

# Perancangan Sistem Informasi *Stock Tire* Berbasis Web Di Departemen Finished Goods Warehouse Plant A

Lutfi Hinda Mukti<sup>1)</sup>

Teknik Elektronika, Politeknik Gajah Tunggal  
lutfihinda5555@gmail.com

Muhammad Harist Al Ikhwan<sup>2)</sup>

Teknologi Industri, Politeknik Gajah Tunggal  
haritsalikhwan17@gmail.com

Muhammad Ridwan Arif Cahyono<sup>3)</sup>

Teknik Elektronika, Politeknik Gajah Tunggal  
ridwan@poltek-gt.ac.id

## ABSTRAK

Teknologi informasi untuk kegiatan pengolahan data dalam berbagai kegiatan perusahaan khususnya di bagian gudang merupakan bagian penting dari setiap perusahaan. Dengan teknologi ini, data yang diolah dapat menjadi informasi yang cepat dan akurat. Namun saat ini banyak perusahaan yang masih menggunakan cara pencatatan manual untuk mencatat barang yang masuk dan keluar gudang. Sementara itu, di era digital yang berkembang pesat yang kemudian didukung dengan munculnya sistem operasi Android, dengan kemampuan teknologi *mobile* membuat masyarakat mulai meninggalkan penggunaan kertas dan beralih ke metode komputer untuk mendukung kegiatan mereka sehari-hari. Oleh karena itu, solusi terbaik untuk mengatasi permasalahan yang ada adalah dengan membuat sistem informasi *stock* berbasis web karena lebih baik dibandingkan dengan sistem pencatatan manual. Perancangan sistem informasi ini menggunakan PHP, HTML, serta MYSQL sebagai *databasenya*. Pada penelitian ini juga dilakukan perancangan aplikasi sistem informasi *stock* dengan menggunakan metode *unified modelling language* (UML). Dari hasil perhitungan dan menggunakan data yang diperoleh, nilai rata – rata waktu perhitungan *stock tire* pada data sebelum perubahan yaitu sebesar 130.0571 detik. Sedangkan pada data sesudah perubahan diperoleh rata – rata waktu perhitungan *stock tire* sebesar 112.3429 detik. Jadi didapat rata – rata penurunan waktu perhitungan *stock tire* yaitu sebesar 17.714 detik setiap *size*.

*Kata kunci:* Sistem informasi, Persediaan barang, *Unified Modelling Language*, Aplikasi web

## V. PENDAHULUAN

### 1. Latar Belakang

Saat ini teknologi sangat berperan penting dalam dunia industri. Seiring berkembangnya teknologi, terdapat banyak sekali penggunaan alat otomasi yang berfungsi untuk mempermudah pekerjaan manusia atau juga sebagai pengingat di saat ada kesalahan yang terjadi. Ketatnya persaingan dalam dunia industri semakin menuntut perusahaan untuk terus melakukan perbaikan secara kontinu terhadap pelayanan maupun kualitas produk yang dihasilkan supaya dapat bersaing dengan perusahaan lainnya dan juga sebagai bentuk upaya memberikan kepuasan terhadap konsumen. Dalam bidang industri manufaktur, gudang sebagai ujung tombak perusahaan juga memiliki peran yang sangat penting dalam memberikan pelayanan dari perusahaan kepada konsumen.

PT. XYZ merupakan perusahaan yang bergerak di bidang industri manufaktur dengan produk yang berupa ban luar, ban dalam, dan *flap* untuk jenis kendaraan sepeda motor, mobil, truk dan bus dengan berbagai macam tipe dan ukuran. Untuk menyimpan barang jadi, PT. XYZ memiliki beberapa gudang, salah satunya yaitu FGWH-A (*Finished Goods Warehouse-A*). FGWH-A bertanggung jawab terhadap proses pelaksanaan serah terima barang hasil produksi ke gudang ban, bertanggung jawab terhadap proses pengiriman barang dari gudang menuju pelanggan, bertanggung jawab untuk menjaga kualitas produk agar tidak mengalami penurunan selama proses penerimaan, penyimpanan, hingga barang dikirimkan ke pelanggan, dan bertanggung jawab untuk melaksanakan *stock opname*.

Berdasarkan hasil observasi yang telah dilakukan penulis, ada beberapa kondisi yang perlu adanya suatu *improvement* seperti belum adanya sistem yang mempermudah dan mempercepat informasi mengenai perhitungan jumlah *stock* sehingga berakibat butuh kertas yang dipakai dan butuh waktu yang lebih lama untuk mengetahui jumlah *stock*, *serial number* dan lokasi yang ada di gudang.

Adapun lamanya waktu perhitungan *stock*, *update serial number* dan lokasi serta gambar kertas yang digunakan adalah sebagai berikut.

Tabel 1. Data Time Study Sebelum Perubahan

No.	Data waktu perhitungan Stock, Serial Number dan Lokasi Aktual/detik
1.	133
2.	126
3.	129
4.	134

Tabel 2. Lanjutan

No.	Data waktu perhitungan Stock, Serial Number dan Lokasi Aktual/detik
5.	136
6.	132
7.	127
8.	126
9.	131
10.	125
11.	134
12.	136
13.	132
14.	128
15.	124
16.	134
17.	123
18.	127
19.	134
20.	140
21.	136
22.	123
23.	124
24.	120
25.	128
26.	134
27.	139
28.	126
29.	127
30.	130
31.	135
32.	136
33.	126
34.	125
35.	132

No	Size	Count	Total
1	766 L9 PRO	100 58 50 139 259 205 76 +80 +6 +100	980
2	815	281 324 8 292 34 52 +200 -10	1142
3	1100 16 S6	50 1 70	41
4	756 L9 PRO	76 128 151 52 100 37 95 +300 +56 -95	880
5	815	292 52 281 339 8 +200 -44	1102
6	1100 16 S6	60 1 -20	41
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			

Gambar 1. Kertas Perhitungan Stock

Gambar 2. Kertas Pengisian Lokasi dan Serial Number

Kondisi tersebut menjadi latar belakang penelitian ini. Perbaikan sistem informasi *stock tire* di *Finished Goods Warehouse A* menjadi fokus penelitian penulis dengan merancang sistem informasi yang berbasis web. Kemudian pada penelitian ini juga dilakukan perancangan sistem aplikasi menggunakan metode *Unified Modeling Language* (UML) untuk mendukung proses kerja yang lebih efisien. Dengan dilakukannya penelitian ini diharapkan dapat mempercepat proses perhitungan jumlah *stock tire* dan penentuan *serial number* dan lokasi ban yang ada di *Finished Goods Warehouse Plant A*.

## 2. Penelitian Sebelumnya

Sebelumnya pernah dilakukan penelitian tentang Sistem Inventori gudang untuk mengontrol persediaan barang pada gudang berbasis *web* [1]. Hasil dari penelitian tersebut adalah, dengan adanya aplikasi berbasis *web* ini dapat memberikan laporan pada pengolahan data barang dari gudang ke kantor pusat secara cepat, akurat dan tepat sasaran.

