PENGEMBANGAN *FRAMEWORK* OTOMATISASI MITIGASI KASUS *ROGUE ACCESS POINT* PADA JARINGAN *WIRELESS* IEEE 802.1X

TESIS

Diajukan sebagai Salah Satu Syarat
untuk Memperoleh Gelar Pascasarjana
Magister Informatika



Disusun oleh:

Nama : Ninki Hermaduanti

NIM : 14917129

MAGISTER INFORMATIKA
PASCASARJANA FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2016

PENGEMBANGAN *FRAMEWORK* OTOMATISASI MITIGASI KASUS *ROGUE ACCESS POINT* PADA JARINGAN *WIRELESS* IEEE 802.1X

TESIS

Diajukan sebagai Salah Satu Syarat
untuk Memperoleh Gelar Pascasarjana
Magister Informatika



Disusun oleh:

Nama : Ninki Hermaduanti

NIM : 14917129

MAGISTER INFORMATIKA
PASCASARJANA FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2016

LEMBAR PENGESAHAN PEMBIMBING

PENGEMBANGAN *FRAMEWORK* OTOMATISASI MITIGASI KASUS *ROGUE ACCESS POINT* PADA JARINGAN *WIRELESS* IEEE 802.1X

TESIS



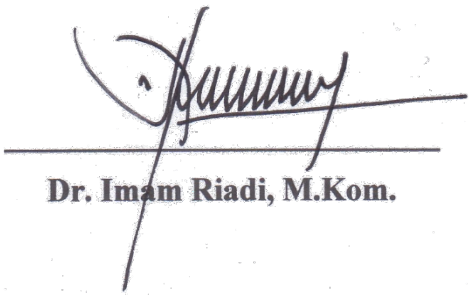
Disusun oleh:

Nama : Ninki Hermaduanti

NIM : 14917129

Yogyakarta, 2016

Pembimbing,



Dr. Imam Riadi, M.Kom.

LEMBAR PENGESAHAN PENGUJI

PENGEMBANGAN *FRAMEWORK* OTOMATISASI
MITIGASI KASUS *ROGUE ACCESS POINT*
PADA JARINGAN *WIRELESS* IEEE 802.1X

TESIS

Disusun oleh:

Nama : Ninki Hermaduanti
NIM : 14917129

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Pascasarjana Magister Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 18 Desember 2016

Tim Penguji,

Dr. Imam Riadi, M.Kom.
Penguji I

Yudi Prayudi, S.Si., M.Kom.
Penguji II

Ahmad Luthfi, S.Kom., M.Kom.
Penguji III

Mengetahui,
Direktur Program Pascasarjana Fakultas Teknologi Industri
Universitas Islam Indonesia

Dr. R. Teduh Dirgahayu, S.T., M.Sc.

LEMBAR PERNYATAAN KEASLIAN TESIS

Saya yang bertandatangan di bawah ini,

Nama	: Ninki Hermaduanti
NIM	: 14917129
Program Studi	: Magister Informatika
Konsentrasi Bidang	: Forensika Digital

Menyatakan bahwa di dalam tesis saya yang berjudul "PENGEMBANGAN *FRAMEWORK* OTOMATISASI MITIGASI KASUS *ROGUE ACCESS POINT* PADA JARINGAN *WIRELESS* IEEE 802.1X" tidak terdapat karya yang pernah diajukan orang lain untuk memperoleh gelar tertentu di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan di dalam daftar pustaka. Apabila di kemudian hari terbukti bahwa pernyataan ini tidak benar, maka saya bersedia menerima sanksi atau hukuman apapun sesuai dengan peraturan yang berlaku.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 18 Desember 2016



Ninki Hermaduanti

HALAMAN PERSEMBAHAN

Rasa syukur kehadirat Allah SWT atas limpahan berkah dan kasih sayang-Nya

Untuk Rasulullah SAW... dan untuk Islam agamaku...

Untuk Ayah & Ibu tercinta,

Terima kasih atas segala yang diberikan, jerih payah dan dukungan, do'a dan kasih sayangmu. You're trully my superheroes...

Untuk Kakak tercinta,

Terima kasih atas dukungan, nasihat, do'a dan pengertiannya.

Spesial untuk Suami tercinta,

Terima kasih untuk semua dukungan, kritik, saran, pengertian, dan do'anya…

HALAMAN MOTTO

"Jadikanlah sabar dan sholat sebagai penolongmu, sesungguhnya Allah beserta orang-orang yang sabar."

(Q.S. Al Baqarah ayat 153)

"Sesunguhnya sholatku, ibadahku, hidupku, dan matiku hanyalah untuk Allah, Tuhan semesta alam."

(Q.S. Al-An'am ayat 162)

"Wahai jiwa yang tenang. Kembalilah kepada Tuhan-mu dengan hati yang puas lagi diridhoi-Nya. Masuklah ke dalam jemaah hamba-hamba-Ku, dan masuklah ke dalam surga-Ku."

(Q.S. Al-Fajr ayat 27-30)

"Don't walk behind me, I may not lead. Don't walk in front of me, I may not follow. Just walk beside me, and be my friend."

(Anonymous)

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Alhamdulillah, segala puji bagi Allah SWT atas segala rahmat, hidayah, dan inayah-Nya, sehingga penulisan laporan tesis sebagai salah satu syarat memperoleh gelar Pascasarjana Magister Informatika Fakultas Teknologi Industri Universitas Islam Indonesia yang berjudul "PENGEMBANGAN *FRAMEWORK* OTOMATISASI MITIGASI KASUS *ROGUE ACCESS POINT* PADA JARINGAN *WIRELESS* IEEE 802.1X" dapat penulis selesaikan dengan baik. Shalawat serta salam semoga senantiasa tercurah atas Nabi Muhammad SAW, para sahabat, serta pengikutnya hingga hari kiamat nanti.

Penyusunan laporan tesis ini tidak lepas dari bimbingan, dukungan, dan bantuan dari berbagai pihak. Oleh karena itu dalam kesempatan ini dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang setulus-tulusnya kepada:

1. Allah SWT, yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis selalu diberi kesehatan dan kemudahan selama masa pengerjaan tesis ini.
2. Ayah, ibu, kakak, suami, beserta keluarga besar yang telah memberikan do'a restu dan dukungannya.
3. Bapak Rektor dan seluruh jajaran Rektorat Universitas Islam Indonesia.
4. Dr. R. Teduh Dirgahayu, S.T., M.Sc. selaku Direktur Program Pascasarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Dr. Imam Riadi, M.Kom. selaku dosen pembimbing yang telah memberikan pengarahan, bimbingan, masukan, serta dorongan semangat selama pelaksanaan tesis dan penulisan laporan.

6. Dosen-dosen Magister Informatika dan seluruh jajaran staf program Pascasarjana. Terima kasih atas semua ilmu pengetahuan, saran, motivasi, serta bantuannya.
7. Rekan-rekan Angkatan X. Terima kasih atas semua dukungan dan kerja samanya selama ini. Selamat berjuang.
8. Keluarga besar Magister Informatika. Terima kasih atas kerja samanya.
9. Teman-teman yang jauh di sana dan selalu mendoakan, terima kasih atas semuanya.
10. Semua pihak yang telah memberikan bantuan dan dorongan yang tidak dapat penulis sebutkan satu persatu.

Semoga segala kebaikan yang telah diberikan kepada penulis, akan dibalas Allah dengan yang lebih baik. Amin.

Penulis menyadari bahwa dalam penyusunan laporan tesis ini masih banyak terdapat kekurangan. Untuk itu penulis menyampaikan permohonan maaf serta sangat mengharapkan kritik dan saran yang membangun untuk penyempurnaan di masa mendatang.

Akhir kata semoga laporan ini dapat bermanfaat bagi kita semua. Amin.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 18 Desember 2016

Ninki Hermaduanti

ABSTRAK

Teknologi jaringan nirkabel (*wireless network*) makin marak digunakan di tempat-tempat umum. Selain menawarkan kemudahan koneksi, penggunaan *wireless network* ini juga menimbulkan isu keamanan karena berada di area publik yang sifatnya terbuka. *Port-based authentication system* atau dikenal sebagai standar IEEE 802.1X merupakan *framework* penyedia kontrol akses ke jaringan untuk proteksi dari penyusup. Namun tetap tidak menutup kemungkinan akan adanya serangan terhadap jaringan *wireless* berbasis IEEE 802.1X, salah satunya adalah *rogue Access Point* (*rogue* AP).

Rogue AP menyamar sebagai *Access Point* asli, menipu *client* untuk melakukan koneksi ke *Access Point* yang salah. Diperlukan langkah mitigasi kasus *rogue* AP termasuk di dalamnya adalah tahap deteksi dan eliminasi *rogue* AP untuk mengurangi dampak yang mungkin timbul akibat keberadaan *rogue* AP. Deteksi *rogue* AP dalam penelitian ini menggunakan metode *live forensics*, yaitu metode forensik yang dilakukan pada sistem yang sedang berjalan (*live*/*running*), dalam hal ini adalah jaringan *wireless* dan *rogue* AP yang sedang *running* sehingga keberadaan *rogue* AP tersebut dapat dideteksi.

Berdasarkan hasil pengujian yang dilakukan didapatkan hasil berupa kerangka kerja (*framework*) mitigasi kasus *rogue* AP pada jaringan *wireless* berbasis IEEE 802.1X yang dijadikan dasar otomatisasi langkah mitigasi. Temuan penelitian berupa: (1) Otomatisasi memberikan perbedaan terhadap cara manual pada proses pencatatan data *Access Point*, namun dengan adanya *script* otomatisasi, maka *script* cukup dieksekusi di awal dan tidak perlu mengetikkan berbagai macam perintah secara manual. (2) Tahap deteksi *rogue* AP berbasis *software* pada WLAN IEEE 802.1X dapat dilakukan menggunakan parameter SSID, MAC *address*, kekuatan sinyal, tipe enkripsi, dan jenis protokol EAP yang bekerja pada Layer 2 (*Data Link*). Penelitian ini telah dapat menyederhanakan parameter pengujian karena tidak sampai melibatkan parameter IP *address* pada Layer 3 (*Network*). Kontribusi mitigasi bahwa *rogue* AP dapat didisfungsikan sehingga tidak dapat melayani permintaan koneksi dari *client* saat dilakukan *deauthentication* pada tahap eliminasi karena *rogue* AP mengalami serangan *Denial of Service* (DoS).

Kata kunci: *framework*, *rogue*, *Access Point*, *forensics*, IEEE 802.1X

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING ii
LEMBAR PENGESAHAN PENGUJI iii
LEMBAR PERNYATAAN KEASLIAN TESIS iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTTO vi
KATA PENGANTAR vii
ABSTRAK ix
DAFTAR ISI x
DAFTAR GAMBAR xiii
DAFTAR TABEL xvi
DAFTAR ISTILAH xvii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 5
1.3 Batasan Masalah 5
1.4 Tujuan Penelitian 6
1.5 Manfaat Penelitian 6
1.6 *Review* Penelitian 7
1.7 Metodologi Penelitian 12
1.8 Sistematika Penulisan 14
BAB II LANDASAN TEORI 15
2.1 *Digital Forensics* 15
2.1.1 *Network Forensics* 16
2.1.2 *Wireless Forensics* 18
2.1.3 *Live Forensics* 19
2.2 IEEE 802.1X 20
2.2.1 Komponen IEEE 802.1X 23

2.2.2 Protokol IEEE 802.1X 24
2.3 *Rogue Access Point* 28
2.3.1 *Hardware Based Rogue* AP 30
2.3.2 *Software Based Rogue* AP 30
BAB III METODOLOGI PENELITIAN 35
3.1 Studi Literatur 35
3.2 Persiapan Sistem 35
3.3 Simulasi Kasus 36
3.4 Mitigasi Kasus dan Pengembangan *Framework* 38
3.4.1 Populasi Data *Authorized Access Point* 38
3.4.2 Deteksi *Rogue Access Point* 39
3.4.3 Eliminasi *Rogue Access Point* 43
3.5 Penyusunan Laporan 47
BAB IV HASIL DAN PEMBAHASAN 48
4.1 Studi Literatur Terhadap Protokol EAP 48
4.1.1 EAP (*Extensible Authentication Protocol*) 48
4.1.2 EAP-PEAP (*Protected Extensible Authentication Protocol*) 51
4.2 Persiapan Sistem WLAN Berbasis IEEE 802.1X 51
4.2.1 Spesifikasi Sistem 52
4.2.2 Topologi Sistem 52
4.2.3 Langkah Persiapan Sistem 53
4.2.4 Langkah Uji Coba Sistem 54
4.3 Simulasi Kasus *Rogue Access Point* 56
4.3.1 Spesifikasi *Rogue Access Point* 57
4.3.2 Skenario Simulasi 57
4.3.3 Langkah-langkah Simulasi 58
4.4 Mitigasi Kasus *Rogue Access Point* dan Pengembangan *Framework* 64
4.4.1 Spesifikasi dan Persiapan Sistem Mitigasi 64
4.4.2 Tahap Populasi Data *Authorized Access Point* 69
4.4.3 Tahap Deteksi *Rogue Access Point* 86
4.4.4 Tahap Eliminasi *Rogue Access Point* 103

4.4.5 *Framework* Mitigasi Kasus *Rogue Access Point* 112
4.4.6 Validasi Sistem 116
4.4.7 Penjadwalan Otomatisasi Deteksi dan Eliminasi *Rogue Access Point* 119
4.4.8 Analisis Temuan Penelitian 123
BAB V KESIMPULAN DAN SARAN 127
5.1 Kesimpulan.....127
5.2 Saran.....128
DAFTAR PUSTAKA 129
LAMPIRAN.....134
Lampiran 1: *Script* Otomatisasi Persiapan *Hardware*.....134
Lampiran 2: *Script* Otomatisasi Tahap Populasi Data *Authorized Access Point* 135
Lampiran 3: *Script* Otomatisasi Fase *Generate* Data *Access Point* pada Tahap Populasi Data *Authorized Access Point* 136
Lampiran 4: *Script* Otomatisasi Fase *Generate* Data *Access Point* pada Tahap Deteksi *Rogue Access Point* 137
Lampiran 5: *Script* Otomatisasi Komparasi Data pada Tahap Deteksi *Rogue Access Point* 139
Lampiran 6: *Script* Otomatisasi Tahap Eliminasi *Rogue Access Point* 142

DAFTAR GAMBAR

Gambar 1.1 Statistik tempat akses internet di Indonesia 2
Gambar 1.2 Arsitektur WLAN berbasis IEEE 802.1X.....12
Gambar 2.1 Pembagian subcabang *network forensics*.....17
Gambar 2.2 Proses autentikasi secara sederhana.....21
Gambar 2.3 Port *Ethernet wired network* menyediakan koneksi *link* fisik 22
Gambar 2.4 Asosiasi *wireless client* ke AP membuat sebuah *virtual link*.....22
Gambar 2.5 Hubungan antara komponen IEEE 802.1X dan EAP 27
Gambar 2.6 Serangan yang mungkin terjadi akibat *rogue* AP 29
Gambar 2.7 Konfigurasi *soft* AP pada *Windows Network Bridge*.....31
Gambar 2.8 Konfigurasi *soft* AP pada laptop dengan fitur ICS 32
Gambar 2.9 Konfigurasi *soft* AP pada perangkat tambahan.....33
Gambar 2.10 Konfigurasi *soft* AP dengan *Virtual Wifi Interface*.....34
Gambar 2.11 Konfigurasi *soft* AP pada perangkat genggam.....34
Gambar 3.1 Alur metode penelitian.....35
Gambar 3.2 Tahapan implementasi WLAN berbasis IEEE 802.1X.....36
Gambar 3.3 Skenario *rogue* AP pada WLAN berbasis IEEE 802.1X.....37
Gambar 3.4 Tahapan simulasi kasus setelah *attacker* mendapatkan *username* dan *password client* 37
Gambar 3.5 Tahapan populasi data *authorized* AP dan deteksi *rogue* AP.....42
Gambar 3.6 *Flowchart* tahapan mitigasi kasus *rogue* AP secara umum 44
Gambar 3.7 *Flowchart* otomatisasi tahapan mitigasi kasus *rogue* AP 45
Gambar 3.8 Tahapan yang dilalui untuk penyusunan laporan.....47
Gambar 4.1 Paket EAP di dalam jaringan berbasis IEEE 802.1X.....49
Gambar 4.2 Topologi WLAN berbasis IEEE 802.1X 53
Gambar 4.3 Form login SSID Pascasarjana.....54
Gambar 4.4 *Radius Server Certificate* 55
Gambar 4.5 Detail dari *Radius Server Certificate* 55
Gambar 4.6 Detail IP *address* yang didapatkan oleh *client*.....56
Gambar 4.7 Topologi dan skenario simulasi *rogue* AP.....58
Gambar 4.8 File konfigurasi *soft* AP 58
Gambar 4.9 Servis RADIUS sedang berjalan.....59
Gambar 4.10 *Soft* AP telah aktif 59
Gambar 4.11 *Client* mengisikan *username* dan *password*.....60
Gambar 4.12 *Username* dan *challenge*/*response* dari *password client* 60
Gambar 4.13 *Username* dan *challenge*/*response* dari *password client* 60
Gambar 4.14 *Username* dan *password client* didaftarkan di RADIUS 61

Gambar 4.15 Proses autentikasi oleh server RADIUS 61
Gambar 4.16 IP *address client* dari hasil koneksi ke *soft* AP.....62
Gambar 4.17 *Client* melakukan proses "traceroute" ke situs web.....62
Gambar 4.18 *Flowchart* simulasi *rogue* AP pada WLAN berbasis 802.1X.....63
Gambar 4.19 *Wireless interface* wlan0 dan wlan1 65
Gambar 4.20 Set *interface* wlan1 ke mode monitor 66
Gambar 4.21 *Wireless interface* wlan0 dan wlan1mon 66
Gambar 4.22 Set *interface* wlan1 ke mode monitor oleh sistem 67
Gambar 4.23 *Flowchart* persiapan *hardware* secara manual 68
Gambar 4.24 *Flowchart* otomatisasi persiapan *hardware*.....68
Gambar 4.25 Potongan *script* otomatisasi persiapan *hardware* 69
Gambar 4.26 *Flowchart* tahap populasi data *authorized* AP secara manual 72
Gambar 4.27 *Flowchart* otomatisasi tahap populasi data *authorized* AP.....73
Gambar 4.28 Potongan *script* otomatisasi tahap populasi data *authorized* AP 74
Gambar 4.29 Eksekusi perintah Iwlist untuk *scan Access Point*.....75
Gambar 4.30 Ekstrak data hasil *scanning Access Point* 76
Gambar 4.31 Input parameter hasil ekstraksi ke tabel *authorized* AP.....76
Gambar 4.32 *Capture* trafik *wireless* untuk mendapatkan paket EAP 77
Gambar 4.33 Analisis file hasil *capture* trafik *wireless*.....77
Gambar 4.34 Input parameter jenis protokol EAP.....78
Gambar 4.35 File konfigurasi Wpa_supplicant 80
Gambar 4.36 *Capture* trafik *wireless* dan *trigger* paket EAP otomatis.....80
Gambar 4.37 Hasil Tshark mengetahui ada tidaknya protokol EAP 81
Gambar 4.38 Pencatatan isi variabel ke dalam file tabel *Access Point* 81
Gambar 4.39 *Flowchart* fase *generate* data *Access Point* secara manual.....82
Gambar 4.40 *Flowchart* otomatisasi fase *generate* data *Access Point* 83
Gambar 4.41 Potongan *script* otomatisasi tahap populasi data *authorized* AP 85
Gambar 4.42 Menjalankan *script* otomatisasi deteksi *rogue* AP.....87
Gambar 4.43 *Scanning Access Point* oleh sistem 87
Gambar 4.44 *Script* menganalisis file hasil *capture* trafik *wireless* 88
Gambar 4.45 Pencatatan isi variabel ke dalam tabel *unauthorized* AP 89
Gambar 4.46 Potongan *script* otomatisasi fase *generate* data *Access Point* pada tahap deteksi *rogue* AP 89
Gambar 4.47 *Framework* fase *generate* data *Access Point* pada tahap populasi data *authorized* AP.....92
Gambar 4.48 *Framework* fase *generate* data *Access Point* pada tahap deteksi *rogue* AP 93
Gambar 4.49 Data *unauthorized* AP dan data *authorized* AP 94
Gambar 4.50 Komparasi SSID pada tahap deteksi *rogue* AP 96
Gambar 4.51 Komparasi SSID dan MAC *address* pada tahap deteksi *rogue* AP .97

Gambar 4.52 Pengecekan kekuatan sinyal pada tahap deteksi *rogue* AP.....97
Gambar 4.53 Komparasi tipe enkripsi pada tahap deteksi *rogue* AP 98
Gambar 4.54 Komparasi jenis protokol EAP pada tahap deteksi *rogue* AP 98
Gambar 4.55 Hasil akhir komparasi data *Access Point* 99
Gambar 4.56 Potongan *script* otomatisasi komparasi data pada tahap deteksi *rogue* AP 100
Gambar 4.57 *Flowchart* komparasi data *unauthorized* AP dan data *authorized* AP 101
Gambar 4.58 Hasil deteksi *rogue* AP dengan tiga kali iterasi *script* 102
Gambar 4.59 *Framework* tahap deteksi *rogue* AP dengan *live forensics* 103
Gambar 4.60 Pengecekan *client rogue* AP 104
Gambar 4.61 *Containment rogue* AP.....105
Gambar 4.62 Tidak ada *client* yang terkoneksi ke *rogue* AP setelah *containment* dijalankan.....105
Gambar 4.63 Pengecekan *client rogue* AP oleh sistem 106
Gambar 4.64 Data *client rogue* AP yang ditampilkan oleh sistem.....107
Gambar 4.65 *Containment rogue* AP oleh sistem 107
Gambar 4.66 *Containment rogue* AP selesai dijalankan 108
Gambar 4.67 Potongan *script* otomatisasi tahap eliminasi *rogue* AP 109
Gambar 4.68 *Flowchart* tahap eliminasi *rogue* AP secara manual 110
Gambar 4.69 *Flowchart* otomatisasi tahap eliminasi *rogue* AP.....111
Gambar 4.70 *Framework* tahap eliminasi *rogue* AP 112
Gambar 4.71 *Framework* mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X secara manual.....114
Gambar 4.72 *Framework* otomatisasi mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X.....115
Gambar 4.73 File konfigurasi Wpa_supplicant untuk koneksi ke *rogue* AP 117
Gambar 4.74 Koneksi ke *rogue* AP dengan *tool* Wpa_supplicant 118
Gambar 4.75 IP *address* yang didapatkan oleh *client rogue* AP 118
Gambar 4.76 *Traceroute* yang dilakukan oleh *client rogue* AP 119
Gambar 4.77 Penjadwalan otomatisasi tahap deteksi dan eliminasi *rogue* AP ...120
Gambar 4.78 Penjadwalan otomatisasi berdasarkan menit.....120
Gambar 4.79 Penjadwalan otomatisasi berdasarkan jam.....121
Gambar 4.80 Penjadwalan otomatisasi harian 121
Gambar 4.81 Penjadwalan otomatisasi mingguan 122
Gambar 4.82 Penonaktifan penjadwalan otomatisasi 122

DAFTAR TABEL

Tabel 1.1 Perbandingan protokol keamanan pada WLAN 3
Tabel 1.2 *Literature review* terhadap penelitian sebelumnya.....10
Tabel 3.1 Presentasi data *authorized Access Point*.....38
Tabel 3.2 Presentasi data *unauthorized Access Point*.....41
Tabel 4.1 Segmentasi WLAN berbasis IEEE 802.1X 53
Tabel 4.2 Konfigurasi IP *address* pada simulasi kasus *rogue* AP 57
Tabel 4.3 Penugasan parameter hasil ekstraksi ke dalam variabel 79
Tabel 4.4 Variabel yang dicatatkan ke dalam file tabel *authorized* AP.....81
Tabel 4.5 Perbedaan fase *generate* data AP pada tahap populasi data *authorized* AP secara manual dan terotomatisasi 84
Tabel 4.6 Perbedaan fase *generate* data AP pada tahap populasi data *authorized* AP dan tahap deteksi *rogue* AP 91
Tabel 4.7 Rangkuman temuan penelitian.....124
Tabel 4.8 Perbandingan jenis *rogue* AP dan parameter penelitian.....126

DAFTAR ISTILAH

back end	layanan atau servis yang tidak langsung berhadapan dengan pengguna servis tersebut
chain of custody	dokumentasi kronologis untuk rantai posesi barang bukti
client	*user* atau pengguna
clock skew	perbedaan waktu datangnya *clock signal* ke masing-masing *chip* di dalam sirkuit digital
containment	pencegahan penyebaran konsekuensi negatif dari suatu kejadian
crack	meretas, memecahkan
deauthentication	serangan terhadap komunikasi *wireless* di antara *client* dan *Access Point*
frame	unit transmisi data digital pada jaringan komputer yang bekerja pada layer *data link*
framework	struktur yang didesain untuk mendukung sesuatu
gateway	perangkat yang berfungsi sebagai pintu gerbang ke jaringan komputer
hardware	perangkat keras
hotspot	tempat yang menyediakan akses internet tanpa kabel
interface	antarmuka
memory dump	proses menyalin *memory* ke media output
misconfigured	konfigurasi yang tidak tepat
parsing	memilih data berdasarkan format tertentu
passphrase	kode untuk mengontrol akses ke sistem, program, atau data
rogue	sesuatu yang tidak normal atau bertentangan dengan aturan
RSSI	*Recieved Signal Strength Indicator*, kekuatan sinyal
search warrant	surat izin penggeledahan
sniff	memonitor data yang lewat di dalam jaringan komputer
soft AP	*Access Point* berbasis *software*
SSID	*Service Set Identifier*, nama jaringan *wireless*
time stamp	informasi mengenai kapan sebuah peristiwa terjadi
update	memperbarui atau mengubah data
volatile	rentan
war driving	deteksi jaringan *wireless* dari dalam kendaraan yang bergerak
white paper	laporan atau panduan pengguna mengenai sebuah isu
wired	menggunakan kabel
wireless	tanpa kabel (nirkabel)

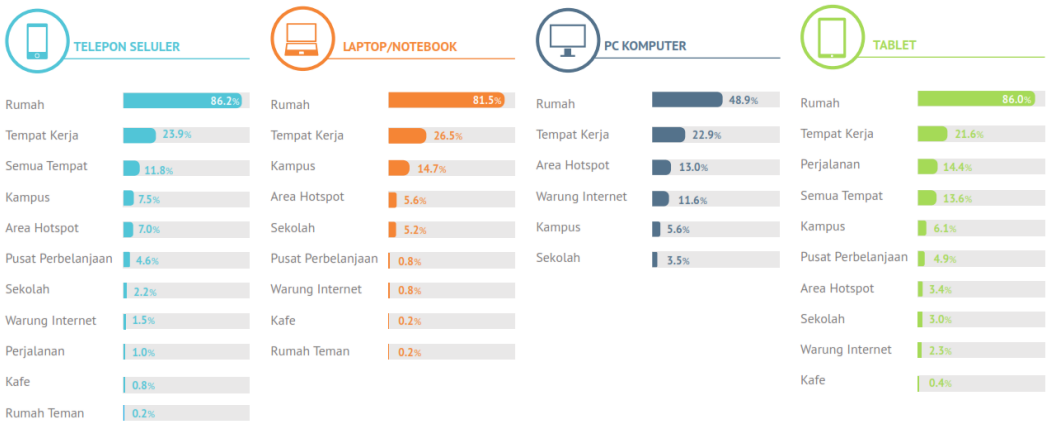
BAB I

PENDAHULUAN

1.1 Latar Belakang

Menurut laporan survey yang disusun oleh Pusat Kajian Komunikasi Universitas Indonesia (Puskakom UI) dan Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) pada 2015 tentang "Profil Pengguna Internet Indonesia 2014", internet di Indonesia mulai berkembang pada awal 1980-an di dalam universitas-universitas negeri di Indonesia (Triastuti, 2013). Kemudian menjadi populer pasca runtuhnya rezim Soeharto pada tahun 1998 (Hill & Sehn, 2001). Internet dijadikan alat untuk melakukan pergerakan politik oleh para mahasiswa. Sejak saat itu, masyarakat di Indonesia mulai mengenal adanya aktivitas berbasis internet hingga berkembang seperti sekarang ini. Perangkat seperti telepon seluler, laptop/*notebook*, *Personal Computer* (PC), dan tablet menjadi perangkat yang paling sering digunakan oleh pengguna internet di Indonesia untuk mengakses internet. Sedangkan rumah, tempat kerja (kantor), dan tempat-tempat umum yang menyediakan akses internet nirkabel (*wireless*), seperti kampus, area *hotspot*, pusat perbelanjaan, dan kafe dijadikan para pengguna internet di Indonesia sebagai tempat pilihan untuk mengakses internet (APJII, 2015). Gambar 1.1 menunjukkan statistik tempat pilihan pengguna internet di Indonesia beserta perangkat yang digunakan untuk mengakses internet.

Internet *wireless* sebagaimana yang disediakan di tempat-tempat umum seperti kampus, area *hotspot*, pusat perbelanjaan, dan kafe ini merupakan implementasi dari teknologi jaringan berbasis nirkabel (*wireless network*) yang saat ini semakin marak digunakan. Selain menawarkan kemudahan dalam koneksinya, penggunaan teknologi *wireless network* ini juga ternyata menimbulkan isu keamanan yang patut dipertimbangkan karena berada di area publik yang sifatnya terbuka (Faqih, Rochim, & Isnanto, 2012).



Gambar 1.1 Statistik tempat akses internet di Indonesia
Sumber: APJII, 2015

Penjagaan keamanan perlu dipastikan telah diaplikasikan secara tepat dalam mengimplementasikan jaringan komputer, baik yang berbasis kabel (*wired network*) maupun nirkabel (*wireless network*). Perlu adanya mekanisme yang dapat mengontrol akses ke jaringan untuk memproteksi jaringan komputer dari penyusup. *Framework* yang menyediakan kontrol akses ke jaringan disebut sebagai *port-based authentication system* atau dikenal sebagai standar IEEE 802.1X yang dirilis oleh *Institute of Electrical and Electronics Engineers* (IEEE), yaitu organisasi internasional yang bertujuan untuk mengembangkan inovasi dan teknologi untuk meningkatkan harkat kemanusiaan. Konsep implementasi IEEE 802.1X ini melakukan verifikasi terhadap informasi milik pengguna (*client*) yang berupa *username* dan *password*. Jika informasi tersebut valid, maka *client* diperbolehkan mengakses jaringan (Geier, 2008). Dalam *white paper* "*Introduction to 802.1X for Wireless LANs*" (Networks, 2006) disebutkan bahwa pada awalnya IEEE 802.1X hanya didesain untuk memenuhi kebutuhan keamanan pada *wired network* dengan cakupan *Local Area Network* (IEEE 802 LAN) dengan cara menyediakan mekanisme autentikasi perangkat yang terhubung ke port LAN. Namun seiring berkembangnya teknologi jaringan dan dengan munculnya keberadaan teknologi *wireless*, standar IEEE 802.1X juga dikembangkan untuk memenuhi kebutuhan keamanan terhadap *wireless network* dengan cakupan *Wireless Local Area Network* (IEEE 802.11 LAN) atau disebut sebagai WLAN.

Enkripsi yang digunakan pada WLAN non-802.1X adalah WEP (*Wired Equivalent Privacy*), WPA (*Wi-Fi Protected Access*), atau WPA2-Personal. Oleh karena *client* diautentikasi ke jaringan menggunakan *passphrase*, maka penyusup dapat mengakses WLAN dengan cara memecahkan (*cracking*) *passphrase*-nya (Hoffman, 2013). Sedangkan pada WLAN berbasis IEEE 802.1X, enkripsi yang digunakan adalah WPA-Enterprise (Sukhija & Gupta, 2012) atau WPA2-Enterprise yang menggunakan server RADIUS (*Remote Authentication Dial-In User Service*) sebagai server autentikasi. *Client* diautentikasi ke jaringan menggunakan *username* dan *password*-nya masing-masing (Hoffman, 2013). Tabel 1.1 menunjukkan perbandingan enkripsi pada WLAN.

Tabel 1.1 Perbandingan protokol keamanan pada WLAN

	WEP	**WPA**	**WPA2**
Tujuan	Menyediakan keamanan seperti pada jaringan berbasis kabel	Mengatasi kelemahan WEP tanpa memerlukan *hardware* baru	Sebagai peningkatan dari enkripsi WPA
Autentikasi	WEP-Open dan WEP-Shared	WPA-PSK dan WPA-Enterprise	WPA2-Personal dan WPA2-Enterprise
Kompatibilitas *Hardware*	Bekerja pada *hardware* yang sudah ada	Bekerja pada *hardware* yang sudah ada melalui proses *upgrade firmware* pada *Network Interface Card* (NIC)	Didukung pada perangkat dengan sertifikasi Wi-Fi sejak 2006 dan tidak bekerja menggunakan NIC yang lama
Kompleksitas Implementasi	Mudah untuk di-*setup* dan dikonfigurasi	Proses *setup* yang kompleks untuk WPA-Enterprise	Proses *setup* yang kompleks untuk WPA2-Enterprise

Sumber: Sukhija & Gupta, 2012

Penerapan standar IEEE 802.1X ditujukan untuk menyediakan mekanisme autentikasi perangkat yang dipakai oleh *client* ke WLAN. Namun, meskipun standar IEEE 802.1X ini telah diterapkan, ternyata masih dimungkinkan adanya celah keamanan untuk melakukan serangan. Ada bermacam-macam serangan yang dapat dilakukan terhadap WLAN, termasuk yang menerapkan standar IEEE

802.1X. Di antaranya adalah *war driving* (Jacob, Hutchinson, & Abawajy, 2011), yang merupakan tahap pengenalan dalam aktivitas serangan terhadap WLAN, yaitu dengan melakukan pemetaan secara geografis terhadap *wireless hotspot* dan tipe keamanan atau jenis enkripsinya dari dalam kendaraan yang berjalan. *Denial of Service* (DoS) *attack*, yaitu menghabiskan sumber daya target serangan, seperti *memory* dan *bandwidth* atau dengan cara mengeksploitasi celah pada protokol yang digunakan (Ozan, 2004). Serangan lainnya berupa *rogue Access Point* (*rogue* AP) sebagai *rogue authenticator* yang menyamar sebagai perangkat autentikator asli. Hal ini berbahaya, mengingat *rogue* AP dapat digunakan untuk mendapatkan data *client* dan memperoleh akses ke jaringan (Geier, 2008).

Banyaknya serangan yang mungkin terjadi pada jaringan *wireless* yang salah satunya berupa *rogue* AP, menjadikan perlu dilakukannya langkah mitigasi kasus keberadaan *rogue* AP tersebut. Mitigasi kasus *rogue* AP berarti melakukan kegiatan untuk mengurangi dampak atau resiko yang mungkin timbul akibat keberadaan *rogue* AP. Sifat dari *rogue* AP yang menyamar sebagai perangkat autentikator (Geier, 2008), menjadikannya sulit dibedakan dari perangkat yang asli. Sehingga langkah pertama dalam mitigasi kasus yang perlu dilakukan adalah mendeteksi keberadaan *rogue* AP tersebut (Saruhan, 2007). Deteksi *rogue* AP sebagai bagian dari mitigasi kasus dapat dilakukan dengan pendekatan *live forensics analysis*, yaitu metode forensik dengan mengumpulkan informasi, menganalisis, dan mempresentasikannya menggunakan berbagai macam *tools* forensik pada saat sistem masih berjalan (Rafique & Khan, 2013). Berbeda dengan *static forensics analysis* yang menggunakan pendekatan tradisional dengan cara mengambil *memory dump* dan mematikan sistem yang akan diforensik, *live forensics* dapat dilakukan apabila terjadi intrusi pada sistem atau jaringan, bertujuan untuk mengamankan barang bukti yang rentan (*volatile evidence*) pada sistem yang sedang berjalan selama proses investigasi berlangsung (Luttgens, Pepe, & Mandia, 2014).

Banyak penelitian yang dilakukan mengenai kasus *rogue* AP, namun tidak ada yang menegaskan bahwa penelitian dilakukan di lingkungan WLAN berbasis IEEE 802.1X. Selain itu, penelitian-penelitian tersebut kebanyakan hanya

membahas metode-metode untuk deteksi *rogue* AP. Oleh karena itu, topik yang diangkat dalam penelitian ini adalah mengenai mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X, meliputi proses deteksi dan eliminasi *rogue* AP. Diharapkan output yang dihasilkan dalam penelitian ini adalah berupa pengembangan kerangka kerja (*framework*) mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X yang sekaligus digunakan sebagai dasar untuk melakukan otomatisasi langkah mitigasi kasus *rogue* AP tersebut. Otomatisasi dilakukan dengan tujuan untuk membantu administrator jaringan meminimalisir tugas-tugas manual yang dikerjakan dalam langkah mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X.

1.2 Rumusan Masalah

Merujuk kepada latar belakang yang telah dipaparkan sebelumnya, maka dapat diambil rumusan masalah di dalam penelitian ini, yaitu sebagai berikut:

1. Bagaimana tahapan mitigasi kasus *rogue Access Point* (*rogue* AP) pada WLAN berbasis IEEE 802.1X.
2. Bagaimana merancang *framework* otomatisasi mitigasi kasus rogue AP pada WLAN berbasis IEEE 802.1X.
3. Bagaimana melakukan otomatisasi langkah mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X.
4. Bagaimana dampak mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X.

1.3 Batasan Masalah

Beberapa batasan masalah yang ditetapkan di dalam penelitian ini adalah sebagai berikut:

1. Ruang lingkup penelitian dilakukan di lingkungan *Wireless Local Area Network* (WLAN) 802.11b/g yang berbasis IEEE 802.1X.
2. Simulasi kasus *rogue Access Point* (*rogue* AP) pada WLAN berbasis IEEE 802.1X dilakukan menggunakan *rogue* AP berbasis *software* (*soft* AP) karena

jika menggunakan yang berbasis *hardware* maka diperlukan akses fisik ke *switch* di dalam jaringan komputer. Selain itu, penggunaan *soft* AP menjadikan simulasi lebih fleksibel dan dapat berpindah-pindah tempat.

3. Otomatisasi langkah mitigasi kasus *rogue* AP tidak sepenuhnya membuat sistem yang dibuat untuk mitigasi berjalan otomatis, tetapi hanya meminimalisir tugas-tugas manual yang dikerjakan oleh administrator jaringan.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan, maka dapat ditentukan tujuan penelitian. Adapun tujuan dari dilakukannya penelitian ini adalah sebagai berikut:

1. Merancang tahapan mitigasi kasus *rogue Access Point* (*rogue* AP) pada WLAN berbasis IEEE 802.1X.
2. Merancang *framework* otomatisasi mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X.
3. Mengimplementasikan otomatisasi langkah mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X.
4. Menganalisis dampak mitigasi kasus *rogue* AP terhadap *rogue* AP beserta *client*-nya pada WLAN berbasis IEEE 802.1X.

1.5 Manfaat Penelitian

Manfaat dilakukannya penelitian ini adalah sebagai berikut:

1. Memberikan manfaat terhadap pengembangan ilmu *digital forensics* pada umumnya dan *wireless forensics* pada khususnya.
2. Menambah pengetahuan mengenai mitigasi kasus *rogue Access Point* (*rogue* AP) pada WLAN, khususnya WLAN yang berbasis IEEE 802.1X, meliputi langkah deteksi dengan *live forensics* dan juga langkah eliminasi *rogue* AP.
3. Mengetahui proses untuk mendapatkan bukti digital pada kasus *rogue* AP di lingkungan WLAN berbasis IEEE 802.1X.

4. Otomatisasi langkah mitigasi kasus *rogue* AP diharapkan dapat meminimalisir tugas-tugas manual yang dikerjakan oleh administrator jaringan.

1.6 *Review* Penelitian

Seperti telah dipaparkan sebelumnya bahwa salah satu serangan yang terjadi pada WLAN adalah *rogue Access Point* (*rogue* AP), maka dalam *review* penelitian ini akan diambil domain penelitian berupa kasus *rogue* AP. *Rogue* AP dalam penelitian yang dilakukan oleh Shivaraj, Song, dan Shetty (2008), dideteksi menggunakan pendekatan statistik, yaitu dengan *Hidden Markov Model* (HMM). Terdiri dari dua tahap, yaitu tahap pertama, *data training* pada HMM, dan dilanjutkan tahap kedua, yaitu deteksi *rogue* AP berdasarkan hasil pada tahap pertama. Deteksi *rogue* AP dipicu oleh salah satu dari tiga jenis DoS *attack* yang dilancarkan oleh *host* yang terkoneksi ke *rogue* AP, yaitu DoS dengan paket *ping* berukuran besar, DoS untuk mendeteksi port yang terbuka pada *single host*, dan DoS jenis *syn flood*. Hasil penelitian menunjukkan bahwa tingkat akurasi deteksi *rogue* AP mencapai 85%.

Penelitian Jagtap dan Honwadkar (2010) mengusulkan metode yang *cost effective*, karena tidak memerlukan adanya sensor atau *hardware* tambahan untuk deteksi *rogue* AP. AP dibagi menjadi 3, yaitu *authorized*, *external*, dan *unauthorized*. Kemudian dipisahkan berdasarkan *Service Set Identifier* (SSID), apabila SSID-nya berbeda, maka dikategorikan ke dalam *external* AP, namun apabila sama, dapat dicurigai sebagai *unauthorized* AP. Kemudian dilanjutkan dengan pemeriksaan kekuatan sinyal dan IP *address* AP, dan terakhir dibandingkan dengan data *authorized* AP yang ada pada *file system*.

Penelitian lain untuk deteksi *rogue* AP adalah dengan mengaplikasikan *mobile Multi-agents* pada jaringan, yaitu *Master Mobile Agents* yang dipasang pada server dan *Slave Mobile Agents* yang dipasang pada *client* (Sasane & Pathan, 2011). Tujuannya adalah untuk mengaplikasikan konsep otomatisasi ke dalam proses *scanning* untuk deteksi dan eliminasi *rogue* AP menggunakan interval *skew*. Penelitian ini tidak membutuhkan *hardware* khusus. Hasil dari penelitian adalah dapat melakukan deteksi dan eliminasi *rogue* AP dari jaringan.

Deteksi *rogue* AP juga dapat dilakukan menggunakan pendekatan *client-side* (Nikbakhsh, Manaf, Zamani, & Janbeglou, 2012). Penelitian ini mengusulkan metode praktis yang dapat memperingatkan *client* agar terhindar dari koneksi ke *rogue* AP, yaitu dengan membandingkan *gateway* dan *route* dari paket yang berjalan, untuk menentukan apakah sebuah AP itu adalah AP yang *legitimate* atau tidak. Deteksi *rogue* AP ini dilakukan menggunakan perangkat yang terkoneksi ke AP. Hasil penelitiannya adalah metode deteksi dari *client-side* dapat mendeteksi *rogue* AP dengan jenis *Man in the Middle* (MITM) dan *evil twin attack*.

Thakur dan Bodhe (2013) mengusulkan algoritma RAPD untuk deteksi *rogue* AP dengan cara melakukan *sniffing* terlebih dahulu kemudian membandingkan data hasil *sniffing* dengan data *auhorized* AP berdasarkan SSID dan MAC *address*-nya. Terdapat tiga kategori yang dihasilkan, yaitu *completely matched* untuk AP yang SSID dan MAC *address*-nya sama, *completely unmatched* untuk SSID dan MAC *address* yang sama sekali berbeda, dan *partially matched* untuk salah satu dari SSID atau MAC *address*-nya yang sama dengan data *authorized* AP. Untuk kategori *completely unmatched* dapat dicurigai bahwa AP tersebut merupakan *neighborhood* AP atau *rogue* AP milik *employee*, sehingga diperlukan analisis karakteristik trafik lebih lanjut. Sedangkan untuk kategori *partially matched*, dapat dicurigai bahwa AP tersebut merupakan *misconfigured* AP atau *rogue* AP milik *attacker*, oleh karena itu diperlukan pemeriksaan lebih lanjut pada kekuatan sinyal dan *channel* yang digunakan.

Penelitian Wang (2013) adalah mengenai *rogue* AP yang digunakan untuk serangan *Man in The Middle* (MITM). Teknik deteksi *rogue* AP yang digunakan adalah dengan pendekatan berbasis RSSI (*Received Signal Strength Indicator*). Ketika *receiver* (*client*) menerima sebuah *frame* dari jaringan *wireless*, maka *frame* tersebut biasanya akan direkam RSSI-nya berikut juga SSID yang mentransmisikannya. *Rogue* AP biasanya menggunakan SSID yang sama dengan *legitimate* AP untuk menipu *client*. Jika *frame* dengan SSID yang sama memiliki nilai RSSI yang berbeda, maka dapat disimpulkan terdapat *rogue* AP yang muncul.

Teknik *fingerprinting* menggunakan *clock skew* (interval *time stamp*) juga dapat dilakukan untuk deteksi *fake* AP (Jadhav, Vanjale, & Mane, 2014). Penelitian

ini memperhitungkan *clock skew* milik *authorized* AP dan menjadikannya sebagai nilai *threshold*. Setelah itu, perhitungan *clock skew* dilakukan pada *tested* AP, kemudian membandingkan hasilnya dengan hasil perhitungan *clock skew* pada *authorized* AP. Jika nilai dari hasil perhitungan *clock skew* pada *tested* AP lebih besar daripada nilai *threshold*, maka AP tersebut merupakan *fake* AP.

Penelitian Vanjale dan Mane (2014) mengusulkan deteksi dan eliminasi *rogue* AP dalam tiga tahap, yaitu tahap *learning mode*, tahap di mana daftar *authorized* AP dibuat, dilanjutkan tahap *detection mode* di mana sistem memeriksa SSID yang sama dari dua AP atau lebih, MAC *address* dan nilai *Received Signal Strength* (RSS)-nya untuk mendapatkan *rogue* AP, kemudian tahap ketiga yaitu tahap *prevention mode*, dengan mengimplementasikan aturan *blocking* pada AP yang terdeteksi sebagai *rogue* AP.

Penelitian Chougule, Vanjale, dan Mane (2015) mengusulkan teknik deteksi *rogue* AP menggunakan berbagai macam parameter, seperti SSID, MAC *address*, *authentication type*, *channel*, *frequency*, dan *signal*. Jika AP yang dites (*tested* AP) memiliki SSID yang sama, maka pengecekan berlanjut kepada parameter berikutnya yaitu MAC *address*. Jika MAC *address* sama, maka pengecekan berlanjut ke parameter *authentication type*. Begitu seterusnya, sampai ditemukan kesamaaan dengan parameter-parameter yang lain. Jika banyak parameter yang sama, maka dapat disimpulkan bahwa *tested* AP tersebut adalah *rogue* AP.

Nivangune, Vanjale, dan Mane (2015) dalam penelitiannya menggunakan teknik deteksi *rogue* AP dengan pendekatan dari sisi *client* (*client-side*). Pendekatan ini menggunakan bantuan dari perhitungan CTT (*Complete Tour Time*). Konsepnya adalah *client* perlu terasosiasi dengan jaringan melalui server lokal dan mencatat waktu total yang digunakan untuk setiap *request*-nya ke server. Jika nilai rata-rata waktunya lebih besar dari nilai *threshold*, maka dapat disimpulkan bahwa AP di mana *client* tersebut terasosiasi merupakan *rogue* AP.

Rangkuman *literature review* terhadap penelitian-penelitian yang telah disebutkan sebelumnya, dapat dilihat pada Tabel 1.2.

Tabel 1.2 *Literature review* terhadap penelitian sebelumnya

No.	Peneliti / Tahun	Domain Penelitian	Pendekatan/ Metode Penelitian	Hasil Penelitian
1	Gayathri Shivaraj, Min Song, Sachin Shetty (2008)	Deteksi *rogue* AP dengan pendekatan statistik.	Metode deteksi *rogue* AP dengan *Hidden Markov model* (HMM) yang dipicu oleh DoS *attack* yang diinisiasi oleh *client* dari *rogue* AP.	Tingkat akurasi deteksi *rogue* AP mencapai 85%.
2	Ms Shital V Jagtap, Prof. K. N. Honwadkar (2010)	Deteksi *rogue* AP dengan analisis trafik dan *behavior* jaringan.	Metode deteksi dan komparasi SSID, kekuatan sinyal, dan IP *address* AP.	Metode yang dihasilkan *cost effective* tanpa tambahan sensor atau *hardware*.
3	Priyanka G. Sasane, S. K. Pathan (2011)	Deteksi dan eliminasi *rogue* AP dengan konsep otomatisasi.	Metode *scanning* untuk deteksi dan eliminasi *rogue* AP menggunakan *mobile Multi-Agents* (*Master* dan *Slave*).	Metode ini tidak membutuhkan *hardware* khusus dan terbukti dapat melakukan deteksi dan eliminasi *rogue* AP.
4	Somayeh Nikbakhsh, Azizah Bt Abdul Manaf, Mazdak Zamani, Maziar Janbeglou (2012)	Deteksi *rogue* AP jenis MITM dan *evil twin attack*.	Metode *client-side* yang memanfaatkan perangkat *client* yang terkoneksi ke AP untuk membandingkan *gateway* dan *route* dari paket yang berjalan.	Metode *client-side* terbukti dapat mendeteksi *rogue* AP jenis MITM dan *evil twin attack*.
5	Dr. Sanjay Thakur, Abhijit Bodhe (2013)	Deteksi *rogue* AP.	Algoritma RAPD untuk deteksi *rogue* AP dengan cara melakukan *sniffing*	Menghasilkan tiga kategori, yaitu *completely matched*, *completely unmatched*, dan *partially matched*.

			dan membandingkan data hasil *sniffing* dengan data *authorized* AP.	
6	Le Wang (2013)	Deteksi *rogue* AP dengan pendekatan berbasis RSSI.	Metode pencatatan *frame* dengan melihat SSID dan RSSI-nya.	Dikatakan terdapat *rogue* AP jika *frame* dengan SSID yang sama memiliki RSSI yang berbeda.
7	Miss Swati Jadhav, Sandeep Vanjale, Dr. P. B. Mane (2014)	Deteksi *fake* AP dengan teknik *fingerprinting*.	Metode perhitungan *clock skew* pada *authorized* AP yang dijadikan nilai *threshold* dan *tested* AP, kemudian dibandingkan nilainya.	Jika nilai *clock skew* pada *tested* AP lebih besar dari nilai *threshold*, maka AP tersebut adalah *fake* AP.
8	Prof. Sandeep Vanjale dan Dr. P. B. Mane (2014)	Deteksi dan eliminasi *rogue* AP.	*Learning mode*, *detection mode*, dan *prevention mode*.	Tiga tahap dalam metode penelitian ini dapat mendeteksi dan melakukan *blocking* terhadap *rogue* AP.
9	Amit A. Chougule, Prof. Sandip B. Vanjale, Dr. P. B. Mane (2015)	Deteksi *rogue* AP menggunakan berbagai macam parameter.	*Tested* AP dicek parameternya seperti SSID, MAC *address*, *authentication type*, *channel*, *frequency*, dan *signal*.	Jika banyak parameter yang sama, maka dapat disimpulkan bahwa *tested* AP tersebut adalah *rogue* AP.
10	M. K. Nivangune, Prof. S. B. Vanjale, Dr. P. B. Mane (2015)	Deteksi *rogue* AP menggunakan pendekatan *client-side*.	Perhitungan CTT (*Complete Tour Time*) pada *client* yang terasosiasi dengan jaringan melalui server lokal dan mencatat waktu total untuk setiap *request* ke server.	Jika nilai rata-rata waktu lebih besar dari nilai *threshold*, maka AP di mana *client* tersebut terasosiasi merupakan *rogue* AP.

