PERANCANGAN MODEL BASIS DATA RELASIONAL DENGAN METODE DATABASE LIFE CYCLE

Wahyu Sindu Prasetya

STMIK Pontianak
Sistem Informasi STMIK Pontianak
e-mail : wahyusinduprasetya@gmail.com

Abstrak

Pemanfaatan basis data pada bidang penjualan memungkinkan untuk dapat menyimpan data, melakukan perubahan dan menampilkan. Ada aspek yang sulit dalam merancang database bahwa perancang, programmer dan pemakai akhir cenderung melihat data dengan cara yang berbeda sehingga diperlukan sebuah metodologi yang menggunakan prosedur, teknik, peralatan, dan dokumentasi untuk mendukung dan memfasilitasi proses perancangan basis data. Metode penelitian yang digunakan adalah metode penelitian deskriptif. Metodologi perancangan basis data menggunakan DBLC (*database life cycle*) dengan variabel penelitian adalah perancangan model basis data relasional dengan metode *database life cycle*. Perancangan basis data relasional meliputi *Conceptual Database Design, Logical Database Design* dan *Physical Database Design*. Hasil dari rancangan basis data model relasional penjualan barang dimaksudkan untuk menjaga integritas data dari setiap tabel yang berrelasi. Hasil akhir dari penelitian ini adalah menghasil 10 (sepuluh) tipe entitas konsepsual, menghasilkan diagram hubungan entitas dari kesepuluh entitas tersebut pada logikal dan menghasilkan rancangan pisikal yang terdiri jenis, merk, satuan, supplier, barang, komsumen, master beli, detil beli, master jual dan detil jual. Dengan adanya rancangan basis data relational ini akan memberikan manfaat bagi para pengembang aplikasi penjualan dengan model basis data yang baik dan benar.

Kata Kunci : Basis Data, Data Relasional, DBLC dan Penjualan.

1. PENDAHULUAN

Basis data merupakan urat nadi sistem informasi sehingga perananannya dalam membentuk konsep laporan sangatlah penting yang membuat para pemakai dapat menggunakannya sesuai dengan kebutuhan[1]. Model basis data relasional merupakan suatu cara untuk merepresentasikan model data dalam perancangan basis data dimana model dari basis data relasional didasarkan pada record[4]. Perancangan basis data merupakan proses membuat desain yang akan mendukung operasional dan tujuan perusahaan[2]. Pemanfaatan database dalam sebuah aplikasi memungkinkan untuk dapat menyimpan data atau melakukan perubahan dan menampilkan kembali data tersebut dengan cepat dan mudah. Metodologi perancangan terdiri dari beberapa fase dimana setiap fase mengandung beberapa langkah yang akan menuntun desainer dalam menggunakan teknik yang sesuai pada setiap tahap dalam proyek sehingga membantu desainer untuk merencanakan, mengelola, mengatur, dan mengevaluasi pengembangan proyek database[3]. Perancangan basis data konseptual membangun model data yang digunakan dalam suatu perusahaan, serta terbebas dari semua pertimbangan fisik. Perancangan basis data logikal merancang model data yang digunakan dalam suatu perusahaan berdasarkan pada model data yang spesifik. Perancangan basis data fisikal menghasilkan deskripsi implementasi basis data pada penyimpanan sekunder, menggambarkan hubungan dasar, organisasi file, dan indeks yang digunakan untuk mencapai akses yang efisien terhadap data dan setiap kendala integritas terkait dan langkah-langkah keamanan[7]. Database relasional mempresentasikan semua data dalam database sebagai tabel dua dimensi[5]. Sumber daya dalam komputerisasi berupa perangkat lunak, perangkat keras, media penyimpanan, orang yang menggunakan dan mengatur[6]. Toko Sinar Elektronik merupakan sebuah toko yang bergerak dibidang penjualan barang elektronik memerlukan dukungan aplikasi penjualan untuk memperlancar dalam menjalankan aktivitas bisnisnya. Hasil penelitian ini diharapkan dapat memberikan manfaat bagi para pengembang aplikasi dalam berbagai kasus terutama yang terkait dengan aktivitas penjualan barang dengan model basis data yang baik dan benar. Penelitian ini memiliki kesamaan dengan penelitian terdahulu dimana sama-sama membangun basis data berdasarkan model relasional. Makalah ini lebih terfokus kepada model dari suatu database

yang dibangun dengan konsep perancangan *database life cycle* dengan menerapkan teknik *conseptual database design, logical database design* dan *physical database design*. Teknik inilah yang membedakan makalah ini dengan penelitian terdahulu seperti yang telah dijelaskan pada latar belakang.