Ani Oktarini (2017) dalam penelitiannya mencoba memanfaatkan tentang sistem informasi persediaan barang berbasis *web* dengan metode *Fast* [2]. Dengan hasil penelitian tersebut adalah, pengolahan data untuk keluar masuk barang lebih efektif dan efisien, pencarian data dapat lebih efisien karena data sudah terorganisir dengan baik sesuai dengan level akses.

Beberapa penelitian sebelumnya telah dilakukan penelitian tentang sistem informasi permohonan pembayaran resititusi SPPD dengan metode UML *Unified Modeling Language* [3]. Hasil dari penelitian tersebut adalah, Perancangan perangkat lunak dengan menggunakan UML

membuat kebutuhan class dalam perangkat lunak tergambar dengan jelas..

Dwinda (2018), dalam seminar memaparkan penelitian tentang perancangan sistem informasi kepegawaian menggunakan metode UML *Unified Modeling Language* [4]. Hasil dari penelitian tersebut adalah, dengan sistem berbasis komputer diharapkan dapat mempermudah dalam menyimpan dan mengelola data kepegawaian secara akurat dan tepat.

Selain penelitian diatas, sebelumnya juga pernah dilakukan penelitian tentang perancangan aplikasi penjualan hewan ternak berbasis *web* [5]. Hasil dari penelitian tersebut adalah, aplikasi tersebut dapat mempermudah konsumen dalam melakukan pembelian dan pemilihan hewan qurban serta dapat meningkatkan penjualan hewan ternak.

## 3. Tujuan Kajian

Adapun tujuan dari penelitian ini yaitu sebagai berikut:

1. Merancang sistem informasi perhitungan *stock tire* dengan metode *Unified Modeling Language* (UML).
2. Merancang sistem informasi perhitungan *stock tire* di *Finished Goods Warehouse A* berbasis *web*.
3. Membandingkan sistem informasi perhitungan *stock tire* yang berjalan di *Finished Goods Warehouse A* sebelum dan setelah perubahan.

## 4. Manfaat Kajian

Adapun manfaat yang diperoleh setelah melakukan penelitian ini. Berikut manfaat penelitian yang dapat dilihat pada poin-poin di bawah ini:

1. Memudahkan penyampaian informasi *stock tire*, *serial number* dan lokasi *tire* antar bagian.
2. Mempercepat informasi mengenai jumlah *stock tire*, *serial number tire* dan lokasi *tire*.
3. Mempercepat proses *input* mengenai jumlah *stock tire*, *serial number tire* dan lokasi *tire* di Gudang.

## 5. Rumusan Masalah

Berdasarkan penjelasan pada latar belakang, maka dapat dirumuskan perumusan masalah yaitu sebagai berikut:

1. Proses perhitungan mengenai jumlah *stock tire* dan penentuan *serial number* dan lokasi di gudang masih dilakukan secara manual dengan menggunakan kertas.
2. Lambatnya sistem informasi yang didapatkan mengenai *stock tire* yang ada di gudang.
3. Masih melakukan pengetikan secara manual mengenai jumlah *stock tire* yang ada di gudang.

## 6. Batasan Masalah

Agar pembahasan pada penelitian ini lebih fokus dan terarah berikut batasan masalah dalam penelitian ini:

1. Penelitian hanya dilakukan di APW 1 Finished Goods Warehouse Plant A.
2. *Output* yang dihasilkan hanya berupa usulan perbaikan dalam bentuk perancangan.
3. Tidak membahas terkait biaya pembuatan alat bantu.

### 7. Pertanyaan Kajian

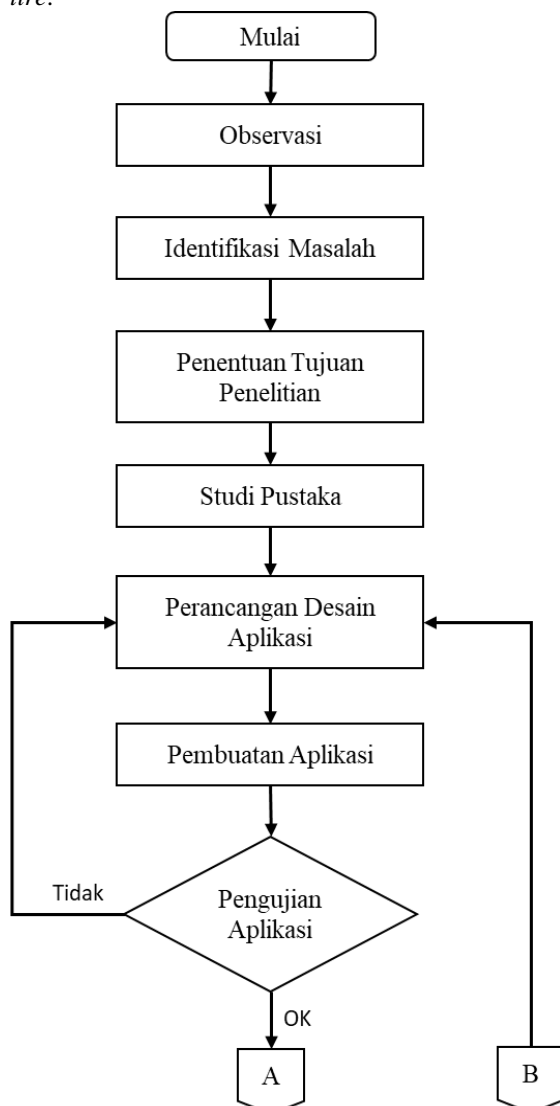
Berdasarkan latar belakang terkait permasalahan yang akan dikaji, maka pertanyaan terkait penelitian tersebut, yaitu:

1. Bagaimana cara merancang aplikasi sistem informasi *stock tire* berbasis web?
2. Bagaimana cara merancang tampilan aplikasi untuk dapat digunakan oleh pengguna secara baik?
3. Bagaimana proses perhitungan *stock tire* saat ini dan yang diterapkan dengan menggunakan aplikasi?

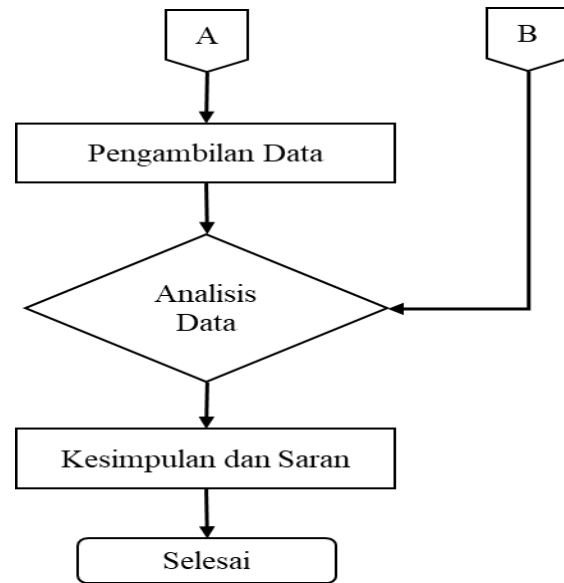
## VI. METODE PENELITIAN

### 1. Alur Penelitian

Berikut adalah alur penelitian yang dilakukan untuk merancang aplikasi sistem informasi *stock tire*.



