

IMPLEMENTASI INTERKONEKSI JARINGAN ANTAR HALTE BUS MENGUNAKAN INTERNET PADA HALTE

IMPLEMENTATION OF NETWORK INTERCONNECTION BETWEEN BUS STOP USING INTERCONNECTION NETWORK ON SMART BUS STOP

Rizki Satya Pradana¹, Angga Rusdinar², Iswahyudi Hidayat³

^{1,2,3}Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

¹rizkisyatya@students.telkomuniversity.ac.id, ²anggarusdinar@telkomuniversity.ac.id,

³iswahyudihidayat@telkomuniversity.ac.id

. Abstrak

Transportasi merupakan sebuah kebutuhan yang sangat penting bagi setiap orang saat ini. Dengan adanya transportasi ini, akan memudahkan pengguna dalam berpergian menuju tempat yang ingin dituju. Tetapi saat ini banyak pengguna transportasi yang lebih memilih transportasi pribadi sehingga hal ini menimbulkan sebuah masalah baru yaitu kemacetan. Ini diakibatkan karena terlalu banyaknya pengguna transportasi pribadi di jalan raya. Oleh karena itu pemerintah memberikan sebuah himbauan kepada masyarakat untuk menggunakan layanan jasa transportasi untuk mengurangi masalah kemacetan ini. Tetapi minat pada masyarakat terhadap penggunaan layanan jasa transportasi masih kurang dikarenakan masih kurangnya ketepatan informasi yang diberikan oleh layanan jasa transportasi.

Dengan adanya kondisi ini penulis tertarik untuk membuat sebuah inovasi untuk menanggulangi masalah tersebut, dengan cara pembuatan sebuah halte atau tempat pemberhentian yaitu "halte pintar", yang didalamnya terdapat sebuah sistem informasi estimasi kedatangan transportasi. Sistem ini bekerja melalui web dan berfungsi untuk memberikan informasi tentang pemberitahuan estimasi kedatangan yang lebih moderen dan presisi. Dengan adanya sistem ini maka akan meningkatkan minat pengguna layanan jasa transportasi sehingga masalah tingkat kemacetan akan berkurang .

Parameter keberhasilan dari penelitian ini adalah mampu mengirim dan menerima data dengan adanya sedikit error data yang dikirim dan diterima pada web server yang nantinya akan ditampilkan pada halaman web.

Kata Kunci : Halte pintar, Sistem Informasi estimasi kedatangan Transportasi, Web server

Abstracts

Transportation is a very important requirement for everyone today. With this transportasi, will facilitate the user in traveling to where you want to go. But now there are many transportation users who prefer private transportation so that this raises a new problem that is the bottleneck. This is caused by too many private transportation users on the highway. Therefore, the government gives an appeal to the community to use transportation service to solve this congestion problem. However, public interest in the use of transportation services is still lacking due to the lack of accurate information provided by transportation services.

Given this condition the authors are interested in making an innovation to tackle the problem, by way of making a stop, is a "smart stop", in which there is an information system of transportation estimated arrival . This system works over the web and serves to provide information on more modern and precision arrival estimation notices. With this system will increase the user interest of transportation service so that the problem of congestion level will decrease.

The success parameter of this research is able to send and receive data without less error data sent and received on web server which will be displayed on web page.

Keywords: Smart stop, Information Systems of transportation arrival , Web server

1. Pendahuluan

Memasuki era yang semakin modern ini, Transportasi merupakan sebuah kebutuhan yang sangat penting bagi setiap orang. Transportasi berguna untuk memudahkan manusia dalam melakukan aktivitas sehari-hari yaitu mempercepat waktu yang dibutuhkan dalam berpergian dari suatu tempat ke tempat yang ingin dituju. Saat ini sudah banyak masyarakat yang memiliki transportasi pribadinya masing-masing. Jumlah dan pertumbuhan kendaraan transportasi di Indonesia sangat tinggi, hal ini menyebabkan terjadinya kemacetan dimana-mana.

