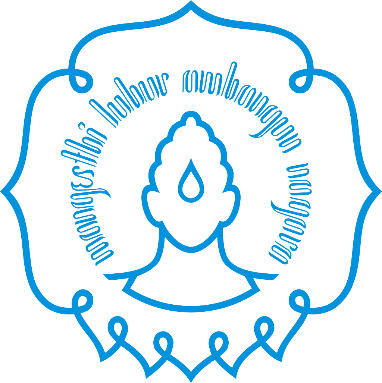
LAPORAN PRATIKUM BASIS DATA

**Kuliah Praktisi SQL – Injection**



**Disusun Oleh:**

**Puput Surya Ningtyas**

**(V3923015)**

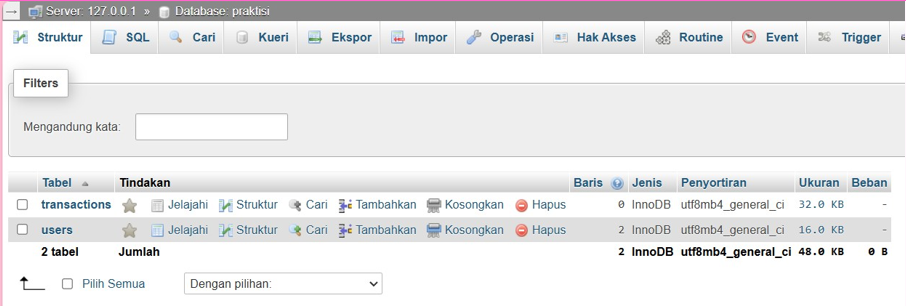
**Dosen Pembimbing:**

**Ibu Masbahah, S.Pd, M.Pd.**

**PS D-III TEKNIK INFORMATIKA SEKOLAH VOKASI UNIVERSITAS SEBELAS MARET 2024**

# HASIL DAN PEMBAHASAN

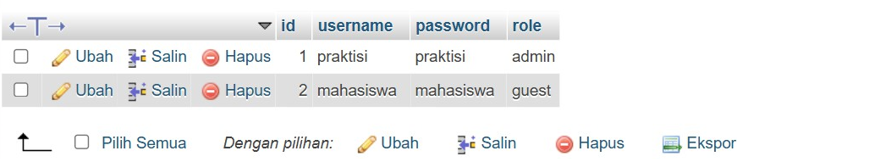
1. **Membuat Database**
   1. Membuat Tabel



