PERTEMUAN XIII

QUEUE

TUJUAN PRAKTIKUM

a) Mahasiswa dapat menjelaskan pengerian dan pembuatan Queue dengan C++.
b) Mahasiswa dapat melakukan operasi penyisipan dan penghapusan elemen dalam queue dengan C++.
c) Mahasiswa dapat mengimplementasikan Queue dengan C++.

TEORI DASAR

a) Pendahuluan

Queue atau Antrian merupakan kumpulan elemen dengan penyisipan dan penghapusan elemen yang dilakukan dari sisi/gerbang yang berbeda. Penyisipan dilakukan dari gerbang belakang dan penghapusan dilakukan dari gerbang depan. Hal ini menunjukan bahwa untuk Queue mempunyai dua gerbang yaitu gerbang depan dan gerbang belakang. Dengan demikian dapat dilihat bahwa Queue mempunyai sifat FIFO (first In Firs Out), yaitu elemen yang pertama masuk akan keluar pertama juga. Queue dapat direpresentasikan dengan menggunakan Array atau Linked List.

b) Operasi-operasi pada Queue

Sama halnya dengan stack, operasi yang dapat dilakukan pada suatu Queue pada dasarnya adalah penyisipan elemen dan penghapusan elemen. Disamping itu juga dapat dilakukan operasi untuk mengecek apakah Queue kosong atau penuh, macam-macam operasi untuk mengecek Queue antara lain :

1) Operasi Inisialisasi
2) Operasi Queue Kosong
3) Operasi Queue Penuh
4) Operasi Mengosongkan Queue
5) Operasi Penyisipan Elemen Queue
6) Operasi Penghapusan Elemen Queue
7) Operasi Pencetakan Isi Queue

TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Queue!
2. Tuliskan Deklarasi sintaks Queue!
3. Sebutkan dan Jelaskan operasi-operasi pada Queue!
4. Jelaskan Aplikasi-Aplikasi Queue dalam dunia nyata!

JAWABAN

1. Queue atau Antrian merupakan kumpulan elemen dengan penyisipan dan penghapusan elemen yang dilakukan dari sisi/gerbang yang berbeda.
2. **Deklarasi Awal Queue**.

```
#define max 7

int data[max];
int head=-1, tail=-1;
IsEmpty.
bool IsEmpty(){
  if(head == -1 && tail == -1)
     return true;
  else
     return false;
}
```

IsFull.

```
</pre>
<pre style="text-align: justify;">bool IsFull(){
  if(tail == max-1)
     return true;
  else
     return false;
}
```

Enqueue.

```
void Enqueue(){
  if(IsFull()) {
     cout<<"Antrian sudah penuh, mohon tunggu beberapa saat lagi ";
     getch();
  } else {
     if (IsEmpty()){
        head=tail=0;
        cout<<"Masukkan data : ";cin>>data[tail];
     } else {
        tail++;
        cout<<"Masukkan data : ";cin>>data[tail];
     }
  }
}
```

Dequeue.

```
void Dequeue(){
     if(IsEmpty()){
        cout<<"Antrian kosong ! ";
        getch();
     } else {
     cout<<"Data yang diambil : "<<data[head];
     for(a=head;a<=tail-1;a++)
        data[a]=data[a+1];
     tail--;
     if(tail == -1)
head = -1;
     }
}
```

Clear.

```
void Clear(){
    head=tail=-1;
    cout<<"Antrian sudah dikosongkan ! ";getch();
}
```

View.

```
void View(){
    if(IsEmpty()){
       cout<<"Antrian kosong ! ";
       getch();
    } else {
       for(a=head;a<=tail;a++)
           cout<<"Data pada antrian ke "<<a<<" = "<<data[a]<<endl;
       getch();
    }
}
```

3. Operasi-operasi pada Queue:
   a. **Create()**

   Untuk menciptakan dan menginisialisasi Queue
   Dengan cara membuat Head dan Tail = -1.

   b. **IsEmpty()**

   Untuk memeriksa apakah Antrian sudah penuh atau belum
   Dengan cara memeriksa nilai Tail, jika Tail = -1 maka empty
   Kita tidak memeriksa Head, karena Head adalah tanda untuk kepala antrian (elemen pertama dalam antrian) yang tidak akan berubah-ubah
   Pergerakan pada Antrian terjadi dengan penambahan elemen Antrian kebelakang, yaitu menggunakan nilai Tail.

   c. **IsFull**

   Untuk mengecek apakah Antrian sudah penuh atau belum
   Dengan cara mengecek nilai Tail, jika Tail >= MAX-1 (karena MAX-1 adalah batas elemen array pada C) berarti sudah penuh.

   d. **Enqueue**

Untuk menambahkan elemen ke dalam Antrian, penambahan elemen selalu ditambahkan di elemen paling belakang

Penambahan elemen selalu menggerakan variabel Tail dengan cara increment counter Tail terlebih dahulu.

e. **Dequeue()**

Digunakan untuk menghapus elemen terdepan/pertama (head) dari Antrian

Dengan cara menggeser semua elemen antrian kedepan dan mengurangi Tail dgn 1

Penggeseran dilakukan dengan menggunakan looping.

f. **Clear()**

Untuk menghapus elemen-elemen Antrian dengan cara membuat Tail dan Head =

-1.

Penghapusan elemen-elemen Antrian sebenarnya tidak menghapus arraynya, namun hanya mengeset indeks pengaksesan-nya ke nilai -1 sehingga elemen-elemen Antrian tidak lagi terbaca.

g. **Tampil()**

Untuk menampilkan nilai-nilai elemen Antrian

Menggunakan looping dari head s/d tail.

4. Queue dalam kehidupan sehari-hari seperti :

   Antrian pada penjualan tiket kereta api, dimana orang yang pertama datang adalah orang yang pertama kali dilayani untuk membeli tiket. Jika ada orang baru yang datang akan membeli tiket, maka posisinya berada pada urutan paling belakang dalam antrian tersebut.

