ISSN: 2477-5126
e-ISSN:2548-9356

Embedded Wids Kismet Sebagai Perangkat Deteksi Serangan Data Link Layer Wi-Fi Access Point

Rizky Fachrurozy [1], Muhammad Yusuf Bambang Setiadji [2], Dimas Febriyan Priambodo [3*)]
[1,2,3] Jurusan Keamanan Siber, Politenik Siber dan Sandi Negara, Bogor
[1]Jln. H.Usa, Putat Nutug, Ciseeng, Bogor, Jawa Barat, 16120, Indonesia
email: [1] rizky.fachrurozy@bssn.go.id, [2] yusuf.setiadji@poltekssn.ac.id, [3] dimas.febriyan@poltekssn.ac.id

Abstract – **The rapid development of wireless network technology has an impact with a significant increase of users. Wi-Fi is one of the wireless technologies that is currently used widely by humans as a medium of transmission to exchange information. The increasing in the number of users is directly proportional to the increase the risk of crime occurring. In this case an attack on a Wi-Fi network aimed at taking certain advantage of the user or it's victim was done by threatening aspects of information security. Wireless Intrusion Detection System (WIDS) is a tool used to detect intrusions that occur against wireless network technology, one of it is Wi-Fi into the form of logs. Kismet is one of the wids that is free and opensource with complete WiFi attack detection capability. The implementation of WIDS Kismet applied to Raspberry Pi 4 was shown as an alternative wids device that has a low cost for use in small Wi-Fi network environments such as home networks or small enterprise. The analysis is done by calculating the performance of accuracy, precision, recall, and f-measure values from the implementation results using confusion matrix method. The result of the calculation obtained by the highest value for each calculated performance with 99.83% accuracy, 97.96% precision, 100% recall, and 98.9% fmeasure. The calculation aims to provide information that can be utilized for users who want to do a similar implementation and use it to detect Wi-Fi networks.**

Abstrak – **Pesatnya perkembangan teknologi jaringan nirkabel berdampak pada peningkatan pengguna yang cukup signifikan. Wi-Fi merupakan salah satu teknologi nirkabel yang saat ini banyak digunakan oleh manusia sebagai media transmisi untuk bertukar informasi. Peningkatan jumlah pengguna berbanding lurus dengan peningkatan risiko terjadinya kejahatan. Dalam hal ini serangan terhadap jaringan Wi-Fi yang ditujukan untuk mengambil keuntungan tertentu dari pengguna atau korbannya dilakukan dengan mengancam aspek keamanan informasi. *Wireless Intrusion Detection System* (WIDS) adalah alat yang digunakan untuk mendeteksi intrusi yang terjadi terhadap teknologi jaringan nirkabel, salah satunya adalah Wi-Fi ke dalam bentuk log. Kismet adalah salah satu wid yang gratis dan *open source* dengan kemampuan deteksi serangan WiFi yang lengkap. Implementasi WIDS Kismet yang diterapkan pada Raspberry Pi 4 ditunjukkan sebagai perangkat wids alternatif yang memiliki biaya rendah untuk digunakan di lingkungan jaringan Wi-Fi kecil seperti jaringan rumahan atau perusahaan kecil. Analisis dilakukan dengan menghitung kinerja nilai akurasi, presisi, *recall*, dan *f-measure* dari hasil implementasi menggunakan metode *confusion matrix*. Hasil perhitungan diperoleh nilai tertinggi untuk masing-masing performance yang dihitung dengan akurasi 99,83%, presisi 97,96%, recall 100%, dan fmeasure 98,9%. Perhitungan tersebut bertujuan untuk memberikan informasi yang dapat dimanfaatkan bagi pengguna yang ingin melakukan implementasi serupa dan menggunakannya untuk mendeteksi jaringan Wi-Fi.**

Kata Kunci – Confusion Matrix, **Kismet, Raspberry Pi,WIDS, Wi-Fi.**

*) **penulis korespondensi**:Rizky Fachrurozy
Email: rizky.fachrurozy@bssn.go.id

I. PENDAHULUAN

Teknologi wireless merupakan sebuah teknologi gelombang elektromagnetik pada frekuensi tertentu yang berfungsi mengantarkan data atau suara sebagai sarana untuk berkomunikasi [1]. Wi-Fi merupakan sebuah teknologi wireless yang menggunakan spektrum gelombang elektromagnetik untuk menghubungkan area jaringan lokal secara nirkabel atau disebut dengan wireless local area network (WLAN) [1]. Teknologi WLAN diciptakan dan dikembangkan oleh sebuah komite bernama IEEE 802.11 yang berada di bawah organisasi Institute of Electrical and Electronics Engineers (IEEE) yang memiliki tugas untuk membuat, mengembangkan, dan mengatur standar untuk teknologi WLAN atau saat ini dikenal dengan nama dagang Wi-Fi. Hingga tahun 2020, jumlah pengguna Wi-Fi telah mencapai 18,21 Miliar gawai yang terkoneksi pada jaringan Wi-Fi [2].

Berdasarkan prinsip kerjanya, teknologi Wi-Fi merupakan beroperasi pada data link layer dan physical layer pada OSI layer. Wi-Fi memiliki sifat yang terbuka karena menggunakan gelombang elektromagnetik sebagai media transmisi data, hal tersebut menjadi salah satu faktor yang mempermudah serangan terhadap jaringan Wi-Fi, karena penyerangan tidak perlu secara langsung menyentuh fisik dari target [3]. Sampai saat ini jenis serangan yang umum terjadi pada Wi-Fi terdapat tujuh jenis serangan, yaitu *Injection, Eavesdropping, Unauthorized Access*[4]*, Session Hijacking, Fake AP, Man-in-the-middle Attack* dan *Denial of Service* [5]. Serangan–serangan tersebut merupakan serangan terhadap Wi-Fi yang dapat mengancam keamanan informasi dari pengguna yang dapat mengakibatkan tidak terpenuhinya aspek keamanan informasi.

Proses serangan-serangan tersebut dapat dideteksi oleh *Intrusion Detection System* (IDS), IDS merupakan sebuah sistem yang memiliki fungsi untuk memberikan informasi mengenai serangan dengan mengumpulkan informasi dari sistem atau jaringan lalu melakukan analisis informasi tersebut untuk memberikan informasi kemungkinan masalah keamanan [6]. Kismet merupakan sebuah *opensource Intrusion Detection System* yang dapat digunakan untuk mendeteksi serangan pada teknologi *wireless* seperti Bluetooth, Wi-Fi dan beberapa Software Defined Radio (SDR) atau disebut juga dengan

Wireless Intrusion Detection System (WIDS). Kismet memiliki aturan-aturan (*rules*) yang dapat digunakan untuk mendeteksi beberapa jenis serangan pada jaringan *wireless*. Kismet merupakan WIDS yang dapat diterapkan pada sistem operasi berbasis Linux, Mac, dan Windows 10 yang ditulis dengan beberapa bahasa pemograman seperti C++, Java, dan Python [7].

Penerapan WIDS Kismet dilakukan diatas sistem operasi yang berjalan didalam sebuah komputer. Raspberry Pi merupakan sebuah mini *single board computer* yang dikembangkan oleh Raspberry Foundation [8]. Raspberry Pi memiliki fungsi atau kegunaan seperti komputer pada umumnya yang memiliki Processor, RAM dan *onboard* GPU, perangkat tersebut disusun pada sebuah single printed circuit board [9]. Raspberry Pi menggunakan arsitektur *Processor* ARM. Raspberry Pi memiliki ukuran yang ringkas dan konsumsi daya yang rendah dengan fitur dan fungsi yang lengkap seperti PC desktop ataupun laptop.

Permasalahan yang terjadi saat ini adalah ketersediaan perangkat WIDS untuk mendeteksi serangan pada *Wi-Fi Access Point* (AP) masih banyak dijual oleh pengembang besar seperti Cisco, Wathchguard, Airdefense yang dikhususkan untuk jaringan luas bagi perusahaan besar (big enterprise). WIDS tersebut dijual dengan harga yang cukup tinggi, sebagai contoh untuk WIDS dari Cisco Meraki memiliki harga termurah berkisar $790 USD atau setara Rp.11.5 Juta Rupiah. Dengan demikian diperlukan sebuah alternatif perangkat WIDS yang dibuat untuk keperluan deteksi serangan pada Wi-Fi AP untuk pengguna pada perusahaan kecil atau home network yang ingin melakukan identifikasi jaringan Wi-Fi miliknya dari serangan.

Berkaitan dengan permasalahan tersebut penelitian ini melakukan implementasi perangkat WIDS dengan memanfaatkan *opensource* WIDS kismet dengan singleboard computer Raspberry Pi sebagai perangkat deteksi serangan pada Wi-Fi AP yang memiliki biaya sepuluh kali lebih rendah yang berkisar pada $79 USD atau setara Rp. 1.5 Juta Rupiah. Hasil implementasi tersebut selanjutnya akan dilakukan analisis performanya menggunakan metode *confusion matrix*. Diharapkan hasil implementasi WIDS Kismet dan Raspberry Pi tersebut dapat bermanfaat bagi pengguna yang ingin melakukan identifikasi serangan dengan melakukan deteksi pada jaringan Wi-Fi miliknya, sehingga dapat mengurangi dampak yang akan ditimbulkan apabila terjadi serangan..

II. PENELITIAN YANG TERKAIT

A. e-Health Wireless IDS with SIEM Integration.

Penelitian yang berfokus pada implementasi dan penggunaan WIDS pada Raspberry Pi untuk pemantauan jaringan wireless pada peralatan elektronik kesehatan [10]. Hasil pemantauan tersebut akan dimasukan kedalam sebuah *System Information and Event Management* (SIEM), dengan proses deteksi menggunakan IDS Kismet dan Snort. Serangan – serangan yang diuji pada penelitian tersebut yaitu merupakan serangan yang meliputi port scanning, replay attack terhadap peralatan elektronik medis.

