

Optimalisasi Pengurusan Data Dengan Algoritma Selection Sort Studi Kasus Dalam Pengelolaan Stok Barang

Agus Safrianti¹, Ezwarsyah²

¹Program Studi Teknik Informatika, Jurusan Teknik Informatika, Fakultas Teknik, Universitas Malikussaleh

²Program Studi Teknologi Informasi, Jurusan Teknik Informatika, Fakultas Teknik, Universitas Malikussaleh
Email: agus.220170002@mhs.unimal.ac.id, ezwarsyah@unimal.ac.id

ABSTRAK

Pengelolaan stok barang adalah bagian penting dari operasi bisnis yang efektif karena memainkan peran penting dalam memastikan bahwa produk yang dibutuhkan oleh pelanggan selalu tersedia dan bahwa proses bisnis berjalan secara optimal. Manajemen stok barang yang baik tidak hanya meningkatkan kepuasan pelanggan dengan memastikan bahwa produk tersedia tepat waktu, tetapi juga menghemat biaya penyimpanan dan menghindari kekurangan atau kelebihan stok yang dapat mengganggu operasi. Kami meneliti penggunaan algoritma Sort Selection untuk mengelola data stok barang dalam jurnal ini. Algoritma ini dipilih karena mudah digunakan dan efektif untuk dataset kecil hingga menengah. Algoritma ini bekerja dengan memilih elemen terkecil dari bagian yang belum diurutkan dan menukarnya dengan elemen pertama dari bagian tersebut. Prosedur ini diulang hingga seluruh dataset terurut. Penelitian ini menerapkan algoritma Selection Sort pada sistem manajemen stok barang menggunakan bahasa pemrograman Python. Penelitian ini menemukan bahwa algoritma ini dapat meningkatkan efisiensi pengurutan data, membuat pengguna lebih mudah menemukan data yang mereka butuhkan, dan mengurangi jumlah waktu yang diperlukan untuk mengelola stok. Selain itu, sistem ini memudahkan karyawan dan manajer toko untuk mengakses informasi ketersediaan produk dengan cepat dan mudah, yang mempercepat proses pengambilan keputusan dalam pengelolaan stok. Secara keseluruhan, penggunaan algoritma Sorting Selection dalam manajemen stok barang telah terbukti efektif dalam mengoptimalkan proses pengelolaan data. Ini menghasilkan sistem manajemen stok yang lebih teratur, terkontrol, dan efisien, yang pada akhirnya meningkatkan kepuasan pelanggan dan profitabilitas perusahaan. Oleh karena itu, penelitian ini membantu mengembangkan sistem manajemen stok yang lebih baik dan efisien dalam pengelolaan stok barang.

Kata kunci: *Algoritma Selection Sort*, Pengelolaan Stok Barang, Sistem Manajemen Stok, Python, Manajemen Data.

Penulis koresponden : Agus Safrianti

Tanggal terbit : 15 Juni 2024

Tautan : <https://jurnal.komputasi.org/index.php/jst/article/view/25>

1. PENDAHULUAN

Hampir setiap lembaga pemerintahan dan swasta telah menggunakan komputer untuk mengolah data pekerjaan dan bahkan telah dimodifikasi dalam beberapa program software. Ini menunjukkan bahwa komputer bukan hal yang baru lagi di era globalisasi saat ini. Hal ini menunjukkan bahwa keberadaan komputer sangat membantu dan berdampak besar pada tatanan kerja yang membutuhkan proses kerja yang serba cepat. Sampai saat ini, kita telah merasakan berbagai hasil kerja komputer yang menguntungkan kehidupan manusia. Terdapat kelebihan dan kekurangan dari berbagai pola pengurutan data algoritma yang tersedia, dan penggunaan masing-masing algoritma sangat bergantung pada kebutuhan dan kondisi saat ini. Pilihan algoritma pengurutan data biasanya bergantung pada seberapa sederhana atau kering alurnya dan seberapa cepat proses eksekusinya[1]. Beberapa metode pengurutan data adalah Sorting Selection, Sorting Insertion, Sorting Bubble, Sorting Merge, Sorting Quick, Sorting Shell, Sorting Heap, dan Sorting Radix[2].

