2401960435 – Reynaldy Sentosa – UTS DBD.

1. Database System Development Lifecycle (DSLDC) merupakan pendekatan terstruktur dari pengembangan sebuah software basis data yang prosesnya diamati dari perspektif kebutuhan organisasi secara mendalam dan tahapan-tahapannya tidak berurutan begitu saja, tetapi melibatkan pengulangan dari tahap-tahap sebelumnya. Dalam hal ini, yang menjadi tahap awal adalah Database Planning. Tahap tersebut bertujuan agar tahapan-tahapan DSDLC berikutnya dapat berjalan secepat (efektif) dan sehemat (efisien) mungkin (Connolly & Begg, 2015, hal. 346-347).

2. Mission statement berisikan tujuan utama agar pembuatan sebuah sistem basis data dapat berjalan secara efisien dan efektif. Selanjutnya, mission objectives berisikan tugas-tugas tertentu yang harus dapat didukung oleh sistem basis data agar mission statement terpenuhi (Connolly & Begg, 2015, hal. 349). Dalam hal ini, mission statement dapat diperoleh dengan menanyakan tujuan perpustakaan, mengapa perpustakaan memerlukan basis data, dan bagaimana basis data dapat memecahkan masalah perpustakaan kepada (kepala) pustakawan. Singkat cerita, kita dapat mengasumsikan bahwa mission statement dari sistem basis data perpustakaan adalah: “Memelihara data-data yang dihasilkan dan digunakan untuk mendukung kegiatan peminjaman buku oleh pembaca terhadap pustakawan”. Selanjutnya, mission objectives dapat diperoleh dengan menanyakan peran, tugas atau tanggung jawab, bentuk data, jenis laporan, hal-hal yang perlu diawasi, dan layanan perpustakaan kepada para pustakawan. Singkat cerita, kita dapat mengasumsikan bahwa mission objectives dari sistem basis data perpustakaan adalah: “Memelihara (Insert, Update, dan Delete), melakukan pencarian terhadap, menyampaikan laporan tentang, dan mengawasi status data pustakawan, buku, pembaca / peminjam, dan detail peminjaman” (Connolly & Begg, 2015, hal. 386-392).

3. Database Design Methodology merupakan salah satu tahap DSDLC yang dilakukan setelah menganalisis keseluruhan kebutuhan (perusahaan) klien kita. Sebelumnya, Design Methodology mendukung tahap desain dengan sejumlah prosedur, teknik, dan alat dokumentasi untuk membantu para desainernya dalam menganalisis dan memodelkan serangkaian kebutuhan basis data secara terorganisir. Dalam hal ini, Conceptual Database Design memodelkan data-data milik klien secara konseptual, sehingga terbebas dari berbagai aspek fisik atau teknis. Selanjutnya, Logical Database Design memodelkan data-data konseptual tersebut secara spesifik. Terakhir, Physical Database Design menjelaskan bagaimana pengimplementasian sistem basis data mempengaruhi Logical Database Design. Untuk sistem basis data online shop, maka kita perlu memulai perancangannya dengan membuat Entity Relationship Diagram (ERD) pada tahap Conceptual Design. Hal ini dilakukan dengan menentukan para entitasnya, seperti produk, pembeli, transaksi, dsb. Kemudian, kita dapat menentukan seluruh atribut dari masing-masing entitas, misalnya entitas produk dapat beratributkan namaProduk, kodeProduk, dsb. Selanjutnya, kita perlu menentukan key atau atribut penting pada tiap entitas, misalnya kodeProduk dijadikan atribut perwakilan (primary key) pada entitas produk, dsb. Akhirnya, Conceptual Design dapat dilakukan dengan mengecek perulangan pada ERD terkini. Dengan demikian, tahap Conceptual Design ini dapat dibilang cukup mirip seperti tahapan awal Normalisasi. Berikutnya, kita terus memperbarui ERD pada tahap Logical Design. Hal ini dilakukan dengan menentukan apa saja jenis relationship beserta multiplicity, misalnya entitas pembeli berelasi dengan entitas produk. Dalam hal ini, pembeli membeli produk dan memiliki multiplicity (0..\* : 1..\*). Akhirnya, metodologi ini dilanjutkan dengan globalisasi ERD. Selanjutnya, kita beralih ke tahap Physical Database Design. Hal ini dilakukan dengan menentukan bagaimana hubungan dasar pada setiap atribut, misalnya atribut namaProduk bertipe data string, berukuran 255, dan tidak boleh bernilai NULL. Kemudian, kita dapat menambahkan CONSTRAINT yang dapat berguna saat CREATE ataupun UPDATE table. Setelah itu, kita dapat menentukan pengorganisasian file dan indexing, pemakaian disk space, dan keamanan sistem. Maka dari itu, metodologi ini terlihat cukup kompleks karena sudah merambah ke bidang pemrograman dan penyesuaian perangkat keras (Connolly & Begg, 2015, hal. 503-583).