B. RaspyAir: Personal Wireless Intrusion Detection System Monitoring using Raspberry Pi.

Penelitian yang melakukan pembuatan sebuah WIDS bernama RaspyAir yang diimplementasikan pada Raspberry Pi. Pembuatan RaspyAir menggunakan cara kerja IDS signature based detection dalam melakukan pendeteksian terhadap serangan yang dapat melakukan deteksi secara realtime [11]. Prototipe RaspyAir ini diciptakan dengan memanfaatkan Tshark dan Airodump-ng. Uji serangan yang dilakukan meliputi lima jenis serangan yang menyerang *availability, access control, confidentiality, integrity* dan *authentication*. Data hasil uji serangan tersebut dianalisis dengan menggunakan metode *Wireless Security Assesment Methodology* (WSAM).

C. Membangun Sensor Wireless Intrusion Detection System pada Raspberry Pi untuk Mendeteksi Rogue Access Point.

Pada penelitian tersebut dilakukan sebuah rancang bangun sensor WIDS dengan melakukan implementasi WIDS Kismet pada Raspberry Pi yang digunakan untuk mendeteksi serangan pada AP dengan jenis serangan yang diujikan adalah Rogue Access point [12]. Hasil dari penelitian tersebut adalah analisis kemampuan kismet dalam mendeteksi sebuah serangan *Rogue Access Point* dengan sensor integrasi kismet dengan Raspberry Pi.

D. Implementasi Sensor WIDS dan Analisa Trafik RTT pada Pendeteksian Rogue Access Point.

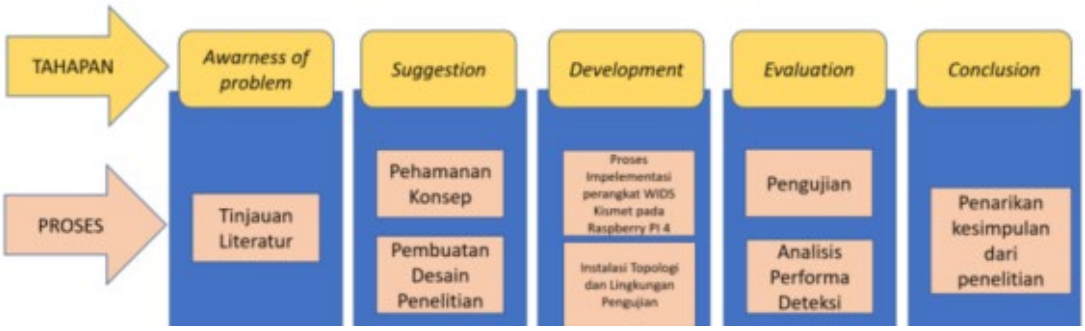
Penelitian yang melakukan implementasi sensor WIDS pada sebuah komputer desktop dengan menggunakan WIDS kismet yang digunakan untuk mendeteksi *serangan Rogue Access Point* serta melakukan analisis terhadap trafik RTT pada proses serangan Access Point [13]. Hasil analisis dari penelitian tersebut adalah analisis kemampuan kismet pada serangan *Rogue Access Point* berserta analisis Trafik RTT.

Beberapa penelitian diatas kecuali penelitian oleh Adhitya akbar [13], semuanya dilakukan dengan perangkat embedded yaitu raspberry pi seperti yang dilakukan dalam penelitian ini. Selain itu penggunaan open source kismet juga sejalan dengan yang dilakukan kecuali pada penelitian zulharim & osman [11] yang menggunakan RaspAir. Berdasarkan objek uji, nespoli & gomez [10] yang menerapkan WIDS pada perangkat medis dan membahas hanya port scaning dan replay attack, reyhand paath [12] untuk mendeteksi evil twin attack sejalan dengan adhitya akbar [13] terakhir zulrahim [11] evil twin dan metode otentikasi WPA bruteforce, ARP request replay, WEP key cracking. Penelitian ini melengkapi deteksi dengan Deauthentication Flood, Rogue Access Point (Evil Twin Attack),WPS Brute Force dan KRACK Attack yang belum pernah diterapkan sebelumnya.

III. METODE PENELITIAN

Penelitian ini menggunakan *Design Science Research* (DSR)[14]. Metodologi DSR terdiri dari lima tahapan yang saling terhubung satu sama lain [13]. Tahapan-tahapan tersebut antara lain *Awareness of Problem, Suggestion, Development, Evaluation* dan *Conclusion* dapat dilihat pada Gbr.1.

ISSN: 2477-5126
e-ISSN:2548-9356



Gbr. 1 Metodologi Penelitian.

A. Awareness of Problem

Tahap ini dilakukan pencarian referensi atau bukti yang dapat mendukung asumsi untuk merumuskan tujuan dari penelitian yang dilakukan secara realistis. Referensi atau bukti terkait ini akan berhubungan langsung dengan penelitian yang akan dilakukan dalam melakukan pemecahan masalah.

a. Serangan pada Wi-Fi

Serangan pada Wi-Fi merupakan serangkaian proses yang dilakukan oleh suatu pihak yang ingin mendapatkan informasi yang terdapat didalam jaringan WLAN. Secara umum serangan pada Wi-Fi terbagi menjadi dua berdasarkan aktivitasnya yaitu serangan pasif dan serangan aktif [5], [15]. Serangan pasif adalah serangan yang dilakukan oleh penyerang untuk mendapatkan informasi ketika informasi tersebut sedang ditransmisikan atau diterima didalam jaringan, serangan ini sulit dideteksi karena biasanya dilakukan tanpa mengubah informasi didalamnya dengan contoh serangan ini adalah analisis lalu lintas jaringan dan penyadapan. Serangan aktif adalah serangan yang dilakukan penyerang dengan tujuan bisa mendapatkan informasi dan dapat mengubah isi konten dari informasi tersebut. Serangan – serangan tersebut merupakan serangan yang mengancam aspek keamanan informasi. Pada Tabel I merupakan tabel mengenai hubungan antara serangan pada Wi-Fi dan ancaman yang ditimbulkan terhadap aspek keamanan informasi beserta kategori serangan berdasarkan referensi [5].

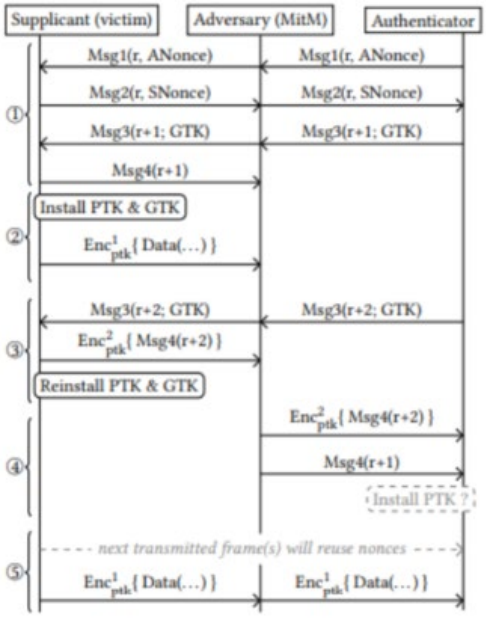
TABEL I
HUBUNGAN SERANGAN WI-FI

Jenis Serangan	**Aspek Keamanan Informasi**	**Dampak**	**Kategori Serangan**
Deuthenticati on Flood	Availability	Wi-Fi AP tidak dapat diakses	Denial of Service
Evil Twin Attack	Data Confidentiality dan Data Integrity	Kebocoran Data (data breach) dan Modifikasi Data	Rogue Access Point dan Man in the Middle Attack (MITM)
WPS Brute Force Attack	Access Control	Mendapatkan akses kontrol illegal	Unauthorized Access
Key Reinstallatio n Attacks (KRACK)	Data Confidentiality dan Data Integrity	Kebocoran Data (data breach) dan Modifikasi Data	Man in the Middle Attack (MITM)

Deauthentication Flood merupakan suatu serangan terhadap Wi-Fi dengan melakukan pengiriman paket *deauth* sebanyak mungkin terhadap Wi-Fi AP sehingga terjadi proses disasosiasi antara pengguna dengan Wi-Fi AP, hal tersebut dapat menyebabkan korban sulit terkoneksi dengan Wi-Fi AP.

Evil Twin Attack merupakan serangan pada Wi-Fi yang memanfaatkan pengetahuan korban terhadap sebuah Wi-Fi AP, penyerang melakukan proses imitasi atau clone terhadap sebuah Wi-Fi AP dengan tujuan mengelabui korban untuk memasuki Clone AP (*evil twin* AP). Korban yang terkoneksi pada Clone AP, lalu lintas datanya dapat dikontrol oleh penyerang. Hal tersebut dapat mengancam kerahasian dan integritas data dari korban, karena seluruh lalu lintas jaringan korban dapat dilihat oleh penyerang.

KRACK atau *Key Reinstallation Attacks* merupakan serangan yang menyerang protokol keamanan WPA2 pada Wi-Fi AP.Serangan KRACK ini dapat menyebabkan kerahasiaan data dan integritas data terancam, karena dengan serangan ini penyerang dapat mengontrol dan melihat seluruh data yang keluar masuk dari pengguna terhadap Wi-Fi AP sehingga seluruh data atau informasi terenkripsi yang bertransmisi pada jaringan wifi tersebut dapat dilihat dalam bentuk teks terang (*plaintext*). Penyerang menggunakan jenis teknik serangan novel attack untuk melakukan proses pemasangan ulang kunci dengan memanfaatkan kelemahan pada proses *4-way handshake* protokol WPA 2 pada Wi-Fi. Proses tersebut membuat penyerang dapat melakukan reset terhadap nonce dengan melakukan pengumpulan dan membalas pengiriman ulang pesan dari tahap *4-way handshake* [16]. Serangan ini secara garis besar ditunjukkan pada Gbr 2.



