**LAPORAN PRAKTIKUM**

**METODE NUMERIK**

**Judul: Sistem Persamaan Linier**



**DISUSUN OLEH**

**ILHAM NUR ROMDONI M0520038**

**PROGRAM INFORMATIKA**

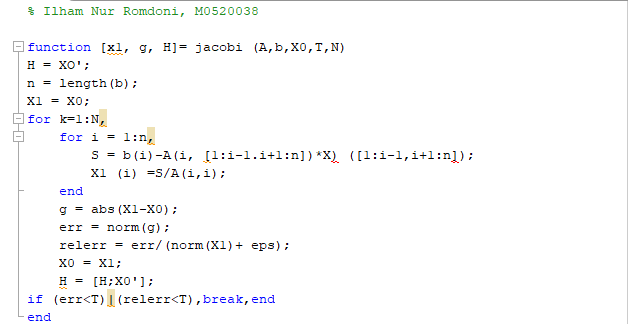
**FAKULTAS MIPA**

**UNIVERSITAS SEBELAS MARET**

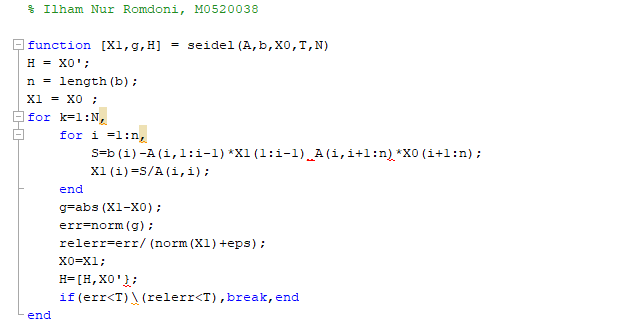
**2021**

**SCREENSHOT**

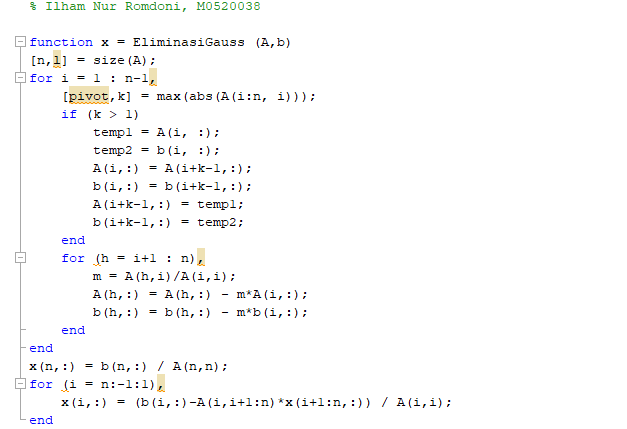
1. **Screenshot Praktikum**
2. **Metode Jacobi**

