**PERTEMUAN XIII**

**QUEUE**

**TUJUAN PRAKTIKUM**

1. Mahasiswa dapat menjelaskan pengerian dan pembuatan Queue dengan C++.
2. Mahasiswa dapat melakukan operasi penyisipan dan penghapusan elemen dalam queue dengan C++.
3. Mahasiswa dapat mengimplementasikan Queue dengan C++.

**TEORI DASAR**

1. **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.

1. **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();

     }

}

1. Operasi-operasi pada Queue:
2. **Create()**

Untuk menciptakan dan menginisialisasi Queue

Dengan cara membuat Head dan Tail  = -1.

1. **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.

1. **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.

1. **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.

1. **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.

1. **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.

1. **Tampil()**

Untuk menampilkan nilai-nilai elemen Antrian

Menggunakan looping dari head s/d tail.

1. 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.

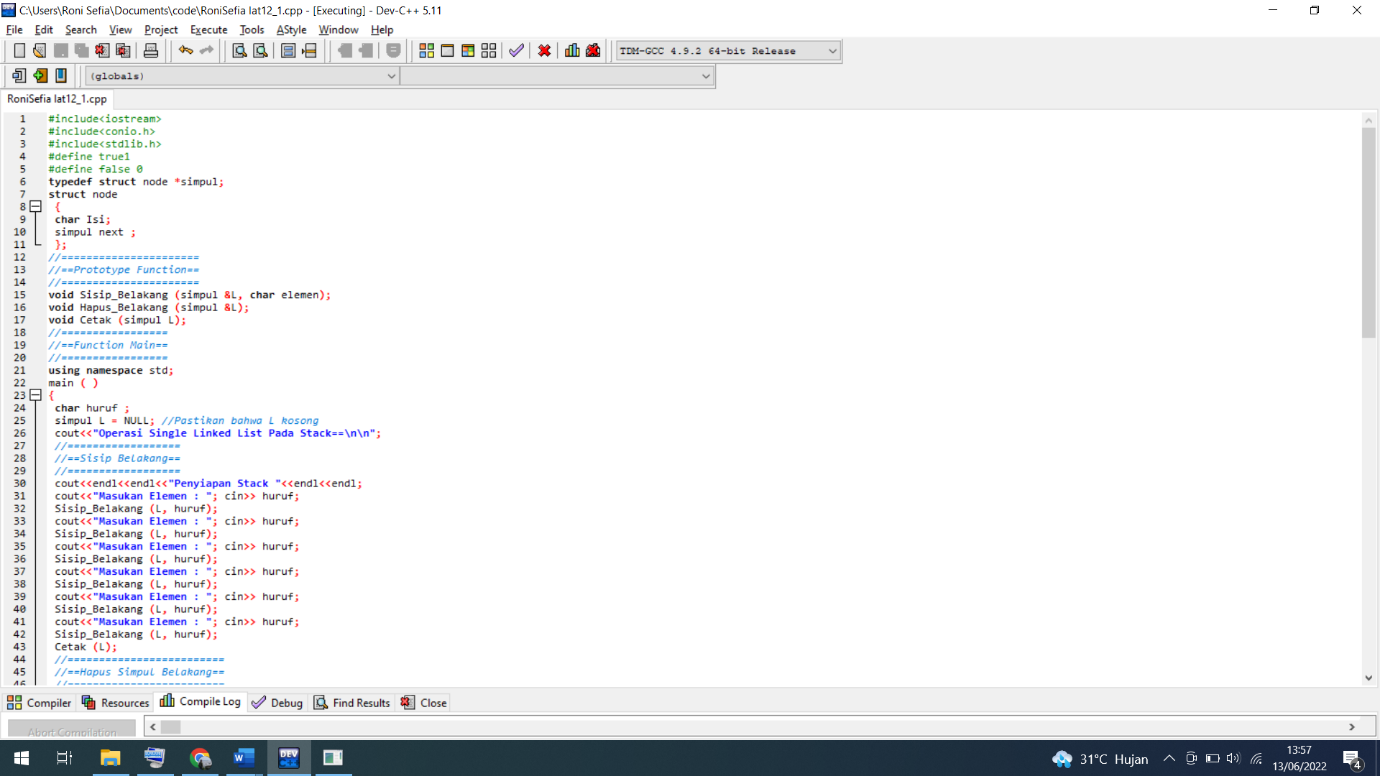
Contoh aplikasi Queue dalam dunia nyata :

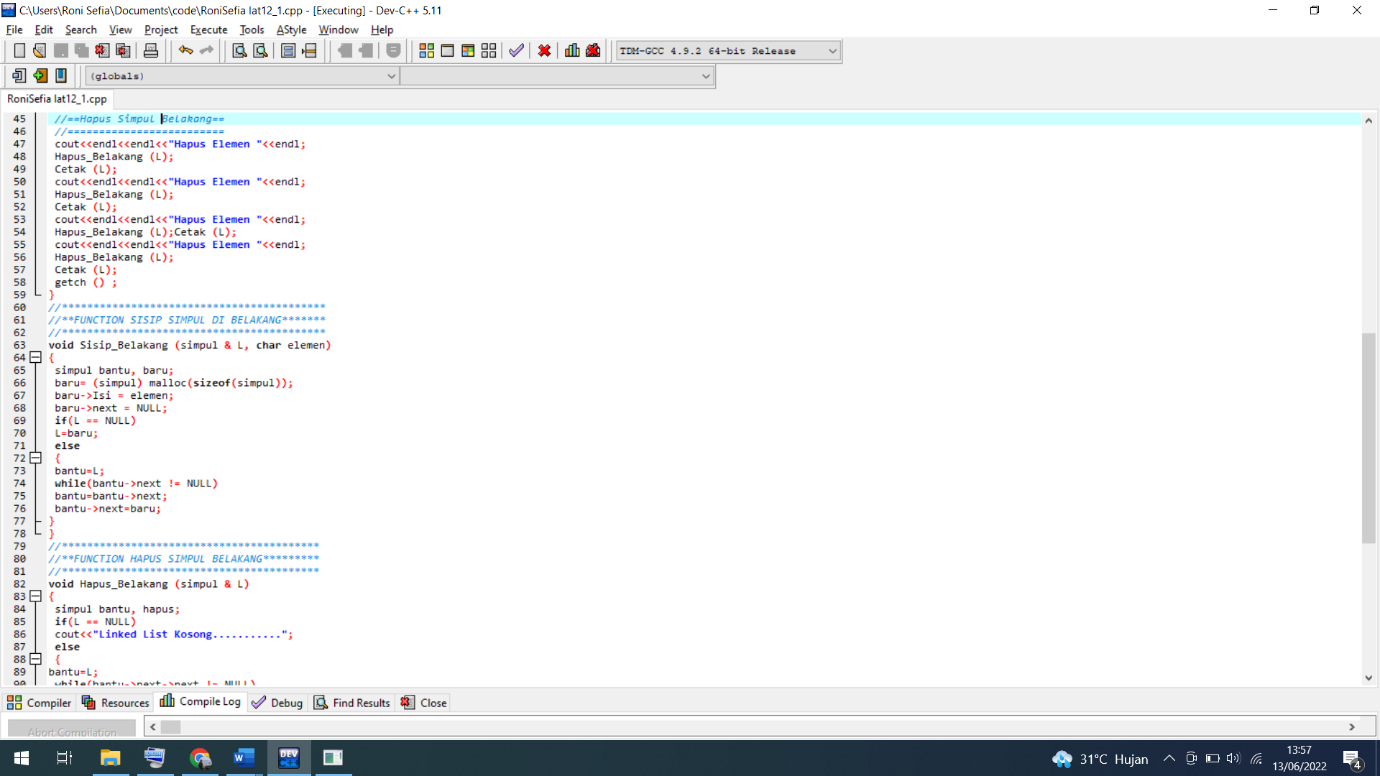
1. Aplikasi antrian di jalan Tol.
2. Aplikasi antrian saat mengantri di loket.
3. Aplikasi antraian reservasi tiket kereta api, dll.

