Sistem Penampil Stok Kendaraan pada *Carpool* di Perusahaan Otomotif

Melvin Nurrohman, Kiki Prawiroredjo, E. Shintadewi Julian
Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Trisakti.

Jl. Kyai Tapa No. 1 Grogol DKI Jakarta 60111, Indonesia

E-mail: melvinnurrohman@gmail.com

ABSTRACT

The prototype of vehicle stock information display system in PT Krama Yudha Tiga Berlian Motor's carpool is based on the problem of delivering real-time vehicle stock information and inefficient processing of vehicle stock data. The function of this system is to convey information of the number of vehicle stock, the remaining parking slots and vehicles in and out in realtime. This prototype has Android applications, desktop applications, website applications and automatic portals. Based on the results of testing the Android application, the average scan time for barcode scanning of vehicles in and out carpool is 2,47 seconds and for the application of barcode scanning of vehicles locations in carpool is 2,36 seconds. The average transition time between web pages is 20,16 seconds. The average response time for gate in portal is 1,36 seconds and gate out portal is 3,36 seconds.

Keywords: Android, NodeMCU, information, Stock, Vehicle

ABSTRAK

Prototipe sistem penampil stok kendaraan di carpool PT Krama Yudha Tiga Berlian Motor didasari dari masalah penyampaian informasi stok kendaraan yang tidak real time dan tidak efisiennya mengolah data stok kendaraan. Prototipe sistem ini bertujuan untuk menyampaikan informasi jumlah stok kendaraan, sisa slot parkir, kendaraan masuk dan keluar secara real time. Prototipe ini memiliki aplikasi Android, aplikasi desktop, aplikasi website dan portal otomatis. Berdasarkan hasil pengujian scan barcode kendaraan pada aplikasi Android, ratarata waktu scan 2,47 detik untuk aplikasi scan barcode kendaraan masuk dan keluar dan 2,36 detik untuk aplikasi scan barcode letak kendaraan di carpool. Waktu rata-rata transisi antar halaman website adalah 20,16 detik. Waktu rata-rata respon portal gate in adalah 1,36 detik dan gate out 3,36 detik.

Kata kunci: Android, NodeMCU, Informasi, Stok, Kendaraan

1. Pendahuluan

PT Krama Yudha Tiga Berlian Motor (PT KTB) merupakan agen tunggal pemegang merek Mitsubishi Fuso di Indonesia dan distributor resmi Mitsubishi Fuso di Indonesia. Sebagai distributor resmi Mitsubishi Fuso di Indonesia PT KTB bertugas untuk menerima produk kendaraan Mitsubishi Fuso yaitu Colt Diesel dan Fuso *Truck* dari PT Krama Yudha Ratu Motor (PT KRM) dan untuk menjual produk Mitsubishi Fuso di Indonesia melalui perwakilan *dealer* Mitsubishi Fuso yang ada di Indonesia.

Penerimaan kendaraan Mitsubishi Fuso dari PT KRM ke PT KTB diterima oleh *Vehicle Control Departement* (VCD) PT KTB. Lokasi VCD dekat dengan lokasi pabrik perakitan produk Mitsubishi Fuso yaitu PT KRM. VCD bertugas untuk menerima kendaraan dari PT KRM, mengatur dan menyerahkan kendaraan kepada perwakilan *dealer* Mitsubishi Fuso. Untuk menunjang pekerjaan VCD memiliki *carpool* atau gudang mobil dengan area seluas 49,748 m² untuk tempat penyimpanan kendaraan sebelum diserahkan kepada perwakilan *dealer* Mitsubishi Fuso.

Dalam mengatur area lahan kendaraan, staf VCD menggunakan kode area yang dapat membantu dalam pencarian lokasi kendaraan. Pada Gambar 1 diperlihatkan contoh *layout* area A yang memiliki kode untuk menandai posisi kendaraan. Ketika kendaraan dipindahkan dari satu lokasi ke lokasi lainnya maka staf akan menulis perubahan lokasi tersebut di kartu kunci kendaraan di sebelah *barcode* nomor rangka kendaraan. Setelah staf menulis lokasi terbaru, maka staf akan memperbarui data di file *Microsoft Excel* secara manual berdasarkan data nomor rangka. Pembaruan lokasi kendaraan menggunakan *Microsoft Excel* ini dinilai kurang efisien karena ukuran file yang besar, rawan akan kesalahan memasukkan nomor rangka kendaraan, dan memakan waktu dalam memperbarui data. Untuk itu diperlukan *software* atau sistem yang bisa mempersingkat waktu dan mengurangi kesalahan memasukkan nomor rangka kendaraan. Kebutuhan informasi secara *real time* diperlukan untuk mengetahui sisa lahan yang terpakai sangat penting untuk dapat mengatur penempatan kendaraan dengan cepat.

