

**LAPORAN LUARAN KHUSUS
PROJECT BASED LEARNING (PBL)
PEMROGRAMAN BASIS DATA**



KELOMPOK : 2
ANGGOTA KELOMPOK : Tamaris Roulina Silitonga - 4342211050
Nisa Fathul Jannah - 4342211051
Alifzidan Risky Fadillah - 4342211045
Ade Galih Rezky Sinaga - 4342211057
Joel Pasaribu - 4342211033
KELAS : TRPL - 2B Malam
JUDUL PROYEK : Sistem Manajemen Hak Kekayaan Intelektual
DOSEN KOORDINATOR : Ahmadi Irmansyah Lubis, S.Kom., M.Kom.
DOSEN PENGAMPU : Ahmadi Irmansyah Lubis, S.Kom., M.Kom.

**PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM**

2023

Luaran 1: Penerapan Normalisasi Database

Hasil Normalisasi database yang dilakukan untuk sistem manajemen hak kekayaan intelektual sesuai kebutuhan sistem terbagi menjadi 6 Tabel Seperti berikut ini:

TABEL PEMOHON

id	username	image	name	email	password
43442211050	Tamaris.Roulina	d.png	Tamaris Roulina Silitonga	tamarissilitonga@gmail.com	%1324bvam

TABEL HAK CIPTA

id	pemohon_id	admin_id	jenis_ciptaan	sub_jenis_ciptaan	Judul	uraian_singkat	Tgl_pertama	Kota_pertama	keterangan	status
NPM 001	4342211050	1234	Karya tulis	Novel	Aku	Novel yang menceritakan aku	12/02/2022	Batam	null	1
NPM 002	4342211050	1234	Karya tulis	Novel	Cinta	Novel yang menceritakan aku	12/02/2022	Batam	revisi	2

TABEL ADMIN

id	username	image	name	email	password
1234	admin	d.png	sentrahki	sentra@gmail.com	%mmnmna.

TABEL PENCIPTA

id	hak_cipta_id	nama	alamat	Kode pos	provinsi	Kota	email
1	NPM001	Tamaris	Tiban	12345	Kepulauan riau	Batam	tama@gmail
2	NPM001	Ade	Bengkong	12345	Kepulauan riau	Batam	ade@gmail

TABEL LAMPIRAN

id	hak_cipta_id	ktp	surat_pernyataan	bukti_pengalihan	link_ciptaan	contoh_ciptaan
1	NPM001	d.pdf	d.pdf	d.pdf	Hai.com	Ciptaan.pdf

TABEL SERTIFIKAT

id	hak_cipta_id	sertifikat
1	NPM001	d.jpg

Penjelasan:

Normalisasi yang dilakukan sesuai dengan kebutuhan sistem dan implementasi. Normalisasi dilakukan hingga tahap 2NF, di mana setiap tabel memiliki primary key dan foreign key yang sesuai untuk menjaga integritas data dan hubungan antar tabel. Pada tahap ini, atribut-atribut yang terkait dengan ketergantungan fungsional telah dipisahkan ke dalam tabel terpisah.

Pemisahan atribut jenis ciptaan dan sub jenis ciptaan menjadi tabel terpisah seharusnya dilakukan untuk mencapai normalisasi hingga 3NF. Namun, karena ada kendala dalam implementasi, kami menggunakan alternatif yang sesuai dengan kebutuhan sistem yang telah dirancang.

Dalam implementasi tersebut, pemohon dapat melihat sub jenis yang sesuai dengan pilihan jenis ciptaan melalui tampilan form yang disediakan. Dengan menggunakan inputan yang tepat, sistem memastikan bahwa inputan subjenis dapat menyesuaikan dengan pilihan jenis ciptaan yang dipilih oleh pemohon.

Meskipun normalisasi hingga 3NF diinginkan, keputusan untuk menggunakan alternatif ini telah disesuaikan dengan kebutuhan sistem dan kendala implementasi yang ada.

