**LAPORAN PRAKTIKUM**

**“Post Test Pertemuan ke-II”**

Diajukan untuk memenuhi salah satu praktikum Mata Kuliah Matematika Diskrit yang di ampu oleh:

Nur Rochmah Dyah PA, S.T., M.Kom.



Disusun Oleh:

Mohammad Farid Hendianto 2200018401

**UNIVERSITAS AHMAD DAHLAN**

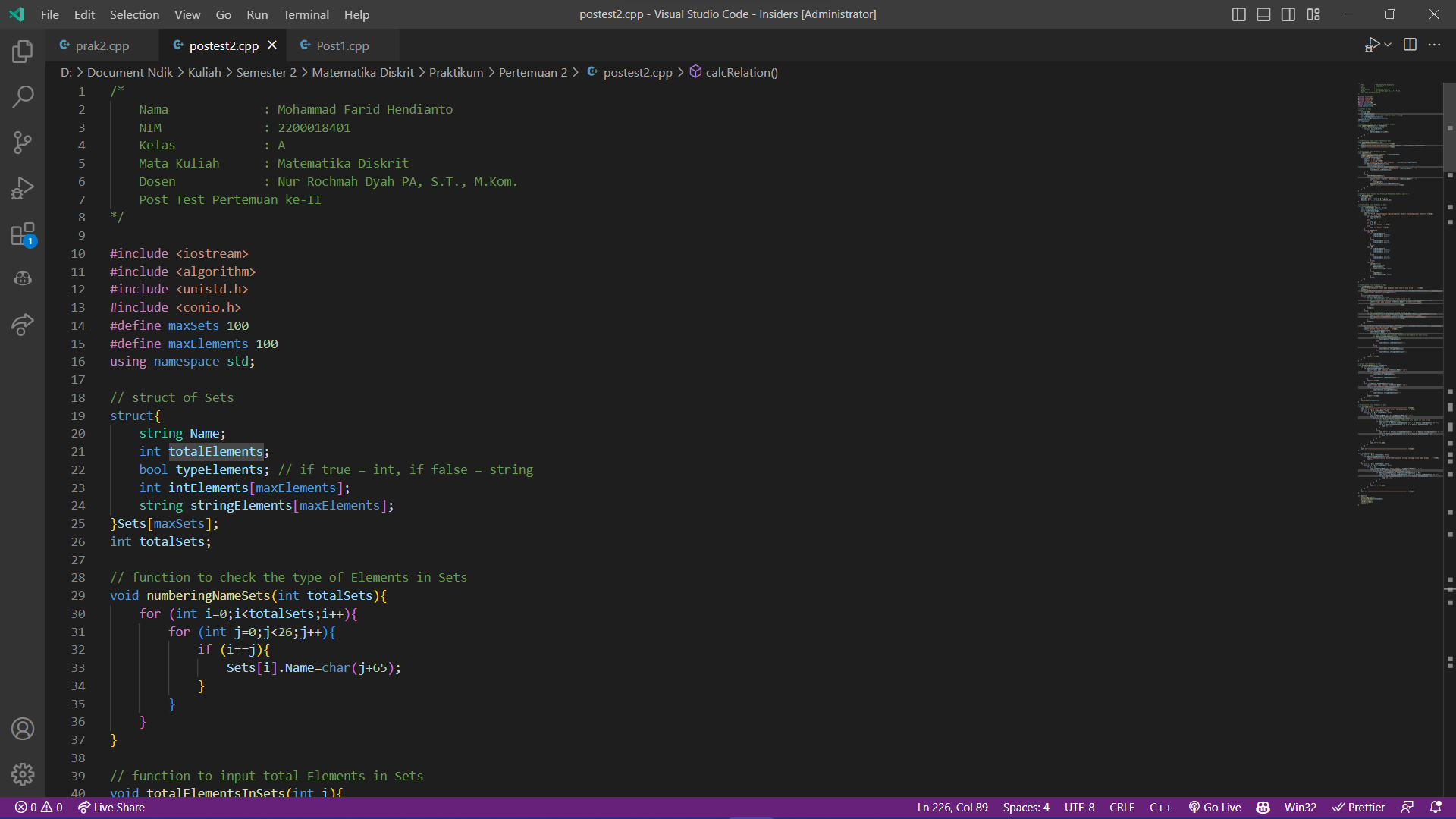
