Jurnal Infosecure Vol.5 No.2, Oktober 2024
p-ISSN: 2087-1023
MTI-UNLA

Threat Modelling dan Penyusunan Kontrol Mitigasi pada Sistem Penilaian Ujian Praktikum Pemrograman Dasar SMK Informatika Sumedang

Apep Wahyudin[1], Hadi Prasetyo Utomo[2], Arief Zulianto[3]

Magister Teknik Informatika, Pascasarjana, Universitas Langlangbuana[1,2,3]

[1]apep@smkifsu.sch.id
[2]hadi@informatika.unla.ac.id
[3]madzul@unla.ac.id

Abstrak— Penelitian ini bertujuan untuk mengidentifikasi ancaman keamanan pada system penilaian ujian praktikum pemrograman dasar SMK Informatika Sumedang, memberikan penilaian peringkat risiko, serta menyusun langkah mitigasi untuk mengurangi dampak dari ancaman yang teridentifikasi menggunakan pendekatan *threat modelling*. Penelitian ini menggunakan model STRIDE untuk mengklasifikasikan ancaman, dan model DREAD untuk menilai peringkat risiko dari setiap ancaman. Dekomposisi aplikasi dilakukan menggunakan *Data Flow Diagram (DFD)* untuk memetakan aliran dan perubahan data saat data mengalir. Hasil kasifikasi ancaman menggunakan model STRIDE menunjukkan terdapat tujuh ancaman utama yang terdiri dari satu ancaman *spoofing*, tiga ancaman *tampering*, dua ancaman *denial of service*, dan satu ancaman *elevation of privilege*. Berdasarkan penilaian ancaman menggunakan metode DREAD, diketahui bahwa terdapat enam ancaman dengan peringkat tinggi dan satu ancaman dengan peringkat sedang. Berdasarkan hasil pemodelan ancaman menggunakan model STRIDE dan DREAD, kontrol mitigasi disusun untuk menjadi acuan dalam upaya meminimalkan risiko yang ditimbulkan oleh setiap ancaman dengan memprioritaskan ancaman dengan peringkat tinggi.

Kata kunci— *Threat modeling*, Sistem penilaian, Keamanan data, STRIDE, DREAD, Mitigasi.

I. PENDAHULUAN

Proses penilaian dan pemeriksaan hasil praktikum mata pelajaran pemrograman dasar di SMK Informatika Sumedang masih dilakukan secara manual oleh guru. Proses manual ini seringkali memakan waktu yang cukup lama karena setiap jawaban siswa harus diperiksa secara individual dengan teliti. Untuk mengatasi permasalahan di atas, SMK Informatika Sumedang mengembangakan system penilaian ujian praktikum berbasis web. Aplikasi ini dirancang untuk memberikan penilaian yang objektif, efektif dan efisien. Guru dapat memberikan soal pemrograman kepada siswa. Di sisi lain, siswa yang telah memiliki akun dapat mengerjakan soal pemrograman tersebut. Setelah soal-soal dikerjakan, siswa dapat mendapatkan nilai secara otomatis.

Penggunaan jaringan internet pada sistem penilaian ujian praktikum memunculkan potensi kerentanan baru, yakni berbagai gangguan yang berpotensi menghambat kelancaran operasional aplikasi tersebut. Berdasarkan Lanskap Keamanan Siber tahun 2023 yang dirilis oleh Badan Siber dan Sandi Negara [1], tercatat sekitar 327 kasus yang diduga sebagai insiden siber, dengan kategori dugaan insiden yang paling menonjol adalah kebocoran data. Sementara itu, BSSN telah menginformasikan adanya 189 kejadian perusakan tampilan situs web. Dari jumlah tersebut, mayoritas kasus terklasifikasi sebagai modifikasi tidak sah pada bagian tersembunyi dari laman web. Data di atas menunjukkan ancaman siber terhadap suatu sistem di dalam jaringan internet yang cukup tinggi. Untuk itu, diperlukan adanya desain keamanan dari sebuah sistem yang dikembangkan sehingga risiko ancaman dapat dimitigasi.

