Jurnal TIKomSiN, Vol. 10, No. 2, Oktober 2022 ISSN Cetak : 2338-4018
DOI: https://doi.org/10.30646/tikomsin.v10i2.639 ISSN Online : 2620-7532

Implementasi KRACK dan KRACK Detector terhadap Wpa_Supplicant pada Perangkat Android dan Linux Ubuntu

Dozy Arti Insani[1)]; Nanang Trianto[2)]; Dimas Febiyan Priambodo[3*)]
[1)]Badan Siber dan Sandi Negara
[2,3)]Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara
[1)]dozy.arti@bssn.go.id; [2)]nanang.trianto@bssn.go.id ; [3)]dimas.febriyan@poltekssn.ac.id

ABSTRACT

Nowadays WiFi services is available in many public places to accesses the information. Most of WiFi services use the Wifi Protected Access 2 (WPA2) security. This protocol uses a 4-way handshake mechanism for authentication process, but there is a weakness by using the 4-way handshake mechanism that possible to re-install the key (KRACK). This weakness can be used by attackers to hold up communications so that attackers can enter the network. This study, KRACK analysis was carried out on wpa_supplicant Android and Linux Ubuntu which connected to WiFi and the WPA2 security protocol to know the weaknesses. The analysis is doing on the user's device by connecting to the Rogue AP generated by the Vanhoef script. This dangerous line is compared to normal line. Analyzing attacks on 4-way handshakes, this research create implementation of KRACK Detector and the result of detection will be used to get the KRACK characteristics. The information getting from the result can prevent the disadvantages which coming by attacks. The validation of the research was carried out by using Wireshark to make sure that the third massage sends which show threat of attack.

Keywords*: 4-way handshake, KRACK Detector, WiFi, WPA2*

I. PENDAHULUAN

Perkembangan pesat teknologi informasi menempatkan sistem informasi sebagai elemen penting dalam aktivitas sehari-hari [1], [2]. Internet sebagai jaringan komputer global mempunyai fungsi yang penting dalam tersedianya informasi yang dibutuhkan [3] dan dapat diakses salah satunya menggunakan *Wireless Fidelity* (WiFi). Teknologi WiFi telah memperlihatkan perkembangan yang sangat signifikan dalam beberapa tahun terakhir. Hal ini terbukti dengan banyaknya fasilitas umum yang menawarkan *hotspot area* (area yang terdapat jaringan internet berbasis WiFi) yang dapat diakses oleh semua orang baik secara gratis ataupun berbayar [4]. Selain kecepatan koneksi yang relatif cepat, WiFi juga menawarkan harga yang terjangkau untuk perimeter area yang luas[3]. Namun dibalik semua kelebihan yang ditawarkan, teknologi WiFi juga terdapat celah keamanan dimana WiFi tidak menjamin segala hal terkait aspek autentikasi atau enkripsi [5]

Teknologi WiFi memiliki beberapa protokol keamanan yaitu WEP, WPA, WPA2, dan yang terbaru WPA3. Sebagian besar layanan WiFi menggunakan protokol keamanan *WiFi Protected Access 2* (WPA2) dan *4-way handshake* yang didefinisikan pada standar protokol jaringan *wireless* 802.11i [6]. WPA2 merupakan protokol yang dikembangkan untuk menjawab solusi dari kerentanan pengamanan WiFi sebelumnya, yakni WEP dan WPA. Namun seiring dengan berkembangnya teknologi, ditemukan kerentanan dalam protokol keamanan WPA2 yaitu adanya proses penginstalan ulang kunci yang sudah digunakan. Serangan ini disebut *Key Reinstallation Attacks* (KRACK) di mana penyerang dapat melakukan *interception* informasi *personal* pada *range* suatu *Access Point* (AP) [6]. KRACK berbahaya terhadap versi wpa_supplicant 2.6 dan di atasnya bagi pengguna Linux, dan Android 6.0 dan di atasnya karena menerapkan aturan untuk menghapus kunci enkripsi dari memori setelah dipasang untuk pertama kalinya [7]. Hal ini membuat penyerang dengan mudah melakukan *intercept* dan memanipulasi lalu lintas yang dikirim oleh perangkat Linux dan Android. KRACK memanfaatkan kerawanan pada 802.11i yakni pada proses *4-way handshake* dimana kunci yang digunakan pada *4-way handshake* untuk memastikan pengguna dan WiFi memiliki kunci yang sama dipakai kembali pada enkripsi *traffic* data antara perangkat pengguna dengan WiFi [6]. Kunci inilah yang ditargetkan oleh penyerang untuk melakukan KRACK, dimana seharusnya kunci yang aman adalah kunci yang hanya boleh digunakan satu kali namun tidak dijamin pada protokol WPA2. KRACK sangat efektif

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

dilakukan pada sistem operasi yang menggunakan wpa_supplicant versi 2.6 [6]. Mekanisme *4-way handshake* ini memiliki celah keamanan dimana penyerang dapat memaksa korban untuk menggunakan *all zero encryption key* yang dapat diprediksi sebelumnya [7].

Publikasi literatur mengenai KRACK oleh Vanhoef pada Oktober 2017, kebanyakan pengguna mengira bahwa penyedia perangkat WiFi akan selalu melakukan pembaharuan dengan melakukan *patching* sehingga perangkat tidak lagi rawan oleh serangan. Namun pada praktiknya masih banyak *Access Point* (AP) yang tidak mendukung atau tidak dapat melakukan *patching* [8]. Terlebih lagi tidak adanya peringatan atau deteksi ancaman serangan mengakibatkan pengguna sulit untuk melakukan mitigasi pada AP yang rawan terhadap KRACK. Deteksi terhadap KRACK perlu dilakukan untuk melakukan identifikasi AP agar terhindar dari segala kemungkinan kerugian yang terjadi.

Penelitian ini melakukan implementasi KRACK *Detector* untuk mendeteksi KRACK dan analisis KRACK terhadap wpa_supplicant Android dan Linux Ubuntu yang terkoneksi WiFi dengan protokol WPA2 yang hasilnya digunakan untuk mencari karakteristik KRACK sebagai informasi pengguna sehingga dapat mengambil langkah untuk mengantisipasi KRACK.

