

# Perancangan Basis Data Perusahaan Distribusi dengan Menggunakan Oracle

J. I. Maanari<sup>(1)</sup>, R. Sengkey, ST., MT.<sup>(2)</sup>, Ir. H. F. Wowor, M.Kom.<sup>(3)</sup>, Y. D. Y. Rindengan, ST., MM., Msc.<sup>(4)</sup>.

(1)Mahasiswa (2)Pembimbing 1 (3)Pembimbing 2 (4)Pembimbing 3

jngwiemalmsteen@gmail.com rizal.sengkey@unsrat.ac.id hans.wowor@unsrat.ac.id  
yaulie.rindengan@unsrat.ac.id

Jurusan Teknik Elektro-FT, UNSRAT, Manado-95115

## Abstract

*Oracle is a DBMS which have sophisticate feature for effective and efficiently database managing. However database design that will be implemented to DBMS have significant role and of course cannot be ignored. A well designed database will make its implementation on application become easier besides increase the performance of DBMS itself.*

*The database design in this final project is using a case study of Distribution Company. The design has implemented in Oracle-based database management system integrated with information system application.*

**Keyword :** Database Design, Database Management System, Distribution Company, Oracle.

## Abstrak

Oracle merupakan sebuah sistem manajemen basis data yang dikenal memiliki fitur dan kecanggihan yang membuat pengelolaan basis data menjadi efisien dan efektif. Namun perancangan basis data yang diterapkan pada suatu sistem manajemen basis data tidak kalah pentingnya. Perancangan basis data yang baik akan mempermudah implementasi aplikasi serta mengoptimalkan kinerja dari sistem manajemen basis data itu sendiri.

Pada tugas akhir ini di lakukan perancangan basis data dengan menggunakan contoh kasus perusahaan distribusi dan mengimplementasikan perancangan tersebut pada sistem manajemen basis data berbasis Oracle yang terintegrasi dengan aplikasi sistem informasi.

**Kata kunci:** Oracle, Perancangan Basis Data, Perusahaan Distribusi, Sistem Manajemen Basis Data.

## I. PENDAHULUAN

Suatu sistem informasi yang baik harus memiliki komponen basis data yang handal dan sesuai dengan kebutuhan penggunanya. Suatu basis data secara langsung sangat mempengaruhi informasi yang diproduksi oleh sistem informasi, disamping kemampuan dari sistem informasi itu sendiri.

Basis data merupakan kumpulan data dari berbagai entitas yang saling berhubungan. Basis data dikelola melalui sebuah perangkat lunak yang disebut database managements System (DBMS). Melalui DBMS manusia dapat berinteraksi dengan basis data. Salah satu

DBMS paling populer saat ini adalah Oracle, perangkat lunak produksi Oracle USA, Inc yang merupakan aplikasi Relational DBMS dengan berbagai macam fitur-fitur yang dapat mengoptimalkan pengelolaan basis data. Dalam dunia bisnis saat ini, suatu perusahaan (contohnya sebuah distributor) membutuhkan suatu sistem yang mampu menangani dan mengolah data dari proses bisnisnya dengan cepat dan akurat tanpa membuang banyak biaya maupun waktu. Hal ini tentunya juga sangat penting mengingat persaingan yang semakin ketat dalam dunia bisnis. Dengan adanya sebuah sistem informasi yang didukung sebuah basis data yang baik diharapkan mampu menambah daya saing dari perusahaan serta tentunya mewujudkan prinsip ekonomi yaitu menambah keuntungan dan menekan pengeluaran sekecil mungkin.

Dengan kelebihan dari segi performansi dan efisiensi DBMS Oracle sangat cocok dengan proses bisnis perusahaan dengan aktivitas transaksi yang cukup padat seperti perusahaan distribusi. Namun berbagai kecanggihan tersebut akan sia-sia jika basis data yang dibangun memiliki struktur yang asal-asalan. Hal ini sangat mempengaruhi performansi dari DBMS maupun sistem informasi, oleh karena itu diperlukan sebuah perancangan basis data yang benar-benar tepat dan efisien serta mengacu pada proses bisnis yang berjalan, sehingga informasi yang disimpan dan diproduksi oleh sistem informasi mampu memenuhi kebutuhan bisnis yang ada. Dengan alasan inilah penulis mengambil judul "Perancangan Basis Data Perusahaan Distribusi Dengan Menggunakan Oracle".

## II. LANDASAN TEORI

### A. Basis Data

Basis data merupakan kumpulan data yang menjabarkan suatu aktivitas dari satu atau beberapa entitas yang berhubungan. Sebagai contoh basis data sebuah universitas memiliki entitas seperti mahasiswa, staf pengajar, mata kuliah, dan ruang kelas. Hubungan antara entitas tersebut seperti mahasiswa yang mendaftar untuk mengikuti mata kuliah, staf pengajar yang mengajarkan mata kuliah, dan penggunaan ruang kelas untuk suatu mata kuliah menurut (Ramakrishnan & Gehrke, 2007, 3).

## 1. Sistem Manajemen Basis Data

Sistem manajemen basis data atau DBMS, adalah perangkat lunak yang di desain untuk membantu menangani koleksi data dalam jumlah besar yang dibutuhkan dalam sebuah sistem dengan pertumbuhan data yang sangat cepat. DBMS umumnya merupakan sebuah bagian dari komputer sains, tujuan penggunaan dan teknik penggunaannya sangat luas, seperti pada bahasa pemrograman, pemrograman berorientasi objek, sistem operasi, struktur data, pemrograman konkuren, kecerdasan buatan, dan masih banyak lagi. (Ramakrishnan & Gehrke, 2007, 4)

Adapun keuntungan dari menggunakan system manajemen basis data atau DBMS adalah: (Ramakrishnan & Gehrke, 2007, 3)

