

TUTORIAL MEMBUAT DATABASE DENGAN MICROSOFT ACCESS 2013- BAGIAN 3

Setelah mempelajari modul ini diharapkan Anda dapat:

1. Memahami konsep dasar pembuatan database
2. Memahami pengertian entitas dan relationship
3. Memahami pengertian relasi antartabel, kunci dan jenis-jenis relasi
4. Mempraktikkan pembuatan relationship tabel

Konsep Dasar Pembuatan Database

Database merupakan kumpulan data yang saling berhubungan yang disimpan sedemikian rupa tanpa adanya *redundancy* (perulangan) yang tidak perlu. Database terbentuk dari sekelompok data-data yang memiliki jenis atau sifat yang sama. Contohnya: data mahasiswa, data dosen, data mata kuliah, data karyawan, dll. Untuk dapat membuat suatu database yang baik, diperlukan suatu rancangan yang tepat dan sesuai kebutuhan. Konsep dasar dalam pembuatan database antara lain:

a. Entitas dan Relationship

Hal yang sangat mendasar dan harus dipahami adalah pemodelan **entitas dan relationship**. **Entitas adalah berbagai hal dalam dunia nyata yang informasinya disimpan dalam database**. Sebagai contoh, kita dapat menyimpan informasi tentang dosen yang mengajar mahasiswa. Dalam hal ini, dosen dan mahasiswa merupakan entitas.

Relationship hubungan antara entitas. Sebagai contoh, dosen mengajar mahasiswa. Mengajar merupakan relationship antara entitas dosen dan entitas mahasiswa.

Relationship terdiri dari 3 derajat yang berbeda, yaitu:

1. *One-to-one* menghubungkan secara tepat dua entitas dengan satu kunci (key). Misalnya seorang mahasiswa memiliki satu dosen wali.
2. *One-to-many (many-to-one)* merupakan hubungan antar entitas dimana kunci (key) pada satu tabel muncul berkali-kali pada tabel lainnya. Misalnya banyak mahasiswa memiliki satu dosen wali.
3. *Many-to-many* merupakan hubungan antar entitas dimana kunci utama pada tabel pertama dapat muncul beberapa kali pada tabel kedua, dan sebaliknya. Misalnya seorang mahasiswa dapat mengambil banyak matakuliah, dan satu matakuliah bisadiambil oleh banyak mahasiswa.

b. **Relasi atau Tabel**

Relasi merupakan tabel yang mewakili entitas, dimana didalamnya terdapat kolom-kolom yang merupakan atribut dari entitas. Penamaan tabel juga disesuaikan dengan nama dari entitas agar mudah dipahami. Contoh tabel pegawai merupakan tabel yang mewakili entitas pegawai.

c. **Kolom atau Attribute**

Kolom atau attribute merupakan bagian dari tabel yang mewakili ciri dari suatu entitas. Contoh entitas mahasiswa. Maka atribut yang ada antara lain NIM, nama, dan kelas.

d. **Kunci (key)**

Kunci atau key atau *primary key* merupakan suatu nilai dalam sebuah tabel yang dapat digunakan untuk mengidentifikasi suatu baris dalam tabel. Contoh dalam tabel mahasiswa, didalamnya terdapat kolom NIM, nama, dan kelas. Untuk mengidentifikasi suatu baris dalam tabel, maka cukup menggunakan NIM.

Relasi Antar Tabel

a. **KUNCI (KEY)**

Sebelum membahas mengenai relasi, faktor penting yang menyangkut relasi antar table adalah kunci (*key*). Kunci (*key*) merupakan satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*record*) dalam tabel secara unik.

Macam-macam kunci dibedakan menjadi:

1. **KUNCI PRIMER**

Kunci Primer merupakan sebuah field pada tabel yang merupakan identitas bagi field-field lainnya. Kunci Primer ini tidak boleh memiliki record kembar. Kunci Primer biasanya berada pada tabel induk.

