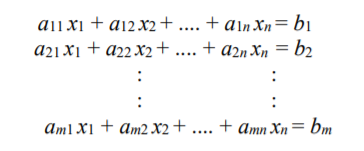
**BAB I**

**DESKRIPSI MASALAH**

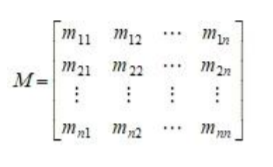
* 1. **Abstraksi**

Sistem persamaan linier (SPL) Ax = b dengan n peubah (variable) dan m persamaan adalah berbentuk

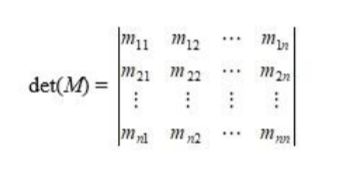


yang dalam hal ini xi adalah peubah, aij dan bi adalah koefisien ∈ R. Sembarang SPL dapat diselesaikan dengan beberapa metode, yaitu metode eliminasi Gauss, metode eliminasi Gauss-Jordan, metode matriks balikan (x = A -1 b), dan kaidah Cramer (khusus untuk SPL dengan n peubah dan n persamaan). Solusi sebuah SPL mungkin tidak ada, banyak, atau hanya satu (unik/tunggal).

Sebuah matriks M berukuran n × n



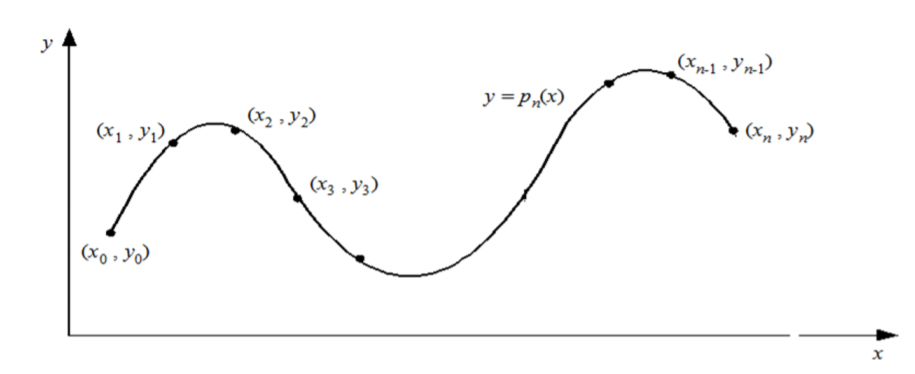
determinannya adalah

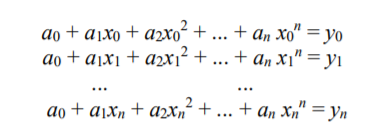


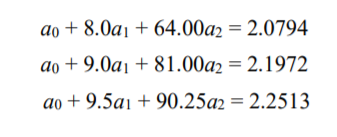
Determinan matriks M berukuran n × n dapat dihitung dengan beberapa cara: reduksi baris dan ekspansi kofaktor. SPL memiliki banyak aplikasi dalam bidang sains dan rekayasa, dua diantaranya diterapkan pada tugas besar ini, yaitu interpolasi polinom dan regresi linier.

* 1. **Interpolasi Polinom**

Persoalan interpolasi polinom adalah sebagai berikut: Diberikan n+1 buah titik berbeda, (x0, y0), (x1, y1), ..., (xn, yn). Tentukan polinom pn(x) yang menginterpolasi (melewati) semua titik-titik tersebut sedemikian rupa sehingga yi = pn(xi) untuk i = 0, 1, 2, …, n.

Setelah polinom interpolasi pn(x) ditemukan, pn(x) dapat digunakan untuk menghitung perkiraan nilai y di sembarang titik di dalam selang [x0, xn].

Polinom interpolasi derajat n yang menginterpolasi titik-titik (x0, y0),(x1, y1), ..., (xn, yn). adalah berbentuk pn(x) = a0 + a1x + a2x 2 + … + anx n . Jika hanya ada dua titik, (x0, y0) dan (x1, y1), maka polinom yang menginterpolasi kedua titik tersebut adalah p1(x) = a0 + a1x yaitu berupa persamaan garis lurus. Jika tersedia tiga titik, (x0, y0), (x1, y1), dan (x2, y2), maka polinom yang menginterpolasi ketiga titik tersebut adalah p2(x) = a0 + a1x + a2x 2 atau persaman kuadrat dan kurvanya berupa parabola. Jika tersedia empat titik, (x0, y0), (x1, y1), (x2, y2), dan (x3, y3), polinom yang menginterpolasi keempat titik tersebut adalah p3(x) = a0 + a1x + a2x 2 + a3x 3 , demikian seterusnya. Dengan cara yang sama kita dapat membuat polinom interpolasi berderajat n untuk n yang lebih tinggi asalkan tersedia (n+1) buah titik data. Dengan menyulihkan (xi, yi) ke dalam persamaan polinom pn(x) = a0 + a1x + a2x 2 + … + anx n untuk i = 0, 1, 2, …, n, akan diperoleh n buah sistem persamaan lanjar dalam a0, a1, a2, …, an,

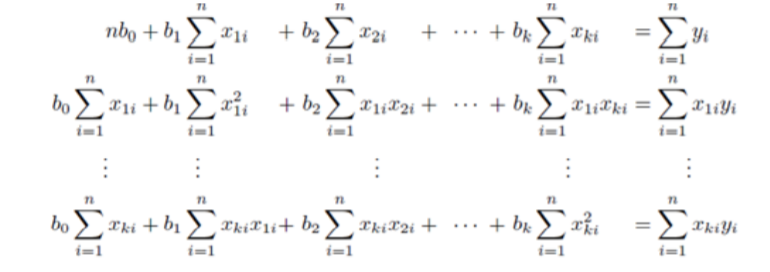
Solusi sistem persamaan lanjar ini, yaitu nilai a0, a1, …, an, diperoleh dengan menggunakan metode eliminasi Gauss yang sudah anda pelajari. Sebagai contoh, misalkan diberikan tiga buah titik yaitu (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513). Tentukan polinom interpolasi kuadratik lalu estimasi nilai fungsi pada x = 9.2. Polinom kuadratik berbentuk p2(x) = a0 + a1x + a2x 2 . Dengan menyulihkan ketiga buah titik data ke dalam polinom tersebut, diperoleh sisten persamaan lanjar yang terbentuk adalah

Penyelesaian sistem persamaan dengan metode eliminasi Gauss menghasilkan a0 = 0.6762, a1 = 0.2266, dan a2 = -0.0064. Polinom interpolasi yang melalui ketiga buah titik tersebut adalah p2(x) = 0.6762 + 0.2266x - 0.0064x 2 . Dengan menggunakan polinom ini, maka nilai fungsi pada x = 9.2 dapat ditaksir sebagai berikut: p2(9.2) = 0.6762 + 0.2266(9.2) - 0.0064(9.2)2 = 2.2192.

* 1. **Regresi Linier Berganda**

Regresi Linear (akan dipelajari lebih lanjut di Probabilitas dan Statistika) merupakan salah satu metode untuk memprediksi nilai selain menggunakan Interpolasi Polinom. Meskipun sudah ada rumus jadi untuk menghitung regresi linear sederhana, terdapat rumus umum dari regresi linear yang bisa digunakan untuk regresi linear berganda, yaitu

Untuk mendapatkan nilai dari setiap βi dapat digunakan Normal Estimation Equation for Multiple Linear Regression sebagai berikut:



Sistem persamaan linier tersebut diselesaikan dengan menggunakan metode eliminasi Gauss.

* 1. **Spesifikasi tugas**

Buatlah program dalam **Bahasa Java** untuk

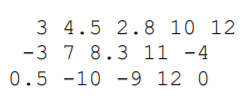
1. Menghitung solusi SPL dengan metode eliminasi Gauss, metode Eliminasi Gauss-Jordan, metode matriks balikan, dan kaidah Cramer (kaidah Cramer khusus untuk SPL dengan n peubah dan n persamaan).

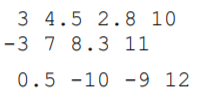
2. Menyelesaikan persoalan interpolasi dan regresi linier.

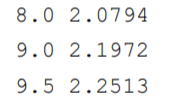
3. Menghitung matriks balikan

4. Menghitung determinan matriks dengan berbagai metode (reduksi baris dan ekspansi kofaktor).

Spesifikasi program adalah sebagai berikut:

1. Program dapat menerima masukan (input) baik dari keyboard maupun membaca masukan dari file text. Untuk SPL, masukan dari keyboard adalah m, n, koefisien aij , dan bi. Masukan dari file berbentuk matriks augmented tanpa tanda kurung, setiap elemen matriks dipisah oleh spasi. Misalnya

2. Untuk persoalan menghitung determinan dan matriks balikan, masukan dari keyboard adalah n dan koefisien aij. Masukan dari file berbentuk matriks, setiap elemen matriks dipisah oleh spasi. Misalnya,

3. Untuk persoalan interpolasi, masukannya jika dari keyboard adalah n, (x0, y0), (x1, y1), ..., (xn, yn), dan nilai x yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung. Misalnya jika titik-titik datanya adalah (8.0, 2.0794), (9.0, 2.1972), dan (9.5, 2.2513), maka di dalam file text ditulis sebagai berikut:

4. Untuk persoalan regresi, masukannya jika dari keyboard adalah n (jumlah peubah x), semua nilai-nilai x1i, x2i, ..., xni, nilai yi, dan nilai-nilai xk yang akan ditaksir nilai fungsinya. Jika masukannya dari file, maka titik-titik dinyatakan pada setiap baris tanpa koma dan tanda kurung.

5. Untuk persoalan SPL, luaran (output) program adalah solusi SPL. Jika solusinya tunggal, tuliskan nilainya. Jika solusinya tidak ada, tuliskan solusi tidak ada, jika solusinya banyak, maka tuliskan solusinya dalam bentuk parametrik (misalnya x4 = -2, x3 = 2s – t, x2 = s, dan x1 = t.)

6. Untuk persoalan determinan dan matriks balikan, maka luarannya sesuai dengan persoalan masing-masing

7. Untuk persoalan polinom interpolasi dan regresi, luarannya adalah persamaan polinom/regresi dan taksiran nilai fungsi pada x yang diberikan.

8. Luaran program harus dapat ditampilkan pada layar komputer dan dapat disimpan ke dalam file.

9. Bahasa program yang digunakan adalah Java.

10. Program tidak harus berbasis GUI, cukup text-based saja, namun boleh menggunakan GUI (memakai kakas Eclipse misalnya).

11. Program dapat dibuat dengan pilihan menu. Urutan menu dan isinya dipersilakan dirancang masing-masing. Misalnya, menu:

MENU

1. Sistem Persamaaan Linier

2. Determinan

3. Matriks balikan

4. Interpolasi Polinom

5. Regresi linier berganda

6. Keluar

Untuk pilihan menu nomor 1 ada sub-menu lagi yaitu pilihan metode:

1. Metode eliminasi Gauss

2. Metode eliminasi Gauss-Jordan

3. Metode matriks balikan

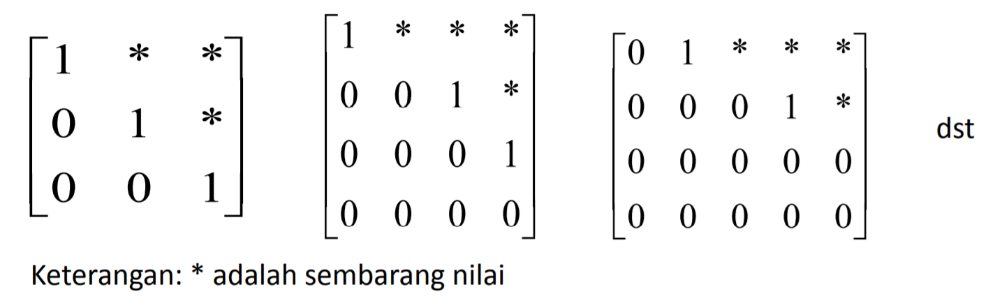
4. Kaidah Cramer

Begitu juga untuk pilihan menu nomor 2 dan 3.

