

**LAPORAN TUGAS BESAR
BASIS DATA I**

SISTEM INFORMASI PERMINTAAN BARANG



Oleh:

Mariana

A1317041

PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK NEGERI TANAH LAUT

PELAIHARI

2018

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan laporan praktikum ini. Yang mana laporan ini adalah tugas besar untuk mata kuliah basis data 1. Laporan ini merupakan hasil dari tugas praktikum bagi para mahasiswa, untuk mempelajari dan memahami perancangan untuk membangun suatu sistem informasi. Penulisan laporan ini bertujuan untuk menumbuhkan proses belajar mandiri kepada mahasiswa, agar kreativitas dan penguasaan materi kuliah dapat optimal sesuai dengan yang diharapkan.

Laporan ini disusun sebagai syarat UAS Praktikum mata kuliah Basis Data 1. Semoga laporan ini dapat bermanfaat dan senantiasa menjadi pembelajaran untuk meraih prestasi yang gemilang. Kritik dan saran dari dosen pengajar mata kuliah dan juga bagi semua pembaca, sangat penulis harapkan untuk perbaikan dan penyempurnaan dalam pembelajaran pada masa mendatang. Karena penulis sadra masih banyak kekurangan ang ada pada laporan ini.

Pelaihari, 2018

Penulis,

Mariana
A1317041

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB 1	1
PENDAHULUAN	1
1.1. Latar Belakang	1
BAB 2	2
LANDASAN TEORI	2
2.1 Pengertian Basis Data	2
2.2 DBMS	2
2.3 SQL	4
2.4 Entity Relationship Diagram (ERD)	4
2.4.1. Simbol-simbol ERD	5
2.4.2. Derajat relasi	6
2.4.3. Kardinalitas	7
2.5 Conceptual Data Model (CDM)	9
2.5.1. Simbol-simbol CDM	9
2.5.2. Aturan-aturan CDM	10
2.6 Physical Data Model (PDM)	12
2.6.1. Simbol-simbol PDM	12
BAB 3	13
PEMBAHASAN	13
3.1 Studi Kasus	13
3.2 Studi Kasus ERD	15
3.2.1. Definisi Entitas dan Atribut	15
3.2.2. Definisi Relasi	17
3.2.3. Diagram ER	19
3.3 Studi Kasus CDM	20
3.4 Studi Kasus PDM	21
BAB 4	22
PHPMYADMIN	22

4.1	Pengertian PHPMyAdmin	22
4.2	Membuat database dari Mysql PHPMyAdmin	23
4.3	Import database dari tools power designer.....	32
4.3.1	Membuat CDM.....	32
4.3.2	Generate dari CDM ke PDM	41
4.3.3	Generate Database dari hasil PDM.....	42
4.4	Import dan export csv data.....	47
4.4.1	Import data CSV	47
4.4.2	Export database ke CSV.....	53

BAB 1

PENDAHULUAN

1.1. Latar Belakang

MySQL merupakan *software database open source* yang populer di dunia, dimana saat ini digunakan lebih dari 100 juta pengguna diseluruh dunia. Dengan kehandalan, kecepatan dan kemudahan penggunaannya, MySQL menjadi pilihan utama bagi banyak pengembang *software* dan aplikasi baik di *platform web* maupun *desktop*. Pengguna MySQL tidak hanya sebatas pengguna perseorangan maupun perusahaan kecil, namun perusahaan seperti Yahoo!, Alcalter-Lucent, Google, Nokia, Youtube, Wordpress dan Facebook juga merupakan pengguna MySQL.

MySQL pertama kali dibuat dan dikembangkan di Swedia, yaitu oleh David Axmark, Allan Larson dan Michael “Monty” Widenius. Mereka mengembangkan MySQL sejak tahun 1980-an. Saat ini versi MySQL yang sudah stabil mencapai versi 5x dan sedang dikembangkan versi 6x. untuk lebih lengkapnya dapat dilihat di situs resmi MySQL.

BAB 2

LANDASAN TEORI

2.1 Pengertian Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kasus ini menggunakan basis data relasional yang diimplementasikan dengan tabel-tabel yang saling memiliki relasi.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DBMS). Kebutuhan basis data dalam sistem informasi meliputi memasukkan, menyimpan, dan mengambil data, serta membuat laporan berdasarkan data yang telah disimpan.

Tujuan dari dibuatnya tabel-tabel dikasus ini adalah untuk menyimpan data ke dalam tabel-tabel agar mudah diakses. Oleh karena itu, untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris-baris data (*record*) dimana setiap baris terdiri dari beberapa kolom.

2.2 DBMS

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut.

- a. Menyediakan fasilitas untuk mengelola akses data
- b. Mampu menangani integritas data
- c. Mampu menangani akses data
- d. Mampu menangani *backup* data

Karena pentingnya data bagi suatu organisasi/perusahaan, maka hampir sebagian besar perusahaan memanfaatkan DBMS dalam mengelola data yang mereka miliki. Pengelolaan DBMS sendiri biasanya ditangani oleh tenaga ahli yang spesialis mengenai DBMS yang disebut sebagai DBA (*Database Administrator*).

DBMS sudah mulai berkembang sejak tahun 1960an. Kemudian sekitar tahun 1970an mulai berkembang teknologi *Relational DBMS* yaitu DBMS berbasis relasional model. Relasional model pertama kali dikembangkan oleh Edgar J. Codd pada tahun 1970. Secara sederhana relasional model dapat dipahami sebagai suatu model yang memandang data sebagai sekumpulan tabel yang saling terkait. Hampir semua DBMS komersial dan *open source* saat ini berbasis *Relational DBMS* atau RDBMS.

Pada tahun 1980an mulai berkembang *Object Oriented DBMS* (OODBMS). OODBMS berkembang seiring dengan perkembangan teknologi pemrograman berorientasi objek. Saat ini OODBMS juga cukup berkembang namun belum dapat menggeser kepopuleran RDBMS.

Berikut ini adalah 4 (empat) macam DBMS versi komersial yang paling banyak digunakan didunia saat ini, yaitu.

- a. Oracle
- b. Microsoft SQL Server
- c. IBM DB2
- d. Microsoft Access

Sedangkan DBMS versi *open source* yang cukup berkembang dan paling banyak digunakan saat ini adalah sebagai berikut.

- a. MySQL
- b. PostgreSQL
- c. Firebird
- d. SQLite

Hampir semua DBMS mengadopsi SQL sebagai bahasa untuk mengelola data pada DBMS.

2.3 SQL

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. SQL mulai berkembang pada tahun 1970an. SQL mulai digunakan sebagai standar yang resmi pada tahun 1986 oleh ANSI (*American National Standards Institute*) dan pada tahun 1987 oleh ISO (*International Organization for Standardization*) dan disebut sebagai SQL-86. Pada perkembangannya, SQL beberapa kali dilakukan revisi. Berikut ini sejarah perkembangan SQL sampai saat ini.

- 1) Tahun 1986, SQL-86
- 2) Tahun 1989, SQL-89
- 3) Tahun 1992, SQL-92
- 4) Tahun 1999, SQL:1999
- 5) Tahun 2003, SQL:2003
- 6) Tahun 2006, SQL:2006
- 7) Tahun 2008, SQL:2008
- 8) Tahun 2011, SQL:2011

Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagai besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal *query* yang dimasukkan sama.

2.4 Entity Relationship Diagram (ERD)

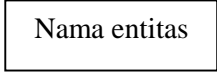
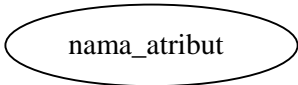
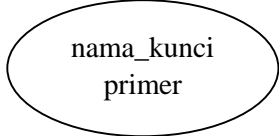
Entity Relationship Diagram (*ERD*) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. *Entity Relationship Diagram* merupakan model jaringan yang menekankan pada struktur dan hubungan antardata. *Entity Relationship Diagram* juga memperlihatkan hubungan antardata *store* pada *Data Flow Diagram*. *Entity Relationship Diagram* atau lebih dikenal dengan *E-R* adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang disimpan (*storage data*) dalam sistem secara abstrak. *Entity Relationship Diagram* tidak menyatakan

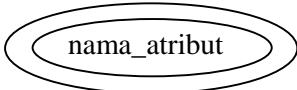


bagaimana memanfaatkan data, membuat data, mengubah data dan menghapus data.

Entity Relationship Diagram (ERD) dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen. Berikut simbol-simbol yang digunakan pada ERD dengan notasi Chen:

2.4.1. Simbol-simbol ERD

Tabel 1.1 Simbol-simbol Entity Relationship Diagram

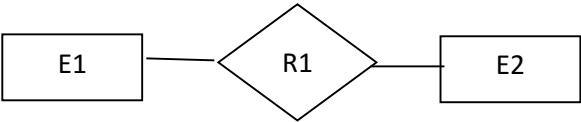
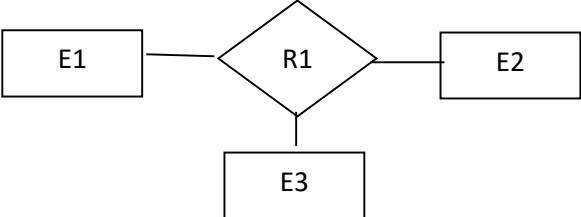
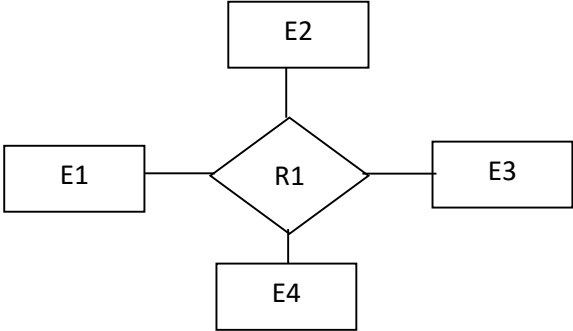
Nama simbol	Simbol	Deskripsi
Entitas / entity	 Nama entitas	Entitas merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
Atribut	 nama_atribut	Field atau kolom data yang butuh disimpan dalam suatu entitas.
Atribut kunci primer	 nama_kunci primer	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)

Atribut multi nilai / multivalue		Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu. Misalnya riwayat pendidikan, nomer handphone, email dan lain sebagainya.
Relasi		Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi / association		<p>Penghubung antara relasi dan antitas di mana kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian.</p> <p>Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many menghubungkan entitas A dengan entitas B.</p>

2.4.2. Derajat relasi

Menunjukkan banyaknya himpunan entitas yang saling berelasi. ERD biasanya memiliki hubungan binary (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi ternary (satu relasi menghubungkan tiga buah relasi) atau N-ary (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan ternary atau N-ary. Berikut adalah contoh bentuk hubungan relasi dalam ERD.

Tabel 2.1 Derajat relasi dalam Entity Relationship Diagram

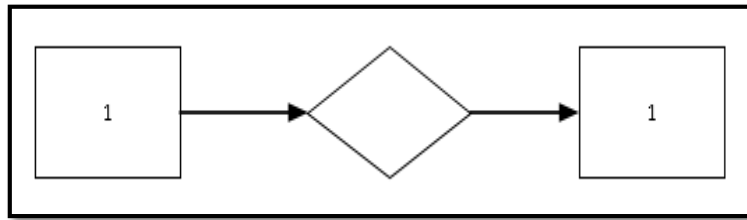
Nama	Gambar
Binary	
Ternary	
N-ary	

2.4.3. Kardinalitas

Relationship mempunyai tiga tipe kardinalitas yang mana tiap tipe menunjukkan jumlah *record* dari setiap tabel yang direlasikan ke *record* pada tabel lain. Ketiga tipe tersebut adalah sebagai berikut :

1. Hubungan satu ke satu (*One to one relationship*)

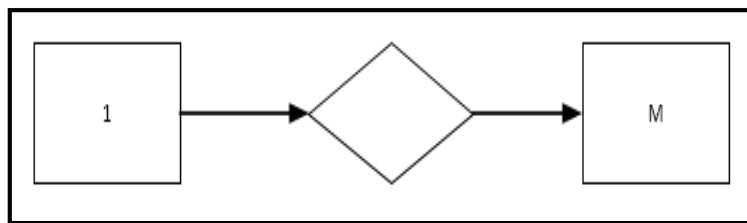
Hubungan antara file pertama dan file kedua berbanding satu. Dalam hubungan ini, tiap *record* dalam tabel A hanya memiliki satu *record* yang cocok dalam tabel B dan tiap *record* dalam tabel B hanya memiliki satu *record* dalam tabel A. Logika penalaran matematika dari *one to one relationship* adalah pemetaan dengan “perkawan satu-satu” atau sering disebut dengan korespondensi satu-satu. Contoh *One to one relationship* adalah satu pasien mempunyai satu tempat tidur.



Gambar 2.1 *One to one relationship*

2. Hubungan satu ke banyak (*One to many relationship*)

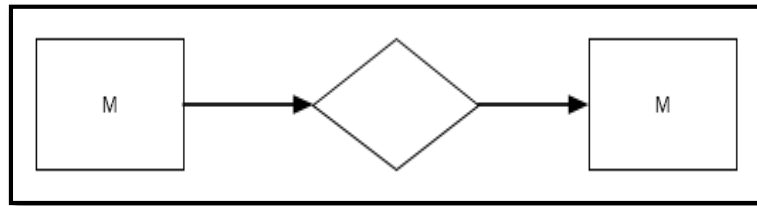
Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak. Dalam hubungan ini tiap *record* dalam tabel A memiliki beberapa *record* yang cocok dalam tabel B. Logika penalaran matematika dari *one to many relationship* adalah “Perkawanan satu ke banyak”. Contoh *One to many relationship* adalah satu dosen mengajar banyak mata kuliah.



Gambar 2.2 *One to many relationship*

3. Hubungan banyak ke banyak (*Many to many relationship*)

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Dalam hubungan ini tiap *record* dalam tabel A memiliki beberapa *record* yang cocok dalam tabel B dan tiap *record* dalam tabel B hanya memiliki satu *record* yang cocok dalam tabel A. Logika penalaran matematika dari *many to many relationship* adalah pemetaan “Perkawanan banyak ke banyak”. Contoh *many to many relationship* adalah banyak mahasiswa memiliki banyak mata kuliah dan banyak mata kuliah memiliki banyak mahasiswa. Hubungan *many to many* tidak dapat diimplementasikan ke dalam *database relationship* sehingga hubungan ini harus dipecah menjadi hubungan *one to many*.



Gambar 2.3 *Many to Many relationship*

2.5 Conceptual Data Model (CDM)

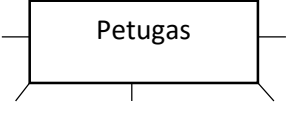
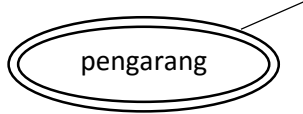
Conceptual Data Model(CDM) atau model konsep data merupakan konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data. CDM dibuat sudah dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke basis data.

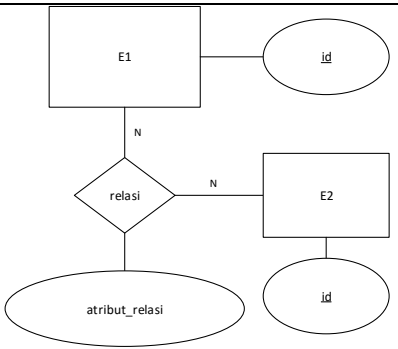
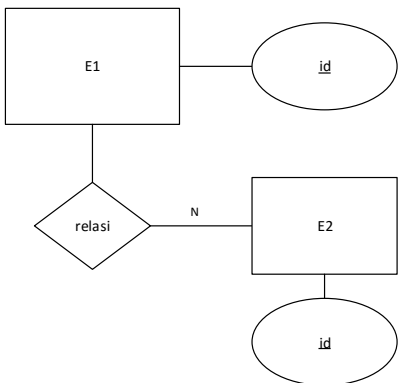
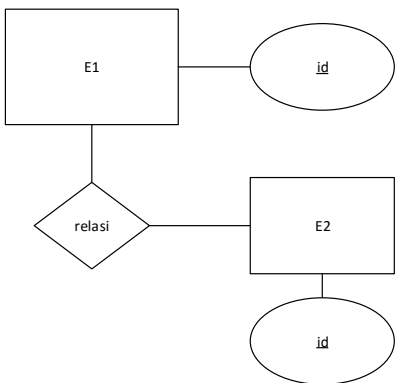
CDM merupakan hasil penjabaran lebih lanjut dari ERD. Ada aturan-aturan yang harus diikuti dalam melakukan konversi ERD menjadi CDM.

2.5.1. Simbol-simbol CDM

Simbol	Deskripsi
Entitas/Tabel <div style="border: 1px solid black; padding: 5px; width: fit-content;"> nama_tabel <hr/> <div style="display: flex; border-bottom: 1px solid black;"> <div style="width: 20px; height: 15px;"></div> </div> <div style="display: flex; border-bottom: 1px solid black;"> <div style="width: 20px; height: 15px;"></div> </div> </div>	Entitas atau tabel yang menyimpan data dalam basis data.
Relasi <div style="display: flex; align-items: center; gap: 10px;"> <div style="border-left: 1px solid black; padding-left: 5px;"> 1..* nama relasi 1..* </div> </div>	Relasi antar tabel yang terdiri atas nama relasi dan <i>multiplicity</i>

2.5.2. Aturan-aturan CDM

ERD	CDM												
 <p>entitas</p>	<table border="1" data-bbox="901 398 1149 734"> <tr><td colspan="2">Petugas</td></tr> <tr><td>PK</td><td><u>username</u></td></tr> <tr><td></td><td>password</td></tr> <tr><td></td><td>nama</td></tr> <tr><td></td><td>no_petugas</td></tr> <tr><td></td><td>hak_akses</td></tr> </table> <p>Menjadi sebuah tabel tersendiri</p>	Petugas		PK	<u>username</u>		password		nama		no_petugas		hak_akses
Petugas													
PK	<u>username</u>												
	password												
	nama												
	no_petugas												
	hak_akses												
 <p>atribut <i>multivalue</i></p>	<table border="1" data-bbox="901 1059 1149 1283"> <tr><td colspan="2">Pengarang</td></tr> <tr><td>PK</td><td><u>id_pustaka</u></td></tr> <tr><td>PK</td><td><u>pengarang</u></td></tr> <tr><td></td><td></td></tr> </table> <p>1..* </p> <p>Menjadi sebuah tabel tersendiri dengan kunci primer (<i>primary key</i>) adalah kunci primer pada entitas dan memiliki atribut dengan nama seperti pada atribut entitas</p>	Pengarang		PK	<u>id_pustaka</u>	PK	<u>pengarang</u>						
Pengarang													
PK	<u>id_pustaka</u>												
PK	<u>pengarang</u>												
	<table border="1" data-bbox="901 1830 1173 2000"> <tr><td colspan="2">Relasi</td></tr> <tr><td>PK</td><td><u>id_E1</u></td></tr> <tr><td>PK</td><td><u>id_E2</u></td></tr> </table>	Relasi		PK	<u>id_E1</u>	PK	<u>id_E2</u>						
Relasi													
PK	<u>id_E1</u>												
PK	<u>id_E2</u>												

 <p>Relasi dengan kardinalitas <i>many to many</i></p>	<table border="1" data-bbox="901 224 1173 280"> <tr> <td></td> <td>atribut_relasi</td> </tr> </table> <p>Menjadi sebuah tabel tersendiri dengan kunci primer adalah atribut yang menjadi kunci primer di kedua entitas yang direlasikannya</p>		atribut_relasi						
	atribut_relasi								
 <p>Relasi dengan kardinalitas <i>one to many</i></p>	<table border="1" data-bbox="901 750 1173 974"> <tr> <td colspan="2">E2</td> </tr> <tr> <td>PK</td> <td><u>id_E1</u></td> </tr> <tr> <td>PK</td> <td><u>id_E2</u></td> </tr> <tr> <td></td> <td>atribut_relasi</td> </tr> </table> <p>Kunci primer entitas yang memiliki hubungan <i>one</i> akan dijadikan kunci primer di entitas yang memiliki hubungan <i>many</i> dengan kata lain, relasi tidak menjadi tabel sendiri</p>	E2		PK	<u>id_E1</u>	PK	<u>id_E2</u>		atribut_relasi
E2									
PK	<u>id_E1</u>								
PK	<u>id_E2</u>								
	atribut_relasi								
 <p>Relasi dengan kardinalitas <i>one to one</i></p>	<table border="1" data-bbox="901 1411 1173 1635"> <tr> <td colspan="2">E2</td> </tr> <tr> <td>PK</td> <td><u>id_E1</u></td> </tr> <tr> <td>PK</td> <td><u>id_E2</u></td> </tr> <tr> <td></td> <td>atribut_relasi</td> </tr> </table> <p>Kunci primer salah satu entitas akan dijadikan kunci asing (<i>foreign key</i>) pada tabel yang lain dan kunci asing itu dijadikan kunci primer juga, dengan kata</p>	E2		PK	<u>id_E1</u>	PK	<u>id_E2</u>		atribut_relasi
E2									
PK	<u>id_E1</u>								
PK	<u>id_E2</u>								
	atribut_relasi								

	lain, relasi tidak menjadi tabel sendiri.
--	---

2.6 Physical Data Model (PDM)

Physical Data Model (PDM) atau Model relasional adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data. Setiap tabel mempunyai sejumlah kolom dimana setiap kolom memiliki nama yang unik beserta tipe datanya. PDM merupakan konsep yang menerangkan detail dari bagaimana data disimpan di dalam basis data. PDM sudah merupakan bentuk fisik perancangan basis data yang sudah siap diimplementasikan ke dalam DBMS sehingga nama tabel juga sudah merupakan nama asli tabel yang akan diimplementasikan ke dalam DBMS.

