

Model Data Semantic Merancang Konseptual Sistem Basis Data Pada Model Object Relationship Attribute – Semi Structured (ORA-SS)

Naomi Putri Sion Silaban¹, Devi Romauli Br Sitanggang², Ing Madya Kurnia Sandi Sinaga³, Nadia Fathurrahmi Lawita⁴

^{1,2,3,4}Program Studi Akuntansi, Fakultas Ekonomi dan Bisnis, Universitas Muhammadiyah Riau

e-mail: naoumiputrisionsilaban28@gmail.com¹, deviomaulisitanggang@gmail.com², ingmadya9@gmail.com³, nadia.fathurrahmi@umri.ac.id⁴

Abstrak

Pada saat ini banyak sistem komputer yang menghasilkan dan memakai sejumlah data yang besar. Hal ini dipengaruhi oleh tingkat kebutuhan dari pengguna yang menggunakan sistem komputer baik secara individual, maupun korporasi. Sistem komputer yang didalamnya terdapat aplikasi tidak dapat dipisahkan peran sistem basis data sebagai tempat penyimpanan. Pada penelitian ini, pemodelan *Object Relationship Attribute Semi Structured* (ORA-SS) digunakan untuk menangani sistem basis data semi terstruktur dimana sistem tersebut menggunakan sistem file yang direkam (*record*), dimanipulasi serta diakses kembali (*retrieve*). Kondisi seperti ini memacu timbul redundansi dan anomali. Metode penelitian dimulai dari studi pustaka, observasi dan lain-lain kemudian setiap kebutuhan data dianalisa dan dirancang dengan pemodelan ORA-SS sehingga hasil akhir bahwa data yang terindikasi mengulang atau kosong bisa mengurangi redundansi dan anomali. Didalam penelitian ini menunjukkan bahwa, hampir semua atribut yang ada pada entitas kendaraan maupun pelanggan dapat mereferensikan nilai atribut ke entitas lainnya melalui entitas tersarang tanpa harus membuat entitas yang baru, sehingga hal ini dapat mengurangi redundansi bahkan anomali input ketika data akan dimasukkan kedalam tabel.

Kata kunci: objek, atribut, keterhubungan, pemodelan, semiterstruktur

Abstract

At this time many computer systems that generate and use large amounts of data. This is influenced by the level of needs of users who use computer systems either individually or corporately. A computer system in which there are applications cannot be separated from the role of the database system as a storage area. In this study, Object Relationship Attribute Semi Structured (ORA-SS) modeling is used to handle a semi-structured database system where the system uses a file system that is recorded (recorded), manipulated and accessed again (retrieve). Conditions like this trigger redundancies and anomalies. The research method starts from literature study, observation, etc., then every data requirement is analyzed and designed with ORA-SS modeling so that the final result is that data indicated as repeating or empty can reduce redundancies and anomalies. In this study, it shows that, almost all attributes that exist in vehicle and customer entities can reference attribute values to other entities through nested entities without having to create a new entity, so that this can reduce redundancy and even input anomalies when data will be entered into the table.

Keywords : *object, attribute, relationship, modeling, semi-structure*

PENDAHULUAN

Dalam pandangan terhadap basis data, terdapat tiga level abstraksi data, yaitu pandangan para pemakai, pandangan konseptual dan pandangan fisik oleh seorang

perancang basis data. Para pemakai basis data umumnya adalah orang-orang yang awam terhadap konsep dan teknologi yang digunakan dalam basis data. Permasalahannya adalah pada saat merancang, seorang perancang perlu mengetahui kebutuhan data dan informasi yang diinginkan oleh para pemakai. Dalam hal ini, maka seorang perancang basis data/analisis sistem harus selalu berkomunikasi dengan para pemakai basis data. Untuk mengkomunikasikan rancangan basis data dan sistem yang akan dikembangkan tersebut, diperlukan suatu cara yang mudah dipahami secara logika oleh para pemakai basis data.

Para pemakai tidak perlu tahu kompleksitas dan kerumitan dalam teknis penyimpanan data dalam media penyimpanan. Pemakai juga tidak akan memperhatikan bagaimana data disimpan dalam media penyimpanan secara fisik. Untuk kepentingan ini, maka diperlukan apa yang disebut sebagai model data. Model data merupakan suatu cara untuk menjelaskan tentang data-data yang tersimpan dalam basis data dan bagaimana hubungan antar data tersebut untuk para pemakai (*user*) secara logis.