Gambar 3. Alur Kajian



Gambar 3. Lanjutan

### 2. Observasi

Pada tahap ini, dilakukan pengamatan terhadap proses perhitungan *stock control* di bagian Finished Goods Warehouse Plant A PT. XYZ.

### 3. Identifikasi Masalah

Pada tahap ini penulis melakukan identifikasi masalah yang terjadi pada bagian *stock control* dalam perhitungan jumlah *stock tire* dan penentuan *serial number* dan lokasi *tire*. Masalah yang terjadi pada *stock control* adalah masih lamanya proses perhitungan *stock tire* dan penentuan *serial number* dan lokasi *tire*, disebabkan sistem perhitungan masih dilakukan secara manual dan menggunakan kertas sebagai *form* perhitungannya. Pada tahap identifikasi masalah dilakukan pengambilan data *time study* proses perhitungan pada kondisi aktual.

### 4. Penentuan Tujuan Penelitian

Pada penelitian ini penulis memiliki tujuan untuk membuat perancangan sistem informasi *stock tire* yang dapat mempercepat proses perhitungan *stock* serta memonitor ketersediaan jumlah *tire*, *serial number tire* dan lokasi *tire*.

### 5. Study Pustaka

Pada tahap ini dilakukan studi yang didasarkan oleh penelitian yang telah dilakukan sebelumnya.

### 6. Perancangan Desain Aplikasi

Penulis membuat desain sistem aplikasi untuk memudahkan prosedur perhitungan jumlah *stock tire*, *serial number* dan penentuan lokasi *tire*. Pembuatan sistem ini dilakukan untuk memberikan gambaran rancangan aplikasi yang sesuai dengan prosedur dalam perhitunga. Perancangan desain aplikasi perhitungan *stock tire*, *serial number* dan penentuan lokasi ini dimulai dengan membuat UML (*Unified Modeling Language*).

### 7. Pembuatan Aplikasi

Pembuatan aplikasi sesuai dengan kebutuhan *User* didasarkan oleh UML yang telah disusun pada tahap sebelumnya.

## 8. Pengujian Aplikasi

Desain yang telah dibuat kemudian dilakukan uji validasi dengan melakukan pengujian efisiensi dan kemudahan pendataan alat dengan metode *trial & error* pada sistem yang terdapat pada perhitungan *stock tire* di bagian *stock control* dengan sistem yang didesain oleh peneliti.

## 9. Pengambilan Data

Pada tahap ini dilakukan pengambilan data proses kerja setelah aplikasi diterapkan. Data proses kerja setelah perubahan ini kemudian akan dibahas pada tahap selanjutnya

## 10. Analisis Data

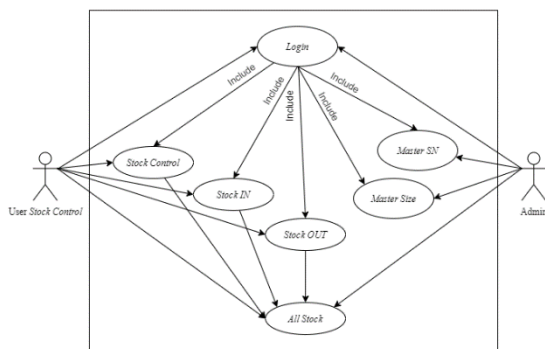
Dari data-data yang telah dikumpulkan, kemudian penulis mengolah data yang ada menggunakan tingkat kebutuhan yang disesuaikan dengan tingkat waktu perhitungan jumlah *stock tire*, *serial number* dan lokasi dengan memperhatikan prosedur dan teknik pengolahan datanya. Pengolahan data ini dilakukan dengan merekap lamanya waktu perhitungan *stock tire* sebelum dan sesudah perbaikan. Data ini yang kemudian akan dijadikan untuk menentukan metode perhitungan yang lebih efisien dan efektif, dengan melihat banyaknya jumlah *stock* yang dihitung dengan prosedur perhitungan yang dilakukan oleh bagian *stock control*. Beberapa pengujian yang dilakukan untuk analisis data adalah uji kecukupan data, uji keseragaman data, dan uji normalitas data. Selanjutnya dengan uji T 2 sampel proses perhitungan setelah perubahan dianalisis untuk melihat perbandingan sebelum dan sesudah perubahan.

## VII. HASIL DAN PEMBAHASAN

### 1. Unified Modelling Language

#### 1.1 Use Case Diagram

Pada sistem yang dibangun, ada 2 aktor yaitu *admin*, *User stock control*. *Admin* sebagai pengelola sistem dan *User* sebagai pengguna sistem.



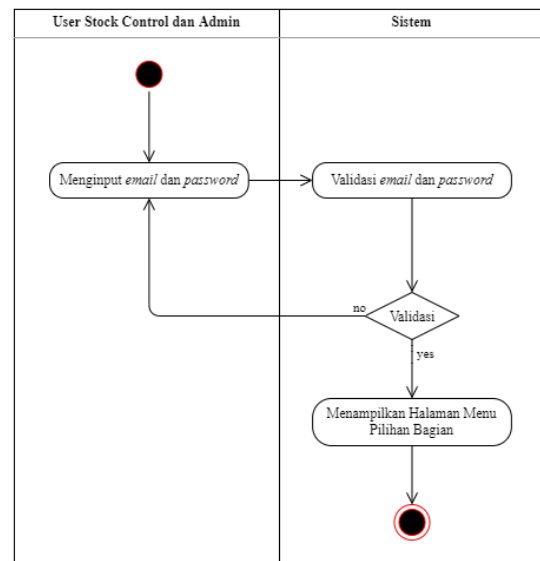
Gambar 4. Use Case Diagram

#### 1.2 Activity Diagram

*Activity diagram* adalah salah satu jenis *diagram* pada UML yang dapat memodelkan proses apa saja yang terjadi pada sistem. Berikut ini merupakan *activity diagram* dari sistem informasi

*stock tire* berdasarkan *profile User*.

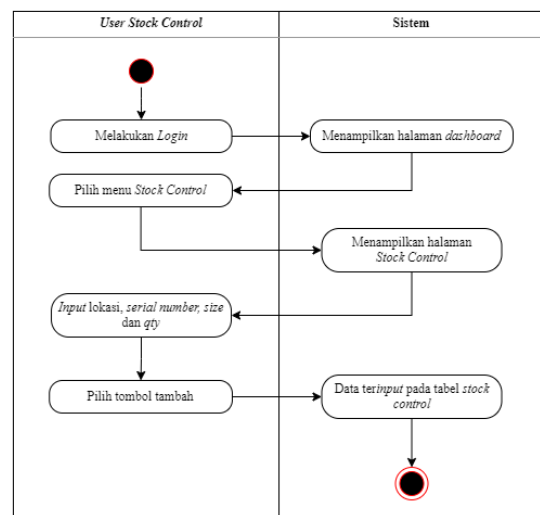
#### 1.2.1 Actuvuty Diagram Login



Gambar 5. Activity Diagram Login

Pada tahap awal kegiatan, *user* dan *admin* melakukan *login* dengan memasukkan *email* dan *password*. Ketika sudah melakukan *input*, sistem akan melakukan validasi *email* dan *password* apakah sudah sesuai dengan data yang terdaftar atau belum. Jika salah akan kembali ke halaman awal dan jika benar sistem akan langsung menampilkan halaman *dashboard*.