2.6.1. Simbol-simbol PDM

Berikut adalah simbol-simbol yang ada pada PDM:

Simbol	Deskripsi
Tabel nama_tabel	Tabel yang menyimpan data dalam basis data
Relasi → id_tbl1 = id_fk_tbl2	Relasi antar tabel yang terdiri dari persamaan antara <i>primary key</i> (kunci primer) tabel yang diacu dengan kunci yang menjadi referensi acuan di tabel lain.

BAB 3

PEMBAHASAN

3.1 Studi Kasus

Dalam modul ini akan membahas sebuah studi kasus tentang Sistem Informasi Permintaan Barang di Gudang dalam suatu perusahaan.

1. Deskripsi

Sistem Informasi Permintaan Barang merupakan suatu sistem informasi untuk melakukan proses permintaan barang di gudang suatu perusahaan. Yang meliputi permintaan barang oleh mandor, yang diproses oleh kepala gudang kemudian akan diterima oleh mandor yang meminta.

2. Aturan

Aturan yang harus diatasi dalam sistem informasi yang akan dimodelkan ini adalah sebagai berikut:

- a. Permintaan yang dilakukan mandor dengan menulis pada bon permintaan.
- b. Setiap mandor dapat menulis pada bon permintaan dalam satu waktu yang sama dan boleh lebih dari satu barang.
- c. Satu barang akan disimpan sebagai satu data dengan id yang unik.
- d. Pada bon_permintaan terdapat keterangan digunakannya dari barang yang diminta.

3. Analisa

Permintaan barang pada suatu perusahaan meliputi fungsi-fungsi sebagai berikut:

- a. Validasi users
 - Login
 - Logout
- b. Mengelola data users yang memiliki hak akses untuk mengelola data users, meliputi:
 - Menambah data users baru
 - Mengubah data users
 - Mencari data users
 - Melihat data users

- Menghapus data users
- c. Mengelola data bon_permintaan, meliputi:
 - Menambah data bon_permintaan
 - Mengubah data bon_permintaan
 - Mencari data bon_permintaan
 - Melihat data bon_permintaan
 - Menghapus data bon_permintaan
- d. Mengelola data barang, meliputi:
 - Menambah data barang.
 - Mengubah data barang.
 - Mencari data barang.
 - Melihat data barang
 - Menghapus data barang
- e. Mengelola data kelompok
 - Menambah data kelompok
 - Mengubah data kelompok
 - Mencari data kelompok
 - Melihat data kelompok
 - Menghapus data kelompok
- f. Mengelola data nomor_akun
 - Menambah data nomor_akun
 - Merubah data nomor_akun
 - Mencari data nomor_akun
 - Melihat data nomor_akun
 - Menghapus nomor_akun
- g. Mengelola data pegawai
 - Menambah data pegawai
 - Mengubah data pegawai
 - Mencari data pegawai
 - Melihat data pegawai
 - Menghapus data pegawai

3.2 Studi Kasus ERD

Studi kasus untuk membuat ERD menggunakan sistem informasi permintaan barang dengan deskripsi seperti pada subbab sebelumnya.

3.2.1. Definisi Entitas dan Atribut

No.	Entitas	Atribut
1.	users entitas yang menyimpan data users yang berhak login ke aplikasi untuk mengelola data	username atribut untuk melakukan proses login password merupakan kata sandi dari masing-masing users untuk login pada sistem hak_akses atribut untuk mengetahui hak akses users yang berhak mengelola data pada sistem atau tidak (biasanya disebut sebagai admin/administrator/yang mengurus administrasi)
2.	pegawai entitas yang menyimpan data mandor	id_pegawai merupakan suatu atribut yang menjadi identitas pegawai nama_pegawai merupakan atribut dari nama pegawai alamat atribut alamat pegawai no_telp atribut nomer telepon pegawai
3.	jabatan entitas yang menyimpan data jabatan	kode_jab merupakan atribut yang menjadi identitas dari jabatan. jabatan merupakan atribut keterangan jabatan
4.	barang	kode_barang

	entitas yang menyimpan data-data barang	merupakan suatu atribut untuk menjadi identitas dari barang
		nama_barang merupakan atribut nama barang
		satuan merupakan atribut satuan dari barang.
		stok merupakan atribut untuk keterangan stok barang.
5.	kelompok entitas yang menyimpan data kelompok barang	kode_kel merupakan suatu atribut untuk menjadi identitas dari kelompok
		kelompok Merupakan suatu atribut untuk mneyimpan keterangan dari kelompok
6.	nomor_akun entitas yang menyimpan detail dari penggunaan barang nomor akun	no_akun atribut yang menjadi identitas dari nomor_akun
		deskripsi atribut yang menjadi keterangan dari masing-masing nomor akun
7.	bon_permintaan entitas yang menyimpan permintaan barang dari mandor	id_bon atribut yang menjadi identitas dari bon_permintaan
		id_pegawai atribut <i>foreign key</i> dari tabel pegawai
		kode_barang atribut foreign key dari tabel barang
		no_akun atribut <i>foreign key</i> dari tabel nomor_akun
		tgl_minta

		atribut tanggal permintaan
		jumlah_barang atribut dari jumlah barang yang akan diminta
		keterangan Atribut dari keterangan digunakannya bahan, dari bon_permintaan
		status merupakan atribut dari status bon_permintaan

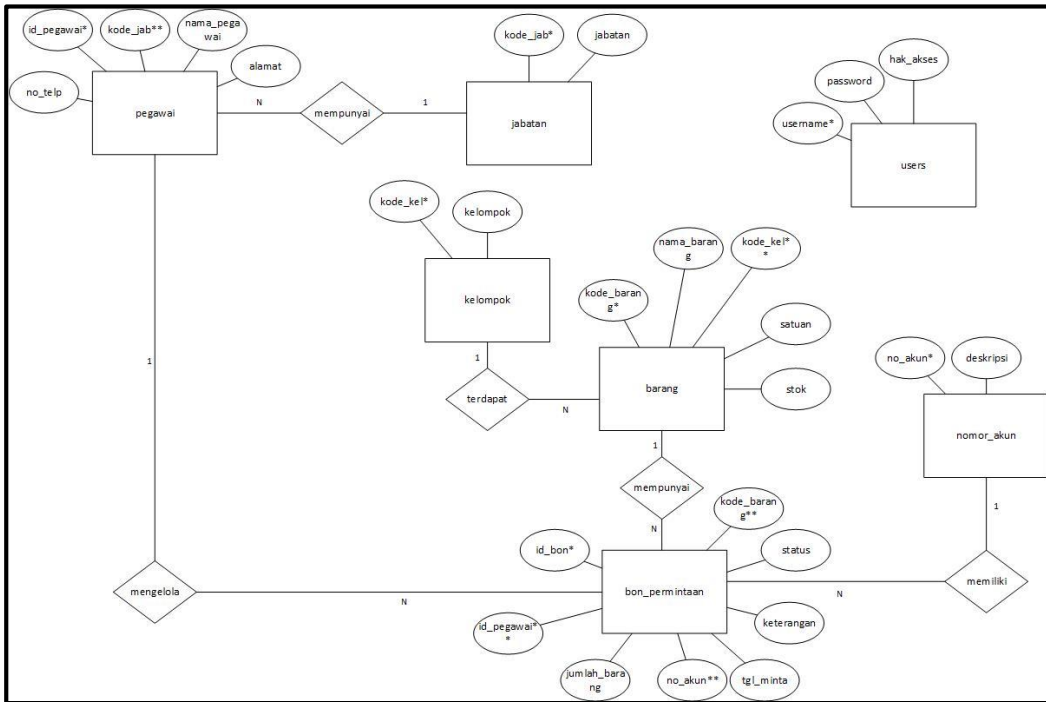
3.2.2. Definisi Relasi

No.	Relasi	Atribut
1.	mengelola dalam bon_permintaan	Merupakan relasi antara entitas pegawai dan entitas bon_permintaan dimana mengelola memiliki makna bahwa bon_permintaan dikelola oleh pegawai yang disimpan pada entitas bon_permintaan. Kardinalitas antara entitas pegawai dan entitas bon_permintaan adalah one to many karena seorang pegawai dapat terlibat dengan banyak bon_permintaan.
2.	mempunyai dalam bon_permintaan	Merupakan relasi antara barang dengan bon_permintaan. Dimana maksud dari mempunyai pada bon_permintaan adalah bon_permintaan dimiliki oleh barang yang disimpan pada entitas bon_permintaan. Kardinalitas dari barang dengan bon_permintaan adalah one to many

		karena suatu barang dapat terlibat dengan banyak bon_permintaan.
3.	memiliki dalam bon_permintaan	<p>Merupakan relasi antara nomor_akun dengan bon_permintaan. Dimana maksud dari memiliki pada bon_permintaan adalah bon_permintaan dimiliki oleh nomor_akun yang disimpan pada entitas bon_permintaan.</p> <p>Kardinalitas dari nomor_akun dengan bon_permintaan adalah one to many karena suatu nomor_akun dapat terlibat dengan banyak bon_permintaan.</p>
4.	mempunyai dalam jabatan	<p>Merupakan relasi antara pegawai dengan jabatan. Dimana maksud dari mempunyai pada jabatan adalah pegawai mempunyai jabatan yang disimpan pada entitas pegawai.</p> <p>Kardinalitas dari pegawai dengan jabatan adalah many to one karena banyak pegawai dapat terlibat dengan satu jabatan.</p>
5.	terdapat dalam barang	<p>Merupakan relasi antara kelompok dengan pegawai. Dimana maksud dari terdapat pada barang adalah kelompok mempunyai barang-barang yang disimpan pada entitas barang.</p> <p>Kardinalitas dari barang dengan kelompok adalah many to one karena</p>

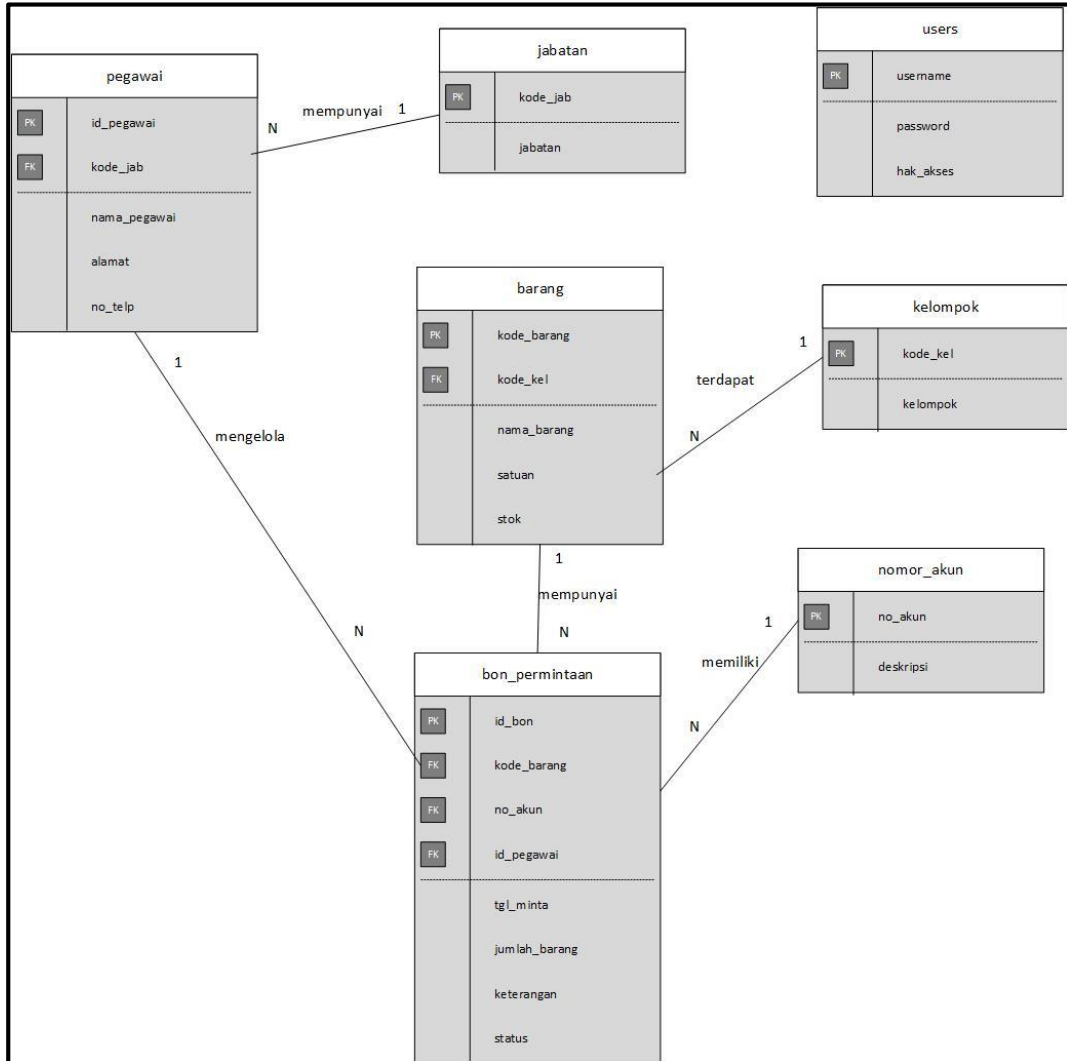
	banyak barang dapat terlibat dengan satu kelompok.
--	--

3.2.3. Diagram ER



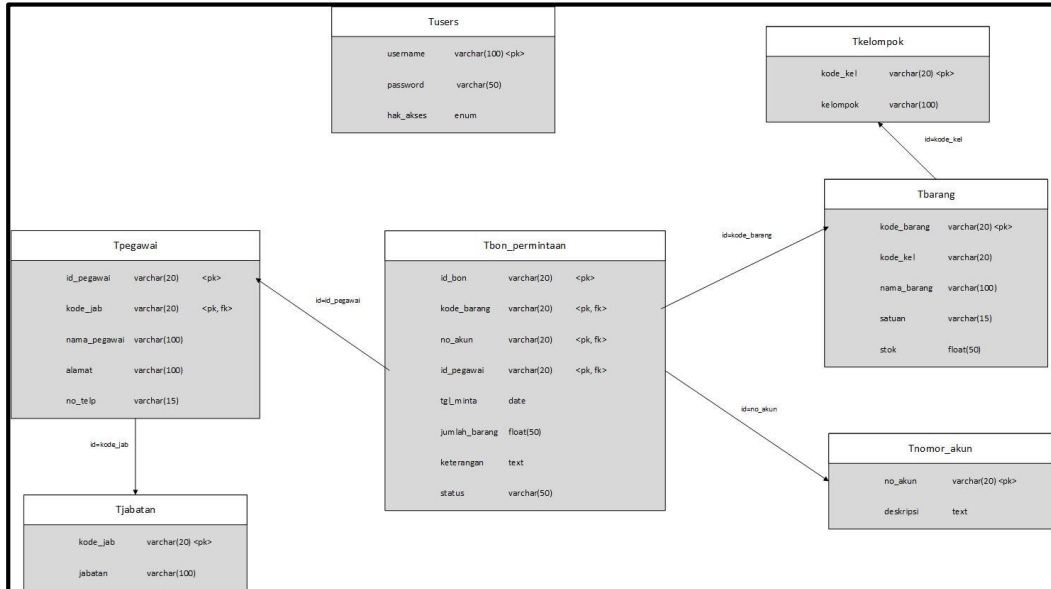
3.3 Studi Kasus CDM

Berikut adalah CDM dari studi kasus Sistem Informasi Permintaan Barang pada studi kasus sebelumnya.



3.4 Studi Kasus PDM

Berikut adalah PDM dari studi kasus Sistem Informasi Permintaan Barang seperti pada kasus bab sebelumnya.



BAB 4

PHPMYADMIN

4.1 Pengertian PHPMyAdmin

PHPmyadmin adalah script website berbasis PHP yang berfungsi untuk manajemen database Mysql. PHPmyadmin dimulai perkembangannya oleh Tobias Ratchiller pada sekitar tahun 1998 kemudian dilanjutkan oleh penerus lainnya.

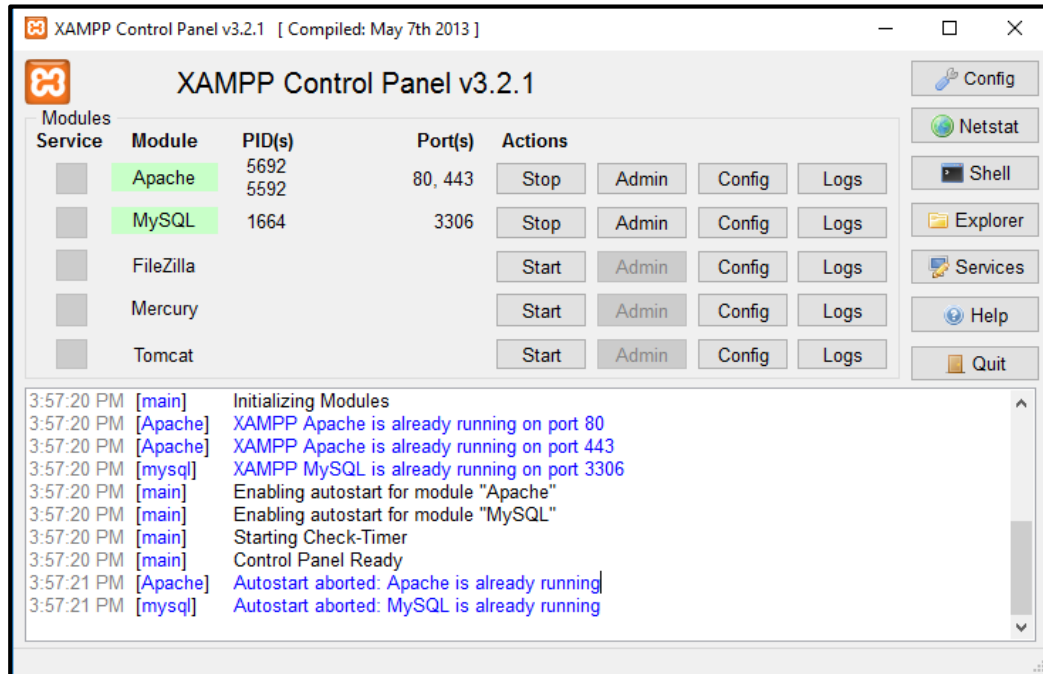
Adapun fitur-fitur PHPmyadmin yaitu:

1. Melihat dan menghapus databases, tables, views, columns dan indexes.
2. Menampilkan multi results melalui stored procedures or queries
3. Menciptakan, copy, menghapus, me-rename dan merubah databases, tables, columns dan indexes.
4. Maintenance server, databases dan tables, berdasarkan konfigurasi server
5. Menjalankan, edit dan mem-bookmark perintah *SQL*.
6. Me-load file text kedalam tabel.
7. Membuat dan membaca dump tabel.
8. Export data ke berbagai jenis format seperti *CSV*, *XML*, *PDF*, *ISO/IEC – OpenDocument* Text dan Spreadsheet, Microsoft Word 2000, and LATEX formats.
9. Import data dan struktur *MySQL*.
10. Manajemen multiple servers.
11. Manajemen MySQL users dan perijinan.
12. Mengecek integritas tabel MyISAM.
13. Menciptakan layout grafik database dalam bentuk *PDF*.
14. Melakukan pencarian secara global pada database.
15. Mendukung tipe tabel InnoDB dan foreign keys.

