

## Tugas Pertemuan 2

Martin Caesar Partogi

242310034

TI-24-PA2

1.

1.

Quick sort

```
#include <iostream>
using namespace std;

// Fungsi untuk menukar dua elemen dalam array
void swap (int arr[], int pos1, int pos2) {
    int temp; // Variabel sementara untuk penyimpanan
    temp = arr[pos1]; // Simpan nilai elemen di pos1
    arr[pos1] = arr [pos2]; // Ganti nilai elemen di pos1 dengan nilai elemen di pos2
    arr[pos2] = temp; // Ganti nilai elemen di pos2 dengan nilai yang disimpan di temp
}

// Fungsi untuk mempartisi array untuk quicksort
int partition (int arr[], int low, int high, int pivot) {
    int i = low; // Indeks untuk iterasi array
    int j = low; // Indeks untuk elemen yang lebih kecil dari pivot
    while (i <= high) { // Iterasi dari low hingga high
        if (arr[i] > pivot) { // Jika elemen saat ini lebih besar dari pivot
            i++; // Pindahkan indeks i ke elemen berikutnya
        }
        else { // Jika elemen saat ini kurang dari atau sama dengan pivot
            swap (arr, i, j); // Tukar elemen di i dan j
            i++; // Pindahkan indeks i ke elemen berikutnya
            j++; // Pindahkan indeks j ke elemen berikutnya (elemen yang lebih kecil dari pivot)
        }
    }
    return j-1; // Kembalikan posisi pivot setelah partisi
}

// Fungsi quicksort rekursif
void quicksort(int arr[], int low, int high) {
    if (low < high) { // Jika ada lebih dari satu elemen dalam sub-array
        int pivot = arr[high]; // Pilih elemen terakhir sebagai pivot
        int pos = partition(arr, low, high, pivot); // Partisi array dan dapatkan posisi pivot

        quicksort(arr, low, pos-1); // Urutkan sub-array kiri dari pivot
        quicksort(arr, pos+1, high); // Urutkan sub-array kanan dari pivot
    }
}

int main ()
{
    int n; // Variabel untuk menyimpan panjang array
    cout << "Tentukan panjang array = ";
    cin>> n; // Baca panjang array dari input

    int arr[n]; // Deklarasikan array dengan panjang n (perhatikan bahwa ini adalah VLA, yang tidak standar di C++)
    for (int i = 0; i < n; i++) { // Loop untuk membaca elemen-elemen array
```

```

        quicksort(arr, low, pos-1); // Urutkan sub-array kiri dari pivot
        quicksort(arr, pos+1, high); // Urutkan sub-array kanan dari pivot
    }
}

int main ()
{
    int n; // Variabel untuk menyimpan panjang array
    cout << "Tentukan panjang array = ";
    cin >> n; // Baca panjang array dari input

    int arr[n]; // Deklarasikan array dengan panjang n (perhatikan bahwa ini adalah VLA, yang tidak standar di C++)
    for (int i = 0; i < n; i++) { // Loop untuk membaca elemen-elemen array
        cin >> arr[i]; // Baca elemen array dari input
    }
    quicksort(arr, 0, n-1); // Panggil fungsi quicksort untuk mengurutkan array
    cout << "Berikutnya adalah array yang telah di sortir = ";
    for (int i = 0; i < n; i++) { // Loop untuk mencetak elemen-elemen array yang sudah diurutkan
        cout << arr[i] << "\t"; // Cetak elemen array diikuti dengan tab
    }
    cout << endl; // Cetak newline untuk baris baru.
    return 0; // Kembalikan 0 untuk menunjukkan eksekusi berhasil
}

