

STRATEGI IMPLEMENTASI PERANCANGAN SISTIM INFORMASI MENGUNAKAN METODE ENTITAS DAN MNEMONIC

Yunus Fadhillah S, S.Kom., M.Kom¹⁾

¹⁾Staf Pengajar Teknik Informatika

Institut Bisnis dan Informatika Indonesia Kwik Kian Gie

Jl.Yos Sudarso Kav.87 Sunter, Jakarta Utara

yunus.fadhillah@kwikkiangie.ac.id

ABSTRAK

Perancangan sistim informasi selalu melalui tahapan dan siklus sebuah software, perencanaan yang matang memudahkan saat implementasi dan tahapan perawatan sebuah software di kemudian hari. Penggunaan strategi yang tepat dalam perancangan dan implementasi akan memudahkan tahapan tersebut dan selanjutnya. Konsep entitas dan Mnemonic merupakan metode umum yang sangat mudah diterapkan dalam pengembangan dan implementasi sebuah sistim informasi.

Kata Kunci: *Entity, Entity Relationship , Database, Database Design , Information System, Software Engineering, Software Development, Software Development Strategy*

1.PENDAHULUAN

1.1.Latar Belakang Masalah

Perancangan sistim informasi saat ini dalam rekayasa dan pengembangan perangkat lunak merupakan tahapan yang sangat vital untuk kelangsungan dan penggunaan sebuah aplikasi dalam skala besar terutama dalam skala *enterprise*. Banyak masalah yang sering muncul pada saat dan pasca tahap implementasi atau *go live* ketika akan dilakukan pemeliharaan, rekayasa dan pengembangan perangkat lunak dalam memenuhi kebutuhan organisasi. Masalah yang sering dihadapi para pengembang pasca tahap implementasi atau di tahap pemeliharaan adalah :

- a. Dokumentasi kompleks
- b. Koreksi terhadap bugs, upgrade dan penambahan menjadi tidak mudah
- c. Perawatan dan pengembangan lanjutan menjadi lebih sulit
- d. Tingkat kompleksitas sistim informasi dan kerugian penggunaan RDBMS menjadi tinggi

Hal tersebut disebabkan pada tahap perancangan terjadi ketidakselarasan antara *database, programming modules dan user interface* dalam sistim informasi. Rancangan awal yang terintegrasi, efektif dan matang sangat diperlukan untuk kelanjutan dalam pemeliharaan, rekayasa dan pengembangan sistim informasi selanjutnya.

Perancangan database pada aplikasi skala *enterprise* menimbulkan masalah baru dalam sisi pemrograman dan perawatan data karena adanya kekurangan dalam konsep RDBMS, yaitu rumit, ukuran yang besar, biaya database, tambahan hardware dan konversi, performance serta resiko kegagalan sistim sehingga menimbulkan kesulitan dalam perawatan dan pengembangan aplikasi ke depan.

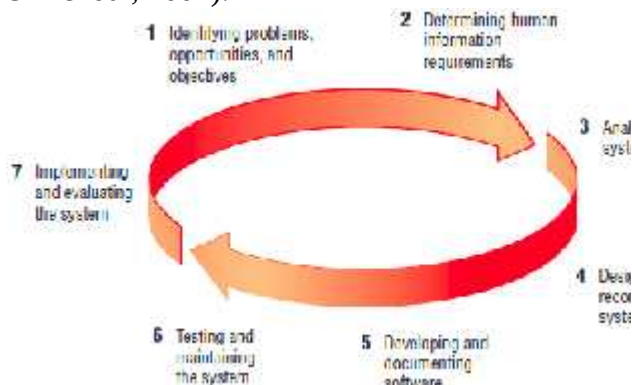
Entity Relationship Model diperkenalkan oleh Codd untuk manajemen database dan dibuat bentuk logikanya oleh Chen pada tahun 1976. Model ini digunakan untuk menggambarkan secara abstrak bentuk database dengan notasi yang sederhana dan mudah dimengerti. Dengan menggunakan

model diatas, maka kita dapat membuat strategi dalam merancang database serta aplikasi agar memudahkan perawatan dan pengembangan aplikasi tersebut di masa yang akan datang.

