Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0893
p-ISSN: 2089-3787

Analisis Kebocoran Data Sistem Informasi Pendaftaran Mahasiswa Baru Dari Serangan *SQL Injection*

Ilham Idfiana[1*], Deni Ahmad Jakaria[2]
Teknik Informatika, STMIK DCI, Tasikmalaya, Indonesia
*e-mail *Corresponding*: ilham.idfiana@transtrack.id

Abstract

The new student registration information system at educational institutions can support administration as a process for searching for quality prospective students. Universities use technology in the form of websites, to provide services to students in an effort to facilitate access to services. However, personal data leaks can occur, one of which is through SQL Injection cyber attacks. This research aims to provide a risk rating assessment based on data leaks in SQL Injection attacks that target new student registration systems with low security. The risk ranking data method refers to FERPA (Family Educational Rights and Privacy Act) and NIK (Population Identification Number) as sensitive data, risk assessment test results refer to CVSS V3 (Common Vulnerability Scoring System) and statistical values use the min-max difference method The level of risk when tested includes site A showing a low indication because when testing using the Havij application it did not show a response when SQL Injection penetration was carried out, the results of site A were said to be low because there was no data that experienced a data leak, while site B was said to be medium with a scale value of 5.76 out of 10, because there was a data leak when performing SQL Injection penetration.
Keywords: *Information System; SQL Injection; Information Security*

Abstrak

Sistem informasi pendaftaran mahasiswa baru pada institusi pendidikan dapat menunjang administrasi sebagai proses untuk pencarian calon mahasiswa yang berkualitas. Perguruan tinggi menggunakan teknologi berupa situs web, untuk memberikan layanan kepada mahasiswa dalam upaya memudahkan akses layanan. Namun, kebocoran data pribadi dapat terjadi, salah satunya melalui serangan siber *SQL Injection*. Penelitian ini bertujuan untuk memberikan penilaian *risk rating* berdasarkan kebocoran data dalam insiden serangan SQL Injection yang menargetkan sistem pendaftaran mahasiswa baru dengan keamanan yang rendah. Metode *Risk ranking* data tersebut mengacu pada FERPA (*Family Educational Rights and Privacy Act*) dan NIK (Nomor Induk Kependudukan) sebagai data sensitif, hasil pengujian penilaian risiko mengacu pada CVSS V3 (*Common Vulnerability Scoring System*) dan nilai statistik menggunakan metode min-max perbedaan tingkatan risiko ketika diuji diantaranya situs A menunjukkan indikasi rendah karena pada saat pengujian menggunakan aplikasi *Havij* tidak menunjukkan respon ketika dilakukan penetrasi *SQL Injection*, hasilnya situs A dikatakan rendah karena tidak ada data yang mengalami kebocoran data, sedangkan situs B dikatakan medium dengan skala nilai 5.76 dari 10, karena terdapat data yang mengalami kebocoran ketika melakukan penetrasi *SQL Injection.*
Kata kunci: *Sistem Informasi; SQL Injection; Keamanan Informasi*

1. Pendahuluan

Sistem informasi pendaftaran mahasiswa baru atau (PMB) yang pada setiap institusi pendidikan dalam perguruan tinggi ialah dapat menunjang administrasi sebagai proses untuk calon mahasiswa baru[1]. Banyak perguruan tinggi saat ini menggunakan teknologi berupa situs website untuk melayani calon mahasiswa, sehingga memudahkan akses bagi yang ingin mendaftar pada suatu universitas [2]. Jalur akses dapat melalui perangkat berbasis *mobile* maupun perangkat komputer portable. Pada beberapa Perguruan Tinggi, panitia atau operator pendaftaran mahasiswa baru belum dapat melakukan pengontrolan data mahasiswa secara

maksimal, seperti pada [3] dan [4], dimana penerimaan mahasiswa baru masih menggunakan manual, yaitu calon mahasiswa mendatangi perguruan tinggi untuk mengumpulkan formulir pendaftaran beserta persyaratan administrasi. Dengan adanya sistem informasi pendaftaran mahasiswa baru berbasis website, mampu menyediakan akses bagi yang ingin melakukan pendaftaran ke salah satu institusi pendidikan tinggi yang terhubung secara daring [5].

