

SORTING

INFORMATIKA

ANGGOTA KELOMPOK

- **JOBDESK**

Lista Apiq Mudrikah

Mencari materi tentang jenis-jenis sorting.

Salwa Najwa

Memasukkan materi implementasi sorting ke dalam ppt.

Aisyah Hamida

Mencari materi tentang definisi sorting.

Kuni Nur Diana

Mencari template ppt & mencari materi implementasi sorting.

Taqiya Mir'atul Waro'

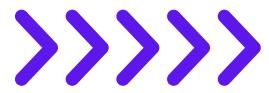
Memasukkan materi implementasi sorting ke dalam ppt.

Gifta Qurrota

Memasukkan materi tentang definisi sorti ke dalam ppt.

Syifa Arrohmah Anwar

Memasukkan materi jenis sorting ke dalam ppt.



DEFINISI

Pengurutan adalah operasi penting dalam pengolahan data dan memiliki aplikasi luas di berbagai disiplin ilmu, termasuk ilmu komputer, matematika, ilmu data, dan sebagainya.

Pengurutan (sorting) adalah proses mengatur atau menyusun elemen-elemen dari sebuah himpunan data dalam suatu urutan tertentu, baik itu berdasarkan kriteria numerik (misalnya, dari terkecil ke terbesar), urutan alfabet, atau berdasarkan kriteria lainnya.

INOVASI BISNIS

TEKNOLOGI
INFORMASI



JENIS-JENIS SORTING

TEKNOLOGI
INFORMASI

BUBBLE SORT

Bubble sort adalah membandingkan dan menukar pasangan elemen secara berurutan sampai data terurut

SELECTION SORT

Selection sort memilih elemen terkecil dari sisa data dan menukar dengan elemen pertama. Kemudian, elemen terkecil dipilih dari sisa data dan seterusnya

INSERTION SORT

Biasanya tipe jenis ini memeriksa setiap elemen dan memasukkannya ke posisi yang sesuai dalam sublist yang terurut sebelumnya

MERGE SORT

Merge Sort membagi data menjadi bagian-bagian kecil, mengurutkan masing-masing bagian, dan kemudian menggabungkannya kembali dalam urutan yang benar

QUICK SORT

Quick Sort memilih elemen pivot, membagi data menjadi dua bagian berdasarkan pivot, dan mengurutkan masing-masing bagian secara terpisah.

HEAP SORT

Heap Sort menggunakan struktur data heap untuk mengurutkan data. Ini membangun heap maksimum dari data, kemudian secara berulang mengambil elemen terbesar dan memperbaiki struktur heap



JENIS-JENIS SORTING



RADIX SORT

Radix Sort mengurutkan data berdasarkan digit individu mulai dari digit terendah hingga digit tertinggi.