Layanan jasa transportasi yang layak dan efektif merupakan bagian yang sangat penting. Saat ini alat transportasi yang dipakai tidak hanya dituntut untuk dapat mengantarkan orang maupun barang dengan cepat akan tetapi juga menuntut kenyamanan, keamanan dan kelayakan dari transportasi itu sendiri. Tetapi hal yang terjadi saat ini adalah pengguna layanan transportasi yang masih harus menunggu berjam-jam di tempat pemberhentian karena kekurangan informasi tentang kedatangan transportasi sehingga takut tertinggal oleh transportasi yang ingin digunakan. Hal ini menyebabkan ketidaknyamanan bagi para pengguna dan akan berdampak pada turunnya minat pengguna layanan jasa transportasi, sehingga menyebabkan sebagian masyarakat lebih memilih berpergian menggunakan transportasi pribadinya sendiri. Oleh karena itu dibutuhkan suatu cara untuk menangani permasalahan tersebut.

Salah satu terobosan atau inovasi yang dapat membantu menyelesaikan masalah tersebut adalah dengan meningkatkan minat pengguna jasa layanan transportasi. Maka muncul sebuah gagasan yaitu pembuatan "Halte pintar" yang digunakan untuk bus, dimana tempat pemberhentian ini akan memberikan sebuah informasi estimasi kedatangan yang lebih presisi berdasarkan estimasi yang diprediksi oleh *google-maps* yang datanya akan diolah secara otomatis ketika bus tiba pada halte. Data tersebut akan diunggah ke sebuah web, sehingga pengguna layanan jasa transportasi akan mendapatkan informasi yang cukup tentang estimasi kedatangan dan dapat mengaksesnya melalui *device* pengguna masing-masing tentang jadwal kedatangan bus yang ingin digunakan pada pemberhentian. Hal ini akan meningkatkan minat pengguna dalam menggunakan layanan jasa transportasi, sehingga membantu meningkatkan jumlah pengguna layanan jasa transportasi bus dan tingkat perkembangan penggunaan transportasi pribadi akan berkurang.

Maka dengan munculnya solusi dan masalah tersebut, penulis mencoba membuat sesuatu yang dapat menyelesaikan masalah tersebut yaitu **"Implementasi Interkoneksi Jaringan Antar Halte Bus Menggunakan Internet Pada Halte Pintar"**.

2. Dasar Teori

2.1. Cara Kerja Konsep Solusi

Berdasarkan rumusan masalah, cara kerja konsep solusi adalah sebagai berikut: akan dibuat pemrograman *web* untuk "halte pintar" yang memuat sistem informasi estimasi kedatangan yang berbentuk *countdown*/hitungan mundur, dimana waktu kedatangannya telah disesuaikan dengan estimasi waktu perjalanan yang telah dikalkulasikan oleh *google-maps* dan sudah disesuaikan dengan rute yang ditentukan. Kemudian data akan diolah dalam sebuah *single board computer*(SBC) yaitu raspberry-pi 3 model b.

Sistem ini memiliki input berupa data yang diterima dari RF-ID dan output berupa tampilan display pada monitor yang diolah dalam sebuah *single board computer*. Pengolahan data pada *single board computer* dalam bentuk bahasa pemrograman PHP, yang didalamnya terdapat script untuk menghubungkan RF-ID dengan *single board computer* dan membuat sebuah tampilan yang akan ditampilkan dalam monitor. Ketika id yang didapat ketika transportasi datang pada sebuah halte teridentifikasi oleh RF-ID pada sebuah halte maka data tersebut akan diolah oleh *single board computer* untuk melakukan perubahan

waktu untuk menghitung estimasi kedatangan transportasi untuk mencapai halte selanjutnya yang kemudian data tersebut akan ditampilkan pada monitor halte.