Pada saat ini kebocoran data terutama di pendidikan sering terjadi salah satunya mengalami kebocoran data seperti NIK (Nomor Induk Kependudukan)[6], kebocoran data pribadi ini terjadi karena salah satu pihak yang tidak bertanggung jawab melakukan tindakan illegal salah satunya serangan siber *SQL Injection* [7], [8]. Pada penelitian yang dilakukan oleh [9] situs sistem informasi tersebut dalam proses pengoperasian mengalami hambatan dalam sisi keamanan dan tidak adanya ketentuan dalam proses akses Dosen, Mahasiswa maupun Karyawan sehingga mengalami adanya gangguan seperti kebocoran data

SQL Injection adalah salah satu ancaman serangan terhadap situs web [10] serangan ini menargetkan situs yang memiliki celah sehingga dapat dilakukan eksploitasi oleh peretas yang menargetkan data pribadi dalam basis data [10][11]. Berdasarkan penelitian [10] menemukan serangan dari *SQL Injection* itu memiliki tingkat akurasi sebesar 90,07%, data pribadi berupa NIK(Nomor Induk Kependudukan) merupakan data yang menyangkut data geografis kependudukan di Indonesia sangat penting untuk dilindungi dari kebocoran data hal ini dikarenakan data NIK telah aset bagi setiap penduduk yang bersifat *immateril* [12] dalam sistem pendaftaran mahasiswa baru terdapat formulir yang diharuskan pengguna mengisikan data tersebut.

Analisis penilaian risiko digunakan untuk melakukan identifikasi berupa ancaman terhadap suatu organisasi atau ancaman yang menargetkan kepada organisasi tujuan baik dalam internal maupun eksternal [2] dalam penelitian ini institusi pendidikan dijadikan analisis penilaian risiko dalam sistem pendaftaran mahasiswa baru dan memberikan suatu penilaian seperti Rendah,Menengah dan Tinggi [13].

Pada penelitian ini bertujuan untuk menganalisis keamanan pada sistem informasi pendaftaran mahasiswa yang difokuskan dalam penetrasi serangan *SQL Injection* sebagai uji coba terhadap situs pendidikan mahasiswa baru yang rentan, untuk data pribadi calon mahasiswa baru berupa data *dummy* yang didapatkan dari database MySQL data tersebut bersifat sekunder yang bersifat tidak asli namun data tersebut mengacu dari aturan FERPA (*Family Educational Rights and Privacy Act*) [14], NIK (Nomor Induk Kependudukan) sebagai data sensitif untuk dijadikan penilaian analisis risiko, dengan menggunakan CVSS (*Common Vulnerability Scoring System*) dan metode *min-max* minimal dan maksimal untuk mengetahui nilai skor indeks terkait berapa banyak data yang mengalami kebocoran pada situs tersebut.

Manfaat yang dapat dihasilkan yaitu dapat dilakukan evaluasi sisi keamanan pada setiap situs pendaftaran mahasiswa baru dengan hasil dari penilaian risiko berdasarkan data-data sensitif yang terungkap atau seberapa jumlah data sensitif mengalami kebocoran data secara lengkap, sehingga pihak institusi pendidikan di bagian departemen IT akan sangat *aware* dengan kebocoran data.

