

Nama : Muchamad Rofii

NPM : 242310030

Kelas : TI-24-PA

Matkul : Desain Analisis Algoritma

Quicksort Asc:

```
Penjelasan Quicksort Secara Ascending.cpp
1  #include <iostream>
2  using namespace std;
3
4  // Fungsi untuk menukar dua nilai dalam array
5  void swap (int arr[], int pos1, int pos2) {
6      int temp; // Variabel sementara untuk menyimpan nilai
7      temp = arr[pos1]; // Simpan nilai dari posisi pertama
8      arr[pos1] = arr[pos2]; // Pindahkan nilai dari posisi kedua ke posisi pertama
9      arr[pos2] = temp; // Masukkan nilai yang disimpan tadi ke posisi kedua
10 }
11
12 // Fungsi untuk membagi array menjadi dua bagian untuk quicksort
13 int partition (int arr[], int low, int high, int pivot) {
14     int i = low; // Penunjuk untuk elemen yang sedang dicek
15     int j = high; // Penunjuk untuk tempat menyimpan elemen yang lebih kecil dari pivot
16
17     // Proses pengecekan elemen dari posisi low sampai high
18     while (i <= high) {
19         if (arr[i] > pivot) {
20             // Jika nilai sekarang lebih besar dari pivot, lanjut ke elemen berikutnya
21             i++;
22         }
23         else {
24             // Jika nilai sekarang lebih kecil atau sama dengan pivot,
25             // tukar posisinya ke bagian "kiri" array
26             swap(arr, i, j);
27             i++;
28             j--;
29         }
30     }
31     return j - 1; // Kembalikan posisi akhir dari elemen yang jadi pembatas
32 }
33
34 // Fungsi utama quicksort yang bekerja secara rekursif
35 void quicksort (int arr[], int low, int high) {
36     if (low < high) { // Selama masih ada lebih dari satu elemen
37         int pivot = arr[high]; // Ambil elemen terakhir sebagai pivot
38         int pos = partition(arr, low, high, pivot); // Bagi array dan dapatkan posisi pivot yang baru
39
40         // Lanjutkan quicksort untuk bagian kiri dan kanan dari pivot
41         quicksort(arr, low, pos - 1); // Urutkan bagian kiri
42         quicksort(arr, pos + 1, high); // Urutkan bagian kanan
43     }
44 }
45
46 int main () {
47     int n; // Banyaknya jumlah elemen dalam array
48     cout << "Tentukan panjang array = ";
49     cin >> n; // Input jumlah elemen dari pengguna
50
51     int arr[n]; // Buat array dengan panjang sesuai input
52
53     // Input nilai-nilai untuk array
54     for (int i = 0; i < n; i++) {
55         cin >> arr[i];
56     }
57
58     // Panggil fungsi quicksort untuk mengurutkan array
59     quicksort(arr, 0, n - 1);
60
61     // Tampilkan hasil array yang sudah diurutkan
62     cout << "Berikutnya adalah array yang telah di sortir = ";
63     for (int i = 0; i < n; i++) {
64         cout << arr[i] << " ";
65     }
66     cout << endl; // Baris baru untuk merapikan tampilan
67     return 0; // Program selesai dengan sukses
68 }
```

```
C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\Penjelasan Quicksort Se...
321
123
413
532
Berikutnya adalah array yang telah di sortir = 123    321    413    532

-----
Process exited after 9.7 seconds with return value 0
Press any key to continue . . .
```

MergeSort:

Penjelasan MergeSort Secara Ascending.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  // Fungsi untuk menggabungkan dua subarray yang telah terurut menjadi satu subarray yang juga terurut
5  void merge(int arr[], int l, int m, int r) {
6      int x, y, z; // Indeks untuk proses iterasi
7      int n1 = m - l + 1; // Jumlah elemen pada subarray sebelah kiri
8      int n2 = r - m;      // Jumlah elemen pada subarray sebelah kanan
9
10     int L[n1], R[n2]; // Array sementara untuk menyimpan elemen-elemen dari masing-masing subarray
11
12     // Menyalin elemen dari array utama ke dalam 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 untuk L[]
19     y = 0; // Indeks awal untuk R[]
20     z = l; // Indeks awal untuk array gabungan arr[]
21
22     // Proses penggabungan elemen dari L[] dan R[] ke dalam arr[] dengan urutan yang terjaga
23     while (x < n1 && y < n2) {
24         if (L[x] <= R[y]) { // Jika elemen di L lebih kecil atau sama dengan elemen di R
25             arr[z] = L[x]; // Salin elemen dari L ke arr
26             x++;           // Pindah ke elemen berikutnya di L
27         } else {           // Jika elemen di R lebih kecil dari elemen di L
28             arr[z] = R[y]; // Salin elemen dari R ke arr
29             y++;           // Pindah ke elemen berikutnya di R
30         }
31         z++; // Lanjutkan ke posisi berikutnya dalam array gabungan
32     }
33 }
```

Penjelasan MergeSort Secara Ascending.cpp

