program CollegeStudentWithDoubleLinkedList

{I. S : Program memasukkan data mahasiswa}

{F. S : Menghasilkan data mahasiswa}

{ Kamus Global }

uses crt

type

Ptr = ^Data

CollegeStudent = record

id, name : string

gpa : real

end

Data = record

info : CollegeStudent

pPrev, pNext : Ptr

end

var

pHead, pTail, pHead2, pTail2 : Ptr

select : integer

n : integer // total data saat ini

toSearch : integer // search choices

procedure create(var pHead, pTail : Ptr)

{I.S. : pHead dan pTail belum terdefinisi}

{F.S. : Menciptakan nilai nil pada pHead dan pTail}

{Kamus Tidak Ada}

{Algoritma}

begin

pHead <- nil

pTail <- nil

end // EndProcedure create()

procedure insertAtFront(var pHead, pTail : Ptr)

{I.S. : }

{F.S. : }

{Kamus Lokal}

var

pNew : Ptr

i : integer

{Algoritma}

begin

new(pNew)

pNew^.pPrev <- nil

i <- 1

gotoxy(15,9) writeln('------------------------------------------------------------------------------------')

gotoxy(15,10) writeln('| Input Student Data |')

gotoxy(15,11) writeln('------------------------------------------------------------------------------------')

gotoxy(15,12) writeln('| No | Student ID | Name | GPA |')

gotoxy(15,13) writeln('------------------------------------------------------------------------------------')

gotoxy(15,14) writeln('| | | | |')

gotoxy(15,15) writeln('------------------------------------------------------------------------------------')

gotoxy(17,14) write(i)

gotoxy(24,14) readln(pNew^.info.id)

gotoxy(53,14) readln(pNew^.info.name)

gotoxy(92,14) readln(pNew^.info.gpa)

if (pHead = nil) then

begin

pNew^.pNext <- nil

pTail <- pNew

end

else

begin

pNew^.pNext <- pHead

pHead^.pPrev <- pNew

end

pHead <- pNew

inc(n)

end // EndProcedure insertAtFront

procedure insertAtMiddle(var pHead, pTail : Ptr)

{I.S. : }

{F.S. : }

{Kamus Lokal}

var

pNew, pHelp : Ptr

i : integer

{Algoritma}

begin

new(pNew)

new(pHelp)

if (n > 1) then

begin

i <- 1

gotoxy(15,9) writeln('------------------------------------------------------------------------------------')

gotoxy(15,10) writeln('| Input Student Data |')

gotoxy(15,11) writeln('------------------------------------------------------------------------------------')

gotoxy(15,12) writeln('| No | Student ID | Name | GPA |')

gotoxy(15,13) writeln('------------------------------------------------------------------------------------')

gotoxy(15,14) writeln('| | | | |')

gotoxy(15,15) writeln('------------------------------------------------------------------------------------')

gotoxy(17,14) write(i)

gotoxy(24,14) readln(pNew^.info.id)

gotoxy(53,14) readln(pNew^.info.name)

gotoxy(92,14) readln(pNew^.info.gpa)

// if (pHead = nil) then

// begin

// pNew^.pPrev <- nil

// pNew^.pNext <- nil

// pHead <- pNew

// pTail <- pNew

// end

// else

// begin

pHelp <- pHead

while (pHelp^.pNext <> nil) do

begin

pHelp <- pHelp^.pNext

end

pNew^.pPrev <- pHelp^.pPrev

pNew^.pNext <- pHelp

pHelp^.pPrev^.pNext <- pNew

pHelp^.pPrev <- pNew

inc(n)

end

else

begin

write('Data harus lebih dari 2!') readln

end

end // EndProcedure insertAtMiddle

procedure insertAtBack(var pHead, pTail : Ptr)

{I.S. : }

{F.S. : }

{Kamus Lokal}

var

pNew : Ptr

i : integer

{Algoritma}

begin

new(pNew)

pNew^.pNext <- nil

i <- 1

gotoxy(15,9) writeln('------------------------------------------------------------------------------------')

gotoxy(15,10) writeln('| Input Student Data |')

gotoxy(15,11) writeln('------------------------------------------------------------------------------------')

gotoxy(15,12) writeln('| No | Student ID | Name | GPA |')

gotoxy(15,13) writeln('------------------------------------------------------------------------------------')

gotoxy(15,14) writeln('| | | | |')

gotoxy(15,15) writeln('------------------------------------------------------------------------------------')

gotoxy(17,14) write(i)

gotoxy(24,14) readln(pNew^.info.id)

gotoxy(53,14) readln(pNew^.info.name)

gotoxy(92,14) readln(pNew^.info.gpa)

if (pHead = nil) then

begin

pNew^.pPrev <- nil

pHead <- pNew

end

else

begin

pNew^.pPrev <- pTail

pTail^.pNext <- pNew

end

pTail <- pNew

inc(n)

end // EndProcedure insertAtBack

procedure deleteAtFront(var pHead, pTail : Ptr)