2. TINJAUAN PUSTAKA

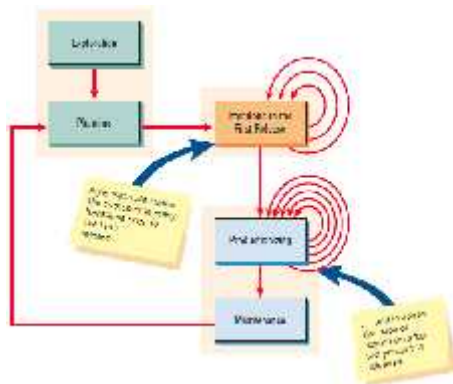
2.1. Metode Analisa Dan Perancangan Sistim Informasi

Dalam analisa dan perancangan sistim informasi, saat ini pada umumnya menggunakan beberapa pendekatan, (Kendal & Kendal, 2007).



Gambar 2.1. The System Development Life Cycle

Pendekatan Siklus Pengembangan Sistim (The System Development Life Cycle), pendekatan yang dilakukan secara sistematik dalam menganalisa dan merancang sistim informasi.



Gambar 2.2. The Agile Approach

Pendekatan Tangkas (The Agile Approach), pendekatan ini cenderung berfokus pada *The System Development Life Cycle* (SDLC) dan paling banyak digunakan dan diterapkan dilapangan. Pendekatan ini adalah pendekatan pengembangan perangkat lunak berbasis pada nilai-nilai, prinsip, dan praktis. Keempat nilai adalah komunikasi, kesederhanaan, umpan balik, dan keberanian.



Gambar 2.3. Object-Oriented System Analysis And Design

Analisa dan Perancangan Berorientasi Objek (Object-Oriented System Analysis and Design), pendekatan ini dimaksudkan untuk memfasilitasi pembuatan sistim secara cepat berubah dalam merespon lingkungan bisnis yang dinamis.

Semua pendekatan analisa dan perancangan sistim informasi diatas bila dibandingkan dalam setiap prosesnya maka akan terlihat dalam tabel dibawah ini. Secara umum dapat penulis simpulkan untuk semua pendekatan analisa dan perancangan sistim informasi mempunyai proses atau tahapan identifikasi, analisa, perancangan, pembuatan dan pemeliharaan.

The Systems Development Life Cycle		The Analysis Phase		Design/Development Phase	
Step	Process	Step	Process	Step	Process
1	Identify problem, requirements, and objectives	1	Explore	1	Problem definition Phase
2	Determine system information requirements				
3	Analyze system requirements			2	System Requirements Phase
4	Design system architecture and design details	2	Plan	3	System Design Phase
5	Develop and code the system	3	Implement		
6	Test and maintain the system	4	Monitor		
7	Implement and maintain the system	5	Review		

Gambar 2.4. Tabel Perbandingan 3 Pendekatan Analisa dan Perancangan Sistem Informasi

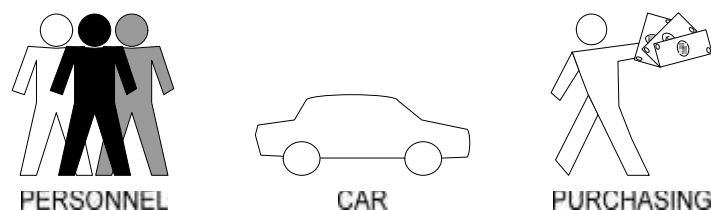
2.2. Teori Entity Relationship Model

Pertama kali diformulasi dan diperkenalkan oleh Codd (Codd:1969) untuk manajemen database dan Entity Relations Diagrams (Peter,Chen:1976) digunakan untuk menggambarkan database berdasarkan persepsi dari dunia nyata yang mengandung himpunan dari objek-objek yang disebut entitas dan hubungan antara objek-objek tersebut. Setiap entitas bersifat unik berdasarkan dari atribut yang dimilikinya.

Tujuan utama ERD ini adalah untuk mewakili objek data dan hubungan data dengan beberapa file yang berhubungan.

2.2.1. Entitas

Entitas adalah objek yang keberadaannya dapat dibedakan dari objek yang lainnya. Entitas ini dapat bersifat nyata dan abstrak, seperti orang, benda, tempat kejadian dan konsep. Setiap Entitas memiliki karakteristik khusus yang disebut Attribute.