1.7 Metodologi Penelitian

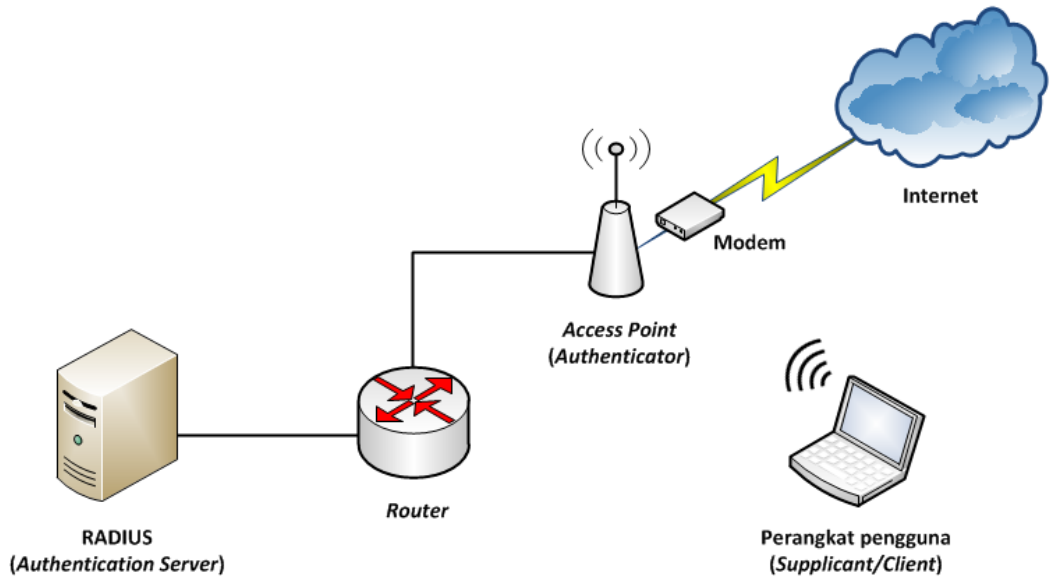
Perlu disusun langkah-langkah penyelesaian penelitian secara sistematis yang disebut sebagai metodologi penelitian. Penelitian ini menggunakan beberapa tahapan metode penelitian yaitu:

1. **Studi Literatur**

   Studi literatur dilakukan untuk mendapatkan informasi mengenai topik penelitian yang dapat bersumber dari dokumen, buku, artikel, atau bahan tertulis lainnya, yang berupa teori, laporan penelitian, atau penemuan sebelumnya, baik bersifat *online source* maupun *offline source.*

2. **Persiapan Sistem**

   Merupakan tahap perancangan dan implementasi WLAN berbasis IEEE 802.1X yang akan digunakan sebagai objek penelitian. Terdiri dari tiga komponen, yaitu server RADIUS sebagai *authentication server*, *wireless Access Point* sebagai *authenticator*, dan perangkat pengguna seperti laptop/tablet sebagai *supplicant*/*client* seperti yang ditunjukkan pada Gambar 1.2.



Gambar 1.2 Arsitektur WLAN berbasis IEEE 802.1X

3. **Simulasi Kasus**

   Merupakan tahap dilakukannya simulasi kasus *rogue Access Point* (*rogue* AP) pada WLAN yang berbasis IEEE 802.1X. Simulasi kasus bertujuan untuk

melakukan pengujian keamanan WLAN berbasis IEEE 802.1X terhadap keberadaan *rogue* AP sekaligus menunjukkan bahaya yang dapat ditimbulkan oleh *rogue* AP. Simulasi kasus yang akan dijalankan menggunakan skenario *client* yang terkoneksi ke *rogue* AP yang telah di-*setting* sehingga tampak mirip dengan AP yang terotorisasi (*authorized* AP). Karena ketidaktahuan *client* tersebut kemudian mengisikan informasi berupa *username* dan *password* untuk dapat login ke WLAN berbasis IEEE 802.1X, sehingga data berupa *username* dan *password* yang valid tersebut dapat diketahui oleh *attacker*. Namun *attacker* tidak dapat langsung mengetahui *password client* karena tidak berbentuk teks jelas (*clear text*). Oleh karena itu, diperlukan proses *cracking* untuk mengetahui *password* milik *client* yang sebenarnya.

4. **Mitigasi Kasus dan Pengembangan *Framework***

   Mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X terdiri dari tahap populasi data *authorized Access Point* (*authorized* AP), deteksi *rogue* AP, dan eliminasi *rogue* AP. Tahap populasi data *authorized* AP dilakukan hanya jika ada perubahan data *authorized* AP di dalam jaringan, yaitu pemasangan *Access Point* baru dan pelepasan *Access Point*. Tahap deteksi *rogue* AP menggunakan *live forensics*, meliputi proses *scanning Access Point* dan *capture* trafik *wireless*, ekstraksi data hasil *scanning*, analisis file hasil *capture*, serta komparasi data yang diperoleh dari hasil *scan* dan *capture* dengan data *authorized* AP sehingga dapat menghasilkan kesimpulan *Access Point* mana yang merupakan *rogue* AP. Tahap eliminasi *rogue* AP yaitu proses pengamanan (*containment*) terhadap *rogue* AP tersebut. Dari langkah mitigasi kasus *rogue* AP tersebut, dikembangkanlah *framework* mitigasi kasus. *Framework* ini akan dijadikan dasar untuk mengimplementasikan otomatisasi langkah mitigasi *rogue* AP pada WLAN berbasis IEEE 802.1X.

5. **Penyusunan Laporan**

   Merupakan tahap pembuatan laporan terhadap hasil pengujian untuk memberikan gambaran secara menyeluruh mengenai topik penelitian ini, serta dapat memberikan rekomendasi yang bermanfaat untuk penelitian-penelitian selanjutnya.

1.8 Sistematika Penulisan

Laporan penelitian ini disusun dengan sistematika penulisan yang dapat mempermudah proses pembahasan penelitian. Adapun sistematika penulisan yang dimaksud adalah sebagai berikut:

BAB I PENDAHULUAN

Pendahuluan merupakan pengantar terhadap permasalahan yang akan dibahas. Bab ini memuat latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, *literature review*, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini memuat teori-teori penunjang yang digunakan sebagai dasar penelitian analisis forensik kasus *rogue Access Point* terhadap WLAN berbasis IEEE 802.1X.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang langkah-langkah penelitian termasuk di dalamnya adalah skenario kasus serangan dan gambaran umum langkah penyelesaian kasus tersebut. Bab ini berisi perancangan dan persiapan sistem, skenario dan simulasi kasus, serta langkah investigasi kasus.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas tentang pengolahan dan analisis data untuk menghasilkan temuan serta analisis dari hasil temuan tersebut. Pengolahan data dilakukan berdasarkan prosedur penelitian yang telah diuraikan pada Bab III.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari hasil penelitian serta saran dan rekomendasi untuk penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 *Digital Forensics*

Forensik digital (*digital forensics*) merupakan ilmu dan metode yang digunakan di dalam pelestarian, pengumpulan, validasi, identifikasi, analisis, interpretasi, dokumentasi, dan presentasi barang bukti digital yang diperoleh dari sumber digital dengan tujuan untuk memfasilitasi atau membuat kemajuan dalam proses rekonstruksi kejadian kriminal, atau membantu dalam antisipasi tindakan yang mengganggu jalannya investigasi yang telah direncanakan (Palmer, 2001). Dari definisi tersebut dapat diketahui bahwa *digital forensics* berguna dalam proses investigasi suatu tindak kejahatan kriminal yang melibatkan adanya barang bukti elektronik atau digital.

Penelitian Karie dan Venter (2014) menyebutkan bahwa *digital forensics* dapat dibagi menjadi enam cabang utama, yaitu *computer forensics*, *software forensics*, *database forensics*, *multimedia forensics*, *device forensics*, dan *network forensics*. Selanjutnya dari keenam cabang tersebut dapat dibagi lagi ke dalam beberapa subcabang, seperti misalnya *network forensics* dibagi ke dalam *cloud forensics*, *telecom network forensics*, *internet forensics*, dan *wireless forensics*.

Penelitian Rafique dan Khan (2013) telah menjelaskan bahwa *digital forensics* dibagi menjadi dua mode, yaitu *static forensics* dan *live forensics*. *Static forensics* menggunakan pendekatan tradisional di mana barang bukti elektronik dibuatkan *bit-by-bit image* untuk dilakukan proses forensik. Proses forensiknya sendiri berjalan pada sistem yang tidak *running* (*off*). Sedangkan pada *live forensics*, proses forensik dilakukan dengan cara mengumpulkan dan menganalisis informasi barang bukti digital pada saat sistem sedang *running* (*on*). *Live forensics* bertujuan untuk melakukan analisis barang bukti tanpa mempengaruhi fungsionalitas sistem, sehingga keseluruhan fungsi yang dijalankan sistem tidak akan terganggu selama proses analisis digital dilakukan.

2.1.1 *Network Forensics*

Seperti telah disebutkan sebelumnya bahwa salah satu dari enam cabang utama *digital forensics* adalah *network forensics*. *Network forensics* atau disebut sebagai forensik jaringan adalah cabang *digital forensics* yang menggunakan teknik yang terbukti secara ilmiah untuk mengumpulkan, menggunakan, mengidentifikasi, menguji, menghubungkan, menganalisis, dan mendokumentasikan barang bukti digital dari beberapa sumber digital yang aktif memproses dan mengirimkannya. Hal ini bertujuan untuk menyingkap fakta, mengukur keberhasilan suatu aktivitas yang tidak terotorisasi, seperti mengganggu, merusak, atau menyusup ke dalam komponen sistem, dan juga untuk menyediakan informasi yang berguna dalam pemulihan sistem terkait dari aktivitas merugikan tersebut (Palmer, 2001).

Network forensics erat kaitannya dengan informasi yang dinamis dan rentan. Maksudnya adalah informasi yang diperlukan dalam *network forensics* mudah hilang setelah ditransmisikan melalui jaringan. Selain itu, barang bukti dalam *network forensics* seperti file-file log di dalam *host* yang telah disusupi juga dapat dihapus oleh penyerang (Karie & Venter, 2014).

Network forensics memiliki beberapa subcabang seperti yang tertera pada Gambar 2.1, yaitu:

1. ***Cloud forensics***

   Cloud forensics merupakan bidang yang erat kaitannya dengan aplikasi teknik *digital forensics* dalam lingkungan *cloud computing* dan merupakan subcabang dari *network forensics* (Keyun, R., Carthy, J., & Kechadi, 2011).

2. ***Telecom network forensics***

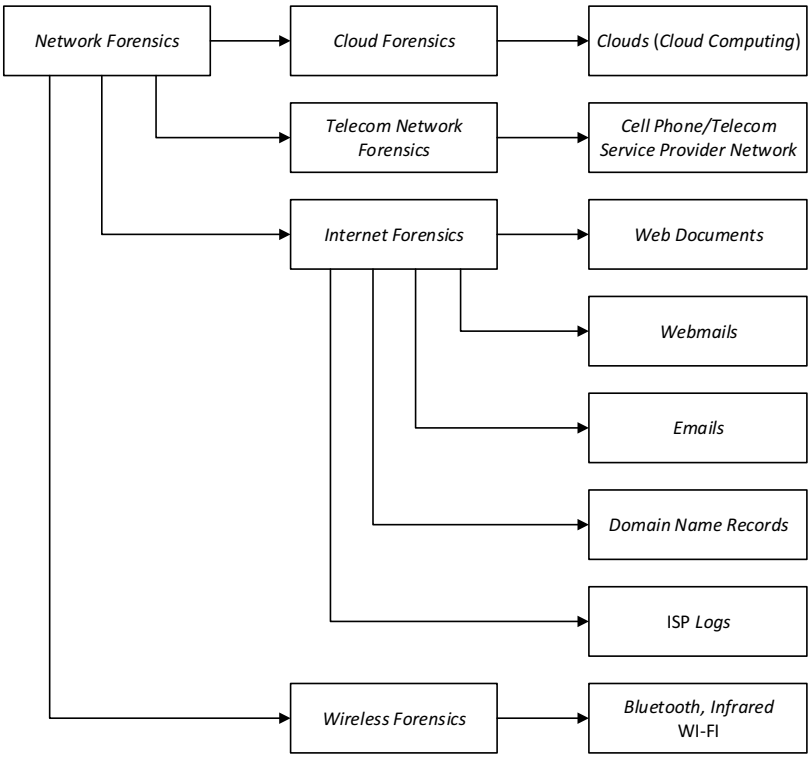
   Sistem persinyalan (*signaling core*) dari telepon pada jaringan publik dapat menghasilkan data yang berharga mengenai pola panggilan telepon yang mungkin digunakan dalam investigasi kejahatan kriminal, terlebih lagi dengan berkembangnya sistem *Voice over* IP (VoIP). Namun sayangnya, banyak data yang dihasilkan dari sistem persinyalan telepon yang tidak dipelihara oleh *service provider*, sehingga data yang diperlukan tersebut tidak tersedia bagi badan penegak hukum (Moore, Meehan, Manes, & Shenoi, 2005).

3. ***Internet forensics***

   Internet forensics merupakan bidang ilmu yang erat kaitannya dengan analisis aktivitas yang terjadi di internet. Tujuannya adalah untuk mendapatkan petunjuk mengenai orang atau komputer yang terlibat dalam kejahatan internet, seperti penipuan kartu kredit dan pencurian identitas. Hal-hal lain yang dapat dilakukan dalam *internet forensics* adalah mengekstrak informasi di dalam *email*, halaman web, atau web server. Informasi tersebut berpotensi mengandung barang bukti digital yang dapat dianalisis untuk kepentingan forensik (Karie & Venter, 2014).

4. ***Wireless forensics***

   Adopsi teknologi *wireless* oleh banyak organisasi berpotensi menimbulkan permasalahan dari segi kontrol dan keamanan. *Wireless forensics* yang muncul sebagai hasil dari adanya teknologi *wireless* difokuskan untuk menangkap (*capture*) atau mengumpulkan data barang bukti digital di jaringan nirkabel (*wireless network*), sehingga pada akhirnya barang bukti digital tersebut dapat dipresentasikan sebagai barang bukti yang valid di pengadilan (Siles, 2007).



Gambar 2.1 Pembagian subcabang *network forensics*
Sumber: Karie & Venter, 2014

2.1.2 *Wireless Forensics*

Langkah pertama yang harus dilakukan untuk menemukan barang bukti dalam lingkungan *wireless network* adalah menemukan peralatan yang mengganggu *wireless network* tersebut dan apakah peralatan tersebut merupakan peralatan yang diotorisasi atau tidak. Banyak perangkat *wireless* yang dapat mengganggu berjalannya *wireless network*, tidak hanya terbatas pada perangkat *wireless* tradisional saja di mana perangkat tersebut digunakan untuk kepentingan konektivitas, seperti *Access Point* (AP), *network adapter* dan *repeater*, namun juga *wireless hard drive*, kamera, bahkan ponsel dan PDA (*Personal Digital Assistant*). Kismet, Netstumbler, dan Airsnort merupakan beberapa *tools* yang berguna dalam menangkap trafik *wireless*. Setelah trafik tersebut dikumpulkan dan diverifikasi, analisis trafik *wireless* dapat dilakukan menggunakan *tools* forensik tradisional (Kipper, 2007).

Langkah-langkah yang dilakukan dalam investigasi forensik di lingkungan *wireless* adalah sebagai berikut (EC-Council, 2010):

1. Mendapatkan *search warrant*.

   Investigator harus memastikan bahwa *search warrant* tersebut mencakup pemeriksaan *on-site* semua komputer dan perangkat *wireless* terkait kasus. Investigator dapat melakukan analisis hanya pada perangkat yang disebutkan di dalam *search warrant* tersebut.
2. Identifikasi perangkat *wireless*.

   Investigator harus mengidentifikasi semua perangkat *wireless* yang terkoneksi ke jaringan. Pengecekan lokasi fisik diperlukan untuk mengidentifikasi *wireless hardware* seperti *wireless router*, *wireless Access Point*, *wireless modem*, *wireless network adapter*, *repeater*, *hard drive*, dan antena.
3. Dokumentasi tempat kejadian dan pemeliharaan *chain of custody*.

   Investigator harus melakukan hal-hal berikut di tempat kejadian:
   a. Mendokumentasikan semua perangkat yang terkoneksi ke *wireless network*.
   b. Mengambil foto semua barang bukti.
   c. Mendokumentasikan status setiap perangkat ketika melakukan pengamanan barang bukti.

d. Melakukan pemeliharaan *chain of custody* dari dokumen, foto, dan barang bukti.

4. Mendeteksi koneksi *wireless*.

   Investigator dapat mendeteksi koneksi *wireless* menggunakan *tools* untuk *scanning* seperti ClassicStumbler, MacStumbler, iStumbler, Airport Signal, Airfart, dan Kismet.

5. Menentukan kekuatan sinyal *wireless*.

   Investigator dapat menggunakan *tool* Field Strength Meter (FSM) untuk menentukan kekuatan sinyal *wireless*.

6. Melakukan pemetaan terhadap zona *wireless* dan *hotspot*.

   Setelah mendeteksi koneksi *wireless* dan mendapatkan informasi lain mengenai perangkat *wireless* yang terlibat di dalam kasus, investigator dapat menganalisis informasi tersebut untuk membuat pemetaan terhadap zona *wireless* dan *hotspot*.

7. Melakukan koneksi ke *wireless network*.

   Metode untuk mengakses *wireless Access Point* adalah sebagai berikut:

   a. Koneksi langsung ke *wireless Access Point*.
   b. Mengendus (*sniff*) trafik di antara *wireless Access Point* dan perangkat yang terkoneksi ke *wireless Access Point* tersebut.

8. Melakukan akuisisi dan analisis data.

   Investigator dapat menggunakan *tools* seperti Wireshark dan tcpdump untuk menangkap trafik *wireless*. Hal lain yang mungkin perlu dilakukan adalah mendapatkan log DHCP, log *firewall*, dan log jaringan.

9. Membuat laporan.

   Laporan investigasi mencakup hal-hal seperti nama investigator, daftar barang bukti *wireless*, dokumen barang bukti, daftar *tool* yang digunakan dalam investigasi, perangkat dan *setup* yang digunakan dalam pemeriksaan, deskripsi singkat langkah pemeriksaan, detail temuan, dan kesimpulan dari investigasi.

2.1.3 *Live Forensics*

Secara tradisional, *static forensics* digunakan untuk investigasi insiden digital. Namun seiring berkembangnya teknologi yang pesat, dan fakta bahwa saat

ini *hacker* dapat mengembangkan *malware* yang tidak meninggalkan jejak pada *hard disk*, maka dibutuhkan analisis forensik secara *live*. Teknik *live forensics* telah berkembang dalam dekade terakhir, seperti analisis konten *memory* untuk mendapatkan gambaran yang lebih baik mengenai aplikasi dan proses yang sedang berjalan (Rahman & Khan, 2015).

Live forensics dilakukan dengan cara mengumpulkan data ketika sistem yang terkena serangan masih berjalan (*running*/*alive*). Data forensik yang dikumpulkan melalui sistem yang *live* tersebut dapat memberikan bukti yang tidak dapat diperoleh dari *static disk image*. Data yang dikumpulkan tersebut merupakan representasi dari sistem yang dinamis dan tidak mungkin untuk diproduksi ulang pada waktu berikutnya (Adelstein, 2006).

Investigator dalam *network forensics* dapat menggunakan teknik *live forensics* untuk mendapatkan informasi yang sebanyak-banyaknya mengenai aktivitas yang terjadi di jaringan. Investigator dapat memasang *packet sniffer* untuk menangkap trafik yang berjalan di jaringan dan menganalisisnya (Reyes et al., 2007). *Tools* seperti Network Miner, tcpflow, Wireshark, dan NetSleuth dapat digunakan dalam *live forensics* di lingkungan *network* (Rafique & Khan, 2013).

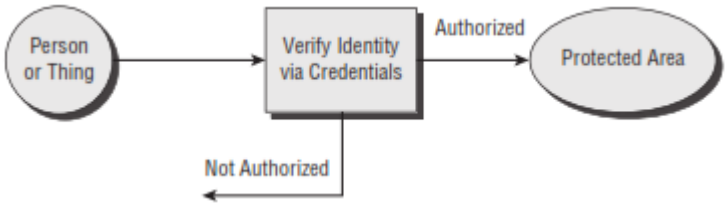
2.2 IEEE 802.1X

IEEE (*Institute of Electrical and Electronics Engineers*), sebuah organisasi internasional yang bertujuan untuk mengembangkan inovasi dan teknologi untuk meningkatkan harkat kemanusiaan telah mengeluarkan standar mengenai keamanan menggunakan *Port-based Network Access Control* (PNAC). Standar ini bernama IEEE 802.1X. Konsep dari IEEE 802.1X ini adalah untuk menangani autentikasi *client* ke jaringan (Kumar, Kumar, & Gambhir, 2014).

Dalam *white paper* "*Introduction to 802.1X for Wireless LANs*" (Networks, 2006) disebutkan bahwa pada awalnya IEEE 802.1X hanya didesain untuk memenuhi kebutuhan keamanan pada *wired Local Area Network* (IEEE 802 LAN) dengan cara menyediakan mekanisme autentikasi perangkat yang terhubung ke port LAN. Namun seiring dengan munculnya teknologi *wireless*, standar IEEE 802.1X

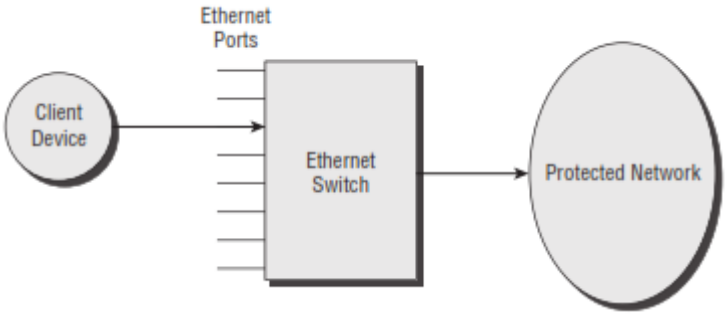
juga dikembangkan untuk memenuhi kebutuhan keamanan terhadap *Wireless Local Area Network* (IEEE 802.11 LAN) atau disebut sebagai WLAN.

Autentikasi adalah proses untuk mengidentifikasi seseorang atau mesin. Proses autentikasi sederhana ditunjukkan pada Gambar 2.2. Dasar proses autentikasi adalah verifikasi terhadap identitas seseorang atau mesin apakah valid atau tidak. Sesuai Gambar 2.2, jika identitas tersebut valid, maka orang atau mesin yang bersangkutan dapat mengakses ke area yang diproteksi oleh mekanisme autentikasi tersebut (Geier, 2008).



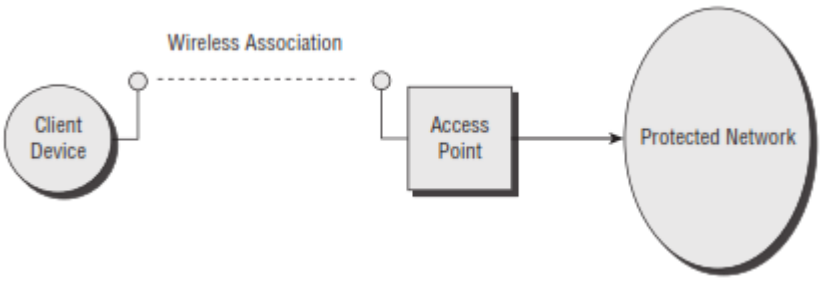
Gambar 2.2 Proses autentikasi secara sederhana
Sumber: Geier, 2008

Seperti telah dijelaskan sebelumnya bahwa IEEE 802.1X merupakan *Port-based Network Access Control*, maka proses autentikasi berlangsung ketika pengguna mengkoneksikan perangkatnya ke port 802.1X. Untuk jaringan berbasis kabel (*wired network*), istilah port merujuk kepada port *Ethernet* di dalam sebuah *switch*. Koneksi *link* yang lengkap dibangun dari *Layer* 1 (*Physical*) dan *Layer* 2 (*Data Link*), di mana kabel *Ethernet* digunakan sebagai sarana interkoneksi fisik dari perangkat pengguna ke port *switch*, sementara mekanisme autentikasi 802.1X melakukan verifikasi identitas untuk perangkat yang terkoneksi ke port *switch* melalui kabel *Ethernet*. Proses autentikasi 802.1X ini berada pada *Layer* 2 (*Data Link*). Gambar 2.3 menunjukkan port *Ethernet* pada *wired network* yang menyediakan koneksi *link* fisik (Geier, 2008).



Gambar 2.3 Port *Ethernet wired network* menyediakan koneksi *link* fisik
Sumber: Geier, 2008

Port pada jaringan berbasis nirkabel (*wireless network*) merujuk kepada perangkat *Access Point* (AP). Perangkat pengguna yang mendukung fitur koneksi *wireless* seperti laptop, akan melakukan proses asosiasi dengan sebuah AP. Ketika perangkat pengguna tersebut pertama kali dihidupkan, perangkat tersebut akan melakukan *scanning* semua *channel* dan mengidentifikasi keberadaan semua AP yang ada di area sekitarnya. Perangkat tersebut kemudian akan melakukan usaha untuk mengasosiasikan dirinya dengan AP yang memiliki sinyal paling kuat. Proses asosiasi dengan AP yang berhasil, akan memperbolehkan perangkat *wireless* (*wireless client*) tersebut berdasarkan MAC *address*-nya, untuk berkomunikasi dengan perangkat lain di dalam infrastruktur jaringan melalui AP. Namun sebelum dapat mengakses perangkat lain dalam jaringan, asosiasi antara *wireless client* dan AP-nya ini menyediakan sebuah *virtual link* di mana *client* tersebut akan dikenai proses autentikasi. Gambar 2.4 menunjukkan proses asosiasi *wireless client* ke AP akan membuat sebuah *virtual link* (Geier, 2008).



Gambar 2.4 Asosiasi *wireless client* ke AP membuat sebuah *virtual link*
Sumber: Geier, 2008

2.2.1 Komponen IEEE 802.1X

Diperlukan tiga komponen dalam membentuk sebuah mekanisme autentikasi 802.1X pada jaringan. Komponen-komponen tersebut adalah sebagai berikut (Geier, 2008):

1. ***Supplicant* (*Client*)**

   Supplicant merupakan perangkat pengguna yang perlu diautentikasi sebelum mengakses jaringan. *Supplicant* dianggap sebagai pengguna yang belum diketahui identitasnya, sampai pengguna tersebut melakukan autentikasi menggunakan identitas yang valid dan diakui oleh sistem. Contoh *supplicant* adalah perangkat pengguna yang memiliki fitur koneksi *wireless*, seperti laptop maupun IP *phone*.

2. ***Authenticator***

   Authenticator merupakan perangkat jaringan yang bekerja pada *Layer* 2, seperti *Ethernet switch* pada *wired network* atau *Access Point* (AP) pada *wireless network*. *Authenticator* bertindak sebagai pintu gerbang antara *supplicant* dan jaringan yang akan diakses oleh *supplicant* tersebut. Port pada *authenticator* masih akan menutup akses ke jaringan sampai *supplicant* dapat memberikan identitas yang valid dan diautentikasi oleh sistem. Setelah sistem setuju bahwa identitas *supplicant* tersebut valid, maka *authenticator* akan membuka port-nya sehingga *supplicant* dapat melakukan akses ke jaringan. *Authenticator* juga berfungsi sebagai jembatan komunikasi antara *supplicant* dan *authentication server*. Semua komunikasi antara *supplicant* dan *authentication server* akan melewati *authenticator*.

3. ***Authenticator Server***

   Supplicant dan *authentication server* akan berkomunikasi mengenai proses autentikasi. *Authentication server* akan meminta identitas dari *supplicant*, dan akan dijawab oleh *supplicant* dengan memberikan identitasnya kepada *authentication server*. Oleh karena *authentication server* yang sering digunakan dalam implementasi jaringan berbasis IEEE 802.1X adalah server RADIUS (*Remote Access Dial-In User Service*), maka server RADIUS dijadikan standar yang dikenal dalam industri jaringan.

2.2.2 Protokol IEEE 802.1X

EAP (*Extensible Authentication Protocol*) adalah sebuah protokol yang umum digunakan dalam autentikasi pengguna jaringan berbasis IEEE 802.1X (Chen & Wang, 2005). Namun, EAP sebenarnya lebih tepat dikatakan sebagai sebuah *framework. Framework* tersebut dijadikan dasar dari berbagai jenis protokol yang diturunkan dari EAP dan protokol baru yang dapat ditambahkan ke dalamnya. EAP juga bertindak sebagai *framework* komunikasi antara *client* dan server, dan protokol yang digunakan dalam komunikasi tersebut dapat dipilih dari protokol yang tersedia di dalam *framework* EAP itu sendiri. Namun antara *client* dan server harus menggunakan protokol yang sama untuk autentikasi dan komunikasi (Kumar et al., 2014).

Beberapa protokol yang ada di dalam *framework* EAP atau disebut juga sebagai EAP *Method* dan umum digunakan untuk berkomunikasi di dalam jaringan berbasis IEEE 802.1X adalah sebagai berikut (Kumar et al., 2014):

1. **EAP-MD5**

   Ketika *client* mengetikkan *password*-nya, maka server akan mengambil MD5 *hash* dari *password* tersebut dan menyimpannya. Jika pada kesempatan yang lain *client* tersebut login kembali, *client* diharuskan mengetikkan *password*-nya. Protokol MD5 yang ada pada sisi *client* akan mengkonversi *password* tersebut menjadi MD5 *hash* dan meneruskannya ke server. Server yang telah menerima MD5 *hash* dari *password client* tersebut akan membandingkan nilainya dengan MD5 *hash* yang telah disimpan di server. Hasil perbandingan akan menentukan apakah *client* bisa login atau tidak. EAP-MD5 ini tidak mendukung adanya autentikasi mutual, karena yang diperiksa hanyalah validitas *client* dan tidak memeriksa validitas server.

2. **EAP-LEAP**

   LEAP (*Lightweight Extensible Authentication Protocol*) merupakan protokol yang dikembangkan oleh Cisco untuk mengatasi kelemahan pada WEP. Berikut adalah cara kerjanya:

   a. Sebuah kunci rahasia (*secret key*) di-*share* di antara *client* dan server.

b. *Client* mengirimkan *challenge* yang dienkripsi dengan *Preshared Session Key* (PSK) secara acak kepada server. Jika server tersebut benar valid, maka server akan dapat mendekripsi *challenge* tersebut dengan kunci rahasia yang telah di-*share* pada langkah pertama tadi.
c. Server memberikan respon kepada *client*.
d. *Client* mendekripsi respon dari server dan membandingkannya dengan *challenge* yang tadi dikirimkan *client*. Jika respon server sesuai dengan *challenge*, maka server tersebut benar valid.
e. *Client* juga diperiksa validitasnya dengan cara yang sama. Jika masing-masing telah diverifikasi validitasnya, maka telah terjadi autentikasi mutual.
f. Informasi yang dipertukarkan di antara *client* dan *server* selama proses autentikasi berlangsung menggunakan *Preshared Session Key* (PSK) untuk mengenkripsi komunikasi tersebut.

EAP-LEAP tidak menyediakan proteksi terhadap identitas *client* karena dikirimkan dalam format *plain text*.

3. **EAP-TLS**

TLS (*Transport Layer Security*) bekerja berdasarkan *Digital Certificate*. *Certificate* untuk *client* dan server ini dikeluarkan oleh pihak ketiga yang terpercaya (*trusted third party*), tetapi untuk mendapatkannya memerlukan biaya yang cukup besar. *Certificate* ini menggunakan PKI (*Public Key Infrastructure*) untuk autentikasi baik dari sisi *client* maupun server, dan terdiri dari dua fase:

a. **TLS *record protocol***

Koneksinya bersifat rahasia dan menggunakan enkripsi data dengan kunci simetris (*data encryption symmetric keys*). Kunci simetris ini bersifat unik untuk setiap koneksi dan didasarkan pada negosiasi rahasia dengan menggunakan protokol lain seperti TLS *handshake protocol*. Banyak di antara protokol dengan level yang lebih tinggi menggunakan TLS *record protocol* untuk enkapsulasi.

b. **TLS *handshake protocol***

Tugas utamanya adalah untuk menyediakan keamanan koneksi. Tugas lainnya adalah untuk memperbolehkan *client* dan *server* melakukan

autentikasi satu sama lain dan membantu negosiasi algoritma *client* dan server seperti halnya pada negosiasi kunci simetris yang digunakan untuk komunikasi. TLS *handshake protocol* memiliki properti sebagai berikut:

- *Client* diidentifikasi berdasarkan *public key*.
- Negosiasi kunci simetris di antara *client* and server selalu aman.
- Negosiasi harus aman sehingga tidak ada penyerang yang dapat membuat perubahan tanpa seizin *client* dan server.

EAP-TLS dipertimbangkan sebagai protokol terbaik dalam EAP namun dalam implementasinya dibutuhkan biaya untuk mendapatkan *Digital Certificate* dari pihak *trusted third party*.

4. **EAP-TTLS**

TTLS (*Tunnel Transport Layer Security*) dikembangkan untuk mengatasi kelemahan TLS yang memerlukan *Digital Certificate* untuk *client* dan server. Pada TTLS hanya server yang membutuhkan *Digital Certificate* dari pihak *trusted third party*. Keunggulan dari TTLS adalah memperbolehkan penggunaan protokol dengan tingkat keamanan yang lebih rendah untuk autentikasi *client*, namun identitas *client* tetap akan disembunyikan oleh TTLS. TTLS terdiri dari dua fase:

a. Menggunakan *server certificate* untuk memeriksa validitas server dan membangun *tunnel* yang dienkripsi secara simetris (*symmetric encrypted tunnel*).
b. Verifikasi identitas *client* dengan cara menggunakan protokol lain namun pemeriksaan identitasnya terjadi di dalam *tunnel* sehingga identitas *client* tersebut aman. Protokol lain yang dapat digunakan untuk kepentingan ini adalah EAP *Method* (EAP-MD5, EAP-LEAP, dan lain-lain) atau *legacy method* (CHAP, MS-CHAP, MS-CHAPv2, PAP, dan lain-lain).

Fungsi utama dari *tunnel* adalah untuk mengamankan metode autentikasi untuk validasi *client*. Jika *client* telah divalidasi, maka *tunnel* akan hancur dengan sendirinya. Setelah itu, merupakan tanggung jawab *client* dan *server* untuk membangun komunikasi yang aman dengan menggunakan *tunnel* baru dengan enkripsi WEP. EAP-TTLS menyediakan tingkat keamanan yang tinggi selama

proses autentikasi dan mendukung semua EAP *Method* dan *legacy method* untuk autentikasi *client*. Selain itu, biaya implementasi tidak sebesar pada EAP-TLS karena hanya memerlukan *Digital Certificate* untuk server saja.

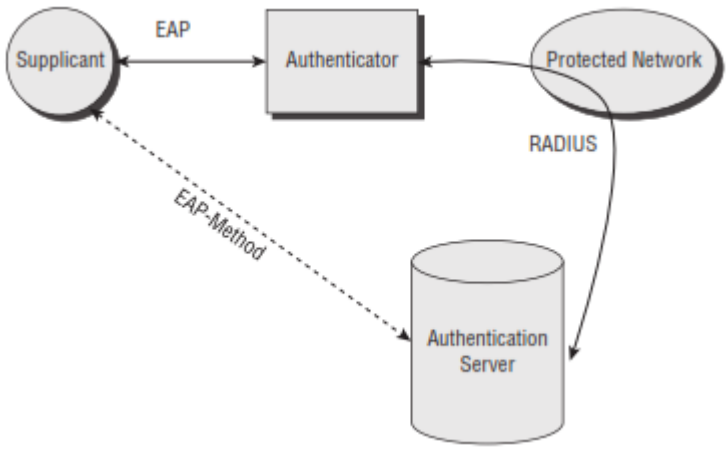
5. **EAP-PEAP**

   PEAP (*Protected Extensible Authentication Protocol*) juga disebut sebagai EAP di dalam EAP. PEAP menggunakan pendekatan *tunnel* yang mirip dengan TTLS karena PEAP juga menggunakan *server certificate* untuk memeriksa validitas server. TLS berbasis *tunnel* digunakan untuk menyediakan enkripsi dan autentikasi. Berbagai jenis serangan dapat dihindari karena pesan dienkapsulasi di dalam *tunnel*. PEAP beroperasi dalam dua fase:

   a. *Client* melakukan validasi terhadap server berdasarkan *server certificate* dan kemudian terjadi negosiasi dan terbentuklah TLS *session*.
   b. Server melakukan validasi *client* dengan hanya menggunakan EAP *Method*, bukan *legacy method*.

   Singkatnya, ketika PEAP *client* melakukan validasi terhadap server berdasarkan *server certificate*, maka terbentuklah sebuah *tunnel* yang aman. Kemudian validasi terhadap *client* dapat dimulai dengan menggunakan EAP *Method* yang mana saja, dan EAP *Method* ini diproteksi oleh TLS.

Gambar 2.5 menunjukkan hubungan *supplicant* (*client*), *authenticator*, *authenticator server*, dan EAP.



Gambar 2.5 Hubungan antara komponen IEEE 802.1X dan EAP
Sumber: Geier, 2008

2.3 *Rogue Access Point*

Rogue Access Point (*rogue* AP) merupakan *wireless Access Point* yang dipasang dalam jaringan komputer yang aman pada sebuah organisasi, namun tanpa otorisasi dari pihak yang mengelola jaringan tersebut. *Rogue* AP juga bisa merupakan *wireless Access Point* yang bertujuan untuk memperbolehkan penyusup melakukan serangan *man in the middle* (Shetty, Song, & Ma, 2007).

Keberadaan *rogue* AP dapat menjadi ancaman serius di dalam jaringan komputer organisasi tersebut, karena *rogue* AP dapat berpotensi membukakan akses bagi pihak yang tidak terotorisasi (*unauthorized parties*) yang bisa saja bermaksud mencuri informasi yang bersifat rahasia atau bahkan melakukan serangan *Denial of Service* (DoS) ke jaringan tersebut (Shivaraj et al., 2008).

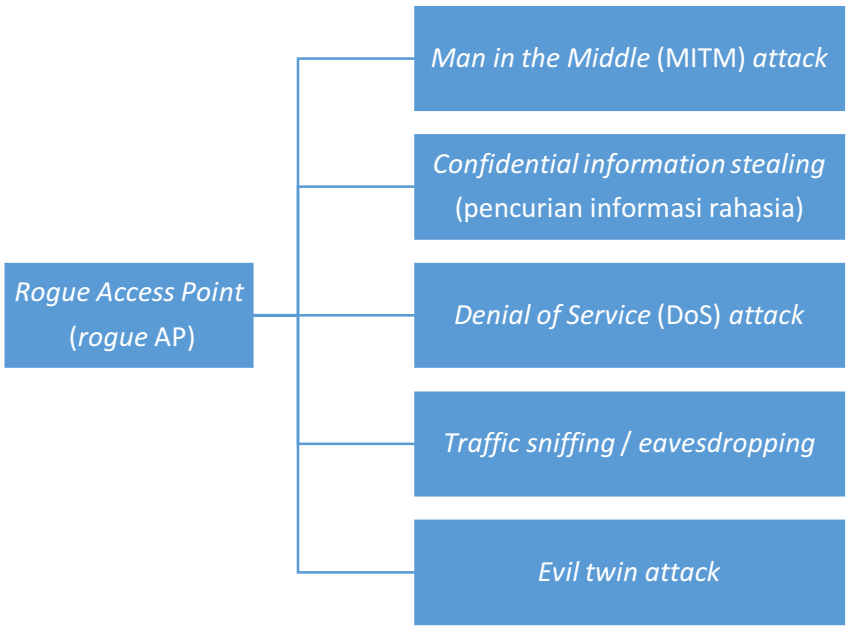
Menurut penelitian Sasane dan Pathan (2011), *rogue* AP umumnya merujuk kepada *Access Point* (AP) yang tidak terotorisasi. AP jenis ini dapat digunakan oleh pihak tertentu yang mungkin melakukan spionase dan melakukan serangan ke jaringan. Keberadaan *rogue* AP merupakan tantangan besar bagi pihak pengelola jaringan karena berbeda dengan ancaman keamanan lain yang umumnya membutuhkan pengetahuan teknis level tinggi atau perangkat penyusup yang canggih dan mahal, *rogue* AP dapat dengan mudah dibuat oleh pihak dengan latar belakang keamanan (*security background*) yang terbatas. Terlebih lagi, perangkat seperti *wireless network card* yang dapat menangkap trafik *wireless* dapat dibeli di situs *online* dengan harga yang relatif murah.

Pekerja dalam sebuah organisasi dapat mengkonfigurasi *rogue* AP dengan dua alasan yang memungkinkan. Alasan pertama adalah pekerja tersebut tidak menyadari aturan keamanan di dalam organisasi tersebut. Sedangkan alasan kedua adalah pekerja tersebut dengan sengaja memasang perangkat untuk melakukan serangan dari dalam organisasi dengan tujuan yang berbahaya. Media yang digunakan dalam komunikasi *Wireless* LAN (WLAN) adalah media *wireless*, sehingga memungkinkan adanya serangan berupa *eavesdropping* untuk mengetahui komunikasi data *wireless* yang terjadi di dalam WLAN (Jadhav et al., 2014).

Rogue AP juga dapat dikonfigurasi mirip dengan AP yang asli (*legitimate* AP). Konfigurasi yang mirip ini menimbulkan serangan berupa *evil twin attack*.

Banyak pengguna *wireless* dapat dengan mudah melakukan koneksi ke AP dengan sinyal yang paling kuat. Padahal bisa saja AP tersebut adalah *rogue* AP jenis *evil twin* yang sengaja dipasang oleh penyusup. Ketika seorang pengguna melakukan koneksi ke sebuah *rogue* AP, penyusup tersebut dapat melakukan *monitoring* dan menangkap semua trafik jaringan pengguna tersebut (Nikbakhsh et al., 2012).

Gambar 2.6 menunjukkan serangan-serangan yang mungkin terjadi akibat keberadaan *rogue* AP dalam jaringan komputer sebuah organisasi.



Gambar 2.6 Serangan yang mungkin terjadi akibat *rogue* AP

Berdasarkan studi yang dilakukan oleh Gartner, keberadaan *rogue* AP mencapai sekitar 20% dari jaringan komputer *enterprise*. Alasan di balik banyaknya *rogue* AP ini adalah perkembangan *hardware* dan *software* yang memungkinkan tugas-tugas seperti instalasi AP, *discovery* AP, dan penyalahgunaan AP menjadi lebih mudah untuk dilakukan oleh penyerang (Shivaraj et al., 2008). *Rogue* AP dibagi menjadi dua jenis, yaitu *hardware based rogue* AP dan *software based rogue* AP (Saruhan, 2007).

2.3.1 *Hardware Based Rogue* AP

Hardware based rogue AP merupakan *rogue* AP konvensional yang berbasis *hardware*, seperti *unmanaged wifi router* yang dipasang di jaringan (Chaskar, 2010). *Rogue* AP juga dapat berupa *unmanaged* AP yang dipasang di jaringan komputer *enterprise* oleh pihak-pihak yang memiliki maksud berbahaya, seperti *malicious employee* atau bahkan *malicious visitor* (K.N. & Chaskar, 2009).

Dalam *white paper* "*All You Wanted to Know About WiFi Rogue Access Points*" (K.N. & Chaskar, 2009) disebutkan bahwa *rogue* AP ini dicolokkan ke jaringan komputer *enterprise* melalui port pada *switch* dan dapat mengekspos jaringan kabel ke pihak luar melalui kebocoran sinyal *Radio Frequency* (RF) dari AP tersebut. Beberapa contoh dari *rogue* AP berbasis *hardware* ini adalah sebagai berikut:

1. AP dengan koneksi *wireless* yang terenkripsi (*encrypted*).
2. AP dengan koneksi *wireless* yang tidak terenkripsi (*open*).
3. AP pada *Virtual* LAN yang berbeda-beda di dalam LAN, termasuk *Virtual* LAN dengan *subnet* yang tidak dipergunakan untuk *wifi*.

2.3.2 *Software Based Rogue* AP

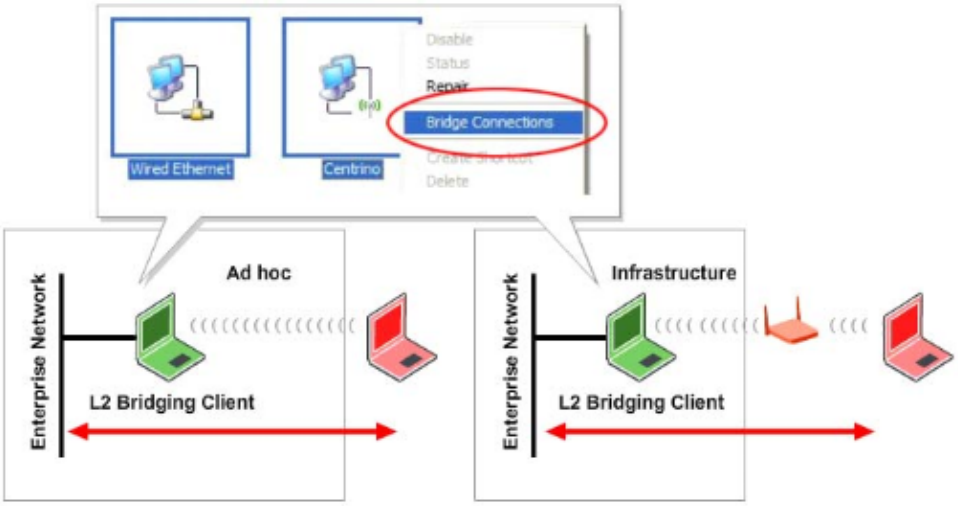
Software based rogue AP atau *software based* AP (*soft* AP) adalah sebuah laptop atau perangkat lain yang mendukung fitur *wireless* yang meneruskan trafik antara *wireless interface*-nya dengan *interface* lain yang terkoneksi ke jaringan. *Soft* AP ini dapat menjadi sebuah *rogue* AP yang berbasis *software* di dalam jaringan *enterprise*. Hal ini dapat terjadi secara tidak sengaja, seperti misalnya saat karyawan sebuah perusahaan mengaktifkan fitur *internet connection sharing* di laptop kantornya saat berada di rumah, namun lupa menonaktifkan fitur tersebut saat kembali ke kantor. *Soft* AP juga dapat di-*install* untuk tujuan berbahaya mengingat *soft* AP ini merupakan *rogue* AP yang dapat menghindarkan diri dari fitur keamanan 802.1X, *Network Access Control* (NAC), dan *rogue* AP *scanner* pada jaringan berbasis kabel (Chaskar, 2010).

White paper "*Conquering the Minefield of Soft Rogue APs in the Enterprise*" (Chaskar, 2010) menyebutkan bahwa *soft* AP pada masa sekarang ini meningkat penggunaannya di dalam jaringan komputer *enterprise*. Alasan utamanya adalah perangkat pengguna telah mengaktifkan *soft* AP melalui konfigurasi yang tertanam di dalam *wireless interface*-nya. Pada *hardware based rogue* AP diperlukan pemasangan sebuah *hardware* seperti *wireless router* di dalam jaringan komputer *enterprise*, sedangkan pada *software based rogue* AP, *soft* AP-nya sendiri telah tertanam di dalam perangkat pengguna.

Berikut adalah beberapa contoh konfigurasi *soft* AP yang berpotensi menjadi *software based rogue* AP (Chaskar, 2010):

1. ***Windows Network Bridge***

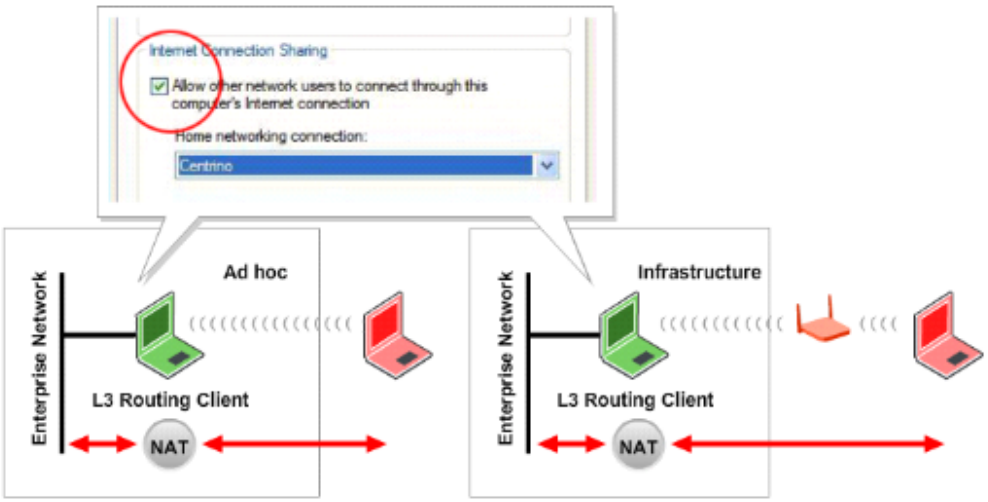
   Sebuah *network bridge* dapat dibuat di antara *wired interface* dan *wireless interface* pada sebuah laptop dengan sistem operasi Windows. Jika laptop dengan *wired interface* terkoneksi ke jaringan, maka laptop tersebut dapat membuat *bridge connection* ke laptop lain melalui *wireless interface*-nya. Mungkin saja terdapat laptop penyusup yang melakukan koneksi ke *wireless interface* laptop yang membuat *bridge connection*, sehingga penyusup akan bisa mendapatkan akses *Layer* 2 ke jaringan tersebut. Gambar 2.7 menunjukkan konfigurasi *soft* AP pada *Windows Network Bridge*.



Gambar 2.7 Konfigurasi *soft* AP pada *Windows Network Bridge*
Sumber: Chaskar, 2010

2. ***Internet Connection Sharing* (ICS)**

Mirip seperti pada konfigurasi *Windows Network Bridge*. Hanya saja fitur yang diaktifkan di laptop Windows bukan *Bridge Connections*, melainkan fitur *Internet Connection Sharing* (ICS). Pada saat ICS diaktifkan pada laptop yang terkoneksi ke jaringan melalui *wired interface-nya*, maka *routing service* akan terbentuk di antara *wired interface* dan *wireless interface*. Jika laptop penyusup melakukan koneksi ke laptop yang mengaktifkan fitur ICS, maka penyusup akan bisa mendapatkan akses *Layer* 3 ke jaringan. Gambar 2.8 menunjukkan konfigurasi *soft* AP pada laptop dengan fitur ICS.

