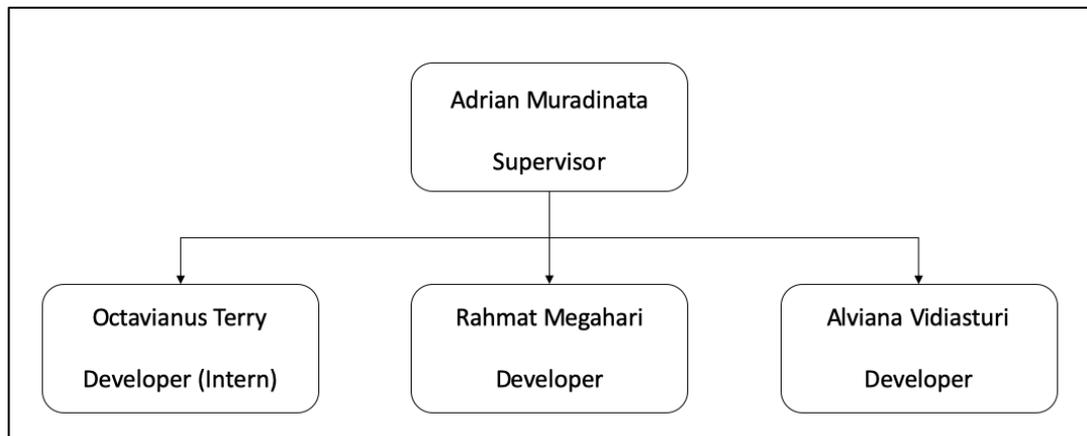


## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1. Kedudukan dan Koordinasi

Kerja magang dilakukan dalam sebuah tim yang berisi 3 orang *developer* di bawah pengawasan Bapak Adrian Muradinata sebagai *supervisor* dan pembimbing kerja magang di CV Trijaya Mandiri. Adapun bagan struktur koordinasi kerja magang di CV Trijaya Mandiri berdasarkan penjelasan di atas dapat dilihat pada Gambar 3.1.



Gambar 3.1 Struktur Koordinasi Kerja Magang

#### 3.2. Tugas yang Dilakukan

Selama melaksanakan kerja magang, peserta kerja magang melakukan beberapa pekerjaan setiap minggunya. Adapun aktivitas peserta magang dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tabel Realisasi Kerja Magang

Minggu ke-	Jenis Pekerjaan yang Dilakukan
1.	<ul style="list-style-type: none"><li>• Training</li><li>• Develop sistem stok</li></ul>
2.	<ul style="list-style-type: none"><li>• Develop <i>user interface</i> modul penjualan</li><li>• <i>Weekly Meeting</i></li></ul>
3.	<ul style="list-style-type: none"><li>• Implementasi dan revisi <i>user interface</i> modul penjualan.</li><li>• <i>Weekly Meeting</i></li></ul>
4.	<ul style="list-style-type: none"><li>• Implementasi metode pembayaran modul penjualan</li></ul>

	<ul style="list-style-type: none"> <li>• <i>Weekly Meeting</i></li> </ul>
5.	<ul style="list-style-type: none"> <li>• <i>Bug fixing</i> metode pembayaran</li> <li>• <i>Weekly Meeting</i></li> </ul>
6.	<ul style="list-style-type: none"> <li>• Implementasi update stok <i>database</i> dalam modul penjualan</li> <li>• <i>Weekly Meeting</i></li> </ul>
7.	<ul style="list-style-type: none"> <li>• Implementasi <i>history</i> penjualan dalam modul penjualan</li> <li>• <i>Weekly Meeting</i></li> </ul>
8.	<ul style="list-style-type: none"> <li>• <i>Bug fixing</i> modul penjualan</li> <li>• <i>Weekly Meeting</i></li> </ul>
9.	<ul style="list-style-type: none"> <li>• <i>Bug fixing</i> modul penjualan</li> </ul>

Sebelum melakukan kerja magang, semua anggota melakukan *training* terlebih dahulu di mana *developer* akan mempelajari mengenai aplikasi yang akan dibuat, bahasa pemrograman yang digunakan, dan dasar-dasar sistem penjualan dalam suatu perusahaan. Tahap selanjutnya pembagian tugas di mana setiap *developer* mengerjakan masing-masing modul yang diberikan. Dalam pengerjaan satu modul akan diberikan deadline oleh *supervisor* dan harus diselesaikan sebelum *deadline*. Pengerjaan modul juga termasuk perancangan *user interface* yang dilakukan oleh *developer*.

Setelah pembagian tugas dilakukan, pengerjaan terbagi atas beberapa modul di mana modul yang dikerjakan antara lain modul penjualan, stok, dan *history* penjualan. Dalam tahap pengerjaan, modul yang dikerjakan adalah modul penjualan yang terdiri dari pembayaran, pengurangan stok, menyimpan, dan menampilkan *history* penjualan.

Setelah pengerjaan modul selesai dilakukan, pekerjaan yang dilakukan selama melaksanakan kerja magang adalah melakukan *bug fixing* dan melakukan perubahan sesuai revisi yang ada pada *product* TMCashier. Oleh karena itu, dalam pengerjaan laporan magang ini, isi yang ditulis akan difokuskan pada modul penjualan. Modul penjualan

adalah modul yang berperan dalam sistem penjualan barang yang ada di dalam perusahaan. Modul penjualan terdiri dari pembayaran dan pengurangan stok dalam sistem penjualan. Data-data yang digunakan berupa harga barang, stok barang, total harga barang, dan total yang dibayar oleh *customer*. Selanjutnya stok barang akan dikurangi sesuai dengan jumlah barang yang dibeli kemudian akan dilakukan update pada *database* stok barang. *User* juga dapat melihat *history* penjualan pada pilihan *history* penjualan.

### **3.3. Pelaksanaan Kerja Magang**

Pelaksanaan kerja magang dibagi menjadi beberapa tahap yaitu Analisa kebutuhan, perancangan sistem, desain antar muka hingga implementasi sistem.

