

Analisis Keamanan Dalam Desain Database PMB (Penerimaan Mahasiswa Baru) dengan MySQL

*Security Analysis in Database Design PMB (New Admissions)
with MySQL*

Sofyan Pariyasto*¹, Ema Utami², Suwanto Raharjo³

^{1,2}Program Pasca Sarjana Universitas Amikom Yogyakarta, 3IST AKPRIND Yogyakarta
E-mail: *sofyanpariyasto@gmail.com, ema.u@amikom.ac.id,

wa2n@akprind.ac.id

Abstrak

Keamanan dalam desain suatu database merupakan hal yang sangat penting untuk diterapkan dalam membangun sebuah sistem informasi. Sering kali dalam implementasi keamanan database tidak begitu diperhatikan, baik dari segi tipe data, panjang data, maupun parameter yang berkaitan dengan transaksi data. Kesalahan dalam desain database biasanya baru akan disadari pada saat aplikasi sudah selesai dan sudah digunakan. Untuk memastikan sebuah sistem informasi berjalan dengan lancar maka akan digunakan analisa data berdasarkan rentang data tertinggi dan data terendah. Dengan menggunakan analisa rentang data maka data yang nilainya lebih rendah dari parameter yang ditentukan akan dirubah ke nilai terendah yang sesuai dengan parameter. Begitu juga data yang nilainya lebih tinggi dari parameter yang telah ditentukan akan dirubah ke nilai tertinggi yang sesuai dengan parameter. Dari hasil pengujian yang dilakukan sebelum adanya filter check dan parameter tipe data dengan memasukan data yang berupa huruf, angka minus serta angka yang melebihi batas atas pada kolom nilai, data masih bisa tersimpan ke database. Pengujian selanjutnya dilakukan dengan memasukan angka -2, -3, -4, -5, -7, -7 serta angka diatas ambang batas atas 12,13,14,15,16,17, dari hasil pengujian yang dilakukan dapat disimpulkan bahwa sebuah database dengan fungsi trigger jauh lebih aman dibandingkan dengan database yang hanya menerapkan fungsi pembatasan berdasarkan tipe datanya saja.

Kata Kunci — Keamanan Desain Database MySQL, Analisis Keamanan Database

Abstract

Security in the design of a database is very important to be applied in building an information system. Often in the implementation of a database security is not given much attention, both in terms of data types, length, and parameters relating to data transactions. Errors in database design are usually only noticed when the application is complete and has been used. To ensure an information system runs well, data analysis will be used based on the highest and lowest data ranges. By using data range analysis, data whose value is lower than the specified parameter will be changed to the lowest value in accordance with the parameters. Data whose value is higher than predetermined parameters will be changed to the highest value in accordance with parameters. Results of tests conducted before the filter check and data type parameter by entering data in the form of letters, minus numbers and numbers that exceed the upper limit in the value column, data can still be saved to database. Further testing is done by entering the numbers -2, -3, -4, -5, -7, -7 and numbers above the upper threshold of 12,13,14,15,16,17, from some of the results of tests conducted can be concluded below a database with a trigger function is more secure than a database that only applies restrictions based on its data type.

Keywords — Security Database Design MySQL, Security Analysis Database

1. PENDAHULUAN

Database merupakan kumpulan data yang di kelompokkan kedalam masing-masing kelompok yang saling berkaitan antara satu dengan yang lainnya. Database secara garis besar dibedakan menjadi dua bagaian, yaitu database flat dan database relasional. Database flat adalah sebuah database dimana masing-masing data didalamnya berdiri sendiri dan tidak saling berelasi, sehingga setiap transaksi yang dilakukan dalam sebuah tabel baik itu pengisian data, penghapusan maupun perubahan data tidak akan berpengaruh ke tabel yang lain. Sedangkan database relasional adalah database yang saling berelasi satu dengan yang lainnya, sehingga jika ada transaksi data dalam sebuah tabel akan berpengaruh ke tabel yang terkait [1].

Suatu Database Management System (DBMS) merupakan sekumpulan data yang dapat dimanipulasi baik itu penambahan, penghapusan maupun perubahan data, serta sebuah perangkat lunak yang berfungsi sebagai antara muka pengguna untuk melakukan kegiatan transaksi data dimana database dalam sebuah DBMS harus saling berelasi satu dengan yang lainnya. Jadi dapat disimpulkan bahwa database merupakan sekumpulan file-file ataupun data yang saling berelasi satu dengan yang lainnya, dimana untuk menghubungkan sebuah data dengan data lainnya dalam relasi tersebut dibutuhkan data penghubung (key) [2].

Salah satu peran database yaitu harus mampu memenuhi kebutuhan pengguna dalam memberikan berbagai informasi dan data terkait sistem informasi yang digunakan Database harus mampu memenuhi kebutuhan tersebut baik saat ini maupun dimasa mendatang. Sebuah database juga harus memiliki kemampuan dalam proses pembuatan pencadangan (Backup) karena data harus tetap dijaga sehingga dapat diakses kapanpun [3].

Dalam sebuah sistem informasi tidak terlepas dari keberadaan sebuah database. Dimana database ini memiliki fungsi untuk menampung data-data yang berkaitan dengan sebuah sistem informasi. Seiring dengan berjalannya sebuah sistem informasi tentu pertambahan jumlah data yang tersimpan dalam database juga akan semakin meningkat, hal ini tentu menjadi perhatian penting dimana dengan jumlah data yang terus meningkat dan bersifat rahasia tersebut maka dibutuhkan suatu pengamanan khusus pada database, baik Backup, antisipasi SQL Injection maupun penanganan khusus terhadap transaksi antar tabel baik itu insert, update dan delete. Pengamanan khusus juga dilakukan terhadap transaksi antar tabel yang melibatkan tabel yang saling berelasi dimana hal ini membutuhkan ketelitian dan kehati-hatian karena data saling terkait antara satu tabel dengan tabel yang lainnya.