Detail

Jenis Ciptaan *

Pilih ▼

Sub Jenis Ciptaan *

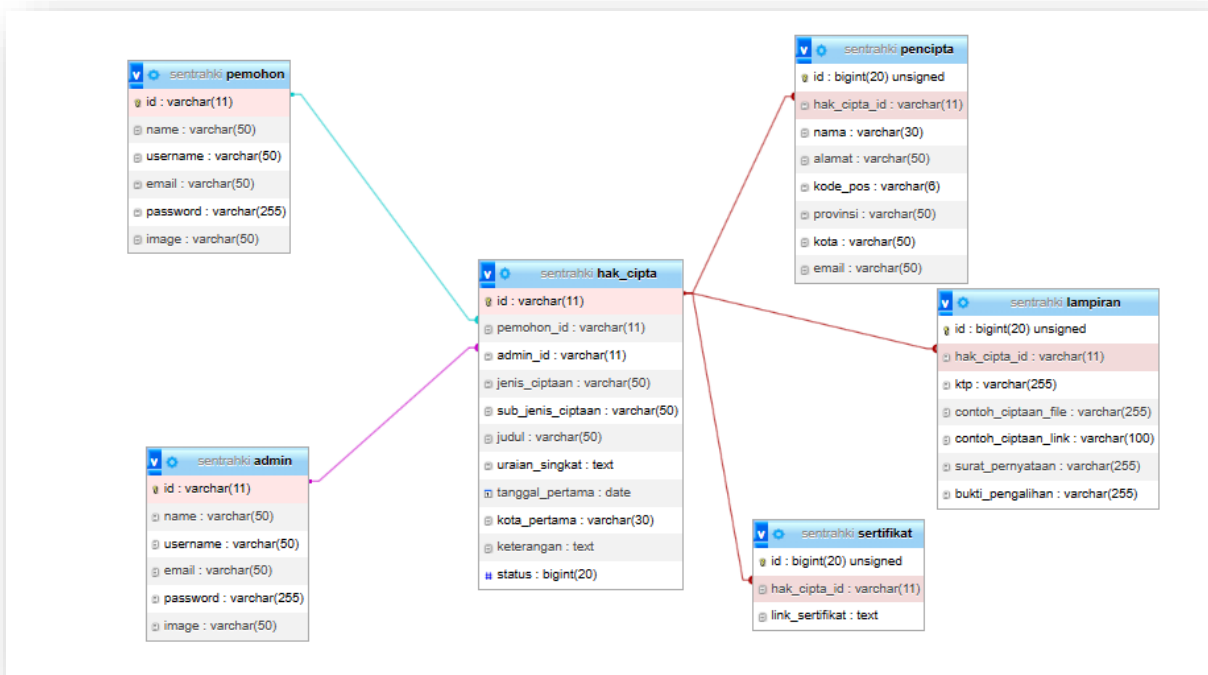
Isi sub jenis ciptaan...

Keterangan: Sub jenis ciptaan dapat dilihat di sini.

Judul *

\

Tabel relasi yang dihasilkan.



- Tabel **Pemohon** dan **Admin** memiliki relasi One-to- Many- dengan Tabel **HakCipta**
- Tabel **Lampiran** memiliki relasi One-to-One dengan tabel **HakCipta**.
- Tabel **HakCipta** memiliki relasi One-to- Many- dengan Tabel **Pencipta**
- Tabel **Sertifikat** memiliki relasi One-to-One dengan tabel **HakCipta**.

Luaran 2: Penerapan Query Lanjutan

Penerapan query tingkat lanjut yang digunakan untuk sistem manajemen hak kekayaan intelektual dapat dilihat sebagai berikut:

1. Penggunaan Query Agregat:

Query agregat digunakan untuk melakukan operasi penghitungan pada data yang memenuhi kondisi tertentu. Fungsi **count()** digunakan untuk menghitung jumlah data yang memenuhi kondisi tertentu.

❖ **\$accPusat = HakCipta::where('status', 1)->has('sertifikat')->count();**

Query ini menghitung jumlah data pada model **HakCipta** yang memiliki nilai kolom 'status' = 1 dan memiliki relasi 'sertifikat'. Fungsi **count()** digunakan untuk menghitung jumlah data yang memenuhi kondisi tersebut.

❖ **\$accAdmin = HakCipta::where('status', 1)->doesntHave('sertifikat')->count();**

Query ini menghitung jumlah data pada model **HakCipta** yang memiliki nilai kolom 'status' = 1 dan tidak memiliki relasi 'sertifikat'. Fungsi **count()** digunakan untuk menghitung jumlah data yang memenuhi kondisi tersebut.