**FAKULTAS TEKNOLOGI INDUSTRI**

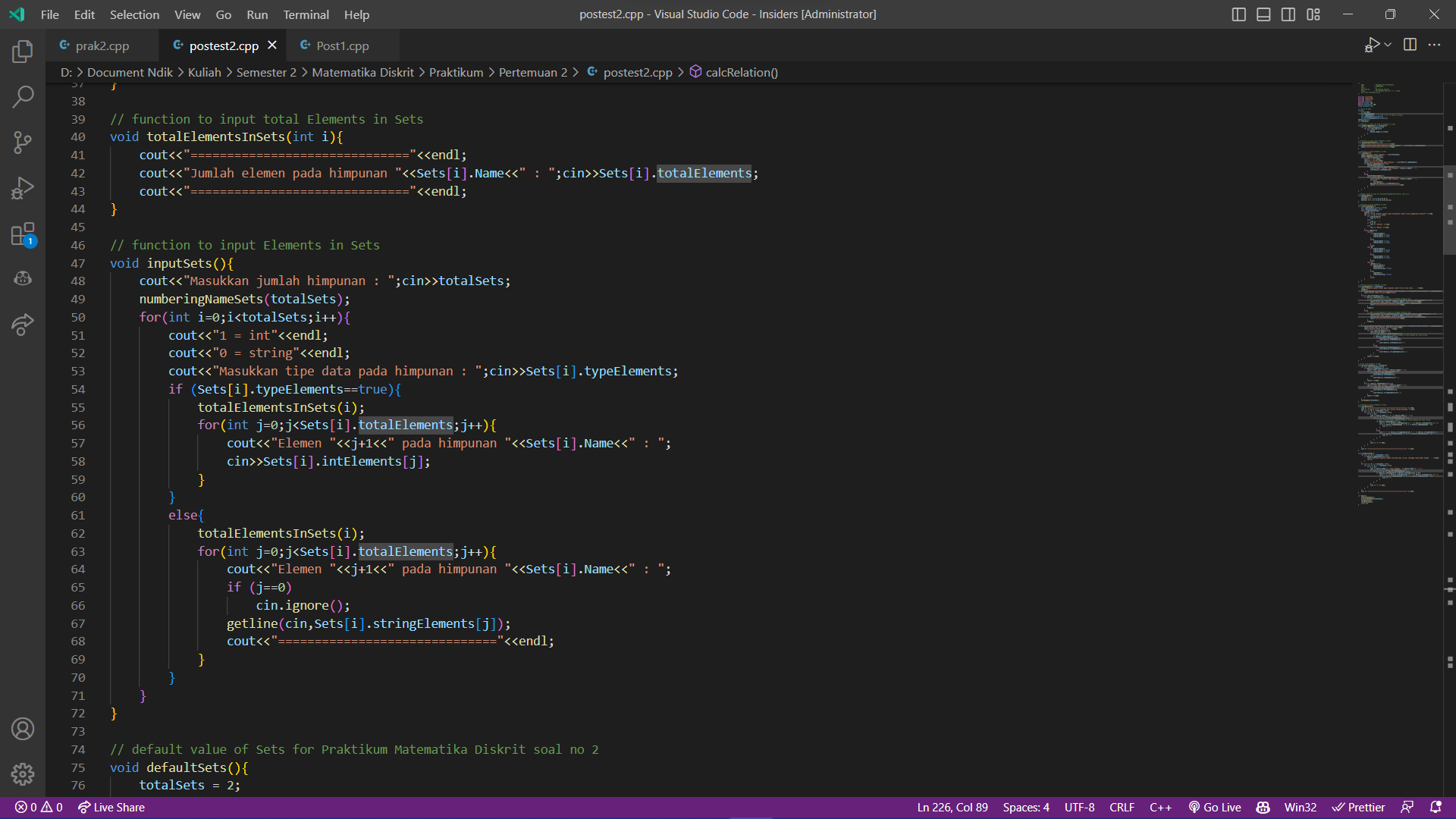
**PROGRAM STUDI INFORMATIKA**

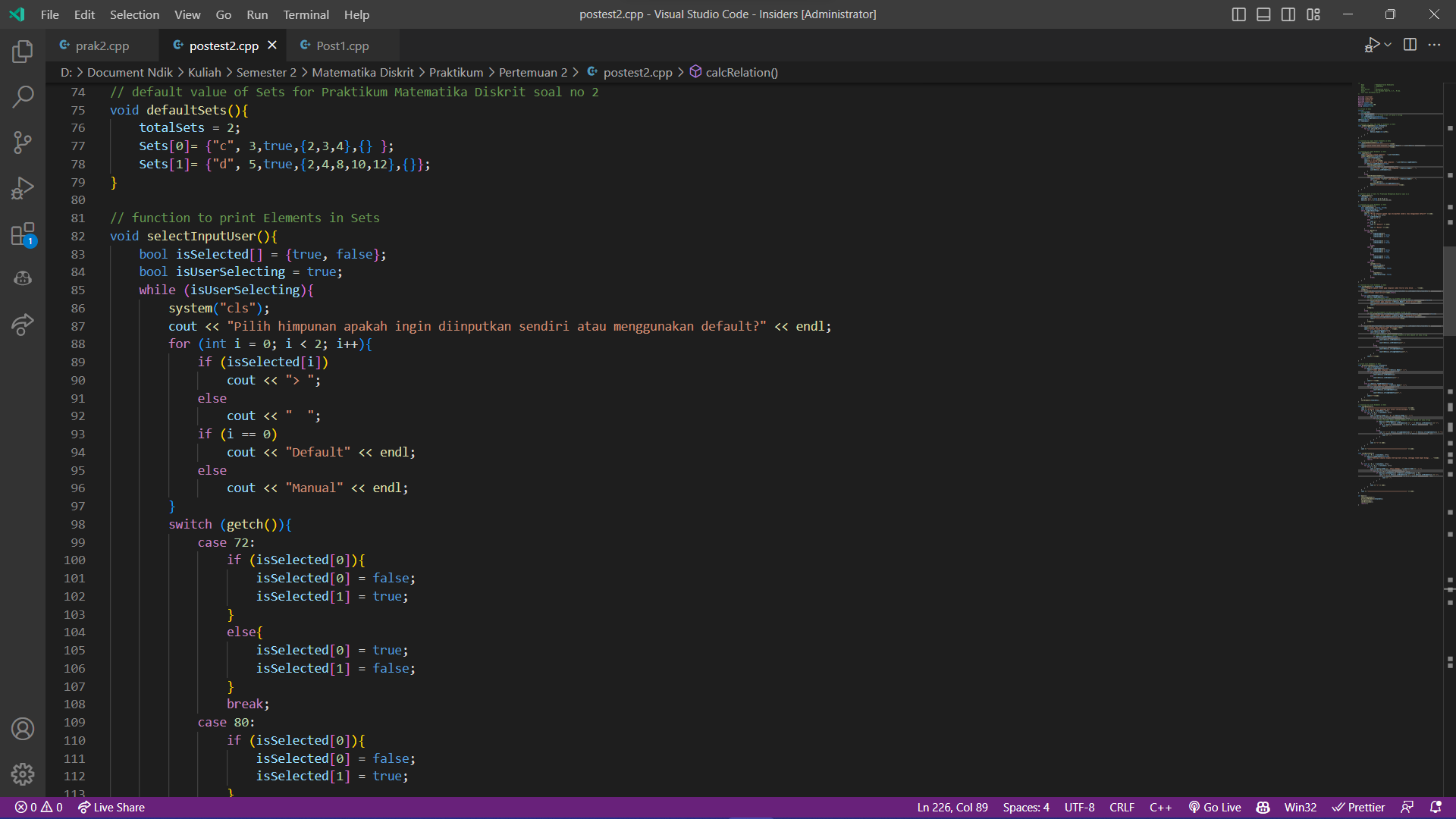
**TAHUN 2023**

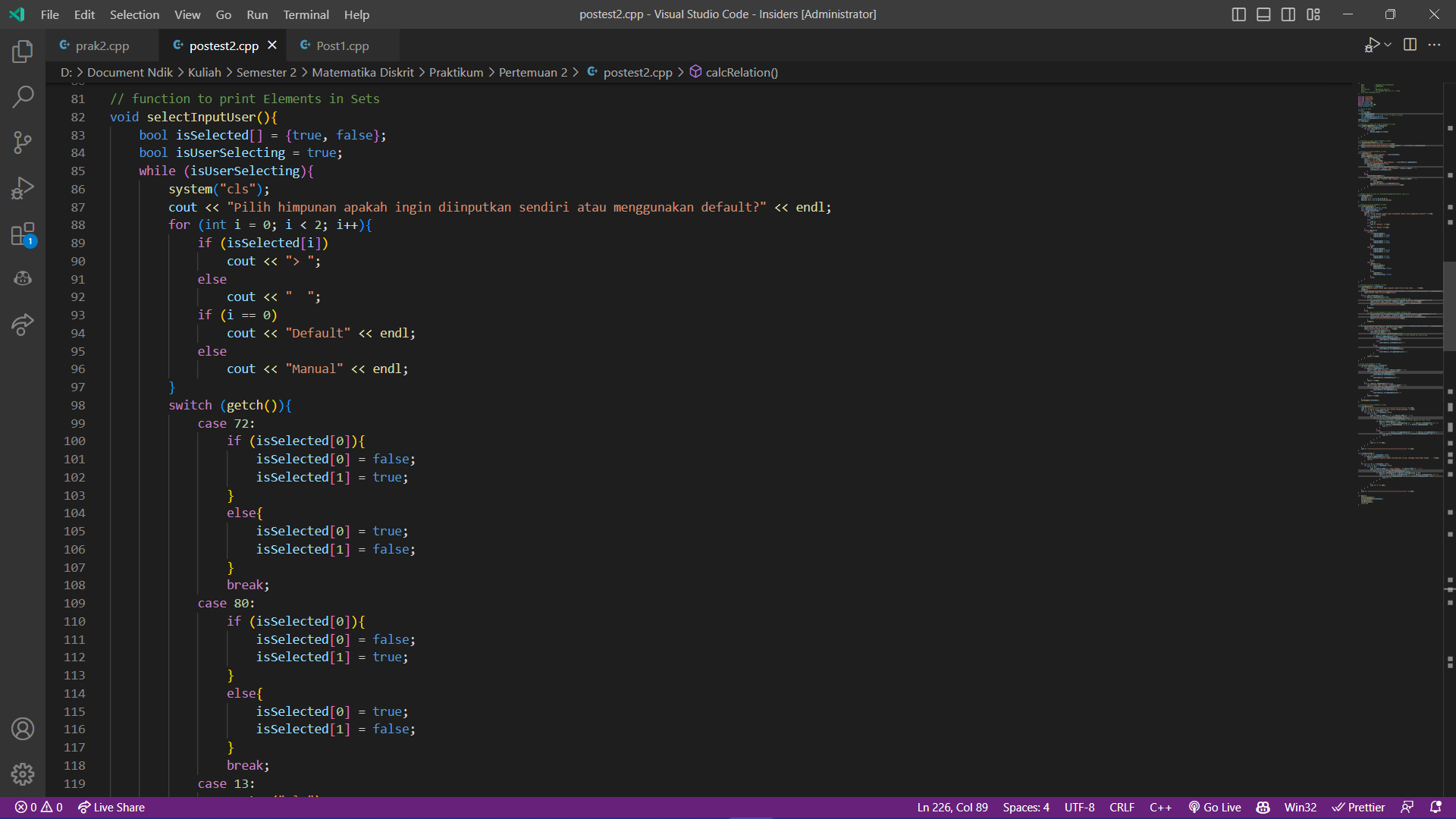
**Modifikasi kodingan materi supaya menjadi inputan dinamis!**

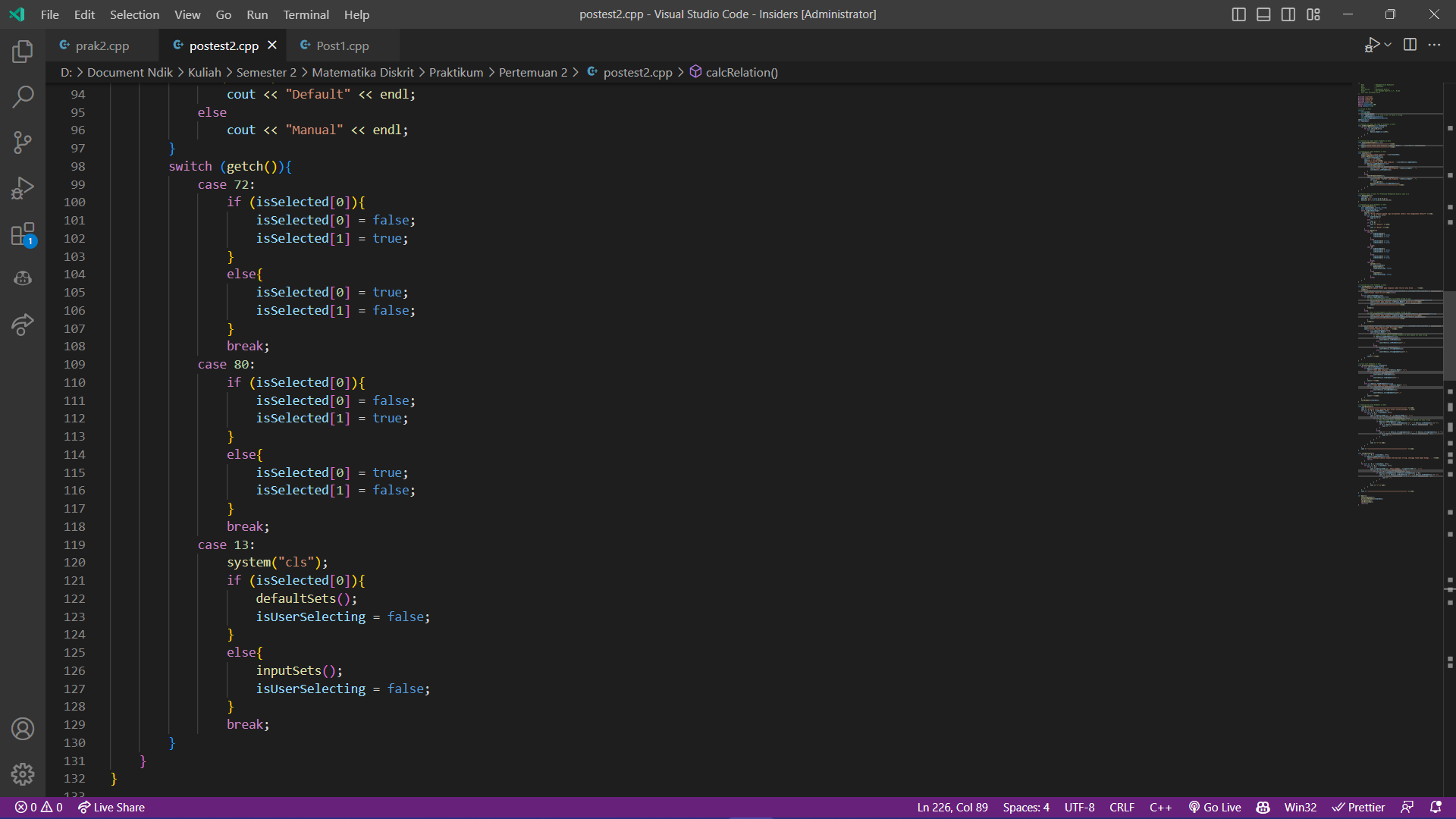
Berikut hasil modifikasi kodingan materi di Visual Studio Code