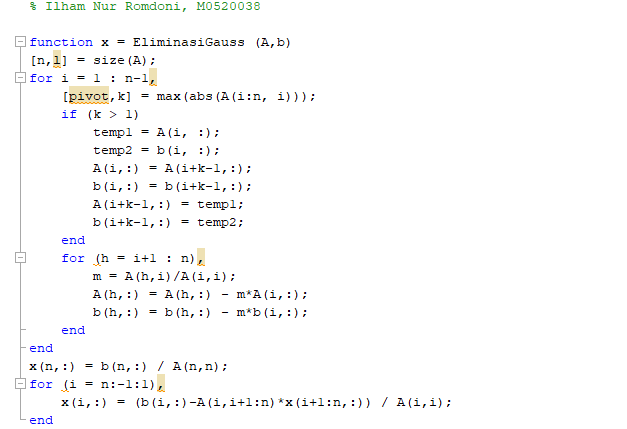
****

1. **Metode Gauss-Seidel**

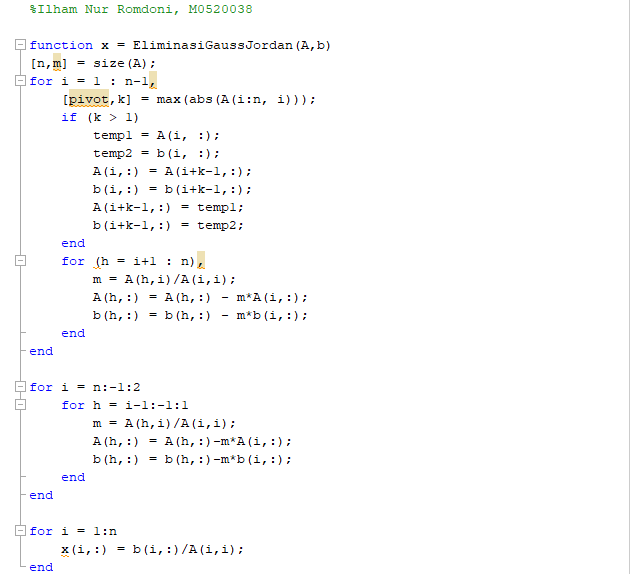
****

1. **Metode Eliminasi Gauss ( OBE )**

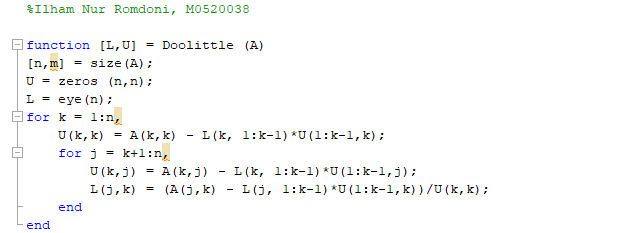
****

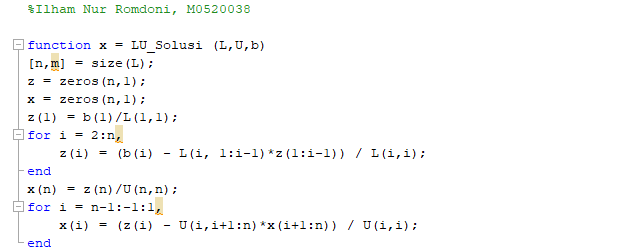
****

1. **Metode Eliminasi Gauss-Jordan**



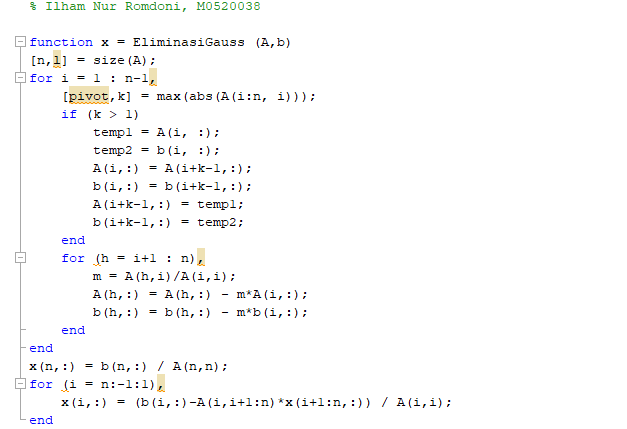
1. **Metode Dekomposisi LU**

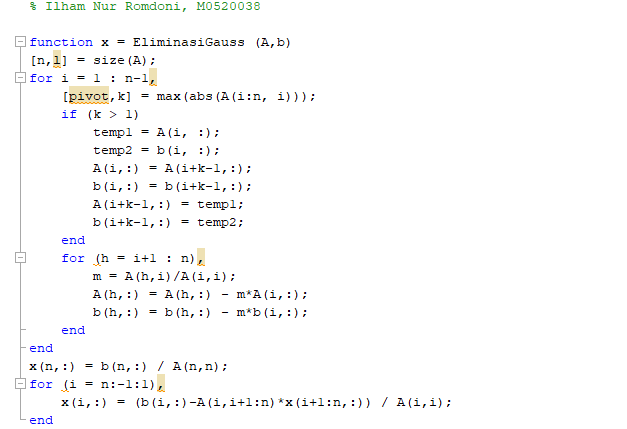
****

****

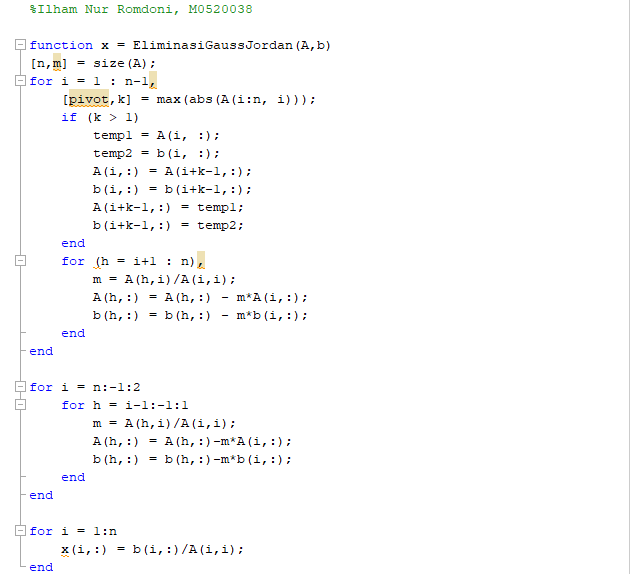
1. **Screenshot Source Code**
2. **Kasus 1**