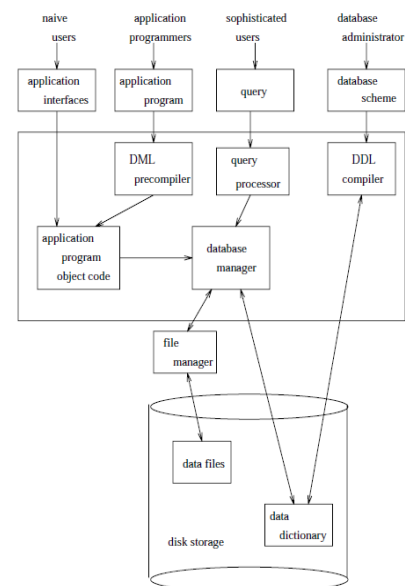
- a) Independensi data  
DBMS mampu mengatur independensi detail representasi dan penyimpanan data dari aplikasi, dengan cara mengabstraksikan data untuk mengisolasi kode program dalam mengakses detail-detail.
- b) Efisiensi data  
DBMS mampu memberikan akses data secara efisien, terlebih jika data tersebut berada pada penyimpanan eksternal, seperti pada sebuah server basis data.
- c) Integritas dan Keamanan Data  
Data yang diakses melewati sebuah DBMS harus mengikuti aturan yang ditentukan tentunya ini menjamin integritas dari data yang digunakan dalam aplikasi, demikian juga dengan hak akses data yang berbeda untuk tiap jenis pengguna data diatur oleh DBMS untuk menjamin keamanan data yang dikelola.
- d) Administrasi Data  
Ketika beberapa user memproduksi data, sangat diperlukan proses administrasi data yang tersentralisasi. Seorang profesional yang mengerti mengenai pentingnya data tersebut diatur dan bagaimana kelompok user mengaksesnya, sangat dibutuhkan dalam menangani administrasi data untuk mencegah redundansi data maupun untuk melakukan berbagai konfigurasi yang membuat pengambilan data menjadi lebih efisien.
- e) Akses Konkuren dan Pemulihan Kerusakan  
DBMS mengatur akses data konkuren sedemikian rupa sehingga user merasa mengakses data yang digunakan hanya oleh 1 user. Lebih jauh lagi sebuah DBMS juga mampu melindungi user dari sebuah kerusakan system.
- f) Mengurangi Waktu Pengembangan Sebuah Perangkat Lunak  
DBMS memiliki berbagai macam fungsi yang biasanya digunakan oleh aplikasi-aplikasi untuk mengakses data yang tersimpan. Dengan kemudahan tersebut pembangunan perangkat lunak tentunya menjadi lebih cepat dan lebih

handal, karena beberapa kegiatan pemrosesan data telah dilakukan oleh DBMS itu sendiri.

Sistem manajemen basis data diolah melalui dua bahasa yaitu DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*). DDL merupakan bahasa yang digunakan untuk menentukan skema basis data sesuai dengan kumpulan ekspresi DDL. Perintah DDL di kompilasi, dan menghasilkan kumpulan tabel yang tersimpan pada sebuah berkas yang disebut data dictionary yang juga berisi metadata. Struktur dan metode akses yang digunakan oleh sistem database. Sedangkan DML merupakan bahasa yang digunakan untuk melakukan manipulasi terhadap database seperti melakukan pengambilan data, memasukkan data, menghapus data, dan memodifikasi data, tujuannya adalah mempermudah manusia untuk berinteraksi dengan sistem. Terdapat 2 tipe DML yakni procedural dimana pengguna menentukan data apa yang diperlukan dan bagaimana data tersebut didapat dan nonprocedural saat pengguna hanya menentukan data apa yang diperlukan. Sebuah bahasa kueri adalah bagian dari DML yang hanya melibatkan pengambilan informasi. Aturan DML dan bahasa kueri biasanya hampir sama. (Zaiane, 1995, 7)

Berikut adalah komponen yang termasuk di dalam struktur sistem basis data (Gambar 1): (Zaiane, 1995, 10)

- a) File Manager bertugas mengatur alokasi kapasitas disk dan struktur data untuk menunjukkan informasi mengenai disk tersebut.
- b) Database Manager merupakan antarmuka antara data low-level dan aplikasi program serta kueri.
- c) Query Processor bertugas menerjemahkan bahasa kueri ke bahasa tingkat rendah yang dimengerti oleh Database Manager.



Gambar 1. Struktur Sistem Basis Data

- d) DML Precompiler mengubah statement DML yang terpatri pada aplikasi menjadi bahasa prosedur pemanggilan standard yang dimengerti oleh host basis data.
- e) DDL Compiler mengubah statement DML menjadi kumpulan tabel berisi metadata yang tersimpan pada data dictionary.
- f) Data Files menyimpan basis data itu sendiri.
- g) Data Dictionary menyimpan informasi mengenai struktur dari basis data.
- h) Database Users Merupakan pengguna basis data seperti application programmer yang berinteraksi dengan sistem melalui DML yang terpatri pada bahasa pemrograman, sophisticated user yang berinteraksi dengan sistem melalui bahasa kueri tanpa menulis program, Specialized User yang merupakan sophisticated user yang menulis program basis data khusus seperti expert system, dan naïve user yang merupakan unsophisticated user yang berinteraksi dengan sistem basis data melalui program jadi seperti pengguna ATM Bank.
- i) Database Administrator adalah orang yang memiliki hak kontrol data dan program yang mengakses data tersebut. Pekerjaan Database Administrator adalah Mendefinisikan scheme, memodifikasi scheme dan organisasi data, mengatur user dan hak akses user, menentukan batasan jenis data untuk menjamin integritas data.

Structured Query Language (SQL) merupakan bahasa yang banyak digunakan dalam berbagai produk database. SQL dibangun di laboratorium IBM-San Jose California sekitar akhir tahun 70-an. Pertama kali dikembangkan sebagai bahasa di produk database DB2 yang sampai saat ini merupakan produk database andalan IBM. SQL sering di lafalkan dengan “sequel”. Saat ini organisasi standar America (ANSI) menetapkan standar bahasa SQL yaitu ANSI-92 standard. Masing-masing vendor database memiliki dialeknya sendiri sebagaimana besar spesifikasinya mengacu pada standar ANSI tersebut dengan berbagai ekstensi tambahan. SQL Server menggunakan bahasa Transact-SQL dalam produknya, sedangkan Oracle menggunakan PL/SQL. (Choirul, 2003, 1)

Oracle merupakan database server dengan model relasional (RDBMS). Sebagai server, Oracle bertugas melayani permintaan client atau proses untuk menggunakan sumberdaya database seperti data, memori, dan proses server. Basis data Oracle dibentuk dari tabel. Dimensi kolom tabel menggambarkan elemen data, field, atau atribut dan dimensi baris mendeskripsikan record yang dibentuk dari kumpulan field atau atribut. (Situmorang, 2004, 1)

Oracle tersusun atas dua komponen utama, yaitu instance dan database (Gambar 2). Kedua komponen ini sangat berbeda, tetapi saling berhubungan dan tidak dapat dipisahkan. Database merupakan kumpulan informasi yang disimpan dalam sebuah penyimpanan fisik, sedangkan instance adalah kumpulan dari Oracle process dan alokasi memori yang digunakan untuk mengakses informasi-informasi yang tersimpan di database. Konfigurasi instance dan database ini dibedakan menjadi dua cara yang pertama, satu instance mengakses satu database, dan yang kedua beberapa instance yang berjalan pada server yang berbeda mengakses satu database yang sama. Konfigurasi kedua ini dinamakan Oracle Parallel Server atau yang sekarang lebih dikenal dengan nama Oracle Real Application Cluster. Seorang user tidak dapat mengakses secara langsung sebuah informasi dalam database tanpa melalui instance, jika instance sedang drop, semua informasi pada database tidak dapat diakses melalui system operasi. (Syamsiar, 2004, 1)