2. Tinjauan Pustaka

Beberapa penelitian yang bersangkutan dengan analisa kebocoran data dan keamanan sistem informasi dengan analisis penilaian risiko ini yaitu menurut [6] terkait dengan kebocoran data di pendidikan setiap pendaftar memberikan file atau data-data persyaratan untuk pendaftaran dengan dokumen seperti foto, akta kelahiran dan masih banyak dokumen persyaratan yang terjadinya kebocoran data, maka peneliti ini menggunakan metode enkripsi data, lalu brdasarkan penelitian yang berjudul "*Security Risk Analysis of Information System in Academic Institution based on Business Perspective: A Case Study*" ditulis oleh Prajna Deshanta Ibnugraha, menggunakan metode CVSS(*Common Vulnerability Scoring System*) sebagai penilaian analisis risiko keamanan berupa *Low, Medium, High* [7]. Dari penelitian dengan judul "*The risk ranking of projects: a methodology*" mengemukakan *Risk Rank* atau penilaian risiko yaitu untuk menilai tingkat relatif dari risiko sehingga tingkat yang sesuai usaha dapat diterapkan untuk pengelolaan [15] Menurut [19] yang berjudul "Analisis Keamanan Website Menggunakan Metode Scanning Dan Perhitungan Security Metriks" *Acunetix* yang merupakan aplikasi open source pemindai keamanan *Web*, dengan tujuan untuk mengidentifikasi ancaman dan kelemahan dalam arsitektur aplikasi *Web.* Penulis menggunakan aplikasi *Acunetix* sebagai mengumpulkan informasi. Dari penelitian selanjutnya yang berjudul

"Manajemen Risiko Sistem Informasi Akademik pada Perguruan Tinggi Menggunakan Metoda *Octave Allegro*" ditulis oleh Deni Ahmad Jakaria terkait dengan manajemen risiko. Tujuan dari penilaian risiko ialah untuk melakukan identifikasi ancaman terhadap organisasi atau ancaman yang disematkan kepada organisasi lain [2].

Beberapa penelitian yang bersangkutan dengan ancaman terhadap sistem informasi dalam hal ini difokuskan pendaftaran mahasiswa baru dan serangan dari *SQL Injection*. Menurut [16] ancaman atau *threats* merupakan setiap peristiwa yang terjadi dapat mengalami kerusakan pada sistem informasi sehingga hilangnya kerahasiaan, ketersediaan dan integritas. Ancaman terhadap sistem informasi berbahaya seperti dilakukan modifikasi data-data sampai penghapusan data. Menurut [10], [17], dan [18] pengertian dari *SQL Injection* merupakan serangan yang menargetkan sistem yang mengalami *vulnerable* atau tingkat kerentanan yang rendah sehingga pihak ke 3 melakukan dengan mengirimkan *query illegal* atau perintah kueri ke server.

3. Metodologi

Penelitian ini menggunakan metode *risk rating*[7] sebagai analisis risiko keamanan dari sistem pendaftaran mahasiswa baru, penelitian ini dimulai dari melibatkan implementasi dari *SQL Injection* terhadap sistem pendaftaran mahasiswa baru yang bersifat *local* dan analisis penilaian risiko dari CVSS(*Common Vulnerability Scoring System*) sebagai acuan untuk digunakan menilai kerentanan keamanan dapat ditunjukkan sebagai berikut[7], [21](*equation 1*).

Risk Score = Probability x Impact ..... (1)

Probability diartikan sebagai kemungkinan dari serangan keamanan yang merugikan organisasi[22]. Dalam FERPA(*Family Educational Rights and Privacy Act*) kemungkinan kebocoran data dari serangan *SQL Injection* sistem informasi pendaftaran mahasiswa baru data yang bersifat sensitif dapat dilihat pada tabel 1.

Tabel 1. FERPA (*Family Educational Rights and Privacy Act*)

Parameter / *Group Metrics*	Indikator
Directory Information(DI)	Nama Institusi / Sekolah
	Alamat Institusi / Sekolah
	No Telp Institusi / Sekolah
	Akreditasi
Educational Information(EI)	No Identitas
	Riwayat Pendidikan
	Nilai
	Catatan Rekam Medis
Personally Identifiable(PI)	Nama
	No Hp
	Tempat Tanggal Lahir
	Agama
	Jenis Kelamin
	Nama Orang Tua
	Foto Diri
	Alamat IP
	Ras
	Email
	Password