**Metode Gauss**

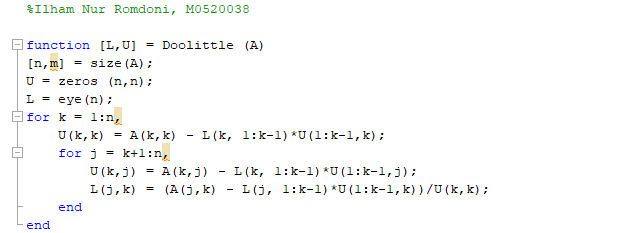
****

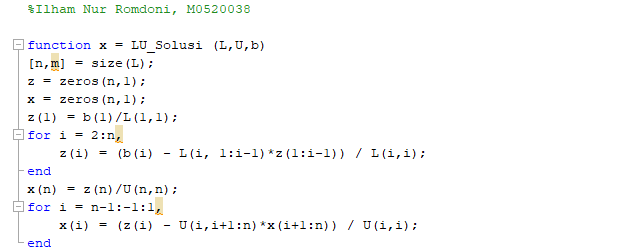
****

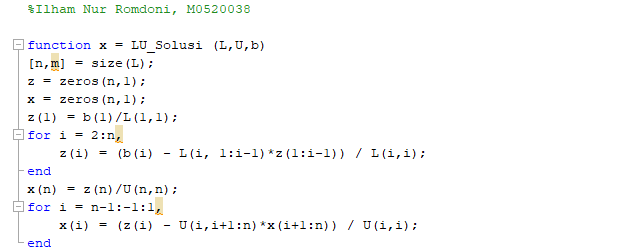
**Metode Gauss-Jordan**



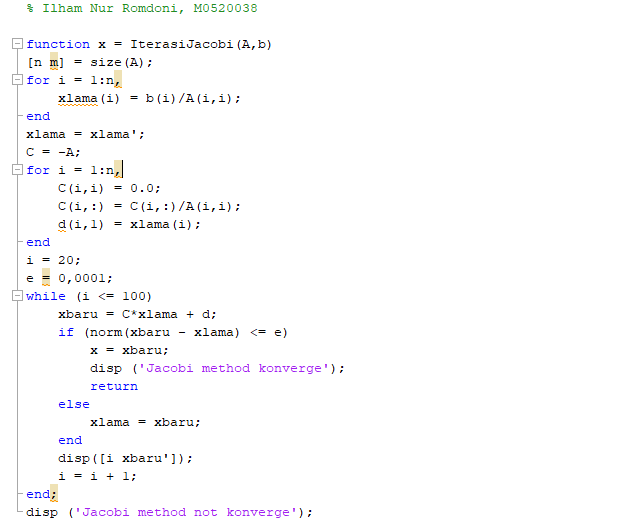
**Metode LU Dekomposisi**

****

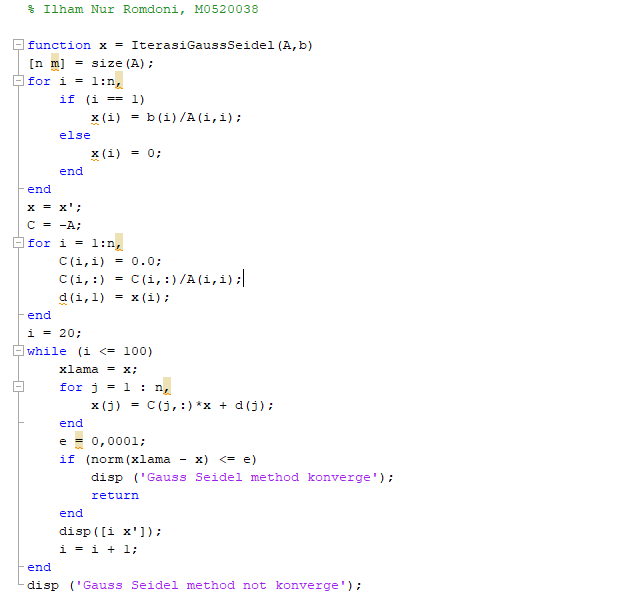
****

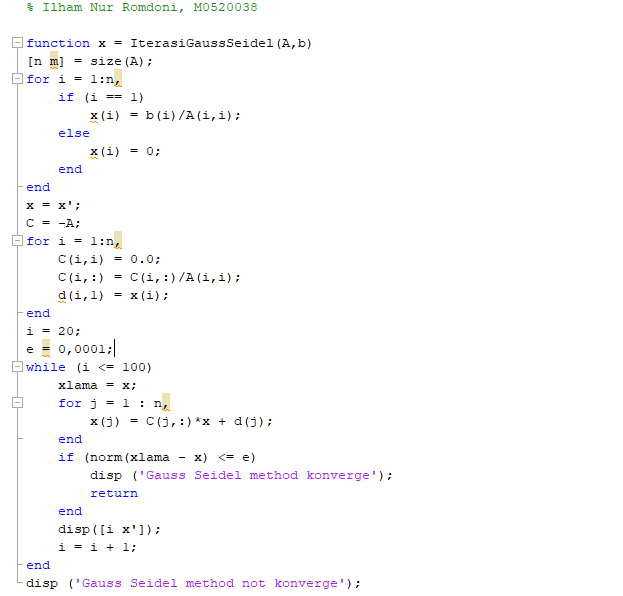
****

**Metode Jacobi**

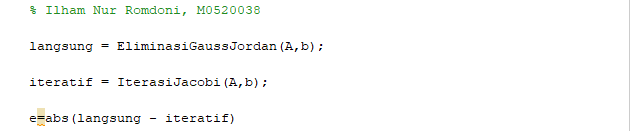
****

**Metode Gauss-Seidel**

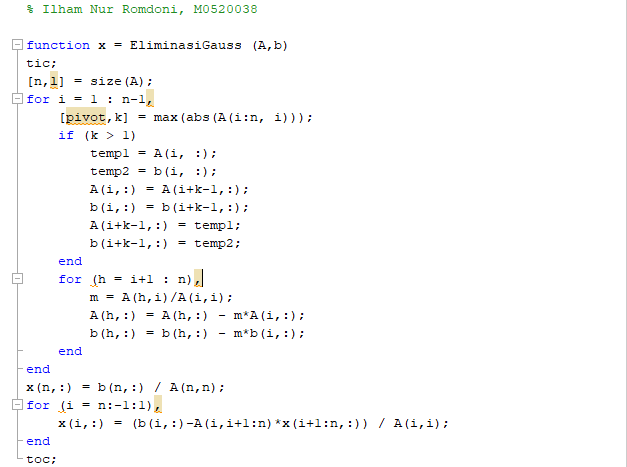
****

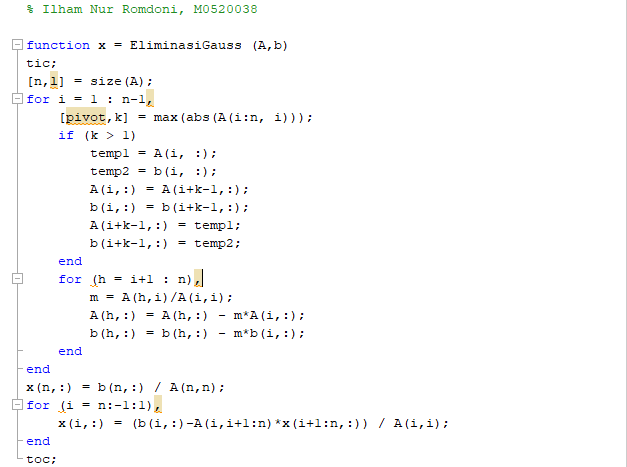
****

1. **Kasus 2**

****

1. **Kasus 3**

****

****

**ANALISIS**

1. **Analisis Source Code**
2. **Kasus 1**

**Metode Gauss**

Fungsi EliminasiGauss dengan parameter A dan b disimpan pada variabel x. Size (A,1) akan mengambil nilai dari size A pada kolom ke-1 dan disimpan pada variabel n. Dilakukan perulangan dari 1 hingga n-1. Terdapat nilai pivot yang merupakan nilai maksimum dari mutlak A baris I hingga n kolom i, letak pivot ditunjukkan dengan nilai k. Terdapat percabangan, jika k lebih dari 1 maka nilai A baris ke-i akan diganti dengan nilai A baris ke i+k-1 dan nilai B baris ke-i akan diganti dengan nilai B baris ke i+k-1. Terdapat perulangan nilai h dari i+1 hingga n. Nilai m adalah nilai dari matriks A baris ke h kolom ke i dibagi dengan matriks A baris ke i kolom ke i. Nilai dari matriks A baris ke h di semua kolom adalah A baris ke h di semua kolom dikurangi dengan m kali A baris ke i di semua kolom. Sedangkan nilai matriks B baris ke h di semua kolom adalah nilai B baris ke h di semua kolom dikurangi dengan m kali B baris ke i di semua kolom. Algoritma yang digunakan seperti pada *source* *code* di atas.