#### **3.3.1. Analisa Kebutuhan**

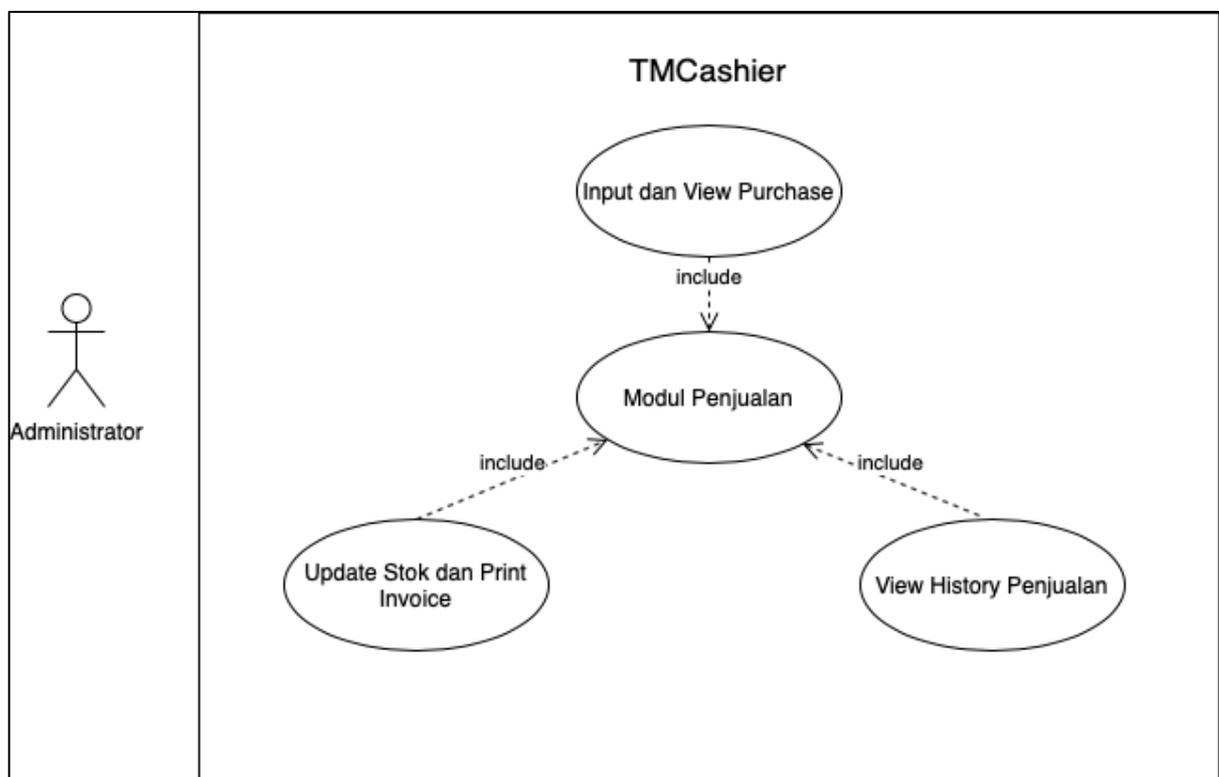
Berikut merupakan beberapa tools yang digunakan dalam membangun aplikasi TMCashier:

- a. Bahasa pemrograman yang digunakan yaitu *HyperText Markup Language*, *Cascading Style Sheets*, dan Javascript dengan menggunakan *framework Code Igniter*.
- b. *Integrated Development Environment* yang digunakan dalam pengembangan aplikasi TMCashier adalah *Visual Studio Code*.
- c. Sistem manajemen *database* dalam aplikasi TMCashier menggunakan MySQL.

### 3.3.2. Perancangan Sistem

#### A. Use Case Diagram

Use Case Diagram merupakan konstruksi untuk menggambarkan secara sederhana fungsi-fungsi utama dari sistem dan berbagai user yang akan berinteraksi dengan sistem tersebut (Suryasari dkk., 2012). Berikut adalah use case diagram modul penjualan dimana terdapat sistem pembayaran dan pengurangan stok dapat dilihat pada Gambar 3.2.



Gambar 3.2 Use Case Diagram

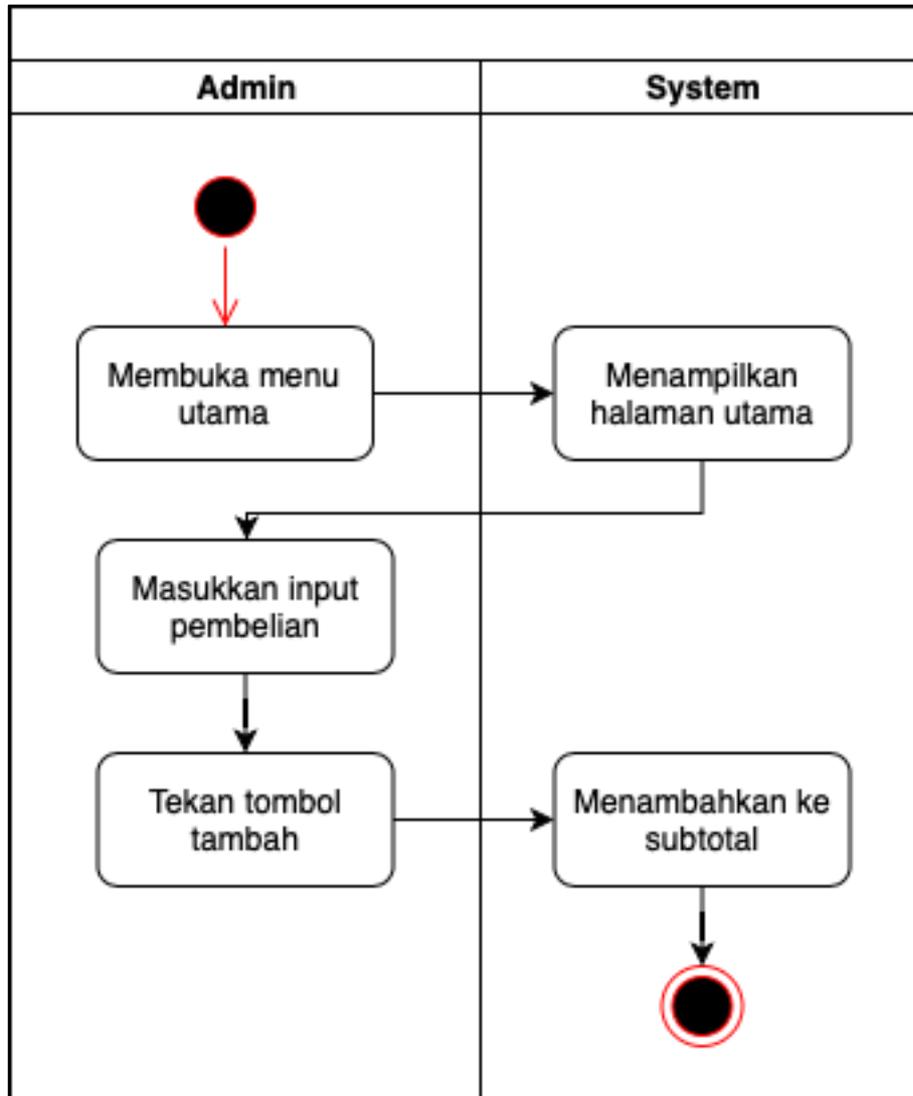
#### B. Activity Diagram

*Activity diagram* dibentuk berdasarkan *use case* diagram. Sistem ini memiliki 3 buah aktivitas yaitu *input dan view purchase*, *update stok dan print invoice*, dan *view history penjualan*.