❖ **\$tolakAdmin = HakCipta::whereIn('status', [2, 4])->where('keterangan', '<>', null)->count();**

Query ini menghitung jumlah data pada model **HakCipta** yang memiliki nilai kolom 'status' yang merupakan salah satu dari [2, 4] dan nilai kolom 'keterangan' tidak sama dengan null. Fungsi **count()** digunakan untuk menghitung jumlah data yang memenuhi kondisi tersebut.

❖ **\$tolakPusat = HakCipta::whereIn('status', [2, 4])->where('keterangan', null)->count();**

Query ini menghitung jumlah data pada model **HakCipta** yang memiliki nilai kolom 'status' yang merupakan salah satu dari [2, 4] dan nilai kolom 'keterangan' bernilai null. Fungsi **count()** digunakan untuk menghitung jumlah data yang memenuhi kondisi tersebut.

```
if (Auth::guard('admin')->check()) {  
    $accPusat = HakCipta::where('status', 1)->has('sertifikat')->count();  
    $accAdmin = HakCipta::where('status', 1)->doesntHave('sertifikat')->count();  
    $tolakAdmin = HakCipta::whereIn('status', [2, 4])->where('keterangan', '<>', null)->count();  
    $tolakPusat = HakCipta::whereIn('status', [2, 4])->where('keterangan', null)->count();  
}
```

2. Penggunaan Subquery:

Subquery digunakan untuk mengambil data berdasarkan hasil dari subquery atau query internal. Hasil subquery digunakan untuk mendapatkan data yang memenuhi kondisi tertentu.

- ❖ **\$tolakAdmin = HakCipta::whereIn('status', [2, 4])->where('keterangan', '<>', null)->get();** Query ini mengambil data pada model **HakCipta** yang memiliki nilai kolom 'status' yang merupakan salah satu dari [2, 4] dan nilai kolom 'keterangan' tidak sama dengan null. Fungsi **get()** digunakan untuk mengambil data yang memenuhi kondisi tersebut.
- ❖ **\$tolakPusat = HakCipta::whereIn('status', [2, 4])->where('keterangan', null)->get();** Query ini mengambil data pada model **HakCipta** yang memiliki nilai kolom 'status' yang merupakan salah satu dari [2, 4] dan nilai kolom 'keterangan' bernilai null. Fungsi **get()** digunakan untuk mengambil data yang memenuhi kondisi tersebut.

```
if (Auth::guard('admin')->check()) {
    $tolakAdmin = HakCipta::whereIn('status', [2, 4])->where('keterangan', '<>', null)->get();
} else {
    $tolakAdmin = HakCipta::whereIn('status', [2, 4])->where('pemohon_id', Auth::user()->id)->where('keterangan', '<>', null)->get();
}
return view('hak_cipta_tolak_admin', [
    'page' => 1,
    'hakCipta' => $tolakAdmin,
]);
```

3. Penggunaan Constraint dan PL/SQL:

Penggunaan fungsi **where()** dan **whereIn()** digunakan untuk mengatur kondisi dan memfilter data berdasarkan nilai kolom yang diberikan.

- ❖ **\$hakCipta = HakCipta::whereIn('status', [0, 3])->get();**
Query ini mengambil data pada model **HakCipta** yang memiliki nilai kolom 'status' yang merupakan salah satu dari [0, 3]. Fungsi **get()** digunakan untuk mengambil data yang memenuhi kondisi tersebut.
- ❖ **\$cek1 = HakCipta::where('status', -1)->where('admin_id', \$uid)->first();**
Query ini mengambil data pertama pada model **HakCipta** yang memiliki nilai kolom 'status' = -1 dan nilai kolom 'admin_id' = \$uid. Fungsi **first()** digunakan untuk mengambil data pertama yang memenuhi kondisi tersebut.
- ❖ **\$cek2 = NULL;** Variabel **\$cek2** diinisialisasi dengan nilai null.
- ❖ **if (\$cek1) { ... }** Pada blok kode ini, dilakukan pengecekan apakah **\$cek1** memiliki nilai atau tidak. Jika **\$cek1** memiliki nilai (data yang memenuhi kondisi), maka blok kode dalam if statement akan dieksekusi.

```
if (Auth::guard('admin')->check()) {
    $hakCipta = HakCipta::whereIn('status', [0, 3])->get();
} else {
    $hakCipta = HakCipta::where('pemohon_id', Auth::user()->id)->get();
}
```

