PEMBUATAN APLIKASI DENGAN MENGGUNAKAN METODE LIVE IMAGING ACQUSITION UNTUK EKSPLORASI BARANG BUKTI DIGITAL PADA MEDIA PENYIMPANAN PONSEL ANDROID

Disusun Oleh:

N a m a : Insan Fadhilah
NIM : 13523038

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2020

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PEMBUATAN APLIKASI DENGAN MENGGUNAKAN METODE LIVE IMAGING ACQUSITION UNTUK EKSPLORASI BARANG BUKTI DIGITAL PADA MEDIA PENYIMPANAN PONSEL ANDROID

TUGAS AKHIR

Disusun Oleh:

N a m a	: Insan Fadhilah
NIM	: 13523038

Yogyakarta, 21 Agustus 2020

Pembimbing,

(Fietyata Yudha S.Kom., M.Kom)

HALAMAN PENGESAHAN DOSEN PENGUJI

PEMBUATAN APLIKASI DENGAN MENGGUNAKAN METODE LIVE IMAGING ACQUSITION UNTUK EKSPLORASI BARANG BUKTI DIGITAL PADA MEDIA PENYIMPANAN PONSEL ANDROID

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 9 September 2020

Tim Penguji

Fietyata Yudha, S.Kom., M.Kom. _____

Anggota 1

Syarif Hidayat, Dr., S.Kom, M.I.T. _____

Anggota 2

Ahmad Fathan Hidayatullah,S.T., M.Cs. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Insan Fadhilah

NIM : 13523038

Tugas akhir dengan judul:

PEMBUATAN APLIKASI DENGAN MENGGUNAKAN METODE LIVE IMAGING ACQUSITION UNTUK EKSPLORASI BARANG BUKTI DIGITAL PADA MEDIA PENYIMPANAN PONSEL ANDROID

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 21 Agustus 2020

(Insan Fadhilah)

HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim. Alhamdulillah,

segala puji bagi *Allah Subhanahu Wa Ta'ala* atas segala nikmat, rahmat, dan karunia-Nya. Tidak lupa shalawat serta salam senantiasa tercurahkan untuk junjungan kita Nabi *Muhammad Shallallahu 'Alaihi Wa Salam*, yang telah membawa kita dari zaman jahiliyah menuju zaman terang benderang seperti saat ini. Tugas Akhir ini saya persembahkan kepada:

1. Kedua Orang Tua saya, Almarhum Bapak Didi Sutardi dan Ibu Elin Anita yang telah memberikan dukungan, semangat, kasih sayang, dan doa yang terbaik.
2. Kedua Kakak perempuan saya, Novie Yuanita Indah Lestari dan Julian Dwi Kusuma Indah Lestari yang juga selalu memberikan semangat, kasih sayang dan doa yang terbaik.
3. Fietyata Yudha S.Kom., M.Kom., selaku Dosen Pembimbing yang selalu memberikan dukungan, semangat dan kesempatan.
4. Teman dan Sahabat yang selalu ada disaat suka dan duka.

HALAMAN MOTO

"Maka sesungguhnya bersama kesulitan itu ada kemudahan. Sesungguhnya bersama kesulitan itu ada kemudahan" (Qs. Al-Insyirah:5-6).

"Allah tidak akan membebani seseorang melainkan sesuai dengan kadar kesanggupannya" (Qs. Al-Baqarah:286)

"Waktu bagaikan pedang. Jika engkau tidak memanfaatkannya dengan baik (untuk memotong), maka ia akan memanfaatkanmu (dipotong)" (HR. Muslim).

"... dan jangan kamu berputus asa dari rahmat Allah. Sesungguhnya tiada berputus asa dari rahmat Allah, melainkan kaum kafir"(Qs. Yusuf:87).

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdullilah, penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat, hidayah dan karunia-Nya, sehingga laporan tugas akhir ini dapat penulis selesaikan. Tak lupa shalawat dan salam penulis haturkan kepada junjungan kita Nabi Muhammad SAW, yang telah membawa kita dari zaman jahiliyah menuju zaman terang benderang seperti sekarang ini.

Tugas akhir ini dibuat sebagai satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana di Jurusan Teknik Informatika Universitas Islam Indonesia. Adapun tugas akhir ini dilakukan penulis berjudul "Analisis dan Perancangan Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Root*".

Pelaksanaan Tugas Akhir ini merupakan salah satu mata kuliah wajib jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia. Selain itu merupakan sarana bagi penulis untuk menambah pengetahuan dan pengalaman dalam menerapkan keilmuan. Sesuai yang sudah diambil di masa bangku perkuliahan.

Dalam pembuatan tugas akhir ini, penulis mendapatkan bantuan, dan yang lainnya untuk mengerjakan tugas akhir ini, oleh karena itu pada kesempatan kali ini penulis ingin menyampaikan terima kasih kepada :

1. Fathul Wahid, S.T., M.Sc., Phd.D., selaku Rektor Universitas Islam Indonesia.
2. Prof. Dr. Ir. Hari Purnomo, M.T., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Hendrik, S.T., M.Eng., selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku ketua Program Studi Informatika-Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Fietyata Yudha S.Kom., M.kom., selaku Dosen Pembimbing Tugas Akhir yang selalu memberikan dukungan, ilmu, semangat dan kesempatan selama pelaksanaan Tugas Akhir.
6. Hendrik, S.T., M.Eng., selaku Dosen Pembimbing Akademik yang selalu membimbing kepada hal yang benar.
7. Bapak dan Ibu Dosen Informatika yang telah memberikan ilmu, wawasan, dan pengetahuan yang bermanfaat selama masa perkuliahan.

8. Kedua Orang Tua saya, Almarhum Bapak Didi Sutardi dan Ibu Elin Anita yang telah memberikan dukungan, semangat, kasih sayang, dan doa yang terbaik.
9. Kedua Kakak perempuan saya, Novie Yuanita Indah Lestari dan Julian Dwi Kusuma Indah Lestari yang juga selalu memberikan semangat, kasih sayang dan doa yang terbaik.
10. Teman dan Sahabat yang selalu ada di saat suka dan duka.
11. Semua pihak yang telah banyak membantu dalam proses penyelesaian Tugas Akhir dan tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa Tugas Akhir ini memiliki beberapa kekurangan dan ketidaksempurnaan. Oleh karena itu, kritik dan saran sangat penulis harapkan sebagai bahan evaluasi untuk menjadi lebih baik lagi di masa yang akan dating. Akhir kata, semoga Tugas Akhir ini dapat diterima dengan baik dan bermanfaat sesuai dengan kebutuhannya.

Yogyakarta, 21 Agustus 2020

(Insan Fadhilah)

SARI

Perangkat ponsel android sudah digunakan mayoritas masyarakat pengguna ponsel pintar di Indonesia. Penggunaan perangkat ponsel pintar sebagai alat komunikasi dan kegiatan lainnya dalam keseharian yang dapat membantu pengguna perangkat ponsel menyelesaikan berbagai macam hal sesuai kebutuhannya, sayangnya beberapa pengguna ponsel pintar tidak menggunakan untuk hal-hal yang positif, tetapi mereka menggunakannya untuk hal-hal negatif, seperti melakukan tindak kejahatan atau tindakan yang melanggar hukum. Ketika smartphone tersebut sudah digunakan untuk tindakan yang melanggar hukum.

Dalam dunia *mobile forensic* salah satu cara untuk mengamankan barang bukti digital (*digital evidence)* adalah *disk imaging* atau lebih sering dikenal *disk cloning. Disk imaging* adalah suatu proses menyalin seluruh konten yang ada pada perangkat ponsel android yang memiliki sebuah media penyimpanan yang akan dianalisis sesuai dengan keadaan perangkat media penyimpanan pada saat ditemukan dan mengurangi resiko berubahnya struktur atau konten dari *disk image* ketika dilakukannya analisis.

Berdasarkan uraian masalah di atas, maka peneliti membuat sebuah aplikasi untuk mengakuisisi media penyimpanan dengan metode *Live Imaging* yang memanfaatkan *root* pada pada perangkat ponsel android untuk mengamankan barang bukti digital berupa *file* image yang dapat dianalisis secara menyeluruh untuk kepentingan peradilan.

Kemudian peneliti membuat beberapa pengujian terhadap aplikasi diantaranya pengujian dengan menggunakan metode *blackbox testing*, pengujian *error handling*, pengujian oleh pakar yang di uji oleh bapak Yusuf Hadiwinata Sutandar, RHCT., RHCVA., RHCI, RHCX., RHCSA., RHOS., CEI., CEH., CHFI., CND., EDRP., CCNA., MTCNA. Beliau merupakan seorang Vice President Operation & Services PT. Biznet Gio Nusantara selain Vice President Operation & Service di PT. Biznet Gio Nusantara beliau juga seorang ketua dalam komunitas yang bernama Forensika id, kemudian pengujian berupa kuesioner melalui *google form* yang disebarkan kepada responden untuk menanggapi dari aplikasi yang dihasilkan lalu dihitung menggunakan skala linkert, pengujian intregitas data, dan yang terakhir penulis melakukan pengujian performa pada saat apikasi dijalankan dengan dua skenario yaitu mengakuisisi media penyimpanan secara bersamaan dan mengakuisisi salah satu media penyimpanan.

Dengan aplikasi ini diharapkan mampu untuk melakukan akuisisi terhadap perangkat media penyimpanan pada ponsel android baik itu media penyimpanan internal maupun

eksternal dengan penggunaan *Root* di dalam ponsel. Sedangkan untuk hasil yang diperoleh aplikasi berupa *file image* dan *file log* tersebut dapat dianalisis dan dijadikan barang bukti digital dalam proses peradilan.

Kata kunci : *imaging android, forensic imaging android, application imaging, disk cloning*

GLOSARIUM

Mobile Forensics	Forensika Digital pada perangkat ponsel.
Forensic Imaging	Penggandaan media penyimpanan
Disk Imaging	Penggandaan media penyimpanan pada suatu perangkat
Storage	Media penyimpanan data digital
Hashing md5	Pencocokan md5 file
Filesystem	system file yang digunakan pada perangkat ponsel
Root	Hak super user dari suatu perangkat

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

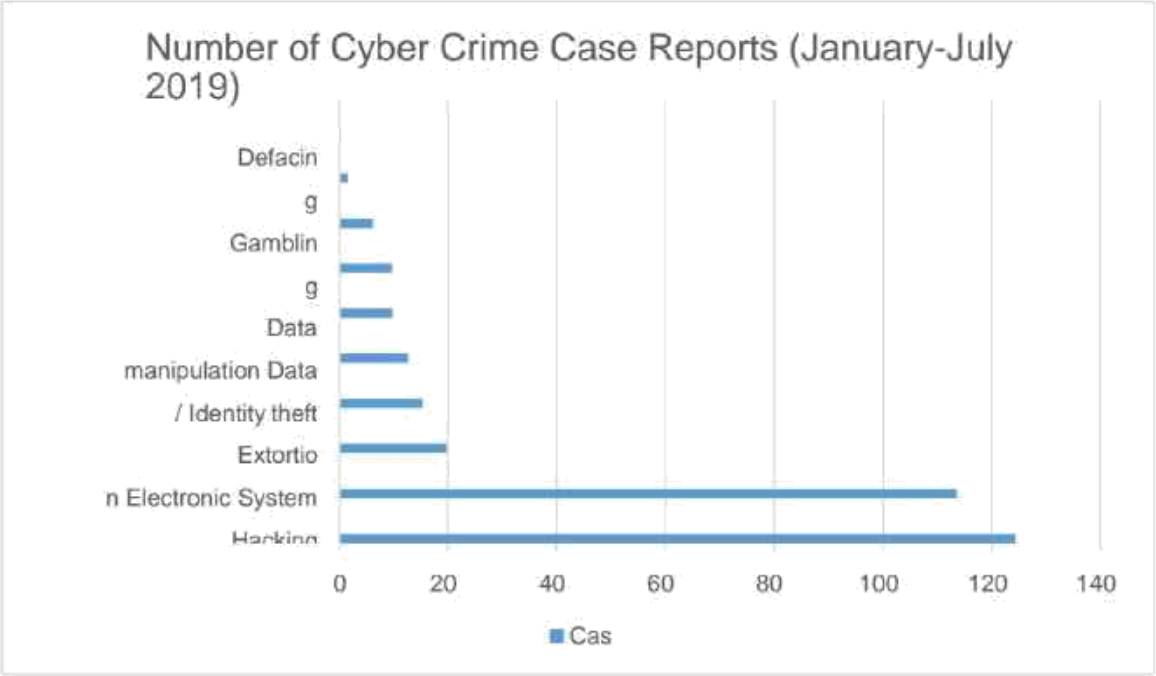
BAB I
PENDAHULUAN

1.1 Latar Belakang

Android merupakan salah satu sistem operasi yang digunakan pada smartphone Android dibangun berdasarkan kernel Linux dan beberapa software open source yang didesain focus untuk perangkat layar sentuh seperti smartphone dan tablet. Sebagai sistem operasi yang Open Source, Google merilis lisensi perizinan pada android yang memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi (Shankland, 2007).

Saat ini pengguna ponsel pintar di dunia terus meningkat dari tahun ke tahun, pada tahun 2019 pengguna ponsel pintar naik 5.6 persen dari tahun sebelumnya yaitu 3,2 miliar pengguna ponsel pintar di dunia. Sementara perangkat aktif yang digunakan mencapai 3,8 miliar unit (katadata.co.id, 2020). Sedangkan dilihat dari segi sistem operasi, Android telah menguasai lebih dari 87 persen pangsa pasar sistem operasi ponsel pintar dunia. data IDC menunjukan bahwa dari 344,7 juta penjualan ponsel pintar di dunia pada triwulan dua 2016, sebanyak 301,8 juta atau sekitar 87,7 persen menggunakan sistem operasi android besutan Google Inc, di posisi kedua iOS yang merupakan sistem operasi ponsel pintar iPhone menguasai 11,7 persen (katadata.co.id, 2016).

Sayangnya beberapa pengguna smartphone tidak menggunakan untuk hal-hal yang positif, tetapi mereka menggunakannya untuk hal-hal negatif, seperti melakukan tindak kejahatan atau tindakan yang melanggar hukum. Ketika telepon pintar tersebut sudah digunakan untuk tindakan yang melanggar hukum, maka bisa dijadikan bukti oleh para aparat penegak hukum untuk menjerat para pelaku kejahatan. Akan tetapi, bukti-bukti seperti SMS, daftar panggilan, chat media sosial yang dapat digunakan sebagai bukti bisa saja dihapus oleh para pelaku untuk menghilangkan jejak atau bukti perbuatannya. Maka perlu dilakukan sebuah proses akuisisi agar barang bukti dapat diperoleh untuk dilakukan sebuah penyelidikan.



Gambar 1.1 Data Kejahatan smartphone (katadata.co.id,2019)

Pada Gambar 1.1 adalah statistik data kejatahan dengan media ponsel pintar, Direktorat Tindak Pidana Siber Bareskrim Polri menerima setidaknya sejumlah 3.130 laporan kasus kejatahan siber sepanjang periode bulan januari sampai juli tahun 2019. laporan kasus kejahatan siber tentang penipuan online paling mendominasi yaitu sebanyak 1.243 kasus.

Sistem akuisisi data dapat didefinisikan sebagai suatu sistem yang berfungsi untuk. akuisisi/ data dapat mengubah output yang belum diolah dari satu atau lebih sensor ke dalam sinyal digital yang ekuivalen untuk dipakai pada proses lebih lanjut, seperti kendali dan aplikasi display (Tompkins and Webster, 1988; Vudhivanich and Sriwongsa, 2011).

Berdasarkan penjelasan di atas dapat disimpulkan bahwa perangkat ponsel android sangat berpotensi sebagai salah satu objek penelitian digital forensik yang menghasilkan barang bukti digital dari sebuah perangkat dan perangkat tersebut adalah bukti dari kasus, sedangkan penggunaan root mempunyai beberapa kelebihan salah satunya adalah mengamankan barang bukti digital secara real-time saat dilakukan proses pencitraan penyimpanan perangkat ponsel android. sehingga penelitian ini dilakukan untuk mengembangkan aplikasi dengan menggunakan metode Live Imaging Acqusition untuk ekspolarsi barang bukti digital pada media penyimpanan ponsel android yang memanfaatkan root pada perangkat ponsel android, dengan melakukan pengembangan dalam tampilan aplikasi untuk menampilkan progress bar

secara real time saat melakukan proses akusisi dan juga menjaga intregritas data barang bukti digital. Root pada ponsel android berfungsi pada proses akuisisi media penyimpanan ponsel android untuk melakukan akuisisi secara real time dan kondisi ponsel android dalam keadaan menyala.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas, maka merumuskan beberapa masalah sebagai berikut :

1. Bagaimana mengembangkan aplikasi yang dapat melakukan *forensic imaging* pada penyimpanan ponsel android?
2. Bagaimana melakukan *Forensic Imaging* pada telepon pintar berbasis Android dengan memanfaatkan metode *Root* pada ponsel Android?

1.3 Batasan Masalah

Adapun batasan-batasan masalah pada penelitian adalah sebagai berikut:

a. Penyimpanan yang terdapat dalam ponsel android (Unified Storage) merujuk untuk penyimpanan internal pada partisi "mmcblk0" sedangkan untuk penyimpanan eksternal pada partisi "mmcblk1" yang terdapat pada perangkat Android.
b. Perangkat Android harus terpasang aplikasi Kingoroot yang akan digunakan untuk mendapatkan hak akses root.
c. Perangkat Android harus ter-install Aplikasi *BusyBox* yang akan dibaca oleh *Root*.
d. *USB debugging* dalam keadaan aktif.
e. Menjalankan proses akuisisi untuk 1 perangkat ponsel Android.
f. Aplikasi hanya bisa dijalankan di Sistem operasi Linux.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

a. Mengembangkan aplikasi dengan menampilkan berjalannya perkembangan proses akuisisi secara real-time berupa *progress bar* dalam tampilan aplikasi, dapat mengakuisisi penyimpanan internal maupun eksternal ponsel android, dan memanfaatkan *Root* pada ponsel Android.

b. Mengakuisisi penyimpanan ponsel android yang sudah terpasang *Root* pada ponsel dan ponsel android dalam kondisi menyala.

1.5 Manfaat Penelitian

Dengan dilakukannya penelitian ini, penulis berharap dapat memberikan beberapa manfaat, dengan rincian sebagai berikut :

Bagi peneliti :

a. Mendapatkan ilmu tentang forensika digital pada ponsel Android
b. Mempelajari bahasa pemrograman python

Bagi Masyarakat/Seseorang dalam bidang forensika digital :

a. Aplikasi dapat dipergunakan dengan baik yang dapat mengekstraksi data penyimpanan internal maupun eksternal pada ponsel Android yang akan menghasilkan keluaran file berupa disk image yang akan di analisis dan investigasi oleh forensika.

1.6 Metodologi Penelitian

1. Pengumpulan Data

   Pada tahapan ini dilakukan pengumpulan data untuk mengetahui apa saja yang dibutuhkan untuk melakukan proses *forensic imaging* pada perangkat android, meliputi kebutuhan data, kebutuhan antarmuka dan tools sebagai referensi atau rujukan untuk menganalisa tampilan GUI yang akan dibuat

2. Desain

   Pada tahapan ini membuat rancangan aplikasi Forensic Imaging dengan memanfaatkan *Root* pada perangkat android yang ingin dibuat dengan membuat alur untuk aplikasi GUI dalam bentuk *flowchart* dan pengembangan aplikasi melalui proses analisis model penyimpanan data pada ponsel Android. Pada proses ini dilakukan untuk mengetahui model penyimpanan yang terdapat pada perangkat Android.

3. Implementasi

   Pada tahap ini merupakan membangun aplikasi sesuai kebutuhan dan aplikasi dibuat berdasarkan rancangan yang telah dibuat sebelumnya pada tahap ini akan

diimplementasikan menggunakan bahasa pemrograman *python* serta modul-modul *python* yang dibutuhkan dalam pembuatan aplikasi.

4. Pengujian

   Untuk mengetahui dan menilai keberhasilan dari implementasi maka dilakukan pengujian terhadap aplikasi yang telah dibuat dan pengujian dilakukan untuk menghasilkan berupa disk image dalam bentuk file, membandingkan hasil dari aplikasi dan sumber.

1.7 Sistematika Penulisan

Agar Memudahkan pembaca memahami permasalahan yang akan dibahas pada tulisan ini secara menyeluruh, maka pada laporan tugas akhir ini dibagi menjadi lima bab. Untuk sistematika penulisan laporan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bagian ini menjelaskan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bagian tentang semua yang mendasari dalam penelitian, meliputi konsep-konsep yang akan dipergunakan untuk memecahkan masalah dalam penelitian ini. Bahasan yang akan dibahas pada bab ini yaitu penelitian sebelumnya, digital forensics,mobile forensics, forensics imaging dan lainnya.

BAB III METODOLOGI

Pada bab ini memuat tentang teori-teori yang dipergunakan sebabai landasan-landasan untuk memecahkan masalah yang akan dibahas pada penelitian ini.

Metodologi yang akan dibahas pda bab ini meliputi analisis kebutuhan system dan perancangan, perancangan yang akan dibahas antara lain meliputi *flowchart*, dan rancangan antarmuka

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini memuat uraian hasil dan pembahasan tentang implementasi sistem, tahap proses pembuatan berdasarkan rancangan yang telah dibuat sebelumnya. Bab ini juga membahas beberapa hasil implementasi dari scanning ponsel android, mengkoneksikan ponsel dan proses akuisisi perangkat android dalam pengujian aplikasi dan analisis file.

BAB V KESIMPULAN DAN SARAN

Pada bab ini memuat tentang kesimpulan-kesimpulan dari hasil perancangan.

BAB II
LANDASAN TEORI

2.1 Penelitian Sebelumnya

Penelitian ini membutuhkan landasan teori untuk mendapatkan informasi-informasi guna untuk mendukung dan menunjang dalam menyelesaikan penelitian ini, informasi yang didapatkan dari penelitian sebelumnya telah dilakukan dan memiliki memiliki pembahasan yang sesuai dengan penelitian ini, penelitian tersebut antara lain :

Data acquisition techniques in mobile forensik adalah penelitian yang dilakukan oleh (Sneha C Sathe,& Nilima M Dongre, 2018) memperkenalkan teknik akuisisi logis dan fisik yang digunakan oleh penguji forensik dan karenanya memberikan studi perbandingan yang tekniknya memberikan pendekatan yang lebih baik dalam memperoleh bukti digital pada ponsel. Selain itu melakukan penelitian eksperimental pada smartphone android Samsung Galaxy Grand Duos GT-I9082 dan mencoba memperoleh bukti berdasarkan teknik performa terbaik dari perbandingan di atas.

Mobile Device Forensics: Extracting and Analysing Data from an Android-Based Smartphone (Normaziah A. Aziz, Fakhrulrazi Mokhti M,&Nadhar M. Nozri, 2015) mempelajari dan bereksperimen beberapa metode tentang bagaimana data dalam ponsel cerdas dapat di ekstraksi dan dianalisis dengan menggunakan tools Sleuthkit Autopsy.

Ada beberapa penelitian yang dapat digunakan sebagai pendukung rujukan dari penelitian ini dapat dilihat pada tabel 2.1.

Tabel 2.1 Tabel Penelitian Lainnya

NO	JUDUL	PENULIS	PENCAPAIAN	SARAN

1.	Android Forensics:Tools and Techniques for manual Data Extraction	(Animesh Kumar Agrawal, Arman Sharma, Pallavi Khatri)	Penelitian ini membuat metodologi untuk menganalisis dan memeriksa ponsel android dengan cara yang mudah dan sederhana tanpa bantuan alat komersial.	penelitian ini hanya mengakuisisi data pada perangkat android dengan cara manual, kedepannya dibuatkan aplikasi untuk mengakusisi ponsel android terotomatisasi
2.	Logical acquisition and analysis of data from android mobile devices.	Himanshu Srivastava Shashikala Tapaswi	Tujuan dari makalah ini adalah untuk mengusulkan pendekatan yang membantu dalam perolehan data langsung serta data yang disimpan dalam memori internal / eksternal perangkat mobile android mengingat bahwa data pada perangkat tidak banyak berubah selama proses ekstraksi.	aplikasi ini hanya mengakuisisi data logical dari perangkat ponsel android, untuk kedepannya penelitian ini dapat dikembangkan lagi.

3.	Forensics data acquisition methods for mobilephones	Khawla Abdulla, Andy Jones, T.A Martin	Di makalah ini penulis telah melakukan survei tentang metode akuisisi data saat ini. Kemudian, peneliti memberikan analisis komparatif antara metode akuisisi data saat ini.	kedepannya jurnal ini dapat dikembangkan lagi sehingga menjadi bahan pembelajaran untuk memahami metode akusisi data saat ini.
4.	A Study on Mobile Forensic Data Acquisition Method Based on Manufacturer's Backup Mobile App.	Choi Jaewon, Kim Seung-joo	Dalam tulisan ini, penulis menjelaskan proses memperoleh data menggunakan aplikasi seluler cadangan yang disediakan oleh pabrikan tanpa mengorbankan integritas ponsel cerdas terbaru.	penelitian ini dapat dibuatkan aplikasi dengan memanfaatkan seluler cadangan yang disediakan oleh pabrik.

Dalam penelitian ini terdapat perbedaan dan persamaan dengan penelitian sebelumnya, diantara lain dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Tabel perbedaan dan persamaan dengan penelitian sebelumnya

No	JUDUL	PENCAPAIAN	PERSAMAAN	PERBEDAAN

1	Android Forensics:Tools and Techniques for manual (Animesh Kumar Agrawal,Arman Sharma,Pallavi Khatri)	Penelitian ini membuat metodologi untuk menganalisis dan memeriksa ponsel android dengan cara yang mudah dan sederhana tanpa bantuan alat komersial.	Penggunaan root yang berfungsi mengakuisisi penyimpanan pada ponsel android	Penelitian sebelumnya menggunakan Android Tamer sebagai mesin virual untuk mengakuisisi penyimpanan model android, sedangkan penelitan yang dilakukan penulis memanfaatkan ADB untuk proses akuisisi.
2.	Logical acquisition and analysis of data from android mobile devices. (Himanshu Srivastava,Shashikala Tapaswi)	Tujuan dari makalah ini adalah untuk mengusulkan pendekatan yang membantu dalam perolehan data langsung serta data yang disimpan dalam memori internal / eksternal perangkat mobile android mengingat bahwa data pada perangkat tidak banyak berubah selama proses ekstraksi.	Menggunakan ADB Deamon sebagai perantara akuisisi	Penelitian sebelumnya hanya mengakuisisi data logical yang tersimpan pada perangkat ponsel android sedangkan peneliti megakuisisi semua data baik itu data logical maupun data physical pada penyimpanan ponsel andoid baik itu internal maupun eksternal
3.	Forensics Data Acquisition Methods for Mobile Phones (Khawla Abdulla, Andy Jones,)	Di makalah ini penulis telah melakukan survei tentang metode	–	Penelitian ini lebih banyak melakukan akuisisi penyimpanan pada perangkat

		akuisisi data saat ini. Kemudian, peneliti memberikan analisis komparatif antara metode akuisisi data saat ini.		ponsel android secara manual dengan bantuan alat fisik seperti menggunakan flasher box,fernico ZRT dan lainnya.
4.	A Study on Mobile Forensic Data Acquisition Method Based on Manufacturer's Backup Mobile App.	Dalam tulisan ini, penulis menjelaskan proses memperoleh data menggunakan aplikasi seluler cadangan yang disediakan oleh pabrikan tanpa mengorbankan integritas ponsel cerdas terbaru.	–	Penelitian ini memanfaatkan fitur backup untuk proses akuisisi dan nantinya akan di analisis, sedangkan penelitian yang penulis lakukan mengakuisisi penyimpanan perangkat ponsel android dengan metode Live Imaging dengan memanfaatkan ADB Deamon dan Root untuk proses akuisisinya.

21 Analisis Perancangan

21.1 Pengertian Analisis

Analisis ini diumpamakan sebagai "membaca" sebuah teks. Proses tersebut akan menyambungkan berbagai tanda dan juga menempatkan tanda tersebut di dalam proses komunikasi yang dinamis. Tanda tersebut bisa dilihat dengan melalui pesan yang disampaikan dengan proses.(Robert J. Schreiter).

2.2.2 Analisis Data

Analisis data adalah proses mengatur urutan data, merorganisasikanya ke dalam suatu pola, kategori, dan satuan uraian dasar (Lexy J. Moleong, 2002).

Kegiatan dalam analisis data biasanya adalah mengelompokan sejumlah variabel dan jenis responden, mentabulasi data berdasarkan variabel dari seluruh responden, menyajikan data dari tiap variabel-variabel yang diteliti, melakukan sebuah perhitungan untuk menjawab rumusan masalah dan melakukan perhitungan untuk menguji hipotesis yang diajukan. Adapun tahapan-tahapan dalam analisis data adalah sebagai berikut :

1. Tahap pengumpulan data
2. Tahap editing, pada tahap ini yaitu memeriksa kelengkapan dan kejelasan mengenai pengisian instrumen pengumpulan data.
3. Tahap coding, maksud pada tahap ini adalah melakukan proses identifikasi dan proses klarifikasi dari tiap-tiap pernyataan yang terdapat pada instrumen pengumpulan data berdasarkan variabel yang sedang diteliti.
4. Tahap tabulasi, yaitu melakukan kegiatan mencatat ataupun memasukan data kedalam tabel-tabel induk dalam suatu penelitian.
5. Tahap pengujian hipotesis, tahap ini merupakan tahapan pengujian terhadap proposisi apakah ditolak atau bisa diterima dan memiliki makna atau tidak atas dasar hipotesis. Pada tahap ini sebuah keputusan akan dibuat.

2.2.3 Pengertian Perancangan

Perencanaan adalah suatu fungsi atau suatu tahapan yang sangat penting, sebuah rencana akan sangat mempengaruhi sukses atau tidaknya sebuah pekerjaan. Adapun pengertian perencanaan menurut (Kaufman, 1972) perencanaan adalah suatu proyeksi tentang apa yang diperlukan dalam rangka mencapai tujuan abash dan bernilai.

2.3 Digital Forensics

2.3.1 Pengertian Digital Forensics

Pengertian forensic secara umum adalah sebuah proses ilmiah untuk mengumpulkan, menganalisis dan menyajikan bukti pada pengadilan. Pada umumnya sebuah tahap forensic dilakukan dengan asumsi bahwa data-data yang telah dikumpulkan akan digunakan sebagai alat bukti di pengadilan.

Digital Forensic disebut juga sebagai ilmu forensik digital merupakan cabang ilmu dari ilmu foreksik meliputi pemulihan dan investigasi dari bahan yang ditemukan dalam perangkat digital berupa media penyimpananya. *Digital Forensic* adalah ilmu yang menganalisa barang bukti digital sehingga dapat dipertanggungjawabankan dipengadilan. barang bukti digital tersebut termasuk *handphone, notebook, server,* alat teknologi yang mempunyai media penyimpanan dan dapat dianalisa (Ruby Alamsyah, 2013).

