



**ANALISIS DAN PERANCANGAN SISTEM BASIS DATA OPTIK PADA
RUMAH SAKIT SRIWIJAYA EYE CENTER PALEMBANG**

SKRIPSI

Diajukan guna mengikuti ujian komprehensif

OLEH :

CHANDRA ARDIANSYAH

09.142.020

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BINA DARMA

PALEMBANG

2013

**ANALISIS DAN PERANCANGAN SISTEM BASIS DATA OPTIK PADA
RUMAH SAKIT SRIWIJAYA *EYE* CENTER PALEMBANG**

OLEH :

CHANDRA ARDIANSYAH

09.142.020

SKRIPSI

Salah satu syarat untuk mengikuti ujian komprehensif

Disetujui,

Dosen Pembimbing I

Alex Wijaya, M.IT

Dosen Pembimbing II

Evi Yulianingsih, MM., M.Kom.

Program Studi Teknik Informatika

Universitas Bina Darma,

Ketua,

Syahril Rizal, S.T.,M.M.,M.Kom.

ABSTRAK

Di era *globalisasi* saat ini, perkembangan teknologi telah mengalami kemajuan yang sangat pesat, tentu saja hal ini berdampak pada kehidupan terutama di dunia bisnis. Perkembangan teknologi dimanfaatkan oleh para pelaku bisnis dalam menjadikan perusahaan yang dikelolanya menjadi lebih baik. Sehingga perusahaan tersebut siap dalam menghadapi persaingan di dunia bisnis yang semakin ketat. Data merupakan komponen vital bagi suatu perusahaan untuk melakukan kegiatan operasionalnya dan menentukan kualitas informasi yang dihasilkan. Banyaknya sumber data dari kegiatan operasional yang disimpan pada tempat terpisah dapat menyebabkan terjadinya duplikasi data. Duplikasi data dapat menyebabkan ketidakkonsistenan data dan berujung pada informasi yang dihasilkan menjadi tidak akurat. Sistem yang akan dilakukan yaitu merancang basis data yang baik dan benar sehingga data menjadi lebih konsisten dan terintegrasi antarbagian, dan membuat data yang tersimpan menjadi lebih aman dan nantinya rancangan basis data yang dibuat ini dapat dimanfaatkan dalam pembuatan aplikasi optik pada perusahaan rumah sakit.

Kata Kunci : Analisis, Perancangan, Sistem, Basis data.

KATA PENGANTAR



Puji dan syukur atas segala nikmat yang diberikan oleh Allah SWT yang selalu memberikan berkah, rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan proposal skripsi yang berjudul Analisis dan Perancangan Sistem Basis Data Optik Pada Rumah Sakit Sriwijaya *Eye Center* Palembang.

Proposal ini disusun sebagai salah satu syarat untuk dilanjutkan menjadi skripsi sebagai proses akhir dalam menyelesaikan pendidikan strata satu (S1). Dalam penulisan proposal skripsi ini penulis telah berusaha semaksimal mungkin memberikan dan menyajikan yang terbaik. Tetapi penulis juga menyadari bahwa proposal skripsi ini masih jauh dari sempurna, hal ini dikarenakan terbatasnya pengetahuan yang dimiliki penulis. Oleh karena itu, penulis mengharapkan saran dan kritik yang bersifat membangun untuk kesempurnaan proposal skripsi ini.

Pada kesempatan ini, tidak lupa penulis mengucapkan terimah kasih kepada semua pihak yang telah memberikan bimbingan, pengarahan, nasihat, dan pemikiran dalam menyelesaikan proposal skripsi ini, terutama kepada :

1. Prof. Ir. Bochari Rahman, M.Sc. selaku Rektor Universitas Bina Darma Palembang.
2. M. Izman Herdiyansyah, S.T., M.M., Ph.D. Selaku Dekan Fakultas Ilmu Komputer Universitas Bina Darma Palembang.
3. Syaril Rizal, S.T.,M.M.,M.Kom. Selaku Kepala Program Studi Teknik

Informatika Universitas Bina Darma Palembang.

4. Alex Wijaya, M.IT. Selaku pembimbing I yang telah memberikan bimbingan dan arahan dalam penulisan proposal skripsi.
5. Evi Yulianingsih, MM., M.Kom. Selaku pembimbing II yang telah memberikan bimbingan dan arahan dalam penulisan proposal skripsi ini.
6. Orang tua tercinta yg selalu memotivasi, saudara-saudaraku, seluruh teman-teman dan pihak yang tidak bisa disebutkan yang telah memberikan bantuan baik materi maupun spiritual yang tak ternilai harganya.

Palembang, 29 Juli 2013

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
ABSTRAK.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	x

I. PENDAHULUAN

1.1. Latar Belakang	1
1.2. Perumusan Masalah	2
1.3. Batasan Masalah	3
1.4. Tujuan dan Manfaat	
1.4.1. Tujuan	3
1.4.2. Manfaat	3
1.5. Metode Penelitian	4
1.5.1. Metode Perancangan Basis Data	4
1.6. Metode Pengumpulan Data	5

II. TINJAUAN PUSTAKA

2.1. Sistem Basis Data	6
2.1.1 Keuntungan Sistem Basis Data	6
2.1.2 Basis Data	7
2.1.3 <i>Structured Query Language (SQL)</i>	8
2.1.4 <i>Entity Relationship Modeling</i>	9
2.1.4.1 Tipe Entitas	9
2.1.4.2 Tipe Hubungan (<i>Relationship</i>).....	10
2.1.4.3 Tipe Atribut	11
2.1.4.4 Kunci (<i>Key</i>)	12
2.1.5 <i>Normalisasi</i>	12
2.1.6 Siklus Perancangan Basis Data	14
2.1.7 Perancangan Basis Data	12
2.2. Penelitian Sebelumnya	18

III. ANALISIS DAN PERANCANGAN

3.1. Perencanaan Basis Data (<i>Database Planning</i>)	20
3.2. Definisi Sistem (<i>System Definition</i>).....	21
3.3. Analisis dan Pengumpulan	21
3.3.1. Analisis sistem yang berjalan.....	21
3.3.1.1. Prosedur Penjualan	21
3.3.1.2. Prosedur Pembelian	23
3.3.2. Permasalahan Yang Dihadapi dan Solusi Pemecahan Masalah.....	26
3.4. Perancangan Basis Data (<i>Design Database</i>).....	26
3.4.1. Perancangan Basis Data Konseptual.....	26
3.4.1.1. Identifikasi Tipe Entitas	27
3.4.1.2. Identifikasi Tipe Relasi.....	28
3.4.1.3. Identifikasi Atribut dengan Tipe Entitas	28
3.4.1.4. Menentukan Domain Atribut	29
3.4.1.5. Menentukan Atribut <i>Candidate</i> dan <i>Primary Key</i>	31
3.4.2. Perancangan Basis Data Logikal	32
3.4.2.1. Menghilangkan Fitur yang Tidak Kompatibel dengan Model Relasional.....	32
3.4.2.2. Menurunkan relasi untuk Model Data <i>Logical Local</i>	34
3.4.3. Perancangan Basis Data <i>Fisikal</i>	53
3.4.3.1. Merancang Relasi Dasar.....	53
3.4.3.2. Memperkirakan kebutuhan <i>disk space</i>	57
3.4.3.3. Merancang Mekanisme Keamanan	59

IV. HASIL DAN PEMBAHASAN

4.1. Hasil.....	61
4.2. Pembahasan.....	62
4.2.1. Tabel Master	63
4.2.1.1. Tabel Kacamata.....	63
4.2.1.2. Tabel Satuan Kacamata.....	64
4.2.1.3. Tabel Pasien	64
4.2.1.4. Tabel Karyawan	65
4.2.1.5. Tabel Hak Akses.....	66
4.2.1.6. Tabel Pemasok	67
4.2.1.7. Tabel Kontak.....	68
4.2.1.8. Tabel Tipe Kontak	69
4.2.2. Tabel Penjualan	70
4.2.2.1. Tabel Order Penjualan.....	70
4.2.2.2. Tabel detail order penjualan	71
4.2.2.3. Tabel Status Bayar	72
4.2.2.5. Tabel Detail Pembayaran Penjuala.....	72
4.2.3. Tabel Pembelian	74
4.2.3.1. Tabel Order pembelian.....	4
4.2.3.2. Tabel Detail Order Pembelian.....	75

4.2.3.3. Tabel Pembayaran Pembelian	76
4.2.3.4. Tabel Detail Pembayaran Pembelian.....	77
4.2.4. <i>Trigger</i>	78
4.2.4.1. <i>Trigger</i> Pembelian_Kacamata	78
4.2.4.2. <i>Trigger</i> Penjualan_Kacamata.....	79

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan	80
5.2. Saran.....	80

DAFTAR TABEL

	Halaman
Tabel 4.1. Identifikasi Tipe Entitas	29
Tabel 4.2. Identifikasi Tipe Relasi	30
Tabel 4.3. Identifikasi Tipe Atribut Dengan Tipe Entitas.....	31
Tabel 4.4. Domain Atribut Pelanggan.....	32
Tabel 4.5. Domain Atribut Karyawan.....	32
Tabel 4.6. Domain Atribut Pemasok.....	33
Tabel 4.7. Domain Atribut Kacamata	33
Tabel 4.8. Domain Atribut <i>Order</i> Penjualan.....	33
Tabel 4.9. Domain Atribut Pembayaran Penjualan.....	33
Tabel 4.10. Domain Atribut <i>Order</i> Pembelian.....	34
Tabel 4.11. Domain Atribut Pembayaran Pembelian.....	34
Tabel 4.12. Atribute <i>Candidate Key</i> Dan <i>Primary Key</i>	35
Tabel 4.13. Relasi Untuk Model Data Logikal	44
Tabel 4.14. Struktur Tabel Karyawan Setelah Normalisasi	49
Tabel 4.15. Struktur Tabel Hakakses Setelah Normalisasi	49
Tabel 4.16. Struktur Tabel Pemasok Setelah Normalisasi	49
Tabel 4.17. Struktur Tabel Pelanggan Setelah Normalisasi.....	49
Tabel 4.18. Struktur Tabel Kontak Setelah Normalisasi	50
Tabel 4.19. Struktur Tabel Tipe Kontak Setelah Normalisasi	50
Tabel 4.20. Struktur Tabel Kacamata Setelah Normalisasi	50
Tabel 4.21. Struktur Tabel Satuan Kacamata Setelah Normalisasi	50
Tabel 4.22. Struktur Tabel <i>Order</i> Penjualan Setelah Normalisasi.....	50
Tabel 4.23. Struktur Tabel Detail <i>Order</i> Penjualan Setelah Normalisasi..	51
Tabel 4.24. Struktur Tabel Status Bayar Setelah Normalisasi	51
Tabel 4.25. Struktur Tabel Pembayaran Penjualan Setelah Normalisasi..	51
Tabel 4.26. Struktur Tabel Detail Pembayaran Penjualan Normalisasi....	51
Tabel 4.27. Struktur Tabel <i>Order</i> Pembelian Setelah Normalisasi.....	51
Tabel 4.28. Struktur Tabel Detail <i>Order</i> Pembelian Setelah Normalisasi	52
Tabel 4.29. Struktur Tabel Pembayaran Pembelian Setelah Normalisasi.	52
Tabel 4.30. Struktur Tabel Detail Pembayaran Pembelian Normalisasi...	52
Tabel 4.31. Ukuran Entitas	57
Tabel 4.32. Ukuran Entitas Per Bulan.....	58
Tabel 4.33. Rancangan Mekanisme Keamanan	59
Tabel 5.1. Struktur Tabel Kacamata	62
Tabel 5.2. Hasil Struktur Tabel Kacamata	62
Tabel 5.3. Struktur Tabel Satuan Kacamata.....	63
Tabel 5.4. Hasil Struktur Tabel Satuan Kacamata.....	63
Tabel 5.5. Struktur Tabel Pelanggan.....	64
Tabel 5.6. Hasil Struktur Tabel Pelanggan	64
Tabel 5.7. Struktur Tabel Karyawan	65
Tabel 5.8. Hasil Struktur Tabel Karyawan.....	65
Tabel 5.9. Struktur Tabel Hakakses	66
Tabel 5.10. Hasil Struktur Tabel Hakakses.....	66

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Siklus hidup pengembangan basis data.....	14
Gambar 2.2. Notasi <i>flowchart</i>	17
Gambar 3.1. Struktur organisasi perusahaan.....	22
Gambar 4.1. Definisi sistem.....	24
Gambar 4.2. <i>Flowchart</i> Penjualan	25
Gambar 4.3. <i>Flowchart</i> Pembelian	27
Gambar 4.4. Atribut <i>multi value</i> pada entitas karyawan.....	35
Gambar 4.5. Atribut <i>multi value</i> pada entitas Pelanggan.....	35
Gambar 4.6. Atribut <i>multi value</i> pada entitas Pemasok	36
Gambar 4.7. <i>Entity Relational Diagram</i> (ERD).....	44
Gambar 4.8. Normalisasi Pelanggan	45
Gambar 4.9. Normalisasi Karyawan	45
Gambar 4.10. Normalisasi Karyawan (Lanjutan)	46
Gambar 4.11. Normalisasi Pemasok	46
Gambar 4.12. Normalisasi Pemasok (Lanjutan)	47
Gambar 4.13. Normalisasi Kacamata.....	47
Gambar 4.14. Normalisasi <i>Order</i> Penjualan	48
Gambar 4.15. Normalisasi Pembayaran Penjualan	49
Gambar 4.16. Normalisasi <i>Order</i> Pembelian	49
Gambar 4.17. Normalisasi <i>Order</i> Pembelian (Lanjutan)	50
Gambar 4.18. Normalisasi Pembayaran Pembelian	50
Gambar 4.19. Normalisasi Pembayaran Pembelian (Lanjutan)	51
Gambar 5.1. <i>Database</i> penjualan_pembelian	64

BAB I

PENDAHULUAN

1.1 Latar belakang

Di era *globalisasi* saat ini, perkembangan teknologi telah mengalami kemajuan yang sangat pesat, tentu saja hal ini berdampak pada kehidupan terutama di dunia bisnis. Perkembangan teknologi dimanfaatkan oleh para pelaku bisnis dalam menjadikan perusahaan yang dikelolanya menjadi lebih baik. Sehingga perusahaan tersebut siap dalam menghadapi persaingan di dunia bisnis yang semakin ketat.

Data merupakan komponen vital bagi suatu perusahaan untuk melakukan kegiatan operasionalnya dan menentukan kualitas informasi yang dihasilkan. Banyaknya sumber data dari kegiatan operasional yang disimpan pada tempat terpisah dapat menyebabkan terjadinya duplikasi data. Duplikasi data dapat menyebabkan ketidakkonsistenan data dan berujung pada informasi yang dihasilkan menjadi tidak akurat.

Basis data merupakan kumpulan data yang saling berhubungan beserta dengan deskripsinya yang dapat di-*share* sesuai dengan keperluan masing-masing tingkatan penggunaannya yang digunakan untuk memenuhi kebutuhan perusahaan akan informasi. Basis data merupakan pusat media penyimpanan dan pengolahan

data dalam jumlah yang besar yang berasal dari tiap-tiap bagian pada perusahaan sehingga integritas data dapat terjaga dengan baik.

Rumah sakit Sriwijaya Eye Center merupakan perusahaan yang bergerak di bidang *Medical Eye*. Pada bagian optik untuk pengolahan sistem basis data yang lebih spesifik, sehingga belum terintegrasinya data yang berhubungan dengan transaksi seperti penjualan, pembelian dan stok. Pendataan optik masih disimpan dalam arsip-arsip. Oleh karena besar kemungkinan terjadinya duplikasi data, kurangnya *integritas* data, masalah keamanan, kesulitan dalam pencarian dan penyimpanan data.

Dari permasalahan yang telah diuraikan maka perlu dilakukan perancangan basis data dengan menggunakan aplikasi *MySQL* agar data menjadi lebih konsisten dan terintegrasi antarbagian data, dan membuat data yang tersimpan menjadi lebih aman dan nantinya rancangan basis data yang dibuat ini dapat dimanfaatkan dalam pembuatan aplikasi optik pada rumah sakit. Berdasarkan uraian diatas maka akan dilakukan penelitian dengan judul “**Analisis dan Perancangan Sistem Basis Data Optik Pada Rumah Sakit Sriwijaya Eye Center Palembang**”.

1.2 Perumusan Masalah

Berdasarkan latar belakang diatas maka permasalahan yang dapat diambil adalah “Bagaimana menganalisis dan merancang sistem basis data optik pada Rumah Sakit Sriwijaya Eye Center?”

1.2 Batasan Masalah

Dalam penelitian ini penulis membatasi permasalahan agar tetap terarah dan tidak menyimpang dari apa yang sudah direncanakan sebelumnya. Adapun batasan masalah dalam penelitian ini terdiri dari:

1. Pembuatan basis data yang dikhususkan pada perancangan basis data optik pada Rumah Sakit Sriwijaya *Eye Center*.
2. Melakukan analisis terhadap perancangan basis data optic.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan dari penelitian ini adalah untuk merancang basis data dikhususkan pada optik Rumah Sakit Sriwijaya *Eye Center*.

1.4.2 Manfaat

Adapun manfaat penelitian ini adalah sebagai berikut :

1. Basis data yang dirancang nantinya dapat dimanfaatkan untuk pembuatan sistem atau aplikasi optik agar data dapat terintegrasi.
2. Untuk menghindari duplikasi data dalam sebuah sistem.
3. Penelitian ini dapat menambah wawasan bagi pembaca dan dapat dipergunakan sebagai referensi untuk penelitian-penelitian selanjutnya.

1.5 Metode Penelitian

Dalam penelitian ini digunakan metode penelitian *deskriptif*. Penelitian *deskriptif* dapat diartikan sebagai proses pemecahan masalah yang diselidiki dengan menuliskan keadaan subyek dan obyek penelitian saat ini berdasarkan fakta-fakta yang tampak atau bagaimana adanya. (Sugiyono, 2005:21)

1.5.1 Metode Perancangan Basis Data

Dalam penelitian ini proses analisis dan perancangan basis data akan dilakukan dengan *siklus database life cycle* dengan tahapan-tahapan sebagai berikut (Indrajani, 2007:111):

1. *Database Planning*
2. *Definisi system*
3. *Requirement Collection and Analsys*
4. Perancangan Basis Data (*Design Database*)

Perancangan basis data secara fisik merupakan proses pemilihan struktur-struktur penyimpanan dan jalur-jalur akses pada file-file basis data untuk mencapai penampilan yang terbaik pada bermacam-macam aplikasi. Adapun perancangan basis data yaitu antara lain:

1. Perancangan Basis Data *Conceptual*
 - a. Mengidentifikasi tipe entitas
 - b. Mengidentifikasi tipe relasi
 - c. Mengidentifikasi atribut dengan tipe entitas
 - d. Menentukan domain atribut
 - e. Menentukan atribut *candiadte key* dan *primary key*

2. Perancangan Basis Data *Logical*
 - a. Menghilangkan fitur yang tidak kompetibel dengan model data relasional
 - b. Menurunkan relasi untuk model data logical local
 - c. Validasi menggunakan normalisasi
4. Perancangan Basis Data *Fisikal*
 - a. Melakukan relasi dasar
 - b. Estimasi kebutuhan *disk space*
 - c. Mendesain mekanisme keamanan

1.6 Metode Pengumpulan Data

Adapun metode pengumpulan data yang penulis terapkan yakni :

1. Observasi
Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan langsung ke Rumah Sakit Sriwijaya *Eye Center*.
2. Wawancara
Tahap ini untuk mengumpulkan data dengan cara mengadakan tanya jawab secara langsung kepada pegawai yang bersangkutan mengenai data-data yang dibutuhkan.
3. Studi Pustaka
Dengan cara mencari bahan yang berkaitan atau mendukung dalam penyelesaian masalah melalui buku-buku, majalah, dan internet yang erat kaitannya dengan masalah yang sedang dibahas.

BAB II

TINJAUAN PUSTAKA

Dalam bagian berikut ini akan dijelaskan definisi-definisi teori yang berhubungan dengan penelitian, tinjauan obyek penelitian dan alat bantu yang digunakan dalam penelitian.

