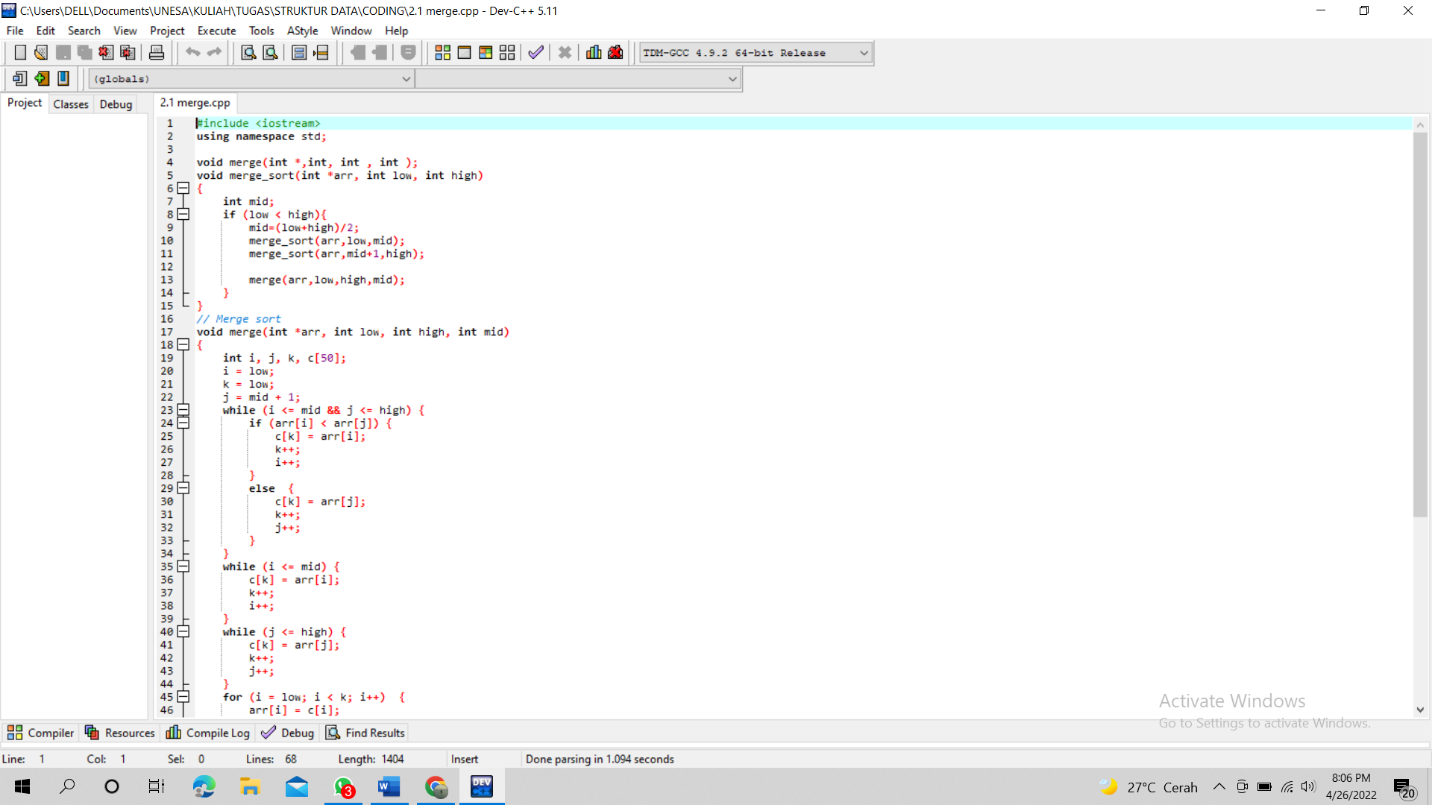
