GuruVirtual.ID

Algoritma Pengurutan (*Sorting*) dan Implementasinya di Python *(bubble, insertion, selection)*

Hartono, S.Pd., M.T.I

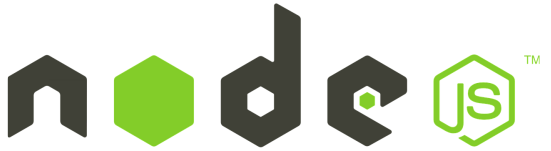
Universitas Muhammadiyah Kotabumi

Sorting (Pengurutan)

- Pengurutan (sorting) adalah proses mengatur elemen-elemen dalam suatu koleksi data secara berurutan berdasarkan kriteria tertentu, seperti nilai numerik atau urutan abjad.
- Konsep algoritma pengurutan merupakan dasar dalam ilmu komputer dan pemrograman, karena pengurutan sangat umum digunakan dalam berbagai aplikasi untuk mengatur data agar lebih mudah dicari, diakses, dan dianalisis.

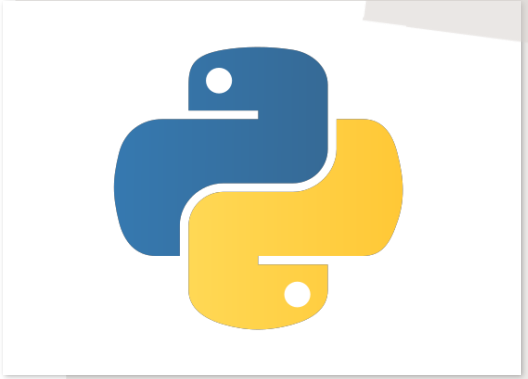
Konsep Algoritma Sorting

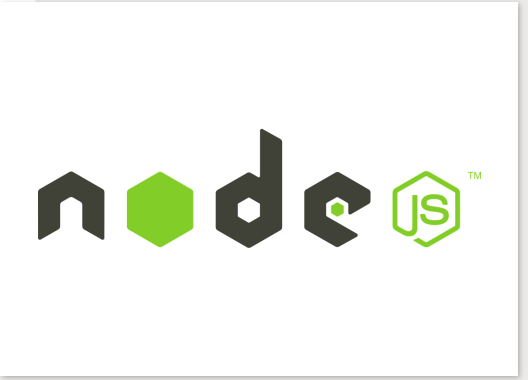
- **Komparasi dan Penukaran (Comparison and Exchange):** Algoritma pengurutan membandingkan elemen-elemen data untuk menentukan urutannya. Jika ditemukan elemen yang tidak berada pada posisi yang benar, elemen tersebut akan ditukar dengan elemen lain agar urutannya sesuai.
- **Stabilitas:** Pengurutan stabil mempertahankan urutan relatif dari elemen-elemen yang memiliki nilai yang sama. Misalnya, jika ada dua data dengan nilai yang sama, yang pertama muncul dalam urutan asli akan tetap berada di atas dalam urutan yang dihasilkan.
- **Efisien:** Algoritma pengurutan perlu efisien dalam hal waktu dan ruang yang dibutuhkan. Dalam banyak kasus, algoritma yang memiliki kompleksitas waktu lebih rendah lebih diinginkan karena dapat mengurangi waktu pemrosesan data.
- **Adaptif:** Algoritma pengurutan adaptif dapat memanfaatkan informasi tentang data yang hampir terurut untuk mengurangi komparasi yang diperlukan.

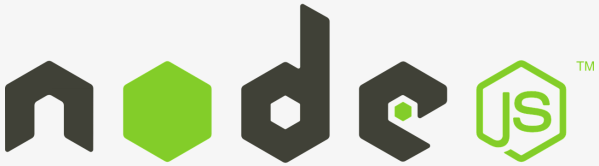


Keperluan Algoritma Pengurutan

1. **Pencarian Efisien:** Data yang terurut memungkinkan pencarian efisien. Misalnya, dengan menggunakan algoritma pencarian biner, pencarian dapat dilakukan dengan cepat karena data terurut.
2. **Pemrosesan Data:** Dalam aplikasi yang memerlukan pemrosesan data berulang kali, data yang terurut dapat menghemat waktu pemrosesan.
3. **Presentasi Data:** Data yang terurut lebih mudah dipresentasikan kepada pengguna dalam bentuk yang terstruktur dan mudah dibaca.
4. **Analisis Statistik:** Data yang terurut memudahkan analisis statistik, seperti menghitung median, kuartil, dan nilai ekstrem.





