**Praktikum Algoritma dan Struktur Data**

“Quick Sort, Merge Sort”

**Oleh:**

Marlon Immanuel Sarapang / 5223600010

**Program Studi Sarjana Terapan Teknologi Game**

**Departemen Teknologi Multimedia Kreatif**

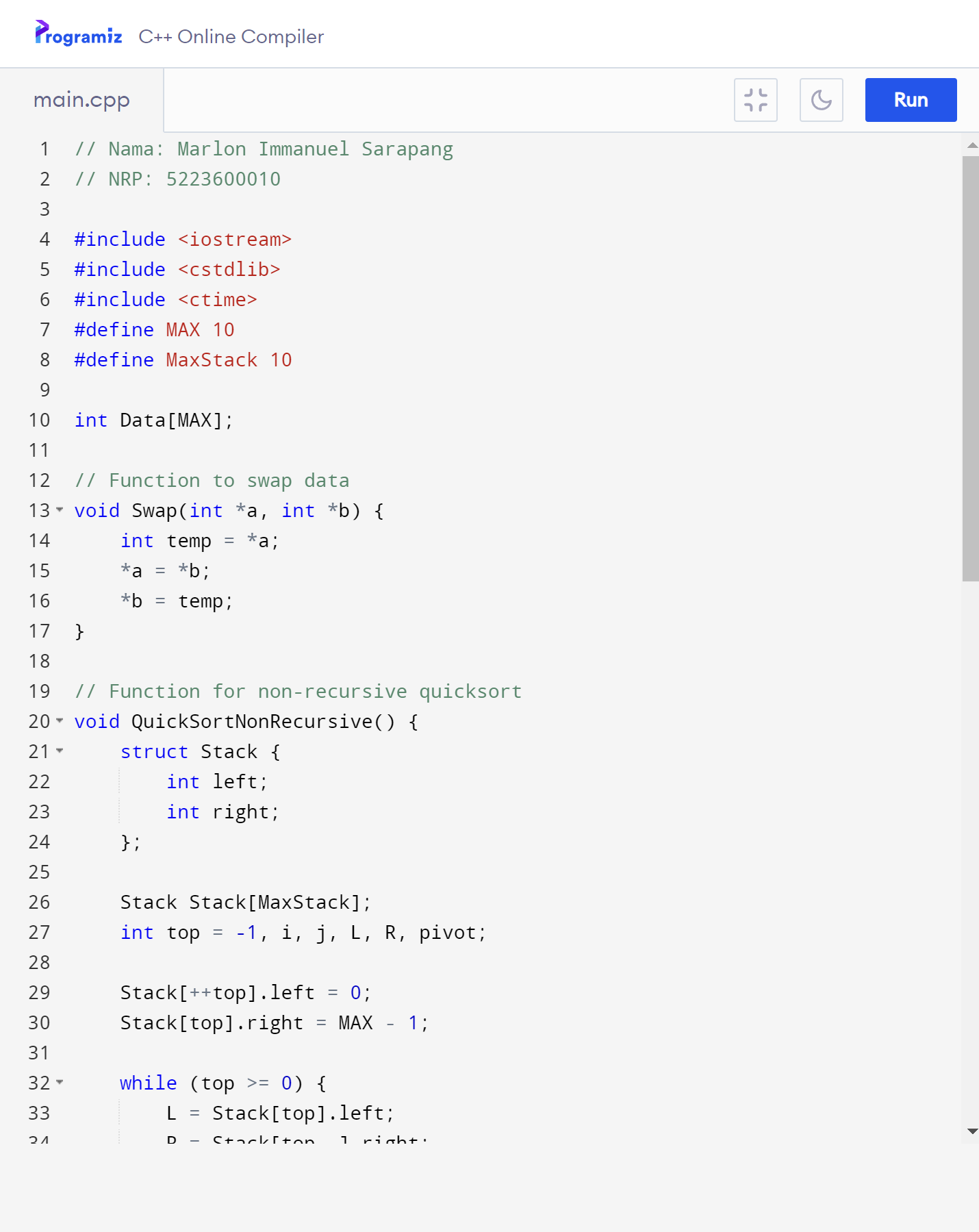
**Politeknik Elektronika Negeri Surabaya**

