

IMPLEMENTASI SEDERHANA APLIKASI BUKU TELEPON OFFLINE DENGAN MENGGUNAKAN SELECTION SORT DI PYTHON

M Rosyid Fauzan, Feby Juliana Silalahi, Fauzan Azima Lubis, Fanny Ramadhani

Ilmu Komputer, Universitas Negeri Medan

Jalan Willem Iskandar, Pasar V Medan Estate, Percut Sei Tuan, Deli Serdang

rosyid.4233250009@mhs.unimed.ac.id

ABSTRAK

Di zaman digital sekarang, penggunaan buku telepon offline tetap memiliki peranan penting, terutama di daerah dengan akses internet yang terbatas. Namun, pengelolaan data dalam buku telepon seringkali menjadi tantangan, terutama saat jumlah entri meningkat. Penelitian ini memiliki tujuan untuk merancang dan mengimplementasikan aplikasi Buku Telepon yang dapat berfungsi secara offline dengan menggunakan bahasa pemrograman Python. Aplikasi kontak ini dirancang untuk menyimpan dan mengelola data pengguna secara efisien. Salah satu fitur utama yang dihadirkan dalam aplikasi ini ialah algoritma pengurutan Selection Sort, yang berfungsi untuk mengurutkan daftar kontak berdasarkan nama secara efektif. Metode pengembangan yang diadopsi adalah pendekatan berbasis objek, dengan memanfaatkan pustaka GUI Tkinter untuk menciptakan antarmuka pengguna. Aplikasi ini memberikan kemampuan kepada pengguna untuk menambah, mengedit, mengurutkan, mencari, dan menghapus data kontak, serta menampilkan daftar kontak yang telah diurutkan. Pengujian dilakukan untuk memastikan aplikasi berfungsi dengan baik dan efisiensi algoritma pengurutan yang diterapkan. Hasil pengujian menunjukkan bahwa aplikasi ini bekerja dengan baik dan dapat menyajikan data kontak secara cepat dan terorganisir. Dan pada penelitian ini, bisa diketahui bahwasannya metode selection sort bisa beroperasi sedikit lebih cepat dibandingkan dengan Insertion sort. Diharapkan aplikasi ini dapat memberikan kemudahan bagi pengguna dalam pengelolaan informasi kontak secara praktis tanpa memerlukan koneksi internet.

Kata kunci : Seleksi, Algoritma penyortiran, Buku telepon, Struktur data, Python

1. PENDAHULUAN

Bahkan di zaman digital ini, penggunaan buku telepon offline tetap memiliki peranan penting, terutama di daerah dengan akses internet yang terbatas. Namun, pengelolaan data dalam buku telepon seringkali menjadi tantangan, terutama saat jumlah entri meningkat. Salah satu masalah utama adalah efisiensi dalam proses penyortiran data. Oleh karena itu, sangatlah krusial untuk menerapkan algoritma pengurutan yang efektif guna mempercepat pengambilan dan pengelolaan data. Di antara berbagai jenis algoritma pengurutan, *selection sort* merupakan metode yang sederhana namun efektif untuk kumpulan data yang kecil. Meskipun begitu, masih sedikit penelitian yang mengeksplorasi efektivitas dan efisiensi penggunaan penyortiran selektif dalam konteks buku telepon offline. Temuan awal menunjukkan bahwa *Selection sort* memiliki beberapa kelemahan berkaitan dengan kompleksitas waktu jika dibandingkan dengan algoritma pengurutan lainnya.

Meskipun demikian, algoritma ini mungkin masih menjadi pilihan yang baik untuk aplikasi sederhana, seperti direktori telepon offline. Rencana pemecahan masalah ini mencakup analisis mendalam mengenai kinerja *selection sort* dalam mengurutkan data nama dan nomor telepon, serta perbandingannya dengan algoritma lain seperti Insertion sort. Hal ini dilakukan dengan menganalisis waktu eksekusi dan konsumsi sumber daya, yang diharapkan dapat membantu menemukan solusi yang efektif untuk pengelolaan buku telepon offline. Tujuan dari

percobaan ini adalah untuk menganalisis efektivitas metode selection sort dalam mengurutkan data yang terdapat pada buku telepon offline. Selanjutnya, hasil pengurutan data pada buku telepon offline dengan menggunakan metode selection sort tersebut akan dibandingkan dengan metode insertion sort, dengan tujuan untuk mengetahui apakah *selection sort* lebih cepat dibandingkan insertion sort atau sebaliknya. Selain itu, selection sort biasanya banyak digunakan untuk mengurutkan data yang berupa angka atau nilai tertentu. Namun, pada percobaan ini, data yang akan diurutkan terdiri atas kata-kata yang selanjutnya akan diurutkan berdasarkan urutan huruf A-Z.

