# BAB III
# METODOLOGI PENELITIAN

## 3.1 Objek dan Subjek penelitian

Dalam penelitian ini, objek yang dikaji adalah penyisipan *malware* pada aplikasi Baidu Browser serta pengujian menggunakan *Mobile Security Framework (MobSF)*. Pengujian dilakukan dengan cara melakukan implementasikan satu perangkat laptop yang telah terinstall sebuah sistem operasi *kali linux*. Kemudian pada penelitian ini subjek yang digunakan adalah dua alat yang digunakan, yaitu *Metasploit* dan *FatRAT*. *Metasploit* digunakan sebagai kerangka dasar dalam penetrasi yang luas dan kuat pada penelitian ini, sementara itu *FatRAT* merupakan *tools* khusus yang digunakan untuk membantu menyisipkan *payload* berjenis *android_reverse_tcp* kedalam aplikasi android. Penelitian ini memanfaatkan sumber data yang didapatkan dari hasil pengujian penyisipan *malware* pada aplikasi Baidu Browser. Data yang didapatkan berisi informasi mengenai proses penyisipan *malware* ke dalam aplikasi tersebut, termasuk proses implementasi, teknik yang digunakan pada penelitian ini, serta hasil yang diperoleh.

## 3.2 Alat dan Bahan Penelitian

Dalam penelitian ini, dibutuhkan beberapa perangkat yang dapat mendukung untuk menyelesaikan penelitian ini. Perangkat yang dibutuhkan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*). Tabel di bawah ini merupakan spesifikasi perangkat yang dibutuhkan dalam penelitian ini, yaitu :

Tabel 3.1 Kebutuhan *Hardware* (Perangkat keras)

| No | Perangkat Keras | Spesifikasi | Keterangan |
|---|---|---|---|
| 1. | 1 Buah Laptop | Amd Ryzen 7 5700u, 8GB DDR4L, SSD 512Gb NVME, 1TB SATA (Tripleboot) | Laptop sebagai penyisip dan penyerang |
| 2 | 1 Buah Smartphone Android | Qualcomm MSM8917 Snapdragon 425, RAM 4GB, ROM 64GB | Digunakan sebagai target |
| 3 | 1 Buah Kabel Data | Micro Usb | Memindahkan aplikasi dari laptop ke target |

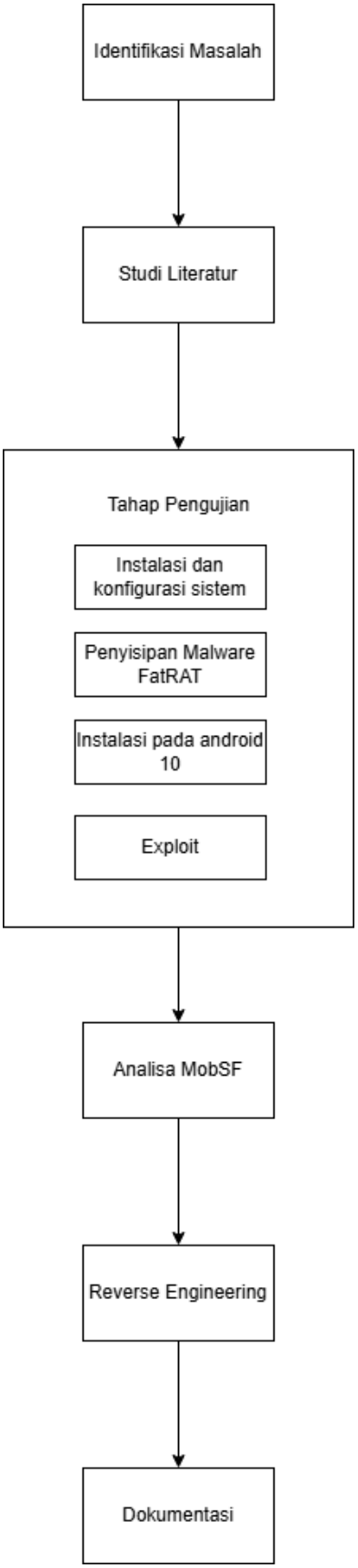
Tabel 3.2 Kebutuhan Software (Perangkat Lunak)

