

BASIS DATA

(C3) KELAS XII

Penulis :

Drs. Supriyanto, M.T

PT. KUANTUM BUKU SEJAHTERA

BASIS DATA

SMK/MAK Kelas XII

Penulis	: Drs. Supriyanto, M.T
Editor	: Tim Quantum Book
Perancang sampul	: Tim Quantum Book
Perancang letak isi	: Tim Quantum Book
Penata letak	: Tim Quantum Book
Ilustrator	: Tim Quantum Book
Tahun terbit	: 2019
ISBN	: (terlampir di cover)
Penerbit	: PT. Kuantum Buku Sejahtera
Alamat	: Jalan Pondok Blimbing Indah Selatan X N6 No 5 Malang - Jawa Timur

Tata letak buku ini menggunakan program Adobe InDesign CS3, Adobe Illustrator CS3, dan Adobe Photoshop CS3.

Font isi menggunakan Myriad Pro (10 pt)

B5 (17,6 × 25) cm

x + 181 halaman

© Hak cipta dilindungi oleh undang-undang.

Dilarang menyebarluaskan dalam bentuk apapun
tanpa izin tertulis

Undang-Undang Republik Indonesia Nomor 19 Tahun 2002 Tentang Hak Cipta Pasal 72 Ketentuan Pidana Sanksi Pelanggaran.

1. Barang siapa dengan sengaja dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 2 ayat (1) atau Pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp5.000.000.000,00 (lima miliar rupiah).
2. Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran Hak Cipta atau Hak Terkait sebagaimana dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

Kata Pengantar

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, yang telah memberikan anugerah-Nya, sehingga penulis mampu menyelesaikan penulisan buku pembelajaran untuk SMK/MAK Ini.

Buku ini ditulis sebagai salah satu sumber belajar siswa SMK/MAK kelas XII untuk mempelajari dan memperdalam materi Basis Data. Selain itu, buku ini ditulis secara umum dalam rangka ikut serta mencerdaskan bangsa Indonesia di era perkembangan ilmu pengetahuan dan teknologi saat ini.

Setiap bab dalam buku ini dilengkapi dengan Kompetensi Inti, Kompetensi Dasar, Kata Kunci, Tujuan Pembelajaran, Peta Konsep, Aktivitas Siswa, Tugas Siswa, Info, Rangkuman, Uji Kompetensi, dan Tugas Proyek. Pembahasan materi disajikan dengan bahasa yang lugas dan mudah kita pahami, dari pembahasan secara umum ke pembahasan secara khusus.

Dengan demikian, buku ini diharapkan dapat menjadi teman sekaligus menjadi bacaan yang menyenangkan bagi Anda untuk mempelajari lebih dalam tentang Basis Data dan menerapkannya dalam kehidupan sehari-hari untuk diri sendiri dan lingkungan.

Akhirnya, semoga buku pelajaran Basis Data SMK/MAK Kelas XII ini bermanfaat bagi siswa dan seluruh pembaca dalam memperoleh pengetahuan.

Selamat belajar, semoga sukses.

Penulis

Daftar Isi

BAB 1	Perintah SQL dalam Akses Multi Table	1
	A. Sudut Pandang Basis Data.....	3
	B. <i>Select Multitable</i>	4
	Uji Kompetensi.....	13
BAB 2	Memaniplulasi Basis Data dalam SQL	17
	A. Dasar-dasar Perintah Select Bertingkat.....	19
	B. Membuat Select Option Bertingkat (Combo Box).....	24
	Uji Kompetensi.....	40
BAB 3	Implementasi SQL Pada Sistem Basis Data Client-Server	43
	A. SQL pada Sistem Basis Data Client-Server	45
	B. Perintah Koneksi Basis Data	61
	Uji Kompetensi.....	67
BAB 4	Arsitektur Relasional DBMS	71
	A. Sistem Relasi Basis Data.....	73
	B. Relasi Antar tabel dalam Database.....	81
	Uji Kompetensi.....	86
BAB 5	Pembuatan Basis Data pada RDBMS	89
	A. Konsep Relational Database Model	91
	B. Pembuatan Basis Data Berbasis RDBMS.....	95
	Uji Kompetensi.....	104
BAB 6	Struktur Penyimpanan pada RDBMS	107
	A. Prosedur Penyimpanan Pada RDBMS.....	109
	B. Spesifikasi Penyimpanan pada RDBMS.....	114
	Uji Kompetensi.....	122
BAB 7	Pengendalian Server pada RDBMS	125
	A. Proses Administrasi Server Basis Data Berbasis RDBMS	127
	B. Pengelolaan Server RDBMS.....	131
	Uji Kompetensi.....	139
BAB 8	Pengamanan Sistem Basis Data pada RDBMS	143
	A. Pengamanan Sistem Basis Data.....	145
	B. Pembatasan Kewenangan Hak Akses User Basis Data RDBMS	150
	Uji Kompetensi.....	158

BAB 9 Replikasi Basis Data 161

 A. Dasar-Dasar Replikasi Basis Data..... 163

 B. Penggunaan Replikasi Basis Data 168

 Uji Kompetensi..... 172

Daftar Pustaka 175

Glosarium 177

Indeks 179

Daftar Penulis 181

Daftar Gambar

Bab 1 Perintah SQL dalam Akses *Multi Table*

Gamabr 1.1	Tabel pada basis data	1
Gambar 1.2	Implementasi table	3
Gambar 1.3	Data yang tersimpan dalam database	4
Gambar 1.4	Bentuk SQL join	6
Gambar 1.5	Diagram venn untuk inner join.....	7
Gambar 1.6	Left (outer) Join.....	9
Gambar 1.7	Diagram venn untuk Right Join.....	10
Gambar 1.8	Diagram venn untuk Full Join.....	11

Bab 2 Norma, Standar, Prosedur, Kaidah Kearsipan

Gambar 2.1	Basis data	17
Gambar 2.2	Penggunaan select.....	19
Gambar 2.3	Implementasi SELECT Query	20
Gambar 2.4	Penggunaan Subquery	20
Gambar 2.5	Penggunaan nested query	21
Gambar 2.6	Bentuk combo box pada oracle.....	24
Gambar 2.7	Hasil akhir	39

