Made Naradeon Handika Promesta / 103032300101
Scal 1
a. type akun <
Uservenne : string
Password String
type infotype : akun
type address : pointer to elimlist
type address : pointer to elimlist type elimlist <
info: infotype
hext : oddress
preov: address
type List < first: address
last : oddress
> Akan Akan I
procedure create List (What All L: List).
Function new Elm (akun: info type) -> ordin address
Procedure insertlast (inat l: List, in p: address) Function find Akun (l: List, usorname: string) -> address
tunction find Akun (L: List, usorname: string) -> address
Porcelus souls live + 1:154 in Okun : infotume)
Procedure delete First ("Mout (: List, out p: Oddress")
Procedure delete Apter (1m q: address, out p: address)
Procedure delete Lost (i) out & List, out p. address)
Procedure delete First ("Yout L: List, out p: oddress) Procedure delete Apter (in q: oddress, out p: address) Procedure delete Lost (in/out L: List, out p: address) Procedure remove Akun (in username: String, in/out L: List)
). (i) Function new Elm (akun: infotype) -> address
Kamus
p: oddress
Algoritum
allowte (p)
p-> next = NIL
p-> prev = NIL
p-7 info = okun
return p
end function '
(SÍDU)

robe No	rudeen Handitu Promesta / 1030 32300 101
(ii) 1/	rocedure insert last (Wort L: Lost, inp. address)
k	amus
A	lgor: tma
	of L-first == NIL AND L. Lost == NIL then
	L.E.ret = D
	l. f;rst = p l. last = p
	0 00
	l. lost => next =p p-> prev = l. lost Mu l. lost = p
	9-7 prev = l.loxt
	Mar L. lost = P
	endif
en	A P COO CCHOO
(iii) F	unction Rind Akun (List Username: String) -> address
	dprogram unction find Akun (l: List, Username: String) -> address kamus
	p: address
A	lgoritma
	if lifirst == NIL then
	return NIL
	9ke
	p = l.first
	while of = NIL do
	if p-> info.username == Username then
	return p
	and from end of
	A ANGO AROGAL
	p=p->next
	enduhile
	return NIL
	endif
0	nd function

Made Naradean Hondita Pramesto / 102022300101
(iv) Parent simple (iv) + + 1 - 1 - 1 - 1 - 1 - 1 - 1
(iv) Procedure signup (in/out & l: list, in akun: infotype) kamus
p: address
newElm (akun : infotype) -> oddress
insert Last ("Mut lilist in proddress)
insert Last (infut likt, in proddress) find Akun (likt, username: String) - s address
Algoritma P= find Akun (L, akun. username) if p ≠ NIL then
output ("Account has been registered")
else
p = mently (akun)
insert last (l,p)
entif
Endprogram (in ())
C.(i) Procedure delete First (in/out 1: List, out p: address)
Kamush
Algoritima a
P = l.first
l.f.rsf = p-7 next
D->next = Nil
1. first > prev = VIL
end procedure
Kamus
Δ1
Algoritma $p = 0 \rightarrow \text{next}$
$q \rightarrow \text{next} = p \rightarrow \text{next}$
$q \rightarrow rext = p \rightarrow next$ $p \rightarrow next \rightarrow prev = q$
p->next = NIL
p-prext = NIL p-prev = NIL endprocedure
end procedure.

(iii) Procedure delete last ("Yout l: List, out p: a	ddress)
kamus	
Algoritum	× 94.15.
P = 1 lost	
1. Lost = p-> prev	
P = l. last l. last = p-> prev l. last > next = NIL	
p->prev = IVIL	
lubprocedure	
(iv) Procedure remove Akun (in username: string, injut,	l: List)
Kamus	
p: address	egalais.
find Akun (1: Last, username: String) -> address	
deleterist (in out 1: List, out p: address)	
delete First (in out 1: List, out p: address) delete Aptor (in q: address, out p: address) delete Last (in out 1: List, out p: address)	
All L.	
Algoritma if Lifirst == NIL then	
output ("list is empty")	
p = find Akun (1, Username) if p == NIL then	
if p== NIL then	
output (Account has not found)	
else '	
if p == l. first then	
else if $p = 1$ last then	
delete lost (L,p)	``
else	
deleteAfter (p->prev, p)	
onl rt-	
deallocote (P)	
edif	
endif	
end procedure	