

BAB II

LANDASAN TEORI

2.1 Sistem Informasi

Sistem informasi adalah komponen yang saling terkait yang mengumpulkan (memasukkan), memanipulasi (proses), dan menyebarkan (keluaran) data dan informasi dan memberikan mekanisme umpan balik untuk memenuhi tujuan organisasi [2].

Sistem informasi berisi berbagai informasi penting tentang orang, tempat, dan hal-hal di dalam atau sekitar organisasi. Informasi ini menggambarkan sebuah organisasi atau salah satu sistem kuncinya tentang apa yang terjadi di masa lalu, sekarang, dan yang mungkin terjadi di masa depan [3].

Penggunaan dari sistem informasi ditujukan untuk mengolah berbagai informasi yang dikelola oleh organisasi, sehingga sumber daya yang dibutuhkan tidak terlalu besar dan dapat mempersingkat waktu penanganan proses.

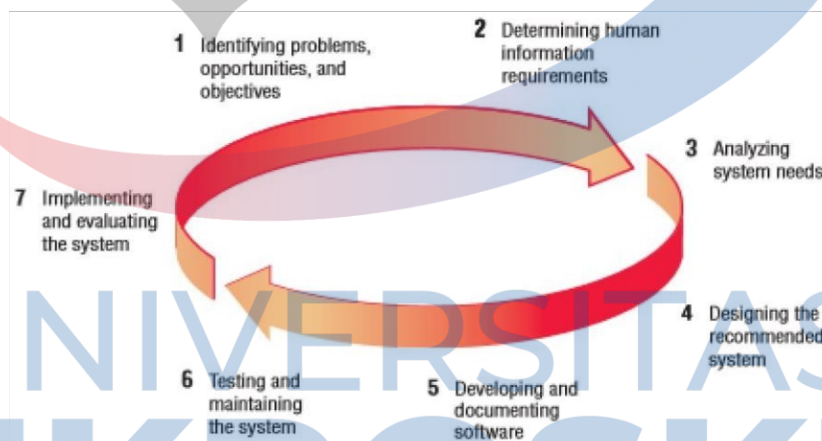
2.2 Digital Library

Digital Library adalah organisasi yang menyediakan sumber daya, termasuk staf khusus, untuk memilih, menyusun, menawarkan akses intelektual, menafsirkan, mendistribusikan, melestarikan integritas, dan memastikan kegigihan dari waktu ke waktu koleksi karya digital sehingga mereka siap dan tersedia secara ekonomis untuk digunakan oleh komunitas tertentu atau sekumpulan komunitas.

Digital Library dikaitkan dengan organisasi yang mengoleksi rujukan ke sumber daya yang berbasis web di internet. *Digital Library* tidak lagi diukur berdasarkan jumlah dan karakter koleksi yang dimiliki secara fisik melainkan lebih berdasarkan luas cakupan jaringan informasi yang terbentuk oleh jasa yang disediakan[4].

2.3 Metodologi SDLC

Sistem Development Life Cycle (SDLC) atau Siklus Hidup Pengembangan Sistem adalah suatu metode tradisional yang digunakan untuk membangun, memelihara dan mengganti suatu sistem informasi. SDLC dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Beberapa contoh metodologi pengembangan perangkat lunak yang tersedia, antara lain *waterfall*, *prototyping*, *incremental*, *spiral*, RAD. Siklus Hidup Pengembangan Sistem merupakan penerapan pendekatan sistem untuk mengembangkan dan menggunakan sistem berbasis komputer. Konsep ini umumnya merujuk pada sistem komputer atau informasi SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak[5].



Gambar 2.1 Tahapan SDLC

Berikut Tahapan dalam SDLC [6] :

1. Mengidentifikasi Masalah, Peluang dan Tujuan

Fase ini adalah yang pertama dilakukan, dalam SDLC, analisis harus bisa dengan benar menganalisa masalah, peluang dan tujuan.

Fase ini akan memiliki pengaruh yang besar pada fase selanjutnya, karena jika ada kesalahan dalam fase ini maka kemungkinan akan terjadi kesalahan dalam fase berikutnya.

2. Menentukan Syarat-Syarat Informasi

Tahap ini menentukan penentuan kebutuhan dari pengguna yang terlibat dan memahami bagaimana pengguna berinteraksi dengan dalam konteks kerja dalam sistem informasi. Analisis dilakukan dengan metode interaktif seperti wawancara, pengambilan sampel, dan kuesioner. Termasuk metode yang dengan menggunakan metode pengamatan perilaku para pengambil keputusan dan lingkungan.

