

# Implementasi Algoritma *Quick Sort* dalam Sistem Pemesanan Makanan di Rumah Makan

Desi Rosdiana<sup>1</sup>, Amri<sup>2</sup>

<sup>1</sup>Program Studi Teknik Informatika, Jurusan Teknik Informatika, Fakultas Teknik, Universitas Malikussaleh

<sup>2</sup>Program Studi Jurusan Teknologi Informasi dan Komputer, Politeknik Negeri Lhokseumawe

Email: desi.220170203@mhs.unimal.ac.id, amri@pnl.ac.id

---

---

## ABSTRAK

Penjadwalan pekerjaan adalah proses krusial dalam manajemen waktu dan sumber daya, khususnya dalam konteks manufaktur dan produksi. Artikel ini mengeksplorasi penerapan algoritma greedy untuk penjadwalan pekerjaan dengan tujuan memaksimalkan keuntungan, dengan mempertimbangkan tenggat waktu, tingkat kesulitan, dan durasi pekerjaan. Studi kasus yang digunakan mencakup lima pekerjaan berbeda: membuat kemeja, topi, syal, kaos, dan jeans. Setiap pekerjaan memiliki karakteristik unik seperti tenggat waktu, kesulitan, durasi, dan benefit yang bervariasi. Algoritma greedy diimplementasikan dengan mengutamakan pekerjaan yang memiliki rasio benefit per jam tertinggi. Hasil penelitian menunjukkan bahwa metode ini efektif dalam memberikan solusi penjadwalan yang optimal, dengan total keuntungan mencapai Rp 580.000 dengan pengalokasian jobs. Penelitian ini menegaskan bahwa algoritma greedy dapat memberikan solusi yang efisien dan optimal dalam penjadwalan pekerjaan. Meskipun algoritma greedy tidak selalu menjamin solusi optimal untuk semua kasus, dalam konteks studi kasus ini, hasil yang didapat sangat memuaskan. Diharapkan hasil ini akan menunjukkan peningkatan efisiensi dan keuntungan dalam penjadwalan pekerjaan.

Kata kunci: penjadwalan pekerjaan, algoritma *greedy*, optimasi waktu, keuntungan

---

Penulis koresponden : Hayatun Nufus

Tanggal terbit : 15 Juni 2024

Tautan : <https://jurnal.komputasi.org/index.php/jst/article/view/28>

---

---

## 1. PENDAHULUAN

Pemesanan makanan berbasis pemrograman merupakan sebuah inovasi teknologi yang telah membawa manfaat besar bagi industri penjualan[1]. Dengan menggunakan aplikasi ini, pegawai dapat lebih efisien dalam melayani pelanggan, sehingga waktu yang dibutuhkan untuk proses pemesanan dapat diminimalisir.

Pemesanan barang, terutama makanan, seringkali menjadi tugas yang membingungkan jika daftar menu yang tersedia tidak terorganisir dengan baik. Dalam situasi ini, penting untuk memiliki sistem pengelolaan data yang efisien untuk mengatur daftar menu dengan lebih mudah. Salah satu metode yang umum digunakan dalam pengelolaan data adalah pengurutan, atau sorting. Sorting merupakan proses menyusun angka atau karakter yang semula acak menjadi urutan yang lebih terstruktur dan mudah dicari[2]. Ada berbagai algoritma pengurutan yang tersedia, dan salah satu yang paling populer dalam dunia pemrograman adalah algoritma quick sort.

Algoritma quick sort dikembangkan oleh seorang ilmuwan komputer asal Inggris, Tony Hoare, pada tahun 1959 dan dipublikasikan pada tahun 1961[2]. Keunggulan utama dari algoritma quick sort adalah efisiensi dalam penggunaan memori serta kemampuannya untuk melakukan pengurutan dengan cepat dibandingkan dengan algoritma pengurutan lainnya[3]. Oleh karena itu, algoritma quick sort sering dipilih dalam sistem pemesanan makanan untuk mencapai beberapa tujuan utama.