#### 1.2.2 Actuvuty Diagram Stock Control

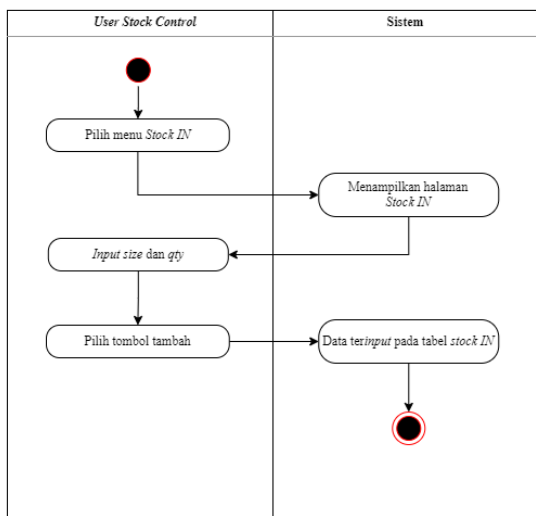


Gambar 6. Activity Diagram Stock Control

Setelah *user stock control* berhasil melakukan *login*, sistem akan menampilkan halaman menu *dashboard*. Kemudian *user* memilih menu *stock control* dan sistem akan langsung menampilkan halaman menu *stock control*. Ketika sudah di halaman *stock control*, selanjutnya *user* melakukan *input* data lokasi *tire*, *serial number*, *size* dan *quantity* sesuai dengan *tire* yang sudah dihitung. Jika sudah, *user* memilih tombol tambah dan secara

langsung sistem akan menginput data tersebut pada tabel *stock control*.

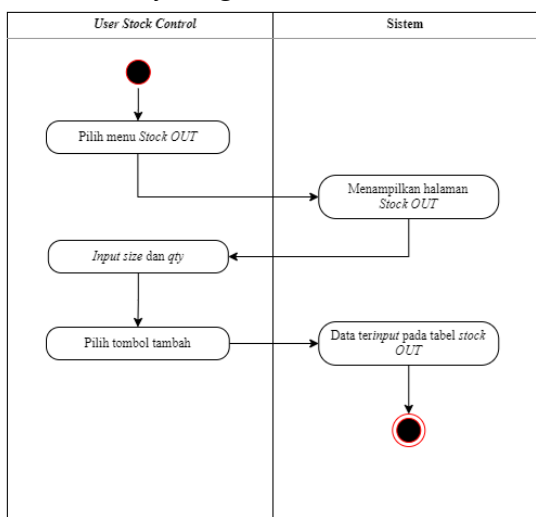
### 1.2.3 Actuvuty Diagram Stock In



Gambar 7. Activity Diagram Stock IN

Setelah *user stock control* berhasil melakukan *login*, sistem akan menampilkan halaman menu *dashboard*. Kemudian *user* memilih menu *stock IN* dan sistem akan langsung menampilkan halaman menu *stock IN*. Ketika sudah di halaman *stock IN*, selanjutnya *user* melakukan *input* data *size* dan *quantity* *tire* sesuai dengan *tire* yang ada pada rekapan *recorder* bagian penerimaan. Jika sudah, *user* memilih tombol tambah dan secara langsung sistem akan menginput data tersebut pada tabel *stock IN*.

### 1.2.4 Actuvuty Diagram Stock out

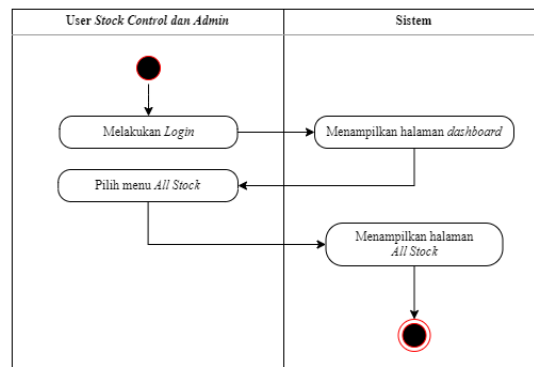


Gambar 8. Activity Diagram Menu Data OUT

Setelah *user stock control* berhasil melakukan *login*, sistem akan menampilkan halaman menu *dashboard*. Kemudian *user* memilih menu *stock OUT* dan sistem akan langsung menampilkan halaman menu *stock OUT*. Ketika sudah di halaman *stock OUT*, selanjutnya *user* melakukan

*input* data *size* dan *quantity* *tire* sesuai dengan *tire* yang ada pada *slip order* bagian *loading*. Jika sudah, *user* memilih tombol tambah dan secara langsung sistem akan menginput data tersebut pada tabel *stock OUT*.

### 1.2.5 Actuvuty Diagram Stock All

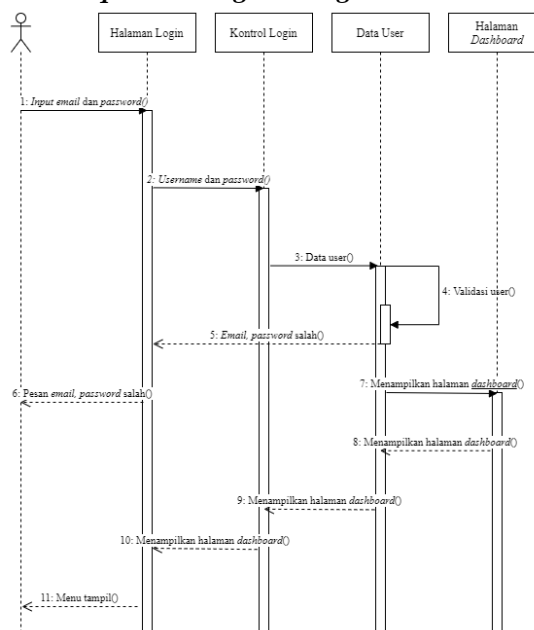


Gambar 9. Activity Diagram All Stock

Setelah *user stock control* dan *admin* berhasil melakukan *login*, sistem akan menampilkan halaman menu *dashboard*. Kemudian *user* dan *admin* memilih menu *All stock* dan sistem akan langsung menampilkan halaman menu *All stock*. Di halaman *All stock*, *user* dan *admin* dapat melihat data yang sudah di *input* oleh *user* pada hari tersebut.

