

Simulasi *Online Taxi's Dispatch System* Dengan Metode *Combinatorial Optimization*

Vincent Christian Andoko, Andreas Handoyo, Henry Novianus Palit
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658

E-Mail: vincentchran@gmail.com, handoyo@petra.ac.id, hnpalit@petra.ac.id

ABSTRAK

Taksi merupakan salah satu bagian penting dalam kehidupan masyarakat saat ini. Taksi berfungsi sebagai salah satu sarana transportasi umum yang sangat berperan melayani masyarakat dalam mencapai tempat tujuan yang diinginkan dan mudah dijumpai sebab layanan taksi tersedia di sebagian besar kota di seluruh dunia. Mengikuti perkembangan teknologi, munculah taksi *online*. Taksi *online* merupakan sarana transportasi umum yang menggunakan aplikasi berbasis *web* atau *mobile* untuk menghubungkan penumpang dengan taksi. Taksi *online* memberi kemudahan bagi penumpang dan taksi untuk mengetahui lokasi tujuan melalui peta *online* dengan GPS atau nama dari tujuan yang diinginkan serta menampilkan faktor penting seperti jarak tempuh dan durasi dengan cepat.

Dibalik fitur-fitur yang disajikan oleh taksi *online* tentunya dibutuhkan metode *dispatch* taksi yang optimal. Kondisi jalan di setiap daerah tentunya berbeda-beda contohnya seperti kemacetan pada tengah kota berbeda dengan kemacetan di pinggir pinggir kota. Dari kondisi tersebut maka tidak mungkin hanya menggunakan satu jenis cara untuk proses *dispatch* taksi pada seluruh daerah tersebut. Dengan menggunakan simulasi yang memiliki beberapa metode, dapat ditentukan cara mana yang optimal untuk daerah yang ditentukan dengan memberikan gambaran proses *dispatch* oleh taksi beserta perbandingan hasil dari cara-cara yang digunakan pada simulasi tersebut.

Hasil akhir dari program ini yaitu suatu simulasi yang dapat menjalankan 3 jenis algoritma untuk memasangkan taksi dan penumpang berdasarkan 3 faktor yaitu durasi, jarak, dan total penumpang yang sudah dilayani setiap taksi. Simulasi ini dapat memberikan perbandingan dari hasil ketiga algoritma dan perhitungan faktor yang digunakan pada skenario yang sama untuk mempermudah pengguna menentukan algoritma dan rumus perhitungan faktor yang sesuai pada daerah dan waktu tertentu.

Kata Kunci: *hungarian algorithm, first come first served, random, haversine formula, google maps api, statis, dinamis, simulasi*

ABSTRACT

Taxis are an important part of people's lives today. Taxis function as one of the means of public transportation that plays a role in serving the community in reaching the desired destination and is easily found because taxi services are available in most cities throughout the world. Following the development of technology, online taxis emerged. Online taxis are a means of public transportation using web or mobile based applications to connect passengers with taxis. Online taxis make it easy for passengers and taxis to find out the destination via an online map with GPS or the name of the desired destination and display important factors such as distance and duration quickly.

Behind the features presented by online taxis, an optimal taxi dispatch method is needed. The road conditions in each area are different, for example traffic jams in the middle of the city are different from the traffic jams on the edge of the city. From these conditions it is not possible to use only one type of method for the taxi dispatch process in the entire area. By using simulations that have several methods, it can be determined which method is optimal for the area determined by giving an overview of the dispatch process by taxi along with the comparison of the results of the methods used in the simulation.

The final result of this program is a simulation that can run 3 types of algorithms to pair taxis and passengers based on 3 factors, namely the duration, distance, and total passengers that each taxi has served. This simulation can provide a comparison of the results of the three algorithms and the calculation of the factors used in the same scenario to make it easier for users to determine the algorithm and calculation's formula for the appropriate factors in a particular region and time.

Keywords: *hungarian algorithm, first come first served, random, haversine formula, google maps api, static, dynamic, simulation*

1. PENDAHULUAN

Taksi merupakan salah satu bagian penting dalam kehidupan masyarakat saat ini. Taksi berfungsi sebagai salah satu sarana transportasi umum yang sangat berperan melayani masyarakat dalam mencapai tempat tujuan yang diinginkan dan mudah dijumpai sebab layanan taksi tersedia di sebagian besar kota di seluruh dunia. Pasar taksi menjadi hal yang cukup penting di kota-kota, di mana pangsa mobilitas berkisar antara 1% dan 15% pada awal abad ke-20 [3]. Dalam 10 tahun terakhir sektor layanan taksi ini telah berubah secara signifikan oleh kemajuan teknologi.

Pada awal abad ke-20 sebagian besar taksi menjadikan *dispatching center* sebagai sarana komunikasi dengan penumpang, namun dengan pada akhir abad ke-20 dengan adanya *smartphone* dan kecepatan internet yang mumpuni, penumpang dapat berkomunikasi secara langsung dengan supir taksi tanpa melalui *dispatching center*. Mengikuti perkembangan teknologi, munculah taksi *online*.