2.1 Sistem Basisdata

Menurut Nugroho (2006:5) menyatakan Sistem Manajemen Basisdata (DBMS) adalah perangkat lunak yang didesain untuk membantu memelihara dan memanfaatkan kumpulan data yang besar. Kebutuhan akan sistem termasuk pula penggunaannya yang berkembang pesat. Alternatif sistem manajemen basisdata penggunaan adalah menyimpan data dalam *file* dan menulis kode aplikasi tertentu untuk mengaturnya.

2.1.1 Keuntungan Sistem Basis Data

Menurut Nugroho (2006:13) bahwa keuntungan sistem basis data yaitu :

1. Mengurangi pengulangan data, apabila dibandingkan dengan *file-file* komputer yang disimpan terpisah di setiap aplikasi *computer*, DBMS

mengurangi jumlah total *file* dengan menghapus data yang terduplikasi diberbagai *file*. Data terduplikasi selebihnya dapat ditempatkan dalam satu *file*.

2. Mencapai *independasi* data, spesifikasi data disimpan dalam skema pada tiap program aplikasi perubahan dapat dibuat struktur data tanpa memengaruhi program yang mengakses data.
3. Mengintegrasikan data, saat data dibentuk sehingga menyediakan kaitan logis, maka organisasi fisik bukan merupakan kendala. Organisasi logis, pandangan pengguna dan program aplikasi tidak harus tercermin pada media penyimpanan fisik.
4. Mengambil data dan informasi dengan cepat, hubungan-hubungan logis, bahasa manipulasi data, serta bahasa *query* memungkinkan pengguna mengambil data dalam hitungan detik atau menit.
5. Meningkatkan keamanan, DBMS *mainframe* maupun *computer mikro* dapat menyertakan beberapa lapis keamanan seperti kata sandi, direktori pemakai dan bahasa sandi sehingga data yang dikelola akan lebih aman.

2.1.2 Basis Data

Menurut Poerbanintyas (2000:58), adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basis data, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data

disimpan dalam basisdata menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan kedalam basisdata, dimodifikasi dan di hapus.

Menurut Simarmata (2009:1) Basis data adalah suatu aplikasi terpisah yang menyimpan suatu koleksi data. Masing-masing basis data memiliki satu API atau lebih yang berbeda untuk menciptakan, mengakses, mengelola, mencari dan mereplikasi data.

Basis data (*database*) adalah kumpulan dari berbagai data yang saling berhubungan satu dengan yang lainnya. Basis data tersimpan di perangkat keras, serta dimanipulasi dengan menggunakan perangkat lunak. Pendefinisian basis data meliputi spesifikasi dari tipe data, struktur dan batasan dari data atau informasi yang akan disimpan. Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi pada para pengguna atau user.

2.1.3 *Structured Query Language (SQL)*

Menurut Kadir (2009:10) *Structured Query Language (SQL)* merupakan bahasa *query* yang paling banyak dipilih oleh DBMS dan *Development Tools* dalam menyediakan media bagi penggunaannya untuk berinteraksi dengan basis data. Bahasa ini dibangun dengan dasar aljabar relasional dan sedikit kalkulus relasional. Pada tahun 1992 telah ditetapkan standar awal dari bahasa SQL, terutama menyangkut sintaksnya, yang kemudian dikenal dengan SQL-92. Dalam perjalanannya, karena adanya kebutuhan-kebutuhan baru dalam pengelolaan dan

pengolahan data, masing-masing pembuat DBMS melakukan pengembangan sendiri-sendiri, yang pada akhirnya melahirkan varian-varian SQL.

Jenis-jenis perintah SQL dapat dikelompokkan menjadi tiga :

1. DDL (*Data Definition Language*), berhubungan dengan perintah untuk mendefinisikan data. Termasuk dalam kelompok ini adalah perintah untuk membuat database, menghapus database, menciptakan tabel, menghapus tabel dan mengubah tabel.
2. DML (*Data Manipulation Language*), berhubungan dengan perintah untuk memanipulasi data. Termasuk dalam kelompok ini adalah pernyataan-pernyataan untuk mengambil data, menambahkan data, mengubah data dan menghapus data.
3. DCL (*Data Control Language*), berhubungan dengan perintah yang digunakan untuk mengontrol data. Termasuk dalam kelompok ini adalah pernyataan untuk mengatur hak akses terhadap data dan mencabut hak akses.

2.1.4 Entity Relationship Modeling

2.1.4.1 Tipe Entitas

Entitas adalah sesuatu dalam dunia nyata keberadaanya tidak bergantung pada yang lain (Elmasri dan Navathe, 1994). Sedangkan Entitas menurut Hoffer, dkk, (2005) menjelaskan bahwa entitas dapat berupa seseorang, sebuah tempat, sebuah objek, sebuah kejadian atau suatu konsep.

Tipe entitas dapat dikelompokkan menjadi:

1. Tipe entitas kuat

Sebuah tipe entitas yang keberadaannya tidak tergantung pada beberapa tipe entitas lain. Contoh, entitas Mahasiswa, Buku.

2. Tipe entitas lemah

Sebuah tipe entitas yang keberadaannya tergantung pada beberapa tipe entitas lain. Contoh, entitas Anak bergantung pada entitas Dosen.

2.1.4.2 Tipe Hubungan (*Relationship*)

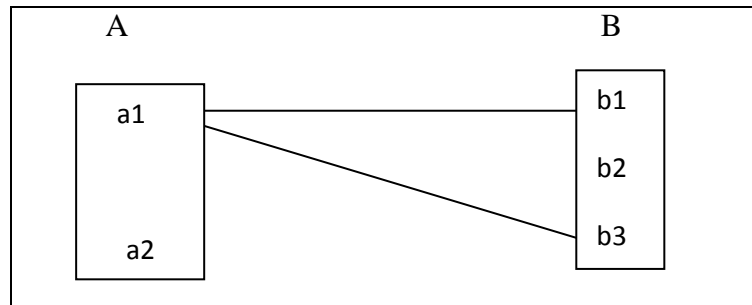
Simarmata (2009:97) Hubungan (*Relationship*) merupakan keterkaitan antara beberapa tipe entitas. Jenis Hubungan didalam skema E-R untuk suatu himpunan relasi biner R antara himpunan entitas A dan B, pemetaan kardinalitas harus salah satu dari berikut:

1. *One-to-one*, sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A.



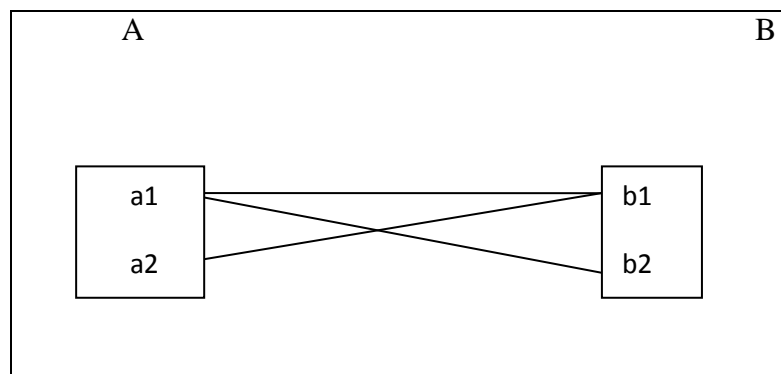
Gambar 2.1. Hubungan *One-to-one*

2. *One-to-many*, sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan paling banyak satu entitas pada A.



Gambar 2.2. Hubungan *One-to-many*

3. *Many-to-many*, sebuah entitas pada A berhubungan dengan nol lebih entitas pada B dan sebuah entitas pada B dapat dihubungkan nol atau lebih entitas pada A.



Gambar 2.3. Hubungan *Many-to-many*

2.1.4.3 Tipe Atribut

Atribut adalah properti atau *karakteristik* yang terdapat pada setiap entitas. Setiap atribut dinyatakan dengan kata benda. Supaya *konsisten*, Hoffer, dkk. (2005) menggunakan huruf kapital untuk setiap awal kata dan huruf kecil untuk yang lain. Jika atribut menggunakan lebih dari satu kata, antarkata dipisahkan oleh karakter garis-bawah (_).

Single-valued attribute adalah *attribute* yang memiliki satu nilai pada setiap *entity*, sedangkan *multi-valued attribute* adalah *attribute* yang mempunyai beberapa nilai pada setiap *entity* (Connolly,2002,p340). Contoh dari *single-valued*

attribute adalah Nim, nama_Mhs, tanggal_lahir, dan lain-lain. Sedangkan untuk *multi-valued attribute* contohnya adalah jam_pelajaran, hobi, dan lain-lain.

2.1.4.4 Kunci (Key)

Kunci kandidat adalah sebuah atribut atau gabungan beberapa atribut yang digunakan untuk membedakan antara satu baris dan dengan baris yang lain. Dengan kata lain kunci tersebut dapat bertindak sebagai identitas yang unik bagi baris-baris dalam suatu relasi. Kunci *Primer* adalah kunci kandidat yang dipilih sebagai identitas untuk membedakan satu baris dengan baris yang lain dalam satu relasi. Sebuah relasi hanya memiliki satu kunci primer. Sedangkan kunci asing (*foreign key*) adalah sebuah atribut atau gabungan beberapa atribut dalam suatu relasi yang merujuk ke kunci primer relasi lain. Kunci asing dalam suatu relasi yang mengacu pada kunci primer milik relasi lain merupakan perwujudan untuk membentuk hubungan antar-relasi.

2.1.5 Normalisasi

Normalisasi adalah suatu proses yang digunakan untuk menentukan pengelompokkan atribut-atribut dalam sebuah relasi sehingga diperoleh relasi yang berstruktur baik. Dalam hal ini yang dimaksud dengan relasi yang berstruktur baik adalah relasi yang memenuhi dua kondisi berikut :

- 1) Mengandung redundansi sesedikit mungkin
- 2) Memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi dan dihapus tanpa menimbulkan kesalahan atau ketidakkonsistenan.

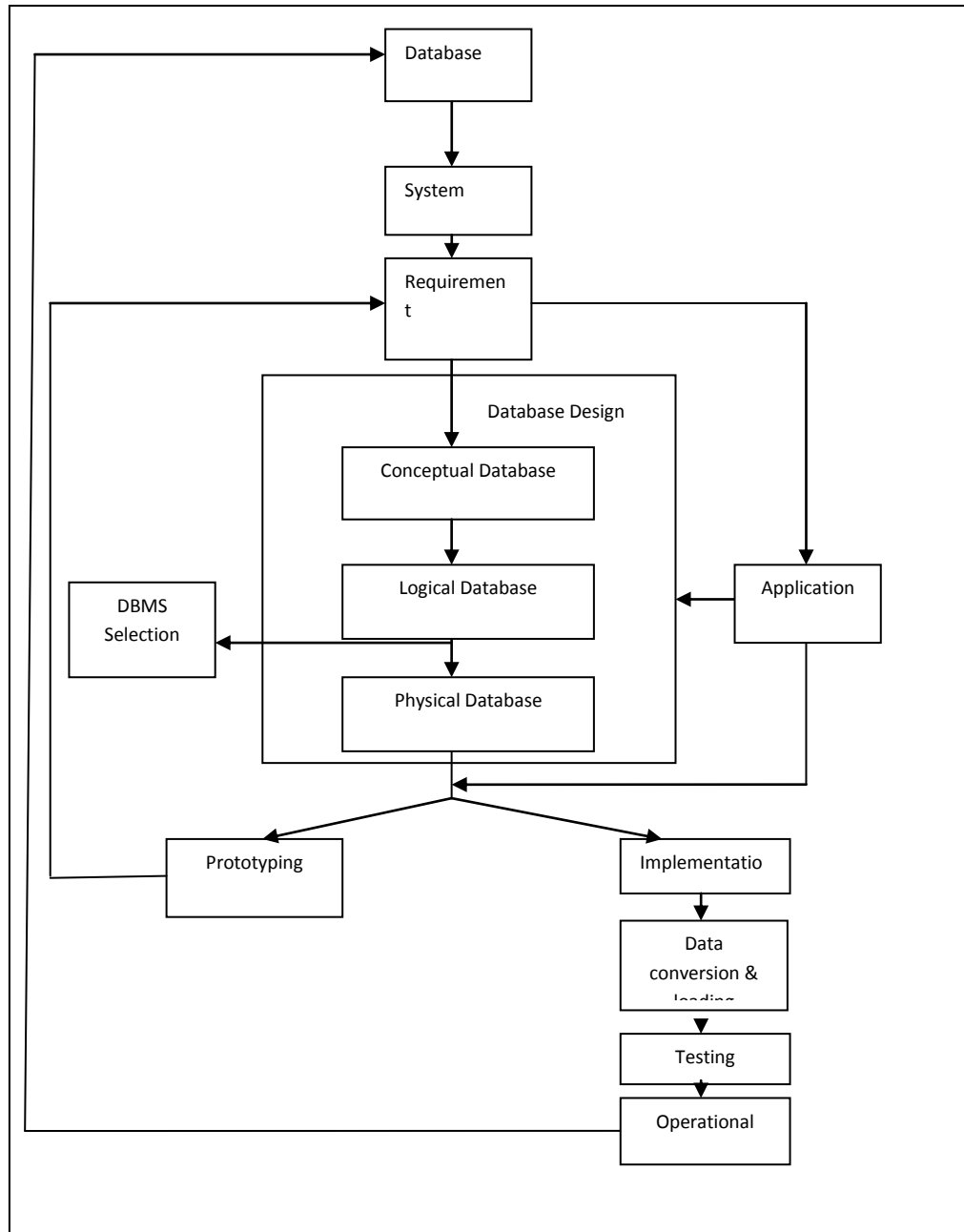
Normalisasi sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal (*normal form*) tertentu. Dalam hal ini yang disebut bentuk normal adalah “suatu keadaan relasi yang dihasilkan oleh penerapan aturan-aturan sederhana yang berhubungan dengan dependensi fungsional terhadap relasi tersebut”(Hoffer,dkk.,2005).

Tahapan-tahapan *normalisasi* yang umum sering digunakan :

- 1) Bentuk normal pertama (*1NF/First normal form*), bentuk normal pertama (1NF) adalah suatu keadaan yang membuat setiap perpotongan baris dan kolom dalam relasi hanya berisi satu nilai. Contohnya : Nip:107, Nama: Adi, Jabatan: Analisis senior, Keahlian: Access, Oracle.
- 2) Bentuk normal kedua (*2NF/Second normal form*), bentuk normal kedua (2NF) adalah suatu bentuk yang menyaratkan bahwa relasi harus sudah berada dalam bentuk normal pertama dan tidak mengandung dependensi parsial. Contohnya : Tabel 1 terdiri dari Nim: 001, Nama: Adi, Dosen_Wali: Dr. Erwin dan Tabel 2 terdiri dari Nim : 001, Kode_Mk:001.
- 3) Bentuk normal ketiga (*3NF/Third normal form*), bentuk normal ketiga (3NF) adalah suatu keadaan yang mensyaratkan bahwa relasi harus sudah berada dalam bentuk normal kedua dan tidak mengandung dependensi transitif. Contohnya: Tabel 1 terdiri dari Nim: 001, Kode_Subjek: TI01,Nilai: A, Bobot: 4, Tabel 2 terdiri dari Kode_Subjek:TI01, Nama_Subyek: Fisika, Sks: 3, Tabel 3 terdiri dari Nim: 1234, Nama_mhs: Tim, Tgl_lhr: 17-5-1990.

2.1.6. Siklus Perancangan Basis Data

Menurut Simarmata (2009:11) Tahapan siklus perancangan basis data dalam penelitian ini terdapat pada siklus hidup pengembangan basis data berikut:



Gambar 2.4. Siklus Hidup Pengembangan Basis Data

Aktifitas-aktifitas yang berhubungan dengan Tahapan siklus perancangan basis data dan termasuk fase-fasenya sebagai berikut :

1. Database planning

Pada aktifitas ini akan disusun bagaimana langkah-langkah siklus hidup dapat direalisasikan secara lebih efisien dan efektif.

2. System definition

Definisi ruang lingkup database (misal : para pemakai, aplikasi-aplikasinya, dsb.)

3. *Requirement Collection and Analysis*

Yakni kegiatan pengumpulan dan analisis informasi mengenai bagian dari perusahaan yang akan didukung oleh aplikasi basis data.

4. Database Design

Yakni kegiatan perancangan konseptual, logical dan fisik pada basis data.

5. *DBMS Selection (Optional)*

Yakni kegiatan menyeleksi DBMS yang cocok untuk diterapkan pada aplikasi basis data.

6. Application Design

Yakni kegiatan perancangan user interface dan program aplikasi yang akan digunakan dan akan memproses data.

7. Prototyping

Yakni kegiatan membangun model pekerjaan atau kegiatan pada aplikasi basis data, yang memungkinkan perancang atau pengguna untuk memvisualisasikan dan mengevaluasi bagaimana tampilan dan kegunaan dari sistem yang dihasilkan.

8. Implementation

Membuat bagian luar (external), konseptual dan mendefinisikan basis data internal serta program aplikasi.

9. Data conversion dan loading

Yaitu peralihan dengan pemuatan sistem lama ke sistem yang baru.

10. Testing

Dengan menguji coba kesalahan atau error pada aplikasi basis data dan memvalidasikan penentuan kebutuhan pengguna.

11. Operational maintenance

Pada stage ini aplikasi basis data secara penuh diterapkan, dimana sistem secara terus menerus diawasi dan dipelihara. Akan sangat penting apabila kebutuhan yang baru tergabung pada aplikasi basis data melalui stage sebelumnya pada siklus hidup.

2.1.7 Perancangan Basis Data

Menurut Simarmata (2009:2) Perancangan basis data adalah yang dikemukakan sebagian besar dengan pengetahuan dan pengalaman pengembang, seperti halnya produk *automated design* (AD) yang tersedia. Perancangan basis data mengacu pada aktivitas yang memusatkan pada perancangan dari struktur basis data yang akan digunakan untuk menyimpan dan mengatur pengguna akhir data. Basis data yang baik yaitu adalah, basis data yang memenuhi semua kebutuhan pengguna; strukturnya harus dirancang secara hati-hati. Sesungguhnya, perancangan basis data adalah aspek yang rumit dari pekerjaan dengan basis data.

1. Perancangan basis data *Conceptual*, merupakan suatu proses pembentukan model yang berasal dari informasi yang digunakan dalam perusahaan yang bersifat independen dari keseluruhan aspek fisik. Model data tersebut dibangun dengan menggunakan informasi dalam spesifikasi kebutuhan *user* dan merupakan sumber informasi untuk fase desain logikal.
2. Perancangan basis data *Logikal*, merupakan suatu proses pembentukan model yang berasal dari informasi yang digunakan dalam perusahaan yang berdasarkan model data tertentu, namun independen terhadap DBMS tertentu dan aspek fisik lainnya. Misalnya relasional. Model data konseptual yang telah dibuat sebelumnya, diperbaiki dan dipetakan kembali ke dalam model data logikal.
3. Perancangan basis data *Fisikal*, merupakan proses yang menghasilkan deskripsi implementasi basis data pada penyimpanan sekunder. Menggambarkan struktur penyimpanan dan metode akses yang digunakan untuk mencapai akses yang efisien terhadap data. Dapat dikatakan juga, desain fisikal merupakan cara pembuatan menuju DBMS tertentu.

2.2. Penelitian Sebelumnya

Penelitian sebelumnya digunakan untuk dapat dijadikan bahan pertimbangan dan diharapkan dapat membantu dalam penelitian yang baru.

Menurut Andry Chowanda dengan judul penelitian “Analisis dan Perancangan Sistem Basis Data Pembelian, Penjualan dan Persediaan pada PT Interjaya Surya Megah”. Sistem yang digunakan belum sepenuhnya maksimal sehingga belum terintegrasi dengan benar. Dihasilkan rancangan basis data untuk proses bisnis penjualan, pembelian dan persediaan kaca mata. Dikembangkan sebuah prototype aplikasi untuk memastikan sistem akhir akan terlihat dan berfungsi. Dengan adanya rancangan basis data dan pengembangan aplikasi, sehingga proses bisnis terlaksana secara cepat dan akurat.