2. TINJAUAN PUSTAKA

Selection sort adalah algoritma yang digunakan untuk mengurutkan data, baik dari yang terkecil ke yang terbesar (ascending) maupun dari yang terbesar ke yang terkecil (descending). Algoritma ini bekerja dengan menyisipkan data ke posisi yang tepat, dimulai saat menemukan data yang lebih kecil. Pencarian posisi yang tepat dilakukan dengan mencari secara berurutan dalam urutan elemen, dan selama pencarian tersebut, pergeseran elemen dapat terjadi [1].

Metode *selection sort* merupakan teknik pengurutan yang membandingkan elemen pertama dengan elemen berikutnya hingga elemen terakhir. Apabila ditemukan elemen lain yang lebih kecil dari elemen pertama, maka posisi elemen tersebut akan disimpan dan langsung ditukar. Proses perbandingan

terus berlanjut hingga tidak ada lagi pertukaran data[2].

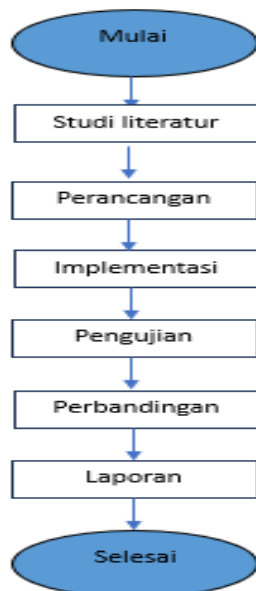
Cara kerja algoritma ini melibatkan teknik perulangan, di mana program secara berulang memilih bilangan dari blok pertama dan membandingkannya dengan bilangan yang ada di blok kedua, ketiga, dan seterusnya. Apabila ditemukan bilangan yang lebih kecil dari bilangan di blok pertama, maka posisi swap akan diganti[3].

Selection sort adalah algoritma pengurutan yang sederhana, yang melakukan beberapa lintasan untuk memilih item data. Algoritma ini dikenal karena kesederhanaannya dan seringkali menunjukkan kinerja yang lebih baik dibandingkan algoritma lain yang lebih kompleks dalam berbagai situasi [4].

Algoritma *selection sort* adalah suatu perbaikan dari algoritma *bubble sort* yang bertujuan untuk mengurangi jumlah perbandingan [5].

Dalam penelitian yang berjudul “Implementasi Algoritma Selection Sort dalam Membangun Aplikasi Android Pemesanan Jasa Make-up Palembang” hasil dari penelitian ini adalah terwujudnya sebuah aplikasi pemesanan jasa make-up yang dapat membantu pelanggan dalam memesan jasa MUA dan juga membantu MUA dalam menawarkan jasa makeup artist dan juga dapat diterapkannya algoritma *selection sort* dalam melakukan sorting pada pemesanan paket yang disediakan oleh MUA berdasarkan harga sehingga memudahkan pelanggan dalam memilih paket berdasarkan biaya yang dimiliki oleh pelanggan[6]

3. METODE PENELITIAN



Gambar 1. Diagram alir pengerjaan proyek

3.1. Metode Pengumpulan Data

Penulisan jurnal berjudul “Implementasi sederhana aplikasi Buku Telepon offline dengan menggunakan Selection Sort di Python” tentu memerlukan kajian yang mendalam terhadap materi

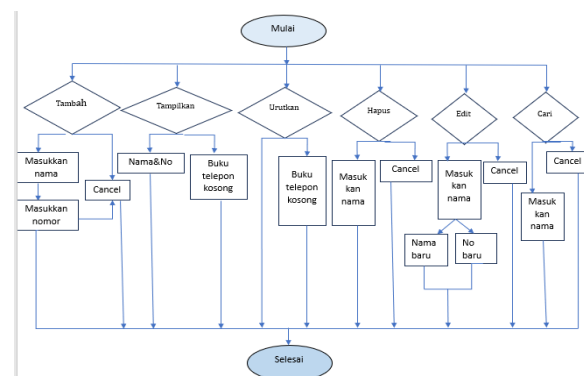
yang dimaksud. Ini dilakukan untuk memahami lebih lanjut tentang proyek yang akan kami kerjakan. Metode studi literatur adalah serangkaian kegiatan yang berkaitan dengan cara pengumpulan data pustaka, yang meliputi membaca dan mencatat, serta mengelola bahan penelitian. Metode studi literatur mencakup pengumpulan data dengan menggali informasi dari berbagai sumber, seperti buku, literatur, jurnal, artikel, dan laporan yang relevan dengan permasalahan yang ingin diatasi atau yang berkaitan dengan penelitian yang pernah dilakukan sebelumnya.