Untuk meminimalisir terjadinya permasalahan seperti yang telah dijelaskan sebelumnya maka akan diuraikan tahapan-tahapan apa yang harus dilakukan pada database yang digunakan dalam Sistem informasi PMB. Juga akan dijelaskan hal-hal apa saja yang menjadi dasar adanya perbaikan maupun penambahan filter tertentu pada database yang digunakan pada Sistem Infomasi PMB.

1.1. Tinjauan Pustaka

Pemberian Constraint Check dalam basis data akan sangat membantu sebuah database dalam melakukan penyaringan data dalam sebuah transaksi. Dengan adanya Constraint Check hanya data yang memenuhi syaratlah yang akan di proses transaksinya baik itu penambahan, pengubahan maupun dalam penghapusan data. Sehingga dengan adanya Constraint Check ini dapat menyederhanakan program antara muka yang nantinya akan dibuat [4].

Keamanan dalam sebuah database merupakan bagian penting dari desain dan fungsi dari basis data terdistribusi. Dalam sebuah basis data terdistribusi ada tiga bagian penting yaitu pysical, User dan Network. Ketiga bagian tersebut harus saling bekerja sama dengan Policies, Standar dan procedure untuk dapat menghasilkan suatu basis data terdistribusi yang optimal. Khususnya paada bagian pysical database harus bekerja sama dengan Standar dan Procedure tambahan untuk meningkatkan kemananan dalam sebuah database [5].

Salah satu cara untuk mengamankan sebuah database adalah dengan melakukan enkripsi pada sebuah database dengan menggunakan metode penghitungan hash dan watermark. Dengan menggunakan metode ini diharapkan tidak terjadi perubahan data dan duplikasi data [6].

Kesalahan konfigurasi dan parameter dalam sebuah database telah menjadi penyebab utama kegagalan perangkat lunak. Kesalahan konfigurasi akan berakibat pada tidak sesuainya data yang masuk dalam sebuah database [7].

Dari beberapa penelitian terdahulu telah dijelaskan metode yang digunakan dalam mengamankan sebuah database dan betapa pentingnya pengamanan sebuah database untuk memastikan sebuah sistem informasi dapat berjalan dengan lancar. Pada penelitian disini difokuskan pada penentuan tipe data, panjang karakter serta penggunaan Trigger dalam validasi transaksi data.

2. METODE PENELITIAN

Analisa dalam keamanan dalam desain database penerimaan mahasiswa baru dalam penelitian ini akan menggunakan metode yang menjadi standard MySQL yaitu Constraint, Function serta pengguna Tipe Data. Pada penerapannya nanti fungsi yang dibuat tersebut akan diujicoba dengan transaksi yang berkaitan dengan tabel-tabel yang akan diuji dengan melakukan pengisian data, manipulasi data bahkan hingga penghapusan data. Data yang diujicoba adalah data yang tidak sesuai dengan aturan yang telah diterapkan pada sistem informasi PMB sehingga diketahui seberapa efektifkah penerapan Constraint dan Function yang dibuat untuk memberikan keamanan pada Database yang digunakan.

2.1. Batasan Masalah

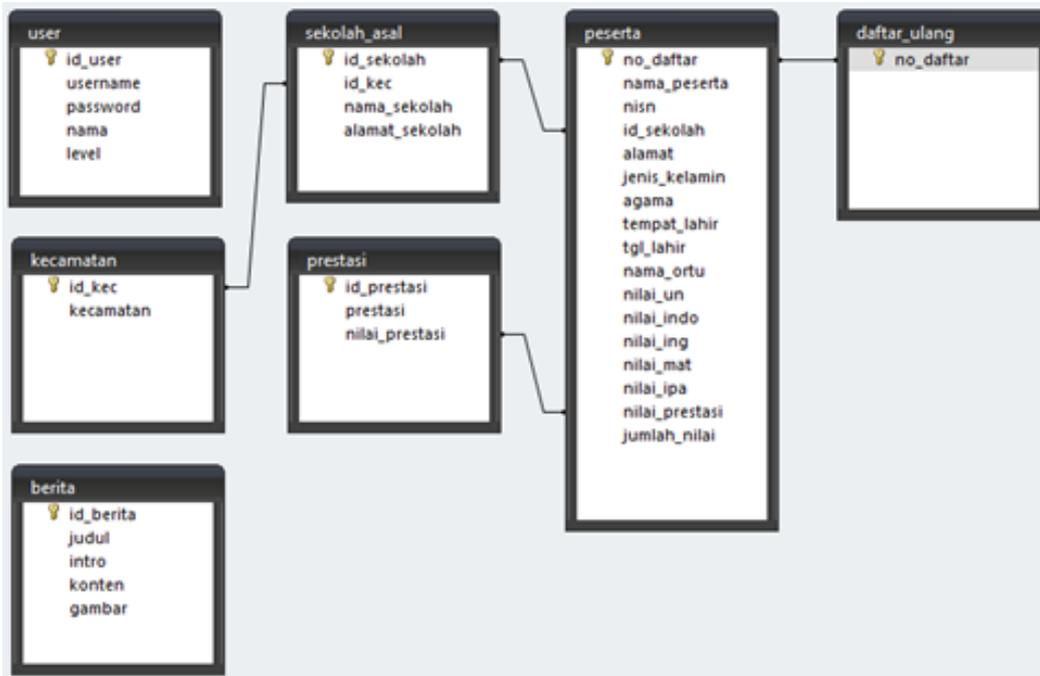
Agar pembahasan dalam penelitian ini tidak terlalu melebar maka penelitian akan difokuskan pada analisis berdasarkan kesalahan penentuan tipe data, analisis berdasarkan tipe data dalam transaksi tabel, analisis berdasarkan filter terhadap transaksi tabel dan analisis berdasarkan data yang salah sebelum adanya filter. Sehingga dengan adanya batasan ini dapat menjadi acuan dan menjadi salah ladsan dalam penelitian ini agar tetap berada pada titik permasalahan.

