**INPUT NILAI**

**PROPOSAL**

****

OLEH :

**DEVI RIDMAYANA**

2002505010

**FAKULTAS ILMU KOMPUTER**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS TOMAKAKA MAMUJU**

**2021**

**KATA PENGANTAR**

Puji syukur saya panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayahnya sehingga saya dapat menyelesaikan proposal “**INPUT NILAI**”

Proposal ini disusun untuk memenuhi tugas MID (uts) mata kuliah Pemrograman Mobile. Selain itu, proposal ini juga bertujuan untuk menambah wawasan tentang perancangan dan pembuatan aplikasi.

Saya mengucapkan terima kasih kepada Bapak Musliadi KH, S.Kom,M.Kom. selaku dosen pengampuh mata kuliah Pemrograman Mobile yang telah memberikan tugas ini sehingga dapat menambah pengetahuan dan wawasan sesuai program studi yang saya tekuni. Saya juga mengucapkan terima kasih kepada semua pihak yang telah membagi sebagian pengetahuannya sehingga saya dapat menyelesaikan proposal ini.

Dalam penyusunan proposal ini saya sangat menyadari akan banyaknya kekurangan dari berbagai sisi. Karena itu, saya mengharapkan kritik dan saran yang sifat nya membangun demi kesempurnaan penulisan kedepannya.

Mamuju, 25 November 2021

Devi ridmayana

**DAFTAR ISI**

**HALAMAN JUDUL i**

**KATA PENGANTAR ii**

**DAFTAR ISI iii**

**BAB I PENDAHULUAN**

* 1. **Latar Belakang iv**
  2. **Rumusan Masalah v**
  3. **Batasan Masalah vi**
  4. **Tujuan dan Manfaat Penelitian vii**
     1. **Tujuan Penelitian 1**
     2. **Manfaat Penelitian 2**

**BAB II LANDASAN TEORI**

**2.1 Alat Perancang Sistem viii**

**2.1.1 Alat Bantu Perancang Sistem 3**

**2.1.2 Alat Bantu Perancangan Logika Program 4**

**2.1.3 Alat Bantu Perancangan Databes 5**

**2.2 Sistem Basis Data ix**

**2.2.1 Pengertian Basis Data 7**

**2.2.2 Normalisasi 8**

**2.3 Adobe XD x**

**2.3.1 Pengertian Adobe XD 9**

**2.3.2 Fungsi Adobe XD 10**

**2.4 Android Studio xii**

**2.4.1 Pengertian Android Studio 11**

**2.4.2 Fungsi Android Studio 12**

**2.5 Waterfall (Air Terjun) xiii**

**BAB III PERANCANGAN SISTEM**

**3.1 Waktu Dan Tempat Penelitian xiv**

**3.1.1 Waktu Penelitian 13**

**3.1.2 Tempat Penelitian 14**

**3.2 Metode Penelitian xv**

**3.2.1 Metode Waterfall (Air Terjun) 15**

**3.3 Teknik Pengumpulan Data xvi**

**3.4 Analisa Sistem Berjalan xvii**

**3.5 Rancangan Sistem Yang Diusulkan xviii**

**3.6 Instrumen Penelitian xix**

**3.6.1 Perangkat Keras (Hardware) 16**

**3.7 Jadwal Penelitian xx**

**PENDAHULUAN**

## Latar Belakang

Perkembangan telepon selular (handphone) pada jaman sekarang sudah mengalami kemajuan yang sangat pesat, menjadi suatu hal yang berguna dan penting bagi masyarakat. Perkembangan teknologi pada telepon selular sekarang sudah berkembang dengan menggunakan teknologi sistem operasi Android yang memungkinkan penggunaan telepon selular untuk membuat aplikasi berbasis sistem operasi Android yang dapat dijalankan pada telepon selular yang mendukung Android platform.

Untuk menyelesaikan tugas MID (uts) dan Final (UAS) kami diberikan tugas untuk membuat proposal dan aplikasi yang dimana kita menyetor judul aplikasi yang ingin kita rancang dan dibuat. Yang nantinya dibuatkan proposal, untuk aplikasinya dirancang di Adobe XD serta dibuat di Android Studio.

## Rumusan Masalah

Bagaimana perancangan dan pembuatan aplikasi input nilai dari sebuah nilai yang kita input ke dalam aplikasinya agar nanti aplikasinya dapat memproses nilai yang kita input hasilnya menjadi nilai huruf dan sebuah keterangan.

## Batasan Masalah

Dalam perancangan dan pembuatan aplikasi input nilai berbasis android studio mempunyai batasannya terdiri dari :

1. Aplikasi ini dibuat secara sederhana
2. Aplikasi ini tidak menghitung sebuah nilai
3. Aplikasi ini hanya dapat memproses nilai yang kita input menjadi nilai huruf dan keterangan.

## Tujuan dan Manfaat Penelitian

### Tujuan Penelitian

Tujuan perancagan dan pembuatan aplikasi ini bertujuan untuk menginput dan melihat hasil nilai dan keterangan dari nilai yang kita input.

### Manfaat Penelitian

Manfaatnya terbagi menjadi :

1. Manfaat untuk penulis, menambah ilmu dan pengetahuan tentang perancagan dan pembuatan sebuah aplikasi.
2. Manfaat untuk pembaca, dapat dipergunakan sebagai referensi untuk para pembacanya.

**LANDASAN TEORI**

## Alat Perancangan Sistem

Menurut Asmy, (2016)Pada dasarnya alat perancangan sistem informasi terbagi 3 bagian yaitu :

1. ASI (Aliran Sistem Informasi)
2. Context Diagram
3. DFD (Data Flow Diagram)

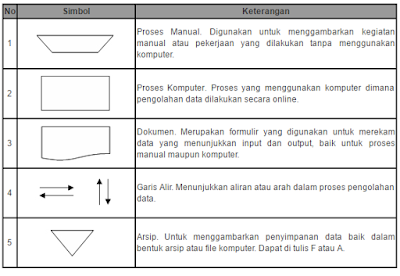
### Alat Bantu Perancangan Sistem

Alat bantu perancangan sistem terbagi atas 3 bagian, yaitu :

1. ASI (Aliran Sistem Informasi)
2. Context Diagram
3. DFD (Data Flow Diagram)
4. **Aliran Sistem Informasi (ASI)**

Aliran sistem informasi sangat berguna untuk mengetahui permasalahan yang adda pada suatu sistem. Dari sini dapat diketahui apakah system informasi tersebut masih layak dipakai atau tidak, masih manual atau komputerisasi. Jika sistem informasinya tidak layak lagi maka perlu adanya perubahan dalam pengolahan datanya sehingga menghasilkan informasi yang cepat dan akurat serta keputusan yang lebih baik.

Berikut simbol-simbol Aliran Sistem Informasi (ASI) :

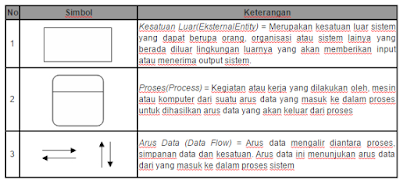
[](https://4.bp.blogspot.com/-tVauFJCVWgs/VwT_etOFdLI/AAAAAAAAA5I/SA7JsSR-wjEIXw3GtAG8xxR3qt0DJ9tTw/s1600/1.png)

1. **Context Diagram**

Context Diagram adalah gambaran umum tentang suatu sistem yang terdapat didalam suatu organisasi yang memperlihatkan batasan (boundary) sistem, adanya interaksi antara eksternal entity  dengan suatu sistem  dan informasi secara umum mengalir diantara entity dan sistem. Context Diagram merupakan alat bantu yang digunakan dalam menganalisa sistem yang akan dikembangkan.

Simbol-simbol yang digunakan di dalam Context Diagram hampir sama dengan simbol-simbol yang ada pada DFD, hanya saja pada Context Diagram tidak terdapat simbol file.

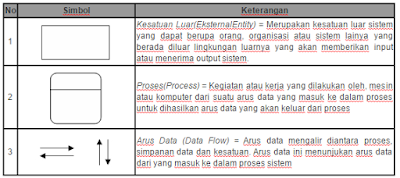
Berikut simbol-simbol Context Diagram

[](https://2.bp.blogspot.com/-kT-Ko7rv9rY/VwT_x7TXgpI/AAAAAAAAA5M/HD9SddUETZIdeVa9oEYOb_7EQN_qhCiyA/s1600/2.png)

1. **Data Flow Diagram (DFD)**

DFD merupakan ganbaran sistem secara logika yang tidak tergantung pada perangkat keras, lunak, struktur data dan organisasi file. Keuntungan dari DFD adalah untuk memudahkan pemakai yang kurang menguasai bidang komputer untuk mengerti system yang akan dikerjakan atau dikembangkan.

