

## PERANCANGAN BASISDATA “ Online Shop Buku Dokter ” dengan Menggunakan MySQL

Safar Kurniawan <sup>(1)</sup>, Thomas Kristian Jeriko<sup>(2)</sup>, Yohanes Edwin Wijaya <sup>(3)</sup>,

Jurusan Sistem Informasi , STIKOM Yos Sudarso  
Jalan. SMP 5 Karangklesem, Purwokerto, Jawa Tengah , Telp. (0281) 6845088

**Abstract:** This study aims to create a database on the sale of goods on the Online Shop Book doctor who housed Jl. Salemba Tengah II No.10, RT.5 / RW.8, Paseban, Senen, Central Jakarta. Creation of this database is useful to produce information fast, precise, accurate and relevant in live penjualan transaction on Online Shop Book Doctor. In this study before the database was designed using ERD as a conceptual form, continued the normalization process to eliminate the existence of multiple attributes on the entities, redundancies and anomalies that may occur in the database. Making Database Online Shop This Doctor Book uses MySQL as a DBMS in managing the information in the database. The researcher only makes recording of transaction penjualan on Online Shop Book Doctor with output transaction report and delivery of goods in the absence of financial statement, so this design as saranan owner of Online Shop Book Doctor in monitoring transaction on sale and delivery transaction.

**Keyword :** Online Shop, Basis Data, My SQL

**Abstrak :** Penelitian ini bertujuan untuk membuat *database* penjualan barang pada *Online Shop* Buku dokter yang bertempat Jl. Salemba Tengah II No.10, RT.5 / RW.8, Paseban, Senen, Jakarta Pusat. Pembuatan *database* ini berguna untuk menghasilkan informasi yang cepat , tepat , akurat serta relevan dalam menjalani transaksi penjualan pada *Online Shop* Buku Dokter. Dalam penelitian ini sebelum pembuatan *database* dilakukan perancangan menggunakan ERD sebagai bentuk konseptual , dilanjutkan proses normalisasi untuk menghilangkan adanya atribut ganda pada entitas , redundansi serta anomali yang mungkin terjadi pada *database*. Pembuatan *database Online Shop* Buku Dokter ini menggunakan MySQL sebagai DBMS dalam memanajemen informasi dalam *database*. Peneliti hanya membuat pencatatan transaksi penjualan pada *Online Shop* Buku Dokter dengan output laporan transaksi dan pengiriman barang tanpa adanya laporan keuangan , sehingga perancangan ini sebagai saranan pemilik *Online Shop* Buku Dokter dalam memonitor transaksi pada penjualan dan transaksi pengiriman.

**Keyword :** Online Shop, Basis Data, My SQL

## I. PENDAHULUAN

### 1.1 Latar Belakang

Pada era yang semakin berkembang ini kemajuan teknologi dan cepatnya penyebaran informasi menuntut pelayanan yang serba efisien dan efektif . Pelayanan serba cepat ini dapat terlaksana jika seluruh data yang dibutuhkan tersebut terkumpul, tersusun, dan terorganisir dalam suatu basis data (*database*) yang dapat diakses menurut keperluan kapan saja diperlukan. Dalam era yang secanggih ini segala sesuatu yang kita kerjakan secara tidak langsung berkaitan dengan data dan informasi. Dalam segala bidang *database* di perlukan

agar informasi dapat diakses dengan mudah contohnya bidang pendidikan, pemerintahan, kesehatan serta bisnis mereka semua memerlukan database agar dapat bekerja dengan cepat, efektif dan efisien. Bahkan beberapa lembaga tidak akan berfungsi bila tidak didukung oleh data dan informasi misalkan pemerintahan, bank serta mediamasa. Basis data merupakan koleksi data yang terintegrasi, diorganisasikan dan disimpan dengan suatu alat dan cara yang memudahkan pengambilannya kembali. Dengan adanya sistem basis data, memudahkan dalam mencari informasi (kumpulan data yang telah mengalami proses). Data-data tersebut diatur/diorganisir secara tepat, sehingga dapat dengan cepat memperoleh informasi dari susunan data yang diinginkan.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang di atas, penulis dapat merumuskan permasalahan sebagai berikut :

**"Bagaimana merancang database sistem penjualan Onlineshop Buku Dokter" ,** sehingga menghasilkan informasi yang cepat, tepat, akurat dan relevan.