```
33
34 // Menyalin sisa elemen dari L[], jika masih ada
35 while (x < n1) {
36     arr[z] = L[x];
37     x++;
38     z++;
39 }
40
41 // Menyalin sisa elemen dari R[], jika masih ada
42 while (y < n2) {
43     arr[z] = R[y];
44     y++;
45     z++;
46 }
47 }
48
49 // Fungsi utama untuk melakukan merge sort secara rekursif
50 void mergeSort(int arr[], int l, int r) {
51     if (l < r) { // Jika masih terdapat Lebih dari satu elemen dalam subarray
52         int m = l + (r - l) / 2; // Menentukan titik tengah dari subarray
53
54         // Memanggil mergeSort secara rekursif untuk bagian kiri dan kanan
55         mergeSort(arr, l, m);
56         mergeSort(arr, m + 1, r);
57
58         // Menggabungkan dua bagian yang telah diurutkan
59         merge(arr, l, m, r);
60     }
61 }
62 }
```

```

62
63 // Fungsi untuk menampilkan isi 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 jumlah elemen array
71     cout << "\nMasukan Banyak Data : ";
72     cin >> size; // Membaca jumlah elemen dari input pengguna
73
74     int arr[size]; // Mendeklarasikan array dengan panjang sesuai input
75
76     // Input nilai-nilai array dari pengguna
77     for (int i = 0; i < size; i++) {
78         cout << "\nMasukan Data array ke-" << i << " : ";
79         cin >> arr[i];
80     }
81
82     // Memanggil fungsi merge sort untuk mengurutkan array
83     mergeSort(arr, 0, size - 1);
84
85     // Menampilkan hasil array yang telah diurutkan
86     cout << "\nHasil\n";
87     show(arr, size);
88
89     return 0; // Mengembalikan 0 menandakan bahwa program telah selesai dijalankan dengan sukses
90 }

```

C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\Penjelasan MergeSort Secara Ascending.exe

Masukan Banyak Data : 4

Masukan Data array ke-0 : 31

Masukan Data array ke-1 : 22

Masukan Data array ke-2 : 523

Masukan Data array ke-3 : 123

Hasil

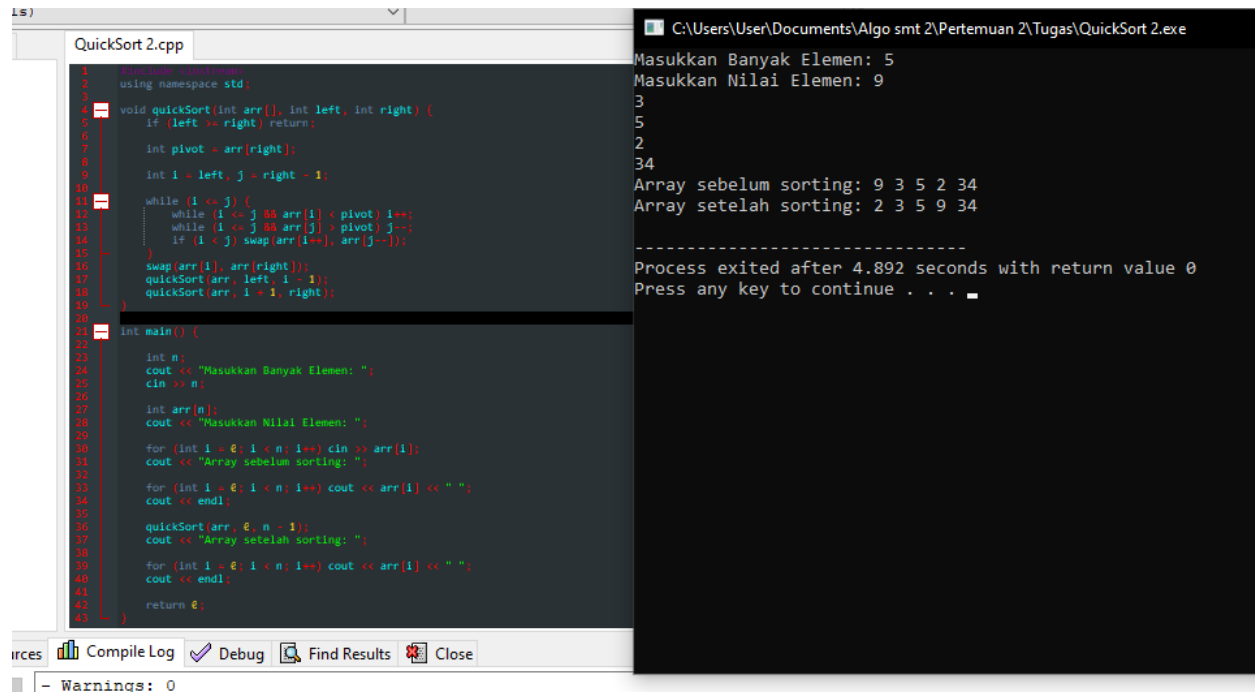
22 31 123 523

Process exited after 24.83 seconds with return value 0

Press any key to continue . . .

B.

QuickSort:



The screenshot shows a C++ IDE with two windows. The left window, titled 'QuickSort 2.cpp', contains the following code:

```
1 #include <iostream>
2 using namespace std;
3
4 void quickSort(int arr[], int left, int right) {
5     if (left >= right) return;
6
7     int pivot = arr[right];
8
9     int i = left, j = right - 1;
10
11     while (i < j) {
12         while (i < j && arr[i] < pivot) i++;
13         while (i < j && arr[j] > pivot) j--;
14         if (i < j) swap(arr[i++], arr[j--]);
15     }
16     swap(arr[i], arr[right]);
17     quickSort(arr, left, i - 1);
18     quickSort(arr, i + 1, right);
19 }
20
21 int main() {
22     int n;
23     cout << "Masukkan Banyak Elemen: ";
24     cin >> n;
25
26     int arr[n];
27     cout << "Masukkan Nilai Elemen: ";
28
29     for (int i = 0; i < n; i++) cin >> arr[i];
30     cout << "Array sebelum sorting: ";
31
32     for (int i = 0; i < n; i++) cout << arr[i] << " ";
33     cout << endl;
34
35     quickSort(arr, 0, n - 1);
36     cout << "Array setelah sorting: ";
37
38     for (int i = 0; i < n; i++) cout << arr[i] << " ";
39     cout << endl;
40
41     return 0;
42 }
```

The right window shows the execution output:

```
C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\QuickSort 2.exe
Masukkan Banyak Elemen: 5
Masukkan Nilai Elemen: 9
3
5
2
34
Array sebelum sorting: 9 3 5 2 34
Array setelah sorting: 2 3 5 9 34
-----
Process exited after 4.892 seconds with return value 0
Press any key to continue . . .
```

At the bottom of the IDE, there are buttons for 'Compile Log', 'Debug', 'Find Results', and 'Close'. Below these buttons, it says 'Warnings: 0'.

MergeSort:



The screenshot shows a C++ IDE with a single window titled 'MergeSort 2.cpp'. The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3
4 void merge(int arr[], int left, int mid, int right) {
5
6     int n1 = mid - left + 1;
7     int n2 = right - mid;
8     int L[n1], R[n2];
9
10    for (int i = 0; i < n1; i++) L[i] = arr[left + i];
11    .....
12    for (int j = 0; j < n2; j++) R[j] = arr[mid + 1 + j];
13
14    int i = 0, j = 0, k = left;
15
16    while (i < n1 && j < n2) {
17        if (L[i] <= R[j]) arr[k++] = L[i++];
18        else arr[k++] = R[j++];
19    }
20
21    while (i < n1) arr[k++] = L[i++];
22    while (j < n2) arr[k++] = R[j++];
23 }
24
25 void mergeSort(int arr[], int left, int right) {
26
27     if (left >= right) return;
28
29     int mid = left + (right - left) / 2;
30
31     mergeSort(arr, left, mid);
32
33     mergeSort(arr, mid + 1, right);
34
35     merge(arr, left, mid, right);
36 }
37
```

```

37
38 int main() {
39
40     int n;
41     cout << "Masukkan Banyak Elemen: ";
42     cin >> n;
43
44     int arr[n];
45     cout << "Masukkan Nilai Elemen: ";
46
47     for (int i = 0; i < n; i++) cin >> arr[i];
48
49     cout << "Array sebelum sorting: ";
50
51     for (int i = 0; i < n; i++) cout << arr[i] << " ";
52     cout << endl;
53
54     mergeSort(arr, 0, n - 1);
55     cout << "Array setelah sorting: ";
56
57     for (int i = 0; i < n; i++) cout << arr[i] << " ";
58     cout << endl;
59
60     return 0;
61 }

```

```

C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\MergeSort 2.exe
Masukkan Banyak Elemen: 5
Masukkan Nilai Elemen: 323
412
1123
123
532
Array sebelum sorting: 323 412 1123 123 532
Array setelah sorting: 123 323 412 532 1123

-----
Process exited after 12.76 seconds with return value 0
Press any key to continue . . .