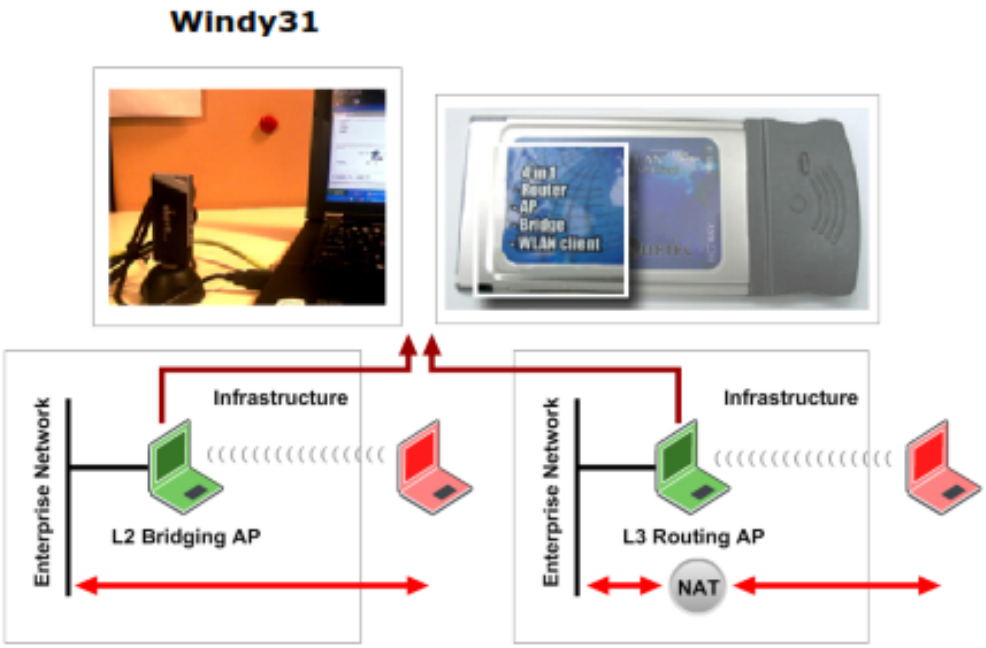


Gambar 2.8 Konfigurasi *soft* AP pada laptop dengan fitur ICS
Sumber: Chaskar, 2010

3. **Perangkat tambahan pada laptop**

Perangkat eksternal dapat dikoneksikan ke laptop untuk membuat laptop tersebut menjadi sebuah *soft* AP. Sebagai contoh, perangkat USB seperti Windy31 dan kartu PCMCIA seperti WP1200 dapat bertindak seperti *wireless Access Point* ketika dicolokkan ke laptop. Perangkat seperti Windy31 ini telah memiliki *software* AP di dalamnya yang *auto-install* ketika perangkat tersebut dicolokkan ke laptop. *Wireless* AP yang berjalan pada perangkat-perangkat tersebut dapat melakukan *bridging* atau *routing* trafik ke jaringan kabel melalui laptop. Gambar 2.9 menunjukkan konfigurasi *soft* AP yang dilakukan dengan

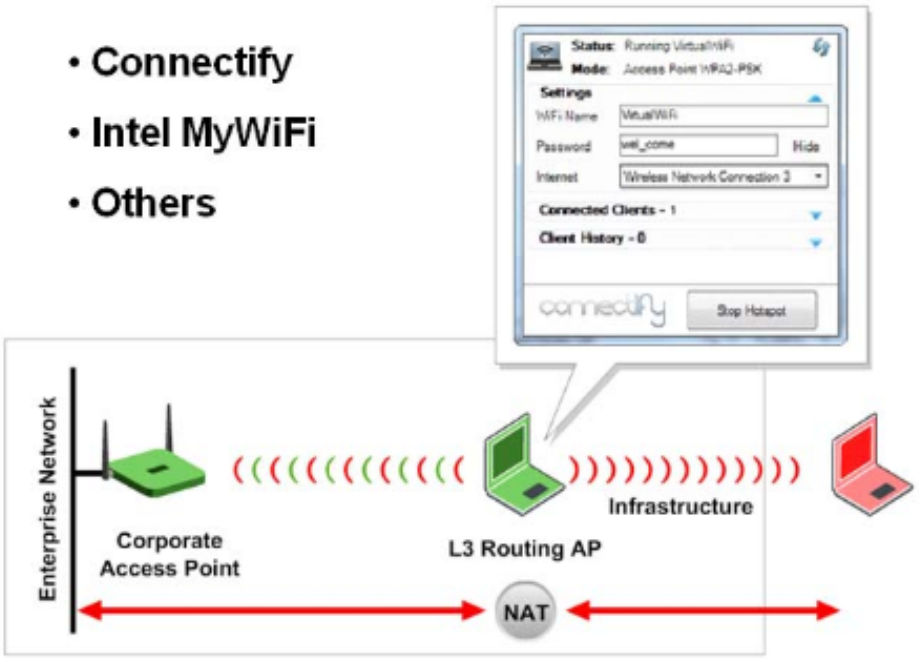
menambahkan perangkat yang memiliki *software* AP di dalamnya yang dicolokkan ke laptop.



Gambar 2.9 Konfigurasi *soft* AP pada perangkat tambahan
Sumber: Chaskar, 2010

4. ***Virtual Wifi Interface* (Windows 7)**

   Windows 7 memperkenalkan fitur *Virtual Wifi Interface* yang berguna untuk mengaktifkan sebuah *radio interface* agar dapat bertindak menjadi beberapa perangkat *wireless* secara simultan. Penggunaan *tool* Connectify dapat mengaktifkan *client* dan AP secara simultan pada *radio interface* laptop Windows 7. Jika *radio interface* pada laptop tersebut terkoneksi ke jaringan dan terbaca sebagai *wireless client* yang terotorisasi, maka *client* yang terkoneksi ke laptop tersebut melalui *radio interface* yang sama, juga dapat melakukan akses ke jaringan. Gambar 2.10 menunjukkan konfigurasi *soft* AP menggunakan *Virtual Wifi Interface* pada laptop Windows 7.



Gambar 2.10 Konfigurasi *soft* AP dengan *Virtual Wifi Interface*
Sumber: Chaskar, 2010

5. ***Soft* AP pada perangkat genggam**

   Perangkat genggam umumnya tidak memiliki *wired interface*, sehingga metode yang dapat digunakan untuk membuat perangkat genggam menjadi *soft* AP adalah dengan penggunaan *virtual interface*. Hal ini mirip seperti konfigurasi *Virtual Wifi Interface* pada Windows 7. Gambar 2.11 menunjukkan *wireless radio* pada perangkat genggam dapat bertindak sebagai *wireless client* yang terotorisasi dan sebagai *soft* AP yang tidak terotorisasi (*soft rogue* AP).

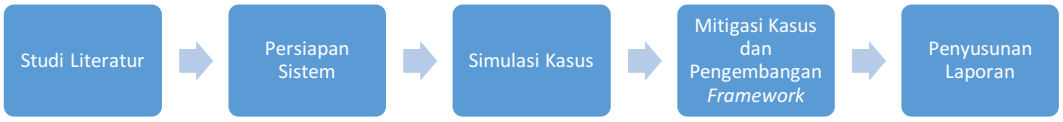


Gambar 2.11 Konfigurasi *soft* AP pada perangkat genggam
Sumber: Chaskar, 2010

BAB III

METODOLOGI PENELITIAN

Bab ini menjelaskan bagaimana cara penelitian dilakukan sehingga dapat diketahui rincian tentang urutan langkah-langkah yang dibuat secara sistematis dan dapat dijadikan pedoman yang jelas dalam menyelesaikan permasalahan, membuat analisis terhadap hasil penelitian, serta kesulitan-kesulitan yang dihadapi. Adapun langkah-langkah atau tahapan-tahapan pada penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur metode penelitian

3.1 Studi Literatur

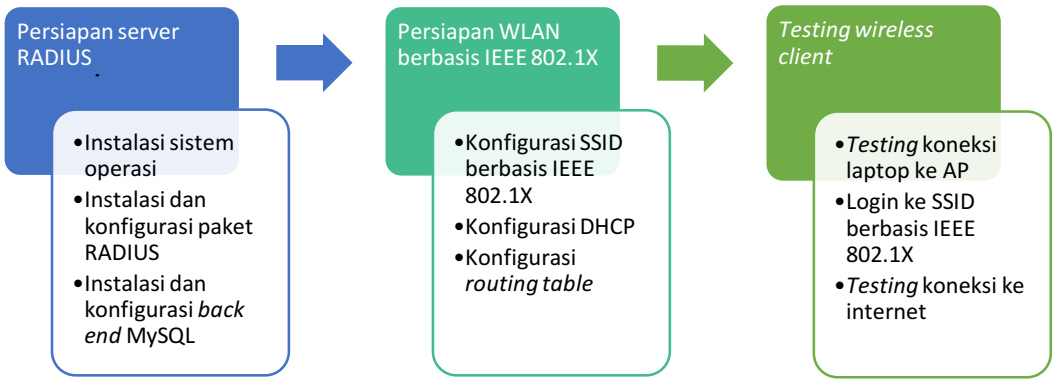
Studi literatur dilakukan untuk mendapatkan informasi mengenai topik penelitian yang dapat bersumber dari dokumen, buku, artikel, atau bahan tertulis lainnya, yang berupa teori, laporan penelitian, atau penemuan sebelumnya, baik bersifat *online source* maupun *offline source*.

Review atau kajian pustaka dilakukan terhadap penelitian yang terkait dengan masalah-masalah pada deteksi *rogue Access Point* (*rogue* AP), berikut juga metode-metode yang digunakan untuk melakukan proses deteksi, sehingga dapat menunjang tujuan akhir dilakukannya penelitian ini.

3.2 Persiapan Sistem

Merupakan tahap perancangan dan implementasi WLAN berbasis IEEE 802.1X yang akan digunakan sebagai objek penelitian. Terdiri dari tiga komponen, yaitu server RADIUS sebagai *authentication server*, *wireless Access Point* sebagai

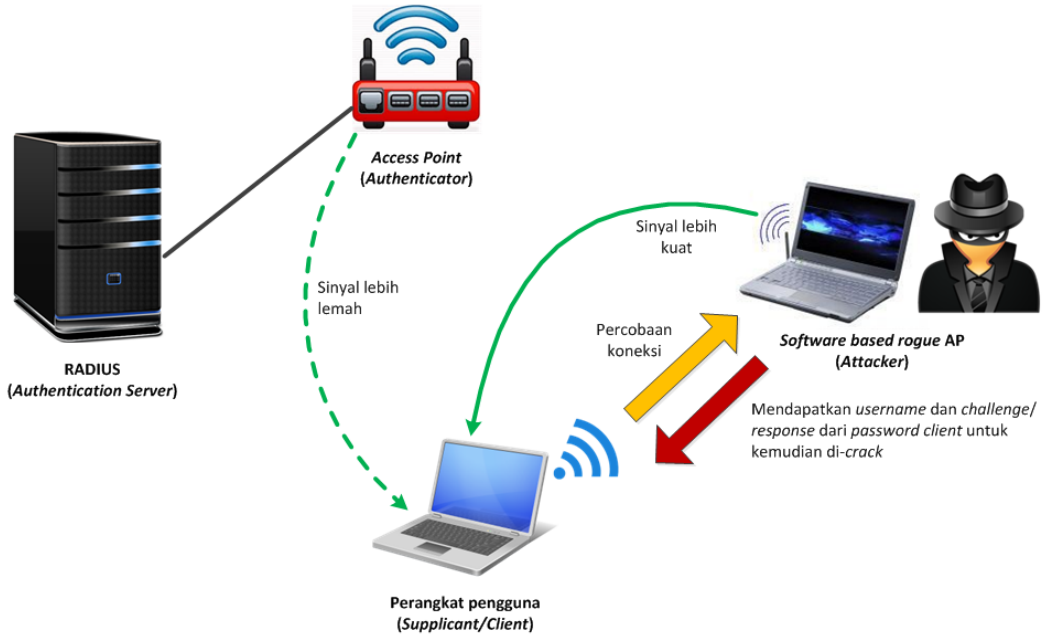
authenticator, dan perangkat pengguna seperti laptop/tablet sebagai *supplicant*/*client*. Gambar 3.2 menunjukkan tahap implementasi WLAN berbasis IEEE 802.1X.



Gambar 3.2 Tahapan implementasi WLAN berbasis IEEE 802.1X

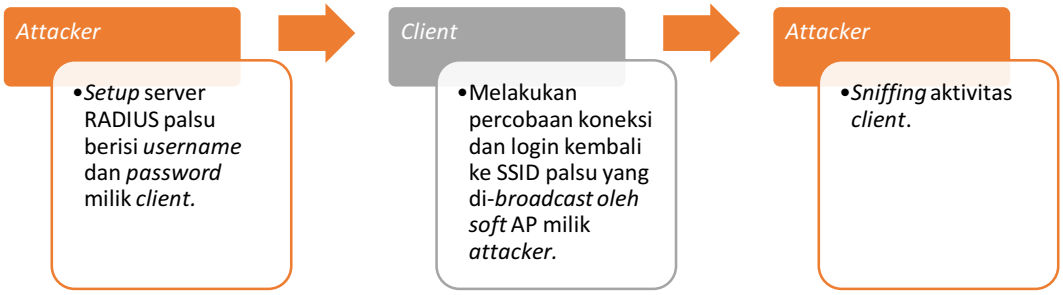
3.3 Simulasi Kasus

Merupakan tahap dilakukannya simulasi kasus *rogue Access Point* (*rogue* AP) pada WLAN yang berbasis IEEE 802.1X. Simulasi kasus bertujuan untuk melakukan pengujian keamanan WLAN berbasis IEEE 802.1X terhadap keberadaan *rogue* AP sekaligus menunjukkan bahaya yang dapat ditimbulkan oleh *rogue* AP. Simulasi kasus yang akan dijalankan menggunakan skenario *client* yang terkoneksi ke *rogue* AP yang telah di-*setting* sehingga tampak mirip dengan AP yang terotorisasi (*authorized* AP). Karena ketidaktahuan *client* tersebut kemudian mengisikan informasi berupa *username* dan *password* untuk dapat login ke WLAN berbasis IEEE 802.1X, sehingga data berupa *username* dan *password* yang valid tersebut dapat diketahui oleh *attacker*. Namun *attacker* tidak dapat langsung mengetahui *password client* karena tidak berbentuk teks jelas (*clear text*). Oleh karena itu, diperlukan proses *cracking* untuk mengetahui *password* milik *client* yang sebenarnya. Gambar 3.3 menunjukkan gambaran umum dari skenario kasus *rogue* AP pada WLAN berbasis IEEE 802.1X.



Gambar 3.3 Skenario *rogue* AP pada WLAN berbasis IEEE 802.1X

Skema serangan adalah setelah *attacker* mendapatkan *username* dan *password client*, maka *attacker* dapat membuat server RADIUS-nya sendiri dengan mendaftarkan *username* dan *password client* tersebut ke dalam server RADIUS-nya. Apabila *client* tersebut terkoneksi kembali ke *rogue* AP yang telah di-*setting* oleh *attacker*, maka *client* dapat login ke SSID palsu milik *attacker* dengan menggunakan *username* dan *password*-nya tersebut, sehingga *attacker* dapat memantau aktivitas (*sniffing*) yang dilakukan *client*, seperti *browsing* dan lain-lain. Gambar 3.4 menunjukkan tahapan simulasi kasus setelah *attacker* mendapatkan *username* dan *password client*.



Gambar 3.4 Tahapan simulasi kasus setelah *attacker* mendapatkan *username* dan *password client*

3.4 Mitigasi Kasus dan Pengembangan *Framework*

Mitigasi kasus *rogue Access Point* (*rogue* AP) pada WLAN berbasis IEEE 802.1X terdiri dari tahap populasi data *authorized Access Point* (*authorized* AP), deteksi *rogue* AP, dan eliminasi *rogue* AP. Tahap populasi data *authorized* AP dilakukan hanya jika ada perubahan data *authorized* AP di dalam jaringan, yaitu pemasangan *Access Point* baru dan pelepasan *Access Point*. Tahap deteksi *rogue* AP menggunakan *live forensics*, meliputi proses *scanning Access Point* dan *capture* trafik *wireless*, ekstraksi data hasil *scanning*, analisis file hasil *capture*, serta komparasi data yang diperoleh dari hasil *scan* dan *capture* dengan data *authorized* AP sehingga dapat menghasilkan kesimpulan *Access Point* mana yang merupakan *rogue* AP. Tahap eliminasi *rogue* AP yaitu proses pengamanan (*containment*) terhadap *rogue* AP tersebut. Dari langkah mitigasi kasus *rogue* AP tersebut, dikembangkanlah *framework* mitigasi kasus. *Framework* ini akan dijadikan dasar untuk mengimplementasikan otomatisasi langkah mitigasi *rogue* AP pada WLAN berbasis IEEE 802.1X.

3.4.1 Populasi Data *Authorized Access Point*

Tahap populasi data *authorized Access Point* (*authorized* AP) dilakukan jika ada perubahan dalam jaringan. Maksudnya adalah penambahan data *authorized* AP dilakukan jika ada *Access Point* yang baru selesai dipasang di jaringan. Sebaliknya, jika ada *Access Point* yang dilepas dari jaringan, maka data *Access Point* tersebut akan dihapus. Hal ini dilakukan agar data-data dari *authorized* AP ter-*update* dengan baik.

Tabel 3.1 Presentasi data *authorized Access Point*

No.	**SSID**	**BSSID (MAC *Address*)**	***Channel***	***Power***	***Encryption***	**EAP *Type***

Hasil populasi data akan dipresentasikan seperti tampak pada Tabel 3.1 kolom SSID, BSSID (MAC *Address*), *Channel*, *Power*, *Encryption*, dan EAP *Type*. EAP *Type* merupakan jenis protokol EAP yang digunakan pada WLAN berbasis IEEE 802.1X.

3.4.2 Deteksi *Rogue Access Point*

Tahap deteksi *rogue Access Point* (*rogue* AP) menggunakan metode *live forensics* karena tahap ini dapat dilakukan hanya jika sistem sedang berjalan. Tahap deteksi *rogue* AP ini terdiri dari beberapa proses, yaitu:

1. ***Capture***

   Proses *Capture* yang merupakan proses pertama dalam *live forensics* untuk deteksi *rogue* AP adalah *scanning Access Point* dan *capture* trafik *wireless*. *Scanning Access Point* bertujuan untuk mengetahui keberadaan *Access Point* yang menyebarkan (*broadcast*) SSID pada jaringan *wireless*. Informasi yang diinginkan dari proses *scanning* ini di antaranya adalah nama SSID, MAC *address Access Point*, *channel* yang digunakan, kekuatan sinyal (*power*), dan tipe enkripsi yang digunakan SSID *Access Point*. Sedangkan *capture* trafik *wireless* bertujuan untuk mengetahui ada tidaknya protokol EAP yang digunakan SSID tersebut dan jenis protokol EAP yang digunakan jika ada. *Tools* yang akan digunakan dalam proses *Capture* ini adalah sebagai berikut:

   a. **Iwlist**

      Iwlist merupakan *wireless detection tool* yang melakukan *scanning* terhadap nama SSID, MAC *address Access Point*, *channel* yang digunakan, kekuatan sinyal (*power*), dan tipe enkripsi yang digunakan SSID tersebut.

   b. **Airmon-ng**

      Airmon-ng digunakan untuk mengubah *wireless interface* ke mode monitor. Berbeda dengan mode *promiscuous* yang melakukan *capture* trafik *wireless* hanya jika terasosiasi dengan *Access Point*, mode monitor dapat *capture* trafik *wireless* tanpa harus terasosiasi ke *Access Point* (Wikipedia, 2015). Mengubah *wireless interface* ke mode monitor dilakukan agar dapat

melakukan *capture* trafik *wireless* tanpa harus terkoneksi ke *Access Point* tertentu.

c. Airodump-ng

Digunakan untuk *capture frame* dari trafik *wireless* berbasis IEEE 802.1X, yaitu protokol 802.1X (EAP) dengan menggunakan mode monitor pada *wireless interface*. *Tool* Airodump-ng ini dapat mengumpulkan informasi berupa nama SSID, MAC *address* perangkat *wireless*, *channel*, *power*, tipe enkripsi yang digunakan SSID, dan ada tidaknya *client* yang terkoneksi ke *Access Point* tersebut. Airodump-ng tidak memberikan informasi secara kasat mata mengenai jenis protokol EAP yang digunakan, sehingga untuk mengetahuinya diperlukan analisis terhadap hasil *capture* Airodump-ng menggunakan *tool* lain seperti Wireshark (Alamanni, 2015). Informasi yang diinginkan dari hasil *capture* Airodump-ng ini adalah ada tidaknya protokol EAP yang digunakan SSID *Access Point* dan jenis protokol EAP-nya jika ada.

2. *Note*

Proses *Note* dilakukan untuk mencatat parameter yang diperlukan dalam proses deteksi *rogue* AP. Data hasil *scanning Access Point* diekstrak sesuai parameter-parameter yang berupa SSID, MAC *address*, *channel*, *power*, dan tipe enkripsi yang digunakan SSID. Data hasil ekstraksi ini kemudian dicatatkan atau dipresentasikan seperti pada Tabel 3.2 kolom SSID, BSSID (MAC *Address*), *Channel*, *Power*, dan *Encryption*. Selain itu, nantinya dilakukan juga pencatatan terhadap hasil analisis *capture* trafik *wireless*. Data yang berupa jenis protokol EAP ini kemudian dicatatkan atau dipresentasikan seperti pada Tabel 3.2 kolom EAP *Type*. Namun untuk mendapatkan jenis protokol EAP terlebih dahulu harus dilakukan proses *Analyze*.

3. *Analyze*

Proses *Analyze* atau analisis dilakukan terhadap file hasil *capture* dari Airodump-ng untuk mengetahui jenis protokol EAP yang digunakan pada WLAN berbasis IEEE 802.1X. Analisis dapat dilakukan menggunakan *tool* Wireshark maupun Tshark. Data yang berupa jenis protokol EAP ini kemudian dicatatkan atau dipresentasikan melalui proses *Note*.

4. *Report*

Tabel 3.2 Presentasi data *unauthorized Access Point*

No.	**SSID**	**BSSID (MAC *Address*)**	***Channel***	***Power***	***Encryption***	**EAP *Type***

Proses *Report* menghasilkan data *Access Point* yang merupakan hasil dari proses *Capture*, *Note*, dan *Analysis*. Selain itu, proses *Report* juga menghasilkan kesimpulan *Access Point* mana yang merupakan *rogue* AP. Untuk menghasilkan kesimpulan tersebut, dilakukan komparasi data dengan cara memperbandingkan data pada Tabel 3.2 dengan data *authorized* AP seperti yang dipresentasikan pada Tabel 3.1.

a. Komparasi data kolom SSID, BSSID (MAC *Address*), *Encryption*, dan EAP *Type*.
b. Kolom *Channel* tidak diperbandingkan karena berapapun *channel* yang digunakan oleh *Access Point*, tetap ada kemungkinan untuk menjalankan *rogue* AP. Namun data *channel* tetap akan diperlukan untuk melakukan *capture* trafik *wireless* yang nantinya digunakan dalam analisis jenis protokol EAP.
c. Untuk kolom *Power*, akan ditentukan kekuatan sinyal minimum dalam satuan dBm sebagai nilai *threshold*, sehingga jika *power*-nya lebih dari atau sama dengan nilai yang ditentukan, maka terdapat kemungkinan bahwa *Access Point* tersebut merupakan *rogue* AP.

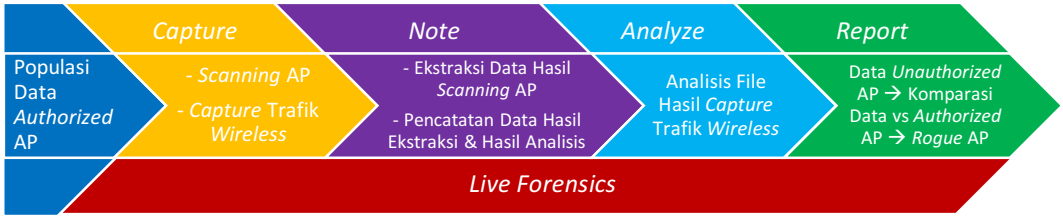
Pada awalnya, data *Access Point* pada Tabel 3.2 dikategorikan sebagai *unauthorized* AP. Kemudian proses komparasi dimulai dengan cara memperbandingkan data SSID terlebih dahulu. Jika SSID pada Tabel 3.2 berbeda dengan SSID yang ada pada Tabel 3.1, maka dapat dikategorikan sebagai *neighbor* AP. Kemudian dilanjutkan dengan proses komparasi SSID dan MAC *address* yang ada pada Tabel 3.2. Apabila SSID dan MAC *address* sama dengan data yang ada pada Tabel 3.1, maka AP tersebut merupakan *authorized*

AP. Selanjutnya data *unauthorized* AP yang masih tersisa akan diproses lebih lanjut untuk mendeteksi *Access Point* mana yang merupakan *rogue* AP. Digunakan aturan atau ketentuan sebagai berikut:

a. SSID yang digunakan oleh *rogue* AP sama dengan SSID milik *authorized* AP.
b. MAC *address* milik *rogue* AP berbeda dengan MAC *address* milik *authorized* AP.
c. *Channel* yang digunakan oleh *rogue* AP bisa sama ataupun berbeda dari *channel* yang digunakan oleh *authorized* AP. Biasanya *Access* Point dikonfigurasi menggunakan *channel* 1, 6, atau 11 untuk menghindari interferensi (Metageek, n.d.-b).
d. Kekuatan sinyal (*power*) yang ditransmisikan oleh *rogue* AP berada pada rentang sinyal yang baik untuk melakukan konektivitas (*basic connectivity*), yaitu minimum sinyal -80 dBm (Metageek, n.d.-a).
e. Tipe enkripsi yang digunakan oleh *rogue* AP sama dengan tipe enkripsi yang digunakan oleh *authorized* AP.
f. Mengetahui ada atau tidaknya protokol EAP yang digunakan oleh suatu *Access Point*, jika ada maka harus diketahui jenis protokol EAP-nya.
g. Protokol EAP yang digunakan oleh *rogue* AP sama dengan protokol EAP yang digunakan oleh *authorized* AP.

Namun apabila tidak semua aturan di atas terpenuhi oleh sebuah AP, maka AP tersebut dapat dikategorikan sebagai *neighbor* AP.

Gambar 3.5 menunjukkan tahap populasi data *authorized* AP dan tahap deteksi *rogue* AP pada WLAN berbasis IEEE 802.1X.



Gambar 3.5 Tahapan populasi data *authorized* AP dan deteksi *rogue* AP

Tahap deteksi *rogue* AP dalam penelitian ini diadopsi dari beberapa penelitian yang telah dilakukan sebelumnya, yaitu:

1. Metode deteksi dan komparasi SSID, kekuatan sinyal, dan IP *address Access Point* (Jagtap & Honwadkar, 2010).
2. Deteksi *rogue* AP menggunakan algoritma RAPD dengan cara melakukan *sniffing* dan membandingkan data hasil *sniffing* dengan data *authorized* AP (Thakur & Bodhe, 2013).
3. Deteksi *rogue* AP menggunakan berbagai macam parameter, seperti SSID, MAC *address*, *authentication type*, *channel*, *frequency*, dan *signal*. Jika banyak parameter yang sama, maka dapat disimpulkan bahwa *Access Point* tersebut adalah *rogue* AP (Chougule et al., 2015).

Oleh karena penelitian ini dilakukan di lingkungan WLAN berbasis IEEE 802.1X, maka perlu diketahui jenis protokol EAP yang digunakan. Adopsi metode deteksi *rogue* AP dari ketiga penelitian yang telah disebutkan sebelumnya akan dikembangkan dengan penambahan data EAP *type* yang digunakan.

Proses deteksi *rogue* AP dilakukan sebagai langkah preventif untuk mengatasi munculnya *rogue* AP yang dapat mengakibatkan terjadinya pencurian *username* dan *password* milik *client* yang login ke SSID berbasis IEEE 802.1X. Apabila *rogue* AP telah dapat dideteksi, maka langkah selanjunya adalah melakukan proses eliminasi *rogue* AP tersebut.

3.4.3 Eliminasi *Rogue Access Point*

Merupakan tahap dilakukannya langkah-langkah eliminasi *rogue Access Point* (*rogue* AP), meliputi langkah untuk mengecek apakah ada *client* yang terkoneksi ke *rogue* AP dan langkah untuk melakukan pengamanan (*containment*) terhadap *rogue* AP tersebut. Dengan asumsi bahwa *rogue* AP telah dapat dideteksi, maka langkah-langkah eliminasinya dapat dijabarkan sebagai berikut:

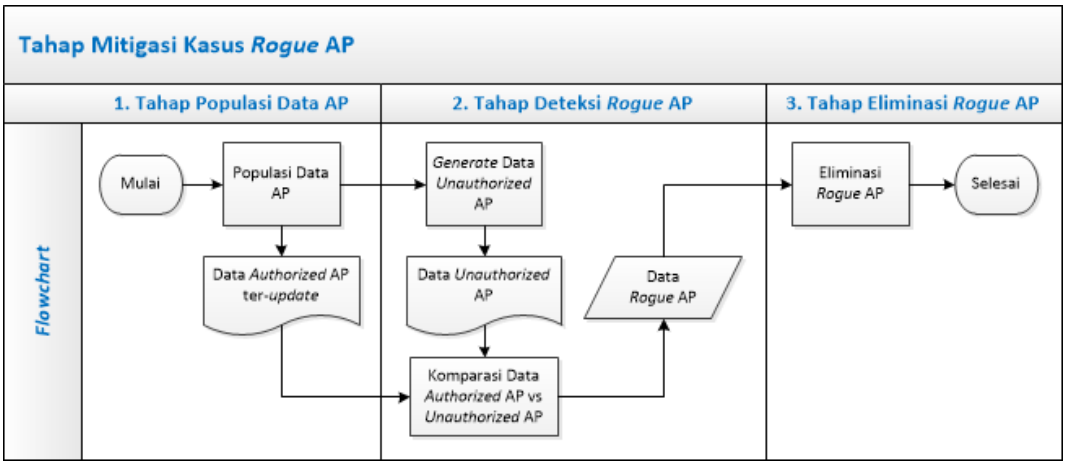
1. Pemeriksaan apakah ada *client* yang terkoneksi ke *rogue* AP.
   Pemeriksaan dilakukan dengan cara *capture* trafik *rogue* AP menggunakan parameter MAC *address* dan *channel*.
2. Pengamanan (*containment*) terhadap *rogue* AP.

Containment rogue AP bertujuan untuk mendisfungsikan *rogue* AP sehingga tidak dapat melayani permintaan *client* yang akan melakukan koneksi. *Containment* dilakukan melalui proses *deauthentication*, yang bertujuan:

a. Memutuskan koneksi *client* dari *rogue* AP.
b. Mengirimkan paket DoS (*Denial of Service*) untuk mendisfungsikan *rogue* AP tersebut.

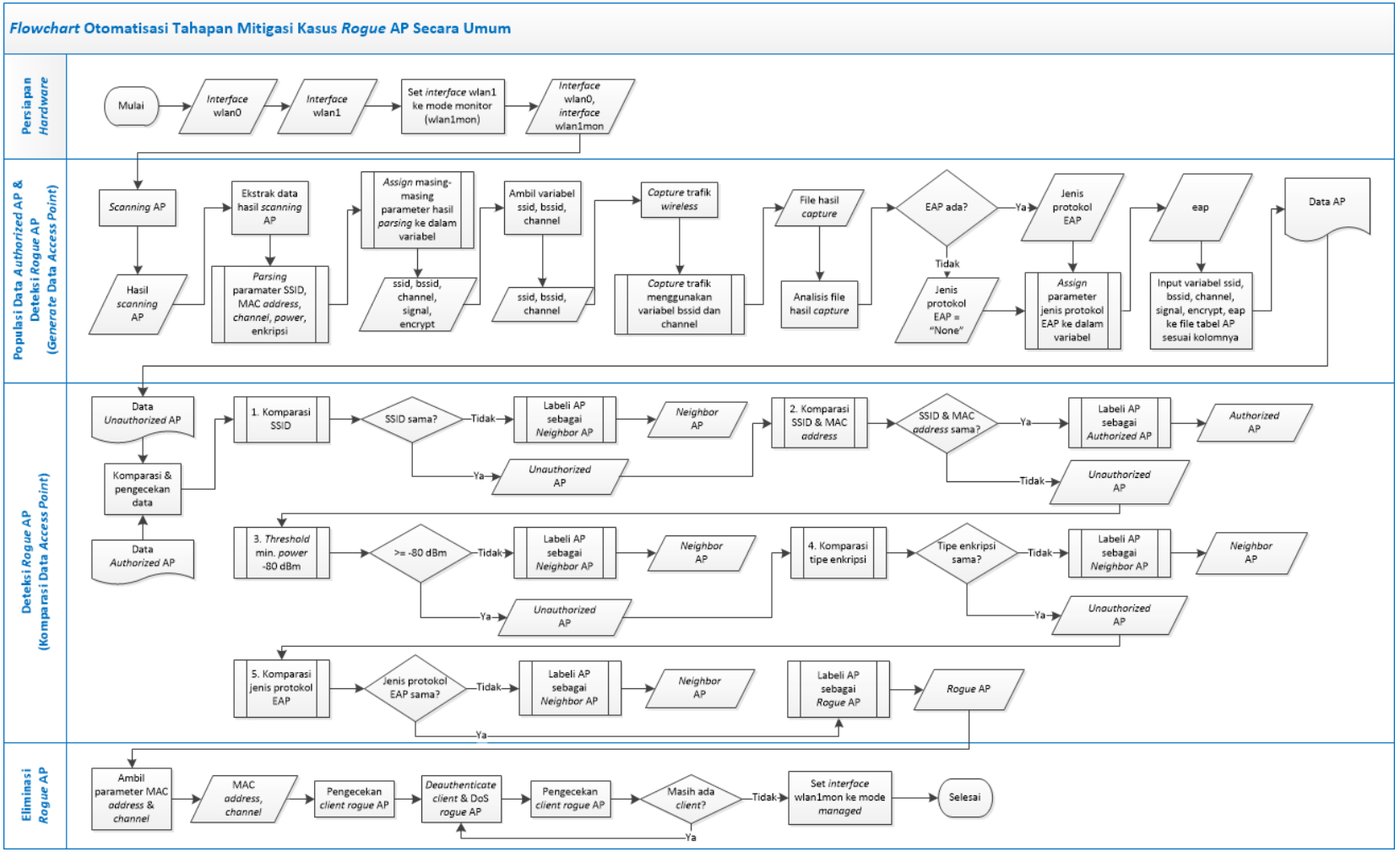
3. Pemeriksaan apakah masih ada *client* yang terkoneksi ke *rogue* AP.

Flowchart tahapan mitigasi kasus *rogue* AP secara umum dalam penelitian ini ditunjukkan pada Gambar 3.6.



Gambar 3.6 *Flowchart* tahapan mitigasi kasus *rogue* AP secara umum

Flowchart pada Gambar 3.6 tersebut dapat didetailkan lagi untuk perancangan otomatisasinya. *Flowchart* pada Gambar 3.7 menunjukkan detail perancangan otomatisasi tahapan mitigasi kasus *rogue* AP. Penjabaran berikutnya adalah algoritma otomatisasi tahapan mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X yang secara umum akan digunakan dalam penelitian ini.



Gambar 3.7 *Flowchart* otomatisasi tahapan mitigasi kasus *rogue* AP secara umum

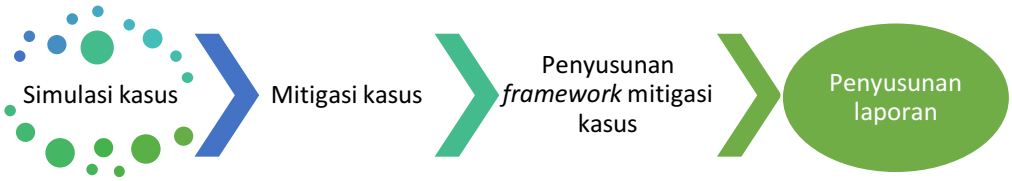
Algoritma otomatisasi tahapan mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X secara umum dapat dijabarkan sebagai berikut:

```
==========================================================================
     ALGORITMA OTOMATISASI TAHAPAN MITIGASI KASUS ROGUE AP
                  PADA WLAN IEEE 802.1X
==========================================================================
Step 1  : Start
Step 2  : Prepare hardware wlan0, wlan1.
Step 3  : If (wlan0 in Managed mode) then {do nothing};
              Else {turn wlan0 to Managed mode}.
          If (wlan1 in Managed mode) then {turn wlan1 to Monitor Mode};
              Else {do nothing}.
Step 4  : Scan AP.
Step 5  : While (read data from AP Scanning);
              Do {extract SSID, MAC address, channel, signal strength,
              encryption type}.
Step 6  : Declare variables SSID, BSSID, CHANNEL, SIGNAL, ENCRYPT.
Step 7  : Assign each of data extracted from AP scanning to
          corresponding variables.
Step 8  : Capture wireless traffic for each SSID, MAC address, and
          channel.
Step 9  : Analyze wireless traffic capture file for EAP type data.
Step 10 : Declare variable EAP.
Step 11 : Assign EAP type data to variable EAP.
          If (EAP type data found) then {EAP ← EAP type};
              Else {EAP ← "None"}.
Step 12 : Record the data of variables SSID, BSSID, CHANNEL, SIGNAL,
          ENCRYPT, EAP to corresponding column in a DB table.
Step 13 : While (read rows in DB table);
              Do {read variables in each column}.
Step 14 : While (read rows in Authorized DB table);
              Do {read variables in each column}.
Step 15 : Compare variables in DB table vs variables in Authorized AP
          DB table.
          If (SSID != SSID Authorized AP) then {AP ← Neighbor AP};
              Else {AP ← Unauthorized AP}.
              Next proceed with Unauthorized AP only.
          If (BSSID = BSSID Authorized AP) then {AP ← Authorized AP};
              Else {AP ← Unauthorized AP}.
              Next proceed with Unauthorized AP only.
          If (SIGNAL <= -80 dBm) then {AP ← Neighbor AP};
              Else {AP ← Unauthorized AP}.
              Next proceed with Unauthorized AP only.
          If (ENCRYPT != ENCRYPT Authorized AP) then {AP ← Neighbor
          AP};
              Else {AP ← Unauthorized AP}.
              Next proceed with Unauthorized AP only.
          If (EAP = EAP Authorized AP) then {AP ← Rogue AP};
              Else (AP ← Neighbor AP).
Step 16 : While (read Rogue AP);
              Do {read variables BSSID and CHANNEL}.
Step 17 : Rogue AP's client checking using variables BSSID and CHANNEL.
Step 18 : Eliminate Rogue AP using variable BSSID.
Step 19 : Repeat step 17.
          If (any client presents) then {repeat step 18};
              else {done}.
Step 20 : Turn wlan1 to Managed mode.
Step 21 : Finish
```

Flowchart beserta langkah-langkah dalam algoritma otomatisasi mitigasi kasus *rogue* AP ini juga diharapkan dapat dijadikan dasar untuk menghasilkan output berupa *script* yang akan mengotomatisasi proses-proses di dalam tahap populasi data *authorized* AP, deteksi *rogue* AP, dan eliminasi *rogue* AP.

3.5 Penyusunan Laporan

Merupakan tahap pembuatan laporan terhadap hasil pengujian untuk memberikan gambaran secara menyeluruh mengenai topik penelitian ini. Laporan berisi mengenai pendahuluan, kajian pustaka, metodologi penelitian, hasil dan pembahasan, serta penutup. Kesimpulan yang diperoleh dari penelitian ini akan dimasukkan ke dalam bagian penutup dari laporan, berikut juga dengan rekomendasi atau saran untuk penelitian-penelitian selanjutnya, khususnya yang mengambil domain penelitian mengenai *rogue Access Point*. Gambar 3.8 menunjukkan tahapan penyusunan laporan dalam penelitian ini.



Gambar 3.8 Tahapan yang dilalui untuk penyusunan laporan

BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas tentang langkah-langkah detail penelitian, analisis, dan hasil yang didapatkan dari penelitian ini. Pembahasan dalam bab ini meliputi tahap studi literatur, khususnya mengenai jenis protokol EAP yang digunakan dalam penelitian, kemudian tahap persiapan sistem untuk menguji WLAN berbasis IEEE 802.1X dapat berjalan normal, tahap simulasi kasus di mana dilakukan *setup* terhadap *rogue Access Point* (*rogue* AP), tahap mitigasi kasus yang meliputi tahap populasi *authorized Access Point* (*authorized* AP), tahap deteksi *rogue* AP, dan tahap eliminasi *rogue* AP, serta tahap pengembangan *framework* mitigasi kasus dan laporan yang bertujuan mengembangkan sebuah kerangka kerja mitigasi *rogue* AP yang dapat dijalankan secara otomatis. Otomatisasi mitigasi ini diharapkan dapat meminimalisir pekerjaan yang dilakukan oleh administrator jaringan jika dihadapkan pada kasus *rogue* AP.

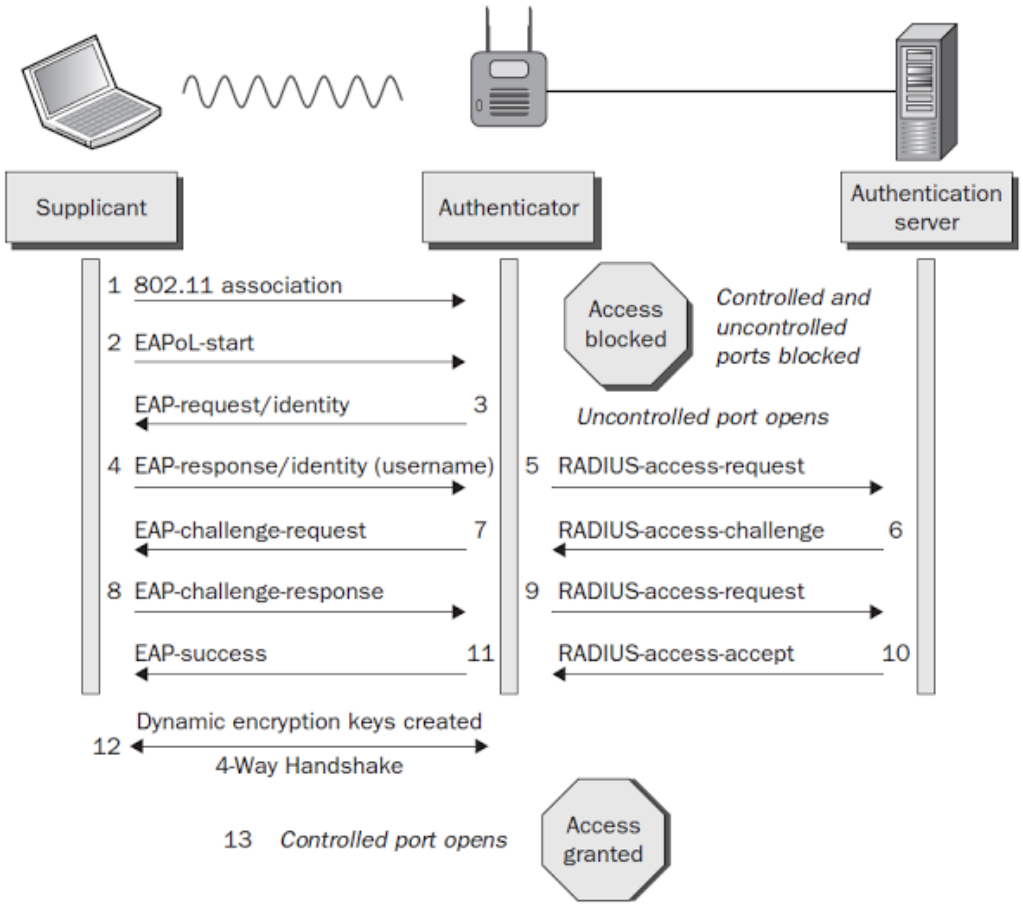
4.1 Studi Literatur Terhadap Protokol EAP

Bagian ini membahas tentang studi literatur yang dilakukan terhadap protokol EAP yang digunakan di dalam jaringan berbasis IEEE 802.1X.

4.1.1 EAP (*Extensible Authentication Protocol*)

EAP (*Extensible Authentication Protocol*) sebenarnya lebih tepat dikatakan sebagai sebuah *framework* daripada protokol. Disebut demikian karena *framework* tersebut dijadikan dasar dari berbagai jenis protokol yang diturunkan dari EAP dan protokol baru yang dapat ditambahkan ke dalamnya. EAP juga bertindak sebagai *framework* komunikasi antara *client* dan server, dan protokol yang digunakan dalam komunikasi tersebut dapat dipilih dari protokol yang tersedia di dalam *framework* EAP itu sendiri. Namun antara *client* dan server harus menggunakan protokol yang sama untuk autentikasi dan komunikasi (Kumar et al., 2014).

Sebelumnya pada bagian 2.2.1 mengenai komponen IEEE 802.1X, telah dijelaskan bahwa diperlukan tiga komponen dalam membentuk sebuah mekanisme autentikasi 802.1X pada jaringan. Komponen-komponen tersebut adalah *authentication server*, *authenticator*, dan *supplicant* (Geier, 2008). *Frame* EAP berjalan di antara ketiga komponen tersebut. Gambar 4.1 menunjukkan bagaimana proses pertukaran *frame* EAP terjadi di dalam jaringan berbasis IEEE 802.1X.



Gambar 4.1 Paket EAP di dalam jaringan berbasis IEEE 802.1X
Sumber: Coleman, Westcott, Harkins, & Jackman, 2011

Berikut adalah penjelasan dari pertukaran *frame* EAP seperti pada Gambar 4.1 pada langkah nomor 1 – 11 (Coleman et al., 2011):

1. *Supplicant* melakukan percobaan koneksi (asosiasi) ke *authenticator*.

2. *Supplicant* menginisiasi proses autentikasi EAP dengan mengirimkan *frame* 802.11 "EAPOL-Start" (EAP *over* LAN - Start). *Frame* ini merupakan *frame* yang opsional karena tidak semua jenis protokol EAP menggunakannya.
3. *Authenticator* mengirimkan *frame* "EAP-Request/Identity" yang berisi permintaan identitas *supplicant*.
4. *Supplicant* mengirimkan *frame* "EAP-Response/Identity" kepada *authenticator*, berisi *username* milik *supplicant* dalam format teks jelas (*clear text*).
5. *Authenticator* mengenkapsulasi *frame* "EAP/Response/Identity" dari *supplicant* ini ke dalam paket "RADIUS-Access-Requset" dan meneruskannya ke *authentication server*.
6. *Authentication server* melihat *username* milik *supplicant* dan memeriksanya di dalam *database*, kemudian mengirimkan *password challenge* kepada *supplicant* yang dienkapsulasi ke dalam paket "RADIUS-Access-Challenge".
7. *Authenticator* meneruskan *password challenge* tersebut kepada *supplicant* menggunakan format *frame* "EAP-Challenge-Request".
8. *Supplicant* mengambil *password challenge* tersebut dan melakukan *hashing* dengan algortima seperti MD5 atau MS-CHAPv2, kemudian mengirimkan respon *hash* di dalam *frame* "EAP-Challenge-Response" kepada *authentication server*.
9. *Authenticator* meneruskan respon *hash* tersebut kepada *authentication server* menggunakan format paket "RADIUS-Access-Request".
10. *Authentication server* menjalankan *hash* yang identik dan memeriksa apakah respon *hash* dari *supplicant* benar atau tidak. Setelah itu, *authentication server* mengirimkan pesan berhasil (paket "RADIUS-Access-Accept") jika *hash* dari *supplicant* benar, dan pesan gagal (paket "RADIUS-Access-Reject") jika *hash* dari *supplicant* salah. Namun kasus pada Gambar 4.1 *hash* dari *supplicant* benar, sehingga yang dikirimkan adalah paket "RADIUS-Access-Accept".
11. *Authenticator* meneruskan pesan "RADIUS-Access-Accept" kepada *supplicant* menggunakan format *frame* "EAP-Success". *Supplicant* berhasil diautentikasi.

4.1.2 EAP-PEAP (*Protected Extensible Authentication Protocol*)

Seperti telah dibahas pada subbab 2.2.2 mengenai protokol IEEE 802.1X, bahwa terdapat beberapa protokol yang ada di dalam *framework* EAP yang disebut juga sebagai EAP *Method*, dan umum digunakan untuk berkomunikasi di dalam jaringan berbasis IEEE 802.1X. Protokol tersebut di antaranya adalah EAP-MD5, EAP-LEAP, EAP-TLS, EAP-TTLS, dan EAP-PEAP (Kumar et al., 2014).

Penelitian ini nantinya menggunakan jenis protokol EAP-PEAP (*Protected Extensible Authentication Protocol*) juga disebut sebagai EAP di dalam EAP. PEAP menggunakan *server certificate* untuk memeriksa validitas server. Singkatnya, ketika PEAP *client* melakukan validasi terhadap server berdasarkan *server certificate*, maka terbentuklah sebuah *tunnel* yang aman. Kemudian validasi terhadap *client* dapat dimulai dengan menggunakan EAP *Method* yang mana saja (Kumar et al., 2014).

Alasan digunakannya EAP-PEAP dalam penelitian ini adalah karena EAP-PEAP hanya membutuhkan *certificate* dari sisi server (Sankar, Balinsky, Miller, & Sundaralingam, 2005). Selain itu, tidak seperti EAP-TLS, EAP-PEAP dapat menghindarkan isu terkait dengan instalasi *digital certificate* pada setiap *client* (Cisco, 2007). EAP-PEAP juga termasuk *password-based* EAP sehingga lebih mudah dalam manajemennya (Sankar et al., 2005). Dan oleh karena EAP-PEAP termasuk EAP berbasis *password*, maka lebih cocok digunakan untuk simulasi kasus dalam penelitian ini, yaitu ancaman *rogue* AP yang dapat merekam *username* dan *challenge/response* dari *password client*, untuk kemudian di-*crack* dan didapatkan *password*-nya dalam format *clear text*.

4.2 Persiapan Sistem WLAN Berbasis IEEE 802.1X

Tahap persiapan sistem ini dilakukan untuk membangun jaringan *wireless* (WLAN) berbasis IEEE 802.1X. Tujuannya adalah untuk melakukan uji coba login ke SSID menggunakan *username* dan *password* yang valid sehingga *client* dapat mengakses jaringan/internet. Hal ini dilakukan agar dapat diketahui bahwa sistem telah berjalan dengan baik dan normal.

WLAN berbasis IEEE 802.1X ini memiliki tiga komponen, yaitu:

1. *Authentication Server*

   Authentication server digunakan sebagai server autentikasi sekaligus *database* yang berisi daftar *username* dan *password client* yang akan login ke WLAN berbasis IEEE 802.1X. Server yang digunakan adalah server RADIUS (*Remote Access Dial-In User Service*).

2. *Authenticator*

   Berupa perangkat *Access Point* yang mendukung fitur 802.1X, di mana *Access Point* tersebut melemparkan autentikasi ke server RADIUS eksternal.

3. *Supplicant*

   Supplicant atau *client* berupa perangkat yang mendukung fitur *wireless*, seperti laptop, tablet, atau *smartphone*.

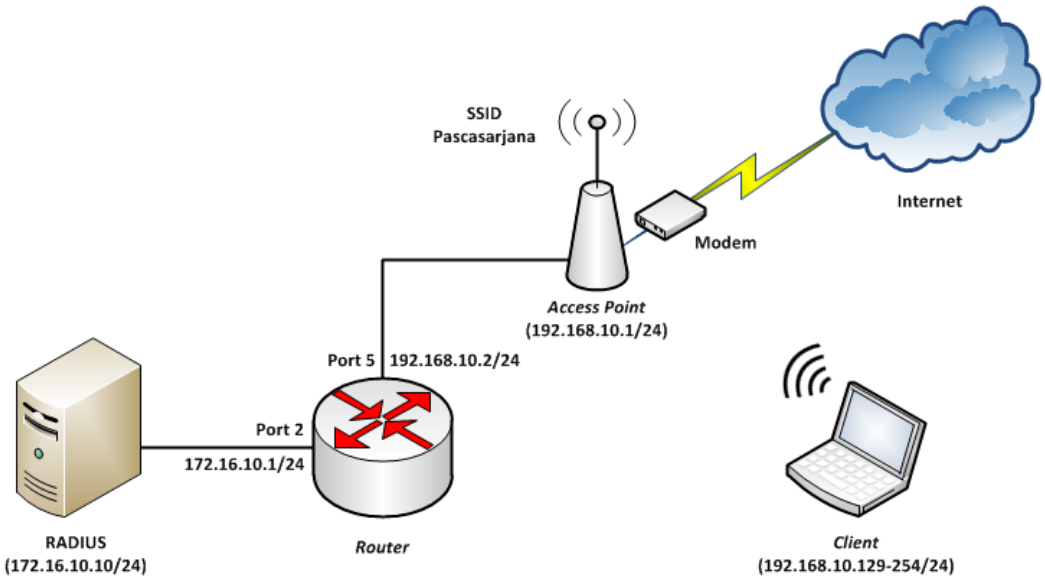
4.2.1 Spesifikasi Sistem

Spesifikasi sistem yang digunakan dalam membangun WLAN berbasis IEEE 802.1X yang akan digunakan untuk uji coba dalam penelitian ini adalah sebagai berikut:

1. Server RADIUS : Sistem operasi Linux CentOS 6.7, Freeradius, MySQL
2. *Access Point* : *Wireless* AP/*router* TP-Link TL-MR3220
3. *Client* : Tablet/*smartphone*
4. *Router* : Mikrotik RB750
5. SSID : Pascasarjana (WPA2-Enterprise)

4.2.2 Topologi Sistem

Topologi sistem WLAN berbasis IEEE 802.1X yang akan digunakan dalam uji coba penelitian ini ditunjukkan pada Gambar 4.2.



Gambar 4.2 Topologi WLAN berbasis IEEE 802.1X

Segmentasi WLAN berbasis IEEE 802.1X sesuai dengan topologi pada Gambar 4.2, ditunjukkan pada Tabel 4.1.

