BAB V
PENUTUP

5.1 Simpulan

Berdasarkan hasil pelaksanaan penetration testing terhadap aplikasi sistem manajemen siswa yang dibangun menggunakan framework Python Flask, dapat disimpulkan beberapa poin penting sebagai berikut :

1. Aplikasi belum aman dari ancaman umum seperti *SQL Injection*. Pengujian pada fitur login menunjukkan bahwa sistem masih menggunakan *query SQL* yang rawan terhadap serangan, karena tidak ada proteksi seperti *prepared statements* atau validasi input yang memadai.
2. Kontrol akses belum diterapkan dengan baik. Pengujian terhadap fitur manajemen data siswa menunjukkan adanya kerentanan *Insecure Direct Object References (IDOR)*, yang memungkinkan pengguna mengakses atau memodifikasi data pengguna lain hanya dengan mengubah parameter *ID* di *URL*.
3. Fitur *upload file* tidak memiliki filter keamanan. Aplikasi memungkinkan file berbahaya seperti *.php atau .py* untuk diunggah ke server tanpa adanya pengecekan *MIME type*, isi file, atau sanitasi nama file. Hal ini membuka kemungkinan terjadinya serangan *Remote Code Execution (RCE)* melalui *backdoor*.
4. Manajemen sesi belum optimal. Setelah pengguna login, sesi tetap aktif tanpa batas waktu yang jelas, sehingga rentan jika ditinggalkan terbuka. Hal ini menunjukkan bahwa pengelolaan *session timeout* perlu diperkuat.

5.2 Saran

Dari hasil yang diperoleh selama pelaksanaan penetration testing, berikut beberapa saran yang dapat diberikan untuk perbaikan sistem;

1. Gunakan *prepared statement* dan validasi input. Untuk mencegah *SQL Injection*, pengembangan sistem sebaiknya menghindari penggunaan *query* mentah yang menerima input langsung dari pengguna, dan menggantinya dengan mekanisme *prepared statements* serta *filtering input* secara ketat.
2. Terapkan otorisasi berbasis *session* dan *ID*. Aplikasi harus memverifikasi setiap permintaan berdasarkan sesi pengguna dan memastikan bahwa *ID* yang dikirim memang milik pengguna yang sedang login. Ini akan mencegah eksploitasi *IDOR*.
3. Batasi jenis file yang dapat diunggah. Fitur *upload* harus menyaring file berdasarkan *MIME type*, membatasi hanya file yang dibutuhkan (misalnya *PDF*, *JPG*), serta menggunakan *secure_filename()* untuk menghindari *path traversal* atau injeksi nama file.