### B.1. Activity Diagram Input dan View Purchase

Berikut adalah *activity diagram input* dan *view purchase* dapat dilihat pada Gambar

3.3.



Gambar 3.3 Activity Diagram Input dan View Purchase

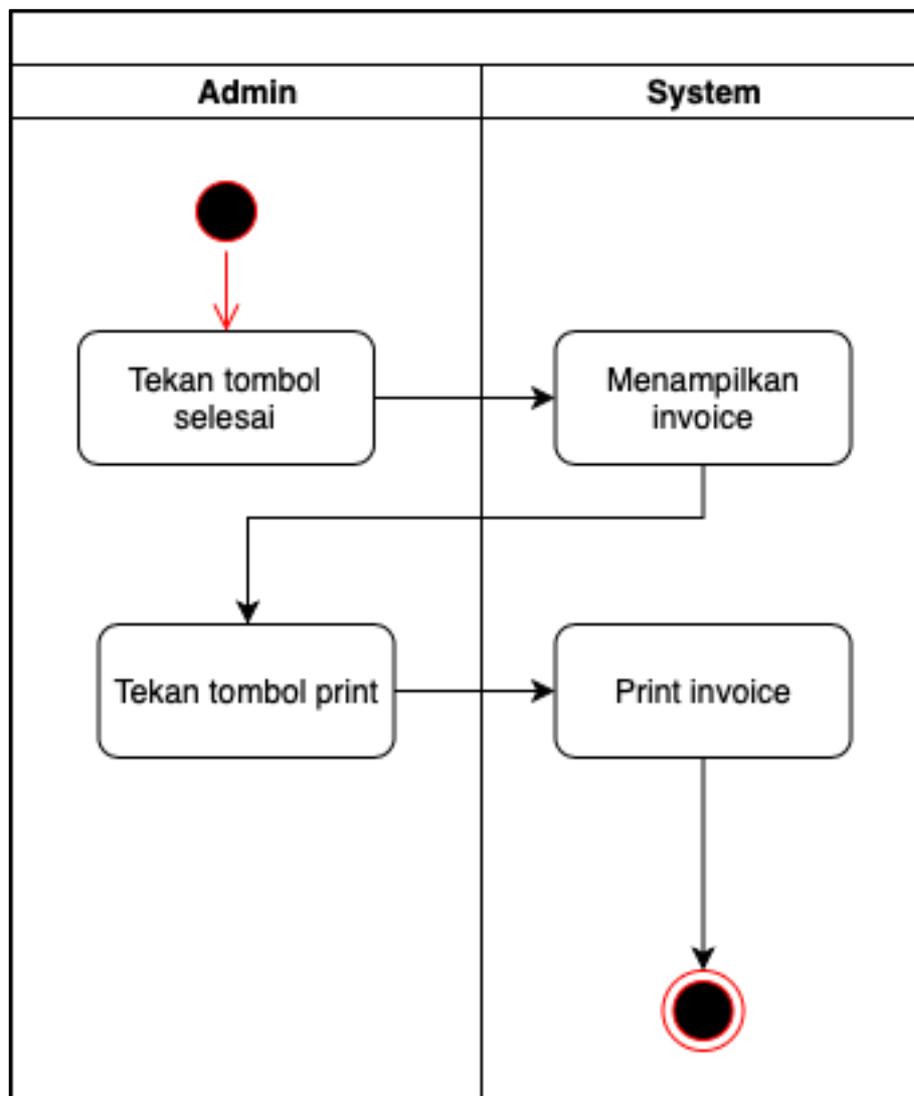
Pada Gambar 3.3 dijelaskan tentang *activity diagram input* dan *view purchase*, dimana administrator membuka menu utama dimana menampilkan tampilan utama. Pada halaman ini, administrator dapat memasukkan input pembelian sesuai dengan yang

diinginkan. Jika administrator menekan tombol tambah maka barang yang diinput akan ditambahkan ke *subtotal*.

## B.2. Activity Diagram Update Stok dan Print Invoice

Berikut adalah *activity diagram update* stok dan *print invoice* dapat dilihat pada

Gambar 3.4.



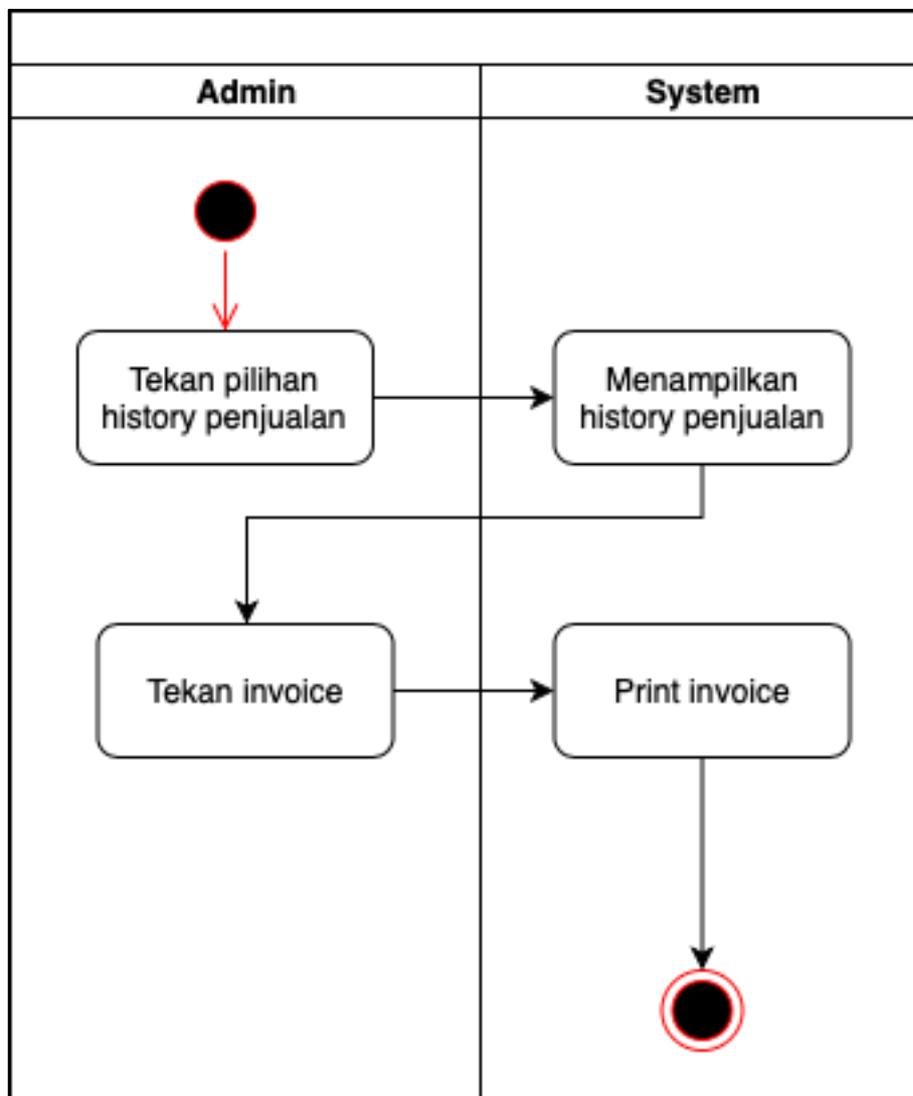
Gambar 3.4 Activity Diagram Update Stok dan Print Invoice

