

INVENTARIS PERPUSTAKAAN	
TANGGAL	NO. INDIK
6/2/2018	032018001

## MODUL PRAKTIKUM BASIS DATA



# polman astra



Disusun Oleh : Rida Indah Fariani, S.Si, M.T.I

**PMA-MP-INF- 07**

**PROGRAM STUDI MANAJEMEN INFORMATIKA**

**POLITEKNIK MANUFAKTUR ASTRA**

**2018**


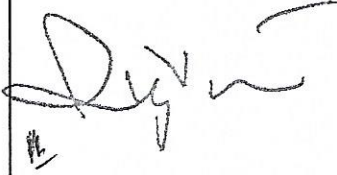



**polman astra**

## LEMBAR PENGESAHAN

### MODUL PRAKTIKUM BASIS DATA

No Dokumen	:	PMA-MP-INF-07
Revisi	:	00
Tanggal	:	2 Februari 2018
Dikaji Ulang oleh	:	Ketua Prodi MI
Dikendalikan oleh	:	Badan Pengendali Mutu

Disusun Oleh Dosen Basis Data	Diperiksa Oleh Ketua Program Studi MI	Disahkan Oleh Wakil Direktur I
 Rida Indah Fariani, S.Si,M.T.I	 Yongky L. Tantra Dipl. Ing	 Tonny Pongoh, S.H, LLM

## KATA PENGANTAR

Modul praktikum ini dibuat sebagai pedoman mahasiswa dalam melakukan kegiatan praktikum Basis Data di Politeknik Manufaktur Astra. Modul praktikum ini terdiri dari 8 modul praktikum yang membahas tentang *Relational Model, Data Definiton Language, Data Manipulation Language, Join Table, Entity Relationship Diagram*, dan normalisasi. Modul praktikum ini diharapkan dapat membantu mahasiswa/i dalam melaksanakan praktikum dengan lebih baik, terarah, dan terencana. Pada setiap topik telah ditetapkan tujuan pelaksanaan praktikum dan semua kegiatan yang harus dilakukan oleh mahasiswa/i serta teori singkat untuk memperdalam pemahaman mahasiswa/i mengenai materi yang dibahas.

Penyusun menyakini bahwa dalam pembuatan Modul Praktikum Basis Data ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan modul praktikum ini dimasa yang akan datang.

Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Jakarta, 18 Januari 2018

Rida Indah Fariani

# DAFTAR ISI

PENGESAHAN .....	<b>Error! Bookmark not defined.</b>
KATA PENGANTAR .....	i
DAFTAR ISI.....	ii
MODUL BASIS DATA – PRAKTIKUM 1 DASAR-DASAR DAN LINGKUNGAN BASIS DATA .....	1
1. Tujuan.....	1
2. Durasi Waktu .....	1
3. Dasar Teori .....	1
4. Peralatan .....	4
5. Percobaan.....	5
6. Latihan dan Evaluasi .....	5
7. Referensi .....	5
MODUL BASIS DATA – PRAKTIKUM 2 RELASIONAL MODEL .....	6
1. Tujuan.....	6
2. Durasi Waktu .....	6
3. Dasar Teori .....	6
4. Peralatan : .....	8
5. Percobaan.....	8
6. Latihan dan Evaluasi .....	11
7. Referensi .....	11
MODUL BASIS DATA – PRAKTIKUM 3 DATA DEFINITION LANGUAGE (DDL) .....	12
1. Tujuan.....	12

2.	Durasi Waktu .....	12
3.	Dasar Teori .....	12
4.	Peralatan : .....	14
5.	Percobaan.....	14
6.	Latihan dan Evaluasi .....	24
7.	Referensi.....	24
MODUL BASIS DATA – PRAKTIKUM 4 DATA MANIPULATION LANGUAGE (DML).....		
1.	Tujuan.....	25
2.	Durasi Waktu .....	25
3.	Dasar Teori .....	25
4.	Peralatan : .....	27
5.	Percobaan.....	27
6.	Latihan dan Evaluasi .....	29
7.	Referensi.....	29
MODUL BASIS DATA – PRAKTIKUM 5 DATA MANUPULATION LANGUAGE (DML II) .....		
1.	Tujuan.....	30
2.	Durasi Waktu .....	30
3.	Dasar Teori .....	30
4.	Peralatan : .....	32
5.	Percobaan.....	32
6.	Latihan dan Evaluasi .....	36
7.	Referensi.....	36
MODUL BASIS DATA – PRAKTIKUM 6 DATA MANIPULATION LANGUAGE (DML III) & JOIN TABEL .....		
		37

1.	Tujuan.....	37
2.	Durasi Waktu.....	37
3.	Dasar Teori.....	37
4.	Peralatan : .....	39
5.	Percobaan.....	39
6.	Latihan dan Evaluasi .....	39
7.	Referensi.....	41
MODUL BASIS DATA – PRAKTIKUM 7 ENTITY RELATIONSHIP DIAGRAM (ERD).....		42
1.	Tujuan.....	42
2.	Durasi Waktu.....	42
3.	Dasar Teori.....	42
4.	Peralatan : .....	48
5.	Percobaan.....	48
6.	Latihan dan Evaluasi .....	48
7.	Referensi.....	54
MODUL BASIS DATA – PRAKTIKUM 8 NORMALISASI .....		55
1.	Tujuan.....	55
2.	Durasi Waktu.....	55
3.	Dasar Teori.....	55
4.	Peralatan : .....	58
5.	Percobaan.....	58
6.	Latihan dan Evaluasi .....	58
7.	Referensi.....	60

# **MODUL BASIS DATA – PRAKTIKUM 1**

## **DASAR-DASAR DAN LINGKUNGAN BASIS DATA**

### **1. Tujuan**

Tujuan Instruksional Umum :

Mampu menjelaskan mengenai dasar-dasar Basis Data dan Lingkungan Basis Data

Tujuan Instruksional Khusus :

1. Mampu menjelaskan pentingnya basis data
2. Mampu menjelaskan pengertian basis data
3. Mampu menjelaskan perbedaaan proses file tradisional dengan basis data
4. Mampu mengidentifikasi adanya *redundancy* dan duplikasi
5. Mampu menjelaskan konsep dasar basis data
6. Mampu memberikan contoh keuntungan dan resiko pendekatan basis data

### **2. Durasi Waktu**

2 x 55 menit

### **3. Dasar Teori**

Pentingnya Data dalam Sistem Informasi

Sistem Informasi adalah pengelolaan Data, Orang/Pengguna, Proses dan Teknologi Informasi yang berinteraksi untuk mengumpulkan, memproses, menyimpan, dan menyediakan sebagai output informasi yang diperlukan untuk mendukung sebuah organisasi. (Jeffery L. Whitten dkk, 2004)

Menurut Stephens & Plew (2000)

Basis Data adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basis data pengguna dapat menyimpan data secara terorganisasi.

Setelah data disimpan, informasi harus mudah diambil. Cara data disimpan dalam basis data menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam basis data, dimodifikasi, dan dihapus.

Menurut Silberschatz, dkk (2002)

Basis data adalah kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan.

Menurut McLeod, dkk (2001)

Basis data adalah kumpulan seluruh sumber daya berbasis komputer milik organisasi.

Menurut Ramakrishnan dan Gehrke (2003)

Basis data adalah kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang saling berhubungan.

Basis data universitas mungkin berisi informasi mengenai hal berikut :

1. Entitas seperti mahasiswa, fakultas, mata kuliah, dan ruang kuliah.
2. Hubungan antar entitas seperti registrasi mahasiswa dalam matakuliah, fakultas yang mengajarkan matakuliah dan pengguna ruang kuliah.

File tradisional

1. Suatu aplikasi terdiri atas sekumpulan program aplikasi, file data, dan prosedur yang mengerjakan suatu proses atau fungsi.
2. Setiap program aplikasi di dalam suatu lingkungan pemrosesan file tradisional, khusus beroperasi pada file data yang dibuat spesifik untuk aplikasi itu.



3. Antar file data (di dalam satu aplikasi atau antar aplikasi) tidak ada hubungan, dan pada umumnya data didefinisikan dan disusun dengan cara yang berbeda untuk setiap aplikasi.

#### Keterbatasan Pemrosesan File Tradisional

1. Data menjadi terpisah dan terisolasi, karena antar file data tidak terhubung.
2. Munculnya redundansi data, yang tidak dapat dihindarkan karena setiap aplikasi mempunyai file data masing-masing.
3. Berpotensi terjadinya inkonsistensi data, yaitu jika dilakukan modifikasi data di suatu file akan tetapi di file yang lain (yang berisi data yang sama dengan data yang dimodifikasi) tidak dilakukan hal yang sama.
4. Munculnya data yang membingungkan (data confusion), yaitu apabila data yang sama disajikan dengan terminologi yang berbeda.
5. Program aplikasi tergantung pada format file (program-datadependence), yaitu kapan saja format data berubah maka seluruh program yang menggunakan data tersebut harus dimodifikasi.
6. Sulit untuk menyajikan objek data yang kompleks.

#### Redudansi data

1. Penyimpanan data yang sama secara berulang.

NIM	KODE_MK	SKS	NILAI
A10	MK_01	3	A
A10	MK_02	2	B
A11	MK_01	3	A
A12	MK_01	3	A
A12	MK_02	2	B
A12	MK_03	3	B

NIM	NIP_WALI		NIP_WALI	NAMA_WALI
A10	ADN_011		ADN_011	MAX
A11	AND_012		AND_012	ROBERT

Tabel a)

Tabel b)

NAMA_WALI	TELP_WALI		NIM	TELP_WALI
MAX	(024)001		A10	(024)001
ROBERT	(024)002		A11	(024)002

Tabel c)

Tabel d)

Nim	Nama_Mhs		Kd_MK	Nm_MK	SKS
A10	MAX		MK_01	Pancasila	2
A11	ROBERT		MK_02	Agama	2

Tabel Mahasiswa

Tabel Matakuliah

Nim	Nama_Mhs	Nm_MK	SKS	Kd_MK	Nilai
A10	MAX	Pancasila	2	MK_01	A
A11	ROBERT	Agama	2	MK_02	B

Tabel Nilai

1. Menyebabkan masalah pada waktu memperbarui (update) data.
2. Ruang penyimpanan yang boros.
3. Dapat menimbulkan tidak konsistennya data

#### 4. Peralatan

1. SQL Server Management Studio

## **5. Percobaan**

1. Sebutkan minimal 4 instansi pemerintahan yang menurut Anda menggunakan sistem basis data.
2. Dari 4 instansi tersebut, pilih salah satu untuk Anda jelaskan :
  - a. Mengapa memerlukan sistem basis data
  - b. Contoh jenis data apa yang digunakan dalam sistem basis data dalam instansi tersebut
3. Jelaskan peran DBMS dalam sistem basis data.
4. Jelaskan 3 level abstraksi data dan beri contoh.
5. Kumpulkan tugas Anda melalui folder yang disediakan dengan ketentuan :
  - a. Nama File : SI241P02XXX (XXX adalah 3 digit terakhir NIM Anda)
  - b. File dalam bentuk dokumen word

## **6. Latihan dan Evaluasi**

-

## **7. Referensi**

1. Database system, A Practical Approach to Design, Implementation, and Management; Thomas Connolly, Carolyn Bagg, Pearson 6<sup>th</sup> edition, 2015

## **MODUL BASIS DATA – PRAKTIKUM 2 RELASIONAL MODEL**

### **1. Tujuan**

Tujuan Instruksional Umum :

Mampu menjelaskan Model Relational

Tujuan Instruksional Khusus :

1. Mampu menjelaskan Terminologi Relasional Model
2. Mampu mengetahui bagaimana tabel digunakan untuk mempresentasikan data
3. Mampu mengetahui property dari hubungan basis data
4. Mampu mengidentifikasikan CK, PK, dan FK.
5. Mampu mengetahui arti integrity constraint

### **2. Durasi Waktu**

3 x 55 menit

### **3. Dasar Teori**

Relational Model Terminology

**A relation** is a table with columns and rows. Only applies to logical structure of the database, not the physical structure.