Sebelum dapat membuat desain keamanan dari sebuah sistem, pengembang diharuskan untuk menganalisis risiko serangan yang berpotensi terjadi terhadap sistem tersebut. Proses di atas bisa dilakukan dengan bantuan *Threat Modeling* [2]. *Threat Modeling* adalah suatu pendekatan sistematis yang ditujukan untuk mengenali potensi celah dan risiko keamanan, mengevaluasi tingkat keseriusan setiap potensi bahaya, serta menentukan urutan prioritas langkah-langkah pengamanan guna meinimalisasi risiko serangan terhadap sistem [3].

Untuk menganalisis ancaman, Microsoft mengembangan STRIDE sebagai metode untuk klasifikasi ancaman. Metode ini mengelompokkan ancaman menjadi enam kategori, yaitu: *Spoofing* (pencurian identitas), *Tempering* (modifikasi data), *Repudiation* (penyangkalan), *Information* Disclosure (pengeksposan informasi rahasia), *Denial of Service* (gangguan layanan), dan *Elevation of Privilege* (perolehan hak akses lebih). Guna memahami dampak yang ditimbulkan oleh ancaman tersebut, diperlukan serangkaian proses penilaian risiko terhadap ancaman tersebut menggunakan metode DREAD, yang juga dirancang oleh Microsoft. Metode ini mencakup penilaian *Damage potential* (potensi kerusakan), *Reproducibility* (reproduksibilitas), *Exploitability*

(eksploitasi), *Affected user* (pengguna terdampak), dan *Discoverability* (dapat ditemukan).

Pendekatan *threat modeling* dengan menggunakan model STRIDE dan DREAD sangat relevan dalam konteks ini. STRIDE membantu mengidentifikasi berbagai jenis ancaman yang berpotensi dihadapi oleh sistem, sementara DREAD digunakan untuk menilai tingkat risiko dari ancaman-ancaman tersebut. Dengan diadopsinya metode ini, ancaman terhadap sistem penilaian ujian praktikum pemrograman dasar di SMK Informatika Sumedang dapat dimitigasi.

II. METODE

Penelitian ini mengadopsi metode penelitian yang dikemukakan oleh Fruhlinger [4] yang terdiri dari pengumpulan data, dekomposisi aplikasi, klasifikasi ancaman, penilaian ancaman dan mitigasi ancaman.

A. Pengumpulan Data

Data pada penelitian ini didapatkan melalui dua metode yaitu studi literatur dan observasi. Dalam penelitian ini dilakukan tinjauan literatur untuk mengumpulkan, menelaah, dan meninjau referensi dari berbagai sumber seperti buku, jurnal, artikel, dan *website* terkait dengan pemodelan ancaman, model STRIDE, model DREAD, dan keamanan perangakat lunak

Sebagaimana dalam [5] bahwa observasi adalah metode pengumpulan data dengan cara melakukan pengamatan secara langsung oleh peneliti agar mampu memahami konteks data dalam keseluruhan situasi sosial sehingga dapat diperoleh data yang menyeluruh. Observasi tidak terbatas pada pengamatan terhadap manusia, melainkan juga terhadap proses kerja, gejala alam dan bila responden yang diamati tidak terlalu besar. Pada penelitian ini, observasi dilakukan untuk mengamati proses kerja Sistem Penilaian Ujian Praktikum Pemrograman Dasar di SMK Informatika Sumedang agar untuk mengidentifikasi ancaman yang berpotensi terjadi terhadap aplikasi tersebut.

B. Dekomposisi Aplikasi

Tahap pertama dalam proses *threat modeling* adalah mempelajari aplikasi dan bagaimana aplikasi tersebut berinteraksi dengan entitas eksternal. Langkah ini penting untuk memahami penggunaan aplikasi, mengidentifikasi potensi celah keamanan, menemukan aset yang mungkin menjadi target penyerang, serta menilai tingkat kepercayaan terkait akses yang diberikan aplikasi kepada pihak luar.

Dikutip dari [6], salah satu teknik untuk dekomposisi sebuah aplikasi adalah dengan membangun diagram alir data *(data flow diagram)*. Sebagaimana dikatakan oleh [7] STRIDE memulai proses pemodelan ancaman dengan menampilkan DFD. Selanjutnya, laporan ancaman dihasilkan berdasarkan DFD ini. Dengan memetakan aliran data dari satu komponen ke komponen lainnya di dalam sebuah sistem, DFD membantu mengungkapkan titik-titik kritis di mana potensi ancaman dapat muncul. Dalam proses ini, setiap entitas, aliran data, penyimpanan data, dan proses diidentifikasi dan dianalisis untuk menemukan celah keamanan yang mungkin dimanfaatkan oleh penyerang.