**BAB II**

**TEORI SINGKAT**

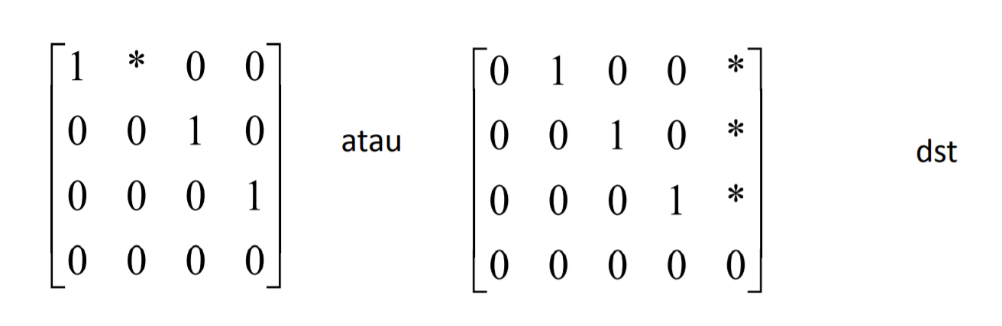
1. **Eliminasi Gauss**

Menurut Anton & Rorres (2013), Eliminasi Gauss merupakan metode penyelesaian sistem persamaan linier dengan cara mengeliminasi setiap elemen di matriks, sampai terbentuk suatu matriks eselon baris. Anton & Rorres (2013) juga menyebutkan bahwa matriks eselon baris merupakan matriks yang memiliki angka satu utama (*leading one*) dalam setiap barisnya, kecuali baris yang seluruh elemennya adalah 0. Jika ada baris dengan seluruh elemennya adalah 0, baris tersebut akan dipindahkan ke baris paling bawah dari matriks.

Gambar 1: Contoh Matriks Eselon Baris

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2018-2019/Matriks-Eselon.pdf>

1. **Eliminasi Gauss-Jordan**

Menurut Anton & Rorres (2013), Eliminasi Gauss-Jordan merupakan metode penyelesaian sistem persamaan linier dengan cara mengeliminasi setiap elemen di matriks, sampai terbentuk suatu matriks eselon baris tereduksi. Anton & Rorres (2013) juga menyebutkan bahwa matriks eselon baris tereduksi merupakan matriks yang memiliki angka satu utama (*leading one*) di setiap barisnya dan 0 dibawah dan diatas satu utamanya, kecuali untuk baris yang seluruh elemennya adalah 0. Jika ada baris dengan seluruh elemennya adalah 0, baris tersebut akan dipindahkan ke baris paling bawah dari matriks.

Gambar 2: Contoh Matriks Eselon Baris Tereduksi

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2018-2019/Matriks-Eselon.pdf>

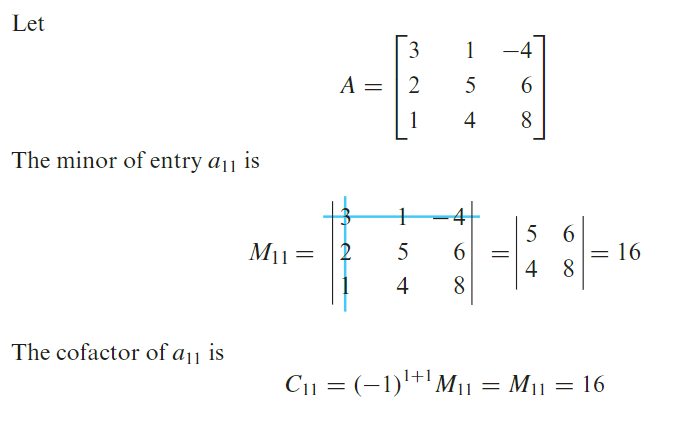
1. **Determinan**

Menurut Anton & Rorres (2013), Kata determinan mulai digunakan oleh matematikawan Carl Friedrich Gauss pada tahun 1801 untuk menjelaskan suatu properti dari fungsi tertentu. Dalam konteks matriks, determinan digunakan untuk menjelaskan sebuah angka skalar yang merepresentasikan fungsi matriks itu sendiri. Seperti yang telah disebutkan dalam bab sebelumnya, determinan matriks n × n dapat dihitung dengan beberapa cara yaitu reduksi baris dan ekspansi kofaktor.

1. **Matriks Balikan**

Menurut Rao & Mitra (1972), Untuk matriks tidak singular A, terdapat suatu matriks A-1 sedemikian sehingga AA-1 = I. I yang dimaksud merupakan matriks identitas, yaitu matriks dengan ukuran n x n yang memiliki nilai 1 di seluruh elemen diagonalnya dan 0 di elemen lain. Untuk mencari matriks balikan, metode yang dapat digunakan adalah dengan eliminasi Gauss-Jordan atau ekspansi kofaktor.

1. **Matriks Kofaktor**

Menurut Anton & Rorres (2013), untuk matriks persegi A, kofaktor Aij merupakan hasil determinan dari sub-matrix A setelah eliminasi baris ke i dan kolom ke j dari matriks A. Matriks kofaktor adalah suatu matriks yang memuat entri kofaktor dari matriks A sesuai dengan letak elemennya. Berikut contohnya

Gambar 3: Contoh Cara Mencari Entri Kofaktor

Sumber: Anton, H., & Rorres, C. (2013). *Elementary linear algebra: applications version*. John Wiley & Sons.

1. **Matriks Adjoin**

Menurut Anton & Rorres (2013), matriks adjoin adalah hasil transpose dari matriks kofaktor. Menurut Anton & Rorres (2013), transpose adalah operasi pada matriks yang membalik elemen pada matriks berdasar diagonalnya untuk matriks persegi. Untuk matriks tidak persegi, transpose juga menukar ukuran kolom dan baris dari matriks tersebut.

1. **Kaidah Cramer**

Menurut Anton & Rorres (2013), kaidah Cramer adalah suatu aturan untuk menyelesaikan sistem persamaan linier yang dapat direpresentasikan dalam matrix Ax = b. Kaidah ini menyatakan bahwa untuk sistem persamaan linier yang dapat direpresentasikan koefisiennya kedalam suatu matriks A, terdapat solusi unik untuk setiap x yaitu xn = det(An) / det(A). Dalam konteks ini, n mengarah kepada kolom yang akan diubah seluruhnya dengan entri jawaban pada spl. Sebagai contoh:

Misal spl berikut

1. x + y = 5
2. 2x + 4y = 18

Maka matriks A= [ ] (matriks koefisien spl) dan matriks B= [ ] (matriks entri hasil)

Dari kedua matriks diatas didapatkan matriks A1= [] dan A2= [] dengan menukar kolom ke n dengan matriks entri hasil

Maka, akan didapatkan x= dan y= sesuai ketentuan dari kaidah cramer.

1. **Interpolasi Polinom**

Menurut Anton & Rorres (2013), untuk n banyak titik di bidang xy yang memiliki koordinat x tertentu, terdapat sebuah solusi unik berbentuk polinomial dengan derajat setidaknya n - 1 yang melewati semua titik tersebut. Perhitungan atau pencarian polinomial ini disebut dengan interpolasi polinom. Untuk melakukan interpolasi polinom, dengan asumsi titik-titik sudah diketahui nilainya, hal yang harus dilakukan adalah memasukan semua titik tersebut ke dalam persamaan polinom, yang dapat direpresentasikan dalam bentuk matriks, dan melakukan eliminasi Gauss-Jordan untuk mendapat hasil koefisien dari setiap polinom tersebut. Contoh kita mengetahui 3 titik, maka kita perlu memasukan ketiga titik tersebut ke persamaan dengan derajat n - 1 = 2 yaitu a0 + a1x + a2x2 = y dan menyelesaikan seluruhnya dari situ.

1. **Regresi Linier Berganda**

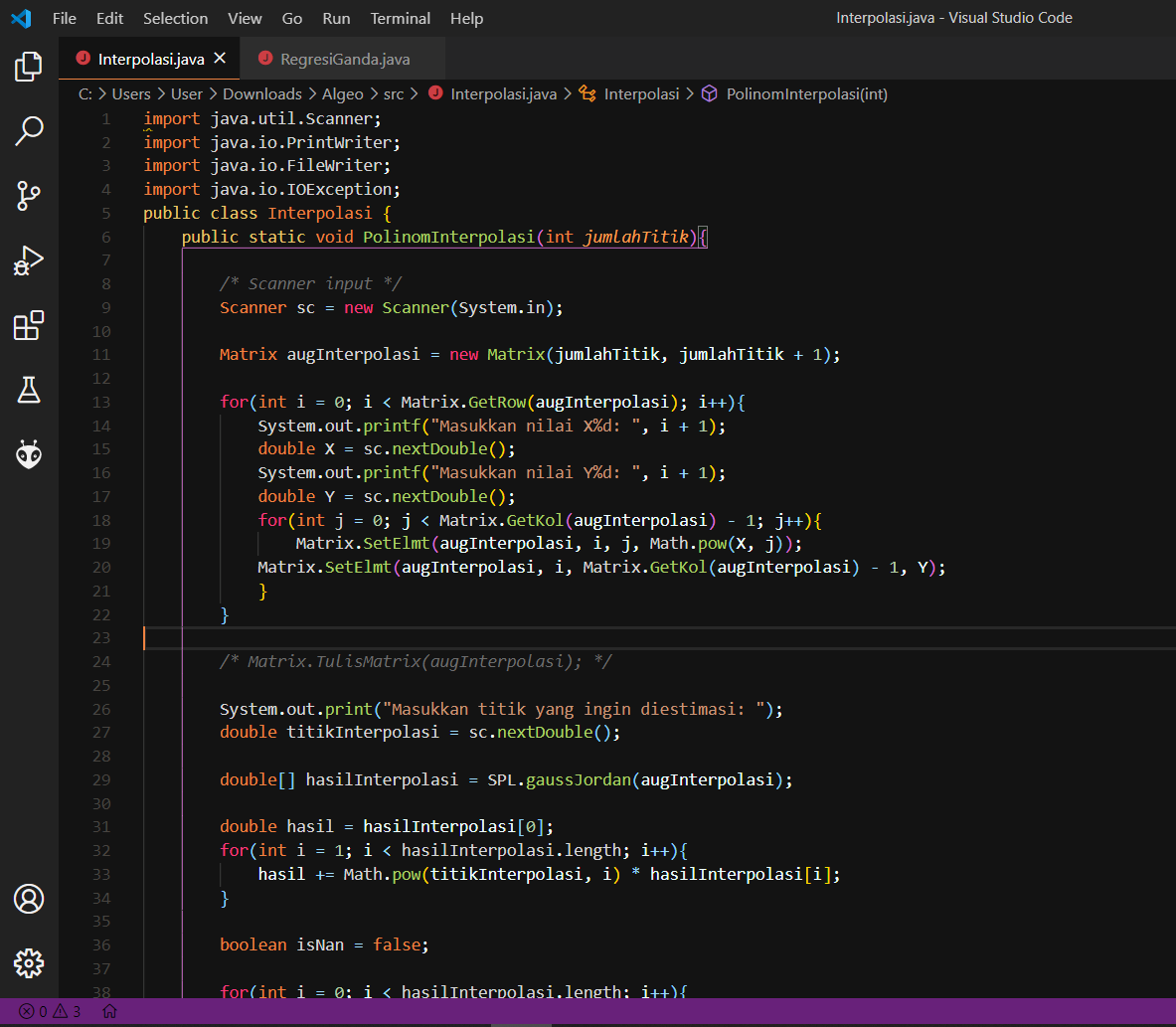
Menurut Montgomery, Peck, & Vining (2012), regresi linier berganda adalah pendekatan pencarian suatu nilai tertentu yang menggunakan banyak regressor. Seperti yang kita tahu, regresi linier adalah pendekatan pencarian suatu nilai menggunakan garis linier. Garis linier tersebut disusun berdasar letak titik yang kita masukan untuk dianalisis. Regresi linier berganda menggunakan banyak garis sebagai parameternya. Garis tersebut, karena juga direpresentasikan dalam polinom, dapat dimasukan kedalam suatu matriks yang dapat dicari solusinya menggunakan eliminasi Gauss-Jordan.