Oleh karena itu setelah pengumpulan barang bukti para praktisi forensic menjaga dan mengontrol bukti tersebut untuk mencegah terjadinya modifikasi data. Istilah forensika digital pada awalnya identik dengan forensik komputer tetapi kini telah diperluas untuk menyelidiki semua perangkat yang dapat menyimpan data digital. Forensic digital diperlukan karena biasanya data yang ada di perangkat target dikunci, dihapus ataupun disembunyikan. Adapun tahap-tahap dalam digital forensik adalah dibawah berikut :

1. Assessment, pemeriksa computer forensic harus menilai bukti digital sepenuhnya dengan mematuhi ruang lingkup dari kasus untuk menentukan tindakan yang harus diambil.
2. **Acquisition** Secara alami, bukti digital rentan dan dapat diubah, rusak, atau dihancurkan oleh pemeriksaan atau penanganan yang tidak tepat. Pemeriksaan yang paling tepat dilakukan pada copy dari bukti asli tersebut. Bukti asli harus diperoleh dengan cara melindungi dan mempertahankan integritas dari bukti tersebut.
3. **Examination** Tujuan dari proses ini adalah untuk mengekstrak dan menganalisis bukti digital. Ekstrak disini mengacu pada proses pemulihan data (recovery data) dari sebuah media. Analisisnya mengacu pada penafsiran dari data dan menempatkannya dalam format logis dan berguna.
4. **Documenting and reporting** Tindakan dan observasi harus didokumentasikan selama proses forensic berlangsung. Hal ini termasuk dengan persiapan laporan tertulis dari temuan yang ada.

2.3.2 Mobile Forensics

Mobile device forensic adalah cabang yang berkaitan dengan pemulihan bukti digital atau data dari perangkat mobile dengan menggunakan teknik tertentu sesuai kaidah forensika digital, namun juga dapat berhubungan dengan perangkat digital yang memiliki media

penyimpanan seperti memori internal maupun eksternal dan kemampuan komunikasi (Rohit Tamma, 2015).

forensik mobile juga menggunakan prosedur yang sama dengan investigasi forensik umum, ada beberapa teknik yang perlu diikuti untuk melakukan standar penyelidikan di forensik mobile. metode penyelidikannya kurang lebih sama dengan investigasi digital (Yadi & Kunang, 2014).

2.3.3 Mobile Forensics pada Perangkat Android

Mobile forensics pada perangkat tertentu dengan sistem operasi tertentu yang terdapat pada perangkat tersebut. Pada perangkat ponsel yang bersistem operasi Android untuk melakukan proses forensika digital dan melakukan dengan prosedur umum dalam mobile forensics, memiliki beberapa tahapan untuk melakukan prosedur mobile forensics (Murphy, 2012).

2.4 Disk Imaging

Disk imaging adalah suatu proses dari file tunggal atau sebuah media penyimpanan yang mengandung isi lengkap dengan strukturnya yang kemudian diperbanyak dengan isi struktur yang sama persis atau serupa dari data yang asli tanpa selisih ukuran didalamnya. Dalam arti lain disk imaging adalah proses memetakan, menggandakan barang bukti dengan metode *bit by bit copy*. *disk image* adalah hasil dari melakukan proses *disk imaging* yaitu file penyimpanan yang berisi semua data tersimpan dari sumber barang bukti seperti smartphone (Garfinkel & Ph, 2009; Stone, 2006).

2.4.1 Format Raw Image (.dd)

Format Raw Image adalah salinan *bit-by-bit copy* dari raw image pada media sumber tanpa penambahan ataupun penghapusan. Image yang diproduksi dalam format raw tidak mengandung metadata apapun, namun metadata ini disimpan dalam file tambahan. alat seperti dd dan turunannya biasanya untuk membuat file image dalam format raw (forensicswiki.org, n.d.).

Dd (disk defition) berasal dari Jb Control Language IBM dan bisa digunakan untuk menduplikat, menyalin, dan menyalin media penyimpanan. Perintah ini dianggap penting oleh seorang *sysadmin* karena bermanfaat untuk mengatur data-data pada media penyimpanan.

2.5 Bahasa Pemrograman Python

Python adalah suatu bahasa pemrograman dinamis yang sering dipergunakan dalam pengembangan aplikasi dalam berbagai aspek, python pada dasarnya memiliki gaya penulisan yang mirip dengan *pseudocode*, perbedaanya hanyalah python hanya bisa dijalankan di komputer dan menampilkan hasil. bahasa pemrograman python mudah dimengerti dan tidak perlu *compiling* (Hall, 2015).

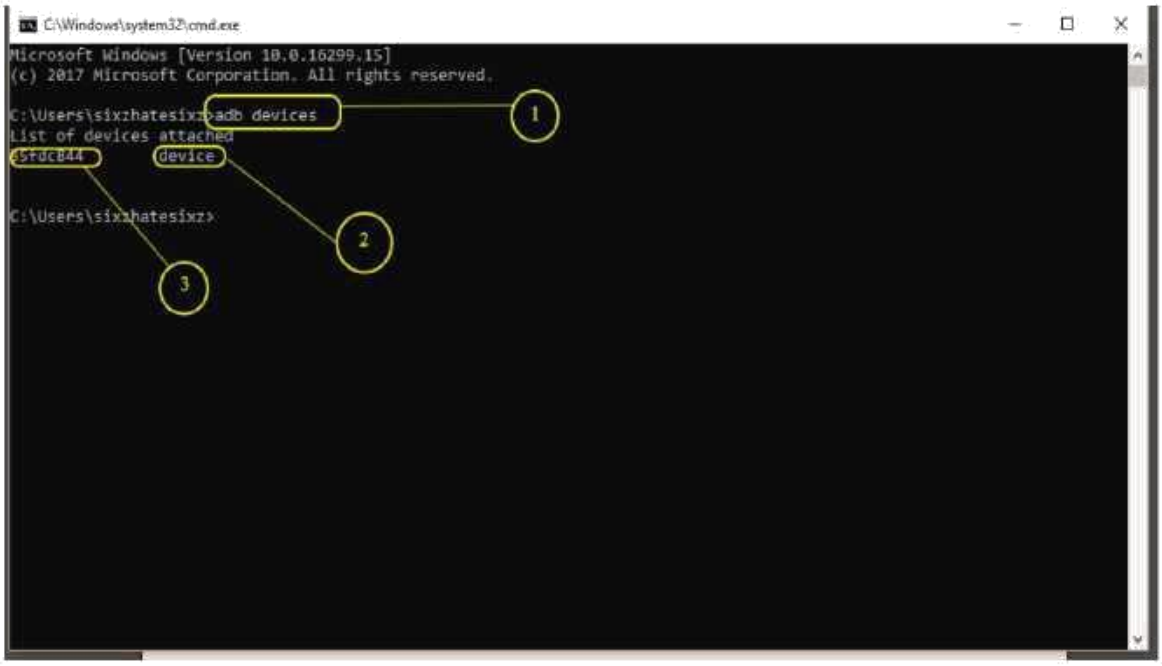
2.6 Root

Secara harfiah root dapat diartikan sebagai akar dalam bahasa Indonesia, root pada ponsel android adalah tindakan yang dapat memberikan akses penuh kepada pengguna untuk mengakses sistem maupun subsistem, dengan kata lain pengguna memiliki kekuasaan untuk mengakses dan mengeksekusi semua berkas dan semua perintah yang ada dalam ponsel android, singkatnya root memberikan kontrol tanpa batas kepada pengguna ponsel android untuk melakukan hal-hal yang diinginkan (Jurnalapps.co.id, 2018).

2.7 Analisis Model Penyimpanan Pada Perangkat Android

Metode Disk imaging adalah suatu proses menggandakan data atau file dari sebuah file penyimpanan pada ponsel android yang akan menghasilkan sebuah file image atau disebut juga sebagai disk image dan dapat dianalisis secara penuh. Proses analisis ini bertujuan untuk mengetahui beberapa jenis sejumlah file yang ada pada perangkat ponsel android seperti ukuran media penyimpanan, dan mengetahui file sistem apa saja yang ada pada perangkat tersebut (techtarget.com, n.d).

Untuk melakukan sejumlah analisis dibutuhkan sebuah perangkat ponsel android sebagai objek yang sudah terinstall busybox, memiliki hak istimewa root dan kondisi ponsel USB debugging aktif. Sedangkan untuk perangkat komputer maupun laptop yang dipergunakan untuk menganalisis perangkat ponsel tersebut harus terhubung dengan menggunakan kabel USB dan mempersiapkan alat Android Debugging Bridge atau biasa disebut ADB sebagai alat penghubung hak akses root pada perangkat android. Untuk melihat fungsi ADB dalam ponsel android dapat dilihat pada Gambar 2.1.



Gambar 2.1 Perangkat ponsel Android yang terhubung dengan komputer melalui ADB

Pada gambar 2.1 menjelaskan perintah (1)"adb devices" adalah perintah yang berfungsi menampilkan perangkat ponsel android yang terhubung ke perangkat komputer maupun laptop menggunakan kabel USB, sendangkan (2) "35fdc844" adalah nomor serial pada perangkat ponsel android yang terhubung dengan perangkat komputer ataupun laptop, dan (3) adalah keadaan "device" yang sudah terhubung dengan perangkat komputer maupun laptop.

BAB III
ANALISIS DAN PERANCANGAN

3.1 Alur Penelitian

Alur penelitian yang penulis lakukan ini bertujuan untuk mengembangkan Aplikasi *Forensic Imaging* pada perangkat ponsel Android dengan memanfaatkan *Root* memiliki tahapan-tahapan untuk membuat Aplikasi *Forensic Imaging*, tahapan-tahapan tersebut bisa dilihat di Gambar 3.1.

Gambar 3.1 Tahapan alur penelitian

Gambar 3.1 merupakan tahapan alur penelitian dimulai dari studi pustaka lalu selanjutnya menganalisis kebutuhan sistem untuk menentukan kebutuhan sistem yang dibutuhkan unutk menjalankan aplikasi, lalu tahapan ketiga adalah merancang desain sistem yang berisikan diagram alir dan *pseudocode*, lalu tahapan ke empat adalah merancang kebutuhan masukan dan keluaran aplikasi, tahapan setelah itu adalah pengujian aplikasi baik itu menyebarkan kuesioner, pengujian blackbox, pengujian error handling dan pengujian aplikasi oleh pakar, lalu tahapan terakhir itu hasil jadi aplikasi.

3.1 Analisis Kebutuhan Sistem

Untuk pengembangan aplikasi *Forensics Imaging* pada perangkat ponsel Android dengan memanfaatkan *Root* membutuhkan sebuah perangkat ponsel pintar berbasis android, serta memahami model penyimpanan data perangkat tersebut, aplikasi pendukung, dan beberapa modul bahasa pemrograman *python* agar dapat dilakukan perancangan, menjalankan dan pembuatan tampilan aplikasi. Untuk itu harus dilakukan analisis tahapan lanjut terhadap kebutuhan aplikasi tersebut seperti diantaranya analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak, analisis model penyimpanan data pada perangkat ponsel android, analisis kebutuhan masukan (*input)*, analisis kebutuhan proses, dan analisis kebutuhan keluaran (*output).*

3.1.1 Analisis Kebutuhan Perangkat Keras

Pada perancangan dan pembuatan sebuah aplikasi biasanya dibutuhkan beberapa komponen perangkat keras yang akan digunakan untuk membuat dan menjalankan sebuah aplikasi. perangkat keras yang dimaksud adalah suatu perangkat elektronik yang berperan sebagai wadah pembinaan maupun pengembangan aplikasi dan juga perangkat ponsel android untuk keperluan mengakuisisi penyimpanan datanya. Perangkat keras yang dibutuhkan pada penelitian ini diantaranya:

1. Perangkat laptop
2. Perangkat keras sebuah laptop dengan processor Intel i5 Gen 7 berkecepatan 2.5 Ghz dan RAM 8GB yang telah terinstall sistem operasi Kali Linux 64-bit sebagai perangkat keras yang peneliti gunakan untuk merancang, membuat dan menjalankan aplikasi.
3. Perangkat Ponsel Android
4. Perangkat keras android yang peneliti gunakan untuk mengakuisisi datanya bersistem operasi Android 4.3 yang memiliki RAM sebesar 512MB dan penyimpanan internal sebesar 4GB. Perangkat ini sudah mendukung dan ter-*install Root.*
5. Kabel USB
6. Perangkat keras selanjutnya adalah kabel USB yang berfungsi untuk menghubungkan perangkat keras android ke dalam perangkat laptop untuk kepentingan proses akuisisi penyimpanan data ponsel android.

3.1.2 Analisis Kebutuhan Perangkat Lunak

Pada perancangan dan pembuatan sebuah aplikasi biasanya selain dibutuhkan beberapa komponen perangkat keras yang akan digunakan untuk membuat dan menjalankan sebuah aplikasi dibutuhkannya juga sebuah perangkat lunak untuk membuat dan menjalankan sebuah aplikasi. perangkat lunak yang dimaksud adalah sebuah tools yang akan digunakan untuk membuat tampilan, menjalankan dan melakukan analisis. Perangkat tersebut diantaranya :

1. Kali Linux 64-bit

Kali linux 64-bit ini adalah sistem operasi yang digunakan untuk membuat dan menjalankan aplikasi forensic imaging pada ponsel android.

2. Aplikasi *Text Editor*

Aplikasi text editor adalah sebuah aplikasi yang memungkinkan pengguna untuk membuat,mengubah atau mengedit file text berupa text biasa, namun bisa juga digunakan

untuk membuat dan mengubah kode program. Secara umum aplikasi text editor ini ditujukan untuk mempermudah proses pemrograman.

3. Aplikasi Tkinter

Aplikasi Tkinter adalah aplikasi yang digunakan untuk merancang, mendesain dan membuat sebuah tampilan antarmuka yang ingin dibuat atau dikembangkan.

4. Bahasa Pemrograman *Python*

Bahasa pemrograman *Python* adalah bahasa yang akan digunakan untuk mengembangkan dan membuat aplikasi *Forensics Imaging* pada ponsel android dengan menggunakan bahasa pemrograman *Python* dan terhubung kedalam perangkat laptop.

5. Kingoroot

Kingoroot adalah sebuah tools atau aplikasi yang sudah terpasang pada perangkat ponsel android yang berguna untuk memberikan hak akses istimewa guna untuk kepentingan proses akuisisi.

6. Busybox Bionic

Busybox Bionic adalah sebuah tools atau aplikasi yang sudah terpasang pada perangkat ponsel android dengan fungsi untuk menjalankan perintah-perintah proses akuisisi dalam *shell* ponsel android.

7. ADB Daemon

ADB Daemon adalah sebuah tools atau aplikasi yang terdapat pada perangkat keras laptop yang berfungsi untuk mendapatkan hak akses *shell* pada perangkat ponsel android.

8. Netcat versi lorem ipsum

Netcat versi lorem ipsum adalah sebuah tools atau aplikasi yang sudah tersedia pada sistem operasi Kali Linux yang berfungsi untuk membuat *file imaging* yang dihasilkan pada proses akuisisi yang terjadi dalam aplikasi *Forensic imaging* yang telah dibuat. Tools ini berfungsi untuk mengambil data dari suatu port yang telah dilakukan dalam *shell* sebuah ponsel android dengan menggunakan tools *busybox*.

9. Autopsy

Autopsy adalah sebuah tools atau aplikasi yang berguna untuk membuka dan menganalisis hasil dari proses akuisisi pada ponsel android yang berisi data-data penyimpanan. Aplikasi autopsy berfungsi untuk mengekstrak *file image* menjadi susunan data-data yang berbentuk sesuai dari formatnya.

3.1.3 Analisis Kebutuhan Masukan

Didalam pembuatan aplikasi *forensics Imaging* ini dibutuhkan juga analisis kebutuhan masukan. Pada proses ini pengguna diminta untuk menghubungkan perangkat ponsel android ke komputer atau laptop yang akan digunakan dan sudah terinstall aplikasi menggunakan kabel USB, selanjutnya pengguna atau *user* membuka aplikasi dan mengkoneksikan perangkat android, setelah aplikasi terbuka pengguna akan dimintai memasukan nama file yang berguna untuk penamaan file yang akan dihasilkan oleh aplikasi, selain itu pengguna juga dimintai untuk memasukan nama *examiner* yang bertujuan untuk mengetahui siapa saja orang yang menggunakan aplikasi, *output location* berguna untuk menetapkan dimana *file image* akan disimpan setelah melakukan proses akuisisi, dan memilih media penyimpanan internal dan external pada perangkat ponsel android yang akan diakuisisi.

3.1.4 Analisis Kebutuhan Proses

Pada analisis kebutuhan proses untuk aplikasi *forensics imaging* pada perangkat android ini dengan memanfaatkan *root* dibutuhkan beberapa mekanisme proses diantaranya adalah proses *scanning* perangkat ponsel android, pada proses ini adalah proses aplikasi yang sudah ter*pasang* ke dalam perangkat laptop atau komputer mencari perangkat ponsel android yang sudah terkoneksi ke dalam perangkat laptop atau komputer yang akan dibaca oleh Adb Daemon, setelah melakukan proses *scanning* dan menemukan perangkat ponsel android yang terkoneksi, pada proses ini membutuhkan perangkat yang sudah ter-*install* root dan aplikasi akan mengkoneksikan perangkat melalui port *tcp:8888* sehingga aplikasi dapat melakukan proses akuisisi.

Setelah melakukan proses diatas dapat dilanjutkan ke proses akuisisi, proses akuisisi ini melalui 2 proses, yaitu proses pertama adalah proses yang terjadi dalam *shell* perangkat ponsel android yaitu proses pembuatan *file image* yang berformat (raw .dd) yang terjadi pada port *tcp:8888*, setelah proses tersebut selesai dilanjutkan ke proses yang kedua yaitu pembuatan *file image* yang akan dihasilkan ke dalam perangkat laptop atau komputer yang diambil melalui port *tcp:8888* melalui netcat.

3.1.5 Analisis Kebutuhan Keluaran

Analisis kebutuhan keluaran pada proses analisis kebutuhan keluaran ini aplikasi *forensics imaging* dengan memanfaatkan *root* pada perangkat ponsel android ini akan menghasilkan keluaran berupa *file imaging, file log* dan detail informasi proses akuisisi dalam aplikasi. untuk detail dari hasil keluaran tersebut sebagai berikut :

a. *File imaging* adalah file penyimpanan yang bersikan data-data yang tersimpan pada perangkat ponsel android baik itu internal maupun external yang sudah di ekstraksi. Peletakan *file image* terdapat pada folder yang sudah ditentukan di dalam kolom *output location* di dalam aplikasi.
b. *File log* adalah file text yang berisikan semua informasi mengenai proses akuisisi yang telah dilakukan didalam aplikasi. informasi tersebut diantaranya nama *examiner,* sumber direktori dari media penyimpanan ponsel android yang diakuisisi, ukuran penyimpanan baik internal maupun eksternal, informasi waktu mulai dan selesai proses akuisisi, dan nama *file image* yang dihasilkan.
c. Detail informasi proses akuisisi adalah sama halnya seperti *file log* yang membedakan hanya saja ditampilkan dalam aplikasi.

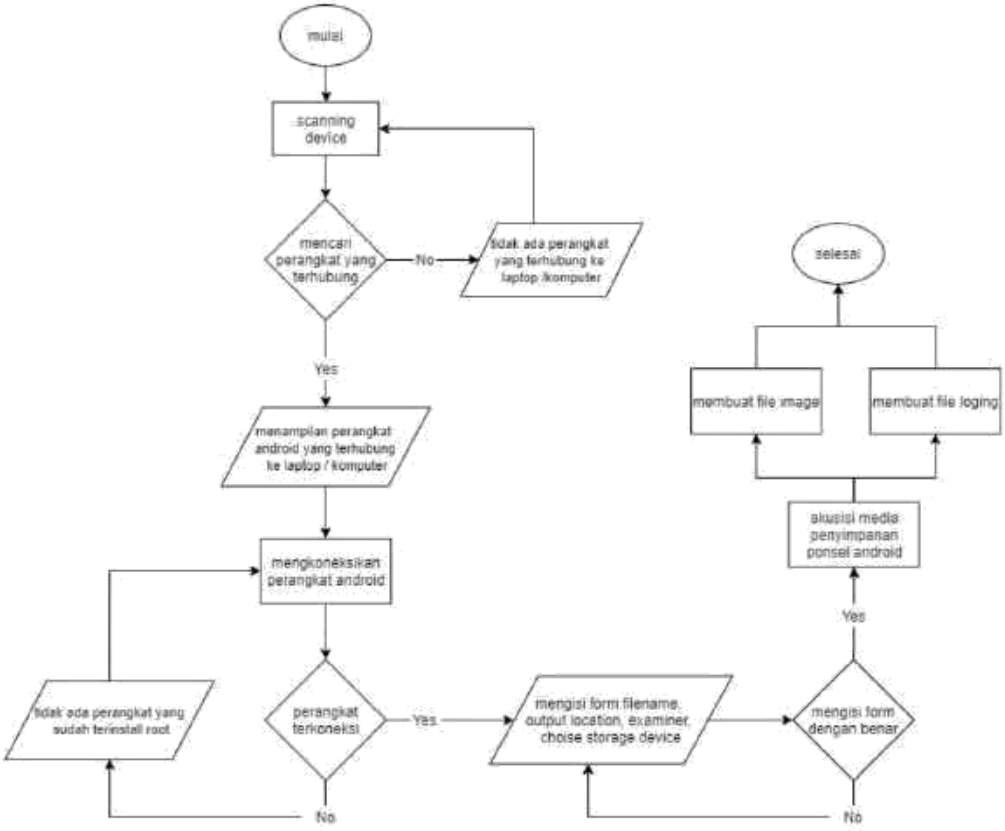
3.2 Perancangan Aplikasi

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi Perancangan sistem dapat dirancang dalam bentuk bagan alir sistem (*system flowchart*), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukan urutan-urutan proses dari sistem (Syifaun Nafisah, 2003 : 2). Perancangan aplikasi adalah suatu proses kegiatan untuk membuat konsep, dan mendesain alur kerja yang berguna untuk proses pembuatan aplikasi *Forensics imaging* pada perangkat android dengan memanfaatkan *root* sehingga aplikasi dapat berjalan dengan semestinya. Pada proses perancangan aplikasi ini meliputi 3 aspek rancangan yaitu *flowchart, pseudocode* dan desain antarmuka.

3.2.1 Diagram Alir

Flowchart adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. flowchart (bagan alir) adalah sebuah gambaran dari hasil

pemikiran dalam menganalisa suatu permasalahan dalam komputer. Karena setiap analisa akan menghasilkan hasil yang bervariasi antara satu dan lainnya, secara garis besar setiap perancangan flowchart selalu terdiri dari tiga bagian, yaitu input, proses dan output. Penulis membuat alur kerja (*flowchart*) sebagai langkah-langkah dalam proses penggunaan aplikasi dapat dilihat pada Gambar 3.2.



Gambar 3.2 Flowchart Aplikasi

Pada Gambar 3.2 adalah *Flowchart* yang menjelaskan cara kerja sistem. Ketika pengguna membuka aplikasi pertama kali, hal pertama yang harus dilakukan adalah pengguna melakukan *scanning* perangkat ponsel android yang terhubung pada perangkat laptop atau komputer, jika aplikasi menemukan perangkat yang terhubung pada laptop atau komputer maka aplikasi akan menampilkan *serial number* dan status perangkat android, dan jika aplikasi tidak menemukan perangkat yang terhubung pada komputer atau laptop maka aplikasi akan memberitahu bahwa tidak ada perangkat ponsel android yang terhubung pada komputer atau laptop. Selanjutnya pengguna diminta untuk mengkoneksikan perangkat ponsel android, jika tidak dapat terkoneksi dengan perangkat laptop atau komputer ada beberapa kemungkinan diantaranya adalah perangkat tidak ditemukan atau perangkat tidak terhubung pada perangkat komputer atau

laptop, kedua perangkat ponsel android belum ter-*install root*, jika perangkat ponsel android sudah ter-*install root* maka akan ditampilkan sebuah *serial number* pada perangkat android sudah terkoneksi melalui sebuah port dalam aplikasi. selanjutnya pengguna akan diminta untuk melanjutkan pada tampilan kedua aplikasi.

Setelah sudah masuk pada tampilan 2 pada Aplikasi *Forensics Imaging* pengguna akan diminta memasukan *file name, examiner,* output location untuk keluaran hasil berupa *file image* pada folder yang ditentukan, dan *radio box* untuk memilih media penyimpanan pada perangkat android. Jika pengguna memasukan semua dengan benar dan lengkap pada aplikasi dapat dilakukan ke proses akuisisi, dan jika pengguna tidak memasukan dengan benar dan lengkap maka pengguna maka tidak bisa dilanjutkan ke proses akuisisi. Jika aplikasi sudah selesai melakukan proses akuisisi maka aplikasi akan membuat *file log* di dalam folder yang sudah ditentukan pada *output location* bersamaan dengan *file image* dan menampilkan keseluruhan informasi pada *file log* yang akan menunjukan semua detail dari proses akuisisi.

3.2.2 Pseudocode

Pseudocode merupakan pengembangan dari algoritma, dimana sesuai dengan namanya pseudocode menggunakan kode-kode tertentu untuk memberikan penjelasan mengenai cara kerja atau penyelesaian dari suatu masalah, singkatnya pseudocode adalah sebuah algoritma yang sudah diubah menjadi kode-kode tertentu yang dapat dimengerti oleh orang awam yang memiliki struktur lebih ringkas dari algoritma biasanya. Pseudocode pada aplikasi *Forensics Imaging* pada ponsel Android dengan memanfaatkan *Root* terdiri dari 2 pseudocode yaitu :

a. Pseudocode Tampilan 1

Pseudocode tampilan 1 aplikasi Forensics Imaging pada ponsel Android dengan memanfaatkan Root menggunakan beberapa modul yang terdapat pada bahasa pemrograman Python yang memiliki fungsi masing-masing tiap modulnya. Untuk melihat modul tersebut bisa dilihat pada Tabel 3.1

Tabel 3.1 Tabel Modul pada Tampilan 1

No	Nama Modul	Fungsi Modul	Kegunaan
1	Sys	Sys	Untuk mengakses konfigurasi yang terdapat dalam kode program.

2	Subprocess	Getoutput	Menjalankan perintah adb dan mendapatkan keluaran
3	Tkinter	tkinter	Untuk membuat aplikasi.

Pada Tabel 3.1 merupakan tabel modul yang digunakan pada tampilan 1 aplikasi untuk membuat algoritma yang dapat menjalankan proses-proses yang dibutuhkan pada tampilan 1 aplikasi, modul tersebut memiliki fungsi yang sangat penting untuk menjalankan aplikasi dan pembuatan algoritma.

Tampilan 1 : Aplikasi Forensic Imaging pada perangkat android dengan memanfaatkan Root (Tampilan yang memiliki fungsi untuk Scanning, Install Root, install busybox dan mengkoneksikan perangkat)

Kamus :

PerintahADB : Adb Deamon
scanningPerangkat : String
InstallRoot : String
installBusyBox : String
koneksiPerangkat : String
statusPerangkat : String
checkRoot : String
checkBusyBox : String
subprocess(getoutput) : modul
subprocess : String

Deskripsi algoritma :

```
FUNCTION clk_scn(self):
   subprocess.getoutput('adb start-server')
   serial ← subprocess.getoutput('adb devices -l')
   forSerial ← subprocess.getoutput('adb get-serialno')
   forStatus ← subprocess.getoutput('adb get-state')
   serialNo ← forSerial
```

```
    status ← forStatus
    model ← serial[86:95]
    type ← serial[57:120]

    IF forSerial = 'unknown':
      OUTPUT(END, "No device found")
    ELSE:
      OUTPUT( "Device is Ready")
      OUTPUT("Serial No : "+serialNo")
      OUTPUT("Status : " + status")
      OUTPUT("Type : " + str(type)")
    ENDIF

FUNCTION clk_check(self):
    var ← subprocess.getoutput('adb forward --list')
    text ← "{var}".format(var=var)
    device ← text[0:16]
    port ← text[16:50]

    IF text = '':
      OUTPUT("No Connected Devices")
    ELSE:
      OUTPUT("Connected Device to : " + str(device))
      OUTPUT("Port : " + str(port))
    ENDIF

FUNCTION clk_root(self):
    subprocess.getoutput('adb install KingoRoot.apk')

FUNCTION clk_BusyBox(self):
    subprocess.getoutput('adb install BusyBox.apk')

FUNCTION clk_connect(self):
    var2 ← subprocess.getoutput("adb get-state")
    var4 ← subprocess.getoutput("adb shell su -c ls /data")
    var5 ← subprocess.getoutput("adb shell pm list packages -u stericson.busybox ")
    output ← "/system/bin/sh: su: not found"
    output1 ← "error: no devices/emulators found"
```

```
output2 ← "package:stericson.busybox"

IF var2 = 'device':
   IF var4 = output:
     OUTPUT("Devices has not installed Root, Please Install")
   ELSEIF var5 = "":
     OUTPUT( "Devices has not installed BusyBox, Please Install")
   ELSEIF var5 = output2:
     IF var4 = var4:
       var ← subprocess.getoutput("adb forward tcp:8888 tcp:8888")
       text ← "{var}".format(var=var)
       IF text = "":
          OUTPUT( "Devices Has Root"+"\n")
          OUTPUT( "Device Connected")
          nextButton(NORMAL)
       ENDIF
     ELSEIF var4 = output:
        OUTPUT( "Devices has not installed Root, Please Install")
     ENDIF
ELSEIF var2 = output1:
      OUTPUT("Device Not Connected")
ENDIF
```

Gambar 3.3 Pseudocode Tampilan 1 aplikasi

Pada Gambar 3.3 adalah *pseudocode* tampilan 1 aplikasi *Forensic Imaging* pada ponsel android berbasis *Root*, untuk penjelasanya pada variabel *scanningPerangkat* adalah variabel yang dapat menyimpan string dari suatu perintah pencarian ponsel android dalam command prompt, lalu untuk *statusPerangkat* merupakan variabel yang dapat menyimpan string dari suatu perintah untuk pengecekan ponsel android sudah terhubung atau belum dalam command prompt, lalu ada variabel *installRoot* yang berfungsi untuk menginstall aplikasi *Root* yang berguna untuk proses akuisisi, dan untuk variabel *installBusyBox* berfungsi untuk menginstall aplikasi *BusyBox* pada ponsel android, lalu ada variabel, lalu ada variabel *checkRoot* dan *checkBusyBox* yang berfungsi untuk menyimpan string dari suatu perintah untuk pengecekan ponsel android apakah ponsel tersebut sudah terinstall *Root* dan *BusyBox* atau belum, selanjutnya ada variabel *koneksiPerangkat* yang mempunyai untuk menjalankan suatu perintah untuk mengkoneksikan perangkat android dalam *command prompt*. Dari semua variabel menggunakan bantuan modul *subprocess dalam* bahasa pemograman *python*.