**Metode Gauss-Jordan**

Fungsi EliminasiGaussJordan dengan parameter A dan b disimpan pada variabel x. Size (A,1) akan mengambil nilai dari size A pada kolom ke-1 dan disimpan pada variabel n. Dilakukan perulangan dari 1 hingga n-1. Terdapat nilai pivot yang merupakan nilai maksimum dari mutlak A baris I hingga n kolom i, letak pivot ditunjukkan dengan nilai k. Terdapat percabangan, jika k lebih dari 1 maka nilai A baris ke-i akan diganti dengan nilai A baris ke i+k-1 dan nilai B baris ke-i akan diganti dengan nilai B baris ke i+k-1. Terdapat perulangan nilai h dari i+1 hingga n. Nilai m adalah nilai dari matriks A baris ke h kolom ke i dibagi dengan matriks A baris ke i kolom ke i. Nilai dari matriks A baris ke h di semua kolom adalah A baris ke h di semua kolom dikurangi dengan m kali A baris ke i di semua kolom. Sedangkan nilai matriks B baris ke h di semua kolom adalah nilai B baris ke h di semua kolom dikurangi dengan m kali B baris ke i di semua kolom. Algoritma yang digunakan seperti pada *source* *code* di atas.

**Metode LU Dekomposisi**

Fungsi LU Dekomposisi di atas menggunakan metode Doolittle untuk menemukan nila L dan U menggunakan parameter A. Algoritma yang digunakan seperti pada *source* *code* di atas. LU\_Solusi digunakan setelah nilai L dan U ditemukan dengan fungsi Doolitle. Fungsi LU\_Solusi menggunakan parameter L, U, dan b. Algoritma yang digunakan seperti pada *source* *code* di atas.

**Metode Jacobi**

Fungsi IterasiJacobi dengan parameter A, b dengan e didefinisikan dengan 0,0001 dan N adalah iterasi maksimal yaitu 100 disimpan pada variabel x. Size (A,1) akan mengambil nilai dari *size* A pada kolom ke-1 dan disimpan pada variabel n. Dilakukan perulangan dari 1 hingga n. Tiap perulangan akan mengeksekusi nilai b pada baris ke -i dibagi dengan nilai A pada baris ke-i kolom i dan hasilnya disimpan pada variabel xlama. Nilai xlama akan ditransposisi. Kemudian terdapat fungsi perulangan dimulai dari i hingga n. Setiap perulangan akan mengeksekusi nilai C pada baris ke-i kolom i = 0,0. Nilai C pada baris i untuk setiap kolom adalah Nilai C pada baris i untuk setiap kolom dibagi nilai A baris ke-i kolom i. Nilai d baris i kolom i adalah nilai dari xlama baris ke-i. Fungsi perulangan dimulai dari 1 hingga N. Nilai xbaru merupakan nilai C dikali nilai xlama ditambah dengan d. Kemudian terdapat fungsi percabangan jika nilai absolut dari xbaru-xlama kurang dari sama dengan nilai e, maka nilai x=xbaru akan menampilkan *output* “Jacobi method converge” dan perulangan berhenti. Namun, jika tidak memenuhi, maka akan menampilkan *output* “Jacobi method not converge”.

**Metode Gauss-Seidel**

Fungsi IterasiGaussSeidel dengan parameter A, b dengan e didefinisikan dengan 0,0001 dan N adalah iterasi maksimal yaitu 100 disimpan pada variabel x. Size (A,1) akan mengambil nilai dari *size* A pada kolom ke-1 dan disimpan pada variabel n. Dilakukan perulangan dari 1 hingga n. Tiap perulangan akan mengeksekusi nilai b pada baris ke -i dibagi dengan nilai A pada baris ke-i kolom i dan hasilnya disimpan pada variabel xlama. Nilai xlama akan ditransposisi. Kemudian terdapat fungsi perulangan dimulai dari i hingga n. Setiap perulangan akan mengeksekusi nilai C pada baris ke-i kolom i = 0,0. Nilai C pada baris i untuk setiap kolom adalah Nilai C pada baris i untuk setiap kolom dibagi nilai A baris ke-i kolom i. Nilai d baris i kolom i adalah nilai dari xlama baris ke-i. Fungsi perulangan dimulai dari 1 hingga N. Nilai x merupakan nilai C dikali nilai xlama ditambah dengan d. Kemudian terdapat fungsi percabangan jika nilai absolut dari xlama-x kurang dari sama dengan nilai e, maka nilai x akan menampilkan *output* “Gauss Seidel method converge” dan perulangan berhenti. Namun, jika tidak memenuhi, maka akan menampilkan *output* “Gauss Seidel method not converge”.

1. **Kasus 2**

Perhitungan nilai *error* menggunakan nilai mutlak dari nilai salah satu metode langsung – salah satu metode iteratif. Pada praktikum ini didefinisikan langsung adalah nilai dai EliminasiGaussJordan (A,b) – IterasiJacobi (A,b). Kemudian hasil *error* akan ditampilkan.

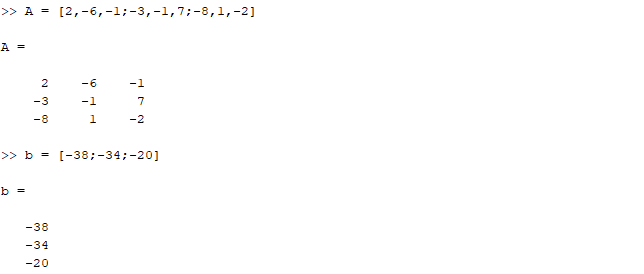
1. **Kasus 3**

Ketikkan tic dan toc yang berguna untuk menghitung waktu dari eksekusi program atau running time. Cara penulisaannya tic; algoritma toc;.

