

SISTEM REAL-TIME UNTUK MANAJEMEN MOBIL ANTARKOTA MENGGUNAKAN NODE JS BERBASIS TCP/IP

^[1]Muhammad Rizki Samsul Ariefin, ^[2]Cucu Suhery, ^[3]Yulrio Brianorman

^{[1] [2] [3]}Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura

Jl. Ahmad Yani, Pontianak

Telp./Fax.: (0561) 577963

e-mail:

^[1]rizkisamsul@gmail.com, ^[2]csuhery@gmail.com, ^[3]yulrio.brianorman@siskom.untan.ac.id

ABSTRAK

Proses penjemputan penumpang mobil antarkota berdasarkan data antrian yang diberikan pengelola (operator) tidak efisien karena supir mobil antarkota dapat mengulangi rute penjemputan sehingga penggunaan bahan bakar minyak (BBM) pada mobil menjadi lebih banyak dan proses penjemputan penumpang menjadi lebih lama. Permasalahan lain yang akan terjadi adalah jika penumpang memberikan alamat yang tidak lengkap atau kurang jelas sehingga proses penjemputan menjadi lama. Untuk menjawab permasalahan tersebut, dalam penelitian ini dibuat sistem yang dapat mengoptimalkan penjemputan penumpang jasa transportasi mobil antarkota pada Kota Pontianak, yaitu dengan menggunakan perangkat GPS. Sistem real-time dapat diterapkan pada TCP/IP sehingga memberikan keluaran dengan waktu tunda yang singkat. Perangkat keras utama yang digunakan adalah Mikrokontroler Arduino beserta GPS Shield dan Mini Komputer Raspberry Pi. Perangkat lunak utama yang digunakan adalah Node Js dan Google Maps Versi ke-3 sebagai penyedia peta elektronik Kota Pontianak. Setelah implementasi dan pengujian maka sistem real-time untuk manajemen penjemputan penumpang mobil antarkota menggunakan Google Maps ini dapat memberikan solusi berupa akurasi koordinat posisi, efisiensi respon memuat halaman website dan efisiensi harga operasional penggunaan data pada manajemen penjemputan jasa transportasi mobil antarkota melalui website sistem informasi, yang dapat diakses melalui TCP/IP.

Kata kunci- internet protocol, mobil antarkota, Node Js, Raspberry Pi, real-time.

1. PENDAHULUAN

Pada Kota Pontianak terdapat masyarakat yang berpergian menggunakan jasa transportasi ke daerah lainnya di Kalimantan Barat. Jasa transportasi antarkota yang umum digunakan oleh masyarakat Kota Pontianak melalui jalur darat adalah mobil antarkota [1]. Terdapat masalah jika supir mobil antarkota mengulangi rute penjemputan karena proses penjemputan berdasarkan data antrian yang diberikan pengelola (operator). Masalah tersebut menyebabkan Penggunaan bahan bakar minyak (BBM) pada mobil antarkota menjadi lebih banyak dan proses penjemputan penumpang menjadi lebih lama. Permasalahan lain yang akan terjadi adalah jika penumpang memberikan alamat yang tidak lengkap atau kurang jelas sehingga proses penjemputan menjadi lama.

Untuk menjawab permasalahan tersebut, dibutuhkan sebuah sistem yang dapat meng-

optimalkan jasa transportasi mobil antarkota pada Kota Pontianak. Salah satu solusi yang ditawarkan terdapat dalam cabang ilmu komputer yaitu Sistem Waktu Nyata (*Real-time System*) sehingga dapat memberikan data koordinat posisi pada waktu sebenarnya tanpa waktu tunda (*delay*) yang panjang.

Saat ini telah digunakan model konvensional dalam sistem manajemen pada jasa penjemputan mobil antarkota yaitu penumpang memesan melalui telepon atau *Short Message Service (SMS)* kepada operator (pengelola), kemudian operator mengatur jadwal penjemputan dan memberikan pesan kepada supir mobil antarkota berupa informasi alamat jelas penumpang. Sistem konvensional tersebut tidak berkerja optimal karena supir dapat tersesat dalam melakukan penjemputan dan supir dapat melakukan kecurangan dalam melaporkan jumlah penumpang yang di bawa kepada pengelola. Teknologi *Global*

Positioning System (GPS) untuk mendeteksi telah digunakan, tetapi hanya melalui *SMS* [2]. Pada alat tersebut memiliki kelemahan yaitu biaya operasional *SMS* yang mahal dan terdapat waktu tunda dalam penerimaan perintah hingga pengiriman data koordinat posisi menjadi tidak akurat.

Bagi pengelola jasa transportasi mobil antarkota akan sangat terbantu ketika dibuat suatu sistem manajemen *real-time* pada jasa tersebut sehingga mampu memberikan solusi dalam mengatur dan mengelola jasa tersebut secara efektif dan efisien. Supir dan penumpang akan dapat saling mengetahui koordinat posisi. Hal ini berbeda dengan sistem konvensional yang hanya dapat digunakan oleh pengelola saja untuk mengetahui koordinat posisi mobil.

Transmission Control Protocol of Internet Protocol (TCP/IP) adalah standar komunikasi data yang digunakan dalam jaringan *internet* dalam proses penyebaran data. Melalui *TCP/IP* maka perangkat elektronik yang berbeda dapat terhubung dengan luas, mudah dan cepat. Sistem *realtime* dapat diterapkan pada *TCP/IP* sehingga memberikan keluaran dengan waktu tunda yang singkat. Biaya pengoperasian dari sistem menggunakan *TCP/IP* lebih murah, karena proses pada sistem terus terjadi pengiriman (*transmit*) dan penerimaan (*receive*) setiap detik hanya menghabiskan Kilo Byte data.