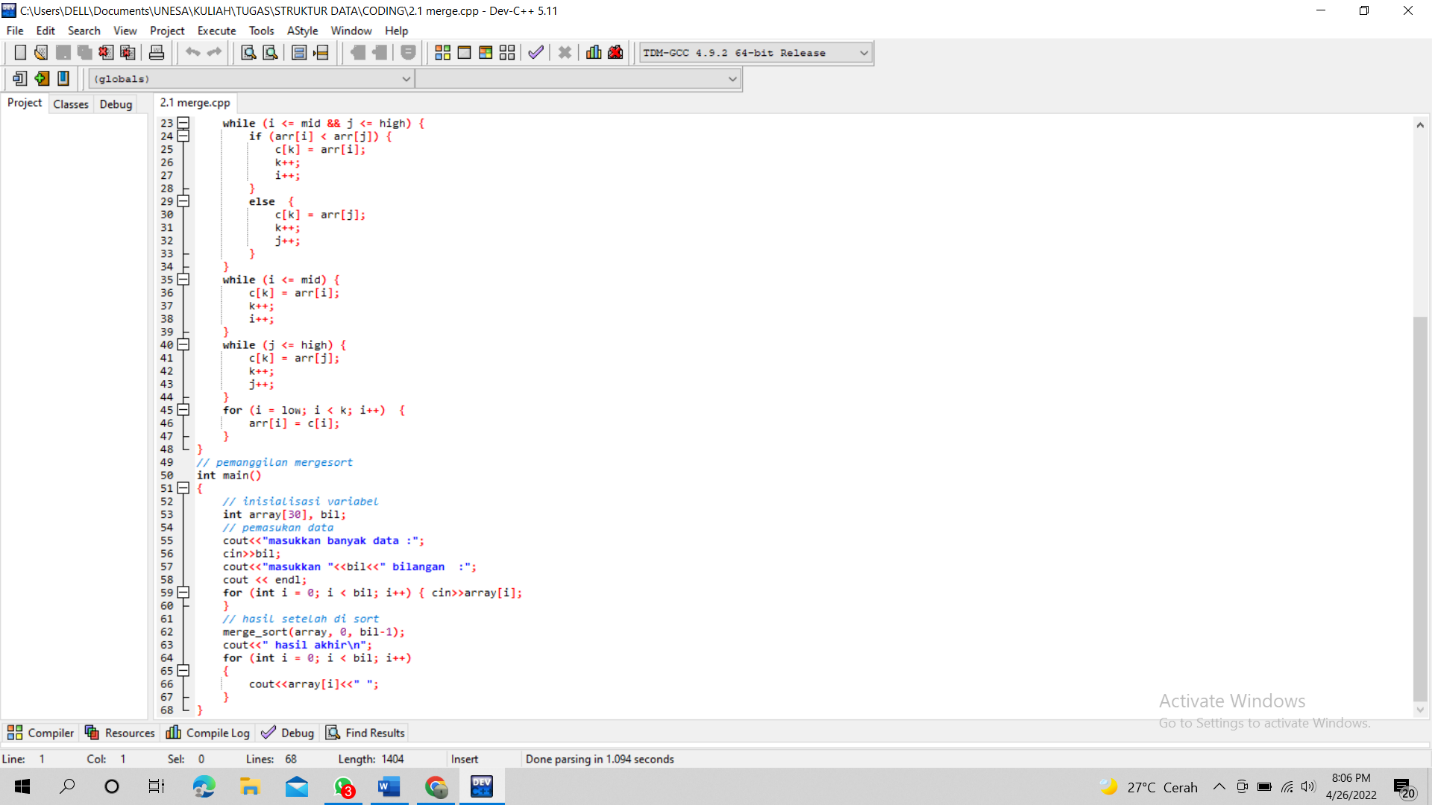
NAMA : ALWAN ROFIQI

