

BAB IV

DESKRIPSI PEKERJAAN

Dalam merancang, membangun dan mengimplementasikan basis data yang dibuat dengan aplikasi sistem informasi pergudangan menjadi satu kesatuan yang utuh, maka terdapat pembagian tugas dalam pembuatan aplikasi tersebut. Penulis dalam hal ini memperoleh bagian dalam merancang basis data sistem informasi pergudangan beserta meng-integrasikan data – data tersebut kedalam grafik – grafik laporan penjualan selama 7 hari, 7 bulan, dan 7 tahun.

Dalam merancang basis data sistem informasi pergudangan pastinya diperlukan adanya pengumpulan data – data dan juga informasi yang akan digunakan sebagai pedoman dalam merancang basis data agar sesuai dengan proses bisnis perusahaan. Untuk memperoleh data – data dan informasi tersebut dilakukan beberapa metode antara lain sebagai berikut :

1. Wawancara

Proses wawancara yang dilakukan sama seperti pada umumnya yaitu dengan melakukan tanya jawab kepada Manajer Teknologi Informasi atau Direktur Utama perusahaan guna memperoleh data dan juga informasi yang dibutuhkan dalam merancang basis data sesuai dengan proses bisnis perusahaan.

2. Observasi

Melakukan pengamatan proses alur transaksi masuk dan transaksi keluar yang terjadi di perusahaan. Setelah melakukan pengamatan terhadap transaksi atau alur proses bisnis perusahaan, maka selanjutnya dilakukan

analisis terhadap transaksi masuk dan juga transaksi keluar barang tersebut. Dari hasil analisis yang telah diperoleh, selanjutnya dibuatlah basis data sesuai dengan hasil pengamatan yang terjadi di lapangan.

3. Studi Pustaka

Mencari referensi dari berbagai macam sumber terkait informasi – informasi apa saja yang diperlukan dalam merancang basis data perusahaan. Informasi – informasi tersebut selanjutnya digunakan sebagai pedoman dalam membuat aplikasi sistem informasi pergudangan.

Setelah dilakukan pengamatan terhadap proses bisnis yang berjalan pada PT. Eyro Digital Teknologi, ditemukan beberapa permasalahan yang terjadi yaitu terdapat beberapa resiko yang dialami oleh PT. Eyro Digital Teknologi seperti hilangnya data – data transaksi perusahaan dikarenakan data – data transaksi tersebut hanya dicatat menggunakan sebuah buku saja yang akan sangat beresiko sekali apabila buku penjualan tersebut hilang.

Selain itu permasalahan lain yang timbul pada PT. Eyro Digital Teknologi yaitu pihak Direktur Utama mengalami kesulitan dalam melihat laporan penjualan produk setiap 7 hari, 7 bulan, dan 7 tahun sekali. Sehingga guna mengatasi permasalahan tersebut dilakukan beberapa langkah sebagai berikut :

1. Melakukan analisa sistem penjualan dan sistem pembelian bahan baku yang ada dalam perusahaan.
2. Menyusun *procedure* yang mampu menampilkan laporan penjualan selama 7 hari, 7 bulan, dan juga 7 tahun.
3. Merancang basis data yang mampu menyimpan data sesuai dengan analisa proses bisnis perusahaan dan kebutuhan perusahaan.

4.1. Analisa Sistem

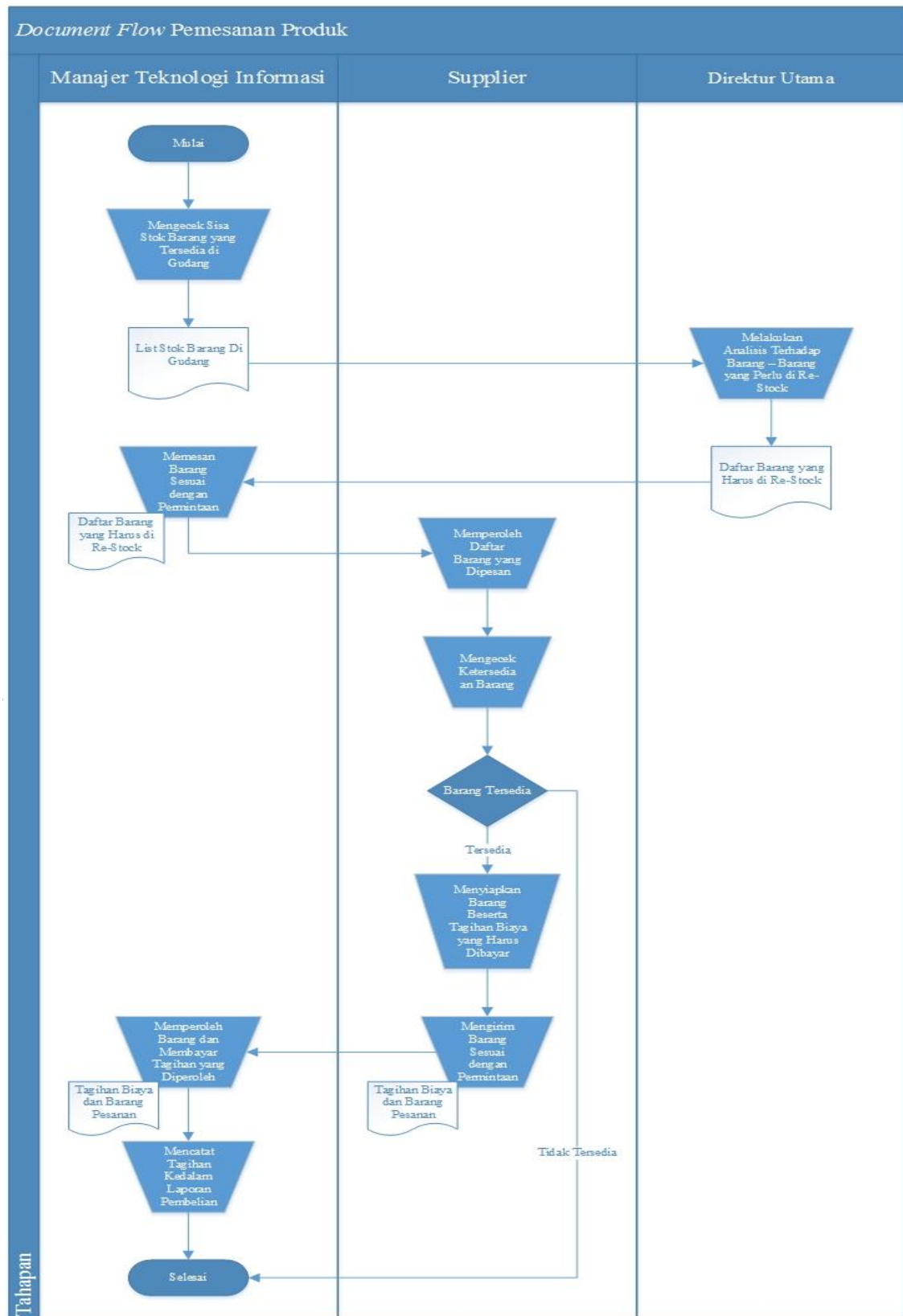
Dalam merancang sebuah perangkat lunak diperlukan adanya analisa sistem terhadap proses bisnis yang dijalankan oleh perusahaan. Dalam hal ini analisa sistem yang hendak dibuat merupakan langkah awal dalam merancang suksesnya sebuah perangkat lunak. Langkah – langkah tersebut berupa analisis terhadap penyimpanan transaksi masuk dan transaksi keluar perusahaan beserta laporan penjualan sesuai dengan permintaan Direktur Utama. Berdasarkan survei lapangan diperoleh celah permasalahan yaitu pihak Direktur Utama kesulitan dalam melihat laporan penjualan setiap 7 hari, 7 bulan, dan 7 tahun sekali.

Permasalahan lain yang dialami perusahaan adalah penyimpanan data – data perusahaan masih dilakukan secara *manual*, sehingga akan beresiko sekali apabila terjadi bencana, pencurian data, dan lain sebagainya. Oleh karena itu untuk mengatasi beberapa permasalahan tersebut, dibuatlah perancangan basis data sesuai dengan analisis proses bisnis perusahaan guna membantu Direktur Utama dalam melakukan pengambilan keputusan beserta untuk mengetahui jumlah stok yg tersedia.