   Orang yang berada pada posisi terakhir dalam antrian adalah yang terakhir kali dapat dilayani dan memperoleh tiket kereta api (kalau kurang beruntung, maka akan kehabisan tiket). Contoh lain adalah nasabah yang antri di teller bank, paket data yang menunggu untuk di transmisikan lewat internet, antrian printer dimana terdapat antrian print job yang menunggu giliran untuk menggunakan printer dan sebagainya.

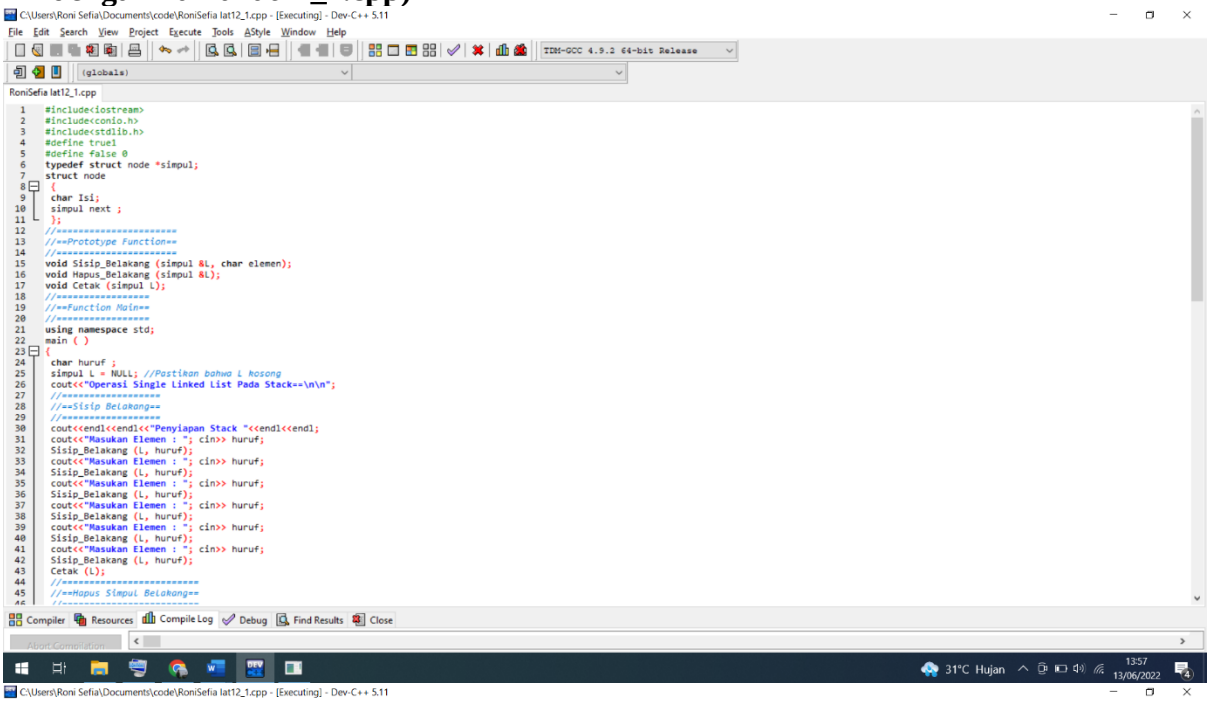
   Contoh aplikasi Queue dalam dunia nyata :

   1) Aplikasi antrian di jalan Tol.
   2) Aplikasi antrian saat mengantri di loket.
   3) Aplikasi antraian reservasi tiket kereta api, dll.

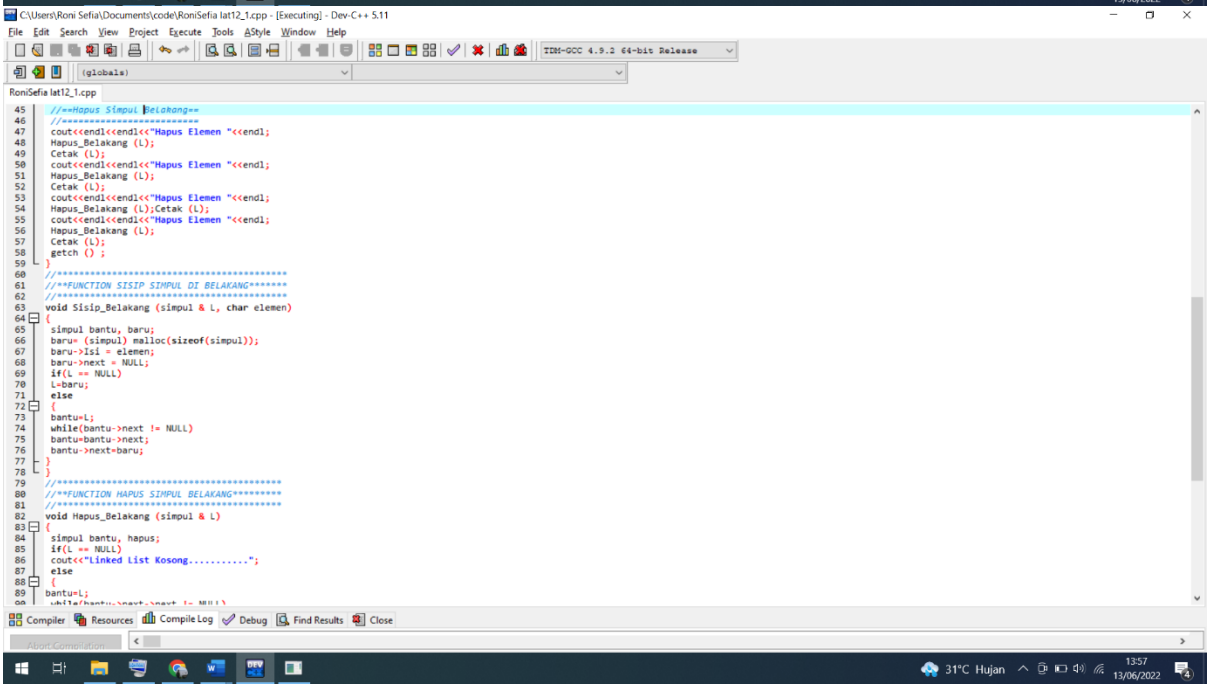
   Semua itu menggunakan aturan FIFO (First In, First Out).