2.1.1. Halte Pintar

Halte pintar adalah sebuah tempat pemberhentian transportasi dimana telah dirancang dan didesain agar dapat melakukan penyesuaian diri, pemberi edukasi, dapat melakukan multifungsi serta memiliki *zero energy* yang tujuan utamanya adalah untuk meningkatkan kenyamanan pengguna transportasi

2.1.2. HTML(Hypertext Markup Language)

HTML(Hypertext Markup Language) adalah bahasa standard yang digunakan untuk menampilkan halaman web.

2.1.3. CSS(Cascading Style Sheet)

CSS(Cascading Style Sheet) digunakan untuk mengatur gaya tampilan/layout dari halaman web agar lebih menarik. CSS ini telah didukung oleh semua web browser karena CSS telah distandarkan oleh World Wide Web Consortium (W3C).

2.1.4. Bootstrap

Bootstrap adalah framework yang berfungsi untuk membangun desain web secara responsif dan interaktif. Yang dimaksud responsif disini untuk membuat tampilan web yang dibuat agar tampilannya menyesuaikan ukuran layar dari browser yang digunakan

2.1.5. JavaScript dan JQuery

JavaScript adalah sebuah bahasa skrip yang digunakan pada website agar website menjadi lebih interaktif. JavaScript ini memiliki jenis bahasa pemrograman *client side*, dimana semua sintaks dan perintah program dijalankan di web browser. Pada penelitian ini JavaScript mengambil library dari JQuery sehingga lebih sederhana dan mudah.

2.1.6. WebServer XAMPP

Web server diperlukan untuk menguji apakah aplikasi web berjalan dengan baik atau tidak. Web server ini adalah tempat untuk menyimpan aplikasi web dan kemudian mengaksesnya melalui internet. Setiap perubahan yang terjadi akan di upload ke webserver yang kemudian akan diperiksa apakah sudah sesuai yang anda inginkan atau belum.

Web Server ini dibutuhkan karena berperan sebagai *server side script*, dimana semua sintaks dan perintah program akan diproses oleh web server. XAMPP adalah sebuah solusi yang praktis dimana didalam XAMPP sudah terdapat Apache, MySQL, PHP tanpa perlu menginstal semua satu-persatu. Kata XAMPP sendiri berasal dari:

- 1) X yang berarti *cross platform* karena XAMPP dapat dijalankan di windows, Linux, Mac dsb.
- 2) A yang berarti Apache sebagai *web server*-nya
- 3) M yang berarti MySQL sebagai database management system(DBMS) nya
- 4) PP yang berarti PHP dan Perl sebagai bahasa yang didukungnya

2.1.7. Apache

Apache adalah sebuah web server yang tugasnya bertanggung jawab pada request dan response HTTP sehingga dapat memfungsikan suatu situs web. Apache ini memiliki fitur canggih yaitu pesan kesalahan yang dapat dikonfigur, autentikasi berbasis data dan lainnya.

2.1.8. MySQL

MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para developer pemrograman aplikasi web. DBMS (*database management system*) sendiri adalah aplikasi yang dipakai untuk mengolah basis data.

2.1.9. PHP

PHP adalah singkatan dari PHP Hypertext Preprocessor. PHP sudah menjadi bahasa scripting yang umum dan telah banyak digunakan bagi developer pemrograman web.

2.1.10.Proxy Server Reverse

proxy adalah *server* yang menyediakan suatu layanan untuk meneruskan setiap permintaan user kepada server lain yang terdapat di internet. Salah satu jenis *proxy server* adalah *proxy server reverse*, dimana *reverse proxy* ini adalah salah satu jenis *proxy server* yang berfungsi sebagai perantara antara *client* dengan satu atau lebih *web server*. *Reverse proxy* ini menjadi sebuah perantara *user* ke *user* di internet terhadap akses ke *web-web server* yang berada pada *local area network*.

