



Praktikum Algoritma dan Pemrograman

Modul Praktikum PAP-04

Laboratorium Dasar FIK - UDINUS

Modul PAP-04

Deklarasikan fungsi-fungsi di bawah ini ke dalam **pustaka.h**, serta buat realisasi fungsi tersebut pada file **pustaka.c**, kemudian buat uji cobalah semua fungsi dengan membuat program pemanggil pada file **main.c**.

Fungsi	Deskripsi
void compressArray(int data[])	<p>Fungsi ini akan menghilangkan semua index array yang bernilai 0 dan berada di tengah array yang bernilai lebih dari 0. Fungsi ini akan menggeser semua array yang berada di sebelah kanan index yang bernilai 0 sehingga tidak ada lagi nilai 0 di tengah array.</p> <pre>int arr1[] = { 3, 5, 38, 0, 44, 47}; int arr2[] = { 3, 0, 44, 38, 0, 5, 47 }; int arr3[] = { 2, 15, 0, 0, 0, 26, 27, 36 }; int arr4[] = { 15, 36, 27, 0, 0, 2, 26 }; - compressArray (arr1) -> arr1[] = { 3, 5, 38, 44, 47} - compressArray (arr2) -> arr2[] = { 3, 44, 38, 5, 47 } - compressArray (arr3) -> arr3[] = { 2, 15, 26, 27, 36 } - compressArray (arr4) -> arr4[] = { 15, 36, 27, 2, 26 }</pre>
void selectionSort1(int data[])	<p>Fungsi ini akan mengurutkan larik integer yang diberikan pada parameter data dan mencetak ke layar setiap iterasi pengurutan yang dilakukan.</p> <p>Pada fungsi ini algoritma pengurutan yang digunakan adalah selection sort tanpa menggunakan pengecekan urut. Sehingga meskipun di tengah iterasi data telah terurut, fungsi ini akan tetap melanjutkan proses hingga semua iterasi selesai dijalankan.</p> <pre>int arr1[] = { 3, 44, 38, 5, 47 }; int arr2[] = { 15, 36, 27, 2, 26 }; - selectionSort1(arr1) -> arr1 = { 3, 5, 38, 44, 47 }</pre> <div><pre>3 44 38 5 47 3 5 38 44 47 3 5 38 44 47 3 5 38 44 47 3 5 38 44 47</pre></div> <pre>- selectionSort1(arr2) -> arr2 = { 2, 15, 26, 27, 36 }</pre>



Praktikum Algoritma dan Pemrograman

Modul Praktikum PAP-04

Laboratorium Dasar FIK - UDINUS

Fungsi	Deskripsi
	<pre>2 36 27 15 26 2 15 27 36 26 2 15 26 36 27 2 15 26 27 36 2 15 26 27 36</pre>
<code>void selectionSort2(int data[])</code>	<p>Fungsi ini akan mengurutkan larik integer yang diberikan pada parameter data dan mencetak ke layar setiap iterasi pengurutan yang dilakukan.</p> <p>Pada fungsi ini algoritma pengurutan yang digunakan adalah selection sort dengan menggunakan pengecekanurut. Sehingga jika di tengah iterasi data telah terurut, maka fungsi akan berhenti tanpa menyelesaikan iterasi lagi.</p> <pre>int arr1[] = { 3, 44, 38, 5, 47 }; int arr2[] = { 15, 36, 27, 2, 26 }; - selectionSort2(arr1) -> arr1 = { 3, 5, 38, 44, 47 } 3 44 38 5 47 3 5 38 44 47 - selectionSort2(arr2) -> arr2 = { 2, 15, 26, 27, 36 } 2 36 27 15 26 2 15 27 36 26 2 15 26 36 27 2 15 26 27 36</pre>
<code>int getMedian(int data[])</code>	<p>Fungsi ini akan mengembalikan nilai median dari sebuah larik yang diberikan. Jika larik pada parameter belum terurut, maka fungsi ini akan melakukan pengurutan terlebih dahulu sebelum melakukan proses pencarian median.</p> <pre>int arr1[] = { 3, 44, 38, 5, 47 }; int arr2[] = { 15, 36, 27, 2, 26 }; - getMedian(arr1) -> 38 - getMedian(arr2) -> 26</pre>