# APLIKASI NAVIGASI PERJALANAN PAKET DENGAN SISTEM PEMILIHAN RUTE TERCEPAT MENGGUNAKAN ALGORITMA GREEDY

## *PACKAGE TRAVEL NAVIGATION APP WITH THE FASTEST ROUTE SELECTION SYSTEM USING THE GREEDY ALGORITHM*

**Djorgy[1], Anton Siswo Raharjo Ansori,[2] M Husni Syahbani,[3]**

[1,2,3] Universitas Telkom, Bandung
**[1]djorgydjo@student.telkomuniversity.ac.id, [2]raharjo@telkomuniversity.ac.id, [3]eksmud@telkomuniversity.ac.id**

**Abstrak**

Pengantaran paket merupakan kegiatan pengantaran barang secara langsung dari suatu ekspedisi ke konsumen yang dilakukan oleh seorang kurir, seorang kurir tidak hanya mengantarkan satu paket tetapi menghantarkan sebanyak yang kurir mampu dan tentunya tidak mungkin sedikit. Karena kurir banyak membawa paket dengan demikian kurir kesulitan dalam menimalisir waktu pengantaran karena alamat paket tertera harus dicarinya satu persatu.

Dengan adanya permasalahan itu dirancanglah suatu aplikasi yang dapat mengatur rute perjalanan paket dari yang terdekat hingga yang terjauh. Pengurutan alamat dari aplikasi ini berdasarkan info dari QR code yang discan oleh kurir lalu sistem pemilih rute akan mensortir alamat dari yang terdekat hingga terjauh dengan menggunakan Algoritma Greedy. Keluaran dari aplikasi ini adalah berupa urutan alamat paket yang akan diantar dari rute yang terdekat hingga terjauh. Dengan demikian kurir tidak kesulitan dalam mencari alamat dan mengantarkan paket dengan waktu yang cepat.

Berdasarkan pengujian yang dilakukan pencarian jarak terpendek dari suatu titik sampai ke titik tujuan, pada penelitian ini algoritma greedy berhasil menentukan jarak terpendek. Hasil pengujian program menghasilkan rute dari titik awal 1 ke 100 titik tujuan yang ada dan waktu komputasi dari program dengan waktu yang terbilang cepat dengan rata-rata 0,122 detik

Kata Kunci : kurir, Algoritma *Greedy*, Navigasi perjalanan paket, *Traveling Salesman Problem (TSP)*

**Abstract**

*Package delivery is an activity of delivering goods directly from an expedition to a consumer carried out by a courier, a courier not only delivers one package but delivers as much as the courier can afford and of course not a little. Because the courier carries a lot of packages, the courier has difficulty in minimizing the delivery time because the package address listed must be searched for one by one.With this problem, an application is designed that can manage the route of the package from the closest to the furthest. The sorting of addresses from this application is based on info from the QR code scanned by the courier then the route selector system will sort the addresses from closest to furthest using the Greedy Algorithm. The output of this application is in the form of a sequence of packet addresses that will be delivered from the closest to the furthest route. Thus the courier has no difficulty in finding an address and delivering the package in a fast time.Based on the tests carried out the shortest distance from a point to the destination point, in this study the greedy algorithm succeeded in determining the shortest distance. The test results of the program produce a route from the starting point of 1 to 100 existing destination points and the computation time of the program with a fairly fast with avarage time 0,122 seconds.*

*Keywords: Courier, Greedy algorithm, package navigation, Traveling Salesman Problem (TSP)*