2.1.11.Pengenalan RF-ID

RFID (Radio Frequency Identification) adalah teknologi yang menggunakan gelombang radio untuk mengidentifikasi suatu objek secara otomatis. RFID dapat di setting hanya dapat dibaca saja (Read Only) atau dapat dibaca dan ditulis (Read/Write), tidak memerlukan kontak langsung maupun jalur cahaya

untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi.

2.1.12. RF-ID Reader

RF-ID Reader adalah perangkat *input* data yang membaca data dari media penyimpanan berbentuk kartu baik dengan *barcode*, *magnetic strip*, *Computer chip* atau media penyimpanan medium lainnya. Ada dua macam RF-ID reader yaitu reader pasif (PRAT) dan reader aktif (ARPT).

2.1.13. RF-ID Tag

Tag RFID adalah device yang dibuat dari rangkaian elektronika dan antena yang terintegrasi di dalam rangkaian tersebut. Rangkaian elektronik dari tag RFID umumnya memiliki memori sehingga tag ini mempunyai kemampuan untuk menyimpan data.

2.1.14. Single Board Computer

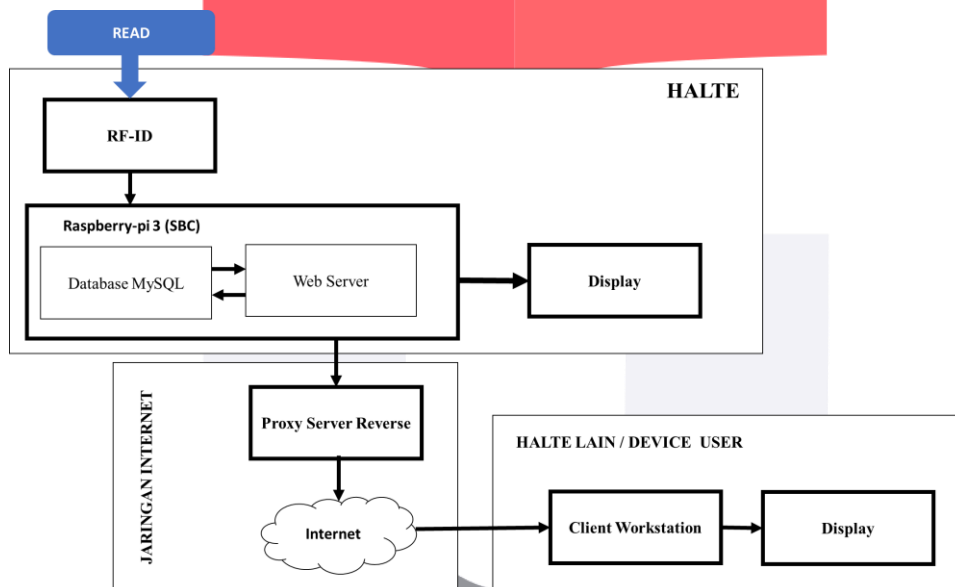
Single Board Computer (SBC) adalah komputer yang lengkap yang dibangun di *single board computer* atau papan sirkuit tunggal dengan *microprocessor*, *memory*, *input/output* (I/O).

3. Perancangan

3.1. Desain Sistem

Sistem yang akan dibuat pada tugas akhir ini yaitu adalah sistem informasi estimasi kedatangan transportasi yang berbasis web. Sistem ini bekerja dengan cara memanipulasi data pada database yang terdapat pada MySQL, yang nantinya data tersebut akan diolah pada script yang menggunakan bahasa pemrograman PHP, HTML, JavaScript, dan CSS.