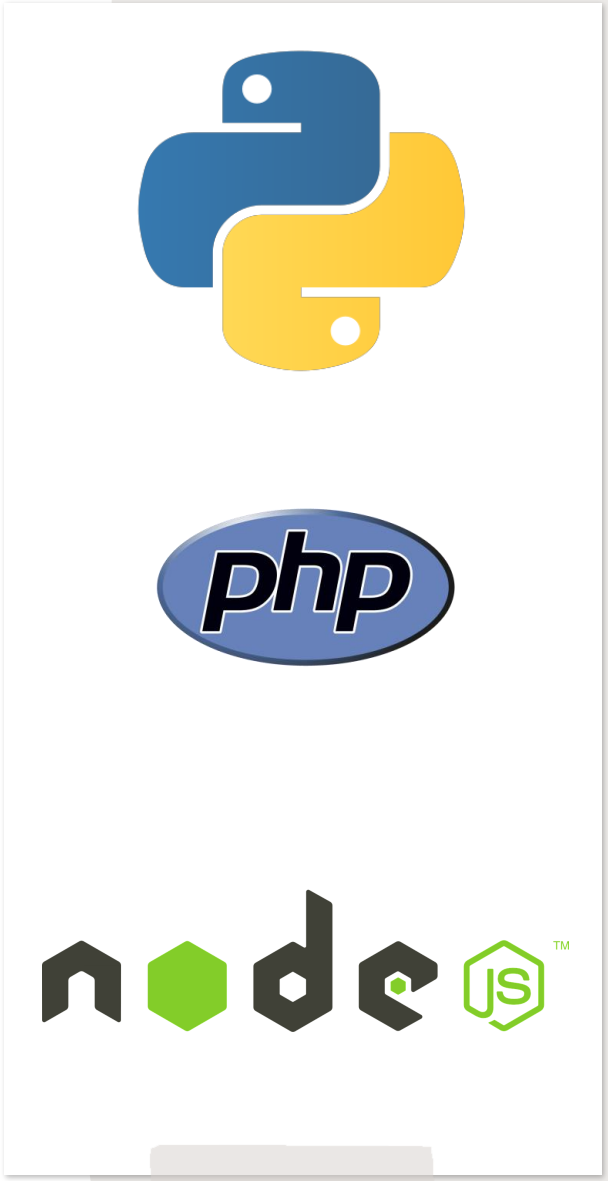


Keperluan Algoritma Pengurutan

5. **Penghapusan Duplikat:** Pengurutan memudahkan identifikasi dan penghapusan elemen duplikat dalam data.
6. **Penggabungan Data:** Dalam proses penggabungan dua atau lebih set data, data yang terurut dapat diintegrasikan dengan lebih efisien.
7. **Optimisasi Algoritma Lain:** Beberapa algoritma lain, seperti algoritma untuk mencari elemen terbanyak atau menghitung inversi, dapat dioptimalkan dengan menggunakan data yang terurut.

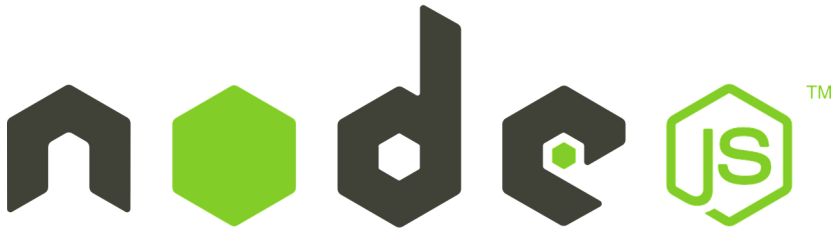
Jenis-Jenis Algoritma Pengurutan

1. **Bubble Sort:** Algoritma sederhana di mana elemen-elemen dibandingkan secara berpasangan dan ditukar jika urutannya salah. Ini berlanjut hingga tidak ada lagi pertukaran yang dibutuhkan. Meskipun mudah dipahami, Bubble Sort memiliki kinerja yang buruk untuk jumlah data yang besar.
2. **Insertion Sort:** Algoritma ini menganggap elemen yang belum diurutkan sebagai satu set kartu yang diambil satu per satu dan dimasukkan ke tangan yang telah diurutkan secara benar. Ini cocok untuk data yang sudah dalam keadaan hampir terurut.
3. **Selection Sort:** Algoritma ini secara berulang memilih elemen terkecil dari sisa data dan menukar elemen tersebut dengan elemen pertama yang belum terurut. Ini diulang hingga seluruh data terurut. Seperti Bubble Sort, Selection Sort tidak efisien untuk jumlah data besar.