4.1.1. *Document Flow* pada PT. Eyro Digital Teknologi

Pada *Document Flow* terdapat proses atau alur jalannya beberapa proses bisnis perusahaan yaitu proses pemesanan produk dan proses penjualan produk. *Document Flow* memiliki proses bisnis yang dijalankan secara *manual*. Terdapat 4 aktor yang terlibat dalam proses bisnis PT. Eyro Digital Teknologi yaitu Manajer Teknologi Informasi, Direktur Utama, *Supplier*, *Customer*, *Mechatronics*, *Content and Admin*. Berikut penjelasan tentang kedua proses bisnis perusahaan :

1. Document Flow Pemesanan Produk



Gambar 4.1 Document Flow Pemesanan Produk

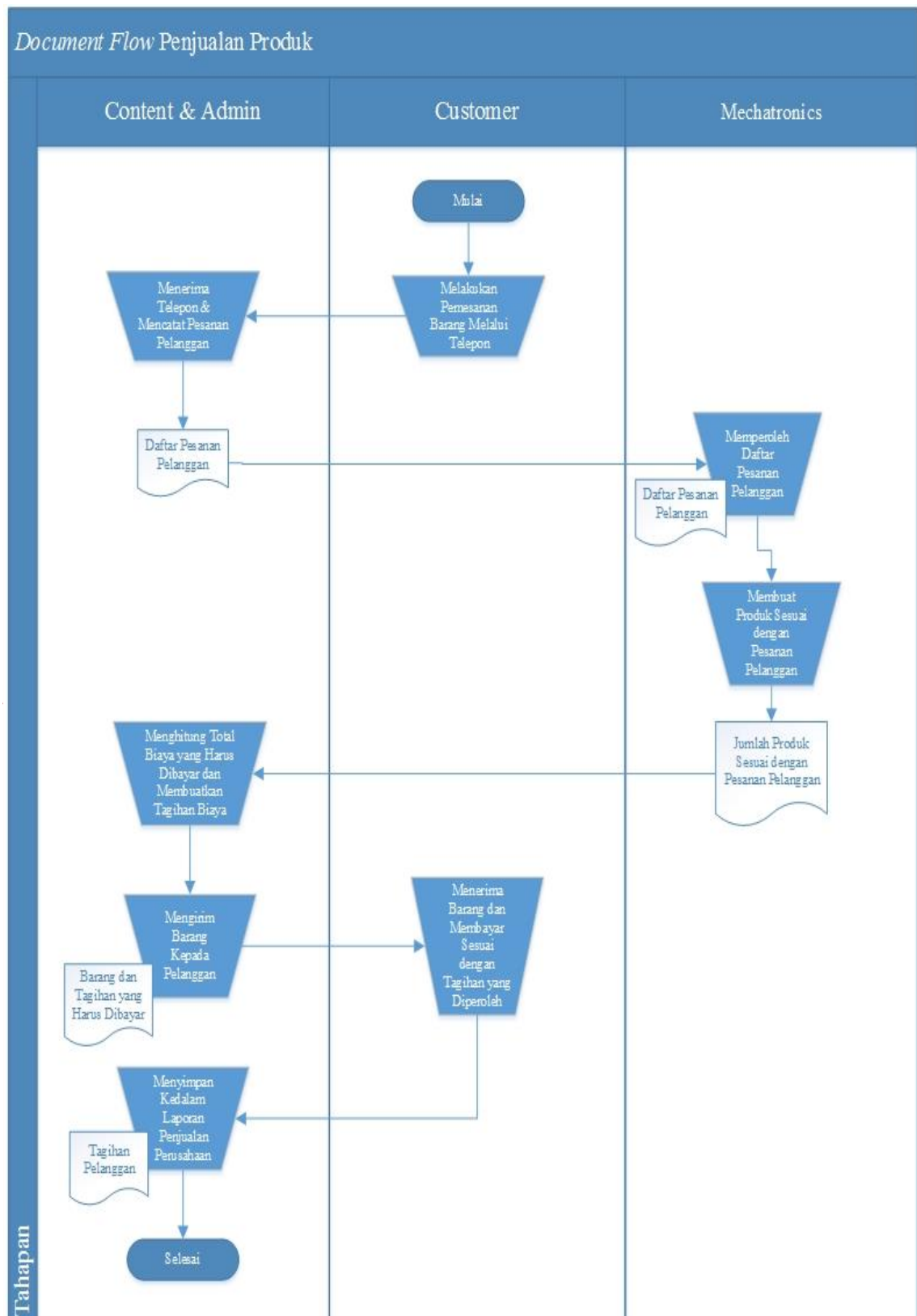
Pada Gambar 4.1 menjelaskan bahwa Manajer Teknologi Informasi melakukan pengecekan terlebih dahulu terhadap stok barang yang tersedia di gudang. Setelah menghitung seluruh stok barang yang tersedia didalam gudang, maka Manajer Teknologi Informasi memberikan laporan persediaan barang kepada Direktur Utama.

Direktur Utama melakukan analisis terhadap sisa stok barang yang tersedia didalam gudang. Setelah melakukan analisis terhadap sisa stok barang yang tersedia didalam gudang, maka selanjutnya Direktur Utama meminta Manajer Teknologi Informasi untuk melakukan *re-stock* dengan memberikan daftar barang – barang yang perlu dilakukan *re-stock*.

Manajer Teknologi Informasi akan melakukan pemesanan barang kepada *Supplier* dan *Supplier* akan mengecek apakah barang yang dipesan oleh PT. Eyro Digital Teknologi masih tersedia atau sudah habis. Jika barang masih tersedia, maka *Supplier* akan menyiapkan barang – barang tersebut beserta tagihan yang harus dibayar oleh PT. Eyro Digital Teknologi.

Setelah barang – barang sudah disiapkan oleh *Supplier*, maka barang – barang pesanan tersebut akan dikirim ke PT. Eyro Digital Teknologi. Setelah Manajer Teknologi Informasi menerima seluruh pesanan barang, maka beliau akan membayar sesuai dengan harga yang telah diberikan. Kemudian struk tagihan tersebut akan dimasukkan kedalam laporan pembelian perusahaan yang selanjutnya akan disimpan kedalam sebuah rak buku laporan pembelian perusahaan.

2. Document Flow Penjualan Produk



Gambar 4.2 Document Flow Penjualan Produk

Pada Gambar 4.2 menjelaskan bahwa untuk penjualan produk dimulai dari pelanggan yang melakukan pemesanan dengan cara menelepon perusahaan dan menyebutkan produk apa saja yang hendak dipesan. Bagian *admin* yang menerima telepon dari pelanggan tersebut selanjutnya mencatat pesanan pelanggan dan meminta bagian *mechatronics* untuk membuat produk sesuai dengan pesanan pelanggan tersebut.

Setelah bagian *mechatronics* menyelesaikan pesanan pelanggan tersebut, maka *mechatronics* akan memberikan produk – produk tersebut kepada *admin* untuk dicek kembali apakah barang sudah sesuai dengan permintaan pelanggan. Setelah itu *admin* akan menghitung total biaya yang harus dibayar oleh pelanggan tersebut dan membuat tagihan kepada pelanggan tersebut.

Tagihan beserta barang akan dikirim oleh *admin* serta mencatat kedalam laporan penjualan terhadap transaksi tersebut. Selanjutnya pelanggan akan menerima barang pesanan dan membayar sesuai dengan tagihan yang diberikan oleh perusahaan.

4.2. Perancangan Sistem

Setelah *Document Flow* telah dibuat, maka selanjutnya akan dilakukan perancangan sistem guna memperbaiki proses *manual* yang terjadi dalam perusahaan. Perancangan Sistem yang dibuat berupa diagram – diagram alurnya sistem informasi pergudangan dan hal ini diperlukan dengan tujuan aplikasi yang dibuat tetap sesuai dengan proses bisnis perusahaan dan sesuai dengan

permintaan Direktur Utama. Dalam merancang sebuah sistem nantinya terdapat beberapa pemodelan yang akan digunakan yaitu *System Flow*, *Hierarchy Input Process Output (HIPO)*, *Context Diagram*, *Data Flow Diagram*, dan *Entitiy Relationship Diagram*.

