DATABASE SQL

Pesantren PeTIK II YBM PLN
Jl. KH. Bisri Syansuri RT/01 RW/05, Plosogeneng,
Kec. Jombang, Kabupaten Jombang, Jawa Timur

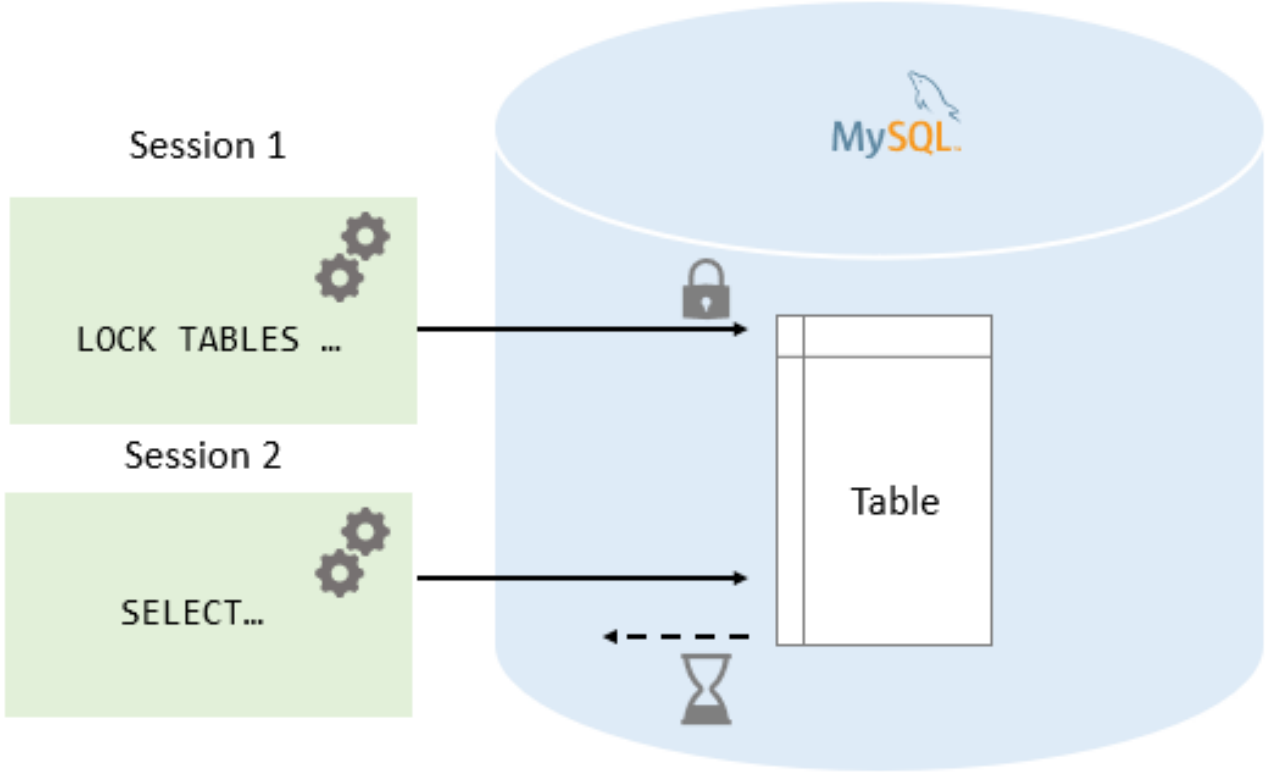
Pertemuan Ke-15

Materi

1. Pengantar Database
2. Pemodelan Data
3. Model Relasional Database
4. Normalisasi Database
5. Pengantar SQL
6. Perintah SQL SELECT 1
7. Perintah SQL SELECT 2
9. Fungsi Aggregate dan Grouping Data
10. Sub Query & SQL Join Table
11. View dan Analisa Query
12. Store Procedure dan Function
13. Trigger dan Transaction
14. Manajemen User
15. **Backup dan Restore**

15. Backup dan Restore

Lock Table



Lock Table adalah mekanisme mengkunci table, yang dapat digunakan membatasi sesi klien atau klien lain mengakses table selama periode tertentu.