```

## Merge sort

```
1  #include <iostream>
2  using namespace std;
3
4  // Fungsi untuk menggabungkan dua subarray yang sudah terurut menjadi satu subarray terurut
5  void merge(int arr[], int l, int m, int r) {
6      int x, y, z; // Indeks untuk iterasi array
7      int n1 = m - l + 1; // Ukuran subarray kiri
8      int n2 = r - m; // Ukuran subarray kanan
9
10     int L[n1], R[n2]; // Array sementara untuk subarray kiri dan kanan
11
12     // Salin data ke array sementara L[] dan R[]
13     for (x = 0; x < n1; x++)
14         L[x] = arr[l + x];
15     for (y = 0; y < n2; y++)
16         R[y] = arr[m + 1 + y];
17
18     x = 0; // Indeks awal subarray pertama
19     y = 0; // Indeks awal subarray kedua
20     z = l; // Indeks awal subarray yang digabungkan
21
22     // Gabungkan array sementara kembali ke arr[l..r]
23     while (x < n1 && y < n2) {
24         if (L[x] <= R[y]) { // Jika elemen L[x] lebih kecil atau sama dengan R[y]
25             arr[z] = L[x]; // Salin L[x] ke arr[z]
26             x++; // Pindahkan indeks subarray kiri
27         } else { // Jika elemen R[y] lebih kecil dari L[x]
28             arr[z] = R[y]; // Salin R[y] ke arr[z]
29             y++; // Pindahkan indeks subarray kanan
30         }
31         z++; // Pindahkan indeks subarray yang digabungkan
32     }
33
34     // Salin elemen sisa dari L[], jika ada
35     while (x < n1) {
36         arr[z] = L[x];
37         x++;
38         z++;
39     }
40
41     // Salin elemen sisa dari R[], jika ada
42     while (y < n2) {
43         arr[z] = R[y];
44         y++;
45         z++;
46     }
47 }
```

```

47 }
48
49 // Fungsi utama merge sort rekursif
50 void mergeSort(int arr[], int l, int r) {
51     if (l < r) { // Jika ada lebih dari satu elemen
52         int m = l + (r - l) / 2; // Temukan titik tengah
53
54         // Urutkan setengah pertama dan setengah kedua
55         mergeSort(arr, l, m);
56         mergeSort(arr, m + 1, r);
57
58         // Gabungkan setengah yang diurutkan
59         merge(arr, l, m, r);
60     }
61 }
62
63 // Fungsi untuk menampilkan array
64 void show(int A[], int size) {
65     for (int i = 0; i < size; i++)
66         cout << A[i] << " ";
67 }
68
69 int main() {
70     int size; // Variabel untuk menyimpan ukuran array
71
72     // Fungsi untuk menampilkan array
73     void show(int A[], int size) {
74         for (int i = 0; i < size; i++)
75             cout << A[i] << " ";
76     }
77
78     int main() {
79         int size; // Variabel untuk menyimpan ukuran array
80         cout << "\nMasukan Banyak Data : ";
81         cin >> size; // Baca ukuran array dari input
82
83         int arr[size]; // Deklarasikan array dengan ukuran yang ditentukan (VLA warning)
84         for (int i = 0; i < size; i++) {
85             cout << "\nMasukan Data array ke-" << i << " : ";
86             cin >> arr[i]; // Baca elemen array dari input
87         }
88
89         mergeSort(arr, 0, size - 1); // Panggil fungsi merge sort untuk mengurutkan array
90
91         cout << "\nHasil\n";
92         show(arr, size); // Tampilkan array yang sudah diurutkan
93
94         return 0; // Kembalikan 0 untuk menunjukkan eksekusi berhasil
95     }
96 }

```

2.

