ANALISIS DAN PERANCANGAN APLIKASI *FORENSIC IMAGING* PADA PONSEL ANDROID DENGAN MEMANFAATKAN *CUSTOM RECOVERY*



Disusun Oleh:

N a m a : Waldy Nur Hamzah

NIM : 13523267

PROGRAM STUDI TEKNIK INFORMATIKA-PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

HALAMAN PENGESAHAN DOSEN PEMBIMBING

ANALISIS DAN PERANCANGAN APLIKASI *FORENSIC IMAGING* PADA PONSEL ANDROID DENGAN MEMANFAATKAN *CUSTOM RECOVERY*

TUGAS AKHIR

Disusun Oleh:

N a m a	**: Waldy Nur Hamzah**
NIM	**: 13523267**

Yogyakarta, 21 Oktober 2018

Pembimbing,

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

HALAMAN PENGESAHAN DOSEN PENGUJI

ANALISIS DAN PERANCANGAN APLIKASI *FORENSIC IMAGING* PADA PONSEL ANDROID DENGAN MEMANFAATKAN *CUSTOM RECOVERY*

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Informatika
di Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, 3 Januari 2019

Tim Penguji

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

Anggota 1

Erika Ramadhani, S.T., M.Eng.

Anggota 2

Chanifah Indah Ratnasari, S.Kom,. M.Kom.

Mengetahui,
Ketua Program Studi Teknik 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 : Waldy Nur Hamzah

NIM : 13523267

Tugas akhir dengan judul:

ANALISIS DAN PERANCANGAN APLIKASI *FORENSIC IMAGING* PADA PONSEL ANDROID DENGAN MEMANFAATKAN *CUSTOM RECOVERY*

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, 7 Desember 2018



(Waldy Nur Hamzah)

HALAMAN PERSEMBAHAN

Bismillahirrohmanirrohhim, dengan rahmah Allah yang maha pengasih lagi maha penyayang. Tugas akhir ini peneliti persembahkan kepada kedua orang tua, bapak Usman dan ibu Sri Sudarsih yang selalu memberikan dukungan dan mendoakan untuk menyelesaikan tugas akhir dalam jenjang kuliah. Kepada saudara-saudari kandung yang selalu memberikan semangat dalam mengerjakan tugas akhir. Kepada teman-teman peneliti yang senantiasa memberikan bantuan, kesenangan ketika mendapatkan kendala dalam mengerjakan tugas akhir. Maka dari itu tanpa adanya dukungan dan doa orang tua, saudara-saudari, dan temen-teman peneliti, penelti tidak dapat menyelesaikan tugas akhir ini.

HALAMAN MOTO

"*Tidak ada balasan untuk kebaikan selain kebaikan(pula)*" **(Q.s. Ar-Rahman, 30)**

"*Allah akan meninggikan orang-orang yang beriman diantaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat*" **(Q.s. Al-Mujadalah, 11)**

"*Tuhanku, tambahkanlah ilmu pengtahuan kepadaku*" **(Q.s. Ta-Ha, 114)**

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah, 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 jaman terang benderang seperti sekarang ini.

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

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 wawasan serta pengalaman dalam menerapkan keilmuan, sesuai dengan yang diambil di bangku perkuliahan.

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

1. Kedua orang tua, kakak dan adik atas segala doa dan dukungan selama penulis untuk menyelesaikan Tugas Akhir.
2. Bapak Fathul Wahid, S.T., M.Sc., Ph.D. selaku Rektor Universitas Islam Indonesia
3. Bapak Hari Purnomo, Prof., Dr., Ir., M.T. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Bapak Hendrik, S.T., M.Eng. selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc selaku Ketua Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
6. Ibu Nur Wijayaning Rahayu, S.Kom., M.CS., selaku Sekretaris Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
7. Bapak Fietyata Yudha, S.Kom., M.Kom., selaku Dosen Pembimbing Tugas Akhir yang selalu memberikan solusi untuk menyelesaikan Tugas Akhir penulis.
8. Razan Maulida K, teman sekontrakan yang selalu ada untuk saling berdiskusi mengenai topik Tugas Akhir dan menjadi teman untuk mengerjakan Tugas Akhir.

9. Alifiriani Kusuma Ningrum selalu memberikan semangat dan menjadi teman untuk mengerjakan Tugas Akhir.
10. Teman Kontrakan. Agus , Argo, Aji, Razan, Sulistyo, dan Burhan selalu menjadi teman yang selalu memberikan kesenangan saat ada hambatan mengerjakan Tugas Akhir.
11. Teman-teman Informatika 2013 khususnya sahabat-sahabat penulis yang menjadi teman seperjuangan di bangku kuliah, sedikit banyak memberi dukungan dan mempengaruhi penulis untuk segera menyelesaikan Tugas Akhir.
12. Semua pihak yang telah banyak membantu penulis dalam pelaksanaan Tugas Akhir yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih belum sempurna karena keterbatasan kemampuan dan pengalaman penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan Laporan Tugas Akhir ini. Akhir kata, penulis berharap agar laporan ini dapat bermanfaat bagi semua pihak.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 7 Desember 2018

(Waldy Nur Hamzah)

SARI

Perangkat *mobile* berbasis Android digunakan oleh mayoritas masyarakat pengguna ponsel pintar di seluruh dunia. Penggunaan ponsel pintar sebagai alat komunikasi dan lainnya dalam keseharian yang dapat membantu pengguna menyelesaikan berbagai hal sesuai kebutuhannya, namun dari penggunaan ponsel pintar sebagai alat komunikasi dan memanfaatkannya secara positif, tentu saja ada penggunaan ponsel pintar tertentu yang memanfaatkan ponsel pintar untuk melakukan tindakan buruk yang bertentangan dengan hukum dan undang - undang ITE. Dalam proses hukum yang berlaku dan terkait dengan informasi dan transaksi elektronik yang dilakukan dalam ponsel pintar diperlukan proses *Mobile Forensic* digital untuk mendapatkan barang bukti yang sah dan dapat digunakan di pengadilan. Dalam dunia *mobile forensic* salah satu metode untuk mengamankan barang bukti digital (*digital evidence*) adalah *disk imaging* atau sering dikenal *disk clonning*. *Disk imaging* adalah suatu proses menyalin seluruh konten dalam perangkat Android yang memiliki sebuah penyimpanan yang akan dapat dianalisis sesuai dengan keadaan penyimpanan pada perangkat saat ditemukan dan mengurangi resiko berubahnya suatu struktur atau konten dari *disk image* ketika dilakukan analisis. Berdasarkan uraian masalah di atas, penelitian mengangkat judul tugas akhir "Analisis dan Perancangan Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*".

Dengan aplikasi ini diharapkan mampu melakukan akuisisi terhadap media penyimpanan ponsel android baik itu internal maupun eksternal dan juga mampu menjaga sebuah integritas data saat dilakukan proses akuisisi dengan penggunaan *Custom Recovery* TWRP dalam ponsel android. Sedangkan hasil yang diperoleh dari aplikasi berupa *file image* dan *file log* tersebut dapat dianalisis dan dapat dijadikan barang bukti digital dalam proses peradilan.

Oleh karena itu penulis membuat aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *Custom Recovery* yang bertujuan untuk mengamankan barang bukti digital berupa *file image* yang dapat dianalisis secara menyeluruh dengan bantuan aplikasi Autopsy dan menjaga integritas *file image* yang dihasilkan untuk kepentingan peradilan. Kemudian melakukan pengujian berupa kuesioner melalui *google form* yang disebarkan kepada responden untuk menanggapi dari aplikasi yang dihasilkan dengan menggunakan perhitungan skala linkert.

Kata kunci: *imaging android, Forensic Imaging android, application imaging, disk clonning*

GLOSARIUM

Mobile Forensics	Forensika Digital pada perangkat bergerak.
Forensic Imaging	Penggandaan media penyimpanan
Disk Imaging	Penggandaan media penyimpanan ponsel android
Storage	Media peyimpanan data digital
Disk Block	Bagian terkecil dari lokasi pengelompokan data pada perangkat keras media peyimpanan
Filesystem	system file yang digunakan dalam ponsel android
Hashing md5	pencocokan md5 file
Recovery mode	mode pemulihan dalam ponsel android
Download mode	mode unduh dalam ponsel android
Bootloader mode	mode fastboot dalam ponsel android

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI..... ix
GLOSARIUM x
DAFTAR ISI xi
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan 4
1.5 Manfaat Penelitian 4
1.6 Metode Penelitian 5
1.7 Sistematika Penulisan 6
BAB II LANDASAN TEORI 7
2.1 Penelitian Sebelumnya 7
2.2 Analisis dan Perancangan 10
2.2.1 Pengertian Analisis 10
2.2.2 Analisis Data 10
2.2.3 Pengertian Perancangan 11
2.3 *Digital Forensic* 11
2.3.1 Pengertian *Digital Forensic* 11
2.3.2 *Mobile Forensics* 11
2.3.3 *Mobile forensics* pada perangkat Android 12
2.4 *Disk Imaging* 14
2.5 Bahasa Pemograman *Python* 14
2.6 *Custom Recovery* 14
2.7 Analisis Model Penyimpanan Data pada Ponsel Android 15
BAB III METODOLOGI PENELITIAN 17
3.1 Alur Penelitian 17
3.2 Analisis Kebutuhan Sistem 18
3.2.1 Analisis Kebutuhan Perangkat Keras 18
3.2.2 Analisis Kebutuhan Perangkat Lunak 19
3.2.3 Analisis Kebutuhan Masukan 21
3.2.4 Analisis Kebutuhan Proses 21

3.2.5 Analisis Kebutuhan Keluaran 22
3.3 Perancangan Aplikasi 22
3.3.1 *Flowchart* 23
3.3.2 *Pseudocode* 24
3.3.3 Desain Antarmuka 31
3.4 Pengujian Kuesioner Skala Linkert 33
BAB IV HASIL DAN PEMBAHASAN 38
4.1 Hasil 38
4.1.1 Hasil Pembuatan Aplikasi 38
4.1.2 Hasil Proses Akuisisi Dari Aplikasi 40
4.1.3 Hasil Pengujian Kuesioner Skala Linkert Aplikasi 43
4.1.4 Hasil Pengujian Aplikasi oleh Pakar 44
4.2 Pembahasan Implementasi 45
4.2.1 *Custom Recovery* 45
4.2.2 Implementasi Antarmuka 46
4.2.3 Implementasi Kode Program Proses *Scanning* Ponsel Android 50
4.2.4 Implementasi Kode Program Proses Mengkoneksikan Ponsel Android 51
4.2.5 Implementasi Kode Program Proses Akuisisi dan *Hashing Md5* 53
4.3 Proses Pengujian Aplikasi 59
4.3.1 Pengujian Black Box 59
4.3.2 Pengujian Integritas Data *file image* dari Aplikasi 68
4.3.3 Pengujian Performa Aplikasi 70
4.3.4 Total Skor Kuesioner Skala Linkert Aplikasi 75
4.4 Proses Analisis 77
4.4.1 Analisis Model Penyimpanan Data pada Ponsel Android 77
4.4.2 Analisis Data *File Image* yang Dihasilkan Aplikasi 83
BAB V KESIMPULAN DAN SARAN 91
5.1 Kesimpulan 91
5.2 Saran 91
DAFTAR PUSTAKA 92
LAMPIRAN 94

DAFTAR TABEL

Tabel 2.1 Tabel Penelitian Lainnya 8
Tabel 3.1 Tabel Modul Tampilan 1 24
Tabel 3.2 Tabel Modul Tampilan 2 27
Tabel 3.3 Nilai Skala Jawaban 35
Tabel 3.4 Rentang Skala Linkert 35
Tabel 3.5 Uraian Kuesioner 36
Tabel 4.1 Hasil Pengujian Kuesioner Skala Linkert 43
Tabel 4.2 Pertanyaan Kuesioner ke Pakar. 44
Tabel 4.3 Hasil Performa Mengakuisisi 2 Penyimpanan Secara Bersamaan 71
Tabel 4.4 Hasil Performa Mengakuisisi Salah Satu Penyimpanan 74
Tabel 4.5 Hasil Skor Kuesioner Skala Linkert 75

DAFTAR GAMBAR

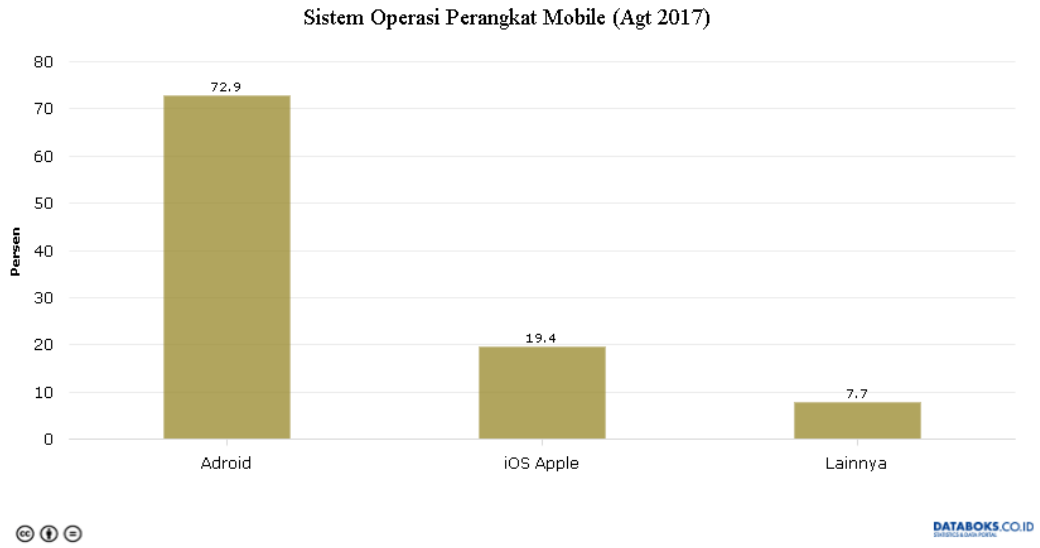
Gambar 1.1 Jumlah Pengguna Sistem Operasi Ponsel Pintar Seluruh Dunia.....1
Gambar 2.1 Ponsel Android Terhubung ke PC melalui ADB.....16
Gambar 3.1 Tahapan Alur Penelitian.....17
Gambar 3.2 *Flowchat* Aplikasi *Forensic Imaging* perangkat Android.....23
Gambar 3.3 *pseudocode* Interface.py.....26
Gambar 3.4 *pseudocode* interface2.py.....29
Gambar 3.5 Rancangan Antarmuka Tampilan 1 Aplikasi.....32
Gambar 3.6 Rancangan Antarmuka Tampilan 2 Aplikasi.....33
Gambar 4.1 Hasil Tampilan 1 Aplikasi.....38
Gambar 4.2 Hasil Tampilan 2 Aplikasi.....39
Gambar 4.3 Hasil Informasi Dalam Aplikasi.....41
Gambar 4.4 Hasil File Image.....42
Gambar 4.5 Hasil Rincian File Log.....42
Gambar 4.6 *Custom Recovery* TWRP.....46
Gambar 4.7 Antarmuka Tampilan 1 Aplikasi.....47
Gambar 4.8 Antarmuka Tampilan 2 Aplikasi.....49
Gambar 4.9 Kode Program Proses *Scanning* Perangkat Android.....50
Gambar 4.10 Hasil Proses *Scanning* Perangkat Android.....51
Gambar 4.11 Kode Program Proses Mengkoneksikan Perangkat Android.....52
Gambar 4.12 Hasil Proses Mengkoneksikan Perangkat Android.....53
Gambar 4.13 Kode Program Proses Akusisi dan *Hashing md5* Perangkat Android.....57
Gambar 4.14 Proses akuisisi Saat Berjalan.....58
Gambar 4.15 Selesai Proses Akusisi Kedua Penyimpanan.....59
Gambar 4.16 Tombol Menu Exit.....60
Gambar 4.17 Tombol Menu About.....61
Gambar 4.18 Tombol Bantuan *Flashing File Recovery*.....62
Gambar 4.19 Tombol Bantuan Jika Sudah Dalam *Custom Recovery*.....62
Gambar 4.20 Tombol Scan.....63
Gambar 4.21 Tombol Connect dan Check.....64
Gambar 4.22 Tombol Next.....64
Gambar 4.23 Tombol Select Direktori.....65
Gambar 4.24 Hasil Select Direktori.....65

Gambar 4.25 Tombol Back 66
Gambar 4.26 Tombol Extract 66
Gambar 4.27 *Error Handling* Tombol Bantuan 67
Gambar 4.28 *Error Handling* Tombol Connect 68
Gambar 4.29 *Error Handling* Tombol Extract 68
Gambar 4.30 *Md5* dari *File Image* yang Dihasilkan Aplikasi 69
Gambar 4.31 Proses Pencocokan MD5 dengan Aplikasi WinMD5 69
Gambar 4.32 Hasil Pencocokan MD5 Aplikasi WinMD5 70
Gambar 4.33 Penggunaan CPU 1 Perangkat Laptop 72
Gambar 4.34 Penggunaan CPU 2 Perangkat Laptop 72
Gambar 4.35 Penggunaan Memori Perangkat Laptop. 73
Gambar 4.36 Grafik Kecepatan Transfer Data Mengakuisisi 2 Penyimpanan. 73
Gambar 4.37 Grafik Kecepatan Transfer Data Salah Satu Penyimpanan 75
Gambar 4.38 Struktur Direktori Ponsel Android 78
Gambar 4.39 List Partisi Penyimpanan pada File “partitions” 79
Gambar 4.40 *filesystem* yang digunakan pada Perangkat Android 80
Gambar 4.41 Rincian *File system* Internal dan Eksternal. 80
Gambar 4.42 Hasil Analisis Ukuran Internal Ponsel Android. 81
Gambar 4.43 Hasil Analisis Ukuran Eksternal Ponsel Android. 82
Gambar 4.44 *Form Case Information* Aplikasi Autopsy 83
Gambar 4.45 *Form Additional Information* Aplikasi Autopsy 84
Gambar 4.46 *Form* menentukan Tipe Data yang Dianalisis 85
Gambar 4.47 *Form* Mengambil Data yang ingin Dianalisis. 85
Gambar 4.48 *Form* penentuan *ingest modules* Aplikasi Autopsy. 86
Gambar 4.49 Data *Source* dalam Aplikasi Autopsy 87
Gambar 4.50 Partisi dalam Data Source aplikasi Autopsy. 87
Gambar 4.51 Detail Data Source dari Aplikasi Autopsy 88
Gambar 4.52 Isi Konten Android dalam Data Source Aplikasi Autopsy 89

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pada saat ini perkembangan Android sebagai *mobile operating system* (OS) secara umum meningkat cukup signifikan. Sejak kisaran tahun 2006-2007 Google telah mulai mengembangkan dan memperbarui sistem operasi Android untuk menambahkam fitur baru, meningkatkan kinerjanya, dan memperbaiki berbagai bug yang ada di versi sebelumnya. Setiap versi dari Android tersebut diberi nama dengan nama makanan dan dibuat secara alphabet (berurutan sesuai abjad) dari versi 1.1 yaitu Android Apple Pie hingga versi terbaru yang dirilis pada 21 maret 2017 adalah 8.0 yaitu Android Oreo.



