E-ISSN : 2988-1986
https://ejournal.cahayailmubangsa.institute/index.php/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 10 No 7 Tahun 2025

IMPLEMENTASI ALGORITMA BUBBLE SORT DALAM PENGURUTAN DATA STOK PRODUK YANG PALING BANYAK TERJUAL (STUDI KASUS: FRESH SMOOTHIE)

Amanda Tantri Sulistiawati [1], Kesya Alya [2], Ayu Nabila [3], Safrizal [4]
[1,2,3] Program Studi Sistem Informasi, Fakultas Ilmu Komputer, STMIK Kaputama Binjai
[4] Program Studi Sistem Informasi, Fakultas Ekonomi Sains dan Teknologi, Universitas Muhammadiyah Asahan
E-mail: amandatantri28@gmail.com [1], keyalyaa298@gmail.com [2], ayunabila16222@gmail.com [3], rizalsyl75@gmail.com [4]

ABSTRACT

This research examines the application of the Bubble Sort algorithm in sorting Fresh Smoothie product stock data based on the number of units sold. The algorithm is used to sort sales data automatically in both ascending and descending order, facilitating better stock management. The system is developed using PHP, CSS, and JavaScript, with MySQL as the database for data storage. The results indicate that the implementation of the Bubble Sort algorithm improves the efficiency of stock and sales management, reduces human errors, and accelerates decision-making processes related to inventory management.

Keywords: Bubble Sort algorithm, data sorting, product stock, inventory management, web-based system.

ABSTRAK

Penelitian ini mengkaji penerapan algoritma *Bubble Sort* dalam pengurutan data stok produk Fresh Smoothie berdasarkan jumlah stok yang terjual. Algoritma ini digunakan untuk mengurutkan data penjualan secara otomatis dalam urutan menaik (*ascending*) dan menurun (*descending*), sehingga memudahkan pengelolaan stok produk. Sistem ini dibangun dengan menggunakan PHP, CSS, dan *JavaScript*, serta *MySQL* sebagai *database* untuk penyimpanan data. Hasil penelitian menunjukkan bahwa penerapan algoritma Bubble Sort dapat meningkatkan efisiensi pengelolaan stok dan penjualan, mengurangi kesalahan manusia, serta mempercepat proses pengambilan keputusan terkait pengelolaan inventaris produk.

Kata kunci: algoritma Bubble Sort, pengurutan data, stok produk, pengelolaan inventaris, sistem berbasis web.

Article History
Received: Oktober 2025
Reviewed:Oktober 2025
Published: Oktober 2025

Plagiarism Checker No 235
Prefix DOI : 10.8734/Kohesi.v1i2.365
Copyright : Author
Publish by : Kohesi

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License

1. PENDAHULUAN

Perkembangan teknologi informasi telah memberikan dampak besar terhadap efisiensi dalam mengelola data, terutama di sektor bisnis makanan dan minuman. Dalam usaha pengelolaan produk, seperti dalam bisnis Fresh Smoothie, tantangan utama adalah mengatur data penjualan dan stok secara efisien. Banyak usaha kecil masih mengandalkan metode manual dalam mencatat dan mengelola stok produk, yang seringkali berujung pada kesalahan dan

pemborosan waktu (Panggabean et.al, 2023). Untuk mengatasi masalah ini, dibutuhkan sistem yang dapat mengotomatisasi proses pengurutan data penjualan sehingga operasional menjadi lebih efisien.

Pada umumnya, bisnis makanan dan minuman seperti Fresh Smoothie, yang memiliki berbagai varian rasa, membutuhkan sistem yang dapat melacak produk yang paling banyak terjual setiap harinya. Dalam hal ini, penting untuk memiliki batasan stok harian yang tidak melebihi kapasitas tertentu, misalnya 50 cup per produk per hari. Tanpa adanya sistem pengurutan yang baik, pengelolaan stok bisa menjadi tidak terkendali (Maria et.al, 2023). Salah satu solusi untuk masalah ini adalah dengan menerapkan algoritma pengurutan yang dapat secara otomatis mengelompokkan produk berdasarkan *volume* penjualannya. Salah satu algoritma yang dapat digunakan untuk tujuan ini adalah *Bubble Sort*, yang memungkinkan pengurutan data secara otomatis dalam urutan menaik atau menurun, sesuai dengan kebutuhan (Aziz et.al, 2022).