3. Menganalisis Kebutuhan Sistem

Kebutuhan sistem dianalisis di tahapan ini. Perangkat yang dibutuhkan untuk membantu analisis dalam menentukan kebutuhan dapat meliputi diagram aliran data. Pada tahapan ini keputusan terstruktur dimana kondisi alternatif, tindakan serta aturan tindakan yang ada juga turut dianalisis.

4. Merancang Sistem yang Direkomendasikan

Tahapan ini menuntun analisis sistem untuk mulai melakukan desain sistem informasi yang logis. Perangkat yang digunakan dapat meliputi peralatan antarmuka pengguna.

5. Mengembangkan dan Mendokumentasikan Perangkat Lunak

Pada tahapan ini sistem *analisis* bekerja sama dengan pemrogram untuk mulai mengembangkan perangkat lunak. Teknik untuk merancang dan mendokumentasikan perangkat lunak seperti *Nassi-Shneiderman charts*, dan *pseudocode* juga turut dilibatkan.

6. Menguji dan Mempertahankan Sistem

Kegiatan pengujian dilaksanakan pada tahapan ini, pengujian dapat dilakukan dalam beberapa tahapan baik oleh pemrogram sendiri, bersama dengan *analisis* sistem bahkan dapat diujikan mulai dengan menggunakan data contoh hingga data aktual dari sistem yang ada.

7. Mengimplementasikan dan Mengevaluasi Sistem

Tahapan ini merupakan tahapan dimana sistem analisis melakukan *implementasi* sistem yang sudah dikembangkan. Kegiatan ini melibatkan pelatihan dan *konversi* dari sistem lama ke sistem baru. Evaluasi disertakan sebagai bagian dari tahapan akhir dari SDLC untuk kepentingan diskusi.

2.4 PIECES

Analisis PIECES merupakan metode yang digunakan untuk mengidentifikasi masalah yang terjadi, dengan melakukan analisis masalah terhadap kinerja sistem informasi, ekonomi, keamanan aplikasi, efisiensi, dan pelayanan pelanggan[7].

Metode PIECES memiliki enam variabel penilaian dalam melakukan evaluasi[8]:

a. *Performance* (Kinerja)

Menilai Apakah proses atau prosedur yang ada masih mungkin ditingkatkan kinerjanya. Juga dengan response time yaitu waktu yang dibutuhkan untuk menyelesaikan serangkaian kegiatan untuk menghasilkan output atau deliverables tertentu.

b. *Information* (Informasi)

Sebuah penilaian Apakah prosedur yang ada saat ini masih dapat diperbaiki sehingga kualitas informasi yang dihasilkan menjadi semakin baik.

c. *Economics* (Ekonomi)

Menilai Apakah prosedur yang ada saat ini masih dapat ditingkatkan manfaatnya (nilai gunanya) atau diturunkan biaya penyelenggaraannya.

d. *Control* (Kontrol)

Menilai Apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian menjadi semakin baik, dan kemampuannya untuk mendeteksi kesalahan atau kecurangan menjadi semakin baik pula.

e. *Efficiency* (Efisiensi)

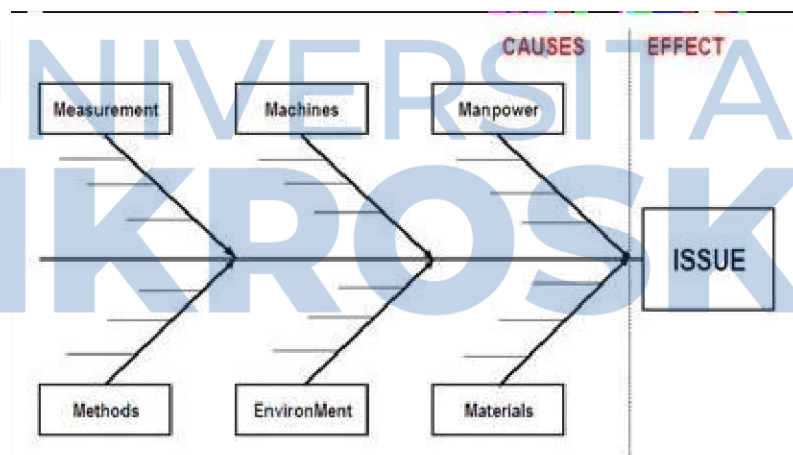
Menilai Apakah prosedur yang ada saat ini masih dapat diperbaiki sehingga tercapai peningkatan efisiensi operasi.

f. *Service* (Layanan)

Menilai Apakah prosedur yang ada saat ini masih dapat diperbaiki kemampuannya untuk mencapai peningkatan kualitas layanan.