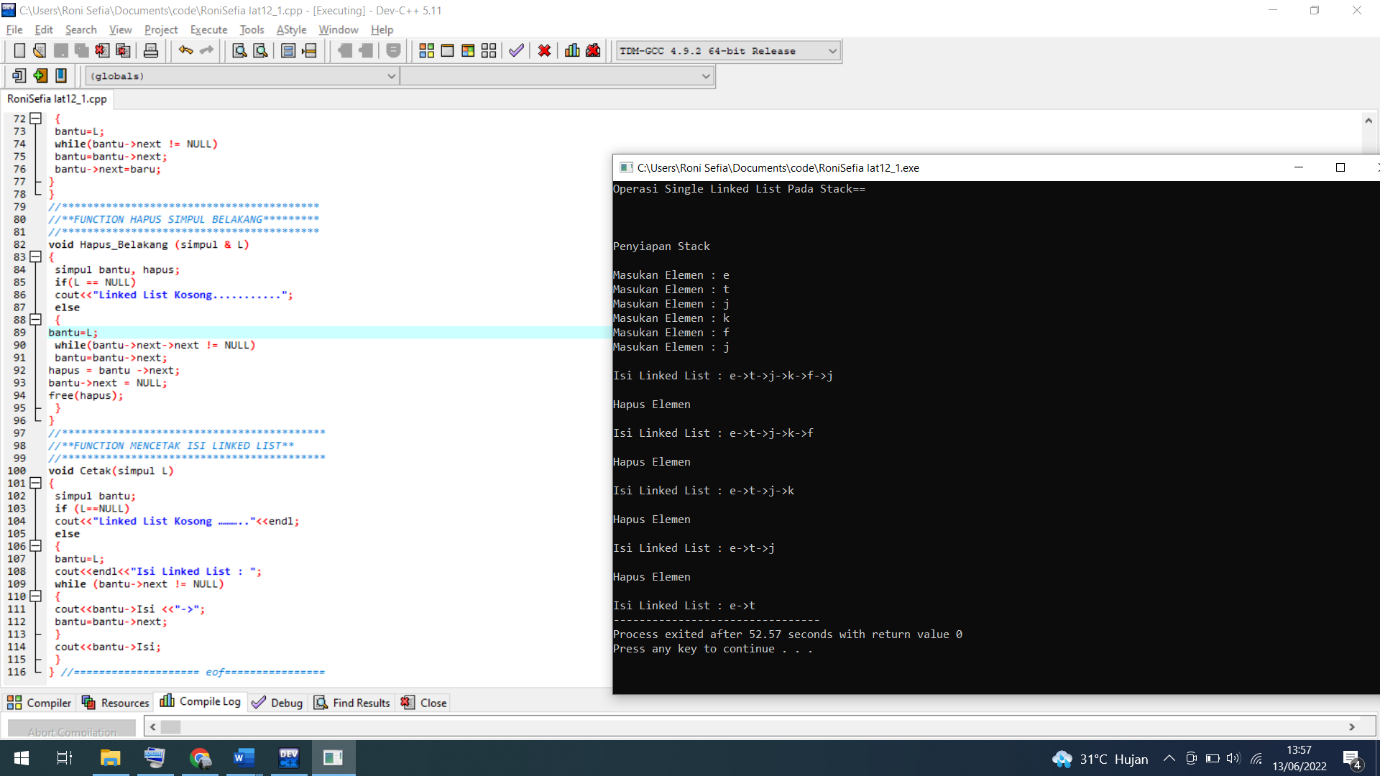
Semua itu menggunakan aturan FIFO (First In, First Out).

**TUGAS PRAKTIKUM**

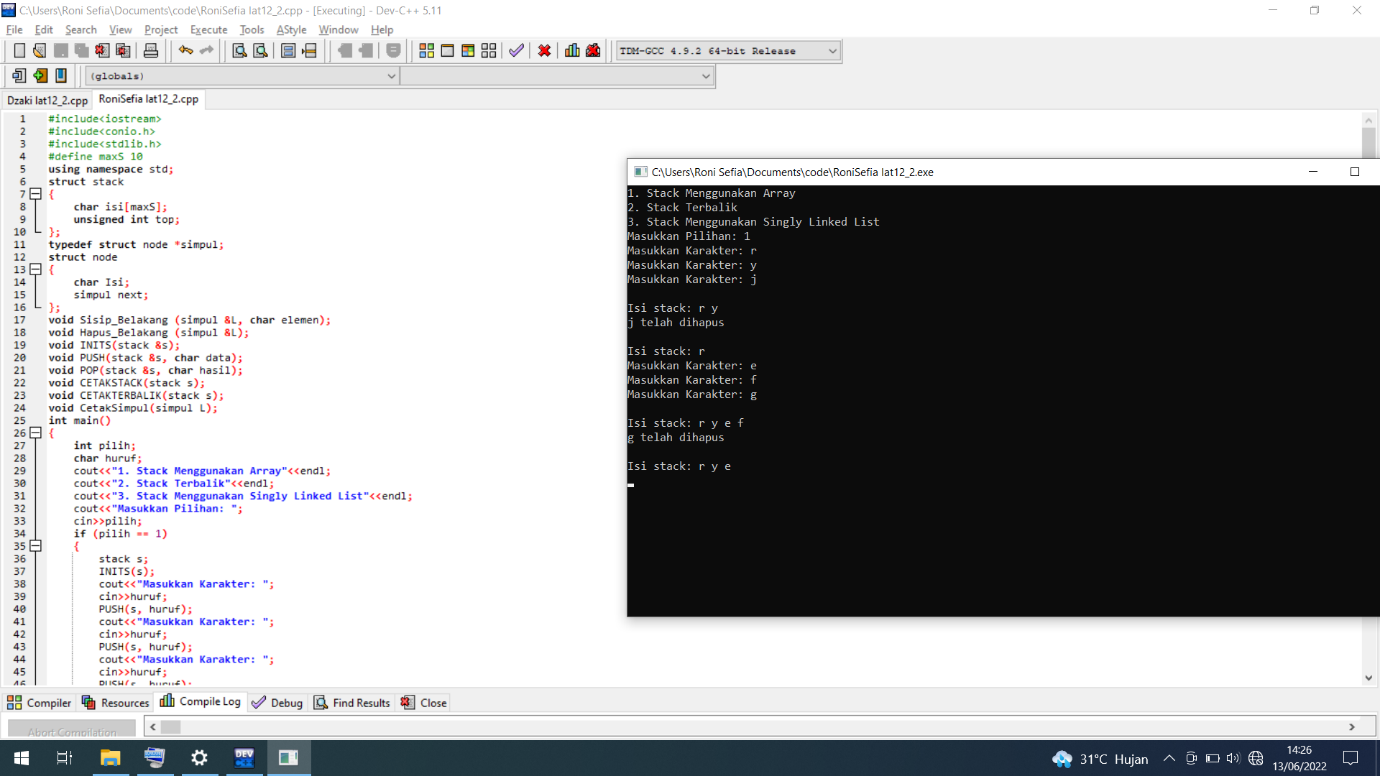
1. **Buatlah program Stack dengan menggunakan Singly Linked List (simpan dengan nama lat12\_1.cpp)**

