

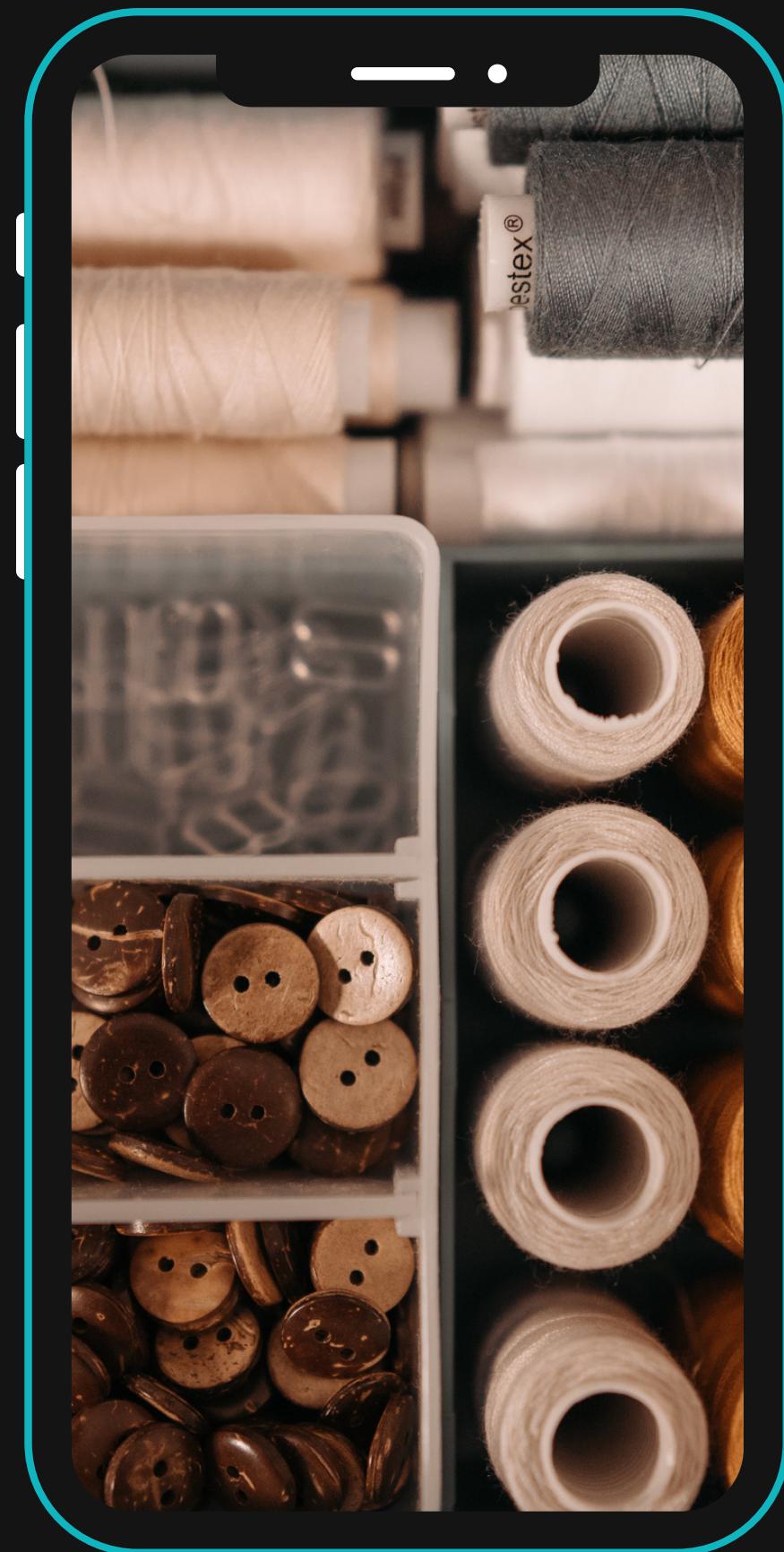
Pertemuan 1

PENGURUTAN (SORTING)

Institut Bisnis dan Informatika Kesatuan Bogor



Apa itu Sorting?



