Nama : Annisa Charisma Wijayanti

NIM : 211220122140086

Mata Kuliah : Metode Numerik

Jurusan : Teknik Komputer

**Implemmentasi Sistem Persamaan Linear**

1. **Metode Matriks Balikan**

Metode matriks balikan digunakan untuk menyelesaikan sistem persamaan linear 𝐴𝑥=𝑏 dengan mengalikan invers dari matriks koefisien 𝐴 dengan vektor konstanta 𝑏. Penjelasan alur kode dengan menggunakan bahasa pemrograman python yaitu:

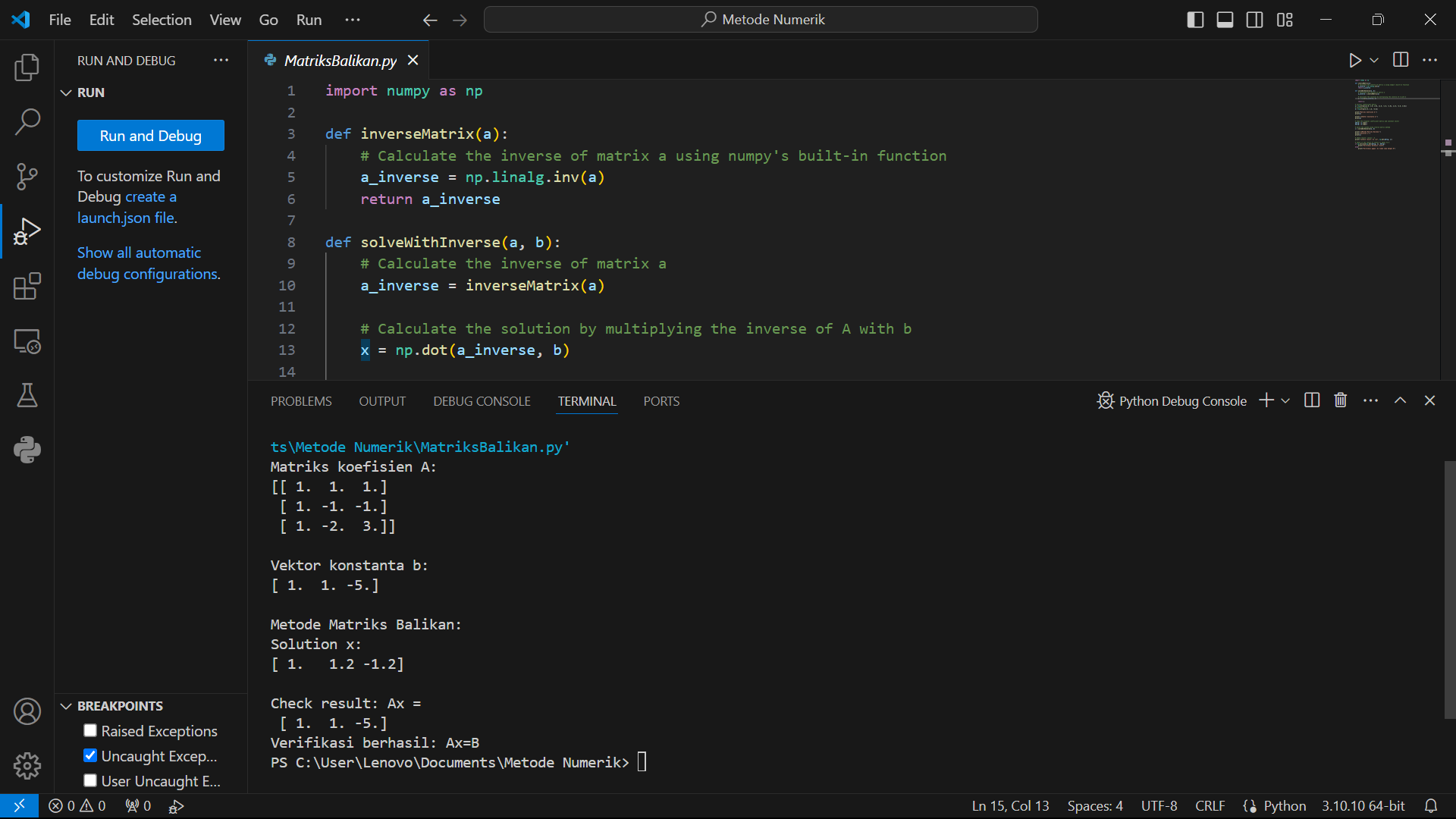
1. Import NumPy
2. Definisi Fungsi inverseMatrix(a):

