

# **MAKALAH**

## **BASIS DATA**



DISUSUN OLEH :

NAMA : ZIKRILLAH SADDAM FATHURRAHMAN

NIS : 181901945

KELAS : XI RPL 2

JURUSAN : Rekayasa Perangkat Lunak

SMK NEGERI 4 KOTA BOGOR

Jl. Raya Tajur Biotrop kp. Buntar Rt 02/06 Bogor Selatan

TAHUN PELAJARAN 2019/2020

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>DAFTAR ISI .....</b>	<b>ii</b>
<b>KATA PENGANTAR.....</b>	<b>iii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	1
1.3 Tujuan Dan Manfaat .....	1
<b>BAB 2 PEMBAHASAN.....</b>	<b>2</b>
2.1 Struktur Hirarki Basis Data .....	2
2.2 Diagram Hubungan Entitas .....	4
2.3 Normalisasi Basis Data .....	7
2.4 Kelompok Perintah DDL, DML, DCL.....	10
2.5 Tipe Data Dalam Basis Data .....	11
2.6 Record, Table, Field.....	12
2.7 Bahasa SQL Dasar Untuk Mengelola Tabel .....	13
2.8 Bahasa SQL Untuk Memanipulasi Tabel.....	13
2.9 Fungsi Agregasi Dalam Basis Data.....	14
<b>BAB 3 STUDI KASUS.....</b>	<b>15</b>
<b>BAB 4 KESIMPULAN, PENUTUP, DAFTAR PUSTAKA .....</b>	<b>17</b>
4.1 Kesimpulan.....	17
4.2 Penutup.....	17
4.3 Daftar Pustaka.....	18

## KATA PENGANTAR

Puji syukur alhamdulillah kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena telah melimpahkan rahmat-Nya berupa kesempatan dan pengetahuan sehingga makalah ini bisa selesai pada waktunya.

Terima kasih juga kami ucapkan kepada teman-teman yang telah berkontribusi dengan memberikan ide-idenya sehingga makalah ini bisa disusun dengan baik dan rapi.

Kami berharap semoga makalah ini bisa menambah pengetahuan para pembaca. Namun terlepas dari itu, kami memahami bahwa makalah ini masih jauh dari kata sempurna, sehingga kami sangat mengharapkan kritik serta saran yang bersifat membangun demi terciptanya makalah selanjutnya yang lebih baik lagi.

Bogor, 8 Nopember 2019

Zikrillah Saddam F

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Basis data adalah kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari database tersebut. Perangkat lunak yang digunakan untuk mengolah dan mengambil kueri (query) basis data disebut sistem manajemen basis data.

### 1.2 Rumusan Masalah

- Apa itu struktur hirarki basis data?
- Bagaimana cara menormalisasi basis data dengan baik?
- Apa yang harus dilakukan untuk bisa memanipulasi sebuah tabel?

### 1.3 Tujuan dan Manfaat

- Memahami struktur hirarki basis data
- Memahami bentuk diagram hubungan antar entitas
- Menerapkan teknik normalisasi basis data
- Memahami kelompok perintah dalam basis data
- Memahami tipe-tipe data dalam basis data
- Menerapkan *record*, *table* dan *field*
- Menerapkan bahasa SQL dasar untuk mengelola tabel
- Menerapkan bahasa SQL dasar untuk memanipulasi tabel
- Menerapkan fungsi agregasi dalam basis data

