TUGAS KEAMANAN SIBER

SQL INJECTION

Disusun Oleh:

Nama : Rangga Prabaswara

NIM : 3.34.21.0.21

Kelas : IK-2A

Jurusan : Teknik Elektro

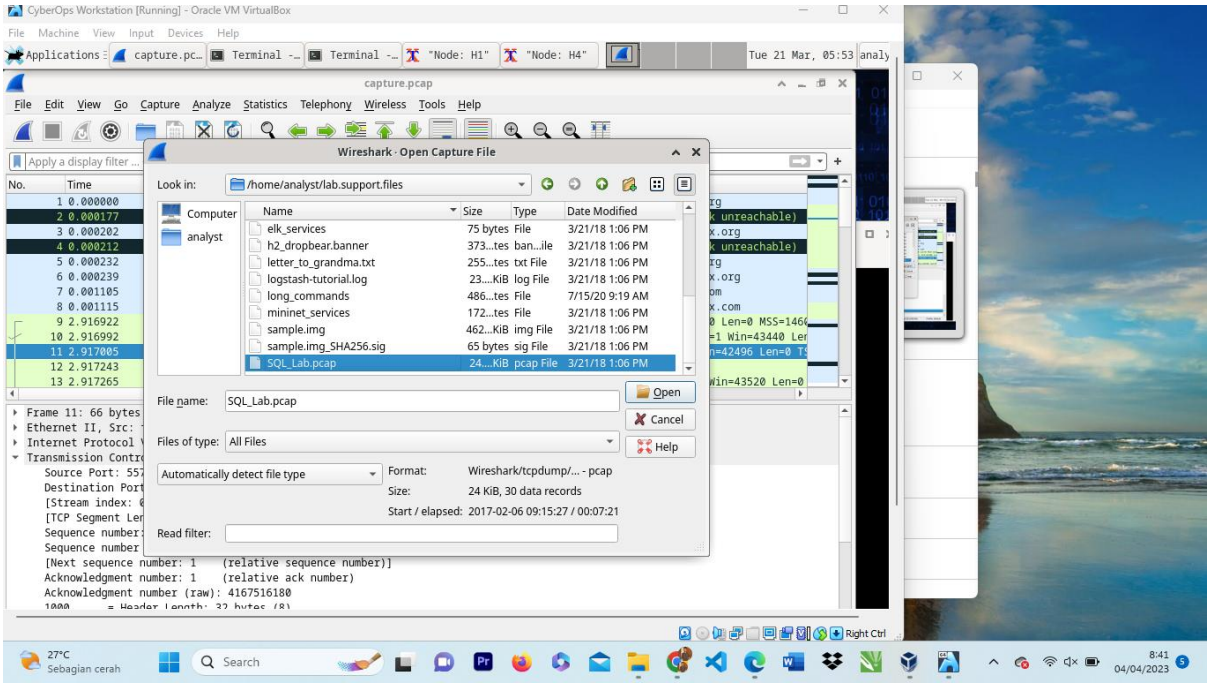