https://www.mysqltutorial.org/mysql-table-locking/

LOCK TABLES (a)

- Berkenaan dengan penguncian sebuah table
- Contoh Kasus :

Sebuah database dapat digunakan oleh beberapa aplikasi, misalnya untuk internal aplikasi desktop sedangkan untuk eksternal berbasis web. Terkadang kedua aplikasi mengakses tabel secara bersamaan maka akan mungkin terjadi kesalahan data, contohnya klasik adalah tentang stok barang, aplikasi desktop sedang melakukan penambahan stok pada tabel produk, dan aplikasi web sedang melakukan penghitungan stok ketika proses penambahan belum selesai

LOCK TABLES (b)

- Pada kasus stok barang (produk) ini bisa dihindari dengan menerapkan LOCK TABLES
- LOCK TABLES melakukan penguncian tabel ketika proses sedang berlangsung, kemudian akan membebaskan table (UNLOCK TABLES) ketika proses telah berakhir
- LOCK TABLES telah disupport oleh MySQL versi 5
- Ada dua jenis LOCK TABLES

1. READ LOCK TABLES
2. WRITE LOCK TABLES

READ LOCK TABLES

- Penguncian table sehingga table hanya bisa dilakukan proses baca saja yang diterapkan pada semua session
- Jika terjadi manipulasi tabel seperti INSERT. UPDATE dan DELETE, truncate status nya akan selalu waiting hingga proses UNLOCK TABLES dilakukan
- Contoh :

```
mysql> LOCK TABLE produk READ;
   mysql> update produk set stok=stok + 10 where  id=1;
ERROR 1099 (HY000): Table 'produk' was locked with a READ lock and
    can't be updated
```

Contoh READ LOCK TABLES

- Contoh LOCK table produk dalam session yang sama :

```
mysql> LOCK TABLE produk READ;
```

```
   mysql> update produk set stok=stok + 10 where  id=1;
ERROR 1099 (HY000): Table 'produk' was locked with a READ lock and can't
    be updated
```

- Jika dilakukan pada session lain maka perintah akan dieksekusi ketika sudah dijalankan UNLOCK TABLES;

```
mysql> update produk set stok=stok + 10 where id=1;
```

Proses diatas masih menunggu proses UNLOCK TABLES

WRITE LOCK TABLES

- Penguncian table sehingga table hanya bisa dilakukan proses SELECT,INSERT,UPDATE,DELETE pada session sendiri saja
- Jika terjadi manipulasi tabel seperti INSERT. UPDATE dan DELETE, truncate pada session lain status nya akan selalu waiting hingga proses UNLOCK TABLES dilakukan
- Session lain hanya bisa melakukan proses read atau SELECT saja

Contoh WRITE LOCK TABLES

- Contoh LOCK table produk dalam session yang sama :

```
mysql> LOCK TABLE produk WRITE;
mysql> update produk set stok=stok + 10 where  id=1;
```

Query update sukses dijalankan

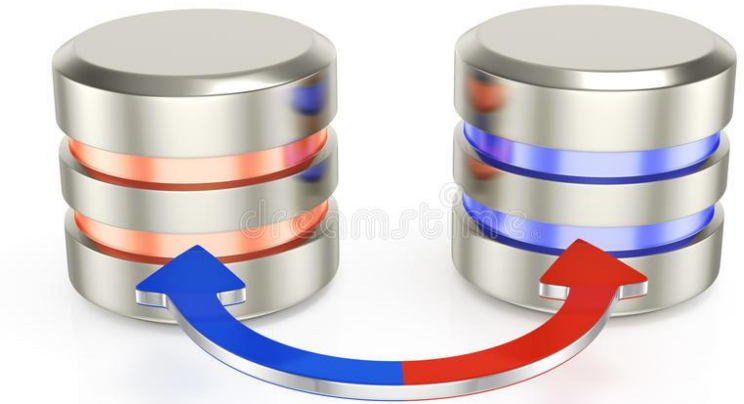
- Jika dilakukan pada session lain maka perintah akan dieksekusi ketika sudah dijalankan UNLOCK TABLES;

```
mysql> update produk set stok=stok + 10 where id=1;
```