Sebuah konsep atau proses yang mengurutkan elemen-elemen array yang acak mulai dari yang terkecil ke terbesar atau sebaliknya. Sorting dari yang terkecil keterbesar disebut Ascending, sedangkan dari yang terbesar ke terkecil disebut Descending.

Tujuan Sorting

untuk mengatur ulang array atau elemen daftar tertentu sesuai dengan operator perbandingan pada elemen.

ATAU

untuk mengatur elemen data agar lebih mudah ditemukan pada saat proses pencarian.

Untuk mengurutkan dataset, algoritma pengurutan terbagi menjadi 2.
Untuk set input kecil dapat menggunakan **Bubble-Sort, insert-Sort dan Selection-Sort**.
Sedangkan untuk data set besar dapat menggunakan **Merge-Sort, Quick-Sort dan Heap-Sort**.



Algoritma Bubble Sort

algoritma penyortiran paling sederhana yang bekerja dengan teknik pengurutan data dengan cara menukar dua data yang bersebelahan jika urutan dari data tersebut salah.

Contoh:

Pass Pertama:

(5 1 4 2 8) \rightarrow (1 5 4 2 8), Di sini, algoritma membandingkan dua elemen pertama, dan swap sejak $5 > 1$.

(1 5 4 2 8) \rightarrow (1 4 5 2 8), Swap sejak $5 > 4$

(1 4 5 2 8) \rightarrow (1 4 2 5 8), Swap sejak $5 > 2$

(1 4 2 5 8) \rightarrow (1 4 2 5 8), Sekarang, karena elemen-elemen ini sudah dalam rangka ($8 > 5$), algoritma tidak menukarnya.

Pass Kedua:

(1 4 2 5 8) \rightarrow (1 4 2 5 8)

(1 4 2 5 8) \rightarrow (1 2 4 5 8), Swap sejak $4 > 2$

(1 2 4 5 8) \rightarrow (1 2 4 5 8)

(1 2 4 5 8) \rightarrow (1 2 4 5 8)

Setelah, array diurutkan, algoritma membutuhkan satu parameter dimana seluruh elemen lulus tanpa swap untuk mengetahui itu diurutkan untuk mengetahui apakah pengurutan sudah selesai dilakukan.

Pass Ketiga:

(1 2 4 5 8) \Rightarrow (1 2 4 5 8)

(1 2 4 5 8) \Rightarrow (1 2 4 5 8)

(1 2 4 5 8) \Rightarrow (1 2 4 5 8)

(1 2 4 5 8) \Rightarrow (1 2 4 5 8)

Algoritma Insertion Sort

algoritma insertion sort dimulai dari memilah data yang akanurutkan menjadi 2 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 tak ada lagi elemen tersisa pada bagian array yang belum diurutkan.

Algoritma Insertion Sort

Hal ini berkaitan dengan prinsip kerja atau metode insertion sort dimana terdapat sebuah metode yang mengurutkan bilangan-bilangan yang telah terbaca, dan berikutnya secara berulang akan menyisipkan bilangan-bilangan dalam array yang belum terbaca ke sisi kiri array yang telah terurut.

Berikut adalah illustrasi dari *insertion sort* yang digunakan untuk mengurutkan bilangan 1-10 yang diacak secara *ascending*.

1. Langkah pertama ambil bilangan yang paling kiri. Bilangan tersebut dikatakan urut terhadap dirinya sendiri karena bilangan yang di bandingkan baru 1.

3	10	4	6	8	9	7	2	1	5
---	----	---	---	---	---	---	---	---	---

Gambar 1.1 Langkah 1 *insertion sort*

2. Selanjutnya, periksa bilangan ke 2 (10) apakah lebih kecil dari bilangan yang ke 1(3). Apabila lebih kecil maka ditukar. Tapi kali ini bilangan ke 1 lebih kecil dari bilangan ke 2, maka tidak ditukar.

