

Analisis Database Pada Sistem Informasi Simpan Pinjam

Lalu Delsi Samsumar¹, Emi Suryadi², Yumarlin MZ³

STMIK Mataram^{1,2}

Universitas Janabadra³

emisuryadi@gmail.com

Abstrak – Database ini dibuat dengan menggunakan MySQL, yang melewati tahap analisis dan perancangan. Begitu pentingnya era digital atau era data pada saat ini, tidak heran administrator database berhati-hati dalam memberikan batasan hak akses kepada pengguna, dimaksudkan agar data atau informasi dapat dijaga dan tidak setiap pengguna dapat mengakses data atau informasi dari setiap tabel. Adanya batasan hak akses ini dapat menjamin keaslian isi dari database yang dibuat oleh administrator, setiap transaksi yang dilakukan oleh pengguna yang bukan administrator memiliki hak akses terbatas untuk mengelola isi database, dengan begitu keamanan yang dimiliki sistem database ini cukup baik. Untuk menjaga keamanan data atau informasi yang ada di database diharapkan memiliki back-up datanya, dikarenakan untuk mengantisipasi jika terjadi kehilangan dan kerusakan data.

Kata Kunci : Database, MySQL, Transaksi, Keamanan dan Back-up.

1. Latar Belakang

Kemajuan teknologi informasi memberikan kemudahan bagi pekerjaan manusia, sehingga pekerjaan yang dianggap sulit dan rumit dapat terselesaikan dengan mudah. Teknologi informasi dimanfaatkan oleh banyak orang baik dari perusahaan swasta atau negeri yang semakin bertambah. Data-data atau informasi yang semakin hari semakin meningkat jumlah dan ukurannya, perlu menggunakan sebuah media penyimpanan yang memiliki ukuran yang besar sehingga data yang tersimpan tersebut dapat terjaga kerahasiannya dengan baik. Teknologi yang digunakan untuk mengelola dan menyimpan data-data atau informasi yang dianggap penting tersebut dapat menggunakan teknologi Database.

Database adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktifitas untuk memperoleh informasi. Data-data atau informasi yang tersimpan di sebuah database dapat diakses oleh *user* yang berhak untuk mengakses data tersebut. Data atau informasi yang bersifat privasi dapat terjaga keasliannya dengan baik.

Hak akses perlu diterapkan pada sebuah data atau informasi yang ada pada database, dengan tujuan untuk menjaga kerahasiaan atau keaslian data. Tanpa disadari banyak para *developer* (pengembang) *software* berlomba-lomba membuat sebuah sistem agar dapat mengolah dan mengamankan data dari orang lain yang tidak diberikan hak akses atau wewenang. Batasan hak akses ini sangat penting diterapkan karena menyangkut tentang transaksi yang sifatnya rahasia, yang tidak perlu diketahui oleh orang

yang tidak berwenang, dengan begitu sistem keamanan database ini dibuat agar menjadi sistem yang mampu memberikan pengamanan terhadap isi dari database yang dibuat.

2. Kajian Pustaka

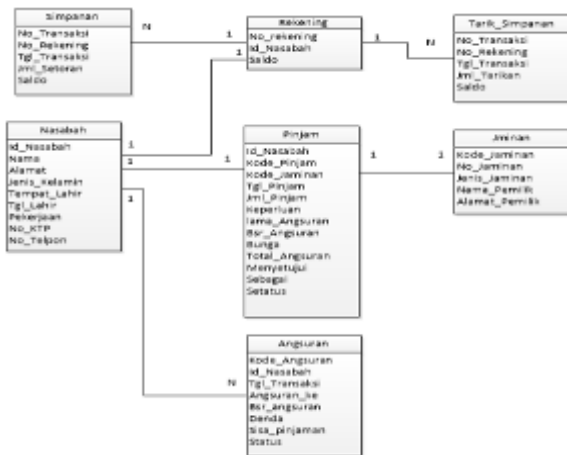
Database merupakan sekumpulan data yang saling terintegrasi satu sama lain dan terorganisasi berdasarkan sebuah skema atau struktur tertentu dan tersimpan pada sebuah *hardware* komputer. Database terdiri dari beberapa tabel (lebih dari satu tabel) yang saling terorganisir. Tabel digunakan untuk menyimpan data yang terdiri dari baris dan kolom. Data tersebut dapat ditampilkan, dimodifikasi, dan dihapus dari tabel. Setiap pemakai (*user*) yang diberi wewenang (otoritas) saja yang dapat melakukan akses terhadap data tersebut. (Arief, 2006). Basis data memiliki operasi dasar, yaitu, Pembuatan basis data (*create database*), Penghapusan basis data (*drop database*), Pembuatan file atau tabel (*create tabel*), Penghapusan file atau tabel (*drop tabel*), Penambahan atau pengisian data baru (*insert*), Pengambilan data (*retrieve atau search*), Pengubahan data (*update*), Penghapusan data (*delete*)

a. MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL,

namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya yakni SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Hasil jurnal publikasi yang dimiliki oleh Vindhy Agus Setiawan (STMIK Amikom Yogyakarta) dengan judul, Analisis Dan Perancangan Sistem Informasi Simpan Pinjam Pada LKM Gerembeng Bali. Dari hasil jurnal tersebut penulis mengambil bahan analisa khususnya pada bagian relasi antar tabel. Disini pemilik jurnal publikasi tersebut tidak mencantumkan *constraint check* dan tipe data yang digunakan pada setiap tabel di jurnal publikasinya, seperti yang dapat dilihat pada Gambar 1 di bawah ini.



Gambar 1 Relasi Antar Tabel

Pada Gambar 1 di atas tidak mencantumkan tipe data dan *constraint check* pada relasi tabel yang dibuat. Jurnal publikasi tersebut dianalisis dan ditentukan tipe data dari setiap tabel yang ada.

b. Perancangan Database.

Sistem basis data (database) diartikan sebagai kumpulan *file* (tabel) yang saling berhubungan (dalam sebuah basis data pada sistem komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan atau program lain untuk mengakses dan memanipulasi *file-file* (tabel-tabel) tersebut. Untuk memperoleh database yang diinginkan harus melalui tahap normalisasi. Tujuan dilakukan normalisasi ini adalah menghasilkan struktur tabel yang normal dan meminimalisasi kerangkapan data (*redundancy data*). Berikut hasil deskripsi setiap tabel yang ada dapat dilihat pada Tabel 1 berikut ini.

Tabel 1. Dekripsi Tabel

```
mysql> desc tarik_simpanan;
```

Field	Type	Null	Key	Default	Extra
No_Transaksi	char(12)	NO	PRI	NULL	
No_Rekening	char(20)	NO	UNI	NULL	
Tanggal_Transaksi	date	YES		NULL	
Jumlah_Tarikan	char(12)	YES		NULL	
Saldo	char(15)	YES		NULL	

```
mysql> desc rekening;
```

Field	Type	Null	Key	Default	Extra
No_Rekening	char(20)	NO	PRI	NULL	
ID_Nasabah	char(12)	NO	UNI	NULL	
Saldo	char(15)	YES		NULL	

```
mysql> desc simpanan;
```

Field	Type	Null	Key	Default	Extra
No_Transaksi	char(12)	NO	PRI	NULL	
No_Rekening	char(20)	NO	UNI	NULL	
Tanggal_Transaksi	date	YES		NULL	
Jumlah_Setoran	char(12)	YES		NULL	
Saldo	char(15)	YES		NULL	

```
mysql> desc jaminan;
```

Field	Type	Null	Key	Default	Extra
Kode_Jaminan	char(12)	NO	PRI	NULL	
No_Jaminan	char(12)	NO	UNI	NULL	
Jenis_Jaminan	varchar(20)	NO		NULL	
Nama_Pemilik	varchar(20)	NO		NULL	
Alamat_Pemilik	varchar(25)	YES		NULL	

```
mysql> desc nasabah;
```