TUGAS PRAKTIKUM

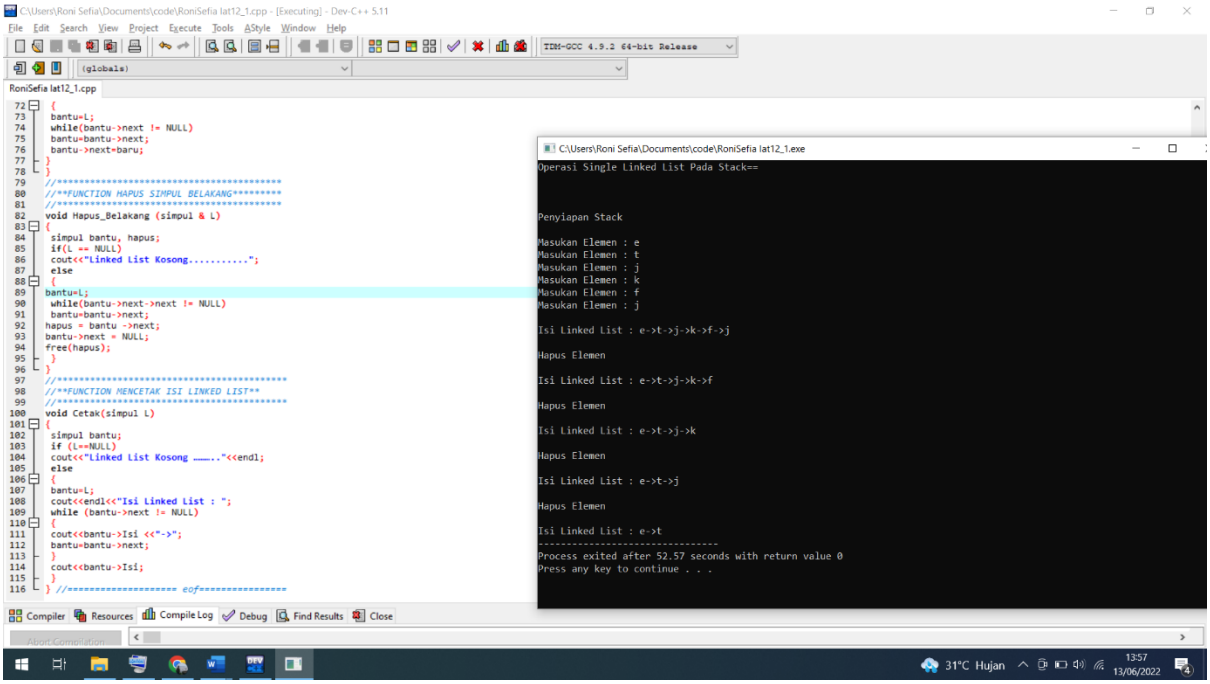
a) Buatlah program Stack dengan menggunakan Singly Linked List (simpan dengan nama lat12_1.cpp)



```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
#define true1
#define false 0
typedef struct node *simpul;
struct node
{
 char Isi;
 simpul next ;
};
//=====================
//==Prototype Function==
//=====================
void Sisip_Belakang (simpul &L, char elemen);
void Hapus_Belakang (simpul &L);
void Cetak (simpul L);
//=================
//==Function Main==
//=================
using namespace std;
main ( )
{
 char huruf ;
 simpul L = NULL; //Pastikan bahwa L kosong
 cout<<"Operasi Single Linked List Pada Stack==\n\n";
 //=================
 //==Sisip Belakang==
 //=================
 cout<<endl<<endl<<"Penyiapan Stack "<<endl<<endl;
 cout<<"Masukan Elemen : "; cin>> huruf;
 Sisip_Belakang (L, huruf);
 cout<<"Masukan Elemen : "; cin>> huruf;
 Sisip_Belakang (L, huruf);
 cout<<"Masukan Elemen : "; cin>> huruf;
 Sisip_Belakang (L, huruf);
 cout<<"Masukan Elemen : "; cin>> huruf;
 Sisip_Belakang (L, huruf);
 cout<<"Masukan Elemen : "; cin>> huruf;
 Sisip_Belakang (L, huruf);
 cout<<"Masukan Elemen : "; cin>> huruf;
 Sisip_Belakang (L, huruf);
 Cetak (L);
 //========================
 //==Hapus Simpul Belakang==
 //========================
```