## BAB II

### PEMBAHASAN

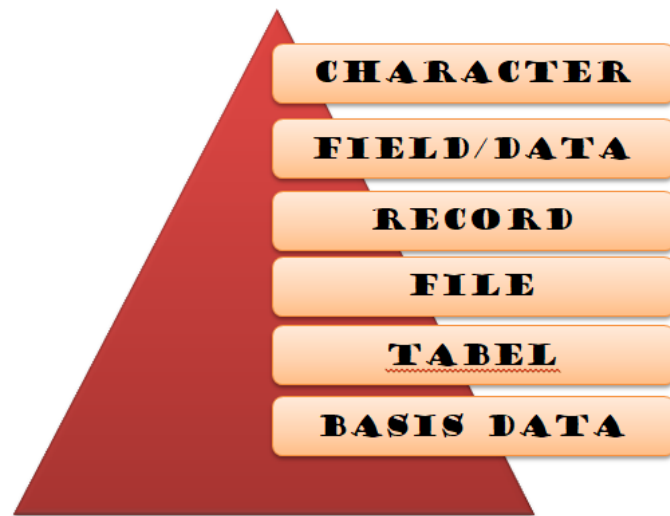
#### 2.1 Struktur Hirarki Basis Data

Basis data atau *database*, berasal dari kata basis dan data. **Basis** yaitu markas atau gudang ataupun tempat berkumpul, sedangkan **Data** ialah representasi nyata dan sebuah fakta yang mempunyai sesuatu objek. Secara garis besar **Basis data** itu sendiri merupakan kumpulan dari data yang berhubungan dengan yang lainnya dan disimpan di dalam komputer secara bersamaan semirip mungkin tanpa adanya sebuah pengulangan yang mungkin tidak digunakan dengan berbagai metode yang ada.

Pengertian Basis Data menurut para ahli, yaitu:

1. Database yaitu sekumpulan suatu informasi penting yang di simpan di sebuah komputer sehingga dapat di lihat untuk mendapatkan petunjuk dari data base tersebut.
2. Database yaitu pencerminan semua kenyataan yang berkaitan di simpan secara bersama sedemikian mirip dan tanpa pengulangan yang tidak berguna, untuk melengkapi semua kebutuhan tersebut.
3. Database yaitu sekumpulan informasi yang saling berhubung pada suatu benda atau permasalahan tertentu pada tujuan tertentu juga.

Struktur hirarki basis data merupakan model data yang mana data tersebut di atur dengan struktur tree, yaitu: character, field atau data, record, file, tabel dari basis data tersebut.