Taksi *online* merupakan sarana transportasi umum yang menggunakan aplikasi berbasis *web* atau *mobile* untuk menghubungkan penumpang dengan taksi. Taksi *online* memberi kemudahan bagi penumpang dan taksi untuk mengetahui lokasi tujuan melalui peta *online* dengan GPS atau nama dari tujuan yang diinginkan serta menampilkan faktor penting seperti jarak tempuh dan durasi dengan cepat.

Dibalik fitur-fitur yang disajikan oleh taksi *online* tentunya dibutuhkan metode *dispatch* taksi yang optimal. Kondisi jalan di

setiap daerah tentunya berbeda-beda contohnya seperti kemacetan pada tengah kota berbeda dengan kemacetan di pinggir pinggir kota. Dari kondisi tersebut maka tidak mungkin hanya menggunakan satu jenis cara untuk proses dispatch taksi pada seluruh daerah tersebut. Dengan menggunakan simulasi yang memiliki beberapa metode, dapat ditentukan cara mana yang optimal untuk daerah yang ditentukan dengan memberikan gambaran proses dispatch oleh taksi beserta perbandingan hasil dari cara-cara yang digunakan pada simulasi tersebut.

2. DASAR TEORI

2.1 Combinatorial Optimization

Digunakan untuk menghasilkan rangkaian uji coba yang teroptimisasi dengan *me-sampling* konfigurasi *input* [1]. *Combinatorial optimization* bertujuan untuk mencari objek yang optimal dari jumlah objek yang terbatas (*finite*). *Combinatorial optimization* merupakan bagian dari optimisasi matematika yang berhubungan untuk *research*, teori algoritma, dan komputasi teori kompleks.

2.2 First Come First Serve(FCFS)

Merupakan metode yang sederhana yang dapat digunakan pada berbagai macam permasalahan [8]. Algoritma dasar yang digunakan untuk metode *first come first serve* adalah *first in first out*(FIFO) sehingga metode ini memiliki kelemahan bila ada antrian yang *stuck* maka proses tidak akan bisa lanjut.

2.3 Hungarian Algorithm

Hungarian algorithm merupakan salah satu *metode combinatorial optimization* yang bertujuan untuk memecahkan masalah penempatan atau pemasangan dari 2 jenis objek dengan mencari nilai optimal dari *variable* yang menghubungkan 2 jenis objek tersebut. *Hungarian algorithm* merupakan algoritma untuk *assignment problem* yang optimal dengan menghasilkan pemasangan *variable* dengan sumber daya yang paling kecil [5].

2.4 Mersenne Twister Random

Mersenne Twister merupakan *pseudorandom number generator* (PRNG) dan merupakan algoritma yang umum digunakan untuk menghasilkan nilai *random*. Salah satu penggunaan algoritma *mersennet twister* adalah PHP dengan fungsi *mt_rand()* [7].

2.5 Haversine Formula

Rumus *haversine* adalah persamaan penting dalam navigasi, memberikan jarak lingkaran besar antara dua titik pada bola / *sphere* dari garis bujur(*longitude*) dan garis lintang(*latitude*). [4]

$$d = 2r \arcsin\left(\sqrt{\text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)}\right)$$

$$= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

- d adalah jarak antara dua titik pada permukaan bola / sphere.
- r adalah jari-jari bola.
- φ_1 adalah latitude titik 1.
- φ_2 adalah latitude titik 2.
- λ_1 adalah longitude titik 1.
- λ_2 adalah longitude titik 2.

2.6 Zend Engine

Untuk waktu yang lama, bahasa pemrograman PHP tidak dianggap sebagai bahasa yang serius untuk membuat aplikasi *web* yang besar. PHP dulu dikenal sebagai bahasa pemrograman bagus untuk pembuatan aplikasi *web* yang kecil saja, namun akhir-akhir ini cara pandang *programmer* terhadap PHP berubah dengan adanya 3 *framework* PHP yang memperluas kemampuan PHP yaitu *Symfony*, *CakePHP*, dan *Zend Engine* [6]. *Zend Engine* adalah pusat dari PHP, dan merupakan komponen yang *me-parsing* dan menjalankan semua file PHP [9].