Algoritma *Bubble Sort* merupakan salah satu metode pengurutan yang paling sederhana dan banyak digunakan untuk mengelola data berukuran kecil hingga sedang. Dalam penelitian ini, algoritma ini diterapkan untuk mengurutkan data penjualan produk Fresh Smoothie agar dapat memprioritaskan produk yang paling banyak terjual dan memastikan stok tidak melebihi batas yang telah ditetapkan (Trianasyah et.al, 2019). Dengan implementasi sistem berbasis PHP, CSS, dan *JavaScript* serta *MySQL* sebagai *database*, diharapkan pengelolaan data penjualan dan stok dapat berjalan lebih efisien dan efektif (Yulian et.al, 2021). Hal ini akan memudahkan para pengelola untuk memastikan bahwa produk yang paling banyak terjual dapat diidentifikasi dengan cepat dan tepat, membantu pengelolaan inventaris secara otomatis dan mengurangi pemborosan (Aziz et.al, 2022).

2. METODE PENELITIAN

Penelitian ini menggunakan metode pengurutan data dengan dua pendekatan, yaitu *Ascending* (menaik) dan *Descending* (menurun), untuk mengelola data penjualan produk Fresh Smoothie. Algoritma yang digunakan dalam penelitian ini adalah *Bubble Sort*, sebuah metode sederhana namun efektif dalam menyortir data. Data yang digunakan mencakup nama produk, harga, dan stok yang terjual, yang akan diurutkan berdasarkan jumlah penjualan produk (dalam hal ini, jumlah stok yang terjual) dengan dua cara: dari yang paling sedikit terjual (*ascending*) hingga yang paling banyak terjual (*descending*). *Dataset* yang digunakan untuk penelitian ini disajikan dalam tabel berikut sebagai sampel.

Tabel 1.1 Dataset Penjualan Produk Fresh Smoothie

Nama Produk	Harga	Stok
Protein Packed Peanut Butter and Banana Smoothie	Rp 22.000	40/50 CUP
Red Berry and Beet Smoothie	Rp 20.000	45/50 CUP
Green Apple Cucumber Cleanse Smoothie	Rp 25.000	50/50 CUP
Dragon Fruit Smoothie Recipes	Rp 18.000	50/50 CUP
Kiwi Quencher Tropical Smoothie	Rp 25.000	50/50 CUP
Mango Pineapple Paradise Smoothie	Rp 23.000	35/50 CUP
Chocolate Banana Energy Smoothie	Rp 24.000	30/50 CUP

Avocado Honey Cream Smoothie	Rp 21.000	48/50 CUP
Strawberry Yogurt Delight Smoothie	Rp 19.000	44/50 CUP
Blueberry Oat Boost Smoothie	Rp 26.000	32/50 CUP

2.1 Tahapan Penelitian

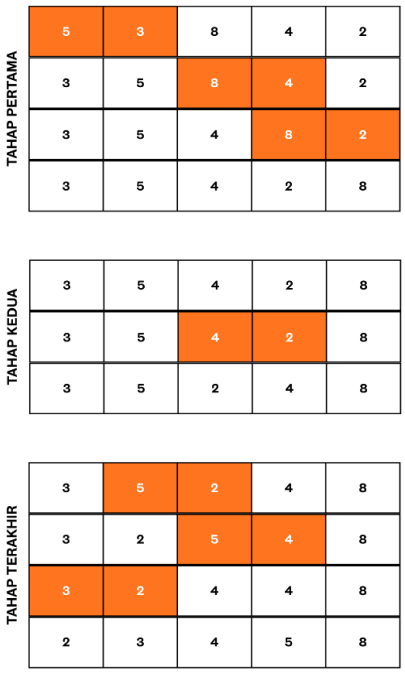
Agar proses penelitian berlangsung sistematis dan terarah, tahapan penelitian dalam studi implementasi algoritma Bubble Sort untuk pengurutan data stok produk Fresh Smoothie disusun sebagai berikut.