Dalam penelitian Mhd Hanif dkk “Analisis dan Perancangan Sistem Basis Data Penjualan, Pembelian dan Pelayanan di Bidang Otomotif Berbasis Web Pada Dealer Suzuki Lenteng Agung”. Menginovasi perusahaan dalam aspek pembelian, pelayanan dan penjualan agar dapat terus bersaing ketat dalam industry otomotif Indonesia, menganalisa kelemahan-kelemahan yang selama ini dianggap biasa. Penulis merancang suatu sistem dengan menggunakan beberapa file *database*. Hasil yang dicapai dapat meningkatkan keunggulan sistem yang kompetitif dan mampu bersaing dengan perusahaan otomotif lainnya. Simpulannya adalah menerapkan sistem *database* berbasis *web* untuk menyelesaikan permasalahan proses kerja yang berhubungan dengan data.

BAB III

ANALISIS DAN PERANCANGAN

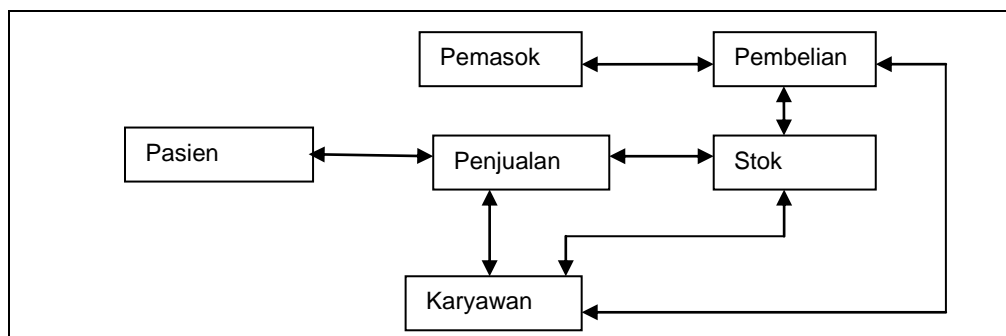
Adapun analisis dan perancangan basis data menggunakan metode database life cycle yang terdiri dari perencanaan basis data(*database planning*), definisi sistem(*system definition*), analisi dan pengumpulan, perancangan basis data(*design database*).

3.1. Perencanaan Basis Data (*Database Planning*)

Perancangan basis data pada Rumah Sakit Sriwijaya *Eye Center* bertujuan untuk menyimpan dan mengelolah data penjualan dan pembelian dan memfasilitasi penggunaan data bersama antar bagian. Pernyataan tujuan dalam perancangan basis data ini dijabarkan dalam beberapa kebutuhan akan informasi yang ada pada umah Sakit Sriwijaya *Eye Center*. Adapun kebutuhan akan proses basis data antara lain untuk mengelola (memasukkan, memperbaruhi, dan menghapus) data-data yang berhubungan dengan penjualan dan pembelian pada rumah Sakit Sriwijaya *Eye Center*.

3.2. Definisi Sistem (*System Definition*)

Definisi sistem merupakan alur dari sistem yang mencakup penjualan dan pembelian yang ada pada umah Sakit Sriwijaya *Eye Center*, dimana definisi sistem yang ada saling terkait dan skema yang ada pada perusahaan mencakup pasien yang berhubungan dengan penjualan, penjualan berhubungan dengan bagian gudang, bagian gudang berhubungan dengan bagian pembelian, bagian pembelian berhubungan dengan pemasok.



Gambar 3.1. Definisi Sistem

3.3. Analisis dan Pengumpulan

3.3.1. Analisis Sistem Yang Berjalan

Berikut ini merupakan urutan prosedur dalam sistem yang berjalan :

3.3.1.1. Prosedur Penjualan

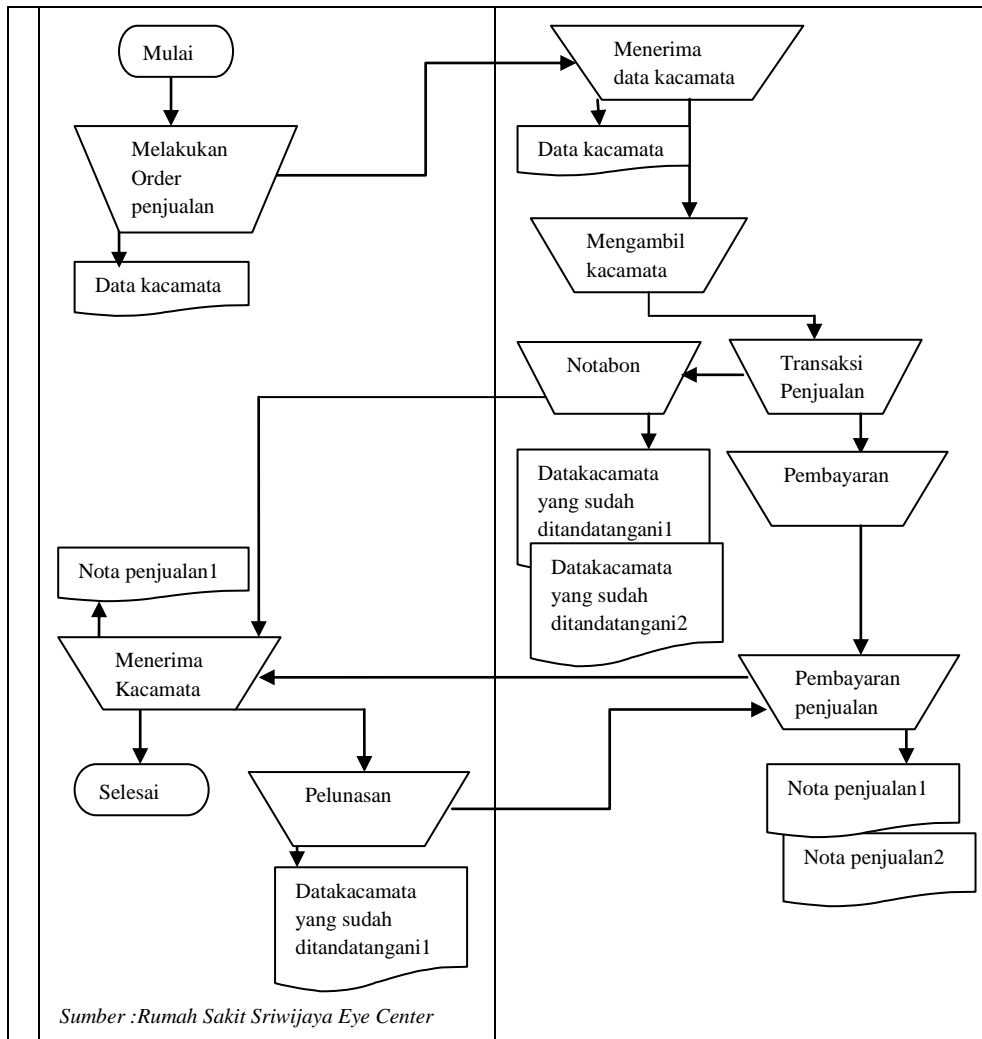
Prosedur penjualan yang sedang berjalan pada Sakit Sriwijaya *Eye Center* ialah sebagai berikut :

1. Pembeli melakukan *order* penjualan kepada karyawan.
2. Karyawan menerima *order* penjualan dan melengkapi kaca mata yang di *order* oleh pembeli.

3. Karyawan membuat tagihan pembayaran.
4. Tagihan pembayaran di Sakit Sriwijaya *Eye Center* meliputi tunai dan notabon.
5. Pembeli yang melakukan pembayaran tunai langsung melakukan pembayaran penjualan.
6. Pembeli yang melakukan pembayaran notabon, setelah memberikan list kacamata yang dipesan dan telah mentandatangani daftar pesanan kacamata tersebut pembeli dapat membawa kacamata tersebut.
7. Pembeli melakukan pelunasan piutang dan melakukan pembayaran penjualan sesuai kesepakatan.

Untuk lebih jelasnya dapat dilihat pada gambar 3.2 *flowchart* penjualan.

Prosedur Penjualan yang berjalan	
Pembeli	Kasir



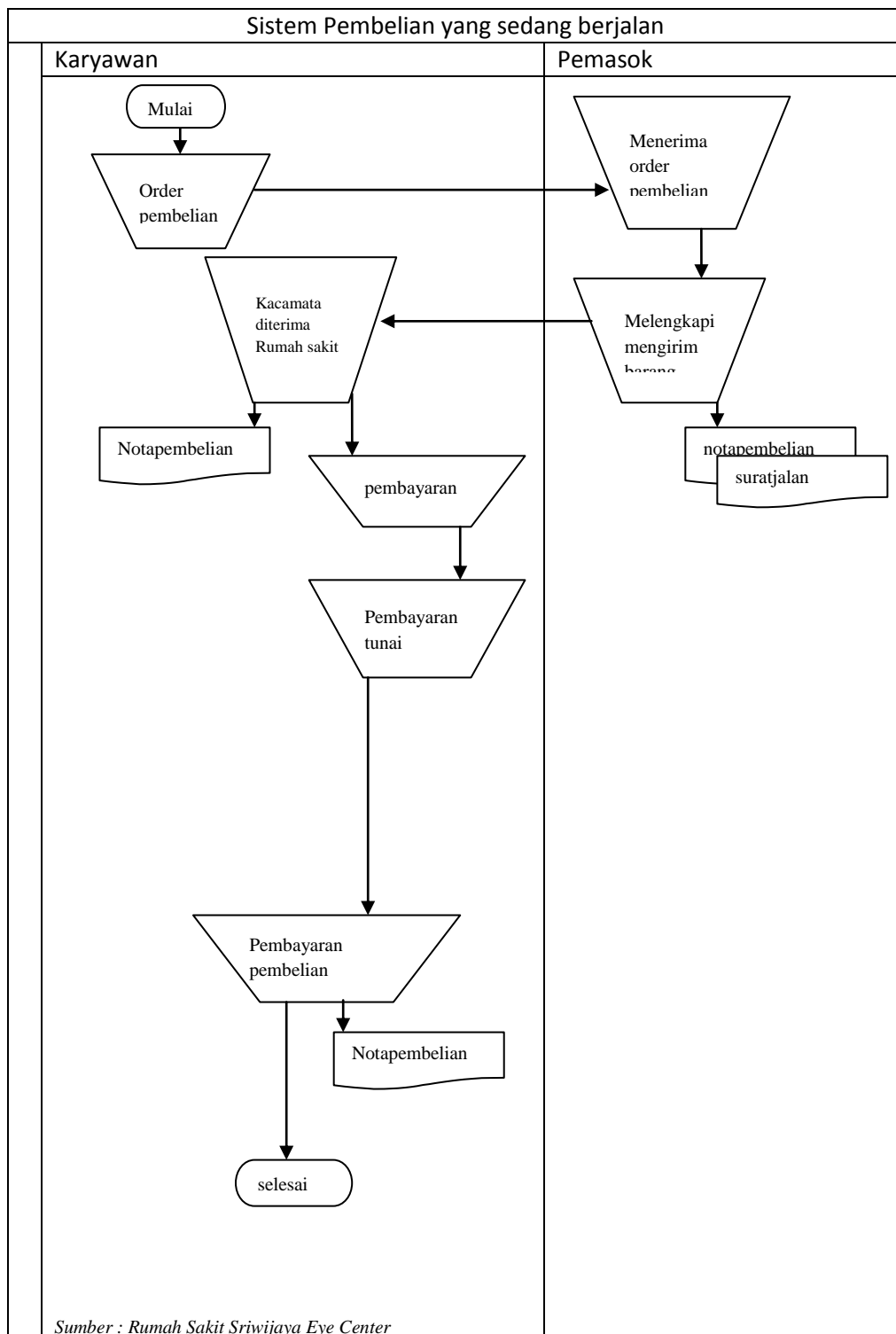
Gambar 3.2. *Flowchart* sistem penjualan

3.3.1.2. Prosedur Pembelian

Prosedur pembelian yang sedang berjalan pada Sakit Sriwijaya *Eye Center* ialah sebagai berikut :

1. Karyawan melakukan *order* pembelian kepada pemasok.
2. Pemasok menerima *order* pembelian dan melengkapi kacamata yang di order oleh rumah sakit dan mengirim kacamata ke rumah sakit.
3. Karyawan menerima kacamata.
4. Tagihan pembayaran pembelian di Sakit Sriwijaya *Eye Center*, meliputi tunai.
5. Perusahaan jika akan melakukan pembayaran tunai langsung melakukan pembayaran pembelian.

Untuk lebih jelasnya dapat dilihat pada gambar 3.3 *flowchart* sistem pembelian.



Gambar 3.3. Flowchart sistem pembelian.

3.3.2. Permasalahan Yang Dihadapi dan Solusi Pemecahan Masalah

Permasalahan yang terjadi saat ini, adapun uraian permasalahan dan rencana usulan pemecahan masalah adalah sebagai berikut :

1. Usulan Pemecahan Masalah, merancang basis data untuk mengurangi redundansi data dan duplikasi data saat melakukan manipulasi data karena basis data penjualan dan pembelian sudah terintegrasi dan menjaga keamanan data.

3.4. Perancangan Basis Data (*Design Database*)

Dalam tahap perancangan basis data terdiri dari 3 tahapan yang disesuaikan dengan kebutuhan informasi dari Rumah Sakit Sriwijaya *Eye Center*.

Tiga buah tahapan perancangan basis data tersebut, yaitu :

1. Perancangan basis data konseptual
2. Perancangan basis data logikal
3. Perancangan basis data fisik

3.4.1. Perancangan Basis Data Konseptual

Perancangan basis data konseptual merupakan suatu proses pembentukan model yang berasal dari informasi yang digunakan dalam perusahaan yang bersifat independen dari kebutuhan aspek fisik. Langkah-langkah yang digunakan dalam perancangan basis data konseptual adalah sebagai berikut :

1. Mengidentifikasi tipe entitas
2. Mengidentifikasi tipe relasi
3. Mengidentifikasi atribut dengan tipe entitas

4. Menentukan domain atribut
5. Menentukan atribut *candidate key* dan *primary key*

3.4.1.1. Identifikasi Tipe Entitas

Tujuan mengidentifikasi tipe suatu *entity* adalah mengetahui tipe entiti utama yang diperlukan untuk view. *Entity-entity* yang menjadi kebutuhan dari perusahaan Rumah Sakit Sriwijaya *Eye Center* setelah melalui proses analisis sistem yang berjalan.

Tabel 3.1. Identifikasi Tipe Entitas

Nama Entitas	Deskripsi	Kejadian
Pasien	Mendeskripsikan orang yang melakukan pemesanan untuk transaksi penjualan pada rumah sakit	Setiap pasien melakukan satu atau lebih pemesanan penjualan
Karyawan	Mendeskripsikan orang yang berkerja pada rumah sakit	Setiap karyawan berkerja pada masing-masing bagian
Pemasok	Mendeskripsikan orang yang menerima pembelian kacamata dari rumah sakit	Setiap pemasok memasok satu atau lebih kacamata
Kacamata	Mendeskripsikan kacamata yang dijual oleh rumah sakit	Setiap kacamata dapat dijual ke pasien
OrderPenjualan	Mendeskripsikan penjualan yang dilakukan rumah sakit	Setiap penjualan yang dilakukan rumah sakit
Pembayaranpenjualan	Mendeskripsikan pembayaran penjualan dari pasien kepada rumah sakit	Setiap pembayaran penjualan dibuat berdasarkan satu penjualan
OrderPembelian	Mendeskripsikan pembelian berdasarkan pesanan rumah sakit	Setiap pembelian yang dilakukan rumah sakit
Pembayaranpembelian	Mendeskripsikan pembayaran pembelian dari rumah sakit kepada pemasok	Setiap pembayaran pembelian dibuat berdasarkan satu penjualan

3.4.1.2. Identifikasi Tipe Relasi

Setelah dilakukan dan didapat entitas-entitas yang terkait. Tahapan berikutnya yaitu melakukan identifikasi relasi atau hubungan antar entitas-entitas tersebut. Identifikasi tipe relasi model relasional, setelah tabel dilengkapi dengan kunci yang bersifat unik. Langkah selanjutnya adalah menerjemahkan hubungan antar tabel ke dalam kunci tamu. Cara melakukan penerjemahan ditentukan oleh jenis hubungan antar tabel. Dalam hal ini ada tiga macam hubungan yaitu : hubungan ke satu (1:1), hubungan satu ke banyak (1:M) dan hubungan banyak ke banyak(M:M).

Identifikasi tipe relasi sebagai berikut : entitas Pasien memiliki hubungan satu ke satu(1:1) melakukan relasi dengan tabel orderpenjualan sehingga menjadi hubungan satu ke banyak(1:M). Entitas karyawan memiliki hubungan satu ke satu(1:1) menerima relasi dengan tabel orderpenjualan sehingga menjadi hubungan satu ke banyak(1:M), menerima relasi tabel pembayaran penjualan sehingga menjadi hubungan satu ke banyak (1:M), menerima relasi tabel orderpenjualan sehingga menjadi hubungan satu ke banyak (1:M) dan menerima relasi tabel pembayaran pembelian sehingga menjadi hubungan satu ke banyak (1:M). Entitas pemasok memiliki hubungan satu ke satu (1:1) akan menerima orderpembelian sehingga menjadi hubungan satu ke banyak(1:M), Entitas kacamata memiliki hubungan satu ke banyak (1:M) akan melibatkan orderpenjualan sehingga menjadi hubungan satu ke banyak (1:M) dan dilibatkan oleh orderpenjualan menjadi hubungan satu ke banyak (1:M), Entitas orderpenjualan memiliki hubungan satu ke satu (1:1) akan mendasari pembayaran penjualan menjadi hubungan satu ke banyak (1:M) dan melibatkan tabel kacamata yang hubungannya dari satu ke banyak. Entitas pembayaran penjualan memiliki

hubungan dari satu ke banyak(1:M) didasari pada order penjualan yang memiliki hubungan satu ke satu (1:1). Order pembelian memiliki hubungan satu ke satu (1:1) mendasari tabel pembayaran pembelian menjadi satu ke banyak (1:M). Entitas pembayaran pembelian memiliki hubungan dari satu ke banyak (1:M) didasari oleh order pembelian sehingga menjadi hubungan satu ke satu (1:1).

Keterangan :

3.4.1.3. Identifikasi Atribut dengan Tipe Entitas

Atribut yang diperlukan untuk masing-masing entitas dalam perancangan basis data Rumah Sakit Sriwijaya *Eye Center* dijelaskan dalam tabel 3.3 Tabel Identifikasi Atribut dengan Tipe Entitas.

Tabel 3.2. Tabel Identifikasi Atribut dengan Tipe Entitas

Entitas	Atribut
Pasien	Kdpasien,namapasien,telppasien,Hppasien,fakspasien,alamatpasien
Karyawan	Kdkaryawan,username,password,namakaryawan,Hpkaryawan,alamatkaryawan,hakakses
Pemasok	Kdpemasok,namapemasok,alamatpemasok,telppemasok,Hppemasok,fakspemasok
Kacamata	Noreg, merk, hargabeli, hargajual, jumlah, satuan
OrderPenjualan	Kdorderpenjualan,tglorderpenjualan,kdkaryawan, namakaryawan, kdpasien,namapasien,Noreg,merk,jumlah,hargajual,statusbayar
Pembayaranpenjualan	Kdpembayaranpenjualan,tglpembayaranpenjualan,kdorderpenjualan,kdkaryawan,namakaryawan,nilaipembayaran

Tabel 3.3. Tabel Identifikasi Atribut dengan Tipe Entitas (Lanjutan)

Entitas	Atribut
OrderPembelian	Kdorderpembelian,tglorderpembelian,kdkaryawan,namakaryawan, kdpemasok,namapemasok,noreg,merk,jumlah,hargabeli,statusbayar
Pembayaranpembelian	Kdpembayaranpembelian,tglpembayaranpembelian,kdorderpembelian,kdkaryawan,namakaryawan,nilaipembayaran

3.4.1.4. Menentukan Domain Atribut

Entitas-entitas yang telah dilakukan identifikasi tipe relasi kemudian dilakukan penentuan domain atribut. Domain atribut merupakan penentuan tipe data dari setiap atribut yang ada dalam tiap-tiap entitas.