## 1. Pendahuluan

Kurir adalah sebuah aktivitas pengiriman barang yang dilakukan secara langsung. Sebenarnya pekerjaan menjadi kurir sudah ada sejak jaman dahulu kala. Dimana seseorang akan berlari untuk menyampaikan pesan atau barang. Dari situlah awalnya muncul kata courier dalam bahasa Inggris yang berasal dari bahasa Yunani curere yang berarti lari. Masyarakat Indonesia sendiri lebih mengenal kurir sebagai sebuah pekerjaan untuk mengantarkan barang. Biasanya kurir akan mengendarai sepeda atau motor agar lebih cepat dalam proses pengirimannya. Walaupun sebenarnya ada juga kurir yang menggunakan mobil, namun masyarakat Indonesia belum terlalu familiar. Menjadi kurir atau pengantar barang pastinya memiliki beberapa kesulitan yang dialaminya, entah itu masalah pada pemilihan rute sehingga membuat pengiriman menjadi terlambat, kondisi cuaca yang menghambat pengantaran paket, hingga keadaan lalu lintas di kota-kota besar yang selalu padat. Hal-hal ini membuat pengantaran barang yang harusnya sudah ditentukan akan sampai pada hari yang ditentukan bisa terkena delay atau pengunduran waktu antar. Tetapi kurir harus tetap profesional dalam mengerjakan tugasnya dan disiplin waktu, untuk memudahkan pengantaran paket dan mempergunakan waktu dengan baik seorang kurir harus tau jalur atau pemilihan rute yang tepat untuk paket-paket yang akan dikirimnya agar barang sampai dengan tepat waktu. Pemilihan rute yang tidak karuan membuat pengiriman menjadi lama, dan membuat kurir itu sendiri kesulitan dalam mencari alamat pengirimiman yang tidak terurut sehingga membuat kurir itu sendiri harus mengecek satu persatu alamat dari paket tersebut. Dengan permasalahan yang telah dijelaskan sebelumnya salah satu cara untuk mengatasi permasalah diatas adalah dengan membuat aplikasi berbasis android dengan fitur yang memberikan kemudahan

dalam memilih rute yang efisien kepada kurir atau pengantar barang ke konsumen yang akan dibuat dengan menggunakan algoritma greedy sebagai penentu titik-titik rute perjalanan yang harus ditempuh kurir secara terurut dari rute yang terdekat dengan lokasi kurir hingga rute terjauh.

## 2. Kajian Pustaka

### 1. Algoritma *greedy*

Algoritma adalah Langkah dalam mencari solusi atas sebuah masalah dalam membangun sebuah program, salah satunya adalah algoritma greedy. Algoritma greedy merupakan metode yang paling popular untuk memecahkan persoalan optimasi. Algoritma greedy hanya mengambil pilihan yang terbaik yang dapat diperoleh saat itu dan berharap bahwa dengan memilih optimum lokal pada setiap langkah mencapai optimum global.[1]

### 2. Graf

Dalam bidang matematika dan ilmu komputer, teori graf mempelajari tentang graf yaitu struktur yang menggambarkan relasi antar objek dari sebuah koleksi objek. Biasanya graf digambarkan sebagai kumpulan titik-titik yang dihubungkan oleh garis. Banyak sekali struktur yang dapat direpresentasikan dengan graf, dan banyak masalah yang bisa diselesaikan dengan graf. Simpul pada graf dapat dinomori dengan huruf alphabet, dengan bilangan, atau gabungan dari keduanya. Secara geometri graf digambarkan sebagai sekumpulan noktah didalam bidang dwimitra yang dihubungkan dengan sekumpulan garis[2]

### 3. Lintasan Terpendek (*Shortest Path*)

Persoalan mencari lintasan terpendek di dalam graf merupakan salah satu persoalan optimasi. graf yang digunakan dalam pencarian lintasan terpendek adalah graf berbobot (weighted graph), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya.[3].

