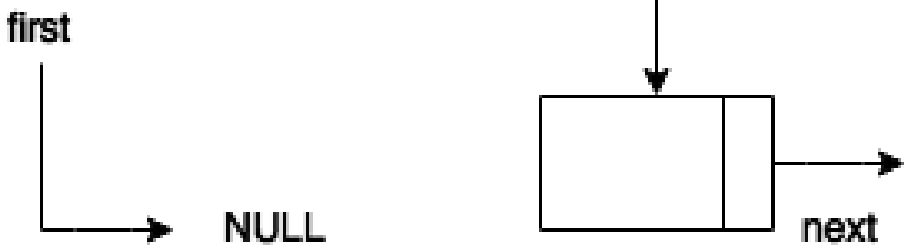
```
int emptyElement(list L){
  int hasil = -1;

  if(countElement(L) < 10){
    int ketemu = 0;

    int i = 0;
    while((ketemu == 0)&&(i <
 10)){
      if(L.data[i].next == -2){
        hasil = i;
        ketemu = 1;
      }
      else{
        i = i + 1;
      }
    }
  }
    return hasil;
}
```

PENAMBAHAN ELEMEN DI AWAL LIST (ADDFIRST) LIST KOSONG

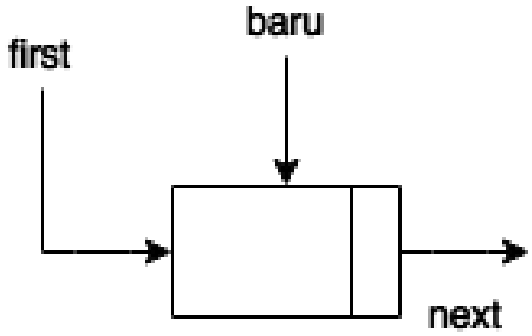
1



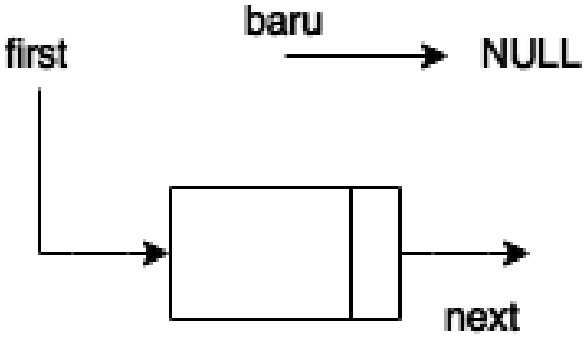
2



3

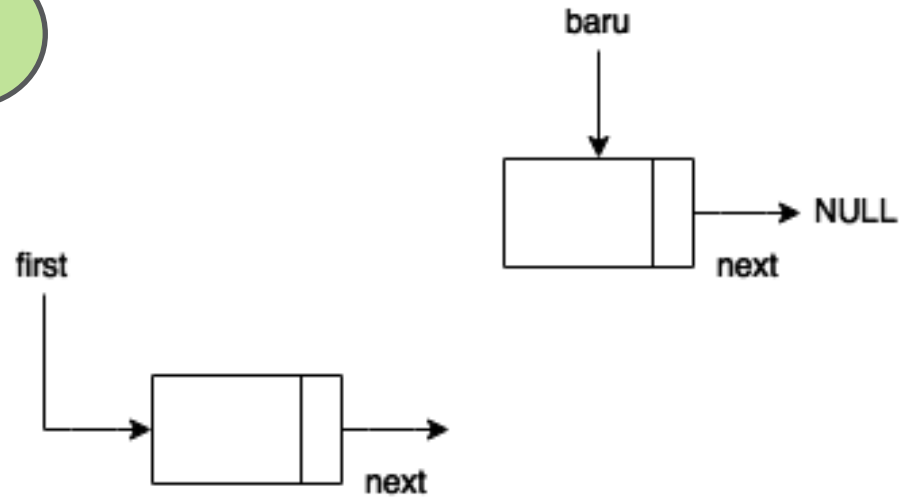


4

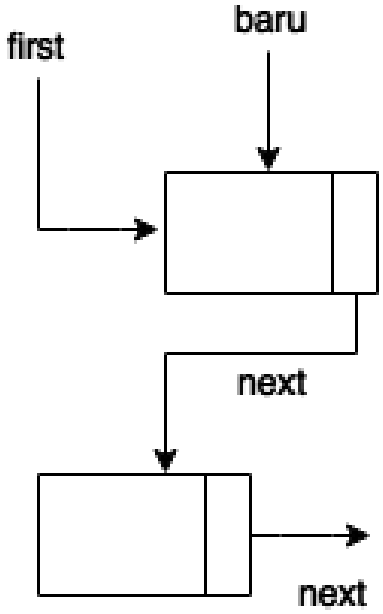


PENAMBAHAN ELEMEN DI AWAL LIST (ADDFIRST)
BANYAK ELEMEN

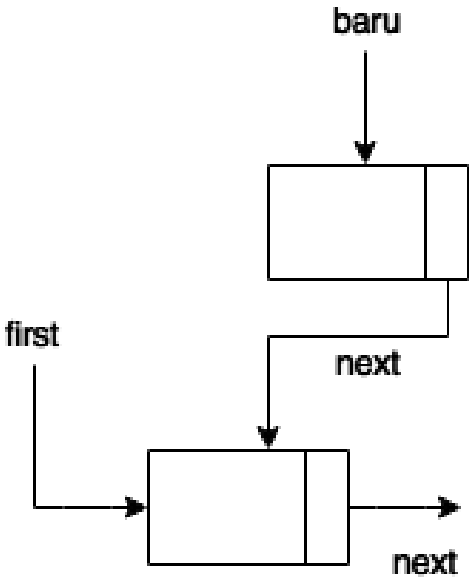
1



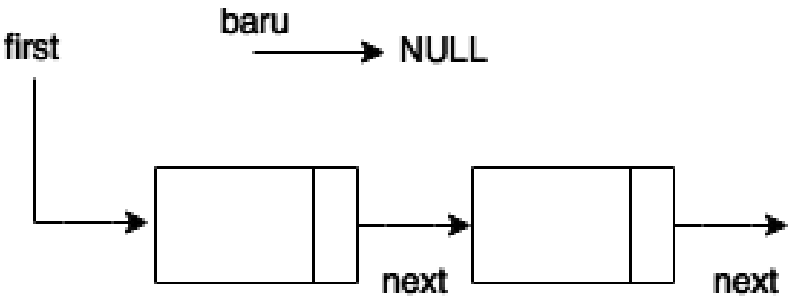
3



2



4



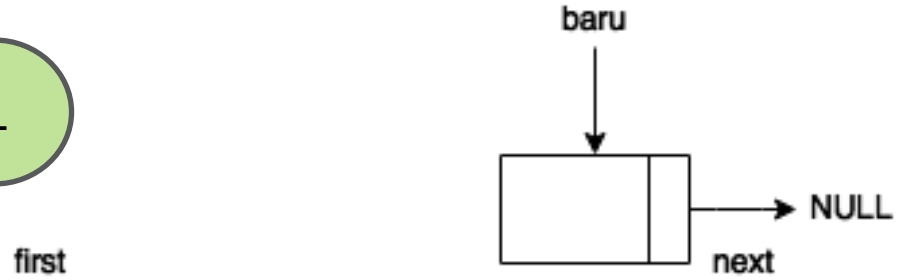