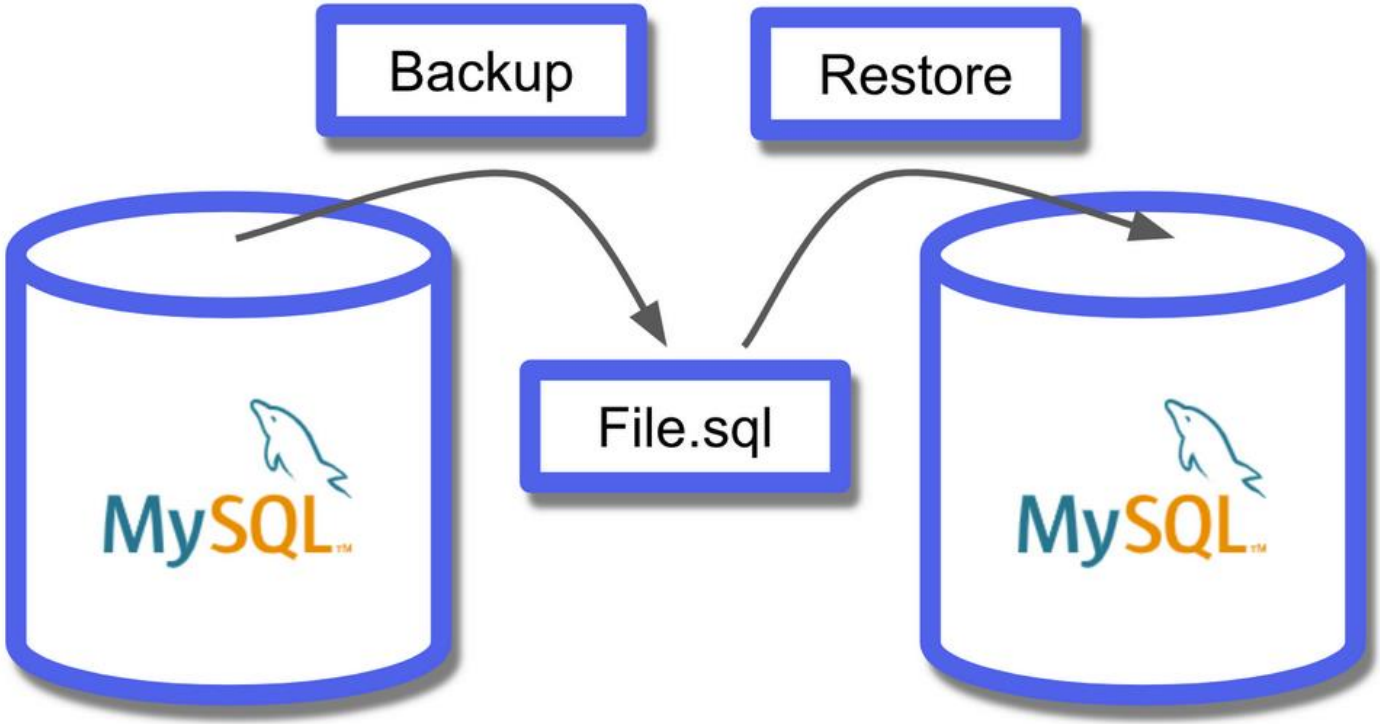
Proses diatas masih menunggu proses UNLOCK TABLES

Backup & Restore

- Backup dan Restore database adalah praktik untuk membuat salinan berkala data pada sistem database ke perangkat sekunder yang terpisah dan kemudian menggunakan salinan tersebut untuk memulihkan data ke sebuah sistem database lain.



Backup dan Restore



https://vocon-it.com/2018/11/27/mysql-backup-restore-on-openshift/

Backup MySQL :: mysqldump (1)

- Perintah **mysqldump** untuk backup database ke format file SQL

By default, **mysqldump** writes information as SQL statements to the standard output. You can save the output in a file:

```
shell> mysqldump [arguments] > file_name
```

To dump all databases, invoke **mysqldump** with the `--all-databases` option:

```
shell> mysqldump --all-databases > dump.sql
```

To dump only specific databases, name them on the command line and use the `--databases` option:

```
shell> mysqldump --databases db1 db2 db3 > dump.sql
```