Berikut simbol-simbol dari Data Flow Diagram (DFD) :

[](https://4.bp.blogspot.com/-YWw0Tnqd5zg/VwT_-Nx2ooI/AAAAAAAAA5Q/cyneg1hWQ7YJP6czsQTb_s5q_VAPanCZw/s1600/3.png)

1. **Aturan dalam pembuatan DFD**
2. Tidak boleh menghubungkan external entity ke external entity secara langsung.
3. Tidak boleh menghubungkan data storage ke data storage lainnya secara langsung.
4. Tidak boleh menghubungkan data storage dengan external entity secara lansung.
5. Pada setiap proses harus ada data flow masuk dan keluar dan sebaliknya.
6. Tidak boleh ada proses dari arus data tidak memiliki nama (nama harus ada)
7. Tidak boleh ada proses yang tidak memiliki nomor.
8. **Metode pembuatan DFD**
9. Mulai dari yang umum sampai yang detail
10. Jabarkan setiap proses
11. Pelihara konsistensi antar proses
12. Berikan label nama yang bermakna untuk ke empat simbol tersebut
13. Menjaga konsistensi dengan model lainnya.
14. **Tahapan Pembuatan DFD**
15. Buat Context Diagram (Top Level Diagram)
16. Buat diagram level 0
17. Buat diagram level1 (diagram detail)

Diagram ini digunakan untuk menjelaskan tahapan-tahapan proses dari diagram level 0.

1. **Cara pembuatan DFD**
2. Identifikasi semua external entity dalam sistem
3. Identifikasi semua input dan ouput yang terlibatdengan external entity.
4. Urutan pengambaran dimulai dari context diagram, diagram level 0, diagram level1.

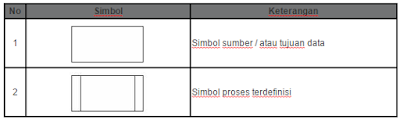
### Alat Bantu Perancangan Logika Program

Adapun alat bantu dalam pernacngan logika program terdiri atas 2 bagian yaitu :

1. Struktur Program
2. Flowchart
3. **Struktur Program**

Menggambarkan menu utama pada program yang akan dirancang juga menampilkan apa yang dikerjakan pada sebuah sistem atau membuat bagian bentuk spesifikasi dari modul-modul program yang dikerjakan pada sebuah sistem.

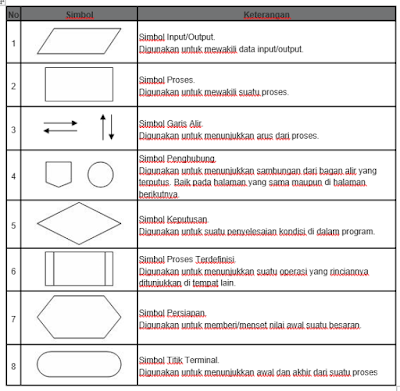
Berikut simbol-simbol dari struktur program :

[](https://2.bp.blogspot.com/-KKVmuy0mb5c/VwUAZm7Q1TI/AAAAAAAAA5Y/fc5Lt7pL0T0prd2YMO-dzuZUzbqPZxzug/s1600/4.png)

1. **Flowchart**

Program Flowchart adalah diagram alir yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Bagan alir program (program flowchart) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program.

Berikut simbol-simbol dari Flowchart :

[](https://4.bp.blogspot.com/-tYPWQbltreo/VwUBYnY2oGI/AAAAAAAAA5o/nGJMLSPKgpAaawh7v9BbYXGw64IBAWZkw/s1600/5.png)

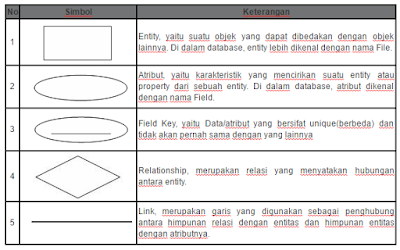
### Alat Bantu Perancngan Database

Adapun alat bantu dalam perancagan database terdiri dari 2 bagian yaitu :

1. Entity Relationship Diagram (ERD)
2. Normalisasi Database
3. **Entity Relationship Diagram (ERD)**

Model ERD berisi komponen-komponen entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan  seluruh fakta yang ditinjau sehingga dapat diketahui hubungan antara entity-entity yang ada dengan atribut-atributnya. Selain itu juga bisa menggambarkan hubungan yang ada dalam pengolahan data, seperti hubungan many to many, one to many, one to one.

Berikut simbol-simbol dari ERD

[](https://4.bp.blogspot.com/-F_AvxOpQKFk/VwUB9M72Z0I/AAAAAAAAA5w/3lTpYmjRdQQCdGmMyco4Wol8YZPDWgzjA/s1600/6.png)

1. **Normalisasi Database**

Normalisasi adalah suatu teknik untuk meng-organisasi data ke dalam tabel-tabel  untuk memenuhi kebutuhan pemakai di dalam suatu organisasi. Adapun tujuan dari normalisal adalah :

1. Untuk menghilangkan kerangkapan data
2. Untuk mengurangkan kompleksitas
3. Untuk mempermudah pemodifikasian data

Normalisasi database biasanya jarang dilakukan dalam database skala kecil, dan dianggap tidak diperlukan pada penggunaan personal. Namun seiring dengan berkembangnya informasi yang dikandung dalam sebuah database, proses normalisasi akan sangat membantu dalam menghemat ruang yang digunakan oleh setiap tabel di dalamnya, sekaligus mempercepat proses permintaan data.

1. **Proses Normalisasi**
2. Data diuraikan dalam bentuk tabel, selanjutnya  dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
3. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.
4. **Tahap-tahap normalisasi :**

Bentuk Tidak Normal *(Menghilangkan perulangan group)* Bentuk Normal Pertama **(1NF)***(Menghilangkan ketergantungan sebagian)* Bentuk Normal Kedua **(2NF)** *(Menghilangkan ketergantungan transitif)*Bentuk Normal Ketiga **(3NF)** *(Menghilangkan anomali-anomali hasil dari ketergantungan fungsional).*

1. **Bentuk Normal Kesatu (1NF)**

Suatu relasi dikatakan sudah memenuhi Bentuk Normal Kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data.

1. **Bentuk Normal Kedua (2NF)**

Suatu relasi dikatakan sudah memenuhi Bentuk Normal Kedua bila relasi tersebut sudah memenuhi bentuk Normal kesatu, dan atribut yang bukan key sudah tergantung penuh terhadap key-nya.

1. **Bentuk Normal Ketiga (3NF)**

Suatu relasi dikatakan sudah memenuhi Bentuk Normal ketiga bila relasi tersebut sudah memenuhi bentuk Normal kedua dan atribut yang bukan key tidak tergantung transitif terhadap key-nya.

## Sistem Basis Data

### Pengertian Basis Data

Menurut Pendidikan, (2021) Basis Data adalah kumpulan informasi yang disimpan dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari database. Software yang digunakan untuk mengelola dan permintaan panggilan (query) basis data yang disebut sistem manajemen database (database management system, DBMS).

Sistem database dipelajari dalam ilmu informasi. Istilah “database” berawal dari ilmu komputer. Meskipun maka itu berarti lebih luas, meletakkan segala sesuatu dalam luar bidang elektronika, artikel ini mengenai basis data komputer. Perhatikan bahwa mirip dengan database sebenarnya ada sebelum revolusi industri dalam bentuk buku besar, kuitansi dan kumpulan data terkait dengan bisnis. Konsep dasar dari database adalah kumpulan dari catatan, atau potongan dari pengetahuan.

Basis Data memiliki gambaran terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Benda Skema menggambarkan yang mewakili database, dan hubungan antara objek-objek tersebut. Ada banyak bagaimana mengatur skema, atau memodelkan struktur database: ini dikenal sebagai model basis data atau model data.

Model yang umum digunakan saat ini adalah model relasional, yang menurut istilah awam mewakili semua informasi dalam bentuk tabel yang saling berhubungan dengan setiap meja terdiri dari baris dan Kolom (definisi yang sebenarnya menggunakan terminologi matematika).

Dalam model ini, hubungan antar tabel tempat diwakili dengan menggunakan nilai yang sama antar tabel. Model-model lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

Database merujuk pada pengumpulan data yang saling berhubungan, dan perangkat lunak harus mengacu pada sistem manajemen database (database management system / DBMS). Jika konteksnya adalah jelas, banyak administrator dan programer menggunakan istilah basis data untuk kedua arti.