A11	A21	A31	A41
A12	A22	A32	A42
A13	A23	A33	A43
A14	A24	A34	A44
A15	A25	A35	A45
A16	A26	A36	A46
A17	A27	A37	A47
A18	A28	A38	A48
A19	A29	A39	A49
A110	A210	A310	A410

Keterangan Kode

A = Kode Area

digit pertama = Kolom

2 digit terakhir = Baris

Gambar 1 Contoh Layout Area A

Sistem yang ada saat ini juga masih memiliki kekurangan pada keamanan stok kendaraan, dimana unit kendaraan yang masuk dan keluar tidak terpantau secara detail oleh pihak keamanan.

Berdasarkan penjelasan di atas, maka dirancang suatu sistem penampil stok kendaraan di *carpool* PT KTB yang dapat menyampaikan informasi total stok kendaraan, stok kendaraan di setiap area, slot parkir tersedia, jumlah kendaraan keluar dan masuk secara *real time*, dan berbasis *web service*. Selain itu sistem yang dirancang akan menggunakan *smartphone* Android untuk memindai data kendaraan dan aplikasi *desktop* untuk melihat data stok, mencari data kendaraan berdasarkan nomor rangka kendaraan, dan untuk mencari riwayat aktivitas berdasarkan bulan, tanggal dan per hari. Sistem yang dibangun ini menggunakan bahasa pemograman C# yang diolah di *software* Visual Studio dan juga *data base MySQL. Data base* adalah kumpulan berkas yang saling berelasi, relasi tersebut biasanya ditunjukkan dengan kunci dari tiap berkas yang ada [1].

2. Kajian Pustaka

Pada penelitian yang telah dilakukan oleh Fakri dirancang sistem parkir cerdas dengan memanfaatkan kartu *Radio Frequency Identification* (RFID) untuk membuka motor servo yang dipasangi palang parkir dan Arduino sebagai pengolah datanya. Pada gedung parkir juga ditempatkan sensor cahaya yaitu *Light Emitting Diode* (LED) untuk memberi informasi ketersediaan tempat parkir pada *personal computer* (PC) menggunakan program Visual Basic [2].

Penelitian yang telah dilakukan oleh [3] menghasilkan suatu sistem simulasi parkir otomatis untuk menentukan tempat parkir yang kosong di antara sepuluh tempat parkir. Sistem ini terdiri atas rangkaian sensor LDR, mikrokontroler AT89S51, rangkaian motor *stepper*, dan *display seven segment* [3].

Pada [4] dirancang *smart parking* berbasis mikrokontroler yang dapat menginformasikan dan mengarahkan pengendara mobil ke area parkir yang kosong. Pada *prototipe* ini *Liquid Crystal Display* (LCD) akan menampilkan slot parkir yang sudah terisi dan slot parkir yang kosong dan pemantauan melalui *Smartphone* dengan menggunakan aplikasi Blynk [4].

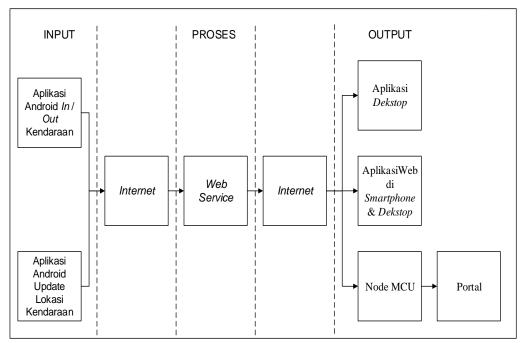
Pada penelitian yang dilakukan oleh [5] dirancang sistem parkir dengan rekomendasi lokasi parkir. Pengendara akan menerima *printout* tiket dari *printer thermal* yang bertuliskan lokasi parkir yang disarankan yaitu yang terdekat dengan pintu masuk bangunan atau fasilitas.

