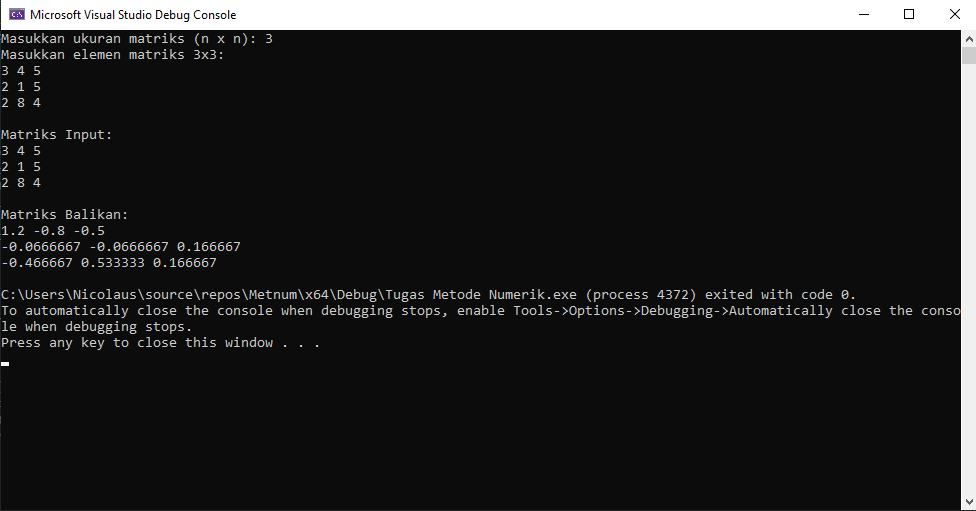
Nama : Nicolaus Evan Widyatna

NIM : 21120122140140

Kelas : D

METODE NUMERIK (penjelasan alur kode)

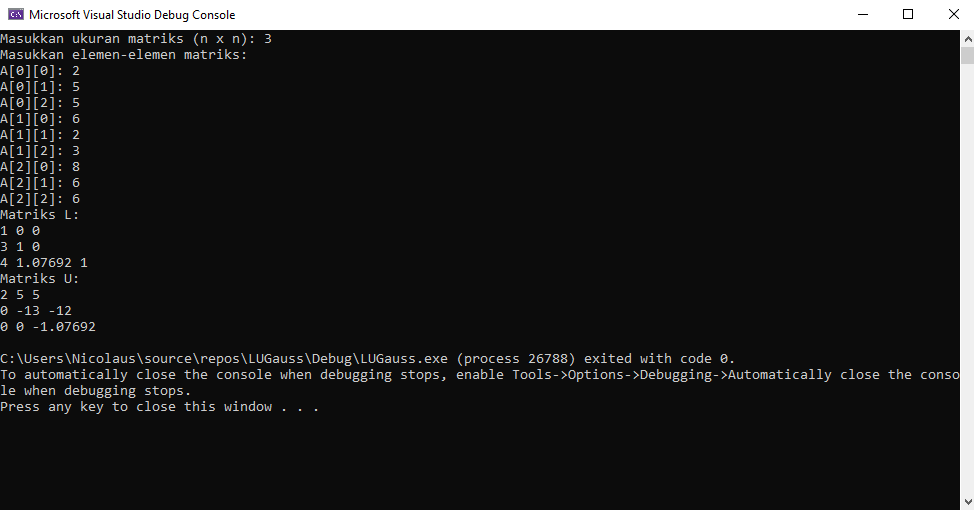
1. **Metode Matriks Balikan**  
   Kode ini adalah sebuah program C++ yang digunakan untuk menghitung matriks balikan dari sebuah matriks input menggunakan metode eliminasi Gauss-Jordan. Berikut adalah penjelasan alur kode tersebut:
2. **Pendefinisian Fungsi-fungsi:**
   * **printMatrix**: Fungsi ini digunakan untuk mencetak matriks ke layar.
   * **createIdentityMatrix**: Fungsi ini digunakan untuk membuat matriks identitas dengan ukuran yang ditentukan.
   * **swapRows**: Fungsi ini digunakan untuk menukar dua baris dalam matriks.
   * **rowOperation**: Fungsi ini digunakan untuk melakukan operasi baris elementer pada matriks.
3. **Fungsi inverseMatrix (Menghitung Matriks Balikan):**
   * Fungsi ini menerima sebuah matriks sebagai argumen dan mengembalikan matriks balikan.
   * Pertama-tama, matriks input disalin ke dalam matriks sementara (**tempMatrix**) dan matriks identitas dibuat menggunakan **createIdentityMatrix**.
   * Kemudian, dilakukan eliminasi Gauss-Jordan untuk membuat matriks input menjadi matriks identitas.
   * Setiap iterasi dari eliminasi Gauss-Jordan, mencari pivot non-nol, menukar baris pivot dengan baris saat ini, dan membuat elemen diagonal menjadi 1 dengan membagi baris pivot dengan elemen pivot.
   * Selanjutnya, dilakukan operasi baris elementer untuk membuat elemen di kolom pivot (selain pivot) menjadi 0 dengan menggunakan faktor yang sesuai.
   * Proses ini dilakukan hingga semua elemen diagonal menjadi 1 dan semua elemen di luar diagonal menjadi 0.
   * Jika matriks input adalah singular (tidak dapat dibalik), program akan mencetak pesan kesalahan dan mengembalikan matriks kosong.
4. **Fungsi main:**