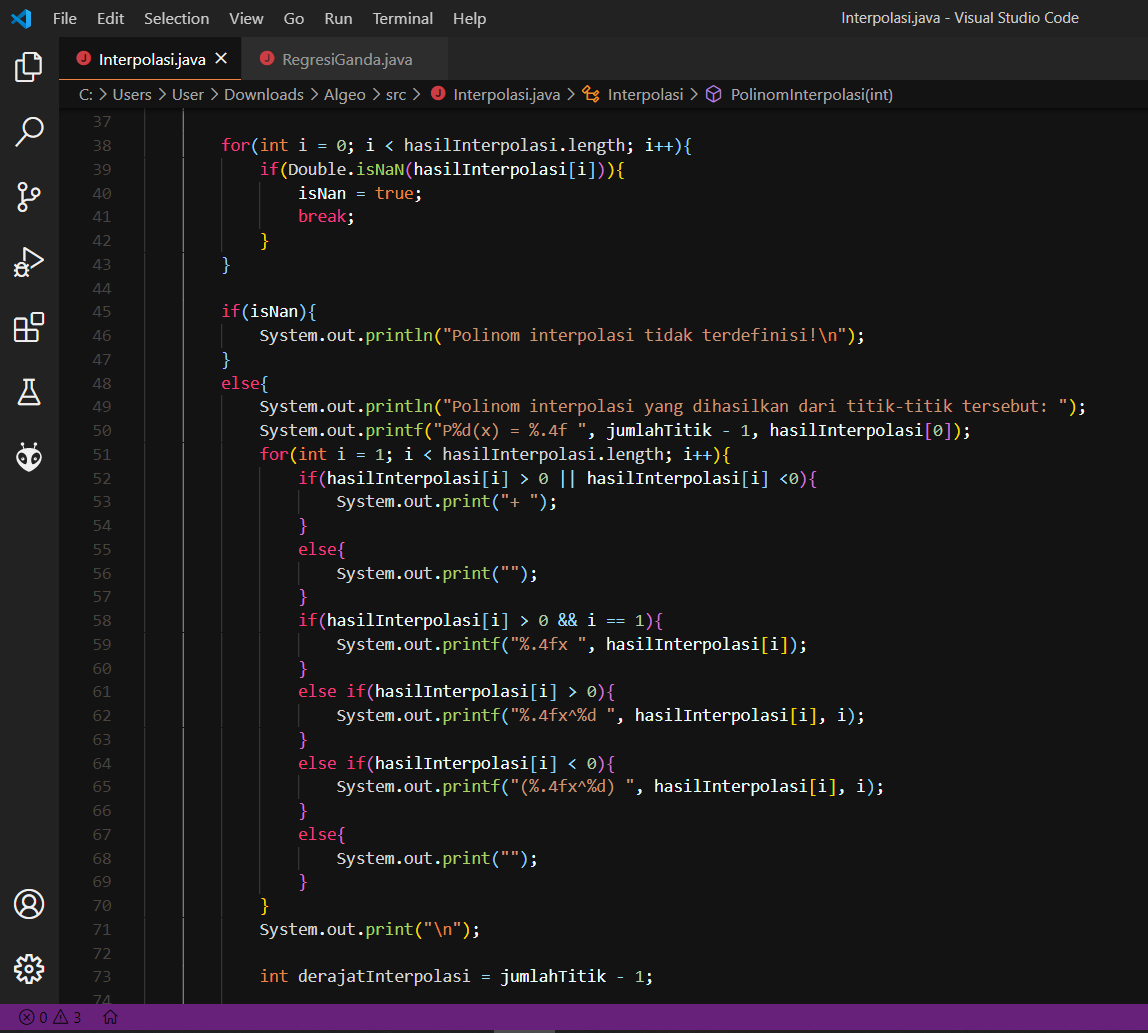
**BAB III**

**IMPLEMENTASI PROGRAM**

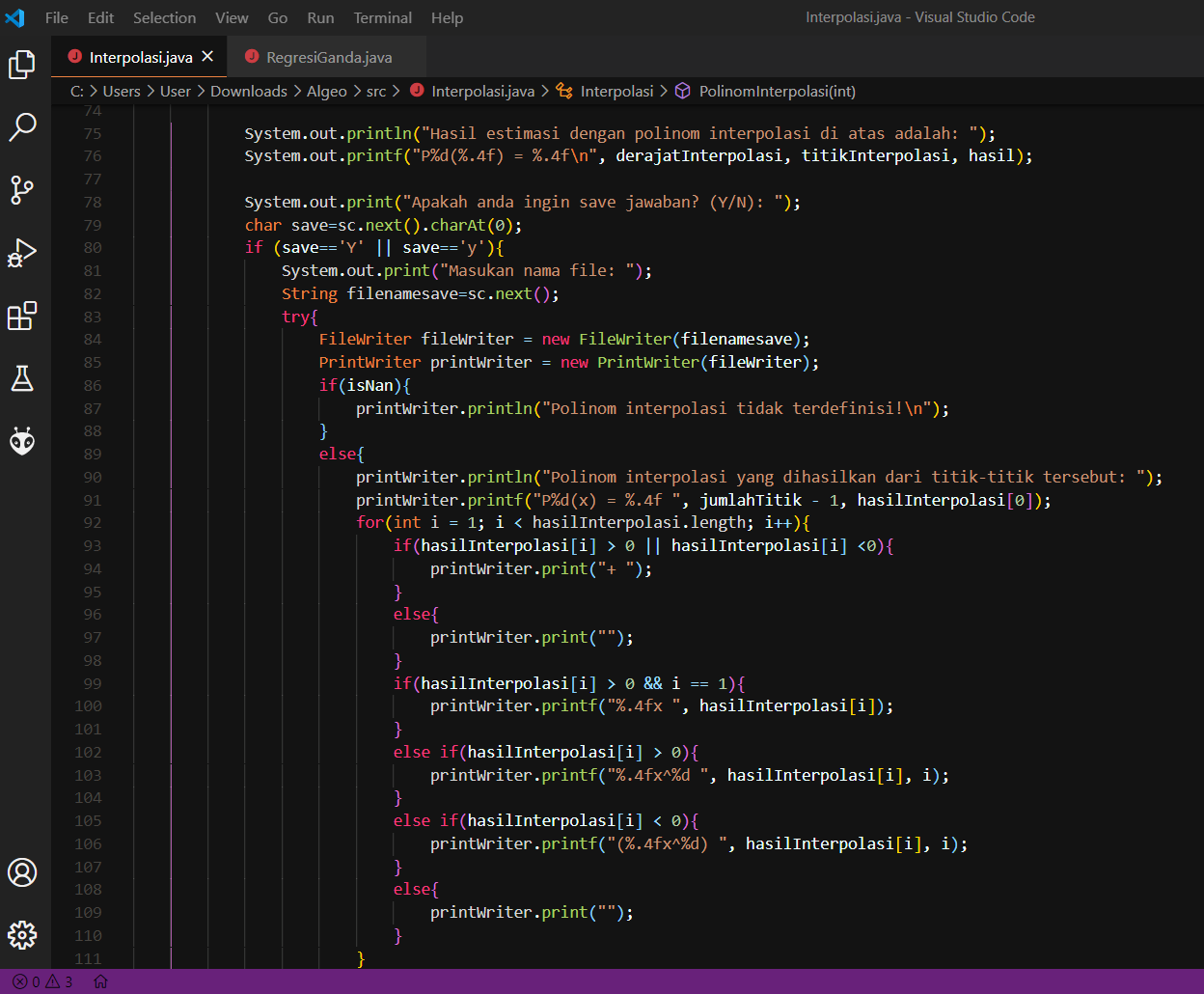
1. **Interpolasi**

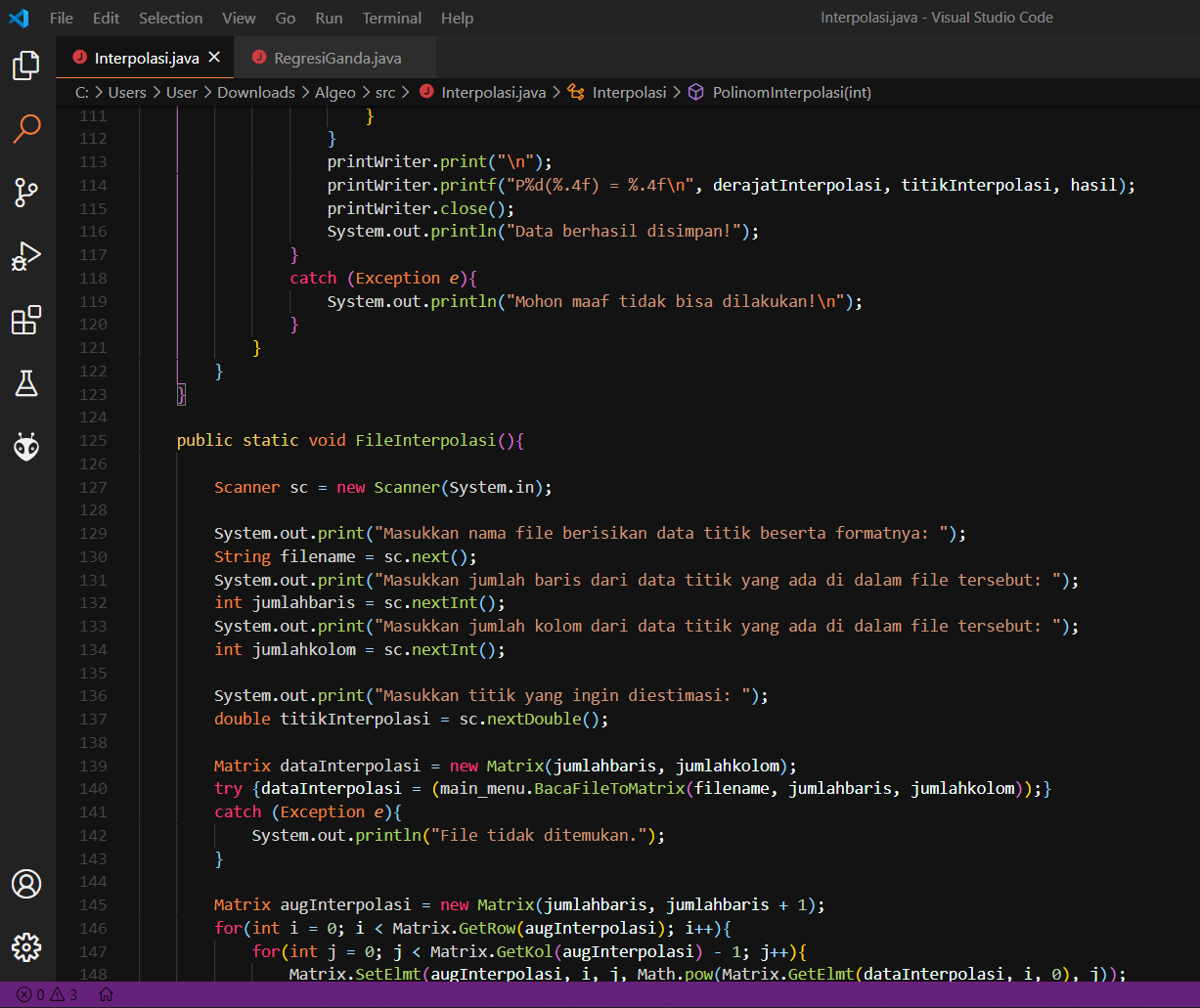
Berikut merupakan *screenshots* dari implementasi program terkait dengan interpolasi dalam bahasa Java:

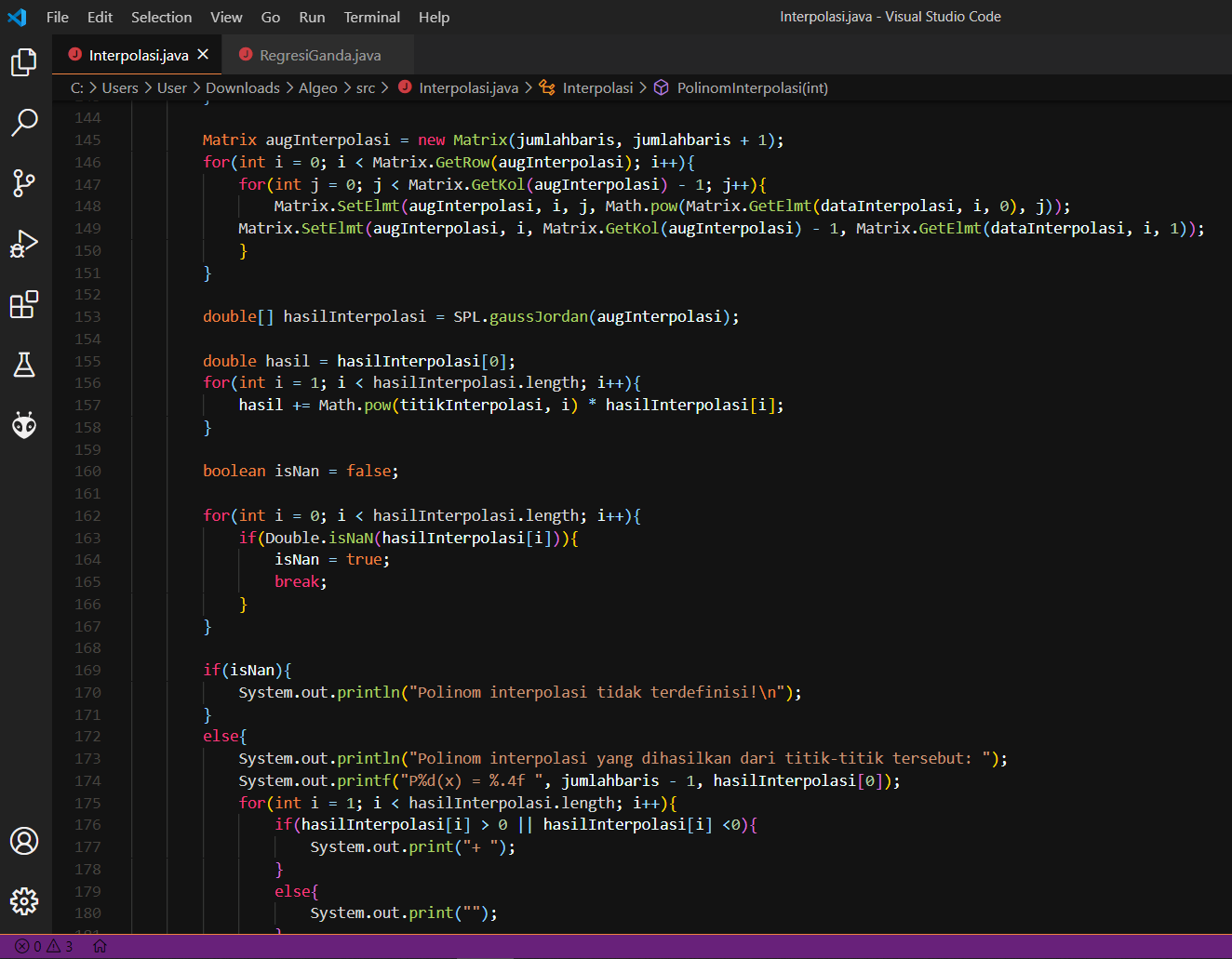


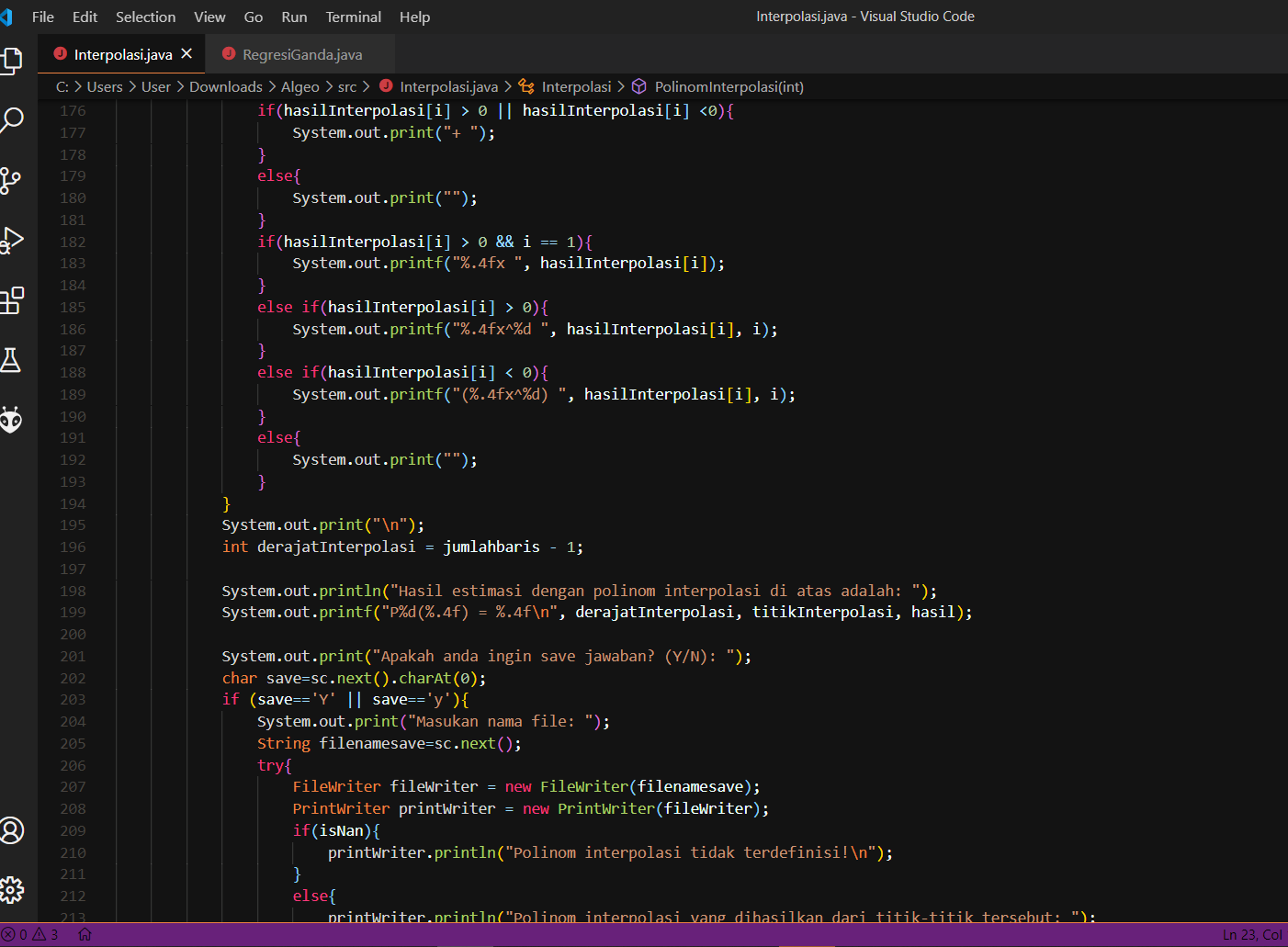


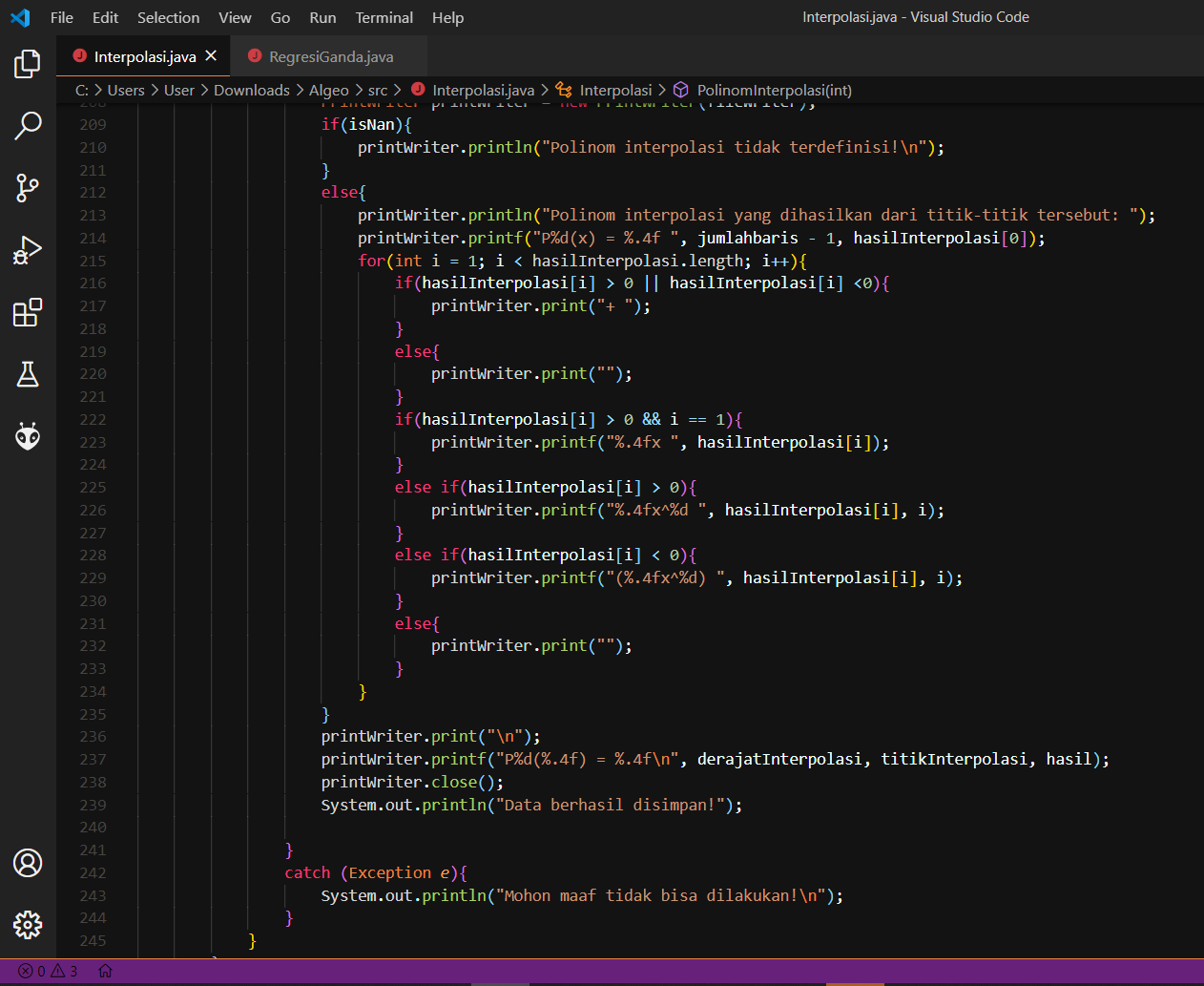
1. **Regresi Linier Berganda**

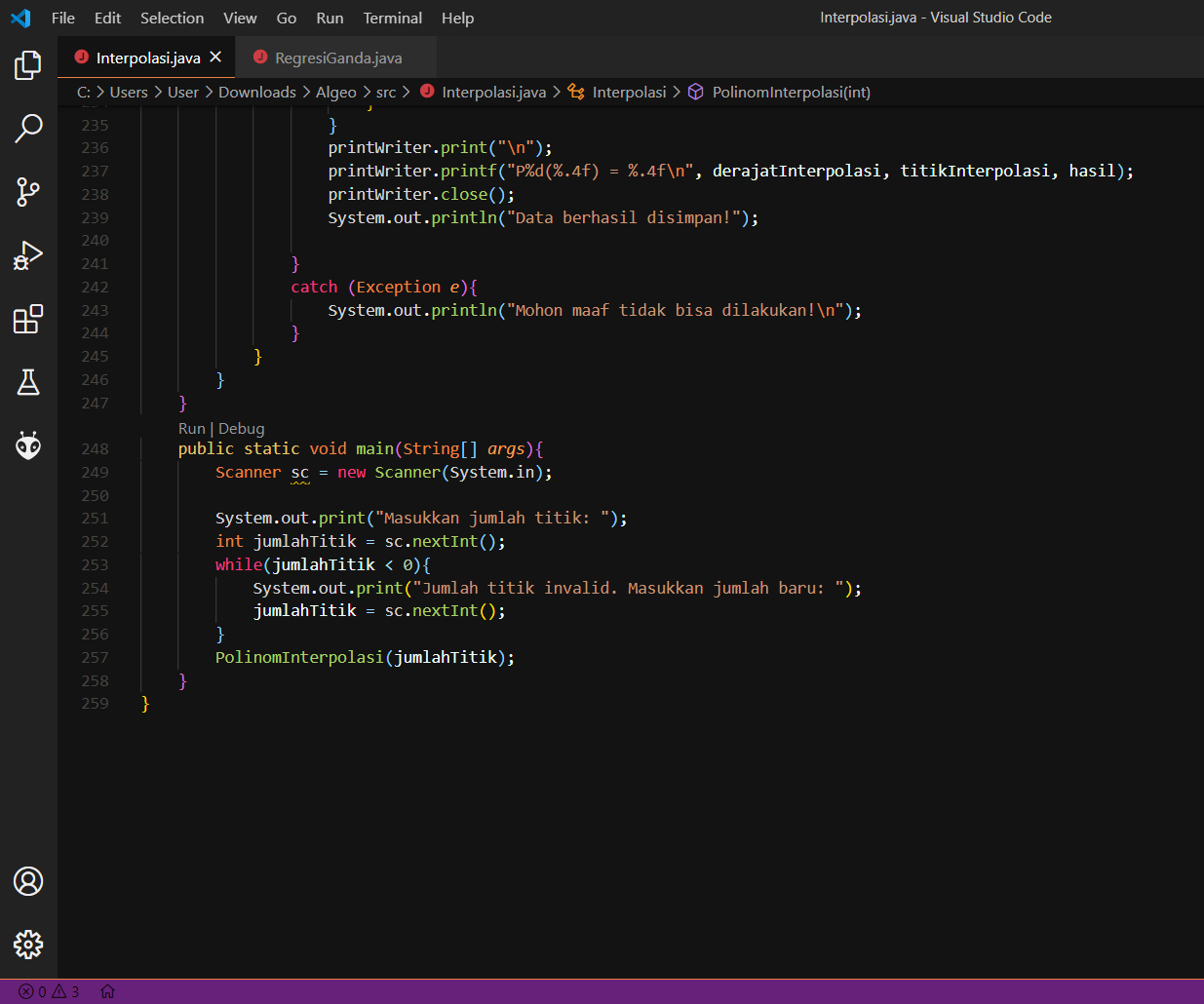
****

****

****

****

****

****

Pada bagian awal, program akan dilakukan *import* *library* yang dibutuhkan dalam program tersebut. Setelah itu, didefinisikan sebuah *class* bernama Interpolasi yang bersifat *public* sesuai dengan nama file yang dibuat*.* Di dalam *class* tersebut tidak terdapat atribut apapun dan hanya terdiri dari dua buah *method* dan sebuah *main*.

Bagian *main* di sini akan berperan untuk mengambil input jumlah titik yang akan digunakan untuk *method* interpolasi yang meneriima data dari *keyboard* pengguna. Program akan terus meminta pengguna untuk memasukkan input titik yang bernilai valid. Setelah itu, program akan memanggil *method* tersebut dan *method* tersebut akan berjalan sesuai dengan ketentuan berikut.

*Method* pertama bernama PolinomInterpolasi yang bersifat *public static* dan bertipe *void*. Bagian ini merupakan bagian yang telah disebutkan sebelumnya, yaitu melakukan operasi interpolasi dengan menggunakan input yang berasal dari *keyboard*. *Method* ini akan menerima sebuah parameter berupa integer yang berisikan jumlah titik yang akan diinput. Setelah itu, program akan membuat sebuah matriks *augmented* berdasarkan besar yang sesuai dengan jumlah titik yang telah diinput sebelumnya dengan menggunakan primitif matriks yang ada pada file Matrix.java. Isi dari matriks tersebut adalah titik-titik yang secara manual dimasukkan oleh pengguna, terurut dari absis dan kemudian ordinat.

Kemudian, program akan meminta pengguna untuk menginput sebuah titik yang ingin diestimasi. Titik ini akan diestimasi dengan menggunakan matriks yang dibuat yang berisi hasil interpolasi yang diselesaikan dengan menggunakan operasi SPL pada matriks *augmented* yang telah dibuat.