1. **Studi Literatur**: Penelitian dimulai dengan mempelajari teori tentang Bubble Sort dan pengelolaan stok produk di sektor makanan dan minuman.
2. **Identifikasi Permasalahan**: Mengidentifikasi kesulitan dalam pengelolaan stok Fresh Smoothie secara manual, yang dapat menyebabkan ketidakteraturan stok.
3. **Pengumpulan Data**: Data penjualan produk dikumpulkan untuk menguji algoritma Bubble Sort dalam pengurutan ascending dan descending.
4. **Implementasi Algoritma**: Algoritma Bubble Sort diterapkan menggunakan PHP, CSS, JavaScript, dan MySQL untuk mengurutkan produk berdasarkan penjualan.
5. **Komparasi dan Pengujian**: Pengujian dilakukan untuk memastikan algoritma bekerja sesuai konsep teoretis, dengan mengevaluasi ketepatan dan efisiensi pengurutan.
6. **Analisis Hasil dan Kesimpulan**: Hasil pengurutan dianalisis untuk menilai efektivitas algoritma dalam mengelola stok produk secara otomatis dan terstruktur.

2.2 Algoritma *Bubble Sort*

Algoritma *Bubble Sort* adalah metode pengurutan sederhana yang membandingkan elemen-elemen berdekatan dan menukarnya jika urutannya salah. Dalam penelitian ini, algoritma ini digunakan untuk mengurutkan stok produk Fresh Smoothie berdasarkan jumlah stok terjual, baik secara menaik (*ascending*) maupun menurun (*descending*). Pada ascending, produk dengan penjualan terendah ditempatkan di awal, sementara pada descending, produk dengan penjualan tertinggi berada di posisi pertama. Proses pengurutan ini dilakukan dengan cara "mengapungkan" elemen terbesar ke posisi akhir array melalui iterasi berulang hingga seluruh data terurut. Meskipun sederhana, Bubble Sort efektif untuk dataset kecil hingga menengah (Sari et al., 2022). Berikut adalah ilustrasi pengurutan dengan algoritma Bubble Sort: [5, 3, 8, 4, 2] → [2, 3, 4, 5, 8].

Gambar 1. Ilustrasi array pengurutan dengan Bubble Sort



Array awal: [5, 3, 8, 4, 2]
Array setelah pengurutan: [2, 3, 4, 5, 8]

2.3 Implementasi Algoritma Bubble Sort Kedalam Rumus

Untuk mengurutkan data menggunakan *Bubble Sort*, kita mendefinisikan langkah-langkah algoritma dalam bentuk rumus sebagai berikut:

Misalkan $A = [a_1, a_2, a_3, \dots, a_n]$ adalah *array* data penjualan produk, di mana $a_i$mewakili jumlah stok yang terjual dan n adalah jumlah produk.

2.3.1 Langkah-langkah Algoritma

1. **Inisialisasi array**

   Mulai dengan array A yang berisi data jumlah stok yang terjual.

2. **Bandingkan dan tukar elemen**

   Bandingkan elemen berturut-turut a_i dan $a_i + 1$. Jika urutannya salah, tukar kedua elemen tersebut:

   $$Jika\ a_i > a_i + 1, maka\ tukar\ a_i\ dengan\ a_i + 1$$

3. **Pengulangan hingga array terurut**

   Ulangi langkah 2 untuk seluruh elemen array, mengabaikan elemen terakhir yang sudah terurut pada setiap iterasi.

4. **Berhenti jika tidak ada pertukaran**

   Jika dalam iterasi tertentu tidak ada pertukaran, array dianggap telah terurut dan algoritma selesai.

2.4 Pengurutan Ascending (Menaik)

Pada metode *ascending*, produk dengan stok terjual paling sedikit akan berada di posisi pertama. Rumus pengurutan *ascending* adalah:

$$Jika\ a_i < a_i + 1, maka\ tukar\ a_i\ dengan\ a_i + 1$$

2.4.1 Penerapan Rumus

Misalnya, kita memiliki data penjualan sebagai berikut:

$$A = [40,45,50,30,35]$$

1. Iterasi pertama membandingkan elemen-elemen dan menukarnya jika perlu:

   40 > 45, tidak ditukar.
   45 > 50, tidak ditukar.
   50 > 30, ditukar $\rightarrow\ A\ =\ [40, 45, 30, 50, 35]$.
   50 > 35, ditukar $\rightarrow\ A\ =\ [40, 45, 30, 35, 50]$.
2. Iterasi berikutnya:

   40 > 45 , tidak ditukar.
   45 > 30, ditukar $\rightarrow A\ =\ [40, 30, 45, 35, 50]$.
   45 > 35, ditukar $\rightarrow A\ =\ [40, 30, 35, 45, 50]$.