COUNTING SORT

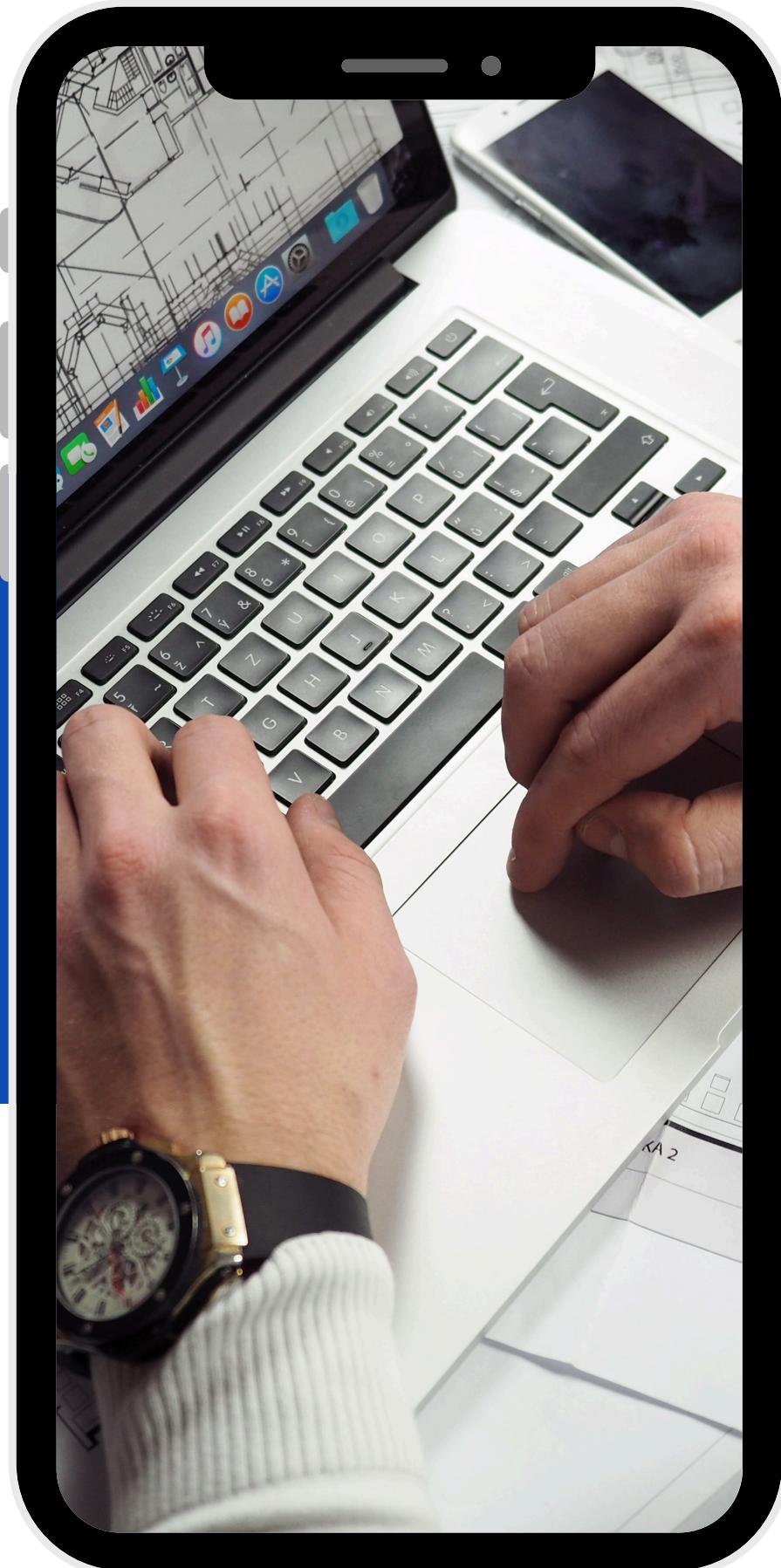
Counting Sort bekerja dengan menghitung frekuensi masing-masing elemen dan kemudian membangun hasil berdasarkan frekuensi tersebut

BUCKET SORT

Bucket Sort membagi data ke dalam sejumlah “ember” atau “bucket” dan mengurutkan masing-masing bucket secara terpisah, kemudian menggabungkannya

SHELL SORT

biasanya bekerja dengan cara membandingkan suatu data lain yang memiliki jarak tertentu sehingga membuat sebuah sub-list



IMPLEMENTASI SORTING



BUBBLE SHORT

IMPLEMENTASI:

```
echo "//bubble sortn";
$data=array(6,5,3,1,8,7,2,4);
function bubble_sort($data){
    $n=count($data);
    for ($i = 0;$i<$n;$i++){ for ($j = $n-1;$j>$i;$j--){
        if ($data[$j] < $data[$j-1]){
            $dummy=$data[$j];
            $data[$j]=$data[$j-1];
            $data[$j-1]=$dummy;
        }
    }
    return $data;
}
print_r(bubble_sort($data));
```

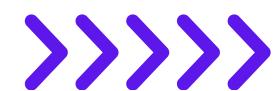
Algoritma pengurutan paling tua dengan metode pengurutan paling sederhana. Pengurutan yang dilakukan dengan membandingkan masing-masing item dalam suatu list secara berpasangan, menukar item jika diperlukan, dan mengulaginya sampai akhir list secara berurutan, sehingga tidak ada lagi item yang dapat ditukar.