II. TINJAUAN PUSTAKA

2.1 PMK, PTK dan GTK

PMK (*Pairwise Master Key*) merupakan kunci urutan tertinggi yang berasal dari PSK (*Pre Shared Key*). Kunci ini digunakan sebagai autentikasi pengguna yang dibuat antara AP dan *supplicant*. Nilai PMK dan nilai acak yang dihasilkan baik dari AP dan *supplicant* (Nonce) digunakan untuk mendapatkan kunci baru, yang disebut dengan PTK. Gabungan PMK dan Nonce milik AP dan *supplicant* menghasilkan PTK. Kunci ini dibagi menjadi beberapa kunci yakni: satu untuk menandatangani pesan *4-way handshake*; satu untuk mengamankan paket data yang dikirimkan antara *supplicant* dan AP; dan satu untuk mengenkripsi kunci grup ke *supplicant* selama *4-way handshake* berlagsung. GTK (*Group Temporal Key*) berasal dari GMK yang pertama kali dikirim oleh *authenticator* kepada *supplicant* dalam proses *4-way handshake* [9]. AP akan memeroleh GTK baru secara berkala dari GMK dan menggunakan *2-way handshake* untuk mengirimkan pada semua *supplicant* pada jaringan. Bersama dengan pesan ketiga, dikirimkan pula GTK dari *authenticator* kepada *supplicant*. Pada jaringan WPA2, pengguna memerlukan GTK untuk menerima *multicast* dan melakukan *broadcast frame* yang dienkripsi menggunakan kunci ini.

2.2 Key Reinstallation Attack (KRACK)

KRACK merupakan serangan yang menyalahgunakan desain atau kelemahan implementasi dalam protokol kriptografi untuk menginstal ulang kunci yang sudah digunakan [6]. Serangan dilakukan terhadap proses *handshaking* pada *4-way handshake* di mana penyerang mengelabui korban agar menginstal kembali kunci yang sudah digunakan. Hal ini dicapai dengan memanipulasi dan membalas pesan *handshaking*. Terkhusus serangan terhadap Android 6.0 dan Linux berdampak luar biasa di mana serangan ini memaksa pengguna untuk menggunakan *all-zero encryption key* (kunci enkripsi semua nol) yang dapat diprediksi sebelumnya [8]. Proses autentikasi protokol keamanan WPA2 menggunakan mekanisme *4-way handshake* dan protokol *Extensible Authentication Protocol Over* LAN (EAPOL) yang merupakan protokol yang digunakan pada *filter* Wireshark untuk mengetahui proses autentikasi yang terjadi.

a. Mekanisme 4-way *Handshake*

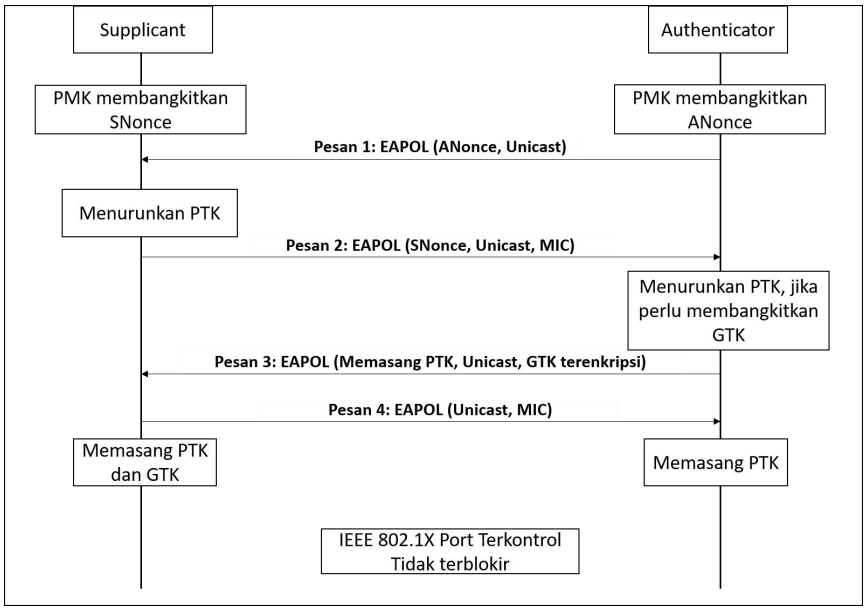
Mekanisme *4-way handshake* dalam Gambar 1 diawali dengan *authenticator* mengirimkan pesan *EAPOL-Key* 1 yang berisi ANonce yang dihasilkan secara acak dan memiliki *reply counter* r. Kemudian, *supplicant* memperoleh PTK dari ANonce dan SNonce dan mengirimkan pesan *EAPOL-Key* 2 yang berisi SNonce. Pesan ini juga berisi informasi permintaan *pairwise* dan *group cipher suites* [8]. Selain itu, *message-2* ini berisi MIC dan *replay counter* r. *Authenticator* memeroleh PTK dari ANonce dan SNonce kemudian memvalidasi MIC dalam pesan *EAPOL-Key* 2 yang diterima. Setelah validasi, *authenticator* mengirim pesan *EAPOL-Key* 3 yang berisi ANonce sama seperti pada pesan 1 dan informasi dari pesan *beacon* atau *probe response* tentang *pairwise* dan *group cipher suites*. Berisi MIC untuk menunjukkan apakah akan menginstal *temporal key* dan GTK yang dienkapsulasi atau tidak, selain itu juga berisi *replay counter* +1. *Supplicant* mengirimkan pesan *EAPOL-Key* 4 untuk mengonfirmasi

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

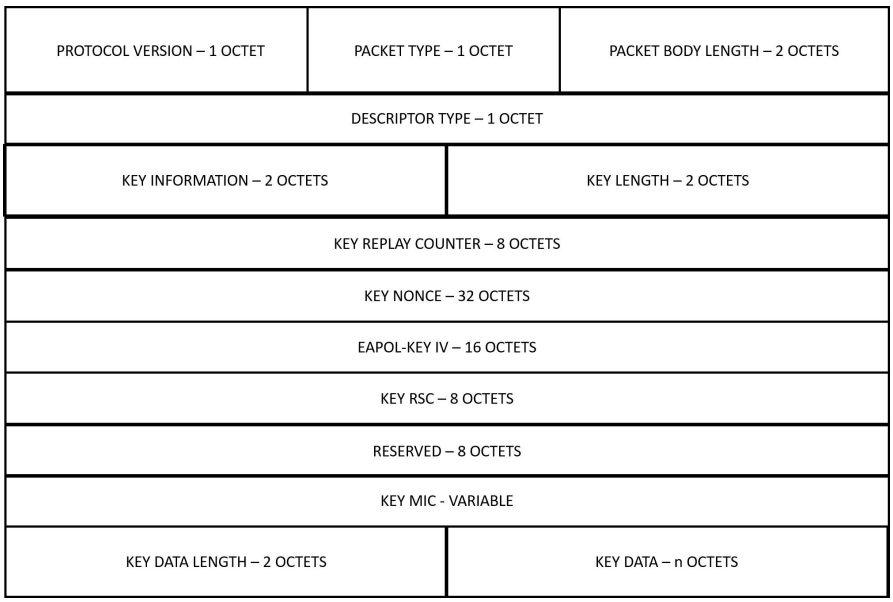
bahwa *temporal key* berhasil dipasang. Pesan ini juga mengandung *reply counter* r+1.