Prodi : D3 - Teknik Informatika

JURUSAN TEKNIK ELEKTRO

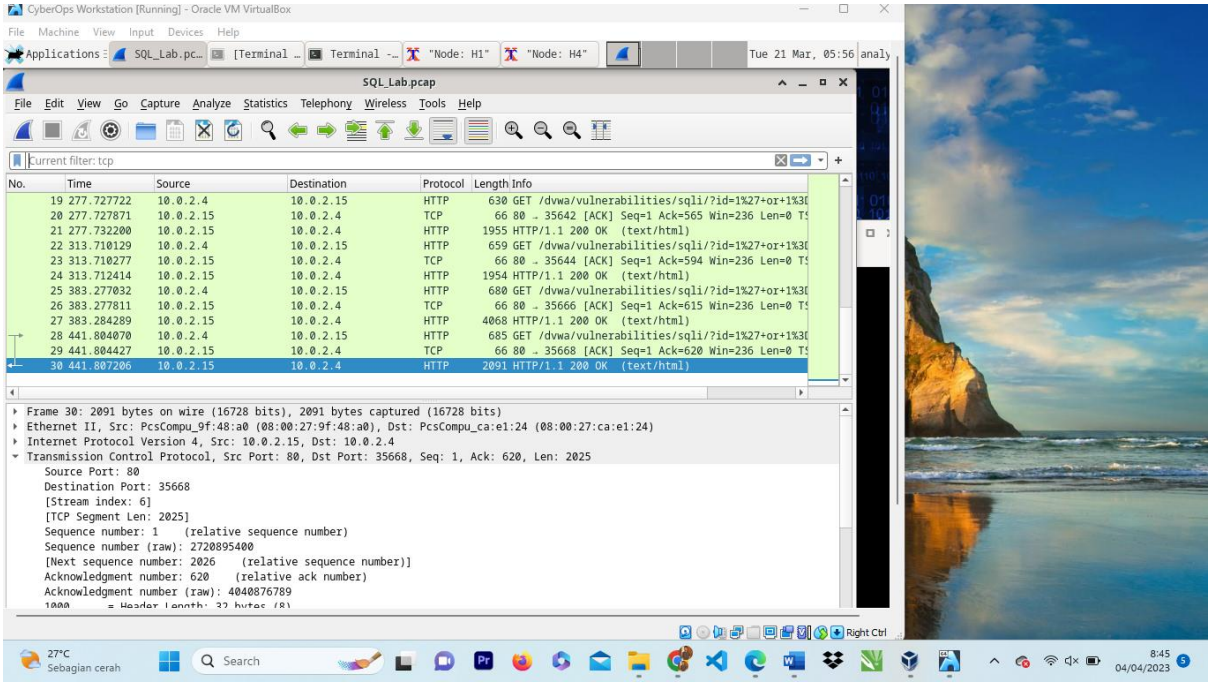
PROGRAM STUDI D3 TEKNIK INFORMATIKA

POLITEKNIK NEGERI SEMARANG