Penulis berharap pada penelitian sistem sistem *real-time* untuk manajemen jasa transportasi mobil antarkota berbasis *TCP/IP* dapat diterapkan pada Kota Pontianak untuk mengoptimalkan pengelolaan jasa tersebut.

2. TINJAUAN PUSTAKA

2.1 Real-time System

Real-time system (sistem waktu nyata) harus menghasilkan respon yang tepat dalam batas waktu yang telah ditentukan. Jika respon melewati batas waktu tersebut, maka terjadi degradasi performansi atau kegagalan sistem. Sebuah Sistem *real-time* adalah sistem yang kebenarannya secara logis didasarkan pada kebenaran hasil-hasil keluaran sistem dan ketepatan waktu hasil-hasil tersebut saat dikeluarkan. *Real-time* adalah sistem yang dalam menghasilkan keluaran (*output*) atau respon tidak melebihi batas waktu (*response-time constraint*) yang telah ditetapkan [3]. Menurut fungsinya sistem ini terbagi menjadi

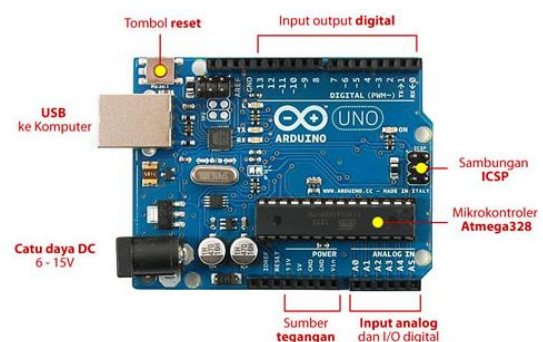
Hard Real-time System dan *Soft Real-time System*.

Hard Real-time System dibutuhkan untuk menyelesaikan *critical task* dengan jaminan waktu tertentu. Jika kebutuhan waktu tidak terpenuhi, maka aplikasi akan gagal. Secara umum, sebuah proses di kirim dengan sebuah pernyataan jumlah waktu dimana dibutuhkan untuk menyelesaikan atau menjalankan proses *input-output (I/O)*.

Soft Real-time System memiliki sedikit kelonggaran waktu dibandingkan dengan sistem *hard real-time* karena perangkat lunak hanya mendapatkan prioritas dalam menjalankan pemrosesan sehingga resiko kegagalan tidak berakibat fatal. Adanya keterlambatan waktu respon akan mengakibatkan penurunan kinerja dan mengakibatkan waktu tunda yang lebih lama. Pada penelitian ini menggunakan sistem *soft real-time (SRTS)* karena penelitian dilakukan terhadap perangkat lunak untuk mendapatkan waktu tunda yang singkat dalam pengiriman dan penerimaan data berbasis *TCP/IP* sehingga tidak berakibat fatal pada perangkat keras atau sistem kendali mobil antarkota.

2.2 Arduino

Arduino adalah komponen elektronik atau papan rangkaian elektronik yang bersifat *open-source* (bebas) dan di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan arsitektur AVR dari perusahaan Atmel. Bagian dan komponen dari Mikrokontroler Arduino UNO dapat dilihat pada Gambar 1.

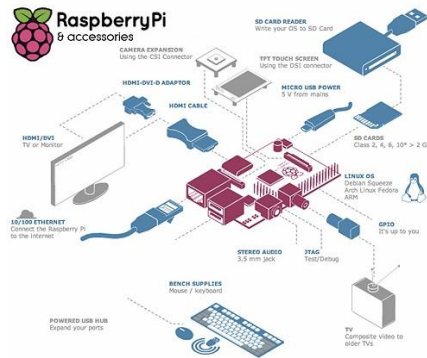


Gambar 1. Model board Arduino Uno

2.3 Raspberry Pi

Raspberry Pi adalah sebuah mini komputer papan tunggal (*single-board mini-computer*) berukuran dan menggunakan daya

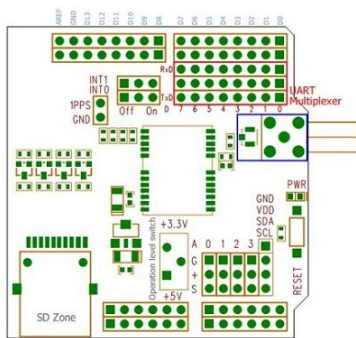
lebih kecil (5 watt) dari *Mainboard Personal Computer* (>200watt). Raspberry Pi telah dilengkapi dengan semua fungsi seperti sebuah komputer lengkap yang memiliki sistem operasi linux yang bebas (*open-source*), menggunakan *SoC (System-on-a-chip)* dengan arsitektur ARM. Raspberry Pi mendukung *Port I/O* yaitu *HDMI*, *RJ45 (TCP/IP)*, *SD Card* dan *USB 2.0*. Raspberry Pi dapat dilihat pada Gambar 2.



Gambar 2. Komponen I/O pada Raspberry Pi

2.4 Global Positioning System (GPS)

Global Positioning System (GPS) adalah perangkat elektronik yang memiliki *Integrated Circuit (IC)* dan berfungsi sebagai penangkap (*receiver*) sinyal radio yang dipancarkan oleh satelit dalam orbitnya mengelilingi bumi. *GPS* mengambil informasi dari setiap satelit dan dengan menggunakan metode Triangulasi pada *IC* untuk menghitung lokasi Pengguna dengan tepat. Perangkat *GPS* dapat dilihat pada Gambar 3.