2.5 *Fishbone Diagram*

Diagram Tulang Ikan atau *Fishbone Diagram* dikembangkan oleh Dr Kaoru Ishikawa di Universitas Tokyo pada tahun 1943. Dengan demikian, *Fishbone Diagram* juga disebut sebagai diagram "*Ishikawa*". Diagram ini juga disebut sebagai "*Cause & Effect*" atau diagram CE. Diagram ini banyak digunakan untuk mengidentifikasi semua akar penyebab yang berkontribusi dari proses yang mungkin menghasilkan masalah. *Diagram Fishbone* umumnya digunakan sebagai langkah awal saat melakukan screening awal untuk aktivitas *Process Improvement*. Diagram digunakan untuk menjawab beberapa pertanyaan dasar yang sering muncul selama latihan pemecahan masalah[9].



Gambar 2.2 Contoh Diagram *Fishbone*

Diagram dirancang sebagai struktur tulang ikan. Ini pada dasarnya memiliki lima elemen utama [10]:

Man: terdiri dari sebab-sebab yang timbul karena campur tangan manusia dalam prosesnya

Machine: terdiri dari sebab-sebab yang timbul karena gangguan mekanis dalam proses.

Method: terdiri dari sebab-sebab yang timbul karena metode proses.

Material: terdiri dari sebab-sebab yang timbul karena material yang digunakan dalam proses.

Measurement: terdiri dari sebab-sebab yang timbul akibat teknik pengukuran dalam prosesnya.



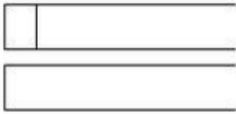
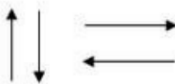
Mother Measure: terdiri dari sebab-sebab yang dialami dan timbul karena faktor eksternal lain yang terlibat dengan proses.

2.6 Data Flow Diagram

Data Flow Diagram (DFD) atau disebut juga dengan Diagram Aliran Data (DAD) merupakan diagram yang menggambarkan aliran data suatu sistem. Alur yang digambarkan adalah alur input-proses output. Biasa digunakan untuk menjelaskan proses apa saja yang terjadi, apa masukannya, serta apa keluarannya dari suatu sistem. DFD memetakan arus data untuk setiap proses pada sistem informasi. DFD menunjukkan data yang akan dimasukkan ke proses dan output dari proses tersebut, bagaimana data mengalir dari satu proses ke proses lainnya, dan di mana data akan disimpan. DFD tidak menunjukkan informasi tentang waktu proses atau apakah proses akan beroperasi secara berurutan atau paralel[11].

DFD menggunakan simbol grafis seperti persegi panjang, lingkaran dan panah. serta label teks pendek, untuk menampilkan input data, atau keluaran. DFD bisa digunakan untuk menganalisis sistem yang berjalan. Selain itu DFD secara visual menggambarkan hal-hal yang sulit dijelaskan dengan kata-kata, baik untuk khalayak teknis dan nonteknis, mulai dari pengembang hingga *stakeholder*. Meskipun DFD bekerja dengan baik untuk menggambarkan arus data pada sistem[12].

Tabel 2.1 Simbol - simbol pada *Data Flow Diagram*

Simbol	Keterangan
	<i>External Entity</i> , merupakan kesatuan di lingkungan luar sistem yang bisa berupa orang, organisasi atau sistem lain.
	<i>Process</i> , merupakan proses seperti perhitungan aritmatik penulisan suatu formula atau pembuatan laporan
	<i>Data Store</i> (Simpan Data), dapat berupa suatu file atau database pada sistem komputer atau catatan manual
	<i>Data Flow</i> (arus data), arus data ini mengalir diantara proses, simpan data dan kesatuan luar