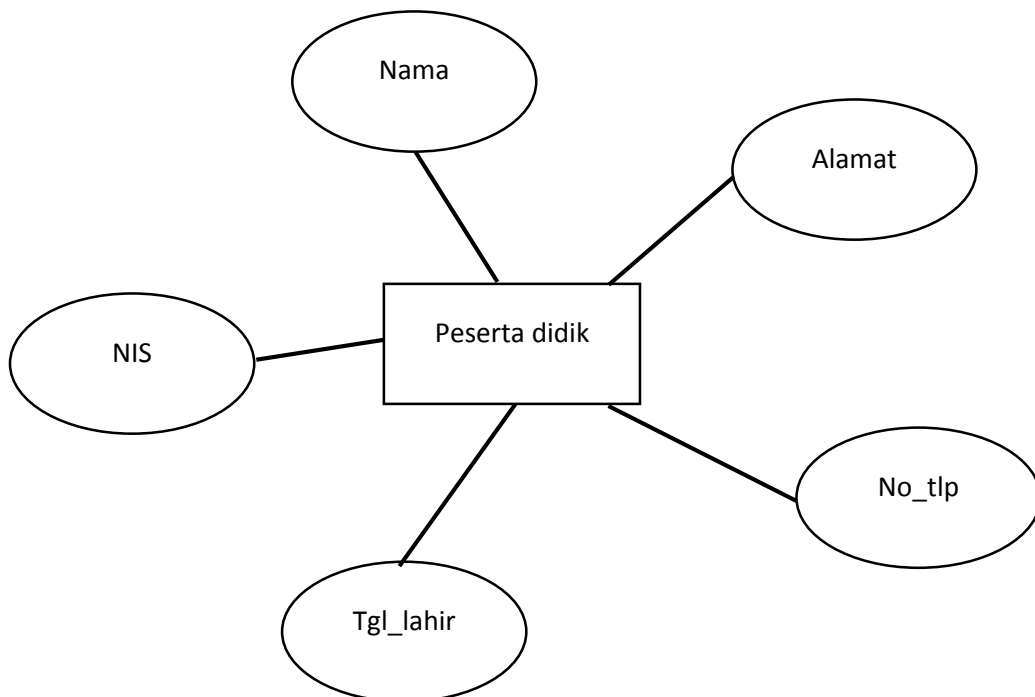


1. **Character** merupakan data yang paling kecil yang membuat sebuah data.
2. **Field atau data** adalah cerminan suatu atribut dari record sebuah data.
3. **Record** merupakan kumpulan dari beberapa data.
4. **File** yaitu bagian yang terdiri dari record yang sama.
5. **Tabel** merupakan hal yang pokok di penyimpanan sebuah data.

## 2.2 Diagram Hubungan Entitas

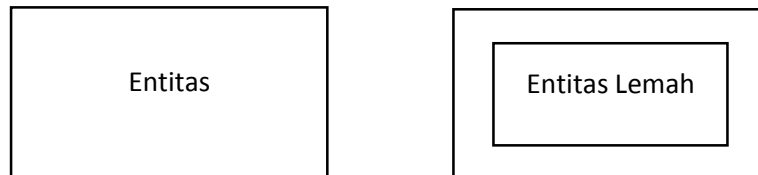
Diagram Hubungan Entitas atau *Entity Relationship Diagram*(ERD) adalah sebuah model data yang didasarkan atas persepsi dunia nyata, yang terdiri dari objek dasar yang disebut dengan entitas dan hubungan (relasi) antara objek-objek tersebut.

Ketika mendokumentasikan data atau proses hingga menjadi database, kita dapat meningkatkan pemahaman atas sistem tersebut dengan melihatnya dari beberapa perspektif. Oleh karena itu, agar dapat menjadi sistem database yang rapi dan terstruktur, kita membutuhkan Entity Relationship Diagram (ERD), yaitu sebuah model untuk menyusun database agar dapat menggambarkan data yang mempunyai relasi dengan database yang akan dibuat. Diagram ER biasanya berhubungan langsung dengan diagram *data flow* untuk menampilkan konten *data store*.



## 1. Entitas (*entity*)

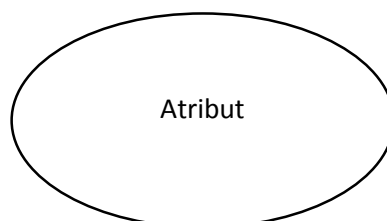
Kumpulan objek yang dapat diidentifikasi secara unik atau saling berbeda. Simbol dari entitas biasanya digambarkan dengan persegi panjang. Selain itu, ada juga “Entitas Lemah” yang dilambangkan dengan gambar persegi panjang kecil di dalam persegi panjang yang lebih besar. Disebut entitas lemah karena harus berhubungan langsung dengan entitas lain sebab dia tidak dapat teridentifikasi secara unik.



## 2. Atribut (*attribute*)

Adalah sifat atau karakteristik deskriptif suatu entitas. Gambar atribut diwakili oleh simbol elips dan terbagi menjadi beberapa jenis.

1. Atribut kunci (*key*): atribut yang digunakan untuk menentukan entitas secara unik. Contoh: NIS (Nomor Induk Siswa).
2. Atribut simpel: atribut bernilai tunggal yang tidak dapat dipecah lagi (*atomic*). Contoh: Alamat, tahun terbit buku, nama penerbit.
3. Atribut multivalui (*multivalue*): atribut yang memiliki sekelompok nilai untuk setiap entitas instan. Contoh: nama beberapa pengarang dari sebuah buku pelajaran.
4. Atribut gabungan (*composite*): atribut yang terdiri dari beberapa atribut yang lebih kecil dengan arti tertentu. Contoh: nama lengkap yang terbagi menjadi nama depan, tengah, dan belakang.
5. Atribut derivatif: atribut yang dihasilkan dari atribut lain dan tidak wajib ditulis dalam diagram ER. Contoh: usia, kelas, selisih harga.