C. Klasifikasi Ancaman

Setelah ancaman teridentifikasi melalui analisis *data flow diagram*, ancaman tersebut kemudian diklasifikasikan. Metode STRIDE yang dikembangkan oleh Microsoft dipilih sebagai metode untuk melakukan klasifikasi ancaman. STRIDE merupakan singkatan dari beberapa kategori, yaitu *Spoofing*, *Tampering*, *Repudiation*, *Information disclosure*, *Denial of service*, dan *Elevation of privilege*. Dikatakan oleh [8], STRIDE merupakan metode yang efektif untuk mengidentifikasi risiko keamanan dalam sistem. Ini adalah pilihan yang baik bagi organisasi yang baru memulai dalam bidang keamanan, karena mudah diterapkan dan dipahami. Hal ini sesuai dengan kondisi SMK Informatika Sumedang yang baru melakukan pemodelan ancaman untuk pertama kalinya.

D. Penilaian Ancaman

Model DREAD diterapkan untuk mengevaluasi, membandingkan, dan menentukan prioritas tingkat risiko yang diakibatkan oleh setiap ancaman. DREAD adalah akronim dari lima kategori risiko, yaitu *Damage potential*, *Reproducibility*, *Exploitability*, *Affected users*, dan *Discoverability*. Dikemukakan oleh [7], bahwa DREAD memiliki kemampuan untuk mengukur tingkat dampak keamanan dari setiap ancaman secara kuantitatif.

Dikutip dari [4], penilaian untuk peringkat tinggi dinilai dengan angka 3, peringkat sedang dinilai dengan angka 2, dan peringkat rendah dinilai dengan angka 1. Nilai total dari setiap kategori ancaman menentukan tingkat keparahannya. Masing-masing kategori memiliki nilai minimal 1 dan maksimal 3. Jika semua kategori mendapat nilai 1, maka totalnya adalah 5. Sebaliknya, jika semua kategori mendapat nilai maksimal 3, maka totalnya adalah 15. Dengan metode ini, dari 5 kategori ancaman, diperoleh rentang nilai total antara 5 dan 15. Hasil total perhitungan menghasilkan rentang nilai sebagai berikut:

TABEL I
RENTANG PENILAIAN RISIKO

Rentang Nilai	Reingkat Risiko
12-15	Tinggi
8-11	Sedang
5-7	Rendah

E. Mitigasi

Langkah ini adalah rekomendasi mekanisme kontrol yang dirancang untuk mencegah dan mengurangi dampak dari ancaman yang telah diidentifikasi. Setelah menentukan peringkat risiko setiap ancaman, peringkat risiko dapat diurutkan dari yang tertinggi hingga terendah. Dengan

demikian, upaya mitigasi dapat diprioritaskan pada ancaman yang memiliki risiko paling tinggi.

III. HASIL DAN PEMBAHASAN

A. *External Dependency*

Dependensi eksternal adalah elemen di luar kode aplikasi yang dapat menimbulkan potensi ancaman terhadap aplikasi. Pada sistem penilaian ujian praktikum pemrograman dasar, dependensi eksternal didokumentasikan dengan memberikan nomor unik dan deskripsi untuk masing-masing dependensi. Dependensi eksternal disajikan dalam table berikut ini:

TABEL II
DEPENDENSI EKSTERNAL

ID	Deskripsi
1	Sistem penilaian praktikum pemrograman dasar beroperasi pada *server hosting* dengan Apache sebagai *web server*
2	*Server* basis data menggunakan *MySQL*
3	Koneksi antara *server web* dan *database server* menggunakan jaringan internet
4	Protokol komunikasi menggunakan TLS (*Transport Layer Security*)

B. *Level Kepercayaan*

Dalam konteks *threat modeling*, level kepercayaan mengacu pada tingkat kepercayaan yang diberikan kepada entitas atau komponen tertentu dalam sistem berdasarkan hak akses dan kemampuan yang dimilikinya. Level kepercayaan ini membantu dalam mengidentifikasi dan mengelompokkan berbagai entitas berdasarkan risiko yang dibawanya, serta menentukan seberapa besar akses dan kontrol yang dimiliki dalam sistem. Level kepercayaan yang telah diidentifikasi dicatat diberikan nomor ID, nama entitas, dan deskripsi dari masing-masing level. Tabel di bawah ini menunjukkan daftar level kepercayaan yang berhasil diidentifikasi.