| No | Perangkat Lunak | Keterangan |
|---|---|---|
| 1 | Kali Linux 2023.1 | Sistem Operasi berbasis *open source* yang digunakan untuk melakukan penyerangan. |

| | | |
|---|---|---|
| 2 | TheFatRAT | Perangkat lunak untuk melakukan *injeksi malware*. |
| 3 | Metasploit | *Tools* untuk melakukan eksploitasi terhadap target. |
| 4 | ApkTools | *Tools* pendukung TheFatRAT untuk melakukan *decompile* dan *recompile* otomatis ketika melakukan injeksi |
| 5 | ApkSigner | *Tools* pendukung TheFatRAT |
| 6 | JADX | Perangkat lunak untuk *decompile* dan analisa manual APK. |
| 7 | MobSF | Perangkat lunak untuk melakukan analisa secara otomatis. |
| 8 | Docker | Perangkat lunak untuk menjalankan MobSF. |
| 9 | NGROK | Perangkat lunak untuk membuat *Ihose*t dan *Iport* menjadi satu *segmen* tcp. |
| 10 | Baidu Browser | Aplikasi android yang digunakan sebagai media penyisipan *malware*. |

## 3.3 Diagram Alur Penelitian

Pada alur penelitian ini dilakukan secara sistematis dan secara runtut terstruktur melalui beberapa rangkaian tahapan yang saling berkaitan satu sama lain. Penelitian diawali dengan dilakukannya identifikasi masalah yang kemudian dilanjutkan dengan melakukan studi literatur, dan diakhiri dengan tahap pengujian. Tahap pengujian sendiri terdiri atas beberapa langkah. Pertama, dilakukan persiapan instalasi serta konfigurasi sistem yang akan digunakan untuk melakukan penelitian. Kedua, melakukan penyisipan *malware FatRAT* ke dalam aplikasi baidu browser, kemudian diikuti oleh proses instalasi pada versi android yang ditargetkan. Ketiga, pengujian eksploitasi dilakukan untuk mengetahui dan memahami perilaku *malware*. Keempat, tahap analisis yang dibagi menjadi dua yaitu menggunakan *reverse engineering*. Dalam hal ini, dilakukan analisis mendalam terhadap kode dan struktur aplikasi untuk memahami lebih lanjut cara kerja *malware*. Langkah terakhir, analisis akan didokumentasikan dalam bentuk laporan. Laporan ini mencakup bukti-bukti yang ditemukan selama penelitian, dan temuan serta rekomendasi yang dihasilkan sebagai hasil dari proses penelitian ini. Adapun pemaparan alur diagram yang dirancang pada penelitian ini, seperti yang tertera pada gambar 3.1


Identifikasi Masalah
Studi Literatur
Tahap Pengujian
Instalasi dan konfigurasi sistem
Penyisipan Malware FatRAT
Instalasi pada android 10
Exploit
Analisa MobSF
Reverse Engineering
Dokumentasi