3. Metode Penelitian

3.1 Diagram Blok Perancangan Sistem

Diagram blok rancangan sistem penampil stok kendaraan di *carpool* PT Krama Yudha Tiga Berlian Motor terdapat pada Gambar 2 yang terdiri dari:

 Aplikasi Android In / Out kendaraan yang diberi nama Melv_InOutBarcode ini berfungsi untuk memindai barcode kendaraan yang masuk atau keluar dari carpool. Aplikasi ini dipasang pada smartphone Android yang sudah dikoneksikan dengan internet. 2. Aplikasi Android *update* lokasi kendaraan yang diberi nama Malv_RotBarcode ini berfungsi untuk memasukan data lokasi parkir untuk kendaraan yang baru masuk ke *carpool* maupun kendaraan yang pindah lokasi parkir. Aplikasi ini dipasang pada *smartphone* Android yang sudah dikoneksikan dengan internet.

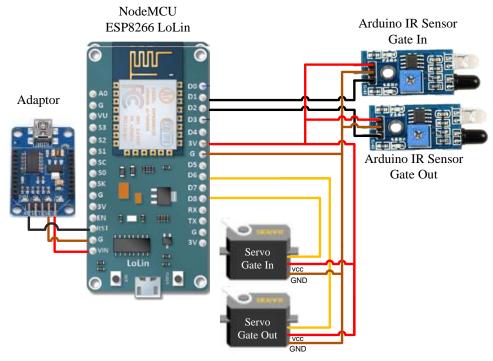


Gambar 2 Diagram Blok Rancangan Sistem Penampil Stok Kendaraan

- 3. Web service digunakan untuk mengumpulkan aplikasi data dan software agar dapat diakses oleh berbagai piranti serta menjadi penghubung antara aplikasi Android, aplikasi desktop, NodeMCU LoLin, dan website menggunakan media komunikasi internet.
- 4. Aplikasi *desktop* digunakan untuk melihat informasi data kendaraan.
- 5. *Website* untuk menampilkan informasi terkait stok kendaraan dan lokasi parkir kendaraan yang dapat diakses dari *browser* pada *desktop* atau *smartphone*.
- 6. NodeMCU digunakan untuk koneksi antara sistem dengan portal secara wireless.
- 7. Portal digunakan untuk meningkatkan keamanan dan ketertiban keluar masuknya kendaraan yang dapat diakses melalui aplikasi Android.

3.2 Perangkat Keras Sistem Penampil Stok Kendaraan di Carpool PT KTB

Gambar 3 memperlihatkan perangkat keras portal otomatis untuk keluar dan masuk kendaraan. Portal otomatis ini terdiri dari modul NodeMCU, dua buah modul sensor *infrared* dan dua buah motor servo. Karena NodeMCU bekerja pada tegangan 5 V maka pada alat ini diperlukan adaptor guna menurunkan tegangan 220V ke 5V.

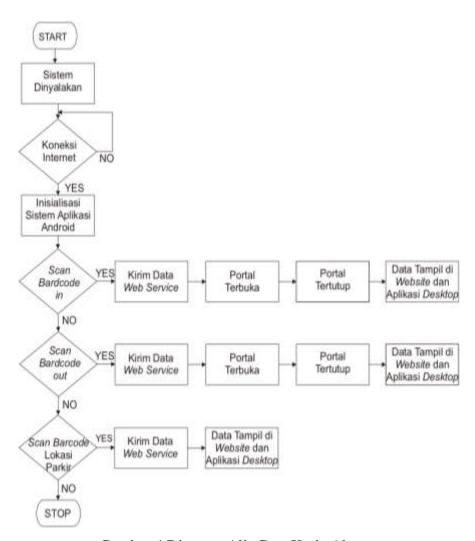


Gambar 3 Perangkat Keras Sistem Penampil Stok Kendaraan di Carpool PT KTB

Fungsi dari NodeMCU adalah untuk meneruskan perintah dari web service yang berupa data untuk membuka portal dan memproses modul infrared untuk dapat menutup portal setelah kendaraan berhasil melewati portal.

3.3 Diagram Alir Cara Kerja Alat

Cara kerja *prototipe* sistem penampil stok kendaraan di *carpool* PT KTB digambarkan dalam bentuk diagram alir seperti dapat dilihat pada Gambar 4. Cara kerja sistem ini dimulai dari koneksi internet sebagai media komunikasi semua piranti.