**2024**

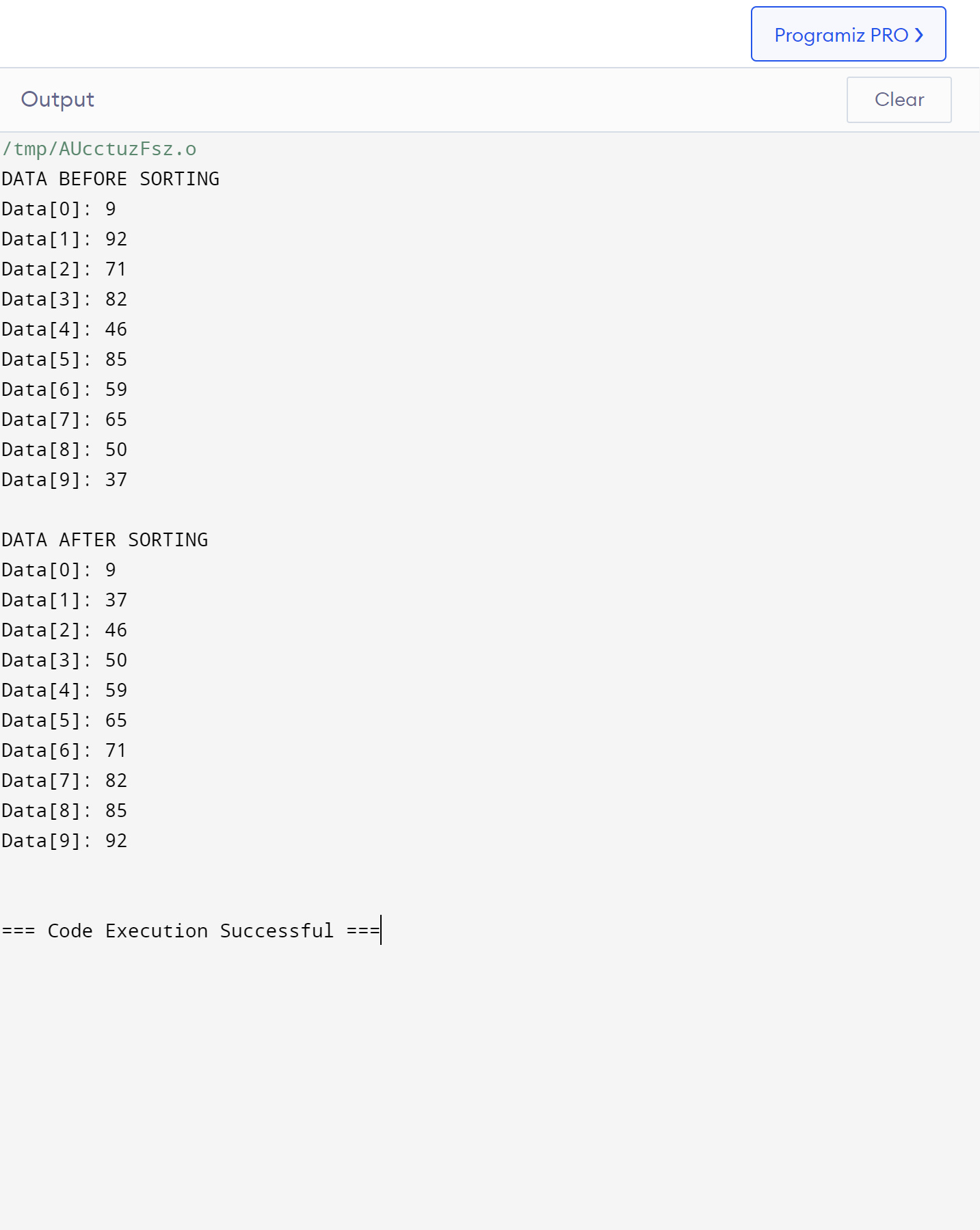
**Percobaan**

1. Percobaan Quick Sort Non-Rekursif

**Source code**

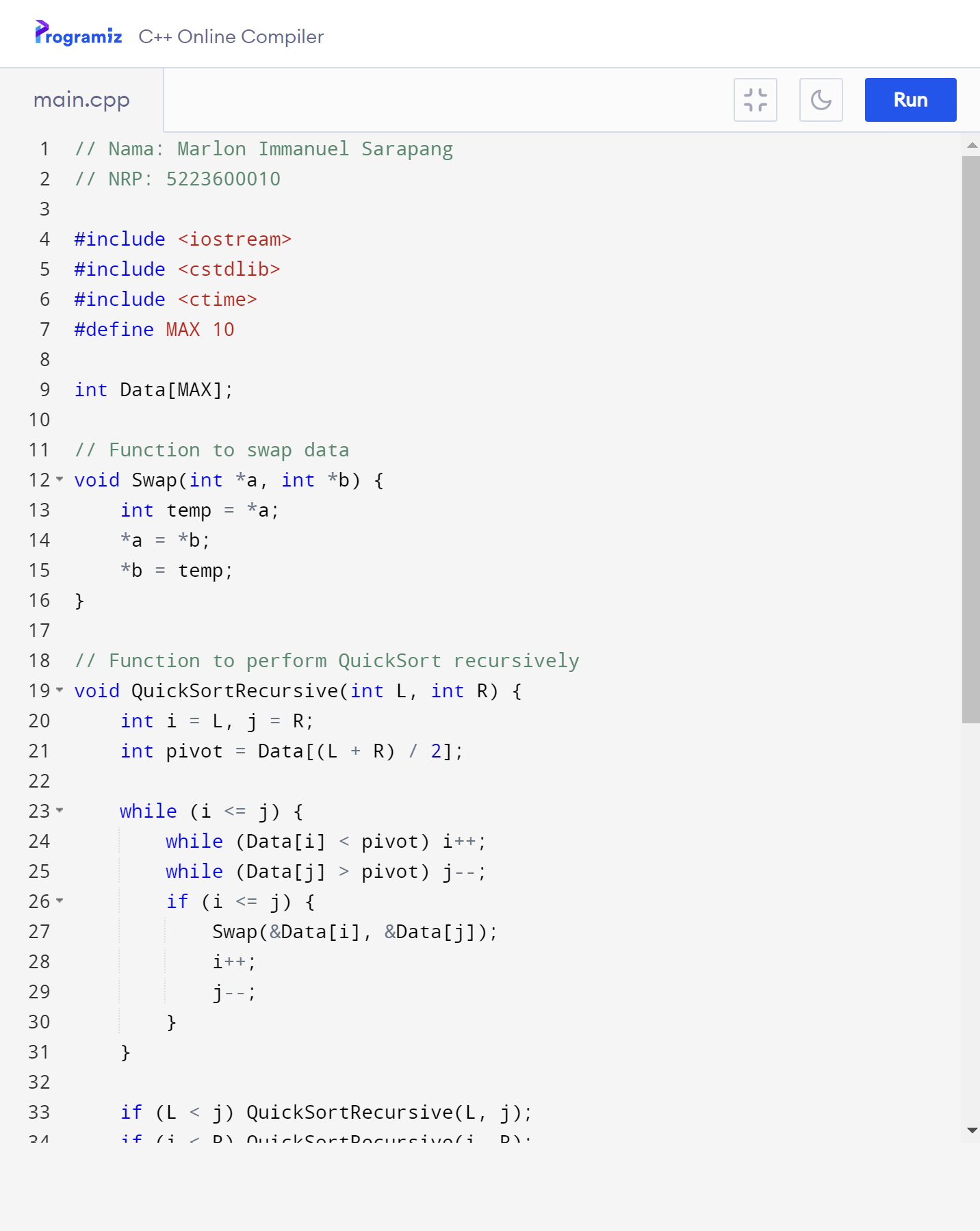