b. Extensible Authentication Protocol Over LAN (EAPOL)

Sebuah *eapol-key frame* seperti dalam Gambar 2 merupakan tipe dari EAPOL *frame* yang digunakan untuk melakukan pertukaran informasi kunci kriptografi antara *supplicant* dan *authenticator* [8]. Karakteristik dari kunci ditentukan dari *flags* yang disimpan pada *field Key Information.* Format dari EAPOL-*Key frame* ditunjukkan pada Gambar 2.



Gambar 1. Mekanisme 4-way *handshake*



Gambar 2. EAPOL

2.3 WPA_Supplicant

WPA_supplicant merupakan implementasi dari komponen *supplicant* WPA yang berjalan di sisi pengguna. Wpa_suppllicant mengimplementasikan negosiasi kunci WPA dengan WPA *Authenticator* dan autentikasi EAP dengan *Authentication Server*. Selain itu juga mengontrol *roaming* dan autentikasi IEEE 802.11/asosiasi *driver* LAN nirkabel [10]. Wpa_supplicant dirancang menjadi program *daemon* yang berjalan secara tidak terlihat dan bertindak sebagai komponen *backend* yang mengendalikan koneksi nirkabel [10].

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

2.4 KRACK Detector

KRACK *Detector* adalah *script* menggunakan bahasa pemrograman Python untuk mendeteksi kemungkinan serangan terhadap perangkat pengguna dalam suatu jaringan [11]. *Script* ini nantinya dijalankan pada sisi pengguna yang bekerja dengan cara melakukan *monitoring interface* WiFi dan menunggu pesan ketiga dari proses *4-way handshake*.

2.5 Penelitian Terkait

a. *Mitigation of Key Reinstallation Attacks in WPA2 WiFi Networks by Detection of Nonce Reuse*

Penelitian atau eksperimen yang dilakukan oleh Naitik et al pada tahun 2018. Penelitian ini mengusulkan solusi keamanan jabat tangan (*handshaking*) dengan cara menangkap dan menganalisis paket untuk mencegah penggunaan kembali seperti yang terjadi ketika dilakukan instalasi ulang [12].

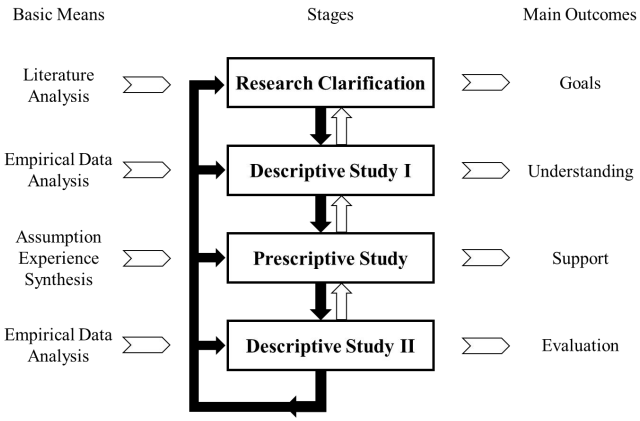
b. *KRACKCover: A Wireless Security Framework for Covering KRACK Attacks*

Merupakan penelitian yang dilakukan oleh Tommy Chins dan Kaiqi Xiong pada tahun 2018. Penelitian ini mengusulkan suatu kerangka kerja (*framework*) untuk mendeteksi adanya ancaman *Key Reinstallation Attacks* (KRACK) dengan memberikan peringatan agar pengguna WiFi dapat meningkatkan keamanan privasinya. *Framework* ini mendeteksi serangan dengan sementara mengalihkan koneksi pengguna ke lokasi perantara seperti *portal splash* dan menghilangkan kebutuhan pengguna untuk menginstal perangkat lunak tambahan pada sistem komputer mereka untuk menerima peringatan.

c. *A Software-defined Networking-based Detection and Mitigation Approach against KRACK*

Penelitian yang dilakukan oleh Yi Li et al. mengusulkan kerangka yang memanfaatkan karakteristik pengontrol SDN (*Software-Defined Networking*) untuk memantau dan mengelola lalu lintas jaringan WiFi. *Framework* ini terdiri dari dua modul utama, yakni deteksi dan mitigasi. Kerangka ini menerapkan SDN ke WiFi AP di mana komunikasi jarak pendek WiFi dikonversi menjadi komunikasi jarak jauh ke internet menggunakan SDN. Konversi data komunikasi ini diproses oleh sakelar SDN di bawah pengelolaan SDN pengontrol. Dalam modul deteksi yang diusulkan *framework*, pengontrol SDN memeriksa setiap permintaan jaringan WiFi yang masuk di mana duplikasi *message-3* dari *4-way handshake* akan terdeteksi saat seorang penyerang meluncurkan KRACK. Disebutkan bahwa *framework* ini secara efisien mendeteksi dan mengurangi KRACK [13].

III. METODE PENELITIAN



Gambar 3. Design Research Methodology

Penelitian ini menggunakan metode Design Research Methodology (DSR) seperti dapat dilihat pada Gambar 3 yang terdiri dari looping 4 tahapan antara lain research clarification, descriptive study I, prescreptive study dan descriptive studi II.

