

LANGKAH-LANGKAH BACKUP & RECOVERY PADA DATABASE MYSQL

Bella Arif Primadana

H1L011029

Program Studi Teknik Informatia

Univesitas Jenderal Soedirman
bellaarifprimadana@unsoed.ac.id

ABSTRAK

Data dan basis data merupakan komponen terpenting dalam suatu sistem informasi dan tentunya aplikasi untuk sistem informasi harus tersedia, keduanya harus saling tergantung.

Suatu aplikasi sistem informasi tidak ada gunanya jika tidak memiliki data yang lengkap, demikian juga sebaliknya jika memiliki data tetapi tidak mempunyai aplikasi yang digunakan

maka tidak akan menghasilkan suatu informasi, statistik, dsb. Karena pentingnya basis data maka diperlukan sesuatu yang dapat melindungi sebuah basis data dari kerusakan. Salah satu metode untuk melindungi database dari kerusakan adalah *backup* dan *recovery/restore*.

Backup data merupakan pengelolaan database untuk melakukan penyalinan sistem, data, dan aplikasi. Backup dilakukan untuk menjaga basis data dari kerusakan dari sistem dari luar ataupun dari dalam sistem, yang disengaja atau tidak disengaja. Sedangkan

recovery/restore merupakan proses pengembalian *backup* ke dalam sistem. *Restore* dilakukan untuk mengembalikan keadaan sistem kembali pada keadaan semula, keadaan terakhir pada saat operasional, sebelum terjadi kerusakan pada sistem. Pada proses ini

dilakukan pengembalian data baik struktur maupun isi dari database.

Key Word : Backup, Recovery, Database, MySQL.

PENDAHULUAN

Basis data telah menjadi bagian yang menyatu hampir setiap kehidupan manusia. Tanpa basis data, banyak sesuatu yang kita kerjakan akan menjadi sangat membosankan dan tidak terstruktur dengan

baik, bahkan menjadi sesuatu yang tidak dapat dikerjakan. Beberapa contoh organisasi yang sangat bergantung pada

sistem basis data adalah perpustakaan, universitas, kantor pemerintahan, bank, dsb. Di dunia internet, *search engine*, *online shop*, dan setiap *website* yang menyediakan banyak data tidak akan bekerja tanpa menggunakan basis data. Basis data yang

sudah diimplementasikan atau digunakan di komputer biasanya dihubungkan dengan *database server*. Salah satu *database server*

yang sering digunakan adalah MySQL, sebuah sever/klien database SQL yang berasal dari Skandinavia. MySQL terdiri atas server SQL, klien program untuk mengakses server, *tools* untuk administrasi, dan *interface* program untuk menulis program. Karena pentingnya sebuah basis data untuk menyimpan data-data sebuah organisasi maka diperlukan metode untuk melindungi basis data dari kerusakan. Metode yang digunakan adalah *backup* dan *recovery*. *Backup* dilakukan untuk melakukan penyalinan sistem, data, dan aplikasi yang nantinya bisa di-*recovery* untuk mengembalikan data yang ada pada sebuah basis data kembali seperti semua sebelum adanya kerusakan pada data-data di sebuah sistem.

LANDASAN TEORI

Backup dan *recovery* terjadi karena adanya kerusakan atau kegagalan operasi pada basis data, beberapa sebab kerusakan atau kegagalan operasi tersebut antara lain :

- a. Aliran listrik terputus, hal ini dapat mengakibatkan hilangnya informasi yang ada di memori utama dan register.
- b. Kesalahan operator (*human error*), manusia yang menjadi *database administrator* melakukan kesalahan operasi yang disengaja maupun yang tidak disengaja.
- c. Kesalahan perangkat lunak, hal ini dapat mengakibatkan hasil pengolahan tidak

benar, informasi yang disajikan salah, dan database menjadi tidak konsisten.

- d. *Disk* rusak, mengakibatkan hilangnya informasi atau rusaknya database yang ada di dalam *disk*.

Dari penyebab di atas dapat terjadi beberapa jenis kerusakan diantaranya adalah :

- a. Kegagalan transaksi (*transaksi failure*)
Kegagalan transaksi dapat terjadi karena kesalahan logika (*logical error*) dimana program tidak dapat melanjutkan eksekusi program kerana adanya kondisi internal tertentu seperti masukan yang salah/rusak, data yang tidak tersedia, nilai data di luar batas, logika program tidak tepat. Selain itu, dapat juga terjadi karena kesalahan sistem (*system error*) dimana sistem telah memasuki kondisi yang tidak diharapkan (*deadlock*) sebagai hasil dari tidak tereksekusinya program secara normal.
- b. Kerusakan sistem (*system crash*)
Salah satu penyebabnya adalah *hardware* macet (*hang*) yang mengakibatkan penyimpanan sementara hilang.
- c. Kegagalan/kerusakan disk (*disk failure*)
Terjadinya *bad sector* atau *disk* macet pada saat berlangsungnya operasi I/O ke *disk*.

Kerusakan terhadap *disk* (media penyimpanan), kerusakan data karena aktivitas pemakai ataupun kerusakan data oleh aplikasi eksternal dapat diantisipasi dengan melakukan operasi *backup* secara periodik. Berdasarkan waktu pelaksanaan atau strategi terdapat dua jenis operasi

backup yaitu :

- *Backup* statis, dimana *backup* dilakukan dengan lebih dulu menonaktifkan basis data secara keseluruhan.
- *Backup* dinamis, dimana *backup* dilakukan tanpa menonaktifkan basis data.

Terdapat tiga pilihan skema untuk menjalankan mekanisme *recovery* secara otomatis begitu kerusakan atau kegagalan sistem telah terjadi, skema tersebut adalah :

- File Log dengan Penundaan Pengubahan (*Incremental Log with Deferred Update*)
- File Log dengan Pengubahan Langsung (*Incremental Log with Immediate Updates*)
- Page Bayangan (*Shadow Paging*), yang memerlukan akses ke disk yang lebih sedikit.

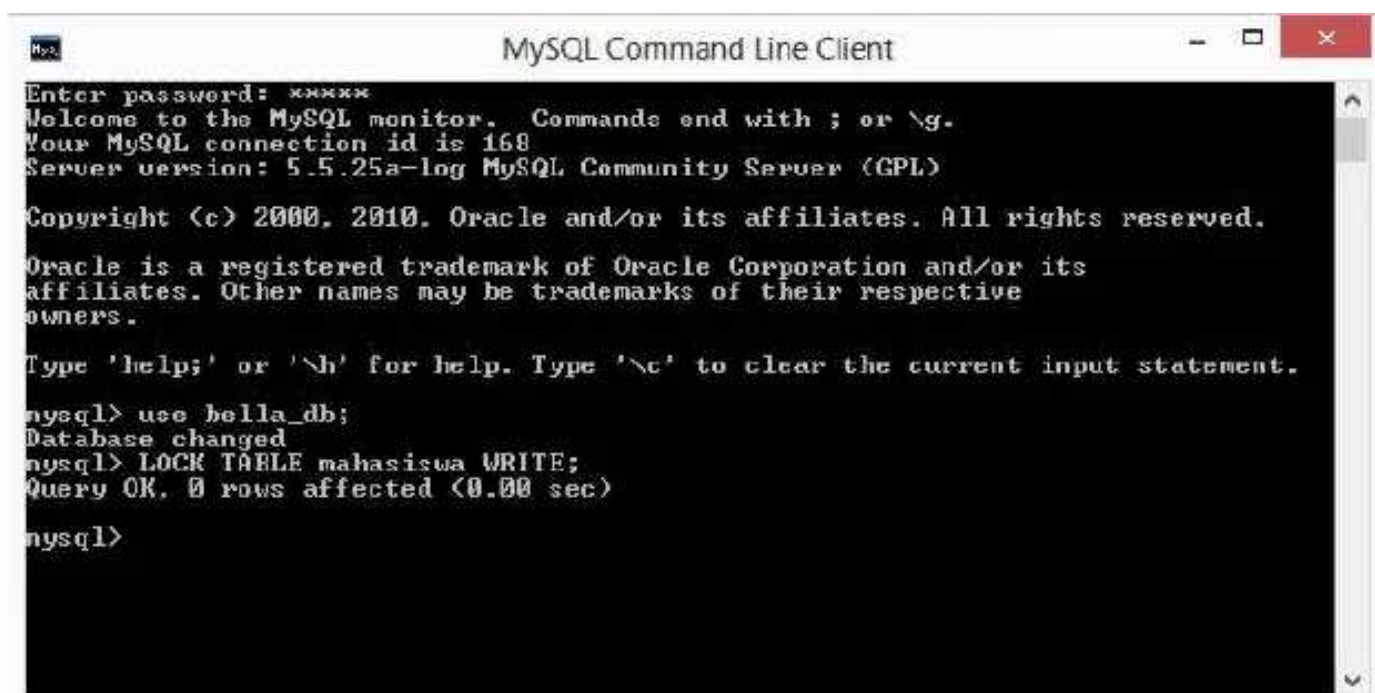
Proses *backup* dan *recovery* dapat dilakukan dengan menggunakan perintah SQL, perintah-perintah yang digunakan adalah `SELECT INTO OUTLIFE`, `BACKUP TABLE`, dan `LOAD DATA INFILE`. Sebelum proses *backup* dilakukan, harus dilakukan proses penguncian pada table untuk memastikan tidak ada proses penulisan atau perubahan data dalam tabel. Contoh berikut akan menampilkan proses backup dan recovery pada table 'mahasiswa' yang berada pada database 'bella_db' :