Gbr. 2 Skema Serangan KRACK attack.

Gbr. 2 merupakan proses serangan dari KRACK attack, tahap pertama merupakan proses manipulasi proses *handshake* dengan melakukan pemblokiran *message* 4 dari korban terhadap otentikator. Tahap kedua setelah proses pengiriman *message* 4 korban akan melakukan install PTK dan GTK dan pada tahap ini korban sudah melakukan proses transmisi informasi. Pada tahap ketiga otentikator mengirimkan transmisi ulang *message* 3 karena tidak menerima *message* 4. Selanjutnya penyerang akan meneruskan transmisi *message* 3 tersebut ke korban yang akan mengakibatkan proses *install* ulang dari PTK dan GTK yang akan terjadi reset terhadap nonce yang digunakan oleh *data-confidetiality protocol* untuk enkripsi. Sehingga dengan begitu penyerang dapat mengontrol penggunaan nonce yang digunakan untuk proses data-confidentiality protocol yang membuat penyerang dapat melakukan dekripsi terhadap proses transmisi pada jaringan WiFi tersebut.

b. Wireless Intrusion Detection System (WIDS)

WIDS merupakan jenis IDS yang melakukan monitor lalu lintas jaringan nirkabel atau *wireless*, seperti WLAN. WIDS melakukan deteksi serangan yang terjadi pada sebuah *Access Point* (AP) dan Station atau perangkat pengguna, deteksi ini dilakukan agar admin dari jaringan nirkabel tersebut dapat mengetahui AP mana saja yang dijadikan target serangan oleh penyerang dalam rangka melindungi informasi didalam jaringan tersebut. Cara kerja IDS dalam melakukan pendeteksian serangan menggunakan beberapa metode antara lain *signature based detection, anomaly based detection* dan *stateful protocol analysis based detection*.

Kismet merupakan *opensource* WIDS yang dapat diterapkan pada sistem operasi berbasis Linux, Mac, dan Windows 10 yang ditulis dengan beberapa bahasa pemograman seperti C++, Java, dan Python [7]. Kismet merupakan IDS yang bekerja pada *Data Link Layer* dan *Physical Layer* dengan memanfaatkan wireless interface card sebagai sensor yang menangkap data jaringan *wireless*.

c. Confusion matrix

Confusion matrix atau matriks konfusi adalah suatu metode untuk melakukan evaluasi pada suatu permasalahan pengklasifikasian suatu objek atau kelas [17]. Ukuran matriks dari suatu kelas atau objek bergantung pada jumlah perbedaan kelas yang dideteksi, dengan tujuan akhir dari metode ini adalah untuk mendapatkan pengukuran performa atau kapabilitas dari suatu objek atau kelas yang akan diukur. Pengukuran yang dapat dilakukan pada *confusion matrix*. Confusion matrix menggunakan empat jenis data sebagai perhitungan yaitu *true positive, false positive, true negative* dan *false negative*.

TABEL III
CONFUSION MATRIX

	Predicted Class		
Actual Class		Yes	No
	Yes	TP	FN
	No	FP	TN

Nilai nilai yang dapat diukur dari *confussion matrix* antara lain akurasi yang merupakan rasio prediksi Benar terhadap data yang bersifat *True positive* dan *True Negative* berbanding dengan jumlah keseluruhan data yang diterima seperti ditunjukkan dalam Formula 1. Presisi yang merupakan pengukuran atau rasio tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Presisi dalam hal ini digunakan untuk mengukur ketepatan antara informasi yang diminta pengguna untuk mendeteksi serangan dengan data yang diberikan oleh sistem WIDS seperti ditunjukkan dalam Formula 2. True *Positive Rate* atau *Recall* merupakan sebuah pengukuran terhadap proporsi data salah yang dianggap anomali terhadap jumlah total data normal yang dikumpulkan dalam sebuah pengujian sepperti ditunjukkan dalam Formula 3. *False positive rate* atau FPR merupakan sebuah pengukuran terhadap proporsi data salah yang dianggap anomali terhadap jumlah total data normal yang dikumpulkan dalam sebuah pengujian.

Ket: Pos= TP+FN Neg= FP + TN

$$\text{Akurasi} = \frac{TP+TN}{Pos+Neg} \quad (1)$$

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (2)$$

$$TPR = \frac{TP}{Pos} \quad (3)$$

$$FPR = \frac{FP}{Neg} \quad (4)$$

Nilai lain yang dapat diukur [18] antara lain adalah *True Negative Rate* merupakan nilai dari data *True Negative* dari hasil bagi dengan seluruh data *Neg* ditunjukkan dalam Formula 5. *False Negative Rate* merupakan nilai rate dari hasil jumlah data *False negative* dibagi dengan jumlah data *Pos* dalam Formula 6. *F-Measure* merupakan nilai perhitungan dari *harmonic mean* antara *presisi* dan *brecall,* dengan jarak nilai antara 0 s.d 1 tergambar dalam Formula 7.

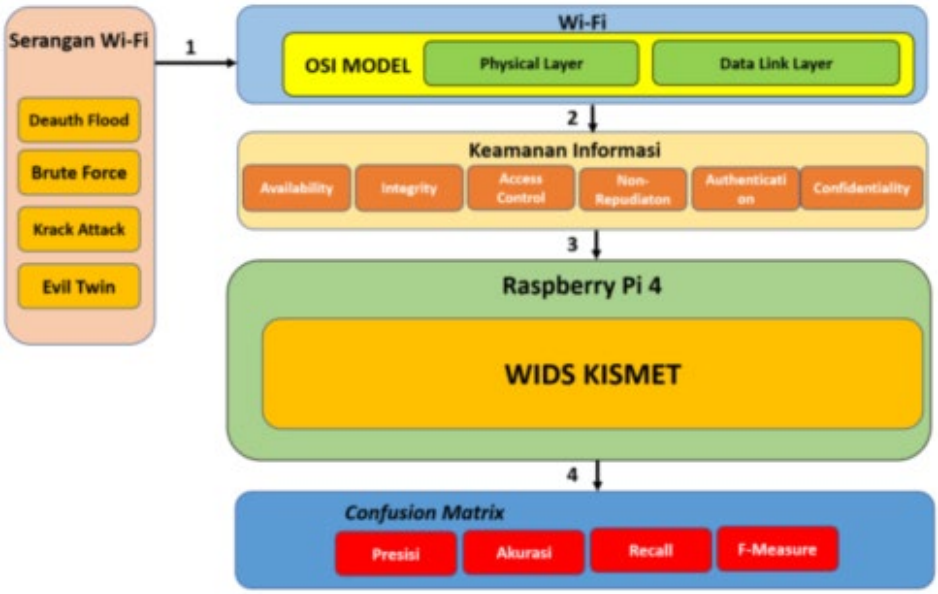
$$TNR = \frac{TN}{Neg} \quad (5)$$

$$FNR = \frac{FN}{Pos} \quad (6)$$

$$F - Measure = 2.\frac{presisi\,.\,recall}{presisi+recall} \quad (7)$$

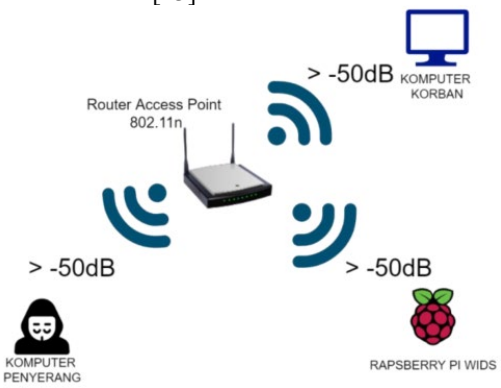
B. Suggestion

ISSN: 2477-5126
e-ISSN:2548-9356



Gbr 3. Kerangka Konseptual

Gbr 3 menunjukkan skenario dan besaran penelitian yang dilakukan. Skenario diatas diperjelas dengan gambar topologi dalam Gbr 4 yang terdiri dari empat perangkat utama yaitu Router Access Point, Penyerang, Korbano dan WIDS pada Raspberry Pi. Setiap perangkat akan diletakan pada rentang jarak > -50 dB (lebih besar dari) yang pada rentang kekuatan sinyal tersebut merupakan kategori "sangat baik", sehingga perangkat Wi-Fi dapat mentransmisikan data dengan kemampuan maksimal [13].



Gbr 4. Topologi Simulasi Penelitian

deauthentication flood attack yang dilakukan pada penelitian ini memiliki urutan sesuai dengan urutan sebagai berikut.

1. Korban atau klien akan terkoneksi kepada Wi-Fi AP yang dijadikan sebagai target serangan.
2. Penyerang melakukan serangan dengan pengiriman paket *deauthentication flood* terhadap Wi-Fi AP yang dijadikan sebagai target serangan, paket *deauthentication flood* yang akan dikirimkan sebanyak 100 paket untuk setiap simulasi serangan pada skenario.
3. WIDS Kismet pada Raspberry Pi sudah berada pada kondisi pemantauan untuk mendeteksi serangan yang terjadi.
4. Korban atau klien terputus dari jaringan Wi-Fi AP yang dijadikan sebagai target serangan *deauthentication flood*.

evil twin attack yang dilakukan hampir sama dengan urutan diatas namun pada tahap 2 Penyerang melakukan pembuatan Wi-Fi AP palsu dan mengirimkan paket *broadcast* serta *deauthentication* sehingga akan membuat korban terputus dari Wi-Fi target. Penambahan tahapan ke lima korban atau klien akan tersambung secara otomatis ke pada Wi-Fi AP palsu yang dibuat oleh penyerang. Skenario *WPS brute force attack* sedikit lebih singkat dari skenario lainnya dengan perbedaan pada tahap pertama dengan melakukan proses *bruteforce* terhadap *access point* dengan tipe keamanan WPS dengan memanfaatkan tools serangan Reaver. Tahap kedua proses *bruteforce* WPS berlangsung dengan waktu sekitar 10 – 12 jam dengan mengirimkan 60.000 Paket serangan sampai bisa berhasil mendapatkan informasi password dari AP. Sebagai penutup tahap ketiga adalah pemantauan sama seperti skenario yang lain. Untuk skenario terakhir yaitu KRACK dijalankan sebagai berikut.