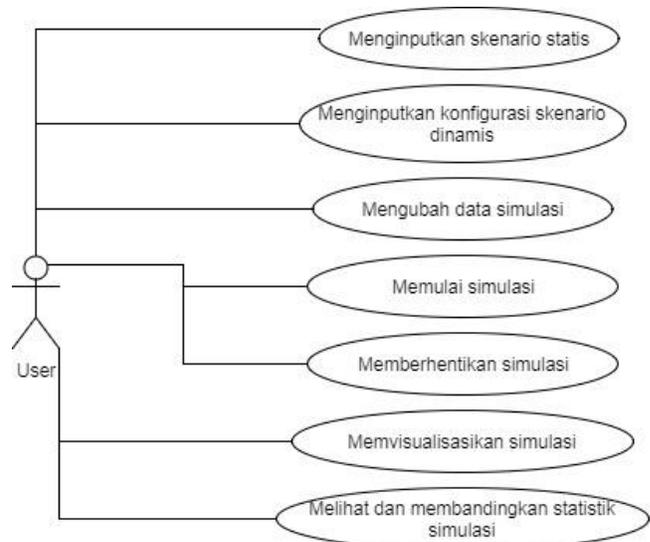
2.7 Google Maps Web Service API

Google maps web service API merupakan *Application Programming Interface* untuk mengimplementasikan *google maps* ke dalam halaman *web* atau mengambil data dari *google maps* seperti *traffic jam* yang disediakan oleh *google* untuk *developer* dan memungkinkan untuk penggunaan sederhana atau ekstensif [2].

3. DESAIN SISTEM

3.1 Use Case Diagram

Pada *usecase diagram* Gambar 1 menunjukkan hanya terdapat 1 aktor yaitu *user*. *User* mengatur segala hal yang ada pada simulasi ini. *User* memiliki aktivitas-aktivitas yaitu *me-input* skenario statis, *me-input* ketentuan skenario dinamis, memulai simulasi, memberhentikan simulasi, melanjutkan simulasi, melihat statistik, dan visualisasi simulasi.



Gambar 1. Use Case Diagram

3.2 Perhitungan Faktor dengan Algoritma dan Rumus yang Ditentukan

Proses ini menunjukkan sekilas penggunaan rumus dan cara kerja dari algoritma yang digunakan. Untuk rumus yang ditentukan *user* akan diterima berupa *string* dan ditranslasi oleh program ke variabel dan digunakan untuk perhitungan faktor setelah normalisasi.

Pada *Hungarian algorithm*, matriks akan dilakukan pengurangan nilai pada setiap elemennya dengan ketentuan nilai pada baris A – nilai terkecil yang ada di baris A kemudian nilai pada kolom A dikurangi nilai pada kolom A. Bila semua baris dan kolom memiliki nilai 0 maka proses selesai, namun bila ada baris atau kolom yang

tidak memiliki nilai 0 maka akan dilakukan optimisasi *Hungarian* untuk mendapatkan nilai *assignment* yang optimal.

Untuk proses dengan algoritma MVSF (*Minimum Value Of Selected Factor*) menggunakan algoritma *first come first server* dan terdapat *batch* untuk mengumpulkan data terlebih dahulu.

Pada algoritma terakhir yaitu *random* digunakan algoritma *mersenne twister* untuk menghasilkan angka acak. Algoritma terakhir ini digunakan sebagai pembanding apakah 2 algoritma sebelumnya memiliki manfaat atau dampak yang bisa dilihat atau tidak.

Persamaan normalisasi *min max* dapat dilihat dibawah.

$$v' = \frac{v - \min}{\max - \min} (\text{new_max} - \text{new_min}) + \text{new_min}$$

3.3 Insert Data Pemasangan Taksi dan Penumpang Ke Dalam Database

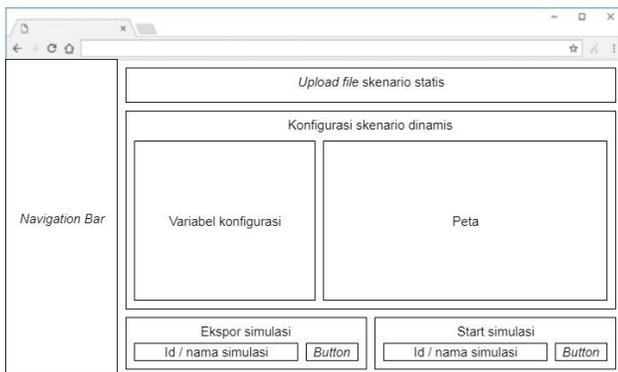
Setelah data sudah dikalkulasi dengan rumus dan algoritma yang ditentukan, berikutnya data tersebut dimasukkan ke dalam *database* kemudian status *database driver* dan *customer* pada simulasi yang dijalankan diubah ke 0 yang artinya *driver* dan *customer* telah dipasangkan atau dijalankan di dalam simulasi.

3.4 Halaman Home

Di halaman home Gambar 2 berisikan bagan untuk *me-input* skenario statis, konfigurasi skenario dinamis, ekspor skenario dan memulai simulasi.

Untuk bagian skenario statis disediakan *form* untuk menentukan lokasi *file* statis beserta tombol untuk *upload* data dari *file* tersebut. Pada bagian konfigurasi *file* dinamis di bagian kiri terdapat *form* untuk *me-inputkan* variabel konfigurasi dan peta pada bagian kanannya untuk mempermudah menentukan *latitude* dan *longitude* yang akan di-*input* ke variabel konfigurasi.