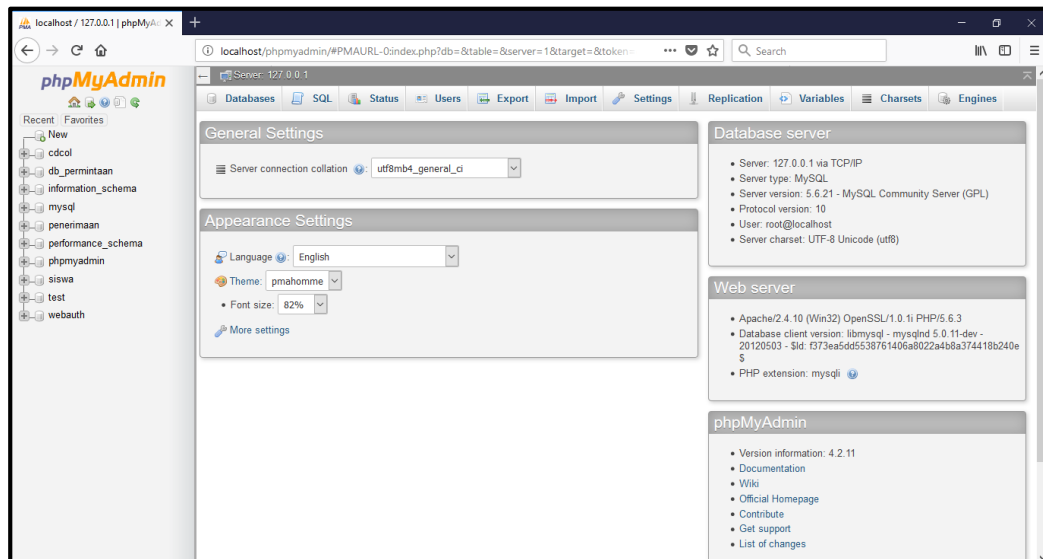
4.2 Membuat database dari Mysql PHPMyAdmin

Berikut adalah cara mengakses PHPMyAdmin:

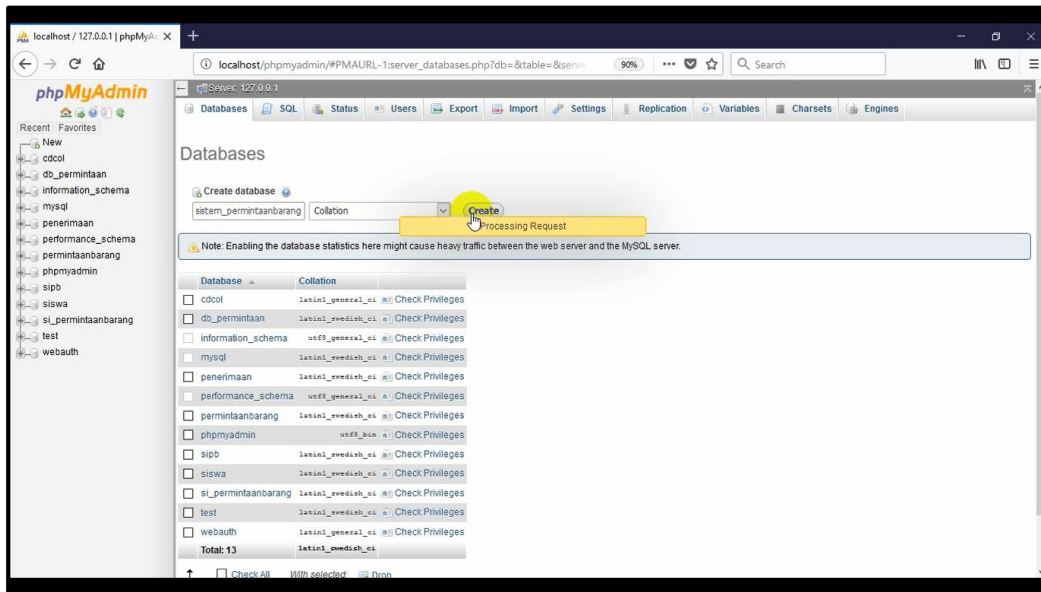
1. Mengaktifkan Apache dan MySQL pada Aplikasi Xampp



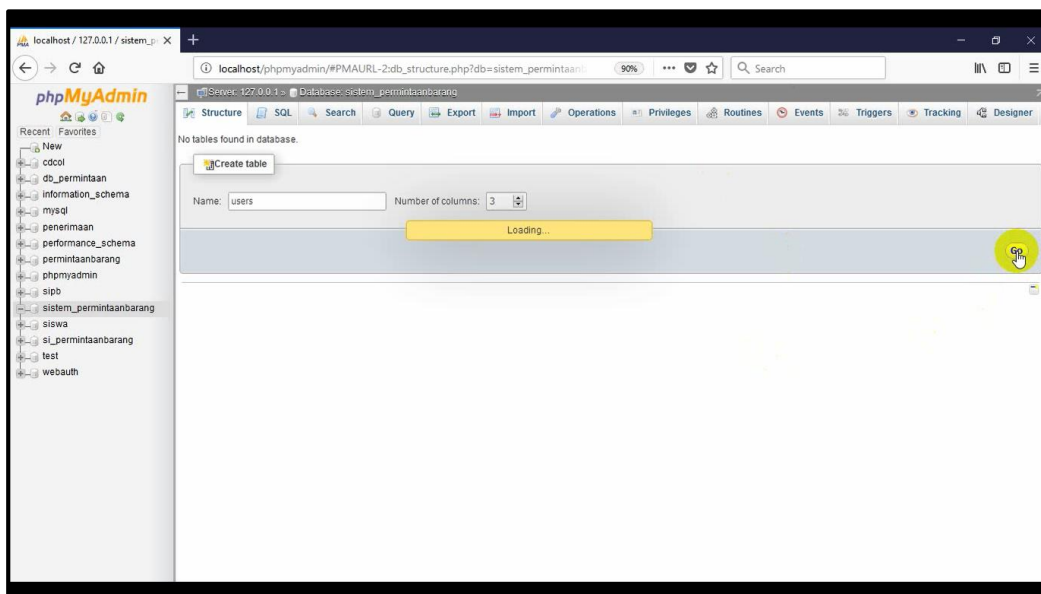
2. Mengetikan source kode untuk masuk ke PHPMyAdmin melalui penjelajah browser. Dengan *Source kode* <http://localhost/phpmyadmin>



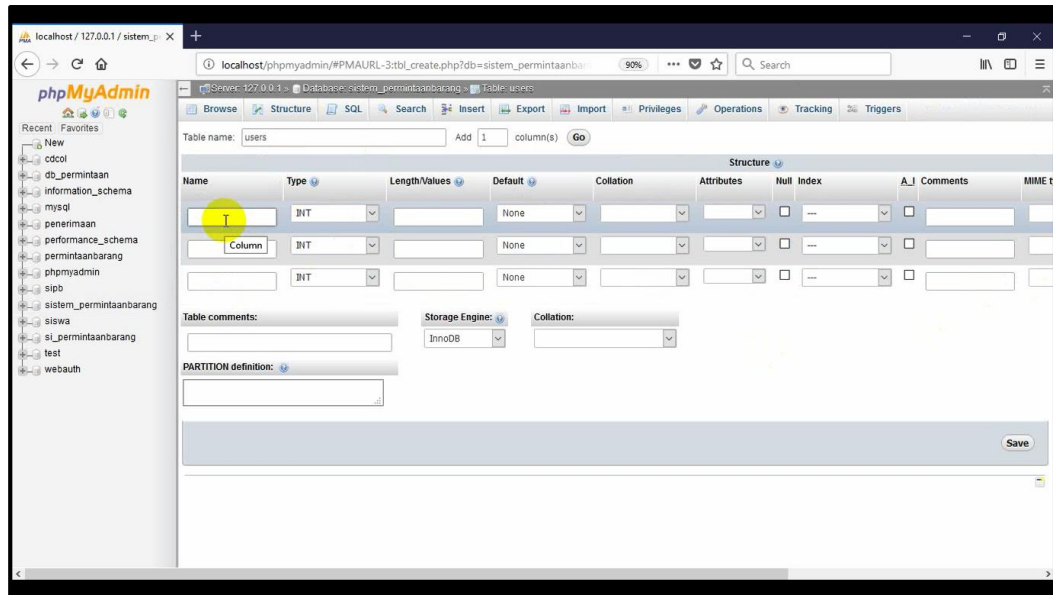
3. Buat database, tulis nama database dan jumlah kolomnya(atribut), kemudian klik go.



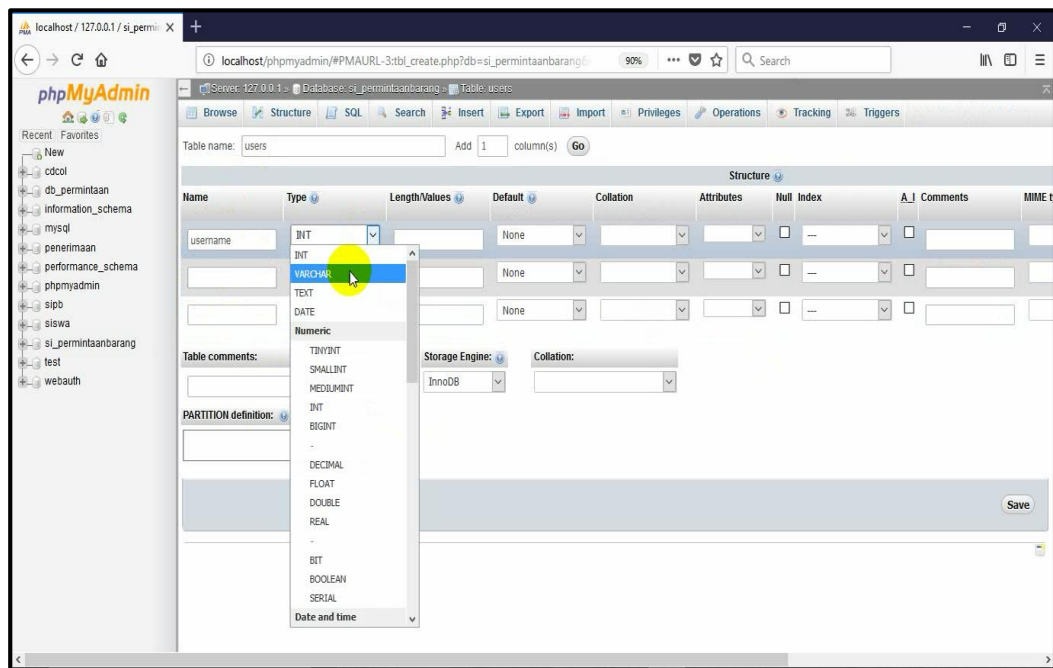
4. Buat table, keudian isi berapa banyak kolom yang diperlukan sesuai banyaknya tribute dari entitas pada ERD.



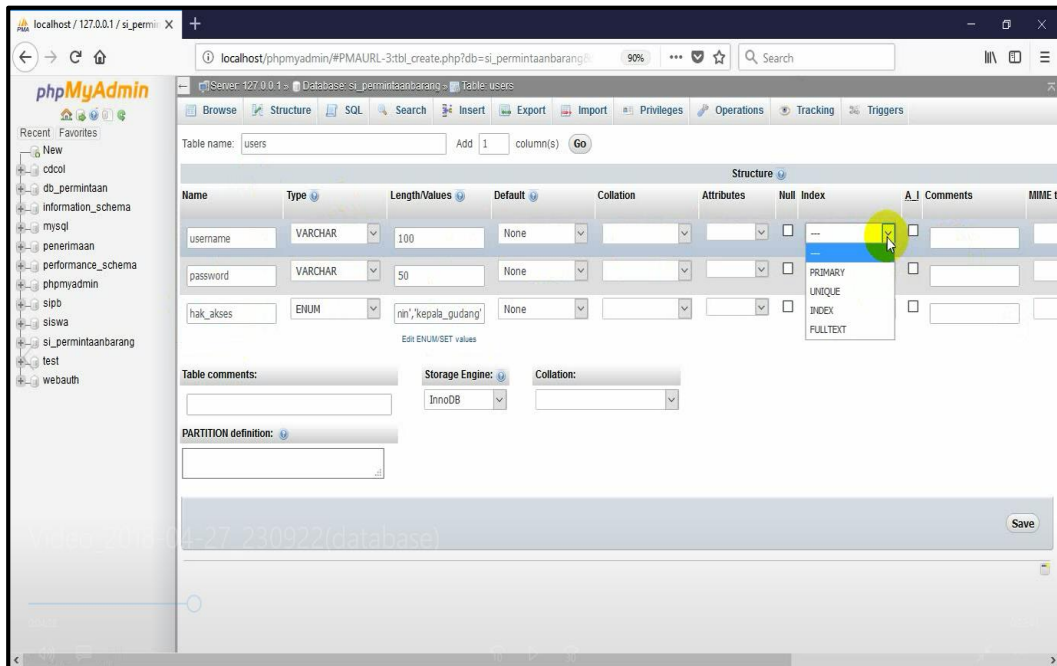
5. Tulis nama-nama atributnya serta tentukan type data dan length.



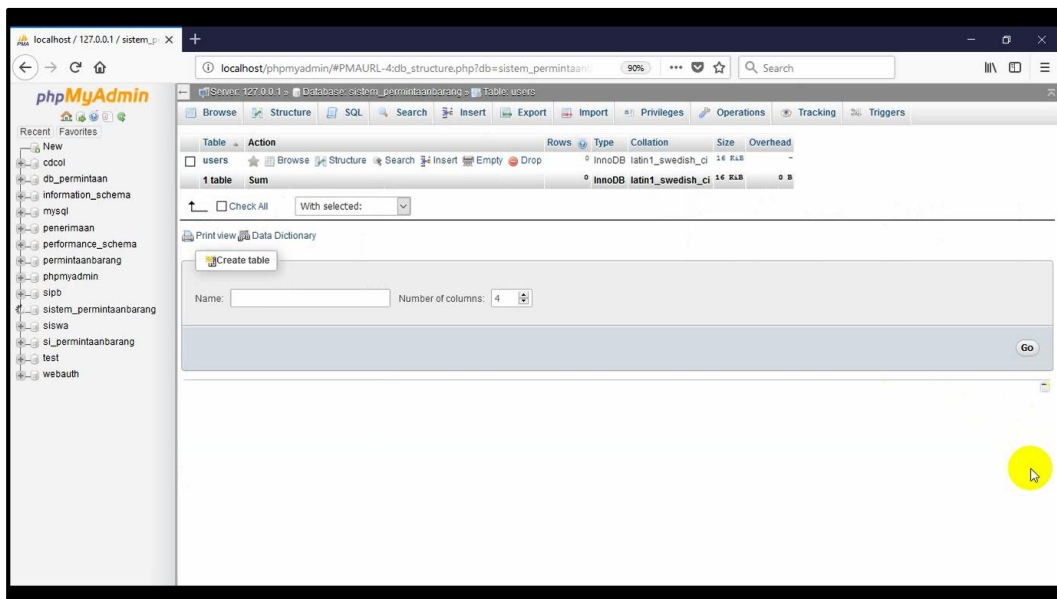
- Pilih type data dari setiap atribut.



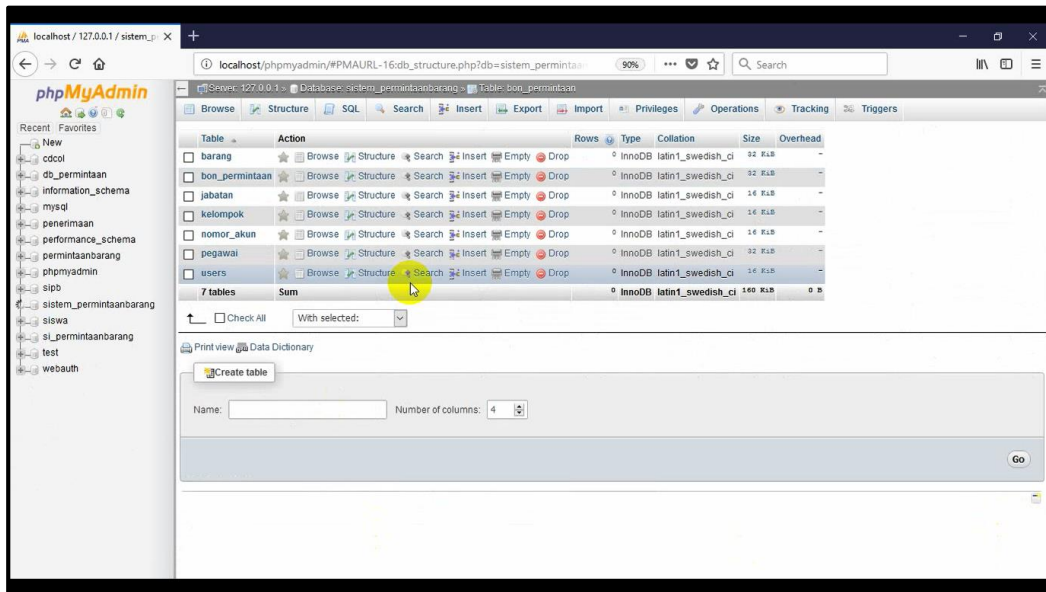
- Untuk menentukan bahwa atribut itu primary key dengan cara klik panah pada tabel index kemudian pilih primary. Jika atributnya merupakan foreign key maka pilih "index". Lalu klik save.



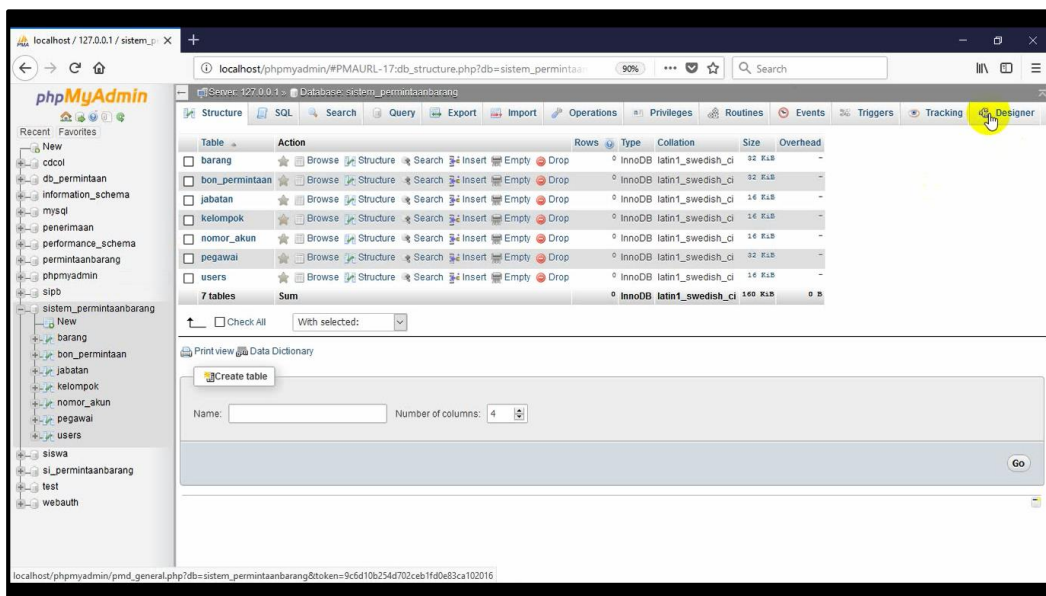
6. Hasil dari tabel yang sudah dibuat.