Untuk menjelaskan algoritmanya adalah pertama pengguna diminta untuk melakukan *scanning* ponsel android yang sudah terhubung ke laptop/host melalui adb daemon dengan menggunakan modul *subprocess* dalam bahasa pemrograman *python*, aplikasi akan menjalankan *subprocess* dan akan menghasilkan keluaran dan menampilkannya berupa *serial number*, status dan tipe ponsel tersebut. Lalu pengguna diminta untuk mengoneksikan perangkat android yang sudah terhubung pada laptop/host melalui Adb daemon dengan menggunakan modul *subprocess*. Tetapi untuk mengkoneksikan perangkat android ke laptop/host aplikasi memiliki beberapa syarat, yaitu ponsel android harus sudah diberikan izin untuk mengakses *superuser* dan sudah memasang *BusyBox*. Jika ponsel belum diberikan izin *root* dan menginstall *BusyBox* maka pengguna diminta untuk menginstall sebuah aplikasi guna untuk memberikan akses *root* dan pengguna juga diminta untuk menginstall aplikasi *BusyBox* yang berguna untuk proses akuisisi sebuah ponsel android dengan metode *Root*. Setelah ponsel android sudah diberikan izin untuk mengakses *superuser dan* sudah menginstall aplikasi *BusyBox* maka proses mengkoneksikan android bisa dijalankan. Proses mengkoneksikan ponsel android sama halnya dengan proses scanning dengan menggunakan modul *subprocess*, modul tersebut menghasilkan keluaran yang akan menghubungkan perangkat pada suatu port yaitu port tcp:8888, setelah menghubungkan perangkat dalam suatu port selesai aplikasi dapat menampilkan informasi tentang perangkat tersebut sudah terhubung dengan host/laptop melalui sebuah port dan juga pengguna bisa melanjutkan proses ke Tampilan 2 aplikasi.

b. Pseudocode Tampilan 2

Pseudocode tampilan 2 aplikasi *Forensics Imaging* pada ponsel Android dengan memanfaatkan *Root* menggunakan beberapa modul yang terdapat pada bahasa pemrograman *Python* yang memiliki fungsi masing-masing tiap modulnya. Untuk melihat modul tersebut bisa dilihat pada Tabel 3.2

Tabel 3.2 Modul Tampilan 2 pada Aplikasi

No	Nama Modul	Fungsi Modul	Kegunaan
1.	Sys	sys	Mengakses konfigurasi yang ada dalam kode program
2.	Subprocess	getoutput	Menjalankan perintah adb dan mendapatkan keluaran

3.	Tkinter	tkinter	Membuat tampilan aplikasi dan menampilkan beberapa keluaran yang dihasilkan aplikasi
4.	Process	process	Menjalankan beberapa proses secara bersamaan tanpa mengganggu proses lainnya
5.	Pyadb3	run_shell_cmd	Melakukan proses akuisisi di dalam *shell* perangkat ponsel android
6.	Hashlib	md5	Proses *hashing md5* pada *file image* hasil proses akuisisi
7.	Os	getsize	Mendapatkan ukuran *file image* saat proses akuisisi berjalan agar dapat menjalankan fungsi progress bar
8.	Logging	file log	Membuat *file log* yang .berisikan informasi-informasi dari proses akuisisi
9.	Time	sleep	Memberikan jeda waktu pada proses akuisisi penyimpanan pada perangkat android

Pada Tabel 3.2 merupakan tabel modul yang dipergunakan untuk Tampilan 2 aplikasi dalam membuat algoritma yang dapat menjelaskan proses-proses dalam pembuatan tampilan 2 aplikasi, modul tersebut mempunyai peran yang sangat penting dalam menjalankan aplikasi dan pembuatan algoritma.

Sedangkan untuk algoritma aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *Root*, dapat dijelaskan dengan *pseudocode*. Untuk *pseudocode* tampilan 2 dapat dilihat pada Gambar 3.4

```
FUNCTION selectOutput(self, window):
    folder_selected ← filedialog.askdirectory(title="Select output
    Folder") outputfolder ← folder_selected
```

```
FUNCTION FileName(self, window):
    Name ← tk.Label(window, text="File Name :")
    txtFileName ← tk.Entry(window,textvariable="")

FUNCTION CheckBox(self,
    window): CheckVar1 ←
    IntVar()
    CheckVar2 ← IntVar()
    C1 ← Checkbutton(window, text ← "Internal", variable ←
    CheckVar1) C2 ← Checkbutton(window, text ← "Eksternal",
    variable ← CheckVar2)

FUNCTION startdumpShell(self, window):
    command ← "su -c 'dd if=/dev/block/{} = busybox nc -l -p 8888'".format(
    choosenDir) dev ← adbutils.adb.device()
    dev.shell(command)

FUNCTION outputDumpShell(self,
    window): padb ← pyadb3.ADB()
    partitionList ← padb.run_shell_cmd("cat /proc/partitions = grep {}".format( choosenDir))
    partitionResult ← 0
    IF len(partitionList) > 0:
      partitionSplit ← partitionList.split()[2]
      partitionSize ← "{partitionSplit}".format(partitionSplit ← partitionSplit).replace("b", "").replace("",
"")
      partitionResult ← int(partitionSize)/ (1024 *

    1024) RETURN partitionResult

 FUNCTION hashmd5(self,
    window): time.sleep(5)
    fDirect ← outputfolder
    nDirect ← "/_Ext" + txtFileName.get() +".dd"
    alldirect ← fDirect + nDirect
    block_size ← 2 ** 20
    md5a ←
    hashlib.md5()
    IF( os.path.exists(alldirect) ) :
```

```
        with open(alldirect, 'rb') as fileCloning:
          while True:
            readCloning ←
            fileCloning.read(block_size) IF not
            readCloning:
              brea
            k
            ENDIF
            md5a.update(readCloning)
          ENDWHILE
      ENDIF

      hasilCloning ←
      md5a.hexdigest() adbHash ←
      pyadb3.ADB()
      coHash ← adbHash.run_shell_cmd("su -c md5sum /dev/block/{}".format(
      choosenDir)) splitHash ← "{coHash}".format(coHash=coHash)
      hasilHash ←
      splitHash[2:34]
      isHashMatched ← ""

      IF hasilHash = hasilCloning:
        isHashMatched ← "MD5 Hash
      Matched" ELSE:
        isHashMatched ← "MD5 Hash Not
      Matched" ENDIF

FUNCTION startOuputDumpInt(self,
      window): content ← txtFileName.get()
      process ← subprocess.run(['cd '+ outputfolder + '&&' + 'nc -v 127.0.0.1 8888 >'+ "_Int" + str(content)
                              + ".dd"], shell=True)

FUNCTION startOuputDumpExt(self,
      window): content ← txtFileName.get()
      process ← subprocess.run(['cd '+ outputfolder + '&&' + 'nc -v 127.0.0.1 8888 >'+"_Ext"+ str(content)
                              + ".dd"], shell=True)

FUNCTION startProcess(self,
      window): startTime ←
      datetime.now()
      sizeoutputInt ← 0
```

```
sizeoutputExt ← 0
IF ( CheckVar1.get() = 1 AND CheckVar2.get() = 1):
  choosenDir ← "mmcblk0"
  sizeoutputInt ← outputDumpShell(window)
  p ← Process(target= lambda: startdumpShell(window))
  q ← Process(target= lambda:
  startOuputDumpInt(window)) IF return_code !=0:
    p.start()
    time.sleep(6
    ) q.start()
  ELSE:
    p.terminate(
    )
    time.sleep(
    2)
    q.terminate()
  ENDIF

  time.sleep(3)

  alldirect ← "{}/_Int{}.dd".format( outputfolder, txtFileName.get())
  currentOutputSize ← os.stat(alldirect).st_size / (1024 * 1024 *
  1024) progressInt['maximum'] ← sizeoutputInt
  state ← True

  while state:
    state ← progressIntBar(window, currentOutputSize,
    sizeoutputInt) root.update_idletasks()
    currentOutputSize ← os.stat(alldirect).st_size / (1024 * 1024 *
  1024) ENDWHILE

  hashmd5(window)
  createLogInt()

  p ← Process(target= lambda: startdumpShell(window))
  q ← Process(target= lambda:

  startOuputDumpExt(window)) choosenDir ← "mmcblk1"
```

```
sizeoutputExt ← outputDumpShell(window)

IF return_code !=0:
  p.start()
  time.sleep(6
  ) q.start()
ELSE:
  p.terminate(
  )
  time.sleep(
  2)
  q.terminate()
ENDIF

time.sleep(3)

alldirect ← "{}/_Ext{}.dd".format( outputfolder, txtFileName.get())
currentOutputSize ← os.stat(alldirect).st_size / (1024 * 1024 *
1024) progressExt['maximum'] ← sizeoutputExt
state ←
True while
state:
  state ← progressExtBar(window, currentOutputSize, sizeoutputExt)
  root.update_idletasks()
  currentOutputSize ← os.stat(alldirect).st_size / (1024 * 1024 *
1024) ENDWHILE

hashmd5(window)
createLogExt()

ELSEIF CheckVar1.get() = 1:

p ← Process(target= lambda: startdumpShell(window))
q ← Process(target= lambda:
startOuputDumpInt(window)) choosenDir ← "mmcblk0"
sizeoutputInt ← outputDumpShell(window)

IF return_code
  !=0: p.start()
```

```
    time.sleep(6)
    q.start()
  ELSE:
    p.terminate()
    time.sleep(
    2)
    q.terminate()
  ENDIF

  time.sleep(3)
  alldirect ← "{}/_Int{}.dd".format( outputfolder, txtFileName.get())
  currentOutputSize ← os.stat(alldirect).st_size / (1024 * 1024 *
  1024) progressInt['maximum'] ← sizeoutputInt
  state ← True

  while state:
    state ← progressIntBar(window, currentOutputSize,
    sizeoutputInt) root.update_idletasks()
    currentOutputSize ← os.stat(alldirect).st_size / (1024 * 1024 *
  1024) ENDWHILE

  hashmd5(window)
  createLogInt()

ELSEIF  CheckVar2.get() = 1:
  choosenDir ← "mmcblk1"
  p ← Process(target= lambda: startdumpShell(window))
  q ← Process(target= lambda:
  startOuputDumpExt(window))
  outputDumpShell(window)
  sizeoutputExt ← outputDumpShell(window)

  IF return_code
    !=0: p.start()
    time.sleep(6)
    q.start()
  ELSE:
    p.terminate()
```

```
    time.sleep(2
    )
    q.terminate()
ENDIF

time.sleep(3)
alldirect ← "{}/_Ext{}.dd".format( outputfolder, txtFileName.get())
currentOutputSize ← os.stat(alldirect).st_size / (1024 * 1024 *
1024) progressExt['maximum'] ← sizeoutputExt
state ← True

while state:
  state ← progressExtBar(window, currentOutputSize, sizeoutputExt)
  root.update_idletasks()
  currentOutputSize ← os.stat(alldirect).st_size / (1024 * 1024 * 1024)
ENDWHILE

hashmd5(window)
createLogExt()
```

Gambar 3 4 Pseudocode tampilan 2 aplikasi

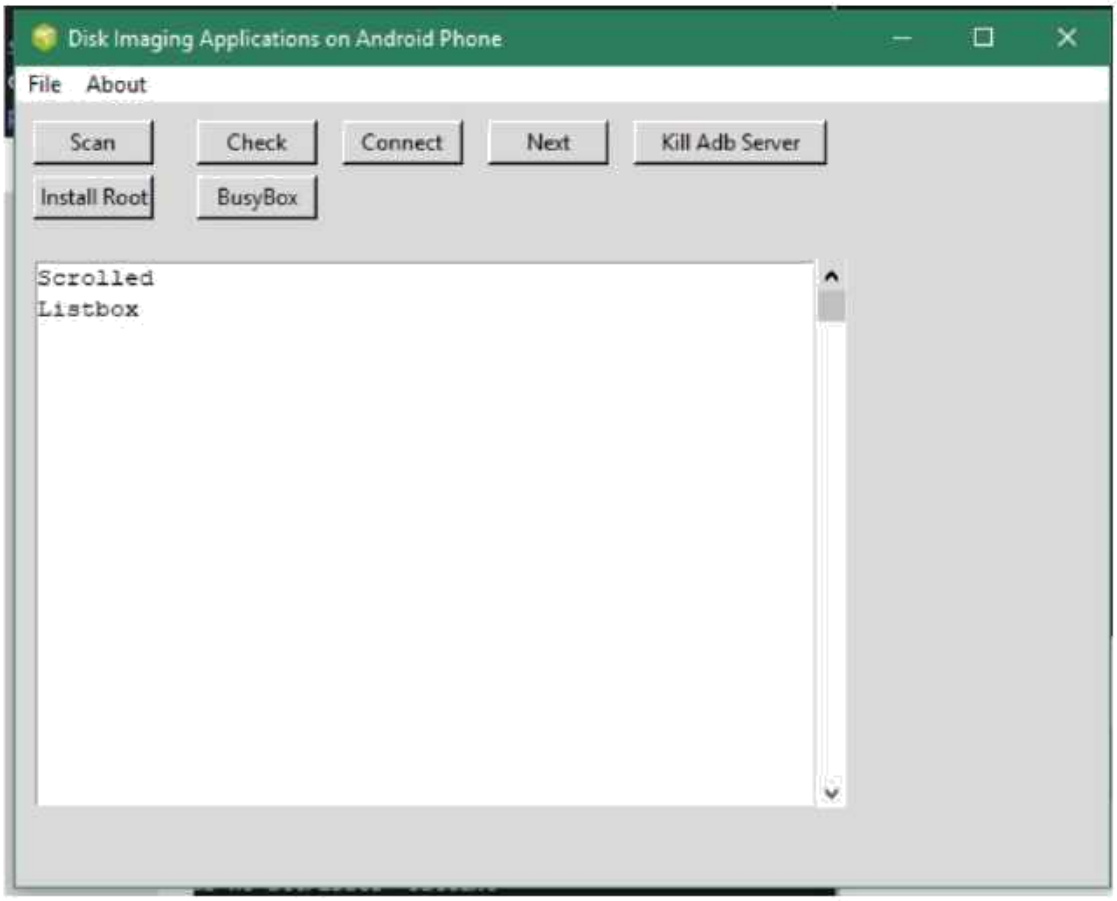
Pada Gambar 3.4 adalah pseudocode tampilan 2 pada aplikasi ForensicsImaging pada ponsel android dengan memanfaatkan root, untuk mendeskripsikan algoritmanya, bahwa algoritma diatas menunjukkan ada class function yang berguna untuk menjalankan semua proses akuisisi media penyimpanan pada perangkat ponsel android dalam aplikasi dan proses berjalannya progress bar untuk mengetahui perkembangan proses akuisisi secara langsung, untuk penjelaskan dapat dijelaskan sebagai berikut :

Class Function selectOutput menjelaskan fungsi untuk memilih folder keluaran file image yang dihasilkan aplikasi, function FileName menjelaskan fungsi untuk menentukan nama keluaran file image yang dihasilkan aplikasi, selanjutnya ada function startDumpShell yang menjelaskan fungsi untuk menjalankan proses akuisisi yang terjadi di dalam shell ponsel android baik itu media penyimpanan internal maupun eksternal di dalam modul ini membutuhkan hak akses shell pada perangkat ponsel android. Modul tersebut adalah modul adbutils yang ada didalam bahasa pemrograman python, modul tersebut memiliki banyak fungsi salah satu diantara fungsinya adalah mendapatkan hak akses dalam perangkat shell pada perangkat ponsel android melalui modul ini aplikasi menjalankan sebuah perintah dalam shell

pada perangkat ponsel android yang akan menjalankan sebuah protokol TCP dalam sebuah port untuk menjalankan proses akuisisi. Sedangkan function outputDumpShell menjelaskan fungsi untuk mengambil ukuran media penyimpanan baik itu media penyimpanan internal maupun eksternal guna untuk keperluan menjalankan progressbar di dalam aplikasi, dalam fungsi ini terdapat modul pyadb3 yang terdapat pada bahasa pemrograman python yang berfungsi untuk mendapatkan ukuran media penyimpanan pada perangkat ponsel android yang dijalankan pada shell perangkat ponsel android. Selanjutnya adalah function hashmd5 yang berfungsi untuk mengeluarkan hasil hashmd5 pada fileimage yang dihasilkan lalu hasilnya dicocokan dengan md5 dari sumber perangkat media penyimpanan baik itu internal maupun eksternal pada perangkat ponsel android. Selanjutnya ada function startOutputDumpInt yang berfungsi untuk menjalankan proses akuisisi pada media penyimpanan internal pada perangkat ponsel android dengan bantuan sebuah tool yang bernama etcat. Sedangkan function startOutputDumpExt berfungsi untuk menjalankan proses akuisisi pada media penyimpanan eksternal pada perangkat ponsel android dengan bantuan aplikasi pada perangkat laptop/host yang bernama netcat. Selanjutnya function terakhir yaitu function startProsess yang memiliki fungsi untuk menjalankan semua class function yang sudah dijelaskan diatas.

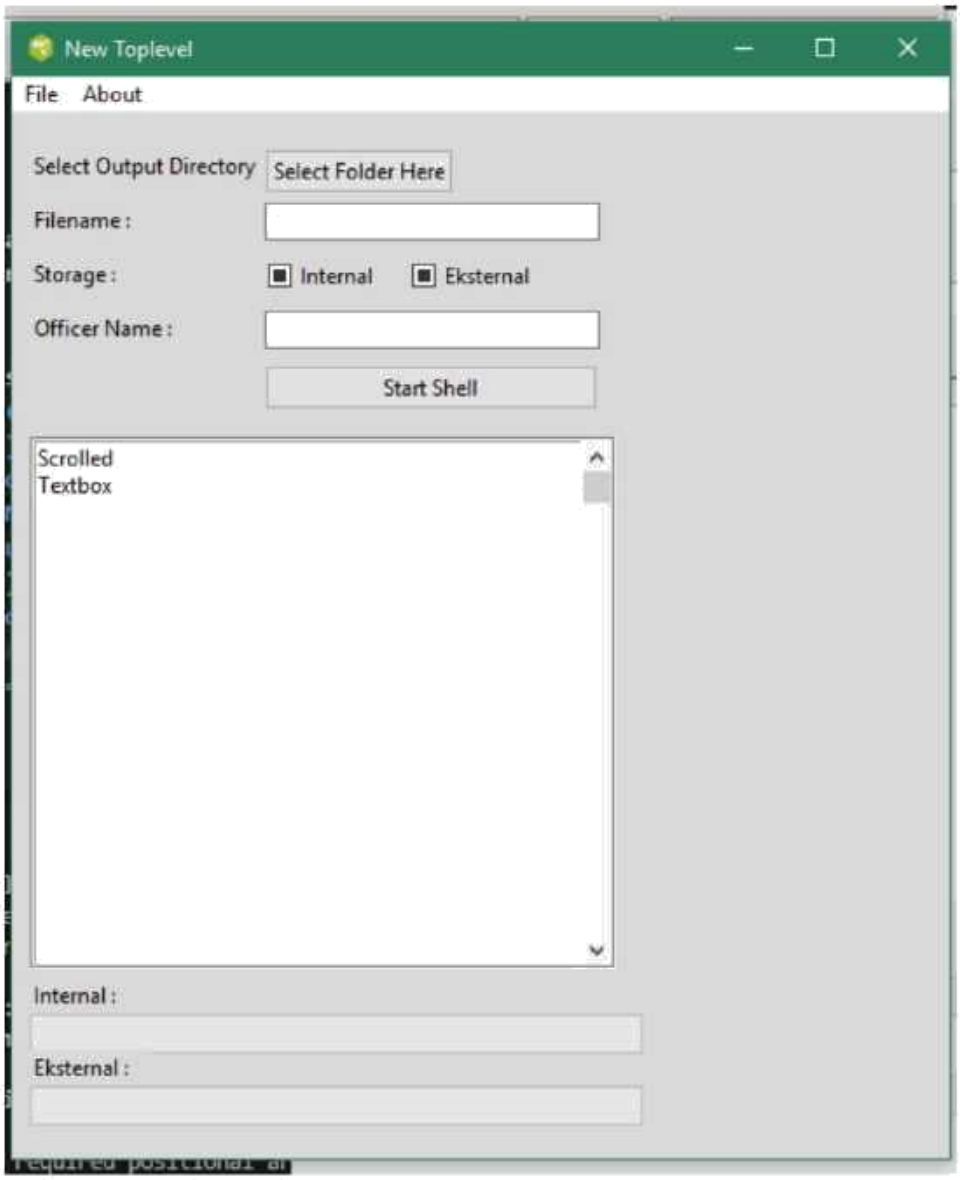
3.2.3 Desain Antarmuka

Desain antarmuka atau yang sering disebut *design interface* merupakan salah satu tahapan yang sangat penting dalam pengembangan sebuah sistem untuk memberikan gambaran-gambaran kepada pengguna mengenai desain antarmuka pada aplikasi Forensics Imaging pada ponsel Android dengan memanfaatkan root yang akan dibuat. Dalam membuat desain antarmuka ini penulis memanfaatkan tools PAGE yang digunakan untuk membuat rancangan desain antarmuka aplikasi. pada aplikasi Forensics Imaging pada ponsel Android yang memanfaatkan root penulis membuat 2 tampilan antarmuka. Untuk melihat rancangan bisa dilihat pada Gambar 3.5.



Gambar 3.4 Rancangan antarmuka Tampilan 1

Pada Gambar 3.4 merupakan desain antarmuka tampilan 1 saat aplikasi pertama kali dijalankan. Pada tampilan ini pengguna diminta untuk mencari perangkat android yang terhubung dengan mengklik tombol Scan, lalu aplikasi akan menampilkan sebuah *serial number* dan status perangkat android yang terhubung dengan laptop yang akan ditampilkan di *Sroclled Listbox,* lalu ada tombol *install root* yang berfungsi untuk menginstall aplikasi yang dibutuhkan pada prosess *rooting* ponsel android, sedangkan tombol *Busybox* yang berfungsi untuk memasang aplikasi *Busybox* pada ponsel android yang digunakan pada proses akuisisi ponsel. Setelah semua proses selesai maka pengguna dapat menekan tombol *Next* yang berguna untuk menampilan Tampilan 2. Untuk desain antarmuka Tampilan 2 bisa dilihat pada Gambar 3.5



Gambar 3.5 Rancangan antarmuka Tampilan 2

Pada Gambar 3.5 merupakan rancangan antarmuka Tampilan 2, pada tampilan 2 ini pengguna diminta untuk mengisikan *form-form* yang dibutuhkan untuk proses akuisisi. *Form* tersebut antara lain *Filename* untuk penamaan *File Image*, *Checkbox* untuk memilih media penyimpanan mana yang akan diakuisisi, *Select Output Directory* untuk peletakan *File Image* yang akan disimpan, sedangkan *form Officer Name* yang berfungsi untuk menyimpan nama pengguna yang melakukan akuisisi ponsel android. Setelah pengguna mengisi semua *form* dengan lengkap maka pengguna diminta menekan tombol *Start Shell* yang berfungsi untuk menjalankan proses akuisisi ponsel android.

3.3 Pengujian

Setelah impemlentasi pengembangan aplikasi selesai, tahapan selanjutnya adalah tahap pengujian untuk semua fungsi yang ada didalam aplikasi telah bekerja dengan baik dan benar ataupun belum. Pengujian ini meliputi pengujian fungsionalitas yang terdapat pada aplikasi, pengujian integritas data yang dihasilkan aplikasi, pengujian performa aplikasi saat menjalankan proses akuisisi dengan skenario mengakuisisi media penyimpanan secara bersamaan dan mengakusisi salah satu media penyimpanan pada 2 perangkat ponsel android baik itu mengakuisisi media penyimpanan internal maupun eksternal, selanjutnya pengujian kuisioner kepada pihak-pihak yang paham ataupun mengerti dalam bidang forensika digital dengan memberikan uraian-uraian pertanyaan yang dijadikan sampel penelitian (responden), kemudian untuk perhitungan skor kuesioner penulis menggunakan perhitungan dengan skala likert, lalu pengujian aplikasi oleh pakar yang akan diuji oleh seseorang yang sudah ahli dalam bidang forensika digital sedangkan pengujian integritas data adalah mencocokan nilai hash MD5 yang dihasilkan aplikasi dengan aplikasi pencocokan MD5.

3.3.1 Pengujian Integritas Data yang Dihasilkan

Dalam pengujian integritas data yang dihasilkan aplikasi adalah pengujian yang dilakukan oleh penulis untuk membandingkan *hash md5* dari file *imaging* yang dihasilkan dengan nilai *hash md5* yang dihasilkan aplikasi pencocok *md6*. Aplikasi tersebut adalah WinMd5 yang berfungsi untuk mencocokan nilai *hash md5* dari *file imaging* yang dihasilkan oleh aplikasi *forensic imaging* pada ponsel android dengan metode *root*.

Dengan dilakukannya pengujian ini bisa memastikan bahwa *file imaging* yang dihasilkan aplikasi yang telah dibuat oleh penulis telah terjamin integritas datanya dan dapat dijadikan barang bukti yang sah ketika dibawa ke pengadilan.

3.3.2 Pengujian Kuesioner Skala Likert

Tahapan berikutnya adalah tahapan pengujian dengan menggunakan kuesioner kepada pihak-pihak yang paham atau pun mengerti dalam bidang forensika digital dengan memberikan beberapa uraian pertanyaan mengenai aplikasi yang dijadikan sampel penelitian (*responden)*, kemudian untuk menghitung kuesioner penulis menggunakan metode skala likert.

Skala likert adalah metode pengukuran yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau kelompok orang tertentu tentang fenomena sosial (Sugiono, 2012). Perhitungan dengan dengan metode skala likert merupakan skala yang paling banyak

dipergunakan dalam riset-riset berupa survey penelitian. Survei tersebut biasanya berupa kuesioner menggunakan metode skala likert yang diwajibkan membuat pertanyaan ataupun pernyataan yang jelas dan tidak mengandung ambigu. Selanjutnya penulis mengajak responden-responden untuk mengungkapkan bentuk persetujuan mereka terhadap pernyataan ataupun pertanyaan yang telah dibuat melalui Google form dan disebarluaskan ke responden. Untuk mendapatkan skor hasil dari kuesioner yang telah diisi oleh responden dengan menggunakan rumus (3.3), setelah itu menghitung rata-rata skor yang didapat, untuk mendapatkan skor rata-rata dari uraian kuesioner dengan menggunakan rumus (3.4).

$$TotalSkor = Jumlah\ Responden \times PilihanNilaiLinkert \quad (3.1)$$

$$Rata - rata = \frac{TotalSkor}{TotalResponden} \quad (3.2)$$

Skala likert mempunyai 5 butir nilai skala jawaban untuk mengungkapkan beberapa persetujuan dari responden terhadap setiap pernyataan ataupun pertanyaan yang ada di dalam kuesioner. Untuk mendapatkan perhitungan nilai skala likert, penulis menggunakan 5 format nilai skala jawaban pada setiap pertanyaan maupun pernyataan dalam uraian kuesioner yang telah disebarluaskan kepada responden, nilai skala jawaban dapat dilihat di Tabel 3.5.

Tabel 3.5 Skala jawaban

Skala jawaban	**Nilai**
Sangat tidak setuju	1
Kurang setuju	2
Cukup setuju	3
Setuju	4
Sangat setuju	5

Setelah menentukan nilai-nilai yang akan dipergunakan untuk kuesioner dengan metode skala likert, selanjutnya penulis melakukan perhitungan dengan rumus Rentang Skala (RS) yaitu nilai yang paling tinggi dikurangi dengan nilai yang paling rendah kemudian dibagi dengan jumlah kategori jawaban. Untuk melihat rumus Rentang Skala (RS), dapat dilihat pada rumus (3.6), dengan menggunakan rumus tersebut maka penulis bisa mendapatkan hasil

rentang skala 0.8. hasil tersebut didapatkan dengan menggunakan rumus rentang skala dengan menggunakan perhitungan skor paling tinggi (5) lalu dikurangi skor paling rendah (1) dan dibagi jumlah kategori jawaban (5) adalah (5-1)/5 = 0.8. untuk melihat hasil yang didapatkan dengan menggunakan rumus Rentang Skala (RS) dapat dilihat pada Tabel 3.7.

$$RentangSkala = \frac{SkorPalingtinggi - skorpalingrendah}{jumlahkategorijawaban} \quad (3.3)$$

Tabel 3.7 Rentang Skala linkert

Nilai	Rentang	Keterangan
1	1,00 – 1,8	Sangat tidak bermanfaat
2	1,81 – 2,60	Tidak bermanfaat
3	2,61 – 3,40	Netral
4	3,41 – 4,20	Bermanfaat
5	4,21 – 5,00	Sangat bermanfaat

Dengan menggunakan skala jawaban pada Tabel 3.5. Dan melalui perhitungan dengan rumus Rentang Skala (RS) likert pada Tabel. Maka penulis membuat beberapa pernyataan maupun pertanyaan yang akan disebarluaskan kepada responden dalam bidang forensika digital untuk mengungkapkan persetujuan mereka terhadap hasil penelitian yang penulis lakukan melalui *Google form*. Untuk uraian kuesioner dapat dilihat pada Tabel 3.8.

Tabel 3.8 Uraian kuesioner

NO	URAIAN	SKOR				
A.	**Manfaat**	**1**	**2**	**3**	**4**	**5**
1.	Aplikasi *Forensics Imaging* pada ponsel android dengan memanfaatkan *root* ini memiliki manfaat bagi saya yang bergelut pada bidang forensika digital ?					

2.	Aplikasi *Forensics Imaging* pada ponsel android dengan memanfaatkan *root* ini mudah digunakan ?					
NO	**URAIAN**	**SKOR**				
3.	Menggunakan aplikasi ini pada saat ingin melakukan Cloning data pada penyimpanan Android ?					
4.	Saya berharap aplikasi ini digunakan oleh orang banyak yang bergelut pada bidang forensika digital ?					
5.	Keluaran aplikasi ini sudah sesuai dengan yang saya harapkan ?					
B.	**Tampilan**	**1**	**2**	**3**	**4**	**5**
1.	Tampilan antarmuka aplikasi ini menarik bagi saya?					
2.	Tampilan antarmuka aplikasi ini mudah dikenali ?					
3.	Menu yang ditampilkan pada aplikasi ini mudah dipahami?					
4.	Peletakan semua fitur dan tombol aplikasi sudah sesuai dengan yang diharapkan?					
C.	**Fungsionalitas**	**1**	**2**	**3**	**4**	**5**
1.	Fitur untuk penginstallan root sudah berjalan dengan semestinya?					
2.	Fitur untuk penginstalan busybox sudah berjalan dengan semestinya?					
3.	Fitur untuk scanning perangkat berjalan dengan baik?					

4.	Fitur untuk mengkoneksikan perangkat sudah berjalan dengan baik?					
	URAIAN	**SKOR**				
5.	Fitur untuk proses akusisi penyimpanan pada ponsel android sudah berjalan dengan baik?					
6.	Semua tombol dapat bekerja dengan normal ?					