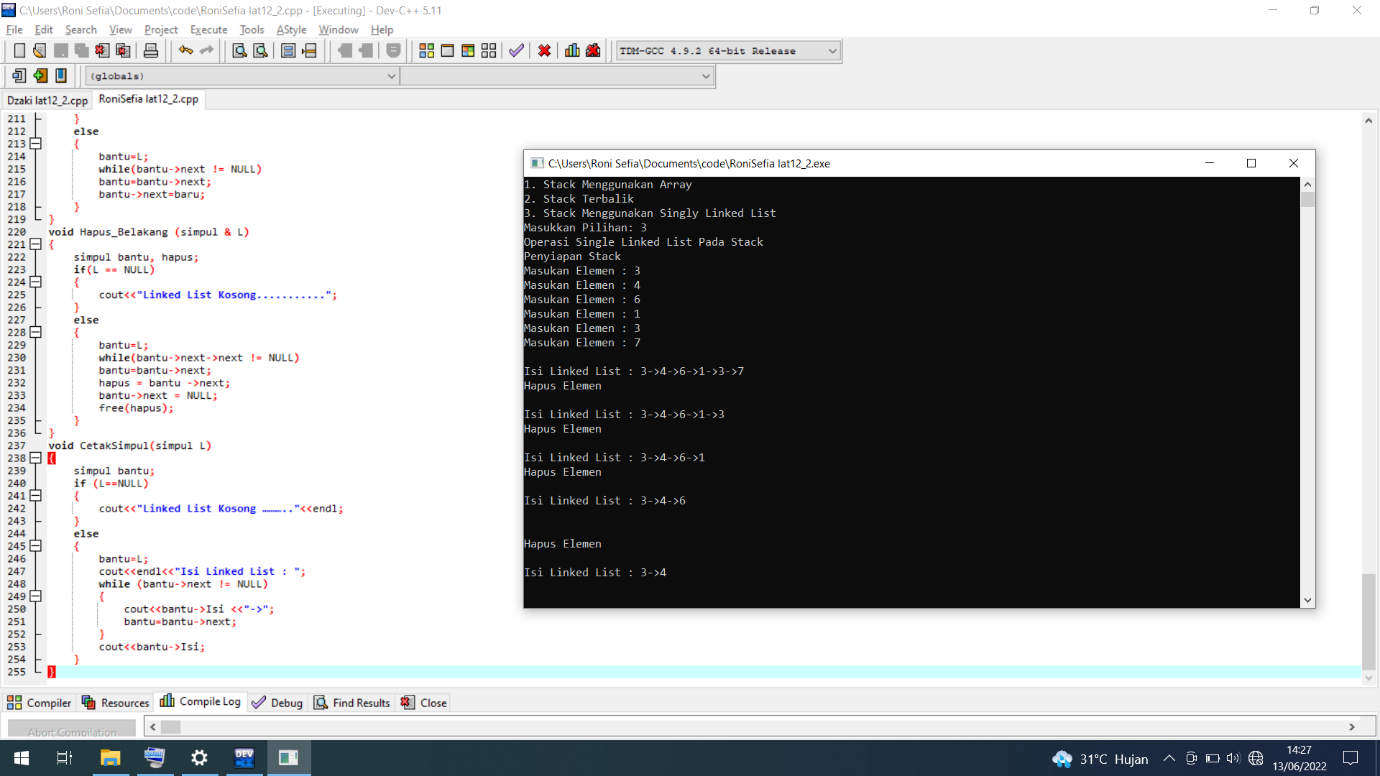
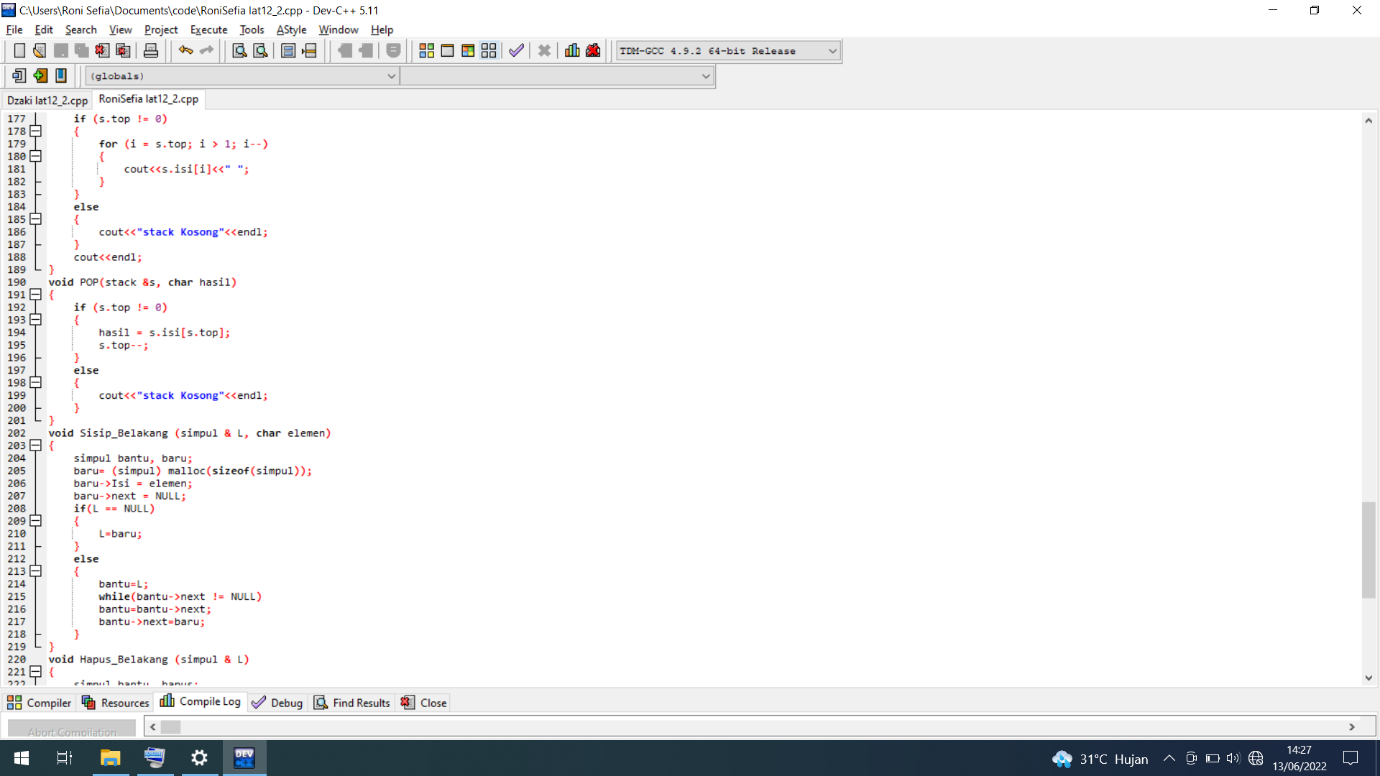
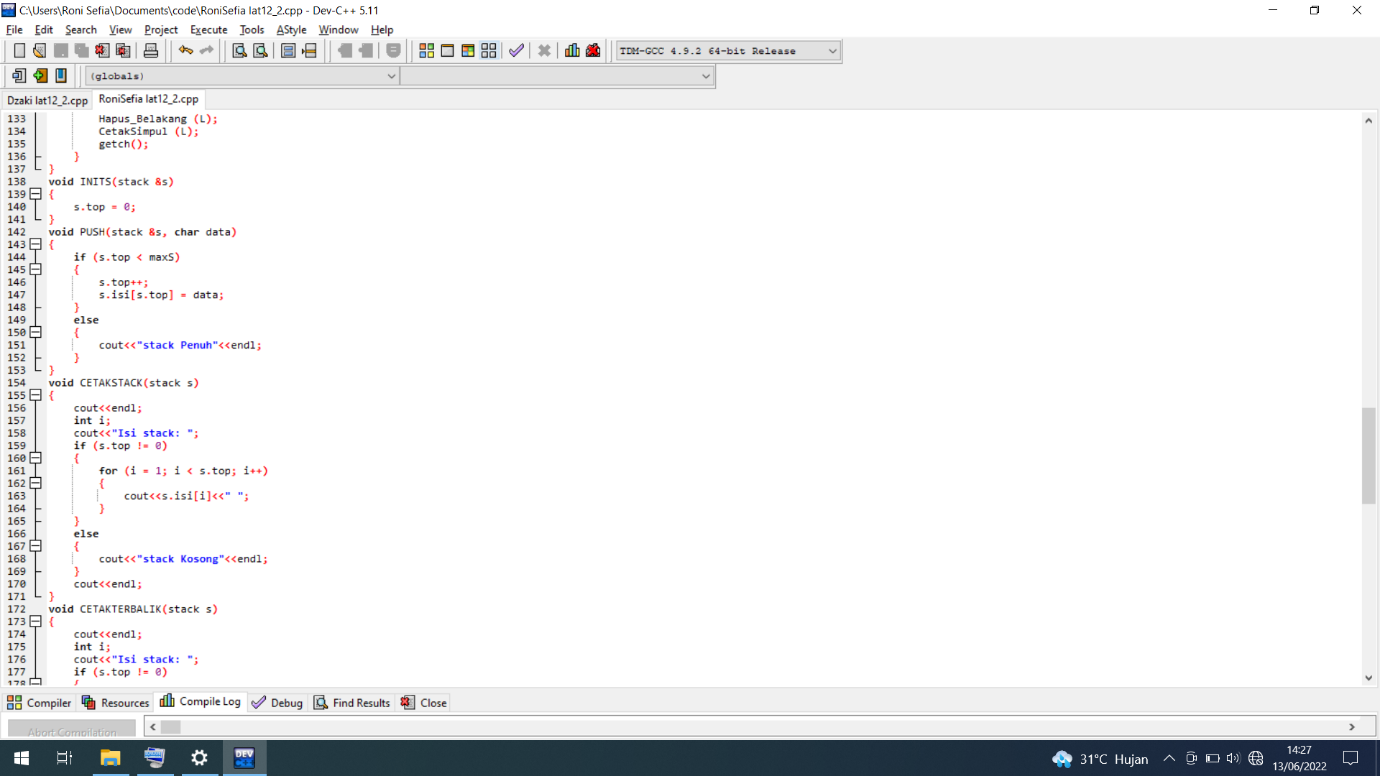