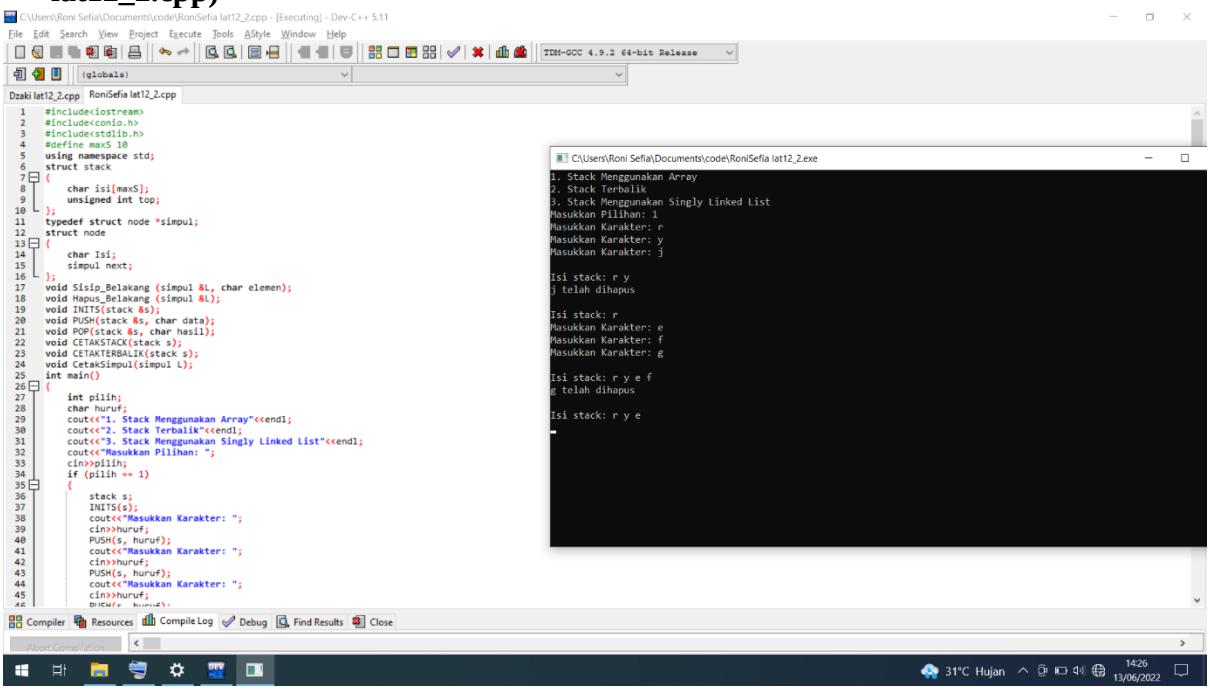
```
 //==Hapus Simpul Belakang==
 //========================
 cout<<endl<<endl<<"Hapus Elemen "<<endl;
 Hapus_Belakang (L);
 Cetak (L);
 cout<<endl<<endl<<"Hapus Elemen "<<endl;
 Hapus_Belakang (L);
 Cetak (L);
 cout<<endl<<endl<<"Hapus Elemen "<<endl;
 Hapus_Belakang (L);Cetak (L);
 cout<<endl<<endl<<"Hapus Elemen "<<endl;
 Hapus_Belakang (L);
 Cetak (L);
 getch () ;
}
//*****************************************
//**FUNCTION SISIP SIMPUL DI BELAKANG*******
//*****************************************
void Sisip_Belakang (simpul & L, char elemen)
{
 simpul bantu, baru;
 baru= (simpul) malloc(sizeof(simpul));
 baru->Isi = elemen;
 baru->next = NULL;
 if(L == NULL)
 L=baru;
 else
 {
 bantu=L;
 while(bantu->next != NULL)
 bantu=bantu->next;
 bantu->next=baru;
}
}
//*****************************************
//**FUNCTION HAPUS SIMPUL BELAKANG*********
//*****************************************
void Hapus_Belakang (simpul & L)
{
 simpul bantu, hapus;
 if(L == NULL)
 cout<<"Linked List Kosong.....";
 else
 {
 bantu=L;
```



```
 {
  bantu=L;
  while(bantu->next != NULL)
  bantu=bantu->next;
  bantu->next=baru;
 }
 }
 //*******************************************
 //**FUNCTION HAPUS SIMPUL BELAKANG*********
 //*******************************************
 void Hapus_Belakang (simpul & L)
 {
  simpul bantu, hapus;
  if(L == NULL)
  cout<<"Linked List Kosong.....";
  else
  {
 bantu=L;
  while(bantu->next->next != NULL)
  bantu=bantu->next;
 hapus = bantu ->next;
 bantu->next = NULL;
 free(hapus);
  }
 }
 //*******************************************
 //**FUNCTION MENCETAK ISI LINKED LIST**
 //*******************************************
 void Cetak(simpul L)
 {
  simpul bantu;
  if (L==NULL)
  cout<<"Linked List Kosong ……..."<<endl;
  else
  {
  bantu=L;
  cout<<endl<<"Isi Linked List : ";
  while (bantu->next != NULL)
  {
  cout<<bantu->Isi <<"->";
  bantu=bantu->next;
  }
  cout<<bantu->Isi;
  }
 } //****************** eof***************
```

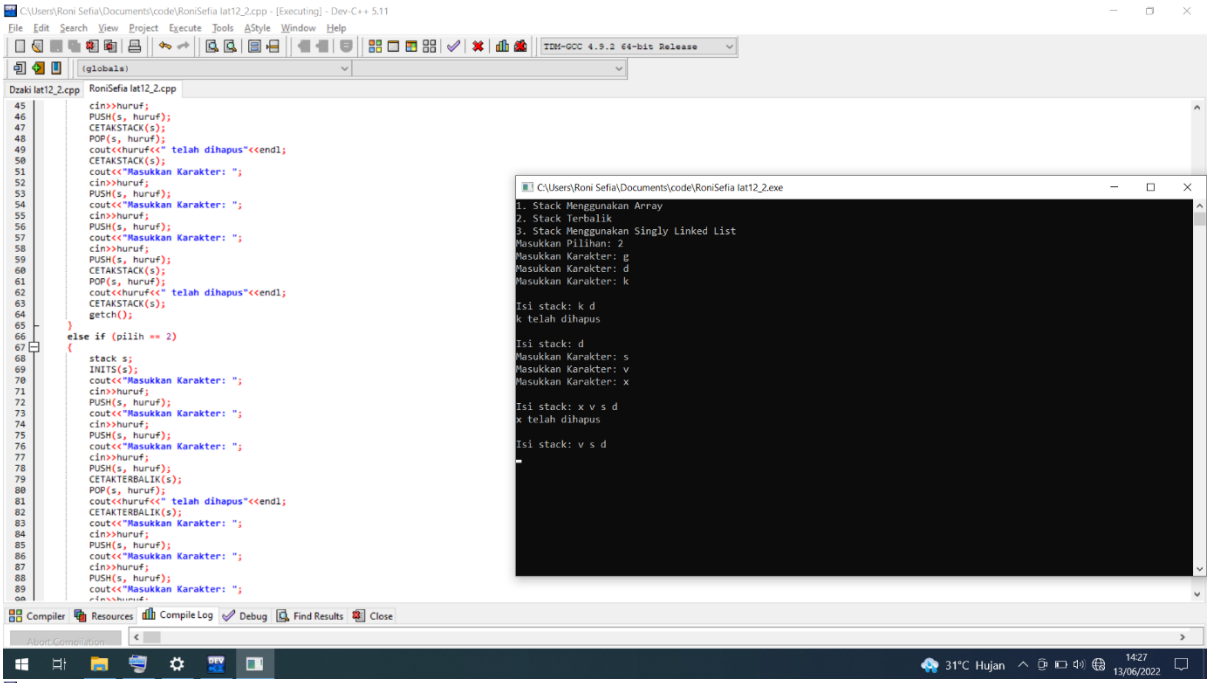
```
Operasi Single Linked List Pada Stack==



Penyiapan Stack

Masukan Elemen : e
Masukan Elemen : t
Masukan Elemen : j
Masukan Elemen : k
Masukan Elemen : f
Masukan Elemen : j

Isi Linked List : e->t->j->k->f->j

Hapus Elemen

Isi Linked List : e->t->j->k->f

Hapus Elemen

Isi Linked List : e->t->j->k

Hapus Elemen

Isi Linked List : e->t->j

Hapus Elemen

Isi Linked List : e->t
--------------------------------
Process exited after 52.57 seconds with return value 0
Press any key to continue . . .
```