Tabel 4.1 Segmentasi WLAN berbasis IEEE 802.1X

Perangkat	**IP *Address***	***Netmask***	***Gateway***
Server RADIUS	172.16.10.10	/24 (255.255.255.0)	172.16.10.1
Access Point	192.168.10.1	/24 (255.255.255.0)	192.168.10.1
Client	192.168.10.129 – 254	/24 (255.255.255.0)	192.168.10.1
Router	172.16.10.1	/24 (255.255.255.0)	172.16.10.1
	192.168.10.2	/24 (255.255.255.0)	192.168.10.1

4.2.3 Langkah Persiapan Sistem

Langkah-langkah yang dilakukan dalam membangun WLAN berbasis IEEE 802.1X dalam penelitian ini adalah sebagai berikut:

1. *Install* dan konfigurasi server RADIUS
   a. Freeradius

Menangani proses autentikasi *username* dan *password* pengguna jaringan *wireless*.

b. MySQL

Sebagai *database* tempat penyimpanan *username* dan *password* pengguna jaringan *wireless*.

2. Konfigurasi *Access Point*

Dilakukan agar *Access Point* dapat mem-*broadcast* SSID yang berbasis IEEE 802.1X, sehingga mengharuskan *client* mengisi *username* dan *password*-nya untuk login ke jaringan *wireless*.

a. SSID : Pascasarjana

b. Enkripsi : WPA2-Enterprise

c. DHCP

- IP *address* : 192.168.10.129 – 192.168.10.254
- *Netmask* : /24 (255.255.255.0)

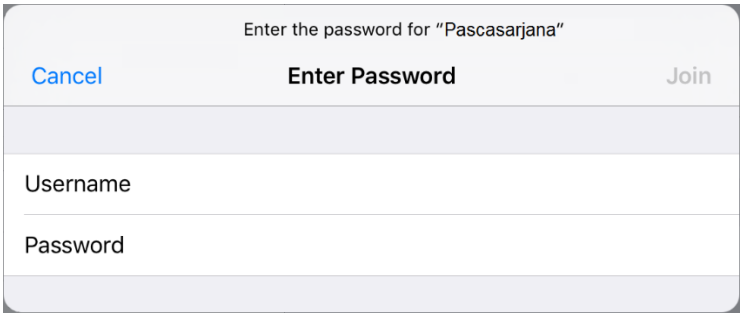
3. Konfigurasi *router*

Dilakukan untuk membedakan segmen IP *address* server dan *client*.

4.2.4 Langkah Uji Coba Sistem

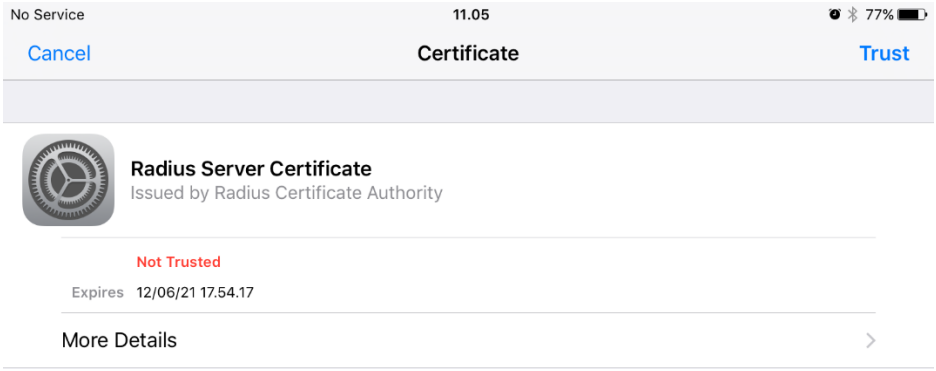
Uji coba WLAN berbasis IEEE 802.1X dilakukan dengan langkah-langkah sebagai berikut:

1. *Client* mencoba melakukan koneksi ke SSID Pascasarjana.
2. Form login (*username* dan *password*) akan muncul seperti tampak pada Gambar 4.3.



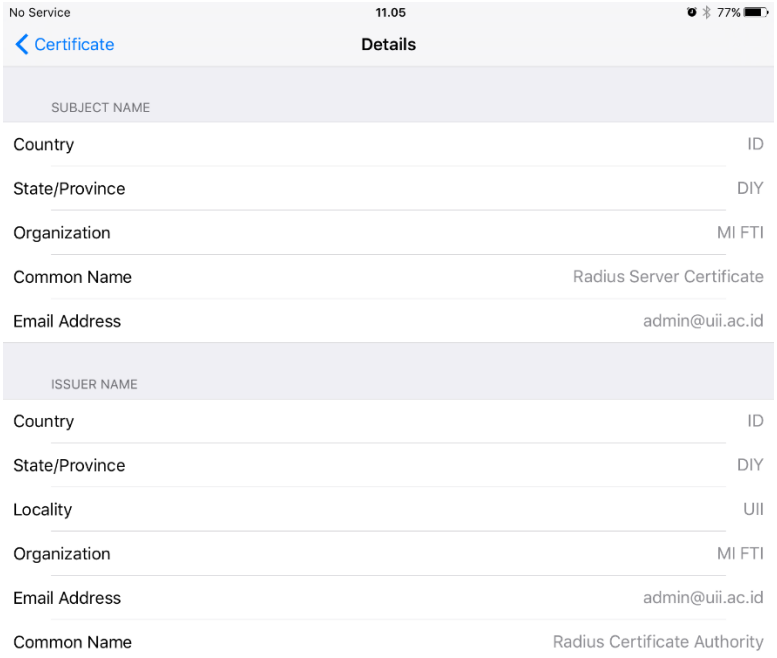
Gambar 4.3 Form login SSID Pascasarjana

3. *Client* login dengan mengisikan *username* dan *password*-nya.
4. Muncul *window* yang menunjukkan *server certificate* yang dapat diperiksa oleh *client* pada bagian "More Details" seperti ditunjukkan pada Gambar 4.4.



Gambar 4.4 *Radius Server Certificate*

5. Tampak detail *server certificate* seperti yang ditunjukkan pada Gambar 4.5.



Gambar 4.5 Detail dari *Radius Server Certificate*

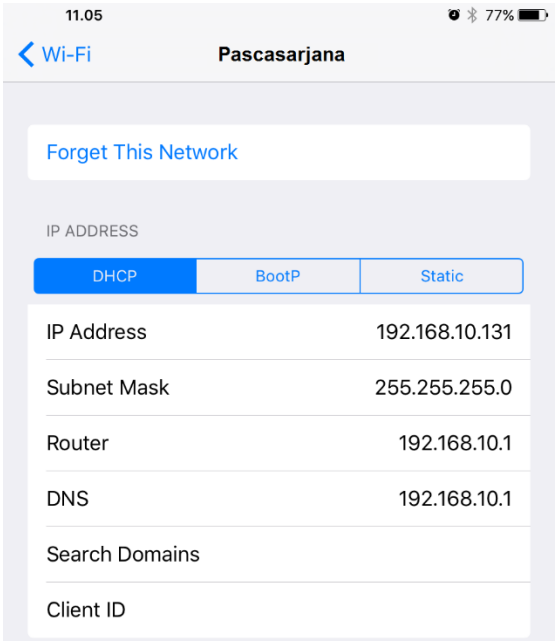
6. Kembali ke Gambar 4.4, *client* dapat memilih opsi "Trust" pada bagian kanan atas. Dan oleh karena *client* mengisikan *username* dan *password* yang valid, maka *client* berhasil login ke SSID Pascasarjana dan mendapatkan:

a. IP *address* : di antara 192.168.10.129 – 192.168.10.254

b. *Netmask* : /24 (255.255.255.0)

c. *Gateway* : 192.168.10.1

Seperti yang ditunjukkan pada Gambar 4.6.



Gambar 4.6 Detail IP *address* yang didapatkan oleh *client*

7. Setelah mendapatkan IP *address*, *client* dapat mengakses jaringan atau internet.

Berdasarkan langkah-langkah uji coba sistem, *client* telah mendapatkan IP *address* sesuai dengan segmennya, yaitu berada di jaringan 192.168.10.0/24. Pada tahap ini dapat dikatakan bahwa sistem WLAN berbasis 802.1X yang digunakan di dalam penelitian ini telah berjalan dengan baik dan normal.

4.3 Simulasi Kasus *Rogue Access Point*

Simulasi kasus yang akan dijalankan adalah dengan melakukan *setup* sebuah *rogue Access Point* (*rogue* AP) yang berbasis *software* atau disebut juga sebagai *soft* AP. Walaupun berbasis *software*, pada kenyataannya tetap diperlukan *hardware* berupa *wireless card* atau *wireless adapter* untuk melakukan *broadcast* SSID maupun memonitor trafik *wireless* yang lewat.

4.3.1 Spesifikasi *Rogue Access Point*

Spesifikasi *rogue Access Point* (*rogue* AP) yang digunakan untuk simulasi kasus dalam penelitian ini adalah sebagai berikut:

1. Tipe *rogue* AP : S*oft* AP
2. Sistem operasi : Kali Linux 2016.1 - Rolling *version* (*Virtual Machine*)
3. *Wireless adapter* : *Wireless* USB TP-Link TL-WN722N
4. *Software*
   a. Hostapd dengan *patch* Hostapd-WPE
      Digunakan untuk konfigurasi dan *broadcast* SSID *soft* AP yang mendukung fitur 802.1X.
   b. Freeradius-server dengan *patch* Freeradius-WPE
      Berfungsi sebagai server RADIUS untuk:
      - Menangkap *username client* sekaligus *challenge*/*response*-nya, yang nantinya di-*crack* dan diketahui *password client* dalam format *clear text*.
      - Menyimpan *username* dan *password client*.
      - Melakukan proses autentikasi *client* yang terkoneksi ke *soft* AP.

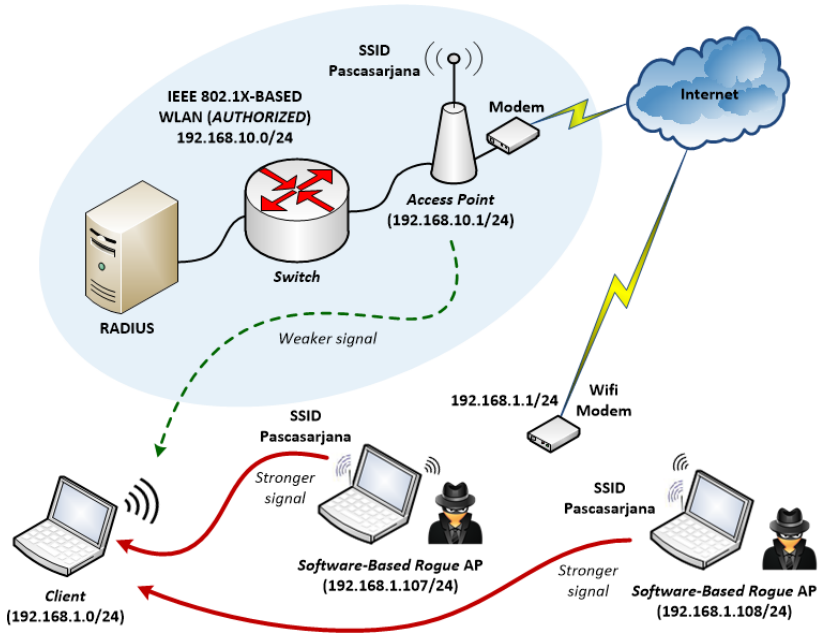
4.3.2 Skenario Simulasi

Konfigurasi IP *address* pada simulasi kasus *rogue* AP berbasis *software* (*soft* AP) ditunjukkan pada Tabel 4.2.

Tabel 4.2 Konfigurasi IP *address* pada simulasi kasus *rogue* AP

Perangkat	IP *Address*	*Netmask*	*Gateway*
3G Modem	192.168.1.1	/24 (255.255.255.0)	-
Soft AP	192.168.1.107	/24 (255.255.255.0)	192.168.1.1
Soft AP	192.168.1.108	/24 (255.255.255.0)	192.168.1.1
Client	192.168.1.2 – 254	/24 (255.255.255.0)	192.168.1.1

Topologi dan skenario simulasi *rogue* AP berbasis *software* (*soft* AP) pada WLAN berbasis IEEE 802.1X ditunjukkan pada Gambar 4.7.



Gambar 4.7 Topologi dan skenario simulasi *rogue* AP

4.3.3 Langkah-langkah Simulasi

Langkah-langkah simulasi *rogue* AP berbasis *software* (*soft* AP) pada WLAN berbasis IEEE 802.1X dilakukan sebagai berikut:

1. *Install wireless adapter* agar terbaca sebagai *wireless interface* (wlan0) pada sistem operasi yang digunakan untuk menjalankan *rogue* AP.
2. *Setup* dan konfigurasi *software* Hostapd-WPE untuk *soft* AP.

```
root@blackhat: /home/master/hostapd-2.2/hostapd
root@blackhat:/home/master/hostapd-2.2/hostapd# cat rap.conf
interface=wlan0
driver=nl80211
bridge=br0
ssid=Pascasarjana
country_code=ID
logger_stdout=-1
logger_stdout_level=0
#dump_file=/tmp/hostapd.dump
ieee8021x=1
eapol_key_index_workaround=0
own_ip_addr=127.0.0.1
auth_server_addr=127.0.0.1
auth_server_port=1812
auth_server_shared_secret=testing123
auth_algs=3
wpa=2
wpa_key_mgmt=WPA-EAP
channel=1
wpa_pairwise=CCMP
rsn_pairwise=CCMP
root@blackhat:/home/master/hostapd-2.2/hostapd#
```

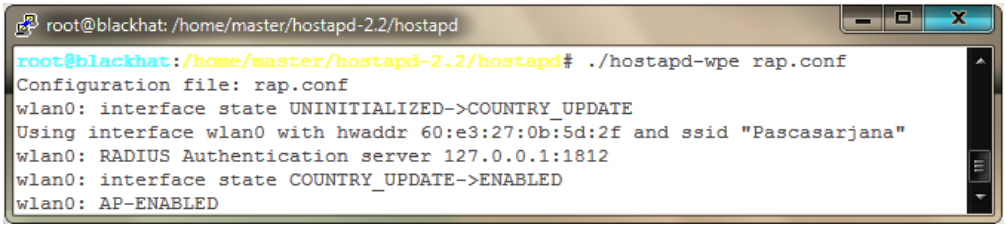
Gambar 4.8 File konfigurasi *soft* AP

3. *Setup* dan konfigurasi *software* Freeradius-WPE untuk server RADIUS.
4. Menjalankan servis RADIUS. Servis RADIUS dapat dikatakan berjalan dengan baik dan tidak ada *error* jika pada log RADIUS telah muncul informasi "Ready to process requests." seperti tampak pada Gambar 4.9.



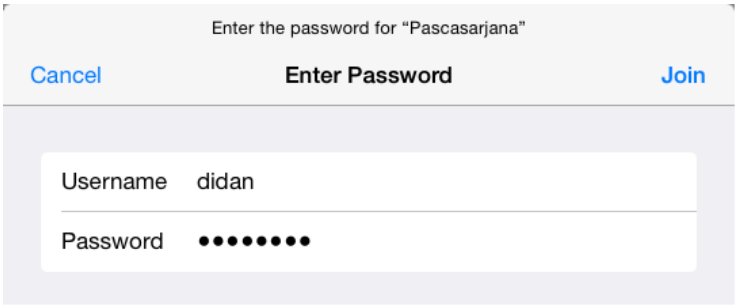
Gambar 4.9 Servis RADIUS sedang berjalan

5. Menjalankan *soft* AP. *Soft* AP yang aktif telah dapat melakukan *broadcast* SSID seperti tampak pada Gambar 4.10.



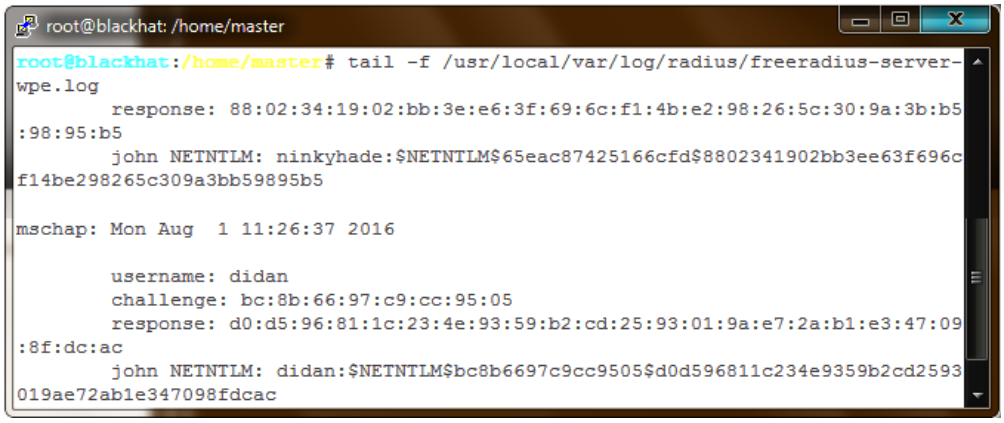
Gambar 4.10 *Soft* AP telah aktif

6. *Client* melakukan percobaan koneksi ke *soft* AP dan mengisikan *username* dan *password*-nya. Gambar 4.11 menunjukkan form login SSID *soft* AP yang telah diisi informasi *credential* milik *client*, yaitu:
   a. *Username* : didan
   b. *Password* : password



Gambar 4.11 *Client* mengisikan *username* dan *password*

7. Server RADIUS menangkap *username* beserta *challenge*/*response* dari *password client* seperti tampak pada Gambar 4.12.



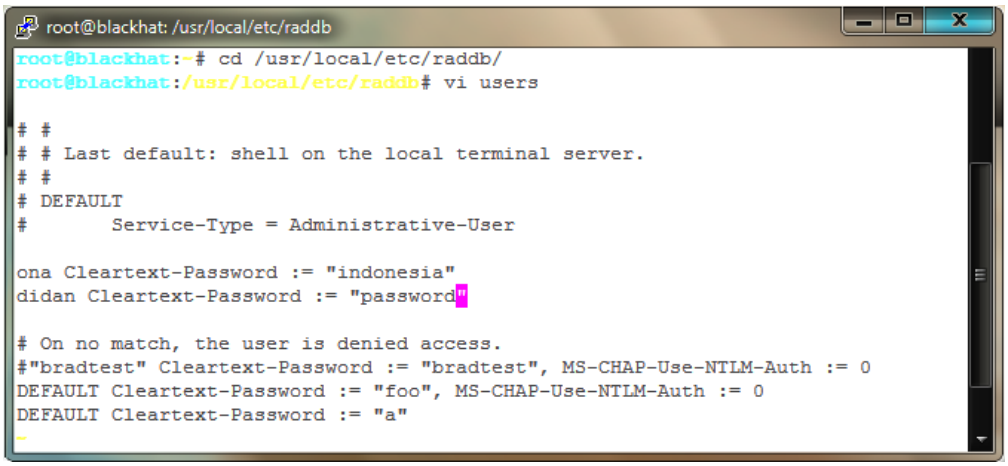
Gambar 4.12 *Username* dan *challenge*/*response* dari *password client*

8. *Challenge*/*response* di-*crack* dengan program *password cracking* seperti Asleap untuk mendapatkan *clear text password* seperti pada Gambar 4.13.



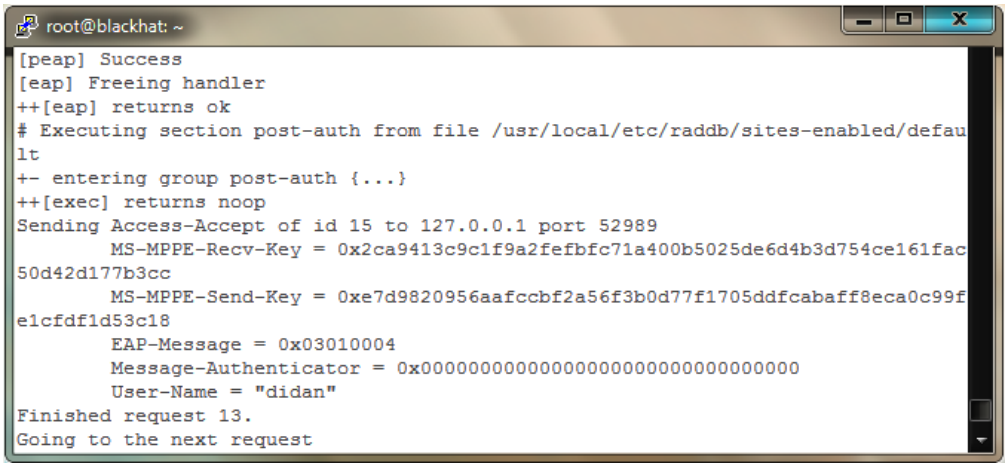
Gambar 4.13 *Username* dan *challenge*/*response* dari *password client*

9. Setelah *password* didapatkan, *username* dan *password client* kemudian didaftarkan ke server RADIUS di dalam file "users" seperti yang ditunjukkan pada Gambar 4.14.



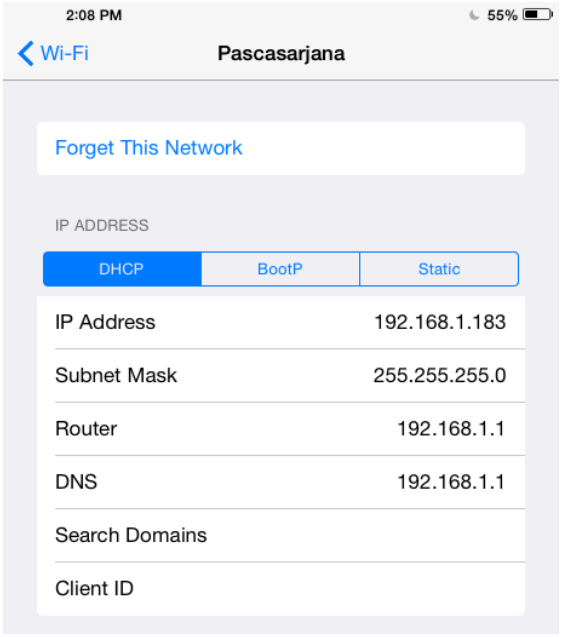
Gambar 4.14 *Username* dan *password client* didaftarkan di RADIUS

10. *Restart* servis RADIUS agar dapat mengautentikasi *username* dan *password* yang baru didaftarkan tersebut.
11. Jika *client* kembali melakukan percobaan koneksi ke *soft* AP, maka *username* dan *password client* tersebut akan diautentikasi oleh server RADIUS seperti tampak pada Gambar 4.15.



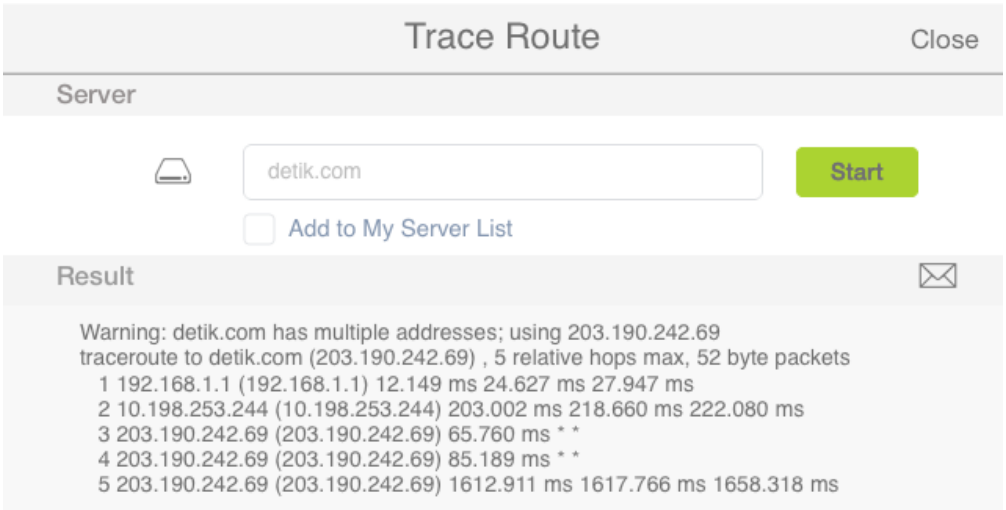
Gambar 4.15 Proses autentikasi oleh server RADIUS

12. *Client* mendapatkan IP *address* seperti yang ditunjukkan pada Gambar 4.16.



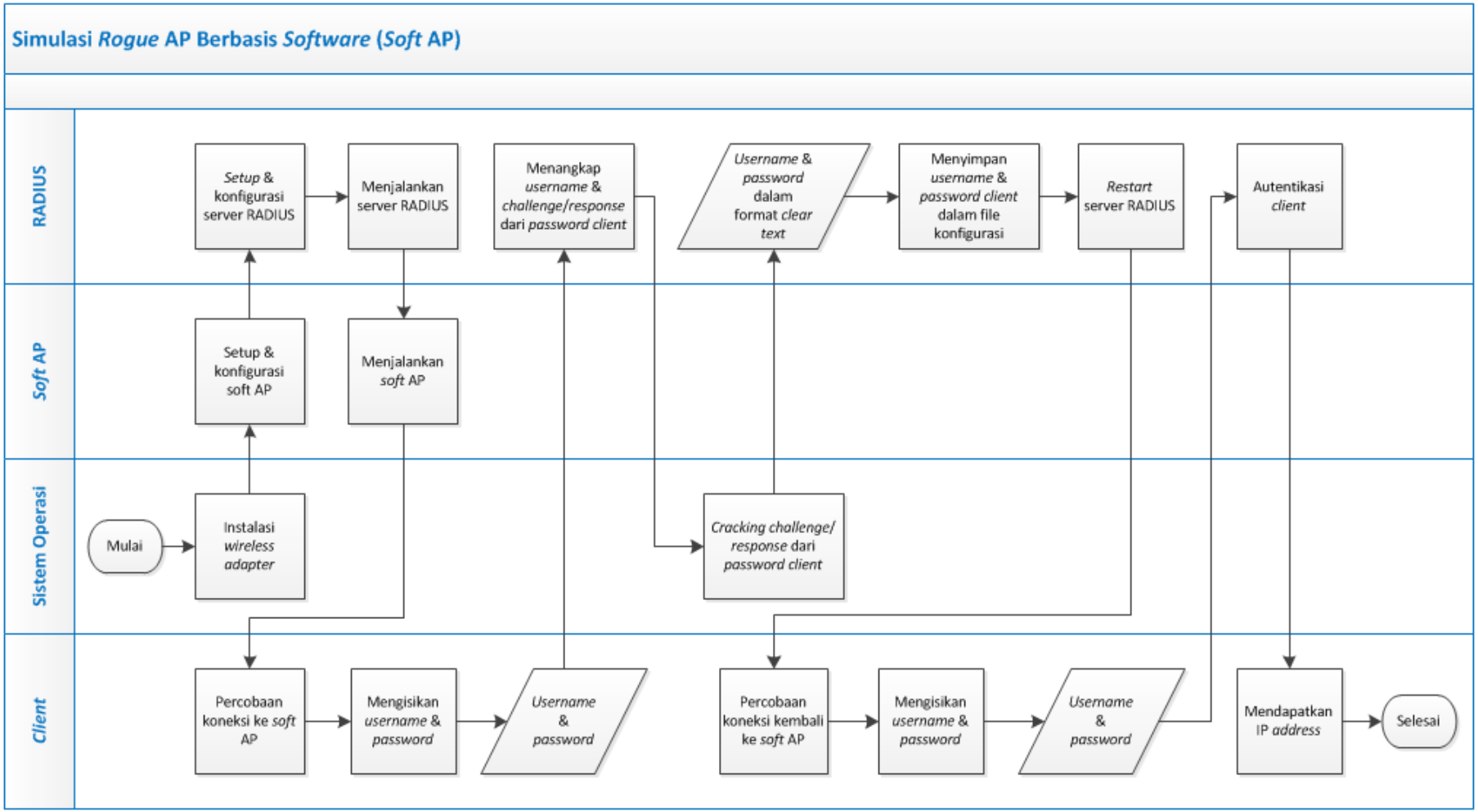
Gambar 4.16 IP *address client* dari hasil koneksi ke *soft* AP

13. Setelah mendapatkan IP *address*, *client* dapat mengakses jaringan atau internet. Misalnya, *client* melakukan proses "traceroute" ke salah satu situs web agar dapat mengetahui rute yang dilalui untuk mencapai situs web tersebut, seperti yang ditunjukkan pada Gambar 4.17.



Gambar 4.17 *Client* melakukan proses "traceroute" ke situs web

Langkah-langkah simulasi *rogue* AP menggunakan *soft* AP pada WLAN berbasis IEEE 802.1X ditunjukkan oleh *flowchart* pada Gambar 4.18.



Gambar 4.18 *Flowchart* simulasi *rogue* AP pada WLAN berbasis 802.1X

4.4 Mitigasi Kasus *Rogue Access Point* dan Pengembangan *Framework*

Merupakan tahap dilakukannya populasi data *authorized Access Point* (*authorized* AP), *live forensics* untuk deteksi *rogue Access Point* (*rogue* AP) pada WLAN berbasis IEEE 802.1X, dan dilanjutkan dengan eliminasi *rogue* AP. Dari langkah mitigasi kasus *rogue* AP tersebut, dikembangkanlah *framework* mitigasi kasus. *Framework* ini akan dijadikan dasar untuk mengimplementasikan otomatisasi langkah mitigasi *rogue* AP pada WLAN berbasis IEEE 802.1X.

Perlu ditekankan bahwa *framework* otomatisasi mitigasi kasus *rogue* AP ini digunakan sebagai dasar untuk membuat *script* yang dapat meminimalisir tugas-tugas manual yang dikerjakan administrator jaringan. Jadi, tidak semua tugas administrator jaringan dikerjakan secara otomatis. Selain menawarkan *framework* otomatisasi mitigasi kasus *rogue* AP, penelitian ini terlebih dahulu akan membahas mengenai mitigasi kasus *rogue* AP yang dilakukan secara manual. Tujuannya adalah untuk mengetahui perbedaan di antara *framework* mitigasi kasus *rogue* AP secara manual dan terotomatisasi.

4.4.1 Spesifikasi dan Persiapan Sistem Mitigasi

Spesifikasi sistem mitigasi yang digunakan untuk mendeteksi dan mengeliminasi *rogue Access Point* (*rogue* AP) pada WLAN berbasis IEEE 802.1X dalam penelitian ini adalah sebagai berikut:

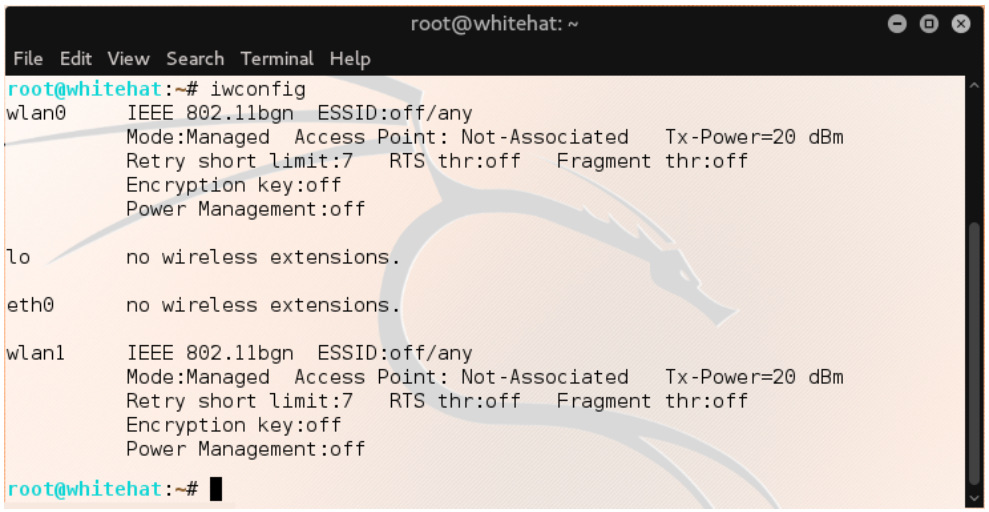
1. Sistem operasi : Kali Linux 2016.1 - Rolling *version* (*Virtual Machine*)
2. *Wireless adapter* : 2 unit *Wireless* USB TP-Link TL-WN722N
   a. *Wireless adapter* 1

      Digunakan sebagai *interface* wlan0, yaitu *interface* yang digunakan untuk melakukan *scanning Access Point*.
   b. *Wireless adapter* 2

      Digunakan sebagai *interface* wlan1mon, yaitu *interface* mode monitor yang digunakan untuk melakukan *capture* trafik *wireless* di mana paket EAP akan dilewatkan.

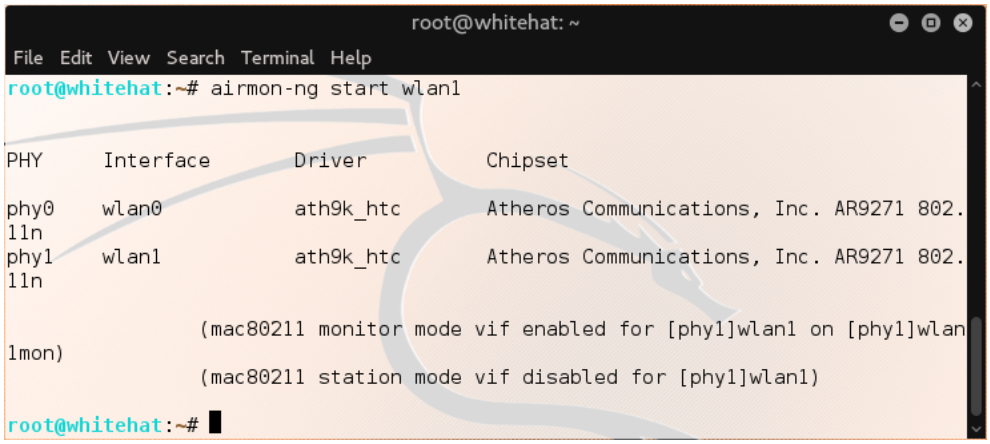
Persiapan sistem mitigasi dilakukan dengan cara mempersiapkan *hardware* yang digunakan agar dapat terbaca sebagai *wireless interface* pada sistem operasi dan *setting wireless interface* tersebut sesuai dengan kebutuhan, yaitu *interface* wlan0 dan wlan1mon. Langkah-langkah persiapan *hardware* akan dijabarkan sebagai berikut:

1. **Persiapan *hardware* secara manual**
   a. Administrator melakukan instalasi *driver wireless adapter* pada sistem operasi Kali Linux.
   b. Administrator mengkoneksikan *wireless adapter* 1 ke port USB laptop.
   c. Sistem akan membaca *wireless adapter* 1 sebagai *interface* wlan0 seperti pada Gambar 4.19.
   d. Administrator mengkoneksikan *wireless adapter* 2 ke port USB laptop.
   e. Sistem akan membaca *wireless adapter* 2 sebagai *interface* wlan1 seperti pada Gambar 4.19.



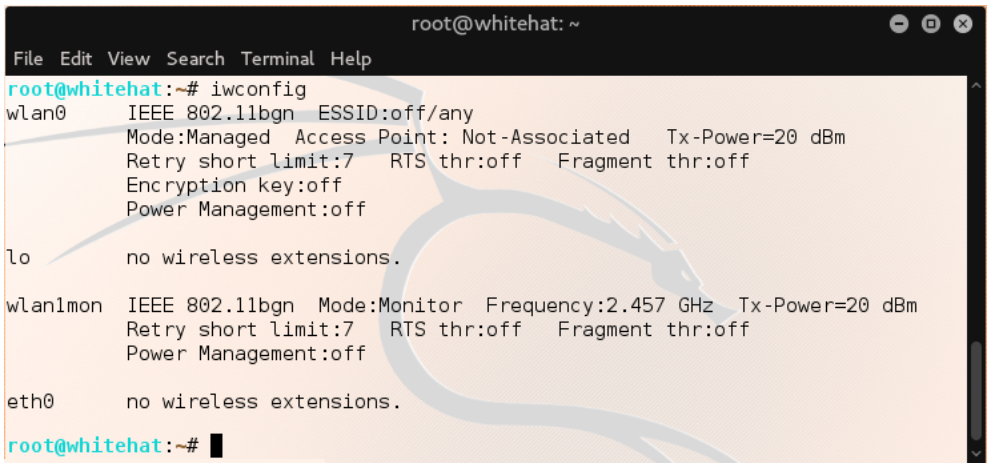
Gambar 4.19 *Wireless interface* wlan0 dan wlan1

   f. Administrator melakukan *setting interface* wlan1 ke mode monitor dengan menggunakan *tool* Airmon-ng seperti pada Gambar 4.20. *Tool* ini mengubah mode bawaan (*default*) wlan1 yaitu mode *Managed* ke mode monitor, yang berfungsi untuk melakukan *capture* trafik *wireless* tanpa harus terkoneksi ke *Access Point* tertentu.



Gambar 4.20 Set *interface* wlan1 ke mode monitor

g. Sistem memiliki *interface* wlan0 dan wlan1mon seperti pada Gambar 4.21.



Gambar 4.21 *Wireless interface* wlan0 dan wlan1mon

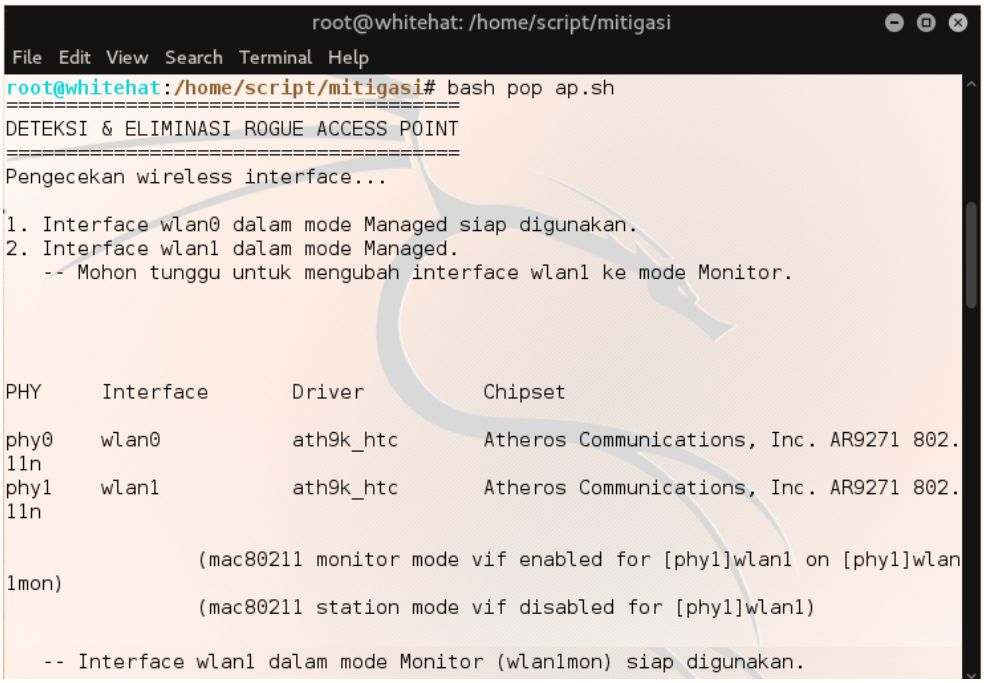
Flowchart persiapan *hardware* secara manual ini ditunjukkan pada Gambar 4.23.

2. Otomatisasi persiapan *hardware*

Algoritma otomatisasi persiapan *hardware* dapat dijabarkan sebagai berikut:

```
Step 1 : Start
Step 2 : Prepare hardware wlan0, wlan1.
Step 3 : If (wlan0 in Managed mode) then {do nothing};
              else {turn wlan0 to Managed mode}.
         If (wlan1 in Managed mode) then {turn wlan1 to Monitor
         Mode};
              else {do nothing}.
Step 4 : Finish
```

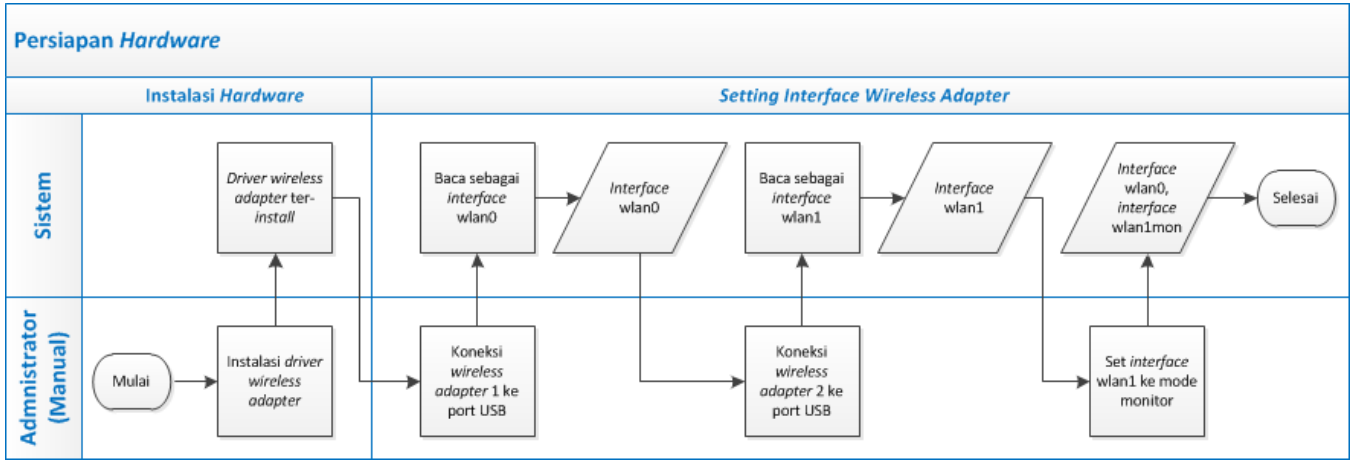
a. Administrator melakukan instalasi *driver wireless adapter* pada sistem operasi Kali Linux.
b. Administrator mengkoneksikan *wireless adapter* 1 ke port USB laptop.
c. Sistem akan membaca *wireless adapter* 1 sebagai *interface* wlan0.
d. Administrator mengkoneksikan *wireless adapter* 2 ke port USB laptop.
e. Sistem akan membaca *wireless adapter* 2 sebagai *interface* wlan1.
f. Sistem mengubah *interface* wlan1 ke *interface* wlan1mon seperti pada Gambar 4.22.



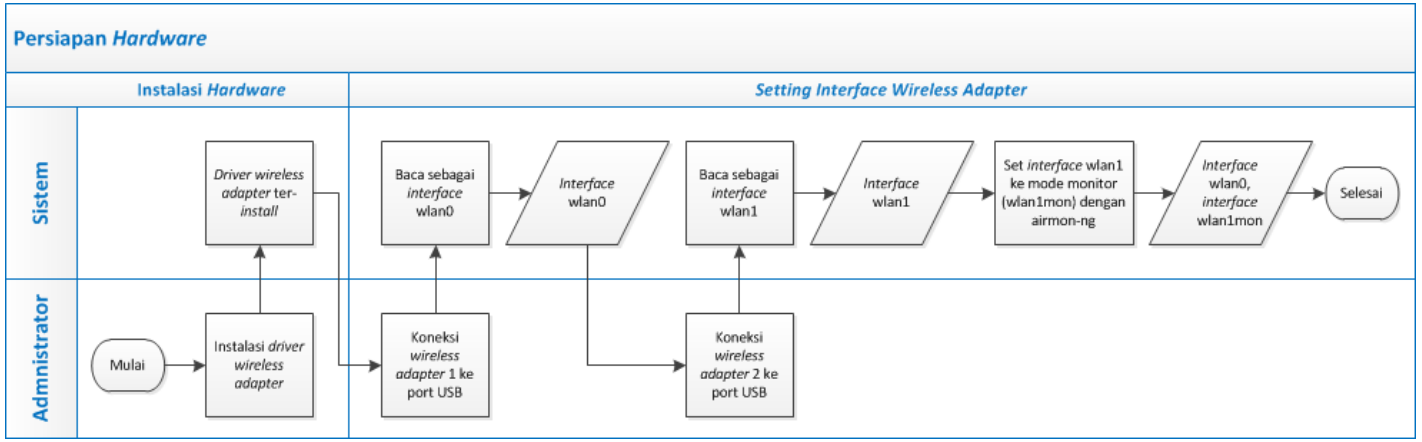
Gambar 4.22 Set *interface* wlan1 ke mode monitor oleh sistem

g. Sistem memiliki *interface* wlan0 dan wlan1mon.

Flowchart otomatisasi persiapan *hardware* ini ditunjukkan pada Gambar 4.24.

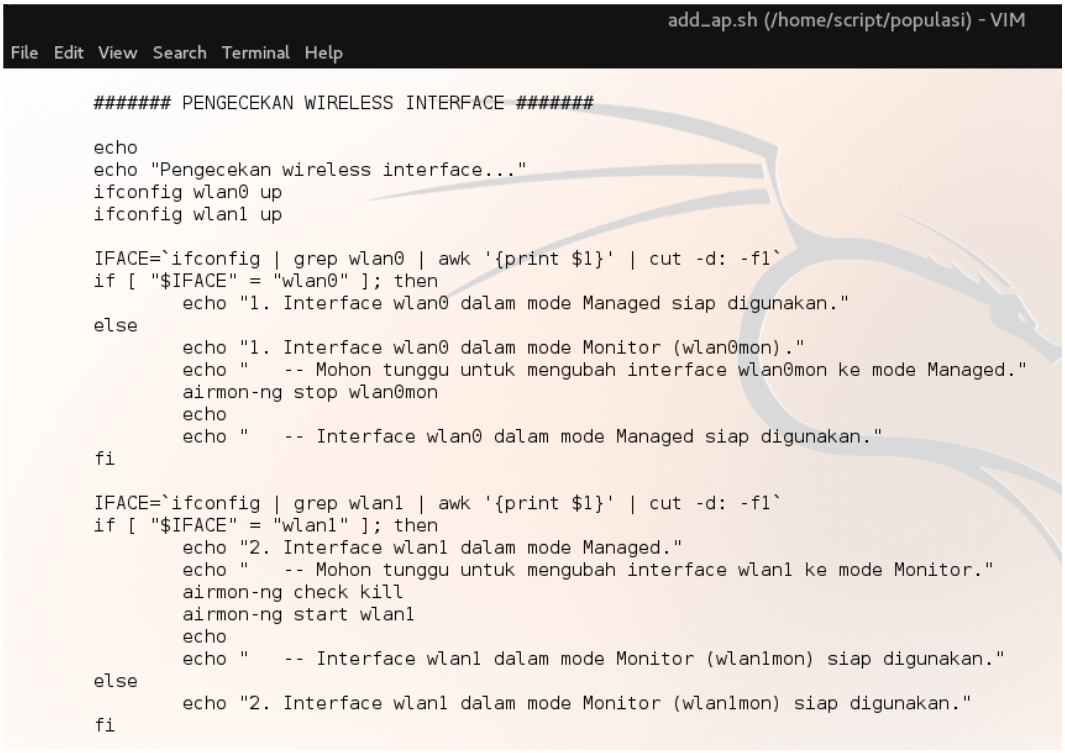


Gambar 4.23 *Flowchart* persiapan *hardware* secara manual



Gambar 4.24 *Flowchart* otomatisasi persiapan *hardware*

Berdasarkan langkah-langkah persiapan *hardware* secara manual dan terotomatisasi yang telah dijabarkan tersebut, maka dapat diperoleh kesimpulan bahwa perbedaannya terletak pada langkah *setting interface* wlan1 ke mode monitor. Jika pada cara manual perintah kepada sistem operasi untuk *setting interface* wlan1 ke mode monitor dieksekusi oleh administrator, maka pada *script* otomatisasi tugas tersebut dieksekusi sendiri oleh sistem operasi. Gambar 4.25 menunjukkan potongan *script* otomatisasi persiapan *hardware*. *Script* otomatisasi persiapan *hardware* lebih lengkap ditunjukkan pada Lampiran 1.



Gambar 4.25 Potongan *script* otomatisasi persiapan *hardware*

4.4.2 Tahap Populasi Data *Authorized Access Point*

Tahap populasi data *authorized Access Point* (*authorized* AP) mengadopsi penelitian Thakur dan Bodhe (2013) yang memiliki data *auhorized* AP untuk dibandingkan dengan data SSID dan MAC *address Access Point* dari hasil *sniffing*. Tahap populasi data *authorized* AP dalam penelitian ini juga mengadopsi penelitian Vanjale dan Mane (2014) yang salah satu tahapnya adalah tahap *learning mode*, yaitu tahap di mana daftar *authorized* AP dibuat, meliputi data SSID, MAC *address*,

dan nilai *Received Signal Strength* (RSS). Kedua penelitian ini telah dijelaskan pada bagian 1.6 mengenai *review* penelitian.

Tahap populasi data *authorized* AP dilakukan hanya jika:

1. Terdapat pemasangan *authorized* AP baru di dalam jaringan.
2. Terdapat pelepasan *authorized* AP di dalam jaringan.

Kedua hal tersebut akan berpengaruh pada data *authorized* AP, karena jika ada pemasangan *Access Point* baru di dalam jaringan, maka data *Access Point* baru tersebut harus ditambahkan ke dalam data *authorized* AP. Dan sebaliknya, jika ada pelepasan *Access Point* di dalam jaringan, maka data *Access Point* tersebut harus dihapus dari data *authorized* AP. Tahap populasi data *authorized* AP bertujuan untuk menyediakan data *authorized* AP yang ter-*update*. Langkah-langkah dalam tahap populasi data *authorized* AP akan dijabarkan sebagai berikut:

1. Populasi data *authorized* AP secara manual

a. Administrator mencari data MAC *address Access Point* yang baru dipasang atau dilepas dari jaringan di dalam file tabel *authorized* AP.
b. Jika data MAC *address* yang dimaksud ada di dalam file, maka administrator dapat memilih opsi untuk menghapus data tersebut secara manual apabila *Access Point* dilepas dari jaringan.
c. Jika data MAC *address* yang dimaksud tidak ada di dalam file, maka administrator dapat memilih opsi untuk menambah data tersebut secara manual apabila *Access Point* baru dipasang di jaringan.

Flowchart tahap populasi data *authorized* AP secara manual ini ditunjukkan pada Gambar 4.26.