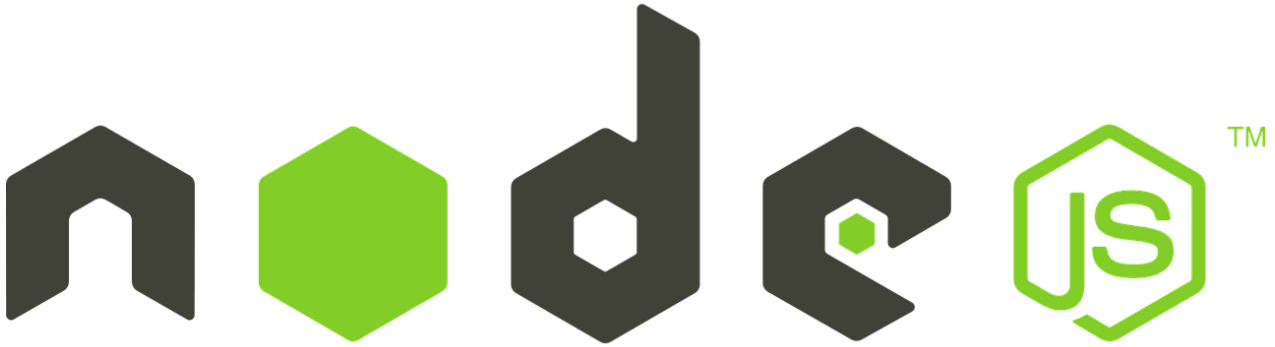
Bubble Sort

- Bubble Sort adalah salah satu algoritma pengurutan yang sederhana, tetapi biasanya tidak efisien untuk digunakan pada data dengan jumlah yang besar.
- Prinsip dasar Bubble Sort adalah dengan secara berulang membandingkan pasangan elemen bersebelahan dalam koleksi data dan menukar mereka jika urutan mereka salah.
- Proses ini terus diulang hingga tidak ada lagi pertukaran yang diperlukan, menunjukkan bahwa data sudah terurut.



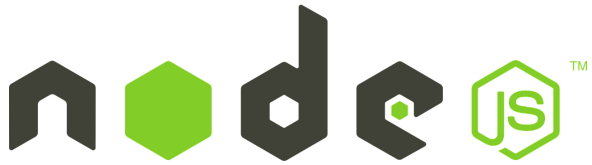
Langkah-Langkah Bubble Sort

1. Bandingkan dua elemen bersebelahan dalam koleksi data.
2. Jika urutan kedua elemen tersebut salah (elemen pertama lebih besar dari elemen kedua), tukar elemen tersebut.
3. Pindah ke pasangan elemen berikutnya dan ulangi langkah 1-2.
4. Ulangi langkah 1-3 untuk seluruh koleksi data, hingga tidak ada lagi pertukaran yang dibutuhkan.