1. Nama Entitas : Pasien

Tabel 3.4. Tabel Domain Atribut Pasien

Atribut	Domain Atribut
KdPasien	Char(10)
NamaPasien	Varchar(50)
TelpPasien	Varchar(14)
HpPasien	Varchar(14)
AlamatPasien	Varchar(50)

2. Nama Entitas : Karyawan

Tabel 3.5. Tabel Domain Atribut Karyawan

Atribut	Domain Atribut
KdKaryawan	Char(10)
Username	Varchar(20)
Password	Varchar(10)
NamaKaryawan	Varchar(50)
AlamatKaryawan	Varchar(70)
TelpKaryawan	Varchar(14)
HpKaryawan	Varchar(14)
Hakakses	Varchar(10)

3. Nama Entitas : Pemasok

Tabel 3.6. Tabel Domain Atribut Pemasok

Atribut	Domain Atribut
KdPemasok	Char(6)
NamaPemasok	Varchar(50)
TelpPemasok	Varchar(14)
HpPemasok	Varchar(14)
FaksPemasok	Varchar(14)
AlamatPemasok	Varchar(50)

4. Nama Entitas : Kacamata

Tabel 3.7. Tabel Domain Atribut Kacamata

Atribut	Domain Atribut
NoReg	Char(6)
Merk	Varchar(50)
HargaBeli	Int(25)
HargaJual	Int(25)
Jumlah	Int(5)
WarnaFrame	Varchar(25)
UkuranLensa	Varchar(25)
Kdsatuan	Int(5)

5. Nama Entitas : OrderPenjualan

Tabel 3.8. Tabel Domain Atribut OrderPenjualan

Atribut	Domain Atribut
KdOrderPenjualan	Char(10)
TglOrderPenjualan	Datetime
KdKaryawan	Char(10)
NamaKaryawan	Varchar(50)
KdPasien	Char(10)
NamaPasien	Varchar(50)
Noreg	Char(10)
Merk	Varchar(50)
Jumlah	Int
HargaJual	Int
StatusBayar	Varchar(10)

6. Nama Entitas : PembayaranPenjualan

Tabel 3.9. Tabel Domain Atribut PembayaranPenjualan

Atribut	Domain Atribut
KdPembayaranPenjualan	Char(10)
TglPembayaranPenjualan	Datetime
KdOrderPenjualan	Char(10)
KdKaryawan	Char(10)
NamaKaryawan	Varchar(50)
NilaiPembayaran	Int

7. Nama Entitas : OrderPembelian

Tabel 3.10. Tabel Domain Atribut OrderPembelian

Atribut	Domain Atribut
KdOrderPembelian	Char(10)
TglOrderPembelian	Datetime
KdKaryawan	Char(6)
Namakaryawan	Varchar(50)
Kdpemasok	Char(10)
Namapemasok	Varchar(50)
NoReg	Char(6)
Merk	Varchar(50)
Jumlah	Int
HargaBeli	Int
StatusBayar	Varchar(10)

8. Nama Entitas : PembayaranPembelian

Tabel 3.11. Tabel Domain Atribut PembayaranPembelian

Atribut	Domain Atribut
KdPembayaranPembelian	Char(10)
TglPembayaranPembelian	Datetime
KdOrderPembelian	Char(10)
KdKaryawan	Char(10)
NamaKaryawan	Varchar(50)
NilaiPembayaran	Int

3.4.1.5. Menentukan Atribut *Candidate* dan *Primary Key*

Atribut yang terdapat didalam tiap-tiap entitas kemudian dilakukan pemberian *primary key*. Pemberian *primary key* diawali dengan dilakukan pengelompokkan terhadap atribut-atribut yang dimungkinkan untuk dijadikan *primary key*. Atribut yang tergolong kedalam calon *primary key* disebut *candidate key*.

Tabel 3.12. Attribute *Candidate key* dan *Primary key*

Nama Entitas	<i>Candidate Key</i>	<i>Primary Key</i>
Pasien	KdPasien TelpPasien HpPasien	Kdpasien
Karyawan	Kdkaryawan Telpkaryawan HpKaryawan	Kdkaryawan
Pemasok	Kdpemasok TelpPemasok Hppemasok	Kdpemasok
Kacamata	Noreg	Noreg
Orderpenjualan	Kdorderpenjualan Kdkaryawan Noreg	Kdorderpenjualan
Pembayaranpenjualan	Kdpembayaranpenjualan Kdorderpenjualan kdkaryawan	kdpembayaranpenjualan
Orderpembelian	Kdorderpembelian Kdkaryawan Noreg	Kdorderpembelian
Pembayaranpembelian	Kdpembayaranpembelian Kdorderpembelian Kdkaryawan	Kdpembayaranpembelian

3.4.1.6. Perancangan Basis Data Logikal

Perancangan basis data logikal merupakan lanjutan dari tahapan konseptual. Pada tahapan perancangan logikal akan menggambarkan apa yang sebenarnya disimpan dalam basis data dan hubungannya. Langkah-langkah yang digunakan dalam perancangan basis data logikal adalah sebagai berikut :

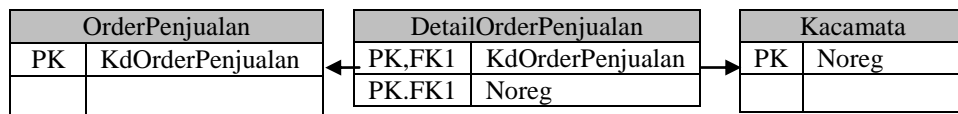
1. Menghilangkan fitur yang tidak kompatibel dengan model relasional.
2. Menurunkan relasi untuk model data *logical local*.

3. Validasi relasi menggunakan normalisasi.

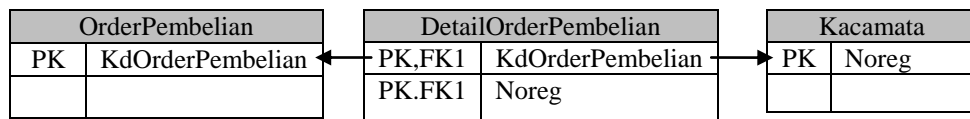
3.4.2.1. Menghilangkan Fitur yang Tidak Kompatibel dengan Model Relasional.

1. Menghilangkan tipe relasi *binary many to many* (M:M), pada model konseptual, terdapat beberapa relasi *biner many to many*, antara lain:

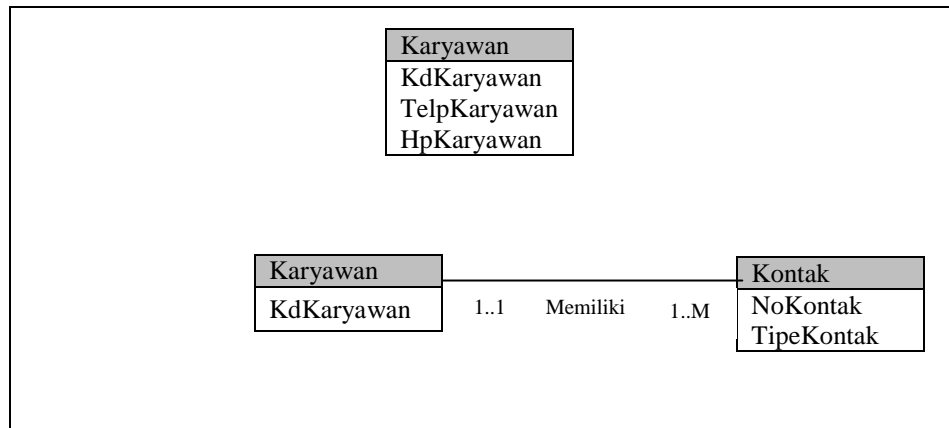
a. Relasi *many to many* antara entitas Orderpenjualan dan kacamata membentuk entitas baru yaitu Detailorderpenjualan.



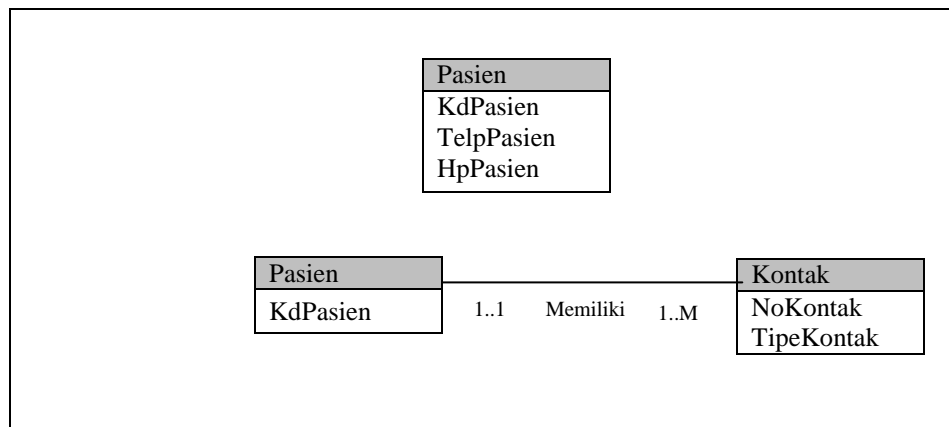
b. Relasi *many to many* antara entitas Orderpembelian dan kacamata membentuk entitas baru yaitu Detailorderpembelian.



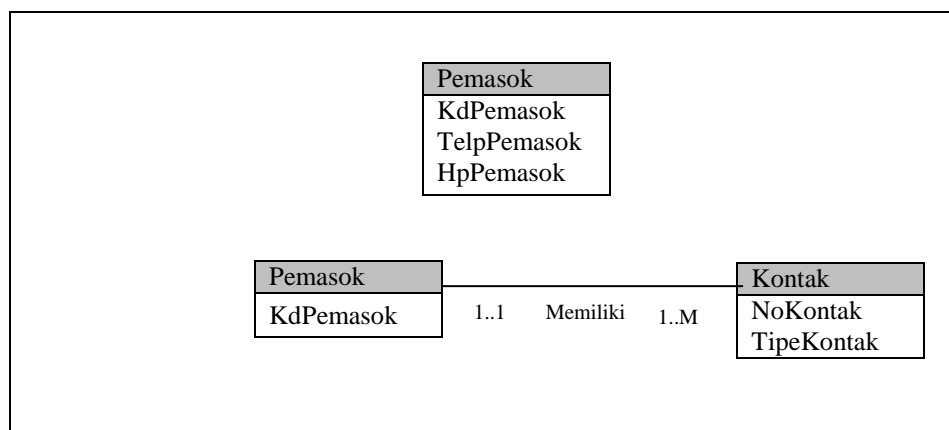
2. Menghilangkan Atribut *Multi-Value*, pada model konseptual, terdapat beberapa entitas yang memiliki atribut dengan nilai jamak (*multi-valued*). Hal tersebut tidak diperbolehkan didalam model relasional. Oleh karena itu, diperlukan proses dekomposisi pada atribut tersebut dan membuat entiti baru yang terpisah. Proses menghilangkan *multi-value* dapat dilihat pada gambar dibawah ini :



Gambar 3.4. Atribut *multi value* pada entitas karyawan



Gambar 3.5. Atribut *multi value* pada entitas Pasien



Gambar 3.6. Atribut *multi value* pada entitas Pemasok

3.4.2.2. Menurunkan relasi untuk Model Data *Logical Local*

1. Tipe Entitas Kuat

Entitas Kuat merupakan entitas yang tidak bergantung pada entitas lainnya.

Berikut adalah himpunan entitas kuat :

- a. **Pasien** (Kdpasien,namapasien, alamatpasien)

Primary key Kdpasien

- b. **Karyawan**(Kdkaryawan,username,password,namakaryawan,alamatkaryawan,hakakses)

Primary key Kdkaryawan

- c. **Pemasok** (Kdpemasok, namapemasok,alamatpemasok)

Primary key Kdpemasok

- d. **Kacamata** (Noreg, merk, hargabeli, hargajual, jumlah, satuan)

Primary key Noreg

- e. **Orderpenjualan**(Kdorderpenjualan,tglorderpenjualan,kdkaryawan,namakaryawan,kdpasien,namapasien, statusbayar)

Primary key KdOrderPenjualan

Foreign key Kdkaryawan **references** karyawan (kdkaryawan)

Foreign key Kdpasien **references** pasien (kdpasien)

- f. **Pembayaranpenjualan**(Kdpembayaranpenjualan,tglpembayaranpenjualan,kdorderpenjualan,kdkaryawan,namakaryawan,nilaipembayaran)

Primary key kdpembayaranpenjualan

Foreign key kdorderpenjualan **references** orderpenjualan (kdorderpenjualan)

Foreign key kdkaryawan **references** karyawan (kdkaryawan)

- g. **Orderpembelian**(Kdorderpembelian,tglorderpembelian,kdkaryawan,namakaryawan, kdpemasok,namapemasok,statusbayar)

Primary key kdorderpembelian

Foreign key kdkaryawan **references** karyawan (kdkaryawan)

Foreign key kdpemasok **references** pemasok (kdpemasok)

- h. **Pembayaranpembelian**(Kdpembayaranpembelian,tglpembayaranpembelian,kdorderpembelian,kdkaryawan,namakaryawan,nilaipembayaran)

Primary key kdpembayaranpembelian

Foreign key kdorderpembelian **references** orderpembelian (kdorderpembelian)

Foreign key kdkaryawan **references** karyawan (kdkaryawan)

2. Tipe Entitas Lemah

Tipe entitas lemah merupakan entitas yang kemunculannya tergantung pada eksistensinya dalam sebuah relasi terhadap entitas lain (entitas kuat). Berikut adalah himpunan entitas lemah :

a. **Kontak** (KdKontak, NoKontak, TipeKontak)

Primary Key KdKontak

Foreign key Kdkaryawan **references** karyawan (kdkaryawan)

Foreign key Kdpasien **references** pasien (kdpasien)

Foreign key Kdpemasok **references** pemasok (kdpemasok)

b. **Detailorderpenjualan**(Kdorderpenjualan, noreg, merk, jumlah, warnaframe, ukuranlensa, harga jual)

Primary key kdorderpenjualan, noreg

Foreign key kdorderpenjualan **references** orderpenjualan (kdorderpenjualan)

Foreign key noreg **references** (noreg)

c. **Detailorderpembelian**(Kdorderpembelian, noreg, merk, jumlah, warnaframe, ukuranlensa, hargabeli)

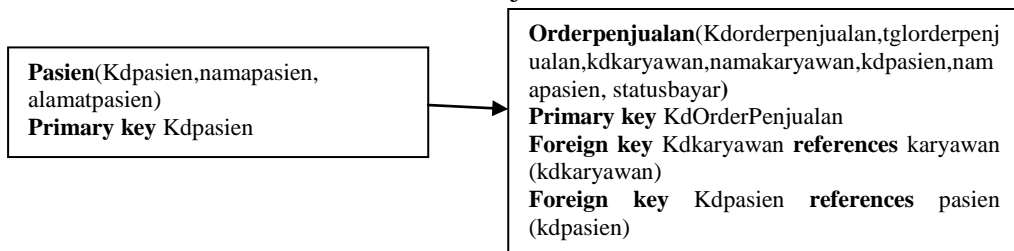
Primary key kdorderpembelian, noreg

Foreign key kdorderpembelian **references** orderpembelian (kdorderpembelian)

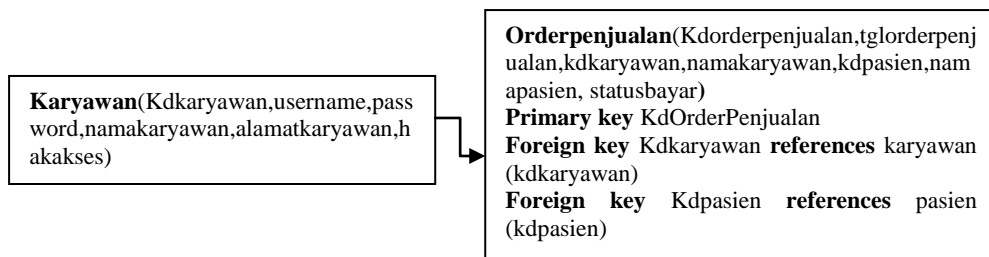
Foreign key noreg **references** kaca mata (noreg)

3. Tipe Hubungan Biner *One-to-Many* (1:M), relasi biner *one to many* (1:M) dibangun dengan cara mengandalkan atribut kunci primer dari entitas yang berada pada sisi *one* (1:1) kepada entitas yang berada pada sisi *many* (1:M) guna berperan sebagai *foreign key*.

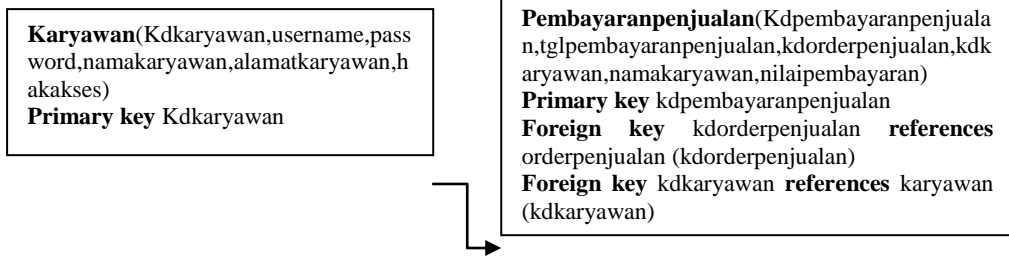
a. Relasi antara Pasien dan OrderPenjualan



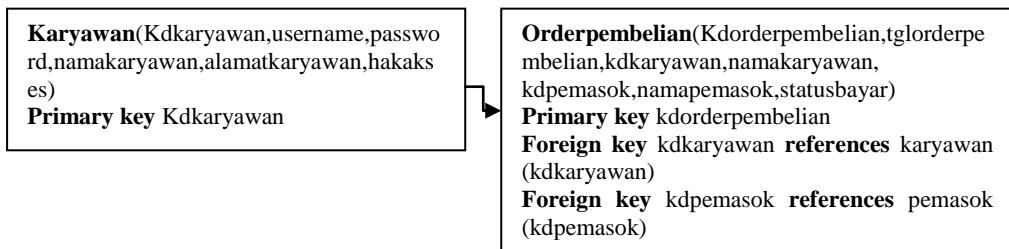
b. Relasi antara Karyawan dan OrderPenjualan



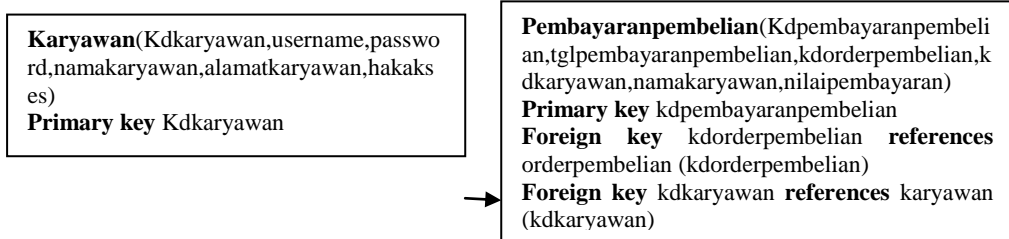
c. Relasi antara Karyawan dan PembayaranPenjualan



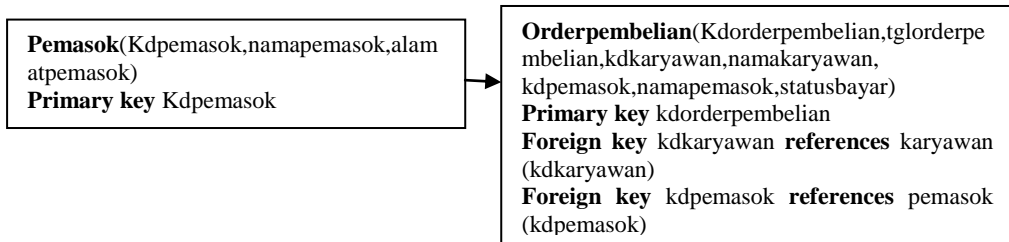
d. Relasi antara Karyawan dan OrderPembelian