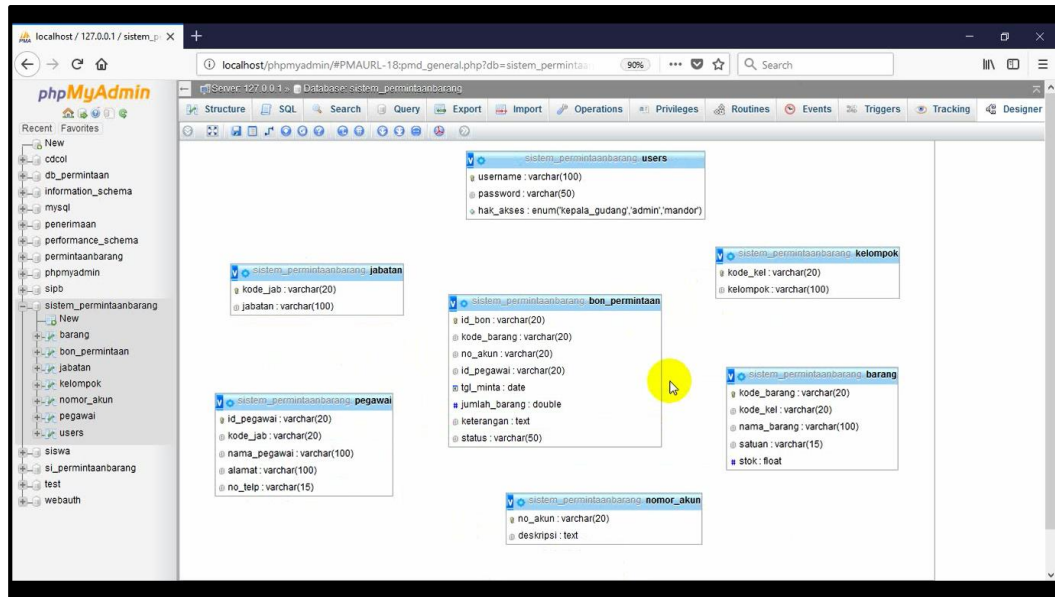
- Dan ini adalah 7 tabel yang sudah saya buat.



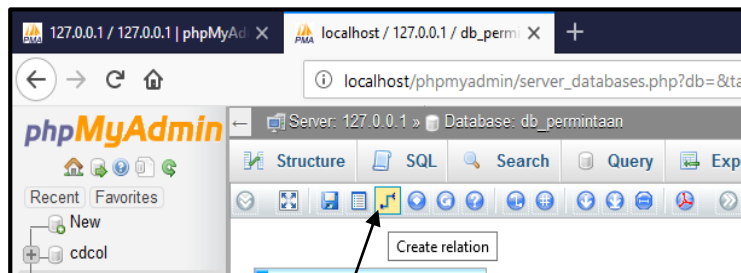
7. Untuk memudahkan dalam penginputan data, maka perlu direlasikan tabel-tabel yang saling berhubungan. Dengan cara klik “designer”.



8. Maka akan tampil tabel-tabel yang sudah dibuat sebelumnya.

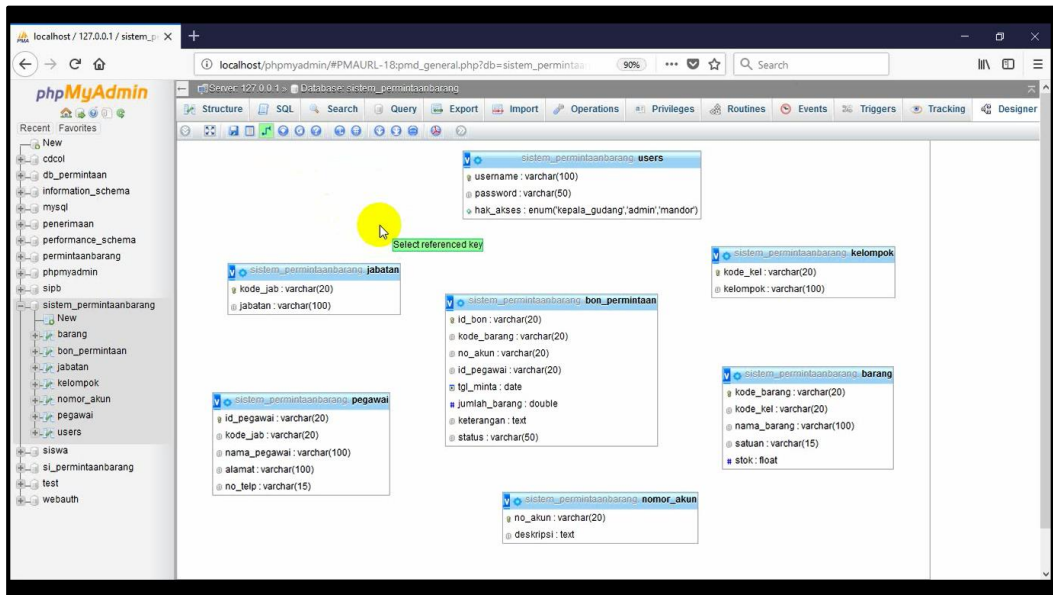


9. Untuk merelasikannya klik “create relation”

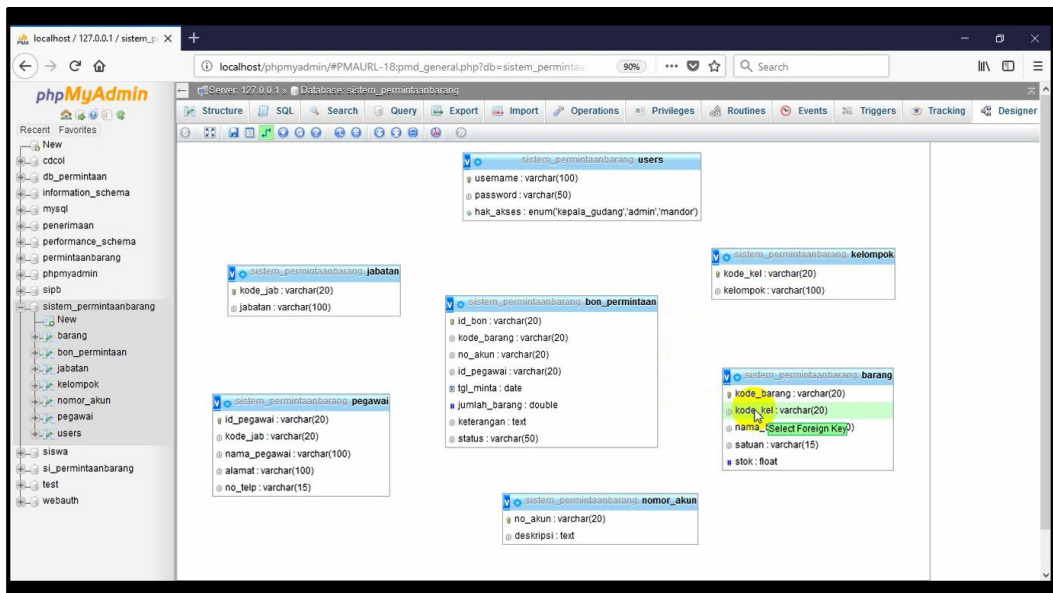


10. Tentukan tabel mana yang ingin direlasikan. Pada contoh berikut “kode_barang” pada tabel “barang” adalah primary key dan menjadi foreign key pada tabel “bon_permintaan”. Maka saya akan merelasikannya setelah memilih “create relation” lalu pilih primary key nya kemudian pilih foreign key-nya.

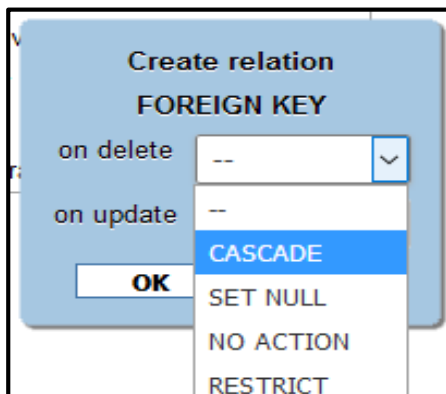
- Pilih primary key-nya.



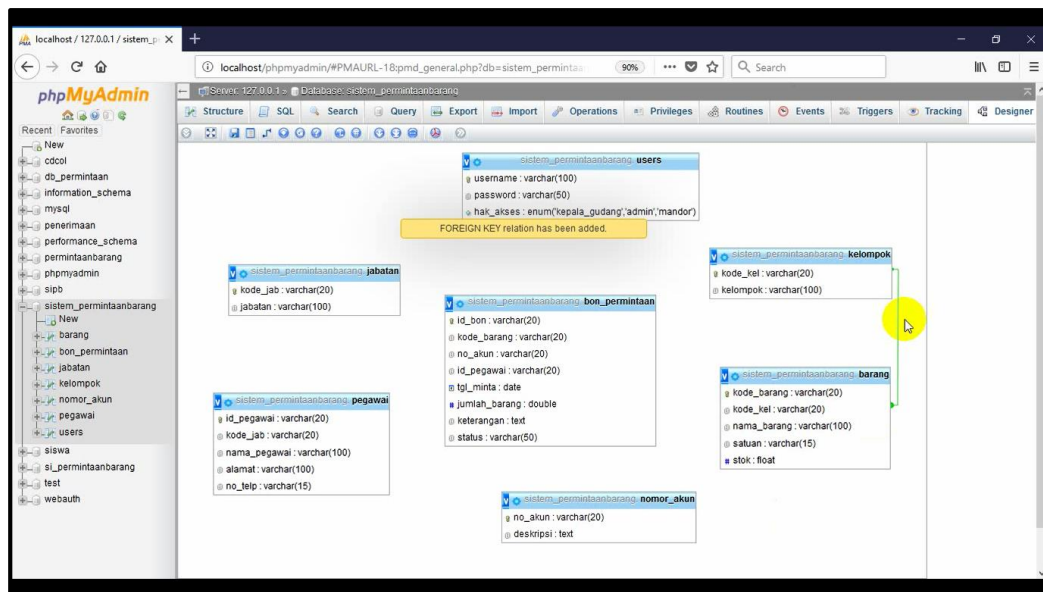
- Pilih foreign key-nya.



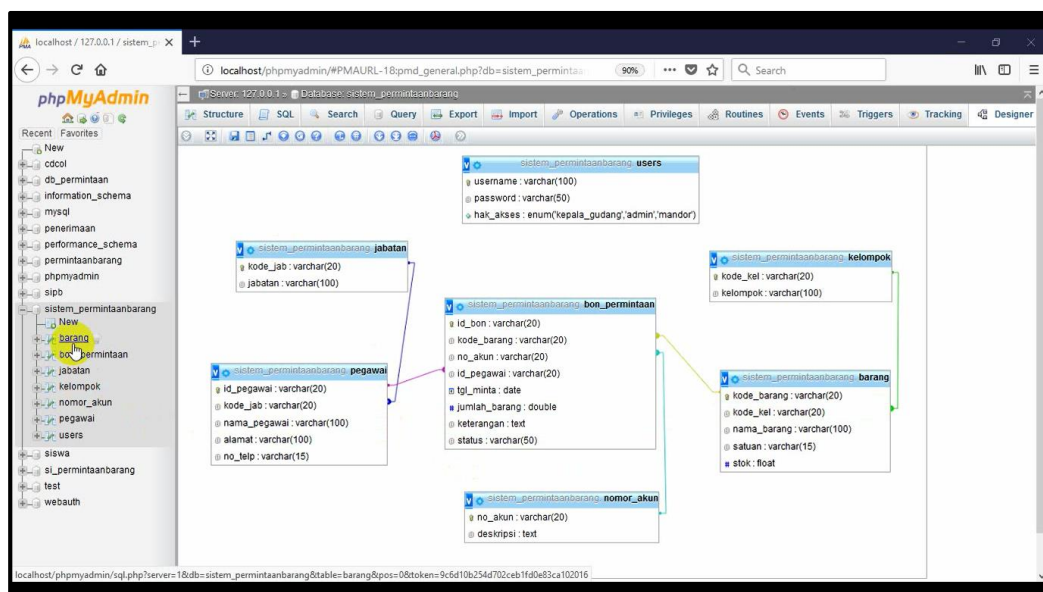
11. Jika sudah dipilih maka akan muncul jendela berikut, klik panah yang ada disamping, pilih cascade pada “on delete” dan “on update”. Lalu klik “ok”.



12. Maka akan terbentuk sebuah relasi antara tabel “barang” dengan tabel “bon_permintaan” yang atributnya “kode_barang”.

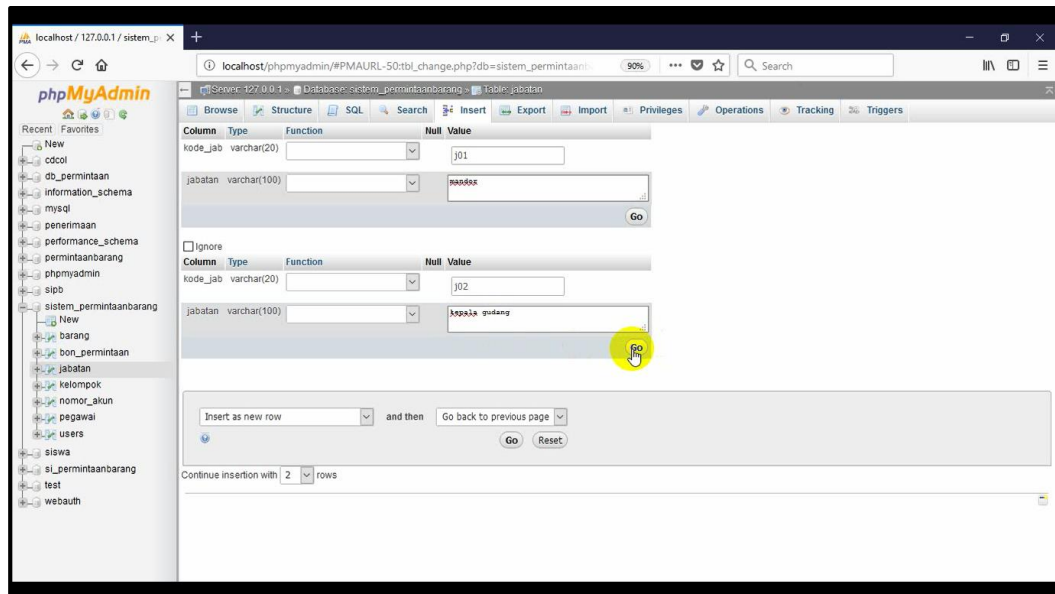



13. Kemudian buatlah relasi tabel-tabel yang lain seperti langkah tadi. Dan ini adalah hasil relasi dari database saya. Seperti pada gambar berikut.

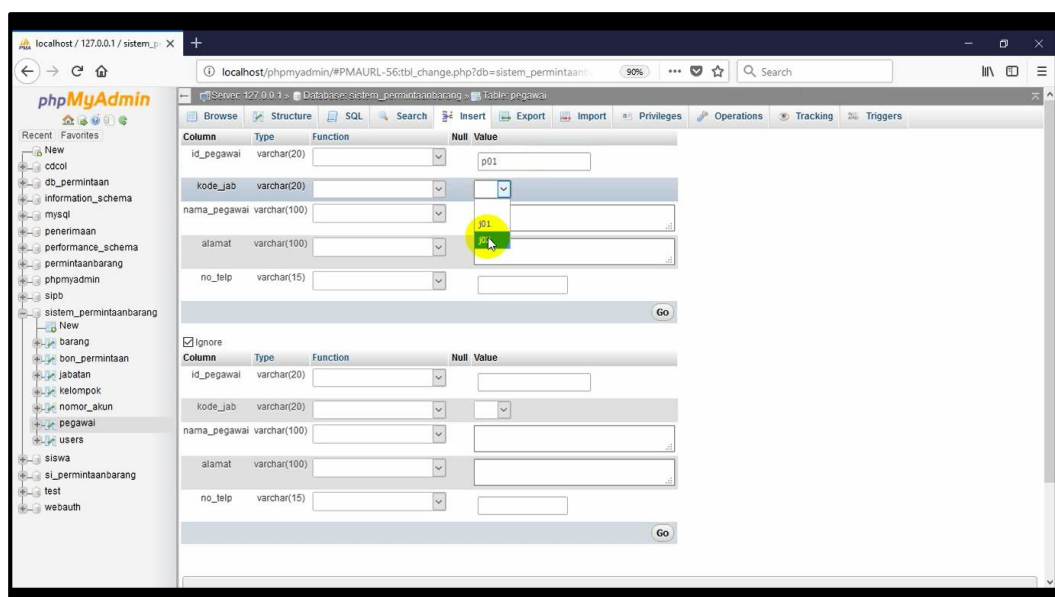


14. Untuk membuktikan bahwa tabelnya sudah berelasi yaitu isi data pada tiap-tiap tabel yang berelasi. Saya akan mencontohkan 2 tabel yang saling berelasi. Pada tabel “jabatan”, tabel “pegawai”. Disini yang menjadi tabel master adalah tabel (jabatan). Maka data yang perlu diisi terlebih dahulu adalah tabel master. Berikut contoh penginputan datanya.

a. Penginputan data pada tabel “jabatan”.



b. Berikut adalah contoh pengisian pada tabel “pegawai” yang mana ini sudah berelasi dengan tabel master di atas. Terlihat pada saat ingin menginputkan kode_jabatan terdapat tanda  yang menandakan bahwa ada pilihan data di dalamnya yang sudah diinputkan pada tabel master.



4.3 Import database dari tools power designer

Dalam mengimport database dari tools power designer, maka perlu dibuat CDM dan PDM terlebih dulu. Berikut adalah caranya:

4.3.1 Membuat CDM

Langkah-langkah menjalankan aplikasi Sybase power designer:

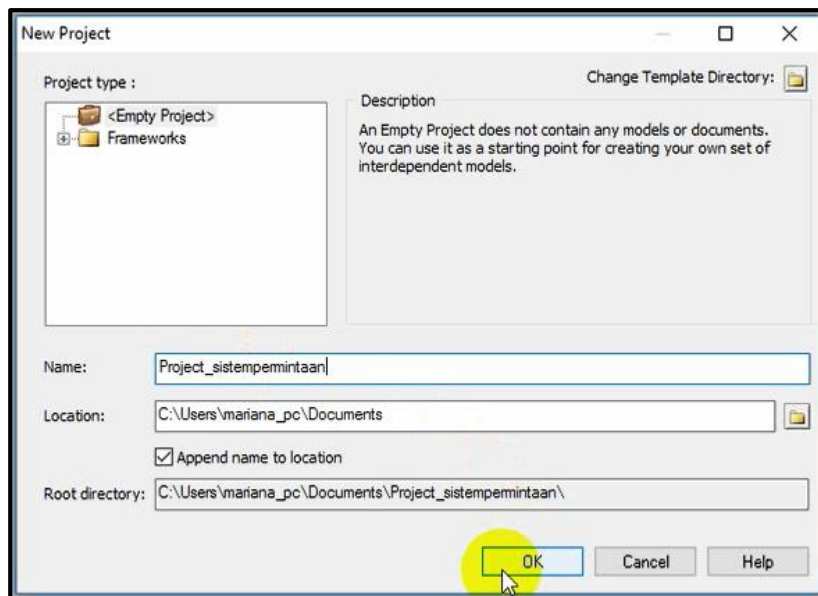
1. Klik di ikon Power Designer pada desktop, atau bisa juga di jalankan dari **Start-Menu-All Program-Sybase PowerDesigner.**



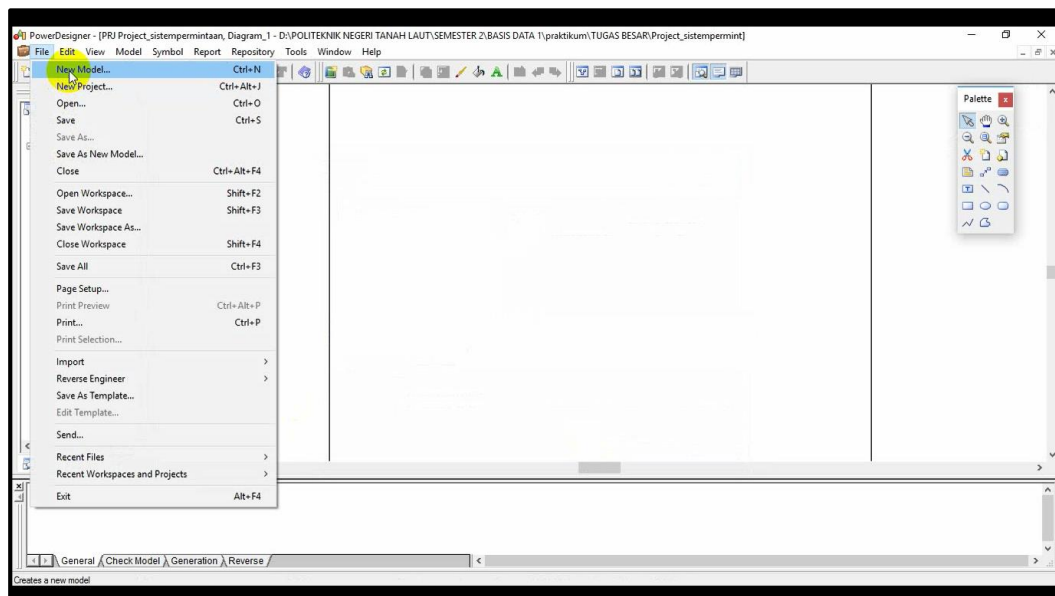
2. Akan tampil jendela berikut, pilih new project.



