JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika

[E-ISSN 2656-0321]
[Vol 19 No 2]
[Agustus 2025]

MINIMALISASI *ATTACK SURFACE* PADA SERVER WEB MELALUI PENDEKATAN KEAMANAN BERLAPIS

Musa Amin
IAIN Pontianak
musaiainptk@gmail.com

ARTICLE INFO

Keywords

server security, web security, proxy, virtual private network

ABSTRACT

Servers used to run WordPress-based web applications face serious challenges due to a wide attack surface, especially when backend and SSH access are exposed to the public. This study designs and implements an access protection architecture for servers and websites by combining Cloudflare proxy, Virtual Private Network (VPN), and Web Application Firewall (WAF) to mitigate these risks. An experimental method is employed by building a simulated LAMP and WordPress-based infrastructure on two Virtual Private Servers (VPS), where all public traffic is routed through Cloudflare and administrative access is strictly limited to VPN tunnels. The test results show that the proposed architecture effectively eliminates unauthorized access to backend pages and SSH services without disrupting public access to the website. This approach demonstrates that a simple layered defense strategy can be practically applied to enhance server security while providing a protection model that can be replicated for similar infrastructures.

1. Pendahuluan

Perkembangan teknologi informasi dan internet yang pesat telah mendorong transformasi digital di berbagai sektor, menempatkan situs web sebagai aset digital krusial bagi institusi, perusahaan, maupun perorangan. Situs web tidak lagi hanya berfungsi sebagai medium penyampaian informasi, tetapi juga sebagai gerbang utama interaksi, transaksi, dan representasi identitas di dunia maya. Seiring dengan meningkatnya peran strategis tersebut, ancaman keamanan siber terhadap infrastruktur web juga mengalami eskalasi yang signifikan. Keamanan *web server* menjadi prioritas utama untuk menjamin ketersediaan layanan (*availability*), integritas data (*integrity*), dan kerahasiaan informasi (*confidentiality*) [1].

Salah satu *platform Content Management System* (CMS) yang paling populer di dunia adalah WordPress, yang berjalan di atas LAMP (Linux, Apache, MySQL, PHP). Popularitas ini, sayangnya, menjadikannya target utama bagi para penyerang siber [2]. Arsitektur *server* konvensional, di mana *Virtual Private Server* (VPS) diekspos secara langsung ke internet publik, menghadirkan berbagai titik serangan. Antarmuka administratif, seperti halaman *login backend*, dan protokol manajemen *server* jarak jauh, seperti *Secure Shell* (SSH), menjadi titik masuk yang rentan [3], [4]. Penyerang dapat dengan mudah melakukan pemindaian *port*, melancarkan serangan *brute-force* untuk menebak kredensial, atau mengeksploitasi kerentanan yang belum ditambal langsung pada alamat IP *server*. Konfigurasi awal seperti ini menciptakan *attack surface* yang luas dan sangat berisiko [5].

Berbagai pendekatan telah dilakukan dalam penelitian sebelumnya untuk meningkatkan keamanan *server* dan aplikasi web, khususnya WordPress. Salah satu metode yang banyak diterapkan adalah penggunaan *firewall* berbasis host seperti iptables untuk memfilter akses jaringan secara langsung di tingkat *server* [6]. Selain itu, teknik penyembunyian alamat IP *server* menggunakan *proxy* menjadi strategi populer untuk melindungi *origin server* dari pemindaian dan serangan langsung [7]. Di sisi aplikasi, implementasi *Web Application Firewall* (WAF) digunakan untuk mendeteksi dan memblokir lalu lintas berbahaya yang menargetkan kerentanan di layer aplikasi web [8], [9]. Namun, pendekatan-

pendekatan tersebut seringkali diterapkan secara terpisah, sehingga efektivitasnya menjadi terbatas dalam menghadapi serangan yang kompleks dan terkoordinasi.

Untuk mengatasi tantangan keamanan tersebut, diperlukan sebuah pendekatan arsitektur keamanan berlapis (*layered security*) atau pertahanan mendalam (*defense-in-depth*) [10]. Model keamanan ini bertujuan untuk melindungi aset informasi dengan menempatkan beberapa lapis kontrol keamanan di seluruh sistem. Jika satu lapisan gagal, lapisan berikutnya siap untuk mencegah serangan lebih lanjut. Penelitian ini mengusulkan sebuah rancangan arsitektur keamanan terintegrasi untuk *web server* berbasis LAMP dan WordPress dengan memanfaatkan kombinasi teknologi iptables, *Virtual Private Network* (VPN), *proxy* berbasis *cloud*, dan *Web Application Firewall* (WAF).