* Fungsi ini menerima matriks 𝐴 sebagai argumen.
* Menggunakan fungsi np.linalg.inv(a) dari NumPy untuk menghitung invers dari matriks 𝐴.
* Mengembalikan matriks invers 𝐴−1.

1. Definisi Fungsi solveWithInverse(a, b):

* Fungsi ini menerima dua argumen, yaitu matriks koefisien 𝐴 dan vektor konstanta 𝑏.
* Memanggil fungsi inverseMatrix(a) untuk menghitung invers dari matriks 𝐴.
* Menghitung solusi 𝑥 dengan mengalikan invers dari matriks 𝐴 dengan vektor 𝑏 menggunakan fungsi np.dot(a\_inverse, b).
* Mengembalikan vektor solusi 𝑥.

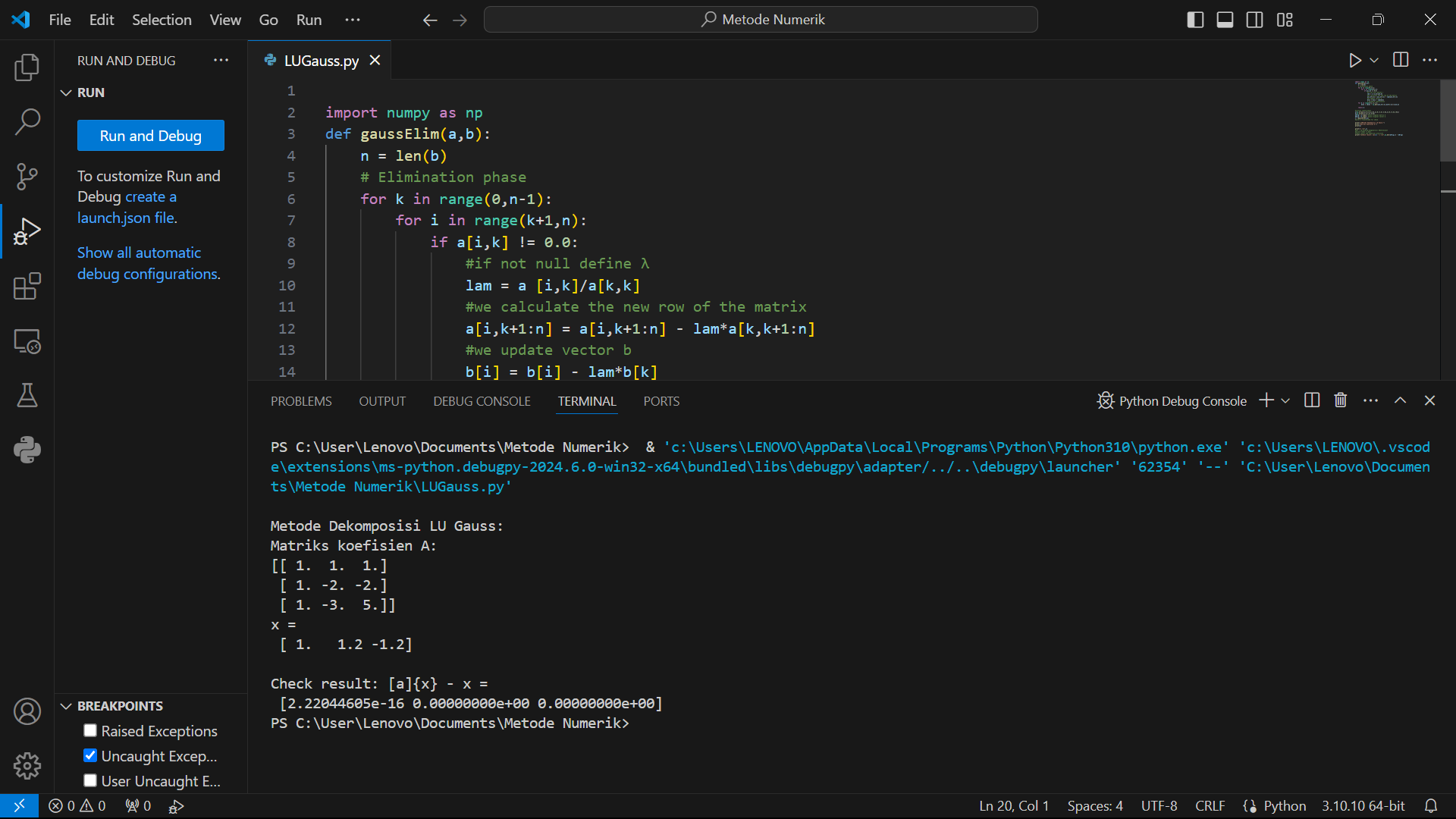
1. Inisialisasi Matriks koefisien 𝐴 dan vektor konstanta 𝑏 dengan nilai-nilai tertentu.
2. Matriks koefisien asli 𝐴 dan vektor konstanta asli 𝑏 disalin untuk verifikasi hasil nantinya.
3. Pemanggilan Fungsi solveWithInverse(a,b) sebagai argumen untuk mendapatkan solusi dari sistem persamaan linear 𝐴𝑥=𝑏.
4. Cetak Solusi 𝑥
5. Verifikasi kebenaran solusi yang ditemukan, hasil dari perkalian matriks koefisien asli 𝐴 dengan solusi 𝑥 (yaitu 𝐴𝑥) dihitung menggunakan np.dot(aOrig, x), lalu hasilnya dibandingkan dengan vektor konstanta asli 𝑏. Jika hasilnya mendekati nol atau sangat kecil (dengan menggunakan np.allclose()), maka verifikasi berhasil. Jika tidak, maka verifikasi gagal.