{I.S. : }

{F.S. : }

{Kamus Lokal}

var

pDel : Ptr

{Algoritma}

begin

new(pDel)

if (n > 0) then

begin

pDel <- pHead

pHead <- pHead^.pNext

pHead^.pPrev <- nil

dispose(pDel)

end

else

begin

writeln('There is no data')

end

dec(n)

end // EndProcedure deleteAtFront

procedure deleteAtMiddle(var pHead, pTail : Ptr)

{I.S. : }

{F.S. : }

{Kamus Lokal}

var

pDel : Ptr

data : string

{Algoritma}

begin

new(pDel)

if (n > 0) then

begin

pDel <- pHead

write('Student Id to delete: ') readln(data)

while (pDel <> nil) and (pDel^.info.id <> data) do

begin

pDel <- pDel^.pNext

end

pDel^.pPrev^.pNext <- pDel^.pNext

pDel^.pNext^.pPrev <- pDel^.pPrev

dispose(pDel)

end

else

begin

writeln('There is no data')

end

dec(n)

end // EndProcedure deleteAtMiddle

procedure deleteAtBack(var pHead, pTail : Ptr)

{I.S. : }

{F.S. : }

{Kamus Lokal}

var

pDel : Ptr

i : integer

{Algoritma}

begin

new(pDel)

if (n > 0) then

begin

pDel <- pTail

pTail <- pTail^.pPrev

pTail^.pNext <- nil

dispose(pDel)

end

else

begin

writeln('There is no data')

end

dec(n)

end // EndProcedure deleteAtBack

procedure destroy(var pHead, pTail : Ptr)

{I.S. : pHead dan pTail belum terdefinisi}

{F.S. : Menghancurkan nilai pada pHead dan pTail}

{Kamus Lokal}

var

pDel : Ptr

{Algoritma}

begin

pDel <- pHead

while (pDel <> nil) do

begin

pHead <- pHead^.pNext

dispose(pDel)

pDel <- pHead

end

pTail <- nil

end // EndProcedure destroy()

procedure showMenu(var select : integer)

{I.S. : }

{F.S. : Menampilkan menu}

{Kamus Tidak Ada}

{Algoritma}

begin

clrscr

gotoxy(50,5) writeln('-------------------------------------------')

gotoxy(50,6) writeln('| College Student Admission |')

gotoxy(50,7) writeln('-------------------------------------------')

gotoxy(50,8) writeln('| 1. Insert Data |')

gotoxy(50,9) writeln('| 2. Delete Data |')

gotoxy(50,10) writeln('| 3. Search Data |')

gotoxy(50,11) writeln('| 4. Show Data |')

gotoxy(50,12) writeln('| 5. Exit |')

gotoxy(50,13) writeln('-------------------------------------------')

gotoxy(50,14) write('Select : ') readln(select)

end // EndProcedure showMenu()

procedure addData(var pHead, pTail : Ptr)

{I.S. : }

{F.S. : }

var

select : integer

{ Algoritma }

begin

gotoxy(50,5) writeln('-------------------------------------------')

gotoxy(50,6) writeln('| Select insertion method |')

gotoxy(50,7) writeln('-------------------------------------------')

gotoxy(50,8) writeln('| 1. Insert at Front |')

gotoxy(50,9) writeln('| 2. Insert at Middle |')

gotoxy(50,10) writeln('| 3. Insert at Back |')

gotoxy(50,11) writeln('-------------------------------------------')

gotoxy(50,12) write('Select : ') readln(select)

clrscr

//mencari berdasarkan judul

if (select = 1) then

begin

insertAtFront(pHead, pTail)

// write('front') readln

end

else if (select = 2) then

begin

insertAtMiddle(pHead, pTail)

// write('middle') readln

end

else if (select = 3) then

begin

insertAtBack(pHead, pTail)

// write('back') readln

end

end

procedure deleteData(var pHead, pTail : Ptr)

{I.S. : }

{F.S. : }

var

select : integer

{ Algoritma }

begin

gotoxy(50,5) writeln('-------------------------------------------')

gotoxy(50,6) writeln('| Select deletion method |')

gotoxy(50,7) writeln('-------------------------------------------')

gotoxy(50,8) writeln('| 1. Delete at Front |')

gotoxy(50,9) writeln('| 2. Delete at Middle |')

gotoxy(50,10) writeln('| 3. Delete at Back |')

gotoxy(50,11) writeln('-------------------------------------------')

gotoxy(50,12) write('Select : ') readln(select)

clrscr

if (select = 1) then

begin

deleteAtFront(pHead, pTail)

// write('front') readln

end

else if (select = 2) then

begin

deleteAtMiddle(pHead, pTail)

// write('middle') readln

end

else if (select = 3) then

begin

deleteAtBack(pHead, pTail)

// write('back') readln

end

end

procedure showData(pHead : Ptr)