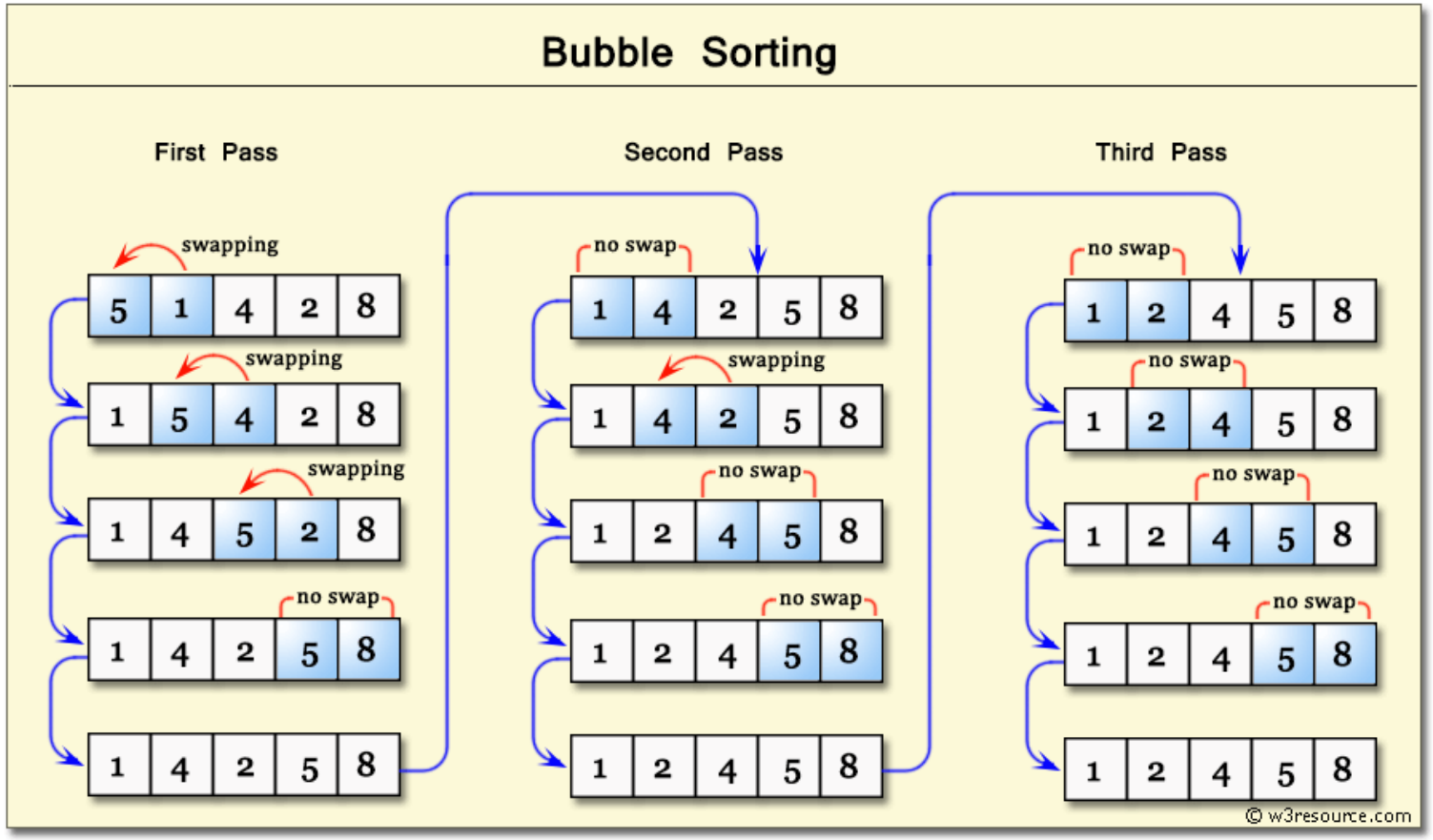


Pseudo-Code

```
prosedur BubbleSort(A : daftar item yang dapat diurutkan)
   n = panjang(A)
   lakukan
      ditukar = false
      untuk i = 1 hingga n-1
         jika A[i-1] > A[i]
            tukar(A[i-1], A[i])
            ditukar = true
         akhir jika
      end untuk
      n = n - 1
   selama ditukar
akhir prosedur
```



Pseudo-Code



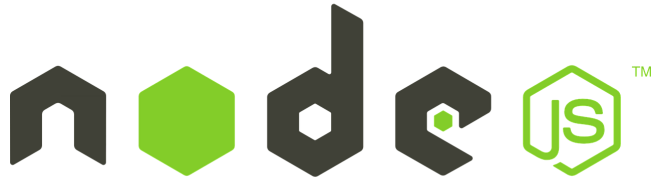
Sumber: https://www.w3resource.com/w3r_images/bubble-short.png

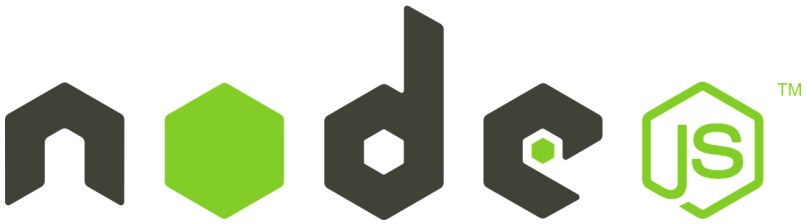
Kelemahan Bubble Sort

- Kelemahan utama dari Bubble Sort adalah kompleksitas waktu yang buruk dalam kasus-kasus terburuk dan rata-rata.
- Waktu eksekusi Bubble Sort adalah O(n^2), di mana n adalah jumlah elemen dalam koleksi data.
- Algoritma ini seringkali digunakan hanya untuk tujuan pendidikan atau dalam kasus-kasus di mana jumlah data sangat kecil dan performa bukanlah faktor kritis.

