

2023

PBW

Praktikum 7 – NodeJS MySQL

Silakan buat halaman web sesuai petunjuk. Gunakan template yang telah disediakan.

(Adopted and edited from Modul 9 - PHP MySQL Access (2017) by Maria Veronica)

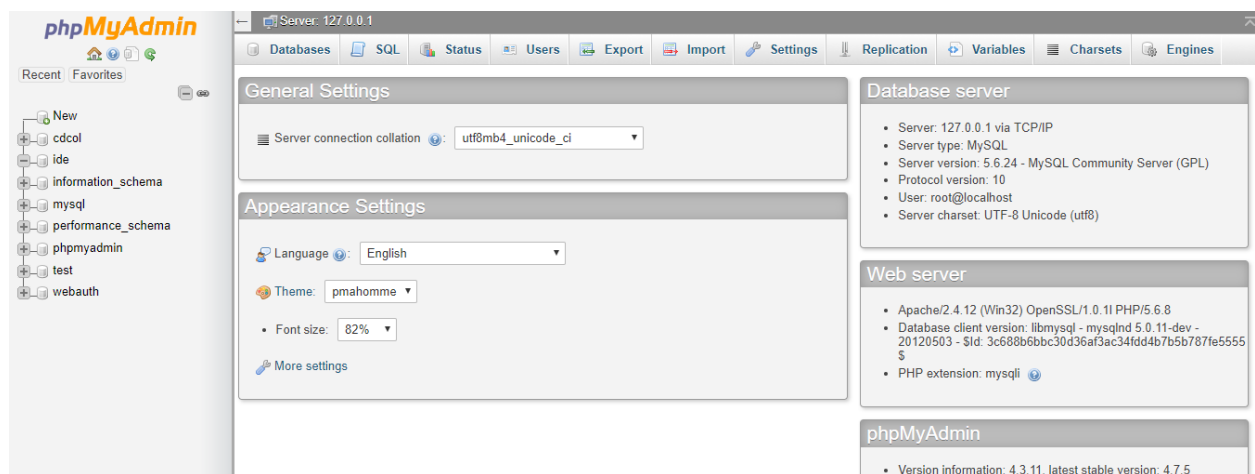


PETUNJUK Pengerjaan

1. Rename folder template menjadi T07xxyyy.
2. Gunakan template yang sudah diberikan.
3. Diasumsikan anda sudah dapat melakukan import basis data
4. JANGAN MENGUBAH NAMA FILE.
5. Masukkan file txt ke dalam zip dan upload pekerjaan Anda ke IDE sesuai jam pengumpulan yang tertera.

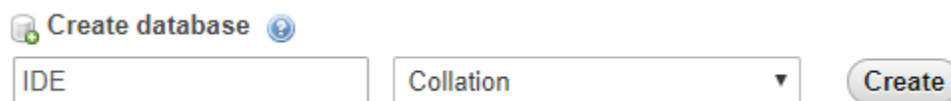
MySQL Access

Pada modul ini anda akan mempelajari cara mengakses MySQL *database* melalui NodeJS. Pertama-tama, bukalah halaman phpMyAdmin melalui browser anda. Jika berhasil, maka akan tampil halaman seperti Gambar 1.



Gambar 1 – Tampilan awal phpMyAdmin.

Klik tombol **New** yang terdapat pada bagian kiri. Buatlah sebuah database dengan nama **IDE**.



Pada folder **DATA** sudah disediakan sebuah file **ide.sql** yang berisi query sql yang dibutuhkan di modul ini. Bukalah menu **Import** pada halaman phpMyAdmin kemudian pilih file **ide.sql** untuk di-import. Jika berhasil, maka akan muncul 2 buah tabel beserta isinya pada database **ide**. Sebagai informasi, berikut adalah keterangan dari tabel **users**.

Nama Field	Tipe Data	Sifat
ID_U	int	Auto increment, Primary key
userID	varchar(10)	Unique

name	varchar(50)	-
username	varchar(9)	Unique
pass	varchar(50)	-
ID_UG	int	Foreign Key (dari tabel UserGroups)

MySQL Connection.

Pada contoh ini, lokasi *database* berada di *localhost* dengan nama database adalah **IDE**. Jika anda sebelumnya belum melakukan konfigurasi apapun terhadap MySQL, maka username default yang digunakan adalah 'root' dengan passwordnya adalah '' (kosong). Jika anda sempat mengganti, silahkan sesuaikan dengan kondisi masing-masing.