Data Flow Diagram mencakup empat elemen utama:

a. Entitas Eksternal

Juga dikenal sebagai aktor, sumber (*source*), atau terminator. Entitas eksternal merupakan segala sesuatu di luar sistem yang menghasilkan dan atau mengonsumsi data yang mengalir pada sistem yang digambarkan. Karena mereka berada di luar sistem yang dianalisis, entitas ini biasanya ditempatkan pada tepi diagram. Entitas eksternal tidak hanya manusia namun bisa mewakili sistem lain.

b. Proses

Merupakan aktivitas yang mengolah data atau memproduksi informasi. Semua proses pada DFD harus memiliki input dan output. Proses biasanya berorientasi dari atas ke bawah atau kiri ke kanan pada pembuatan laporan.

c. *Data Store*

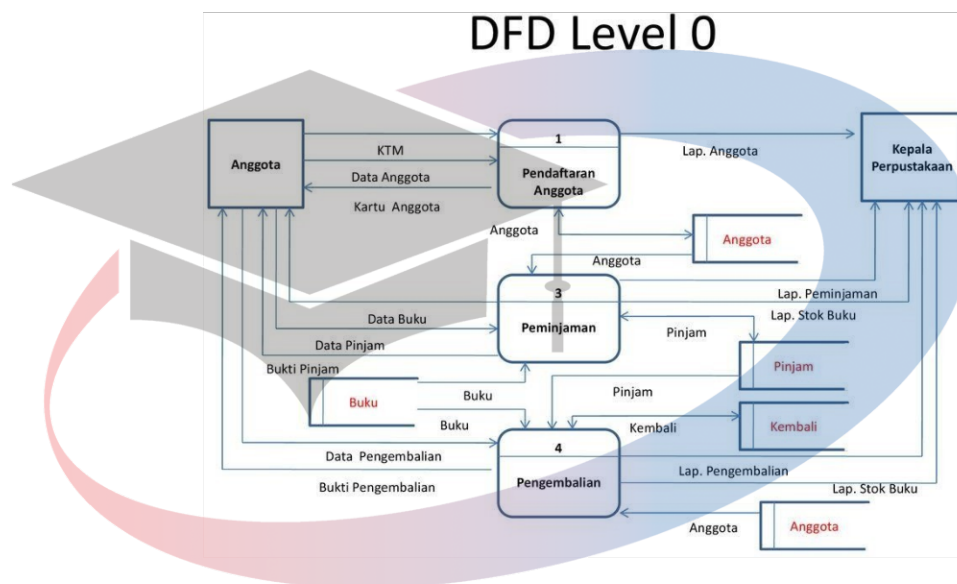
Data Store (penyimpanan data) tidak menghasilkan operasi apapun melainkan hanya menyimpan data untuk proses selanjutnya. *Data Store* bisa berupa tabel pada sebuah basis data atau sekumpulan dokumen yang tersimpan sementara menunggu untuk diproses. Input mengalir ke *Data Store* meliputi operasi yang mengubah data yang tersimpan. Arus output-nya adalah data yang diambil.

d. Aliran Data

Data flow (aliran data) merupakan pergerakan data antara entitas eksternal, proses dan penyimpanan data yang diwakili dengan simbol garis panah, yang mengindikasikan arah arus. Data ini bisa berupa elektronik, tertulis atau lisan. Aliran data dapat diberi label berdasarkan jenis data atau proses yang terkait. Label ini ditulis di atas di bawah garis panah.

DFD diberi tingkatan 0, 1, 2, 3 atau lebih selama dibutuhkan. Tingkat detail yang diperlukan tergantung pada lingkup dari detail apa yang ingin dicapai.