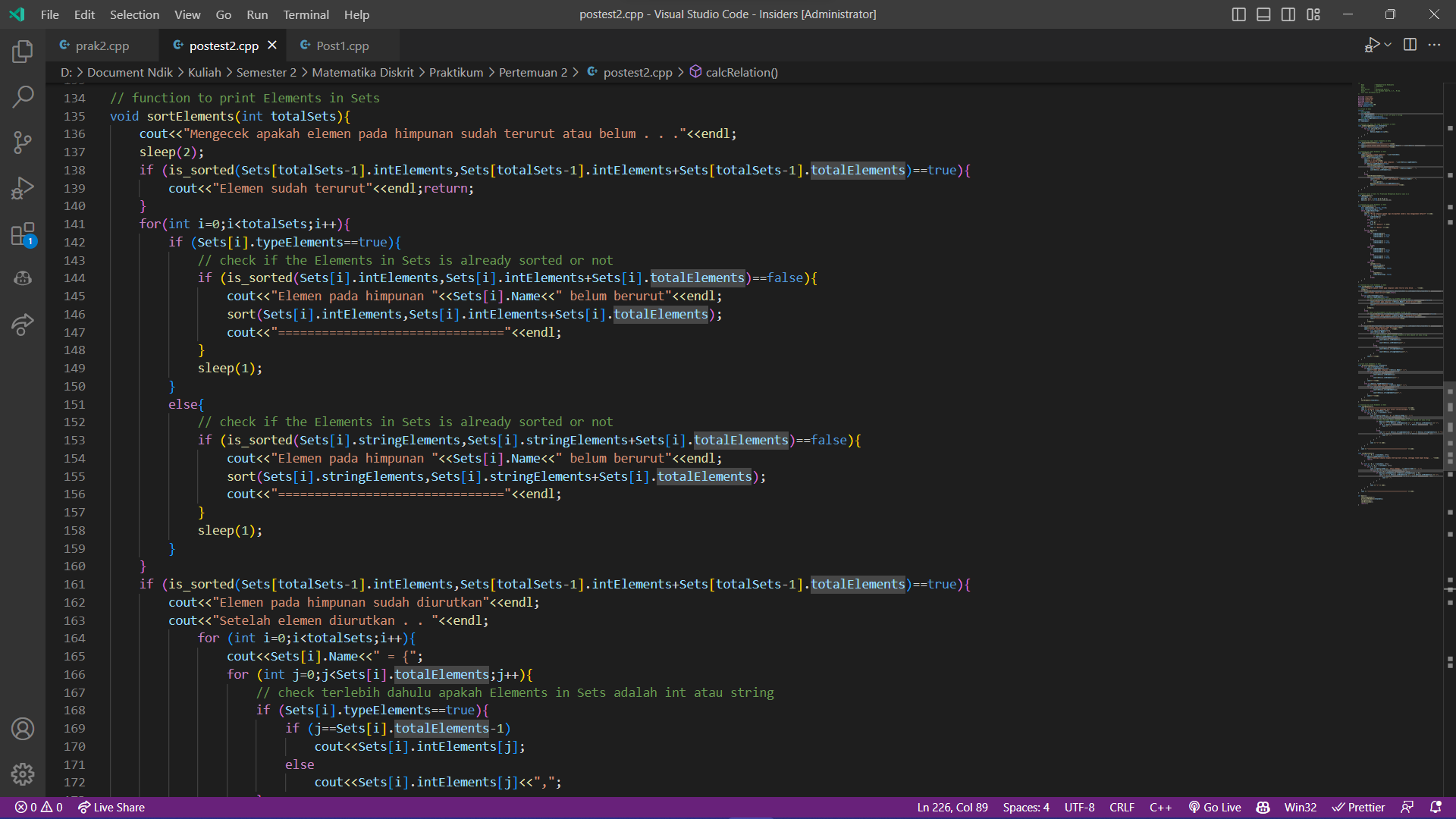


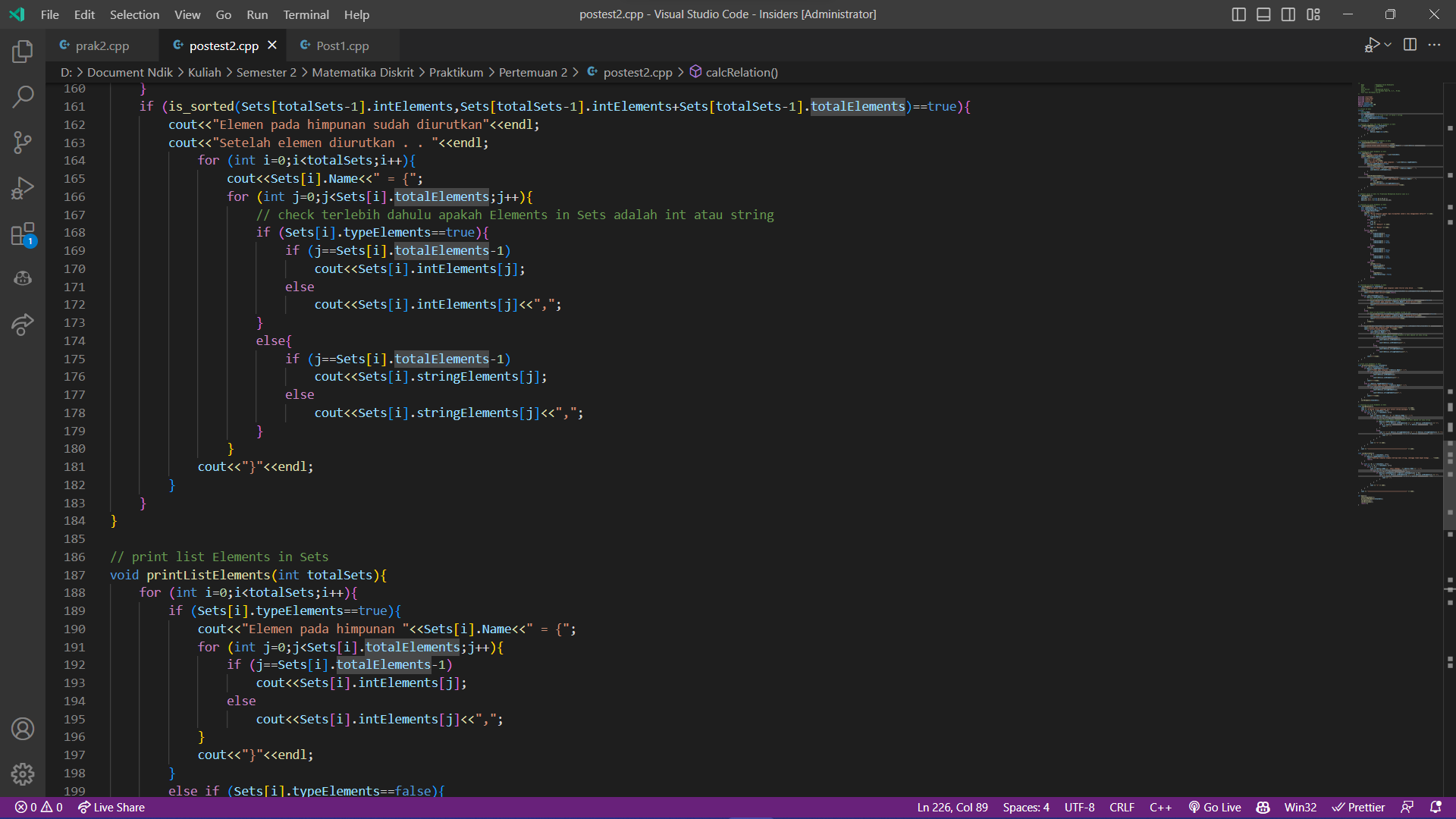


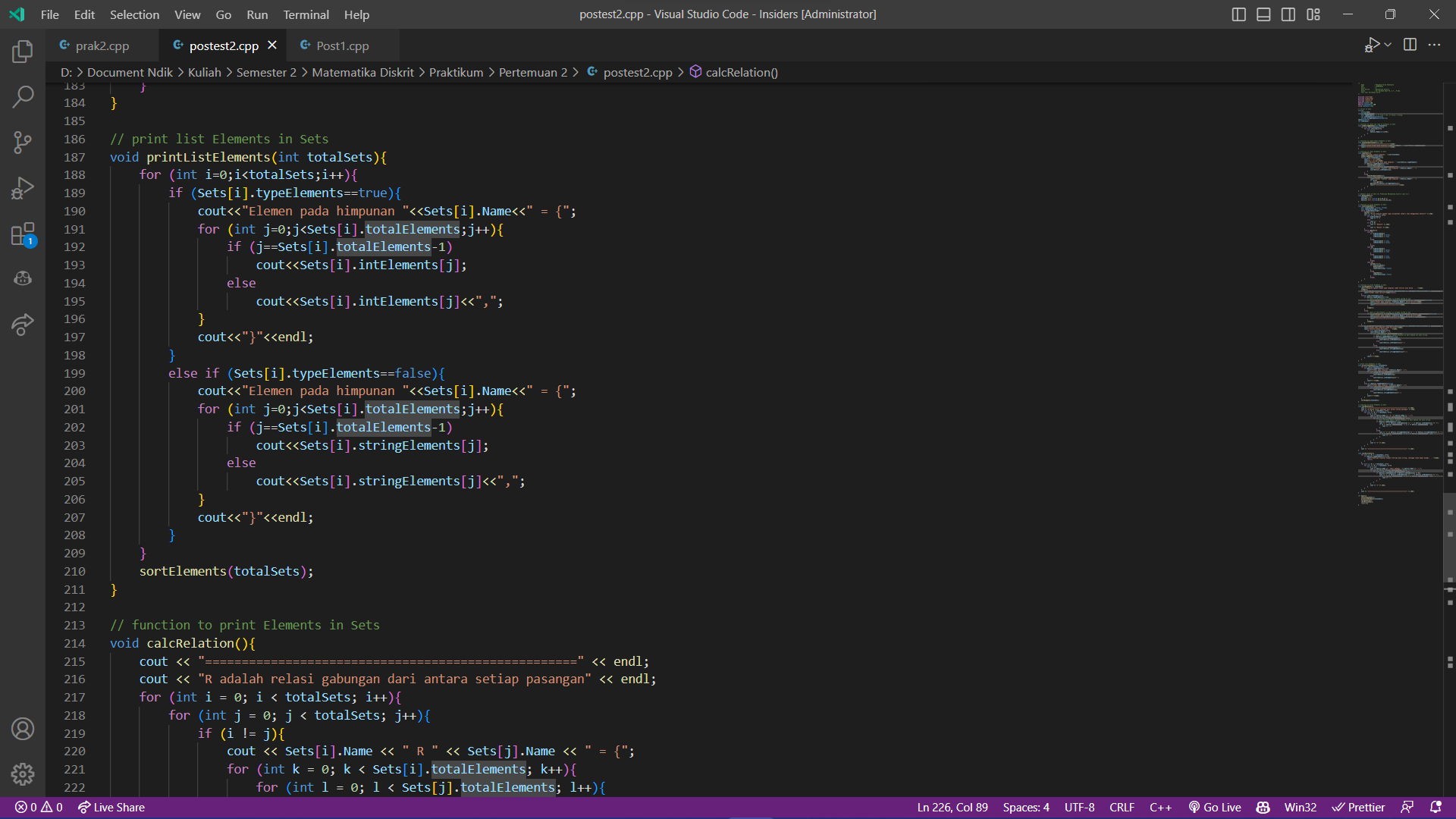


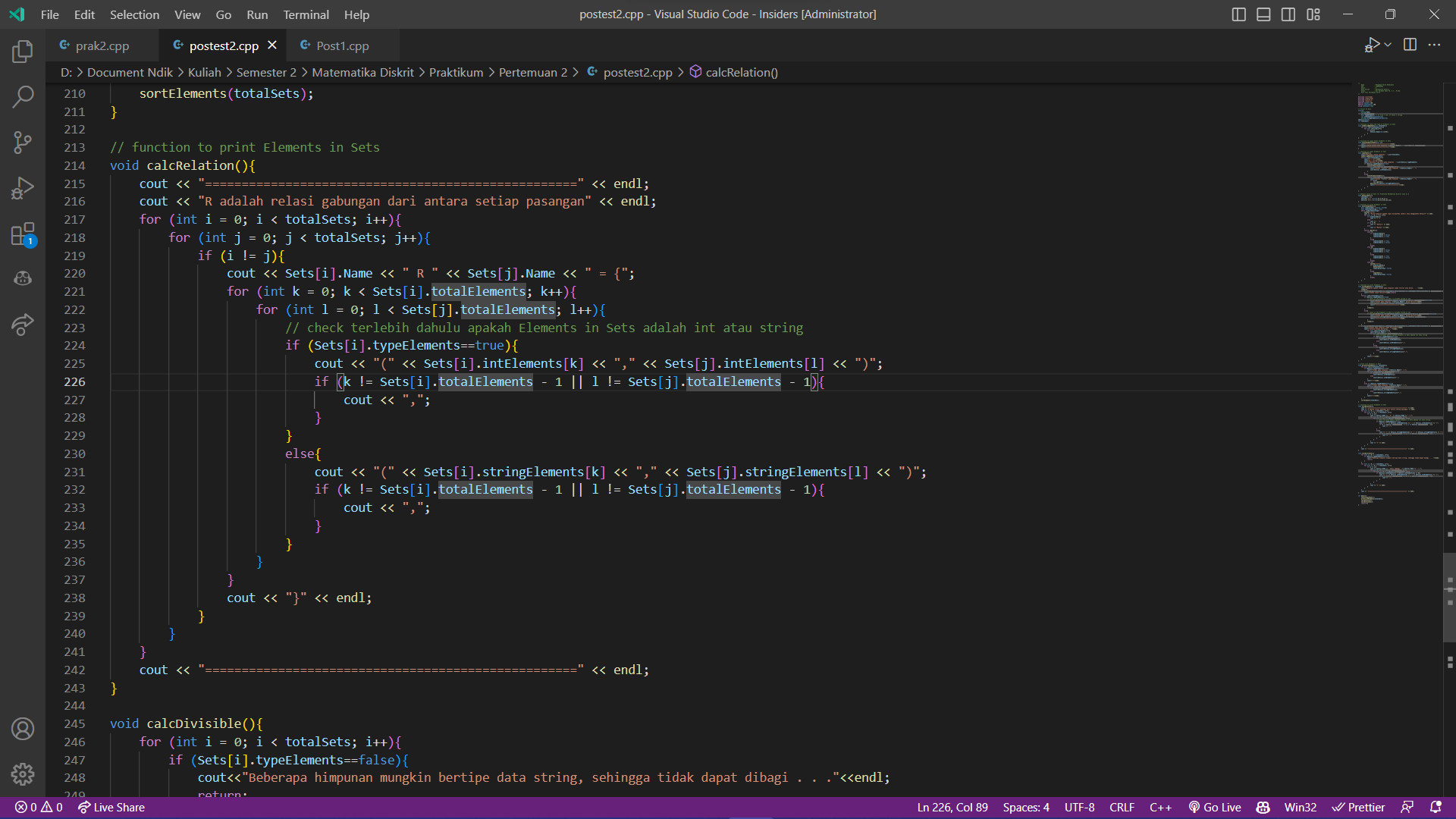


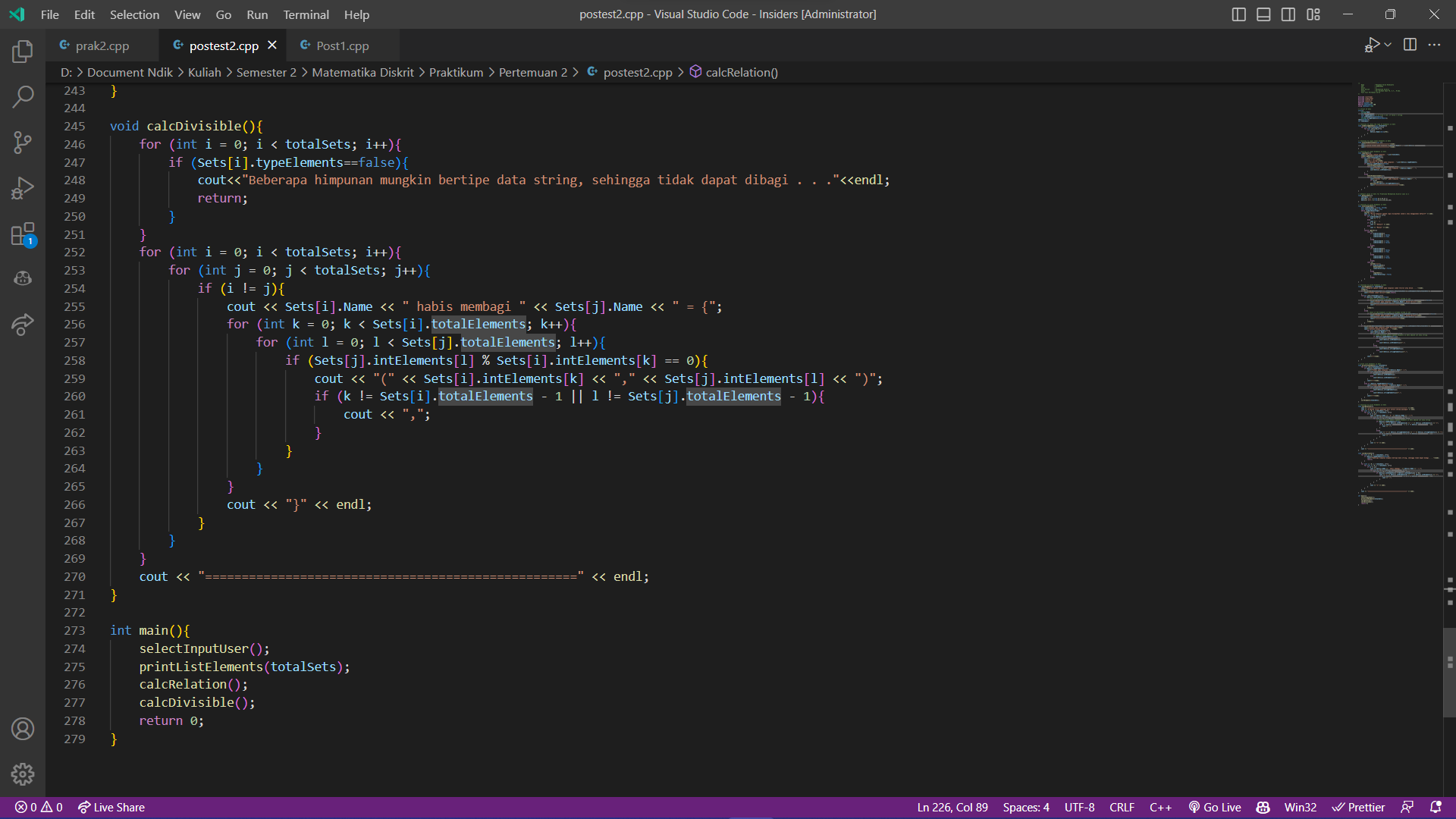






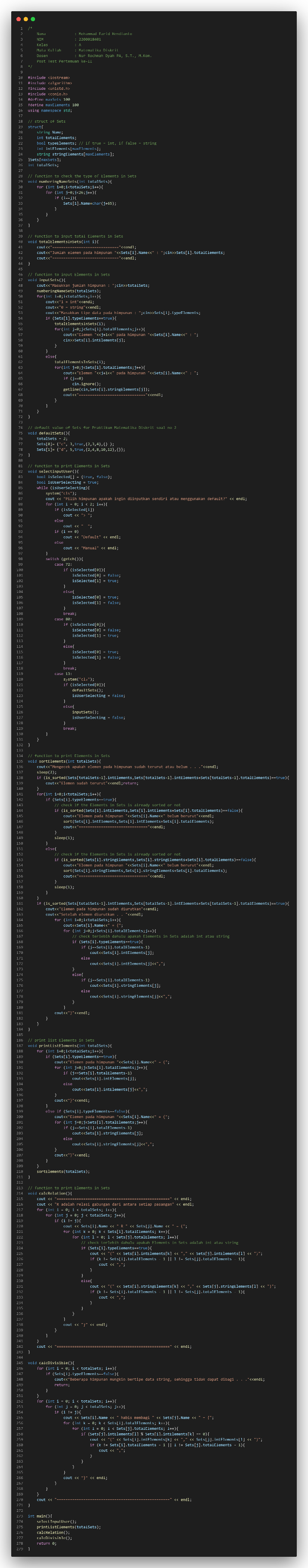