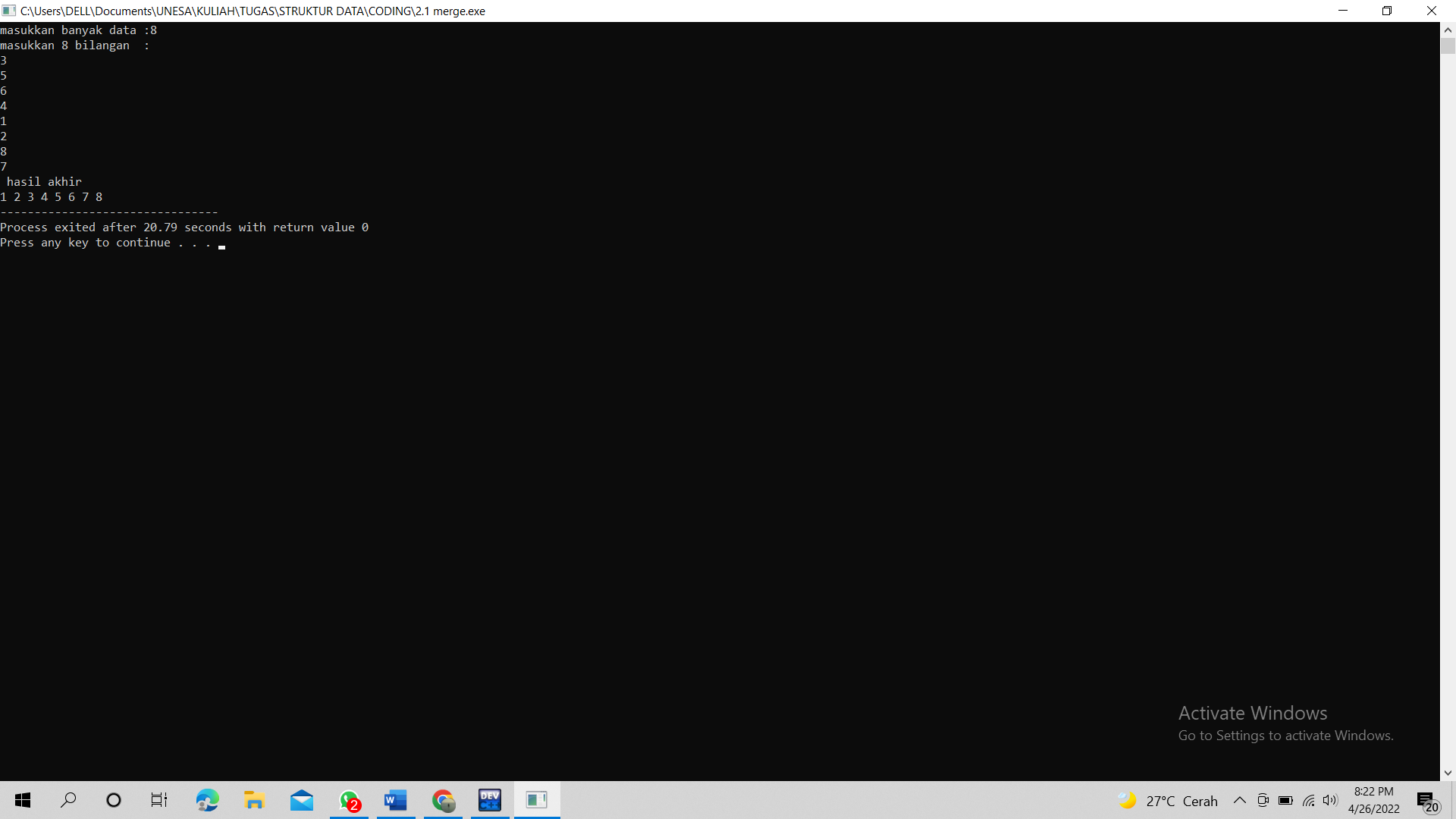
NIM : 21091397020

PRODI/ KELAS : D4 MANAJEMEN INFORMATIKA/ 2021 B

SCRIPT MERGE SORT



OUPUT



PENJELASAN PROSES

1. Pertama inputkan jumlah data yang akan disorting pada line 55-56
2. Kemudian inputkan data yang akan disorting sebanyak jumlah data yang diinputkan diawal script pada line 57
3. Pada hasil output diatas telah diinputkan data sebanyak 8 yaitu 3, 5, 6, 4, 1, 2, 8, 7
4. Proses sorting yang pertama kali larik tersebut dibagi menjadi dua bagian, {3,5,6,4} dan {1,2,8,7}, Kedua larik kemudian diurutkan secara terpisah sehingga menjadi {3,5}, {6,4}, {1, 2}, dan {8, 7}, Ketiga larik kemudian diurutkan secara terpisah sehingga menjadi {3}, {5}, {6}, {4}, {1}, {2}, {8}, dan {7}, Sebuah larik baru dibentuk yang sebagai penggabungan dari setiap dua larik dan diurutkan, sehingga masing-masing larik memilik nilai {3, 5}, {4,6}, {1,2}, dan {7,8} Bentuk larik baru lagi yang merupakan penggabungan dari setiap dua larik dan diurutkan, sehingga masing-masing lari memilik nilai {3,4, 5, 6} dan {1,2, 7, 8}, Langkah berikutnya adalah penggabungan dari masing-masing larik ke dalam larik baru yang dibuat sebelumnya, sehingga memiliki nilai {1, 2, 4, 5, 6, 7, 8}.

MENGHITUNG BIG O KOMPLEKSITAS WAKTU

Kompleksitas algoritma untuk larik dengan n elemen dan jumlah pergeseran (T) dihitung melalui relasi rekursif berikut ini: T(1) = 0 T(n) = 2T ( )+ 2n Adapun M(n) dihitung lewat cara berikut: (a) T(n) = 2( ( ) ( )) 2n = 4T( )+4n = 4( ( ) ( )) + 4n = 8T( )+6n = 2i T( )+2in Memilih I = log n sedemikian sehingga