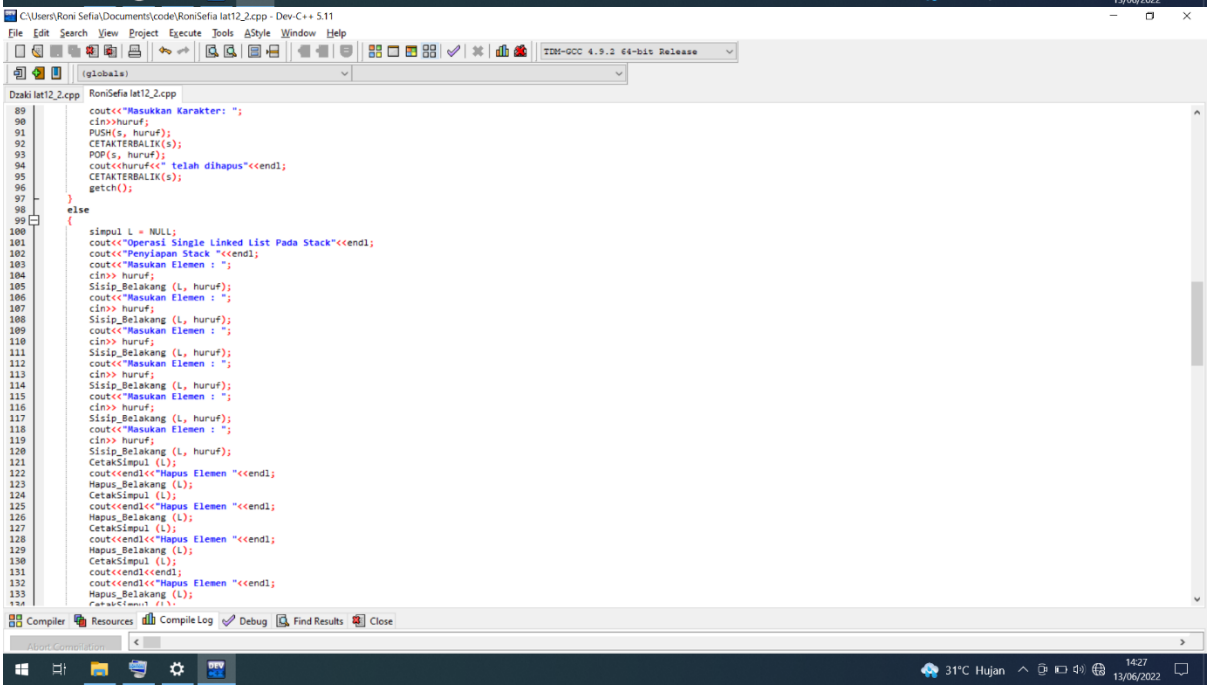
b) Buatlah program menu untuk menampilkan program-program pada pertemuan XI dan XII, menggunakan perintah IF (simpan dengan nama lat12_2.cpp)

