

# LAPORAN MERGE SORT

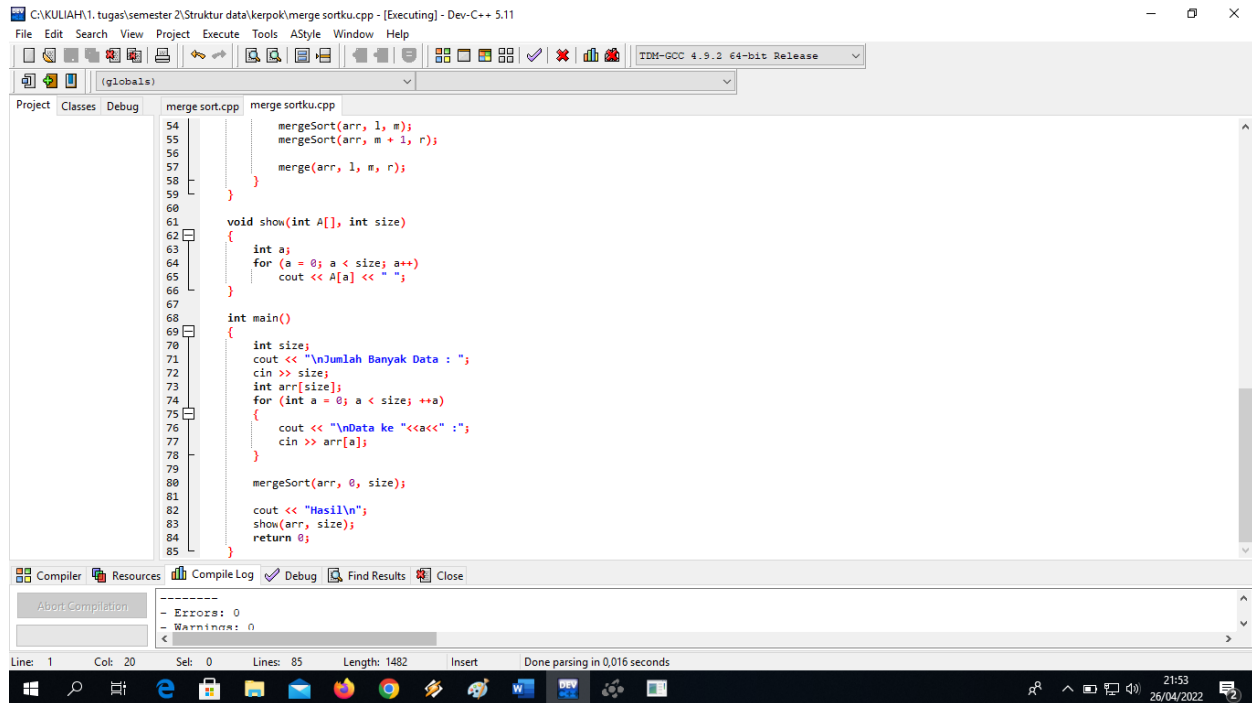
Nama: Affandika Febrian Putra Yunanto

NIM: 030

Kelas: 2021B D4 Manajemen Informatika

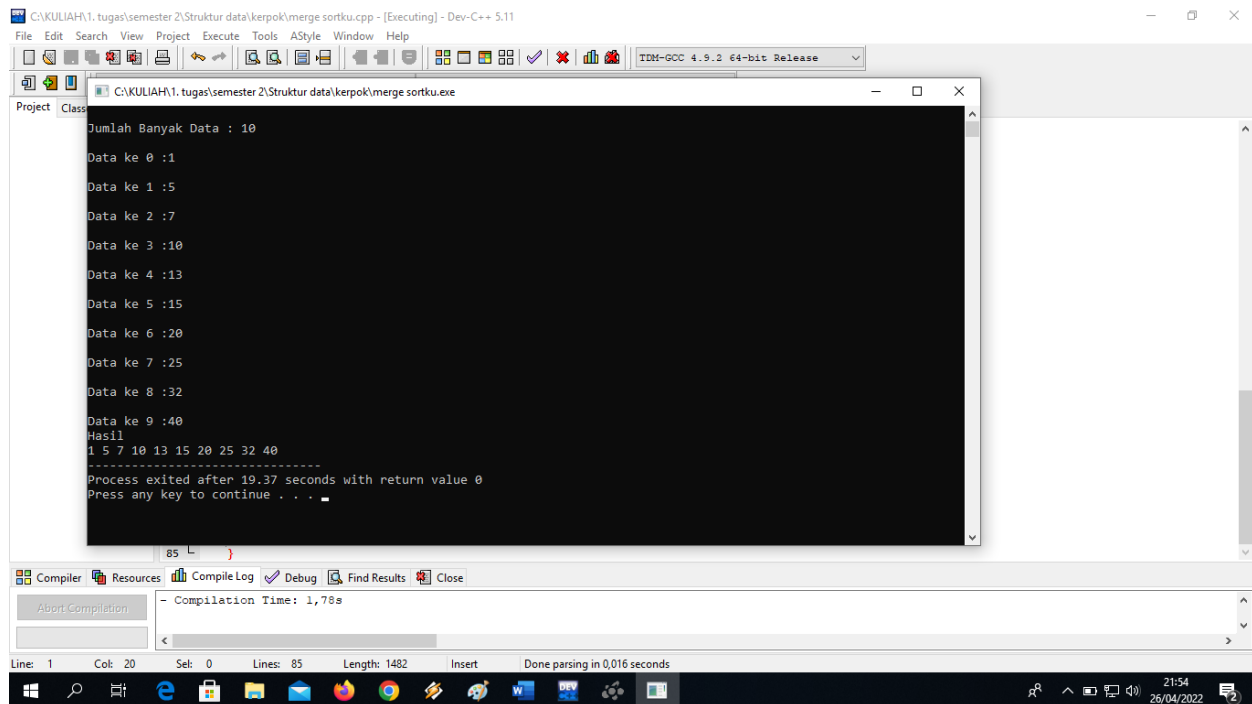
Scrip merge sort:

```
C:\KULIAH\1. tugas\semester 2\Struktur data\kerpok\merge sortku.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug
merge sortku.cpp
31      }
32      c++;
33
34      while (a < n1)
35      {
36          arr[c] = L[a];
37          a++;
38          c++;
39      }
40
41      while (b < n2)
42      {
43          arr[c] = R[b];
44          b++;
45          c++;
46      }
47
48
49      void mergeSort(int arr[], int l, int r)
50      {
51          if (l < r)
52          {
53              int m = l + (r - l) / 2;
54              mergeSort(arr, l, m);
55              mergeSort(arr, m + 1, r);
56
57              merge(arr, l, m, r);
58          }
59
60          void show(int A[], int size)
61          {
62
Compiler Resources Compile Log Debug Find Results Close
About Completion
- Errors: 0
- Warnings: 0
Line: 1 Col: 20 Sel: 0 Lines: 85 Length: 1482 Insert Done parsing in 0,016 seconds
C:\KULIAH\1. tugas\semester 2\Struktur data\kerpok\merge sortku.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug
merge sortku.cpp
1 #include <iostream>
2 using namespace std;
3
4 void merge(int arr[], int l, int m, int r)
5 {
6     int a, b, c;
7     int n1 = m - l + 1;
8     int n2 = r - m;
9     int L[n1], R[n2];
10
11     for (a = 0; a < n1; a++)
12         L[a] = arr[l + a];
13     for (b = 0; b < n2; b++)
14         R[b] = arr[m + 1 + b];
15
16     a = 0;
17     b = 0;
18     c = l;
19     while (a < n1 && b < n2)
20     {
21         if (L[a] <= R[b])
22         {
23             arr[c] = L[a];
24             a++;
25         }
26         else
27         {
28             arr[c] = R[b];