BAB IV
IMPLEMENTASI DAN HASIL PENELITIAN

4.1. Implementasi

4.1.1 Root

Pada tahap ini menjelaskan tentang fungsi dari *root* dalam ponsel android untuk pengembangan aplikasi *Forensics imaging* pada ponsel android dengan memanfaatkan *root* yang digunakan oleh penulis adalah *Kingoroot* yang mempunyai banyak manfaat, untuk melihat tampilan *Kingoroot* dalam ponsel android dapat dilihat pada Gambar 4.6

Gambar 4.1 Tampilan aplikasi kingoroot pada ponsel

Pada Gambar 4.1 merupakan tampilan dalam *Kingoroot* yang terdapat pada ponsel android yang memiliki banyak fungsi yaitu diantaranya untuk menonaktifkan aplikasi yang tidak perlu atau dikenal dengan *bloatware* yang terinstall dari pabrik, menginstall aplikasi yang membutuhkan hak *superuser*, fungsi utama untuk mengembalikan performa smartphone tanpa

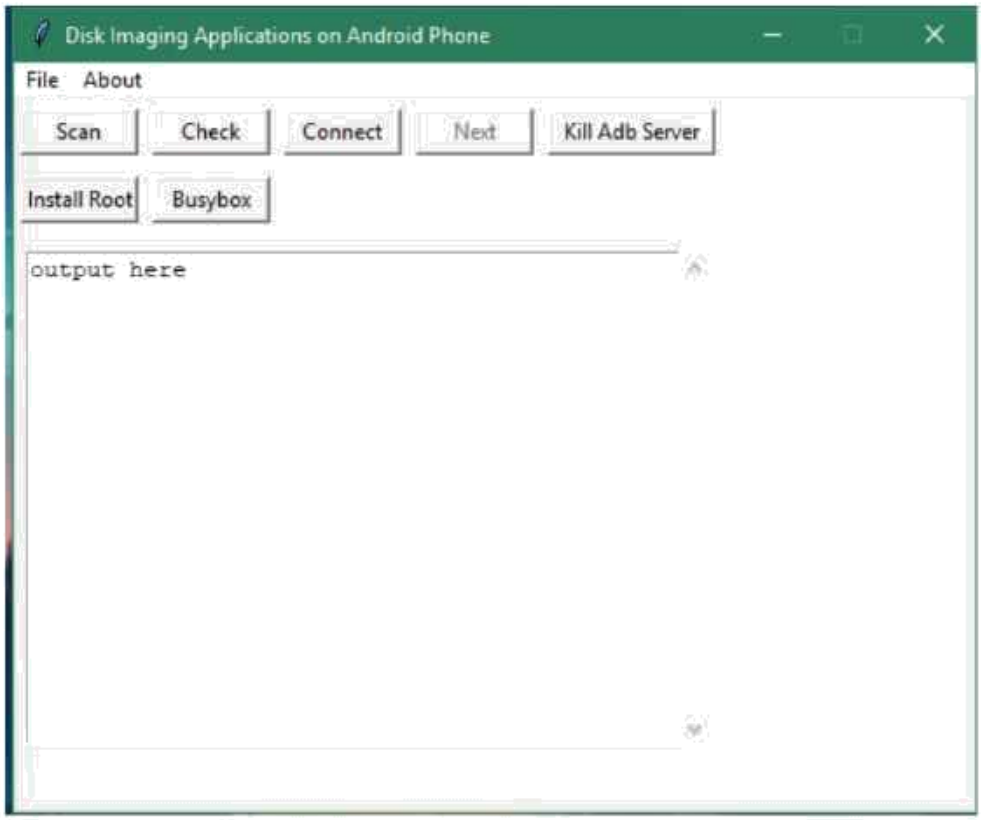
dibebani program walau perubahanya belum tentu terlihat dan fungsi utama yang dibutuhkan penulis dalam pembuatan aplikasi ini adalah memberikan ponsel android dengan hak *superuser* untuk keperluan proses akuisisi media penyimpanan pada ponsel android.

4.1.2 Implementasi Antarmuka

Tahapan ini menjelaskan tentang tampilan antarmuka yang terdapat pada aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *root*. pada aplikasi yang penulis buat ini terdapat 2 tampilan antarmuka yaitu tampilan 1 dan tampilan 2 aplikasi yang mempunyai fungsi untuk melakukan proses akuisisi media penyimpanan pada ponsel android baik media penyimpanan internal maupun eksternal. Untuk penjelasan kedua antarmuka adalah sebagai berikut :

a. Tampilan antarmuka 1 aplikasi

Pada tampilan 1 aplikasi mempunyai fungsi untuk menjalankan proses pencarian perangkat ponsel android (*scanning*) dan mengkoneksikan perangkat pada suatu *port* untuk keperluan melakukan proses akuisisi yang akan dijalankan pada tampilan 2 aplikasi untuk tampilan 1 dapat dilihat pada Gambar 4.2



Gambar 4.2 Antarmuka tampilan 1 aplikasi

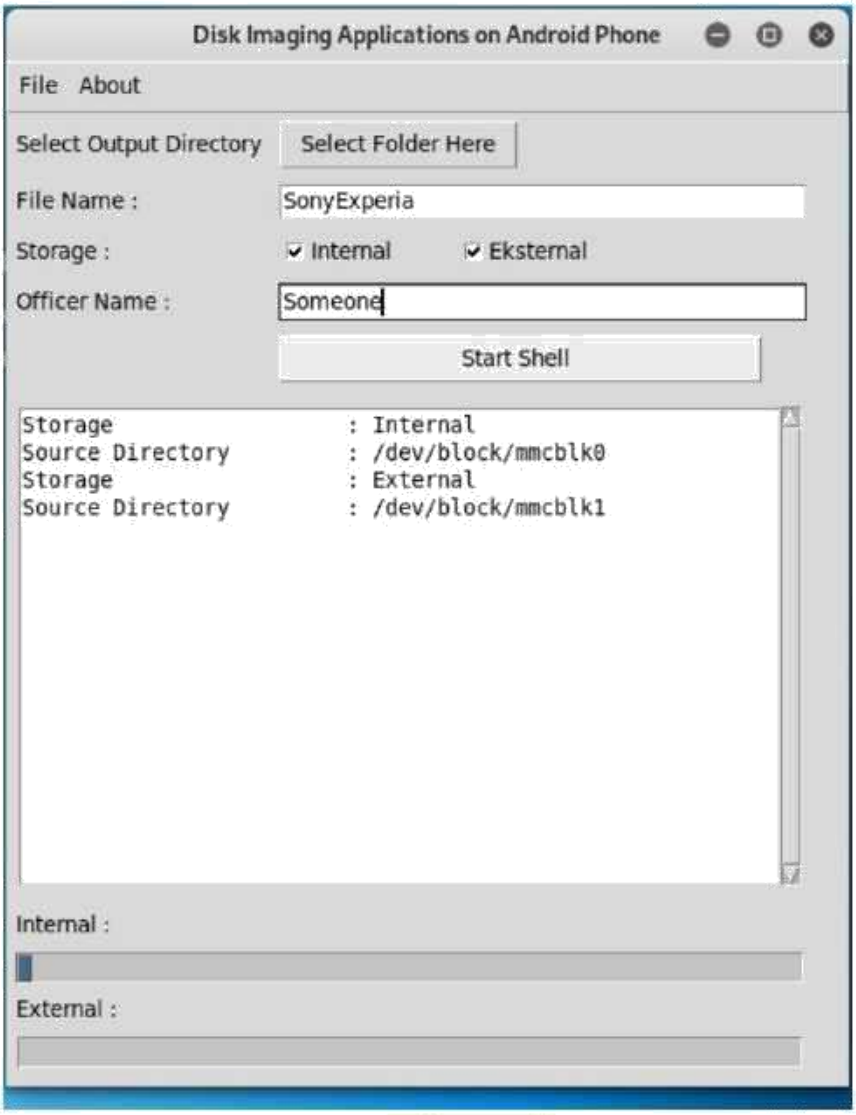
Pada Gambar 4.2 aplikasi mempunyai fungsi untuk menginstall *Kingoroot* dan *BusyBox* pada ponsel android untuk keperluan proses akuisisi, untuk melakukan pencarian ponsel android (*scanning*) dan mengkoneksikan ponsel android dengan aplikasi untuk kepentingan proses akuisisi pada tampilan 2 aplikasi. untuk penjelasan dan fungsi masing-masing dapat dijelaskan sebagai berikut :

1. Menu "File", untuk menu file memiliki beberapa sub menu yaitu "scan" dan "exit". Sub menu scan mempunyai fungsi untuk mencari ponsel android yang terhubung dengan laptop/host yang akan ditampilkan pada *textbox scrolled* sedangkan exit berfungsi untuk keluar dari aplikasi.
2. Menu "About" berfungsi untuk menampilkan suatu dialog box yang berisikan informasi tentang cara-cara pembuatan aplikasi.
3. Tombol "Scan" yang berfungsi untuk mencari ponsel android yang terhubung dengan laptop/host dan menghasilkan keluaran berupa *serial number, state* dan *type* kedalam *scrolled text.*
4. Tombol "Check" mempunyai fungsi untuk menampilkan informasi perangkat yang terhubung pada laptop/host dari *serial number* maupun *port* yang digunakan untuk menghubungkan ponsel android dengan laptop/host yang akan ditampilkan pada *scrolled text.*
5. Tombol "Install *Root*" mempunyai fungsi untuk menginstall aplikasi *Kingoroot* yang dibutuhkan untuk memberikan hak *superuser* pada ponsel android yang nantinya akan dibutuhkan pada proses akuisisi.
6. Tombol "Install *BuxyBox*" mempunyai fungsi untuk menginstall aplikasi *busybox* yang dibutuhkan pada proses akuisisi media penyimpanan ponsel android.
7. Tombol "*Kill Adb-sever*" mempunyai fungsi untuk memberhentikan fungsi *adb* yang dibutuhkan ketika proses aplikasi tidak berjalan dengan baik maupun ketika aplikasi telah selesai melakukan proses akuisisi media penyimpanan pada ponsel android.
8. Tombol "Connect" yang mempunyai fungsi untuk menghubungkan ponsel android pada suatu *port*.
9. "*Scrolled Text*" yang berfungsi untuk menampilkan informasi-informasi tentang perangkat android yang terkoneksi.

10. Tombol "Next", pada tombol "Next" akan berfungsi ataupun aktif ketika semua proses-proses yang dibutuhkan untuk mengkoneksikan perangkat android telah selesai dijalankan. Sehingga tombol "next" akan berfungsi untuk melanjutkan ke antarmuka tampilan 2 aplikasi.

b. Antarmuka tampilan 2 aplikasi

Pada antarmuka tampilan 2 aplikasi ini mempunyai fungsi untuk menjalankan proses akuisisi media penyimpanan pada ponsel android baik itu media penyimpanan internal maupun eksternal dan menjalankan proses *hashing md5* dari *file image* yang dihasilkan aplikasi dengan file sumber penyimpanan pada ponsel android untuk melihat tampilan 2 dapat dilihat pada Gambar 4.3



Gambar 4.3 Antarmuka tampilan 2 aplikasi

Pada Gambar 4.3 merupakan antarmuka tampilan 2 pada aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *root.* terdapat beberapa kolom pada aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *root* diantaranya adalah "Select output directory" untuk menentukan peletakan *file image* yang akan dihasilkan aplikasi, lalu ada kolom "file name" untuk menentukan nama dari *file image* yang dikeluarkan aplikasi, selanjutnya ada kolom storage untuk menentukan media penyimpanan mana yang akan diakuisisi dan dapat memilih kedua media penyimpanan tersebut dan kolom "officer name" yang bertujuan untuk menentukan nama pengguna aplikasi saat melakukan proses akuisisi. Setelah itu ada *scrolled text* yang berfungsi untuk menampilkan informasi-informasi yang telah dilakukan di dalam aplikasi.

4.1.3 Implementasi Kode Program Proses Scanning Ponsel Android

Tahapan awal yang harus dilakukan sebelum memulai proses akuisisi adalah melakukan proses *scanning* perangkat ponsel android yang terhubung pada laptop/host yang akan dibaca oleh aplikasi. proses ini menggunakan modul *subprocess* dalam bahasa pemrograman *python* melalui bantuan *tool* ADB Daemon. Kode program untuk proses *scanning* dapat dilihat pada Gambar 4.4

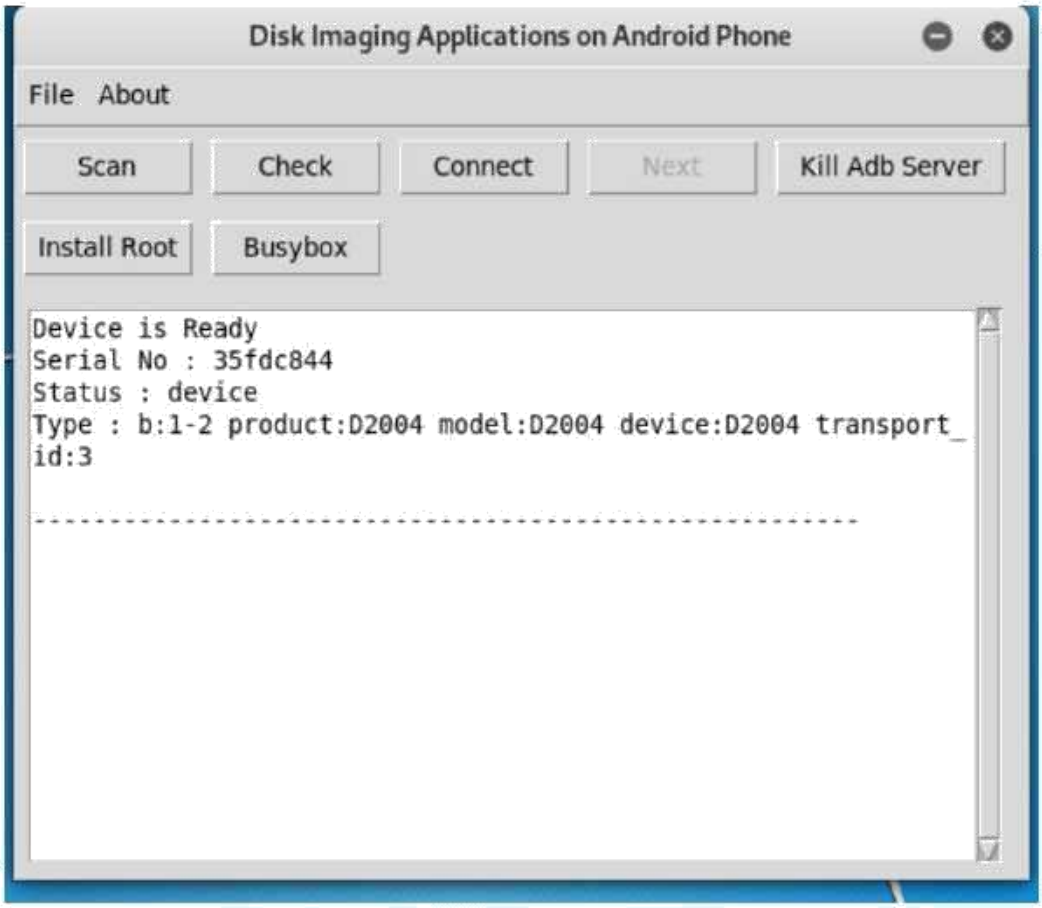
```
497  def clk_scn(self):
498  subprocess.getoutput('adb start-server')
499  serial = subprocess.getoutput('adb devices -l')
500  forSerial = subprocess.getoutput('adb get-serialno')
501  forStatus = subprocess.getoutput('adb get-state')
502  serialNo = forSerial
503  status = forStatus
504  model = serial[86:95]
505  type = serial[57:120]
506
507  if forSerial == 'unknown':
508  self.outText.delete("1.0", END)
509  self.OUTPUT(END, "No device found"+"\n")
510  self.OUTPUT(END,"---------------------------------------------------------------" +"\n"+"\n")
511  mbox.showerror("Caution", "No device found")
512
513  else:
514  self.outText.delete("1.0", END)
```

```
515  self.OUTPUT(END, "Device is Ready"+"\n")
516  self.OUTPUT(END, "Serial No : "+serialNo+"\n")
517  self.OUTPUT(END, "Status : " + status+"\n")
518  self.OUTPUT(END, "Type : " + str(type)+"\n")
519  self.OUTPUT(END,"....." +"\n"+"\n")
```

Gambar 4.4 Kode program proses scanning

Pada Gambar 4.4 merupakan kode program untuk proses *scanning* ponsel android yang akan dibaca oleh aplikasi yang terjadi pada tampilan 1 aplikasi. pada baris 471 merupakan fungsi *scanning* yang akan dijalankan aplikasi setelah pengguna menekan tombol "scan". Selanjutnya pada kode program baris 472 merupakan fungsi untuk membuka *server* awal pencarian perangkat. Pada kode program baris 473 sampai dengan 479 merupakan suatu variabel yang akan menyimpan *string* yang dihasilkan dari perintah yang dilakukan kode program baris 473. Pada kode program baris baris 467 terdapat variabel "serialNo" yang menyimpan nilai *string* mengenai *serial number* perangkat ponsel android. Selanjutnya terdapat variabel pada kode program baris 477 yang berfungsi untuk menyimpan nilai *string* mengenai status perangkat ponsel android. Lalu pada kode program baris 478 dan 479 terdapat variabel model dan type yang berfungsi untuk menyimpan nilai *string* mengenai tipe dan model perangkat ponsel android yang sudah terhubung dengan laptop/host.

Selanjutnya pada kode program baris 481 hingga 493 menjelaskan kondisi perangkat android, pada kode program baris 481 jika menghasilkan hasil "unknown" dari suatu fungsi maka aplikasi akan menghasilkan suatu keluaran pada *scrolled text* berupa text bahwa tidak ada perangkat android yang terhubung atau ditemukan, dan jika memberikan hasil selain kata "unknown" maka aplikasi akan memberikan keluaran informasi bahwa ada perangkat yang terhubung atau ditemukan dan menghasilkan keluaran informasi yang ditampilkan di *scrolled text*. Untuk melihat hasil proses *scanning* perangkat yang ditemukan dapat dilihat pada Gambar 4.5.



Gambar 4.5 Hasil proses scanning

4.1.4 Implementasi Kode Program Proses Mengkoneksikan Ponsel Android

Tahapan selanjutnya setelah tahapan *scanning* adalah tahapan mengkoneksikan perangkat android yang akan dilakukan aplikasi. proses ini menggunakan modul *subprocess* dalam bahasa pemrograman *python* melalui *tool* ADB Daemon. Untuk kode program proses mengkoneksikan perangkat android dapat dilihat pada Gambar 4.6.

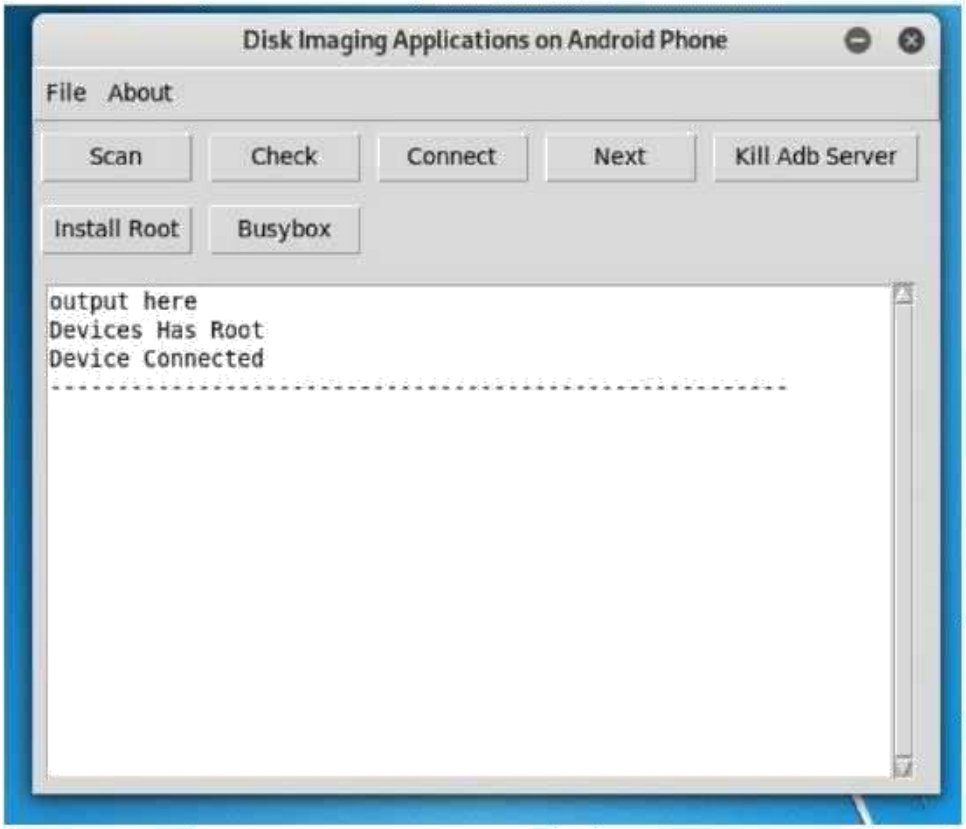
```
471  def clk_connect(self):
472  var2 = subprocess.getoutput("adb get-state")
473  var4 = subprocess.getoutput("adb shell su -c ls /data")
474  var5 = subprocess.getoutput("adb shell pm list packages -u stericson.busybox ")
475  output = "/system/bin/sh: su: not found"
476  output1 = "error: no devices/emulators found"
477  output2 = "package:stericson.busybox"
478  if var2 == 'device':
479  if var4 == output:
480  mbox.showerror("Validation", "Devices has not installed Root, Please Install")
```

```
481  elif var5 == "":
482  mbox.showerror("Validation","Devices has not installed BusyBox, Please Install")
483  elif var5 == output2:
484  if var4 == var4:
485  var = subprocess.getoutput("adb forward tcp:8888 tcp:8888")
486  text = "{var}".format(var=var)
487  if text == "":
488  self.OUTPUT(END, "Devices Has Root"+"\n")
489  self.OUTPUT(END, "Device Connected" + "\n")
490  self.OUTPUT(END,"----------------------------------------------------" +"\n"+"\n")
491
492  self.nextButton(NORMAL)
493  elif var2 == output1:
494  mbox.showerror("Validation","No Device Detected, Or Device Not Conected")
495
```

Gambar 4.6 Kode program mengkoneksikan perangkat ponsel dengan aplikasi

Pada Gambar 4.6 merupakan kode program untuk mengkoneksikan perangkat android melalui sebuah port yang terjadi pada tampilan 1 aplikasi. pada kode program 446 merupakan fungsi untuk menjalankan proses mengkoneksikan perangkat android setelah pengguna menekan tombol "connect" pada aplikasi. pada kode program baris 446 adalah variabel untuk menyimpan nilai *string* yang dihasilkan suatu perintah untuk mengetahui keadaan sebuah ponsel android. Selanjutnya pada kode program baris 447 merupakan fungsi dan menyimpan sebuah nilai *string* untuk menjalankan perintah untuk mengecek keadaan sebuah ponsel sudah diberikan hak akses root atau belum. Sedangkan pada kode program baris 448 merupakan sebuah fungsi dan variabel nilai *string* yang berfungsi untuk mengecek sebuah aplikasi sudah terpasang atau belum pada ponsel android. Pada kode program baris 452 hingga baris 469 menjelaskan keadaan dan mengkoneksikan perangkat android. Pada kode program baris 452 menjelaskan jika status ponsel android mengeluarkan nilai "device" maka ada perangkat ponsel android yang sudah terhubung pada laptop/host dan jika keluaranya berupa nilai "error: no devices/emulators found" maka tidak ada perangkat ponsel android yang terhubung dengan laptop/host. Pada kode program baris 452 menjelaskan jika nilai yang dikeluarkan ponsel android berupa suatu nilai *string* maka aplikasi akan mengeluarkan informasi berupa dialog box bahwa ponsel android tersebut belum diberikan hak *superuser*. Pada kode program baris

454 menjelaskan jika ponsel android tidak memberikan suatu nilai string setelah menjalankan fungsi untuk pengecekan sebuah *packages name* maka ponsel android tersebut belum terinstall aplikasi *busybox*, setelah itu pada kode program baris 456 menjelaskan jika aplikasi menjalankan sebuah fungsi untuk pengecekan *packages name* dan ponsel android memberikan sebuah nilai *string* maka aplikasi busybox sudah terinstall dan dapat dilanjut untuk pengecekan hak superuser pada kode program baris 457. Lalu jika sudah selesai semua pengecekan fungsi-fungsi dan aplikasi penunjang akuisisi sebuah media penyimpanan pada ponsel android maka program akan mengkoneksikan perangkat android melalui sebuah *port*. Untuk melihat hasil kode program mengkoneksikan perangkat android yang ditemukan dapat dilihat pada Gambar 4.7.



Gambar 4.7 Hasil mengkoneksikan ponsel dengan aplikasi

4.1.5 Implementasi Kode Program Proses Akuisisi

Setelah semua tahapan pada antarmuka tampilan 1 aplikasi selesai selanjutnya adalah tahapan akuisisi perangkat ponsel android pada antarmuka tampilan 2 aplikasi. proses ini menggunakan modul *pyadb3* dan *subprocess* sedangkan untuk tool yang digunakan adalah tool

netcat dalam bahasa pemrograman *python*. Kode program proses akuisisi dan *hashing md5* dapat dilihat pada Gambar 4.8.

```
98  def startProcess(self, window):
99      self.startTime = datetime.now()
100     self.sizeoutputInt = 0
101     self.sizeoutputExt = 0
102
103     if (self.CheckVar1.get() == 1 and self.CheckVar2.get() == 1):
104         self.outText.delete("1.0", END)
105         self.outText.insert(END, "Storage \t\t\t : Internal" + "\n")
106         self.outText.insert(END, "Source Directory \t\t\t : /dev/block/mmcblk0" + "\n")
107         self.outText.insert(END, "Storage \t\t\t : External" + "\n")
108         self.outText.insert(END,"Source Directory \t\t\t : /dev/block/mmcblk1"+"\n")
109         self.outText.insert(END,"-------------------------------------------------" +"\n"+"\n")
110         self.choosenDir = "mmcblk0"
111         self.sizeoutputInt = self.outputDumpShell(window)
112
113         p = Process(target= lambda: self.startdumpShell(window))
114         q = Process(target= lambda: self.startOuputDumpInt(window))
115
116         if self.return_code !=0:
117             p.start()
118             time.sleep(6)
119             q.start()
120         else:
121             p.terminate()
122             time.sleep(2)
123             q.terminate()
124
125         time.sleep(3)
126
127
128         alldirect = "{}/_Int{}.dd".format(self.outputfolder, self.txtFileName.get())
129         currentOutputSize = os.stat(alldirect).st_size / (1024 * 1024 * 1024)
130         self.progressInt['maximum'] = self.sizeoutputInt
131
132         state = True
133         while state:
134
```

```
135             state = self.progressIntBar(window, currentOutputSize, self.sizeoutputInt)
136             root.update_idletasks()
137             currentOutputSize = os.stat(alldirect).st_size / (1024 * 1024 * 1024)
138
139         self.hashmd5(window)
140         self.createLogInt()
141
142
143         p = Process(target= lambda: self.startdumpShell(window))
144         q = Process(target= lambda: self.startOuputDumpExt(window))
145         self.choosenDir = "mmcblk1"
146         self.sizeoutputExt = self.outputDumpShell(window)
147
148         if self.return_code !=0:
149             p.start()
150             time.sleep(6)
151             q.start()
152         else:
153             p.terminate()
154             time.sleep(2)
155             q.terminate()
156
157         time.sleep(3)
158
159         alldirect = "{}/_Ext{}.dd".format(self.outputfolder, self.txtFileName.get())
160         currentOutputSize = os.stat(alldirect).st_size / (1024 * 1024 * 1024)
161         self.progressExt['maximum'] = self.sizeoutputExt
162
163         state = True
164         while state:
165             state = self.progressExtBar(window, currentOutputSize, self.sizeoutputExt)
166             root.update_idletasks()
167             currentOutputSize = os.stat(alldirect).st_size / (1024 * 1024 * 1024)
168
169         self.hashmd5(window)
170         self.createLogExt()
171         self.FinishedTextInt(window)
172         self.FinishedTextExt(window)
173         mbox.showinfo("Finished", "Aqusition Finished")
174         self.endTime = datetime.now()
175
176
```

```
177
178         elif self.CheckVar1.get() == 1:
179             self.outText.delete("1.0", END)
180             self.outText.insert(END, "Storage \t\t\t : Internal" + "\n")
181             self.outText.insert(END,"Source Directory \t\t\t : /dev/block/mmcblk0"+"\n")
182             self.outText.insert(END,"----------------------------------------------------" +"\n"+"\n")
183             p = Process(target= lambda: self.startdumpShell(window))
184             q = Process(target= lambda: self.startOuputDumpInt(window))
185             self.choosenDir = "mmcblk0"
186             self.sizeoutputInt = self.outputDumpShell(window)
187
188             if self.return_code !=0:
189                 p.start()
190                 time.sleep(6)
191                 q.start()
192             else:
193                 p.terminate()
194                 time.sleep(2)
195                 q.terminate()
196
197             time.sleep(3)
198
199             alldirect = {}/_Int{}.dd".format(self.outputfolder, self.txtFileName.get())
200             currentOutputSize = os.stat(alldirect).st_size / (1024 * 1024 * 1024)
201             self.progressInt['maximum'] = self.sizeoutputInt
202
203             state = True
204             while state:
205                 state = self.progressIntBar(window, currentOutputSize, self.sizeoutputInt)
206                 root.update_idletasks()
207                 currentOutputSize = os.stat(alldirect).st_size / (1024 * 1024 * 1024)
208
209             self.hashmd5(window)
210             self.createLogInt()
211             self.FinishedTextInt(window)
212
213             mbox.showinfo("Finished", "Aqusition Finished")
214
215
216
217         elif self.CheckVar2.get() == 1:
218             self.outText.delete("1.0", END)
             self.outText.insert(END, "Storage \t\t\t : External" + "\n")
```

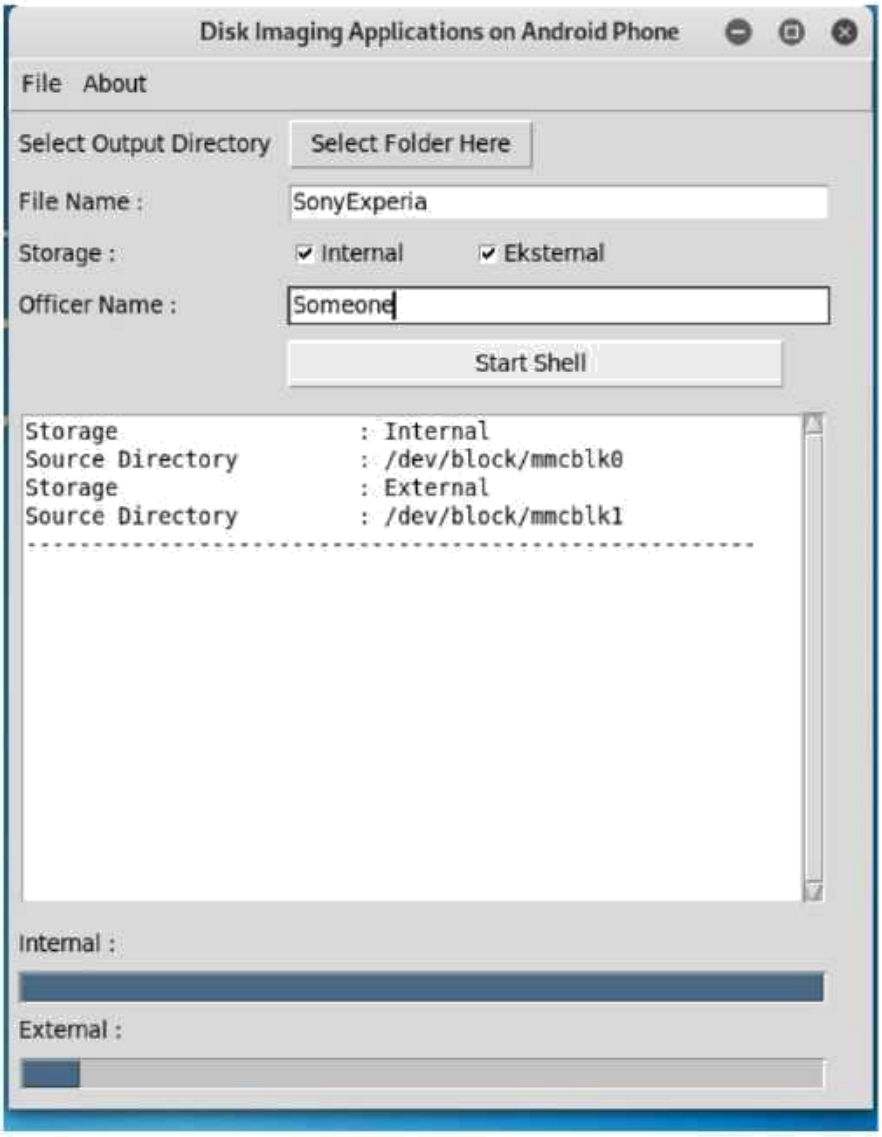
```
219         self.outText.insert(END,"Source Directory \t\t\t : /dev/block/mmcblk1"+"\n")
220         self.outText.insert(END,"-------------------------------------------------" +"\n"+"\n")
221         self.choosenDir = "mmcblk1"
222         p = Process(target= lambda: self.startdumpShell(window))
223         q = Process(target= lambda: self.startOuputDumpExt(window))
224         self.outputDumpShell(window)
225         self.sizeoutputExt = self.outputDumpShell(window)
226
227         if self.return_code !=0:
228             p.start()
229             time.sleep(6)
230             q.start()
231         else:
232             p.terminate()
233             time.sleep(2)
234             q.terminate()
235
236         time.sleep(3)
237
238         alldirect = "{}/_Ext{}.dd".format(self.outputfolder, self.txtFileName.get())
239         currentOutputSize = os.stat(alldirect).st_size / (1024 * 1024 * 1024)
240         self.progressExt['maximum'] = self.sizeoutputExt
241
242         state = True
243         while state:
244             state = self.progressExtBar(window, currentOutputSize, self.sizeoutputExt)
245             root.update_idletasks()
246             currentOutputSize = os.stat(alldirect).st_size / (1024 * 1024 * 1024)
257
248         self.hashmd5(window)
249         self.createLogExt()
250         self.FinishedTextExt(window)
251       mbox.showinfo("Finished", "Aqusition Finished")
252

279 def hashmd5(self, window):
271     time.sleep(5)
272     fDirect = self.outputfolder
273     nDirect = "/_Ext" + self.txtFileName.get() +".dd"
274     alldirect = fDirect + nDirect
275     block_size = 2 ** 20
276     md5a = hashlib.md5()
```