Dalam arsitektur yang diusulkan, semua lalu lintas publik dialihkan melalui *proxy* untuk menyembunyikan alamat IP *server* yang sebenarnya (*origin server*) serta memfilter lalu lintas berbahaya. Lebih lanjut, akses terhadap antarmuka administratif dan akses SSH dibatasi secara ketat, di mana keduanya hanya dapat dijangkau melalui koneksi VPN yang aman. Kebijakan ini diperkuat dengan aturan pada WAF yang hanya mengizinkan alamat IP dari *server* VPN untuk mengakses halaman administratif. Dengan demikian, penelitian ini bertujuan untuk merancang, mengimplementasikan, dan menganalisis efektivitas sebuah model keamanan yang secara drastis mengurangi *attack surface*, meningkatkan keamanan *server*, dan melindungi aset digital dari ancaman siber yang terus berkembang. Kontribusi utama dari penelitian ini adalah menyajikan sebuah kerangka kerja keamanan yang praktis, efektif, dan dapat direplikasi untuk pengelola sistem informasi.

2. Metodologi Penelitian

Metodologi penelitian ini menggunakan pendekatan eksperimental dengan merancang dan membangun *prototype* arsitektur keamanan. Penelitian dilakukan secara sistematis melalui beberapa tahapan yang terstruktur, mulai dari persiapan lingkungan, implementasi konfigurasi, hingga pengujian fungsional untuk memvalidasi efektivitas arsitektur yang diusulkan.

2.1. Perangkat dan Kebutuhan Sistem

Simulasi dan implementasi arsitektur ini memanfaatkan sumber daya *cloud computing* dan perangkat lunak sebagai berikut:

1) *Web Server*: 1 vCPU, 1 GB RAM, 25 GB *storage*, sistem operasi Ubuntu 24.04, Apache *web server*, MariaDB *database*, WordPress, dan satu alamat IP publik statis.
2) *VPN Server*: 1 vCPU, 1 GB RAM, 25 GB *storage*, sistem operasi Ubuntu 24.04, Outline VPN *server*, dan satu alamat IP publik statis.
3) Cloudflare: digunakan sebagai *DNS Manager*, *proxy*, dan *Web Application Firewall* (WAF).

2.2. Tahapan Penelitian

Proses implementasi arsitektur dibagi menjadi empat tahapan:

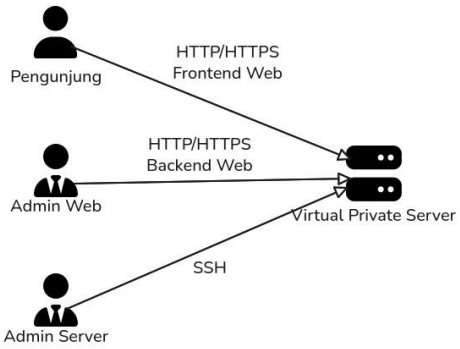
1) Persiapan Arsitektur Awal
   Tahap ini dimulai dengan mempersiapkan kondisi awal yang merepresentasikan arsitektur yang rentan.
   a) Melakukan *provisioning* pada dua unit VPS sesuai spesifikasi yang telah ditentukan.
   b) Pada VPS untuk *web server*, dilakukan instalasi dan konfigurasi sistem operasi Ubuntu 24.04, diikuti dengan instalasi Apache, MariaDB, PHP dan WordPress.
   c) Pada titik ini, sub domain (simulasi memakai test.aminlabs.my.id) diarahkan langsung ke IP publik VPS. Halaman depan, halaman administrator web (`/wp-admin`), dan port SSH (port 22) dapat diakses dari alamat IP mana pun di internet.
2) Instalasi VPN *Server*
   Tahap kedua adalah membangun jalur akses yang aman untuk administrator.

