

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Kerja magang dilakukan pada Divisi IT sebagai IT Developer sesuai dengan kebutuhan perusahaan. Selama kerja magang di kantor, koordinasi dilakukan dengan Bapak Kok Ling sebagai Senior Manager. Begitu juga dengan tugas yang dikerjakan, diberikan langsung oleh Bapak Kok Ling.

Koordinasi kerja magang dilakukan dengan media WhatsApp selama kerja di rumah. Setiap minggunya, Bapak Kok Ling meminta untuk melaporkan perkembangan melalui media Whatsapp dengan mengirimkan video singkat terkait sistem yang telah dikerjakan. Apabila menemukan kesulitan dan membutuhkan informasi terkait sistem, dapat langsung didiskusikan dengan Bapak Kok Ling.

#### 3.2 Tugas yang Dilakukan

Kerja magang dilaksanakan selama 3 (tiga) bulan atau 11 minggu dengan *timeline* kerja yang dapat dilihat pada Tabel 3.1.

Tabel 3.1 Jadwal Kerja Magang

Nama Kegiatan	Minggu Ke-										
	1	2	3	4	5	6	7	8	9	10	11
Pengenalan Lingkungan Kerja	■										
Studi Literatur		■									
Analisa dan perancangan website			■	■							
Proses Pembuatan Sistem					■	■	■	■	■	■	
Revisi Kebutuhan Website						■	■	■	■	■	
Cek sistem yang sudah dibuat										■	
Dokumentasi Sistem											■

Perkenalan lingkungan kerja dilakukan pada minggu pertama. Pada hari pertama dilakukan pengenalan kepada staf dan karyawan di PT Antero Dwi Perkasa yang didampingi oleh Bapak Kok Ling selaku Senior Manager. Selanjutnya, pengarahan dan pemberian tugas diberikan langsung oleh Bapak Kok Ling selaku pembimbing lapangan.

Studi literatur dilakukan di minggu kedua. Kegiatan studi literatur bertujuan mencari mencari referensi terhadap proyek yang akan dikerjakan. Hal yang menjadi fokus utama dalam kegiatan ini adalah mempelajari PHP dan Bootstrap. Adapun kegiatan mempelajari metode yang digunakan untuk sistem yang dibuat.

Analisis dan perancangan kebutuhan sistem dilakukan pada minggu ketiga dan keempat. Pada kegiatan ini, dilakukan analisis dan perancangan apa saja yang dibutuhkan dalam sistem ini. Struktur sistem dirancang berdasarkan apa yang disampaikan pembimbing lapangan.

Proses pembuatan sistem dilakukan di minggu kelima hingga minggu kesepuluh. Pada kegiatan ini, dilakukan tahap *coding* berdasarkan hasil analisis dan perancangan yang telah dilakukan sebelumnya.

Revisi kebutuhan sistem dilakukan di minggu keenam hingga minggu kesepuluh. Pada kegiatan ini, pembimbing lapangan mengecek apakah sistem yang sudah dibuat sudah sesuai dengan yang diinginkan perusahaan. Revisi dilakukan di beberapa halaman, yaitu *index*, *history*, dan *export\_pdf*.

Pengecekan sistem yang sudah dibuat dilakukan di minggu kesepuluh. Pengecekan sistem dilakukan untuk mengetahui apakah fungsionalitasnya sudah memenuhi syarat dan apakah ada *bug* dan *error*.

Dokumentasi sistem dilakukan di minggu kesebelas. Pada kegiatan kali ini dilakukan dokumentasi cara penggunaan sistem. Dokumentasi terdiri dari *screenshot* sistem yang telah dibuat dan penjelasan singkat terkait penggunaan sistem.

### **3.3 Uraian Pelaksanaan Kerja Magang**

Model pengembangan sistem *Agile* diterapkan sebagai bentuk yang dilaksanakan selama praktik kerja magang. Metode *Agile* memungkinkan pengembangan perangkat lunak yang memiliki requirement yang mudah berubah dengan cepat. Tahap analisis dilakukan di awal project dan pada model masih ada kemungkinan bahwa hasil dari tahap analisis mengalami perubahan tergantung dari kebutuhan sistem (Sigit, 2018).

#### **3.3.1 Proses Pelaksanaan**

##### **A. User Requirements**

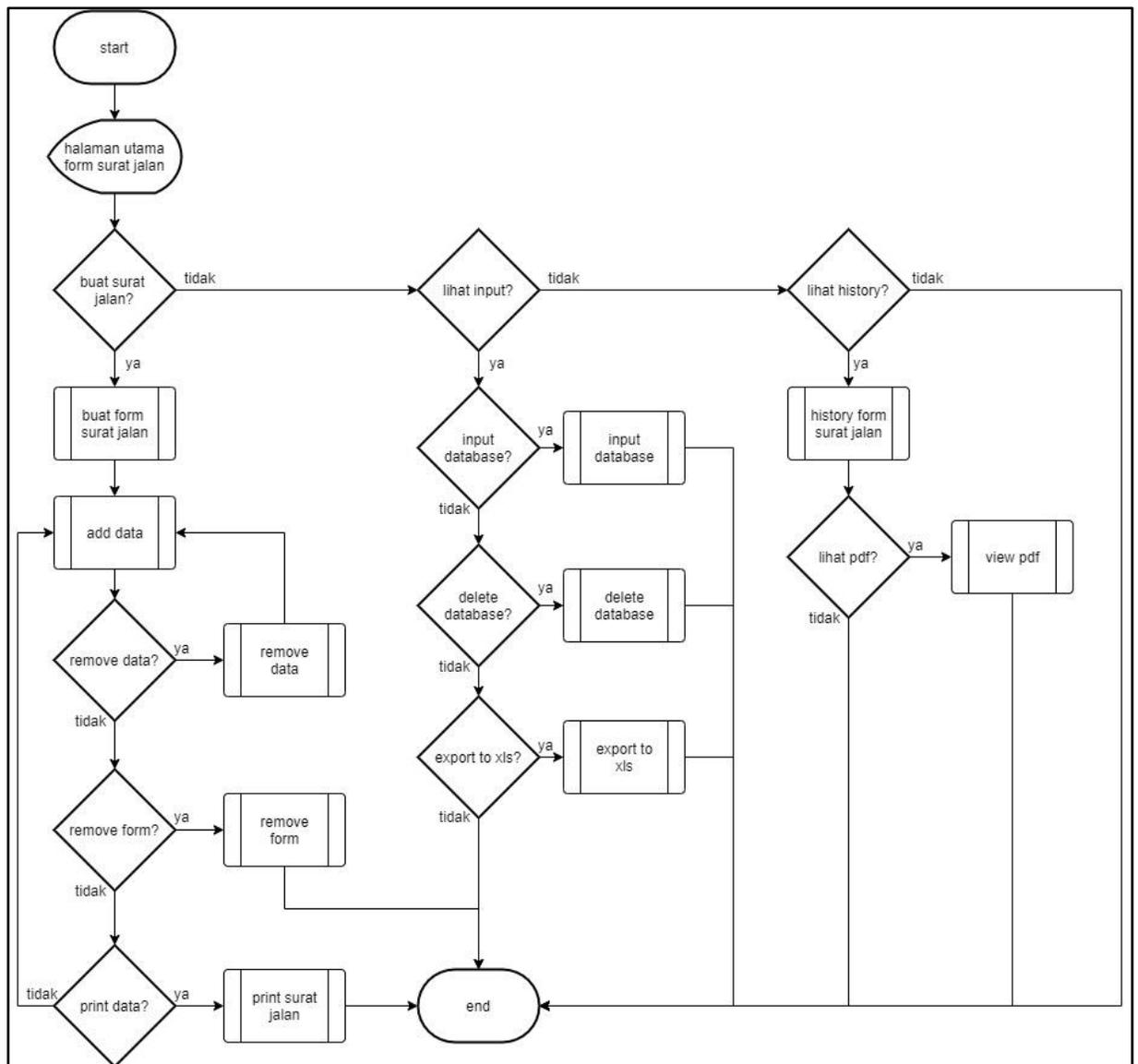
Perusahaan meminta agar sistem yang dibangun dapat melakukan beberapa fitur sebagai berikut.