1. Lakukan proses penguncian table dengan perintah :

`LOCK TABLE nama_table WRITE;`

JENIS-JENIS *BACKUP* DAN *RECOVERY* PADA MYSQL

A. *Backup* dan *Recovery* Menggunakan Perintah SQL



```

MySQL Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 168
Server version: 5.5.25a-log MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use bella_db;
Database changed
mysql> LOCK TABLE mahasiswa WRITE;
Query OK, 0 rows affected (0.00 sec)

mysql>
  
```

2. Lakukan pengosongan memori atau FLUSH. Langkah ini diperlukan untuk memastikan tidak ada proses yang

berlangsung terhadap data pada tabel 'mahasiswa'. Flush dilakukan dengan perintah :
FLUSH TABLES;

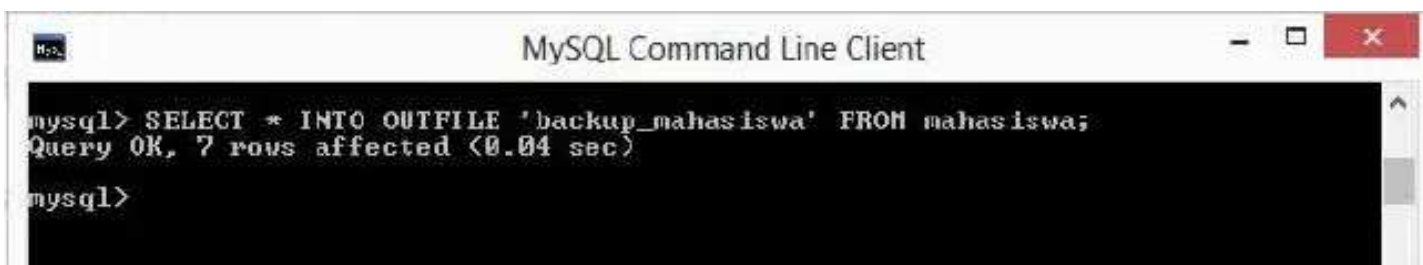


```
mysql> FLUSH TABLES;
Query OK, 0 rows affected (0.07 sec)

mysql>
```

3. Lakukan proses *backup* table 'mahasiswa' dengan perintah :

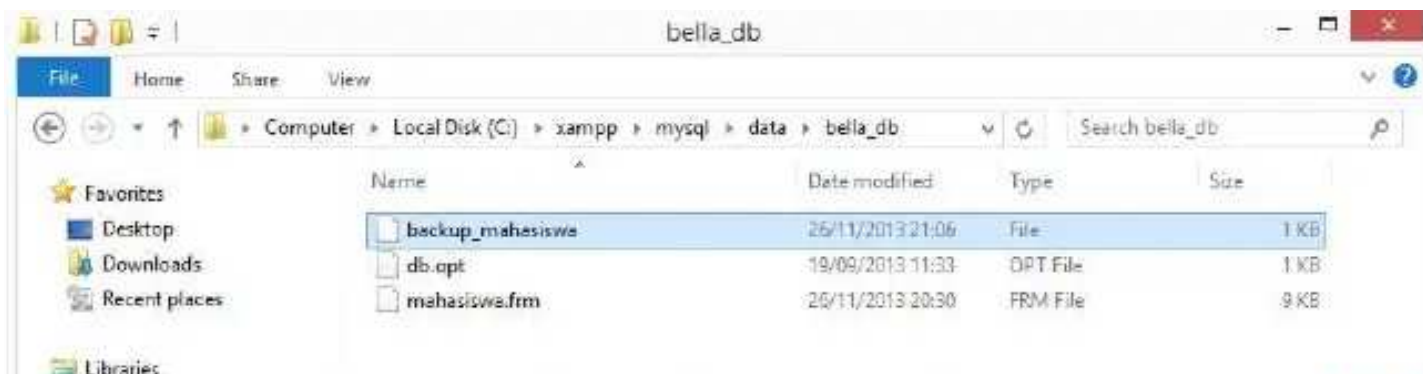
SELECT * INTO OUTFILE
'backup_mahasiswa' FROM
mahasiswa;



```
mysql> SELECT * INTO OUTFILE 'backup_mahasiswa' FROM mahasiswa;
Query OK, 7 rows affected (0.04 sec)

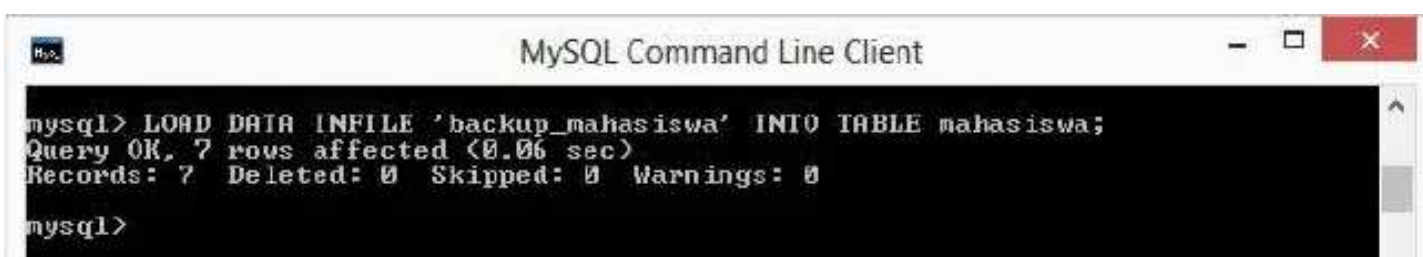
mysql>
```

Jika proses *backup* berhasil maka akan muncul file backup_mahasiswa pada direktori C:\xampp\mysql\data.