## **1.3. Batasan Masalah**

Dengan memperhatikan permasalahan yang ada maka guna memudahkan pembahasan dalam penulisan , pembatasan masalah dalam hal ini dibatasi pada :

- a. Proses pendataan meliputi pendataan nama barang, harga, stok, jabatan dan gaji karyawan.
- b. Software yang digunakan adalah Microsoft Word, XAMPP,SQLyog , Diagram Designer, dan NaviCat Premium.

## **1.4. Tujuan Penelitian**

Tujuan yang ingin dicapai dalam pembuatan penelitian ini adalah :

"Menghasilkan rancangan database sistem informasi OnlineShop Buku Dokter yang mampu memberikan informasi dengan cepat, akurat dan relevan."

### 1.5. Manfaat Penelitian

Manfaat diadakannya penelitian ini adalah mempermudah “Online Shop Buku Dokter” dalam memanajemen barang serta transaksi penjualan yang terjadi pada Online Shop Buku Dokter.

## II. METODE PENELITIAN

### A. Jenis Penelitian :

Studi Kasus : eksplorasi mendalam dari sistem terkait (misalnya, kegiatan, acara, proses, individu) berdasarkan pengumpulan data yang luas.

### B. Waktu dan Tempat :Rumah Owner Buku Dokter (Jl. Salemba Tengah II No.10, RT.5 / RW.8, Paseban, Senen jakarta pusat, Daerah Khusus Ibukota Jakarta,10440).

### C. Subyek : Online Shop Buku Dokter (JAKARTA).

### D. Teknik Pengumpulan Data :menggunakan tehnik Wawancara dengan pemilik / Owner Buku Dokter dan Observasi / mengamati langsung.

## III. DASAR TEORI

Data merupakan Fakta-fakta mentah yang mewakili kejadian-kejadian yang berlangsung dalam organisasi atau lingkungan fisik sebelum ditata dan diatur ke dalam bentuk yang dapat dipahami dan digunakan orang.Laudon dan Laudon (1998).

Informasi merupakan Data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.Davis (1999).

Pengertian sistem informasi adalah “Kombinasi antara prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi”. (Alter, 1992).

Database merupakan Kumpulan data yang umumnya menjabarkan aktivitas-aktivitas dari satu atau lebih dari satu organisasi yang terkait.Ramakrishnan dan Gerke, 2000.

DBMS atau *Database Management System* adalah sebuah perangkat lunak yang bertugas untuk mempermudah dalam memanajemen data seperti membaca (*Read*), menulis (*Create*)

,memperbarui (*Update*) , serta menghapus data (*Delete*) atau dengan singkat CRUD (*Create Read Update Delete*) .Macam macam DBMS seperti Oracle, MySQL, SQL server dan sebagainya.

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, di mana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, di mana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

Entitas menunjukan obyek-obyek dasar yang terkait didalam sistem. Obyek dasar dapat berupa orang, benda, atau hanya keterangan yang perlu disimpan di dalam *database* .Dalam *database* entitas bias juga disebut sebuah *table*.

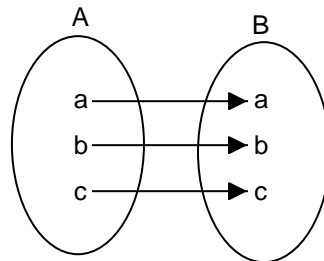
Atribut sering pula disebut sebagai *property*, merupakan keterangan keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam *database*. Atribut dalam *database* adalah sebuah *field*.

ER (*Entitiy Relationship*) atau Relasi antar Entitas adalah hubungan antar dua buah entitas. Dimana hubungan atau relasi yang terjadi merupakan sebuah kejadian atau transaksi diantara dua buah entitas tersebut. Kejadian atau Transaksi tersebut bias disimpan dalam *database* atau bias juga tidak disimpan walaupun benar benar terjadi. Dalam ER terdapat bermacam macam relasi antara lain :

1. *One to One*(1:1)

*One to One* mempunyai arti bahwa satu entitas pertama hanya membutuhkan satu nilai dari entitas kedua dan hanya muncul satu kali dan berlaku sebaliknya untuk entitas yang lain.