Sumber: [14]

Informasi direktori dalam institusi pendidikan data tersebut memiliki dampak yang rendah[14], dalam hal ini sistem informasi pendaftaran mahasiswa baru tidak dipublikasikan karena data tersebut jika pengguna ingin melihat kembali sesudah di isi dalam form maka pengguna tersebut seharusnya melakukan *Log-in* dahulu. Adapun dampak jika *Educational Information(EI)* dan *Personally Identifiable(PI)* dipublikasikan tanpa seijin dari mahasiswa maka akan terjadinya tingkat kepercayaan yang berkurang dari mahasiswa hal ini dari penelitian[14],

[23] dan ditambahkan NIK(Nomor Induk Kependudukan) di Indonesia[12] kalkulasi data sensitif *Probability* sebagai berikut dengan menggunakan metode min-max minimal dan maksimal (*equation* 2).

$$normalisasi(probability) = \frac{dataleak - batasmin}{batasmax - batasmin} \quad \text{..... (2)}$$

Normalisasi (*Probability)*: adalah nilai keseluruhan,
dataleak: adalah data nilai valid / kebocoran data,
batasmin: adalah batas minimal skor nilai,
batasmax: adalah batas maximal skor nilai.

Berdasarkan CVSS (*Common Vulnerability Scoring System*) skor minimal 0 dan skor maksimal 10, skor tersebut dilihat pada tabel 2.

Tabel 2. CVSS(*Common Vulnerability Scoring System*)

Nilai Risiko	Deskripsi
0.0	Tidak Ada
0.1 – 3.9	Rendah
4.0 – 6.9	Medium
7.0 – 8.9	Tinggi
9.0 – 10.0	Sangat Tinggi

Pada nilai dari *Base Score* setelah di pindai oleh aplikasi *Acunetix* semakin besar skor kerentanan dari serangan maka perlu ditangani [24], oleh karena itu nilai dari *Impact* ditentukan berdasarkan CVSS V3 *base vector* dari *SQL Injection* berdasarkan dari *Acunetix* dilihat pada tabel 3.

Tabel 3. *SQL Injection CVSS Base Vector Acunetix*

	Parameter	Nilai
CVSS V3	*Attack Vector(AV)*	*Network*
	Attack Complexity(AC)	*Low*
	Privilege Required(PR)	*None*
	User Interaction(UI)	*None*
	Scope(S)	Changed
	Confidentiality Impact(CI)	*High*
	Integrity Impact(II)	*High*
	Availability Impact(AI)	*None*
Base Score		10

Sumber :[24]

4. Hasil dan Pembahasan

4.1. Proses Analisis Celah Keamanan

Proses dari serangan *SQL Injection* dapat diketahui jika perangkat lunak memindai dari keamanan sistem mendeteksi adanya celah dari *SQL Injection* maka proses untuk penilaian risiko kebocoran data dapat dilakukan. Berikut adalah proses untuk melakukan analisis risiko [13], untuk melakukan identifikasi serangan *SQL Injection* menggunakan dengan perangkat lunak memindai situs *Acunetix*[24], proses penyerangan atau (*Attack Vulnerability*), lalu data didapatkan sebagai sampel untuk melakukan penilaian risiko dan berdasarkan hasil dari uji reliabilitas terhadap data yang mengacu pada aturan FERPA(*Family Educational Rights and Privacy Act*)[14], dan melakukan analisis risiko CVSS(*Common Vulnerability Scoring System*) berdasarkan 3 kategori yaitu *LOW,MEDIUM & HIGH.* Berikut adalah gambar 1 sebagai proses untuk melakukan analisis keamanan

Gambar 1 Proses Analisis Kebocoran Data

4.2. Hasil Celah Terhadap 2 Sistem Pendaftaran Mahasiswa Baru PMB

Sebelum melakukan tahapan dari *SQL Injection* langkah pertama menggunakan *Acunetix* sebagai untuk mencari kelemahan terkait sistem keamanan pada situs sebagai berikut pada tabel 4.