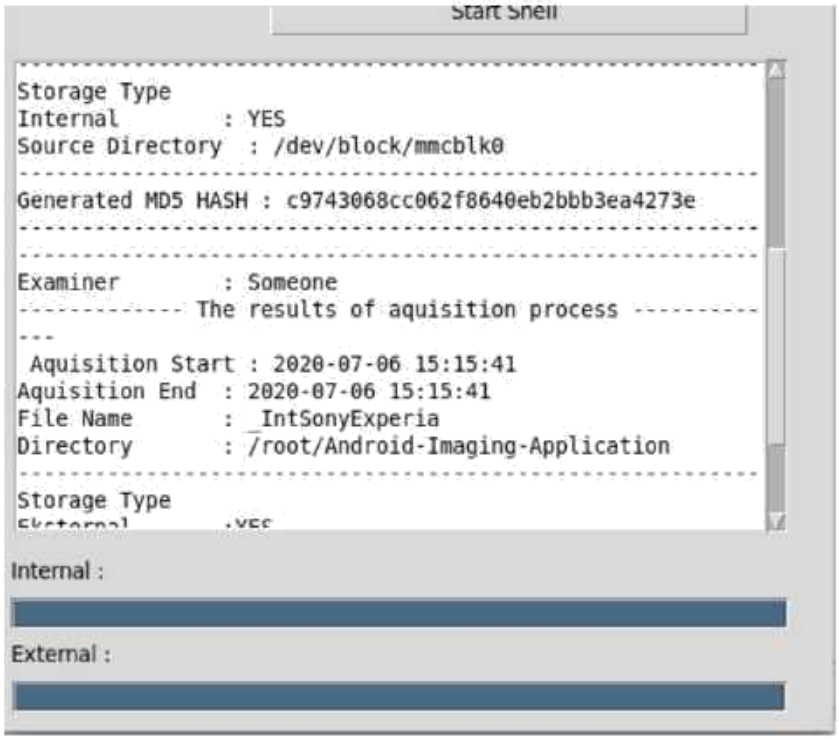
```
277         if( os.path.exists(alldirect) ) :
278             with open(alldirect, 'rb') as fileCloning:
279                 while True:
280                     readCloning = fileCloning.read(block_size)
281                     if not readCloning:
282                         break
283                     md5a.update(readCloning)
284
285         self.hasilCloning = md5a.hexdigest()
286
287         adbHash = pyadb3.ADB()
288         coHash = adbHash.run_shell_cmd("su -c md5sum /dev/block/{}".format(self.choosenDir))
289         splitHash = "{coHash}".format(coHash=coHash)
290         self.hasilHash = splitHash[2:34]
291         self.isHashMatched = ""
292
293         if self.hasilHash == self.hasilCloning:
294             self.isHashMatched = "MD5 Hash Matched"
295         else:
296             self.isHashMatched = "MD5 Hash Not Matched"
297
```

Gambar 4.8 Kode program proses akuisisi ponsel android

Pada kode program baris 98 merupakan fungsi yang dijalankan setelah pengguna telah mengisi semua kolom yang dibutuhkan untuk proses akuisisi dan menekan tombol "start shell". Kode program dari baris 103 hingga 131 menjelaskan jika *check box* internal dan eksternal dalam terceklis maka kode program dari baris 103 hingga kode program baris 138 maka aplikasi akan menjalankan proses akuisisi penyimpanan internal dalam *shell* ponsel android, menjalankan fungsi *hashing md5* dan membuat file log, selanjutnya kode program 140 hingga 171 maka aplikasi akan menjalankan proses akuisisi media penyimpanan eksternal dalam shell ponsel android, menjalankan fungsi *hashing md5* dan membuat *file log*. Kode program dari baris 174 sampai 209 menjelaskan jika *check box* internal saja yang terceklis maka aplikasi akuisisi media penyimpanan internal dalam shell ponsel android, menjalankan fungsi *hashing md5* dan membuat *file log* penyimpanan internal. Lalu selanjutnya pada kode program 212 hingga 249 menjelaskan bahwa jika *check box* penyimpanan eksternal saja yang di ceklis maka aplikasi menjalankan proses akuisisi media penyimpanan eksternal dalam *shell* ponsel android, menjalankan fungsi *hashing md5* dan membuat *file log* media penyimpanan eksternal. Sedangkan kode program baris 251 hingga 278 merupakan fungsi utama untuk proses *hashing md5* dari *file image* dan pada baris 275 hingga 278 adalah proses pencocokan *file image* dengan direktori sumber media penyimpanan ponsel android. Untuk melihat hasil proses akuisisi kedua penyimpanan yang telah selesai dapat dilihat pada Gambar 4.9.



Gambar 4. 9 Proses akuisisi ponsel android



Gambar 4.10 Proses akuisisi ponsel android yang sudah selesai

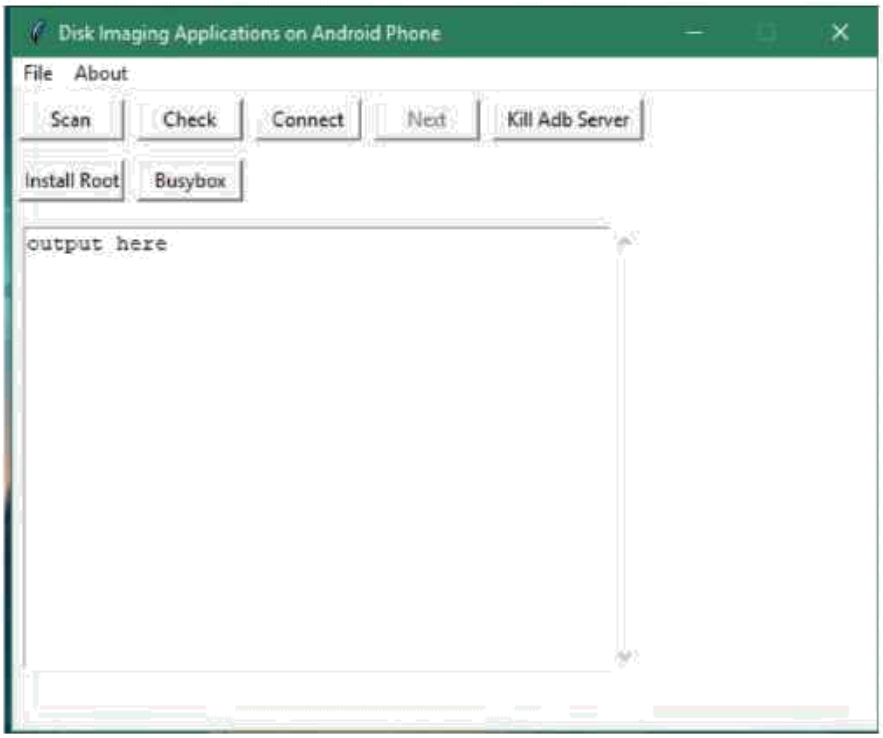
4.2 Hasil

4.2.1 Hasil Pembuatan Aplikasi

Hasil yang didapatkan adalah sebuah aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *Root* yang dapat mengakuisisi penyimpanan pada ponsel android baik internal maupun eksternal dan juga pengguna dapat membantu proses akuisisi dengan menampilkan *progress bar* dalam tampilan aplikasi. Hasil implementasi aplikasi dijelaskan aplikasi ini memiliki 2 tampilan yang dapat dijelaskan sebagai berikut:

a. Hasil tampilan 1 aplikasi

Hasil dari mengimplementasikan salah satunya adalah tampilan 1 aplikasi, tampilan pertama aplikasi adalah tampilan awal pada saat aplikasi pertama dibuka yang sudah dirancang dengan bantuan tool *PAGE*. Untuk melihat hasil tampilan 1 aplikasi dapat dilihat pada Gambar 4.11.

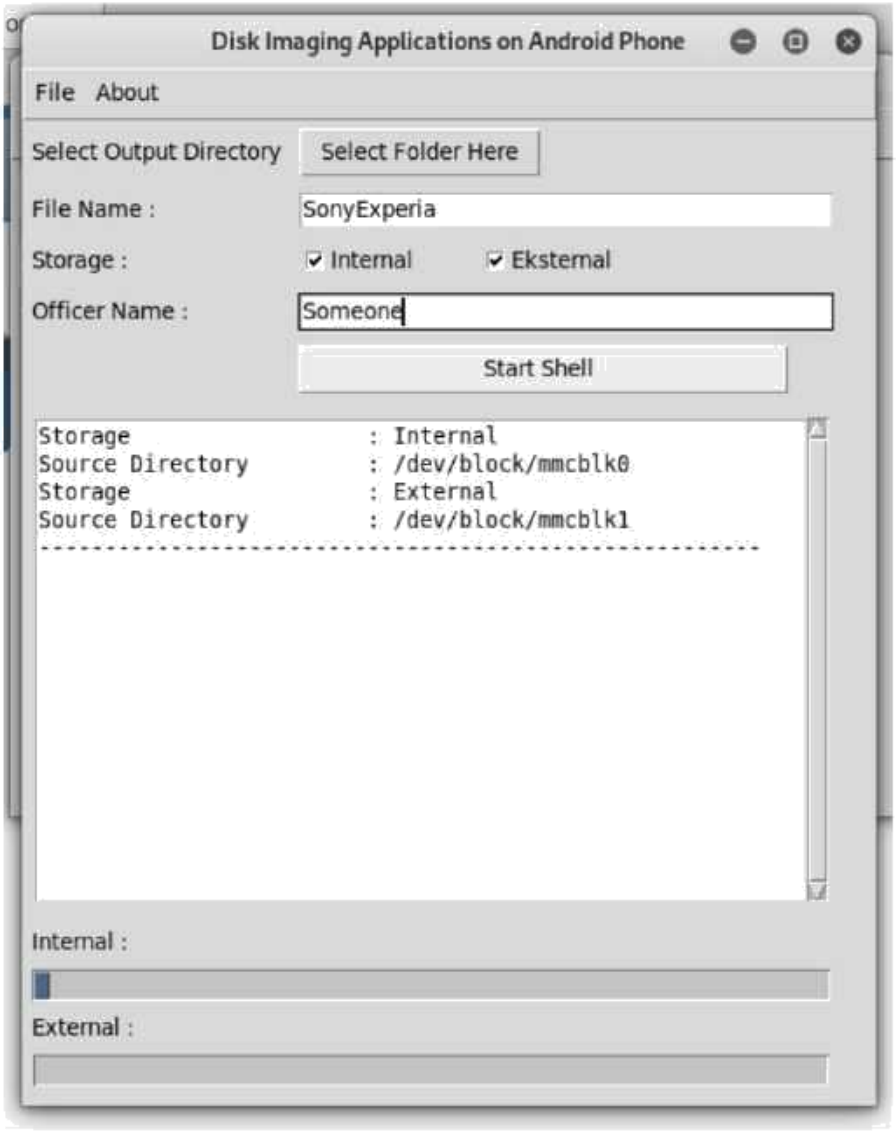


Gambar 4.11 Hasil implementasi tampilan 1

Pada Gambar 4.11 adalah hasil implementasi pada tampilan 1 aplikasi *Forensics Imaging* pada ponsel android dengan metode *root*, tampilan tersebut adalah tampilan awal saat pertama kali aplikasi dibuka. Dalam tampilan 1 aplikasi terdapat beberapa fungsi yang berguna untuk proses akuisisi android, fungsi tersebut diantaranya fungsi untuk menginstall aplikasi yang berguna untuk memberikan hak akses *root* pada ponsel dengan menekan tombol Install *Root*, fungsi kedua yaitu tombol Install *BusyBox* yang berguna untuk menginstall aplikasi *BusyBox* pada ponsel android, aplikasi *BusyBox* adalah salah satu peranan aplikasi yang penting dalam proses akuisisi ponsel android, lalu Fungsi ketiga adalah melakukan pencarian ponsel *(scanning)* perangkat android dengan menekan tombol Scan, lalu yang keempat yaitu melakukan pengecekan koneksi antara laptop/host dan ponsel android dengan menekan tombol *check*. Fungsi kelima yaitu mematikan koneksi *ADB Server* jika dalam proses akuisisi terdapat masalah dengan menekan tombol *Kill ADB Server*. Dan yang terakhir yaitu fungsi untuk menghubungkan ponsel android dengan laptop/host yang sudah terbaca pada aplikasi untuk melakukan tahap selanjutnya yaitu proses akuisisi penyimpanan android pada tampilan 2.

b. Hasil tampilan 2 aplikasi

Hasil implementasi setelah tampilan 1 adalah tampilan 2 aplikasi yaitu tampilan akhir dari aplikasi yang telah dirancang dan dibuat dengan bantuan tool PAGE. Untuk melihat hasil tampilan 2 aplikasi bisa dilihat pada Gambar 4.12



Gambar 4.12 Hasil implementasi tampilan 2 aplikasi

Pada Gambar 4.12 merupakan hasil implementasi tampilan 2 dengan bantuan tool PAGE, tampilan 2 adalah tampilan akhir pada aplikasi Forensic Imaging pada ponsel android dengan memanfaatkan root. Dalam tampilan 2 juga memiliki beberapa fungsi, fungsi tersebut antara lain adalah aplikasi dapat mengakuisisi media penyimpanan pada ponsel android baik penyimpanan internal maupun eksternal yang akan menghasilkan keluaran file image sebagai

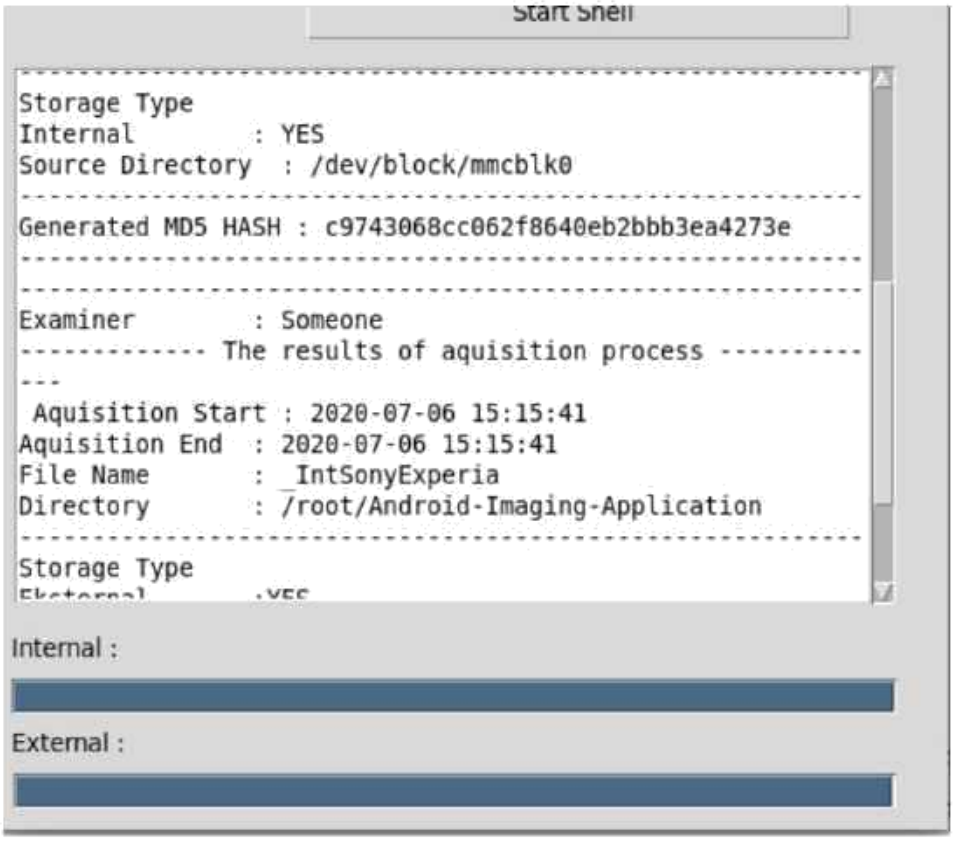
objek analisis. Sedangkan fungsi kedua adalah proses *hashing md5* dari file image yang dihasilkan pada aplikasi dengan file sumber media penyimpanan ponsel android yang akan dijadikan bukti bahwa aplikasi dapat menjaga integritas data saat dilakukannya proses akuisisi, proses hashing md5 dilakukan setelah aplikasi selesai mengakuisisi media penyimpanan ponsel android. Setelah kedua fungsi telah selesai dijalankan dengan baik maka aplikasi akan menghasilkan 3 keluaran yaitu informasi yang ada dalam aplikasi dan beberapa file, file tersebut diantaranya adalah file image dan file log sebagai keluaran fisik yang dapat dijadikan objek analisis untuk mencari informasi di dalam file tersebut.

4.2.2 Hasil Proses Akuisisi dari Aplikasi

Hasil yang didapatkan dari melakukan proses akuisisi media penyimpanan pada ponsel android baik itu media penyimpanan internal maupun eksternal dalam aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *root* yaitu 3 hasil yang didapatkan, antara lain adalah informasi dalam proses akuisisi dan menghasilkan keluaran berupa *file image* dan *file log*. Untuk penjelasan 3 hasil tersebut dapat diuraikan sebagai berikut :

a. Informasi dalam aplikasi

Hasil yang didapatkan setelah melakukan proses akuisisi media penyimpanan pada ponsel android baik media penyimpanan internal maupun eksternal salah satunya adalah informasi mengenai proses akuisisi yang telah dilakukan di dalam aplikasi. informasi tersebut antara lain adalah nama *fileimage* yang dihasilkan, rincian direktori peletakan *file image* di dalam suatu folder pada laptop/host, informasi hasil proses *hashing md5* dari file sumber penyimpanan pada ponsel android dengan *file image* yang dihasilkan aplikasi, waktu mulai dan selesai proses akuisisi, nama media penyimpanan yang diakuisisi pada aplikasi seperti media penyimpanan internal maupun eksternal dan direktori sumber media penyimpanan pada ponsel android yang diakuisisi. Untuk melihat hasil informasi dalam aplikasi dapat dilihat pada Gambar 4.13.



Gambar 4. 13 Informasi dalam aplikasi

b. File Image atau *Disk Image*

Hasil kedua yang didapatkan setelah melakukan proses akuisisi media penyimpanan pada ponsel android baik itu media penyimpanan internal maupun eksternal dalam aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *root* adalah berupa *file image* atau yang sering disebut juga *disk image* yang berformatkan file (.dd), *file image* adalah suatu file yang berisikan data-data yang tersimpan pada media penyimpanan ponsel android baik itu media penyimpanan internal maupun eksternal yang didapatkan setelah proses akuisisi media penyimpanan pada ponsel android selesai dijalankan, dari *file image* tersebut dapat dilakukan analisis secara menyeluruh dengan tujuan untuk mencari informasi-informasi yang bisa dijadikan barang bukti yang sah untuk proses peradilan dengan menggunakan aplikasi forensika digital saat ini.

Aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *root* akan menghasilkan 2 *file image* dari media penyimpanan ponsel android baik itu media penyimpanan internal maupun eksternal, jika pengguna aplikasi melakukan proses akuisisi secara bersamaan dalam aplikasi, akan tetapi jika pengguna hanya memilih salah satu media

penyimpanan yang diakuisisi dalam aplikasi, maka *file image* yang dihasilkan hanya 1 *file image* dari salah satu pilihan yang ada dalam aplikasi baik itu media penyimpanan internal maupun eksternal pada ponsel android. Untuk melihat 2 *file image* yang dihasilkan aplikasi dapat dilihat pada Gambar 4.14.

Gambar 4.14 Hasil keluaran aplikasi berupa file image

c. File Log.

Hasil ketiga atau hasil terakhir yang dihasilkan aplikasi Forensic Imaging pada ponsel android dengan memanfaatkan root adalah file log, file log adalah file yang berisikan informasi-informasi mengenai seluruh proses akuisisi yang dilakukan oleh aplikasi namun yang dihasilkan aplikasi hanya informasi yang penting saja. Seperti nama pengguna aplikasi (Officer Name), nama penyimpanan (storage) baik itu internal maupun eksternal, nama file image yang dihasilkan aplikasi, waktu mulai dan selesai proses akuisisi dan hashing md5 dari file image dengan sumber media penyimpanan. Untuk melihat rincian file log yang dihasilkan dapat dilihat pada Gambar 4.15.

Gambar 4.15 Rincian file log

4.2.3 Hasil Pengujian Kuesioner Skala Likert Aplikasi

Pada tahapan ini yaitu menjelaskan hasil yang didapatkan dari pengujian kuesioner Skala Likert yang telah penulis sebarkan ke melalui *Google form* untuk menanggapi terhadap aplikasi *Forensic imaging* pada ponsel android dengan memanfaatkan *root.* yang dapat menentukan hasil dari kuesioner tersebut dengan menggunakan rumus Rentang Skala (RS) yang dapat dilihat pada rumus (3.6) akan tetapi untuk menghitung dengan rumus Rentang Skala, pertama yang dilakukan adalah mendapatkan total skor dari setiap pernyataan maupun pertanyaan yang ada dalam kuesioner tersebut dengan menggunakan rumus (3.3) lalu dihitung rata-rata setiap pertanyaan maupun pernyataan dengan menggunakan rumus (3.4). Setelah mendapatkan hasil nilai rata-rata dari setiap pernyataan maupun pertanyaan dapat dilihat pada Tabel, selanjutnya dapat menentukan hasil kuesioner yang dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil pengujian kuesioner skala likert

No	Aspek	Nomer pertanyaan atau pernyataan	Nilai rata-rata setiap pertanyaan atau pernyataan	Hasil nilai rata-rata	Keterangan
1	Manfaat	1,2,3,4,5	4,18 + 4,14 + 4,12 + 4,34 + 4,38 = 21.15 / 5	4.23	Sangat setuju
2	Tampilan	1,2,3,4	4,12 + 4,20 + 4,22 + 4,00 = 16,54 / 4	4,13	Menarik
3	Fungsionalitas	1,2,3,4,5,6	4,04 + 4,08 + 4,36+ 4,06 + 4,04 + 4,00= 24,58 / 6	4,09	Baik
Total Rata-rata Keseluruhan				4,15	Bermanfaat

Setelah kuesioner yang diperoleh dari 50 responden yang ditujukan kepada seorang yang mengerti dalam bidang forensika digital, dan data telah diolah dengan menggunakan skala likert yang dapat dilihat pada Tabel 4.1. dari hasil pengujian ini didapatkan hasil dalam aspek manfaat penggunaan aplikasi mendapatkan nilai rata-rata sebesar 4,23/5,00 yang dikategorikan kedalam "sangat setuju", sedangkan pada aspek tampilan mendapatkan nilai rata-rata 4,13/5,00 yang dapat dikategorikan kedalam "menarik" dan aspek fungsionalitas mendapatkan hasil nilai rata-rata 4,09/5,00 yang dapat dikategorikan "baik". Dan perhitungan dari segala aspek yang terkait dalam kuesioner, selanjutnya penulis melakukan perhitungan dengan rumus rentang skala (RS) likert untuk memperoleh hasil perhitungan rata-rata dari segala aspek yang terkait

dalam kuesioner dan mendapatkan hasil kesimpulan bahwa aplikasi bermanfaat untuk digunakan oleh seseorang yang bergelut pada bidang forensika digital.

4.2.4 Hasil Pengujian Aplikasi oleh Pakar

Dalam pengujian oleh pakar yaitu melakukan pengujian aplikasi yang ditujukan kepada ahli dalam bidang forensika digital untuk menanggapi suatu kuesioner yang telah dibuat oleh penulis terhadap aplikasi. ahli forensika digital tersebut bernama bapak Yusuf Hadiwinata Sutandar, RHCT., RHCVA., RHCI, RHCX., RHCSA., RHOS., CEI., CEH., CHFI., CND., EDRP., CCNA., MTCNA. Beliau merupakan seorang Vice President Operation & Services PT. Biznet Gio Nusantara selain Vice President Operation & Service di PT. Biznet Gio Nusantara beliau juga seorang ketua dalam komunitas yang bernama Forensika id. Adapun peneliti memberikan uraian pertanyaan-pertanyaan yang penulis ajukan kepada pakar forensika digital seputar aplikasi yang diujikan dan disertai hasil dari kuesioner dalam pengujian aplikasi oleh pakar. Berikut adalah uraian pertanyaan dan juga hasilnya :

Tabel 4. 2 Hasil pengujian aplikasi oleh pakar

No	URAIAN	SKOR				
A.	**Manfaat**	**1**	**2**	**3**	**4**	**5**
1.	Aplikasi Forensics Imaging pada ponsel android dengan memanfaatkan root ini memiliki manfaat bagi saya yang bergelut pada bidang forensika digital ?				✓	
2.	Aplikasi Forensics Imaging pada ponsel android dengan memanfaatkan root ini mudah digunakan ?				✓	
3.	Menggunakan aplikasi ini pada saat ingin melakukan Cloning data pada penyimpanan Android ?			✓		
4.	Saya berharap aplikasi ini digunakan oleh orang banyak yang bergelut pada bidang forensika digital ?				✓	
5.	Keluaran aplikasi ini sudah sesuai dengan yang saya harapkan ?				✓	
B.	**Tampilan**	**1**	**2**	**3**	**4**	**5**
1.	Tampilan antarmuka aplikasi ini menarik bagi saya?				✓	

2.	Tampilan antarmuka aplikasi ini mudah dikenali ?				✓	
3.	Menu yang ditampilkan pada aplikasi ini mudah dipahami?				✓	
4.	Peletakan semua fitur dan tombol aplikasi sudah sesuai dengan yang diharapkan?				✓	
C.	**Fungsionalitas**	**1**	**2**	**3**	**4**	**5**
1.	Fitur untuk penginstallan root sudah berjalan dengan semestinya?				✓	
2.	Fitur untuk penginstalan busybox sudah berjalan dengan semestinya?				✓	
3.	Fitur untuk scanning perangkat berjalan dengan baik?				✓	
4.	Fitur untuk mengkoneksikan perangkat sudah berjalan dengan baik?				✓	
5.	Fitur untuk proses akuisisi penyimpanan pada ponsel android sudah berjalan dengan baik?				✓	
6.	Semua tombol dapat bekerja dengan normal ?				✓	
Total Rata-rata Keseluruhan		3,80 + 4 + 4 = 11,8/3 = 3,93				
		Bermanfaat				

Setelah pertanyaan dalam kuesioner yang telah penulis buat telah ditanggapi atau dijawab oleh seorang ahli pada bidang forensika digital kemudian data tersebut dikelola dengan menggunakan perhitungan pada skala likert yang dapat dilihat pada Tabel. Dari hasil pengujian ini dapat disimpulkan dari ketiga aspek dalam pertanyaan kuesioner, mendapatkan hasil dari aspek manfaat penggunaan nilai aplikasi sebesar 3,8/5,00 yang termasuk dalam kategori "Bermanfaat", sedangkan dalam aspek tampilan aplikasi mendapatkan nilai sebesar 4/5,00 yang termasuk kategori "Menarik", sedangkan aspek ketiga aplikasi adalah aspek fungsionalitas dalam aplikasi mendapatkan nilai sebesar 4/5,00 yang termasuk dalam kategori " ", selanjutnya dari nilai ketiga aspek tersebut dilakukan perhitungan rentang skala (RS) likert dengan mendapatkan nilai 3,93 dengan nilai tersebut dapat disimpulkan bahwa aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *Root* adalah bermanfaat untuk digunakan oleh seorang yang bergelut pada bidang forensika digital.

4.3 Pengujian

4.3.1 Proses Pengujian Aplikasi

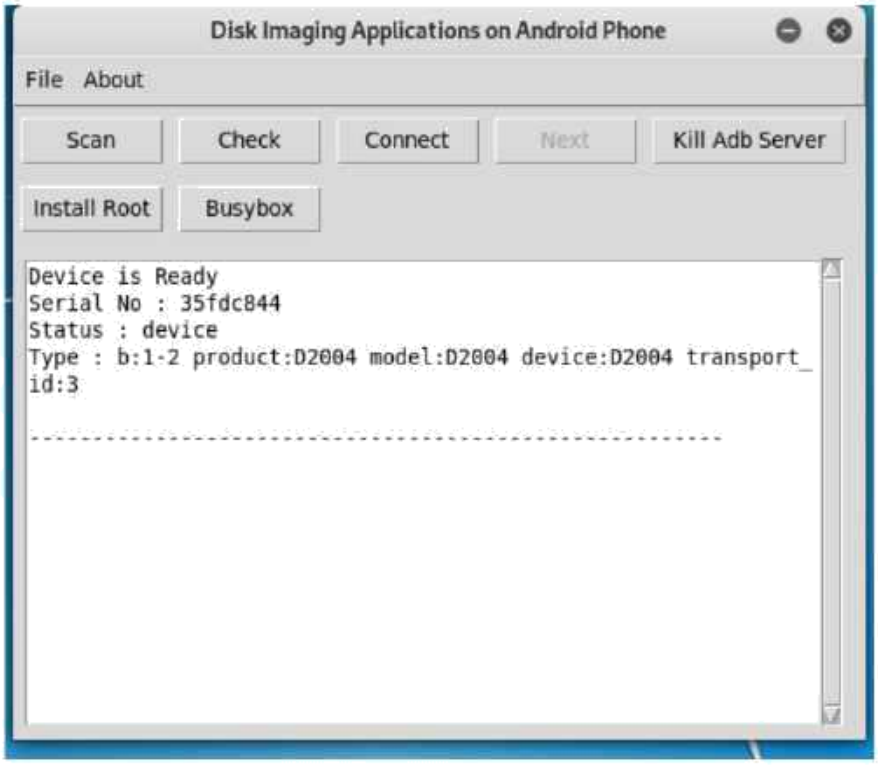
Pada tahapan ini adalah melakukan pengujian terhadap aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *root* yang telah dikembangkan dengan cara menggunakan aplikasi secara langsung untuk melakukan proses akuisisi media penyimpanan ponsel android. Dan akan melakukan sebuah pengujian, pengujian tesebut adalah pengujian dengan metode *black box testing* yang terdiri dari pengujian fungsionalitas dan *error handling* pada tombol-tombol yang ada didalam aplikasi, pengujian performa aplikasi untuk melihat kecepatan transfer data pada saat proses akuisisi dijalankan dan melakukan perhitungan dari pengujian kuesioner skala likert yang telah disebarkan ke responden untuk menanggapi terhadap aplikasi yang telah dibuat.