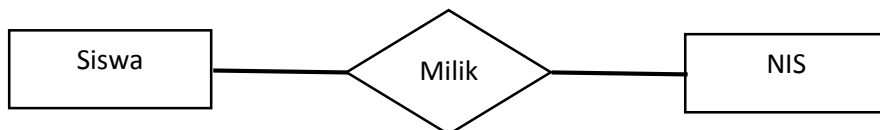




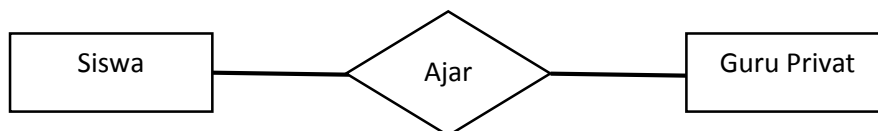
### 3. Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Gambar relasi diwakili oleh simbol belah ketupat. Relasi juga terbagi menjadi beberapa jenis.

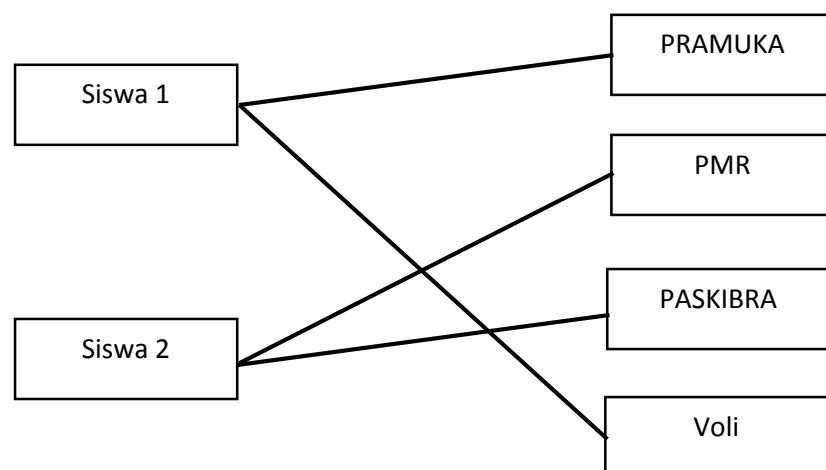
1. One to one. Setiap entitas hanya bisa mempunyai relasi dengan satu entitas lain. Contoh: siswa dengan nomor induk siswa



2. One to many. Hubungan antara satu entitas dengan beberapa entitas dan sebaliknya. Contoh: guru privat dengan murid dan sebaliknya.



3. Many to many. Setiap entitas bisa mempunyai relasi dengan entitas lain, dan sebaliknya. Contoh: siswa dan ekstrakurikuler.



### 4. Garis

Garis yang menghubungkan antar atribut untuk menunjukkan hubungan entitas pada diagram ER.

## 2.3 Normalisasi Basis Data

Normalisasi basis data merupakan sebuah teknik sistematis untuk meminimalkan redundansi data pada suatu *database* agar *database* tersebut dapat bekerja dengan optimal.

Tujuan normalisasi basis data adalah untuk menghilangkan dan mengurangi redundansi data dan tujuan yang kedua adalah untuk 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.

Bentuk – bentuk normalisasi basis data :

### 1. Normal Form Pertama (1NF)

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

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

NIM	NAMA	TGL_LAHIR	KODE_SUBJEK	NAMA_SUBJEK	SKS	NILAI	BOBOT
67	Andi	22/05/1992	TE-404	Fisika	3	A	4
68	Joni	17/05/1990	TE-400	Matematika	3	B	3
69	Udin	20/09/1991	SAE-100	Bahasa	2	A	4
69	Udin	20/09/1991	TE-404	Fisika	3	A	4