Gambar 1.1 Jumlah Pengguna Sistem Operasi Ponsel Pintar Seluruh Dunia
Sumber: Databoks (2017)

Pada Gambar 1.1 menunjukan grafik penggunaan sistem operasi yang ditanamkan ke dalam ponsel pintar seluruh dunia pada tahun 2017, dari data tersebut menunjukan bahwa sebanyak 72,9 persen pengguna ponsel pintar menggunakan sistem operasi Android, sedangkan 19,4 persen pengguna ponsel pintar menggunakan sistem operasi iOS Apple dan 7,7 persen menggunakan sistem operasi lainnya.(Databoks, 2017). Menurut Statista pada tahun 2017 pengguna ponsel pintar seluruh dunia mencapai lebih dari 2 miliar penguna ponsel pintar, dari data tersebut diperkirakan pengguna ponsel pintar akan terus meningkat di setiap tahunnya,

diperkirakan tahun 2019 pengguna ponsel pintar mencapai angkat 2.71 miliar sedangkan tahun 2020 diperkirakan akan mencapai 2.87 miliar pengguna ponsel pintar di seluruh dunia (Teknologi.id, 2018).

Android adalah salah satu sistem operasi (OS) yang ditanamkan pada ponsel pintar. OS Android secara umum sangat banyak digunakan di seluruh dunia. 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). Android memiliki sejumlah besar komunitas *developer* aplikasi yang memperluas fungsionalitas perangkat, dan pada bulan Oktober 2013 ada lebih dari satu juta aplikasi yang tersedia untuk Android, dan sekitar 50 miliar aplikasi telah diunduh dari *Google Play*.

Meningkatnya penggunaan ponsel pintar berbasis *platform* Android dapat memberikan juga tantangan baru pengguna ponsel pintar ini yang akan dikaitkan dengan kegiatan *cyber crime*. *Digital Forensic Analysis Team* (DFAT) menangani barang bukti elektronik di tahun 2010 dengan porsi terbesarnya adalah *handphone* dan *hardisk* komputer. Trend kejahatan di Indonesia yang melibatkan perangkat digital, mengalami peningkatan signifikan sejak tahun 2010 hingga Oktober 2012 diperkirakan telah masuk lebih dari 400 kasus (Krisis, Unep, Arsitektur, & Energi, 2012). Peningkatan tingkat kejahatan elektronik yang signifikan menuntut alat bantu analisis dan investigasi forensik Android yang mumpuni.

Mobile Forensics merupakan cabang dari digital forensik yang berkaitan dengan pemulihan bukti digital atau data dari perangkat *mobile,* namun juga dapat berhubungan dengan perangkat digital berupa *smartphone* yang memiliki memori internal dan kemampuan komunikasi. Dalam dunia *mobile forensics* salah satu metode untuk mengamankan barang bukti digital (*digital evidence*) adalah dengan *disk cloning* atau lebih dikenal dengan *Disk Imaging*. *Disk Imaging* adalah proses pembuatan sebuah arsip atau salinan seluruh konten pada sebuah hard drive yang akan menghasilkan berupa *Disk Image, Disk Image* adalah file penyimpanan yang berisi semua data yang tersimpan dari sumber hard drive (Stone, 2006). dari *Disk Image* tersebut menjadi barang bukti digital yang dihasilkan dari proses *Forensic Imaging* yang memungkinkan dapat dianalisis dan di investigasi secara menyeluruh.

Namun untuk melakukan proses *disk imaging* pada ponsel android memiliki 2 metode. Metode pertama ialah metode dengan melakukan *root* dan *rooting* dalam ponsel android, dengan metode ini hanya memasang sebuah aplikasi SuperSu dan busyBox dalam ponsel android yang sudah tersedia di *playstore*, namun metode ini tidak menjamin atau rentan

terhadap berubahnya data saat dilakukan proses akuisisi karena ponsel android dalam keadaan sistem. Sedangkan metode kedua ialah melakukan dengan kustom pemulihan (*recovery mode*) yang ter-*install* dalam ponsel android tanpa harus *rooting* ponsel, dengan metode ini dapat menjaga sebuah integritas data sehingga kemungkinan berubahnya data saat dilakukan proses akuisisi sangat kecil dan memastikan kepentingan dibidang *digital forensics* untuk mengamankan barang bukti yang sah dan dapat dipergunakan dalam peradilan, namun dengan metode ini tidak semua ponsel android mendukung *custom recovery* seperti TWRP.

Berdasarkan penjelasan di atas dapat disimpulkan bahwa perangkat ponsel android sangat berpotensi sebagai salah satu objek penelitian digital forensik yang dapat menghasilkan barang bukti dari sebuah perangkat dan perangkat tersebut adalah bukti dari kasus, sedangkan penggunaan *custom recovery* TWRP mempunyai kelebihan untuk mengamankan barang bukti dalam memastikan tidak ada data yang berubah saat dilakukan pencitraan penyimpanan perangkat android. Sehingga penelitian ini dilakukan untuk mengembangkan aplikasi *Forensic Imaging* pada ponsel Android dengan melakukan pengembangan dalam tampilan aplikasi untuk menampilkan *progress bar* secara *real time* saat melakukan proses akuisisi dan juga menjaga integritas data yang dihasilkan dengan memanfaatkan *Custom Recovery* demi keamanan barang bukti yang dapat mengurangi berubahan pada data saat melakukan proses akuisisi, maka penulis mengajukan Tugas Akhir dengan judul "Analisis dan Perancangan Aplikasi *Forensic Imaging* Ponsel Android dengan memanfaatkan *Custom Recovery*".

.

1.2 Rumusan Masalah

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

a. Bagaimana mengembangkan aplikasi yang dapat melakukan *forensic imaging* pada penyimpanan ponsel android?
b. Bagaimana melakukan *forensic imaging* dengan memanfaatkan *Custom Recovery*?
c. Bagaimana melakukan pengujian hasil *forensic imaging* meliputi pengujian integritas data dan performa aplikasi?

1.3 Batasan Masalah

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 ter-*install Custom recovery* TWRP.
c. Perangkat Android harus ter-*install BusyBox* yang akan dibaca TWRP.
d. USB debugging dalam keadaan aktif.
e. Menjalankan proses akuisisi untuk 1 perangkat ponsel Android.

1.4 Tujuan

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

a. Mengembangkan aplikasi dengan menampilkan berjalannya perkembangan proses akuisisi *secara real-time* berupa p*rogress bar* dalam tampilan aplikasi, dapat mengakuisisi penyimpanan internal maupun eksternal ponsel android, dan memanfaatkan *custom recovery* TWRP yang dapat menjaga integritas data agar tidak berubah saat dilakukan proses akuisisi.
b. Mengakuisisi penyimpanan ponsel android yang sudah ter-*install custom recovery* TWRP dan ponsel android dalam kondisi *custom recovery*.
c. Mengujikan integritas data yang dihasilkan, mengujikan performa kecepatan transfer data saat dilakukan proses akuisisi penyimpanan, pengujian kuesioner skala linkert yang ditujukan ke seseorang yang mengerti bidang forensika, dan melakukan analisis terhadap hasil *disk image* yang didapatkan dari aplikasi dengan bantuan *tool* forensika digital.

1.5 Manfaat Penelitian

Dengan melakukan penelitian ini, diharapkan dapat memberikan beberapa manfaat, dengan rincian sebagai berikut:

Bagi peneliti :

a. Mendapatkan ilmu tentang *forensic imaging* pada ponsel android.
b. Mempelajari bahasa pemograman *python.*

Bagi Masyarat/sSeorang dalam bidang forensika:

a. Aplikasi ini dapat menjaga integritas data agat tidak berubah saat dilakukan proses akuisisi dengan memanfaatkan *custom recovery*.
b. Aplikasi dapat digunakan dengan baik yang dapat mengektraksi peyimpanan internal maupun eksternal pada perangkat yang menghasilkan berupa *disk image* yang akan dianalisis dan diinvestigasi forensika.

1.6 Metode Penelitian

Peneliti akan menggunakan beberapa metode dalam melakukan penelitian sebagai berikut:

a. Analisis Kebutuhan

Peneliti mengumpulkan data untuk mengetahui apa saja yang dibutuhkan untuk melakukan pencitraan pada perangkat Android. Data yang diperoleh melalui studi pustaka, kebutuhan data, kebutuhan antarmuka, dan kebutuhan *tools* sebagai referensi atau rujukan untuk menganalisa tampilan aplikasi GUI yang ingin dibuat. Dalam studi pustaka dilakukan dengan mengumpulkan dan mempelajari buku-buku, jurnal penelitian dan artikel yang terkait mengenai *Forensic Imaging* pada perangkat Android, dan kegunaan TWRP untuk keperluan pencitraan perangkat. Peneliti menggunakan jurnal dan artikel penelitian sebagai referensi utama dalam peneletian ini yang didapat melalui pencarian di internet maupun di perpustakaan.

b. Perancangan dan Pengembangan Aplikasi

Pada tahap ini membuat rancangan aplikasi *Forensic Imaging* pada perangkat Android yang ingin dibuat dengan membuat alur program untuk aplikasi GUI dalam bentuk *Flowchart* dan untuk pengembangan aplikasi melalui proses analisis model penyimpanan data pada perangkat Android. Pada proses ini dilakukan untuk mengetahui model penyimpanan yang terdapat pada perangkat android untuk mengetahui *Partitions Layout* dan *file system* pada perangkat setelah mengetahui *Partitions Layout* yang ada pada perangkat dengan merujuk ke suatu partisi. Dan partisi tersebut adalah target untuk melakukan pencitraan perangkat.

c. Implementasi

Pada tahap ini merupakan membangun sebuah aplikasi sesuai kebutuhan dan aplikasi dibuat berdasarkan dari tahap rancangan yag telah dibuat sebelumnya pada tahap ini akan di implementasikan menggunakan perangkat laptop yang sistem operasinya adalah windows dan menggunakan bahasa pemograman *python* serta modul-modul *python* yang dibutuhan dalam pembuatan aplikasi.

d. Pengujian

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

dan sumber, dan mengetahui kekurangan aplikasi tersebut kemudian dapat dilakukan perbaikan.

1.7 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun sebagai gambaran umum untuk memudahkan dalam memahami penelitian yang dijalankan. secara garis besar, sistematika penulisan penelitian ini adalah sebagai berikut:

a. BAB I : PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

b. BAB II : LANDASAN TEORI

Bab ini berisi penjelasan teori-teori yang digunakan sebagai landasan untuk menyelesaikan permasalahan yang dibahas pada penelitian ini. Bahasan dalam penelitian ini yaitu penelitan sebelumnya, *digital forensic*, *mobile forensics*, *Forensic Imaging*, dan lainnya

c. BAB III : METODOLOGI PENELITIAN

Bab ini berisi penjelasan metode yang akan digunakan peneliti selama melakukan penelitian. Selain itu bab ini juga berisi penjelasan mengenai rancangan dan kebutuhan dalam penelitian ini.

Metodologi yang akan dibahas meliputi Analisis kebutuhan system dan Perancangan sitem. Perancangan yang akan dibahas antara lain meliputi *flowrchart* sistem, rancangan antarmuka tampilan 1 dan 2 Aplikasi.

d. BAB IV : HASIL DAN PEMBAHASAN

Bab ini berisi hasil implementasi penelitian dan pembuatan Aplikasi yang telah dilakukan, berdasarkan perancangan yang telah dibuat sebelumnya. Bab ini juga membahas hasil beberapa implementasi dari *scanning* ponsel Android, mengkoneksikan ponsel Android dan melakukan proses akuisisi perangkat Android dalam pengujian apliksi dan melakukan analisis dan perbandingan dari *file* image yang dihasilkan aplikasi.

e. BAB V : KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan dari keseluruhan penelitian yang telah dilakukan. Kemudian dari kesimpulan yang didapatkan, peneliti akan memberikan saran guna mengembangkan penelitian yang telah dilakukan peneliti.

BAB II
LANDASAN TEORI

2.1 Penelitian Sebelumnya

Penelitian ini dilakukan membutuhkan landasan teori untuk mendapatkan informasi – informasi yang sangat dibutuhkan untuk menunjang dan mendukung menyelesaikan tugas akhir ini. Informasi yang didapatkan dari penelitian-penelitian yang sebelumnya telah dilakukan dan memiliki pembahasan yang sejalur dengan penelitian ini, penelitian tersebut antara lain adalah :

Rapiq Differential Forensic Imaging of Mobile Devices adalah jurnal penelitian yang dilakukan oleh (Guido, Buttner, & Grover, 2016) memperkenalkan teknik atau metode akuisisi menggunakan tool *Hawkeye* dan algoritma *forensic diferensial* otomatis yang menggunakan kumpulan data. Penelitian ini menjelaskan bagaimana mengekstraksi *physical extraction* pada penyimpanan ponsel Android dengan cepat dan menggunakan *custom recovery TeamWin Recovery Project* (TWRP) dan juga membandingkan penggunaan *tool Hawkeye* dengan tool atau teknik yang sering digunakan untuk keperluan *mobile forensics* saat ini. Sehingga penelitian ini dapat digunakan sebagai referensi atau rujukan untuk mengerjakan tugas akhir yang akan berfokus pada penggunaan *custom recovery* TWRP yang memiliki hak istimewa *root* pada ponsel dan juga metode ekstraksi yaitu *physical extraction*.

Analisis Forensik pada platform Android adalah jurnal penelitian yang di lakukan oleh (Yadi & Kunang, 2014) menjelaskan bagaimana menganalisis, memperoleh data dan bukti serta mengumpulkan informasi berharga pada ponsel android menggunakan beberapa tool yang tersedia untuk mengekstrak data dari perangkat Android, dan penelitian ini bertujuan untuk mengevaluasi kinerja tool ekstraksi yang mendukung ponsel Android. Pada penelitian ini menghasilkan data ektraksi dari android berupa format (.dd). sehingga penelitian ini dapat juga sebagai referensi atau rujukan untuk mengerjakan tugas akhir yang berfokus menghasilkan sebuah *Disk Image* berformat (.dd) dari proses ektraksi dengan metode *physical extraction* pada perangkat Android dan menganalisis data tersebut dengan bantuan tool forensika digital.

Ada bebarapa penelitan lainnya yang digunakan sebagai pendukung rujukan dari penelitian di atas dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tabel Penelitian Lainnya

No	Judul	Penulis	Pencapaian	Saran
1.	Fastdd: an Open Source Forensics Imaging Tool	(Bertasi & Zago, 2013)	Penelitian yang menghasilkan Tool *Disk Imaging* bernama Fastdd. Tool *Forensic Imaging* yang kinerjanya memiliki kebutuhan sumber daya yang lebih sedikit dan alat tersebut cepat, ringan dibandingan dengan alat saat ini dan cocok untuk perangkat manapun. Fastdd menggunakan sistem sumber daya yang bijak bias dengan cepat membuat *Forensic Imaging* dengan lebih rendah penggunaan CPU.	Untuk kedepannya penelitian ini menginginkan bahwa *tool Fastdd* ini dapat bekerja atau ditanamkan dalam perangkat kecil, seperti *raspberry Pi*. Saat ini ada penelitian yang sedang berekperimen dengan menggunakan Rasberry Pi dan ingin kedepannya penelitian ini dikembangkan lagi.
2.	Automating Disk Forensics Processing with SleuthKit, XML, and Python	(Garfinkel & Ph, 2009)	Penelitian yang menghasilkan sebuah program yang bernama fiwalk yang menghasilkan XML terperinci yang menjelaskan semua partisi dan filepada harddrive atau disk images, serta metadata yang diekstrak dari dokumen file. Penelian ini membuat aplikasi forensic disk otomatis dengan menggunakan sebuah modul dalam Bahasa pemograman Python, modul tersebut dapat membaca file XML dari fiwalk. Program fiwalk ini menghasilkan XML yang terkait dengan metada system file dan modul Python berguna untuk menelah file	Aplikasi fiwalk harus dapat dikembangkan untuk dapat lebih terintegrasi dengan SleuthKit.

No	Judul	Penulis	Pencapaian	Saran
			XML menjadi model sederhana secara otomatis dari disk images dan perangkat penyimpanan eksternal.	
3.	Mobile Forensics: Android	(Forensics & Santos, 2015)	Merupakan sebuah jurnal yang bertujuan untuk menyampaikan bagaimana belajar memahami struktur direktori , penyimpanan data dan mengekstrak data dari perangkat android. dan juga jurnal ini menjelaskan bagaimana mempelajari cara mengakses penyimpanan data android, dan mempelajari kebutuhan dan alur untuk mengekstrak data dari android.	Kedepannya jurnal ini harus dikembangkan lagi sehingga menjadi bahan pelajar untuk mempelajari system dalam android.
4.	Logical Acquisition and Analisys of Data from Android Mobile Devices	(Srivastava & Tapaswi, 2015)	Penelitian yang menjelaskan sebuah pendeketan yang membantu dalam perolehan data langsung serta data yang tersimpan dalam memori internal maupun eksternal dalam sebah perangkat ponsel Android, agar data pada perangkat tidak berubah selama proses ekstraksi. Selain itu penelitian ini penekananya diletakkan pada pengujian validitas alat forensic yang ada terhadap data yang diperoleh secara manual. Metode penelitian ini menggunakan metode *logical extraction* untuk mengekstrak data dari perangkat android menggunakan *Android Debug Bridge*(ADB).	Kedepannya penelitian ini harus dikembangkan dengan melakukan ke versi android selanjutnya yang memiliki penyimpanan lebih besar. Seperti android versi 6.0 dan seterusnya.
5.	Mobile Forensics Phone	(Heriyanto, 2016)	Ini adalah sebuah buku yang mempelajaris tentang *filesystem* yang terdapat pada perangkat Android	

2.2 Analisis dan Perancangan

2.2.1 Pengertian Analisis

Analisis adalah kajian yang dilaksanakan terhadap sebuah Bahasa guna meneliti struktur bahasa tersebut lebih dalam. Menurut Kamus Besar Bahasa Indonesia (KBBI, 2001) pengertian analisis adalah penguraian suatu pokok persoalan atas bagian-bagian, penelaahan bagian-bagian tersebut, dan hubungan antar bagian untuk mendapatkan pengertian yang tepat dengan pemahaman secara keseluruhan.