4.3.2 Pengujian *Black box*

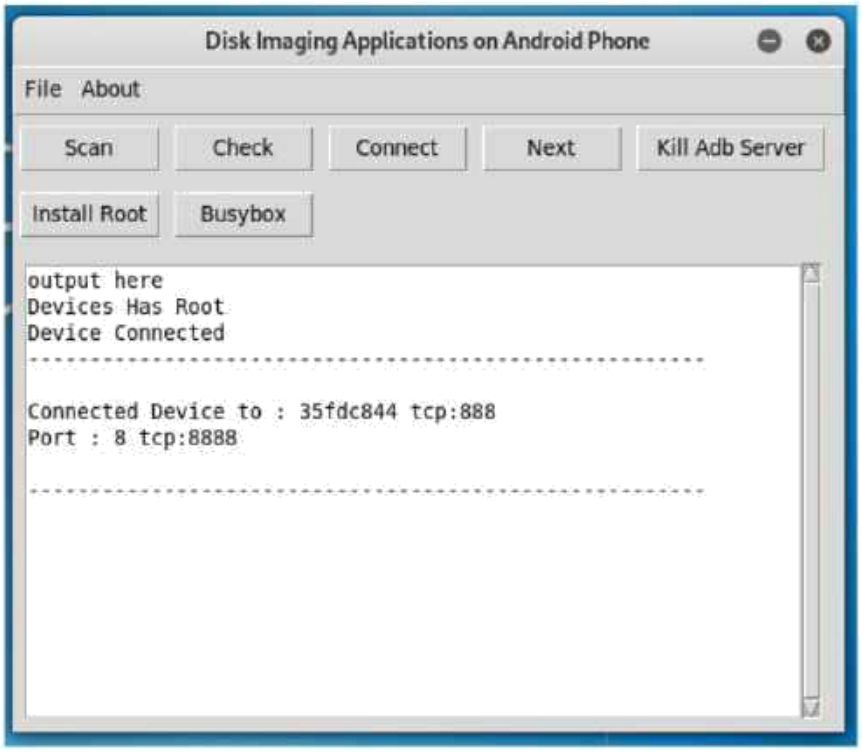
Proses pengujian dengan menggunakan *black box testing* adalah pengujian yang dilakukan dengan cara mengamati secara langsung ke dalam tampilan aplikasi, adapun pengujian yang dilakukan oleh penulis adalah pengujian fungsionalistas dan *error handling* pada tombol-tombol yang ada didalam aplikasi.

4.3.3 Pengujian Fungsionalitas Tombol

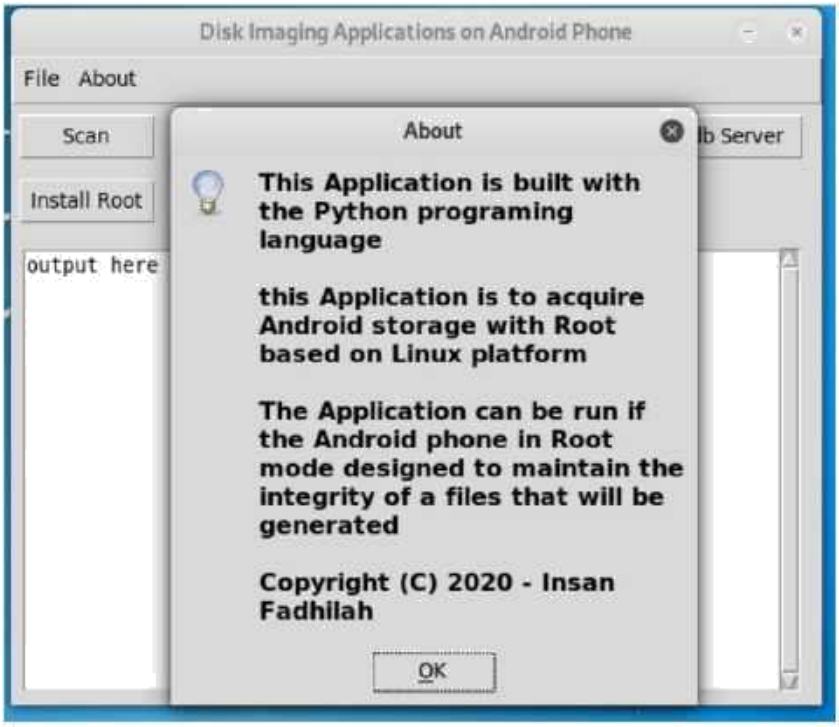
Pada tahapan ini melakukan pengujian terhadap fungsionalitas tombol yang ada didalam aplikasi, pengujian yang pertama diuji adalah pengujian tombol-tombol yang terdapat pada tampilan 1 aplikasi, tombol yang diuji diantaranya adalah tombol scan, tombol connect, tombol *kill adb-server* dan tombol check, untuk pengujian tombol "scan" dapat dilihat pada Gambar 4.16, sedangkan untuk tombol check dapat dilihat pada Gambar 4.17, tombol about yang berfungsi untuk menampilkan informasi tentang aplikasi dapat dilihat pada Gambar 4.18, tombol exit berfungsi untuk keluar dari aplikasi dapat dilihat pada Gambar 4.19 dan yang terakhir untuk tombol *kill Adb-server* dapat dilihat pada Gambar 4.20. Hasil dari pengujian tombol menu aplikasi ternyata tombol tersebut dapat bekerja dengan baik dan menjalankan sesuai fungsinya.



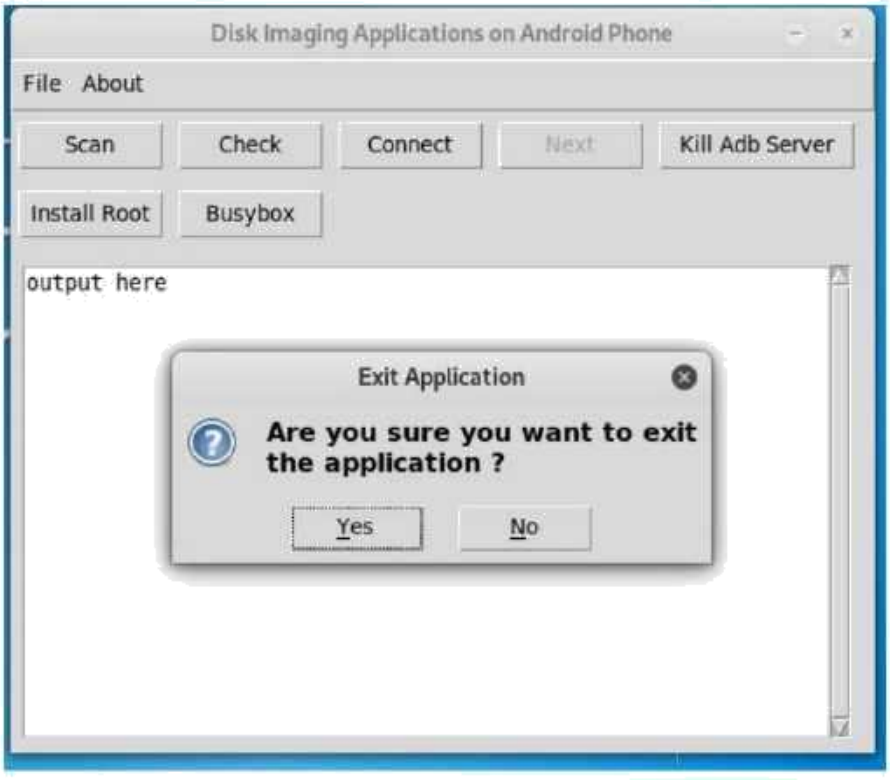
Gambar 4.16 Hasil pengujian tombol scan



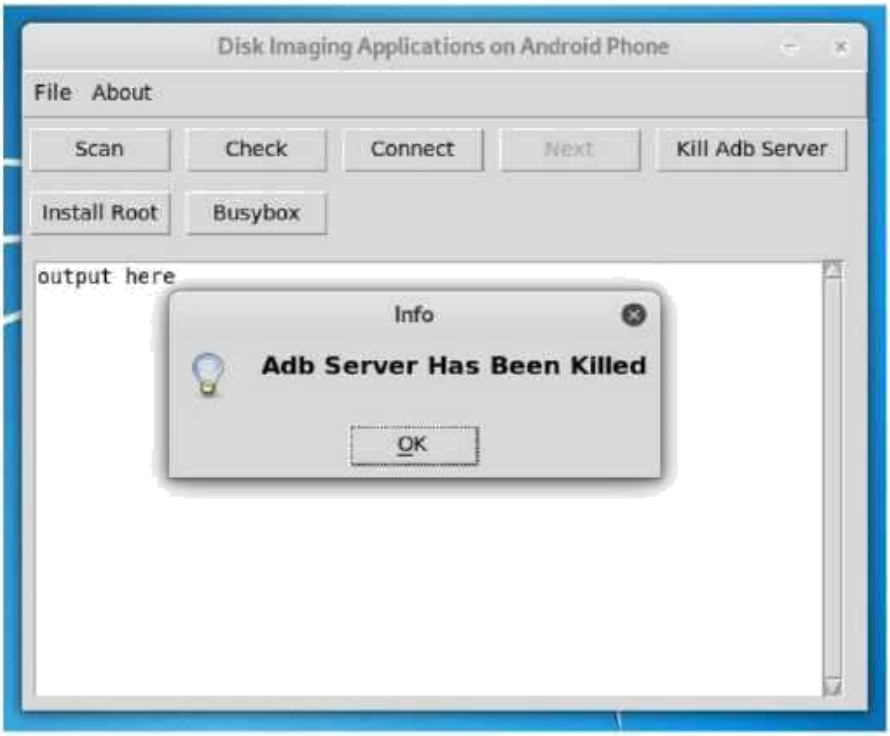
Gambar 4.17 Hasil pengujian tombol check



Gambar 4.18 Hasil pengujian tombol about

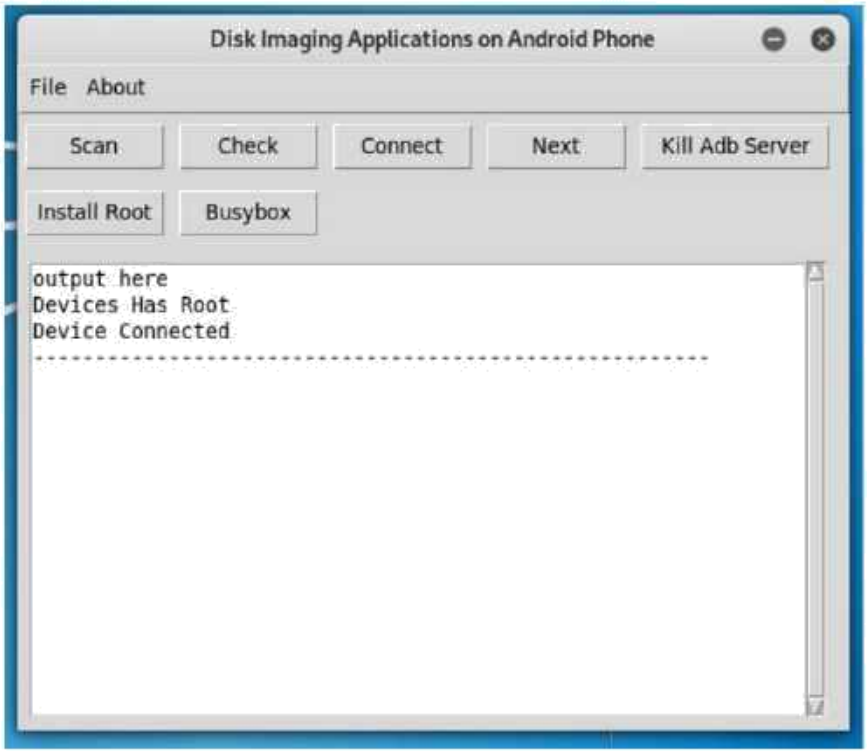


Gambar 4.19 Hasil pengujian tombol exit



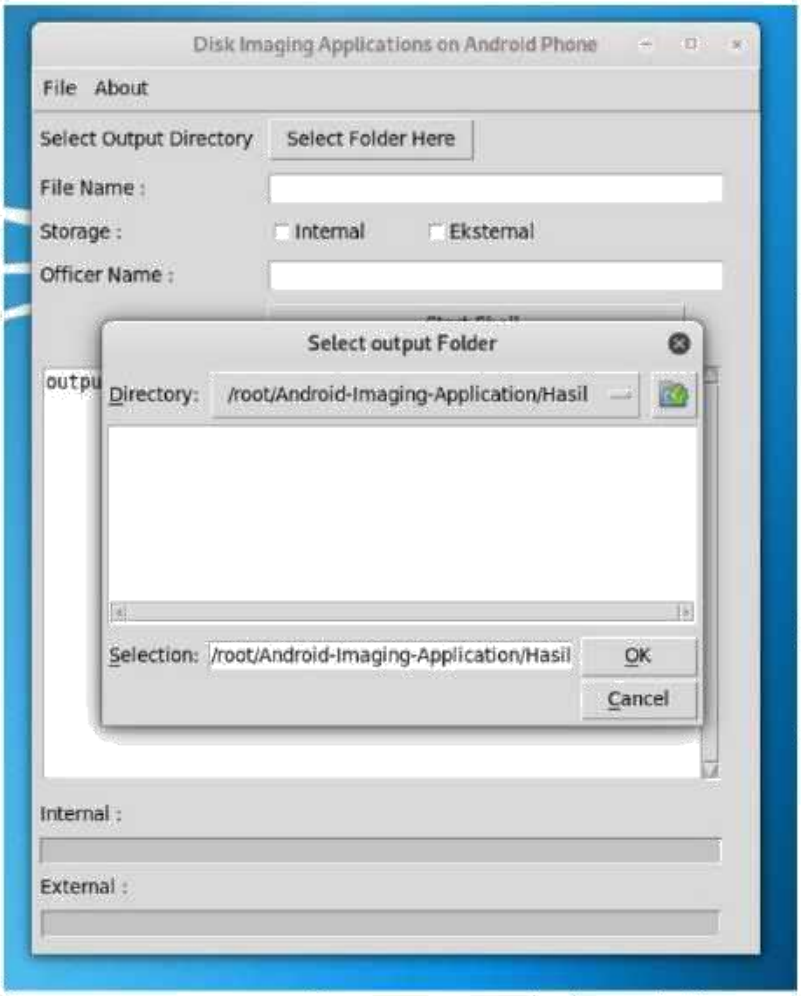
Gambar 4.20 Hasil pengujian tombol kill adb-server

Tombol "scan" berfungsi untuk menjalankan proses *scanning* perangkat ponsel android yang terhubung dengan perangkat laptop menggunakan kabel USB dan akan menghasilkan keluaran informasi tentang *serial number, status, state* dan *type* perangkat ponsel android yang ditemukan aplikasi. sedangkan tombol connect berfungsi untuk menjalankan proses mengkoneksikan perangkat ponsel android pada aplikasi menggunakan port tertentu dan berfungsi juga untuk mengaktifkan tombol "next" jika mengkoneksikan perangkat ponsel android dan aplikasi berhasil. Dan tombol "check" berguna untuk mencari perangkat ponsel android yang sudah terhubung ke aplikasi dengan menggunakan port tertentu.selanjutnya tombol exit pada sub menu berfungsi untuk keluar dari aplikasi, tombol kill adb server berfungsi untuk memberhentikan ADB pada port tertentu pada laptop/host, tombol next berfungsi untuk melanjutkan ke tampilan 2 aplikasi untuk keperluan proses akuisisi perangkat penyimpanan ponsel android. Namun tombol next akan aktif apabila sudah melakukan koneksi terhadap ponsel android. Untuk melihat hasil pengujian tombol next dapat dilihat pada Gambar 4.21. Dari pengujian tombol yang dilakukan untuk menjalankan semua proses yang dibutuhkan dalam tampilan 1 hasilnya bekerja dan berfungsi dengan baik.



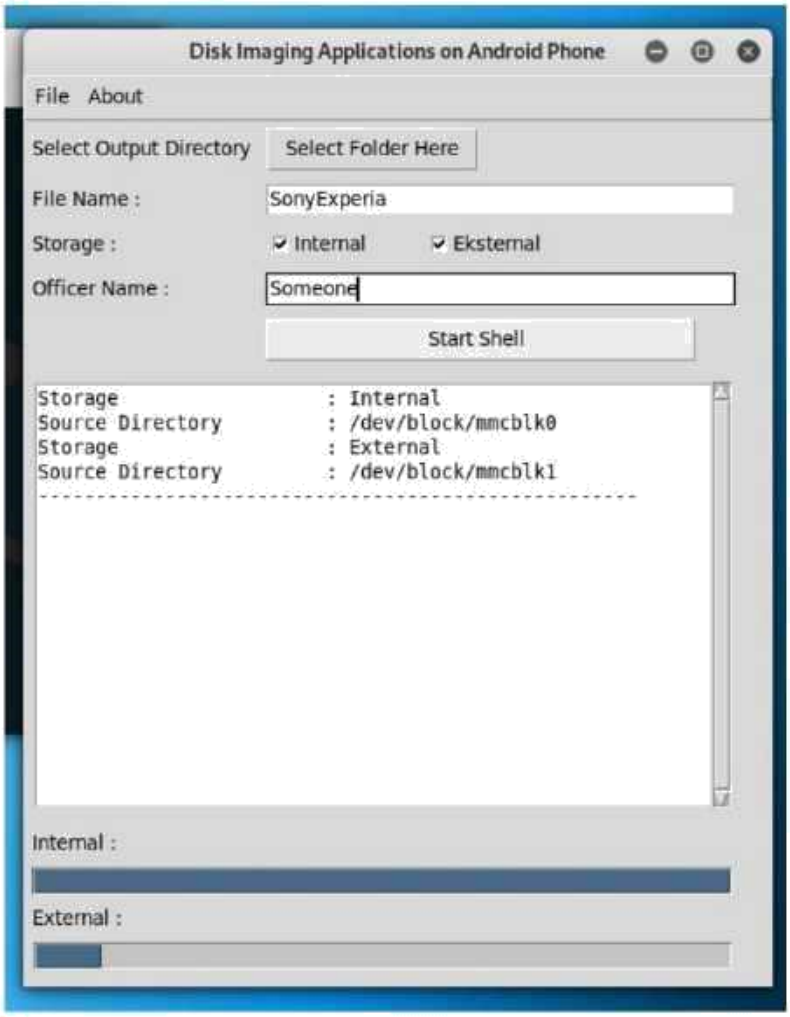
Gambar 4.21 Hasil pengujian tombol next

Setelah proses pada tampilan 1 selesai maka dilanjutkan pada tampilan 2 yang akan diuji fungsionalitas tombolnya. Tampilan 2 aplikasi terdiri dari beberapa tombol yang memiliki masing-masing. Tombol pertama yang diuji adalah tombol select output directory yang bertempat pada samping kolom output location yang berfungsi untuk memilih suatu folder dalam perangkat laptop/host untuk menentukan peletakan *file image* dan *file log* yang akan dihasilkan aplikasi nantinya. Untuk pengujian select output directory sudah bekerja dengan sebagaimana mestinya atau tidak, maka penulis mengisi kolom output location dengan direktori "/root/Android-Imaging-Aplication/Hasil" yang dapat dilihat pada Gambar 4.22 dan untuk melihat hasil dalam kolom output location aplikasi dapat dilihat pada Gambar 4.22. Dari hasil pengujian tombol select output direktori ternyata tombol berjalan dengan baik sesuai dengan fungsinya.



Gambar 4.22 Hasil pengujian tombol select output directory

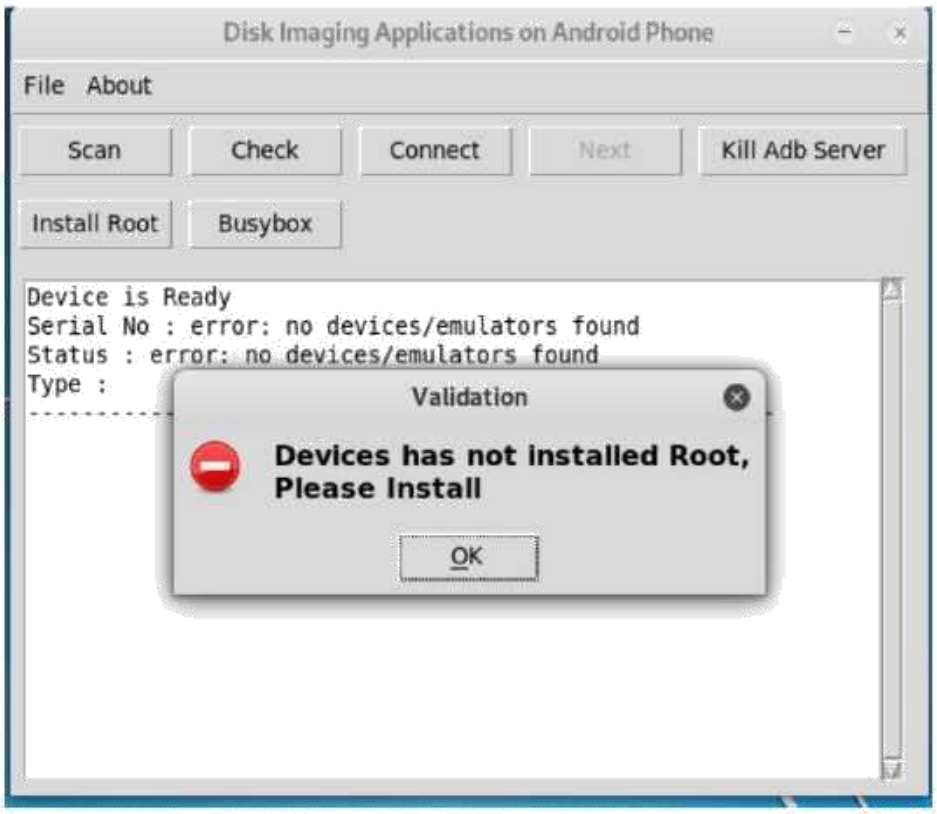
Selanjutnya pengujian fungsionalitas tombol "start shell" pada tampilan 2 aplikasi yang berfungsi untuk menjalankan proses akuisisi penyimpanan pada perangkat ponsel android dalam aplikasi dapat dilihat pada Gambar 4.23. Dari pengujian tombol "start shell" ternyata tombol tersebut berjalan dengan baik dan sesuai dengan fungsinya.



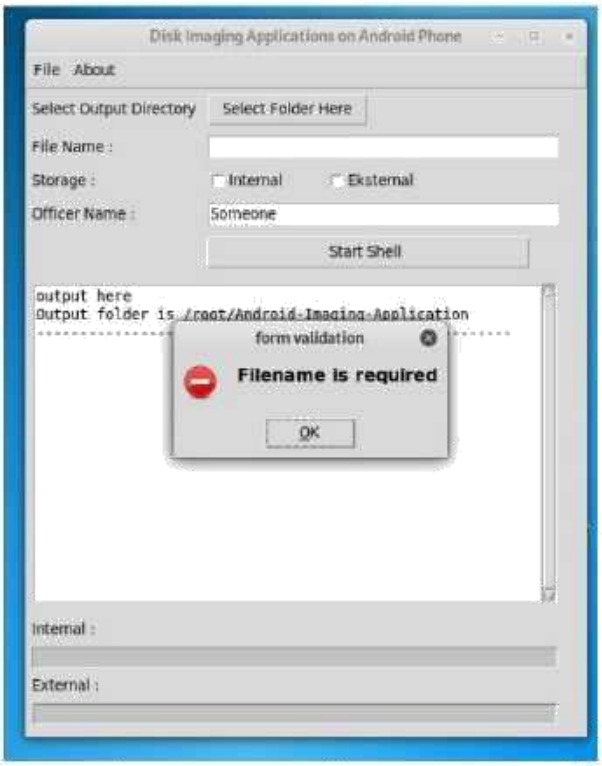
Gambar 4.23 Hasil pengujian tombol start shell

4.3.4 Pengujian Error Handling

Tahap ini berisi pengujian *error handling* yang ada pada tombol dalam aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *root*. pada pengujian tahap ini dilakukan terhadap tombol-tombol yang memungkinkan aka nada kesalahan penggunaan aplikasi, salah satu tombol yang memiliki sifat *error handling* adalah tombol connect pada tampilan 1 aplikasi. tombol tersebut memberikan informasi pada *dialog box* yang akan mendeteksi kesalahan pengguna jika ponsel android yang terhubung dengan laptop belum terinstall aplikasi *busybox* dan belum diberikan hak akses *superuser* ataupun tidak ada perangkat ponsel yang terhubung dengan laptop. Jika informasi yang diberikan bahwa ponsel belum terinstall aplikasi *busybox* dan belum diberikan hak akses *superuser* maka aplikasi tidak dapat melakukan proses mengkoneksikan ponsel android dengan aplikasi dan aplikasi juga tidak akan dapat mengaktifkan tombol "next" yang dapat dilihat pada Gambar 4.24. Dan selanjutnya tombol "start shell" adalah tombol yang akan berfungsi jika semua kolom dalam *form* tampilan 2 sudah diisikan dengan lengkap dan benar, jika semua kolom belum diisi ataupun jika nama file sudah ada di dalam folder maka aplikasi akan menampilkan dialog box yang bisa dilihat pada Gambar 4.25. Untuk melihat *error handling* tampilan 2 aplikasi dapat dilihat pada Gambar berikut :



Gambar 4.24 Hasil pengujian error handling tombol connect



Gambar 4.25 hasil pengujian error handling tombol start shell

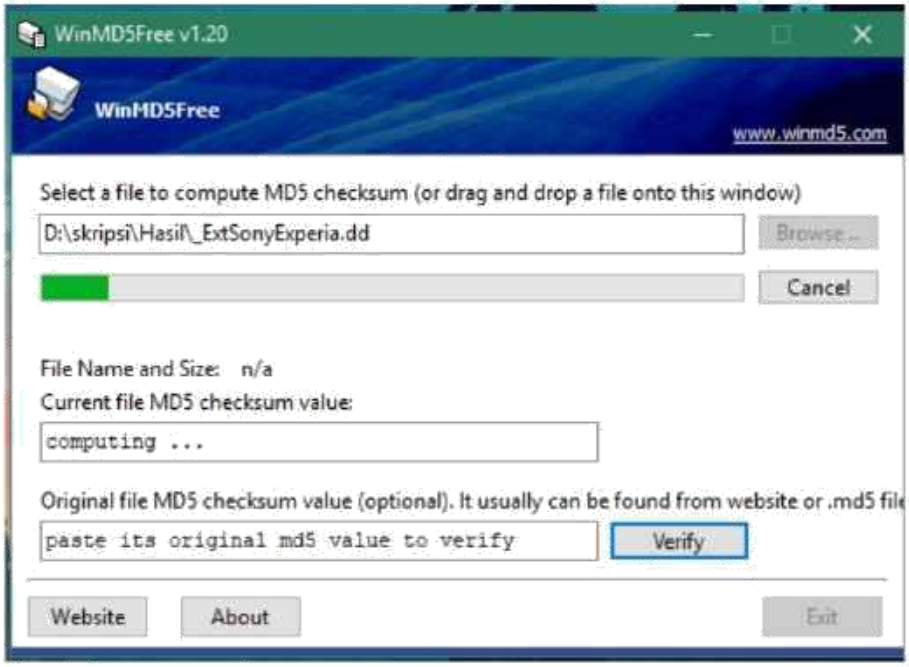
4.3.5 Pengujian Integritas Data *file Image* dari Aplikasi

Tahap ini penulis melakukan pengujian perbandingan integritas data dari *file image file image* penyimpanan external pada ponsel android yang dihasilkan aplikasi dengan menggunakan modul *hashlib* yang terdapat dalam bahasa pemrograman *python* dengan

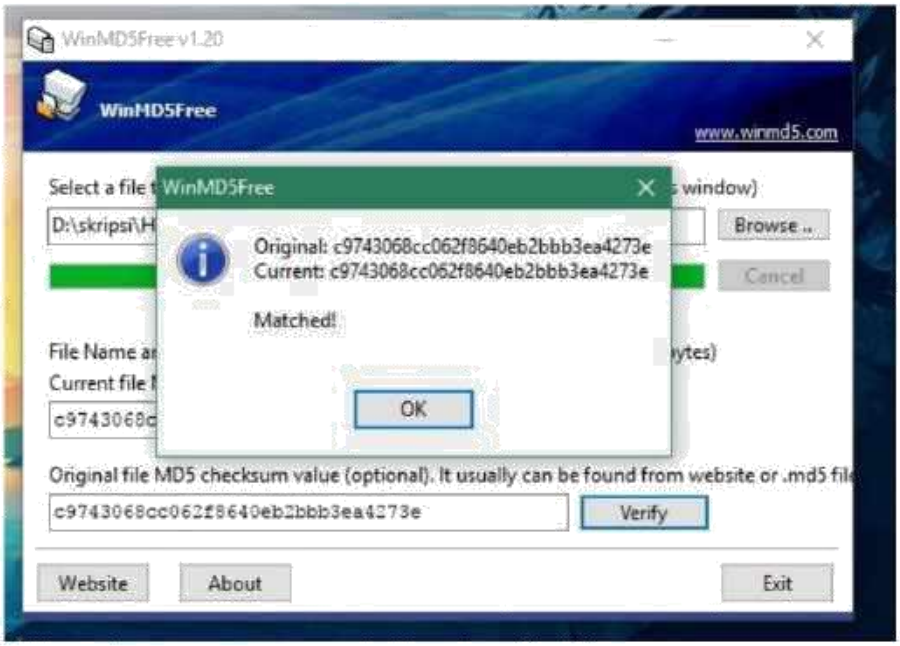
menggunakan aplikasi WinMD5 di sistem operasi windows.WinMD5 berfungsi untuk proses pengujian pencocokan *md5-checksum* untuk mengambil hasil md5 dari *file image* yang dihasilkan oleh aplikasi dapat dilihat pada Gambar 4.26. Sedangkan untuk melihat hasil proses pencocokan *md5* dengan menggunakan aplikasi WinMD5 pada sistem operasi windows dapat dilihat pada Gambar 4.27. Setelah proses pencocokan hasil *md5* yang dilakukan WinMD5 telah selesai maka hasilnya dapat dilihat pada Gambar 4.28 dan dari hasil itu menyatakan bahwa hasil md5 *file image* yang dihasilkan aplikasi yang tersimpan dalam *file log* dengan kalkulasi *md5* menggunakan aplikasi WinMD5 pada sistem operasi windows ternyata hasilnya sama yang artinya bahwa integritas data *file image* yang dihasilkan aplikasi terjaga.

```
Examiner              : Someone------------ The results of aquisition process ------------
Aquisition Start      : 2020-07-06 14:16:38
Aquisition End        : 2020-07-06 15:15:41
File Name             : _ExtSonyExperia
Directory             : /root/Android-Imaging-Application
-------------------------------------------------------------
Storage Type
Internal              : YES
Size                  : 3.640625
Eksternal             : YES
Size                  : 15.2353515625
-------------------------------------------------------------
Source MD5 Hash       : c9743068cc062f8640eb2bbb3ea4273e
Generated MD5 HASH    : c9743068cc062f8640eb2bbb3ea4273e
-------------------------------------------------------------
MD5 Hash Matched
```

Gambar 4.26 Hasil md5 yang dihasilkan oleh aplikasi



Gambar 4.27 Proses pencocokan dengan WinMD5



Gambar 4.28 Hasil pencocokan dengan menggunakan WinMD5

4.3.6 Pengujian Performa Aplikasi

Tahapan ini berisikan tahapan dalam pengujian performa aplikasi saat melakukan proses akuisisi media penyimpanan ponsel android didalam aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *root*. pengujian dilakukan atas 2 ponsel android yaitu Sony Experia E bermodel D2004 dan Advan S5E bermodelkan 5001, kedua ponsel tersebut memiliki media penyimpanan internal sebesar 4GB dan untuk media penyimpanan eksternal yang diuji yaitu 16GB. Pengujian aplikasi dapat dilihat dari proses kecepatan *transfer data* saat dilakukannya proses akuisisi dari media penyimpanan internal maupun eksternal yang ada dalam perangkat ponsel android. Pengujian performa aplikasi sendiri dilakukan dengan 2 skema, diantaranya sebagai berikut :

a. Mengakuisisi Kedua penyimpanan secara bersamaan

Pada skema ini melakukan proses akuisisi kedua media penyimpanan yang ada di dalam ponsel android secara bersamaan di dalam aplikasi. dengan melakukan akuisisi kedua media penyimpanan secara bersamaan akan berpengaruh pada kecepatan *transfer data* yaitu turun 11% hingga 13% dari kecepatan *transfer data* dengan melakukan akuisisi salah satu penyimpanan. Untuk melihat hasil rincian yang diperoleh dari proses akuisisi media penyimpanan secara bersamaan di dalam aplikasi dapat dilihat pada Tabel 4.3.