The `--databases` option causes all names on the command line to be treated as database names. Without this option, **mysqldump** treats the first name as a database name and those following as table names.

https://dev.mysql.com/doc

Backup MySQL :: mysqldump (2)

- Perintah **mysqldump** untuk backup database ke format file

To dump a single database, name it on the command line:

```
shell> mysqldump --databases test > dump.sql
```

In the single-database case, it is permissible to omit the `--databases` option:

```
shell> mysqldump test > dump.sql
```

To dump only specific tables from a database, name them on the command line following the database name:

```
shell> mysqldump test t1 t3 t7 > dump.sql
```

https://dev.mysql.com/doc

Latihan Backup Database

- Backup database dbpos

```
#mysqldump –u root –p dbpos > D:\dbpos_20210326.sql
```

- Backup database dbpos hanya skema table saja tanpa data

```
#mysqldump –u root –p --no-data dbpos > D:\dbpos_20210326.sql
```

- Backup database dbpos hanya data saja

```
#mysqldump –u root –p --no-create-info dbpos > D:\dbpos_20210326.sql
```

- Backup tabel jenis_produk, produk pada database dbpos

```
#mysqldump –u root –p dbpos jenis_produk produk > D:\dbpos.sql
```

Restore MySQL :: Reload File SQL Backup

If the file is a single-database dump not containing `CREATE DATABASE` and `USE` statements, create the database first (if necessary):

```
shell> mysqladmin create db1
```

Then specify the database name when you load the dump file:

```
shell> mysql db1 < dump.sql
```

Alternatively, from within **mysql**, create the database, select it as the default database, and load the dump file:

```
mysql> CREATE DATABASE IF NOT EXISTS db1;
mysql> USE db1;
mysql> source dump.sql
```

https://dev.mysql.com/doc

Latihan Restore Database

- Buat database baru dbpos2 sebagai backup database dbpos

```
#mysqladmin create dbpos2 –u root –p
```

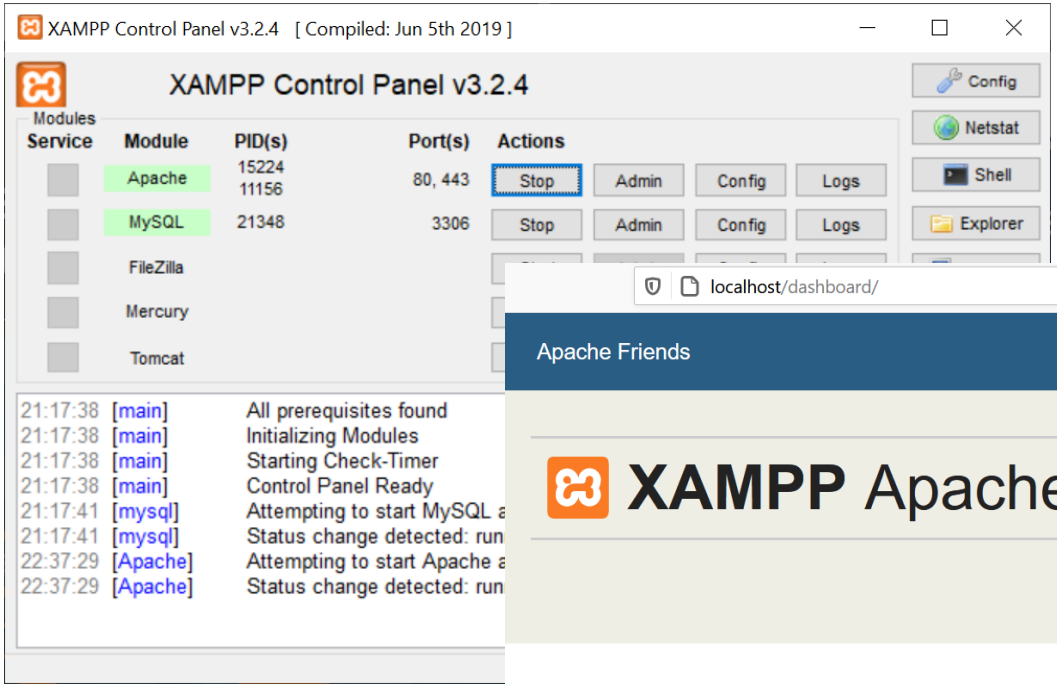
- Restore file hasil backup ke dbpos2

```
#mysql –u root –p dbpos2 < D:\dbpos_20210326.sql
```