2.2. Perancangan Desain database PMB (Penerimaan Mahasiswa Baru)

Dalam pelaksanaan penelitian ini mengambil skripsi dari Irwin Nugroho dengan NIM 07520244075, program studi Pendidikan Teknik Informatika Fakultas Teknik, dari Universitas Negeri Yogyakarta. Ruang lingkup database yang akan dijabarkan adalah berkaitan dengan penerimaan mahasiswa baru (PMB). Dalam sistem informasi yang dibangun pada skripsi tersebut terdapat tujuh tabel yaitu sebagai berikut [8]:

1. Tabel User
 2. Tabel Kecamatan
 3. Tabel Berita
 4. Tabel Sekolah Asal
 5. Tabel Prestasi
 6. Tabel Peserta
 7. Tabel Daftar Ulang
-

Ketujuh tabel tersebut kemudian akan saling berelasi satu dengan yang lainnya, adapun desain relasi antar tabel dari aplikasi PMB tersebut terlihat seperti Gambar 1 berikut [8].



Gambar 1. Relasi Antar Tabel Database PMB

Adapun Tabel user memiliki property seperti terlihat pada Tabel 1. Pada tabel user terdapat 5 field yaitu id_user, username, password dan level, dalam tabel ini berisi data user yang berhak login kedalam aplikasi PMB.

Tabel 1. Tabel User pada database PMB

Field	Type	Size	Null
id_user (*)	varchar	5	not null
username	varchar	45	not null
password	varchar	45	not null
nama	varchar	45	not null
level	varchar	45	not null

Selanjutnya terdapat tabel kecamatan yang akan menampung data kecamatan dan kode kecamatan, adapun tabelnya terlihat seperti pada Tabel 2. Tabel kecamatan terdiri dari 2 field yaitu id_kec, dan kecamatan dimana seluruh data kecamatan yang digunakan pada PMB.

Tabel 2. Tabel Kecamatan pada database PMB

Field	Type	Size	Null
id_kec (*)	varchar	2	not null
kecamatan	varchar	45	not null

Tabel sekolah asal menampung data yang berkaitan dengan asal sekolah peserta PMB, dimana dalam tabel ini terdapat field yang saling berelasi dengan tabel kecamatan. Terdapat empat field dalam tabel ksekolah yaitu id sekolah, id kecamatan, nama sekolah dan alamat sekolah, seperti ditunjukkan pada Tabel 3.

Tabel 3. Tabel Sekolah asal pada database PMB

Field	Type	Size	Null
id_sekolah (*)	integer	3	not null
id_kec (**)	integer	2	not null
nama_sekolah	varchar	25	not null
alamat_sekolah	varchar	45	not null

Tabel peserta, ditunjukkan seperti Tabel 4, merupakan tabel utama dalam database PMB (Penerimaan Mahasiswa Baru) dimana dalam tabel ini terdapat filed-filed yang saling berelasi dengan filed dari tabel pendukung lainnya. Seluruh data peserta PMB nantinya akan ditampung dalam tabel ini sehingga bisa dikatakan tabel ini adalah tabel inti dari aplikasi PMB (Penerimaan Mahasiswa Baru).

Tabel 4. Tabel Peserta pada database PMB

Field	Type	Size	Null
no_daftar (*)	integer	4	not null
nama_peserta	varchar	45	not null
id_sekolah (**)	integer	3	not null
alamat	varchar	45	not null
tempat_lahir	varchar	15	not null
tanggal_lahir	date		not null
agama	varchar	10	not null
nama_ayah	varchar	45	not null
nilai_un	varchar	5	not null
nilai_indo	varchar	5	not null
nilai_ing	varchar	5	not null
nilai_mat	varchar	5	not null
nilai_ipa	varchar	5	not null
nilai_prestasi (**)	varchar	5	not null
jumlah_nilai	varchar	5	not null

Tabel prestasi nantinya akan digunakan untuk menampung data prestasi peserta PMB, dimana terdapat tiga field dalam tabel ini yaitu, id prestaso, prestasi, dan nilai prestasi, ditunjukkan pada Tabel 5.

Tabel 5. Tabel Prestasi pada database PMB

Field	Type	Size	Null
id_prestasi (*)	varchar	6	not null
prestasi	varchar	45	not null
nilai_prestasi	varchar	5	not null

Tabel Daftar ulang hanya berisi satu field yaitu filed no daftar, yang mana no daftar sebagai primary key sekaligus sebagai foreign key, ditunjukkan pada Tabel 6.

Tabel 6. Tabel Daftar Ulang pada database PMB

Field	Type	Size	Null
no_daftar (*)(**)	integer	4	not null

Tabel berita menampung informasi serta pengumuman pada aplikasi PMB, ditunjukkan pada Tabel 7. Tabel data pada database PMB ini tidak berelasi dengan tabel manapun.