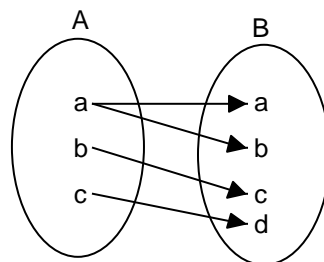
Sehingga tidak ada kemunculan data yang sama dalam entitas pertama maupun kedua. Contoh: Satu Karyawan hanya Satu Mesin kasir berlaku sebaliknya..



Gambar 1 Ilustrasi One to One

## 2. *One to Many*(1:N)

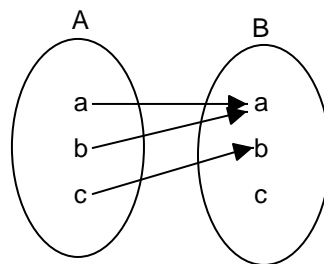
*One to Many* merupakan relasi antar entitas satu dan entitas dua dimana entitas satu dengan satu nilai tetapi entitas dua mengambil nilai entitas satu untuk banyak nilai. Contoh: Satu mahasiswa mengambil banyak matakuliah.



Gambar 2 Ilustrasi One to Many

## 3. *Many to One*(N:1)

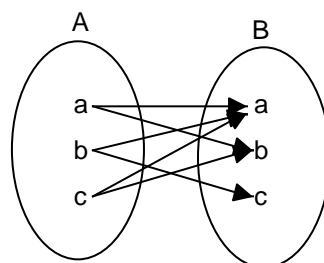
*Many to One* merupakan relasi antar entitas dimana entitas satu memiliki banyak nilai yang berdasar dari satu nilai pada entitas dua. Contoh : Banyak Matakuliah dipilih oleh satu mahasiswa..



Gambar 3 Ilustrasi Many to one

#### 4. *Many to Many*(M:N)

*Many to Many* merupakan relasi antar entitas satu dan entitas dua dimana keduanya saling mengambil banyak nilai. Contoh : banyak Mahasiswa memilih banyak matakuliah.



Gambar 4 Ilustrasi Many to Many

Macam – macam ketergantungan data :

1. Dependensi fungsional
2. Dependensi fungsional penuh
3. Dependensi total
4. Dependensi Transitif

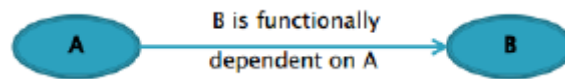
#### 1. **Dependensi fungsional**

Ketergantungan Fungsional (functional dependency) sering disingkat dengan KF. Kata dependency sendiri berarti bergantung, ya memang ketergantungan itu yang akan kita bahas.

Functional dependency (ketergantungan fungsional) menggambarkan hubungan kaitan antara atribut-atribut dalam relasi, dikatakan fungsional dependent karena nilai dari suatu atribut digunakan oleh atribut yang lain. Simbol untuk menyatakan functional dependent adalah  $\Rightarrow$  dibaca secara fungsional menentukan.

Contoh :  $A \Rightarrow B$  (A secara fungsional menentukan B)

Artinya bahwa A secara fungsional menentukan B atau B bergantung pada A



Gambar 5 Ilustrasi Ketergantungan Fungsional

Contoh :

Tabel Nilai

MataKuliah	NIM	NamaMhs	NilaiHuruf
Struktur Data	980001	Ali Akbar	A
Struktur Data	980004	Indah Susanti	B
Basis Data	980001	Ali Akbar	A
Basis Data	980002	Budi Haryanto	B

Gambar 6 Contoh Ketergantungan Fungsional

Ketergantungan fungsional dari tabel Nilai di atas adalah :

$NIM \Rightarrow NamaMhs$

NIM menentukan Nama Mahasiswa karena NIM yang sama menunjukkan nama Mahasiswa yang sama.

$\{Matakuliah, NIM\} \Rightarrow NilaiHuruf$

Kenapa harus Matakuliah dan NIM kok tidak NIM saja atau Matakuliah saja. Karena kalau hanya NIM saja satu NIM dapat memiliki 2 nilai, Jika Matakuliah saja juga terdapat banyak nilai, oleh karena itu kita menggunakan Matakuliah dan NIM untuk menentukan Nilai Huruf.

#### MACAM-MACAM FUNCTIONAL DEPENDENCY

1. Full functional dependency
2. Partially dependency
3. Transitive dependency

## FULL DEPENDENCY

Ketergantungan penuh. Full dependency menunjukkan atribut A dan B dalam satu relasi, dimana B bergantung penuh terhadap A, maksudnya penuh disini adalah B hanya ditentukan oleh A dan hanya A saja yang menentukan B bukan subset dari A seperti contoh sebelumnya.