Saat ini, beberapa atau banyak sistem komputer yang menghasilkan dan memakai sejumlah data yang besar. Hal ini dipengaruhi oleh tingkat kebutuhan dari pengguna yang menggunakan sistem komputer baik secara individual, maupun korporasi. Sistem komputer yang didalamnya terdapat aplikasi tidak dapat dipisahkan peran sistem basis data sebagai tempat penyimpanan. Sistem basis data muncul dari kenyataan bahwa seiringnya kebutuhan sejumlah data yang besar dan selama ketika banyak orang yang memerlukan *query* untuk memperoleh informasi (Groppe, 2011).

Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. (Lawita, 2021)

Sistem basis data semi terstruktur hadir di beberapa tempat yang ditandai dengan XML (*Extended Markup Language*) (Ganguly & Sarkar, 2012). Selain itu pula, didalam sistem tersebut data disimpan dalam bentuk teks yang menjadi cikal bakal sistem file (*flat file system*) tidak seperti pada sistem basis data relational, dimana data sudah tersimpan dan tertata dengan baik (terstruktur). Sistem file tersebut menimbulkan beberapa masalah dalam organisasi data atau file bahkan pada sistem basis data. Selanjutnya masalah yang paling signifikan adanya redundansi data, anomali data baik anomali yang disebabkan penyisipan (*insert*), perubahan (*update*), dan penghapusan (*delete*). Keadaan seperti ini terlihat di Adira Finance Tegal sebagai perusahaan kredit kendaraan bermotor yang menangani pembayaran kredit dari setiap pelanggan (*customer*).

Hampir setiap bagian (*department*) terkait memiliki sistem file serta memiliki struktur file yang berbeda dengan bagian (*department*). Meskipun demikian, akibatnya adalah ketika data akan diolah (*generate*) dari berbagai bagian (*department*) terkait muncul anomali data sampai pada data tersebut tidak konsisten ketika terjadi perubahan. Namun rumusan masalahnya adalah bagaimana membuat sistem basis data yang sifat datanya seperti sistem file namun dapat mengurangi ketidakkonsistenan, dan anomali data yang terjadi di sebuah perusahaan kredit kendaraan bermotor dengan lintas data dari berbagai bagian (*department*). Oleh sebab itu, sistem file (*flat file system*) dapat diselesaikan ketika dengan kondisi data yang semi terstruktur dengan pemodelan data yang berorientasi pada

keterhubungan objek dengan objek lain melalui relasi atribut (*Object Relationship Attribute Semi Structured* (ORA-SS)).

Pada penelitian ini, pemodelan Object Relationship Attribute Semi Structured (ORA-SS) digunakan untuk menangani sistem basis data semi terstruktur dimana sistem tersebut menggunakan sistem file yang direkam (*record*), dimanipulasi serta diakses kembali (*retrieve*) (Groppe, 2011).

Pengertian Model Data Semantic

Semantic model merupakan suatu model berdasarkan obyek. Semantic model digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna

secara logik. Semantic model didasarkan pada suatu persepsi bahwa real world terdiri atas obyek- obyek dasar yang mempunyai hubungan/kerelasian antarobyek-obyek dasar tersebut. Semantic model digambarkan dalam bentuk diagram yang disebut diagram semantic (semantic diagram) (Martin, 1975).

Data yang dikembangkan Semantic model hampir sama dengan Entity Relationship Model. Perbedaannya adalah kerelasian antarobyek dasar tidak dinyatakan dengan simbol. tapi secara semantic, yaitu menggunakan kata-kata (semantic). Selain itu, sejauh ini simbol yang digunakan juga belum memiliki ketentuan baku bagaimana dalam ER D (Sutanta,2004).

Komponen Diagram Semantic

Bagaimana dalam ER D. diagram semantic tersusun atas tiga komponen, yaitu entitas, atribut, dan kerelasian antar entitas. Perbedaannya adalah dalam penggunaan simbol, terutama kerelasian antar entitas (Sutanta, 2004).

a. Entitas (*Entity*)

Entitas adalah objek-objek dasar yang terkait di dalam sistem, dapat berupa orang, objek, atau hal yang perlu dilakukan di dalam data. Dalam diagram semantic, entitas dapat digambarkan dengan berikut (Sutanta, 2004):

- Entitas dinyatakan dengan simbol persegi panjang atau elips
- Nama entitas dituliskan di dalam simbol persegi panjang
- Nama entitas terdiri: kata benda, tunggal
- Nama entitas sedapat mungkin menggunakan nama yang disetujui dan dapat diumumkan maknanya dengan jelas

Penggunaan *tanda garis bawah (hyphenunderscore)*, *pemendekan*, *singkatan juga lazim digunakan untuk memberikan nama entitas seba mana dalam ER_D. Gambar 8.1 menunjukkan contoh-contoh entitas data semantic model.*