3.1.1. Diagram Blok



Gambar 1. Diagram Blok Sistem

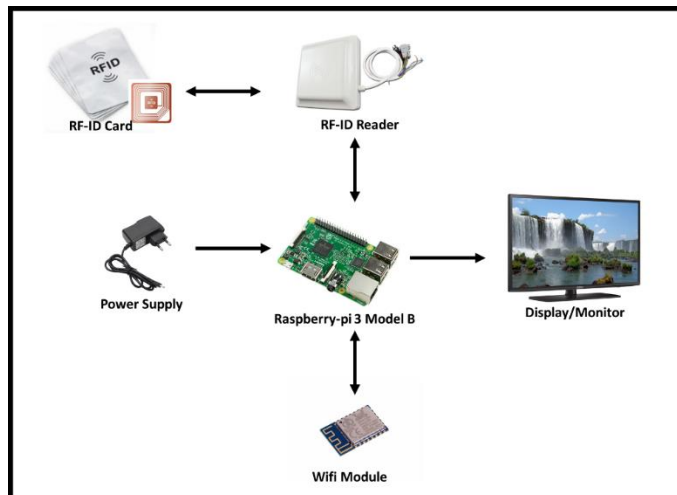
sistem dimulai dari pembacaan data pada database yang terdapat pada MySQL yang kemudian akan diproses di web server untuk request dan response HTTP sehingga dapat memfungsikan suatu situs web yang masih bersifat localhost server. Kemudian situs web tersebut dibaca oleh client workstation yang akan tampil pada display. Fungsi dari proxy server sendiri adalah untuk membuat penghubung antar pengakses localhost ke internet sehingga dapat diakses secara publik.

3.2. Kebutuhan Sitem

Program yang digunakan dalam pembuatan website adalah sebagai berikut:

1. Sistem Operasi: Windows (yang digunakan), Linux, Mac
2. Software: XAMPP digunakan sebagai Web Sever
MySQL digunakan sebagai Database
Sublime Text sebagai Text Editor
Ngrok sebagai *proxy server reverse*

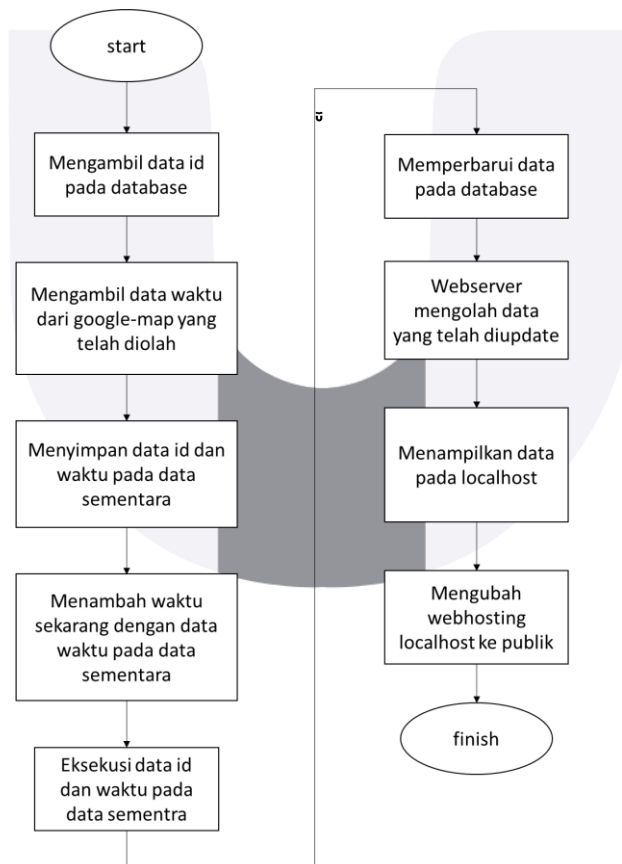
3.3. Desain Perangkat Keras



Gambar 2. Desain Perangkat Keras pada halte

Sebuah mikrokomputer digunakan sebagai pengolah pengoperasian sistem yang akan dibuat. Mikrokomputer akan menjadi pusat dan penghubung antara masukan dan keluaran pada sistem. Mikrokomputer yang akan digunakan adalah Raspberry-pi 3. *Power supply* digunakan sebagai penyuplai sumber listrik untuk mikrokomputer. Wifi module untuk sebagai penghubung mikrokomputer dengan internet yang sudah tertanam pada raspberry-pi seri 3. Monitor untuk menampilkan hasil akhir yang telah diolah pada mikrokomputer yang pada sistem informasi estimasi kedatangan transportasi ini berbentuk halaman web.