n= 2i, maka diperoleh: (b) T(n) = 2 i T( )+2in = nT(1)+2n. log n = 2n .log n = Ο(n log n) Kasus terburuk (worst case) terjadi bila selama pemanggilan fungsi setiap rekursif merge, nilai terbesar dari setiap elemen terletak di larik yang berbeda. Hal ini memaksa fungsi merge untuk melakukan pengurutan secara berpindah-pindah antar larik, Kondisi Worst Case Pada Merge Sort Maka kompleksitas pada kondisi worst case adalah O (n log n). Kasus terbaik (best case) untuk metode ini dijumpai pada kondisi dimana elemen memiliki nilai terbesar yang lebih kecil dibandingkan dengan seluruh nilai pada elemen yang lain, Kondisi Best Case Pada Merge Sort Pada skenario ini hanya n/2 perbandingan dari elemen yang diperlukan. Menggunakan proses perhitungan yang sama sebagaimana dalam kasus terburuk Dengan kata lain, diperoleh juga kompleksitas yang sama, O (n log n) [6].

KELEBIHAN DAN KEKURANGAN MERGE SORT

KELEBIHAN :

1. Dibanding dengan algoritma lain, merge sort ini termasuk algoritma yang sangat efisien dalam penggunaannya sebab setiap list selalu dibagi bagi menjadi list yang lebih kecil, kemudian digabungkan lagi sehingga tidak perlu melakukan banyak perbandingan.
2. Cocok untuk sorting akses datanya lambat misalnya tape drive atau hard disk.
3. Cocok untuk sorting data yang biasanya diakses secara sequentially (berurutan),  
   misalnya linked list, tape drive, dan hard disk.

KEKURANGAN :

1. Kekurangan Merge Sort yaitu terlalu banyak menggunakan ruang pada memori.
2. Merge Sort membutuhkan lebih banyak ruang daripada jenis sorting lainnya.