## 2. Normal Form Kedua (2NF)

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

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

NIM	KODE_SUBJEK	NILAI	BOBOT
67	TE-404	A	4
68	TE-400	B	3
69	SAE-100	A	4
69	TE-404	A	4

KODE_SUBJEK	NAMA_SUBJEK	SKS
TE-404	Fisika	3
TE-400	Matematika	3
SAE-100	Bahasa	2
TE-404	Fisika	3

NIM	NAMA	TGL_LAHIR
67	Andi	22/05/1992
68	Joni	17/05/1990
69	Udin	20/09/1991
69	Udin	20/09/1991

Pada tabel 1 merupakan pemecahan tabel dari normalisasi bentuk pertama dengan mengambil atribut NIM, Kode Subjek, Nilai dan Bobot. Dimana atribut tersebut memiliki ketergantungan dan keterkaitan terhadap atribut NIM yang merupakan primary key atau kunci dari tabel tersebut.

Pada tabel 2 juga merupakan pemecahan dari normalisasi bentuk pertama, tetapi diletakkan dalam tabel kedua dengan atribut Kode Subjek, Nama Subjek dan SKS, dimana atribut Kode Subjek menjadi Primary Key atau kunci dalam suatu form.

Pada tabel3 juga merupakan pemecahan dari normalisasi bentuk pertama. Dimana atribut NIM, Nama Mahasiswa, dan Tanggal Lahir terdapat dalam tabel 3 dengan Primary Key atau Kunci Utama yaitu NIM.

### 3. Normal Form Ketiga (3NF)

Normalisasi database dalam bentuk 3NF bertujuan untuk menghilangkan seluruh atribut atau field yang tidak berhubungan dengan primary key. Dengan demikian tidak ada ketergantungan transitif pada setiap kandidat key. Syarat dari bentuk normal ketiga atau 3NF adalah :

- Memenuhi semua persyaratan dari bentuk normal kedua.
- Menghapus kolom yang tidak tergantung pada primary key.

NIM	KODE_SUBJEK	NILAI
67	TE-404	A
68	TE-400	B
69	SAE-100	A
69	TE-404	A

NILAI	BOBOT
A	4
B	3
A	4
A	4

Pada tabel 1 merupakan pemecahan dari tabel 1 yang terdapat dalam normalisasi bentuk kedua, yang dimana atribut NIM, Kode Subjek, dan Nilai memiliki nilai sendiri dalam setiap field.

Pada tabel 2 yang terdapat dalam normalisasi bentuk ketiga juga merupakan lanjutan pemecahan dari tabel 1 yang terdapat dalam normalisasi bentuk kedua, yang dimana atribut Nilai menjadi Primary Key atau kunci utama dan diikuti oleh atribut lainnya sebagai atribut yang berkegantungan pada kunci utama. Sehingga data tersebut menjadi data yang minimalis karena user tidak kebingungan dalam penyusunan database yang tersedia, dan mudah untuk dimengerti.

## 2.4 Kelompok Perintah DDL, DML, dan DCL

DDL, DML, dan DCL itu merupakan suatu anjuran dari bahasa **SQL** atau *Structured Query Language*. SQL ialah sebuah bahasa pemrograman yang berfungsi untuk mengambil semua atau field data dari dalam database yang berhubungan. Istilah query sering terdengar untuk SQL tersebut yang berguna sebagai suatu bahasa yang dasar. Ketiga perintah itu yang terkait dengan SQL sangat berguna untuk dipahami dalam menguasai dalam membuat basis data.

### 1. DDL (Data Definition Language)

DDL atau yang mempunyai arti kecil sebagai bahasa pendefinisian data yang berkaitan dengan definisi dari struktur basis data. DDL dalam penggunaannya memiliki beberapa fungsi.

- Membuat sebuah objek database
- Melakukan perubahan objek database
- Menghapus data dan objek database

Adapun perintah yang berkaitan dengan DDL itu sendiri, ialah CREATE, ALTER, dan DROP:

1. **CREATE**, perintah yang berguna untuk menulis atau membuat sesuatu.
2. **Alter**, berfungsi untuk mengubah struktur tabel yang telah dibuat.
3. **Drop**, berfungsi untuk menghapus database atau tabel.