Contoh :

NIM	Nama	Ipk	idRuang	Dosen
A11.2078	Paijo	3,20	301	Agus C
A11.2906	Tukijo	2,02	602	Joko P
A12.3456	Parjo	3,09	302	Rima U
A11.3254	Ngatini	3.40	301	Santo M
A12.3098	Tumini	2,75	602	Setyo P

Gambar 7 Contoh Ketergantungan Fungsional Utuh

Dikatakan full dependency kalau hanya A yang menentukan B dalam hal ini adalah NIM  $\Rightarrow$  id Ruang. Jika Nama dan NIM yang menentukan idRuang maka hal ini tidak dapat dikatakan sebagai full dependency.

## PARTIALLY DEPENDENCY

Ketergantungan parsial atau sebagian. Partionally dependency ini memiliki 2 atribut dari A untuk menentukan B, namun untuk menentukan B tidak harus 2 atribut artinya jika salah satu atribut A yang menentukan B dapat dihilangkan namun tidak merubah arti relasi dan masih tetap berelasi ketergantungan.

Contoh :

NIM	Nama	Ipk	idRuang	Dosen
A11.2078	Paijo	3,20	301	Agus C
A11.2906	Tukijo	2,02	602	Joko P
A12.3456	Parjo	3,09	302	Rima U
A11.3254	Ngatini	3.40	301	Santo M
A12.3098	Tumini	2,75	602	Setyo P

Gambar 8 Contoh Ketergantungan Parsial

Ketergantungan sebagian  $\{NIM, Nama\} \Rightarrow idRuang$  dimana jika Nama dihilangkan maka ketergantungan tetap ada.



## TRANSITIVE DEPENDENCY

Transitive dependency biasanya terjadi pada tabel hasil relasi, atau kondisi dimana terdapat tiga atribut A,B,C. Kondisinya adalah  $A \Rightarrow B$  dan  $B \Rightarrow C$ . Maka C dikatakan sebagai transitive dependency terhadap A melalui B.

Contoh :

NIP	Nama	Jabatan	Gaji	kdCabang	almCabang
10.002.1	Paio	Quality Control	1500	01	Semarang
11.089.2	Tukijo	Supervisor	1250	02	Kendal
11.098.1	Parjo	Quality Control	1500	01	Semarang
12.101.3	Ngatini	Cleaning Service	750	03	Tengaran
12.110.1	Tumini	Quality Control	1500	01	Semarang

Gambar 9 Contoh Ketergantungan Transitif

NIP  $\Rightarrow$  {Nama, Jabatan, Gaji, kdCabang, almCabang}

kdCabang  $\Rightarrow$  almCabang

NIP menentukan semua atribut dan atribut almCabang ditentukan oleh NIP namun melalui kdCabang..

Dalam modifikasi data memungkinkan terjadinya penyimpangan (*Anomally*). Terjadinya penyimpangan bisa di kategorikan menjadi tiga antara lain :

1. Penyimpangan saat penghapusan (*Delete Anomally*)

Hilangnya informasi yang rinci saat penghapusan data sehingga akan mengakibatkan penyimpangan..

2. Penyimpangan saat penyisipan (*Insert Anomally*)

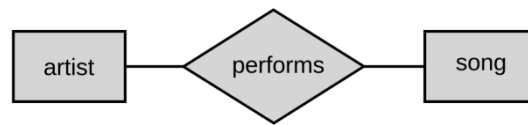
Proses menyisipkan suatu nilai rinci data yang mengakibatkan perlunya penyisipan pada nilai rinci data lain yang tidak mempunyai kerelasian secara logis.

3. Penyimpangan saat pembaruan (*Update Anomally*)

Penyimpangan akibat suatu nilai rinci data diubah dan mengakibatkan nilai rinci yang lain perlu di ubah juga.

Ketergantungan Data (*Data Dependency*) adalah ketergantungan antar atribut atau rinci data entitas satu dengan entitas lainnya. Ada macam macam ketergantungan data antara lain :

ERD (*Entity Relationship Diagram*) adalah diagram yang menggambarkan bentuk relasi entitas (ER) dalam *database*.