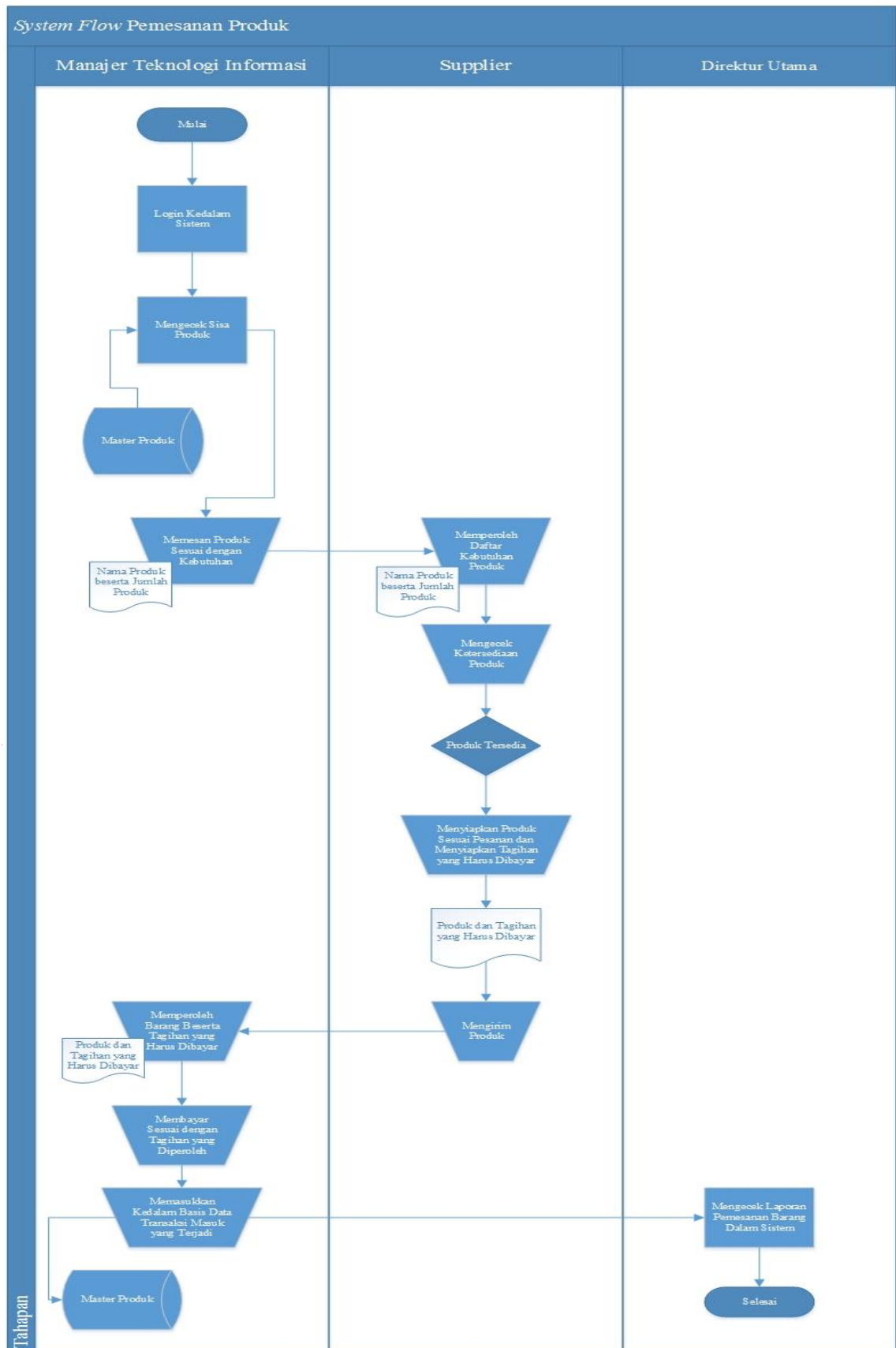
4.2.1. *System Flow*

System Flow memuat tentang perbaikan dari *Document Flow* yang sebelumnya beberapa proses masih dilakukan secara *manual*, kemudian diubah menjadi berbasis teknologi informasi. Pembuatan *System Flow* harus sesuai dengan proses bisnis perusahaan dan juga sesuai dengan permintaan Direktur Utama.

Dalam PT. Eyro Digital Teknologi terdapat dua *System Flow* yang akan digunakan yaitu *System Flow* Pemesanan Barang dan *System Flow* Penjualan Barang. Berikut penjelasan lebih detil tentang *System Flow* yang dibuat untuk memperbaiki proses bisnis yang terjadi pada PT. Eyro Digital Teknologi :

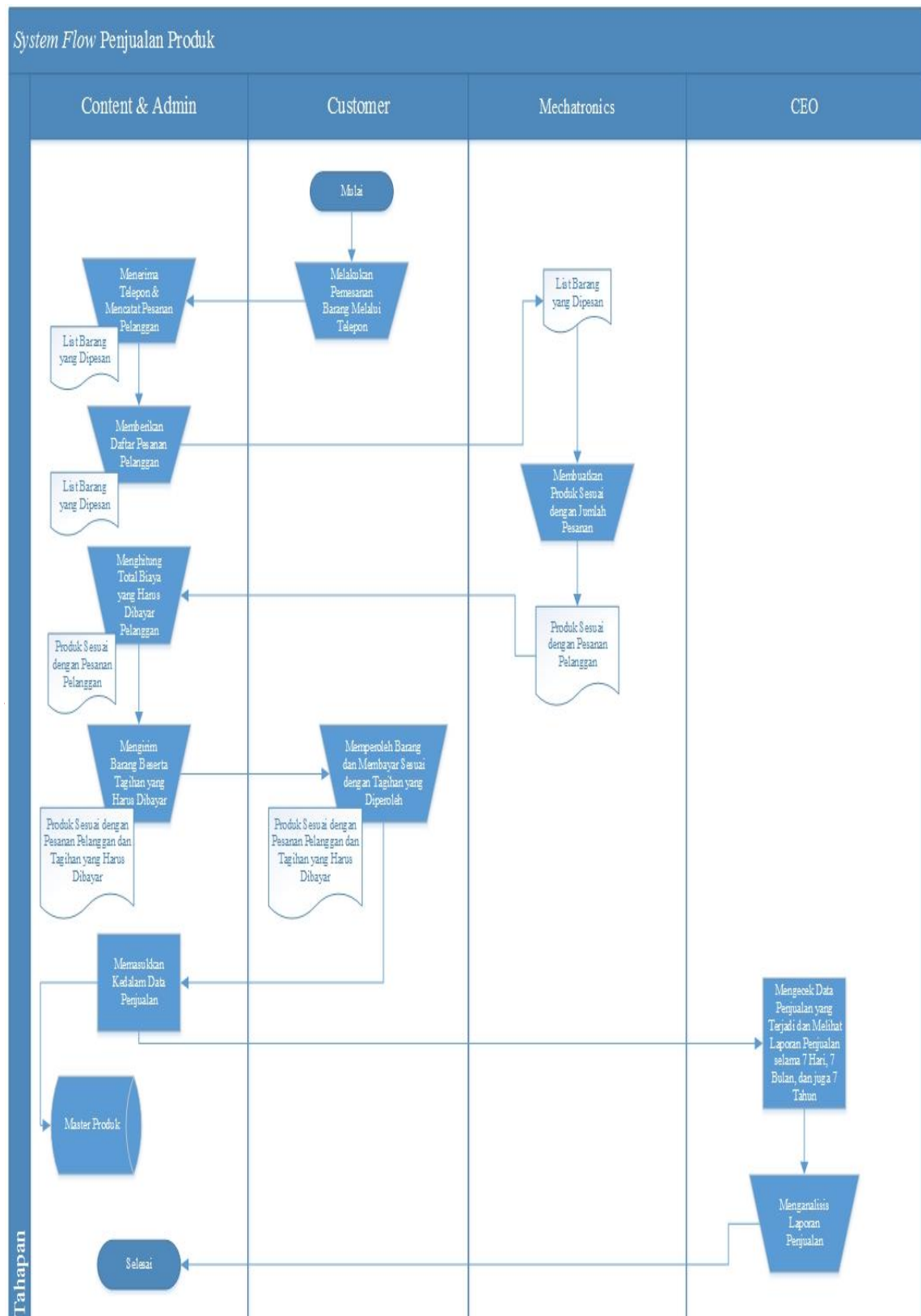
1. *System Flow* Pemesanan Produk

Dalam *System Flow* Pemesanan Produk, Manajer Teknologi Informasi melakukan *login* terlebih dahulu kedalam sistem guna mengetahui stok barang yang tersedia didalam gudang. Selanjutnya Manajer Teknologi Informasi melakukan pemesanan barang kepada *supplier* sesuai dengan kebutuhan. Selanjutnya *supplier* akan mengecek apakah barang yang dipesan tersedia didalam gudang. Jika barang yang dipesan tersedia didalam gudang, maka selanjutnya *supplier* akan menyiapkan barang – barang pesanan tersebut dan membuat tagihan biaya yang harus dibayar. Selanjutnya, Manajer Teknologi Informasi menerima barang pesanan beserta tagihan yang harus dibayar.



