**Laporan Individu**

**Merge SORT**



Muhammad Al Faiz Putra Jalasenandra

21091397072

D4 MANAJEMEN INFORMATIKA

1 A. Kodingan merge sort

#include <iostream>

using namespace std;

void merge(int arr[], int l, int m, int r)

{

int i, j, k;

int n1 = m - l + 1;

int n2 = r - m;

int L[n1], R[n2];

for (i = 0; i < n1; i++)

L[i] = arr[l + i];

for (j = 0; j < n2; j++)

R[j] = arr[m + 1 + j];

i = 0;

j = 0;

k = l;

while (i < n1 && j < n2)

{

if (L[i] <= R[j])

{

arr[k] = L[i];

i++;

}

else

{

arr[k] = R[j];

j++;

}

k++;

}

while (i < n1)

{

arr[k] = L[i];

i++;

k++;

}

while (j < n2)

{

arr[k] = R[j];

j++;

k++;

}

}

void mergeSort(int arr[], int l, int r)

{

if (l < r)

{

int m = l + (r - l) / 2;

mergeSort(arr, l, m);

mergeSort(arr, m + 1, r);

merge(arr, l, m, r);

}

}

void show(int A[], int size)

{

int i;

for (i = 0; i < size; i++)

cout << A[i] << " ";

}

int main()

{

int size;

cout << "\nMasukan Banyak Data : ";

cin >> size;

int arr[size];

for (int i = 0; i < size; ++i)

{

cout << "\nMasukan Data array ke "<<i<<" :";

cin >> arr[i];

}

mergeSort(arr, 0, size);

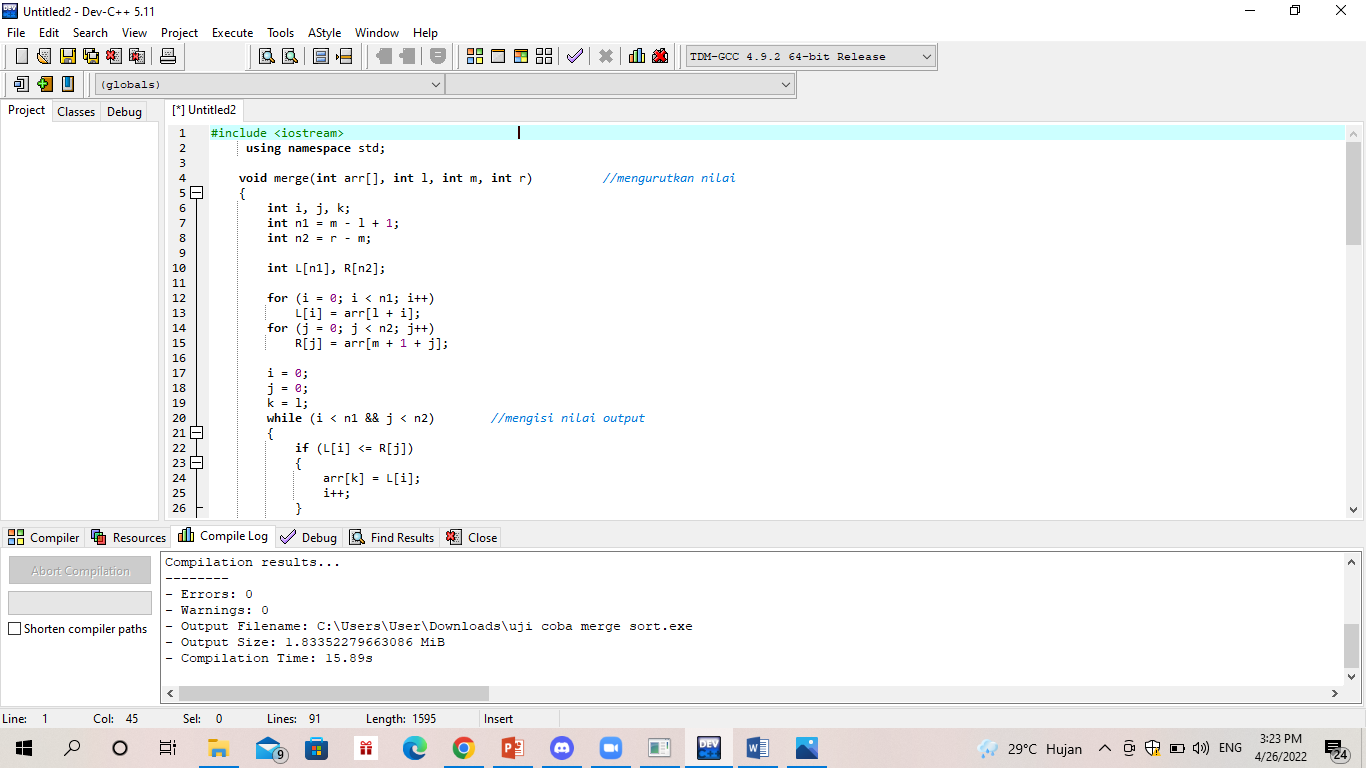
cout << "Hasil\n";

show(arr, size);

