Optimasi Rute Angkutan Kota dengan Menggunakan Metode Discrete Bat Algorithm (DBA) (Studi Kasus Rute Angkutan Kota di Kota Bandung)

Muhammad Ihsan Fajri[1], Drs. Mahmud 'Imrona, MT.[2], Irma Palupi, S.Si., M.Si.[3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[4]Divisi Digital Service PT Telekomunikasi Indonesia
[1]ihsanfajri@student.telkomuniversity.ac.id, [2]mahmudimrona@telkomuniversity.ac.id,
[3]irmapalupi@telkomuniversity.ac.id

Abstrak

Pada saat ini semua warga Kota Bandung membutuhkan kendaraan untuk melakukan perjalanan ke suatu tempat yang tidak bisa ditempuh melalui jalan kaki, oleh karena itu pemerintahan daerah memberikan sarana Angkutan Kota. Permasalahan pada penelitian ini adalah rute angkutan kota yang telah ditetapkan oleh pemerintah daerah masih ada jalan utama yang belum dilalui oleh Angkutan Kota. Oleh karena itu, penelitian ini dibuat untuk mendapatkan rute rekomendasi baru yang optimal dengan memperhatikan keinginan dan kebutuhan sopir, penumpang, dan pemerintah daerah dengan menggunakan metode *Discrete Bat Algorithm*. Rute yang diimplementasikan oleh *Discrete Bat Algorithm* akan dibandingkan dengan rute lama yang berdasarkan surat keputusan Walikota Bandung. Dari hasil analisis, dapat disimpulkan rute rekomendasi baru menghasilkan rute yang optimal, karena rute rekomendasi baru menghasilkan jumlah jarak yang lebih jauh, pendapatan sopir yang lebih banyak, dan jumlah ruas jalan yang lebih banyak jika dibandingkan dengan rute lama.

Kata kunci : Travelling Salesman Problem, Optimazition, Bat Algorithm, Discrete Bat Algorithm

Abstract

At this time all residents of Bandung need a vehicle to travel to a place that cannot be reached by foot, therefore the local government provides City Transportation facilities. The problem in this study is that the city transportation routes that have been determined by the local government are still main roads that have not been traversed by the City Transportation. Therefore, this research was made to obtain new optimal recommended routes by taking into account the wants and needs of drivers, passengers, and local governments by using the Discrete Bat Algorithm method. The route implemented by the Discrete Bat Algorithm will be compared with the old route based on the Bandung Mayor's decree. From the analysis, it can be concluded that the new recommended route produces the optimal route, because the new recommended route produces a greater number of distances, more driver income, and a higher number of roads compared to the old route.

Keyword* : *Travelling Salesman Problem, Optimazition, Bat Algorithm, Discrete Bat Algorithm

1. Pendahuluan

Latar Belakang

Pada saat ini semua warga Kota Bandung membutuhkan kendaraan untuk melakukan perjalanan ke suatu tempat yang tidak bisa ditempuh dengan jalan kaki saja. Akan tetapi tidak semua orang mempunyai kendaraan, oleh karena itu pemerintah memberikan sarana berupa kendaraan umum kepada warga Kota Bandung, salah satunya ialah Angkutan Kota. Angkutan Kota adalah sebuah kendaraan yang mengangkut penumpang ataupun barang yang berada di suatu wilayah kota yang sudah terikat dalam trayek yang tetap dan teratur [1]. Saat ini Pemerintah Kota Bandung mempunyai Kendaraan Angkutan Kota berjumlah 39 Trayek [2]. Setiap Trayek Angkutan Kota mempunyai tarif yang berbeda-beda. Tarif pada Trayek Angkutan Kota berkisaran dari Rp. 2.900,00/Km sampai Rp. 4.400,00/Km [3]. Trayek Angkutan Kota sudah ditetapkannya nama jurusan dan nomor Angkutan Kota, dimana pada setiap masing masing jurusan Trayek Angkutan Kota mempunyai 2 rute, yaitu rute masuk, dan rute keluar.

Rute yang dilalui kendaraan Angkutan kota merupakan jalan utama di Kota Bandung. Akan tetapi dari banyaknya jumlah Trayek Angkutan Kota tersebut, masih ada jalan utama yang belum dilalui oleh Angkutan Kota tersebut. Jalan utama yang belum dilewati oleh angkutan kota, bisa berpotensi mendapatkan banyak penumpang, dimana itu akan membuat penghasilan sopir menjadi bertambah. Selain itu, penumpang yang berada di jalan yang belum dilewati oleh Angkutan Kota tersebut harus menempuh jauh ke lokasi rute Angkutan Kota tersebut.

Oleh karena itu, dengan masih ada jalan utama yang belum dilewati oleh Angkutan Kota, dibutuhkan optimasi pada rute Angkutan Kota tersebut. Dalam melakukan optimasi terhadap rute angkutan kota dengan menggunakan algoritma Discrete Bat algorithm (DBA).

Topik dan Batasannya

Topik permasalahan pada penelitian ini adalah bagaimana menentukan formulasi permasalahan penentuan rute angkutan kota yang optimal di wilayah kota bandung dengan memperhatikan keinginan dan kebutuhan sopir, penumpang, dan pemerintah.

Ada juga Batasan dalam penelitian ini, yaitu berdasarkan surat keputusan dari direktur jendral perhubungan darat, pada jenis angkutan kota hanya bisa menampung atau mengangkut penumpang sebanyak 8 penumpang, dan maksimum waktu tempuh setiap kendaraan angkutan kota dari tempat asal ke tempat tujuan adalah 2 jam [1].

Tujuan

Tujuan dari penelitian ini adalah mendapatkan rute Angkutan Kota yang optimal di wilayah Kota Bandung dengan memperhatikan keinginan dan kebutuhan sopir, penumpang, dan pemerintah daerah.

Organisasi Tulisan

Penulisan Tugas Akhir ini memiliki beberapa bagian, antara lain: bagian 2 (studi terkait) merupakan bagian yang menjelaskan studi yang terkait dengan penelitian ini, bagian 3 (sistem yang dibangun) merupakan bagian yang menjelaskan sistem seperti apa yang akan dibangun pada penelitian ini, bagian 4 (evaluasi) merupakan bagian yang akan menjelaskan hasil pengujian dan hasil analisis dari pengujian system yang sudah dibangun, dan bagian 5 (kesimpulan) merupakan bagian yang menyimpulkan dari hasil penelitian ini.

2. Studi Terkait

Saat ini, sudah ada banyak penelitian yang terkait dengan optimasi rute, berikut jurnal dan paper yang terkait dengan penelitan ini:

Penelitian [4] mengusulkan suatu algoritma dalam versi diskrit dari algoritma *Bat Algorithm* (BA) untuk menyelesaikan *Symmetric Travelling Salesman Problem* (STSP). Penelitian ini akan membandingkan *Discrete Bat Algortihm* (DBA) dengan beberapa algoritma-algoritma metaheuristic yang lainnya. Algoritma tersebut adalah *Discrete Particle Swarm Optimazation* (DPSO), *Genetic Simulated Annealing Ant Colony System with Particle Swarm Optimazation Techniques* (GSA-ACS-PSOT), dan *Discrete Cuckoo Search* (DCS). Hasil dari algoritma yang diusulkan pada penelitian ini, memberikan hasil yang baik dalam menyelesaikan masalah STSP tersebut. Adanya perkembangan yang akan dilakukan dari hasil yang telah di dapat pada penelitian ini, yaitu memperluas DBA untuk menyelesaikan masalah optimasi kombinatorial lainnya seperti masalah rute kendaraan, masalah penjadwalan, dan masih banyak lainnya.

Penelitian [5] menjelaskan tentang penyelesaian masalah Traveling Salesman Problem (TSP) dengan Menggunakan algoritma diskrit kelelawar (DBA). Hasil perhitungan dari algoritma yang digunakan pada penelitian ini dibandingkan dengan beberapa algoritma lainnya seperti *Max-Min Ant System* (MMAS), *Ant System* (AS), *Ant System with Elitist Strategy* (ASE), *Rankbased Version of Ant System* (ASrank), *Ant Colony System* (ACS), *Genetic Algorithm* (GA), *Particle Swarm Optimization* (PSO), *Simulated Annealing* (SA), dan *Chaotic Ant Swarm* (CAS). Dimana hasil yang didapatkan dari algoritma yang diusulkan lebih efisien dalam menyelesaikan masalah pada TSP tersebut.

Penelitian [6] menjelaskan tentang penyelesaian suatu masalah pada *Traveling Salesman Problem* (TSP) dan masalah *Asymmetric Traveling Salesman Problem* (ATSP). Penelitian ini mengusulkan peningkatan *Bat Algorithm* (BA) dalam menyelesaikan masalah perutean. Peningkatan terhadap algoritma BA yang disebut juga dengan *Improved Discrete Bat Algorithm*, dilakukannya dengan menambahkannya kecerdasan pada kelelawar tersebut sehingga membuat kelelawar mengikuti pola pergerakan yang berbeda tergantung pada titik ruang solusi dimana mereka berada. Dalam penelitian ini hasil dari algoritma *Improved Discrete Bat Algorithm* akan dibandingkan dengan 5 algoritma metaheuristik lainnya yaitu: *Genetic Algorithm* (GA), *Evolutionary Simulated Annealing*, *Island Based Distributed Genetic Algorithm*, *Discrete firefly algorithm*, dan *Discrete Imperialist Competitive Algorithm*. Dari hasil yang diperoleh, menunjukan algoritma *Improved Discrete Bat Algorithm* untuk TSP dan ATSP mempunyai kinerja baik.

Penelitian [7] ini terdapat pengujian terhadap kinerja pada komputasi DBA dalam menyelesaikan masalah pada *Traveling Salesman Problem* (TSP). Pengujian tersebut juga akan dibandingkan 2 metode yang berbeda, yaitu: metode Chen and Chien's, dan metode Marinakis et al.'s. hasil dari penelitian tersebut berhasil menunjukan algoritma DBA memiliki kinerja yang lebih baik dibandingkan dengan algoritma yang lainnya.

2.1 Traveling Salesman Problem (TSP)

Traveling Salesman Problem adalah suatu optimasi kombinatorial yang memiliki tujuan untuk mencari biaya terendah dari lintasan Hamilton [8]. Dalam masalah *Traveling Salesman Problem* dapat

diselesaikan dengan berbagai cara, salah satunya dengan menggunakan teknik *Swarm Inteligance*. Sudah banyak penelitian yang menggunakan teknik *Swarm Inteligance* untuk menyelesaikan masalah pada *Traveling Salesman Problem*, seperti *Discrete Bat Agorithm* (DBA) [7] [9], *Bee Colony Optimization* [10], dan masih ada Algoritma Swarm Inteligance lainnya. Pada penelitian yang dilakukan terhadap masalah *Traveling Salesman Problem* dengan menggunakan Teknik *Swarm Intelligance* memberikan hasil solusi yang baik dari beberapa pengujian.

2.2 *Bat Algortihm* (BA)

Bat Algorithm adalah algoritma yang terinspirasi terhadap perilaku kelelawar yang sedang mencari makanan dengan kemampuan ekolokasi yang mampu mengeluarkan bunyi dan mendengarkan pantulan bunyi objek yang terdekat. Algoritma ini pertama kali diusulkan oleh Xin-She Yang pada tahun 2010 [11]. *Bat algorithm* dapat menyelesaikan permasalahan pada *Traveling Salesman Problem*, seperti pada melakukan optimasi pada rute kendaraan. Algoritma yang dikembangkan oleh Xin-She Yang ini mempunyai ketentuan dalam proses algoritmanya [11], ketentuan tersebut antara lain:

1. Semua kelelawar menggunakan Ekolokasi untuk merasakan jarak dan mereka juga tahu perbedaan antara makanan atau mangsa dan latar belakang hambatan dalam beberapa cara ajaib.
2. Kelelawar terbang secara acak dengan kecepatan v_i pada Posisi x_i dengan frekuensi tetap f_{min}, panjang gelombang λ, dan kenyaringan A_0 untuk mencari mangsa. Mereka dapat secara otomatis menyesuaikan panjang gelombang (atau frekuensi) dari getaran yang dipancarkan dan menyesuaikan laju emisi getaran *r* dalam kisaran [0,1], tergantung pada kedekatan target mereka;
3. Meskipun kenyaringan dapat bervariasi dalam banyak hal, kami mengasumsikan bahwa kenyaringannya bervariasi dari besar A_0 (positif) sampai nilai konstanta minimum.

Penelitian [4] menawarkan bentuk Psuedo-code untuk *Bat Algorithm* yang akan ditampilkan kedalam sebuah flowchart seperti pada **Error! Reference source not found.**.