Gambar 10 Simple ERD (sumber : <http://bit.ly/2tPEKlb> dengan sedikit modifikasi)

Pada *Gambar 5* di atas dapat kita lihat hubungan entitas *artist* dengan *song* dimana mempunyai relasi *performs*.

Normalisasi adalah proses pormalan database yang berguna untuk memperkecil redudansi, menghilangkan atribut ganda atau menghilangkan anomali pada table. (Simarmata, 2007).

Terdapat enam bentuk dalam normalisasi antara lain:

#### 1. Bentuk Normal Pertama (1NF).

Bentuk Normal Pertama berfungsi menghilangkan atribut bernilai ganda.

Normal 1NF

NIM	Nama	Kode_Matkul
201010241	Siti	001
201010241	Siti	004
201010241	Siti	008
201010242	Beti	002
201010242	Beti	005
201010243	Eli	005
201010243	Eli	008

Gambar 11 Ilustrasi 1NF

#### 2. Bentuk Normal Kedua (2NF).

Untuk mencapai bentuk 2NF, syarat yang harus dipenuhi adalah mencapai bentuk 1NF. Pada 2NF hal yang kita normalisasikan adalah atribut yang memiliki dependensi parsial.

Normal 2NF

<u>NIM</u>	Nama
201010241	Siti
201010242	Beti
201010243	Eli
201010244	Rosi
201010245	Eci

Gambar 12 Ilustrasi 2NF

<u>NIM</u>	Kode_Matkul
201010241	001
201010241	004
201010241	008
201010242	002
201010242	005
201010243	005
201010243	008
201010244	003
201010244	006
201010245	001
201010245	007
201010245	009

Gambar 13 Ilustrasi 2NF

### 3. Bentuk Normal Ketiga (3NF).

Untuk mencapai bentuk 3NF , syarat yang haru di penuhi adalah mencapai bentuk 2NF. Pada 3NF atribut yang dinormalisasikan adalah atribut yang berdependensi transitif.

3NF

<u>NIM</u>	Nama
201010241	Siti
201010242	Beti
201010243	Eli
201010244	Rosi
201010245	Eci

Gambar 14 Ilustrasi 3NF

<u>Nama</u>	Kode_Matkul
Siti	001
Siti	004
Siti	008
Beti	002
Beti	005
Eli	005
Eli	008
Rosi	003
Rosi	006
Eci	001
Eci	007
Eci	009

Gambar 15 Ilustrasi 3NF

#### 4. Bentuk Normal Boyce Codd (BCNF).

Bentuk BCNF berguna untuk memperbaiki bentuk 3NF yang masih memungkinkan adanya anomaly.

Normal BCNF

<u>NIM</u>	Nama
201010241	Siti
201010242	Beti
201010243	Eli
201010244	Rosi
201010245	Eci

Gambar 16 Ilustrasi BCNF

<u>Nama</u>	Kode_Matkul
Siti	001
Siti	004
Siti	008
Beti	002
Beti	005
Eli	005
Eli	008
Rosi	003
Rosi	006
Eci	001
Eci	007
Eci	009

Gambar 17 Ilustrasi BCNF

## 5. Bentuk Normal Keempat (4NF).

Bentuk keempat ini tercapai jika sudah memenuhi kriteria BCNF. Bisa dikatakan memenuhi kriteria 4NF jika setiap atribut yang memiliki ketergantungan pada banyak nilai adalah bergantung secara fungsional (*functionally dependency*).

## 6. Bentuk Normal Kelima (5NF).

Untuk mencapai bentuk ke 5 maka harus sudah memenuhi kriteria 4NF dan kerelasian antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang sederhana.

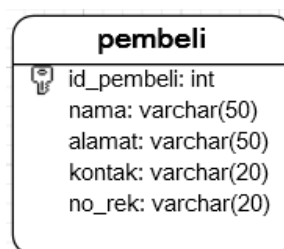
Efek samping dengan menggunakan normalisasi antara lain:

1. Proses dekomposisi relasi akan mengakibatkan munculnyaduplikasi kerincian data pada atribut kunci tamu (*foreign key*).
2. Dekomposisi relasi membuka kemungkinan tidak terpenuhinya intergritas referensial dalam *database*.
3. Dekomposisi relasi menghasilkan semakin banyak jumlah relasi baru sehingga mengakibatkan inefisiensi proses menampilkan kembali data data dari dalam basis data.
4. Adanya batasan penerapan pada beberapa DBMS untuk ukuran komputer pribadi / PC (*Personal Computer*), terutama berkaitan dengan batas maksimal relasi yang dapat dibuka secara bersamaan.