1. *User* dapat langsung masuk ke dalam sistem tanpa harus melakukan *login*.
2. *User* dapat membuat *form* surat jalan.
3. *User* dapat memasukkan produk baru ke dalam *database*.
4. *User* dapat menghapus produk dari *database*.
5. *User* dapat men-*download* data produk pada *database* ke dalam format xls.
6. *User* dapat menambahkan *form* surat jalan dari data produk yang tersedia.
7. *User* dapat menyimpan *form* surat jalan dalam format pdf, men-*download* surat jalan tersebut, dan mencetaknya dalam bentuk fisik.

8. *User* dapat melihat *history* yang bisa diurutkan berdasarkan *company* dan tanggal.
9. *User* tidak perlu meng-*input* tanggal karena tanggal sudah ditambahkan secara otomatis dari sistem.

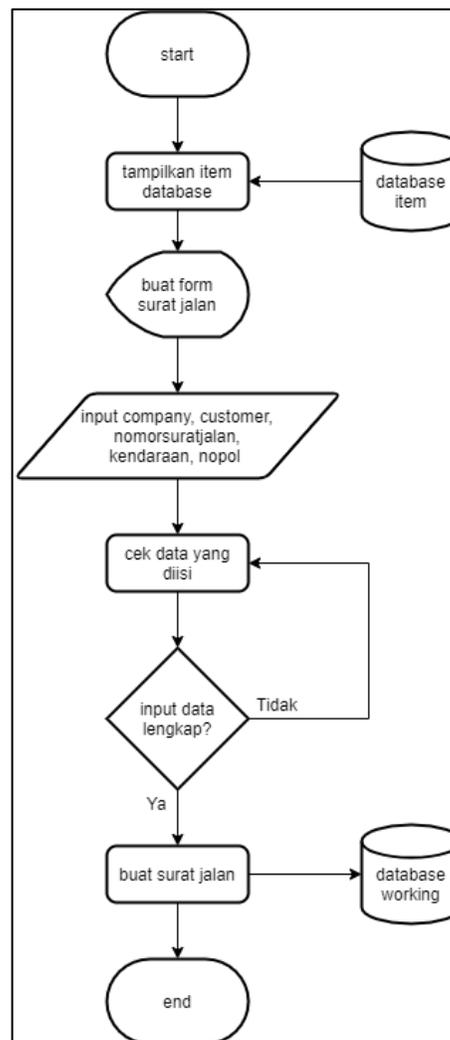
## B. Perancangan Aplikasi

### B.1 Flowchart



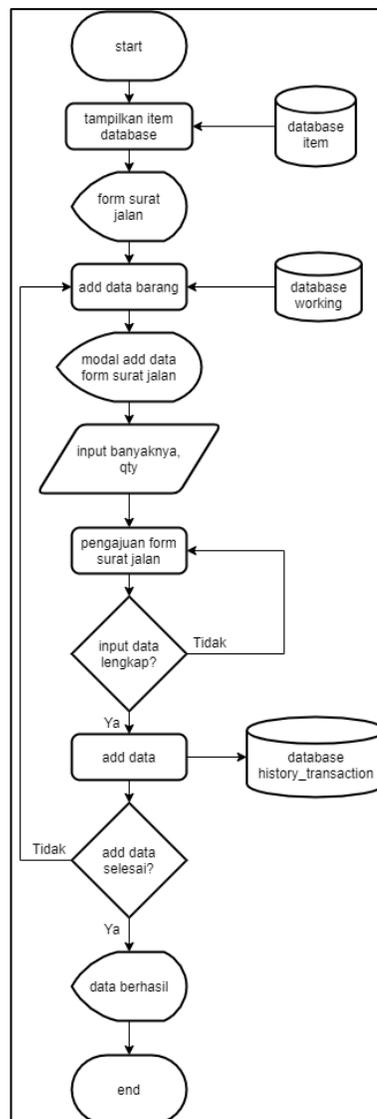
Gambar 3.1 Flowchart Form Surat Jalan

Gambar 3.1 menunjukkan urutan proses berjalannya dari *form* surat jalan. Apabila *user* menjalankan sistem, *user* akan masuk ke dalam *form* surat jalan. Jika *user* telah melewati proses *form* surat jalan, *user* dapat *add* data pada form surat jalan. Ketika proses *add* data selesai, *user* dapat memilih proses *remove* data, *remove* form, dan *print* surat jalan. Di dalam proses diatas sistem dapat *input* data ke *database*, *delete* data di *database*, dan *export file* menjadi *file* xls. Selain itu pada sistem form surat jalan, *user* dapat melihat *history form* surat jalan yang pernah dibuat. *History form* surat jalan juga dapat dilihat sebagai *file* pdf.



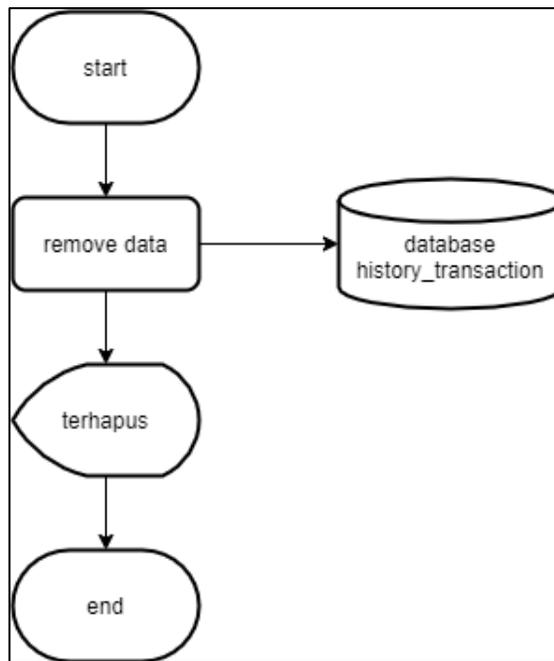
Gambar 3.2 *Flowchart* Buat *Form* Surat Jalan

Gambar 3.2 menunjukkan urutan proses dari halaman *form* surat jalan. Pada *flowchart* tersebut dijelaskan bahwa pada saat *user* masuk ke dalam *form* surat jalan. *User* harus menginput data *company*, *customer*, nomor surat jalan, kendaraan, dan nomer polisi. Setelah *user* sudah menginput data, maka sistem mengecek apakah data yang diisii sudah lengkap atau tidak. Ketika belum lengkap *user* harus mengisi data dengan lengkap. Apabila datanya sudah lengkap diisi *user*, maka baru dapat membuat surat jalan.