3.1 Research Clarification

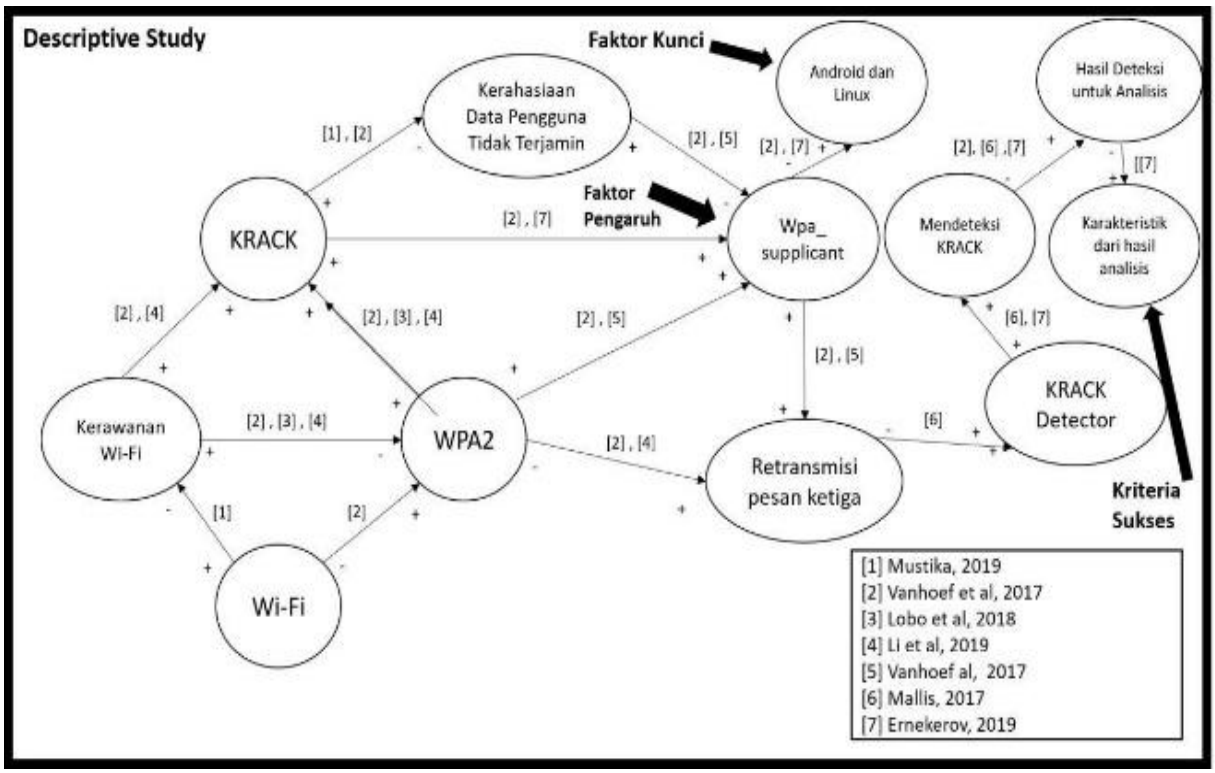
Dilakukan studi literatur untuk mengidentifikasi tujuan yang diharapkan pada penelitian. Studi literatur yang dilakukan memiliki tujuan untuk menentukan fokus penelitian yaitu melakukan implementasi KRACK *Detector* untuk mendeteksi KRACK dan melakukan analisis KRACK terhadap wpa_supplicant Android dan Linux yang terkoneksi WiFi dengan protokol WPA2. Hasil analisis akan digunakan untuk mencari karakteristik KRACK sebagai informasi pengguna sehingga dapat mengambil langkah untuk mengantisipasi KRACK.

3.2 Descriptive Study 1

Mendapatkan referensi pendukung untuk menjawab permasalahan dengan menentukan kriteria kesuksesan, faktor kunci, dan faktor pengaruh yang menunjukkan keterkaitan dengan permasalahan yang ada. Pada umumnya wpa_supplicant versi 2.4 dan 2.5 digunakan pada Linux [7] di mana versi wpa_supplicant ini rawan terhadap *all zero encryption key*. Ketika perangkat melakukan

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

all zero encryption key pada penginstalan ulang kunci maka akan memudahkan penyerang dalam melakukan dekripsi paket. Berbeda ketika menyerang perangkat lain, akan lebih sulit untuk mendekripsi paket karena tidak melakukan *all zero encryption key* meskipun sebagian besar paket tetap dapat didekripsi seperti dapat dilihat pada Gambar 4.



Gambar 4. *Descriptive Study*

3.3 Prescriptive Study

Dilakukan penjabaran dari desain penelitian yang telah ditentukan pada tahap Descriptive Study 1. Berdasarkan studi literatur yang dilakukan, untuk mencapai kriteria sukses dan memenuhi tujuan penelitian yang ditentukan, agar dapat mendeteksi adanya ancaman KRACK maka perlu diimplementasikan suatu metode deteksi atau framework agar dampak serangan dapat dicegah ataupun diminimalisir. Melakukan implementasi script Vanhoef untuk melakukan uji kerentanan perangkat terhadap ancaman KRACK serta mengimplementasikan

KRACK Detector sebagai alternatif tools yang dapat digunakan untuk melakukan deteksi KRACK sesuai dengan skenario.

3.4 Descriptive Study 2

Tahap ini merupakan tahapan terakhir dari serangkaian tahap DRM. Pada tahap ini dilakukan evaluasi mengenai implementasi *script* Vanhoef dalam melakukan pengujian perangkat dan implementasi KRACK *Detector* dalam melakukan deteksi ancaman serangan.