### 4. *K-Nearest Neighbour*

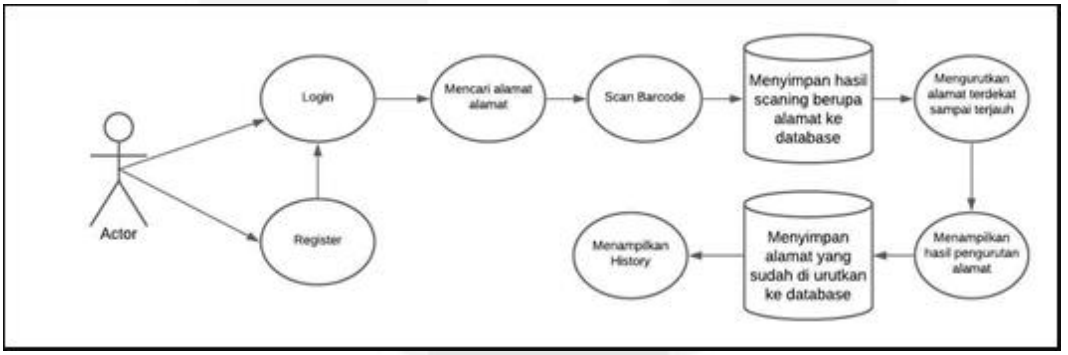
Prinsip kerja dari klasifikasi K-Nearest Neighbor (KNN) adalah mencari jarak terdekat antara data yang dievaluasi dengan K tetangga (neighbor) terdekatnya yang ada pada data latih. Serupa dengan teknik *clustering*, mengelompokkan data baru berdasarkan jarak data baru ke beberapa tetangga terdekat (tetangga) [7].

### 5. *Traveling Salesman Problem (TSP)*

Metode Traveling Salesman Problem adalah sebuah metode yang digunakan untuk meminimasi biaya distribusi dengan cara mencari jarak dan rute terdekat, waktu tercepat dan biaya yang minimal. Traveling Salesman Problem dikenal sebagai suatu permasalahan optimasi yang bersifat klasik dan Non Deterministic Polynimial-time Complete(NPC), dimana tidak ada penyelesaian yang paling optimal selain mencoba seluruh kemungkinan penyelesaian yang ada. Permasalahan ini melibatkan seorang traveling salesman yang harus melakukan kunjungan sekali pada semua kota dalam sebuah lintasan sebelum dia kembali ke titik awal, sehingga perjalannya dikatakan sempurna. Permasalahan rute terpendek adalah permasalahan untuk menemukan rute terpendek antara titik asal (initial node) menuju titik tujuan (final node) dalam suatu jaringan jalan. Permasalahan travelling salesman problem adalah model permasalahan yang bertujuan untuk menemukan rute terpendek bagi seorang penjual keliling untuk berkeliling (touring) mengunjungi setiap pelanggannya sebanyak satu kali.[8]

## 3. Implementasi Pengujian

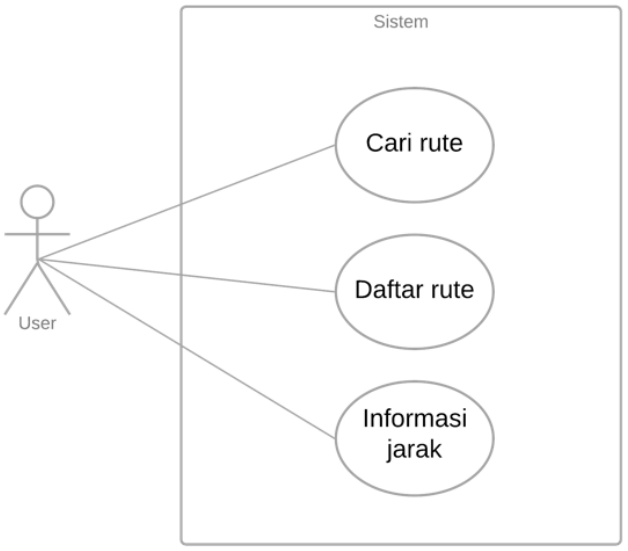
### 3.1 Gambaran Umum Sistem




Gambar 3. 1 gambaran umum sistem

Gambar diatas merupakan gambaran umum sistem dimana *user* akan ditunjukan dua pilihan yaitu *login* dan *register*. Jika *user* tidak memiliki akun maka *user* dapat melakukan *register* akun. Ketika sudah memiliki akun *user* akan ditampilkan halaman awal dimana terdapat menu mencari alamat dan *scan barcode/qr code*. Ketika *user* melakukan *scan* maka akan terjadi proses *querry* pada *database* dimana data akan tersimpan. Data ini berisikan informasi *lat* dan *lon*. Data yang sudah terbaca oleh *database* akan ditampilkan pada menu halaman *history*. *User* dapat memasukan alamat lain, ketika memasukan alamat sistem akan langsung melakukan *sorting* alamat dari jarak yang terdekat .