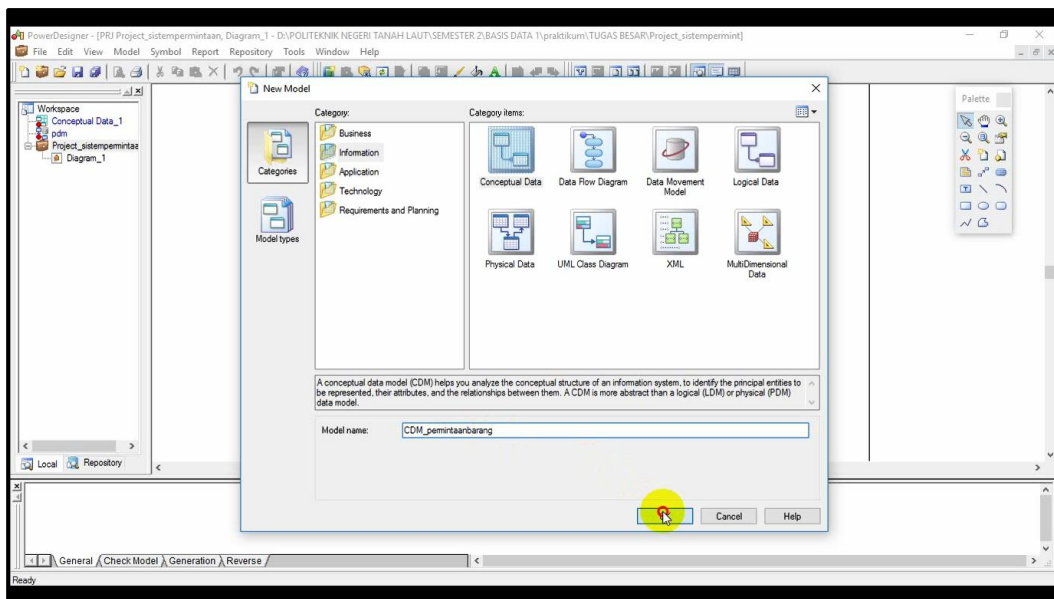
3. Beri nama project yang ingin dibuat dan atur letak penyimpanannya, lalu klik OK



4. Kemudian klik *file-new model*.



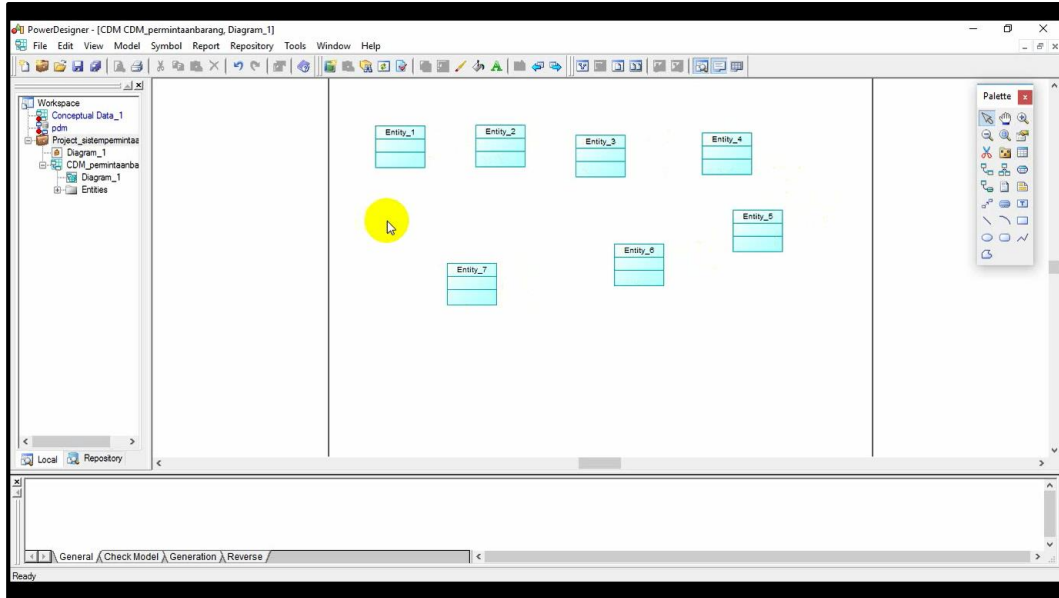
5. Maka kan tampil jendela berikut. Pilih *Information-Conceptual Data-Model name* (ketik nama model yang ingin dibuat)-OK.



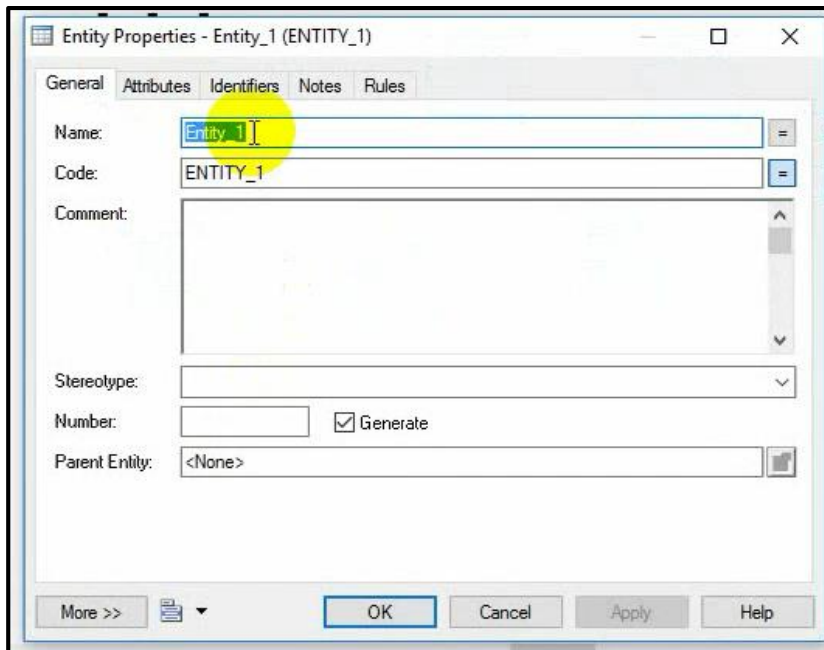
6. Pilih entity untuk membuat entitas.

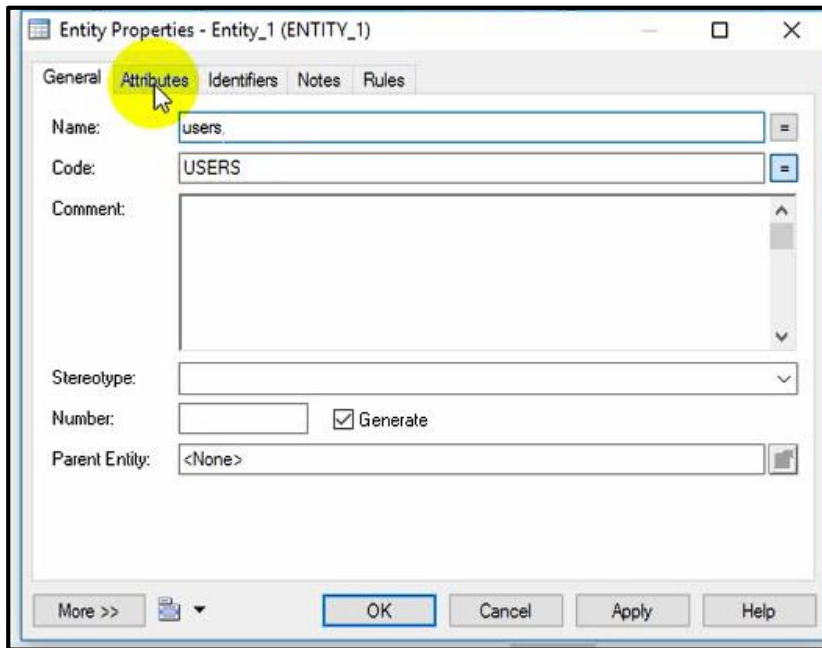


7. Buat beberapa entitas sesuai yang diperlukan

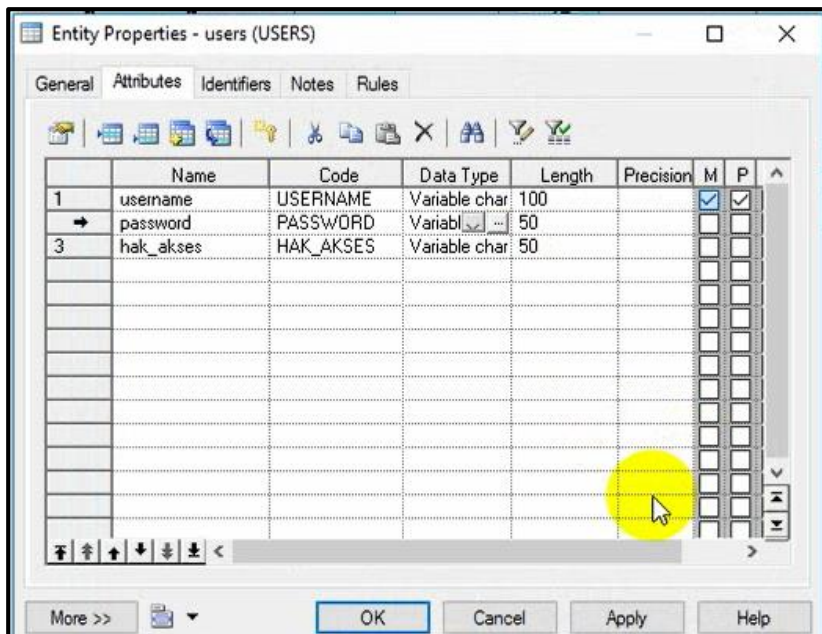


8. Double klik pada entitas, kemudian beri nama entitas.

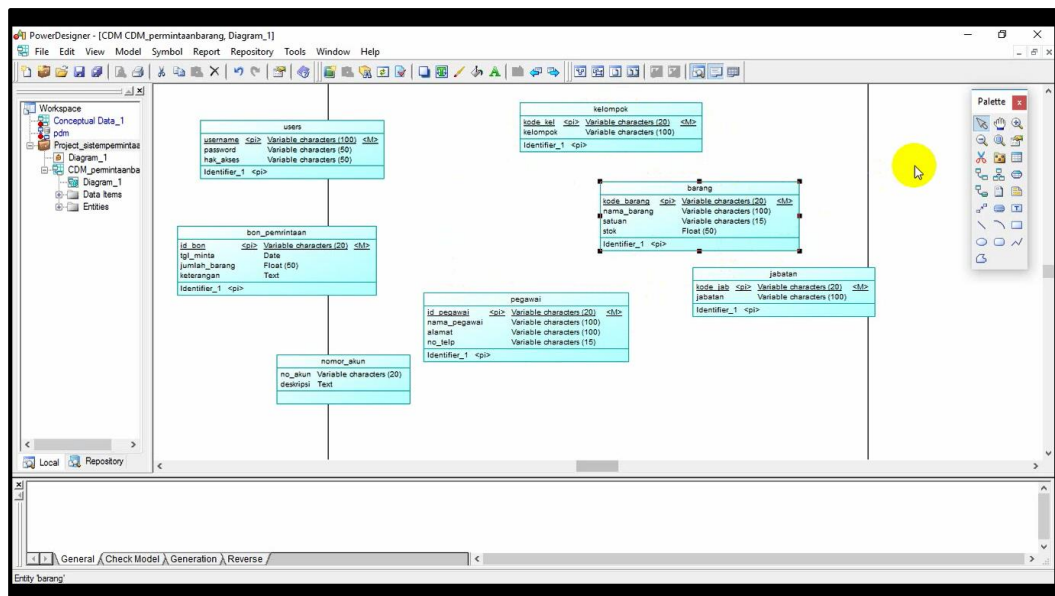




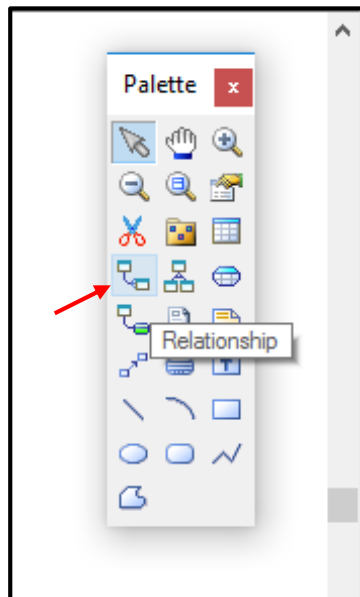
9. Pilih tool *attributes* akan tampil jendela berikut. Tentukan nama, type data, length dari setiap attributes.



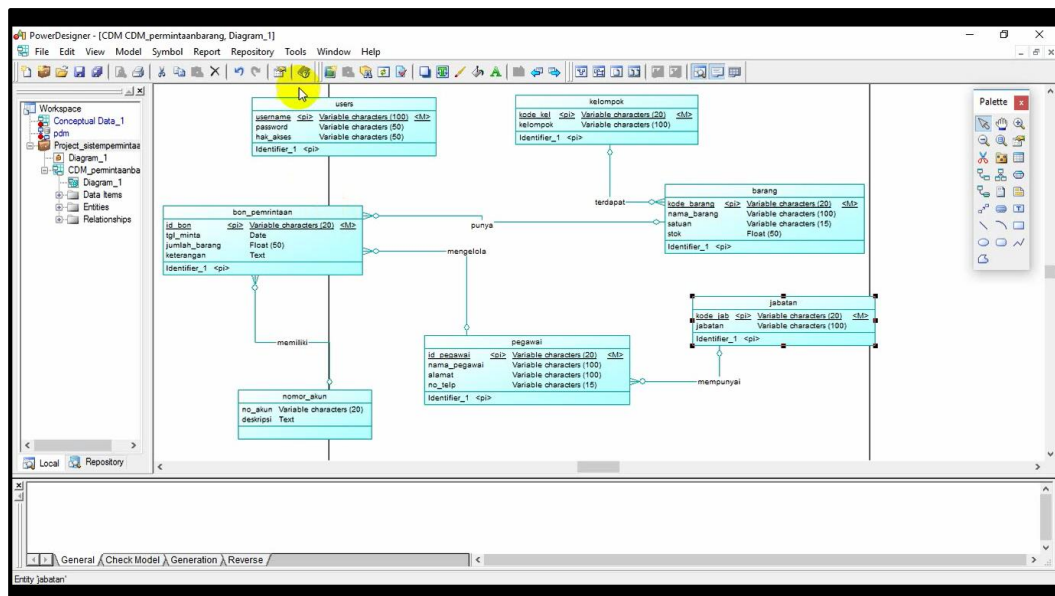
12. Merupakan contoh hasil CDM dari studi kasus yang sudah saya buat sebelumnya.



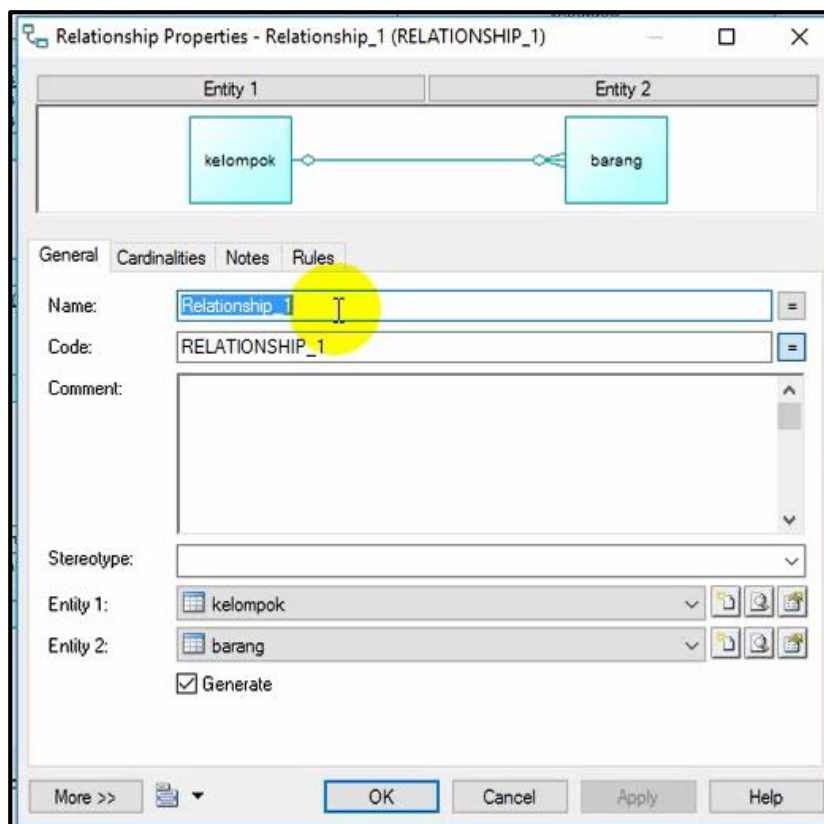
13. Pilih relationship untuk merelasikan masing-masing entitas.



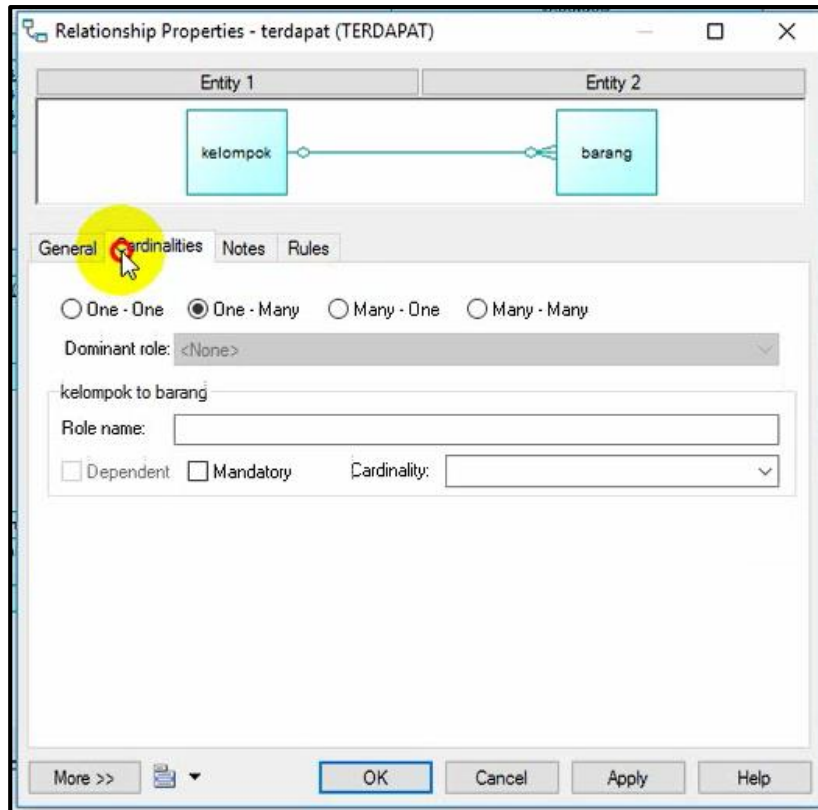
14. Drag **relationship** dari satu tabel ke tabel lain yang berhubungan. Berikut adalah hasil dari relsai antar entitas dari CDM yang telah dibuat.