Gambar 3.1 Diagram Alur Penelitian

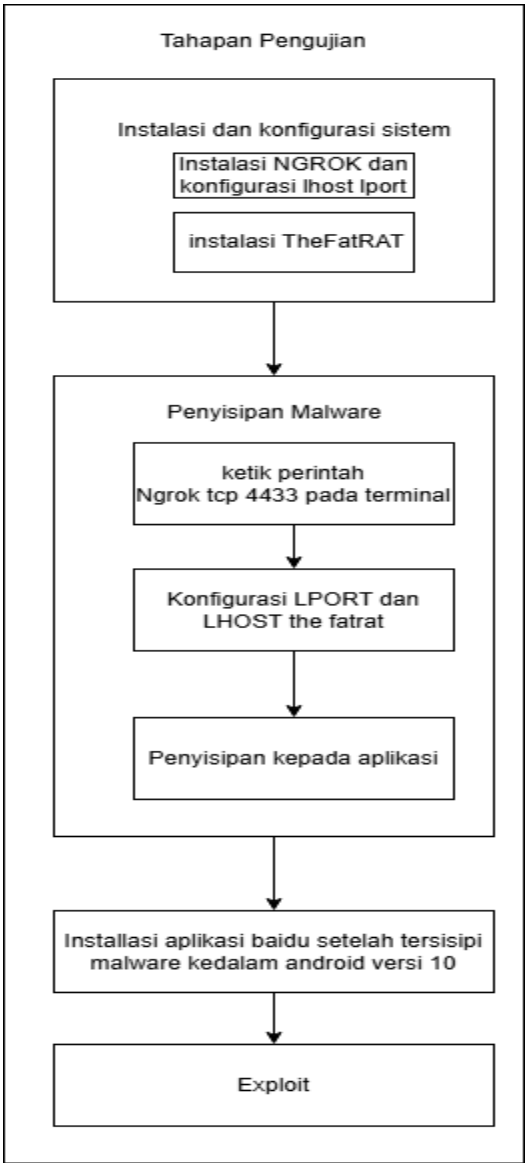
### 3.3.1 Identifikasi Masalah

Meningkatnya serangan *malicious software* atau *malware* pada platform sistem android menjadi sebuah perhatian yang utama bagi para pengguna karena meningkatnya risiko pegunduhan aplikasi yang telah terjangkit *malware*. *Malware* dirancang dengan sengaja oleh seseorang dengan maksud dan tujuan jahat untuk mencuri data berharga. Maka dengan adanya *malware* yang terjangkit pada android pengguna dapat menyebabkan kerugian besar. Salah satu teknik yang umum digunakan oleh penyerang adalah dengan membuat sebuah *backdoor* yang kemudian hasil *backdoor* tersebut disisipkan ke dalam aplikasi android. *Backdoor* yang berfungsi sebagai jalan belakang memberikan akses secara penuh kepada penyerang serta meninggalkan sebuat *footprint* dari keamanan yang dimanfaatkan pada sistem android. Hal ini meninggalkan potensi ancaman terhadap keamanan perangkat android serta data pribadi pengguna android yang tersimpan di dalamnya.

### 3.3.2 Studi Literatur

Dalam penelitian ini, studi literatur menjadi hal yang sangat penting dalam mendukung pengetahuan dasar dalam melakukan analisis, implementasi, dan pengujian terkait dengan serangan *malware* pada aplikasi android. Penelitian ini menggunakan teori-teori yang relevan dengan identifikasi masalah sebagai studi literatur, seperti peningkatan serangan *malware* pada platform sistem android, mekanisme pembuatan dan penyisipan *backdoor* dalam aplikasi, serta teknik – teknik yang digunakan oleh *peretas* dalam menciptakan dan menyebarkan *malware.* Informasi dan pemahaman didapatkan dari sumber literatur seperti buku, jurnal ilmiah, serta *website* terkait dengan topik menjadi sumber pengetahuan yang diperlukan untuk melakukan analisis secara mendalam mengenai serangan *malware*, serta untuk merumuskan strategi pengujian yang efektif dengan tujuan mendapatkan hasil yang maksimal.

### 3.3.3 Tahap Pengujian




Gambar 3.2 Alur Pengujian

1. Instalasi dan Konfigurasi Sistem

   Pada tahap instalasi dan konfigurasi *system kali linux* dimulai dari *penginstalan dependensi* yang dibutuhkan seperti *penginstalan apktool* sebagai dependensi yang digunakan oleh *TheFatRAT* untuk melakukan proses penyisipan *malware* ke dalam aplikasi yang digunakan sebagai media *malware*, kemudian dilanjutkan dengan melakukan *penginstalan*