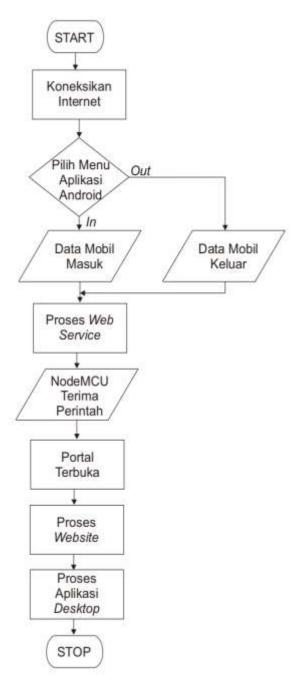


Gambar 4 Diagram Alir Cara Kerja Alat

3.4 Realisasi Program Aplikasi Android

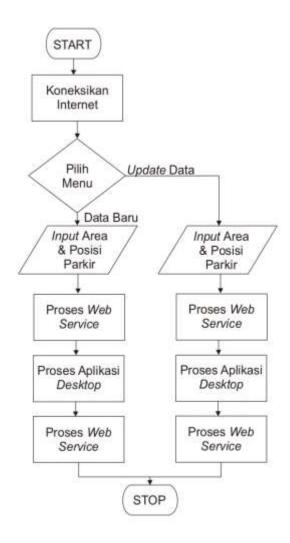
3.4.1 Diagram Alir Aplikasi Scan Barcode

Pada Gambar 5 diperlihatkan diagram alir kendaraan masuk dan keluar dimana sistem berkomunikasi melalui jaringan internet. Pada saat mobil hasil produksi masuk, *User* memindai *barcode* kendaraan lalu memilih tombol kendaraan masuk di aplikasi Android *scan barcode*. Data mobil masuk diproses oleh *web service* lalu mengirimkan perintah ke NodeMCU untuk membuka portal. Cara ini berlaku juga untuk mobil keluar.



Gambar 5 Diagram Alir Aplikasi Scan Barcode.

Diagram alir Aplikasi *Update* Lokasi Kendaraan terdapat pada Gambar 6.



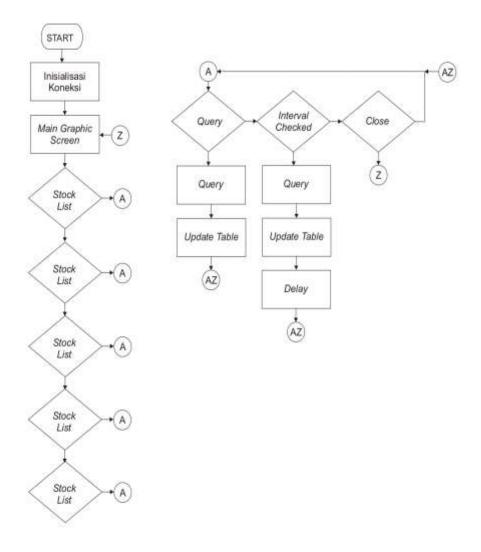
Gambar 6 Realisasi Pembuatan Aplikasi Android di Website Appinventor

Pembuatan aplikasi Android pada sistem ini menggunakan *website* APPInventor dengan mengakses situs http://appinventor.mit.edu lalu klik Create Apps.

3.4.2 Realisasi Program Aplikasi *Dekstop* (Parksys)

Prototipe sistem penampil stok kendaraan di *carpool* PT KTB ini memiliki aplikasi *desktop* yaitu Parksys untuk melihat data kendaraan. Pada Gambar 7 menunjukkan diagram alir realisasi program aplikasi *desktop*. Dimulai dari inisialisasi koneksi internet. *Main graphic screen* terdiri dari beberapa tombol pilihan yaitu *Stock*

List, Search by Chasis, History by Date, History by Day, dan Summary yang berfungsi untuk menampilkan informasi stok kendaraan.



Gambar 7 Diagram Alir Aplikasi Desktop

3.4.3 Realisasi Aplikasi Website

Prototipe sistem penampil stok kendaraan di *carpool* PT KTB ini memiliki aplikasi *website* yang dapat diakses dimanapun menggunakan *browser* yang terhubung ke jaringan internet.