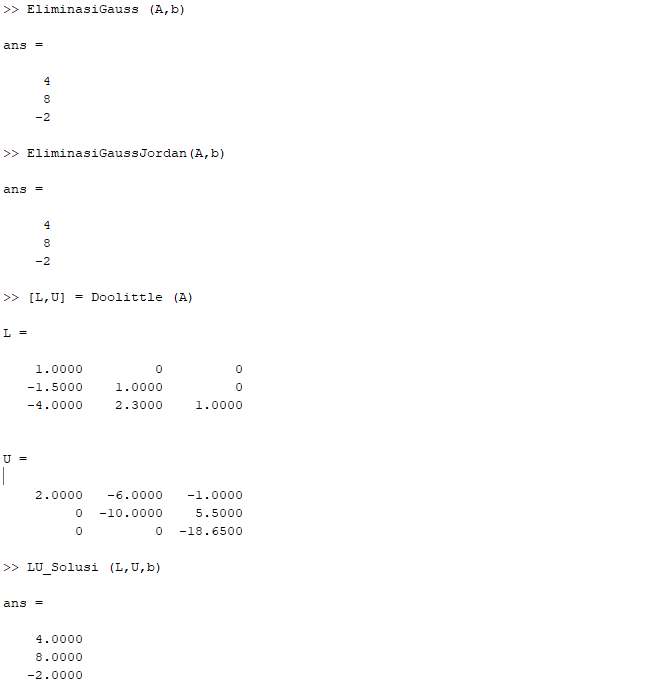
1. **Analisis Jalannya Program**
2. **Kasus 1**
3. **2x – 6y – z = -38**