2.2.2 Analisis Data

Analisis data merupakan sebuah cara untuk mengolah data menjadi informasi agar karakteristik data tesebut mudah dipahami dan bermanfaat untuk solusi permasalahan, terutama yang berkaitan dengan penelitian. Analisis data juga diartikan sebagai kegiatan yang dilakukan merubah data dari hasil penelitian menjadi informasi yang nantinya dapat dipergunakan untuk mengambil kesimpulan. Analisis data merupakan bagian yang amat penting saat melakukan penelitian karena dapat memberi makna yang berguna untuk masalah penelitian. Data yang dihasilkan dan dikumpulkan penelitian tidak akan berguna apabila tidak dianalisis terlebih dahulu.

Untuk analisis data dilakukan melalui tahapan-tahapannya, supaya dalam melakukan analisis data mendapatkan hasil yang kemudian dapat dimanfaatkan untuk proses selanjutnya yaitu untuk dapat mengambil sebuah kesimpulan. Berikut ini tahapan-tahapan untuk menganalisis data (SORA N, 2015) :

a. Pengumpulan data adalah mengumpulkan data yang akan dianalisis
b. Reduksi data yaitu memilih hal-hal pokok yang sesuai dengan fokus penelitian, atau juga sebagai bentuk analisis yang menggolongkan, mengarahkan, membuang yang tidak perlu, dan mengorganisasikan data-data untuk memberikan gambaran yang jelas tentang data tersebut dan mempermudah peneliti untuk mencari data saat diperlukan.
c. Penyajian data adalah sekumpulan informasi tersusun yang memberikan kemungkinan yang dapat ditarik kesimpulan dan pengambilan tindakan. Penyajian data juga berarti peneliti dapat dengan mudah memahami dan menguasi data.
d. Pengambilan kesimpulan dan verifikasi, pada tahap ini peneliti mencari pola model, tema, hubungan, persamaan, hipotesis dan sebagainya, sehingga dari data tersebut peneliti dapat mengambil kesimpulan.

2.2.3 Pengertian Perancangan

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 system (*system flowchart*), yang merupakan gambar bentuk grafik yang dapat digunakan untuk menunjukan ururat-urutan proses dari sistem (Cahyono, 2015). Perancangan sistem adalah kegiatan merancang dan menentukan cara mengelolah sistem informasi dari hasil analisa sistem sehingga sistem tersebut sesuai *requirement* (Nafisah, 2003). Pada saat membuat sebuah sistem yang akan digunakan untuk kepentingan *mobile forensic*, setiap pengembang aplikasi diharuskan membuat sebuah rancangan dari sistem yang ingin dibuat seperti *flowchart.* Rancangan ini bertujuan untuk memberi gambaran umum dari sistem yang akan berjalan kepada setiap pengguna sistem.

2.3 *Digital Forensic*

2.3.1 Pengertian *Digital Forensic*

Digital forensic disebut sebagai ilmu forensik digital merupakan cabang dari ilmu forensik meliputi pemulihan dan investigasi dari bahan yang ditemukan dalam perangkat digital berupa media penyimpanananya. *Digital Forensic* adalah ilmu yang menganalisa barang bukti digital sehingga dapat dipertanggungjawabkan di pengadilan. Barang bukti digital tersebut termasuk *handphone, notebook, server,* alat teknologi apapun yang mempunyai media penyimpanan dan bisa dianalisa (Ruby Alamsyah, 2013).

Dari pengertian di atas dapat disimpulkan bahwa digital forensik merupakan teknik atau cara menangani barang bukti digital untuk diproses dan menghasilkan informasi yang berguna untuk keperluan pengadilan (Ruby Alamsyah, 2013), dalam *digital forensics* memiliki metodologi yang berguna sebagai alur kerja penyelesaiana kasus, yaitu:

a. *Collection*
b. *Examination*
c. *Analisys*
d. *Reporting*

2.3.2 *Mobile Forensics*

Mobile Device Forensics merupakan cabang dari *digital forensic* yang berkaitan dengan pemulihan bukti digital atau data dari perangkat *mobile* dengan menggunakan teknik tertentu

sesuai kaidah forensik digital, namun juga dapat berhubungan dengan perangkat digital yang memiliki media penyimpanan seperti memori internal maupun eksternal dan kemampuan komunikasi (Rohit Tamma, 2015).

Forensik perangkat *mobile* merupakan forensik yang datanya diambil dari sebuah ponsel dan dapat dijadikan sebagai bukti. Bukti-bukti ini bisa menjadi landasan ketika menyelidiki suatu perkara oleh lembaga penegak hukum. Terdapat sejumlah bukti yang dapat di ekstraksi dari ponsel, seperti media penyimpanannya Intenal dan Eksternal. Dari media penyimpanan itu dapat diektraksi menjadi beberapa data-data dari ponsel antara lain nomor kontak, sms, file doc, file audio, email dan internet *history* (Yadi & Kunang, 2014).

Forensik *mobile* juga menggunakan metode yang sama dengan investigasi forensik umum, ada beberapa teknik yang perlu diikuti untuk melakukan standar penyelidikan di foresnik *mobile*. Metode penyelidikan kurang lebih sama dengan investigasi digital. Tahapan proses penyelidikan sebagai berikut (Yadi & Kunang, 2014):

a. *Collection* : merupakan langkah awal dan paling penting dalam penyelidikan. Tujuannya adalah untuk mengumpulkan sumber-sumber bukti potensial seperti ponsel, kartu SIM, dan lainnya.
b. *Identification* ; yang berfokus pada pengenalansumber barang bukti dengan pelabelan.
c. *Acquisition*: berkaitan dengan proses ekstraksi data atau bukti potensial dari berbagai sumber yang telah dikumpulkan.

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), antara lain adalah :

a. Tahap Intake/pengambilan bukti. Pada tahap ini melibatkan prosedur di mana permintaan pemeriksaan di setujui. Dan fase ini pengambilan bukti umumnya mencakup formulir untuk mendokumentasikan pengambilan perangkat seluler untuk pemeriksaan.
b. Tahap *identificatio*n. Pada tahap ini pemeriksa harus dapat mengidentifikasi beberapa hal :
    1. Kewenangan hukum untuk memeriksa perangkat.
    2. Tujuan pemeriksaan

3. Identifikasi perangkat, dari merk perangkat, model, spesifikasi, atau informasi yang terkait dengan barang bukti yang didapat.

Pada point ke 3 terdapat salah satu rintangan yaitu ; banyaknya *merk* dari perangkat ponsel saat ini.

c. Tahap *preparation*, pada tahap ini dilakukan setelah data pendukung sudah didapatkan maka dilakukan persiapan, seperti menggunakan metode dan keperluan *tools* apa saja yang akan digunakan dalam proses ekstraksi dan analisis.
d. Tahap *isolation*. Pada tahap ini adalah proses yang penting agar perangkat tidak terhubung dengan jaringan komunikasi seperti jaringan telpon seluler, bluetooth, dan WIFI. Dan tahap ini mencegah perangkat ada penambahan data ke dalam perangkat melalui jaringan komunikasi. Pada tahap ini biasa juga menggunakan *faraday bag* untuk membantu agar perangkat terisolasi dari komunikasi luar.
e. Tahap *processing*, tahap ini adalah memproses barang bukti tersebut dengan melakukan ekstraksi dari barang bukti setelah mengetahui metode ekstraksi yang sudah ditentukan pada tahap "Preparation" dan itu dilakukan analisa terkait temuan-temuan yang didapat dari proses ekstraksi tersebut.
f. Tahap *verification*, tahap ini melakukan proses verifikasi keakuratan data ekstraksi yang didapat dari tahap processing. Verifikasi data yang diekstrak yang didapat dengan cara:
    1. Membandingkan data yang telah diekstrak dengan data dalam barang bukti yang didapat.
    2. Memeriksa *hex* dari data yang di ekstraksi.
    3. Mencoba beberapa *tools* untuk membandingkan hasilnya.
    4. Menggunakan *hash value* untuk membandingkan hasil ekstraksi berupa file *image* dengan *hash value* dari barang bukti. Seperti MD5, SHA512, SHA1 dan lainnya.
g. *Documentation & Reporting*, pada tahap ini adalah proses dokumentasi yang dilakukan dari tahap perolehan samapai pada tahapan-tahapan selajutnya, setalah itu pelaporan dilakukan dengan baik secara ringkas dan jelas agar mudah dipahami pleh pihak-pihak yang pumya otoritas.

Dalam dunia *mobile forensik* salah satu metode untuk mengamankan barang bukti digital (*digital evidence*) adalah dengan *Disk Imaging* atau lebih dikenal dengan *disk clonning.*

2.4 *Disk Imaging*

Disk Imaging atau sering dikenal dengan *disk cloning* adalah proses atau teknik untuk menyalin langsung dari sector perangkat penyimpanan fisik (*physical storage device)* / media penyimpanan dari sebuah disk termasuk *file, folder, unallocated*, dan ruang bebas secara *bitstream image* yaitu menyalin data *bit – bit* dan mengambil *sector –by- sector* yang terdapat pada *disk block* dan menghasilkan berupa file *image / disk image*. *Disk image* adalah hasil dari melakukan proses *disk Imaging* yaitu file penyimpanan yang berisi semua data yang tersimpan dari sumber barang bukti. Seperti smartphone (Garfinkel & Ph, 2009; Stone, 2006).

2.4.1 Format *Raw Image* (.dd)

Format *raw iamge* pada dasarnya adalah salinan *bit-for-bit* dari data *raw* baik dari disk, tanpa ada tambahan dan penghapusan. Tidak ada metadata yang disimpan dalam format *raw*, namun terkadang metadata disimpan dalam file tambahan. Format *raw image* digunakan untuk menyimpan *disk* atau *volume image*, ada beberapa skema untuk penamaan file format *raw image* yang digunakan untuk *disk* atau *volume image* yaitu salah satunya file format (.dd)

Dd atau sering disebut GNU dd merupakan alat pencitraan tertua yang masih digunakan. Alat pencitraan tersebut hanya membutuhkan sedikit sumber daya untuk dijalankan. Dd adalah program baris perintah yang menggunakan beberapa argument baris perintah yang tidak dikenal untuk mengontrol pencitraan. Namun dari baris perintah yang tidak dikenal tersebut dapat menghancurkan media sumber yang akan digandakan atau dilakukan proses akuisisi.

2.5 Bahasa Pemograman *Python*

Python adalah bahasa pemograman berorientasi objek, interpretasi, dan interaktif yang hebat. Dikarenakan sering dibandingkan dengan *Lisp, Tcl, Perl, Ruby, C#, Visual Basic, Visual Fox Pro, dan Java*. *Python* menggunakan sintaks yang sangat jelas dan juga *Python* memiliki modul (*modules*), kelas (*classes*), pengecualian (*exception*), tipe data yang dinamis, dan pengetikan dinamis, serta memiliki antarmuka untuk banyak panggilan sistem dan pustaka.

2.6 *Custom Recovery*

Custom Recovery adalah *Thrid-party recovery environment* yaitu *custom recovery* yang mengganti *stock recovery* yang ada dalam ponsel android, setiap ponsel android memiliki *stock recovery* pabrikan namun kegunaannya sangat minim. Namun untuk *custom recovery* memiliki banyak manfaat dan sangat penting untuk dipasang dalam ponsel android karena dapat

bermanfaat untuk penginstalan Mod, membersihkan *Dalvik cache* yang bergunaan mengatasi *lagging* pada ponsel, mengatasi *bootloop*, mem-*backup* semua data dalam ponsel, dan masih banyak lagi. Jadi *custom recovery* dapat mencegah hal-hal yang tidak diinginkan pada ponsel android, saat ini *custom recovery* yang sangat popular adalah TWRP dan CWM.

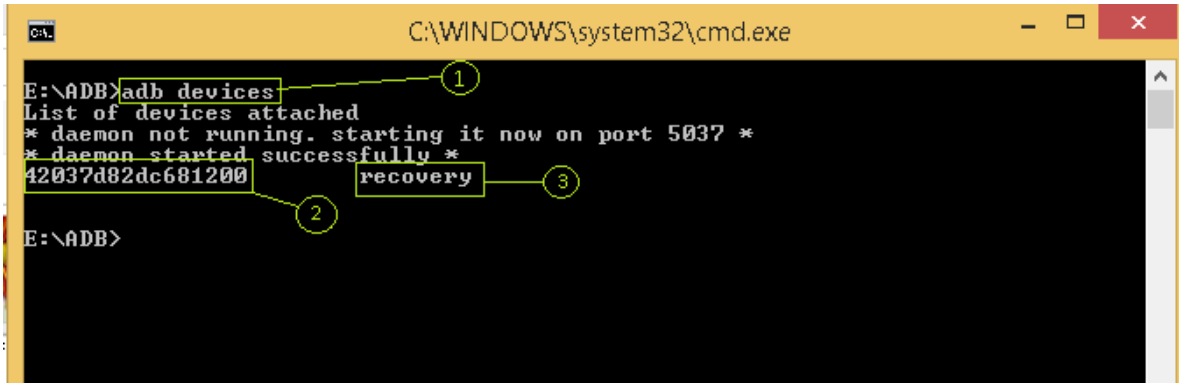
2.7 Analisis Model Penyimpanan Data pada Ponsel Android

Motode *Disk Imaging* adalah proses menggandakan data dari sebuah media penyimpanan yang terdapat dalam ponsel Android yang akan menghasilkan sebuah file image atau sering dikenal disk image dana dapat dianalisis secara menyeluruh(Stone, 2006). Proses analisis bertujuan untuk mengetahui berbagai jenis data apa saja yang terdapat dalam ponsel android, seperti ukuran media penyimpanan, partisi penyimpanan, dan mengetahui *file system* apa saya yang digunakan dalam ponsel android untuk menyimpan data-data (Regan, 2009).

Namun untuk mengetahui ukuran, partisi dan *file system* yang digunakan dalam penyimpanan ponsel android. Penulis akan melakukan beberapa analisis untuk mengetahui semua informasi tentang model penyimpanan yang terdapat dalam ponsel android. Analisis tersebut diantaranya, yaitu :

a. Analisis *structure directory* ponsel android
b. Analisis *partition layout* ponsel android.
c. Analisis *filesystem* ponsel android.
d. Analisis ukuran penyimpanan ponsel android

Untuk melakukan semua analisis di atas dibutuhkan sebuah ponsel android sebagai objek analisis yang sudah ter-*isntall custom recovery* dan *busybox* dan juga mengkondisikan ponsel android dalam *custom recovery* yang memiliki hak akses istimewa *root* dan USB *debugging* aktif. Sedangkan untuk perangkat laptop yang digunakan untuk analisis pastikan ponsel android yang sudah terhubung dengan menggunakan kabel USB dan mempersiapkan *tool Android Debug Bridge*(ADB) sebagai *tool* penghubung hak akses *custom recovery* dalam ponsel android. Untuk melihat fungsi *tool* ADB melalui *command prompt* perangkat laptop untuk mengakses *custom recovery* dalam ponsel android dapat dilihat pada Gambar 2.1



Gambar 2.1 Ponsel Android Terhubung ke PC melalui ADB

Pada Gambar 2.1 menjelaskan pada perintah (1)“ädb devices” adalah perintah yang berguna menampilkan ponsel android yang terhubung ke laptop/*host* menggunakan kabel USB, sedangkan (2) “42037D82DC681200” menunjukan *serial number* pada ponsel android yang terhubung dengan pc/laptop, (3) “*recovery”* menunjukan keadaan perangkat dalam *recovery mode.*

.

BAB III
METODOLOGI PENELITIAN

3.1 Alur Penelitian

Alur penelitian yang penulis lakukan untuk mengembangkan Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recocvery* memiliki beberapa tahapan untuk melakukan pembuatan Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*, tahapan-tahapan tersebut seperti pada Gambar 3.1.



Gambar 3.1 Tahapan Alur Penelitian

3.2 Analisis Kebutuhan Sistem

Pembuatan untuk pengembangan Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* membutuhkan sebuah perangkat keras terutama perangkat ponsel Android, serta memahami model penyimpanan data perangkat Android, aplikasi pendukung, dan beberapa modul dalam bahasa pemograman *python* yang dapat dilakukan perancangan, menjalankan, dan pembuatan tampilan aplikasi. Untuk mengetahui semua itu maka diperlukan analisis lebih lanjut terhadap kebutuhan aplikasi ini, diantaranya adalah analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak, analisis model penyimpanan data pada perangkat Android, analisis kebutuhan masukan (*input*), analisis kebutuhan proses, dan analisis kebutuhan keluaran (*output*).