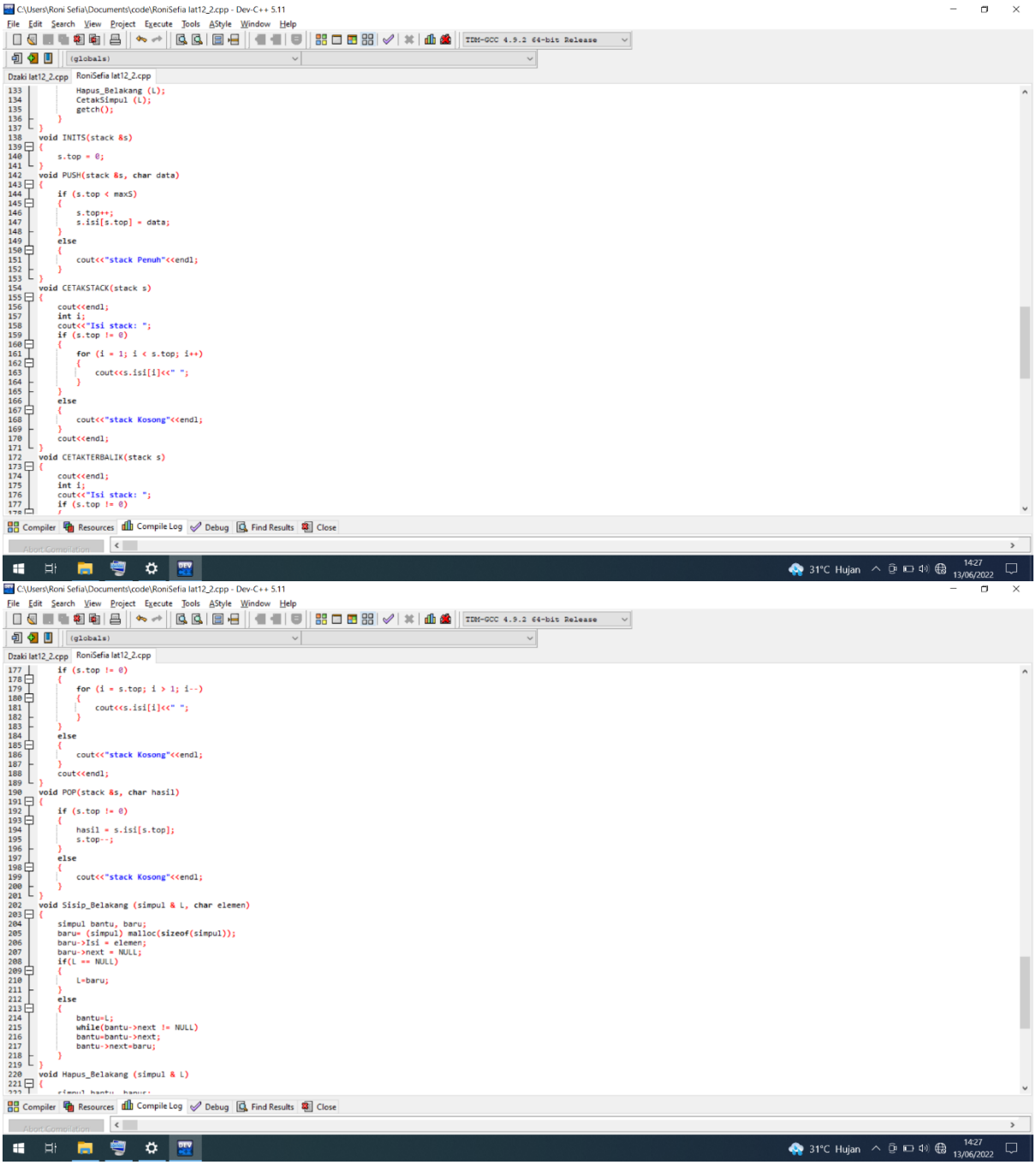


```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
#define maxS 10
using namespace std;
struct stack
{
    char isi[maxS];
    unsigned int top;
};
typedef struct node *simpul;
struct node
{
    char Isi;
    simpul next;
};
void Sisip_Belakang (simpul &L, char elemen);
void Hapus_Belakang (simpul &L);
void INITS(stack &s);
void PUSH(stack &s, char data);
void POP(stack &s, char hasil);
void CETAKSTACK(stack s);
void CETAKTERBALIK(stack s);
void CetakSimpul(simpul L);
int main()
{
    int pilih;
    char huruf;
    cout<<"1. Stack Menggunakan Array"<<endl;
    cout<<"2. Stack Terbalik"<<endl;
    cout<<"3. Stack Menggunakan Singly Linked List"<<endl;
    cout<<"Masukkan Pilihan: ";
    cin>>pilih;
    if (pilih == 1)
    {
        stack s;
        INITS(s);
        cout<<"Masukkan Karakter: ";
        cin>>huruf;
        PUSH(s, huruf);
        cout<<"Masukkan Karakter: ";
        cin>>huruf;
        PUSH(s, huruf);
        cout<<"Masukkan Karakter: ";
        cin>>huruf;
```

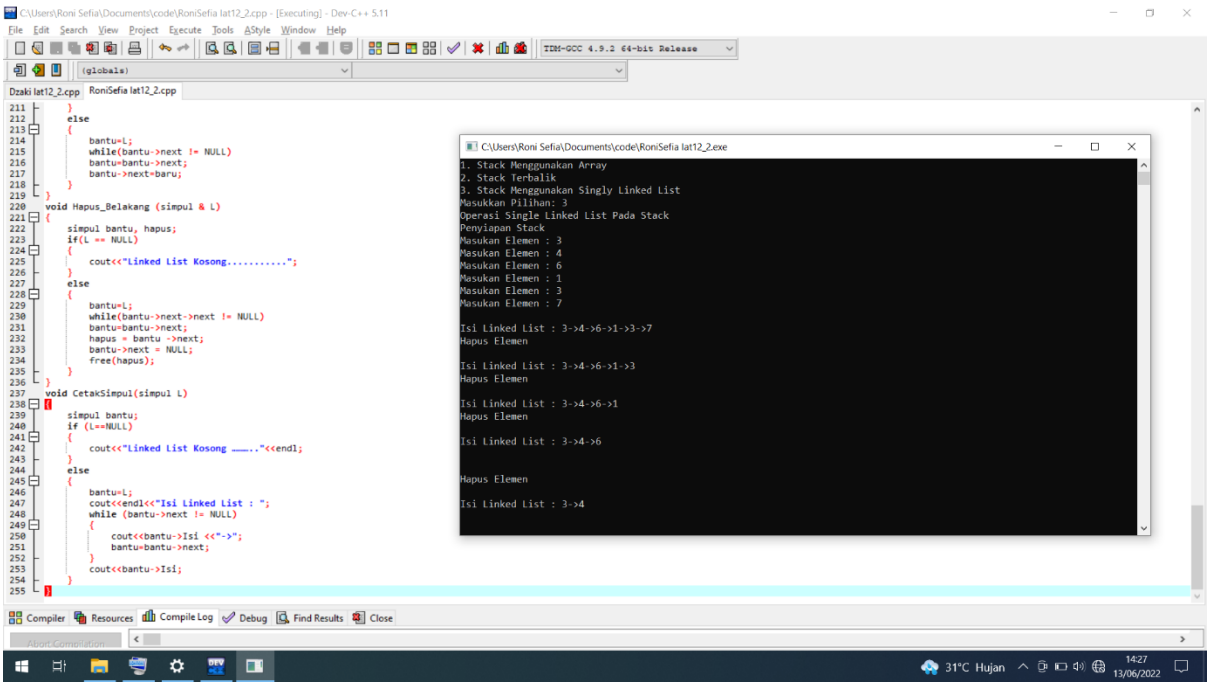
```
1. Stack Menggunakan Array
2. Stack Terbalik
3. Stack Menggunakan Singly Linked List
Masukkan Pilihan: 1
Masukkan Karakter: r
Masukkan Karakter: y
Masukkan Karakter: j

Isi stack: r y
j telah dihapus

Isi stack: r
Masukkan Karakter: e
Masukkan Karakter: f
Masukkan Karakter: g

Isi stack: r y e f
g telah dihapus

Isi stack: r y e
```