Pertama, penggunaan algoritma quick sort bertujuan untuk meminimalkan kesalahan dalam pencatatan pesanan. Dengan daftar menu yang tersusun rapi, kasir atau pegawai dapat dengan mudah mencatat pesanan pelanggan tanpa kebingungan atau kesalahan penulisan.

Kedua, algoritma quick sort membantu dalam mempercepat waktu transaksi bagi setiap pelanggan. Dengan daftar menu yang terurut dengan baik, proses pemilihan dan pencatatan pesanan menjadi lebih cepat dan efisien, menghemat waktu baik bagi pelanggan maupun kasir[4].

Selain itu, algoritma quick sort juga mendukung dalam menyusun transaksi pemesanan secara rapi dan benar. Dengan daftar menu yang teratur, transaksi dapat disusun dengan urutan yang logis dan memudahkan dalam pelacakan serta pemrosesan pesanan.

## 2. METODELOGI

### A. Metode Pengumpulan Data

Melakukan pengamatan dengan memeriksa dan mengumpulkan informasi dari daftar menu yang akan dimasukkan ke dalam program. Proses ini mencakup mengumpulkan data tentang berbagai pilihan menu, termasuk harga dan detail lain yang penting untuk proses pemesanan[5].

### B. Metode Pengembangan Sistem

#### 1) Pendefinisian Kebutuhan

Langkah pertama dalam mempersiapkan program adalah dengan mendefinisikan dengan jelas permasalahan yang hendak diselesaikan. Ini melibatkan identifikasi kebutuhan pengguna serta tujuan dari aplikasi pemesanan makanan yang akan dibuat. Dengan demikian, kita dapat memahami jenis program yang diperlukan dalam konteks ini.

#### 2) Merancang Alur Pembuatan Program

Merancang kode program yang melibatkan penulisan dan pengkodean fungsi-fungsi yang dapat dilakukan dalam berbagai bahasa pemrograman seperti Python, C++, atau Java. Untuk pengembangan versi webnya, PHP dan MySQL dapat digunakan, di mana PHP digunakan untuk membuat halaman web dan MySQL sebagai basis data untuk membangun aplikasi web[6].

#### 3) Pengujian Program

Setelah program selesai dibuat, tahap penting selanjutnya adalah melakukan pengujian untuk memastikan bahwa semua fitur berfungsi dengan baik dan tidak ada bug atau kesalahan lain yang dapat mengganggu kinerja program. Proses ini dilakukan secara menyeluruh dan berkala untuk memastikan bahwa program tetap stabil dan dapat diandalkan. Dengan melakukan pengujian secara berulang, kita dapat memastikan bahwa program memiliki kualitas yang baik dan siap digunakan oleh pengguna.

### 2.1 Simulasi Algoritma Quick Sort

Simulasi pengurutan menggunakan algoritma quick sort dimulai dengan memilih salah satu angka sebagai pivot. Proses ini dimulai dengan menginisialisasi dua variabel, yaitu  $i$  untuk bagian paling kiri dari data dan  $j$  untuk bagian paling kanan. Pivot biasanya dipilih dari nilai tengah dari data yang akan diurutkan. Selanjutnya, nilai  $i$  akan bergerak ke arah kanan dan nilai  $j$  akan bergerak ke arah kiri, mencari nilai yang harus ditukar posisinya. Ketika nilai  $i$  menemukan angka yang lebih besar dari pivot, dan  $j$  menemukan angka yang lebih kecil dari pivot, keduanya akan bertukar posisi[7]. Proses ini akan terus berlanjut hingga tidak ada lagi nilai yang perlu ditukar.

Selama proses ini berlangsung, setiap nilai yang ditukar posisinya akan bergerak ke posisi yang sesuai dengan urutan yang diinginkan. Ini akan terus berlanjut hingga seluruh data terurut dengan benar, menghasilkan entropi yang minimal atau semua data telah terurut[8].

