LAPORAN AKHIR PRAKTIKUM

Mata Praktikum	: PERANCANGAN DAN ANALISIS ALGORITMA
Kelas	: 3IA11
Praktikum ke-	: 2
Tanggal	: 4/112/2024
Materi	: Algoritma Devide dan Conquer
NPM	: 51422161
Nama	: MUHAMMAD TARMIDZI BARIQ
Ketua Asisten	: MURAD
Jumlah Lembar	: 4

LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2024

1. Carikan contoh implementasi logika algoritma devide and conquer berikan visualisasinya

Algoritma mengurutkan uang kertas dari terkecil ke terbesar menggunakan devide dan conquer

Mulai dengan array: [10000, 50000, 2000, 100000, 5000, 20000]

1. Divide:

   [10000, 50000, 2000] [100000, 5000, 20000]

2. Divide lagi:

   [10000, 50000] [2000] [100000, 5000] [20000]

3. Divide lagi:

   [10000] [50000] [2000] [100000] [5000] [20000]

4. Conquer:

   [10000, 50000] [2000] [5000, 100000] [20000]

5. Combine:

   [2000, 10000, 50000] [5000, 20000, 100000]

6. Combine semua:

   [2000, 5000, 10000, 20000, 50000, 100000]

main.cpp

Share Run

```cpp
#include <iostream>
#include <vector>

void merge(std::vector<int>& arr, int left, int mid, int right) {
    int n1 = mid - left + 1;
    int n2 = right - mid;

    std::vector<int> L(n1), R(n2);

    for (int i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (int j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];

    int i = 0, j = 0, k = left;
    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++;
    }

```

main.cpp

Share Run

```
    }

    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

void mergeSort(std::vector<int>& arr, int left, int right) {
    if (left >= right)
        return;

    int mid = left + (right - left) / 2;
    mergeSort(arr, left, mid);
    mergeSort(arr, mid + 1, right);
    merge(arr, left, mid, right);
}

int main() {
    std::vector<int> uang_kertas = {10000, 50000, 2000, 100000, 5000, 20000};
    std::cout << "Nominal uang kertas sebelum diurutkan: ";
    for (int num : uang_kertas)
        std::cout << num << " ";
    std::cout << std::endl;

    mergeSort(uang_kertas, 0, uang_kertas.size() - 1);

    std::cout << "Nominal uang kertas setelah diurutkan: ";
    for (int num : uang_kertas)
        std::cout << num << " ";
    std::cout << std::endl;

    return 0;
}

```

Output

Output Clear

```
Nominal uang kertas sebelum diurutkan: 10000 50000 2000 100000 5000 20000
Nominal uang kertas setelah diurutkan: 2000 5000 10000 20000 50000 100000


=== Code Execution Successful ===
```