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 19 No 2]
[Agustus 2025]

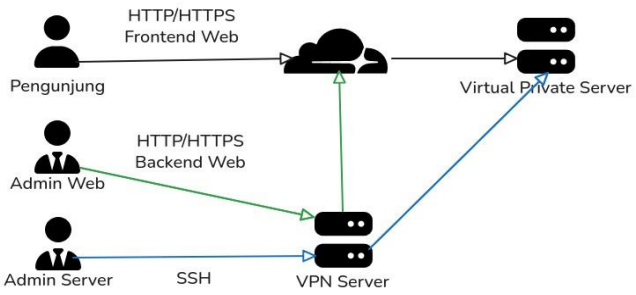
a) Instalasi dan konfigurasi Outline VPN Server.
b) *Access key* unik dibuat untuk digunakan oleh peran Admin Web dan Admin Server.

3) Implementasi *Proxy* dan *Firewall* (Arsitektur Usulan)
Tahap ini adalah inti dari implementasi arsitektur yang diusulkan, yaitu menyembunyikan dan melindungi server utama.
a) Sub domain test.aminlabs.my.id diintegrasikan dengan layanan Cloudflare.
b) Pada pengaturan DNS di Cloudflare, *A record* untuk sub domain diarahkan ke IP publik VPS *web server* dengan status proxy diaktifkan. Tindakan ini menyembunyikan IP asli VPS *web server* dari publik dan mengalihkan semua lalu lintas HTTP/HTTPS melalui jaringan Cloudflare.
c) Mengkonfigurasi aturan pada WAF di Cloudflare. Sebuah aturan dibuat untuk memblokir semua akses ke *path* URL `/wp-admin/*` dan `/wp-login.php`, kecuali jika permintaan berasal dari alamat IP publik milik VPN Server.

4) *Server Hardening*
Tahap terakhir adalah mengamankan jalur manajemen *server* yang paling kritis.
a) Pada VPS *web server*, konfigurasi layanan SSH dimodifikasi.
b) Mengkonfigurasi iptables yang merupakan *firewall* di Linux. Konfigurasinya hanya mengizinkan akses dari IP milik Cloudflare untuk proxy HTTP/HTTPS dan dari IP milik VPN *server* untuk koneksi ke protokol SSH, sementara akses dari IP lainnya akan ditolak.



Ga
mbar 1: Arsitektur awal



Gambar 2: Arsitektur usulan

2.3. Skenario Pengujian dan Analisis

Untuk mengukur keberhasilan implementasi, dilakukan pengujian berbasis skenario untuk membandingkan kondisi sebelum dan sesudah penerapan arsitektur usulan.

1. Pengujian Arsitektur Awal

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 19 No 2]
[Agustus 2025]

a) Mencoba mengakses halaman `https://test.aminlabs.my.id/wp-admin` dari IP publik acak. Hasil yang diharapkan, berhasil menampilkan halaman login.
b) Mencoba melakukan koneksi SSH ke user@IP-Web-Server dari IP publik acak. Hasil yang diharapkan, berhasil mendapatkan prompt login SSH.

2. Pengujian Arsitektur Usulan
   a) Tanpa VPN: mencoba mengakses https://test.aminlabs.my.id/wp-admin dari IP publik acak. Hasil yang diharapkan, akses diblokir oleh Cloudflare WAF (menampilkan galat HTTP 403).
   b) Tanpa VPN: mencoba melakukan koneksi SSH ke user@IP-Web-Server dari IP publik acak. Hasil yang diharapkan, koneksi ditolak atau *timeout*.
   c) Menggunakan VPN: mencoba mengakses https://test.aminlabs.my.id/wp-admin. Hasil yang diharapkan, berhasil menampilkan halaman *login*.
   d) Menggunakan VPN: mencoba melakukan koneksi SSH ke user@IP-Web-Server. Hasil yang diharapkan, berhasil mendapatkan *prompt login* SSH.

Analisis dilakukan dengan membandingkan hasil pengujian. Keberhasilan arsitektur usulan dinilai dari kemampuannya untuk menolak semua akses administratif yang tidak sah (tidak melalui VPN) dan hanya memperbolehkan akses yang sah (melalui VPN), sehingga membuktikan bahwa *attack surface* telah berhasil diminimalisir secara efektif.