Bab 3 Peralatan Kearsipan

Gambar 3.1	Arsitektur client-server dalam database.....	43
Gambar 3.2	Aplikasi basis data	45
Gambar 3.3	MySQL client.....	45
Gambar 3.4	Arsitektur client-server pada MySQL	46
Gambar 3.5	Redundansi data	47
Gambar 3.6	Pengaksesan data	47
Gambar 3.7	Isolasi data untuk standarisasi pada rumah sakit.....	48
Gambar 3.8	Multiple user	48
Gambar 3.9	Mengaktifkan MySQL command line client.....	49
Gambar 3.10	Jendela cmd windows.....	50
Gambar 3.11	Perintah CMD	50
Gambar 3.12	Tampilan command prompt.....	51
Gambar 3.13	Mengubah folder	51
Gambar 3.14	Menjalankan mysqld menggunakan Windows 10	51
Gambar 3.15	Tampilan dari file my.ini.....	52

Gambar 3.16	Memasukkan password.....	52
Gambar 3.17	Lokasi mysqld.exe.....	53
Gambar 3.18	Menghentikan MySQL server secara manual.....	53
Gambar 3.19	Tab details di task manager.....	54
Gambar 3.20	End process.....	54
Gambar 3.21	Menjalankan MySQL Client.....	55
Gambar 3.22	Menjalankan MySQL client dari cmd windows.....	55
Gambar 3.23	Shortcut cmd.....	56
Gambar 3.24	MySQL Installer.....	56
Gambar 3.25	Jendela MySQL installer.....	56
Gambar 3.26	Jendela accounts and roles.....	57
Gambar 3.27	Windows service.....	57
Gambar 3.28	Apply configuration.....	57
Gambar 3.29	Component services.....	58
Gambar 3.30	Menghapus windows service.....	58
Gambar 3.31	Password dari user.....	59
Gambar 3.32	Keluar dari client.....	60
Gambar 3.33	Menginputkan password.....	60
Gambar 3.34	TCP/IP (Transmission Control Protocol/Internet Protocol).....	61
Gambar 3.35	MySQL workbench.....	63
Gambar 3.36	Folder instalasi.....	63
Gambar 3.37	Folder bin.....	64
Gambar 3.38	Folder data.....	64
Gambar 3.39	Folder lib.....	65
Gambar 3.40	Folder share.....	65
Gambar 3.41	Show database.....	66

Bab 4 Penanganan Surat Masuk

Gambar 4.1	Sistem informasi dabatase.....	71
Gambar 4.2	Relational database management system (RDBMS).....	73
Gambar 4.3	Struktur RDBMS.....	74
Gambar 4.4	Penggunaan database.....	74
Gambar 4.5	Penggunaan Access 2016.....	75
Gambar 4.6	Membuat field.....	76
Gambar 4.7	Menggunakan design view.....	77
Gambar 4.8	Primary key pada "id" pada salah satu tabel.....	77
Gambar 4.9	Penggunaan foreign key.....	78

Gambar 4.10	Contoh candidate key	79
Gambar 4.11	Penggunaan index	80
Gambar 4.12	Kolom (column) dalam relational database	81
Gambar 4.13	Relasi one-to-one.....	81
Gambar 4.14	Relasi one-to-many	82
Gambar 4.15	Relasi many-to-many (N-M).....	83

Bab 5 Penanganan Surat Keluar

Gambar 5.1	Penggunaan basis data	89
Gambar 5.2	Dr. Edgar F. Codd	91
Gambar 5.3	Penggunaan database	91
Gambar 5.4	Mencetak transaksi di perbankan membutuhkan DBMS.....	92
Gambar 5.5	Sistem Manajemen Perpustakaan	93
Gambar 5.6	DBMS di rumah sakit.....	94
Gambar 5.7	Michael Widenius.....	95
Gambar 5.8	Membuat Database Perpustakaan	95
Gambar 5.9	Login MySQL sebagai Root	96
Gambar 5.10	Bentuk akhir relasi	102

Bab 6 Klasifikasi dan Indeks Arsip

Gambar 6.1	Bentuk ERD rental mobil	107
Gambar 6.2	Entity Relationship Model.....	109
Gambar 6.3	Model Data Storage pada RDBMS	110
Gambar 6.4	Model Hierarki	110
Gambar 6.5	Model Jaringan	111
Gambar 6.6	Model Relasional.....	111
Gambar 6.7	IBM DB2 RDBMS	112
Gambar 6.8	Instalasi pada DB2	112
Gambar 6.9	Object-Oriented Database Model	113
Gambar 6.10	Sistem penyimpanan data berbentuk relasi DBMS	114
Gambar 6.11	Instalasi pada MySQL versi 5.7.19	114
Gambar 6.12	Logo MariaDB.....	115
Gambar 6.13	Reover innodb database dengan file .frm dan .ibd	115
Gambar 6.14	MyISAM.....	116
Gambar 6.15	Penyimpanan CSV.....	117
Gambar 6.16	Aplikasi XAMPP.....	118
Gambar 6.17	Salah satu bentuk Direct Access Storage Device (DASD).....	119
Gambar 6.18	Salah satu bentuk Sequential Access Storage Device (SASD)	119

Gambar 6.19	Postgre	120
Bab 7	Sistem Penyimpanan Arsip	
Gambar 7.1	Sistem basisdata komputer.....	125
Gambar 7.2	Administrasi Table di SQLYog	127
Gambar 7.3	Tampilan MySQL Query Browser	127
Gambar 7.4	PHPMyAdmin	130
Gambar 7.5	MySQL-Front	130
Gambar 7.6	Software stack AMPPS.....	131
Gambar 7.7	Database versi open source	132
Gambar 7.8	Backup MySQL	134
Gambar 7.9	Bentuk perintah database (SQL Dump).....	134
Gambar 7.10	Pilih menu DB	135
Gambar 7.11	Dump Data	135
Gambar 7.12	Restore dari MySQL.....	136
Gambar 7.13	Pilih menu DB	136
Gambar 7.14	Memilih file backup.....	137
Gambar 7.15	Proses import	137
Gambar 7.16	Proses impor data pada SQLYog.....	138
Bab 8	Penggunaan Arsip	
Gambar 8.1	Pengelolaan keamanan data server	143
Gambar 8.2	Proses basis data	145
Gambar 8.3	Bentuk enkripsi password.....	146
Gambar 8.4	Enkripsi password	147
Gambar 8.5	Perintah Transact SQL.....	148
Gambar 8.6	Pengaturan Hak Akses (Privileges)	149
Gambar 8.7	Penggunaan hak akses berbasis GUI	150
Bab 9	Pemeliharaan Arsip	
Gambar 9.1	Skema replikasi MySQL.....	161
Gambar 9.2	Penempatan replikasi data.....	163
Gambar 9.3	Replicated database environment (DBE)	163
Gambar 9.4	Jaringan petri algoritme master dalam algoritma replikasi master-slave.....	164
Gambar 9.5	Bentuk replikasi	165
Gambar 9.6	Skema master dan slave	166
Gambar 9.7	Skema master dan slave	169
Gambar 9.8	Cara kerja replikasi database	170