*TheFatRAT* dengan cara melakukan clone repositori *TheFatRAT* dengan perintah *git clone* https://github.com/Screetsec/TheFatRat.git perintah tersebut digunakan untuk *mendownload TheFatRAT* ke dalam sistem lokal komputer dengan menggunakan git. Selanjutnya masuk ke dalam folder hasil *download* melalui terminal dengan perintah *cd TheFatRat* dilanjutkan dengan perintah *chmod +x setup.sh*. Perintah ini akan memberikan akses *root* pada *TheFatRAT* agar dapat diinstal, setelah memasukkan perintah tersebut dan *TheFatRAT* terinstal dengan sempurna, proses persiapan instalasi dilanjutkan dengan *menginstall ngrok*.

Pada penelitian ini ngrok digunakan untuk membuat sebuah *protocol tcp* agar *ip address* target dengan laptop dapat terhubung, untuk melakukan *instalasi ngrok* dilakukan dengan cara memasukkan perintah *curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee /etc/apt/trusted.gpg.d/ngrok.asc >/dev/null && echo "deb https://ngrok-agent.s3.amazonaws.com buster main" | sudo tee /etc/apt/sources.list.d/ngrok.list && sudo apt update && sudo apt install ngrok* perintah tersebut akan *mendownload* dan *menginstall ngrok* melalui *apt*, setelah *ngrok terinstall* dilanjutkan dengan memasukkan token dari akun *ngrok* ke dalam *ngrok* yang telah *terinstall* dengan perintah *ngrok config add-authtoken* <token> setelah ngrok *terinstall* dengan sempurna maka dilanjutkan dengan tahap ke dua untuk tahapan instalasi.

2. Penyisipan *Malware*

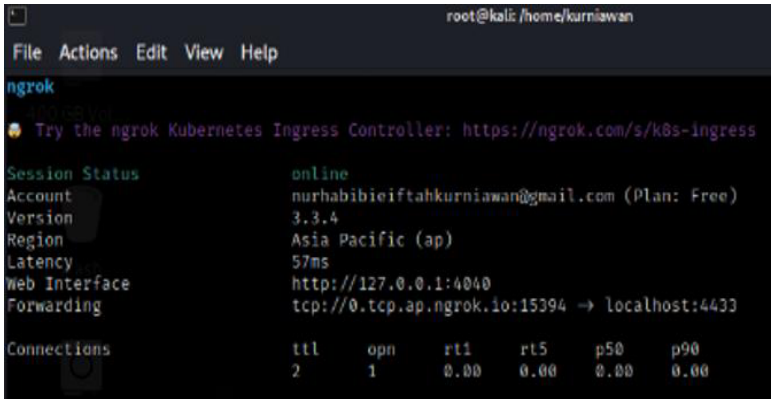
Setelah selesai melakukan penyiapan dan konfigurasi sistem selanjutnya adalah melakukan penyisipan malware yang diawali dengan melakukan konfigurasi *ngrok* dan *thefatrat* yang dibagi menjadi dua tahapan pada tahap awal melakukan konfigurasi ngrok dan tahap kedua melakukan konfigurasi terhadap *thefatrat*.

a. Tahap Instalasi dan Konfigurasi Ngrok

| |
|---|
| 1. Membuka terminal pada kali linux<br>2. Jalankan perintah *sudo su*<br>3. Masukkan perintah ini<br>Wget<br>https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz<br>4. Setelah berhasil terunduh masukkan perintah sudo tar xvzf ./ngrok-v3-stable-linux-amd64.tgz -C /usr/local/bin untuk menyalin file kedalam direktori /usr/local/bin<br>5. Kemudian masukan authtoken dengan perintah ngrok authtoken NGROK_AUTHTOKEN |

Tabel 3.3 Tahap Instalasi Ngrok

Tabel 3.3 merupakan langkah awal untuk melakukan instalasi dan konfigurasi terhadap aplikasi ngrok yang berperan sebagai *tunnelling* agar *smartphone* dapat terhubung dengan laptop penyerang.