IMPLEMENTASI:

```
echo "//insertion sortn";
$data=array(6,5,3,1,8,7,2,4);
function insertion_sort($data){
    $n=count($data);
    for ($i = 1;$i<$n;$i++){ for ($k = $i; $k>0; $k--) {
        if($data[$k]<$data[$k-1]){
            $dummy=$data[$k];
            $data[$k]=$data[$k-1];
            $data[$k-1]=$dummy;
        }
    }
}
return $data;
}
print_r(insertion_sort($data));
```

INSERTION SHORT

Algoritma insertion sort pada dasarnya memilah data yang akan diurutkan menjadi dua bagian, yang belum diurutkan dan yang sudah diurutkan. Elemen pertama diambil dari bagian array yang belum diurutkan dan kemudian diletakkan sesuai posisinya pada bagian lain dari array yang telah diurutkan. Langkah ini dilakukan secara berulang hingga tidak ada lagi elemen yang tersisa pada bagian array yang belum diurutkan.

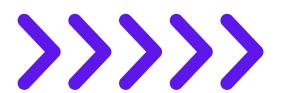


SELECTION SHORT

Ide utama dari algoritma selection sort adalah memilih elemen dengan nilai paling rendah dan menukar elemen yang terpilih dengan elemen ke-i. Nilai dari i dimulai dari 1 ke n, dimana n adalah jumlah total elemen dikurangi 1.

IMPLEMENTASI:

```
echo "//selection sortn";
$data=array(6,5,3,1,8,7,2,4);
function selection_sort($data){
    $n=count($data);
    for ($i = 0;$i<$n;$i++){
        $k = $i;
        for ($j = $i+1;$j<$n;$j++){
            if ($data[$j] < $data[$k]) $k = $j;
        }
        $dummy=$data[$i];
        $data[$i]=$data[$k];
        $data[$k]=$dummy;
    }
    return $data; Lorem ipsum dolor sit amet,
}                               consectetur adipiscing
print_r(selection_sort($data));
```



SHELL SORT

Merupakan algoritma yang satu jenis dengan insertion sort, dimana pada setiap nilai i dalam n/i item diurutkan. Pada setiap pergantian nilai, i dikurangi sampai 1 sebagai nilai terakhir.

IMPLEMENTASI:

```
echo "//shell sortn";
$data=array(6,5,3,1,8,7,2,4);
function shell_sort($data){
    $n=count($data);
    $k=0;
    $gap[0]=(int) ($n / 2);
    while($gap[$k]>1){
        $k++;
        $gap[$k]=(int)($gap[$k-1]/2);
    }
    for($i=0;$i<=$k;$i++){
        $step=$gap[$i];
        for($j=$step;$j<$n;$j++){
            $temp=$data[$j];
            $p=$j-$step;
            while($p>=0 && $temp<$data[$p]){
                $data[$p+$step]=$data[$p];
                $p=$p-$step;
            }
            $data[$p+$step]=$temp;
        }
    }
    return $data;
}
print_r(shell_sort($data));
```



MERGE SORT

Algoritma dirumuskan dalam 3 langkah berpola divide-and-conquer. Berikut menjelaskan langkah kerja dari Merge sort.

- Divide : Memilah elemen – elemen dari rangkaian data menjadi dua bagian.
- Conquer : Conquer setiap bagian dengan memanggil prosedur merge sort secara rekursif
- Kombinasi : Mengkombinasikan dua bagian tersebut secara rekursif untuk mendapatkan rangkaian data berurutan

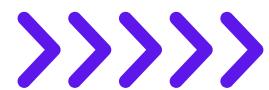
Proses rekursi berhenti jika mencapai elemen dasar. Hal ini terjadi bilamana bagian yang akan diurutkan menyisakan tepat satu elemen. Sisa pengurutan satu elemen tersebut menandakan bahwa bagian tersebut telah terurut sesuai rangkaian.