Pada Gambar 3.4 dijelaskan tentang *activity diagram update* stok dan *print invoice*, dimana administrator dapat menekan tombol selesai untuk menampilkan *invoice*, kemudian menekan tombol *print* untuk mencetak *invoice*.

### B.3. Activity Diagram View History Penjualan

Berikut adalah *activity diagram view history* penjualan dapat dilihat pada Gambar

3.5.



Gambar 3.5 Activity Diagram View History Penjualan

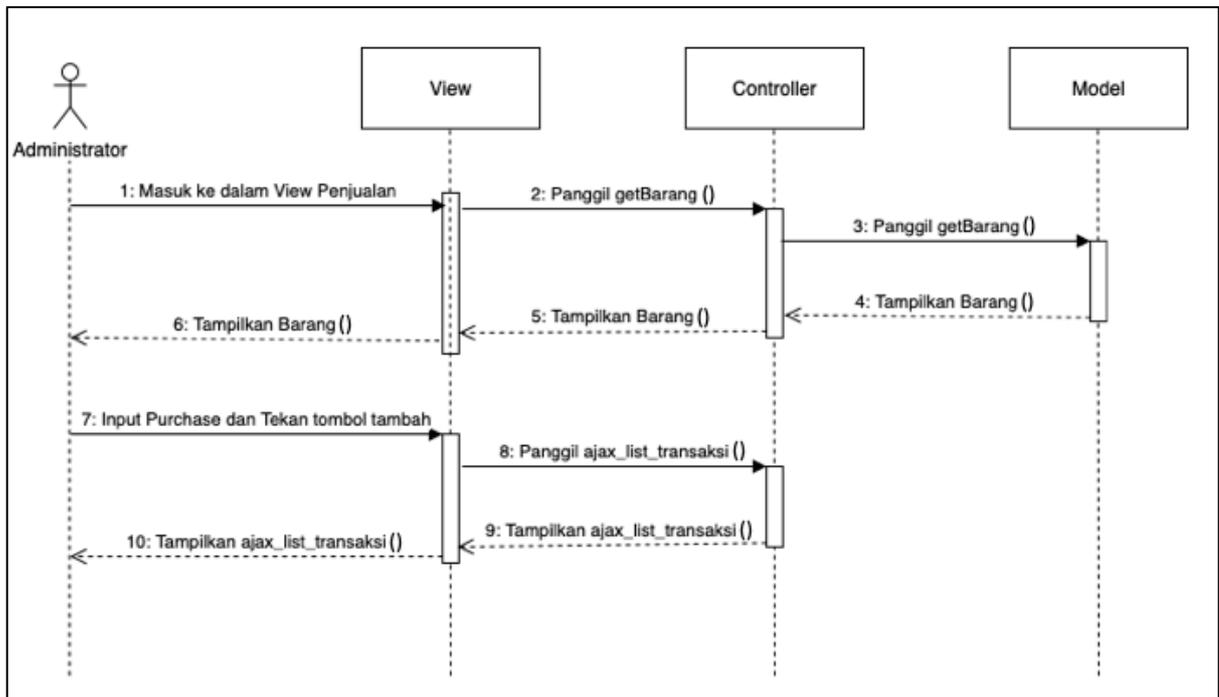
Pada Gambar 3.5 dijelaskan tentang *activity diagram history* penjualan, dimana administrator dapat menekan pilihan *history* penjualan untuk menampilkan history, kemudian menekan tombol *invoice* untuk menampilkan *invoice* dari *history* penjualan.

### C. Sequence Diagram

Tahap desain berikutnya adalah pembuatan *sequence diagram*. Jumlah *sequence diagram* yang dibuat sesuai dengan *activity diagram*.

#### C.1. Sequence Diagram Input dan View Purchase

Berikut adalah *sequence diagram input* dan *view purchase* dapat dilihat pada Gambar 3.6.



Gambar 3.6 Sequence Diagram Input dan View Purchase

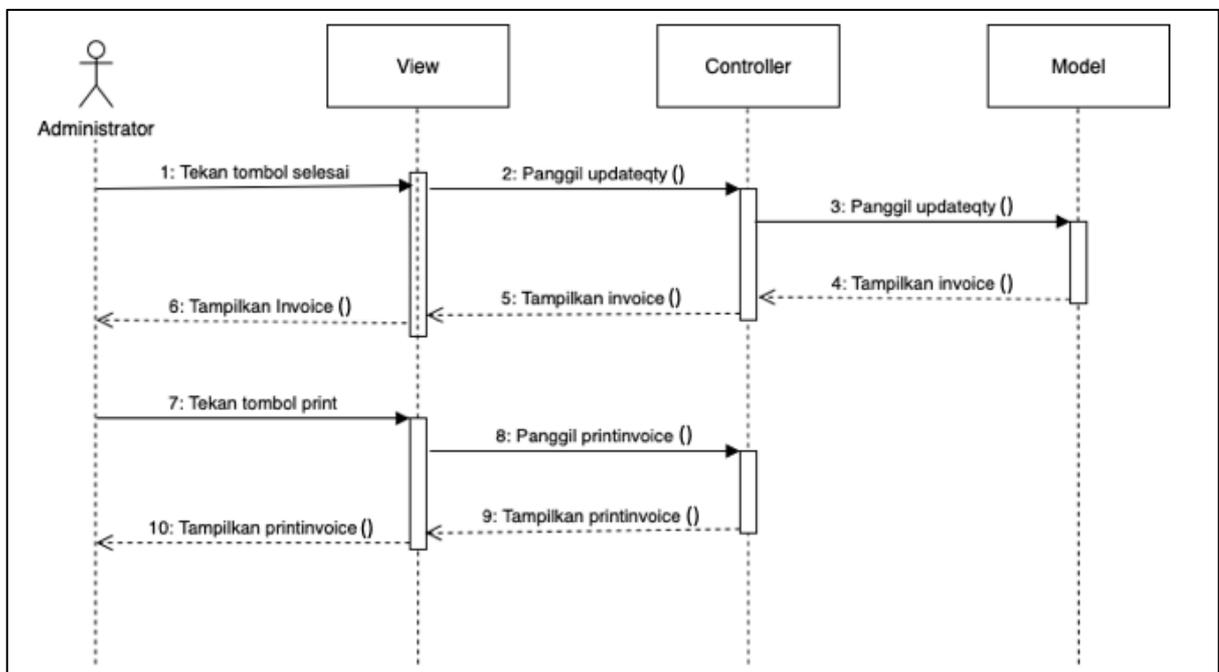
Pada Gambar 3.6 dijelaskan tentang *sequence diagram input* dan *view purchase*, dimana administrator dapat *input* dan *view purchase* pada modul penjualan.