Objek dasar	symbol entitas (pilihan 1)	atau	symbol entitas (pilihan 2)
Mahasiswa	mahasiswa	atau	Mahasiswa
Dosen	dosen	atau	Dosen
Ruang	ruang	atau	Ruang
Propinsi	propinsi	atau	Propinsi
Agama	agama	atau	Agama

Gambar 1 contoh Entitas

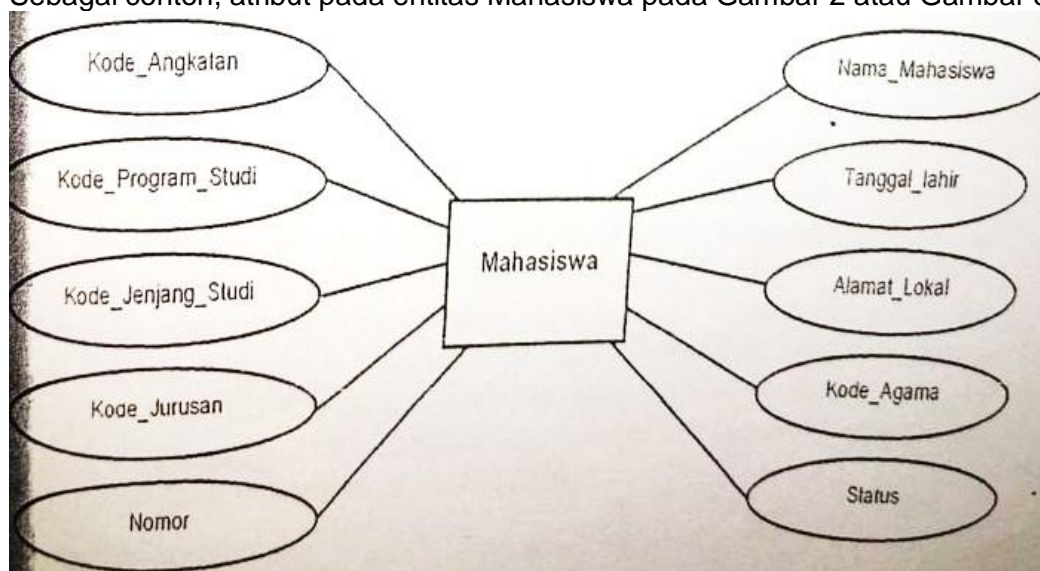
Konsep isian entitas, himpunan entitas, entitas regular, entitas dependen, entitas super type dan entitas sub type yang dikenal dalam model ER, juga berlaku dalam semantic model.

b. Atribut (*Atribut*)

Dalam semantic model, atribut atau properti (properti) merupakan rerangan-keterangan yang terkait pada entitas yang perlu diambil berdasarkan data dasar. (Sutanta, 2004):

- Atribut yang disetujui dengan simbol elips
- Atribut atribut yang dituliskan di dalam simbol elips
- Atribut atribut kata benda, tunggal
- Atribut atribut dapat digunakan dengan nama yang mudah diubah dan dapat menyatakan maknanya dengan jelas
- Atribusi disetujui dengan entitas yang sesuai dengan menggunakan garis (seyogianya menggunakan garis lurus, namun dalam kondisi yang tidak dapat digunakan menggunakan garis lurus)

Konsep atribut sederhana (atribut sederhana) dan atribut campuran (komposit altribut) yang dikenal dalam model ER juga tetap berlaku pada semantik model. Sebagai contoh, atribut pada entitas Mahasiswa pada Gambar 2 atau Gambar 3.



Gambar 2 Contoh atribut pada entitas Mahasiswa (pilihan simbol 1)

c. Kerelasiaan Antar Entitas (*Relationship*)

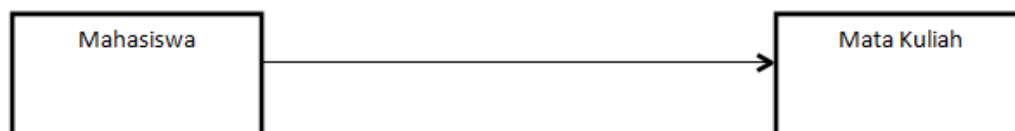
Kerelasiaan antar entitas yang menyatakan kejadian atau transaksi yang terjadi di antara dua buah entitas yang keterangannya perlu disimpan dalam basis data, dalam semantic model dapat digambarkan dengan cara berikut (Sutanta 2004).