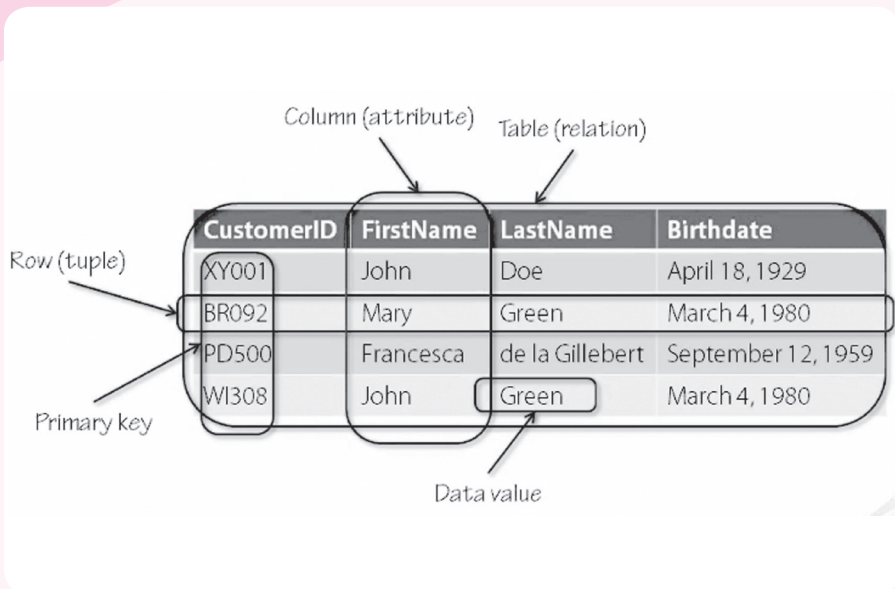
Daftar Tabel

Tabel 1.1 Elemen-elemen pada tabel.....	4
Tabel 1.2 Data calon siswa baru	5
Tabel 7.1 Keterangan gambar tampilan MySQL Query Browser.....	128
Tabel 7.2 Empat golongan alat pengolah data RDBMS.....	132
Tabel 8.1 Komponen-komponen di dalam database MySQL.....	149
Tabel 8.2 Prosedur penambahan dan pengaturan hak asasi user di MySQL.....	156
Tabel 9.1 Cara pengelolaan penguncian objek.....	167

BAB

1

Perintah SQL dalam Akses *Multi Table*



Gambar 1.1 Tabel pada basis data

Sumber: https://miro.medium.com/max/736/0*MG1zABIBry16OTOH.PNG

Kompetensi Dasar

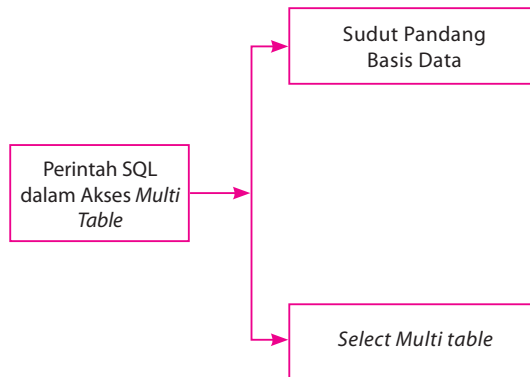
- 3.10 Menerapkan penggunaan SQL untuk mengakses *multi table*.
- 4.10 Membuat perintah SQL untuk mengakses *multi table*.

Tujuan Pembelajaran

Setelah mempelajari bab ini, siswa diharapkan mampu:

1. menjelaskan penggunaan SQL untuk mengakses *multi table*.
2. memahami pembuatan perintah SQL untuk mengakses *multi table*.

Peta Konsep

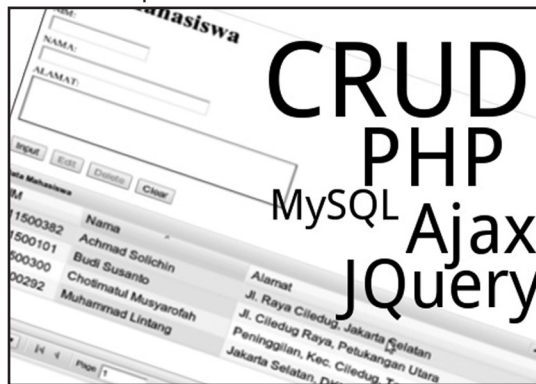


Istilah basis data banyak menimbulkan interpretasi yang berbeda. Pada saat maraknya paket pemrograman dbase sebuah berkas dengan ekstensi *.DBF dapat disebut basis data. Suatu *Database Manajemen System* (DBMS) akan berisi satu koleksi data yang akan saling berelasi dan 1 set program untuk mengakses data yang sudah diciptakan. Jadi sebenarnya DBMS terdiri dari *database* dan set program pengelola untuk memproses data. Prinsip utamanya adalah pengaturan data dan arsip. Tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data atau arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, basis data menggunakan media penyimpanan elektronis seperti *disk* (disket atau *harddisk*). Hal ini merupakan konsekuensi logis, karena lemari arsip langsung dikelola dan ditangani oleh manusia, sementara basis data dikelola dan ditangani melalui perantaraan alat mesin pintar elektronis (komputer). Perbedaan media ini yang selanjutnya melahirkan perbedaan-perbedaan lain yang menyangkut jumlah dan jenis metode atau cara yang dapat digunakan dalam upaya penyimpanan.