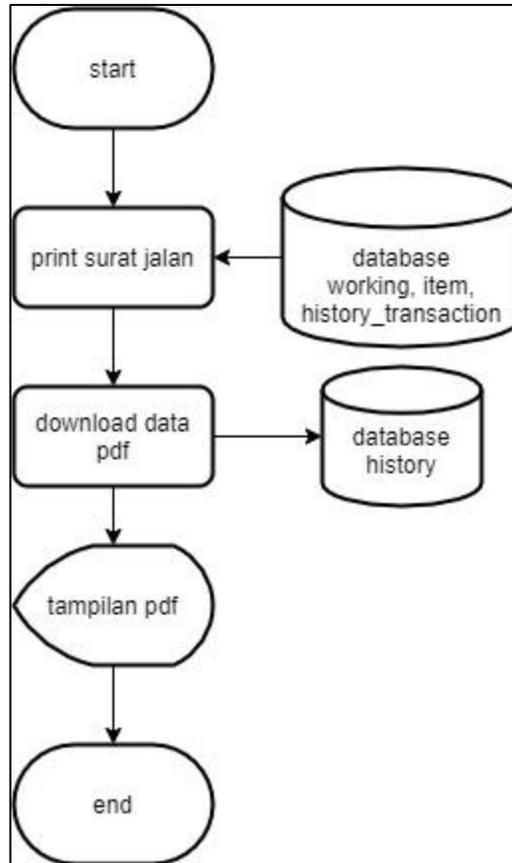
Gambar 3.3 *Flowchart Add Data*

Gambar 3.3 menunjukkan urutan proses dari halaman *form* surat jalan setelah input data. Pada *flowchart* tersebut dijelaskan bahwa pada saat *user* masuk ke dalam *form* surat jalan dan telah menginput data *company*, *customer*, nomor surat jalan, kendaraan, dan nomer polisi. Maka *user* dapat *add* data barang ke dalam halaman *form* surat jalan yang ada. *User* dapat *input* data barang dengan data barang yang sudah ada. Selain itu *user* dapat input data barang yang dibuat sendiri dengan memasukan data nama barang dan *quantity* barang. Setelah data barang dibuat, data tersebut masuk ke dalam *list* barang. Jika mau memasukan data barang ke *form* surat jalan, cukup menekan *add* barang.



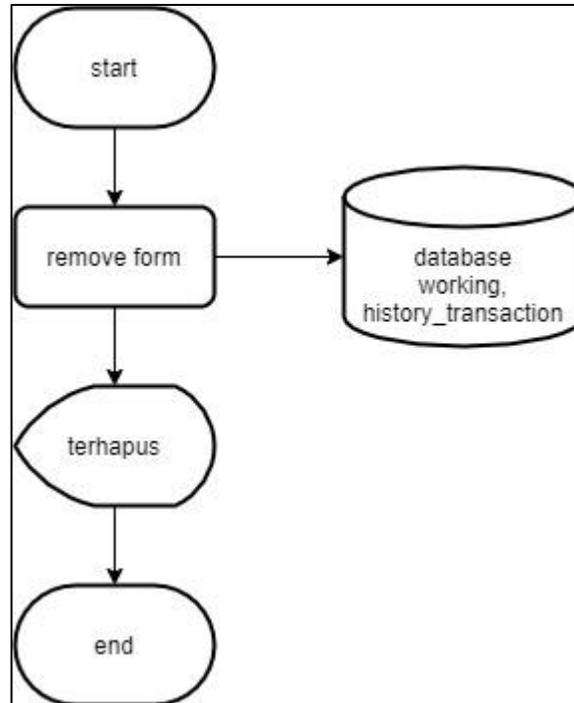
Gambar 3.4. *Flowchart Remove Data*

Gambar 3.4 menunjukkan urutan proses dari *remove list* data dari *form* surat jalan. Pada *flowchart* tersebut dijelaskan bahwa pada saat *user* ingin menghapus suatu *list* data dari *form* surat jalan, maka *user* akan menekan tombol *remove data*. Setelah itu, sistem akan langsung otomatis menghapus *list* data dari *database*.



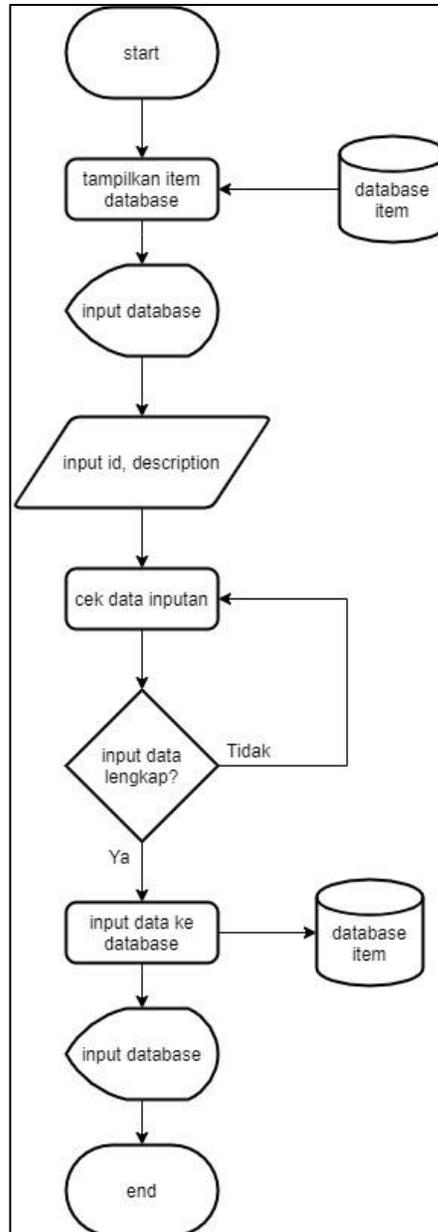
Gambar 3.5 *Flowchart Print Surat Jalan*

Gambar 3.5 menunjukkan urutan proses dari *print* surat jalan. Pada *flowchart* tersebut dijelaskan bahwa pada saat *user* masuk ke dalam *form* surat jalan dan telah menginput data *company*, *customer*, nomor surat jalan, kendaraan, dan nomor polisi, serta *user* telah melakukan *add* data barang, maka *user* dapat melakukan *print* surat jalan. Pada proses diatas ketika *user* menekan tombol *print* surat jalan maka dengan otomatis sistem melakukan *export* data yang dapat langsung dilihat di halaman *export\_pdf.php*. Data pdf yang telah dibuat secara otomatis akan di-*download* di dalam direktori yang sudah ditentukan.