1. **Metode Dekomposisi LU Gauss**

Metode dekomposisi LU Gauss adalah salah satu metode dalam aljabar linier yang digunakan untuk menyelesaikan sistem persamaan linear 𝐴𝑥=𝑏. Metode ini melibatkan faktorisasi matriks koefisien 𝐴 menjadi dua matriks segitiga, yaitu matriks segitiga bawah 𝐿 dan matriks segitiga atas 𝑈, di mana 𝐴=𝐿𝑈. Pada metode ini, proses dekomposisi dilakukan dengan melakukan eliminasi Gauss pada matriks koefisien 𝐴. Faktorisasi dilakukan dengan menemukan matriks segitiga atas 𝑈 terlebih dahulu, diikuti dengan pencarian matriks segitiga bawah 𝐿 yang memungkinkan 𝐴=𝐿𝑈. Penjelasan alur kode dengan menggunakan bahasa pemrograman python yaitu:

* 1. Definisi Fungsi gaussElim(a, b):
* Fungsi ini menerima dua argumen, yaitu matriks koefisien *A* dan vektor konstanta *b*.
* Variabel *n* diinisialisasi dengan panjang vektor *b*, yang juga merupakan ukuran matriks koefisien *A*.
* Fase eliminasi dilakukan menggunakan nested loop: iterasi pertama untuk setiap baris (kecuali baris terakhir), dan iterasi kedua untuk setiap baris di bawah baris saat ini.
* Pada setiap iterasi, dilakukan pengecekan apakah elemen diagonal utama *akk* tidak nol. Jika nol, langkah eliminasi dilewati karena akan menghasilkan pembagian dengan nol.
* Jika elemen diagonal utama tidak nol, faktor pengali λ dihitung dan digunakan untuk memperbarui matriks *A* dan vektor *b*.
* Tahap substitusi mundur dilakukan untuk menghitung solusi *x*.
* Vektor solusi *x* dikembalikan.
  1. Inisialisasi Matriks koefisien *A* dan vektor konstanta *b* diinisialisasi dengan nilai-nilai tertentu.
  2. Menyalin Matriks koefisien asli *A* dan vektor konstanta asli *b* untuk verifikasi hasil nantinya.
  3. Panggil Fungsi gaussElim(a, b) dengan matriks koefisien *A* dan vektor konstanta *b* sebagai argumen untuk mendapatkan solusi dari sistem persamaan linear *Ax = b*.
  4. Cetak Matriks *A* dan Solusi *x* untuk ditampilkan kepada pengguna.
  5. Verifikasi Hasil: Hasil dari *Ax* dihitung menggunakan np.dot(aOrig, x), lalu hasilnya dibandingkan dengan vektor konstanta asli *b*. Jika hasilnya mendekati nol atau sangat kecil (dengan menggunakan np.allclose()), maka verifikasi berhasil. Jika tidak, maka verifikasi gagal.



1. **Metode Dekomposisi Crout**

Metode Dekomposisi Crout merupakan metode yang memfaktorkan matriks koefisien *A* dari sistem persamaan 𝐴𝑋 = 𝑌 menjadi hasil kali suatu matriks segitiga bawah 𝐿 yang elemen diagonal utamanya bernilai tak nol dan matriks segitiga atas 𝑈 yang elemen diagonal utamanya bernilai satu. Dengan demikian sistem persamaan linear akan berubah menjadi *AX = LUX = Y*. Pada metode ini, proses dekomposisi dilakukan dengan pendekatan Crout, di mana elemen-elemen matriks 𝐿 dan 𝑈 dihitung secara bergantian. Faktorisasi dilakukan dengan menemukan matriks segitiga bawah 𝐿 terlebih dahulu, diikuti dengan pencarian matriks segitiga atas 𝑈 yang memungkinkan 𝐴=𝐿𝑈. Penjelasan alur kode dengan menggunakan bahasa pemrograman python yaitu:

* 1. Definisi Fungsi croutDecomposition(a):
* Fungsi ini menerima matriks koefisien *A* sebagai argumen.
* Membuat matriks *L* dan *U* dengan ukuran yang sama dengan matriks koefisien *A*, dan diinisialisasi dengan nilai nol.
* Melakukan dekomposisi Crout dengan menggunakan nested loop:
* Iterasi pertama untuk setiap kolom. Iterasi kedua untuk setiap baris, dimulai dari indeks kolom saat ini hingga indeks terakhir.
* Pada setiap iterasi, menghitung elemen *L* dan *U* menggunakan rumus dekomposisi Crout.
* Mengembalikan matriks *L* dan *U*.
  1. Definisi Fungsi solveWithCrout(a, b):
* Fungsi ini menerima matriks koefisien *A* dan vektor konstanta *b* sebagai argumen.
* Memanggil fungsi croutDecomposition(a)untuk mendapatkan matriks *L* dan *U*.
* Menghitung solusi *x* menggunakan substitusi maju dan mundur.
* Substitusi maju dilakukan untuk menyelesaikan sistem *Ly = b* dengan matriks segitiga bawah *L*.
* Substitusi mundur dilakukan untuk menyelesaikan sistem *Ux = y* dengan matriks segitiga atas *U*.
* Mengembalikan vektor solusi *x*.
  1. Inisialisasi Matriks koefisien *A* dan vektor konstanta *b* dengan nilai-nilai tertentu.
  2. Panggil Fungsi solveWithCrout(a, b) dengan matriks koefisien *A* dan vektor konstanta *b* sebagai argumen untuk mendapatkan solusi dari sistem persamaan linear *Ax = b*.
  3. Cetak Matriks *A* dan Solusi *x*