Untuk melakukan *input* dan *view purchase*, admin dapat memasukkan data barang yang ingin dibeli kemudian menekan tombol tambah, sistem akan memanggil *list data* melalui *controller* dan menampilkan data barang yang dimasukkan.

## C.2. Sequence Diagram Update Stok dan Print Invoice

Berikut adalah *sequence diagram update* stok dan *print invoice* dapat dilihat pada

Gambar 3.7.



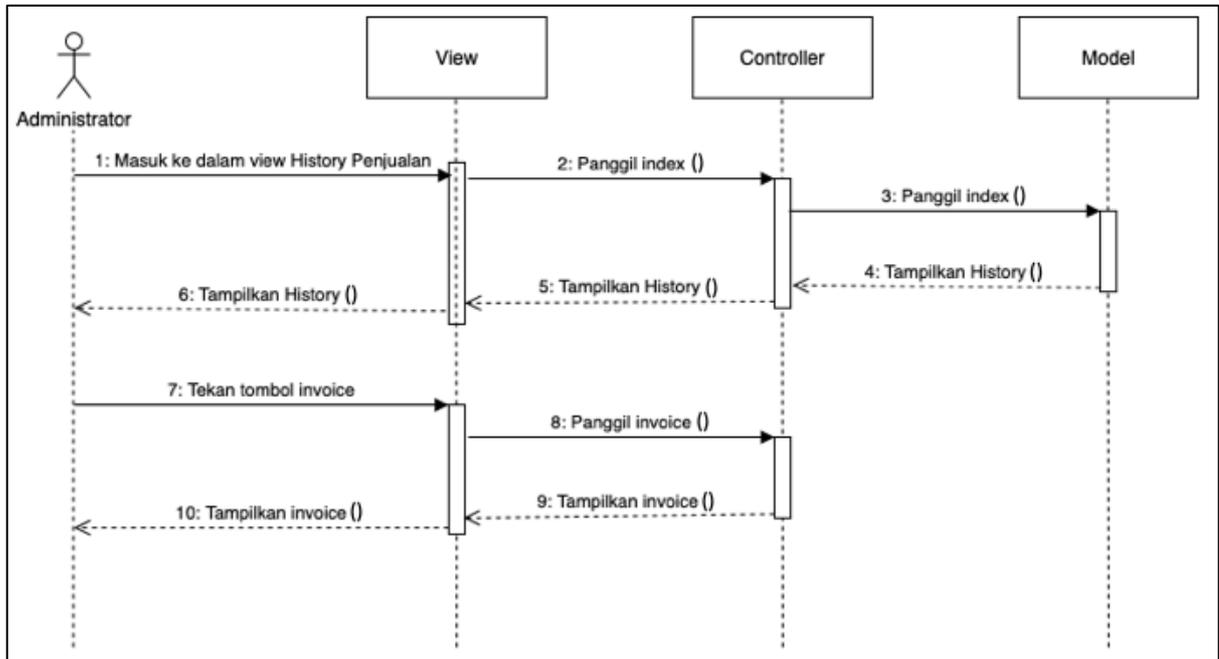
Gambar 3.7 Sequence Diagram Update Stok dan Print Invoice

Pada Gambar 3.7 dijelaskan tentang *sequence diagram update* stok dan *print invoice*, dimana administrator dapat *update* stok dan *print invoice* atas barang yang telah terjual. Untuk *update* stok, admin dapat menekan tombol selesai, kemudian stok pada *database* akan berkurang, kemudian dengan menekan tombol *print* akan mencetak *invoice* atas penjualan barang.

### C.3. Sequence Diagram View History Penjualan

Berikut adalah *sequence diagram view history* penjualan dapat dilihat pada

Gambar 3.8.



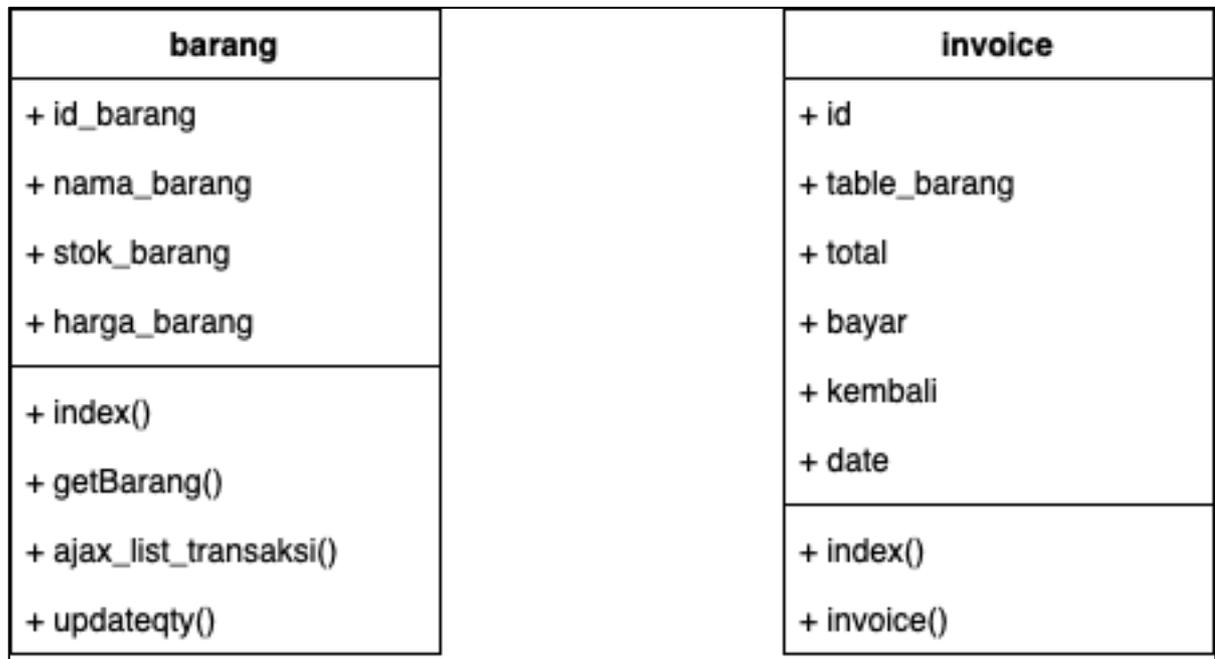
Gambar 3.8 Sequence Diagram View History Penjualan