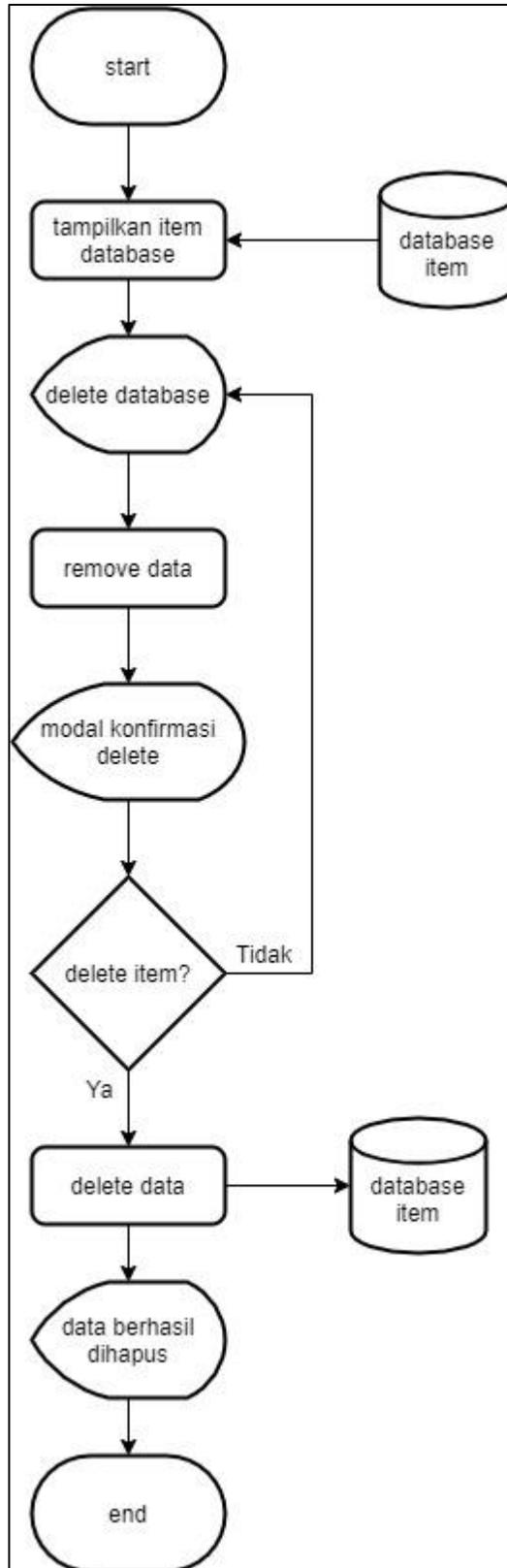
Gambar 3.6 *Flowchart Remove Form*

Gambar 3.6 menunjukkan urutan proses dari *remove form* surat jalan. Pada *flowchart* tersebut dijelaskan bahwa pada saat *user* telah membuat *form* surat jalan. Surat jalan yang telah dibuat dapat di-*remove* dengan menekan tombol *remove form*. Lalu ketika telah menekan tombol *remove form*, data *form* surat jalan berhasil dibuang.



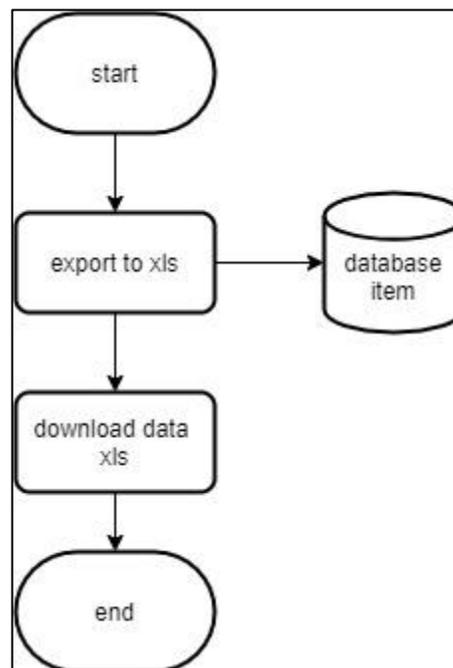
Gambar 3.7 *Flowchart Input Database*

Gambar 3.7 menunjukkan urutan proses dari *input* data barang ke *database*. Pada *flowchart* tersebut dijelaskan bahwa pada saat *user input* data barang ke dalam *database*. Input disini berupa *id* dan deskripsi, setelah input data sistem mengecek apakah data sudah di *input* sudah diisi atau belum. Jika data sudah diisi secara lengkap, maka data yang sudah diisi langsung di *input* ke dalam *database*.



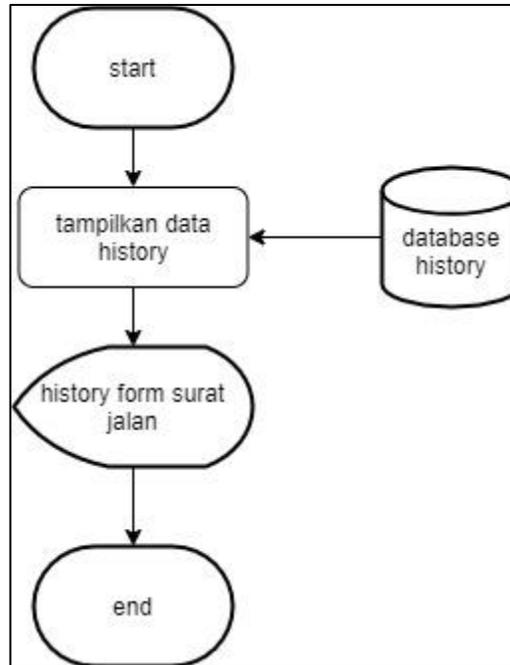
Gambar 3.8 *Flowchart Delete Database*

Gambar 3.8 menunjukkan urutan proses dari *delete* data barang di *database*. Pada *flowchart* tersebut dijelaskan bahwa pada saat *user* menekan tombol *delete database* maka data yang ingin di *remove* dikonfirmasi oleh *modal* yang berisi *delete verification*. Lalu muncul pertanyaan *delete item* atau tidak, kalau memilih tidak *delete* maka langsung kembali ke halaman data barang. Kalau memilih *delete* data maka data barang berhasil dihapus.



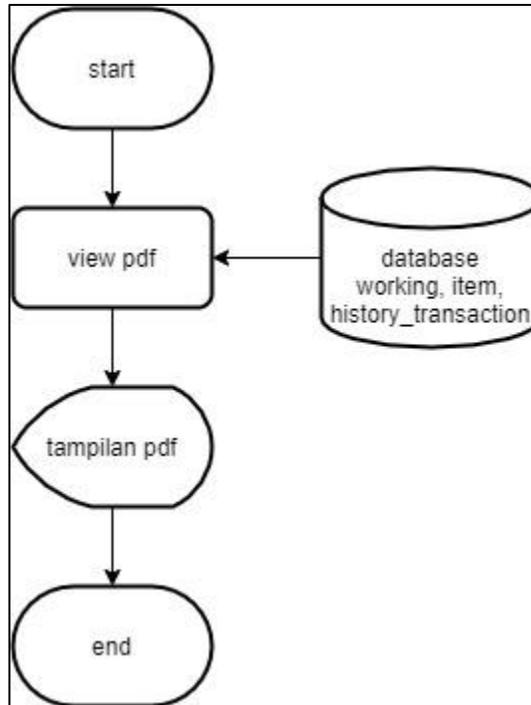
Gambar 3.9 *Flowchart Export to XLS*

Gambar 3.9 menunjukkan urutan proses dari *export* data pengeluaran barang menjadi file xls. Pada *flowchart* tersebut dijelaskan bahwa *user* dapat *export* data pengeluaran barang yang terdapat pada surat jalan menjadi *file* xls. pada saat *user* menekan tombol *export to xls* maka dengan otomatis data pengeluaran barang yang terdapat pada surat jalan langsung data di-*download* dalam bentuk xls.



Gambar 3.10 *Flowchart History Form Surat Jalan*

Gambar 3.10 menunjukkan merupakan urutan proses dari *history form* surat jalan. Pada *flowchart* tersebut dijelaskan bahwa *user* dapat melihat *history form* surat jalan yang telah dibuat sebelumnya. Surat jalan yang sudah dibuat dapat dilihat kembali mengenai list data *company*, nomor surat jalan, tanggal surat yang dibuat, dan tombol untuk membuka *form* surat jalan yang telah dibuat sebelumnya.