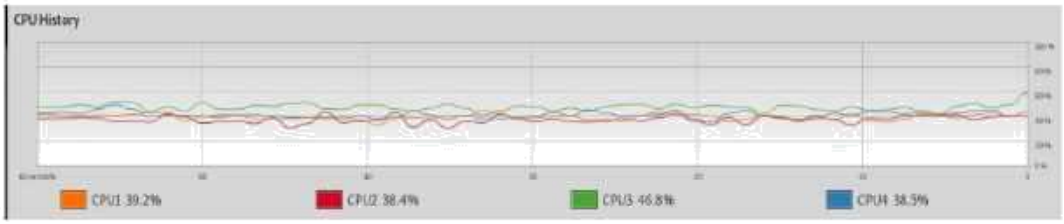
Tabel 4.3 Hasil akusisi 2 media penyimpanan secara bersamaan

No	Perangkat android	Sampel	Penyimpanan	Size	Total Waktu	Rata-rata kecepatan transfer data
1	Sony Experia E	1	Internal	4GB	11 Menit	7.4 MB/detik
			Eksternal	16GB	34 Menit	8.4 MB/detik
2	Advan S5E	1	Internal	4GB	14 Menit	6 MB/detik
			Eksternal	16GB	42 Menit	5.8 MB/detik

Pada Tabel 4.3 adalah hasil rincian yang diperoleh dari performa proses akuisisi yang dilakukan pada aplikasi saat melakukan proses kedua penyimpanan secara bersamaan baik itu internal maupun eksternal. Pengujian performa aplikasi diuji dari 2 perangkat ponsel android. Perangkat ponsel pertama dengan mengakuisisi 2 media penyimpanan secara bersamaan yaitu media penyimpanan internal 4Gb dan media penyimpanan eksternal sebesar 16GB. Dan perangkat ponsel kedua yang diuji adalah media penyimpanan internal sebesar 4GB dan eksternal 16GB.

Pada pengujian performa aplikasi pada perangkat ponsel 1 yang mempunyai ukuran media penyimpanan internal sebesar 4GB dan eksternal sebesar 16GB saat dilakukan proses akuisisi kedua media penyimpanan secara bersamaan melalui aplikasi diperoleh kecepatan transfer data rata-rata untuk media penyimpanan sebesar 4GB yaitu 7.4 MB/detik dengan durasi waktu selama 11 menit dan untuk media penyimpanan eksternal sebesar 16GB diperoleh transfer data sebesar 8.4MB/detik dengan lama waktu 34 menit yang dapat dilihat pada Tabel 4.3 dari hasil tersebut dipengaruhi oleh beberapa hal yaitu penggunaan perangkat lunak Netcat dan menggunakan protocol TCP melalui sebuah port. Port TCP (*Transmission Control Protocol)* adalah sebuah jaringan untuk mengirimkan sebuah paket data yang besar yang akan dikirimkan dan diperiksa kembali. Kelebihan menggunakan port TCP adalah baik digunakan untuk pengiriman sebuah paket data yang besar dengan keamanan yang bagus, walaupun membutuhkan waktu yang lebih lama. Sedangkan untuk penggunaan perangkat lunak netcat menurut data center host rata-rata kecepatan perangkat lunak dd dengan necat adalah 6MB/detik namun penulis juga melakukan monitoring penggunaan sumber data perangkat laptop dengan menggunakan tools system monitoring pada sistem operasi linux. Setelah dilakukannya monitoring maka mendapatkan hasil dari beberapa subjek seperti penggunaan

CPU dalam perangkat laptop yang memiliki 4 CPU. CPU pertama maximum penggunaan 50% dan mendapatkan rata-rata sebesar 42.2%, CPU kedua maksimum penggunaan sebesar 45% dengan rata-rata penggunaan sebesar 39.4%. CPU 3 penggunaan maksimum adalah 60% dengan rata-rata penggunaan sebesar 46.2% dan yang terakhir CPU 4 penggunaan maksimum sama seperti CPU 3 yaitu sebesar 60% dan penggunaan rata-rata sebesar 42.2% yang dapat dilihat pada Gambar 4.29. sedangkan penggunaan memori sebesar 17.2% yaitu 1.3 GB dari total memori yang tersedia adalah 7.7 GB yang dapat dilihat pada Gambar 4.30. Dari hasil monitoring sumber data perangangkat laptop dapat disimpulkan perangkat laptop mencapai penggunaan maksimun CPU dan memori belum maksimal kemungkinan dikarenakan oleh penggunaan protocol TCP dan Netcat.



Gambar 4.29 Hasil penggunaan CPU pada pengujian performa aplikasi

Gambar 4.30 Hasil penggunaan memori pada pengujian aplikasi

Dari pengujian pada 2 perangkat ponsel android dilakukan masing-masing 1 sampel yang terlihat pada Tabel 4.3. Pada sampel pertama didapatkan transfer data untuk media penyimpanan internal 4GB sebesar 7.4 MB/detik dengan durasi waktu selama 11 menit, selanjutnya transfer kecepatan data untuk media penyimpanan eksternal 16GB sebesar 8.4 MB/detik dengan durasi waktu selama 34 menit.

b. Mengakuisisi salah satu media penyimpanan

Skema yang kedua adalah skema mengakuisisi salah satu media penyimpanan yang akan mendapatkan kecepatan transfer data yang lebih baik dari segi waktu dengan metode salah satu media penyimpanan terlebih dahulu. Untuk melihat hasil rincian yang didapat dari proses

akuisisi salah satu media penyimpanan pada perangkat ponsel android dapat dilihat pada Tabel 4.4

Tabel 4.4 Rincian Performa aplikasi mengakuisisi salah satu media penyimpanan

NO	Ponsel Android	Penyimpanan	Size	Total waktu	Rata-rata transfer data
1	Sony Experia E	Internal	4Gb	9 menit	8.4MB/detik
2	Advan S5E	Internal	4GB	14 Menit	6 MB/detik

Pada Tabel 4.4 adalah hasil rincian yang dikeluarkan dari performa proses akuisisi yang dilakukan aplikasi pada saat melakukan proses akuisisi salah satu media penyimpanan internal perangkat ponsel android. Pengujian performa aplikasi yang diujikan pada 2 perangkat ponsel android yang memiliki media penyimpanan internal masing-masing sebesar 4GB.

Pada pengujian performa aplikasi untuk melihat kecepatan transfer data salah satu media penyimpanan internal pada perangkat ponsel android didapatkan transfer data untuk perangkat ponsel android Sony Experia E yang berisikan media penyimpanan internal sebesar 4GB adalah 8.6MB/detik dengan rentang waktu selama 9 menit. Sedangkan untuk perangkat ponsel android Advan S5E yang mempunyai kapasitas media penyimpanan internal sebesar 4GB adalah 6MB/detik selama 14 menit.

4.3.7 Total Skor Kuesioner Skala Likert Aplikasi

Pada tahap ini menerangkan hasil yang didapat dari melakukan pengujian kuesioner yang dibagikan ke 50 responden dengan melakukan perhitungan skor menggunakan skala likert dengan hasil yang didapat. Untuk melihat hasil skor dari perhitungan likert dapat dilihat pada Tabel 4.5.

Tabel 4.5 Total Kuesioner Skala Linkert

No	Uraian	Skor					Total Skor
A	Manfaat	1	2	3	4	5	
1.	Aplikasi Forensics Imaging pada ponsel android dengan memanfaatkan root ini			5	31	14	(3x5) + (4x31) + (5*14) = 209:50 = 4,18

	memiliki manfaat bagi saya yang bergelut pada bidang forensika digital ?						
2.	Aplikasi Forensics Imaging pada ponsel android dengan memanfaatkan root ini mudah digunakan ?		1	5	30	14	(2x1) + (3x5) + (4x30) + (5x14) = 207:50 = 4.14
3,	Menggunakan aplikasi ini pada saat ingin melakukan Cloning data pada penyimpanan Android ?		1	5	31	13	(2x1) + (3x5) + (4x31) + (5x13) = 206:50 = 4,12
4.	Saya berharap aplikasi ini digunakan oleh orang banyak yang bergelut pada bidang forensika digital ?			2	29	19	(3x2) + (4x29) + (5x19) = 217:50 = 4,34
5.	Keluaran aplikasi ini sudah sesuai dengan yang saya harapkan ?			5	21	24	(3x5) + (4x21) + (5x24) = 219:50 = 4,38
B.	**Tampilan**	**1**	**2**	**3**	**4**	**5**	
1.	Tampilan antarmuka aplikasi ini menarik bagi saya?		1	9	23	17	(2x1) + (3x9) + (4x23) + (5x17) = 206:50 = 4,12
2.	Tampilan antarmuka aplikasi ini mudah dikenali ?		2	1	32	15	(2x2) + (3x1) + (4x32) + (5x15) = 210:50 = 4,20
3.	Menu yang ditampilkan pada aplikasi ini mudah dipahami?			7	25	18	(3x7) + (4x25) + (5x18) = 211:50 = 4,22
4.	Peletakan semua fitur dan tombol aplikasi sudah sesuai dengan yang diharapkan?			12	26	12	(3x12) + (4x26) + (5x12) = 200:50 = 4
C.	**Fungsionalitas**	**1**	**2**	**3**	**4**	**5**	

1.	Fitur untuk penginstallan root sudah berjalan dengan semestinya?			10	28	12	(3x10) + (4x28) + (5x12) = 202:50 = 4,04
2.	Fitur untuk penginstalan busybox sudah berjalan dengan semestinya?			8	30	12	(3x8) + (4x30) + (5x12) = 204:50 = 4,08
3.	Fitur untuk scanning perangkat berjalan dengan baik?			8	26	18	(3x8) + (4x26) + (5x18) = 218:50 = 4,36
4.	Fitur untuk mengkoneksikan perangkat sudah berjalan dengan baik?		1	9	26	14	(1x2) + (3x9) + (4x26) + (5x14) = 203:50 = 4,06
5.	Fitur untuk proses akusisi penyimpanan pada ponsel android sudah berjalan dengan baik?			8	32	10	(3x8) + (4x32) + (5x10) = 202:50 = 4,04
6.	Semua tombol dapat bekerja dengan normal ?		1	10	27	12	(2x1) + (3x10) + (4x27) + (5x12) = 200:50 = 4

4.4 Implementasi Hasil

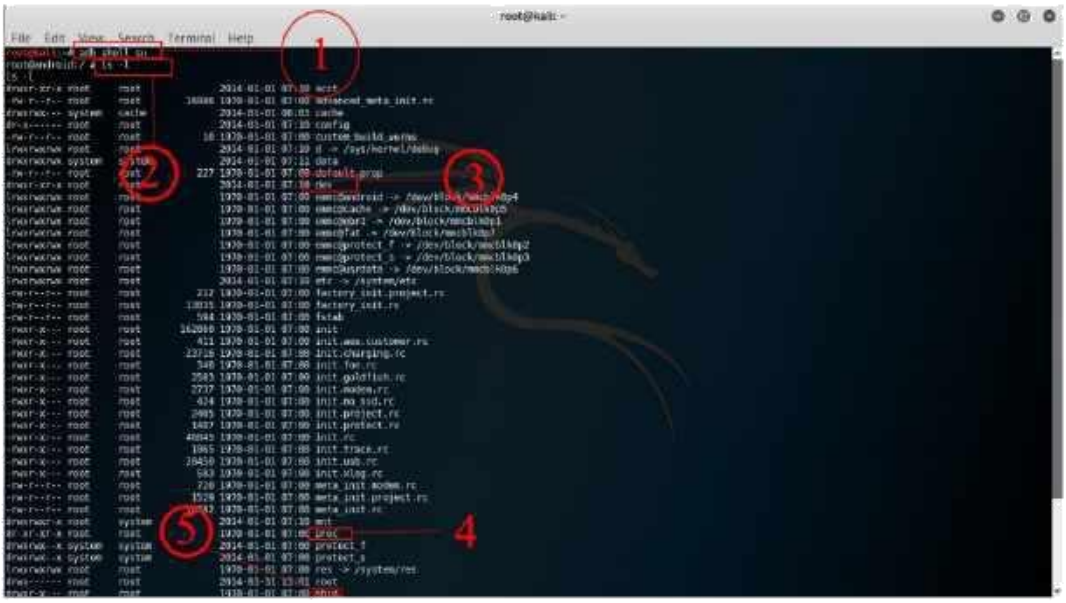
4.4.1 Analisis Model Penyimpanan Data pada Ponsel Android

Pada tahap ini penulis melakukan analisis model media penyimpanan data pada perangkat ponsel android dengan hak akses *superuser* terhadap ponsel yang dijadikan contoh barang bukti yang berguna untuk mendapatkan informasi-informasi mengenai ukuran media penyimpanan, partisi dan *file system* yang digunakan pada perangkat ponsel android. Untuk mendapatkan semua hasil informasi tersebut dilakukan beberapa analisis yang dilakukan penulis, untuk rinciannya adalah sebagai berikut:

a. Analisis *Directory Structure* perangkat ponsel android

Ponsel android memiliki struktur direktori khusus untuk peletakan dan penataan media penyimpanan, untuk mengetahui dan melihat struktur direktori pada perangkat ponsel android dengan menggunakan perintah "adb shell su" dan dapat juga melihat struktur direktori

menggunakan DDMS (*Dalvik Debug Monitor Server).* Setelah melakukan perintah "adb shell su" maka akan masuk ke dalam *shell* perangkat ponsel android dengan hak akses *superuser* dan melakukan perintah "ls –l" di dalam *shell* untuk melihat struktur direktori perangkat tersebut, untuk melihat direktori yang terdapat pada perangkat ponsel android "Sony Experia" dapat dilihat pada Gambar 4.33.



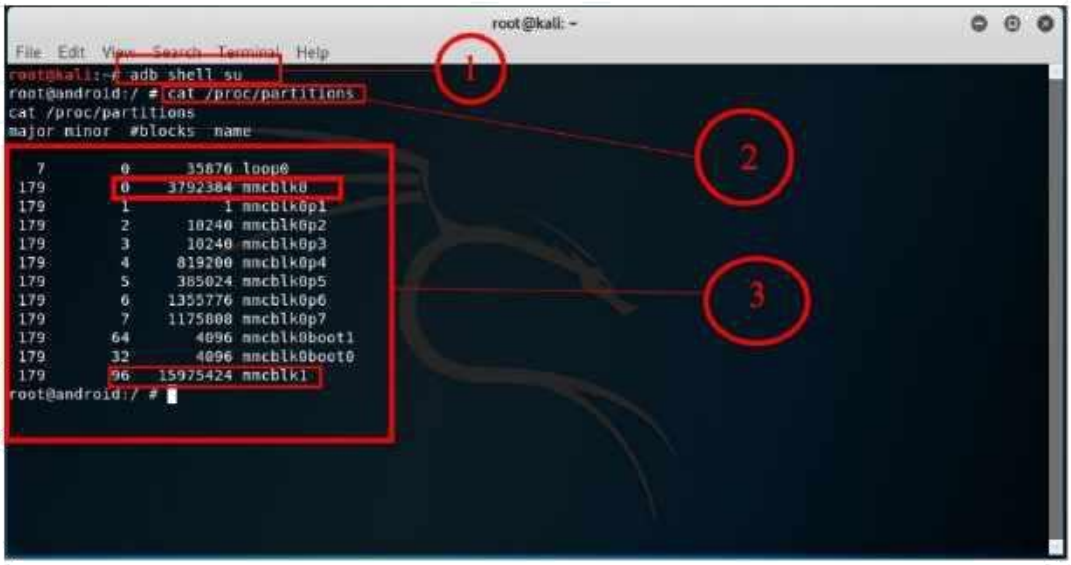
Gambar 4. 31 Hasil analisis struktur direktori perangkat

Pada Gambar 3.3 menjelaskan pada perintah (1) "adb shell su" adalah perintah adb untuk masuk ke dalam *shell* perangkat ponsel android dengan hak akses *superuser*, setelah berhasil masuk ke dalam *shell* perangkat ponsel android lalu melakukan perintah (2)"ls –l" perintah ini bertujuan untuk melihat struktur dalam perangkat ponsel android, pada (3)"/dev" merupakan nama partisi utama yang akan digunakan untuk tujuan proses akuisisi dalam perangkat ponsel android. (4)"/proc"/ merupakan partisi yang dapat mengetahui dan menampilkan penamaan partisi dalam media penyimpanan dan *filesystem* apa saja yang digunakan dalam perangkat ponsel android. (5)"/sbin" adalah partisi yang meletakan file *busybox* yang memiliki berbagai macam fungsi yang dimanfaatkan untuk proses akusisi, mengetahui ukuran penyimpanan dan lainnya.

b. Analisis *Partition Layout* perangkat ponsel Android

Setelah penulis melakukan analisis terhadap struktur direktori perangkat, kita dapat mengetahui direktori apa saja yang menyimpan informasi tentang *partitions layout* untuk mengetahui *partition layout* pada perangkat ponsel android yaitu menggunakan perintah "cat", perintah "cat" berguna untuk menampilkan informasi yang terdapat dalam partisi. Dan pada

partisi "/proc/partitions" adalah partisi yang menyimpan informasi tentang penamaan partisi media penyimpanan, maka digunakanlah perintah "cat /proc/partitions" yang berfungsi untuk melihat semua partisi yang digunakan dalam penyimpanan perangkat ponsel android yang dapat dilihat pada Gambar 4.34.



Gambar 4.32 Hasil analisis partition layout perangkat android

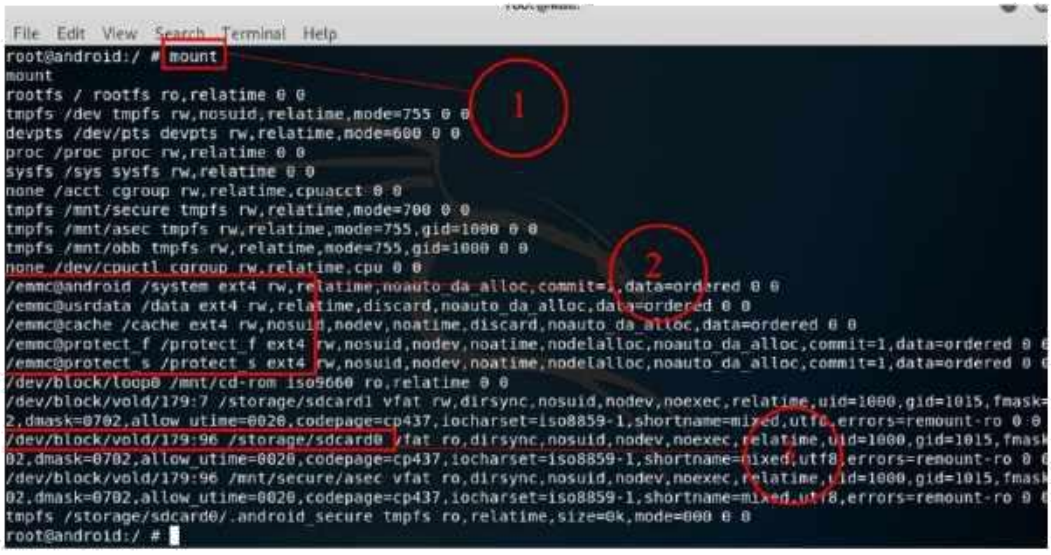
Pada Gambar 4.34 menjelaskan perintah (1)"adb shell su" yang digunakan untuk mengakses shell pada perangkat android dengan hak akses *superuser* , kemudian perintah (2)"cat /proc/partitions" yang berfungsi untuk menampilkan list media penyimpanan pada perangkat android dan pada (3) adalah list penamaan dan ukuran sebuah partisi media penyimpanan perangkat ponsel android. Pada list penamaan partisi media penyimpanan terdapat partisi "mmclk0" partisi tersebut adalah partisi media penyimpanan internal pada sebuah perangkat ponsel android, sedangkan untuk partisi media penyimpanan eksternal android ditunjuk dengan nama "mmcblk1".

c. Analisis *filesystem* perangkat ponsel android

Tahap ini file system pada perangkat ponsel android dianalisis yang bertujuan untuk mengetahui file system yang digunakan media penyimpanan data dalam ponsel android Sony Experia yang telah diketahui nama partisi yang dapat melihat semua filesystem dengan menggunakan perintah "cat" dalam sebuah partisi yang bernama "/proc/filesystems" yang ada dalam perangkat ponsel android. Untuk melihat hasil perintah untuk melihat filesystem dapat dilihat pada Gambar 4.35.

Gambar 4.33 Hasil analisis file sistem pada perangkat android

Pada Gambar 4.35 merupakan semua *filesystem* yang digunakan dalam perangkat ponsel android. Pada (1)"adb shell su" adalah perintah untuk memasukan shell pada perangkat ponsel android dengan hak akses *superuser*. Pada (2)"cat /proc/filesystems" adalah perintah yang berfungsi untuk melihat semua *file system* yang digunakan pada perangkat ponsel android, (3)"nodev" adalah *Filesystem* yang memiliki entri "Nodev" yang merujuk pada *filesystem* tersebut tidak dapat di*mounting*. Sedangkan untuk (4) *file system* yang digunakan pada media penyimpanan data internal maupun eksternal yang tidak memiliki entri "nodev" yaitu "ext2,3,4" dan "vfat, msdos, iso9660 dan fuseblk" dan untuk melihat rinciannya dapat dilakukan dengan perintah "mount" dan dapat dilihat pada Gambar 4.36.

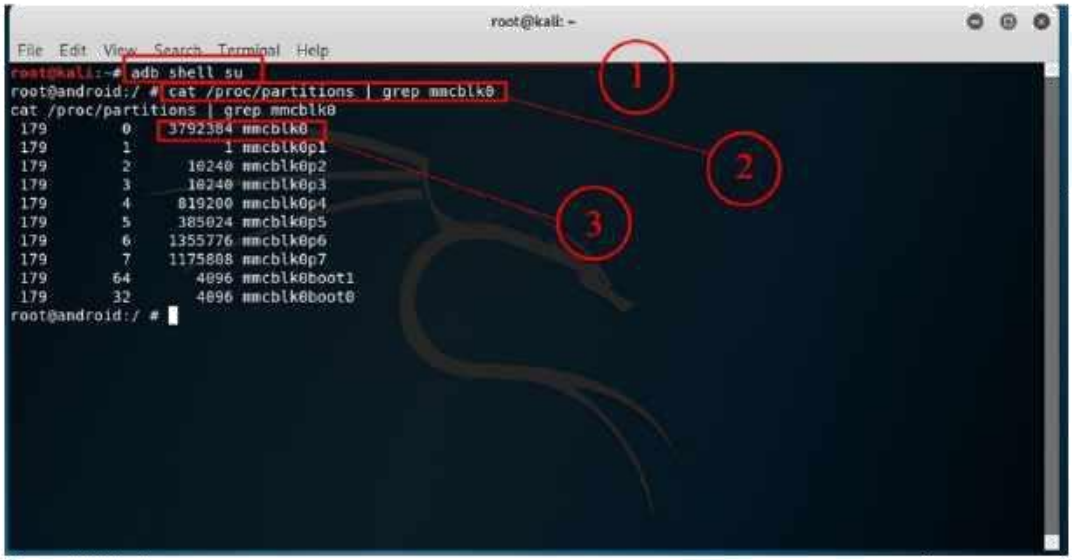


Gambar 4.34 Hasil mounting file system pada perangkat android

Pada Gambar 4.36 merupakan rincian *file system* yang digunakan dalam media penyimpanan internal maupun eksternal pada perangkat ponsel android. Dapat dilihat pada petunjuk (1)"mount" merupakan perintah untuk melihat *file system* yang dapat di*mount*. Dan pada petunjuk (2) adalah daftar partisi "mmcblk0" yang merujuk pada media penyimpanan internal ponsel android menggunakan *filesystem* "ext4", dari ifile system tersebut menunjukan bahwa media penyimpanan internal ponsel android dapat dilakukan proses akuisisi, dan petunjuk (4) adalah partisi "mmcbl1" yang merujuk ke media penyimpanan eksternal ponsel android dengan menggunakan *file system* "vfat", dari *file system* "vfat" menunjukan bahwa media penyimpanan eksternal ponsel android juga dapat dilakukan proses akuisisi.

d. Analisis ukuran media penyimpanan perangkat ponsel android

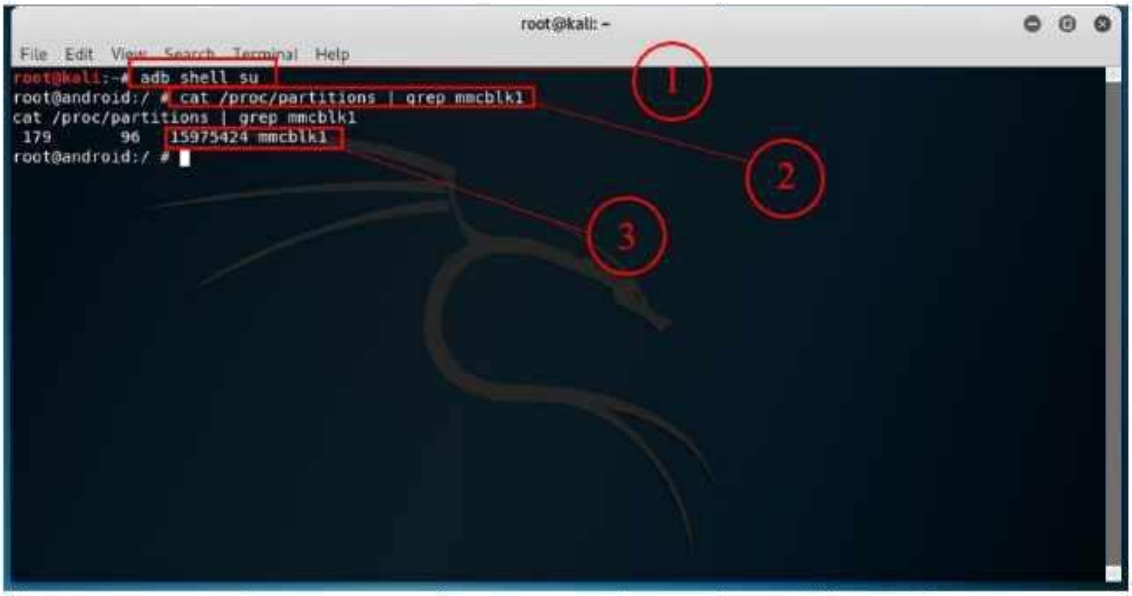
Tahap ini merupakan analisis untuk ukuran media penyimpanan ponsel android baik itu media penyimpanan internal maupun media penyimpanan eksternal yang bertujuan untuk kepentingan membuat *acquisition progress* pada tampilan 2 aplikasi dan dapat mengetahui perkembangan sejauh mana proses akuisisi telah berjalan di dalam aplikasi secara *real-time*. Pertama yang penulis lakukan adalah menganalisa ukuran media penyimpanan internal ponsel android yang merujuk pada partisi "mmcblk0". Untuk melihat hasil analisis ukuran penyimpanan internal ponsel android dapat dilihat pada Gambar 4.37.



Gambar 4.35 Hasil analisis ukuran media penyimpanan ponsel android

Pada Gambar 4.37 merupakan hasil analisis ukuran media penyimpanan internal ponsel android, dan menjelaskan bahwa pada perintah (1)"adb shell su" digunakan untuk mengakses *shell* pada perangkat ponsel android dengan hak akses *superuser*, setelah masuk kedalam *shell*

perangkat ponsel android kemudian penulis melakukan perintah (2)"cat /proc/partitions | grep mmcblk0" yang berfungsi untuk menampilkan semua list partisi media penyimpanan internal ponsel android yang hanya merujuk kepada "mmcblk0" dan pada petunjuk (3)"3792384" adalah ukuran utama media penyimpanan internal pada ponsel android yang dijadikan sebagai *set maximum "acquisition progress Int"* dalam tampilan 2 aplikasi. sedangkan untuk menganalisis ukuran media penyimpanan eksternal yang merujuk pada "mmcblk1" dapat dilihat pada Gambar 4.38.



Gambar 4.36 Hasil analisis ukuran media penyimpanan ponsel android

Pada Gambar 4.38 merupakan hasil analisis ukuran media penyimpanan internal ponsel android, dan dapat dijelaskan pada perintah(1)"adb shell su" digunakan untuk mengakses *shell* pada perangkat ponsel android dengan hak akses *superuser*, setelah masuk kedalam *shell* perangkat ponsel android kemudian penulis melakukan perintah (2)"cat /proc/partitions | grep mmcblk1" yang digunakan untuk menampilkan semua partisi pada media penyimpanan android yang bernama "mmcblk1" dan pada (3) "15975424" adalah ukuran media penyimpanan eksternal pada perangkat ponsel android yang dijadikan set maximum "acquisition progress ext" dalam tampilan 2 aplikasi.