### Pengertian Basis Data Menurut Para Ahli

Berikut ini terdapat beberapa pengertian basis data menurut para ahli, terdiri atas:

1. S. Attre mengungkapkan bahwa database merupakan kumpulan data yang saling berhubungan antara yang satu dengan yang lain dengan penggunaan yang beragam.
2. Gordon C. Everest, database merupakan kumpulan data yang bersifat mekanis, terbagi dan terdefinisi dengan formal melalui suatu pengorganisasian.
3. Toni Fabbri, menyatakan bahwa database adalah sebuah sistem file terintegrasi yang memiliki setidaknya satu primary key sebagai pengulangan.
4. C. J. Date, database ialah data operasional yang digunakan oleh system aplikasi dari sebuah pengorganisasian.

### Komponen Basis Data

Berikut ini terdapat beberapa komponen basis data, terdiri atas :

1. **Perangkat Keras (Hardware)**

Sistem basis data dapat dilihat bahwa basis data pada intinya adalah disimpan pada media penyimpanan elektronik (hardisk), sedangkan database adalah terdiri dari beberapa file / table yang saling berelasi (berhubungan). Basis data tersebut dikelola oleh DBMS (database management system) dan database tersebut dapat dimanfaatkan oleh beberapa user (pemakai) yang dapat melakukan manipulasi pada database.

Perangkat keras yang biasanya terdapat dalam sistem basis data adalah sebagai berikut :

1. Komputer (satu untuk stand alone atau llebih dari satu untuk komputer jaringan)
2. Memori sekunder yang on-line (hardisk).
3. Memori sekunder yang offline (tape) untuk keperluan backup data
4. Media / perangkat komunikasi (untuk sistem jaringan)
5. **Sistem Operasi (Operating System)**

Sistem Operasi Merupakan program yang mengaktifkan / memungsikan sistem komputer, mengendalikan seluruh sistem daya dalam komputer dan melakukan operasioperasi dasar dalam computer (operasi input/output), pengelolaan file, dan lain sebagainya. Program pengelola basis data (DBMS) akan aktif (running) jika sistem operasi yang dikehendakinya telah aktif.

1. **Basis Data (Database)**

Sebuah sistem basis data dapat memiliki beberapa basis data. Setiap basis data dapat memiliki sejumlah objek basis data (seperti file/table, store procedure, indeks, dan lainya). Disamping berisi / menyimpan data, setiap basis data juga mengandung / menyimpan definisi struktur (baik untuk basis data maupun objek- objeknya secara detail).

1. **Sistem Pengelolaan Basis Data (Dbms)**

Sistem (aplikasi/perangkat lunak) pengelola basis data (DBMS) Pengelolaan basis data secara fisik tidak ditangani langsung oleh user (pemakai), tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus / spesifik. Perangkat inilah disebut DBMS, yang akan menentukan bagaimana data diorganisasi, disimpan, diubah, dan diambil kembali. Perangkat tersebut juga menerapkan mekanisme pengamanan data (security), pemakaian data secara bersama (sharing data), pemaksaan keakuratan / konsistensi data, dan sebagainya.

1. **Pemakai (User)**

Ada beberapa jenis / tipe pemakai pada sistem basis data, berdasarkan cara mereka berinteraksi pada basis data, diantaranya adalah:

1. **Programmer**

Programmer adalah pemakai yang berinteraksi dengan basis data Aplikasi melalui DML (data manipulation language), yang disertakan dalam program yang ditulis dalam bahasa pemrograman induk (seperti pascal, cobol, clipper, foxpro, dan lainnya).

1. **User Mahir (Casual Users)**

User Mahir (Casual Users) adalah pemakai yang berinteraksi dengan sistem tanpa menulis modul program. Mereka menyatakan query (untuk akses data), dengan bahasa query yang telah disediakan oleh suatu DBMS.

1. **User Umum**

User Umum adalah pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan satu program aplikasi permanen, yang telah ditulis / disediakan sebelumnya.

1. **User Khusus**

User Khusus adalah pemakai yang menulis aplikasi basis data non konvensional untuk keperluan khusus.

1. **Aplikasi (perangkat lunak) lain (bersifat optional)**

Aplikasi AI, Sistem Pakar, Pengolahan Citra, dan lainnyal, yang bisa saja mengakses basis data dengan / tanpa DBMS. Aplikasi (perangkat lunak) lain Aplikasi lain ini bersifat optional, ada tidaknya tergantung pada kebutuhan kita. DBMS yang kita gunakan lebih berperan dalam pengorganisasian data dalam basis data, sementara bagi pemakai basis data (khususnya yang menjadi end user) dapat disediakan program khusus untuk melakukan pengisian, pengubahan dan pengambilan data.

### Tujuan Basis Data

Berikut ini terdapat beberapa tujuan basis data, terdiri atas :

1. **Kecepatan dan Kemudahan (speed)**

Yakni agar pengguna basis data bisa :

1. Menyimpan Data
2. Melakukan Perubahan/Manipulasi Terhadap Data
3. Menampilkan Kembali Data Dengan Lebih Cepat Dan Mudah Dibandingkan Dengan Cara Biasa (Baik Manual Ataupun Elektronis).
4. **Efisiensi Ruang Penyimpanan (Space)**

Dengan basis data kita mampu melakukan penekanan jumlah redundansi (pengulangan) data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi antara kelompok data yang saling berhubungan.

Agar data sesuai dengan aturan dan batasan tertentu dengan cara memanfaatkan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (constraint) tipe data, domain data, keunikan data dsb.

1. **Ketersediaan (Availability)**

Agar data bisa diakses oleh setiap pengguna yang membutuhkan, dengan penerapan teknologi jaringan serta melakukan pemindahan/penghapusan data yang sudah tidak digunakan / kadaluwarsa untuk menghemat ruang penyimpanan.

1. **Kelengkapan (Completeness)**

Agar data yang dikelola senantiasa lengkap baik relatif terhadap kebutuhan pemakai maupun terhadap waktu, dengan melakukan penambahan baris- baris data ataupun melakukan perubahan struktur pada basis data; yakni dengan menambahkan field pada tabel atau menambah tabel baru.

1. **Keamanan (Security)**

Agar data yang bersifat rahasia atau proses yang vital tidak jatuh ke orang / pengguna yang tidak berhak, yakni dengan penggunaan account (username dan password) serta menerapkan pembedaan hak akses setiap pengguna terhadap data yang bisa dibaca atau proses yang bisa dilakukan.

1. **Kebersamaan (Sharability)**

Agar data yang dikelola oleh sistem mendukung lingkungan multiuser (banyak pemakai), dengan menjaga / menghindari munculnya problem baru seperti inkonsistensi data (karena terjadi perubahan data yang dilakukan oleh beberapa user dalam waktu yang bersamaan) atau kondisi deadlock (karena ada banyak pemakai yang saling menunggu untuk menggunakan data).

### Manfaat Basis Data

Berikut ini terdapat beberapa tujuan basis data, terdiri atas:

1. **Kecepatan dan Kemudahan**

Database memiliki kemampuan untuk memilih data sehingga menjadi kelompok diurutkan dengan cepat. Inilah yang ahirnya dapat menghasilkan informasi yang dibutuhkan dengan cepat pula. seberapa cepat diolah oleh database juga tergantung pada desain database.

1. **Dapat Digunakan Bersama**

Database dapat digunakan oleh siapa saja dalam sebuah perusahaan. Misalnya dalam database siswa perguruan tinggi diperlukan oleh beberapa bagian, seperti admin, keuangan, bagian akademik. Semua bidang ini memerlukan database mahasiswa, tetapi tidak perlu setiap bagian dibuat database itu sendiri, cukup dari database mahasiswa disimpan pada server pusat. Kemudian aplikasi masing-masing bagian dapat dihubungkan ke database siswa.

1. **Kontrol data terpusat**

Terkait dengan menunjuk ke dua, meskipun pada sebuah perusahaan memiliki banyak bagian atau divisi tapi database yang diperlukan tetap menjadi salah satu saja. Ini memfasilitasi data kontrol seperti ketika Anda ingin memperbarui data siswa, maka kita perlu memperbarui semua data dalam setiap bagian atau divisi, tapi cukup dalam satu database yang ada di server pusat.

1. **Perangkat hemat biaya**

Dengan memiliki database terpusat maka dalam setiap divisi tidak memerlukan perangkat untuk menyimpan database karena database hanya diperlukan satu yang disimpan di server pusat, ini akan memotong biaya pembelian perangkat.