TABEL III
DEPENDENSI EKSTERNAL

ID	Nama	Deskripsi
LK1	Pengguna web anonim	Seseorang yang mengakses aplikasi namun tidak memiliki hak akses untuk login
LK2	Pengguna dengan kredensial login valid	Seseorang yang mengakses aplikasi dan telah login menggunakan kredensial login yang valid
LK3	Pengguna dengan kredensial login tidak valid	Seseorang yang mengakses aplikasi dan mencoba untuk login menggunakan kredensial login yang tidak valid
LK4	Siswa	Seseorang siswa yang dapat mengerjakan ujian praktikum di dalam sistem
LK5	Guru	Seseorang tenaga pendidik mata pelajaran pemrograman dasar di SMK Informatika Sumedang
LK6	Ketua Program Keahlian	Seseorang dari pihak program keahlian Rekayasa Perangkat Lunak yang memiliki kewenangan tertentu
LK7	Administrator *database server*	Seseorang administrator yang memiliki hak akses penuh terhadap *database server* aplikasi
LK8	Administrator web aplikasi	Penanggungjawab yang memiliki hak akses secara keseluruhan untuk mengatur jalannya aplikasi.
LK9	Proses *web server*	Entitas yang dijalankan oleh *web server* sebagai kode spesifik dan memiliki kemampuan untuk melakukan proses autentikasi langsung ke *database server*.
LK10	Database read/write user	Akun pengguna *database* yang memiliki hak akses baca dan tulis

C. *Identifikasi Aset*

Aset merupakan hal yang dimiliki oleh aplikasi baik fisik maupun non-fisik yang menjadi sasaran penyerang. Pada dasarnya, aset adalah alasan penyerang melakukan aksi serangan.

TABEL IV
ASET APLIKASI

ID	Nama	Deskripsi
A1	Pengguna layanan aplikasi	Aset yang berkaitan dengan pengguna
A.1.1	Kredensial Login	Informasi autentikasi pengguna untuk masuk ke dalam sistem aplikasi
A.1.2	Data pribadi siswa	Informasi pribadi mengenai siswa
A.1.3	Data pribadi guru	Informasi pribadi mengenai guru
A.2	Sistem	Aset yang berkaitan dengan sistem
A.2.1	Data ujian	Detail ujian termasuk soal ujian, jawaban ujian dan nilai ujian

ID	Nama	Deskripsi
A.2.2	Ketersediaan aplikasi	Aplikasi penilaian ujian harus tersedia dan dapat diakses oleh pengguna selama 24 jam
A.2.3	Ketersediaan database	Database aplikasi harus tersedia dan dapat diakses oleh pengguna selama 24 jam
A.2.4	Eksekusi kode program aplikasi	Kemampuan untuk menjalankan kode pemrograman di server web
A.2.5	Eksekusi perintah *SQL read/write database*	Kemampuan untuk menjalankan perintah SQL baca dan tulis bagi pengguna yang telah berhasil login ke dalam aplikasi

D. Identifikasi Ancaman

Berdasarkan hasil analisis data flow diagram pada sistem penilaian ujian praktikum pemrograman dasar, terdapat enam ancaman yang berhasil teridentifikasi. Setiap ancaman ini diberi kode T (threat) diikuti nomor urut dari setiap ancaman.

Ancaman T1 berpotensi terjadi ketika siswa memasukkan kode program berbahaya ke dalam *form* untuk menjawab soal pemrograman. Kode program yang dimaksud bisa berupa kode program destruktif seperti perintah penghapusan file atau kode program yang menjalankan perulangan tanpa henti *(infinite looping)*. Hal ini terjadi karena kode yang ditulis oleh siswa akan dibuat menjadi sebuah *file* php dan disimpan di dalam penyimpanan *server*. Setelah itu, sistem akan menjalankan sebuah *file* testcase.php untuk menguji *output* dari kode yang ditulis oleh siswa tersebut.