## 1.3 Sequence Diagram

### 1.3.1 Sequence Diagram Login

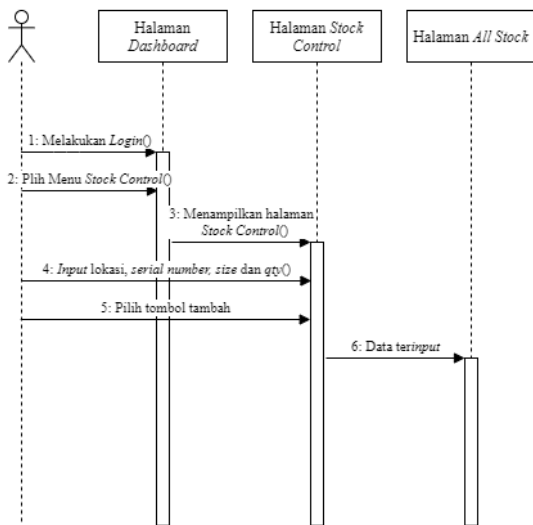


Gambar 10. Sequence Diagram Login

Pada tahap awal kegiatan, *user* dan *admin* melakukan *login* dengan memasukkan *email* dan *password*. Ketika sudah melakukan *input*, sistem akan melakukan validasi *email* dan *password* apakah sudah sesuai dengan data yang terdaftar atau belum. Jika salah akan kembali ke halaman awal dengan melakukan pemberian informasi jika *email* dan *password* salah. Jika benar sistem akan

langsung menampilkan halaman *dashboard*.

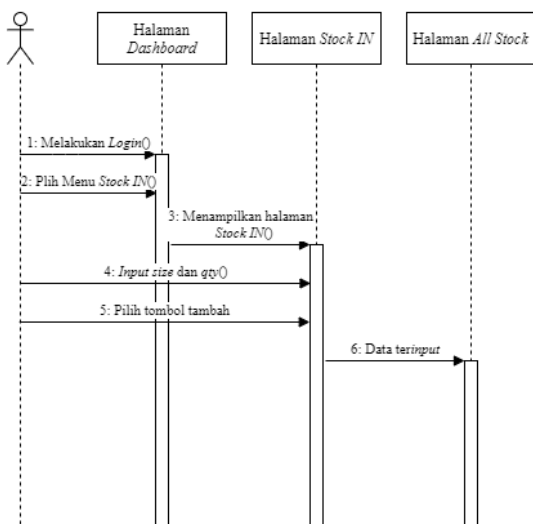
### 1.3.2 Sequence Diagram Stock Control



Gambar 11. Sequence Diagram Stock Control

Setelah *user stock control* berhasil melakukan *login*, sistem akan menampilkan halaman menu *dashboard*. Kemudian *user* memilih menu *stock control* dan sistem akan langsung menampilkan halaman menu *stock control*. Ketika sudah di halaman *stock control*, selanjutnya *user* melakukan *input* data lokasi *tire*, *serial number*, *size* dan *quantity* sesuai dengan *tire* yang sudah dihitung. Jika sudah, *user* memilih tombol tambah dan secara langsung sistem akan menginput data tersebut pada tabel *stock control*.

### 1.3.3 Sequence Diagram Stock In

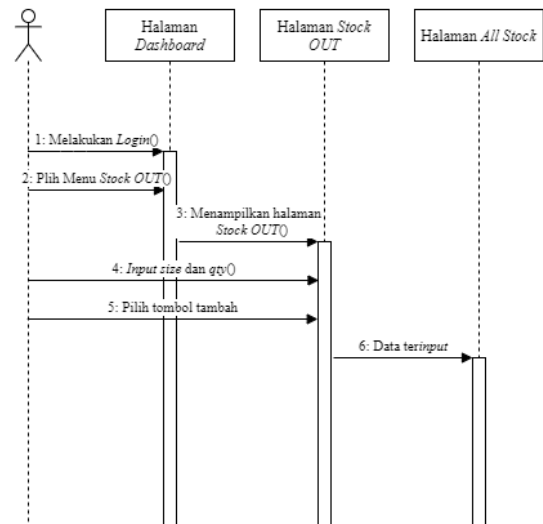


Gambar 12. Sequence Diagram Stock IN

Setelah *user stock control* berhasil melakukan *login*, sistem akan menampilkan halaman menu *dashboard*. Kemudian *user* memilih menu *stock IN* dan sistem akan langsung menampilkan halaman menu *stock IN*. Ketika sudah di halaman *stock IN*, selanjutnya *user* melakukan *input* data *size* dan

*quantity tire* sesuai dengan *tire* yang ada pada rekapan *recorder* bagian penerimaan. Jika sudah, *user* memilih tombol tambah dan secara langsung sistem akan menginput data tersebut pada tabel *stock IN*.

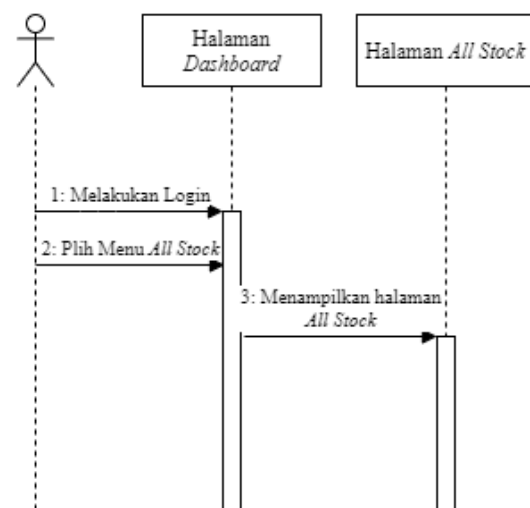
### 1.3.4 Sequence Diagram Stock Out



Gambar 13. Sequence Diagram Stock OUT

Setelah *user stock control* berhasil melakukan *login*, sistem akan menampilkan halaman menu *dashboard*. Kemudian *user* memilih menu *stock OUT* dan sistem akan langsung menampilkan halaman menu *stock OUT*. Ketika sudah di halaman *stock OUT*, selanjutnya *user* melakukan *input* data *size* dan *quantity tire* sesuai dengan *tire* yang ada pada *slip order* bagian *loading*. Jika sudah, *user* memilih tombol tambah dan secara langsung sistem akan menginput data tersebut pada tabel *stock OUT*.

### 1.3.5 Sequence Diagram Stock All



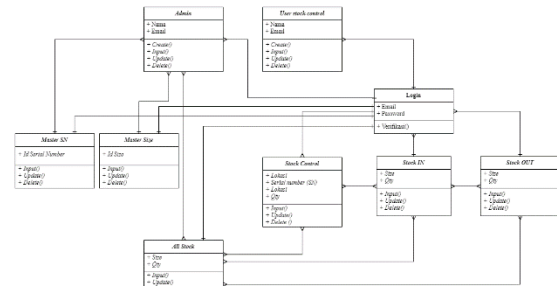
Gambar 14. Sequence Diagram All Stock

Setelah *user stock control* dan *admin* berhasil melakukan *login*, sistem akan menampilkan halaman menu *dashboard*. Kemudian *user* dan *admin* memilih menu *All stock* dan sistem akan langsung menampilkan halaman menu *All stock*. Di

halaman *All stock*, *user* dan *admin* dapat melihat data yang sudah di *input* oleh *user stock control* pada hari tersebut.