Insertion Sort

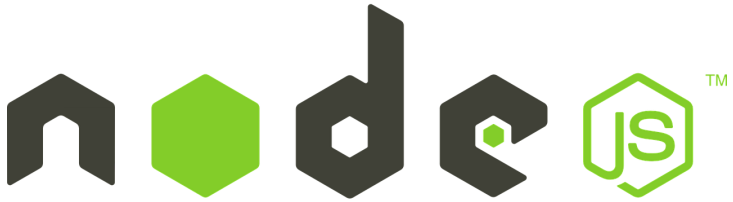
- Insertion Sort adalah salah satu algoritma pengurutan sederhana yang bekerja dengan cara membandingkan dan menyisipkan elemen-elemen data dalam urutan yang benar satu per satu.
- Prinsip dasar Insertion Sort adalah bahwa algoritma ini mempertimbangkan setiap elemen dalam data dan memasukkannya ke dalam posisi yang tepat dalam subkoleksi yang telah diurutkan sebelumnya.





Cara Kerja Insertion Sort

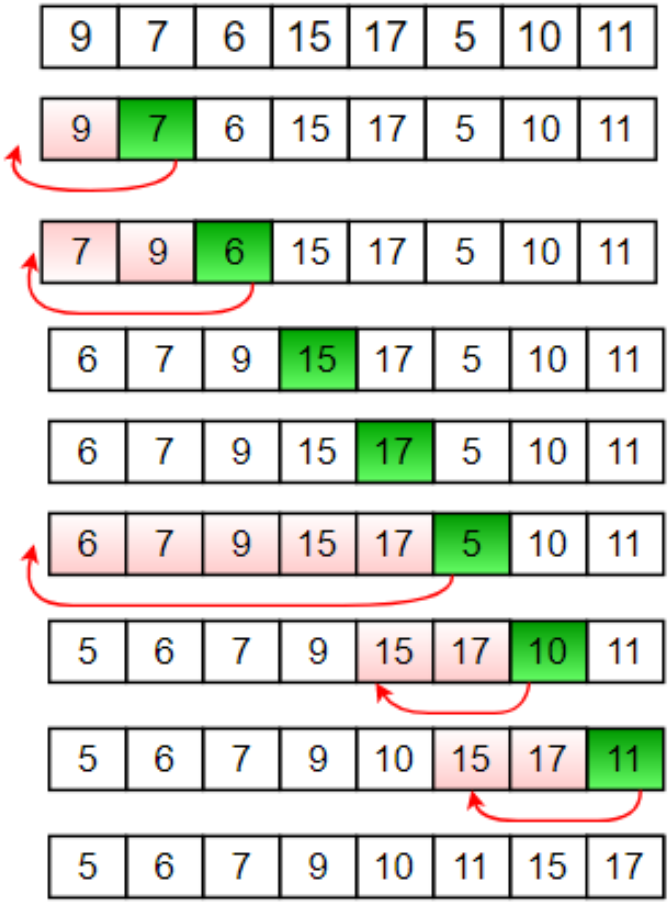
1. Algoritma memulai dengan menganggap bahwa elemen pertama dalam data sudah dalam keadaan terurut.
2. Algoritma kemudian memilih elemen kedua dan membandingkannya dengan elemen pertama. Jika elemen kedua lebih kecil, algoritma menukar posisi keduanya.
3. Algoritma kemudian memasukkan elemen ketiga ke dalam subkoleksi yang terurut sebelumnya. Algoritma membandingkan elemen ketiga dengan elemen-elemen dalam subkoleksi dan memindahkan elemen-elemen yang lebih besar ke kanan untuk membuat ruang bagi elemen ketiga.
4. Langkah ini diulang untuk setiap elemen dalam data, di mana setiap elemen dimasukkan ke dalam subkoleksi yang telah diurutkan sebelumnya dengan cara membandingkan dan menyisipkannya ke posisi yang tepat.
5. Setelah semua elemen dimasukkan ke dalam subkoleksi yang telah diurutkan, hasil akhir adalah data yang terurut.



