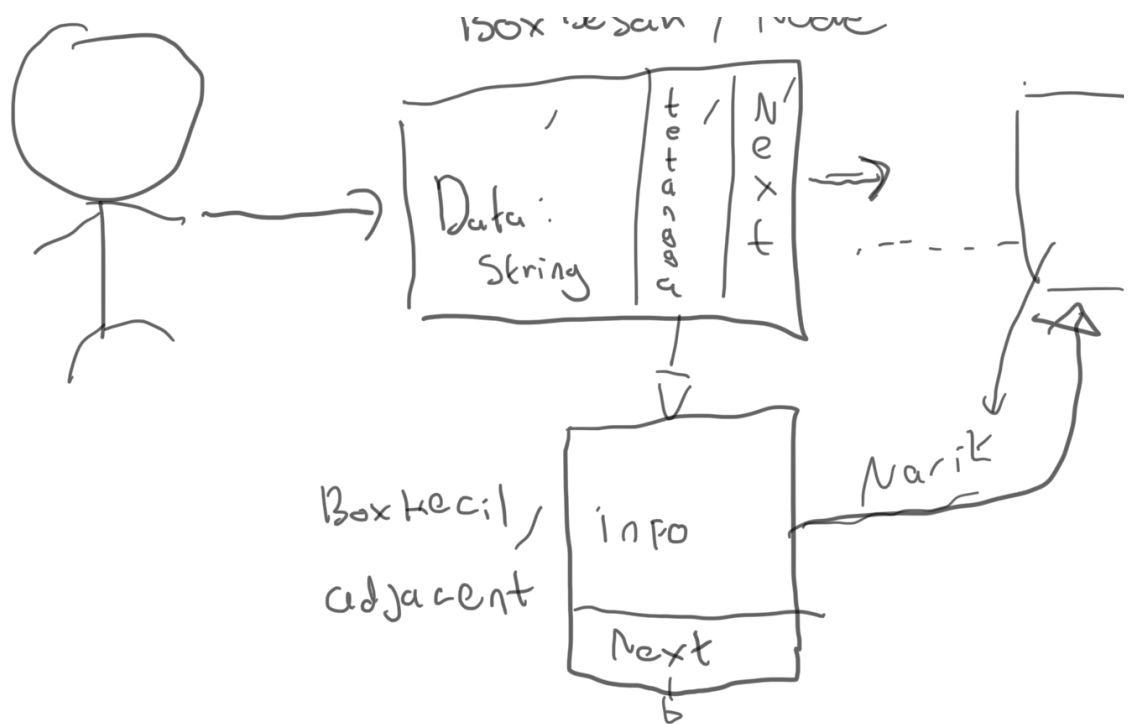


Declaras: Data

in mc ... / Neta



type Box besar = < Data: String
 Next: pointer to box besar
 tetangga: pointer to box kecil

type Box kecil = < info: pointer to box besar
 First: pointer to box besar Next: pointer to box kecil

type address besar = pointer to box besar

type address kecil = pointer to box kecil

type Box besar = < Data: String
 Next: address besar
 tetangga: address kecil

type Box kecil = < info: address besar
 Next: address kecil

First: address besar

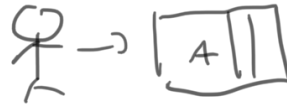
Insert Graph

- ↳ Insert Node
- ↳ Insert tetangga

Insert Node → Buat Node

A B C

Insert Node (A)



Insert Node (B)



(B)

Insert Node C



(C)

(B)

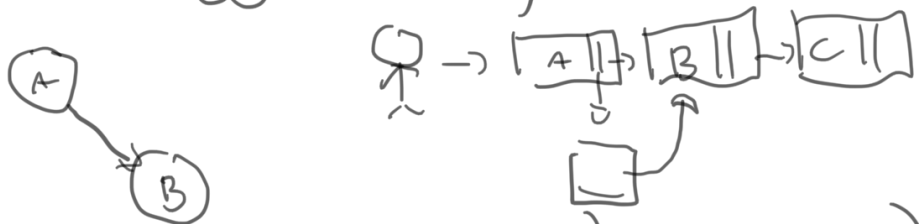
Insert tetangga → Bisa dilakukan
kalau node sudah
ada.

Ada
↳ menghubungkan Node
Satu dengan Node
lainnya

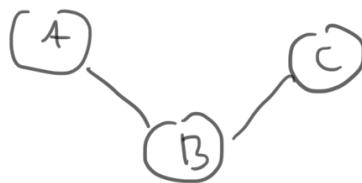
Insert Node A B C



insert tetangga (A, B) (B, A)



insert tetangga (B, C), (C, B)

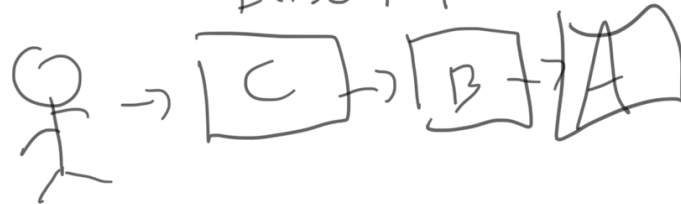
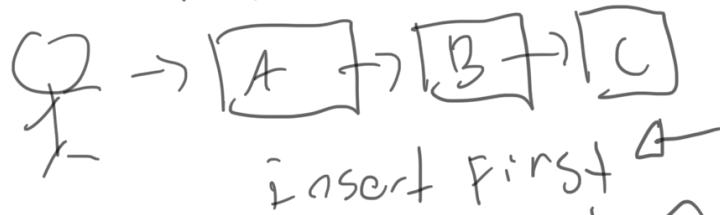