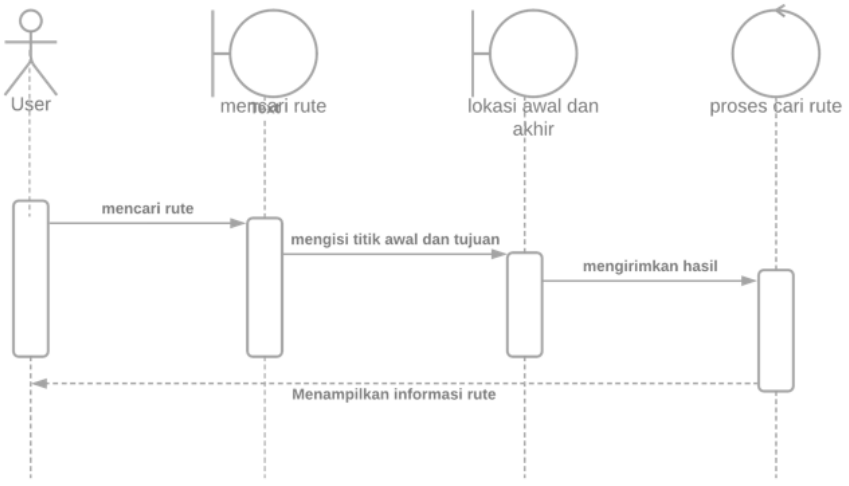
### *3.2 Usecase Diagram*




Gambar 3. 2 *Usecase Diagram*

Pada usecase diagram terdapat 3 aksi yaitu cari rute,dafar rute, dan informasi jarak. Pada aksi cari rute user akan menginputkan titik awal dan titik tujuan yang nantinya informasi tersebut akan dimasukan kedalam database. Aksi daftar rute user akan mendapatkan informasi daftar urutan lokasi awal sampai akhir, dimana user akan diarahkan untuk melalui rute yang sesuai dengan yang sistem berikan. Aksi informasi jarak user akan menerima informasi berupa jarak dalam Km antar titik titik lokasi .