Gambar 3. Arduino GPS Shield

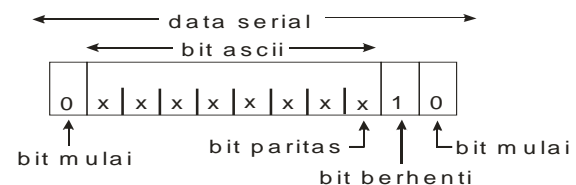
2.5 Node Js

Node Js adalah sebuah *platform* untuk membangun aplikasi *real-time*. Node Js dapat menangani aksi *event (production, detection,*

consumption of, dan reaction) pada server secara *non-blocking* yang berarti data yang sampai di server secara langsung dapat dimanipulasi atau terdistribusi ke *client* tanpa disimpan terlebih dahulu. Node Js berupa *framework* yang dapat digunakan untuk membuat *scalable network applications* dengan menggunakan model *event-driven*.

2.6 Komunikasi Serial

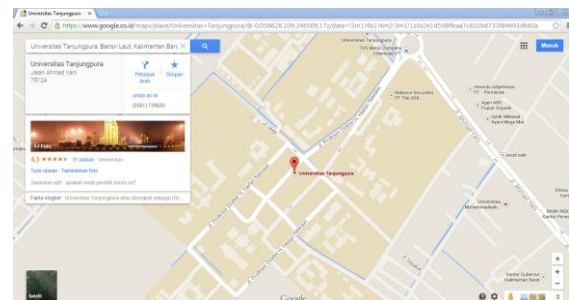
Komunikasi serial adalah komunikasi yang pengiriman datanya setiap bit secara berurutan dan bergantian. Komunikasi pengiriman data dikirim setiap bit sehingga lebih lambat dibandingkan komunikasi paralel, sehingga komunikasi serial merupakan salah satu metode komunikasi data di mana hanya setiap bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Pengiriman data dengan komunikasi serial dapat dilihat pada Gambar 4.



Gambar 4. Pengiriman karakter komunikasi serial

2.7 Google Maps

Google Maps adalah layanan *open-source* dari perusahaan Google yang cukup populer dalam *internet*. Google Maps dapat digunakan oleh setiap Pengguna dalam *website* pribadi melalui Google Maps *Application Programming Interface (API)*. Tampilan dan halaman peta dengan menggunakan Google Maps dapat dilihat pada Gambar 5.



Gambar 5. Halaman Google Maps dengan menggunakan Google Chrome

3. METODOLOGI PENELITIAN

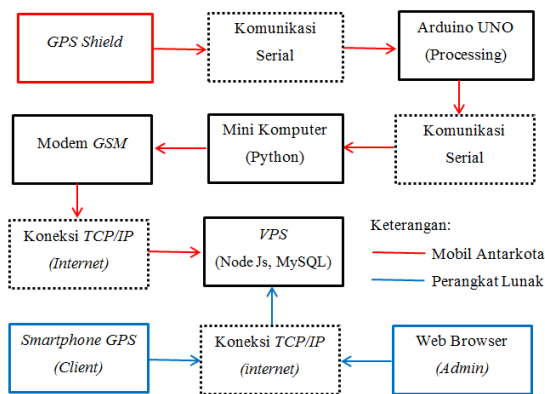
Penelitian ini menggunakan metodologi penelitian yang dimulai dari studi literatur tentang sistem *real-time* dan komponen-komponen yang dibutuhkan dalam pembuatan sistem informasi berbasis *TCP/IP* menggunakan Node Js. Kemudian dilakukan wawancara kepada pengelola dan supir jasa transportasi mobil antarkota.

Tahap selanjutnya adalah perancangan yang diawali dengan analisis kebutuhan dari perangkat lunak maupun perangkat keras. Tahap yang terakhir adalah pengujian. Pengujian dilakukan pada setiap sub sistem.

4. PERANCANGAN SISTEM

4.1. Diagram Blok Perancangan Sistem

Diagram blok perancangan sistem ditunjukkan pada Gambar 6.



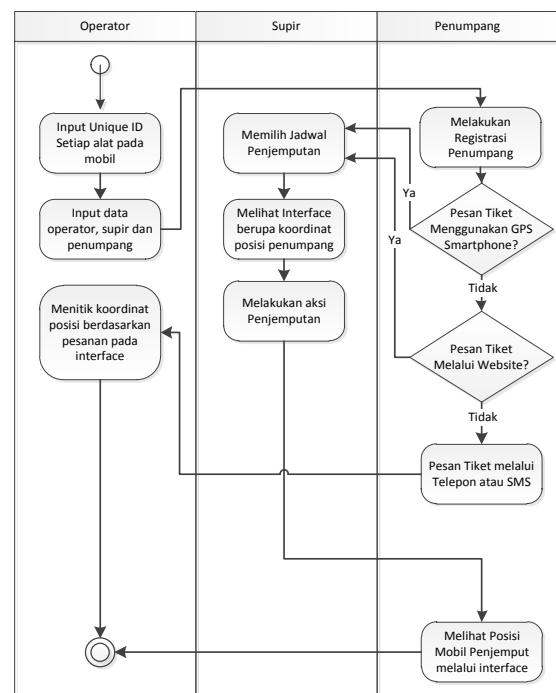
Gambar 6. Diagram Blok Perancangan Sistem

Penjelasan dari diagram blok rancangan sistem pada Gambar 6 adalah sebagai berikut:

1. *GPS Shield* menghasilkan data yang berasal dari satelit.
2. Data *GPS* berupa koordinat Latitude dan Longitude dikirimkan ke Arduino dengan metode komunikasi serial [4].
3. Arduino diprogram untuk terintegrasi dengan *GPS Shield* sehingga data siap untuk dikirim melalui *Port USB*. Data yang dikirim oleh Arduino ke mini komputer dengan metode komunikasi serial berupa koordinat Latitude dan Longitude. Mini Komputer menangkap data yang dikirimkan oleh Arduino dan diteruskan melalui protokol Node Js menggunakan bahasa Python [5].