4. Data yang telah di-*backup* dapat dikembalikan kapan saja bila diperlukan. Sintaks SQL yang digunakan adalah LOAD DATA

INFILE. Perintah yang dijalankan adalah :
LOAD DATA INFILE
'backup_mahasiswa' INTO
TABLE mahasiswa;



```
mysql> LOAD DATA INFILE 'backup_mahasiswa' INTO TABLE mahasiswa;
Query OK, 7 rows affected (0.06 sec)
Records: 7 Deleted: 0 Skipped: 0 Warnings: 0

mysql>
```

B. Backup dan Recovery Menggunakan MySQLDump

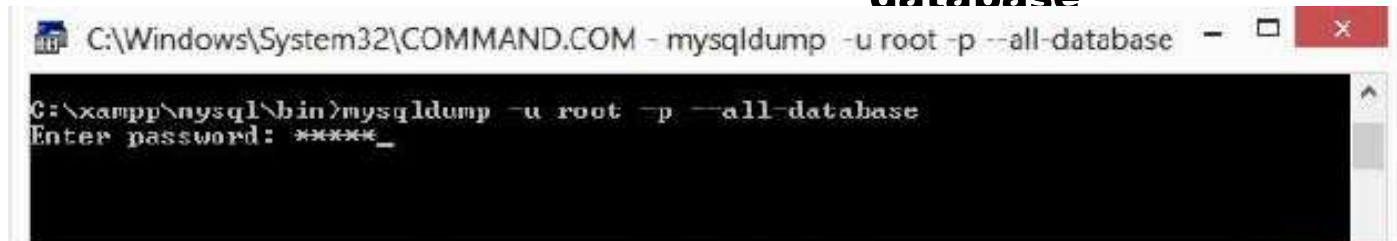
Tool MySQLdump memiliki kelebihan dalam proses *backup database*. Tools ini akan melakukan *backup database* beserta struktur table di dalamnya. Selain itu, file *backup* yang dihasilkan dapat digunakan bagi sistem *database* yang lain. Langkah-langkah penggunaan

MySQLdump dalam proses *backup* dan *recovery database* adalah sebagai berikut :

1. Jalankan *shell* atau *commad-prompt* dan ketikkan perintah berikut untuk memulai *dump database* :

MYSQLDUMP -u root -p -all-

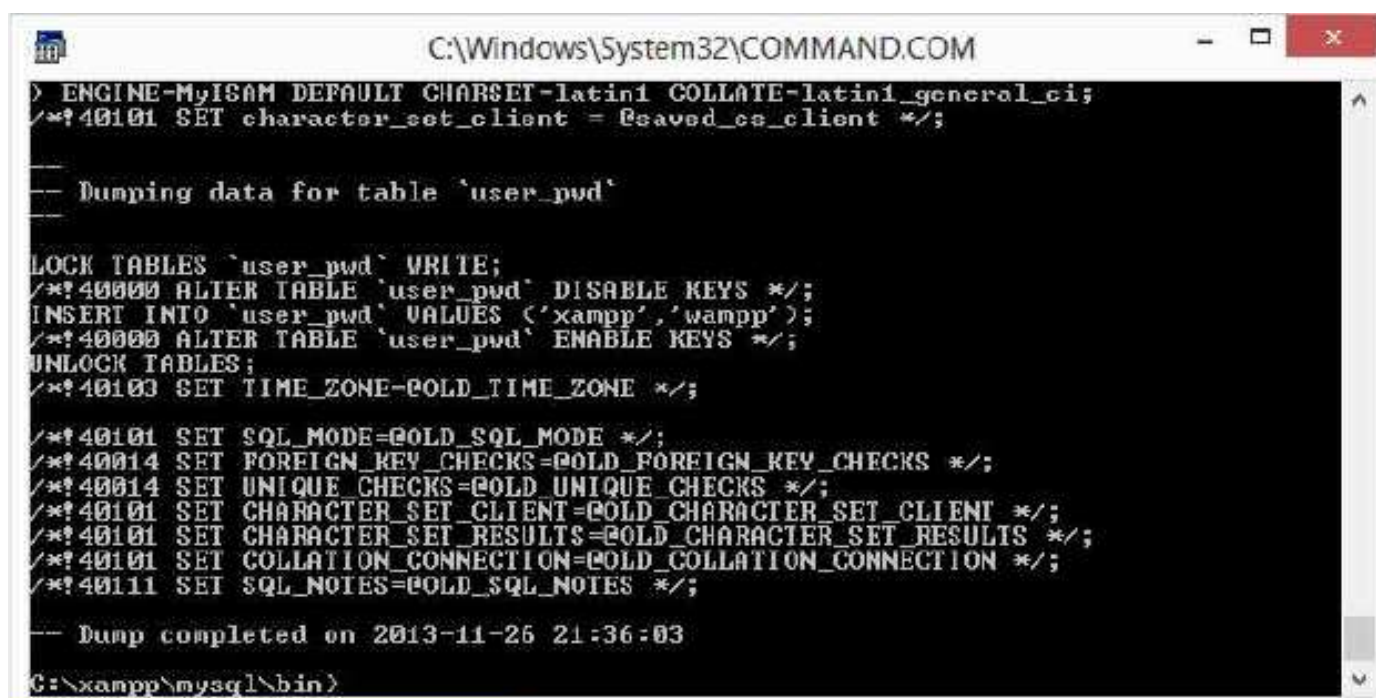
database



```
C:\Windows\System32\COMMAND.COM - mysql -u root -p --all-database
C:\xampp\mysql\bin>mysql -u root -p --all-database
Enter password: *****
```

Perintah di atas akan melakukan *backup* pada semua *database* yang ada pada MySQL.

2. Proses *backup* atau *dumping* akan berlangsung beberapa saat hingga muncul pesan "Dump completed".



```
C:\Windows\System32\COMMAND.COM
> ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Dumping data for table 'user_pwd'

LOCK TABLES `user_pwd` WRITE;
/*!40000 ALTER TABLE `user_pwd` DISABLE KEYS */;
INSERT INTO `user_pwd` VALUES ('xampp','wampp');
/*!40000 ALTER TABLE `user_pwd` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2013-11-26 21:36:03
C:\xampp\mysql\bin>
```