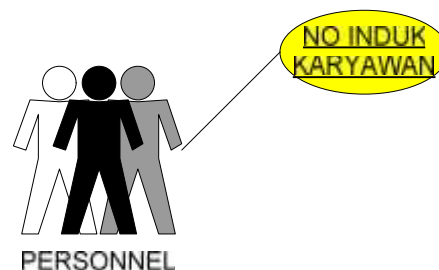


Gambar 2.5. Objek sebagai Entitas

2.2.2. Atribut

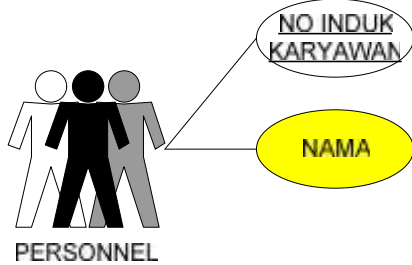
Atribut adalah karakteristik dari entitas yang merupakan penjelasan lebih rinci tentang entitas tersebut. Nilai atribut merupakan data atau informasi aktual yang disimpan. Atribut mempunyai 4 jenis, yaitu:

- Key, untuk menentukan suatu entity secara unik



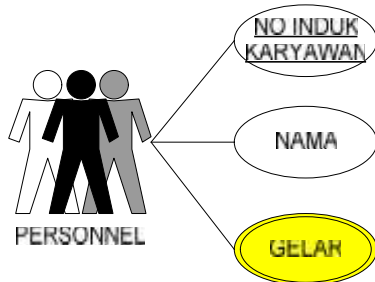
Gambar 2.6. Entitas Dengan Atribut Key

- Simple, atribut yang bernilai tunggal



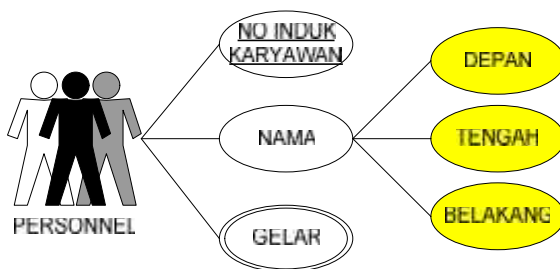
Gambar 2.7. Entitas Dengan Atribut Simple

- Multivalue, atribut yang memiliki sekelompok nilai untuk setiap entity



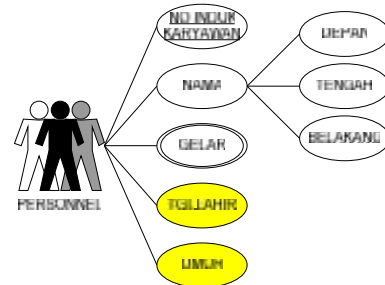
Gambar 2.8. Entitas Dengan Atribut Multivalue

- Composite, atribut yang lebih kecil dan mempunyai arti tertentu



Gambar 2.9. Entitas Dengan Atribut Composite

- Derivatif, atribut yang dihasilkan dari atribut lainnya

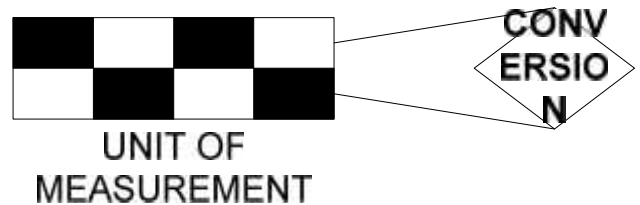


Gambar 2.10. Entitas Dengan Atribut Derivatif

2.2.3. Relationship

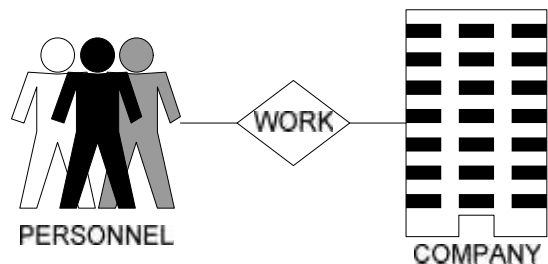
Relationship adalah hubungan yang terjadi diantara satu atau lebih entity dan memiliki derajat relationship. Terdapat 3 jenis derajat dari relationship, yaitu:

- Derajat Satu (Unary Degree)



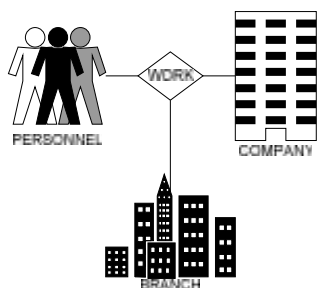
Gambar 2.11. Relasi Derajat Satu

- Derajat Dua (Binary Degree)



Gambar 2.12. Relasi Derajat Dua

- Derajat Tiga (Ternary Degree)



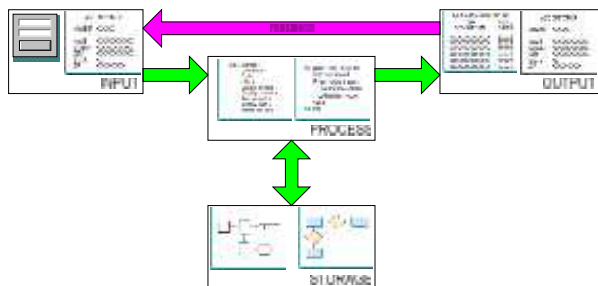
Gambar 2.13. Relasi Derajat Tiga

Saat ini hampir semua sistem informasi memiliki database dengan konsep RDBMS (*Relational Database Management System*), semua pendekatan analisa dan

2.3. Mnemonic
Mnemonic berasal dari kata Yunani mnemonikos yang berarti ingatan atau yang berhubungan dengan ingatan. Dalam perangkat lunak computer, mnemonic merupakan teknik pembelajaran yang membantu kita dalam mengingat sebuah informasi (Soanes, Stevenson, Hawker:

3. METODE IMPLEMENTASI KONSEP ENTITAS DAN MNEMONIC DALAM SISTEM INFORMASI

Dalam konsep sistem informasi, sebuah sistem mempunyai 3 bagian yaitu input, proses dan output. Konsep ini harus ada dan dilakukan pada tahap perencanaan sistem informasi agar ke 3 bagian tersebut dapat terhubung dan terintegrasi dengan baik.



Gambar 3.1. Konsep Sistem Informasi

Sistem informasi memerlukan data sebagai masukan untuk diproses, disimpan,

perancangan sistem informasi merancang database umumnya pada tahapan perancangan. ER merupakan alat bantu dalam merancang database di level logika dan merupakan bagian yang hampir tidak dapat dipisahkan dalam tahapan perancangan sistem informasi.

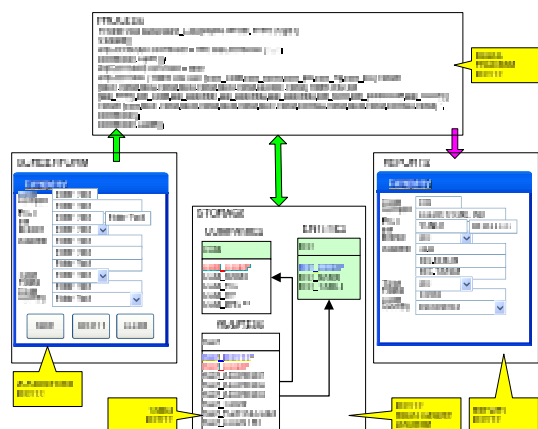
Secara umum untuk perancangan database sudah cukup baik dan memuaskan dalam penggunaan Relasi sehingga hanya diperlukan sedikit modifikasi agar entitas benar-benar jelas diketahui secara visual.

2006) dalam bentuk nama, kata, gambar, istilah, kode, potongan kata atau kombinasinya.

Metode mnemonic digunakan pada semua bidang dan industri, terutama dalam dunia komputer hampir semua identifikasi sebuah entitas menggunakan metode ini.

digunakan dan ditampilkan dari penyimpanan data. Database merupakan hasil dari *Entity Relationship Diagram* yang berasal dari *Data Flow Diagram*.

Metode Entity Relationship diterapkan dengan menggunakan pengkodean mnemonic pada level perancangan ER Diagram dan atributnya. Pada perancangan tampilan, laporan dan modul program dianggap sebagai entitas. Sehingga sebuah sistem informasi merupakan sebuah entitas dan memiliki subsistem entitas yang terdiri dari komponen-komponen entitas yang saling berhubungan. Semua entitas yang telah dikodekan dimasukkan ke dalam Entity Relationship Diagram.