3.2.1 Analisis Kebutuhan Perangkat Keras

Pada proses perancangan dan pengembangan aplikasi terdapat beberapa komponen perangkat keras yang digunakan untuk membuat dan menjalankan aplikasi yang akan dibuat. Perangkat keras tersebut sangat dibutuhkan agar pambuatan aplikasi menjadi lancar, yang dimaksud perangkat keras adalah suatu barang elektronik yang berperan sebagai wadah pembuatan aplikasi dan juga barang bukti digital berupa perangkat ponsel Android untuk keperluan mengakuisisi penyimpanannya. Perangkat keras tersebut diantaranya :

a. Perangkat Laptop

   Perangkat keras sebuah laptop dengan processor AMD memiliki kecepatan 1.48GHz dan RAM 6GB yang bersistem operasi windows 8.1 64-bit sebagai perangkat keras yang digunakan peneliti untuk membangun aplikasi dan menjalakan aplikasi yang telah dibuat.

b. Perangkat Ponsel Android 1

   Perangkat keras sebuah ponsel Android dengan sistem android 4.0 yang memiliki RAM 512 MB dan mempunyai penyimpanan internal sebesar 4GB yang akan diakuisisi, ponsel tersebut sebagai contoh barang bukti 1 yang digunakan untuk melakukan proses akuisisi penyimpanan dalam aplikasi. Ponsel ini telah mendukung dan ter-*install Custom Recovery*.

c. Perangkat Ponsel Android 2

   Perangkat keras sebuah ponsel Android dengan sistem android 5.0 yang memiliki RAM 1GB dan mempunyai penyimpanan internal sebesar 8GB yang akan diakuisisi, ponsel tersebut sebagai contoh barang bukti 2 yang digunakan untuk melakukan proses

akuisisi penyimpanan dalam aplikasi. Ponsel ini telah mendukung dan ter-*install Custom Recovery*.

d. Perangkat Ponsel Android 3

Perangkat keras sebuah ponsel Android dengan sistem android 6.0 yang memiliki RAM 2GB dan mempunyai penyimpanan internal sebesar 16GB yang akan diakuisisi, ponsel tersebut sebagai contoh barang bukti 3 yang digunakan untuk melakukan proses akuisisi penyimpanan dalam aplikasi. Ponsel ini telah mendukung dan ter-*install Custom Recovery*

e. Kabel USB.

Perangkat keras sebuah kabel USB yang berguna untuk menghubungkan perangkat ponsel Android ke dalam perangkat laptop untuk kepentingan proses akuisisi penyimpanan ponsel android.

3.2.2 Analisis Kebutuhan Perangkat Lunak

Pada proses perancangan dan pengembangan aplikasi terdapat beberapa komponen perangkat lunak yang digunakan untuk membuat tampilan aplikasi dan menjalankan proses dalam aplikasi, yang dimaksud perangkat lunak adalah sebuah *tool* yang digunakan untuk membuat tampilan aplikasi, menjalalankan proses yang terjadi diaplikasi, dan untuk melakukan analisis. Perangkat lunak tersebut diantaranya:

a. Windows 8.1 (64 bit) .

Windows 8.1 64-bit adalah sistem operasi yang digunakan dalam perangkat laptop untuk membuat dan menjalankan aplikasi *forensic imaging* pada ponsel android.

b. Aplikasi *Text Editor*.

Aplikasi text editor adalah aplikasi yang menggunakan Bahasa pemograman *python* yang ter-isntall dalam perangkat laptop yang digunakan untuk mengembangkan dan membuat aplikasi forensic imaging pada ponsel android dengan memanfaatkan *Custom Recovery*.

c. Aplikasi QtDesigner.

Aplikasi QtDesigner adalah aplikasi yang berguna untuk merancang atau mendesain tampilan antarmuka aplikasi yang ingin dibuat atau dikembangkan yang ter-*install* dalam perangkat laptop.

d. Bahasa Pemograman *Python*.

Bahasa pemograman python adalah bahasa yang digunakan untuk mengembangkan dan membuat aplikasi *forensic imaging* pada ponsel android dengan menggunakan bahasa pemograman *python* versi 3.6.4 dan terhubung dengan text editor dalam perangkat laptop.

e. *Custom Recovery* TWRP.

Custom Recovery TWRP adalah *tool* yang terpasang dalam ponsel android ke mode pemulihan (*recovery mode*) yang berguna untuk kepentingan proses akuisisi penyimpanan ponsel tersebut melalui aplikasi yang telah dibuat.

f. Odin versi 3.12.3.

Aplikasi Odin versi 3.12.3 adalah aplikasi bantuan untuk *flashing file* TWRP yang telah diunduh ke dalam ponsel Android yang berguna untuk keperluan proses akuisisi penyimpanan ponsel android tersebut.

g. *Busybox Bionic*.

Busybox Bionic adalah *tool* yang terpasang dalam ponsel Andriod dan dapat dibaca di mode pemulihan (*recovery mode*) yang berguna untuk menjalankan perintah proses akuisisi dalam *shell* ponsel.

h. ADB Deamon.

ADB Deamon adalah *tool* yang terdapat dalam perangkat laptop yang berfungsi untuk menghubungkan ponsel android dengan perangkat laptop dan juga berfungsi untuk mendapatkan hak akses *shell* ponsel android dalam perangkat laptop.

i. Netcat versi 1.12.

Netcat versi 1.12 adalah *tool* yang terdapat dalam perangkat laptop yang berguna untuk membuat *file image* yang dihasil dari proses akuisisi yang terjadi dalam aplikasi yang telah dibuat. *Tool* ini berkerja membuat sebuah file yang mengambil sebuah data dari suatu port yang telah dilakukan dalam *shell* ponsel Android dengan mengguankan tool *busybox*.

j. WinMD5

WinMD5 adalah aplikasi untuk mencari *md5 hashing* pada sebuah file. Tool ini digunakan untuk mencocokan hasil *md5* yang dihasilkan aplikasi dengan *md5* dari yang dihasilkan aplikasi WinM5 yang akan menjadi pengujian sebuah integritas data yang dihasilkan.

k. Autopsy.

Autopsy versi 4.5.0 adalah aplikasi yang berguna untuk menganalisis sebuah *file image* yang berisis data-data penyimpanan ponsel android yang dihasilkan dari proses

akuisisi dalam aplikasi. Aplikasi autorpsy bekerja mengekstrak *file image* menjadi susunan data yang terbentuk sesuai dari formatnya, seperti gambar, video, dokumen, dan lainnya

3.2.3 Analisis Kebutuhan Masukan

Setelah melakukan proses analisis kebutuhan perangkat keras dan lunak yang sudah mempersiapkan perangkat tersebut untuk dapat terhubung ke aplikasi maka akan dilakukan proses analisis kebutuhan masukan. Pada proses ini adalah pengguna diminta untuk menghubungkan perangkat ponsel dengan perangkat laptop yang terinstal aplikasi dengan menggunakan kabel USB, selanjutnya pengguna membuka aplikasi dan melakukan mengkoneksikan perangkat Android, setelah itu diminta memasukan *file name*, *file name* ini berguna untuk penamaan *file image* yang akan dihasilkan oleh aplikasi, nama *examiner* yang berguna untuk mengetahui siapa nama yang menggunakan aplikasi, *output location* berguna untuk peletakan *file image* yang akan dihasilkan setelah melakukan proses akusisi selesai, tipe *file image* berguna untuk format *file image* yang dihasilkan dari melakukan proses akuisisi, dan memilih penyimpanan internal dan eksternal pada perangkat Android yang mau di akusisi dapat memilih kedua penyimpanan.

3.2.4 Analisis Kebutuhan Proses

Pada proses yang dibutuhkan oleh aplikasi *Forensic Imaging* Perangkat Android dengan memanfaatkan *Custom Recovery*, memiliki beberapa proses, diantaranya adalah proses *scanning* perangkat, pada proses ini adalah proses mencari perangkat android yang terkoneksi ke laptop/*host* yang akan dibaca oleh aplikasi melalui Adb Deamon, setelah melakukan proses *scanning* dan menemukan perangkat Android. Selanjutnya melakukan proses mengkoneksikan perangkat, pada proses ini membutuhkan perangkat dalam keadaan *recovery mode*/TWRP dan aplikasi akan mengkoneksikan perangkat ke port tcp:8888 sehingga aplikasi dapat melanjutkan ketahap proses akuisisi.

Setelah melakukan proses mengkoneksikan perangkat, dapat dilanjutkan ke proses akusisi, pada proses ini melibatkan 2 proses, proses pertama adalah proses yang terjadi dalam *shell* perangkat ponsel yaitu proses pembuatan *file image* berformat (*raw dd)* yang diletak pada port tcp:8888. Setelah itu dilakukan, kemudian proses kedua yaitu proses pembuatan *file image* yang akan dihasilkan ke dalam perangkat laptop/*host* yang diambil dari port tcp:8888 di proses pertama melalui *netcat*. Pada proses akusisi menggunakan fungsi *QThread* dalam modul *PyQt5* yang berguna melakukan 2 proses dalam aplikasi secara bersamaan. Setelah selesai proses

akusisi telah selesai, selanjutnya melakukan proses *hashing*. Pada proses *hashing* adalah proses membandingan *md5* dari file sumber penyimpanan dan hasil berupa *file image* penyimpanan setelah diakusisi, dari proses ini dapat diketahui apakah file sumber dan file hasil akusisi datanya ada yang berubah atau tidak.

3.2.5 Analisis Kebutuhan Keluaran

Pada tahap analisis ini adalah analisis keluaran yang dihasilkan oleh Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*. aplikasi ini akan menghasilkan keluaran berupa *file image, file log,* detail informasi proses akuisisi dalam aplikasi. Untuk penjelasan dari hasil keluaran sebagai berikut:

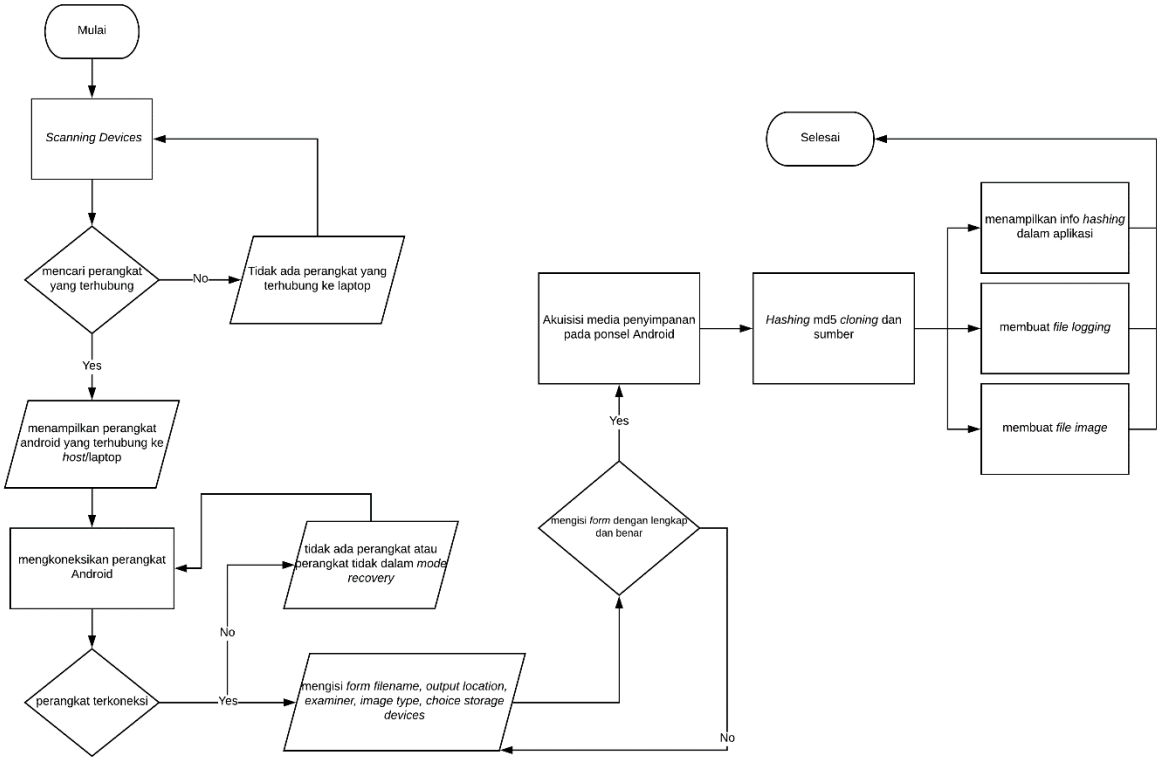
a. *File image* atau sering dikenal *disk image* adalah file penyimpanan yang berisi semua data yang tersimpan dari sumber penyimpanan baik itu internal maupun eksternal pada ponsel yang sudah diekstraksi. Peletakan *file image* terdapat pada sebuah folder yang telah di tentukan dalam kolom *output location* dalam aplikasi.
b. *File log* adalah *file text* yang berisikan semua informasi tentang proses akuisisi yang telah dilakukan dalam aplikasi. Informasi tersebut diantaranya adalah nama *examiner,* sumber direktori dari penyimpanan ponsel Android, ukuran penyimpanan baik itu internal atau eksternal, informasi waktu mulai dan selesai proses akusisi, nama *file image* yang dihasilkan, direktori peletakan *file image* yang dihasilkan, detail proses *hashing md5* dari sumber penyimpanan dengan dengan *file image* yang dihasilkan aplikasi, serta keterangan *hashing md5* sama atau tidak dari kedua file penyimpanan tersebut.
c. Detail informasi dari proses akuisisi dalam aplikasi adalah informasi yang sama dari *file log* tetapi informasi tersebut ditampilkan dalam suatu kolom pada aplikasi.

3.3 Perancangan Aplikasi

Perancangan sistem adalah kegiatan merancang dan menentukan cara mengelolah sistem informasi dari hasil analisis sistem sehingga sistem tersebut sesuai *requirement* (Nafisah, 2003). Maka perancangan aplikasi adalah suatu proses atau kegiatan merancang yang dilakukan untuk membuat konsep dan mendesain alur kerja sistem yang berguna memenuhi kebutuhan Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* sehingga aplikasi dapat bekerja dengan baik sesuai yang diingikan oleh peneliti. Proses atau kegiatan perancang aplikasi meliputi 3 aspek rancangan, yaitu *flowchart, pseudocode,* dan desain antarmuka.

3.3.1 *Flowchart*

Flowchart atau Digram alir adalah sebuah jenis diagram yang mewakili algoritma sistem, alur kerja atau proses yang menampilkan sebuah langkah-langkah dalam bentuk simbol grafis. Pada pembuatan Aplikasi *Forensic Imaging* pada ponsel Android dengan memanfaatkan *Custom Recovery*, penulis membuat alur kerja (*flowchart)* sebagai langkah-langkah menggunakan aplikasi yang dapat dilihat pada Gambar 3.2.



Gambar 3.2 *Flowchat* Aplikasi *Forensic Imaging* perangkat Android

Pada Gambar 3.2 merupakan *Flowchart* atau diagram alir yang menjelaskan cara kerja sistem. Ketika pertama kali pengguna membuka aplikasi, hal pertama yang dilakukan adalah pengguna melakukan *scanning* perangkat Android yang terhubung ke laptop/*host,* jika aplikasi menemukan suatu perangkat Android yang terhubung maka aplikasi akan menampilkan *serial number* dan status perangkat Android, jika aplikasi tidak menemukan suatu perangkat yang terhubung maka aplikasi akan memberitahu pengguna bahwa tidak ada perangkat Android yang terhubung ke laptop/*host*. Selanjutnya pengguna akan diminta mengkoneksikan perangkat Android, jika perangkat tidak dapat terkoneksi ada beberapa kemungkinan, pertama adalah perangkat tidak ditemukan/tidak terhubung, kedua adalah perangkat tersebut tidak dalam

recovery mode, namun jika perangkat sudah terkoneksi maka aplikasi akan memberitahu pengguna bahwa sebuah *serial number* pada perangkat Android sudah terkoneksi melalui suatu port dalam aplikasi. Selanjutnya pengguna diminta untuk menuju tampilan 2 pada aplikasi.

Setelah ke tampilan 2 aplikasi, pengguna akan diminta untuk memasukan *file name, examiner, check box* untuk format file, *radio box* untuk memilih penyimpanan yang akan di akusisi¸ *output locaton* untuk keluaran hasil berupa *file image* dalam folder yang ditentukan, jika pengguna memasukan semua dengan benar dan lengkap maka aplikasi dapat dilanjutkan ketahap proses akusisi, jika pengguna tidak memasukan dengan bener atau ada yang kurang dalam memasukan yang aplikasi butuhkan maka aplikasi tidak dapat dilanjutkan ketahap yaitu proses akusisi. Selanjutnya aplikasi akan melakukan proses akusisi, setelah proses akusisi telah selesai, selanjutnya aplikasi akan melakukan proses *hashing md5* dari sumber penyimpanan dan file hasil dari proses akusisi, jika aplikasi telah selesai melakukan proses *hashing*, aplikasi akan membuat sebuah *file log* dalam folder bersama *file image* dan menampilkan semua informasi pada *file log* ke dalam aplikasi yang akan menunjukan semua detail dari hasil proses akusisi dan proses *hashing*.