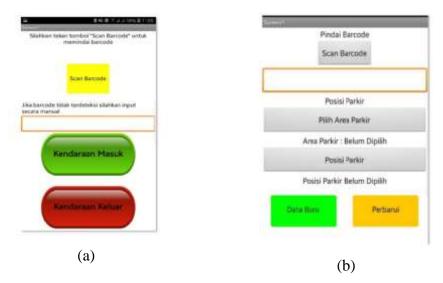
3.4.4 Realisasi Portal

Prototipe sistem penampil stok kendaraan di *carpool* PT KTB ini memiliki portal untuk keamanan dan ketertiban keluar masuk kendaraan hasil produksi dan kendaraan *dealer*. Untuk merealisasikan portal ini digunakan NodeMCU, motor servo dan sensor *infrared*. Portal ini diprogram menggunakan *software* Arduino IDE dengan bahasa pemograman C.

4. Hasil dan Pembahasan

4.1 Pengujian Aplikasi Android Melv_RotBarcode dan Melv_InOutBarcode

Pengujian dilakukan untuk mengetahui rancang bangun sistem penampil stok kendaraan di carpool PT KTB bekerja sesuai atau tidak sesuai dengan perancangan yang dibuat. Bagian-bagian yang diuji yaitu aplikasi Android Melv_RotBarcode dan aplikasi Android Melv_InOutBarcode, fitur dari aplikasi *desktop*, tampilan dan respon di *website*, respon buka dan tutup portal, dan pengujian keseluruhan. Tampilan kedua aplikasi Android terdapat pada Gambar 8.



Gambar 8 Tampilan Aplikasi Android (a) Melv_inOutBarcode dan
(b) Malv_rotBarcode

Dari hasil pengujian tombol-tombol yang ada di aplikasi Parksys semuanya berhasil menampilkan fungsi yang sesuai dengan tiap-tiap tombol yang ditekan. Koneksi internet dibutuhkan untuk mengakses data pada *web service*. Diperlukan koneksi internet yang stabil agar data dapat diakses tanpa kendala.

Hasil pengujian dan analisis aplikasi Android terdapat pada Tabel 1 dan Tabel 2.

Dari hasil pengujian aplikasi Melv_InOutBarcode dan Malv_RotBarcode dengan *sample barcode* nomor rangka kendaraan sebanyak 10 unit, semuanya berhasil memindai data dari *barcode* ke aplikasi *text box* .

Tabel 1 Hasil Pengujian Aplikasi Melv_InOutBarcode

No.	No Rangka	Scan Barcode	Waktu Scan (detik)
1	MHMFN527HJK013811	Berhasil	2,05
2	MHMFE75PFJK008425	Berhasil	2,41
3	MHMFN527HJK013808	Berhasil	2,49
4	MHMFE74P5JK182865	Berhasil	2,07
5	MHMFE71PCJK012356	Berhasil	2,45
6	MHMFE74P5JK183725	Berhasil	2,34
7	MHMFE75PFJK009060	Berhasil	2,54
8	MHMFE74P5JK183709	Berhasil	2,43
9	MHMFN62FPJK000164	Berhasil	3,44
10	MHMFE71P1JK063375	Berhasil	2,54
		Waktu rata-rata	2,47

Tabel 2 Hasil Pengujian Aplikasi Malv_RotBarcode

No.	No Rangka	Scan Barcode	Waktu Scan (detik)
1	MHMFN527HJK013811	Berhasil	2,33
2	MHMFE75PFJK008425	Berhasil	2,46
3	MHMFN527HJK013808	Berhasil	2,01
4	MHMFE74P5JK182865	Berhasil	2,33
5	MHMFE71PCJK012356	Berhasil	2,32
6	MHMFE74P5JK183725	Berhasil	2,63
7	MHMFE75PFJK009060	Berhasil	2,55
8	MHMFE74P5JK183709	Berhasil	2,39
9	MHMFN62FPJK000164	Berhasil	2,32
10	MHMFE71P1JK063375	Berhasil	2,26
		Waktu rata-rata	2,36

Hasil dari pencatatan waktu menggunakan *stopwatch* didapatkan waktu ratarata untuk *scan barcode* dari mulai aplikasi membaca *barcode* sampai dengan

memindahkan data nomor rangka ke *text box* aplikasi Melv_InOutBarcode adalah 2,47 detik sedangkan waktu rata-rata untuk *scan barcode* dari mulai aplikasi membaca *barcode* sampai dengan memindahkan data nomor rangka ke *text box* aplikasi Malv_RotBarcode adalah 2,36 detik.