Gambar 4.3 System Flow Pemesanan Produk

2. System Flow Penjualan Produk

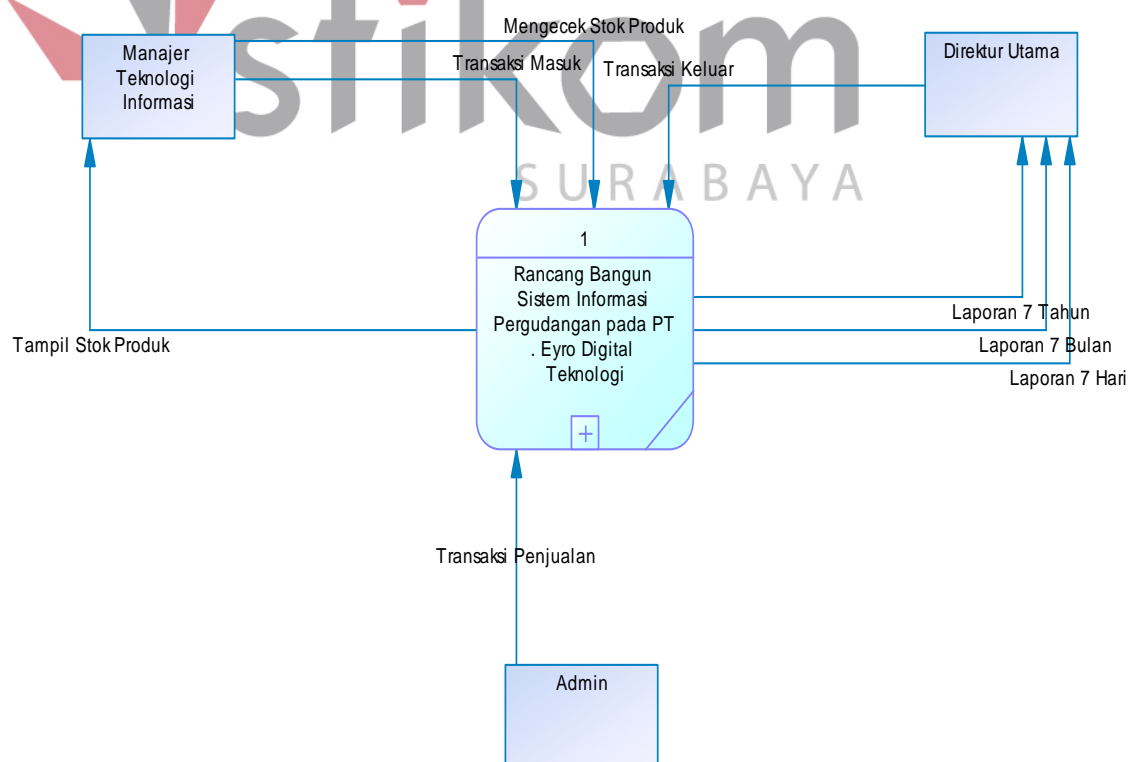


Gambar 4.4 System Flow Penjualan Produk

Pada Gambar 4.4 menjelaskan bahwa bagian *admin* menerima pesanan dari pelanggan melalui telepon dan pesanan tersebut selanjutnya akan diberikan kepada *mechatronics* selaku anggota yang bertugas dalam memproduksi barang sesuai dengan pesanan pelanggan.

Setelah produk telah dibuat sesuai dengan permintaan pelanggan maka produk – produk tersebut akan diberikan kepada *admin* untuk membuat tagihan sesuai dengan jumlah barang yang dipesan oleh pelanggan. Selanjutnya barang akan dikirim kepada pelanggan beserta tagihan yang harus dibayar oleh pelanggan tersebut. Saat barang telah diterima pelanggan, maka pelanggan membayar sesuai dengan biaya tagihan yang diperoleh. Kemudian *admin* akan memasukkan data penjualan tersebut kedalam basis data.

4.2.2. Context Diagram



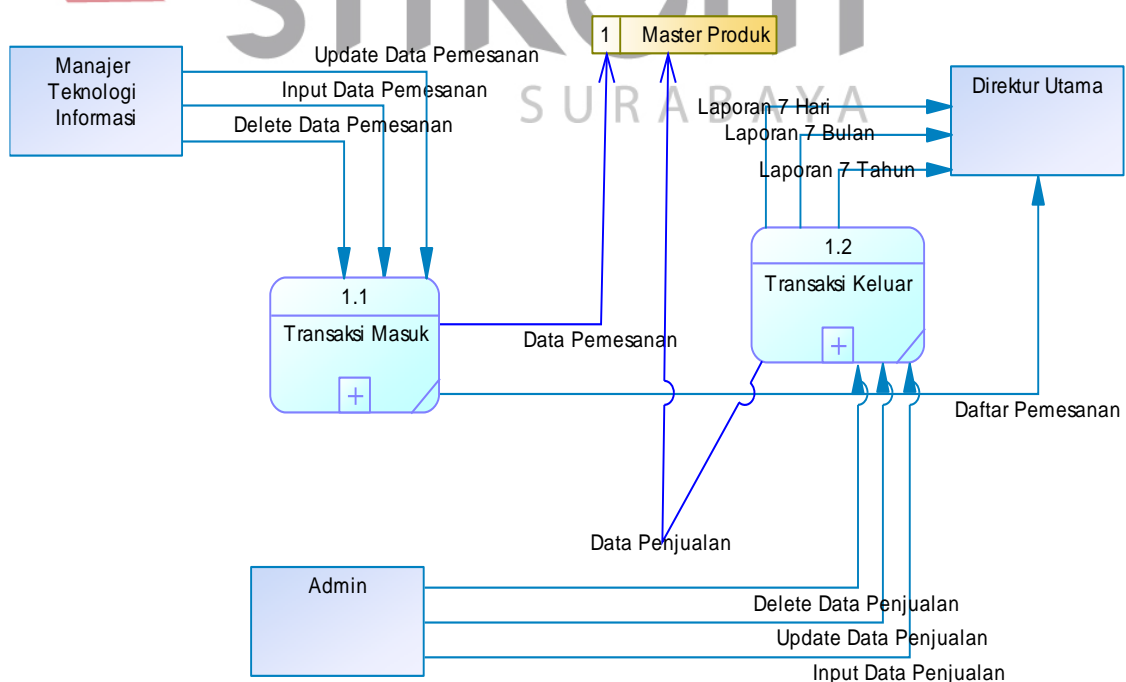
Gambar 4.5 Context Diagram

Berdasarkan Gambar 4.5 menjelaskan bahwa simbol kotak merupakan anggota atau seseorang yang berperan sebagai *Admin*, Manajer Teknologi Informasi, dan juga Direktur Utama. Dalam aplikasi sistem informasi pergudangan, *admin* bertugas untuk melakukan proses bisnis penjualan perusahaan. Manajer Teknologi Informasi bertugas untuk melakukan proses bisnis pemesanan produk dan Direktur Utama bertugas untuk mengecek laporan penjualan yang terjadi setiap 7 hari, 7 bulan, dan juga 7 tahun.

4.2.3. Data Flow Diagram

Data Flow Diagram atau DFD merupakan bagan yang menjelaskan lebih detail tentang arus data dalam suatu sistem yang akan dirancang atau dibangun. *Data Flow Diagram* juga merupakan uraian dari *context diagram* secara lebih detail lagi terhadap arus jalannya program yang akan dibuat. DFD menggambarkan seluruh kegiatan yang ada pada sistem dengan jelas dan juga terperinci.

A. Data Flow Diagram Level 0

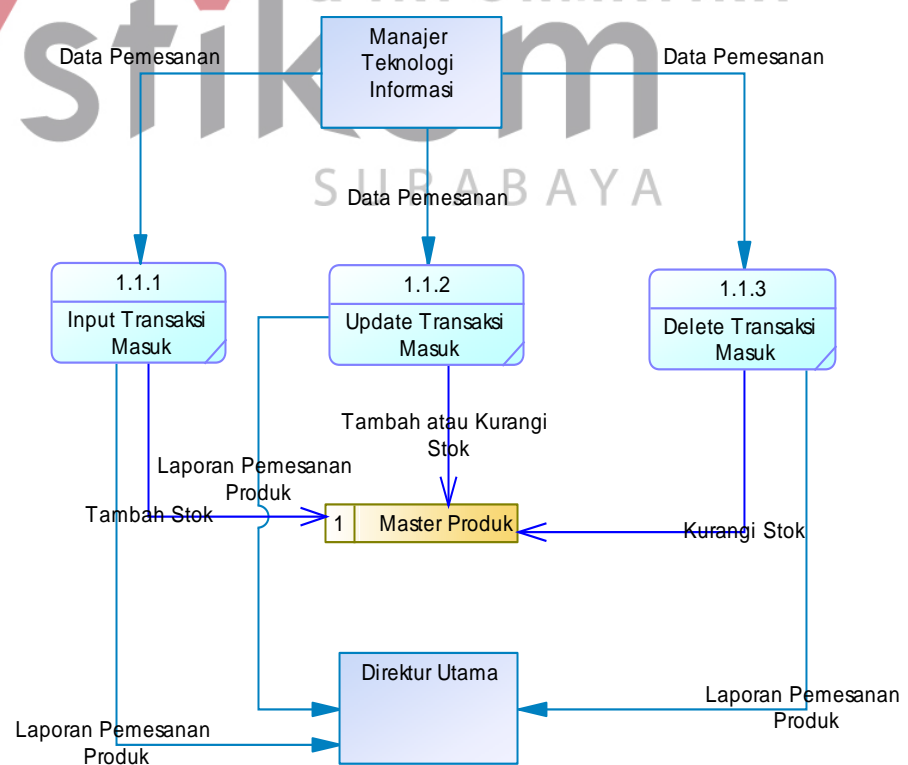


Gambar 4.6 Data Flow Diagram Level 0

Data Flow Diagram Level 0 merupakan gambaran tentang fungsi – fungsi utama yang terdapat didalam aplikasi sistem informasi perdagangan seperti yang terlihat pada Gambar 4.6 yang memiliki 2 fungsi utama dalam aplikasi yaitu transaksi pemesanan produk dan transaksi penjualan produk. Selanjutnya kedua fungsi – fungsi tersebut akan dipecah kembali menjadi *Data Flow Diagram Level 1* untuk menjelaskan lebih detail tentang fitur – fitur apa sajakah yang terdapat didalam 2 proses utama tersebut. Dari gambar diatas terlihat bahwa laporan penjualan yang dihasilkan terdapat 3 jenis laporan penjualan yang meliputi laporan penjualan 7 hari, laporan penjualan 7 bulan, dan juga yang terakhir adalah laporan penjualan selama 7 tahun.