Membuka wireshark dan membuka file SQL.Lab.pcap

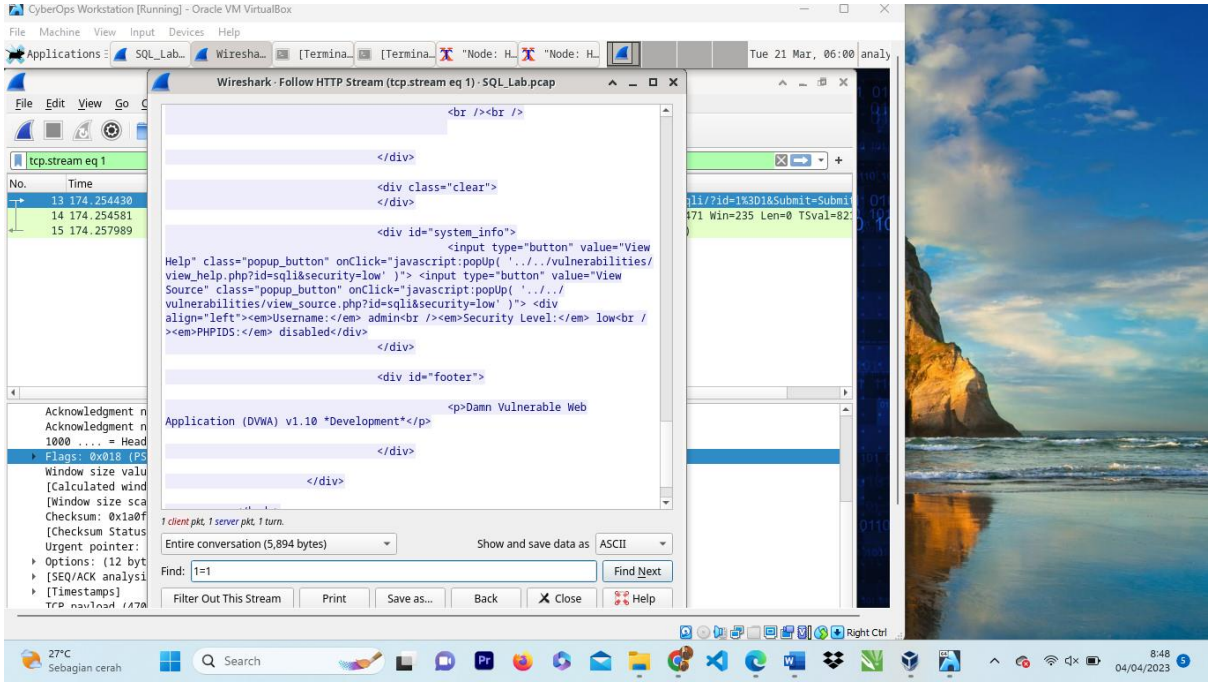


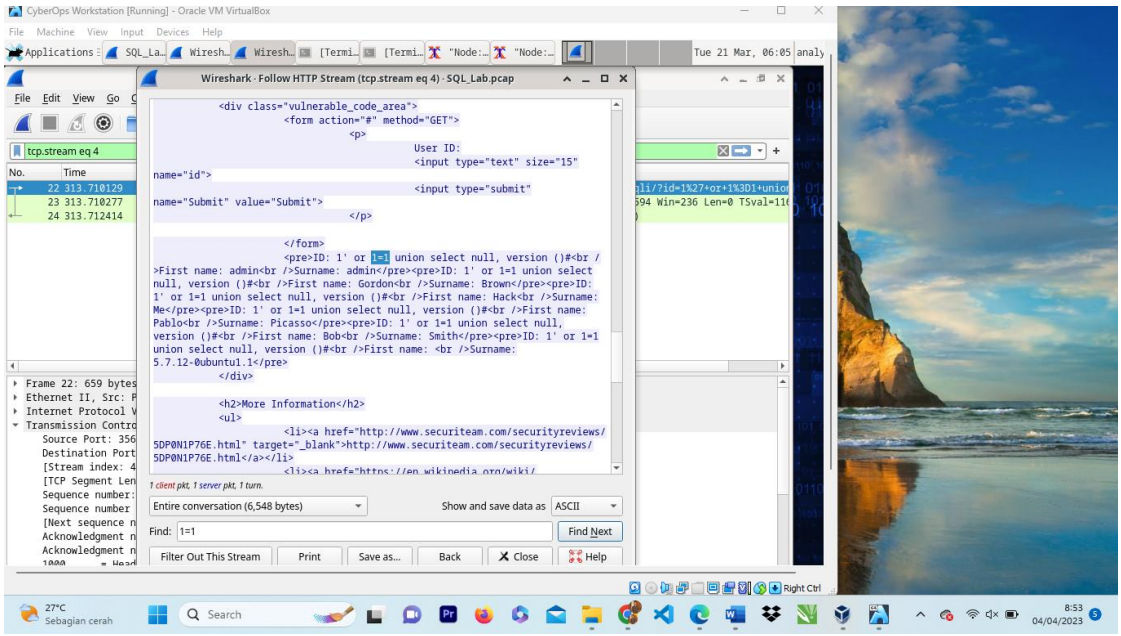
Tampilan setelah membuka file SQL.Lab.pcap