ADD FIRST

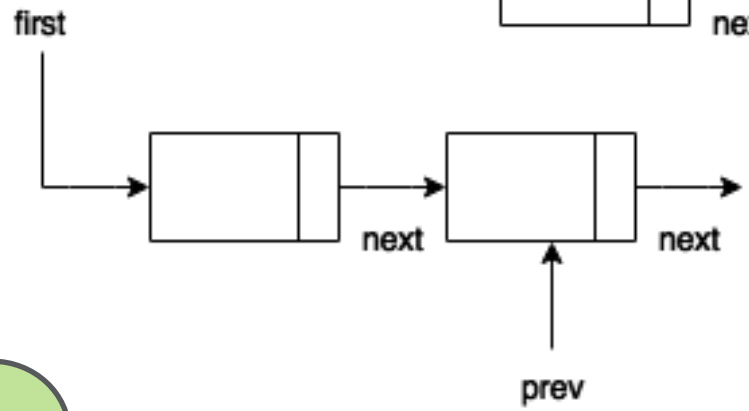
```
void addFirst(char nim[], char
   nama[], char nilai[], list *L){
  if(countElement(*L) < 10){
    int baru = emptyElement(*L);

   strcpy((*L).data[baru].elmt.nim,
   nim);
   strcpy((*L).data[baru].elmt.nama,
   nama);
   strcpy((*L).data[baru].elmt.nilai
   , nilai);

    if((*L).first == -1){
      /*jika list kosong*/
    (*L).data[baru].next = -1;
    }
```

```
  else{
       /*jika list tidak
   kosong*/
    (*L).data[baru].next =
   (*L).first;
    }

    (*L).first = baru;
  }
  else{
    /*proses jika array
   penuh*/
    printf("sudah tidak dapat
   ditambah\n");
  }
}
```

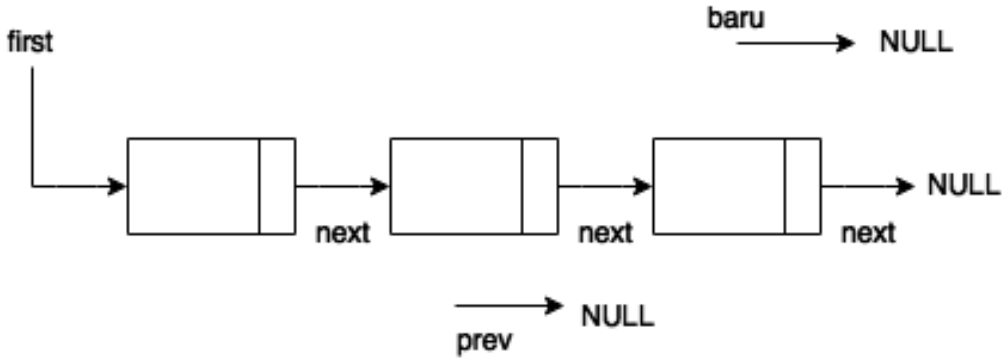
PENAMBAHAN ELEMEN DI TENGAH (ADDAFTER) JIKA ELEMEN SEBELUM ADALAH ELEMEN TERAKHIR