C:\Users\Roni Sefia\Documents\code\RoniSefia lat12_2.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Dzaki lat12_2.cpp
RoniSefia lat12_2.cpp
cin>>huruf;
PUSH(s, huruf);
CETAKSTACK(s);
POP(s, huruf);
cout<<huruf<<" telah dihapus"<<endl;
CETAKSTACK(s);
cout<<"Masukkan Karakter: ";
cin>>huruf;
PUSH(s, huruf);
cout<<"Masukkan Karakter: ";
cin>>huruf;
PUSH(s, huruf);
cout<<"Masukkan Karakter: ";
cin>>huruf;
PUSH(s, huruf);
CETAKSTACK(s);
POP(s, huruf);
cout<<huruf<<" telah dihapus"<<endl;
CETAKSTACK(s);
getch();
}
else if (pilih == 2)
{
stack s;
INITS(s);
cout<<"Masukkan Karakter: ";
cin>>huruf;
PUSH(s, huruf);
cout<<"Masukkan Karakter: ";
cin>>huruf;
PUSH(s, huruf);
cout<<"Masukkan Karakter: ";
cin>>huruf;
PUSH(s, huruf);
CETAKTERBALIK(s);
POP(s, huruf);
cout<<huruf<<" telah dihapus"<<endl;
CETAKTERBALIK(s);
cout<<"Masukkan Karakter: ";
cin>>huruf;
PUSH(s, huruf);
cout<<"Masukkan Karakter: ";
cin>>huruf;
PUSH(s, huruf);
cout<<"Masukkan Karakter: ";
C:\Users\Roni Sefia\Documents\code\RoniSefia lat12_2.exe
1. Stack Menggunakan Array
2. Stack Terbalik
3. Stack Menggunakan Singly Linked List
Masukkan Pilihan: 2
Masukkan Karakter: g
Masukkan Karakter: d
Masukkan Karakter: k
Isi stack: k d
k telah dihapus
Isi stack: d
Masukkan Karakter: s
Masukkan Karakter: v
Masukkan Karakter: x
Isi stack: x v s d
x telah dihapus
Isi stack: v s d
Compiler Resources Compile Log Debug Find Results Close
31°C Hujan
14:27
13/06/2022

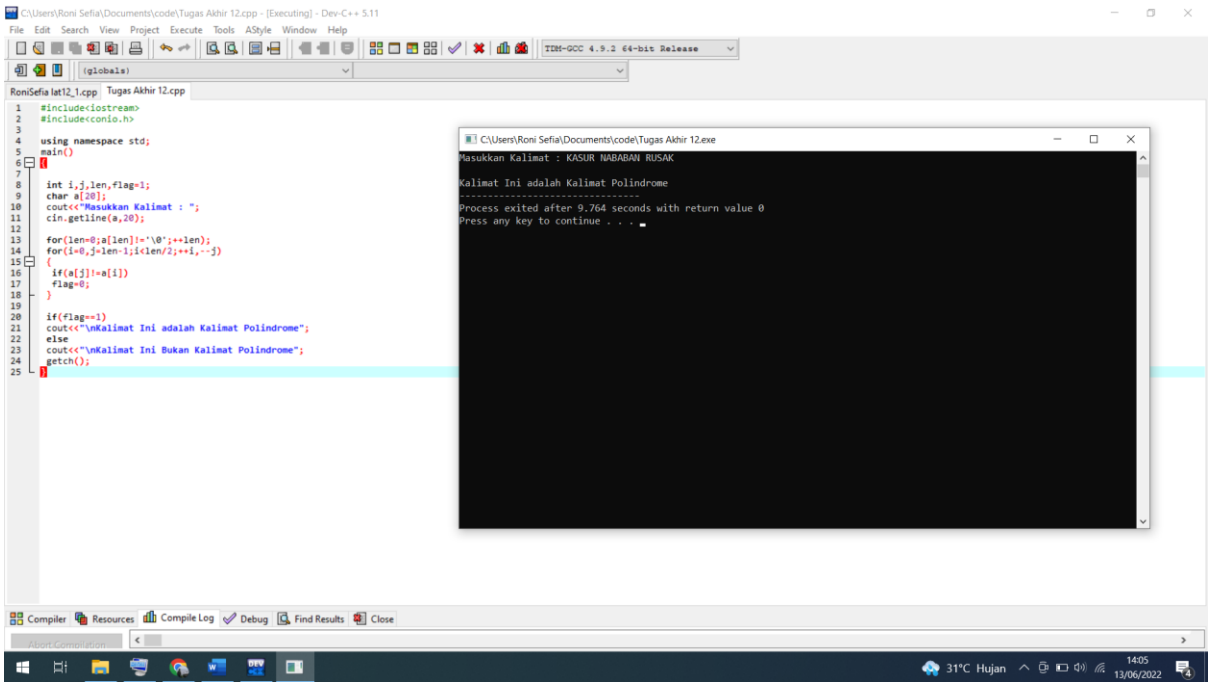

C:\Users\Roni Sefia\Documents\code\RoniSefia lat12_2.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Dzaki lat12_2.cpp
RoniSefia lat12_2.cpp
cout<<"Masukkan Karakter: ";
cin>>huruf;
PUSH(s, huruf);
CETAKTERBALIK(s);
POP(s, huruf);
cout<<huruf<<" telah dihapus"<<endl;
CETAKTERBALIK(s);
getch();
}
else
{
simpul L = NULL;
cout<<"Operasi Single Linked List Pada Stack"<<endl;
cout<<"Penyiapan Stack "<<endl;
cout<<"Masukan Elemen : ";
cin>> huruf;
Sisip_Belakang (L, huruf);
cout<<"Masukan Elemen : ";
cin>> huruf;
Sisip_Belakang (L, huruf);
cout<<"Masukan Elemen : ";
cin>> huruf;
Sisip_Belakang (L, huruf);
cout<<"Masukan Elemen : ";
cin>> huruf;
Sisip_Belakang (L, huruf);
cout<<"Masukan Elemen : ";
cin>> huruf;
Sisip_Belakang (L, huruf);
cout<<"Masukan Elemen : ";
cin>> huruf;
Sisip_Belakang (L, huruf);
CetakSimpul (L);
cout<<endl<<"Hapus Elemen "<<endl;
Hapus_Belakang (L);
CetakSimpul (L);
cout<<endl<<"Hapus Elemen "<<endl;
Hapus_Belakang (L);
CetakSimpul (L);
cout<<endl<<"Hapus Elemen "<<endl;
Hapus_Belakang (L);
CetakSimpul (L);
cout<<endl<<endl;
cout<<endl<<"Hapus Elemen "<<endl;
Hapus_Belakang (L);
Compiler Resources Compile Log Debug Find Results Close
31°C Hujan
14:27
13/06/2022