Pengelolaan stok barang merupakan bagian penting dari operasi bisnis yang efektif. Dalam bisnis modern, pengelolaan stok barang yang baik tidak hanya memastikan bahwa produk yang dibutuhkan pelanggan tersedia, tetapi juga sangat penting untuk menghemat uang dan mengoptimalkan proses bisnis. Selama proses jual beli manual, orang sering lupa mencatat berapa banyak barang yang dibeli, dan catatan masuk dan keluar kadang-kadang tidak sama. Kemudian, ketika orang mencari sesuatu, ternyata barang tersebut sudah habis, dan penjual lupa untuk merestok ulang barang tersebut[3]. Hal-hal seperti ini dapat dipermudah untuk mengingatkan apa yang perlu distok. Jadi pengelolaan stok yang optimal adalah kunci keberhasilan dalam menjaga kepuasan pelanggan dan profitabilitas perusahaan karena memungkinkan perusahaan untuk memenuhi permintaan pelanggan tepat waktu, mengurangi biaya penyimpanan, dan menghindari kekurangan atau kelebihan stok yang dapat mengganggu operasional bisnis.

Algoritma sangat penting untuk perangkat lunak atau pemrograman, jadi Anda harus memahami konsep dasar mereka. Proses pengurutan data menurut aturan tertentu agar terurut secara teratur dikenal sebagai algoritma pengurutan atau sortasi pemilihan. Data yang diurutkan berdasarkan beberapa kriteria sangat penting. Bahasa pemrograman PHP memungkinkan fungsi sortir untuk mengurutkan data array dalam urutan naik atau turun dari nilai terkecil ke nilai terbesar[4].

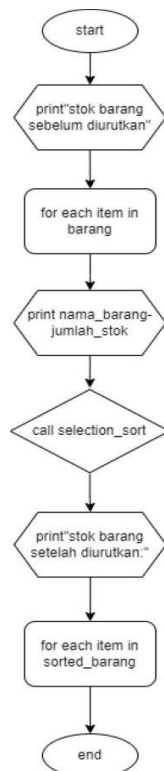
Selection sort menurut Yahya (2014:136) adalah metode pengurutan yang membandingkan elemen saat ini dengan elemen berikutnya hingga elemen terakhir. Jika ada elemen lain yang posisinya lebih kecil dari elemen saat ini, posisinya dicatat dan elemen tersebut langsung ditukar. Metode pemilihan pengurutan melibatkan pemilihan nilai terkecil dan menukarnya dengan elemen paling awal. Kemudian, bandingkan elemen saat ini dengan elemen berikutnya sampai dengan elemen terakhir, dan seterusnya sampai tidak ada lagi pertukaran data[5]. Selection sort adalah salah satu algoritma pengurutan yang paling umum digunakan karena kemudahannya dan kemudahan penggunaan. Namun secara komputasi, algoritma ini tidak selalu menjadi yang paling efisien dibandingkan dengan algoritma pengurutan lainnya seperti quicksort atau mergesort[6].

Berdasarkan latar belakang tersebut, dibutuhkan sebuah sistem yang dapat mempermudah pengelolaan data barang. Penggunaan algoritma seleksi sortir pada sistem pengelolaan stok barang diharapkan dapat membantu pengguna menggunakan data yang ada dengan mudah dan tepat, serta memungkinkan untuk menemukan data yang dibutuhkan dengan cepat dan mudah[7].

2. METODELOGI

Konsep penting dalam ilmu komputer dan pemrograman adalah algoritma sebagai strategi cerdas yang merupakan cara cerdas untuk memecahkan masalah atau isu tertentu. Untuk menggambarkan pendekatan terstruktur dan sistematis untuk mencapai tujuan atau solusi dalam konteks yang dapat diterapkan dalam bahasa pemrograman[8]. Untuk memahami karakteristik, keunggulan, dan keterbatasan algoritma pemilihan sort, analisis dilakukan. Untuk kasus terbaik, rata-rata, dan terburuk, pilihan selection sort adalah algoritma pengurutan sederhana dengan kompleksitas waktu $O(n^2)$. Algoritma ini bekerja dengan menemukan elemen terkecil dari bagian array yang tidak terurut dan menukarnya dengan elemen pertama dari bagian tersebut. Prosedur ini diulang sampai seluruh array terurut. Karena Selection Sort adalah algoritma in-place, itu mudah digunakan dan tidak membutuhkan banyak memori. Namun, kekurangan utamanya adalah tidak efektif untuk dataset yang besar; sebaliknya, itu lebih baik untuk dataset kecil hingga menengah.

Bahasa pemrograman Python digunakan untuk menerapkan algoritma Sort Selection untuk mengelola stok barang. Nama, kategori, jumlah, dan harga barang adalah atribut yang dapat dimiliki oleh objek yang disimpan dalam array data stok.