Tabel 4 Hasil Pindai *Acunetix*

Nama Situs	*Severity (Threat Level)*	Bagian halaman situs yang terdampak	Bagian yang terkena SQL Injection
A	HIGH	Login.php	password
B	HIGH	Login.php	Email, password
		Register.php	Email, full_name, id_user, password

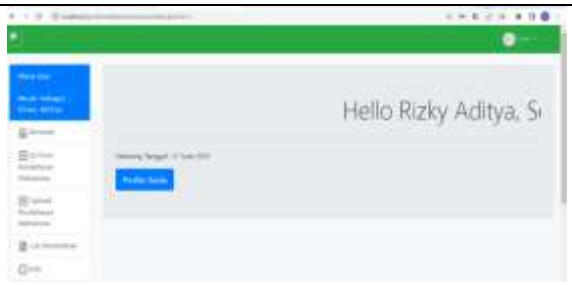
Dari 2 situs tersebut pada saat di pindai menggunakan *Acunetix* menunjukkan tingkatan risiko *HIGH* dari *SQL Injection*

4.3. Hasil Penetrasi Terhadap 2 Sistem Pendaftaran Mahasiswa Baru PMB

Setelah melakukan scan dengan tingkat kerentanan *SQL Injection* maka tahap selanjutnya menguji dengan cara manual seperti `-` `or`[25] pada *login form*. Hasil uji coba penetrasi dilihat pada tabel 5.

Tabel 5 Hasil Penetrasi Manual *SQL Injection*

Nama Web	Uji Coba Manual SQL Injection	Keterangan
A		Tidak masuk
B		Masuk

Nama Web	Uji Coba Manual SQL Injection	Keterangan
		

Pada tabel 5 menunjukkan perbedaan ketika *login* terhadap 2 sistem pendaftaran mahasiswa baru yaitu situs A tidak masuk sedangkan situs B dapat masuk dengan menggunakan perintah `-` `or`, menurut hasil penelitian[25] menggunakan query atau perintah *illegal* seperti `-` `or` atau ` OR 1= 1; /* maka username dan passwordnya akan bersifat *TRUE* sehingga dapat melakukan *bypass login*.

Adapun pengujian menggunakan aplikasi *Havij* dapat dilihat pada tabel 6.

Tabel 6 Pengujian *Havij*

Nama Web	Hasil Havij	Keterangan
A	-	Tidak dapat masuk
B		Dapat masuk

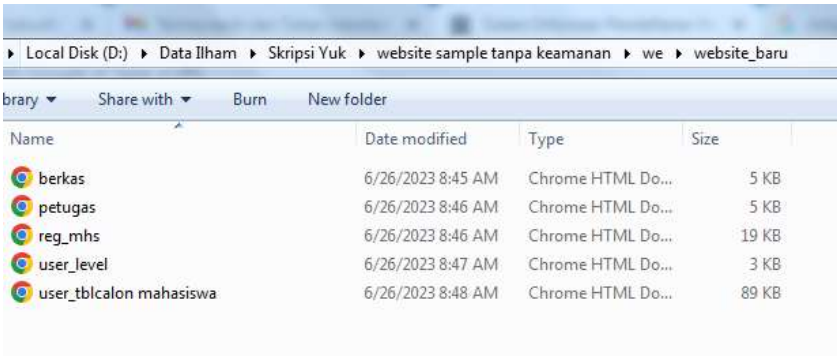
4.4. Analisis Hasil Perhitungan Kebocoran Data

Adapun hasil data-data yang sudah dilakukan serangan menggunakan *havij* pada tabel 7 sebagai berikut, untuk data tersebut berupa tabel sedangkan kolom tidak dapat ditampilkan karena alasan keamanan.