Ancaman T2 berpotensi terjadi ketika pengguna menuliskan perintah SQL pada *form input* sehingga dapat memodifikasi perintah SQL yang dijalankan oleh *server*. Serangan ini bisa terjadi pada *form login* maupun *form input* yang digunakan oleh siswa, guru, dan ketua program keahlian.

Ancaman T3 dideskripsikan sebagai ancaman yang berasal dari pengunggahan *file* lokal komputer ke dalam *server* aplikasi. Ancaman ini teridentifikasi pada *form* pengunggahan data siswa yang dapat dilakukan oleh Ketua Program Keahlian. Pada *form* ini, data siswa diunggah menggunakan file berekstensi .xlsx (Microsoft Excel) oleh pengguna. Sistem aplikasi kemudian akan memecah setiap baris yang ada di dalam *file* dan menyimpan data pada baris tersebut ke dalam tabel siswa di *database*. Ancaman T3 berpotensi terjadi jika *file* yang diunggah adalah file dengan ekstensi yang lain, misalnya *file* berisi kode program yang kemudian dapat dijalankan di dalam *server* aplikasi.

Ancaman T4 dideskripsikan sebagai pemalsuan identitas pengguna yang sah dengan mencoba semua kemungkinan kombinasi kata sandi secara berulang hingga menemukan yang benar *(brute force attack).* Serangan ini berpotensi terjadi pada *form login* yang tersedia untuk lima level kepercayaan yaitu (LK1) Pengguna web anonim, (LK3) Pengguna dengan kredensial login tidak valid, (LK4) Siswa, (LK5) Guru, dan (LK6) Ketua Program Keahlian.

Ancaman T5 terjadi ketika pengguna memodifikasi parameter dalam URL aplikasi untuk mengakses data yang tidak sah atau mengubah perilaku aplikasi. Sistem penilaian ujian praktikum menggunakan parameter pada url untuk mengakses data tertentu. Misalnya untuk mengakses soal dengan id_soal 2, maka URL aplikasi diberi parameter &id_soal=2. Parameter tersebut kemudian akan dimasukkan ke dalam perintah SQL yaitu:

*"SELECT * FROM soal WHERE id_soal='$id_soal'"*

Melalui mekanisme di atas, seorang penyerang dapat menambahkan parameter tambahan untuk menyerang sistem sehingga URL berubah menjadi:

?page=soal&id_soal=2; DROP table soal. Parameter URL tersebut akan mengubah pernyataan SQL menjadi *"SELECT * FROM soal WHERE id_soal='$id_soal'; DROP table soal;".* Saat URL di atas diakses, maka sistem akan menjalankan perintah untuk menghapus tabel soal dari database.

Ancaman T6 berpotensi terjadi ketika pengguna menuliskan kode JavaScript berbahaya melalui form input ke halaman web yang sah. Kode ini dapat digunakan untuk mencuri informasi pengguna, mengubah tampilan aplikasi, hingga menyebarkan *malware*. Ancaman T6 bisa dilakukan salah satunya pada form input soal yang bisa diakses oleh pengguna guru.

Ancaman T7 berpotensi terjadi ketika penyerang mengirimkan banyak *request* sehingga membanjiri *bandwidth* dengan maksud memperlambat kinerja sistem (*DoS attack*). Berdasarkan hasil pengujian serangan DoS menggunakan aplikasi slowhttptest terhadap domain aplikasi, serangan DoS berhasil membanjiri lalu lintas data sehingga mengakibatkan domain tidak dapat diakses.

E. Klasifikasi Ancaman

Klasifikasi ancaman dilaksanakan untuk mengelompokkan jenis-jenis ancaman yang diidentifikasi sebagai potensi yang dapat terjadi pada sistem penilaian ujian praktikum pemrograman dasar. Ancaman tersebut kemudian dikelompokkan berdasarkan standarisasi STRIDE untuk menentukan jenis ancaman. Klasifikasi ancaman hasil identifikasi secara lengkap disajikan pada tabel berikut ini.