**Attribute** is a named column of a relation.

**Domain** is the set of allowable values for one or more attributes.

**Tuple** is a row of a relation

**Degree** is the number of attributes in a relation.

**Cardinality** is the number of tuples in a relation.

**Relational Database** is a collection of normalized relations with distinct relation names.

#### Database Relations

1. Relation schema is named relation defined by a set of attribute and domain name pairs.
2. Relational database schema is set of relation shcemas, each with a distinct name.

#### Properties of Relations/Table

1. Relation name is distinct from all other relation names in relational schema.
2. Each cell of relation contains exactly one atomic (single) value.
3. Each attribute has a distinct name.
4. Values of an attribute are all from the same domain
5. Each tuple/row is distinct; there are no duplicate tuple/row.
6. Order of attributes has no significance.
7. Order of tuples has no significance, theoretically.

#### Relational Keys

1. Superkey  
An attribute, or set of attributes, that uniquely identifies a tuple within a relation.
2. Candidate Key  
Superkey (K) such that no proper subset is a superkey within the relation. In each tuple of Relation (R), values of K uniquely identify that tuple (uniqueness). No proper subset of K has the uniqueness property (irreducibility).
3. Primary Key  
Candidate key selected to identify tuples uniquely within relation.
4. Alternate Key
5. Candidate keys that are not selected to be primary key.

## 6. Foreign Key

Attribute, or set of attributes, within one relation that matches candidate key of some (possibly same) relation.

## Integrity Constraints

### 1. Null

Represents value for an attribute that is currently unknown or not applicable for tuple. Deals with incomplete or exceptional data. Represents the absence of a value and is not the same as zero or spaces, which are values.

### 2. Entity Integrity

In a base relation, no attribute of a primary key can be null.

### 3. Referential Integrity

If foreign key exists in a relation, either foreign key value must match a candidate key value of some tuple in its home relation or foreign key value must be wholly null.

### 4. General Constraint

Additional rules specified by users or database administrators that define or constrain some aspect of the enterprise.

## 4. Peralatan :

1. SQL Server Management Studio

## 5. Percobaan

### PERCOBAAN 1 : Menentukan CK, PK, dan FK

Di bawah ini terdapat beberapa *table* DBMS relasional:

1. tbl\_mahasiswa (nama, alamat, no\_telp, nim, tgl\_lahir, tmp\_lahir, status)

2. tbl\_dosen (nama, alamat, no\_telp, tgl\_lahir, tmp\_lahir, id\_dosen, pendidikan\_terakhir, status)
  3. tbl\_mata\_kuliah (desk\_matkul, jum\_sks, id\_matkul)
  4. tbl\_ambil\_matkul (nim, kd\_matkul, kd\_dosen, tgl, kelas)
1. Tentukan *candidate key* dari *table* di atas!
  2. Tentukan *primary key* dan *foreign key* dari *table* diatas!
  3. Berikan contoh isi dari masing-masing *table* di atas, dengan mengikuti aturan *referential integrity*!

## PERCOBAAN 2 : Studi Kasus

### Mine PET Shop

Mine PET Shop merupakan salon dan *grooming pet shop*, yang didirikan dengan konsep menawarkan berbagai perawatan bagi hewan peliharaan seperti anjing, kucing, *hamster*, *reptile*, kelinci dan sebagainya. Selama dua tahun berdiri, kegiatan operasional Mine PET Shop masih *manual*, sehingga pemilik merasa ingin mengembangkan sistem yang terkomputerisasi yang membantu kegiatan operasionalnya.

Proses bisnis Mine PET Shop dimulai ketika, seseorang pengunjung mendatangi Mine PET Shop akan dilayani oleh *Customer Service* (CS). Apabila hewan peliharaan yang dibawa oleh pengunjung belum terdaftar, maka akan didaftarkan dahulu oleh CS. Informasi yang harus dimiliki oleh hewan tersebut, diantaranya: nama, umur, warna, jenis hewan, no hewan.

Setelah itu, CS akan menjelaskan mengenai paket perawatan (deskripsi, lama perawatan, harga, no perawatan) yang ada pada Mine PET Shop, seperti spa, pembersihan tubuh serta perapihan bulu hewan tersebut. Namun sebelum melakukan

perawatan, hewan tersebut harus melakukan *medical check* terlebih dahulu. *Medical check* ini merupakan layanan yang disediakan Mine PET Shop secara gratis, untuk setiap hewan yang berkunjung. Bagian Kesehatan akan membuat Lembar Hasil Pengecekan (LHP) yang diberikan kepada pengunjung. Informasi yang terlampir dalam LHP tersebut adalah mengenai kondisi hewan (id pemeriksaan, tanggal pemeriksaan, id hewan, hasil pemeriksaan) yang melakukan kunjungan, serta daftar paket perawatan apa saja yang diijinkan untuk diambil oleh pengunjung bagi hewan peliharaannya.

Kemudian, pengunjung akan melakukan pemesanan perawatan pada bagian CS. Pengunjung juga bisa memilih *Groomer* (id groomer, nama groomer) yang ada pada Mine PET Shop, yang kemudian akan dicatat oleh CS ke dalam Form Pemesanan berdasarkan pilihan perawatan pengunjung. Dalam Form Pemesanan tersebut, pengunjung boleh memilih lebih dari satu paket perawatan. Informasi yang terdapat di dalam Form tersebut, yaitu id pesanan, id pemeriksaan, id hewan, id perawatan, id groomer, keterangan pesanan. *Groomer* yang menerima panggilan dalam pemesanan tersebut segera melaksanakan tugasnya dalam perawatan hewan tersebut. Form Pemesanan tersebut kemudian diberikan kepada pengunjung.

Untuk melakukan pembayaran, pengunjung akan mendatangi bagian CS, kemudian CS akan membuatkan Bukti Pembayaran berdasarkan Form Pemesanan pengunjung. Informasi yang terdapat dalam Bukti Pembayaran, yaitu id pembayaran, id pesanan, id CS, jumlah pembayaran. Setiap akhir bulan, CS akan membuat Laporan Pemesanan yang berisi jumlah pemesanan pada bulan tersebut. Laporan tersebut dibuat berdasarkan Form Pemesanan yang dicatat bagian CS dan diberikan kepada *owner*.

1. Gambarkan *table* dengan atributnya lengkap yang dapat muncul, dari deskripsi di atas!
2. Tentukan *super key*, *candidate key* dari deskripsi di atas!
3. Tentukan *primary key* dan *foreign key* dari deskripsi di atas!



4. Berikan contoh isi masing-masing *table* dari deskripsi di atas, dengan mengikuti aturan *referential integrity*!
5. Berikan contoh dari *table* yang Anda buat, perbedaan antara domain, tuple, dan atribut.
6. Buatlah *view* untuk melihat Laporan Pemesanan yang dibuat oleh CS!

## **6. Latihan dan Evaluasi**

-

## **7. Referensi**

1. Database system, A Practical Approach to Design, Implementation, and Management; Thomas Connolly, Carolyn Bagg, Pearson 6<sup>th</sup> edition, 2015

## **MODUL BASIS DATA – PRAKTIKUM 3**

### **DATA DEFINITION LANGUAGE (DDL)**

#### **1. Tujuan**

Tujuan Instruksional Umum :

Mampu menjelaskan dan menguasai SQL.

Tujuan Instruksional Khusus :

1. Mampu mengetahui tipe data yang didukung oleh standar SQL
2. Mampu mengetahui tujuan dari fitur SQL
3. Mampu menjelaskan bagaimana integrity constraint menggunakan SQL
4. Mampu memahami CREATE dan ALTER TABLE *statements*

#### **2. Durasi Waktu**

2 x 55 menit

#### **3. Dasar Teori**

Tipe Data pada SQL

Common data types	SQL Server system-supplied data types	ANSI synonym	Number of bytes
Integer	<b>int</b>	<i>integer</i>	4
	<b>bigint</b>	–	8
	<b>smallint, tinyint</b>	–	2, 1
Exact numeric	<b>decimal[(p[, s])]</b>	<i>dec</i>	2–17
	<b>numeric[(p[, s])]</b>	–	
Approximate numeric	<b>float[(n)]</b>	<i>double precision,</i> <i>float[(n)] for n=8-15</i>	8
	<b>real</b>	<i>float[(n)] for n=1-7</i>	4
Monetary	<b>money, smallmoney</b>	–	8, 4
Date and time	<b>Datetime,</b>	–	8
	<b>smalldatetime</b>		4

Common data types	SQL Server system-supplied data types	ANSI synonym	Number of bytes		
Character	<b>char[(n)]</b>	<i>character[(n)]</i>	0–8000		
	<b>varchar[(n)]</b>	<i>char VARYING[(n)]</i>	0–2 GB		
	<b>text</b>	<i>character VARYING[(n)]</i>			
	–	–			
Unicode character	<b>nchar[(n)]</b>	–	0–8000		
	<b>nvarchar[(n)]</b>	–	(4000 characters)		
	<b>ntext</b>	–	0–2 GB		
Binary	<b>binary[(n)]</b>	–	0–8000		
	<b>varbinary[(n)]</b>	<i>binary VARYING[(n)]</i>			
Image	<b>image</b>	–	0–2 GB		
Global identifier	<b>uniqueidentifier</b>	–	16		
Special	<b>bit, cursor,</b>	–	1, 0–8		
	<b>uniqueidentifier</b>				
	<b>timestamp</b>			<b>rowversion</b>	8
	<b>sysname</b>			–	256
	<b>table</b>			–	0–8016
	<b>sql_variant</b>			–	

## Data Integrity

### 1. Domain Integrity

2. Entity Integrity
3. Refential Integrity

#### **4. Peralatan :**

1. SQL Server Management Studio

#### **5. Percobaan**

Buatlah basis data yang bernama DreamHome dalam skema Anda. Dalam basisdata DreamHome terdapat 6 tabel yaitu tabel Cabang, Staff, Client, Property, Pemilik dan Registrasi. Lakukan langkah-langkah berikut :