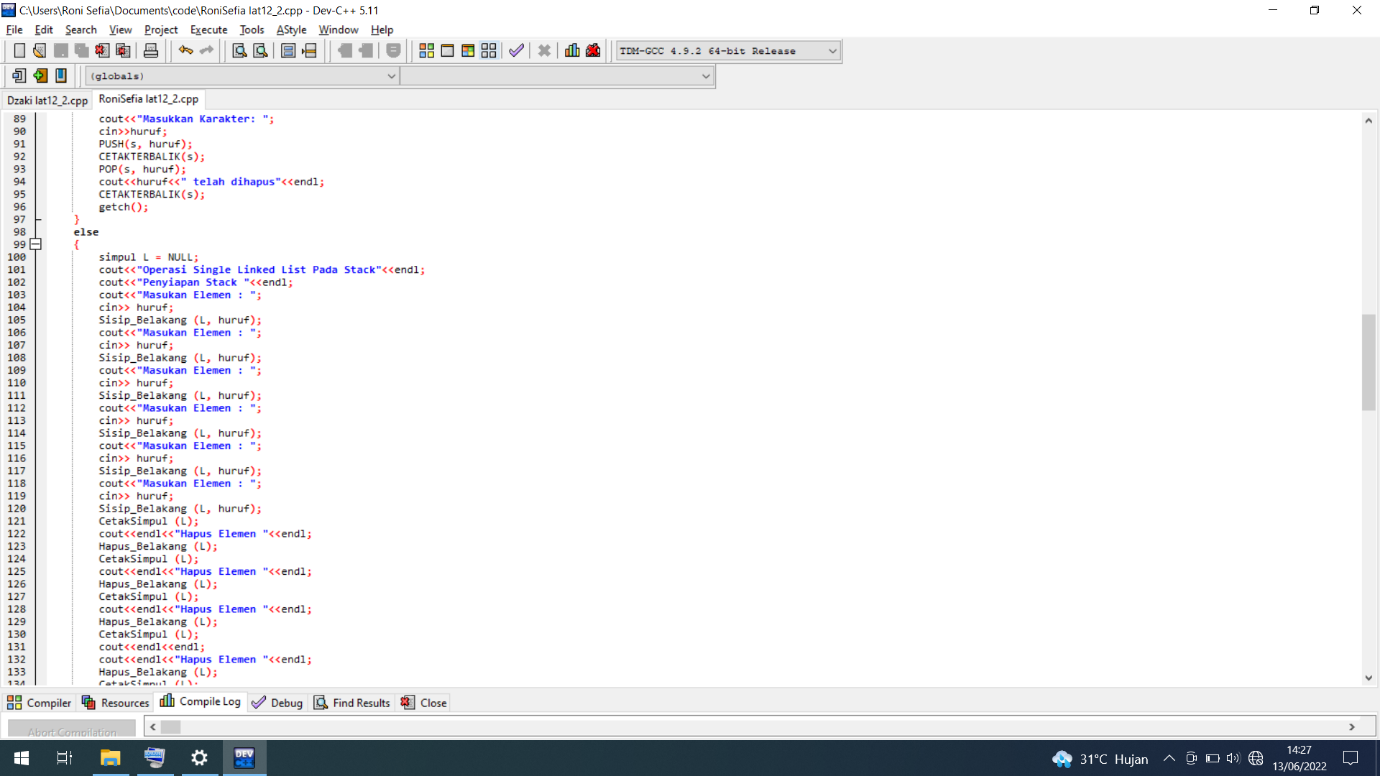
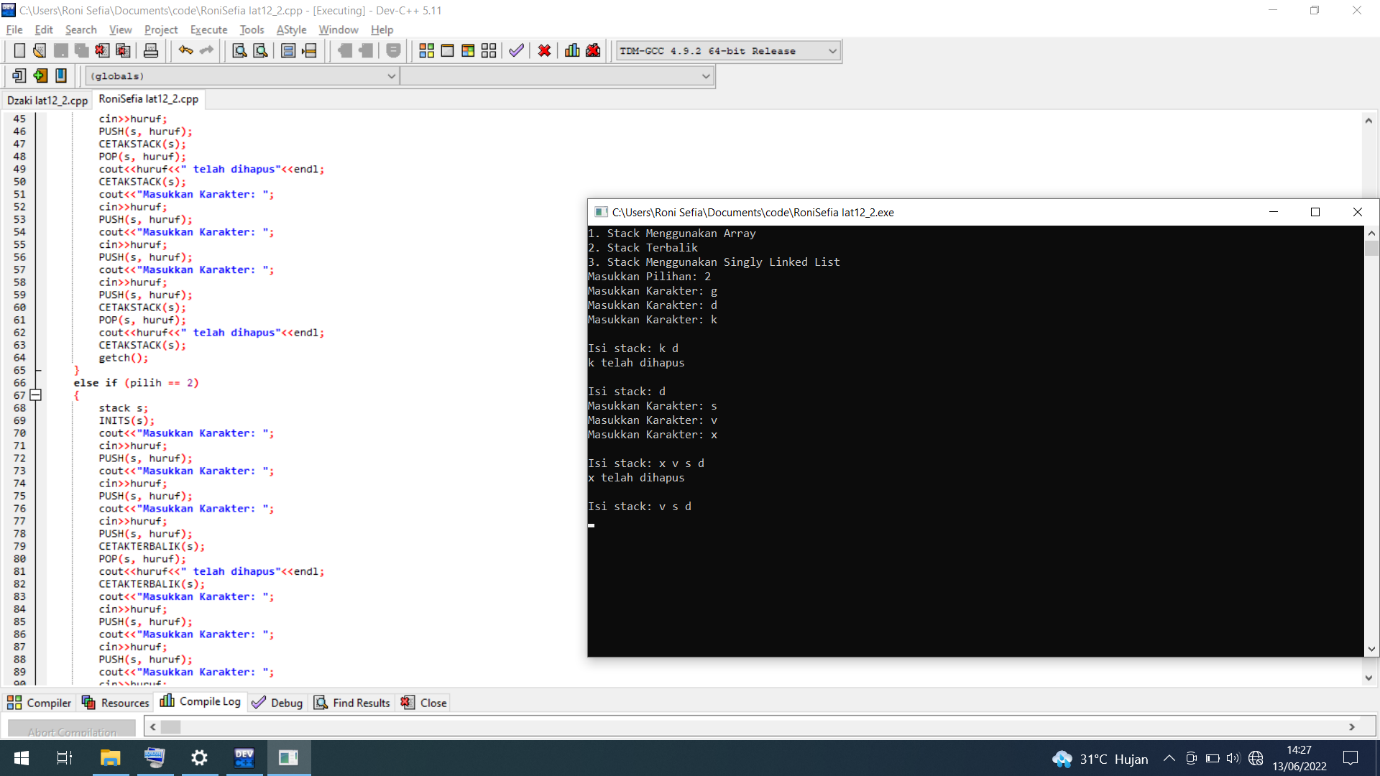