Pada Gambar 3.8 dijelaskan tentang *sequence diagram view history* penjualan, dimana administrator dapat melihat kembali *history* penjualan dan *invoice* penjualan sebelumnya. Untuk melihat *history* penjualan, admin dapat menekan pilihan *history* penjualan, kemudian tekan tombol *invoice* untuk menampilkan *invoice* pada *history* penjualan.

### D. Class Diagram

*Class diagram* mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terjadi. Pada aplikasi TMCashier terdapat 2 kelas utama yang

dibangun dalam modul penjualan diantaranya barang dan *invoice*. Berikut *class diagram* aplikasi TMCashier dapat dilihat pada Gambar 3.9.

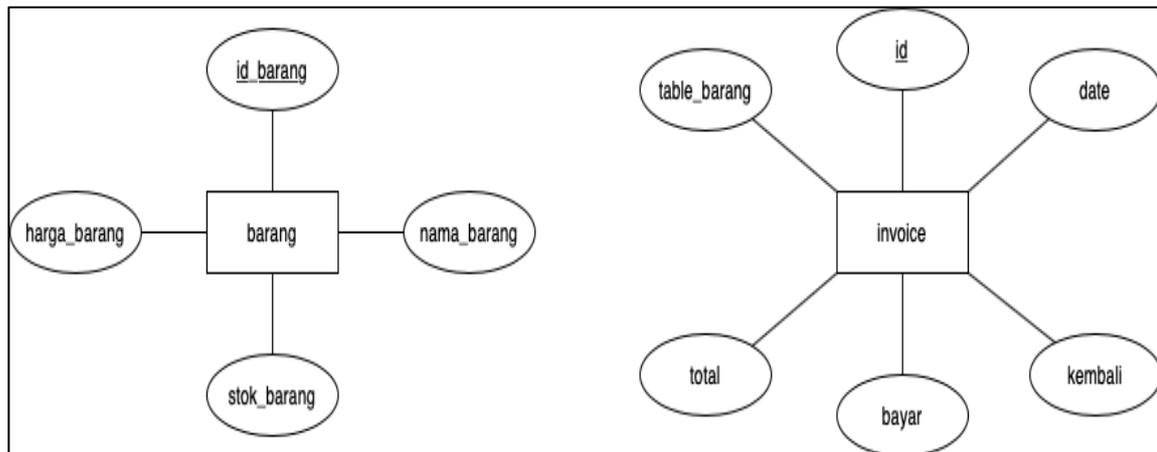


Gambar 3.9 Class Diagram

Pengerjaan modul dilakukan secara *object oriented* dengan menggunakan *class* barang. Dalam *class* tersebut terdapat fungsi yang dapat memasukkan data, mengambil data, dan mengubah data. Dalam memasukkan data penjualan, akan menggunakan fungsi `getBarang()`. *Class* barang akan menggunakan fungsi `ajax_list_transaksi()` untuk menampilkan barang yang akan diinput untuk penjualan. *Class* barang akan menggunakan fungsi `updateqty()` untuk mengubah data stok pada *database* untuk barang yang telah terjual.

## E. Entity Relationship Diagram

*Entity Relationship Diagram* adalah suatu diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem. Berikut adalah *entity relationship diagram* dimana hanya merepresentasikan nama entitas dan nama setiap atributnya.



Gambar 3.10 Entity Relationship Diagram

## F. Struktur Tabel

Penyimpanan data pada sebuah aplikasi membutuhkan database yang terdiri dari beberapa tabel. Berikut tabel dalam rancang bangun model penjualan:

### F.1. Tabel Barang

Nama : Barang

Fungsi : Menyimpan data barang dalam suatu perusahaan

Tabel 3.2 Struktur Tabel Barang

No.	Nama Field	Tipe Data	Panjang	Keterangan
1	id_barang	Varchar	10	Primary Key
2	nama_barang	Varchar	50	nama barang
3	stok_barang	Int	5	jumlah stok barang
4	harga_barang	Int	10	harga barang

## **F.2. Tabel Invoice**

Nama : Invoice

Fungsi : Menyimpan data invoice dalam suatu perusahaan

Tabel 3.3 Struktur Tabel Invoice

<b>No.</b>	<b>Nama Field</b>	<b>Tipe Data</b>	<b>Panjang</b>	<b>Keterangan</b>
1	id	Int	10	Primary Key
2	table_barang	Varchar	700	Tabel barang
3	total	Int	10	Total harga
4	bayar	Int	10	Total yang dibayar
5	kembali	Int	10	Total uang kembali
6	date	Varchar	30	Tanggal penjualan

### **3.3.3. Desain Antar Muka**

Setelah melakukan analisa kebutuhan, tahap selanjutnya dalam pengembangan aplikasi TMCashier adalah melakukan desain antar muka. Adapun desain antar muka dari aplikasi TMCashier adalah sebagai berikut:

#### **A. Desain Antar Muka Master Data Penjualan**

##### **A.1. Desain Antar Muka Penjualan Barang**

TMCashier

Penjualan ▲

Tambah Penjualan

History Penjualan

Barang ▼

ID Barang: 02

Nama Barang: Stadler

Harga (Rp): 5000

Quantity: 3

Sub Total (Rp): 15000

Tambah

Total (Rp): 15000

Bayar (Rp): 50000

Kembali (Rp): 35000

ID	Nama Brg	Harga(Rp)	Quantity
02	Stadler	5000	3

Selesai

Gambar 3.11 Desain Antar Muka Penjualan Barang

Halaman penjualan barang akan menampilkan halaman untuk *input* id barang, *quantity*, dan total yang dibayar. Halaman akan menampilkan nama barang, harga barang, subtotal, total harga, dan total kembali setelah semua input telah dimasukkan. Pengguna dapat menyelesaikan transaksi dengan menekan tombol selesai untuk melanjutkan ke halaman *print invoice*.