Tabel 7. Tabel Berita pada database PMB

Field	Type	Size	Null
id_berita (*)	integer	4	not null
judul	text		not null
intro	text		not null
konten	text		not null
gambar	text		not null

Adapun Code SQL dalam pembuatan database PMB adalah sebagai berikut.

```
1. CREATE DATABASE IF NOT EXISTS `pmb`
2. USE `pmb`;
3.
4. CREATE TABLE IF NOT EXISTS `berita` (
5.   `id_berita` int(4) NOT NULL,
6.   `judul` text NOT NULL,
7.   `intro` text NOT NULL,
8.   `konten` text NOT NULL,
9.   `gambar` text NOT NULL,
10.  PRIMARY KEY (`id_berita`)
11. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
12.
13. CREATE TABLE IF NOT EXISTS `daftar_ulang` (
14.   `no_daftar` int(4) NOT NULL,
15.  PRIMARY KEY (`no_daftar`)
16. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
17.
18. CREATE TABLE IF NOT EXISTS `kecamatan` (
19.   `id_kec` varchar(2) NOT NULL,
20.   `kecamatan` varchar(45) NOT NULL,
21.  PRIMARY KEY (`id_kec`)
22. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
23.
24. CREATE TABLE IF NOT EXISTS `peserta` (
25.   `no_daftar` int(4) NOT NULL AUTO_INCREMENT,
26.   `nama_peserta` varchar(45) NOT NULL,
27.   `id_sekolah` int(3) NOT NULL,
28.   `alamat` varchar(45) NOT NULL,
29.   `tempat_lahir` varchar(15) NOT NULL,
30.   `tanggal_lahir` date NOT NULL,
31.   `agama` varchar(10) NOT NULL,
32.   `nama_ayah` varchar(45) NOT NULL,
33.   `nilai_un` varchar(5) NOT NULL,
34.   `nilai_indo` varchar(5) NOT NULL,
35.   `nilai_ing` varchar(5) NOT NULL,
36.   `nilai_mat` varchar(5) NOT NULL,
37.   `nilai_ipa` varchar(5) NOT NULL,
38.   `nilai_prestasi` varchar(5) NOT NULL,
39.   `jumlah_nilai` varchar(5) NOT NULL,
40.  PRIMARY KEY (`no_daftar`)
41. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
42.
43. CREATE TABLE IF NOT EXISTS `prestasi` (
44.   `id_prestasi` varchar(6) NOT NULL,
45.   `prestasi` varchar(45) NOT NULL,
46.   `nilai_prestasi` varchar(5) NOT NULL,
47.  PRIMARY KEY (`id_prestasi`)
48. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
49.
50. CREATE TABLE IF NOT EXISTS `sekolah_asal` (
51.   `id_sekolah` int(11) NOT NULL,
52.   `id_kec` int(11) DEFAULT NULL,
53.   `nama_sekolah` varchar(25) DEFAULT NULL,
```

```

54. `alamat_sekolah` varchar(45) DEFAULT NULL,
55. PRIMARY KEY (`id_sekolah`),
56. KEY `id_kec` (`id_kec`)
57. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
58.
59. CREATE TABLE IF NOT EXISTS `user` (
60. `id_user` int(5) NOT NULL,
61. `username` varchar(45) NOT NULL,
62. `password` varchar(45) NOT NULL,
63. `nama` varchar(45) NOT NULL,
64. `level` varchar(45) NOT NULL,
65. PRIMARY KEY (`id_user`)
66. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

3. HASIL DAN PEMBAHASAN

3.1. Analisis Keamanan Database PMB

3.1.1. Analisis Berdasarkan Kesalahan Penentuan Tipe Data

Pembatasan Transaksi data dengan basis data disamping menjamin keakuratan data juga dapat meningkatkan keamanan data. Sebuah sistem informasi yang memiliki basis data yang baik akan menjadi pondasi yang kuat bagi pengembangan sistem informasi tersebut [9]. Dari Tabel 1 tersebut ada beberapa point yang menjadi sorotan antara lain. Untuk username panjang karakternya harusnya tidak sampai 45 karakter, karena pada umumnya penggunaan username dibawah 20 karakter. Untuk field nama panjangnya dapat di set ke 30 karakter. Untuk field level bisa menggunakan integer dengan panjang 1 karakter jika level menggunakan angka, atau bisa menggunakan tipe data enum jika level sudah ditentukan sebelumnya contoh (admin, pegawai).

Dari tabel asal sekolah (tabel 3) ada beberapa point yang menjadi sorotan diantaranya, id_sekolah seharusnya bisa menggunakan NPSN (nomor pokok sekolah nasional). Adapun NPSN (nomor pokok sekolah nasional) dapat diakses di alamat <http://referensi.data.kemdikbud.go.id/index11.php>. Selain penggunaan kode NPSN panjang karakter untuk id_sekolah juga harus dibatasi sesuai NPSN sepanjang 8 karakter. Tampilan NPSN ditunjukkan seperti Gambar 2.

DATA REFERENSI
KEMENTERIAN PENDIDIKAN & KEBUDAYAAN

Profil Satuan Pendidikan / Lembaga

Identitas Satuan Pendidikan / Lembaga | Dokumen dan Perjinan | Sarana Prasarana | Yayasan | Kontak | Gallery Foto | Map | Histori Data

Nama : SD ISLAM TERPADU MADANI
NPSN : 69831530
Alamat : CANDRA KENCANA RK 1
Kode Pos :
Desa/Kelurahan : CANDRA KENCANA
Kecamatan/Kota (LN) : Tulang Bawang Tengah
Kab.-Kota/Negeri (LN) : Kab. Tulang Bawang Barat
Propinsi/Luar Negeri (LN) : Lampung
Status Sekolah : SWASTA
Waktu Penyelenggaraan :
Jenjang Pendidikan : SD

Link Terkait Master Referensi
1. Peserta Didik
2. Pendidik dan Tenaga Kependidikan

HelpDesk NPSN
• Telp : 021-57905184
• Facebook : HelpDesk NPSN
• Email : npsn_pdsp@yahoo.co.id

HelpDesk NISN
• Telp : 021-57905777/57904804
• Email : pdsp@kemdikbud.go.id

HelpDesk KEMENAG
• Telp : 021-34833235

Pusat Data Dan Statistik Pendidikan (PDS) Kementerian Pendidikan dan Kebudayaan Gedung E, Lantai 1, Kompleks Kemdikbud Jalan Jenderal Sudirman, Senayan, Jakarta 10270

Gambar 2. Tampilan NPSN dari Website Kemendikbud

Kemudian untuk field alamat sekolah harusnya menggunakan tipe data text, hal ini dilakukan untuk mengantisipasi jika alamat sekolah cukup panjang dan lebih dari 45 karakter. Sedangkan untuk id_kecamatan bisa langsung menggunakan kode kecamatan dari kemendagri

yang dapat diakses pada alamat berikut <http://www.kemendagri.go.id/pages/data-wilayah>, adapun panjang kode kecamatan nasional berdasarkan data dari kemendagri adalah 13 karakter, seperti ditunjukkan pada Tabel 8.

Tabel 8. Tampilan data kecamatan Nasional Kemendagri

Kode	Nama Provinsi/ Kabupaten/ Kota	Jumlah		Nama/jumlah			Luas Wilayah (Km ²)	Jumlah Penduduk (Jiwa)
		Kab	Kota	Kec	Kel	Desa		
							586,28	445.655
34	D.I. Yogyakarta			12	1	87		
34.01	Kab Kulon Progo			Temon	-	15		
34.01.01.2001						1 Jangkar		
34.01.01.2002						2 Sindutan		
34.01.01.2003						3 Palihan		
34.01.01.2004						4 Glagah		
34.01.01.2005						5 Kalidengen		
34.01.01.2006						6 Plumbon		
34.01.01.2007						7 Kedundang		
34.01.01.2008						8 Demen		
34.01.01.2009						9 Kulur		
34.01.01.2010						10 Kaligintung		
34.01.01.2011						11 Temon Wetan		
34.01.01.2012						12 Temon Kulon		
34.01.01.2013						12 Kebon Rejo		
34.01.01.2014						14 janten		
34.01.01.2015						15 karangwuluh		

Pembahasan selanjutnya adalah mengenai tabel Daftar ulang dimana pada tabel ini hanya terdapat satu field, hal ini tentu menjadi rancu karena jika hanya menggunakan satu field, seharusnya tabel daftar ulang bisa digabung ke tabel pendaftaran peserta pmb.

3.1.2. Analisis Berdasarkan Tipe Data Dalam Transaksi Tabel

Membatasi input data agar data yang datang sesuai dengan persyaratan dan pengaturan yang ditetapkan adalah salah satu cara untuk menjaga integritas basis data, karena basis data merupakan sumber informasi utama yang digunakan oleh sebuah sistem informasi [10]. Kesalahan penentuan tipe data pada database akan berakibat fatal pada sistem informasi yang dibangun. Kesalahan penentuan tipe data juga berakibat data yang masuk tidak sesuai dengan kebutuhan data yang diharapkan, terlebih jika tidak diterapkan filter pada aplikasi yang ada. Untuk meminimalisir kesalahan serta ketidaksesuaian data yang masuk ke dalam sebuah database maka harus dilakukan perencanaan yang cukup matang dalam penentuan tipe data dalam database. Sebagai contoh kesalahan penentuan tipe data dalam desain database PMB bisa dilihat pada tabel peserta (tabel 4), pada field nilai_un hingga jumlah_nilai terlihat tipe data yang digunakan adalah varchar dan panjang karakter sebanyak 5 karakter seperti pada Tabel 9.

Tabel 9. Tampilan tipe data field Nilai

Nilai_un	Varchar	5	Not null
Nilai_indo	Varchar	5	Not null
Nilai_ing	Varchar	5	Not null
Nilai_ipa	Varchar	5	Not null
Nilai_prestasi (**)	Varchar	5	Not null
Jumlah_nilai	Varchar	5	Not null

Hal ini berakibat pada saat terjadi transaksi pada tabel peserta baik input data maupun update data, ada data yang tidak sesuai dengan kebutuhan yang tersimpan ke dalam data peserta khususnya pada field nilai.

3.1.2.1. Proses pengujian sebelum ditambahkan filter

Proses pengujian dilakukan dengan memasukan nilai_un dengan huruf hal ini dimaksudkan untuk melihat hal yang terjadi jika terjadi kesalahan penginputan dalam penggunaan aplikasi. Adapun perintah yang digunakan adalah sebagai berikut.