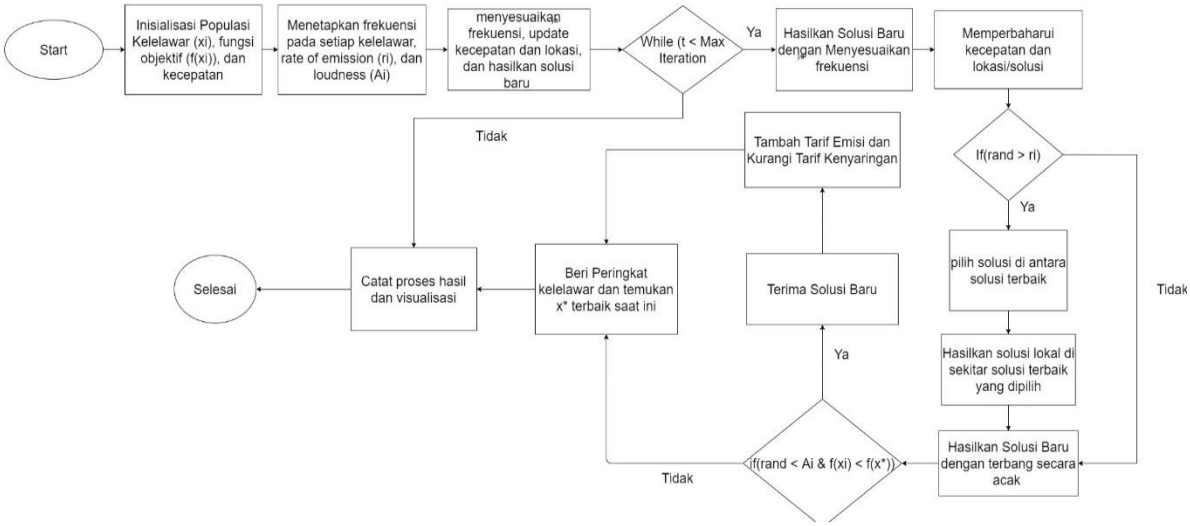


Figure 1 Flowchart Psuedo-code Bat Algorithm

Di bawah ini merupakan penjelasan langkah-langkah Psuedo-code *Bat Algorithm* yang sudah disusun menjadi flowchart seperti pada Error! Reference source not found..

Pertama, Menginisialisasi populasi kelelawar: posisi x_i, Kecepatan v_i, dan Frekuensi f_i. Pergerakan kelelawar virtual diberikan dengan memperbarui kecepatan mereka v_i^t dan posisi mereka x_i^t pada waktu t langkah menggunakan persamaan:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad \textbf{(1)}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \quad \textbf{(2)}$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

Di mana $\beta \in [0, 1]$ menunjukan vektor yang dihasilkan secara acak dari distribusi seragam. Variabel x_* menunjukan lokasi terbaik global saat ini (solusi) yang terletak setelah membandingkan semua solusi yang disediakan oleh kelelawar.

Kedua, nomor acak diterapkan, jika nomor acak ini memverifikasi kondisi $(if(rand > r))$ dan setelah pemilihan satu solusi di antara solusi terbaik kelelawar saat ini, solusi baru akan diterima di sekitar solusi terbaik saat ini.

$$x_{new} = x_{old} + \varepsilon A^t \quad (4)$$

Dimana $\varepsilon \in [-1,1]$ adalah nilai angka random, sedangkan $A^t = (A_i^t)$ kenyaringan rata rata dari semua kelelawar pada generasi saat ini.

Ketiga, kenyaringan A_i dan getaran emisi r_i akan diperbaharui, dan solusi akan diterima jika nomor acak kurang dari kenyaringan A_i dan $f(x_i) < f(x_*)$. A_i dan r_i akan diperbarui dengan persamaan dibawah ini:

$$A_i^{t+1} = \propto A_i^t \quad (5)$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \quad (6)$$

Dimana $\propto, \gamma$ adalah konstanta. Algoritma akan diulang-ulang sampai jumlah siklus maksimum tercapai.

2.3 *Discrete Bat Algorithm* (DBA)

Discrete Bat Algorithm merupakan algoritma yang diusulkan oleh Yassie Saji dan Mohammed Essaid Riff [9]. Dimana DBA tersebut memiliki konsep dan langkah-langkah yang tidak jauh berbeda dengan algoritma BA yang dasar, perubahan pada BA ke DBA hanya pada algoritma BA untuk beralih ke ruang diskrit algoritma DBA yang telah diusulkan [9]. Terdapat langkah-langkah dari algoritma DBA yang telah di usulkan untuk menyelesaikan TSP, langkah-langkah tersebut akan dijelaskan secara singkat di bawah ini.

1. Dilakukan Inisialisasi ukuran populasi kelelawar: Posisi x_i, Kecepatan v_i, dan Frekuensi f_i, Pergerakan kelelawar diberikan dengan memperbarui kecepatan mereka v_i^t dan posisi mereka x_i^t pada waktu t , dengan menggunakan persamaan sebagai berikut:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (1)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

   Hasilkan posisi awal acak x_i untuk setiap kelelawar. Inisialisasi tangkat emisi $r_i \in [0,1]$ dan kenyaringan $A_i \in [0,1]$. tentukan frekuensi getaran f_{min} dan f_{max} dalam kisaran [1, n].
2. Akan dilakukannya penghitungan solusi global yang terbaik
3. Untuk setiap kelelawar, hitung solusi baru dengan frekuensi dalam persamaan (1), dan memperbaharui kecepatan dan lokasi, dengan menggunakan persamaan (2) dan (3).
4. Gunakan pembangkitan bilangan acak untuk menentukan r_i, yang selanjutnya akan dipilih solusi terbaik waktu setiap iterasi.
5. Mengevaluasi solusi baru menggunakan persamaan sebagai berikut:

$$f(\pi) = \sum_{i=1}^{n-1} d_{\pi_i,\ \pi_{i+1}} + d_{\pi_n,\ \pi_1} \quad (7)$$

Dimana dengan $\pi = [\pi_1, \pi_2, \pi_3, \dots, \pi_n]$, $\pi_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$, dan

$$d_{\pi_i,\pi_j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad \textbf{(8)}$$

6. Hitung solusi baru dengan terbang secara acak
7. Hasil angka acak, jika angka ini kurang dari kenyaringan A_i, dan solusi terakhir yang evaluasi lebih baik daripada solusi terbaik jadi terima solusi baru.
8. Hentikan algoritma jika jumlah iterasi masksimum tercapai, atau mengulang Kembali langkah pada no.3 dan seterusnya.

Langkah-langkah yang sudah dijelaskan di atas merupakan penjelasan dari pseudo-code *Discrete Bat Algorithm* pada paper [4], pseudo-code tersebut akan digambarkan dengan bentuk flowchart di bawah ini.

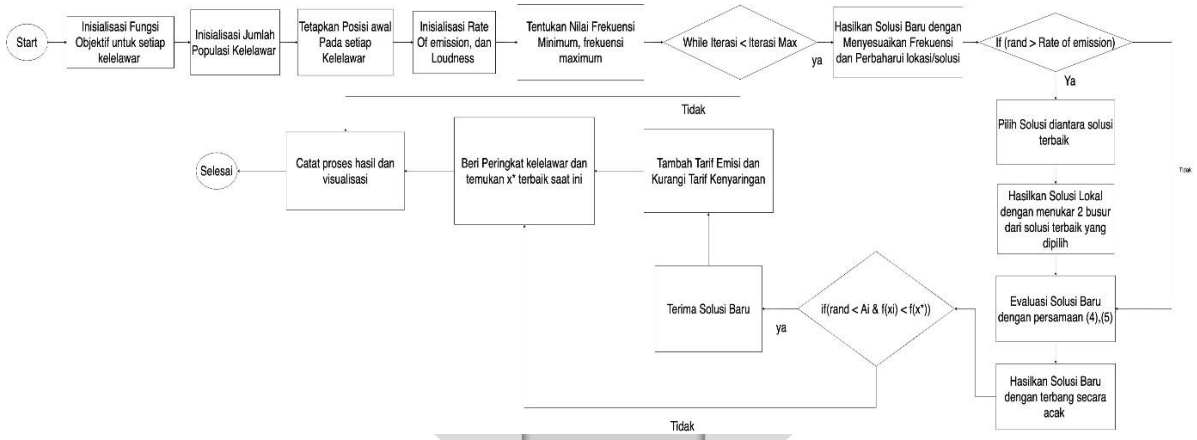


Figure 2 Flowchart Psuedo-code Discrete Bat Algorithm

3. Sistem yang Dibangun

Pada Penelitian ini telah membangun sebuah system, dimana system tersebut memiliki tahapan-tahapan untuk menyelesaikan permasalahan pada kasus ini, tahapan-tahapan pada sistem tersebut telah digambarkan pada diagram flowchart di bawah ini.

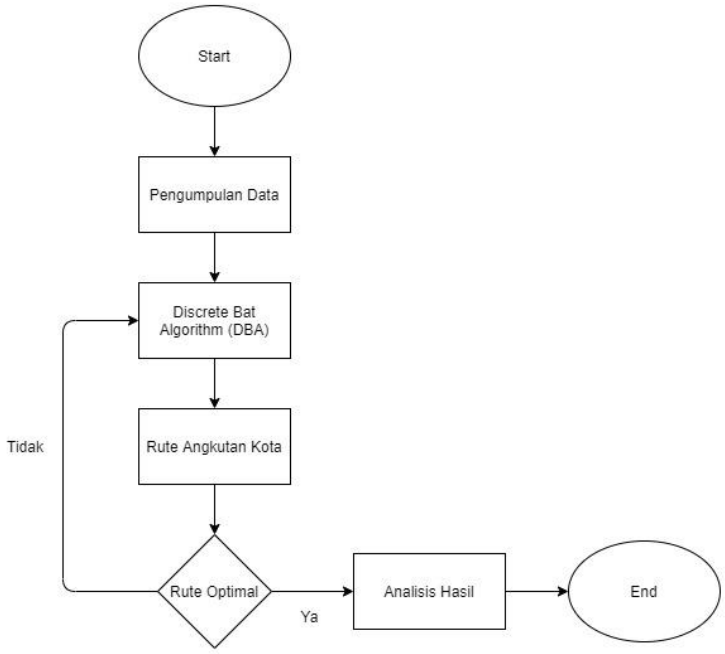


Figure 3 Diagram Flowchart Sistem yang di bangun

Pada penelitian ini membutuhkan data untuk diproses oleh algoritma *Discrete Bat Algorithm* (DBA). data-data yang digunakan antara lain: titik koordinat (latitude, longitude), nama jalan, jarak, dan data penumpang. Pada data titik latitude, titik longitude dan nama jalan diambil dari aplikasi website yang bernama *birdtheme.org*, dimana pengambilan data titik latitude, titik longitude, dan nama jalan tersebut berdasarkan pada surat keputusan walikota bandung [12]. Selain data yang diambil berdasarkan pada surat keputusan walikota, data titik koordinat dan nama jalan tersebut juga diambil dari beberapa rekomendasi rute angkutan kota yang dibuat sendiri. Rute rekomendasi tersebut diambil dengan jarak yang tidak jauh dari rute yang telah diputuskan oleh walikota, kurang lebih radius jarak rute yang diambil adalah 10 Km. Pada data jarak didapatkan dengan menggunakan google maps, untuk mengumpulkan data jarak diperlukan data titik latitude dan titik longitude. Pada pengumpulan data-data tersebut dilakukan dengan secara manual.

Data yang sudah dikumpulkan tersebut akan diproses dengan Discrete Bat Algorithm (DBA), lalu akan menghasilkan sebuah rute angkutan kota baru yang sudah dioptimasi. Proses tersebut akan dilakukan ulang dimana jika rute yang didapat belum optimal maka akan kembali diproses oleh algoritma DBA tersebut. Setelah sudah mendapatkan rute baru pada angkutan kota yang telah dioptimasi dengan Discrete Bat Algorithm (DBA), akan dilakukannya analisis perbandingan antara rute baru dengan rute yang berdasarkan surat keputusan walikota bandung [12].