Luaran 3: Penerapan Skema Manajemen Hak Akses Client – Server

1. Proses Koneksi Antar Device pada Project:

- Device client terhubung ke jaringan menggunakan koneksi internet atau jaringan lokal yang sama dengan server.

```
C:\Users\HP>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 11:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

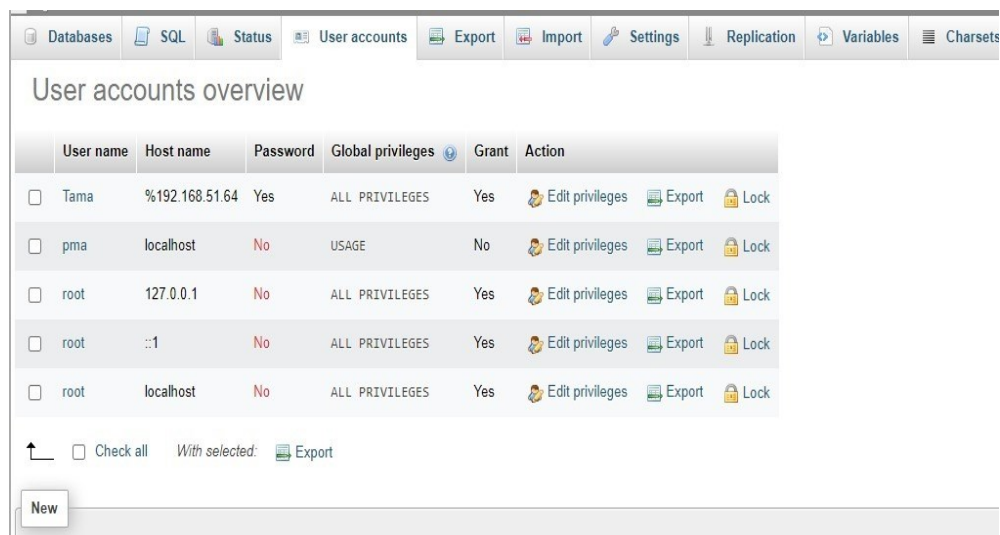
Wireless LAN adapter Local Area Connection* 12:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::e9b8:f1c2:299b:41c%11
    IPv4 Address. . . . . : 192.168.51.64
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.51.125
```

