BAB X
EKSPOR DAN IMPOR DATA

Secara manual, kita memasukkan data ke dalam database dengan menggunakan perintah manipulasi data yang terdapat dalam mysql. Tetapi, kita dapat memasukkan data yang sudah kita miliki ke dalam database yang baru kita buat. Selain itu kita juga kita dapat mengeksporkan data yang sudah kita buat agar bisa digunakan oleh database lain.

X.1 Ekspor Data menggunakan OUTFILE

Dengan menggunakan perintah ekspor data, maka kita dapat mengekspor data yang telah kita buat ke dalam bentuk file lain, seperti file.txt. Berikut ini merupakan perintah-perintah yang digunakan untuk ekspor data:

Kode	Keterangan
SELECT[data/field]	Untuk menyeleksi fields mana saja yang ingin di ekspor.
INTO OUTFILE 'nama_file'	Untuk menentukan nama file tempat qta akan menyimpan data yang kita expor. Hasil keluarannya akan otomatis tersimpan pada folder mysql/ mysql5.1.36 /data.
FIELDS	Untuk memisahkan penyimpanan data berdasarkan setiap field.
➢ **TERMINATED BY 'isi'**	Merupakan subdause dari syntax FIELDS, berguna untuk memberi pembatas untuk setiap field nya. Jika tidak diatur maka default pemisah antar setiap field adalah TAB.
➢ **ENCLOSED BY 'isi'**	Merupakan subdause dari syntax FIELDS, berguna untuk menutup setiap fields dengan karakter yang di tentukan. Defaultnya adalah kosong.
➢ **ESCAPED BY 'isi'**	Merupakan subdause dari syntax FIELDS, berguna untuk menandai setiap karakter istimewa yang terdapat di dalam field yang akan di copy, misalnya karakter kutip satu (‘).
LINES	Untuk memisahkan penyimpanan data berdasarkan

	setiap barisnya.
➢ **STARTING BY 'isi'**	Merupakan subclause dari syntax LINES, berguna untuk memberikan string pada awal setiap baris.
➢ **TERMINATED BY 'isi'**	Merupakan subclause dari syntax LINES, berguna untuk memberi pembatas setiap baris datanya.
FROM nama_tabel	Menentukan nama tabel yang fields nya akan di ekspor.

Karakter yang biasa digunakan untuk perintah THERMINATED BY, ENCLOSED BY:

Karakter	**Arti**
\'	Kutip Satu
****	Backslash
\n	Enter / Pindah Baris
\r	Carriage return
\s	Spasi
\t	Tab

Berikut ini merupakan contoh penggunaan syntax untuk ekspor data:

1. Pertama, buat database, table dan masukan values ke dalam table tersebut.

```
CREATE DATABASE contoh1;
use contoh1;
CREATE TABLE lagu
   (
   kode smallint unsigned not null auto_increment primary key,
   judul varchar(50) not null,
   band varchar(20),
   album varchar(50),
   tahun varchar(5)
    );

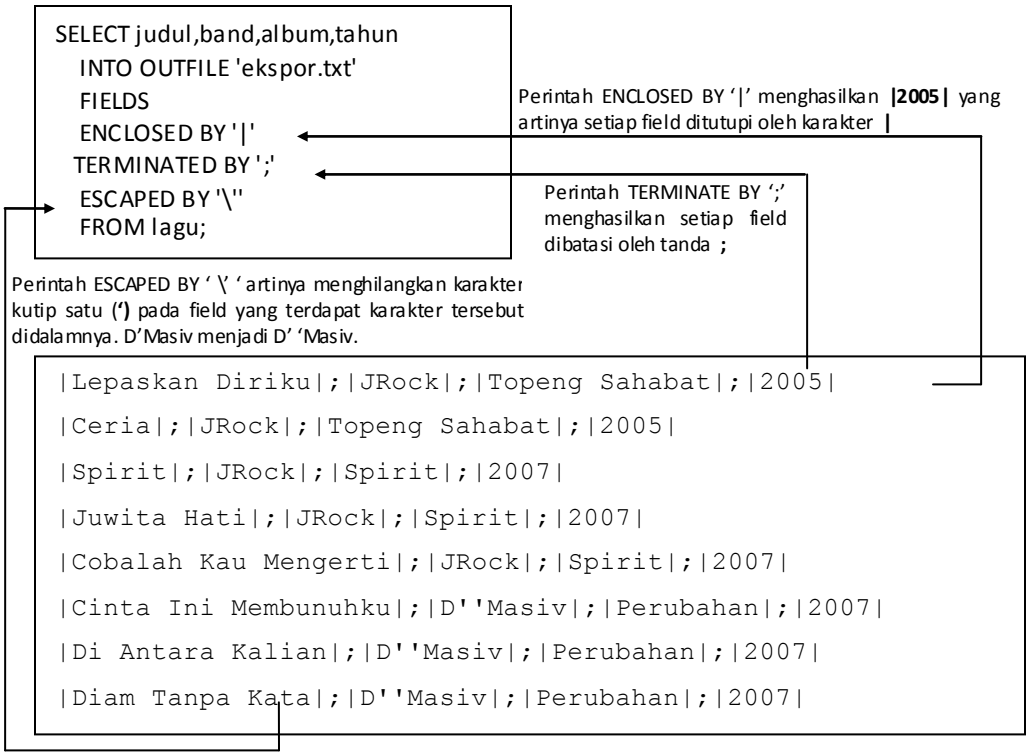
 INSERT INTO lagu VALUES
    (null,'Lepaskan Diriku','JRock','Topeng Sahabat','2005'),
    (null,'Ceria','JRock','Topeng Sahabat','2005'),
    (null,'Spirit','JRock','Spirit','2007'),
    (null,'Juwita Hati','JRock','Spirit','2007'),
    (null,'Cobalah Kau Mengerti','JRock','Spirit','2007'),
    (null,'Cinta Ini Membunuhku','D\'Masiv','Perubahan','2007'),
    (null,'Di Antara Kalian','D\'Masiv','Perubahan','2007'),
    (null,'Diam Tanpa Kata','D\'Masiv','Perubahan','2007'),
    (null,'Merindukanmu','D\'Masiv','Perubahan','2007'),
    (null,'Dan Kamu','D\'Masiv','Perubahan','2007');
```

2. Kemudian Ketik Syntax untuk mengekspor, ,

   Hasil dari sintax ekspor akan di buat file ekspor.txt di dalam folder mysql/ mysql5.1.36 /data, dalam contoh ini, file tersebut dibuka menggunakan wordpad.