1. Gunakan DDL untuk semua latihan Anda.
2. Buatlah basis data DreamHome.
3. Buatlah keenam table yang terdapat pada basis data DreamHome. Perhatikan keterhubungan dalam relasi atau table tersebut. Tabel yang dijadikan referensi table lain harus dibuat terlebih dahulu.
4. Berikut definisi data dari keenam table pada basis data DreamHome :

#### **Tabel Cabang**

#### **Skema Relasi**

Cabang (NoCabang, Jalan, Area, Kota, KodePos, Telepon, Fax)

#### **Primary Key**

NoCabang

#### **Atribut :**

<b>Nama Kolom</b>	<b>Tipe Data dan Ukuran</b>	<b>Domain</b>	<b>Harus ada isi</b>	<b>Nilai Default</b>
<u>NoCabang</u>	3 variable character		Y	-
Jalan	25 variable character		Y	-
Area	15 variable character		T	-
Kota	15 variable character		Y	-
KodePos	8 variable character		T	-
Telepone	13 variable character		T	-
Fax	13 variable character		T	-

### **Tabel Staff**

#### **Skema Relasi**

Staff (NoStaff, Nama1, Nama2, Posisi, Sex, TglLahir, Gaji, NoCabang)

#### **Primary Key**

NoStaff

#### **Foreign Key**

NoCabang references Cabang (NoCabang) on delete NO ACTION on update CASCADE

#### **Atribut**

<b>Nama Kolom</b>	<b>Tipe Data dan Ukuran</b>	<b>Domain</b>	<b>Harus ada isi</b>	<b>Nilai Default</b>
NoStaff	5 variable character		Y	-
Nama1	20 variable character		Y	-

Nama2	20 variable character		Y	-
Posisi	20 variable character	Nilai yang diperbolehkan : 'Manager', 'Supervisor', 'Deputy', 'Assistant', 'Secretary'	Y	-
Sex	1 fixed character	Nilai yang diperbolehkan 'L' atau 'P'	Y	-
TglLahir	Date		T	-
Gaji	Currency		Y	-
NoCabang	3 variable character	Seperti pada NoCabang	Y	-

### **Tabel Client**

#### **Skema Relasi**

Client (NoClient, Nama1, Nama2, , Telepon, Type, MaxRent, NoCabang)

#### **Primary Key**

NoClient

#### **Foreign Key**

NoCabang references Cabang (NoCabang) on delete NO ACTION on update CASCADE

#### **Atribut**

<b>Nama Kolom</b>	<b>Tipe Data dan Ukuran</b>	<b>Domain</b>	<b>Harus ada isi</b>	<b>Nilai Default</b>
NoClient	5 variable character		Y	-
Nama1	20 variable character		Y	-
Nama2	20 variable character		Y	-
Telepon	13 variable character		T	-
Type	1 fixed character	Nilai yang diperbolehkan: B (bungalow), C (cottage), D (detached), E (end-terrace), F (flat) M (mid-terrace), S (semi-detached)	T	-
MaxRent	Currency		T	-
NoCabang	3 variable character		Y	-

### **Tabel Pemilik**

#### **Relation schema**

Pemilik (NoMilik, Nama1, Nama2, Alamat, Telepon)

#### **Primary Key**

NoMilik

#### **Atribut**

<b>Nama Kolom</b>	<b>Tipe Data dan Ukuran</b>	<b>Domain</b>	<b>Harus ada isi</b>	<b>Nilai Default</b>
NoMilik	5 variable character		Y	-
Nama1	20 variable character		Y	-
Nama2	20 variable character		Y	-
Alamat	50 variable character		Y	-
Telepon	13 variable character		T	-

### **Tabel Property**

#### **Skema Relasi**

Properti (NoProperty, Jalan, Kota, KodePos, Type,Ruang, Pinjam, NoMilik, , NoStaff, NoCabang)

#### **Primary Key**

NoProperty

#### **Foreign Keys**

NoMilik references Pemilik (NoMilik on delete NO ACTION on update CASCADE)

NoStaff references Staff (NoStaff) on delete SET NULL on update CASCADE

NoCabang references Cabang (NoCabang) on delete SET DEFAULT on update CASCADE

#### **Atribut:**

<b>Nama Kolom</b>	<b>Tipe Data dan Ukuran</b>	<b>Domain</b>	<b>Harus ada isi</b>	<b>Nilai Default</b>
NoProperty	5 variable character		Y	



<b>Nama Kolom</b>	<b>Tipe Data dan Ukuran</b>	<b>Domain</b>	<b>Harus ada isi</b>	<b>Nilai Default</b>
Jalan	25 variable character		Y	
Kota	15 variable character		Y	
KodePos	8 variable character		T	
Type	1 Fixed Character	Nilai yang diperbolehkan: B (bungalow), C (cottage), D (detached), E (end-terrace), F (flat) M (mid-terrace), S (semi-detached)		
Ruang	Integer	Nilai yang diperbolehkan 1-15	T	4
Pinjam	Currency		T	600
NoMilik	5 variable character	Seperti NoMilik pada tabel Pemilik	T	
NoStaff	5 variable character	Seperti NoStaff pada tabel Staff	T	
NoCabang	3 variable character	Seperti NoCabang pada tabel Cabang	Y	

### View

#### Skema Relasi

View (NoClient, NoProperty, TglLihat, Komentar)

### **Primary Key**

NoClient, NoProperty, TglLihat

### **Foreign Key**

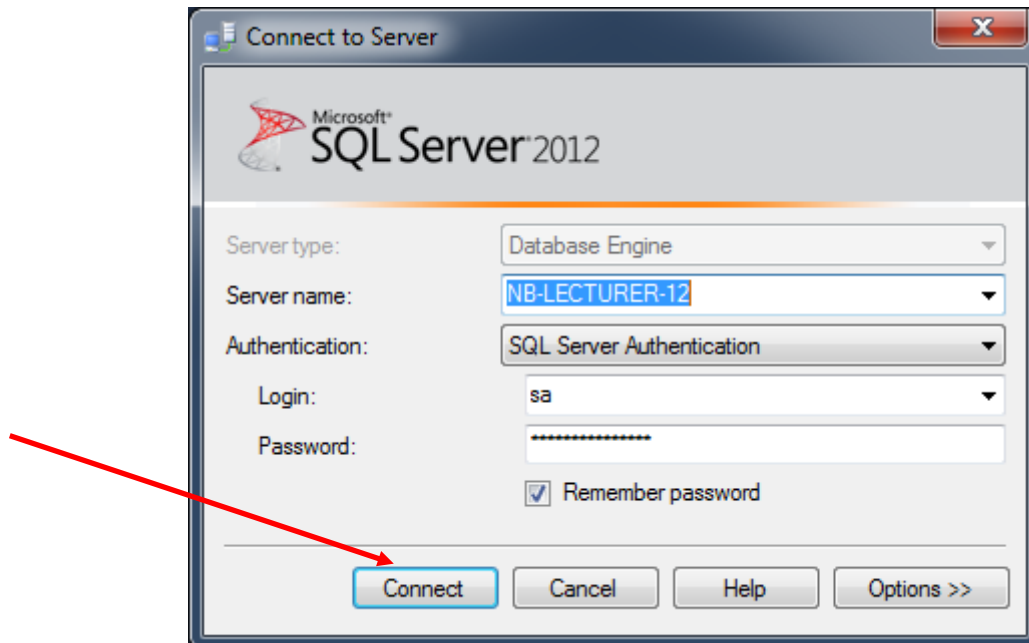
NoProperty references Property (NoProperty) on delete CASCADE on update CASCADE

NoClient references Client (NoClient) on delete CASCADE on update CASCADE

### **Atribut**

<b>Nama Kolom</b>	<b>Tipe Data dan Ukuran</b>	<b>Domain</b>	<b>Harus ada isi</b>	<b>Nilai Default</b>
NoProperty	5 variable character	Seperti didefinisikan di table Property	Y	
NoClient	5 variable character	Seperti didefinisikan di table Client	Y	
TglLihat	Date		Y	
Komentar	50 variable character		T	

5. Eksekusi DDL anda dalam SQL Server. Berikut contoh penggunaan statement SQL dalam SQL server :
  - a. Bukalah SQL Server Management Studio dalam komputer Anda
  - b. Lakukan koneksi ke server basis data dengan cara menekan tombol Connect :

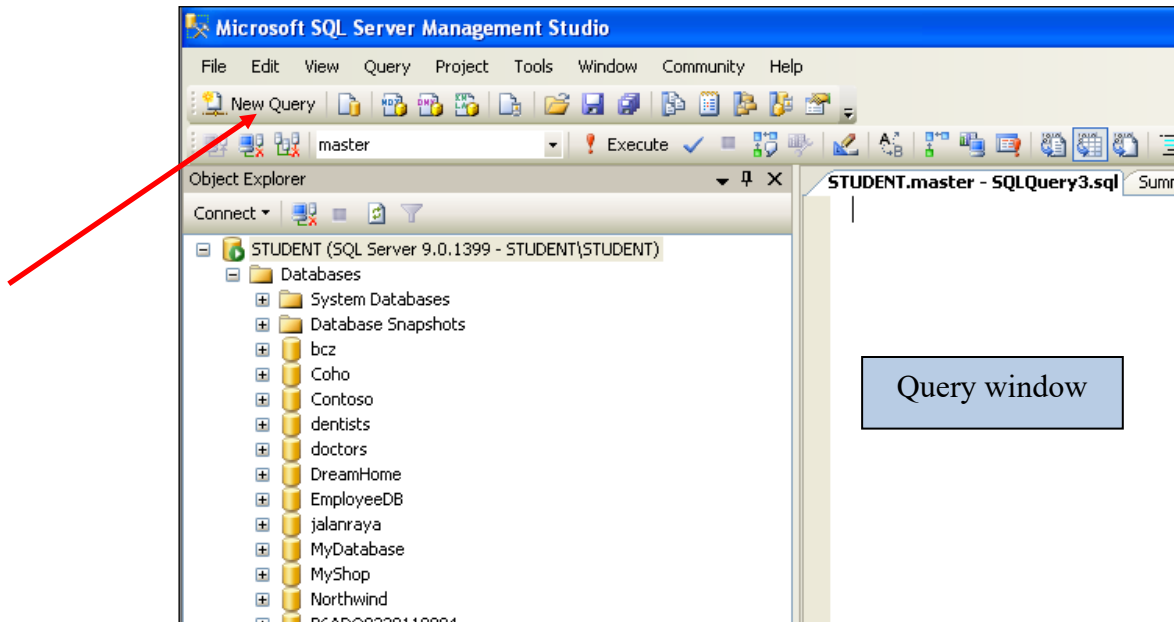


**Note**

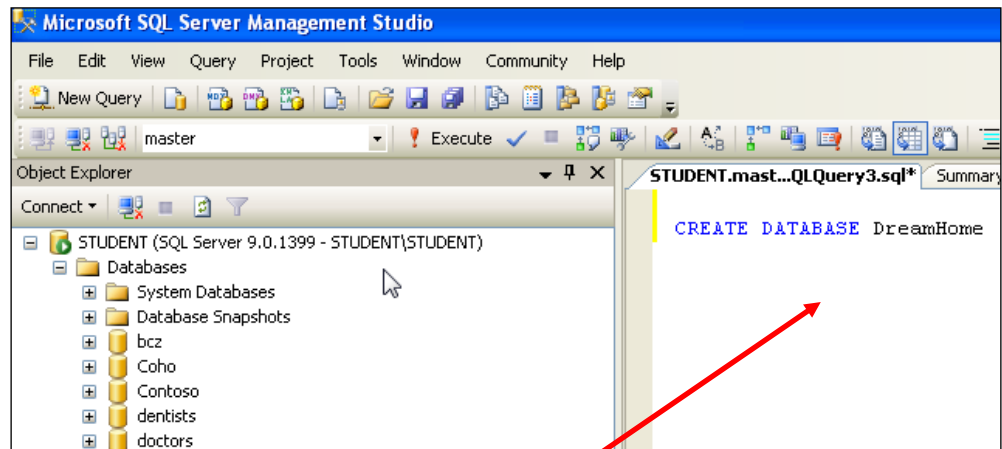
:

Pada komputer Anda bisa saja tampilan berbeda dalam hal authentication dapat menggunakan cara lain yaitu windows authentication (tanpa menggunakan login dan password).