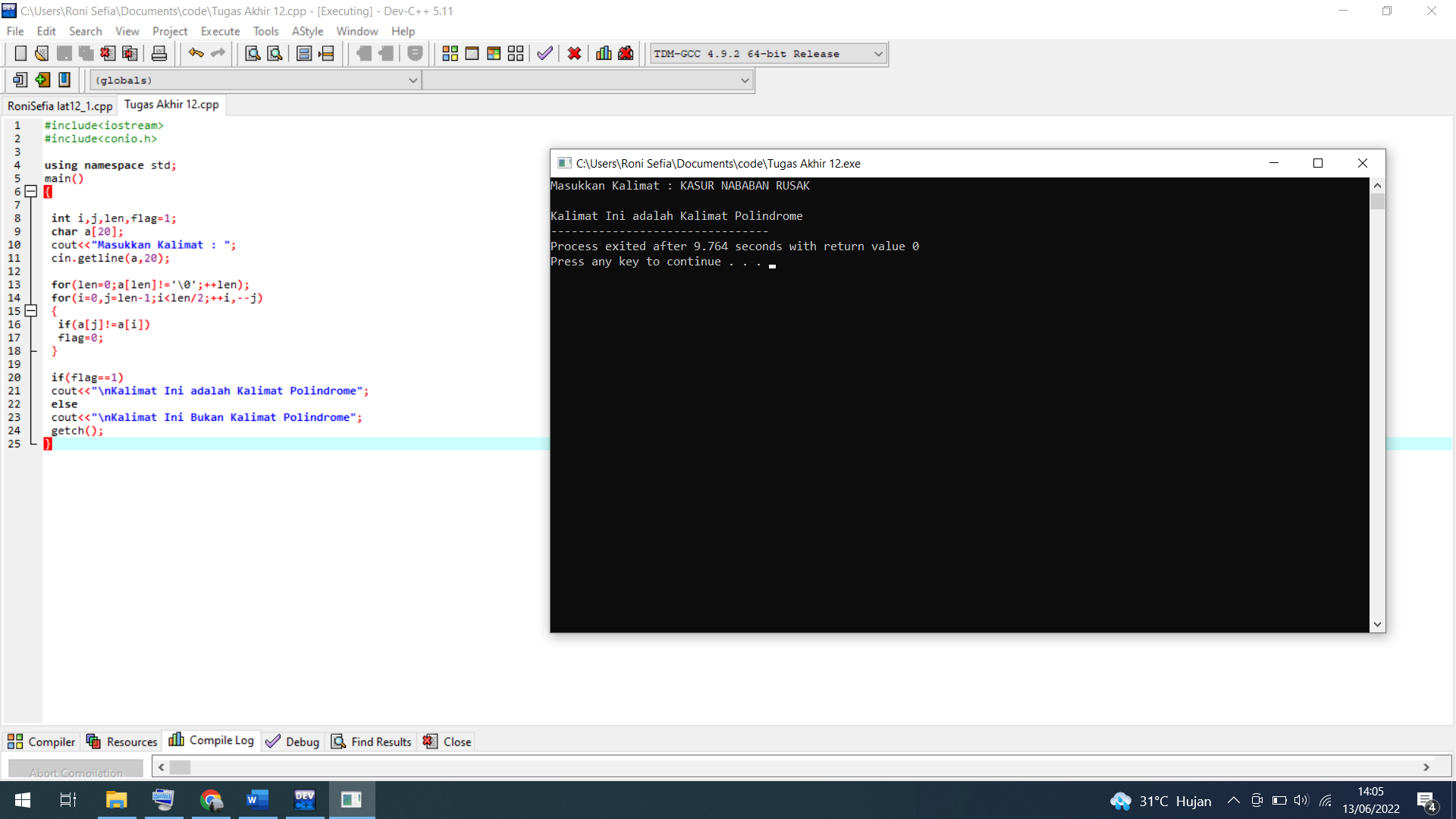


1. **Buatlah program menu untuk menampilkan program-program pada pertemuan XI dan XII, menggunakan perintah IF (simpan dengan nama lat12\_2.cpp)**