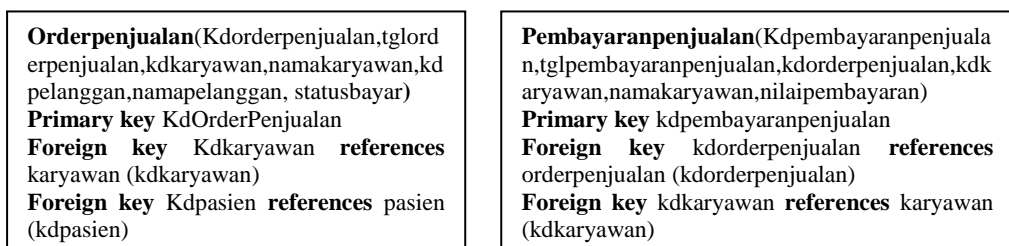
e. Relasi antara Karyawan dan PembayaranPembelian



f. Relasi antara Pemasok dan OrderPembelian

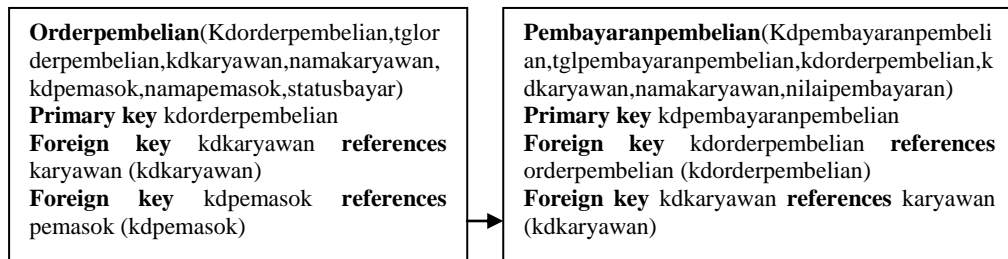


g. Relasi antara OrderPenjualan dan PembayaranPenjualan



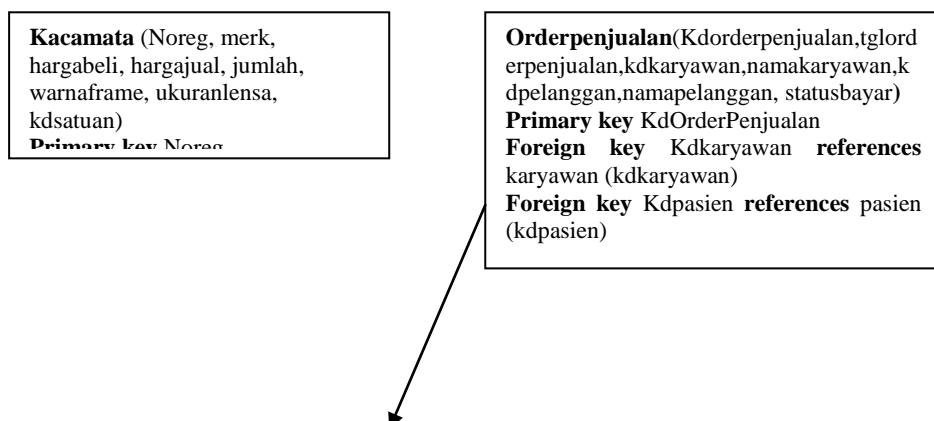


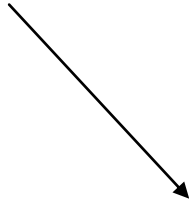
h. Relasi antara OrderPembelian dan PembayaranPembelian



4. Tipe Hubungan Biner *Many-to-Many* (M:M), setiap relasi yang mempunyai hubungan tiap entitas *many-to-many* (M:M) harus dihilangkan. Untuk menghilangkan relasi tersebut dibuat sebuah relasi yang mempresentasikan hubungan antara entitas dan memasukkan seluruh atribut yang merupakan bagian dari tabel relasi. *Primary key* disalin ke relasi baru dan akan bertindak sebagai *foreing key*.

- a. Relasi antara kacamata dan order penjualan yang membentuk entitas yaitu detailorderpenjualan, dimana atribut *primary key* dari masing-masing entitas yaitu noreg, kdorderpenjualan.



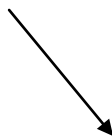


Detailorderpenjualan(Kdorderpenjualan,Noreg,merk,ju
mlah,hargajual)
Primary key kdorderpenjualan,noreg
Foreign key kdorderpenjualan **references** orderpenjualan
(kdorderpenjualan)
Foreign key noreg **references** kacamata (noreg)

- b. Relasi antara kacamata dan order pembelian yang membentuk entitas yaitu detailorderpembelian, dimana atribut *primary key* dari masing-masing entitas yaitu noreg, kdorderpembelian.

Barang (Noreg, merk, hargabeli, hargajual, jumlah, warnaframe, ukuranlensa, kdsatuan)
Primary key noreg

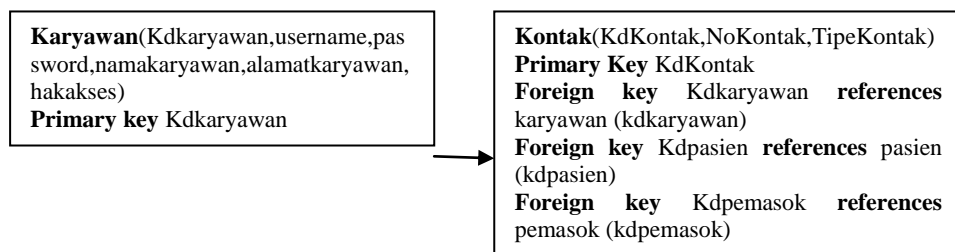
Orderpembelian(Kdorderpembelian,tglorderpembelian,kdkaryawan,namakaryawan, kdpemasok,namapemasok,statusbayar)
Primary key kdorderpembelian
Foreign key kdkaryawan **references** karyawan (kdkaryawan)
Foreign key kdpemasok **references** pemasok (kdpemasok)



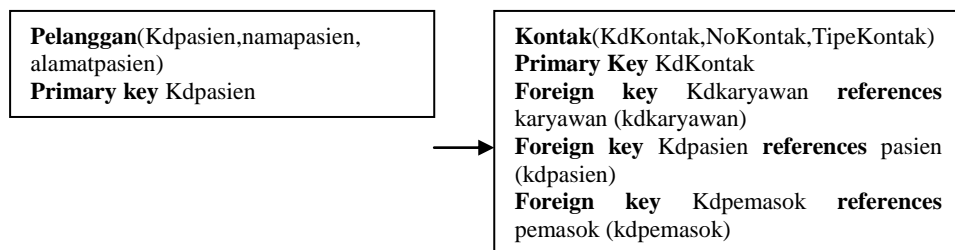
Detailorderpembelian(Kdorderpembelian,noreg,merk,ju
mlah, warnaframe,ukuranlensa,hargabeli)
Primary key kdorderpembelian,noreg
Foreign key kdorderpembelian **references** orderpembelian (kdorderpembelian)
Foreign key noreg **references** kacamata (noreg)

5. Atribut *Multi-Valued*, pada model konseptual, terdapat beberapa entitas yang memiliki atribut dengan nilai jamak (*multi-valued*), diantaranya yaitu :

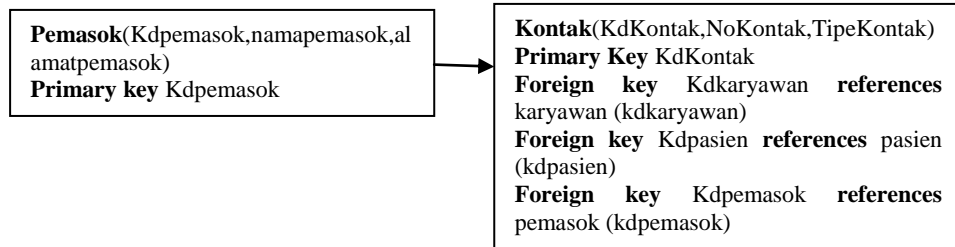
a. Hubungan karyawan dengan kontak



b. Hubungan pasien dengan kontak



c. Hubungan pemasok dengan kontak



Berikut ini ditunjukkan bagaimana entitas kuat mengirimkan *primary key*-nya ke dalam entitas lemah sehingga terjadi hubungan dan relasional yang berkaitan. Maka didapatkan hasil relasi dari proses menentukan model data logikal lokal.

Tabel 3.13. Relasi untuk Model Data Logika

Entitas	Atribut
Pasien	Pasien (Kdpasien,namapasien, alamatpasien) Primary key Kdpasien
Karyawan	Karyawan (Kdkaryawan,username,password,namakaryawan,alamatkaryawan,hakakses) Primary key Kdkaryawan
Pemasok	Pemasok (Kdpemasok,namapemasok,alamatpemasok) Primary key Kdpemasok
Kacamata	Kacamata (noreg, merk, hargabeli, hargajual, jumlah, warnaframe, ukuranlensa, kdsatuan) Primary key noreg
Kontak	Kontak (KdKontak,NoKontak,TipeKontak) Primary Key KdKontak Foreign key Kdkaryawan references karyawan (kdkaryawan) Foreign key Kdpasien references pasien (kdpasien) Foreign key Kdpemasok references pemasok (kdpemasok)

Orderpenjualan	Orderpenjualan (Kdorderpenjualan,tglorderpenjualan,kdkaryawan,nama karyawan,kdpasien,namapasien, statusbayar) Primary key KdOrderPenjualan Foreign key Kdkaryawan references karyawan (kdkaryawan) Foreign key Kdpasien references pasien (kdpasien)
Detailorderpenjualan	Detailorderpenjualan (Kdorderpenjualan,noreg,merk,jumlah,hargajual) Primary key kdorderpenjualan,noreg Foreign key kdorderpenjualan references orderpenjualan (kdorderpenjualan) Foreign key noreg references kacamata (noreg)
Pembayaranpenjualan	Pembayaranpenjualan (Kdpembayaranpenjualan,tglpembayaranpenjualan,kdorderpenjualan,kdkaryawan,namakaryawan,nilaipembayaran) Primary key kdpembayaranpenjualan Foreign key kdorderpenjualan references orderpenjualan (kdorderpenjualan) Foreign key kdkaryawan references karyawan (kdkaryawan)
Orderpembelian	Orderpembelian (Kdorderpembelian,tglorderpembelian,kdkaryawan,namakaryawan, kdpemasok,namapemasok,statusbayar) Primary key kdorderpembelian Foreign key kdkaryawan references karyawan (kdkaryawan) Foreign key kdpemasok references pemasok (kdpemasok)
Detailorderpembelian	Detailorderpembelian (Kdorderpembelian,noreg,merk,jumlah,hargabeli) Primary key kdorderpembelian,noreg Foreign key kdorderpembelian references orderpembelian (kdorderpembelian) Foreign key noreg references kacamata (noreg)
Pembayaranpembelian	Pembayaranpembelian (Kdpembayaranpembelian,tglpembayaranpembelian,kdorderpembelian,kdkaryawan,namakaryawan,nilaipembayaran) Primary key kdpembayaranpembelian Foreign key kdorderpembelian references orderpembelian (kdorderpembelian) Foreign key kdkaryawan references karyawan (kdkaryawan)

Berikut ini ditunjukkan juga model ERD dari model data logikal.

Validasi relasi menggunakan normalisasi, normalisasi data dimulai dari *Unnormalized Form (UNF)*, *First Normal Form (1NF)*, *Second Normal Form (2NF)*, dan *Third Normal Form (3NF)* sebagai berikut :

1. Pada tabel pasien akan meminimalkan duplikasi penggunaan attribute bernilai *multi valued* seperti telppasien, hppasien dan fakspasien yang berulang ulang sehingga menghasilkan tabel baru yaitu tabel kontak dan tabel tipe kontak.

<p>UNF:</p> <p>Pasien= Kdpasien + namapasien + {telppasien + Hppasien} + alamatpasien</p> <p>1NF:</p> <p>Pasien = @Kdpasien+namapasien+alamatpasien Kontak=@KdKontak + NoKontak + KdTipeKontak + TipeKontak</p> <p>2NF :</p> <p>Pasien = @Kdpasien+namapasien+alamatpasien Kontak=@KdKontak + NoKontak + KdTipeKontak + TipeKontak</p> <p>3NF :</p> <p>Pasien = @Kdpasien+namapasien+alamatpasien Kontak=@KdKontak + NoKontak + #KdTipeKontak TipeKontak=@ KdTipeKontak + TipeKontak</p>
--

Gambar 3.8. Normalisasi pada tabel Pasien

2. Pada tabel Karyawan akan meminimalkan duplikasi penggunaan attribute bernilai *multi valued* seperti telpkaryawan, hpkaryawan dan hakakses yang berulang ulang sehingga menghasilkan tabel baru yaitu tabel kontak dan tabel tipe kontak dan hakakses.

<p>UNF:</p> <p>Karyawan= Kdkaryawan + username + password + namakaryawan + {telpkaryawan + hpkaryawan} + alamatkaryawan + hakakses</p> <p>1NF:</p> <p>Karyawan=@Kdkaryawan + username + password + namakaryawan + alamatkaryawan + kdhakakses + hakakses</p> <p>Kontak=@KdKontak + NoKontak + KdTipeKontak + TipeKontak</p> <p>2NF:</p> <p>Karyawan=@Kdkaryawan + username + password + namakaryawan + alamatkaryawan + kdhakakses + hakakses</p> <p>Kontak=@KdKontak + NoKontak + KdTipeKontak + TipeKontak</p> <p>3NF :</p> <p>Karyawan=@Kdkaryawan + username + password + namakaryawan + alamatkaryawan + #kdhakakses</p> <p>Kontak=@KdKontak + NoKontak + #KdTipeKontak</p> <p>TipeKontak=@ KdTipeKontak + TipeKontak</p> <p>Hakakses=@ kdhakakses + hakakses</p>

Gambar 3.9. Tabel karyawan

3. Pada tabel Pemasok akan meminimalkan duplikasi penggunaan attribute bernilai *multi valued* seperti telppemasok, hppemasok dan fakspemasok yang berulang ulang sehingga menghasilkan tabel baru yaitu tabel kontak dan tabel tipe kontak.

<p>UNF:</p> <p>Pemasok=Kdpemasok + namapemasok + alamatpemasok + {telppemasok + hppemasok + faks}</p> <p>1NF:</p> <p>Pemasok=@Kdpemasok + namapemasok + alamatpemasok</p> <p>Kontak=@KdKontak + NoKontak + KdTipeKontak + TipeKontak</p>

Gambar 3.10. Normalisasi pada tabel Pemasok

Gambar 3.11. Normalisasi pada tabel Pemasok (Lanjutan)

2NF :
Pemasok =@Kdpemasok + namapemasok + alamatpemasok
Kontak =@KdKontak + NoKontak + KdTipeKontak + TipeKontak
3NF :
Pemasok =@Kdpemasok + namapemasok + alamatpemasok
Kontak =@KdKontak + NoKontak + #KdTipeKontak
TipeKontak =@ KdTipeKontak + TipeKontak

Gambar 3.11. Tabel pemasok

4. Pada tabel Kacamata akan meminimalkan duplikasi penggunaan attribute bernilai *multi valued* seperti satuan kacamata yang berulang ulang sehingga menghasilkan tabel baru yaitu tabel satuan kacamata.

UNF:
Kacamata =Noreg + merk + hargabeli + harga jual + jumlah + warnaframe + ukuranlensa + kdsatuan
1NF
Kacamata =@Noreg + merk + hargabeli + harga jual + warnaframe + ukuranlensa + jumlah + kdsatuan + satuan
2NF :
Kacamata =@Noreg + merk + hargabeli + harga jual + jumlah + warnaframe + ukuranlensa + kdsatuan + satuan
3NF :
Kacamata =@Noreg + merk + hargabeli + harga jual + jumlah + warnaframe + ukuranframe + #kdsatuan
Satuan =@ kdsatuan + satuan

Gambar 3.12. Normalisasi pada tabel Kacamata

5. Pada tabel *Order Penjualan* akan meminimalkan duplikasi penggunaan attribute bernilai *multi valued* seperti noreg, merk, jumlah kacamata dan status bayar yang berulang ulang sehingga menghasilkan tabel baru yaitu tabel detail order penjualan dan status pembayaran.

<p>UNF :</p> <p>OrderPenjualan=Kdorderpenjualan + tglorderpenjualan + kdkaryawan + namakaryawan + kdpasien + namapasien + {noreg + merk + jumlah} + harga jual+ statusbayar</p> <p>1NF :</p> <p>OrderPenjualan=@Kdorderpenjualan + tglorderpenjualan + kdkaryawan + namakaryawan + kdpasien + namapasien + kdstatusbayar + statusbayar + harga jual DetailOrderPenjualan=@Kdorderpenjualan + @noreg + merk + jumlah</p> <p>2NF :</p> <p>OrderPenjualan=@Kdorderpenjualan + tglorderpenjualan + kdkaryawan + namakaryawan + kdpasien + namapasien + kdstatusbayar + statusbayar + harga jual DetailOrderPenjualan=@Kdorderpenjualan + @noreg + merk + jumlah</p> <p>3NF:</p> <p>OrderPenjualan=@Kdorderpenjualan + tglorderpenjualan + #kdkaryawan + #kdpasien + #kdstatusbayar + harga jual DetailOrderPenjualan=@Kdorderpenjualan + @noreg + jumlah Statusbayar=@ kdstatusbayar + statusbayar</p>
--

Gambar 3.13. Normalisasi pada tabel *Order Penjualan*

6. Pada tabel Pembayaran penjualan akan meminimalkan duplikasi penggunaan attribute bernilai *multi valued* seperti kdorderpenjualan dan nilai pembayaran yang berulang ulang sehingga menghasilkan tabel yaitu tabel Detail Pembayaran penjualan.

UNF :	
Pembayaranpenjualan	=Kdpembayaranpenjualan + tglpembayaranpenjualan + kdkaryawan + namakaryawan + {kdorderpenjualan + nilaipembayaran}
1NF :	
Pembayaranpenjualan	=@Kdpembayaranpenjualan + tglpembayaranpenjualan + kdkaryawan + namakaryawan
DetailPembayaranpenjualan	=@Kdpembayaranpenjualan + @kdorderpenjualan + nilaipembayaran
2NF :	
Pembayaranpenjualan	=@Kdpembayaranpenjualan + tglpembayaranpenjualan + kdkaryawan + namakaryawan
DetailPembayaranpenjualan	=@Kdpembayaranpenjualan + @kdorderpenjualan + nilaipembayaran
3NF :	
Pembayaranpenjualan	=@Kdpembayaranpenjualan + tglpembayaranpenjualan + #kdkaryawan
DetailPembayaranpenjualan	=@Kdpembayaranpenjualan + @kdorderpenjualan + nilaipembayaran

Gambar 3.13. Normalisasi pada tabel Pembayaran penjualan

7. Pada tabel *Order* Pembelian akan meminimalkan duplikasi penggunaan attribute bernilai *multi valued* seperti noreg, merk, jumlah dan status bayar yang berulang ulang sehingga menghasilkan tabel baru yaitu yaitu tabel *Detail Order* Pembelian dan status bayar.

UNF :	
OrderPembelian	=Kdorderpembelian + tglorderpembelian + kdkaryawan + namakaryawan + kdpemasok + namapemasok + {noreg + merk + jumlah} + hargabeli + statusbayar

Gambar 3.14. Normalisasi pada tabel *Order* Pembelian

Gambar 3.15. Normalisasi pada tabel *Order* Pembelian (Lanjutan)

1NF :
OrderPembelian =@Kdorderpembelian + tglorderpembelian + kdkaryawan + namakaryawan + kdpemasok + namapemasok + kdstatusbayar + statusbayar + hargabeli DetailOrderPembelian =@Kdorderpembelian + @noreg + merk + jumlah
2NF :
OrderPembelian =@Kdorderpembelian + tglorderpembelian + kdkaryawan + namakaryawan + kdpemasok + namapemasok + kdstatusbayar + statusbayar + hargabeli DetailOrderPembelian =@Kdorderpembelian + @noreg + merk + jumlah
3NF :
OrderPembelian =@Kdorderpembelian + tglorderpembelian + #kdkaryawan + #kdpemasok + #kdstatusbayar + hargabeli DetailOrderPembelian =@Kdorderpembelian + @noreg + jumlah Statusbayar = @ kdstatusbayar + statusbayar

8. Pada tabel Pembayaran Pembelian akan meminimalkan duplikasi penggunaan attribute bernilai *multi valued* seperti kdorderpenjualan dan nilai pembayaran yang berulang ulang sehingga menghasilkan tabel yaitu tabel Detail Pembayaran Pembelian.