- Cara lain dalam prompt mysql

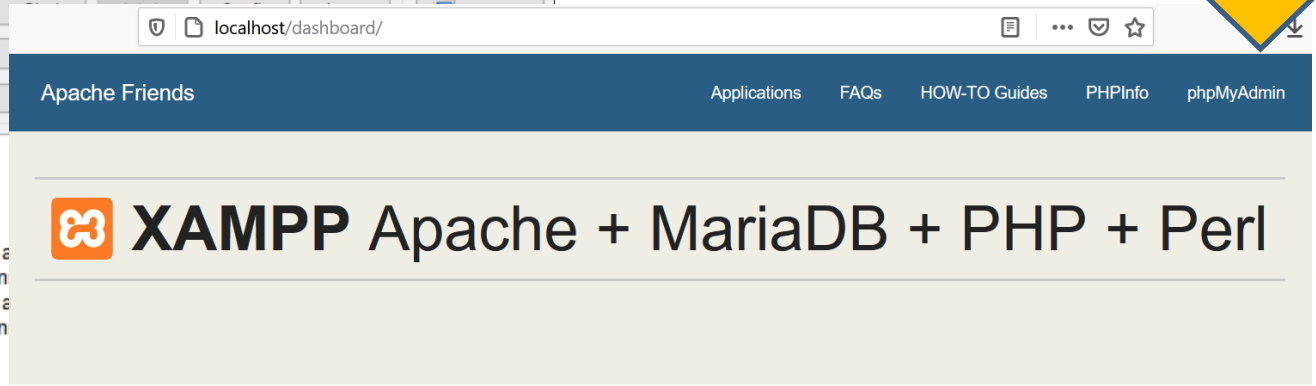
```
mysql> CREATE DATABASE IF NOT EXIST dbpos2;
mysql> USE dbpos2
mysql> source dbpos_20210326.sql
```

Backup & Restore :: phpMyAdmin (1)