1. **Keamanan Data**

Hampir semua sekarang memiliki aplikasi manajemen database fasilitas manajemen pengguna. Manajemen pengguna ini mampu menciptakan hak akses yang berbeda tergantung disesuaikan dengan kepentingan dan posisi pengguna. selain itu data yang disimpan dalam database diperlukan password untuk mengaksesnya.

1. **Memfasilitasi pembuatan Aplikasi baru**

Pada titik ini database dirancang dengan sangat baik, sehingga perusahaan membutuhkan aplikasi baru tidak perlu membuat database baru juga, atau tidak perlu mengubah struktur database yang sudah ada. Sehingga pengembang aplikasi atau programmer Si hanya cukup untuk membuat atau antarmuka aplikasi regulasi saja.

Dengan segudang manfaat dan kegunaan yang dimiliki oleh database maka seharusnya semua perusahaan yang baik Para pengusaha kecil terutama perusahaan besar memiliki database dibangun dengan desain yang baik. Ditambah dengan penggunaan teknologi jaringan komputer, manfaat dari database ini akan semakin besar.

Penggunaan database di teknologi jaringan komputer yang sama telah banyak digunakan oleh berbagai Perusahaan, misalnya, hanya bank-bank yang memiliki cabang di setiap kota. Bank Perusahaan hanya memiliki database yang disimpan pada server pusat, sedangkan cabang terhubung melalui jaringan komputer untuk mengakses database yang terletak di pusat.

### Jenis-Jenis Pemakaian Basis Data

Berikut ini terdapat beberapa jenis-jenis pemakaian basis data, terdiri atas:

1. **Programmer Aplikasi (PA)**

Adalah profesional komputer yang berinteraksi dengan sistem lewat DML yang dibuat dengan bahasa C, Cobol dan lainnya. Program program yang dibuat disebut sebagai program aplikasi, misalnya untuk perbankan, administrasi, akuntansi dan lain lain. Syntax DML berbeda dengan syntax bahasa komputer umumnya.

1. **Casual User (sepintas lalu, tidak tetap)**

Pemakai yang telah berpengalaman, berinteraksi dengan sistem tanpa menulis program, tetapi memakai bahasa query. Setiap Query akan mengajukan ke query processor yang mengambil dari perintah DML.

1. **Naive User**

Pemakai yang tidak berpengalaman, berinteraksi dengan sistem tanpa menulis program, tinggal menjalankan satu menu dan memilih proses yang telah ada atau telah dibuat sebelumnya oleh programmer.

1. **Specialized User**

Pemakai khusus yang menuliskan aplikasi database tidak dalam kerangka data processing yang tradisional. Aplikasi tersebut diantaranya adalah Computer Aided Design System, Knowledge Base, Expert System, sistem yang menyimpan data dalam bentuk data yang komplek misalnya data grafik, data audio.

1. **Optional Software**

Perangkat lunak pelengkap yang mendukung. Bersifat opsional.

### Tahapan Desain Basis Data

Desain database merupakan upaya untuk membangun database dalam lingkungan bisnis. untuk membangun database ada langkah-langkah Sebagai berikut :

1. Database perencanaan
2. Mendefinisikan sistem
3. Analisis dan kebutuhan mengumpulkan
4. Desain database
5. Aplikasi desain
6. Membuat prototipe
7. Pelaksanaan
8. Konversi data
9. Pemeriksaan
10. Pemeliharaan operasional

Dilihat dari jenisnya, database dibagi menjadi 2, yaitu :

1. **Database Flat-File**

Database flat-file sangat ideal untuk data kecil dan dapat diubah dengan mudah. Pada dasarnya, mereka terdiri dari satu set string dalam satu atau lebih file yang bisa diurai untuk informasi yang disimpan. Database datar-file yang digunakan untuk menyimpan daftar atau data sederhana dan dalam jumlah kecil.

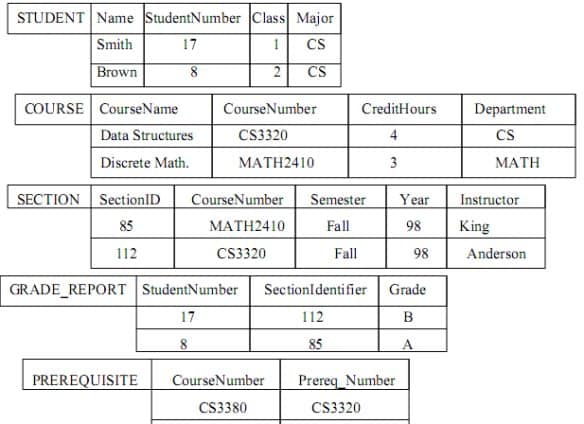
Database flat-file akan menjadi sangat rumit ketika digunakan untuk menyimpan data dengan struktur yang kompleks meskipun juga memungkinkan untuk menyimpan data tersebut. salah satu masalah dengan menggunakan jenis data base rentan terhadap korupsi data akibat tidak adanya penguncian terpasang bila data yang digunakan atau dimodifikasi.

1. **Database Relasional**

Database ini memiliki lebih logis struktur terkait sarana penyimpanan. kata “relasional” berasal dari fakta bahwa tabel dalam database dapat dihubungkan ke salah satu lainnya. Database relasional menggunakan sekumpulan tabel dua dimensi, setiap tabel terdiri dari baris (tuple) dan kolom (atribut). Untuk menciptakan hubungan antara dua atau lebih tabel, gunakan kunci (atribut key) yang merupakan kunci utama dalam satu meja dan kunci asing di meja lain.

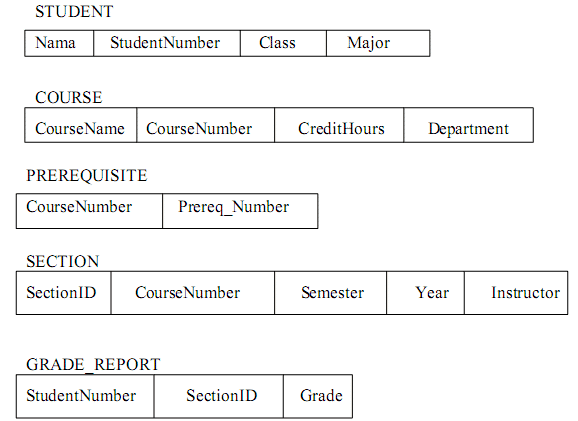
Saat ini, database relasional menjadi pilihan karena keunggulannya. Beberapa kelemahan yang mungkin dirasakan ke database Tipe ini lebih sulit untuk pelaksanaan sejumlah besar data dengan tingkat kerumitan tinggi dan proses pencarian informasi lebih lambat karena kebutuhan untuk menghubungkan tabel di muka maju jika data tersebar di beberapa tabel.

### Contoh Basis Data

[](https://www.dosenpendidikan.co.id/wp-content/uploads/2019/09/Contoh-Basis-Data.jpg)

Gambar 1. Basis Data

Dapat dinyatakan dalam bentuk skema :

[](https://www.dosenpendidikan.co.id/wp-content/uploads/2019/09/Basis-Data-Bentuk-Skema.png)

Gambar 2. Basis Data Bentuk Skema

### Normalisasi

Menurut Tanoto, (2021) Normalisasi merupakan sebuah teknik logical desain dalam sebuah basis data yang mengelompokkan atribut dari berbagai entitas dalam suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redudansi/pengulangan data) serta sebagian besar ambiguity bisa dihilangkan.

Atau pengertian singkatny, Normalisasi Databse adalah proses pengelompokan atribut data yang membentuk entitas sederhana, nonredundan, fleksibel, dan mudah beradaptasi, Sehingga dapat dipastikan bahwa database yang dibuat berkualitas baik.

Normalisasi database terdiri dari banyak bentuk, dalam ilmu basis data ada setidaknya 9 bentuk normalisasi yang ada yaitu 1NF, 2NF, 3NF, EKNF, BCNF, 4NF, 5NF, DKNF, dan 6NF.

Database 1NF, 2NF, dan 3NF akan sering ditemui ketika akan membuat sebuah database yang optimal. Jika Anda ingin menjadi seorang Database Administrator (DBA), harus tahu bagaimana cara normalisasi database yang optimal. Misalkan suatu saat ketika website yang Anda buat mengalami penurunan kinerja, mungkin Anda akan ditanya apakah database tersebut sudah dinormalisasi dengan benar.