2.5 Pengurutan Descending (Menurun)

Pada metode *descending*, produk dengan stok terjual paling banyak akan berada di posisi pertama. Rumus pengurutan *descending* adalah:

$$Jika\ a_i > a_i + 1, maka\ tukar\ a_i\ dengan\ a_i + 1$$

2.5.1 Penerapan Rumus

Misalnya, kita memiliki data penjualan sebagai berikut:

$$A = [40,45,50,30,35]$$

1. Iterasi pertama membandingkan elemen-elemen dan menukarnya jika perlu:

   40 < 45, ditukar $\rightarrow$ A = $[45, 40, 50, 30, 35]$.
   40 < 50, ditukar $\rightarrow$ A = $[45, 50, 40, 30, 35]$.
   40 < 30, tidak ditukar.
   30 < 35, ditukar $\rightarrow$ A = $[45, 50, 40, 35, 30]$.
2. Iterasi berikutnya:

   45 < 50, ditukar $\rightarrow$ A = $[50, 45, 40, 35, 30]$.
   45 < 40, tidak ditukar.
   40 < 35, tidak ditukar.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

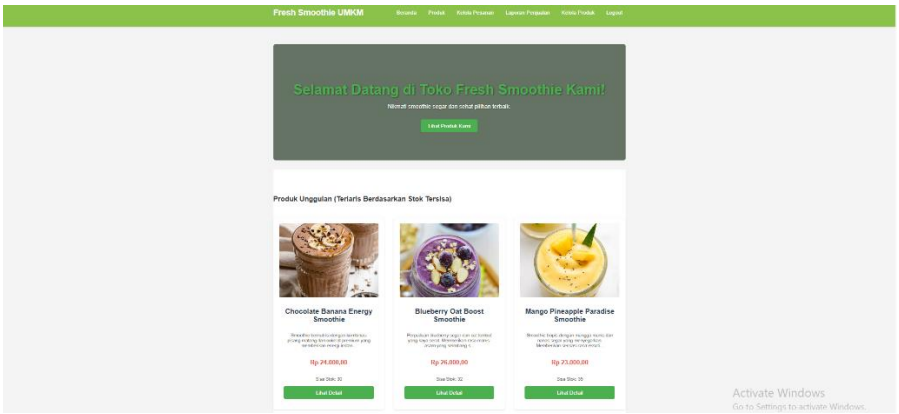
Sistem yang dikembangkan dalam penelitian ini adalah aplikasi berbasis web yang dirancang untuk memudahkan pengelolaan data penjualan dan stok produk Fresh Smoothie. Aplikasi ini dibangun menggunakan teknologi PHP, CSS, dan *JavaScript* untuk antarmuka pengguna, serta MySQL sebagai basis data untuk menyimpan informasi yang diperlukan.

a. Halaman Dashboard (Halaman Utama)

Halaman dashboard merupakan tampilan utama yang muncul ketika admin login ke dalam sistem. Pada halaman ini, admin dapat melihat ringkasan informasi terkait transaksi harian,

seperti jumlah pesanan, total pendapatan, dan produk yang terjual. Menu navigasi pada halaman ini memungkinkan admin untuk mengakses berbagai fitur seperti pengelolaan data produk dan laporan penjualan.