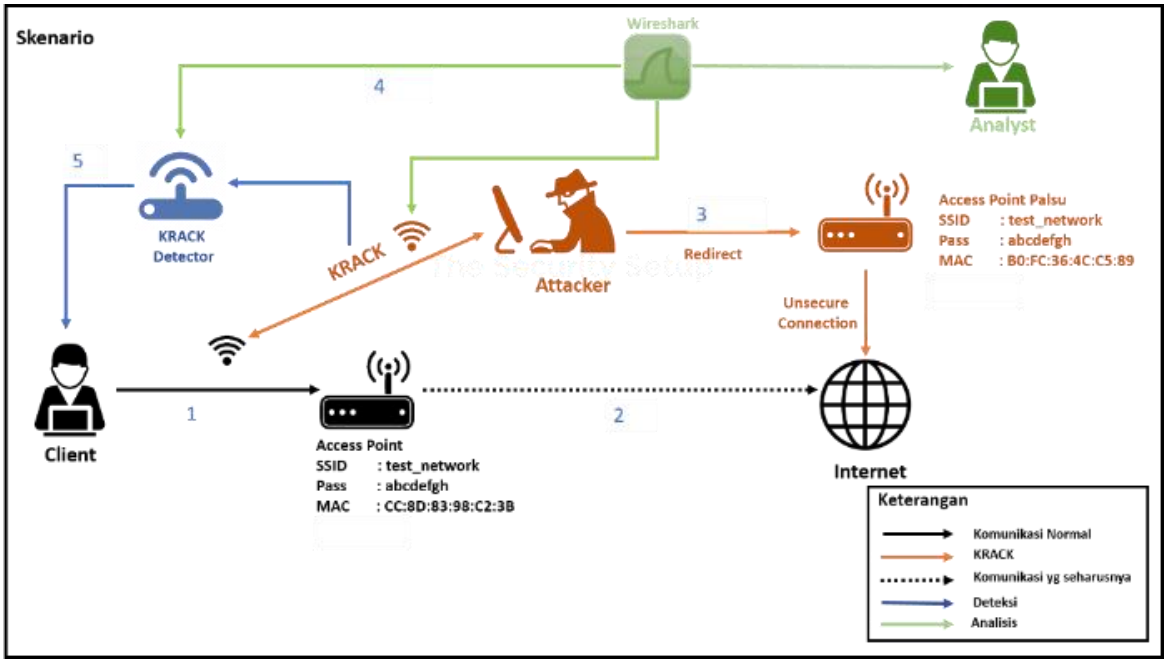
IV. HASIL DAN PEMBAHASAN

Pemodelan penelitian adalah sebagai berikut:

a. Baik pengguna dan jaringan yang digunakan harus dalam perimeter jangkauan penyerang.
b. Kekuatan sinyal *Rogue AP* harus lebih baik daripada WiFi asli untuk dapat mengecoh pengguna agar terkoneksi pada *Rogue AP.*
c. Pengguna pernah melakukan koneksi pada WiFi dengan SSID “test_network” dan *password* “abcdefgh” di mana perangkat pengguna dapat mengingat WiFi tersebut.

Pemodelan penelitian yang telah dibuat, diperlihatkan pada Gambar 5:

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532



Gambar 5 Skenario Kerawanan

Selain persyaratan diatas terrdapat keterbatasan implementasi *script* Vanhoef, sebagai berikut:

a. Implementasi *script* Vanhoef tidak dapat dilakukan jarak jauh melainkan pengguna dan jaringan yang digunakan harus dalam jangkauan penyerang.
b. *Rogue AP* yang dibangkitkan oleh *script* Vanhoef tidak dapat digunakan untuk mengakses internet.
c. Serangan yang dilakukan tidak melanggar atau merusak sifat keamanan dari proses *4-way handshake*.
d. Adanya gangguan dalam implemetasi *script* Vanhoef akan mengurangi keandalannya.

Script Vanhoef tidak benar-benar melakukan serangan, hanya melakukan pengujian kerentanan perangkat terhadap serangan KRACK.

4.1 Research Clarification

Berdasarkan hasil studi literatur yang dilakukan, KRACK merupakan suatu serangan Man In the Middle Attack (MiTMA) di mana penyerang dapat memanfaatkan kerawanan proses autentikasi dengan membalas pesan handshaking pada mekanisme 4-way handshake. Namun serangan ini tidak terjadi pada semua perangkat Android dan Linux, melainkan hanya pada perangkat Android dan Linux yang menggunakan versi wpa_supplicant tertentu. Mengingat dampak yang dapat ditimbulkan akibat dari eksploitasi serangan ini, script serangan tidak pernah dipublikasikan demi mencegah terjadinya penyalahgunaan oleh pihak yang tidak bertanggung jawab. Namun Vanhoef mempublikasikan suatu script yang dapat digunakan untuk melakukan uji kerentanan perangkat terhadap serangan KRACK. Sama seperti prinsip serangan KRACK, *script* ini memanfaatkan proses handshaking yang dilakukan dengan melakukan retransmisi pesan ketiga sehingga terjadi lebih dari satu kali proses handshaking. Oleh sebab itu, dalam penelitian ini dilakukan implementasi KRACK dan KRACK Detector dengan skenario dan penerapan pada perangkat tertentu untuk mendapatkan hasil berupa hasil deteksi dan analisis yang bermanfaat bagi pengguna.

4.2 Descriptive Study 1

Sistem secara umum dimulai dari proses awal pengguna melakukan koneksi pada WiFi asli, kemudian terkecoh dengan adanya *Rogue AP* sehingga terkoneksi pada *Rogue AP*, hingga adanya retransmisi pesan ketiga terdeteksi oleh KRACK *Detector*. Kemudian berlanjut ke skenario terstruktur menggunakan *data flow* serangan KRACK antara *supplicant* dan *authenticator. Data flow* menunjukkan variabel yang sedang melakukan komunikasi yakni *supplicant* dan *authenticator* di mana memungkinkan adanya penyerang (*Man in*

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

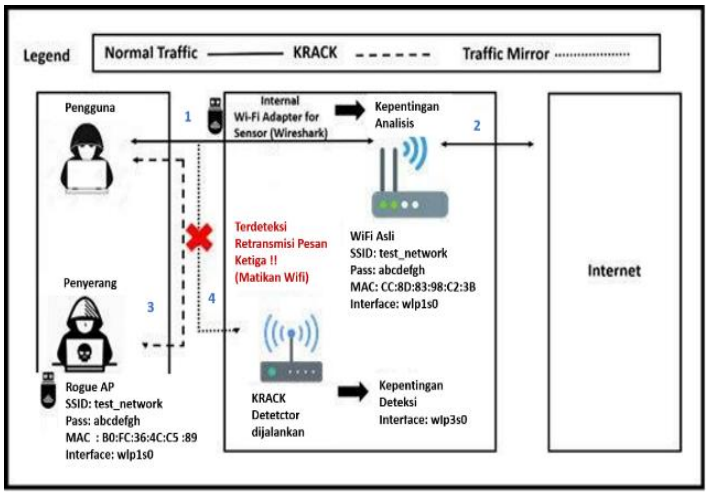
The Middle Attack) di antara komunikasi yang berlangsung.

a. Gambaran Umum

Gambar 6 menjelaskan gambaran umum sistem dengan penjelasan sebagai berikut:

1) Pengguna melakukan koneksi pada WiFi asli dengan SSID "test_network" dan *password* "abcdefgh".
2) Kondisi pertama di mana pengguna terkoneksi pada WiFi yang aman maka pengguna dapat langsung terhubung ke internet.
3) Kondisi kedua *script* Vanhoef dijalankan di mana *script* ini berfungsi sebagai *Rogue AP* dan pengguna melakukan koneksi pada *Rogue AP* dengan SSID dan *password* yang sama seperti WiFi asli.