TABEL V
KLASIFIKASI ANCAMAN

ID	Deskripsi Ancaman	Kategori STRIDE
T1	Pengguna Siswa menuliskan kode program berbahaya melalui form jawaban ujian praktikum dan menjalankannya melalui fitur *Run Code*.	D
T2	Pengguna menuliskan pernyataan SQL melalui input pengguna untuk	T

ID	Deskripsi Ancaman	Kategori STRIDE
	mengakses, mengubah atau menghapus data di database secara tidak sah.	
T3	Pengguna dapat menyertakan file lokal pada server web, sehingga mereka dapat menjalankan kode mereka sendiri di server.	E
T4	Penyerang memalsukan identitas pengguna yang sah dengan mencoba semua kemungkinan kombinasi kata sandi secara berulang hingga menemukan yang benar (*brute force attack)*.	S
T5	Penyerang memodifikasi parameter dalam URL untuk mengakses data yang tidak sah atau mengubah perilaku aplikasi.	T
T6	Penyerang dapat menyuntikkan kode JavaScript berbahaya melalui form input ke dalam halaman web yang sah. Kode ini dapat digunakan untuk mencuri informasi pengguna, mengendalikan browser pengguna, atau menyebarkan malware (*Cross-Site Scripting*).	T
T7	Penyerang mengirimkan banyak request sehingga membanjiri bandwidth dengan maksud memperlambat kinerja sistem.	D

F. Penilaian Ancaman

Tahapan penilaian ancaman dilakukan menggunakan pendekatan DREAD. Proses penilaian ini bertujuan untuk mengetahui ranking dari setiap ancaman sehingga dapat diketahui prioritasnya untuk penyusunan langkah-langkah mitigasi. Hasil penilaian disajikan dalam table berikut ini:

TABEL VI
PENILAIAN ANCAMAN

ID	LK	D	R	E	A	D	TOTAL	RISIKO
T1	LK4	2	3	2	1	2	10	Sedang
T1	LK9	3	3	2	3	2	13	Tinggi
T2	LK1	3	3	2	3	2	13	Tinggi
T2	LK2	3	3	2	3	2	13	Tinggi
T2	LK3	3	3	2	3	2	13	Tinggi
T2	LK4	3	3	2	3	2	13	Tinggi
T2	LK5	3	3	2	3	2	13	Tinggi
T2	LK6	3	3	2	3	2	13	Tinggi
T3	LK6	3	3	1	3	1	11	Tinggi
T4	LK1	2	3	3	2	2	12	Tinggi
T4	LK3	2	3	3	2	2	12	Tinggi
T4	LK4	2	3	3	1	2	11	Tinggi
T4	LK5	3	3	3	3	2	14	Tinggi
T4	LK6	3	3	3	3	2	14	Tinggi
T5	LK2	3	3	2	3	2	13	Tinggi
T5	LK4	2	3	2	2	2	11	Sedang
T5	LK5	3	3	2	3	2	13	Tinggi
T5	LK6	3	3	2	3	2	13	Tinggi
T6	LK4	2	3	1	2	1	9	Sedang
T6	LK5	3	3	1	3	1	11	Sedang
T6	LK6	3	3	1	3	1	11	Sedang
T7	LK9	3	3	2	3	2	13	Tinggi
T7	LK10	3	3	2	3	2	13	Tinggi

G. Mitigasi

Kontrol mitigasi disusun untuk dijadikan acuan dalam mengurangi risiko serta dampak yang ditimbulkan dari setiap ancaman. Penyusunan kontrol ini dilaksanakan berdasarkan hasil penilaian sebagai landasan prioritas penganganan ancaman. Penyusunan langkah-langkah mitigasi ancaman pada sistem penilaian ujian praktikum pemrograman dasar disajikan sebagai berikut.

Ancaman T1 dapat dimitigasi menggunakan dua langkah mitigasi, yaitu pembatasan waktu eksekusi kode dan penyusunan daftar hitam *(blacklist)* perintah-perintah yang berbahaya. Pembatasan waktu eksekusi kode bertujuan untuk mencegah penggunaan sumber daya yang berlebihan oleh perintah yang mungkin disalahgunakan, sehingga mengurangi risiko terjadinya serangan *denial-of-service (DoS)* atau serangan lainnya yang memanfaatkan eksekusi kode yang tidak terkendali.