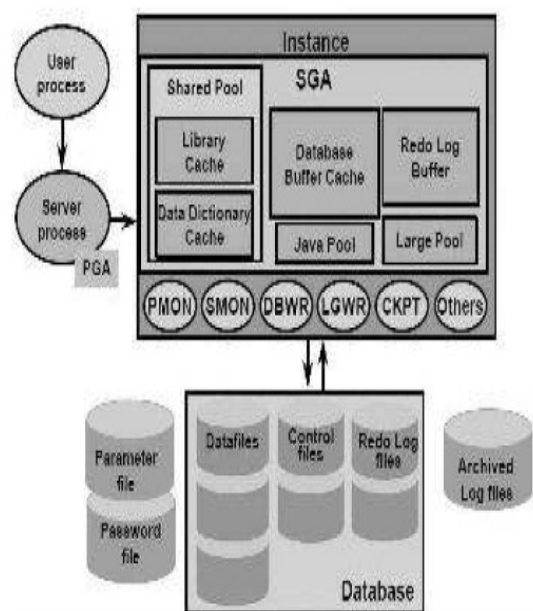
*B. Perancangan Basis Data*

Perancangan basis data adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem. (Falani, 2010, 4)

1. Tujuan Perancangan Basis Data

Menurut (Falani, 2010, 4) tujuan perancangan basis data adalah:

- a) untuk memenuhi informasi yang berisikan kebutuhan-kebutuhan user secara khusus dan aplikasi-aplikasinya.
- b) memudahkan pengertian struktur informasi.
- c) Mendukung kebutuhan-kebutuhan pemrosesan dan beberapa obyek penampilan ( response time, processing time, dan storage space )



Gambar 2. Arsitektur Basis Data Oracle

## 2. Proses Perancangan Basis Data

6 Fase proses perancangan database :

- a) Pengumpulan data dan analisis
- b) Perancangan database secara konseptual  
Pemilihan DBMS
- c) Perancangan database secara logika (data model mapping)
- d) Perancangan database secara fisik
- e) Implementasi Sistem database.

6 fase di atas tidak harus diproses berurutan. Pada beberapa hal, rancangan tersebut dapat dimodifikasi dari yang pertama dan sementara itu mengerjakan fase yang terakhir (feedback loop antara fase) dan feedback loop dalam fase sering terjadi selama proses perancangan. (Falani, 2010, 5)

## 3. Alasan Perancangan Basis Data

- a) Sistem basis data telah menjadi bagian dalam sistem informasi suatu organisasi
- b) Kebutuhan menyimpan data dalam jumlah besar semakin mendesak
- c) Fungsi-fungsi dalam organisasi semakin dikomputerisasikan
- d) Semakin kompleks data & aplikasi yang digunakan, maka relationship antar data harus dimodelisasikan
- e) Dibutuhkannya kemandirian data  
(Falani, 2010, 16)

## 4. Model Entitas-relasional

Model entitas-relasional atau lebih dikenal dengan model E-R adalah sebuah alat untuk mengonstruksi model data konseptual, yang mencerminkan struktur dan batasan dari basis data. Namun untuk mendapatkan suatu basis data yang benar-benar efisien bentuk konseptual seperti ini kadang perlu melalui proses pemetaan ataupun normalisasi. (Nugroho, 2011, 53)

## 5. Entitas dan Atribut

Entitas adalah sebuah objek pada kenyataan yang terpisahkan dari objek yang lainnya, suatu entitas yang memiliki sifat yang sama disebut kumpulan entitas. Pada pemodelan entitas ini dikenal sebagai tabel. Entitas di deskripsikan dengan atribut-atributnya. Atribut merupakan properti yang membedakan suatu entitas dari entitas lainnya. Suatu kumpulan entitas bisa dikatakan terdiri dari entitas yang memiliki tipe atribut yang serupa. Kunci kandidat merupakan suatu atribut yang dapat digunakan sebagai pembeda suatu entitas dengan entitas lainnya. Primary key adalah satu atribut yang bukan hanya mengidentifikasi secara unik suatu kejadian spesifik, tetapi juga dapat mewakili setiap kejadian dari suatu entitas. Foreign key adalah satu atribut atau satu set minimal atribut yang melengkapi satu hubungan yang menunjukkan ke induknya. Foreign key ditempatkan pada entity anak dan sama dengan primary key induk yang direlasikan. Hubungan antara entitas induk dengan anak adalah hubungan satu ke banyak (one to many relationship). (Ramakrishnan & Gehrke, 2007, 27)

## 6. Relasi

Relasi adalah hubungan suatu kumpulan entitas dengan kumpulan entitas lainnya. Misalnya, entitas mahasiswa memiliki hubungan tertentu dengan entitas matakuliah (mahasiswa mengambil matakuliah). (Nugroho, 2011, 56)

## 7. ERD (*Entity Relational Diagram*)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. (Falani, 2010, 40)

## 8. Relasi pada ERD

- a) Satu ke satu (One to one)  
Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.
- b) Satu ke banyak (One to many)  
Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.
- c) Banyak ke banyak (Many to many)  
Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

## 9. Perancangan Basis Data Secara Logika

Perancangan basis data secara logika merupakan proses penerjemahan perancangan basis data secara konseptual ke perancangan basis data secara logika yang dapat langsung diimplementasikan ke sistem basis data. (Nugroho, 2011, 158)

## 10. Teknik Normalisasi

Proses normalisasi adalah proses pengelompokan elemen data menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Pada proses normalisasi dilakukan pengujian pada beberapa kondisi apakah ada kesulitan pada saat menambah/menyisipkan, menghapus, mengubah dan mengakses pada suatu basis data. Bila terdapat kesulitan pada pengujian tersebut maka perlu dipecahkan relasi pada beberapa tabel lagi atau dengan kata lain perancangan basis data belum optimal. (Falani, 2010, 20)

## 11. Kebergantungan Fungsional dan Dekomposisi

Kebergantungan fungsional memainkan peran kunci dalam menghasilkan perancangan basis data yang baik. Suatu atribut B dikatakan bergantung secara fungsional pada atribut A jika dan hanya jika untuk setiap nilai A tertentu akan didapat nilai B yang sama. Misalnya atribut nama\_mahasiswa bergantung secara fungsional terhadap atribut NIM atau atribut nilai bergantung secara fungsional terhadap atribut NIM. (Nugroho, 2011, 192)

Dekomposisi merupakan proses untuk menurunkan suatu tabel dengan banyak atribut menjadi beberapa tabel dengan atribut yang lebih sedikit agar

didapat struktur tabel yang lebih baik. (Nugroho, 2011, 196)

12. Tahap Normalisasi

- a) Bentuk Normal Pertama  
Bentuk normal pertama (1NF) adalah kondisi dimana suatu tabel memiliki atribut yang atomik dimana semua nilai atributnya tidak dapat dibagi lagi menjadi nilai-nilai yang lebih kecil misalnya atribut alamat tidak dapat dikatakan atomik karena atribut alamat dapat dibagi lagi menjadi nilai-nilai jalan, kota, kode pos. (Nugroho, 2011, 190)
- b) Bentuk Normal Kedua  
Bentuk normal kedua (2NF) adalah kondisi dimana suatu tabel telah berada pada bentuk normal pertama dan semua atribut bukan kunci primer bergantung sepenuhnya pada kunci primer. (Nugroho, 2011, 200)
- c) Bentuk Normal Ketiga  
Bentuk normal ketiga (3NF) adalah kondisi dimana suatu tabel telah berada pada bentuk normal kedua dan tidak lagi ditemukannya sebuah kebergantungan transitif dimana suatu atribut bukan kunci juga bergantung pada atribut bukan kunci lainnya. (Nugroho, 2011, 201)
- d) Bentuk Boyce-Codd  
Bentuk Boyce-Codd (BCNF) adalah kondisi dimana suatu tabel telah berada pada bentuk normal ketiga dan jika atribut kunci merupakan satu-satunya kunci kandidat atau kunci calon. (Nugroho, 2011, 205)

III. METODOLOGI PENELITIAN

A. Tempat dan Waktu Penelitian

Dalam pelaksanaan tugas akhir ini penulis mengambil tempat penelitian pada rumah penulis dan perusahaan distribusi yakni PT. Arasmas Anugerah yang menjadi tempat studi kasus.

