Procedure _ InsVfirst (L : List , X : info tree)

P bertipe address

P ← Alokasi (x)
if (P != Nil) then
P.next ← L↑.First
L↑.First ← P
endif
End _ Procedure

Visualization

Procedure - CreateList (L : List)
      L. First ← Nil
End Procedure

```
┌─[Visualization]─────────────┐
│                             │
│        □ First              │
│        │                    │
│        │                    │
│        ─┴─  Nil             │
│                             │
└─────────────────────────────┘
```
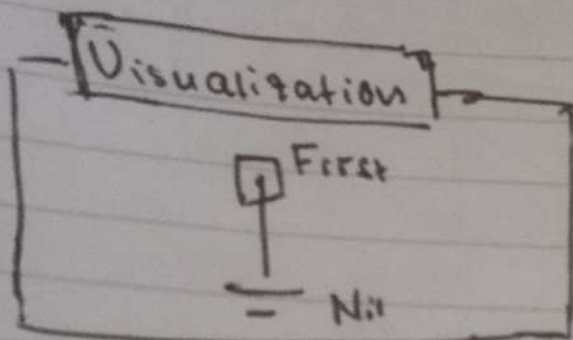
Function Alokasi (X : Infotype)
    P bertipe address

    P ← alloc (Elm_List, address)
    if (P != Nil) then
    | P.info ← X
    | P.next ← Nil
    endif
    return (P)
End Function
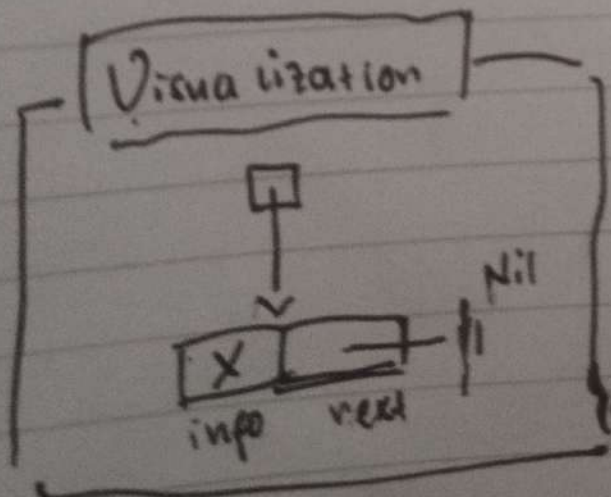
```
┌─[Visualization]──────────────┐
│                              │
│        □                     │
│        │                     │
│        ∨            Nil       │
│     ┌──────┬───┐    ││        │
│     │  X   │ ─┼───  ││        │
│     └──────┴───┘              │
│      info   next             │
└──────────────────────────────┘
```

Procedure_InsVFirst (L: List , x : info type)

P bertipe address

P ← Alokasi (x)
if (P != Nil) then
    P.next ← L↑.First
    L↑.First ← P
endif
End_Procedure
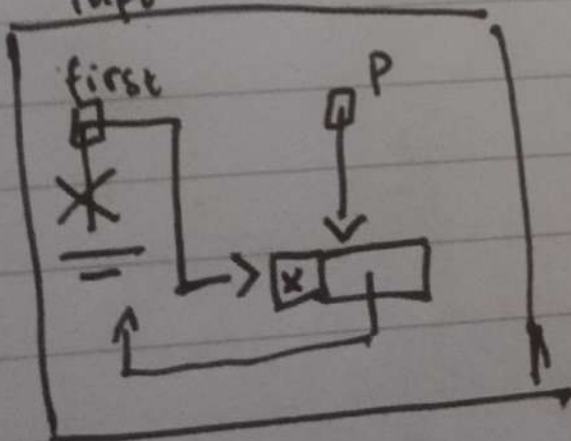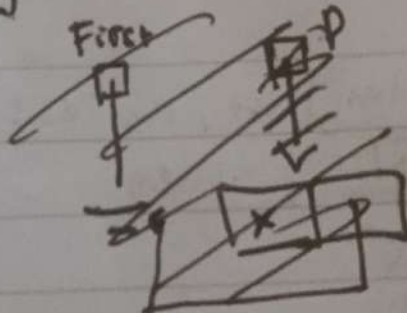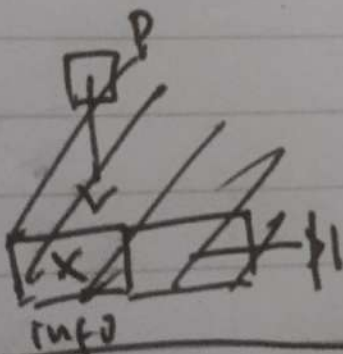
Visualization

```
Procedure_InsVLast (L↑: List, x: infotype)
    P, temp  bertipe address
    P ← Alokasi (x)
    if (P != Nil) then
        P.next ← Nil
        if (L↑.First = Nil) then
            L↑.First ← P
        else    then
            temp ← L↑.First
                while (temp.next != Nil) then
                    temp ← temp.next
                endwhile
            temp.next ← P
        endif
    endif
```