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  // Fungsi untuk menukar dua elemen
7  void swap(int &a, int &b) {
8      int temp = a;
9      a = b;
10     b = temp;
11 }
12
13 // Fungsi untuk membagi array berdasarkan pivot
14 int partition(vector<int> &arr, int low, int high) {
15     int pivot = arr[high]; // Pivot diambil dari elemen terakhir
16     int i = low - 1;
17
18     for (int j = low; j < high; j++) {
19         if (arr[j] < pivot) { // Jika elemen lebih kecil dari pivot, tukar
20             i++;
21             swap(arr[i], arr[j]);
22         }
23     }
24     swap(arr[i + 1], arr[high]); // Menempatkan pivot di posisi yang benar
25     return (i + 1);
26 }
27
28 // Fungsi rekursif untuk melakukan Quick Sort
29 void quickSort(vector<int> &arr, int low, int high) {
30     if (low < high) {
31         int pivotIndex = partition(arr, low, high);
32
33         // Rekursif untuk bagian kiri dan kanan dari pivot
34         quickSort(arr, low, pivotIndex - 1);
35         quickSort(arr, pivotIndex + 1, high);
36     }
37 }
38
39 // Fungsi untuk mencetak array
40 void printArray(const vector<int> &arr) {
41     for (size_t i=0; i < arr.size(); i++) {
42         cout << arr[i] << " ";
43     }
44     cout << endl;
45 }
46
47 int main() {
48     vector<int> arr;
```



```

49     int n, element;
50
51     // Output header dan meminta input dari user
52     cout << "Program Quick Sort\n";
53     cout << "Masukkan jumlah elemen: ";
54     cin >> n;
55
56     // Mengambil input array dari user
57     cout << "Masukkan elemen array: ";
58     for (int i = 0; i < n; i++) {
59         cin >> element;
60         arr.push_back(element);
61     }
62
63     // Menampilkan array sebelum diurutkan
64     cout << "Array sebelum diurutkan: ";
65     printArray(arr);
66
67     // Melakukan Quick Sort
68     quickSort(arr, 0, n - 1);
69
70     // Menampilkan hasil array yang sudah diurutkan
71     cout << "Array setelah diurutkan: ";
72     printArray(arr);
73
74     return 0;
75 }

```

```

Program Quick Sort
Masukkan jumlah elemen: 5
Masukkan elemen array: 3
4
1
2
5
Array sebelum diurutkan: 3 4 1 2 5
Array setelah diurutkan: 1 2 3 4 5

-----
Process exited after 12.03 seconds with return value 0
Press any key to continue . . .

```

2.

a.

```
1  #include <iostream>
2  using namespace std;
3
4  void printArray(int arr[], int size) {
5      cout << "[";
6      for (int i = 0; i < size; i++) {
7          cout << arr[i] << (i < size - 1 ? " " : "");
8      }
9      cout << "]" << endl;
10 }
11
12 int partition(int arr[], int low, int high, int size) {
13     int pivot = arr[high];
14     int i = low - 1;
15
16     for (int j = low; j < high; j++) {
17         if (arr[j] > pivot) {
18             i++;
19             swap(arr[i], arr[j]);
20         }
21     }
22     swap(arr[i + 1], arr[high]);
23
24     printArray(arr, size);
```

```

25     return (i + 1);
26 }
27
28 void quickSort(int arr[], int low, int high, int size) {
29     if (low < high) {
30         int pi = partition(arr, low, high, size);
31
32         quickSort(arr, low, pi - 1, size);
33         quickSort(arr, pi + 1, high, size);
34     }
35 }
36
37 int main() {
38     int numElements;
39     cout << "Data yang akan di sort : ";
40     cin >> numElements;
41
42     int arr[numElements];
43     cout << "Masukkan " << numElements << " angka: ";
44     for (int i = 0; i < numElements; i++) {
45         cin >> arr[i];
46     }
47
48     cout << "Quick Sort :" << endl;

```



```

30         int pi = partition(arr, low, high, size);
31
32         quickSort(arr, low, pi - 1, size);
33         quickSort(arr, pi + 1, high, size);
34     }
35 }
36
37 int main() {
38     int numElements;
39     cout << "Data yang akan di sort : ";
40     cin >> numElements;
41
42     int arr[numElements];
43     cout << "Masukkan " << numElements << " angka: ";
44     for (int i = 0; i < numElements; i++) {
45         cin >> arr[i];
46     }
47
48     cout << "Quick Sort :" << endl;
49     printArray(arr, numElements);
50
51     quickSort(arr, 0, numElements - 1, numElements);
52     return 0;
53 }

```