4. Modem menghubungkan koneksi *TCP/IP* sehingga Mini Komputer terhubung pada *internet*.
5. *Virtual Private Server (VPS)* menangkap data yang dikirim oleh Node Js dan mengolah data sesuai dengan permintaan Pengguna atau kebutuhan sistem [6].
6. *Smartphone* dengan fitur *GPS* atau *Web Browser* terhubung dengan *server* melalui *internet* dan mengirimkan data berupa koordinat posisi pada penumpang atau operator.

4.2. Activity Diagram Operator, Supir dan Penumpang



Gambar 7. Activity Diagram pada Sistem Informasi

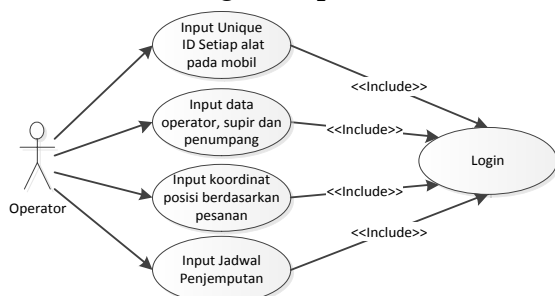
Rancangan *Activity Diagram* secara umum pada Gambar 7 dibedakan menjadi 3 blok yang saling berhubungan yaitu operator, supir dan penumpang. Alir dari aktifitas yang dilakukan pada *Activity Diagram* adalah:

1. Operator memasukkan Identitas unik pada alat *GPS* sehingga setiap alat memiliki identitas yang berbeda dan dimanfaatkan untuk mengatur setiap mobil.
2. Operator memasukkan data operator, supir dan penumpang kedalam *database* berdasarkan identitas.
3. Setelah memiliki akun yang terdaftar pada sistem kemudian penumpang melakukan pemesanan tiket menggunakan *Smart-*

phone yang memiliki Fitur *GPS* sehingga supir melakukan penjemputan dan dapat melihat posisi pemesan tiket (penumpang).

4. Jika penumpang tidak memiliki *Smartphone* yang memiliki fitur *GPS* maka dapat memesan melalui *website* dengan menentukan titik melalui Google Maps sehingga supir melakukan penjemputan dan dapat melihat posisi pemesan tiket (penumpang).
5. Jika penumpang tidak menggunakan akses *internet* untuk terhubung dengan aplikasi, maka penumpang dapat memesan tiket melalui telepon atau *SMS* kepada operator.
6. Jika penumpang sudah melakukan pemesanan tiket maka penumpang dapat melihat tiket yang telah dipesan dan penumpang diperbolehkan memesan tiket untuk hari-hari selanjutnya.

4.3. Use-case Diagram Operator



Gambar 8. Use-case Diagram pada operator

Use-case Diagram yang menggambarkan fungsionalitas sistem yang dilakukan oleh operator ditunjukkan pada Gambar 8. Operator memiliki beberapa proses yang harus melakukan proses *login* terlebih dahulu. Proses yang dilakukan operator dalam melakukan manajemen pada sistem adalah:

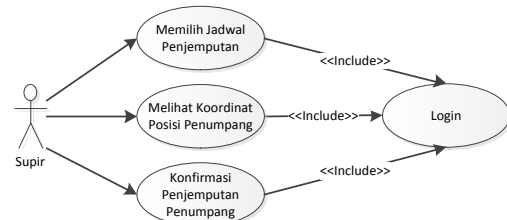
1. Memasukan identitas unik pada setiap mobil sehingga menghasilkan identitas mobil yang berbeda. Data unik tersebut berfungsi sebagai penentu data yang akan dikirim oleh alat pada sistem.
2. Melakukan manajemen data akun pengguna pada level operator, supir dan penumpang. Dalam manajemen ini dilakukan proses *input-output (I/O)* pada *database* dengan menggunakan *Query Select, Insert, Delete* dan *Update*.

3. Memasukan data koordinat posisi pada pesanan yang melalui metode konvensional yaitu telepon atau *SMS*.
4. Melakukan manajemen pada jadwal penjemputan sesuai dengan rute yang di lalui pada setiap mobil.

4.4. Use-case Diagram Supir

Use-case Diagram yang menggambarkan fungsionalitas sistem yang dilakukan oleh supir di-tunjukkan pada Gambar 9. Supir memiliki beberapa tugas yang harus melakukan proses *login* terlebih dahulu. Proses yang dilakukan supir adalah:

1. Melakukan pemilihan jadwal untuk melihat data penumpang yang memesan tiket pada mobil. Setelah melakukan pemilihan jadwal, supir dapat melihat koordinat posisi, nomor telepon, alamat dan nomor kursi dari setiap penumpang yang memesan tiket.
2. Setelah melihat koordinat posisi dan data alamat, maka supir memilih penumpang untuk dijemput.
3. Setelah sampai pada tempat penjemputan, maka supir melakukan konfirmasi penjemputan pada sistem, sehingga sistem dapat membedakan penumpang yang masih memiliki tiket aktif dan penumpang yang telah berada di mobil (tiket tidak aktif).