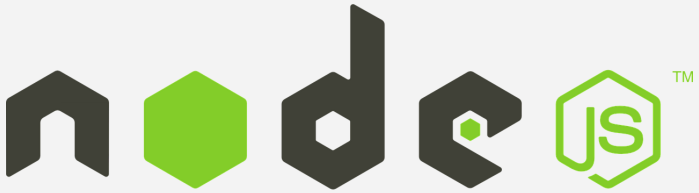
Kelebihan dan Kekurangan

- Keuntungan utama dari Insertion Sort adalah bahwa algoritma ini efektif untuk data yang sudah dalam keadaan hampir terurut atau memiliki jumlah elemen yang sangat sedikit.
- Namun, Insertion Sort tidak efisien untuk data dengan jumlah elemen yang besar karena kompleksitas waktu rata-ratanya adalah O(n^2), di mana n adalah jumlah elemen dalam data.
- Meskipun Insertion Sort tidak digunakan secara luas dalam praktik untuk pengurutan data besar, ia masih bermanfaat sebagai alat pembelajaran dan memahami prinsip dasar pengurutan.

Insertion Sort



Sumber: https://www.geeksforgeeks.org/recursive-insertion-sort/

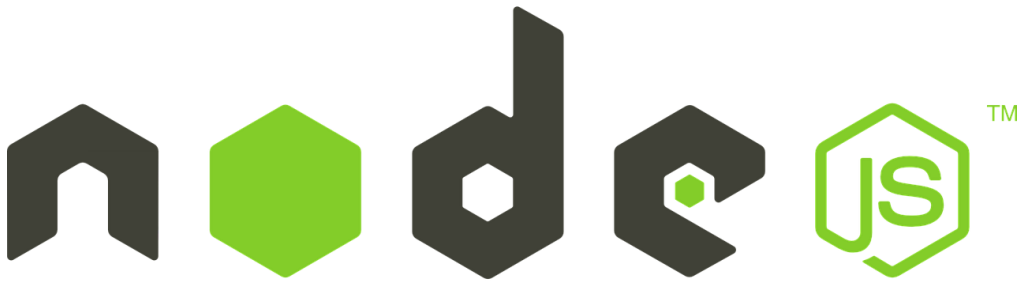


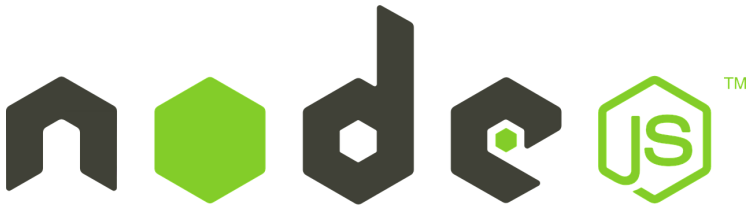
Selection Sort

- Selection Sort adalah salah satu algoritma pengurutan sederhana yang bekerja dengan memilih elemen terkecil dari koleksi data dan menukarkannya dengan elemen pertama.
- Kemudian, algoritma memilih elemen terkecil dari sisa koleksi data (tanpa elemen pertama) dan menukarnya dengan elemen kedua. Proses ini berlanjut hingga seluruh koleksi data terurut.

Cara Kerja Selection Sort

1. Algoritma membagi koleksi data menjadi dua bagian: bagian yang sudah terurut dan bagian yang belum terurut.
2. Algoritma mencari elemen terkecil dalam bagian yang belum terurut dan menukar elemen tersebut dengan elemen pertama dalam koleksi.
3. Algoritma menganggap elemen pertama sebagai bagian yang sudah terurut dan melanjutkan dengan mencari elemen terkecil dalam bagian yang belum terurut (tanpa elemen pertama) untuk menukarkannya dengan elemen kedua.
4. Proses ini terus berlanjut hingga semua elemen dalam koleksi terurut.





Kelebihan dan Kekurangan

- Keuntungan dari Selection Sort adalah bahwa algoritma ini sederhana dan mudah dipahami. Namun, kelemahannya adalah kompleksitas waktu yang buruk.
- Dalam semua kasus, termasuk terbaik dan terburuk, kompleksitas waktu Selection Sort adalah O(n^2), di mana n adalah jumlah elemen dalam koleksi data. Oleh karena itu, algoritma ini tidak efisien untuk koleksi data yang besar.
- Walaupun Selection Sort jarang digunakan dalam aplikasi nyata karena keterbatasan efisiensi, konsepnya masih bermanfaat sebagai cara untuk memahami dasar-dasar pengurutan dan dalam konteks pembelajaran.

Selection Sort