3. Data yang telah di-*backup* dapat di *restrore* kembali ke dalam *database* dengan perintah :

MySQLdump -u root -p

(nama_database) <

c:\file_backup.sql

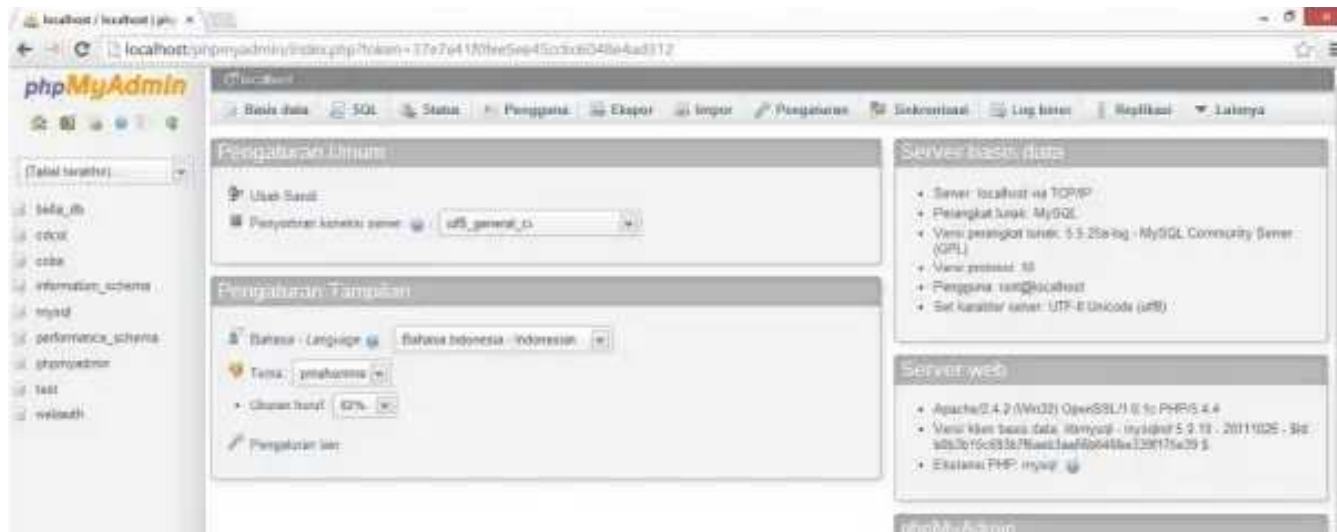
C. Backup dan Recovery Melalui

PHPMyAdmin

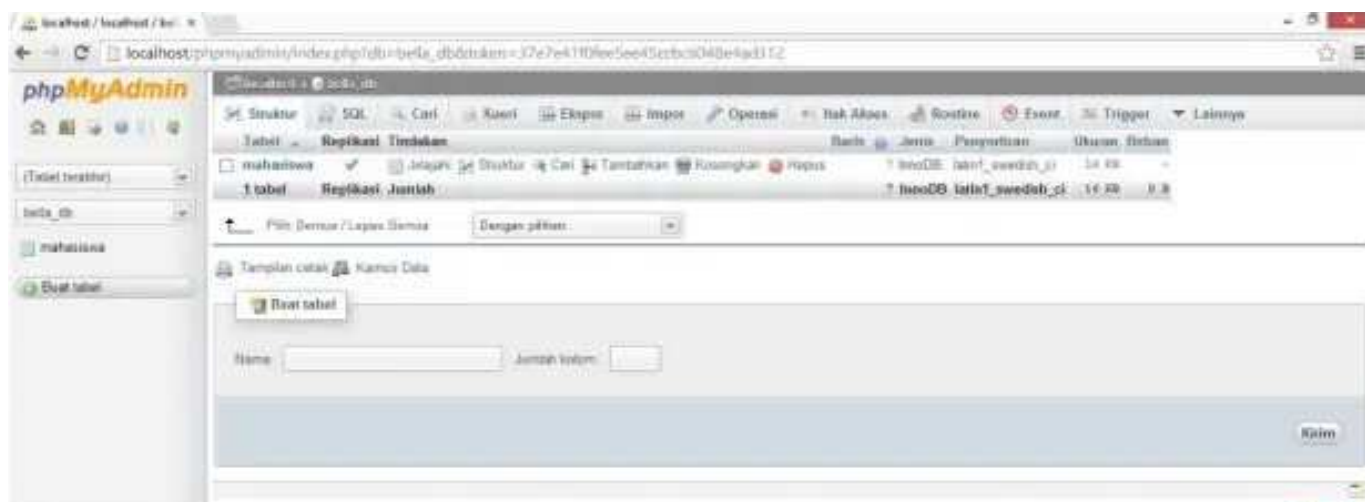
PHPMyAdmin adalah salah satu alat bantu dalam mengelola *database* MySQL. Fasilitas yang ada pada

PHPMyAdmin menggunakan GUI jadi mempermudah melakukan manipulasi data, dari pembuatan *database* sampai pada manipulasi dan organisasi data di dalamnya. Langkah-langkah dalam melakukan *backup* dan *recovery* pada PHPMyAdmin adalah sebagai berikut :