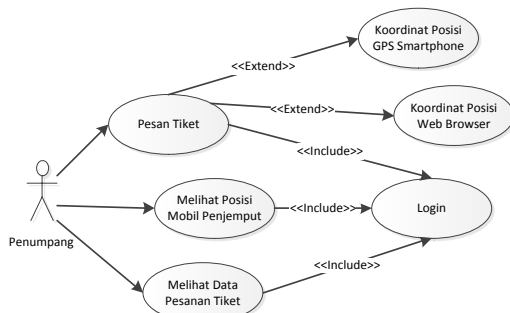
Gambar 9. Use-case Diagram pada supir

4.5. Use-case Diagram Penumpang

Use-case Diagram yang menggambarkan fungsionalitas sistem yang dilakukan oleh penumpang ditunjukkan pada Gambar 10. Penumpang memiliki beberapa tugas yang harus melakukan proses *login* terlebih dahulu. Proses yang dilakukan penumpang adalah:

1. Melakukan pemesanan tiket melalui *Smartphone* yang memiliki fitur *GPS*. Kemudian data koordinat posisi disimpan melalui protokol *TCP/IP* menggunakan *Node Js*. Selanjutnya penumpang dapat melihat posisi mobil yang dipesan melalui antarmuka (*interface*) pada *Smartphone*.

2. Jika penumpang tidak memiliki *Smart-phone* dengan fitur *GPS* atau penumpang ingin memesan tiket untuk tempat lain di tempatnya berada, maka penumpang dapat menggunakan fitur dengan memanfaatkan Google Maps pada *website*. Setelah melakukan pemesanan menggunakan peta elektronik yang disediakan oleh Google Maps maka penumpang dapat melihat posisi mobil yang dipesan melalui antarmuka (*interface*) pada *website*.
3. Penumpang dapat memesan tiket untuk waktu hari ini atau sesuai tanggal yang diinginkan. Seluruh data tiket yang telah dipesan dapat dilihat oleh penumpang dan data yang sudah tidak aktif, maka tidak ditampilkan pada antarmuka.

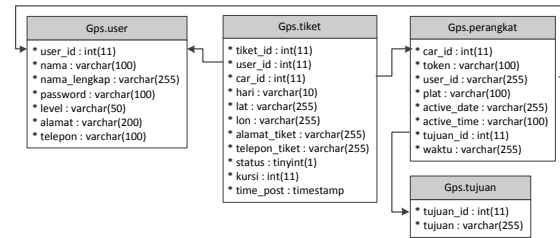


Gambar 10. Use-case Diagram pada penumpang

4.6. Entity Relationship Diagram (ERD) pada Database

Entity-Relationship Diagram yang menggambarkan hubungan antar tabel pada *data-base* dapat dilihat pada Gambar 11. Tabel yang digunakan adalah sebanyak 4 tabel dengan *primary key* pengguna_id pada tabel pengguna, tiket_id pada tabel tiket, car_id pada tabel perangkat, dan tujuan_id pada tabel tujuan.

Tabel pengguna digunakan sebagai tempat penyimpanan data pengguna tabel tujuan berfungsi sebagai tempat penyimpanan data kota tujuan sehingga data dapat berubah sesuai dengan rute perusahaan mobil antarkota. Tabel tiket digunakan sebagai tempat penyimpanan data pada saat terjadi aksi pemesanan tiket dan memiliki relasi ke seluruh tabel. Tabel perangkat digunakan sebagai tempat penyimpanan data yang berisi informasi pada alat yang diletakkan pada mobil antarkota.



Gambar 11. ERD pada Database

4.7. Perancangan Antarmuka (Interface)

Antarmuka berfungsi sebagai penghubung antara sistem dan Pengguna, oleh karena itu desain adalah hal yang sangat penting untuk mempermudah Pengguna menggunakan sistem. Desain pada aplikasi berbasis *website* yang dibuat menggunakan model *responsive design*, sehingga satu halaman ukuran besar yang umumnya dibuka melalui *Personal Computer* dapat menyesuaikan layar perangkat *Smartphone*.

5. IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1. Implementasi Perancangan Perangkat Keras

Pada Gambar 12 dapat dilihat lampu indikator Modem *GSM* menyala bewarna biru, hal ini menandakan bahwa perangkat terhubung pada jaringan *internet*. Antena *GPS* memiliki magnet dan panjang 3 meter akan diletakkan pada atap mobil untuk mendapatkan sinyal *GPS*.



Gambar 12. Perangkat Keras secara keseluruhan

5.2. Implementasi Perancangan Sistem Informasi untuk manajemen Mobil Antarkota menggunakan Node Js Berbasis TCP/IP

Aplikasi Node Js berjalan dengan *console terminal* sistem operasi linux, menggunakan *Port 8888* dan *Port 2121* yang menghubungkan perangkat lunak dan perangkat keras ditunjukkan pada Gambar 13.

```
Last login: Tue Aug 26 20:15:52 2014 from 36.83.7.144
root@vps:~# dir
app
root@vps:~# cd app
root@vps:~/app# nodejs server.js
Service GPS Tracking Berjalan pada PORT 8888
Service Database Socket.io Berjalan pada PORT 2121
```

Gambar 13. Aplikasi dengan menggunakan Node Js pada sisi *server*

5.3. Tampilan antarmuka untuk Web Browser Personal Computer

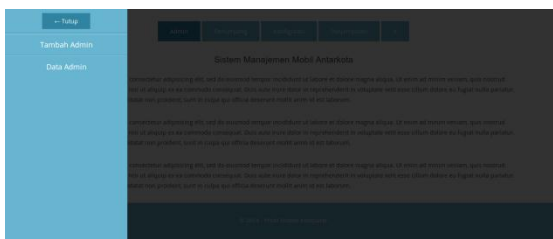
Desain tampilan antarmuka website dapat dilihat pada Gambar 14. Tampilan ini umumnya digunakan oleh operator.