3.1 *Discrete Bat Algorithm*

Pada penentuan rute rekomendasi baru memiliki langkah-langkah sebagai berikut,

Pertama, menginisialisasi jumlah kelelawar yang berisikan jumlah titik pecabangan jalan. kelelawar tersebut memiliki nilai yang berisikan jumlah penumpang. Kedua, inisialisasi kecepatan perpindahan kelelawar dari titik lama ke titik baru, dimana setiap kelelawar akan menghasilkan kecepatan masing – masing. Selanjutnya inisialisasi frekuensi,frekuensi memiliki nilai batas minimum dan maximum. Langkah berikutnya menginisialisasi Loudness dan rate of emission. Kemudian inisialisasi solusi dan posisi awal kelelawar, pada posisi awal kelelawar sudah tentukan berdasarkan data titik koordinat. Setiap trayek memiliki posisi awal kelelawar yang berbeda-beda. Kemudian menghitung setiap fitness kelelawar, setelah itu sistem akan mendapatkan nilai fitness terbaik. setelah mendapatkan fitness terbaik langkah selanjutnya menentukan iterasi maximumnya pada setiap kelelawar. Kemudian masuk ke proses update posisi dan kecepatan. Pertama update frekeunsi pada setiap kelelawar dengan menggunkan **persamaan (1).** Kemudian update posisi dengan menggunakan **persamaan (3)** dan juga update kecepatan pada setiap kelelawar dengan **persamaan (2).**

kemudian masuk ke proses perbandingan antara nilai acak dengan nilai rate of emission, jika nilai acak lebih besar dari nilai rate of emission maka solusi kelelawar akan di perbarui dengan menggunakan

persamaan (4). Kemudian masuk ke proses perbandingan antara nilai acak dengan nialai loudness dan nilai fitness terbaru. Jika nilai acak lebih kecil dari nilai loudness, dam frekuensi baru lebih kecil dari fruensi lama, maka update nilai loudnessnya dengan menggunakan **persamaan (5)** dan update nilai rate of emission dengan menggunakan **persamaan (6).**

Jika nilai iterasi sudah sampai nilai iterasi maximum yang sudah ditemtukan di awal, maka akan mendapatkan titik selanjutnya, proses tersebut akan berhenti ketika proses pada algoritma sampai ke titik tujuan yang sudah di tetapkan dari awal.

Setelah proses algoritma selesai akan menghasilkan rute trayek baru, jarak tempuh, jumlah pendapatan sopir. Jumlah pendapatan sopir dihitung dengan menggunkan persamaan sebagai berikut

Pertama menentukan tarif penumpang per kilometer

$$Tarif\ perkilometer = \frac{Tarif\ Angkutan}{Jarak\ Angkutan\ Kota}$$

Lalu setelah menentukan tarif penumpang perkilometer, lalu menghitung jumlah pendapatan sopir

$$Pendapatan\ Sopir = Tarif\ Perkilometer * jumlah\ orang\ yang\ turun$$

4. Evaluasi

4.1 Hasil Pengujian

Dalam penelitian ini, Rute Trayek yang di uji ada 6 trayek, yaitu Trayek 20 (Ciroyom – Bumi Asri), Trayek 21 (Ciroyom – Cikudapateuh), Trayek 22 (Sederhana – Cipagalo), Trayek 23 (Sederhana – Cijerah), Trayek 24 (Sederhana – Cimindi), dan Trayek 25 (Ciwastra – Ujung Berung). Dalam pengujian pada Trayek-Trayek tersebut diuji dengan menggunakan *Discrete Bat Algorithm*. Hasil dari pengujian tersebut akan dibandingkan dengan data Trayek yang sudah ada berdasarkan surat keputusan walikota bandung [12].

Tabel 1 Perbandingan Hasil Pengujian dengan Rute berdasarkarkan Walikota Bandung

No. Trayek	Nama Trayek	Hasil Pengujian			Keterangan
		Jarak Tempuh	Jumlah Pendapatan	Jumlah Ruas Jalan	
20	Ciroyom – Bumi Asri (Keluar)	16,5 Km	Rp. 118.000	395 Ruas Jalan	Rute Lama
	Ciroyom – Bumi Asri (Keluar)	17,3 Km	Rp. 139.000	453 Ruas Jalan	Rute Rekomendasi Baru
	Ciroyom – Bumi Asri (Masuk)	7,9 Km	Rp. 82.000	186 Ruas Jalan	Rute Lama
	Ciroyom – Bumi Asri (Masuk)	16,09 Km	Rp. 200.000	402 Ruas Jalan	Rute Rekomendasi Baru
21	Ciroyom – Cikudapateuh (Keluar)	13,3 Km	Rp. 162.000	323 Ruas Jalan	Rute Lama
	Ciroyom – Cikudapateuh (Keluar)	18,4 Km	Rp. 197.000	400 Ruas Jalan	Rute Rekomendasi Baru
	Ciroyom – Cikudapateuh (Masuk)	12,3 Km	Rp. 166.000	311 Ruas Jalan	Rute Lama
	Ciroyom – Cikudapateuh (Masuk)	11,9 Km	Rp. 144.000	284 Ruas Jalan	Rute Rekomendasi Baru
22	Sederhana – Cipagalo (Keluar)	15,6 Km	Rp. 222.000	454 Ruas Jalan	Rute Lama
	Sederhana – Cipagalo (Keluar)	17,22 Km	Rp. 215.000	443 Ruas Jalan	Rute Rekomendasi Baru
	Sederhana – Cipagalo (Masuk)	16,01 Km	Rp. 199.500	414 Ruas Jalan	Rute Lama
	Sederhana – Cipagalo (Masuk)	15,54 Km	Rp. 189.000	412 Ruas Jalan	Rute Rekomendasi Baru
23	Sederhana – Cijerah (Keluar)	7,7 Km	Rp. 102.500	216 Ruas Jalan	Rute Lama
	Sederhana – Cijerah (Keluar)	9,8 Km	Rp. 118.500	242 Ruas Jalan	Rute Rekomendasi Baru

	Sederhana – Cijerah (Masuk)	7,6 km	Rp. 99.500	207 Ruas Jalan	Rute Lama
	Sederhana – Cijerah (Masuk)	11,19 Km	Rp. 138.500	218 Ruas Jalan	Rute Rekomendasi Baru
24	Sederhana – Cimindi (Keluar)	9,4 Km	Rp. 124.000	251 Ruas Jalan	Rute Lama
	Sederhana – Cimindi (Keluar)	14,02 Km	Rp. 160.000	341 Ruas Jalan	Rute Rekomendasi Baru
	Sederhana – Cimindi (Masuk)	8,1 Km	Rp. 116.500	249 Ruas Jalan	Rute Lama
	Sederhana – Cimindi (Masuk)	10,018 Km	Rp. 131.000	268 Ruas Jalan	Rute Rekomendasi Baru
25	Ciwastra – Ujung Berung (Keluar)	22,4 Km	Rp. 250.500	516 Ruas Jalan	Rute Lama
	Ciwastra – Ujung Berung (Keluar)	22,4 Km	Rp. 250.500	516 Ruas Jalan	Rute Rekomendasi Baru
	Ciwastra – Ujung Berung (Masuk)	14,48 Km	Rp. 193.500	391 Ruas Jalan	Rute Lama
	Ciwastra – Ujung Berung (Masuk)	9,4 Km	Rp. 126.500	251 Ruas Jalan	Rute Rekomendasi Baru

Pada **Tabel 1** diatas, menampilkan perbandingan rute rekomendasi baru hasil dari *Discrete Bat Algorithm* dengan rute lama yang berdasarkan surat keputusan Walikota Bandung. Terdapat 3 bagian komponen yang dibandingkan pada penelitian ini, yaitu jarak tempuh, Jumlah pendapatan Sopir, dan jumlah ruas jalan. Rute yang dinyatakan optimal ialah ketika jarak tempuh pada rute rekomendasi baru lebih jauh dari rute lama, jumlah pendapatan sopir meningkat dari rute lama, dan jumlah ruas jalan pada rute rekomendasi baru lebih banyak dari rute lama.

4.2 Analisis Hasil Pengujian

4.2.1 Rute Angkutan Kota Trayek 20 (Ciroyom – Bumi Asri)

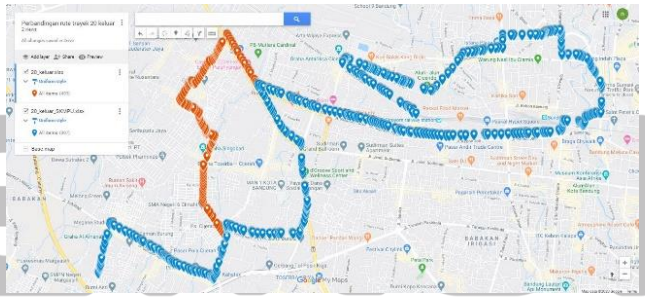


Figure 4 Perbandingan rute Trayek 20 pada rit keluar

Pada **Figure 4** yang ada diatas merupakan gambar perbandingan Trayek 20 pada rit keluar. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari *Discrete Bat Algortima*, dan untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. Rute rekomendasi baru menghasilkan jumlah jarak tempuh 17,2 Km, sedangkan rute lama memiliki jarak tempuh sejauh 16,5 Km. Pendapatan sopir dari rute rekomendasi baru menghasilkan Rp. 139.000, sedangkan untuk rute lama hanya menghasilkan Rp. 118.000. Untuk jumlah ruas pada rute rekomendasi baru mendapatkan 453 ruas jalan, sedangkan dari rute lama memiliki 395 ruas jalan. pada perubahan rute rekomendasi baru trayek 20 rit keluar, menghasilkan rute yang optimal, karerna dari perbandingan tersebut hasil dari rute rekomendasi baru menghasilkan jarak yang lebih jauh, pendapatan sopir meningkat, dan juga jumlah ruas jalan menjadi lebih banyak.

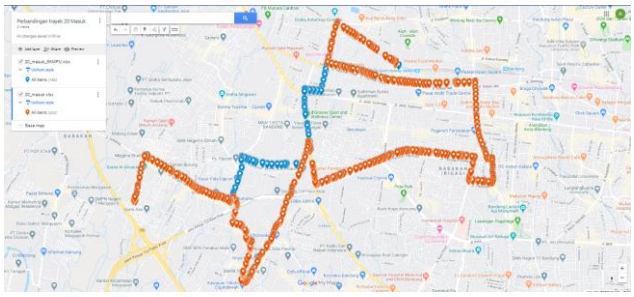


Figure 5 Perbandingan rute Trayek 20 pada rit masuk

Pada **Figure 5** yang ada diatas merupakan gambar perbandingan Trayek 20 pada rit masuk. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari *Discrete Bat Algorithm*, dan untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. Rute rekomendasi baru menghasilkan jumlah jarak tempuh 16,09 Km, sedangkan rute lama memiliki jarak tempuh sejauh 7,9 Km. Pendapatan sopir dari rute rekomendasi baru menghasilkan Rp. 200.000, sedangkan untuk rute lama hanya menghasilkan Rp. 82.000. Untuk jumlah ruas pada rute rekomendasi baru mendapatkan 402 ruas jalan, sedangkan dari rute lama memiliki 186 ruas jalan. pada perubahan rute rekomendasi baru trayek 20 rit masuk, menghasilkan rute yang optimal, karerna dari perbandingan tersebut hasil dari rute rekomendasi baru menghasilkan jarak yang lebih jauh, pendapatan sopir meningkat, dan juga jumlah ruas jalan menjadi lebih banyak.

4.2.2 Rute Angkutan Kota Trayek 21 (Ciroyom – Cikudapateuh)

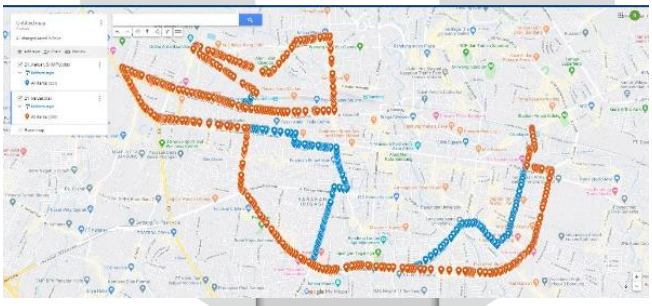


Figure 6 Perbandingan rute Trayek 21 pada rit keluar

Pada **Figure 6** yang ada diatas merupakan gambar perbandingan Trayek 21 pada rit keluar. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari *Discrete Bat Algorithm*, dan untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. Rute rekomendasi baru menghasilkan jumlah jarak tempuh 18,4 Km, sedangkan rute lama memiliki jarak tempuh sejauh 13,3 Km. Pendapatan sopir dari rute rekomendasi baru menghasilkan Rp. 197.000, sedangkan untuk rute lama hanya menghasilkan Rp. 162.000. Untuk jumlah ruas pada rute rekomendasi baru mendapatkan 400 ruas jalan, sedangkan dari rute lama memiliki 323 ruas jalan. pada perubahan rute rekomendasi baru trayek 21 rit keluar, menghasilkan rute yang optimal, karena dari perbandingan tersebut hasil dari rute rekomendasi baru menghasilkan jarak yang lebih jauh, pendapatan sopir meningkat, dan juga jumlah ruas jalan menjadi lebih banyak.