1. Korban atau klien terkoneksi kepada Wi-Fi AP yang dijadikan sebagai target
2. Penyerang akan memutus koneksi korban terhadap Wi-Fi AP, dan membuat Wi-Fi AP clone sehingga korban masuk kedalam Wi-FI AP *clone* tersebut.
3. Ketika korban memasuki Wi-Fi AP clone tersebut, maka penyerang akan mulai melakukan serangan KRACK dengan melakukan *Key Reinstallation Attack* terhadap protokol WPA2.

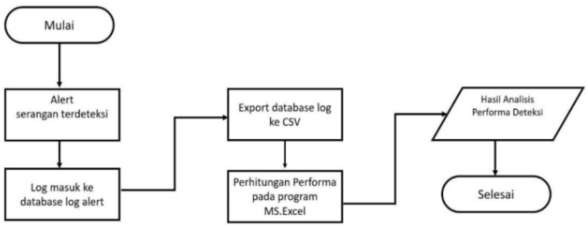
WIDS Kismet pada Raspberry Pi pada kondisi monitor untuk mendeteksi lalu lintas jaringan Wi-FI untuk mendeteksi serangan yang terjadi.

C. Development

Melakukan penerapan dari hasil desain dan topologi pada tahap suggestion. Pada tahap *development* ini akan dilakukan beberapa proses seperti Instalasi dan konfigurasi sensor WIDS

atau proses memasukkan *rules* Kismet, Instalasi dan konfigurasi Wi-Fi AP, Instalasi dan konfigurasi tools serangan.

D. Evaluation



Gbr 5. Diagram Alir Perhitungan dan Analisis Performa

Pada proses analisis performa deteksi dari WIDS dalam mendeteksi serangan seperti dalam Gambar 5 alert dalam bentuk log dalam database berekstensi .kismet diexport kedalam .csv. Log yang telah dikonversi selanjutnya dilakukan perhitungan dengan menggunakan perangkat lunak Microsoft Excel, data-data log tersebut akan dipisahkan secara manual menjadi empat kategori yaitu true positive, false positive, true negative dan false negative. Setelah data berhasil dipisahkan akan dihitung kedalam confusion matrix untuk mengetahui performa. Hasil analisis performa tersebut akan ditampilkan dalam bentuk grafik untuk memudahkan melihat hasil performa deteksinya dan pengambilan kesimpulan.

E. Conclusion

Dilakukan penarikan kesimpulan penelitian berdasarkan hasil proses implementasi dan analisis performa yang telah didapat sebelumnya yang akan dibentuk dalam sebuah grafik.

IV. HASIL DAN PEMBAHASAN

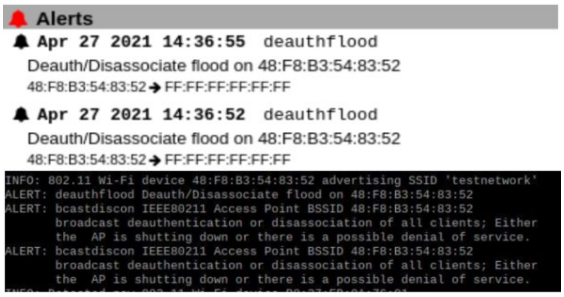
A. Pengujian deteksi Deauthentication Flood Attack

Proses pengujian serangan *deauthentication flood attack* yang dilakukan menggunakan tools serangan aircrack-ng dan deteksi serangan tersebut akan bekerja berdasarkan rules pada Kismet. Kode deteksi untuk serangan ini dapat dilihat pada script di Gbr 6.

```
if (bssid_dot11 != NULL && (dot11info->subtype ==
packet_sub_disassociation ||
      dot11info->subtype == packet_sub_deauthentication)) {
     // if we're w/in time of the last one, update, otherwise
clear
     auto now = time(0);

     if (now - bssid_dot11->get_client_disconnects_last() >
1)
         bssid_dot11->set_client_disconnects(1);
     else
         bssid_dot11->inc_client_disconnects(1);

     bssid_dot11->set_client_disconnects_last(now);
     if (bssid_dot11->get_client_disconnects() > 10) {
         if (d11phy->alertracker->potential_alert(d11phy-
>alert_deauthflood_ref)) {
             std::string al = "Deauth/Disassociate flood on "
+ dot11info->bssid_mac.mac_to_string();

             d11phy->alertracker->raise_alert(d11phy-
>alert_deauthflood_ref, in_pack,
             dot11info->bssid_mac, dot11info->source_mac,
             dot11info->dest_mac, dot11info->other_mac,
              dot11info->channel, al);
```

Gbr 6. *Rules* Deteksi *Deauthentication Flood*

Pada rules tersebut terlihat proses deteksi serangan *deauthentication flood* memiliki sumber data utama dari paket data *disassociation* yang merupakan paket data atau *frame* yang dikirim oleh penyerang terhadap sebuah Wi-Fi AP. Ketika proses serangan terjadi WIDS pada Raspberry Pi akan melakukan deteksi terhadap lalu lintas paket data pada terhadap jaringan Wi-Fi, sehingga serangan deauthentication flood tersebut akan terdeteksi sebagai alert serangan yang ditampilkan seperti pada Gbr 7 dan WIDS Kismet pada Raspberry Pi dapat mendeteksi serangan *deauthentication flood* yang terjadi pada SSID dengan alamat MAC **48:F8:B3:54:83:52** dengan memberikan *alert* **deauthflood** dan **bcastdiscon**.



Gbr 7. Alert Serangan *Deauthentication flood*

B. Pengujian deteksi Evil Twin Attack.

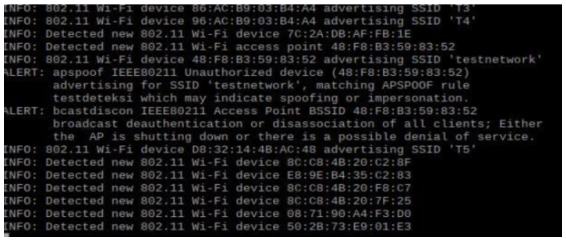
Sama dengan serangan sebelumnya proses serangan dilakukan menggunakan bantuan tools aircrack-ng dan tools Airgeddon-ng untuk membuat Wi-Fi AP palsu. untuk melakukan deteksi serangan evil twin maka dibuat rules seperti ditunjukkan dalam Gbr 8. Pada rules tersebut terlihat proses *compare* atau membandingkan sebuah ssid dengan ssid yang berada pada *whitelist* yang terletak pada fungsi compare_ssid, sehingga ketika terjadi perbedaan antara alamat MAC pada AP *evil twin* dengan AP yang telah di daftarkan pada whitelist akan memicu alert bahwa terdapat sebuah AP spoofing yang mengindikasikan serangan *evil twin attack*.

```
   if (ssid_str.length() != 0 &&
          d11phy->alertracker->potential_alert(d11phy
>alert_ssidmatch_ref)) {
       for (const auto& s : *d11phy->ssid_regex_vec) {
           std::shared_ptr<dot11_tracked_ssid_alert> sa =
              std::static_pointer_cast<dot11_tracked_ssid_alert>(s);

           if (sa->compare_ssid(ssid_str, commoninfo->source)) {
               auto al = fmt::format("IEEE80211 Unauthorized device ({})
advertising "
                "for SSID '{}', matching APSPOOF rule {} which may indicate
"
                "spoofing or impersonation.", commoninfo->source,
                 ssid_str, sa->get_group_name());

               d11phy->alertracker->raise_alert(d11phy->alert_ssidmatch_ref,
in_pack,
                       commoninfo->network,
                       commoninfo->source,
                       commoninfo->dest,
                       commoninfo->transmitter,
                       commoninfo->channel, al);
               break;
           }
       }
   }
```

Gbr 8. *Rules* Deteksi *Evil Twin Attack*

Gbr 9 terlihat bahwa WIDS Kismet pada Raspberry Pi memberika *alert* "apspoof" bahwa terdapat SSID dengan nama "testnetwork" tidak sama dengan alamat MAC yang terdapat pada *rule.*