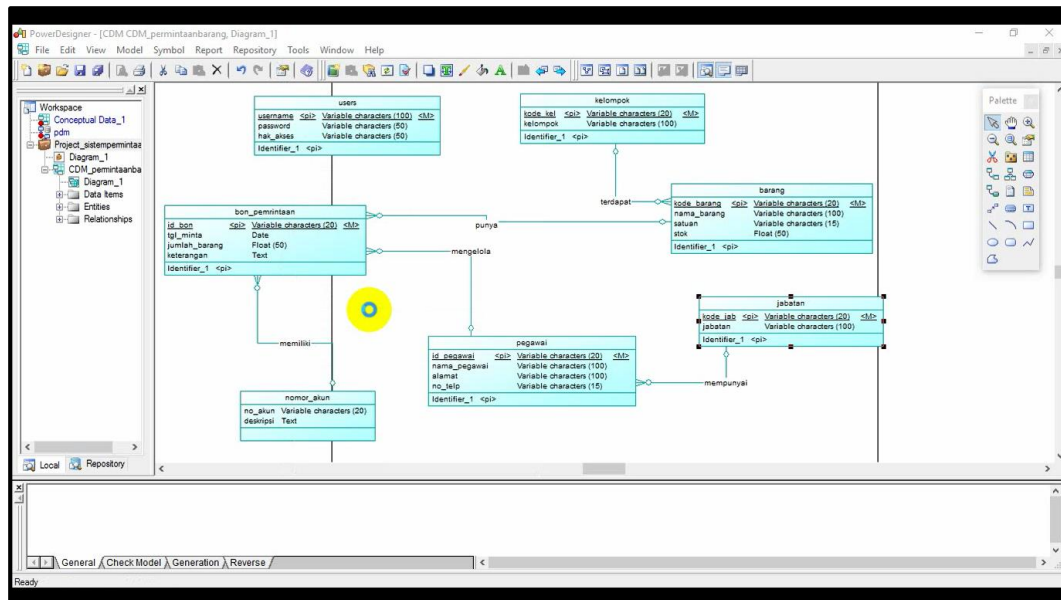
15. Double klik pada relationship, lalu beri nama pada relasi yang terhubung.



16. Klik cardinalitas untuk menentukan cardinalitas dari relasi. Klik OK.



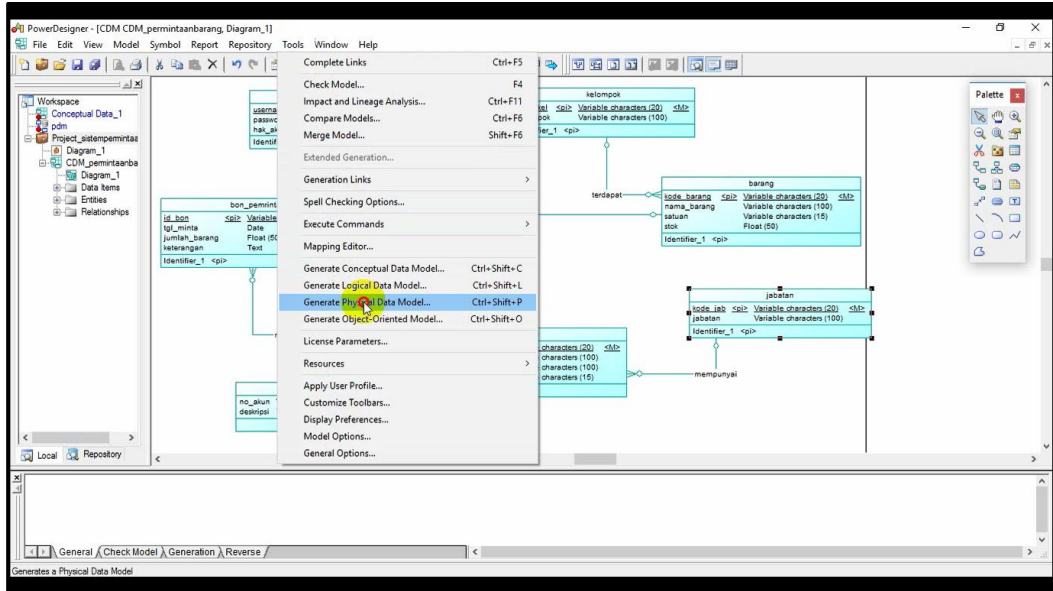
17. Berikut adalah hasil CDM yang sudah selesai.



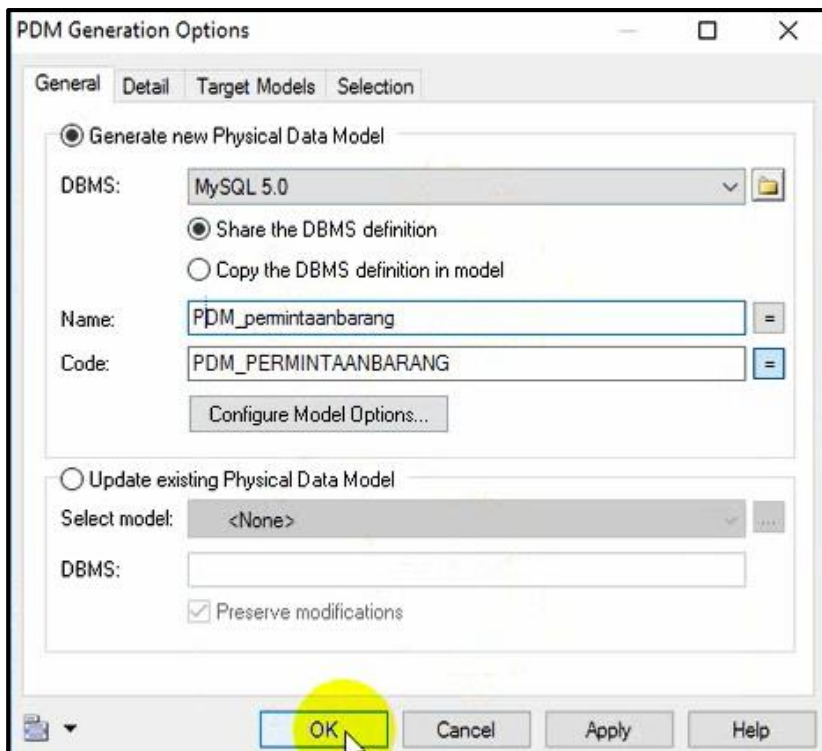
4.3.2 Generate dari CDM ke PDM

Berikut langkah-langkah mengubah hasil CDM ke bentuk PDM.

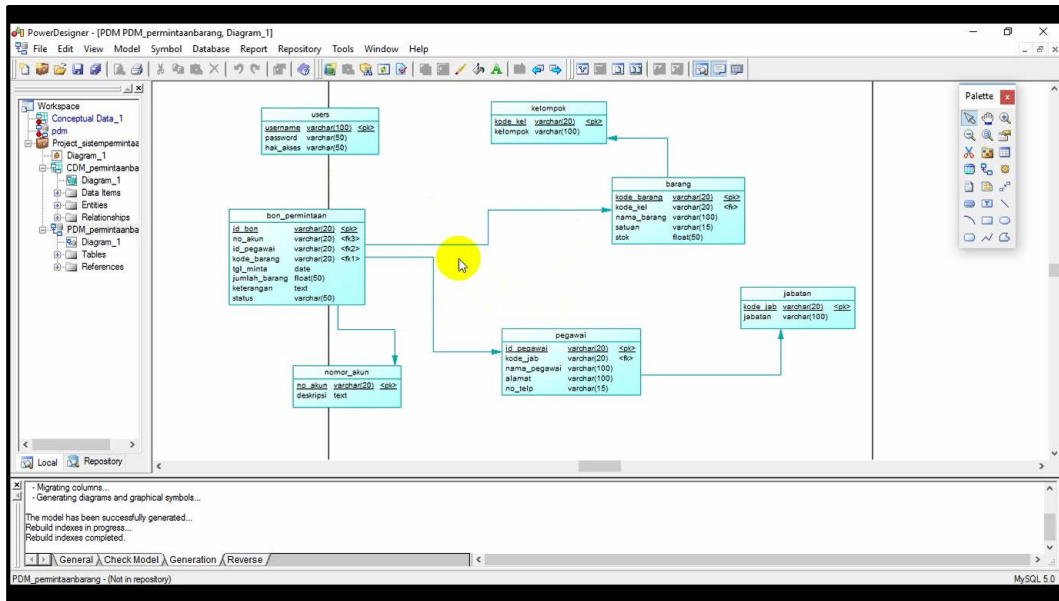
1. Buka CDM yang sudah dibuat. Pilih **tools-generate physical data model**.



2. Akan muncul jendela berikut, beri nama pada file PDM. Lalu OK.



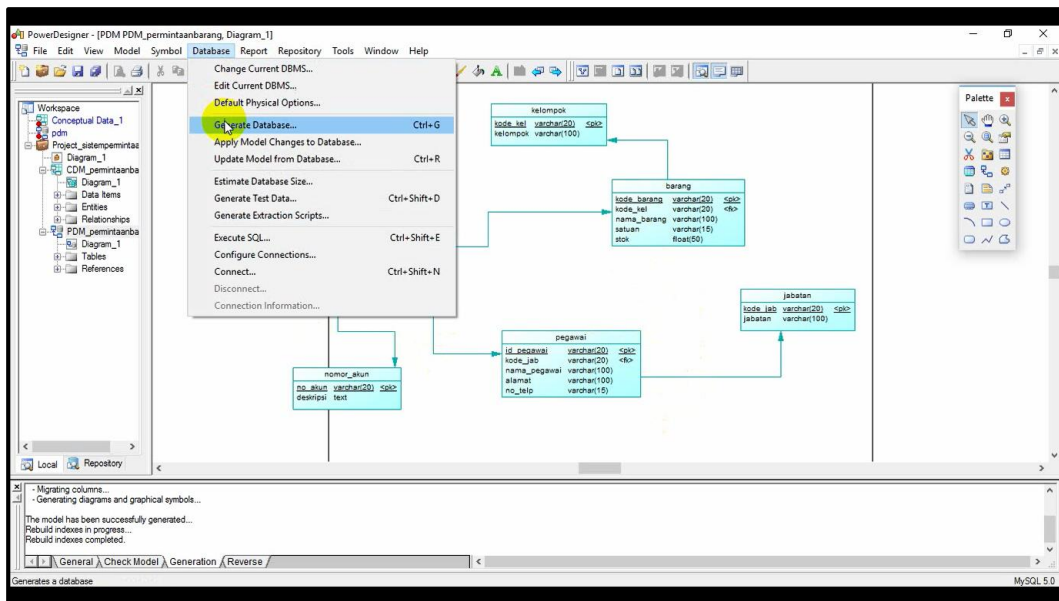
3. Hasil dari generate PDM



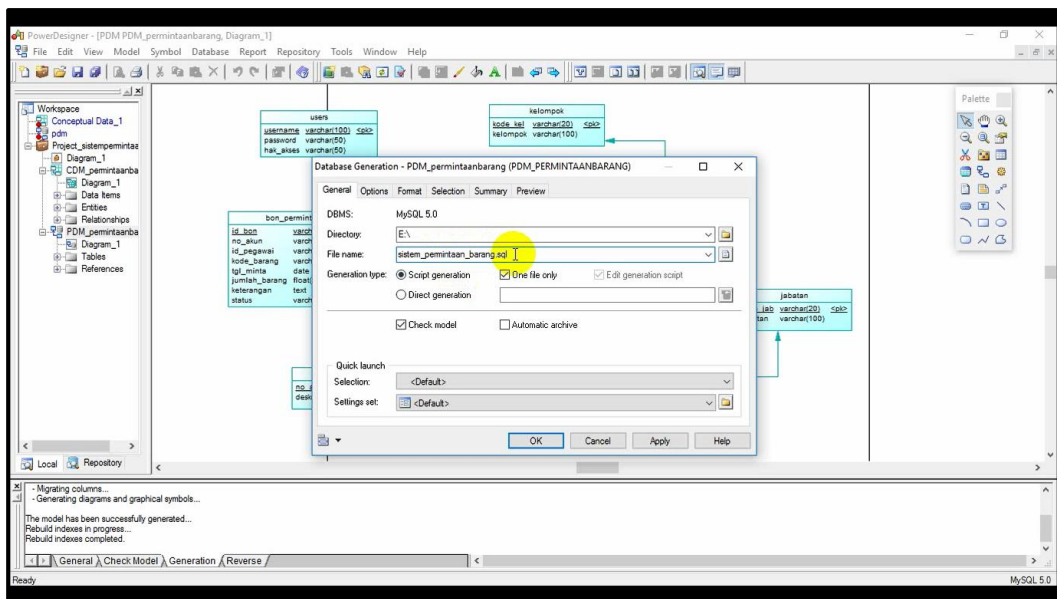
4.3.3 Generate Database dari hasil PDM

Berikut langkah-langkah untuk generate database ke Mysql.

1. Buka file PDM yang sudah dibuat. Pilih tool **Database-generate database**.

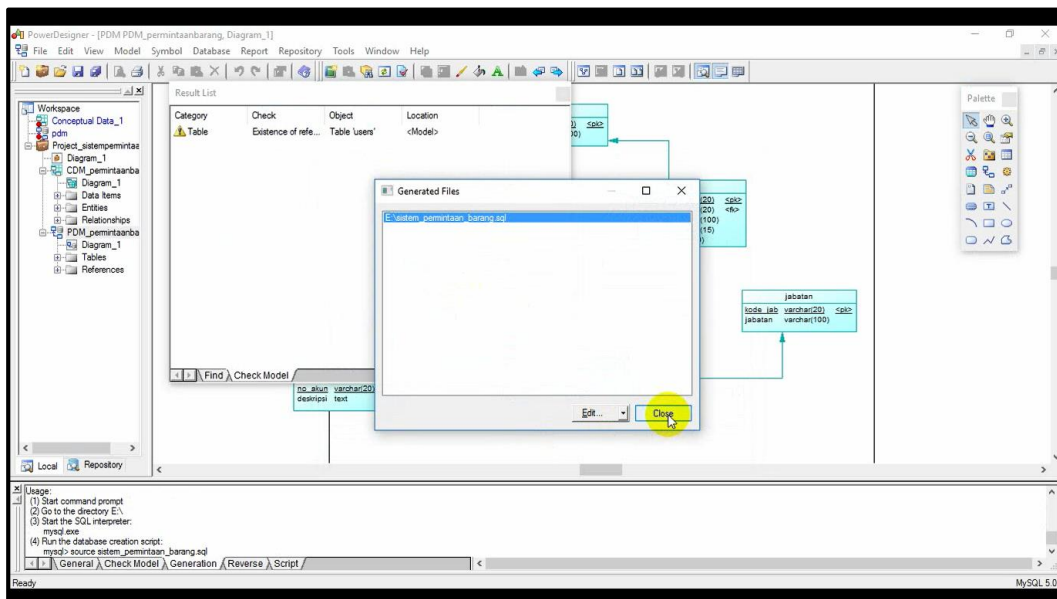


2. Akan muncul jendela berikut. pilih tempat penyimpanan dan beri nama file yang sudah berekstensi “.sql” lalu klik OK.

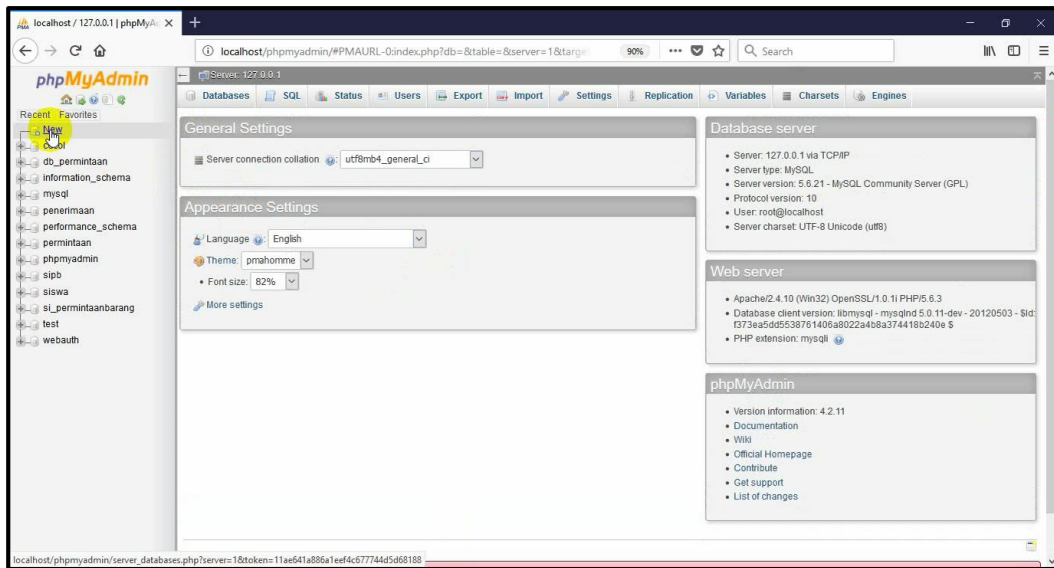


3. Dan akan tampil jendela berikut, yang menandakan bahwa generate berhasil.

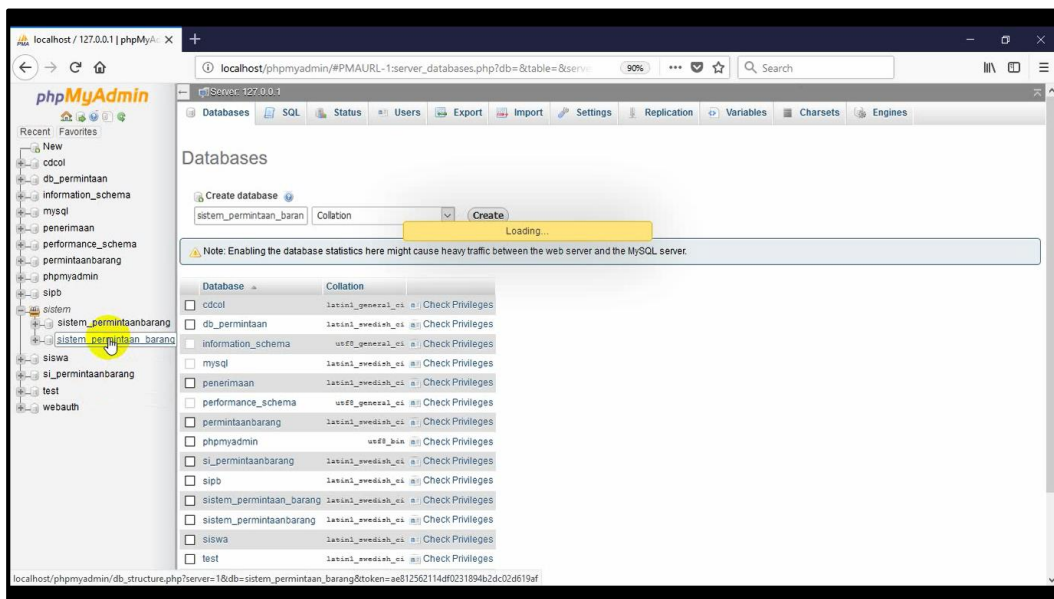
Klik close



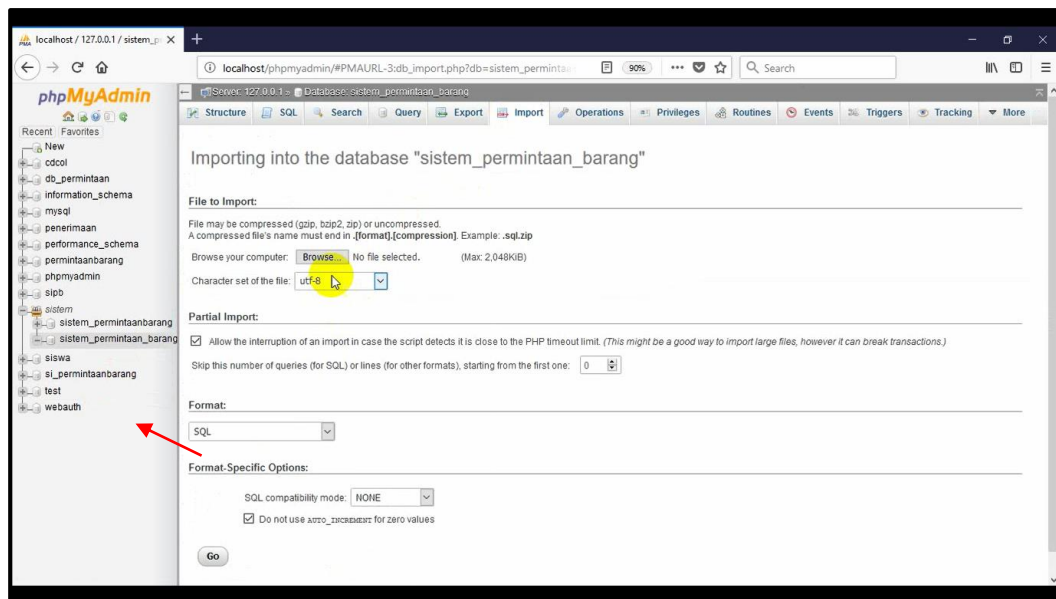
4. Buka “localhost/phpmyadmin/” untuk mengimport data yang sudah berekstensi “.sql”