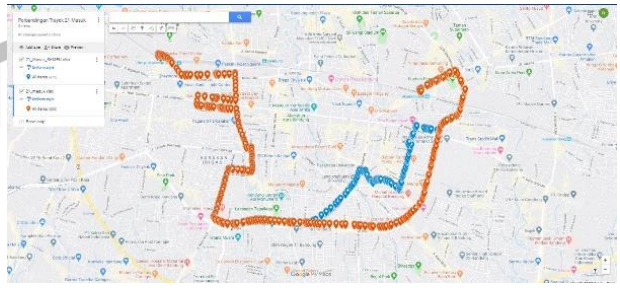


Figure 7 Perbandingan rute Trayek 21 pada rit masuk

Pada **Figure 7** yang ada diatas merupakan gambar perbandingan Trayek 21 pada rit masuk. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari *Discrete Bat Algortihm*, dan

untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. Rute rekomendasi baru menghasilkan jumlah jarak tempuh lebih dekat dari pada rute lama, rute rekomendasi baru menghasilkan jarak tempuh sejauh 11,9 Km, sedangkan rute lama memiliki jarak tempuh sejauh 12,3 Km. Pada hasil pendapatan pada rute rekomendasi baru menghasilkan pendapatan sebanyak Rp. 144.000, sedangkan pendapatan sopir pada rute lama lebih banyak dibandingkan dengan rute rekomendasi baru yang menghasilkan pendapatan sebanyak Rp. 166.000. jumlah ruas yang dihasilkan oleh rute rekomendasi baru sebanyak 284 ruas jalan, sedangkan rute lama menghasilkan 311 ruas jalan. Rute rekomendasi baru pada trayek 21 rit masuk menghasilkan rute yang kurang optimal, dikarenakan jumlah jarak tempuh lebih dekat, hasil pendapatan supir lebih rendah, dan jumlah ruas jalan lebih sedikit dibandingkan dengan rute lama.

4.2.3 Rute Angkutan Kota Trayek 22 (Sederhana – Cipagalo)

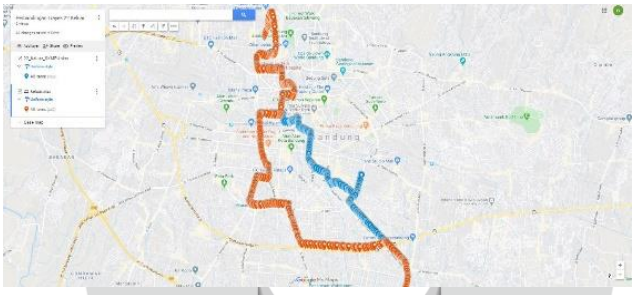


Figure 8 Perbandingan rute Trayek 22 pada rit keluar

Pada **Figure 8** yang ada diatas merupakan gambar perbandingan Trayek 22 pada rit keluar. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari *Discrete Bat Algortihm*, dan untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. Rute rekomendasi baru menghasilkan jarak tempuh sejauh 17,22 Km, sedangkan rute lama memiliki jumlah jarak tempuh lebih dekat yaitu 15,6 Km. Rute rekomendasi baru menghasilkan jumlah pendapatan lebih sedikit dari rute lama. Rute baru menghasilkan pendapatan sopir sebanyak Rp. 215.000 sedangkan rute lama menghasilkan Rp. 222.000. Selain hasil pendapatan sopir pada rute rekomndasi baru lebih sedikit dibandingkan dengan rute lama, hasil jumlah ruas jalan juga lebih sedikit dari jumlah ruas jalan pada rute lama. Rute rekomenaasi baru menghasilkan jumlah ruas jalan sebanyak 443 ruas jalan, sedangkan pada rute lama sebanyak 454 ruas jalan. Perubahan rute tersebut masih dinyatakan meghasilkan rute yang optimal, walaupun hasil jumlah pendapatan sopir dan hasil jumlah ruas jalan lebih sedikit dibandingkan dengan rute lama. Dikarenakan selisih pendapatan sopir dan juga jumlah ruas jalan tidak telalu jauh.

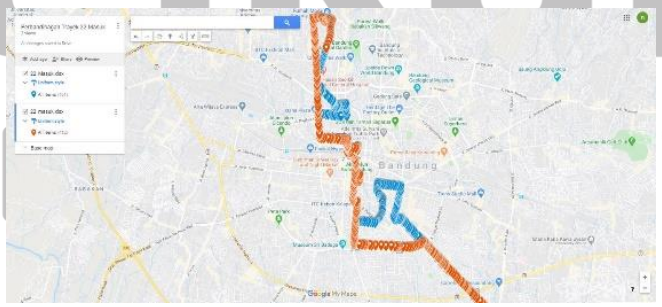


Figure 9 Perbandingan rute Trayek 22 pada rit masuk

Pada **Figure 9** yang ada diatas merupakan gambar perbandingan Trayek 22 pada rit masuk. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari *Discrete Bat Algortihm*, dan untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. Rute rekomendasi baru menghasilkan jarak tempuh sejauh 15,54 Km, sedangkan rute lama memiliki jumlah jarak tempuh lebih dekat yaitu 16,01 Km Rute rekomendasi baru lebih meghasilkan pendapatan sopir sebanyak Rp. 189.000. sedangkan pada rute lama menghasilkan Rp. 199.500. Jumlah ruas jalan yang dihasilkan oleh rute rekomendasi baru sebanyak 412 ruas jalan, sedangkan rute baru memiliki 414 ruas jalan. Perubahan rute tersebut masih dinyatakan meghasilkan rute yang optimal, walaupun rute rekomendasi baru menghasilkan jumlah jarak tempuh, jumlah pendapatan sopir, dan

jumlah ruas jalan lebih sedikit dibandingkan dengan rute lama. Dikarenakan selisih pada jumlah jarak tempuh, pendapatan sopir, dan juga jumlah ruas jalan tidak telalu jauh.

4.2.4 Rute Angkutan Kota Trayek 23 (Sederhana – Cijerah)

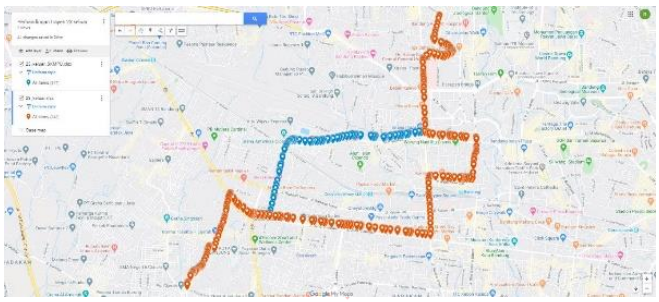


Figure 10 Perbandingan rute Trayek 23 pada rit keluar

Pada **Figure 10** yang ada diatas merupakan gambar perbandingan Trayek 23 pada rit keluar. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari *Discrete Bat Algortihm*, dan untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. Rute rekomendasi baru menghasilkan jumlah jarak tempuh 9,8 Km, sedangkan rute lama memiliki jarak tempuh sejauh 7,7 Km. Pendapatan sopir dari rute rekomendasi baru menghasilkan Rp. 118,500, sedangkan untuk rute lama hanya menghasilkan Rp. 102.500. Untuk jumlah ruas pada rute rekomendasi baru mendapatkan 242 ruas jalan, sedangkan dari rute lama memiliki 216 ruas jalan. pada perubahan rute rekomendasi baru trayek 23 rit keluar, menghasilkan rute yang optimal, karerna dari perbandingan tersebut hasil dari rute rekomendasi baru menghasilkan jarak yang lebih jauh, pendapatn sopir meningkat, dan juga jumlah ruas jalan menjadi lebih banyak.

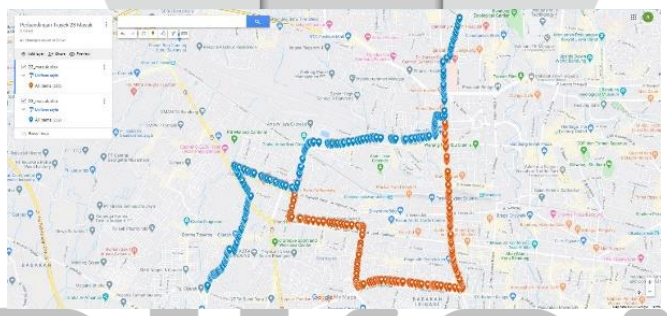


Figure 11 Perbandingan rute Trayek 23 pada rit masuk

Pada **Figure 11** yang ada diatas merupakan gambar perbandingan Trayek 23 pada rit masuk. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari *Discrete Bat Algortihm*, dan untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. Rute rekomendasi baru menghasilkan jumlah jarak tempuh 11,19 Km, sedangkan rute lama memiliki jarak tempuh sejauh 7,6 Km. Pendapatan sopir dari rute rekomendasi baru menghasilkan Rp. 138.500, sedangkan untuk rute lama hanya menghasilkan Rp. 99.500. Untuk jumlah ruas pada rute rekomendasi baru mendapatkan 281 ruas jalan, sedangkan dari rute lama memiliki 207 ruas jalan. Rute rekomendasi baru merupakan rute yang optimal, karena rute rekomendasi baru menghasilkan jarak tempuh lebih jauh, pendapatan sopir meningkat, dan jumlah ruas jalan lebih banyak dibandingkan dengan rute lama yang berdasarkan surat keputusan Walikota Bandung.

4.2.5 Rute Angkutan Kota Trayek 24 (Sederhana – Cimindi)

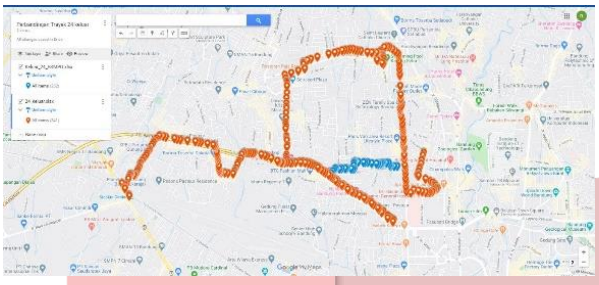


Figure 12 Perbandingan rute Trayek 24 pada rit keluar

Pada **Figure 12** yang ada diatas merupakan gambar perbandingan Trayek 24 pada rit keluar. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari *Discrete Bat Algortihm*, dan untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. Rute rekomendasi baru menghasilkan jumlah jarak tempuh 14,02 Km, sedangkan rute lama memiliki jarak tempuh sejauh 9,4 Km. Pendapatan sopir dari rute rekomendasi baru menghasilkan Rp. 160.000, sedangkan untuk rute lama hanya menghasilkan Rp. 124.000. Untuk jumlah ruas pada rute rekomendasi baru mendapatkan 341 ruas jalan, sedangkan dari rute lama memiliki 251 ruas jalan. Rute rekonedasi baru merupakan rute yang optimal, karena rute rekomendasi baru menghasilkan jarak tempuh lebih jauh, pendapatan sopir bertambah, dan jumlah ruas jalan lebih banyak dibandingkan dengan rute lama.

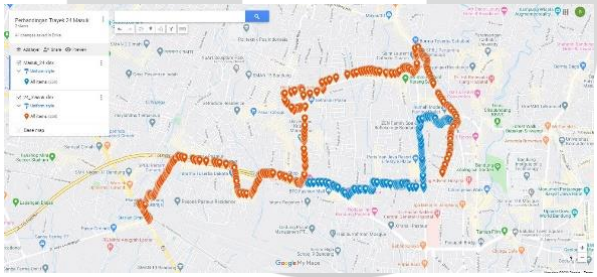


Figure 13 Perbandingan rute Trayek 24 rit masuk

Pada **Figure 13** yang ada diatas merupakan gambar perbandingan Trayek 24 pada rit masuk. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari *Discrete Bat Algortihm*, dan untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. Rute rekomendasi baru menghasilkan jumlah jarak tempuh 10,018 Km, sedangkan rute lama memiliki jarak tempuh sejauh 8,1 Km. Pendapatan sopir dari rute rekomendasi baru menghasilkan Rp. 131.000, sedangkan untuk rute lama hanya menghasilkan Rp. 116.500. Untuk jumlah ruas pada rute rekomendasi baru mendapatkan 268 ruas jalan, sedangkan dari rute lama memiliki 249 ruas jalan. Karena rute rekomendasi baru menghasilkan jarak tempuh lebih jauh, pendapatan sopir bertambah, dan jumlah ruas jalan lebih banyak dibandingkan dengan rute lama. Oleh karena itu, Rute rekomendasi baru dikatakan rute yang optimal.