2. METODE PENELITIAN

Penelitian ini menggunakan metode penelitian deskriptif. Metodologi perancangan basis data yang penulis gunakan adalah DBLC (*database life cycle*), yaitu metode yang menjelaskan mengenai siklus hidup dari database. DBLC ini akan terus kembali ketitik awal karena sebuah basis data yang akan dibuat pasti akan membutuhkan perbaikan sesuai dengan perkembangan. Proses dalam DBLC dibagi menjadi tiga tahap, yaitu perancangan basis data konseptual, logikal, dan fisikal.

3. HASIL DAN PEMBAHASAN

Permasalahan yang dihadapi pada waktu perancangan adalah bagaimana basis data yang akan dibangun ini dapat memenuhi kebutuhan saat ini dan masa yang akan datang. Oleh sebab itu diperlukan perancangan basis data baik dalam bentuk fisik maupun dalam bentuk konseptual. Perancangan konseptual akan menunjukkan entity dan relasinya berdasarkan proses yang diinginkan oleh perusahaan. Untuk menentukan entity dan relasinya perlu dilakukan analisis data tentang informasi yang ada dalam spesifikasi di masa yang akan datang. Teknik yang digunakan pada perancangan basis data dibagi dalam tiga tahap, yaitu perancangan basis data konseptual (*conseptual database design*), perancangan basis data logikal (*logical database design*) dan perancangan basis data fisikal (*physical database design*).

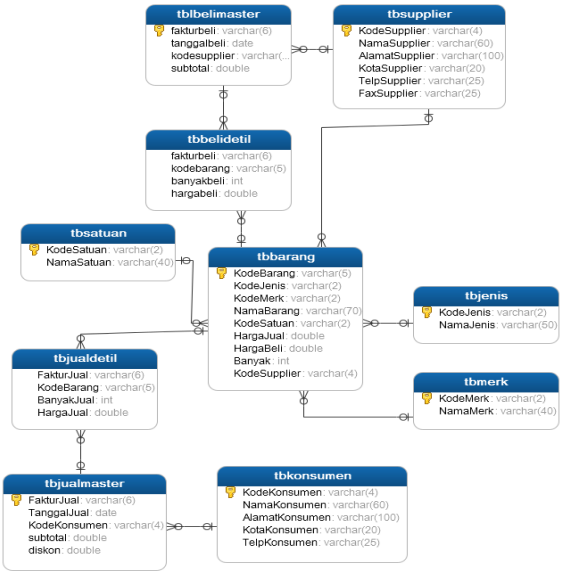
a. Conseptual Database Design

Conceptual database design adalah proses membangun suatu model berdasarkan informasi yang digunakan oleh perusahaan atau organisasi, tanpa pertimbangan perencanaan fisik dan bersifat independen dari semua pertimbangan fisikal. Tahap desain konseptual database dimulai dengan membuat model data konseptual dari perusahaan dengan rincian implementasi seperti target DBMS, program aplikasi, bahasa pemrograman, hardware, platform, performance dan segala pertimbangan fisikal lain nya (tabel 1. Identifikasi Tipe Entitas).

Tabel 1. Identifikasi Tipe Entitas

Nama Entity	Keterangan Entity	Kegiatan
Jenis	Berisi informasi mengenai data jenis barang.	Pengelompokan data barang berdasarkan jenis barang dan satu jenis barang bisa terdiri dari beberapa barang.
Merk	berisi informasi mengenai data merk barang.	Pengelompokan data barang berdasarkan merk dan satu merk barang bisa terdiri dari beberapa barang.
Satuan	Berisi informasi mengenai data satuan barang.	Pengelompokan data barang berdasarkan satuan dan satu satuan barang bisa terdiri dari beberapa barang.
Supplier	Berisi informasi mengenai data supplier barang.	Pengelompokan data barang berdasarkan supplier dan satu supplier bisa terdiri dari beberapa barang.
Barang	Berisi informasi mengenai data barang.	Setiap barang dapat dijual kepada satu atau beberapa konsumen
Konsumen	Berisi informasi mengenai data konsumen.	Konsumen dapat melakukan beberapa kali pembelian terhadap barang
Master Jual	Berisi informasi mengenai data transaksi penjualan master.	Dapat menyimpan hanya satu jenis faktur dalam transaksi penjualan.
Detil Jual	Berisi informasi mengenai data transaksi penjualan secara detil.	Satu faktur penjualan bisa terdiri dari satu barang atau beberapa barang.
Master Beli	Berisi informasi mengenai data transaksi pembelian master.	Dapat menyimpan hanya satu jenis faktur dalam transaksi pembelian.
Detil Beli	Berisi informasi mengenai data transaksi pembelian secara detil.	Satu faktur pembelian bisa terdiri dari satu barang atau beberapa barang.