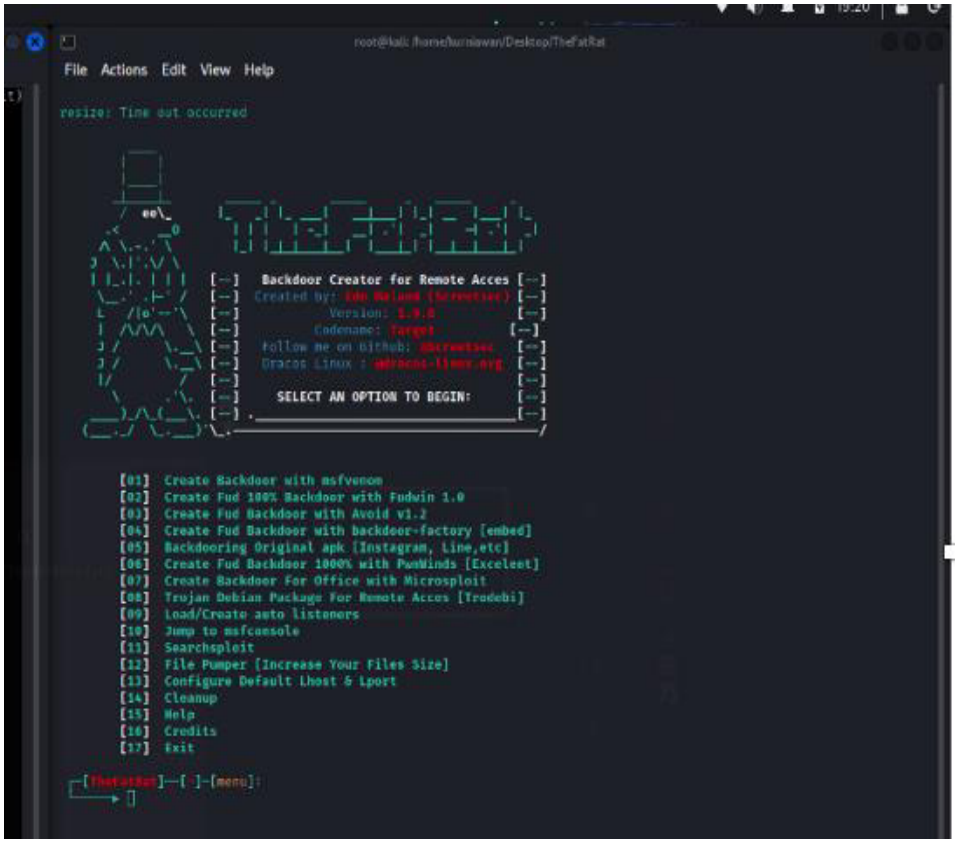



Gambar 3.3 Tampilan *Ngrok* Setelah Berhasil Terinstal Dan Dijalankan

b. Pada gambar 3.3 merupakan tampilan *ngrok* yang menampilkan informasi akun *ngrok*, jenis *tunneling* yang digunakan, *port* yang

digunakan untuk melakukan *forwarding*, *region* dan versi *ngrok* yang dijalankan sebagai *tunneling*. Untuk melakukan konfigurasi *ngrok* setelah diinstall gunakan perintah *ngrok tcp 4433* yang menandakan *tunneling* yang digunakan berjenis tcp dengan *port* 4433. setelah terset selanjutnya dilakukan konfigurasi pada *thefatrat*.