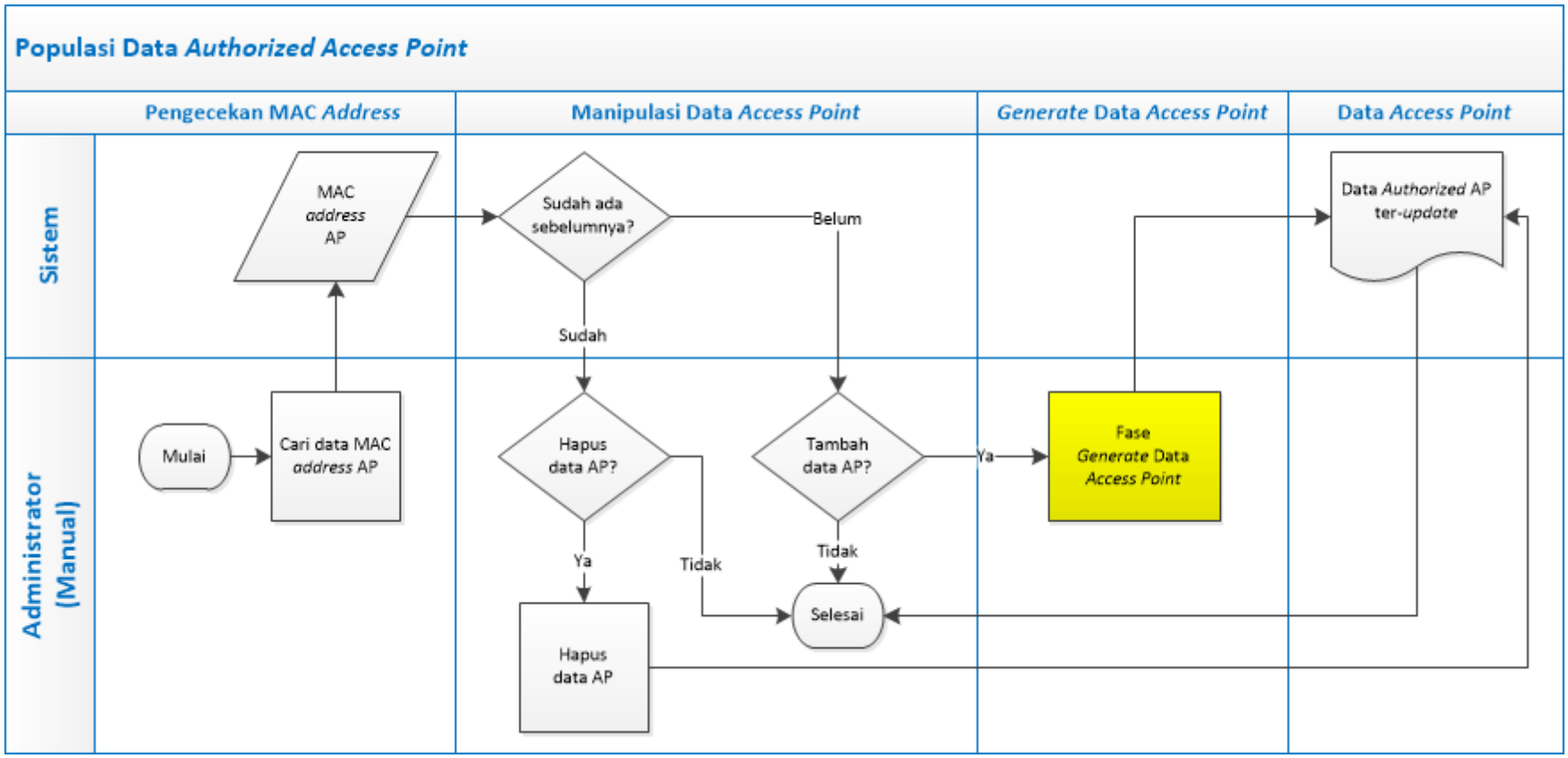
2. Otomatisasi populasi data *authorized* AP

Algoritma otomatisasi populasi data *authorized* AP dapat dijabarkan sebagai berikut:

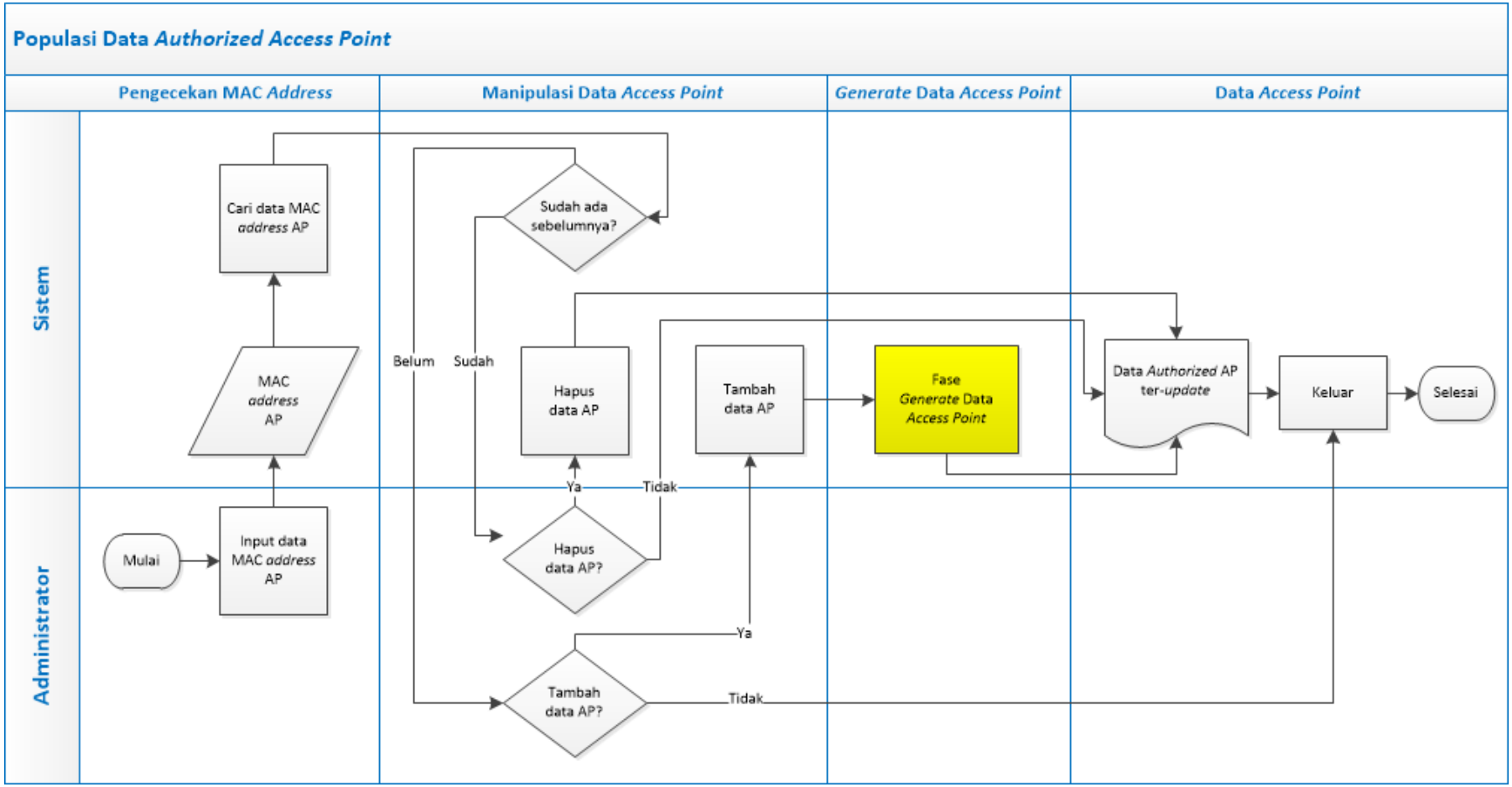
```
Step 1 : Start
Step 2 : Check AP's MAC address in authorized AP DB table.
         If (AP's MAC address exists && AP is dismantled) then
         {remove AP data};
         Else {
         If (AP's MAC address doesn't exist && AP is installed)
         then {add AP data}
         }.
Step 3 : Finish
```

a. Administrator menginputkan data MAC *address Access Point* yang baru dipasang atau dilepas dari jaringan ke dalam sistem.
b. Sistem akan melakukan pencarian terhadap MAC *address* yang diinputkan oleh administrator.
c. Jika data MAC *address* yang dimaksud ada di dalam file, maka sistem memberikan opsi untuk menghapus data *Access Point* dengan MAC *address* yang dilepas dari jaringan tersebut.
d. Jika administrator memilih untuk menghapus data, maka sistem akan mengeksekusi penghapusan data.
e. Jika data MAC *address* yang dimaksud tidak ada di dalam file, maka sistem memberikan opsi untuk menambah data *Access Point* dengan MAC *address* yang baru dipasang di jaringan tersebut.
f. Jika administrator memilih untuk menambah data, maka sistem akan mengeksekusi penambahan data.

Flowchart otomatisasi tahap populasi data *authorized* AP ini ditunjukkan pada Gambar 4.27.

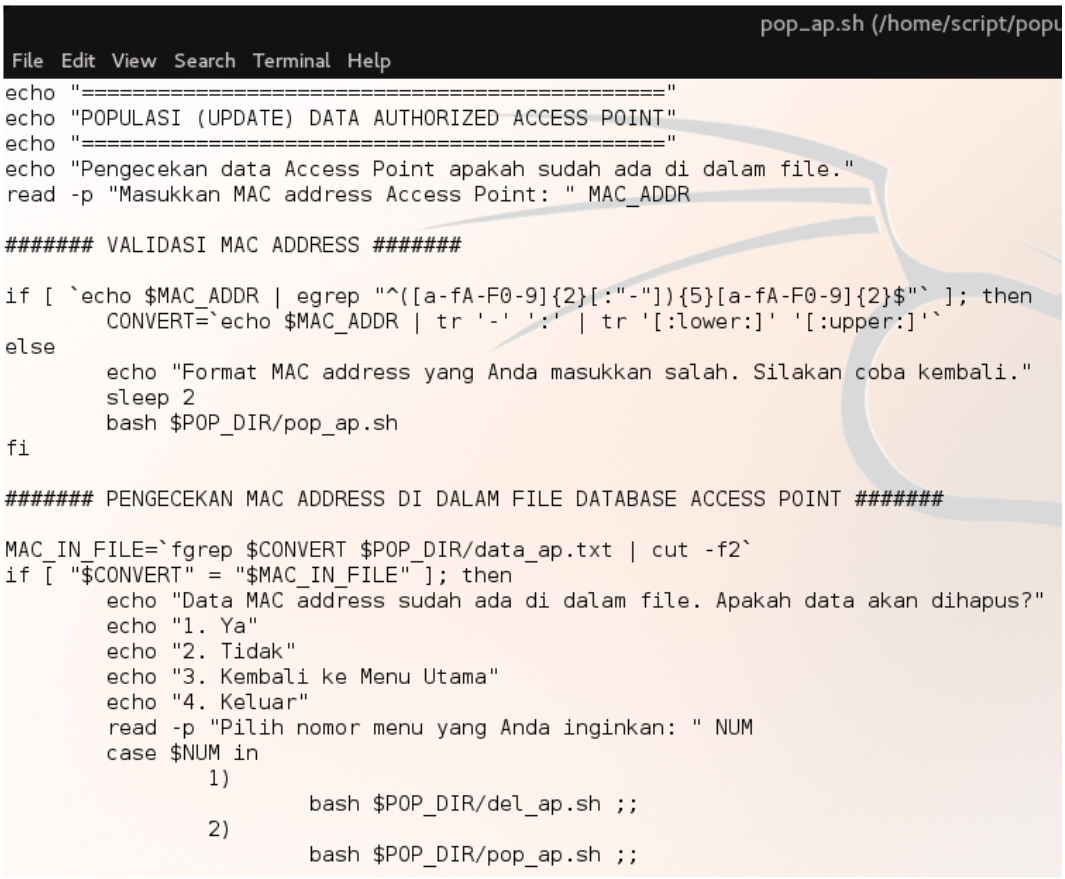


Gambar 4.26 *Flowchart* tahap populasi data *authorized* AP secara manual



Gambar 4.27 *Flowchart* otomatisasi tahap populasi data *authorized* AP

Berdasarkan tahap populasi data *authorized* AP secara manual dan terotomatisasi yang telah dijabarkan tersebut, dapat diperoleh kesimpulan bahwa perbedaannya terletak pada langkah penambahan atau penghapusan data. Jika pada cara manual perintah kepada sistem untuk menambah atau menghapus data dieksekusi oleh administrator, maka pada *script* otomatisasi tugas tersebut dieksekusi sendiri oleh sistem. Kesimpulan lain yang dapat diambil dari otomatisasi tahap populasi data *authorized* AP adalah bahwa sistem menyediakan menu interaktif yang tetap memerlukan inputan dari administrator, seperti saat administrator menginputkan data MAC *address Access Point* maupun saat administrator menginputkan pilihan terhadap opsi penambahan atau penghapusan data *Access Point*. Gambar 4.28 menunjukkan potongan *script* otomatisasi tahap populasi data *authorized* AP. *Script* otomatisasi tahap populasi data *authorized* AP lebih lengkap terdapat pada Lampiran 2.

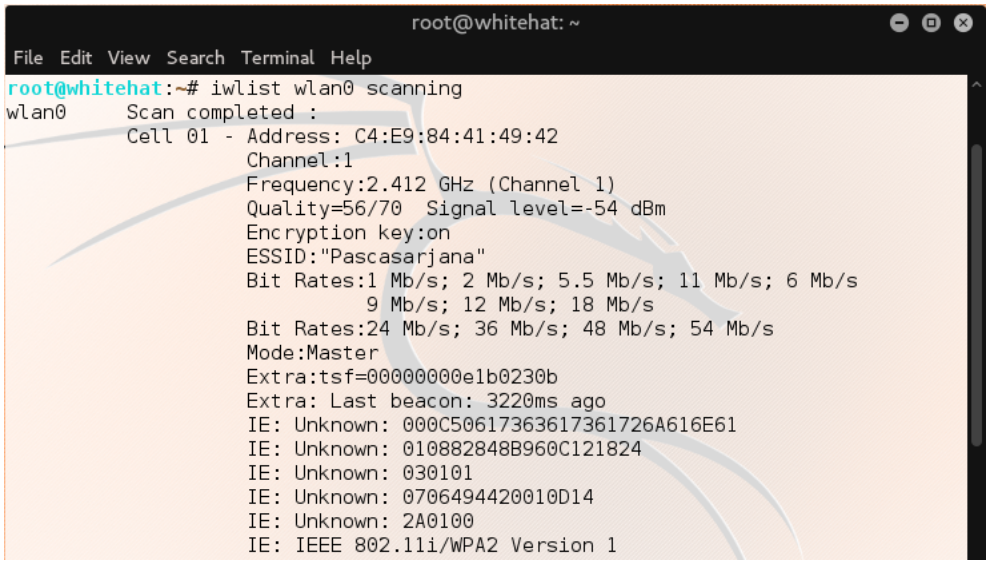


Gambar 4.28 Potongan *script* otomatisasi tahap populasi data *authorized* AP

Flowchart tahap populasi data *authorized* AP yang ditunjukkan pada Gambar 4.26 dan Gambar 4.27 memiliki fase *generate* data *Access Point*. Fase ini digunakan untuk melakukan penambahan data *Access Point* dan dapat dijabarkan sebagai berikut:

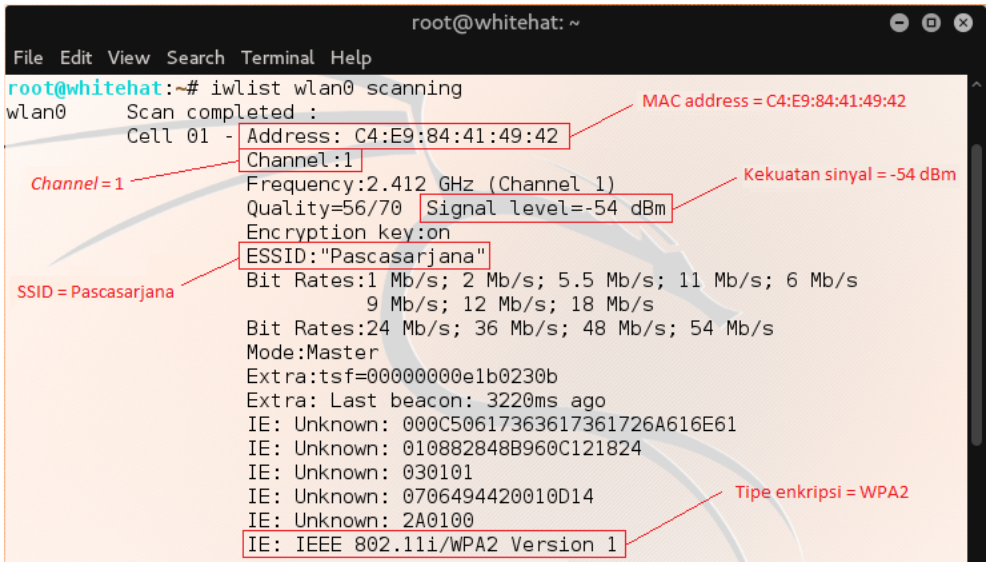
1. Fase *generate* data *Access Point* secara manual

a. Administrator mengeksekusi perintah untuk *scanning Access Point* menggunakan *tool* Iwlist pada *interface* wlan0 seperti yang ditunjukkan pada Gambar 4.29.



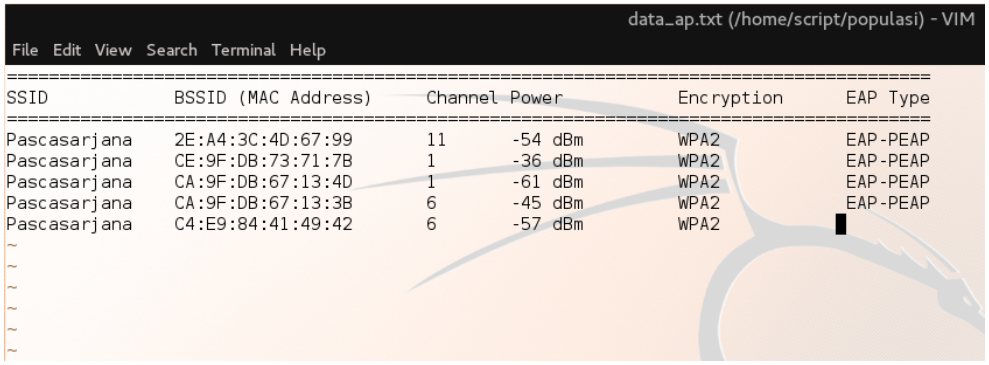
Gambar 4.29 Eksekusi perintah Iwlist untuk *scan Access Point*

b. Administrator mengekstrak data hasil *scanning Access Point* untuk mendapatkan parameter SSID, MAC *address*, *channel*, *power*, dan tipe enkripsi seperti pada Gambar 4.30.



Gambar 4.30 Ekstrak data hasil *scanning Access Point*

c. Administrator menginputkan parameter hasil ekstraksi tersebut ke dalam file tabel *authorized* AP sesuai dengan kolomnya seperti pada Gambar 4.31.



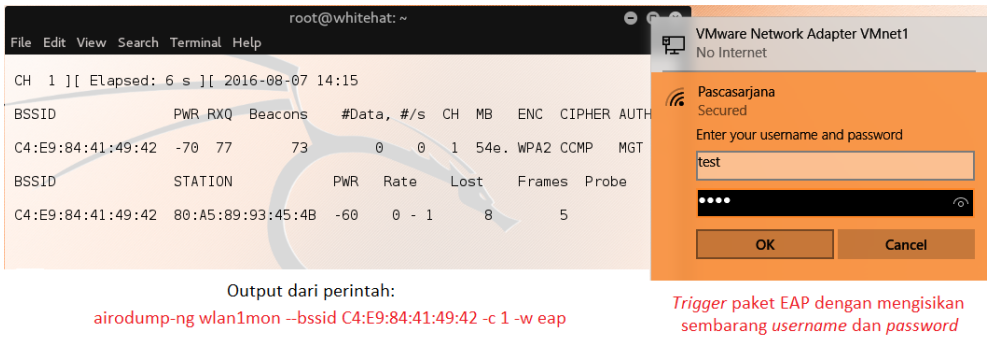
Gambar 4.31 Input parameter hasil ekstraksi ke tabel *authorized* AP

d. Administrator menggunakan data MAC *address* dan *channel* untuk mengeksekusi perintah *capture* trafik *wireless*.

e. *Capture* trafik *wireless* dilakukan menggunakan *tool* Airodump-ng pada *interface* wlan1mon yang dipasangkan dengan opsi berikut:

- Opsi --bssid = MAC *address* → C4:E9:84:41:49:42
- Opsi -c = *channel* → 1
- Opsi -w = nama file hasil *capture* → eap (ekstensi .cap)

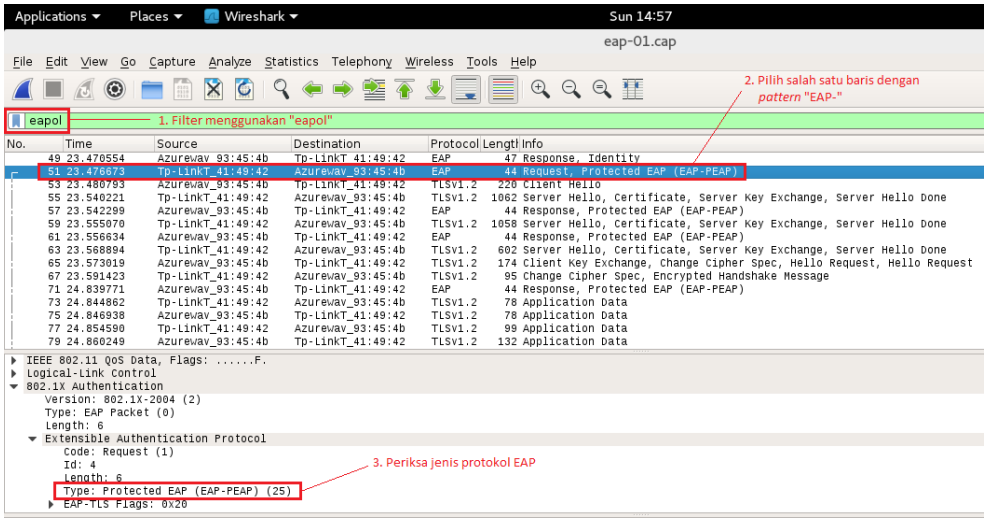
f. Saat aktivitas *capture* trafik *wireless* berlangsung, administrator melakukan percobaan koneksi ke SSID yang MAC *address*-nya sedang di-*capture* trafik. Percobaan koneksi ke SSID akan memunculkan form login di mana administrator menginputkan sembarang *username* dan *password* seperti pada Gambar 4.32 untuk memicu (*trigger*) pengiriman paket EAP.



Gambar 4.32 *Capture* trafik *wireless* untuk mendapatkan paket EAP

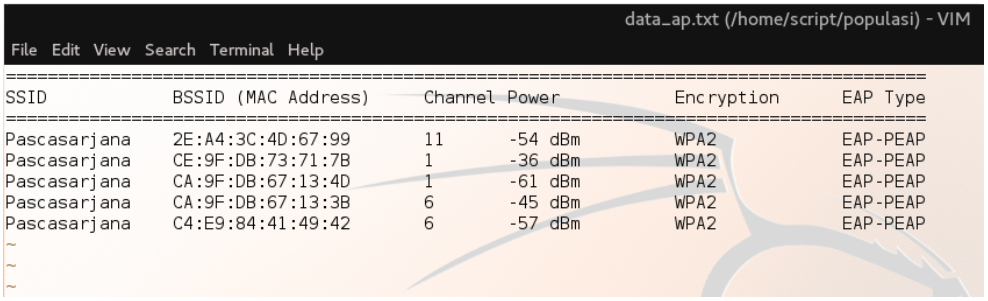
g. Administrator menghentikan aktivitas *capture* trafik *wireless*.

h. Administrator menganalisis file hasil *capture* trafik *wireless* menggunakan *tool* Wireshark untuk melihat jenis protokol EAP yang digunakan oleh SSID dengan MAC *address* yang tadi di-*capture* trafiknya seperti ditunjukkan pada Gambar 4.33. Ada atau tidak ada protokol EAP, hasilnya tetap akan diinputkan ke dalam file tabel *authorized* AP.



Gambar 4.33 Analisis file hasil *capture* trafik *wireless*

i. Administrator menginputkan data jenis protokol EAP dari hasil analisis ke dalam file tabel *authorized* AP. Jika ditemukan jenis protokol EAP-nya, maka jenis protokol EAP tersebut dituliskan pada kolom EAP *Type* seperti pada Gambar 4.34. Namun jika tidak ditemukan, maka cukup dituliskan "None".



Gambar 4.34 Input parameter jenis protokol EAP

Flowchart fase *generate* data *Access Point* secara manual ini ditunjukkan pada Gambar 4.39.

2. Otomatisasi fase *generate* data *Access Point*

Algoritma otomatisasi fase *generate* data *Access Point* dapat dijabarkan sebagai berikut:

```
Step 1  : Start
Step 2  : Scan AP.
Step 3  : While (read data from AP Scanning);
              Do {extract SSID, MAC address, channel, signal strength,
              encryption type}.
Step 4  : Declare variables SSID, BSSID, CHANNEL, SIGNAL, ENCRYPT.
Step 5  : Assign each of data extracted from AP scanning to
          corresponding variables.
Step 6  : Capture wireless traffic for each SSID, MAC address, and
          channel.
Step 7  : Analyze wireless traffic capture file for EAP type data.
Step 8  : Declare variable EAP.
Step 9  : Assign EAP type data to variable EAP.
          If (EAP type data found) then {EAP ← EAP type};
              Else {EAP ← "None"}.
Step 10 : Record the data of variables SSID, BSSID, CHANNEL, SIGNAL,
          ENCRYPT, EAP to corresponding column in a DB table.
Step 11 : Finish
```

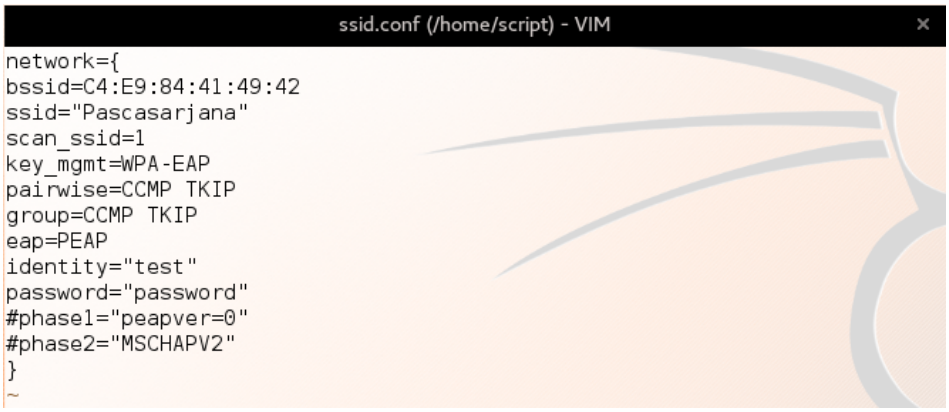
a. Sistem melakukan *scanning Access Point* menggunakan *tool* Iwlist pada *interface* wlan0.

b. Sistem mengekstrak data hasil *scanning Access Point* menggunakan *script* Python untuk mendapatkan parameter SSID, MAC *address*, *channel*, *power*, dan tipe enkripsi.
c. Sistem menugaskan (*assign*) parameter hasil ekstraksi tersebut ke dalam variabel seperti ditunjukkan pada Tabel 4.3.

Tabel 4.3 Penugasan parameter hasil ekstraksi ke dalam variabel

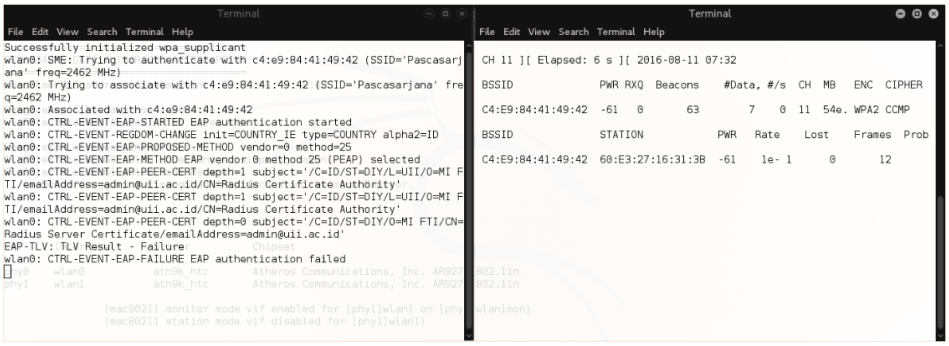
Parameter	Variabel	Fungsi
SSID	SSID	Menyimpan nama SSID
MAC *address*	BSSID	Menyimpan MAC *address* SSID
Channel	CHANNEL	Menyimpan nomor *channel* SSID
Power	SIGNAL	Menyimpan nilai kekuatan sinyal SSID
Tipe enkripsi	ENCRYPT	Menyimpan tipe enkripsi SSID

d. Sistem mengambil variabel "SSID", "BSSID", dan "CHANNEL" untuk digunakan dalam aktivitas *capture* trafik *wireless*.
   - Agar dapat melakukan *capture* terhadap paket EAP pada SSID dengan MAC *address* tertentu, *tool* Airodump-ng yang dijalankan pada *interface* wlan1mon harus menggunakan parameter *channel*. Di sinilah variabel "BSSID" (MAC *address*) dan variabel "CHANNEL" digunakan.
   - Saat aktivitas *capture* trafik *wireless* berlangsung, sistem melakukan percobaan koneksi ke SSID yang MAC *address*-nya sedang di-*capture* trafik. Percobaan koneksi ini bertujuan agar sistem dapat memberikan input berupa *username* dan *password* sembarang, bukan yang valid. Hal ini dilakukan untuk memicu (*trigger*) pengiriman paket EAP.
   - Namun sebelum *trigger* paket EAP dijalankan menggunakan *tool* Wpa-supplicant, sistem terlebih dahulu harus melakukan *setting* file konfigurasi yang akan dijalankan oleh Wpa-supplicant. File konfigurasi seperti yang ditunjukkan pada Gambar 4.35 menggunakan parameter SSID dan MAC *address*, sehingga di sinilah variabel "SSID" dan "BSSID" berperan.



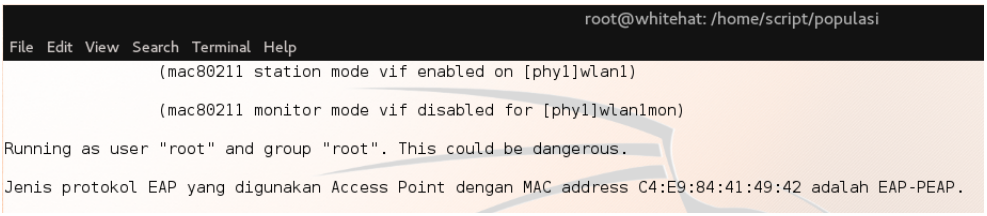
Gambar 4.35 File konfigurasi Wpa_supplicant

- Gambar 4.36 menunjukkan aktivitas *capture* trafik *wireless* menggunakan *tool* Airodump-ng (*window* sebelah kanan) dan *trigger* pengiriman paket EAP menggunakan *tool* Wpa_supplicant (*window* sebelah kiri) yang dilakukan ketika proses *capture* berlangsung.



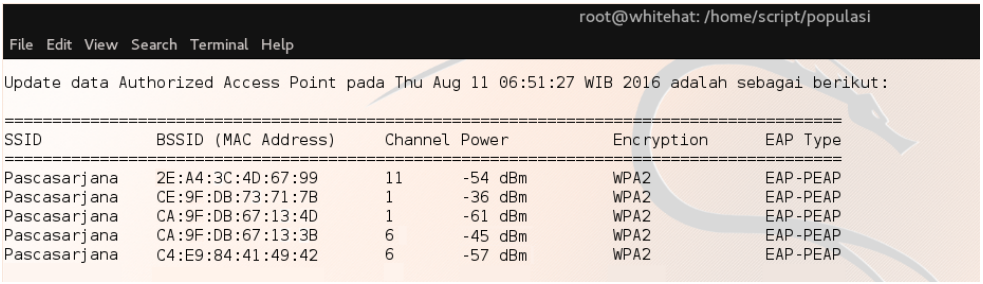
Gambar 4.36 *Capture* trafik *wireless* dan *trigger* paket EAP otomatis

e. Sistem menghentikan aktivitas *capture* trafik *wireless* setelah beberapa saat.
f. Sistem menganalisis file hasil *capture* trafik *wireless* menggunakan *tool* Tshark untuk memeriksa jenis protokol EAP yang digunakan oleh SSID dengan MAC *address* yang tadi di-*capture* trafiknya. Hasil analisis disimpan dalam variabel “EAP”. Misalnya, jika di dalam file *capture* ditemukan jenis protokol EAP-PEAP, maka variabel “EAP” akan berisi “EAP-PEAP”. Jika tidak ditemukan, maka variabel “EAP” akan berisi “None”.



Gambar 4.37 Hasil Tshark mengetahui ada tidaknya protokol EAP

g. Gambar 4.38 menunjukkan sistem telah menginputkan isi variabel seperti yang dijelaskan pada Tabel 4.4 ke dalam file tabel *authorized* AP.



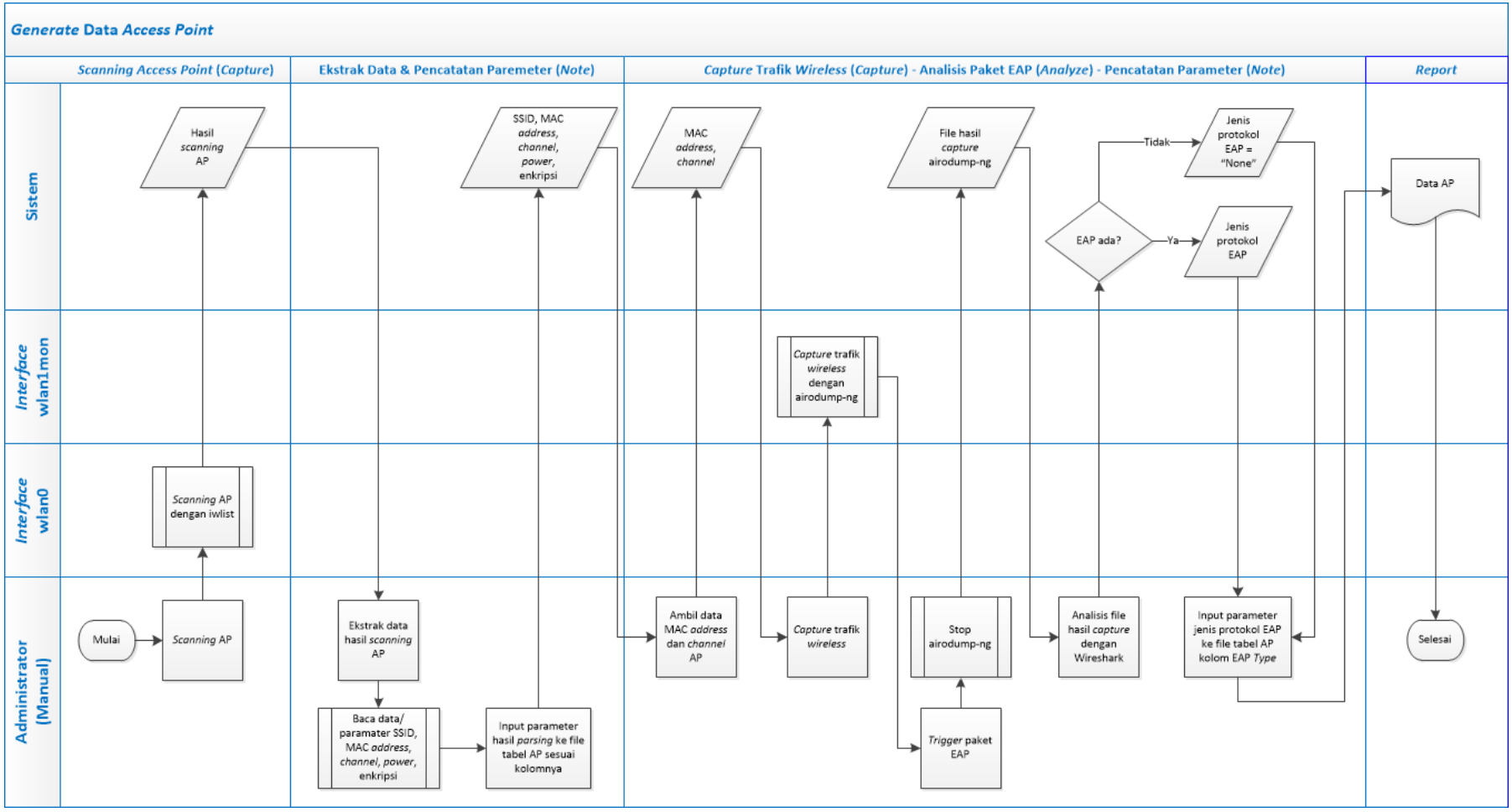
Gambar 4.38 Pencatatan isi variabel ke dalam file tabel *Access Point*

Tabel 4.4 Variabel yang dicatatkan ke dalam file tabel *authorized* AP

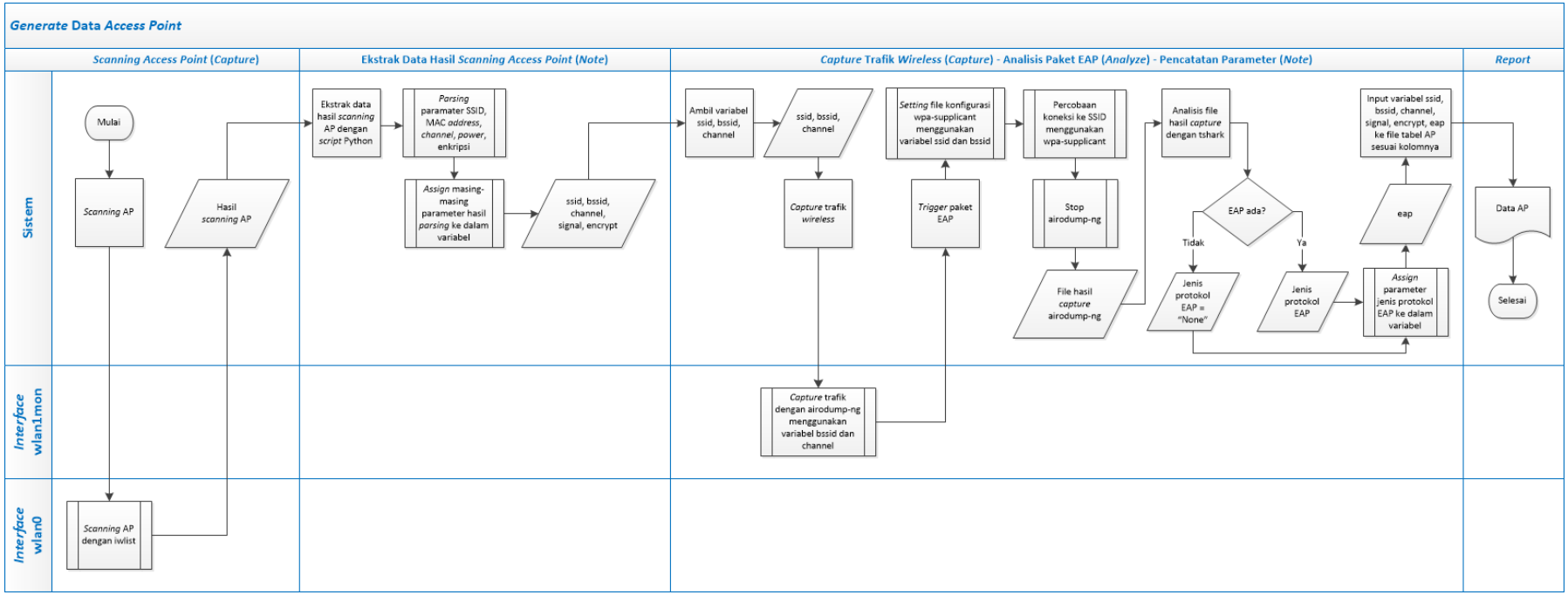
Parameter	Variabel	Dicatat pada Kolom
SSID	SSID	SSID
MAC *address*	BSSID	BSSID (MAC *Address*)
Channel	CHANNEL	*Channel*
Power	SIGNAL	*Power*
Tipe enkripsi	ENCRYPT	*Encryption*
Jenis protokol EAP	EAP	EAP *Type*

Flowchart otomatisasi fase *generate* data *Access Point* ini ditunjukkan pada Gambar 4.40.

Berdasarkan fase *generate* data *Access Point* secara manual dan terotomatisasi yang telah dijabarkan tersebut, dapat diperoleh kesimpulan bahwa perbedaan keduanya terletak pada langkah-langkah yang ditunjukkan dalam Tabel 4.5.



Gambar 4.39 *Flowchart* fase *generate* data *Access Point* secara manual



Gambar 4.40 *Flowchart* otomatisasi fase *generate* data *Access Point*

Tabel 4.5 Perbedaan fase *generate* data AP pada tahap populasi data *authorized* AP secara manual dan terotomatisasi

Langkah	Administrator (Manual)	Sistem (Otomatisasi)
Scanning AP	Eksekusi perintah Iwlist secara manual	Eksekusi perintah Iwlist
Ekstrak data	Membaca data secara manual	*Script* Python
Capture trafik *wireless*	Eksekusi perintah Airodump-ng (dengan parameter MAC *address* dan *channel*) secara manual	Eksekusi perintah Airodump-ng (dengan parameter MAC *address* dan *channel*)
Trigger paket EAP	Percobaan koneksi ke SSID yang di-*capture* trafiknya secara manual	• *Setting* file konfigurasi Wpa-supplicant (dengan parameter SSID dan MAC *address*) • Eksekusi perintah Wpa-supplicant
Analisis file hasil *capture*	Analisis jenis protokol EAP menggunakan Wireshark	Memeriksa jenis protokol EAP menggunakan Tshark
Pencatatan parameter ke dalam file tabel *authorized* AP	Input parameter secara manual dan dilakukan secara bertahap, yaitu: • Parameter SSID, MAC *address*, *channel*, *power*, dan tipe enkripsi dicatatkan setelah langkah ekstrak data hasil *scanning* AP selesai. • Parameter jenis protokol EAP dicatatkan setelah langkah analisis file hasil *capture* selesai.	Input parameter SSID, MAC *address*, *channel*, *power*, tipe enkripsi, dan jenis protokol EAP dilakukan secara bersamaan setelah langkah analisis file hasil *capture* selesai.

Kesimpulan lain yang dapat diambil adalah terdapat kelemahan pada langkah *trigger* paket EAP secara manual dengan melakukan percobaan koneksi ke SSID yang sedang di-*capture* trafiknya menggunakan pasrameter MAC *address*, yaitu administrator tidak mengetahui SSID mana yang memiliki MAC *address* yang sedang di-*capture* trafiknya tersebut. Misalnya terdapat lebih dari satu SSID yang sama, administrator bisa saja melakukan percobaan koneksi ke SSID yang MAC *address*-nya tidak sedang di-*capture* trafiknya. Oleh karenanya, tugas percobaan koneksi secara manual tersebut dapat diperbaiki menggunakan *script* otomatisasi dengan menjalankan *tool* Wpa-supplicant karena file konfigurasinya menggunakan parameter SSID dan MAC *address*. Gambar 4.41 menunjukkan potongan *script* otomatisasi tahap populasi data *authorized* AP. *Script* otomatisasi tahap populasi data *authorized* AP lebih lengkap terdapat pada Lampiran 3.



Gambar 4.41 Potongan *script* otomatisasi tahap populasi data *authorized* AP

4.4.3 Tahap Deteksi *Rogue Access Point*

Satu metode yang telah digunakan dalam banyak penelitian untuk mendeteksi adanya *rogue Access Point* (*rogue* AP) adalah metode pengukuran *Round Trip Time* (RTT). *Client* dapat menggunakan informasi waktu untuk mendeteksi *Access Point* palsu karena *Access Point* palsu tersebut memiliki *wireless link* tambahan dibandingkan *Access Point* asli. Tambaha link tersebut dapat menyebabkan adanya penundaan (*delay*) dalam transmisi data. *Client* dapat menemukan RTT dengan mengontak (*probe*) server di jaringan lokal dengan cara mengirimkan pesan seperti *ping request* atau paket data TCP dan menunggu jawaban. *Client* mengulangi proses ini selama beberapa kali dan mencatat semua RTT. Jika nilai rata-rata dari RTT secara statistik lebih besar dari batas (*threshold*) tertentu, maka sebuah AP dapat dikategorikan sebagai *rogue* AP (Sonawane, Chavan, & Ghodeswar, 2013). Penggunaan RTT untuk mendeteksi *rogue* AP perlu memperhatikan tiga isu berikut. Isu pertama adalah server mana yang akan dicoba untuk dikontak. Sebuah server di jaringan lokal lebih diutamakan daripada server *remote* di internet karena metode berbasis RTT sensitif terhadap keterlambatan yang ada dalam jaringan kabel. *Probe* ke server *remote* akan menyebabkan variasi RTT yang signifikan disebabkan jalur *routing* yang dinamis dan besarnya trafik internet. Isu kedua adalah jenis pesan *probe* untuk digunakan. Pesan *probe* yang digunakan haruslah pesan yang tidak dapat dengan mudah dimanipulasi oleh *rogue* AP dan dapat mencapai server terlepas dari konfigurasi jaringannya. Pesan *ping* sederhana dapat dengan mudah dikembalikan oleh *rogue* AP untuk menghindari deteksi dan mungkin diblokir oleh beberapa administrator jaringan. Selain itu, pesan *probe* harus mematuhi protokol jaringan yang sudah ada sehingga tidak diperlukan bantuan dari penyedia jaringan. Isu ketiga mengenai pertimbangan efek dari kondisi trafik jaringan. Jaringan dengan trafik yang sibuk dapat mempengaruhi RTT dan menyebabkan kesalahan dalam mendeteksi *rogue* AP (Han, Sheng, Tan, Li, & Lu, 2011).

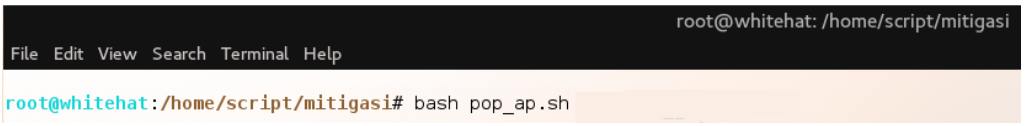
Tahap deteksi *rogue* AP dalam penelitian ini tidak menggunakan parameter RTT karena *rogue* AP yang digunakan adalah yang berbasis *software*. *Rogue* AP berbasis *software* ini tidak memerlukan akses fisik langsung ke jaringan kabel.

Tahap deteksi *rogue* AP seperti halnya tahap populasi data *authorized Access Point* (*authorized* AP) dalam penelitian ini juga menggunakan fase *generate* data *Access Point*. Perbedaannya adalah pada tahap populasi data *authorized* AP menghasilkan data *authorized* AP, sedangkan pada tahap deteksi *rogue* AP menghasilkan data *unauthorized Access Point* (*unauthorized* AP).

4.4.3.1 Fase *Generate* Data *Access Point*

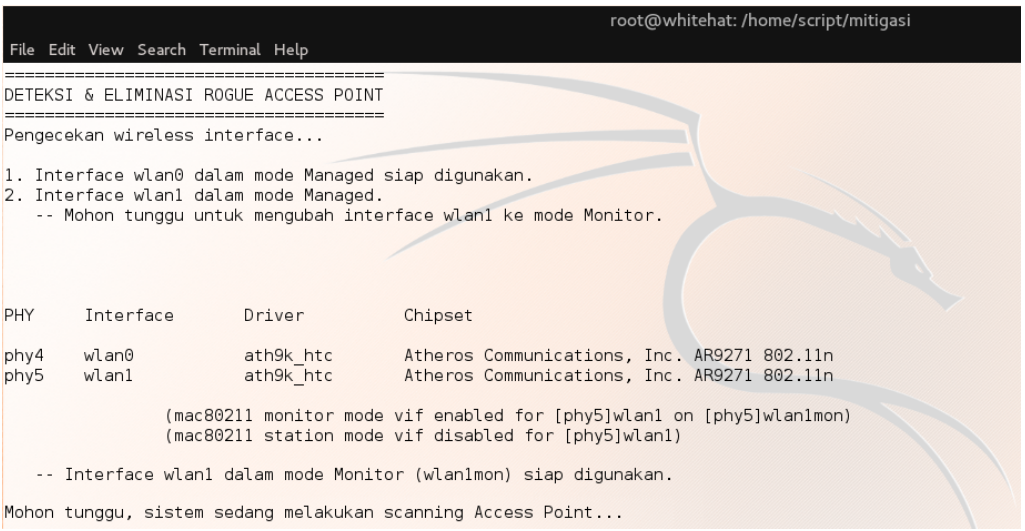
Menggunakan fase *generate* data *Access Point* yang telah dijabarkan sebelumnya di bagian 4.4.2, otomatisasi langkah-langkah pada tahap deteksi *rogue* AP sampai dengan menghasilkan data *unauthorized* AP adalah sebagai berikut:

1. Administrator mengeksekusi *script* deteksi *rogue* AP seperti pada Gambar 4.42.



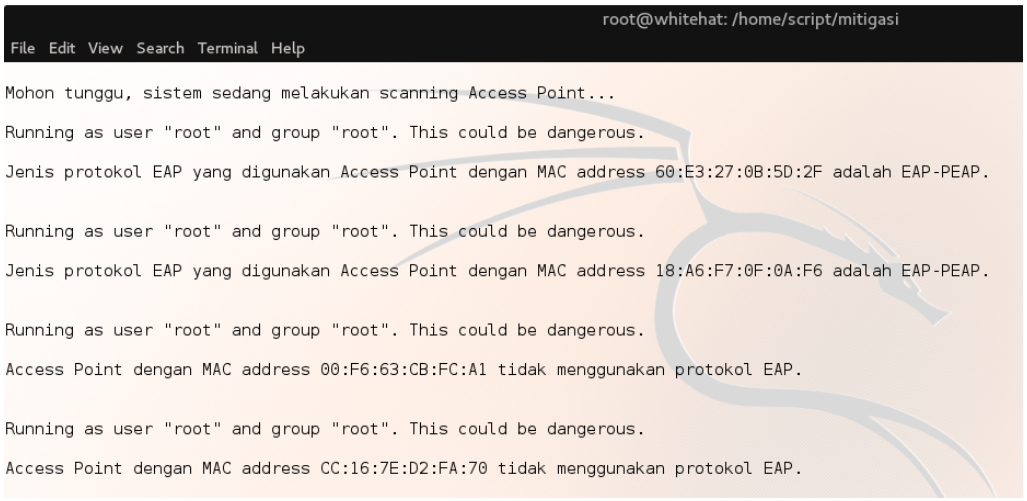
Gambar 4.42 Menjalankan *script* otomatisasi deteksi *rogue* AP

2. Sistem melakukan *scanning Access Point* menggunakan *tool* Iwlist pada *interface* wlan0 seperti yang ditunjukkan pada Gambar 4.43.



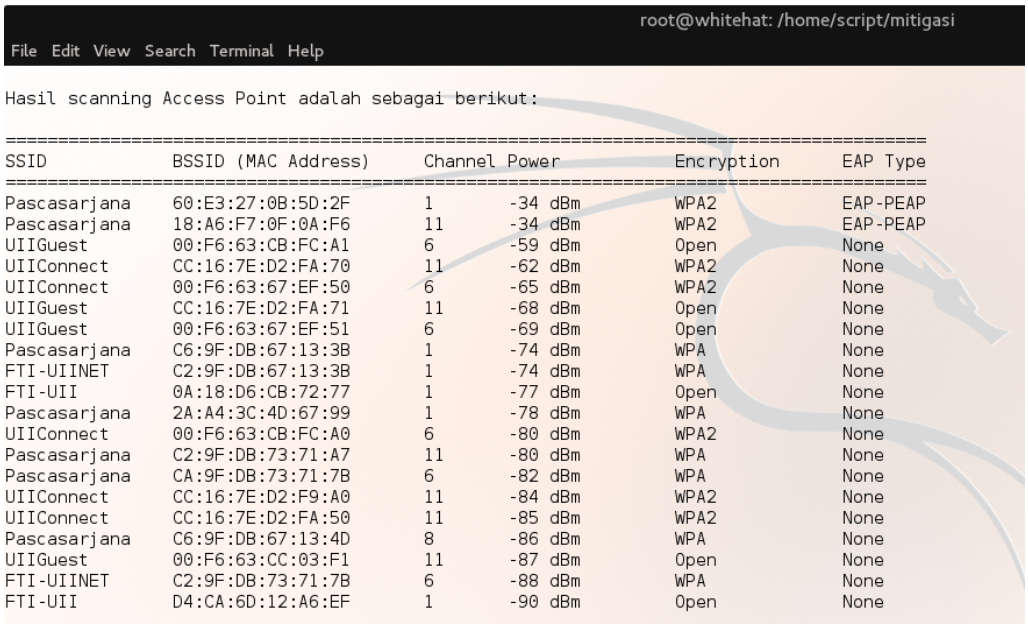
Gambar 4.43 *Scanning Access Point* oleh sistem

3. Sistem mengekstrak data hasil *scanning Access Point* menggunakan *script* Python untuk mendapatkan parameter SSID, MAC *address*, *channel*, *power*, dan tipe enkripsi, kemudian menugaskan (*assign*) parameter hasil ekstraksi tersebut ke dalam variabel "SSID", "BSSID", "CHANNEL", "SIGNAL", dan "ENCRYPT".
4. Sistem mengambil variabel "SSID", "BSSID", dan "CHANNEL" untuk digunakan dalam aktivitas *capture* trafik *wireless* sekaligus *trigger* pengiriman paket EAP.
5. Sistem menghentikan aktivitas *capture* trafik *wireless* setelah beberapa saat.
6. Sistem menganalisis file hasil *capture* trafik *wireless* menggunakan *tool* Tshark untuk memeriksa jenis protokol EAP yang digunakan oleh SSID dengan MAC *address* yang tadi di-*capture* trafiknya, seperti yang tampak pada Gambar 4.44. Jenis protokol EAP ini akan di-*assign* ke dalam variabel "EAP".