Pada bagian ekspor simulasi disediakan *list* simulasi yang dapat dipilih untuk diekspor. Sedangkan *list* simulasi untuk dijalankan terdapat pada sebelah kanan ekspor simulasi. Pada bagian *navigation bar* terdapat *link* untuk mengarahkan ke halaman *visualization* dan halaman statistik.

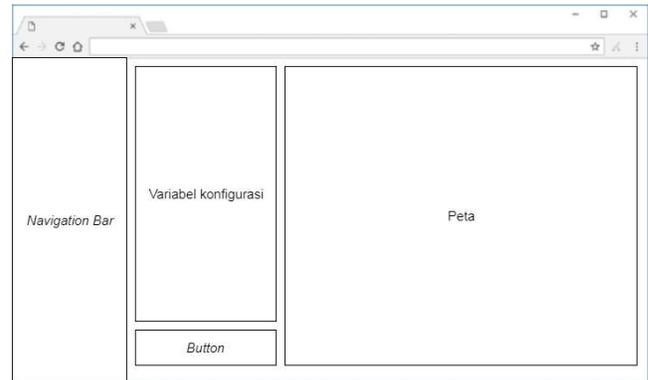


Gambar 2. Tampilan Halaman Home

3.5 Halaman Add Generate Place

Halaman *Add Generate Place* Gambar 3 digunakan untuk menambahkan lokasi *spawn* taksi dan penumpang beserta variabel-variabel yang dibutuhkan. Terdapat 2 bagian utama yaitu *form*

untuk *me-input* variabel konfigurasi dan peta. Pada bagian *navigation bar* terdapat *link* untuk mengarahkan ke halaman *home*, halaman *visualization*, dan halaman statistik.



Gambar 3. Tampilan Halaman Add Generate Place

3.6 Halaman Edit Simulation

Halaman *edit simulation* Gambar 4 berfungsi untuk mengubah variabel konfigurasi simulasi yang belum dijalankan. Dan menghapus data *generate place* bila tidak diperlukan.



Gambar 4. Tampilan Halaman Edit Simulation

3.7 Halaman Proses Kalkulasi

Pada halaman proses kalkulasi Gambar 5 menunjukkan statistik, tabel informasi sementara. Pada halaman ini juga proses simulasi akan berjalan. Terdapat tombol untuk memberhentikan simulasi. Terdapat pula tombol-tombol lain untuk kembali ke halaman *home* bila tombol-tombol tersebut ditekan maka simulasi akan secara otomatis berhenti dan diarahkan ke halaman yang diinginkan.

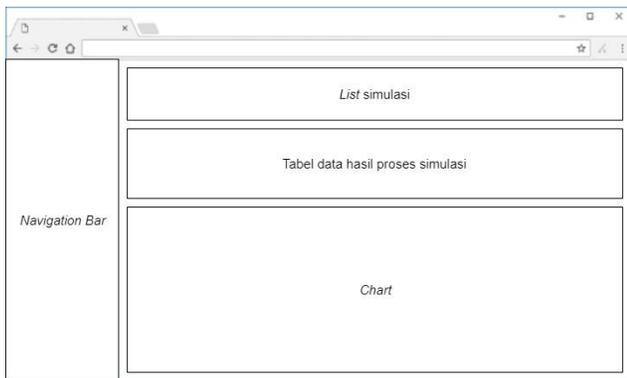


Gambar 5. Halaman Proses Kalkulasi

Pada bagian informasi 1 terdapat informasi lama simulasi berjalan dan waktu dijalkannya. Di bagian informasi 2 terdapat statistik untuk menunjukkan jumlah keberhasilan dan jumlah pembatalan. Terdapat *line chart* dibawah informasi 1 untuk menunjukkan peningkatan dan penurunan faktor dari 20 event terakhir. Dan pada bagian tabel *event* statistik akan ditampilkan semua *event* yang terjadi selama proses kalkulasi berlangsung.

3.8 Halaman Statistik

Halaman statistik Gambar 6 berfungsi untuk menunjukkan statistik dari hasil simulasi sekaligus membandingkannya dengan statistik dari hasil simulasi lain hingga maksimal 5 statistik yang ditampilkan. Pada *navigation bar* terdapat *link* untuk mengarahkan ke halaman *home* dan halaman *visualization*

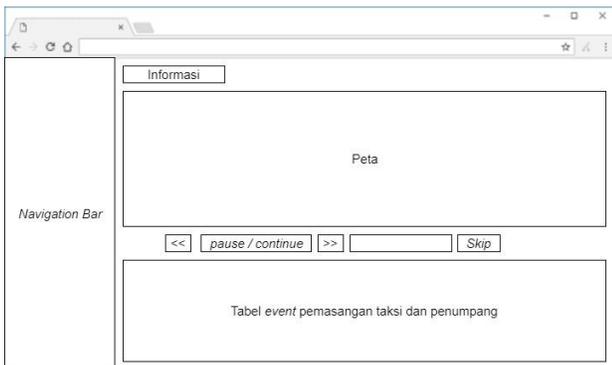


Gambar 6. Tampilan Halaman Statistik

3.9 Halaman Visualization

Halaman *visualization* Gambar 7 bertujuan untuk memberikan gambaran bagaimana proses simulasi berjalan dengan menggunakan *google maps web service API* beserta tabel informasinya untuk memberikan informasi pemasangan *driver* dengan *customer*.