### 2. DML (Data Manipulation Language)

DML berfungsi sebagai anjuran yang bertujuan dalam kegiatan proses manipulator terhadap data yang ada pada data base. Fungsi utama dari DML ini ialah untuk melakukan proses olah data di dalam data base..

1. **INSERT**, berfungsi sebagai perintah untuk memasukkan data yang baru ke database.
2. **SELECT**, perintah berfungsi untuk menyeleksi atau mengambil data baru dalam database
3. **UPDATE**, perintah yang digunakan dalam pembaruan data di database
4. **DELETE**, perintah digunakan dalam penghapusan sebuah data dalam database

### 3. DCL (Data Control Language)

Perintah DCL berfungsi untuk mengontrol data dan benda di *database*. DCL ini berfungsi dalam melakukan control ke *server database*. Adapun yang termasuk perintah dasar dalam DCL ini, yaitu GRANT dan REVOKE.

1. **Grant**, digunakan untuk memberikan hak akses oleh admin kepada user.
2. **Revoke**, digunakan untuk mencabut hak akses yang telah diberikan.

## 2.5 Tipe Data Dalam Basis Data

Perangkat lunak yang di gunakan untuk mengelola *database* disebut dbms. Aplikasi dbms antra lain: Oracle 10g, SQL-Server 2005, MySQL 5, MS-Access. Masing- masing aplikasi mempunyai tipe data sendiri , tetapi mempunyai fungsi yang sama.

1. **Tipe data Karakter (character/char)**  
Jenis tipe data ini hanya memungkinkan sebuah variabel menyimpan informasi sebuah single karakter, seperti huruf "P", "c", "7", atau "@".
2. **Tipe data String**  
Tipe data string ini memungkinkan variabel menyimpan informasi untaian karakter, seperti kata atau kalimat.
3. **Tipe data Integer (int)**  
Jenis tipe data integer digunakan jika ingin menyimpan informasi angka (bilangan bulat) dalam variabel, tetapi bukan bilangan pecahan.
4. **Tipe data Floating Point Number (float)**  
Jenis tipe data ini dapat menyimpan informasi angka dengan format pecahan atau angka desimal.
5. **Tipe data Boolean (bool)**  
Jenis tipe data ini biasanya digunakan hanya untuk merepresentasikan dua kondisi, yakni nilai TRUE dan FALSE atau bernilai 1 dan 0 atau kondisi ON dan OFF.
6. **Tipe data Array**  
Tipe data ini pada dasarnya merupakan kumpulan sejumlah variabel bertipe data sama dengan ukuran tertentu, yang tersusun secara runtun. perlu Anda ketahui bahwa variabel pertama yang ada dalam array bukan memiliki indeks 1, melainkan indeks 0.

## **2.6 Record, Table dan Field**

### **RECORD**

Record merupakan kumpulan dari elemen-elemen data yang terkait dalam sebuah basis data. Secara ringkas, basis data dapat dikatakan sebagai sebuah tabel yang memiliki baris alias record dan kolom atau field.

### **TABLE**

Tabel merupakan daftar yang mana di dalamnya berisi berbagai Ikhtisar dari sebagian atau keseluruhan data informasi yang biasanya di tulis dengan bilangan/angka atau kata-kata yang tersusun secara sistematis

### **FIELD**

Field adalah suatu atribut yang menunjukkan/menampilkan suatu item yang berasal dari data, seumpama seperti nama, alamat dan lain sebagainya. Kumpulan yang berasal dari field membentuk suatu record.

Ada tiga jenis field perlu diketahui, yaitu :

1. Field Name : Wajib diberi nama untuk membedakan field yang satu dengan lainnya
2. Field Representasion : Jenis field (karakter, teks, tanggal, angka, dsb), lebar field (ruang maksimum yang bisa diisi bersama dengan karakter-karakter data).
3. Field Value : Isi yang berasal dari field untuk tiap-tiap record.