**Hasil**

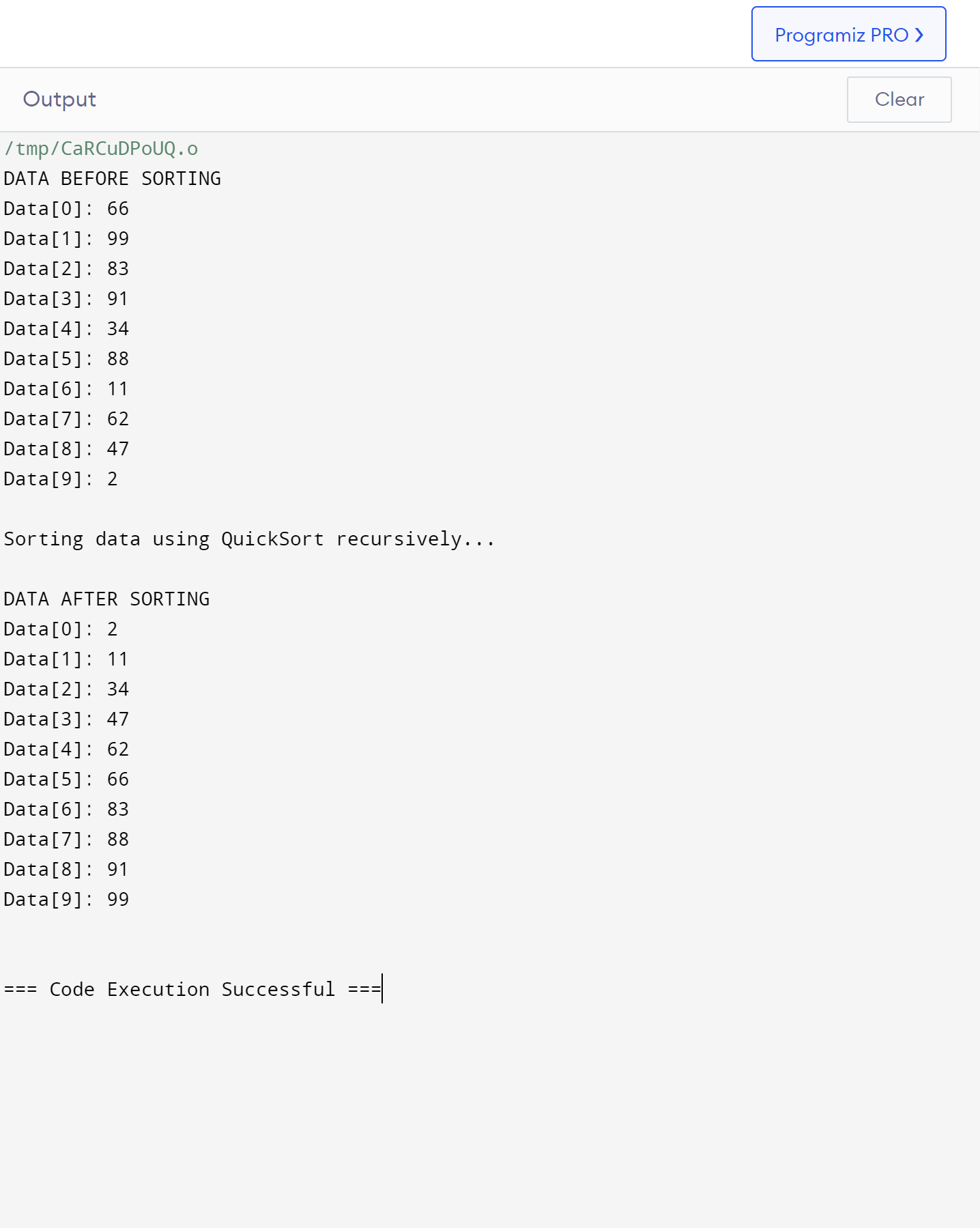


1. Percobaan Quick Sort Rekursif

**Source code**

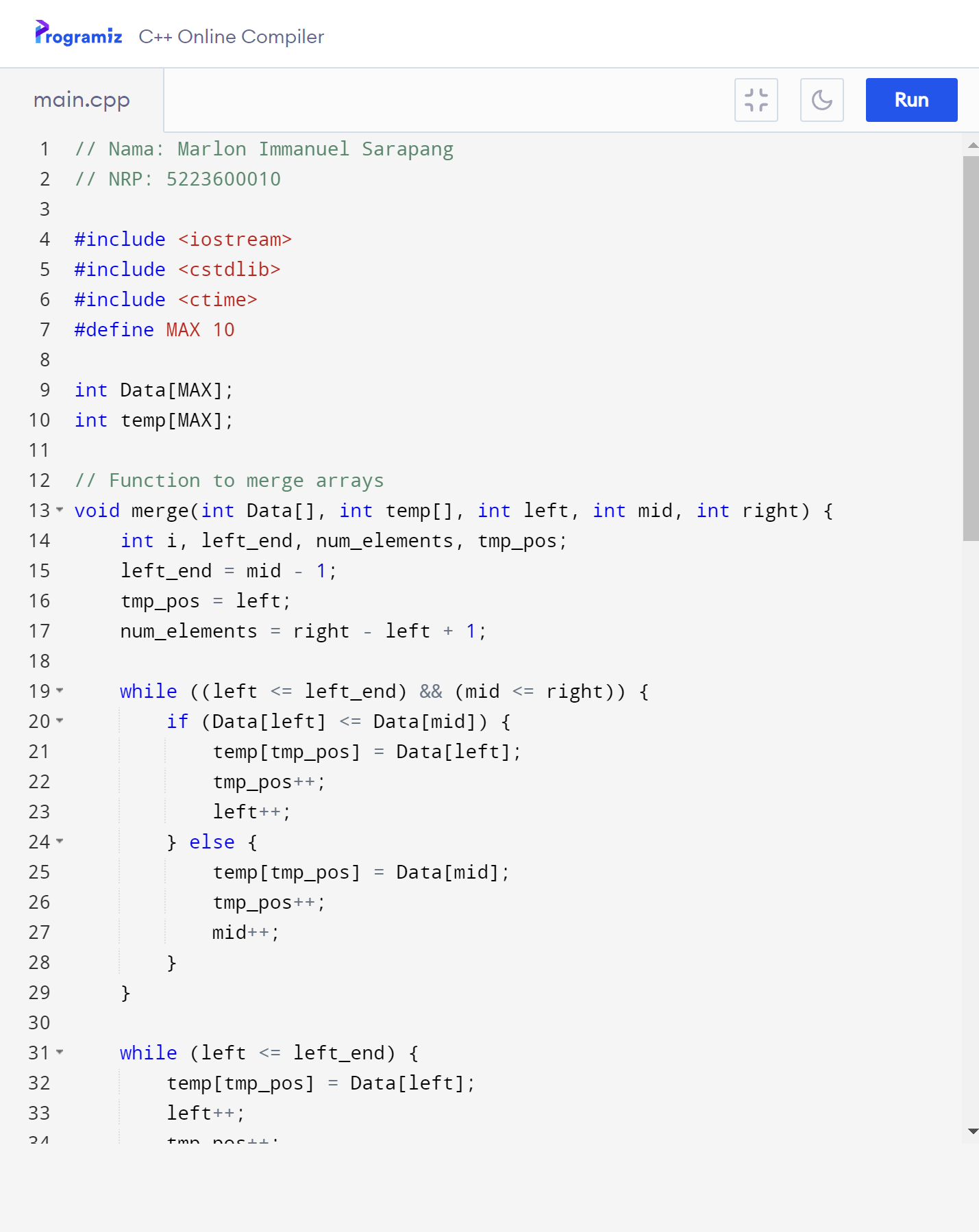