Gambar 2. Tampilan Halaman Dashboard Sistem Fresh Smoothie

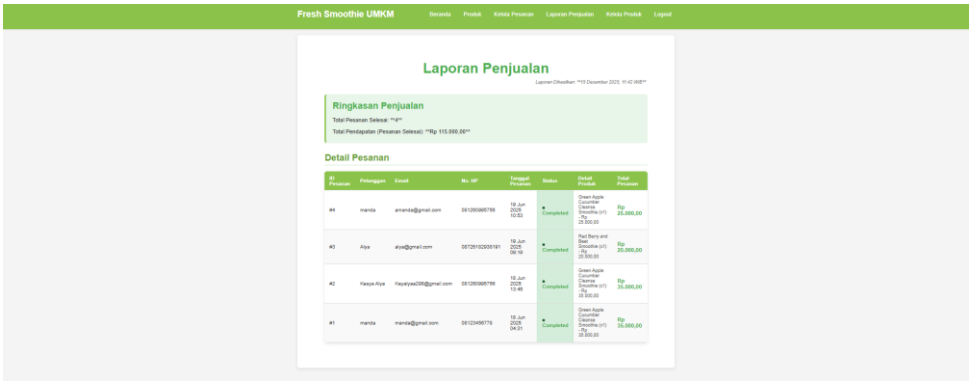


Pada gambar ini, terlihat tampilan halaman utama sistem yang menampilkan produk unggulan. Admin dapat dengan mudah mengakses informasi penting mengenai transaksi dan produk yang paling banyak terjual, serta memonitor pengelolaan data secara efisien.

b. Halaman Data Transaksi

Halaman ini menyajikan daftar lengkap transaksi yang terjadi, dengan rincian informasi seperti ID pesanan, nama pelanggan, Email, No. Hp, Tanggal Pesanan, Status, Detai produk (produk yang dibeli), Total pesanan.

Gambar 3. Tampilan Halaman Data Transaksi Sistem



Gambar ini memperlihatkan tampilan halaman data laporan penjualan yang menampilkan tabel dengan informasi lengkap tentang setiap transaksi. Sistem secara otomatis dapat mengurutkan data berdasarkan berbagai kriteria, seperti nama produk atau tanggal transaksi, untuk memudahkan pengelolaan data. Contoh data transaksi yang ditampilkan pada halaman ini dapat dilihat pada **Tabel 2.** Berikut:

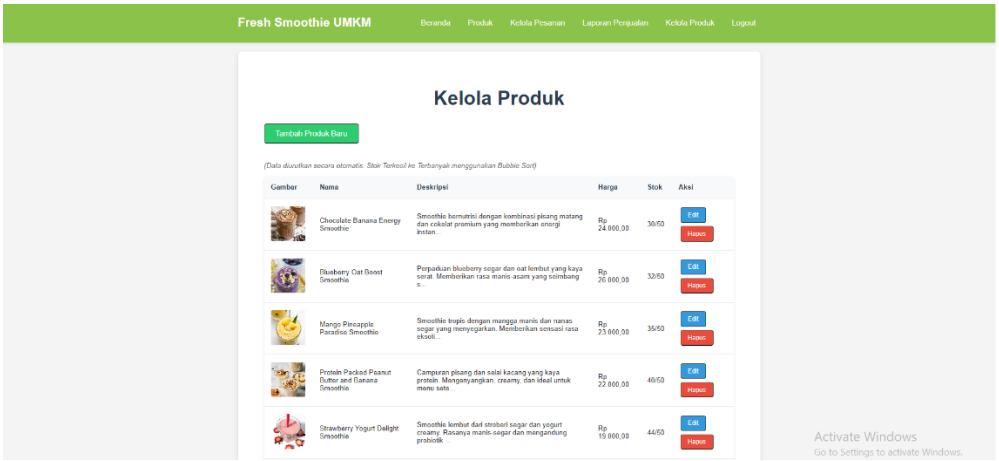
Tabel 2. Data yang Sudah Terurut

Nama Produk	Harga	Stok
Chocolate Banana Energy Smoothie	Rp 24.000	30/50 CUP
Blueberry Oat Boost Smoothie	Rp 26.000	32/50 CUP
Mango Pineapple Paradise Smoothie	Rp 23.000	35/50 CUP
Protein Packed Peanut Butter and Banana Smoothie	Rp 22.000	40/50 CUP
Strawberry Yogurt Delight Smoothie	Rp 19.000	44/50 CUP
Red Berry and Beet Smoothie	Rp 20.000	45/50 CUP
Avocado Honey Cream Smoothie	Rp 21.000	48/50 CUP
Green Apple Cucumber Cleanse Smoothie	Rp 25.000	50/50 CUP
Dragon Fruit Smoothie Recipes	Rp 18.000	50/50 CUP
Kiwi Quencher Tropical Smoothie	Rp 25.000	50/50 CUP

c. Hasil Pengurutan di Sistem

Sistem ini juga memungkinkan pengurutan data transaksi dan produk berdasarkan jumlah penjualan, baik secara menaik (ascending) maupun menurun (descending). Fitur ini membantu admin untuk dengan cepat mengetahui produk mana yang paling banyak terjual pada periode tertentu.