Gambar 1. Flochart

Tabel 1. Tabel Stok Barang

No	Nama barang	Stok Barang
1	Pensil	20
2	Buku	15
3	Penghapus	10
4	Pensil warna	25
5	Spidol	30

3. HASIL DAN PEMBAHASAN

Pengurutan adalah bagian integral dari dunia komputer. Untuk memenuhi kebutuhan terhadap proses pengurutan, berbagai teknik pengurutan diciptakan. Anda dapat membuat program yang efisien dari segi waktu dan hasil dengan menggunakan algoritma yang baik. Kadang-kadang data berada dalam bentuk yang tidak berpola, atau mereka dapat berada dalam bentuk yang sesuai dengan pola yang diinginkan. Secara umum, dua jenis pengurutan data adalah model naik (ascending), yang mengurutkan data dari nilai terkecil sampai terbesar, dan model turun (descending), yang mengurutkan data dari nilai terkecil sampai terbesar[9].

Dalam penelitian ini, satu metode pengurutan data—Selection Sort—digunakan pada suatu baris data. Data diurutkan secara menurun (dari kecil ke besar), bergerak dari kiri ke kanan. Implementasi dilakukan dengan menggunakan bahasa pemrograman Python.

3.1 Prinsip Kerja

Untuk mengoptimalkan pengurutan data berdasarkan stok barang, algoritma pemilihan sortir bekerja dengan memilih elemen terkecil dari bagian yang belum diurutkan dan menukarnya dengan elemen pertama dari bagian tersebut. Berikut adalah langkah-langkah rinci yang dilakukan oleh algoritma pemilihan sortir:

Data Awal

No	Nama Barang	Stok Barang
1	Pensil	20
2	Buku	15
3	Penghapus	10
4	Pensil warna	25

Iterasi Pertama:

Langkah 1: Inisialisasi min_idx sebagai 0 (Pensil dengan stok 20).

Langkah 2: Bandingkan stok barang pada min_idx dengan indeks berikutnya:

Stok Buku (15) < Stok Pensil (20), jadi min_idx diubah menjadi 1.

Stok Penghapus (10) < Stok Buku (15), jadi min_idx diubah menjadi 2.

Stok Pensil warna (25) > Stok Penghapus (10), jadi min_idx tetap 2.

Langkah 3: Tukar elemen di posisi min_idx (2) dengan elemen di posisi awal (0).

Hasil setelah iterasi pertama

Semua simbol yang telah digunakan dalam persamaan harus didefinisikan dalam teks berikutnya.

```
| No | Nama Barang | Stok Barang |
|----|-----|-----|
| 3 | Penghapus | 10 |
| 2 | Buku | 15 |
| 1 | Pensil | 20 |
| 4 | Pensil warna | 25 |
```

Iterasi Kedua:

Langkah 1: Inisialisasi min_idx sebagai 1 (Buku dengan stok 15).

Langkah 2: Bandingkan stok barang pada min_idx dengan indeks berikutnya:

Stok Pensil (20) > Stok Buku (15), jadi min_idx tetap 1.

Stok Pensil warna (25) > Stok Buku (15), jadi min_idx tetap 1.

Langkah 3: Tidak ada perubahan karena min_idx tetap sama.

Hasil setelah iterasi kedua

```
| No | Nama Barang | Stok Barang |
|----|-----|-----|
| 3 | Penghapus | 10 |
| 2 | Buku | 15 |
| 1 | Pensil | 20 |
| 4 | Pensil warna | 25 |
```

Iterasi Ketiga:

Langkah 1: Inisialisasi min_idx sebagai 2 (Pensil dengan stok 20).

Langkah 2: Bandingkan stok barang pada min_idx dengan indeks berikutnya:

Stok Pensil warna (25) > Stok Pensil (20), jadi min_idx tetap 2.

Langkah 3: Tidak ada perubahan karena min_idx tetap sama.

Hasil setelah iterasi ketiga

No	Nama Barang	Stok Barang
3	Penghapus	10
2	Buku	15
1	Pensil	20
4	Pensil warna	25

Iterasi Keempat:

Tidak ada perubahan yang diperlukan karena hanya satu elemen tersisa yang tidak diurutkan.

Hasil akhir tetap sama

No	Nama Barang	Stok Barang
3	Penghapus	10
2	Buku	15
1	Pensil	20
4	Pensil warna	25

Prinsip untuk Optimalisasi Sorting Efisiensi Algoritma Pilihan:

1. Analisis Kompleksitas: Dalam kasus terbaik, rata-rata, dan terburuk, jenis pilihan memiliki kompleksitas waktu $O(n^2)$. Namun, algoritma ini berfungsi dengan baik untuk dataset yang kecil atau hampir terurut.
2. Penggunaan Memori: Algoritma ini bersifat in-place, artinya hanya membutuhkan beberapa variabel sementara. Ini meningkatkan penggunaan memori.