**-3x – y + 7z = -34**

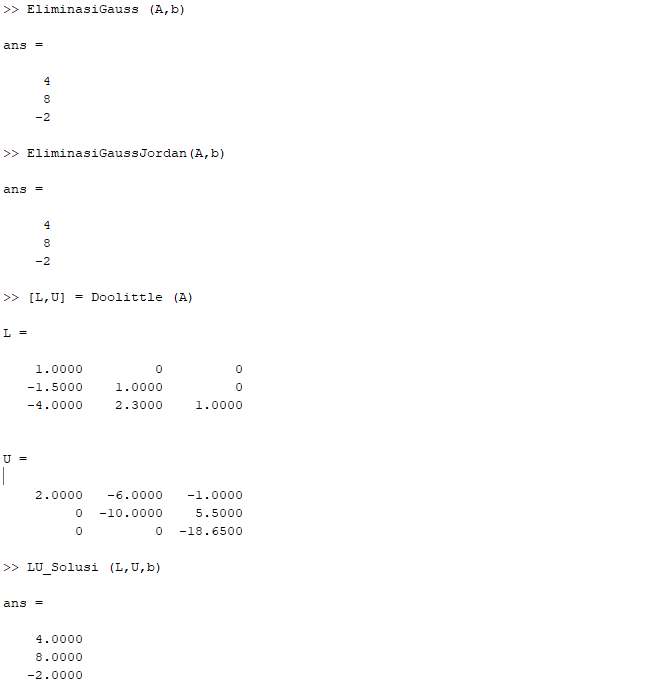
**-8x + y – 2z = -20**



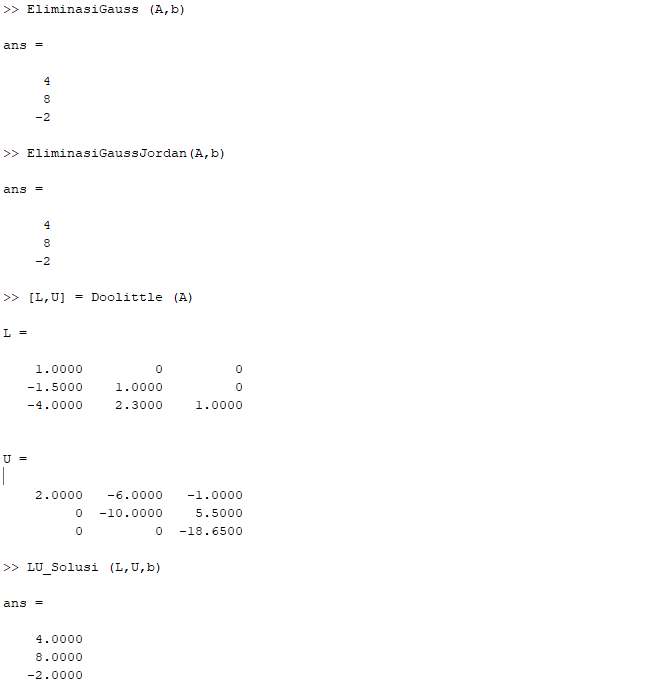
**Metode Gauss**

****

**Metode Gauss-Jordan**

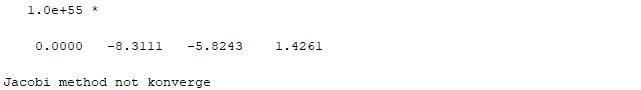
****

**Metode LU Dekomposisi**

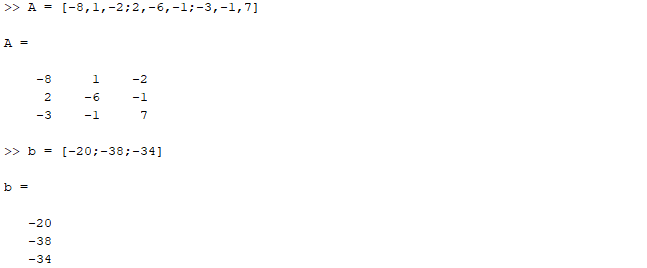
****

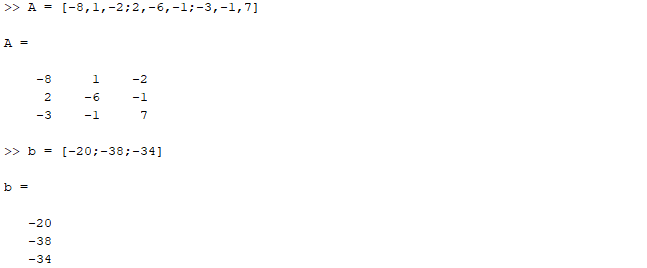
**Metode Jacobi**

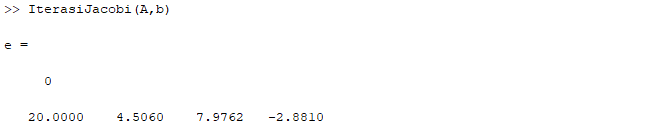
****

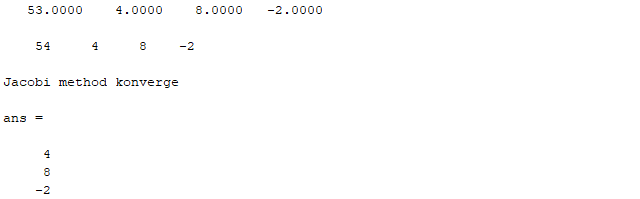
****

Metode tak konvergen maka matriks diubah menjadi *diagonally dominant*.

****

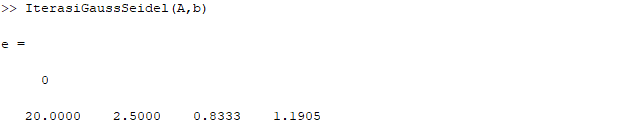
****

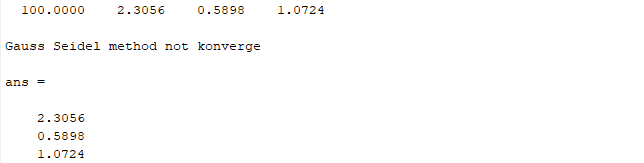
****

****

Penyelesaian SPL ditemukan pada iterasi ke-54.

**Metode Gauss-Seidel**





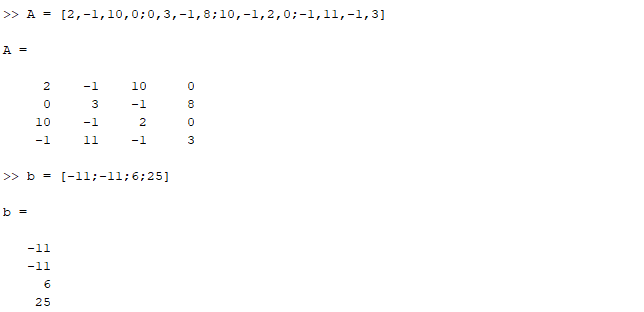
Walaupun sudah pada iterasi maksimum, penyelesaian belum ditemukan.