4.2.6 Rute Angkutan Kota Trayek 25 (Ciwastra – Ujung Berung)

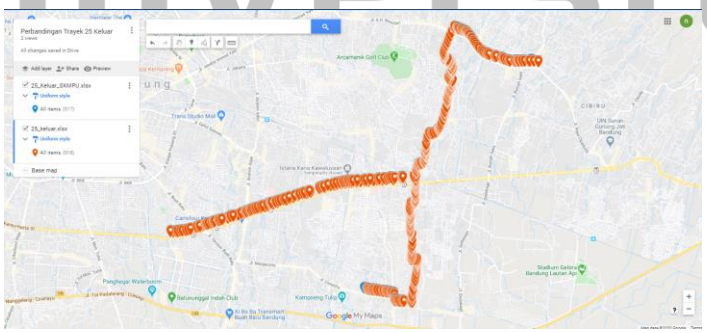


Figure 14 Perbandingan rute Trayek 25 pada rit keluar

Pada **Figure 14** yang ada diatas merupakan gambar perbandingan Trayek 20 pada rit keluar. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari Discrete Bat Algortima,

dan untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. Rute rekoemndasi baru pada Trayek 20 pada rit keluar tidak mengalami perubahan sama sekali dari rute lama, rute rekomendasi baru dan juga rute lama menghasilkan jarak tempuh 22,4 Km, pendapatan sopir sebanyak Rp. 250.500, dan menghasilkan jumlah ruas jalan sebanyak 516 ruas jalan.

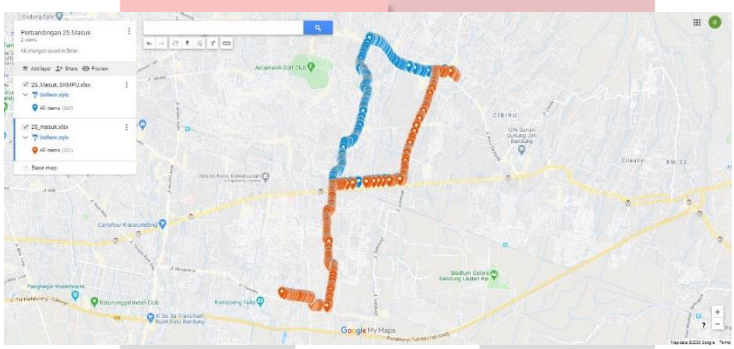


Figure 15 Perbandingan rute Trayek 25 pada rit masuk

Pada **Figure 15** yang ada diatas merupakan gambar perbandingan Trayek 25 pada rit masuk. Titik yang berwarna merah merupakan titik rute rekomendasi baru hasil dari *Discrete Bat Algortihm*, dan untuk titik yang berwarna biru merupakan titik rute lama yang berdasarkan surat keputusan Walikota Bandung. pada Trayek 25 rit masuk, setelah di implementasikan dengan *Discrete Bat Algorithm*, mendaoatkan hasil yang belum optimal, karena pada rute rekomendasi baru menghasilkan jarak tempuh yang lebih dekat dibandingkan dengan jarak tempuh pada rute lama, dimana rute rekomendasi baru hanya menghasilkan jarak 9,4 Km dibandingkna dengan rute lama yang memilki jarak 14,4 Km. Begitu juga dengan pendapatan sopir hanya mendapatkan Rp. 126.500 sedangkan rute lama mendapatkan pendapatan sebesar Rp. 193.500. jumlah ruas jalan juga menghasilkan lebih sedikit dibandingkan dengan rute lama, dimana rute rekomendasi baru menghasilkan ruas jalan sebanyak 251 ruas jalan, sedangkan rute lama menghasilka ruas 391 ruas jalan.

5. Kesimpulan

Hasil analisis pada rute rekomendasi baru trayek 20 hingga 24 menghasilkan rute yang optimal juga memenuhi keinginan dan kebutuhan sopir, penumpang, dan pemerintah daerah. Rute rekomendasi baru trayek tersebut menghasilkan jarak tempuh yang lebih jauh jika dibandingkan dengan rute lama. Pada jumlah pendapatan sopir juga meningkat jika dibandingkan dengan rute sebelumnya. Jumlah ruas jalan bertambah lebih banyak dari rute sebelumnya. Jumlah jarak tempuh lebih jauh dan pendapat lebih banyak merupakan keinginan serta kebutuhan sopir angkutan kota. Sedangkan jumlah ruas jalan lebih banyak adalah keinginan serta kebutuhan penumpang dan pemerintah daerah.

Sedangkan hasil analisis pada rute rekomendasi baru trayek 25 menghasilkan rute yang kurang optimal juga belum memenuhi keinginan serta kebutuhan sopir, penumpang, dan juga pemerintah daerah. Dimana rute rekomendasi baru trayek 25 menghasilkan jarak yang kurang jauh, jumlah pendapatan lebih rendah, dan jumlah ruas jalan lebih sedikit jika dibandingkan dengan rute sebelumnya.

Daftar Pustaka

[1] D. J. P. Darat, Pendoman Teknis Penyelenggaraan Angkutan Penumpang Umum di Wilayah Perkotaan Dalam Trayek Tetap dan Teratur, Jakarta: Departemen Perhubungan RI, 2002.

[2] P. K. Bandung, "Jumlah Armada Angkutan Kota di Wilayah Kota Bandung," Dinas Perhubungan Kota Bandung, Bandung, 2008.

[3] P. K. Bandung, "Penetapan Tarif Angkutan Penumpang Umum Di Kota Bandung," Pemerintah Kota Bandung, Bandung, 2016.

[4] Y. Saji and M. E. Riff, "A novel discrete bat algorithm for solving the travelling salesman Problem," *Neural Computer & Application*, 2015.

[5] Y. Saji, M. E. Riffi and B. Ahiod, "Discrete bat-inspired algorithm for travelling salesman problem," *2014 Second World Conference on Complex Systems (WCCS)*, 2014.

[6] E. Osaba, X.-S. Yang, F. Diaz and P. Lopez-Garcia, "An improved discrete bat algorithm for symmetric and asymmetric Traveling Salesman Problem," *Engineering Applications of Artificial Intelligence*, 2016.

[7] Z. Jiang, "Discrete Bat Algorithm for Traveling Salesman Problem," *2016 3rd International Conference on Information Science and Control Engineering*, 2016.

[8] M. T. A. S. S. S. Paulo Henrique Siqueira*, "A new approach to solve the traveling salesman problem," *Neurocomputing*, vol. 70, no. 4-6, pp. 1013-1021, 2007.

[9] M. E. R. Yassine Saji, "A novel discrete bat algorithm for solving the travelling salesman problem," *Neural Computing and Applications*, pp. 1853-1866, 2016.

[10] M. Y. H. L. C. S. C. Li-Pei Wong, "A Bee Colony Optimization Algorithm for Traveling Salesman Problem," in *Second Asia International Conference on Modelling & Simulation*, Nanyang Avenue, Singapure, 2008.

[11] X.-S. Yang, A New Metaheuristic Bat-Inspired Algorithm, Cambridge: Springer, 2010.

[12] W. Bandung, "Penetapan Trayek dan Jumlah Kendaraan Penumpang Umum Dalam Setiap Trayek yang Beroperasi di Wilayah Kota Bandung," Pemerintah Kota Bandung, Bandung, 2008.

Lampiran

database.py bat_algo.py

```python
import pymysql
import pandas as pd
from pandas import ExcelWriter

class conndata:
    def connect(self):
        conn = pymysql.connect(
                host = "localhost",
                user = "root",
                password = "",
                database = "angkot"
        )
        return conn

    def table(self,table):
        con = self.connect()
        cur = con.cursor()
        sql = "SELECT * FROM "+table
        cur.execute(sql)
        data = cur.fetchall()
        con.close()
        return data

    def getnext(self, titik_a):
        tm = []
        con = self.connect()
        cur = con.cursor()
        sql = 'SELECT * FROM matrix_25_keluar WHERE titik_a =%s'
        val = titik_a
        cur.execute(sql,val)
        hasil = cur.fetchall()
        con.close()
        for data in hasil:
            titik_b = data[2]
```

CRLF UTF-8 Python GitHub Git (0)

Figure 16 Kodingan Database (1)

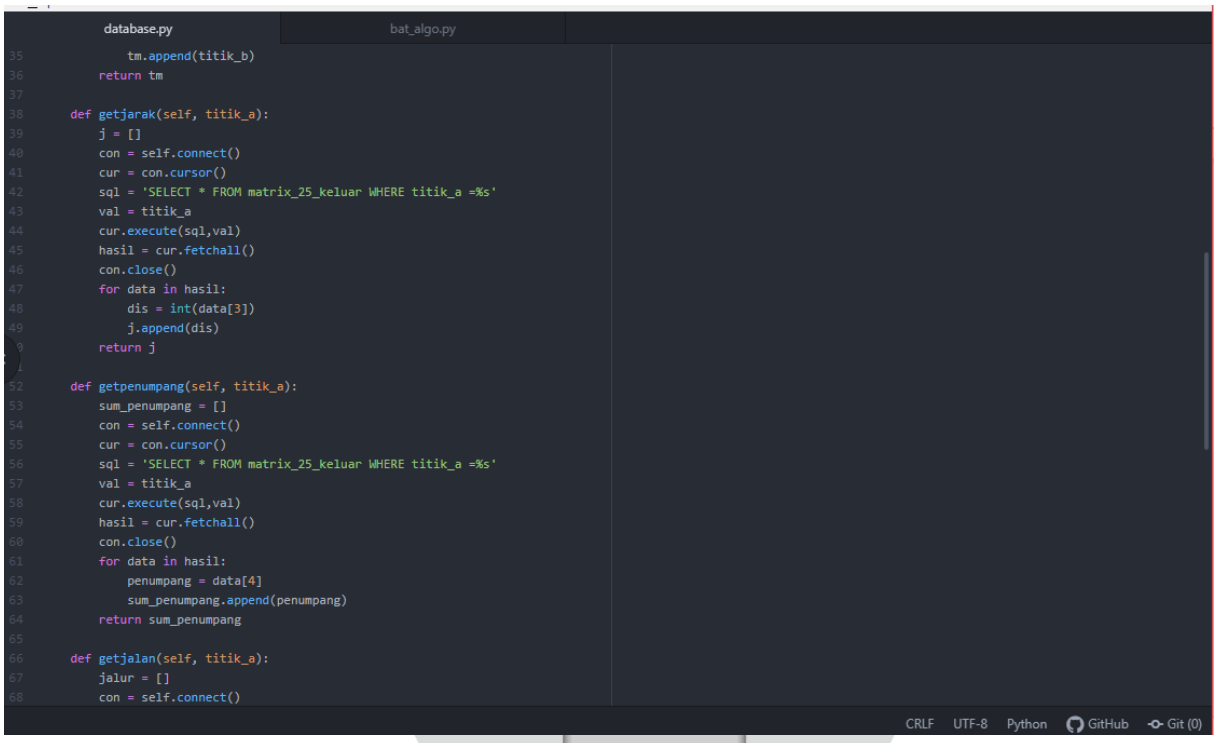


Figure 17 Kodingan Database (2)