3.2. Perancangan Program

Dalam pembuatan jurnal ini, kami menggunakan bahasa pemrograman Python untuk menjalankan program pembuatan buku telepon offline. Algoritma yang diterapkan adalah algoritma pengurutan. Penyortiran membantu dalam mengatur item data, sehingga memudahkan situasi seperti ini, kami memilih untuk menggunakan *selection sort* sebagai metode penyortiran. Algoritma *selection sort* sering kali juga dikenal sebagai metode maksimum atau minimum. Disebut metode maksimum karena algoritma ini berfokus pada pemilihan data atau elemen maksimum sebagai dasar untuk melakukan pengurutan. Konsepnya meliputi pemilihan elemen terbesar dan menukarnya dengan elemen yang paling akhir untuk pengurutan ascending, serta dengan elemen pertama untuk pengurutan descending.

Berdasarkan penjelasan tersebut, penulis merangkum tahapan dalam algoritma *selection sort* sebagai berikut:

- Langkah pertama yang perlu dilakukan adalah mencari nilai terkecil atau minimum dalam struktur data jika diurutkan secara ascending, sementara jika descending, carilah nilai yang maksimum.
- Setelah itu, tukar nilai tersebut dengan nilai yang berada di posisi pertama dalam struktur data.
- Selanjutnya, ulangi langkah-langkah di atas untuk bagian struktur data yang tersisa hingga data tersebut tersusun dengan rapi dalam urutan ascending atau descending[7][8][9]



Gambar 2. Flowchart program aplikasi buku telepon

Gambar 2 merupakan gambaran alur program yang akan dibuat. Jadi, pengguna akan diminta untuk memilih beberapa opsi yang tersedia. Adapun alur program tersebut akan diterapkan pada Visual Studio Code dengan menggunakan Python dan Tkinter sebagai GUI.

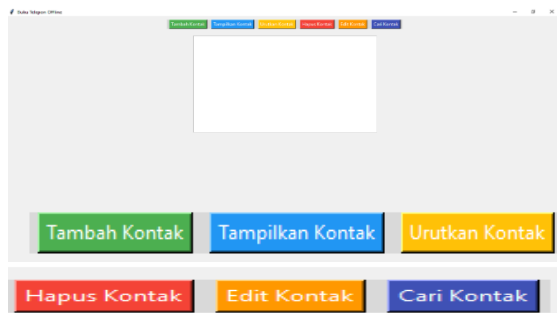
3.3. Perbandingan dan Laporan

Setelah program selesai dijalankan, langkah berikutnya adalah membandingkan hasilnya dengan algoritma lainnya. Tujuan dari perbandingan ini adalah untuk mengidentifikasi algoritma mana yang lebih efektif. Algoritma pengurutan yang dipilih untuk perbandingan adalah *Insertion Sort*. *Insertion Sort* sendiri merupakan algoritma pengurutan yang sederhana, yang prinsip kerjanya mirip dengan cara mengurutkan kartu remi. Array dibagi menjadi dua bagian, yaitu bagian yang teratur dan bagian yang belum teratur. Kemudian, nilai yang belum diurutkan akan dipilih dan ditempatkan pada posisi yang sesuai dalam urutan yang benar. Ada beberapa alasan di balik pemilihan *Insertion Sort* sebagai algoritma untuk perbandingan, antara lain: kesederhanaan, kompleksitas waktu yang setara, penggunaan ruang memori, performa pada dataset kecil, stabilitas, dan kesesuaian untuk input tertentu.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Source Code pada program dan hasilnya

Berikut adalah hasil dari implementasi source code beserta output program yang telah dibuat dengan menggunakan Tkinter sebagai antarmuka GUI dan Python sebagai bahasa pemrograman.