Gambar 1.1 kode testing matriks balikan

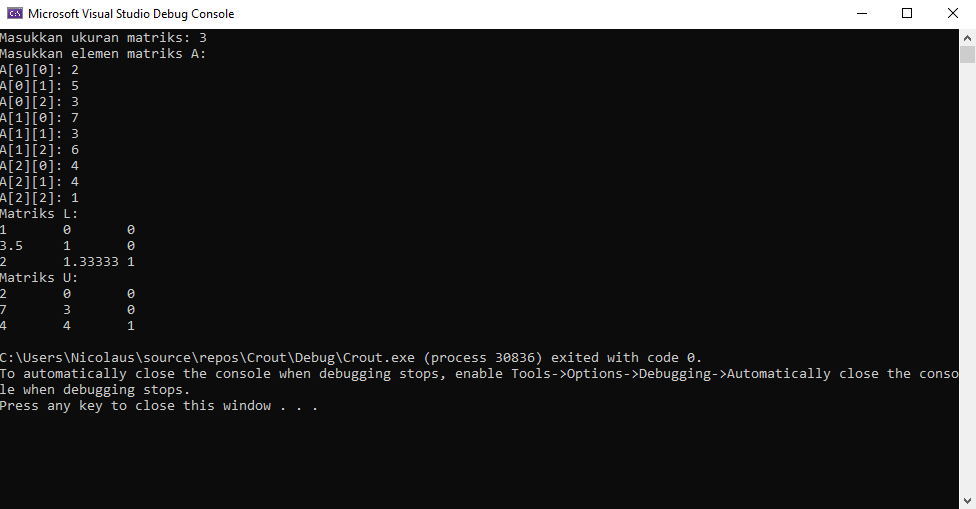
* + Program utama mulai dengan meminta pengguna untuk memasukkan ukuran matriks.
  + Kemudian, membaca elemen-elemen matriks dari pengguna.
  + Mencetak matriks input.
  + Memanggil fungsi **inverseMatrix** untuk menghitung matriks balikan dari matriks input.
  + Jika matriks balikan berhasil dihitung, mencetak hasil dengan metode matriks balikan tersebut.

1. **Metode Dekomposisi LU Gauss**

****

Gambar 1.2 kode testing matriks Dekomposisi LU Gauss

Kode ini merupakan implementasi dalam C++ dari algoritma dekomposisi LU menggunakan metode Gauss :

1. Kode dimulai dengan inklusi beberapa pustaka standar C++, yaitu **<iostream>** dan **<vector>**. Ini diperlukan untuk input-output dan penggunaan struktur data vektor.
2. Fungsi **gaussianLUDecomposition** digunakan untuk melakukan dekomposisi LU dari matriks input **A**. Ini akan menghasilkan dua matriks, **L** (lower triangular) dan **U** (upper triangular), yang merupakan hasil dari dekomposisi. Ini mengikuti algoritma Gauss yang membagi matriks **A** menjadi matriks segitiga atas **U** dan matriks segitiga bawah **L**.
3. Di dalam **main()**, pengguna diminta untuk memasukkan ukuran matriks dan elemen-elemen matriksnya.
4. Setelah masukan dimasukkan, fungsi **gaussianLUDecomposition** dipanggil dengan matriks input **A** dan dua matriks kosong **L** dan **U** sebagai argumen. Ini akan mengisi **L** dan **U** dengan hasil dekomposisi.
5. Setelah dekomposisi selesai, hasilnya dicetak ke layar. Pertama-tama, matriks **L** ditampilkan, diikuti oleh matriks **U**.
6. **Metode Dekomposisi Crout**

Gambar 1.3 kode testing matriks Dekomposisi Crout

1. Kode dimulai dengan inklusi pustaka **<iostream>** dan **<vector>** untuk mengakses fungsi-fungsi yang digunakan dalam program. Penggunaan **using namespace std;** memungkinkan penggunaan fungsi-fungsi standar dari namespace **std** tanpa menyertakan awalan **std::**.
2. **Fungsi printMatrix**: Fungsi ini digunakan untuk mencetak matriks ke layar. Menerima matriks sebagai parameter dan mengeluarkan elemen-elemen matriks secara berbaris.
3. **Fungsi croutDecomposition**: Fungsi ini melakukan dekomposisi Crout dari matriks masukan A menjadi matriks L (lower triangular) dan U (upper triangular). Dekomposisi dilakukan dengan menggunakan algoritma Crout. Fungsi ini menerima tiga parameter: matriks masukan A, dan matriks L dan U yang akan diisi dengan hasil dekomposisi.
4. **Fungsi main**: Fungsi utama dari program. Di dalamnya, terdapat langkah-langkah berikut:
   * Menerima input ukuran matriks dari pengguna.
   * Membuat matriks masukan A dengan ukuran yang telah ditentukan, dan meminta pengguna untuk memasukkan elemen-elemennya.
   * Membuat matriks L dan U yang akan diisi dengan hasil dekomposisi Crout.
   * Memanggil fungsi **croutDecomposition** untuk melakukan dekomposisi Crout.
   * Mencetak matriks L dan U ke layar.
5. **Input dan Output**: Program meminta pengguna untuk memasukkan ukuran matriks dan elemen-elemen matriks. Setelah itu, mencetak matriks L dan U hasil dekomposisi ke layar.
6. **Dekomposisi Crout**: Dekomposisi Crout dilakukan dengan algoritma yang menghasilkan matriks L (lower triangular) dan U (upper triangular) dari matriks masukan A. Proses ini melibatkan penghitungan elemen-elemen matriks L dan U menggunakan rumus-rumus yang sesuai dengan algoritma Crout.