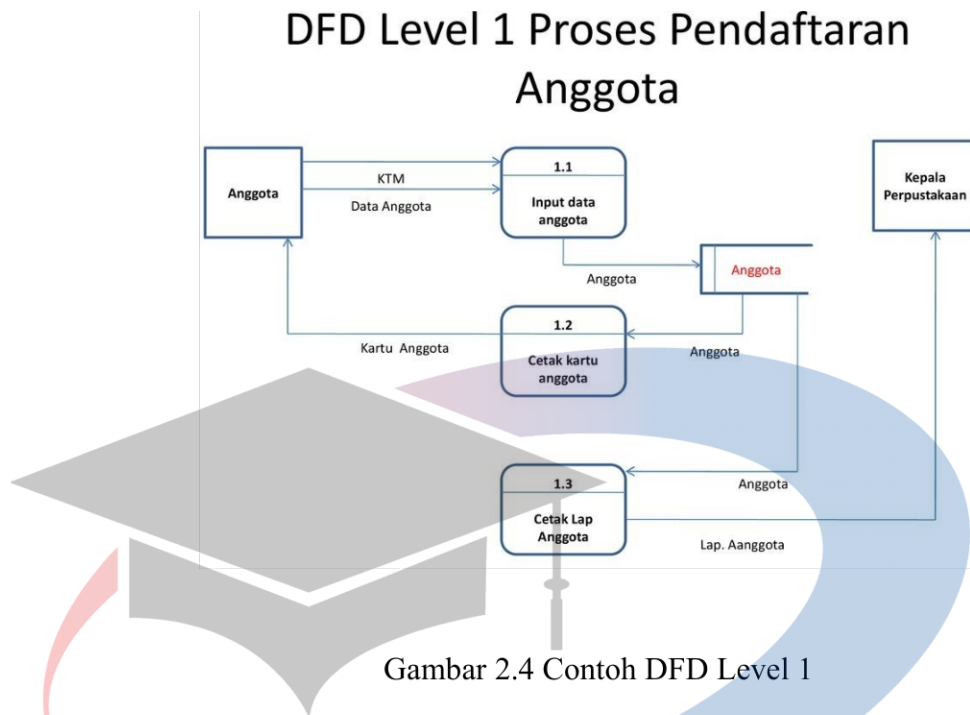
DFD Level 0 juga disebut Diagram Konteks. Diagram ini dirancang untuk menjadi pandangan keseluruhan secara sekilas yang menunjukkan sistem sebagai satu proses dan hubungannya dengan entitas eksternal. Diagram ini harus mudah dipahami oleh khalayak luas, termasuk *stakeholder*, analis bisnis, analis data dan pengembang.



Gambar 2.3 Contoh DFD Level 0

Setelah selesai membuat DFD level 0, maka tahap selanjutnya adalah merinci setiap proses yang ada pada DFD level 0, sehingga setiap *event* yang ada dalam suatu proses dapat digambarkan menjadi lebih detail pada DFD level 1. DFD level 1 bertujuan untuk memberikan pandangan mengenai keseluruhan sistem dengan lebih mendalam. Proses-proses utama yang ada akan dipecah menjadi sub-proses. *Data Store* yang digunakan dalam proses-proses utama juga diidentifikasi dalam DFD level

1. DFD Level 1 menyediakan detail bagian-bagian diagram level konteks yang lebih rinci, level 1 akan menyoroti fungsi utama yang dilakukan oleh sistem saat memecah proses diagram konteks menjadi sub prosesnya. Pada DFD level 1 merupakan sub sistem/sub-proses pada DFD level 0. Sebagai contoh pada DFD level digambarkan konteks sistem perpustakaan maka pada level 1 dapat berupa proses peminjaman buku, proses pengembalian buku dan proses daftar anggota. Jika diperlukan maka akan DFD akan dilanjutkan ke level yang lebih detail lagi[12].



2.7 Kamus Data

Kamus data (*data dictionary*) berisi penjelasan rinci yang dibuat oleh perancang basis data tentang semua tabel yang terdapat dalam basis data. Dengan demikian, kamus data berisi semua atribut nama dan karakteristik untuk setiap tabel dengan kata lain kamus data berisi meta data tabel-tabel dalam basis data. Fungsi dari kamus data ini adalah sebagai alat bantu komunikasi untuk memastikan semua anggota tim desain dan implementasi basis data menggunakan tabel, atribut dan karakteristiknya yang sama[13].

Kamus data pertama kali muncul berbasis kamus dokumen tersimpan dalam suatu bentuk *hard copy* dengan mencatat semua penjelasan data dalam bentuk dokumen yang dicetak. Saat ini kamus berbasis dokumen sudah sangat langka dijumpai, praktik yang umum saat ini adalah mempergunakan kamus data yang berbasis komputer. Pada kamus data berbasis komputer, penjelasan data dimasukkan ke dalam komputer dengan melalui DBMS. Kamus data pada umumnya merupakan salah satu fitur bawaan dari RDBMS. Kamus data pada RDMBS disimpan secara internal dan berisi informasi tambahan tentang tipe hubungan, pemeriksaan integritas entitas dan integritas referensial, dan indeks[13].