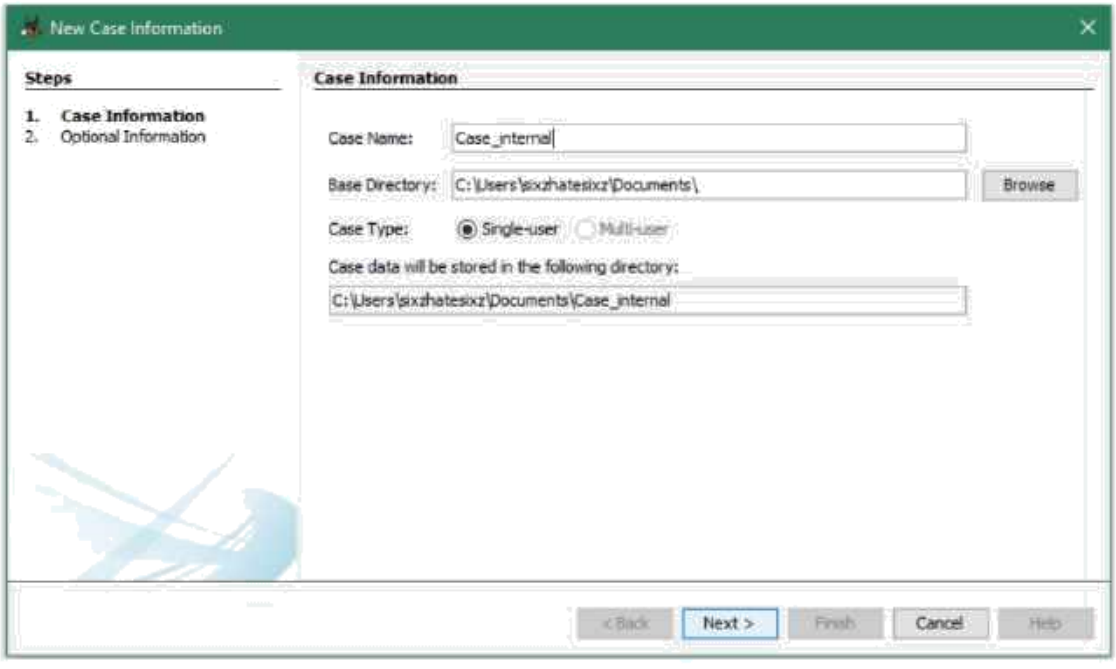
Dari semua tahapan proses analisis model media penyimpanan data pada perangkat ponsel android yang dilakukan penulis, diperoleh data yang dapat digunakan oleh penulis sebagai rujukan untuk pengembangan aplikasi *Forensics Imaging* perangkat ponsel android dengan memanfaatkan *root*. data tersebut diperoleh dari beberapa proses analisis di atas, dari data tersebut penulis dapat mengidentifikasi partisi utama dari media penyimpanan perangkat

ponsel android, *file system* yang digunakan pada perangkat ponsel android dan juga ukuran media penyimpanan yang berfungsi untuk membuat *progress bar* di dalam aplikasi.data-data yang didapatkan adalah sebagai berikut :

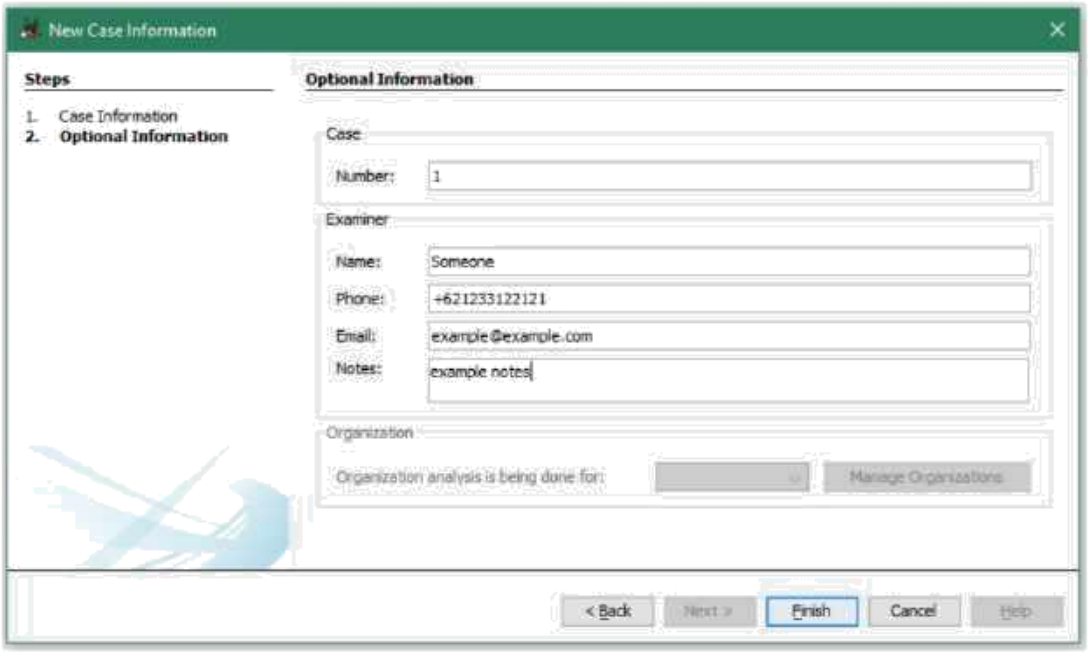
1. Dari kedua perangkat ponsel android yang sudah dilakukan analisis tersebut menunjukan bahwa partisi utama yang merujuk kepada media penyimpanan internal maupun eksternal adalah sama. Yaitu partisi"/dev/block/mmcblk0" yang merujuk ke sebuah media penyimpanan internal dalam perangkat ponsel android, sedangkan untuk media penyimpanan eksternal dapat ditunjukan dengan partisi"/dev/block/mmcblk1".
2. *File system* yang digunakan pada penyimpanan internal adalah EXT4 sedangkan untuk eksternal adalah VFAT
3. ukuran media penyimpanan internal dan eksternal akan dijadikan set maksimum"*progress acquisition*" dalam tampilan 2 aplikasi untuk perkembangan proses akuisisi yang sudah dijalankan.

4.4.2 Analisis Data *File Image* yang Dihasilkan Aplikasi

Tahap ini untuk menganalisis *file image* yang dikeluarkan oleh aplikasi dengan menggunakan *tools* Autopsy dalam sistem operasi Windows. Saat membuka aplikasi *Autopsy* pertama yang dilakukan mengisikan beberapa *form* beberapa diantaranya adalah *case info* untuk penamaan dan tata letak kasus yang di analisis dan *Additional information* untuk memberi urutan kasus dan siapa nama pengguna yang melakukan analisis. Untuk melihat kedua *form* dapat dilihat pada Gambar 4.39 dan Gambar 4.40.



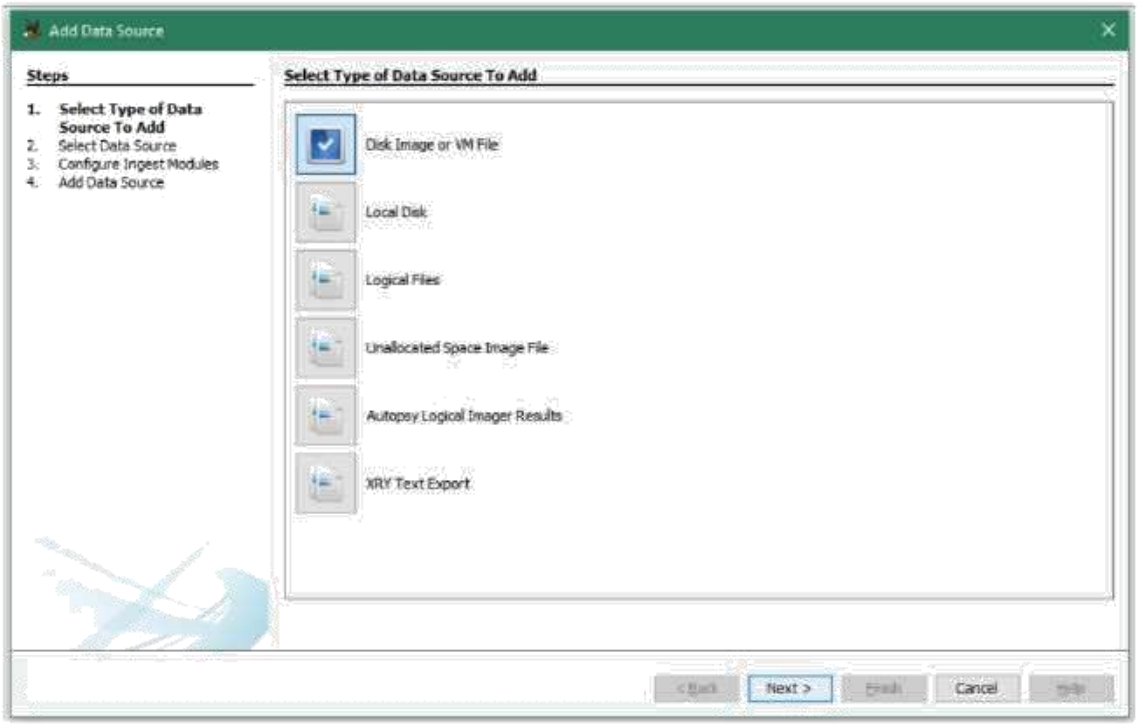
Gambar 4.37 Form case pada aplikasi autopsy



Gambar 4.38 form additional information pada aplikasi autopsy

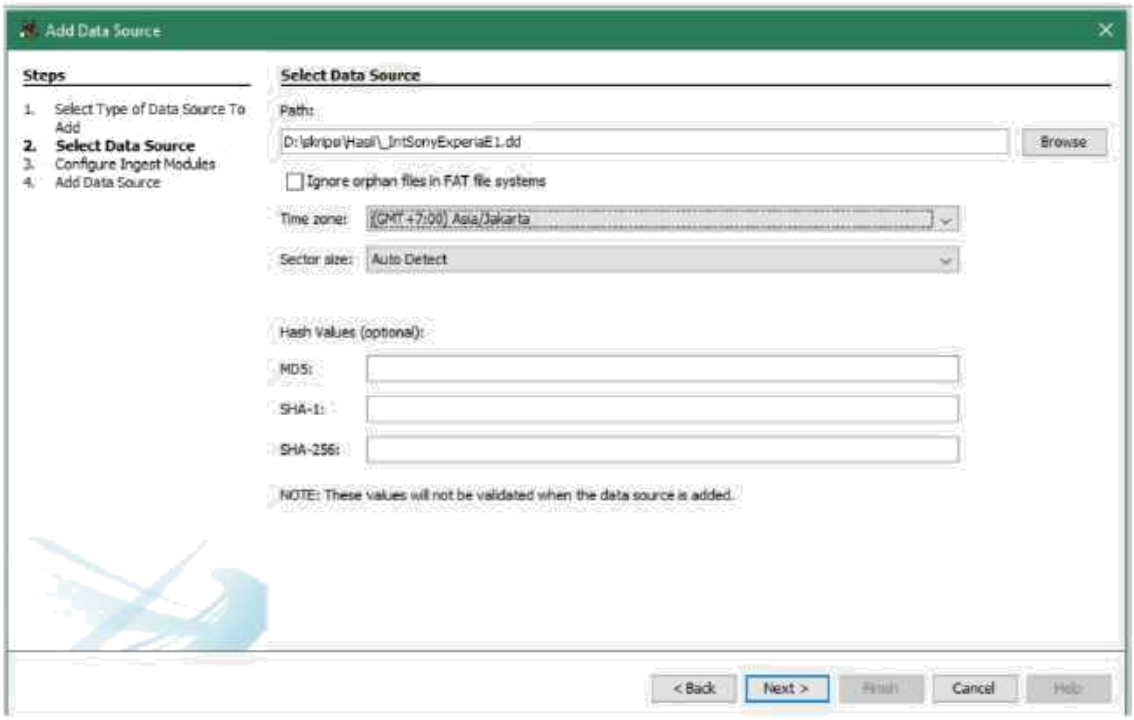
Pada Gambar 4.39 merupakan *form case information* pada aplikasi *autopsy* di sistem operasi windows. Pada form tersebut berisikan mengenai kasus yang ingin di analisis mulai dari mengisikan nama kasus, pemilihan letak direktori untuk menyimpan kasus tersebut yang akan mengeluarkan sebuah dokumen. Sedangkan pada Gambar 4.40 merupakan *form additional information* yang berisikan mengenai penomoran kasus, nama pengguna aplikasi autopsy yang melakukan analisis, nomortelepon pengguna aplikasi, email dan catatan kasus. Dari kedua form tersebut berfungsi untuk mendapatkan hasil rekam analisis kasus yang dapat dipertanggung jawabkan pada sidang.

Setelah tahapan pengisian *form* untuk kepentingan analisis kasus, maka langkah berikutnya masuk kedalam tahap-tahap untuk melakukan proses analisis dari *file image* yang dihasilkan aplikasi *Forensic imaging* pada ponsel android dengan memanfaatkan *root.* tahap tersebut diantaranya adalah menentukan jenis data yang akan dianalisis berupa *file image*. Mengambil *file image* yang akan dianalisis hasil dari aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *root*. menentukan *ingest modules* dalam aplikasi autopsy yang berfungsi untuk menganalisis data, dan data *source* yang nantinya akan ditampilkan pada aplikasi autopsy yang sudah teridentifikasi data-datanya. Untuk melihat tahapan tersebut dapat dilihat pada Gambar 4.41.



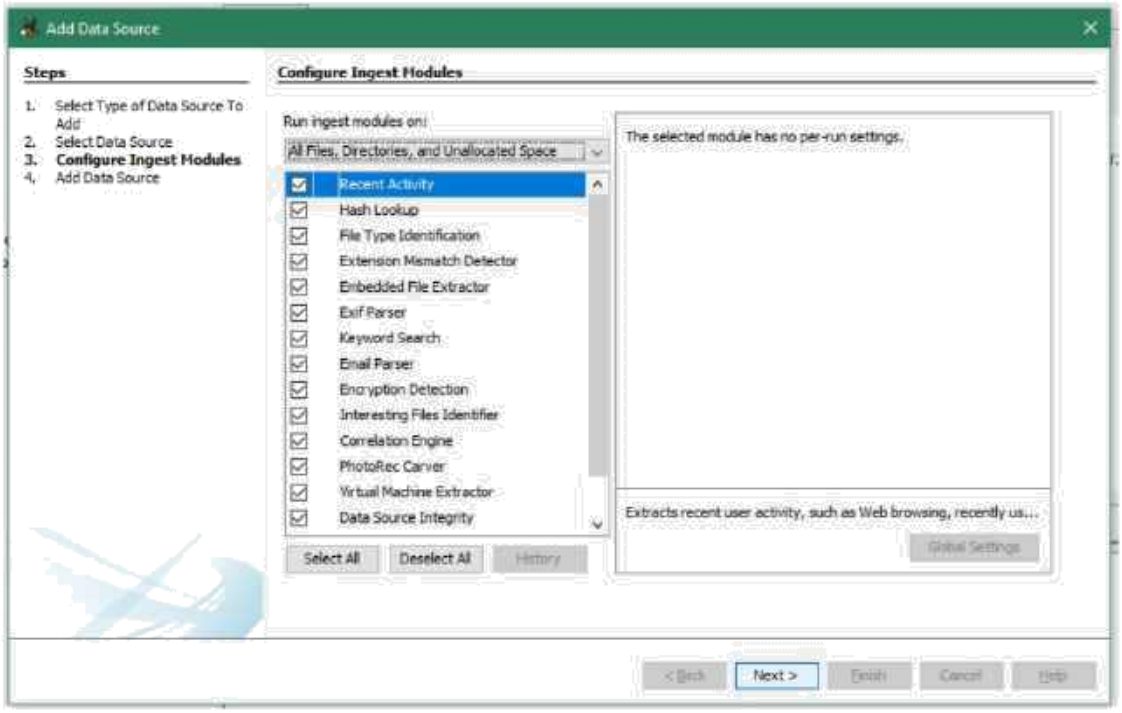
Gambar 4.39 form untuk menentukan tipe data

Pada Gambar 4.41 adalah *form* dalam aplikasi autopsy untuk memilih tipe data file yang akan dianalisis, untuk file yang akan penulis analisis adalah *file image* yang dihasilkan aplikasi *Forensic imaging* pada ponsel android dengan memanfaatkan *root*, maka dari itu penulis memilih pilihan 1 yaitu *disk image* karena file yang dihasilkan aplikasi yang penulis telah buat bertipekan *disk image* dan pilihan 1 dapat untuk menganalisis file *image* yang telah dihasilkan aplikasi yang telah penulis buat secara menyeluruh. Untuk pilihan 2 dan 3 adalah pilihan untuk menganalisis sebuah file dari *localdisk* dan *logical files*, pilihan 4 adalah untuk menganalisis sebuah *file image* yang terbatas dalam sector *unallocated space* dari *file image* yang akan dianalisis, lalu ada pilihan 5 yaitu *Autopsy logical imager result*, pilihan ini berfungsi untuk menambahkan hasil koleksi dari *logical imager*. Sedangkan pilihan terakhir adalah XRY Text Export yang berfungsi untuk menambahkan *data source* berupa banyak file yang berextensikan txt dalam sebuah folder untuk dianalisis. setelah memilih pilihan 1 selanjutnya adalah *form* untuk mengambil *file image* yang dihasilkan aplikasi untuk dimasukan kedalam aplikasi autopsy dan dapat dilihat pada Gambar 4.42.



Gambar 4.40 Form pengambilan file image pada aplikasi autopsy

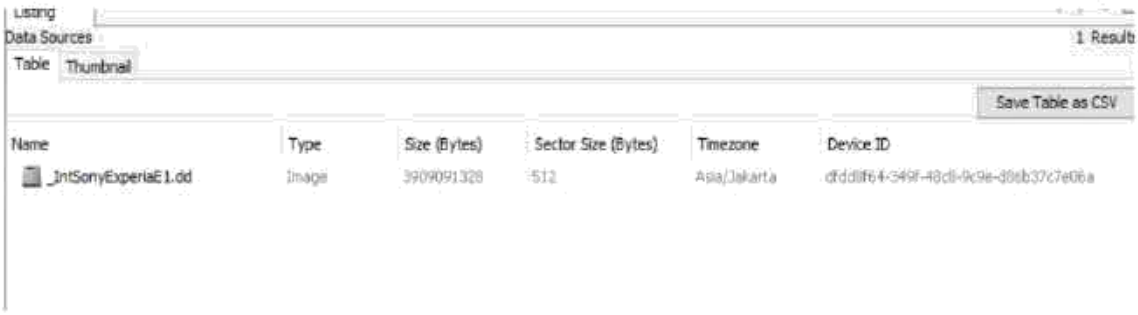
Gambar 4.42 adalah sebuah *form* untuk mengambil sebuah data *file image* yang dihasilkan aplikasi yang nantinya akan dimasukan dalam aplikasi autopsy untuk dilakukannya analisis dari data tersebut, setelah data sudah dipastikan masuk dan terbaca oleh aplikasi autopsy maka selanjutnya aplikasi akan menampilkan *form ingest modules* dan seorang yang sedang menganalisis akan mengisikan form tersebut untuk menentukan *form ingest modules* yang dibutuhkan dalam keperluan analisis file di aplikasi autopsy dan dapat dilihat pada Gambar 4.43.



Gambar 4. 41 Form penentuan ingest modules pada aplikasi autopsy

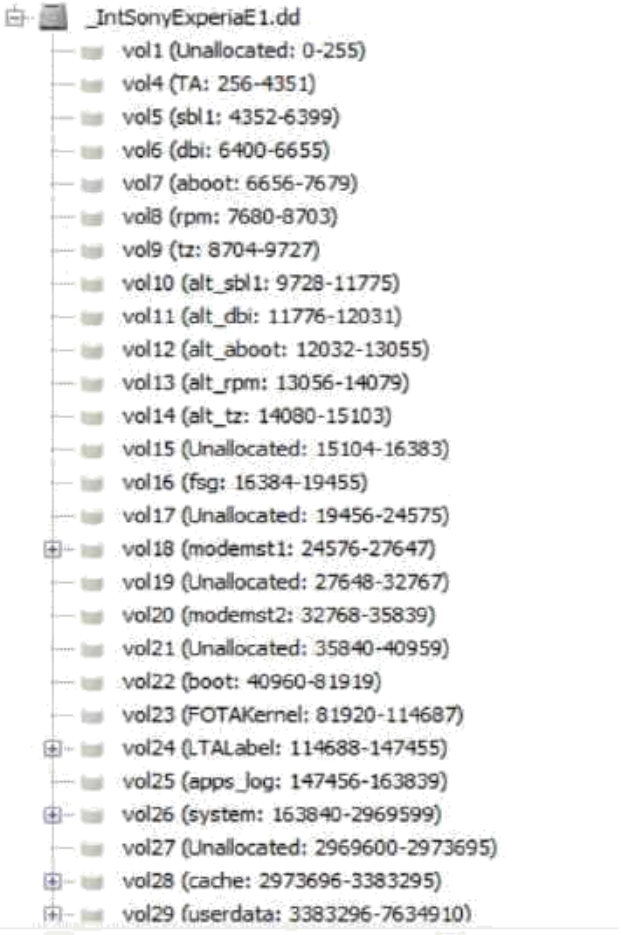
Pada Gambar 4.43 merupakan *form* untuk memilih *ingest modules* dalam aplikasi

autopsy yang memiliki fungsi untuk membagi data yang dibutuhkan dalam melakukan proses analisis dari *file image* yang sudah dimasukan dalam aplikasi autopsy yang bertujuan untuk menyusun hasil analisis agar rapi dan dapat dianalisis secara menyeluruh dalam aplikasi autopsy, dari *ingest modules* dalam aplikasi autopsy dengan banyak pilihan yang memiliki fungsi untuk kepentingan analisis agar mudah dilakukan. Setelah menentukan *ingest modules,* maka aplikasi autopsy akan menerapkan fungsi *ingest modules* yang telah dipilih dan akan masuk dalam menu utama untuk melakukan analisis dari *file image* yang telah dimasukan dalam aplikasi autopsy. Untuk melihat data *file image* penyimpanan internal ponsel android yang sudah terdeteksi oleh aplikasi autopsy dan menjadi *data source* guna untuk keperluan proses analisis yang terbagi dari beberapa komponen data dari fungsi *ingest modules* yang sudah dijalankan oleh aplikasi autopsy dapat dilihat pada Gambar 4.44.



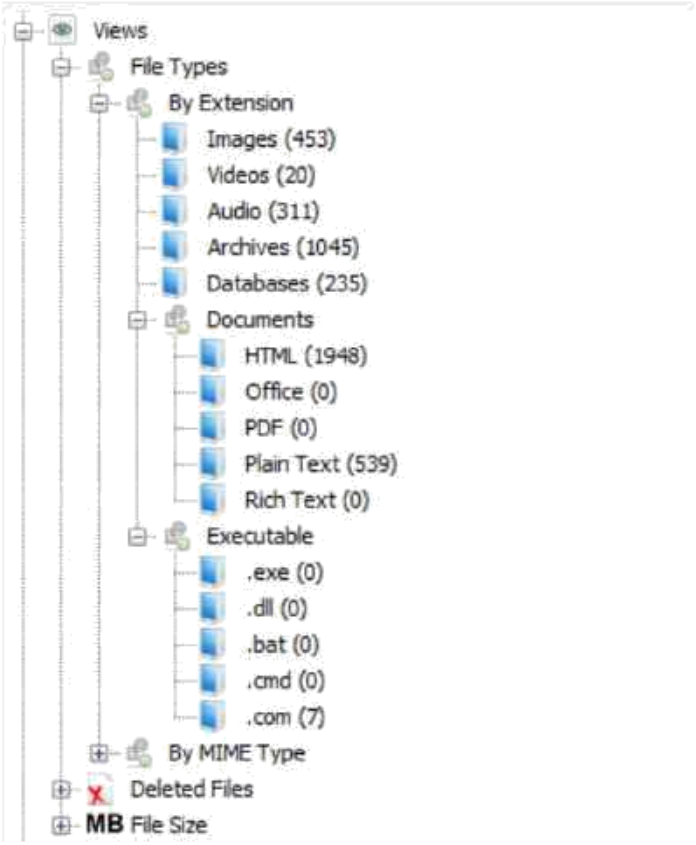
Gambar 4.42 Data source pada aplikasi autopsy

Pada Gambar 4.44 merupakan data *source* yang sudah terdeteksi dan dihasilkan oleh aplikasiautopsy dari *file image* yang sudah dimasukan. Untuk menjalankan proses analisis dari *file image* tersebut, namun setelah masuk dan terdeteksi sebagai data *source* dalam aplikasi autopsy, selanjutnya aplikasi autopsy akan menjalankan proses untuk membaca semua isi dari *file image* agar terangkai dengan baik dan mudah untuk dilakukannya proses analisis.untuk melihat partisi dalam *file image* pada media penyimpanan internal ponsel android yang telah menjadi data *source* yang dihasilkan autopsy dapat dilihat pada Gambar 4.45.



Gambar 4.43 Daftar partisi pada data source dalam aplikasi autopsy

Pada Gambar 4.45 merupakan daftar partisi dari *file image* media penyimpanan internal perangkat ponsel android yang sudah dijadikan data *source* di dalam aplikasi autopsy.dari daftar partisi tersebut aplikasi autopsy dapat melihat semua data-data yang ada di dalam partisi tertentu yang dihasilkan dari data *source* dalam aplikasi autopsy dan dapat juga di ekstrak ke dalam perangkat laptop yang akan dijadikan file barang bukti digital yang dihasilkan dari proses analisis. Dari semua daftar partisi yang ada dalam *file image* media penyimpanan internal pada perangkat ponsel android . ada beberapa partisi utama yang dijadikan bahan analisis oleh seseorang yang ahli dalam forensika digital, partisi tersebut adalah partisi "system" pada vol 26 yang bekerja untuk melihat data sistem yang digunakan pada media penyimpanan internal ponsel android, yang kedua adalah partisi "userdata" pada vol 29 yang bermanfaat untuk melihat semua data-data yang tersimpan pada media penyimpanan internal android. Setelah melakukan analisis dari kedua partisi utama dalam media penyimpanan internal ponsel android yang dihasilkan oleh data *source* pada aplikasi autopsy, selanjutnya aplikasi juga memberikan alat untuk memeriksa semua data-data yang memiliki *file type*. Untuk melihat semua data berdasarkan *file typenya* dan telah terstruktur dengan rapi dapat dilihat pada Gambar 4.46.

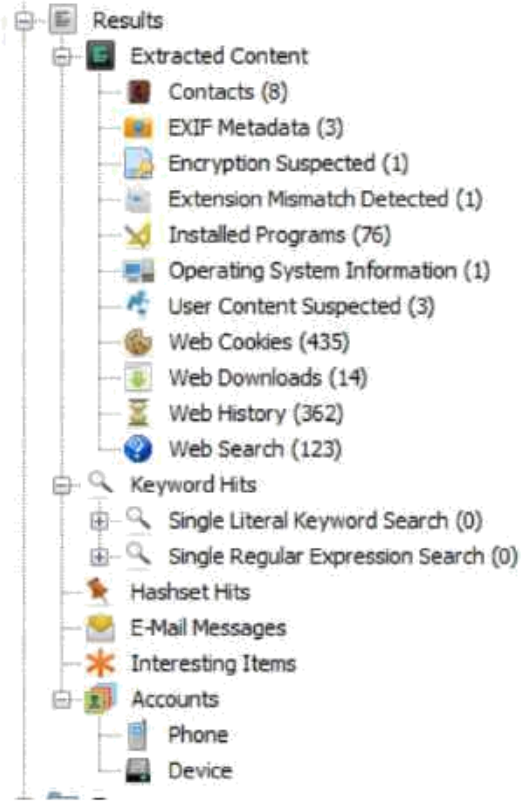


Gambar 4.44 Detail data souce pada aplikasi autopsy

Pada Gambar 4.46 merupakan fitur dari autopsy untuk melihat semua data-data yang ada dalam media penyimpanan internal perangkat ponsel android yang telah terstruktur dengan rapi dan telah menjadi data source di dalam aplikasi autopsy, dari detail data source tersebut dibagi menjadi beberapa komponen data dari media penyimpanan internal perangkat ponsel android, data tersebut adalah berdasarkan dari file type, delete files dan file size yang berukuran cukup besar.

Data yang berdasarkan file types dibagi menjadi 2 bagian, bagian data yang pertama adalah data yang bersumber dari extension yang berguna untuk melihat semua data-data yang mempunyai format seperti gambar, video , audio dan dokumen yang diperoleh dari proses analisis dan juga di ekstrak ke dalam perangkat laptop untuk dijadikan barang bukti digital dari hasil proses analisis,dan untuk tipe data MIME type adalah data yang berisikan seperti aplikasi, aplikasi pesan dan lain-lain.

Data berdasarkan deleted file berguna untuk melihat data-data yang sudah terhapus sebelumnya pada media penyimpanan internal perangkat ponsel android yang bersumber dari data file system dan juga data-data lainnya yang sudah dihapus oleh pengguna ponsel android. Selanjutnya data yang berdasarkan file size yang berguna untuk melihat data yang terdapat pada media penyimpanan internal perangkat ponsel android yang berukuran cukup besar mulai dari ukuran 50-200Mb, 200M- 1GB dan 1GB lebih . setelah melihat data-data yang ada pada perangkat ponsel android selanjutnya autopsy juga memberikan fungsi untuk melihat isi dari semua konten dalam ponsel android yang dapat dilihat pada Gambar 4.47.



Gambar 4. 45 Isi konten pada data source dalam aplikasi autopsy

Pada Gambar 4.47 adalah hasil dari konten yang terdapat pada perangkat ponsel android, seperti histori panggilan, web cookies, web search, web history dan web download di dalam web browser perangkat ponsel android dari data tersebut diperoleh hasil dari jejak rekam penggunaan perangkat ponsel android yang dapat diekstrak dan dijadikan barang bukti digital berupa dokumen dan konten lainnya dari penggunaan email yang berisikan isi email.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian dan pembahasan tugas akhir mengenai Pembuatan Aplikasi dengan metode Live Imaging Acqusition untuk eksplorasi barang bukti digital pada media penyimpanan ponsel dapat disimpulkan bahwa hasil implementasi pengembangan aplikasi sesuai dengan perancangan yang telah dibuat dan semua fungsionalitas sistem berjalan dengan baik, sebagai berikut :

a. Aplikasi menampilkan progress bar untuk mengetahui perkembangan berjalannya proses akusisi media penyimpanan secara real-time dan juga dapat mengakuisisi media penyimpanan internal maupun eksternal baik itu salah satu penyimpanan maupun kedua penyimpanan secara bersamaan pada saat kondisi ponsel android dalam keadaan menyala.
b. Aplikasi terbukti dapat menjaga integritas data dari file yang dihasilkan, mendapatkan kecepatan transfer data lebih dari 6 MB/detik saat mengakuisisi baik itu kedua penyimpanan maupun salah satu media penyimpanan.

5.2 Saran

Untuk pengembangan lebih lanjut, diharapkan Aplikasi *Forensic Imaging* pada ponsel Android dengan memanfaatkan *Root* kedepannya dapat membuat fitur untuk dapat melakukan akuisisi ponsel android dalam keadaan terkunci jika barang bukti berupa ponsel android yang didapatkan oleh penyelidik dalam keadaan terkunci. Selain itu perlu ditambahkan beberapa fitur seperti memilih perangkat ponsel android lebih dari 1 yang terbaca oleh aplikasi untuk melakukan proses akuisisi sehingga dapat lebih mudah dan sesuai jika barang bukti berupa ponsel android yang cukup banyak.

DAFTAR PUSTAKA

forensicwiki.org. (n.d). Forensic Imaging and their Formats - DD (raw). Diambil kembali dari https://www.securitynik.com/2015/06/forensic-imaging-and-their-formats-dd.html

Garfikel, S., & Ph, D. (2009). . Automating Disk Forensic Processing with SleuthKit, XML and Python.- National Technical Reports Library V3.0. Diambil kembali dari https://ntrlr3-ntisgov.spot.lib.auburn.edu/view.php?pid=NTIS:ADA549270%5Cnfile:///C:/DOCUME~1/ WOHRLA~1.AUB/LOCALS~1/Temp/ADA549270.pdf

Kaufman. (1972). Educational system planning . *Educational system planning* .

Lexy, M. (202). Metodologi Penelitian Kualitatif. Bandung PT. remaja Rosdakarya.

Murphy. (2012). Developing Process for Mobile Device Forensic. 1-9. Diambil kembali dari http://www.mobileforensicscentral.com/mfc/documents/Mobile Device Forensic Process v3.0.pdf

Ruby, A. (2013). On Digital Forensics. Diambil kembali dari http://ondigitalforensics.weebly.com/digital-forensics/comparison-defenisi-digitalforensics-forensika-digital-atau-yang-biasa-disebut-forensik-computer#.W-WS_dUzbIU

Shankland. (2007). Google's Android parts ways with Java industry group.

techtarget.com. (n.d.). What is disk cloning? Diambil kembali dari https://whatis.techtarget.com/definition/disk-cloning