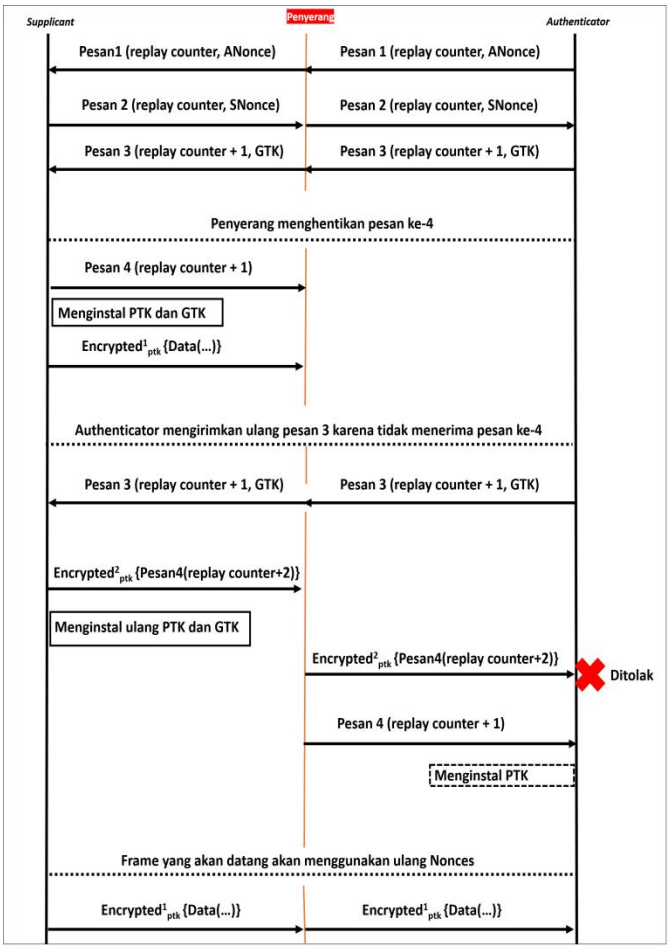
Capturing paket menggunakan Wireshark pada kondisi pertama dan kedua. Deteksi adanya KRACK dilakukan oleh KRACK *Detector*. Deteksi serangan yang berjalan pada sisi pengguna kemudian dilakukan analisis lebih lanjut untuk mendapatkan karakteristik deteksi serangan.



Gambar 6 Gambaran Umum

b. *Data Flow*

Data flow dimulai dari awal proses komunikasi untuk dapat melakukan koneksi hingga terjadi kehilangan pesan dan dilakukannya retransmisi pesan yang menunjukkan adanya ancaman serangan seperti ditunjukkan oleh Gambar 7.



Gambar 7. Data Flow

4.3 Prescriptive Study

a. Lingkungan Implementasi

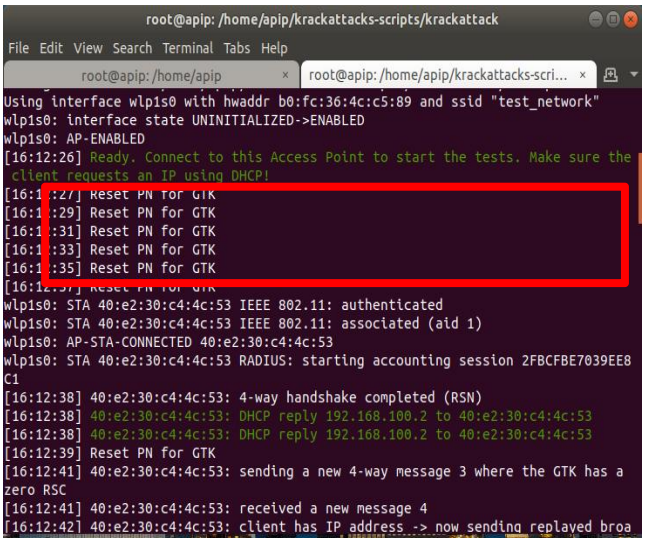
Lingkungan implementasi yang digunakan berupa *Access Point,* dua buah laptop, empat buah *smartphone* Android dan sistem operasi Linux Ubuntu versi 16 dan 18.

1. Laptop :
   - Merk : ASUS X441UA dan Lenovo idepad 5
   - Processor: Intel Core i3-6006U dan AMD A9-9420
   - RAM: 4.00GB dan 8.00GB
   - Sistem Operasi: Linux Ubuntu versi 16 dan 18
   - Versi wpa_supplicant : 2.4 dan 2.6

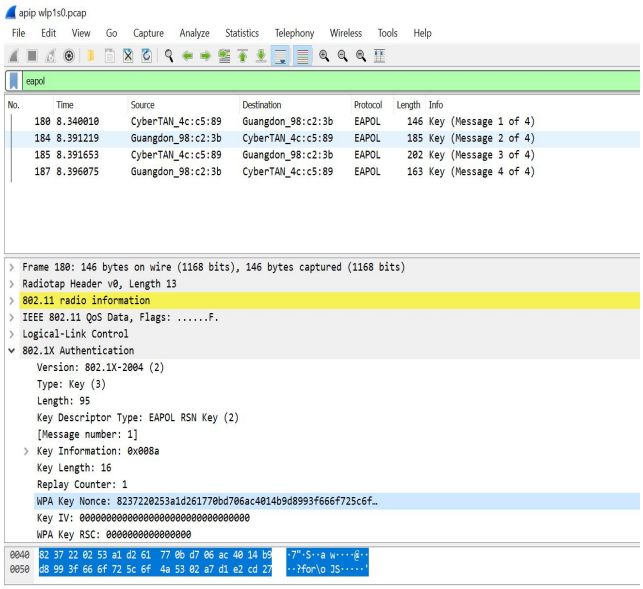
2. Android :
   - Merk : OPPO A37, Huawei GR 3, Samsung SM-G355H, dan Samsung GT-I9060
   - Sistem Operasi : Android (Lolipop, Kitkat, dan Jelly Bean)
   - RAM : 2GB dan 3GB