Berikut adalah full kode source kodingan dinamis

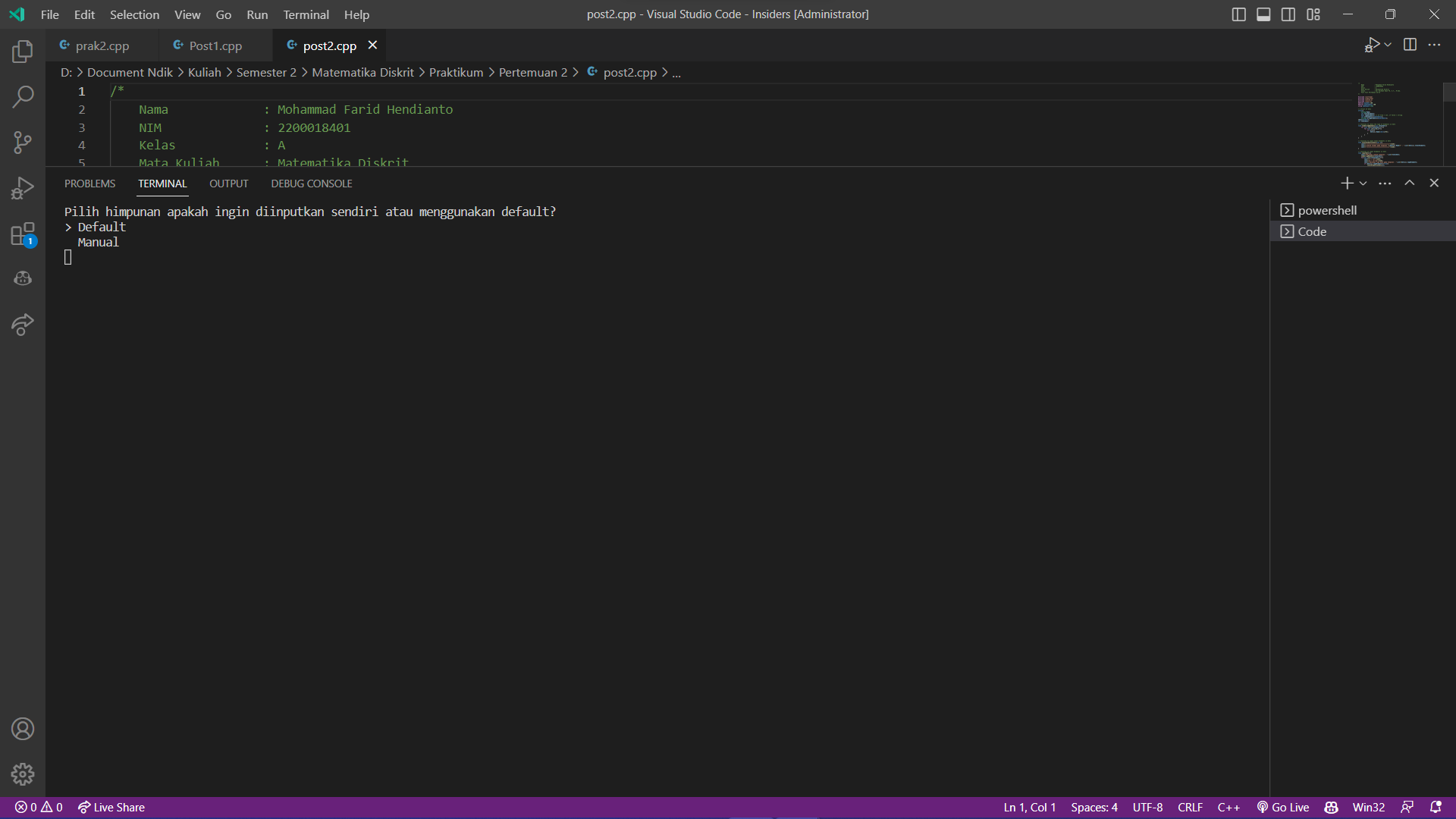




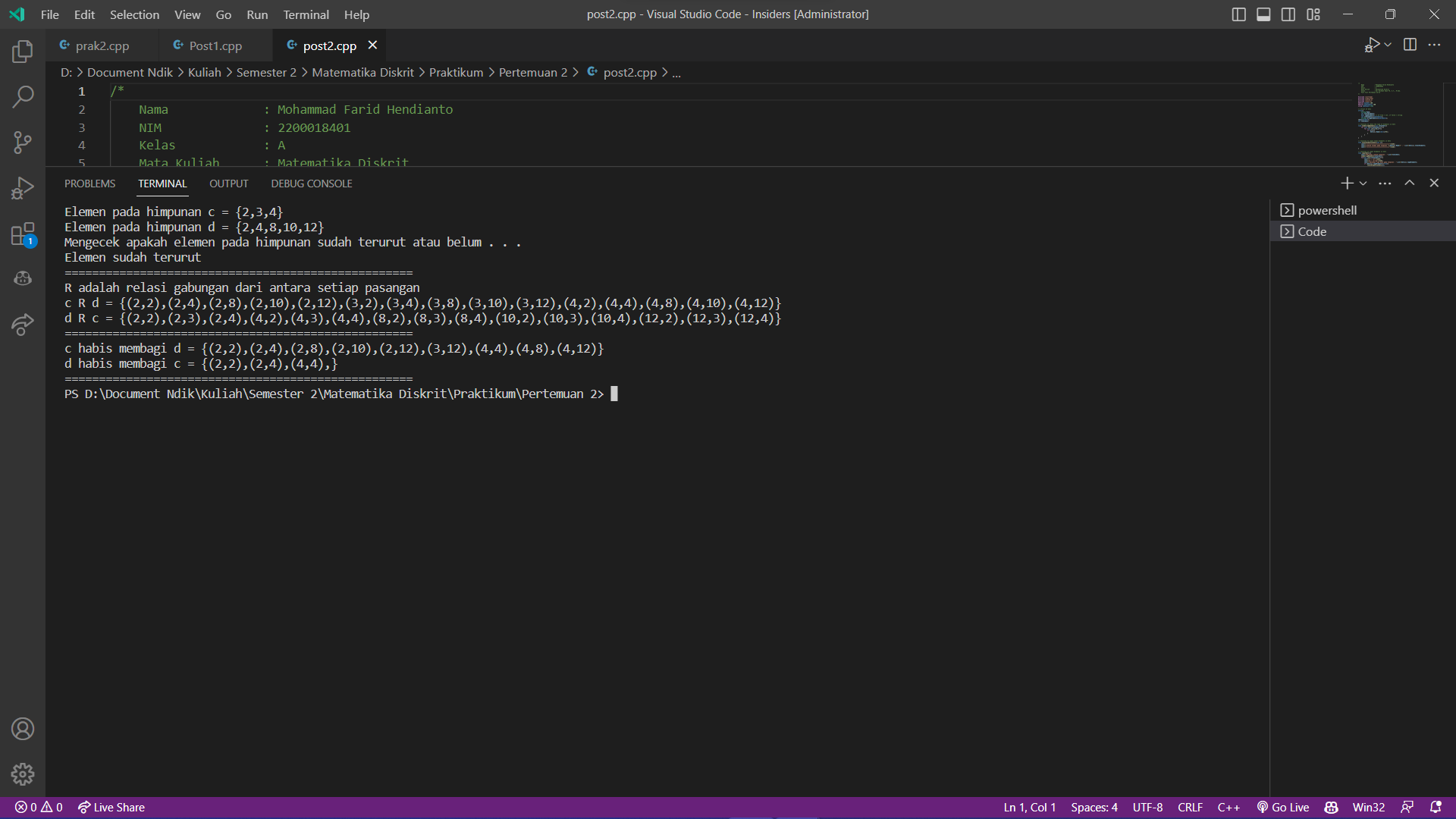
Untuk full source code pada program dinamis tentang relasi, dapat mengakses link sebagai berikut