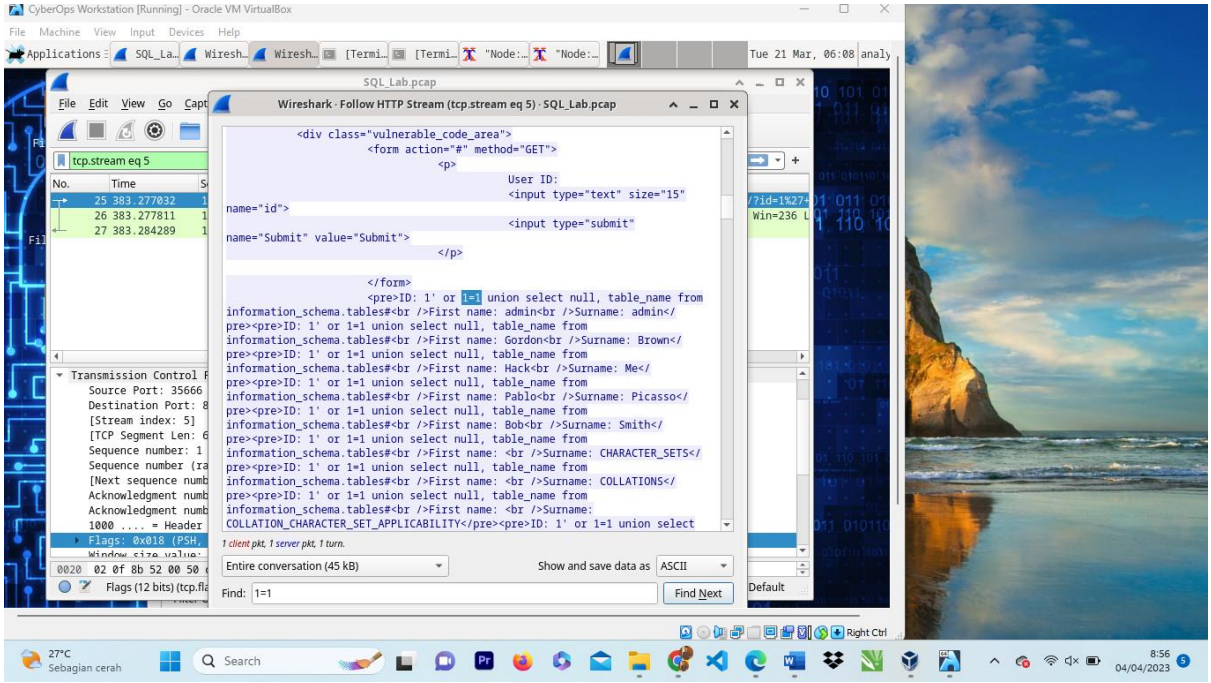


Melihat Serangan Injeksi SQL dengan cara klik kanan line yang ingin dilihat, pilih follow lalu HTTP Stream

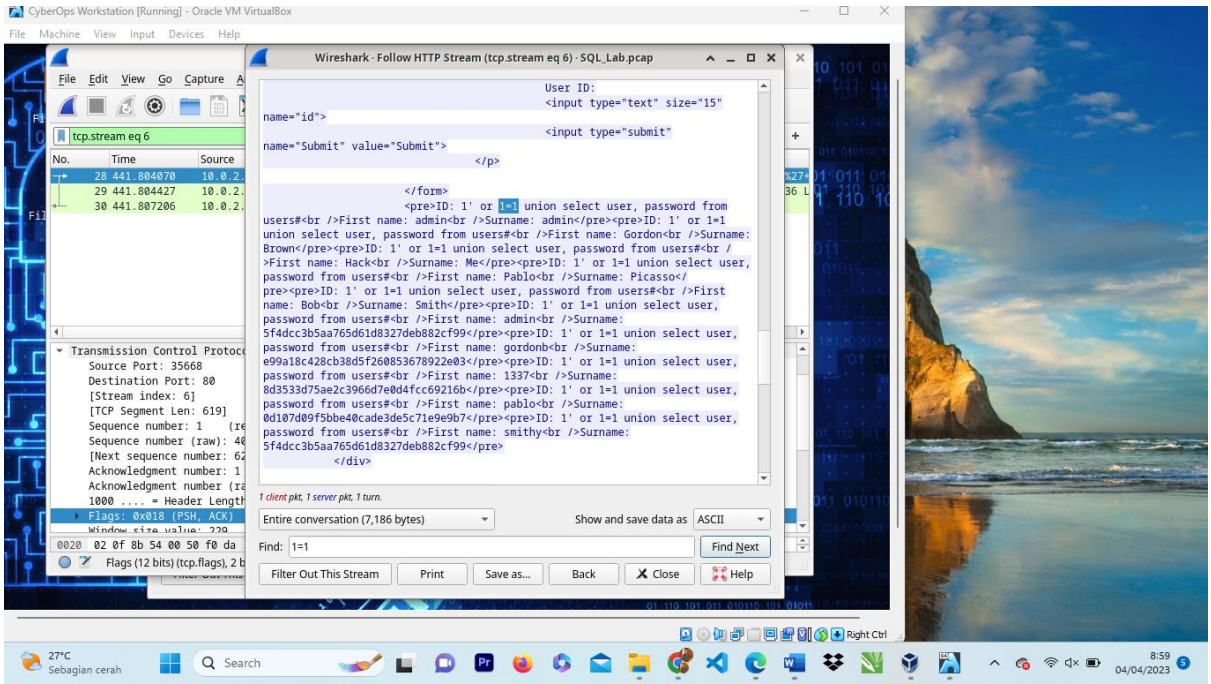
Kemudian Mencari 1=1 dengan mengisi find, setelah sudah pilih find text