Setelah didapatkan sebuah hasil interpolasi, maka akan dicek apakah hasil tersebut bernilai NaN atau tidak. Jika bernilai NaN, maka program akan mengeluarkan pesan kesalahan dan polinom interpolasi tidak terdefinisi. Sebaliknya, program akan menampilkan persamaan polinom interpolasi dalam bentuk y = a + bx+ (-cx2) dan seterusnya sesuai dengan hasil yang didapatkan. Selain itu, juga akan ditampilkan hasil estimasi yang didapat oleh program dengan menggunakan titik yang telah dimasukkan tadi.

Setelah menampilkan output, program akan mengeluarkan prompt bagi pengguna untuk menyimpan output yang telah didapat ke dalam file atau tidak. Jika iya, maka file akan disimpan sesuai dengan input nama dan format yang diinput pengguna ke direktori program tersebut. Sebaliknya, jika tidak disimpan, program akan lanjut dan kembali ke bagian *main menu* dari program tersebut. Program akan melemparkan sebuah *exception* jika ada kesalahan saat melakukan hal tersebut.

*Method* kedua sendiri bernama FileInterpolasi, juga bersifat *public static* dan bertipe *void*. Bagian ini merupakan bagian program yang melaksanakan operasi interpolasi dengan input yang dibaca dari file yang dimasukkan oleh pengguna. Di sini, *method* ini tidak membutuhkan parameter apapun untuk bekerja. Pengguna akan diminta untuk memasukkan nama file beserta dengan formatnya, jumlah kolom dan jumlah baris dari matriks yang ada di dalam tersebut, serta titik yang ingin diestimasi oleh pengguna.

Kemudian, program akan mencoba untuk membaca isi dari file tersebut ke dalam bentuk matriiks dengan menggunakan fungsi BacaFileToMatrix yang ada di main\_menu.java. Jika berhasil, maka data tersebut akan disimpan ke dalam variabel matriks dan jika gagal, maka program akan mengeluarkan sebuah pesan *exception*.

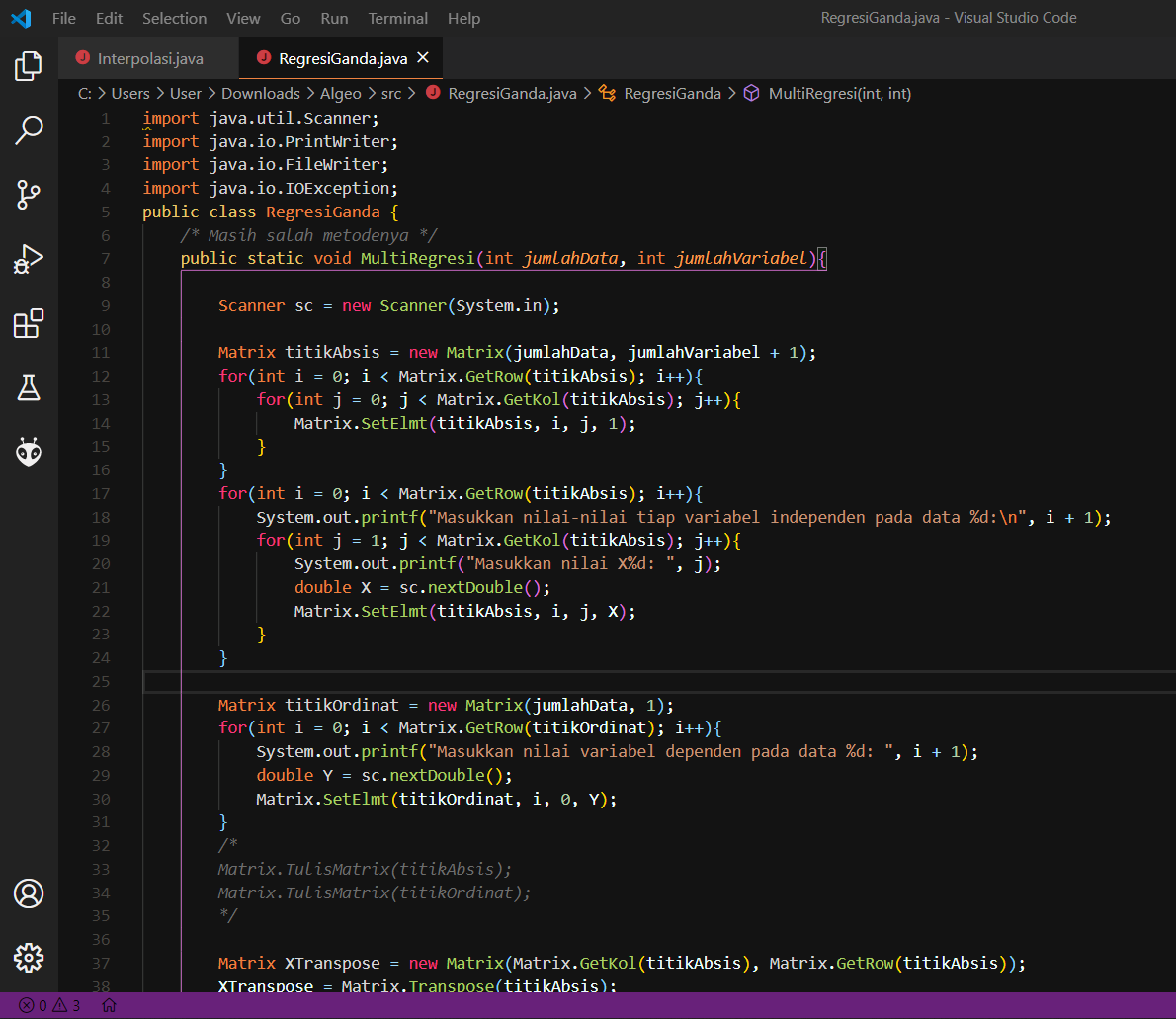
Segmen-segmen program selanjutnya sama dengan segmen program yang ada di *method* PolinomInterpolasi, yaitu melakukan kalkulasi hasil, pengecekan isNaN, penampilan output sesuai hasil pengecekkan tersebut, dan bagian program untuk menyimpan output perhitungan tersebut ke dalam sebuah file baru.

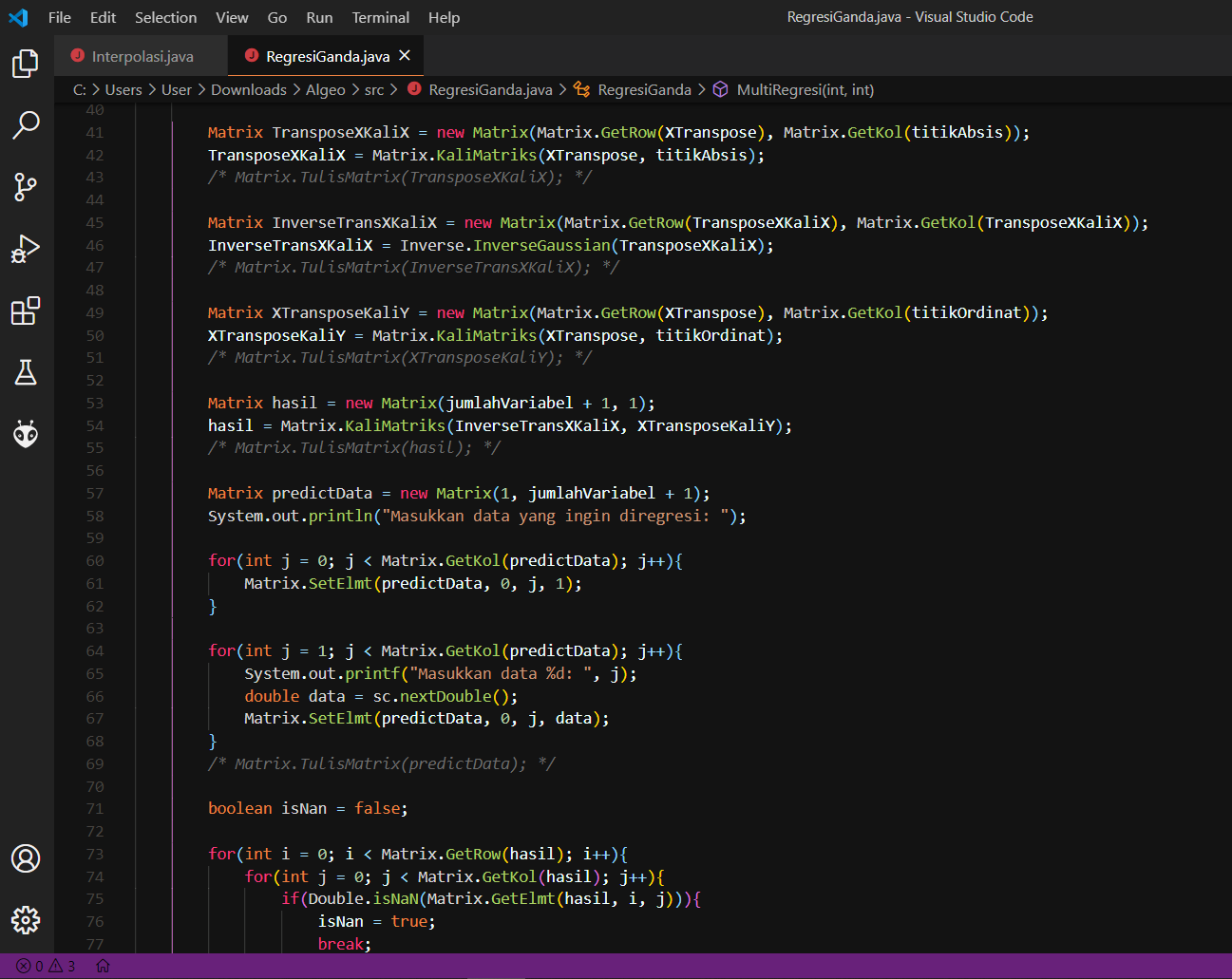
Catatan:

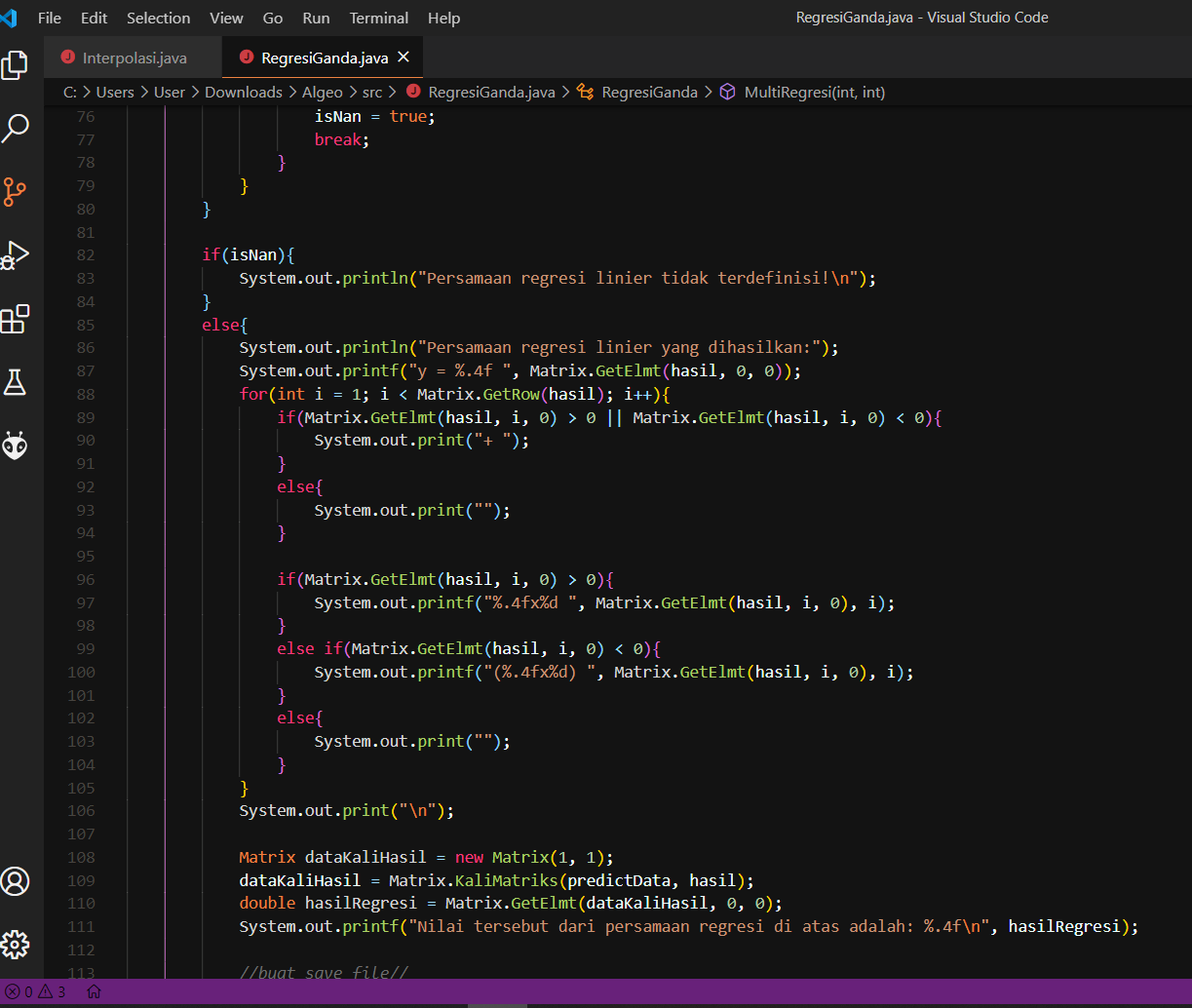
Matriks yang ada di dalam input file yang diberikan berbentuk matriks *augmented* yang terdiri dari titik-titik x dan y yang akan dikalkulasi.

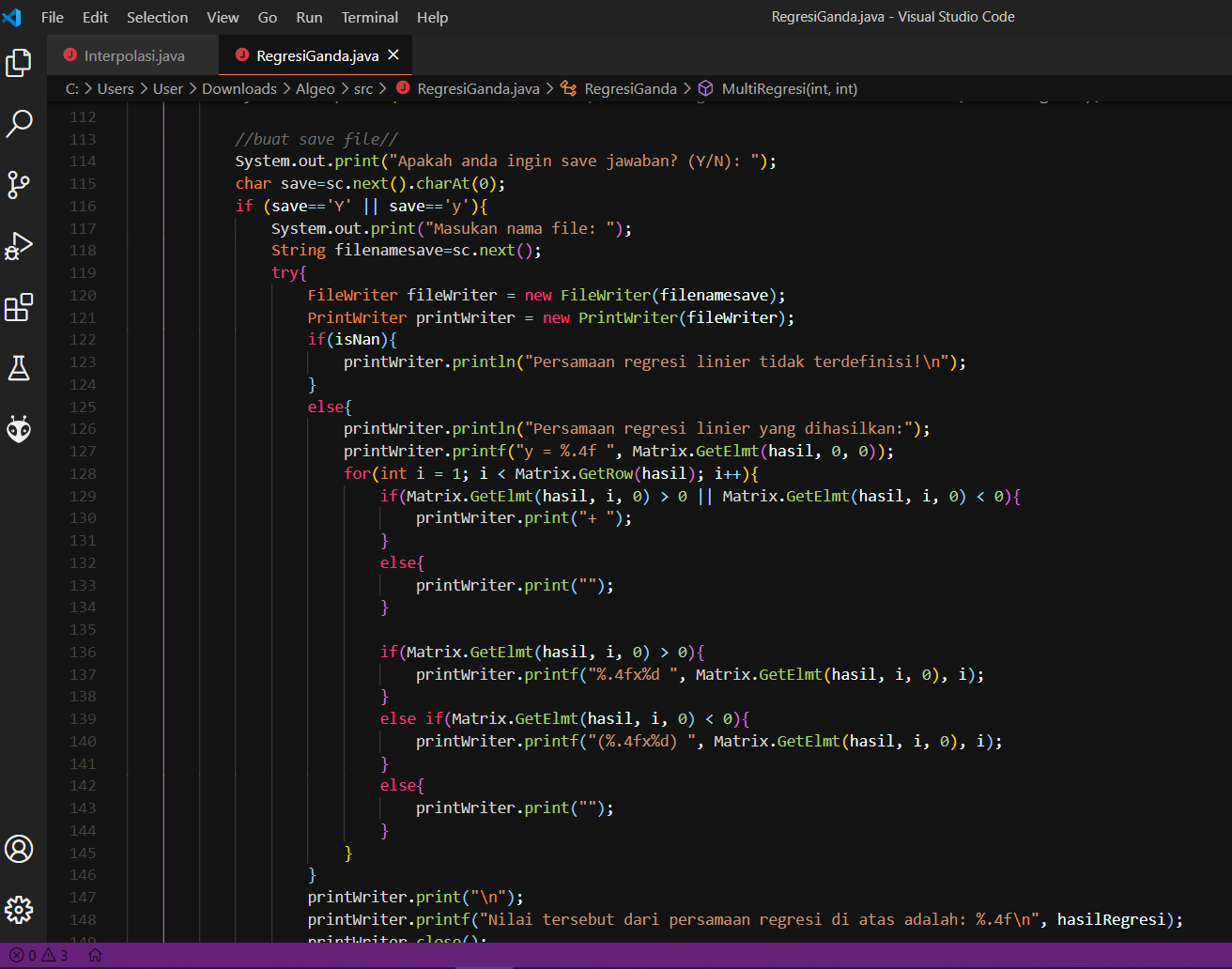
1. **Regresi Linier Ganda**

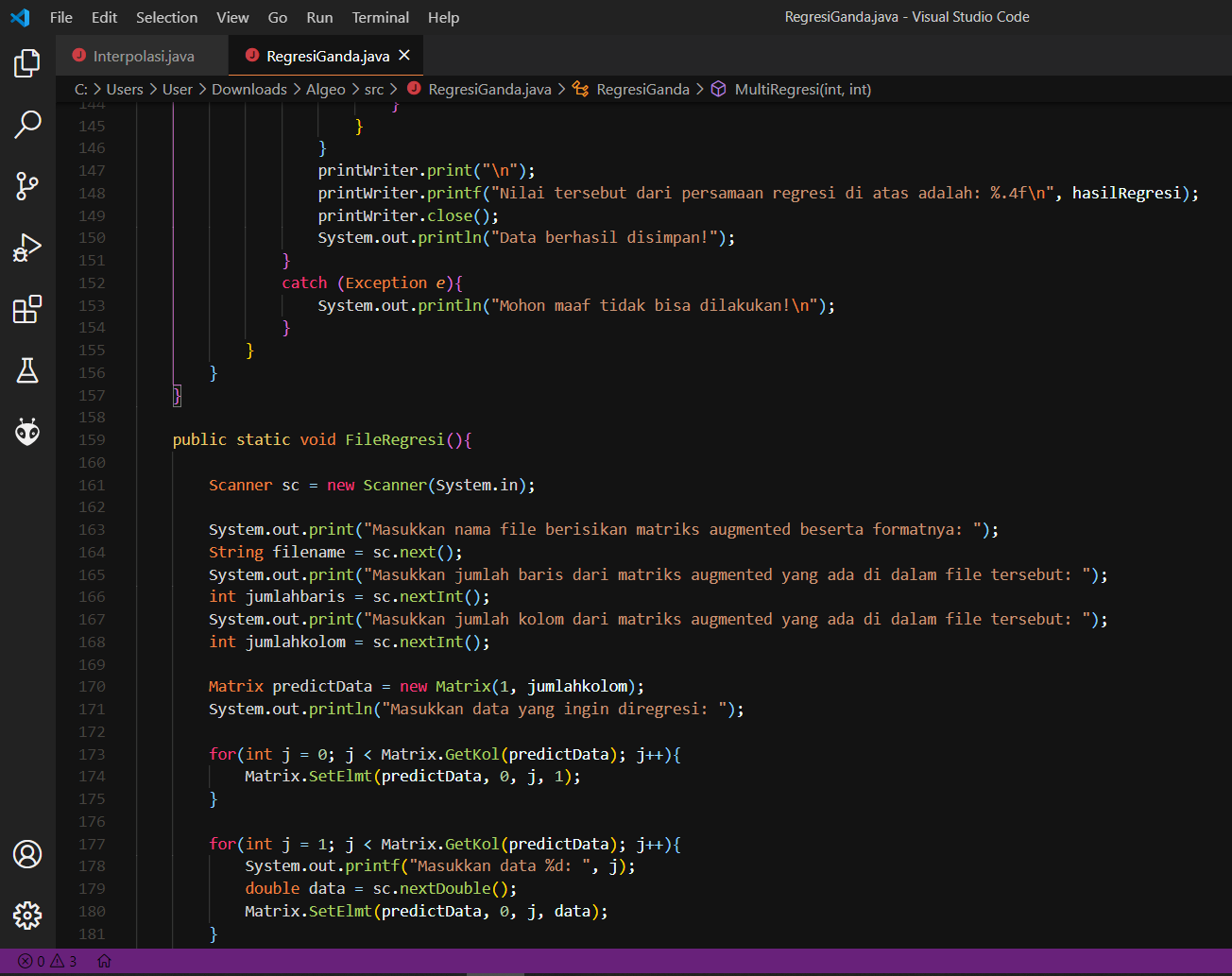
Berikut merupakan *screenshots* dari implementasi program regresi linier ganda dalam bahasa pemrograman Java:

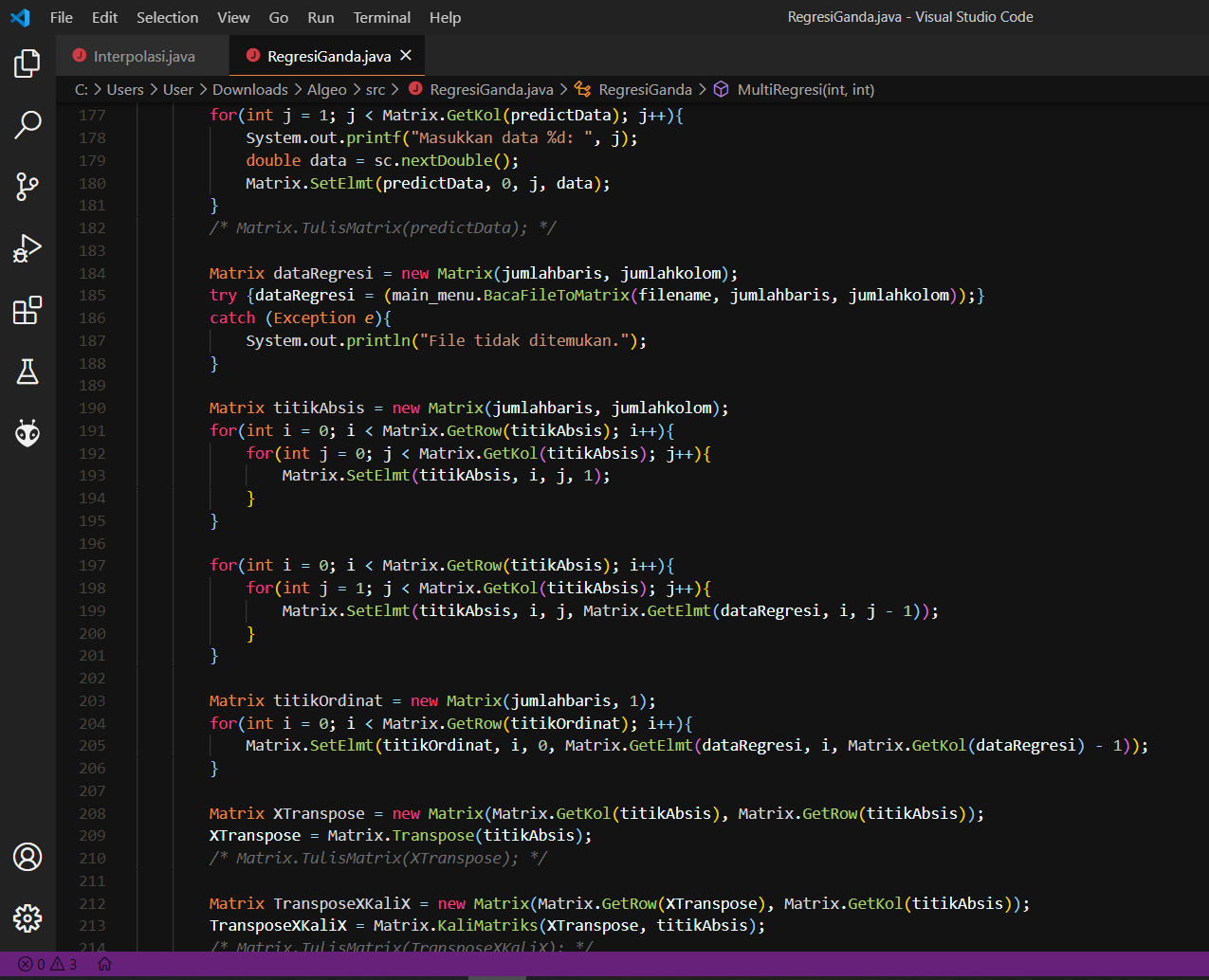


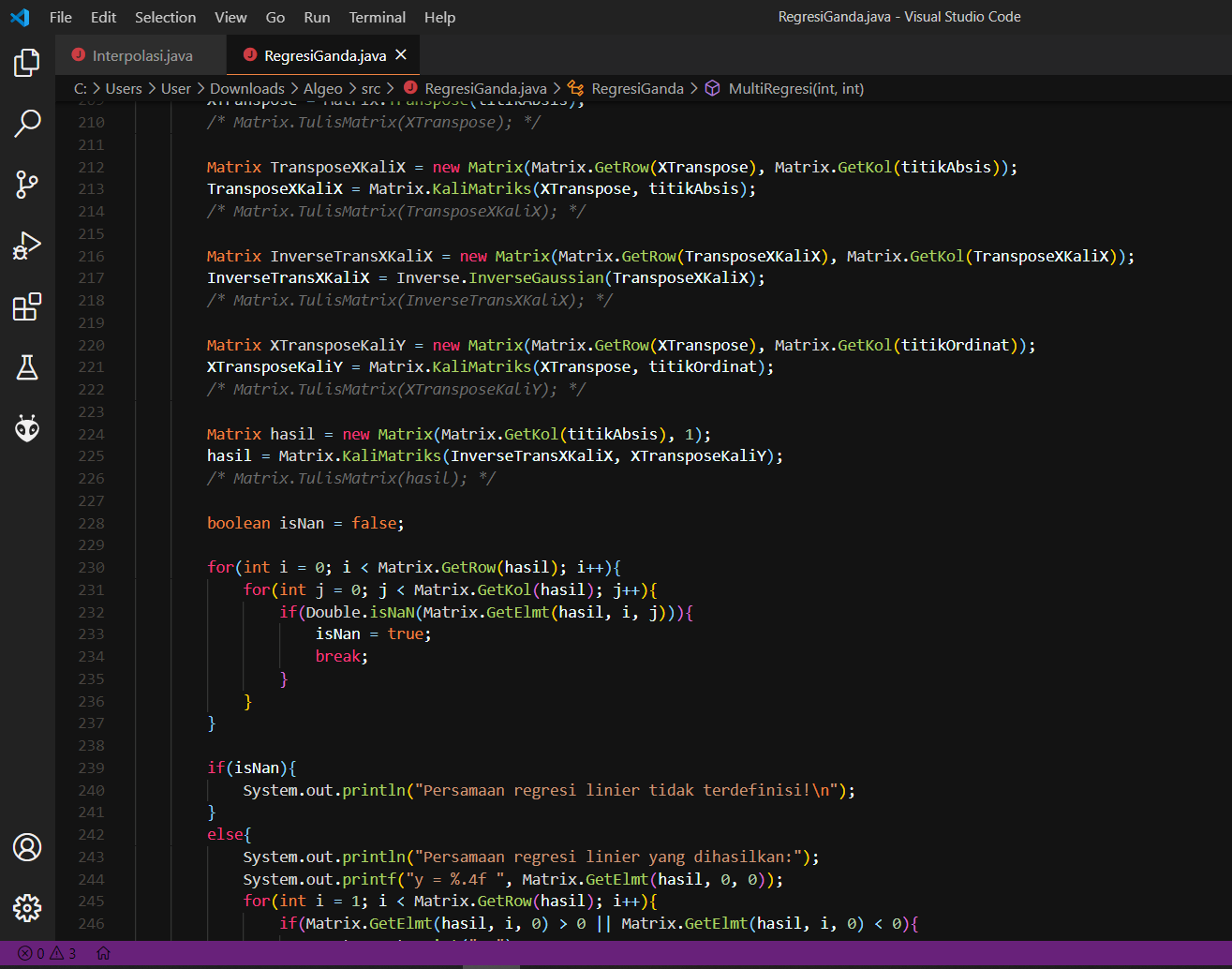
****

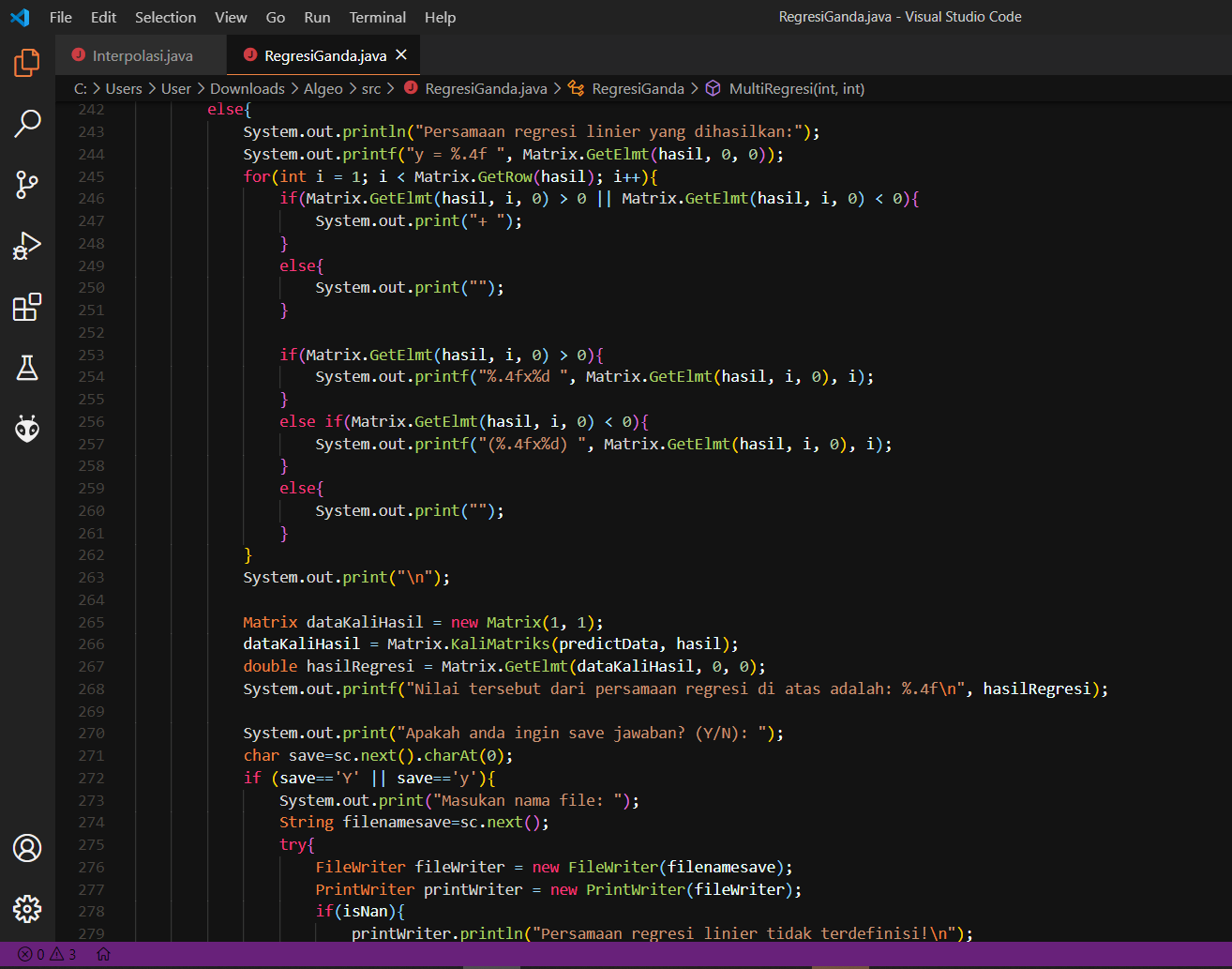
****

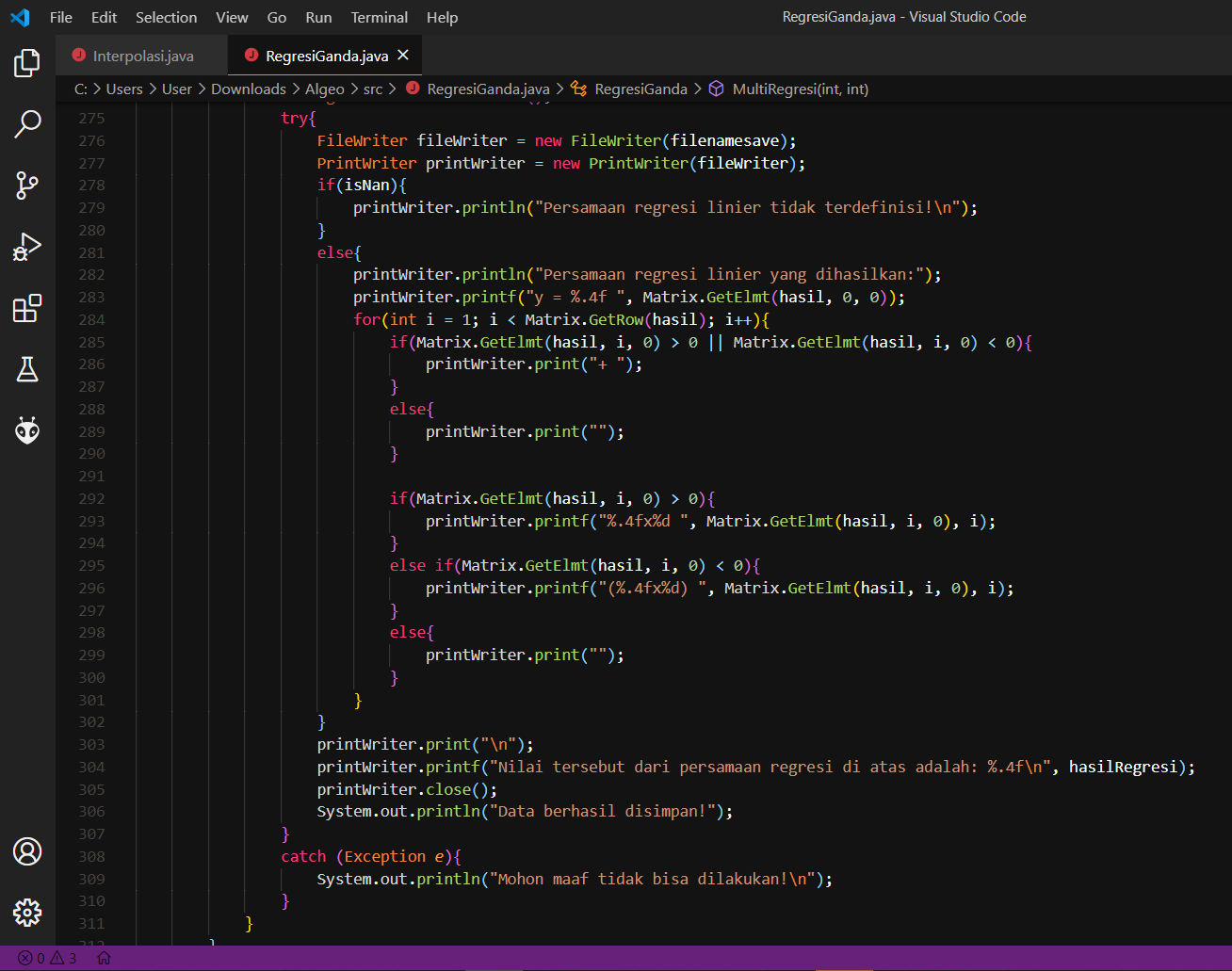
****

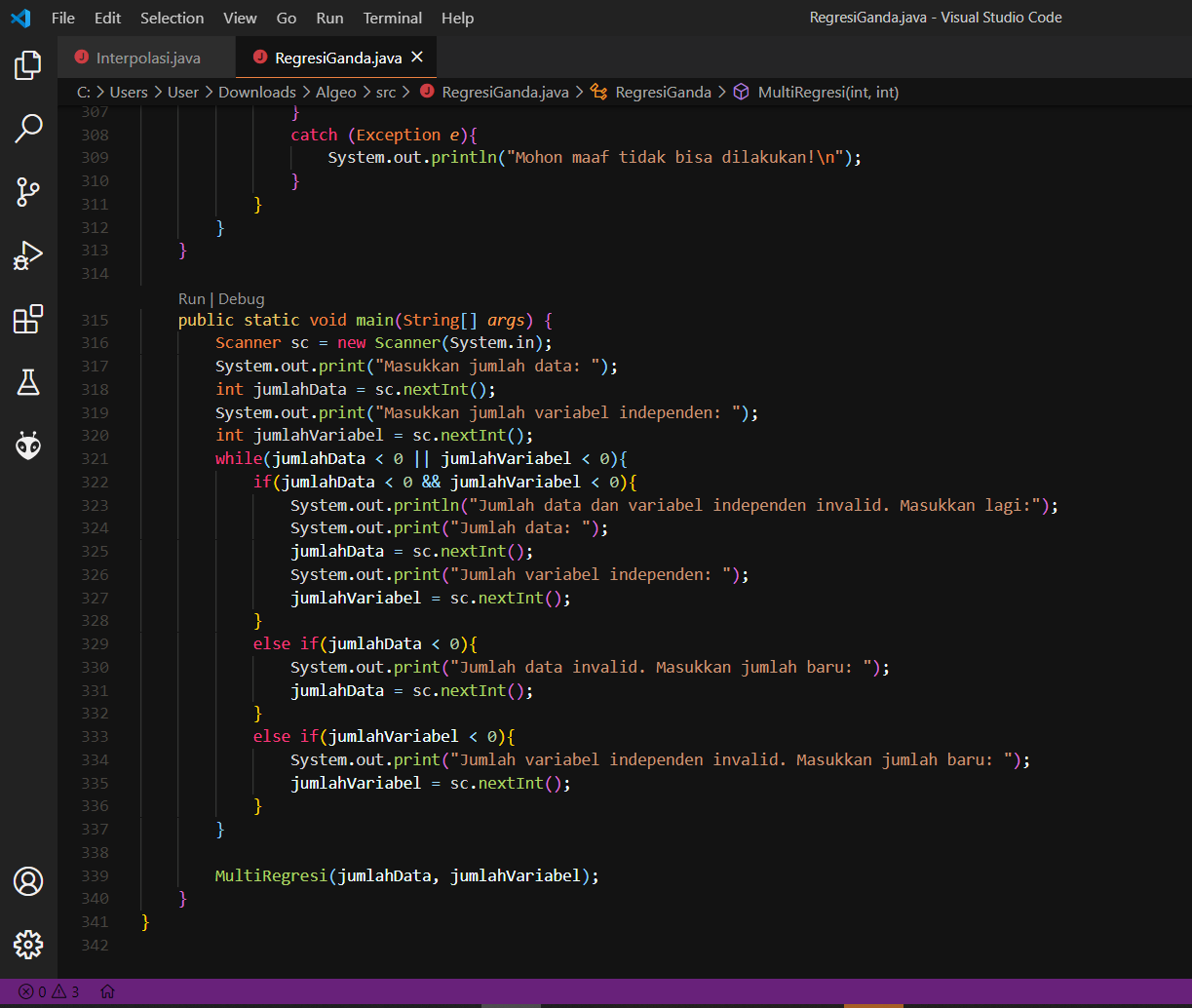
****

****

****

****

****

****

Sama dengan program Interpolasi.java sebelumnya, program RegresiGanda.java ini diawali dengan melakukan *import library* yang digunakan dalam pembuatan program tersebut. Setelah itu, maka akan didefinisikan sebuah kelas dengan sifat *public* yang bernama sama dengan nama file program, yaitu RegresiGanda. Dalam kelas ini sendiri tidak ada atribut apapun, tetapi mengandung dua buah *method* dan satu buah *main*.

*Main* dalam program ini akan meminta pengguna untuk memasukkan dua buah integer, yaitu jumlah data dan jumlah variabel independen dari data yang akan dimasukkan dan dikalkulasi. Jika salah satu atau kedua data yang dimasukkan tidak valid, maka akan dikeluarkan pesan yang bersesuaian dengannya. Jika tidak, maka program akan memanggil *method* untuk mencari persamaan regresi selanjutnya.

*Method* pertama memiliki sifat *public static* dan tidak mengembalikan nilai apapun, diberi nama MultiRegresi. *Method* ini akan melakukan operasi untuk mencari persamaan regresi linier dari input data yang dimasukkan pengguna melalui *keyboard* yang telah disebutkan di atas. Kedua input yang dimasukkan di program *main* itu juga akan digunakan sebagai parameter dari *method* ini.

Bagian pertama dari *method* ini adalah meminta input dari pengguna untuk memasukkan data berupa titik-titik absis dan titik-titik ordinat sebagai data variabel independen dan variabel dependen. Kedua input ini akan disimpan ke dalam variabel bertipe matriks yang sebelumnya telah dideklarasikan. Setelah itu, program akan masuk ke dalam bagian kalkulasi persamaan regresi yang didapat dari data tersebut.

Bagian kalkulasi ini sendiri tidak menggunakan SPL, melainkan menggunakan beberapa matriks perantara, yaitu matriks transpose, matriks invers, dan matriks hasil perkalian dari kedua komponen sebelumnya. Dengan mendeklarasikan sebuah matriks hasil untuk menyimpan hasil dari kalkulasi tersebut, maka di sini program menggunakan operasi (XTX)-1(XTY) untuk mendapatkan hasil tersebut.

Pengguna akan kemudian diminta untuk melakukan input data regresi satu persatu untuk tiap variabel independen yang akan diestimasi dengan persamaan regresi yang nantinya didapat. Input ini juga akan dimasukkan ke dalam matriks dengan ukuran yang disesuaikan dengan faktor-faktor sebelumnya, terutama jumlah variabel independen. Setelah input tersebut didapatkan, dengan menggunakan perkalian matriks kita bisa mendapatkan hasil regresi data tersebut dengan mengalikan matriks data regresi dengan matriks hasil yang sebelumnya telah didapatkan.