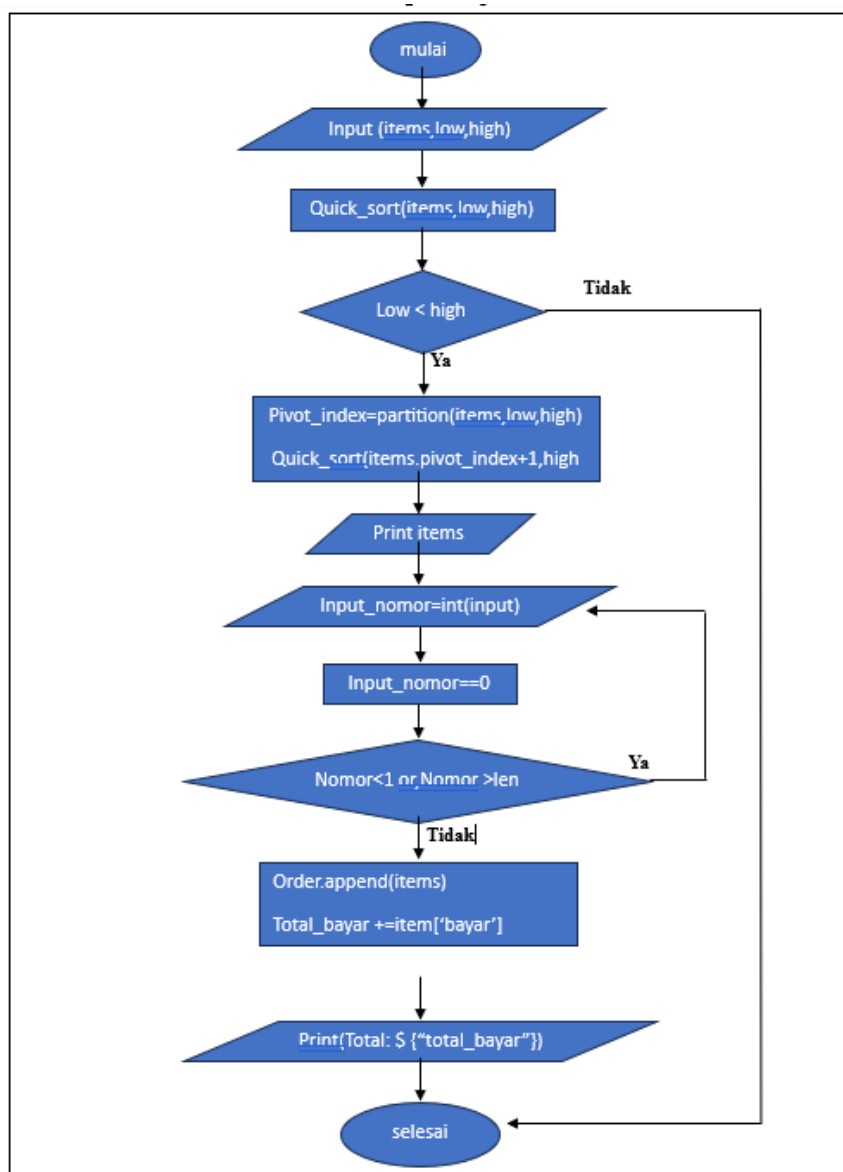
Proses ini memastikan bahwa setiap nilai data ditempatkan di posisi yang sesuai, sehingga data menjadi terurut secara efisien dan efektif. Dengan demikian, hasil pengurutan dapat memenuhi kebutuhan pengguna dengan baik dan memberikan hasil yang optimal dalam waktu yang relatif singkat.