C:\Users\Roni Sefia\Documents\code\RoniSefia lat12_2.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Dzaki lat12_2.cpp
RoniSefia lat12_2.cpp
Hapus_Belakang (L);
CetakSimpul (L);
getch();
}
}
void INITS(stack &s)
{
s.top = 0;
}
void PUSH(stack &s, char data)
{
if (s.top < maxS)
{
s.top++;
s.isi[s.top] = data;
}
else
{
cout<<"stack Penuh"<<endl;
}
}
void CETAKSTACK(stack s)
{
cout<<endl;
int i;
cout<<"Isi stack: ";
if (s.top != 0)
{
for (i = 1; i < s.top; i++)
{
cout<<s.isi[i]<<" ";
}
}
else
{
cout<<"stack Kosong"<<endl;
}
cout<<endl;
}
void CETAKTERBALIK(stack s)
{
cout<<endl;
int i;
cout<<"Isi stack: ";
if (s.top != 0)
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
31°C Hujan
14:27
13/06/2022

C:\Users\Roni Sefia\Documents\code\RoniSefia lat12_2.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Dzaki lat12_2.cpp
RoniSefia lat12_2.cpp
if (s.top != 0)
{
for (i = s.top; i > 1; i--)
{
cout<<s.isi[i]<<" ";
}
}
else
{
cout<<"stack Kosong"<<endl;
}
cout<<endl;
}
void POP(stack &s, char hasil)
{
if (s.top != 0)
{
hasil = s.isi[s.top];
s.top--;
}
else
{
cout<<"stack Kosong"<<endl;
}
}
void Sisip_Belakang (simpul & L, char elemen)
{
simpul bantu, baru;
baru= (simpul) malloc(sizeof(simpul));
baru->Isi = elemen;
baru->next = NULL;
if(L == NULL)
{
L=baru;
}
else
{
bantu=L;
while(bantu->next != NULL)
bantu=bantu->next;
bantu->next=baru;
}
}
void Hapus_Belakang (simpul & L)
{
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
31°C Hujan
14:27
13/06/2022


C:\Users\Roni Sefia\Documents\code\RoniSefia lat12_2.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Dzaki lat12_2.cpp
RoniSefia lat12_2.cpp
}
else
{
bantu=L;
while(bantu->next != NULL)
bantu=bantu->next;
bantu->next=baru;
}
}
void Hapus_Belakang (simpul & L)
{
simpul bantu, hapus;
if(L == NULL)
{
cout<<"Linked List Kosong.....";
}
else
{
bantu=L;
while(bantu->next->next != NULL)
bantu=bantu->next;
hapus = bantu ->next;
bantu->next = NULL;
free(hapus);
}
}
void CetakSimpul(simpul L)
{
simpul bantu;
if (L==NULL)
{
cout<<"Linked List Kosong ....."<<endl;
}
else
{
bantu=L;
cout<<endl<<"Isi Linked List : ";
while (bantu->next != NULL)
{
cout<<bantu->Isi <<"->";
bantu=bantu->next;
}
cout<<bantu->Isi;
}
}
C:\Users\Roni Sefia\Documents\code\RoniSefia lat12_2.exe
1. Stack Menggunakan Array
2. Stack Terbalik
3. Stack Menggunakan Singly Linked List
Masukkan Pilihan: 3
Operasi Single Linked List Pada Stack
Penyiapan Stack
Masukan Elemen : 3
Masukan Elemen : 4
Masukan Elemen : 6
Masukan Elemen : 1
Masukan Elemen : 3
Masukan Elemen : 7
Isi Linked List : 3->4->6->1->3->7
Hapus Elemen
Isi Linked List : 3->4->6->1->3
Hapus Elemen
Isi Linked List : 3->4->6->1
Hapus Elemen
Isi Linked List : 3->4->6
Hapus Elemen
Isi Linked List : 3->4
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
31°C Hujan
14:27
13/06/2022

TUGAS AKHIR


C:\Users\Roni Sefia\Documents\code\Tugas Akhir 12.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
RoniSefia lat12_1.cpp
Tugas Akhir 12.cpp
#include<iostream>
#include<conio.h>
using namespace std;
main()
{
int i,j,len,flag=1;
char a[20];
cout<<"Masukkan Kalimat : ";
cin.getline(a,20);
for(len=0;a[len]!='\0';++len);
for(i=0,j=len-1;i<len/2;++i,--j)
{
if(a[j]!=a[i])
flag=0;
}
if(flag==1)
cout<<"\nKalimat Ini adalah Kalimat Polindrome";
else
cout<<"\nKalimat Ini Bukan Kalimat Polindrome";
getch();
}
C:\Users\Roni Sefia\Documents\code\Tugas Akhir 12.exe
Masukkan Kalimat : KASUR NABABAN RUSAK
Kalimat Ini adalah Kalimat Polindrome
Process exited after 9.764 seconds with return value 0
Press any key to continue . . .
Compiler
Resources
Compile Log
Debug
Find Results
Close
Abort Compilation
31°C Hujan
14:05
13/06/2022

LAPORAN AWAL

PRAKTIKUM
STRUKTUR DATA

LAPORAN KE - 9

Disusun Oleh :

Nama : Roni Sefia
NIM : 201011401617
Kelas : 04TPLP016

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG
Jl. Surya Kencana No.1, Pamulang Bar., Kec. Pamulang, Kota Tangerang Selatan, Banten 15417

LAPORAN AKHIR

PRAKTIKUM
STRUKTUR DATA

LAPORAN KE - 8

Disusun Oleh :

Nama : Roni Sefia
NIM : 201011401617
Kelas : 04TPLP016

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG
Jl. Surya Kencana No.1, Pamulang Bar., Kec. Pamulang, Kota Tangerang Selatan, Banten 15417