a.) buatlah definisi Struktur ADT-nya

```
type akun <
    username, password : string
>
type infotype : akun
type adr      : pointer to elmlist
type elmlist <
    info : infotype
    next : adr
    prev : adr
>
type List <
    first : adr
    last : adr
>
Procedure createList (in/out L:List)
Function createNewElm (akun : infotype) -> adr
Procedure InsertLast (in/out L:List, in P : adr)
Function findAkun (L:List, username : string)
-> adr
Procedure signUp (in/out L:List, in akun:infotype)
```

CS Dipindai dengan CamScanner

```
Procedure deleteFirst(in/out L:List, out p:adr)
Procedure deleteAfter(in q:adr, out p:adr)
Procedure deleteLast(in/out L:List, out p:adr)
Procedure removeAkun(in username : string,
in/out L:List)
```

b.) buat implementasi fungsi /procedure berikut untuk melakukan regristasi / sigg up akun :

```
(i)Function createNewElm(akun::infotype) -> adr
Kamus
   p : adr
Algoritma
   allocate (p)
   p -> next = NIL
   p -> prev = NIL
   p -> info = akun
   return p
end function
(ii)Procedure InsertLast(in/out L:List; in p:adr)
Kamus

Algoritma
```

Dipindai dengan CamScanner

```
    if L.first ==NIL and L.Last ==NIL then
        L.first = p
        L.last = p
    else
        L.last->next = p
        p->prev = L.last
        L.last = p
    endif
endprocedure
(iii) Function findAkun(L:List, username:string)->adr
Kamus
    p:adr
Algoritma
    if L.first == NIL then
        return NIL
    else
        p = L.first
        while p != NIL do
            if p->info.username == username then
                return p
            endif
```

CS Dipindai dengan CamScanner

```
            p = P->next
        endwhile
        return NIL
    endif
endfunction
```

(iv) Procedure signUp(in/out L:List, in akun:infotype)

```
Kamus
    P:adr
    createNewElm(akun:infotype)-> adr
    InsertLast(in/out L:List, in p:adr)
    findAkun(L:List, username:string)-> adr
Algoritma
    P = findAkun(L, akun.username)
    if P != NIL then
        output("Account hasbeen registered.")
    else
        P = createNewElm(akun)
        Insert Last(L, P)
    endif
endprocedure
```

c.) buat implementasi prosedur berikut untuk

CS Dipindai dengan CamScanner

men-remove sebuah akun:

```
(i)Procedure deleteFirst(in/out L:List, out p:adr)
Kamus

Algoritma
    p = L.first
    L.first = p->next
    p->next = NIL
    L.first->prev = NIL
endprocedure
(ii)Procedure deleteAfter(in q:adr, out p:adr)
Kamus

Algoritma
    P = q-> next
    q -> next = p -> next
    P -> next -> prev = q
    P -> next = NIL
    P -> prev = NIL
endprocedure
```

CS Dipindai dengan CamScanner

```
(iii) Procedure deleteLost(in/out L:List, out p:adr)
Kamos

Algoritma
    p = L.last
    L.last = p -> prev
    L.last -> next = NIL
    p -> prev = NIL
endprocedure
(iv) Procedure removeAkun(in username:string, in/out
L:List)
Kamos
    p : adr
    findAkun(L:List, username : string) -> adr
    deleteFirst(in/out L:List, out p : adr)
    deleteAfter(in q:adr, out p:adr)
    deleteLast(in/out L:List, out p: adr)
Algoritma
  if L.first == NIL then
     output("List is empty")
  else
```

Dipindai dengan CamScanner

```
    p = find Akun (L, username)
    if p == NIL then
      output("Account has not found.")
    else
      if p == L.first then
        deleteFirst (L, p)
      else if p == L.last then
        deleteLast (L, p)
      else
        deleteAfter(p->prev, p)
      endif
      deallocate (p)
    endif
  endif
endprocedure
```