**Hasil**

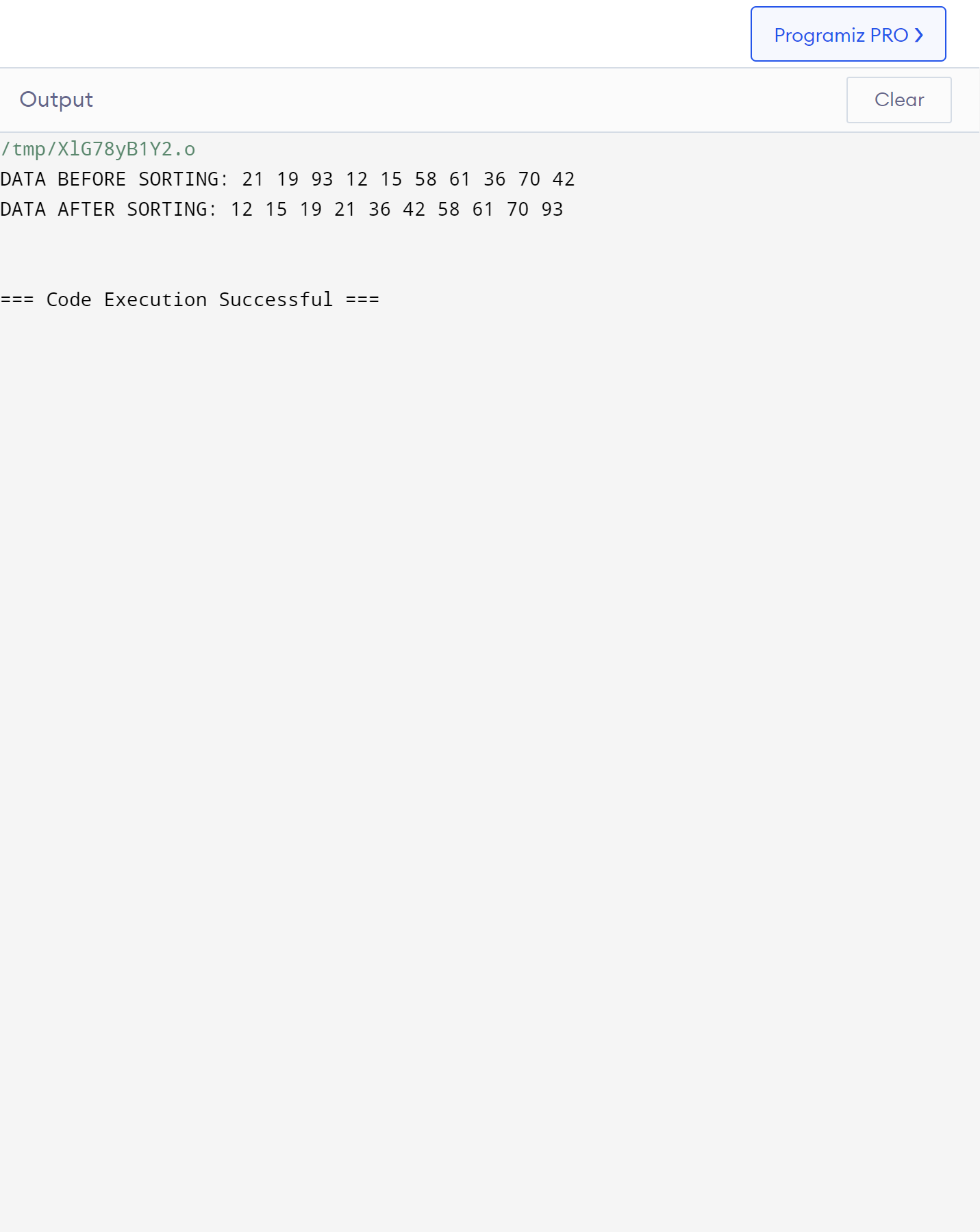


1. Percobaan Merge Sort

**Source code**



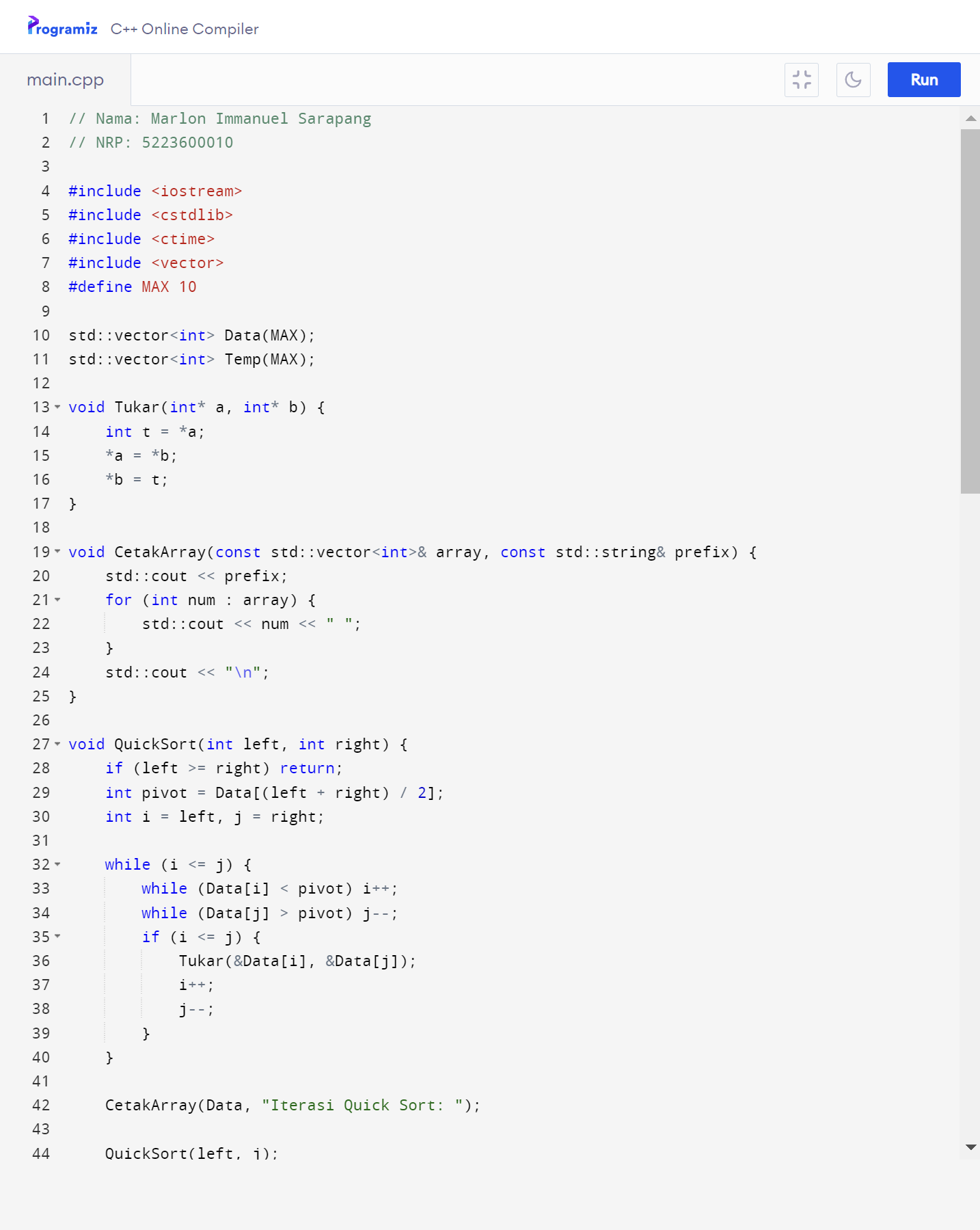
**Hasil**



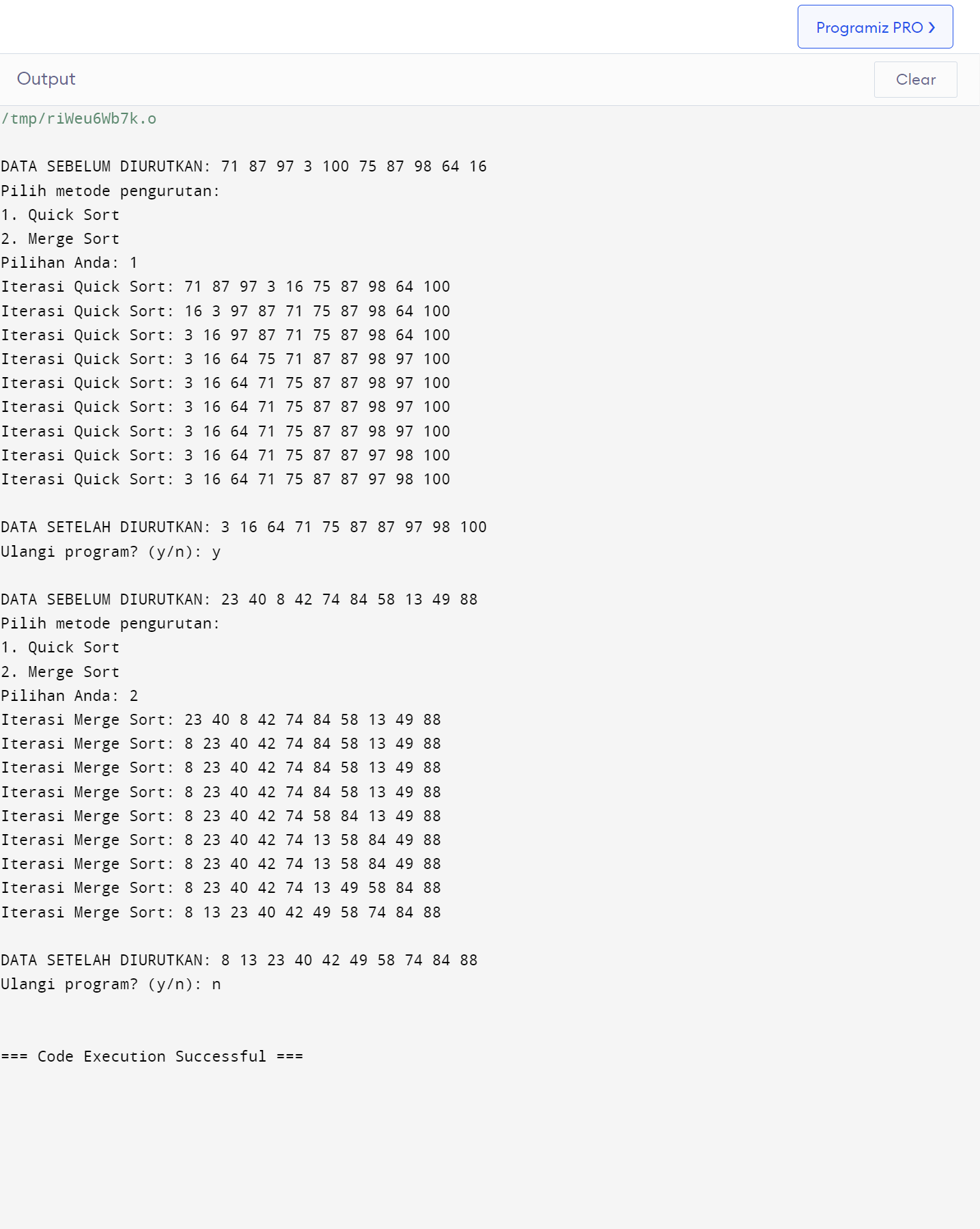
**Latihan**

1. Tambahkan kode program untuk menampilkan perubahan setiap iterasi dari proses pengurutan dengan quick sort dan merge sort.