c. Tahap Konfigurasi TheFatRAT




Gambar 3.4 Tampilan *TheFatRAT*

Pada gambar 3.4 merupakan tampilan menu dari *TheFatRAT* pada tahap ini yang perlu dilakukan adalah memasukan *port* acak yang diberikan oleh *ngrok* ke dalam *thefatrat*, pada menu awal pilih nomor 5 setelah memilih akan diarahkan ke dalam jendela konfigurasi LHOST dan LPORT untuk LHOST masukkan 0.tcp.ngrok.io pada LPORT masukkan 15394 atau sesuai dengan *port* yang diberikan oleh *ngrok*

angka yang ada pada ngrok setelah host. ketika lport dan lhost telah diatur berikan destinasi aplikasi yang akan dilakukan injeksi ke dalam *thefatrat file destination.*

d. *LHOST*

*LHOST* pada *metasploit* digunakan untuk menentukan alamat IP di mana *payload* yang dieksekusi oleh target akan terhubung kembali. Sebagai contoh, jika seorang peneliti keamanan menggunakan *metasploit* untuk menyerang sistem jarak jauh dan menggunakan payload *reverse shell*, *LHOST* akan diatur ke alamat IP dari mesin peneliti. *Payload* tersebut akan mengonfigurasi *shell* terbalik untuk membuat koneksi kembali ke *LHOST* tersebut.

e. LPORT

Dalam penggunaan *metasploit*, *LPORT*, atau *local PORT*, adalah parameter yang menentukan nomor port di sistem lokal (misalnya, komputer penyerang) yang akan digunakan untuk mendengarkan koneksi masuk dari target setelah *payload* berhasil dijalankan. *LPORT* adalah bagian penting dari konfigurasi sebuah eksploitasi karena ini menyangkut bagaimana komunikasi antara sistem target dan sistem penyerang akan dilakukan. Sebagai contoh, ketika menggunakan sebuah *reverse shell payload*, penyerang harus menentukan *LPORT* sebelum mengirim *payload* ke sistem target. Setelah payload diaktifkan pada sistem target, maka *payload* akan mencoba terhubung kembali ke alamat IP dan nomor *port* yang telah ditentukan (*LPORT*) pada sistem penyerang. *Port* ini harus terbuka dan dapat diakses dari jaringan eksternal atau sesuai dengan jaringan yang terlibat dalam pengujian.

f. Handler


Gambar 3.5 *Handler*

*Handler* merupakan pogram yang menjalankan interupsi tertentu. *Handler* melakukan kontrol terhadap input maupun output, *file*, atau lainnya dalam menjalankan interupsi tersebut. Sedangkan dalam konteks *metasploit handler* adalah bagian yang membantu peneliti keamanan dan/atau penguji penetrasi untuk dapat berinteraksi dengan sistem yang telah disusupi karena memiliki tanggung jawab mendengarkan dan menangani koneksi balik dari *shell* atau *payload* yang dieksekusi pada sistem target.

Dengan demikian saat penyerang menggunakan *metasploit* untuk mengeksploitasi sistem target, penyerang tersebut akan memilih sebuah *exploit* dan *payload* yang sesuai. *Payload* ini dijalankan di sistem target dan membuat koneksi balik ke mesin penyerang. *Handler* bertugas untuk mendengarkan koneksi masuk ini. Apabila koneksi berhasil terbentuk, *handler* akan membangun sesi antara mesin penyerang dan target. Hal ini memungkinkan penyerang untuk menjalankan berbagai perintah dan mengontrol sistem target.

g. Meterpreter


Gambar 3.6 *Meterpreter*

*Meterpreter* adalah *payload* dalam kerangka kerja *metasploit* yang memberikan kontrol interaktif dan kuat atas sistem yang berhasil disusupi. Dengan fitur-fitur seperti komunikasi terenkripsi, pemuatan modul dinamis, migrasi proses, manipulasi sistem file dan *registry*, serta interaksi dengan sistem operasi, *meterpreter* memungkinkan pengguna untuk melakukan berbagai tindakan pada sistem target setelah berhasil dieksploitasi.

Komponen-komponen utama *meterpreter* meliputi *core* (inti), *stdapi* (API standar), *priv* (fitur dengan hak istimewa tinggi), dan *incognito*

(fitur penyamaran). Penggunaan *meterpreter* melibatkan pemilihan *exploit* yang sesuai, pengaturan *meterpreter* sebagai *payload*, konfigurasi opsi seperti alamat IP dan *port*, dan eksekusi *exploit*. Jika berhasil, meterpreter akan terhubung ke sistem target, memberikan kontrol yang luas kepada pengguna.