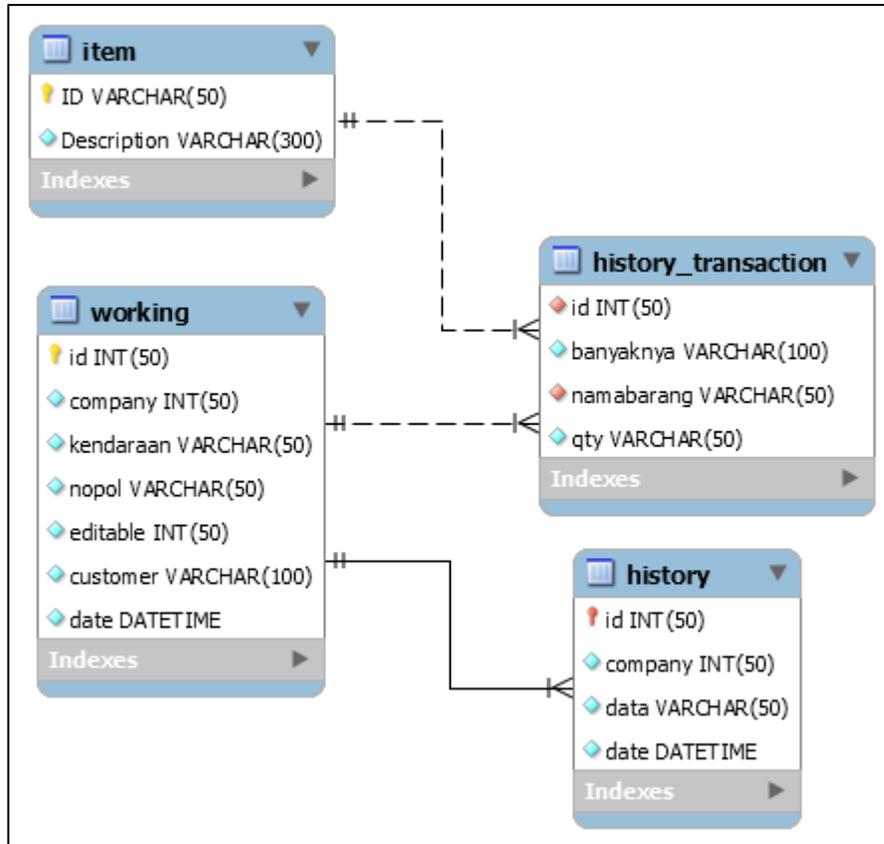


Gambar 3.11 *Flowchart View PDF*

Gambar 3.11 menunjukkan proses melihat *form* surat jalan yang telah dibuat didalam laman *history*. Pada *flowchart* tersebut dijelaskan bahwa pada saat *user* menekan tombol *view pdf*, maka pdf akan dipanggil melalui *web* direktori *server* yang sesuai dengan kolom yang dipilih.

## B.2 Database Schema

Gambar 3.12 menunjukkan *database schema* yang terdiri dari tabel *item*, *history\_transaction*, *working*, dan *history*. Tabel *item* memiliki kolom *id primary key* yang langsung *connect* ke tabel *history\_transaction* pada kolom *namabarang* yang merupakan *foreign key* dari tabel *item*. Tabel *working* memiliki kolom *id primary key* yang langsung *connect* ke tabel *history\_transaction* dan tabel *history* dimana tabel *history\_transaction* memiliki kolom *id foreign key* dari tabel *working*, serta tabel *history* memiliki kolom *id foreign key* dari tabel *working*.



Gambar 3.12 Gambar *Database Schema* Sistem

Berikut merupakan penjabaran dari tabel-tabel yang berada di dalam *database* yang digunakan selama proses pembuatan sistem.

1. Tabel *Item*

Tabel 3.2 Struktur Tabel *Item*

Nama Kolom	Tipe data	Keterangan
<i>id</i>	Varchar(50)	<i>Primary key</i>
<i>Description</i>	Varchar(300)	Data deskripsi barang

Tabel 3.2 menggambarkan struktur tabel *item* yang terdapat di *database*. Tabel ini menyimpan data-data produk yang ada pada sistem. Di tabel *item* ini terdapat *id* yang merupakan *primary key* dari setiap indentitas dari *item* atau barang. Kolom *description* merupakan penjelasan mengenai penjelasan barang atau *item*.

2. Tabel *Working*

Tabel 3.3 Struktur Tabel *Working*

<b>Nama Kolom</b>	<b>Tipe data</b>	<b>Keterangan</b>
<i>id</i>	Int(50)	<i>Primary key</i>
<i>company</i>	Int(50)	Data company
kendaraan	Varchar(50)	Data kendaraan
nopol	Varchar(50)	Data nomor polisi kendaraan
<i>editable</i>	Int(50)	Untuk memberitahu form surat jalan sedang dikerjakan atau belum
<i>customer</i>	Varchar(100)	Data customer
<i>date</i>	Datetime	Data tanggal dikirim

Tabel 3.3 menggambarkan struktur tabel *working* yang terdapat di *database*. Tabel ini menyimpan data yang berhubungan dengan indentias surat jalan. Di tabel *working* ini terdapat kolom *id* yang merupakan *primary key*. Kolom *id* ini memiliki isi yaitu nomor surat jalan. Kolom *company* merupakan data berisi nama perusahaan yang mengirimkan barang. Kolom *kendaraan* merupakan data berisi nama kendaraan yang melakukan pengiriman barang. Kolom *nopol* merupakan data berisi nomor polisi kendaraan yang mengirimkan barang. Kolom *editable* merupakan data berisi nomor yang digunakan untuk memberitahu *form* surat jalan sedang dikerjakan atau tidak. Kolom *customer* merupakan nama penerima barang yang akan dikirimkan. Sedangkan kolom *date* merupakan data tanggal surat jalan dibuat.

3. Tabel *History\_transaction*

Tabel 3.4 Struktur Tabel *History\_transaction*

<b>Nama Kolom</b>	<b>Tipe data</b>	<b>Keterangan</b>
<i>id</i>	Int(50)	<i>Foreign Key</i>
banyaknya	Varchar(100)	Data banyak barang
namabarang	Varchar(50)	<i>Foreign key</i>
<i>qty</i>	Varchar(50)	Data banyaknya barang

Tabel 3.4 menggambarkan struktur tabel *history\_transaction* yang terdapat di *database*. Tabel ini menyimpan beberapa data yang berhubungan dengan *list* data

yang tercatat dalam *form* surat jalan. Di tabel ini terdapat *id* yang merupakan nomor surat jalan sekaligus *foreign key* dari tabel *working* pada kolom *id*. Kolom banyaknya merupakan data gabungan produk dari beberapa *qty* yang akan dikirimkan. Kolom *namabarang* merupakan data nama barang yang dikirimkan. Kolom *qty* merupakan kuantitas barang yang dikirimkan.

#### 4. Tabel *History*

Tabel 3.5 Struktur Tabel *History*

Nama Kolom	Tipe data	Keterangan
id	Int(50)	Foreign Key
company	Int(50)	Data company
data	Varchar(50)	Direktori web server
date	Datetime	Data tanggal dikirim

Tabel 3.5 menggambarkan struktur tabel *history* yang terdapat di *database*. Tabel ini menyimpan beberapa data yang berhubungan dengan penyimpanan *form* surat jalan yang sudah dibuat. Di tabel *history* terdapat kolom *id* yang merupakan nomor surat jalan sekaligus *foreign key* dari tabel *working* pada kolom *id*. Kolom *company* merupakan data yang berisi nama perusahaan yang membuat *form* surat jalan. Kolom *data* merupakan data yang berisi *web* direktori *server* dimana *file* pdf disimpan. Sedangkan kolom *date* merupakan data tanggal surat jalan dibuat.

### B.3 Rancangan Antarmuka

Rancangan Antarmuka merupakan sebuah *mockup* yang akan diajukan berdasarkan apa yang diminta oleh pembimbing lapangan.