## 1.8. Perancangan


### A. Tabel dan ERD

Tabel Pembeli




Gambar 18 Tabel Pembeli

Tabel Transaksi

transaksi	
	id_transaksi: int
	tanggal: datetime
	id_pembli: int
	id_karyawan: int
	keterangan: varchar(100)
	status: varchar(50)


Gambar 19 Tabel Transaksi

Tabel Data Angsuran

data_angsuran	
	id_angsur: int
	tanggal_waktu: datetime
	id_transaksi: int
	id_kry: int
	jmlh_angs: double
	sumber_trans: varchar(30)


Gambar 20 Tabel Data Angsuran

Tabel Detail Transaksi

detail_transaksi	
	id_detail_transaksi: int
	id_transaksi: int
	id_barang: int
	jumlah_brg: int
	total_harga: double


Gambar 21 Tabel Detail Transaksi

Tabel Pengiriman

prngiriman	
	id_pengiriman: int
	tanggal: datetime
	id_transaksi: int
	id_karyawan: int
	jasa_kurir: varchar(20)
	keterangan: varchar(100)
	status: varchar(50)


Gambar 22 Tabel Pengiriman

TabelBarang

barang	
	id_barang: int
	nama: varchar(50)
	kategori: varchar(50)
	stok: int
	harga: double

Gambar 23 Tabel Barangs


Tabel Karyawan

karyawan	
	id_karyawan: int
	nama: varchar(50)
	alamat: varchar(50)
	tempat_lahir: varchar(20)
	tgl_lahir: date
	kontak: varchar(20)
	jenis_kelamin: varchar(20)
	bonus: double
	id_jabatan: varchar(20)

Gambar 24 Tabel Karyawan




### Tabel Histori Detail Transaksi

<b>his_trans_det</b>	
	id_his_td: int
	sumber: varchar(20)
	tgl_edit: datetime
	idtrans: int
	iddet: int
	idbrg: int
	jml: int
	tothrg: double


Gambar 25 Tabel Riwayat Transaksi Detail

### Tabel Jabatan

<b>jabatan</b>	
	id_jabatan: varchar(20)
	jabatan: varchar(50)
	gaji: double


Gambar 26 Tabel Jabatan

### Tabel Histori Barang

<b>his_barang</b>	
	id_his_brg: int
	tgl_edit: datetime
	idbrg: int
	source: varchar(20)
	nem: varchar(20)
	kat: varchar(20)
	stok: int
	hrg: double


Gambar 27 Tabel Riwayat Barang

Tabel Histori Angsuran

his_ang	
	id_his_angs: int
	sumber: varchar(20)
	tgl_edit: datetime
	idang: int
	idtrans: int
	idkar: int
	jmlh: varchar(100)
	pembayaran: varchar(30)

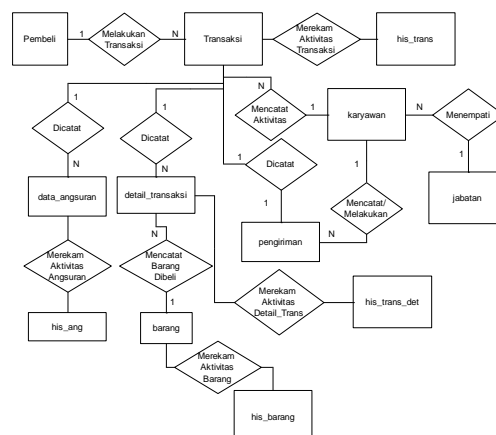
Gambar 28 Tabel Riwayat Angsuran

Tabel Histori Transaksi

his_trans	
	id_his_trans: int
	sumber: varchar(20)
	tgl_edit: datetime
	idtrans: int
	idpembli: int
	idkar: int
	ket: varchar(100)
	status: varchar(20)

Gambar 29 Tabel Riwayat Transaksi

Relasi antar Entitas



Gambar 30 ERD pada Basisdata penjualan Buku Dokter

Pada gambar bisa dilihat masing masing entitas mewakili tabel yang telah di tampilkan sebelumnya.

## B. Normalisasi

Berdasarkan ERD pada gambar maka kita akan membuat normalisasi dari database yang akan di buat.