**Source code**

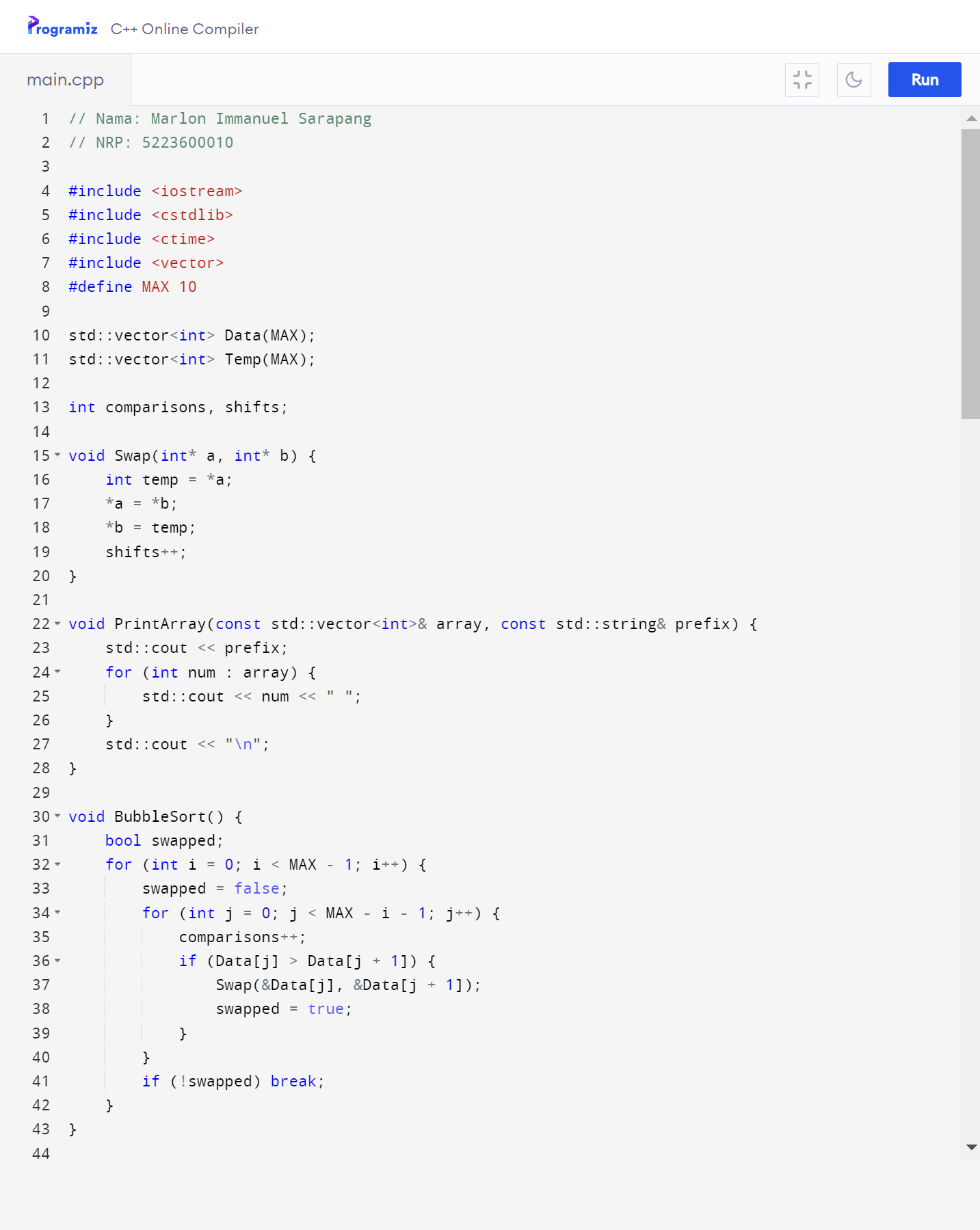


**Hasil**



1. Tambahkan kode program untuk menghitung banyaknya perbandingan dan pergeseran pada algoritma quick sort dan merge sort.