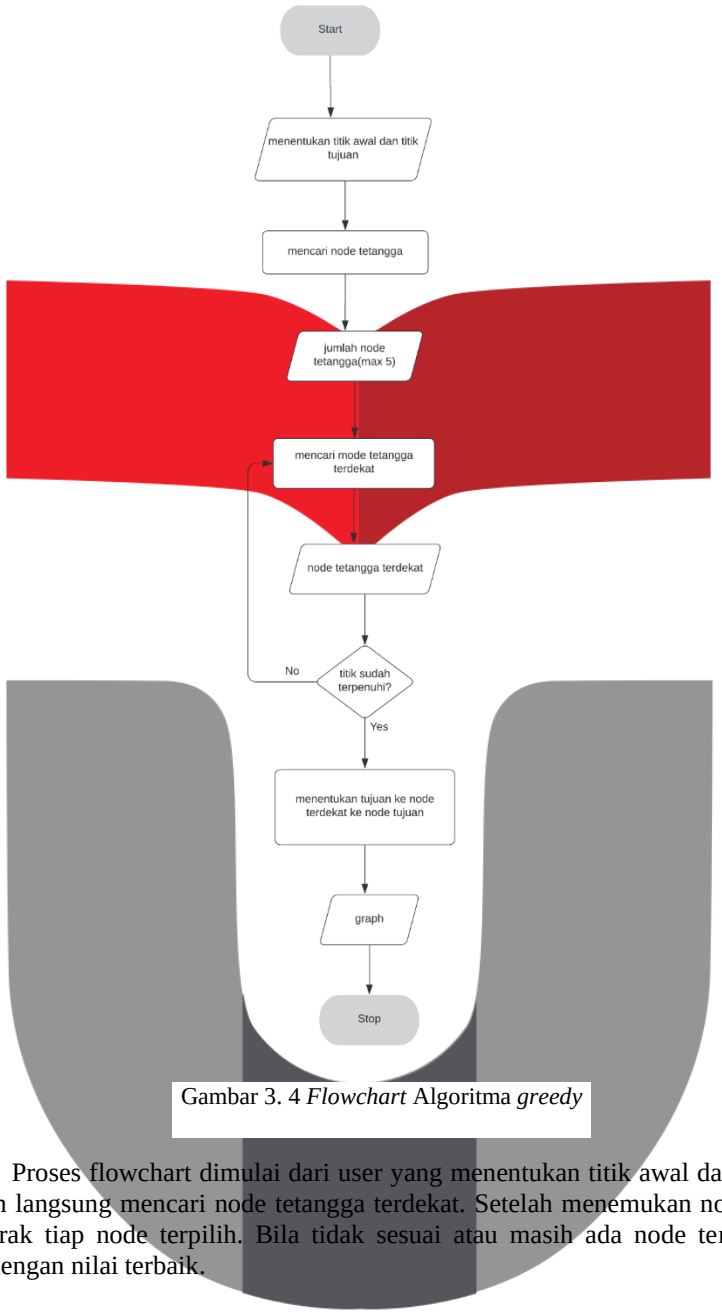
### *3.3 Sequence Diagram*




Gambar 3. 3 *Sequence Diagram*

Pada saat membuka aplikasi makan user akan ditampilkan menu halaman, user dapat memilih menu cari rute. Pada menu cari rute terdapat form yang harus user isikan yaitu berupa titik awal dan titik tujuan. Ketika user mengisi form titik awal dan tujuan sistem akan mengirimkan hasil tersebut dan melakukan proses pencarian rute dan menampilkan informasi rute kepada user.

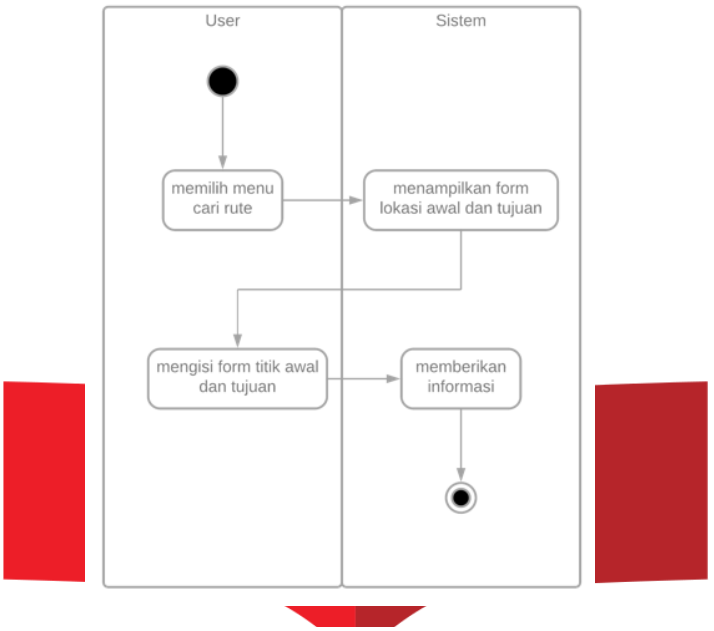
### *3.4 Flowchart* Algoritma *Greedy*




Gambar 3. 4 *Flowchart* Algoritma *greedy*

Proses flowchart dimulai dari user yang menentukan titik awal dan titik tujuan terlebih dahulu, lalu algoritma akan langsung mencari node tetangga terdekat. Setelah menemukan node terdekat makan algoritma akan mengecek jarak tiap node terpilih. Bila tidak sesuai atau masih ada node terkecil makan algoritma akan menganttikannya dengan nilai terbaik.

### *3.5 State Diagram*




Gambar 3. 5 *State Diagram*

Gambar diatas merupakan state diagram untuk bagian login page dimana user dapat memasukkan informasi berupa e-mail dan password. Pada saat user melakukan login maka akan terdapat dua kemungkinan yaitu login berhasil dan login gagal. Login akan berhasil bila user memasukkan data e-mail dan password sesuai dengan data yang terdapat pada database sistem. Login gagal disebabkan bisa karena user salah memasukkan data, bisa karena e-mail salah atau password yang salah. Gagal login bisa disebabkan juga user belum memiliki akun.