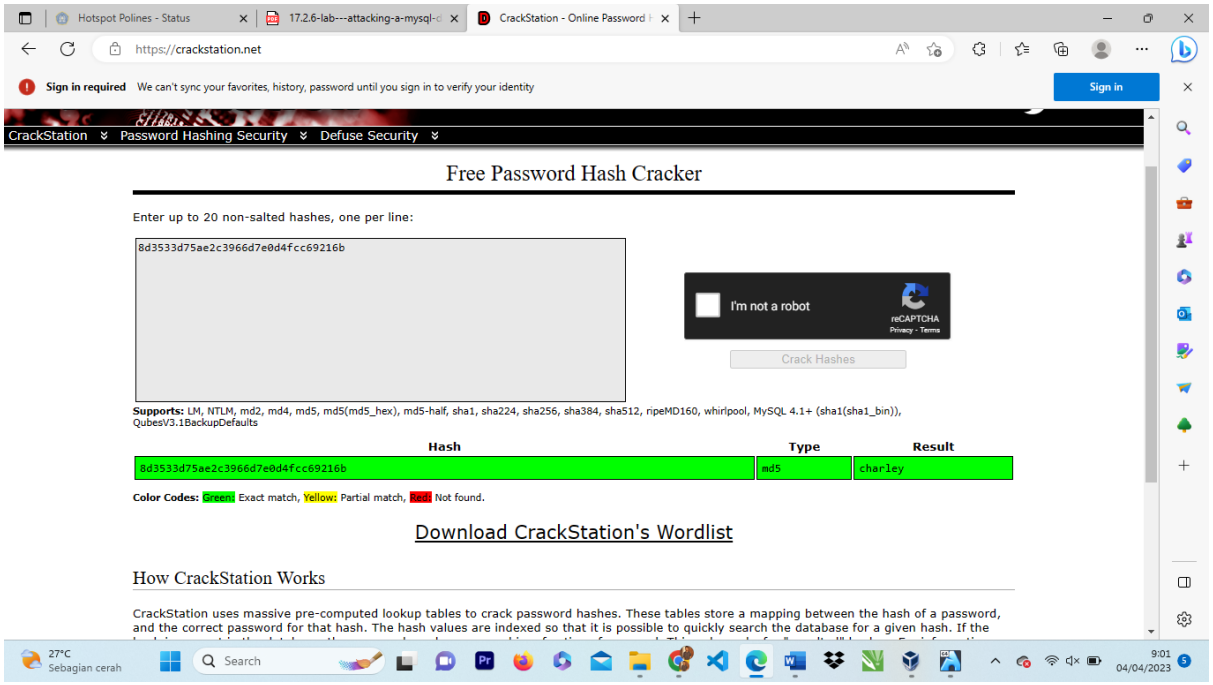




Setelah dicari menggunakan find text, pilih password yang ingin dicek dan di ambil.

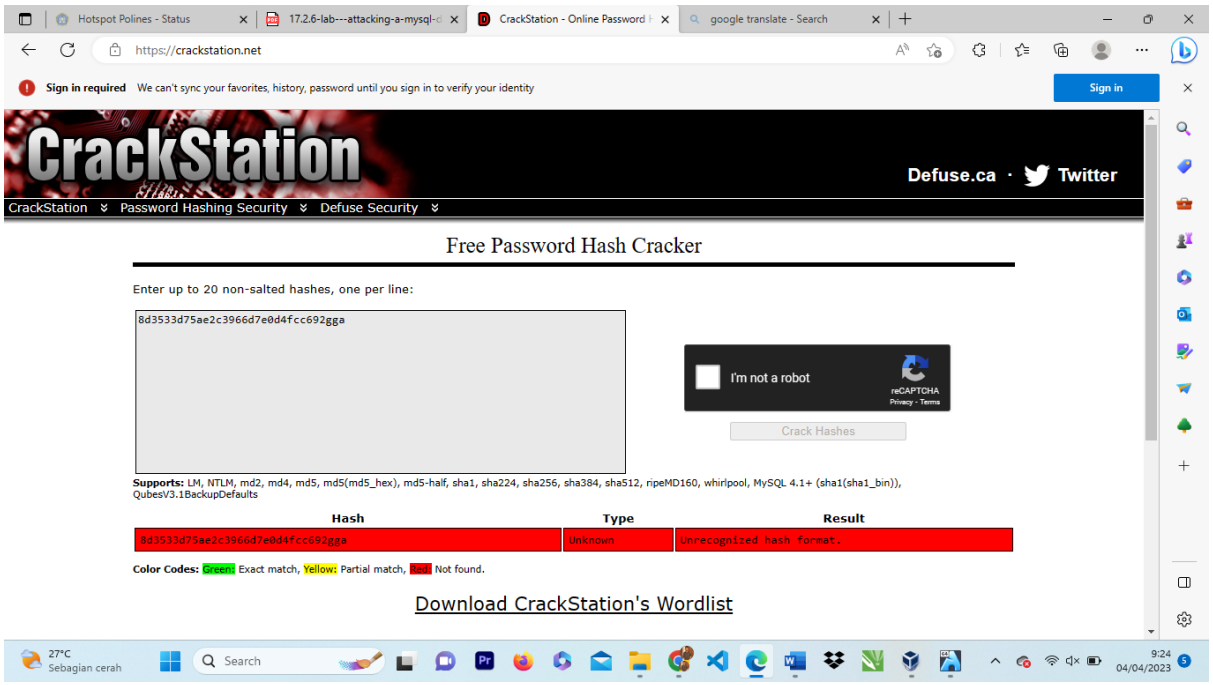


Setelah sudah mengambil password, masuk ke website crackstation.net, masukkan password dan setelah sudah pilih crack hashes.