Gambar 4.44 *Script* menganalisis file hasil *capture* trafik *wireless*

7. Sistem akan menginputkan isi variabel "SSID", "BSSID", "CHANNEL", "SIGNAL", "ENCRYPT", dan "EAP" ke dalam file tabel *unauthorized* AP sesuai dengan kolomnya masing-masing, yaitu kolom SSID, BSSID (MAC *Address*), *Channel*, *Power*, *Encryption*, dan EAP *Type* seperti yang ditunjukkan pada Gambar 4.45.



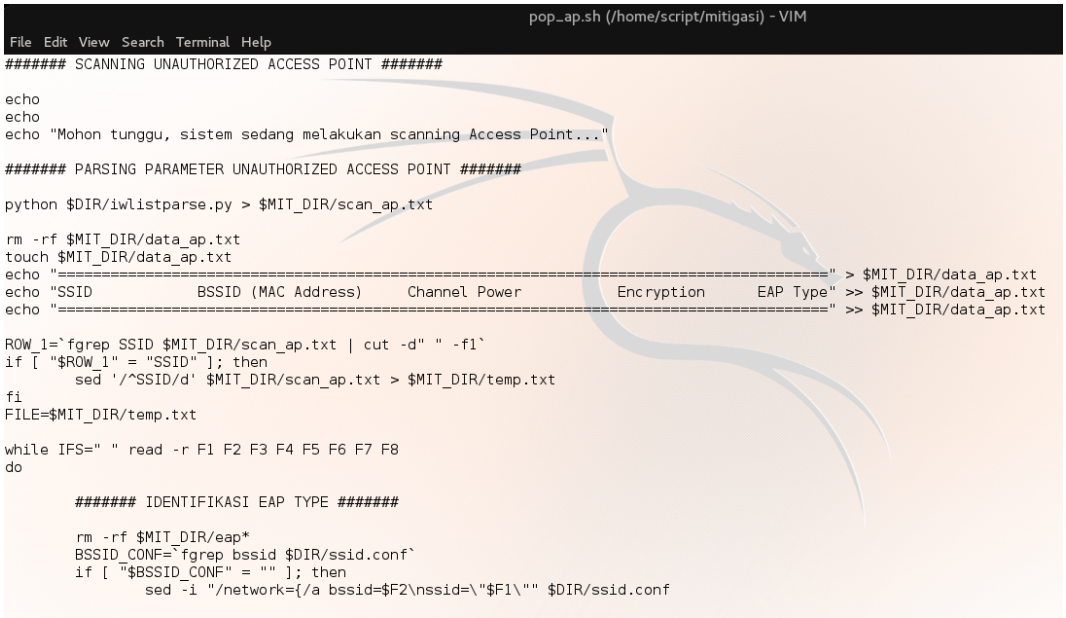
```
root@whitehat: /home/script/mitigasi
File  Edit  View  Search  Terminal  Help

Hasil scanning Access Point adalah sebagai berikut:

================================================================================================
SSID            BSSID (MAC Address)     Channel Power          Encryption      EAP Type
================================================================================================
Pascasarjana    60:E3:27:0B:5D:2F       1       -34 dBm        WPA2            EAP-PEAP
Pascasarjana    18:A6:F7:0F:0A:F6       11      -34 dBm        WPA2            EAP-PEAP
UIIGuest        00:F6:63:CB:FC:A1       6       -59 dBm        Open            None
UIIConnect      CC:16:7E:D2:FA:70       11      -62 dBm        WPA2            None
UIIConnect      00:F6:63:67:EF:50       6       -65 dBm        WPA2            None
UIIGuest        CC:16:7E:D2:FA:71       11      -68 dBm        Open            None
UIIGuest        00:F6:63:67:EF:51       6       -69 dBm        Open            None
Pascasarjana    C6:9F:DB:67:13:3B       1       -74 dBm        WPA             None
FTI-UIINET      C2:9F:DB:67:13:3B       1       -74 dBm        WPA             None
FTI-UII         0A:18:D6:CB:72:77       1       -77 dBm        Open            None
Pascasarjana    2A:A4:3C:4D:67:99       1       -78 dBm        WPA             None
UIIConnect      00:F6:63:CB:FC:A0       6       -80 dBm        WPA2            None
Pascasarjana    C2:9F:DB:73:71:A7       11      -80 dBm        WPA             None
Pascasarjana    CA:9F:DB:73:71:7B       6       -82 dBm        WPA             None
UIIConnect      CC:16:7E:D2:F9:A0       11      -84 dBm        WPA2            None
UIIConnect      CC:16:7E:D2:FA:50       11      -85 dBm        WPA2            None
Pascasarjana    C6:9F:DB:67:13:4D       8       -86 dBm        WPA             None
UIIGuest        00:F6:63:CC:03:F1       11      -87 dBm        Open            None
FTI-UIINET      C2:9F:DB:73:71:7B       6       -88 dBm        WPA             None
FTI-UII         D4:CA:6D:12:A6:EF       1       -90 dBm        Open            None
```

Gambar 4.45 Pencatatan isi variabel ke dalam tabel *unauthorized* AP

Gambar 4.46 menunjukkan potongan *script* otomatisasi fase *generate* data *Access Point* pada tahap deteksi *rogue* AP. *Script* otomatisasi fase *generate* data *Access Point* pada tahap deteksi *rogue* AP lebih lengkap terdapat pada Lampiran 4.



```
pop_ap.sh (/home/script/mitigasi) - VIM
File  Edit  View  Search  Terminal  Help
####### SCANNING UNAUTHORIZED ACCESS POINT #######

echo
echo
echo "Mohon tunggu, sistem sedang melakukan scanning Access Point..."

####### PARSING PARAMETER UNAUTHORIZED ACCESS POINT #######

python $DIR/iwlistparse.py > $MIT_DIR/scan_ap.txt

rm -rf $MIT_DIR/data_ap.txt
touch $MIT_DIR/data_ap.txt
echo "================================================================================" > $MIT_DIR/data_ap.txt
echo "SSID            BSSID (MAC Address)     Channel Power          Encryption      EAP Type" >> $MIT_DIR/data_ap.txt
echo "================================================================================" >> $MIT_DIR/data_ap.txt

ROW_1=`fgrep SSID $MIT_DIR/scan_ap.txt | cut -d" " -f1`
if [ "$ROW_1" = "SSID" ]; then
        sed '/^SSID/d' $MIT_DIR/scan_ap.txt > $MIT_DIR/temp.txt
fi
FILE=$MIT_DIR/temp.txt

while IFS=" " read -r F1 F2 F3 F4 F5 F6 F7 F8
do

        ####### IDENTIFIKASI EAP TYPE #######

        rm -rf $MIT_DIR/eap*
        BSSID_CONF=`fgrep bssid $DIR/ssid.conf`
        if [ "$BSSID_CONF" = "" ]; then
                sed -i "/network={/a bssid=$F2\nssid=\"$F1\"" $DIR/ssid.conf
```

Gambar 4.46 Potongan *script* otomatisasi fase *generate* data *Access Point* pada tahap deteksi *rogue* AP

Fase *generate* data *Access Point* pada tahap deteksi *rogue* AP ini mewakili metode *live forensics* seperti pada Gambar 3.5 mengenai tahapan deteksi *rogue* AP yang secara umum meliputi proses, subproses, dan sub-subproses seperti berikut:

1. *Capture*, terdiri dari dua subproses:
   a. *Scanning Access Point*
   b. *Capture* trafik *wireless*, memiliki sub-subproses *trigger* paket EAP.
2. *Note*, terdiri dari dua subproses:
   a. Ekstraksi data hasil *scanning Access Point*
   b. Pencatatan data hasil ekstraksi dan hasil analisis
3. *Analyze*, untuk menganalisis file hasil *capture* trafik *wireless*.
4. *Report*, menghasilkan data *unauthorized* AP.

   Fase *generate* data *Access Point* ini masih belum mencakup proses *Report* untuk subproses komparasi data dan data *rogue* AP.

Berdasarkan penjabaran di atas dan dikombinasikan dengan Gambar 4.39 serta Gambar 4.40 mengenai *flowchart* fase *generate* data *Access Point* secara manual dan terotomatisasi, perbedaan fase tersebut baik pada tahap populasi data *authorized* AP maupun tahap deteksi *rogue* AP dapat dijelaskan kronologinya pada Tabel 4.6. Terlihat bahwa perbedaan antara cara manual dan terotomatisasi adalah pada proses *Note*. Untuk cara manual, dilakukan dua kali pencatatan data, yaitu pencatatan data hasil ekstraksi dilakukan setelah proses *Capture* subproses *scanning Access Point*, serta pencatatan data hasil analisis dilakukan setelah proses *Analyze*.

Tabel 4.6 Perbedaan fase *generate* data AP pada tahap populasi data *authorized* AP dan tahap deteksi *rogue* AP

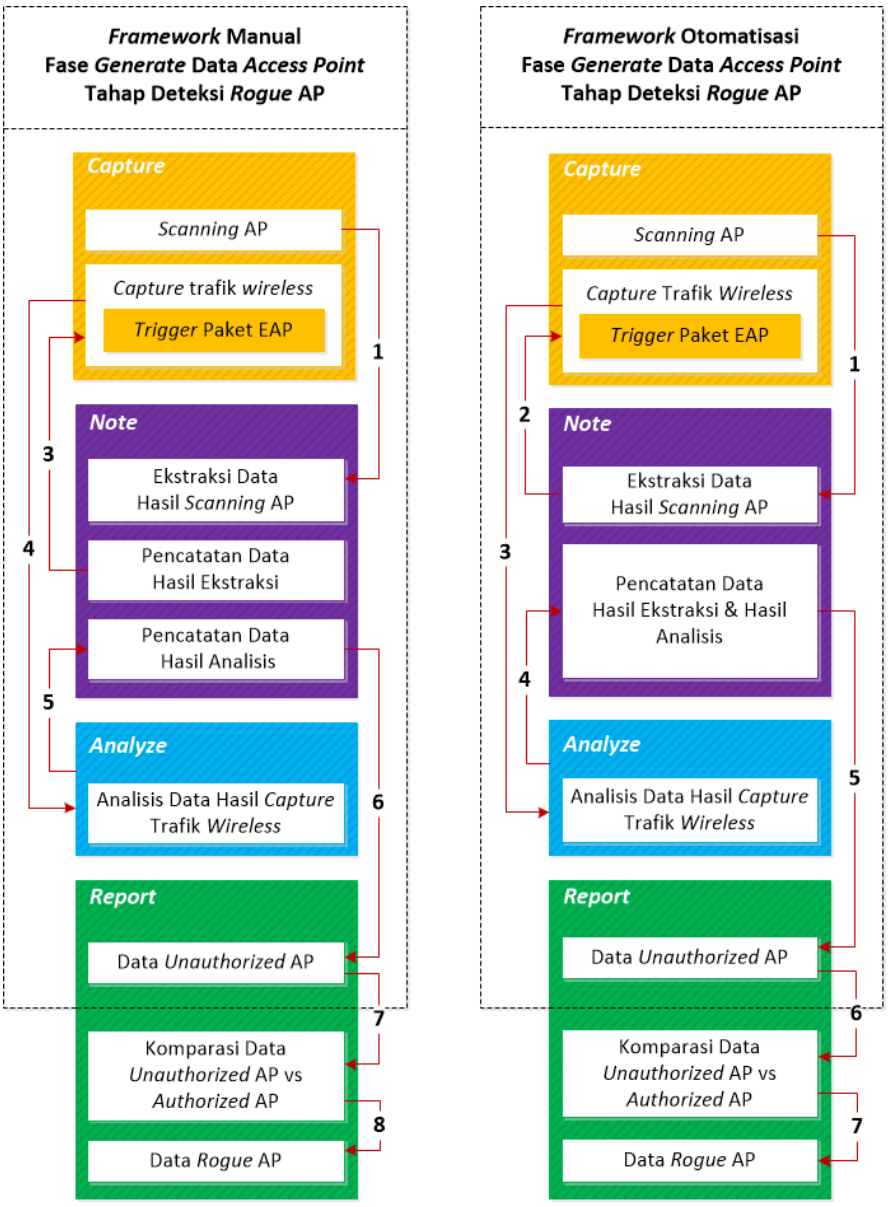
Proses	**Tahap Populasi Data *Authorized* AP**		**Tahap Deteksi *Rogue* AP**	
	Manual	**Otomatisasi**	**Manual**	**Otomatisasi**
Capture	*Scanning* AP	*Scanning* AP	*Scanning* AP	*Scanning* AP
Note	Ekstraksi data hasil *scanning* AP	Ekstraksi data hasil *scanning* AP	Ekstraksi data hasil *scanning* AP	Ekstraksi data hasil *scanning* AP
Note	Pencatatan data hasil ekstraksi	-	Pencatatan data hasil ekstraksi	-
Capture	*Capture* trafik *wireless*	*Capture* trafik *wireless*	*Capture* trafik *wireless*	*Capture* trafik *wireless*
Analyze	Analisis file hasil *capture* trafik *wireless*	Analisis file hasil *capture* trafik *wireless*	Analisis file hasil *capture* trafik *wireless*	Analisis file hasil *capture* trafik *wireless*
Note	Pencatatan data hasil analisis	Pencatatan data hasil ekstraksi & hasil analisis	Pencatatan data hasil analisis	Pencatatan data hasil ekstraksi & hasil analisis
Report	Data *authorized* AP	Data *authorized* AP	Data *unauthorized* AP	Data *unauthorized* AP

Berdasarkan Tabel 4.6 tersebut, dikembangkanlah *framework* fase *generate* data *Access Point* tahap populasi data *authorized* AP seperti ditunjukkan pada Gambar 4.47 dan *framework* fase *generate* data *Access Point* tahap deteksi *rogue* AP seperti ditunjukkan pada Gambar 4.48.



Gambar 4.47 *Framework* fase *generate* data *Access Point* pada tahap populasi data *authorized* AP

Framework fase *generate* data *Access Point* tahap populasi data *authorized* AP secara manual terdiri dari enam langkah karena terjadi dua kali subproses pencatatan data. Sedangkan dengan cara otomatisasi, dapat dipersingkat menjadi lima langkah karena subproses pencatatan data dapat dilakukan satu kali secara bersamaan.

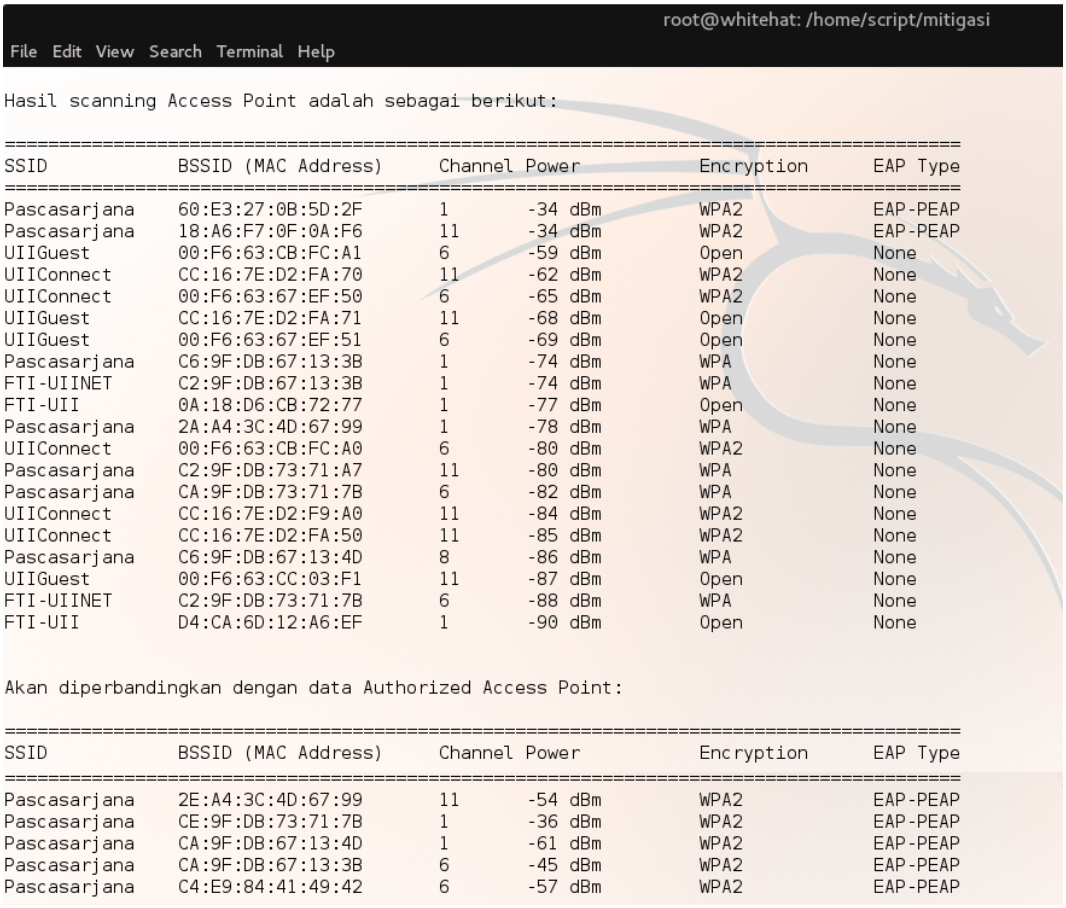


Gambar 4.48 *Framework* fase *generate* data *Access Point* pada tahap deteksi *rogue* AP

Framework fase *generate* data *Access Point* tahap deteksi *rogue* AP secara manual juga terdiri dari enam langkah karena terjadi dua kali subproses pencatatan data. Sedangkan dengan cara otomatisasi, juga dapat dipersingkat menjadi lima langkah karena subproses pencatatan data dapat dilakukan satu kali secara bersamaan. Dan perlu digarisbawahi bahwa fase *generate* data *Access Point* tahap deteksi *rogue* AP ini belum mencakup proses *Report* subproses komparasi data dan data *rogue* AP.

4.4.3.2 Komparasi Data dan Data *Rogue* AP

Proses lanjutan dari fase *generate* data *Access Point* pada tahap deteksi *rogue* AP adalah proses *Report* dengan subproses komparasi data dan data *rogue* AP. Komparasi data dilakukan dengan memperbandingkan data *unauthorized* AP yang dihasilkan dari fase *generate Access Point* tahap deteksi *rogue* AP (Gambar 4.45) dengan data *authorized* AP yang dihasilkan dari tahap populasi data *authorized* AP (Gambar 4.38) seperti yang ditunjukkan pada Gambar 4.49.



Gambar 4.49 Data *unauthorized* AP dan data *authorized* AP

Algoritma komparasi data yang meliputi parameter SSID, MAC *address*, *power*, tipe enkripsi, dan jenis *protokol* EAP dapat dijabarkan sebagai berikut:

```
Step 1  : Start
Step 2  : While (read rows in DB table);
              Do {read variables in each column}.
Step 3  : While (read rows in Authorized DB table);
          Do {read variables in each column}.
Step 4  : Compare variables in from each DB table
          If (SSID != SSID Authorized AP) then {AP ← Neighbor AP};
              Else {AP ← Unauthorized AP}.
              Next proceed with Unauthorized AP only.
          If (BSSID = BSSID Authorized AP) then {AP ← Authorized AP};
              Else {AP ← Unauthorized AP}.
              Next proceed with Unauthorized AP only.
          If (SIGNAL <= -80 dBm) then {AP ← Neighbor AP};
              Else {AP ← Unauthorized AP}.
              Next proceed with Unauthorized AP only.
          If (ENCRYPT != ENCRYPT Authorized AP) then {AP ← Neighbor
          AP};
              Else {AP ← Unauthorized AP}.
              Next proceed with Unauthorized AP only.
          If (EAP = EAP Authorized AP) then {AP ← Rogue AP};
              Else (AP ← Neighbor AP).
Step 5  : Finish
```

Ketentuan dari algoritma komparasi data yang telah dijabarkan tersebut dijelaskan sebagai berikut:

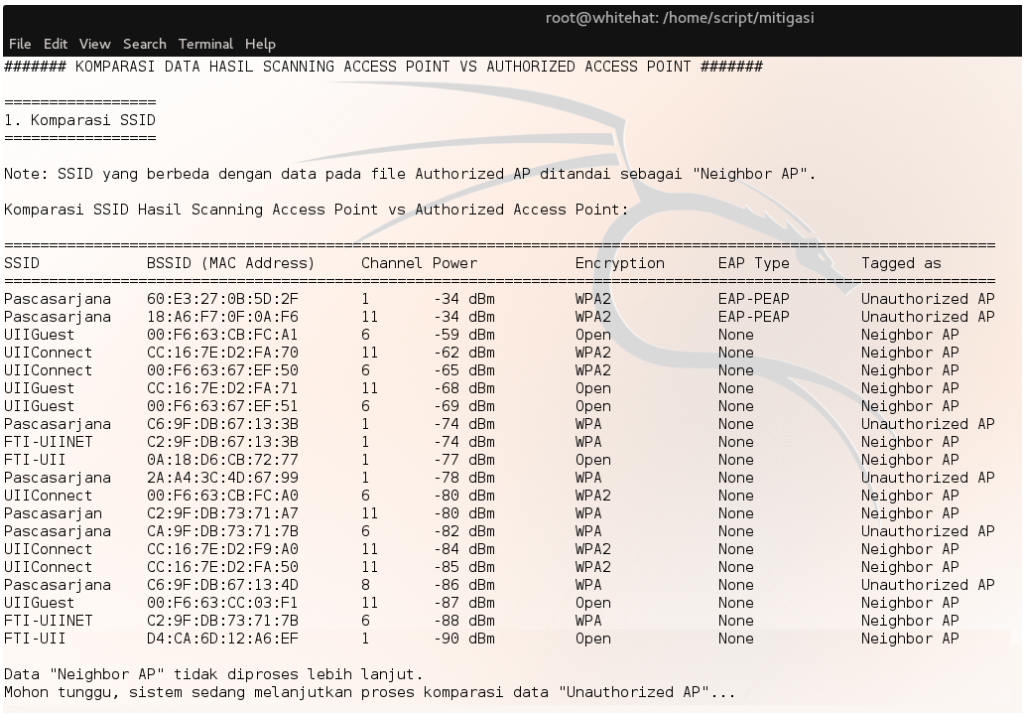
1. **Komparasi SSID**

   Komparasi SSID seperti yang ditunjukkan pada Gambar 4.50 dapat dijelaskan sebagai berikut:

   a. Jika SSID yang digunakan *unauthorized* AP tidak sama dengan data *authorized* AP maka *Acces Point* tersebut dikategorikan sebagai *neighbor Access Point* (*neighbor* AP).

   b. Jika SSID yang digunakan *unauthorized* AP sama dengan data *authorized* AP maka *Acces Point* tersebut tetap dikategorikan sebagai *unauthorized* AP.

   Data *neighbor* AP tidak diproses lebih lanjut.



Gambar 4.50 Komparasi SSID pada tahap deteksi *rogue* AP

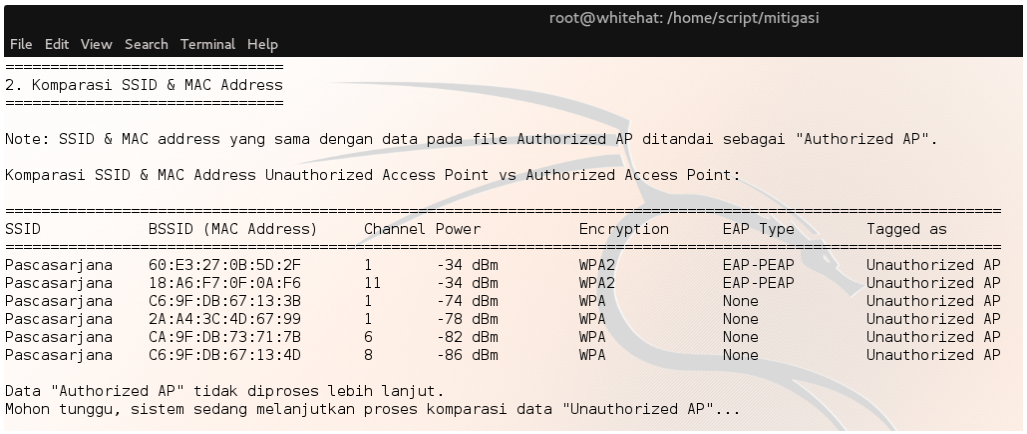
2. **Komparasi SSID dan MAC *address***

   Komparasi SSID dan MAC *address* seperti yang ditunjukkan pada Gambar 4.51 dapat dijelaskan sebagai berikut:

   a. Jika SSID dan MAC *address* yang digunakan *unauthorized* AP tidak sama dengan data *authorized* AP maka *Acces Point* tersebut tetap dikategorikan sebagai *unauthorized* AP.

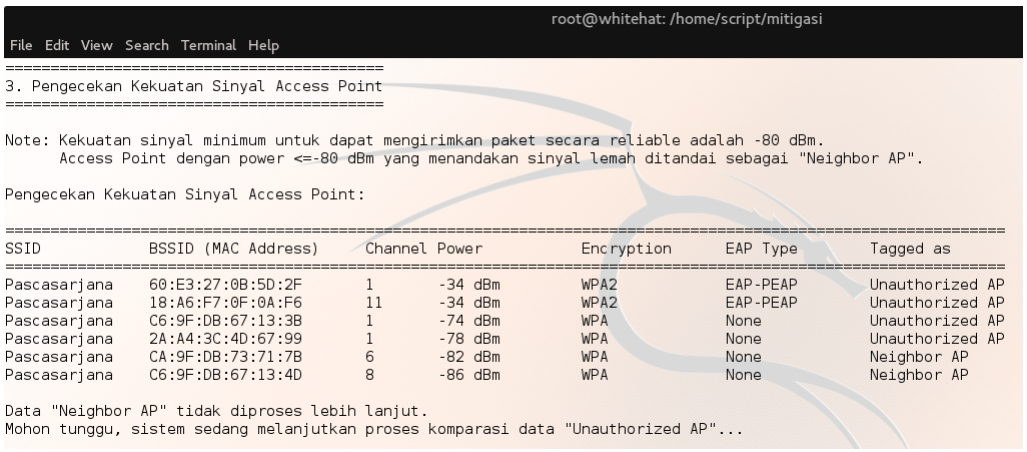
   b. Jika SSID dan MAC *address* yang digunakan *unauthorized* AP sama dengan data *authorized* AP maka *Acces Point* tersebut dikategorikan sebagai *authorized* AP.

   Data *authorized* AP tidak diproses lebih lanjut.



Gambar 4.51 Komparasi SSID dan MAC *address* pada tahap deteksi *rogue* AP

3. **Pengecekan kekuatan sinyal (*power*)**



Gambar 4.52 Pengecekan kekuatan sinyal pada tahap deteksi *rogue* AP

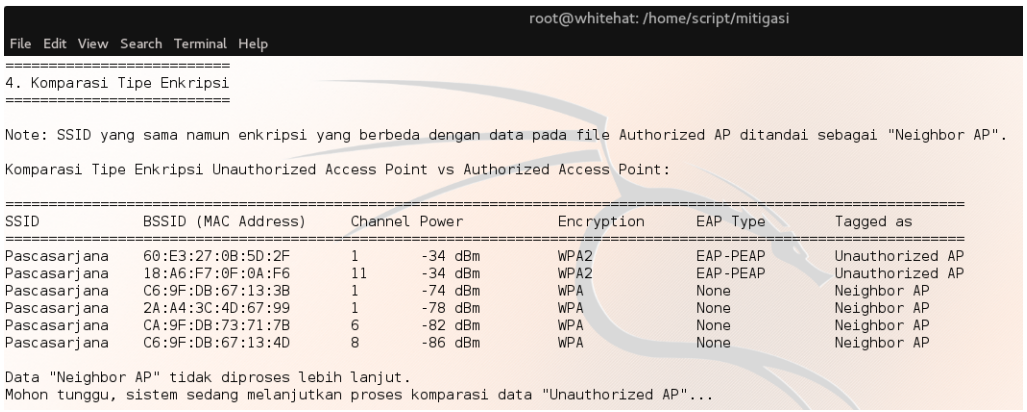
Pengecekan kekuatan sinyal seperti ditunjukkan pada Gambar 4.52. Kekuatan sinyal yang ditransmisikan oleh *Access Point* berada pada rentang sinyal yang baik untuk melakukan konektivitas (*basic connectivity*), yaitu minimum sinyal -80 dBm (Metageek, n.d.-a).

a. Jika *power unauthorized* AP lebih kecil daripada -80 dBm (contohnya -83 dBM, -85 dBm, dan seterusnya), maka *Acces Point* tersebut dikategorikan sebagai *neighbor* AP.

b. Jika *power unauthorized* AP lebih besar daripada -80 dBm (contohnya -70 dBM, -60 dBm, dan seterusnya), maka *Acces Point* tersebut tetap dikategorikan sebagai *unauthorized* AP.

Data *neighbor* AP tidak diproses lebih lanjut.

4. **Komparasi tipe enkripsi**



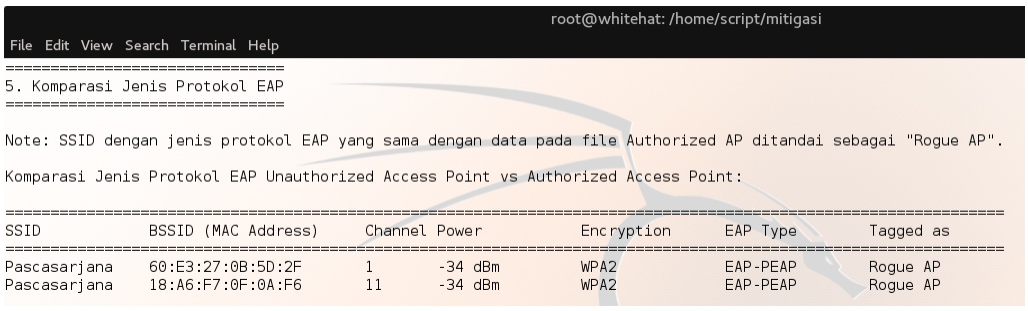
Gambar 4.53 Komparasi tipe enkripsi pada tahap deteksi *rogue* AP

Komparasi tipe enkripsi seperti yang ditunjukkan pada Gambar 4.53 dapat dijelaskan sebagai berikut:

a. Jika tipe enkripsi yang digunakan *unauthorized* AP tidak sama dengan data *authorized* AP maka *Acces Point* tersebut dikategorikan sebagai *neighbor* AP.
b. Jika tipe enkripsi yang digunakan *unauthorized* AP sama dengan data *authorized* AP maka *Acces Point* tersebut tetap dikategorikan sebagai *unauthorized* AP.

Data *neighbor* AP tidak diproses lebih lanjut.

5. **Komparasi jenis protokol EAP**

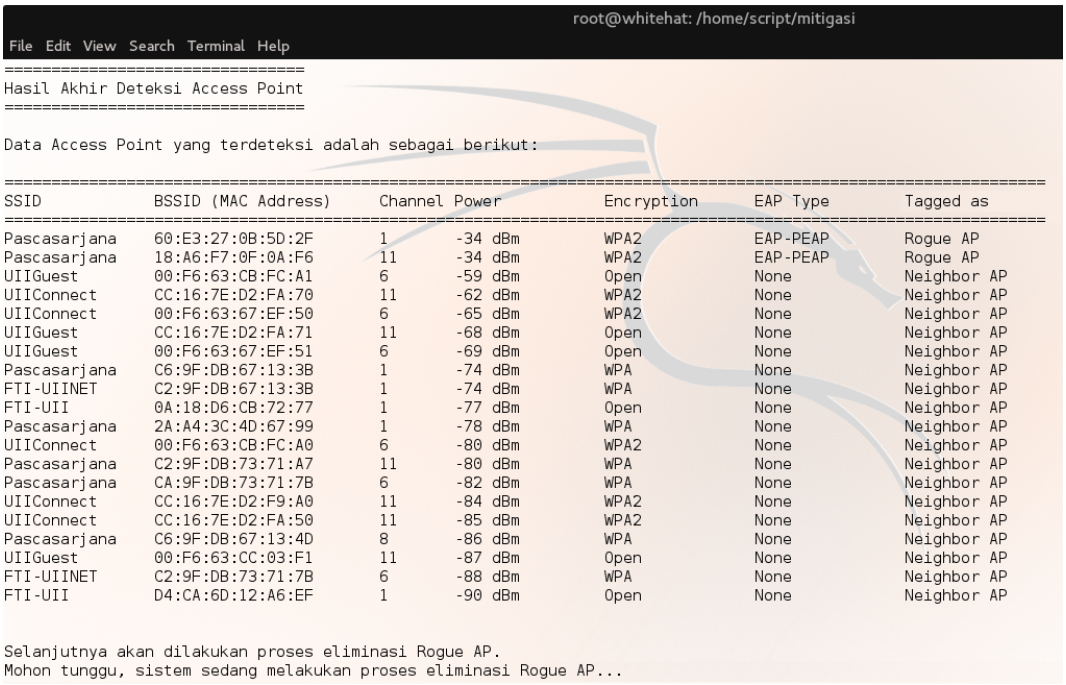


Gambar 4.54 Komparasi jenis protokol EAP pada tahap deteksi *rogue* AP

Komparasi jenis protokol EAP seperti yang ditunjukkan pada Gambar 4.54 dapat dijelaskan sebagai berikut:

a. Jika jenis protokol EAP *unauthorized* AP tidak sama dengan data *authorized* AP maka *Acces Point* tersebut dikategorikan sebagai *neighbor* AP.
b. Jika jenis protokol EAP *unauthorized* AP sama dengan data *authorized* AP maka *Acces Point* tersebut dikategorikan sebagai *rogue* AP.

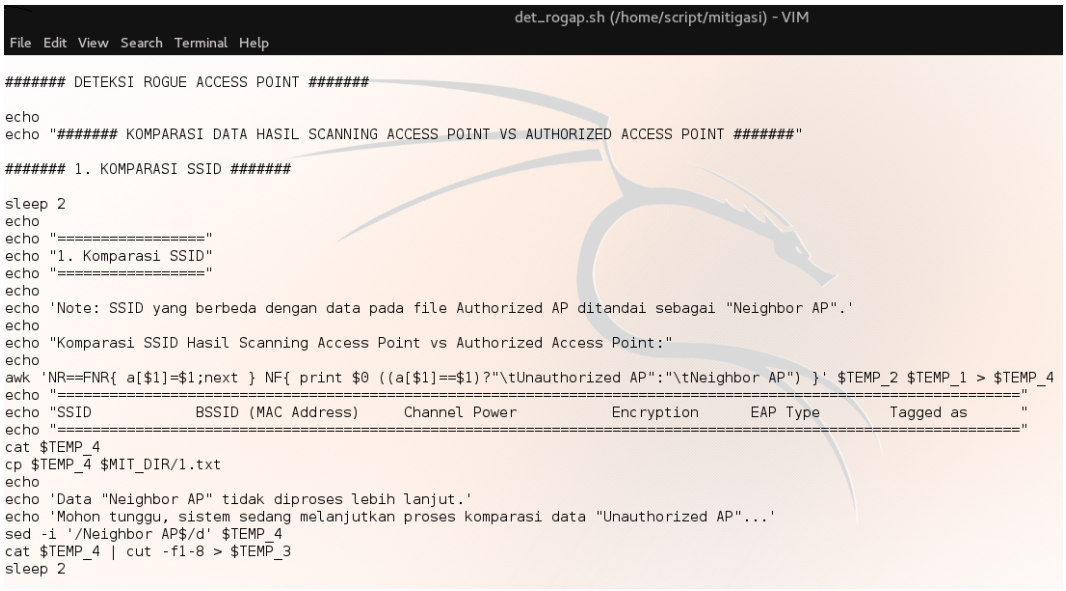
Hasil akhir dari tahap deteksi *rogue* AP adalah berupa data *Access Point* yang telah ditandai (*tagged as*), seperti ditunjukkan pada Gambar 4.55.



Gambar 4.55 Hasil akhir komparasi data *Access Point*

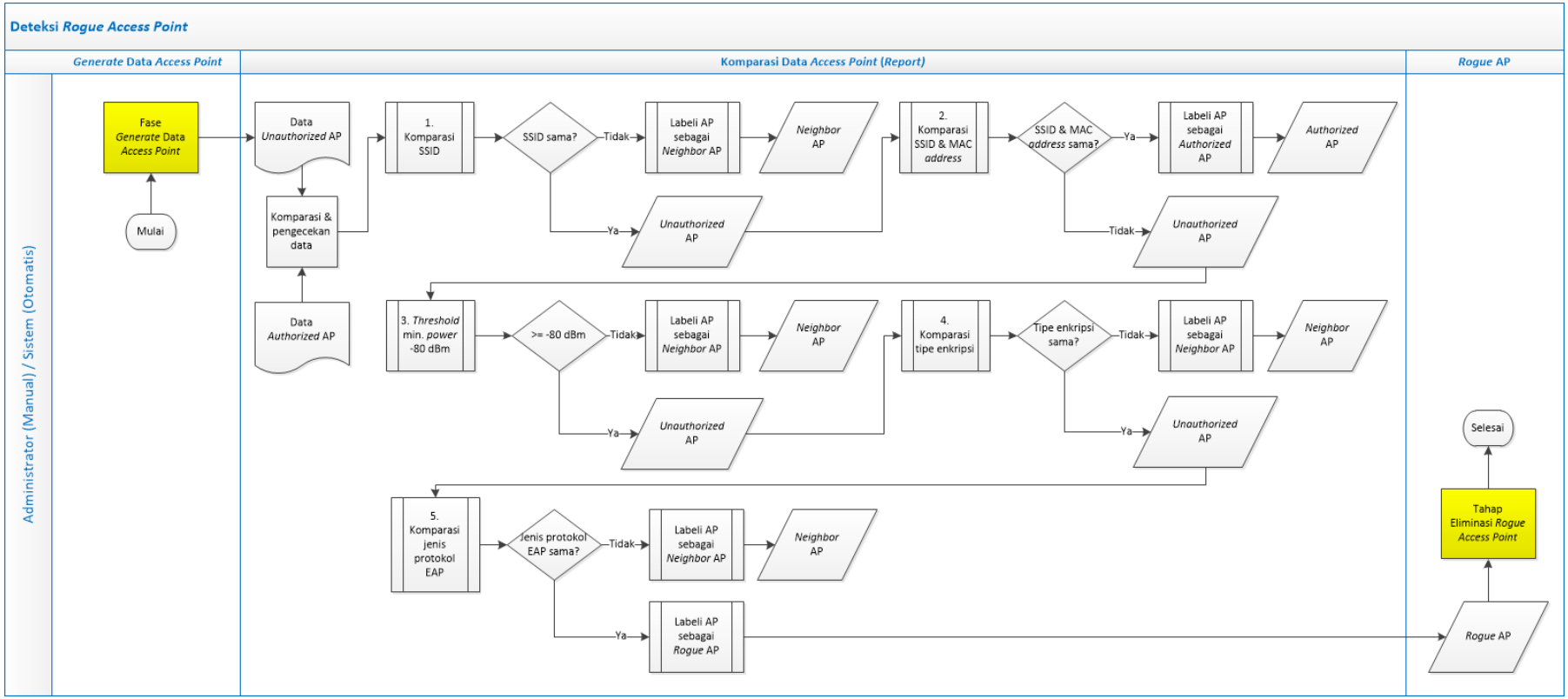
Data *Access Point* yang ditandai sebagai *rogue* AP akan diproses lebih lanjut ke tahap eliminasi *rogue* AP.

Gambar 4.56 menunjukkan potongan *script* otomatisasi komparasi data pada tahap deteksi *rogue* AP. *Script* otomatisasi komparasi data pada tahap deteksi *rogue* AP lebih lengkap terdapat pada Lampiran 5.



Gambar 4.56 Potongan *script* otomatisasi komparasi data pada tahap deteksi *rogue* AP

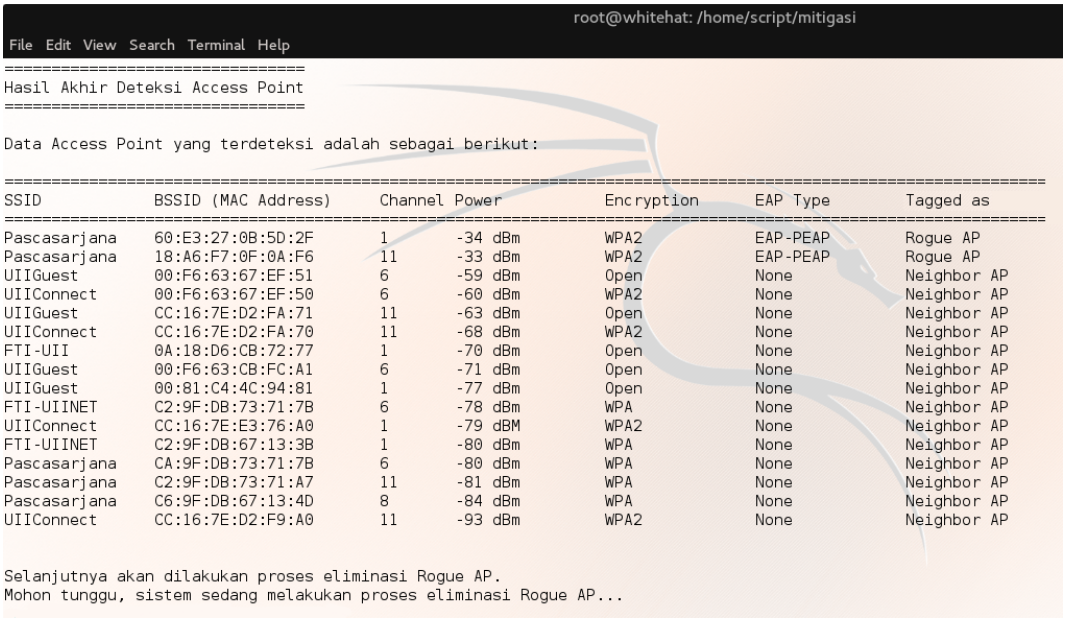
Penjabaran mengenai komparasi data *unauthorized* AP dan data *authorized* AP digambarkan dalam bentuk *flowchart* yang dapat dilihat pada Gambar 4.57. Data yang dihasilkan dari *flowchart* komparasi data ini adalah data *rogue* AP yang akan diteruskan ke tahap eliminasi *rogue* AP.



Gambar 4.57 *Flowchart* komparasi data *unauthorized* AP dan data *authorized* AP

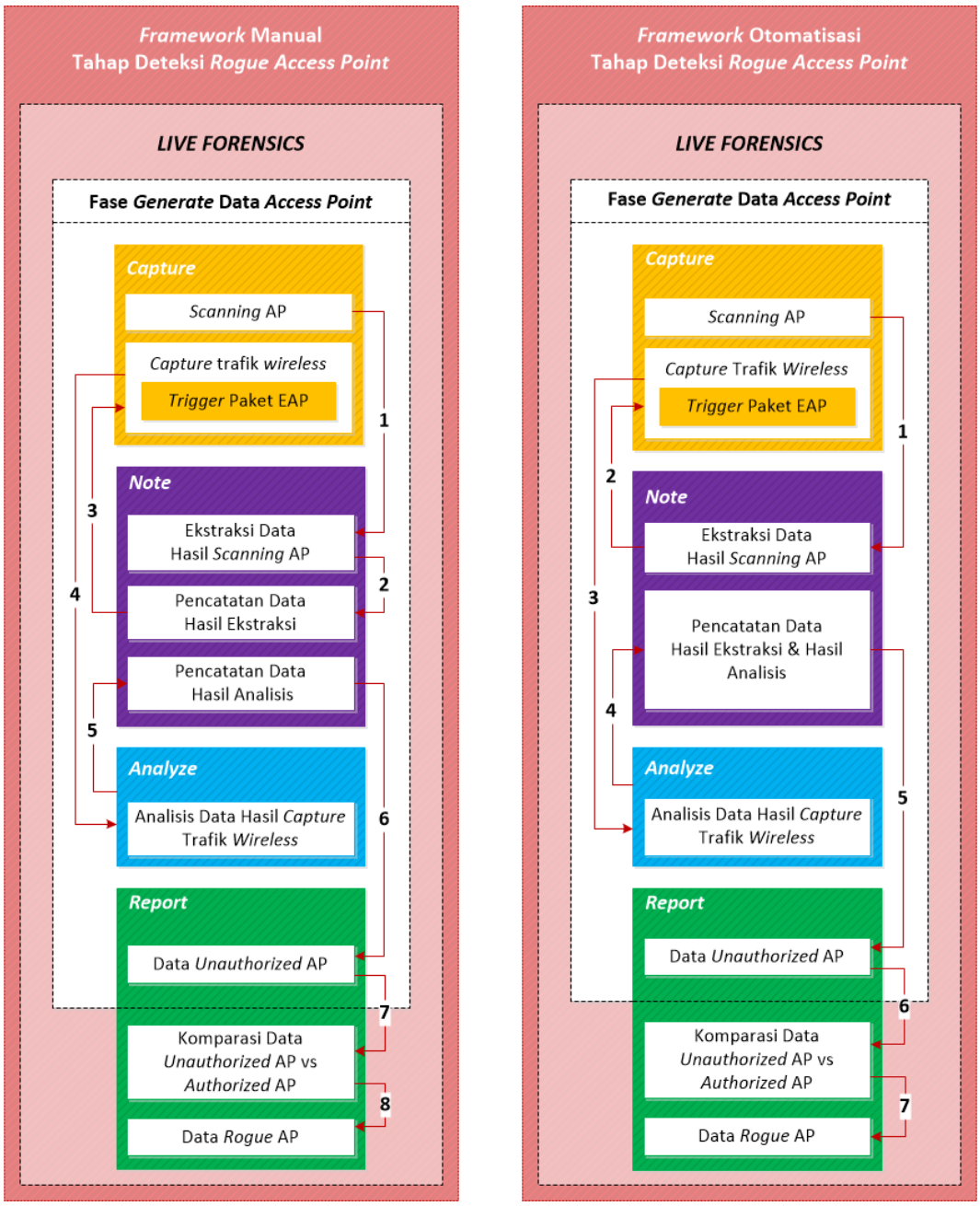
Percobaan *script* deteksi *rogue* AP apabila dijalankan dengan tiga kali iterasi akan menghasilkan kesimpulan data *Access Point* seperti yang ditunjukkan pada Gambar 4.58. Percobaan ini dilakukan untuk mendapatkan hasil *rogue* AP yang lebih valid.

Gambar 4.58 menunjukkan tiga kali iterasi *script* deteksi *rogue* AP ternyata memberikan hasil *rogue* AP yang sama dengan *script* deteksi rogue AP yang dijalankan satu kali (Gambar 4.55), yaitu *rogue* AP dengan SSID Pascasarjana dan MAC *address* 60:E3:27:0B:5D:2F serta *rogue* AP dengan SSID Pascasarjana dan MAC *address* 18:A6:F7:0F:0A:F6.



Gambar 4.58 Hasil deteksi *rogue* AP dengan tiga kali iterasi *script*

Framework tahap deteksi *rogue* AP dapat dikembangkan berdasarkan *framework* fase *generate* data *Access Point* tahap deteksi *rogue* AP (Gambar 4.48) yang dikombinasikan dengan proses *Report* subproses komparasi data dan subproses data *rogue* AP, sehingga menghasilkan sebuah *framework* yaitu *framework live forensics* untuk deteksi *rogue* AP seperti yang ditunjukkan pada Gambar 4.59.



Gambar 4.59 *Framework* tahap deteksi *rogue* AP dengan *live forensics*

4.4.4 Tahap Eliminasi *Rogue Access Point*

Data *rogue Access Point* (*rogue* AP) yang didapatkan dari tahap deteksi *rogue* AP akan diproses ke tahap eliminasi *rogue* AP. Langkah-langkah yang dilakukan dalam tahap eliminasi *rogue* AP ini adalah sebagai berikut:

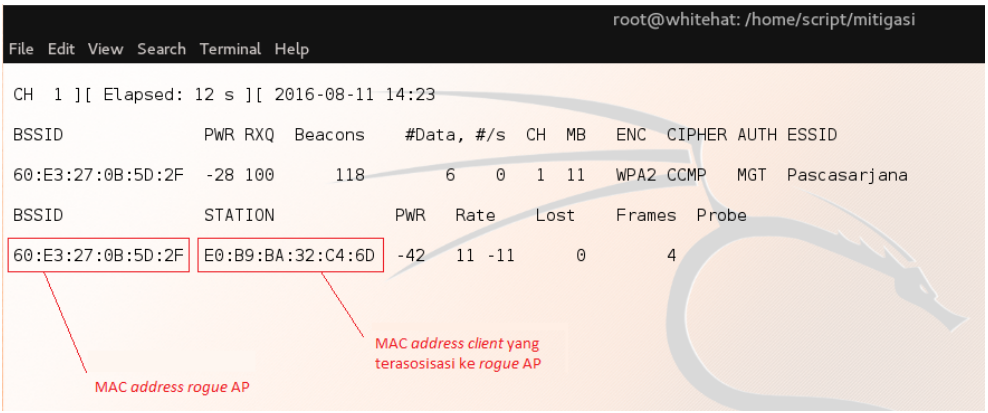
1. **Eliminasi *rogue* AP secara manual**

   a. Administrator membaca data *rogue* AP yang telah terdeteksi seperti pada Gambar 4.55, yaitu sebagai berikut:

   - SSID : Pascasarjana
   - MAC *address* : 60:E3:27:0B:5D:2F
   - *Channel* : 1
   - Kekuatan sinyal : -32 dBm
   - Tipe enkripsi : WPA2
   - Jenis protokol EAP : EAP-PEAP

   b. Dari data *rogue* AP tersebut, diambil parameter MAC *address* dan *channel*-nya.

   c. Administrator melakukan pengecekan seperti pada Gambar 4.60 untuk melihat *client* yang terkoneksi ke *rogue* AP dengan cara melakukan *capture* trafik *wireless* menggunakan Airodump-ng dengan parameter MAC *address* dan *channel* yang telah diambil tadi. Ada atau tidak ada *client*, proses pengamanan (*containment*) *rogue* AP tetap dijalankan.

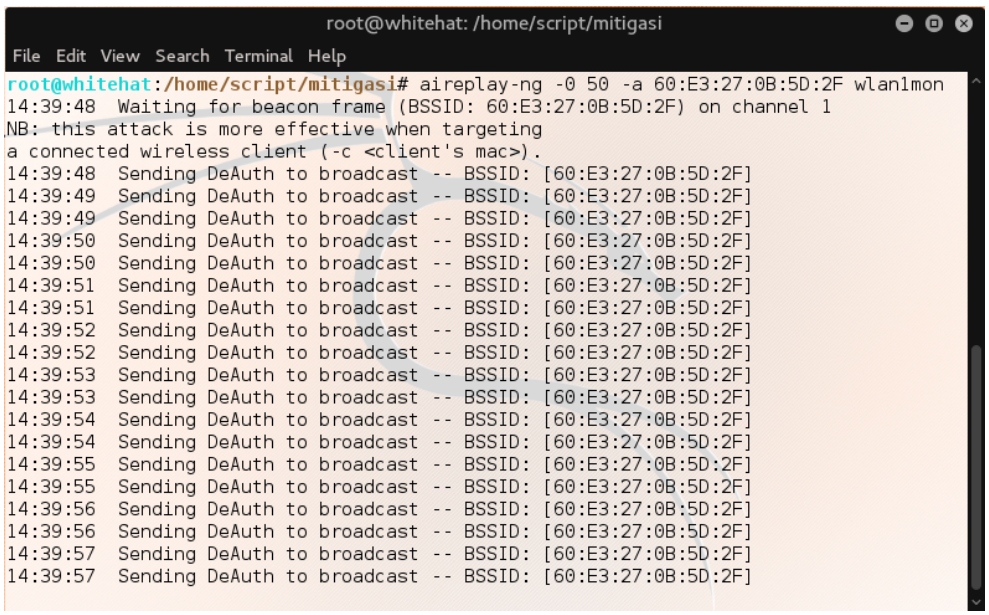


Gambar 4.60 Pengecekan *client rogue* AP

   d. Administrator melakukan pengamanan (*containment*) terhadap *rogue* AP seperti ditunjukkan pada Gambar 4.61 dengan cara mengirimkan paket DoS (*Denial of Service*) terhadap MAC *address rogue* AP menggunakan Aireplay-ng yang bertujuan:

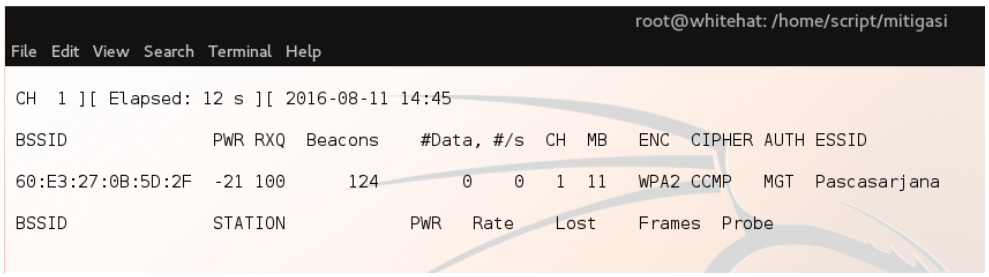
   - Memutuskan koneksi *client* dari *rogue* AP (*deauthentication*).