{I.S. : }

{F.S. : }

{Kamus Lokal}

var

pHelp : Ptr

i, k : integer

{Algoritma}

begin

gotoxy(15,9) writeln('------------------------------------------------------------------------------------')

gotoxy(15,10) writeln('| Display Student Data |')

gotoxy(15,11) writeln('------------------------------------------------------------------------------------')

gotoxy(15,12) writeln('| No | Student ID | Name | GPA |')

gotoxy(15,13) writeln('------------------------------------------------------------------------------------')

pHelp <- pHead

i <- 1 k <- 13

while (pHelp <> nil) do

begin

gotoxy(15,k+i) writeln('| | | | |')

gotoxy(17,k+i) write(i)

gotoxy(24,k+i) write(pHelp^.info.id)

gotoxy(53,k+i) write(pHelp^.info.name)

gotoxy(92,k+i) write(pHelp^.info.gpa:2:1)

inc(i)

pHelp <- pHelp^.pNext

end

gotoxy(15,k+i) writeln('------------------------------------------------------------------------------------')

readln

end

procedure sort(pHead : Ptr var pHead2, pTail2 : Ptr)

var

pNew, pHelp, pHelp2 : Ptr

method : integer

begin

pHead2 <- nil

pTail2 <- nil

pHelp <- pHead

write('Sort by ID (1) / Name (2): ') readln(method)

while (pHelp <> nil) do

begin

New(pNew)

pNew^.info <- pHelp^.info

if (pHead2 = nil) then

begin

pNew^.pPrev <- nil

pNew^.pNext <- nil

pHead2 <- pNew

pTail2 <- pNew

end

else

begin

case (method) of

1: begin

if (pNew^.info.id < pHead2^.info.id) then

begin

//SisipDepan

pNew^.pPrev <- nil

pNew^.pNext <- pHead2

pHead2^.pPrev <- pNew

pHead2 <- pNew

end

else

begin

if(pNew^.info.id > pTail2^.info.id) then

begin

//SisipBelakang

pNew^.pNext <- nil

pNew^.pPrev <- pTail2

pTail2^.pNext <- pNew

pTail2 <- pNew

end

else

begin

//SisipTengah

pHelp2 <- pHead2

while (pHelp2^.info.id < pNew^.info.id) do

begin

pHelp2 <- pHelp2^.pNext

end

pNew^.pNext <- pHelp2

pNew^.pPrev <- pHelp2^.pPrev

pHelp2^.pPrev^.pNext <- pNew

pHelp2^.pPrev <- pNew

end

end

end

2: begin

if (pNew^.info.name < pHead2^.info.name) then

begin

//SisipDepan

pNew^.pPrev <- nil

pNew^.pNext <- pHead2

pHead2^.pPrev <- pNew

pHead2 <- pNew

end

else

begin

if(pNew^.info.name > pTail2^.info.name) then

begin

//SisipBelakang

pNew^.pNext <- nil

pNew^.pPrev <- pTail2

pTail2^.pNext <- pNew

pTail2 <- pNew

end

else

begin

//SisipTengah

pHelp2 <- pHead2

while (pHelp2^.info.id < pNew^.info.id) do

begin

pHelp2 <- pHelp2^.pNext

end

pNew^.pNext <- pHelp2

pNew^.pPrev <- pHelp2^.pPrev

pHelp2^.pPrev^.pNext <- pNew

pHelp2^.pPrev <- pNew

end

end

end

end // EndDependOn

end

pHelp <- pHelp^.pNext

end

end // EndProcedure sort

procedure searchIds(var pHead, pTail : Ptr)

var

studentId : string

pHelp : Ptr

found : boolean

begin

write('Student Id to search : ')

readln(studentId)

pHelp <- pHead

found <- false

while (Not found) and (pHelp <> Nil) do

begin

if(pHelp^.Info.id = studentId)

then

begin

found <- true

end

else

begin

pHelp <- pHelp^.pNext

end

end

if (found) then

begin

showData(pHelp)

end

else

begin

writeln('student id ',studentId, ' not found')

end

end

procedure searchName(var pHead, pTail : Ptr)

var

studentName : string

pHelp : Ptr

found : boolean

begin

write('Student Name to search : ')

readln(studentName)

pHelp <- pHead

found <- false

while (Not found) and (pHelp <> Nil) do

begin

if(pHelp^.Info.name = studentName)

then

begin

found <- true

end

else

begin

pHelp <- pHelp^.pNext

end

end

if (found) then

begin

showData(pHelp)

end

else

begin

writeln('student name ',studentName, ' not found')

end

end

{ Algoritma Utama }

begin

create(pHead, pTail)

n <- 0 // total awal data

repeat

showMenu(select)

clrscr

case (select) of

1:

begin

addData(pHead, pTail)

end

2:

begin

deleteData(pHead, pTail)

end

3:

begin

write('Sort by ID (1) / Name (2): ') readln(toSearch)

case (toSearch) of

1:

begin

searchIds(pHead, pTail)

end

2:

begin

searchName(pHead, pTail)

end

end // EndDependOn

end

4:

begin

sort(pHead, pHead2, pTail2)

showData(pHead2)

end

end // EndDependOn

until (select <= 0) or (select > 4)

destroy(pHead, pTail)

writeln('Data has been successfully destroyed!')

end.