B. *Data Flow Diagram Level 1*

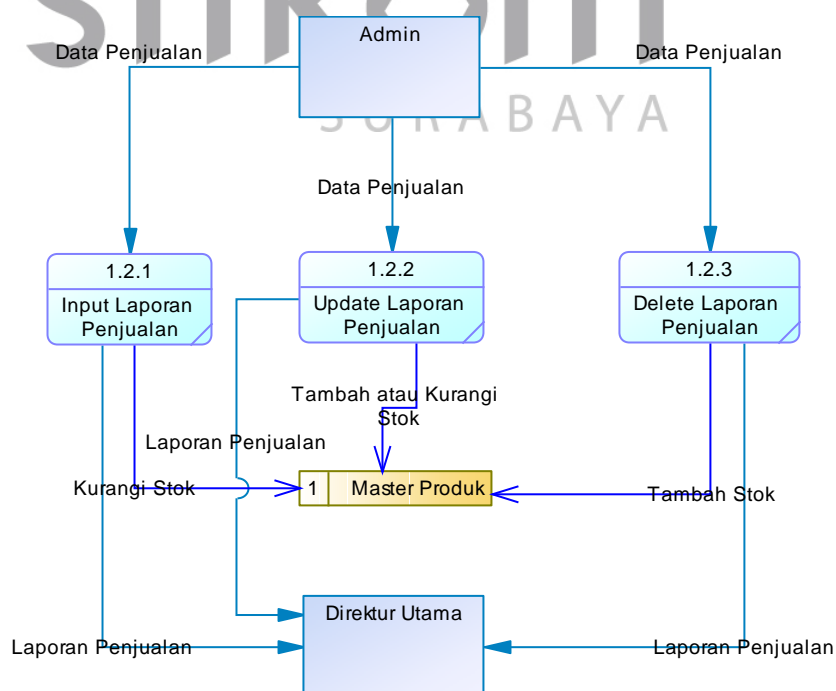
1. *Data Flow Diagram Pemesanan Barang*



Gambar 4.7 Data Flow Diagram Pemesanan Barang

Dalam DFD Pemesanan Barang terdapat tiga proses yang dapat dilakukan oleh Manajer Teknologi Informasi yaitu memasukkan data pemesanan barang, menghapus data pemesanan barang, dan yang terakhir adalah memperbaharui data pemesanan barang yang sudah ada sesuai dengan keinginan dari Manajer Teknologi Informasi. Melalui Gambar 4.7 terlihat bahwa seluruh proses tersebut akan otomatis meng-*update* stok barang yang tersedia digudang. Salah satu contohnya seperti saat pengguna memasukkan jumlah pemesanan barang yang baru, maka jumlah stok barang yang tersedia di gudang otomatis akan ditambah. Begitu pula dengan proses menghapus data transaksi pemesanan barang. Maka stok barang yang sebelumnya telah ditambah akan dikurangi kembali sesuai dengan jumlah yang ditambahkan sebelumnya. Sehingga stok barang yang tersedia sama seperti jumlah stok sebelumnya.

2. Data Flow Diagram Penjualan Barang



Gambar 4.8 Data Flow Diagram Penjualan Barang

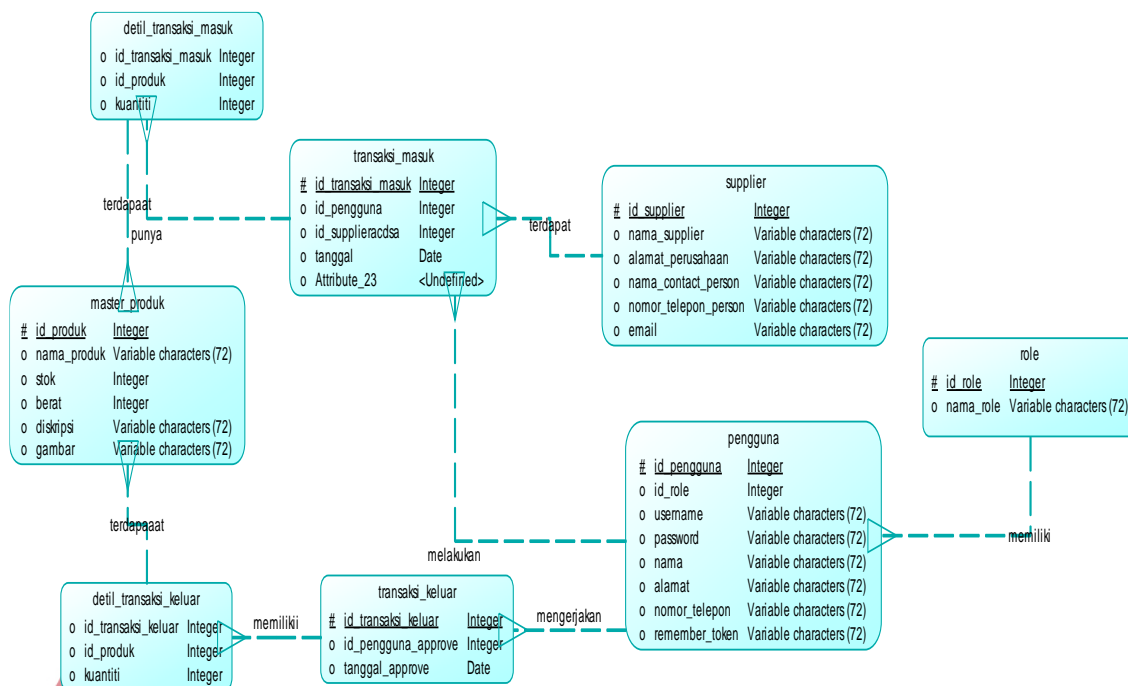
Berdasarkan Gambar 4.8 dapat dilihat bahwa terdapat 3 proses utama dalam melakukan penjualan barang yaitu memasukkan data transaksi penjualan, memperbaharui data transaksi penjualan, dan juga menghapus data transaksi penjualan.

Hampir sama halnya dengan proses pemesanan barang, dalam *Data Flow* penjualan barang juga terhubung secara langsung dengan master produk yang artinya setiap proses yang terjadi seperti memasukkan transaksi keluar atau transaksi penjualan barang akan otomatis mengurangi stok barang yang tersedia di gudang. Apabila menghapus data transaksi penjualan, maka akan otomatis menambah jumlah stok yang tersedia di master produk sesuai dengan jumlah stok yang sebelumnya telah dikurangi.

4.2.4. *Entity Relationship Diagram*

Entity Relationship Diagram atau ERD berisi tentang desain basis data yang merupakan representasi yang ada pada sistem informasi pergudangan yang dibuat. Dalam *Entity Relationship Diagram* terdapat 2 model yang digunakan dalam memodelkan basis data yang akan dibuat, yaitu *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM).