### Tujuan Normalisasi Database

Tujuannya adalah untuk menghilangkan dan mengurangi redudansi data dan tujuan yang kedua adalah memastikan dependensi data (Data berada pada tabel yang tepat). Jika data dalam database tersebut belum di normalisasi maka akan terjadi 3 kemungkinan yang akan merugikan sistem secara keseluruhan.

1. INSERT Anomali : Situasi dimana tidak memungkinkan memasukkan beberapa jenis data secara langsung di database.
2. DELETE Anomali: Penghapusan data yang tidak sesuai dengan yang diharapkan, artinya data yang harusnya tidak terhapus mungkin ikut terhapus.
3. UPDATE Anomali: Situasi dimana nilai yang diubah menyebabkan inkonsistensi database, dalam artian data yang diubah tidak sesuai dengan yang diperintahkan atau yang diinginkan.

### Database Seperti Apa yang dinormalisasi ?

Tidak semua database bisa dinormalisasi, hanya tipe “relational database“ yang bisa dinormalisasi. Banyak vendor DBMS (Database Management System) diantaranya Oracle, MySQL, SQL Server, PostgreSQL, dll.

### Tahapan Normalisasi Database

Untuk melakukan normalisasi database kita harus mengidentifikasi data seperti apa yang akan disimpan, dan berikut adalah contohnya :

1. **Bentuk Tidak Normal (unnormalize)**

Bentuk tidak normal (unnormalized) merupakan kumpulan data yang direkam tidak ada keharusan dengan mengikuti suatu format tertentu.

Pada bentuk tidak normal terdapat repeating group (Pengulangan Group), sehingga pada kondisi ini data menjadi permasalahan dalam melakukan manipulasi data (insert, update, dan delete) atau biasa disebut anomali.



Gambar 3. Bentuk Tidak Normal (unnormalize)

1. **1NF / First Normal Form**

1NF mensyaratkan beberapa kondisi dalam sebuah database, berikut adalah fungsi dari bentuk normal pertama ini.

1. Menghilangkan duplikasi kolom dari tabel yang sama.
2. Buat tabel terpisah untuk masing-masing kelompok data terkait dan mengidentifikasi setiap baris dengan kolom yang unik (primary key).



Gambar 4. 1NF / First Normal Form

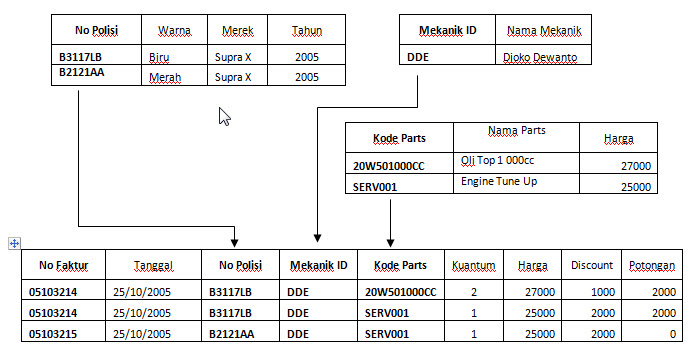
Pada intinya bentuk normalisasi 1NF ini mengelompokkan beberapa tipe data atau kelompok data yang sejenis agar dapat dipisahkan sehingga anomali data dapat di atasi.

Contoh adalah ketika kita ingin menghapus, mengupdate, atau menambahkan data peminjam, maka kita tidak bersinggungan dengan data buku atau data penerbit. Sehingga inkonsistensi data dapat mulai di jaga.

1. **2NF**

Syarat untuk menerapkan normalisasi bentuk kedua ini adalah data telah dibentuk dalam 1NF, berikut adalah beberapa fungsi normalisasi 2NF.

1. Menghapus beberapa subset data yang ada pada tabel dan menempatkan mereka pada tabel terpisah.
2. Menciptakan hubungan antara tabel baru dan tabel lama dengan menciptakan foreign key.
3. Tidak ada atribut dalam tabel yang secara fungsional bergantung pada candidate key tabel tersebut.



Gambar 5. 2NF

1. **3NF**

Pada 3NF tidak diperkenankan adanya partial “transitive dependency“ dalam sebuah tabel. Transitive dependency biasanya terjadi pada tabel hasil relasi, atau kondisi dimana terdapat tiga atribut A, B, C. Kondisinya adalah A ⇒ B dan B ⇒ C. Maka C dikatakan sebagai transitive dependency terhadap A melalui B.

Intinya pada 3NF ini, jika terdapat suatu atribut yang tidak bergantung pada primary key tapi bergantung pada field yang lain maka atribut-atribut tersebut perlu dipisah ke tabel baru.

Contohnya ada pada atribut **qty**, kolom tersebut tidak bergantung langsung pada primary key **kode\_faktur** melainkan bergantung pada kolom **kode\_barang**. Jadi setelah dinormalisasi 3NF akan menghasilkan tabel berikut:



Gambar 6. 3NF

Dari gambar tabel di atas dapat dilihat pada tahap normalisasi 3NF menghasilkan 1 tabel baru dari hasil pemecahan **tabel transaksi** yaitu **tabel detail barang** yang isinya menampung barang-barang yang dibeli.

Kolom **harga** pada **tabel detail** **barang** digunakan untuk menyimpan harga barang pada saat proses transaksi. Jadi, meskipun kolom **harga** pada **tabel barang** berubah (naik/turun), harga barang yang ada pada **tabel** **detail barang** tidak ikut berubah (fixed). Bayangkan jika kita tidak menambahkan kolom **harga** pada **pada tabel detail barang,** maka yang terjadi total invoice dari transaksi akan berubah seiring berubahnya harga barang.

1. **BCNF Boyce–Codd normal form**

Merupakan sebuah teknik normalisasi database yang sering disebut 3.5NF, memiliki hubungan yang sangat erat dengan bentuk 3NF. Pada dasarnya adalah untuk menghandle anomali dan overlooping yang tidak dapat di handle dalam bentuk 3NF. Normalisasi database bentuk ini tergantung dari kasus yang disediakan, tidak semua tabel wajib di normalisasi dalam bentuk BCNF.

Untuk tabel untuk memenuhi Bentuk Normal Boyce-Codd, harus memenuhi dua kondisi berikut :

1. yaitu Table harus dalam Bentuk Normal Ketiga.
2. Dan, untuk ketergantungan apa pun A → B, A harus menjadi super key.

Poin kedua terdengar agak rumit, kan? Dengan kata sederhana, itu berarti, bahwa untuk ketergantungan A → B, A tidak dapat menjadi atribut non-prima, jika B adalah atribut utama.

### Pentingnya Normalisasi

Suatu rancangan database disebut buruk jika :

1. Data yang sama tersimpan di beberapa tempat (file atau record).
2. Ketidakmampuan untuk menghasilkan informasi tertentu.
3. Terjadi kehilangan informasi.
4. Terjadi adanya redudansi (pengulangan) atau duplikasi data sehingga memboroskan ruang penyimpanan dan menyulitkan saat proses updating data.
5. Timbul adanya NULL VALUE..
6. Kehilangan informasi bisa terjadi bila pada waktu merancang database (melakukan proses dekomposisi yang keliru).
7. Bentuk normalisasi yang sering digunakan adalah 1st NF, 2nd NF, 3rd NF,dan BCNF.

### ERD (Entity Relationship Diagram)

### Pengertian ERD (Entity Relationship Diagram)

Menurut Sulthon, (2021) ERD adalah pemodelan data atau sistem dalam database yang sudah sering digunakan oleh banyak lembaga. Fungsinya ERD adalah untuk memodelkan struktur dan hubungan antar data yang relatif kompleks. Keberadaan sistem ERD sangat penting untuk perusahaan dalam mengelola data yang dimilikinya.

Bentuknya seperti diagram yang menjelaskan hubungan antar objek data. Untuk menggambarkannya dibutuhkan :

1. Notasi ialah seperangkat lambang yang menggambarkan data
2. Simbol sebagai lambang sebagai penanda
3. Bagan merupakan rancangan atau skema untuk mempermudah penafsiran
4. dan lain sebagainya.

### Komponen Umum Penyusun ERD

Untuk membuat basis data ERD dibutuhkan tiga komponen utama sebagai penyusunnya atau bisa juga disebut sebagai notasi.

1. **Entitas (entity)**

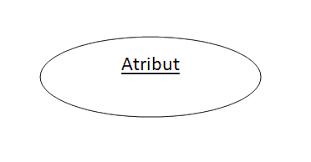
****

Gambar 7. Entitas (entity)

Entitas adalah sebuah objek berwujud nyata yang dapat dibedakan dengan objek lainnya. Objeknya dapat bersifat konkret maupun abstrak. Data konkret adalah sesuatu yang benar-benar ada atau dapat dirasakan oleh alat indra, sedangkan abstrak tidak berwujud.