3.3.2 *Pseudocode*

Pseudocode adalah notasi yang menyerupai notasi bahasa pemrograman tingkat tinggi (Oktaviani, 2015). *Pseudocode* pada umumnya digunakan karena dapat mempermudah perubahan notasi ke dalam bentuk bahasa pemorograman. *Pseudocode* Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* terdiri dari 2 *pseudocode*, yaitu:

A. *Pseudocode* Tampilan 1

Pseudocode tampilan 1 Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* menggunakan beberapa modul yang terdapat dalam bahasa pemograman *python* yang memiliki fungsi masing-masing setiap modulnya. Untuk melihat modul tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tabel Modul Tampilan 1

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

No	Nama Modul	Fungsi Modul	Kegunaan
2.	Subprocess	call getoutput	Menjalankan perintah ADB Untuk membuka server adb saat pertama kali diakses. Menjalankan perintah adb dan mendapatkan keluaran
3.	PyQt5	QmainWindow QCore	Untuk membuat tampilan aplikasi. Untuk memasukan icon/gambar ke dalam tampilan aplikasi

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

Sedangkan Algoritma Aplikasi *Forensic Imaging* pada Perangkat Android dengan memanfaatkan *Custom Recovery* dapat dijelaskan dengan *pseudocode*. Untuk *pseudocode* tampilan1 aplikasi memiliki beberapa variabel yaitu variabel *scanningperangkat, statusperangkat,* dan *connectperangkat* yang berfungsi untuk menjalankan proses pencarian dan mengkoneksikan ponsel android melalui aplikasi yang dapat dilihat pada Gambar 3.3.

```
Tampilan 1 : Aplikasi Forensic Imaging pada perangkat Android dengan memanfaatkan
Custom Recovery
{Tampilan yang  memiliki fungsi untuk Scanning  dan  mengkoneksikan perangkat
Android }
```

```
Kamus :
PerintahAdb : adbdeamon
scanningperangkat : string
statusPerangkat   : string
connectPerangkat  : string
Subprocess (getoutput)  :  modul
subprocess : string
```

```
Deskripsi algoritma :
Subprocess  (getoutput)  ← subprocess

Subprocess.getoutput('perintahAdb')  ← scaningPerangkat
   If (scaningPerangkat == 'unknown') then
      Output('Perangkat tidak ditemukan')
   Else
      Output('perangkat ditemukan')

Subprocess.getoutput ('perintahAdb')  ← statusPerangkat
   If (statusPerangkat == 'unknown') then
      Output ('tidak dan perangkat yang terdeteksi')
   Elif (statusPerangkat == 'device') then
      Output ('perangkat tidak dalam mode recovery')
   Elif (statusPerangkat =='recovery') then
      Subprocess.getoutput('perintahAdb')  ← connectPerangkat
```

```
If (connectPerangkat == ‘’) then
   Output (‘perangkat terkoneksi’)
Else
   Output (‘perangkat tidak terkoneksi’)
```

Gambar 3.3 *pseudocode* Interface.py

Pada Gambar 3.3 adalah *pseudocode* tampilan 1 aplikasi *Forensic Imaging* pada ponsel android, untuk penjelasannya pada variabel *scanningperangkat* adalah varibel yang dapat menyimpanan nilai *string* dari suatu perintah pencarian ponsel android dalam *command promt*, selanjutnya variabel *statusperangkat* adalah variabel yang menyimpnan nilai *string* dari perintah mengetahui status ponsel android dalam *command prompt*, sedangkan variabel *connectperangkat* adalah variabel yang hanya menjalankan suatu perintah untuk mengkoneksikan ponsel android dalam *command promt*. Dari semua variabel menggunakan bantuan modul *subprocess* dalam bahasa pemograman *python*.

untuk mendeskripsikan algoritmanya adalah pertama pengguna diminta melakukan *scanning* ponsel Android yang sudah terhubung ke laptop/*host* melalui Adb Deamon dengan menggunakan modul *subprocess* dalam bahasa pemograman *python*, aplikasi akan menjalankan *subprocess* yang akan menghasilkan keluaran dan menampilkan *serial number* dan status ponsel android yang terhubung dalam suatu kolom pada aplikasi. Setelah menampilkan *serial number* dan status ponsel android, selanjutnya pengguna diminta mengkoneksikan ponsel android yang sudah terhubung ke laptop/*host* melalui Adb Deamon dengan menggunakan modul *subprocess*. Namun untuk dapat mengkoneksikan perangkat Android memiliki syarat yaitu mengkondisikan perangkat Android dalam *recovery mode*/TWRP. Setelah mengkondisikan ponsel Android dalam *recovery mode* maka dapat dilakukan proses mengkoneksikan perangkat. Proses mengkoneksikan perangkat sama dengan proses *scanning* dengan menggunakan modul *subprocess*, modul tersebut menghasilkan keluaran yang akan mengkoneksikan perangkat ke dalam suatu port yaitu tcp:8888. Setelah mengkoneksikan perangkat dalam suatu port telah selesai, aplikasi dapat menampilkan informasi tentang perangkat tersebut sudah terkoneksi dengan melalui port apa dan juga dapat dilanjutkan ke tampilan 2 aplikasi.

B. *Pseudocode* Tampilan 2

Pseudocode tampilan 2 Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* menggunakan beberapa modul yang terdapat dalam bahasa

pemograman python yang memiliki fungsi masing-masing setiap modulnya. Untuk melihat modul tersebut dapat dilihat pada Tabel 3.2.

Tabel 3.2 Tabel Modul Tampilan 2

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.	PyQt5	QmainWindow QThread QCore	Untuk membuat tampilan aplikasi. Untuk menjalankan beberapa proses bersamaan tanpa menggangu proses lainnya. Untuk memasukan icon/gambar ke dalam tampilan aplikasi
4.	Pyadb3	run_shell_cmd	Untuk melakukan proses akusisi di dalam *shell* perangkat Android
5.	Hashlib	md5	Untuk proses *hashing* md5 pada *file image* hasil proses akusisi
6.	Os	getsize	Untuk mendapatkan ukuran *file image* saat berjalannya proses akusisi agar dapat menjalankan *Acquisition progress*.
7.	logging	File log	Untuk membuat *file log* yang berisikan detail informasi dari proses akusisi
8.	Time	sleep	Untuk memberikan waktu jeda pada proses akuisisi penyimpanan.

Pada Tabel 3.2 merupakan tabel modul yang digunakan dalam tampilan 2 aplikasi untuk membuat algortima yang dapat menjalankan proses-proses yang dibutuhkan pada tampilan 2 aplikasi, modul tersebut memiliki peran yang sangat penting untuk menjalankan aplikasi dan pembuatan algoritma.

Sedangkan Algoritma Aplikasi *Forensic Imaging* pada Perangkat Android dengan memanfaatkan *Custom Recovery* dapat dijelaskan dengan *pseudocode*. Untuk *pseudocode*

tampilan 2 aplikasi yang berfungsi untuk menjalankan proses akuisisi dan *Hashing md5* penyimpanan ponsel android melalui aplikasi dapat dilihat pada Gambar 3.4.

```
Tampilan 2 : Aplikasi Forensic Imaging pada perangkat Android dengan memanfaatkan
Custom Recovery
{Tampilan yang  memiliki fungsi untuk proses akusisi  dan  hashing md5  }
```

```
Kamus :
perintahShell   : shell
perintahHashing : shell
str             : string
inputShell      : string
time_limit      : integer
forSleep        : integer
int             : integer
sizeFileImage   : integer
outputTimeEnd   : integer
forProgressBar  : integer
os (getSize)    :  modul
Pyadb3 (run_shell_cmd) : modul
Time (sleep)           : modul
PyQt5 (QThread)        : modul
Datetime               : modul
Logging               : modul
Shell          : string
subprocess     : string
```

```
Deskripsi algoritma :
Pyadb3(run_shell_cmd)    ←   shell
Class .... (QThread)     ←   Function

Begin
Class ShellInt(QThread):
   Function run():
      Shell.run_shell_cmd('perintahhShellInt')    ←   InputShellInt

Class ShellExt(QThread):
   Function run():
      Shell.run_shell_cmd('perintahShellExt')     ←   InputShellExt

Class NetcatInt(QThread):
   Function run():
      Subprocess.getoutput('perintahNCInt')       ←   outputNCInt
      If (outputNcInt == '') then
         Shell.run_shell_cmd('perintahHashing')   ←   hashingMd5Int
         Output(datetime('perintahDatetime'))     ←   outputTimeEndInt
         Output(hashingMd5)                       ←   outputHashingInt
         Output(Logging(fileLog))                 ←   outputFileLog

Class NetcatExt(QThread):
   Function run():
      Subprocess.getoutput('perintahNCExt')       ←   outputNCExt
      If (outputNcExt == '') then
         Shell.run_shell_cmd('perintahHashing')   ←   hashingMd5Ext
         Output(datetime('perintahDatetime')      ←   outputTimeEndExt
         Output(hashingMd5)                       ←   outputHashingExt
         Output(logging(fileLog))                 ←   outputFileLog

Class progressBarInt(QThread):
   Int = function(int)
   Function run():
      Int
      Function (int < time_limit) then
         Int++
         time.sleep(int)
```

```
         Os.path.getsize('runFileImageInt')              ←   sizeFileImageInt
         If (sizeFileImageInt>=sizeFileSumberInt) then   ←   forProgressBarInt
               Break

Class progressBarExt(QThread):
   Int = function(int)
   Function run():
      Int
      Function (int < time_limit) then
         Int++
         Time.sleep(int)
         Os.path.getsize('runFileImageExt')              ←   sizeFileImageExt
         If (sizeFileImageExt>=sizeFileSumberExt) then   ←   forProgressBarExt
            break

Function disbaleExtract():
   If statusRawdd.isChecked() and statusInternal.isChecked() and
   Function(statusFileName.str() and statusExaminer.str() and
   statusOutLoc.str()) >0: then
      Function extract.disable(false)
      If statusEksternal.isChecked():
         Function extract.disable(false)

   Elif statusRawdd.isChecked() and statusEnternal.isChecked() and
   Function(statusFileName.str() and statusExaminer.str() and
   statusOutLoc.str()) >0: then
      Function extract.disable(false)
      If statusIxternal.isChecked():
         Function extract.disable(false)

Function extract():
   If (statusInternal.isChekced and statusEsternal.isChecked) then
      Shell.run_shell_cmd('perintahGetSizeSumberInt')   ←   sizeFileSumberInt
      Shell.run_shell_cmd('perintahGetSizeSumberExt')   ←   sizeFileSumberExt
      ShellInt.Function()                               ←   runFunctionInt
      Time.sleep(int)
      NetcatInt.function()                              ←   runFunctionInt
      progresBarInt.function()                          ←   runFunctionInt
      Time.sleep(int)
      ShellExt.function()                               ←   runFunctionExt
      Time.sleep(int)
      NetcatExt.function()                              ←   runFunctionExt
      ProgressBarExt.function()                         ←   runFunctionExt

   Elif statusInternal.ischecked() then
      Shell.run_shell_cmd('perintahGetSizeSumberInt')   ←   sizeFileSumberInt
      ShellInt.Function()                               ←   runFunctionInt
      Time.sleep(int)
      NetcatInt.function()                              ←   runFunctionInt
      progresBarInt.function()                          ←   runFunctionInt

   Elif statusEksternal.isChecked() then
      Shell.run_shell_cmd('perintahGetSizeSumberExt')   ←   sizeFileSumberExt
      ShellExt.function()                               ←   runFunctionExt
      Time.sleep(int)
      NetcatExt.functuon()                              ←   runFunctionExt
      ProgressBarExt.function()                         ←   runFunctionExt
End
```

Gambar 3.4 *pseudocode* interface2.py

Pada Gambar 3.4 adalah *pseudocode* tampilan 2 aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*, untuk mendeskripsikan algoritmanya, dapat

diketahui bahwa algoritma di atas menunjukan bahwa ada enam *class QThread* yang berguna untuk menjalankan proses akuisisi penyimpanan ponsel android dalam aplikasi dan proses berjalannya *progress bar* untuk mengetahui berkembangnya proses akuisisi secara *real-time*, untuk penjelasan keenaam *QThread* tersebut sebagai berikut:

QThread pertama dan kedua berfungsi untuk menjalankan proses akuisisi penyimpanan yang terjadi dalam *shell* ponsel android baik itu penyimpanan internal maupun eksternal, pada *QThread* pertama dan kedua ini membutuhkan sebuah modul yang mendapatkan hak akses *shell* perangkat Android, modul tersebut adalah modul *Pyadb3* yang terdapat dalam bahasa pemograman *pyhton*, modul ini memiliki banyak fungsi dan salah satunya mendapatkan akses *shell* perangkat Android, lewat modul ini dapat menjalankan perintah yang akan membuat file berformat (.dd) dan disimpan dalam suatu protokol TCP dalam suatu port untuk keperluan proses akusisi. *QThread* ketiga dan keempat berfungsi untuk menjalankan proses akuisisi dalam laptop/*host* lewat aplikasi dengan bantuan perangkat lunak *netcat*, pada *QThread* ketiga dan keempat ini membutuhkan sebuah modul yang dapat mengakses *command prompt* pada sistem operasi *windows* , modul tersebut adalah modul *subprocess* yang terdapat dalam bahasa pemograman *python*, modul ini memiliki banyak fungsi dan salah satunya adalah *getoutput* yang dapat menghasilkan keluaran berupa nilai *string* daru suati perintah dalam *command prompt*, dengan modul ini dapat menjalankan perintah untuk menghasilkan keluaran berupa sebuah *file image* menggunakan perangkat lunak *netcat* dalam aplikasi yang diambil dari protocol tcp dalam suatu port yang terjadi diproses *QThread* pertama maupun *QThread* kedua. *QThread* kelima maupun keenam ini berfungsi untuk menjalankan *progressBar* akusisi, pada *QThread* kelima dan keenam ini membutuhkan sebuah modul yang dapat menghasilkan keluaran ukuran *file image* yang sedang berjalan pada *QThread* tiga maupun empat, modul tersebut adalah modul *os* yang terdapat dalam bahasa pemograman *python*, modul ini memiliki banyak fungsi dan salah satunya adalah *getsize* yang dapat menghasilkan keluaran ukuran pada sebuah *file* dalam suatu folder, dengan modul ini dapat menjalankan sebuah *progressBar* akuisisi dalam tampilan aplikasi.

Setelah mengetahui fungsi dari keenam *class QThread*, selanjutnya bagaimana cara menjalankan fungsi dari keenam *class QThread* di dalam *class* utama pada tampilan 2 aplikasi. Pada *pseudocode* di atas baris setelah *class QThread* adalah fungsi dalam *class* utama dan untuk mendeskripsikan algortimanya dapat dilihat bahwa di bawah keenam *class QThread* terdapat 2 fungsi utama, fungsi pertama adalah fungsi menonaktifkan tombol *"extract"*, pada fungsi ini pengguna wajib mengisi *form* untuk penamaan dan peletakan *file image* yang

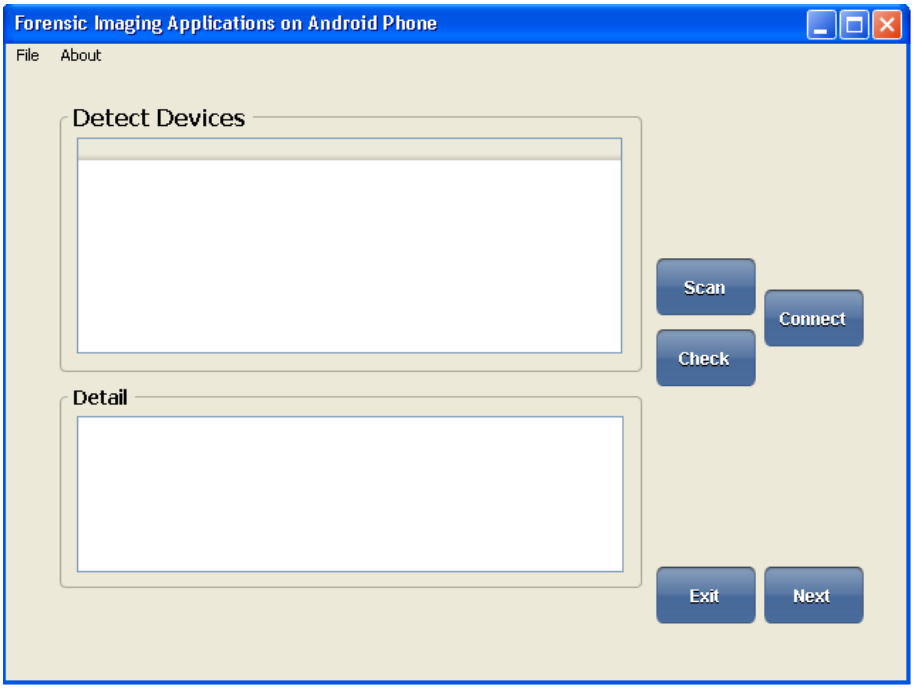
dihasilkan aplikasi pada tampilan 2 aplikasi dengan benar dan lengkap, diantaranya adalah *form file name, examiner, checkbox, radiobox,* dan *output location,* jika *form* tersebut sudah terisi dengan benar dan lengkap maka tombol "*extract"* pada aplikasi akan aktif, jika *form* tersebut salah satunya ada yang tidak lengkap maka tombol "*extract*" pada aplikasi tidak akan aktif.

Setelah fungsi pertama telah dilakukan dengan benar dan tombol "*extract*" pada aplikasi aktif maka dapat diteruskan ke fungsi kedua. Fungsi kedua adalah fungsi untuk menjalankan proses akuisisi penyimpanan Android. Pada fungsi kedua memiliki 2 kemungkinan, kemungkinan pertama adalah jika pengguna memilih kedua penyimpanan dari internal dan eksternal untuk diakuisisi maka aplikasi akan menjalankan 2 proses, proses pertama adalah melakukan proses akuisisi penyimpanan internal ponsel android terlebih dahulu yang menjalankan 3 *class QThread* yaitu *Qthread shellInt, QThread NetcatInt,* dan *QThread progressBarInt,* setelah proses pertama selesai dan berjalan maka aplikasi melanjutkan ke proses kedua adalah melakukan proses akuisisi penyimpanan eksternal ponsel android yang menjalankan 3 *class QThread* yaitu *Qthread shellExt, QThread NetcatExt,* dan *QThread progressBarExt*.

Dan kemungkinan kedua adalah jika hanya memilih antara penyimpanan internal atau eksternal untuk diakuisisi maka aplikasi akan menjalankan 1 proses, proses tersebut adalah proses akuisisi penyimpanan internal atau eksternal ponsel android dengan menjalankan 3 *class QThread* yang berfungsi mengakuisisi penyimpanan internal atau eksternal.

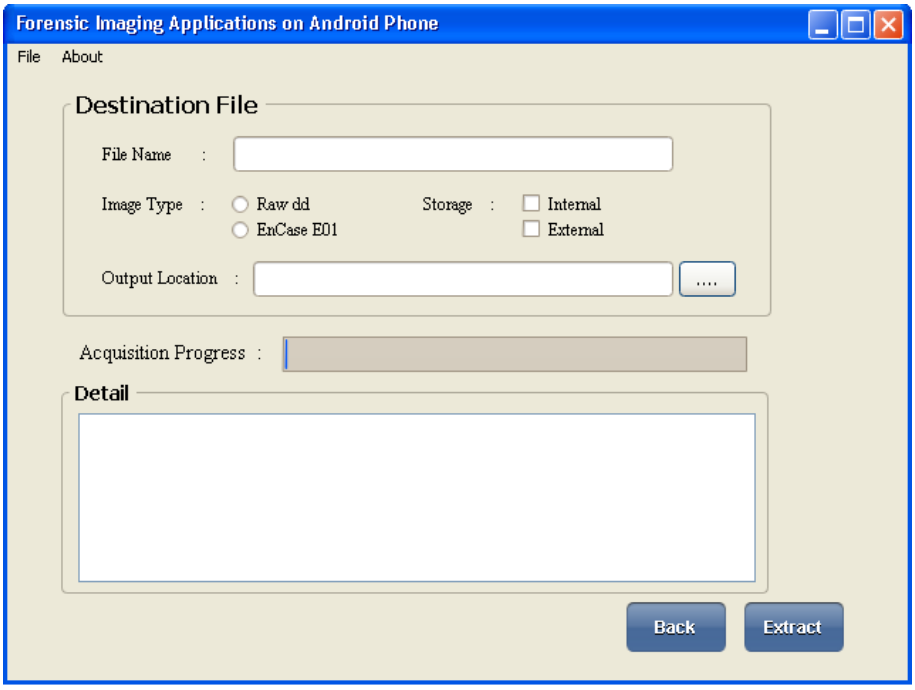
3.3.3 Desain Antarmuka

Desain antarmuka atau sering dikenal *design interface* merupakan tahap yang sangat penting untuk memberikan gambaran kepada pengguna mengenai antarmuka pada Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* yang akan dibuat. Dalam membuat antarmuka aplikasi penulis menggunakan *tools Pencil* untuk keperluan membuat rancangan tampilan aplikasi. Pada Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* memiliki 2 tampilan antarmuka. Untuk melihat rancangan antarmuka tampilan 1 aplikasi dapat dilihat pada Gambar 3.5.