Pada folder **TEMPLATE**, sudah disediakan file-file yang dibutuhkan untuk mengerjakan modul ini. Silahkan lakukan install untuk menambah library yang dibutuhkan untuk pengerjaan ini. Halaman html sudah dibuat secukupnya. Jika dirasa kurang, silahkan melakukan perubahan secara pribadi. Silahkan pelajari terlebih dahulu kode apa yang sudah dibuat. Fokus dari praktikum ini adalah koneksi dengan basis data, sehingga untuk keindahan tampilan, mungkin dapat dikesampingkan terlebih dahulu.

Penting! Jika anda menemukan error akibat bagian instruksi yang belum dicapai, silahkan ubah menjadi komentar pada bagian tersebut atau membuat file baru, lalu pindahkan secara bertahap. Ini umumnya berada pada bagian *ejs* yang membutuhkan data dari server atau bagian *include* yang sebenarnya belum dibutuhkan pada bagian tersebut.

SELECT

Pertama-tama kita akan melihat halaman utama (menggunakan tampilan **users.ejs**). Halaman ini akan menampilkan tabel data users seperti dapat dilihat pada Gambar 2.

User ID	Name	Role
20160065	Maria Veronica Claudia	lecturer
20130053	Husnul Hakim	lecturer
2011730053	MARIA VERONICA	student
2011730103	JOHANES MARIO ADRIANO	student
2012730027	NICHOLAS MARTIN PRIBADI	student
2012730078	RIZQI PUTRA PRATAMA	student
2012730093	YOHAN STURE ENANDER	student
2013730001	ALVIN IRAWAN	student
2013730002	CHERIA	student
2013730003	FADHIL AHSAN	student
2013730004	CTARA	student

Gambar 2 – Tampilan pada url '/'

Query SQL yang digunakan untuk menampilkan data tersebut adalah sebagai berikut.

```
SELECT * FROM user_data
```

Intruksi pengerjaan adalah sebagai berikut:

1. Implementasikan fungsi untuk mengeksekusi query yang sudah dibuat, lalu data yang dihasilkan dikirim ke sisi klien dan ditampilkan satu per satu.
2. Dengan melihat implementasi pada halaman utama, implementasikan pula halaman userGroup dengan cara yang sama (menggunakan tampilan di **userGroup.ejs**). Halaman ini dapat dilihat pada Gambar 3.



The image shows a web interface for managing roles. At the top, there is a dropdown menu labeled 'Role' with three options: 'lecturer', 'student', and 'admin'. Below this, there is a section titled 'Add New Role' which contains a text input field and an 'ADD' button.

Gambar 3 – Halaman userGroup

SELECT ... WHERE ...

Misalkan Anda ingin mencari user yang namanya mengandung huruf-huruf tertentu. Anda hanya perlu menambahkan sintaks WHERE dalam kode MySQL. Untuk mencari teks string maka digunakan keyword LIKE. Pada halaman user terdapat form untuk mencari berdasarkan nama. Jika dimasukan teks pada input lalu ditekan tombol SEARCH, maka halaman akan tetap di halaman ini, namun isi dari tabel sudah berubah sesuai dengan hanya menampilkan data yang memiliki potongan teks di namanya. Contoh jika diinput keyword “vi” maka akan dimunculkan 4 buah data.

Hint: pastikan anda menggunakan cara eksekusi query yang tidak akan menyebabkan SQL injection.

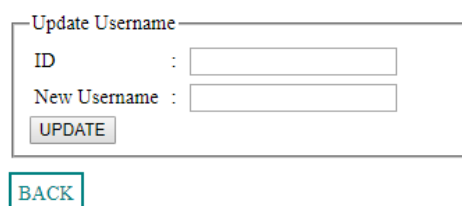
INSERT

Salah satu operasi dasar dalam pemrograman basis data adalah *insert*. Operasi ini dilakukan untuk menambahkan *record* baru ke dalam tabel pada basis data. Pada modul ini, anda akan mencoba menambahkan role baru sesuai input dari pengguna dengan mengisi form yang disediakan. Pada halaman userGroup, jika tombol ADD ditekan, maka dilakukan penambahan row baru pada tabel usergroups. Setelah itu halaman akan kembali ke halaman itu namun data sudah bertambah.