Materi Pembelajaran

A. Sudut Pandang Basis Data

Basis data bukan hanya sekedar penyimpanan data secara elektronis dengan bantuan komputer. Artinya tidak semua bentuk penyimpanan data secara elektronis bisa disebut basis data. Kita dapat menyimpan dokumen berisi data dalam *file* teks dengan program pengolah kata, *file spread sheet* dan lain-lain, tetapi tidak dapat disebut sebagai basis data. Karena di dalamnya tidak ada pemilahan dan pengelompokkan data sesuai jenis dan fungsi data, sehingga akan menyulitkan pencarian data kelak. Hal yang sangat ditonjolkan dalam basis data adalah pengaturan, pemilahan, pengelompokkan, dan pengorganisasian data yang akan disimpan sesuai fungsi dan jenisnya. Pemilahan, pengelompokkan, pengorganisasian dapat berbentuk sejumlah *file* atau tabel terpisah atau dalam bentuk pendefinisian kolom-kolom *field-field* data dalam setiap file dan tabel.



Gambar 1.2 Implementasi table

Sumber: <http://nramini.net/wp-content/uploads/2012/09/crud-php-jquery.png>

Basis data dapat didefinisikan dalam sejumlah sudut pandang sebagai berikut:

1. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika memiliki arti implisit. Sehingga data yang terkumpul secara acak dan tanpa memiliki arti tidak dapat disebut basis data.
2. Basis data merupakan penyajian suatu aspek dari dunia nyata (*the real world* atau *miniworld*). Misalnya basis data perbankan, perpustakaan, pertanahan, dan perpajakan.
3. Basis data perlu dirancang, dibangun dan dikumpulkan untuk suatu tujuan. Basisdata dapat digunakan oleh beberapa pemakai dan aplikasi yang sesuai dengan kepentingan pemakai.
4. Himpunan kelompok data (arsip) saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
5. Kumpulan data yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu untuk memenuhi berbagai kebutuhan.

6. Kumpulan *file*, tabel, dan arsip yang saling berhubungan dan disimpan dalam media penyimpanan elektronis.

Berdasarkan batasan-batasan di atas maka dapat dikatakan bahwa basis data memiliki berbagai sumber data dalam pengumpulan data, bervariasi derajat interaksi kejadian dari dunia nyata, dirancang dan dibangun agar dapat digunakan oleh beberapa *user* untuk berbagai kepentingan. Hal yang perlu diperhatikan adalah adanya penyimpanan data yang rangkap. Misalnya saja data siswa dalam suatu jurusan akan berbeda penanganannya dengan menangani data seluruh sekolah yang lebih kompleks.

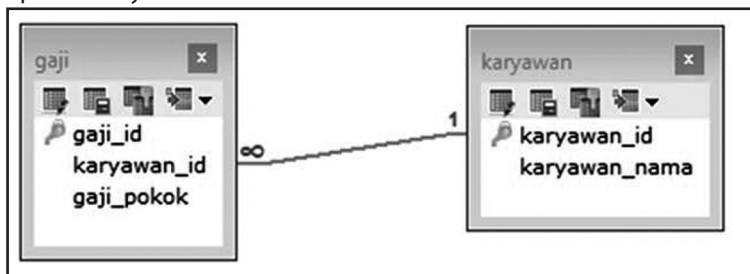
Tabel 1.1 Elemen-elemen pada tabel

Nama Field	Tipe	Panjang
id_pelanggan*	varchar	5
nama_pelanggan	varchar	30
alamat	text	-
telepon	varchar	15
email	varchar	30

<https://bahasaweb.com/wp-content/uploads/2016/10/table.jpg>

B. Select Multitable

Pada pembuatan suatu aplikasi, terkadang memerlukan tampilan data yang tidak hanya berasal dari 1 (satu) tabel, namun bisa dari beberapa tabel sekaligus. Misalnya ingin menampilkan nama provinsi berikut dengan kota-kota di dalamnya. Dari contoh tersebut, kita harus dapat menggabungkan minimal dua tabel, yaitu provinsi dan kota. Atau menampilkan data karyawan lengkap dengan gajinya, sehingga harus menggabungkan tabel data karyawan dan data gaji. Data yang tersimpan dalam *database* dapat bervariasi dalam hal jumlah dan tingkat kompleksitasnya.



Gambar 1.3 Data yang tersimpan dalam database

Sumber: <http://mfikri.com/assets/images/7970566dcf4ba5a2ddd9abe34d9dce9b.jpg>

1. Data Selection

Adapun untuk melihat data (Selection), Query yang digunakan adalah SELECT diikuti beberapa pernyataan khusus yang berhubungan dengan tabel yang diseleksi. Pada dasarnya, SELECT merupakan perintah DML (Data Manipulation Language) untuk menghasilkan recordset dengan mengekstrak data dari *database*. Perintah SELECT ini sering dipakai dan memiliki bentuk yang kompleks. Bentuk sederhana dari SELECT adalah sebagai berikut.

```
SELECT select_list [INTO table_new] FROM table_source [WHERE
search_condition] [GROUP BY group_by_expression] [HAVING search_
condition] [ORDER BY order_expression [ASC | DESC] ]
```

Misalnya memiliki tabel nilai_siswa dengan field (kolom) nama, kelas, mapel, nilai, dan keterangan. Maka perintah SQL untuk menampilkan khusus pada field nama, mapel, dan nilai adalah sebagai berikut.

```
SELECT Nama, mapel, nilai FROM nilai_siswa
```

Namun jika ingin menampilkan seluruh field pada tabel Customer1 dapat menggunakan perintah asterik (*) sebagai berikut.

```
SELECT * FROM nilai_siswa
```

2. Fungsi Select Multiple Table

Select multiple table berfungsi untuk menampilkan data yang berhubungan dari dua buah tabel, misalnya ingin menampilkan data nm_siswa dan nm_kelas tiap siswa. Pada tabel siswa, nama kelas siswa ditunjukkan oleh id_kelas agar dapat melihat langsung nm_kelas tiap siswa, maka perintah *select multiple table* dapat digunakan. Pada syntax from nama tabel siswa ditulis lebih dahulu dibandingkan nama tabel tb_kelas. Hal tersebut menyebabkan *data ascending* menurut table tb_kelas. Misalnya data dalam tb_kelas pada urutan teratas adalah data BSD. Jika nama table tb_kelas ditulis lebih dulu maka data yang ditampilkan akan *ascending* sesuai dengan tabel siswa. Misalnya data tabel siswa pada urutan teratas adalah Yurizka.