Orang, buku, pegawai,perusahaan merupakan jenis entitas konkret. Berbeda dengan mata kuliah, kejadian, pekerjaan adalah benda tak berwujud.

1. **Atribut (field)**



Gambar 8. Atribut (field)

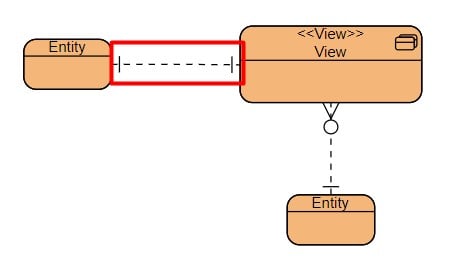
Pengertian ERD kedua yaitu field atau disebut sebagai atribut. Setiap entitas memiliki atribut untuk mendeskripsikan karakteristik dari suatu entitas. Untuk jenisnya dibedakan menjadi beberapa jenis, yaitu

Atribut key, atribut yang unik dan berbeda. Misalnya, Nomor pokok mahasiswa (NPM), NIM dan nomor pokok lainnya.

Atribut Composite, atribut yang terdiri dari beberapa sub atribut yang memiliki arti tertentu. Contohnya, nama lengkap yang dipecah menjadi nama depan, tengah, dan belakang.

Dan atribut deviratif, yang dihasilkan dari atribut atau relasi lain. Jenis atribut ini tidak wajib ditulis dalam diagram ER atau pun disimpan dalam database. Sebagai contoh deriative attribute adalah usia, kelas, selisih harga, dan lain-lain.

1. **Relasi (relation)**



Gambar 9. Relasi (relation)

Selanjutnya ada relasi, hubungan antar entitas untuk menunjukkan adanya koneksi di antara sejumlah entitas yang berasal dari himpunan entitas berbeda. Misalnya, dalam hubungan entitas sistem akademik antara mahasiswa dan mata kuliah adalah “mengambil”.  Mahasiswa mengambil mata kuliah.

Dalam ERD terdapat kardinalitas relasi atau rasio kardinalitas untuk memetakan bagaimana data berhubungan satu sama lain yang terbagi menjadi empat, yaitu:

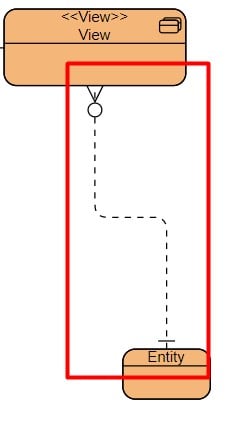
Pengertian ERD Pertama, One to One (1:1). Apa maksud dari satu ke satu ini? Misalnya terdapat entitas A dan B. Setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B, begitu pun sebaliknya. Jadi, setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B saja. Contohnya, satu siswa (1) memiliki satu nomor siswa (1), dan sebaliknya.

Pengertian ERD Kedua, One to many (1:M). Satu ke Banyak ini maksudnya adalah setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B. Dengan kata lain, setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B. Akan tetapi, tidak sebaliknya. Contoh dari relasi One to Many ini adalah satu kelas (1) berisi banyak siswa (M), atau siswa mengikuti banyak ekstrakurikuler.

Pengertian ERD Ketiga, Many to One (M:1). Relasi ini merupakan kebalikan dari relasi sebelumnya. Untuk contohnya, yaitu banyak pegawai (M) bekerja dalam satu departemen (1), atau banyak dosen mengajar dalam satu mata kuliah.

Pengertian ERD Keempat, Many to Many (M:N). Setiap entity pada kumpulan entitas A dapat berhubungan dengan banyak entitas pada kumpulan data entitas B. Misalnya, banyak siswa (M) mempelajari banyak pelajaran (N). Demikian pula sebaliknya, banyak pelajaran (N) dipelajari banyak siswa (M).

1. **Garis**



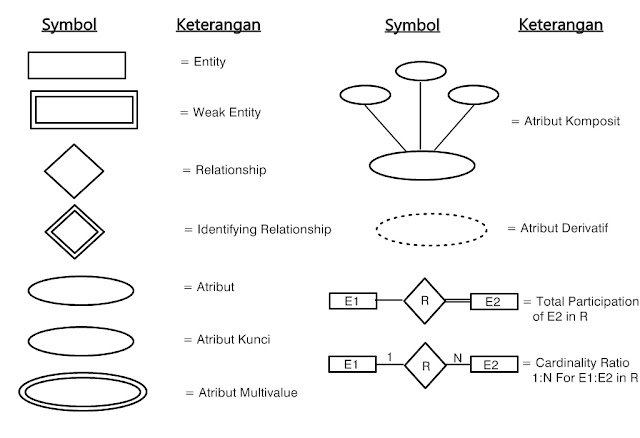
Gambar 10. Garis

Fungsi dari garis ini tidak hanya sebatas penghubung antar himpunan relasi dengan himpunan entitas, serta himpunan entitas dengan atributnya. Garis dapat mempermudah pengguna untuk melihat dan mengetahui alur sebuah ERD sehingga nampak jelas awal dan akhirnya.

### Simbol Notasi dalam ERD

Pengertian ERD berbentuk ER Diagram maka diperlukan simbol dan konektor yang berbeda untuk memvisualiasaikan informasi penting yang digunakan dalam desain database.

Entitas digambarkan dengan persegi panjang. Attribut diwakili dengan elips atau oval. Untuk jenis atribut lainnya seperti key attribute ditandai dengan garis di dalamnya.



Gambar 11. Simbol Notasi dalam ERD

Beda halnya dengan composite attribute yang digambarkan dengan lingkaran ditambah lingkaran lainnya yang dihubungkan dengan garis. Hal tersebut menandakan bahwa jenis entitas ini terdiri dari beberapa atribut kecil. Dan jenis derivative disimbolkan dengan lingkaran bergaris putus-putus.

Untuk menggambarkan relasi, biasanya simbol yang digunakan berbentuk diamond atau belah ketupat.

### Tahapan Umum Pembuatan ERD

Setelah memahami entitas, atribut, dan kardinalitas relasi, saatnya melakukan langkah-langkah berikut jika ingin membuat model ERD.

1. **Tahap 1 Menentukan entitas yang akan terlibat atau menentukan tabel.**

Contoh masalah berasal dari SIAKAD Perkuliahan. Dalam sistem SIAKAD suatu perguruan tinggi, mahasiswa dapat melakukan input KRS, melihat KHS, mengambil banyak mata kuliah, dan mata kuliah diambil oleh para mahasiswa.

Sebagai contoh, untuk penentuan entitasnya bisa dijadikan sebagai Mahasiswa, Dosen, Ruang, dan Mata\_Kuliah.

1. **Tahap 2 Menentukan atribut-atribut key dari masing-masing himpunan entitas.**

Di tahap penentuan atributnya maka akan menjadi seperti ini:

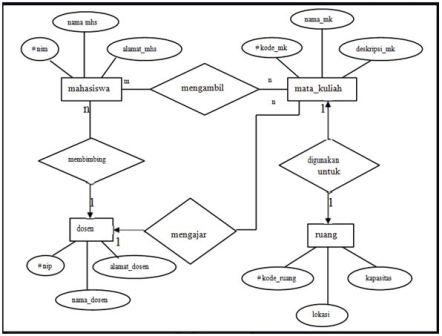
1. Mahasiswa: nim, nama\_mhs, almt\_mhs
2. Dosen: nip, nama\_dsn, almt\_dsn
3. Mata kuliah: kd\_mk, nama\_mk
4. Ruang: kd\_ruang, lokasi, kapasitas
5. **Tahap 3 Menetapkan seluruh himpunan relasi di antara himpunan entitas yang ada beserta foreign key-nya dan kardinalitas relasi.**

Setelah menentukan atribut-atributnya, maka langkah selanjutnya menetukan relasi. Inilah contoh relasi beserta kardinalitasnya.

1. Mahasiswa (M) mengambil Mata\_Kuliah (N), yaitu banyak mahasiswa mengambil banyak mata kuliah.
2. Dosen (1) membimbing Mahasiswa (N), yaitu satu dosen membimbing banyak mahasiswa.
3. Dosen (1) mengajar Mata\_Kuliah (N), yaitu satu dosen mengajar banyak mata kuliah.
4. Ruang (1) digunakan untuk Mata\_Kuliah (1), yaitu satu ruang dapat digunakan untuk satu mata kuliah saja.
5. **Tahap 4 Membuat model ERD.**

Setelah menetapkan hubungan dan rasio kardinalitas, selanjutnya membuat diagram ER. Dalam tahap ini hanya menggabungkan seluruh himpunan dan relasi dari tahap ketiga.