ISSN: 2477-5126
e-ISSN:2548-9356

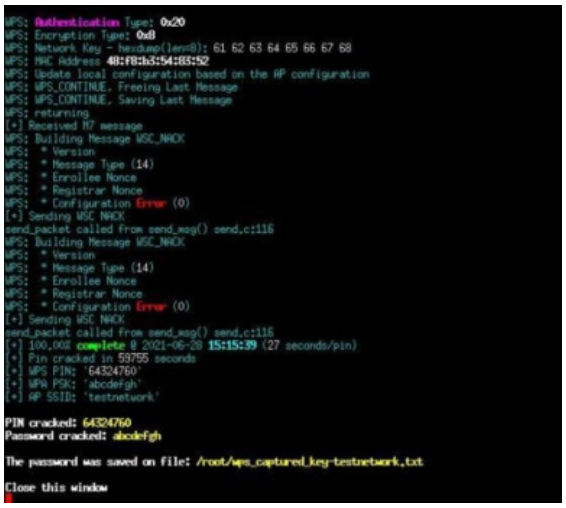


Gbr 9. *Alert* Serangan *Evil Twin*

C. *Pengujian deteksi WPS Bruteforce Attack*

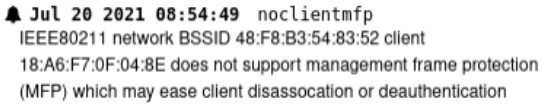
Serangan *Bruteforce* menggunakan bantuan airgeddon-ng yang didalamnya terdapat perangkat lunak Reaver dan *alert* menggunakan kode seperti dalam Gbr 10. Pada bagian *rules* deteksi dari WPS *Bruteforce,* Kismet mendeteksi serangan berdasarkan perhitungan terhadap paket data yang terkirim dari proses serangan, paket data tersebut merupakan perhitungan *m-messages* pada proses otentikasi WPS.

```
        int wps = d11phy->packet_dot11_wps_m3(in_pack);
        if (wps) {
            // if we're w/in time of the last one, update, otherwise clear
            auto now = time(0);

            if (now - source_dot11->get_wps_m3_last() > (60 * 5))
                source_dot11->set_wps_m3_count(1);
            else
                source_dot11->inc_wps_m3_count(1);

            source_dot11->set_wps_m3_last(now);

            if (source_dot11->get_wps_m3_count() > 5) {
                if (d11phy->alertracker->potential_alert(d11phy-
>alert_wpsbrute_ref)) {
                    std::string al = "IEEE80211 AP " + dot11info-
>bssid_mac.mac_to_string() +
                        " sending excessive number of WPS messages which may "
                        "indicate a WPS brute force attack such as Reaver";

                    d11phy->alertracker->raise_alert(d11phy-
>alert_wpsbrute_ref,
                            in_pack,
                            dot11info->bssid_mac, dot11info->source_mac,
                            dot11info->dest_mac, dot11info->other_mac,
                            dot11info->channel, al);
                }

                source_dot11->set_wps_m3_count(1);
            }
        }
    }
```
Gbr 10. *Rules* Deteksi WPS *Brutefroce*



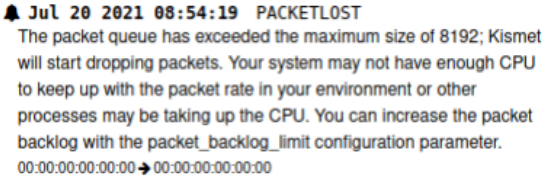
Gbr 11. Hasil serangan WPS Bruteforce

Gbr 11 Merupakan proses ketika serangan WPS bruteforce berjalan selama kurang lebih 10 jam. Ketika proses serangan tersebut berlangsung alert yang diharapkan untuk muncul adalah alert WPSBRUTE dan NOCLIENTMFP, tetapi dari hasil uji simulasi ini Kismet hanya dapat memberikan alert NOCLIENTMFP yang menandakan serangan awal dari WPS bruteforce attack seperti pada Gbr 12.



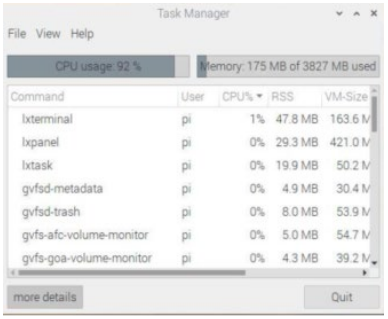
Gbr 12. *Alert* deteksi serangan WPS *Bruteforce*

Pada saat proses deteksi berjalan terdapat alert pada sistem Kismet yang menandakan telah terjadi packet lost atau kehilangan paket yang ditandai dengan alert PACKETLOST seperti pada Gbr 13.



Gambar 13. *Alert Packet Lost* pada proses deteksi

Alert PACKETLOST terjadi akibat ketidakmampuan device untuk mengolah paket data dengan jumlah besar dikarenakan tidak memiliki CPU yang cukup untuk memproses jumlah paket data yang besar tersebut. Hal ini terlihat dari penggunaan CPU sebanyak 92% ketika proses pengolahan data trafik lalu lintas jaringan dari deteksi simulasi serangan WPS *bruteforce*, yang terlihat seperti pada Gbr 14. Hal tersebut menyebabkan hilangnya banyak paket *m-message* yang diperlukan sebagai parameter alert WPSBRUTE, sehingga Kismet tidak dapat melakukan deteksi tersebut karena dalam proses deteksi *rules* kismet memerlukan perhitungan *m-message* yang cukup untuk mendeteksi serangan.



Gambar 14. Penggunaan CPU

D. *Pengujian deteksi KRACK*

Untuk mendeteksi KRACK maka dibuat *rules* kismet seperti dalam Gbr 15. Proses deteksi serangan KRACK yaitu dengan melakukan perbandingan penggunaan WPA2 *nonce*, Kismet akan melakukan pengecekan apakah *nonce* tersebut

ISSN: 2477-5126
e-ISSN:2548-9356

telah digunakan sebelumnya. Penggunaan *nonce* dengan nilai yang sama secara berulang akan terdeksi sebagai proses serangan KRACK. Proses perbandingan tersebut terdapat pada fungsi **get_eapol_nonce_bytes()**. Hasil dari perbandingan tersebut nantinya akan menjadi pemicu dari *alert* "**NONCEREUSE"** yang menandakan terdapat indikasi serangan KRACK *attack* pada lalu lintas jaringan Wi-Fi AP.

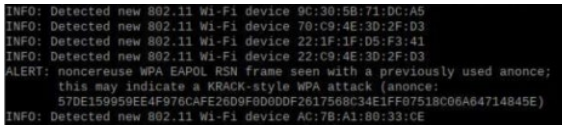
```
    if (eapol->get_eapol_msg_num() == 3 &&
            eapol-
>get_eapol_nonce_bytes().find_first_not_of(std::string("\x00", 1)) !=
            std::string::npos) {
        dupe_nonce = false;
        new_nonce = true;

        for (const auto& i : *(dest_dot11->get_wpa_nonce_vec())) {
            std::shared_ptr<dot11_tracked_nonce> nonce =
                std::static_pointer_cast<dot11_tracked_nonce>(i);

        // If the nonce strings match
        if (nonce->get_eapol_nonce_bytes() == eapol
>get_eapol_nonce_bytes()) {
                new_nonce = false;

                if (eapol->get_eapol_replay_counter() <=
                        nonce->get_eapol_replay_counter()) {

                  // Is it an earlier (or equal) replay counter? Then we
                    // have a problem; inspect the timestamp
                    double tdif =
                        eapol->get_eapol_time() -
                        nonce->get_eapol_time();
```

Gambar 15. *rules* deteksi KRACK

KRACK memanfaatkan penggunaan *Nonce berulang* (*nonce reuse*) dengan perintah dalam Gambar 15. Sesaat setelah KRACK dijalankan, indikasi serangan berhasil dikirimkan adalah ketika muncul pemberitahuan pada terminal dengan keterangan "this is bad". Keterangan tersebut menunjukan paket serangan yang digunakan berhasil menyerang korban dan melakukan skema serangan Krack attack pada korban.

Gbr 16. Serangan *KRACK* ter*install*

Proses monitor dari WIDS yang berjalan sejak awal skenario serangan dimulai akan mendeteksi adanya anomali penggunaan nonce yang terdapat pada paket data, hal tersebut akan membuat WIDS memberikan alert telah terjadi Krack. *Alert* hasil deteksi akan terlihat seperti pada Gbr 17. Kismet akan mendeteksi serangan Krack dan memberikan *alert* "noncereuse" yang menandakan telah terjadi Krack serta tertulis nilai dari nonce yang digunakan kembali untuk KRACK terhadap Wi-Fi AP dengan tipe keamanan WPA atau WPA2.



Gbr 17. *Alert* deteksi KRACK

E. Ekstraksi Log Data Deteksi.

Proses ekstraksi log yang dilakukan menggunakan plugin atau perangkat lunak tambahan yaitu kismet_log_to_csv, plugin tersebut merupakan perangkat lunak berbasis pada linux yang memiliki fungsi untuk melakukan ekstraksi data dari berkas log pada kismet yang berekstensi ".kismet". Gbr 18 adalah hasil ekstraksi log alert deteksi serangan. Gbr 19 hasil ekstraksi log datasource. Gbr 20 hasil ekstraksi log data normal. Hasil data log yang diekstraksi dari proses deteksi keempat serangan yang dilakukan pada uji simulasi penelitian ini menghasilkan data yang memiliki total besaran 11,9 GB. Data tersebut kemudian dilakukan analisis untuk pengklasifikasian sehingga data dapat hitung untuk setiap masing-masing serangan agar menghasilkan nilai perhitungan performa menggunakan metode confusion matrix, perhitungan dilakukan secara spesifik untuk setiap serangan yang dilakukan uji simulasi.