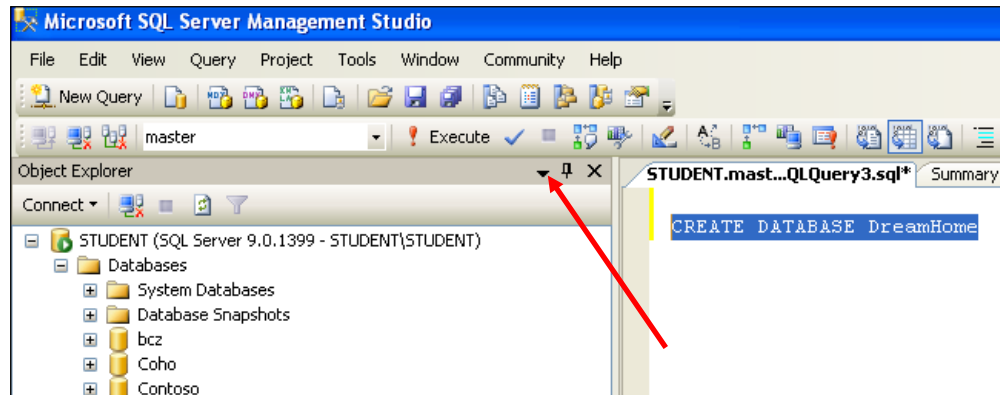
- c. Pilih New Query untuk membuka windows tempat Anda mengeksekusi SQL statements



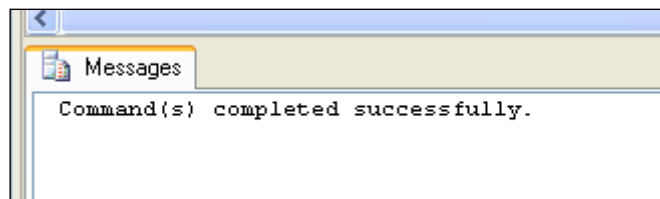
- d. Tuliskan statemen-statenen SQL dalam query window. Sebagai contoh jika akan membuat basis data DreamHome maka statemen SQL yang ditulis adalah : CREATE DATABASE DreamHome. Jika Anda sudah mempunyai file sql sendiri (\*.sql), file dapat langsung dibuka di sql window dan dieksekusi.



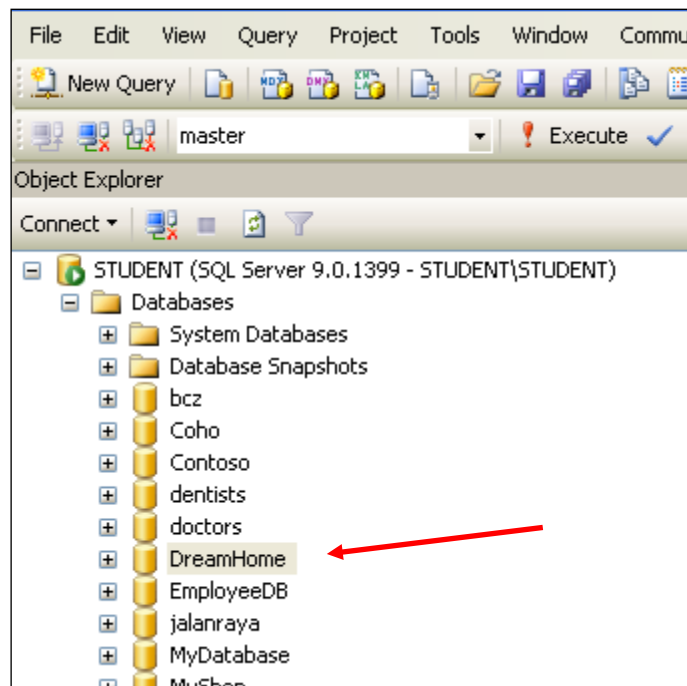
- e. *Block* statement yang akan di eksekusi. Kemudian pilih execute



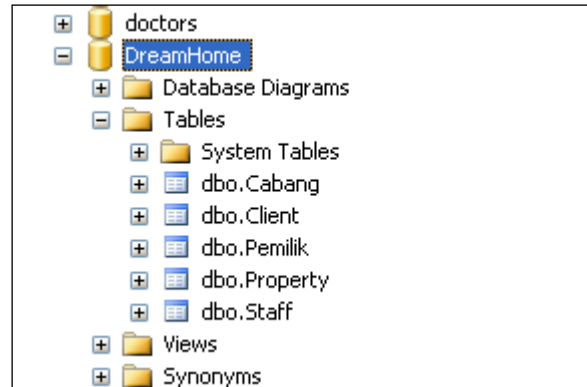
f. Jika berhasil, akan ada pemberitahuan pada bagian bawah query window :



g. Anda sudah membuat skema basis data baru yaitu DreamHome. Pastikan basis data Anda muncul pada list basis data. Jika belum muncul refresh skema Databases Anda dan lihat kembali.



- h. Buatlah statemen SQL untuk membuat keenam tabel sesuai definisi data yang telah diberikan. Eksekusi statemen SQL tersebut dan periksalah tabel Anda.



Simpan hasil kerja Anda dalam satu file sql.

Catatan :

Pembuatan basis data maupun tabel dapat menggunakan interface dari SQL Server berupa wizard dan DDL dapat dibangkitkan. Untuk latihan pada modul ini, tidak diperkenankan menggunakan wizard, Anda diminta membuat DDL sendiri berdasarkan perkuliahan sebelumnya.

## 6. Latihan dan Evaluasi

-

## 7. Referensi

1. Database system, A Practical Approach to Design, Implementation, and Management; Thomas Connolly, Carolyn Bagg, Pearson 6<sup>th</sup> edition, 2015

## **MODUL BASIS DATA – PRAKTIKUM 4**

### **DATA MANIPULATION LANGUAGE (DML)**

#### **1. Tujuan**

Tujuan Instruksional Umum :

Mampu menguasai SQL Query.

Tujuan Instruksional Khusus :

1. Mampu menguasai perintah SELECT, ORDER BY, GROUP BY, HAVING, UNION, INTERSECT, EXCEPT, EXIST, INSERT, UPDATE, DELETE
2. Mampu menguasai Fungsi Agregat
3. Mampu menguasai Sub Query

#### **2. Durasi Waktu**

8 x 55 menit

#### **3. Dasar Teori**

Insert Data

```
INSERT INTO TableName [ (columnList) ]  
VALUES (dataValueList)
```

Keterangan:

1. columnList is optional. If omitted, SQL assumes a list of all columns in their original CREATETABLE order.
2. Any columns omitted must have been declared as NULL when table was created, unless DEFAULT was specified when creating column.
3. dataValueList must match columnList as follows:

- number of items in each list must be same;
- must be direct correspondence in position of items in two lists;
- data type of each item in dataValueList must be compatible with data type of corresponding column.

#### Update Data

```
UPDATE TableName
SET columnName1 = dataValue1
    [, columnName2 = dataValue2...]
[WHERE searchCondition]
```

#### Keterangan:

1. TableName can be name of a base table or an updatable view.
2. SET clause specifies names of one or more columns that are to be updated.
3. WHERE clause is optional:
  - if omitted, named columns are updated for all rows in table;
  - if specified, only those rows that satisfy searchCondition are updated.
4. New dataValue(s) must be compatible with data type for corresponding column.

#### Delete Data

```
DELETE FROM TableName
[WHERE searchCondition]
```

#### Keterangan:

5. TableName can be name of a base table or an updatable view.
6. searchCondition is optional; if omitted, all rows are deleted from table. This does not delete table. If search\_condition is specified, only those rows that satisfy condition are deleted.



ISO standard defines five aggregate functions :

<b>COUNT</b>	returns number of values in specified column.
<b>SUM</b>	returns sum of values in specified column.
<b>AVG</b>	returns average of values in specified column.
<b>MIN</b>	returns smallest value in specified column.
<b>MAX</b>	returns largest value in specified column.

#### 4. Peralatan :

1. SQL Server Management Studio

#### 5. Percobaan

##### INSERT DATA

Gunakan lah basis data yang telah Anda buat pada modul 3 mengenai DDL. Isilah basis data tersebut dengan menggunakan perintah INSERT dengan data-data berikut :

1. Tabel Cabang

NoCabang	Jalan	Area	Kota	KodePos	Telepon	Fax
B2	Jl. Cempaka	A	Cempaka Putih	14330	0181-963-1030	0181-453-7992
B3	Jl. Biduri Bulan VIII	A	Kemayoran	13880	0141-339-2178	0141-339-4439
B4	Jl. Pagelarang III	C	Cipayung	10650	0117-916-1170	0117-776-1114
B5	Jl. Cibeuti 29	B	Cijantung	11115	0171-886-1212	0171-886-1214
B7	Jl. Panglayungan 65	D	Cimanggis	16543	01224-67125	01224-67111

2. Tabel Staff

NoStaff	Nama1	Nama2	Posisi	Sex	TglLahir	Gaji	NoCabang
SA9	Yanti	Ursula	Asisten	P	19-Feb-70	9,000.00	B7
SG14	Kenzie	Syazani	Deputi	L	24-Mar-58	18,000.00	B3
SG37	Budi	Sitepu	Supervisor	L	10-Nov-60	12,000.00	B3
SG5	Susan	Malika	Manager	P	03-Jun-40	24,000.00	B3
SL21	Daffa	Sulthan	Manager	P	01-Oct-45	30,000.00	B5
SL41	Arif	Kayana	Asisten	P	13-Jun-65	9,000.00	B5

Cobalah untuk insert data berikut :

SA10	Cintia	Maramis	Staff	P	20-Jan-1981	10.000.000	B7
------	--------	---------	-------	---	-------------	------------	----

Apakah terjadi error? Mengapa ?