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

b. Kondisi Masuk Jaringan yang Aman



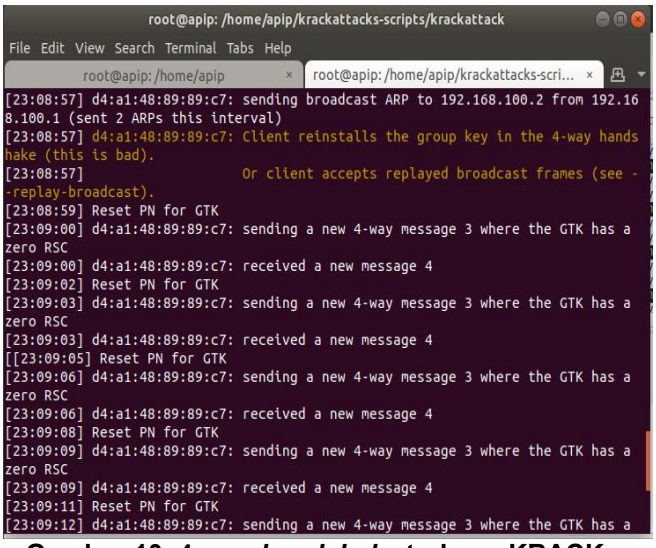
Gambar 8. Informasi Jaringan Aman



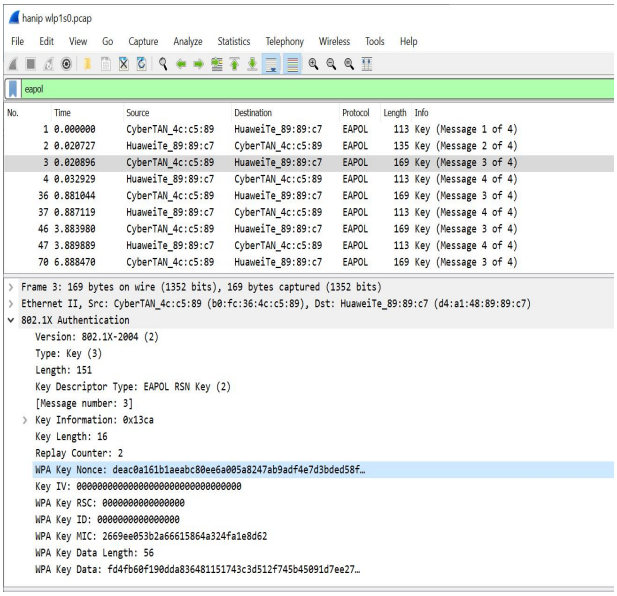
Gambar 9. *Normal 4 Way Handshake*

Kondisi masuk jaringan yang aman ini terjadi ketika pengguna terhubung dengan WiFi asli. Apabila terhubung pada WiFi yang aman, KRACK *Detector* tidak menyatakan adanya retransmisi pesan ketiga yang menunjukkan bahwa WiFi yang digunakan aman dari ancaman serangan seperti dalam gambar 7. Kemungkinan adanya serangan dapat dilihat dari adanya retransmisi pesan ke-3 dan WPA *Key Nonce* yang digunakan saat transmisi. Kondisi pengguna memasuki jaringan yang aman ini dapat dilihat menggunakan Wireshark seperti terlihat pada Gambar 9.

c. Kondisi Masuk Jaringan Tidak Aman



Gambar 10. 4 way-*handshake* terkena KRACK



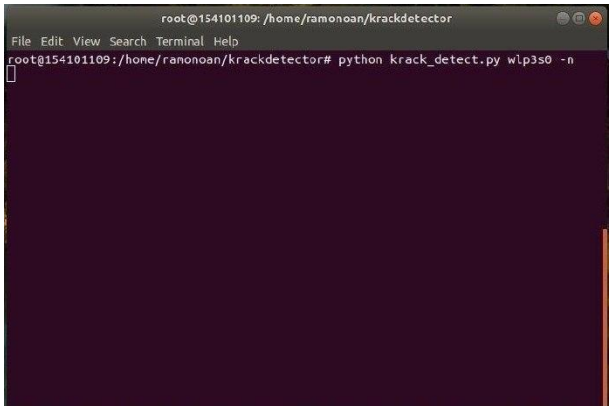
Gambar 11. *Malicious 4-way handshake*

Berbeda ketika pengguna terhubung pada WiFi yang tidak aman, KRACK *Detector* akan memunculkan keterangan bahwa WiFi yang digunakan rawan terhadap serangan KRACK. Adanya retransmisi pesan ke-3 dan WPA *Key Nonce* yang digunakan berulang pada saat retransmisi pesan ketiga memungkinkan adanya serangan KRACK dengan salah satu contoh notifikasi berupa kerawanan *reinstall* GTK dan aman dari *reinstall* PTK, dan *traffic* Wireshark yang dapat dilihat menggunakan Wireshark seperti pada Gambar 10 dan 11. Kondisi dalam jaringan tidak aman ini

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

menggunakan Android versi 5.1 dari produk Huawei R3.

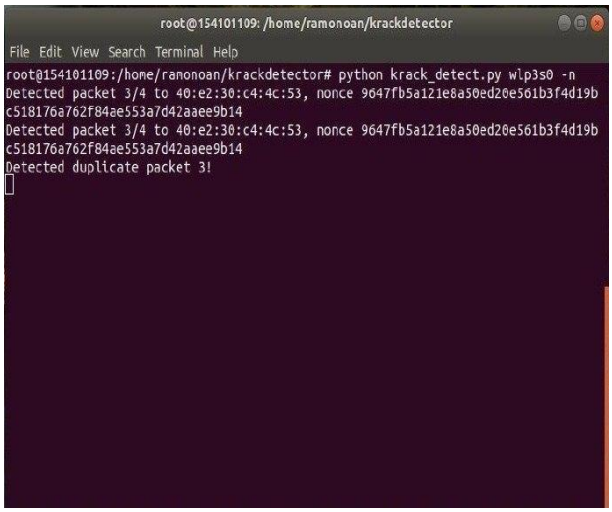
d. Deteksi Masuk Jaringan Aman



Gambar 12 KRACK D*etector* tidak mendeteksi KRACK

Setelah pengguna terkoneksi pada WiFi, KRACK *Detector* dijalankan untuk melakukan deteksi apakah WiFi yang sedang digunakan rawan terhadap ancaman KRACK atau tidak. KRACK *Detector* dijalankan pada sisi pengguna seperti terlihat pada Gambar 12 menunjukkan bahwa WiFi yang digunakan merupakan WiFi yang aman. Ketika terhubung pada WiFi yang aman, KRACK *Detector* yang dijalankan tidak memunculkan hasil apapun dikarenakan menunggu adanya retransmisi pesan ke-3 yang menunjukkan bahwa terdapat ancaman KRACK.

e. Deteksi Masuk Jaringan Tidak Aman



Gambar 13 KRACK *Detector* mendeteksi KRACK

KRACK *Detector* dijalankan untuk melihat ancaman serangan yang ada. Proses di atas seperti pada Gambar 12 apabila terdapat retransmisi pesan ke-3 pada proses *4-way handshake* maka KRACK *Detector* akan memunculkan notifikasi yang menunjukkan bahwa WiFi yang digunakan tidak aman.