**Source code**

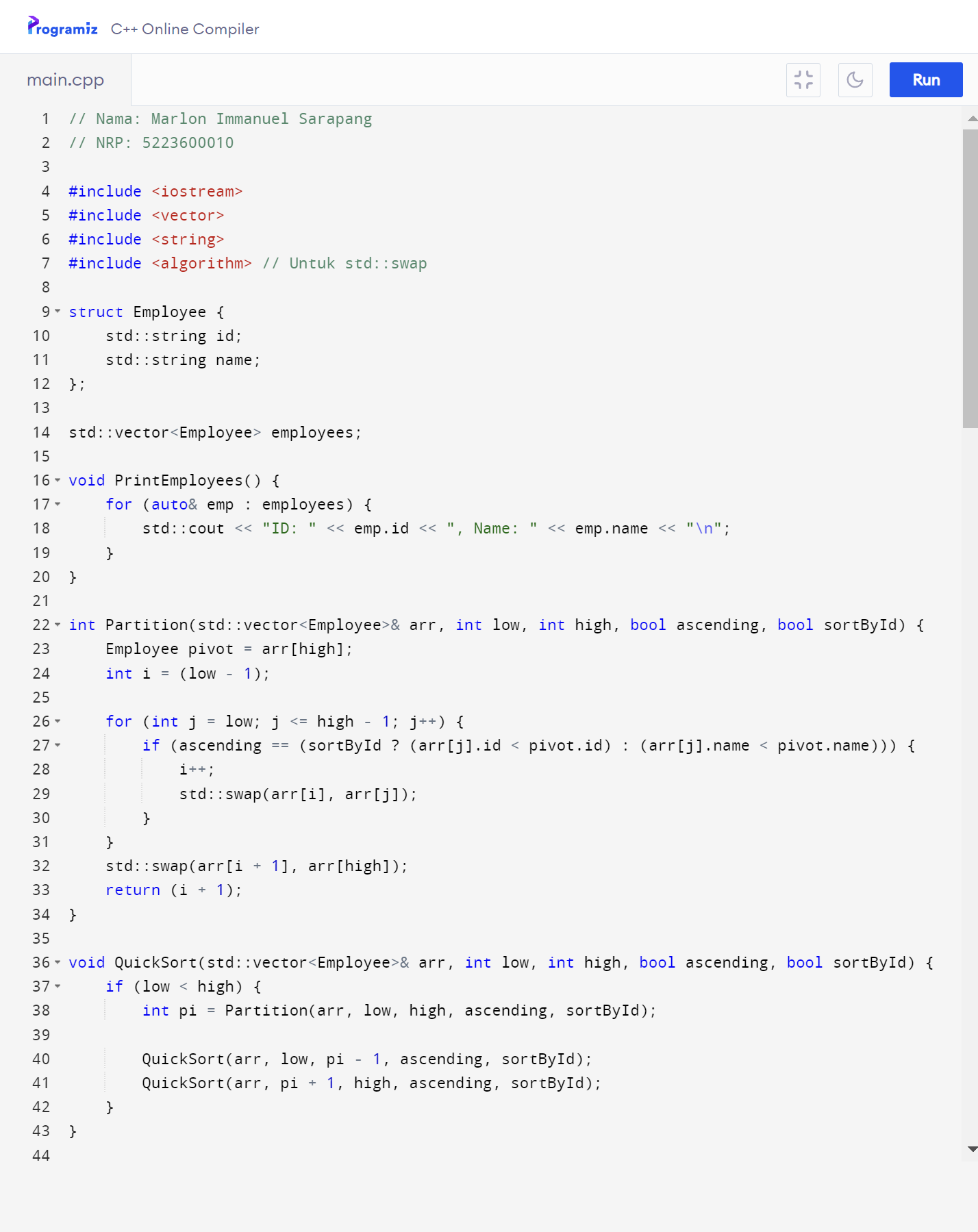


**Hasil**

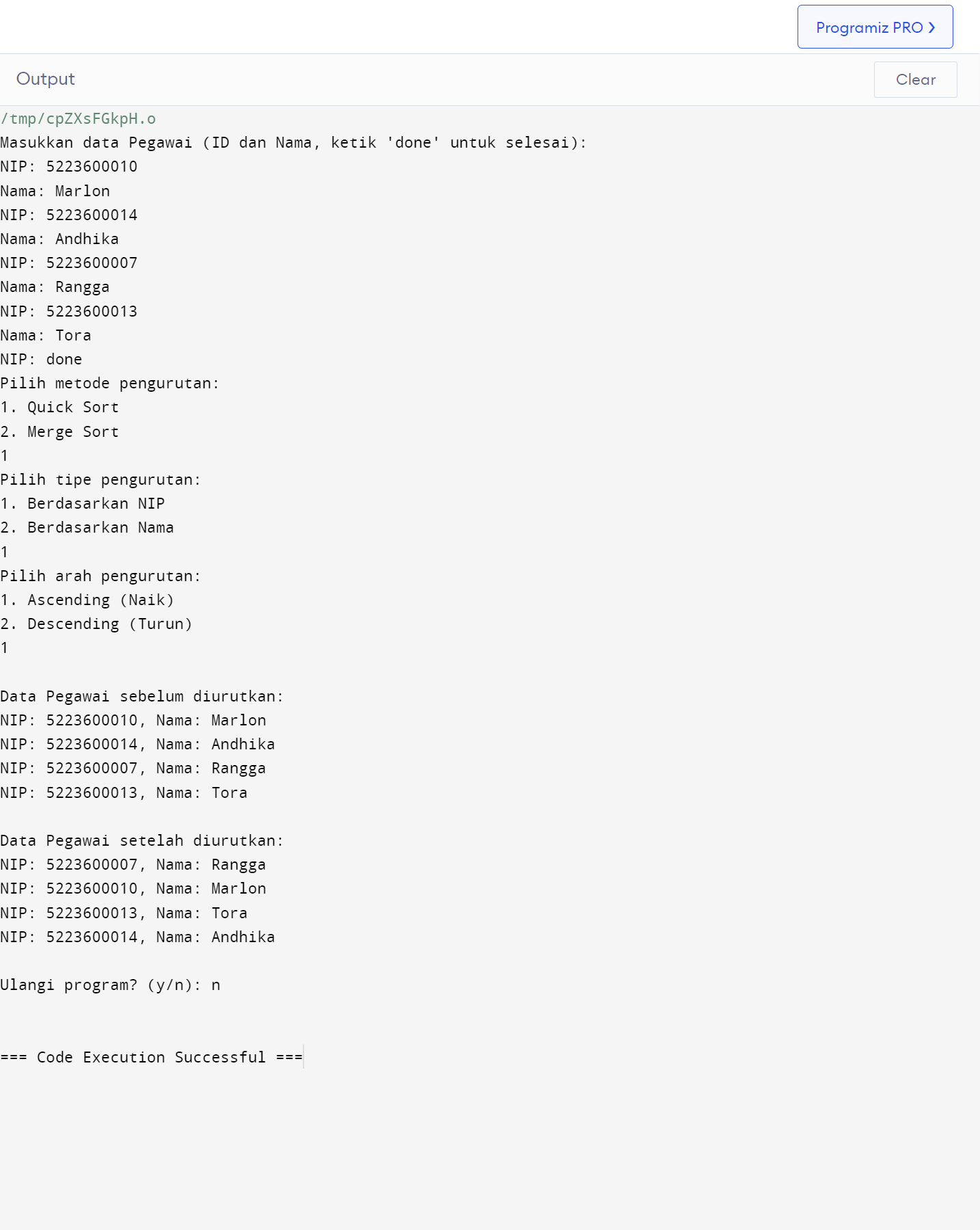
****

1. Implementasikan pengurutan data Pegawai pada tugas pendahuluan dengan ketentuan:
2. Metode pengurutan dapat dipilih.
3. Pengurutan dapat dipilih secara urut naik atau turun.
4. Pengurutan dapat dipilih berdasarkan NIP dan NAMA.
5. Gunakan struktur data array.

**Source code**



**Hasil**



1. Berikan kesimpulan dari percobaan dan latihan yang telah Anda lakukan.

Dalam semua contoh di atas, kami telah menggali dan mengimplementasikan dua algoritma pengurutan yang sangat efektif dan efisien, yaitu Quick Sort dan Merge Sort. Quick Sort adalah algoritma pengurutan yang memilih sebuah elemen sebagai pivot dan mempartisi array ke dalam dua bagian, di mana satu sisi memiliki elemen lebih kecil dari pivot dan sisi lain memiliki elemen lebih besar dari pivot, dan ini dilakukan secara rekursif pada kedua sisi pivot. Ini dikenal karena performanya yang cepat dan efisien, khususnya pada dataset besar, meskipun pilihan pivot yang buruk dapat menurunkan performanya menjadi quadratic. Di sisi lain, Merge Sort adalah algoritma