```

Data yang akan di sort : 9
Masukkan 9 angka: 8
3
4
9
2
6
5
1
7
Quick Sort :
[8 3 4 9 2 6 5 1 7]
[8 9 7 3 2 6 5 1 4]
[9 8 7 3 2 6 5 1 4]
[9 8 7 6 5 4 2 1 3]
[9 8 7 6 5 4 2 1 3]
[9 8 7 6 5 4 3 1 2]
[9 8 7 6 5 4 3 2 1]

-----
Process exited after 12.79 seconds with return value 0
Press any key to continue . . .

```

b.

```
1  #include <iostream>
2  using namespace std;
3
4  void merge(int arr[], int left, int mid, int right) {
5      int n1 = mid - left + 1;
6      int n2 = right - mid;
7      int leftArr[n1], rightArr[n2];
8
9      for (int i = 0; i < n1; i++) leftArr[i] = arr[left + i];
10     for (int i = 0; i < n2; i++) rightArr[i] = arr[mid + 1 + i];
11
12     int i = 0, j = 0, k = left;
13     while (i < n1 && j < n2) {
14         if (leftArr[i] <= rightArr[j]) {
15             arr[k++] = leftArr[i++];
16         } else {
17             arr[k++] = rightArr[j++];
18         }
19     }
20
21     while (i < n1) arr[k++] = leftArr[i++];
22     while (j < n2) arr[k++] = rightArr[j++];
23
24     cout << "Menggabungkan List [";
25     for (int i = left; i <= right; i++) {
26         cout << arr[i] << (i < right ? ", " : "");
27     }
28     cout << "]\n";
29 }
30
31 void mergeSort(int arr[], int left, int right) {
32     if (left < right) {
33         int mid = left + (right - left) / 2;
34
```

```

33         int mid = left + (right - left) / 2;
34
35         cout << "Pecah List [";
36         for (int i = left; i <= right; i++) {
37             cout << arr[i] << (i < right ? ", " : "");
38         }
39         cout << "]\n";
40
41         mergeSort(arr, left, mid);
42         mergeSort(arr, mid + 1, right);
43         merge(arr, left, mid, right);
44     }
45 }
46
47 int main() {
48     int size;
49     cout << "Masukkan jumlah elemen: ";
50     cin >> size;
51
52     int data[size];
53     cout << "Masukkan " << size << " elemen: ";
54     for (int i = 0; i < size; i++) {
55         cin >> data[i];
56     }
57
58     cout << "Input data: [";
59     for (int i = 0; i < size; i++) {
60         cout << data[i] << (i < size - 1 ? ", " : "");
61     }
62     cout << "]\n";
63
64     mergeSort(data, 0, size - 1);
65     return 0;
66 }

```

```
Masukkan jumlah elemen: 10
Masukkan 10 elemen: 9
2
6
5
8
3
7
4
0
1
Input data: [9, 2, 6, 5, 8, 3, 7, 4, 0, 1]
Pecah List [9, 2, 6, 5, 8, 3, 7, 4, 0, 1]
Pecah List [9, 2, 6, 5, 8]
Pecah List [9, 2, 6]
Pecah List [9, 2]
Menggabungkan List [2, 9]
Menggabungkan List [2, 6, 9]
Pecah List [5, 8]
Menggabungkan List [5, 8]
Menggabungkan List [2, 5, 6, 8, 9]
Pecah List [3, 7, 4, 0, 1]
Pecah List [3, 7, 4]
Pecah List [3, 7]
Menggabungkan List [3, 7]
Menggabungkan List [3, 4, 7]
Pecah List [0, 1]
Menggabungkan List [0, 1]
Menggabungkan List [0, 1, 3, 4, 7]
Menggabungkan List [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

-----
Process exited after 21.28 seconds with return value 0
Press any key to continue . . .
```