Tabel 7 Hasil Kebocoran Data

Nama Web	Tabel
A	-
B	Berkas petugas
	Reg_mhs
	Tbl_user_level
	User_tbl

Data tersebut berupa tabel pada basis data sistem pendaftaran mahasiswa baru, file tersebut berformat .html seperti pada gambar 2.



Gambar 2. Data

Setiap data pada masing-masing file memiliki jumlah kebocoran data, namun perlu diketahui setiap data akan melalui tahapan filtering berdasarkan aturan FERPA dapat dilihat pada tabel 1. Apabila terdapat data yang termasuk kedalam ferpa maka akan dihitung 1 calon mahasiswa dengan data masing-masing dengan nilai 1.

4.5. Analisis Hasil Penilaian Risiko Kebocoran Data (*Risk Rating*)

Penilaian risiko ini mengacu pada model risiko CVSS (*Common Vulnerability Scoring System*) dapat dilihat pada tabel 2 dan tabel 3 untuk *base vector*, untuk data sensitif dapat dilihat pada tabel 1. Dan hasil dari kebocoran data tersebut sebagai berikut.

Tabel 8 Skor Situs B

Nama Web	File	Nilai Skor	Keterangan
A	-	0.0	Rendah
B	Berkas.html	0.0	Rendah
	Petugas.html	0.08	Rendah
	Reg_mhs.html	5.76	Medium
	Tbl_user_level.html	0.0	Rendah
	User_tbl.html	1.92	Rendah

Adapun pembanding analisis yang dimana CVSS *Acunetix* dijadikan sumber acuan, hasil dari penilaian analisis dari kebocoran data sebagai berikut pada tabel 9

Tabel 9 Hasil Pengujian Perbandingan

Nama Web	*Risk* CVSS + *Severity*(*Risk Rating*) *minimal maximal method* + ferpa	*Risk* CVSS *Severity Acunetix*	Keterangan
A	*LOW*	*HIGH*	Penyerang dapat melakukan serangan SQL Injection namun data nya tidak muncul pada saat pengujian
B	*MEDIUM*	*HIGH*	Penyerang dapat melakukan serangan SQL Injection namun datanya terdiri atas data pribadi calon mahasiswa baru, data petugas/administrator.

Berdasarkan tabel 9 menunjukkan perbedaan tingkatan risiko ketika diuji diantaranya situs pendaftaran mahasiswa baru A menunjukkan indikasi rendah atau *LOW* karena pada saat pengujian menggunakan aplikasi *Havij* tidak menunjukkan respon ketika dilakukan penetrasi *SQL Injection*, hasilnya situs 1 dikatakan *LOW* karena tidak ada data yang diungkap atau kebocoran data sensitif, dan situs pendaftaran mahasiswa baru B dikatakan medium karena terdapat data yang diungkap atau kebocoran ketika melakukan penetrasi *SQL Injection*, sebagai pembanding menggunakan CVSS dari *Acunetix*.

Berdasarkan hasil pengujian terhadap sampel 2 situs pendaftaran mahasiswa baru didapatkan situs yang A tidak mengalami kebocoran data namun perlu adanya evaluasi dari sisi keamanan dengan cara melakukan update dari situs website dan melakukan backup database rutin dan Situs B sangat perlu adanya keamanan dengan cara dari halaman login digunakan *escape string* di bagian *username* atau *email* dan *password* agar ketika melakukan penetrasi *SQL Injection* melalui halaman log-in tidak dapat menggunakan seperti `-` `or`.