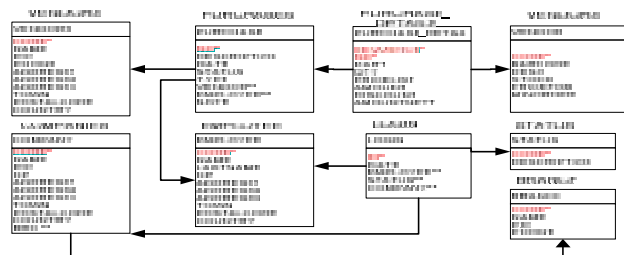
Gambar 3.2. Sistem Informasi Merupakan Sebuah Sistem Entitas

4. STRATEGI IMPLEMENTASI KONSEP ENTITAS DALAM SISTEM INFORMASI

Strategi ini diimplementasikan pada tahapan perancangan sistem informasi setelah bisnis proses selesai digambarkan dalam Data Flow Diagram dimulai dengan perancangan *Entity Relationship Diagram*, *User Interface* dan program atau modul modul program.

4.1. Bentuk Relasi Database Telah Memiliki Bentuk 2nf Atau Lebih

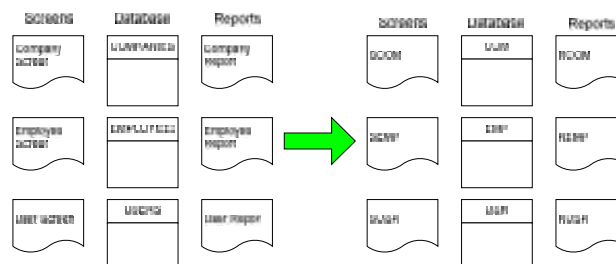
Syarat utama dalam penerapan strategi entitas adalah Relasi telah terbentuk dalam bentuk normal ke 2 atau 2NF.



Gambar 4.1. Relasi dalam bentuk 2NF atau lebih

4.2. Pengkodean Nama Entitas Dengan Teknik Mnemonic

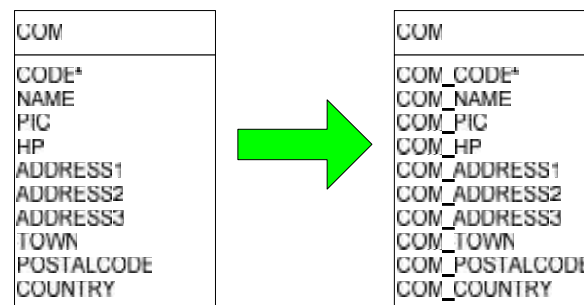
Untuk mempermudah dan mengingat suatu entitas serta penulisannya tidak panjang maka nama entitas dibuat dengan teknik mnemonic. Nama entitas akan berubah menjadi sebuah kode entitas.



Gambar 4.2. Penyederhanaan nama entitas dengan teknik mnemonic

4.3. Turunkan Kode Entitas Kepada Atributnya

Kode entitas yang telah terbentuk harus diturunkan terhadap atribut entitas itu sendiri agar bisa diketahui dan mudah dilacak atribut tersebut milik entitas siapa.

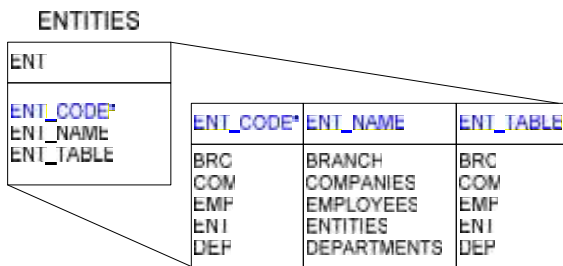


Gambar 4.3. Penurunan kode entitas terhadap atributnya

4.4. Daftarkan Kode Entitas Ke Dalam Relasi Sebagai Bentuk Entitas Baru

Bila semua entitas telah memiliki kode entitas, maka daftarkan kode entitas kedalam relasi dengan membentuk entitas baru. Entitas ini berisi informasi semua kode, nama dan entitas yang sudah ada. Daftar kode entitas ini diperlukan untuk pencatatan, perubahan dan pengembangan

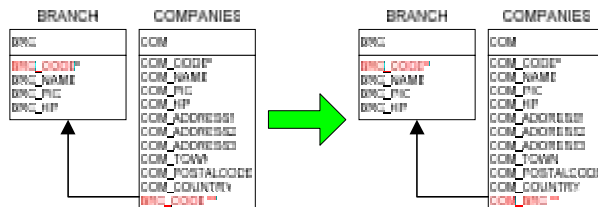
entitas di tahapan pemeliharaan dan pengembangan.