Jika hijau brarti terlihat resultnya, resultnya adalah charlay.

Tampilan not found jika ditambahkan teks/karakter lain didalam password



1. Apa risiko memiliki platform yang menggunakan bahasa SQL?

1. SQL adalah bahasa yang banyak digunakan untuk mengelola basis data relasional, dan meskipun memiliki banyak manfaat, ia juga menimbulkan risiko tertentu saat digunakan dalam platform. Berikut adalah beberapa potensi risiko menggunakan SQL di platform:
2. Risiko keamanan: Serangan injeksi SQL dapat mengeksploitasi kerentanan dalam kode untuk mengeksekusi perintah SQL berbahaya pada database. Hal ini dapat menyebabkan akses tidak sah ke informasi sensitif atau bahkan hilangnya data sama sekali. Serangan injeksi SQL adalah jenis serangan berbasis web yang umum, dan dapat dicegah dengan praktik pengkodean yang tepat dan langkah-langkah keamanan.
3. Risiko kinerja: Kueri SQL yang dirancang dengan tidak benar dapat menyebabkan masalah kinerja, yang dapat berdampak pada keseluruhan kinerja platform. Penting untuk merancang kueri yang dioptimalkan untuk kasus penggunaan tertentu dan skema database untuk meminimalkan risiko masalah kinerja.
4. Risiko integritas data: Kueri SQL yang salah ditulis dapat menyebabkan masalah integritas data, seperti pembaruan atau penghapusan data yang salah. Penting untuk memastikan bahwa kueri ditulis dengan benar dan bahwa proses pengujian dan jaminan kualitas yang sesuai tersedia untuk mencegah jenis masalah ini.
5. Risiko skalabilitas: Saat platform tumbuh dan menangani lebih banyak data, basis data yang mendasarinya mungkin perlu diskalakan untuk menangani peningkatan beban. Hal ini dapat menantang dengan basis data berbasis SQL, dan mungkin memerlukan penggunaan alat atau teknik khusus untuk mempertahankan kinerja dan skalabilitas.

2. Jelajahi internet dan lakukan pencarian pada "mencegah serangan injeksi SQL". Apa saja 2 metode atau langkah yang dapat dilakukan untuk mencegah serangan injeksi SQL?

Ada beberapa metode dan langkah yang dapat diambil untuk mencegah serangan injeksi SQL, namun berikut adalah dua yang umum:

1. Gunakan Pernyataan yang Disiapkan/Kueri yang Diparameterisasi: Pernyataan yang disiapkan atau kueri yang diparameterisasi adalah metode penulisan kueri SQL yang memisahkan kode SQL dari masukan pengguna. Teknik ini memastikan bahwa input pengguna diperlakukan sebagai data dan bukan kode yang dapat dieksekusi, sehingga mencegah eksekusi kode SQL yang berbahaya.

Misalnya, alih-alih menyematkan input pengguna langsung ke kueri SQL, placeholder digunakan. Kemudian kueri SQL dan input pengguna dikirim ke database secara terpisah, dan sistem database memasukkan input pengguna ke dalam placeholder yang sesuai. Dengan cara ini, kode SQL dan input pengguna disimpan terpisah, dan sistem basis data dapat membedakan keduanya.

2. Validasi dan Sanitasi Input: Langkah penting lainnya untuk mencegah serangan injeksi SQL adalah validasi dan sanitasi input. Ini melibatkan memvalidasi dan membersihkan semua input pengguna untuk memastikannya

memenuhi kriteria yang diharapkan dan aman untuk digunakan. Validasi dan sanitasi input membantu menghapus kode atau karakter berbahaya apa pun yang mungkin disertakan dalam input pengguna.

Misalnya, memeriksa tipe data, panjang, dan format input, dan menggunakan ekspresi reguler atau metode lain untuk menghapus karakter yang tidak diinginkan yang dapat digunakan untuk menyuntikkan kode SQL ke dalam kueri. Dengan memvalidasi dan membersihkan input pengguna, Anda dapat mencegah serangan injeksi SQL dengan memastikan bahwa setiap input pengguna yang digunakan dalam kueri SQL aman dan terlindungi.