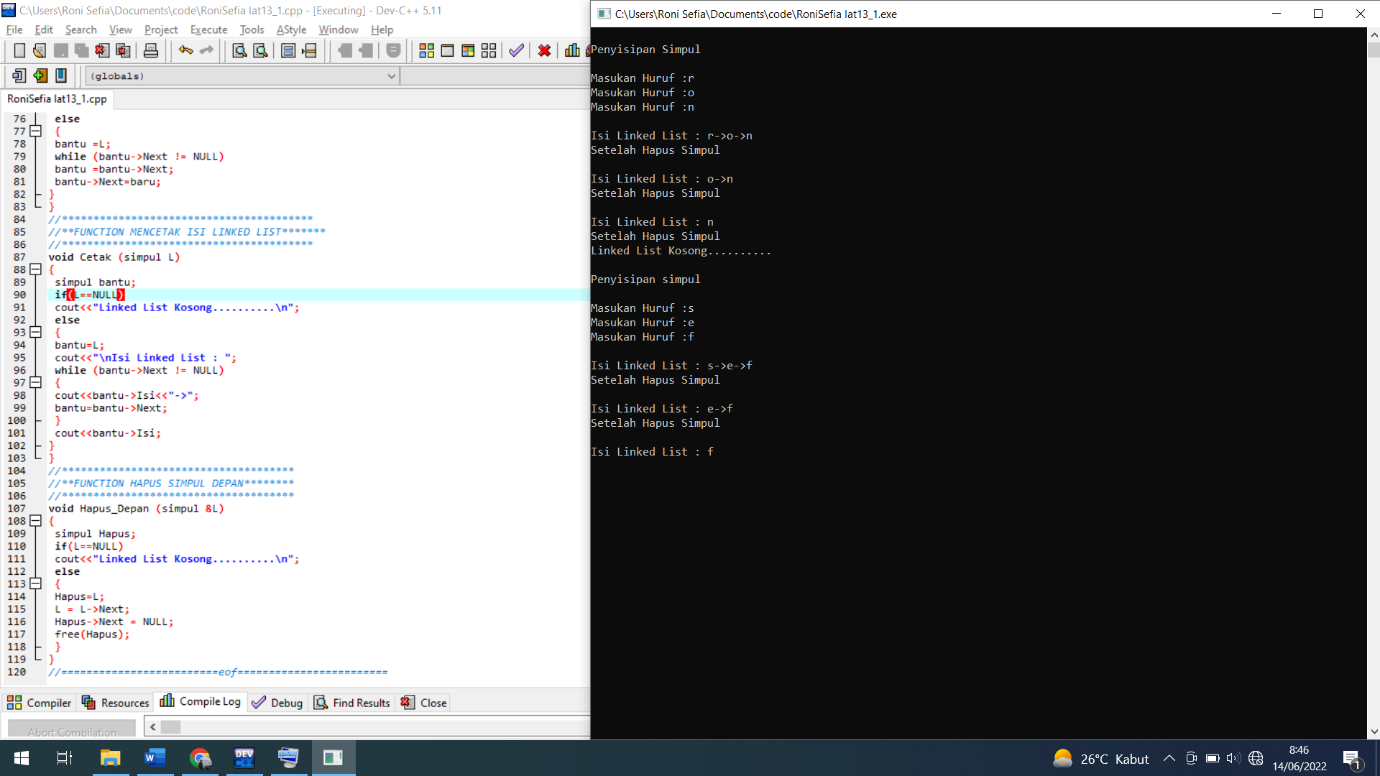
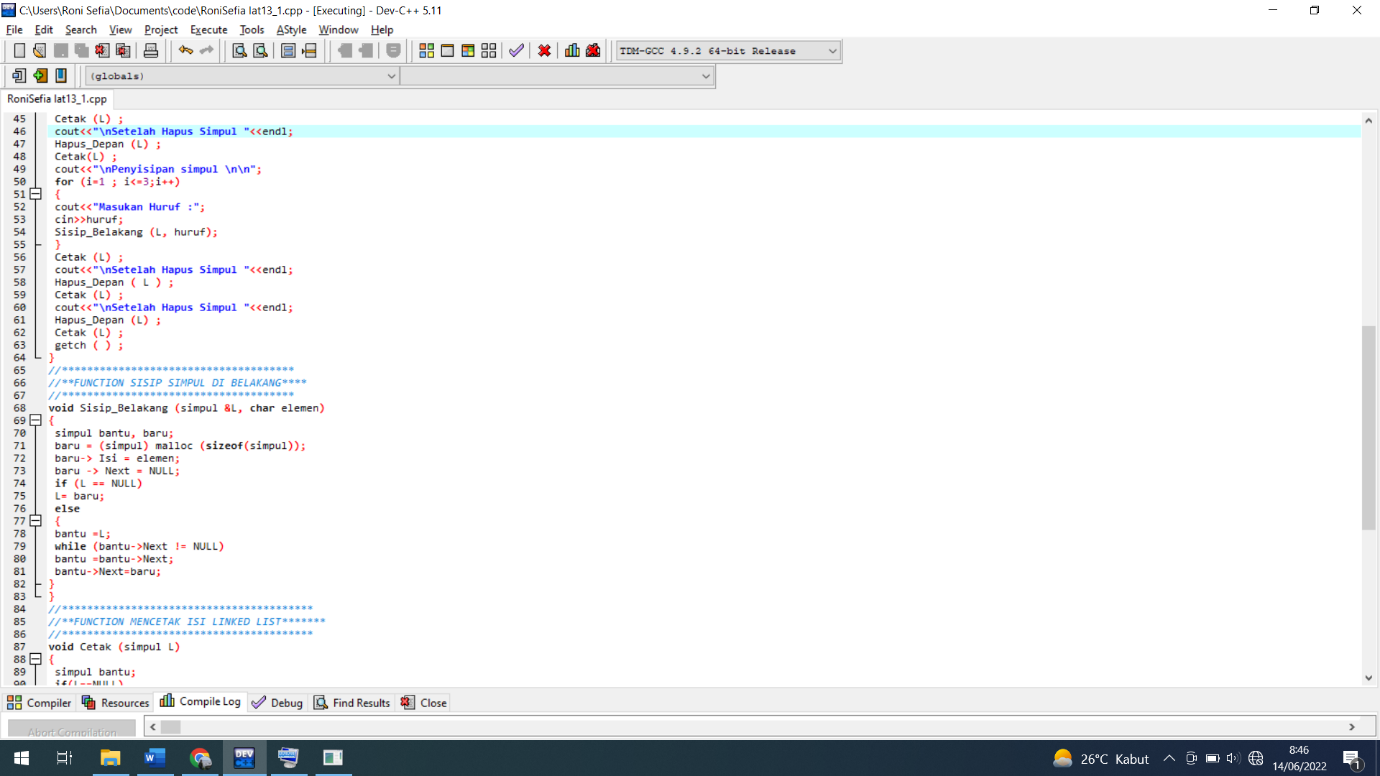
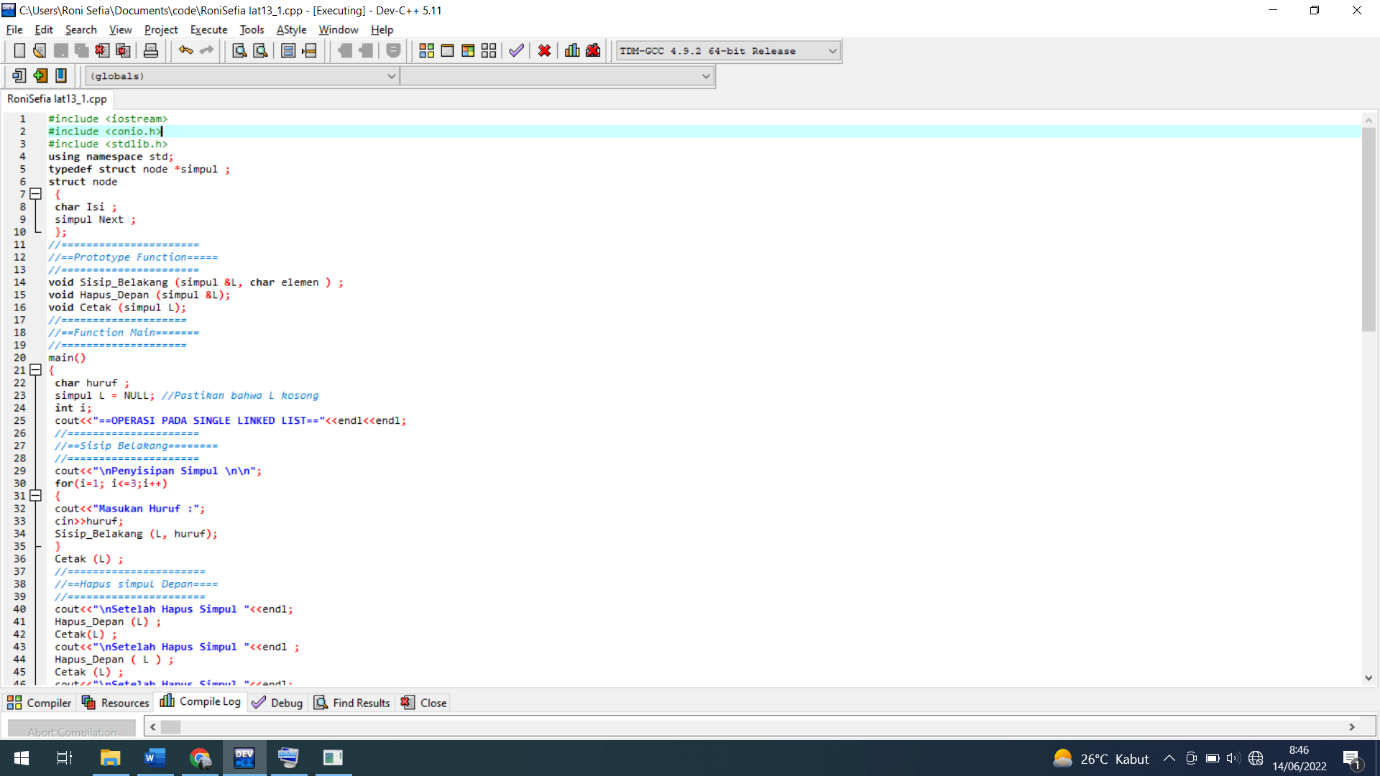