Gambar 4. Hasil Pengurutan di Sistem



Pada gambar ini, terlihat hasil pengurutan produk berdasarkan jumlah stok yang terjual. Produk dengan jumlah penjualan tertinggi akan muncul di urutan teratas, sesuai dengan pengaturan pengurutan yang dipilih oleh admin.

3.2 Pengujian Rumus Berdasarkan Metode Ascending

Rumus pengurutan menggunakan algoritma Bubble Sort untuk urutan menaik (ascending) dapat dinyatakan dengan rumus matematis berikut:

$$Jika\ a_i > a_i + 1,\ maka\ tukar\ a_i\ dan\ a_i + 1$$

Di mana:

- a_i adalah elemen ke-i dari array (stok yang terjual untuk produk ke-i),
- a_i + 1 adalah elemen ke-(i+1) dari array (stok yang terjual untuk produk ke-(i+1)).

Misalkan data awal yang akan diurutkan adalah:

$$\boldsymbol{A = [40, 45, 50, 30, 35]}$$

Iterasi 1:

1. Bandingkan
   $a_1 = 40$ dan $a_2 = 45$: $a_1 < a_2$, tidak perlu ditukar.

   Hasil:

   $$A = [40,45,50,30,35]$$

2. Bandingkan $a_2 = 45$ dan $a_3 = 50$: $a_3 < a_3$, tidak perlu ditukar.

   Hasil:

   $$A = [40,45,50,30,35]$$

3. Bandingkan $a_3 = 50$ dan $a_4 = 30$: $a_3 > a_4$, tukar a_3 dan a_4

   Hasil:

   $$A = [40,45,30,50,35]$$

4. Bandingkan $a_4 = 50$ dan $a_5 = 35$: $a_3 > a_4$, tukar a_4 dan a_5

   Hasil:

   $$A = [40,45,30,35,50]$$

Iterasi 2:

1. Bandingkan $a_1 = 40$ dan $a_2 = 45$: $a_1 > a_2$, tidak perlu ditukar.

   Hasil:

   $$A = [40,45,30,35,50]$$

2. Bandingkan $a_2 = 45$ dan $a_3 = 30$: $a_2 > a_3$, tukar a_2 dan a_3

   Hasil:

   $$A = [40,30,45,35,50]$$

3. Bandingkan $a_3 = 45$ dan $a_4 = 35$: $a_3 > a_4$, tukar a_3 dan a_4

   Hasil:

   $$A = [40,30,35,45,50]$$

Iterasi 3:

1. Bandingkan $a_1 = 40$ dan $a_2 = 30$: $a_1 > a_2$, tukar a_1 dan a_2

   Hasil:

$$A = [30,40,35,45,50]$$

2. Bandingkan $a_2 = 40$ dan $a_3 = 35$: $a_2 > a_3$, tukar a_2 dan a_3

   Hasil:

$$A = [30,35,40,45,50]$$

Hasil ini menunjukkan bahwa algoritma Bubble Sort berhasil mengurutkan data penjualan produk secara menaik (ascending), mulai dari yang paling sedikit terjual hingga yang paling banyak terjual. Pengurutan ini membantu dalam pengelolaan stok untuk mengetahui produk dengan penjualan terendah hingga tertinggi, yang sangat berguna bagi pengelola untuk mengambil langkah-langkah yang diperlukan dalam pengelolaan stok.

3.3 Pengujian Rumus Berdasarkan Metode Descending

Rumus pengurutan menggunakan algoritma Bubble Sort untuk urutan menurun dapat dinyatakan dengan rumus matematis berikut:

$$Jika\ a_i < a_i + 1,\ maka\ tukar\ a_i\ dan\ a_i + 1$$

Di mana:

- a_i adalah elemen ke-i dari array (stok yang terjual untuk produk ke-i),
- $a_i + 1$ adalah elemen ke-(i+1) dari array (stok yang terjual untuk produk ke-(i+1)).