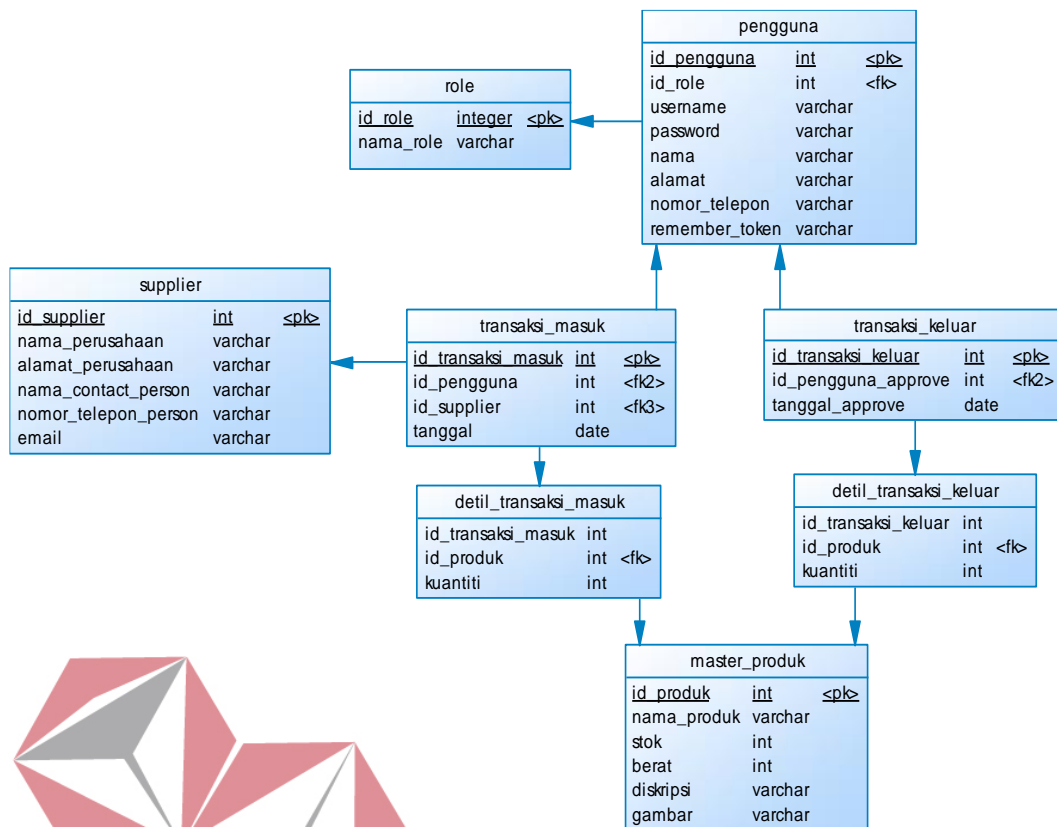
Dari kedua model yang terdapat dalam *Entity Relationship Diagram*, terdapat beberapa objek yaitu entitas, atribut, dan juga relasi. Entitas sendiri merupakan sebuah objek dan atribut merupakan bagian dari objek itu sendiri. Penghubung dari entitas yang satu dengan entitas yang lainnya adalah sebuah relasi dan hubungan tersebut disesuaikan dengan kegiatan yang ada didalam entitas tersebut.



Gambar 4.9 Conceptual Data Model

Conceptual Data Model (CDM) merupakan gambaran secara umum dari struktur aplikasi yang akan dibuat. Dengan adanya CDM, maka penulis dapat membuat desain awal sistem tanpa harus mengkhawatirkan implementasi secara fisik. Melalui *prosedur generation* yang mudah, penulis dapat melakukan *generate* CDM ke PDM dengan cepat. Dari Gambar 4.9, terlihat bahwa setiap *role* bisa dimiliki banyak pengguna.

Kemudian setiap pengguna bisa melakukan banyak transaksi masuk maupun transaksi keluar. Untuk setiap transaksi bisa saja memiliki banyak detail transaksi baik itu detail transaksi masuk maupun detail transaksi keluar. Selanjutnya setiap supplier bisa saja memiliki banyak transaksi masuk. Seluruh detail transaksi baik detail transaksi masuk maupun detail transaksi keluar bisa saja memiliki banyak produk karena dalam melakukan pemesanan produk maupun penjualan produk pasti memiliki banyak pilihan produk untuk di *re-stock* maupun dijual.



Gambar 4.10 Physical Data Model

Physical Data Model (PDM) menggambarkan tentang relasi antar tabel yang satu dengan tabel lainnya yang akan digunakan dalam aplikasi sistem informasi perdagangan sesuai dengan *Database Management Systems* (DBMS). Dari Gambar 4.10 juga merupakan pedoman utama dalam membuat basis data sistem informasi perdagangan agar sesuai dengan proses bisnis yang sedang terjadi didalam perusahaan.

4.2.5. Struktur Tabel

Sebelum membuat basis data diperlukan adanya perancangan terlebih dahulu struktur tabel yang akan digunakan dalam pembuatan sistem informasi perdagangan pada PT. Eyro Digital Teknologi. Struktur tabel tersebut meliputi nama tabel, tipe data, atribut, *primary key*, *foreign key*, dan juga penjelasan fungsi tentang tabel – tabel tersebut sebagai berikut ini :

1. Tabel *Role*

Nama Tabel : *role*

Primary Key : *id_role*

Foreign Key : -

Fungsi : Sebagai pemberi status pengguna yang melakukan *login*.

Tabel 4.1 Tabel *Role*

No.	Field	Tipe Data	Panjang	Keterangan
1	<i>id_role</i>	int	10	PRIMARY_KEY
2	<i>nama_role</i>	varchar	72	



2. Tabel Pengguna

Nama Tabel : pengguna

Primary Key : id_pengguna

Foreign Key : id_role

Fungsi : Menyimpan data pengguna atau data seluruh anggota perusahaan.

Tabel 4.2 Tabel Pengguna

No.	Field	Tipe Data	Panjang	Keterangan
1	id_pengguna	int	10	PRIMARY_KEY
2	id_role	int	10	FOREIGN_KEY
3	username	varchar	72	
4	password	varchar	72	
5	nama	varchar	72	
6	alamat	varchar	72	
7	nomor_telepon	varchar	72	
8	remember_token	varchar	72	

3. Tabel *Supplier*

Nama Tabel : *supplier*

Primary Key : id_supplier

Foreign Key : -

Fungsi : Menyimpan daftar para pemasok bahan produksi barang.

Tabel 4.3 Tabel *Supplier*

No.	Field	Tipe Data	Panjang	Keterangan
1	id_supplier	int	10	PRIMARY_KEY
2	nama_perusahaan	varchar	72	
3	alamat_perusahaan	varchar	72	
4	nama_contact_person	varchar	72	
5	nomor_telepon_contact_person	varchar	72	
6	Email	varchar	72	

4. Tabel Transaksi Masuk

Nama Tabel : transaksi_masuk

Primary Key : id_transaksi_masuk

Foreign Key : id_pengguna, id_supplier

Fungsi : Menyimpan data pemesanan bahan baku produksi.

Tabel 4.4 Tabel Transaksi Masuk

No.	Field	Tipe Data	Panjang	Keterangan
1	id_transaksi_masuk	int	10	PRIMARY_KEY
2	id_pengguna	int	10	FOREIGN_KEY
3	id_supplier	int	10	FOREIGN_KEY
4	Tanggal	date		

5. Tabel Detil Transaksi Masuk

Nama Tabel : detil_transaksi_masuk

Primary Key : -

Foreign Key : id_transaksi_masuk, id_produk

Fungsi : Sebagai detil dari transaksi masuk seperti jumlah produk.

Tabel 4.5 Tabel Detil Transaksi_Masuk

No.	Field	Tipe Data	Panjang	Keterangan
1	id_transaksi_masuk	int	10	
2	id_produk	int	10	
4	Kuantiti	int	11	

6. Tabel Transaksi Keluar

Nama Tabel : transaksi_keluar

Primary Key : id_transaksi_keluar

Foreign Key : id_pengguna_approve

Fungsi : Menyimpan data penjualan perusahaan.

Tabel 4.6 Tabel Transaksi Keluar

No.	Field	Tipe Data	Panjang	Keterangan
1	id_transaksi_keluar	int	10	PRIMARY_KEY
2	id_pengguna_approve	int	10	FOREIGN_KEY
4	tanggal_approve	date		

7. Tabel Detil Transaksi Keluar

Nama Tabel : detil_transaksi_keluar

Primary Key : -

Foreign Key : id_transaksi_keluar, id_produk

Fungsi : Sebagai detil dari penjualan seperti jumlah produk yang terjual.

Tabel 4.7 Tabel Detil Transaksi Keluar

No.	Field	Tipe Data	Panjang	Keterangan
1	id_transaksi_keluar	int	10	
2	id_produk	int	10	
4	Kuantiti	int	11	

8. Tabel Master Produk

Nama Tabel : master_produk

Primary Key : id_produk

Foreign Key : -

Fungsi : Menyimpan daftar seluruh produk yang dijual oleh perusahaan.