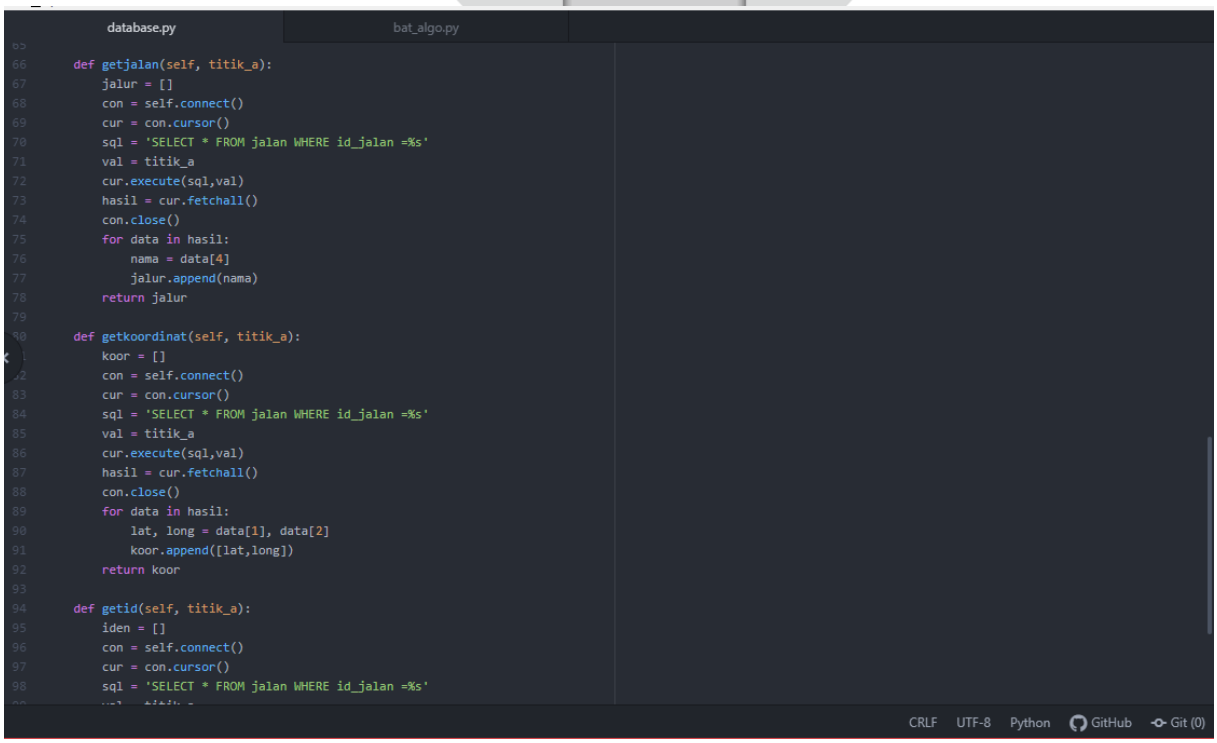


Figure 18 Kodingan Database (3)

database.py bat_algo.py

```python
import math
import random
import numpy as np
from database import conndata
import pandas as pd
from pandas import ExcelWriter

def fungsi(x, min_x, max_x):
    func_obj = (x-min_x)/(max_x-min_x)
    return func_obj


def bat_algorithm(awal, n_bat, v0, A0, r0, alpha, gamma, fmin, fmax, iterasiMax):
    #Inisialisasi awal
    db = conndata()
    jarak = db.getjarak(awal)
    penum = db.getpenumpang(awal)
    x = db.getpenumpang(awal)
    global_best2 = db.getnext(awal)
    v = [v0 for i in range(len(x))]     #Set Kecepatan Setiap Kelelawar
    f = [0.0]*len(x)                    #set Frekuensi
    A = [A0 for i in range(len(x))]     #set Loudness setiap kelelawar
    r = [r0 for i in range(len(x))]     #set  rate of emission
    iterasi = 0
    ratio = 0

    while iterasi < iterasiMax:
        fitness = [fungsi(x[i],0, 8) for i in range(len(x))] #Penumpang

        if ratio == 0:
            ratio = fitness
            local_best = x

        fitness = [i if i < j else j for i,j in zip(fitness, ratio)]
```

CRLF UTF-8 Python GitHub Git (0)

Figure 19 Kodingan Bat Algo (3)

database.py bat_algo.py

```python
    while iterasi < iterasiMax:
        fitness = [fungsi(x[i],0, 8) for i in range(len(x))] #Penumpang

        if ratio == 0:
            ratio = fitness
            local_best = x

        fitness = [i if i < j else j for i,j in zip(fitness, ratio)]
        fitness_min = min(fitness)
        local_best = [x[idx] if i < j else local_best[idx] for idx, (i,j) in enumerate(zip(fitness,ratio))]
        idx = fitness.index(fitness_min)
        global_best = local_best[idx]

        print(f'Nilai Fitness iterasi ke-{iterasi} = {fitness}')
        print(f'Nilai Fitness Terbaik = {fitness_min}')
        print(f'local position = {local_best}')
        print(f'Best Position = {global_best}')
        print('')
        print('==========================================================================')

        #update posisi dan kecepatan
        for i, (posisi, kecepatan) in enumerate(zip(x, v)):
            #definisikan frekuensi tiap kelelawar
            beta = random.uniform(0,1)
            f[i] = fmin + (fmax - fmin) * beta

            #seleksi kondisi update posisi
            rand = random.uniform(0,1)
            if (rand > r[i]):
                #by random walk
                rata2A = np.mean(A)
                epsilon = random.uniform(-1,1)
                x[i] = posisi + epsilon * rata2A
            else:
```

CRLF UTF-8 Python GitHub Git (0)

Figure 20 Kodingan Bat Algo (2)

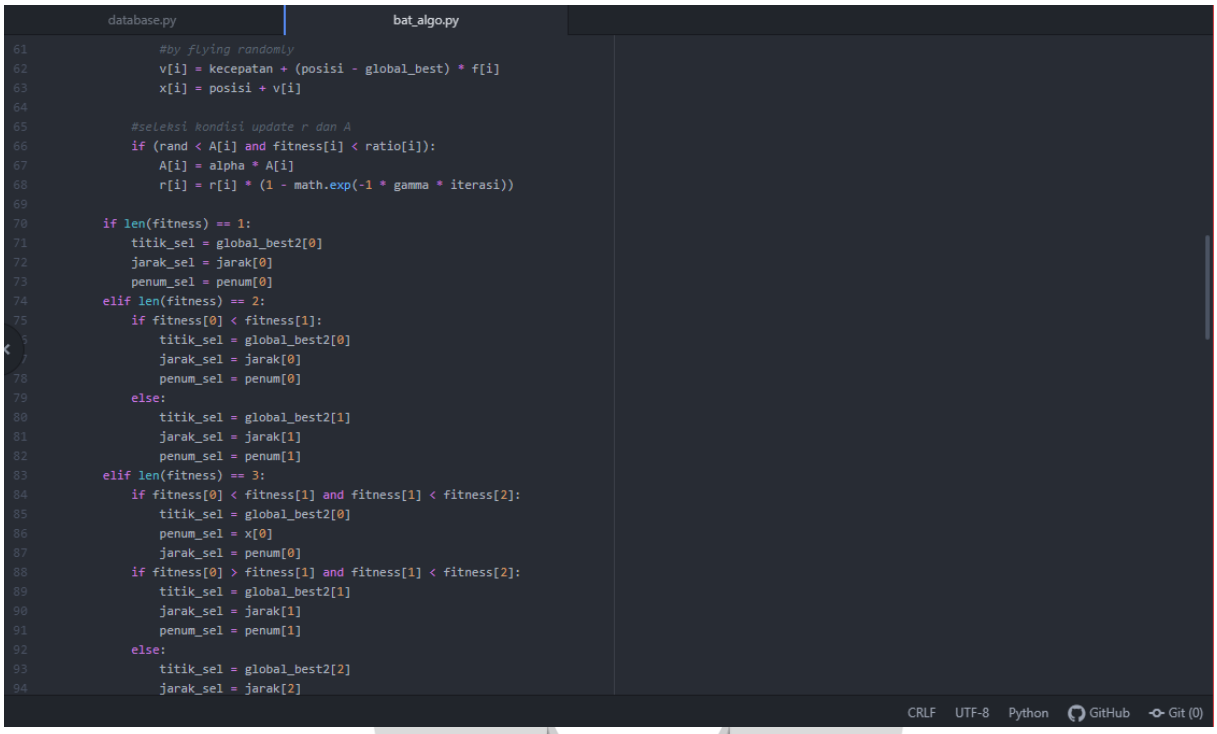


Figure 21Kodingan Bat Algo (3)

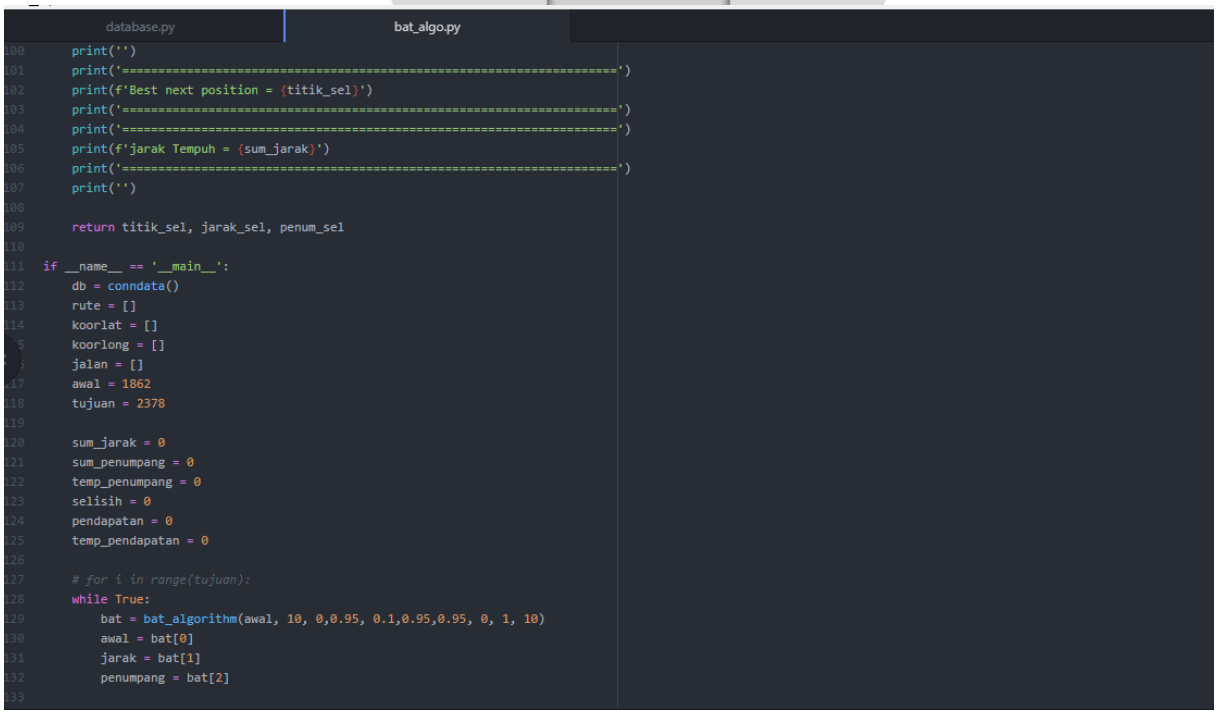


Figure 22 Kodingan Bat Algo (4)

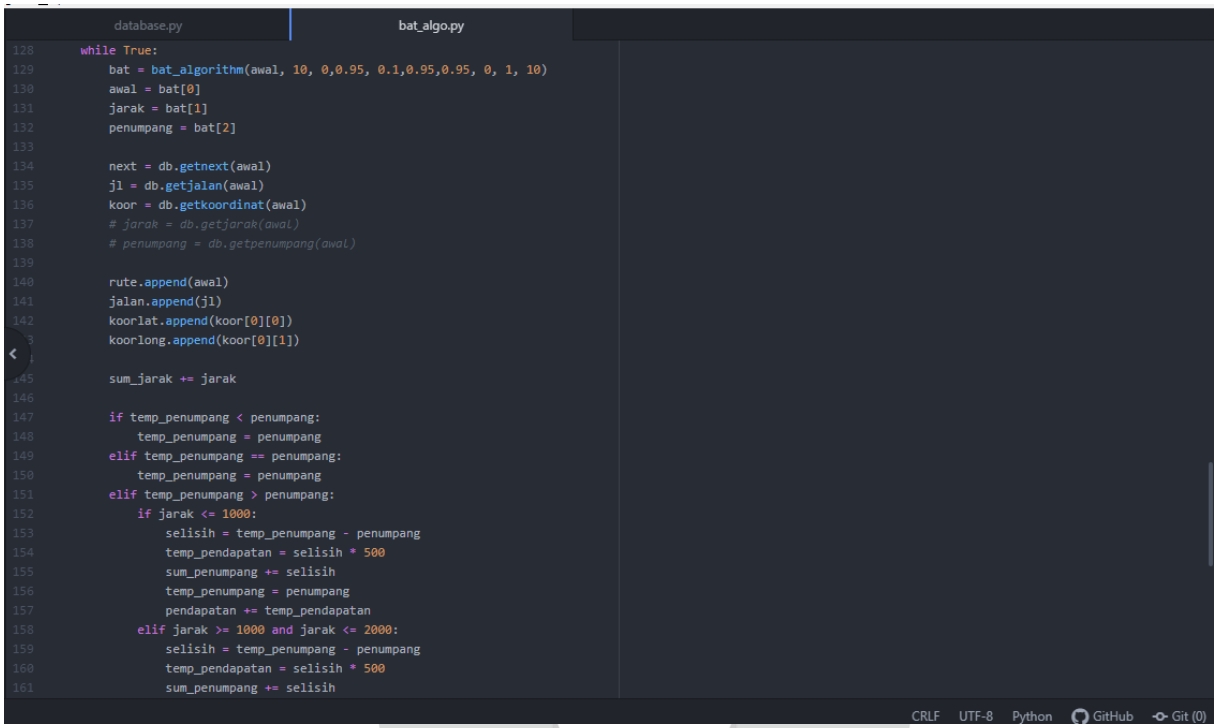


Figure 23 Kodingan Bat Algo (5)