<https://github.com/IRedDragonICY/Matematika-Diskrit/tree/main/Praktikum>

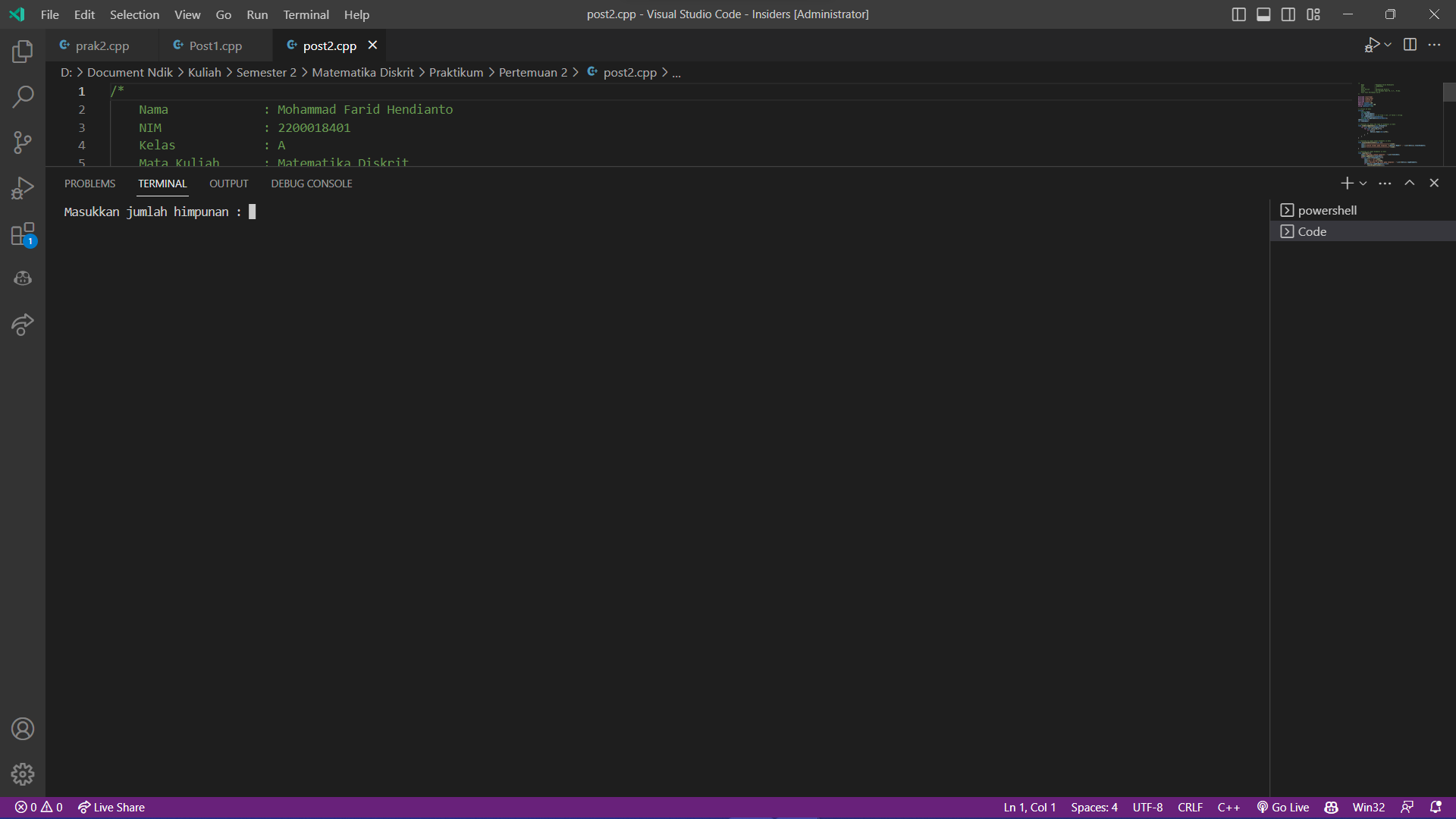
Berikut adalah tampilan program saat dijalankan



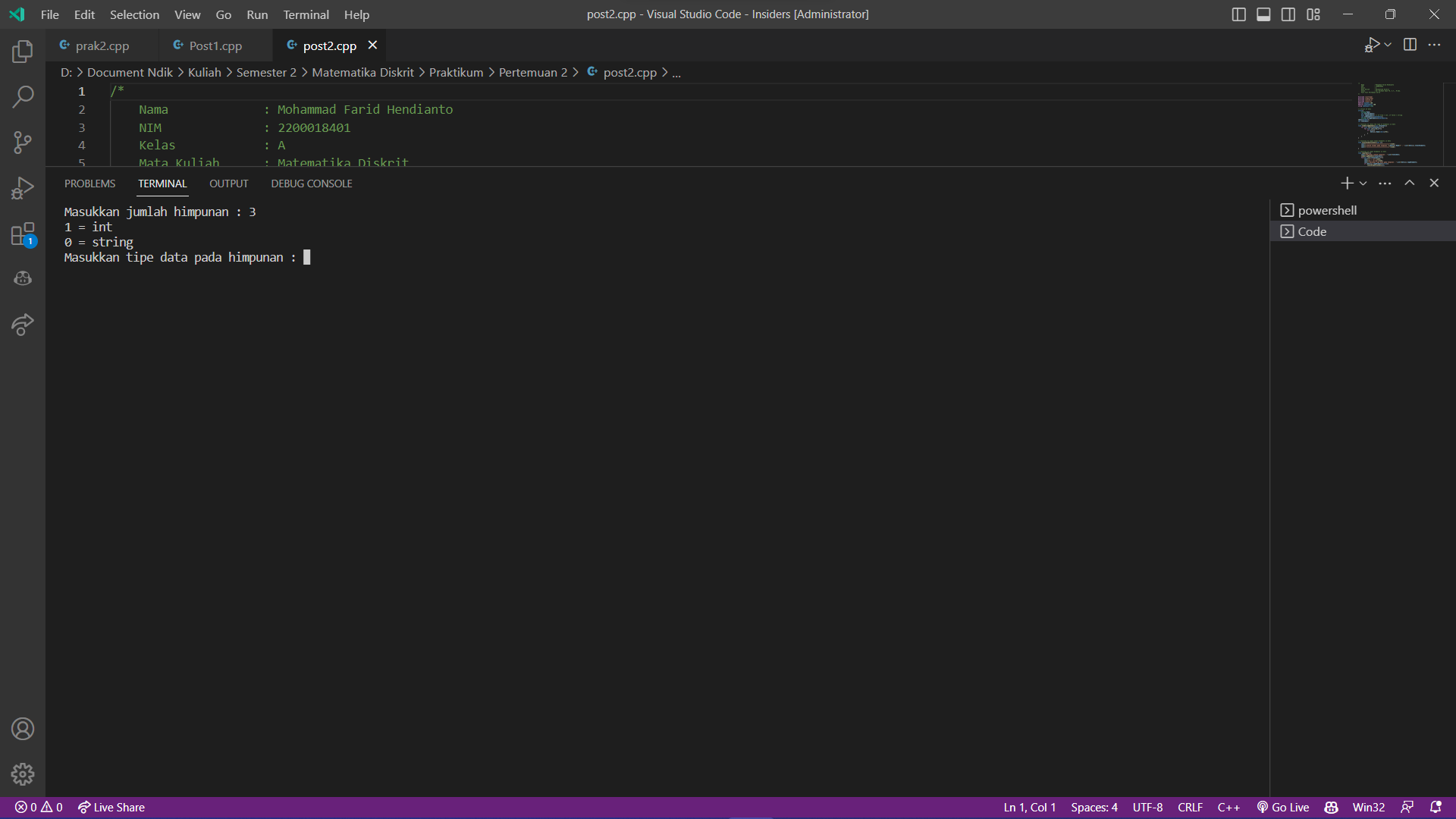
Pertama user akan diminta, apakah ingin menggunakan himpunan yang sudah ada di program (default) atau tidak. Untuk settings default sendiri, untuk menjawab soal pada nomor kedua. yaitu Tampilkan hasil himpunan D habis membagi C. Untuk memilih opsi, bisa diarahkan dengan arah panah atas dan bawah pada keyboard.



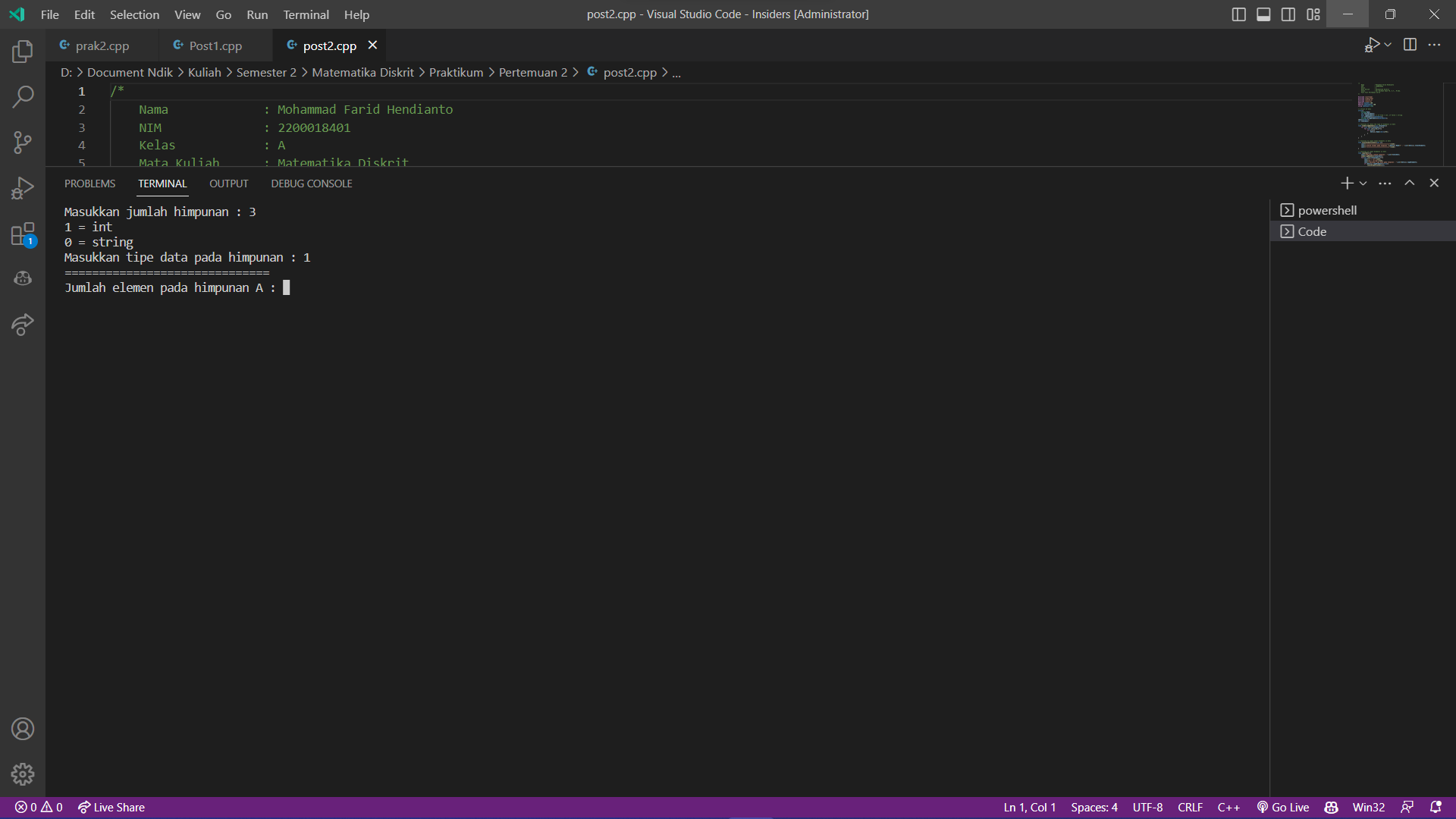
Gambar di atas merupakan tampilan jika memilih opsi default.