Gambar 14. Tampilan pada Web Browser PC

5.4. Tampilan antarmuka dengan submenu untuk Web Browser Personal Computer

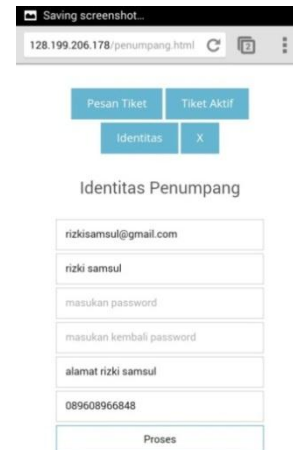
Desain tampilan antarmuka ketika salah satu menu pada *website* dipilih dapat dilihat pada Gambar 15. Tampilan ini umumnya digunakan oleh operator.



Gambar 15. Tampilan dengan sub menu pada Web Browser PC

5.5. Tampilan antarmuka untuk Web Browser Smartphone

Desain tampilan antarmuka *website* ketika menggunakan *Smartphone* dapat dilihat pada Gambar 16. Tampilan ini ukuran layar *Smartphone*. Tampilan ini umumnya digunakan oleh supir dan penumpang.



Gambar 16. Tampilan pada Web Browser Smartphone

5.6. Tampilan antarmuka dengan submenu untuk Web Browser Smartphone

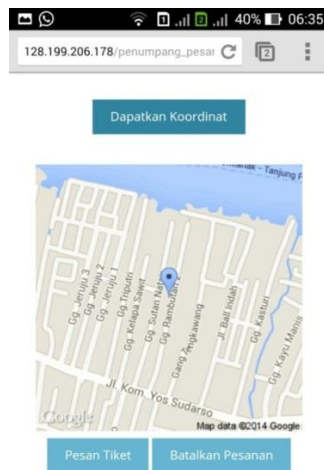
Desain tampilan antarmuka website ketika menggunakan *Smartphone* dan menu dipilih dapat dilihat pada Gambar 17. Tampilan ini umumnya digunakan oleh supir dan penumpang.



Gambar 17. Tampilan dengan submenu pada Web Browser Smartphone

5.7. Prosedur pemesanan tiket mobil antarkota melalui Web Browser Smartphone

Pengguna diharuskan mengaktifkan fitur *GPS* pada *Smartphone* dan akan menghasilkan titik koordinat posisi penjemputan penumpang, yang ditunjukkan pada Gambar 18.



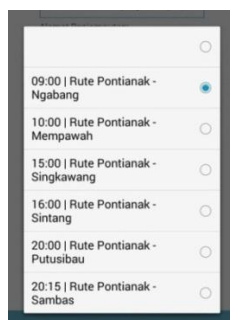
Gambar 18. Tampilan dari submenu pemesanan dengan *GPS Smartphone*

Setelah menentukan titik dan memilih untuk menyimpan tanda koordinat posisi pada Gambar 18 maka *form* penjemputan penumpang akan tampil dan ditunjukkan pada Gambar 19.

 A screenshot of a mobile application interface showing a form for entering passenger pickup details. The form has several fields: 'Alamat Penumpang:' with the value 'Alamat rizki samsul', 'Salin Alamat Penumpang Ke Penjemputan', 'Alamat Penjemputan:' with the value 'Gang Kasturi No.6, West Pontianak, Pontianak, West Kalimantan 78244... Indonesia', and 'Nomor Telepon:' with the value '089608966848'. There are also buttons for 'Pesan Tiket', 'Tiket Aktif', 'Identitas', and 'X'.

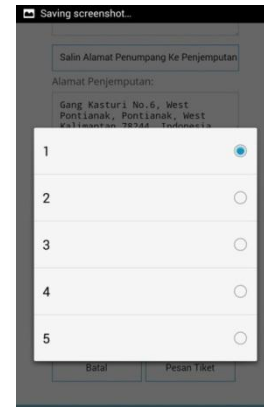
Gambar 19. Tampilan *form* isian pesan tiket

Penumpang mobil antarkota dapat memilih jadwal penjemputan pada sistem yang ditunjukkan pada Gambar 20.



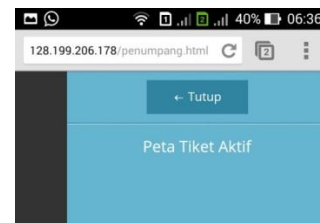
Gambar 20. Tampilan pilihan jadwal pada pesan tiket

Penumpang dapat memilih nomor kursi mobil pada sistem yang dapat ditunjukkan pada Gambar 21. Nomor kursi yang telah dipesan oleh penumpang pada setiap jadwal akan tidak tampil pada antarmuka.



Gambar 21. Tampilan pilihan kursi pada pesan tiket

Jika menu “Tiket Aktif” mendapat aksi klik, maka akan tampil submenu dengan animasi geser dari arah kiri halaman yang dapat ditunjukkan pada Gambar 22.



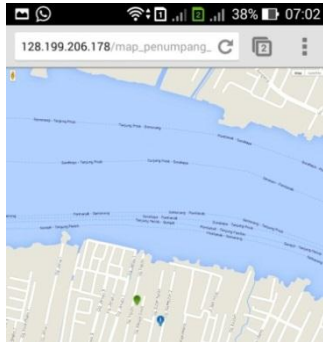
Gambar 22. Tampilan submenu dari menu tiket aktif

Jika submenu “Peta Tiket Aktif” mendapat aksi klik, maka halaman akan memuat data tiket aktif yang dapat melihat posisi mobil penjemputan, dapat dilihat pada Gambar 23.