[Visualizaton]
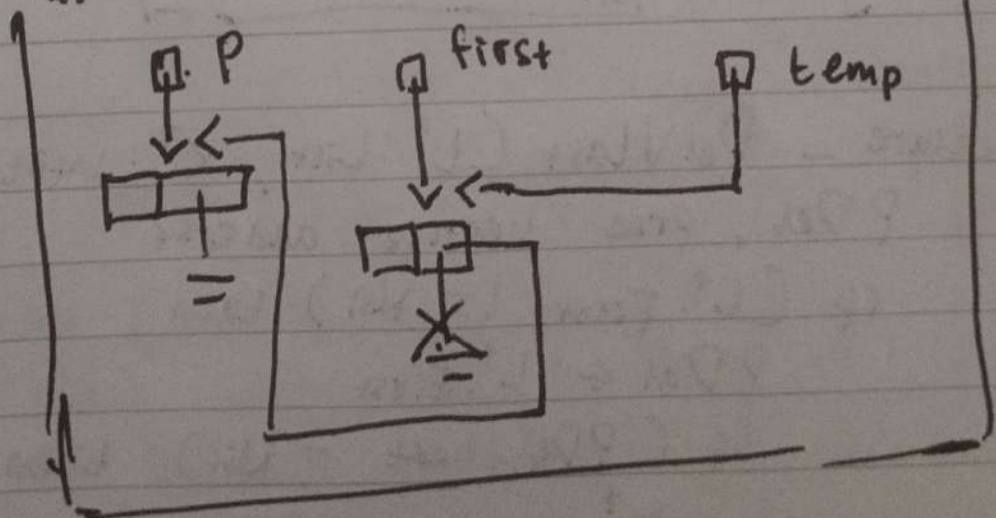
Procedure_DelVFirst $(L^\uparrow : List, x^\uparrow infotype)$
  P bertipe address
  if $(L^\uparrow.First \ != Nil)$ then
    $P \leftarrow L^\uparrow.First$
    $L^\uparrow.First \leftarrow P.next$
    $x^\uparrow \leftarrow P.info$
    $P.next \leftarrow Nil$
    DeAlokasi (P)
  end if
End_Procedure

Visualization

P
First
$x^\uparrow = 10$
10
Nil
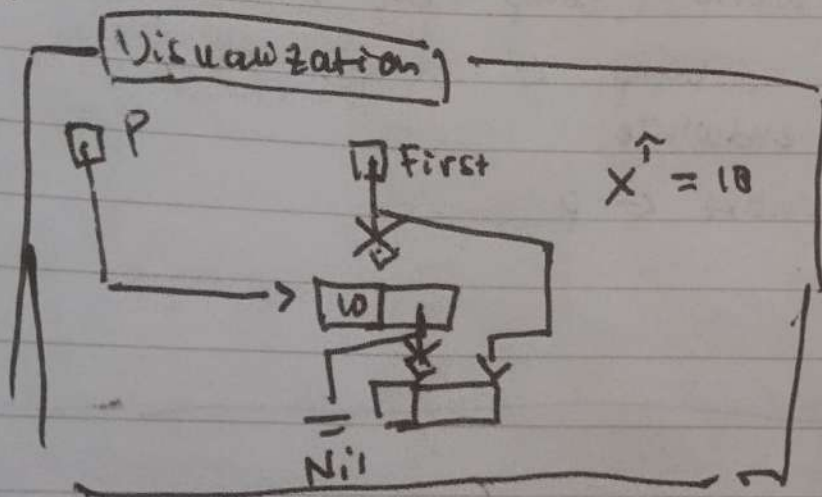
Procedure_DelVLast $(L^\uparrow : List, x^? : Infotype)$
  PDel, prec bertipe address
  if $(L^\uparrow.First \ != Nil)$ then
    $PDel \leftarrow L^\uparrow.First$
    if $(PDel.next = Nil)$ then
      $x^\uparrow \leftarrow PDel.info$
      $L^\uparrow.First \leftarrow Nil$
      De Alokasi (PDel)
    ~~endif~~