Hint: gunakan perintah redirect.

UPDATE

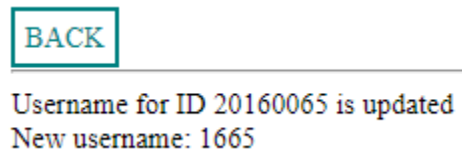
Bukalah halaman update anda yang akan menampilkan halaman pada Gambar 4 berikut.



The image shows a web form titled 'Update Username'. It contains two input fields: 'ID' and 'New Username', each preceded by a colon. Below these fields is an 'UPDATE' button. At the bottom of the form, there is a 'BACK' button.

Gambar 4 – Halaman update.

Halaman ini berguna untuk mengupdate username yang ada di table user dengan ID tertentu. Jika tombol UPDATE ditekan, maka isi dari form akan dikirim ke server. Implementasikan kode NodeJS yang menerima input tersebut, lalu melakukan update pada basis data sesuai dengan yang dikirimkan. Setelah update dilakukan, halaman kembali ke halaman update. Jika berhasil, maka akan ditampilkan keterangan di bawah tombol BACK sesuai dengan Gambar 5.



Gambar 5 – Halaman update jika berhasil melakukan perubahan.

Jika gagal, cek di basis data anda, apakah data sudah berubah. Jika sudah berubah, mungkin ada sedikit kesalahan saat melakukan SELECT sehingga informasi di atas tidak muncul. Jika *database* juga belum *ter-update*, periksa kembali seluruh kode anda untuk bagian update ini.

LIMIT

Saat data yang ditampilkan sangat banyak, biasanya tampilan akan dibagi menjadi beberapa halaman. Sintaks *mysql* untuk membatasi jumlah *record* yang ditampilkan adalah `LIMIT x,y`. Dengan menggunakan sintaks tersebut, *record* yang ditampilkan adalah *record* baris ke-x hingga baris ke-y dari hasil eksekusi *query*. Bukalah kembali file `users.php` anda.

1. Ubahlah agar pada halaman `users` (halaman utama) menampilkan maksimal 8 data per halamannya dan sisanya harus dilakukan dengan berganti halaman.

Syntax SQL = `SELECT ... FROM ... WHERE ... LIMIT start, show.`

2. Start menampilkan halaman dimulai dari baris berapa. Show menunjukkan berapa banyak yang ditampilkan di satu layar. Sebagai contoh pada modul ini akan terbagi menjadi 5 halaman dengan keterangan sebagai berikut.
 - a. Halaman pertama menampilkan data dari baris 0-7
 - b. Halaman kedua menampilkan data dari baris 8-15
 - c. Halaman ketiga menampilkan data dari baris 16-23
 - d. dst.
3. Jika digunakan pagination, maka pada url ditambahkan parameter misal `'?start=16'`. Jika tidak ada parameter ini, maka dianggap sebagai tampilkan semua (tanpa pagination).
4. Link untuk setiap halamannya dibentuk dengan memanfaatkan tampilan pagination yang dapat dilihat pada Gambar 6. HTML pagination dapat dilihat pada file **pagination.ejs**. (BUKAN HARDCODE)

Search by Name

UPDATE

1

2

3

4

5

User ID	Name	Role
2013730002	CHERIA	student
2013730003	FADHIL AHSAN	student
2013730004	CLARA	student
2013730005	ALDY MARCELLINO CHRISTIAN	student
2013730006	ANTONIUS	student
2013730008	ENRICO FINDLEY	student
2013730009	ROMMY KURNIAWAN WIJAYA	student
2013730010	YOSUA YUUTA BIMA PRAMANA	student

Gambar 6 – Tampilan untuk bagian pagination.

- Untuk mengetahui berapa banyak link pagination yang ditampilkan, dapat digunakan perhitungan:

```
const pageCount = jumlah_row / show;
```

- Setiap link dapat ditekan dan mengarah ke halaman yang sesuai. (Tabel sesuai page berapa)
- Challenge (Opsional): Ubahlah program halaman ini dengan mengkombinasikan `where` dengan `limit`. Contoh tampilkan daftar mahasiswa yang mengandung huruf `i` pada halaman ketiga.

Kumpulkan semua pekerjaan dalam bentuk zip dengan nama `T07xxyyy.zip`

= SELAMAT MENGERJAKAN =