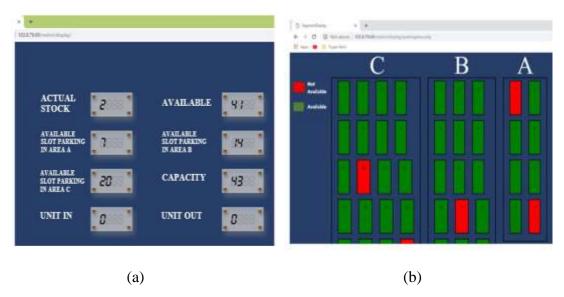
4.2 Pengujian Waktu Transisi Antar Halaman

Hasil pengujian waktu transisi tampilan *website* antar halaman terdapat pada Tabel 3. Dari hasil pengujian tampilan *website* didapatkan waktu rata-rata untuk transisi antar halaman pertama ke halaman kedua adalah 20,16 detik. Terdapat selisih 0,16 detik dari waktu yang telah ditentukan yaitu 20 detik. Hal ini disebabkan karena perhitungan manual menggunakan *stopwatch* dimana perhitungan ini terdapat selisih waktu ketika menekan *stopwatch*.

Tabel 3 Data Waktu Transisi Antar Halaman

Data ke	Waktu Transisi (detik)	
1	20,49	
2	20,13	
3	20,08	
4	20,12	
5	20,10	
6	20,16	
7	20,00	
8	20,12	
9	20,02	
10	20,42	
Waktu rata-rata	20,16	

Tampilan halaman pertama dan kedua pada website terdapat pada Gambar 9. Pada halaman pertama terdapat informasi tentang jumlah tempat parkir pada masingmasing area dan jumlah totalnya. Halaman kedua berisi informasi tentang lokasi parkir kendaraan dan informasi slot tempat parkir yang sudah terisi ditandai dengan warna merah dan slot parkir yang masih kosong ditandai dengan warna hijau.



Gambar 9 Tampilan Aplikasi Website (a) Halaman Pertama dan (b) Halaman Kedua

4.3 Pengujian Respon Waktu Buka dan Tutup Portal Otomatis

Tabel 4 Data Hasil Pengujian Portal

		Respon waktu (detik)	
No.	No Rangka	Membuka	Menutup
		portal	portal
1	MHMFN527HJK013811	1,33	3,33
2	MHMFE75PFJK008425	1,46	3,46
3	MHMFN527HJK013808	1,01	3,01
4	MHMFE74P5JK182865	1,33	3,33
5	MHMFE71PCJK012356	1,32	3,32
6	MHMFE74P5JK183725	1,63	3,63
7	MHMFE75PFJK009060	1,55	3,55
8	MHMFE74P5JK183709	1,39	3,39
9	MHMFN62FPJK000164	1,32	3,32
10	MHMFE71P1JK063375	1,26	3,26
	Waktu rata-rata	1,36	3,36

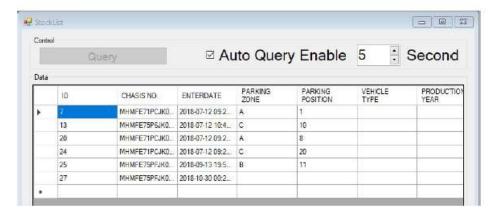
Pada rancang bangun sistem ini memiliki respon waktu 1 detik untuk membuka portal dan 3 detik untuk menutup portal setelah kendaraan melewati sensor *infrared*. Tabel 4 memperlihatkan pada pengujian respon portal didapatkan waktu rata-rata 1,36 detik untuk membuka portal sedangkan untuk menutup portal dibutuhkan waktu 3,36 detik. Dalam proses membuka dan menutup portal terdapat selisih waktu 0,36 detik dari waktu yang telah ditentukan. Hal ini disebabkan karena perhitungan manual

menggunakan *stopwatch* dimana perhitungan ini terdapat selisih waktu ketika menekan *stopwatch*.

4.4 Pengujian dan Analisis Fitur Aplikasi *Dekstop* (Parksys)

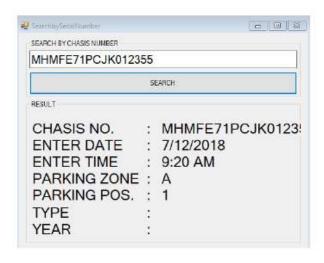
Aplikasi *dekstop* terdiri dari aplikasi *Stock List* yang berfungsi untuk melihat data kendaraan secara *real time*, *history by date* berfungsi untuk mencari data kendaraan berdasarkan tanggal dan bulan, *history by day* berfungsi untuk mencari data kendaraan berdasarkan hari terakhir, *summary* berfungsi untuk mengetahui ringkasan informasi perhitungan stok kendaraan. Hasil pengujian berupa tampilan halaman aplikasi Parksys pada *browser* yang dapat diakses pada *laptop*, PC atau *smartphone*.