- Sebagai server, tambahkan user accounts pada database

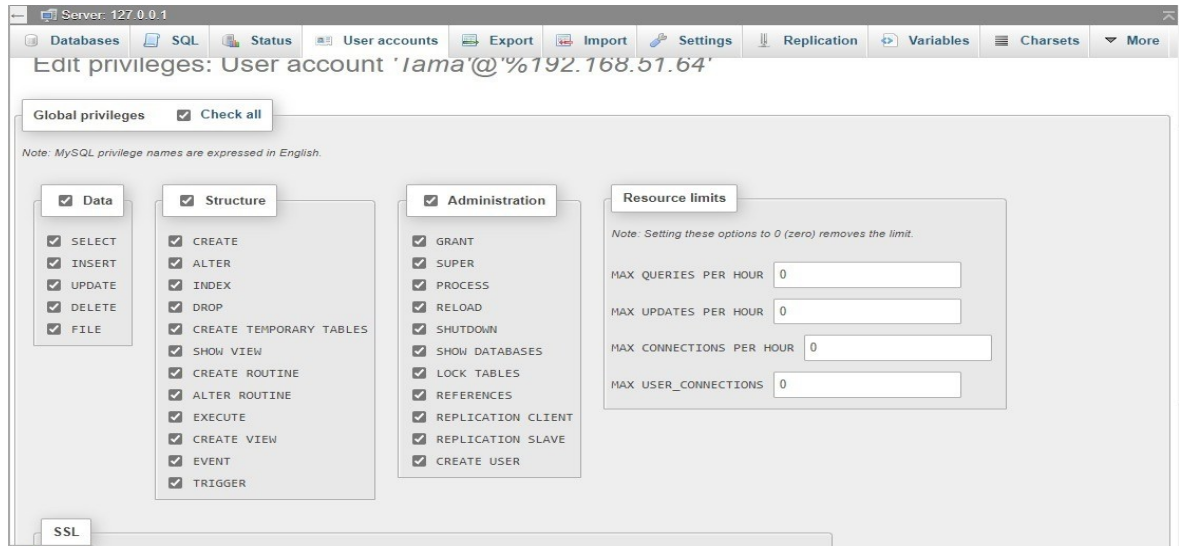


- Lakukan perubahan dalam file .env menjadi seperti ini

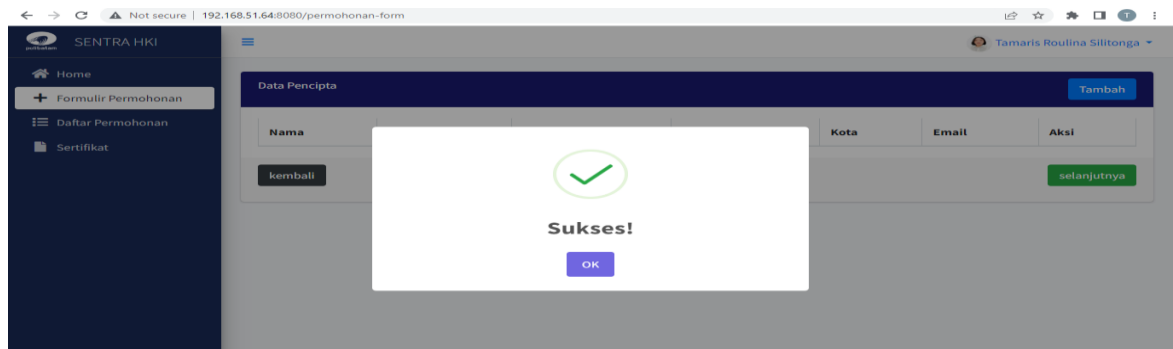
```
10
11 DB_CONNECTION=mysql
12 DB_HOST=192.168.51.1
13 DB_PORT=3306
14 DB_DATABASE=sentra
15 DB_USERNAME=Tama
16 DB_PASSWORD=123456
17
```

2. Pemberian Privileges:

- Hak akses dan privileges diberikan oleh administrator atau pengelola sistem.
 - Administrator atau pengelola sistem dapat menggunakan alat manajemen database untuk memberikan hak akses dan privileges kepada pengguna.
- Admin memberikan hak akses untuk seluruhnya



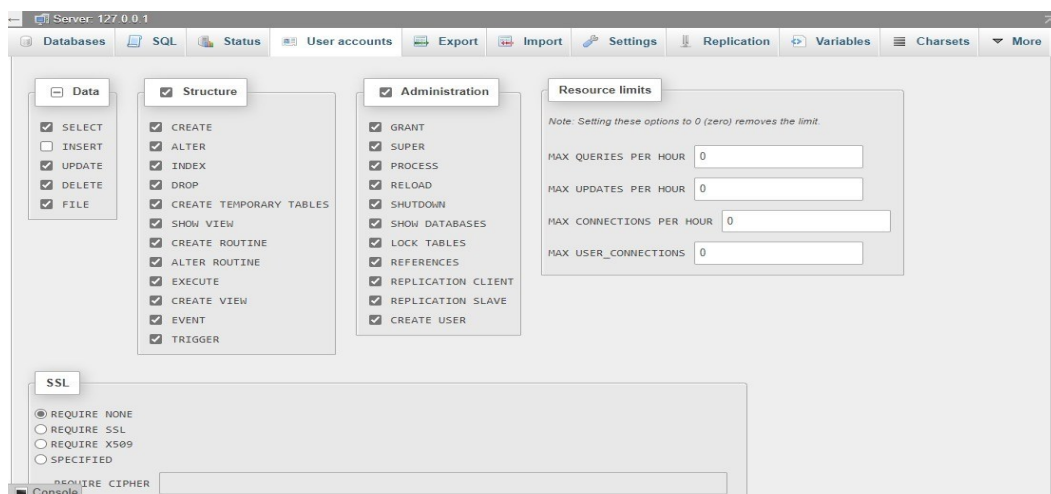
- Setelah hak akses dan privileges diberikan, pengguna dapat mengakses dan berinteraksi dengan database sesuai dengan hak akses yang diberikan.
- Hasilnya data berhasil tersimpan karena pengguna memiliki hak akses untuk melakukan insert data