B. Bahan dan Peralatan

Dalam mengerjakan tugas akhir ini mulai dari mendesain sampai tahap pemrograman penulis menggunakan perlengkapan komputer sebagai media untuk menjalankan program. Secara lebih spesifik perlengkapan komputer beserta pendukung yang digunakan yaitu:

- 1. Spesifikasi Komputer
  - a) Prosesor Intel Core i5-2430M 2.40Ghz
  - b) Memory RAM 4GB DDR3
  - c) Harddisk 750GB
  - d) Kartu Grafis GPU Nvidia GT540M CUDA, 2 GB
- 2. Adapun Aplikasi - aplikasi yang di pakai penulis dalam penyelesaian Tugas Akhir adalah:
  - a) Oracle XE 11.2
  - b) PL/SQL
  - c) Microsoft Visual Studio 2008 C++
  - d) Microsoft Windows 7 OS

C. Prosedur Penelitian

Prosedur yang digunakan dalam perancangan basis data ini adalah:

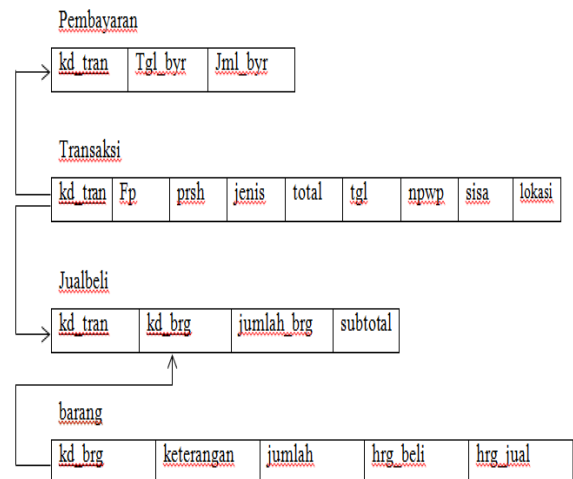
- 1. Mencari literatur mengenai basis data dan perancangan sistem informasi.
- 2. Melakukan studi kasus dengan mengunjungi sebuah perusahaan distribusi (CV. Aras Mas)
- 3. Mempersiapkan perangkat lunak pendukung, seperti Oracle, Microsoft Visual C++.
- 4. Merancang basis data konseptual dan logika menggunakan metode normalisasi dan ERD.
- 5. Merancang dan mendesain antarmuka aplikasi yang akan digunakan dan diintegrasikan dengan basis data yang sudah dibuat.
- 6. Menguji kesesuaian input, proses, dan output dari sistem informasi yang dibuat.

D. Perancangan Basis Data

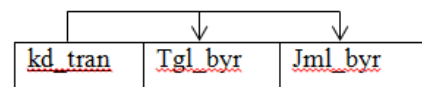
Hasil pemetaan dari model ER proses bisnis PT. Arasmas Anugerah (Gambar 3) akan dinormalisasi.

1. Bentuk Normal Pertama

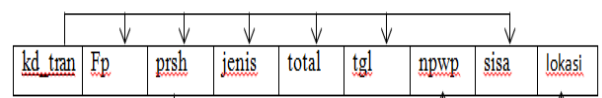
Bentuk normal pertama terpenuhi ketika semua ranah nilai atribut pada suatu tabel bersifat atomik atau tidak dapat dibagi lagi menjadi nilai-nilai yang lebih kecil.



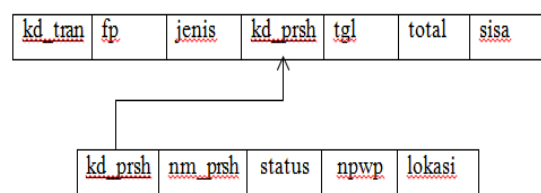
Gambar 3 Pemetaan Model ER Sebelum Normalisasi



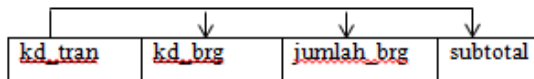
Gambar 4. Kebergantungan Fungsional Tabel Pembayaran



Gambar 5. Kebergantungan Fungsional Tabel Transaksi



Gambar 6. Hasil Dekomposisi Tabel Transaksi



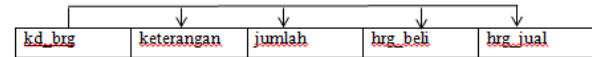
Gambar 7. Tabel Jualbeli

- a) Tabel pembayaran  
Tabel pembayaran sudah tergolong dalam bentuk normal pertama oleh karena ketiga atributnya sudah bersifat atomik.
- b) Tabel transaksi  
Tabel transaksi sudah tergolong dalam bentuk normal pertama oleh karena kedelapan atributnya sudah bersifat atomik.
- c) Tabel jualbeli  
Tabel jualbeli sudah tergolong dalam bentuk normal pertama oleh karena keempat atributnya sudah bersifat atomik.
- d) Tabel barang  
Tabel barang sudah tergolong dalam bentuk normal pertama oleh karena kelima atributnya sudah bersifat atomik.

## 2. Bentuk Normal Kedua

Bentuk normal kedua (2NF) adalah kondisi dimana suatu tabel telah berada pada bentuk normal pertama dan semua atribut bukan kunci bergantung sepenuhnya pada kunci

- a) Tabel pembayaran  
Tabel pembayaran sudah tergolong dalam bentuk 2NF, dapat dilihat dari gambar 4. Semua atribut bukan kunci yaitu tgl\_byr dan jml\_byr bergantung sepenuhnya kepada atribut kunci yakni kd\_tran.
- b) Tabel transaksi  
Tabel transaksi belum termasuk dalam bentuk 2NF, dapat dilihat dari Gambar. 5. Terdapat atribut npwp yang bergantung secara semu terhadap kd\_tran sebagai atribut kunci primer, apabila ada sebuah perusahaan yang baru akan bekerja sama maka data npwp dan lokasi perusahaan itu tidak dapat ditambahkan sampai terjadi suatu transaksi. Kelemahan lainnya adalah perulangan atribut npwp untuk setiap transaksi dari suatu perusahaan dimana kita ketahui npwp suatu perusahaan tidak berubah-ubah. Untuk mengkonversi tabel tersebut menjadi bentuk 2NF adalah dengan mendekomposisi tabel tersebut menjadi dua tabel seperti Gambar. 6. Tabel transaksi didekomposisi menjadi 2 tabel yakni transaksi dan perusahaan. Dengan dekomposisi ini maka tidak ada lagi ketergantungan semu dari atribut bukan kunci kepada kunci primer. Pada tabel transaksi atribut fp, jenis, kd\_prsh, tgl, total, dan sisa bergantung penuh pada atribut kunci primer kd\_trn, sedangkan nm\_prsh, status, npwp, lokasi bergantung penuh pada kd\_prsh sebagai kunci primer pada tabel barang. Apabila ada perusahaan baru yang bekerja sama maka npwp, lokasi serta data perusahaan yang lainnya ditambahkan ke tabel perusahaan.
- c) Tabel jualbeli



Gambar 8. Tabel Barang

Tabel jualbeli sudah tergolong dalam bentuk 2NF, dapat dilihat dari gambar 7.

