

**PENGEMBANGAN APLIKASI DRIVER ONLINE TRANSPORTATION  
TERJADWAL UNTUK OPERATOR PLATFORM**

**NASKAH PUBLIKASI**

Untuk Memenuhi Sebagian Persyaratan  
Mencapai Derajat Sarjana-1



Diajukan Oleh  
Deantari Dwikan Putri Aji  
20140140102

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH YOGYAKARTA**

**2019**

# **PENGEMBANGAN APLIKASI DRIVER ONLINE TRANSPORTATION TERJADWAL UNTUK OPERATOR PLATFORM**

Oleh: Deantari Aji<sup>1</sup>, Dwijoko Purbohadi<sup>2</sup>, Aprilia Kurnianti<sup>3</sup>

<sup>1</sup>Mahasiswa Program Studi Teknik Informatika, Fakultas Teknik, Universitas  
Muhammadiyah Yogyakarta

<sup>2</sup>Dosen Program Studi Teknik Informatika, Fakultas Teknik, Universitas  
Muhammadiyah Yogyakarta

<sup>3</sup>Dosen Program Studi Teknik Informatika, Fakultas Teknik, Universitas  
Muhammadiyah Yogyakarta

## **INTISARI**

Mobilitas merupakan kebutuhan sehari-hari yang dilakukan manusia untuk memenuhi kebutuhan. Karena aktivitas tersebut dilakukan setiap hari muncul jasa penyedia transportasi, salah satunya dengan metode *book-first* secara *online*. Penyedia jasa transportasi *online* bertipe *book first* perlu melakukan proses *rostering*. Proses ini yakni mencocokkan kriteria *order user* terhadap *driver* yang sesuai. Rostering masih dilakukan oleh manusia, yakni *operator*. Dalam proses bisnisnya, aplikasi transportasi *online* terus berjalan sehingga membutuhkan *operator* untuk memahami kapan *driver* memiliki kecocokan dan jadwal luang untuk mengerjakan *order* baru. Demi membantu *operator* mengetahui kapan *driver* memiliki jadwal serta waktu luang, dibuat sebuah aplikasi penjadwalan *driver*. Aplikasi penjadwalan ini tidak boleh terlalu panjang demi menghindari *over-information*. Karena alasan tersebut, dibuat aplikasi penjadwalan *driver* berdasarkan kurun waktu mingguan. Aplikasi penjadwalan yang dibuat menggunakan bahasa pemrograman turunan *javascript* yakni *Node.js* dan framework *Express.js*. Database yang digunakan yakni *MongoDB* serta hasil akhirnya berupa *JSON*.

Keyword: transportasi, javascript, Node.js, Express.js, MongoDB, JSON

**PENGEMBANGAN APLIKASI DRIVER ONLINE TRANSPORTATION  
TERJADWAL UNTUK OPERATOR PLATFORM**

Oleh: Deantari Aji<sup>1</sup>, Dwijoko Purbohadi<sup>2</sup>, Aprilia Kurnianti<sup>3</sup>

<sup>1</sup>Mahasiswa Program Studi Teknik Informatika, Fakultas Teknik, Universitas  
Muhammadiyah Yogyakarta

<sup>2</sup>Dosen Program Studi Teknik Informatika, Fakultas Teknik, Universitas  
Muhammadiyah Yogyakarta

<sup>3</sup>Dosen Program Studi Teknik Informatika, Fakultas Teknik, Universitas  
Muhammadiyah Yogyakarta

***ABSTRACT***

*One thing mankind do to fulfill their necessity is using mobility. This activity is a daily-use and a must therefore an online transportation services exist. One of them is online transportation with book-first method. The client of this service need to book their order within a period of time before they actually using this transportation. In regards of that matter, the operator which is the person responsible to manage the trips of the application needs a scheduler application based by trip. This application used javascript-base language which is Node.js with it is framework Express.js. The database used in this application is MongoDB and the final form is a JSON.*

*Keyword: transportation, javascript, Node.js, Express.js, MongoDB, JSON*

## A. Pendahuluan

Mobilitas merupakan aktivitas penting yang dilakukan manusia untuk memenuhi suatu tujuan atau kebutuhan. Demi memenuhi aktivitas ini, ada banyak cara transportasi yang telah ditempuh. Dimulai dari era ketika manusia bergerak menggunakan kaki hingga ke revolusi industri kendaraan bermesin uap pada abad ke 18 sampai kendaraan terbang yang diperkenalkan oleh Wright bersaudara, aktivitas ini terus dilakukan secara berulang sehingga menimbulkan kebutuhan jasa.

Di Indonesia sendiri terdapat jasa transportasi yang dijadikan budaya, seperti delman, becak, dan ojek. Hal ini membuktikan bahwa jasa transportasi sebenarnya telah diperkenalkan sejak lama, akan tetapi terus berubah sesuai dengan perkembangan zaman yang ada. Mulai dari waktu dan tempat, siapa dan seperti apa penyedia jasa, serta jenis kendaraan yang disediakan harus sesuai dengan keinginan *client*. Untuk memberikan pilihan ini, dibuatlah *online transportation*.