Optimalisasi dalam konteks manajemen stok barang:

1. Praktik Terbaik: Jika selama satu iterasi tidak ada pertukaran yang terjadi, yang menunjukkan bahwa data sudah terurut, algoritma pemilihan sortir dapat dioptimalkan dengan menghentikan proses.
2. Kondisi Penggunaan: Pemilihan sortir dapat diterapkan untuk pengelolaan stok barang pada sistem dengan dataset kecil hingga sedang, di mana kesederhanaan implementasi lebih penting daripada kecepatan eksekusi.

Dengan mengoptimalkan algoritma pemilihan jenis untuk pengelolaan stok barang, kami dapat mencapai pengurutan yang lebih efisien dengan penggunaan memori yang lebih sedikit, yang membantu meningkatkan kinerja operasional dan pengelolaan inventaris.

3.2 Cara Kerja dan Proses

Untuk mengurutkan data stok barang, algoritma Sort Selection digunakan dalam bahasa pemrograman Python sebagai berikut:

Data Awal

```
python Salin kode  
  
data = [  
    {"No": 1, "Nama Barang": "Pensil", "Stok Barang": 20},  
    {"No": 2, "Nama Barang": "Buku", "Stok Barang": 15},  
    {"No": 3, "Nama Barang": "Penghapus", "Stok Barang": 10},  
    {"No": 4, "Nama Barang": "Pensil warna", "Stok Barang": 25},  
]
```

Implementasi Selection Sort dalam Python

```
def selection_sort(data, key):  
    n = len(data)  
    for i in range(n):  
        min_idx = i  
        for j in range(i + 1, n):  
            if data[j][key] < data[min_idx][key]:  
                min_idx = j  
        data[i], data[min_idx] = data[min_idx], data[i]  
  
    # Mengurutkan data berdasarkan "Stok Barang"  
    selection_sort(data, "Stok Barang")  
  
    # Menampilkan hasil pengurutan  
    for item in data:  
        print(f'No: {item["No"]}, Nama Barang: {item["Nama Barang"]}, Stok Barang: {item["Stok  
Barang"]})')
```

Penjelasan Kode

1. **Fungsi selection_sort:**
 - **Parameter:** Fungsi menerima dua parameter yaitu data (daftar yang akan diurutkan) dan key (kunci berdasarkan mana pengurutan akan dilakukan).
 - **Inisialisasi:** n adalah panjang dari daftar data.
 - **Loop Pertama (i):** Loop ini mengiterasi setiap elemen dalam daftar.
 - **Inisialisasi min_idx:** Menyimpan indeks elemen terkecil yang ditemukan dalam sub-daftar yang belum diurutkan.
 - **Loop Kedua (j):** Loop ini mengiterasi elemen setelah indeks i untuk menemukan elemen dengan nilai terkecil.
 - **Pembandingan:** Jika elemen pada indeks j lebih kecil dari elemen pada indeks min_idx, maka min_idx diubah ke j.
 - **Pertukaran:** Setelah loop kedua selesai, elemen pada indeks i ditukar dengan elemen pada indeks min_idx.
2. **Pengurutan Data:**
 - Fungsi selection_sort dipanggil dengan data dan kunci "Stok Barang" untuk mengurutkan daftar berdasarkan stok barang.
3. **Menampilkan Hasil:**
 - Loop terakhir digunakan untuk menampilkan hasil pengurutan dalam format yang diinginkan.

Hasil Eksekusi

Hasil dari eksekusi kode di atas akan mengurutkan data stok barang dari yang terkecil hingga terbesar:

```
yaml
No: 3, Nama Barang: Penghapus, Stok Barang: 10
No: 2, Nama Barang: Buku, Stok Barang: 15
No: 1, Nama Barang: Pensil, Stok Barang: 20
No: 4, Nama Barang: Pensil warna, Stok Barang: 25
```

Optimasi:

Jika tidak ada pertukaran selama satu iterasi, kami dapat menambahkan logika untuk menghentikan iterasi untuk meningkatkan efisiensi. Namun, dalam algoritma pemilihan sort, iterasi selalu melakukan pertukaran, bahkan jika posisinya tidak berubah, jadi optimasi ini lebih relevan untuk algoritma seperti pemilihan balon. Dengan cara ini, algoritma Sorting Selection dapat diimplementasikan dan dioptimalkan dalam bahasa pemrograman Python untuk mengurutkan data stok barang dengan mudah dan efisien.