## A.2. Desain Antar Muka Print Invoice Penjualan

The screenshot shows the 'TMCashier' application window. On the left is a sidebar with a 'Penjualan' menu (expanded to show 'Tambah Penjualan' and 'History Penjualan') and a 'Barang' menu. The main area is titled 'Invoice' and contains a table with one row of data. Below the table, the summary shows a total of Rp 15000, a payment of Rp 50000, and a change of Rp 30000. A 'Print' button is located in the bottom right corner.

N	ID	Nama	Harga(R	Quanti	Sub
1	02	Stadler	5000	3	15000

Total (Rp) :  
**Rp 15000**

Bayar (Rp) :  
**Rp 50000**

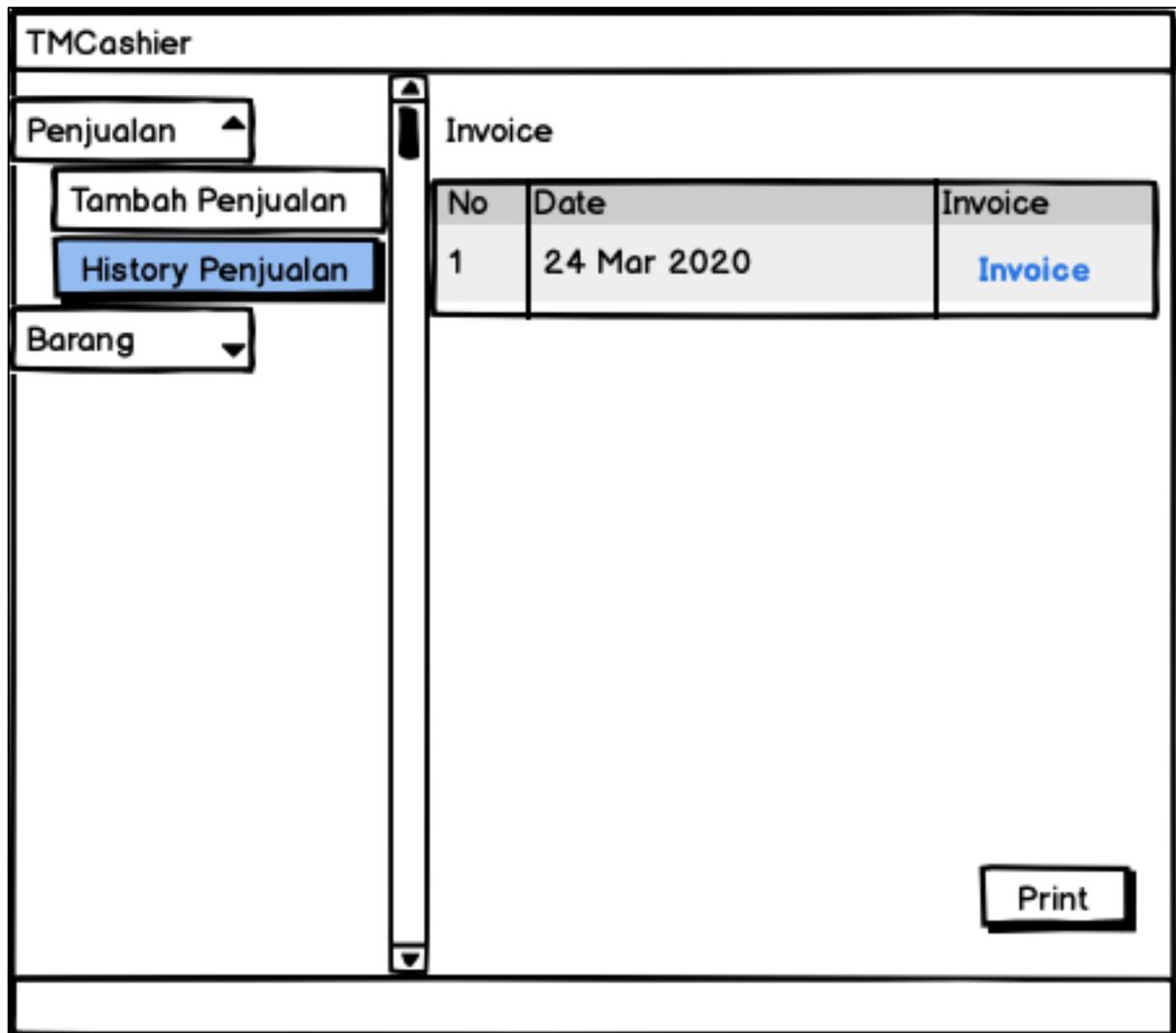
Kembali (Rp) :  
**Rp 30000**

**Print**

Gambar 3.12 Desain Antar Muka Print Invoice

Setelah pengguna menekan tombol selesai di halaman penjualan barang, pengguna akan masuk ke halaman *print invoice*. Pengguna dapat mencetak *invoice* dari penjualan dengan menekan tombol *print*.

### A.3. Desain Antar Muka History Penjualan



Gambar 3.13 Desain Antar Muka History Penjualan

Halaman *history* penjualan akan menampilkan *list data* dari *history* penjualan yang ada sebelumnya. Pengguna dapat melihat *invoice* untuk setiap *history* penjualan barang.

#### 3.3.4. Implementasi Sistem

Setelah melakukan desain antar muka, dilakukan implementasi sistem untuk TMCashier. Adapun tampilan dari implementasi sistem adalah sebagai berikut.

**TMCashier**

Penjualan

Tambah Penjualan

History Penjualan

Barang

Id Barang :

Nama Barang :

Harga (Rp) :

Quantity :  stok : 100

Sub-Total (Rp):

Tambah

Total (Rp) :

Bayar (Rp) :

Kembali (Rp) :

No	Id Barang	Nama Barang	Harga	Quantity	Sub-Total	Aksi
1	02	Stadler	5.000	2	10.000	

Selesai

Gambar 3.14 Halaman Penjualan Barang

Pada halaman penjualan barang, pengguna dapat memasukkan *input* ke *field* id barang dan *quantity* untuk menampilkan nama barang, harga barang, dan subtotal. Kemudian pengguna dapat menekan tombol tambah untuk memasukkan barang ke tabel *list* pembelian dan menampilkan total harga. Setelah pengguna memasukkan jumlah uang yang digunakan untuk membayar, halaman akan menampilkan total kembali. Pengguna dapat menekan tombol selesai untuk melanjutkan ke halaman *invoice*.