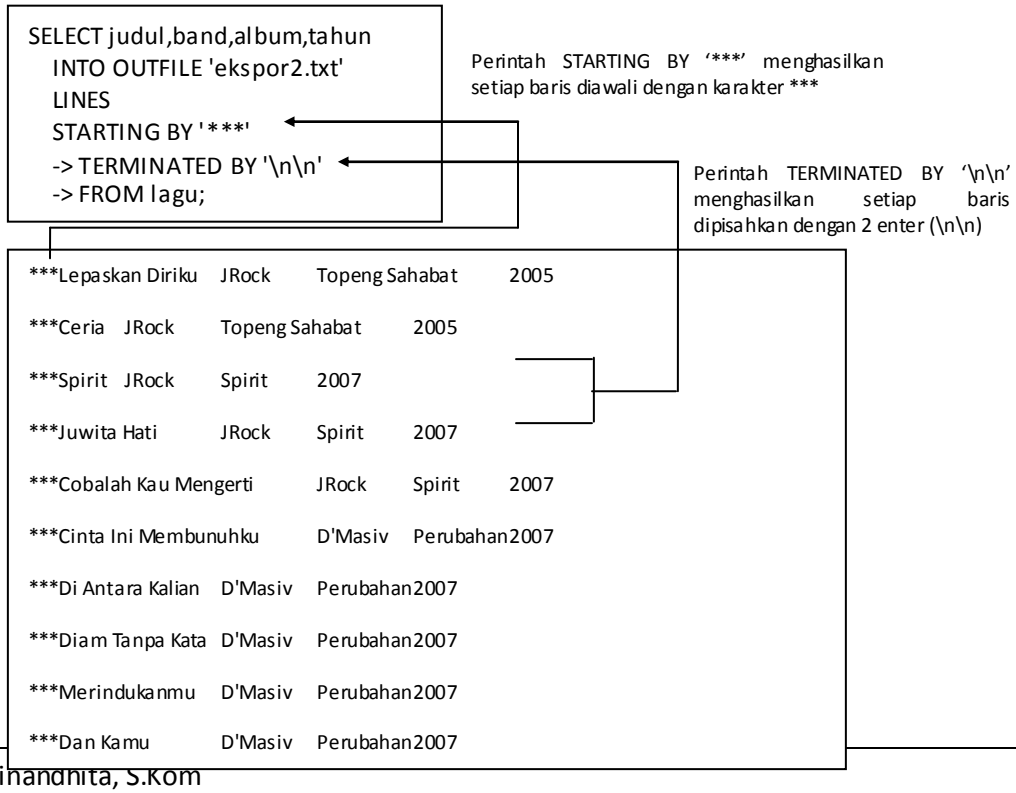
   a. Menggunakan FIELDS

      Dalam perintah FIELDS, semua perintah diterapkan pada setiap fields.



   b. Menggunakan LINES

      Dalam Perintah LINES, setiap perintah diterapkan pada setiap baris.



X.2 Ekspor Data Menggunakan DUMP FILE

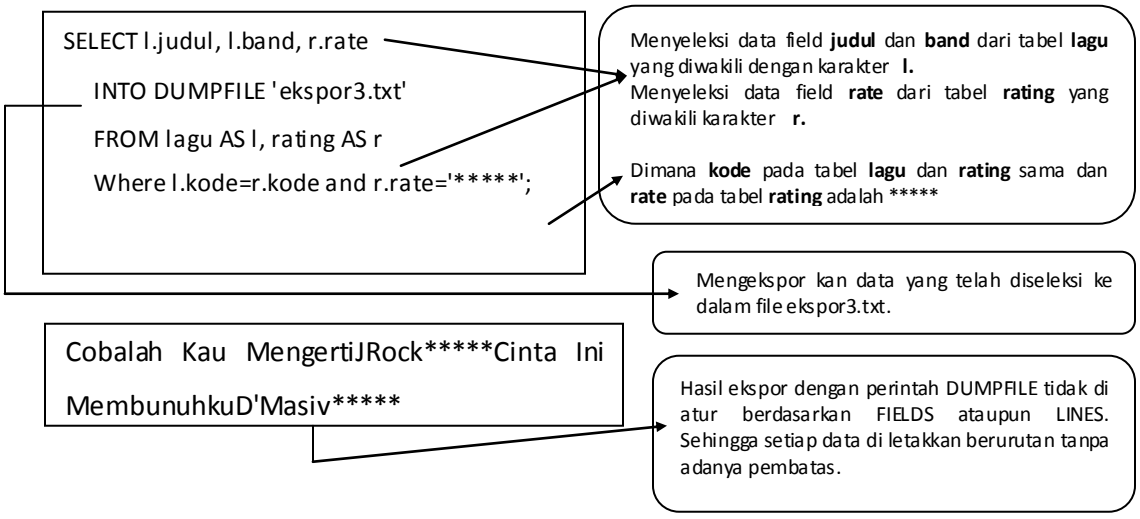
Mengekspor data dengan menggunakan perintah DUMP FILE lebih mudah dari pada menggunakan perintah OUTFILE, karena kita tidak perlu melakukan pengaturan LINES atau FIELDS.

Berikut ini adalah contoh penggunaan perintah DUMP FILE:

1. Tambahkan tabel baru dan value nya dengan nama rating ke dalam database contoh1

```
CREATE TABLE rating
  (
  kode smallint unsigned auto_increment primary key,
  rate varchar(5)
  );
INSERT into rating VALUES
  (NULL,'***'),
  (NULL,'****'),
  (NULL,'**'),
  (NULL,'**'),
  (NULL,'*****'),
  (NULL,'*****'),
  (NULL,'****'),
  (NULL,'***'),
  (NULL,'***'),
  (NULL,'**');
```