Field	Type	Null	Key	Default	Extra
ID_Nasabah	char(12)	NO	PRI	NULL	
Nama_Nasabah	varchar(20)	NO		NULL	
Alamat_Nasabah	varchar(20)	NO		NULL	
Tenis_Kelamin	varchar(15)	YES		NULL	
Tempat_Lahir	varchar(25)	YES		NULL	
Tgl_Lahir	date	YES		NULL	
Pekerjaan	varchar(20)	YES		NULL	
No_KTP	char(25)	NO	UNI	NULL	
No_Telepon	char(12)	NO	UNI	NULL	

```
mysql> desc pinjam;
```

Field	Type	Null	Key	Default	Extra
ID_Nasabah	char(12)	NO	PRI	NULL	
Kode_Pinjam	char(12)	NO	UNI	NULL	
Kode_Jaminan	char(12)	NO	UNI	NULL	
Tanggal_Pinjam	date	YES		NULL	
Jumlah_Pinjam	char(15)	YES		NULL	
Keperluan	varchar(20)	YES		NULL	
Lama_Angsuran	varchar(15)	YES		NULL	
Besar_Angsuran	int(11)	YES		NULL	
Runga	int(11)	YES		NULL	
Total_Angsuran	int(11)	YES		NULL	
Menyetujui	varchar(25)	YES		NULL	
Sebagai	varchar(20)	YES		NULL	
Status	varchar(15)	YES		NULL	

```
mysql> desc angsuran;
```

Field	Type	Null	Key	Default	Extra
Kode_Angsuran	char(12)	NO	PRI	NULL	
ID_Nasabah	char(12)	NO	UNI	NULL	
Tanggal_Transaksi	date	YES		NULL	
Angsuran_Ke	varchar(10)	YES		NULL	
Besar_Angsuran	char(15)	NO		NULL	
Denda	char(15)	YES		NULL	
Sisa_Pinjaman	char(15)	YES		NULL	
Status	varchar(12)	YES		NULL	

Masing – masing tipe data di atas memiliki daya tampung karakter pada setiap data, yang dimana setiap data mampu tersimpan sesuai dengan ukuran yang diberikan. Sehingga jika kalau seorang *user* menginputkan data lebih dari karakter yang disediakan maka data tersebut tidak tersimpan sepenuhnya didalam isian tabel database. Untuk mengetahui *constraint check* yang dimiliki dari beberapa tabel sebagai contoh dapat dilihat Gambar 2 sebagai berikut.

```
mysql> CREATE TABLE Tarik_Simpanan (
-> No_Transaksi char (12) unique not null,
-> No_Rekening char (20) unique not null,
-> Tanggal_Transaksi Date,
-> Jumlah_Tarikan Char (12),
-> Saldo Char (15)
-> );

mysql> CREATE TABLE Jaminan (
-> Kode_Jaminan Char (12) unique not null,
-> No_Jaminan Char (12) unique not null,
-> Jenis_Jaminan Varchar (20) not null,
-> Nama_Pemilik Varchar (20) not null,
-> Alamat_Pemilik Varchar (25),
-> Primary Key (Kode_Jaminan)
-> );

mysql> CREATE TABLE Angsuran (
-> Kode_Angsuran Char (12) unique not null,
-> ID_Nasabah Char (12) unique not null,
-> Tanggal_Transaksi Date,
-> Angsuran_Ke Varchar (10),
-> Besar_Angsuran Varchar (15),
-> Denda Char (15),
-> Sisa_Pinjaman Char (15),
-> Status Varchar (12),
-> Primary Key (Kode_Angsuran)
-> );
```

Gambar 2. Constraint Check

Hasil *constraint check* di atas dapat memberikan keuntungan atau kelebihan dikarenakan mampu meminimalis data yang sama dan bahkan data juga tidak boleh di kosongkan (Not Null). Untuk membuktikan apakah data hasil *Constraint check* di atas baik atau tidak, maka perlu di uji tingkat *optimasi query* setiap tabelnya. Setelah proses memasukkan data ke dalam tabel di database. Data tersebut dapat ditampilkan kembali seperti yang dilihat pada Tabel 2 di bawah ini.