### 3. Tabel Client

NoClient	Nama1	Nama2	Telepon	Type	MaxRent	NoCabang
CR56	Viona	Ajeng	0141-848-1825	F	350.00	B3
CR62	Laura	Saras	01224-196720	F	600.00	B7
CR74	Bagus	Richi	01475-392178	C	750.00	B3
CR76	Jalu	Kay	0171-774-5632	F	425.00	B5

### 4. Tabel Pemilik

NoMilik	Nama1	Nama2	Alamat	Telepon
CO40	Tina	Sonarti	Perum Mitra Batik Blok C No 9	0141-943-1728
CO46	Junaedi	Husen	Jl Kelapa Gading Raya No 11	01224-861212

NoMilik	Nama1	Nama2	Alamat	Telepon
CO87	Cicil	Susanti	Jl. Mutiara II No 8	0141-357-7419
CO93	Toni	Fadilah	Jl. Mustika I No 25	

## 5. Tabel Properti

No Property	Jalan	Kota	Kode Pos	Type	Ruang	Pinjam	No Milik	No Staff	No Cabang
PA14	Tekukur III	Cianjur	26312	D	6	650.00	CO46	SA9	B7
PG16	Delima I	Jakarta Pusat	56735	F	4	450.00	CO93	SG14	B3
PG21	Otista Raya	Jakarta Timur	12537	D	5	600.00	CO40	SG37	B3
PG36	Cagak Raya	Jakarta Pusat	89747	F	3	375.00	CO93	SG37	B3
PG4	Soetta II	Jakarta Pusat	34536	F	3	350.00	CO40	SG14	B3
PL94	Ahmad Yani	Bogor	46579	F	4	400.00	CO87	SL41	B5

## 6. Latihan dan Evaluasi

-

## 7. Referensi

1. Database system, A Practical Approach to Design, Implementation, and Management; Thomas Connolly, Carolyn Bagg, Pearson 6<sup>th</sup> edition, 2015

## **MODUL BASIS DATA – PRAKTIKUM 5 DATA MANIPULATION LANGUAGE (DML II)**

### **1. Tujuan**

Tujuan Instruksional Umum :

Mampu menguasai SQL Query.

Tujuan Instruksional Khusus :

1. Mampu menguasai perintah SELECT, ORDER BY, GROUP BY, HAVING, UNION, INTERSECT, EXCEPT, EXIST, INSERT, UPDATE, DELETE
2. Mampu menguasai Fungsi Agregat
3. Mampu menguasai Sub Query

### **2. Durasi Waktu**

4 x 55 menit

### **3. Dasar Teori**

Insert Data

```
INSERT INTO TableName [ (columnList) ]  
VALUES (dataValueList)
```

Keterangan:

1. columnList is optional. If omitted, SQL assumes a list of all columns in their original CREATETABLE order.
2. Any columns omitted must have been declared as NULL when table was created, unless DEFAULT was specified when creating column.
3. dataValueList must match columnList as follows:

- number of items in each list must be same;
- must be direct correspondence in position of items in two lists;
- data type of each item in dataValueList must be compatible with data type of corresponding column.

#### Update Data

```
UPDATE TableName
SET columnName1 = dataValue1
    [, columnName2 = dataValue2...]
[WHERE searchCondition]
```

#### Keterangan:

1. TableName can be name of a base table or an updatable view.
2. SET clause specifies names of one or more columns that are to be updated.
3. WHERE clause is optional:
  - if omitted, named columns are updated for all rows in table;
  - if specified, only those rows that satisfy searchCondition are updated.
4. New dataValue(s) must be compatible with data type for corresponding column.

#### Delete Data

```
DELETE FROM TableName
[WHERE searchCondition]
```

#### Keterangan:

1. TableName can be name of a base table or an updatable view.
2. searchCondition is optional; if omitted, all rows are deleted from table. This does not delete table. If search\_condition is specified, only those rows that satisfy condition are deleted.

ISO standard defines five aggregate functions :

<b>COUNT</b>	<b>returns number of values in specified column.</b>
<b>SUM</b>	<b>returns sum of values in specified column.</b>
<b>AVG</b>	<b>returns average of values in specified column.</b>
<b>MIN</b>	<b>returns smallest value in specified column.</b>
<b>MAX</b>	<b>returns largest value in specified column.</b>

#### 4. Peralatan :

1. SQL Server Management Studio

#### 5. Percobaan

Lakukan skenario sesuai dengan soal-soal di bawah ini. Tampilkan query (DDL maupun DML) dan output (berupa printscreen output) dari SQL Server untuk masing-masing soal (kecuali untuk soal konseptual dengan jawaban singkat) Jawaban dalam bentuk file microsoft word atau pdf.

Nama File : P07XXX (X 3 digit terakhir NIM Anda)

1. Buat tabel DEPARTMENT dengan struktur data berikut:

<b>KOLOM</b>	<b>TIPE</b>	<b>PANJANG</b>
ID_DEPT	INT	
DEPT_NAME	VARCHAR	100
HEAD	INT	

2. Pada tabel DEPARTMENT, set ID\_DEPT sebagai Primary Key!
3. SQL Server menyediakan perintah untuk melihat struktur tabel. Apa perintahnya? Gunakan perintah tersebut untuk melihat struktur tabel DEPARTMENT!
4. Masukkan data DEPARTMENT sesuai dengan data berikut:

ID_DEPT	DEPT_NAME	HEAD
51	Management	
52	Operations	
53	Finance	
54	Marketing	
55	Sales	

5. Pastikan data yang dimasukkan sudah sesuai dengan melihat data DEPARTMENT yang ada di database!
6. Buat tabel EMPLOYEE dengan struktur data berikut:

KOLOM	TIPE	PANJANG
EMP_ID	INT	
NAME	VARCHAR	500
SEX	VARCHAR	500
POSITION	VARCHAR	500
SALARY	INT	
ID_DEPT	INT	

Hal-hal yang harus diperhatikan :

- a. EMP\_ID adalah primary key
  - b. EMP\_ID bersifat auto increment dimana SQL server akan mengisi secara otomatis pada setiap row yang terbentuk. (Petunjuk : gunakan IDENTITY)
7. Perlihatkan struktur tabel EMPLOYEE!
  8. Masukkan lima EMPLOYEE sesuai dengan data berikut: (EMP\_ID diambil dari sequence/ autoincrement yang telah dibuat)

EMP_NAME	SEX	POSITION	SALARY	ID_DEPT
Andi	Male	Manager	10000000	51
Hary	Male	Supervisor	2500000	51
Bella	Female	Teller	5000000	52
Wati	Female	CS	3000000	56
Ago	Male	Foreman	2000000	53
Arif	Female	Admin	4000000	52
Chintya	Perempuan	Accountant	4000000	53
Dimas	Laki-laki			56
Eko	Male	Security	2500000	52

9. Pastikan data yang dimasukkan sudah sesuai dengan melihat data EMPLOYEE yang ada di database!
10. Untuk menjaga konsistensi, coba set kolom SEX hanya bisa menerima 'Male' or 'Female'! Apakah berhasil? Mengapa demikian? (Petunjuk : gunakan check constraint)
11. Untuk nomor 10, apabila ada record yang tidak sesuai dengan constraint tersebut, sebutkan, sesuaikan, serta coba buat kembali constraint pada nomor 100!

12. Coba masukkan data berikut, apakah berhasil?

NAME	SEX	POSITION	SALARY	DEPT_ID
Fifi	Perempuan	Teller	4500000	52

13. Sesuaikan data pada nomor 12, kemudian coba masukkan lagi!
14. Coba set kolom POSITION menjadi NOT NULL! Apakah berhasil? Mengapa demikian? Record mana yang tidak sesuai?
15. Coba set kolom ID\_DEPT sebagai Foreign Key yang mengacu pada tabel DEPARTMENT! Apakah berhasil? Mengapa demikian?
16. Untuk nomor 15, apabila ada nilai yang tidak sesuai, maka silahkan set menjadi NULL, kemudian coba buat lagi Foreign Key-nya!
17. Hapuslah data departemen Management pada tabel DEPARTMENT. Apakah berhasil? mengapa demikian?
18. Set kembali foreign key pada tabel EMPLOYEE, gunakan perintah ON DELETE CASCADE. Ulangi perintah no 17. Apakah berhasil? Lihat kembali data pada tabel EMPLOYEE, khususnya pada karyawan yang bekerja di departemen Management. Apa yang terjadi?  
Simpulkan jawaban Anda.
19. Set kolom HEAD pada tabel DEPARTMENT sebagai Foreign Key dari EMP\_ID pada tabel EMPLOYEE!



20. Set pegawai dengan nama 'Andi' sebagai ketua departemen 'Management' dan 'Operations', berikutnya set pegawai dengan nama 'Chintya' sebagai ketua departemen 'Finance'!
21. Apa saja yang dilakukan pada data apabila 'Bella' mengundurkan diri?
22. Tampilkan departemen yang tidak memiliki ketua!
23. Tampilkan jumlah pegawai yang ada!
24. Tampilkan pegawai yang belum ditempatkan pada posisi tertentu!
25. Tampilkan pegawai yang memiliki gaji paling tinggi!
26. Tampilkan pegawai yang memiliki gaji paling rendah!
27. Tampilkan nama pegawai yang merupakan ketua departemen!
28. Tampilkan nama departemen beserta nama ketuanya
29. Departemen mana yang memiliki karyawan paling banyak?
30. Fifi mengundurkan diri (keluar bekerja). Hapuslah data Fifi (berdasarkan ID karyawannya).
31. Kemudian ada karyawan baru untuk menggantikan Fifi, yaitu Susi. Susi masuk di departemen yang sama dengan Fifi dan dengan gaji yang sama dengan Fifi. Tampilkan ID karyawan untuk Susi. Apa kesimpulan Anda..? Apakah autoincrement yang di set pada tabel masih berfungsi?
32. Andi mengalami kenaikan gaji 20%. Ubah data dalam tabel.
33. Ternyata Andi tidak jadi mengalami kenaikan gaji. Undo data sebelum anda ubah. (petunjuk : gunakan perintah ROLLBACK). Apa yang bisa Anda jelaskan mengenai ROLLBACK?
34. Bella pindah departemen ke departemen Finance. Ubahlah data pada tabel. Jadikan perubahan ini permanen sehingga tidak dapat dilakukan undo data. (petunjuk : gunakan perintah COMMIT). Apa yang bisa Anda jelaskan mengenai COMMIT?

## **6. Latihan dan Evaluasi**

-

## **7. Referensi**

1. Database system, A Practical Approach to Design, Implementation, and Management; Thomas Connolly, Carolyn Bagg, Pearson 6<sup>th</sup> edition, 2015

## **MODUL BASIS DATA – PRAKTIKUM 6**

### **DATA MANIPULATION LANGUAGE (DML III) & JOIN TABEL**

#### **1. Tujuan**

Tujuan Instruksional Umum :

Mampu menguasai SQL Query.