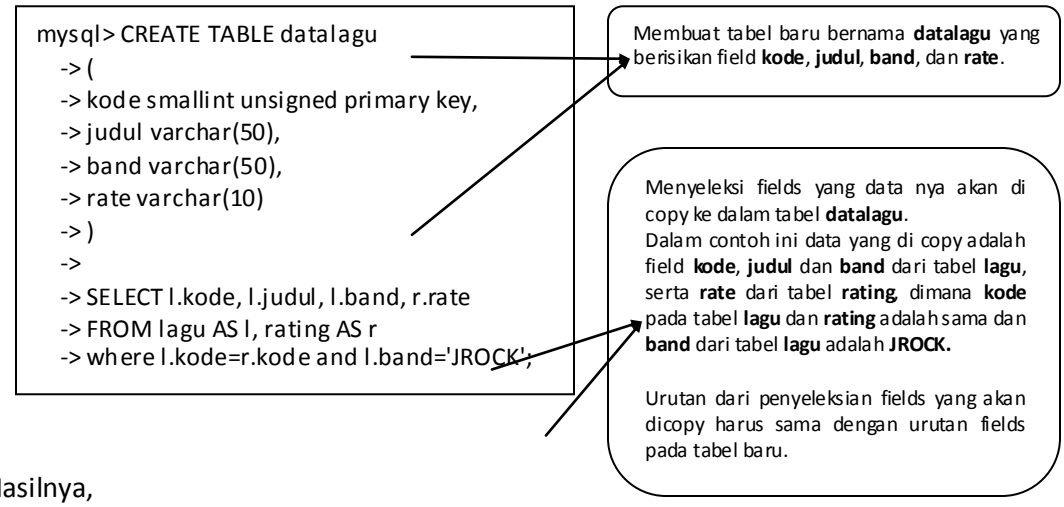
2. Ketikkan Syntax ekspor dengan DUMPFILE



X.3 Meng Copy fields dari tabel ke tabel lain

Mysql menyediakan perintah untuk dapat menyalin fields dan isi nya yang diseleksi dari suatu tabel ke tabel lainnya. Berikut ini merupakan contoh penggunaan perintah untuk menyalin fields tabel:

X.3.1 Menyalin fields tabel ke tabel baru



Hasilnya,

jika kita menampilkan tabel datalagu adalah sbb:

```
select * from datalagu;
+------+----------------------+-------+-------+
| kode | judul                | band  | rate  |
+------+----------------------+-------+-------+
|    1 | Lepaskan Diriku      | JRock | ***   |
|    2 | Ceria                | JRock | ****  |
|    3 | Spirit               | JRock | **    |
|    4 | Juwita Hati          | JRock | **    |
|    5 | Cobalah Kau Mengerti | JRock | ***** |
+------+----------------------+-------+-------+
```

X.3.2 Menyalin Fields Tabel ke Tabel Yang Sudah Ada

a. Menggunakan perintah INSERT

Dengan perintah INSERT, kita dapat menambahkan fields dari suatu tabel ke tabel lain yang memang sudah ada. Dengan catatan bahwa tabel tujuan harus memiliki nama field yang sama. Perintah INSERT akan mendapatkan hasil EROR jika dari data yang di copy terdapat data dengan primary key yang sama dengan field tujuan.

Contoh penggunaan perintah INSERT untuk menyalin fields:

- ✓ Dengan asumsi bahwa kita sudah mempunyai tabel bernama datalagu2, di dalamnya terdapat fields kode, judul, band dan rating, dimana tabel tersebut dalam keadaan kosong.

```
INSERT INTO datalagu2
SELECT l.kode, l.judul, l.band, r.rate
 FROM lagu AS l, rating AS r
 WHERE l.kode=r.kode and l.band='D\'Masiv';
```

datalagu2 merupakan Tabel

Menyeleksi fields yang data nya akan di copy ke dalam tabel **datalagu2**.
Dalam contoh ini data yang di copy adalah field **kode**, **judul** dan **band** dari tabel **lagu**, serta **rate** dari tabel **rating**, dimana **kode** pada tabel **lagu** dan **rating** adalah sama dan **band** dari tabel **lagu** adalah **D'Masiv.**

Hasilnya adalah:

```
SELECT * FROM datalagu2 ;
+------+------------------------------+---------+--------+
| kode| judul                         | band    | rating|
+------+------------------------------+---------+--------+
|   6   | Cinta Ini Membunuhku |D'Masiv|*****|
|   7   | Di Antara Kalian          |D'Masiv|****  |
|   8   | Diam Tanpa Kata           |D'Masiv|***   |
|   9   | Merindukanmu              |D'Masiv|***   |
|  10  | Dan Kamu                     |D'Masiv|**    |
+------+------------------------------+---------+-------+
```

b. Manggunakan perintah REPLACE

Pada dasarnya perintah REPLACE memiliki fungsi yang sama dengan perintah INSERT. Hanya saja, dengan perintah REPLACE, jika ada primary key yang sama, maka data yang ada di tabel tujuan akan di replace dengan data baru yang di copy kan.