Melakukan identifikasi tipe rasional bertujuan untuk menentukan hubungan-hubungan penting yang ada antara jenis-jenis entitas yang telah diidentifikasikan sebelumnya.



Gambar 1. E-R Diagram Konseptual

Domain adalah seluruh kemungkinan nilai yang dapat diberikan kesuatu atribut. Memberi nama domain yang sesuai dengan nilai yang akan dimiliki domain tersebut. Domain menentukan tipe data dari nilai yang akan membentuk domain dan menentukan format dari domain.

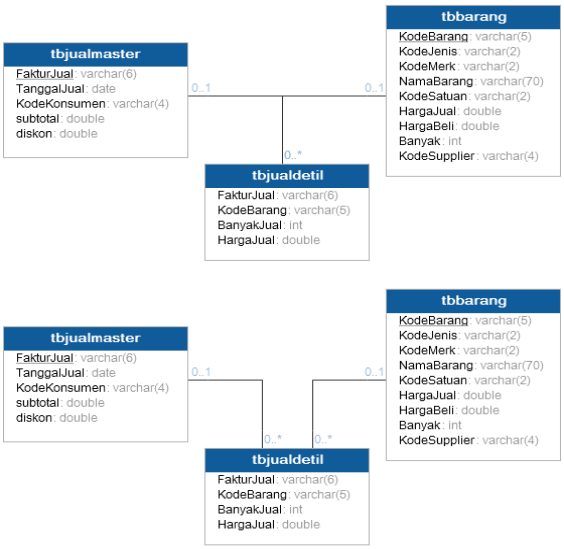
Tabel 2 Tabel Attribute Domain

Entity Name	Attribute	Domain
Jenis	KodeJenis	String dengan panjang maksimal 2 karakter
	NamaJenis	String dengan panjang maksimal 50 karakter
Merk	KodeMerk	String dengan panjang maksimal 2 karakter
	NamaMerk	String dengan panjang maksimal 40 karakter
Satuan	KodeSatuan	String dengan panjang maksimal 2 karakter
	NamaSatuan	String dengan panjang maksimal 40 karakter
		karakter
Supplier	KodeSupplier	String dengan panjang maksimal 2 karakter
	NamaSupplier	String dengan panjang maksimal 60 karakter
	AlamatSupplier	String dengan panjang maksimal 100 karakter
	KotaSupplier	String dengan panjang maksimal 20 karakter
	TelpSupplier	String dengan panjang maksimal 25 karakter
	FaxSupplier	String dengan panjang maksimal 25 karakter
Barang	KodeBarang	String dengan panjang maksimal 5 karakter
	KodeJenis	String dengan panjang maksimal 2 karakter
	KodeMerk	String dengan panjang maksimal 2 karakter
	NamaBarang	String dengan panjang maksimal 70 karakter
	KodeSatuan	String dengan panjang maksimal 2 karakter
	HargaJual	Number
	HargaBeli	Number
	Banyak	Number