1

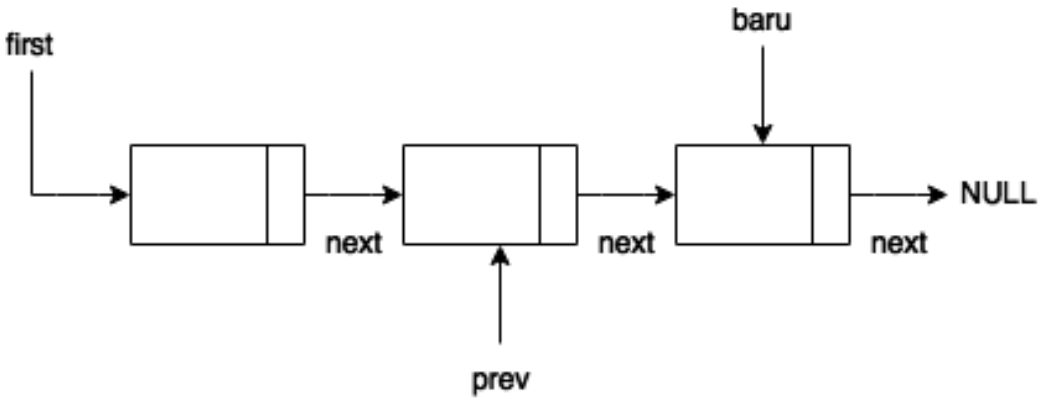




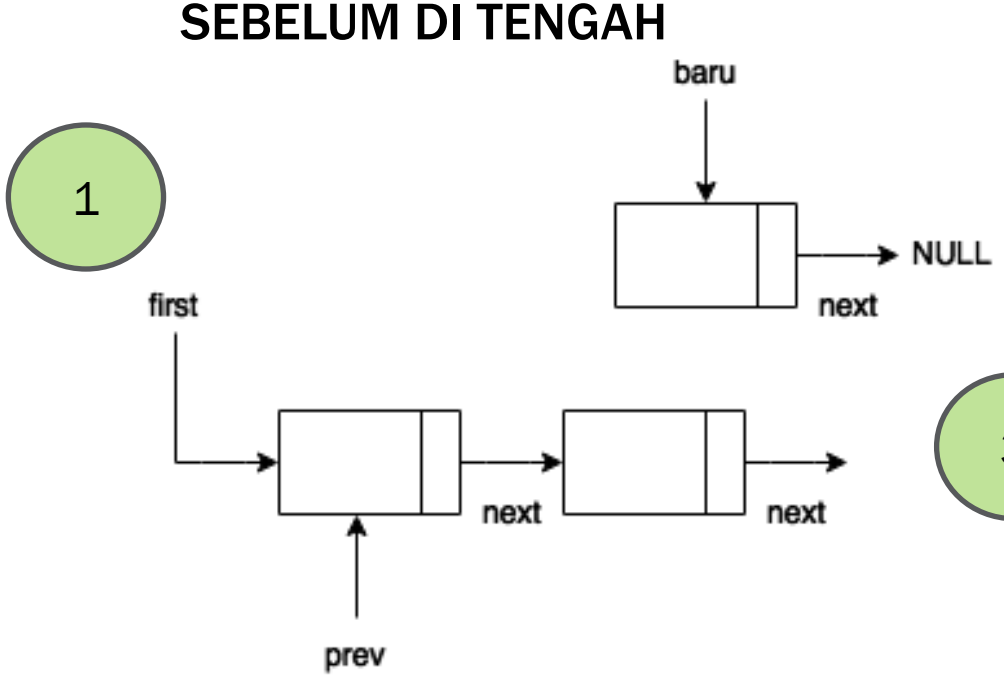
3

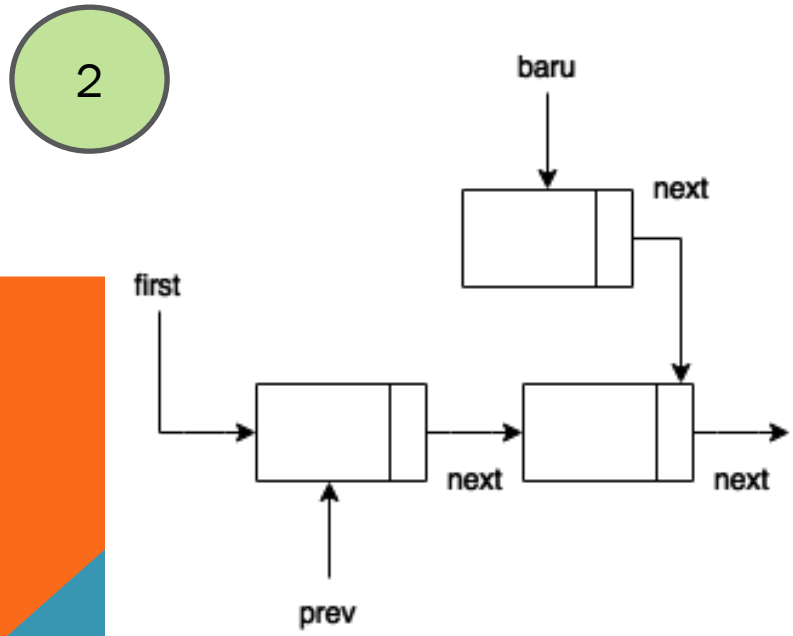


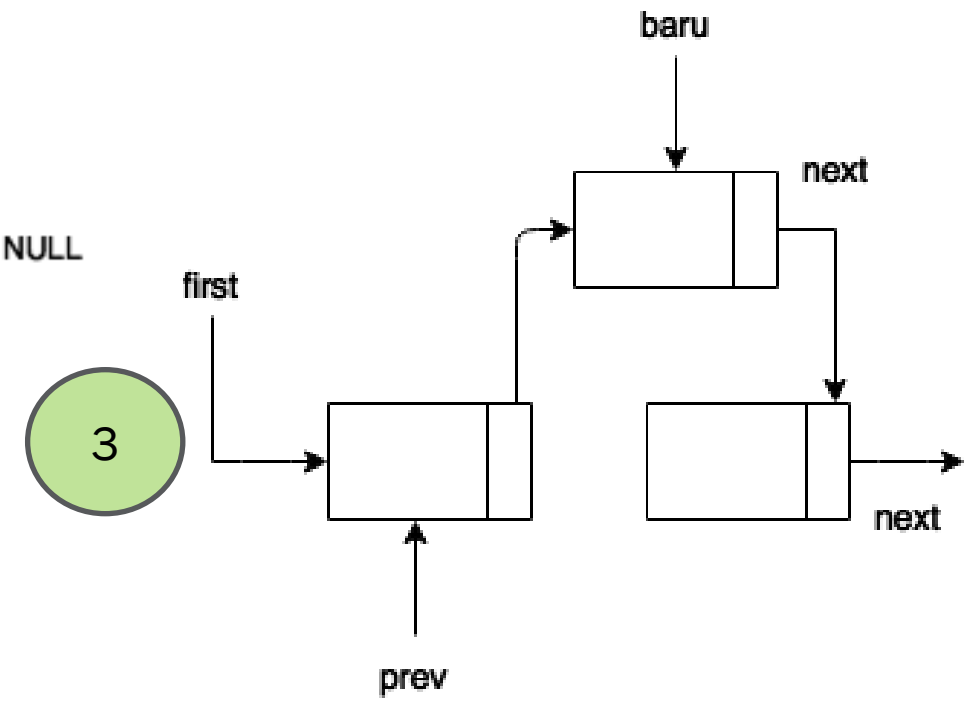
2

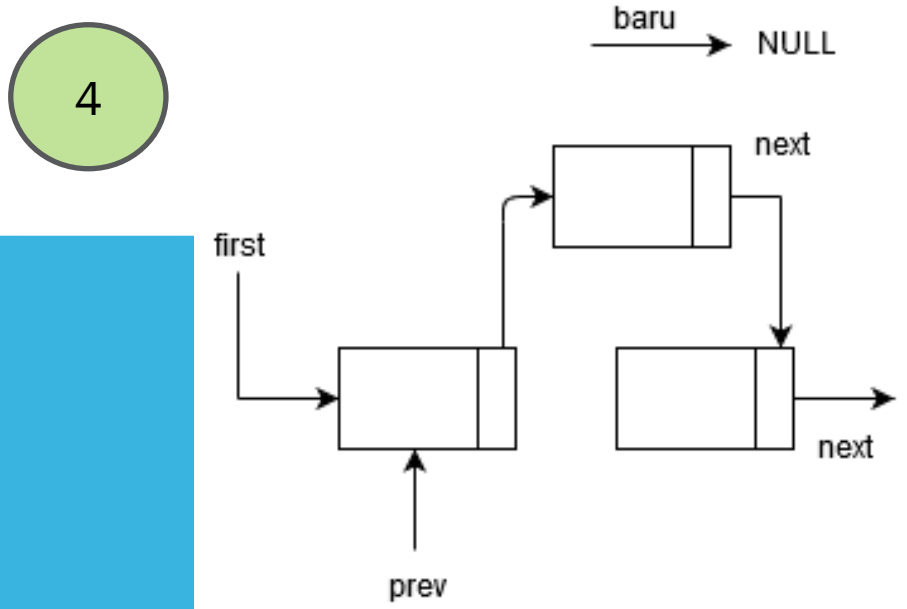


PENAMBAHAN ELEMEN DI TENGAH (ADDAFTER) JIKA ELEMEN SEBELUM DI TENGAH









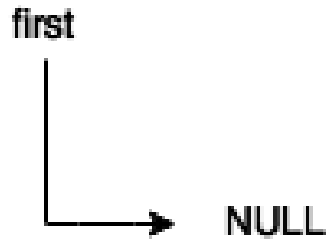
ADD AFTER

```
void addAfter(int prev, char
   nim[], char nama[], char
   nilai[], list *L){


  if(countElement(*L) < 10){

    int baru = emptyElement(*L);


strcpy((*L).data[baru].elmt.nim,
   nim);

strcpy((*L).data[baru].elmt.nama
   , nama);

strcpy((*L).data[baru].elmt.nila
   i, nilai);

  if((*L).data[prev].next == -
   1){

     (*L).data[baru].next = -1;

  }else{

     (*L).data[baru].next =
   (*L).data[prev].next;

  }
```