### 1.4 Class Diagram

*Class diagram* digunakan untuk menampilkan *class* yang ada pada sistem. *Diagram* ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi yang terdapat pada sistem.

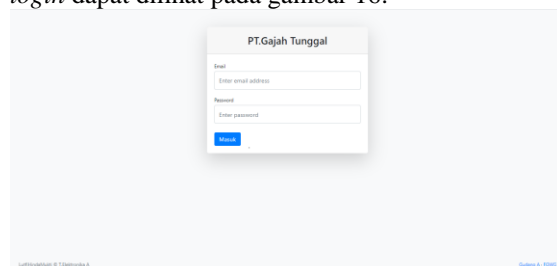


Gambar 15. Class Diagram

## 2. Tampilan Aplikasi Web

### 2.1 Tampilan Login

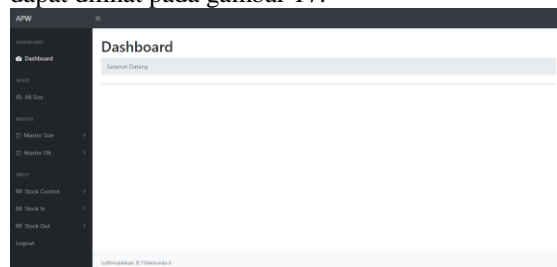
Berikut ini adalah tampilan halaman *login*. Di halaman *login* ini *admin* dan *User* harus memasukkan *email* dan *password* yang sesuai agar dapat masuk ke halaman selanjutnya. Halaman *login* dapat dilihat pada gambar 16.



Gambar 16. Tampilan Login

### 2.2 Tampilan Dashboard Admin

Berikut ini adalah tampilan halaman *dashboard* jika masuk sebagai *admin*. Pada halaman *dashboard* terdapat beberapa pilihan menu yang dapat dipilih oleh *admin*. Halaman *dashboard* dapat dilihat pada gambar 17.

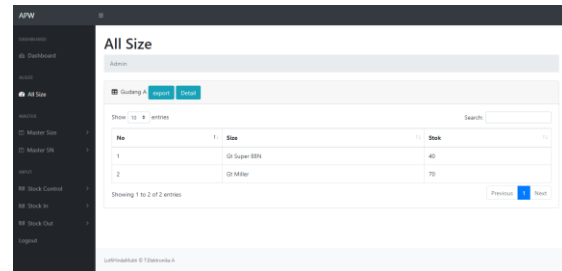


Gambar 17. Tampilan Dashboard Admin

### 2.3 Tampilan Halaman All Size

Berikut ini adalah tampilan halaman *all Size*. Di halaman *all size* dapat melihat seluruh *stock size* yang telah di-*input* oleh dan mengetahui jumlah. Halaman semua *stock* dapat dilihat pada gambar

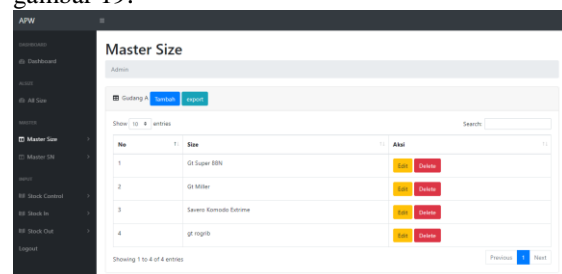
18.



Gambar 18. Tampilan Halaman All Size

### 2.4 Tampilan Halaman Master Size

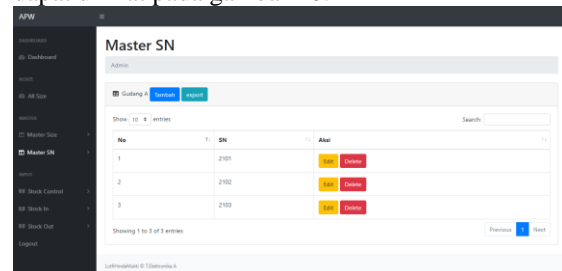
Berikut ini adalah tampilan halaman *master size admin*. Di halaman *master size admin* dapat melakukan *input* atau menambahkan *master size* baru dan juga dapat menghapus *master size* lama. Halaman *master size admin* dapat dilihat pada gambar 19.



Gambar 19. Tampilan Halaman Master Size Admin

### 2.5 Tampilan Halaman Master SN

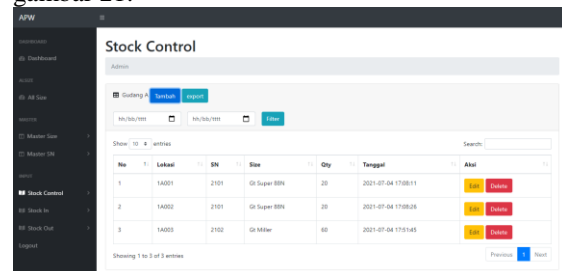
Berikut ini adalah tampilan halaman *master sn*. Pada halaman *master sn admin* dapat melakukan penambahan *sn* baru dan dapat menghapus dan meng-*edit sn* yang sudah ada. Halaman *master sn* dapat dilihat pada gambar 20.



Gambar 20. Tampilan Halaman Master SN Admin

### 2.6 Tampilan Halaman Stock Control

ini adalah tampilan halaman *stock control*. Pada bagian ini adalah tabel yang menampilkan hasil *input stock* hari ini yang sebelumnya sudah di-*input*. Halaman *stock control* dapat dilihat pada gambar 21.

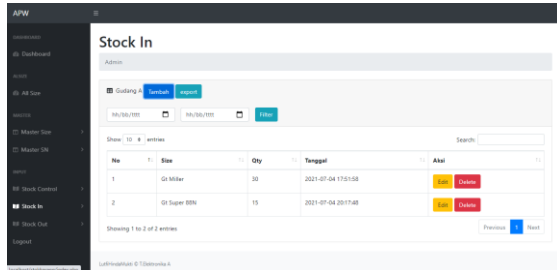


Gambar 21. Tampilan Halaman Stock Control Admin



## 2.7 Tampilan Halaman Stock IN

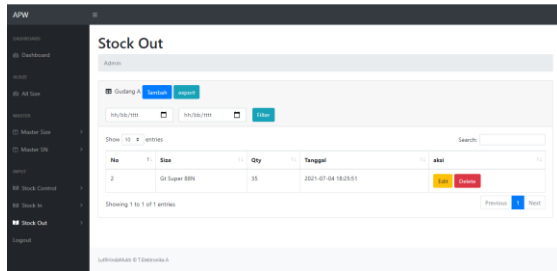
Berikut ini adalah tampilan halaman *stock in*. Pada bagian ini adalah tabel yang menampilkan hasil *input in* hari ini yang sebelumnya sudah di-*input* Halaman *stock in* dapat dilihat pada gambar 22.



Gambar 21. Tampilan Halaman Stock IN Admin

## 2.8 Tampilan Halaman Stock Out

Berikut ini adalah tampilan halaman *stock out*. Pada bagian ini adalah tabel yang menampilkan hasil *input out* hari ini yang sebelumnya sudah di-*input* Halaman *stock out* dapat dilihat pada gambar 23.