Gambar 4.4. Daftar entitas

4.5. Attribut Foreign Key Mengikuti Entitas Relasi Dimana Foreign Key Berada Dan Referensinya

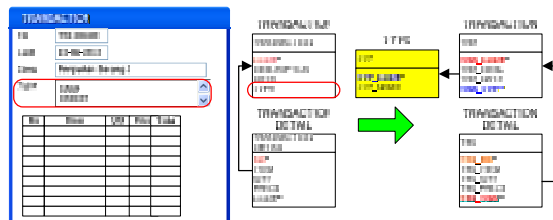
Attribut *foreign key* tetap menggunakan kode entitas dimana *foreign key* berada dan diikuti oleh kode entitas referensinya.



Gambar 4.5. Attribut *Foreign Key* mengikuti kode entitas dimana *Foreign Key* berada

4.6. Jika Ada Attribut Multivalued Dibuat Sebagai Entitas Sendiri, Terutama Attribut Jenis Dan Status

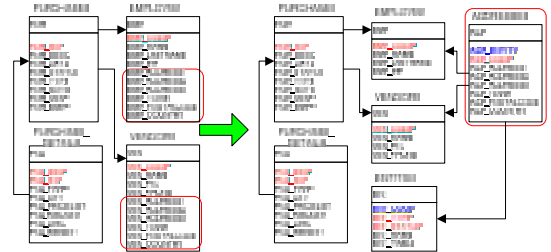
Attribut multivalued seperti jenis atau status sebuah entitas dibuat sebagai atribut sendiri.



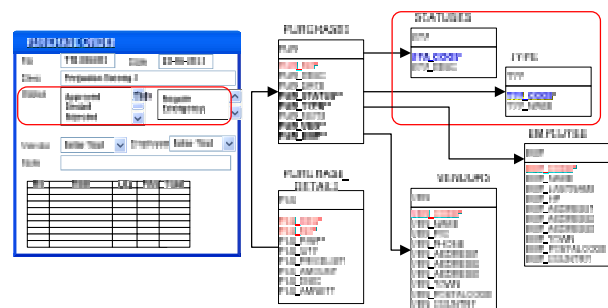
Gambar 4.6. Attribut Multivalue dibuat sebagai entitas sendiri

4.7. Jika Ada Attribute Composite Atau Attribut Yang Memiliki Kelas Yang Sama Dalam Entitas Lain, Dapat Dibuat Sebagai Entitas Baru Atau Tersendiri

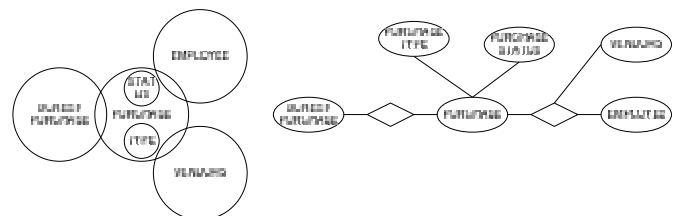
Jika sebuah attribute merupakan attribute composite atau attribute dapat merupakan anggota dalam entitas tertentu, maka hilangkan attribute tersebut dan dapat membentuk entitas baru.