1. Kerelasiaan dinyatakan dengan simbol garis dengan sebuah mata panah
2. Nama kerelasiaan dituliskan di samping garis kerelasiaan
3. Kerelasiaan menghubungkan dua entitas
4. Nama kerelasiaan berupa: kata kerja aktif (diawali dengan awalan me-), tunggal
5. Nama kerelasiaan sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

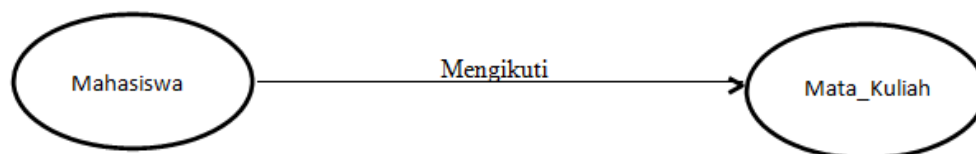
Kerelasiaan antar entitas dalam semantic model juga dikelompokkan dalam tiga jenis, yaitu (Sutanta, 2004):

1. Kerelasiaan jenis 1-ke-1/satu ke satu (one to one)
2. Kerelasiaan jenis n-ke-1/banyak ke satu (many to one) atau 1-ke-n/satu ke banyak (one to many)
3. Kerelasiaan jenis n-ke-n/banyak ke banyak (many to many).

Sebagai contoh, Gambar 4 menunjukkan kerelasiaan antara entitas Mahasiswa dan Mata_Kuliah dalam semantic model untuk pilihan simbol 1 (menggunakan persegi panjang) dan Gambar 5 untuk pilihan simbol 2 (menggunakan elips).



Gambar 4 Contoh kerelasiaan antara entitas mahasiswa dan Mata_Kuliah (pilihan 1)



Gambar 5 Contoh kerelasiaan antara entitas mahasiswa dan Mata_Kuliah (pilihan 2)

Konsep tentang kerelasian rekursif dan kerelasian asosiatif yang dikenal dalam modelER juga dapat digunakan di dalam semantic model (Sutanta, 2004).

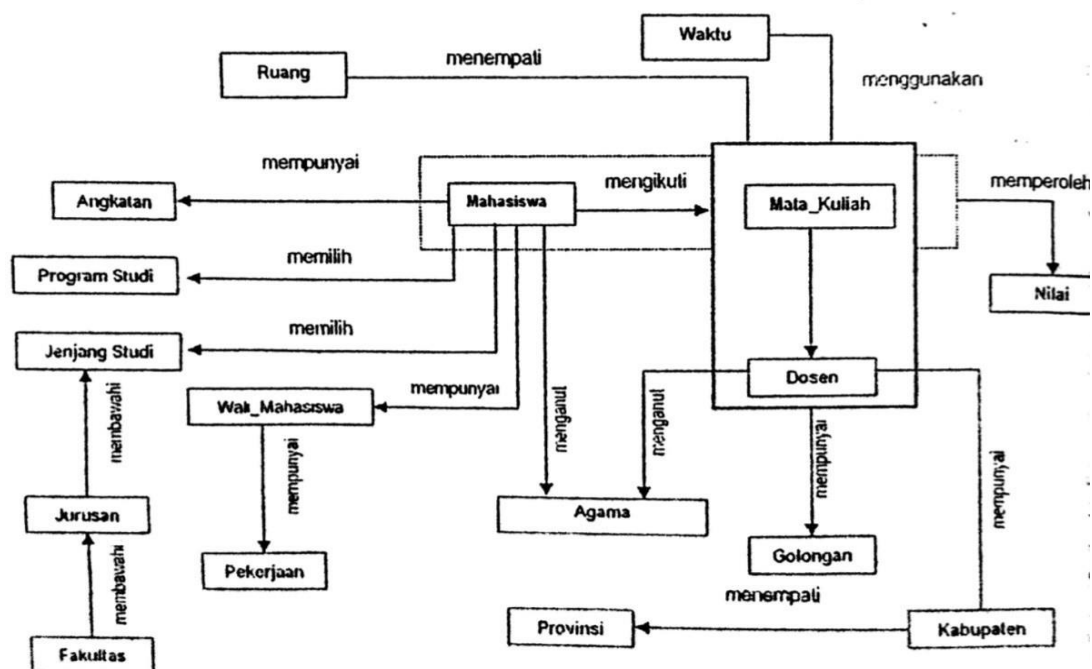
Menggambar Diagram Semantic

Menggambarkan diagram semantic (semantic diagram) secara lengkap dapat dilakukan dengan langkah sebagai berikut (Sutanta, 2004).