5. Simpulan

Pada penelitian ini bertujuan untuk implementasi dari *SQL Injection* terhadap sistem pendaftaran mahasiswa baru dengan metode *risk rating* berdasarkan dari kebocoran data dalam hal ini penulis meneliti terkait kebocoran data dalam insiden serangan *SQL Injection* yang dimana menargetkan sistem pendaftara mahasiswa baru dengan keamanan yang rendah. Dan penilaian risiko dapat memberikan gambaran dari data mana saja yang termasuk data sensitif dengan acuan dari FERPA (*Family Educational Rights and Privacy Act*) ditambah dengan NIK (Nomor Induk Kependudukan). Dan hasil dari pengujian terhadap 2 sistem informasi pendaftaran mahasiswa baru menghasilkan detail bagaimana data sensitif diungkap atau kebocoran data sensitif oleh peretas. Dengan demikian dalam penelitian selanjutnya atau saran untuk penelitian selanjutnya ada kebutuhan yang lebih untuk tidak hanya mendemontrasikan *SQL Injection* namun untuk sisi keamanan data-data pribadi. Untuk penelitian selanjutnya dan rekomendasi untuk pihak manajemen kampus apabila sistem

Daftar Referensi

[1] B. Arismanto and S. Rahmadhani, “Pengembangan Sistem Penerimaan Mahasiswa Baru pada STIES Imam Asy Syafii Pekanbaru,” *J. Intra-Tech*, vol. 3, no. 1, pp. 57–72, 2019.

[2] D. A. Jakaria, R. T. Dirgahayu, and Hendrik, “Manajemen Risiko Sistem Informasi Akademik pada Perguruan Tinggi Menggunakan Metoda Octave Allegro,” *In Seminar Nasional Aplikasi Teknologi Informasi (SNATI), pp. E37-E42, 2013.*

[3] S. Priyanto and H. K. Siradjuddin, “Sistem Informasi Pendaftaran Mahasiswa Baru Berbasis Web Pada Politeknik Sains & Teknologi Wiratama Maluku Utara,” *IJIS - Indones. J. Inf. Syst.*, vol. 3, no. 1, p. 20, 2018, doi: 10.36549/ijis.v3i1.38.

[4] R. Pramana, A.,Watrianthos, “Sistem Informasi Pendaftaran Mahasiswa Baru Berbasis Android,” *J. Inform. Upgris*, vol. 5, no. 2, pp. 121–125, 2019.

[5] D. Wijonarko and F. W. S. Budi, “Implementasi Framework Laravel Dalam Sistem Pendaftaran Mahasiswa Baru Politeknik Kota Malang,” *J. Inform. dan Rekayasa Elektron.*, vol. 2, no. 2, p. 35, 2019, doi: 10.36595/jire.v2i2.116.

[6] M. A. Sutejo and M. Hardjianto, “Pengamanan File Pendaftaran Siswa Baru Menggunakan Metode Algoritme Rc4 Di Tk Nurul Irfan Security of New Student Registration Files Using the Rc4 Algorithm Method in Tk Nurul Irfan,” *Semin. Nas. Mhs. Fak. Teknol. Inf. Jakarta-Indonesia*, vol. 4, no. September, pp. 394–401, 2022.

[7] P. D. Ibnugraha, L. E. Nugroho, and P. I. Santosa, “Security Risk Analysis of Information System in Academic Institution based on Business Perspective : A Case Study,” vol. 8, pp. 87–91, 2019.

[8] J. Fonseca, N. Seixas, M. Vieira, and H. Madeira, “Analysis of field data on web security vulnerabilities,” *IEEE Trans. Dependable Secur. Comput.*, vol. 11, no. 2, pp. 89–100, 2014, doi: 10.1109/TDSC.2013.37.

[9] H. Wahyudi, A. Zulianto, A. Maulana, S. Mardira Indonesia, and U. Langlangbuana, “Audit Keamanan Sistem Informasi Manajemen Akademik Dan Kemahasiswaan Menggunakan SNI ISO/IEC 27001:2013 Studi Kasus STMIK Mardira Indonesia,” *J. Comput. Bisnis*, vol. Vol. 14 No. 1, no. 1, pp. 40–46, 2020.

[10] C. Pinzón, J. F. De Paz, J. Bajo, Á. Herrero, and E. Corchado, “AIIDA-SQL: An Adaptive Intelligent Intrusion Detector Agent for detecting SQL injection attacks,” *2010 10th Int. Conf. Hybrid Intell. Syst. HIS 2010*, pp. 73–78, 2010, doi: 10.1109/HIS.2010.5600026.