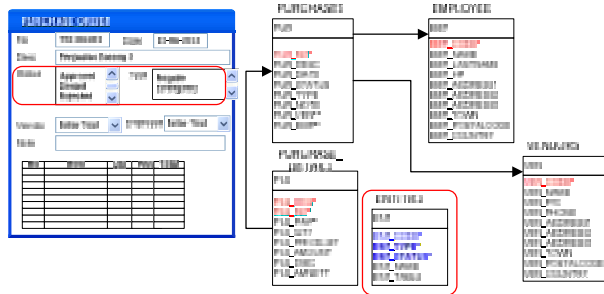
Gambar 4.7. Penggabungan Attribute composite atau kelas yang sama sebagai entitas baru



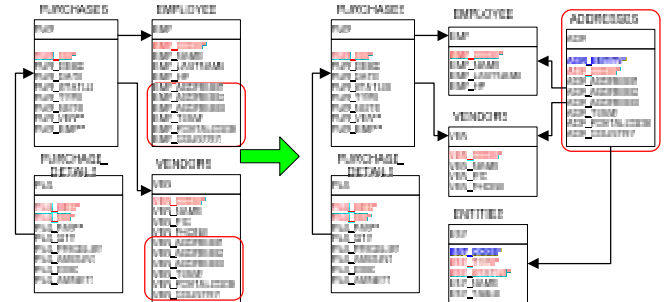
Gambar 4.8. Entitas Form dan Entitas Database



Gambar 4.9. Himpunan Anggota dan Relasi Antar Entitas



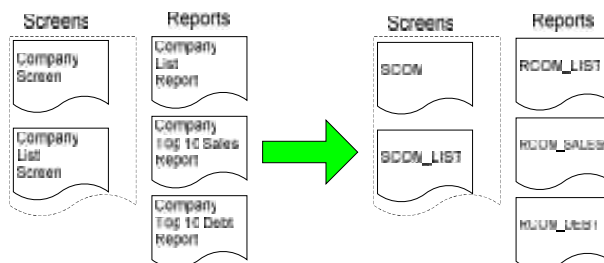
Gambar 4.10. Hasil Masukan Entitas Sebagai Atribut



Gambar 4.12. Pemisahan Atribut Berdasarkan Kelas atau Kelompok Entitas

4.8. Pengkodean Entitas Derivatif Mengacu Pada Entitas Utama Atau Kelasnya

Bila sebuah screen dibagi ke dalam 2 bagian, yaitu bagian header dan detail untuk mempermudah tampilan. Screen header merupakan screen utama, sedangkan screen detail merupakan screen derivatif. Atau report yang terdiri dari banyak format tetapi terkait hanya kepada satu entitas atau subjek.



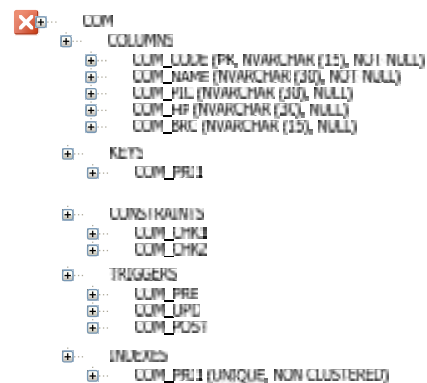
Gambar 4.11. Entitas Derivatif

4.9. Jika Entitas Memiliki Atribut Yang Sama Di Entitas Lainnya, Buat Sebagai Atribut Kelas

Atribut yang sama di beberapa entitas dapat dikelompokkan atau dibuat kelas sebagai entitas baru. Hal ini dapat dilakukan dengan syarat daftar kode entitas sudah terdaftar atau dibuat dalam Entity Relationship Diagram.

4.10. Penamaan Key, Constraints, Trigger Dan Indexes Pada Database Fisik Tetap Mengikuti Kode Entitas Yang Telah Dibuat Pada ERD

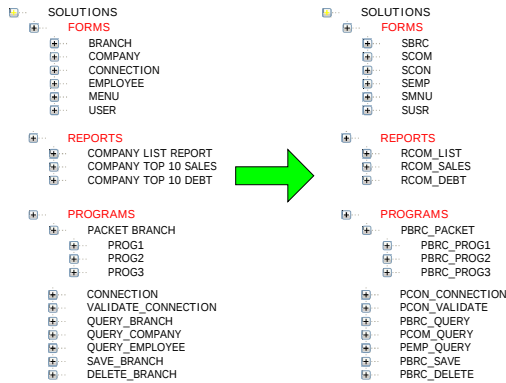
Ketika ERD ditransform menjadi LRS dan database fisik, maka penamaan pembuatan key, constraints, triggers dan indexes tetap mengikuti kode entitas dimana atribut tersebut berada.