*Online transportation* adalah sebuah jasa transportasi berbasis internet dimana *client* dapat membuat order jasa transport secara virtual. Order yang ditangani oleh *online transportation* ada 2 jenis, yakni yang langsung dikerjakan setelah *client* membuat order ( *now* ) dan yang dikerjakan di waktu yang akan datang dengan metode memesan terlebih dulu ( *book first* ). Metode *book first* saat ini diterapkan pada *online transportation* bertipe sewa kendaraan ( *rent a car* ).

Pada proses bisnisnya, sebuah *online transportation* dengan metode sewa kendaraan harus terlebih dahulu mendapatkan order dari *client*. Order tersebut kemudian diproses oleh operator dengan cara mencocokkan kriteria order dengan *driver* yang sesuai, proses ini disebut *rostering*. Apabila *rostering* berhasil, maka order akan berubah status menjadi trip. Trip inilah data final yang digunakan operator untuk melihat jadwal *driver* dan kapan eksekusinya.

Ketika trip dilaksanakan, operator masih memiliki keharusan untuk menerima order baru serta melakukan *rostering* terhadapnya. Demi memastikan *driver* memiliki jadwal kosong yang dapat digunakan untuk menjalankan trip,

operator harus tahu persis kapan dan siapa *driver* yang memiliki jadwal trip beserta waktu tepatnya. Untuk membantu operator mengelola trip yang akan datang, dibutuhkan sebuah aplikasi penjadwalan trip.

## **B. Metode**

Pada penelitian ini digunakan SDLC (*System Development Life Cycle*) mengikuti aplikasi utama online transportation. Karena itu, cukup melakukan penulisan kebutuhan spesifik tentang aplikasi yang dikerjakan. Tahapan yang ditempuh yakni:

1. Analisa Kebutuhan

Pada tahapan ini dilakukan analisa untuk menemukan kebutuhan apa saja yang mampu mengatasi permasalahan serta deskriptif fungsi yang membantu jalannya aplikasi.

2. Kebutuhan Fungsional

Pada tahap ini dilakukan definisi fungsi serta tujuan fungsi tersebut dibuat. Fungsi-fungsi tersebut harus menunjang aplikasi ini mengatasi permasalahan.

3. Mendefinisikan Database yang Digunakan

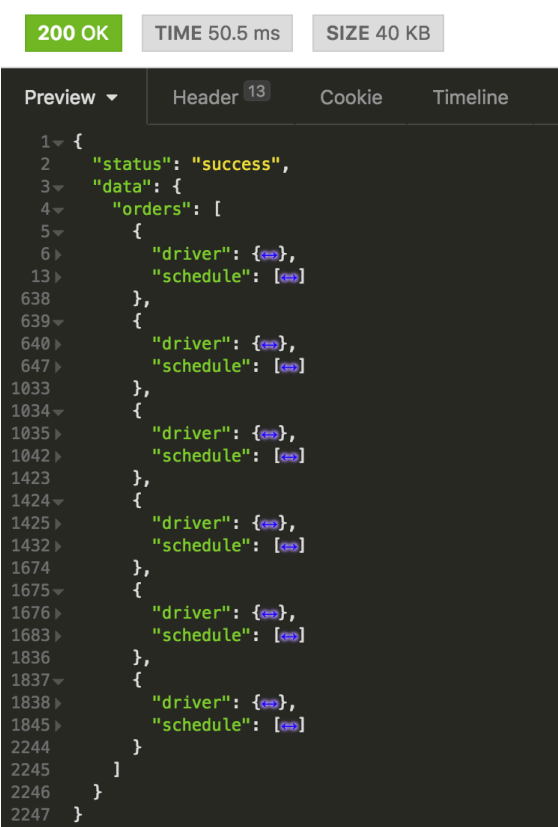
Pada tahap ini digunakan database yang sama dengan yang digunakan di aplikasi utama, yakni MongoDB. MongoDB memiliki atribut geospasial yang membuat data lokasi (*longitude, latitude*) dapat disimpan tanpa perlu dikonversi menjadi tipe data yang lain.

4. Mendefinisikan Implementasi dan Pengujian

Setelah aplikasi selesai dibuat, dilakukan pengujian menggunakan Insomnia sebagai *client REST-API* untuk mendapatkan data dari *Endpoint Service* berupa JSON terstruktur.