	KodeSupplier	String dengan panjang maksimal 2 karakter
Konsumen	KodeKonsumen	String dengan panjang maksimal 4 karakter
	NamaKonsumen	String dengan panjang maksimal 60 karakter
	AlamatKonsumen	String dengan panjang maksimal 100 karakter
	KotaKonsumen	String dengan panjang maksimal 20 karakter
	TelpKonsumen	String dengan panjang maksimal 25 karakter
Master Jual	FakturJual	String dengan panjang maksimal 6 karakter
	TanggalJual	Date dengan panjang maksimal 10 karakter
	KodeKonsumen	String dengan panjang maksimal 4 karakter
	subtotal	Number
	diskon	Number
Detil Jual	FakturJual	String dengan panjang maksimal 6 karakter
	KodeBarang	String dengan panjang maksimal 5 karakter
	BanyakJual	Number
	HargaJual	Number
Master	Fakturbeli	String
Beli		dengan panjang maksimal 6 karakter
	Tanggalbeli	Date dengan panjang maksimal 10 karakter
	Kodesupplier	String dengan panjang maksimal 2 karakter
	Subtotal	Number
Detil Beli	Fakturbeli	String dengan panjang maksimal 6 karakter
	Kodebarang	String dengan panjang maksimal 5 karakter
	Banyakbeli	Number
	Hargabeli	Number

b. Logical Database Design

Dalam langkah ini adalah memilih DBMS yang akan digunakan untuk mengimplementasikan desain database dan mengubah konsep desain database menjadi sebuah skema database dalam model data dari DBMS terpilih. Dalam sistem basis data relasional yang akan digunakan, ada hal-hal dalam perancangan basis data logikal yang tidak bisa diimplementasikan oleh sebab itu, dalam rancangan database relasional perlu diadakan modifikasi, yaitu menghilangkan bagian yang tidak kompatibel dari model data konseptual. Langkah-langkanya antara lain menghilangkan relasi biner *many-to-many*, relasi rekursif *many-to-many*, relasi kompleks dan atribut *multivalued*. Untuk menghilangkan tipe hubungan yang mengandung *many-to-many* (*.*) dipecah dengan mengidentifikasi sebuah entitas baru dan mengganti hubungannya dengan *one-to-many* (1.*) sehingga menghilangkan hubungan *many-to-many*. (gambar 2. Hubungan Barang Dengan Transaksi Penjualan).



Gambar 2. Hubungan Barang Dengan Transaksi Penjualan

Pemakaian normalisasi dimaksudkan untuk meminimalkan kemungkinan terjadinya data rangkap, menghindari data yang tidak konsisten terutama bila dilakukan penambahan atau penghapusan data sebagai akibat karena adanya data yang rangkap dan untuk menjamin bahwa identitas tabel secara tunggal sebagai determinan semua atribut.

1. Bentuk unnormal

{kodebarang, namabarang, hargajual, hargabeli, banyak, namajenis, namamerk, namasatuan, namakonsumen, alamatkonsumen, kotakonsumen, telpkonsumen, namasupplier, alamatsupplier, kotasupplier, telpsupplier, faxsupplier, fakturbeli, tanggalbeli, banyakbeli, hargabeli, fakturjual, tanggaljual, banyakjual, hargajual}

2. Bentuk Normal Pertama

Langkah berikutnya adalah dengan cara memisahkan atribut-atribut yang nilainya sama akan ditulis hanya satu kali.

tabel barang {*kodebarang, namabarang, hargajual, hargabeli, banyak, namajenis, namamerk, namasatuan, namasupplier}
tabel pembelian {*fakturbeli, tanggalbeli, subtotal, banyakbeli, hargabeli, namasupplier}
tabel penjualan {*fakturjual, tanggaljual, diskon, banyakjual, hargajual, namakonsumen}

3. Bentuk Normal Kedua

Langkah selanjutnya adalah dengan cara menentukan ketergantungan fungsional.

Tabel barang {*kodebarang, namabarang, hargajual, hargabeli, banyak, **kodejenis, **kodemerk, **kodesatuan, **kodesupplier}
Tabel jenis {*kodejenis, namajenis}
Tabel merk {*kodemerk, namamerk}
Tabel satuan {*kodesatuan, namasatuan}
Tabel supplier {*kodesupplier, namasupplier, alamatsupplier, kotasupplier, telpsupplier, faxsupplier}
Tabel master beli {*fakturbeli, tanggalbeli, **kodesupplier, subtotal}
Tabel detil beli {**fakturbeli, **kodebarang, banyakbeli, hargabeli}
Tabel mater jual {*fakturjual, tanggaljual, **kodekonsumen, subtotal, diskon}
Tabel deti jual {**fakturjual, **kodebarang, banyakjual, hargajual}
Tabel konsumen {*kodekonsumen, namakonsumen, alamatkonsumen, kotakonsumen, telpkonsumen}

c. Physical Database Design

Pada langkah ini meliputi pembuatan indeks pada tabel dan mengelompokkan beberapa table. Proses perancangan fisik merupakan transformasi dari perancangan logis terhadap jenis DBMS yang digunakan sehingga dapat disimpan secara fisik pada media penyimpanan. *My Structured Query Language* (MySQL) merupakan pilihan DBMS yang tepat untuk mendukung aplikasi basis data.