3	10	4	6	8	9	7	2	1	5
3	10	4	6	8	9	7	2	1	5

3. Pada kotak warna abu - abu sudah dalam keadaan terurut. Kemudian membandingkan lagi pada bilangan selanjutnya yaitu bilangan ke 3 (4). Bandingkan dengan bilangan yang ada di sebelah kirinya. Pada kasus ini bilangan ke 2 bergeser dan digantikan bilangan ke-3.

3	10	4	6	8	9	7	2	1	5
---	----	---	---	---	---	---	---	---	---

3	10	4	6	8	9	7	2	1	5
---	----	---	---	---	---	---	---	---	---

4. Lakukan langkah seperti di atas pada bilangan selanjutnya. 4 bilangan pertama sudah dalam keadaan terurut relatif. Ulangi proses tersebut sampai bilangan terakhir disisipkan.

3	4	10	6	8	9	7	2	1	5
---	---	----	----------	---	---	---	---	---	---

3	4	6	10	8	9	7	2	1	5
---	---	---	----	---	---	---	---	---	---

3	4	6	8	10	9	7	2	1	5
---	---	---	---	----	----------	---	---	---	---

3	4	6	8	9	10	7	2	1	5
---	---	---	---	---	----	----------	---	---	---

3	4	6	7	8	9	10	2	1	5
---	---	---	---	---	---	----	----------	---	---

2	3	4	6	7	8	9	10	1	5
----------	---	---	---	---	---	---	----	---	---

1	2	3	4	6	7	8	9	10	5
---	---	---	---	---	---	---	---	----	----------

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Algoritma Selection Sort

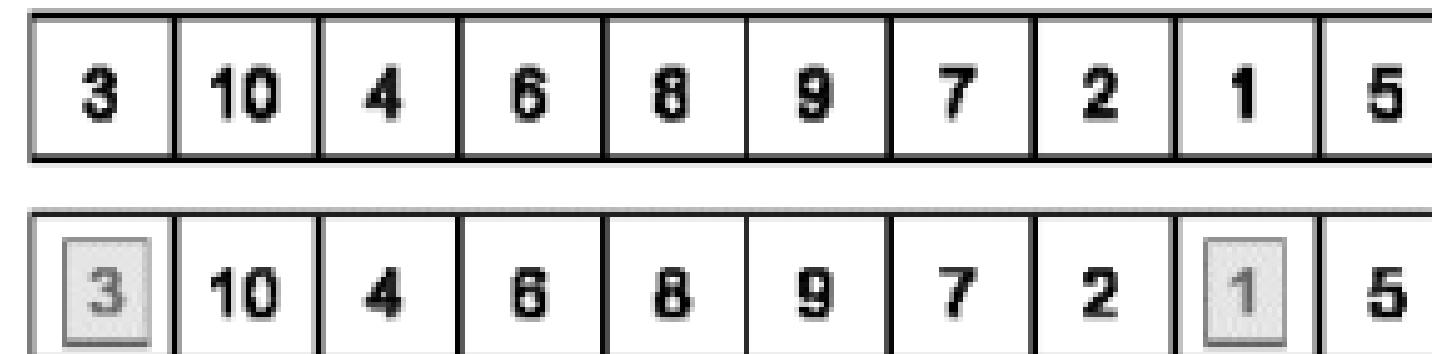
Selection sort merupakan sebuah teknik pengurutan dengan cara mencari nilai tertinggi / terendah di dalam array kemudian menempatkan nilai tersebut di tempat semestinya.

Algoritma Selection Sort

Dapat dikatakan jika algoritma selection sort adalah teknik 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.