Misalkan data awal yang akan diurutkan adalah:

$$\boldsymbol{A = [40, 45, 50, 30, 35]}$$

Iterasi 1:

1. Bandingkan

$$a_1 = 40 \text{ dan } a_2 = 45 :\ a_1 < a_2, tukar\ a_1\ dan\ a_2$$

   Hasil:

$$A = [45,40,50,30,35]$$

2. Bandingkan

$$a_2 = 40 \text{ dan } a_3 = 50 :\ a_2 < a_3, tukar\ a_2\ dan\ a_3$$

   Hasil:

$$A = [45,50,40,30,35]$$

3. Bandingkan

$$a_3 = 40 \text{ dan } a_4 = 30 : \ a_3 > a_4, tukar\ a_3\ dan\ a_4$$

Hasil tetap:

$$A = [45,50,40,30,35]$$

4. Bandingkan

$$a_4 = 30 \text{ dan } a_5 = 35 : \ a_4 < a_5, tukar\ a_4\ dan\ a_5$$

Hasil:

$$A = [45,50,40,35,30]$$

Iterasi 2:

1. Bandingkan

$$a_1 = 40 \text{ dan } a_2 = 50 : \ a_1 < a_2, tukar\ a_1\ dan\ a_2$$

Hasil:

$$A = [50,45,40,35,30]$$

2. Bandingkan

$$a_2 = 45 \text{ dan } a_3 = 40 : \ a_2 > a_3, tukar\ a_2\ dan\ a_3$$

Hasil tetap:

$$A = [50,45,40,35,30]$$

3. Bandingkan

$$a_3 = 45 \text{ dan } a_4 = 40 : \ a_3 > a_4, tukar\ a_3\ dan\ a_4$$

Hasil tetap:

$$A = [50,45,40,35,30]$$

Iterasi 3:

1. Bandingkan

$$a_1 = 50 \text{ dan } a_2 = 45 : \ a_1 > a_2, tukar\ a_1\ dan\ a_2$$

Hasil tetap:

$$A = [50,45,40,35,30]$$

2. Bandingkan

$$a_1 = 50 \text{ dan } a_2 = 45 : \ a_1 > a_2, tukar\ a_1\ dan\ a_2$$

Hasil Akhir:

$$A = [50,45,40,35,30]$$

Hasil ini menunjukkan bahwa algoritma Bubble Sort berhasil mengurutkan data penjualan produk secara menurun (descending), mulai dari yang paling banyak terjual hingga yang paling sedikit terjual.

3.4 Penerapan Kode Program Algoritma Bubble Sort pada Sistem

Pada sistem pengelolaan stok produk Fresh Smoothie, algoritma Bubble Sort diterapkan untuk mengurutkan produk berdasarkan jumlah stok yang terjual. Algoritma ini mempermudah pengelolaan data dengan mengurutkan produk secara otomatis, baik dalam urutan menaik (ascending) maupun menurun (descending), sesuai dengan kebutuhan pengguna. Algoritma ini bekerja dengan cara membandingkan elemen yang berdekatan dalam array dan menukarnya jika urutannya tidak sesuai, sehingga data dapat diurutkan dengan efisien. Berikut adalah kode program yang mengimplementasikan algoritma Bubble Sort pada sistem:

Tabel 3. Kode Program Algoritma Bubble Sort

```
// --- FUNGSI ALGORITMA BUBBLE SORT ---
function bubbleSort($arr, $key, $direction = 'asc') {
    $n = count($arr);
    // Jika array kosong atau hanya berisi 1 elemen, tidak perlu diurutkan
    if ($n <= 1) {
        return $arr;
    }
    // Lakukan N-1 iterasi utama (untuk setiap elemen)
    for ($i = 0; $i < $n - 1; $i++) {
        // Bandingkan elemen yang berdekatan dan lakukan pertukaran
        for ($j = 0; $j < $n - $i - 1; $j++) {
            $compare = false;
            // Logika Perbandingan
```

```
            if ($direction === 'asc') {
                // Urutan Naik (ASC/Terkecil): Tukar jika elemen saat ini lebih besar dari elemen berikutnya
                if ($arr[$j][$key] > $arr[$j+1][$key]) {
                    $compare = true;
                }
            } else {
                // Urutan Turun (DESC/Terbesar): Tukar jika elemen saat ini lebih kecil dari elemen berikutnya
                if ($arr[$j][$key] < $arr[$j+1][$key]) {
                    $compare = true;
                }
            }
            // Pertukaran (Swap)
            // Jika $compare true, elemen $j dan $j+1 ditukar posisinya
            if ($compare) {
                $temp = $arr[$j];
                $arr[$j] = $arr[$j+1];
                $arr[$j+1] = $temp;
            }
        }
    }
    return $arr;
}
// --- AKHIR FUNGSI BUBBLE SORT ---
```