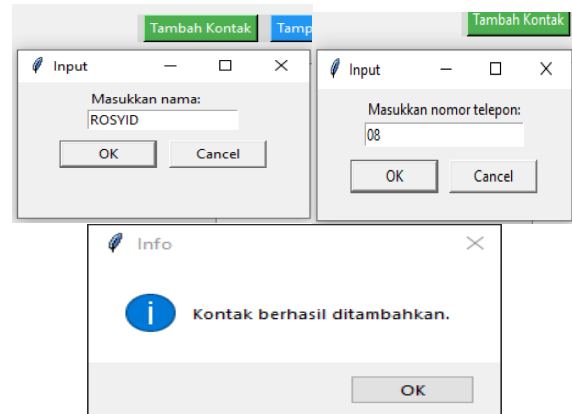
Gambar 3. Tampilan awal program

Ketika program dijalankan, akan muncul tampilan seperti yang ditunjukkan pada Gambar 3. Di tampilan tersebut, terdapat beberapa opsi menu yang dapat kita pilih antara lain: menambah kontak, menampilkan kontak, mengurutkan kontak, menghapus kontak, mengedit kontak, dan mencari kontak.

4.2. Tambah Kontak

Pada tombol untuk menambah kontak, akan ada perintah yang akan menjalankan fungsi untuk menambahkan kontak baru. Saat tombol tersebut ditekan, maka akan menghasilkan output yang

meminta kita untuk memasukkan nama dan nomor telepon ke dalam program. Gambar 4 menampilkan output dari perintah untuk nama dan nomor telepon. Setelah pengguna memasukkan nama dan nomor telepon, serta menekan tombol OK, program akan menampilkan pesan “kontak berhasil ditambahkan”, seperti yang terlihat pada gambar 4.



Gambar 4. Tambah kontak

4.3. Tampilkan kontak

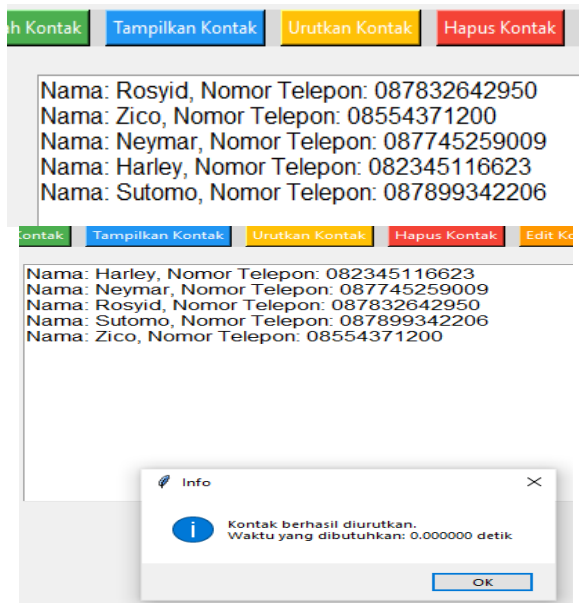
Pada tombol tampilkan kontak, program akan menampilkan kontak yang telah kita tambahkan sebelumnya ke layar. Jika kita belum menambahkan satu kontak pun, maka tampilan kontak tidak akan menunjukkan apapun kecuali pesan bahwa buku telepon kosong.



Gambar 5. Tampilkan kontak

4.4. Urutkan Kontak

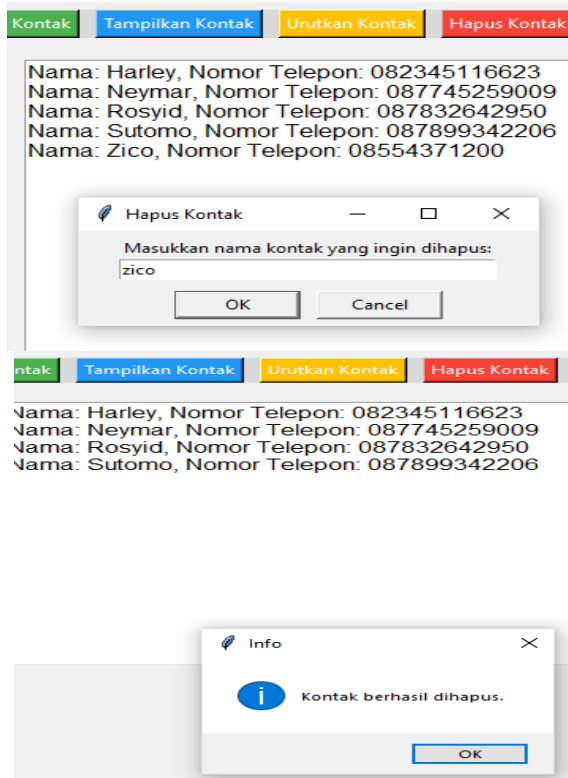
Tombol untuk mengurutkan kontak merupakan poin utama dalam judul jurnal ini. Fungsi ini akan mengatur daftar nama kontak yang masih acak menggunakan metode *Selection Sort*. Daftar nama tersebut akan diurutkan secara ascending, yaitu dari yang terkecil, dengan berdasarkan pada huruf awal masing-masing nama.