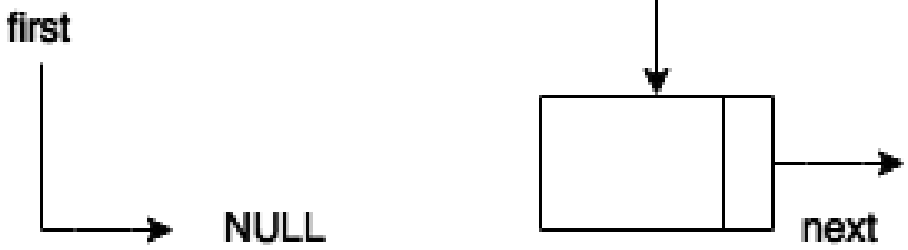
```
   (*L).data[prev].next =
   baru;

}

else{

    /*proses jika array
   penuh*/

    printf("sudah tidak dapat
   ditambah\n");

  }


}
```

PENAMBAHAN ELEMEN DI AKHIR (ADDLAST) JIKA LIST KOSONG MAKA ADDFIRST UNTUK LIST KOSONG

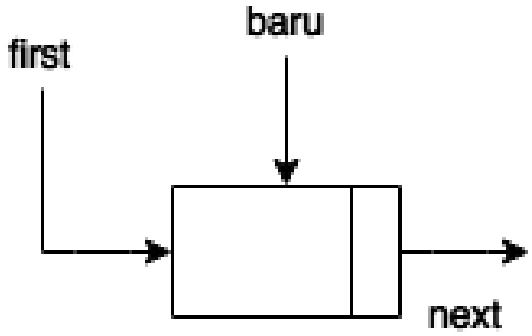
1



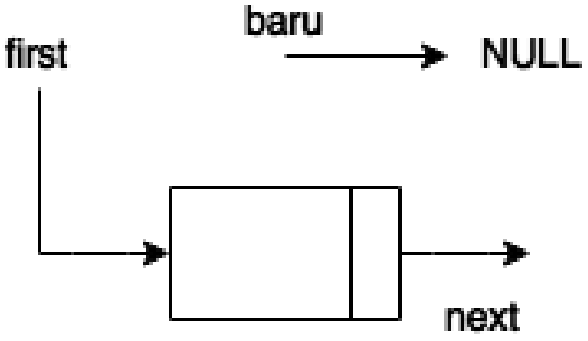
2



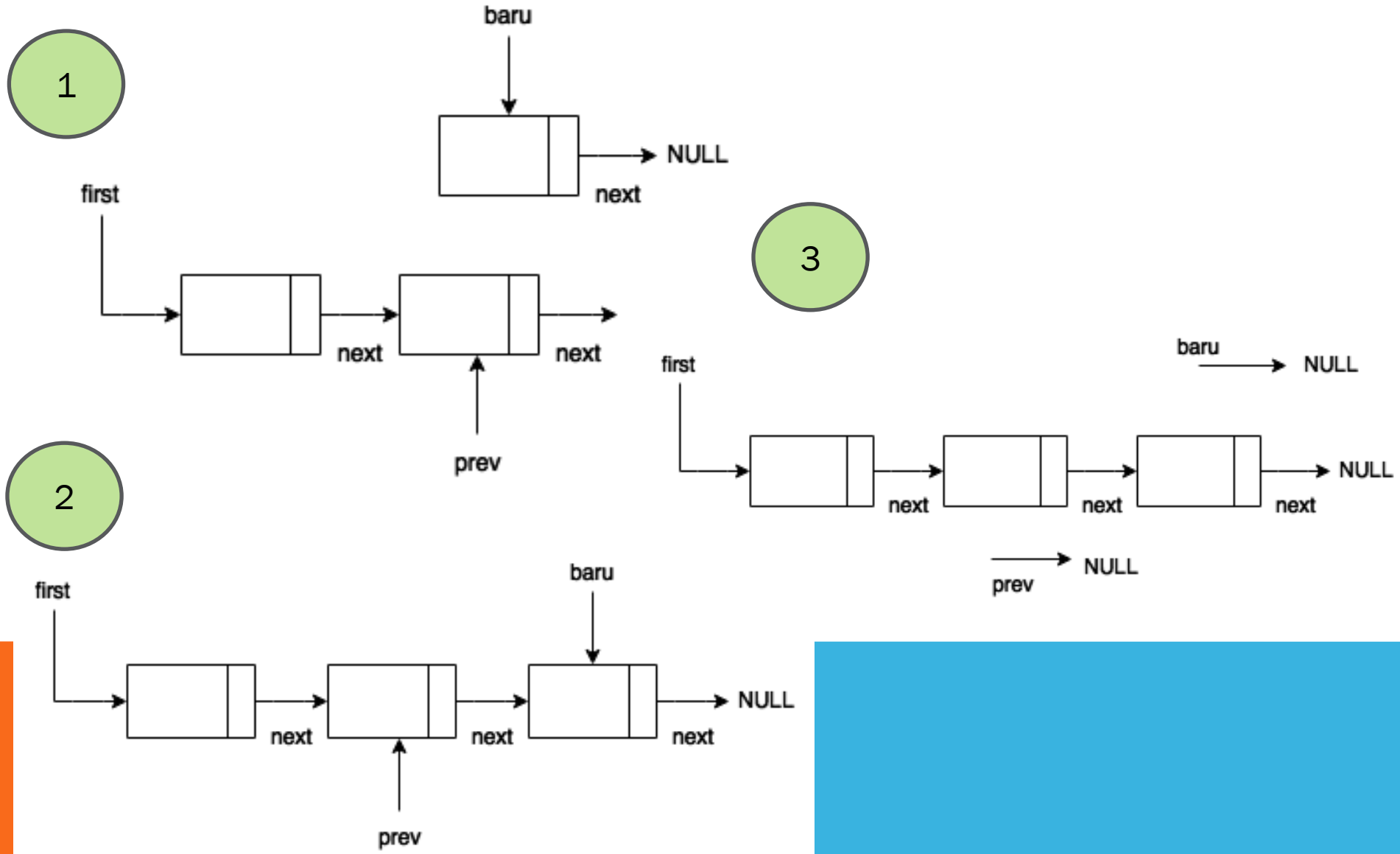
3



4



PENAMBAHAN ELEMEN DI AKHIR (ADDLAST) JIKA ELEMEN BANYAK ELEMEN MAKA ADDAFTER ELEMEN BELAKANG

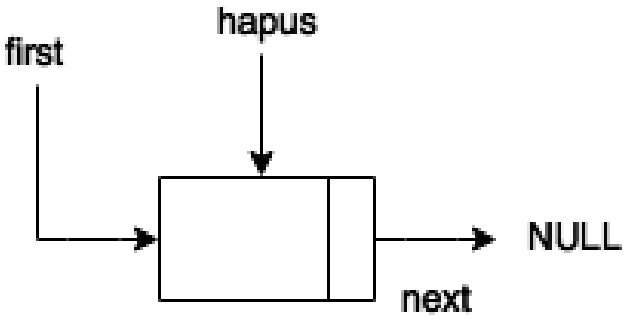


ADD LAST

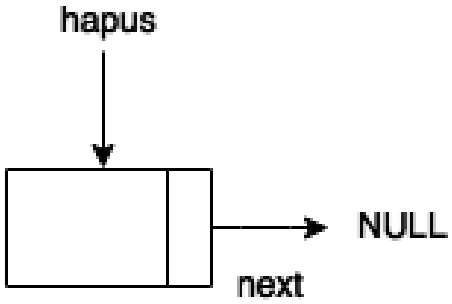
```
void addLast(char nim[], char nama[],
   char nilai[], list *L){
  if((*L).first == -1){
    /*proses jika list masih
   kosong*/
    addfirst(nim, nama, nilai, L);
  } else{
  /*proses jika list telah berisi
   elemen*/
    if(countElement(*L) < 10){
      /*proses jika array belum
   penuh*/
  /*proses mencari elemen terakhir*/
      /*inisialisasi*/
      int prev = (*L).first;

      while((*L).data[prev].next != -
   1){
        /*iterasi*/
        prev = (*L).data[prev].next;
      }
      addafter(prev, nim, nama,
   nilai, L);
    }else{
      /*proses jika array penuh*/
      printf("sudah tidak dapat
   ditambah\n");
    }
  }
}
```