Tabel 1.2 Data calon siswa baru

No.	Nama	Jenis Kelamin	Agama	Nama Ortu/Wali	Nomor Telepon	Alamat
1.	Yurizka	Perempuan	Islam	Yuli	89654771341	Depok
2.	Lidia	Perempuan	Kristen	Yunita	89654771342	Serpong

<https://1.bp.blogspot.com/-X2ND3NKpBKI/V9MEPISdKQI/AAAAAAAAABk/vsUnTbDQ0tMtHqRuKp8huuF8ofJNkRRUwCPcB/s1600/daftar%2Bcalon%2Bsiswa%2Bbaru.JPG>

Di dalam suatu RDBMS menjadi suatu kewajiban jika dalam satu *database* dapat terdiri dari beberapa tabel, di mana masing-masing tabel tersebut berhubungan (relasi) satu sama lain. Relasi antartabel dapat berupa relasi 1-1, 1-M, atau M-N. Misalnya tabel customer1 berhubungan dengan dt_pesan, dt_pesan dengan barang, dan lain-lain. Namun adakalanya memerlukan tampilan data yang tidak hanya berasal dari 1 (satu) tabel, tetapi bisa dari beberapa tabel sekaligus. Hal ini dapat dilihat pada gambar pemodelan data konseptual (*class diagram*). Misalnya dari *class diagram* ingin menampilkan nama customer1 beserta transaksi yang pernah dilakukannya. Oleh sebab itu, bisa dilakukan dengan menggabungkan minimal dua tabel, yaitu customer1 dan dt_pesan. Tabel customer1 hanya menampilkan id, nama, dan e-mail sebagai berikut.

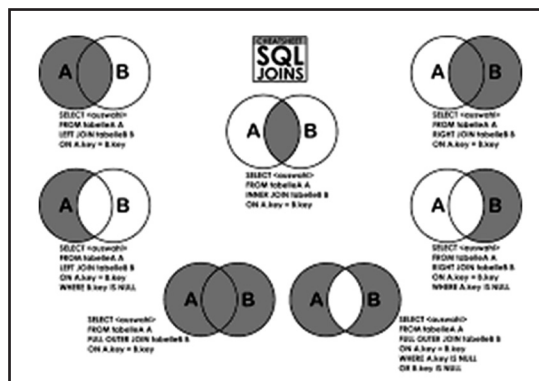
id_cst	nm_customer1	email
CST01	Nanik Ramini	nramini@gmail.com
CST02	Zaskia Sari	zask.sari@gmail.com
CST03	Herman Mario	her_mario19@yahoo.com
CST04	Rangga Mardi	nggamard@msn.com

Sedangkan tabel dt_pesanan hanya menampilkan id_dt_pesanan, id_cst, dan tgl_dt_pesanan sebagai berikut.

id_dt_pesanan	id_cst	tgl_dt_pesanan
1	CST01	2019-04-02
2	CST02	2019-04-05
3	CST02	2019-04-10
4	CST04	2019-01-20
5	CST01	2018-12-14

3. Perintah Join

Biasanya saat mengambil data kita akan mengakses beberapa tabel secara bersamaan. Salah satu cara mengambil data dari beberapa tabel sekaligus dapat dilakukan dengan menggunakan join. Pada SQL terdapat sebuah perintah join yang berfungsi untuk menghubungkan tabel yang satu dengan tabel yang lain yang saling berhubungan atau berelasi. Tujuannya adalah untuk menampilkan beberapa data dari table yang berbeda dengan menggunakan satu perintah. Join menghubungkan satu tabel dengan tabel yang lain. Join memperbolehkan untuk mengambil data dari beberapa tabel melalui satu query. Hanya menggunakan sebuah tabel artinya hanya dapat menyimpan/memperoleh data terbatas atau justru menyimpan/memperoleh data terlalu banyak sehingga tabelnya menjadi kurang baik. Hal inilah yang dimaksud dengan *relational* dari istilah *relational database*.

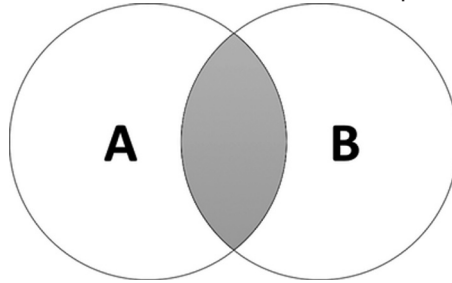


Gambar 1.4 Bentuk SQL join
Sumber: Dokumen penerbit

Perintah Join dikategorikan menjadi sebagai berikut:

a. Inner Join

Data yang akan ditampilkan pada perintah *inner join* hanya data yang memiliki pasangan saja, sedangkan data pada tabel yang tidak memiliki sebuah kesamaan maka data tersebut tidak akan ditampilkan. Inner Join identik dengan sebuah perintah yang digunakan untuk menampilkan sebuah data atau *record* dengan menghubungkan dua tabel atau lebih dalam satu perintah.



Gambar 1.5 Diagram venn untuk inner join
Sumber: <https://static.cdn-cdpl.com/source/998b78e349061b4971c0a2b0e8d6be41/sql1.png>

Format penulisan yang digunakan adalah sebagai berikut.

```
SELECT * FROM table_1 INNER JOIN table_2 ON table_1.PK =  
table_2.FK  
SELECT * FROM POST INNER JOIN category on post.category_  
id=category.id_category;
```

Tabel akan digabungkan dua arah menggunakan Inner Join, sehingga tidak ada data yang NULL di satu sisi. Misalnya menggabungkan tabel *customer1* dan *dt_pesan* dengan tujuan untuk menampilkan daftar *customer1* yang pernah melakukan pemesanan (transaksi).