Buka browser (Firefox/Chrome),
Arahkan URL ke http://localhost

Click

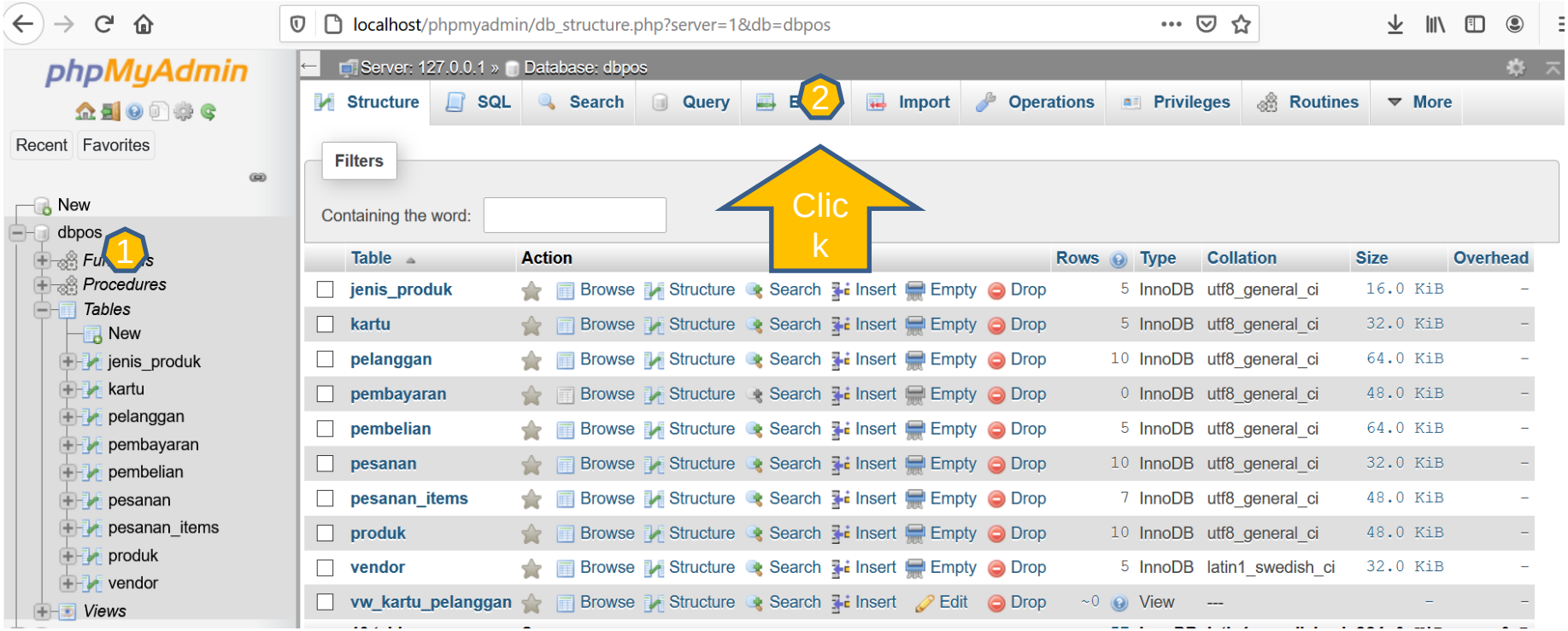


Welcome to XAMPP for Windows 7.4.15

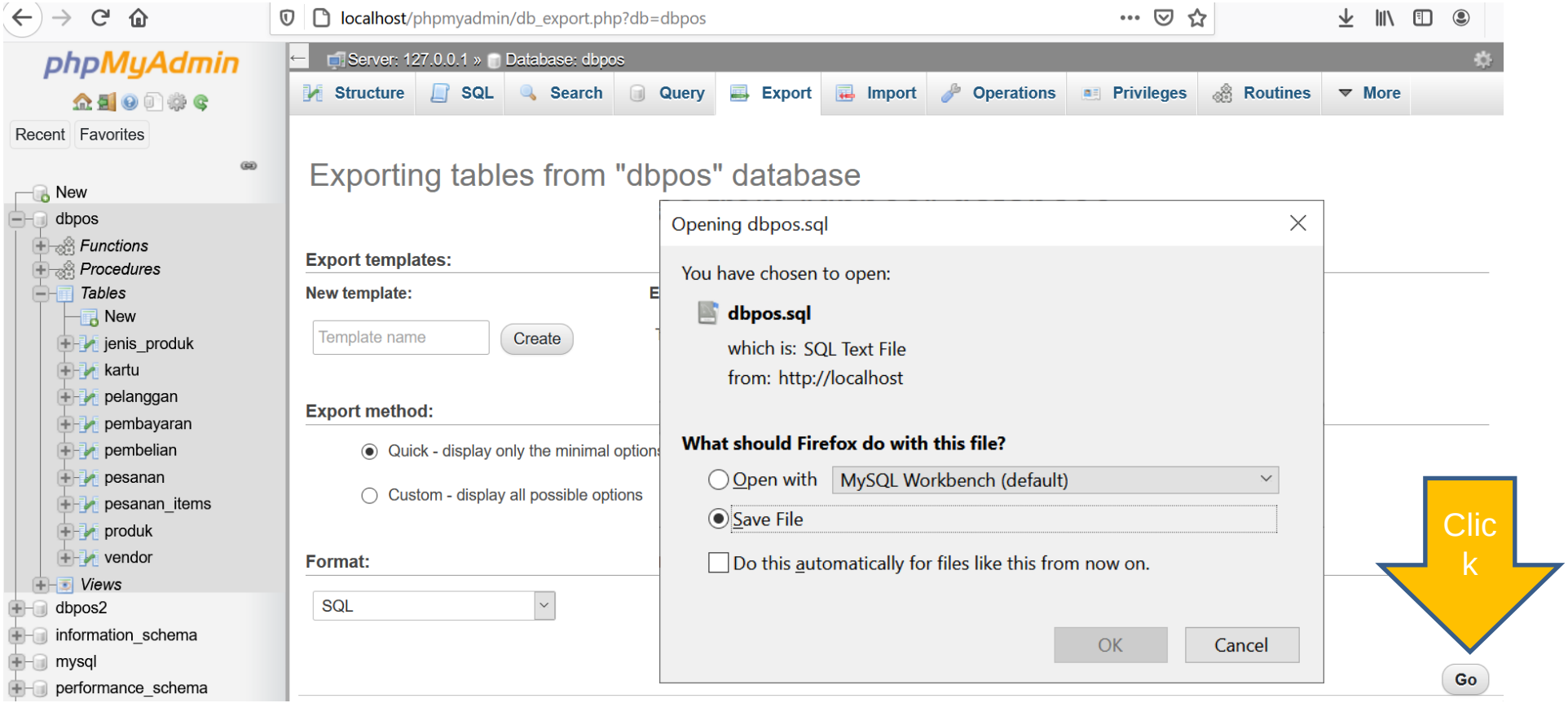
You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