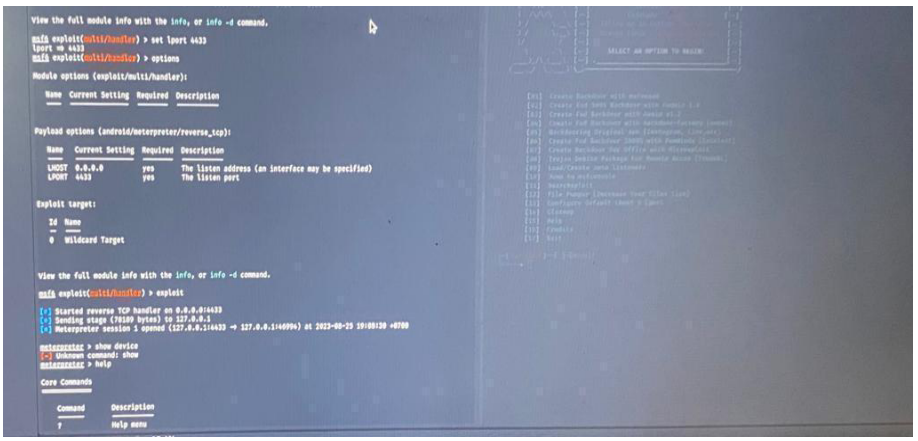
3. Instalasi Aplikasi Baidu Browser Pada Versi Android

   Instalasi aplikasi baidu browser yang telah disisipi *malware* pada versi android adalah salah satu tahap krusial karena bertujuan untuk memastikan bahwa aplikasi yang telah terinjeksi *malware* dapat *terinstall* dan berjalan dengan baik dan dapat berjalan sebagaimana mestinya pada semua versi android atau beberapa versi android saja. Pada penelitian ini dilakukan instalasi pada versi android 13 dengan OneUI 3.0 dan versi android 10 dengan OneUI 2.0. Aplikasi dapat terinstal dengan sempurna pada kedua versi android tersebut namun ketika dilakukan pengujian pada android 13 terlalu sering mengalami *force close* sehingga pada penelitian ini menggunakan android versi 10 untuk melakukan pengujian karena pada versi tersebut hampir seluruh perintah dapat dijalankan terutama pada perintah yang telah digunakan pada tabel 4.1.

4. *Exploit*

   Langkah terakhir dalam tahapan pengujian ini adalah melakukan uji *exploitasi* dengan perintah *exploit,* pada tahap ini aplikasi yang telah disisipi *malware* akan diuji untuk melihat seberapa rentan atau tangguhnya terhadap serangan dari *malware* yang disisipkan sebelumnya. Setelah dilakukan *exploit* kemudian menjalankan *meterpreter*. Perintah untuk melakukan *exploit* adalah dengan menjalankan perintah awal *mfsconsole*, kemudian melakukan *setting multihandler* dengan perintah *use multi/handler* yang digunakan untuk memanggil modul "*handler*" yang digunakan untuk menangani koneksi balik (*reverse shell*) dari target kemudian dilanjutkan dengan melakukan *setting payload* agar *tersetting* sesuai dengan target dengan

menggunakan perintah *set payload android/meterpreter/reverse_tcp* setelah *payload* terpilih lanjutkan dengan melakukan *setting port* dan *host*, untuk mengatur *lport* dan *lhost* masukkan perintah *options* untuk melihat pengaturan *default*, setelah muncul pengaturan *default* ubah pengaturan *default* sesuai dengan *output ngrok* yang digunakan untuk membuat ip satu segmen dengan cara *set lhost 0.0.0.0 set lport 4433 (*sesuaikan dengan *port* yang dimasukkan ketika memanggil *ngrok tcp)* setelah berhasil maka dilanjutkan dengan perintah *exploit*