```
1. INSERT INTO `peserta`
2. (`no_daftar`, `nama_pesertca`, `id_sekolah`, `alamat`, `tempat_lahir`, `tanggal_lahir`,
   `agama`, `nilai_un`, `nilai_indo`, `nilai_ing`, `nilai_mat`, `nilai_ipa`, `nilai_prestasi`,
   `jumlah_nilai`)
3. VALUES
4. ('2', 'Budi', '1', 'Jalan Listrik, Medan', 'Sergei', '2018-01-01', 'Apa Saja', 'AAA', 'BBB', 'CCC', 'DDD', 'EEE', 'FFF', 'GGG');
```

Dikarenakan adanya kesalahan dalam penentuan tipe data maka, transaksi di atas akan dijalankan begitu saja oleh MySQL, dan data akan disimpan ke dalam database, meskipun data yang dimasukan tidak sesuai dengan kebutuhan, ditunjukkan pada Gambar 3.



```
1 select `no_daftar`, `nama_pesertca`, `id_sekolah`, `alamat`, `tempat_lahir`, `tanggal_lahir`, `agama`, `nilai_un`, `nilai_indo`, `nilai_ing`, `nilai_mat`,
2 `nilai_ipa`, `nilai_prestasi`, `jumlah_nilai`
3 from `peserta` order by `no_daftar` desc limit 1
```

no_daftar	nama_pesertca	id_sekolah	alamat	tempat_lahir	tanggal_lahir	agama	nilai_un	nilai_indo	nilai_ing	nilai_mat	nilai_ipa	nilai_prestasi	jumlah_nilai
2	Budi	1	Jalan Listrik, Medan	Sergei	2018-01-01	Apa Saja	AAA	BBB	CCC	DDD	EEE	FFF	GGG

Gambar 3. Tampilan Data yang tidak sesuai berhasil dimasukan ke database

Solusi untuk mengatasi adalah dengan mengubah tipe data dari Varchar menjadi Float, hal ini dilakukan untuk meminimalisir ketidak sesuaian data yang di masukan ke dalam sistem. Adapun syntax untuk mengganti tipe datanya adalah sebagai berikut.