Tabel 2.2 Simbol-simbol Kamus Data

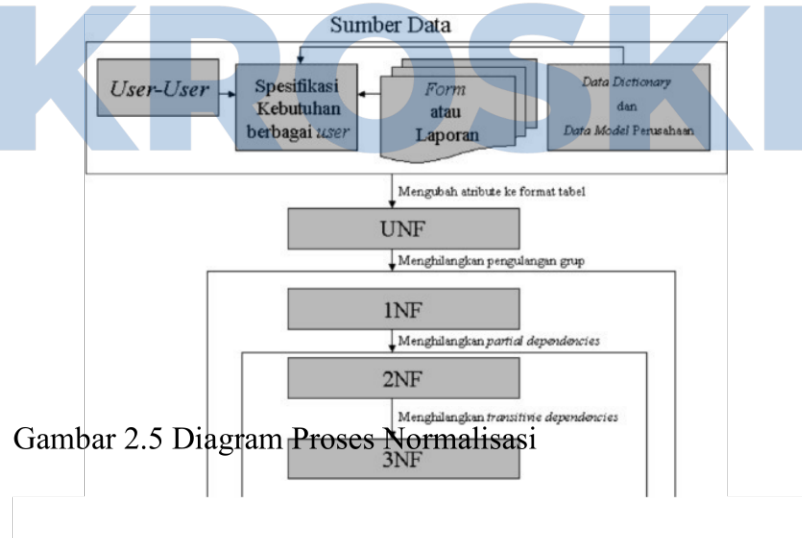
Notasi Struktur Data	
Notasi	Keterangan
=	Terdiri dari
+	Dan atau <i>And</i>
()	Pilihan optional
{ }	<i>Iterasi</i> (Perulangan proses)
[]	Pilih salah satu pilihan yang ada
	Pemisah pilihan didalam tanda []
*	Keterangan atau catatan
@	Field Kunci

2.8 Normalisasi

Normalisasi adalah suatu teknik formal yang dapat digunakan dalam perancangan basis data. Peranan normalisasi dalam hal ini adalah penggunaan pendekatan *bottom-up* dan teknik validasi. Teknik validasi digunakan untuk memeriksa apakah struktur relasi yang dihasilkan oleh *ER modelling* itu baik atau tidak baik[14].

Terdapat empat bentuk normal yang biasa digunakan, yaitu:

1. *First Normal Form* (1NF) atau Normalisasi Tingkat 1
2. *Second Normal Form* (2NF) atau Normalisasi Tingkat 2
3. *Third Normal Form* (3NF) atau Normalisasi Tingkat 3
4. *Boyce-Codd Normal Form* (BCNF)



Gambar 2.5 Diagram Proses Normalisasi

1. *Unnormalized Form* (UNF)

Merupakan suatu tabel yang berisikan satu atau lebih grup yang berulang. Membuat tabel yang unnormalized adalah dengan memindahkan data dari sumber informasi. Contohnya, nota penjualan yang disimpan ke dalam format tabel dengan baris dan kolom.

2. *First Normal Form* (1NF)

Merupakan sebuah relasi di mana setiap baris dan kolom berisikan hanya satu nilai.

3. *Second Normal Form* (2NF)

2NF merupakan sebuah relasi dalam 1NF dan setiap atribut *non-primary key* bersifat *fully functionally dependent* pada *primary key*.

4. *Third Normal Form* (3NF)

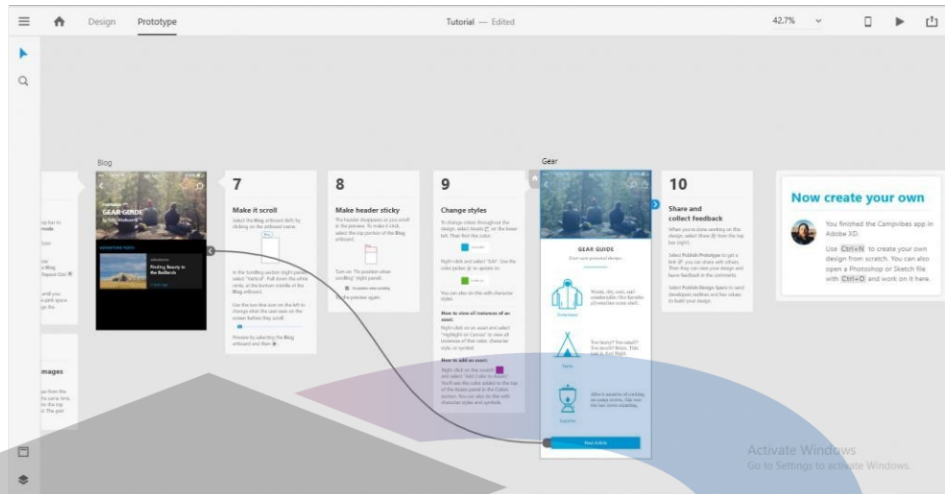
3NF adalah sebuah relasi dalam 1NF dan 2NF dan di mana tidak terdapat atribut *nonprimary key* yang bersifat *transitively dependent* pada *primary key*.