[11] K. Ahmad and M. Karim, “A Method to Prevent SQL Injection Attack using an Improved Parameterized Stored Procedure,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 324–332, 2021, doi: 10.14569/IJACSA.2021.0120636.

[12] Z. Fadhli, S. W. Rahayu, and I. A. Gani, “Perlindungan Data Pribadi Konsumen Pada Transaksi Paylater,” *J. Huk. Magnum Opus*, vol. 5, no. 1, pp. 119–132, 2022.

[13] P. D. Ibnugraha, L. E. Nugroho, Widyawan, and P. I. Santosa, “Risk analysis of database privelege implementation in SQL injection case,” *J. Teknol.*, vol. 78, no. 5–7, pp. 113–116, 2016, doi: 10.11113/jt.v78.8724.

[14] P. Deshanta, A. Satria, F. Sekar, and M. Fahru, “The Reliability Analysis for Information Security Metrics in Academic Environment,” vol. 7, no. March, pp. 92–97, 2023.

[15] D. Baccarini and R. Archer, “The risk ranking of projects: A methodology,” *Int. J. Proj. Manag.*, vol. 19, no. 3, pp. 139–145, 2001, doi: 10.1016/S0263-7863(99)00074-5.

[16] S. K. PANDEY, “A Comparative Study of Risk Assessment Methodologies for Information Systems,” *Bull. Electr. Eng. Informatics*, vol. 1, no. 2, pp. 111–122, 2012, doi: 10.12928/eei.v1i2.231.

[17] A. Bastian, H. Sujadi, and L. Abror, “Analisis Keamanan Aplikasi Data Pokok Pendidikan (DAPODIK) Menggunakan Penetration Testing Dan SQL Injection,” *INFOTECH J.*, vol. 6, no. 2, pp. 65–70, 2020.

[18] A. S. Irawan, E. S. Pramukantoro, and A. Kusyanti, “Pengembangan Intrusion Detection System Terhadap SQL Injection Menggunakan Metode Learning Vector Quantization,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 6, pp. 2295–2301, 2018.

[19] M. Z. Maharani, H. R. Andrian, and S. J. I. Ismail, “Analisis Keamanan Website Menggunakan Metode Scanning Dan Perhitungan Security Metriks,” *e-Proceeding Appl. Sci.*, vol. 3, no. 3, pp. 1775–1782, 2017.

[20] A. Bin Ibrahim and S. Kant, “Penetration Testing Using SQL Injection to Recognize the Vulnerable Point on Web Pages,” *Int. J. Appl. Eng. Res.*, vol. 13, no. 8, pp. 5935–5942, 2018, [Online]. Available: http://www.ripublication.com

[21] L. Arafat, “Sistem Informasi Manajemen Risiko Proyek Di Cv. Artha Jaya,” 2019, [Online]. Available: https://elibrary.unikom.ac.id/id/eprint/864/

[22] Å. Nyre and M. Jaatun, “Seeking Risks : Towards a Quantitative Risk Perception Measure To cite this version,” 2017.

[23] P. D. Ibnugraha, L. E. Nugroho, and P. I. Santosa, “Risk model development for information security in organization environment based on business perspectives,” *Int. J. Inf. Secur.*, vol. 20, no. 1, pp. 113-126, 2020, doi: 10.1007/s10207-020-00495-7.

[24] F. Al Fajar, “Analisis Keamanan Aplikasi Web Prodi Teknik Informatika Uika Menggunakan Acunetix Web Vulnerability,” *Inova-Tif*, vol. 3, no. 2, pp. 110-121, 2020, doi: 10.32832/inova-tif.v3i2.4127.

[25] A. K. Dalai and S. K. Jena, “Neutralizing SQL injection attack using server side code modification in web applications,” *Secur. Commun. Networks*, vol. 2017, no. 3, pp. 158–173, 2017, doi: 10.1155/2017/3825373.