1. Jalankan PHPMyAdmin melalui *browser* dengan mengetikkan <http://localhost/PHPMyAdmin> pada *address bar*.



2. Buka salah satu *database* yang ada pada kolom sebelah kiri.

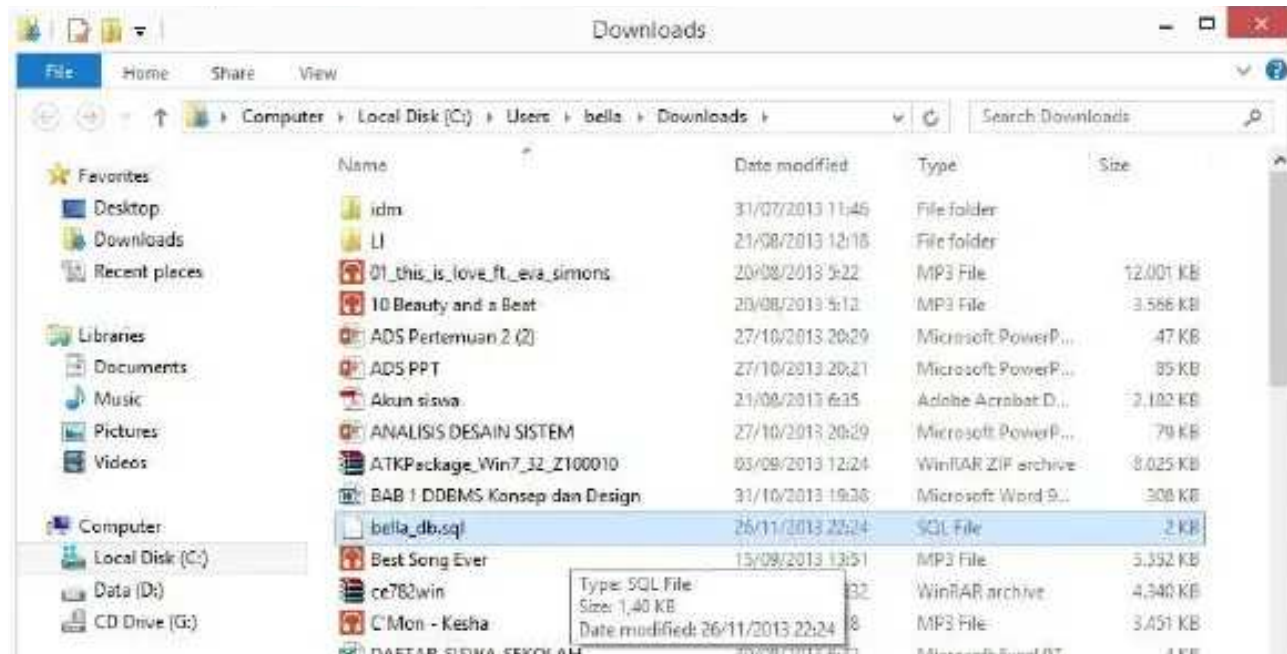


3. Klik Export yang ada pada *menu bar*.

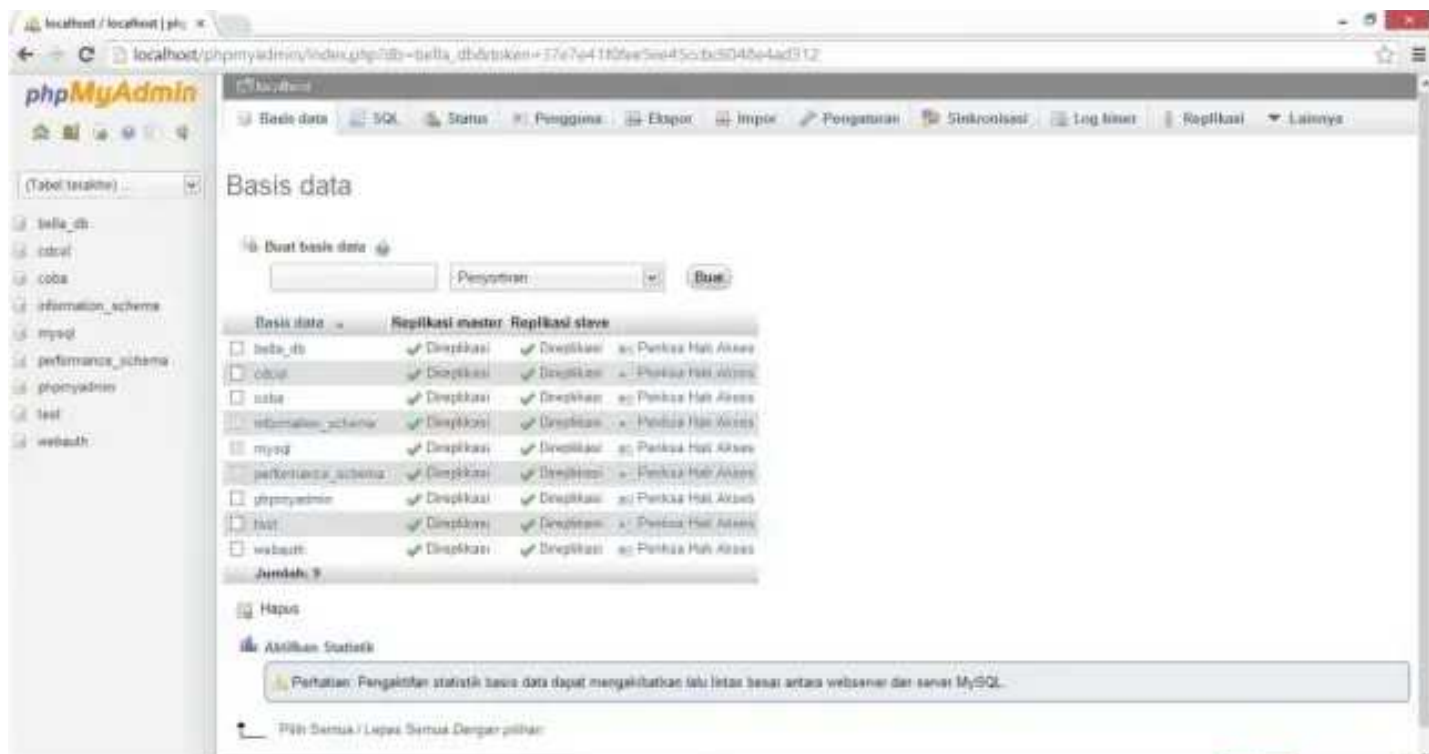


4. Pilihlah salah satu format *backup* yang nantinya akan disimpan, kemudian klik Kirim. Secara otomatis

file backup akan di-*download*, lalu buka file *download*.



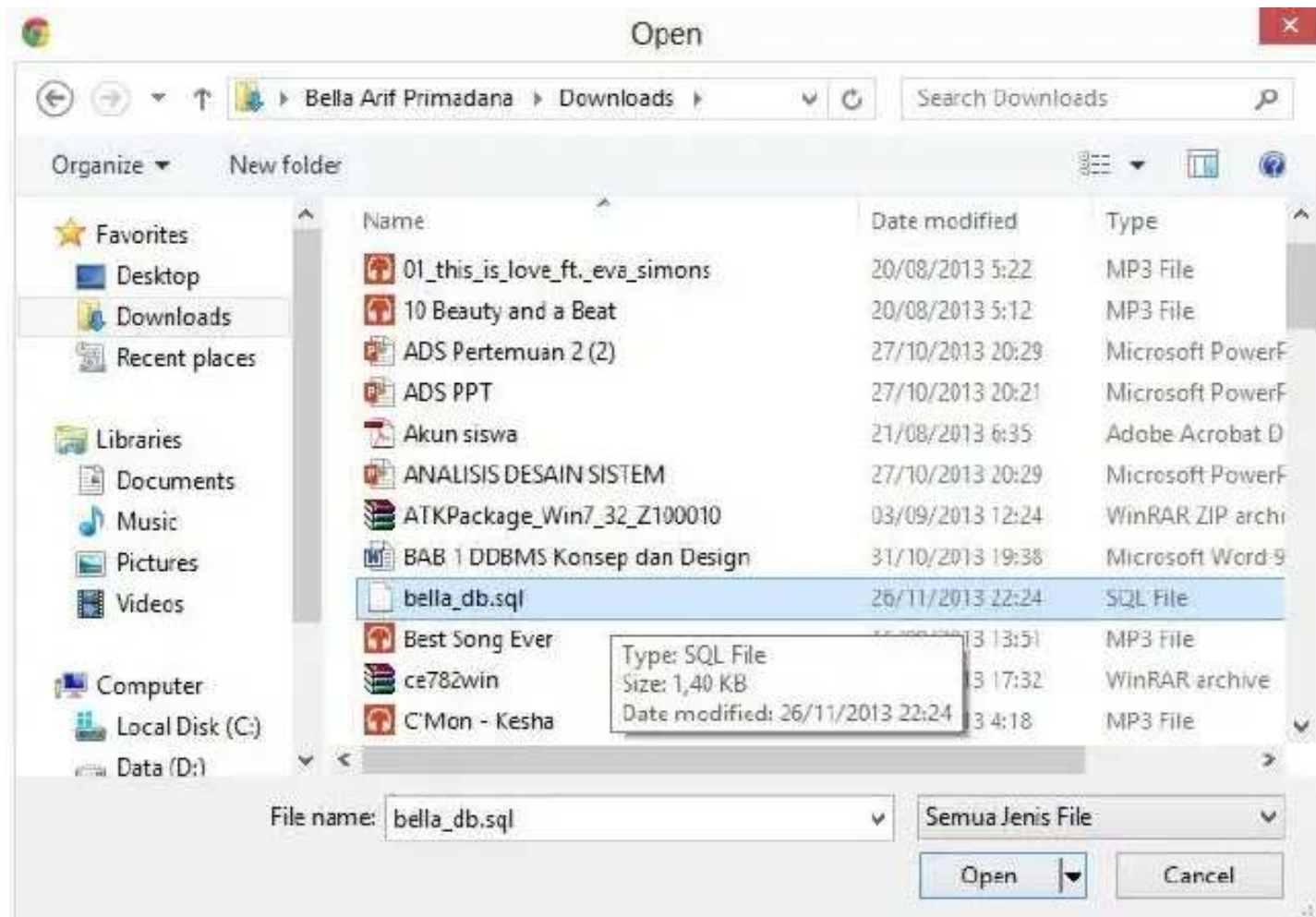
5. Buka kembali PHPMYAdmin untuk melakukan proses *recovery*.



6. Masuk ke dalam *database* yang ingin di-*recovery*. Klik menu Import pada *menu bar* untuk memulai proses *recovery*.



7. Klik *button* Pilih File untuk memilih *file backup*. Tampil kotak dialog untuk memilih lokasi *file backup* berada dan memilih *file backup*.