UNF :
Pembayaranpembelian =Kdpembayaranpembelian + tglpembayaranpembelian + kdkaryawan + namakaryawan + { kdorderpembelian + nilaipembayaran }

Gambar 3.16. Normalisasi pada tabel Pembayaran Pembelian

Gambar 3.17. Normalisasi pada tabel Pembayaran Pembelian (Lanjutan)

1NF :		
Pembayaranpembelian =@ Kdpembayaranpembelian		+
tglpembayaranpembelian + kdkaryawan + namakaryawan		
DetailPembayaranpembelian =@ Kdpembayaranpembelian		+
@kdorderpembelian + nilaipembayaran		
2NF :		
Pembayaranpembelian =@ Kdpembayaranpembelian		+
tglpembayaranpembelian + kdkaryawan + namakaryawan		
DetailPembayaranpembelian =@ Kdpembayaranpembelian		+
@kdorderpembelian + nilaipembayaran		
3NF :		
Pembayaranpembelian =@ Kdpembayaranpembelian		+
tglpembayaranpembelian + # kdkaryawan		
DetailPembayaranpembelian =@ Kdpembayaranpembelian		+
@kdorderpembelian + nilaipembayaran		

Struktur Tabel Hasil Normalisasi adalah sebagai berikut :

1. Nama Tabel : Karyawan
Primary key : Kdkaryawan
Foreign key : Kdhakakses

Tabel 3.14. Struktur Tabel Karyawan Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdkaryawan	char	10
Namakaryawan	Varchar	25
Alamatkaryawan	Varchar	50
Username	Varchar	25
Password	Varchar	50
Kdhakakses	Char	10

2. Nama Tabel : Hak akses
Primary key : Kdhakakses

Tabel 3.15. Struktur Tabel hak akses Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdhakakses	char	10

Hakakses	Varchar	25
----------	---------	----

3. Nama Tabel : Pemasok
Primary key : Kdpemasok

Tabel 3.16. Struktur Tabel Pemasok Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdpemasok	char	10
Namapemasok	varchar	25
Alamatpemasok	Varchar	50

4. Nama Tabel : Pasien
Primary key : Kdpasien

Tabel 3.17. Struktur Tabel Pasien Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdpasien	char	10
Namapasien	varchar	25
Alamatasien	Varchar	50

5. Nama Tabel : Kontak
Primary key : Kdkontak

Tabel 3.18. Struktur Tabel kontak Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
KdKontak	char	10
NoKontak	varchar	25
KdTipeKontak	Char	10

6. Nama Tabel : TipeKontak
Primary key : KdTipeKontak

Tabel 3.19. Struktur Tabel TipeKontak Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
KdTipeKontak	char	10
TipeKontak	varchar	25

7. Nama Tabel : Kacamata
Primary key : Noreg
Foreign key : Kdsatuan

Tabel 3.20. Struktur Tabel Kacamata Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Noreg	char	10

Merk	varchar	25
Hargabeli	int	15
Hargajual	Int	15
Jumlah	Int	15
Warnaframe	Varchar	25
Ukuranlensa	Varchar	25
Kdsatuan	char	10

8. Nama Tabel : Satuan
Primary key : Kdsatuan

Tabel 3.21. Struktur Tabel Satuan Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdsatuan	char	10
Satuan	varchar	15

9. Nama Tabel : Orderpenjualan
Primary key : Kdorderpenjualan
Foreign key : Kdkaryawan, Kdpasien, Kdstatusbayar

Tabel 3.22. Struktur Tabel Orderpenjualan Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdorderpenjualan	char	10
Tglorderpenjualan	Date	
Kdkaryawan	char	10
Kdpasien	Char	10
Hargajual	Int	20
Kdstatusbayar	char	10

10. Nama Tabel : Detailorderpenjualan
Primary key : Kdorderpenjualan
Foreign key : Noreg

Tabel 3.23. Struktur Tabel Detailorderpenjualan Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdorderpenjualan	Char	10
Noreg	Char	10
jumlah	Int	5

11. Nama Tabel : StatusBayar
Primary key : Kdstatusbayar

Tabel 3.24. Struktur Tabel StatusBayarSetelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
kdstatusbayar	Char	10
Statusbayar	Varchar	10

12. Nama Tabel : Pembayaranpenjualan
Primary key : Kdpembayaranpenjualan
Foreign key : Kdkaryawan

Tabel 3.25. Struktur Tabel Pembayaranpenjualan Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdpembayaranpenjualan	Char	10
Tglpembayaranpenjualan	Date	
Kdkaryawan	Char	10

13. Nama Tabel : Detailpembayaranpenjualan
Primary key : Kdpembayaranpenjualan
Foreign key : Kdorderpenjualan

Tabel 3.26. Struktur Tabel Detailpembayaranpenjualan Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdpembayaranpenjualan	Char	10
Kdorderpenjualan	Char	10
Nilaipembayaran	Int	15

14. Nama Tabel : OrderPembelian
Primary key : Kdorderpembelian
Foreign key : Kdkaryawan, Kdpemasok, Kdstatusbayar

Tabel 3.27. Struktur Tabel OrderPembelian Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdorderpembelian	Char	10
Tglorderpembelian	Date	
Kdkaryawan	Char	10
Kdpemasok	Char	10
Hargabeli	Int	20
Kdstatusbayar	Char	10

15. Nama Tabel : Detailorderpembelian
Primary key : Kdorderpembelian
Foreign key : Noreg

Tabel 3.28. Struktur Tabel Detailorderpembelian Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdorderpembelian	Char	10
Noreg	Char	10
Jumlah	int	5
Warnaframe	Varchar	25
Ukuranlensa	Varchar	25

16. Nama Tabel : Pembayaranpembelian
Primary key : KdPembayaranpembelian
Foreign key : Kdkaryawan

Tabel 3.29. Struktur Tabel Pembayaranpembelian Setelah Normalisasi

Nama Atribut	Tipe Data	Ukuran
Kdpembayaranpembelian	Char	10
Tglpembayaranpembelian	Date	

Kdkaryawan	Char	10
------------	------	----

17. Nama Tabel : Detailpembayaranpembelian
 Primary key : KdPembayaranpembelian
 Foreign key : Kdorderpembelian

Tabel 3.30. Struktur Tabel Detailpembayaranpembelian SetelahNormalisasi

Nama Atribut	Tipe Data	Ukuran
Kdpembayaranpembelian	Char	10
Kdorderpembelian	Char	10
Nilaipembayaran	Int	15

3.4.3. Perancangan Basis Data Fisikal

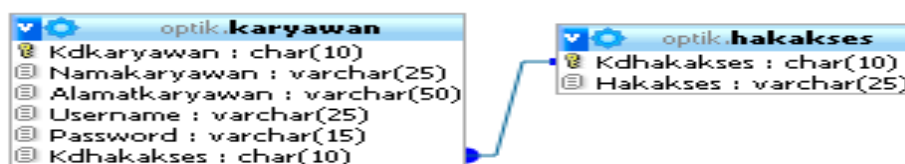
Perancangan basis data *fisikal* merupakan langkah dari perancangan basis data untuk menghasilkan deskripsi dari implementasi basis data pada tempat penyimpanan *sekunder*. Langkah-langkah yang digunakan dalam perancangan basis data *fisikal* adalah sebagai berikut :

1. Merancang relasi dasar
2. Memperkirakan kebutuhan *disk space*
3. Merancang Mekanisme Keamanan

3.4.3.1. Merancang Relasi Dasar

1. Tabel Karyawan

```
ALTER TABLE `karyawan`
  ADD CONSTRAINT `karyawan_ibfk_1` FOREIGN KEY (`kdhakakses`)
  REFERENCES `hakakses` (`kdhakakses`) ON DELETE CASCADE ON
  UPDATE CASCADE;
```



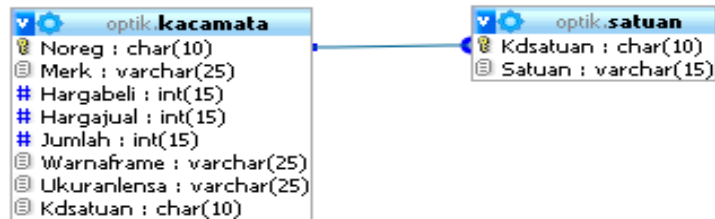
2. Tabel Kontak

```
ALTER TABLE `kontak`  
  ADD CONSTRAINT `kontak_ibfk_1` FOREIGN KEY (`KdtipeKontak`)  
  REFERENCES `tipekontak` (`KdtipeKontak`) ON DELETE CASCADE ON  
  UPDATE CASCADE;
```



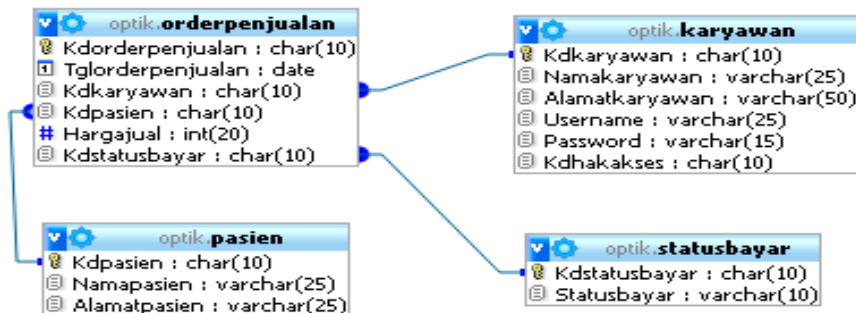
3. Tabel Kacamata

```
ALTER TABLE `kacamata`  
  ADD CONSTRAINT `kacamata_ibfk_1` FOREIGN KEY (`Kdsatuan`)  
  REFERENCES `satuan` (`Kdsatuan`) ON DELETE CASCADE ON UPDATE  
  CASCADE;
```



4. Tabel OrderPenjualan

```
ALTER TABLE `orderpenjualan`  
  ADD CONSTRAINT `orderpenjualan_ibfk_3` FOREIGN KEY  
  (`Kdstatusbayar`) REFERENCES `statusbayar` (`Kdstatusbayar`) ON  
  DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `orderpenjualan_ibfk_1` FOREIGN KEY  
  (`Kdkaryawan`) REFERENCES `karyawan` (`Kdkaryawan`) ON DELETE  
  CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `orderpenjualan_ibfk_2` FOREIGN KEY  
  (`Kdpasien`) REFERENCES `pasien` (`Kdpasien`) ON DELETE CASCADE  
  ON UPDATE CASCADE;
```



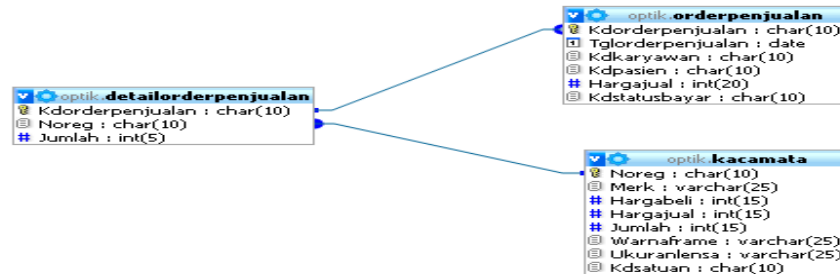
5. Tabel Detail Order Penjualan

```
ALTER TABLE `detailorderpenjualan`
```

```

ADD CONSTRAINT `detailorderpenjualan_ibfk_2` FOREIGN KEY
(`Noreg`) REFERENCES `kacamata` (`noreg`) ON DELETE CASCADE ON
UPDATE CASCADE,
ADD CONSTRAINT `detailorderpenjualan_ibfk_1` FOREIGN KEY
(`Kdorderpenjualan`) REFERENCES `orderpenjualan`
(`Kdorderpenjualan`) ON DELETE CASCADE ON UPDATE CASCADE;

```

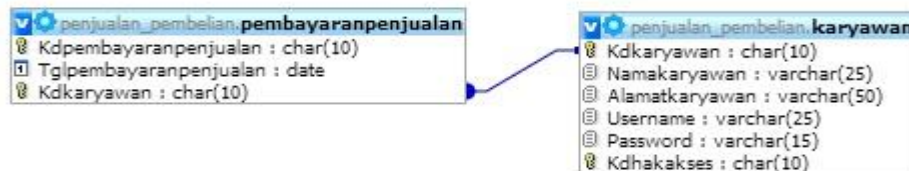


6. Tabel Pembayaran Penjualan

```

ALTER TABLE `pembayaranpenjualan`
ADD CONSTRAINT `pembayaranpenjualan_ibfk_1` FOREIGN KEY
(`Kdkaryawan`) REFERENCES `karyawan` (`Kdkaryawan`) ON DELETE
CASCADE ON UPDATE CASCADE;

```

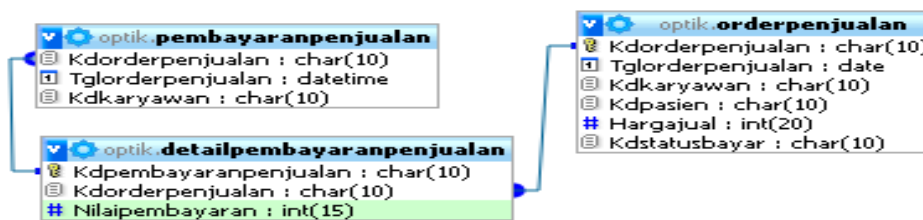


7. Tabel Detail Pembayaran Penjualan

```

ALTER TABLE `detailpembayaranpenjualan`
ADD CONSTRAINT `detailpembayaranpenjualan_ibfk_2` FOREIGN KEY
(`Kdpembayaranpenjualan`) REFERENCES `pembayaranpenjualan`
(`Kdpembayaranpenjualan`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `detailpembayaranpenjualan_ibfk_1` FOREIGN KEY
(`Kdorderpenjualan`) REFERENCES `orderpenjualan`
(`Kdorderpenjualan`) ON DELETE CASCADE ON UPDATE CASCADE

```

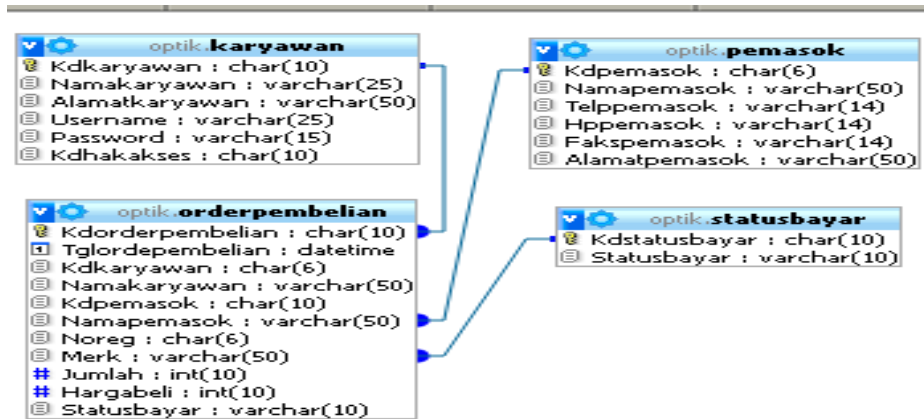


8. Tabel Order Pembelian

```

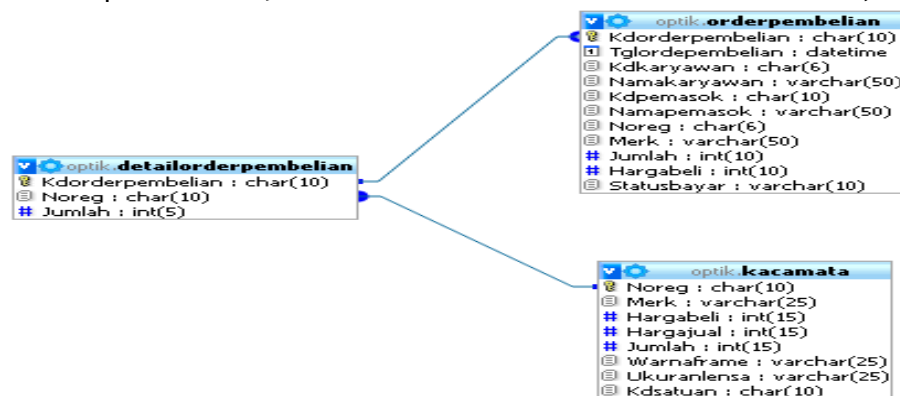
ALTER TABLE `orderpembelian`
ADD CONSTRAINT `orderpembelian_ibfk_3` FOREIGN KEY
(`Kdstatusbayar`) REFERENCES `statusbayar` (`Kdstatusbayar`) ON
DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `orderpembelian_ibfk_1` FOREIGN KEY
(`Kdkaryawan`) REFERENCES `karyawan` (`Kdkaryawan`) ON DELETE
CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `orderpembelian_ibfk_2` FOREIGN KEY
(`Kdpemasok`) REFERENCES `pemasok` (`Kdpemasok`) ON DELETE
CASCADE ON UPDATE CASCADE;

```



9. Tabel Detail Order Pembelian

```
ALTER TABLE `detailorderpembelian`
  ADD CONSTRAINT `detailorderpembelian_ibfk_2` FOREIGN KEY
  (`Noreg`) REFERENCES `kacamata` (`Noreg`) ON DELETE CASCADE ON
  UPDATE CASCADE,
  ADD CONSTRAINT `detailorderpembelian_ibfk_1` FOREIGN KEY
  (`Kdorderpembelian`) REFERENCES `orderpembelian`
  (`Kdorderpembelian`) ON DELETE CASCADE ON UPDATE CASCADE;
```



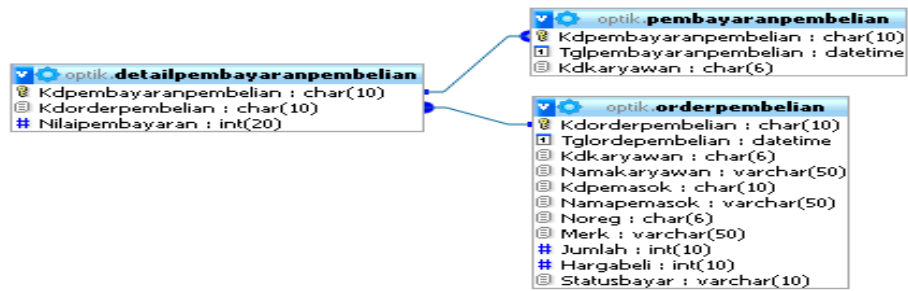
10. Tabel Pembayaran Pembelian

```
ALTER TABLE `pembayaranpembelian`
  ADD CONSTRAINT `pembayaranpembelian_ibfk_1` FOREIGN KEY
  (`Kdkaryawan`) REFERENCES `karyawan` (`Kdkaryawan`) ON DELETE
  CASCADE ON UPDATE CASCADE;
```



11. Tabel Detail Pembayaran Pembelian

```
ALTER TABLE `detailpembayaranpembelian`
  ADD CONSTRAINT `detailpembayaranpembelian_ibfk_2` FOREIGN KEY
  (`Kdorderpembelian`) REFERENCES `orderpembelian`
  (`Kdorderpembelian`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `detailpembayaranpembelian_ibfk_1` FOREIGN KEY
  (`Kdpembayaranpembelian`) REFERENCES `pembayaranpembelian`
  (`Kdpembayaranpembelian`) ON DELETE CASCADE ON UPDATE CASCADE;
```



3.4.3.2. Memperkirakan kebutuhan *disk space*

Pada tahap ini dilakukan penghitungan estimasi kebutuhan media penyimpanan.