Insert Node ↗ Insert First
↘ Insert Last



A B C

Insert Last



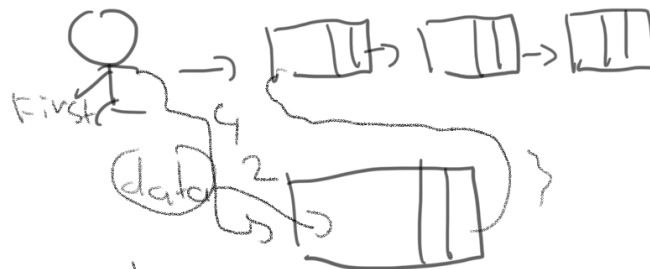
insert Node (Data: string , first : Address)

kamus

kotak_baru: Address baru

Algoritma

1) Alokasi (kotak_baru)



2) kotak_baru.data ← data ↑ 1 Buat box

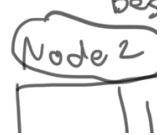
3) kotak_baru.next ← First

4) First ← kotak_baru
dihubungkan

Insert tetangga (Node1, Node2 : address baru , first : add
baru

kamus

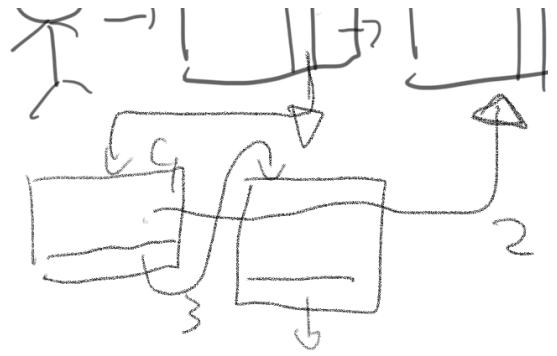
address kecil ()



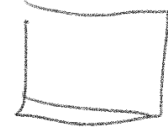
kotak-baru. info ← ...

Algoritma

1) Abkasi: (kotak-baru) →



2) kotak-baru.info ← Node 2



3) kotak-baru.next ← Node 1. tetangga

4) Node 1. tetangga ← kotak-baru

Function Search Node (data: string, first: Address)

kamus

Bantuan: address besar

Algoritma

1.) Bantuan ← first

2.) While (bantuan ≠ Null)

IF (data = bantuan.data)

return bantuan

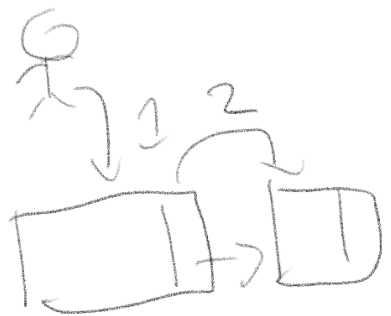
break

else

bantuan ← bantuan.next

Endwhile

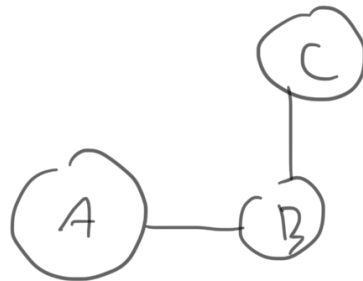
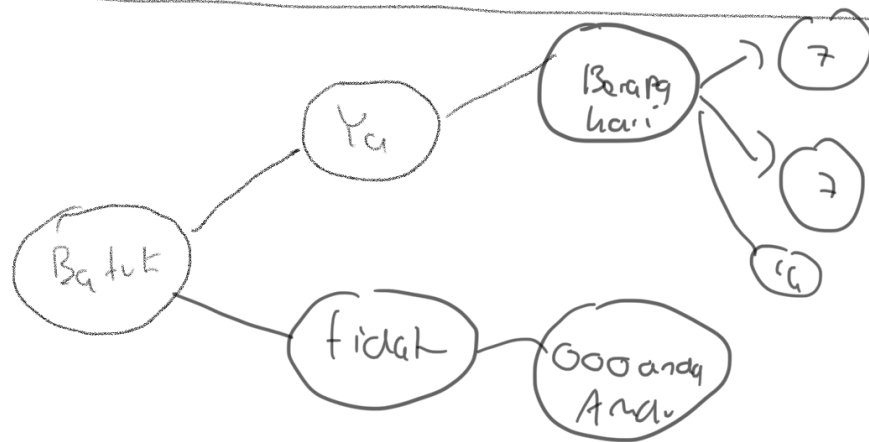
return Null



insert tetangga Node A dan B

...

insert tetangga (Search Node ("A"), search node
, first)



Graf

	A	B	C
A	False	true	False
B	true	False	true
C	False	true	False