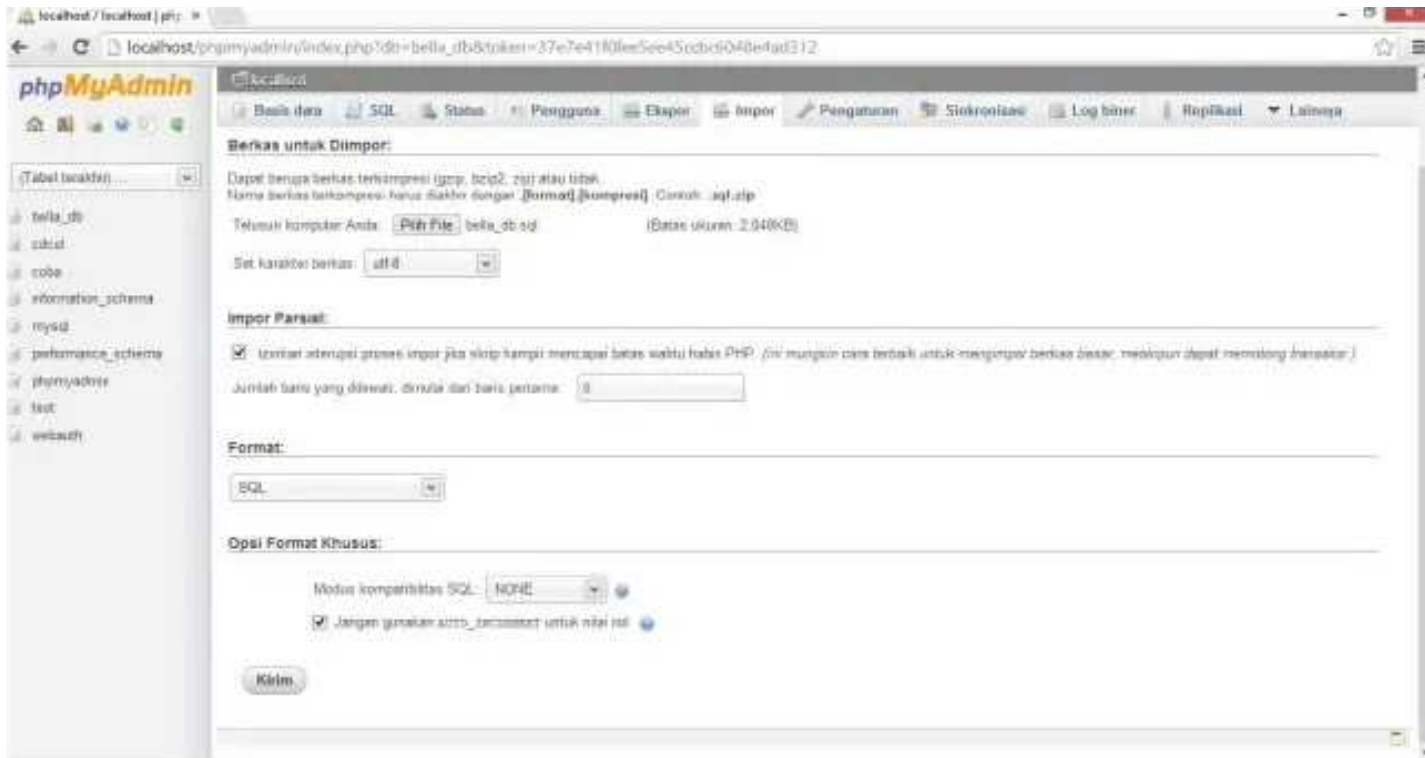
Setelah *file backup* ketemu, klik

dan proses *recovery* akan

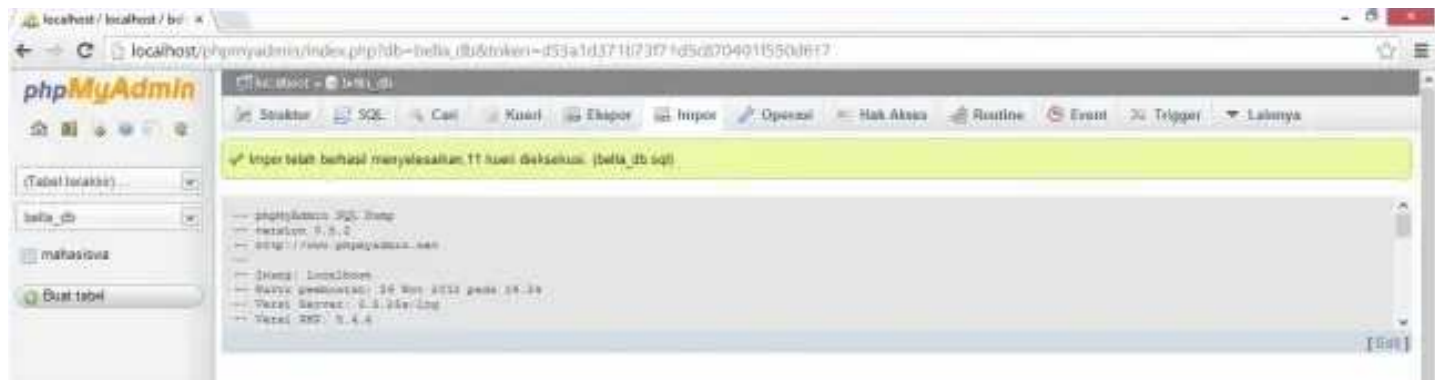
8. ^{Open} Tampilan akan kembali ke *browser*.

berlangsung.

Klik Kirim yang berada di sisi bawah



9. Jika proses *import* berhasil maka apa
muncul pesan bahwa *import*
database berhasil dilakukan.



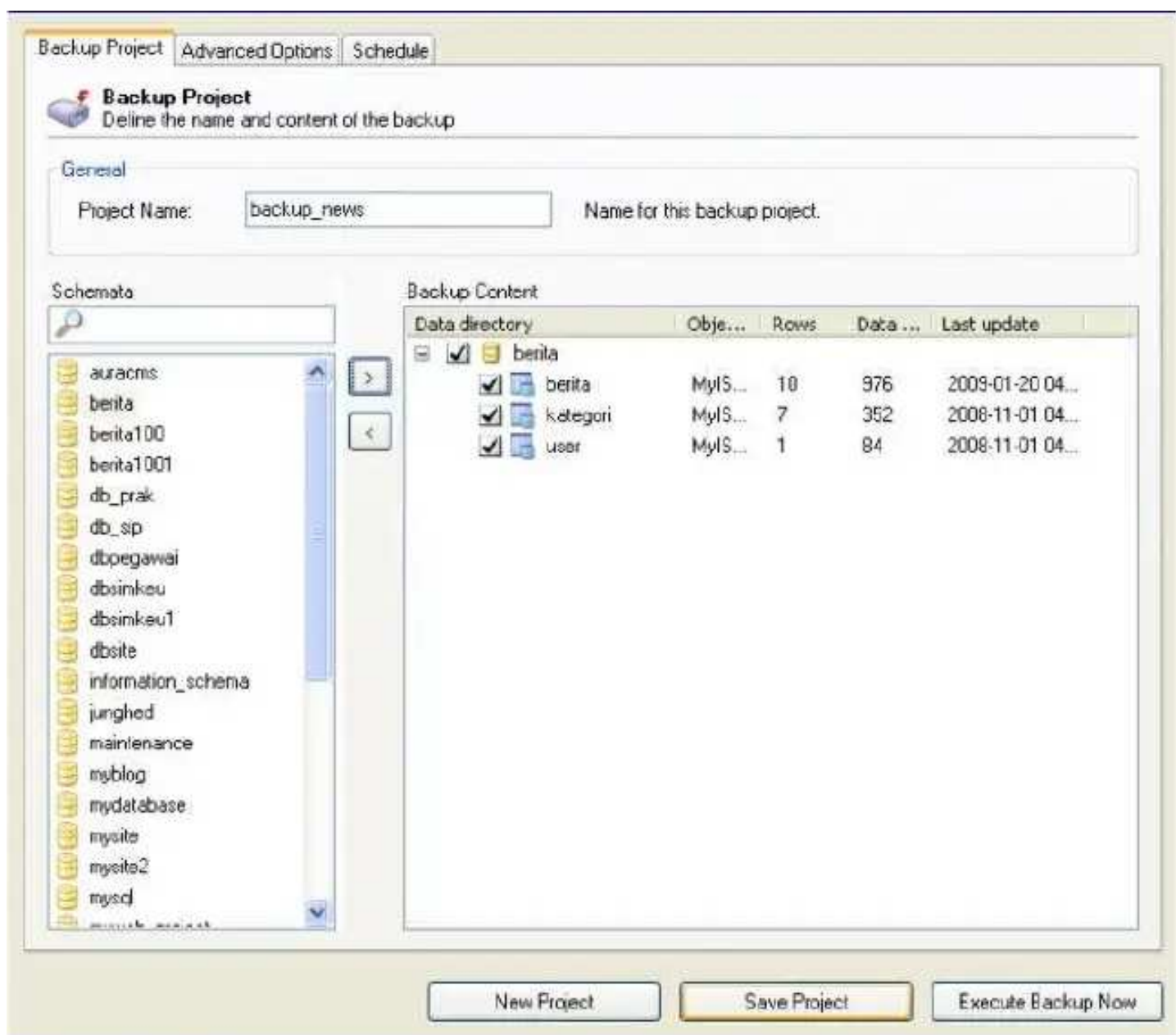
D. **Backup dan Recovery** melalui MySQL Administrator