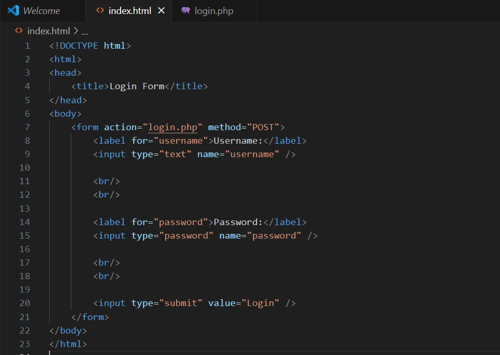


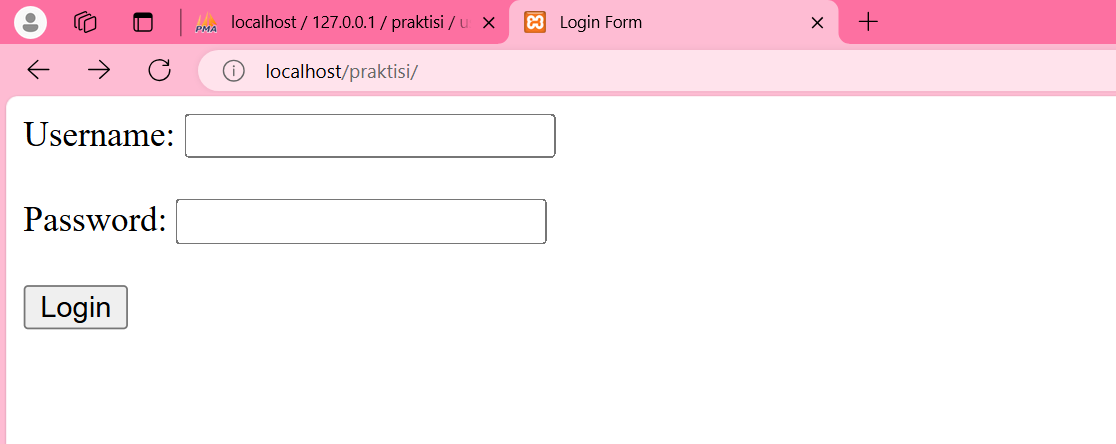


* 1. Memasukkan data ke tabel

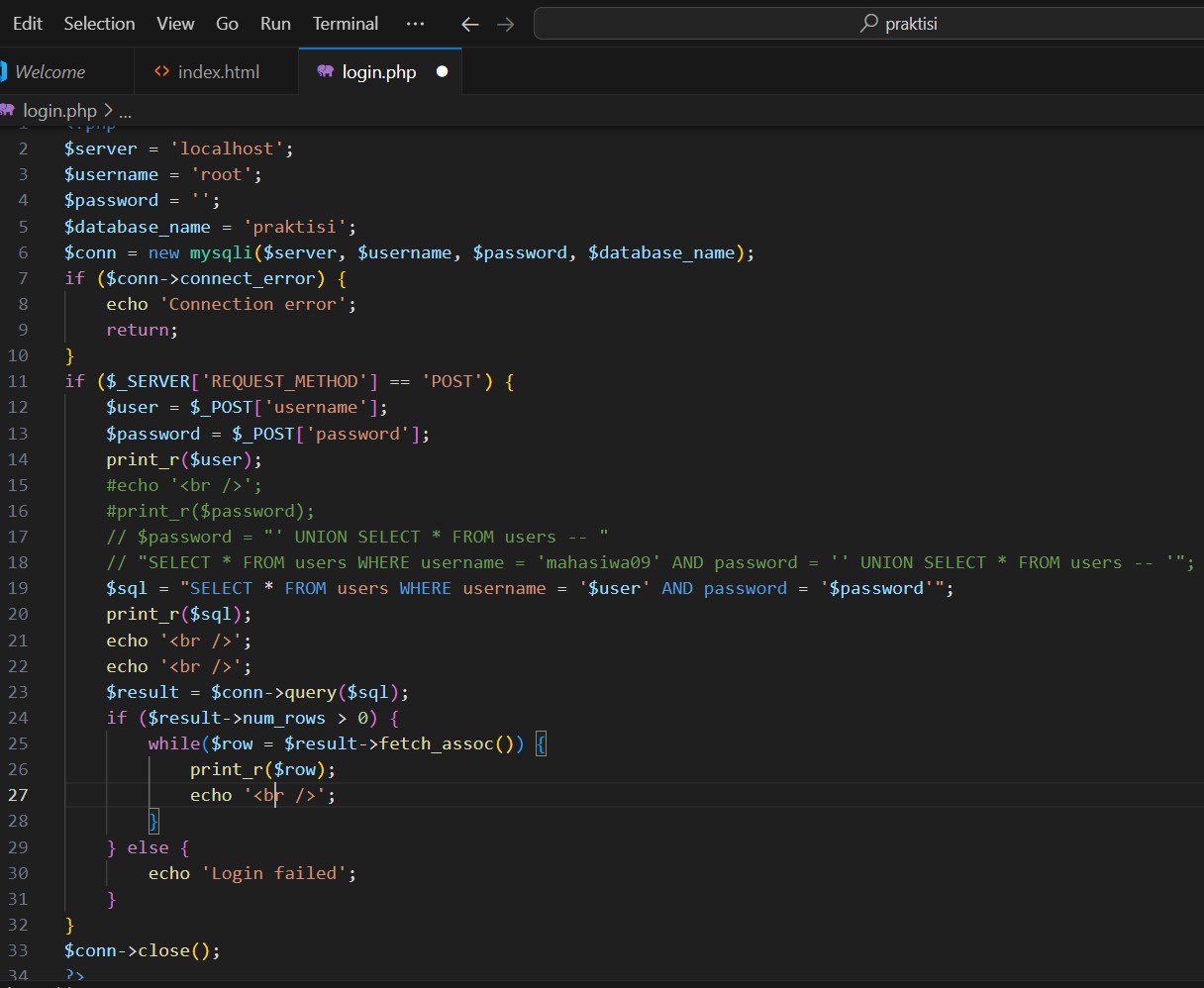


1. **Percobaan SQL Injection**
   1. **Membuat kode html untuk form login**

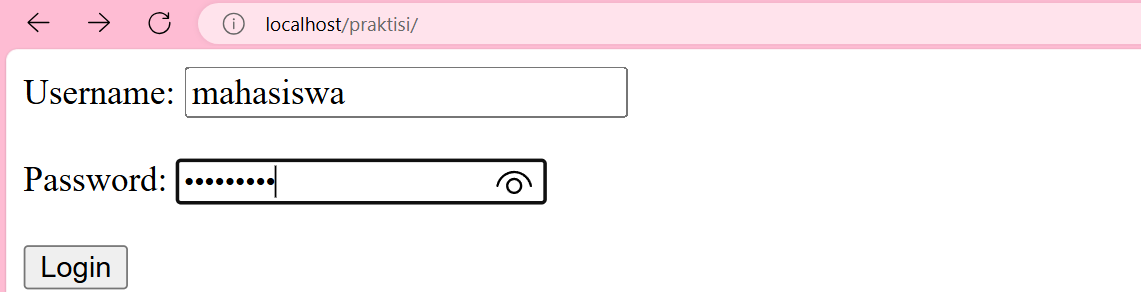
****



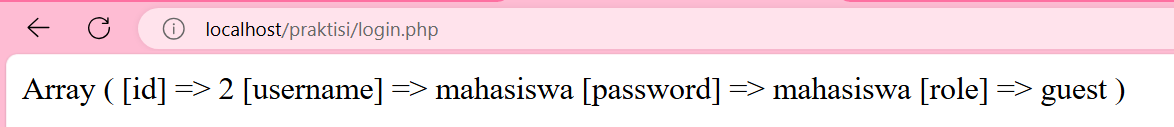
* 1. **Membuat kode php untuk memproses login dan sql injection**