Gambar 6. Urutkan kontak

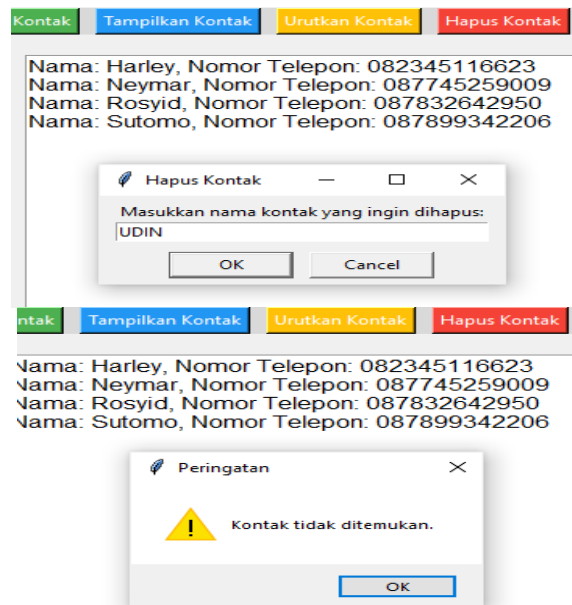
4.5. Hapus Kontak

Hapus kontak akan menghilangkan kontak yang diinginkan yang telah ditambahkan ke dalam program. Program akan meminta pengguna untuk mengisi nama kontak yang ingin dihapus, kemudian program akan mencari kontak yang telah diketik. Jika kontak tersebut ada, program akan menghapusnya dan memberikan pesan bahwa kontak telah berhasil dihapus. Namun, jika kontak tidak ditemukan, program akan menampilkan pesan “kontak tidak ditemukan”.



Gambar 7. Hapus kontak

Terlihat pada gambar 7 bahwa ketika kontak berhasil dihapus, kontak tersebut secara otomatis tidak akan muncul lagi dalam tampilan maupun dalam program.

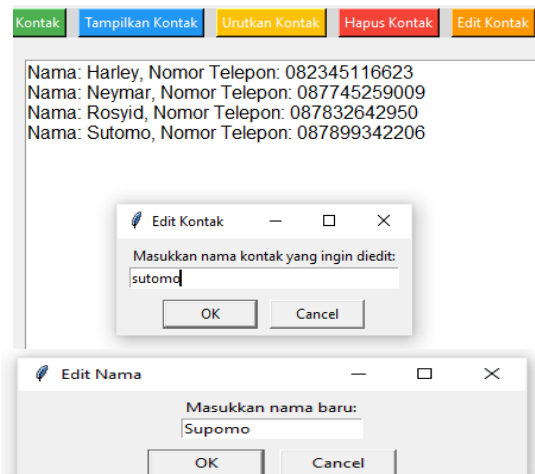


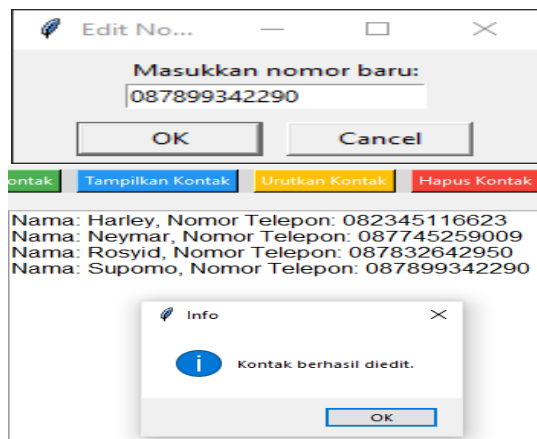
Gambar 8. Ketika menghapus kontak yang belum ada

Pada gambar 8, dapat dilihat bahwa ketika pengguna mencari nama Udin, yang muncul hanya pesan bahwa kontak tidak ditemukan. Situasi ini terjadi karena nama Udin belum ditambahkan ke dalam program. Oleh karena itu, tindakan menghapus nama yang tidak ada hanya akan menghasilkan pesan dari program.

4.6. Edit Kontak

Dalam alat pengeditan kontak, pengguna akan diminta untuk memasukkan nama kontak yang ingin diubah. Yang akan diubah oleh program adalah nama dan nomor telepon. Namun, jika nama yang dicari tidak terdapat dalam daftar kontak, opsi pengeditan hanya akan menampilkan pesan bahwa kontak tersebut tidak ditemukan. Hal ini dapat dilihat pada gambar 8.