Gambar 23. Tampilan data dari tiket aktif

Setelah melakukan aksi klik pada button “Lihat Posisi Mobil” pada Gambar 23, maka Pengguna akan mendapatkan tampilan berupa peta *real-time* yaitu Gambar 24 yang akan menunjukkan posisi mobil antarkota yang bergerak menuju posisi penumpang.



Gambar 24. Tampilan peta realtime posisi mobil

5.8. Pengujian Efisiensi Pada Pengelolaan Sistem Berbasis Node Js Secara Real-Time

Data yang masuk kedalam *server* dapat digunakan sebagai parameter pengukuran waktu untuk mendapatkan jarak tempuh yang terlewatkan pada saat terjadinya waktu tunda. Untuk mendapatkan jarak tempuh tersebut maka dapat dituliskan dalam persamaan:

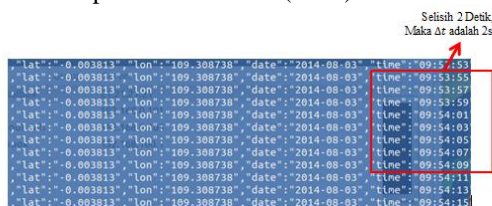
$$S_r = v \times \Delta t \dots\dots\dots (1)$$

Keterangan:

S_r adalah jarak tempuh selama waktu tunda (meter)

v adalah kecepatan (meter/detik)

Δt adalah perubahan waktu (detik)



Gambar 25. Data yang terbaca oleh Aplikasi Node Js pada sisi *Server*

Gambar 25 menunjukkan data yang masuk kedalam *server* secara *real-time*, yang dapat berisi informasi berupa id perangkat, token perangkat, koordinat latitude, koordinat longitude, tanggal dan waktu pengiriman data.

Sehingga terdapat waktu tunda selama 2 detik dalam sistem *real-time* pada mobil antarkota, maka persamaan 1 dapat digunakan

untuk mendapatkan jarak yang ditempuh selama waktu tunda ditunjukkan pada Tabel 1.

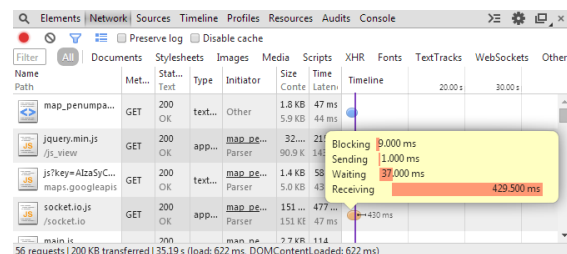
Tabel 1. Jarak yang ditempuh pada saat waktu tunda

Waktu Tunda	V (KM/H)	Jarak
2 detik	10	5,555555556 M
2 detik	20	11,11111111 M
2 detik	30	16,66666667 M
2 detik	40	22,22222222 M
2 detik	50	27,77777778 M
2 detik	60	33,33333333 M
2 detik	70	38,88888889 M
2 detik	80	44,44444444 M

Berdasarkan pengujian jarak tempuh saat waktu tunda yang ditunjukkan pada Gambar 25 dan Tabel 1, maka dapat disimpulkan bahwa sistem dapat bekerja secara efisien dalam akurasi koordinat posisi yaitu maksimal 44,44 meter setiap 2 detik waktu tunda pada saat kecepatan mencapai 80 KM/H.

5.9. Pengujian Waktu Respon pada Web Browser

Google Chrome adalah *Web Browser* yang dapat memberikan informasi berupa elemen struktur *website* dan status jaringan yang digunakan oleh suatu *website*. Gambar dapat dilihat tampilan dari *Google Chrome Network* yang memberikan informasi berupa waktu memuat protokol Node Js pada aplikasi berbasis *website*. Protokol *TCP/IP* Node Js yaitu *socket.io* menggunakan waktu tunggu selama 37 ms dan waktu eksekusi selama 429,500 ms. Sedangkan waktu untuk memuat seluruh data pada aplikasi berbasis *website* adalah 622 ms.



Gambar 26. Respon *Google Chrome* dalam memuat Protokol Node Js

Berdasarkan pengujian waktu respon yang ditunjukkan pada Gambar 26 maka dapat disimpulkan bahwa sistem dapat bekerja

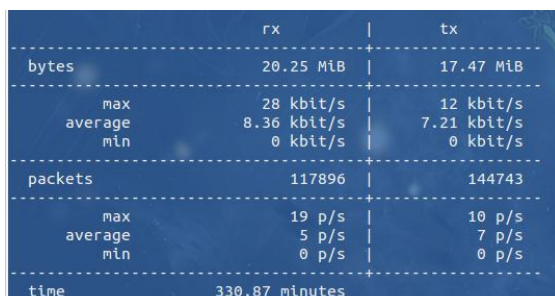
secara efisien dalam respon memuat halaman aplikasi yaitu kurang dari 1 detik.

5.10. Pengujian Perangkat Keras melalui TCP/IP

Sistem Operasi linux dapat digunakan sebagai penghitung ukuran data pada perangkat keras melalui *TCP/IP*. Perintah yang digunakan adalah `vncstat -i wlan0`. Data perhitungan ukuran data yang digunakan dapat ditunjukkan pada Gambar 27. Berdasarkan data tersebut dapat diketahui ukuran penerimaan data dalam 330.87 menit adalah sebesar 20.25 MB dan ukuran pengiriman data dalam 330.87 menit adalah sebesar 17.47 MB.