Gambar 3.5 Rancangan Antarmuka Tampilan 1 Aplikasi

Pada Gambar 3.5 merupakan rancangan antarmuka tampilan 1 saat aplikasi dijalankan. Pada tampilan awal ini, pengguna diminta untuk mencari perangkat Android yang terhubung ke laptop/*host* dengan menekan tombol "Scan" pada tampilan 1 aplikasi dan aplikasi akan menampilkan sebuah *serial number* dan status perangkat Android dalam *group box* "Detect Devices", setelah menemukan perangkat yang terhubung ke laptop/*host* dan terbaca oleh aplikasi, selanjutnya pengguna diminta mengkoneksikan perangkat android dengan menekan tombol "*connect*" untuk mengkoneksikan perangkat Android ke dalam sebuah port untuk kepentingan proses akuisisi dalam tampilan 2 aplikasi. Setelah mengkoneksikan perangkat pengguna dapat mengetahui bahwa perangkat tersebut sudah terhubung atau tidaknya ada dalam *group box* "Detail" dengan menekan tombol "*check*" pada tampilan 1. Setelah proses mencari dan mengkoneksikan perangkat android telah selesai, maka pengguna dapat melanjutkan ke tampilan 2 aplikasi dengan menekan tombol "Next" pada tampilan 1 aplikasi yang dapat dilihat pada Gambar 3.6.



Gambar 3.6 Rancangan Antarmuka Tampilan 2 Aplikasi

Pada Gambar 3.6 merupakan rancangan antarmuka tampilan 2 aplikasi. Pada tampilan 2 ini, pengguna diminta mengisikan *form* dalam *group box* "Destination File" tampilan 2 aplikasi, *form* tersebut antara lain *file name* untuk penamaan *file image*, *radiobox* untuk format *file image*, *checkbox* untuk memilih penyimpanan, *output location* untuk peletakan *file image* yang akan dihasilkan. Setelah semua *form* terisi dengan benar dan lengkap maka pengguna diminta menekan tombol "*extract*" untuk menjalankan proses akusisi penyimpanan perangkat Android.

3.4 Pengujian

Setelah implementasi pengembang aplikasi, selanjutnya melakukan tahap pengujian untuk memastikan semua fungsi dalam aplikasi telah berkerja dengan baik dan benar atau belum. Pengujian ini ada tiga yaitu pengujian fungsionalitas yang terdapat dalam aplikasi, pengujian integritas data yang dihasilkan aplikasi, pengujian performa aplikasi saat menjalankan proses akuisisi, dan pengujian kuisioner kepada pihak-pihak yang mengerti bidang forensika digital dengan memberikan uraian pertanyaan mengenai aplikasi yang dijadikan sampel penelitian (*responden*), kemudian untuk perhitungan skor kuisoner menggunakan skala linkert, sedangkan untuk pengujian integritas data adalah mencocokan nilai *hash MD5* yang dihasilkan aplikasi dengan aplikasi pencocokan nilai *hash md5*.

3.4.1 Pengujian Integritas Data yang Dihasilkan.

Pada pengujian integritas data yang dihasilkan adalah pengujian yang dilakukan peneliti untuk membandingan nilai *hash Md5* dari *file image* yang dihasilkan aplikasi dengan nilai *hash md5* yang dihasilkan aplikasi pencocok. Aplikasi tersebut adalah aplikasi WinMd5 yang berguna pencocokan nilai *hash md5* dari *file image* dengan hasil *hash md5* aplikasi.

Dengan adanya pengujian ini memastikan bahwa *file image* yang dihasilkan aplikasi yang telah dibuat peneliti telah terjaga dan menjamin integritas datanya dan dapat dijadikan barang bukti yang sah diperadilan.

3.4.2 Pengujian Kuesioner Skala Linkert

Setelah implementasi pengembang aplikasi, selanjutnya melakukan tahap pengujian pengujian kuesioner kepada pihak-pihak yang mengerti bidang forensika digital dengan memberikan uraian pertanyaan mengenai aplikasi yang dijadikan sampel penelitian (*responden*), kemudian untuk perhitungan skor kuisoner menggunakan skala linkert.

Skala linkert merupakan metode pengukuran yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau kelompok orang tentang fenomena sosial (Sugiono, 2012). Perhitungan skala linkert merupakan skala yang paling banyak digunakan dalam riset berupa survey penelitian. Survey tersebut berupa kuisoner dengan menggunakan skala linkert yang diharuskan membuat pertanyaan atau penyataan yang jelas dan tidak mengandung ambigu. selanjutnya peneliti mengajak responden untuk mengungkapkan bentuk persetujuan mereka terhadap pertanyaan atau pernyataan yang telah dibuat melalui *google form* dan disebarkan ke respoden. Untuk mendapatkan hasil total skor dari kuesioner yang telah diisi oleh responden dengan menggunakan rumus (3.1), setelah itu melakukan perhitungan rata-rata skor yang didapat. Untuk mendapatkan nilai rata-rata skor dari uraian kuesioner dengan menggunakan rumus (3.2)

$$Total\ Skor = Jumlah\ Responden\ \times Pilihan\ Nilai\ Linkert \qquad (3.1)$$

$$Rata - rata = \frac{Total\ Skor}{Total\ Responden} \qquad (3.2)$$

Skala linkert memiliki 5 butir nilai skala jawaban untuk mengungkapkan persetujuan dari responden terhadap setiap pertanyaan atau pernyataan dalam kuesioner. Untuk dapat

memenuhi perhitungan nilai skala lingker, maa penulis menggunakan 5 format nilai skala jawaban pada setiap pertanyaan atau pernyataan dalam uraian kuesioner yang telah disebar ke respondel, nilai skala jawaban dapat dilihat pada Tabel 3.3.

Tabel 3.3 Nilai Skala Jawaban

Skala Jawaban	Nilai
Sangat Tidak Setuju	1
Kurang Setuju	2
Cukup Setuju	3
Setuju	4
Sangat Setuju	5

Setelah menentukan nilai dari skala jawaban linkert pada Tabel 3.3, selanjutnya melakukan perhitungan dengan menggunakan rumus Rentang Skala (RS) yaitu nilai tertinggi dikurangi nilai terendah kemudian dibagi dengan jumlah kategori jawaban. Untuk melihat rumus Rentang Skala(RS) dapat dilihat pada rumus (3.3). Dengan menggunakan rumus tersebut maka didapatkanlah hasil rentang skala yaitu 0.8. hasil tersebut didapat dengan menggunakan rumus rentang skala melalui perhitungan skor tertinggi (5) dikurangi skor terendah (1) dan dibagi jumlah katagori jawaban (5) yaitu (5-1)/5=0.8. untuk melihat hasil rentang skala yang telah dihitung dengan rumus dapat dilihat pada Tabel 3.4.

$$Rentang\ Skala = \frac{Skor\ Tertinggi - Skor\ Terendah}{Jumlah\ Kategori\ Jawaban} \qquad (3.3)$$

Tabel 3.4 Rentang Skala Linkert

Nilai	Rentang	Keterangan
1	1 – 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 3.3. dan perhitungan Rentang Skala (RS) linkert pada Tabel 3.4. maka peneliti membuat sebuah pertanyaan atau pernyataan dalam kuesioner melalui *google form* untuk dibagikan ke responden yang mengerti dalam bidang forensika digital untuk diminta mengungkapkan persetujuan mereka terhadap hasil penelitian yang telah dibuat peneliti. Untuk uraian kuesioner yang telah dibuat peneliti dapat dilihat pada Tabel 3.5.

Tabel 3.5 Uraian Kuesioner

No	**URAIAN**	**SKOR**				
A.	**Manfaat**	**1**	**2**	**3**	**4**	**5**
1.	Aplikasi *Forensic Imaging* ponsel android dengan memanfaatkan *Custom Recovery* ini memiliki manfaat bagi saya yang bergelut di bidang forensic ?					
2.	Penggunaan *Team Win Recovery Project*(TWRP) pada perangkat Android untuk proses akuisisi mampu menekan perubahan intergitas data yang dihasilkan?					
3.	Aplikasi *Forensic Imaging* Android dengan memanfaatkan *Custom Recovery* ini mudah digunakan?					
4.	Menggunakan aplikasi ini ketika ingin melakukan cloning penyimpanan Android?					
5.	Saya berharap aplikasi ini dapat digunakan orang banyak?					
6.	Keluaran yang dihasilkan 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 mudah untuk dikenali?					
3.	Menu yang ditampilkan mudah untuk dipahami?					
4.	Peletakan semua fitur dan tombol aplikasi sudah sesuai?					
C.	**Fungsionalitas**	**1**	**2**	**3**	**4**	**5**
1.	Fitur untuk flashing file TWRP berjalan dengan semestinya?					

No	URAIAN	SKOR				
2.	Fitur untuk scanning perangkat android berjalan dengan baik?					
3.	Fitur untuk mengkoneksikan perangkat android berjalan dengan baik?					
4.	Fitur untuk proses akuisisi penyimpanan perangkat android berjalan dengan baik?					
5.	Semua tombol dan menu dapat bekerja dengan normal?					

BAB IV
HASIL DAN PEMBAHASAN

Bab ini akan menjelaskan tentang hasil dan pembahasan dari Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* yang telah dibuat oleh peneliti.

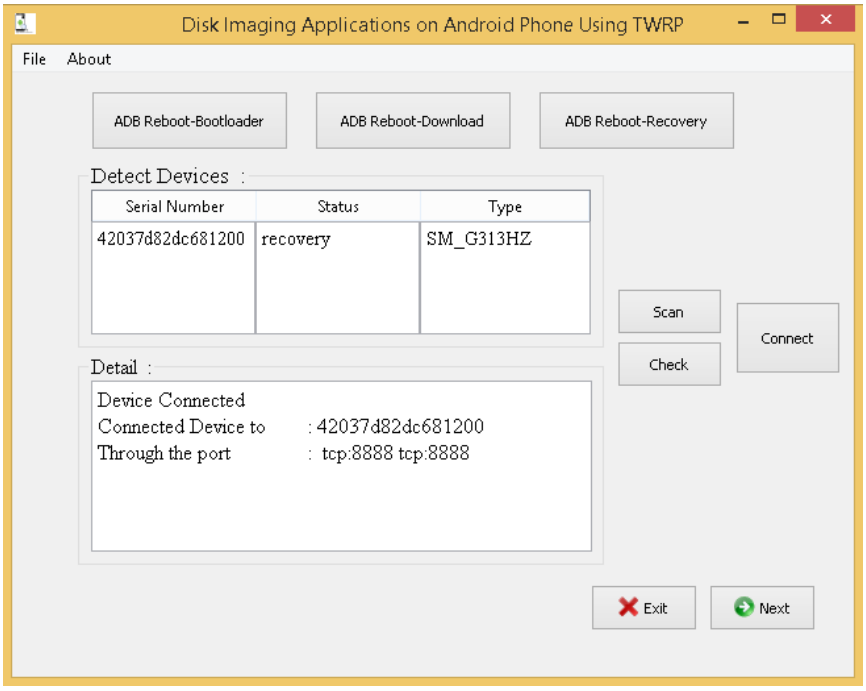
4.1 Hasil

4.1.1 Hasil Pembuatan Aplikasi

Hasil yang didapat adalah sebuah aplikasi *Forensic Imaging* pada ponsel android dengan memanfaatkan *custom recovery* yang dapat mengakuisisi penyimpanan ponsel android baik internal maupun eksternal dan juga dapat memantau perkembangan proses akuisisi dengan menampilkan *progress bar* dalam tampilan aplikasi. Hasil implementasi aplikasi menjelaskan aplikasi ini memiliki 2 tampilan dapat dijelaskan sebagai berikut:

a. Hasil Tampilan 1 Aplikasi

Hasil dari implementasi salah satunya adalah tampilan 1 aplikasi merupakan tampilan awal saat dibukanya aplikasi yang sudah dirancang dengan bantuan tool *Pencil* dan mengalami perubahan sedikit dalam tombol bantuan untuk kepentingan instalasi TWRP. Untuk melihat hasil tampilan 1 aplikasi dapat dilihat pada Gambar 4.1.

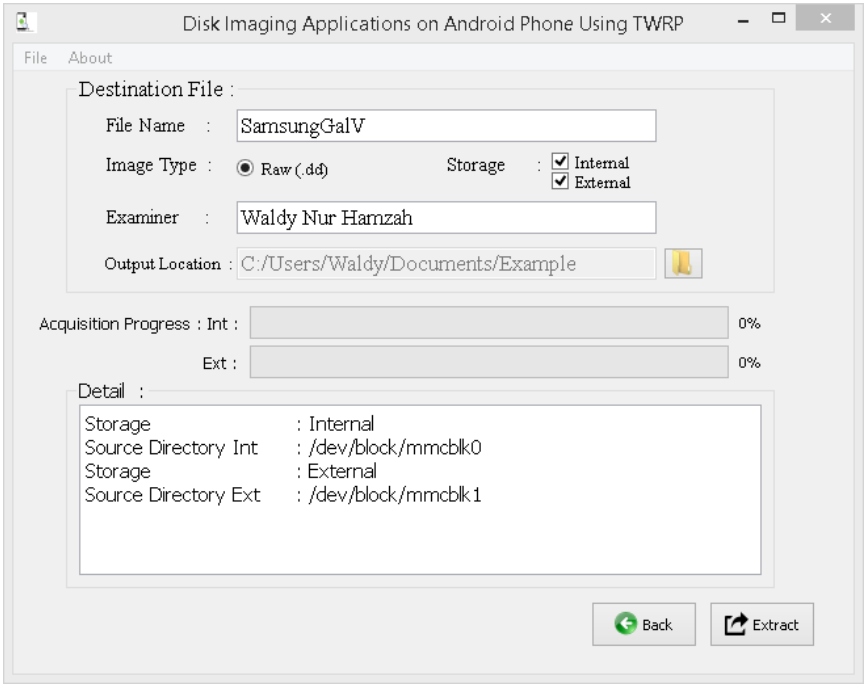


Gambar 4.1 Hasil Tampilan 1 Aplikasi

Pada Gambar 4.1 merupakan Tampilan 1 aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*, tampilan tersebut adalah tampilan awal saat membuka aplikasi. Dalam tampilan 1 aplikasi terdapat beberapa fungsi, fungsi pertama dapat melakukan proses instalasi *custom recovery* TWRP dalam aplikasi dengan tombol bantuan. Fungsi kedua yaitu melakukan proses pencarian (*scanning*) perangkat android yang terhubung dengan laptop/*host* yang akan ditampilan dalam tampilan 1 aplikasi. Sedangkan fungsi ketiga yaitu mengkoneksikan perangkat/ponsel android yang sudah terbaca oleh aplikasi untuk melanjutkan ke tahap selanjutnya yaitu proses akuisisi penyimpanan android dalam tampilan 2 aplikasi.

b. Hasil Tampilan 2 Aplikasi.

Hasil dari implementasi setelah tampilan 1 adalah tampilan 2 aplikasi merupakan tampilan akhir dari aplikasi yang telah dibuat dan dirancang dengan bantuan *tool Pencil*, namun pada tampilan 2 aplikasi ditambahkan dengan adanya 2 *progress bar* untuk mengetahui perkembangan proses akuisisi yang sedang berjalan dalam aplikasi secara *real-time*. Untuk melihat hasil dari tampilan 2 aplikasi dapat dilihat pada Gambar 4.2.



Gambar 4.2 Hasil Tampilan 2 Aplikasi

Pada Gambar 4.2 merupakan tampilan 2 aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*, tampilan tersebut adalah tampilan akhir

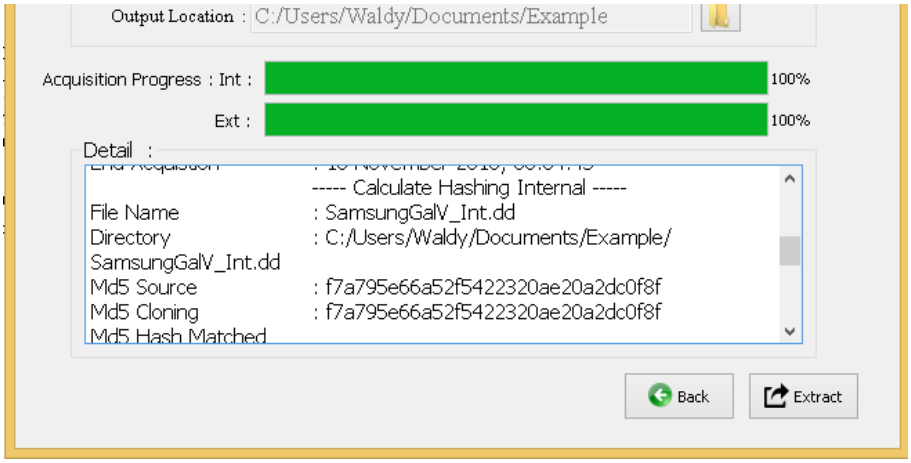
dari aplikasi. Dalam tampilan 2 aplikasi memiliki beberapa fungsi. Fungsi pertama yaitu dapat melakukan proses akuisisi penyimpanan internal maupun eksternal ponsel android yang akan menghasilkan keluaran *file image* sebagai objek analisis. Sedangkan fungsi kedua yaitu melakukan proses *hashing md5* dari *file image* yang dihasilkan aplikasi dengan file sumber penyimpanan dalam ponsel android yang akan dijadikan sebagai bukti bahwa aplikasi dapat menjaga integritas data saat dilakukan proses akuisisi, proses *hashing md5* dilakukan setelah proses akuisisi penyimpanan ponsel android dalam aplikasi telah selesai. Setelah kedua fungsi telah berjalan dan selesai dengan baik maka aplikasi akan menghasilkan 3 keluaran yaitu informasi dalam aplikasi mengenai proses akuisisi yang telah dialkukan dalam aplikasi dan beberapa file. File tersebut antara lain: *file image* dan *file log* sebagai keluaran fisik yang dapat dijadikan objek analisis untuk mencari informasi di dalamnya.