5. *Boyce-Codd Normal Form* (BCNF)

Berdasarkan pada *functional dependencies* yang dimasukkan ke dalam hitungan seluruh *candidate key* dalam suatu relasi, bagaimana pun BCNF juga memiliki batasan-batasan tambahan disamakan dengan definisi umum dari 3NF. Suatu relasi dikatakan BCNF, hanya jika *candidate key*. Setiap relasi dalam BCNF juga merupakan 3NF, tetapi relasi dalam 3NF belum tentu termasuk ke dalam BCNF. Dalam BCNF kesalahan jarang sekali terjadi. Kesalahan dapat terjadi pada relasi yang terdiri dari 2 atau lebih *composite candidate key*, *candidate key overlap*, sedikitnya satu atribut.

2.9 Adobe Xd

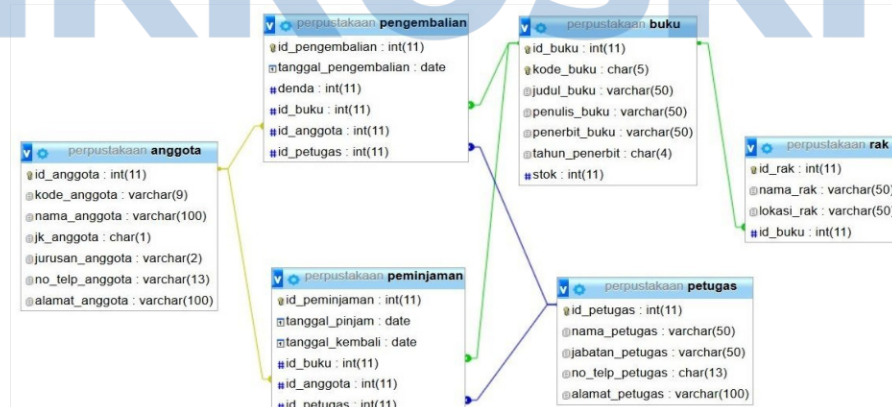
Adobe Xd adalah program untuk membuat maket dan prototipe aplikasi seluler, web, dan perangkat lunak menggunakan serangkaian *artboards* dan menautkannya. Sering digunakan oleh *UI/UX designer* berpengalaman untuk menguji tampilan visual dari desain antarmuka mereka sebelum diserahkan kepada pengembang untuk kemudian dibangun menjadi serangkaian kode. Dengan cara ini, desainer dapat menyelesaikan tata letak dan alur pengguna dari kreasi mereka tanpa pengetahuan pengkodean dan melakukan penyesuaian dan perubahan desain dengan mudah[15].



Gambar 2.6 Contoh Tampilan Adobe Xd

2.10 Mysql

Mysql adalah sebuah program database server yang mampu menerima dan mengirimkan datanya sangat cepat, multi user serta menggunakan perintah dasar SQL (*Structured Query Language*). *Mysql* merupakan sebuah database server yang free, artinya bebas menggunakan database untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. Selain database server, *Mysql* juga merupakan program yang dapat mengakses suatu database *Mysql* yang berposisi sebagai server, yang berarti program kita berposisi sebagai client. Jadi *Mysql* adalah sebuah database yang dapat digunakan sebagai client maupun server database *Mysql* merupakan suatu perangkat lunak database yang berbentuk database relasional atau disebut *Relational Database Management Sistem* (RDBMS) yang menggunakan suatu Bahasa permintaan yang bernama SQL (*Structured Query Language*)[16].



Gambar 2.7 Contoh tampilan database Mysql