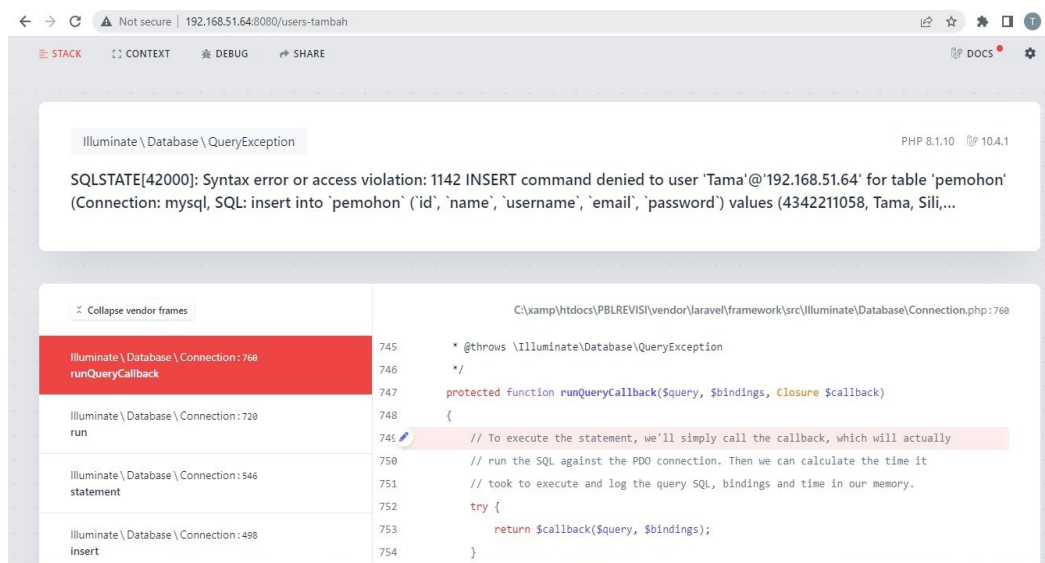
3. Pencabutan Privileges:

- Pencabutan privileges dilakukan oleh administrator atau pengelola sistem ketika pengguna tidak lagi membutuhkan hak akses tersebut atau jika ada pelanggaran kebijakan.
- Administrator atau pengelola sistem menggunakan alat manajemen database untuk mencabut hak akses dan privileges pengguna.
- Setelah pencabutan, pengguna tidak lagi memiliki akses ke sumber daya atau operasi yang sebelumnya diizinkan

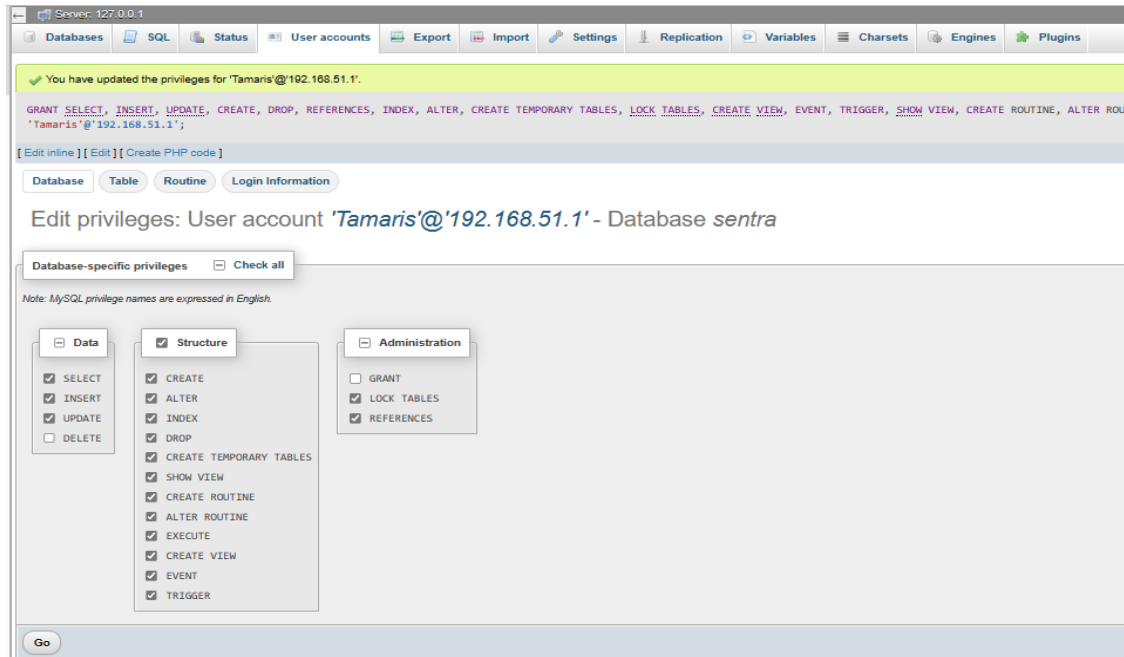
- Admin membatasi hak akses untuk insert