Penyusunan daftar hitam (*blacklist)* memungkinkan sistem untuk memblokir perintah-perintah yang telah diketahui berbahaya, sehingga mencegah eksekusi instruksi-instruksi yang dapat merusak atau mengakses data sensitif. Sebelum kode program dieksekusi, sistem akan mencari perintah-perintah berbahaya berdasarkan daftar yang telah disusun sebelumnya. Jika salah satu perintah berbahaya ditemukan, maka kode akan segera menampilkan pesan kesalahan yang memberitahu pengguna bahwa eksekusi gagal karena kode mereka mengandung perintah berbahaya. Setelah menampilkan pesan tersebut, eksekusi dihentikan menggunakan *exit()* untuk mencegah kode berbahaya dijalankan.

Untuk memitigasi ancaman T2, digunakan dua langkah mitigasi yaitu validasi input pengguna dan penggunaan *Prepared Statements* dan *Parameterized Queries.* Validasi input pengguna dilakukan dengan memastikan bahwa data yang diterima oleh aplikasi sesuai dengan format dan jenis yang diharapkan, sehingga dapat mencegah data berbahaya atau tidak sah masuk ke dalam sistem. Selain itu, penggunaan *Prepared Statements* dan *Parameterized Queries* merupakan teknik yang efektif dalam mencegah serangan SQL Injection, karena query SQL dan data pengguna diproses secara terpisah, menghindari eksekusi perintah SQL yang tidak diinginkan [9].

Ancaman T3 dapat dimitigasi menggunakan validasi ekstensi file, penyimpanan file di direktori sementara serta pembatasan ukuran file yang dapat diunggah. Validasi ekstensi file memastikan bahwa hanya jenis file xlsx yang diizinkan yang dapat diunggah, sehingga mencegah file berbahaya atau tidak sesuai masuk ke sistem. Penyimpanan file di direktori sementara memberikan waktu bagi sistem

untuk melakukan pengecekan lebih lanjut terhadap file yang diunggah sebelum dipindahkan ke direktori utama, sehingga menambah lapisan keamanan dalam proses pengunggahan. Selain itu, pembatasan ukuran file mencegah serangan yang memanfaatkan file berukuran besar untuk membebani sistem atau memicu *denial of service (DoS)*.

Ancaman T4 dimitigasi dengan cara pembatasan jumlah percobaan login, penggunaan CAPTCHA dan penerapan kebijakan kata sandi yang kuat. Pembatasan jumlah percobaan login bertujuan untuk mencegah serangan brute force dengan membatasi berapa kali pengguna dapat mencoba memasukkan kata sandi sebelum akun terkunci sementara atau memerlukan tindakan verifikasi tambahan. Penggunaan CAPTCHA menambahkan lapisan keamanan dengan memastikan bahwa upaya login dilakukan oleh manusia, bukan oleh *bot* atau skrip otomatis. Selain itu, penerapan kebijakan kata sandi yang kuat, seperti mewajibkan kombinasi karakter khusus, huruf besar, dan angka, serta pembaruan kata sandi secara berkala, membantu mengurangi risiko akses tidak sah yang diakibatkan oleh kata sandi yang lemah atau mudah ditebak.

Mitigasi ancaman T5 dilakukan menggunakan validasi dan sanitasi semua *input* yang diterima dari URL pada sisi *server*, guna memastikan bahwa setiap data yang diolah telah memenuhi standar keamanan yang ketat dan tidak mengandung karakter atau string yang dapat digunakan untuk mengeksploitasi kerentanan aplikasi. Validasi bertujuan untuk memverifikasi bahwa *input* sesuai dengan format dan tipe data yang diharapkan, sementara sanitasi menghapus atau menggantikan karakter-karakter berbahaya sebelum diproses lebih lanjut. Selain itu, penggunaan fungsi *addslashes* untuk parameter id menambahkan karakter *backslash* (`\`) sebelum tanda kutip tunggal atau ganda dalam string, yang berfungsi sebagai lapisan perlindungan tambahan terhadap serangan *SQL injection*.

Langkah mitigasi ancaman T6 dilakukan dengan validasi input, yang berfungsi untuk memastikan bahwa data yang diterima sesuai dengan format dan tipe data yang diharapkan, sehingga mengurangi risiko terjadinya eksploitasi dari input yang tidak valid. Selain itu, penggunaan sanitasi diterapkan untuk menghilangkan karakter atau tag yang tidak diinginkan, termasuk elemen-elemen yang berpotensi digunakan untuk serangan *Cross-Site Scripting (XSS)* atau injeksi kode. Sebagai langkah tambahan, fungsi *htmlspecialchars* digunakan pada data yang diinput oleh pengguna sebelum ditampilkan kembali ke antarmuka pengguna.