1) Penggabungan dengan Inner Join

Cara penggabungan dengan Inner Join menggunakan bentuk umum sebagai berikut:

```
SELECT table_1.*, table_2.*  
FROM table_1 INNER JOIN table_2  
ON table_1.PK=table_2.FK;
```

Perintah SQL untuk menggabungkan tabel *customer1* dan *dt_pesan* adalah sebagai berikut:

```
SELECT customer1.id_cst, customer1.nm_customer1,  
dt_pesan.id_dt_pesan, dt_pesan.tgldt_pesan  
FROM customer1 INNER JOIN dt_pesan  
ON customer1.id_cst=dt_pesan.id_cst;
```

Hasilnya adalah sebagai berikut:

id_cst	nm_customer1	id_dt_pesan	tgldt_pesan
CST01	Nanik Ramini	1	2019-04-02
CST01	Nanik Ramini	5	2018-12-14
CST02	Zaskia Sari	2	2019-04-05
CST02	Zaskia Sari	3	2019-04-10
CST04	Rangga Mardi	4	2019-01-20

2) Penggabungan dengan WHERE

Cara penggabungan dengan WHERE menggunakan bentuk umum sebagai berikut:

```
SELECT table_1.*, table_2.* FROM table_1, table_2
WHERE table_1.PK=table_2.FK;
```

Perintah SQL yang digunakan untuk menggabungkan tabel customer1 dan dt_pesan: sebagai berikut:

```
SELECT customer1.id_cst, customer1.nm_customer1,
dt_pesan.id_dt_pesan, dt_pesan.tgldt_pesan
FROM customer1, dt_pesan
WHERE customer1.id_cst=dt_pesan.id_cst;
```

Hasilnya adalah sebagai berikut:

id_cst	nm_customer1	id_dt_pesan	tgldt_pesan
CST01	Nanik Ramini	1	2019-04-02
CST01	Nanik Ramini	5	2018-12-14
CST02	Zaskia Sari	2	2019-04-05
CST02	Zaskia Sari	3	2019-04-10
CST04	Rangga Mardi	4	2019-01-20

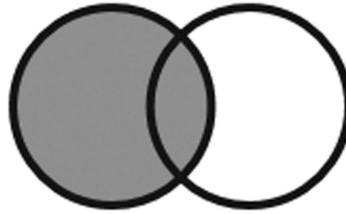
Pada hasil perintah query di atas terdapat 5 (lima) transaksi yang dilakukan oleh 3 (tiga) orang customer1. Namun demikian, terdapat satu customer1 yang memiliki id customer1 CST03 tidak ditampilkan. Customer1 tersebut tidak ditampilkan karena belum pernah melakukan transaksi.

b. Outer Join

Tabel yang akan digabungkan dengan Outer Join menjadi satu arah, sehingga terdapat data NULL (kosong) di satu sisi. Misalnya menggabungkan tabel customer1 dan dt_pesan untuk menampilkan daftar customer1 yang pernah melakukan pemesanan (transaksi). Secara umum, Outer Join terbagi menjadi sebagai berikut.

1) Left Join

LEFT OUTER JOIN dapat ditulis menggunakan dua cara, yaitu dengan klausa LEFT OUTER JOIN atau cukup dengan klausa LEFT JOIN saja. Left Join (*left outer join*) identik dengan suatu perintah yang digunakan untuk mengembalikan semua nilai dari tabel kiri ditambah dengan nilai dari tabel kanan yang sesuai (atau null jika tidak ada nilai yang sesuai).



Gambar 1.6 Left (outer) Join
Sumber: <https://jagowebdev.com/wp-content/uploads/2017/02/Ilustrasi-Left-Outer-Join.png>

Bentuk umum Left Join adalah sebagai berikut:

```
SELECT table_1.*, table_2.*  
FROM table_1 LEFT JOIN table_2  
ON table_1.PK=table_2.FK;
```

Format penulisan query yang digunakan adalah sebagai berikut:

```
SELECT * FROM table_1 LEFT JOIN table_2 ON table_1.PK =  
table_2.FK  
SELECT * FROM POST LEFT JOIN category on post.category_  
id=category.id_category;
```

Perintah SQL untuk menggabungkan tabel customer1 dan dt_pesan menggunakan Left Join adalah sebagai berikut:

```
SELECT customer1.id_cst, customer1.nm_customer1,  
dt_pesan.id_dt_pesan, dt_pesan.tglDt_pesan  
FROM customer1 LEFT JOIN dt_pesan  
ON customer1.id_cst=dt_pesan.id_cst;
```

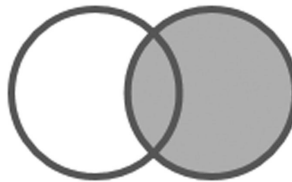
Hasilnya adalah sebagai berikut:

id_cst	nm_customer1	id_dt_pesan	tglDt_pesan
CST01	Nanik Ramini	1	2019-04-02
CST01	Nanik Ramini	5	2018-12-14
CST02	Zaskia Sari	2	2019-04-05
CST02	Zaskia Sari	3	2019-04-10
CST03	Herman Mario	NULL	NULL
CST04	Rangga Mardi	4	2019-01-20

Penggunaan Left Join juga akan menampilkan data customer1 dengan id CST03, walau pun customer1 tersebut belum pernah bertransaksi. Pada kolom id_dt_pesan dan tglDt_pesan untuk customer1 CST03 isinya NULL yang berarti pada tabel dt_pesan untuk customer1 tersebut benar-benar tidak ada. Pada dasarnya, fungsi dari Left Join hampir sama dengan Inner Join dan Right Join. Jika pada Right Join akan membuat sebuah parameter di sebelah kanan, maka pada Left Join justru akan membuat sebuah parameter dari tabel sebelah kiri, dan jika ada data atau *record* yang kosong atau tidak berelasi akan berisi NULL di sebelah kanan.

2) Right Join

Secara mendasar, fungsi dari Right Join hampir sama dengan Inner Join, hanya pada Right Join akan membuat sebuah parameter pada sebelah kanan jika data pada table terdapat data atau *record* yang kosong atau tidak berelasi maka akan berisi NULL.