3.4. Desain Perangkat Lunak



Gambar 3. Desain Perangkat Lunak

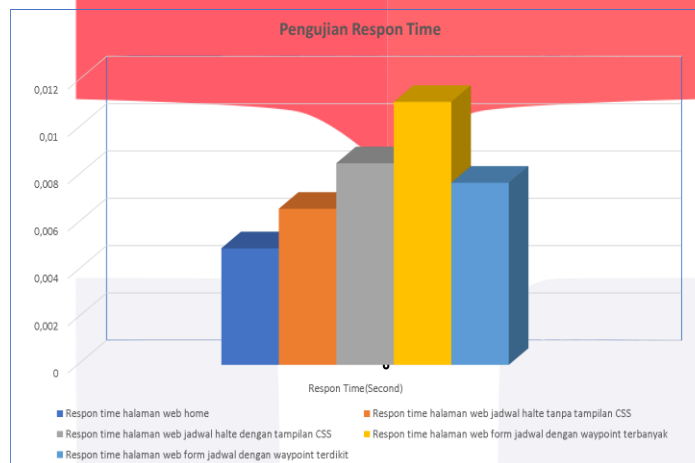
Sistem Informasi estimasi kedatangan transportasi dimulai dengan pengambilan data pada tabel database MySQL yang telah dibuat. Data yang diambil adalah id yaitu sebagai baris data mana yang akan di update nantinya. Kemudian waktu estimasi akan diambil dari estimasi waktu yang dibutuhkan dari *origin*/awal pemberangkatan sampai ke tujuan yang telah melalui beberapa *waypoint*/titik jalur sesuai jalur yang ditentukan. Setelah mendapat waktu estimasi yang diambil dari google-maps, kemudian waktu tersebut akan dijumlahkan dengan waktu saat ini, sehingga didapatkan sebuah estimasi waktu transportasi untuk mencapai tujuan yang dituju. Waktu tersebut akan digunakan untuk *mengupdate*/memperbarui data yang terdapat pada tabel database pada MySQL ketika tombol submit telah ditekan.

Proses selanjutnya adalah pemanggilan database pada tabel MySQL oleh *web server* yang berfungsi untuk mengolah semua *script* yang telah dibuat. Karena *web server* yang digunakan masih bersifat localhost, maka akan di lakukan penyambungan dengan *proxy server reverse* yang gunanya untuk membuat web server yang bersifat localhost sehingga dapat diakses secara publik. Dengan begitu semua pengguna dapat mengakses web server tersebut yang ditampilkan pada *workstation* mereka masing-masing.

3.5. Pengujian Respon Time

Pengujian ini dilakukan untuk mengetahui berapa lama waktu yang dibutuhkan oleh setiap halaman web untuk membaca semua *script* yang terdapat didalam halaman tersebut. Akan dilakukan 10 kali percobaan setiap pengujian untuk mengukur berapa lama waktu *respon time* yang dibutuhkan.

3.5.1. Grafik Perbandingan Pengujian Respon Time



Gambar 4. Grafik Perbandingan Respon Time

Gambar 4 adalah grafik perbandingan dari beberapa pengujian dari respon time terhadap halaman web yang diukur dalam satuan *second*/detik. Dapat dilihat pengujian 1 memiliki waktu respon time tercepat dikarenakan kompleksitas dan jumlah data lebih sedikit dibandingkan halaman web lainnya.