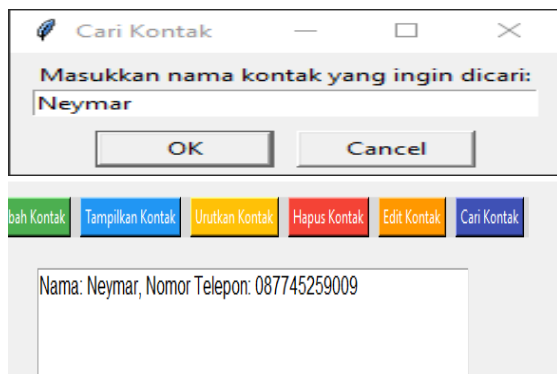




Gambar 9. Sebelum dan sesudah edit kontak

4.7. Cari Kontak

Opsi terakhir yang tersedia adalah tombol cari kontak. Fungsi dari tombol cari kontak adalah untuk mencari nama kontak. Tombol ini sangat efektif digunakan ketika jumlah kontak sudah banyak, sehingga sulit untuk mencarinya satu per satu. Seperti halnya tombol hapus dan edit, tombol cari ini juga akan menampilkan pesan "kontak tidak ditemukan" jika kontak yang dicari tidak terdapat dalam daftar. Hal ini dapat dilihat pada gambar 8



Gambar 10. Cari kontak

Dapat kita perhatikan pada gambar 10, ketika nama yang dicari terdapat dalam daftar kontak, nama tersebut akan secara otomatis ditampilkan pada layar.

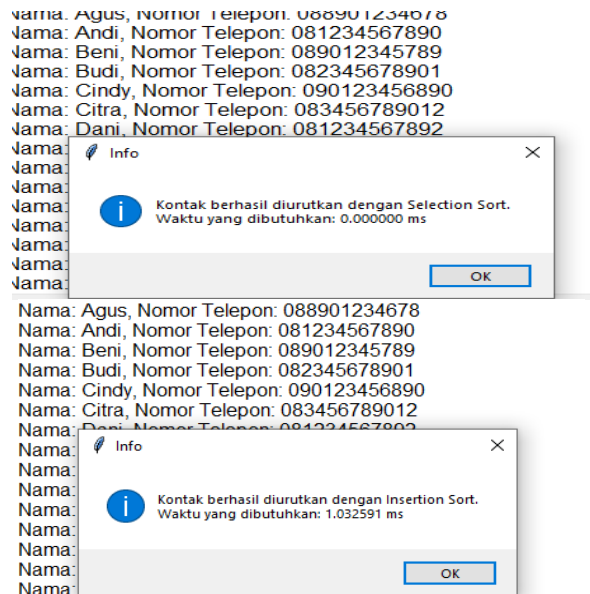
4.8. Perbandingan Selection Sort dengan Insertion Sort

Pada tahap ini, akan dijelaskan mengenai perbandingan performa antara Selection sort dan Insertion sort dalam hal kecepatan. Dalam program yang telah dibuat sebelumnya, kami menambahkan fungsi Insertion sort dengan kode program;

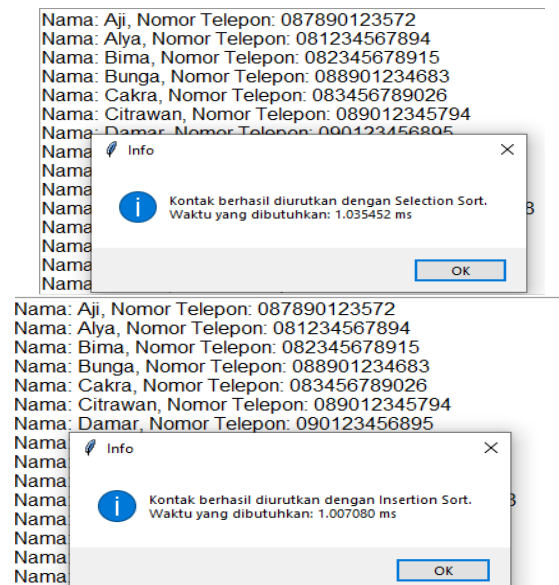
```
# Fungsi untuk melakukan insertion sort
def insertion_sort(contacts):
    for i in range(1, len(contacts)):
        key = contacts[i]
        j = i - 1
        while j >= 0 and key[0] < contacts[j][0]: #
            Mengurutkan berdasarkan nama
            contacts[j + 1] = contacts[j]
```

```
j -= 1
contacts[j + 1] = key
return contacts
```

Kode program ini akan mengurutkan daftar nama menggunakan metode Insertion sort.