HAPUS ELEMEN AWAL (DELFIRST) JIKA SATU ELEMEN

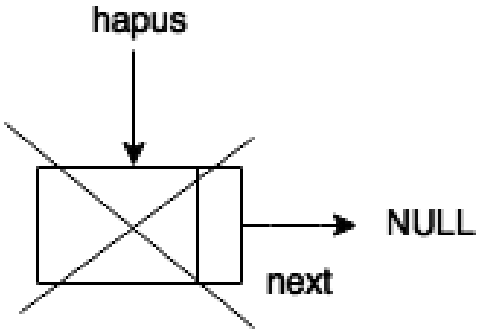
1

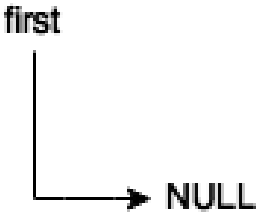


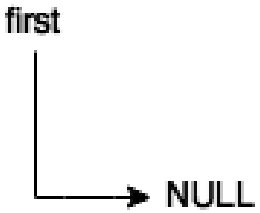
2



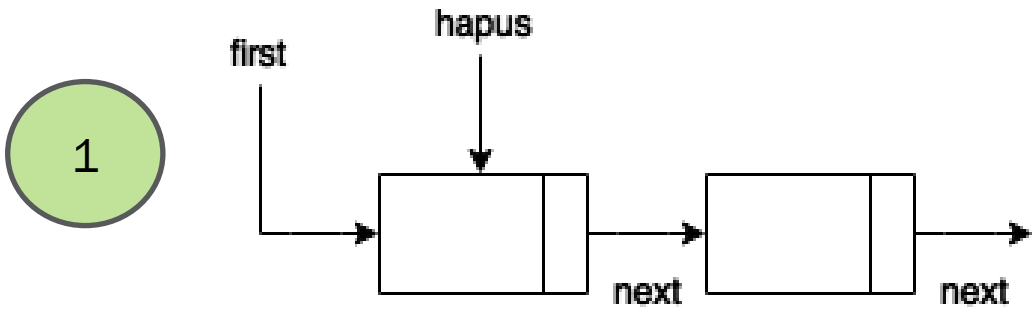
3

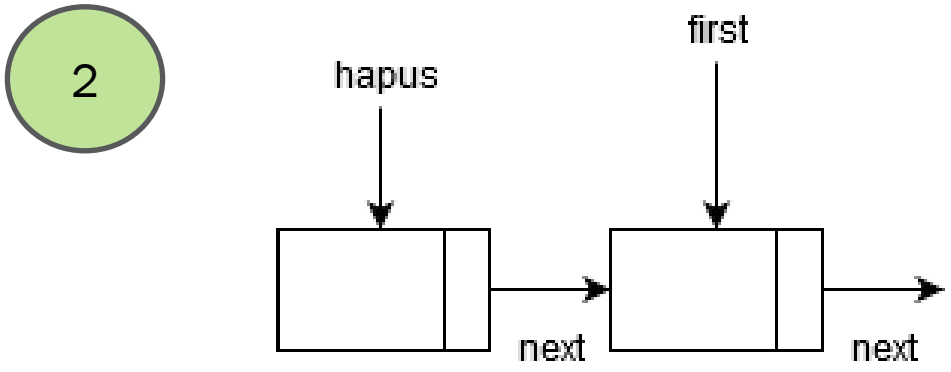


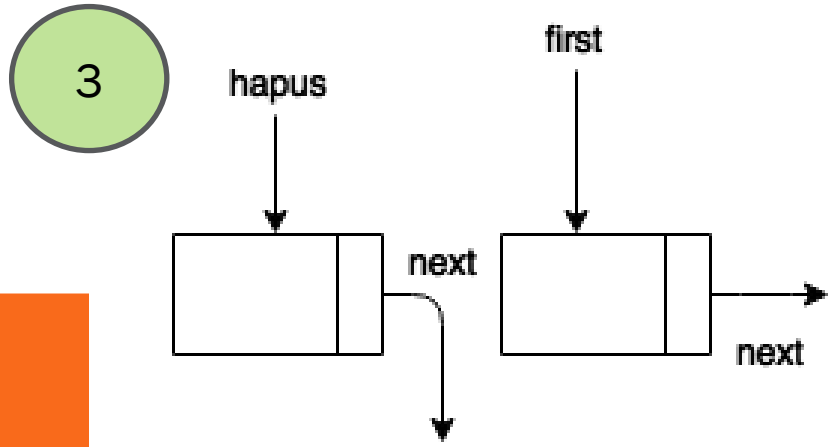


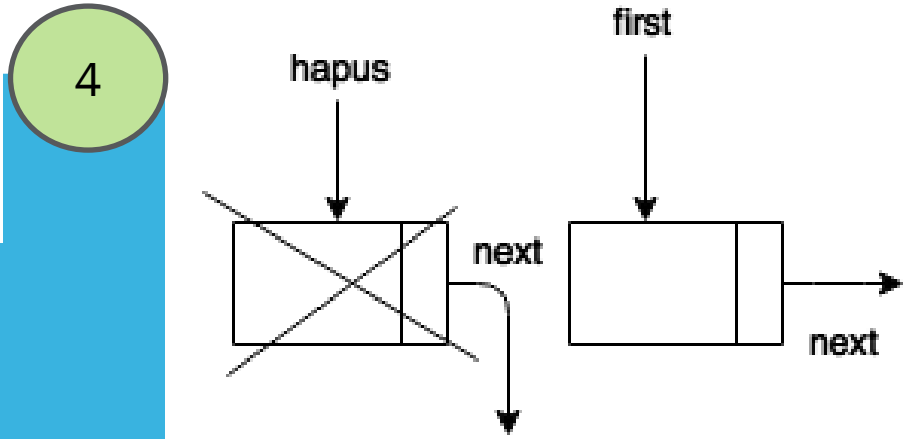


HAPUS ELEMEN AWAL (DELFIRST) JIKA BANYAK ELEMEN ELEMEN







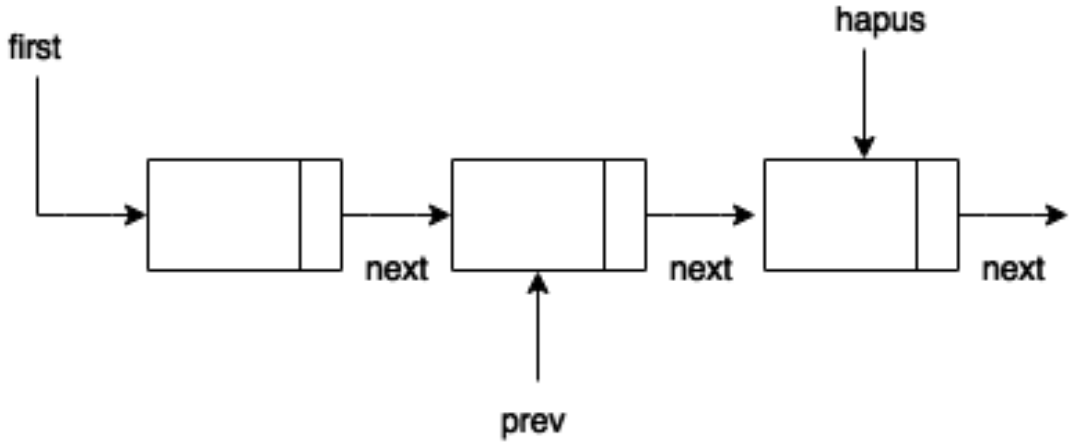


DEL FIRST

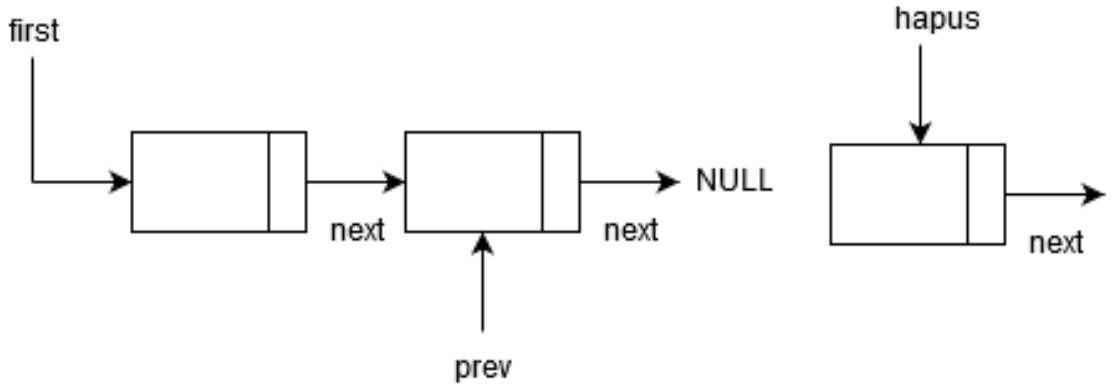
```
void delFirst(list *L){
  if((*L).first != -1){
    int hapus = (*L).first;
    if(countElement(*L) == 1){
       (*L).first = -1;
    }else{
       (*L).first = (*L).data[(*L).first].next;
    }
    /*elemen awal sebelumnya dikosongkan*/
    (*L).data[hapus].next = -2;
  }
  else{
   /*proses jika list kosong*/
    printf("list kosong\n");
  }
}
```