4.1.2 Hasil Proses Akuisisi Dari Aplikasi

Hasil yang didapat dari melakukan proses akuisisi penyimpanan ponsel android baik itu internal maupun eksternal dalam Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* adalah 3 hasil yang didapat, diantaranya adalah informasi dalam aplikasi mengenai proses akuisisi yang telah dilakukan dalam aplikasi dan menghasilkan beberapa file, file tersebut berupa *file image* dan *file log*. Untuk penjelasan 3 hasil yang didapat tersebut sebagai berikut:

a. Informasi Dalam Aplikasi.

Hasil yang didapat setelah melakukan proses akuisisi dalam aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* salah satunya adalah informasi mengenai proses akuisisi yang telah dilakukan dalam aplikasi. Informasi tersebut diantaranya adalah nama penyimpanan yang diakuisisi seperti internal atau eksternal, direktori sumber penyimpanan dalam ponsel android, ukuran penyimpanan baik internal maupun eksternal, waktu mulai dan selesai proses akuisisi penyimpanan, nama *file image* yang dihasilkan, rincian direktori peletakan *file image* dalam suatu folder perangkat komputer, dan informasi hasil proses *hashing md5* dari file sumber penyimpanan ponsel android dengan *file image* yang dihasilkan aplikasi. Untuk melihat hasil informasi dalam aplikasi dapat dilihat pada Gambar 4.3.

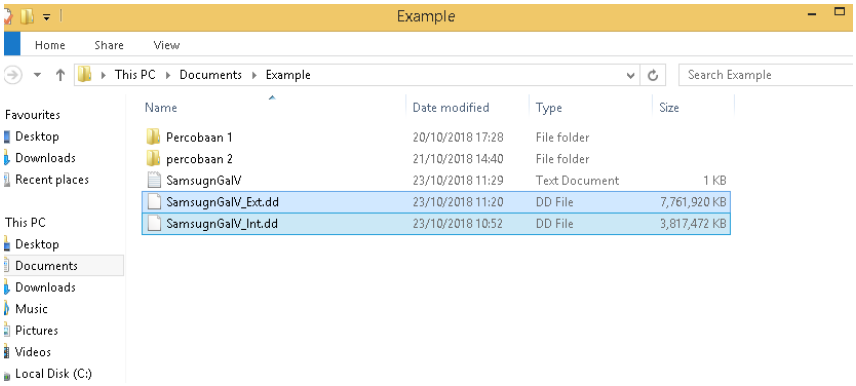


Gambar 4.3 Hasil Informasi Dalam Aplikasi

b. *File Image* atau *Disk Image*.

Hasil kedua yang didapatkan setelah melakukan proses akuisisi penyimpanan ponsel android baik itu internal maupun eksternal dalam aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* adalah berupa *file image* atau sering dikenal *disk image* yang berformat (.dd), *file image* adalah sebuah file penyimpanan yang berisikan semua data-data yang tersimpan dalam sumber penyimpanan ponsel android baik itu internal maupun eksternal yan didapatkan setelah melakukan proses akuisisi penyimpanan ponsel android dalam aplikasi yang telah selesai, dari *file image* tersebut dapat dilakukan analisis secara menyeluruh untuk mencari informasi-informasi yang terdapat dalam *file image* yang dihasilkan dan mencari informasi berharga yang akan dijadikan sebuah barang bukti yang sah untuk proses peradilan dengan menggunakan aplikasi forensika digital saat ini, seperti aplikasi Autopsy dan lain-lain.

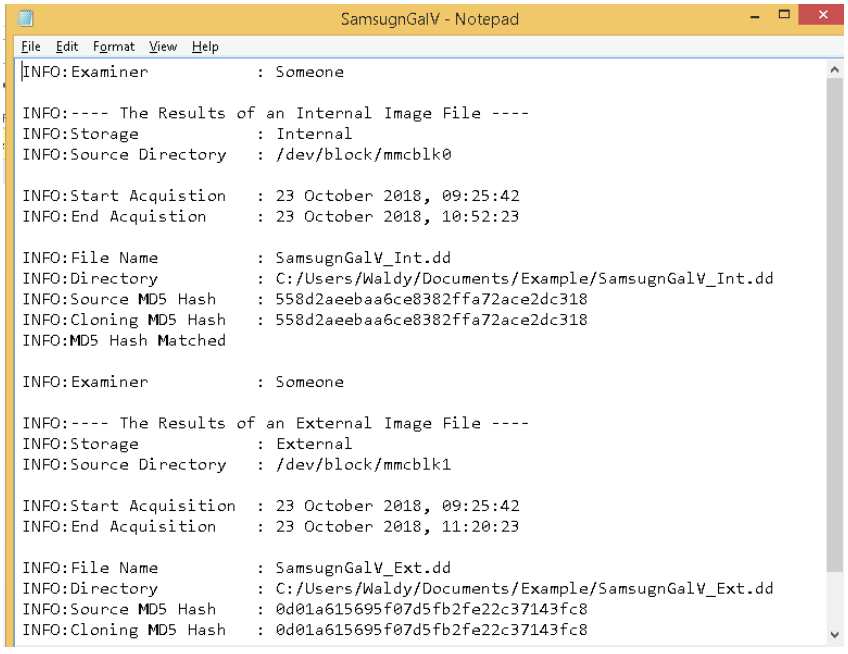
Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* akan menghasilkan 2 *file image* dari penyimpanan internal dan eksternal ponsel android, jika pengguna melakukan proses akuisisi kedua penyimpanan secara bersamaan dalam aplikasi, namun jika pengguna hanya memilih salah satu penyimpanan yang diakuisisi dalam aplikasi maka *file image* yang dihasilkan hanya 1 *file image* dari salah satu penyimpanan baik itu internal maupun eksternal. Untuk melihat 2 *file image* yang dihasilkan aplikasi saat melakukan proses akuisisi kedua penyimpanan secara bersamaan dapat dilihat pada Gambar 4.4.



Gambar 4.4 Hasil File Image

c. *File Log*.

Hasil ketiga atau hasil terakhir yang didapat setelah proses akuisisi dalam aplikasi dalan berupa *file log*, *file log* adalah file yang berisikan informasi-informasi mengenai proses akuisisi yang dilakukan aplikasi, informasi tersebut sama dengan informasi yang dihasilkan dalam aplikasi namun diambil yang terpenting saja. Seperti nama pengguna aplikasi (*Examiner*), nama penyimpanan (*storage*) baik itu internal maupun eksternal, sumber direktori penyimpanan ponsel android, waktu mulai dan selesai proses akuisisi, nama *file image* yang dihasilkan aplikasi, direktori peletak *file image* tersebut, dan hasil *hashing md5* dari *file image* dengan file sumber penyimpanan. Untuk melihat rincian *file log* yang dihasilkan aplikasi dapat dilihat pada Gambar 4.5.



Gambar 4.5 Hasil Rincian File Log

4.1.3 Hasil Pengujian Kuesioner Skala Linkert Aplikasi

Pada tahap ini adalah menjelaskan hasil yang didapat dari pengujian kuesioner Skala linkert kuesioner yang telah dilakukan peneliti dan disebarkan ke 50 responden melalui *google form* untuk menanggapi terhadap Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* dan yang dapat menentukan hasil dari kusioner dengan menggunakan rumus Rentang Skala yang dapat dilihat pada rumus (3.3). namun untuk menghitung Rentang Skala, pertama yang dilakukan adalah mendapatkan total skor dari setiap pertanyaan atau pernyataan dalam kuesioner dengan menggunakan rumus (3.1), setelah itu mencari nilai rata-rata dari setiap pertanyaan dalam kuesioner dengan menggunakan rumus (3.2). setelah mendapatkan nilai rata-rata dari setiap pertanyaan dalam kuesioner yang dapat dilihat pada Tabel 4.5, selanjutnya dapat menentukan hasil dari kuesioner yang dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil Pengujian Kuesioner Skala Linkert

No	**Aspek**	**Nomer Pertanyaan**	**Nilai Rata-rata Setiap Pertanyaan**	**Hasil Nilai Rata-rata**	**Keterangan**
1.	Manfaat	1,2,3,4,5,6	(4,28 + 3,96 + 4,14 + 4,1 + 4,36 + 4,2)/6	4,17	Setuju
2.	Tampilan	1,2,3,4,	(4 + 3,96 + 4,04 + 4,04)/4	4,01	Menarik
3.	Fungsionalitas	1,2,3,4,5	(4,24 + 4,16 + 4,24 + 4,2 + 4,36)/5	4,24	Baik
Total Rata-rata Keseluruhan				4.12	Bermanfaat

Setelah kuesioner yang didapat dari 50 responden yang ditujukan ke seorang yang mengerti bidang forensika digital, kemudian data telah dikelola dengan menggunakan perhitungan skala linkert yang dapat dilihat pada Tabel 4.1 .Dari hasil pengujian ini dapat disimpulkan dalam aspek manfaat penggunaan aplikasi mendapatkan rata-rata nilai jawaban sebasar 4,17/5,00 yang dikategorikan ke dalam “setuju”, sedangkan dalam aspek tampilan aplikasi mendapatkan rata-rata nilai jawab sebesar 4,01/5,00 yang dikategorikan ke dalam “menarik”, dan dalam aspek fungsionalitas aplikasi mendapatkan rata-rata nilai jawaban sebesar 4,24/5,00 yang dikategorikan “baik”. Dari perhitungan semua aspek yang terkait dalam

kuesioner, selanjutnya peneliti melakukan perhitungan dengan menggunakan rumus rentang skala (RS) linkert untuk mendapatkan perhitungan rata-rata dari semua aspek yang terkait dalam kuesinoner dan mendapatkan kesimpulan bahwa aplikasi bermanfaat untuk digunakan oleh seorang forensika digital.

4.1.4 Hasil Pengujian Aplikasi oleh Pakar

Dalam pengujian aplikasi oleh pakar ini adalah melakukan pengujian aplikasi yang ditujukan kepada pakar forensika digital untuk menanggapi suatu kuesioner yang telah dibuat oleh peneliti terhadap aplikasi. Pakar forensika digital tersebut bernama bapak Yudi Prayudi, S.Si., M.Kom. Beliau merupakan seorang dosen Teknik Informatika di FTI Univertias Islam Indonesia dan beliau juga seorang ketua dalam organisai bernama Pusat Studi Forensika Digital (PUSFID) di FTI Universitas Islam Indonesia. Adapun peneliti memberikan uraian pertanyaan-pertanyaan yang penulis ajukan kepada pakar forensika digital seputar aplikasi yang diujikan dan disertai hasil dari kuesioner pengujian aplikasi oleh pakar. Berikut ini adalah uraian pertanyaannya dan juga hasilnya :

Tabel 4.2 Pertanyaan Kuesioner ke Pakar.

No	**URAIAN**	**SKOR**				
A.	**Manfaat**	**1**	**2**	**3**	**4**	**5**
1.	Apakah aplikasi ini memiliki manfaat bagi seorang dalam bidang forensika digital?					✓
2.	Apakah aplikasi ini sudah dapat memfasilitasi dalam melakukan proses akuisisi penyimpanan dengan memanfaatkan custon recovery?					✓
3.	Apakah dengan metode memanfaatkan Custom Recovery mudah digunakan?				✓	
4.	Apakah aplikasi ini dapat dengan mudah melakukan proses akuisisi penyimpanan ponsel android?					✓
5.	Apakah keluaran aplikasi ini berupa file image dan file log sudah dapat menjadi barang bukti ?				✓	
B.	**Tampilan**	**1**	**2**	**3**	**4**	**5**

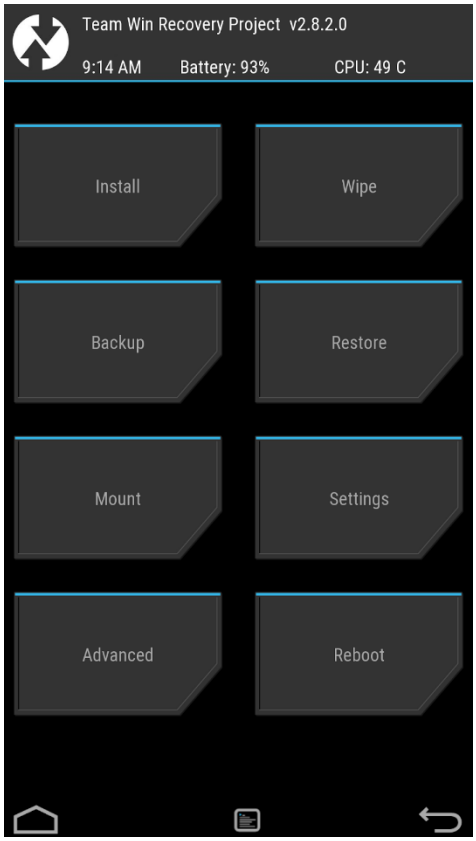
No	URAIAN	SKOR				
1.	Apakah tampilan antarmuka aplikasi ini sudah user-friendly?				✓	
2.	Apakah fitur-fitur untuk instalasi Custom recovery sesuai kebutuhan?				✓	
3.	Apakah tombol dan menu yang ditampilkan mudah untuk dipahami?				✓	
Total Rata-rata Keseluruhan		4,60 + 4 = 8,60/2 = 4,30				
		Sangat Bermanfaat				

Setelah pertanyaan kuesioner yang telah dibuat peneliti telah ditanggapi atau dijawab oleh seorang pakar bidang forensika digital, kemudian data tersebut dikelolah dengan menggunakan perhitungan skala linkert yang dapat dilihat pada Tabel 4.2. Dari hasil pengujian ini dapat disimpulkan dari kedua aspek dalam pertanyaan kuesioner, mendapatkan hasil dari aspek manfaat penggunaan aplikasi nilai jawaban sebesar 4,60/5,00 yang dikategorikan ke dalam "Sangat Setuju", sedangkan dalam aspek tampilan dalam aplikasi mendapatkan nilai jawab sebesar 4/5,00 yang dikategorikan "Menarik". Dari perhitungan kedua aspek kuesioner tersebut, selanjutnya dilakukan perhitungan rentang skala (RS) linkert dengan mendapatkan nilai jawaban sebesar 4,30/5 dengan nilai tersebut dapat disimpulkan "Sangat Bermanfaat". Kesimpulan tersebut menyatakan bahwa aplikasi "Sangat Bermanfaat" untuk digunakan oleh seorang forensika digital.

4.2 Pembahasan Implementasi

4.2.1 *Custom Recovery*

Tahap ini menjelaskan tentang fungsi dari *custom recovery* dalam ponsel android untuk pengembangan aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*. *Custom recovery* yang digunakan peneliti adalah *Team Win Recovery Project* (TWRP) yang memiliki banyak kegunaan. Untuk melihat tampilan *custom recovery* TWRP dalam ponsel android dapat dilihat pada Gambar 4.6.



Gambar 4.6 *Custom Recovery* TWRP

Pada Gambar 4.6 merupakan tampilan dalam *custom recovery* TWRP yang terdapat dalam ponsel android untuk kepentingan proses akuisisi penyimpanan. *Custom recovery* TWRP memiliki banyak fungsi diantaranya adalah untuk instalasi kustom ROM, membersihkan partisi dalam ponsel android, melakukan *backup* data, mengembalikan file yang telah di *backup*, dan salah satu fungsinya ialah mendapatkan izin *permission partisi* seperti halnya hak akses *root* yang digunakan peneliti untuk membuat Aplikasi *Forensic Imaging* pada ponsel Android dengan memanfaatkan *Custom Recovery* untuk mengakuisisi penyimpanannya. Namun untuk dapat menggunakan *custom recovery* TWRP diharuskan untuk menginstal di ponsel android melalui *flashing file recovery* TWRP dengan mencocokan *file recovery* yang telah di unduh dengan ponsel yang digunakan untuk proses akuisisi penyimpanan.

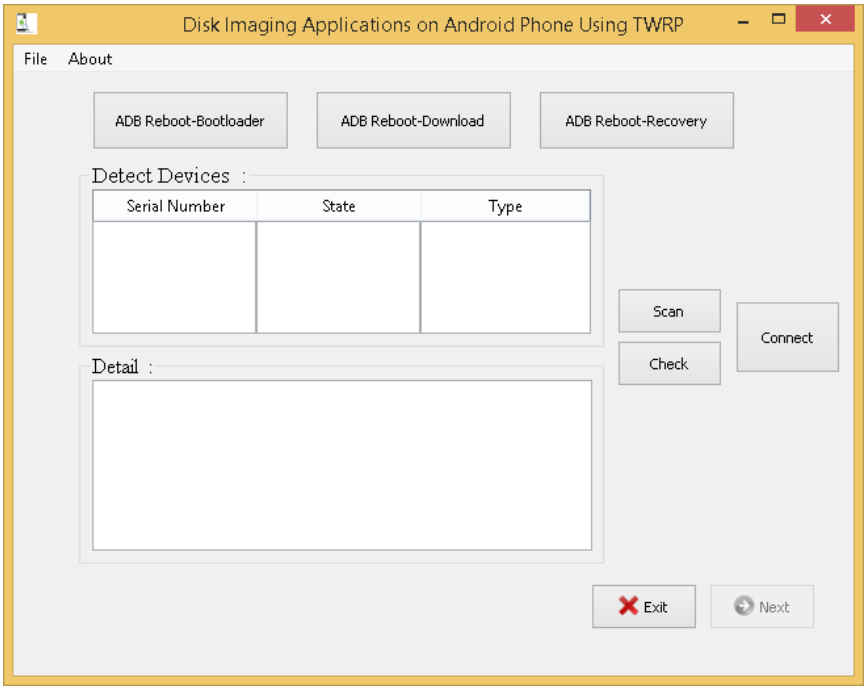
4.2.2 Implementasi Antarmuka

Tahap ini adalah menjelaskan tampilan antarmuka yang terdapat dalam aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* yang telah dibuat. Pada aplikasi ini memiliki 2 tampilan antarmuka, yaitu tampilan 1 dan tampilan 2 aplikasi yang

berguna untuk melakukan proses akuisisi penyimpnana ponsel android. Untuk penjelasan kedua antarmuka tersebut sebagai berikut:

a. Antarmuka tampilan 1 aplikasi.