#### 1. Halaman Utama

Gambar 3.13 menunjukkan halaman utama dari sistem yang dibuat. Halaman ini terdiri atas halaman *form* surat jalan. Pada halaman ini *user* harus *input data* yaitu

*data* perusahaan, perusahaan yang dikirimkan barang, *input* nomor surat, tipe kendaraan, dan nomor polisi. Ketika *user* memilih tombol buat surat, *user* baru dapat menambahkan atau *add* barang.

Antero Dwi Perkasa <a href="#">home</a> <a href="#">tutorial</a> <a href="#">other..</a>	
FORM SURAT JALAN	
Pilih Perusahaan	<input type="text"/>
Kepada	<input type="text"/>
Input No. Surat Jalan	<input type="text"/>
Tipe Kendaraan	<input type="text"/>
No. Pol	<input type="text"/>
<input type="button" value="Buat Surat!"/>	
INPUT FORM SURAT JALAN	
Search data...	
ID	Description

Gambar 3.13 Halaman Utama

## 2. Halaman Utama *Add Data*

Gambar 3.14 menunjukkan halaman utama *add data* dari sistem yang dibuat. Halaman ini terdiri atas halaman *form* surat jalan. Pada halaman ini *user* yang telah *input* data *form* surat jalan dapat menambahkan atau *add* pengeluaran barang pada *form* surat jalan yang ada.

Antero Dwi Perkasa [home](#) [tutorial](#) [other..](#)

### FORM SURAT JALAN

Company :  
 Kepada :  
 No. Surat Jalan :  
 Kendaraan :  
 No. Pol :

banyaknya	Nama Barang	qty	Remove

### INPUT FORM SURAT JALAN

Search data...

ID	Description	Click to Add
		add
		add
		add

Gambar 3.14 Halaman Utama *Add Data*

3. Halaman Utama *Input Database*

Gambar 3.15 menunjukkan halaman utama *input database* data barang. Pada halaman ini *user* dapat menambahkan jumlah barang yang ada. Jumlah barang yang sudah dibuat dapat ditambahkan di *form* surat jalan. Data barang pada halaman ini juga dapat dihapus jika tidak diperlukan lagi. Selain itu *user* dapat *download* tabel yang berisi *id* barang dan deskripsi barang dalam format xls.

Antero Dwi Perkasa [home](#) [tutorial](#) [other->input new data](#)

### INPUT DATABASE

New ID   
 New Desc

### DATABASE

Export to xls

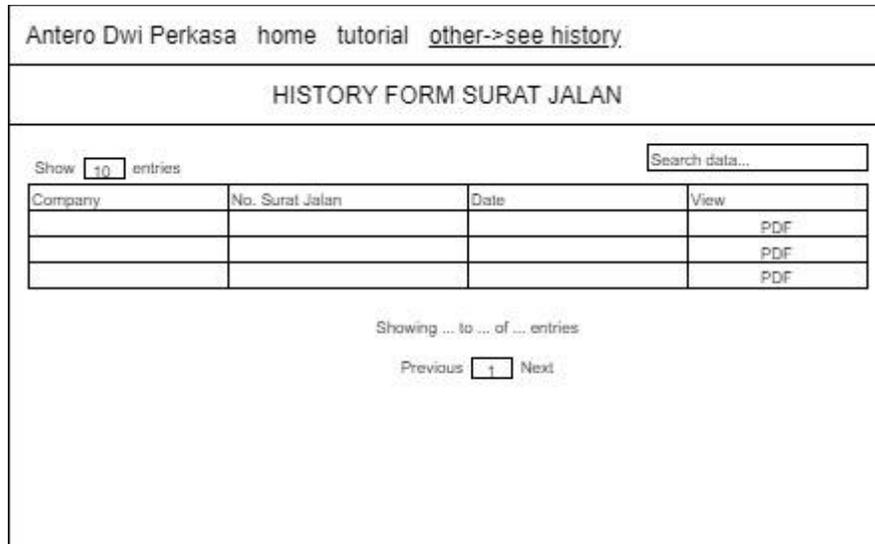
Search data...

ID	Description	
		Delete

Gambar 3.15 Halaman Utama *Input Database*

4. Halaman *History*

Gambar 3.16 menunjukkan halaman *history* dari data *form* surat jalan. Pada halaman ini *user* dapat melihat *history* surat jalan yang sudah dibuat. Pada halaman ini dapat melihat *history* berdasarkan *company*, nomor surat jalan, tanggal, dan juga dapat melihat *form* yang telah dibuat sebelumnya.



Gambar 3.16 Halaman *History*

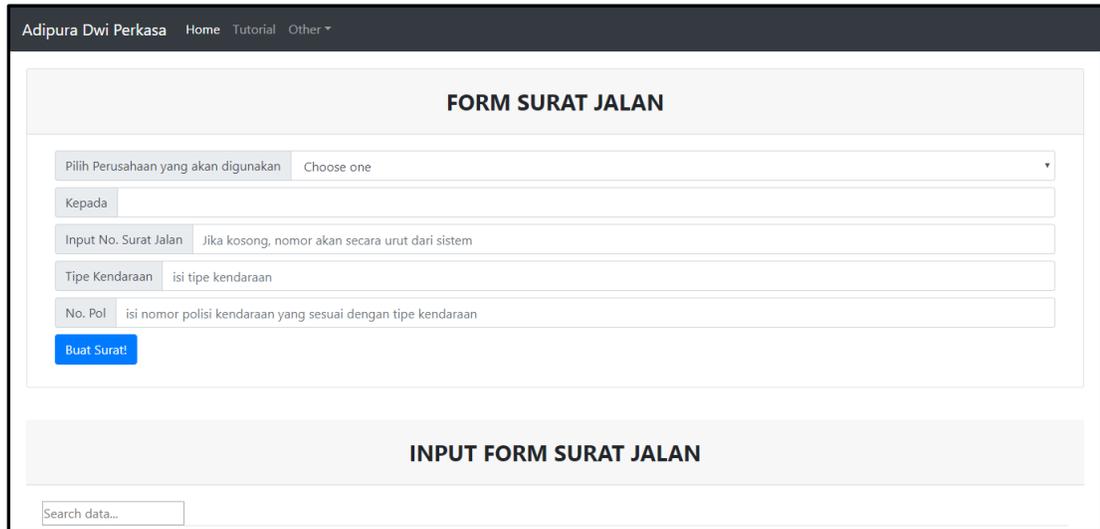
5. Halaman *Export pdf*

Gambar 3.17 menunjukkan halaman *export pdf form* surat jalan. Pada halaman ini *user* dapat *export file* dari *form* surat jalan yang sudah dibuat sebelumnya. Selain itu, halaman ini juga digunakan setelah *form* surat jalan selesai dibuat pada Gambar 3.14.



## C. Implementasi

Gambar 3.19 merupakan tampilan awal dari sistem. Pada halaman ini, *user* dapat membuat *form* surat jalan yang akan dijelaskan lebih lanjut.



The screenshot shows a web application interface for creating a road permit form. At the top, there is a navigation bar with the text 'Adipura Dwi Perkasa' and links for 'Home', 'Tutorial', and 'Other'. The main content area is titled 'FORM SURAT JALAN' and contains several input fields: a dropdown menu for 'Pilih Perusahaan yang akan digunakan' with the placeholder 'Choose one', a text input for 'Kepada', a text input for 'Input No. Surat Jalan' with a note 'Jika kosong, nomor akan secara urut dari sistem', a text input for 'Tipe Kendaraan' with a note 'isi tipe kendaraan', and a text input for 'No. Pol' with a note 'isi nomor polisi kendaraan yang sesuai dengan tipe kendaraan'. A blue button labeled 'Buat Surat!' is positioned below the inputs. Below the form, there is a section titled 'INPUT FORM SURAT JALAN' and a search bar with the placeholder 'Search data...'.