1. **2a – b + 10c = -11**

**3b – c + 8d = -11**

**10a – b + 2c = 6**

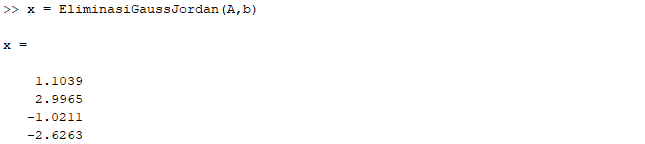
**-a + 11b – c + 3d = 25**

****

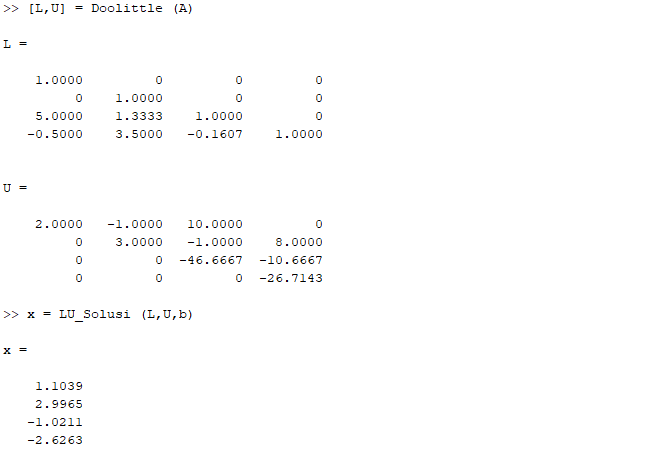
**Metode Gauss**

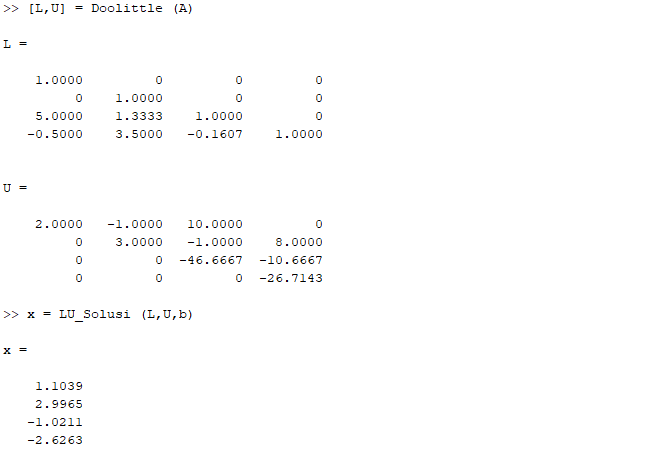
****

**Metode Gauss-Jordan**

****

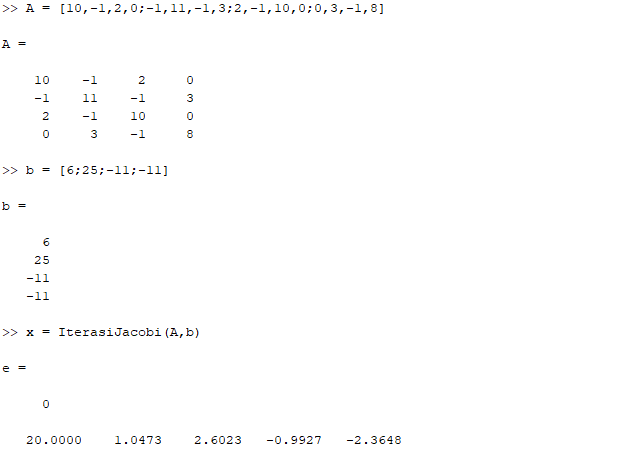
**Metode LU Dekomposisi**

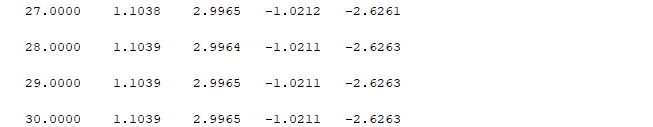
****

****

**Metode Jacobi**

Metode tak konvergen maka matriks diubah menjadi *diagonally dominant*.

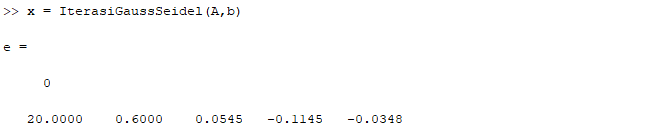
****

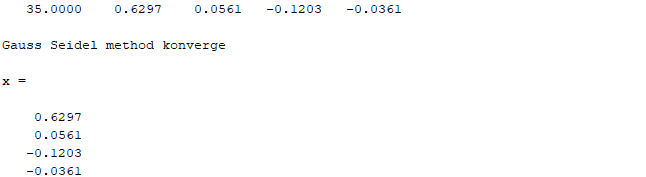
****

****

Walaupun sampai pada iterasi ke-29 dan seterusnya sudah menemukan hasil yang sangat mendekati penyelesaian, tetapi pada iterasi maksimal belum juga ditemukan hasil yang spesifik pada nilai penyelesaian.

**Metode Gauss-Seidel**

****

****

Hasil penyelesaian ditemukan, tetapi sangat jauh dari hasil penyelesaian keempat metode lain.

1. **6p – q – r = -1**

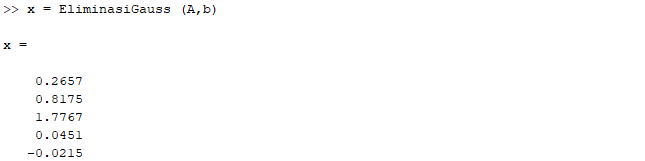
**-p + 5q – r – s = 2**

**-p – q + 4r – s – t = 6**

**r + 4s – 2t = 2**

**q – r + s + 4t = -1**

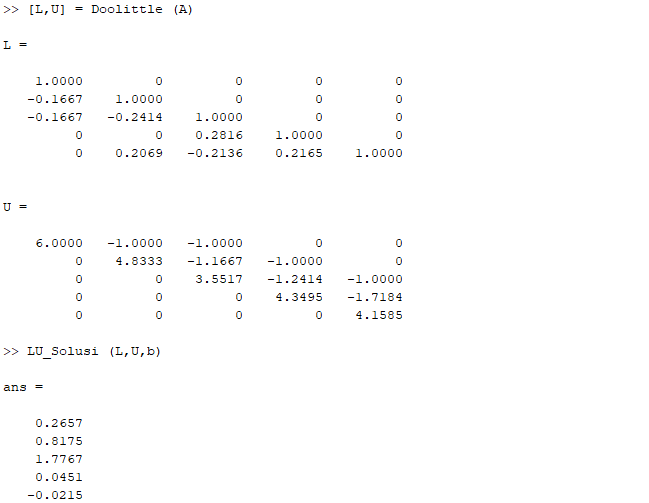
**Metode Gauss**

****

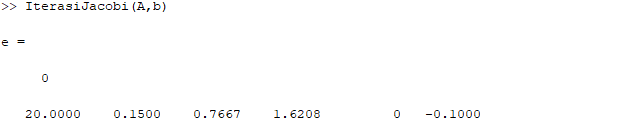
**Metode Gauss-Jordan**

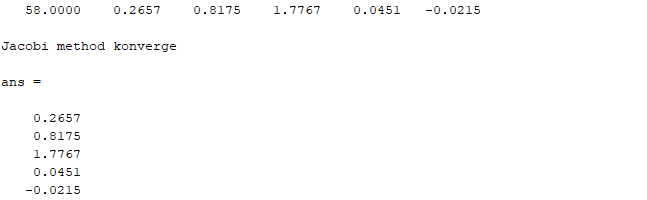
****

**Metode LU Dekomposisi**

****

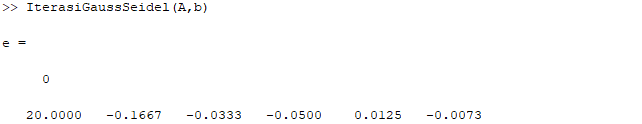
**Metode Jacobi**

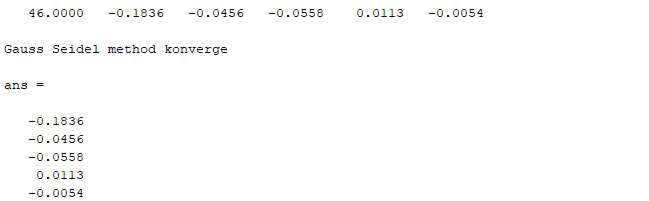
****

****

Pada iterasi ke-58 ditemukan hasil penyelesaian.