Ketiga atribut bukan kunci yakni kd\_brg, jumlah\_brg, subtotal bergantung penuh pada atribut kunci kd\_tran.

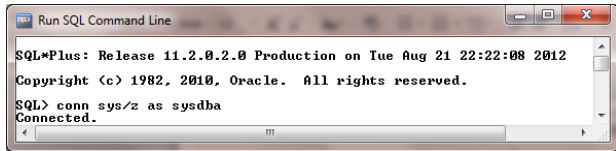
- d) Tabel barang  
Tabel barang sudah tergolong dalam bentuk 2NF, dapat dilihat dari gambar 8. Atribut-atribut bukan kunci seperti keterangan, jumlah, hrg\_beli, hrg\_jual bergantung sepenuhnya terhadap kd\_brg yang merupakan kunci primer.

## 3. Bentuk Normal Ketiga

Bentuk normal ketiga (3NF) adalah kondisi dimana suatu tabel telah berada pada bentuk normal kedua dan tidak lagi ditemukannya sebuah kebergantungan transitif dimana suatu atribut bukan kunci juga bergantung pada atribut bukan kunci lainnya.

- a) Tabel Pembayaran  
Pada tabel pembayaran (gambar 4) kebergantungan yang terlihat tidak terdapat adanya kebergantungan transitif. Tabel pembayaran ini sudah bisa digolongkan telah berada pada bentuk 3NF.
- b) Tabel Transaksi  
Pada tabel transaksi hasil dekomposisi (Gambar. 6) kebergantungan transitif sudah tidak ditemukan. Tidak adanya atribut bukan kunci yang bergantung pada atribut bukan kunci. Dengan demikian tabel ini sudah berada pada bentuk 3NF.
- c) Tabel Perusahaan  
Pada tabel perusahaan hasil dekomposisi (Gambar. 6) tidak ada atribut bukan kunci yang bergantung pada atribut bukan kunci. Dengan demikian tabel ini sudah berada pada bentuk 3NF. Perubahan dari data perusahaan dengan kd\_prsh tertentu tidak akan menimbulkan anomali pada tabel tersebut, penambahan perusahaan baru sebagai penjual atau pembeli juga dapat dilakukan tanpa menimbulkan anomali.
- d) Tabel Jualbeli  
Pada tabel jualbeli (Gambar. 7) tidak ada atribut bukan kunci yang bergantung pada atribut bukan kunci lainnya kd\_brg, dengan demikian tabel jualbeli sudah dalam bentuk 3NF.
- e) Tabel Barang  
Pada tabel barang (Gambar. 8) keempat atribut bukan kunci bergantung hanya pada atribut primernya. Perubahan harga barang atau keterangan tidak akan menimbulkan anomali, demikian dengan penambahan atau penghapusan barang tertentu tidak akan membuat anomali di tabel tersebut.





Gambar 10 Login Admin

4. Bentuk Normal Boyce Codd

Bentuk Boyce-Codd (BCNF) adalah kondisi dimana suatu tabel telah berada pada bentuk normal ketiga dan jika atribut kunci merupakan satu-satunya kunci kandidat atau kunci calon.

a) Tabel Pembayaran

Pada tabel pembayaran (Gambar. 4) terdapat hanya 1 kunci calon atau kunci yang dapat digunakan sebagai pembeda dengan entitas lain yakni atribut kd\_tran. Oleh karena tabel pembayaran sudah dalam bentuk 3NF syarat untuk memenuhi bentuk BCNF sudah terpenuhi.

b) Tabel Transaksi

Pada tabel transaksi pembayaran (Gambar. 5) beberapa atribut memang dapat menjadi kunci kandidat seperti faktur pajak(fp), atau kombinasi atribut kd\_prsh dan tgl, namun dalam proses bisnis yang sebenarnya setiap nota transaksi diberi kode yang unik untuk membedakan setiap transaksi. Dengan demikian kd\_tran sebagai kunci kandidat tunggal dipilih menjadi kunci primer. Oleh karena tabel transaksi sudah dalam bentuk 3NF syarat untuk memenuhi bentuk BCNF sudah terpenuhi.

c) Tabel Perusahaan

Pada tabel perusahaan (Gambar. 6) satu-satunya atribut yang mungkin menjadi kunci kandidat untuk membedakan satu entitas dengan entitas lain yakni kd\_prsh. Oleh karena tabel perusahaan sudah dalam bentuk 3NF maka syarat untuk memenuhi bentuk BCNF sudah terpenuhi.

d) Tabel Jualbeli

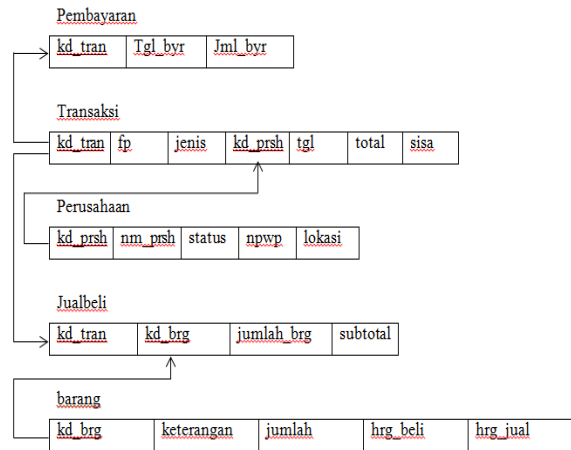
Pada tabel jualbeli (Gambar. 7) atribut kunci kandidat yang dapat ditemui hanyalah kd\_trn. Tabel ini sudah tergolong dalam bentuk BCNF.

e) Tabel Barang

Pada tabel barang (Gambar. 8) kd\_brg menjadi satu-satunya kunci kandidat dan dipilih menjadi kunci primer. Oleh karena tabel ini sudah dalam bentuk 3NF maka tabel ini sudah tergolong dalam bentuk BCNF.

5. Hasil Normalisasi

Setelah melalui proses pemetaan dan normalisasi dari bentuk konseptual, berikut adalah bentuk hasil normalisasi dari perancangan basis data secara konseptual dari PT. Arasmas Anugerah (Gambar. 9).