Gambar 1.7 Diagram venn untuk Right Join
Sumber: <https://jagowebdev.com/wp-content/uploads/2017/02/Illustrasi-Right-Outer-Join.png>

Bentuk umum Right Join adalah sebagai berikut:

```
SELECT * FROM table_1 RIGHT JOIN table_2 ON table_1.PK
= table_2.FK
SELECT * FROM POST RIGHT JOIN category on post.category_
id=category.id_category;
```

Perintah SQL untuk menggabungkan tabel customer1 dan dt_pesan menggunakan Right Join adalah sebagai berikut:

```
SELECT customer1.id_cst, customer1.nm_customer1,
dt_pesan.id_dt_pesan, dt_pesan.tglDt_pesan
FROM customer1 RIGHT JOIN dt_pesan
ON customer1.id_cst=dt_pesan.id_cst;
```

Hasilnya adalah sebagai berikut:

id_cst	nm_customer1	id_dt_pesan	tglDt_pesan
CST01	Nanik Ramini	1	2019-04-02
CST02	Zaskia Sari	2	2019-04-05
CST02	Zaskia Sari	3	2019-04-10
CST04	Rangga Mardi	4	2019-01-20
CST01	Nanik Ramini	5	2018-12-14

Tabel yang menjadi acuan pada Right Join adalah tabel sebelah kanan (tabel dt_pesan), sehingga semua isi tabel dt_pesan akan ditampilkan. Meskipun data customer1 tidak ada di tabel customer1, maka isi tabel dt_pesan tetap ditampilkan.

c. **Full Join**

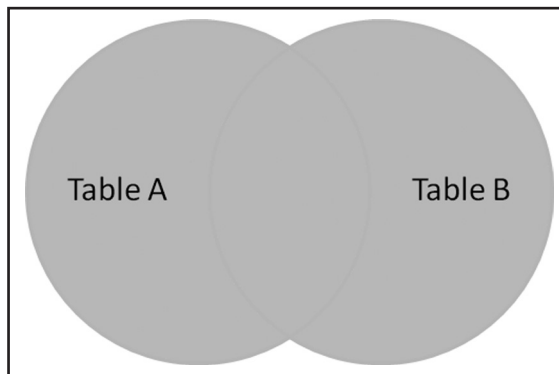
Penggunaan Full Join akan menghasilkan baris data jika ada data yang sama pada salah satu tabel. Bentuk umum Full Join adalah sebagai berikut:

```
SELECT column_name(s) FROM table_name1 FULL JOIN table_name2  
ON table_name1.column_name=table_name2.column_name
```

Perhatikan bentuk implementasi sebagai berikut:

```
SELECT reg.nama,reg.email,order.no_order FROM reg FULL JOIN  
order ON reg.id=order.id ORDER BY reg.nama
```

Adapun bentuk diagram venn untuk Full join adalah sebagai berikut:



Gambar 1.8 Diagram venn untuk Full Join
Sumber: <https://i.stack.imgur.com/3Ll1h.png>

d. **Implisit Join**

Sejauh ini, kita menampilkan data dari beberapa tabel MySQL menggunakan klausa JOIN. Selain menggunakan klausa JOIN, terdapat satu cara untuk menggabungkan tabel MySQL, yaitu menggunakan Implisit Join. Implisit JOIN merupakan metode lama ketika pertama kali standar SQL dibuat, setelah muncul standar yang lebih baru (SQL2) maka mulai digunakanlah klausa JOIN. klausa JOIN lebih mudah dibaca dan dipahami, terutama hubungan antara tabel yang digabungkan. Disebut Implisit Join karena tidak menggunakan klausa JOIN. Kriteria hubungan pada Implisit Join antartabel didefinisikan pada klausa WHERE. Misalnya menggabungkan tabel customer1 dan penjualan dengan menjalankan query berikut.

```
SELECT pl.id_customer1, nm_customer1, tgl_transaksi, total_  
transaksi  
FROM customer1 pl, penjualan pn  
WHERE pl.id_customer1 = pn.id_customer1
```

Hasil yang kita peroleh:

id_customer1	nm_customer1	tgl_transaksi	total_transaksi
CST01	Nanik Ramini	2018-06-22	230000
CST03	Herman Mario	2018-06-22	195000
CST02	Zaskia Sari	2018-06-01	1710000
CST01	Nanik Ramini	2018-06-04	310000

Perhatikan bahwa hasil tersebut sama persis dengan hasil pada contoh INNER JOIN, sehingga dapat disimpulkan bahwa Implisit Join = Inner Join. Implisit Join mensyaratkan kedua tabel memiliki data yang sama (WHERE pl.id_customer1 = pn.id_customer1), sehingga Implisit Join ini hanya berlaku pada INNER JOIN, dan tidak bisa digunakan untuk OUTER JOIN. Hubungan antartabel pada bentuk klausa JOIN dinyatakan pada klausa ON atau USING, sedangkan filter datanya dilakukan pada klausa WHERE. Misalnya sebagai berikut:

```
SELECT pl.id_customer1, nm_customer1, tgl_transaksi, total_transaksi
FROM customer1 pl
LEFT JOIN penjualan pn USING (id_customer1)
WHERE pl.id_customer1 = 2 OR pl.id_customer1 = 1
```

Sedangkan hubungan antartabel dan filter datanya pada Implisit JOIN didefinisikan pada klausa WHERE. misalnya sebagai berikut:

```
SELECT pl.id_customer1, nm_customer1, tgl_transaksi, total_transaksi
FROM customer1 pl, penjualan pn
WHERE pl.id_customer1 = pn.id_customer1 AND (pl.id_customer1 = 2 OR pl.id_customer1 = 1)
```

Aktivitas Siswa

Masing-masing siswa membuat beberapa pertanyaan tentang perintah SQL dalam akses *multitable* yang kurang dimengerti kemudian saling tukar dengan teman sebangku. Mintalah mereka untuk menjawabnya! Apabila mengalami kesulitan bertanyalah pada guru Anda!

Tugas Kelompok

Buatlah sebuah kelompok kemudian kumpulkan data/informasi tentang subquery basis data terutama yang berkaitan dengan implementasinya!

Lembar Kerja Siswa Berbasis STEM

1. Bacalah materi pelajaran tersebut di atas dan cari sumber-sumber lain yang relevan!
2. Buatlah karangan dengan tema “Replikasi Basis Data”!
3. Karangan ditulis pada kertas folio.
4. Panjang karangan kira-kira 1–2 halaman folio.
5. Kumpulkan pada guru mata pelajaran!