User dapat melihat gambaran dari kejadian proses simulasi melalui fitur ini. *User* dapat melihat list pemasangan taksi dan penumpang beserta pembatalannya dari tabel yang tersedia di bagian bawah halaman. Untuk menggunakan fitur ini, user diwajibkan memilih simulasi yang akan divisualisasikan terlebih dahulu. Ketika fitur ini berjalan, *user* dapat mempercepat simulasi ke 10 detik kedepan dengan tombol “>>”, kembali ke 10 detik yang lalu dengan tombol “<<”, dan memberhentikan atau menjalankan simulasi dengan tombol “*pause/continue*” yang dapat berubah namanya sesuai dengan status jalannya simulasi.



Gambar 7. Tampilan Halaman Visualization

User juga dapat *skip* ke detik yang diinginkan dengan memasukan detik ke *field input* dan menekan tombol *skip*. Untuk pindah ke simulasi lain dapat dilakukan dengan menekan tombol “*select simulation*” dan *user* akan diarahkan ke halaman untuk memilih simulasi.

Di bagian bawah tombol kontrol terdapat tabel *event* yang menampilkan informasi pemasangan taksi dan penumpang, sisa durasi tempuh, dan sisa waktu tempuh yang dapat digunakan oleh *user* untuk memantau jalannya simulasi yang sedang dijalankan. Pada bagian *navigation bar* terdapat *link* untuk mengarahkan ke halaman *home* dan halaman *visualization*.

4. IMPLEMENTASI SISTEM

4.1 XAMPP

Simulasi ini membutuhkan *script* PHP untuk *back end*, *apache* untuk mengakses *script file* PHP, dan *mysql* untuk menyimpan data hasil simulasi, maka diperlukan installasi PHP, *apache*, dan *mysql* terlebih dahulu.

Terdapat XAMPP yang merupakan paket *cross-platform web server solution stack* yang bersifat *free / open-source* yang dibuat oleh *apache friends*, dengan menginstall XAMPP ini kita dapat menginstall PHP, *apache*, dan *mysql* dengan *phpMyAdmin* untuk *tools administrator mysql*.

4.2 Implementasi Program pada Modul User

Simulasi ini hanya memiliki 1 modul yaitu *user*. *User* mengatur segala hal dari simulasi ini dari semua bagian *input*, proses, dan *output*.

5. PENGUJIAN SISTEM

5.1 Pengujian Sistem Simulasi

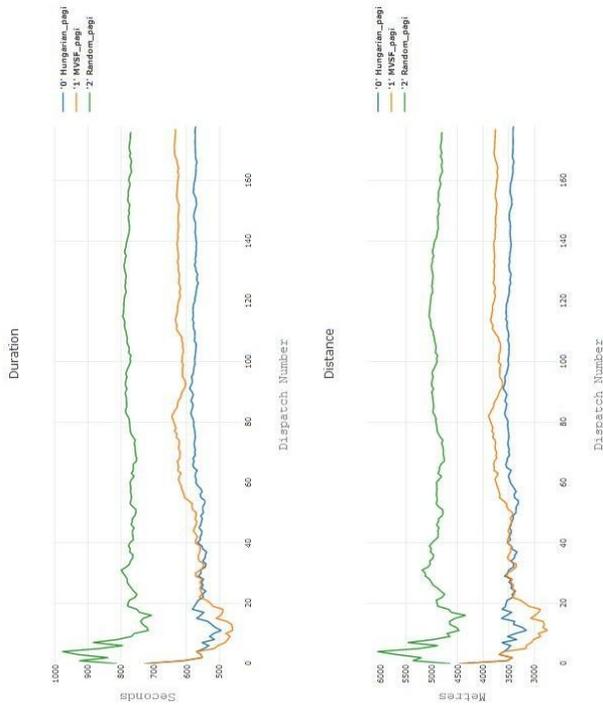
Pengujian dilakukan dengan menggunakan skenario statis yang diekspor dari hasil *generate* taksi dan penumpang pada skenario dinamis. Uji coba dilakukan sebanyak 3 kali dengan skenario yang dijalankan pada 3 jenis algoritma yang berbeda dengan konfigurasi yang sama untuk semua uji coba. Simulasi dijalankan pada waktu dan hari yang berbeda.