No	Keterangan	Proses pengurutan
1	Deklarasikan nilai arraynya data={4 5 2 12 10 8 6 9}	4   5   2   12   10   8   6   9
2	Tentukan nilai pivotnya dan pilih nilai yang ditengah	10
3	$4 < 10$ true maka tidak berpindah posisi	4   10
4	$5 < 10$ True maka tidak berpindah posisi	4   5   10
5	$2 < 10$ True maka tidak berpindah posisi	4   5   2   10
6	$12 < 10$ False maka 12 berpindah ke ruas kanan	4   5   2   10   12
7	$8 > 10$ False maka 8 berpindah ke ruas kiri	4   5   2   8   10   12
8	$6 > 10$ False maka 6 berpindah ke ruas kiri	4   5   2   8   6   10   12
9	$9 > 10$ False maka 9 berpindah ke ruas kiri	4   5   2   8   6   9   10   12

10	Karna partisi sebelah kanan pivot sudah terurut maka kita akan menentukan pivot lagi	4   5   2   8   6   9   10   12
11	$4 < 2$ False maka 4 berpindah ke ruas kanan	5   2   4   8   6   9   10   12
12	$5 < 2$ False maka 5 berpindah ke ruas kanan	2   4   5   8   6   9   10   12
13	Partisi sebelah kiri sudah terurut maka kita ambil pivot tengah lagi	2   4   5   8   6   9   10   12
14	$6 > 8$ False maka 6 berpindah ke ruas kiri	2   4   5   6   8   9   10   12
15	$9 > 8$ True maka 9 tidak berpindah posisi	2   4   5   6   8   9   10   12
16	Sehingga menghasilkan perurutan yang maksimal	2   4   5   6   8   9   10   12

Tabel 1. Simulasi algoritma quick sort

## 2.2 Flowchart dari Algoritma Quick Sort



Gambar 1. Contoh Flowchart dari algoritma quick sort

### 2.3 Pseudocode

- Program pemesanan menggunakan algoritma quick sort
- Deskripsi  
Var items, low, high
- Implementasi  
quick\_sort(items, low, high)  
If low < high:  
  pivot\_index= partition (items,low,high)  
  quick\_sort(items, low, pivot\_index-1)  
  quick\_sort(items, pivot\_index + 1,high)  
  pivot = items [high][‘name’]  
  I = low -1  
  For j inn range (low,high):  
  If items[j][‘name’]<pivot:  
  I+=1  
  Items[i], items[j] = items[i]  
  Item[i+1], item[high], items[i+1]  
  
  Return i+1

### 3. HASIL DAN PEMBAHASAN

```
Menu:
1. Aqua - $5.00
2. Indomie - $10.00
4. Mie Aceh - $10.00
5. Nasi Goreng - $12.00
6. Teh Dingin - $5.00
7. cappucino - $5.00

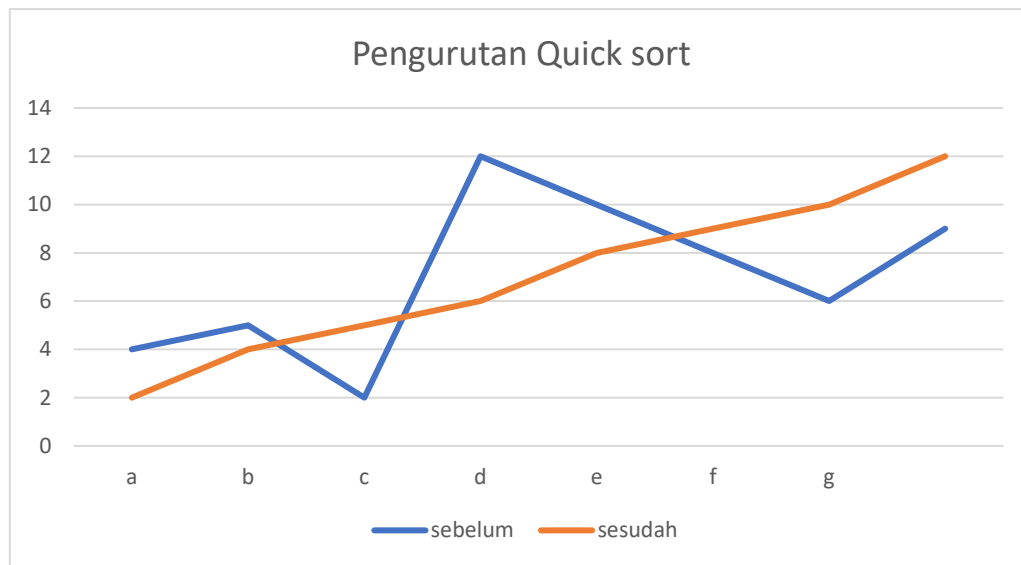
Silahkan melakukan pemesanan dengan memasukkan nomor dari makanan dan minuman yang ingin anda pesan.
Tekan '0' ketika kamu selesai melakukan pemesanan.
ketikkan nomor dari makanan dan minuman yang ingin anda: 3
ketikkan nomor dari makanan dan minuman yang ingin anda: 5
ketikkan nomor dari makanan dan minuman yang ingin anda: 2
ketikkan nomor dari makanan dan minuman yang ingin anda: h
Masukkan tidak valid. silahkan untuk memasukkan angka
ketikkan nomor dari makanan dan minuman yang ingin anda: 9
Masukkan tidak valid.Silahkan masukkan angka antara 1 dan 7
ketikkan nomor dari makanan dan minuman yang ingin anda: 4
ketikkan nomor dari makanan dan minuman yang ingin anda: 0

Pesanan Anda:
Kopi - $5.00
Nasi Goreng - $12.00
Indomie - $10.00
Mie Aceh - $10.00
Total: $37.00
PS C:\xampp\htdocs\desain analisis> █
```