### a. Bentuk Belum ter-normalisasi (UNF)

Bentuk UNF ini merupakan bentuk di mana tabel belum ter-normalisasi dan masih terdapat atribut bernilai ganda.

ID TR AN S	TGL TR ANS	NAMA PEM BELI	NAMA KARYAWAN	BARANG DIBELI	Jumlah	Harga Satuan	STATUS	ANGSURAN	HARGA TOTAL	KETERANGAN
i	2017-01-01	Zulfikar	Anita	1.Buku Pedoman Dokter Gigi 2.Jas Dokter	1)350 2)100	1)140 2)100	Belum Lunas	50000	59000	COD Kredit

Gambar 31 Tabel Belum Ternormalisasi

Dilihat pada *gambar 31* masih ada nilai bernilai ganda pada atribut barang dibeli , jumlah , dan harga satuan. Maka dari itu perlu di adakan normalisasi pertama.

### b. Bentuk Normalisasi Pertama (1 NF)

Bentuk Normalisasi Pertama akan menghilangkan atribut bernilai ganda.

ID TR AN S	TGL TR ANS	NAMA PEM BELI	NAMA KARYAWAN	BARANG DIBELI	Jumlah	Harga Satuan	STATUS	ANGSURAN	HARGA TOTAL	KETERANGAN
i	2017-01-01	Zulfikar	Anita	1.Buku Pedoman Dokter Gigi	1)350	1)140	Belum Lunas	50000	59000	COD Kredit
ii	2017-01-01	Zulfikar	Anita	2.Jas Dokter	2)100	2)100	Belum Lunas	50000	59000	COD Kredit

Gambar 32 Tabel Transaksi Bentuk 1NF

Dilihat pada *gambar 32* pada atribut barang dibeli , jumlah , dan harga satuan sudah dipisahkan ke dalam baris baru, tetapi masih dengan id transaksi yang sama dimana hal ini tidak boleh terjadi karena kunci utama yang sama dan juga pada normalisasi pertama ini masih ada ketergantungan parsial. Maka pada Normalisasi kedua kita akan menghilangkan ketergantungan parsial dan redundansi pada kunci utama.

### c. Bentuk Normalisasi Kedua (2NF)

Bentuk Normalisasi Kedua akan menghilangkan ketergantungan parsial serta redundansi.

Tabel Transaksi Detail

ID Detail	ID Trans	BARANG_DIBELI	Jumlah	Harga_Satuan	harga_total
1	1	Buku Pedoman Dokter Gigi	350	140	49000
2	1	Jas Dokter	100	100	10000

Gambar 33 Tabel Transaksi Detail bentuk 2NF

Tabel Transaksi Detail merupakan normalisasi dari *gambar 32* dimana id transaksi tetap tetapi ditambahi dengan id detail untuk mengatasi redudansi pada kunci utama.

Tabel Transaksi

ID TRANS	TGL_TRANS	NAMA_PEMBELI	NAMA_KARYAWAN	HARGA_TOTAL	KETERANGAN
1	2017-01-01	1	1	59000	COD Kredit

Gambar 34 Tabel Transaksi bentuk 2NF

Pada Tabel Transaksi ini merupakan normalisasi *gambar* untuk mewakili transaksi per-pembeli , sehingga pembeli bisa membeli lebih dari satu id barang , tetapi dengan 1 (satu) id transaksi saja. Sedangkan id barang yang banyak tersebut disimpan pada Tabel Detail Transaksi dengan memasukan atribut id transaksi sebagai kunci tamu.

Tabel Pembeli

ID Pembeli	Nama	alamat	Kontak	No.Rek
1	Zulfikar	Jl. Sido Muncul	081325009896	028012345

Gambar 35 Tabel Pembeli bentuk 2NF

Pada Tabel Pembeli ini merupakan pemisahan pada *gambar 32* sehingga menghilangkan ketergantungan parsial pada tabel transaksi

Tabel Karyawan

ID	Nama	Alamat	Tempat_Lahir	Tgl_Lahir	Kontak	Gender	Jabatan	Gaji	Bonus
1	Anita	Jl. Podo Moro	Jakarta	1989-08-17	081324567345	Perempuan	Sales	2	-

Gambar 36 Tabel Karyawan bentuk 2NF

Pada Tabel Karyawan ini merupakan pemisahan pada *gambar 32* untuk menghilangkan ketergantungan parsial. Walaupun sudah terlihat baik pada bentuk normal kedua ini, tetapi masih ada ketergantungan transitif pada tabel transaksi detail dan tabel karyawan.

