File Body

```
Declare Struct Node
    Data : Integer
    Next : Pointer to Node
End Struct

Function BuatNode (value : integer) return Pointer to Node
    Allocate memory for newNode
    Set newNode.Data = value
    Set newNode.Next = NULL
    Return newNode
End Function

Procedure InsertAwal (top : Reference to pointer to Node, va-
                      lue : Integer)
    newNode ← BuatNode (value)
    newNode.      ← top
    top ← newNode
End Procedure

Procedure InsertAkhir (top : reference to pointer to Node,
                       value : integer)
    newNode ← BuatNode (value)
    If top is NULL
        then top ← newNode
        Return
    End If

    temp ← head
```

No
Date

```
    While temp.Next is not NULL
        do  temp ← temp.Next
    End while

    temp.Next ← newNode
End Procedure

Procedure InsertBetween (top : Pointer to Node, target : Int-
                         eger, value : Integer)
    temp ← head
    while temp is not NULL and temp.Data is not target
        do temp ← temp.Next
    End while

    If temp is NULL
        then Print "value tidak ditemukan"
        return
    End if

    newNode ← BuatNode (value)
    newNode.      ← temp.Next
    temp.Next ← newNode
End Procedure

Procedure InsertbyValue (top : reference to pointer to Node,
                         value : Integer)
    newNode ← BuatNode (value)
    If top = NULL or top.Data >= value
        then newNode.Next ← top
             top ← newNode
```

No
Date

```
            return
        Endif

        temp ← top
        while temp.Next ≠ NULL and temp.Next.Data <
              value
            do temp ← temp.Next
        End while
        newNode.Next ← temp.Next
        temp.Next ← newNode
    End Procedure

Procedure DeleteAwal (top : Reference to Pointer to Node)
    If top = NULL
      then print " list kosong, tidak ada yg bisa dihapus".
           return
    End if

    temp ← top
    top ← top.Next
    Dealokasi temp
End Procedure

Procedure DeleteAkhir (top : reference to pointer to Node)
    If top = NULL
     then print " list kosong, tidak ada yg bisa dihapus"
          return
    End if

    If top.Next = NULL
```

```
        then  dealokasi top
              top ← NULL
              return
    End if

    temp ← head
    while temp.Next.Next ≠ NULL
        do  temp ← temp.Next
    End while

    dealokasi temp.Next
    temp.Next ← NULL
End Procedure

Procedure DeletebyValue (top : reference to pointer to Node,
                         value : Integer)
    If top = NULL
      then print " list kosong, tidak ada yg bisa
                   dihapus "
           return
    End if

    If top.Data = value
      then temp ← top
           top ← top.Next
           dealokasi temp
           return
    End If

    temp ← top
```

```
        while temp.Next ≠ NULL and temp.Next.Data ≠ Value
            do temp ← temp.Next
        End while

        If temp.Next = NULL
          then print " value tidak ditemukan "
               return
        End if

        toDelete ← temp.Next
        temp.Next ← temp.Next.Next
        dealokasi toDelete
    End Procedure

    Procedure DeleteBetween (top : reference to Pointer
                             to Node, target : Integer)
        temp ← top
        While temp ≠ NULL and temp.Data ≠ target
            do temp ← temp.Next
        End while

        If temp ≠ NULL or temp.Next = NULL
            then print " tidak ada elemen utk dihapus
                       selanjutnya", target
                 return
        End if
        toDelete ← temp.Next
        temp.Next ← temp.Next.Next
        dealokasi toDelete
    End Procedure
```

No

Date

Main Program

top as Pointer to Node = NULL

Insert Awal (top, 10)
Insert · (top, 5)
Printlist (top) // Output : 5 → 10 → NULL

Insert Akhir (top, 20)
Insert Akhir (top, 20)
Printlist (top) // Output : 5 → 10 → 20 → 20 → NULL

Insert Between (top, 10, 15)
Printlist (top) // Output : 5 → 10 → 15 → 20 → 30 → NULL

InsertbyValue (top, 25)
Insert by Value (top, 1)
Printlist (top) // Output : 1 → 5 → 10 → 15 → 20 → 25 → 30 → NULL

Delete Awal (top)
Print list (top) // Output : 5 → 10 → 15 → 20 → 25 → 30 → NULL

Delete Akhir (top)
Printlist (top) // Output : 5 → 10 → 15 → 20 → 25 → NULL

No

Date

```
DeletebyValue (top, 15)
Printlist (top) // output : 5 -> 10 -> 20 -> 25 -> NULL

Delete Between (top, 10)
Printlist (top) // output : 5 -> 10 -> 25 -> NULL

Procedure Printlist (top : Pointer to Node)
    temp ← top
    Print "linked list : "

    While temp ≠ NULL
        do print temp.Data, " -> "
            temp ← temp.Next
    End while
    Print "NULL"
End Procedure
```