4. Dalam membangun Logical Data Model pada Logical Database Design, awalnya kita perlu melakukan tahap penurunan hubungan. Dalam hal ini, kita perlu menentukan relasi antarentitas yang sudah diketahui atribut-atribut sederhananya. Dengan demikian, strong entity dapat dikatakan berperan sebagai parent entity karena dapat menyalin atribut perwakilannya sebagai primary key ke dalam weak entity sebagai child entity yang berupa foreign key agar antarentitas tersebut dapat berhubungan (Connolly & Begg, 2015, hal. 528 dan 530). Strong entity dapat berdiri sendiri karena pasti memiliki primary key, sedangkan keberadaan weak entity sangat bergantung pada strong entity (MKS075, 2021).

Contoh gambaran perpustakaan: <https://drive.google.com/file/d/1UwReOpVRhvUlHvpCs7UHCn-quMX9_3sV/view?usp=sharing>

Graphical user interface, table

Description automatically generated

Pada awalnya entitas BorrowHeader (Weak) belum memiliki primary key. Dikarenakan saya ingin menghubungkan entitas tersebut dengan entitas Member (Strong), maka entitas tersebut memperoleh salinan primary key (MemberId) milik entitas Member sebagai foreign key.

5a. Sebagai tahap awal Normalisasi, kita tidak perlu membuat unnormalized table karena soal tersebut sudah menggambarkan sebuah tabel yang memiliki barisan record data yang unik dengan sebuah value pada setiap kolom atributnya (Peterson, 2022). Dengan demikian, keberadaan repeating groups mungkin hanya dapat diasumsikan. Sebaliknya, kita perlu menentukan functional dependency atau hubungan dari satu atribut terhadap atribut lainnya (Connolly & Begg, 2015, hal. 456-459 dan 466).

Menentukan functional dependency:

<https://drive.google.com/file/d/16cZMtKew5AmjlXhWtjlURm06pVe4JmWR/view?usp=sharing>

A map of a city

Description automatically generated with medium confidence

Singkat cerita, kita dapat melihat gambar di atas bahwa terdapat sebuah full functional dependency. Hal ini dikarenakan id\_perawat mempengaruhi keseluruhan atribut dari nama\_perawat hingga jenis kelamin. Sebaliknya, tidak terdapat kebergantungan antara id\_perawat baik dengan Kode\_obat maupun id\_dokter (Connolly & Begg, 2015, hal. 459-460).

5b. Pada tahap 1NF, kita menggunakan hasil functional dependency sebelumnya untuk menentukan candidate key. Dengan demikian, kita dapat menyatakan atribut id\_perawat sebagai primary key karena memiliki partial dependency dengan atribut Nama\_perawat, alamat, No\_telepon, dan Jenis\_kelamin. Selanjutnya, kita pisahkan atribut-atribut tersebut ke dalam tabel berbeda dan menyambungkannya ke tabel awal dengan menyalin primary key sebagai foreign key.