## 2.7 Bahasa SQL Dasar Untuk Mengelola Tabel

Mengelola tabel dapat diartikan juga sebagai mengatur, menyelenggarakan dan mengendalikan. Untuk dapat mengelola tabel dari sebuah database, kita perlu menggunakan fungsi DDL atau *Data Manipulation Language*.

Contoh :

```
CREATE TABLE siswa(nis INTEGER PRIMARY KEY, nama VARCHAR(20) NOT NULL,
alamat VARCHAR(30) NULL, tanggal_lahir DATE NULL);
```

Pada contoh perintah diatas dijelaskan bahwa akan di buat suatu table baru bernama siswa yang memiliki kolom entitas seperti nis yang di beri primary key atau tidak bisa sama dengan data lainnya, di sana akan di buat juga kolom nama dengan tipe data varchar yang memiliki kapasitas tampung sebanyak 20 karakter dan tidak dapat di kosongkan dengan mencatumkan kode NOT NULL, Selain itu ada juga alamat yang bertipe data Varchar dengan kapasitas tampung 30 karakter dan boleh kosong dengan kode NULL. Terakhir ada tanggal\_lahir dengan tipe data DATE dan boleh di kosongkan dengan kode NULL.

## 2.8 Bahasa SQL untuk Memanipulasi Tabel

Memanipulasi berarti merubah atau merekayasa. Untuk memanipulasi tabel, kita memerlukan fungsi DML(*Data Manipulation Language*).

Contoh :

```
INSERT INTO siswa VALUES(171688,"Zikrillah","Dekeng","22/03/"2003")
```

Perintah ini digunakan jika kita ingin mengubah isi suatu tabel dengan cara menambahkan data kedalamnya. Dalam perintah diatas dijelaskan bahwa data dari tabel siswa akan ditambahkan dengan mengetikkan INSERT lalu dengan kata kunci VALUES untuk memilih entitas mana yang akan ditambahkan datanya.



## 2.9 Fungsi Agregasi Dalam Basis Data

Fungsi agregasi(agregat) digunakan untuk menampilkan nilai-nilai atribut yang ada dalam tabel, sering pula ada kebutuhan untuk menampilkan data-data agregasi. Seperti: Banyaknya *record*, total nilai suatu atribut, rata-rata nilai atribut, nilai atribut terbesar atau terkecil. Data agregasi tersebut dapat diperoleh dengan fungsi-fungsi yang juga kita temukan padanannya dalam bahasa *query* formal.

Beberapa fungsi agregasi :

1. COUNT : Digunakan untuk menghitung jumlah (cacah) suatu data.

*Syntax* dasar : SELECT COUNT(nama\_kolom) FROM nama\_tabel

2. SUM : Digunakan untuk mendapatkan nilai total penjumlahan dari data-data yang ada.

*Syntax dasar*: SELECT SUM(nama\_kolom) FROM nama\_tabel

3. MAX : Digunakan untuk mendapatkan nilai terbesar dari data-data yang ada.

*Syntax* dasar : SELECT MAX(nama\_kolom) FROM nama\_tabel

4. MIN : Digunakan untuk mendapatkan nilai terkecil dari data-data yang ada.

*Syntax* dasar : SELECT MIN(nama\_kolom) FROM nama\_tabel

5. AVG : Digunakan untuk mendapatkan nilai rata-rata suatu atribut numerik hasil query.

*Syntax* dasar : SELECT AVG(nama\_field\_yg\_ingin\_dirata\_ratakan) FROM  
nama\_tabel

### BAB III

#### STUDI KASUS

Basis Data memiliki fungsi Agregasi, yaitu fungsi-fungsi khusus yang melibatkan sekelompok data (agregat). Fungsi ini berkaitan erat dengan proses perhitungan. Agregasi dalam SQL merupakan proses untuk mendapatkan nilai dari sekumpulan data yang telah dikelompokkan. Pengelompokan data didasarkan pada kolom atau kombinasi kolom yang dipilih..