Tabel 2. Tarik Simpan

```
mysql> select * from tarik_simpan;
```

No_Transaksi	No_Rekening	Tanggal_Transaksi	Jumlah_Tarikan	Saldo
1021	0156745061	2015-05-15	1500000	3500000
1022	0150524967	2015-05-20	1600000	6200000
1025	0151762792	2015-05-12	1250000	8500000

Menguji tabel yang dibuat dengan cara, memasukkan data yang sudah diinput di database, sehingga data yang didalam tabel tidak bisa sama atau double.

Hasil dari percobaan yang dilakukan dalam menginputkan data yang sama, maka data yang terakhir yang diinputkan tidak dapat dimasukkan oleh tabel atau biasa disebut *Duplicate entry*.

3. Implementasi Sistem dan Hasil

Implementasi sistem (*system implementation*) adalah tahap meletakkan sesuatu sistem supaya siap untuk dioperasikan.

a. Manajemen User

Manajemen user ini digunakan untuk menentukan hak akses *user*, sehingga dengan begitu data atau informasi yang berada di database dapat terjaga kerahasiannya. *User* yang

memiliki hak akses di dalam database dapat melihat semua tabel yang ada di database dan bahkan mampu melakukan perintah-perintah *query* dalam database, akan tetapi *user* yang tidak memiliki hak akses maka untuk mengakses tabel dalam database pun tidak bisa. Membuat *user* dan menentukan hak akses setiap *user* yang diijinkan untuk mengakses tabel-tabel di dalam database dan juga *user* yang tidak bisa melakukan pengaksesan terhadap tabel-tabel tertentu dapat dilihat pada Gambar 3 di bawah ini.

```
mysql> grant all privileges ON *.* to Emi
-> Identified by 'surya2404'
-> with grant Option
-> ;
Query OK, 0 rows affected (0.00 sec)

mysql> grant usage on *.* to Susila
-> Identified by 'emisuya2404'
-> ;
Query OK, 0 rows affected (0.00 sec)
```

Gambar 3. Membuat Hak Akses User

User Susila dapat login ke MySQL tetapi tidak memiliki izin akses, jadi sebagai *blank user*. Seperti yang terlihat pada Tabel 3 bahwa *user* Susila sudah dapat digunakan untuk login ke sistem database.

Tabel 3. User

```
mysql> use mysql
Database changed
mysql> select user, host, password;
```

user	host	password
root	localhost	*1F84F860CF315F884D190379395283D3C676F6A
root	linux	*80000E9963E91E1F8EC1B869F841F5AC49C8AA78
	localhost	
	linux	
pm	localhost	
Emi	localhost	*80000E9963E91E1F8EC1B869F841F5AC49C8AA78
Susila	localhost	*C08A9D67538DECF7AF311ABDAF7FE105C8B43CF95

7 rows in set (0.00 sec)

Disini *user* memiliki password terenkripsi dari hasil password yang sudah diberikan ketika membuat *user*. Hal ini dikarenakan agar tingkat keamanan dalam sistem password itu pun dapat terjaga dengan baik.

Untuk melihat hak akses *user*, berikut *user* dan hak akses yang dimiliki oleh setiap *user* ditunjukkan pada Tabel 4 sebagai berikut.

Tabel 4. Hak Akses

```
mysql> select user, reload_priv, shutdown_priv, process_priv, file_priv, grant_priv from user;
```

user	reload_priv	shutdown_priv	process_priv	file_priv	grant_priv
root	Y	Y	Y	Y	Y
root	Y	Y	Y	Y	Y
	N	N	N	N	N
	N	N	N	N	N
pm	N	N	N	N	N
Emi	Y	Y	Y	Y	Y
Susila	N	N	N	N	N

7 rows in set (0.00 sec)


```
mysql> select user, references_priv, index_priv, alter_priv from user;
+-----+-----+-----+-----+
| user | references_priv | index_priv | alter_priv |
+-----+-----+-----+-----+
| root | Y               | Y         | Y         |
| root | Y               | Y         | Y         |
|      | N               | N         | N         |
|      | N               | N         | N         |
| pma  | N               | N         | N         |
| Emi  | Y               | Y         | Y         |
| Susila | N              | N         | N         |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Penggunaan *user* "Susila" dengan menggunakan password yang sudah diberikan sebelumnya, untuk membuktikan hak akses yang diberikan oleh administrator. *User* "Susila" diijinkan untuk melihat tabel-tabel di setiap database. Tabel datanya dapat dilihat pada Gambar 4 di bawah ini.