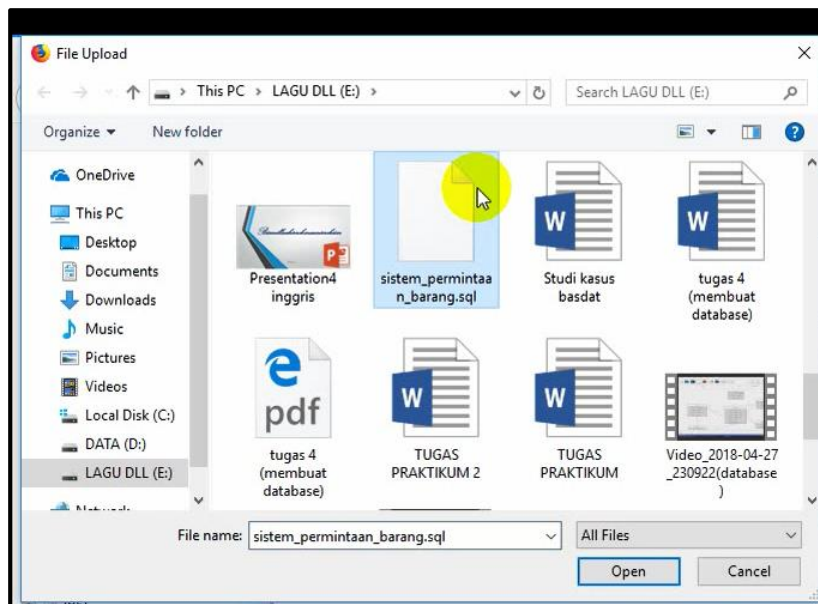
5. Buat database yang sama judulnya dengan file dari hasil generate database yang berekstensi “.sql”



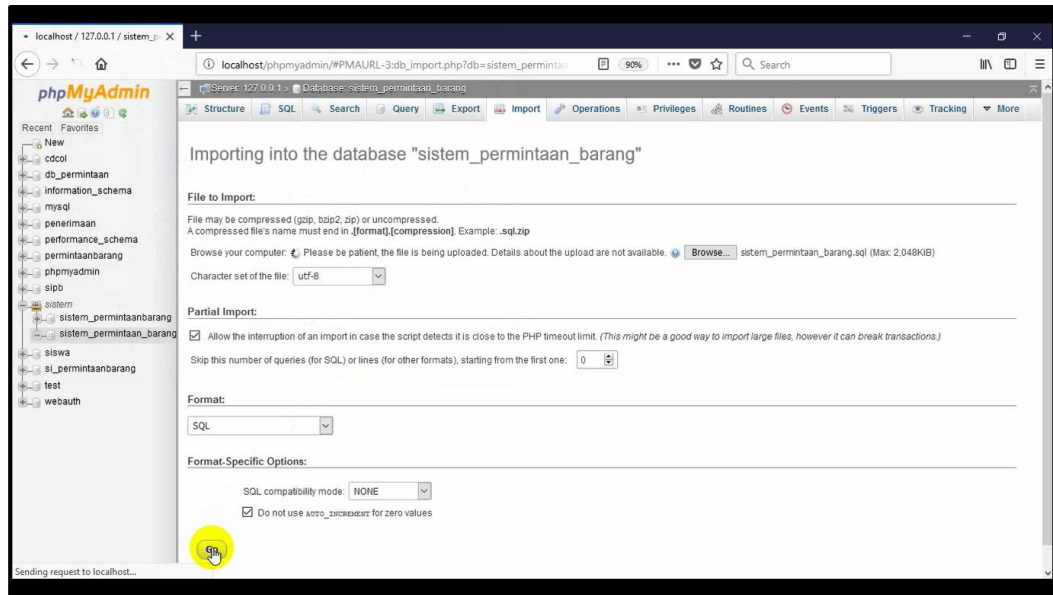
6. Klik database yang sudah dibuat, kemudian pilih “import”. Klik browse untuk mencari file yang ingin di import.



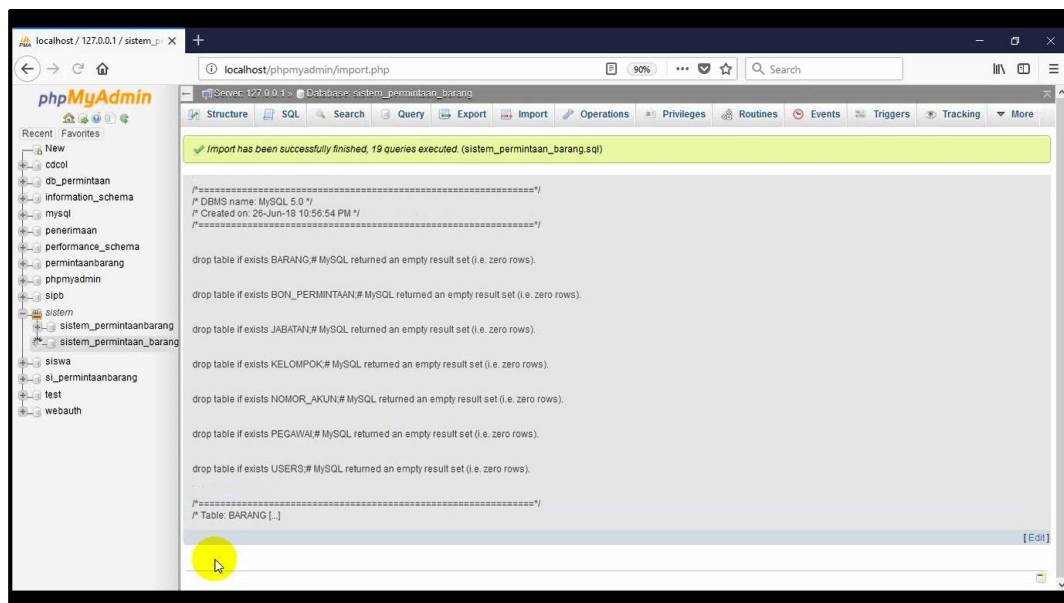
7. Pilih file yang sudah berekstensi “.sql” sebelumnya dengan klik “browse”



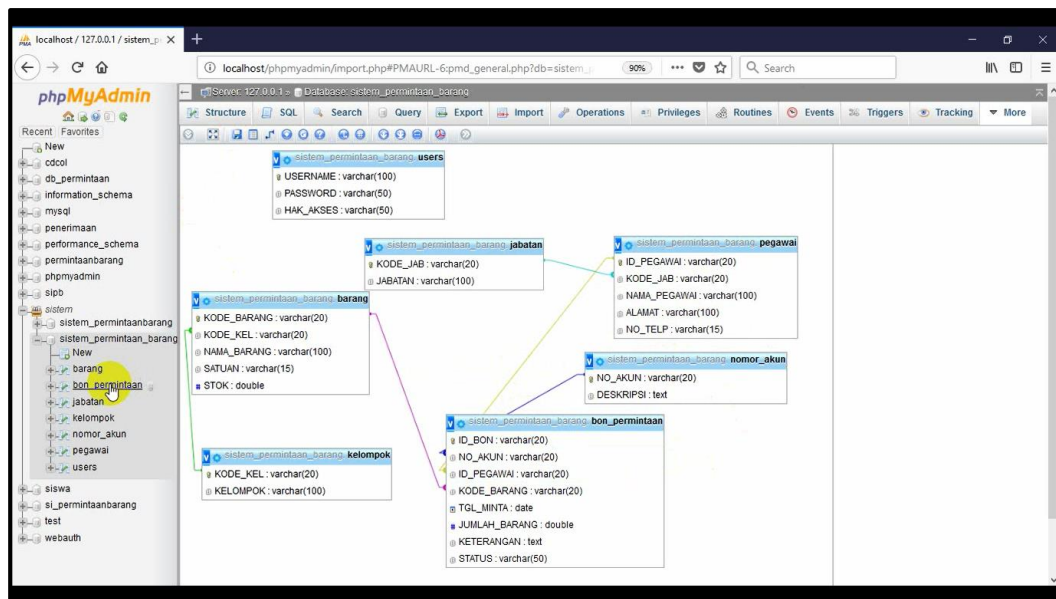
8. Klik Go



9. Database sudah berhasil di import.



10. Cek pada database yang di import di designer, dan database sudah langsung otomatis terelasi.

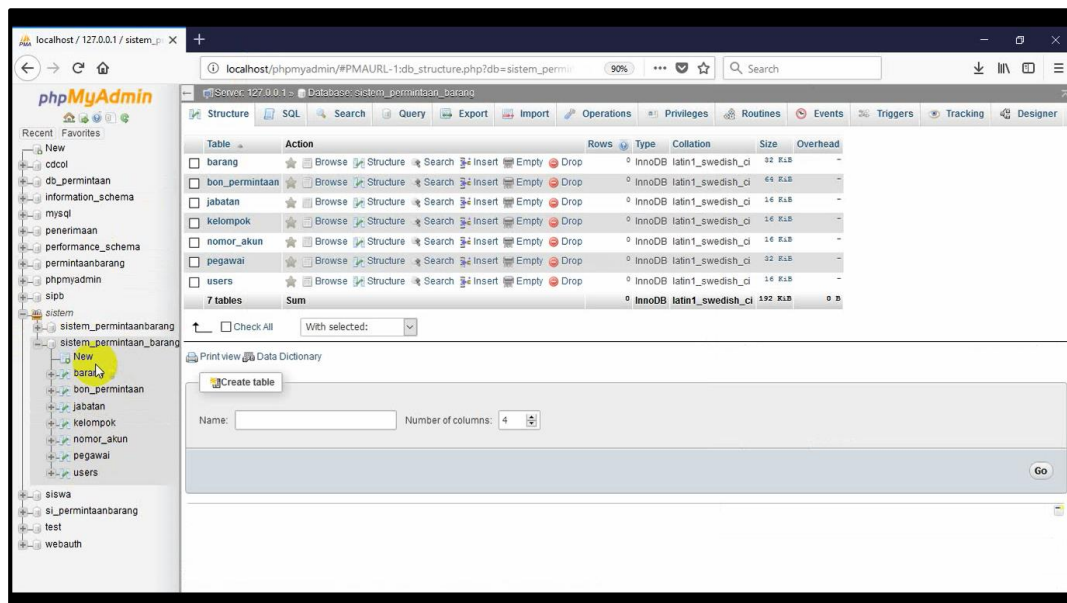


4.4 Import dan export csv data

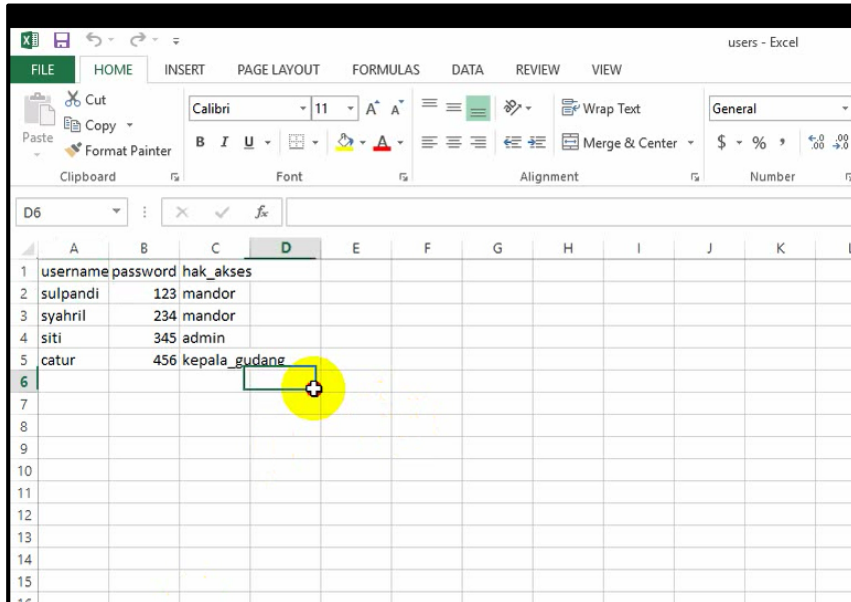
4.4.1 Import data CSV

Berikut langkah-langkah dalam mengimport data manual database dari CSV:

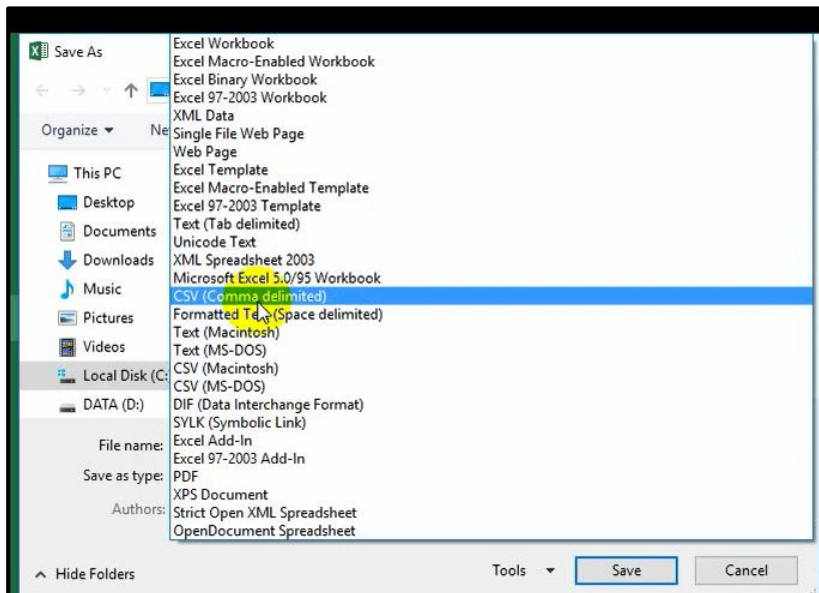
1. Buka database yang sudah dibuat sebelumnya, pada phpmyAdmin.



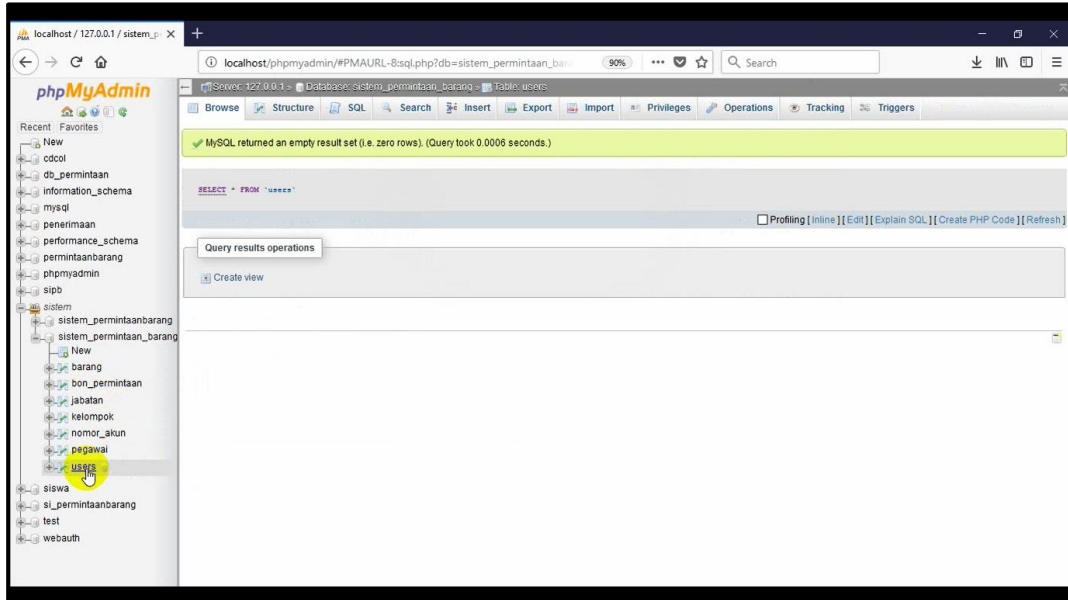
- Buat file excel yang isinya sama dengan tabel pada database yang kita buat. Misal untuk tabel barang yang berisi kode barang, nama barang dan satuan, maka inputkan data-data barang tersebut dalam file excel seperti gambar berikut.



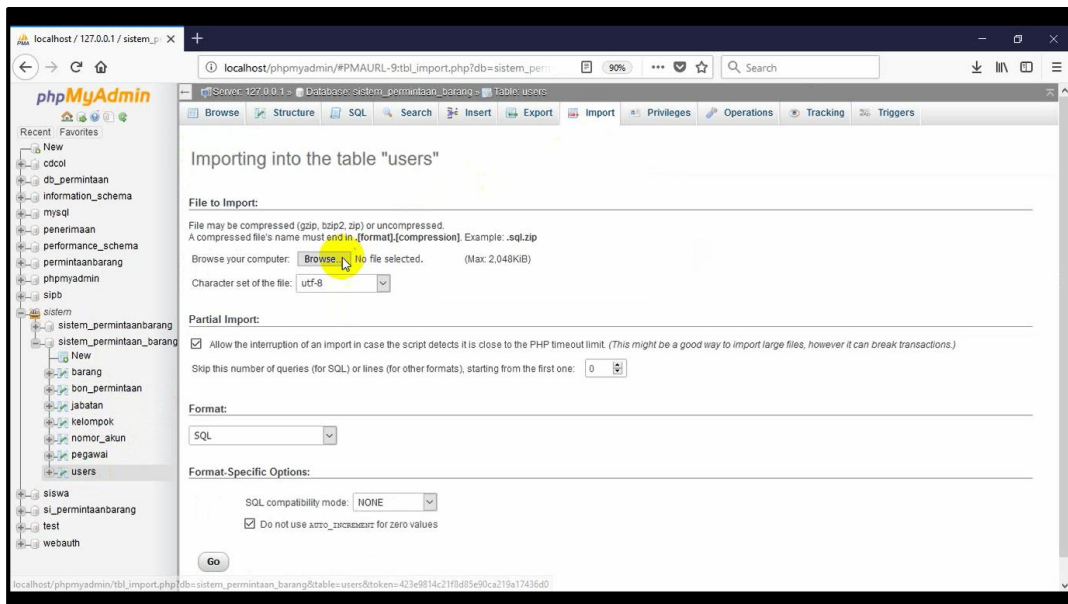
- Simpan file excel dengan nama file berekstensi “.csv”. Kemudian pilih type “CSV Comma Delimited”.



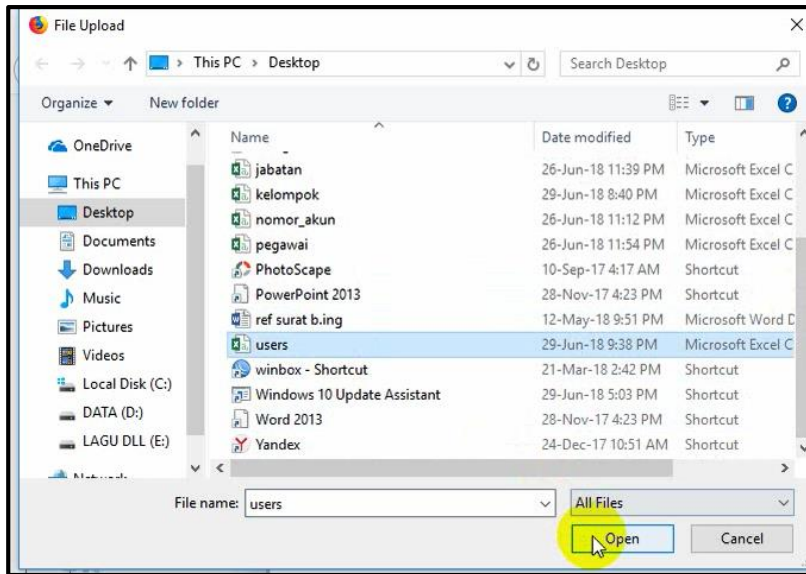
4. Buka tabel users pada database yang ingin diimport. Dapat dilihat pada gambar bahwa pada tabel users masih kosong belum ada data yang diinputkan. Untuk mengimport data, klik **import**.