3. Hasil dan Pembahasan

3.1. Hasil Pengujian

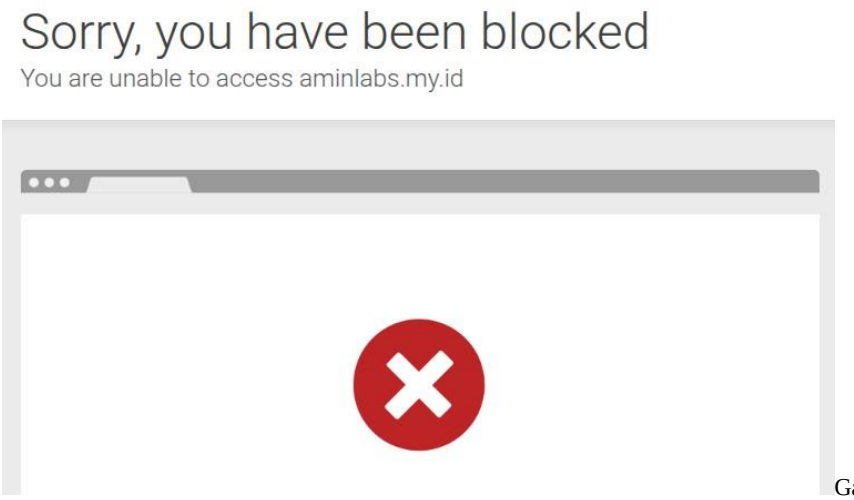
Pengujian dilakukan pada kedua arsitektur untuk memvalidasi efektivitas kontrol keamanan yang diimplementasikan. Hasil dari setiap skenario pengujian dirangkum secara komparatif pada Tabel 1.

Tabel 1: Hasil Perbandingan Pengujian Akses

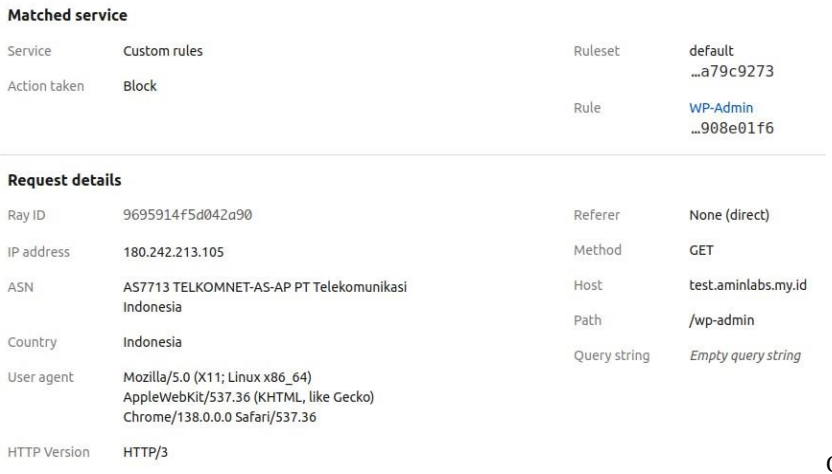
Skenario Uji	Kondisi Akses	Hasil pada Arsitektur Awal	Hasil pada Arsitektur Usulan	Keterangan
Akses Admin Web	Tanpa koneksi VPN	Berhasil	Gagal	Akses diblokir oleh WAF Cloudflare
Akses Admin Web	Pakai koneksi VPN	Berhasil	Berhasil	Akses diizinkan karena IP sumber sesuai dengan aturan Cloudflare WAF
Akses Admin Server	Tanpa koneksi VPN	Berhasil	Gagal	Koneksi ditolak oleh *firewall* di *server*
Akses Admin Server	Pakai koneksi VPN	Berhasil	Berhasil	Koneksi diizinkan karena IP sumber sesuai aturan *firewall* di *server*
Akses Pengunjung	Akses Publik	Berhasil	Berhasil	Akses publik tetap berjalan normal melalui Cloudflare *proxy*

Hasil pengujian secara tegas menunjukkan bahwa arsitektur usulan berhasil mengimplementasikan seluruh kontrol keamanan yang dirancang. Akses administratif, baik ke *backend* web maupun SSH, berhasil dibatasi hanya untuk pengguna yang terhubung melalui VPN, sementara akses publik ke *frontend* situs web tetap tidak terganggu.