	A	B	C	D	E	F	G
1	ts_sec	ts_usec	phyname	devmac	lat	lon	header
2	1618842592	411782	UNKNOWN	00:00:00:00:00:00	0	0	rootuser
3	1618842608	504035	IEEE802.11	48:F8:B3:54:83:52	0	0	bcastdiscon
4	1618842608	504933	IEEE802.11	48:F8:B3:54:83:52	0	0	bcastdiscon
5	1618842608	512741	IEEE802.11	48:F8:B3:54:83:52	0	0	deauthflood
6	1618842611	2197	IEEE802.11	48:F8:B3:54:83:52	0	0	bcastdiscon
7	1618842611	3494	IEEE802.11	48:F8:B3:54:83:52	0	0	bcastdiscon
8	1618842611	8209	IEEE802.11	48:F8:B3:54:83:52	0	0	deauthflood
9	1618842614	862958	IEEE802.11	48:F8:B3:54:83:52	0	0	bcastdiscon
10	1618842614	867094	IEEE802.11	48:F8:B3:54:83:52	0	0	deauthflood

Gbr 18. Hasil ekstraksi *log alert* deteksi serangan

	A	B	C	D	E
1	Column1	Column2	Column3	Column4	Column5
2	uuid	typestring	definition	name	interface
3	5FE308BD-0000-0000-0000-D85D4C87F590	linuxwifi	wlan1:channel=11,freq_hopping=false	wlan1	wlan1

Gbr 19. Hasil ekstraksi *log datasource*

	A	B	C	D	E	F	G	H	I	J	K
1	ts_sec	ts_us	phynan	sourcemac	destmac	transmac	frequen	devke	la	lon	packet_
2	1,62E+09	880830	IEEE802.11	8C:CB:4B:21:1C:6	8C:CB:4B:21:1C:6	00:00:00:00:00:00	24120000	0	0	0	50
3	1,62E+09	882064	IEEE802.11	8C:CB:4B:21:37:9	76:AC:B9:03:B4:A	00:00:00:00:00:00	24120000	0	0	0	68
4	1,62E+09	185193	IEEE802.11	B0:FC:36:BE:7B:2	B0:FC:36:BE:7B:2	00:00:00:00:00:00	24620000	0	0	0	50
5	1,62E+09	234681	IEEE802.11	94:87:E0:70:63:F	2C:F4:32:2E:07:8	00:00:00:00:00:00	24620000	0	0	0	66
6	1,62E+09	237381	IEEE802.11	48:F8:B3:54:83:5	FF:FF:FF:FF:FF:FF	00:00:00:00:00:00	24620000	0	0	0	275
7	1,62E+09	662150	IEEE802.11	8C:CB:4B:20:F8:4	FF:FF:FF:FF:FF:FF	00:00:00:00:00:00	24120000	0	0	0	134
8	1,62E+09	662448	IEEE802.11	B0:52:16:3B:15:3C	FF:FF:FF:FF:FF:FF	00:00:00:00:00:00	24120000	0	0	0	134
9	1,62E+09	662869	IEEE802.11	8C:CB:4B:21:37:F	01:00:5E:7F:FF:F	00:00:00:00:00:00	24120000	0	0	0	290
10	1,62E+09	663356	IEEE802.11	D0:AB:D5:4E:DF:(	01:00:5E:7F:FF:F	00:00:00:00:00:00	24120000	0	0	0	253

Gbr 20. Hasil ekstraksi *log* data normal

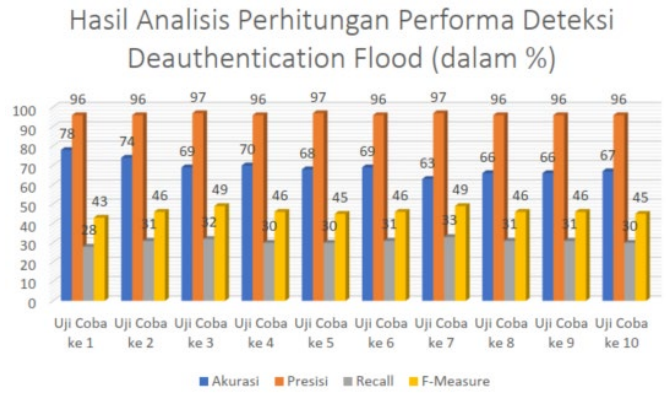
F. Analisis Dan Perhitungan Performa Deteksi Serangan Evil Twin Attack.

TABEL IIIII
CONFUSION MATRIX DETEKSI DEAUTHENTICATION FLOOD ATTACK

Uji Coba Ke -	Hasil deteksi			
	True Positive	True Negative	False Negative	False Positive
1	28	234	72	1
2	31	173	69	1
3	33	123	67	1
4	32	139	72	1
5	33	138	77	1
6	31	130	69	1
7	33	84	67	1
8	31	101	69	1
9	31	105	69	1
10	30	116	70	1

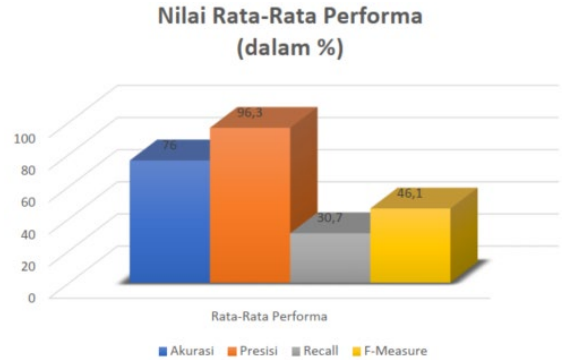
ISSN: 2477-5126
e-ISSN:2548-9356

Data-data pada Tabel III merupakan data hasil klasifikasi log yang hasil ekstraksi yang dilakukan secara manual yang mendefinisikan empat buah kondisi, yaitu data hasil klasifikasi *alert* yang terdeteksi oleh WIDS (*true positive*), data hasil klasifikasi kondisi normal yang terdeteksi oleh WIDS (*true negative*), data hasil klasifikasi alert serangan diluar skenario yang terdeteksi oleh WIDS (*false positive*), data hasil klasifikasi jumlah serangan pada skenario yang tidak terdeteksi oleh WIDS (*false negative*). Hasil perhitungan performa WIDS dalam mendeteksi serangan deauthentication flood berdasarkan perhitungan dengan *confusion matrix* ditampilkan dalam bentuk grafik pada Gbr 21.



Gbr 21. Hasil Perhitungan Performa *Deauthentication Flood*

Hasil dari perhitungan performa yang dilakukan pada sepuluh kali uji simulasi untuk serangan *deauthentication flood attack* memiliki nilai tertinggi untuk akurasi pada 78%, presisi 97%, *recall* 78% dan *f-measure* 45% serta untuk nilai terendah akurasi sebesar 66%, presisi 96%, *recall* 28%, dan *f-meaasure* 43%. Hasil perhitungan data tersebut kemudian dihitung nilai rata-rata yang ditampilkan pada Gbr 22.



Gbr 22. Rata-Rata Performa Deteksi *Serangan Deauthentication Flood*

Rata-rata performa dari deteksi serangan *deauthentication flood attack* yaitu akurasi memiliki nilai 76%, presisi memiliki nilai performa 96.3%, *recall* memiliki nilai performa 30.7%, dan *f-measure* memiliki nilai performa 46.1%. Berdasarkan hasil analisis dan perhitungan tersebut, berikut penjelasan mengenai arti dari nilai-nilai tersebut.

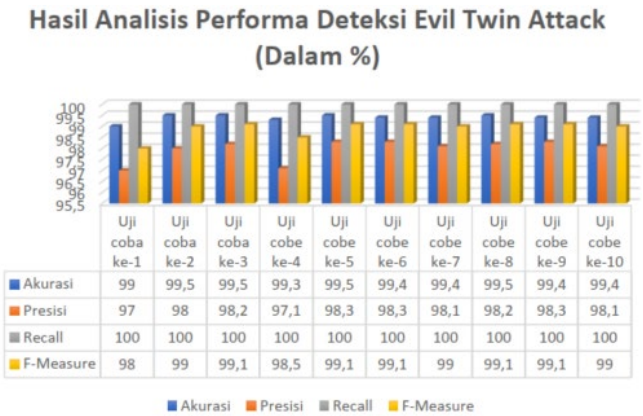
1. Nilai Performa Presisi Merepresentasikan Bagaimana Ketepatan Sistem Dapat Mendeteksi Suatu Keadaan Normal Atau Serangan, Nilai Performa Presisi Pada Proses Deteksi *Deauthentication Flood* Memiliki Kemampuan Tingkat 96.3% Dalam Mendeteksi Suatu Keadaan Normal Atau Serangan.
2. Nilai Performa Akurasi Merupakan Penilaian Bagaimana Sistem Dapat Bekerja Dalam Mendeteksi Seluruh Real Yang Terjadi, Nilai Akurasi Untuk Deteksi *Deauthentication Flood* Memiliki Nilai 76%, Yang Berarti Dari Keseluruhan Kondisi Yang Terjadi Pada Proses Deteksi WIDS Dapat Mendeteksi 76 Kejadian Dari 100 Kejadian Yang Sesungguhnya Terjadi.
3. Nilai Performa *Recall* atau *True Positive Rate* Menjelaskan Bagaimana Sistem Dapat Mendeteksi Serangan Berbanding Dengan Jumlah Paket Serangan Yang Dikirim, Hal Ini Memiliki Arti Dalam Proses Deteksi *Deauthentication Flood* Sistem WIDS Dapat Mendeteksi 30.7 Serangan Dari 100 Serangan Yang Dikirimkan.
4. Nilai Performa *F-Measure* Merupakan Nilai Harmonik Dari Perhitungan Dua Buah Nilai Performa Yaitu Presisi Dan *Recall*, Nilai *F-Measure* Akan Bergantung Dari Ketimpangan Yang Terjadi Antara Antara Presisi Dan *Recall.* Pada Performa Deteksi *Deauthentication Flood* nilai *F-Measure* Berada Pada 46.1%.