4. KESIMPULAN

Dalam jurnal ini akan dibahas bagaimana penerapan algoritma penyortiran telah meningkatkan efisiensi umur simpan produk di toko . Penggunaan algoritma ini membuat proses pemilihan jenis material untuk penggantian stok menjadi lebih fleksibel dan efektif , sehingga menghasilkan sistem manajemen stok yang lebih kuat . Hasilnya , karyawan dan manajer toko dapat mengakses informasi ketersediaan produk dengan cepat dan mudah , sehingga mengurangi jumlah waktu yang diperlukan untuk tugas pembersihan stok . Salah satu kesimpulan dari penelitian ini adalah algoritma pemilihan jenis digunakan untuk membangun pengelolaan stok barang; dan bahwa metode pemilihan sortir ini membuat manajemen stok barang di toko lebih teratur dan terkendali [10]. Oleh karena itu , jurnal ini memberikan informasi tentang bagaimana teknologi dapat diterapkan untuk meningkatkan efisiensi penempatan produk dalam lingkungan yang diatur .

5. DAFTAR PUSTAKA

REFERENCES

- [1] E. Sunandar, “Perbandingan Metode Selection Sort dan Insertion Sort Dalam Pengurutan Data Menggunakan Bahasa Program Java,” *Petir*, vol. 12, no. 2, pp. 172–178, 2019, doi: 10.33322/petir.v12i2.485.
- [2] E. Sunandar, “Implementation Of Bubble Sort Algorithm On 2 Fruit Models Of Data Selection Using The Java Program Language,” *Petir*, vol. 14, no. 2, pp. 159–169, 2021, doi: 10.33322/petir.v14i2.946.
- [3] Z. F. Hakim, “Implementasi Metode Selection Sort Untuk Menentukan Barang Yang Harus Di Stok Ulang Dalam Sistem Informasi Penjualan,” *J. Inf. Eng. Educ. Technol.*, vol. 1, no. 1, p. 18, 2017, doi: 10.26740/jieet.v1n1.p18-26.
- [4] Y. A. Sandria, M. R. A. Nurhayoto, L. Ramadhani, R. S. Harefa, and A. Syahputra, “Penerapan Algoritma Selection Sort untuk Melakukan Pengurutan Data dalam Bahasa Pemrograman PHP,” *Hello World J. Ilmu Komput.*, vol. 1, no. 4, pp. 190–194, 2022, doi: 10.56211/helloworld.v1i4.187.
- [5] A. Syahputra, ““Algoritma Selection Sort” ‘Implementasi Algoritma Selection Sort Untuk Pengurutan Nilai Ipk Mahasiswa Universitas Potensi Utama,’” *J. Tek. Inform. Kaputama*, vol. 6, no. 2, pp. 390–398, 2022.
- [6] F. Aura D.E. and D. Otik, “Penggunaan Algoritma Selection Sort Untuk Menentukan Nilai Tertinggi Siswa,” *JuSiTik J. Sist. dan Teknol. Inf. Komun.*, vol. 6, no. 2, pp. 23–26, 2023, doi: 10.32524/jusitik.v6i2.961.
- [7] M. Ramdhani Yanuarsyah and R. Napianto, “Arsitektur Informasi Pada Sistem Pengelolaan

- Persediaan Barang (Studi Kasus: Upt Puskesmas Rawat Inap Pardasuka Pringsewu),” *J. Teknol. dan Sist. Inf.*, vol. 2, no. 2, pp. 61–68, 2021, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/JTSI>
- [8] N. Mahrozi and M. Faisal, “Analisis Perbandingan Kecepatan Algoritma Selection Sort Dan Bubble Sort,” *J. Ilm. Sain dan Teknol.*, vol. 1, no. 2, pp. 89–98, 2023.
- [9] E. Retnoningsih, “Algoritma Pengurutan Data (Sorting) Dengan Metode Insertion Sort dan Selection Sort,” *Inf. Manag. Educ. Prof.*, vol. 3, no. 1, pp. 95–106, 2018.
- [10] S. N. R. Sika and Putri Aisyiyah Rakhma devi, “Sistem Informasi Persediaan Stok Barang Berbasis Web Pada Toko Putra Gresik,” *J. Fasilkom*, vol. 11, no. 3, pp. 157–164, 2021, doi: 10.37859/jf.v11i3.3163.