Tabel 3. Tabel Jenis

No.	Nama Field	Type	Size
1	kodejenis *	Varchar	2
2	namajenis	Varchar	60

Tabel 4. Tabel Merk

No.	Nama Field	Type	Size
1	kodemerk*	Varchar	2
2	Namamerk	Varchar	40

Tabel 5. Tabel Satuan

No.	Nama Field	Type	Size
1	kodesatuan*	Varchar	2
2	Namasatuan	Varchar	40

Tabel 6. Tabel Supplier

No.	Nama Field	Type	Size
1	KodeSupplier*	Varchar	4
2	NamaSupplier	Varchar	60
3	AlamatSupplier	Varchar	100
4	KotaSupplier	Varchar	20
5	TelpSupplier	Varchar	25
6	FaxSupplier	Varchar	25

Tabel **7.** Tabel Barang

No.	Nama Field	Type	Size
1	KodeBarang*	Varchar	5
2	KodeJenis	Varchar	2
3	KodeMerk	Varchar	2
4	NamaBarang	Varchar	70
5	KodeSatuan	Varchar	2
6	HargaJual	DOUBLE	
7	HargaBeli	DOUBLE	
8	Banyak	INT	
9	KodeSupplier	Varchar	4

Tabel 8. Tabel Master Beli

No.	Nama Field	Type	Size
1	fakturbeli*	Varchar	5
2	Tanggalbeli	DATE	
3	Kodesupplier	Varchar	5
4	Subtotal	DOUBLE	

Tabel 9 Tabel Detil Beli

No.	Nama Field	Type	Size
1	Fakturbeli**	Varchar	5
2	Kodebarang**	Varchar	5
3	Banyakbeli	Int	4
4	Hargabeli	DOUBLE	

Tabel 10 Tabel Master Jual

No.	Nama Field	Type	Size
1	FakturJual**	Varchar	6
2	TanggalJual	DATE	
3	KodeKonsumen**	Varchar	4
4	Subtotal	DOUBLE	
5	Diskon	DOUBLE	

Tabel 11 Tabel Detil Jual

No.	Nama Field	Type	Size
1	FakturJual**	Varchar	5
2	Kodebarang**	Varchar	5
3	BanyakJual	Int	4
4	HargaJual	DOUBLE	

Data Definition Language (DDL) adalah kumpulan perintah SQL yang digunakan untuk membuat (create), mengubah (alter) dan menghapus (drop) struktur dan definisi tipe data dari objek-objek database.

a. SQL CREATE TABLE Jenis

```
CREATE TABLE `dbpenjualan`.`tbjenis` (
`KodeJenis` varchar(2) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL DEFAULT '',`NamaJenis` varchar(50) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
PRIMARY KEY (`KodeJenis`) )
```

b. SQL CREATE TABLE Merk

```
CREATE TABLE `dbpenjualan`.`tbmerk` (
`KodeMerk` varchar(2) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL DEFAULT '',`NamaMerk` varchar(40) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
PRIMARY KEY (`KodeMerk`) )
```

c. SQL CREATE TABLE Satuan

```
CREATE TABLE `dbpenjualan`.`tbsatuan` (
`KodeSatuan` varchar(2) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL DEFAULT '',`NamaSatuan` varchar(40) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
PRIMARY KEY (`KodeSatuan`) )
```

d. SQL CREATE TABLE Supplier

```
CREATE TABLE `dbpenjualan`.`tbsupplier` (
`KodeSupplier` varchar(4) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL DEFAULT '',`NamaSupplier` varchar(60) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,`AlamatSupplier` varchar(100) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,`KotaSupplier` varchar(20) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`TelpSupplier` varchar(25) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,`FaxSupplier` varchar(25) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
PRIMARY KEY (`KodeSupplier`) )
```