Tujuan Instruksional Khusus :

1. Mampu menguasai perintah SELECT, ORDER BY, GROUP BY, HAVING, UNION, INTERSECT, EXCEPT, EXIST, INSERT, UPDATE, DELETE
2. Mampu menguasai Fungsi Agregat
3. Mampu menguasai Sub Query

#### **2. Durasi Waktu**

4 x 55 menit

#### **3. Dasar Teori**

Insert Data

```
INSERT INTO TableName [ (columnList) ]  
VALUES (dataValueList)
```

Keterangan:

4. columnList is optional. If omitted, SQL assumes a list of all columns in their original CREATETABLE order.
5. Any columns omitted must have been declared as NULL when table was created, unless DEFAULT was specified when creating column.
6. dataValueList must match columnList as follows:

- number of items in each list must be same;
- must be direct correspondence in position of items in two lists;
- data type of each item in dataValueList must be compatible with data type of corresponding column.

#### Update Data

```
UPDATE TableName
SET columnName1 = dataValue1
    [, columnName2 = dataValue2...]
[WHERE searchCondition]
```

#### Keterangan:

5. TableName can be name of a base table or an updatable view.
6. SET clause specifies names of one or more columns that are to be updated.
7. WHERE clause is optional:
  - if omitted, named columns are updated for all rows in table;
  - if specified, only those rows that satisfy searchCondition are updated.
8. New dataValue(s) must be compatible with data type for corresponding column.

#### Delete Data

```
DELETE FROM TableName
[WHERE searchCondition]
```

#### Keterangan:

3. TableName can be name of a base table or an updatable view.
4. searchCondition is optional; if omitted, all rows are deleted from table. This does not delete table. If search\_condition is specified, only those rows that satisfy condition are deleted.

ISO standard defines five aggregate functions :

<b>COUNT</b>	<b>returns number of values in specified column.</b>
<b>SUM</b>	<b>returns sum of values in specified column.</b>
<b>AVG</b>	<b>returns average of values in specified column.</b>
<b>MIN</b>	<b>returns smallest value in specified column.</b>
<b>MAX</b>	<b>returns largest value in specified column.</b>

#### 4. Peralatan :

1. SQL Server Management Studio

#### 5. Percobaan

-

#### 6. Latihan dan Evaluasi

Tampilkan **query** dan **output** (berupa printscreen output) dari SQL Server untuk masing-masing soal.

Jawaban dalam bentuk file microsoft word atau pdf.

Nama File : P6XXX (X 3 digit terakhir NIM Anda)

Untuk soal No 1 – 6 gunakan database Northwind

1. Hitunglah jumlah total penjualan untuk masing-masing order didalam tabel order detail. Tampilkan dalam bentuk terurut dari jumlah penjualan besar ke kecil.

2. Dari soal no (1) tampilkan 10 jumlah teratas saja. (petunjuk : gunakan keyword TOP)
3. Modifikasi jawaban Anda no.2 untuk menampilkan jumlah yang duplikasi pada urutan terbawah. (Petunjuk : Gunakan keyword WITH TIES)
4. Tampilkan total jumlah order item untuk dua kategori yang berbeda dari order item yang ada di tabel Order Detail. Contoh tampilkan untuk cateoryID 1 dan 2, tampilan akan seperti gambar berikut :

	categoryid	total_quantity
1	1	9532
2	2	5298

5. Modifikasi jawaban Anda di no.4 sehingga menampilkan untuk semua jenis kategori :

	categoryid	total_quantity
1	3	7906
2	6	4199
3	7	2990
4	1	9532
5	4	9149
6	5	4562
7	2	5298
8	8	7681

6. Tampilkan summary dari jumlah untuk masing-masing orderId untuk semua product, tampilkan hanya yang mempunyai order unit lebih dari 250.

	orderid	total_quantity
1	10515	286
2	10612	263
3	10658	255
4	10678	280
5	10847	288
6	10895	346
7	10990	256
8	11030	330

Untuk soal No 7-10 gunakan database Library

## 7. Referensi

1. Database system, A Practical Approach to Design, Implementation, and Management; Thomas Connolly, Carolyn Bagg, Pearson 6<sup>th</sup> edition, 2015

## **MODUL BASIS DATA – PRAKTIKUM 7**

### **ENTITY RELATIONSHIP DIAGRAM (ERD)**

#### **1. Tujuan**

Tujuan Instruksional Umum :

Mampu memahami, membuat dan menjelaskan Model ER.

Tujuan Instruksional Khusus :

1. Mampu menjelaskan tipe entitas, tipe relasi, atribut, entitas kuat, entitas lemah dan structural constraint
2. Mampu menjelaskan spesialis, generalisasi, agregasi, dan komposisi.

#### **2. Durasi Waktu**

4 x 55 menit

#### **3. Dasar Teori**

Notasi Simbolik:




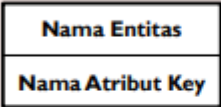
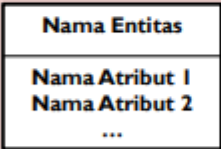
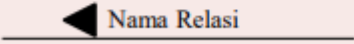

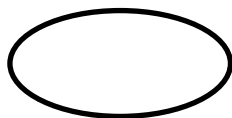
Simbol	Arti Simbol
	Entitas
	Entitas dengan atribut Key
	Entitas dengan atribut
	Nama Relasi dengan panah yang menunjukkan arah relasi
	Relasi dengan constraint

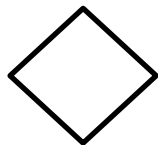
Diagram ERD (alternatif)



Untuk himpunan entitas



Untuk atribut (dengan garis bawah untuk atribut key)



Untuk himpunan relasi



Penghubung

1 atau N

Untuk kardinalitas relasi

Entitas

Entitas merupakan individu yang mewakili sesuatu yang nyata eksistensinya dan dapat dibedakan dari sesuatu yang lain.

Contoh:

Pelanggan A, Mobil Jazz merah, mahasiswa teladan tahun 2013

Himpunan Entitas merupakan sekelompok entitas yang sejenis dan berada dalam lingkup yang sama

Contoh:

Pelanggan, Mobil, mahasiswa

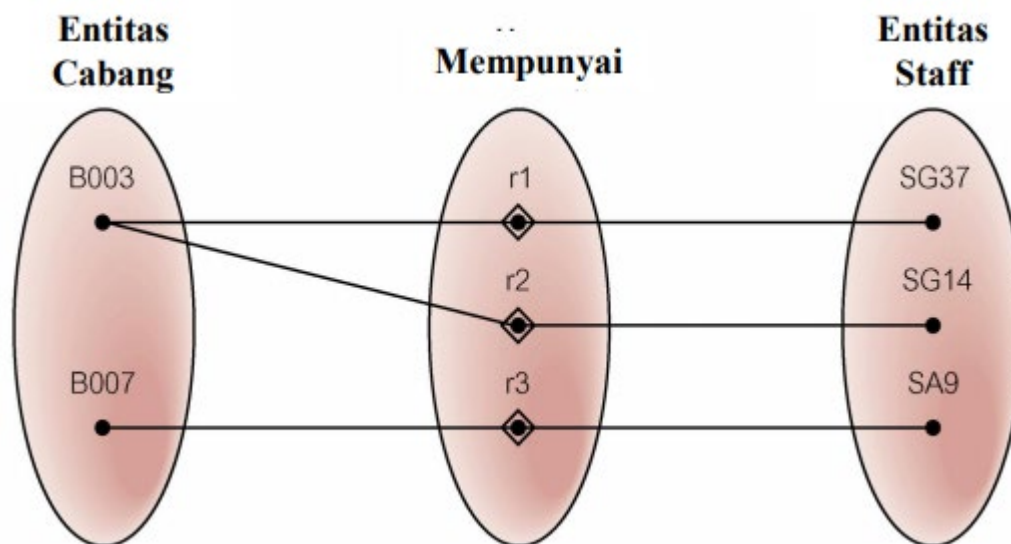
Relasi

Relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

Himpunan Relasi merupakan sekumpulan semua relasi di antara entitas-entitas yang terdapat pada himpunan entitas-himpunan entitas.

Contoh:

Relasi 'mempunyai' antara entitas Cabang dengan Staff



Derajat Relasi

Derajat Relasi menunjukkan jumlah entitas yang terlibat pada sebuah relasi. Entitas yang terlibat dinamakan participant. Jumlah participant menunjukkan derajat/degree

1. Unary / Recursive
2. Binary
3. Ternary
4. Quaternary

### Atribut

Atribut merupakan deskripsi dari karakteristik/property dari sebuah entitas atau relasi. Sedangkan domain atribut merupakan sekumpulan nilai yang dibolehkan dari satu atau lebih atribut.

Tipe-tipe Atribut :

1. Atribut Sederhana (Simple Attribute)  
Atribut yang tidak dapat dibagi lagi menjadi komponen atribut yang lebih kecil.  
Contoh : Atribut posisi dan gaji pada entitas Staff
2. Atribut Majemuk (Composite Attribute)  
Atribut yang terdiri dari lebih dari satu komponen.  
Contoh : Atribut alamat pada tabel Staff sesungguhnya dapat dibagi menjadi atribut jalan, kota dan kode pos.
3. Atribut Bernilai Tunggal (Single-valued Attribute)  
Atribut yang menyimpan nilai tunggal untuk setiap entitas  
Contoh : Atribut KodeCabang pada entitas Cabang.
4. Atribut Bernilai Ganda (Multi-valued Attribute)  
Atribut yang menyimpan nilai ganda untuk setiap entitas  
Contoh : Atribut nomor telepon pada entitas Cabang, dapat bernilai lebih dari satu (jika cabang mempunyai nomor telpon lebih dari satu)
5. Atribut Turunan (Derived Attribute)

Atribut yang merupakan representasi nilai yang dihasilkan dari nilai atribut lainnya, tidak harus berasal dari entitas yang sama.

Contoh : Atribut durasi pada entitas peminjaman berisi nilai yang dihitung dari atribut *tglMulai* dan *tglSelesai*

Key:

1. Candidate Key

Jumlah minimal himpunan dari atribut yang dapat membedakan setiap entitas secara unik.

2. Primary Key

Kandidat Key yang dipilih untuk dijadikan pembeda secara unik.

3. Composite Key

Kandidat Key yang terdiri dari dua atau lebih atribut.

Atribut pada Relasi

1. Relasi juga bisa mempunyai atribut
2. Atribut pada relasi secara simbol digambarkan dengan kotak seperti entitas
3. Untuk membedakan dengan entitas, maka dihubungkan dengan garis putus-putus

Jenis Entitas

1. Entitas Kuat (Strong Entity)

Entitas yang keberadaannya tidak bergantung pada entitas lain. Entitas kuat sering kali direpresentasikan sebagai *parent*, *owner* atau *dominant* entitas.