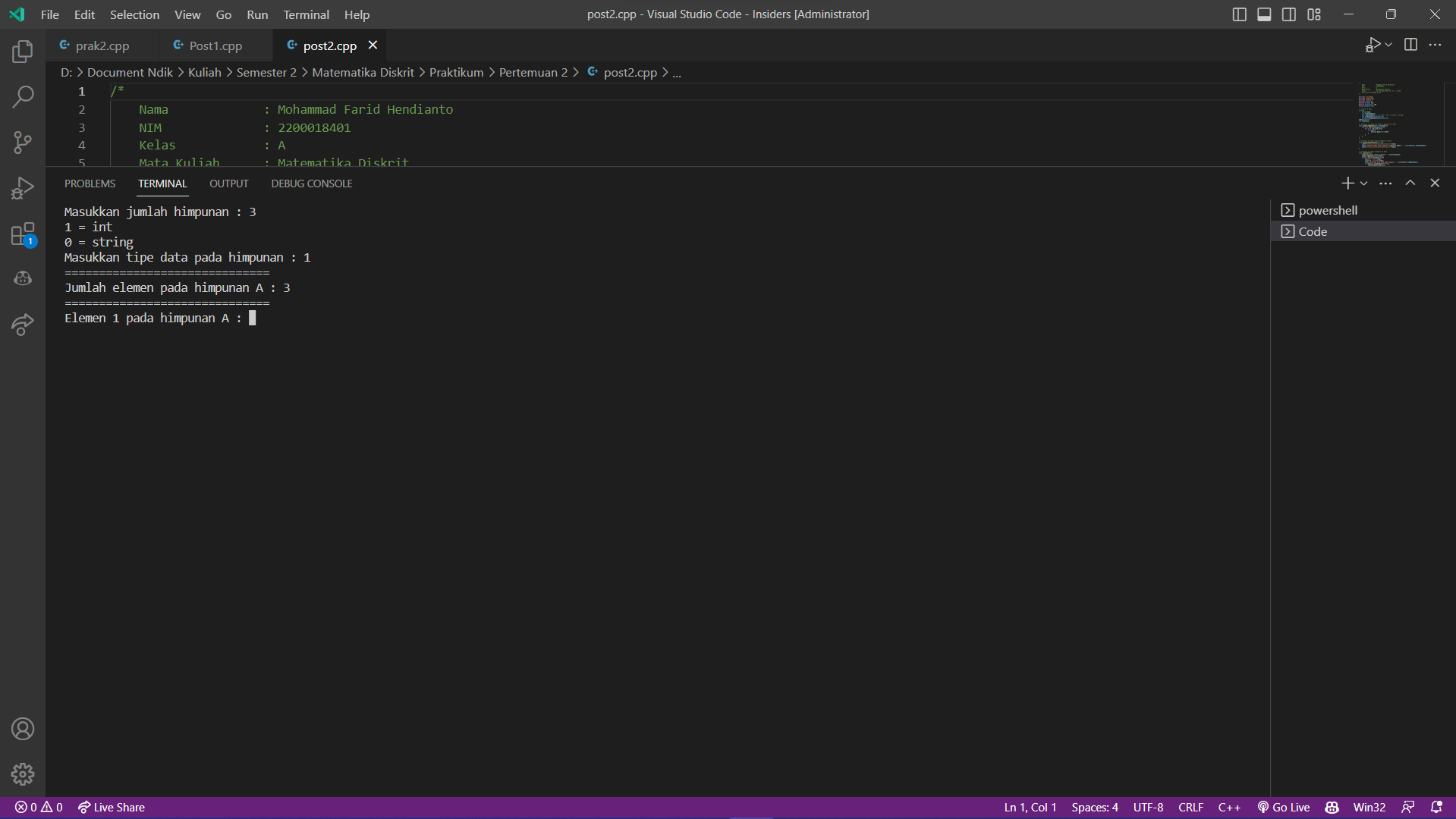


Jika memilih opsi manual, akan muncul pertama berapa banyak jumlah himpunan yang ingin di input.

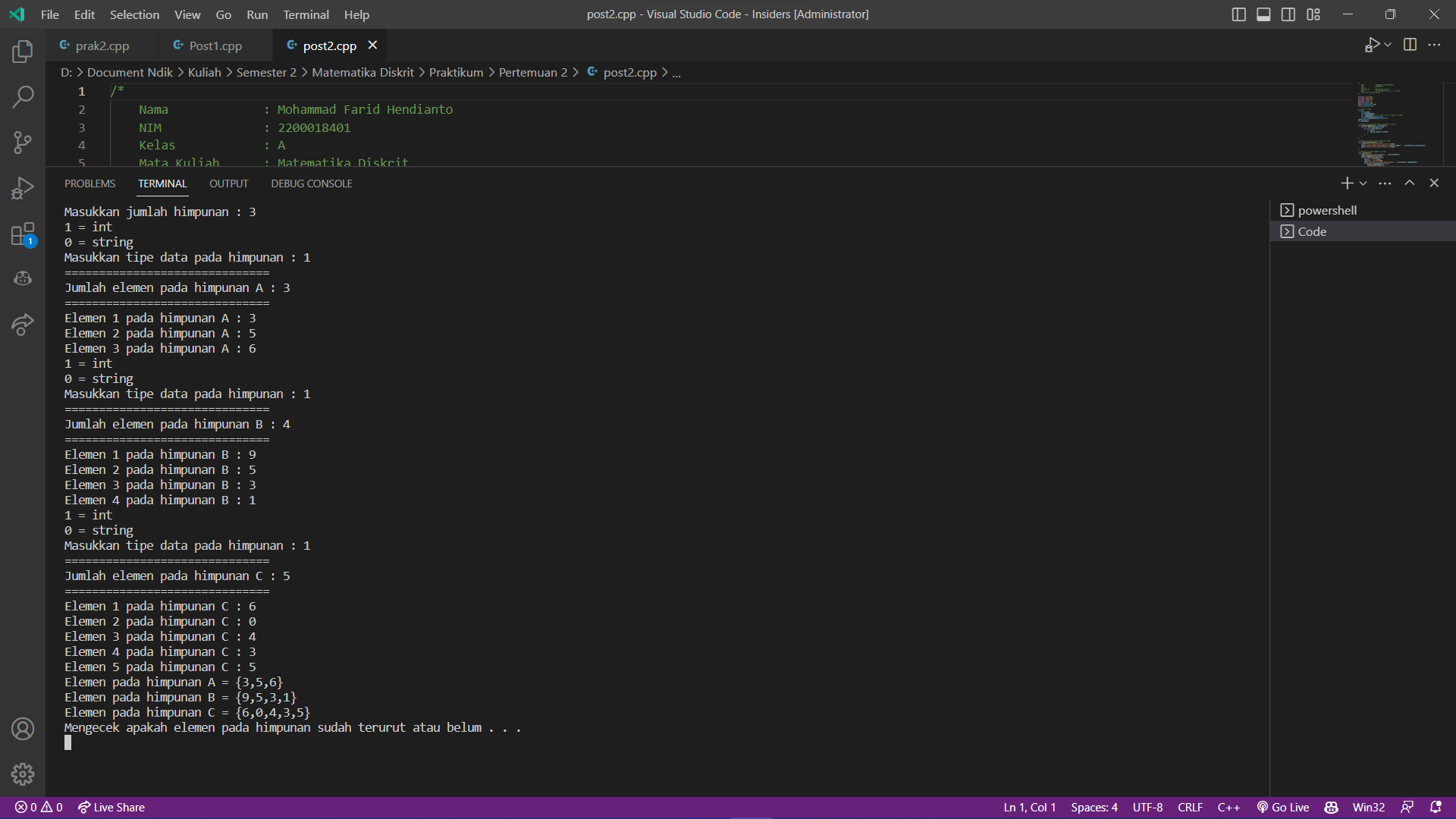


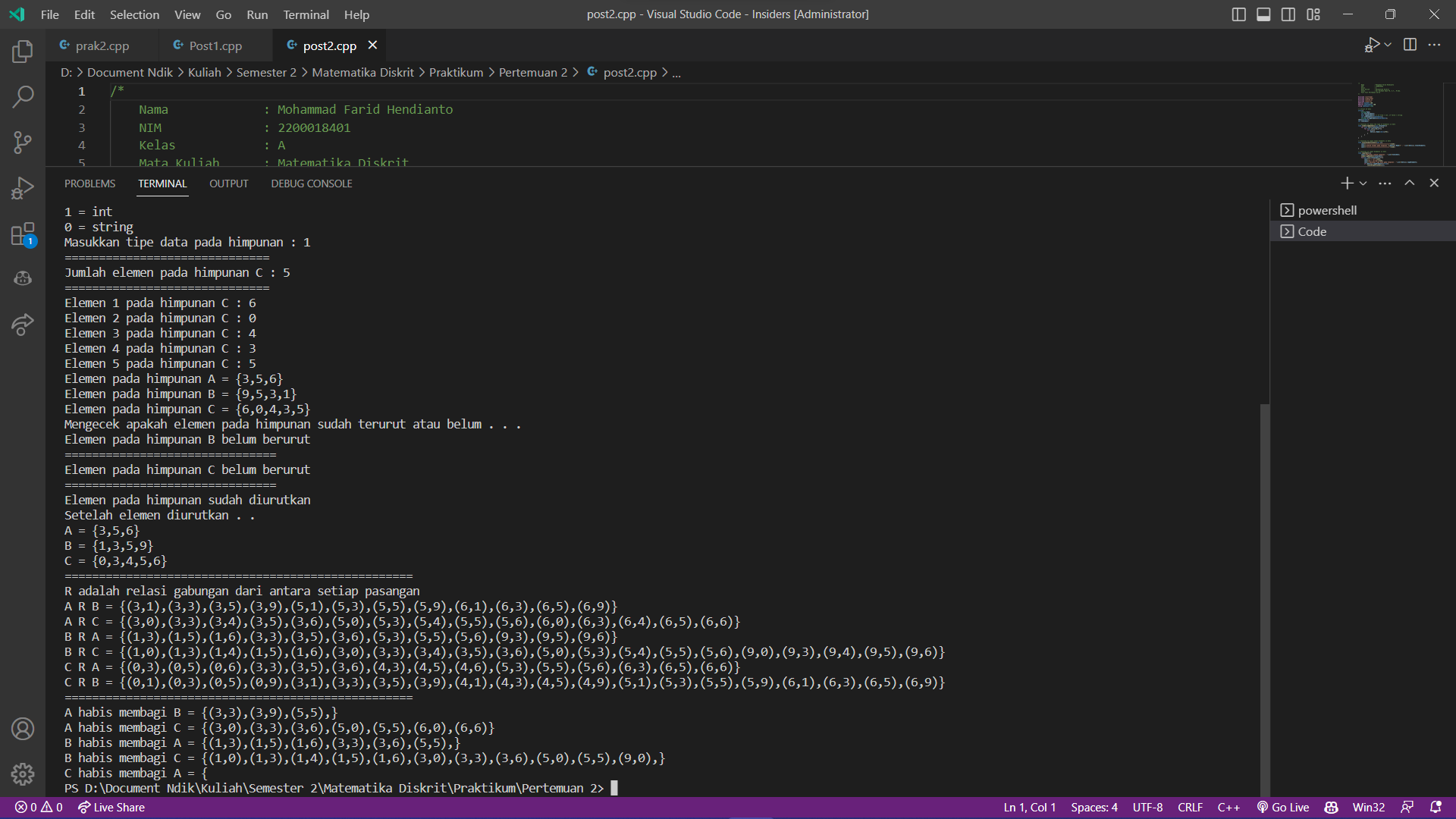
Kemudian akan muncul, tipe data himpunan yang ingin dimasukkan apa, misalnya integer untuk angka, sedangkan string untuk huruf.