JURNAL TEKNOLOGI INFORMASI
Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika
[E-ISSN 2656-0321]
[Vol 19 No 2]
[Agustus 2025]



Gambar 3: Akses ke /wp-admin diblokir



Gambar 4: Log akses yang diblokir

3.2. Pembahasan

Hasil yang diperoleh membuktikan bahwa arsitektur yang diusulkan secara signifikan meningkatkan keamanan *server*. Pembahasan berikut akan menganalisis implikasi dari hasil tersebut dari beberapa perspektif keamanan.

1. Reduksi *Attack Surface*
   Pada arsitektur awal, *attack surface* sangat luas. Siapa pun di internet dapat menemukan dan mencoba berinteraksi langsung dengan tiga titik masuk utama: *frontend web*, *backend web* (wp-admin), dan port SSH. Hal ini membuat *server* rentan terhadap serangan otomatis seperti:
   a) Pemindaian Massal: Bot secara konstan memindai internet untuk mencari halaman /wp-admin dan *port* SSH yang terbuka.
   b) Serangan *Brute-force*: Upaya menebak kombinasi *username* dan *password* secara terus-menerus pada halaman *login* dan SSH.
   c) Eksploitasi *Zero-Day*: Jika ada kerentanan baru pada WordPress atau layanan SSH, *server* dapat dieksploitasi sebelum *patch* sempat diterapkan.

   Arsitektur usulan secara efektif menghilangkan antarmuka administratif dari internet publik. Dengan mewajibkan koneksi VPN, titik masuk untuk /wp-admin dan SSH secara virtual

disembunyikan. Penyerang tidak dapat lagi menargetkan halaman *login* atau *port* SSH secara langsung karena lalu lintas mereka akan ditolak di tingkat jaringan (oleh WAF Cloudflare dan *firewall server*), jauh sebelum mencapai aplikasi WordPress atau layanan SSH itu sendiri.

2. Implementasi Pertahanan Berlapis
   Keberhasilan arsitektur ini terletak pada penerapan prinsip pertahanan berlapis, setiap komponen memberikan lapisan perlindungan yang saling melengkapi.
   a) Lapisan Tepi (*Edge Layer*) - Cloudflare: Lapisan ini bertindak sebagai garda terdepan. Dengan mengaktifkan *proxy*, alamat IP asli *server* tersembunyi, yang merupakan kemenangan keamanan fundamental. Penyerang tidak dapat melancarkan serangan DDoS atau serangan jaringan lainnya secara langsung ke *server*. Selain itu, WAF pada lapisan ini secara efisien menyaring dan memblokir upaya akses ilegal ke *backend* berdasarkan alamat IP, mengurangi beban kerja pada *server* utama.
   b) Lapisan Akses (*Access Layer*) - VPN Server: VPN menciptakan sebuah jalur privat dan terenkripsi untuk para administrator. Berfungsi sebagai titik kontrol akses tunggal. Hanya individu yang memiliki kunci akses VPN yang sah yang dianggap sebagai pengguna tepercaya. Lapisan ini memastikan bahwa identitas pengguna telah diautentikasi bahkan sebelum mereka diizinkan untuk mengetuk pintu *server* utama.
   c) Lapisan Server (*Server Layer*) – Host Firewall: Aturan *firewall* pada *web server*, yang membatasi akses SSH ke IP VPN berfungsi sebagai lapisan pertahanan terakhir.

   Secara keseluruhan, arsitektur ini mengubah model keamanan dari yang tadinya reaktif menjadi proaktif. Sistem tidak lagi hanya menunggu dan berharap dapat menahan serangan pada aplikasi, tetapi secara proaktif memblokir hampir semua lalu lintas yang tidak diinginkan sebelum mencapai aset-aset krusial. Model ini memberikan perlindungan yang kuat terhadap ancaman umum dan otomatis dengan biaya implementasi yang relatif rendah, menjadikannya solusi yang sangat praktis dan efektif untuk mengamankan infrastruktur *server* dan *web*.