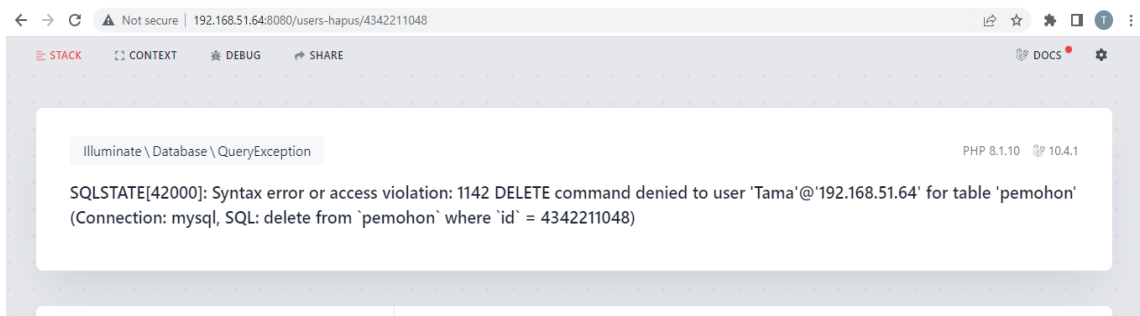
- Setelah hak akses dan privileges dibatasi, pengguna tidak dapat mengakses dan berinteraksi dengan database sesuai dengan hak akses yang dibatasi.
- Hasilnya error karena pengguna tidak memiliki hak akses untuk melakukan insert data



- Admin membatasi hak akses untuk insert
- Setelah hak akses dan privileges dibatasi, pengguna tidak dapat mengakses dan berinteraksi dengan database sesuai dengan hak akses yang dibatasi.
- Hasilnya error karena pengguna tidak memiliki hak akses untuk melakukan insert data



- Hasilnya error karena pengguna tidak memiliki hak akses untuk melakukan insert data



Luaran 4: Penggunaan Tools

1) Jelaskan Tools database yang kalian gunakan.

Pada pengembangan sistem manajemen hak kekayaan intelektual kami menggunakan MySQL sebagai basis data untuk sistem. MySQL adalah sistem manajemen basis data relasional (RDBMS) yang populer dan sering digunakan. RDBMS adalah jenis basis data yang didasarkan pada model relasional, di mana data disimpan dalam bentuk tabel yang terhubung satu sama lain melalui relasi. MySQL menyediakan alat dan fitur untuk menyimpan, mengelola, dan mengakses data dengan efisien.

2) Dan jelaskan alasannya

Menggunakan MySQL sebagai basis data dalam sistem manajemen HKI memiliki beberapa alasan yang kuat. Pertama, MySQL menawarkan performa yang tinggi, sehingga sistem kami dapat memberikan respons yang cepat dalam mengelola data HKI. Kedua, MySQL kompatibel dengan Laravel, memudahkan penggunaan dan integrasi dengan kerangka kerja yang kami gunakan. Ketiga, dukungan komunitas yang luas memastikan ada banyak sumber daya dan bantuan yang tersedia jika kami mengalami masalah. Terakhir, fitur keamanan dan kontrol akses MySQL memungkinkan Anda melindungi data HKI dengan baik dan memastikan hanya pengguna yang diizinkan yang dapat mengaksesnya.

Dengan memilih MySQL untuk sistem manajemen HKI kami, kami dapat mengandalkan performa tinggi, dukungan komunitas, kompatibilitas dengan Laravel, serta keamanan dan kontrol akses yang kuat.