Gambar 22. Tampilan Halaman Stock OUT Admin

## 3. Uji Verifikasi dan Validasi

### 3.1 Uji Verifikasi

Pada uji verifikasi dilakukan pengujian terhadap fitur-fitur yang terdapat pada aplikasi tersebut. Pengujian ini bertujuan untuk mengetahui apakah fitur-fitur tersebut berfungsi sesuai dengan model yang telah dibuat sebelumnya. Hasilnya Semua fungsi berhasil bekerja dengan baik.

Tabel 2. Pengujian Verifikasi

Prosedur	Hasil	Keterangan
Membuka Halaman <i>login</i>	Halaman <i>login</i> tampil	Berhasil
<i>Login Email &amp; Password</i>	Masuk ke halaman <i>Dashboard</i>	Berhasil
Lihat All Size	Masuk ke halaman All Size	Berhasil
Lihat detail	Masuk ke halaman detail	Berhasil
Lihat <i>master Sn</i>	Masuk ke halaman <i>master Sn</i>	Berhasil
Tambah data <i>master Sn</i>	Masuk ke halaman tambah data Sn	Berhasil

Tabel 23. Lanjutan

Prosedur	Hasil	Keterangan
Membuat <i>master Sn</i>	<i>Master Sn</i> berhasil	berhasil
Lihat <i>master Size</i>	Masuk ke halaman <i>master Size</i>	Berhasil
Tambah data <i>master Size</i>	Masuk ke halaman tambah <i>master Size</i>	Berhasil
Membuat <i>master Size</i>	<i>Master Size</i> berhasil bertambah	Berhasil
Lihat <i>Stock Control</i>	Tampil Pilihan <i>Stock Control</i>	Berhasil
Pilih <i>stock control</i>	Masuk ke halaman <i>stock control</i>	Berhasil
Tombol tambah <i>stock control</i>	Menampilkan menu <i>input stock control</i>	Berhasil
Menambahkan <i>stock control</i>	<i>Stock</i> bertambah dan data tampil di tabel	Berhasil
Pilih <i>stock in</i>	Masuk ke halaman data <i>stock in</i>	Berhasil
Tombol tambah <i>stock in</i>	Menampilkan menu <i>stock in</i>	Berhasil
Menambahkan <i>stock in</i>	<i>Stock</i> bertambah dan data tampil di tabel	Berhasil
Pilih data <i>stock out</i>	Masuk ke halaman data <i>stock out</i>	Berhasil
Tombol tambah <i>stock out</i>	Menampilkan menu <i>stock out</i>	Berhasil
Menambahkan <i>stock out</i>	<i>Stock</i> bertambah dan data tampil di tabel	Berhasil
Tombol <i>Edit &amp; Delete</i>	Sesuai dengan fungsi	Berhasil

### 3.2 Uji Validasi

Pada uji validasi dilakukan pengujian terhadap koneksi dengan *database* pada aplikasi tersebut. Pengujian ini bertujuan untuk mengetahui apakah data yang tersimpan sesuai dengan data yang di-*input* oleh *User* dan data yang ditampilkan sesuai dengan data yang tersimpan pada *database*. Hasilnya adalah semua berjalan dengan semestinya.

Tabel 34. Pengujian Validasi

Prosedur	Hasil	Keterangan
Menyimpan data di Database	Data tersimpan sesuai dengan data yang dimasukkan	Berhasil
Melihat data dari Database	Data yang tampil sesuai dengan data yang tersimpan	Berhasil

#### 4. Analisis Data Waktu Sebelum Perubahan

##### 4.1 Hasil Pengambilan Data Sebelum Perubahan

Hasil pengambilan data *time study* diperoleh dari pengambilan data secara langsung secara acak dengan hasil dapat dilihat pada tabel 4.

Tabel 4. Data Time Study Sebelum Perubahan

No.	Data waktu perhitungan <i>Stock, Serial Number</i> dan Lokasi Aktual/detik
1.	133
2.	126
3.	129
4.	134
5.	136
6.	132
7.	127
8.	126
9.	131
10.	125
11.	134
12.	136
13.	132
14.	128
15.	124
16.	134
17.	123
18.	127
19.	134
20.	140
21.	136
22.	123
23.	124
24.	120
25.	128
26.	134
27.	139
28.	126
29.	127
30.	130
31.	135
32.	136

Tabel 4. Lanjutan

No.	Data waktu perhitungan <i>Stock, Serial Number</i> dan Lokasi Aktual/detik
33.	126
34.	125
35.	132

##### 4.2 Uji Normalitas Data

Uji normalitas data dilakukan menggunakan aplikasi IBM SPSS *Statistic 25* dengan menggunakan metode Kolmogorov – Smirnov. Dengan syarat pengujian:

- A. Hipotesis:
  - $H_0$ : Data berdistribusi normal
  - $H_1$ : Data tidak berdistribusi normal
- B. Statistik uji: Uji Kolmogorof-Smirnov
- C.  $\alpha = 0,005$
- D. Daerah kritis:  $H_0$  ditolak jika  $Sig. < \alpha$

Hasil pengujian normalitas data tersebut adalah sebagai berikut:

One-Sample Kolmogorov-Smirnov Test

		Data_Before
N		35
Normal Parameters <sup>a,b</sup>	Mean	130.0571
	Std. Deviation	5.06396
	Most Extreme Differences	
	Absolute	.127
	Positive	.127
	Negative	-.125
Test Statistic		.127
Asymp. Sig. (2-tailed)		.167 <sup>c</sup>

- a. Test distribution is Normal.
- b. Calculated from data.
- c. Lilliefors Significance Correction.

Gambar 24. Hasil Uji Normalitas Data Sebelum Perubahan

Data dapat dikatakan berdistribusi normal jika  $Sig. > 0,05$ . Berdasarkan gambar di atas, dapat dilihat bahwa data waktu perhitungan *stock tire* sebelum modifikasi merupakan data berdistribusi normal dengan nilai  $0,167 > 0,05$ .

#### 5. Analisis Data Waktu Setelah Perubahan

##### 5.1 Hasil Pengambilan Data Sudah Perubahan

Hasil pengambilan data *time study* setelah perubahan diperoleh dari pengambilan data secara langsung secara acak menggunakan aplikasi yang telah dirancang dengan hasil dapat dilihat pada tabel 5.