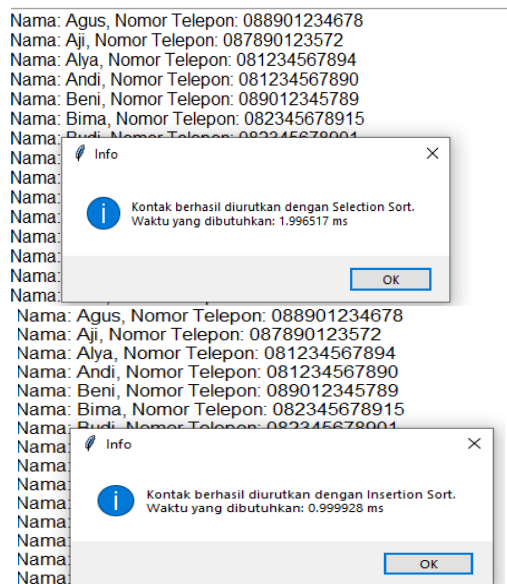


Gambar 11. Ujicoba 1

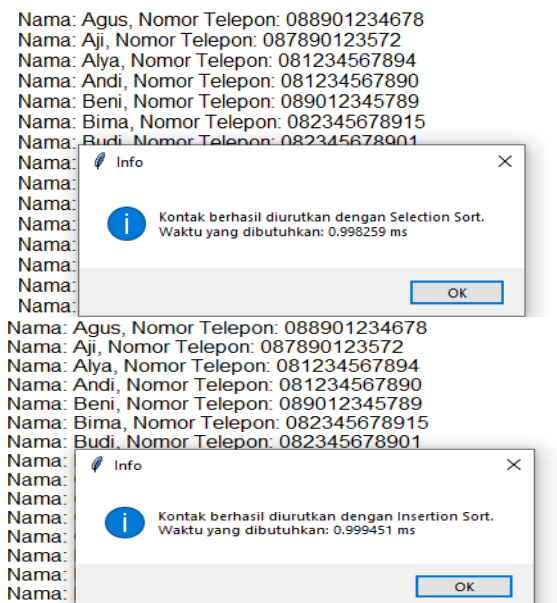


Gambar 12. Ujicoba 2

Kita dapat melihat dengan jelas adanya perbedaan kecepatan antara metode selection sort dan insertion sort. Pada percobaan 1 dan 2, kami menggunakan data kecil yang terdiri dari 50 daftar nama dan nomor telepon yang masih acak. Selanjutnya, 50 data tersebut akan diurutkan menggunakan selection sort dan insertion sort.



Gambar 13. Ujicoba 3



Gambar 14. Ujicoba 4

Uji coba ketiga dan keempat dilakukan dengan menambahkan jumlah data. Jumlah data pada uji coba ketiga telah ditingkatkan menjadi 100 data. Tentunya, kami memperoleh hasil yang berbeda dibandingkan sebelumnya.

4.9. Laporan

Hasil penelitian yang disusun oleh Putri Ghina dalam artikel yang berjudul “Implementasi Stack dan Array pada Pengurutan Lagu dengan Metode Selection Sort” menunjukkan bahwa algoritma selection sort memiliki kecepatan sedikit lebih tinggi dibandingkan dengan algoritma insertion sort. Penelitian ini dilakukan melalui lima percobaan menggunakan 500 data lagu yang diurutkan.[1]. Dalam penelitian berjudul “Implementasi Algoritma Selection Sort Untuk Pengurutan Data Mahasiswa Di Program Studi Teknik Informatika Politeknik Negeri Pontianak”

hasilnya adalah berdasarkan NIM selama pengujian algoritma selection sort, ditemukan rata-rata waktu pengurutan yang dilakukan sebanyak 5 kali yaitu 0,5208 detik. Dan juga Algoritma selection sort menggunakan rata-rata waktu selama 0,5658 detik untuk pengurutan berdasarkan nama yang dilakukan sebanyak 5 kali.[10]

Tabel 1. Perbandingan dua sorting

Percobaan ke-	Selection sort (ms)	Insertion sort (ms)
1	0.000000	1.032591
2	1.035452	1.007080
3	1.996517	0.999928
4	0.998259	0.999451
Rata-rata	1.00756	1.00976

Dari tabel 1, diperoleh hasil bahwa algoritma selection sort sedikit lebih cepat daripada algoritma insertion sort. Algoritma selection sort mengurutkan data dengan rata-rata waktu 0.0022 ms lebih cepat dibandingkan insertion sort. Meskipun dalam beberapa percobaan insertion sort dapat menunjukkan kecepatan yang lebih baik dibandingkan selection sort, secara keseluruhan, total percobaan menunjukkan bahwa selection sort masih lebih efektif.