**Metode Gauss-Seidel**

****

****

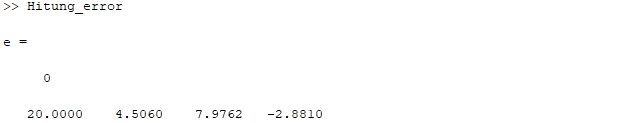
Pada metode Gauss-Seidel ditemukan hasil penyelesaian tetapi dengan hasil yang jauh dari keempat metode lain.

Kesimpulan dari ketiga soal di atas himpunan penyelesaian dari metode Gauss-Seidel selalu ditemukan berbeda dengan keempat metode lain.

1. **Kasus 2**
2. **2x – 6y – z = -38**

**-3x – y + 7z = -34**

**-8x + y – 2z = -20**

****

****

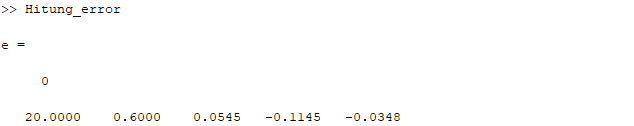
Menunjukkan himpunan penyelesaian dari metode langsung dan iteratif sama.

1. **2a – b + 10c = -11**

**3b – c + 8d = -11**

**10a – b + 2c = 6**

**-a + 11b – c + 3d = 25**

****

****

*Error* ditemukan agak tinggi karena untuk metode iteratif menggunakan hasil penyelesaian metode Gauss-Seidel.

1. **6p – q – r = -1**

**-p + 5q – r – s = 2**

**-p – q + 4r – s – t = 6**

**r + 4s – 2t = 2**

**q – r + s + 4t = -1**

****

*Error* menunjukkan hasil penyelesaian mendekati tetapi tidak sama persis.

1. **Kasus 3**
2. **2x – 6y – z = -38**

**-3x – y + 7z = -34**

**-8x + y – 2z = -20**

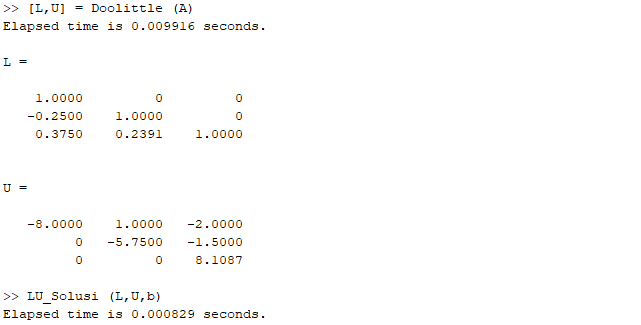
**Metode Gauss**

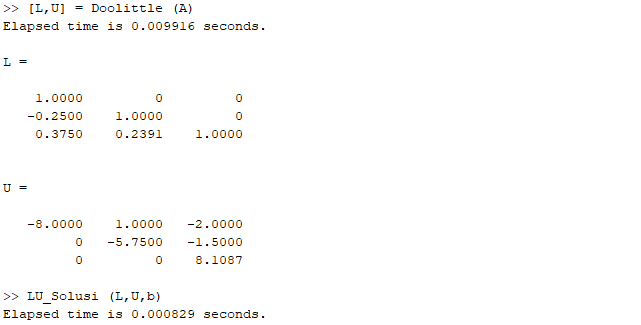
****

**Metode Gauss-Jordan**

****

**Metode LU Dekomposisi**

****

****

**Metode Jacobi**

****

**Metode Gauss-Seidel**

****

1. **2a – b + 10c = -11**

**3b – c + 8d = -11**

**10a – b + 2c = 6**

**-a + 11b – c + 3d = 25**

**Metode Gauss**

****

**Metode Gauss-Jordan**

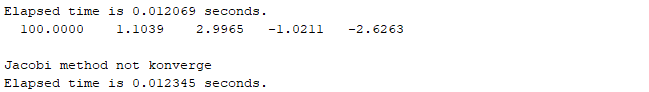
****

**Metode LU Dekomposisi**

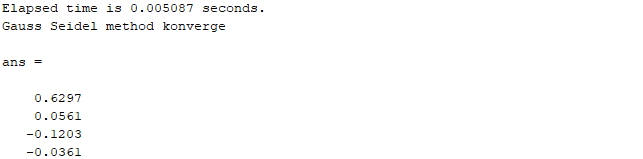
****

****

**Metode Jacobi**

****

**Metode Gauss-Seidel**

****

1. **6p – q – r = -1**

**-p + 5q – r – s = 2**

**-p – q + 4r – s – t = 6**

**r + 4s – 2t = 2**

**q – r + s + 4t = -1**

**Metode Gauss**

****

**Metode Gauss-Jordan**

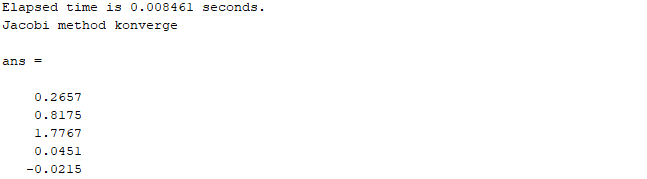
****

**Metode LU Dekomposisi**

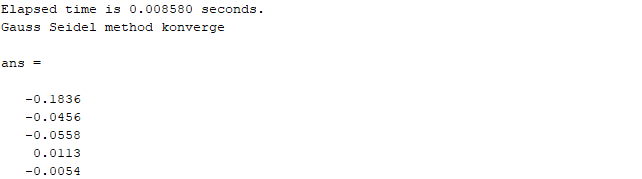
****

****

**Metode Jacobi**



**Metode Gauss-Seidel**

****

Dapat disimpulkan dari semua metode, *running* *time* yang paling singkat adalah menggunakan metode Gauss dan yang paling lama adalah LU Dekomposisi karena menjalankan dua fungsi algoritma. Tetapi dalam singkat lamanya *running* *time*, sebenarnya ditentukan dari seberapa efektif algoritma yang disusun oleh pemrogram.