2. **KUNCI SEKUNDER**

Kunci Sekunder merupakan kebalikan dari kunci primer. Kunci Sekunder biasanya berada pada tabel anak. Kedua kunci ini saling berhubungan, dan karena saling berhubungan maka tipe data yang dipakai harus sama.

b. Jenis-jenis Relasi

Secara garis besar, jenis-jenis relasi dibagi menjadi empat, yaitu:

1. ONE – TO ONE (SATU KE SATU)

Relasi ini adalah relasi yang menghubungkan sebuah record pada table induk ke tepat sebuah record pada table anak, akan tetapi relasi ini relatif jarang dipergunakan karena jika record pada table anak hanya cocok dengan satu record pada table induk, mengapa record pada table anak tersebut tidak dijadikan satu pada table induk dengan menambahkan field-field table anak pada table induk?.

Contoh: Daftar pelanggan sebuah toko online dengan alamat pengiriman berbeda dengan alamat penagihan tetapi ada beberapa kelemahan yaitu:

- Pelanggan yang membedakan alamat pengiriman dengan alamat penagihan tidak banyak.
- Kalau alamat pengiriman dengan alamat penagihan dijadikan dalam satu tabel, maka field penagihan akan banyak yang kosong karena hanya terisi sedikit saja.
- Sebaiknya dibuat tabel sendiri-sendiri yaitu tabel alamat pengiriman dan table alamat penagihan untuk menyimpan informasi dan kemudian masingmasing tabel tersebut direlasikan.
- Satu record alamat pengiriman hanya akan terhubung dengan satu record nama pelanggan.

Contoh:

Tabelalamatpengiriman		Tabelalamatpenagihan	
Nama	Alamatpengiriman	Nama	Alamatpenagihan
A	Jl. Tologomas	-	-
B	Jl. Dinoyo	B	Jl.Mergojoyo

2. ONE – TO – MANY (SATU KE BANYAK)

Relasi yang menghubungkan satu record pada satu tabel dengan beberapa record pada tabel lainnya.

Contoh. Database Perpustakaan dengan beberapa kriteria yaitu:

- Nama satu anggota perpustakaan hanya akan tercatat satu kali dalam tabel anggota.
- Akan tercatat berkali-kali dalam tabel pinjam.

- Satu orang bias meminjam buku sampai berkali-kali.

Contoh:

Tabel Anggota

Nama :
Alamat :

Tabel peminjam

Nama	NIM	TanggalPinjam	KodeBuku
A	0012001	25 Februari 2014	

3. MANY – TO – ONE (BANYAK KE SATU)

- Relasi kebalikan dari *One To Many* (satu ke banyak) atau biasa disebut dengan *look up table relationship*.
- Relasi ini tidak saling terhubung ke kunci primer pada kedua tabel.

Contoh. Database Nama Propinsi dengan kriteria yaitu:

- Tabel informasi propinsi di Indonesia dihubungkan ke tabel alamat pelanggan.
- Record kota yang ada pada alamat pelanggan berelasi dengan record propinsi.
- Banyak kota yang bisa masuk ke dalam sebuah propinsi.

Contoh:

Tabel Propinsi

Kota	Propinsi
Malang, Surabaya, Kediri, Blitar, dll	JawaTimur

Tabel Alamat Pelanggan

Nama	Alamat
Andi	Malang
Asri	Surabaya

4. MANY – TO – MANY (BANYAK KE BANYAK)

- Relasi many to many ini hamper tidak mungkin ada.
- Jika ada biasanya terjadi karena kesalahan dalam perancangan tabel.