Tabel 2. Penggunaan Data data Perangkat Keras

Aliran Data	Waktu	Jumlah Data
Masuk atau Diterima (rx)	330,87 menit atau 5,5 jam	20250 KB
Masuk atau Diterima (rx)	1 Jam	3672 KB
Masuk atau Diterima (rx)	1 menit	61 KB
Masuk atau Diterima (rx)	1 detik	1,02 KB
Keluar atau Dikirim (tx)	330,87 menit atau 5,5 jam	17470 KB
Masuk atau Diterima (rx)	1 Jam	3168 KB
Keluar atau Dikirim (tx)	1 menit	52 KB
Keluar atau Dikirim (tx)	1 detik	0,88 KB



	rx	tx
bytes	20.25 MiB	17.47 MiB
max	28 kbit/s	12 kbit/s
average	8.36 kbit/s	7.21 kbit/s
min	0 kbit/s	0 kbit/s
packets	117896	144743
max	19 p/s	10 p/s
average	5 p/s	7 p/s
min	0 p/s	0 p/s
time	330.87 minutes	

Gambar 27. Tampilan dari penggunaan data pada perangkat keras

Berdasarkan pengujian perangkat keras melalui *TCP/IP* yang ditunjukkan pada Gambar 27 dan Tabel 2, maka dapat disimpulkan

sistem dapat bekerja secara efisien dalam harga operasional yaitu sebesar 1 KB per detik untuk setiap proses pengiriman dan penerimaan data.

Dalam penerapannya, perangkat keras tidak selalu terhubung ke internet karena Modem menggunakan sinyal UMTS yang tidak stabil pada Kota Pontianak. Masalah lainnya adalah arus yang tidak stabil sehingga *Port USB* tambahan yang menghubungkan *GPS Shield*, Arduino dan Modem pada tidak terdeteksi oleh Mini Komputer.

6. KESIMPULAN

Berdasarkan proses perancangan sistem *real-time* untuk manajemen mobil antarkota menggunakan Node Js berbasis *TCP/IP* dan telah dilakukan implementasi beserta pengujian, maka diperoleh kesimpulan antara lain:

1. Dengan adanya sistem yang telah dibuat, maka koordinat posisi dapat dikirim pada mobil antarkota melalui perangkat Mini Komputer (Raspberry Pi) berbasis *TCP/IP*.
2. Dengan adanya sistem yang telah dibuat, maka manajemen penjemputan jasa transportasi mobil antarkota dapat diterapkan pada *client-server* berbasis *TCP/IP* dengan menggunakan Node JS. Sistem menggunakan metode *soft real-time* yang mengolah koordinat posisi, sehingga dapat dengan cepat menyesuaikan perubahan data dalam waktu tunda yang singkat.
3. Dengan adanya sistem yang telah dibuat, maka dapat digunakan sistem informasi berbasis *TCP/IP* pada jasa transportasi mobil antarkota untuk manajemen dan pelayanan penumpang menggunakan Google Maps versi ke-3.

7. SARAN

Berdasarkan penelitian yang telah dilakukan pada sistem *real-time* untuk manajemen mobil antarkota menggunakan Node Js berbasis *TCP/IP* maka diperoleh saran untuk penelitian lebih lanjut yaitu:

1. Memperhatikan faktor estimasi waktu kedatangan mobil antarkota dengan menggunakan perubahan koordinat posisi mobil antarkota menuju koordinat posisi penumpang.
2. Meninjau kembali keamanan dari sistem yang dibuat karena pada penelitian ini

- dibatasi dalam pembahasan mengenai keamanan menggunakan Node Js.
3. Pendaftaran data penumpang baru sebaiknya menggunakan ketentuan secara bebas membuat akun dari halaman utama *website*.
 4. Untuk mengatasi kendala pada sinyal UMTS diharapkan penelitian selanjutnya menggunakan antena tambahan untuk Modem.
 5. Kendala yang disebabkan oleh arus listrik diharapkan dapat diatasi dengan menggunakan sumber listrik yang lebih stabil pada mobil.

DAFTAR PUSTAKA

- [1] Saputra, Darma., 2013 *Optimalisasi Kebutuhan Angkutan Umum (Taksi Dan Bus) Rute Pontianak-Sintang, Pontianak-Nanga Pinoh Dan Pontianak-Putussibau. Jurnal UNTAN 2013.* <http://jurnal.untan.ac.id/index.php/JMHMS/article/download/2159/2101>, diakses tanggal 1 Juni 2014
- [2] Harijanto, Frenki., 2008, *Sistem Pemantauan Posisi Mobil Menggunakan GPS Berbasis Radio. Jurnal STIKOM Surabaya.* 1 Maret 2008. <http://puslit2.petra.ac.id/ejournal/index.php/tek/article/download/16924/16910>, diakses tanggal 1 Juni 2014.
- [3] Laplante, P.A., 2004, *Real-Time Systems Design.* IEEE Press & John Willey and.
- [4] Mulyono, Sri. *Analisis Sistem Alarm Pengaman Mobil Jarak-jauh Via SMS Remote Kontrol Melalui Jaringan GSM dan GPS sebagai Vehicle Tracker Berbasis Mikrokontroller ATmega16. EPrint Universitas Negeri Yogyakarta.* 2012.<http://eprints.uny.ac.id/7910/1/jurnal.pdf>, diakses tanggal 1 Juni 2014.
- [5] D, Andrew K., 2013, *Raspberry Pi Home Automation.* Birmingham: Packt.
- [6] R., Rohit., 2013, *Socket.IO Real-time Web.* Birmingham: Packt.