Tabel 5. Data Time Study Setelah Perubahan

No.	Data waktu perhitungan <i>Stock, Serial Number</i> dan Lokasi Aktual/detik
1.	116
2.	106
3.	111
4.	114
5.	121
6.	114
7.	110
8.	107
9.	112
10.	105
11.	117
12.	112
13.	119
14.	111
15.	119
16.	120
17.	104
18.	108
19.	114
20.	122
21.	106
22.	112
23.	110
24.	102
25.	107
26.	112
27.	118
28.	109
29.	110
30.	117
31.	115
32.	119
33.	105
34.	112
35.	116

### 5.1 Uji Normalitas Data

Uji normalitas data dilakukan menggunakan aplikasi IBM SPSS *Statistic 25* dengan menggunakan metode Kolmogorov – Smirnov. Dengan syarat pengujian:

E. Hipotesis:

$H_0$ : Data berdistribusi normal

$H_1$ : Data tidak berdistribusi normal

F. Statistik uji: Uji Kolmogorof-Smirnov

G.  $\alpha = 0,005$

H. Daerah kritis:  $H_0$  ditolak jika  $Sig. < \alpha$

Hasil pengujian normalitas data tersebut adalah sebagai berikut:

One-Sample Kolmogorov-Smirnov Test		
		Data_After
N		35
Normal Parameters <sup>a,b</sup>	Mean	112.3429
	Std. Deviation	5.28562
Most Extreme Differences	Absolute	.097
	Positive	.097
	Negative	-.070
Test Statistic		.097
Asymp. Sig. (2-tailed)		.200 <sup>c,d</sup>

a. Test distribution is Normal.  
b. Calculated from data.  
c. Lilliefors Significance Correction.  
d. This is a lower bound of the true significance.

Gambar 25. Hasil Uji Normalitas Data Setelah Perubahan

Data dapat dikatakan berdistribusi normal jika  $Sig. > 0,05$ . Berdasarkan gambar di atas, dapat dilihat bahwa data waktu perhitungan *stock tire* sebelum modifikasi merupakan data berdistribusi normal dengan nilai  $0,200 > 0,05$ .

### 6. Uji T

Uji *Paired Sample T Test* atau disebut juga uji T berpasangan dilakukan dalam penelitian ini untuk melihat apakah ada perbedaan atau tidak dari data sebelum dan sesudah modifikasi mesin. Dalam uji T berpasangan yang dilakukan, digunakan *Confidence Level 95%*, taraf nyata 5%.

Dasar pengambilan keputusan pada penelitian kali ini adalah sebagai berikut:

1. Jika nilai  $t > 2.03224$  (diambil dari tabel t di lampiran), dapat diartikan bahwa terdapat perbedaan yang signifikan antara persentase data sebelum modifikasi dengan persentase data sesudah modifikasi.
2. Jika nilai  $t < 2.03224$  (diambil dari tabel t di lampiran), dapat diartikan bahwa tidak terdapat perbedaan yang signifikan antara persentase data sebelum modifikasi dengan persentase data sesudah modifikasi.

Berikut merupakan hasil dari pengujian uji *paired sample T test* menggunakan aplikasi SPSS.

Paired Samples Statistics					
		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	Sebelum Perbaikan	130.0571	35	5.06396	.85697
	Setelah Perbaikan	112.3429	35	5.28562	.89343

Gambar 26. Hasil Uji T Test

Dari gambar, dapat diperoleh hasil yaitu nilai rata – rata waktu perhitungan *stock tire* pada data sebelum perubahan yaitu sebesar 130.0571 detik dari 35 kali pengambilan sampel data, dengan sebaran

data yang diperoleh adalah 5.06396. Sedangkan pada data sesudah perubahan diperoleh rata – rata waktu perhitungan *stock tire* sebesar 112.3429 detik dari 35 kali pengambilan sampel data, dengan sebaran data yang diperoleh adalah 5.28562.

		Paired Differences				t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference			
					Lower	Upper		
Pair 1	Sebelum Perbaikan - Setelah Perbaikan	17.714	4.16266	.70362	16.29	19.14	25.176	.0000000000000000000000000144

Gambar 27. Lanjutan

Dari gambar, diperoleh bahwa nilai t sebesar 25.176. Hal tersebut berarti nilai t lebih dari 2.03224 (diambil dari tabel pada lampiran), sehingga dapat disimpulkan bahwa adanya aplikasi dapat mempengaruhi hasil waktu perhitungan *stock tire*. Dari gambar, rata – rata penurunan waktu perhitungan *stock tire* dari data yang diambil sebanyak 35 sampel yaitu sebesar 17.714 detik setiap *size*.

## VIII. KESIMPULAN

Berdasarkan hasil perancangan dan pembuatan aplikasi untuk sistem informasi *stock tire* pada Finished Goods Warehouse Plant A yang telah dilakukan, dapat disimpulkan hal-hal sebagai berikut:

1. Perancangan sistem informasi *stock tire* dengan menggunakan metode *Unified Modeling Language (UML)* membuat kebutuhan dalam aplikasi tergambar dengan jelas dan sistem informasi *stock tire* sesuai dengan kebutuhan pengguna.
2. Aplikasi sistem informasi *stock tire* berbasis web di Finished Goods Warehouse Plant A telah berhasil dirancang dengan fitur dapat melakukan perhitungan jumlah *stock tire*, penentuan *serial number tire* dan lokasi *tire* sesuai dengan jumlah, umur dan lokasi penyimpanan.
3. Nilai rata – rata waktu perhitungan *stock tire* pada data sebelum perubahan yaitu sebesar 130.0571 detik. Sedangkan pada data sesudah perubahan diperoleh rata – rata waktu perhitungan *stock tire* sebesar 112.3429 detik. Maka didapat rata – rata penurunan waktu perhitungan *stock tire* yaitu sebesar 17.714 detik setiap *size*.

## DAFTAR PUSTAKA

- [1] Agusvianto, H. (2017). *Sistem Informasi Inventori Gudang Untuk Mengontrol Persediaan Barang Pada Gudang Studi Kasus : PT. Alaisys Sidoarjo*. 01, 40–46.
- [2] Budiman, D. A., Nugraha, D. M., & Margahayu, S. A. (2019). *Aplikasi Rapor Online Berbasis Web Menggunakan*

*Framework Codeigniter ( Studi Kasus Di Smk Angkasa 1 Margahayu )*. *Jurnal Computech & Bisnis*, 13(2), 112–121.

- [3] Elu, A. M. (2013). Rancang Bangun Aplikasi Pendeteksi Vulnerability Structured Query Language (SQL) Injection Untuk Keamanan Website. *Jurnal Teknologi Informasi*, VII(1), 111–124.
- [4] Hartati, S. (2020). Perancangan Sistem Informasi Inventaris Barang Pada Kantor Notaris Dan Ppat Ra Lia Kholila, Sh Menggunakan Visual Studio Code. *Jurnal Siskomti*, 3(2), 37–48.
- [5] Iqbal Kurniansyah, M., & Sinurat, S. (2020). Sistem Pendukung Keputusan Pemilihan Server Hosting dan Domain Terbaik untuk WEB Server Menerapkan Metode VIKOR. *Jurnal Sistem Komputer Dan Informatika (JSON)*, 2(1), 14–24. <https://doi.org/10.30865/json.v2i1.2450>
- [6] Kawano, K., Umemura, Y., & Kano, Y. (1983). Field Assessment and Inheritance of Cassava Resistance to Superelongation Disease 1 . *Crop Science*, 23(2), 201–205. <https://doi.org/10.2135/cropsci1983.0011183x002300020002x>
- [7] Nuryadi, Tutut Dewi Astuti, Endang Sri Utami, & Martinus Budiantara. (2017). *Dasar-Dasar Statistika Penelitian*.