Contoh penggunaan perintah REPLACE:

```
REPLACE INTO datalagu
   -> SELECT l.kode, l.judul, l.band, r.rate
   -> FROM lagu AS l, rating AS r
   -> WHERE l.kode = r.kode and r.rate='***';

Query OK, 4 rows affected (0.05 sec)
Records: 3  Duplicates: 1  Warnings: 0
```

Menyalin dengan perintah REPLACE ke dalam tabel **datalagu** yang sebelumnya telah berisi 5 data.

Hasil dari seleksi sebenarnya menghasilkan 3 data, tapi di karenakan ada data yang sama, maka data tersebut di replace dengan data yang baru di salin.

Tabel datalagu sebelum di REPLACE

```
select * from datalagu;
+------+--------------------+-------+-------+
| kode | judul              | band  | rate  |
+------+--------------------+-------+-------+
|   1 | Lepaskan Diriku     | JRock | ***   |
|   2 | Ceria               | JRock | ****  |
|   3 | Spirit              | JRock | **    |
|   4 | Juwita Hati         | JRock | **    |
|   5 | Cobalah Kau Mengerti | JRock | ***** |
+------+--------------------+-------+-------+
```

Tabel datalagu setelah di REPLACE

```
select * from datalagu;
+------+--------------------+---------+-------+
| kode | judul              | band    | rate  |
+------+--------------------+---------+-------+
|   1 | Lepaskan Diriku     | JRock   | ***   |
|   2 | Ceria               | JRock   | ****  |
|   3 | Spirit              | JRock   | **    |
|   4 | Juwita Hati         | JRock   | **    |
|   5 | Cobalah Kau Mengerti | JRock   | ***** |
|   8 | Diam Tanpa Kata     | D'Masiv | ***   |
|   9 | Merindukanmu        | D'Masiv | ***   |
+------+--------------------+---------+-------+
7 rows in set (0.01 sec)
```

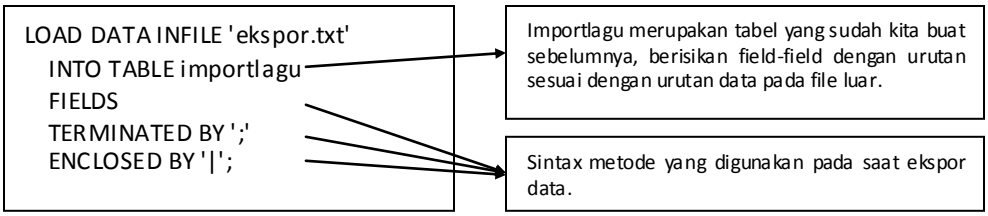
X.4 Mengimport kan data dari file ke dalam tabel database

Selain dapat mengeksporkan dan menyalin data, MYSQL juga menyediakan fasilitas untuk dapat mengimpor data dari file luar ke dalam database. Mengimpor data dapat dilakukan dengan 3 cara, yaitu:

X.4.1 Menggunakan perintah LOAD DATA

Jika kita mnegimportkan data dengan menggunaka perintah LOAD DATA, kita harus memberikan spesifikasi penyimpanan data yang akan kita import. Misalnya jika data yang akan kita import di simpan dengan metode TERMINATED BY , ENCLOSED BY, FIELDS, LINE, dan lainnya, maka kita harus menyebutkan metode tersebut di dalam sintax import data.

Berikut ini merupakan contoh penggunaan LOAD DATA pada file **ekspor.txt** yang merupakan hasil ekspor data pada contoh pertama dalam BAB ini:



X.4.2 Menggunakan perintah MYSQL

Syntax yang digunakan:

mysql test < "c:\program files\mysql\mysql server 4.1\data\test\CDsRock.sql"

X.4.3 Menggunakan perintah SOURCE

Dengan menggunakan perintah SOURCE untuk mengimport data dari luar, kita hanya memerlukan spesifikasi letak dari file tersebut dan nama file nya. Syntax yang digunakan adalah:

mysqlimport --user=root --password=pw1 test "c:\program files\mysql\mysql server4.1\data\test\CDs3.sql"