1. Identifikasikan setiap entitas yang terlibat
2. Identifikasikan setiap atribut pada setiap entitas
3. Identifikasikan setiap kerelasian berikut jenisnya yang terjadi di antara entitas
4. Gambarkan simbol-simbol entitas, atribut, dan kerelasian antar entitas sedemikian sehingga simbol kerelasian dapat digambarkan dengan jelas/tidak saling bertabrakan.
5. Cek diagram semantic yang terbentuk, dalam hal :
 - a. Kelengkapan entitas
 - b. Kelengkapan atribut
 - c. Kelengkapan kerelasian antar entitas
 - d. Jenis kerelasian antar entitas.

Contoh Diagram Semantic

Dengan tanpa melibatkan komponen atribut pada ssetiap entitas maka Gambar 8.6 menunjukkan contoh diagram zemantic untuk subsistem pengeloa data akademik sebagai contoh ER_D sebelumnya.



Gambar 6 Contoh Diagram Semantic

Kelebihan Dan Kelemahan Diagram Semantic

Jika diterapkan dengan benar atau tepat maka penggunaan digram semantic dalam pemodelan data meberikan keuntungan yang sama dengan Er_D baik bagi perancang maupun pengguna. Kelemahan diagram semantic antara laim (Sutanta,2004)

1. Kebutuhan media yang sangat luas
2. Sering kali diagram semantic tampil sangat ruwet
3. Tidak dapat menunjukkan jenis kerelasian antar entitas

METODE PENELITIAN

Penelitian yang dilakukan akan menggunakan metode penelitian terdapat 3 langkah (*step*) didalam merancang sistem basis data semi terstruktur pada pemodelan Object Relationship Attribute Semistructured (ORA- SS). Pertama, peneliti melakukan kajian tinjauan pustaka dengan mencari dan menemukan sumber referensi yang relevan, valid, dan reliabel. Tinjauan pustaka akan mengambil buku yang terkait dengan penelitian sistem basis data yang membahas sistem basis data semi terstruktur secara khusus pemodelan Object Relationship Attribute Semistructured (ORA-SS). Disamping itu, didalam tinjauan pustaka akan didukung oleh beberapa jurnal yang terkait dalam penelitian yang membahas sistem basis data semi terstruktur.

Kedua, langkah observasi dimana peneliti melakukan kunjungan (*visitasi*), pengamatan (*observation*), dan analisa (*analysis*) ke tempat objek penelitian yang terkait dengan pembahasan hal sistem basis data semi terstruktur.

Ketiga, penelitian perancangan pemodelan ORA-SS dengan mengidentifikasi entitas- entitas yang terlibat dalam sistem file yang akan berubah menjadi sistem basis data semi terstruktur. Kemudian didalam pemodelan ORA-SS akan dilakukan identifikasi kebutuhan data pada sistem basis data termasuk membuat prosedural perancangan dan membangun hubungan antar entitas yang akan menjadi kelas objek (*object class*). Berikut ini adalah gambar atau bagan yang menunjukkan alur metodologi penelitian yang akan membahas mengenai sistem basis data semi terstruktur pada pemodelan Object Relationship Attribute Semistructured (ORA- SS).



Gambar 7 . Metodologi sistem basis data semi terstruktur

HASIL DAN PEMBAHASAN

Identifikasi kebutuhan data

Terkait dengan penelitian ini, identifikasi dilakukan ketika objek atau entitas yang terlibat dalam objek penelitian untuk merancang sistem basis data semi terstruktur pada level konseptual. Adapun entitas yang terlibat adalah pelanggan (*customer*) sebagai entitas utama, bagian keuangan (*accounting*), kendaraan, pembayaran (*payment*), dan penagihan (*billing*).

Entitas (<i>entities</i>)	Karakteristik / Atribut (<i>properties</i>)
Pelanggan	NO_PEL, NAMA, ALAMAT, EMAIL, NO_HP, TIPE, STATUS
Staf Admin	NO_PEG, NAMA, JABATAN,
Pembayaran	KD_BYR, PERIODE, NOMINAL, TGL_BYR, STATUS, PAJAK
Penagihan	ID_INV, PERIODE, NOMINAL, TGL_TMP, STATUS, MET_BYR PAJAK
Kendaraan	NO_KD, NAMA, JENIS, MERK, TAHUN, NO_MESIN, WARNA, PRODUSEN

Tabel 1. Identifikasi Entitas Sistem Basis Data Semi Terstruktur

Prosedural Perancangan

Aturan prosedural utama didalam penelitian melalui Object Relationship Attribute Semistructured (ORA-SS) akan diklasifikasi menjadi dua bagian yang akan ditransformasikan kedalam bentuk relasi objek (*object relational*) sebagai berikut :

1. Setiap kelas objek (*object class*) bersama dengan semua atribut akan membentuk relasi bersarang saat atribut *multi-valued* sebagai kelompok pengulangan pada relasi (*object relation*) tersebut.
2. Setiap tipe relasi (*relationship type*) bersama dengan semua atribut yang melengkapinya membentuk relasi bersarang ketika atribut *multi-valued* sebagai bagian kelompok pengulangan pada sebuah relasi (*object relation*).