Hasil pengujian tombol *query* dan *auto query* di window *stock list* berjalan dengan normal, sehingga mengeluarkan data pada tabel yang ada di *window* seperti pada Gambar 10.



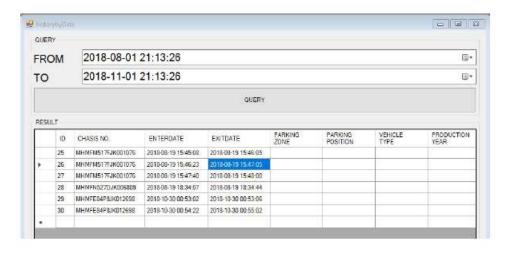
Gambar 10 Hasil Pengujian Tombol Stock List

Hasil pengujian tombol *search by date* berjalan dengan normal dan mengeluarkan data seperti yang terdapat pada Gambar 11.

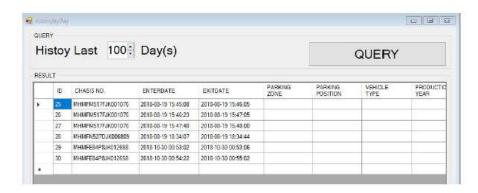


Gambar 11 Hasil Pengujian Tombol Search by Chassis Number

Sedangkan hasil pengujian *history by date* diperlihatkan pada Gambar 12 dan hasil pengujian *history by number of day* diperlihatkan pada Gambar 13.



Gambar 12 Hasil Pengujian Tombol Stock History by Date



Gambar 13 Hasil Pengujian Tombol History by Number of Day

Hasil pengujian dari Gambar 10 sampai dengan 13 menunjukkan hasil tampilan yang benar sesuai data yang dimasukkan.

5. Kesimpulan

Rancang bangun sistem penampil stok kendaran ini terdiri dari empat bagian yaitu aplikasi Android, aplikasi *website*, portal otomatis dan aplikasi *desktop*. Berdasarkan hasil pengujian, untuk *scan barcode* kendaraan aplikasi Android berjalan dengan baik dengan lama waktu *scan* 2,47 detik untuk aplikasi Melv_InOutBarcode dan 2,36 detik untuk aplikasi Malv_RotBarcode. Tiap tombol pada aplikasi *desktop* bekerja sesuai dengan fungsinya masing-masing. Waktu rata-rata transisi antar halaman *website* adalah 20,16 detik. Waktu rata-rata respon portal *gate in* adalah 1,36 detik dan *gate out* 3,36 detik. Pengujian sistem secara keseluruhan dengan sample 10 nomor *chassis* kendaraan berjalan dengan baik. Kelemahan dari sistem yang dibuat adalah tergantung dari ada tidaknya jaringan internet dan kecepatan data pada internet pada saat digunakannya.

Daftar Pustaka

- [1] E. Sutanta, Sistem Basis Data. Yogyakarta: Graha Ilmu, 2004.
- [2] Fakri, "Perancangan Prototype Sistem Parkir Cerdas Berbasis Arduino Uno," Politeknik Negeri Padang, 2016.

- [3] A. P. Sucinata, "Rancang Bangun Kendali Sistem Parkir Otomatis Menggunakan Sensor LDR Berbasis Mikrokontroler AT89S51," Universitas Diponegoro, 2007.
- [4] R. Dinata dan R. Kurniawan, "DESIGN SYSTEM PROTOTYPE SMART PARKING SYSTEM BASED ARDUINO AND MONITORING THROUGH SMARTPHONE," *Ecotipe*, 2017.
- [5] D. Zulkarnain dan E. S. Julian, "PERANCANGAN SISTEM PARKIR DENGAN REKOMENDASI LOKASI PARKIR," *Jetri J. Ilm. Tek. Elektro*, vol. 14, no. 2, hlm. 17–28, Feb. 2017, doi: 10.25105/jetri.v14i2.1611.