Selanjutnya adalah perbandingan antara pengujian ke 2 dan 3 dimana menguji halaman web jadwal halte yang dibedakan menggunakan ada atau tidaknya CSS(*Cascading Style Sheet*), dapat dilihat pada grafik respon time dengan menggunakan CSS akan lebih lama dibandingkan dengan halaman web yang tidak menggunakan CSS dikarenakan bertambahnya jumlah data yang terdapat didalam script, sehingga menyebabkan waktu proses yang dibutuhkan bertambah

Terakhir adalah perbandingan pengujian lamanya respon time terhadap banyak nya jumlah *waypoint*/titik rute pada halaman web form jadwal. Dapat dilihat pada grafik halaman web yang membuat lebih banyak *waypoint*/ titik rute akan membutuhkan respon time yang lebih lama dikarenakan data penjumlahan waktu yang dibutuhkan setiap rute akan bertambah sesuai dengan banyaknya rute yang diminta. Ini menyebabkan penambahan jumlah data pada *script* sehingga membutuhkan waktu respon time yang lebih lama

Dari beberapa pengujian tersebut, penguji dapat menyimpulkan bahwa semakin banyak data yang diproses dalam sebuah *script* maka akan semakin lama *respon time* terhadap halaman web yang dimuat.

3.6. Pengujian Web Server Load Performance Stress

Pengujian ini dilakukan untuk melihat tingkat stress pada suatu aplikasi web yang melewati media http/https pada waktu yang bersamaan. Pengujian ini dilakukan dengan cara mensimulasikan permintaan HTTP yang dihasilkan oleh beberapa *user/pengguna* yang mengakses *webserver* ini secara bersamaan. Penulis menggunakan sebuah software yaitu “Webserver stress tool 8” sebagai aplikasi pembantu dalam melakukan pengujian..

3.6.1. Pengujian Web Stress dengan 5 pengguna setiap 10 detik

Pengujian ini dilakukan selama 1 menit, yang setiap 10 detiknya *webserver* akan diakses oleh 5 pengguna. Yang artinya akan ada ~1800pengguna/jam yang akan mengakses *webserver* tersebut. Pengujian ini menguji 4 buah *url* yaitu halaman home, jadwal halte_1, jadwal halte_2 dan jadwal_halte3.

```

** Results per URL for complete test **

URL#1 (home): Average Click Time 817 ms, 7 Clicks, 0 Errors
URL#2 (jadwal_halte_1): Average Click Time 868 ms, 5 Clicks, 0 Errors
URL#3 (jadwal_halte_2): Average Click Time 1.011 ms, 4 Clicks, 0 Errors
URL#4 (jadwal_halte_3): Average Click Time 825 ms, 3 Clicks, 0 Errors

Total Number of Clicks: 19 (0 Errors)
Average Click Time of all URLs: 872 ms

```

Gambar 5. Hasil pengujian web stress dengan 5 pengguna setiap 10 detik

Dapat disimpulkan bahwa jika 5 pengguna mengakses *webserver* setiap 10 detik maka keadaan *webserver* belum terjadi adanya tingkat stress/masih berjalan dengan normal tanpa adanya error.

3.6.2. Pengujian Web Stress dengan 20 pengguna setiap 10 detik

Pengujian ini dilakukan selama 1 menit, yang setiap 10 detiknya *webserver* akan diakses oleh 20 pengguna. Yang artinya akan ada ~7200pengguna/jam yang akan mengakses *webserver* tersebut. Pengujian ini menguji 4 buah *url* yaitu halaman home, jadwal halte_1, jadwal halte_2 dan jadwal_halte3. Berikut adalah hasil pengujiannya:

```

** Results per URL for complete test **

URL#1 (home): Average Click Time 833 ms, 20 Clicks, 10 Errors
URL#2 (jadwal_halte_1): Average Click Time 824 ms, 17 Clicks, 10 Errors
URL#3 (jadwal_halte_2): Average Click Time 810 ms, 15 Clicks, 9 Errors
URL#4 (jadwal_halte_3): Average Click Time 843 ms, 15 Clicks, 9 Errors

Total Number of Clicks: 67 (38 Errors)
Average Click Time of all URLs: 358 ms