Perhatikan tabel berikut :

	kode_obat	nama_obat	jenis_obat	masa_berlak	harga
	449	Paracetamol	Penenang	01/01/2018	Rp5.000
	425	Amoxicillin	Pereda	01/01/2018	Rp6.500
	633	C.Maleate	Vitamin	01/06/2015	Rp2.500

Hitunglah harga obat tertinggi dan terendah.

#### 1. MAX

Digunakan untuk mendapatkan nilai terbesar dari data-data yang ada.

*Syntax* dasar :

```
SELECT MAX (nama_kolom) FROM nama_tabel
```

Menggunakan fungsi agregasi MAX pada tabel obat

```
SELECT MAX (harga) as harga_tertinggi FROM Obat
```

Hasil dari penggunaan fungsi agregasi MAX

harga_tertinggi
Rp6.500

## 2. *MIN*

Digunakan untuk mendapatkan nilai terkecil dari data-data yang ada.

*Syntax* dasar :

```
SELECT MIN (nama_kolom) FROM nama_tabel
```

Menggunakan fungsi agregasi MIN pada tabel Obat

```
SELECT MIN (harga) as harga_terendah FROM Obat
```

Hasil dari penggunaan fungsi agregasi MIN :

harga_terendah
Rp2.500

## **BAB IV**

### **KESIMPULAN, PENUTUP, DAFTAR PUSTAKA**

#### **4.1 Kesimpulan**

1. Basis data merupakan kumpulan dari data yang saling terintegrasi satu dengan yang lainnya, dan tersimpan dalam perangkat keras komputer serta menggunakan perangkat lunak untuk bantuan dalam mengoperasikannya.
2. Untuk meminimalkan redudansi, digunakan teknik Normalisasi.
3. Tujuan Merancang Basis Data adalah untuk memenuhi informasi yang berisikan kebutuhan-kebutuhan user secara khusus dan aplikasi-aplikasinya., memudahkan pengertian struktur informasi serta mendukung kebutuhan-kebutuhan pemrosesan dan beberapa obyek penampilan
4. Perintah pada SQL terbagi menjadi 3 : DDL, DML dan DCL
5. Fungsi agregasi adalah fungsi yang dihasilkan dari sebuah kelompok data. Agregasi meliputi min(terkecil), max(terbesar), avg(rata-rata), sum(jumlah perhitungan dari suatu kolom), dan Count(mencari jumlah baris yang telah diinputkan).

#### **4.2 Saran**

Semoga makalah ini dapat memberi manfaat bagi pembaca. Kurangnya pengetahuan dalam penyusunan makalah ini diharapkan para pembaca lebih selektif dalam mengambil informasi yang ada. Kritik dan saran yang membangun juga diharapkan dari para pembaca agar makalah selanjutnya dapat lebih baik.

### 4.3 Daftar Pustaka

- <https://bocahkampus.com/cara-membuat-makalah> (03/11/2019)
- <https://garudacyber.co.id/artikel/1414-pengertian-hirarki-basis-data>  
(03/11/2019)
- <http://www.kajianpustaka.com/2012/10/teori-basis-data-database.html>  
(03/11/2019)
- <https://www.dewaweb.com/blog/entity-relationship-diagram> (03/11/2019)
- <https://rudiawan16.wordpress.com/normalisasi-database-beserta-pengertian-dan-contohnya> (03/11/2019)
- <https://garudacyber.co.id/artikel/672-kelompok-perintah-yang-termasuk-dalam-ddl-dml-dan-dcl-dalam-basis-data-dan-fungsinya> (07/11/2019)
- <https://adityarizki.net/mengenal-tipe-data-dan-variabel-dalam-pemrograman/>  
(04/11/2019)
- <https://infopengertian.biz/pengertian-field-character-record-file-data-dan-basis-data.html> (04/11/2019)
- <https://uprint.id/blog/penutup-makalah/> (09/11/2019)
- <https://www.romadecade.org/contoh-daftar-pustaka/#>! (09/11/2019)