4. Kesimpulan

Hasil penelitian menunjukkan bahwa penerapan arsitektur keamanan yang diusulkan berhasil meningkatkan keamanan server secara signifikan. Dengan mengimplementasikan pertahanan berlapis, arsitektur baru ini secara efektif mengurangi *attack surface* dengan membatasi akses administratif ke *backend web* dan SSH hanya melalui koneksi VPN. Akses publik ke situs web tetap berjalan normal, memastikan fungsionalitas bagi pengunjung. Arsitektur ini membuktikan bahwa strategi proaktif dalam memblokir lalu lintas berbahaya di tingkat jaringan, sebelum mencapai *server*, adalah metode yang efektif dan efisien. Keberhasilan ini mengimplikasikan bahwa model keamanan serupa dapat diterapkan secara luas untuk melindungi infrastruktur *server* dan *web* lainnya dari ancaman siber yang umum dan otomatis, menjadikannya solusi praktis bagi banyak organisasi.

Daftar Pustaka

[1] H. Z. Artie, M. Hilman, and S. Yazid, "Penilaian Risiko Keamanan Informasi Pusat Data pada Instansi XYZ," *J. Inform. Ekon. Bisnis*, pp. 270–276, Jun. 2025, doi: 10.37034/infeb.v7i2.1160.

[2] I. Setiawan, A. Widjajarto, and A. Budiyono, "Desain Kontrol Keamanan Pada Content Management System Wordpress Berdasar Aspek Aplikasi Dengan Panduan OWASP," *TEKNIKA*, vol. 19, no. 1, Art. no. 1, 2025, doi: 10.5281/zenodo.13756114.

[3] M. R. A. Fitra, A. A. S. Effendi, S. A. Priscilia, and D. Kiswanto, "Uji Penetrasi Menggunakan Hydra dan Metasploit pada Protokol Secure Shell," *JATI J. Mhs. Tek. Inform.*, vol. 9, no. 1, Art. no. 1, 2025, doi: 10.36040/jati.v9i1.12583.

[4] P. B. Baskara, I. M. Widiartha, and I. G. S. Astawa, "Analisis Resiko Celah Keamanan Website E-Commerce Berbasis Content Management System (CMS) Wordpress Menggunakan Vulnerability Scanning (Studi Kasus: beekella.com)," *Pros. Semin. Nas. Univ. Ma Chung Inform. Sist. Inf. Bhs. Dan Seni Farm.*, vol. 2, pp. 40–49, Sep. 2022.

[5] A. Nasir, "Analisa Celah Kelah Keamanan Terhadap Web Server Menggunakan Metode Attack Surface Dan Kepadatan Kerentanan," *Temat. J. Penelit. Tek. Inform. Dan Sist. Inf.*, pp. 67–72, Sep. 2020, doi: 10.56963/tematika.vi.256.

[6] T. Ariyadi, M. R. Pohan, M. K. Hadi, and A. A. Widodo, "Implementasi Firewall Pada Protokol SSH Linux Ubuntu Menggunakan Iptables," *Pros. Semin. Ris. Mhs.*, vol. 1, no. 1, Art. no. 1, Jan. 2024.
[7] D. Aryachandra, I. F. Yanto, M. M. Khair, and M. R. S. Pahlevi, "Menyembunyikan Alamat IP Webserver dengan Proxy Dns Records Cloudflare," *J. Sos. Teknol.*, vol. 4, no. 4, Art. no. 4, Apr. 2024, doi: 10.59188/jurnalsostech.v4i4.1221.
[8] D. A. Sandi and A. Tedyyana, "Implementasi dan Analisa Sistem Pencegahan Intrusi pada Aplikasi Web Menggunakan Web Application Firewall," *Repeater Publ. Tek. Inform. Dan Jar.*, vol. 2, no. 4, pp. 16–26, Aug. 2024, doi: 10.62951/repeater.v2i4.196.
[9] G. H. A. Kusuma, "Perancangan Skema Sistem Keamanan Jaringan Web Server menggunakan Web Application Firewall dan Fortigate untuk Mencegah Kebocoran Data di Masa Pandemi Covid-19," *J. Inform. Adv. Comput. JIAC*, vol. 2, no. 2, Art. no. 2, Nov. 2021, doi: 10.35814/jiac.v2i2.3259.
[10] S. Slamet, "Taksonomi Pertahanan Cyber Security Menggunakan Model Cyber Kill Chain," *SPIRIT*, vol. 16, no. 1, Art. no. 1, May 2024, doi: 10.53567/spirit.v16i1.332