Gambar 2 contoh pemrograman dalam pemesanan makanan

Dengan menerapkan algoritma quick sort, kita dapat mengembangkan sebuah program pemesanan makanan yang efisien dan mudah digunakan oleh para pegawai. Program ini memudahkan pegawai dalam beberapa hal. Pertama-tama, pegawai dapat dengan cepat mencari menu berdasarkan urutan abjad, sehingga proses pencarian menjadi lebih lancar dan efisien. Setelah itu, pegawai dapat memasukkan semua nomor urut dari daftar menu yang dipesan oleh pelanggan ke dalam sistem. Dengan demikian, tidak ada kebingungan atau kesalahan dalam mencatat pesanan.

Selanjutnya, program akan secara otomatis menghitung total harga makanan dan minuman yang harus dibayarkan oleh pelanggan. Hal ini menghilangkan kebutuhan untuk melakukan perhitungan secara manual, yang dapat memakan waktu dan meningkatkan risiko kesalahan. Dengan menggunakan program ini, proses transaksi menjadi lebih cepat dan efisien[9], sehingga memungkinkan pegawai untuk melayani pelanggan dengan lebih baik dan lebih cepat. Dengan demikian, penerapan algoritma quick sort dalam pengembangan program pemesanan makanan tidak hanya membantu mengorganisir daftar menu dengan baik, tetapi juga meningkatkan efisiensi dan kecepatan dalam proses pelayanan.



Gambar 3. Implementasi pengurutan menggunakan quick sort

Karena menu yang terdapat dalam kumpulan huruf bukan angka, maka dalam mengimplementasikan algoritma quicksort, kita menggunakan pendekatan yang disebut "key comparison". Proses ini melibatkan pemilihan elemen kunci yang didasarkan pada nilai dari data yang diurutkan, yaitu huruf atau karakter. Proses pengerjaannya dimulai dengan memilih nilai maksimum dan minimum dari sekumpulan data, sehingga dapat membantu dalam pembagian data selama proses pengurutan menggunakan algoritma quicksort. Dengan menggunakan kunci elemen yang sesuai, algoritma quicksort dapat mengurutkan kumpulan huruf dengan efisien dan akurat[10].