Menurut dari prosedural perancangan diatas secara garis besar dibagi menjadi dua bagian, yaitu kelas objek (*object class*) dan tipe relasi (*relation type*). Prosedural pertama berupa objek apabila setiap kelas objek membuat sebuah relasi memiliki beberapa aturan sebagaiberikut (Ganguly & Sarkar, 2012).

Pengenal (*identifier*) dan kunci kandidat pada sebuah kelas objek merupakan kunci utama (*primary key*) dan kunci kandidat dari relasi yang dihasilkan.

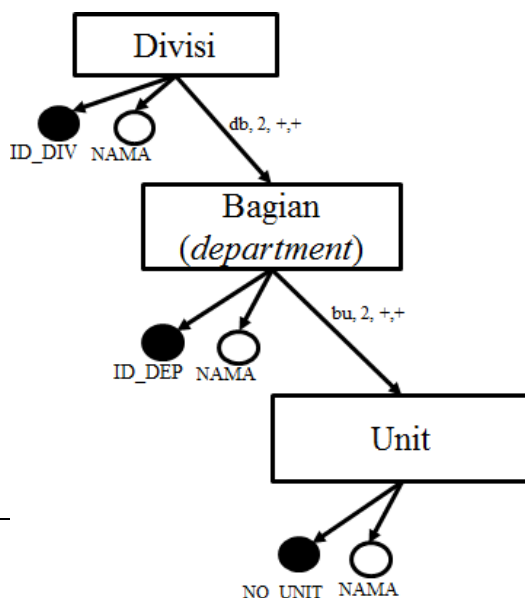
1. Setiap atribut "single-value" pada kelas objek adalah atribut nilai tunggal dari relasi yang telah dihasilkan
2. Setiap atribut "multi-valued" pada kelas objek ini membentuk kelompok berulang pada relasi
3. Setiap referen merupakan kunci tamupada relasi ini
4. Setiap atribut "composite" didalam pemodelan ORA-SS direpresentasikan secara langsung dan diganti oleh komponennya dari relasi yang dihasilkan

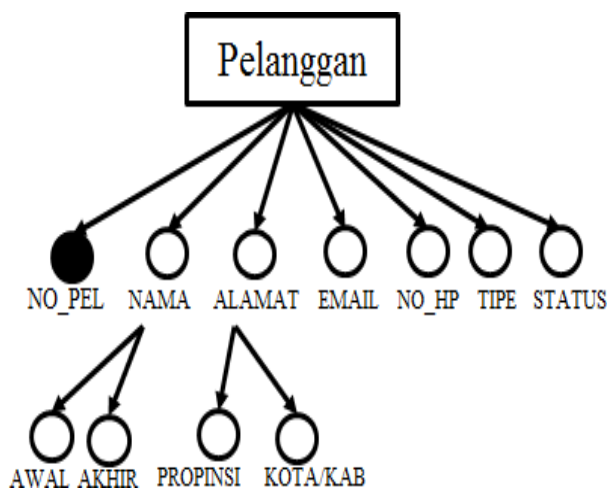
Sedangkan prosedural perancangan dari tipe relasi (*relation type*) membuat sebuah relasi dapat memiliki beberapa aturan sebagai berikut:

1. Pengenal pada semua partisipasi kelas objek pada tipe relasi tersebut membentuk atribut "single-value" dari relasi bersarang, dan kunci tipe relasi dapat ditentukan oleh jumlah partisipan dari tipe relasi.
2. Setiap atribut "single-valued" dari tipe relasi ini adalah atribut "single-valued" dari relasi yang dihasilkan.
3. Setiap atribut "multi-valued" dari tipe relasi ini membentuk kelompok berulang pada relasi tersebut.

Pemodelan basis data dengan ORA-SS

Berdasarkan tabel 1 identifikasi entitas sistem basis bata semi terstruktur, terdapat 5 entitas yang akan menjadi dasar pemodelan basis data semi terstruktur ORA-SS. Salah satu atau semua akan dibuat kelas objek, atribut, dan tipe relasinya. Pemodelan ORA-SS akan terbagi menjadi 2 diantaranya Perancangan basis data dengan ORA-SS schema diagram dan Perancangan basis data dengan ORA-SS instance diagram.