Hasil pengurutan dengan menggunakan algoritma ini memungkinkan admin untuk dengan mudah mengetahui produk yang paling banyak terjual atau yang penjualannya paling sedikit, tergantung pada arah pengurutan yang dipilih. Pengurutan ini tidak hanya meningkatkan efisiensi dalam mengelola stok produk, tetapi juga memastikan bahwa data penjualan produk dapat diakses dengan lebih terstruktur dan tepat waktu. Dengan demikian, sistem ini dapat membantu dalam pengambilan keputusan yang lebih cepat dan tepat terkait dengan pengelolaan stok.

KESIMPULAN

Algoritma *Bubble Sort* yang diterapkan dalam sistem pengelolaan stok produk Fresh Smoothie terbukti efektif dalam mengurutkan data penjualan berdasarkan jumlah stok yang terjual, baik dalam urutan menaik maupun menurun. Dengan menggunakan kode program berbasis PHP, CSS, dan *JavaScript*, pengurutan data dapat dilakukan secara otomatis, memudahkan pengelola untuk mengetahui produk yang paling banyak atau sedikit terjual. Hasil pengujian menunjukkan bahwa algoritma ini meningkatkan efisiensi pengelolaan stok dan penjualan, mengurangi kesalahan manusia, serta memungkinkan pengambilan keputusan yang lebih cepat dan tepat dalam pengelolaan inventaris produk.

DAFTAR PUSTAKA

Maria, P. S., & Susianti, E. (2023). Implementasi algoritma bubble sort untuk ekstraksi kode biner pada Intel hex file. *Jurnal Fokus Elektroda: Energi Listrik, Telekomunikasi, Komputer, Elektronika dan Kendali*, 8(4), 226–233.

Hidayad, R., Basukesti, A., & Indrianingsih, Y. (2013). Implementasi load balancing dengan metode bubble sort studi kasus pengisian KRS di STTA. *Jurnal Komputasi*, 2(1), 131–142.

Triansyah, H., Lubis, G. R., Gunawan, I., & Husaini, M. F. (2019). Implementasi metode bubble sort dalam pengurutan indeks prestasi mahasiswa. *Jurnal Ilmiah Informatika*, 7(1), 49–59.

Panggabean, A. B., Ramadhan Htb, R., Perina, I., & Toro, Y. L. (2023). Implementasi algoritma bubble sort pada sistem pelayanan perpustakaan menggunakan Laravel. *Sudo Jurnal Teknik Informatika*, 2(1). https://doi.org/10.56211/sudo.v2i1.185.

Sitepu, R. R., Yusman, M., & Febriansyah, F. E. (2017). Implementasi algoritma bubble sort dan selection sort menggunakan arraylist multidimensi pada pengurutan data multi prioritas. *Jurnal Komputasi*, 5(1), 81-87.

Ragil, G. S., Supriady, & Hamidin, D. (2023). Implementasi algoritma bubble sort dengan menggunakan metode agile pada perancangan aplikasi sistem layanan informasi rukun warga. *Jurnal Teknik Informatika*, 15(3), 110–121.

Aziz, T., & Asmiatun, S. (2022). Penerapan algoritma bubble sort pada sistem perpustakaan menggunakan framework CodeIgniter. *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, 3(1), 24–33.

Ramadhan, R. P., Widiono, S., & Pajri, R. (2025). Pengembangan aplikasi pemesanan makanan berbasis Android dengan algoritma bubble sort. *Jurnal Informatika Teknologi dan Sains (JINTEKS)*, 7(4), 1722–1727.

Yulian, R. S., Andryana, S., & Sholihati, I. D. (2021). Analisis perbandingan algoritma bubble sort dan selection sort pada sistem pendukung keputusan pemilihan tempat kost berbasis

iOS (iPhone Operating System). *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 6(2), 318–324.

Sari, N., Gunawan, W. A., Khofipah, P., Zikri, I., & Syahputra, A. (2023). Analisis algoritma bubble sort secara ascending dan descending serta implementasinya menggunakan bahasa pemrograman Java. *Jurnal Teknik Informatika*, 7(3), 1-7.