Perawat\_Detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_perawat (PK) | Nama\_perawat | alamat | No\_telepon | Jenis\_kelamin |
| A1110 | Steva | Dago | 085234131818 | Perempuan |
| A1111 | Mumun | Buah Batu | 085234132424 | Laki-laki |
| A1112 | Ardi | Cipularang | 085234135656 | Laki-laki |
| A1114 | Cinta | Ciwastra | 085234131515 | Perempuan |

Perawat

|  |  |  |
| --- | --- | --- |
| id\_perawat (FK) ~~(PK)~~ | Kode\_obat ~~(PK)~~ | Id\_dokter ~~(PK)~~ |
| A1110 | 4 | 112160711 |
| A1111 | 2 | 112171205 |
| A1112 | 1 | 112161211 |
| **NULL** | 3 | 112080909 |
| A1114 | 5 | 112140207 |

Dikarenakan ketiga atribut pada tabel Perawat tidak saling berhubungan satu sama lain, maka kita bisa saja merangkumnya menjadi sebuah composite key. Namun demikian, atribut id\_Perawat pada salah satu record terlihat bernilai NULL. Hal ini mendorong kita agar dapat merevisinya, sehingga didapatlah hasil akhir proses Normalisasi pada tabel-tabel di bawah.

Perawat\_Detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_perawat (PK) | Nama\_perawat | alamat | No\_telepon | Jenis\_kelamin |
| A1110 | Steva | Dago | 085234131818 | Perempuan |
| A1111 | Mumun | Buah Batu | 085234132424 | Laki-laki |
| A1112 | Ardi | Cipularang | 085234135656 | Laki-laki |
| A1114 | Cinta | Ciwastra | 085234131515 | Perempuan |

Perawat

|  |  |  |
| --- | --- | --- |
| id\_perawat (PK) (FK) | Id\_obat (PK) (FK) | NoId\_dokter (PK) (FK) |
| A1110 | O004 | D003 |
| A1111 | O002 | D005 |
| A1112 | O001 | D004 |
| A1114 | O005 | D002 |

Obat

|  |  |
| --- | --- |
| Id\_obat (PK) | Kode\_obat |
| O001 | 1 |
| O002 | 2 |
| O003 | 3 |
| O004 | 4 |
| O005 | 5 |

Dokter

|  |  |
| --- | --- |
| NoId\_dokter (PK) | Id\_dokter |
| D001 | 112080909 |
| D002 | 112140207 |
| D003 | 112160711 |
| D004 | 112161211 |
| D005 | 112171205 |

a. Menentukan setiap entitas beserta atribut-atributnya.

1) Karyawan:

- IdKaryawan (PK)

- nama

- alamat

- kota

- status

- noTelp

2) Obat:

- IdSupplier (PK) (FK)

- IdObat (PK)

- nama

- jenis

- harga

- stok

3) Supplier:

- IdSupplier (PK)

- nama

- alamat

- kota

- noTelp

4) Penjualan:

- no (PK)

- IdKaryawan (FK)

- IdObat (FK)

- IdPelanggan

- tanggal

- jumlah

- total

- pajak

- totalBayar

5) Penyuplaian:

- no (PK)

- IdKaryawan (FK)

- IdSupplier (FK)

- IdObat (FK)

- tanggal

- jumlahObat

- total

- pajak

- totalBayar

b. Pada entitas pertama, saya hanya menggunakan sebuah primary key: IdKaryawan. Pada entitas kedua, saya menggunakan sebuah composite primary key yang salah satu atributnya (IdSupplier) juga berperan sebagai foreign key. Penggunaan pada entitas ketiga mirip entitas pertama: IdSupplier. Terakhir, pada entitas keempat dan kelima, saya menggunakan sebuah primary key dan sebuah composite foreign key.

c. Menentukan setiap relasi beserta kardinalitasnya:

<https://drive.google.com/file/d/1uQpKgNZNfKNQuU7R7FkirutoMxOz2e3h/view?usp=sharing>.

A picture containing text, indoor, sign

Description automatically generated

d. Crow’s Foot ERD:

<https://drive.google.com/file/d/1rFRxHXfWKw_wYTF93wWZaDyP7ULQOFLt/view?usp=sharing>.

A picture containing graphical user interface

Description automatically generated

# **Referensi**

Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management 6th Edition.* Harlow: Pearson Education Limited.

MKS075. (2021, Oktober 25). *Difference between Strong and Weak Entity*. Diambil kembali dari GeeksforGeeks: https://www.geeksforgeeks.org/difference-between-strong-and-weak-entity/

Peterson, R. (2022, April 16). *What is Normalization in DBMS (SQL)? 1NF, 2NF, 3NF, BCNF Database with Example*. Diambil kembali dari Guru99: https://www.guru99.com/database-normalization.html