```
1. ALTER TABLE `peserta`
2. CHANGE COLUMN `nilai_un` `nilai_un` FLOAT NOT NULL AFTER `nama_ayah`,
3. CHANGE COLUMN `nilai_indo` `nilai_indo` FLOAT NOT NULL AFTER `nilai_un`,
4. CHANGE COLUMN `nilai_ing` `nilai_ing` FLOAT NOT NULL AFTER `nilai_indo`,
5. CHANGE COLUMN `nilai_mat` `nilai_mat` FLOAT NOT NULL AFTER `nilai_ing`,
6. CHANGE COLUMN `nilai_ipa` `nilai_ipa` FLOAT NOT NULL AFTER `nilai_mat`,
7. CHANGE COLUMN `jumlah_nilai` `jumlah_nilai` FLOAT NOT NULL
```

3.1.2.2. Proses pengujian sesudah ditambahkan filter

Setelah dilakukan penggantian tipe data selanjutnya akan dilakukan percobaan apakah masih bisa memasukan data yang tidak sesuai dengan aturan, berikut syntaxnya.

```
1. INSERT INTO `peserta`
2. (`no_daftar`, `nama_pesertca`, `id_sekolah`, `alamat`, `tempat_lahir`, `tanggal_lahir`,
   `agama`, `nilai_un`, `nilai_indo`, `nilai_ing`, `nilai_mat`, `nilai_ipa`, `nilai_prestasi`,
   `jumlah_nilai`)
3. VALUES
4. ('3', 'Badu', '1', 'Jalan Listrik, Medan', 'Sergei', '2018-01-01', 'Apa Saja', 'AAA', 'BBB', 'CCC', 'DDD', 'EEE', 'FFF', 'GGG');
```

Adapun tampilan data yang dicoba dimasukan ke dalam database terlihat seperti Gambar 4 berikut. Meskipun kita memasukan nilai AAA, BBB dan karakter lainnya maka database secara otomatis menggantinya menjadi angka 0 (nol).

```
select * from peserta
```

no_daftar	nama_peserta	id_sekolah	alamat	tempat_lahir	tanggal_lahir	agama	nama_ayah	nilai_un	nilai_indo	nilai_ing	nilai_mat	nilai_ipa
2	Budi	1	Jalan Listrik, Medan	Sergei	2018-01-01	Apa Saja		0	0	0	0	0
3	Badu	1	Jalan Listrik, Medan	Sergei	2018-01-01	Apa Saja		0	0	0	0	0

Gambar 4. Tampilan Data yang tidak sesuai berhasil dimasukan ke database

3.1.3. Analisis Berdasarkan filter Terhadap Transaksi Tabel

Pada penerapannya pemilihan tipe data yang baik saja masih belum cukup untuk mengatasi ketidak sesuaian data. Tentu dengan pemilihan tipe data yang baik kesalahan dalam transaksi data akan dapat di minimalisir. Banyak kasus dimana terkadang dalam transaksi data masih banyak data yang masih belum sesuai meskipun sudah diterapkan filter yang berupa penentuan tipe data yang baik.

Pemberian pembatasan data yang diinputkan dapat mencegah terjadinya exploitasi antar muka program ke suatu basis data. Kesalahan dalam pembuatan program antar muka yang melakukan komunikasi ke server basis data dapat menyebabkan munculnya kerawanan keamanan terhadap data yang disimpan, salah satu bentuk kerawanan tersebut adalah serangan SQL Injection [11].

3.1.3.1. Proses pengujian sebelum ditambahkan filter

Tahap selanjutnya akan dilakukan pengujian dengan memasukan angka yang tidak sesuai dengan standar nilai. Jika disini standar nilai yang digunakan adalah 0 sampai dengan 10. Data yang akan dimasukan yaitu data yang bernilai minus dan data dengan nilai lebih besar dari 10.

```
1. INSERT INTO `peserta`
2. (`no_daftar`, `nama_peserta`, `id_sekolah`, `alamat`, `tempat_lahir`, `tanggal_lahir`,
   `agama`, `nilai_un`, `nilai_indo`, `nilai_ing`, `nilai_mat`, `nilai_ipa`, `nilai_prestasi`, `jumlah_nilai`)
3. VALUES
4. ('4', 'Toni', '1', 'Jalan Listrik, Medan', 'Sergei', '2018-01-01', 'Apa Saja', '-1', '-2',
   '-3', '-4', '-5', '-6', '-7'),
5. ('5', 'Tono', '1', 'Jalan Listrik, Medan', 'Sergei', '2018-01-01', 'Apa Saja', '11', '12', '13', '14', '15', '16', '17')
6. ;
```

Dari syntax di atas data tersebut berhasil masuk kedalam database meskipun bernilai minus dan bernilai lebih besar dari 10, seperti yang terlihat pada Gambar 5.

```
select * from peserta
```

no_daftar	nama_peserta	id_sekolah	alamat	tempat_lahir	tanggal_lahir	agama	nama_ayah	nilai_un	nilai_indo	nilai_ing	nilai_mat	nilai_ipa
2	Budi	1	Jalan Listrik, Medan	Sergei	2018-01-01	Apa Saja		0	0	0	0	0
3	Badu	1	Jalan Listrik, Medan	Sergei	2018-01-01	Apa Saja		0	0	0	0	0
4	Toni	1	Jalan Listrik, Medan	Sergei	2018-01-01	Apa Saja		-1	-2	-3	-4	-5
5	Tono	1	Jalan Listrik, Medan	Sergei	2018-01-01	Apa Saja		11	12	13	14	15

Gambar 5. Tampilan Data minus dan lebih besar dari sepuluh

Untuk mengatasi hal tersebut akan dibuat sebuah constraint trigger untuk mengecek bahwa nilai yang dimasukkan harus lebih kecil sama dengan sepuluh dan lebih besar sama dengan nol, serta nilai tidak boleh kosong dan tidak boleh null. Berikut syntaxnya.