Konfigurasi simulasi:

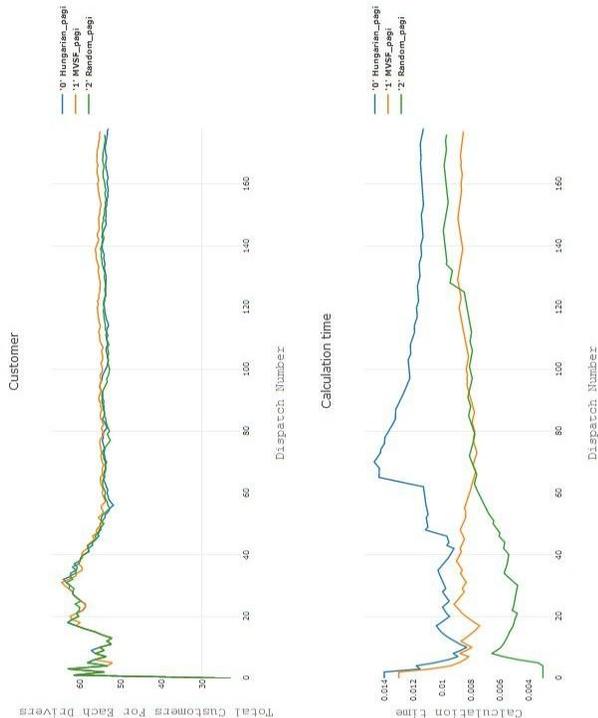
- Rumus faktor : durasi + jarak + total penumpang yang telah dilayani oleh taksi.
- Tingkat kesabaran taksi(detik): 100 – 1000
- Tingkat kesabaran penumpang(detik): 100 – 1000
- Probabilitas pembatalan oleh sisi taksi: 25%
- Probabilitas pembatalan oleh sisi penumpang: 25%
- Durasi *batch* 10 detik.

5.2 Uji Coba 1 - 3

Dari hasil uji coba 1 hingga 3 dapat disimpulkan algoritma *Hungarian* unggul dalam segi rata-rata faktor dengan nilai terendah. Walaupun memiliki rata-rata nilai faktor terendah, algoritma *Hungarian* memiliki total keberhasilan dibawah kedua algoritma lainnya pada 2 simulasi yang dijalankan dalam 2 dari 3 simulasi dan untuk memastikan hasil uji coba 1 - 3 dilakukan pengujian dengan konfigurasi yang sama dengan uji coba 1 - 3 namun dengan durasi *batch* yang lebih lama yaitu 30 detik sebanyak 3 kali (pagi, siang, dan malam) dan 50 detik sebanyak 3 kali (pagi, siang, dan malam) hasil tersebut dilihat dari Gambar 8 dan Gambar 9.



Gambar 8. Line Chart Average Data Value for Each Event Durasi dan Jarak Hasil Uji Coba



Gambar 9. Line Chart Average Data Value for Each Event Total Penumpang yang Sudah Dilayani Tiap Taksi dan Waktu Kalkulasi Hasil Uji Coba

5.3 Uji Coba 4 – 6 (30 detik)

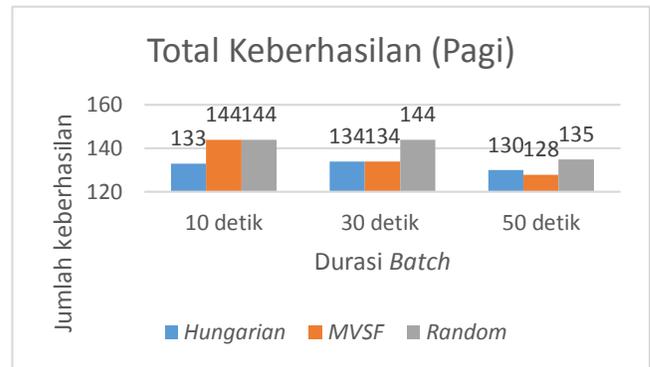
Dari hasil uji coba 4 hingga 6, perkembangan durasi dan jarak sama seperti uji coba 1 hingga 3 yaitu algoritma *Hungarian* yang paling efisien, diurutkan ke-2 algoritma MVSF dan yang terakhir algoritma *random*. Namun pada hasil total keberhasilan terdapat perbedaan yang cukup signifikan pada malam hari. Pada uji coba 6 malam hari dengan durasi *batch* 10 detik, algoritma *Hungarian* memperoleh nilai total keberhasilan paling kecil dan cukup signifikan dibawah 2 algoritma lainnya, namun pada uji coba 9 malam hari dengan durasi *batch* 30 detik algoritma *Hungarian* memperoleh nilai total keberhasilan 2 diatas MVSF dan untuk algoritma *random* memperoleh nilai yang cukup jauh jaraknya dengan 2 algoritma lainnya dengan nilai total keberhasilan 146.