```
mysql> use simpan_pinjam;
Database changed
mysql> show tables;
+-----+
| Tables_in_simpan_pinjam |
+-----+
| angsuran                |
| jaminan                 |
| nasabah                 |
| pinjaman                |
| rekening                |
| simpanan                |
| tarik_simpanan          |
+-----+
7 rows in set (0.00 sec)
```

Gambar 4. Database Simpan Pinjam

Akan tetapi *user* "Susila" tidak diijinkan untuk mengakses atau melihat isi dari setiap tabel, seperti yang terlihat pada Gambar 5 di bawah ini.

```
mysql> select * from angsuran;
ERROR 1142 (42000): SELECT command denied to user 'Susila'@'localhost' for table 'angsuran'
mysql> select * from jaminan;
ERROR 1142 (42000): SELECT command denied to user 'Susila'@'localhost' for table 'jaminan'
```

Gambar 5. Panggil Data

Sedangkan *user* "Emi" dapat melihat isi dari tabel-tabel yang ada di database. *User* "Emi" dapat mengakses dan menambahkan data di setiap tabel di database.

User "Emi" menambahkan satu data pada tabel angsuran. Langkahnya dapat dilihat pada Tabel 5 di bawah ini.

Tabel 5. Menambahkan Data

```
mysql> insert into angsuran values ('PAS-71','B-053','2015-06-27','Semblan','250000','125000','8510000','Belum Lunas');
Query OK, 1 row affected (0.04 sec)

mysql> select * from angsuran;
+-----+-----+-----+-----+-----+-----+-----+
| Kode_Angsuran | ID_Nasabah | Tanggal_Transaksi | Angsuran_Ke | Besar_Angsuran | Denda | Sisa_Pinjaman | Status |
+-----+-----+-----+-----+-----+-----+-----+
| PAS-13        | B-001      | 2015-05-16        | Delapan     | 1250000         | -     | 7750000        | Belum Lunas |
| PAS-15        | B-024      | 2015-06-14        | Empat       | 6500000         | 1000000 | -               | Lunas |
| PAS-32        | B-004      | 2015-06-21        | Tiga       | 5450000         | 150000 | -               | Lunas |
| PAS-71        | B-053      | 2015-06-27        | Semblan     | 2500000         | 125000 | 8510000        | Belum Lunas |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

b. Manajemen Transaksi

Menurut Indrajani dalam bukunya pada halaman 205 mengatakan. Transaksi adalah sebuah aksi atau serangkaian aksi, yang dilakukan oleh pengguna atau aplikasi yang mengakses atau mengubah isi dari basis data. Transaksi adalah sebuah unit kerja logikal dari suatu basis data. Sedangkan program aplikasi merupakan

serangkaian transaksi tanpa pengolahan basis data di dalamnya. Transaksi selalu mengubah basis data dalam satu keadaan yang konsisten ke keadaan konsisten lainnya, walaupun terjadi pelanggaran konsistensi data selama transaksi berjalan. Berikut contoh transaksi sederhana ditunjukkan pada Tabel 6 di bawah ini.

Tabel 6. Contoh Transaksi Sederhana

Transaksi A	Transaksi B
	Delete (staffNo = x)
	For all propertyForRent records, pno
Read (staffNo = x, pinjam)	Begin
Pinjam = pinjam *1,1	Read (propertyNo = pno, staffNo)
White (staffNo = x, new, pinjam)	If (staffNo =x) then
	Begin
	staffNo = newStaffno
	Write (propertyNo = pno, staffno)
	End
	End

Transaksi dapat menghasilkan salah satu outputnya sebagai berikut.