## 4. Implementasi dan Pengujian

### 4.1 Pengujian *alpha*

| No. | Fitur | Fungsi | Pengujian | Sesuai/Tidak Sesuai |
|---|---|---|---|---|
| 1 | Cari rute | Menentukan lokasi awal dan lokasi tujuan | *Blackbox* | Sesuai |
| 2 | Daftar rute | Program menampilkan urutan *node* | *Blackbox* | Sesuai |
| 3 | Informasi jarak | Program menampilkan informasi jarak antar *node* | *Blackbox* | Sesuai |

Kesimpulan nya dari tabel diatas bahwa dari 3 poin skenario pengujian, 3 pengujian program dapat berjalan lancar dan berhasil dilakukan. Untuk dapat mengetahui akurasinya akan digunakan rumus sebagai berikut:

$$Akurasi = \frac{total\ kesesuaian}{jumlah\ skenario\ pengujian} \times 100\%$$

$$Akurasi = \frac{3}{3} \times 100\%$$

$$Akurasi = 100\%$$

Sehingga akurasi dari pengujian alpha adalah 100% artinya seluruh fitur dari program berjalan semua.

### 4.2 Pengujian Algoritma

Pengujian yang dilakukan adalah mengamati hasil rute pecanrian yang dipilih oleh sistem dan waktu komputasi yang dilakukan untuk melakukan pencarian rute. Sistem melakukan pencarian dari titik 1 ke semua titik yang tersedia.

```
From 1 to 2
PathInfo(nodes=[1, 30, 33, 5, 95, 56, 7, 74, 8, 57, 51, 46, 66, 24, 2], edges=[(2.945, 8), (2.451, 1), (2.414, 6
From 1 to 3
PathInfo(nodes=[1, 30, 33, 5, 95, 56, 7, 74, 65, 3], edges=[(2.945, 8), (2.451, 1), (2.414, 6), (2.122, 5), (1.3
From 1 to 4
PathInfo(nodes=[1, 30, 33, 5, 95, 56, 7, 63, 97, 62, 50, 73, 4], edges=[(2.945, 8), (2.451, 1), (2.414, 6), (2.1
From 1 to 5
PathInfo(nodes=[1, 30, 33, 5], edges=[(2.945, 8), (2.451, 1), (2.414, 6)], costs=[2.716468590831919, 0.40799673
From 1 to 6
From 1 to 7
PathInfo(nodes=[1, 30, 33, 5, 95, 56, 7], edges=[(2.945, 8), (2.451, 1), (2.414, 6), (2.122, 5), (1.388, 4), (2.
From 1 to 8
PathInfo(nodes=[1, 30, 33, 5, 95, 56, 7, 74, 8], edges=[(2.945, 8), (2.451, 1), (2.414, 6), (2.122, 5), (1.388,
From 1 to 9
PathInfo(nodes=[1, 30, 33, 5, 95, 56, 7, 74, 8, 57, 51, 42, 23, 9], edges=[(2.945, 8), (2.451, 1), (2.414, 6), (
From 1 to 10
PathInfo(nodes=[1, 30, 33, 5, 95, 56, 10], edges=[(2.945, 8), (2.451, 1), (2.414, 6), (2.122, 5), (1.388, 4), (2
```

Dari gambar diatas sistem menunjukkan arah tujuan untuk sampai ke titik terakhir dan menampilkan jarak antar node dan waktu estimasi.

Berikut bentuk hasil pengujian dalam bentuk tabel:

| No. | Titik Awal | Titik Tujuan | Node yang dilewati |
|---|---|---|---|
| 1 | 1 | 2 | 1-30-33-5-95-56-7-74-8-57-51-46-66-24-2 |
| 2 | 1 | 3 | 1-30-33-5-95-56-7-74-65-3 |
| 3 | 1 | 4 | 1-30-33-5-95-56-7-63-97-62-50-73-4 |
| 4 | 1 | 5 | 1-30-33-5 |
| 5 | 1 | 6 | 1-34-40-6 |
| 6 | 1 | 7 | 1-30-33-5-95-56-7 |
| 7 | 1 | 8 | 1-30-33-5-95-56-7-74-8 |
| 8 | 1 | 9 | 1-30-33-5-95-56-7-74-8-57-51-42-23-9 |
| 9 | 1 | 10 | 1-30-33-5-95-56-10 |
| 10 | 1 | 11 | 1-30-33-5-95-56-7-74-8-57-51-46-66-24-69-11 |

Pada tabel diatas berisikan informasi titik awal yaitu 1 dan titik tujuan ke semua data lain, dan menghasilkan rute node apa saja yang dikunjungi untuk sampai ke tujuan.