HAPUS ELEMEN TENGAH (DELAFTER) JIKA YANG DIHAPUS PALING BELAKANG

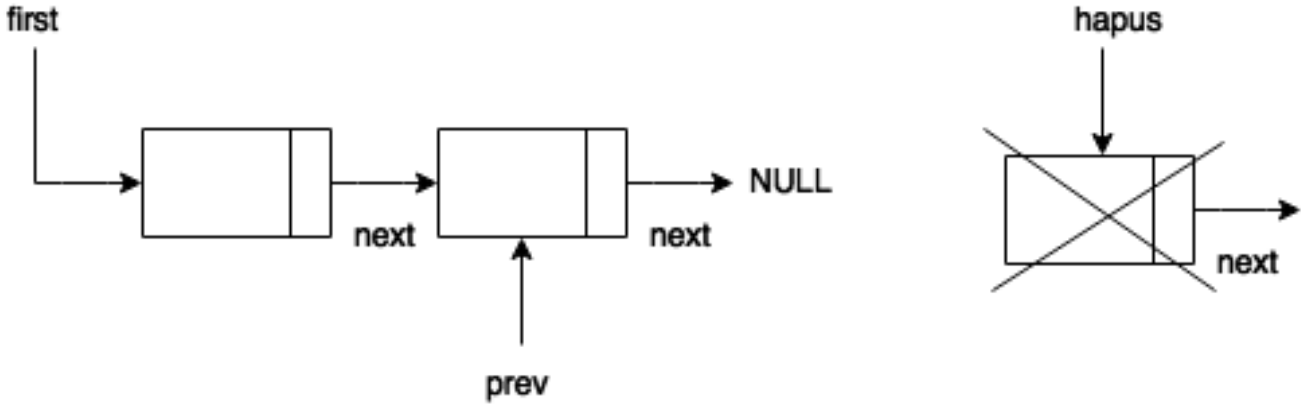
1



2

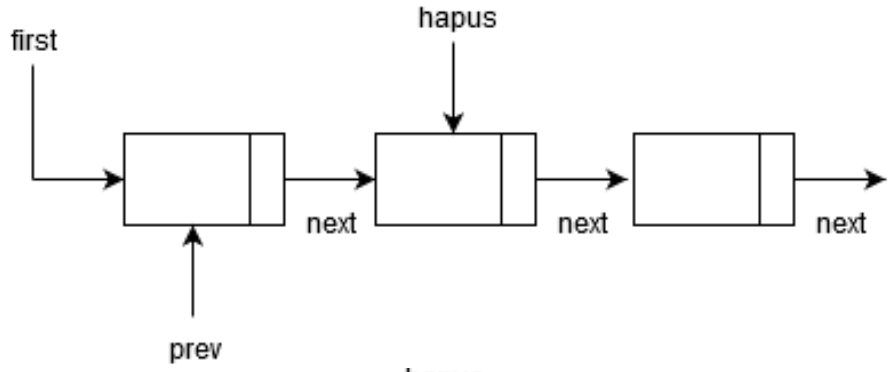


3

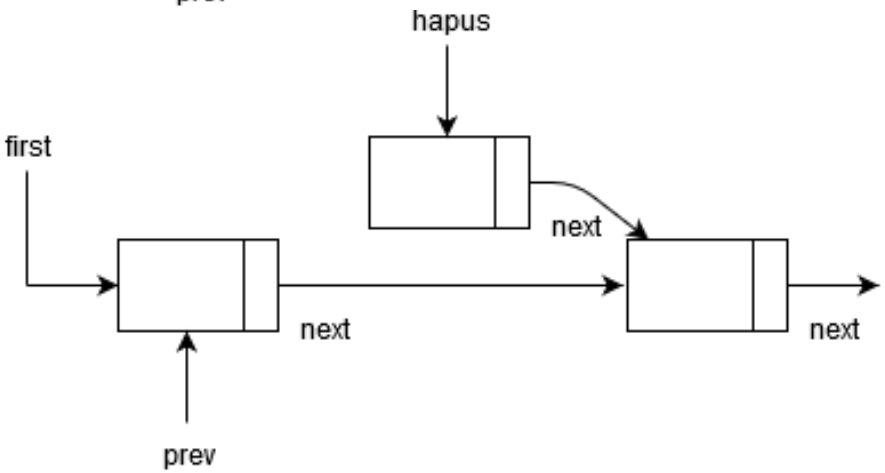


HAPUS ELEMEN TENGAH (DELAFTER) JIKA YANG DIHAPUS DI TENGAH

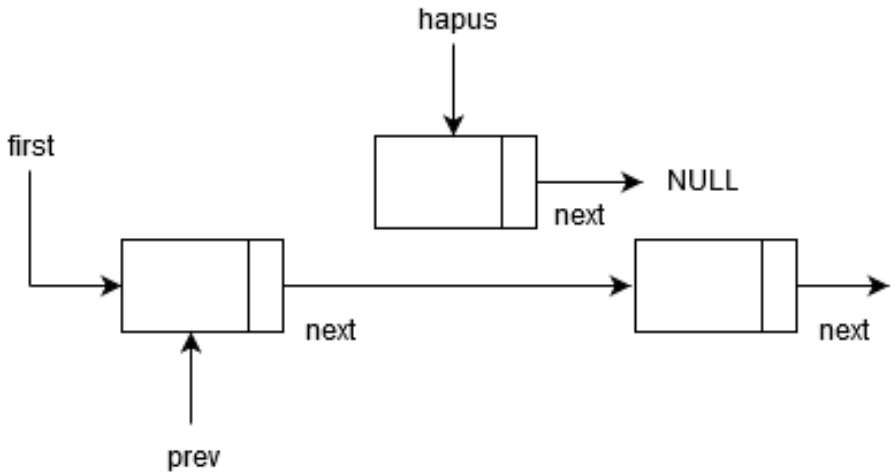
1



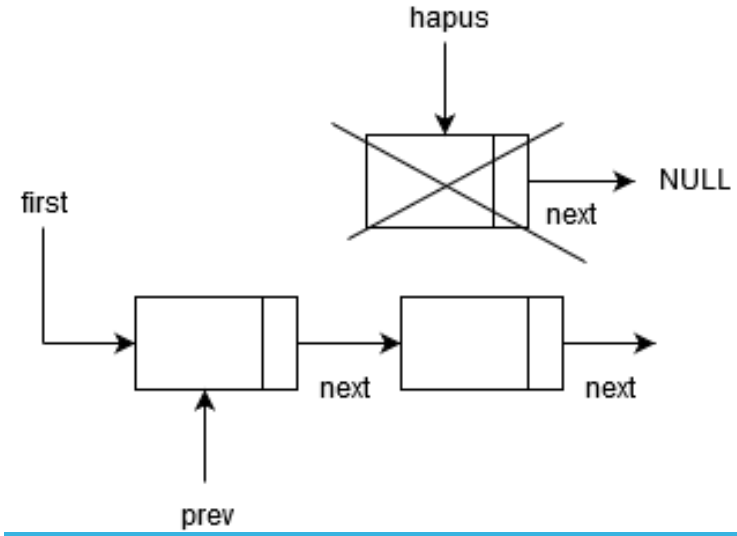
2



3



4

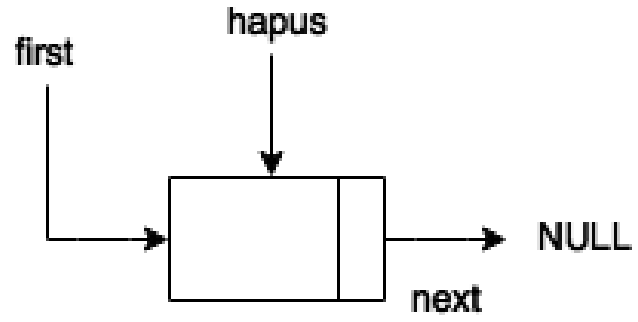


DEL AFTER

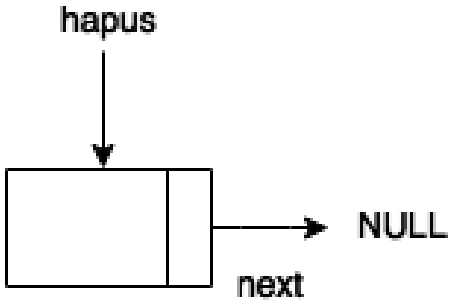
```
void delAfter(int prev, list *L){

  int hapus = (*L).data[prev].next;

  if(hapus != -1){
     if((*L).data[hapus].next == -1){
        (*L).data[prev].next = -1;
     }else{
        (*L).data[prev].next = (*L).data[hapus].next;
     }
     /*pengosongan elemen*/
     (*L).data[hapus].next = -2;
  }

}
```