Gambar 8. Skema ORA-SS Divisi dan Pelanggan

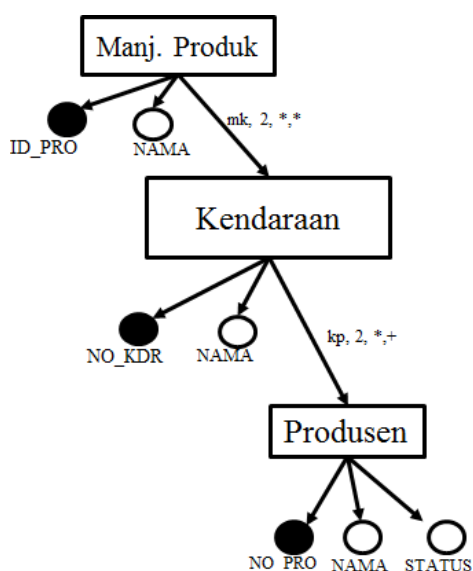
Berdasarkan gambar 8 terdapat diagram yang menjelaskan kelas objek yang terdiri dari divisi, bagian (*department*), dan unit. Ketiga kelas objek tersebut memiliki relasi dengan “n” ary relationship yang dimaksudkan dengan 2 maka menjadibinary relationship. Diagram tersebut memiliki nama relasinya adalah “db” antara kelas objek divisi dengan bagian (*department*) kemudian partisipan “many to many” artinya, banyak divisi memiliki banyak bagian, hal ini begitupula antara bagian (*department*) dengan unit. Selanjutnya, diagram skema ORA-SS kelas objek pelanggan memiliki atribut nama, alamat, dsb. Kelas objek tersebut memiliki kunci atribut yaitu “NO_PEL” yang memiliki peran sebagai pengenal dengan objek lainnya.

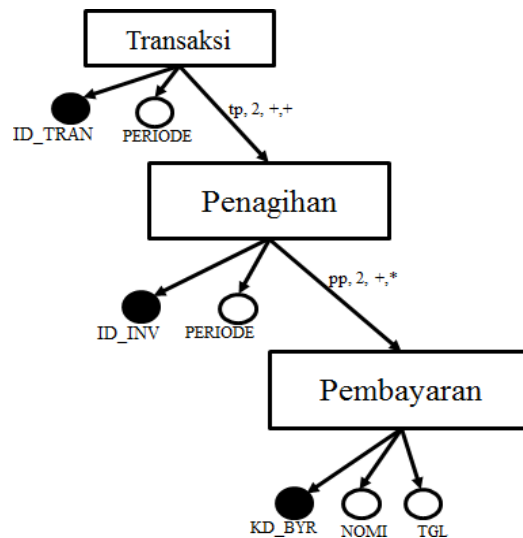
Relasi Objek

Divisi(ID_DIV, NAMA) Bagian(ID_DEP, NAMA) Unit(NO_UNIT, NAMA)
Pelanggan(NO_PEL, NAMA, ALAMAT, EMAIL, NO_HP, TIPE, STATUS)
(AWAL, AKHIR)

Relasi Relationship

db(ID_INV, ID_DEP) bu(ID_DEP, NO_UNIT)





Gambar 9. Diagram Skema ORA-SSManajemen Produk dan Transaksi

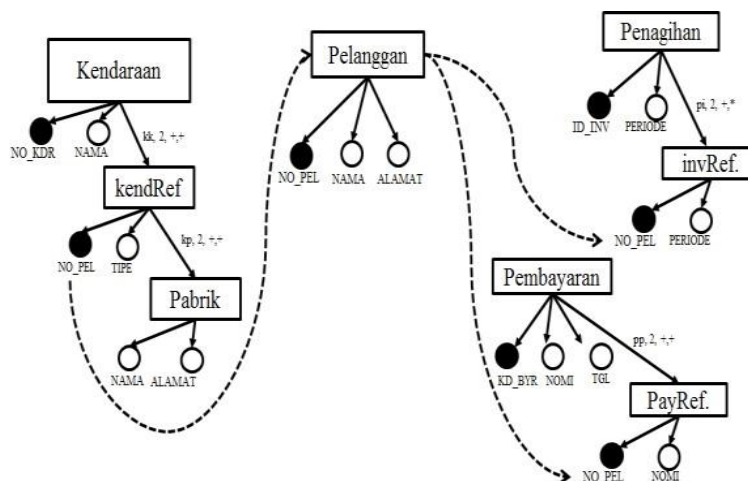
Gambar 9 terdapat 2 diagram yang menjelaskan kelas objek yang terdiri dari manajemen produk, kendaraan, dan produsen. Ketiga kelas objek tersebut memiliki relasi dengan “n” ary relationship yang dimaksudkan dengan 2 maka menjadi binary relationship. Diagram tersebut memiliki nama relasinya adalah “mk” antara kelas objek manj. Produk dengan kendaraan kemudian partisipan “many to many” artinya, banyak produk memiliki banyak kendaraan, hal ini begitupula antara kendaraan dengan produsen. Selanjutnya, diagram skema ORA-SS kelas objek transaksi memiliki atribut ID_TRAN, PERIODE, dsb. Sedangkan kelas objek penagihan memiliki atribut ID_INV, dan PERIODE Kelas objek tersebut memiliki kunci atribut yaitu “ID_INV” yang memiliki peran sebagai pengenal dengan objek lainnya terutama dengan kelas objek pembayaran.