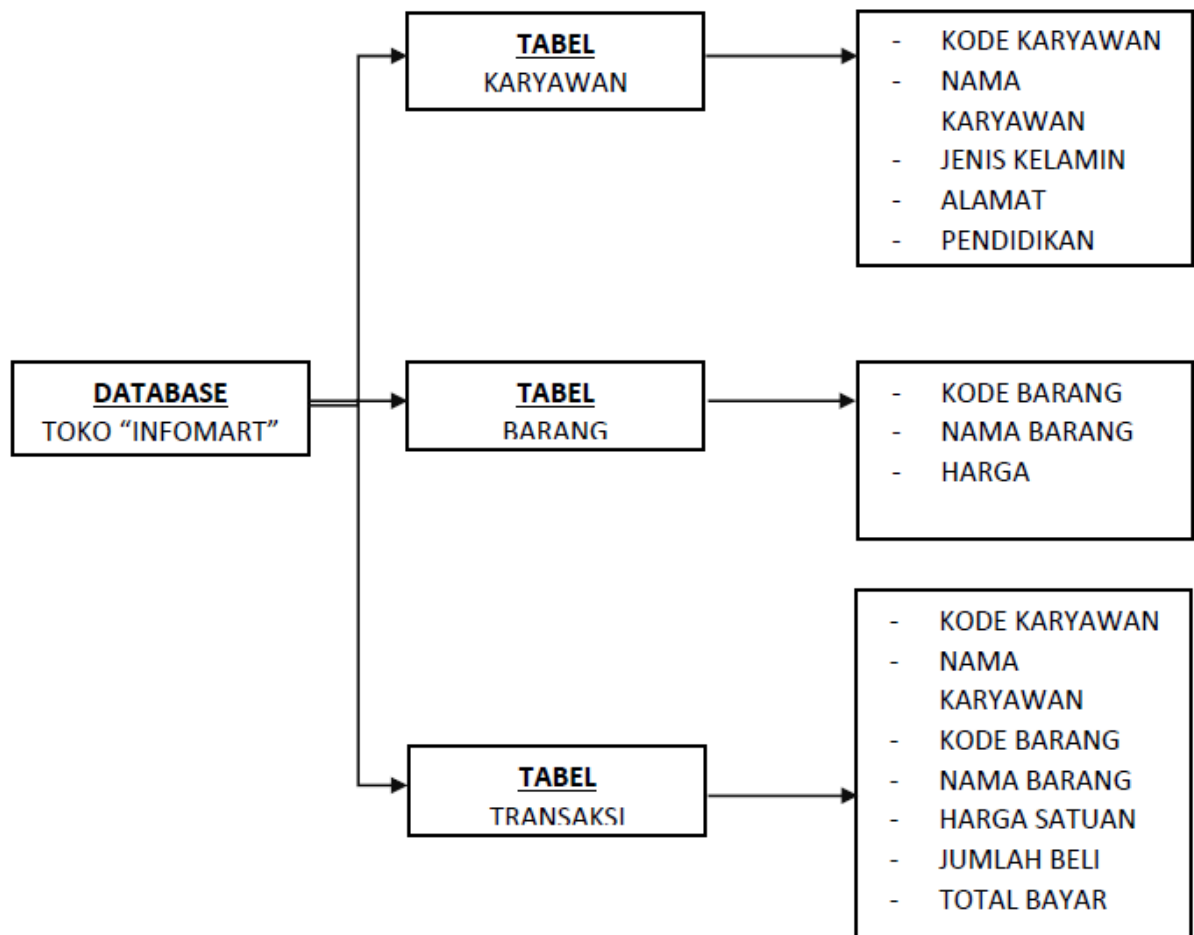
Latihan Membuat Relationship

Langkah 1

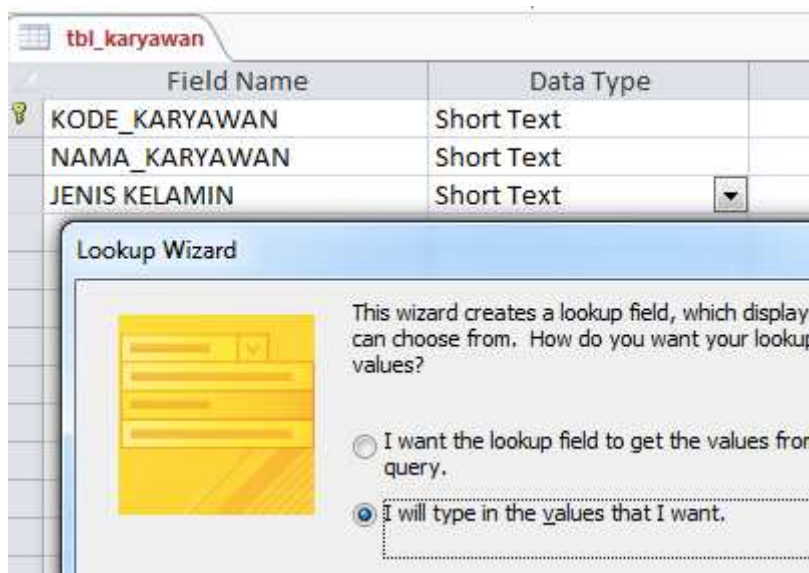
Buatlah database dengan nama **toko** dan 3 buah tabel, **tbl_karyawan**, **tbl_barang**, dan **tbl_transaksi**.
Tentukan **Field Size, Format dan Description** sesuai kebutuhan.