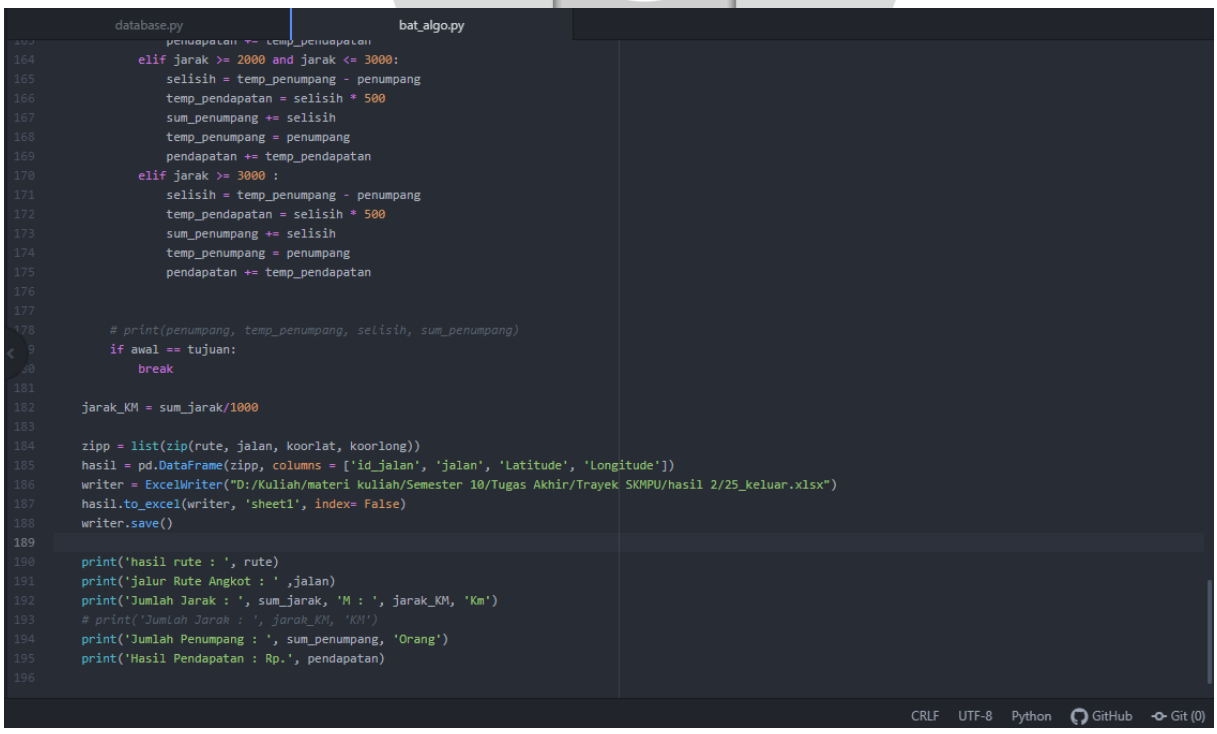


Figure 24 Kodingan Bat Algo (6)

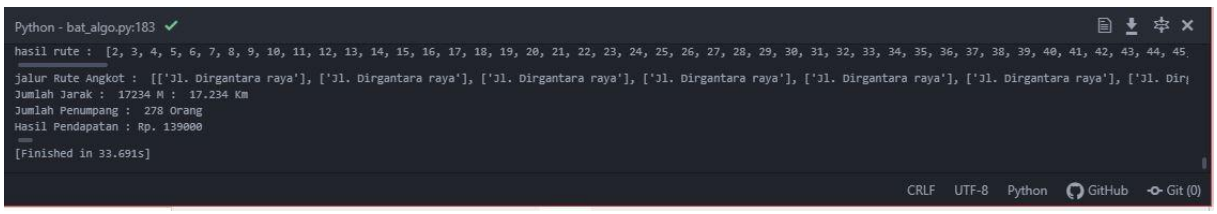
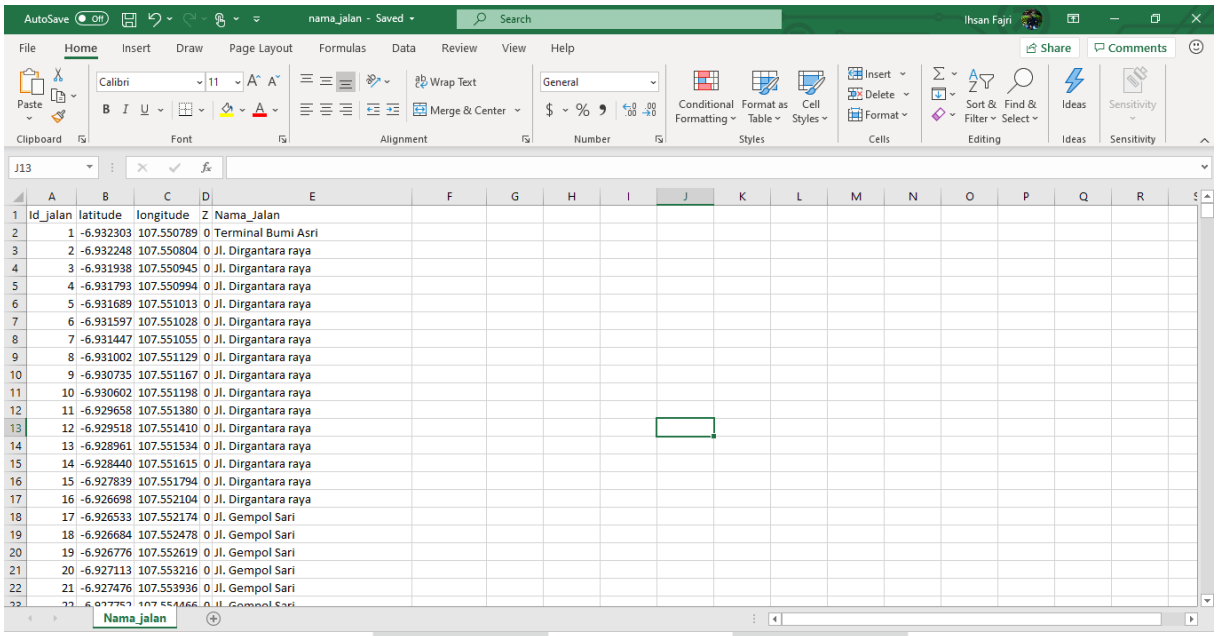
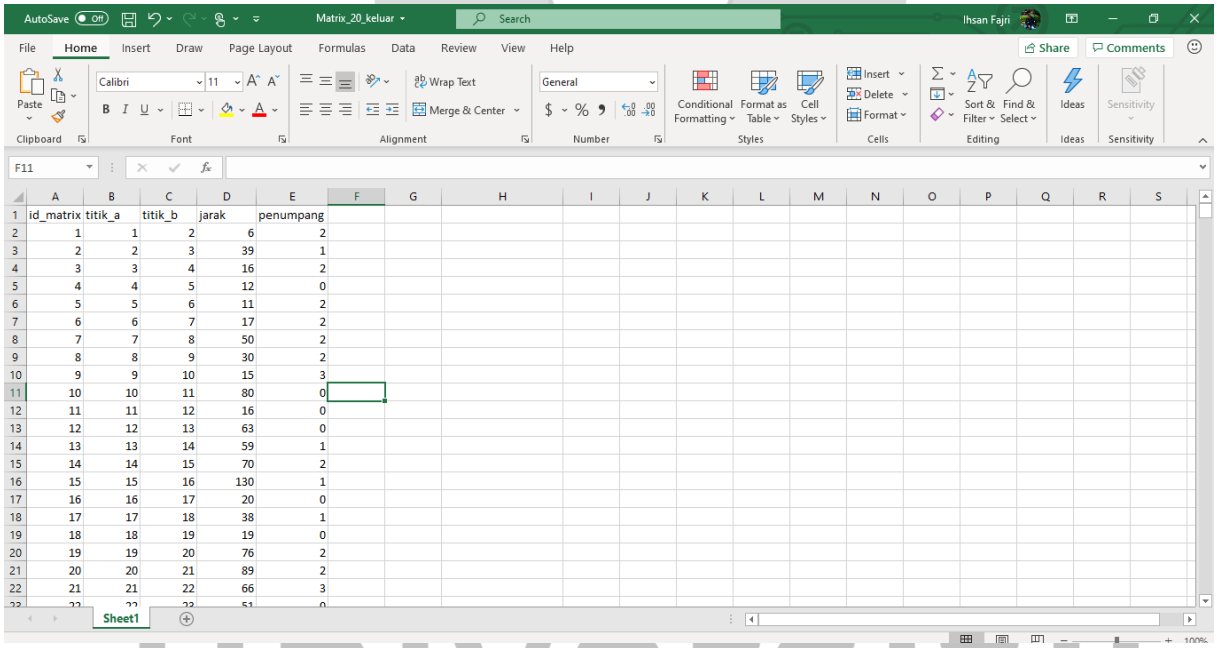


Figure 25 Hasil Run Kodingan



Id_jalan	latitude	longitude	Z	Nama_Jalan
1	-6.932303	107.550789	0	Terminal Bumi Asri
2	-6.932248	107.550804	0	Jl. Dirgantara raya
3	-6.931938	107.550945	0	Jl. Dirgantara raya
4	-6.931793	107.550994	0	Jl. Dirgantara raya
5	-6.931689	107.551013	0	Jl. Dirgantara raya
6	-6.931597	107.551028	0	Jl. Dirgantara raya
7	-6.931447	107.551055	0	Jl. Dirgantara raya
8	-6.931002	107.551129	0	Jl. Dirgantara raya
9	-6.930735	107.551167	0	Jl. Dirgantara raya
10	-6.930602	107.551198	0	Jl. Dirgantara raya
11	-6.929658	107.551380	0	Jl. Dirgantara raya
12	-6.929518	107.551410	0	Jl. Dirgantara raya
13	-6.928961	107.551534	0	Jl. Dirgantara raya
14	-6.928440	107.551615	0	Jl. Dirgantara raya
15	-6.927839	107.551794	0	Jl. Dirgantara raya
16	-6.926698	107.552104	0	Jl. Dirgantara raya
17	-6.926533	107.552174	0	Jl. Gempol Sari
18	-6.926684	107.552478	0	Jl. Gempol Sari
19	-6.926776	107.552619	0	Jl. Gempol Sari
20	-6.927113	107.553216	0	Jl. Gempol Sari
21	-6.927476	107.553936	0	Jl. Gempol Sari

Figure 26 Data Jalan



id_matrix	titik_a	titik_b	jarak	penumpang
1	1	2	6	2
2	2	3	39	1
3	3	4	16	2
4	4	5	12	0
5	5	6	11	2
6	6	7	17	2
7	7	8	50	2
8	8	9	30	2
9	9	10	15	3
10	10	11	80	0
11	11	12	16	0
12	12	13	63	0
13	13	14	59	1
14	14	15	70	2
15	15	16	130	1
16	16	17	20	0
17	17	18	38	1
18	18	19	19	0
19	19	20	76	2
20	20	21	89	2
21	21	22	66	3