Gambar 9. Hasil Pemetaan Setelah Normalisasi

IV. HASIL dan PEMBAHASAN

A. Pembuatan Basis Data CV. Arasmas

Setelah melakukan perancangan basis data dan antarmuka, penulis akan mengimplementasikan rancangan tersebut menggunakan perangkat lunak yakni Oracle XE 11.2 dan Microsoft Visual C++ 2008 express edition.

Rancangan basis data diimplementasikan dengan Oracle XE 11.2 dan perancangan antarmuka menggunakan Microsoft visual c++ 2008 express edition.

1. Membuat Tablespace dan User

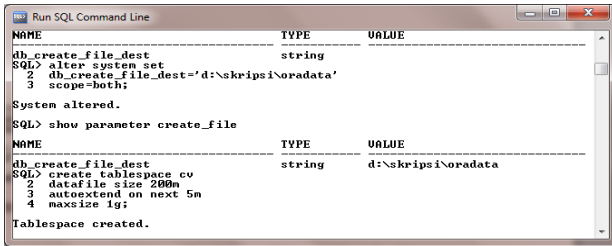
Pembuatan basis data diawali dengan membuat user khusus yang nanti digunakan dalam implementasi rancangan basis data.

Akun admin basis data pada Oracle XE 11.2 ini secara default adalah sys dengan password z. Menggunakan perintah conn untuk login menggunakan sql command line (Gambar 10).

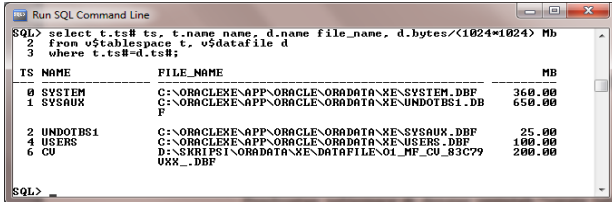
Pembuatan tablespace di dengan perintah “create tablespace” dengan ukuran file data sebesar 200 Mb dengan dengan kemampuan autoextent dengan ukuran maksimum 1 Gb (Gambar 11). Tablespace merupakan lokasi data secara logika sedangkan datafile merupakan bentuk fisik dari sebuah data. Tablespace terdiri dari satu atau lebih datafile. Berikut pada Gambar 12 adalah informasi datafile dari tablespace “cv” yang berhasil dibuat.

Ukuran dari datafile sama seperti ukuran yang dispesifikasikan ketika membuat tablespace “cv” yakni 200 Mb.

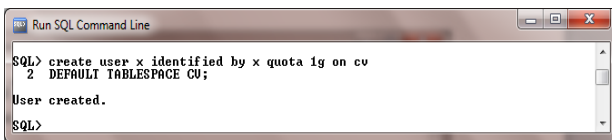
Pembuatan user “x” dengan sandi “x”, banyak kuota 1 Gb pada tablespace cv menggunakan perintah “create user” (Gambar. 13). User ini nantinya yang akan digunakan untuk melakukan pengolahan dari basis data.



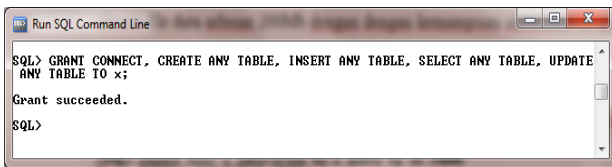
Gambar 11. Membuat Tablespace



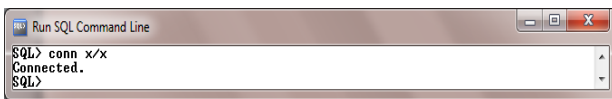
Gambar 12. Informasi Tablespace



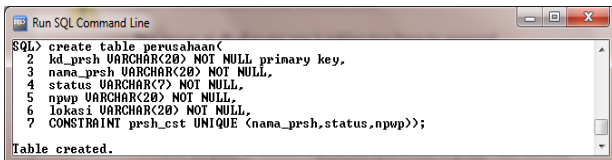
Gambar 13. Membuat User



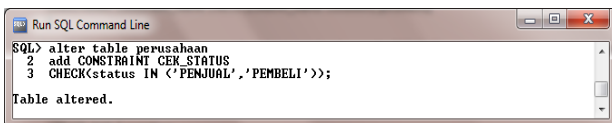
Gambar 14. Membuat Privilege User



Gambar 15 Login Menggunakan User "x"

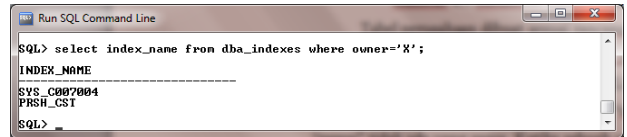


Gambar 16. Membuat Tabel Perusahaan

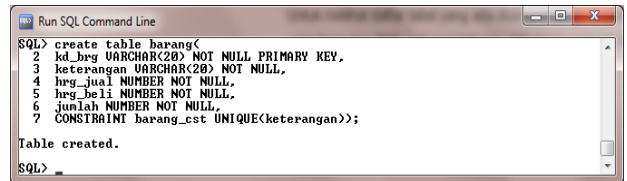


Gambar 17. Membuat Constraint Check Pada Tabel Perusahaan

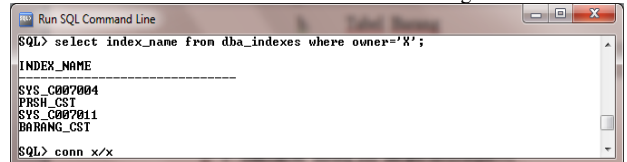
Membuat privilege kepada user "x" dengan perintah "grant". Privilege yang diberikan pada user "x" adalah "CONNECT" yaitu untuk dapat melakukan koneksi ke Oracle, "CREATE ANY TABLE" untuk dapat membuat dan menghapus tabel, "SELECT ANY TABLE" untuk dapat menyeleksi suatu tabel, dan "UPDATE ANY TABLE" untuk dapat memperbaharui suatu tabel (Gambar. 14). Setelah user "x" diberi privilege "CONNECT" maka user sudah bisa digunakan untuk login (Gambar. 15).