e. SQL CREATE TABLE Barang

```
CREATE TABLE `dbpenjualan`.`tbbarang` (
`KodeBarang` varchar(5) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL DEFAULT '',`KodeJenis` varchar(2) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`KodeMerk` varchar(2) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`NamaBarang` varchar(70) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,`KodeSatuan` varchar(2) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`HargaJual` double NULL DEFAULT NULL,
`HargaBeli` double NULL DEFAULT NULL,
`Banyak` int(11) NULL DEFAULT NULL,
`KodeSupplier` varchar(4) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL, PRIMARY KEY (`KodeBarang`),
CONSTRAINT `tbbarang_ibfk_4` FOREIGN KEY (`KodeSupplier`) REFERENCES `dbpenjualan`.`tbsupplier` (`KodeSupplier`),
CONSTRAINT `tbbarang_ibfk_1` FOREIGN KEY (`KodeJenis`) REFERENCES `dbpenjualan`.`tbjenis` (`KodeJenis`),
```

```
CONSTRAINT `tbbarang_ibfk_2` FOREIGN KEY (`KodeMerk`) REFERENCES `dbpenjualan`.`tbmerk` (`KodeMerk`),
CONSTRAINT `tbbarang_ibfk_3` FOREIGN KEY (`KodeSatuan`) REFERENCES `dbpenjualan`.`tbsatuan` (`KodeSatuan`),
INDEX `KodeJenis` (`KodeJenis`),
INDEX `KodeMerk` (`KodeMerk`),
INDEX `KodeSatuan` (`KodeSatuan`),
INDEX `KodeSupplier` (`KodeSupplier`))
```

f. SQL CREATE TABLE Master Beli

```
CREATE TABLE `dbpenjualan`.`tblbelimaster` (`fakturbeli` varchar(6) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL DEFAULT ",`tanggalbeli` date NULL DEFAULT NULL,`kodesupplier` varchar(5) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,`subtotal` double NULL DEFAULT NULL,PRIMARY KEY (`fakturbeli`) ,CONSTRAINT `tblbelimaster_ibfk_1` FOREIGN KEY (`kodesupplier`) REFERENCES `dbpenjualan`.`tbsupplier` (`KodeSupplier`),INDEX `kodesupplier` (`kodesupplier`)) ENGINE=InnoDB DEFAULT CHARACTER SET=latin1 COLLATE=latin1_swedish_ci;
```

g. SQL CREATE TABLE Detil Beli

```
CREATE TABLE `dbpenjualan`.`tbbelidetil` (
`fakturbeli` varchar(6) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`kodebarang` varchar(5) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`banyakbeli` int(4) NULL DEFAULT NULL,
`hargabeli` double NULL DEFAULT NULL,
CONSTRAINT `tbbelidetil_ibfk_2` FOREIGN KEY (`kodebarang`) REFERENCES `dbpenjualan`.`tbbarang` (`KodeBarang`),
CONSTRAINT `tbbelidetil_ibfk_1` FOREIGN KEY (`fakturbeli`) REFERENCES `dbpenjualan`.`tblbelimaster` (`fakturbeli`),
INDEX `fakturbeli` (`fakturbeli`),
INDEX `kodebarang` (`kodebarang`))
```

h. SQL CREATE TABLE Master Jual

```
CREATE TABLE `dbpenjualan`.`tbjualmaster`(
`FakturJual` varchar(6) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL DEFAULT ",
`TanggalJual` date NULL DEFAULT NULL,
`KodeKonsumen` varchar(4) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`subtotal` double NULL DEFAULT NULL,
`diskon` double NULL DEFAULT NULL,
PRIMARY KEY (`FakturJual`) ,
CONSTRAINT `tbjualmaster_ibfk_1` FOREIGN KEY (`KodeKonsumen`) REFERENCES `dbpenjualan`.`tbkonsumen` (`KodeKonsumen`),
INDEX `KodeKonsumen` (`KodeKonsumen`))
```

i. SQL CREATE TABLE Detil Jual

```
CREATE TABLE `dbpenjualan`.`tbjualdetil` (
`FakturJual` varchar(6) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`KodeBarang` varchar(5) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`BanyakJual` int(11) NULL DEFAULT NULL,
`HargaJual` double NULL DEFAULT NULL,
CONSTRAINT `tbjualdetil_ibfk_2` FOREIGN KEY (`KodeBarang`) REFERENCES `dbpenjualan`.`tbbarang` (`KodeBarang`),
CONSTRAINT `tbjualdetil_ibfk_1` FOREIGN KEY (`FakturJual`) REFERENCES `dbpenjualan`.`tbjualmaster` (`FakturJual`),
INDEX `FakturJual` (`FakturJual`),
INDEX `KodeBarang` (`KodeBarang`))
```

