

Nama : Rachel Savitri

NIM : 21120122140111

Kelas : C

Penjabaran Kode Pada Metode

Dekomposisi Crout

```
// Fungsi untuk melakukan dekomposisi Crout
void croutDecomposition(vector<vector<double>>& A, vector<vector<double>>&
L, vector<vector<double>>& U) {
    int n = A.size();

    // Inisialisasi matriks L dan U dengan nilai 0
    L = vector<vector<double>>(n, vector<double>(n, 0));
    U = vector<vector<double>>(n, vector<double>(n, 0));

    // Proses dekomposisi Crout
    for (int i = 0; i < n; i++) {
        // Mengisi diagonal utama matriks U
        U[i][i] = 1;

        // Menghitung elemen-elemen matriks L dan U
        for (int j = i; j < n; j++) {
            double sum = 0;
            for (int k = 0; k < i; k++) {
                sum += L[j][k] * U[k][i];
            }
            L[j][i] = A[j][i] - sum;
        }

        for (int j = i + 1; j < n; j++) {
            double sum = 0;
            for (int k = 0; k < i; k++) {
                sum += L[i][k] * U[k][j];
            }
            U[i][j] = (A[i][j] - sum) / L[i][i];
        }
    }
}

// Fungsi untuk menyelesaikan SPL dengan dekomposisi Crout
vector<double> solveSPL(vector<vector<double>>& A, vector<double>& b) {
    int n = A.size();
    vector<vector<double>> L, U;

    // Melakukan dekomposisi Crout
    croutDecomposition(A, L, U);

    // Penyelesaian SPL dengan dekomposisi Crout
    vector<double> y(n, 0), x(n, 0);

    // Solusi y dari Ly = b
    for (int i = 0; i < n; i++) {
        double sum = 0;
```

```

        for (int j = 0; j < i; j++) {
            sum += L[i][j] * y[j];
        }
        y[i] = (b[i] - sum) / L[i][i];
    }

    // Solusi x dari  $Ux = y$ 
    for (int i = n - 1; i >= 0; i--) {
        double sum = 0;
        for (int j = i + 1; j < n; j++) {
            sum += U[i][j] * x[j];
        }
        x[i] = (y[i] - sum) / U[i][i];
    }

    return x;
}

```

Alurnya dimulai dengan melakukan dekomposisi matriks koefisien A menjadi matriks segitiga bawah L dan matriks segitiga atas U . Setelah mendapatkan dekomposisi, solusi SPL diperoleh dengan menggunakan matriks L dan U , yang kemudian dihitung dengan metode substitusi mundur dan substitusi maju. Hasil solusi SPL tersebut kemudian ditampilkan ke layar. Bagian `main()` merupakan bagian pengujian yang menggunakan SPL contoh dan menampilkan solusinya ke layar.

```

//-----Kode Testing-----//
int main() {
    // Contoh SPL
    vector<vector<double>> A = {{4, -2, 1}, {-2, 4, -2}, {1, -2, 4}};
    vector<double> b = {11, -16, 17};

    // Menyelesaikan SPL dengan dekomposisi Crout
    vector<double> x = solveSPL(A, b);

    // Menampilkan solusi
    cout << "Solusi SPL:" << endl;
    for (int i = 0; i < x.size(); i++) {
        cout << "x[" << i << "] = " << x[i] << endl;
    }

    return 0;
}

```

Bagian ini merupakan kode yang digunakan untuk menguji implementasi fungsi penyelesaian SPL dengan dekomposisi Crout. Di sini, diberikan contoh SPL dalam bentuk matriks koefisien A dan vektor konstanta b . Kemudian, fungsi `solveSPL` dipanggil untuk menyelesaikan SPL menggunakan dekomposisi Crout. Solusi SPL yang diperoleh kemudian ditampilkan ke layar.

Hasil Output

```
"D:\tekkom\smt 4\metode nu" × + v
Solusi SPL:
x[0] = 1
x[1] = -2
x[2] = 3

Process returned 0 (0x0)   execution time : 0.048 s
Press any key to continue.
|
```