Backup & Restore :: phpMyAdmin (2)

- 1) pilih database yg akan dibackup, 2) click menu export

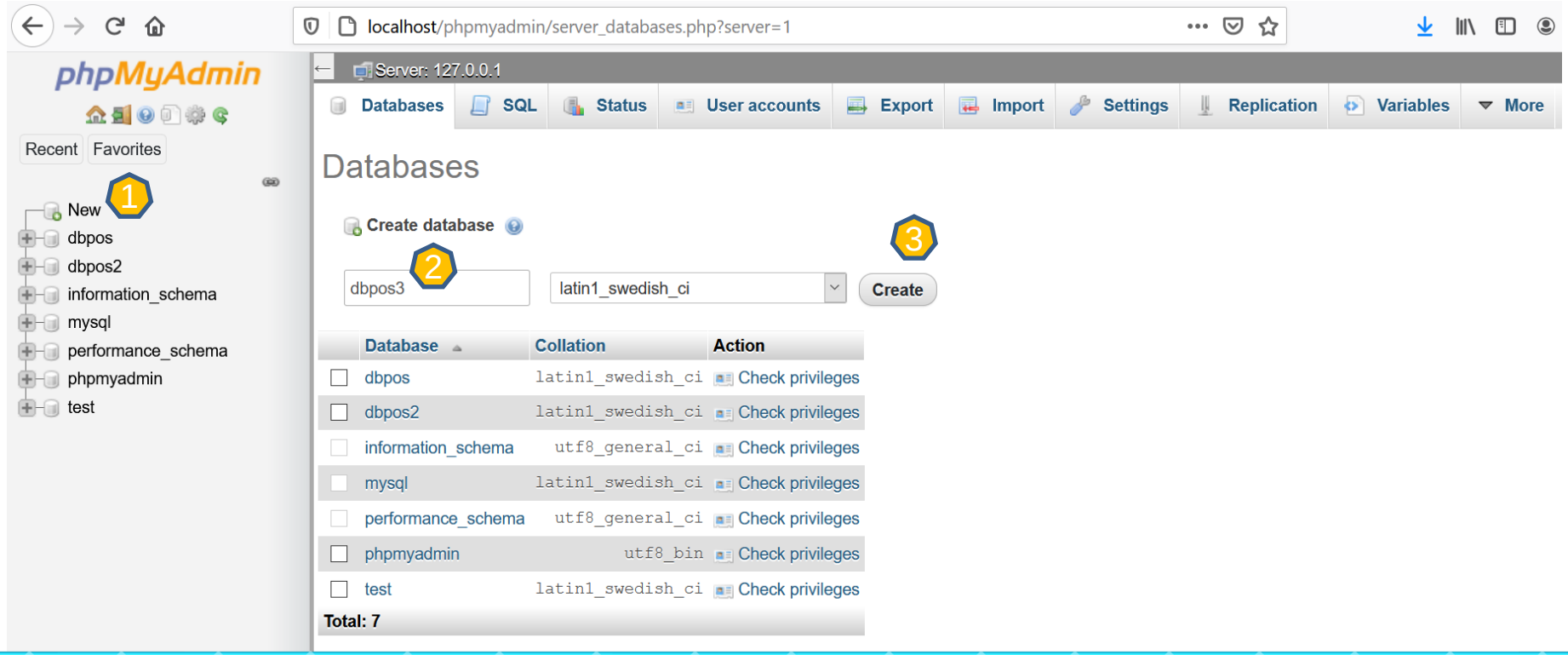


Backup & Restore :: phpMyAdmin (3)