HAPUS DI AKHIR (DELLAST) JIKA HANYA SATU ELEMEN MAKA DELFIRST SATU ELEMEN

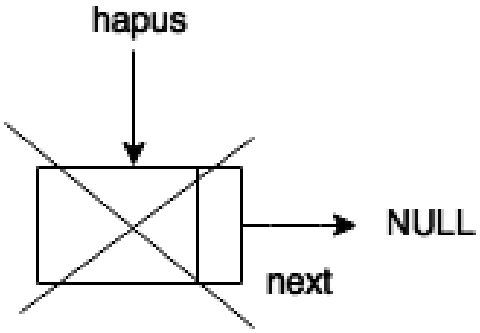
1

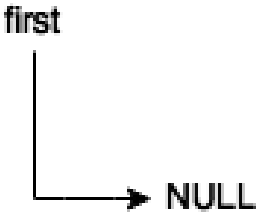


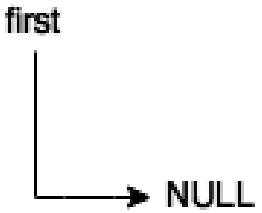
2



3

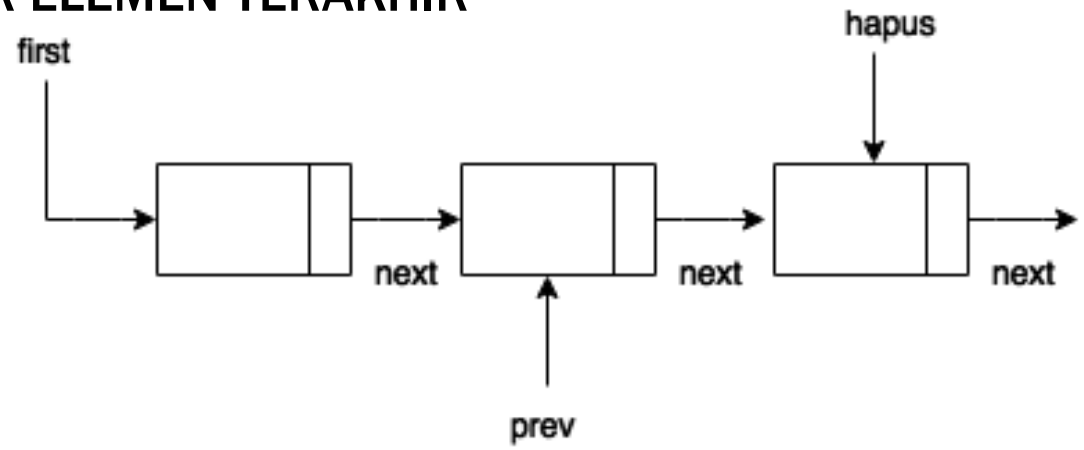




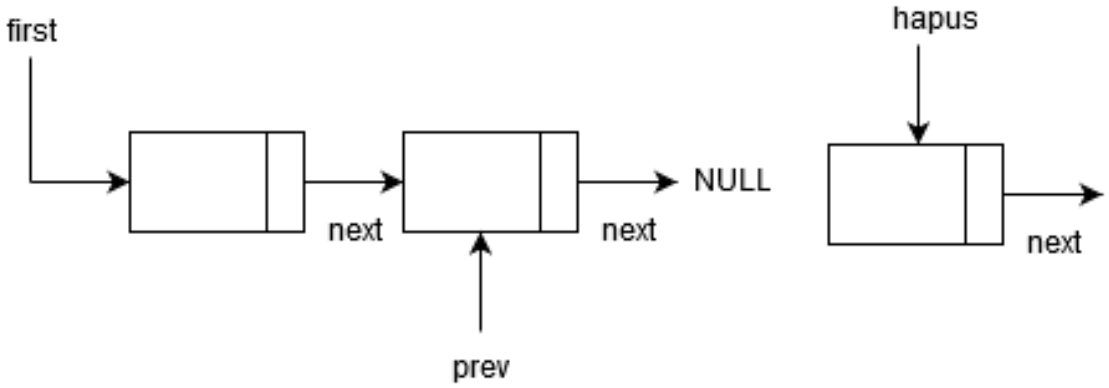


HAPUS DI AKHIR (DELLAST) JIKA BANYAK ELEMEN MAKA DELAFTER ELEMEN TERAKHIR

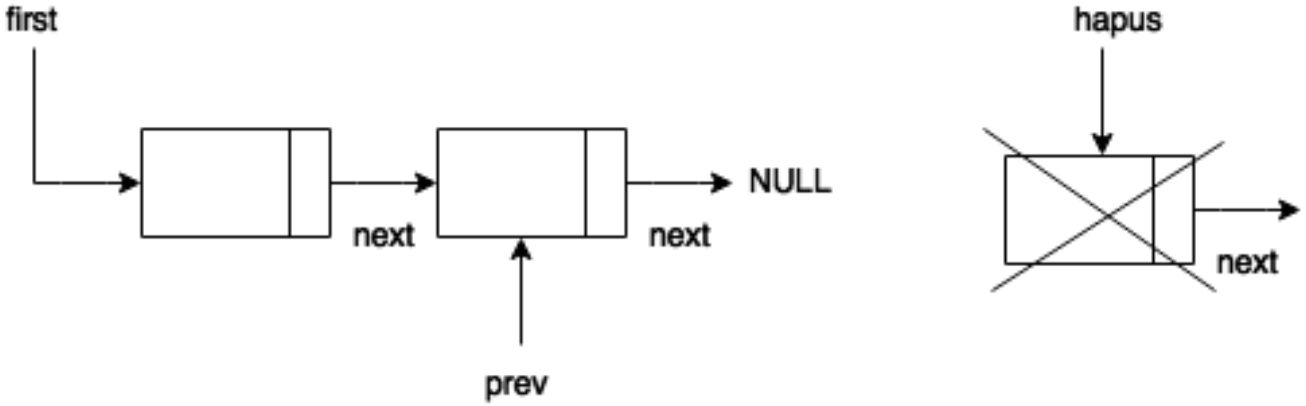
1



2



3



DEL LAST

```
void delLast(list *L){
 if((*L).first != -1){
   if(countElement(*L) == 1){
     /*proses jika list hanya
   berisi satu elemen*/
     delFirst(L);
   }
   else{
     int hapus = (*L).first;
     int prev;
     while((*L).data[hapus].next
   != -1){

       /*iterasi*/
       prev = hapus;
       hapus =
   (*L).data[hapus].next;
     }
   /*elemen sebelum elemen
      terakhir menjadi elemen
      terakhir*/
      delafter(prev, L);
   }
    else{
     /*proses jika list kosong*/
     printf("list kosong\n");
    }
   }
```

PRINT ELEMENT

```
void printElement(list L){
  if(L.first != -1){
    /*inisialisasi*/
    int elmt = L.first;
    int i = 1;
    while(elmt != -1){
      /*proses*/
      printf("elemen ke : %d\n",
   i);
      printf("nim : %s\n",
   L.data[elmt].elmt.nim);
      printf("nama : %s\n",
   L.data[elmt].elmt.nama);
      printf("nilai : %s\n",
   L.data[elmt].elmt.nilai);
      printf("next : %d\n",
   L.data[elmt].next);
      printf("------------
   \n");

      /*iterasi*/
    elmt = L.data[elmt].next;
      i = i + 1;
    }
  }
  else{
   /*proses jika list
   kosong*/
    printf("list kosong\n");
  }
}
```

DEL ALL

```
void delAll(list *L){
  int i;
  for(i=countElement(*L);i>=1;i--){
  /*proses menghapus elemen list*/
    delLast(L);
  }
}
```

MAIN

```
int main(){
  list L;
  createList(&L);
  printElement(L);
  printf("================\n");

  addFirst("1", "Orang_1", "A",
   &L);
  addAfter(L.first, "2",
   "Orang_2", "A", &L);
  addLast("3", "Orang_3", "A",
   &L);
  printElement(L);

  printf("================\n");
  delLast(&L);
  delAfter(L.first, &L);
  delFirst(&L);
  printElement(L);

   printf("================\
   n");

  return 0;

}
```

COBAIN... UNTUK MENELISIK APAKAH BENAR PAHAM ATAU TIDAK

ada n masukan berupa float, masukkan ke dalam list secara addfirst dan tampilkan isi list

Format Masukan:
n, 0 < n <= 100
n baris float

Format Keluaran:
n baris float

Contoh Masukan
5
1.10
2.20
3.30
4.40
5.50

Contoh Keluaran
5.50
4.40
3.30
2.20
1.10

DAFTAR PUSTAKA

S, Rosa A. dan M. Shalahuddin. 2010. Modul Pembelajaran: Struktur Data. Modula: Bandung.