  else

```
        prec <- Nil
        while (PDel. next != Nil) and do
                prec <- PDel
                PDel <- next PDel. next
        end while
            Prec. next  <- Nil
            x↑ = PDel. info
        end if
        Dealokasi (PDel)
    end if
    end procedure
```
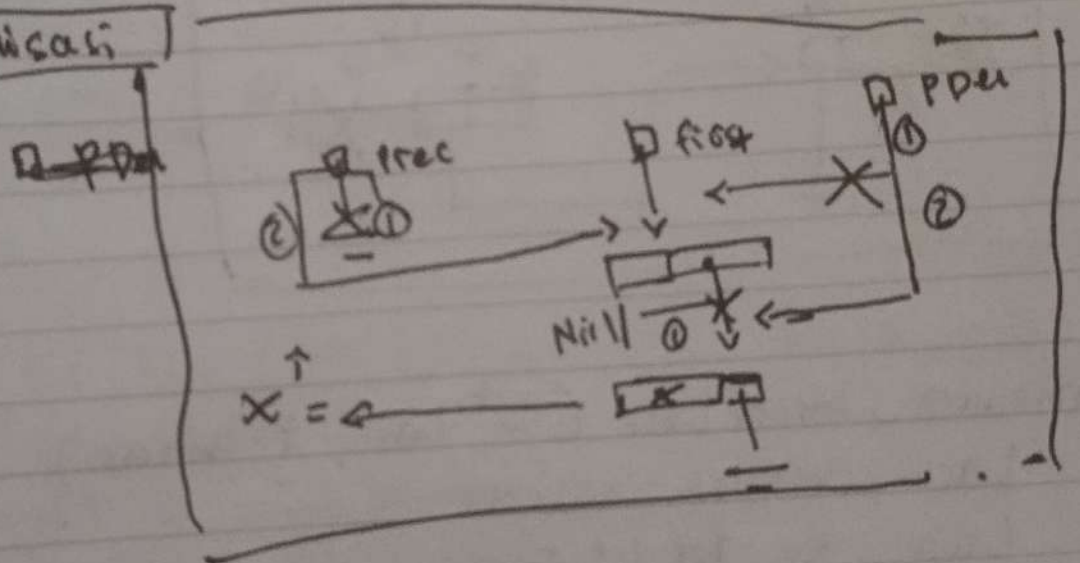