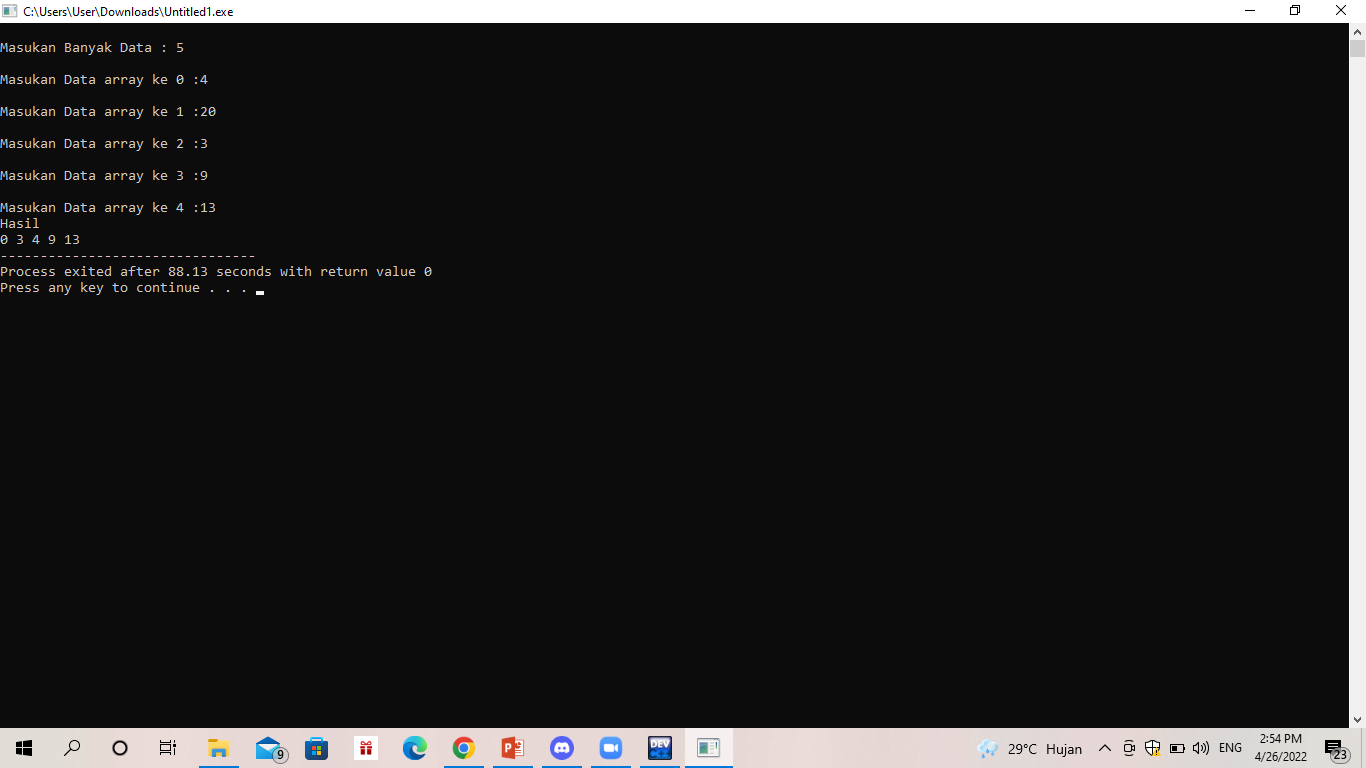
return 0;

}

B. Laporan penjelasan kodingan



C. Hasil RUN kodingan merge sort



Dalam penyelesaian merge sort,bisa dirumuskan dibagi menjadi 3 langkah

* *Divide*: Memilih/memilah elemen dari data menjadi dua bagian.
* *Conquer*: setiap bagian dengan memanggil prosedur merge sort secara rekursif
* *Kombinasi*: Mengkombinasi/Menggabungkan setiap bagian dari rangkaian data tersebut

dari 3 langkah tersebut kita mendapatkan data secara berurutan, dan proses rekursif akan berhenti jika sudah mencapai elemen dasar

2. Menghitung jenis Big o pada merge sort

a. n=1

b. n=5

c. n=10

jawab :

a. n=1

log (1) = log(1)

= 0

B. n = 5

Log(5) = log(5)

= 0,698

C. n = 10

Log(10) = 1og(2.5)

= log 2+ log 5

= 0,301 + 0,698

= 0,999(mendekati 1)

3. A. Kelebihan merge sort

 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.

 Cocok untuk sorting akses datanya lambat misalnya tape drive atau hard disk.

 Cocok untuk sorting data yang biasanya diakses secara sequentially (berurutan),

 misalnya linked list, tape drive, dan hard disk.

B. Kekurangan merge sort

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