**TUGAS AKHIR**

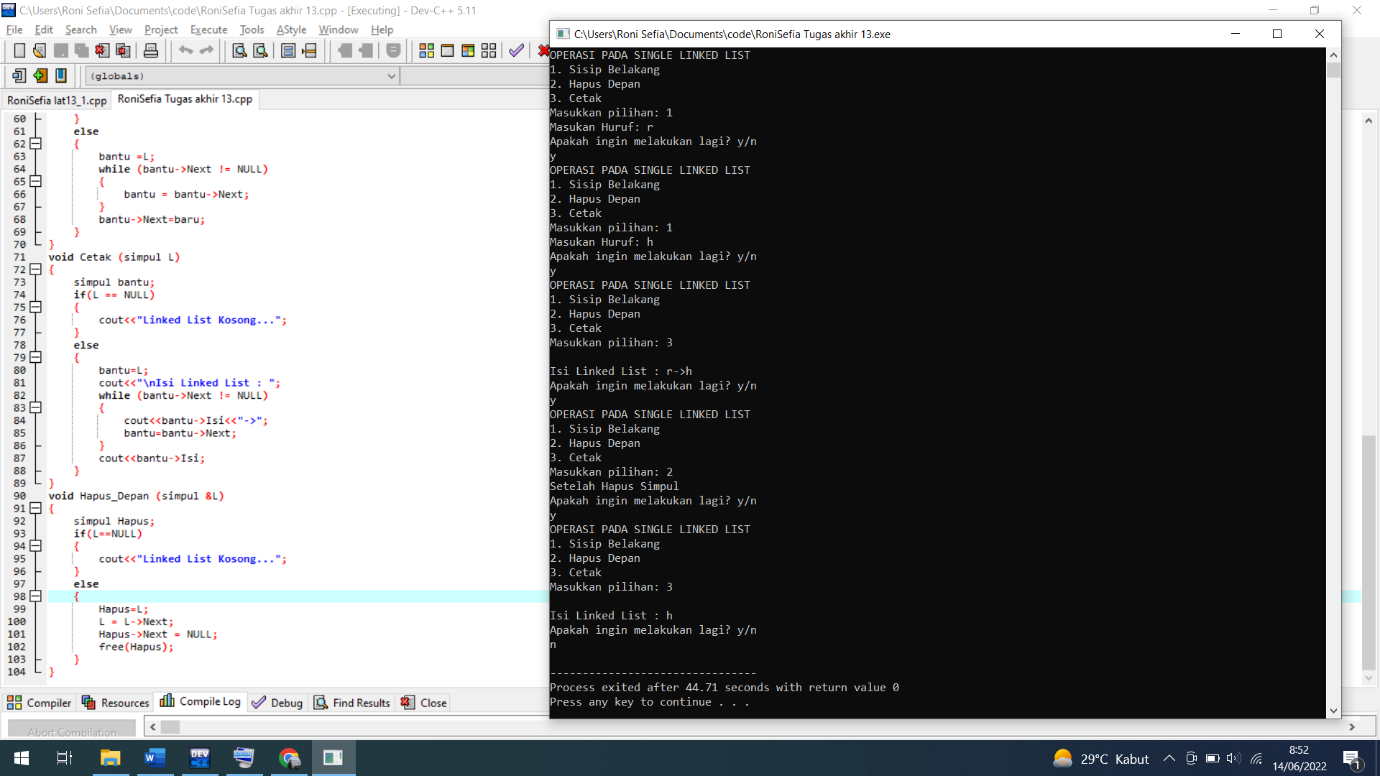
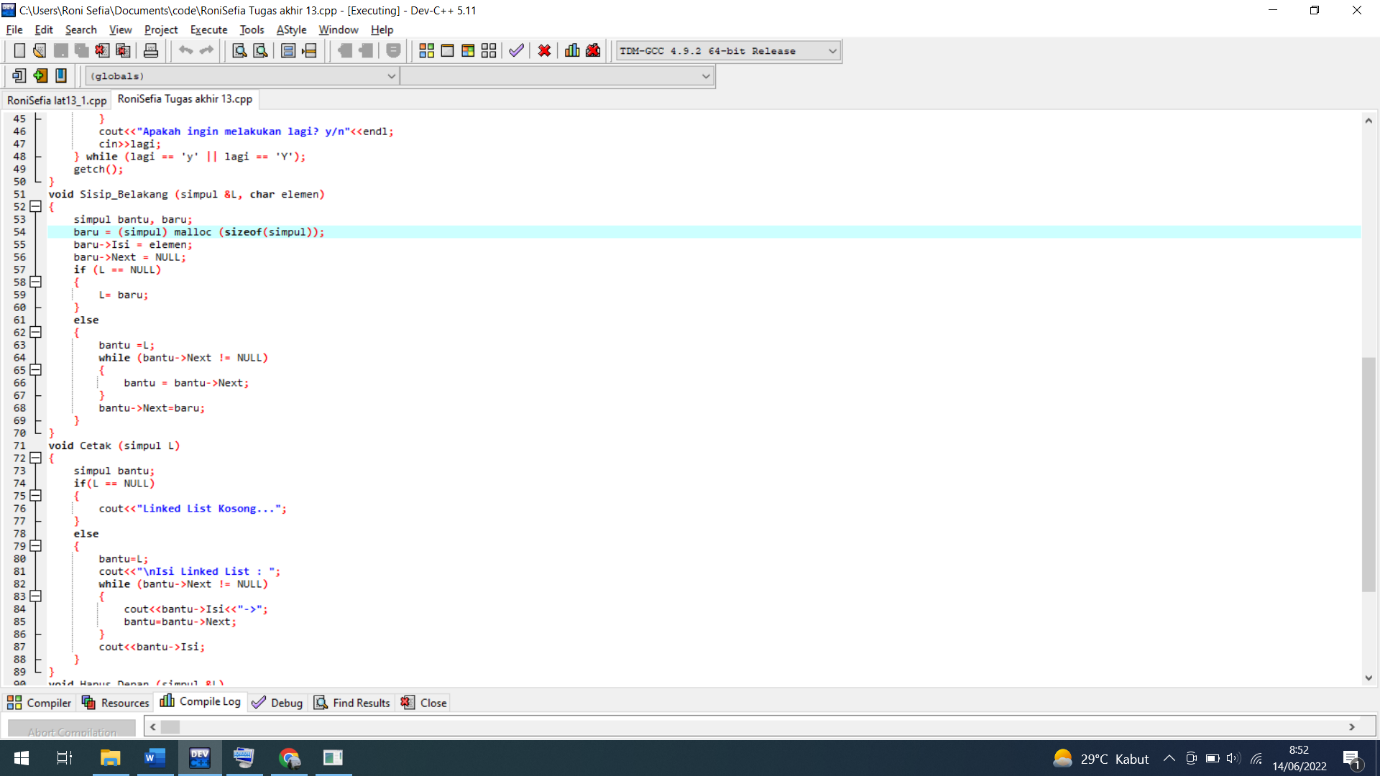
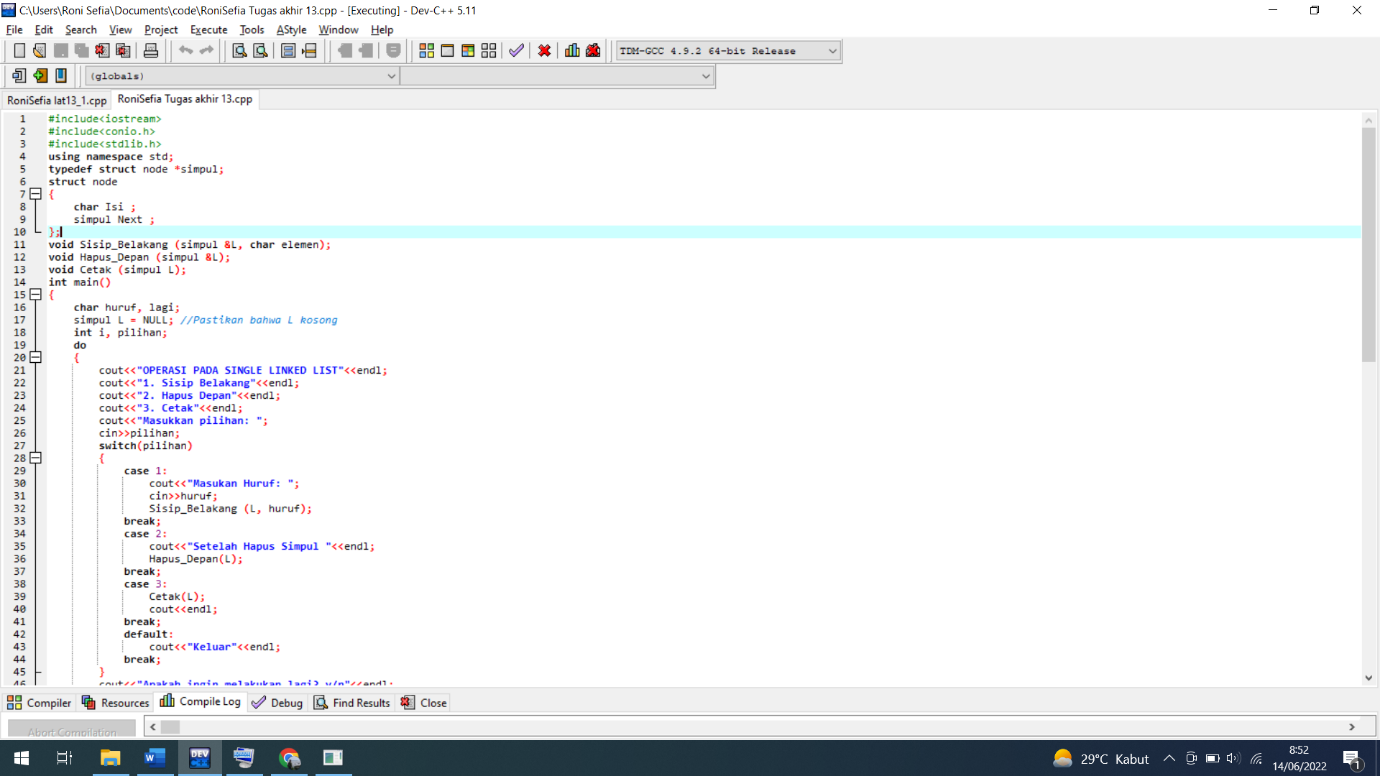


**TUGAS PRAKTIKUM**

1. **Buatlah program untuk operasi Queue dengan menggunakan Singly Linked List (simpan dengan nama lat13\_1.cpp)**



**TUGAS AKHIR**



LAPORAN AWAL

PRAKTIKUM

STRUKTUR DATA

**LAPORAN KE - 12**

****

**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 - 11**

****

**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 - 12**

****

**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**