Kemudian akan diminta jumlah elemen pada himpunan yang dimulai dari A



Kemudian akan mengisikan sesuai jumlah banyak elemen yang ingin diinput

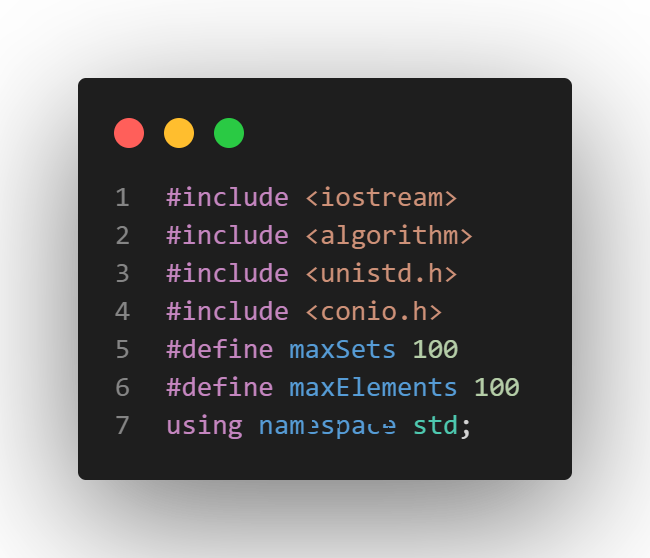
  
Kemudian berulang-ulang hingga input habis, setelah itu akan mengecek apakah himpunannya sudah berurut atau belum.



Terakhir akan dihitung relasi dan sisa bagi

Program ini merupakan sebuah program untuk menghitung relasi dan sifat dari himpunan. Program ini memiliki beberapa fitur antara lain:

* Input Himpunan, User dapat memasukkan himpunan sendiri atau menggunakan default yang telah disediakan.
* Mengecek Elemen pada Himpunan, Program akan mengecek apakah elemen pada himpunan sudah terurut atau belum dan akan melakukan sorting jika diperlukan.
* Menghitung Relasi. Program akan menghitung relasi gabungan dari antara setiap pasangan himpunan.
* Menghitung Pembagian. Program akan menghitung apakah suatu himpunan habis membagi himpunan lain atau tidak.

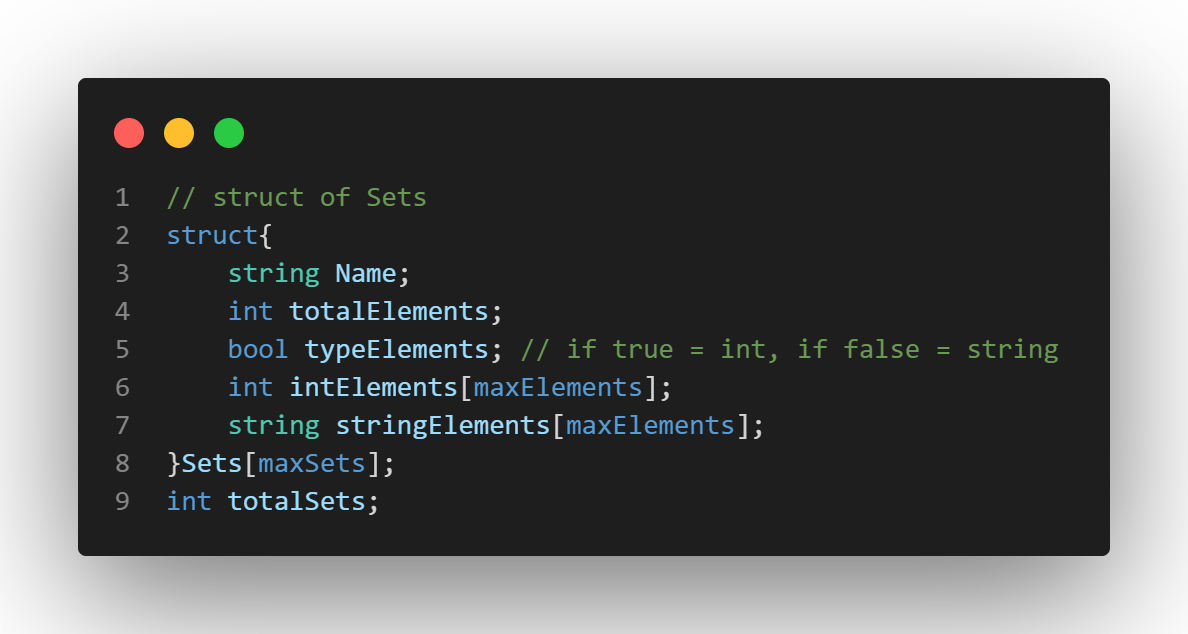


Kode di atas adalah sebuah header file yang digunakan untuk memuat library dan pendefinisian konstanta-konstanta yang akan digunakan pada program utama.

Pada baris-baris awal, ada beberapa library yang dimuat seperti iostream, algorithm, serta unistd.h dan conio.h. Library iostream digunakan untuk mengakses input dan output standar pada program, sedangkan algorithm digunakan untuk melakukan operasi-operasi pada array seperti sorting atau searching. Sedangkan library unistd.h dan conio.h digunakan untuk membuat program berjalan lebih interaktif dan user-friendly dengan menambahkan beberapa fitur seperti delay dan getch.

Selanjutnya, terdapat dua konstanta yang didefinisikan yaitu maxSets dan maxElements. Konstanta maxSets digunakan untuk menentukan jumlah maksimum himpunan yang dapat dimasukkan dalam program, sedangkan maxElements digunakan untuk menentukan jumlah maksimum elemen pada setiap himpunan.

Terakhir, menggunakan namespace std agar kita dapat langsung menggunakan fungsi-fungsi dari library standard C++ tanpa harus menuliskan std:: di depannya.

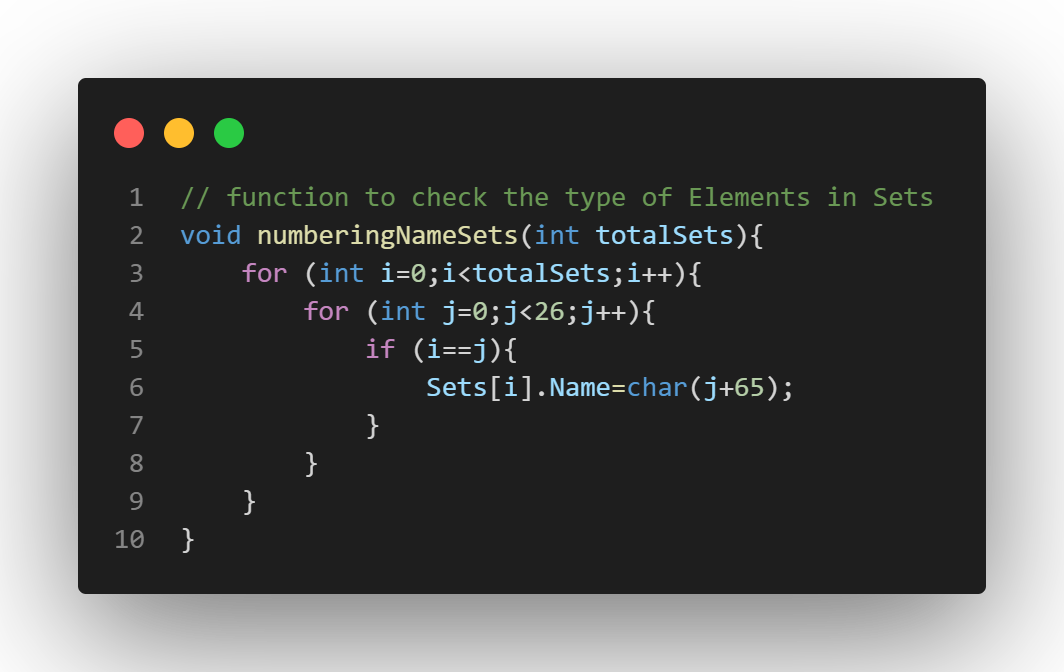


Potongan kode di atas merupakan definisi sebuah struct yang merepresentasikan himpunan dan variabel terkait. Struct ini memiliki beberapa field atau variabel yaitu:

* Name: nama dari himpunan, bertipe string
* totalElements: jumlah elemen yang ada pada himpunan, bertipe integer
* typeElements: tipe elemen yang ada pada himpunan, bertipe boolean. Jika nilainya true, maka elemen pada himpunan adalah integer, jika false maka elemen pada himpunan adalah string.
* intElements: array of integers, digunakan untuk menyimpan elemen pada himpunan jika tipe elemennya adalah integer.
* stringElements: array of strings, digunakan untuk menyimpan elemen pada himpunan jika tipe elemennya adalah string.
* Variabel Sets[maxSets] digunakan untuk menyimpan semua himpunan yang dimasukkan oleh user, dengan batasan maksimum jumlah himpunan yang dapat dimasukkan dalam program adalah sebanyak maxSets.

Terakhir, variabel totalSets digunakan untuk menyimpan jumlah total himpunan yang telah dimasukkan ke dalam program.

Dengan menggunakan definisi struct tersebut dan beberapa variabel terkait, program dapat memproses himpunan-himpunan yang dimasukkan oleh user dan melakukan operasi-operasi matematika diskrit yang berkaitan dengan himpunan.



Potongan kode di atas adalah sebuah fungsi yang digunakan untuk menentukan nama dari himpunan. Fungsi tersebut diberi nama numberingNameSets dengan masukan berupa jumlah total himpunan yang telah dimasukkan ke dalam program.

Pertama-tama, dilakukan perulangan for dengan index i yang berjalan dari 0 sampai totalSets-1 untuk melihat setiap himpunan yang telah dimasukkan pada program. Kemudian, dilakukan perulangan for lagi dengan index j yang berjalan dari 0 sampai 25 untuk menginisialisasi nama dari himpunan.

Dalam perulangan kedua, dilakukan pengecekan apakah nilai index i sama dengan nilai index j. Jika benar, maka dilakukan inisialisasi nama himpunan dengan menggunakan karakter ASCII yang sesuai dengan nilai index j, yaitu antara A hingga Z. Untuk melakukan konversi dari nilai index j ke karakter ASCII, dilakukan penjumlahan dengan 65, karena huruf A memiliki nilai ASCII 65 pada tabel ASCII.

Dengan demikian, setelah fungsi numberingNameSets dipanggil dengan argumen totalSets tertentu, maka setiap himpunan yang ada pada program akan diberikan nama A, B, C, dst sesuai urutan penambahan himpunan pada program. Nama ini nantinya akan digunakan untuk mempermudah pengguna program dalam mengidentifikasi himpunan yang mereka inginkan.



Potongan kode di atas adalah sebuah fungsi untuk menerima input dari pengguna berupa jumlah elemen pada setiap himpunan. Fungsi ini terdiri dari dua sub-fungsi: totalElementsInSets dan inputSets.