5. KESIMPULAN DAN SARAN

Simpulan yang diperoleh dari percobaan ini adalah bahwa selection sort juga dapat mengurutkan data bertipe string (kata/huruf). Hasil dari program yang telah dibuat menunjukkan bahwa Selection Sort dapat digunakan untuk menyusun buku kontak secara teratur. Melalui penerapan Selection Sort, aplikasi ini memperlihatkan cara yang efektif untuk mengatur data berdasarkan nama kontak. Meskipun Selection Sort bukanlah algoritma pengurutan yang paling efisien untuk set data yang besar, penerapannya dalam konteks aplikasi ini memberikan pemahaman yang lebih mendalam tentang dasar-dasar algoritma pengurutan dan cara kerjanya dalam pengelolaan data. Aplikasi Buku Telepon ini juga menawarkan antarmuka pengguna yang sederhana dan intuitif, yang memungkinkan pengguna untuk menambahkan, menampilkan, menghapus, mengedit, mencari, dan mengurutkan kontak dengan mudah.

Dengan demikian, aplikasi ini berfungsi tidak hanya sebagai alat manajemen kontak tetapi juga sebagai sarana edukasi bagi pemrogram pemula untuk memahami konsep dasar pemrograman dan algoritma. Secara keseluruhan, proyek ini berhasil mencapai tujuan awalnya dan dapat menjadi landasan bagi pengembangan aplikasi yang lebih kompleks di masa depan, termasuk penambahan fitur-fitur seperti pencarian kontak, penghapusan kontak, dan penyimpanan data yang lebih permanen.

DAFTAR PUSTAKA

- [1] G. M. Putri, K. A. Di Pradja, M. B. M. Azizi, P. Nurwahid, A. S. Perdana, and . M., “Implementasi Stack dan Array pada

- Pengurutan Lagu dengan Metode Selection Sort,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 6, no. 2, pp. 286–296, Apr. 2024, doi: 10.47233/jteksis.v6i2.1192.
- [2] 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 Jurnal Ilmu Komputer*, vol. 1, no. 4, pp. 190–194, Dec. 2022, doi: 10.56211/helloworld.v1i4.187.
- [3] “Implementation of the Selection Sort Algorithm to Sort Data in PHP Programming Language,” *Journal of Computer Science, Information Technology and Telecommunication Engineering*, vol. 4, no. 1, Mar. 2023, doi: 10.30596/jcositte.v4i1.14362.
- [4] C. Yovita, K. Nasution, and T. Haramaini, “Penerapan Algoritma Naive Bayes dan Selection Sort pada Penilaian Kuis di Aplikasi Pembelajaran Pemrograman Java dan PHP,” *Hello World Jurnal Ilmu Komputer*, vol. 2, no. 3, pp. 120–128, Aug. 2023, doi: 10.56211/helloworld.v2i3.278.
- [5] Y. Wisda Tumarta Arif, F. Ilmu Kesehatan, and U. Duta Bangsa Surakarta, “IMPLEMENTASI METODE SELECTION SORT PADA SISTEM INFORMASI RETENSI DOKUMEN REKAM MEDIS DI KLINIK PKU MUHAMMADIYAH KARANGANOM, KLATEN,” vol. 15, no. 1, pp. 108–117, 2022, [Online]. Available: <http://journal.stekom.ac.id/index.php/elkom/page108>
- [6] F. Harahap, I. Effendy, and J. A. Yani No, “Implementasi Algoritma Selection Sort.”
- [7] Zailani Udin Achmad, Apriyanto Budi, and Zakaria Hadi, *Struktur Data*, 1st ed. Banten: Unpam Press, 2020.
- [8] A. Sukanto, S. Cahyadi, B. Priambodo, Y. Jumaryadi, and D. M. Nashar, *Ade Mulyana Cara Mudah Mempelajari Algoritma dan Struktur Data*. [Online]. Available: www.blogdivapress.com
- [9] A. Udin Zailani Budi Apriyanto Hadi Zakaria Jl Surya Kencana No, P. A. Gd, and U. Pamulang Tangerang Selatan -Banten, *STRUKTUR DATA*. [Online]. Available: www.unpam.ac.id
- [10] F. Pratama, D. Juniardi, and M. Fauzan Arif, “Prosiding Seminar Nasional Sains dan Teknologi Seri 02 Fakultas Sains dan Teknologi,” 2024. [Online]. Available: <https://wawanlaksito.wordpress.com/>