29             b++;
30         }
31         c++;
32     }
Compiler Resources Compile Log Debug Find Results Close
About Completion
- Errors: 0
- Warnings: 0
Line: 1 Col: 20 Sel: 0 Lines: 85 Length: 1482 Insert Done parsing in 0,016 seconds
```



```
54 mergeSort(arr, l, m);
55 mergeSort(arr, m + 1, r);
56
57 }
58 merge(arr, l, m, r);
59
60 }
61
62 void show(int A[], int size)
63 {
64     int a;
65     for (a = 0; a < size; a++)
66         cout << A[a] << " ";
67
68 }
69
70 int main()
71 {
72     int size;
73     cout << "\nJumlah Banyak Data : ";
74     cin >> size;
75     int arr[size];
76     for (int a = 0; a < size; ++a)
77     {
78         cout << "\nData ke "<<a<<" : ";
79         cin >> arr[a];
80     }
81
82     mergeSort(arr, 0, size);
83
84     cout << "Hasil\n";
85     show(arr, size);
86     return 0;
87 }
```

## HASIL RUN:



```
Jumlah Banyak Data : 10
Data ke 0 : 1
Data ke 1 : 5
Data ke 2 : 7
Data ke 3 : 10
Data ke 4 : 13
Data ke 5 : 15
Data ke 6 : 20
Data ke 7 : 25
Data ke 8 : 32
Data ke 9 : 40
Hasil
1 5 7 10 13 15 20 25 32 40
-----
Process exited after 19.37 seconds with return value 0
Press any key to continue . . .
```

## Penjelasan :

Data yang diinputkan akan dibagi menjadi dua, data lalu diurutkan dengan metode merge sort line, output menampilkan jumlah banyak data yang akan diproses dan menampilkan data yang akan diurutkan menggunakan metode merge sort line dan keluar hasil urutannya.

Kelebihan :

1. Waktu pengerjaan yang konsisten
2. Performa sangat bagus untuk List yang memiliki banyak index

Kekurangan :

1. Performa buruk untuk list index sedikit dibanding dengan bubble sort dan insertion sort
2. Jika data sudah tersorting dari awal, maka akan tetap melakukan sorting dari awal
3. Menggunakan memory yang lebih untuk melakukan split data