```
1. DELIMITER $$
2. CREATE TRIGGER `cek_insert_peserta` BEFORE INSERT ON `peserta` FOR EACH ROW
3. begin
4.     declare pesan_erro varchar(100);
5.     if ((new.nilai_un is null) or (new.nilai_un='') or (new.nilai_un < 0)
6.     or (new.nilai_un > 10) ) then
7.         set pesan_erro = 'nilai tidak boleh minus atau lebih besar dari 10
8.     ';
9.         signal sqlstate '45000' set message_text = pesan_erro;
10.    elseif ((new.nilai_indo is null) or (new.nilai_indo='') or (new.nilai_indo < 0) or (new.nilai_indo > 10) ) then
11.        set pesan_erro = 'nilai tidak boleh minus atau lebih besar dari 10
12.    ';
13.        signal sqlstate '45000' set message_text = pesan_erro;
14.    elseif ((new.nilai_ing is null) or (new.nilai_ing='') or (new.nilai_ing < 0) or (new.nilai_ing > 10) ) then
15.        set pesan_erro = 'nilai tidak boleh minus atau lebih besar dari 10
16.    ';
17.        signal sqlstate '45000' set message_text = pesan_erro;
18.    elseif ((new.nilai_mat is null) or (new.nilai_mat='') or (new.nilai_mat < 0) or (new.nilai_mat > 10) ) then
19.        set pesan_erro = 'nilai tidak boleh minus atau lebih besar dari 10
20.    ';
21.        signal sqlstate '45000' set message_text = pesan_erro;
22.    elseif ((new.jumlah_nilai is null) or (new.jumlah_nilai='') or (new.jumlah_nilai < 0) or (new.jumlah_nilai > 10) ) then
23.        set pesan_erro = 'nilai tidak boleh minus atau lebih besar dari 10
24.    ';
25.        signal sqlstate '45000' set message_text = pesan_erro;
26.    end if;
27. end$$
28. DELIMITER ;
```

3.1.3.2. Proses pengujian sesudah ditambahkan filter

Trigger tersebut berfungsi untuk memastikan bahwa nilai yang dimasukkan bernilai antara 0 (nol) hingga 10 (sepuluh). Adapaun akan dilakukan uji coba dengan memasukan nilai minus dan lebih besar dari 10, dengan syntax sebagai berikut.

```
1. INSERT INTO `peserta`
2. (`no_daftar`, `nama_pesertca`, `id_sekolah`, `alamat`, `tempat_lahir`, `tanggal_lahir`,
3. `agama`, `nilai_un`, `nilai_indo`, `nilai_ing`, `nilai_mat`, `nilai_ipa`, `nilai_prestasi`, `jumlah_nilai`)
4. VALUES
5. ('6', 'santi', '1', 'Jalan Listrik, Medan', 'Sergei', '2018-01-01', 'Apa Saja', '1', '-2', '-3', '-4', '-5', '-6', '-7'),
6. ('7', 'maya', '1', 'Jalan Listrik, Medan', 'Sergei', '2018-01-01', 'Apa Saja', '11', '12', '13', '14', '15', '16', '17')
7. ;
```

Setelah dijanjikan maka akan muncul pesan notifikasi yang tidak memperbolehkan data untuk di simpan di dalam database, seperti terlihat pada Gambar 6 berikut.



Gambar 6. Tampilan Pesan Error Filter MySQL

Untuk memastikan apakah data benar-benar masuk atau tidak, dapat dilihat tampilan tabel peserta, seperti terlihat pada Gambar 7.

```
select * from peserta
```

no_daftar	nama_peserta	id_sekolah	alamat	tempat_lahir	tanggal_lahir	agama	nama_ayah	nilai_un	nilai_indo	nilai_ing	nilai_mat	nilai_ipa	nilai_prestasi
2	Budi	1	Jalan Listrik, Medan	Sergei	2018-01-01	Apa Saja		0	0	0			
3	Badu	1	Jalan Listrik, Medan	Sergei	2018-01-01	Apa Saja		0	0	0			
4	Toni	1	Jalan Listrik, Medan	Sergei	2018-01-01	Apa Saja		-1	-2	-3			
5	Tono	1	Jalan Listrik, Medan	Sergei	2018-01-01	Apa Saja		11	12	13			

Gambar 7. Tampilan Data Tidak berubah, karena data gagal disimpan

Penerapan Trigger Check tidak hanya bisa diletakan pada saat proses input data namun juga bisa digunakan untuk melakukan pengecekan pada saat terjadi perubahan data.

3.1.4. Analisa Berdasarkan Data yang salah sebelum Adanya Filter

Sebuah database yang baik tidak terlepas dari pemahaman dan metode yang digunakan dalam merancangnya terutama bagaimana melakukan setiap langkah dalam fase DBLC (Data Base Life Cycle) [12]. Untuk menghilangkan ambiguitas data dalam sebuah database tidak terlepas dari proses normalisasi database yang dilakukan dengan perubahan data sesuai dengan aturan yang berlaku [13].

Sejauh ini penerapan filter yang dilakukan pada saat transaksi data dalam tabel sudah berjalan cukup baik, namun terdapat sebuah masalah yaitu data sudah terlanjur diisi kedalam tabel sebelum diterapkan filter masih belum sesuai dengan data yang diharapkan. Salah satu contoh data yang masih belum sesuai terlihat seperti Gambar 9. Pada baris ke tiga ada data yang masih bernilai minus dan ada data yang bernilai lebih dari 10. Untuk mengatasi hal ini maka perlu dibuat sebuah perintah untuk memperbaiki data dengan catatan data yang bernilai minus di rubah menjadi nol sedangkan data yang lebih dari sepuluh diubah menjadi sepuluh. Dengan kata lain data tidak boleh melewati ambang batas tertinggi dan batas terendah.