### C. Hasil dan Pembahasan

Pada aplikasi yang dibuat terdapat *service* yang berisi program guna memproses data *trip* dari database menjadi data *trip* dalam waktu satu minggu. Proses pemanggilan data dalam seminggu ini dibantu oleh Moment, sebuah package di dalam NPM ekosistem. Setelah data dalam seminggu didapat, dilakukan pemanggilan data semua *driver* yang ada. Hal ini dilakukan guna mendapatkan *list* *dirver* yang akan dideskripsikan jadwalnya. Selanjutnya dilakukan *coding* dengan tujuan menata data menjadi terstruktur sebelum dikonversi menjadi JSON dan dikirim ke *frontend*. Berikut gambar hasil yang didapat :



```
200 OK  TIME 50.5 ms  SIZE 40 KB
Preview  Header 13  Cookie  Timeline
1 {
2   "status": "success",
3   "data": {
4     "orders": [
5       {
6         "driver": {},
13        "schedule": []
638      },
639      {
640        "driver": {},
647        "schedule": []
1033      },
1034      {
1035        "driver": {},
1042        "schedule": []
1423      },
1424      {
1425        "driver": {},
1432        "schedule": []
1674      },
1675      {
1676        "driver": {},
1683        "schedule": []
1836      },
1837      {
1838        "driver": {},
1845        "schedule": []
2244      }
2245    ]
2246  }
2247 }
```

Gambar 4. 43. Bagian dari Hasil Data JSON yang Dikirim Service Sebagai Respon

Data JSON yang didapat dari request menuju endpoint API adalah data yang tersruktur, seperti yang ditampilkan di gambar 4.12. Di dalam data Driver dan Schedule terdapat rincian data *trip* yang ditampilkan di frontend. Keseluruhan hasil pengujian yang dilakukan terlampir.

#### **D. Kesimpulan**

Pengembangan Aplikasi Driver Online Transportation Terjadwal untuk Operator Platform berhasil dilakukan dan telah melalui pengujian pada sistemnya. Dari proses pembuatan aplikasi ini dapat ditarik kesimpulan :

1. Program dapat menampilkan jadwal trip yang akan dikerjakan.
2. Service memiliki data detail trip.
3. Service menyediakan trip berdasarkan *driver* yang mengerjakan trip.
4. Proses inisiasi service berjalan dengan baik sesuai pengujian yang dilakukan.

#### **E. Saran**

Saran penulis terhadap pengembangan aplikasi lain yang akan dilakukan setelah Pengembangan Aplikasi Driver Online Transportation Terjadwal untuk Operator Platform yaitu :

1. Sebelum bergabung dalam *development* aplikasi kolaboratif, sangat disarankan untuk memperdalam ilmu tentang bahasa pemrograman yang digunakan.
2. Dalam *development* aplikasi menggunakan *package*, pastikan untuk membaca dokumentasi resmi dari penerbit *package*.

## F. Daftar Pustaka

- Osetskyi, V. (2019). *Web Application Architecture*. Retrieved from Medium: <https://medium.com/existek/web-application-architecture-da77ea0cb520>
- Express.js. (2016). *Express.js*. Retrieved from Express: <https://expressjs.com/>
- Vivah, L. (2017). *THE BEGINNER'S GUIDE: Understanding Node.js & Express.js fundamentals*. Retrieved from Medium: <https://medium.com/@LindaVivah/the-beginners-guide-understanding-node-js-express-js-fundamentals-e15493462be1>
- Selvaganesh. (2018). *How Nodejs middleware Works?* Retrieved from Medium: <https://medium.com/@selvaganesh93/how-node-js-middleware-works-d8e02a936113>
- Patel, P. (2018). *What Exactly is Node.js*. Retrieved from Medium: <https://medium.freecodecamp.org/what-exactly-is-node-js-ae36e97449f5>
- B, S. V. (2018). *Introduction to MongoDB*. Retrieved from Medium: <https://medium.com/@saivittalb/introduction-to-mongodb-859ed4426994>
- Karnik, N. (2018). *Introduction to Mongoose for MongoDB*. Retrieved from FreeCodeCamp: <https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/>
- JSON. (n.d.). *JSON*. Retrieved from <https://json.org/>
- Wikidata. (2017). *Mercurial*. Retrieved from Wikidata: <https://www.wikidata.org/wiki/Q476543>
- Domrongchai, A. (2017). *Test you API's with Insomnia REST Client*. Retrieved from Medium: <https://medium.com/@artiwarahdamrongchai/test-your-apis-with-insomnia-rest-client-355093f32755>
- Setiawan Honggowibowo, T. S. (18 Juni 2005). *Sistem Reservasi Pesawat Terbang Berbasis Web*. (pp. E-62). Yogyakarta: Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005).
- Bitbucket. (2019). *Bitbucket Documentation*. Retrieved from Bitbucket: <https://confluence.atlassian.com/bitbucket>



Git. (2017). *Git About*. Retrieved from Git: <https://git-scm.com/docs/giteveryday>

NPM. (2018). *NPM About*. Retrieved from NPM: <https://docs.npmjs.com/about-npm/>

Org, N. (2017). *Node.js*. Retrieved from nodejs: <https://nodejs.org/en/>

Medium. (2017). *Front-End, Back-End, Full-Stack*. Retrieved from Medium: <https://medium.com/@makersinstitute/front-end-back-end-full-stack-apa-artinya-36e0f25e8142>

MIT. (2017). *MIT*. Retrieved from MIT Open Source Initiative:

<https://opensource.org/licenses/MIT>

Studio, V. (2016). *Visual Studio Code*. Retrieved from Visual Studio:

<https://code.visualstudio.com/docs>