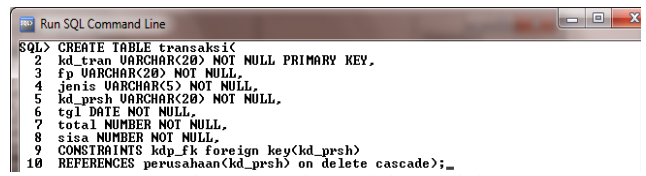
Gambar 18. Indeks Tabel Perusahaan



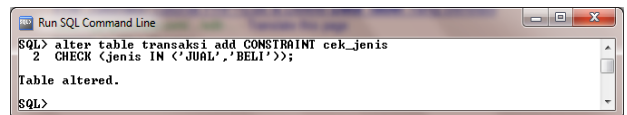
Gambar 19. Membuat Tabel Barang



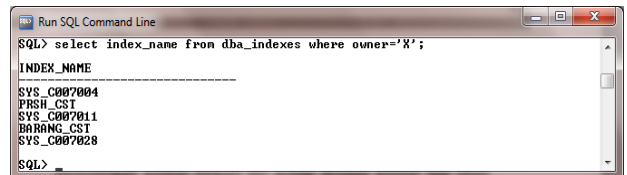
Gambar 20. Indeks Tabel Barang



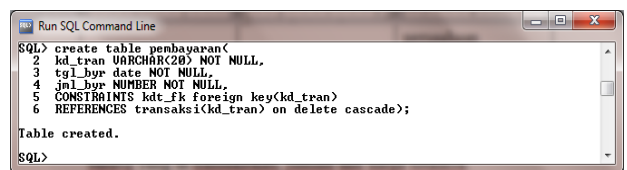
Gambar 21. Membuat Tabel Transaksi



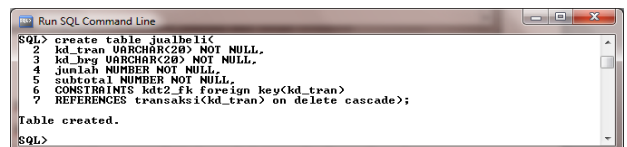
Gambar 22. Membuat Constraint Check Pada Tabel Transaksi



Gambar 23. Indeks Tabel Transaksi



Gambar 24. Membuat Tabel Pembayaran



Gambar 25. Membuat Tabel Jualbeli

## 2. Membuat Tabel

Tabel perusahaan dibuat sesuai rancangan logika basis data dimana terdapat atribut "kd\_prsh" sebagai primary key dan atribut lainnya yang bersifat not null (Gambar. 16). Constraint "prsh\_cst" dibuat untuk memastikan kombinasi atribut "nama\_prsh", "status" dan "npwp" tidak ada yang sama (Gambar. 17). Ketika sebuah primary key dan constraint dibuat maka index untuk constraint tersebut akan dibuat (Gambar. 18).



Constraint “cek\_status” digunakan untuk membatasi input ke kolom status yaitu hanya “pembeli” atau “penjual”.

Tabel barang memiliki 5 atribut yang bersifat not null, sedangkan yang berfungsi sebagai primary key adalah atribut “kd\_brg” (Gambar 19). Constraint “barang\_cst” memastikan tidak ada barang dengan keterangan yang sama. Semua tipe data atribut disesuaikan dengan perancangan logika dari basis data.

Berikut Gambar. 20 adalah indeks yang dibuat ketika tabel barang berhasil dibuat.

Tabel transaksi terdiri dari 7 atribut dimana “kd\_tran” sebagai primary key, pada tabel ini terdapat foreign key yakni kd\_prsh dengan tabel induk “perusahaan”. Constraint “kdp\_fk” memastikan atribut foreign key sesuai dengan atribut dan tabel yang direferensikan dengan klausa “on delete cascade” yang berarti jika data pada induk dihapus maka hal yang sama terjadi pada data di tabel ini (Gambar 21). Constraint “cek\_jenis” berfungsi membatasi input pada kolom jenis yaitu hanya “JUAL” atau “BELI” (Gambar 22). Gambar. 23 menunjukkan indeks dari tabel transaksi.

Tabel pembayaran terdiri dari 3 atribut yang bersifat not null, pada tabel ini terdapat foreign key yakni kd\_tran dengan tabel induk “transaksi”. Constraint “kdt\_fk” memastikan atribut foreign key sesuai dengan atribut dan tabel yang direferensikan dengan klausa “on delete cascade” yang berarti jika data pada induk yakni pada tabel transaksi dihapus maka hal yang sama terjadi pada data di tabel ini (Gambar 24).

Pada tabel jualbeli atribut “kd\_tran” menjadi foreign key tabel induknya adalah tabel transaksi dengan klausa “on delete cascade”. Semua atribut dari tabel ini bersifat not null (Gambar 25).

```
Run SQL Command Line
SQL> create table userdata (
2  username VARCHAR(15) NOT NULL PRIMARY KEY,
3  pass VARCHAR(20) NOT NULL);
Table created.
SQL>
```

Gambar 26. Membuat Tabel Userdata

```
Run SQL Command Line
SQL> create table log(
2  log VARCHAR(50));
Table created.
SQL>
```

Gambar 27. Membuat Tabel Log

```
Run SQL Command Line
SQL> conn x/x
Connected.
SQL> select * from tab;
TNAME          TABTYPE  CLUSTERID
-----
BARANG         TABLE
BIN$JMykBg71qgCBH553LUtgzg==$0 TABLE
BIN$MDY?1xfRqsuRv8nhhhq==$0 TABLE
BIN$PI1Kqy1BTQ2v0j1jpcZju==$0 TABLE
BIN$PLQx7u3BShkLg/40HJwBu==$0 TABLE
BIN$ZotscDj4THG6c6iXTCIS9Q==$0 TABLE
JUALBELI      TABLE
LOG           TABLE
PEMBAYARAN   TABLE
PERUSAHAAN   TABLE
TRANSAKSI    TABLE

TNAME          TABTYPE  CLUSTERID
-----
USERDATA      TABLE

12 rows selected.
SQL>
```

Gambar 28. Semua Tabel Yang Berhasil Dibuat

3. Membuat Trigger

4. Tabel “userdata” akan digunakan untuk menyimpan jenis-jenis pengguna yang akan menggunakan sistem informasi yang dibuat (Gambar 26).

5. Tabel “log” digunakan untuk menyimpan data login dari pengguna yang login, seperti waktu login dan berhasil tidaknya suatu pengguna untuk login ke sistem (Gambar 27).

6. Seluruh tabel yang dibuat pada user “x” dapat dilihat dengan perintah “select \* from tab;” setelah login dengan user “x” (Gambar 28).

Trigger adalah prosedur tersimpan pada Oracle yang secara otomatis dijalankan apabila data di dalam tabel berubah karena eksekusi perintah SQL (INSERT, UPDATE atau DELETE). Untuk dapat membuat trigger suatu user harus terlebih dahulu diberi privilege “CREATE TRIGGER” (Gambar 29).

Gambar. 30 menunjukkan pembuatan trigger “total\_trg” dimana total biaya suatu transaksi akan dihitung berdasarkan akumulasi subtotal dari setiap penjualan atau pembelian dari masing-masing kode transaksi. Trigger ini akan aktif ketika terdapat setiap input ke tabel “jualbeli” setelah itu akan dilakukan update ke tabel transaksi.

Gambar. 32 menunjukkan pembuatan trigger “upd\_barang” dimana trigger ini berfungsi menambah atau mengurangi jumlah stok barang bergantung banyaknya penjualan atau pembelian. Trigger ini akan aktif ketika terdapat input ke tabel jualbeli dan setelah itu akan dilakukan update ke tabel barang.