Tabel 4.8 Tabel Master Produk

No.	Field	Tipe Data	Panjang	Keterangan
1	id_produk	int	10	PRIMARY_KEY
2	nama_produk	varchar	72	
3	Stok	int	11	
4	Berat	int	11	
5	diskripsi	varchar	72	
6	gambar	varchar	72	

4.3. Kebutuhan Sistem

4.3.1. Procedure

1. Laporan 7 Hari Dashboard

Untuk membuat *dashboard* laporan penjualan dibutuhkan adanya prosedur yang mampu menampilkan penjualan selama 7 hari. Prosedur Laporan 7 Hari *Dashboard* berisi tentang data penjualan seluruh barang dalam jangka waktu 7 hari. Nama prosedur tersebut adalah “*Laporan7HariSpecials()*” dan prosedur tersebut akan digunakan

didalam Laravel yang merupakan tugas dari salah satu anggota tim kerja praktik. Berikut *source code* yang merupakan laporan 7 harian pada *dashboard* :

DELIMITER |

CREATE PROCEDURE Laporan7HariSpecials()

BEGIN

SELECT DAY(tk.tanggal_approve) AS tanggal,
MONTH(tk.tanggal_approve) AS bulan, YEAR(tk.tanggal_approve)
AS tahun, SUM(dtk.kuantiti) AS total FROM transaksi_keluar tk

INNER JOIN detil_transaksi_keluar dtk

ON dtk.id_transaksi_keluar = tk.id_transaksi_keluar

INNER JOIN master_produk mp

ON mp.id_produk = dtk.id_produk

WHERE tk.tanggal_approve <= NOW() AND tk.tanggal_approve
>= DATE_ADD(NOW(), INTERVAL- 8 DAY)

GROUP BY DAY(tk.tanggal_approve)

ORDER BY YEAR(tk.tanggal_approve) DESC,
MONTH(tk.tanggal_approve) DESC, DAY (tk.tanggal_approve) ASC;
END;

|

DELIMITER ;

CALL Laporan7HariSpecials();

2. Laporan 7 Bulan Dashboard

Prosedur berikut berisi tentang laporan penjualan selama 7 bulan dalam *dashboard* :

DELIMITER |

```
CREATE PROCEDURE Laporan7BulananSpecials()
```

```
BEGIN
```

```
    SELECT    MONTHNAME(tk.tanggal_approve)    AS    bulan,
    YEAR(tk.tanggal_approve) AS tahun, SUM(dtk.kuantiti) AS total
```

```
FROM transaksi_keluar tk
```

```
    INNER JOIN detil_transaksi_keluar dtk
```

```
    ON dtk.id_transaksi_keluar = tk.id_transaksi_keluar
```

```
    INNER JOIN master_produk mp
```

```
    ON mp.id_produk = dtk.id_produk
```

```
    WHERE tk.tanggal_approve <= NOW() AND tk.tanggal_approve
    >= DATE_ADD(NOW(), INTERVAL-7 MONTH)
```

```
    GROUP BY MONTH(tk.tanggal_approve)
```

```
    ORDER    BY    YEAR(tk.tanggal_approve)    ASC,
    MONTH(tk.tanggal_approve) ASC;
```

```
END;
```

```
|
```

```
DELIMITER ;
```

```
CALL Laporan7BulananSpecials();
```

3. Laporan 7 Tahun Dashboard

Prosedur berikut berisi tentang laporan penjualan selama 7 tahun dalam *dashboard* :

DELIMITER |

```
CREATE PROCEDURE Laporan7TahunanSpecials()
```

```
BEGIN
```

```
    SELECT YEAR(tk.tanggal_approve) AS tahun, SUM(dtk.kuantiti)
```

```
    AS total FROM transaksi_keluar tk
```

```
    INNER JOIN detil_transaksi_keluar dtk
```

```
    ON dtk.id_transaksi_keluar = tk.id_transaksi_keluar
```

```
    INNER JOIN master_produk mp
```

```
    ON mp.id_produk = dtk.id_produk
```

```
    WHERE YEAR(tk.tanggal_approve) BETWEEN YEAR(NOW())-
```

```
    7 AND YEAR(NOW())
```

```
    GROUP BY YEAR(tk.tanggal_approve)
```

```
    ORDER BY YEAR(tk.tanggal_approve) ASC;
```

```
END;
```

```
|
```

DELIMITER ;

```
CALL Laporan7TahunanSpecials();
```

4. Laporan 7 Hari Detil

Dalam detil laporan selama 7 hari terdapat beberapa grafik yang akan ditampilkan dalam Laravel. Penulis memiliki tugas untuk membuat *Query* prosedur yang mampu menampilkan data penjualan selama 7 hari terhadap masing – masing produk. Dikarenakan produk dalam PT. Eyro Digital Teknologi hanya terdapat 10 jenis barang, maka pemanggilan prosedur hanya sampai pada id_produk yang ke sepuluh. Nama dari prosedur yang digunakan untuk menampilkan detil laporan selama 7 hari adalah prosedur “*Laporan7Harian()*”. Dalam tanda kurung tersebut dapat diisi id_produk sesuai dengan yang manajer hendak tampilkan. Dalam aplikasi yang telah penulis dan tim kerja praktik buat, detil laporan penjualan yang ditampilkan adalah seluruh produk, sehingga penulis mengisi angka 1 sampai 10 pada tanda kurung tersebut. Berikut *source code* prosedur tersebut :

DELIMITER |

```
CREATE PROCEDURE Laporan7Harian(IN id_produk_param INT)
BEGIN
    SELECT      DAY(tk.tanggal_approve)      AS      tanggal,
    MONTH(tk.tanggal_approve) AS bulan, YEAR(tk.tanggal_approve)
    AS tahun, mp.nama_produk, SUM(dtk.kuantiti) AS total FROM
    transaksi_keluar tk
    INNER JOIN detil_transaksi_keluar dtk
    ON dtk.id_transaksi_keluar = tk.id_transaksi_keluar
```

```

INNER JOIN master_produk mp

ON mp.id_produk = dtk.id_produk

WHERE tk.tanggal_approve <= NOW() AND tk.tanggal_approve
>= DATE_ADD(NOW(), INTERVAL- 8 DAY) AND mp.id_produk =
id_produk_param

GROUP BY DAY(tk.tanggal_approve), mp.nama_produk

ORDER BY YEAR(tk.tanggal_approve) DESC,
MONTH(tk.tanggal_approve) DESC, DAY (tk.tanggal_approve) ASC;

END;

```



DELIMITER ;

CALL Laporan7Harian (1);

CALL Laporan7Harian (2);

CALL Laporan7Harian (3);

CALL Laporan7Harian (4);

CALL Laporan7Harian (5);

CALL Laporan7Harian (6);

CALL Laporan7Harian (7);

CALL Laporan7Harian (8);

CALL Laporan7Harian (9);

CALL Laporan7Harian (10);

5. Laporan 7 Bulan Detil

Dalam detil laporan selama 7 bulan juga terdapat beberapa grafik yang akan ditampilkan dalam Laravel. Penulis memiliki tugas untuk membuat *Query* prosedur yang mampu menampilkan data penjualan selama 7 bulan terhadap masing – masing produk. Dikarenakan produk dalam PT. Eyro Digital Teknologi hanya terdapat 10 jenis barang, maka pemanggilan prosedur hanya sampai pada id_produk yang ke sepuluh pula. Nama dari prosedur yang digunakan untuk menampilkan detil laporan selama 7 bulan adalah prosedur “*Laporan7Bulanan()*”. Dalam tanda kurung tersebut dapat diisi id_produk sesuai dengan yang Direktur Utama hendak tampilkan. Dalam aplikasi yang telah penulis dan tim kerja praktik buat, detil laporan penjualan yang ditampilkan adalah seluruh produk, sehingga penulis mengisi angka 1 sampai 10 pada tanda kurung tersebut. Berikut *source code* prosedur tersebut :

DELIMITER |

```
CREATE PROCEDURE Laporan7Bulanan(IN id_produk_param INT)
BEGIN
    SELECT    MONTHNAME(tk.tanggal_approve)    AS    bulan,
    YEAR(tk.tanggal_approve)    AS    tahun,    mp.nama_produk,
    SUM(dtk.kuantiti) AS total FROM transaksi_keluar tk
    INNER JOIN detil_transaksi_keluar dtk
    ON dtk.id_transaksi_keluar = tk.id_transaksi_keluar
    INNER JOIN master_produk mp
```

```

ON mp.id_produk = dtk.id_produk

WHERE tk.tanggal_approve <= NOW() AND tk.tanggal_approve
>= DATE_ADD(NOW(), INTERVAL- 7 MONTH) AND
mp.id_produk = id_produk_param

GROUP BY MONTH(tk.tanggal_approve), mp.nama_produk

ORDER BY YEAR(tk.tanggal_approve) ASC,
MONTH(tk.tanggal_approve) ASC;

END;

```



DELIMITER ;

CALL Laporan7Bulanan (1);

CALL Laporan7Bulanan (2);

CALL Laporan7Bulanan (3);

CALL Laporan7Bulanan (4);

CALL Laporan7Bulanan (5);

CALL Laporan7Bulanan (6);