Figure 27 Data Matrix

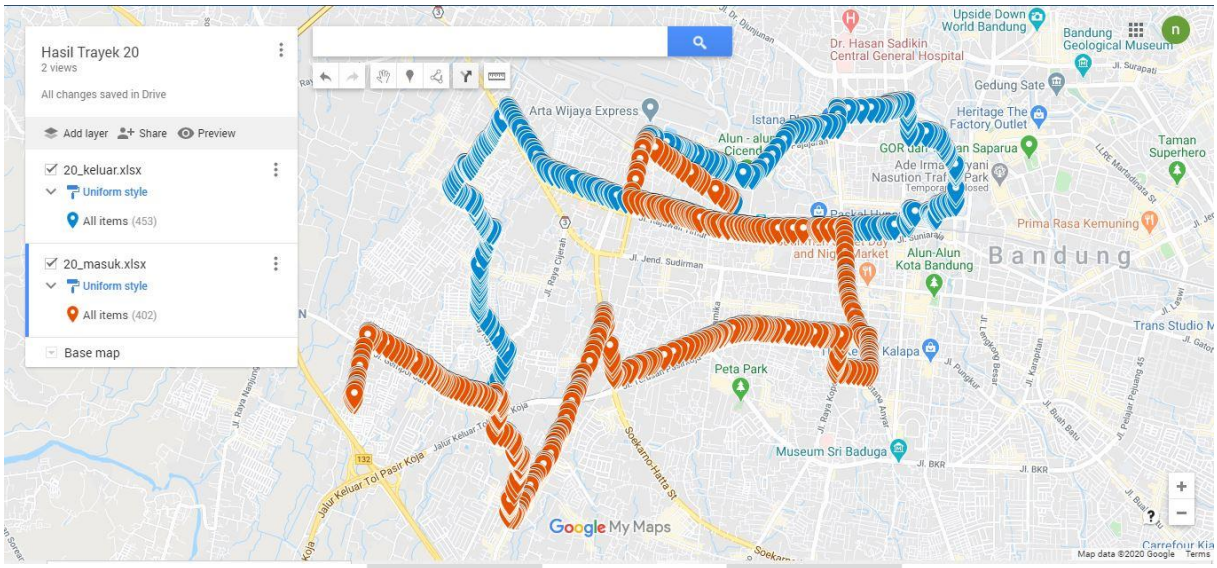


Figure 28 Hasil Rute Trayek 20

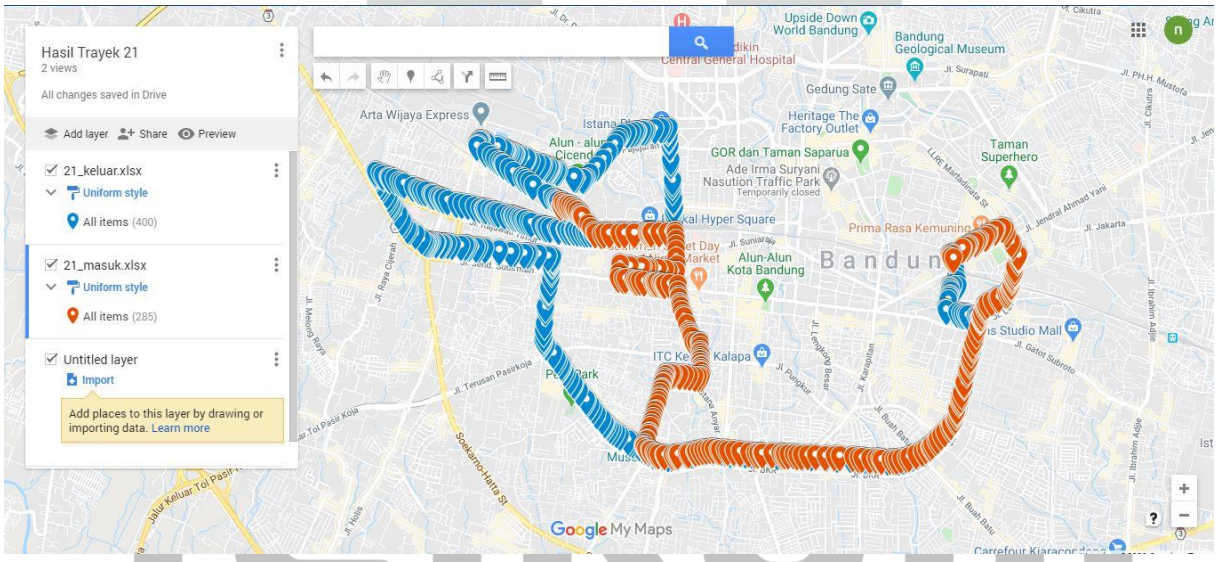


Figure 29 Hasil Trayek 21

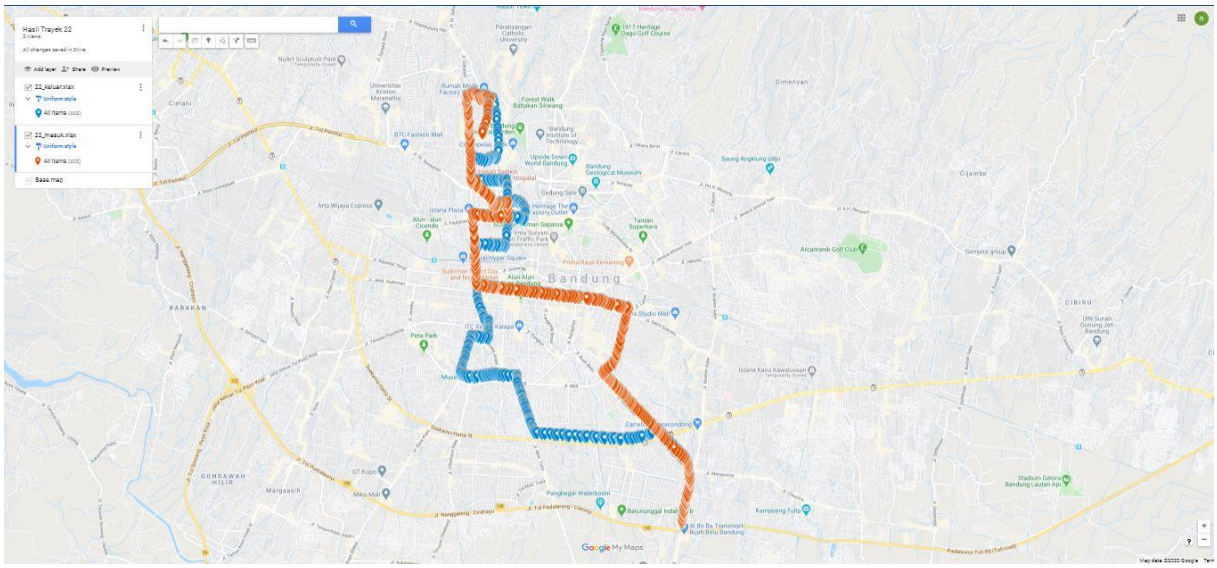


Figure 30 Hasil Trayek 22

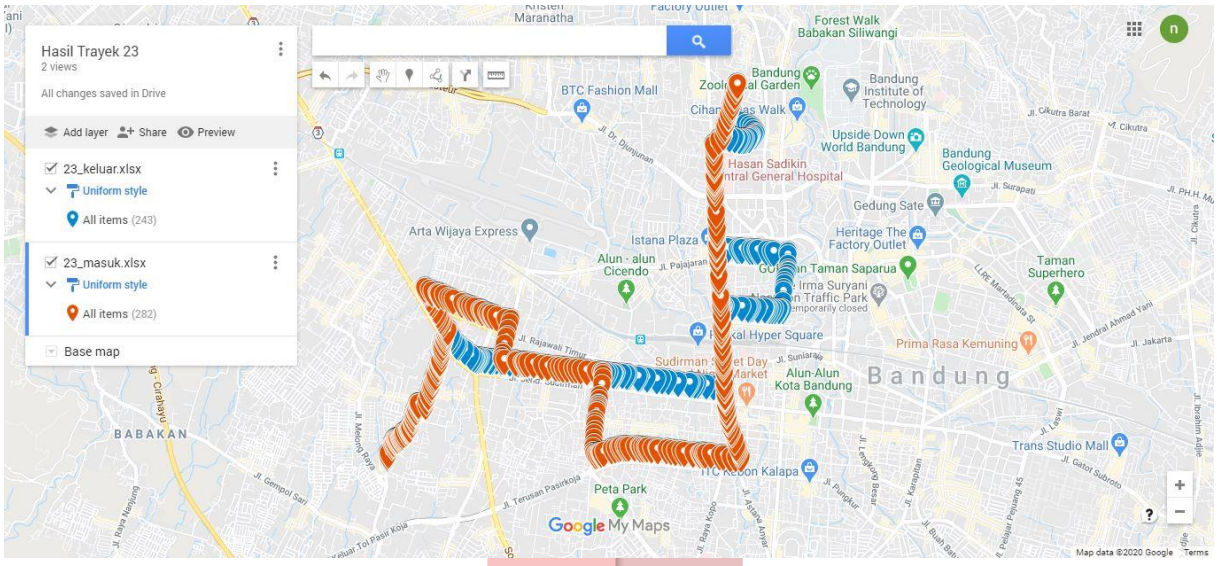


Figure 31 Hasil Trayek 23

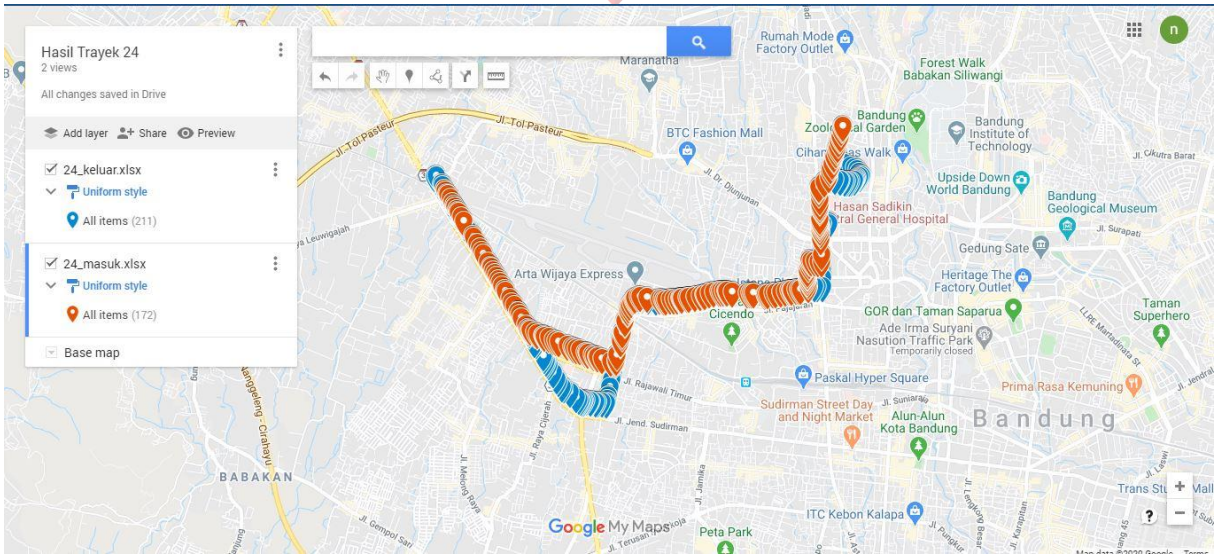


Figure 32 Hasil Trayek 24

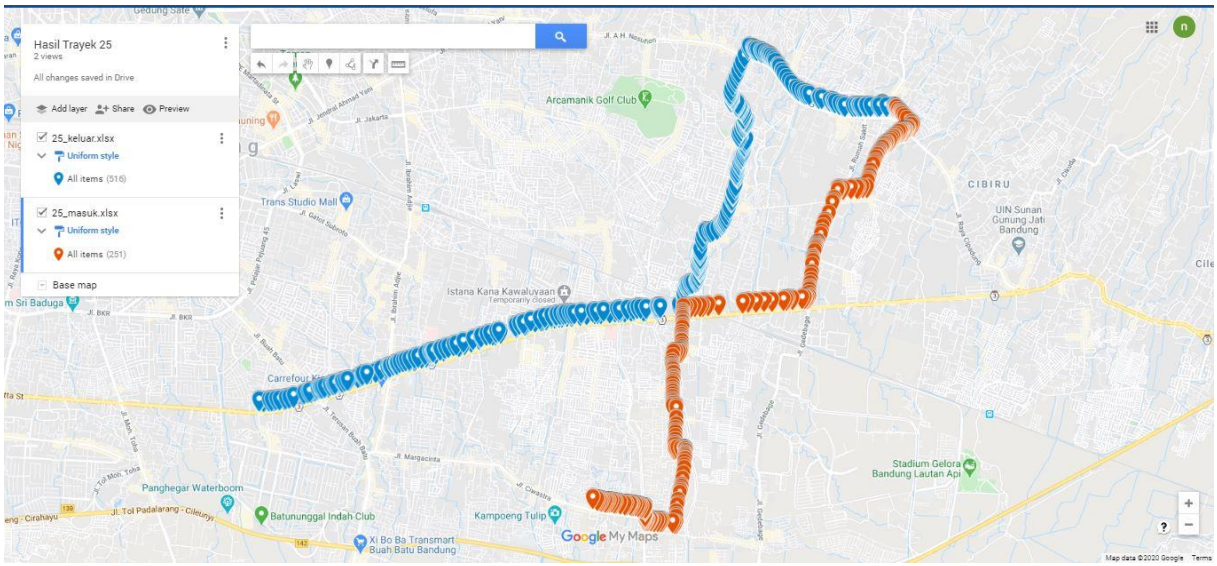


Figure 33 Hasil Trayek 25