Gambar 3.19 Halaman *Indeks*

Pada halaman utama, *user* akan diminta untuk meng-*input* data berupa pilih perusahaan yang akan digunakan, kepada, input nomor surat jalan, tipe kendaraan, dan nomor polisi. Setelah tombol buat surat ditekan, maka muncul tampilan yang terlihat pada Gambar 3.20.

Adipura Dwi Perkasa Home Tutorial Other ▾

### FORM SURAT JALAN

**Remove form**

Company : Antero Dwi Perkasa  
 Kepada : PT Djarum Indonesia TBK, Bapak Lie Seung Gie.  
 No. Surat Jalan : 8.  
 Kendaraan : Mitsubishi Truck.  
 No. Pol : B 1234 DBS.

Banyaknya	Nama Barang	Qty	Remove
-----------	-------------	-----	--------

### INPUT FORM SURAT JALAN

Search data...

ID	Description	Click to add
APM-001	Sedotan buble 820 bungkus plastik @ 10.000 pcs	<b>Add</b>
APM-003	Sedotan sendok hitam @ 10.000 pcs	<b>Add</b>

Gambar 3.20 Halaman *Add Form* Surat Jalan

Pada kolom *input form* surat jalan terdapat tombol untuk menambahkan produk ke dalam *list form* surat jalan. Ketika tombol *add* ditekan, maka akan keluar *modal* seperti pada Gambar 3.21.

No. Pol : B 1234 DBS.

Banyaknya	Nama Barang	Qty	Remove
-----------	-------------	-----	--------

Search data...

ID	Description	Click to add
APM-001	Sedotan buble 820 bungkus plastik @ 10.000 pcs	<b>Add</b>
APM-003	Sedotan sendok hitam @ 10.000 pcs	<b>Add</b>
BIC-001	BG 135 @ 300 set   IC	<b>Add</b>
BIC-002	BG 1770 @ 500 set   IC	<b>Add</b>
BIC-003	CT - 120 @ 300 set   IC	<b>Add</b>

#### Tambah Data Form Surat Jalan

Banyaknya

ID

Nama Barang

Qty

**Submit** **Cancel**

Gambar 3.21 Halaman *Modal Add Form* Surat Jalan

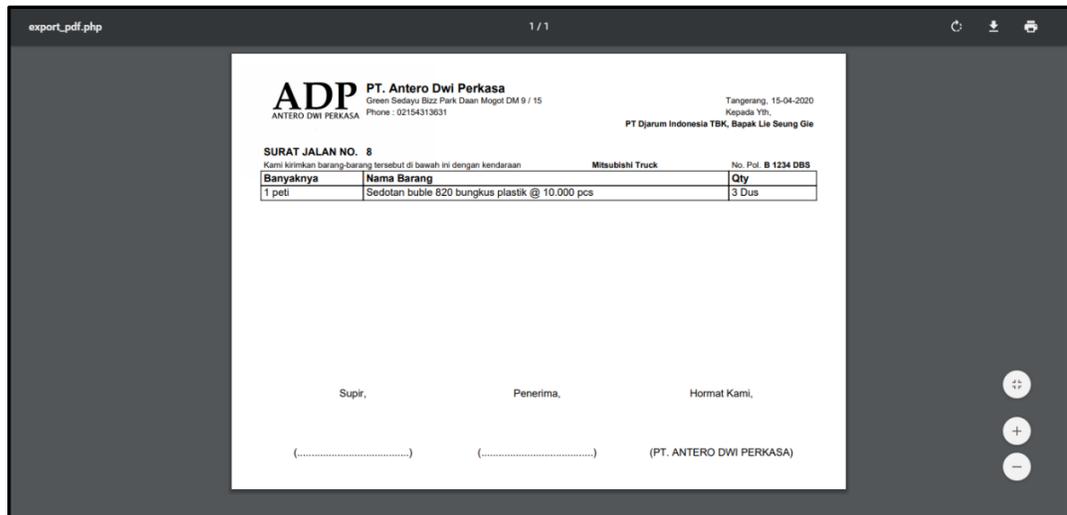
Pada Gambar 3.21 akan diminta untuk meng-*input* banyaknya dan qty. Jika tombol *cancel* ditekan, maka *modal add form* surat jalan akan keluar. Jika tombol *submit* yang ditekan, maka tampilan akan terlihat pada Gambar 3.22.

Banyaknya	Nama Barang	Qty	Remove
1 peti	Sedotan bubble 820 bungkus plastik @ 10.000 pcs	3 Dus	Remove

Gambar 3.22 Halaman Utama

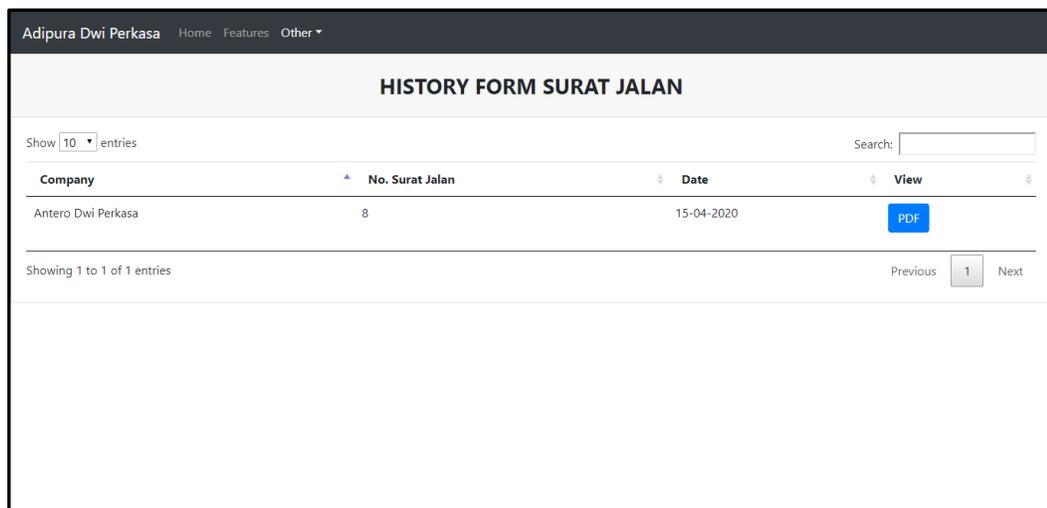
Apabila ingin menambahkan produk lagi, maka diminta untuk mengulangi langkah sebelumnya sesuai dengan kebutuhan. Misal pada *list form* surat jalan ada yang ingin dihapus maka akan diminta untuk menekan tombol *remove*. Ketika tombol *remove* ditekan maka sistem akan menghapus *list* data dalam 1 baris yang sama. Jika *form* surat jalan ingin dihapus maka akan diminta untuk menekan tombol *remove form*. Akan terjadi penghapusan *form* surat jalan yang dibuat dan kembali ke halaman utama seperti pada Gambar 3.19 apabila tombol *remove form* ditekan.

Ketika *list* pada *form* surat jalan dan *form* surat jalan sudah selesai, maka akan diminta untuk menekan tombol *print and save to database*. Jika tombol *print and save to database* ditekan maka akan muncul halaman baru sesuai dengan Gambar 3.23. Pada Gambar 3.23 juga dapat men-*download* file pdf dan *print file* sesuai kebutuhan.