5.4 Uji Coba 7 – 9 (50 detik)

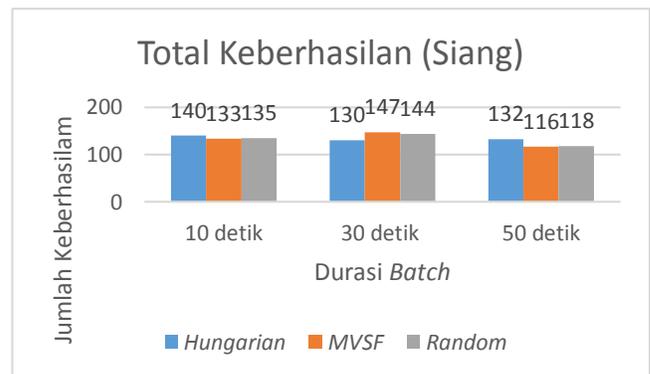
Dari hasil uji coba 6 hingga 9 total *batch* sedikit berkurang dibanding uji coba sebelumnya karena taksi dan penumpang banyak yang habis waktu kesabaran menunggu dikarenakan durasi *batch* yang lebih lama. Pada uji coba 7 siang hari algoritma *random* memperlihatkan hasil yang paling baik dalam nilai total keberhasilan. Dan untuk uji coba 8 dan 9 algoritma *Hungarian* memperlihatkan hasil yang paling baik dengan nilai total keberhasilan 132 dan 133.

5.5 Analisa Algoritma pada Uji Coba 1 – 9

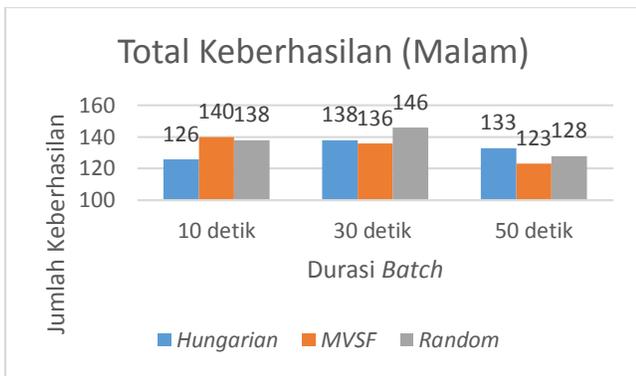
Rata-rata perkembangan Gambar 10, Gambar 11, dan Gambar 12 nilai durasi dan jarak dari semua uji coba algoritma *Hungarian* selalu memiliki nilai terendah, algoritma MVSF berada diatas *Hungarian*, dan algoritma *random* memiliki nilai rata-rata durasi dan jarak tertinggi.



Gambar 10. Bar Chart Total Keberhasilan Pagi



Gambar 11. Bar Chart Total Keberhasilan Siang

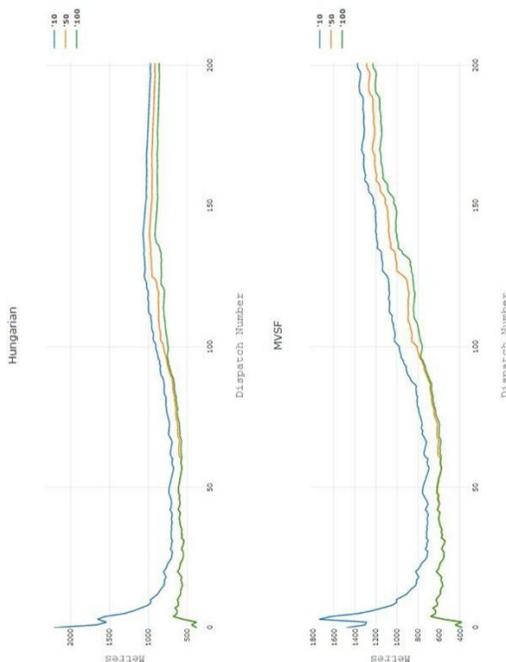


Gambar 12. Bar Chart Total Keberhasilan Malam

5.6 Uji Coba Tambahan

Hasil uji coba 1 hingga 9 saja masih belum dapat digunakan untuk menghasilkan hasil yang *valid* sebab jumlah *request* yang dikirimkan ke *google map service API* masih sama. Hal ini disebabkan karena *google map service API* hanya membatasi 10 *direction request* per detiknya untuk *user* dengan membership standar. Oleh karena itu uji coba ini digunakan untuk memeriksa apakah ada perbedaan hasil algoritma yang signifikan dalam perhitungan komputer dengan perbedaan durasi *batch* karena hanya menggunakan 1 komputer untuk keseluruhan uji coba dan hasilnya tidak memiliki pengaruh signifikan.

Untuk membuktikan perbedaan hasil dari durasi *batch* pada setiap algoritma dibuatkan program terpisah. Program ini bersifat *dummy* dan bertujuan untuk menggantikan batasan *request* dari *google map service API* yang hanya mengizinkan 10 *direction request* per detik. Faktor yang digunakan untuk program ini hanya jarak yang bersifat tegak lurus antara poin A dengan poin B. Untuk mendapatkan jarak tersebut digunakan persamaan *haversine*. Hasil uji coba tambahan dapat dilihat dari Gambar 13.