Relasi Objek

Manj.Produk(ID_PRO,NAMA) Kendaraan(NO_KDR,NAMA) Unit(NO_UNIT, NAMA)
Produsen(NO_PRO, NAMA, STATUS)
Transaksi(ID_TRAN, PERIODE) Penagihan(ID_INV, PERIODE) Pembayaran(KD_BYR,
NOMI, TGL)

Relasi Relationship

mk(ID_PRO,NO_KDR) kp(NO_KDR, NO_PRO)tp(ID_PRO, NO_KDR) pp(ID_INV,
KD_BYR)



Gambar 10. Referensi antar kelas objek diagram skema ORA-SS

SIMPULAN

Didalam penelitian ini, peneliti menggunakan pemodelan ORA-SS untuk merancang sistem basis data dengan kondisi sistem file atau data semi terstruktur. ORA-SS dapat mendeskripsikan tingkat *n-ary relationship* atau atribut dari kelas objek bahkan dapat mengidentifikasi apabila ada indikasi atribut merupakan atribut dari tipe relasi atau atribut dari kelas objek. Didalam penelitian ini dapat disimpulkan bahwa, hampir semua atribut yang ada pada entitas kendaraan maupun pelanggan dapat mereferensikan nilai atribut ke entitas lainnya melalui entitas tersarang tanpa harus membuat entitas yang baru, sehingga hal ini dapat mengurangi redundansi bahkan anomali input ketika data akan dimasukkan kedalam tabel.

DAFTAR PUSTAKA

- Abiteboul, S., Goldman, R., Mchugh, J., Vassalos, V., & Zhuge, Y. (1997). Views for Semistructured Data 1 Introduction 2 View Specification Language. *Workshop on Management of Semistructured Data*, 1–8.
- Arnicans, G., & Karnitis, G. (2006). Intelligent Integration of Information from Semi-Structured Web Data Sources on the Basis of Ontology and Meta-Models, 177–186.
- Aurral, A., Nur Azzahra Wedyati., Nikken Azzahara Z., Nadia Fathurrahmi Lawita, SE., M. Ak.,, 2020, Penerapan Sistem Informasi Manajemen Dalam Tata Kelola Perpustakaan Univesrsitas Muhammadiyah Riaui, *Jurnal Bisnis Darmajaya*, Vol.7. No. 1
- Dekhtyar, A., Goldsmith, J., & Hawkes, S. R. (2001). Semistructured probabilistic databases. *Proceedings Thirteenth International Conference on Scientific and Statistical Database Management. SSDBM 2001*, 36–45.
- Edhy, Sutanta (2004), *Sitem Basis Data*, Penerbit Graha Ilmu Yogyakarta
- Ganguly, R., & Sarkar, A. (2012). Evaluations of Conceptual Models for Semi-structured Database System. *International Journal of Computer Applications*, 50 (18), 5–12.
- Groppe, S. (2011). *Data Management and Query Processing in Semantic Web Databases. Uma ética para quantos?* (Vol. XXXIII).
- Martin, James. 1975. *Database Organization, Part 1*. New Jersey. Prentice Hall, Inc
- Mo, Y., & Ling, T. W. (2002). Storing and maintaining semistructured data efficiently in an object-relational database. *WISE 2002 - Proceedings of the 3rd International Conference on Web Information Systems Engineering*, 247–256.
- Pamuji, A., (2018). Perancangan Konseptual Sistem Basis Data Pada Pemodelan Object Relationship Attribute-Semi Structured (ORA-SS). *Jurnal Teknologi*, 10(2), 2085-1669.
- Sarkar, A. (2011). Conceptual Level Design of Semi-structured Database System: Graph-semantic Based Approach. *International Journal of Advanced Computer Science and Applications*, 2(10), 112–121.