[Singly-linked-list.svg](http://id.wikipedia.org/wiki/Berkas:Singly-linked-list.svg)

source code [pemrograman pascal](https://www.google.com/search?q=pascal" \t "_blank) untuk Single linked list :

Program SingleLinkList;

uses CRT;

Type

SLL=^data;

data=record

info:integer;

next:SLL;

end;

List=SLL;

Node=SLL;

var L:list;

pilih:char;

sum,ElmIn:Integer;

Procedure CreateList (Var L:List);

begin

L:=nil;

end;

Procedure InsertFirst (Var L:List; elm:integer);

var

P:Node;

begin

New(P);

P^.info:=Elm;

if L = nil then

Begin

L:=P;

P^.next:=Nil;

end

else

begin

P^.next:=L;

L:=P;

end;

end;

Procedure InsertLast (Var L:list;elm:integer);

var

Pt,P : Node;

begin

new(P);

P^.info:=elm;

if (L=Nil) then

begin

L:=P;

P^.next:=nil;

end

else

begin

Pt:=L;

while (Pt^.next<>Nil) do

Pt:=Pt^.next;

P^.next:=Nil;

Pt^.next:=P;

end;

end;

procedure DeleteFirst(var L:List);

var P : Node;

begin

if (L<>Nil) then

begin

if L^.next = nil then

begin

P:=L;

dispose(P);

L:=nil;

end

else

begin

P:=L;

L:=L^.next;

P^.next:=Nil;

dispose(P);

end;

end;

end;

procedure DeleteLast(var L:List);

var Prec,Pt : Node;

begin

if (L<>Nil) then

begin

Pt:=L;

Prec:=Nil;

while (Pt^.next<>Nil) do

begin

Prec:=Pt;

Pt:=Pt^.next;

end;

if (Prec=Nil) then

begin

dispose(pt);

L:=nil;

end

else

begin

Prec^.next:=Nil;

dispose(Pt);

end;

end;

end;

Procedure penjumlahan(var L:List; var sum: integer);

var pt: node;

begin

Sum:=0;

Pt:= L;

Sum:=Sum+Pt^.info;

while Pt^.next<>nil do

begin

Pt:=Pt^.next;

Sum:=Sum+Pt^.info;

end;

writeln;

write ('Hasil penjumlahan elemen single link list adalah ',sum,'');

writeln;

end;

Procedure Tampil (Var L:List);

Var P:Node;

Begin

clrscr;

if L<> Nil then

begin

P:=L;

write (P^.info);

write (' ');

P:=P^.next;

while P<>nil do

begin

write (p^.info);

write (' ');

p:=P^.next;

end;

end;

writeln;

end;

label ulang;

begin

createlist(L);

ulang:

clrscr;

tampil(L);

writeln;

writeln;

writeln ('============== Program Single Linked List ==============');

writeln;

writeln('1. Insert First');

writeln('2. Insert Last');

writeln('3. Delete Fisrt');

writeln('4. Delete Last');

writeln('5. Penjumlahan');

writeln('0. Exit');

write('Pilih = ');

readln(pilih);

case pilih of

'1' : begin

write('Masukkan Elemen Single Linked List = ');

Readln (ElmIn);

InsertFirst(L,ElmIn);

tampil(L);

end;

'2' : begin

write('Masukkan Elemen Single Linked List = ');

Readln (ElmIn);

insertlast(L,ElmIn);

tampil(L);

end;

'3' : begin

deletefirst(L);

tampil(L);

end;

'4' : begin

deletelast(L);

tampil(L);

end;

'5' : begin

penjumlahan(L, sum);

end;

'0':exit;

end;

writeln;

writeln ('Tekan Enter Untuk Mengulang Program');

readln;

goto ulang;

end.

untuk single linked circular, dasarnya adalah pentautan node akhir kepada node awal.  
ilustrasinya adalah sebagai berikut :

[Circularly-linked-list.svg](http://id.wikipedia.org/wiki/Berkas:Circularly-linked-list.svg)

berikut adalah source code untuk program single linked list circular :

Program singlelinkedlistcircular;

Uses crt;

Const N = 100;

Type Queve = record

isi : Array[1..N]of integer;

head:integer;

tail: integer;

end;

Procedure CreateQueve(Var S:Queve);

begin

S.head:=0;

S.tail:=0;

end;

Function IsFull(S:Queve):boolean;

begin

IsFull := (S.head = 1) and (S.tail = N);

end;

Function IsEmpty(S:Queve):boolean;

begin

IsEmpty := (S.head = 0 ) and (S.tail = 0);

end;

Procedure add(X:integer; var S:Queve);

begin

If Not IsFull(S) then

begin

S.tail := S.tail + 1;

S.isi[S.tail] := X;

if S.head=0 then

S.head:=S.head+1;

end;

end;

Procedure remove(var X:integer; var S:Queve);

var i:integer;

begin

IF NOt IsEmpty(S) then

begin

X:=S.isi[S.head];

for i:= 2 to S.tail do

S.isi[i-1]:=S.isi[i];

S.tail :=S.tail - 1;

if S.tail=0 then

S.head:=S.head-1;

end;

end;

{Main Program}

Var S : Queve;

X,i,z,m,Y: integer;

Begin

write('masukan jumlah data ');readln(X);

CreateQueve(S);

for i:= 1 to X do

begin

write('masukan data ke',i,' : ');readln(m);

add(m,S);

end;

writeln;

write('jumlah data yang akan dikeluarkan ');readln(Y);

writeln('data yang dikeluarkan : ');

for i:= 1 to Y do

begin

remove(z,S);

writeln(z);

end;

writeln;

writeln('data yang tersisa : ');

for i:= 1 to S.tail do

writeln(S.isi[i]);

writeln;

writeln('S.tail berada di elemen ke ',S.tail);

writeln('S.head berada di elemen ke ',S.head);

End.