CALL Laporan7Bulanan (7);

CALL Laporan7Bulanan (8);

CALL Laporan7Bulanan (9);

CALL Laporan7Bulanan (10);

6. Laporan 7 Tahun Detil

Dalam detil laporan selama 7 tahun juga terdapat beberapa grafik yang akan ditampilkan dalam Laravel. Penulis memiliki tugas untuk membuat *Query* prosedur yang mampu menampilkan data penjualan selama 7 tahun terhadap masing – masing produk. Dikarenakan produk dalam PT. Eyro Digital Teknologi hanya terdapat 10 jenis barang, maka pemanggilan prosedur hanya sampai pada id_produk yang ke sepuluh pula. Nama dari prosedur yang digunakan untuk menampilkan detil laporan selama 7 tahun adalah prosedur “*Laporan7Tahunan()*”. Dalam tanda kurung tersebut dapat diisi id_produk sesuai dengan yang manajer hendak tampilkan. Dalam aplikasi yang telah penulis dan tim kerja praktik buat, detil laporan penjualan yang ditampilkan adalah seluruh produk, sehingga penulis mengisi angka 1 sampai 10 pada tanda kurung tersebut. Detil Laporan 7 Tahunan berikut merupakan isi lebih detil dari laporan penjualan selama 7 tahun dari dashboard berdasarkan dari setiap id_produk. Berikut *source code* prosedur laporan penjualan 7 tahun setiap produk tersebut :

DELIMITER |

```
CREATE PROCEDURE Laporan7Tahunan(IN id_produk_param INT)
```

```
BEGIN
```

```
    SELECT YEAR(tk.tanggal_approve) AS tahun, mp.nama_produk,
```

```
    SUM(dtk.kuantiti) AS total FROM transaksi_keluar tk
```

```
    INNER JOIN detil_transaksi_keluar dtk
```

```

ON dtk.id_transaksi_keluar = tk.id_transaksi_keluar

INNER JOIN master_produk mp

ON mp.id_produk = dtk.id_produk

WHERE YEAR(tk.tanggal_approve) BETWEEN YEAR(NOW())-
7 AND YEAR(NOW()) AND mp.id_produk = id_produk_param

GROUP BY YEAR(tk.tanggal_approve), mp.nama_produk

ORDER BY YEAR(tk.tanggal_approve) ASC;

END;

```



DELIMITER ;

CALL Laporan7Tahunan (1);

CALL Laporan7Tahunan (2);

CALL Laporan7Tahunan (3);

CALL Laporan7Tahunan (4);

CALL Laporan7Tahunan (5);

CALL Laporan7Tahunan (6);

CALL Laporan7Tahunan (7);

CALL Laporan7Tahunan (8);

CALL Laporan7Tahunan (9);

CALL Laporan7Tahunan (10);

4.3.2. Triggers

1. Triggers Menambah Stok Transaksi Keluar

Berfungsi sebagai pemacu apabila terjadi pembatalan transaksi keluar maka akan otomatis menambah kembali stok produk sesuai dengan jumlah sebelumnya yang telah dikurangi pada saat terjadi transaksi keluar.

DELIMITER \$\$

CREATE TRIGGER tambah_stok_transaksi_keluar

AFTER DELETE ON detil_transaksi_keluar

FOR EACH ROW

BEGIN

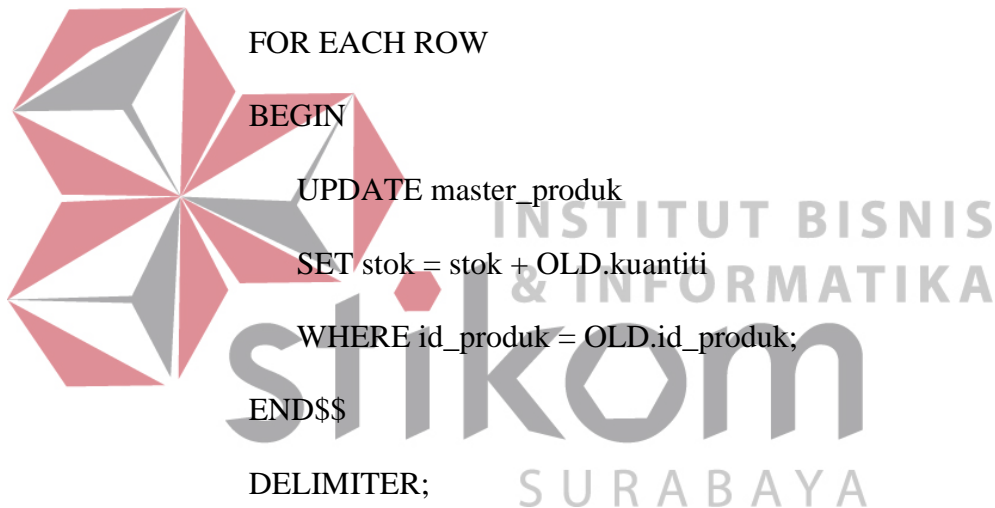
UPDATE master_produk

SET stok = stok + OLD.kuantiti

WHERE id_produk = OLD.id_produk;

END\$\$

DELIMITER;



2. *Triggers* Mengurangi Stok Transaksi Keluar

Berfungsi sebagai pemacu apabila terjadi transaksi keluar maka akan otomatis mengurangi stok produk pada master produk.

```
DELIMITER $$
```

```
CREATE TRIGGER kurangi_stok_transaksi_keluar
```

```
AFTER INSERT ON detail_transaksi_keluar
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE master_produk
```

```
    SET stok = stok - NEW.kuantiti
```

```
    WHERE id_produk = NEW.id_produk;
```

```
END$$
```

```
DELIMITER;
```



3. *Triggers* Memperbaharui Stok Transaksi Keluar

Berfungsi sebagai pemacu apabila terjadi perubahan transaksi keluar maka akan otomatis menambah atau mengurangi kembali stok produk.

DELIMITER \$\$

CREATE TRIGGER update_stok_transaksi_keluar

AFTER UPDATE ON detil_transaksi_keluar

FOR EACH ROW

BEGIN

IF (OLD.kuantiti < NEW.kuantiti) THEN

UPDATE master_produk

SET stok = stok - (NEW.kuantiti-OLD.kuantiti)

WHERE id_produk = OLD.id_produk;

ELSEIF (OLD.kuantiti > NEW.kuantiti) THEN

UPDATE master_produk

SET stok = stok + (OLD.kuantiti-NEW.kuantiti)

WHERE id_produk = OLD.id_produk;

ELSE

UPDATE master_produk

SET stok = stok + 0

WHERE id_produk = OLD.id_produk;

END IF;

END\$\$

DELIMITER;

4. *Triggers* Menambah Stok Transaksi Masuk

Berfungsi sebagai pemacu apabila terjadi transaksi masuk maka akan otomatis menambah stok produk.

```
DELIMITER $$
```

```
CREATE TRIGGER tambah_stok_transaksi_masuk
```

```
AFTER INSERT ON detail_transaksi_masuk
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE master_produk
```

```
    SET stok = stok + NEW.kuantiti
```

```
    WHERE id_produk = NEW.id_produk;
```

```
END$$
```

```
DELIMITER;
```



5. *Triggers* Mengurangi Stok Transaksi Masuk

Berfungsi sebagai pemacu apabila terjadi pembatalan transaksi masuk maka akan otomatis mengurangi kembali stok produk sesuai dengan jumlah sebelumnya yang telah ditambah pada saat terjadi transaksi masuk.

DELIMITER \$\$

CREATE TRIGGER kurangi_stok_transaksi_masuk

AFTER DELETE ON detail_transaksi_masuk

FOR EACH ROW

BEGIN

UPDATE master_produk

SET stok = stok - OLD.kuantiti

WHERE id_produk = OLD.id_produk;

END\$\$

DELIMITER;



6. *Triggers* Memperbaharui Stok Transaksi Masuk

Berfungsi sebagai pemacu apabila terjadi perubahan transaksi masuk maka akan otomatis menambah atau mengurangi kembali stok produk.

DELIMITER \$\$

CREATE TRIGGER update_stok_transaksi_masuk

AFTER UPDATE ON detil_transaksi_masuk

FOR EACH ROW

BEGIN

IF (OLD.kuantiti < NEW.kuantiti) THEN

UPDATE master_produk

SET stok = stok + (NEW.kuantiti-OLD.kuantiti)

WHERE id_produk = OLD.id_produk;

ELSEIF (OLD.kuantiti > NEW.kuantiti) THEN

UPDATE master_produk

SET stok = stok - (OLD.kuantiti-NEW.kuantiti)

WHERE id_produk = OLD.id_produk;

ELSE

UPDATE master_produk

SET stok = stok + 0

WHERE id_produk = OLD.id_produk;

END IF;

END\$\$

DELIMITER;