Mitigasi ancaman T7 dilakukan dengan cara membatasi jumlah permintaan yang dapat dilakukan oleh pengguna atau alamat IP dalam jangka waktu tertentu, yang dikenal sebagai teknik *rate limiting.* Teknik ini efektif dalam mencegah penyerang melakukan serangan *Denial of Service* (DoS) dengan cara mengirimkan permintaan berulang dalam jumlah besar ke server. Selain itu, dalam kasus di mana alamat IP terdeteksi melakukan aktivitas mencurigakan atau serangan DoS, langkah lanjutan dilakukan dengan memblokir alamat IP tersebut melalui fitur keamanan yang tersedia di Cpanel.

IV. SIMPULAN

Berdasarkan hasil dekomposisi aplikasi dan *threat modeling* menggunakan model STRIDE, diidentifikasi tujuh ancaman utama yang berpotensi terjadi terhadap sistem penilaian ujian praktikum pemrograman dasar SMK Informatika Sumedang. Ancaman-ancaman tersebut adalah penulisan kode berbahaya oleh siswa *(denial of service)*, SQL *Injection* pada *form* login *(tampering)*, penyertaan *file* lokal berbahaya pada *server web (Elevation of privilege)*, *brute force attack (spoofing)*, modifikasi URL secara ilegal *(tampering)*, *cross-site scripting* (XSS) *(tampering)*, dan serangan *Denial of Service* (DoS *attack*). Berdasarkan penilaian ancaman menggunakan metode DREAD menghasilkan informasi mengenai nilai dari setiap ancaman. Berdasarkan penilaian tersebut, diketahui bahwa terdapat enam ancaman dengan peringkat tinggi dan satu ancaman dengan peringkat sedang. Berdasarkan hasil pemodelan ancaman menggunakan model STRIDE dan DREAD, kontrol mitigasi disusun yang terdiri dari pembatasan waktu eksekusi kode dan penyusunan daftar hitam perintah berbahaya, penggunaan *prepared statement* dan *parameterized queries*, validasi ekstensi *file*, penggunaan CAPTCHA dan pembatasan percobaan login, penerapan sanitasi dan validasi input, serta penerapan *rate limiting*.

REFERENSI

[1] Badan Siber dan Sandi Nasional. (2023). *Lanskap Keamanan Siber Indonesia 2023*. [Online]. Available: https://www.bssn.go.id/wp-content/uploads/2024/03/Lanskap-Keamanan-Siber-Indonesia-2023.pdf.

[2] Obara, Sebastian. (2023). *Why Threat Modeling is Important*. [Online]. Available: https://www.securing.pl/en/why-threat-modeling-is-important.

[3] EC-Council. (2020). *What is Stride Methodology in Threat Modeling?* [Online]. Available: https://blog.eccouncil.org/what-is-stride-methodology-in-threat-modeling.

[4] Fruhlinger, J. (2020). *Threat modeling explained: A process for anticipating cyber attacks*. [Online]. Available: https://www.csoonline.com/article/3537370/threat-modeling-explained-a-process-for-anticipating-cyber-attacks.html.

[5] Sugiyono (2020). *Metode Penelitian Kualitatif*. Bandung: Alfabeta.

[6] OWASP. (2021). *Threat Modeling Process*. [Online]. Available: https://owasp.org/www-community/Threat_Modeling_Process.

[7] Das, P.; Asif, M.R.A.; Jahan, S.; Ahmed, K.; Bui, F.M.; Khondoker, R. (2024). *STRIDE-Based Cybersecurity Threat Modeling, Risk Assessment and Treatment of an In-Vehicle Infotainment System*. *Vehicles* 2024, *6*, 1140-1163. https://doi.org/10.3390/vehicles6030054.

[8] Hiremath, O. (2024). *Comparison of STRIDE, DREAD & PASTA*. Diakses pada 27 Agustus 2024, dari https://www.softwaresecured.com/post/comparison-of-stride-dread-pasta#what-is-threat-modeling-how-does-it-relate-to-penetration-testing.

[9] Holy, T. (2023). *Secure Coding Guidelines for PHP*. Brno: Brno University of Technology