G. Analisis Dan Perhitungan Performa Deteksi Serangan Deauthentication Flood Attack.

TABEL IVV
CONFUSION MATRIX DETEKSI EVIL TWIN ATTACK

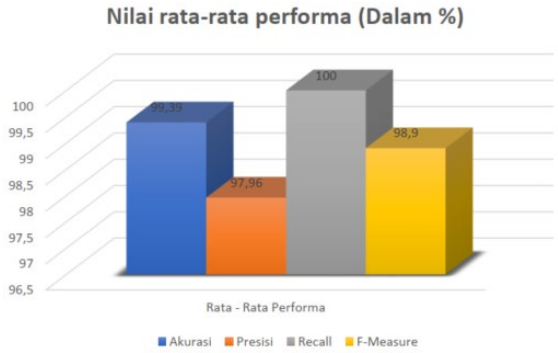
Uji Coba Ke -	Hasil deteksi			
	True Positive	True Negative	False Negative	False Positive
1	31	234	0	1
2	49	151	0	1
3	56	146	0	1
4	34	122	0	1
5	58	143	0	1
6	59	138	0	1
7	52	137	0	1
8	56	143	0	1
9	59	150	0	1
10	53	138	0	1

Data-data pada Tabel IV merupakan data hasil klasifikasi log yang hasil ekstraksi yang dilakukan secara manual. Hasil perhitungan performa WIDS dalam mendeteksi serangan *evil twin attack* berdasarkan perhitungan dengan *confusion matrix* dari data hasil klasifikasi ditampilkan dalam bentuk grafik pada Gbr 22 .



Gbr 22. Hasil Perhitungan Performa *Deteksi Evil Twin Attack*

Hasil dari perhitungan performa yang dilakukan pada sepuluh kali uji simulasi untuk serangan Evil Twin attack memiliki nilai tertinggi untuk akurasi pada 99,5%, presisi, 98,3%, recall 100% dan f-measure 99,1% serta untuk nilai terendah akurasi sebesar 99%, presisi 97%, dan f-meaasure 98%. Hasil perhitungan data tersebut kemudian diambil nilai rata-rata yang ditampilkan pada Gbr 23.



Gbr 23. Rata-Rata Performa Deteksi Serangan *Evil Twin*

Rata-rata performa dari deteksi serangan *evil twin attack* yang dilakukan yaitu akurasi memiliki nilai 99.39%, presisi memiliki nilai performa 97.96%, *recall* memiliki nilai performa 100%, dan *f-measure* memiliki nilai performa 98.9%. Berdasarkan hasil analisis dan perhitungan tersebut, berikut penjelasan mengenai arti dari nilai-nilai tersebut.

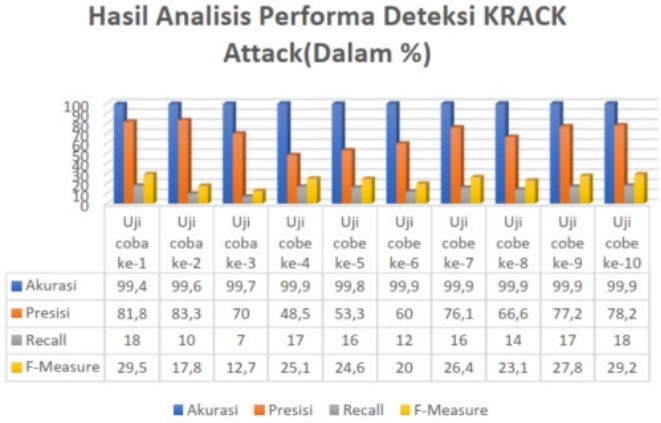
1. Nilai performa presisi merepresentasikan bagaimana ketepatan sistem dapat mendeteksi suatu keadaan normal atau serangan, nilai performa presisi pada proses deteksi *evil twin* memiliki kemampuan tingkat 99.39% dalam mendeteksi suatu keadaan normal atau serangan.
2. Nilai performa akurasi merupakan penilaian bagaimana sistem dapat bekerja dalam mendeteksi seluruh kejadian real yang terjadi, nilai akurasi untuk deteksi *evil twin* memiliki nilai 97.96%, yang berarti dari keseluruhan kondisi yang terjadi pada proses deteksi WIDS dapat mendeteksi 97,96 kejadian dari 100 kejadian yang sesungguhnya terjadi.
3. Nilai performa *recall* atau *true positive rate* menjelaskan bagaimana sistem dapat mendeteksi serangan berbanding dengan jumlah paket serangan yang dikirim, hal ini memiliki arti dalam proses deteksi *evil twin* sistem WIDS dapat mendeteksi 100 serangan dari 100 serangan yang dikirimkan.
4. Nilai performa *F-Measure* merupakan nilai harmonik dari perhitungan dua buah nilai performa yaitu presisi dan *recall*, nilai *F-measure* akan bergantung dari ketimpangan yang terjadi antara antara presisi dan recall. Pada performa deteksi *evil twin* nilai *f-measure* berada pada 98.9%.

H. Analisis Dan Perhitungan Performa Deteksi KRACK.

TABEL V
CONFUSION MATRIX DETEKSI KRACK

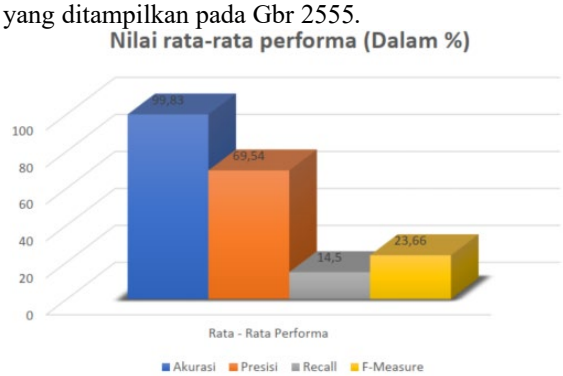
Uji Coba Ke -	Hasil deteksi			
	True Positive	True Negative	False Negative	False Positive
1	18	15478	4	82
2	10	25116	2	90
3	7	45514	3	93
4	17	440690	18	83
5	16	69598	14	84
6	12	101673	8	88
7	16	140118	5	84
8	14	141300	7	86
9	17	166618	5	83
10	18	125373	5	82

Data-data pada Tabel V merupakan data hasil klasifikasi log yang hasil ekstraksi yang dilakukan secara manual. Hasil perhitungan performa WIDS dalam mendeteksi KRCK berdasarkan perhitungan dengan confusion matrix dari data hasil klasifikasi ditampilkan dalam bentuk grafik pada Gbr 24.



Gbr 24. Hasil Perhitungan Performa Deteksi KRACK

Hasil dari perhitungan performa yang dilakukan pada sepuluh kali uji simulasi untuk serangan KRACK memiliki nilai tertinggi untuk akurasi pada 99,9%, presisi 83,3%, *recall* 18% dan *f-measure* 29,5% serta untuk nilai terendah akurasi sebesar 99,4%, presisi 60%, recall 7%, dan *f-measure* 12,7%. Hasil perhitungan data tersebut kemudian diambil nilai rata-rata yang ditampilkan pada Gbr 2555.



Gbr 25. Rata-Rata Performa Deteksi KRACK

Rata-rata performa dari deteksi serangan KRACK yang dilakukan yaitu akurasi memiliki nilai 99.83%, presisi memiliki nilai performa 69,54%, *recall* memiliki nilai performa 14,5%, dan *f-measure* memiliki nilai performa 23.66%. Berdasarkan hasil analisi dan perhitungan tersebut, berikut penjelasan mengenai arti dari nilai-nilai tersebut.

1. Nilai performa presisi merepresentasikan bagaimana ketepatan sistem dapat mendeteksi suatu keadaan normal

atau serangan, nilai performa presisi pada proses deteksi KRACK memiliki kemampuan tingkat 69.54% dalam mendeteksi suatu keadaan normal atau serangan.

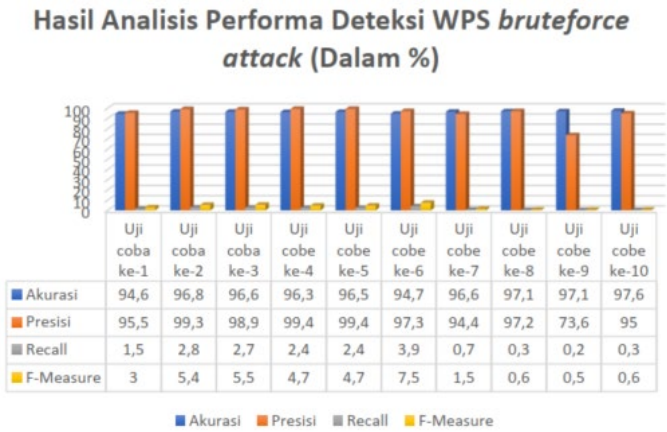
2. Nilai performa akurasi merupakan penilaian bagaimana sistem dapat bekerja dalam mendeteksi seluruh kejadian real yang terjadi, nilai akurasi untuk deteksi KRACK memiliki nilai 99.83%, yang berarti dari keseluruhan kondisi yang terjadi pada proses deteksi WIDS dapat mendeteksi 99.83 kejadian dari 100 kejadian yang sesungguhnya terjadi.
3. Nilai performa *recall* atau *true positive rate* menjelaskan bagaimana sistem dapat mendeteksi serangan berbanding dengan jumlah paket serangan yang dikirim, hal ini memiliki arti dalam proses deteksi KRACK sistem WIDS dapat hanya dapat mendeteksi 14,5 serangan dari 100 serangan yang dikirimkan.
4. Nilai performa *f-measure* merupakan nilai harmonik dari perhitungan dua buah nilai performa yaitu presisi dan recall, nilai *f-measure* akan bergantung dari ketimpangan yang terjadi antara antara presisi dan *recall*. Pada performa deteksi KRACK nilai *f-measure* berada pada 23.66%.