Dimulai dari mahasiswa. Mahasiswa mengambil Mata\_Kuliah, Dosen membimbing Mahasiswa, Dosen mengajar Mata\_Kuliah, dan Ruang untuk Mata\_Kuliah.



Gambar 12. Tahap 4 Membuat model ERD.

Selanjutnya menentukan foreig key, primary key yang termasuk pada atribut entitas lain. Di antara entitas Mahasiswa mengambil Mata\_Kuliah maka kunci utamanya terletak pada relasi mengambil, yaitu nim dan kd\_mk.  Sedangkan untuk foreign key Dosen membimbing Mahasiswa terletak pada nim.

Dalam SIAKAD, mahasiswa juga dapat melihat KHS atau kumpulan nilai selama belajar. Atribut Nilai tidak diletakan pada entitas, namun pada relasi Mengambil.

Jika atribut ini ditempatkan pada entitas Mahasiswa, maka semua mata kuliah yang diambil oleh satu Mahasiswa menghasilkan nilai yang sama. Jika aribut Nilai ditempatkan pada entitas Mata\_Kuliah, maka semua mahasiswa yang mengambil Mata\_Kuliah itu akan memiliki nilai yang sama. Keduanya sama-sama tidak realistis.

Maka dengan adanya atribut Nilai pada relasi Mengambil, maka seorang mahasiswa tertentu yang mengambil mata kuliah tertentu akan mendapatkan nilai yang tertentu pula.

Sebenarnya relasi tidaklah memiliki atribut. Namun, jika diantara dua entitas relasinya Many to Many, maka akan membentuk entitas baru. Nantinya di dalam entitas baru tersebut terdapat primary key dari entitas Mahasiswa dan Mata\_Kuliah untuk dijadikan foreign key pada entitas baru.

Entitas baru tersebut bernama Mhs\_ambil\_MK dengan atribut password, nim, kd\_mk, dan nilai. Dari semua atributnya, password akademiklah yang dijadikan sebagai foreign keynya. Karena satu mahasiswa memiliki password yang berbeda dengan mahasiswa lain. Maka hasil ERD akan seperti berikut.

Mahasiswa mengambil Mhs\_ambil\_MK diambil Mata\_Kuliah, kemudian disambungkan dengan Ruang dengan relasi digunakan untuk. Sambungkan entitas Dosen membimbing Mahasiswa dan Dosen mengajar Mata\_Kuliah. Yang berbeda dari bentuk ERD sebelumnya hanya terletak pada “Mhs\_ambil\_MK” saja agar nilai mata kuliah mahasiswa menjadi realistis.

### Tips Membuat ERD

Agar efektif dan mudah dipahami, ikuti beberapa tips ini saat membuat basis data ER diagram.

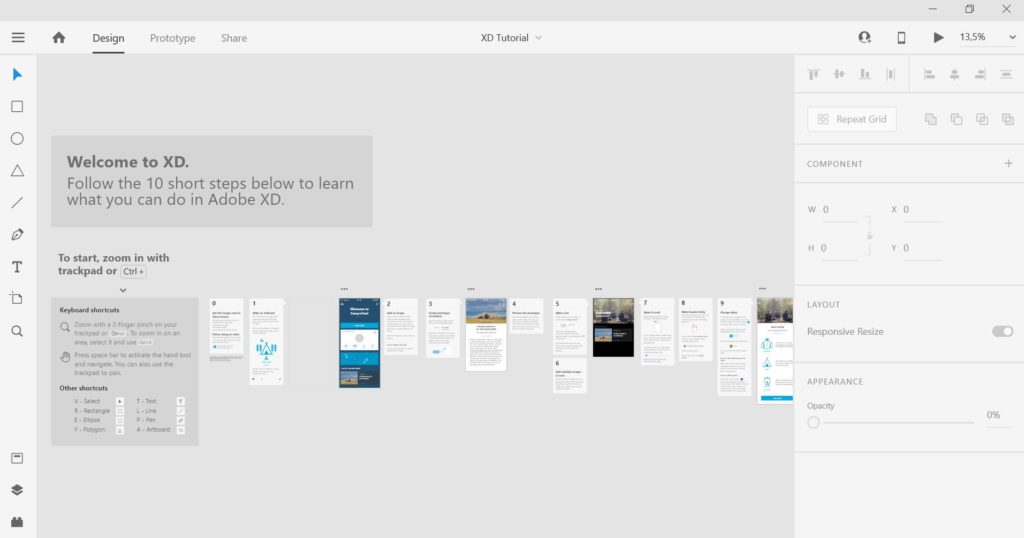
1. Pastikan nama entitas hanya muncul sekali dalam setiap model ERD.
2. Berikan nama untuk setiap entitas, atribut, dan relasi.
3. Cek kembali relasi antar entitas, apakah semuanya benar-benar dibutuhkan atau tidak. Atau malah ada relasi yang terlewatkan. Kemudian, jangan gabungkan relasi dengan satu sama lain dan buang relasi yang digunakan berulang.
4. Gunakan warna yang berbeda, misalnya warna terang untuk menandai ERD paling penting.

## Adobe XD

### Pengertian Adobe XD

Menurut Nastainullah, (2020) Adobe XD adalah sebuah alat yang disediakan gratis oleh Adobe untuk desain UI / UX dan prototyping berbagai platform termasuk web, ponsel, tablet, dan lainnya. Hal pertama yang akan Anda perhatikan ketika membuat aplikasi adalah start screen. Sebagai pengguna baru Adobe XD, saya sangat menyarankan mengklik tombol Begin Tutorial. Dimana Anda akan dibawa ke layar dengan panel yang menjelaskan proses menggunakan Adobe XD.

Informasi dengan cepat memandu Anda melalui dasar-dasar aplikasi. Anda akan belajar bagaimana menyelesaikan tugas-tugas penting seperti membuat artboards, mengimport gambar, menggunakan repeat grid dan bagaimana menghubungkan artboards bersama-sama untuk pembuatan prototipe. Jika Anda adalah pengguna Creative Cloud, banyak elemen yang terasa familiar.



Gambar 13. Adobe XD

Setelah Anda selesai meninjau proyek tutorial, sekarang saatnya untuk membuat yang baru. Anda dapat melakukan ini dengan memilih File> Baru. Ini akan mengembalikan Anda ke layar yang mirip dengan Layar Selamat Datang. Di sini Anda dapat dengan cepat memilih resolusi layar target berdasarkan perangkat, ukuran web, atau memasukkan nilai untuk ukuran khusus.

Setelah memilih ukuran untuk proyek Anda, Anda akan memasuki antarmuka Adobe XD. Anda akan segera melihat bahwa antarmuka dipecah menjadi beberapa tab: Desain dan Prototipe. Di dalam tab Desain Anda akan membangun semua elemen Anda pada artboards, dengan masing-masing artboard mewakili layar yang berbeda. Adobe XD juga menawarkan beberapa alat menggambar untuk membuat bentuk dasar atau Anda dapat mengimpor karya seni dari luar aplikasi desain.

### Fungsi Adobe XD

Dari pengertian yang saya jelaskan diatas, sudah jelas fungsi utama Adobe XD memiliki fungsi untuk mengerjakan desain aplikasi mobile dan web.

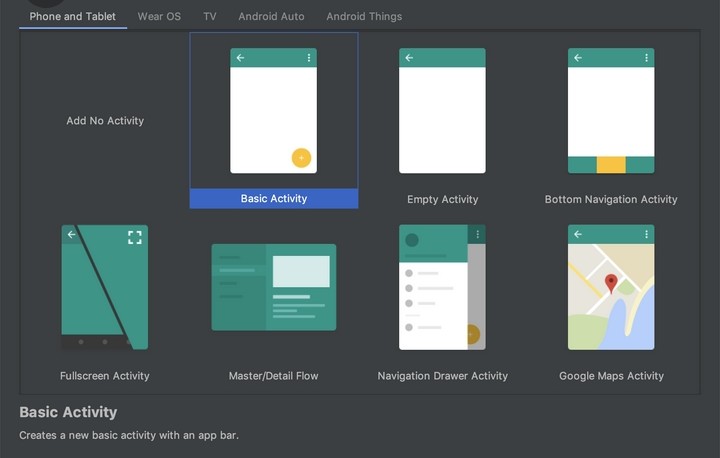
## Android Studio

### Pengertian Android Studio

Menurut Fadilah, (2019) Android Studio adalah perangkat lunak (software) resmi dari Google untuk para pengembang aplikasi Android. Dengan Android studio, para pengembang dapat membuat aplikasi untuk Android hanya menggunakan satu perangkat lunak saja.