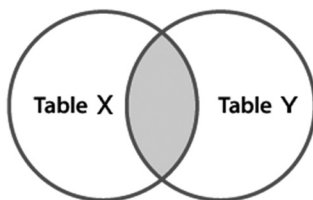
Rangkuman

1. Pada dasarnya, SELECT merupakan perintah DML (*Data Manipulation Language*) untuk menghasilkan *recordset* dengan mengekstrak data dari *database*. Perintah SELECT ini sering dipakai dan memiliki bentuk yang kompleks.
2. Select multiple table berfungsi untuk menampilkan data yang berhubungan dari dua buah tabel.
3. Di dalam suatu RDBMS menjadi suatu kewajiban jika dalam satu *database* dapat terdiri dari beberapa tabel, di mana masing-masing tabel tersebut berhubungan (relasi) satu sama lain.
4. Pada SQL terdapat sebuah perintah Join yang berfungsi untuk menghubungkan tabel yang satu dengan tabel yang lain yang saling berhubungan atau berelasi. Tujuannya adalah untuk menampilkan beberapa data dari tabel yang berbeda dengan menggunakan satu perintah.

Uji Kompetensi

A. Pilihlah jawaban yang tepat!

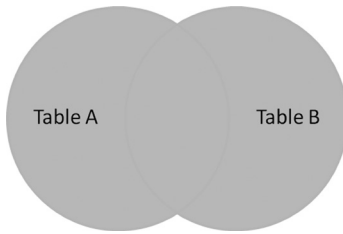
1. Perhatikan gambar berikut!



Nama diagram venn di atas diperuntukkan pada perintah

- a. Full Join
 - b. Half Join
 - c. Inner Join
 - d. Left Join
 - e. Right Join
2. Pada select multiple table dengan syntax from nama tabel siswa ditulis lebih dahulu dibandingkan nama table tb_kelas. Maka hasilnya adalah
- a. data *ascending* menurut tabel siswa
 - b. data *ascending* menurut tabel tb_kelas
 - c. data *descending* menurut tabel siswa
 - d. data *descending* menurut tabel tb_kelas
 - e. data *descending* syntax
3. Pada dasarnya, SELECT masuk ke dalam kategori perintah
- a. DML (*Data Manipulation Language*)
 - b. DDL (*Data Definition Language*)
 - c. DCL (*Data Control Language*)
 - d. CREATE
 - e. ALTER DROP
4. Jika ada data atau *record* yang kosong atau tidak berelasi akan berisi NULL pada prosedur Left Join, maka posisinya akan berada di sebelah
- a. tengah
 - b. kiri
 - c. kanan
 - d. bawah
 - e. atas
5. Sebuah perintah Pada SQL yang berfungsi untuk menghubungkan tabel yang satu dengan tabel yang lain yang saling berhubungan atau berelasi
- a. WHERE
 - b. Join
 - c. Prev
 - d. Compare
 - e. Category
6. Jika ingin menampilkan seluruh *field* pada tabel data diri dapat menggunakan tanda
- a. semicolon (;)
 - b. double apostrof
 - c. dot command
 - d. comma (,)
 - e. asterik (*)
7. Secara mendasar, data yang akan ditampilkan pada perintah Inner Join adalah data yang memiliki
- a. acak
 - b. bebas
 - c. tanpa pasangan
 - d. pasangan
 - e. terdata urut

8. Perhatikan gambar berikut!



Nama diagram venn di atas diperuntukkan pada perintah

- a. Full Join
 - b. Half Join
 - c. Inner Join
 - d. Left Join
 - e. Right Join
9. Perintah SQL yang benar untuk menampilkan khusus pada *field* tmpmt_lahir, tgl_lahir, dan nama_lengkap dari tabel data diri adalah
- a. delete tmpmt_lahir, tgl_lahir, nama_lengkap FROM datadiri
 - b. insert tmpmt_lahir, tgl_lahir, nama_lengkap FROM datadiri
 - c. select tmpmt_lahir, tgl_lahir, nama_lengkap FROM datadiri
 - d. skipper tmpmt_lahir, tgl_lahir, nama_lengkap FROM datadiri
 - e. update tmpmt_lahir, tgl_lahir, nama_lengkap FROM datadiri
10. Tabel yang digabungkan dua arah menggunakan Inner Join, hasilnya adalah tidak ada data yang bernilai
- a. CATEGORY
 - b. FROM
 - c. NULL
 - d. ON
 - e. POT


B. Selesaikan soal-soal berikut dengan tepat!

1. Kumpulan file, tabel, dan arsip yang saling berhubungan dan disimpan dalam media penyimpanan elektronis disebut
2. Hal yang sangat ditonjolkan dalam basis data adalah pengaturan, pemilahan, pengelompokkan, dan pengorganisasian data yang akan disimpan sesuai dengan
3. Basis data dikelola dan ditangani melalui perantaraan alat mesin pintar elektronis yang dikenal dengan nama
4. Untuk menampilkan beberapa data dari tabel yang berbeda dengan menggunakan satu perintah dapat menggunakan
5. Disebut Implisit Join karena tidak menggunakan
6. Kriteria hubungan pada Implisit Join antartabel didefinisikan pada
7. Hubungan antartabel pada bentuk klausa JOIN dinyatakan pada klausa
8. Jenis JOIN yang termasuk metode lama ketika pertama kali standar SQL adalah

9. Relasi antartabel dapat berupa relasi
10. Untuk menampilkan data yang berhubungan dari dua buah tabel dapat menggunakan

C. Jawablah pertanyaan berikut dengan jelas dan benar!

1. Sebutkan berbagai sudut pandang dari definisi basis data!
2. Jabarkan tentang penggabungan dengan WHERE dengan dengan Inner Join beserta contohnya!
3. Jelaskan yang Anda ketahui tentang definisi Left Join!
4. Uraikan yang Anda ketahui tentang Inner Join!
5. Terangkan hubungan antara perintah SELECT dengan perintah DML disertai contohnya!



Penerapan STEM dan Soal HOTS

1. Buatlah kelompok bersama teman sebangku Anda!
2. Bersama kelompok Anda, cari dan buatlah sebuah makalah dari internet yang memuat tentang perintah SQL dalam akses multitable! (STEM)
3. Presentasikan hasil kerja kelompok Anda di depan kelas dengan penuh percaya diri!
4. Setelah itu berikan kesempatan kepada kelompok lain untuk menanggapi!