Visualisasi

```
procedure  Insert First (L↑: List, P: address)
        if (P != Nil) then
            P. next  <- L↑. First
            L↑. First <- P

        endif
    end procedure
        Visualisasi hampir sama dengan
        Del V First
```
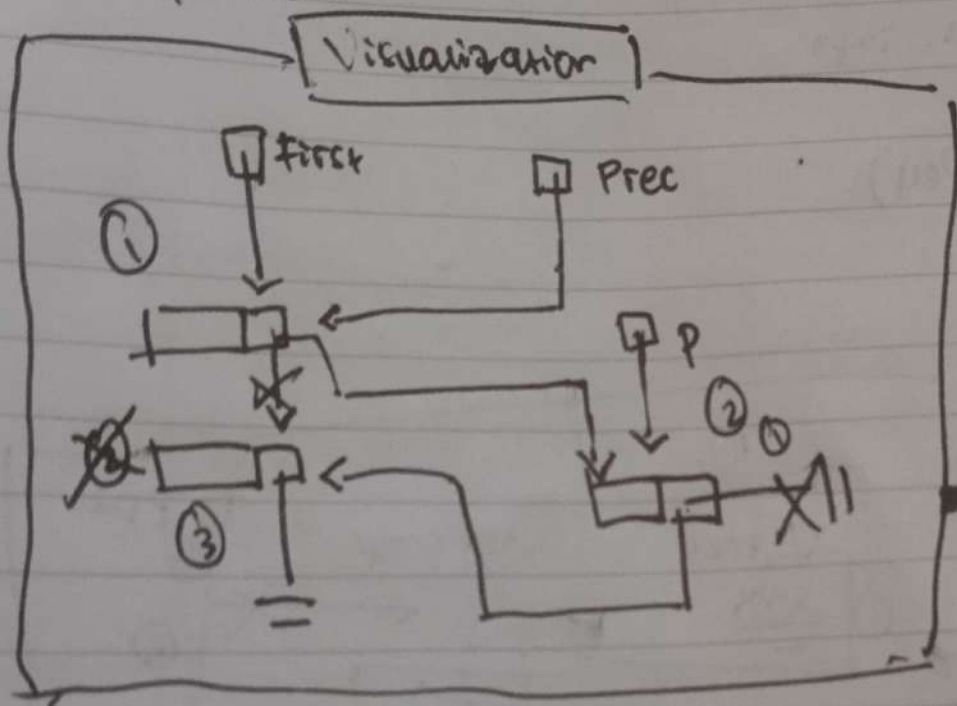
Procedure InsertAfter (L↑: List, P, Prec : address)
    if (P != Nil AND Prec != Nil) then
        P.next <- Prec.next
        Prec.next <- P
    endif
end procedure

Procedure _ InsertLast (L↑: List, P: address)
    - Last vertipe address
    Last <- L↑. first
    while (Last.next != Nil) do
        Last <- Last. next
    endwhile
    Last . next <- P
    P. next <- Nil

end procedure

```
Procedure  DelFirst ( L↑ : List, P↑ : address )
      L↑.First  ←  P↑.next
      P↑.next  ←  Nil
end  procedure


Procedure   Del Last ( L↑ : List, address · P↑ : Address )
      Prec  bertipe  address
      P↑  ←  L↑.first
      while ( P↑.next  != Nil ) then do
          Prec ←  P↑
          P↑  ←  P↑.next
      end while
      Prec.next  ←  Nil
      P↑ ←  Prec

end procedure


Procedure   Del_After ( L↑ : List, PDel↑ : Address, Prec ad
      if ( Prec != Nil AND  Prec.next != Nil ) then
          PPel↑  ←  Prec.next.
          Prec.next ← PDel↑.next
          PDel↑.next  ←  Nil.
```

```
Procedure    Print Info (L : List)
        P bertipe address
        P ← L.first
        if (List Empty (L)) then
            Write ("List Kosong")
    else      then
            while   (P != Nil) do
                Write   layar (info (P) "→)
                P ← P.next
            end while
        write ("Nil") Layar
        end if
end procedure
```

```
Program Utama
  Kamus Data
    My List bertipe List
    isi  bertipe  info type
    P, Prec bertipe address

  BEGIN
  Create List (&My List)
  P ← Alokasi (9)
  Ins V First (& My List, 7)
  Prec = First My List. First
  Ins V Last (& My List, 11)
  ~~Print In~~
  Insert After (& My List, P, Prec)
  Ins V First (& My List, 5)
  Ins V Last (& My List, 13)
  Del V Last (& My List, & isi)
  Prec ← My List. first
  P ← My List. first. next.
  Del After (& My List, &P, Prec)
  Del V first (& My List, & isi)
  DelAll (& My List)
  END
```