Gambar 4.13. Fisik Database

4.11. Pengkodean Entitas Pada Modul Program Aplikasi Sistem Informasi

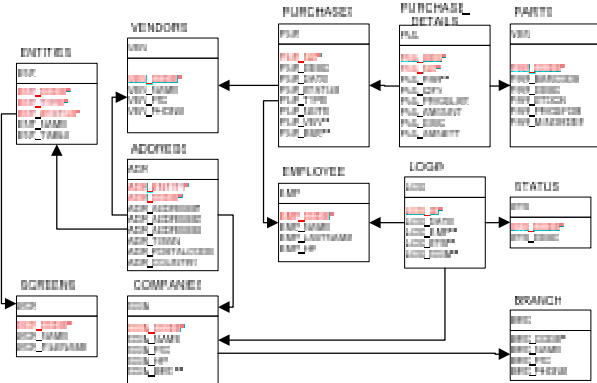
Pengkodean dan penerapan entitas bisa diterapkan dalam modul program dan jenis atau status pada sistem informasi.



Gambar 4.14. Penerapan Dalam Modul Dan Aplikasi Pemograman

5. SIMPULAN DAN SARAN

Penerapan strategi entitas ke dalam ERD akan menghasilkan ERD yang berdasarkan entitas pada contoh dibawah ini.



Gambar 4.15. ERD Akhir Setelah Penerapan Strategi Entitas

5.1. SIMPULAN

Implementasi strategi dengan menggunakan konsep entitas dan mnemonic dapat disimpulkan melalui langkah-langkah dibawah ini:

1. Pengkodean nama entitas dengan mnemonic
2. Penurunan kode entitas kepada atributnya
3. Daftarkan kode entitas kedalam ERD sebagai entitas baru
4. Atribut FK mengikuti entitas ERD dimana FK berada

5. Atribut Multivalue dibuat sebagai entitas sendiri
6. Atribut composite atau atribut yang memiliki kelas yang sama dibuat sebagai entitas sendiri
7. Penamaan key, constraints, trigger dan indexes pada database fisik tetap mengikuti kode entitas yang telah dibuat pada ERD
8. Pengkodean entitas pada *user interface* dan report
9. Pengkodean entitas derivatif tetap mengacu pada entitas awal atau kelasnya
10. Pengkodean entitas pada modul program aplikasi sistim informasi

Dengan melakukan implementasi strategi diatas maka akan memiliki keunggulan:

1. Nama entitas mudah diingat dan atributnya mudah dibedakan
2. Relasi antar entitas lebih jelas dan mudah dilacak
3. Database, User Interface dan modul program lebih terintegrasi
4. Tingkat kerugian penggunaan RDBMS menjadi lebih rendah
5. Dokumentasi menjadi mudah dan tingkat kerumitan lebih rendah
6. Tahapan Perawatan dan pengembangan pada sistim informasi menjadi mudah dan mengurangi biaya

5.2. SARAN

Masih diperlukan penelitian lanjutan untuk mendukung dan membuktikan keunggulan-keunggulan diatas dari sisi sistim analis atau programmernya.

DAFTAR PUSTAKA

- [1]. Soanes, Catherine. Stevenson, Angus. Hawker, Sara. March 29, 2006. Concise Oxford English Dictionary (Computer

- Software), 11th Ed. Oxford University Press
- [2]. Codd, Edgar F. 1969. Derivability, Redundancy and Consistency of Relations stored in Large Data Banks. IBM Research Report
- [3]. Pin Peter, Chen Shan. 1976. The Entity-Relationship Model - Toward a Unified View of Data. ACM Transactions on Database Systems, Vol.1, No.1, March 1976
- [4]. McLeod, Raymond. Management Information System, 7th ed. Prentice Hall, New Jersey, 1998
- [5]. McNurlin, Barbara C. Sparague, Ralph H Jr. Information Systems Management in Practice, 4th ed. Prentice Hall, New Jersey, 1998
- [6]. Kent, William. 1983. A Simple Guide to Five Normal Form in Relational Database Theory, communications of the ACM 26(2), Feb. 1983, 120-125
- [7]. Kendal, Kenneth E. Kendal, Julie E. Systems Analysis and Design, 8th ed. Prentice Hall, 2011