**Invoice**

Penjualan

Tambah Penjualan

History Penjualan

Barang

No.	Id Barang	Nama Barang	Harga	Quantity	Sub Total
1	02	Stadler	5.000	2	10.000

Total (Rp) :  
**Rp 10.000**

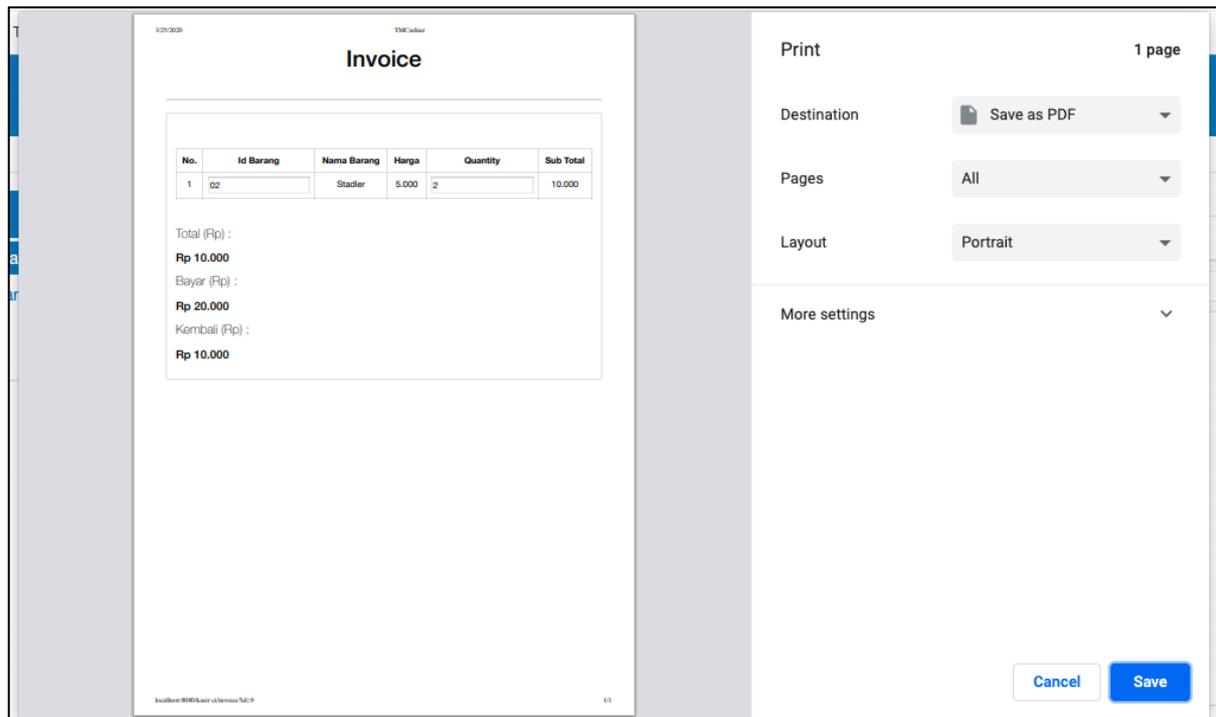
Bayar (Rp) :  
**Rp 20.000**

Kembali (Rp) :  
**Rp 10.000**

Print

Gambar 3.15 Halaman Show Invoice

Setelah menekan tombol selesai pada halaman penjualan, pengguna akan masuk ke halaman *invoice*. Halaman *invoice* akan menampilkan *invoice* barang yang akan dibeli beserta dengan total harga, total yang dibayar, dan total kembali. Pengguna dapat menekan tombol *print* untuk masuk ke halaman *print preview*.



Gambar 3.16 Halaman Print Preview

Setelah menekan tombol *print* pada halaman *invoice*, pengguna akan masuk ke halaman *print preview* untuk menampilkan tampilan yang akan dicetak sebelum *invoice* dicetak.

History Penjualan		
No	Date	Invoice
1	Tuesday, 24 Mar 2020	<a href="#">Invoice</a>
2	Tuesday, 24 Mar 2020	<a href="#">Invoice</a>
3	Tuesday, 24 Mar 2020	<a href="#">Invoice</a>
4	Tuesday, 24 Mar 2020	<a href="#">Invoice</a>
5	Tuesday, 24 Mar 2020	<a href="#">Invoice</a>
6	Tuesday, 24 Mar 2020	<a href="#">Invoice</a>
7	Wednesday, 25 Mar 2020	<a href="#">Invoice</a>
8	Wednesday, 25 Mar 2020	<a href="#">Invoice</a>
9	Wednesday, 25 Mar 2020	<a href="#">Invoice</a>

Gambar 3.17 Halaman History Penjualan

Halaman *history* penjualan menampilkan *list data history* penjualan berupa tanggal dan link untuk membuka *invoice* dari *history* penjualan. Pengguna dapat menekan tombol *invoice* untuk membuka *invoice* untuk setiap transaksi penjualan.

### 3.4. Kendala yang Ditemukan

Beberapa kendala yang ditemukan selama melaksanakan kerja magang di CV Trijaya Mandiri adalah sebagai berikut:

1. Terjadi perombakan sistem dalam setiap *weekly meeting* yang menghambat terselesaikannya proyek ini sesuai dengan *deadline* yang ditentukan.
2. Kesulitan untuk berkomunikasi dengan tim karena kebijakan WFH atau *Work From Home*.

### 3.5. Solusi Atas Kendala yang Ditemukan

Adapun solusi atas kendala yang ditemukan selama pelaksanaan kerja magang di CV Trijaya Mandiri yaitu:

1. Melakukan *backup* data sebelum melakukan perombakan.
2. Meningkatkan intensitas komunikasi dengan tim lewat pesan singkat atau *video call* menggunakan media sosial.

### **3.6. Hasil dan Pengujian**

Aplikasi ini diharapkan dapat membantu CV Trijaya Mandiri dalam sistem kasir yang dapat memudahkan proses penjualan dalam perusahaan. Aplikasi ini berisi mengenai memilih barang, menampilkan harga barang, memasukkan jumlah pembayaran, menampilkan jumlah kembalian, menampilkan *invoice*, mencetak *invoice*, dan mengurangi stok barang dalam *database*. Pengujian dilakukan dengan cara melakukan *testing* dan *debugging* pada setiap modul. Hasil dari pengujian pada setiap modul akan dilaporkan saat *weekly meeting* kepada bapak Adrian Muradinata selaku *supervisor*.