Di Android Studio sudah tersedia beragam fitur yang memudahkan para pengembang, antara lain :

1. Android emulator
2. Editor teks
3. Cloud test lab integration
4. GPU debugger preview
5. Dan masih banyak lagi



Gambar 14. Android Studio

Karena sudah memiliki fitur yang lengkap, Android Studio termasuk software IDE atau Integrated Development Environment. Atau dalam bahasa Indonesia berarti Lingkungan Pengembangan Terintegrasi.

Kita dapat memulai proyek aplikasi baru, mengedit, kemudian menampilkannya langsung dari Android Studio.

Jadi sangat simpel sekali, hanya dengan satu software kita bisa mengembangkan aplikasi Android secara keseluruhan.

### Fungsi Android Studio

Dari pengertian yang saya ujelaskan diatas, jelas fungsi utama Android studio adalah untuk mengembangkan aplikasi Android.

Namun selain itu, Android juga memiliki banyak fitur lain, diantaranya :

1. Mengembangkan aplikasi Android
2. Mengedit kode sumber aplikasi Android
3. Mengembangkan game sederhana
4. Dan masih banyak lagi

Namun intinya fungsi utama Android Studio adalah untuk mengembangkan aplikasi Android.

## Waterfall (Air Terjun)

Menurut Rosa dan Shalahuddin (2013:28) dalam jurnal Dermawan and Hartini, (2017) Model SDLC Waterfall (Air Terjun) sering juga disebut model sekuensial linier (sequential linear) atau alur hidup klasik (classic life cycle). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (support). Berikut adalah gambar model air terjun :



Gambar 15. Waterfall (Air Terjun)

1. Analisis Kebutuhan Perangkat Lunak (Analysis), Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.
2. Pembuatan Desain (Design), Desain perangkat lunak adalah proses multi, Langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.
3. Implementasi (Koding), Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.
4. Pengujian (Testing), Pengujian fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (error) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. 5. Pendukung (Support) atau Pemeliharaan
5. Perawatan (Maintenance), Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

**PERANCANGAN SISTEM**

## Waktu dan Tempat Penelitian

### Waktu Penelitian

Waktu Penelitian ini dilakukan pada bulan November - bulan Desember 2021.

### Tempat Penelitian

Tempat Penelitian dilakukan di rumah saya disebelah rumahnya spongebob Jln. Tuna Sirip Kuning Kabupaten Bikini Bottom .

## Metode Penelitian

### Metode Waterfall (Air Terjun)

Berikut dilembaran selanjutnya tabel untuk penerapan metode waterfall pada aplikasi input nilai :

|  |  |  |
| --- | --- | --- |
| **No** | **Tahapan** | **Uraian** |
| 1 | Alasan menggunakan waterfall | Karena kebutuhan dari dosen untuk metode penelitiannya menggunakan metode Waterfall (Air Terjun) |
| 2 | Analisis | Analisis dilakukan dengan cara mewawancarai. Dari wawancara didapatkan data-data seputar nilai yang akan kita input. |
| 3 | Desain | Perancangan aplikasi ini menggunakan Adobe XD |
| 4 | Implementasi | Aplikasinya akan dibuat menggunakan bahasa pemrograman java berbasis Android Studio. |
| 5 | Pengujian | Pengujian dilakukan pada aspek fungsionalitas kepada ahli sistem informasi. |
| 6 | Perawatan | Perawatan akan dilakukan apabila ada update fitur atau memperbaiki kesalahan yang ditemukan pada saat aplikasi digunakan. |

Tabel 1. Metode Waterfall (Air Terjun)

## Teknik Pengumpulan Data

Teknik yang digunakan untuk pengumpulan data adalah sebagai berikut :

1. Observasi, adalah pengamatan langsung terhadap suatu kegiatan yang sedang dilakukan. Di sini penulis melakukan pengamatan secara langsung sehingga dapat mengetahui proses kegiatan yang berlangsung.
2. Interview (*Wawancara*), Wawancara memungkinkan analis sistem sebagai pewawancara untuk mengumpulkan data secara tatap muka langsung dengan orang yang diwawancarai yang di anggap mempunyai wewenang dan kemampuan dalam memberikan informasi atau keterangan yang diperlukan yang berhubungan dengan objek yang di teliti.
3. Dokumentasi adalah metode yang lebih mudah dilakukan metode-metode lain karena jika ada kekeliruan, sumber datanya masih tetap. Objek yang diamati pada metode dokumentasi kesalahan benda hidup melainkan benda mati.

## Analisa Sistem Berjalan

Dalam proses ini aplikasi ini memasukkan (*input*) nama mahasiswa dan nilai angka yang nantinya sistemnya akan menghitung nilainya. Setelah itu akan dihitung untuk mendapatkan hasil nilai huruf dan keterangan.

## Rancangan Sistem Yang Diusulkan

## Instrumen Penelitian

### Perangkat Keras (Hardware)

Perangkat Keras (*Hardware*) yang digunakan pada penelitian ini adalah :

1. Laptop / Komputer
2. RAM 2 GB
3. HDD 500 GB
4. Processor Intel N2810
5. Smartphone
6. RAM 2 GB
7. HDD 32 GB
8. Processor MediaTek MT6765G Helio G35 (12 nm)

### Perangkat Lunak

Perangkat Lunak (*Software*) yang digunakan pada penelitian ini adalah :

1. Sistem Operasi Windows 10
2. Sistem Operasi Android
3. Adobe XD
4. Android Studio

## Jadwal Penelitian

Jadwal Penelitian di rencanakan selama tiga bulan yang berawal dari minggu pertama bulan September hingga bulan Desember tahun 2021, berikut rincihan pembagian waktu penelitian yang disajikan dalam tabel dibawah ini :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NO. | Aktifitas  Penelitian | Bulan | | | | | | | | | | | |
| September | | | | November | | | | Desember | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1. | Analisis |  |  |  |  |  |  |  |  |  |  |  |  |
| 2. | Desain |  |  |  |  |  |  |  |  |  |  |  |  |
| 3. | Koding |  |  |  |  |  |  |  |  |  |  |  |  |
| 4. | Ujicoba |  |  |  |  |  |  |  |  |  |  |  |  |
| 5. | Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |
| 6. | Laporan |  |  |  |  |  |  |  |  |  |  |  |  |

Tabel 2. Jadwal Penelitian

DAFTAR PUSTAKA

Asmy, H. (2016) *Alat Bantu Dalam Perancangan Sistem Informasi*, *hidayatullahasmyy.blogspot.com*. Available at: http://hidayatullahasmyy.blogspot.com/2016/04/alat-bantu-dalam-perancangan-sistem.html (Accessed: 26 November 2021).

Dermawan, J. and Hartini, S. (2017) ‘IMPLEMENTASI MODEL WATERFALL PADA PENGEMBANGAN SISTEM INFORMASI PERHITUNGAN NILAI MATA PELAJARAN BERBASIS WEB PADA SEKOLAH DASAR AL-AZHAR SYIFA BUDI JATIBENING’, *Jurnal Paradigma*, 19(2), p. 37. doi: 10.1093/nq/s5-VII.159.37-a.

Fadilah, L. (2019) *Pengertian Android Studio, Fungsi dan Cara Menggunakannya*, *androbuntu.com*. Available at: https://androbuntu.com/2019/11/05/android-studio/ (Accessed: 26 November 2020).

Nastainullah, R. (2020) *Apa Itu Adobe XD dan Fiturnya*, *blogs.masterweb.com*. Available at: https://blogs.masterweb.com/apa-itu-adobe-xd/#Kesimpulan (Accessed: 26 November 2021).

Pendidikan, D. 3 (2021) *Basis Data adalah*, *dosenpendidikan.co.id*. Available at: https://www.dosenpendidikan.co.id/basis-data/ (Accessed: 26 November 2021).

Sulthon, A. (2021) *Pengertian ERD Adalah: Contoh, Simbol, dan Penjelasan*, *DomaiNesia*. Available at: https://www.domainesia.com/berita/pengertian-erd-adalah/#Pengertian\_ERD (Accessed: 26 November 2021).

Tanoto, U. (2021) *Normalisasi Database: Pengertian, Tujuan dan Cara Melakukannya*, *jojonomic.com*. Available at: https://www.jojonomic.com/blog/normalisasi-database/ (Accessed: 26 November 2021).