Atur **KODE KARYAWAN** pada tabel karyawan sebagai *primary key*.

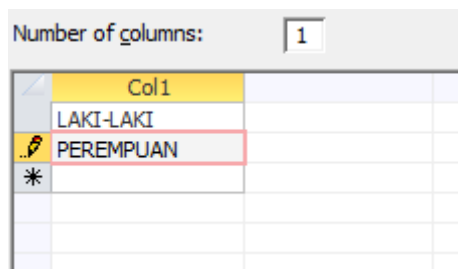
Atur **KODE BARANG** pada tabel barang sebagai *primary key*.



Khusus untuk field **Jenis Kelamin** gunakan **tipe data Lookup Wizard** dan lakukan pengaturan seperti berikut.



Setelah itu klik **Next**, isikan seperti berikut.



Berikutnya klik **Next dan Finish**.

Langkah 2

Isikan data pada masing-masing tbl_karyawan, tbl_barang, tbl_transaksi.

tbl_karyawan

	KODE_KARYA	NAMA_KARYAW	JENIS_KELAI	ALAMAT	PENDIDIKAN
+	K001	ABDI	LAKI-LAKI	YOGYAKARTA	S1
+	K002	BUDI	LAKI-LAKI	SEMARANG	D3
+	K003	CLARA	PEREMPUAN	BANDUNG	S1
+	K004	DANAR	LAKI-LAKI	SEMARANG	S2
+	K005	ENI	PEREMPUAN	YOGYAKARTA	D3
*					

tbl_barang

	KODE_BARANG	NAMA_BARANG	HARGA	Click to Add
+	B001	BUKU	Rp5.000	
+	B002	ROTI	Rp10.000	
+	B003	PARFUM	Rp15.000	
+	B004	INDOMIE	Rp1.500	
+	B005	PENSIL	Rp3.000	
*			Rp0	

tbl_transaksi

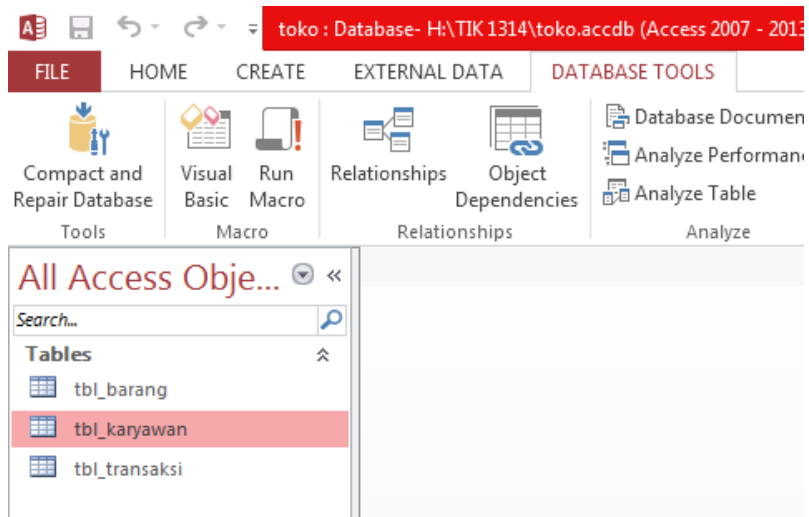
	KODE_KARYAWA	NAMA_KARYAW	KODE_BARA	NAMA_BAR	HARGA_SATUAN	JUMLAH_BELI	TOTAL_BAY
	K001	ABDI	B001	BUKU	Rp5.000	10	Rp50.000
	K001	ABDI	B002	ROTI	Rp10.000	5	Rp50.000
	K002	BUDI	B003	PARFUM	Rp15.000	2	Rp30.000
	K003	CLARA	B003	PARFUM	Rp15.000	4	Rp60.000
	K004	DANAR	B004	INDOMIE	Rp1.500	15	Rp22.500
	K004	DANAR	B002	ROTI	Rp10.000	6	Rp60.000
	K005	ENI	B005	PENSIL	Rp3.000	10	Rp30.000
	K005	ENI	B004	INDOMIE	Rp1.500	5	Rp7.500
	K005	ENI	B001	BUKU	Rp5.000	20	Rp100.000