- Mendisfungsikan *rogue* AP pada saat *containment* berjalan sehingga tidak dapat melayani permintaan *client* yang akan terkoneksi ke *rogue* AP.



Gambar 4.61 *Containment rogue* AP

e. Administrator kembali melakukan pengecekan pada poin (c) seperti yang ditunjukkan pada Gambar 4.62 untuk mengetahui apakah masih ada *client* yang terkoneksi ke *rogue* AP.



Gambar 4.62 Tidak ada *client* yang terkoneksi ke *rogue* AP setelah *containment* dijalankan

f. Jika sudah tidak ada *client* yang terkoneksi ke rogue AP, maka *containment rogue* AP tersebut telah berhasil dijalankan. Namun jika masih ada *client* yang terkoneksi ke *rogue* AP, maka langkah pada poin (d) dan (e) perlu diulang.

Langkah-langkah tahap eliminasi *rogue* AP secara manual ini ditunjukkan pada Gambar 4.68.

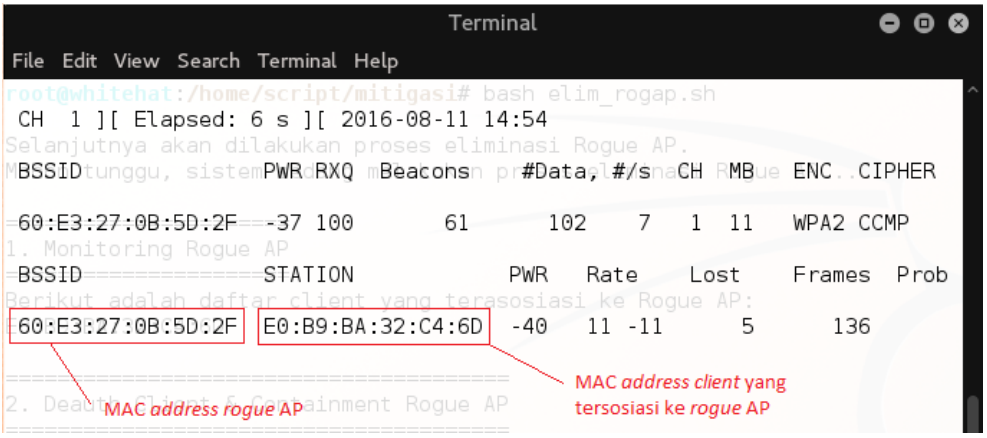
2. Otomatisasi eliminasi *rogue* AP

a. Sistem memiliki data *rogue* AP yang telah terdeteksi seperti pada Gambar 4.55, yaitu sebagai berikut:

- SSID : Pascasarjana
- MAC *address* : 60:E3:27:0B:5D:2F
- *Channel* : 1
- Kekuatan sinyal : -32 dBm
- Tipe enkripsi : WPA2
- Jenis protokol EAP : EAP-PEAP

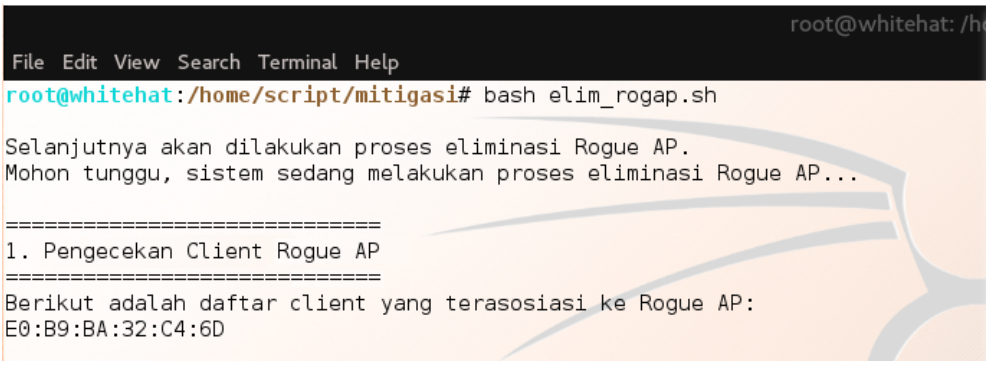
b. Sistem mengambil parameter MAC *address* dan *channel* dari data *rogue* AP.

c. Sistem melakukan pengecekan seperti pada Gambar 4.63 untuk melihat *client* yang terkoneksi ke *rogue* AP dengan cara melakukan *capture* trafik *wireless* dengan Airodump-ng menggunakan parameter MAC *address* dan *channel* yang telah diambil tadi. Ada atau tidak ada *client*, proses pengamanan (*containment*) *rogue* AP tetap dijalankan.



Gambar 4.63 Pengecekan *client rogue* AP oleh sistem

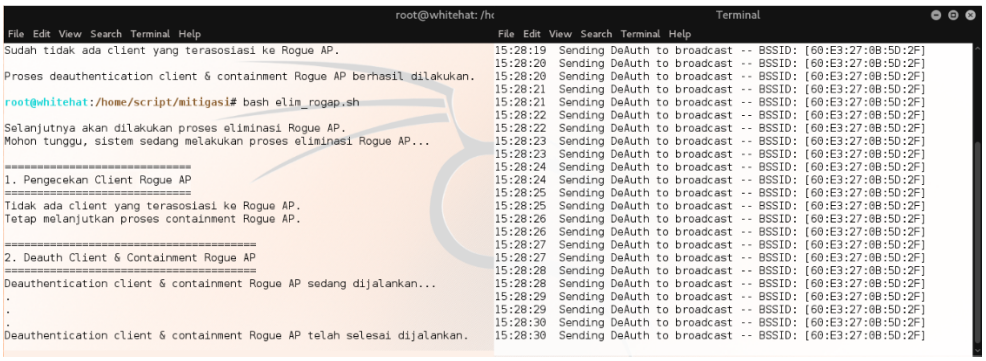
Jika ada *client* yang terkoneksi ke *rogue* AP, maka sistem akan menampilkan data MAC *address client* seperti yang tampak pada Gambar 4.64.



Gambar 4.64 Data *client rogue* AP yang ditampilkan oleh sistem

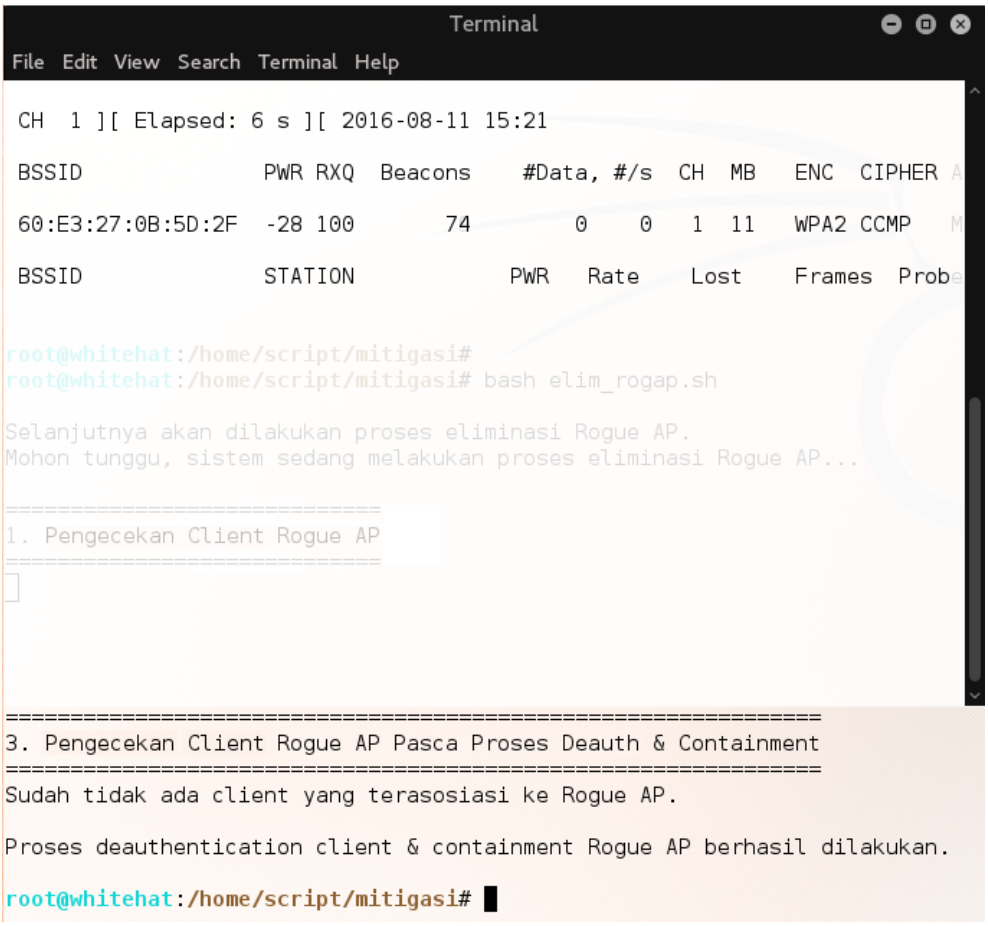
d. Sistem melakukan pengamanan (*containment*) terhadap *rogue* AP seperti ditunjukkan pada Gambar 4.65 dengan cara mengirimkan paket DoS (*Denial of Service*) terhadap MAC *address rogue* AP menggunakan Aireplay-ng yang bertujuan:

- Memutuskan koneksi *client* dari *rogue* AP (*deauthentication*).
- Mendisfungsikan *rogue* AP pada saat *containment* berjalan sehingga tidak dapat melayani permintaan *client* yang akan terkoneksi ke *rogue* AP.



Gambar 4.65 *Containment rogue* AP oleh sistem

g. Sistem kembali melakukan pengecekan pada poin (c) seperti yang ditunjukkan pada Gambar 4.66 untuk mengetahui apakah masih ada *client* yang terkoneksi ke *rogue* AP.



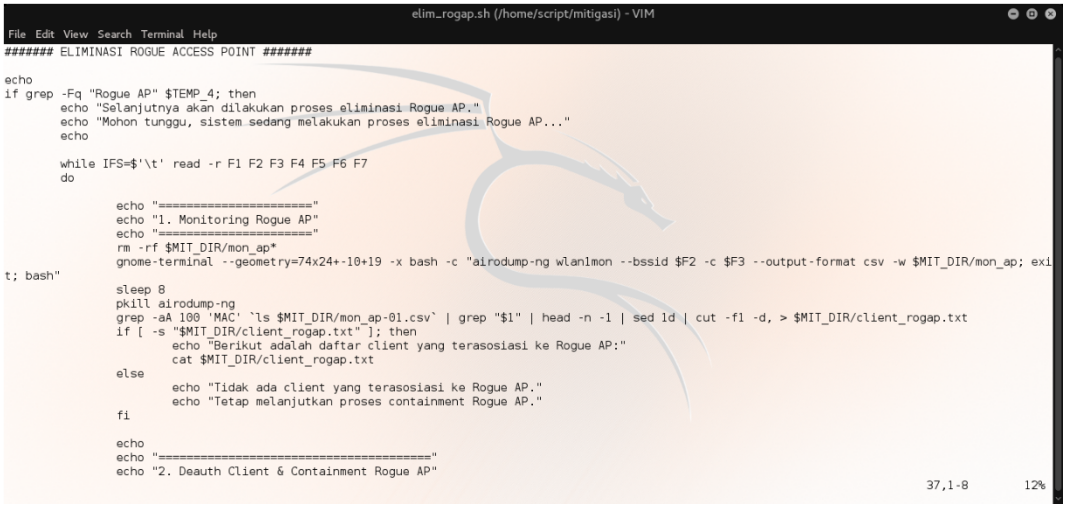
Gambar 4.66 *Containment rogue* AP selesai dijalankan

e. Jika sudah tidak ada *client* yang terkoneksi ke rogue AP, maka *containment rogue* AP tersebut telah berhasil dijalankan. Namun jika masih ada *client* yang terkoneksi ke *rogue* AP, maka langkah pada poin (d) dan (e) perlu diulang.

Langkah-langkah otomatisasi tahap eliminasi *rogue* AP ini ditunjukkan pada Gambar 4.69, sedangkan algoritma otomatisasi eliminasi *rogue* AP dapat dijabarkan sebagai berikut:

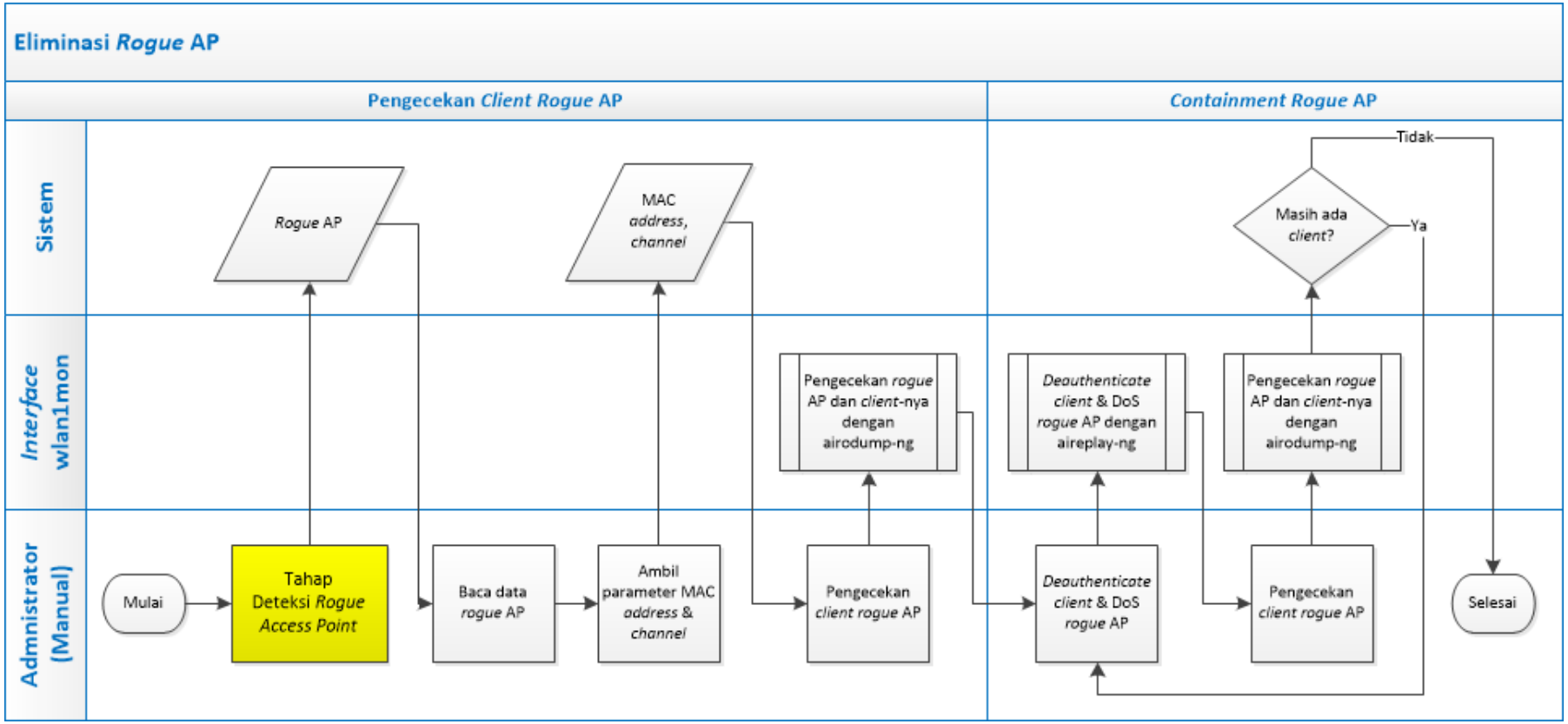
```
Step 1  : Start
Step 2  : While (read Rogue AP);
              Do {read variables BSSID and CHANNEL}.
Step 3  : Rogue AP's client checking using variables BSSID and
          CHANNEL.
Step 4  : Eliminate Rogue AP using variable BSSID.
Step 5  : Repeat step 3.
          If (any client presents) then {repeat step 4};
              else {done}.
Step 6  : Finish
```

Gambar 4.67 menunjukkan potongan *script* otomatisasi tahap eliminasi *rogue* AP. *Script* otomatisasi tahap eliminasi *rogue* AP lebih lengkap terdapat pada Lampiran 6.

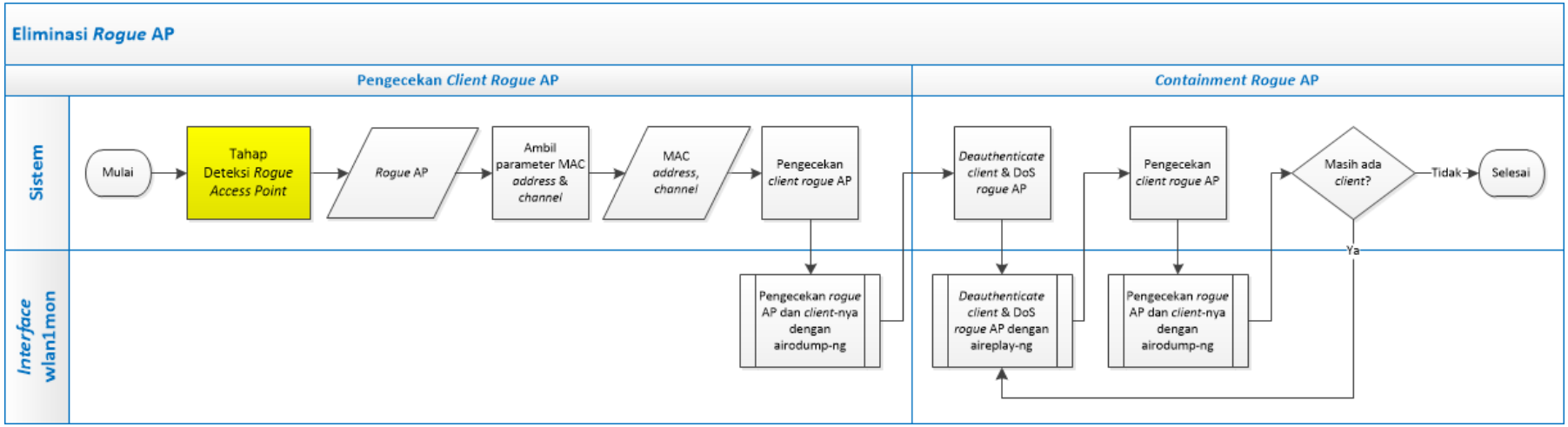


Gambar 4.67 Potongan *script* otomatisasi tahap eliminasi *rogue* AP

Berdasarkan langkah-langkah pada tahap eliminasi *rogue* AP secara manual dan terotomatisasi yang telah dijabarkan tersebut, dapat diperoleh kesimpulan bahwa langkah pengecekan *client* dan *containment rogue* AP dengan cara manual oleh administrator dapat dikerjakan oleh sistem melalui otomatisasi tahap eliminasi *rogue* AP.

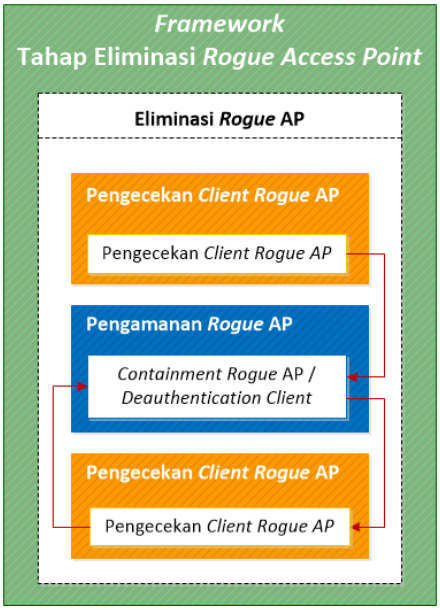


Gambar 4.68 *Flowchart* tahap eliminasi *rogue* AP secara manual



Gambar 4.69 *Flowchart* otomatisasi tahap eliminasi *rogue* AP

Berdasarkan Gambar 4.68 dan Gambar 4.69 mengenai *flowchart* tahap eliminasi *rogue* AP, dikembangkanlah sebuah *framework* yaitu *framework* tahap eliminasi *rogue* AP. Framework tahap eliminasi *rogue* AP ini ditunjukkan pada Gambar 4.70.



Gambar 4.70 *Framework* tahap eliminasi *rogue* AP

4.4.5 *Framework* Mitigasi Kasus *Rogue Access Point*

Mitigasi kasus *rogue Access Point* (*rogue* AP) pada WLAN berbasis IEEE 802.1X terdiri dari tiga tahap yang masing-masing tahapannya memiliki proses-proses tersendiri. Tahapan mitigasi kasus yang dimaksud adalah sebagai berikut:

1. Tahap populasi data *authorized Access Point* (*authorized* AP)
   a. Meliputi:
      - Manipulasi data *Access Point*.
        Terjadi jika ada perubahan (pemasangan baru atau pelepasan) *authorized* AP di dalam jaringan.
      - Fase *generate* data *Access Point*.
        Dilakukan hanya jika ada pemasangan *authorized* AP yang baru di dalam jaringan.

b. Output yang dihasilkan dari tahap populasi data *authorized* AP adalah berupa data *authorized* AP ter-*update* yang meliputi parameter SSID, MAC *address*, *channel*, *power*, tipe enkripsi, dan jenis protokol EAP.

2. Tahap deteksi *rogue* AP

   Menggunakan metode *live forensics* yang meliputi:

   a. Fase *generate* data *Access Point*, terdiri dari proses:

      - *Capture*, terdiri dari subproses:
        - ❖ *Scanning Access Point*
        - ❖ *Capture* trafik *wireless*
      - *Note*, terdiri dari subproses:
        - ❖ Ekstraksi data hasil *scanning Access Point*

          Hasilnya adalah berupa parameter SSID, MAC *address*, *channel*, *power*, dan tipe enkripsi.
        - ❖ Pencatatan data hasil ekstraksi dan hasil analisis *capture* trafik *wireless*.
      - *Analyze*

        Analisis file hasil *capture* trafik *wireless* yang menghasilkan parameter jenis protokol EAP.
      - *Report*

        Menghasilkan data *unauthorized Access Point* (*unauthorized* AP) yang meliputi parameter SSID, MAC *address*, *channel*, *power*, tipe enkripsi, dan jenis protokol EAP.

   b. Lanjutan proses *Report* dengan subproses komparasi data antara data *unauthorized* AP dan data *authorized* AP untuk menentukan *rogue* AP.

3. Tahap eliminasi *rogue* AP

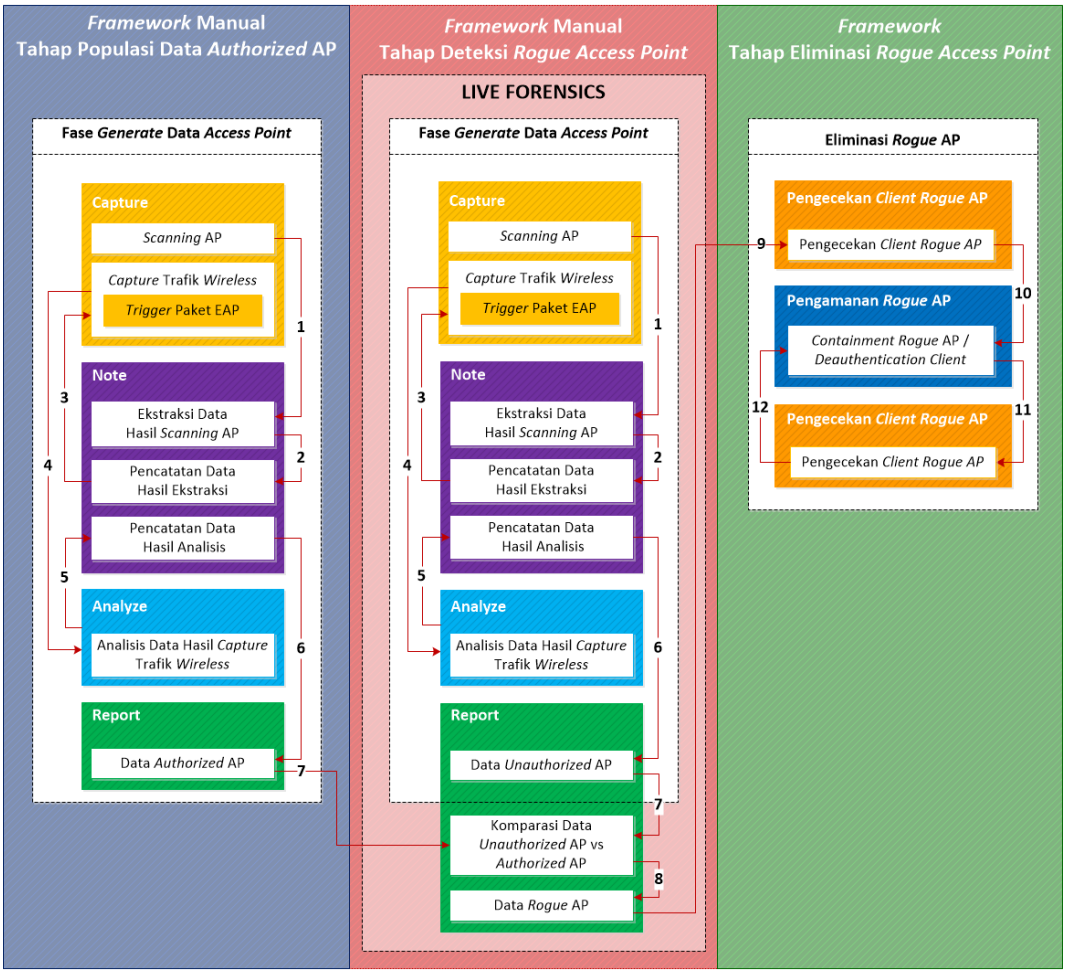
   Meliputi proses:

   a. Pengecekan *client rogue* AP

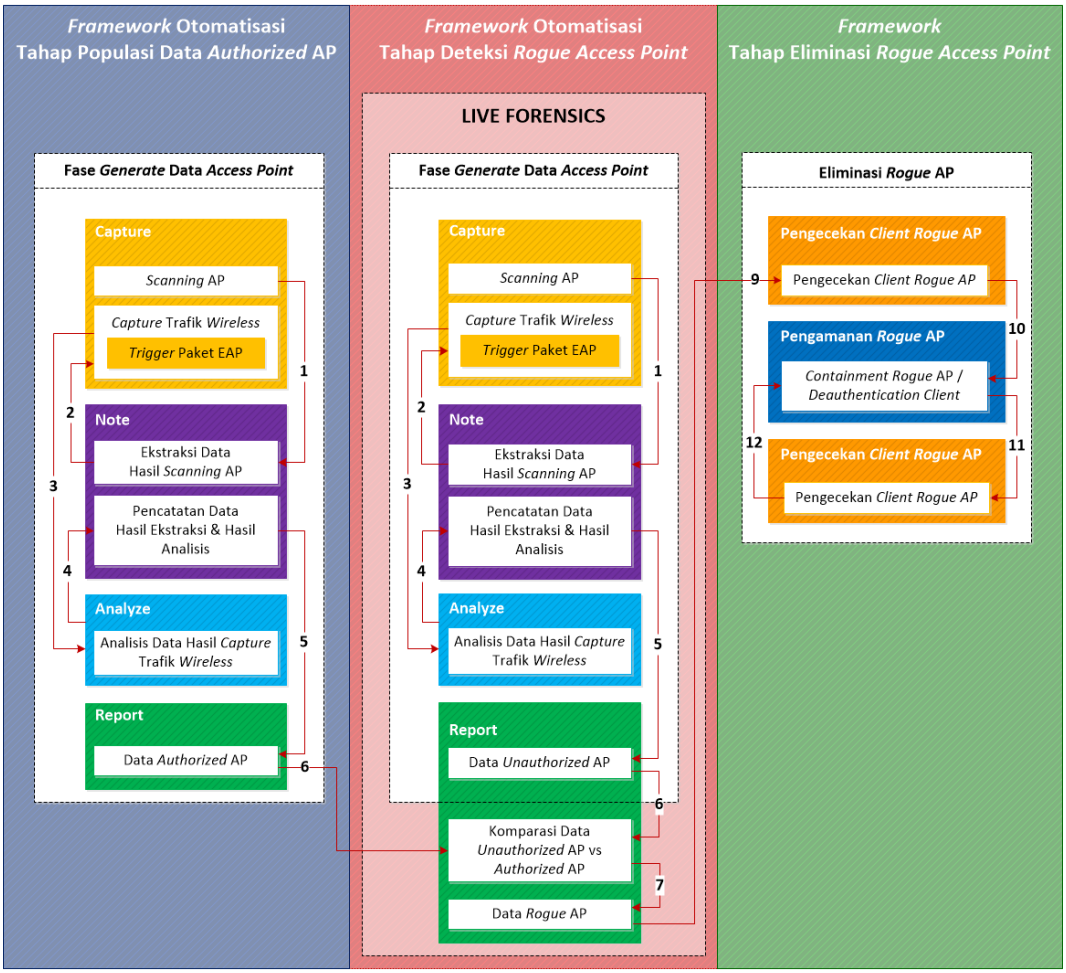
   b. Pengamanan (*containment*) *rogue* AP

Berdasarkan tahap-tahap mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X yang telah dijabarkan tersebut, dikembangkanlah *framework* mitigasi kasus *rogue* AP baik secara manual yang ditunjukkan pada Gambar 4.71 maupun terotomatisasi yang ditunjukkan pada Gambar 4.72.

Perbedaan *framework* pada fase *generate* data *Access Point* baik pada tahap populasi data *authorized* AP maupun tahap deteksi *rogue* AP secara manual (Gambar 4.71) dan terotomatisasi (Gambar 4.72) terdapat pada proses Note subproses pencatatan data. Cara manual terdiri dari enam langkah karena terjadi dua kali subproses pencatatan data. Sedangkan cara otomatisasi terdiri dari lima langkah karena subproses pencatatan data dilakukan satu kali secara bersamaan. Fase *generate* data *Access Point* pada tahap populasi data *authorized* AP menghasilkan data *authorized* AP, sedangkan fase *generate* data *Access Point* pada tahap deteksi *rogue* AP menghasilkan data *unauthorized* AP serta belum mencakup proses *Report* subproses komparasi data dan data *rogue* AP.



Gambar 4.71 *Framework* mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X secara manual



Gambar 4.72 *Framework* otomatisasi mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X

Perlu digarisbawahi bahwa keseluruhan *framework* mitigasi kasus *rogue* AP pada jaringan WLAN berbasis IEEE 802.1X ini hanya mencakup kasus *rogue* AP berbasis *software*. Penelitian ini tidak menggunakan *rogue* AP berbasis *hardware* dikarenakan perlunya akses fisik ke jaringan. Dipilihnya *rogue* AP berbasis *software* adalah karena *rogue* AP berbasis *software* dapat digunakan dalam simulasi kasus secara lebih fleksibel, dapat berpindah-pindah tempat, dan utamanya karena ketiadaan akses fisik ke jaringan.

4.4.6 Validasi Sistem

Sistem yang digunakan untuk mitigasi kasus *rogue Access Point* (*rogue* AP) pada WLAN berbasis IEEE 802.1X perlu divalidasi pada tahap deteksi *rogue* AP untuk menghindari terjadinya kesalahan dalam penentuan *Access Point* mana yang merupakan *rogue* AP.

4.4.6.1 Data *Client Rogue* AP

Pada bagian 4.3 mengenai simulasi kasus *rogue* AP, telah dilakukan simulasi *client* yang terkoneksi ke *rogue* AP mendapatkan detail IP *address* (Gambar 4.16) sebagai berikut:

1. IP *address* : 192.168.1.183 → dari rentang IP *address* 192.168.1.2 – 254
2. *Netmask* : 255.255.255.0
3. Gateway : 192.168.1.1
4. DNS : 192.168.1.1

Dilakukan juga simulasi proses "traceroute" (Gambar 4.17) ke salah satu situs web agar dapat mengetahui rute yang dilalui untuk mencapai situs web tersebut. Rute yang dilalui melewati IP *address gateway* yaitu 192.168.1.1.

4.4.6.2 Data *Rogue* AP

Pembahasan mengenai tahap deteksi *rogue* AP pada bagian 4.4.3 menghasilkan data *rogue* AP (Gambar 4.55) sebagai berikut:

1. SSID : Pascasarjana
2. MAC *address* : 60:E3:27:0B:5D:2F
3. *Channel* : 1
4. Kekuatan sinyal : -32 dBm
5. Tipe enkripsi : WPA2
6. Jenis protokol EAP : EAP-PEAP

Data tersebut akan dijadikan dasar untuk simulasi validasi sistem dengan mengambil parameter SSID dan MAC *address*.

4.4.6.3 Simulasi Validasi Sistem

Simulasi validasi sistem dilakukan untuk membuktikan *Access Point* yang terdeteksi sebagai *rogue* AP apakah benar memang sebuah *rogue* AP. Cara yang dilakukan adalah dengan melakukan koneksi ke *rogue* AP menggunakan *tool* Wpa_supplicant dengan parameter SSID dan MAC *address* dari *rogue* AP. *Client* yang terkoneksi ke *rogue* AP akan mendapatkan IP *address* yang berada pada rentang 192.168.1.2 – 254. Berikut adalah langkah-langkah yang dilakukan:

1. **Persiapan file konfigurasi Wpa_supplicant**

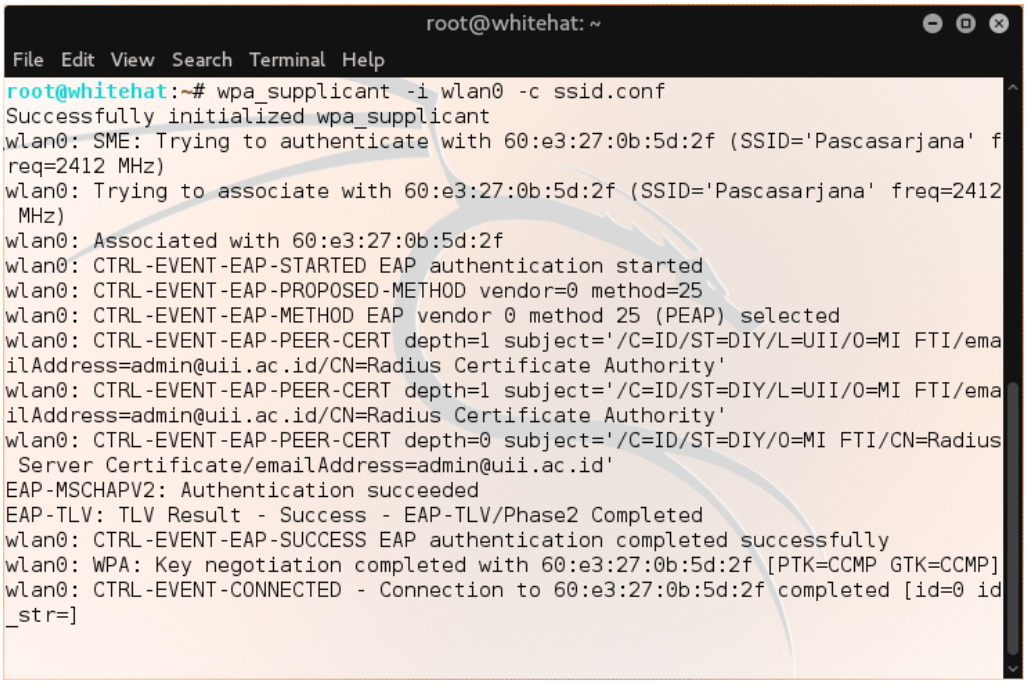
   File konfigurasi Wpa_supplicant bernama "ssid.conf" dibuat menggunakan parameter SSID "Pascasarjana" dan MAC address 60:E3:27:0B:5D:2F seperti yang ditunjukkan pada Gambar 4.73.



Gambar 4.73 File konfigurasi Wpa_supplicant untuk koneksi ke *rogue* AP

2. **Koneksi ke *rogue* AP**

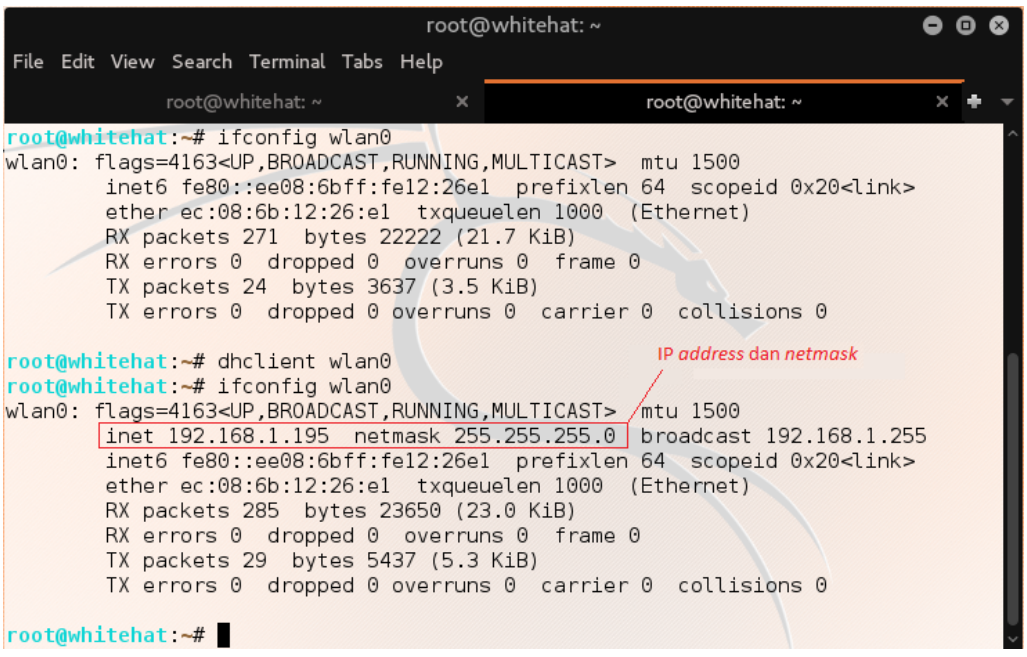
   Koneksi ke *rogue* AP dilakukan menggunakan *tool* Wpa_supplicant pada *interface* wlan0 dengan file konfigurasi "ssid.conf" yang telah dibuat pada poin (1). Koneksi ke *rogue* AP dengan *tool* Wpa_supplicant ditunjukkan pada Gambar 4.74. Setelah terkoneksi ke *rogue* AP yang ditandai dengan munculnya output "wlan0: CTRL-EVENT-CONNECTED - Connection to 60:e3:27:0b:5d:2f completed [id=0 id_str=]", maka perlu dijalankan perintah untuk mendapatkan IP *address*.



Gambar 4.74 Koneksi ke *rogue* AP dengan *tool* Wpa_supplicant

3. IP *address client*

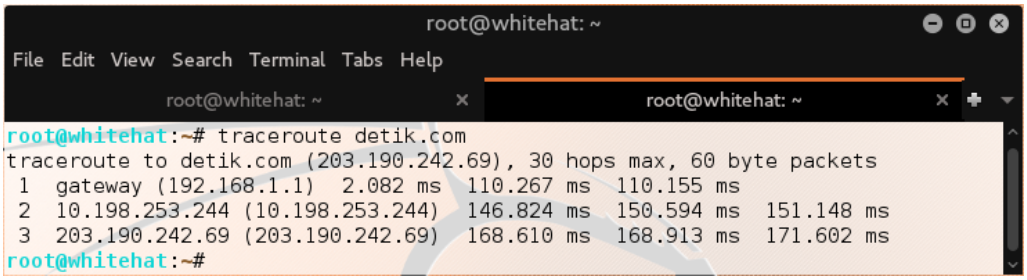
Cara mendapatkan IP *address* untuk *client* adalah dengan mengeksekusi perintah Dhclient seperti yang ditunjukkan pada Gambar 4.75.



Gambar 4.75 IP *address* yang didapatkan oleh *client rogue* AP

4. *Traceroute*

Dilakukan simulasi proses "traceroute" ke salah satu situs web seperti pada Gambar 4.76 agar dapat mengetahui rute yang dilalui untuk mencapai situs web tersebut. Rute yang dilalui melewati IP *address gateway* yaitu 192.168.1.1.



Gambar 4.76 *Traceroute* yang dilakukan oleh *client rogue* AP

Jika dilihat dari rute yang dilalui oleh *client* untuk menuju sebuah situs web melewati IP *address gateway* 192.168.1.1, maka dapat disimpulkan bahwa IP *address gateway* tersebut sama dengan IP *address gateway* dalam simulasi kasus *rogue* AP pada bagian 4.3. Oleh karenanya, dapat disimpulkan pula bahwa Access Point yang terdeteksi sebagai *rogue* AP dalam penelitian ini memang benar adalah sebuah *rogue* AP.

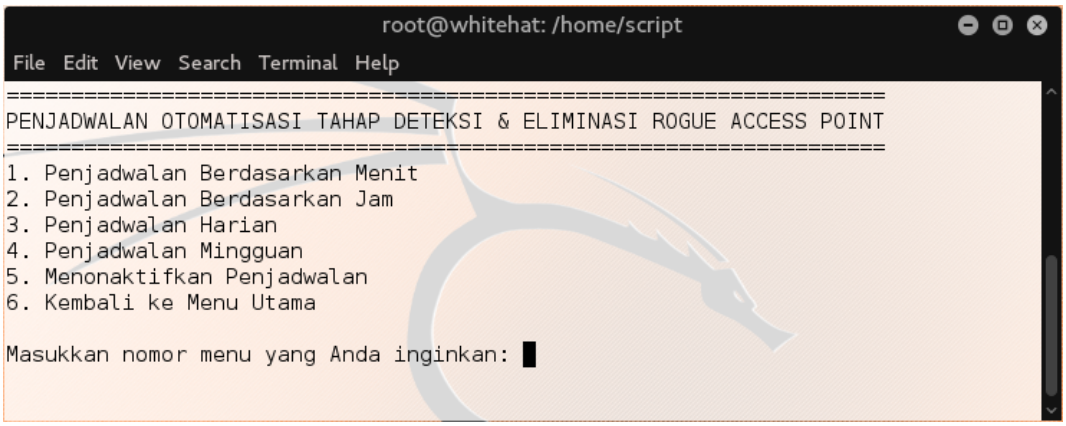
4.4.7 Penjadwalan Otomatisasi Deteksi dan Eliminasi *Rogue Access Point*

Penjadwalan (*task scheduler*/*cron*) otomatisasi tahap deteksi dan eliminasi *rogue Access Point* (*rogue* AP) dilakukan untuk menjalankan *script* deteksi dan eliminasi *rogue* AP dalam interval waktu tertentu secara otomatis, sehingga script dapat dijalankan secara otomatis berdasarkan waktu yang telah ditetapkan dan mempermudah tugas yang dikerjakan oleh administrator jaringan.

Penjadwalan ini dapat dilakukan dengan beberapa pilihan interval waktu seperti yang ditunjukkan pada Gambar 4.77, yaitu:

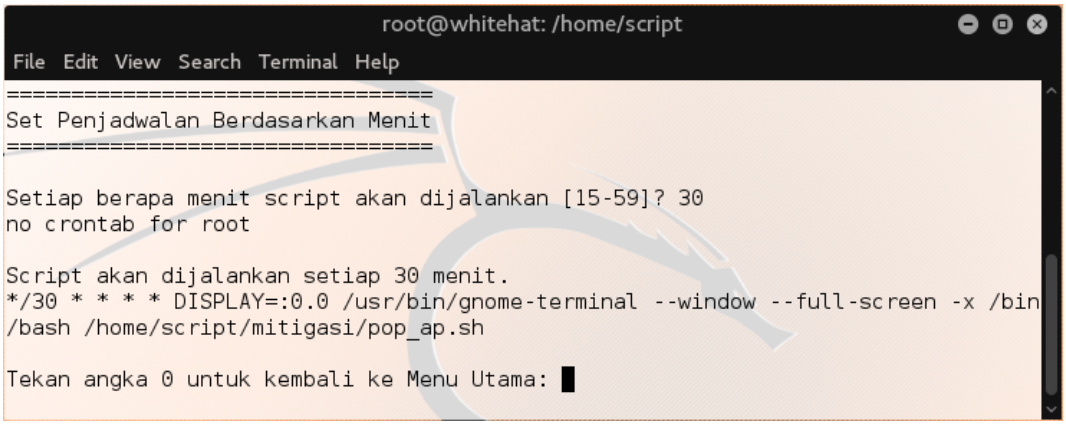
1. Penjadwalan berdasarkan menit
2. Penjadwalan berdasarkan jam
3. Penjadwalan harian
4. Penjadwalan mingguan

5. Menonaktifkan penjadwalan



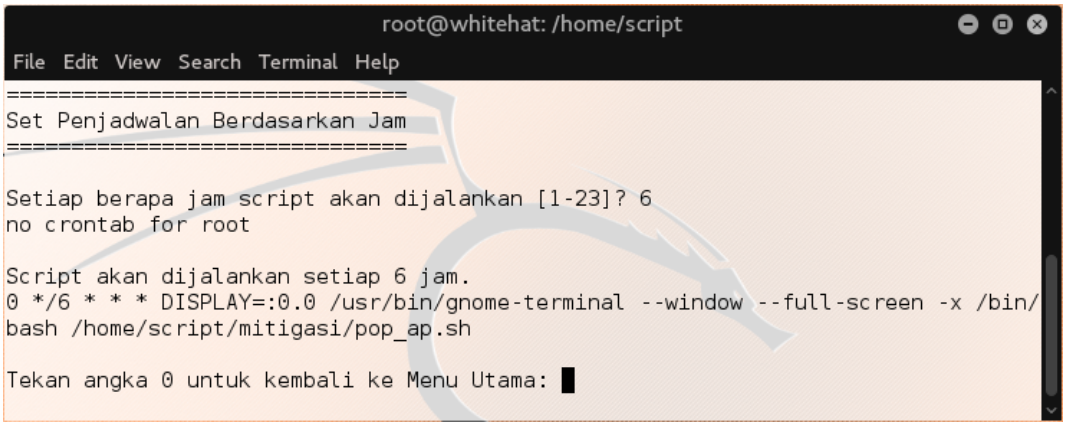
Gambar 4.77 Penjadwalan otomatisasi tahap deteksi dan eliminasi *rogue* AP

Penjadwalan berdasarkan menit dilakukan untuk menjalankan *script* otomatisasi deteksi dan eliminasi *rogue* AP setiap sekian menit bergantung dari *setting* yang dilakukan oleh administrator. Rentang intervalnya di antara 15 – 59 menit. Contoh *setting* penjadwalan berdasarkan menit ditunjukkan pada Gambar 4.78.



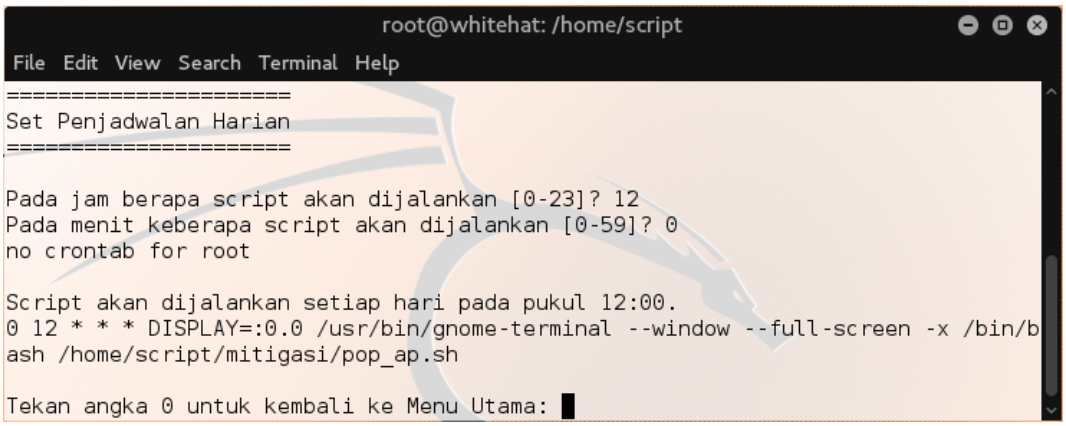
Gambar 4.78 Penjadwalan otomatisasi berdasarkan menit

Penjadwalan berdasarkan jam dilakukan untuk menjalankan *script* otomatisasi deteksi dan eliminasi *rogue* AP setiap sekian jam bergantung dari *setting* yang dilakukan oleh administrator. Rentang intervalnya di antara 1 – 23 jam. Contoh *setting* penjadwalan berdasarkan jam ditunjukkan pada Gambar 4.79.



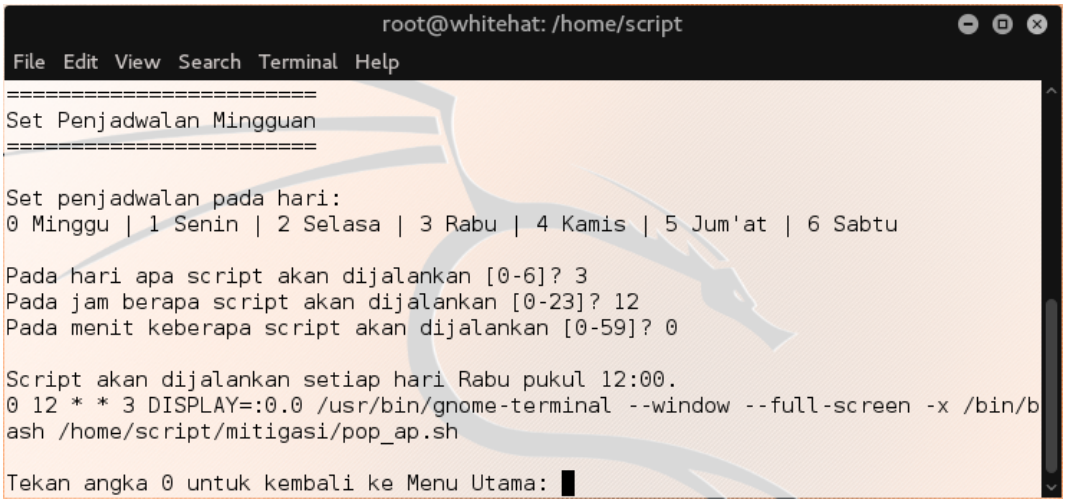
Gambar 4.79 Penjadwalan otomatisasi berdasarkan jam

Penjadwalan harian dilakukan untuk menjalankan *script* otomatisasi deteksi dan eliminasi *rogue* AP setiap hari bergantung dari *setting* yang dilakukan oleh administrator pada pukul berapa (jam dan menit) *script* akan dijalankan. Contoh *setting* penjadwalan harian ditunjukkan pada Gambar 4.80.