Gambar. 31 menunjukkan pembuatan trigger “upd\_sisa” dimana trigger ini aktif ketika terdapat proses insert ke tabel pembayaran. Trigger ini akan melakukan update ke kolom “sisa” yang ada pada tabel transaksi. Setiap terjadi pembayaran maka “sisa” akan dikurangkan dengan “jml\_byr” dan akan didapat nilai atribut “sisa” yang baru.

```
Run SQL Command Line
SQL*Plus: Release 11.2.0.2.0 Production on Fri Aug 24 00:28:43 2012
Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> conn sys/z as sysdba
Connected.
SQL> GRANT CREATE TRIGGER TO x;
Grant succeeded.
SQL>
```

Gambar 29 Memberi Privilege Create Trigger Kepada User “x”

```
Run SQL Command Line
SQL> create or replace trigger total_trg
2  after
3  insert
4  ON jualbeli
5  for each row
6  begin
7  update transaksi
8  set total=total+new.subtotal,
9  sisa=sisa+new.subtotal
10 where kd_tran=new.kd_tran;
11 end;
12 /
Trigger created.
```

Gambar 30. Membuat Trigger Menghitung Total Suatu Transaksi

```

Run SQL Command Line
SQL> create or replace trigger upd_sisa
2 after
3 insert on pembayaran
4 for each row
5 begin
6 update transaksi
7 set sisa=sisa:-new.jnl_byr
8 where kd_tran=:new.kd_tran;
9 end;
10 /
Trigger created.
    
```

Gambar 31. Membuat Trigger Update Sisa Pembayaran

```

Run SQL Command Line
SQL> create or replace trigger upd_barang
2 after
3 insert on jualbeli
4 for each row
5 declare
6 status VARCHAR(20);
7 begin
8 SELECT jenis INTO status from transaksi where kd_tran=:new.kd_tran;
9 if status='JUAL' then
10 update barang
11 set jumlah=jumlah:-new.jumlah
12 where kd_brg=:new.kd_brg;
13 end if;
14 if status='BELI' then
15 update barang
16 set jumlah=jumlah+:new.jumlah
17 where kd_brg=:new.kd_brg;
18 end if;
19 end;
20 /
Trigger created.
    
```

Gambar 32. Membuat Trigger Update Stok Barang

KODE BARANG	KETERANGAN	HARGA BELI	HARGA JUAL	JUMLAH
SEMENTA	SEMENTA	Rp 35.000,00	Rp 50.000,00	10
SEMEN 3	SEMEN PADANG	Rp 50.000,00	Rp 85.000,00	340
SUPERMAB	SUPERM AYAM B.	Rp 10.000,00	Rp 15.000,00	0
SEDAFKARI	MI SEDAP KARI	Rp 12.000,00	Rp 15.000,00	0
PASIR 1	PASIR GUNUNG K.	Rp 20.000,00	Rp 50.000,00	63

Gambar 33. Form Barang

Barang	Jumlah	Total DPP
SEMEN 3	20	1700000

Total transaksi DPP: Rp 1.700.000,00 PPN: Rp 170.000,00

Gambar 34. Form Penjualan

KODE	PERUSAHAAN	TOTAL + PPN	SISA
0002	PT INDOFEED	3.520.000,00	3.520.000,00
0003	CV ARUNDINA	4.675.000,00	4.675.000,00
0001	PT. PADANG JAYA	10.670.000,00	10.670.000,00

Gambar 35. Form Pembayaran Hutang

**B. Implementasi Pada Aplikasi**

Basis data PT. Arasmas Anugerah yang telah dibuat pada DBMS Oracle diimplementasikan dalam aplikasi.

Gambar. 33, Gambar. 34, Gambar. 35, dan Gambar. 36 merupakan hasil implemmentasi rancangan basis data ke dalam aplikasi.

**V KESIMPULAN**

Dari perancangan dan implementasi yang dilakukan pada DBMS dan aplikasi yang dibangun, penulis dapat menarik beberapa kesimpulan yaitu:

1. Pemahaman terhadap proses bisnis dari suatu sistem yang akan dibuat sangat memudahkan dalam perancangan basis data secara konseptual.
2. Rancangan basis data secara konseptual memerlukan proses normalisasi agar dapat dicapai bentuk basis data yang lebih efisien.
3. Perancangan basis data dengan menggunakan metode normalisasi memiliki keuntungan sebagai berikut:
4. Tidak ditemukannya anomali penghapusan saat implementasi, anomali modifikasi baris, dan anomali penyisipan Redudansi data pada tabel-tabel sangat bisa ditekankan

Kode	Nama Perusahaan	NPWP	Lokasi
TOND ASADI	ABADI JAYA	14234234	MANADO
TOND BERKAT	BERKAT PINELENG	42343242	MANADO
TOND JINGGA	JINGGA WARNA	1233144	MANADO
TOND JOYFUL	JOYFUL	1245618	TALAUD
TOND SAYA	SAYA	12421453	TALAUD

Gambar 36. Form Mitra Perusahaan

5. Tabel yang dihasilkan memiliki field yang lebih sedikit sehingga mempermudah proses implementasi kueri yang ingin dibuat dan tidak membingungkan dalam tahap pemrograman.
6. Penggunaan constraints pada oracle memudahkan pengujian keabsahan data yang masuk ke basis data, hal ini menyebabkan aplikasi yang dibuat sudah tidak perlu lagi menyediakan logika untuk menguji keabsahan data yang akan disisipkan.
7. Penggunaan trigger pada oracle meminimalisasi penulisan kode program apabila terjadi penyisipan data pada beberapa tabel yang saling terkait.

## DAFTAR PUSTAKA

- [1] C. Amri. (2003). Tutorial SQL. Jakarta. [Online]. Available: <http://www.ilmukomputer.com>
- [2] L. Ashdown and T. Kyte, “*Oracle Database Concept*”, Oracle, Redwood City, CA, 2011.
- [3] A. Basoufi, “*Normalisasi Database*”, Pens-ITS, Surabaya, 2007.
- [4] R. Elmasri and N. Shamkant B, “*Fundamentals of Database Systems*”, Addison-Wesley, Reading, 2000.
- [5] A. Falani, “*Perancangan Basis Data*”, Fakultas Ilmu Komputer Universitas Narotama, Surabaya, 2010.
- [6] S. Fogel, “*Oracle Database Administrator’s Guide*”, Oracle, Redwood City, CA, 2006.
- [7] R. Greenwald and Stackowiak and J. Stem, “*Oracle Essential 4th Edition*”, O’Reilly Media, USA, 2007.
- [8] L. Long and N. Long, “*Computers Information Technology in Perspective 10th edition*”, Pearson Education, inc, Upper saddle River, New Jersey, 2002.
- [9] A. Nugroho, “*Perancangan dan Implementasi Sistem Basis Data*”, Penerbit Andi, Yogyakarta, 2011.
- [10] S. Rahimi and F. Haug, “*Distributed Database Management System*”, John Wiley & Sons, Inc. New Jersey, 2011..
- [11] R. Ramakrishnan and J. Gehrke, “*Database Management System*”. McGraw Hill Higher Education, USA, 2007.
- [12] Rohmad. (2008). Buku Pegangan Database Oracle Administrator. Jakarta. [Online]. Available: <http://www.rohmad.net>
- [12] O. Zaiane, “*CMPT Lecture Notes*”, Simon Fraser University, Canada, 1995.