pengurutan yang stabil yang menggunakan pendekatan divide-and-conquer untuk membagi array menjadi dua bagian yang hampir sama besar, mengurutkannya secara terpisah, kemudian menggabungkannya. Ini menawarkan keunggulan berupa kestabilan dan efisiensi waktu yang konsisten, tetapi memerlukan ruang tambahan yang proporsional terhadap data yang diurutkan.

Dalam penggunaan praktis, kedua algoritma tersebut telah diterapkan pada data struktural dalam contoh pengurutan data pegawai, memungkinkan pengurutan berdasarkan berbagai kriteria seperti NIP atau nama, dan pengurutan naik atau turun. Fitur ini menunjukkan fleksibilitas kedua algoritma dalam menghadapi berbagai kebutuhan pengurutan. Quick Sort cenderung lebih cepat pada umumnya tetapi dengan risiko penurunan performa pada kasus tertentu, sedangkan Merge Sort menawarkan performa yang konsisten tetapi dengan biaya penggunaan memori tambahan. Pilihan antara kedua algoritma ini akan bergantung pada konteks spesifik aplikasi, termasuk ukuran data, kebutuhan akan kestabilan, dan ketersediaan memori, menegaskan pentingnya memahami karakteristik internal dan trade-offs dari setiap algoritma pengurutan.