MySQL Administrator menawarkan antarmuka grafis dalam proses *backup database*. Untuk *backup* dan *restore database* telah disediakan menu tersendiri pada MySQL Administrator. Langkah-langkah untuk *backup* dan

restore pada MySQL Administrator

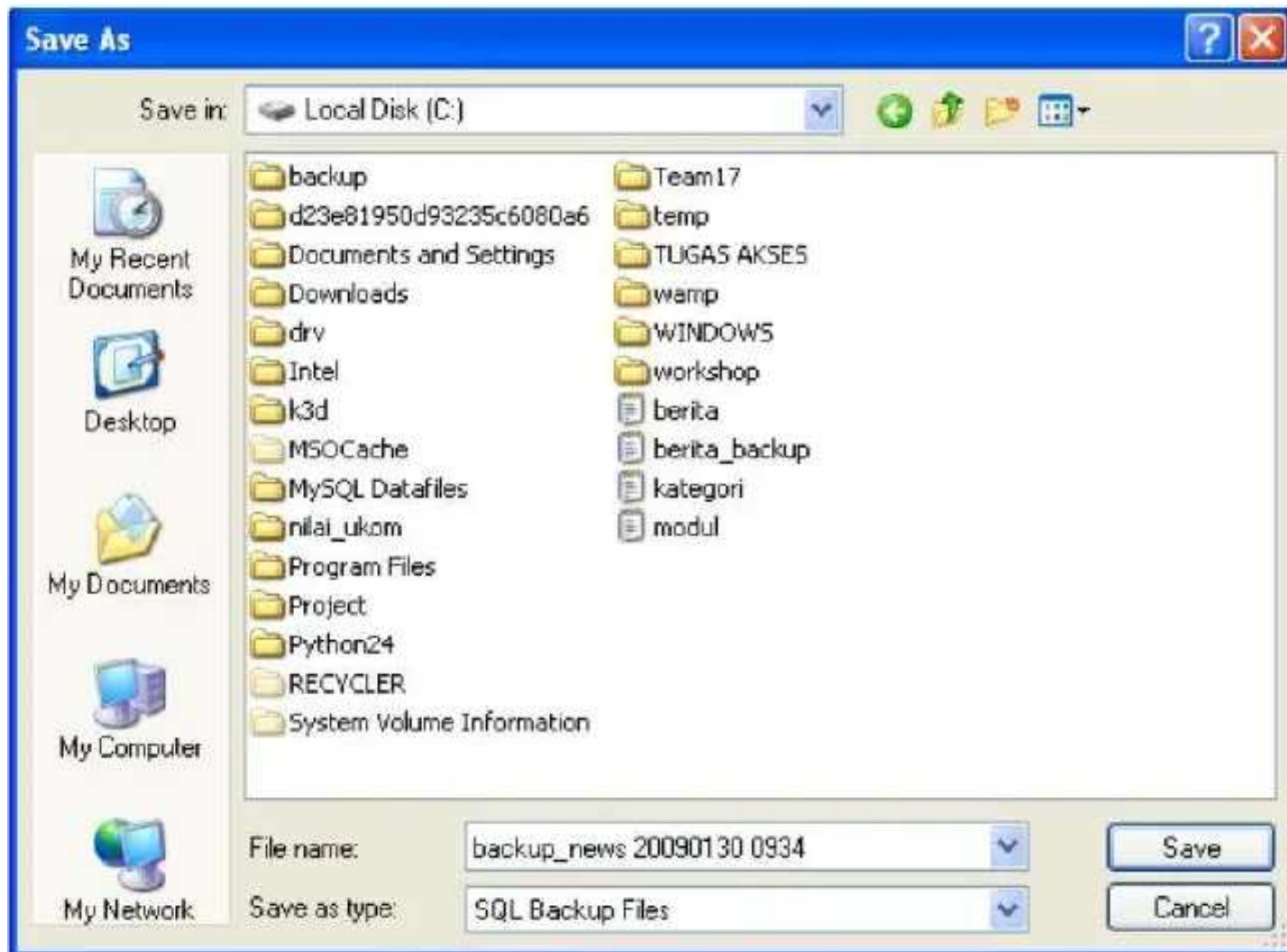
adalah sebagai berikut :

1. Buka MySQL Administrator. Jika ingin menyimpan *backup database* di dalam *project* baru, tekan tombol New Project dan beri nama *project* tersebut. Pilih *database* dan tabel yang akan di-*backup* ke dalam proyek tersebut.

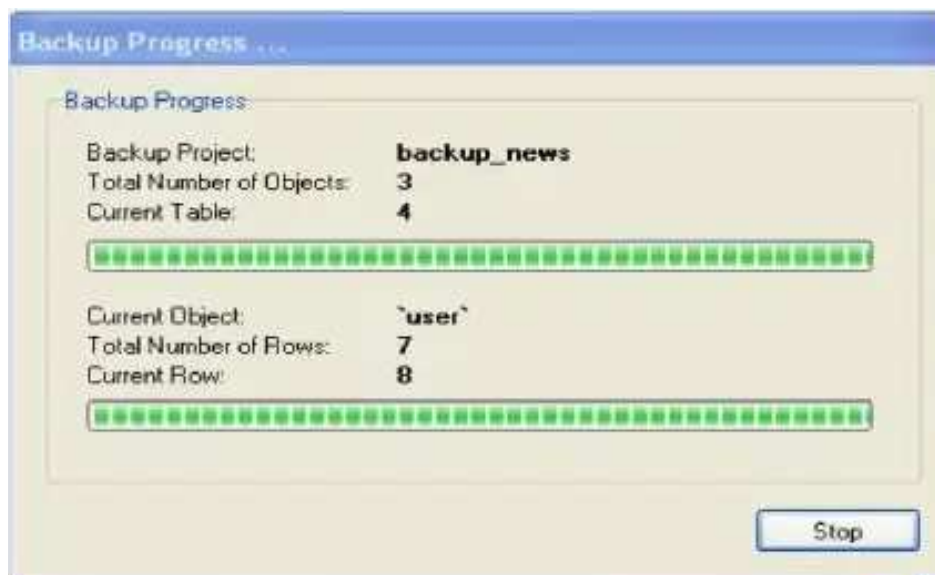


2. Jalankan proses *backup* dengan menekan tombol Execute Backup, MySQL Administrator akan

menampilkan jendela untuk menyimpan file *backup* tersebut.