Tabel 3.31. Ukuran Entitas

Nama Entitas	Nama Atribut	Tipe Data	Ukuran
Karyawan	Kdkaryawan	char	10
	Namakaryawan	Varchar	25
	Alamatkaryawan	Varchar	50
	Username	Varchar	25
	Password	Varchar	50
	Kdhakakses	Char	10
Total			170 byte
Hakakses	Kdhakakses	char	10
	Hakakses	Varchar	25
	Total		
Pemasok	Kdpemasok	char	10
	Namapemasok	varchar	25
	Alamatpemasok	Varchar	50
	Total		
Pasien	Kdpasien	char	10
	Namapasien	varchar	25
	Alamatpasien	Varchar	50
	Total		
Kontak	KdKontak	char	10
	NoKontak	varchar	25
	KdTipeKontak	Char	10
	Total		
Tipekontak	KdTipeKontak	char	10
	TipeKontak	varchar	25
	Total		
Kacamata	Noreg	char	10
	Merk	varchar	25
	Hargabeli	int	15
	Hargajual	Int	15
	Jumlah	Int	15
	Warnaframe	Varchar	25
	Ukuranlensa	Varchar	25
	Kdsatuan	char	10
Total			90 byte

Tabel 3.32. Ukuran Entitas (Lanjutan)

Nama Entitas	Nama Atribut	Tipe Data	Ukuran
Satuan	kdsatuan	char	10
	satuan	varchar	15
	Total		25 byte
Orderpenjualan	Kdorderpenjualan	char	10
	Tglorderpenjualan	Date	
	Kdkaryawan	char	10
	Kdpasien	Char	10
	Hargajual	Int	20
	Kdstatusbayar	char	10
	Total		60 byte
detailorderpenjualan	Kdorderpenjualan	Char	10
	Noreg	Char	10
	jumlah	Int	5
	Total		25 byte
Pembayaranpenjualan	Kdpembayaranpenjualan	Char	10
	Tglpembayaranpenjualan	Date	
	Kdkaryawan	Char	10
	Total		20 byte
detailpembayaranpenjualan	Kdpembayaranpenjualan	Char	10
	Kdorderpenjualan	Char	10
	Nilaipembayaran	Int	15
	Total		35 byte
Orderpembelian	Kdorderpembelian	Char	10
	Tglorderpembelian	Date	
	Kdkaryawan	Char	10
	Kdpemasok	Char	10
	Hargabeli	Int	20
	Kdstatusbayar	Char	10
	Total		60 byte
Detailorderpembelian	Kdorderpembelian	Char	10
	Noreg	Char	10
	Jumlah	int	5
	Total		25 byte
Pembayaranpembelian	Kdpembayaranpembelian	Char	10
	Tglpembayaranpembelian	Date	
	Kdkaryawan	Char	10
	Total		20 byte
Detail Pembayaran Pembelian	Kdpembayaranpembelian	Char	10
	Kdorderpembelian	Char	10
	Nilaipembayaran	Int	15
	Total		35 byte
Statusbayar	Kdstatusbayar	Char	10
	statusbayar	Varchar	15
	Total		25 byte

Dalam satu bulan entitas mengalami perubahan jumlah pemakaian penyimpanan dengan estimasi seperti tabel 3.34.

Tabel 3.33. Ukuran Entitas Per Bulan

Entitas	Ukuran	Estimasi Transaksi Per Bulan	Total Ukuran per Bulan
Karyawan	170	5	850
Hakakses	35	2	70
Pemasok	85	3	255
Pasien	85	5	425
Kontak	45	5	225
Tipekontak	35	5	175
Kacamata	90	8	720
Satuan	25	8	200
Orderpenjualan	60	300	18000
Statusbayar	25	1	25
detailorderpenjualan	25	300	7500
Pembayaranpenjualan	20	80	1600
detailpembayaranpenjualan	35	80	2800
Orderpembelian	60	150	9000
Detailorderpembelian	25	150	3750
Pembayaranpembelian	20	80	1600
DetailPembayaranpembelian	35	80	2800
	Total per bulan		49995

Dari hasil estimasi diatas diperkirakan data disk per tahun sebesar 49995 x 12 = 599940 *byte*/tahun.

3.4.3.3. Merancang Mekanisme Keamanan

Data merupakan sebuah aset yang sangat berharga bagi perusahaan, oleh karena itu perlu diatur mekanisme keamanan data. Adapun mekanisme keamanan data dijabarkan pada tabel rancangan mekanisme keamanan.

Tabel 3.34. Rancangan Mekanisme Keamanan

Relasi/View	Admin/Pemilik	Penjualan	Pembelian
Karyawan	S/I/U/D	S	S
Pemasok	S/I/U/D		S/I/U/D
Pasien	S/I/U/D	S/I/U/D	
Hakakses	S/I/U/D		
Kontak	S/I/U/D	S	S

Tipekontak	S/I/U/D	S	S
Kacamata	S/I/U/D	S	S/I/U/D
Satuan	S/I/U/D	S	S/I
Orderpenjualan	S/I/U/D	S/I	
Detailorderpenjualan	S/I/U/D	S/I	
Statusbayar	S/I/U/D	S	S
Pembayaranpenjualan	S/I/U/D	S/I	
Detailpembayaranpenjualan	S/I/U/D	S/I	

Tabel 3.35. Rancangan Mekanisme Keamanan (Lanjutan)

Relasi/View	Admin/Pemilik	Penjualan	Pembelian
Orderpembelian	S/I/U/D		S/I
Detailorderpembelian	S/I/U/D		S/I
Pembayaranpembelian	S/I/U/D		S/I
Detailpembayaranpembelian	S/I/U/D		S/I

ket :

- Membaca (SELECT/S)
- Menambah (INSERT/I)
- Mengubah (UPDATE/U)
- Menghapus (DELETE/D)

Dengan pemberian mekanisme keamanan seperti diatas, maka rancangan keamanan yang terbentuk akan lebih jelas. Dimana untuk bagian penjualan hanya bisa mengakses basis data untuk penjualan saja dan begitu juga untuk bagian pembelian hanya bisa mengakses kebutuhan akan basis data yang mendukung pembelian saja. dan untuk admin/pimpinan diberi hak akses keseluruhan.

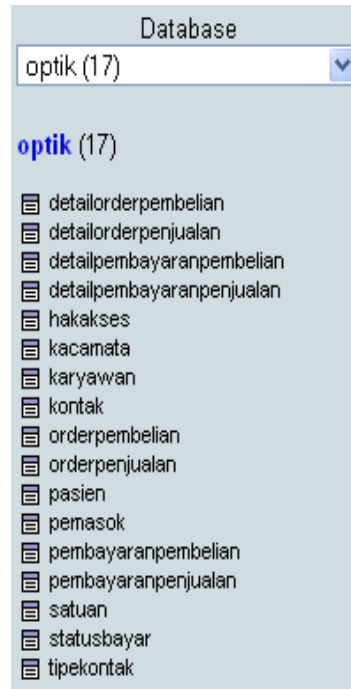
BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil

Adapun hasil dari penelitian yang dilakukan ini ialah berupa rancangan basis data. Rancangan basis data ini memiliki tabel-tabel yang telah terbentuk dari proses fase-fase perancangan *konseptual*, *logical* sampai *fisikal* berbentuk *prototype* menggunakan aplikasi *MySQL* atau *Phpmyadmin* yang telah diuraikan di bab sebelumnya dan diharapkan menjadi rancangan basis data yang baik, sehingga nantinya bisa dimanfaatkan dan untuk mempermudah perusahaan dalam pembuatan aplikasi penjualan dan pembelian pada Rumah Sakit Sriwijaya Eye Center Palembang nantinya.

Hasil dari perancangan basis data ini dihasilkan 17 struktur tabel basis data dan dimasukkan ke dalam *database* yang bernama *penjualan_pembelian*. Pengimplementasian dari rancangan basis data yang terdiri atas 17 struktur tabel yang ada dilakukan dengan menggunakan aplikasi *XAMPP* dengan koneksi *server localhost*.



Gambar 4.1. *Database optik*

4.2. Pembahasan

Berikut ini akan dibahas mengenai rancangan basis data yang telah dibuat, pembahasan ini mencakup 17 struktur tabel yang telah terbentuk dengan disertai dengan tampilan tiap struktur tabel yang ada pada *database optik*.

4.2.1. Tabel Master

4.2.1.1. Tabel Kacamata

Tabel kacamata merupakan tabel yang digunakan untuk menyimpan data kacamata yang ada di perusahaan. Tabel ini terdiri dari 6 atribut yaitu noreg, merk, hargajual, hargabeli, jumlah, kdsatuan. *Primary key* dari tabel ini ialah noreg. Dan *foreign key* dari tabel ini ialah kdsatuan.

Tabel 4.1. Struktur Tabel Kacamata

Nama Atribut	Key	Type	Width
Noreg	Primary key	char	10
Merk		varchar	25
Hargabeli		Int	15
Hargajual		Int	15
Jumlah		int	15
Warnaframe		Varchar	25
Ukuranlensa		Varchar	25
Kdsatuan	Foreign key	char	10
		Total	75

SQL *syntax create table* pada tabel kacamata :

```
--
-- Table structure for table `kacamata`
--
CREATE TABLE IF NOT EXISTS `kacamata` (
  `Noreg` char(10) NOT NULL,
  `Merk` varchar(25) NOT NULL,
  `Hargabeli` int(15) NOT NULL,
  `Hargajual` int(15) NOT NULL,
  `Warnaframe` varchar(25) NOT NULL,
  `Ukuranlensa` varchar(25) NOT NULL,
  `Jumlah` int(15) NOT NULL,
  `Kdsatuan` char(10) NOT NULL,
  PRIMARY KEY (`Noreg`),
  UNIQUE KEY `Kdsatuan` (`Kdsatuan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel kacamata di xampp.

Field	Type	Collation	Attributes	Null	Default	Extra
Noreg	char(10)	latin1_swedish_ci		No		
Merk	varchar(25)	latin1_swedish_ci		No		
Hargabeli	int(15)			No		
Hargajual	int(15)			No		
Jumlah	int(15)			No		
Warnaframe	varchar(25)	latin1_swedish_ci		No		
Ukuranlensa	varchar(25)	latin1_swedish_ci		No		
Kdsatuan	char(10)	latin1_swedish_ci		No		

Tabel 5.2. Hasil struktur Tabel Kacamata

4.2.1.2. Tabel Satuan Kacamata

Tabel satuan kacamata merupakan tabel yang digunakan untuk menyimpan data satuan kacamata yang dijual oleh perusahaan. Tabel ini terdiri dari 2 atribut yaitu kdsatuan dan satuan. *Primary key* dari tabel ini ialah kdsatuan.

Tabel 4.3. Struktur Tabel Satuan Kacamata

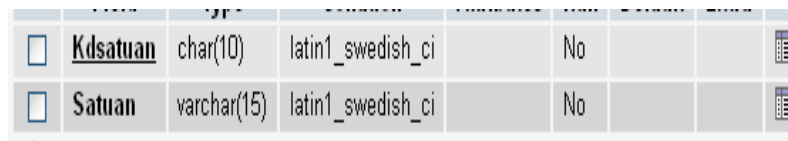
Nama Atribut	Key	Type	width
kdsatuan	<i>Primary key</i>	char	10
Satuan		varchar	15
		Total	25

SQL *syntax create table* pada tabel satuan kacamata :

```
--  
-- Table structure for table `satuan`  
--  
CREATE TABLE IF NOT EXISTS `satuan` (  
  `kdsatuan` char(10) NOT NULL,  
  `satuan` varchar(15) NOT NULL,  
  PRIMARY KEY (`kdsatuan`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel satuan di xampp.

Tabel 4.4. Hasil struktur Tabel Satuan Kacamata



Field	Type	Collation	Index	Uniq	Null	Extra
<input type="checkbox"/> <u>Kdsatuan</u>	char(10)	latin1_swedish_ci		No		
<input type="checkbox"/> Satuan	varchar(15)	latin1_swedish_ci		No		

4.2.1.3. Tabel Pasien

Tabel pasien merupakan tabel yang digunakan untuk menyimpan data pasien yang menjadi pembeli di perusahaan. Tabel ini terdiri dari 3 atribut yaitu kdpasien, namapasien, alamatpasien. *Primary key* dari tabel ini ialah kdpasien.

Tabel 4.5. Struktur Tabel Pasien

Nama Atribut	Key	Type	width
Kdpasien	<i>Primary key</i>	char	10
Namapasien		varchar	25
Alamatpasien		Varchar	50
		Total	85






















SQL *syntax create table* pada tabel pasien :

```
--
-- Table structure for table `pasien`
--

CREATE TABLE IF NOT EXISTS `pasien` (
  `kdpasien` char(10) NOT NULL,
  `namapasien` varchar(25) NOT NULL,
  `alamatpasien` varchar(50) NOT NULL,
  PRIMARY KEY (`kdpasien`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel pasien di xampp.

Tabel 4.6. Hasil struktur Tabel Pasien

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	<u>Kdpasien</u>	char(10)	latin1_swedish_ci		No			      
<input type="checkbox"/>	<u>Namapasien</u>	varchar(25)	latin1_swedish_ci		No			      
<input type="checkbox"/>	<u>Alamatpasien</u>	varchar(25)	latin1_swedish_ci		No			      

4.2.1.4. Tabel Karyawan

Tabel karyawan merupakan tabel yang digunakan untuk menyimpan data karyawan yang menjadi karyawan di perusahaan. Tabel ini terdiri dari 6 atribut yaitu kdkaryawan, namakaryawan, alamatkaryawan, *username*, *password*, kdhakakses. *Primary key* dari struktur tabel ini ialah kdkaryawan. *Foreign key* dari struktur tabel ini ialah kdhakakses.

Tabel 4.7. Struktur Tabel Karyawan

Nama Atribut	Key	Type	Width
Kdkaryawan	Primay key	char	10
Namakaryawan		varchar	25
Alamatkaryawan		Varchar	50
<i>Username</i>		Varchar	25
<i>password</i>		Varchar	15
Kdhakases	<i>Foreign key</i>	Char	10
		Total	135

SQL *syntax create table* pada tabel karyawan :

```
--
-- Table structure for table `karyawan`
--
CREATE TABLE IF NOT EXISTS `karyawan` (
  `Kdkaryawan` char(10) NOT NULL,
  `Namakaryawan` varchar(25) NOT NULL,
  `Alamatkaryawan` varchar(50) NOT NULL,
  `Username` varchar(25) NOT NULL,
  `Password` varchar(15) NOT NULL,
  `Kdhakakses` char(10) NOT NULL,
  PRIMARY KEY (`Kdkaryawan`),
  UNIQUE KEY `Kdhakakses` (`Kdhakakses`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel karyawan di xampp.

Tabel 4.8. Hasil struktur Tabel Karyawan

#	Column	Type	Collation	Attributes	Null	Default	Extra
1	<u>Kdkaryawan</u>	char(10)	latin1_swedish_ci		No	None	
2	Namakaryawan	varchar(25)	latin1_swedish_ci		No	None	
3	Alamatkaryawan	varchar(50)	latin1_swedish_ci		No	None	
4	Username	varchar(25)	latin1_swedish_ci		No	None	
5	Password	varchar(15)	latin1_swedish_ci		No	None	
6	Kdhakakses	char(10)	latin1_swedish_ci		No	None	

4.2.1.5. Tabel Hak Akses

Tabel hak akses merupakan tabel yang digunakan untuk menyimpan data dari hak akses di perusahaan. Tabel ini terdiri dari 2 atribut yaitu kdhakakses dan hakakses. *Primary key* dari tabel ini ialah kdhakakses.

Tabel 4.9. Struktur Tabel Hak Akses

Nama Atribut	Key	Type	Width
Kdhakakses	<i>Primary key</i>	Char	10
hakakses		Varchar	15
		Total	30

SQL *syntax create table* pada tabel hak akses :

```
--
-- Table structure for table `hakakses`
--
CREATE TABLE IF NOT EXISTS `hakakses` (
```



```

`Kdhakakses` char(10) NOT NULL,
`hakakses` varchar(15) NOT NULL,
PRIMARY KEY (`Kdhakakses`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel hak akses di xampp.

Tabel 4.10. Hasil struktur Tabel hak akses

#	Column	Type	Collation	Attributes	Null	Default	Extra
1	<u>Kdhakakses</u>	char(10)	latin1_swedish_ci		No	None	
2	hakakses	varchar(15)	latin1_swedish_ci		No	None	

4.2.1.6. Tabel Pemasok

Tabel pemasok merupakan tabel yang digunakan untuk menyimpan data pemasok yang menjadi pemasok kaca mata kepada perusahaan. Tabel ini terdiri dari 3 atribut yaitu kdpemasok, namapemasok, alamatpemasok. *Primary key* dari tabel ini ialah kdpemasok.

Tabel 4.11. Struktur Tabel Pemasok

Nama Atribut	Key	Type	Width
Kdpemasok	<i>Primary key</i>	char	10
Namapemasok		varchar	25
Alamatpemasok		Varchar	50
		Total	85

SQL *syntax create table* pada tabel pemasok :

```

--
-- Table structure for table `pemasok`
--

CREATE TABLE IF NOT EXISTS `pemasok` (
  `Kdpemasok` char(10) NOT NULL,
  `Namapemasok` varchar(25) NOT NULL,
  `Alamatpemasok` varchar(50) NOT NULL,
  PRIMARY KEY (`Kdpemasok`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel pemasok di xampp.

Tabel 4.12. Hasil struktur Tabel Pemasok

#	Column	Type	Collation	Attributes	Null	Default	Extra
1	<u>Kdpemasok</u>	char(10)	latin1_swedish_ci		No	None	
2	Namapemasok	varchar(25)	latin1_swedish_ci		No	None	
3	Alamatpemasok	varchar(50)	latin1_swedish_ci		No	None	

4.2.1.7. Tabel Kontak

Tabel kontak merupakan tabel yang digunakan untuk menyimpan data dari kontak karyawan, pasien dan pemasok di perusahaan. Tabel ini terdiri dari 3 atribut yaitu kdkontak, nokontak, kdtipekontak. *Primary key* dari tabel ini ialah kdkontak dan atribut kdtipekontak bertindak sebagai *foreign key*.

Tabel 4.13. Struktur Tabel Kontak

Nama Atribut	Key	Type	Width
Kdkontak	<i>Primary key</i>	Char	10
Nokontak		Varchar	25
Kdtipekontak	<i>Foreign key</i>	Char	10
	Total		45

SQL *syntax create table* pada tabel kontak :

```
--
-- Table structure for table `kontak`
--
CREATE TABLE IF NOT EXISTS `kontak` (
  `kdkontak` char(10) NOT NULL,
  `nokontak` varchar(25) NOT NULL,
  `kdtipekontak` char(10) NOT NULL,
  PRIMARY KEY (`kdkontak`),
  UNIQUE KEY `kdtipekontak` (`kdtipekontak`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel kontak di xampp.

Tabel 4.14. Hasil struktur Tabel Kontak

#	Column	Type	Collation	Attributes	Null	Default	Extra
1	<u>Kdkontak</u>	char(10)	latin1_swedish_ci		No	None	
2	Nokontak	varchar(25)	latin1_swedish_ci		No	None	
3	Kdtipekontak	char(10)	latin1_swedish_ci		No	None	

4.2.1.8. Tabel Tipe Kontak

Tabel tipe kontak merupakan tabel yang digunakan untuk menyimpan data dari tipe kontak dari karyawan, pasien dan pemasok di perusahaan. Tabel ini terdiri dari 2 atribut yaitu kdtipekontak, tipekontak. *Primary key* dari tabel ini ialah kdtipekontak .

Tabel 4.15. Struktur Tabel Tipe Kontak

Nama Atribut	Key	Type	Width
Kdtipekontak	<i>Primary key</i>	Char	10
Tipekontak		Vachar	25
		Total	35

SQL *syntax create table* pada tabel tipe kontak :

```
--  
-- Table structure for table `tipekontak`  
--  
CREATE TABLE IF NOT EXISTS `tipekontak` (  
  `kdtipekontak` char(10) NOT NULL,  
  `Tipekontak` varchar(25) NOT NULL,  
  PRIMARY KEY (`kdtipekontak`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel tipe kontak di xampp.