```
1. update peserta set nilai_un=case when nilai_un < 0 then 0 when nilai_un > 10 then 10 else nilai_un end ,
2. nilai_indo=case when nilai_indo < 0 then 0 when nilai_indo > 10 then 10 else nilai_indo end,
3. nilai_ing=case when nilai_ing < 0 then 0 when nilai_ing > 10 then 10 else nilai_ing end,
4. nilai_mat=case when nilai_mat < 0 then 0 when nilai_mat > 10 then 10 else nilai_mat end,
5. nilai_ipa=case when nilai_ipa < 0 then 0 when nilai_ipa > 10 then 10 else nilai_ipa end,
6. nilai_prestasi=case when nilai_prestasi < 0 then 0 when nilai_prestasi > 10 then 10 else nilai_prestasi end
```

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan maka dapat ditarik kesimpulan sebagai berikut.

1. Dari hasil pengujian yang dilakukan sebelum ditambahkan parameter tipe data dan check data yang berupa huruf dan karakter masih bisa dimasukkan kedalam database. Tidak hanya sampai disitu angka minus dan lebih dari 10 juga masih bisa dimasukkan kedalam database. Namun setelah ditambahkan filter hanya angka dengan range 1-10 yang bisa dimasukkan kedalam database.
2. Dari hasil pengujian juga dilakukan penginputan data yang salah setelah ditambahkan parameter tipe data dan check, hasil yang didapatkan meskipun data salah dipaksa untuk masuk kedalam database hasilnya data tersebut tetap gagal disimpan. Ujicoba dilakukan dengan memasukan data minus -2, -3, -4, -5, -6, -7 dan data melebihi angka 10, yaitu 12,13,14,15,16,16,17 angka tersebut merupakan angka diluar batas nilai un yang diberikan.
3. Jika sudah dilakukan filter dari penentuan tipe data dan panjang karakter, terkadang masih ada data yang kurang sesuai yang masuk ke dalam database, untuk itu perlu diperlukan filter tambahan check, Penggunaan Constraint Check selain menjamin keakuratan data juga menjamin keamanan data.

5. SARAN

Dengan adanya penelitian ini diharapkan, dapat dimanfaatkan sebagai salah satu referensi dalam desain sebuah database. Juga diharapkan adanya perhatian khusus dalam penentuan tipe data serta dalam pembuatan database. Penambahan constraint check juga perlu ditambahkan bila memang diperlukan, guna meningkatkan keakuratan data dan keamanan data.

DAFTAR PUSTAKA

- [1] Sudarma, S., 2010, *Panduan Belajar Database MySQL Database Server*, Mediakita, Jakarta
 - [2] Kristanto, K. H., 1993, *Konsep dan Perancangan Database*, Andi Publisher, Yogyakarta.
 - [3] Raharjo, S., Sutanta, E., Utami, E., 2007, *Analisis Aspek-Aspek Kualitas Schema Database (Studi Kasus Pada Database Akademik Ista Yogyakarta)*, Andi Publisher, Yogyakarta.
 - [4] Utami, E., 2014, The Advantages of Using CHECK Constraints in the Academic Database Tables, Yogyakarta, *Journal of Software*, No. 2, Vol. 9, Hal. 382-386
 - [5] Zhang, T., Xing-jun, S., 2013, Status Quo and Prospect on Mobile Database Security, *Telkonnika*, No. 9, Vol. 11, Hal. 4949-4966
 - [6] Yunus, M. A. M., Krishnan, S. K. V. G., Nawi, N. M., Surin, E. S. M., 2017, Study on Database Management System Security Issues, *JOIY International Journal on Informatics Visualization*, No. 4, Vol 1, hal. 192-194
 - [7] Zhou, S., Li, S., Liu, X., Xu, X., Zheng, S., Liao, X., Xiong, Y., 2017, Easier Said Than Done: Diagnosing Misconfiguration via Configuration Constraints Analysis: A Study of the Variance of Configuration Constraints in Source Code, *EASE'17*, Karlskrona, Sweden, June 15 - 16.
 - [8] Nugroho, I., 2011, Sistem Informasi Penerimaan Siswa Baru Berbasis Web dengan PHP dan SQL, *Skripsi*, Fakultas Teknik, Universitas Negeri Yogyakarta (UNY), Yogyakarta.
-

-
- [9] Raharjo, S., 2012, Constraint Basis Data sebagai Fondasi yang Kuat Dalam Pengembangan Sistem Informasi, *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) 2012*, Yogyakarta, 3 November.
- [10] Utami, E., Raharjo, S., 2014, Database Security Model in the Academic Information System, *International Journal of Security and its Applications*, No. 3, Vol. 8, Hal. 163-174
- [11] Raharjo, S., 2013, Integrity Constraint Basis Data Relasional Dengan Menggunakan PL/PgSQL Dan Check Constraint, *Semnas TEKNOMEDIA 2013*, Yogyakarta, 19 Januari.
- [12] Gat, 2015, Perancangan Basis Data Perpustakaan Sekolah dengan Menerapkan Model Data Relasional, *Citec Journal*, No. 4, Vol. 2, Hal. 304-315.
- [13] Hardita, V. C., Yunita, S., Sholeha, E. W., Hasan, P., 2018, Perencanaan Basis Data pada Sistem Reservasi Hotel Tickle Yogyakarta, *Seminar Nasional Sistem Informasi dan Teknologi Informasi 2018 (SENSITEK 2018)*, STMIK Pontianak, 12 Juli 2018