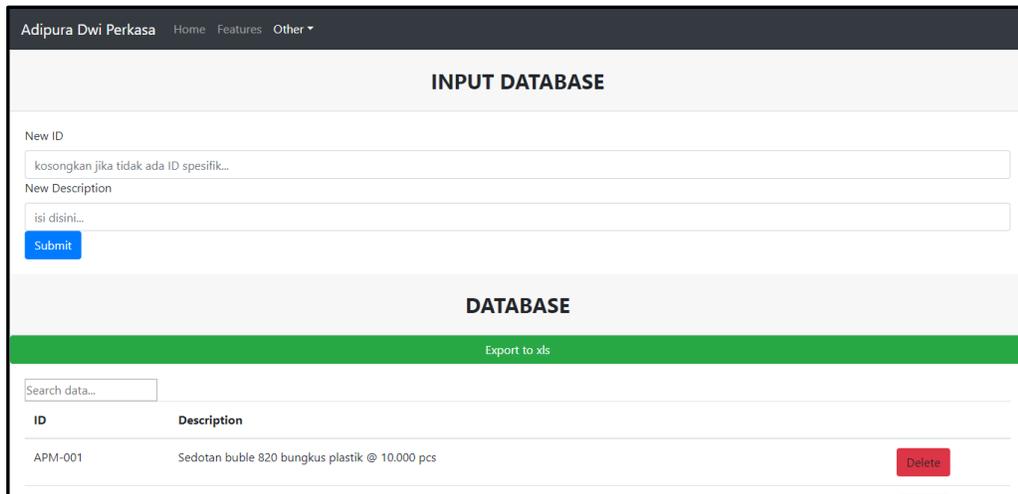
Gambar 3.23 *Export Form* ke PDF

*History form* surat jalan dapat dilihat pada Gambar 3.24. Jika ingin melihat *form* surat jalan yang telah dibuat sebelumnya maka dapat menekan tombol pdf, dan akan terlihat seperti pada Gambar 3.23.



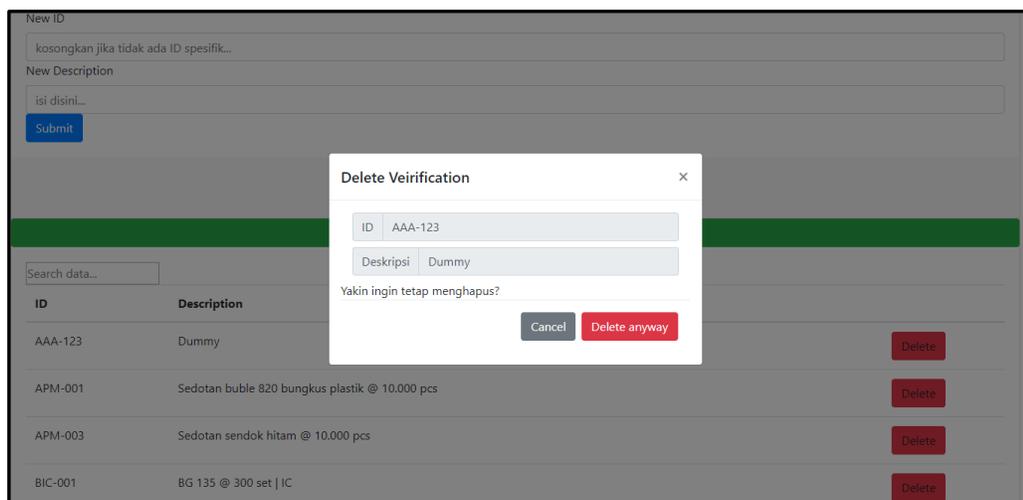
Gambar 3.24 Halaman *History Form* Surat Jalan

Pada Gambar 3.25 menampilkan halaman *input database*. Halaman dapat menambahkan produk baru ke dalam database, menghapus produk yang tidak dibutuhkan, dan men-*download* tabel *database* dengan menekan tombol *export to xls*.



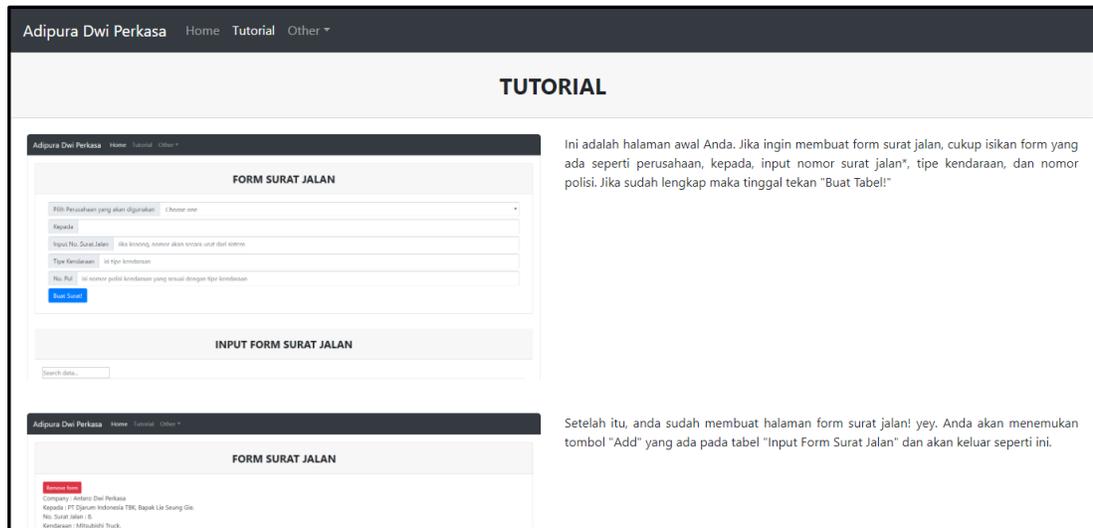
Gambar 3.25 Halaman *Input Data ke Database*

Jika ada produk yang ingin dihapus dapat menekan tombol *delete*. Apabila tombol *delete* ditekan maka akan memunculkan *modal delete verification*. Ketika ingin menghapus data, tekan tombol *delete anyway* dan produk yang dipilih akan dihapus oleh sistem seperti pada Gambar 3.26.



Gambar 3.26 Halaman *Modal Delete Data*

Gambar 3.27 menunjukkan halaman *tutorial sistem form surat jalan*. Pada halaman ini, *user* dapat melihat cara penggunaan sistem *form surat jalan*.



Gambar 3.27 Halaman *Tutorial Form* Surat Jalan

### 3.3.2 Kendala yang Ditemukan

Beberapa kendala yang ditemukan saat pelaksanaan kerja magang adalah sebagai berikut.

1. Terbatasnya pengetahuan mengenai beberapa fungsi dalam bahasa pemrograman PHP dan JavaScript.
2. Dalam perancangan sistem, terkadang tidak sesuai dengan keinginan pembimbing lapangan.
3. Saat menerima tugas baru yang diberikan, maksud yang diinginkan pembimbing lapangan mengenai sistem yang dibuat sering kali salah dimengerti sehingga sering terjadi kesalahpahaman.

### 3.3.3 Solusi atas Kendala yang Ditemukan

Solusi atas kendala yang ditemukan saat pelaksanaan kerja magang adalah sebagai berikut.

1. Explorasi dan belajar secara mandiri mengenai penulisan kode dan mempelajari fungsi-fungsi dalam bahasa pemrograman PHP dan JavaScript.
2. Lebih sering untuk berkoordinasi dengan pembimbing lapangan tentang sistem yang dibuat.
3. Mendengarkan dengan baik instruksi dari pembimbing lapangan dan teliti saat mengerjakan tugas, serta bertanya apabila ada hal yang lupa dan yang kurang dimengerti.