Visualisasi Studi Kasus

① Create List

first

② Insert first (7)

first

sehingga   first   P

7
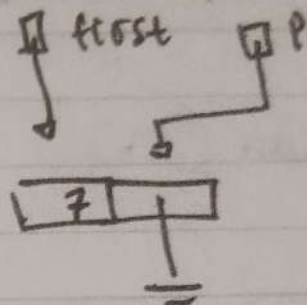
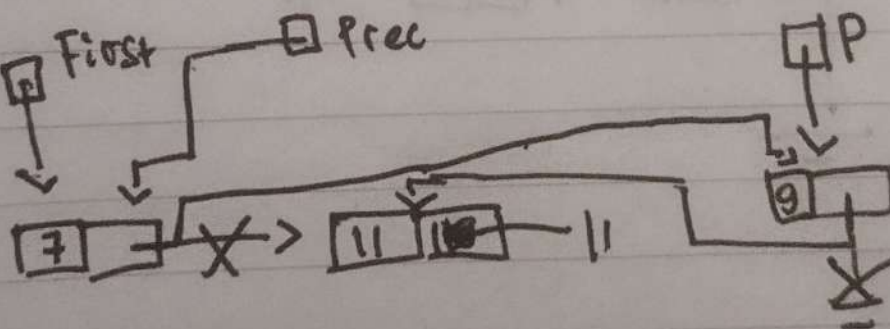③ Insert Last (11)

first   temp   P
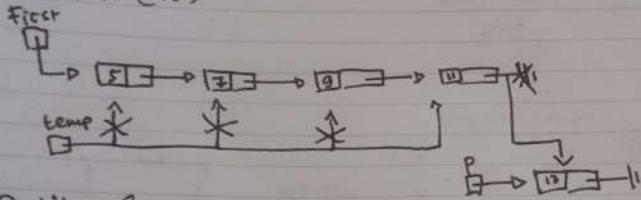
7   11
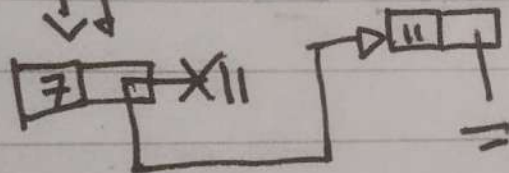
④ Insert After (9)

First   Prec   P

7   11   9

⑤ insert first (5)

First

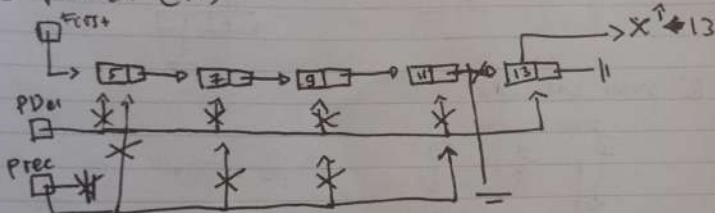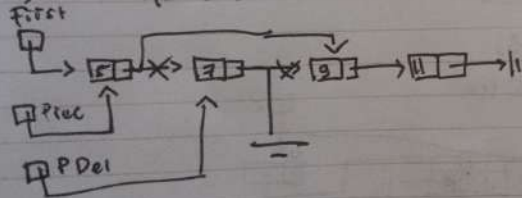7   5   11
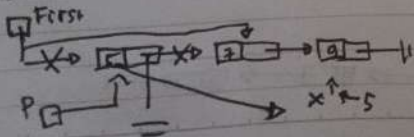
P   5
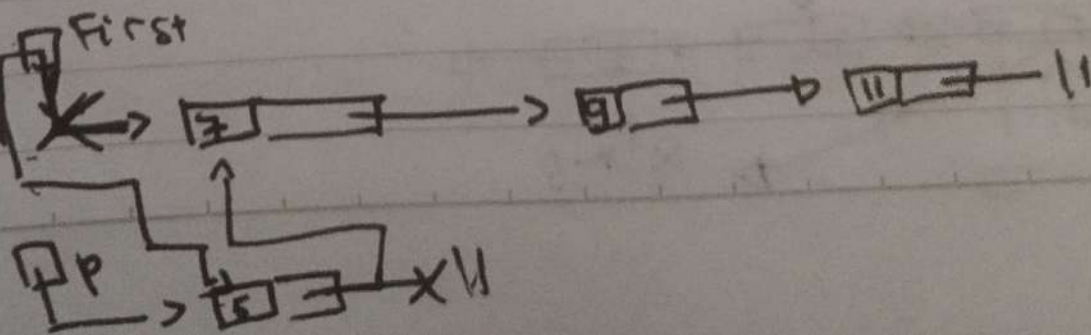
⑥ Ins V Last (13)

⑦ Del V last (12)

⑧ Delete After (7)

⑨ Del V First

⑩ Del All (7, 9)

Dealokasi