I. Analisis Dan Perhitungan Performa Deteksi Serangan WPS Bruteforce.

TABEL VI
CONFUSION MATRIX WPS BRUTEFORCE ATTACK

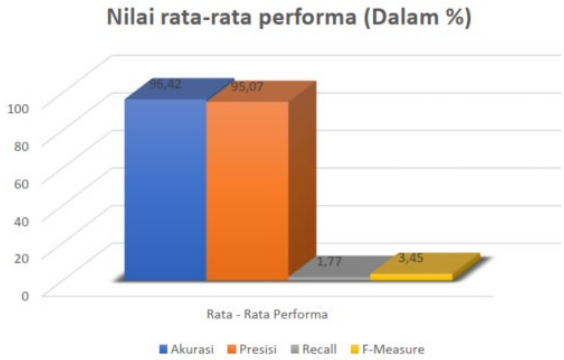
Uji Coba Ke -	Hasil deteksi			
	True Positive	True Negative	False Negative	False Positive
1	935	1048575	39	59065
2	1690	1785675	11	58310
3	1655	1685675	18	58345
4	1472	1652000	8	58528
5	1472	1652113	8	58528
6	2358	1048575	65	57642
7	476	140118	28	59524
8	209	2020935	6	59791
9	159	1903250	57	59841
10	210	2450941	11	59790

Data-data pada Tabel VI merupakan data hasil klasifikasi log yang hasil ekstraksi yang dilakukan secara manual. Hasil perhitungan performa WIDS dalam mendeteksi serangan WPS *bruteforce attack* ditampilkan dalam bentuk grafik terlihat pada Gbr 26.



Gbr 26. Hasil Perhitungan Performa Deteksi WPS *bruteforce attack*

Hasil dari perhitungan performa yang dilakukan pada sepuluh kali uji simulasi untuk serangan WPS *bruteforce attack* memiliki nilai tertinggi untuk akurasi pada 97,1%, presisi 99,3%, *recall* 3,9% dan *f-measure* 7,5% serta untuk nilai terendah akurasi sebesar 94,6%, presisi 73,6%, recall 0,2%, dan *f-measure* 0,5%. Hasil perhitungan data tersebut kemudian diambil nilai rata-rata yang ditampilkan pada Gbr 27.



Gbr 27. Rata-Rata Performa Deteksi WPS *bruteforce attack*

Rata-rata performa dari deteksi WPS *bruteforce attack* yang dilakukan yaitu akurasi memiliki nilai 96,42%, presisi memiliki nilai performa 95,07%, *recall* memiliki nilai performa 1,77%, dan *f-measure* memiliki nilai performa 3,45%. Berdasarkan hasil analisis dan perhitungan tersebut, berikut penjelasan mengenai arti dari nilai-nilai tersebut.

1. Nilai performa presisi merepresentasikan bagaimana ketepatan sistem dapat mendeteksi suatu keadaan normal atau serangan, nilai performa presisi pada proses deteksi WPS *bruteforce attack* memiliki kemampuan tingkat 95,07% dalam mendeteksi suatu keadaan normal atau serangan.
2. Nilai performa akurasi merupakan penilaian bagaimana sistem dapat bekerja dalam mendeteksi seluruh kejadian real yang terjadi, nilai akurasi untuk deteksi WPS *bruteforce attack* memiliki nilai 96.42%, yang berarti dari keseluruhan kondisi yang terjadi pada proses deteksi WIDS dapat mendeteksi 99.83 kejadian dari 100 kejadian yang sesungguhnya terjadi.
3. Nilai performa *recall* atau *true positive rate* menjelaskan bagaimana sistem dapat mendeteksi serangan berbanding dengan jumlah paket serangan yang dikirim, hal ini memiliki arti dalam proses deteksi WPS *bruteforce attack* sistem WIDS dapat hanya dapat mendeteksi 1,77 serangan dari 100 serangan yang dikirimkan.
4. Nilai performa *f-measure* merupakan nilai harmonik dari perhitungan dua buah nilai performa yaitu presisi dan *recall*, nilai *f-measure* akan bergantung dari ketimpangan yang terjadi antara antara presisi dan *recall.* Pada performa deteksi WPS *bruteforce attack* nilai *f-measure* berada pada 3.45%.

V. KESIMPULAN

- Proses Implementasi WIDS Kismet pada Raspberry Pi 4 dapat dilakukan mampu mendeteksi serangan yang diujikan yaitu *deauthentication flood, evil twin*, WPS *bruteforce*, dan KRACK. Untuk serangan WPS *bruteforce attack* hanya dapat mendeteksi satu jenis *alert* dari dua *alert* yang diharapkan, disebabkan kurangnya kemampuan CPU Raspberry Pi 4 dalam mengolah paket data yang besar.
- Implementasi sistem ini dapat mengurangi biaya hingga 10 kali lebih rendah dibandingkan dengan penggunaan WIDS yang dikeluarkan oleh Cisco Meraki yang memiliki fungsi sama untuk mendeteksi serangan pada Wi-Fi AP.
- Hasil implementasi memiliki performa deteksi paling efektif terhadap serangan *evil twin attack*. Hal tersebut ditunjukan dari hasil analisis performa deteksi serangan menunjukan deteksi memiliki nilai performa *recall* mencapai 100%, performa tersebut menunjukan perangkat WIDS dapat mendeteksi seluruh paket serangan yang dikirimkan. Hasil performa tersebut juga sama dengan penelitian sebelumnya pada penelitian terkait yang menerapkan WIDS pada Raspberry Pi 3B+ dalam mendeteksi *rogue access point (evil twin)*.
- Hasil analisis performa menunjukan deteksi paling tidak efektif terjadi pada proses deteksi serangan WPS bruteforce attack karena memiliki nilai performa *recall* 1,77%, performa tersebut menunjukan WIDS hanya dapat memberikan 1 *alert* deteksi dari 100 paket serangan yang dikirimkan.
- Penelitian mendatang masih dapat dilengkapi untuk perangkat wireless lain seperti bluetooth melalui "uberetooth" dan melakukan integrasi deengan ELK stack untuk memudahkan dalam proses analisisnya.
- Penelitian mendatang dapat digunakan penyimpanan yang lebih besar agar durasi deteksi menjadi lebih lama serta menggunakan antena monitor dengan seri Wi-Fi terbaru (802.11ax).

UCAPAN TERIMA KASIH

Terima kasih kepada seluruh author yang berpartisipasi baik dalam perumusan ide hingga analisis dan penyajian datanya serta tidak luput kepada Politeknik Siber dan Sandi Negara untuk bantuan *publication fee*.

DAFTAR PUSTAKA

[1] J. A. Jill West, Tamara Dean, *Network+ Guide to Networks 8th Edition*, 8th ed. Boston: Cengage Learning, 2018.

[2] T. Alsop, "WLAN connected devices worldwide 2016-2021," *Statista*, 2020. https://www.statista.com/statistics/802706/world-wlan-connected-device/ (accessed Jul. 20, 2021).

[3] K. Pahlavan and P. Krishnamurthy, "Evolution and Impact of Wi-Fi Technology and Applications: A Historical Perspective," *Int. J. Wirel. Inf. Networks*, vol. 28, no. 1, pp. 3–19, 2021, doi: 10.1007/s10776-020-00501-8.

[4] M. F. F. A. S. Y. Irawan, "The Wireless Attack Menggunakan Tools Aircrack Pada Kali Linux Untuk Melakukan WPA Attack," *J. Lentera*, vol. 20, no. 1, pp. 63–74, 2021.

[5] M. Waliullah and D. Gan, "Wireless LAN Security Threats & Vulnerabilities," *Int. J. Adv. Comput. Sci. Appl.*, vol. 5, no. 1, pp. 176–183, Jan. 2014, doi: 10.14569/IJACSA.2014.050125.

[6] D. Rozenblum, "Understanding Intrusion Detection Systems," Rockville, 2001.

[7] "Kismet - Wi-Fi, Bluetooth, RF, and more." https://www.kismetwireless.net/ (accessed Dec. 20, 2022).

[8] G. Halfacree, *The Official Raspberry Pi Beginner's Guide: How to use your new computer*. Raspberry Pi Press, 2020.

[9] "Raspberry Pi 4 Model B specifications – Raspberry Pi." https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/ (accessed Dec. 20, 2022).

[10] P. Nespoli, D. Useche Pelaez, D. Díaz López, and F. Gómez Mármol, "COSMOS: Collaborative, Seamless and Adaptive Sentinel for the Internet of Things.," *Sensors (Basel).*, vol. 19, no. 7, Mar. 2019, doi: 10.3390/s19071492.

[11] M. S. A. M. Z. Mohd Nizam Osman, "RaspyAir : Self-Monitoring System for Wireless Intrusion Detection using Raspberry Pi," *J. Comput. Res. Innov.*, vol. 1, no. 1, pp. 14–21, 2016.

[12] W. R. M. S. W. I. D. S. pada R. P. untuk M. R. A. P. M. S. W. I. D. S. pada R. P. untuk M. R. A. P. A. I. Paath, "Membangun Sensor Wireless Intrusion Detection System pada Raspberry Pi untuk Mendeteksi Rogue Access Point Membangun Sensor Wireless Intrusion Detection System pada Raspberry Pi untuk Mendeteksi Rogue Access Point," Salatiga, 2016.

[13] M. A. AKBAR, "Implementasi sensor WIDS dan analisa trafik RTT pada pendeteksian rogue access point," Depok, 2012.

[14] A. Dresch, D. P. Lacerda, and J. A. V Antunes, *Design Science Research: A Method for Science and Technology Advancement*. Springer International Publishing, 2016. [Online]. Available: https://books.google.co.id/books?id=LSNgvgAACAAJ

[15] M. Keerthika and D. Shanmugapriya, "Wireless Sensor Networks: Active and Passive attacks - Vulnerabilities and Countermeasures," *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 362–367, 2021, doi: https://doi.org/10.1016/j.gltp.2021.08.045.

[16] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce Reuse in WPA2," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 1313–1328, 2017, doi: 10.1145/3133956.3134027.

[17] L. R. Adedeji B. Badiru, *Handbook of Measurements*. London: CRC Press, 2017.

[18] Gowtham S R, "Confusion Matrix to No Confusion Matrix in Just 5mins," *Medium*, 2022. https://pub.towardsai.net/confusion-matrix-179b9c758b55 (accessed Jan. 23, 2023).