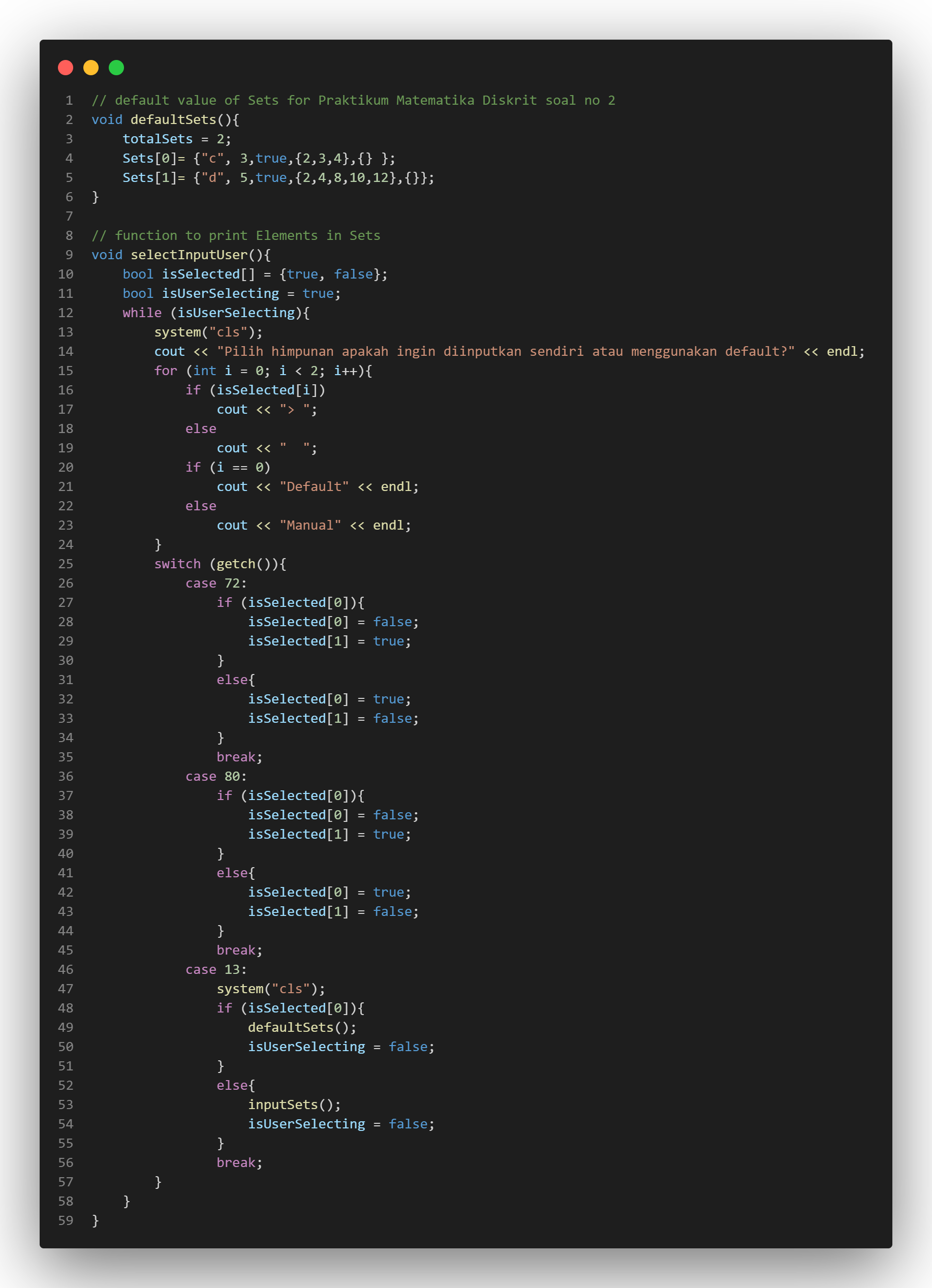
Fungsi totalElementsInSets akan menerima argumen i sebagai index himpunan yang sedang diproses oleh fungsi utama inputSets. Fungsi tersebut akan menampilkan pesan ke layar untuk meminta user memasukkan jumlah elemen pada himpunan dengan menggunakan cout. Kemudian user diminta untuk memasukkan nilai tersebut dengan menggunakan cin, dan nilai tersebut disimpan pada field totalElements di dalam struct Sets yang sedang diproses melalui index i.

Setelah itu, fungsi inputSets akan menjalankan perulangan for sebanyak totalSets kali untuk menginput setiap himpunan secara urut. Pertama-tama, program akan menampilkan pesan ke layar untuk meminta pengguna memasukkan tipe data himpunan (int atau string) melalui cout.

Kemudian, dilakukan pengecekan apakah tipe data yang dimasukkan adalah int atau string. Jika tipe datanya adalah int, maka program akan memanggil fungsi totalElementsInSets dengan mengirimkan index himpunan saat ini sebagai argumen, kemudian menjalankan perulangan for lagi untuk menginput setiap elemen himpunan secara urut, dan menyimpan nilainya pada array intElements di dalam struct Sets yang sedang diproses.

Sedangkan jika tipe datanya adalah string, program juga akan memanggil fungsi totalElementsInSets dengan mengirimkan index himpunan saat ini sebagai argumen. Kemudian, program akan menjalankan perulangan for lagi untuk menginput setiap elemen himpunan secara urut, dan menyimpan nilainya pada array stringElements di dalam struct Sets yang sedang diproses.

Dalam input elemen himpunan tipe data string, karena getline dapat menangkap karakter '\n' yang sebelumnya tidak terbaca oleh cin, maka pada elemen pertama harus menggunakan cin.ignore() untuk membersihkan buffer sebelum melakukan input. Setelah program selesai menerima input dari user, maka semua himpunan sudah siap digunakan untuk melakukan operasi-operasi matematika diskrit.



Potongan kode di atas adalah sebuah fungsi yang digunakan untuk menampilkan menu pada layar dan meminta user memilih apakah ingin menggunakan input default atau menginput sendiri. Fungsi ini diberi nama selectInputUser.

Fungsi tersebut akan menampilkan pesan ke layar dengan menggunakan cout, yaitu "Pilih himpunan apakah ingin diinputkan sendiri atau menggunakan default?" kemudian menampilkan dua pilihan dengan menggunakan perulangan for dan boolean array isSelected.

Setelah itu, program akan membaca input dari user dengan menggunakan getch() yang mengembalikan kode ASCII dari tombol keyboard yang ditekan. Jika kode ASCII yang ditekan adalah 72 (tombol panah atas), maka program akan mengubah nilai array isSelected, jika sebelumnya bernilai false menjadi true dan sebaliknya. Sedangkan jika kode ASCII yang ditekan adalah 80 (tombol panah bawah), maka program juga akan mengubah nilai array isSelected.

Sedangkan jika kode ASCII yang ditekan adalah 13 (tombol enter), maka program akan mengecek nilai array isSelected. Jika isSelected[0] bernilai true, maka akan dipanggil fungsi defaultSets() yang akan mengisi struktur data himpunan secara otomatis. Namun jika isSelected[1] bernilai true, maka program akan meminta user memasukkan struktur data himpunan mereka sendiri melalui panggilan fungsi inputSets().

Setelah memproses pilihan user, program akan keluar dari perulangan while dan kembali ke program utama untuk melakukan operasi-operasi matematika diskrit yang diminta oleh pengguna.



Potongan kode di atas adalah sebuah fungsi untuk mengecek dan mengurutkan elemen pada setiap himpunan. Fungsi ini terdiri dari dua sub-fungsi: printListElements dan sortElements. Fungsi printListElements akan menerima argumen totalSets yang merepresentasikan jumlah himpunan yang telah dimasukkan pada program. Fungsi tersebut akan menjalankan perulangan for sebanyak totalSets kali untuk menampilkan semua elemen himpunan yang sudah dimasukkan oleh pengguna. Jika tipe datanya adalah int, maka program akan menampilkan elemen pada himpunan dengan tipe data int. Namun jika tipe datanya adalah string, maka program akan menampilkan elemen pada himpunan dengan tipe data string.

Setelah itu, program akan memanggil fungsi sortElements dengan mengirimkan totalSets sebagai argumen. Fungsi sortElements akan menampilkan pesan ke layar "Mengecek apakah elemen pada himpunan sudah terurut atau belum . . ." kemudian mengecek satu per satu himpunan yang ada. Jika elemen pada himpunan sudah terurut, maka program akan menampilkan pesan "Elemen sudah terurut" dan langsung keluar dari fungsi sortElements.

Namun jika elemen pada himpunan belum terurut, maka program akan menampilkan pesan "Elemen pada himpunan ... belum berurut", kemudian akan mengurutkan elemen pada himpunan tersebut menggunakan fungsi sort() bawaan C++. Setelah itu, program akan menampilkan ulang seluruh elemen himpunan yang telah diurutkan.

Setelah seluruh elemen pada himpunan diurutkan, program akan menampilkan pesan "Elemen pada himpunan sudah diurutkan" dan kemudian menampilkan ulang seluruh elemen himpunan beserta dengan urutan yang baru. Setelah itu, program keluar dari fungsi printListElements dan kembali ke program utama untuk melakukan operasi-operasi matematika diskrit yang diminta oleh pengguna.



Potongan kode di atas adalah dua buah fungsi yang digunakan untuk menghitung relasi antara elemen pada setiap himpunan serta mengecek apakah suatu himpunan habis membagi himpunan lainnya atau tidak.

Fungsi pertama, yaitu calcRelation(), akan menampilkan pesan ke layar "R adalah relasi gabungan dari antara setiap pasangan" dan kemudian menjalankan dua perulangan for bersarang. Perulangan for yang pertama akan menjalankan variabel i untuk memilih himpunan pertama yang akan dibandingkan dengan himpunan kedua. Sedangkan perulangan for yang kedua akan menjalankan variabel j untuk memilih himpunan kedua yang akan dibandingkan dengan himpunan pertama.

Selanjutnya, program akan mengecek apakah variabel i sama dengan variabel j atau tidak. Jika variabel i tidak sama dengan variabel j, maka program akan menampilkan pesan yang berisi nama himpunan pertama, tanda R (relasi), dan nama himpunan kedua. Setelah itu, program akan menjalankan dua perulangan for bersarang lagi untuk mengecek setiap kombinasi elemen pada himpunan pertama dan himpunan kedua.

Ketika menemukan elemen yang cocok, program akan menampilkan pasangan elemen dalam bentuk kurung siku, misalnya (2,5) dan (3,6). Jika data ketika menjelajahi set bernilai integer maka elemen himpunan ditampilkan dalam format integer (angka), namun jika datanya bertipe string, maka elemen himpunan akan ditampilkan dalam format string (teks).

Fungsi kedua, yaitu calcDivisible(), akan mengecek apakah suatu himpunan habis membagi himpunan lainnya atau tidak. Jika ada himpunan yang menggunakan tipe data string, maka program akan menampilkan pesan ke layar "Beberapa himpunan mungkin bertipe data string, sehingga tidak dapat dibagi . . ." dan keluar dari fungsi.

Namun jika seluruh himpunan memiliki tipe data integer, maka program akan menjalankan dua perulangan for bersarang untuk mengecek setiap kombinasi elemen pada himpunan pertama dan himpunan kedua. Ketika menemukan elemen yang habis membagi elemen lainnya, program akan menampilkan pasangan elemen dalam bentuk kurung siku, misalnya (2,4) dan (3,6).

Setelah menampilkan semua hasil yang ditemukan, program akan menampilkan garis pemisah "===================================================" lalu keluar dari fungsi dan kembali ke program utama untuk melakukan operasi-operasi matematika diskrit yang diminta oleh pengguna.



Potongan kode di atas adalah sebuah fungsi utama (main function) yang akan dipanggil ketika program dijalankan. Fungsi ini akan menjalankan empat buah fungsi lainnya, yakni selectInputUser(), printListElements(), calcRelation(), dan calcDivisible().

Fungsi selectInputUser() akan dijalankan terlebih dahulu untuk meminta input dari pengguna mengenai jumlah himpunan yang ingin dibuat dan elemen yang ingin dimasukkan pada setiap himpunan tersebut.

Selanjutnya, fungsi printListElements() akan dijalankan untuk menampilkan seluruh elemen yang sudah dimasukkan oleh pengguna pada setiap himpunan. Kemudian, fungsi calcRelation() dan calcDivisible() akan dijalankan secara bergantian untuk menghitung relasi antara himpunan-himpunan yang telah dimasukkan oleh pengguna.

Setelah semua perhitungan selesai dilakukan, program akan menampilkan pesan "Press any key to continue . . ." dan menunggu pengguna untuk menekan tombol apa saja sebelum keluar dari program. Akhirnya, fungsi main() akan mengembalikan nilai 0 dan program akan berakhir.

**Tampilkan hasil himpunan D habis membagi C!**