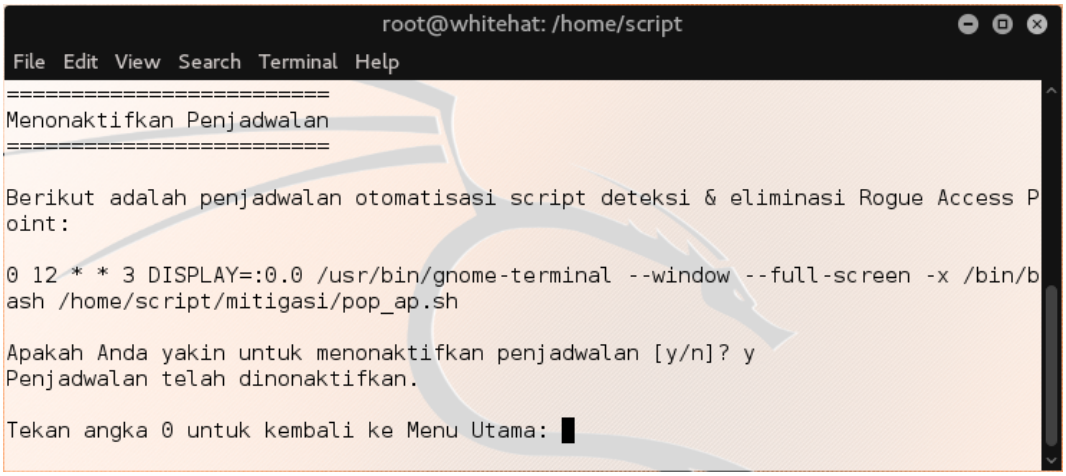
Gambar 4.80 Penjadwalan otomatisasi harian

Penjadwalan mingguan dilakukan untuk menjalankan *script* otomatisasi deteksi dan eliminasi *rogue* AP setiap minggu bergantung dari *setting* yang dilakukan oleh administrator pada hari apa serta pukul berapa (jam dan menit) *script* akan dijalankan. Contoh *setting* penjadwalan mingguan ditunjukkan pada Gambar 4.81.



Gambar 4.81 Penjadwalan otomatisasi mingguan

Penonaktifan penjadwalan dilakukan untuk menghapus *setting* penjadwalan otomatisasi tahap deteksi dan eliminasi *rogue* AP yang telah dibuat. Dengan dinonaktifkannya penjadwalan otomatisasi, jika administrator ingin melakukan deteksi dan eliminasi *rogue* AP, maka administrator tersebut harus mengeksekusi sendiri *script* otomatisasinya. Contoh penonaktifan penjadwalan otomatisasi dapat dilihat pada Gambar 4.82.



Gambar 4.82 Penonaktifan penjadwalan otomatisasi

4.4.8 Analisis Temuan Penelitian

Penelitian mitigasi kasus *rogue Access Point* (*rogue* AP) utamanya kasus *rogue* AP berbasis *software* pada WLAN IEEE 802.1X menghasilkan beberapa temuan sebagai berikut:

1. Otomatisasi dapat memberikan perbedaan terhadap cara manual, yaitu satu langkah saja pada fase *generate* data *Access Point* proses *Note*, namun dengan adanya *script* dan penjadwalan (*task scheduler*/*cron*), maka investigator atau administrator jaringan cukup mengeksekusi perintah untuk *running script* dan tidak perlu mengetikkan berbagai macam perintah secara manual karena dapat dilakukan secara otomatis dari *script* tersebut.
2. Analisis terkait kasus *rogue* AP pada WLAN berbasis IEEE 802.1X telah dapat menyelesaikan problem *rogue* AP berbasis *software*, dibuktikan melalui validasi sistem yang terdapat pada bagian 4.4.6, yaitu dengan membandingkan IP *address default gateway* dari hasil *traceroute client* WLAN berbasis IEEE 802.1X yang *authorized*, dengan IP *address default gateway* hasil *traceroute client* yang terkoneksi ke *rogue* AP.
3. Pada saat dilakukan eksekusi *script* otomatisasi tahap deteksi *rogue* AP sebanyak tiga kali iterasi maka berdampak pada sistem mitigasi yang mengalami perlambatan, khususnya dalam proses *Capture* subproses *scanning Access Point*. Hal ini dimungkinkan karena semakin banyaknya SSID yang di-*scan*, maka akan semakin lama *script* berjalan, sedangkan semakin lama *script* berjalan, maka akan mengakibatkan semakin banyaknya *resource hardware* (*memory* atau CPU *usage*) yang terokupansi. Saran yang dimungkinkan untuk kendala tersebut adalah peningkatan kualitas *hardware* seperti *upgrade memory* atau penggunaan prosesor yang lebih baik, selain itu perlu perbaikan penulisan kode pada *script* agar *script* dapat berjalan lebih efisien.
4. Hasil mitigasi bahwa *rogue* AP tidak dapat bekerja dan tidak dapat melayani permintaan koneksi dari *client* saat dilakukan *deauthentication* menggunakan *tool* Aireplay-ng pada proses pengamanan *rogue* AP. *Rogue* AP mengalami serangan *Denial of Service* (DoS) pada saat *deauthentication* dijalankan, sedangkan *client* yang terlanjur terasosiasi dengan *rogue* AP akan mengalami

putus koneksi ke *rogue* AP namun setelahnya dapat melakukan koneksi ke *Access Point* yang asli.

5. Tahap deteksi *rogue* AP utamanya *rogue* AP berbasis *software* pada WLAN IEEE 802.1X dapat dilakukan dengan menggunakan parameter SSID, MAC *address*, kekuatan sinyal, tipe enkripsi, dan jenis protokol EAP. Apabila dibandingkan dengan adopsi metode deteksi *rogue* AP pada penelitian yang telah disebutkan di bagian 3.4.2 yang menggunakan parameter SSID, kekuatan sinyal, IP *address Access Point*, *authentication type*, *channel*, dan *frequency*, maka parameter penelitian telah dapat disederhanakan, terutama pada penggunaan parameter IP *address Access Point* yang ada pada Layer 3 (*Network*). Penelitian ini tidak memerlukan penggunaan parameter IP *address* dalam mendeteksi *rogue* AP, sehingga metode deteksi *rogue* AP pada penelitian ini cukup pada Layer 2 (*Data Link*) saja.

Rangkuman temuan penelitian dapat dilihat pada Tabel 4.7.

Tabel 4.7 Rangkuman temuan penelitian

No.	Temuan Penelitian	Hasil
1	Otomatisasi vs manual	Satu langkah pada fase *generate* data *Access Point* proses Note.
2	Analisis kasus *rogue* AP	Problem *rogue* AP berbasis *software* pada WLAN berbasis IEEE 802.1X terselesaikan, dibuktikan melalui validasi sistem (bagian 4.4.6).
3	Performa *script* deteksi *rogue* AP pada tiga kali iterasi	Sistem mitigasi mengalami perlambatan proses *Capture* subproses *scanning Access Point*.
4	Hasil mitigasi	*Rogue* AP mengalami disfungsi dan *client rogue* AP putus koneksi ke *rogue* AP.
5	Parameter pengujian	Parameter SSID, MAC *address*, kekuatan sinyal, tipe enkripsi, dan jenis protokol EAP yang bekerja pada Layer 2 (*Data Link*).

Kajian lebih mendalam diterapkan terhadap penelitian-penelitian yang telah dibahas pada bagian 1.6 mengenai *literature review*, dan ditemukan dua penelitian yang menggunakan *rogue* AP berbasis *software* yaitu penelitian Wang (2013) dan Nivangune et al. (2015). Beberapa penelitian menggunakan *rogue* AP berbasis *hardware*, yaitu penelitian Shivaraj et al. (2008), Jagtap dan Honwadkar (2010), Sasane dan Pathan (2011), Thakur dan Bodhe (2013), serta Jadhav et al. (2014). Penelitian lainnya diasumsikan dapat menangani *rogue* AP berbasis *hardware* dan *software* karena tidak menyatakan secara eksplisit jenis *rogue* AP yang digunakan, yaitu penelitian Nikbakhsh et al. (2012), Vanjale dan Mane (2014), serta Chougule et al. (2015).

Paremeter pengujian dalam penelitian ini paling mendekati parameter pengujian dalam penelitian:

1. Thakur dan Bodhe (2013) yang mengusulkan algoritma RAPD untuk deteksi *rogue* AP dengan cara melakukan *sniffing* kemudian membandingkan data hasil *sniffing* dengan data *auhorized* AP berdasarkan SSID dan MAC *address*-nya. Kemudian untuk *Access Point* yang dicurigai sebagai *rogue* AP, diperlukan pemeriksaan terhadap parameter kekuatan sinyal dan *channel* yang digunakan.
2. Vanjale dan Mane (2014) yang mengusulkan deteksi dan eliminasi *rogue* AP dalam tiga tahap, yaitu tahap *learning mode* (pembuatan daftar *authorized* AP), tahap *detection mode* menggunakan parameter SSID, MAC *address* dan nilai *Received Signal Strength* (RSS) untuk mendapatkan *rogue* AP, dan tahap *prevention mode* (implementasi aturan *blocking* pada *rogue* AP).
3. Chougule, Vanjale, dan Mane (2015) yang mengusulkan teknik deteksi *rogue* AP menggunakan berbagai macam parameter, seperti SSID, MAC *address*, *authentication type*, *channel*, *frequency*, dan *signal*.

Tabel 4.8 menunjukkan perbandingan jenis *rogue* AP dan parameter pegujian yang digunakan dalam penelitian.

Tabel 4.8 Perbandingan jenis *rogue* AP dan parameter penelitian

No.	Penelitian	Jenis *Rogue* AP	Parameter Penelitian
1	Thakur dan Bodhe (2013)	*Hardware*	SSID, MAC *address*, kekuatan sinyal, *channel*.
2	Vanjale dan Mane (2014)	*Hardware* & *software*	SSID, MAC *address*, nilai RSS.
3	Chougule, Vanjale, dan Mane (2015)	*Hardware* & *software*	SSID, MAC *address*, *authentication type*, *channel*, *frequency*, *signal*.

Ketiga penelitian tersebut dapat menangani *rogue* AP berbasis *hardware*. Apabila dibandingkan dengan parameter pengujian dalam penelitian ini, yaitu SSID, MAC address, *channel*, kekuatan sinyal, tipe enkripsi, dan jenis protokol EAP, maka dapat dikatakan bahwa hasil pengujian penelitian ini yang menggunakan *rogue* AP berbasis *software* cukup linear dengan beberapa penelitian terdahulu yang menggunakan *rogue* AP berbasis *hardware*. Perbedaan mendasarnya adalah pada parameter jenis protokol EAP karena memang pengujian penelitian ini dilakukan di lingkungan WLAN berbasis IEEE 802.1X di mana diperlukan data berupa jenis protokol EAP yang digunakan dalam jaringan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan penelitian, maka dapat ditentukan kesimpulan sebagai berikut:

1. Tahapan mitigasi kasus *rogue Access Point* (*rogue* AP) pada WLAN berbasis IEEE 802.1X meliputi tahap populasi data *authorized Access Point* (*authorized* AP), tahap deteksi *rogue* AP, dan tahap eliminasi *rogue* AP.
2. *Framework* otomatisasi mitigasi kasus *rogue* AP didapatkan melalui perancangan *framework* mitigasi cara manual kemudian dilanjutkan dengan perancangan algoritma otomatisasi, sehingga dapat diketahui proses-proses dalam tahapan mitigasi yang akan diotomatisasi, yaitu fase *generate* data *Access Point* pada tahap populasi data *authorized* AP, proses *live forensics* pada tahap deteksi *rogue* AP, dan proses pengamanan *rogue* AP dengan *deauthentication* pada tahap eliminasi *rogue* AP. Khusus alur proses tahap deteksi *rogue* AP dengan *live forensics* adalah proses ***Capture*** (*scanning Access Point*), ***Note*** (ekstraksi data hasil *scanning Access Point*), ***Capture*** (*capture* trafik *wireless*), ***Analyze*** (analisis file hasil *capture*), ***Note*** (pencatatan data hasil ekstraksi dan hasil analisis), dan ***Report***.
3. Implementasi otomatisasi langkah mitigasi kasus *rogue* AP pada WLAN berbasis IEEE 802.1X telah dapat dilakukan dengan penulisan dan eksekusi *script* otomatisasi yang memuat langkah-langkah fase *generate* data *Access Point* pada tahap populasi data *authorized* AP, proses *live forensics* pada tahap deteksi *rogue* AP, dan proses pengamanan *rogue* AP dengan *deauthentication* pada tahap eliminasi *rogue* AP.
4. Mitigasi *rogue* AP berdampak *rogue* AP tidak dapat melayani permintaan koneksi dari *client* saat *deauthentication* dijalankan pada tahap eliminasi *rogue* AP karena *rogue* AP mengalami serangan *Denial of Service* (DoS), sedangkan

client yang terlanjur terasosiasi dengan *rogue* AP akan mengalami putus koneksi ke *rogue* AP namun setelahnya dapat melakukan koneksi ke *Access Point* yang asli.

5.2 Saran

Beberapa saran yang mungkin berguna untuk penelitian selanjutnya antara lain sebagai berikut:

1. Penggunaan *hardware based rogue* AP dalam simulasi kasus *rogue* AP untuk meneliti parameter *Round Trip Time* (RTT).
2. Pengembangan aplikasi berbasis GUI (*Graphical User Interface*) yang memberikan kemudahan bagi pengguna (*user friendly*) untuk mengakomodasi personil investigator maupun administrator jaringan yang kurang familiar dengan *shell script*.
3. Penelitian ini tidak melakukan *monitoring rogue* AP sepanjang waktu, namun hanya dilakukan saat *script* otomatisasi sedang berjalan saja. Penelitian berikutnya mungkin dapat dilakukan *monitoring rogue* AP setiap saat atau sepanjang waktu.
4. Penelitian mengenai kasus *rogue Access Point* (*rogue* AP) yang dikombinasikan dengan MAC *address spoofing* yang meniru MAC *address Access Point* asli, sehingga titik berat deteksi dan analisisnya juga ada pada MAC *spoofing*-nya.

DAFTAR PUSTAKA

Adelstein, F. (2006). Diagnosing your system without killing it first. *Communications of the ACM*, *49*(2), 63–66. https://doi.org/10.1145/1113034.1113070

Alamanni, M. (2015). *Kali linux wireless penetration testing essentials*. Birmingham: PACKT Publishing.

APJII. (2015). *Profil pengguna internet indonesia 2014*. Jakarta. Retrieved from http://www.apjii.or.id/read/content/Statistik/305/profil-pengguna-internet-indonesia-2014.html

Chaskar, H. (2010). *Conquering the minefield of soft rogue aps in the enterprise*. Retrieved from http://airtightnetworks.com/fileadmin/pdf/whitepaper/Conquering-the-Minefield-of-Soft-Rogue-APs.pdf

Chen, J.-C., & Wang, Y.-P. (2005). Extensible authentication protocol (eap) and ieee 802.1x: Tutorial and empirical experience. *IEEE Communications Magazine*, *43*(12), 26–32. https://doi.org/10.1109/MCOM.2005.1561920

Chougule, A. A., Vanjale, S. B., & Mane, P. B. (2015). Detection and prevention of rogue access point in the 802.11 using various parameters. *International Journal of Advanced Research in Computer Science and Software Engineering*, *5*(5), 1723–1727. Retrieved from http://www.ijarcsse.com/docs/papers/Volume_5/5_May2015/V5I5-0243.pdf

Cisco. (2007). Cisco protected extensible authentication protocol. Retrieved June 12, 2016, from http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-1200-series/prod_qas0900aecd801764fa.html

Coleman, D. D., Westcott, D. A., Harkins, B. E., & Jackman, S. M. (2011). *Cwsp certified wireless security professional official study guide*. John Wiley & Sons. Retrieved from https://books.google.co.id/books?id=0ZWLn57EdpsC&pg=PA140&lpg=PA140&dq=figure+4.24+generic+eap+exchange&source=bl&ots=CAlraPb1Pf&sig=mCZp5T4RIZLkOUX5Y0XOh9cDgUI&hl=id&sa=X&redir_esc=y#v=onepage&q=figure 4.24 generic eap exchange&f=false

EC-Council. (2010). *Investigating wireless networks and devices* (Vol. 53). New York: CENGAGE Learning.

Faqih, A., Rochim, A. F., & Isnanto, R. R. (2012). *Sistem manajemen hotspot berbasis kuota waktu dan paket data*. Diponegoro University, Semarang, Indonesia. Retrieved from http://eprints.undip.ac.id/32541/

Geier, J. (2008). *Implementing 802.1x security solutions for wired and wireless networks*. Indianapolis: Wiley Publishing, Inc.

Han, H., Sheng, B., Tan, C. C., Li, Q., & Lu, S. (2011). A Timing-Based Scheme for Rogue AP Detection. *IEEE Transactions on Parallel and Distributed Systems*, *22*(11), 1912–1925. Retrieved from https://www.researchgate.net/publication/224256664_A_Timing-Based_Scheme_for_Rogue_AP_Detection

Hoffman, C. (2013). Wpa2, wep, and friends: What's the best way to encrypt your wi-fi? Retrieved November 16, 2015, from http://www.makeuseof.com/tag/wpa2-wep-and-friends-whats-the-best-way-to-encrypt-your-wi-fi/

Jacob, L., Hutchinson, D., & Abawajy, J. (2011). Wi-fi security : Wireless with confidence. In *ASIC 2011: Proceedings of the 4th Australian Security and Intelligence Conference* (pp. 88–96). Perth. https://doi.org/10.1675/1524-4695(2008)31

Jadhav, S., Vanjale, S., & Mane, P. B. (2014). Detecting fake access point into wireless network using clock skews as fingerprinting technique. *International Journal of Advanced Research in Computer Science and Software Engineering*, *4*(7), 26–31. Retrieved from http://www.ijarcsse.com/docs/papers/Volume_4/7_July2014/V4I7-0203.pdf

Jagtap, S., & Honwadkar, K. N. (2010). Rogue access point detection in wlan by analyzing network traffic and behavior. *International Journal of Computer Applications*, *1*(22), 27–29. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.184.7176&rep=rep1&type=pdf

K.N., G., & Chaskar, H. (2009). *All you wanted to know about wifi rogue access points*. Retrieved from http://www.rogueap.com/rogue-ap-docs/RogueAP-FAQ.pdf

Karie, N. M., & Venter, H. S. (2014). Toward a general ontology for digital forensic disciplines. *Journal of Forensic Sciences*, *59*(5), 1231–41. https://doi.org/10.1111/1556-4029.12511

Keyun, R., Carthy, J., & Kechadi, T. (2011). Cloud forensics: An overview. *7th IFIP Conference on Digital Forensics*, (JANUARY), 35–46. Retrieved from https://www.researchgate.net/publication/229021339_Cloud_forensics_An_overview

Kipper, G. (2007). *Wireless crime and forensic investigation*. Auerbach Publications. Retrieved from http://books.google.com/books?id=r6CL4zKtU8cC&dq=Wireless+crime+and+forensic+investigation&lr=&source=gbs_navlinks_s

Kumar, U., Kumar, P., & Gambhir, S. (2014). Analysis and literature review of ieee 802 .1x (authentication) protocols. *International Journal of Engineering*

and Advanced Technology, *3*(5), 163–168. Retrieved from http://www.ijeat.org/attachments/File/v3i5/E3173063514.pdf

Luttgens, J. T., Pepe, M., & Mandia, K. (2014). *Incident response & computer forensics* (3rd ed.). McGraw-Hill Education.

Metageek. (n.d.-a). Understanding rssi.

Metageek. (n.d.-b). Why channels 1, 6 and 11?

Moore, T., Meehan, A., Manes, G., & Shenoi, S. (2005). Using signaling information in telecom network forensics. In M. Pollitt & S. Shenoi (Eds.), *Advances In Digital Forensics* (pp. 177–188). Springer US. https://doi.org/10.1007/0-387-31163-7_15

Networks, I. (2006). *Introduction to 802 .1x for wireless lans* (Vol. 1). Retrieved from http://www.interlinknetworks.com/whitepapers/Intro_802_1X_for_Wireless_LAN.pdf

Nikbakhsh, S., Manaf, A. B. A., Zamani, M., & Janbeglou, M. (2012). A novel approach for rogue access point detection on the client-side. *2012 26th International Conference on Advanced Information Networking and Applications Workshops*, 684–687. https://doi.org/10.1109/WAINA.2012.108

Nivangune, M. K., Vanjale, S. B., & Mane, P. B. (2015). Detecting unauthorized access point in wlan by using ctt. *International Journal of Advanced Research in Computer Science and Software Engineering*, *5*(7), 829–833. Retrieved from http://www.ijarcsse.com/docs/papers/Volume_5/7_July2015/V5I7-0258.pdf

Ozan, O. (2004). *Denial of service attacks on 802.1x security protocol*. Naval Postgraduate School, Monterey, California. Retrieved from http://www.dtic.mil/dtic/tr/fulltext/u2/a422412.pdf

Palmer, G. (2001). A road map for digital forensic research. In *Proceedings of the 2001 Digital Forensics Research Workshop (DFRWS 2004)* (pp. 1–42). https://doi.org/10.1111/j.1365-2656.2005.01025.x

Rafique, M., & Khan, M. N. A. (2013). Exploring static and live digital forensics: Methods, practices and tools. *International Journal of Scientific & Engineering Research*, *4*(10), 1048–1056. Retrieved from http://www.ijser.org/researchpaper%5CExploring-Static-and-Live-Digital-Forensic-Methods-Practices-and-Tools.pdf

Rahman, S., & Khan, M. N. A. (2015). Review of live forensic analysis techniques. *International Journal of Hybrid Information Technology*, *8*(2), 379–388. Retrieved from http://www.sersc.org/journals/IJHIT/vol8_no2_2015/35.pdf

Reyes, A., O'shea, K., Steele, J., Hansen, J. R., Jean, B. R., & Ralph, T. (2007).

Incident response: Live forensics and investigations. In *Cyber Crime Investigations* (pp. 89–109). Elsevier. https://doi.org/10.1016/B978-159749133-4/50000-0

Sankar, K., Balinsky, A., Miller, D., & Sundaralingam, S. (2005). Eap authentication protocols for wlans. Retrieved June 12, 2016, from http://www.ciscopress.com/articles/article.asp?p=369223&seqNum=2

Saruhan, I. H. (2007). *Detecting and preventing rogue devices on the network*. Retrieved from https://www.sans.org/reading-room/whitepapers/detection/detecting-preventing-rogue-devices-network-1866

Sasane, P. G., & Pathan, S. K. (2011). Detecting and eliminating rogue access points in ieee-802.11 wlan - a multi-agent sourcing methodology. *International Journal of Sciences: Basic and Applied Research*, *3*(1), 9–15. Retrieved from http://gssrr.org/index.php?journal=JournalOfBasicAndApplied&page=article&op=download&path[]=944&path[]=1124

Shetty, S., Song, M., & Ma, L. (2007). Rogue access point detection by analyzing network traffic characteristics. *MILCOM 2007 - IEEE Military Communications Conference*, 1–7. https://doi.org/10.1109/MILCOM.2007.4455018

Shivaraj, G., Song, M., & Shetty, S. (2008). A hidden markov model based approach to detect rogue access points. *MILCOM 2008 - IEEE Military Communications Conference*, 1–7. https://doi.org/10.1109/MILCOM.2008.4753358

Siles, R. (2007). Wireless forensics: Tapping the air - part one. Retrieved December 15, 2015, from http://www.symantec.com/connect/articles/wireless-forensics-tapping-air-part-one

Sonawane, S. R., Chavan, S. P., & Ghodeswar, A. A. (2013). Study of different rogue access point detection and prevention techniques in wlan. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(10), 1232–1237. Retrieved from https://www.ijarcsse.com/docs/papers/Volume_3/10_October2013/V3I10-0419.pdf

Sukhija, S., & Gupta, S. (2012). Wireless network security protocols: A comparative study. *International Journal of Emerging Technology and Advanced Engineering*, *2*(1), 357–364. Retrieved from http://www.ijetae.com/files/Volume2Issue1/IJETAE_0112_61.pdf

Thakur, S., & Bodhe, A. (2013). Rapd algorithm: Detection of rogue access point in wireless network. *International Journal of Emerging Technology and Advanced Engineering*, *3*(6), 85–89. Retrieved from http://www.ijetae.com/files/Volume3Issue6/IJETAE_0613_12.pdf

Vanjale, S., & Mane, P. B. (2014). A novel approach for elimination of rogue access point in wireless network. *India Conference (INDICON), 2014 Annual IEEE*, 1–4. https://doi.org/10.1109/INDICON.2014.7030418

Wang, L. (2013). *Detection of man-in-the-middle attacks using physical layer wireless security techniques*. Worcester Polytechnic Institute. Retrieved from https://www.wpi.edu/Pubs/ETD/Available/etd-082713-125108/unrestricted/thesis.pdf

Wikipedia. (2015). Monitor mode. Retrieved May 14, 2016, from https://en.wikipedia.org/wiki/Monitor_mode

LAMPIRAN

Lampiran 1: *Script* Otomatisasi Persiapan *Hardware*

```
####### PENGECEKAN WIRELESS INTERFACE #######

        echo
        echo "Pengecekan wireless interface..."
        ifconfig wlan0 up
        ifconfig wlan1 up

        IFACE=`ifconfig | grep wlan0 | awk '{print $1}' | cut -d: -f1`
        if [ "$IFACE" = "wlan0" ]; then
                echo "1. Interface wlan0 dalam mode Managed siap digunakan."
        else
                echo "1. Interface wlan0 dalam mode Monitor (wlan0mon)."
                echo "   -- Mohon tunggu untuk mengubah interface wlan0mon ke mode
     Managed."
                airmon-ng stop wlan0mon
                echo
                echo "   -- Interface wlan0 dalam mode Managed siap digunakan."
        fi

        IFACE=`ifconfig | grep wlan1 | awk '{print $1}' | cut -d: -f1`
        if [ "$IFACE" = "wlan1" ]; then
                echo "2. Interface wlan1 dalam mode Managed."
                echo "   -- Mohon tunggu untuk mengubah interface wlan1 ke mode
     Monitor."
                airmon-ng check kill
                airmon-ng start wlan1
                echo
                echo "   -- Interface wlan1 dalam mode Monitor (wlan1mon) siap
     digunakan."
        else
                echo "2. Interface wlan1 dalam mode Monitor (wlan1mon) siap
     digunakan."
        Fi
```

Lampiran 2: *Script* Otomatisasi Tahap Populasi Data *Authorized Access Point*

```
echo "============================================="
echo "POPULASI (UPDATE) DATA AUTHORIZED ACCESS POINT"
echo "============================================="
echo "Pengecekan data Access Point apakah sudah ada di dalam file."
read -p "Masukkan MAC address Access Point: " MAC_ADDR

####### VALIDASI MAC ADDRESS #######

if [ `echo $MAC_ADDR | egrep "^([a-fA-F0-9]{2}[:"-"]){5}[a-fA-F0-9]{2}$"` ]; then
     CONVERT=`echo $MAC_ADDR | tr '-' ':' | tr '[:lower:]' '[:upper:]'`
else
     echo "Format MAC address yang Anda masukkan salah. Silakan coba kembali."
     sleep 2
     bash $POP_DIR/pop_ap.sh
fi

####### PENGECEKAN MAC ADDRESS DI DALAM FILE DATABASE ACCESS POINT #######

MAC_IN_FILE=`fgrep $CONVERT $POP_DIR/data_ap.txt | cut -f2`
if [ "$CONVERT" = "$MAC_IN_FILE" ]; then
     echo "Data MAC address sudah ada di dalam file. Apakah data akan dihapus?"
     echo "1. Ya"
     echo "2. Tidak"
     echo "3. Kembali ke Menu Utama"
     echo "4. Keluar"
     read -p "Pilih nomor menu yang Anda inginkan: " NUM
     case $NUM in
        1)
                bash $POP_DIR/del_ap.sh ;;
        2)
                bash $POP_DIR/pop_ap.sh ;;
        3)
                bash $DIR/main.sh ;;
        4)
                clear
                exit ;;
        *)
                echo "Maaf, pilihan Anda tidak ada dalam menu. Silakan pilih menu
     nomor 1-4."
                sleep 2
                bash $POP_DIR/pop_ap.sh ;;
     esac
else
     echo "Data MAC address belum ada di dalam file. Apakah data akan
     ditambahkan?"
     echo "1. Ya"
        echo "2. Tidak"
        echo "3. Kembali ke Menu Utama"
        echo "4. Keluar"
        read -p "Pilih nomor menu yang Anda inginkan: " NUM
     case $NUM in
                 1)
                         bash $POP_DIR/add_ap.sh ;;
                 2)
                         bash $POP_DIR/pop_ap.sh ;;
                 3)
                         bash $DIR/main.sh ;;
                 4)
                         clear
                exit ;;
                 *)
                echo "Silakan pilih menu nomor 1-4!"
                sleep 2
                bash $POP_DIR/pop_ap.sh ;;
     esac
fi
```

Lampiran 3: *Script* Otomatisasi Fase *Generate* Data *Access Point* pada Tahap Populasi Data *Authorized Access Point*

```
####### SCANNING ACCESS POINT #######
     echo "Mohon tunggu, sistem sedang melakukan scanning Access Point..."
     ####### PARSING PARAMETER ACCESS POINT #######
     python $DIR/iwlistparse.py > $POP_DIR/scan_ap.txt
     MAC_SCAN=`fgrep $CONVERT $POP_DIR/scan_ap.txt | awk '{print $2}'`
     if [ "$CONVERT" = "$MAC_SCAN" ]; then
        SSID=`fgrep $MAC_SCAN $POP_DIR/scan_ap.txt | awk '{print $1}'`
        BSSID=$MAC_SCAN
        CHANNEL=`fgrep $MAC_SCAN $POP_DIR/scan_ap.txt | awk '{print $7}'`
        SIGNAL=`fgrep $MAC_SCAN $POP_DIR/scan_ap.txt | awk '{print $5, $6}'`
        ENCRYPT=`fgrep $MAC_SCAN $POP_DIR/scan_ap.txt | awk '{print $8}'`
        ####### IDENTIFIKASI EAP TYPE #######
        rm -rf $POP_DIR/eap*
        ####### PENGECEKAN KONFIGURASI BSSID DI DALAM FILE KONFIGURASI
     WPA_SUPPLICANT #######
        BSSID_CONF=`fgrep bssid $DIR/ssid.conf`
             if [ "$BSSID_CONF" = "" ]; then
                       sed -i "/network={/a bssid=$BSSID\nssid=\"$SSID\""
     $DIR/ssid.conf
               else
                      sed -i '/^bssid=/d' $DIR/ssid.conf
                      sed -i '/^ssid=/d' $DIR/ssid.conf
                      sed -i "/network={/a bssid=$BSSID\nssid=\"$SSID\""
     $DIR/ssid.conf
               fi
        gnome-terminal --geometry=74x24+-10+19 -x bash -c "wpa_supplicant -i wlan0
     -c $DIR/ssid.conf; exit; bash"
        gnome-terminal --geometry=74x24--80+19 -x bash -c "airodump-ng wlan1mon --
     bssid $BSSID -c $CHANNEL --output-format pcap -w $POP_DIR/eap; exit; bash"
        sleep 8
        pkill wpa_supplicant
        pkill airodump-ng
        echo
        airmon-ng stop wlan1mon
        ifconfig wlan1 up
        EAP=`tshark -r $POP_DIR/eap-01.cap | grep "EAP-" | sed
     's/^.*(\(.*\)).*$/\1/g' | uniq`
     fi
     ####### PRESENTASI DATA ACCESS POINT #######
     if [ "$EAP" = "" ]; then
        echo
        echo "Access Point dengan MAC address $BSSID tidak menggunakan protokol
     EAP."
        EAP="None    "
        echo
     else
        echo
        echo "Jenis protokol EAP yang digunakan Access Point dengan MAC address
     $BSSID adalah $EAP."
        echo
     fi
     ####### PENGECEKAN JUMLAH KARAKTER SSID #######
     if [ ${#SSID} -ge 8 ]; then
        printf "%s\t%s\t%s\t%s\t\t%s\t\t%s\n" $SSID $BSSID $CHANNEL "$SIGNAL"
     $ENCRYPT "$EAP" >> $POP_DIR/data_ap.txt
     else
        printf "%s\t\t%s\t%s\t%s\t\t%s\t\t%s\n" $SSID $BSSID $CHANNEL "$SIGNAL"
     $ENCRYPT "$EAP" >> $POP_DIR/data_ap.txt
     fi
     echo
     echo "Update data Authorized Access Point pada `date` adalah sebagai
     berikut:"
     echo
     cat $POP_DIR/data_ap.txt
     echo
fi
```

Lampiran 4: *Script* Otomatisasi Fase *Generate* Data *Access Point* pada Tahap Deteksi *Rogue Access Point*

```
####### SCANNING UNAUTHORIZED ACCESS POINT #######

echo
echo
echo "Mohon tunggu, sistem sedang melakukan scanning Access Point..."

####### PARSING PARAMETER UNAUTHORIZED ACCESS POINT #######

python $DIR/iwlistparse.py > $MIT_DIR/scan_ap.txt

rm -rf $MIT_DIR/data_ap.txt
touch $MIT_DIR/data_ap.txt
echo
                "=============================================================
                ========================" > $MIT_DIR/data_ap.txt
echo "SSID             BSSID (MAC Address)     Channel Power            Encryption
                EAP Type" >> $MIT_DIR/data_ap.txt
echo
                "=============================================================
                ========================" >> $MIT_DIR/data_ap.txt

ROW_1=`fgrep SSID $MIT_DIR/scan_ap.txt | cut -d" " -f1`
if [ "$ROW_1" = "SSID" ]; then
                sed '/^SSID/d' $MIT_DIR/scan_ap.txt > $MIT_DIR/temp.txt
fi
FILE=$MIT_DIR/temp.txt

while IFS=" " read -r F1 F2 F3 F4 F5 F6 F7 F8
do

                ####### IDENTIFIKASI EAP TYPE #######

        rm -rf $MIT_DIR/eap*
                BSSID_CONF=`fgrep bssid $DIR/ssid.conf`
                if [ "$BSSID_CONF" = "" ]; then
                sed -i "/network={/a bssid=$F2\nssid=\"$F1\"" $DIR/ssid.conf
                else
                sed -i '/^bssid=/d' $DIR/ssid.conf
                sed -i '/^ssid=/d' $DIR/ssid.conf
                sed -i "/network={/a bssid=$F2\nssid=\"$F1\"" $DIR/ssid.conf
                fi
        gnome-terminal --geometry=74x24+-10+19 -x bash -c "wpa_supplicant -i wlan0
                -c $DIR/ssid.conf; exit; bash"
        gnome-terminal --geometry=74x24--80+19 -x bash -c "airodump-ng wlan1mon --
                bssid $F2 -c $F7 --output-format pcap -w $MIT_DIR/eap; exit; bash"
        sleep 8
                pkill wpa_supplicant
                pkill airodump-ng
                echo
        EAP=`tshark    -r    $MIT_DIR/eap-01.cap    |    grep    "EAP-"    |    sed
                's/^.*(\(.*\)).*$/\1/g' | uniq`
                sed -i '/^bssid=/d' $DIR/ssid.conf
        sed -i '/^ssid=/d' $DIR/ssid.conf

                ####### PRESENTASI DATA UNAUTHORIZED ACCESS POINT #######

                if [ "$EAP" = "" ]; then
                     echo
                     echo "Access Point dengan MAC address $F2 tidak menggunakan
                protokol EAP."
                     EAP="None    "
                     echo
                else
                     echo
                     echo "Jenis protokol EAP yang digunakan Access Point dengan
                MAC address $F2 adalah $EAP."
```

```
                    echo
                    cp -r --backup=t eap-01.cap evidence.cap
                fi

                if [ ${#F1} -ge 8 ]; then
                printf "%s\t%s\t%s\t%s\t\t%s\t\t%s\n" $F1 $F2 $F7 "$F5 $F6" $F8
                  "$EAP" >> $MIT_DIR/data_ap.txt
        else
                printf "%s\t\t%s\t%s\t%s\t\t%s\t\t%s\n" $F1 $F2 $F7 "$F5 F6" $F8
                  "$EAP" >> $MIT_DIR/data_ap.txt
        fi

done < $FILE

echo "Hasil scanning Access Point adalah sebagai berikut:"
echo
cat $MIT_DIR/data_ap.txt
sleep 5
echo
rm -rf $FILE $MIT_DIR/eap-01.cap

####### DETEKSI ROGUE ACCESS POINT #######

bash $MIT_DIR/det_rogap.sh

####### ELIMINASI ROGUE ACCESS POINT #######

bash $MIT_DIR/elim_rogap.sh

airmon-ng stop wlan1mon
ifconfig wlan1 up
ifconfig wlan0 up
```

Lampiran 5: *Script* Otomatisasi Komparasi Data pada Tahap Deteksi *Rogue Access Point*

```
####### DETEKSI ROGUE ACCESS POINT #######
echo "####### KOMPARASI DATA HASIL SCANNING AP VS AUTHORIZED AP #######"
####### 1. KOMPARASI SSID #######
echo "================="
echo "1. Komparasi SSID"
echo "================="
echo 'Note: SSID yang berbeda dengan data pada file Authorized AP ditandai sebagai
                  "Neighbor AP".'
echo "Komparasi SSID Hasil Scanning Access Point vs Authorized Access Point:"
awk   'NR==FNR{   a[$1]=$1;next   }   NF{   print   $0   ((a[$1]==$1)?"\tUnauthorized
                  AP":"\tNeighbor AP") }' $TEMP_2 $TEMP_1 > $TEMP_4
echo
                  "================================================================
                  ================================================"
echo "SSID              BSSID (MAC Address)      Channel Power            Encryption
                  EAP Type        Tagged as      "
echo
                  "================================================================
                  ================================================"
cat $TEMP_4
cp $TEMP_4 $MIT_DIR/1.txt
echo 'Data "Neighbor AP" tidak diproses lebih lanjut.'
echo 'Mohon tunggu, sistem sedang melanjutkan proses komparasi data "Unauthorized
                  AP"...'
sed -i '/Neighbor AP$/d' $TEMP_4
cat $TEMP_4 | cut -f1-8 > $TEMP_3
sleep 2
####### 2. KOMPARASI SSID & MAC ADDRESS #######
echo "================================"
echo "2. Komparasi SSID & MAC Address"
echo "================================"
echo 'Note: SSID & MAC address yang sama dengan data pada file Authorized AP ditandai
                  sebagai "Authorized AP".'
echo "Komparasi SSID & MAC Address Unauthorized AP vs Authorized AP:"
awk   'NR==FNR{   a[$2]=$1$2;next   }   NF{   print   $0   ((a[$2]==$1$2)?"\tAuthorized
                  AP":"\tUnauthorized AP") }' $TEMP_2 $TEMP_3 > $TEMP_4
echo
                  "================================================================
                  ================================================"
echo "SSID              BSSID (MAC Address)      Channel Power            Encryption
                  EAP Type        Tagged as      "
echo
                  "================================================================
                  ================================================"
cat $TEMP_4
cp $TEMP_4 $MIT_DIR/2.txt
echo 'Data "Authorized AP" tidak diproses lebih lanjut.'
echo 'Mohon tunggu, sistem sedang melanjutkan proses komparasi data "Unauthorized
                  AP"...'
sed -i '/Authorized AP$/d' $TEMP_4
cat $TEMP_4 | cut -f1-8 > $TEMP_3
sleep 2
####### 3. PENGECEKAN KEKUATAN SINYAL AP #######
echo "========================================="
echo "3. Pengecekan Kekuatan Sinyal Access Point"
echo "========================================="
echo "Note: Kekuatan sinyal minimum untuk dapat mengirimkan paket secara reliable
                  adalah -80 dBm."
echo '     Access Point dengan power <=-80 dBm yang menandakan sinyal lemah ditandai
                  sebagai "Neighbor AP".'
echo "Pengecekan Kekuatan Sinyal Access Point:"
awk '{ if (($4 >= -80)) print $0 "\tUnauthorized AP"; else print $0 "\tNeighbor AP";
                  }' $TEMP_3 > $TEMP_4
echo
                  "================================================================
                  ================================================"
```

```
echo "SSID              BSSID (MAC Address)      Channel Power            Encryption
                  EAP Type        Tagged as      "
echo
                  "================================================================
                  ================================================"
cat $TEMP_4
cp $TEMP_4 $MIT_DIR/3.txt
echo 'Data "Neighbor AP" tidak diproses lebih lanjut.'
echo 'Mohon tunggu, sistem sedang melanjutkan proses komparasi data "Unauthorized
                  AP"...'
sed -i '/Neighbor AP$/d' $TEMP_3
cat $TEMP_4 | cut -f1-8 > $TEMP_3
sleep 2
####### 4. KOMPARASI TIPE ENKRIPSI #######
echo "=========================="
echo "4. Komparasi Tipe Enkripsi"
echo "=========================="
echo 'Note: SSID yang sama namun enkripsi yang berbeda dengan data pada file
                  Authorized AP ditandai sebagai "Neighbor AP".'
echo "Komparasi Tipe Enkripsi Unauthorized Access Point vs Authorized Access Point:"
awk 'NR==FNR{ a[$1]=$1$6;next } NF{ print $0 ((a[$1]==$1$6)?"\tUnauthorized
                  AP":"\tNeighbor AP") }' $TEMP_2 $TEMP_3 > $TEMP_4
echo
                  "================================================================
                  ================================================"
echo "SSID              BSSID (MAC Address)      Channel Power            Encryption
                  EAP Type        Tagged as      "
echo
                  "================================================================
                  ================================================"
cat $TEMP_4
cp $TEMP_4 $MIT_DIR/4.txt
echo 'Data "Neighbor AP" tidak diproses lebih lanjut.'
echo 'Mohon tunggu, sistem sedang melanjutkan proses komparasi data "Unauthorized
                  AP"...'
sed -i '/Neighbor AP$/d' $TEMP_4
cat $TEMP_4 | cut -f1-8 > $TEMP_3
sleep 2
####### 5. KOMPARASI JENIS PROTOKOL EAP #######
echo "================================"
echo "5. Komparasi Jenis Protokol EAP"
echo "================================"
echo 'Note: SSID dengan jenis protokol EAP yang sama dengan data pada file Authorized
                  AP ditandai sebagai "Rogue AP".'
echo "Komparasi Jenis Protokol EAP Unauthorized Access Point vs Authorized Access
                  Point:"
awk 'NR==FNR{ a[$1]=$1$7;next } NF{ print $0 ((a[$1]==$1$7)?"\tRogue AP":"\tNeighbor
                  AP") }' $TEMP_2 $TEMP_3 > $TEMP_4
echo
                  "================================================================
                  ================================================"
echo "SSID              BSSID (MAC Address)      Channel Power            Encryption
                  EAP Type        Tagged as      "
echo
                  "================================================================
                  ================================================"
cat $TEMP_4
cat $TEMP_4 | cut -f1-8 > $TEMP_3
sleep 2
####### PRESENTASI HASIL AKHIR DETEKSI ACCESS POINT #######
cp $MIT_DIR/1.txt $MIT_DIR/hasil_komparasi.txt              #  Backup  file  1.txt
                  untuk inisialisasi iterasi
### ITERASI 1
DIFFER=`awk   'FNR==NR{a[$0]++;next}(!($0   in   a))'   $MIT_DIR/hasil_komparasi.txt
                  $MIT_DIR/2.txt | cut -f1-8`
NEW=`awk    'FNR==NR{a[$0]++;next}(!($0    in    a))'    $MIT_DIR/hasil_komparasi.txt
                  $MIT_DIR/2.txt`
if [ "$DIFFER" = "" ]; then
        cp $MIT_DIR/hasil_komparasi.txt $MIT_DIR/1.txt
else
        while read -r -u 3 LINE_1 && read -r -u 4 LINE_2; do
```

```
                sed -i.bak "s/^$LINE_2.*/$LINE_1/" $MIT_DIR/hasil_komparasi.txt
        done 3<<<"$NEW" 4<<<"$DIFFER"
fi
### ITERASI 2
DIFFER=`awk   'FNR==NR{a[$0]++;next}(!($0   in   a))'   $MIT_DIR/hasil_komparasi.txt
                $MIT_DIR/3.txt | cut -f1-8`
NEW=`awk    'FNR==NR{a[$0]++;next}(!($0    in    a))'    $MIT_DIR/hasil_komparasi.txt
                $MIT_DIR/3.txt`
if [ "$DIFFER" = "" ]; then
        cp $MIT_DIR/hasil_komparasi.txt $MIT_DIR/1.txt
else
        while read -r -u 3 LINE_1 && read -r -u 4 LINE_2; do
                sed -i.bak "s/^$LINE_2.*/$LINE_1/" $MIT_DIR/hasil_komparasi.txt
        done 3<<<"$NEW" 4<<<"$DIFFER"
fi
### ITERASI 3
DIFFER=`awk   'FNR==NR{a[$0]++;next}(!($0   in   a))'   $MIT_DIR/hasil_komparasi.txt
                $MIT_DIR/4.txt | cut -f1-8`
NEW=`awk    'FNR==NR{a[$0]++;next}(!($0    in    a))'    $MIT_DIR/hasil_komparasi.txt
                $MIT_DIR/4.txt`
if [ "$DIFFER" = "" ]; then
        cp $MIT_DIR/hasil_komparasi.txt $MIT_DIR/1.txt
else
        while read -r -u 3 LINE_1 && read -r -u 4 LINE_2; do
                sed -i.bak "s/^$LINE_2.*/$LINE_1/" $MIT_DIR/hasil_komparasi.txt
        done 3<<<"$NEW" 4<<<"$DIFFER"
fi
### ITERASI 4
DIFFER=`awk   'FNR==NR{a[$0]++;next}(!($0   in   a))'   $MIT_DIR/hasil_komparasi.txt
                $TEMP_4 | cut -f1-8`
NEW=`awk 'FNR==NR{a[$0]++;next}(!($0 in a))' $MIT_DIR/hasil_komparasi.txt $TEMP_4`
if [ "$DIFFER" = "" ]; then
        cp $MIT_DIR/hasil_komparasi.txt $MIT_DIR/1.txt
else
        while read -r -u 3 LINE_1 && read -r -u 4 LINE_2; do
                sed -i.bak "s/^$LINE_2.*/$LINE_1/" $MIT_DIR/hasil_komparasi.txt
        done 3<<<"$NEW" 4<<<"$DIFFER"
fi
echo "================================"
echo "Hasil Akhir Deteksi Access Point"
echo "================================"
echo "Data Access Point yang terdeteksi adalah sebagai berikut:"
echo
                "================================================================
                ================================================"
echo "SSID            BSSID (MAC Address)     Channel Power           Encryption
                EAP Type        Tagged as      "
echo
                "================================================================
                ================================================"
cat $MIT_DIR/hasil_komparasi.txt
sleep 5
####### PRESENTASI FILE DATABASE ACCESS POINT UNTUK MENU UTAMA #######
FILE_ROGAP="$MIT_DIR/rogue_ap.txt"
if [ -f "$FILE_ROGAP" ]; then :
else
        touch $MIT_DIR/rogue_ap.txt
        echo
                "================================================================
                ================================================="          >
                $MIT_DIR/rogue_ap.txt
        echo "SSID          BSSID (MAC Address)    Channel Power         Encryption
                EAP Type        Time Detected      " >> $MIT_DIR/rogue_ap.txt
        echo
                "================================================================
                ================================================="          >>
                $MIT_DIR/rogue_ap.txt
fi
cat $TEMP_3 | sed "s/$/ \t`date`/" >> $MIT_DIR/rogue_ap.txt
#rm  -rf  $TEMP_1  $TEMP_2  $TEMP_3  $MIT_DIR/1.txt  $MIT_DIR/2.txt  $MIT_DIR/3.txt
                $MIT_DIR/4.txt $MIT_DIR/*.bak
```

Lampiran 6: *Script* Otomatisasi Tahap Eliminasi *Rogue Access Point*

```
#!/bin/bash

DIR=/home/script
POP_DIR=$DIR/populasi
MIT_DIR=$DIR/mitigasi
TEMP_4="$MIT_DIR/komparasi_ap.tmp"           # File data AP dengan tag

####### ELIMINASI ROGUE ACCESS POINT #######

echo
if grep -Fq "Rogue AP" $TEMP_4; then
        echo "Selanjutnya akan dilakukan proses eliminasi Rogue AP."
        echo "Mohon tunggu, sistem sedang melakukan proses eliminasi Rogue AP..."
        echo

        while IFS=$'\t' read -r F1 F2 F3 F4 F5 F6 F7
        do

                echo "============================="
                echo "1. Pengecekan Client Rogue AP"
                echo "============================="
                rm -rf $MIT_DIR/mon_ap*
                gnome-terminal  --geometry=74x24+-10+19  -x  bash  -c  "airodump-ng
wlan1mon --bssid $F2 -c $F3 --output-format csv -w $MIT_DIR/mon_ap; exit; bash"
                 sleep 8
                pkill airodump-ng
                 grep -aA 100 'MAC' `ls $MIT_DIR/mon_ap-01.csv` | grep "$1" | head -
n -1 | sed 1d | cut -f1 -d, > $MIT_DIR/client_rogap.txt
                if [ -s "$MIT_DIR/client_rogap.txt" ]; then
                        echo "Berikut adalah daftar client yang terasosiasi ke Rogue
AP:"
                          cat $MIT_DIR/client_rogap.txt
                 else
                          echo "Tidak ada client yang terasosiasi ke Rogue AP."
                          echo "Tetap melanjutkan proses containment Rogue AP."
                 fi

                 echo
                echo "======================================="
                echo "2. Deauth Client & Containment Rogue AP"
                echo "======================================="
                echo  "Deauthentication  client  &  containment  Rogue  AP  sedang
dijalankan..."
                 gnome-terminal --geometry=74x24--80+19 -x bash -c "aireplay-ng -0
50 -a $F2 wlan1mon; exit; bash"
                 sleep 25
                pkill aireplay-ng
                echo "."
                echo "."
                echo "."
                echo "Deauthentication client & containment Rogue AP telah selesai
dijalankan."

                echo
                echo
"============================================================="
                 echo  "3.  Pengecekan  Client  Rogue  AP  Pasca  Proses  Deauth  &
Containment"
                 echo
"============================================================="
                rm -rf $MIT_DIR/mon_ap*
                 gnome-terminal  --geometry=74x24+-10+19  -x  bash  -c  "airodump-ng
wlan1mon --bssid $F2 -c $F3 --output-format csv -w $MIT_DIR/mon_ap; exit; bash"
                 sleep 8
                pkill airodump-ng
                grep -aA 100 'MAC' `ls $MIT_DIR/mon_ap-01.csv` | grep "$1" | head -n
-1 | sed 1d | cut -f1 -d, > $MIT_DIR/client_rogap.txt
                 if [ -s "$MIT_DIR/client_rogap.txt" ]; then
```

```
                        while [ -s "$MIT_DIR/client_rogap.txt" ]; do
                                gnome-terminal --geometry=74x24--80+19 -x bash -c
"aireplay-ng -0 50 -a $F2 wlan1mon; exit; bash"
                                sleep 25
                              pkill aireplay-ng
                                rm -rf $MIT_DIR/mon_ap*
                                gnome-terminal --geometry=74x24+-10+19 -x bash -c
"airodump-ng wlan1mon --bssid $F2 -c $F3 --output-format csv -w $MIT_DIR/mon_ap;
exit; bash"
                                sleep 8
                              pkill airodump-ng
                              grep -aA 100 'MAC' `ls $MIT_DIR/mon_ap-01.csv` | grep
"$1" | head -n -1 | sed 1d | cut -f1 -d, > $MIT_DIR/client_rogap.txt
                              if [ ! -s "$MIT_DIR/client_rogap.txt" ]; then
                                        echo   "Sudah   tidak   ada   client   yang
terasosiasi ke Rogue AP."
                                    echo
                                    echo   "Proses   deauthentication   client   &
containment Rogue AP berhasil dilakukan."
                                fi
                      done
                else
                       echo "Sudah tidak ada client yang terasosiasi ke Rogue AP."
                      echo
                      echo "Proses deauthentication client & containment Rogue AP
berhasil dilakukan."
                fi

        done < $TEMP_4
else
        echo "Tidak ada Rogue AP yang terdeteksi."
        echo "Proses eliminasi Rogue AP tidak dilakukan."
        echo
fi

echo

#rm -rf $TEMP_4 $MIT_DIR/client_rogap.txt $MIT_DIR/mon_ap*
```