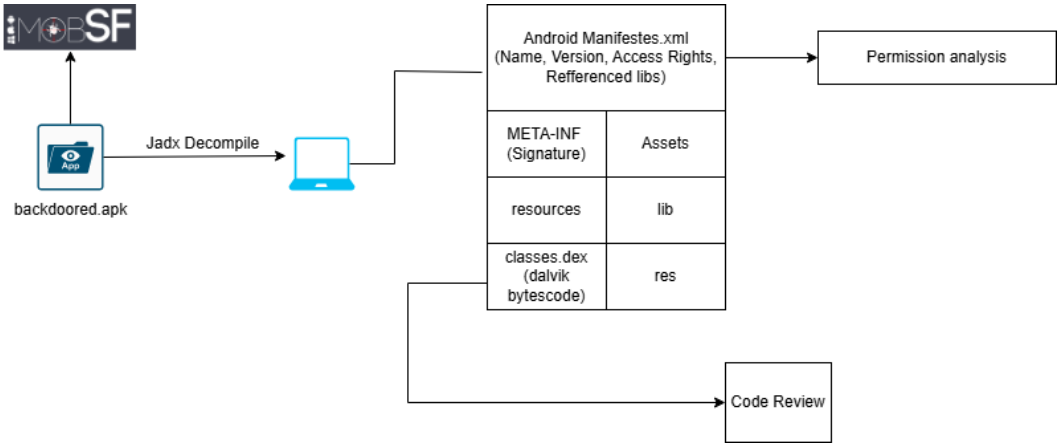



Gambar 3.7 *Exploit*

## 3.4 Tahap Analisis

Pada tahap analisis ini, dilakukan dua proses analisis penting untuk mengetahui secara pasti karakter dan perilaku *malware* yang telah disisipkan ke dalam aplikasi baidu browser, di mana pada tahap awal analisis aplikasi yang telah disisipi *malware* akan diunggah ke dalam *server localhost mobsf* yang telah *terinstall* pada *docker*, kemudian setelah diunggah *mobsf* akan melakukan proses *scanning* dari hasil *scanning* akan dilakukan proses analisis lanjutan secara mendalam menggunakan metode *reverse_engineering* terhadap aplikasi baidu browser untuk mengetahui secara lebih lanjut perubahan dan anomali yang ada pada aplikasi setelah dilakukan injeksi *malware.* Dalam tahap ini akan dilakukan analisis mendalam terhadap susunan

kode sumber aplikasi untuk mengetahui adanya perubahan yang terjadi setelah adanya penyisipan *malware*.




Gambar 3.8 Alur *Reverse Engineering*

Berdasarkan pada gambar 3.8 aplikasi yang telah mengandung *malware* akan dilakukan *decompile* menggunakan *JADX*. Saat aplikasi tersebut ter*decompile* dengan *JADX*, hasil yang didapatkan berupa sejumlah folder dan berkas yang mencakup struktur inti dari aplikasi android. Berkas yang dihasilkan mencakup *AndroidManifest.xml*, yang mana pada *file* tersebut tersimpan *manifest* aplikasi yang berisi informasi penting tentang aplikasi dan izin-izin yang diminta aplikasi kepada perangkat android. Selain itu hasil *decompile* juga akan menghasilkan berkas lain seperti *signature*, *assets*, *resources*, *lib*, *classes.dex* dan *res*, pada berkas *META-INF* berisi informasi *metadata* dari aplikasi yang dijadikan bahan injeksi *malware*, pada berkas assets berisi asset – asset pembangun aplikasi seperti file konfigurasi atau berkas – berkas data yang berkaitan dengan aplikasi. Berkas *resources* berisi sumber daya aplikasi meliputi gambar, *settings*, tata letak, serta *string*. Direktori lib berisi perpustakaan ataupun modul pendukung tambahan dalam aplikasi, pada folder res berisi berkas tambahan seperti *ikon*, dan *file xml*. Terakhir, berkas *classes.dex* berisi kode aplikasi yang telah berhasil terkompilasi.