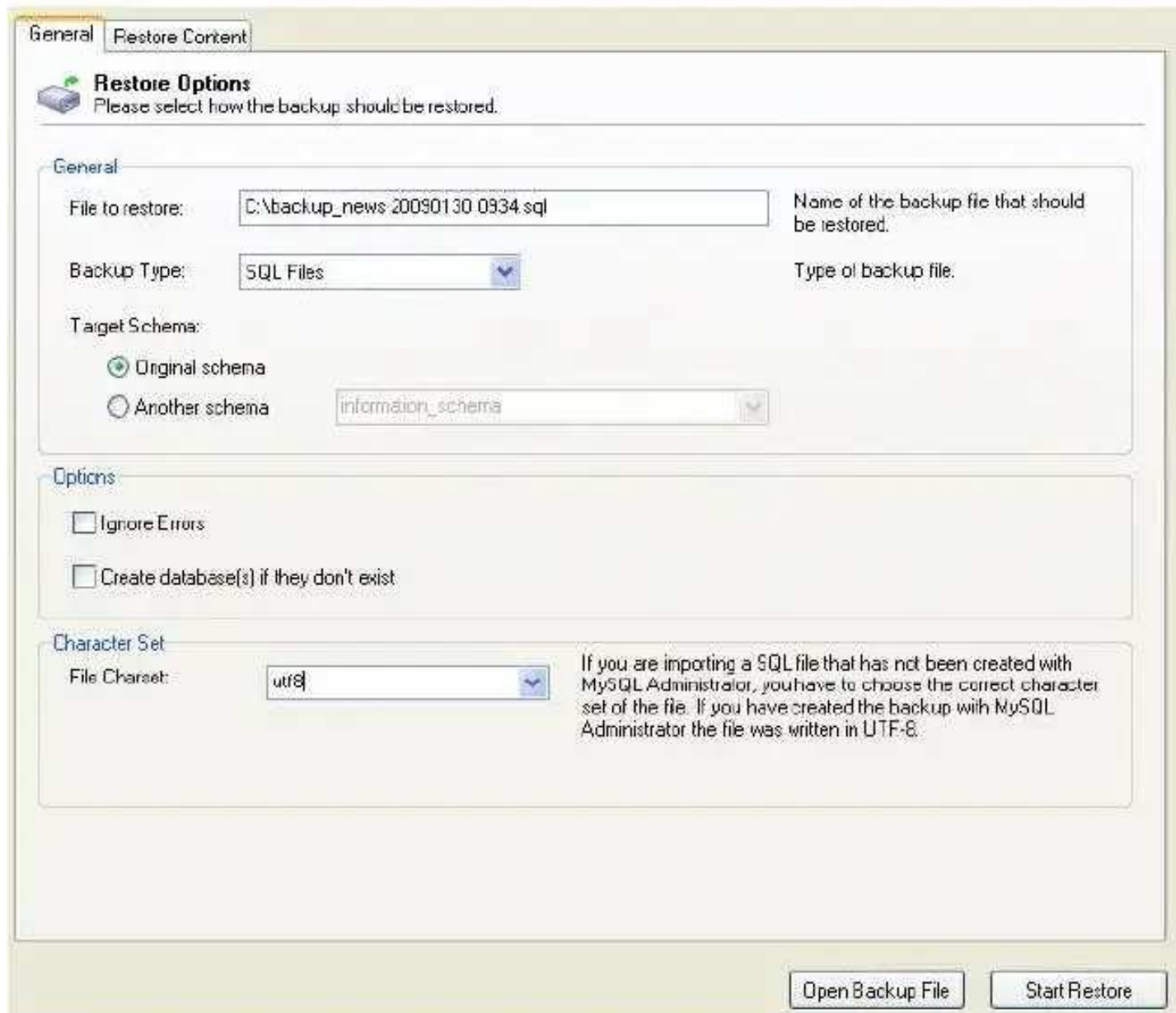
Jika direktori sudah dipilih, kemudian tekan Save sehingga proses *backup* akan berjalan.



3. Setelah proses selesai dan berhasil, maka akan muncul pesan bahwa "backup successfully".

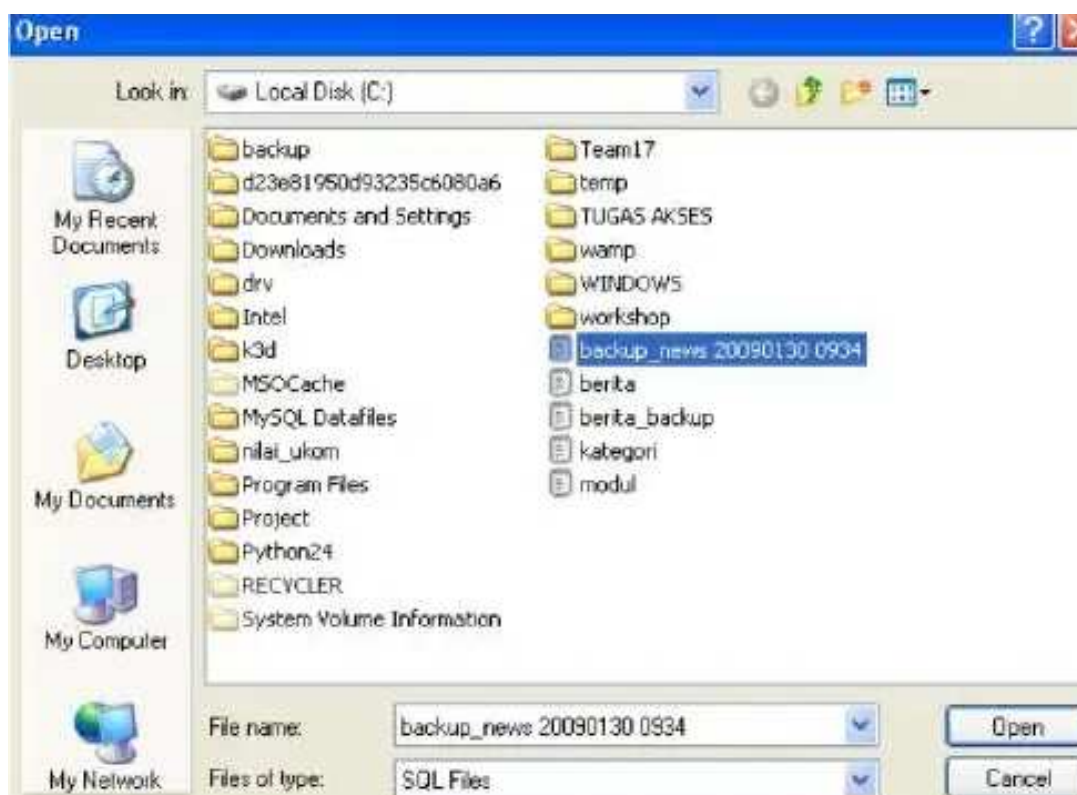


4. Untuk memulai proses *restore*, jalankan MySQL Administrator dan bukalah menu Restore.



Klik Open Backup File, tentukan tabel tujuan *backup file* tersebut di dalam direktori. Tentukan pilihan-pilihan bagi

proses *restore* tersebut. Tekanlah tombol Start Restrore untuk memulai proses *restore*.



5. Proses *restore* akan berlangsung.



KESIMPULAN

Backup dan *recovery* merupakan suatu proses penyalinan dan perbaikan data untuk menghindari terjadinya kerusakan data. Hal ini diperlukan dalam administrasi *database*, dimana *file backup* ini nantinya akan menjadi master data. Jika sewaktu-waktu dibutuhkan, master data akan di-*restore* sehingga data pada suatu sistem akan kembali seperti semua. Manfaat dari *backup* antara lain adalah untuk memudahkan apabila ingin memindahkan *database* dari *server* yang satu ke *server* yang lain, sebagai cadangan dalam preventif terhadap serangan virus/hacker, untuk *mirroring*. Sedangkan fungsi dari *restore/recovery* adalah untuk mengembalikan *database* ke keadaan semua sebelum *database* terjadi kerusakan serta mengambil data yang sudah di *backup*. *Backup* diperlukan karena adanya kegagalan transaksi (*transaksi failure*), kerusakan sistem (*system crash*), kegagalan/kerusakan disk (*disk failure*). *Backup* dapat dilakukan dengan dua cara *backup* statis dimana

database harus dinonaktifkan terlebih dahulu dan *backup* dinamis dimana dapat dilakukan ketika *database* saat beroperasi. Salah satu *database* yang terdapat fasilitas *backup* dan *recovery* adalah *database* MySQL. Terdapat beberapa cara dalam proses *backup* dan *recovery* pada MySQL, yaitu :

- Menggunakan perintah SQL
- Menggunakan MySQLDump
- Menggunakan MySQL Administrator
- Menggunakan PHPMyAdmin

DAFTAR PUSTAKA

Dikerektorat Pembinaan Sekolah Menengah Kejuruan. *Teknik Komputer dan Jaringan Bab 12 Backup dan Restrore Basisdata*. Hermawan, Galih. *Praktikum Sistem Basis Data Materi Backup, Recovery, dan Tutorial Replication*. Teknik Informatika Universitas Komputer Indonesia.
Yeki, Surya. *Pemulihan Proteksi Data – Basis Data*. Teknik Elektro Universitas Ahmad Dahlan.
Finandhita, Alif. 2010. *Modul Praktikum Sistem Basis Data*. Universitas Ilmu Komputer.
Saptono, Ristu. *Backup dan Recovery*.

atiknoviana.blogspot.com. 2010. *Backup dan Recovery*.

lylaqu-dwi.blogspot.com. 2010. *Backup dan Recovery*.