IMPLEMENTASI:

```
echo "//merge sortn";
$data=array(6,5,3,1,8,7,2,4);
function merge_sort ($data){
    if (count($data) <= 1 ) return $data;
    $left =
        merge_sort(array_splice($data,floor(count($data) / 2)));
    $right = merge_sort($data);
    $result = array();
    while (count($left) > 0 && count($right) > 0){
        if ($left[0] <= $right[0])
            array_push($result, array_shift($left));
        else
            array_push($result, array_shift($right));
    }
    while (count($left) > 0)
        array_push($result, array_shift($left));
    while (count($right) > 0)
        array_push($result, array_shift($right));
    return $result;
}
print_r(merge_sort($data));
```

IMPLEMENTASI:

```
echo "//quick sortn";
$data=array(6,5,3,1,8,7,2,4);
function quick_sort($data) {
    if(!count($data)) return $data;
    $pivot= $data[0];
    $low = $high = array();
    $n = count($data);
    for($i=1; $i < $n; $i++) {
        if($data[$i] <= $pivot) {
            $low [] = $data[$i];
        } else {
            $high[] = $data[$i];
        }
    }
    return array_merge(quick_sort($low), array($pivot),
    quick_sort($high));
}
print_r(quick_sort($data));
```



QUICK SORT

Algoritma ini berdasarkan pada pola divide-and-conquer.

- Divide

Memilah rangkaian data menjadi dua sub-rangkaian $A[p \dots q-1]$ dan $A[q+1 \dots r]$ dimana setiap elemen $A[p \dots q-1]$ adalah kurang dari atau sama dengan $A[q]$ dan setiap elemen pada $A[q+1 \dots r]$ adalah lebih besar atau sama dengan elemen pada $A[q]$. $A[q]$ disebut sebagai elemen pivot. Perhitungan pada elemen q merupakan salah satu bagian dari prosedur pemisahan.

- Conquer

Mengurutkan elemen pada sub-rangkaian secara rekursif. Pada algoritma quicksort, langkah "kombinasi" tidak dilakukan karena telah terjadi pengurutan elemen-elemen pada sub-array.

IMPLEMENTASI:

```
echo "//heap sortn";
$data=array(6,5,3,1,8,7,2,4);
function build_heap(&$data, $i, $t){
    $tmp_var = $data[$i];
    $j = $i * 2 + 1;
    while ($j <= $t) {
        if($j < $t)
            if($data[$j] < $data[$j + 1]) {
                $j = $j + 1;
            }
        if($tmp_var < $data[$j]) {
            $data[$i] = $data[$j];
            $i = $j;
            $j = 2 * $i + 1;
        } else {
            $j = $t + 1;
        }
    }
    $data[$i] = $tmp_var;
}

function heap_sort(&$data) {
    $init = (int)floor((count($data) - 1) / 2);
    for($i=$init; $i >= 0; $i--){
        $count = count($data) - 1;
        build_heap($data, $i, $count);
    }
    for ($i = (count($data) - 1); $i >= 1; $i--) {
        $tmp_var = $data[0];
        $data[0] = $data[$i];
        $data[$i] = $tmp_var;
        build_heap($data, 0, $i - 1);
    }
}
heap_sort($data);
print_r($data);
```



HEAP SORT

Heap sort adalah sorting yang menggunakan struktur data heap, dengan nilai parent selalu lebih besar dari pada nilai childnya.

Algoritma :

- Buat suatu heap.
- Ambil isi dari root masukkan kedalam sebuah array.
- Hapus element root dengan mempertahankan properti heap.
- Ulangi sampai tree menjadi kosong



TERIMA KASIH

INFORMATIKA

TEKNOLOGI
INFORMASI