```

Gambar 6. Hasil pengujian web stress dengan 20 pengguna setiap 10 detik

Dapat disimpulkan bahwa tingkat stress dari *webserver* ini adalah 18 pengguna setiap 10 detik. Tingkat stress dari *webserver* sudah sangat tinggi pada kondisi ini yang mengakibatkan terjadinya eror sehingga tidak dapat memuat halaman web pada sistem ini.

4. Kesimpulan

Berdasarkan hasil penelitian dan analisis sistem informasi estimasi kedatangan transportasi ini didapat kesimpulan sebagai berikut:

1. Sistem informasi estimasi kedatangan transportasi ini telah dapat bekerja dengan baik sesuai dengan data estimasi yang diambil dari *google-maps* sehingga membantu pengguna transportasi mengetahui waktu kedatangan transportasi lebih presisi.
2. Sistem informasi estimasi kedatangan transportasi bekerja dengan baik dan kecilnya nilai eror serta memiliki respon time halaman page yang relatif cepat.
3. *Webserver* pada sistem informasi estimasi kedatangan transportasi memiliki tingkat stres *webserver* ~18 pengguna dalam waktu 10 detik.

Daftar Pustaka :

- [1] Priyanto Hidayatullah, Jauhari Khairul Kawistara (2017), "Pemrograman WEB Edisi Revisi", Penerbit Informatika, Bandung.
- [2] Jubilee Enterprise (2017), "Otodidak Pemrograman JavaScript", PT Elex Media Komputindo, Jakarta
- [3] Bootstrap CDN. [ONLINE] Available at: <http://getbootstrap.com/getting-started/>. [Accessed 5 Juni 2017]
- [4] Downloading jQuery. [ONLINE] Available at: <http://jquery.com/>. [Accessed 7 Juni 2017]
- [5] Sublime Text 3 Download. [ONLINE] Available at: <https://www.sublimetext.com/>. [Accessed 10 Juni 2017]
- [6] JavaScript tutorial. [ONLINE] Available at: <https://www.w3schools.com/>. [Accessed 12 Juni 2017]
- [7] PHP tutorial. [ONLINE] Available at: <http://www.phpeasytstep.com/mysql/9.html>. [Accessed 21 Juni 2017]
- [8] Google maps APIs. [ONLINE] Available at: <https://developers.google.com/maps/documentation/directions/intro#Waypoints>. [Accessed 26 Juni 2017]
- [9] WebStress Tool 8 tutorial. [ONLINE] Available at: <https://download-cdn.paessler.com/download/webstressmanual.pdf> [Accessed 29 Juni 2017]
- [10] Kompas News. [ONLINE] Available at: <http://otomotif.kompas.com/read/2016/08/20/103100215/Anda.Tahu.Populasi.Kendaraan.di.Indonesia>. [Accessed 08 Agustus 2017]
- [11] Miniatur halte UNDIP halte pintar. [ONLINE] Available at: <http://www.coroflot.com/spainlouis/Miniatur-Halte-UNDIP-HALTE-PINTAR> [Accessed 08 Agustus 2017]
- [12] Smart bus stop prototype aims to 'make waiting fun'. [ONLINE] Available at: <http://www.channelnewsasia.com/news/singapore/smart-bus-stop-prototype-aims-to-make-waiting-fun-7845486> [Accessed 08 Agustus 2017]
- [13] We may have found the world's most high-tech bus stop. [ONLINE] Available at: <https://www.cnbc.com/2017/03/07/singapore-smart-nation-this-bus-stop-is-transforming-the-daily-commute.html> [Accessed 08 Agustus 2017]