2. Entitas Lemah (Weak Entity)

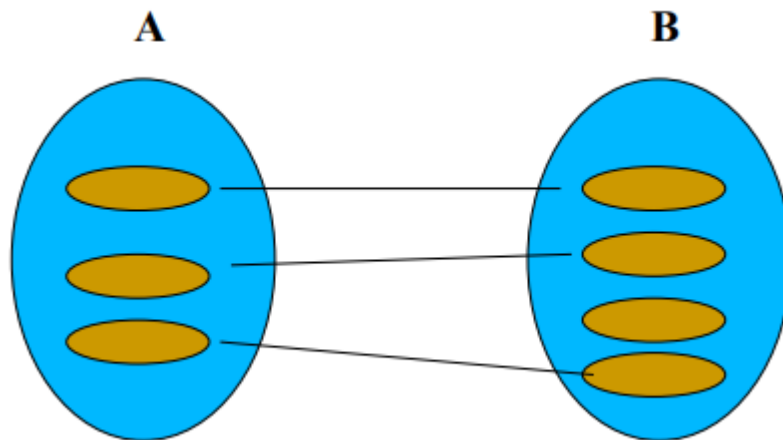
Entitas yang keberadaannya bergantung pada keberadaan entitas lain.

Ciri-ciri entitas lemah dengan tidak mempunyai key dari entitas itu sendiri, key biasanya berasal dari entitas kuatnya. Entitas lemah sering kali direpresentasikan sebagai *child*, *dependent* atau *subordinate* entitas.

## Relasi Binary

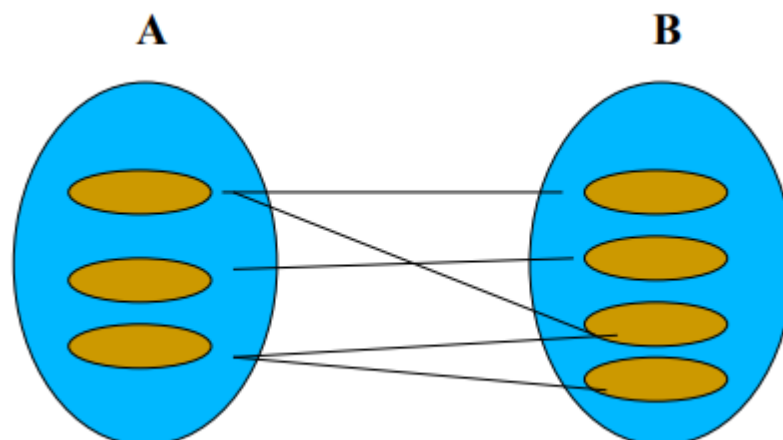
### 1. One-to-one (1:1)

Entitas A dikatakan berelasi One to One kepada entitas B jika setiap entitas di A berhubungan paling banyak dengan satu entitas di B



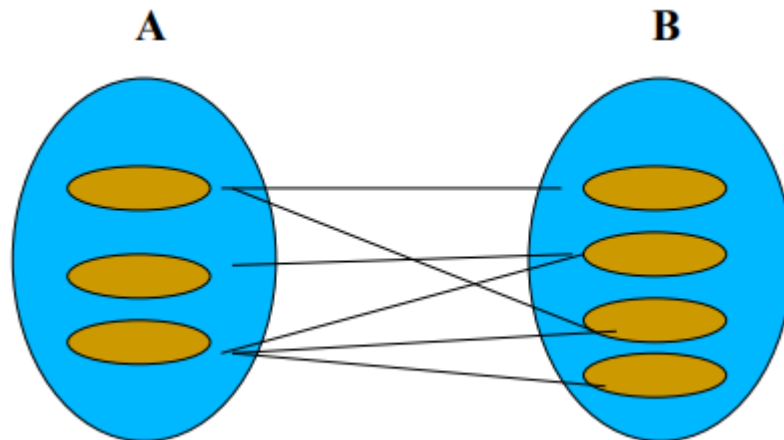
### 2. One-to-many(1:\*)

Entitas A dikatakan berelasi One to Many kepada entitas B jika setiap entitas di A dapat berhubungan dengan banyak entitas di B, tetapi tidak sebaliknya.



### 3. Many-to-many(\*:\*)

Entitas A dikatakan berelasi Many to Many kepada entitas B jika setiap entitas di A dapat berhubungan dengan banyak entitas di B, dan demikian juga sebaliknya, setiap entitas di B dapat berhubungan dengan banyak entitas di A



#### 4. Peralatan :

1. Microsoft Visio

#### 5. Percobaan

#### 6. Latihan dan Evaluasi

Petunjuk pengerjaan :

1. Dalam modul ini terdapat 3 kasus. Setiap kasus dikerjakan oleh dua kelompok yang berlainan :
  - a. Kasus 1 : dikerjakan oleh Kelompok 1 dan 2
  - b. Kasus 2 : dikerjakan oleh Kelompok 3 dan 4
  - c. Kasus 3 : dikerjakan oleh Kelompok 5 dan 6
2. Setiap kelompok membuat ER model berupa ER diagram berdasarkan kasus masing-masing. ER diagram lengkap mengandung :

- Entitas
- Relasi
- Atribut, lengkap dengan penentuan atribut key untuk masing-masing entitas
- Multiplicity Constraint untuk setiap relasi

Kelompok dapat menambahkan asumsi pada kasus jika diperlukan. Jangan lupa menuliskan asumsi tersebut.

3. Setiap kasus dipresentasikan oleh 1 kelompok saja, kelompok lain yang mengerjakan kasus yang sama menjadi penyeimbang/ sumber diskusi.
4. Kelompok yang presentasi adalah kelompok 2, 4 dan 6
5. Pada akhir sesi, hanya ada satu diagram ER untuk setiap studi kasus.

## **KASUS I. TOKO GROSIR GIEHAND**

PT. GieHand berniat mendirikan toko grosir semacam Indo Maret, Alfa, dan sejenisnya yang sudah ada, dengan nama dagang “Gie Store.” Namun demikian, untuk memenangkan persaingan dengan toko-toko sejenis yang sudah ada, PT. GieHand akan membuat para pengunjungnya mudah untuk mengakses informasi mengenai barang yang akan dicarinya.

Informasi dari para pengunjung yang harus dapat dihasilkan oleh sistem komputerisasi adalah atas pertanyaan :

1. Ada atau tidak ada merek dan jenis barang yang akan dicari;
2. Bila ada, tampilkan pula informasi, masih ada berapa unit;
3. Di rak nomor berapa barang tersebut berada, dan;
4. Berapa harga per unitnya.

Ada dua jenis pengunjung, yaitu Anggota (diberikan Kartu Anggota), dan pengunjung biasa (Pembeli). Untuk Anggota, setiap barang yang dibeli diberi potongan 5 persen,

tetapi harus membayar uang iuran anggota sebesar seratus ribu rupiah per tahun. Setiap hari (setelah toko tutup), para Kasir diharuskan untuk melaporkan hasil kegiatannya kepada Pemilik yang berupa :

1. Berapa unit dari masing-masing barang yang terjual (telah dibayarkan kepadanya oleh pengunjung)
2. Berapa uang yang masuk (yang diterimanya dari pengunjung).

Karenanya, pemilik PT. ingin sekali membuat sistem informasi ini ke software house yang ada, agar ia dapat memantau keberadaan barang, aliran uang, dan keuntungan yang diperolehnya.

## **KASUS II. RESERVASI PESAWAT PERUSAHAAN ANGKASA**

Perusahaan Travel 'Angkasa' memasarkan paket wisata dan bisnis perjalanan dari Indonesia ke Amerika dan sebaliknya. Pemilik membutuhkan satu sistem yang mendukung penjadwalan dan pengelolaan dari perusahaan sesuai dengan kebutuhan untuk mendukung contoh perancangan ini.

Agen perjalanan ini bertanggung jawab untuk reservasi (pemesanan tiket pesawat) penumpang bisnis dan penumpang wisata.

1. Informasi mengenai Penumpang
  - a. Perusahaan Angkasa membutuhkan semua identifikasi penumpang, no. KTP dan juga asal negara mereka
  - b. Kebutuhan untuk masa medatang membutuhkan data umur penumpang. Karena ada kebijaksanaan harga khusus untuk anak-anak. Sampai sekarang, tarif hanya dibedakan hanya dengan dasar potongan (discount) pembayaran
  - c. Setiap penumpang mempunyai nama akhir dan nama depan titel
  - d. Setiap penumpang punya satu alamat pengiriman tiket dan satu alamat untuk penagihan sesuai harga tiket tersebut. Dalam kasus penumpang bisnis, alamat



penagihan adalah alamat kantor. Dalam kasus penumpang wisata, alamat kirim tiket dan alamat penagihan jadi satu tempat

- e. Setiap penumpang mempunyai tingkat potongan yang dikeluarkan oleh perusahaan Angkasa. Untuk penumpang bisnis, tingkat potongan ditempatkan pada nama perusahaannya.

## 2. Informasi Reservasi

- a. Agen reservasi mencatat pesawat, metode pembayaran, harga tiket dan tanggal masuk reservasi. Agen juga mencatat identifikasi karyawan yang melayani reservasi.
- b. Untuk menghitung harga tiket, agen menguji tingkat potongan tiap penumpang dengan dasar :

<b>Tingkat Potongan</b>	<b>Potongan</b>
A	-20 %
B	-10 %
C	0 %
D	+ 10 %

- c. Semua tempat duduk tidak untuk merokok dan tidak ada jendelanya

Setelah diadakan analisis didapat fakta tambahan sebagai berikut :

1. Angkasa menjadwal 4 pesawat perhari, dua berangkat dan dua kembali.
2. Terdapat seratus kursi pada setiap pesawat. Rata-rata 75% terpesan. Selebihnya 10 % dipesan dan dibatalkan. Juga ada 8 crew yang menjadi anggota tiap pesawat.
3. Mendekati 50% penumpang pada tiap kapal adalah mengadakan perjalanan wisata dan rata-rata 1.05 kali setiap tahun. Perusahaan mengirim penumpang rata-rata 3 setiap tahun.
4. Kira-kira 3000 penumpang pesan perjalanan wisata dan perjalanan bisnis dalam satu tahun yang sama.

5. Angkasa punya hubungan dengan 1000 perusahaan. Untuk setiap perusahaan rata-rata 6 karyawan dikirim untuk terbang. Terdapat 2000 perusahaan lain tercatat tapi tidak punya jadwal penerbangan yang tetap.
6. Angkasa menggunakan 4 tingkat potongan tiket yaitu ABCD.
7. Penumpang bisnis dan penumpang wisata adalah subtype dari penumpang
8. Setiap penumpang bisnis adalah karyawan dari salah satu perusahaan. Satu perusahaan mengirimkan 6 karyawannya untuk perjalanan tersebut.
9. Setiap reservasi untuk satu pesawat dan satu penumpang bisnis atau satu penumpang wisata. Tentu saja setiap penumpang dan setiap pesawat mempunyai beberapa reservasi.

### **KASUS III. SALON KAYANA**

Kayana adalah pemilik salon yang cukup eksklusif di Jakarta. Bahkan sekarang sudah membuka beberapa cabang di luar Jakarta. Untuk memudahkan bisnisnya, Kayana berniat untuk mengotomasi proses dalam salon miliknya. Untuk itu Kayana membentuk tim khusus yang bertugas melakukan komputerasi dari bisnis proses di salon tersebut. Untuk itu tim membutuhkan beberapa informasi yang menjelaskan bisnis proses di salon Kayana. Penjelasan berikut menggambarkan hasil dari pengamatan tim.

