

Nama : Al Zuchri Mufashal Q

NIM : 21091397062

Kelas : 2021 B

LAPORAN INDIVIDU

❖ Merge Sort

- Merge sort adalah algoritma sorting yang berdasarkan dengan cara membagi dan menggabungkan. Berbeda dengan selection sort, insertion sort, dan bubble sort, algoritma merge sort memiliki kompleksitas yang lebih kecil yakni $O(n \log n)$. Algoritma ini membagi bilangan atau array sampai tidak dapat dibagi kembali dan mengurutkannya setiap bagian. Dengan cara ini kita dapat memperoleh jumlah perbandingan yang lebih sedikit daripada sorting menggunakan selection sort, bubble sort dan insertion sort.
- Algoritma merge sort :
 1. Bagi array bilangan sampai tidak dapat dibagi kembali
 2. Urutkan bilangan yang telah dibagi
 3. Gabungkan kembali bilangan yang telah dibagi
 4. Array bilangan akan terurut.

❖ Coding

```
1 #include <iostream>
2 using namespace std;
3
4 void merge(int arr[], int l, int m, int r)
5 {
6     int i, j, k;
7     int n1 = m - l + 1;
8     int n2 = r - m;
9
10    int L[n1], R[n2];
11
12    for (i = 0; i < n1; i++)
13        L[i] = arr[l + i];
14    for (j = 0; j < n2; j++)
15        R[j] = arr[m + 1 + j];
16
17    i = 0;
18    j = 0;
19    k = l;
20    while (i < n1 && j < n2)
21    {
22        if (L[i] <= R[j])
```

```
22        if (L[i] <= R[j])
23        {
24            arr[k] = L[i];
25            i++;
26        }
27        else
28        {
29            arr[k] = R[j];
30            j++;
31        }
32        k++;
33    }
34
35    while (i < n1)
36    {
37        arr[k] = L[i];
38        i++;
39        k++;
40    }
41
42    while (j < n2)
43    {
```

```
43    {
44        arr[k] = R[j];
45        j++;
46        k++;
47    }
48 }
49
50 void mergeSort(int arr[], int l, int r)
51 {
52     if (l < r)
53     {
54
55         int m = l + (r - l) / 2;
56
57         mergeSort(arr, l, m);
58         mergeSort(arr, m + 1, r);
59
60         merge(arr, l, m, r);
61     }
62 }
63
64 void show(int A[], int size)
```

```

67     for (i = 0; i < size; i++)
68         cout << A[i] << " ";
69     }
70
71     int main()
72     {
73         int size;
74         cout << "\nMasukan Banyak Data : ";
75
76         cin >> size;
77
78         int arr[size];
79
80         for (int i = 0; i < size; ++i)
81         {
82             cout << "\nMasukan Data array ke "<<i<<" :";
83             cin >> arr[i];
84         }
85
86         mergeSort(arr, 0, size);
87
88         cout << "Hasil\n";

```

```

71     int main()
72     {
73         int size;
74         cout << "\nMasukan Banyak Data : ";
75
76         cin >> size;
77
78         int arr[size];
79
80         for (int i = 0; i < size; ++i)
81         {
82             cout << "\nMasukan Data array ke "<<i<<" :";
83             cin >> arr[i];
84         }
85
86         mergeSort(arr, 0, size);
87
88         cout << "Hasil\n";
89         show(arr, size);
90         return 0;
91     }

```

```

Masukan Banyak Data : 6
Masukan Data array ke 0 :9
Masukan Data array ke 1 :4
Masukan Data array ke 2 :7
Masukan Data array ke 3 :3
Masukan Data array ke 4 :8
Masukan Data array ke 5 :2
Hasil
2 3 4 7 8 9
-----
Process exited after 43.66 seconds with return value 0
Press any key to continue . . .

```

❖ Kelebihan dan Kekurangan Merge Sort

- Kelebihan merge sort :

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 merge sort :
 1. Merge Sort terlalu banyak menggunakan ruang pada memori
 2. Merge Sort membutuhkan lebih banyak ruang daripada jenis sorting lainnya.