```

Nomor 2:

QuickSort:

QuickSort Nomor 2.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  void swap(int arr[], int a, int b) {
5      int temp = arr[a];
6      arr[a] = arr[b];
7      arr[b] = temp;
8  }
9
10 void printArray(int arr[], int size) {
11     cout << "[";
12     for (int i = 0; i < size; i++) {
13         cout << arr[i] << " ";
14     }
15     cout << "]" << endl;
16 }
17
18 int partition(int arr[], int low, int high) {
19     int pivot = arr[high];
20     int i = low - 1;
21     for (int j = low; j < high; j++) {
22         if (arr[j] <= pivot) {
23             i++;
24             swap(arr, i, j);
25         }
26     }
27     swap(arr, i + 1, high);
28     printArray(arr, high + 1);
29     return i + 1;
30 }
31
```

```
31
32 void quicksort(int arr[], int low, int high) {
33     if (low < high) {
34         int pos = partition(arr, low, high);
35         quicksort(arr, low, pos - 1);
36         quicksort(arr, pos + 1, high);
37     }
38 }
39
40 int main() {
41     int n;
42     cout << "Banyak Data yang akan di sort sebanyak - ";
43     cin >> n;
44     int arr[n];
45     for (int i = 0; i < n; i++) {
46         cin >> arr[i];
47     }
48     cout << "\nQuick Sort \n" << endl;
49     printArray(arr, n);
50     quicksort(arr, 0, n - 1);
51     return 0;
52 }
```

C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\QuickSort Nomor 2.exe

Banyak Data yang akan di sort sebanyak - 6
4
8
2
3
9
1
Quick Sort
[4 8 2 3 9 1]
[1 8 2 3 9 4]
[1 2 3 4 9 8]
[1 2 3]
[1 2 3 4 8 9]

Process exited after 11.06 seconds with return value 0
Press any key to continue . . .

es

Compile Log Debug Find Results Close

- Warnings: 0

- Output Filename: C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\QuickSort Nomor 2.exe

MergeSort:

MergeSort Nomor 2.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  void showArray(int arr[], int start, int end) {
5      cout << "[";
6      for (int i = start; i <= end; i++) {
7          cout << arr[i];
8          if (i < end) cout << ", ";
9      }
10     cout << "]\n";
11 }
12
13 void merge(int arr[], int l, int m, int r) {
14     int n1 = m - l + 1;
15     int n2 = r - m;
16     int L[n1], R[n2];
17     for (int i = 0; i < n1; i++) L[i] = arr[l + i];
18     for (int j = 0; j < n2; j++) R[j] = arr[m + 1 + j];
19     cout << "Menggabungkan List: "; showArray(L, 0, n1 - 1);
20     cout << "dengan List: "; showArray(R, 0, n2 - 1);
21     int i = 0, j = 0, k = l;
22     while (i < n1 && j < n2) {
23         if (L[i] <= R[j]) arr[k++] = L[i++];
24         else arr[k++] = R[j++];
25     }
26     while (i < n1) arr[k++] = L[i++];
27     while (j < n2) arr[k++] = R[j++];
28 }
29
```

```
29 void mergeSort(int arr[], int l, int r) {
30     if (l < r) {
31         int m = l + (r - l) / 2;
32         cout << "Pecah List "; showArray(arr, l, m);
33         cout << "Pecah List "; showArray(arr, m + 1, r);
34         mergeSort(arr, l, m);
35         mergeSort(arr, m + 1, r);
36         merge(arr, l, m, r);
37         cout << "Menggabungkan List "; showArray(arr, l, r);
38     }
39 }
40
41 int main() {
42     int n;
43     cout << "Input jumlah data: ";
44     cin >> n;
45     int arr[n];
46     for (int i = 0; i < n; i++) {
47         cin >> arr[i];
48     }
49     mergeSort(arr, 0, n - 1);
50     cout << "Hasil akhir: ";
51     showArray(arr, 0, n - 1);
52     return 0;
53 }
54
```

C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\MergeSort Nomor 2.exe

Input jumlah data: 4
22
12
55
23
Pecah List [22, 12]
Pecah List [55, 23]
Pecah List [22]
Pecah List [12]
Menggabungkan List: [22]
dengan List: [12]
Menggabungkan List [12, 22]
Pecah List [55]
Pecah List [23]
Menggabungkan List: [55]
dengan List: [23]
Menggabungkan List [23, 55]
Menggabungkan List: [12, 22]
dengan List: [23, 55]
Menggabungkan List [12, 22, 23, 55]
Hasil akhir: [12, 22, 23, 55]

Process exited after 12.76 seconds with return value 0

Compile Log | Debug | Find Results | Close

- Warnings: 0
- Output Filename: C:\Users\User\Documents\Algo smt 2\Pertemuan 2\Tugas\MergeSort Nomor 2.exe
- Output Size: 1,83401775360107 MiB