4.4 Descriptive Study 2

Kondisi ideal diimplementasikannya sistem baik *script* Vanhoef dan KRACK *Detector* yakni ketika syarat kesuksesan dapat terpenuhi dengan baik sehingga implementasi dalam penelitian dapat berjalan sesuai yang diharapkan. Dengan melakukan implementasi *script* Vanhoef, pengguna dapat mengetahui kerentanan perangkat terhadap ancaman KRACK sekaligus mengetahui variasi bentuk ancaman KRACK. Sedangkan implementasi KRACK *Detetctor* dapat memverifikasi implementasi dari *script* Vanhoef itu sendiri dan tentunya dapat memberikan peringatan kepada pengguna bahwa terdapat ancaman serangan berupa adanya retransmisi pesan ketiga. Hasil implementasi *script* Vanhoef dan KRACK *Detector* dilakukan analisis untuk mendapatkan suatu analisis serangan, karakteristik deteksi, kesimpulan, dan saran terhadap hasil dari seluruh tahapan penelitian.

Menurut teori yang dikemukakan oleh Vanhoef, terdapat beberapa karakteristik dari eksploitasi yang sukses dilakukan berupa *4-way handshake* pesan ketiga yang dikirim ulang pada *4-way handshake* dengan *replay counter* yang bertambah. pengguna akan menerima pesan ketiga secara terus menerus dan menerimanya karena *replay counter* juga bertambah. *PeerKey handshake*: *4-way handshake* yang digunakan dalam fase kedua pada *PeerKey handshake* diserang dengan cara yang sama. *Group key handshake:* Retransmisi pesan ketiga akan mengirimkan ulang GTK. Dengan demikian bersama pesan ketiga yang baru akan dilakukan inisialisasi ulang sesuai *replay counter* GTK di sisi pengguna. Berdasarkan hasil yang telah diperoleh pada tahap implementasi dan analisis pada beberapa perangkat laptop dan *smartphone* Android dengan versi *wpa_supplicant* maka diperoleh hasil seperti yang ditunjukkan pada Tabel 1.

Tabel 1 Kerawanan Berdasar OS dan Wpa_Supplicant

Perangkat	OS / Versi Wpa_Supplicant	Reinstall PTK	Reinstall GTK
Notebook Acer Aspire E 14	Ubuntu 16.04 / v2.4	Not Vurnerable	Not Vurnerable
Notebook Asus	Ubuntu 18.04 / v2.6	Not Vurnerable	Not Vurnerable

ISSN Cetak : 2338-4018
ISSN Online : 2620-7532

Perangkat	OS / Versi Wpa_Supplicant	Reinstall PTK	Reinstall GTK
X441 UV			
Android Huawei R3	Lolipop 5.1 / -	Vurnerable	Vurnerable
Android Oppo A37	Lolipop 5.1 / -	Not Vurnerable	Not Vurnerable
Android Samsung G355H	Kitkat 4.4.2 / -	Vurnerable	Not Vurnerable
Android Samsung GT-I9060	Jelly Bean 4.2.2 / -	Vurnerable	Not Vurnerable

V. PENUTUP

Implementasi Key Reinstallation Attacks (KRACK) dapat dilakukan dengan mengoneksikan perangkat Android dan Linux pada Rogue AP yang dibangkitkan oleh script Vanhoef sehingga kerentanan perangkat terhadap KRACK didapatkan.

Implementasi KRACK Detector dilakukan pada sisi pengguna dengan tujuan untuk mendeteksi KRACK pada jaringan WiFi dengan protokol WiFi Protected Access 2 (WPA2) sehingga dapat memberikan informasi kepada pengguna untuk mengambil langkah pencegahan agar terhindar dari kerugian yang mungkin ditimbulkan.

Berdasarkan hasil penelitian tidak ditemukan kerawanan KRACK pada versi wpa_supplicant 2.4 Linux Ubuntu 16 dan versi wpa_supplicant 2.6 Linux Ubuntu 18 karena telah dilakukan update.

DAFTAR PUSTAKA

[1] B. Kurniawan, “Sistem Pakar Diagnosa Penyakit Paru Pada Anak,” *J. TIKomSiN*, vol. 5, no. 2, pp. 53–60, 2017.

[2] A. N. Puteri, F. Zakiyabarsi, and D. F. Priambodo, “Metode Prototype Pada Sistem Informasi Manajemen Tugas Akhir Mahasiswa Berbasis Website,” *J. Teknol. Inf. dan Komun. Sinar Nusant.*, vol. 10, no. 1, pp. 1–8, 2022, doi: http://dx.doi.org/10.30646/tikomsin.v10i1.606.

[3] R. Setiawan, “Penggunaan Internet sebagai Teknologi Informasi di Kalangan Mahasiswa Ekonomi Akuntansi Universitas Muhammadiyah Surakarta,” 2009.

[4] M. R. Arief, “Teknologi Jaringan Tanpa Kabel (Wireless),” vol. 2007, no. November, pp. 1–8, 2007.

[5] F. Mustika, “Implementasi Purwarupa Intrusion Detection System untuk Mendeteksi Evil Twin Attack pada Jaringan Nirkabel Menggunakan Skema Agarwal et al.,” 2019.

[6] 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.

[7] M. Vanhoef, F. Piessens, and K. U. Leuven, “Key Reinstallation Attacks : Forcing Nonce Reuse in WPA2,” 2017.

[8] J. Ernekerov, “Analysis and Detection of KRACK Attack against WiFi Infrastructure,” 2019.

[9] L. K. Mæhlum, “KrackPlus Sammendrag av Bacheloroppgaven,” 2018.

[10] J. Malinen, “Linux WPA/WPA2/IEEE 802.1X Supplicant,” 2013. .

[11] O. Mallis, “securingsam/krackdetector,” *github.com*, 2017. .

[12] N. S.T, R. Lobo, P. S. Vernekar, and V. G. Shetty, “Mitigation of Key Reinstallation Attack in WPA2 Wi-Fi networks by detection of Nonce Reuse,” *Int. Res. J. Eng. Technol.*, vol. 05, no. 05, 2018.

[13] Y. Li, M. Serrano, T. Chin, K. Xiong, and J. Lin, “A software-defined networking-based detection and mitigation approach against KRACK,” *ICETE 2019 - Proc. 16th Int. Jt. Conf. E-bus. Telecommun.*, vol. 2, pp. 244–251, 2019, doi: 10.5220/0007926202440251.