Tabel 4.16. Hasil struktur Tabel Tipe Kontak

#	Column	Type	Collation	Attributes	Null	Default	Extra
1	<u>Kdtipekontak</u>	char(10)	latin1_swedish_ci		No	None	
2	<u>Tipekontak</u>	varchar(25)	latin1_swedish_ci		No	None	

4.2.2. Tabel Penjualan

4.2.2.1. Tabel Order Penjualan

Tabel order penjualan merupakan tabel yang digunakan untuk menyimpan data orderpenjualan yang dilakukan konsumen kepada perusahaan. Tabel ini terdiri dari 6 atribut yaitu kdorderpenjualan, tglorderpenjualan, kdkaryawan,

kd pasien, harga jual, kd status bayar. *Primary key* dari tabel ini ialah kd order penjualan. Dan *foreign key* dari tabel ini ialah atribut kd karyawan, kd pasien dan kd status bayar.

Tabel 4.17. Struktur Tabel Order penjualan

Nama Atribut	Key	Type	Width
Kdorderpenjualan	Primary key	Char	10
Tglorderpenjualan		Date	
Kdkaryawan	Foreign key	Char	10
Kdpasien	Foreign key	Char	10
Harga jual		Int	20
Kdstatusbayar	Foreign key	Char	10
	Total		60

SQL syntax create table pada tabel order penjualan:

```
--
-- Table structure for table `orderpenjualan`
--
CREATE TABLE IF NOT EXISTS `orderpenjualan` (
  `kdorderpenjualan` char(10) NOT NULL,
  `tglorderpenjualan` date NOT NULL,
  `kdkaryawan` char(10) NOT NULL,
  `kdpasien` char(10) NOT NULL,
  `hargajual` int(20) NOT NULL,
  `kdstatusbayar` char(10) NOT NULL,
  PRIMARY KEY (`kdorderpenjualan`),
  KEY `kdstatusbayar` (`kdstatusbayar`),
  KEY `kdpasien` (`kdpasien`),
  KEY `kdkaryawan` (`kdkaryawan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Hasil dari sql syntax create table diatas menghasilkan struktur tabel order penjualan di xampp.

Tabel 4.18. Hasil struktur Tabel order penjualan

Field	Type	Collation	Attributes	Null	Default	Extra
Kdorderpenjualan	char(10)	latin1_swedish_ci		No		
Tglorderpenjualan	date			No		
Kdkaryawan	char(10)	latin1_swedish_ci		No		
Kdpasien	char(10)	latin1_swedish_ci		No		
Hargajual	int(20)			No		
Kdstatusbayar	char(10)	latin1_swedish_ci		No		

4.2.2.2. Tabel detail order penjualan

Tabel detailorderpenjualan merupakan tabel yang digunakan untuk menyimpan data detailorderpenjualan yang dilakukan konsumen kepada perusahaan. Tabel ini terdiri dari 3 atribut yaitu kdorderpenjualan, noreg, jumlah.

Primary key dari tabel ini ialah *korderpenjualan*. Dan *foreign key* dari tabel ini ialah atribut *noreg*.

Tabel 4.19. Struktur Tabel Detail order penjualan

Nama Atribut	Key	Type	Width
Kdorderpenjualan	<i>Primary key</i>	Char	10
Noreg		Char	10
Jumlah		Int	5
		Total	25

SQL *syntax create table* pada tabel detail order penjualan:

```
--
-- Table structure for table `detailorderpenjualan`
--

CREATE TABLE IF NOT EXISTS `detailorderpenjualan` (
  `korderpenjualan` char(10) NOT NULL,
  `noreg` char(10) NOT NULL,
  `jumlah` int(5) NOT NULL,
  PRIMARY KEY (`korderpenjualan`),
  UNIQUE KEY `noreg` (`noreg`,`korderpenjualan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel detail order penjualan di xampp.

Tabel 4.20. Hasil struktur Tabel detail order penjualan

Field	Type	Collation	Attributes	Null	Default	Extra
Kdorderpenjualan	char(10)	latin1_swedish_ci		No		
Noreg	char(10)	latin1_swedish_ci		No		
Jumlah	int(5)			No		

4.2.2.3. Tabel Status Bayar

Tabel status bayar merupakan tabel yang digunakan untuk menyimpan data status pembayaran yang dilakukan konsumen kepada perusahaan. Tabel ini terdiri dari 2 atribut yaitu *kdstatusbayar*, *statusbayar*. *Primary key* dari tabel ini ialah *kdstatusbayar*.

Tabel 4.21. Struktur Tabel Status Bayar

Nama Atribut	Key	Type	Width
Kdstatusbayar	<i>Primary key</i>	Char	10
Statusbayar		Varchar	10
		Total	20

SQL *syntax create table* pada tabel status bayar:

```
--
-- Table structure for table `statusbayar`
--

CREATE TABLE IF NOT EXISTS `statusbayar` (
  `kdstatusbayar` char(10) NOT NULL,
  `Statusbayar` varchar(10) NOT NULL,
  PRIMARY KEY (`kdstatusbayar`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel status bayar di xampp.

Tabel 4.22. Hasil struktur Tabel status bayar

#	Column	Type	Collation	Attributes	Null	Default	Extra
1	<u>Kdstatusbayar</u>	char(10)	latin1_swedish_ci		No	None	
2	Statusbayar	varchar(10)	latin1_swedish_ci		No	None	

4.2.2.4. Tabel Pembayaran penjualan

Tabel pembayaran penjualan merupakan tabel yang digunakan untuk menyimpan data pembayaran kacamata dari konsumen kepada perusahaan. Tabel ini terdiri dari 3 atribut yaitu kdpembayaranpenjualan, tglpembayaranpenjualan, kdkaryawan. *Primary key* dari tabel ini ialah kdpembayaranpenjualan. Dan *foreign key* dari tabel ini ialah atribut kdkaryawan.

Tabel 4.23. Struktur Tabel Pembayaran penjualan

Nama Atribut	Key	Type	Width
Kdpembayaranpenjualan	<i>Primary key</i>	Char	10
Tglpembayaranpenjualan		Date	
Kdkaryawan	<i>Foreign key</i>	Char	10
		Total	20

SQL *syntax create table* pada tabel Pembayaran penjualan :

```
--
-- Table structure for table `pembayaranpenjualan`
--

CREATE TABLE IF NOT EXISTS `pembayaranpenjualan` (
  `kdpembayaranpenjualan` char(10) NOT NULL,
  `Tglpembayaranpenjualan` date NOT NULL,
  `kdkaryawan` char(10) NOT NULL,
```

```

PRIMARY KEY (`kdpembayaranpenjualan`),
UNIQUE KEY `kdkaryawan` (`kdkaryawan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel Pembayaran penjualan di xampp.

Tabel 4.24. Hasil struktur tabel Pembayaran penjualan

#	Column	Type	Collation	Attributes	Null	Default	Extra
1	<u>Kdpembayaranpenjualan</u>	char(10)	latin1_swedish_ci		No	None	
2	Tglpembayaranpenjualan	date			No	None	
3	Kdkaryawan	char(10)	latin1_swedish_ci		No	None	

4.2.2.5. Tabel Detail Pembayaran Penjualan

Tabel detailpembayaranpenjualan merupakan tabel yang digunakan untuk menyimpan detail data pembayaran kacamata dari konsumen kepada perusahaan. Tabel ini terdiri dari 3 atribut yaitu kdpembayaranpenjualan, kdorderpenjualan, nilaipembayaran. *Primary key* dari tabel ini ialah kdpembayaranpenjualan. Dan *foreign key* dari tabel ini ialah atribut kdorderpenjualan.

Tabel 4.25. Struktur Tabel Detail pembayaran penjualan

Nama Atribut	Key	Type	Width
Kdpembayaranpenjualan	<i>Primary key</i>	Char	10
Kdorderpenjualan	<i>Foreign key</i>	Char	10
Nilaipembayaran		Int	15
		Total	35

SQL *syntax create table* pada tabel detail Pembayaran penjualan :

```

--
-- Table structure for table `detailpembayaranpenjualan`
--

CREATE TABLE IF NOT EXISTS `detailpembayaranpenjualan` (
  `kdpembayaranpenjualan` char(10) NOT NULL,
  `kdorderpenjualan` char(10) NOT NULL,
  `nilaipembayaran` int(15) NOT NULL,
  PRIMARY KEY (`kdpembayaranpenjualan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel detail Pembayaran penjualan di xampp.

Tabel 4.26. Hasil struktur tabel detail pembayaran penjualan

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<u>kdpembayaranpenjualan</u>	char(10)	latin1_swedish_ci		No	None	
2	kdorderpenjualan	char(10)	latin1_swedish_ci		No	None	
3	nilaipembayaran	int(15)			No	None	

4.2.3. Tabel Pembelian

4.2.3.1. Tabel Order pembelian

Tabel order pembelian merupakan tabel yang digunakan untuk menyimpan data order pembelian yang dilakukan oleh perusahaan ke pemasok. Tabel ini terdiri dari 6 atribut yaitu kdorderpembelian, tglorderpembelian, kdkaryawan, kdpemasok, hargabeli, kdstatusbayar. *Primary key* dari tabel ini ialah kdorderpembelian. Dan *foreign key* dari tabel ini ialah atribut kdkaryawan, kdpemasok dan kdstatusbayar.

Tabel 4.27. Struktur Tabel Order Pembelian

Nama Atribut	Key	Type	Width
Kdorderpembelian	<i>Primary key</i>	Char	10
Tglorderpembelian		Date	
Kdkaryawan	<i>Foreign key</i>	Char	10
Kdpemasok	<i>Foreign key</i>	Char	10
yHargabeli		Int	20
Kdstatusbayar	<i>Foreign key</i>	Char	10
		Total	60

SQL *syntax create table* pada tabel Order Pembelian :

```
--
-- Table structure for table `orderpembelian`
--

CREATE TABLE IF NOT EXISTS `orderpembelian` (
  `kdorderpembelian` char(10) NOT NULL,
  `tglorderpembelian` date NOT NULL,
  `kdkaryawan` char(10) NOT NULL,
  `kdpemasok` char(10) NOT NULL,
  `hargabeli` int(20) NOT NULL,
  `kdstatusbayar` char(10) NOT NULL,
```



```

PRIMARY KEY (`kdorderpembelian`),
KEY `kdpemasok` (`kdpemasok`),
KEY `kdstatusbayar` (`kdstatusbayar`),
KEY `kdkaryawan` (`kdkaryawan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel Order Pembelian di xampp.

Tabel 4.28. Hasil struktur tabel Order Pembelian

#	Column	Type	Collation	Attributes	Null	Default	Extra
1	Kdorderpembelian	char(10)	latin1_swedish_ci		No	None	
2	Tglorderpembelian	date			No	None	
3	Kdkaryawan	char(10)	latin1_swedish_ci		No	None	
4	Kdpemasok	char(10)	latin1_swedish_ci		No	None	
5	Hargabeli	int(20)			No	None	
6	Kdstatusbayar	char(10)	latin1_swedish_ci		No	None	

4.2.3.2. Tabel Detail Order Pembelian

Tabel detail order pembelian merupakan tabel yang digunakan untuk menyimpan data detailorderpembelian yang dilakukan oleh perusahaan kepada pemasok. Tabel ini terdiri dari 3 atribut yaitu kdorderpembelian, noreg, jumlah. *Primary key* dari tabel ini ialah kdorderpembelian. Dan *foreign key* dari tabel ini ialah atribut noreg.

Tabel 4.29. Struktur Tabel Detail order pembelian

Nama Atribut	Key	Type	Width
Kdorderpembelian	Primary key	Char	10
Noreg	Foreign key	Char	10
Jumlah		int	5
		Total	25

SQL *syntax create table* pada tabel detail Order Pembelian :

```

--
-- Table structure for table `detailorderpembelian`
--

CREATE TABLE IF NOT EXISTS `detailorderpembelian` (
  `kdorderpembelian` char(10) NOT NULL,
  `noreg` char(10) NOT NULL,
  `jumlah` int(5) NOT NULL,
  PRIMARY KEY (`kdorderpembelian`),
  UNIQUE KEY `noreg` (`noreg`,`kdorderpembelian`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel Detail Order Pembelian di xampp.

Tabel 4.30. Hasil struktur tabel Detail Order Pembelian

Field	Type	Collation	Attributes	Null	Default	Extra
Kdorderpembelian	char(10)	latin1_swedish_ci		No		
Noreg	char(10)	latin1_swedish_ci		No		
Jumlah	int(5)			No		

4.2.3.3. Tabel Pembayaran Pembelian

Tabel pembayaran pembelian merupakan tabel yang digunakan untuk menyimpan data pembayaran kacamata dari perusahaan ke pemasok. Tabel ini terdiri dari 3 atribut yaitu kdpembayaranpembelian, tglpembayaranpembelian, kdkaryawan. *Primary key* dari tabel ini ialah kdpembayaranpembelian. Dan *foreign key* dari tabel ini ialah atribut kdkaryawan.

Tabel 4.31. Struktur Tabel Pembayaran pembelian

Nama Atribut	Key	Type	Width
Kdpembayaranpembelian	Primary key	Char	10
Tglpembayaranpembelian		Date	
Kdkaryawan	Foreign key	Char	10
		Total	20

SQL *syntax create table* pada tabel Pembayaran Pembelian :

```
--
-- Table structure for table `pembayaranpembelian`
--
CREATE TABLE IF NOT EXISTS `pembayaranpembelian` (
  `kdpembayaranpembelian` char(10) NOT NULL,
  `tglpembayaranpembelian` date NOT NULL,
  `kdkaryawan` char(10) NOT NULL,
  PRIMARY KEY (`kdpembayaranpembelian`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel Pembayaran Pembelian di xampp.

Tabel 4.32. Hasil struktur tabel Pembayaran Pembelian

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	kdpembayaranpembelian	char(10)	latin1_swedish_ci		No	None	
2	tglpembayaranpembelian	date			No	None	
3	kdkaryawan	char(10)	latin1_swedish_ci		No	None	

4.2.3.4. Tabel Detail Pembayaran Pembelian

Tabel detail pembayaran pembelian merupakan tabel yang digunakan untuk menyimpan detail data pembayaran kacamata dari perusahaan ke pemasok. Tabel ini terdiri dari 3 atribut yaitu kdpembayaranpembelian, kdorderpembelian, nilaipembayaran. *Primary key* dari tabel ini ialah kdpembayaranpembelian. Dan *foreign key* dari tabel ini ialah atribut kdorderpembelian.

Tabel 4.33. Struktur Tabel Detail pembayaran pembelian

Nama Atribut	Key	Type	Width
Kdpembayaranpembelian	Primary key	Char	10
Kdorderpembelian	Foreign key	Char	10
Nilaipembayaran		Int	20
		Total	40

SQL *syntax create table* pada tabel Detail pembayaran pembelian :

```
CREATE TABLE IF NOT EXISTS `detailpembayaranpembelian` (  
  `kdpembayaranpembelian` char(10) NOT NULL,  
  `kdorderpembelian` char(10) NOT NULL,  
  `nilaipembayaran` int(20) NOT NULL,  
  PRIMARY KEY (`kdpembayaranpembelian`),  
  KEY `kdorderpembelian` (`kdorderpembelian`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Hasil dari sql *syntax create table* diatas menghasilkan struktur tabel Detail pembayaran pembelian di xampp.

Tabel 4.34. Hasil struktur tabel Detail pembayaran pembelian

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 kdpembayaranpembelian	char(10)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	2 kdorderpembelian	char(10)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	3 nilaipembayaran	int(20)			No	None	

4.2.4. Trigger

Trigger pada rancangan basis data penjualan dan pembelian pada penelitian ini digunakan dalam manipulasi data stok kacamata sehingga apabila melakukan transaksi penjualan maupun transaksi pembelian, jumlah atau *stok* yang ada pada tabel kacamata akan otomatis berubah.

4.2.4.1. Trigger Pembelian_Kacamata

Didalam *trigger* pembelian_kacamata yang digunakan ialah tabel detailorderpembelian dan tabel kacamata.

Pertama membuat *trigger* pembelian_kacamata ;

SQL syntax create trigger pembelian_kacamata

```
CREATE TRIGGER pembelian_kacamata after INSERT ON
detailorderpembelian
FOR EACH ROW BEGIN
INSERT INTO kacamata SET
noreg = NEW.noreg
, jumlah=New.jumlah
ON DUPLICATE KEY UPDATE jumlah=jumlah+New.jumlah;
END$$
```

Setelah membuat *trigger* pembelian_kacamata, Skenarionya jika terdapat data pembelian (*insert* pada tabel detailorderpembelian) maka jumlah atau *stok* pada tabel kacamata akan otomatis bertambah (*update* pada tabel kacamata).

Tabel 4.35. Tabel kacamata sebelum *insert* data

Noreg	Merk	Hargabeli	Hargajual	Jumlah	Warnaframe	Ukuranlensa	Kdsatuan
B01	Single Visions Lens	470000	525000	100	Silver Abu-abu	28 mm	Pcs

Tabel 4.36. Tabel detailorderpembelian ketika dilakukan *insert* data

Kdorderpembelian	Noreg	Jumlah
ODR01	B01	50

Tabel 4.37. Tabel kacamata setelah *insert* data di tabel detailorderpembelian

Noreg	Merk	Hargabeli	Hargajual	Jumlah	Warnaframe	Ukuranlensa	Kdsatuan
B01	Single Visions Lens	470000	525000	100	Silver Abu-abu	28 mm	Pcs

4.2.4.2. Trigger Penjualan_Kacamata

Didalam *trigger* penjualan_kacamata yang digunakan ialah tabel detailorderpenjualan dan tabel kacamata.

Pertama membuat *trigger* penjualan_kacamata ;

SQL syntax create trigger penjualan_kacamata

```

CREATE TRIGGER penjualan_kacamata after INSERT ON
detailorderpenjualan
FOR EACH ROW BEGIN
UPDATE kacamata
SET jumlah = jumlah - NEW.jumlah
WHERE
noreg = NEW.noreg;
END$$

```

Setelah membuat *trigger* penjualan_kacamata, Skenarionya jika terdapat data penjualan (*insert* pada tabel detailorderpenjualan) maka jumlah atau stok pada tabel kacamata akan otomatis berkurang (*update* pada tabel kacamata).

Tabel 4.38. Tabel kacamata sebelum *insert* data

Noreg	Merk	Hargabeli	Hargajual	Jumlah	Warnaframe	Ukuranlensa	Kdsatuan
B01	Single Visions Lens	470000	525000	100	Silver Abu-abu	28 mm	Pcs

Tabel 4.39. Tabel detailorderpenjualan ketika dilakukan *insert* data

Kdorderpenjualan	Noreg	Jumlah
ODJ01	B01	35

Tabel 4.40. Tabel kacamata setelah *insert* data

Noreg	Merk	Hargabeli	Hargajual	Jumlah	Warnaframe	Ukuranlensa	Kdsatuan
B01	Single Visions Lens	470000	525000	100	Silver Abu-abu	28 mm	Pcs

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan dari penelitian ini, penulis dapat mengambil kesimpulan sebagai berikut :

1. Dengan dilakukan Rancangan basis data diharapkan dapat menghasilkan suatu rancangan basis data penjualan dan pembelian dengan skema basis data yang baik dan dapat meminimalisir duplikasi data.
2. Hasil dari rancangan basis data ini menghasilkan struktur basis data, dengan nama *database* optik dan terdiri atas 17 struktur tabel.

5.2. Saran

Dari kesimpulan diatas, ada beberapa saran yang penulis ingin sampaikan

1. Dengan rancangan basis data yang telah dihasilkan, perusahaan bisa memanfaatkan basis data tersebut dalam pembuatan suatu aplikasi penjualan dan pembelian pada Optik Sriwijaya Eye Center Palembang.
2. Penelitian ini disarankan dapat dipergunakan dan dikembangkan untuk penelitian selanjutnya.