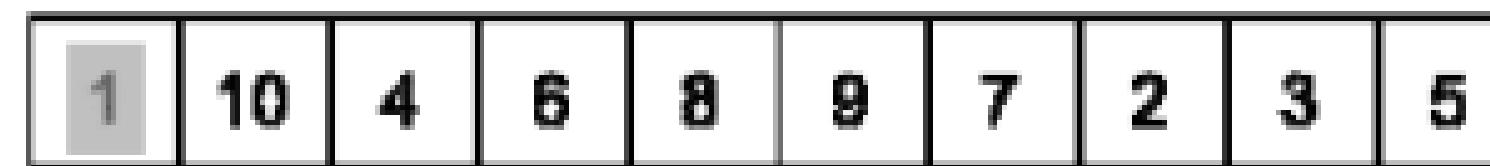
Berikut adalah illustrasi dari *selection sort* yang digunakan untuk mengurutkan bilangan 1-10 yang diacak secara *ascending*.

1. Langkah pertama, periksalah seluruh array dan cari array yang mempunyai nilai terkecil → index 8 (1). Setelah ketemu tukar dengan array yang berada di pojok kiri (3).



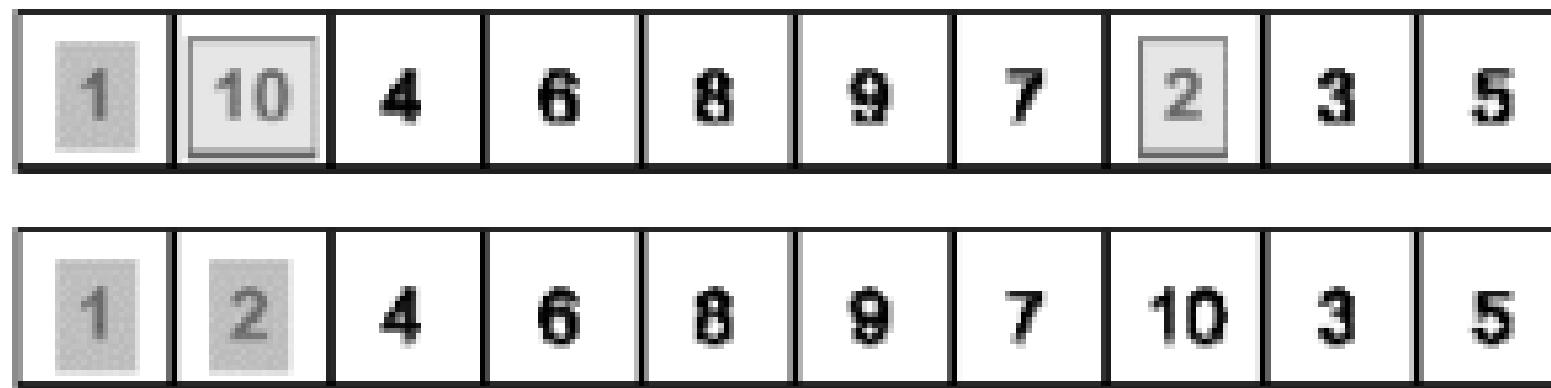
Gambar 1.5 Langkah 1 *selection sort*

2. Selanjutnya, setelah di tukar terdapat bagian yang berwarna abu-abu yang merupakan index yang telah terurutkan.



Gambar 1.6 Langkah 2 *selection sort*

3. Lalu, cari bilangan terkecil selanjutnya (selain di kotak abu-abu) yaitu bilangan 2 dan tukar dengan sebelah array yang telah terurutkan.



Gambar 1.7 Langkah 3 *selection sort*

4. Terakhir, Dua array sudah terurutkan. Kemudian ulangi langkah di atas dan lakukan langkah yang sama yaitu pilih terkecil dan tukar dengan sebelah array yang sudah terurutkan.

1	2	4	8	8	9	7	10	3	5
1	2	3	6	8	9	7	10	4	5
1	2	3	6	8	9	7	10	4	5
1	2	3	4	8	9	7	10	6	5
1	2	3	4	8	9	7	10	6	5
1	2	3	4	5	9	7	10	6	8
1	2	3	4	5	9	7	10	6	8
1	2	3	4	5	6	7	10	9	8
1	2	3	4	5	6	7	10	9	8
1	2	3	4	5	6	7	10	9	8
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

Ada
pertanyaan?