Untuk pengujian waktu komputasi dari tiap titik yang diuji pada tabel diataas adalah sebagai berikut:

| No. | Titik Awal | Titik Tujuan | Waktu Komputasi |
|---|---|---|---|
| 1 | 1 | 2 | 0,4s |
| 2 | 1 | 3 | 0,1s |
| 3 | 1 | 4 | 0,2s |
| 4 | 1 | 5 | 0,1s |
| 5 | 1 | 6 | 0,1s |
| 6 | 1 | 7 | 0,1s |
| 7 | 1 | 8 | 0,3s |
| 8 | 1 | 9 | 0,1s |
| 9 | 1 | 10 | 0,1s |
| 10 | 1 | 11 | 0,2s |
| … | …. | …. | …. |
| 100 | 1 | 100 | 0,1s |
| Total | | | 12,2s |

Pada pengujian waktu komputasi, seluruh data akan di rata-rata kan dengan rumus berikut:

$$rata - rata = \frac{total}{banyak\ data}$$

$$rata - rata = \frac{12{,}2}{100}$$

$$rata - rata = 0{,}122$$

Pada perhitungan diatas dapat disimpulkan rata-rata waktu komputasi dari program adalah 0,122 detik

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Berdasarkan hasil hasil dari Tugas Akhir ini, maka dapat ditarik beberapa kesimpulan yaitu :

1. Sistem yang dibuat untuk melakukan pencarian jarak terdekat dari data keluaran database dapat berkerja dengan baik yang menghasilkan waktu komputasi yang cepat yaitu 0.4 detik..
2. Sistem yang dibuat dapat mengakses 100 data.

### 5.2 Saran

Berdasarkan hasil hasil dari Tugas Akhir ini, penulis memiliki beberapa saran yaitu :

1. Penelitian selanjutnya diharapkan sistem dapat mensorting data lebih banyak dan waktu yang dibutuhkan lebih cepat dan menggunakan algoritma selain greedy

## Referensi

[1] Al Akbar, F. F., & Sumiati. (2016). PENENTUAN RUTE DISTRIBUSI TEH BOTOL MENGGUNAKAN METODE TRAVELING SALESMAN PROBLEM (TSP) UNTUK MINIMASI BIAYA DISTRIBUSI. 11.

[2] Furqon, M., Nasution, Y. R., & Nurdianti, T. S. (2020). PENERAPAN ALGORITMA GREEDY UNTUK MENENTUKAN RUTE TERPENDEK. 9.

[3] Hayati, E. N., & Yohanes, A. (2014). *PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA GREEDY*, 7.

[4] Mahendra, Y. D., Nuryanto, & Burhanuddin, A. (2019). *SISTEM PENENTUAN JARAK TERDEKAT DALAM PENGIRIMAN DARAH DI PMI KOTA SEMARANG DENGAN METODE ALGORITMA GREEDY*, 6.

[5] Pratama, N. A., & Hermawan, C. (2016). *APLIKASI PEMBELAJARAN TES POTENSI AKADEMIK*, 6.

[6] Siregar, A. A. (2019). *PENGERTIAN GRAPH*, 7.

[7] Surahman, S., & Setiawan, E. B. (2017). *Aplikasi Mobile Driver Online Berbasis*, 8.

[8] Zufadla, D. A., Raharjo, M.T., D., & Safitri, S.T.,M.Sc, I. (2019). IDENTIFIKASI SIDIK JARI MENGGUNAKAN METODE HISTOGRAM OF ORIENTED GRADIENTS DAN KLASIFIKASI K-NEAREST NEIGHBOR. 5.