Langkah 3

Relationship berguna untuk menghubungkan antartabel dalam database.

Buatlah relationship seperti berikut.

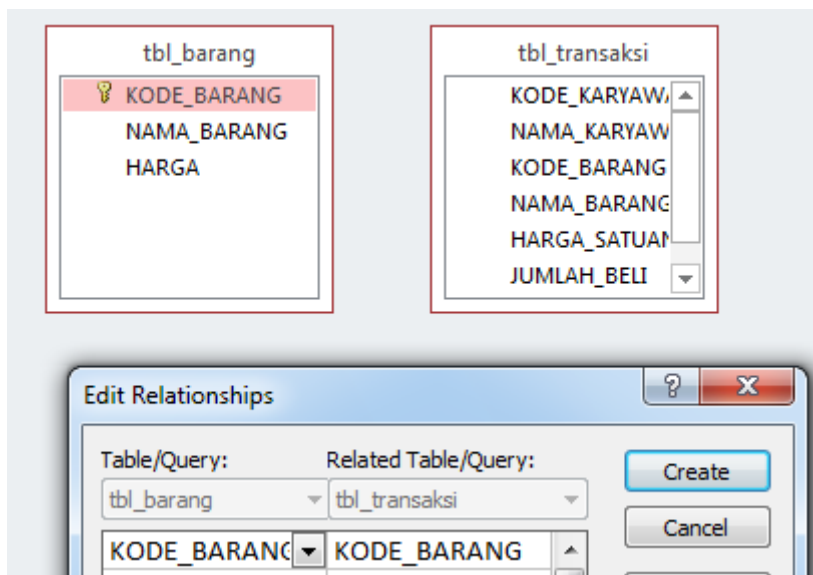
Klik Relationships pada menu *DATABASE TOOLS*.



Klik Add pada Show Table, untuk masing-masing tabel.



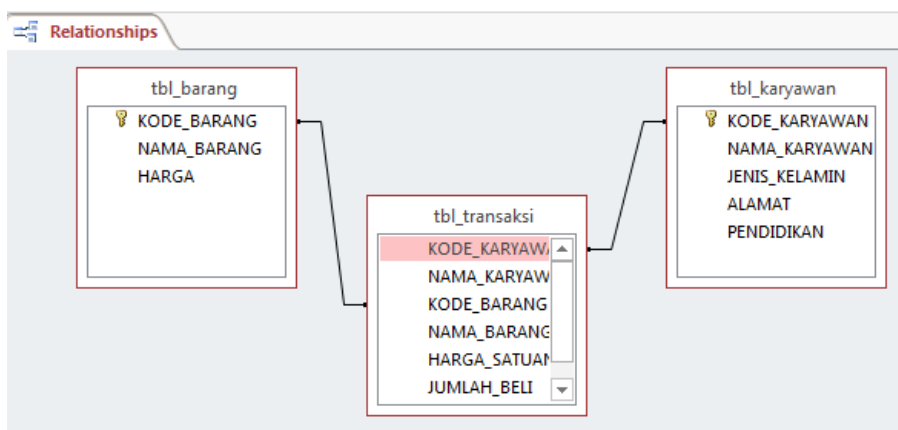
Drag and Drop dari **KODE_BARANG** pada **tbl_barang** ke **KODE_BARANG** di **tbl_transaksi**.



Drag and Drop dari **KODE_KARYAWAN** pada **tbl_transaksi** ke **KODE_KARYAWAN** pada **tbl_karyawan**.



Berikut ini adalah tampilan hasil akhir relationship yang dibuat.



Sumber:

Team Lab Komputer DPP Infokom@2013