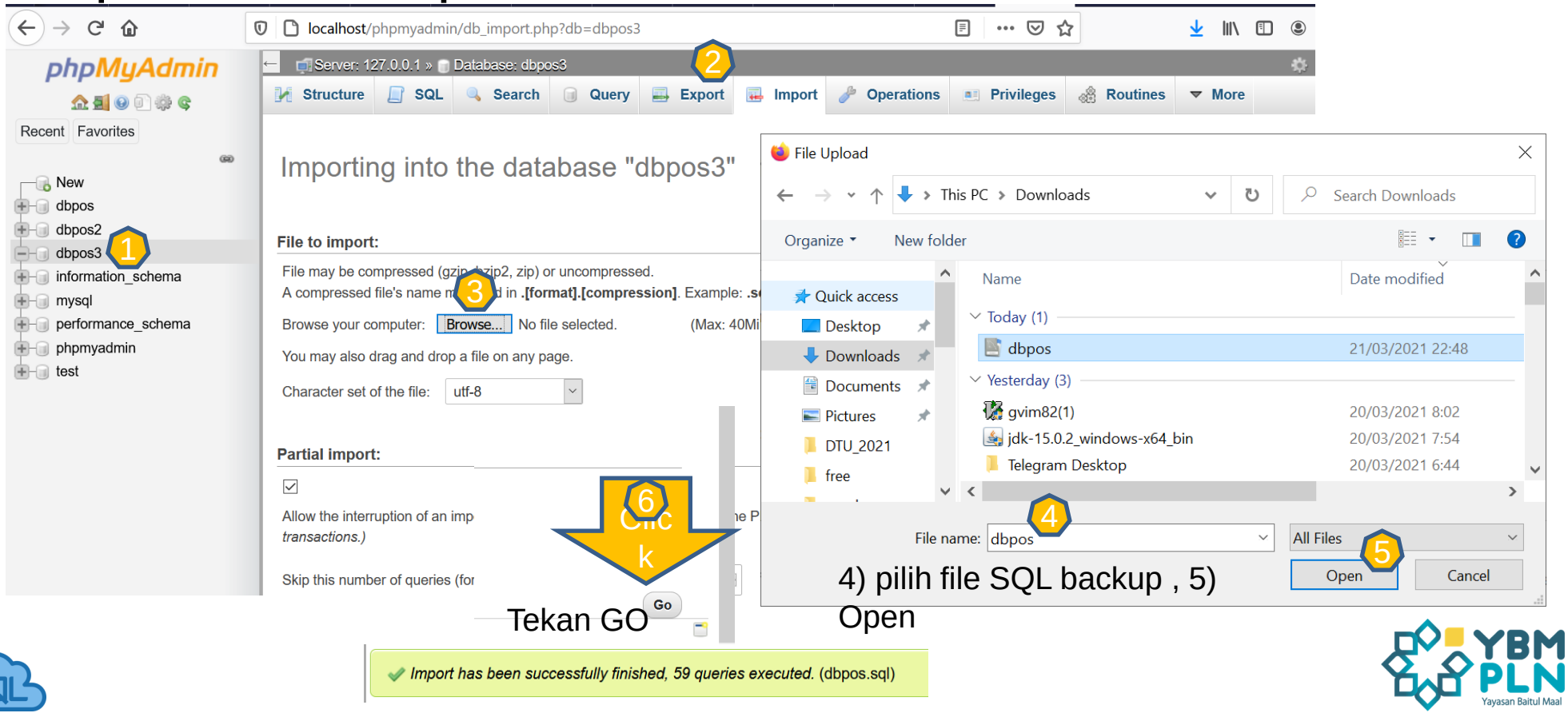
Backup & Restore :: phpMyAdmin (4)

- 1)buat database baru click new, 2) ketik nama databasenya, 3) create



Backup & Restore :: phpMyAdmin (5)

- 1) pilih database target restore 2) export, 3)browse file sql hasil backup



TERIMA KASIH
ATAS SEGALA PERHATIAN
SEMOGA BERMANFAAT...