1. Sukses – transaksi dikatakan *committed* dan basis data mencapai keadaan baru atau berikutnya. Transaksi yang *committed* tidak dapat digagalkan.
2. Gagal – transaksi dikatakan *aborted* dan basis data harus dikembalikan ke keadaan sebelum dilakukannya transaksi. Transaksi seperti ini di sebut *rollback* atau *undone*. Transaksi yang digagalkan akan dapat dilakukan *rollback* yang menyebabkan mulai kembali.

4. Keamanan

Jurnal Publikasi milik dari Beny Rahmantoko (STMIK Amikom Yogyakarta) dengan judul, Sistem Informasi Penjualan Pada Toko Bangunan Bintang Besar Wates. Didalam jurnal publikasinya mengatakan tentang analisa keamanan. Pengendalian atau control dalam sebuah sistem sangat diperlukan untuk menghindari dan mendeteksi secara dini terhadap penyalahgunaan atau kesalahan sistem serta untuk menjamin keamanan data dan informasi yang dihasilkan. Analisa yang didapatkan terhadap keamanan yang dihasilkan pada database ini dapat dilihat pada Tabel 7 di bawah ini.

Tabel 7. Analisa Keamanan

Faktor	Hasil Analisa
Pengendalian Terhadap Hak Akses	Adanya pengubahan data oleh pegawai yang tidak bertanggung jawab karena tidak adanya proteksi terhadap data dan informasi.
Back Up Data	Tidak adanya Back up data dalam bentuk <i>softcopy</i> , maka bila ada data yang hilang, akan membutuhkan waktu yang lama untuk memproses nya kembali.

Jurnal Publikasi Beny Rahmantoko (STMIK Amikom Yogyakarta), Sistem Informasi Penjualan Pada Toko Bangunan Bintang Besar Wates.
 Jurnal Publikasi Vindhy Agus Setiawan (STMIK Amikom Yogyakarta), Analisis Dan Perancangan Sistem Informasi Simpan Pinjam Pada LKM Gerembeng Bali.
 Jurnal Publikasi Mardiyanti (STMIK Amikom Yogyakarta), Perancangan Pplikasi Penjualan Pada UD. Pertanian Rebo Berkah Budiono Di Kebumen.

Jurnal Publikasi milik Mardiyanti (STMIK Amikom Yogyakarta) dengan judul, Perancangan Aplikasi Penjualan Pada UD. Pertanian Rebo Berkah Budiono Di Kebumen.

Didalam jurnal publikasinya memberikan cara memback up database menggunakan *Enterprise Manager SQL Server* sebagai berikut.

1. Masuk ke *Enterprise Manager (Start -> All Program -> Microsoft SQL Server -> Enterprise Manager)*.
2. Setelah itu pilih menu database kemudian klik kanan pada database yang akan di *backup -> All Task -> Backup Database*.
3. Pada Jendela *SQL Server Backup*, pilih *Database-complete*. Pada bagian *destination*, pilih *Add* untuk menentukan lokasi dimana database tersebut akan disimpan.

Terdapat dua pilihan lokasi penyimpanan *backup* database. Menggunakan media *backup* khusus (*backup device*) atau menggunakan layanan *backup* yang dimiliki oleh *SQL Server (file name)*. Memilih lokasi penyimpanan *file name*, lalu klik OK. Setelah menentukan lokasi penyimpanan, klik OK pada *SQL Server Backup*.

5. Kesimpulan

Analisis database yang dilakukan untuk membuat data atau informasi dapat menjadi rahasia dikarenakan setiap *user* memiliki batasan hak akses. Dengan adanya hak akses tersebut pengguna yang lain tidak bisa melihat atau mengakses setiap data yang ada di database. Keamanan data yang ada di database dapat terjaga dengan baik, akan tetapi setiap data diharapkan memiliki *Back up* nya, jika terjadi kehilangan data dapat mengantisipasi hal tersebut.

6. Pustaka

Indrajani 2014, *Database System Case Study All In One*. PT. Elex Media Komputindo.Jakarta.