j. SQL CREATE TABLE Konsumen

```
CREATE TABLE `dbpenjualan`.`tbkonsumen` (
`KodeKonsumen` varchar(4) CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL DEFAULT ",
`NamaKonsumen` varchar(60) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`AlamatKonsumen` varchar(100) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`KotaKonsumen` varchar(20) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`TelpKonsumen` varchar(25) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
PRIMARY KEY (`KodeKonsumen`) )
```

Perancangan basis data ini menghasilkan 10 tabel yang pembuatannya dilakukan dengan menggunakan aplikasi MySQL Server. Perancangan basis data yang mengacu kepada model data relasional khususnya basis data penjualan barang dimaksudkan agar dalam setiap tabel yang terdapat didalam database penjualan barang saling memiliki keterkaitan demi menjamin integritas data. Selain itu, model data relasional akan memberikan gambaran yang jelas dan memberikan kemudahan bagi programmer ketika ingin membangun aplikasi penjualan.

4. KESIMPULAN

Perancangan basis data yang dirancang dengan menggunakan metode perancangan database DBLC (*Data Base Life Cycle*) telah menghasilkan bentuk database relational dengan rincian sebagai berikut ini:

a. *Conceptual database design*
Tipe entitas yang diperlukan berjumlah 10 (sepuluh) entitas dengan memberikan attribute domain pada setiap nama entitas dan menghasilkan diagram hubungan entitas.

b. *Logical Database Design*
Menghasilkan relasi untuk model data logikal lokal yang mempresentasikan entity, relationship, dan attribute yang telah diidentifikasi sebelumnya.

c. *Physical Database Design*
Perancangan database menggukan database MySQL dengan *Data Definiton Language* adalah bahasa yang digunakan untuk mendefinisikan pendefinisian data. Jumlah tabel dalam basis data penjualan ada 10 buah.

d. Kebutuhan untuk menghasilkan sebuah aplikasi yang baik tidak terlepas dari bagaimana sebuah model dari basis data relasional.

5. SARAN

Kebutuhan dalam menghasilkan rancangan basis data yang baik tidak terlepas dari pemahaman proses bisnis dari suatu perusahaan . Oleh karenanya, maka penelitian selanjutkan dapat melakukan analisis lebuh detil dari suatu proses bisnis sehingga dapat menghasilkan racangan *Conceptual Database, Logical Database* dan *Physical Database* dengan tepat.

DAFTAR PUSTAKA

[1] Asmuni, I., & Firdaus, R., 2005., *Basis Data Relasional dalam Kreasi Organisasi File Akuntansi (Suatu Bahasan atas Pendekatan Penyajian Informasi Akuntansi Perusahaan Berbasis Komputer)*., In Seminar Nasional Aplikasi Teknologi Informasi (SNATI).

[2] Connolly, Thomas and Begg, Carolyn., 2010., *Database Systems: A Practical Approach to Design, Implementation, and Management*., Fifth Edition, Pearson Education, Boston.

[3] Connolly, Thomas, Carolyn Begg., 2002., *Database Systems : A Practical Approach to Design, Implementation, and Management*., Third Edition, Pearson Education, Ltd., England.

[4] Indrajani., 2011., *Perancangan Basis Data dalam All in 1, (1st edition).*, Elex Media Komputindo, Jakarta.

[5] Joefrie, Yuri Yudhaswana, and Protus Pieter Kalatiku., 2013., *Desain basis data sistem informasi akademik di Fakultas Teknik Universitas Tadulako*., Jurnal Ilmiah Foristek, vol.2, hal 190-194.

[6] Kusnendar, J., 2009., *Perangkat Lunak untuk Mentransformasikan Model Entity Relationship ke Model Relational*., Jurnal Universitas Pendidikan Indonesia.

[7] Laudon, Kenneth C. dan Laudon, Jane P., 2005., *Management Information Systems: Managing The Digital Firm, 8 th edition.*, Prentice Hall, New Jersey.