Cabang Salon kayana tersebar di setiap kota besar di Jawa seperti Bandung, Tasikmalaya, Surabaya, Malang, Yogyakarta, Semarang dan Solo. Setiap cabang tersebut mempunyai beberapa salon di kota yang sama sebagai sub cabang. Setiap cabang dan sub cabang dikepalai oleh seorang manager.

Salon Kayana sendiri mempunyai beberapa karyawan. Jumlah karyawan setiap cabang berbeda-beda tergantung dari besar kecilnya cabang tersebut. Setiap karyawan wajib menyerahkan data diri meliputi nama, alamat, nomor telepon, usia, pengalaman kerja.

Perusahaan juga mencatat berapa lama karyawan sudah bekerja dalam perusahaan dan berapa jam karyawan bekerja dalam sebulan.

Kayana mempunyai banyak pelanggan yang loyal. Ini dikarenakan Kayana Salon melakukan pencatatan rapi terhadap data pelanggannya. Data ini meliputi nama, detail dari pelanggan seperti alamat, nomor telepon, profesi, model rambut yang disukai, produk yang disukai dan sering dibeli dan seberapa sering pelanggan tersebut melakukan perawatan di salon Kayana.

Untuk mendukung pelayanan terhadap pelanggan, tentu saja Kayana menyediakan beberapa produk yang dapat dijual. Produk mempunyai harga yang berbeda-beda. Ketersediaan produk harus selalu dijaga dengan selalu memperhatikan stok. Tidak semua produk laku dijual, oleh karena itu diperlukan pencatatan rata-rata konsumsi setiap produk perbulan dan juga produk apa saja yang laku untuk musim-musim tertentu.

Dalam penyediaan produk, Kayana bekerja sama dengan beberapa supplier. Beberapa produk dibeli dari supplier tertentu. Supplier juga diidentifikasi mengenai kehandalannya seperti waktu penyediaan, kualitas produk, dan bagaimana kedudukan supplier tersebut (monopoli atau tidak).

Pelanggan dapat melakukan perjanjian terlebih dahulu sebelum melakukan perawatan. Data perjanjian berupa nama pelanggan yang melakukan perjanjian, waktu perjanjian, karyawan yang dapat melayani dan ketersediaan waktu untuk melakukan perjanjian tersebut.

Berikut informasi tambahan yang berhasil dikumpulkan oleh tim :

- Setiap salon Kayana mempunyai banyak pelanggan
- Pelanggan yang membuat perjanjian dengan penata rambut (karyawan) yang mereka pilih sendiri untuk melakukan style yang disukai

- Pelanggan dikenai tagihan sesuai dengan layanan yang diterima dan penata rambut yang menangani
- Setiap salon mempunyai beberapa penata rambut sebagai karyawan yang tercatat pada salon tersebut.
- Manager salon bertanggung jawab terhadap pembelian stok produk
- Produk dapat digunakan dalam perawatan pelanggan dan juga dapat dijual.
- Setiap salon mempunyai beberapa supplier untuk pengadaan produk.

## **7. Referensi**

1. Database system, A Practical Approach to Design, Implementation, and Management; Thomas Connolly, Carolyn Bagg, Pearson 6<sup>th</sup> edition, 2015

## MODUL BASIS DATA – PRAKTIKUM 8

### NORMALISASI

#### 1. Tujuan

Tujuan Instruksional Umum :

Mampu menjelaskan dan melakukan Normalisasi data.

Tujuan Instruksional Khusus :

1. Mampu menjelaskan tujuan normalisasi
2. Mampu menjelaskan bagaimana normalisasi berperan dalam desain basis data
3. Mampu menjelaskan redudansi data dan anomaly
4. Mampu menjelaskan kebergantungan fungsional
5. Mampu menjelaskan proses normalisasi
6. Mampu menjelaskan 1NF, 2NF dan 3NF

#### 2. Durasi Waktu

2 x 55 menit

#### 3. Dasar Teori

Normalisasi

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan. Tahap normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat (5NF). Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Tujuan normalisasi untuk menghasilkan tabel pada basis data yang sesuai dengan kebutuhan perusahaan.

Keuntungan menggunakan basis data yang sesuai dengan kebutuhan :

1. Lebih mudah bagi user untuk mengakses dan memelihara data.
2. Membutuhkan tempat penyimpanan yang minimal pada komputer.

**Staff**

staffNo	sName	position	salary	branchNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

**Branch**

branchNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St, Glasgow

**StaffBranch**

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

Redudansi Data dan Update Anomali

1. Tabel StaffBranch mempunyai redundansi data dimana detail dari cabang (branch) diulang untuk setiap staff
2. Sebaliknya, informasi cabang hanya muncul sekali untuk setiap cabang pada tabel Cabang dan hanya informasi nomor Cabang (branchNo) yang diulang pada tabel Staff sebagai informasi dimana setiap staff berlokasi
3. Tabel yang mempunyai redundansi data memiliki potensi untuk terjadi update anomali.
4. Update anomali termasuk :
  - a. Penambahan data (Insertion)

Misalkan akan ditambahkan (insert) data kedalam tabel StaffBranch yaitu staff baru yang berlokasi di cabang B007. Maka juga harus diinputkan detail dari cabang no B007, dan harus sesuai dengan data B007 lainnya.

Misalkan akan ditambahkan (insert) data kedalam tabel StaffBranch yaitu cabang baru dengan nomor B006 tetapi belum ada staff pada cabang tersebut. Maka harus membuat baris baru dimana staffNo diisi Null, dan karena staffNo adalah PK, maka tidak diperbolehkan.

b. Penghapusan data (Deletion)

Jika akan menghapus data staff tertentu pada cabang yang hanya mempunyai satu staff, akan berakibat menghapus data cabang tersebut.

Misal akan dihapus data staff SA9. Karena B007 adalah cabang yang hanya mempunyai satu staff SA9 saja, maka penghapusan SA9 mengakibatkan data cabang B007 terhapus dari basis data.

c. Modifikasi data

Jika akan memodifikasi atau mengubah satu data, maka harus merubah semua data yang sesuai atau data menjadi tidak konsisten.

Misal akan diubah data alamat cabang B003, maka perubahan data harus dilakukan di semua row yang mengandung branchNo B003 (bayangkan jika data berjumlah banyak row)

### Kebergantungan Fungsional (KF)

KF merupakan konsep penting yang berhubungan dengan normalisasi. KF menjelaskan relasi atau hubungan antar atribut.

Notasi KF :  $\rightarrow A \rightarrow B$  artinya B bergantung secara fungsional terhadap A (A dan B atribut) atau A menentukan B secara fungsional.  $A \rightarrow B$  menandakan setiap nilai dalam

atribut A berhubungan dengan tepat satu nilai atribut B dalam tabel. Dari KF  $A \rightarrow B$ , A dinamakan determinan, yaitu atribut atau kelompok atribut di sisi kiri dari anak panah.

Karakteristik KF :

1. Determinan harus mempunyai jumlah minimum atribut yang disebut kebergantungan fungsional secara penuh
2. KF secara penuh, B bergantung secara fungsional terhadap A tetapi tidak terhadap subset dari A, B dikatakan bergantung fungsional secara penuh terhadap A

Kebergantungan Transitif (Transitive Dependencies)

Kebergantungan transitif adalah suatu kondisi dimana A, B dan C adalah atribut dari suatu tabel dan berlaku jika  $A \rightarrow B$  dan  $B \rightarrow C$ , maka C bergantung secara transitif terhadap A melalui B. keberadaan ketergantungan transitif dapat mengakibatkan update anomaly dalam suatu tabel.

#### **4. Peralatan :**

1. SQL Server Management Studio

#### **5. Percobaan**

-

#### **6. Latihan dan Evaluasi**

##### **KASUS I. Employee Case**



1. Study the table below :

Emp-No	Emp-Name	Dept	Manager	Proj-id	Proj-Start-Date	Location	Weeks-on-Project
005	Smith	Marketing	Jones	A	12-93	Poole	11
				B	6-94	Plymouth	15
				C	09-94	Portsmouth	6
007	Bond	Accounts	Bloggs	B	06-94	Plymouth	3
				D	06-94	Berlin	9
009	King	Info Systems	Hurne	C	09-94	Portsmouth	10
010	Holt	Accounts	Bloggs	A	12-93	Poole	21
				B	06-94	Belfast	10
				D	06-94	Hamburg	12

Answer the following question:

- a. Is the table having a good design? Give your reason !
  - b. Is there a multivalued attributes ?
  - c. Define KF for the table.
  - d. Decompose the tables as good as possible, based on KF you've define.
2. Study the new table you've created (as the result from table decomposition). Are they having a good design? Give your reason.
  3. Apply 1<sup>st</sup> Normal Form if possible. To do so, remove the repeating group/multivalued attributes.
  4. Apply 2<sup>nd</sup> Normal Form if possible. Remember that the 2<sup>nd</sup> Normal Form removes partial dependencies, i.e all non-primary key attribute fully dependent on the primary key (not partially)
  5. Apply 3<sup>rd</sup> Normal Form if possible. Remember that the 3<sup>rd</sup> Normal Form removes transitive dependencies, i.e remove dependencies on the non- primary key attributes.
  6. How many tables do you have now ?

## KASUS II. Project Case

1. Study the table below :

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
5	Evergreen	103	June E. Arbough	Elect. Engineer	\$84.50	23.8
		101	John G. News	Database Designer	\$105.00	19.4
		105	Alice K. Johnson *	Database Designer	\$105.00	35.7
		106	William Smithfield	Programmer	\$35.75	12.6
		102	David H. Senior	Systems Analyst	\$96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	\$48.10	24.6
		118	James J. Frommer	General Support	\$18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	\$96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	\$45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	\$105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	\$96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	\$48.10	23.6
		111	Geoff B. Wabash	Clerical Support	\$26.87	22.0
		106	William Smithfield	Programmer	\$35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	\$35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	\$96.75	45.8
		101	John G. News *	Database Designer	\$105.00	56.3
		114	Annelise Jones	Applications Designer	\$48.10	33.1
		108	Ralph B. Washington	Systems Analyst	\$96.75	23.6
		118	James J. Frommer	General Support	\$18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	\$45.95	41.4

Do this Case exactly with the same way as previous task (as you've done on Case 1)

### KASUS III. Library Case

1. Study this table :

Catalog No	Title	Author	Publisher	Publisher Address	Copy No	Borrower ID	Borrower	Return Date
1	Understanding Business	Betty Schramper	Sams	Jakarta	1	001	Dedi	31/8/04
2					002	Iwan	12/9/2004	
3					003	Sri	12/9/2004	
4	The Other Side of Midnight	Sidney Sheldon	Gramedia	Jakarta	1	004	Ade	31/8/04
5					005	Aditya	12/9/2004	
6					001	Dedi	12/9/2004	

Do this Case exactly with the same way as previous task (as you've done on Case 1)

### 7. Referensi

1. Database system, A Practical Approach to Design, Implementation, and Management; Thomas Connolly, Carolyn Bagg, Pearson 6<sup>th</sup> edition, 2015