* 1. **Mencoba Login dan SQL Injection**
     1. Jika username “ mahasiswa” dan password “ mahasiswa”



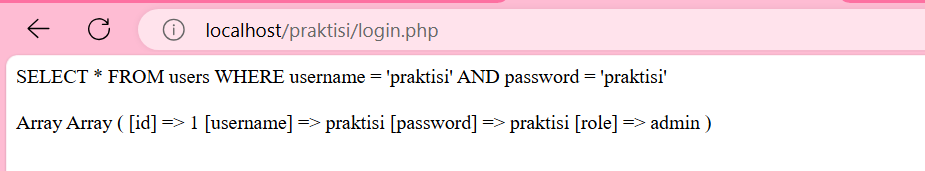
Maka hasil outputnya adalah



* + 1. Jika username “ praktisi” dan password “ praktisi”

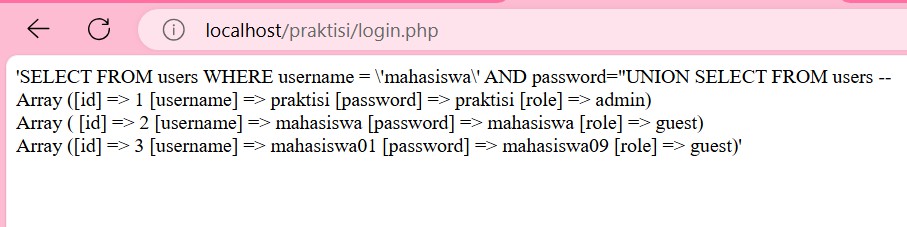


Maka hasil outputnya adalah

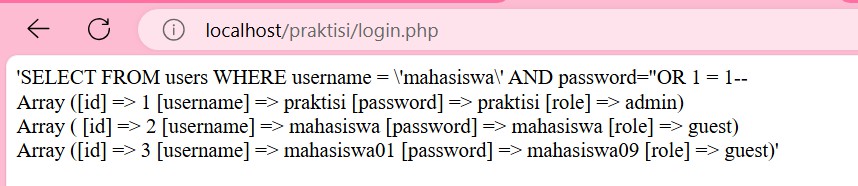


* + 1. Jika username “ mahasiswa” dan password “ ‘ UNION SELECT FROM users -

- ” maka outputnya adalah



* + 1. Jika username “ mahasiswa” dan password “ ‘OR 1 = 1-- ” maka outputnya adalah



* 1. **Penjelasan**

SQL Injection adalah serangan keamanan di mana penyerang memanfaatkan celah dalam validasi input pada aplikasi web untuk menyisipkan perintah SQL tambahan atau memanipulasi query yang dieksekusi oleh database. Tujuannya adalah untuk mencuri data sensitif, mengubah atau menghapus data, atau bahkan mengambil alih kendali aplikasi atau server. Pencegahan meliputi penggunaan parameterized queries, validasi input dengan benar, dan membatasi hak akses database sesuai kebutuhan aplikasi.

# `UNION SELECT \* FROM`

adalah perintah SQL yang digunakan untuk menggabungkan hasil dari dua atau lebih query `SELECT` menjadi satu hasil tunggal. Ini berguna ketika pengguna ingin mengambil data dari dua tabel atau menggabungkan data yang memiliki struktur yang serupa. Dalam penggunaan yang sah, `UNION SELECT` memerlukan bahwa jumlah kolom dan tipe data kolom yang dihasilkan oleh setiap query harus sesuai atau sepadan, sehingga hasilnya dapat digabungkan secara tepat.

Namun, dalam konteks keamanan, frasa ini sering kali dieksploitasi dalam serangan SQL Injection. Dalam serangan ini, penyerang mencoba memanfaatkan celah dalam validasi input pada aplikasi web untuk menyisipkan

`UNION SELECT` bersama dengan query tambahan yang disusun untuk mencuri data sensitif atau mengubah perilaku aplikasi. Misalnya, penyerang bisa mencoba menyisipkan `UNION SELECT` setelah kondisi `WHERE` pada query SQL untuk mendapatkan akses ke informasi pengguna atau data penting lainnya yang seharusnya tidak dapat diakses.

# OR 1=1 - -

Ekspresi `OR 1=1 --` adalah contoh umum dari serangan SQL Injection yang dimaksudkan untuk mengubah atau memanipulasi query SQL. Dengan menggunakan `OR 1=1`, penyerang mencoba melewati kondisi otentikasi atau pembatasan akses dengan menghasilkan pernyataan logika yang selalu benar (`1=1`). Tanda `--` digunakan untuk menghindari bahwa bagian dari query berikutnya diabaikan sebagai komentar dalam SQL, mengakibatkan database mengabaikan sisa query yang mungkin ada setelahnya. Langkah pencegahan melibatkan penggunaan parameterized queries dan validasi input yang ketat untuk mengurangi risiko dari jenis serangan ini.