Gambar 13. Line Chart Perbandingan Perkembangan Jarak pada Uji Coba Tambahan

Pada uji coba tambahan digunakan durasi *batch* 10 detik (Biru), 50 detik (Kuning), dan 100 detik (Hijau). Untuk algoritma *Hungarian* dan *MVSF* dengan diperbesarnya durasi *batch* maka hasil rata-rata jarak yang diperoleh juga semakin rendah. Dari hasil uji coba sudah dapat dilihat efek dari perbedaan durasi *batch*.

6. KESIMPULAN DAN SARAN

Dari hasil perancangan dan pembuatan aplikasi, dapat diambil kesimpulan antara lain:

- Aplikasi dapat menjalankan simulasi statis dari skenario yang ada pada *file csv* dengan format yang telah ditentukan.
- Aplikasi dapat menjalankan skenario dinamis tanpa *file* skenario dengan *input* dari *user* untuk mengkonfigurasi pengaturan pada skenario dinamis.
- Aplikasi memberikan informasi total keberhasilan, total pembatalan pada sisi taksi, total pembatalan pada sisi penumpang, total taksi yang menghilang sebelum dipasangkan dengan penumpang, dan total penumpang yang menghilang sebelum dipasangkan dengan taksi dari hasil simulasi baik statis atau dinamis.
- Aplikasi memberikan informasi rata-rata perkembangan faktor pada setiap algoritma dari hasil proses simulasi.
- Pada bagian pembuktian uji coba tambahan, algoritma *Hungarian* dan *MVSF* yang digunakan memiliki hasil yang semakin baik bila data yang dikumpulkan semakin banyak hal ini dibuktikan dengan semakin besar durasi *batch* maka semakin rendah rata-rata jarak tempuhnya
- Algoritma *Hungarian* memiliki rata-rata jarak berkisar antara 500 meter – 1100 meter, sedangkan *MVSF* berkisar antara 500 meter – 1400 meter, pada awal simulasi algoritma *MVSF* dapat menyaingi algoritma *hungarian* dikarenakan jumlah taksi dan penumpang yang masih sedikit. Seiring dengan berjalannya simulasi peningkatan rata-rata jarak dengan algoritma

Saran yang dapat diberikan untuk mengembangkan dan menyempurnakan aplikasi antara lain:

- Penambahan fitur yang mampu untuk menambahkan faktor lain diluar durasi, jarak, dan total penumpang yang telah dilayani oleh setiap taksi.
- Penambahan fitur *scheduler* untuk dapat menjalankan aplikasi secara otomatis tanpa perintah *user*.
- Penambahan fitur yang untuk menganalisa hasil simulasi dan memberikan rekomendasi pengaturan simulasi.
- Penambahan fitur *cleaning* untuk membersihkan data yang kacau agar tidak merusak jalannya proses simulasi.
- Penambahan fitur untuk menampilkan statistik penggunaan kuota *directions API* dan batasan penggunaan agar tidak berlebihan.

7. DAFTAR PUSTAKA

- [1] Ahmed, B. 2015. Test case minimization approach using fault detection and combinatorial optimization techniques for configuration-aware structural testing. *Engineering Science and Technology, an International Journal*. Manno, Switzerland : Elsevier B.V., pp. 737-753.
- [2] Google LLC. 2018. GoogleMaps Platform Documentation. Retrieved May 8, 2018 from <https://developers.google.com/maps/documentation/>
- [3] Grau, J., Estrada, M., Tzenos, P., & Aifandopoulou, P. 2018. Agent-based Simulation Framework For The Taxi Sector

Modeling. *9th International Conference on Ambient Systems, Networks and Technologies, ANT-2018 and the 8th International Conference on Sustainable Energy Information Technology*. Porto, Portugal : Elsevier B.V., pp. 294-301.

- [4] Ingole P., Nichat M. 2013. Landmark Based Shortest Path Detection by Using Dijkstra Algorithm and Haversine Formula. *International Journal of Engineering Research and Applications (IJERA)*. Amravati, India : Rasoni College of Engineering and Management, pp. 162-165.
- [5] Pi-xu, Z., Hang-yu, W., Zhang-song, Z., Wen-kai, X., and Zhong-hong, W. 2011. Plot Association for Multi-sensor Based on Hungary Algorithm. *2011 International Conference on Advances in Engineering*. Wu Han,China : Elsevier Ltd., pp. 17-21.
- [6] Prokofyeva, N., & Boltunova, V. 2016. Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. Latvia : Elsevier B.V., pp. 52.
- [7] The PHP Development Team. (2018). mt_rand Documentation. Retrieved May 10, 2018 from <http://php.net/manual/en/function.mt-rand.php>
- [8] Xoxa, N., Zotaj, M., Tafa, I., Fejazj, J. 2014. Simulation of First Come First Served (FCFS) and Shortest Job First (SJF) Algorithms. Tirana, Albania : IJCSN - International Journal of Computer Science and Network, Volume 3, Issue 6, December 2014, pp. 444-449.
- [9] Zend Technologies. 2018. Zend Engine Explained. Retrieved May 5, 2018 from http://www.zend.com/products/zend_engine/in_depth