#### 4. KESIMPULAN

Sistem pemesanan ini dirancang dengan tujuan utama untuk memastikan bahwa daftar menu tersusun secara teratur dan rapi, sesuai dengan urutan huruf awalnya. Dengan demikian, pegawai dapat dengan mudah mencatat pesanan menu dan proses pembayaran menjadi lebih efisien. Melalui sistem ini, pencatatan pemesanan tidak hanya menjadi lebih cepat tetapi juga lebih akurat dibandingkan dengan pencatatan manual yang rentan terhadap kesalahan manusia. Penggunaan sistem pemesanan juga membantu mengoptimalkan penggunaan waktu, mengurangi risiko kesalahan, dan meningkatkan produktivitas pegawai. Selain itu, sistem ini memberikan keuntungan bagi pelanggan dengan mempercepat proses layanan dan mengurangi waktu tunggu saat restoran sedang ramai. Dengan antrian yang lebih singkat, pelanggan dapat merasakan pengalaman yang lebih nyaman dan efisien ketika melakukan pemesanan. Keseluruhan, implementasi sistem pemesanan ini tidak hanya membawa manfaat bagi pihak restoran dalam hal efisiensi operasional, tetapi juga meningkatkan kepuasan pelanggan melalui pelayanan yang lebih cepat dan efektif.

#### 5. DAFTAR PUSTAKA

- [1] M. Fauzan Osama, I. Purnamasari, and R. Mayasari, "Rancang Bangun Aplikasi Pemesanan Kue Berbasis Multiplatform Untuk Mendukung Langkah Ekonomi Digital," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 4, pp. 2877–2883, 2024, doi: 10.36040/jati.v7i4.7276.
- [2] M. Luthfi Zulfa, B. Nurina Sari, and U. Singaperbangsa Karawang Abstract, "Analisis Perbandingan Algoritma Bubble Sort, Shell Sort, dan Quick Sort dalam Mengurutkan Baris Angka Acak menggunakan Bahasa Java," *J. Ilm. Wahana Pendidik.*, vol. 2022, no. 13, pp. 237–246, doi: 10.5281/zenodo.6962346.
- [3] D. Rizka Poetra and N. Hayati, "Performa Algoritma Bubble Sort Dan Quick Sort Pada Framework Flutter Dan Dart SDK (Studi Kasus Aplikasi E-Commerce)," *Hal: Jl. Sawo Manila*, vol. 9, no. 2, p. 7806700, 2022, [Online]. Available: <http://jurnal.mdp.ac.id>
- [4] A. Nuzul, N. Hidayah, and H. Supriyono, "Sistem Pemesanan Menu Makanan Dan Minuman Rumah Makan Berbasis Website."
- [5] Reni Nabila Suryati, Ketut Artaye, "Platform Perjanjian Kerja Sama Menggunakan Metode Quick Short Berbasis Web Pada Institut Informasika dan Bisnis Darmajaya" Vol.3, No.5, 2024.
- [6] Z. R. Saputri *et al.*, "Rancang Bangun Sistem Informasi Pemesanan Makanan Berbasis Web Pada Cafe Surabaya."

- [7] R. W. Arifin and D. Setiyadi, "Algoritma Metode Pengurutan Bubble Sort dan Quick Sort Dalam Bahasa Pemrograman C++," *Inf. Syst. Educ. Prof.*, vol. 4, no. 2, pp. 178–187, 2020.
- [8] E. Pujiatiningsih, B. Siswoyo, R. Haviani, J. Teknik Informatika, J. Dipati, and U. Bandung, "Analisis Perbandingan Algoritma Metode Pengurutan Quicksort, Metode Pengurutan Selectionsort Dan Metode Pengurutan Heapsort."
- [9] Rheza Rijaya, Muhammad Ezar Al Rivani, "Perbandingan Penempatan Pivot Pada Quick Sort Berdasarkan Ukuran Pemusatan Data" vol.4, no.2, 2023.
- [10] Al-Amansyah, "Implementasi Algoritma Quicksort Untuk Pembangkitan Kunci Algoritma Elgamal Pada Pengamanan Data File Dokumen," 2022, [Online]. Available: <https://journal.grahamitra.id/index.php/buai>