Setelah melakukan pengecekkan isNaN terhadap matriks yang dihasilkan oleh kalkulasi tersebut, program kemudian akan mengeluarkan output yang bersesuaian dengan hasil pengecekkan tersebut. Jika valid atau bukan NaN, maka output akan berupa persamaan regresi linier yang telah dikalkulasi dalam bentuk y = a + bx1 + (-cx2) dan seterusnya. Selain itu, program juga akan mengeluarkan output nilai regresi yang didapat dari perhitungan data regresi dan persamaan regresi tersebut. Jika tidak, maka akan diberikan pesan kesalahan dan data regresi tidak akan bisa dihitung.

Sama dengan file sebelumnya, pada *method* ini pengguna juga akan diberikan *prompt* jika ingin menyimpan output tersebut ke dalam file. Jika ya, pengguna akan dimintai nama file beserta formatnya dan mengeluarkan pesan berhasil jika output telah berhasil disalin ke dalam file tersebut. Jika tidak, maka program akan lanjut ke bagian *main menu*. Jika terjadi kesalahan dalam proses ini, program akan memberikan sebuah *exception* dan pesan kesalahan yang bersesuaian.

*Method* kedua juga bersifat *public statis* dan bertipe *void*, yaitu FileRegresi. Tidak seperti *method* sebelumnya, *method* ini tidak menerima parameter apapun agar bisa berjalan. Awalnya, pengguna akan diminta untuk memasukkan nama file, jumlah baris dan kolom dari matriks augmented yang ada di dalam file tersebut, serta data yang ingin diregresikan secara satu persatu untuk tiap variabel independen.

Setelah itu, bagian program yang selanjutnya juga terdiri dari segmen-segmen yang sama dengan bagian program dari *method* pertama ketika telah mendapatkan input data regresi. Segmen tersebut berupa kalkulasi persamaan regresi linier serta pembentukan matriks-matriksnya, pengecekkan isNaN untuk hasil tersebut, serta pengeluaran output sesuai dengan hasil pengecekkan yang didapat. Pengguna juga akan diberi *prompt* untuk memilih menyimpan output ke file baru atau tidak.

Catatan:

Matriks yang ada di dalam input file berupa matriks augmented yang berisi nilai variabel-variabel independen dan variabel dependennya.

**REFERENSI**

Anton, H., & Rorres, C. (2013). *Elementary linear algebra: applications version*. John Wiley & Sons.

*https://stattrek.com/multiple-regression/regression-coefficients.aspx?tutorial=reg.*

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to linear regression analysis* (Vol. 821). John Wiley & Sons.

Rao, C. R., & Mitra, S. K. (1972). Generalized inverse of a matrix and its applications. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics*. The Regents of the University of California.