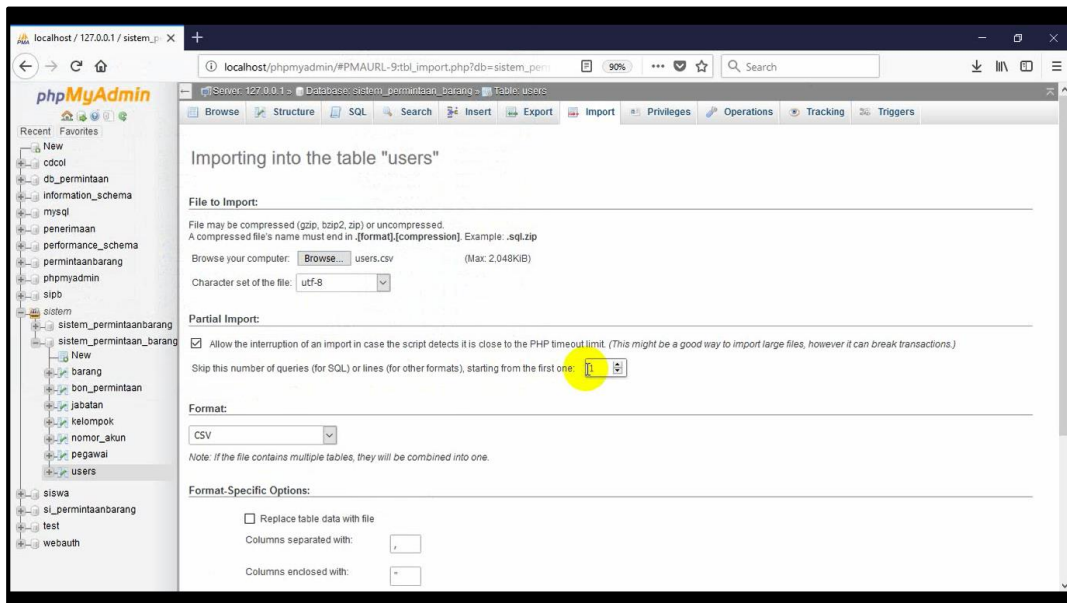
5. Pilih file yang ingin diimport tadi. Klik “browse”, untuk mencari file yang ingin diimport.



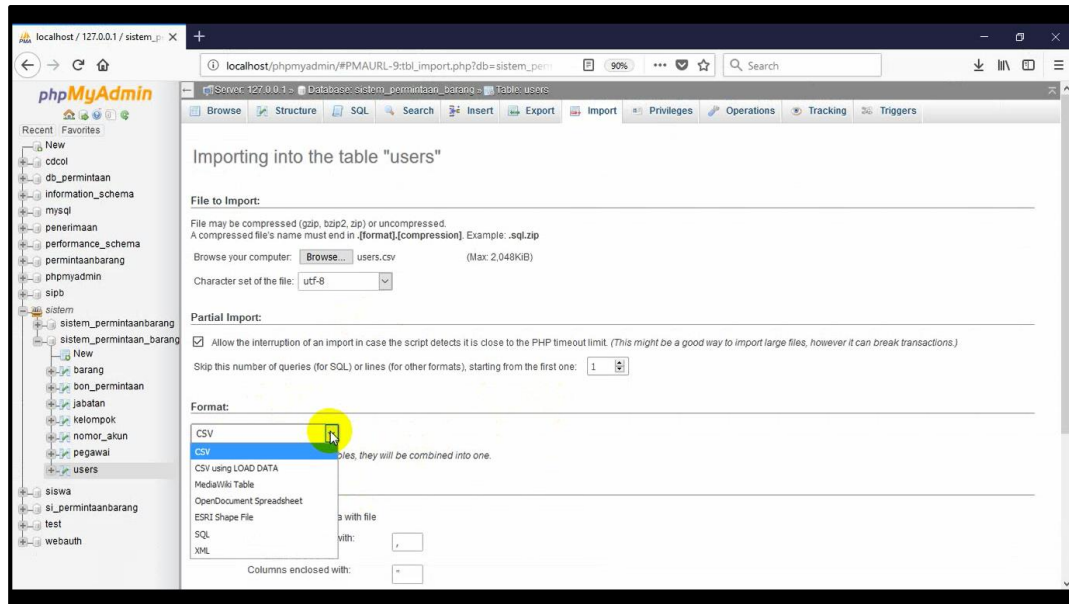
6. Cari file user, klik open.



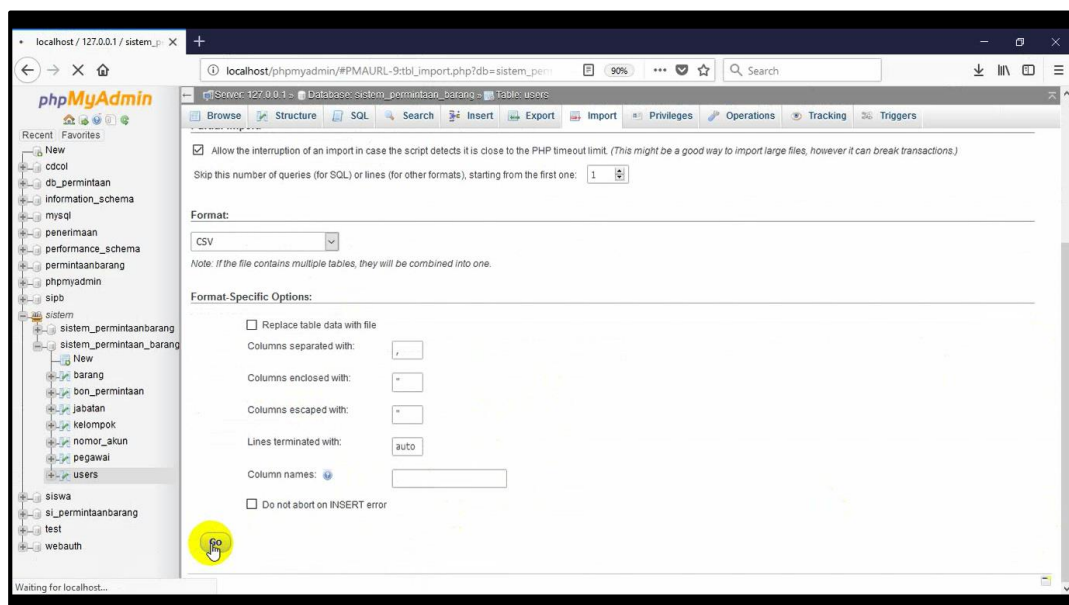
7. Ketik "1" pada form "starting from the first one", untuk melewati satu kolom dari format excel.



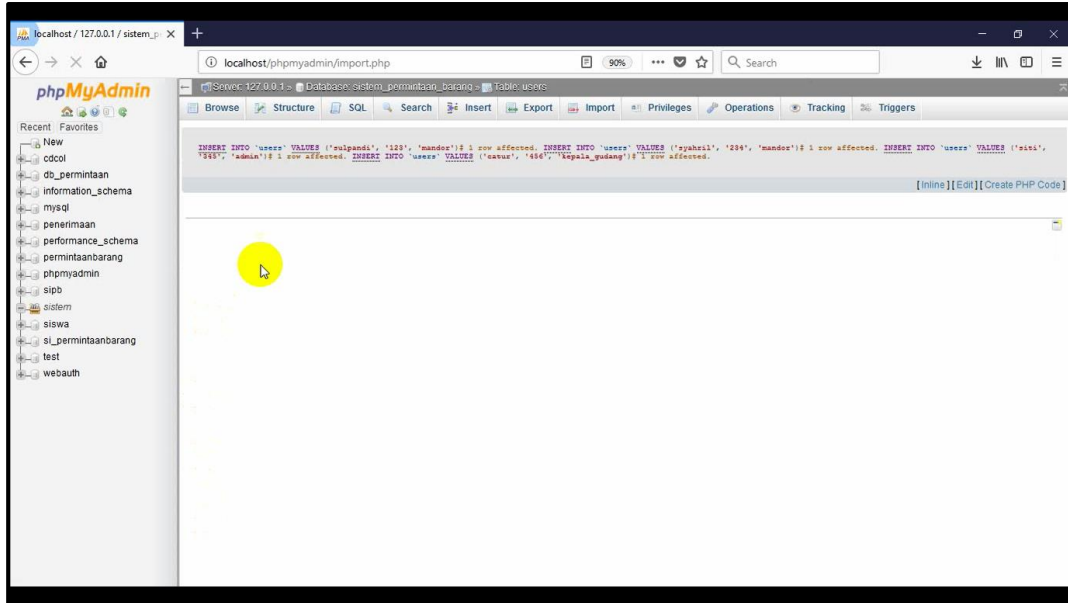
8. Pilih format "CSV" untuk mengimport dari CSV ke database.



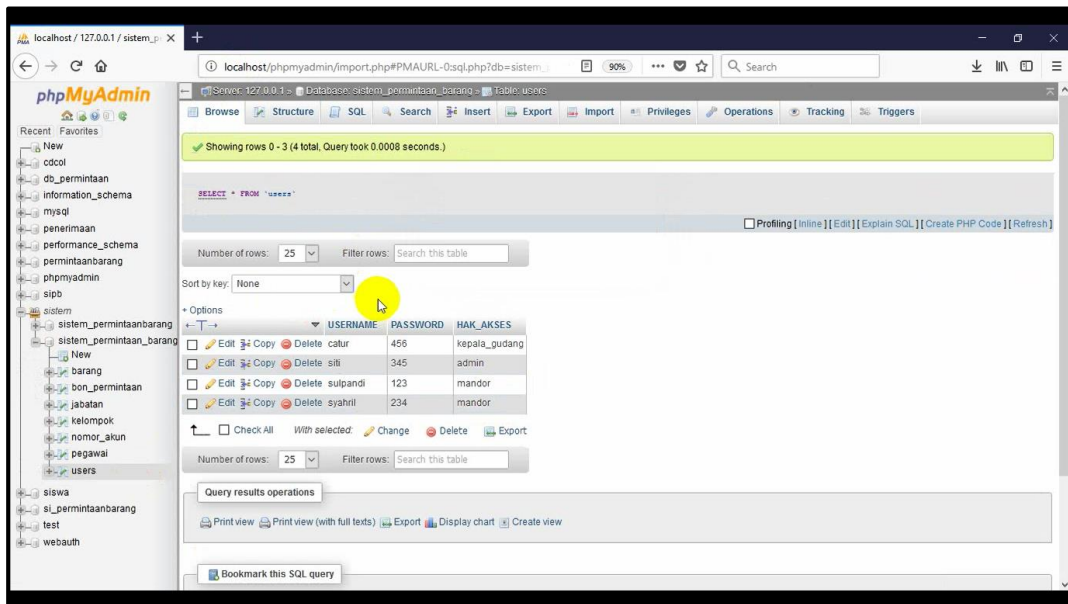
9. Klik GO



10. Ini menandakan bahwa proses import sudah berhasil.



11. Dapat dilihat hasilnya dengan klik “browse”.

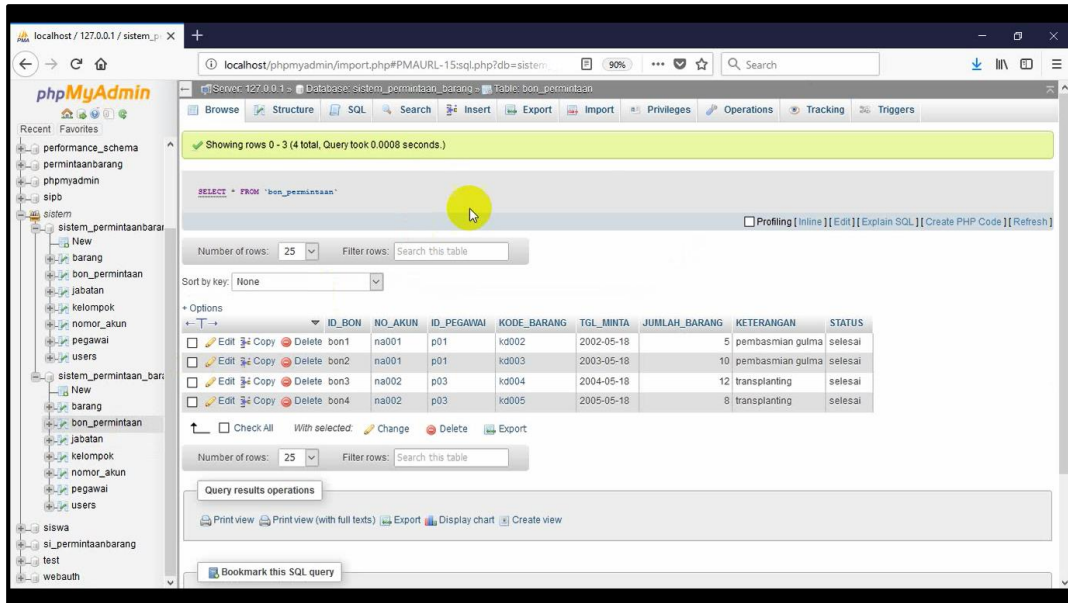


4.4.2 Export database ke CSV

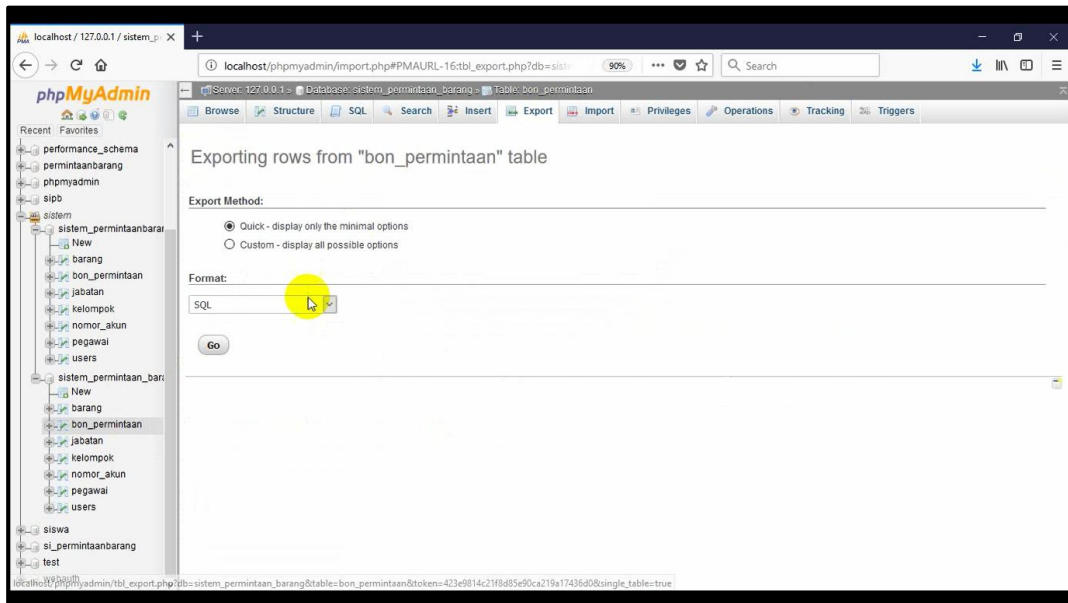
Database pada PHPMyAdmin ini juga dapat di export ke berbagai format.

Contohnya export ke format CSV.

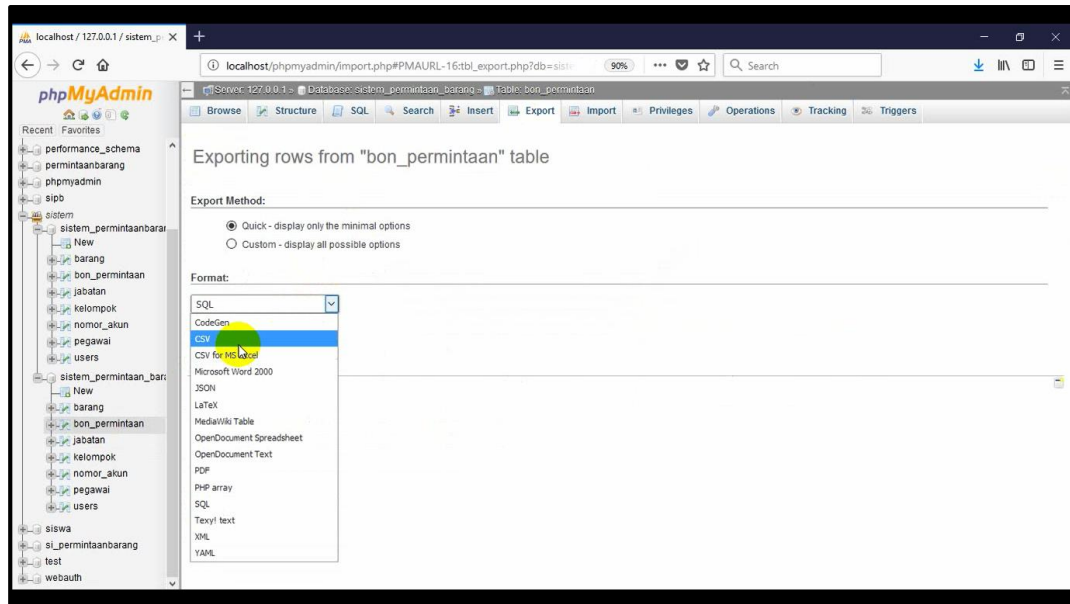
1. Buka table yang ingin di export



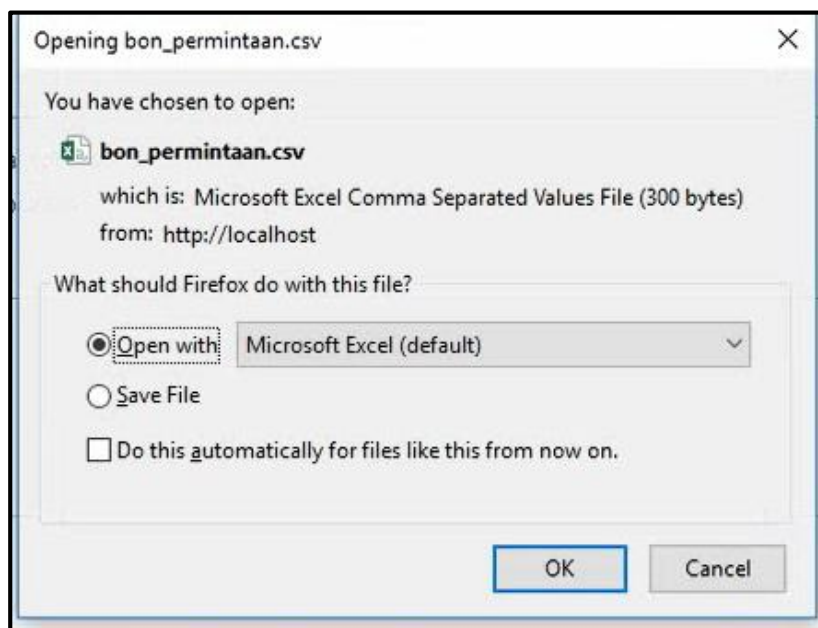
2. Pilih export



3. Pilih format, lalu GO



4. Akan muncul jendela berikut, klik OK



5. Akan tampil hasil dari export yang berupa file excel

	A	B	C	D	E	F	G	H	I	J
1	bon1	na001	p01	kd002	18-05-02	5	pembasm selesai			
2	bon2	na001	p01	kd003	18-05-03	10	pembasm selesai			
3	bon3	na002	p03	kd004	18-05-04	12	transplant selesai			
4	bon4	na002	p03	kd005	18-05-05	8	transplant selesai			
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										