#### d. Bentuk Normalisasi Ketiga (3NF)

Bentuk Normalisasi Ketiga ini akan menghilangkan ketergantungan transitif pada tabel transaksi detail dan tabel karyawan dengan memindahkan atribut yang memiliki ketergantungan transitif ke tabel baru.

Tabel Transaksi Detail

ID Detail	ID Trans	BARANG_DIBELI	Jumlah	harga_total
1	1	1	350	49000
2	1	2	100	10000

Gambar 37 Tabel Transaksi Detail bentuk 3NF

Pada *gambar 33* di normalisasi kedua barang dibeli dan harga satuan memiliki ketergantungan transitif dimana harga satuan tergantung pada barang yang dibeli, sehingga kita perlu memindahkannya menjadi tabel baru dan pada tabel transaksi detail atribut barang dibeli diganti dengan id barang.

Tabel Barang

ID Detail	BARANG_DIBELI	Stok	Harga_Satuan
1	Buku Pedoman Dokter Gigi	400	140
2	Jas Dokter	500	100

Gambar 38 Tabel Barang hasil normalisasi 3NF

Tabel Barang ini merupakan pembuatan tabel untuk menghilangkan ketergantungan transitif.

Tabel Karyawan

ID	Nama	Alamat	Tempat_Lahir	Tgl_Lahir	Kontak	Gender	Jabatan	Bonus
1	Anita	Jl.Podo Moro	Jakarta	1989-08-17	081324567345	Perempuan	JOB004	-

Gambar 39 Tabel Karyawan bentuk 3NF

Pada tabel karyawan ini atribut jabatan dan gaji memiliki ketergantungan transitif karena gaji tergantung pada jabatan.

Tabel Jabatan

ID	Jabatan	Gaji
JOB004	Sales	2

Gambar 40 Tabel Jabatan hasil normalisasi 3NF

Tabel jabatan ini merupakan hasil pemindahan atribut jabatan dan gaji kedalam tabel

## IV. PENUTUP

### 4.1 Kesimpulan

Dari Perancangan Basisdata “Onlineshop Buku Dokter ” Dengan Menggunakan MySQL ini kami menarik beberapa kesimpulan antara lain :

- Dengan adanya rancangan *database* ini maka akan mempermudah pengimplementasian dalam pembuatan perangkat lunaknya.
- Dengan adanya normalisasi akan mengurangi redudansi dan anomali yang terdapat pada entitas entitas.
- Dengan adanya *database* ini , proses bisnis pada Online Shop Buku Dokter menjadi lebih cepat dan penyimpanan data serta informasi yang akurat dan tepat.

### 4.2 Saran

- Perlu diadakan pelatihan dalam penggunaan *database* ini jika digunakan secara langsung.
- Perlunya adanya perangkat lunak atau aplikasi dalam pengoperasian *database* ini sehingga mempermudah pengaksesan.

- c. Memiliki *hosting* atau *server* sendiri atau menyewa jasa *hosting* untuk meningkatkan produktifitas.

## V. DAFTAR PUSTAKA

- [1] Ketergantungan Fungsional. (t.thn.). Dipetik Juli 15, 2017, dari Wordpress: <https://pujirahayu2004l.wordpress.com/2016/01/06/ketergantungan-fungsional-kf/>
- [2] Normalisasi 1NF,2NF,3NF dan BCNF. (t.thn.). Dipetik Juli 15, 2017, dari Wordpress:
  - <https://rachmanmarangga.wordpress.com/2012/01/09/normalisasi-tabel-1nf-2nf-3nf-dan-bcnf/>
- [3] Point, T. (t.thn.). DBMS Normalization. Dipetik Juni 28, 2017, dari Tutorials Point: [https://www.tutorialspoint.com/dbms/database\\_normalization.htm](https://www.tutorialspoint.com/dbms/database_normalization.htm)
- [4] Simarmata, J. (2007). Perancangan Basis Data. Yogyakarta: Penerbit Andi.
- [5] Sutanta, E. (2011). Basis Data dalam Tinjauan Konseptual. Yogyakarta: Penerbit Andi.
- [6] Wikipedia. (t.thn.). MySQL. Dipetik Juli 15, 2017, dari Wikipedia: <https://id.wikipedia.org/wiki/MySQL>