Pada antarmuka tampilan 1 berfungsi untuk menjalankan proses pencarian ponsel android (*Scanning*) dan mengkoneksikan ponsel android dalam tampilan1 aplikasi untuk kepentingan melakukan proses selanjutnya yaitu proses akuisisi penyimpanan dalam tampilan 2. Untuk tampilan 1 aplikasi dapat dilihat pada Gambar 4.7.



Gambar 4.7 Antarmuka Tampilan 1 Aplikasi

Pada Gambar 4.7 merupakan antarmuka tampilan 1 Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* yang terdapat beberapa menu, tombol dan 2 *group box* yang berguna untuk proses pencarian ponsel android dan mengkoneksikan ponsel tersebut melalui aplikasi. Untuk penjelasan dan fungsi masing-masing sebagai berikut:

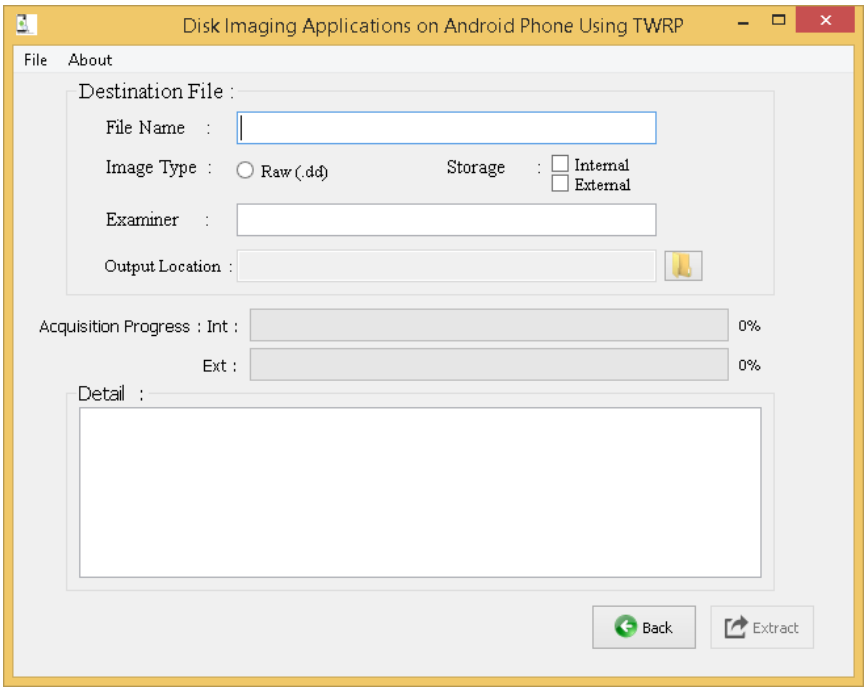
1. Menu “File”, dari menu file memiliki sub-menu yaitu “scan” dan “exit”. Scan berfungsi untuk mencari ponsel Android yang terhubung dengan laptop/*host* yang akan ditampilkan dalam *group box* “Detect Devices” sedangkan exit berfungsi untuk keluar atau menutup aplikasi.

2. Menu “About” berfungsi menampilan suatu *dialog box* yang berisikan informasi tentang cara-cara penggunaan aplikasi.
3. Tombol “ADB reboot-bootloader” adalah tombol bantuan dari aplikasi untuk ponsel Android menuju *bootloader mode* untuk keperluan *flashing file recovery* ke dalam ponsel android secara manual.
4. Tombol “ADB reboot-download” adalah tombol bantuan dari aplikasi untuk ponsel Android menuju *download mode*, biasanya *download mode* hanya dimiliki oleh ponsel Android keluaran *Samsung*. Tombol ini berfungsi untuk *flashing file recovery* ke dalam ponsel Android dengan mudah menggunakan bantuan *tool odin.*
5. Tombol “ADB reboot-recovery” adalah tombol bantuan dari aplikasi untuk ponsel Android menuju *recovery mode* yang sudah ter-*install custom recovery* TWRP.
6. *Group box* “Detect Device” memiliki beberapa *form* yaitu *serial number, state,* dan *type.* Pada *form serial number* berfungsi menampilkan sebuah *serial number* perangkat Android yang terbaca/ditemukan oleh aplikasi melalui tombol “Scan”.
7. Tombol “Scan” adalah berfungsi untuk menjalankan proses *scanning* perangkat Android yang terhubung dengan laptop/*host* dan akan menghasilkan keluaran berupa *serial number, state,* dan *type* ke dalam *group box* “Detect Device”.
8. Tombol “Connect” berfungsi untuk menjalankan proses mengkoneksikan perangkat Android ke dalam suatu port.
9. Tombol “Check” berfungsi untuk menampilkan informasi tentang perangkat Android yang terkoneksi dari *serial number* maupun informasi *port* yang digunakan ke dalam sebuah *group box* “Detail”.
10. *Group box* “Detail” berfungsi untuk menampilkan informasi tentang perangkat Android yang terkoneksi, informasi *serial number* maupun informasi *port* yang digunakan.
11. Tombol “Exit” berfungsi untuk keluar atau menutup aplikasi.
12. Tombol “Next”, pada tombol “Next” akan aktif bila langkah – langkah untuk mengkoneksikan perangkat telah selesai, sehinnga tombol “next” dapat berfungsi untuk melanjutkan ke antarmuka tampilan 2 aplikasi.

b. Antarmuka Tampilan 2 aplikasi

Pada antarmuka tampilan 2 berfungsi untuk menjalankan proses akuisisi penyimpanan ponsel android baik itu internal maupun eksternal dan menjalankan proses

hashing md5 dari *file image* yang dihasilkan dari proses akuisisi dengan file sumber penyimpanan ponsel android, untuk melihat antarmuka tampilan 2 aplikasi dapat dilihat pada Gambar 4.8.



Gambar 4.8 Antarmuka Tampilan 2 Aplikasi

Pada Gambar 4.8 merupakan antarmuka tampilan 2 pada Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* yang terdapat beberapa tombol dan 2 group box. *Group box* pertama adalah "Destination File" terdapat *form* yang terdiri beberapa kolom untuk *file image* yang akan dihasilkan aplikasi. Kolom "File Name" berfungsi untuk penamaan *file image* yang akan dihasilkan, kolom "Image Type" berfungsi menentukan format dari *file image*, kolom "Storage" berfungsi untuk menentukan penyimpanan apa yang ingin diakuisisi dan dapat memilihi kedua penyimpanan, kolom "Examiner" berfungsi untuk siapa nama pengguna aplikasi saat melakukan proses akuisisi, kolom "Output Location" berfungsi untuk peletakan *file image* yang akan dihasilkan aplikasi dapat ditentukan pada tombol yang sudah disiapkan untuk menentukan tata letak penyimpanan *file image* yang akan dihasilkan aplikasi. Sedangkan *group box* kedua adalah "Detail" yang berfungsi untuk menampilkan informasi mengenai proses akuisisi penyimpanan ponsel android yang telah dilakukan dalam aplikasi.

Selain kolom-kolom yang berguna untuk *file image* terdapat juga tombol-tombol yang memiliki fungsi masing-masing. Tombol "Back" berfungsi untuk kembali ketampilan sebelumnya, sedangkan tombol "Extract" berfungsi untuk menjalankan proses akuisisi penyimpanan ponsel android dalam aplikasi. Sedangkan melihat perkembangan proses akuisisi yang sedang berjalan dapat dilihat pada kolom "Acquisition Progress" aplikasi.

4.2.3 Implementasi Kode Program Proses *Scanning* Ponsel Android

Proses pertama yang harus dilakukan saat aplikasi berjalan adalah melakukan proses *scanning* perangkat Android yang terhubung ke laptop/*host* yang akan dibaca oleh aplikasi, proses ini menggunakan modul *subprocess* dalam bahasa pemograman *python* melalui bantuan *tool* ADB Deamon. Kode program untuk proses *scanning* perangkat Android dapat dilihat pada Gambar 4.9.

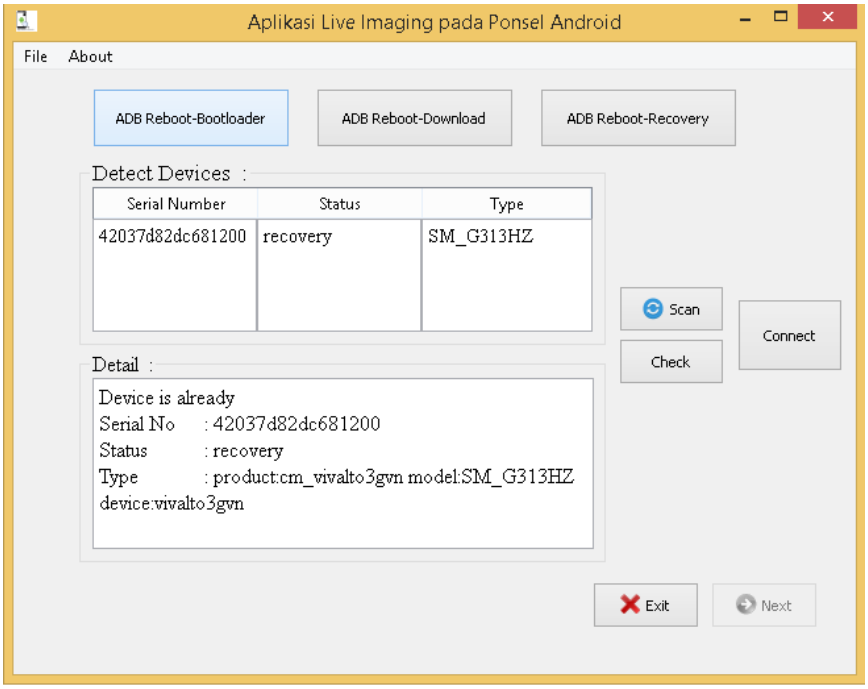
```
142. Def clk_Scan(self):
143.
144.     Subprocess.getoutput('adb start-server')
145.
146.     cekDevice = subprocess.getoutput('adb devices')
147.     SerialNo = subprocess.getoutput('adb get-serialno')
148.     State = subprocess.getoutput('adb get-state')
149.
150.     Self.columnSer.setText(serialNo)
151.     Self.columnStat.setText(state)
152.     Self.columnType.setText(cekDevice)
153.
154.     If serialNo == 'unknown':
155.         Self.textDetail.setPlainText('Device Not Found')
156.     Else:
157.         Self.textDetail.setPlainText('Device is already')
158.         Self.textDetail.append('Serial No :' +serialNo)
159.         Self.textDetail.append('State :' +state)
160.         Self.textDetail.append('Type : ' +cekDevice)
```

Gambar 4.9 Kode Program Proses *Scanning* Perangkat Android

Pada Gambar 4.9 merupakan kode program untuk proses *scanning* perangkat android yang terjadi dalam tampilan 1 aplikasi. Pada kode program baris 142 merupakan fungsi yang akan melakukan proses *scanning* setelah menekan tombal "scan". Selanjutnya, kode program baris 144 merupakan fungsi untuk membuka *server* awal pencarian perangkat. Pada kode baris 146 hingga 148 merupakan suatu variabel yang menyimpan nilai *string* yang dihasilkan dari perintah yang dilakukan. Kode program baris 146 terdapat variabel "cekDevice" yang menyimpan nilai *string* mengenai sebuah perangkat, pada baris 147 terdapat variabel

"serialNo" yang menyimpan nilai *string* mengenai *serial number* perangkat Android, dan baris 148 terdapat variabel "state" yang menyimpan nilai *string* mengenai status perangkat Android.

Selanjutnya, pada kode program baris 150 hingga 152 merupakan keluaran yang bernilai *string* ke dalam suatu kolom pada aplikasi, kode program baris 150 hingga 152 menjelaskan aplikasi akan memberikan keluaran informasi mengenai *serial number*, *state*, dan *type* dalam kolom masing-masing dalam *group box* "Detect Devices" pada aplikasi. Pada kode program baris 154 hingga 160 menjelaskan kondisi perangkat Android, pada baris 154 jika memberikan hasil "unknown" dari suatu fungsi pada baris 147 maka aplikasi akan memberikan informasi bahwa tidak ada perangkat Android yang ditemukan, dan sebaliknya, jika memberikan hasil selain kata "unknown" maka aplikasi akan memberikan informasi bahwa ada perangkat yang ditemukan dan menghasilkan keluaran informasi mengenai perangkat tersebut. untuk melihat hasil proses *Scanning* perangkat yang ditemukan aplikasi dapat dilihat pada Gambar 4.10.



Gambar 4.10 Hasil Proses *Scanning* Perangkat Android

4.2.4 Implementasi Kode Program Proses Mengkoneksikan Ponsel Android

Setelah melakukan proses *scanning* perangkat Android dan menemukan perangkat, maka selanjutnya melakukan proses mengkoneksikan perangkat Android yang akan dilakukan aplikasi. Proses ini menggunakan modul *subprocess* dalam bahasa pemograman *python*

melalui *tool* ADB Deamon. Kode program untuk proses mengkoneksikan perangkat Android dapat dilihat pada Gambar 4.11.

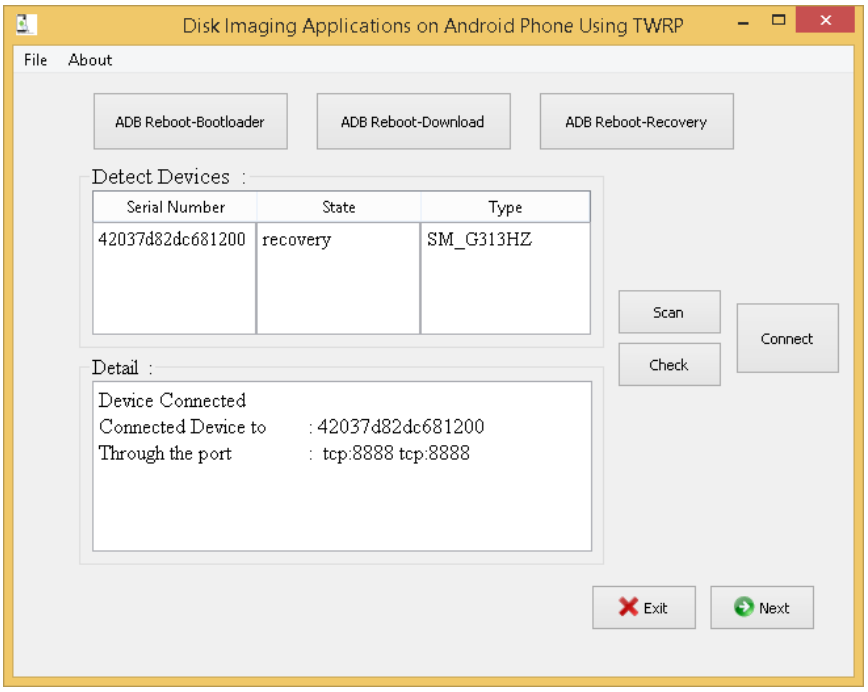
```
180.  Def clk_con(self):
181.      cekDevice = subprocess.getoutput('adb get-state')
182.
183.      if cekDevice == 'recovery':
184.          conDevice = subprocess.getoutput('adb forward tcp:8888 tcp:8888')
185.          if conDevice == '':
186.              self.textDetail.setPlainText("Device Connected")
187.              self.pushNext.setDisabled(False)
188.      elif cekDevice == 'unknown':
189.          QMessageBox.about(self, 'caution', 'Device Not Found')
190.          Self.textDetail.setPlainText("Device Not Connected")
191.      elif cekDevice == 'device':
192.          QMessageBox.about(self, 'caution', 'Device not in Recovery Mode')
```

Gambar 4.11 Kode Program Proses Mengkoneksikan Perangkat Android

Pada Gambar 4.11 merupakan kode program untuk melakukan proses mengkoneksikan ponsel Android yang terjadi dalam tampilan 1 aplikasi. Pada kode program baris 180 merupakan fungsi yang akan menjalankan proses mengkoneksikan ponsel android setelah menekan tombol "connect", kode program baris 181 merupakan variabel yang menyimpan nilai *string* yang dihasilkan dari suatu perintah untuk mengetahui keadaan ponsel Android. Selanjutnya, pada kode program baris 183 hingga 192 menjelaskan status kondisi dan menghubungkan ponsel Android. Pada kode program baris 183 hingga 187 menjelaskan jika status ponsel android dalam keadaan "recovery" yang dihasilkan dari fungsi kode program baris 181 maka proses mengkoneksikan perangkat Android akan dilakukan dalam kode program baris 184, kode program baris 184 merupakan fungsi mengkoneksikan ponsel android yang akan disimpan dalam variabel "conDevices" dan kode program baris 185 menjelaskan jika variabel tersebut sudah berjalan maka aplikasi akan memberikan informasi bahwa ponsel android sudah terkoneksi dan akan mengaktifkan tombol "next" untuk melanjutkan ke proses akuisisi penyimpanan ponsel android pada tampilan 2 aplikasi.

Selanjutnya pada kode program baris 188 hingga 190 menjelaskan jika status ponsel android dalam keadaan "unknown" atau tidak ditemukan yang dihasilkan dari fungsi kode program baris 181, maka aplikasi akan memberikan informasi berupa *text box* bahwa ponsel android tidak ditemukan atau ponsel tersebut memang tidak terhubung ke dalam perangkat laptop , sedangkan pada kode program baris 191 hingga 192 menjelaskan jika status ponsel android dalam keadaan "device" yang dihasilkan dari fungsi kode program baris 181 maka aplikasi akan memberikan informasi berupa *text box* bahwa ponsel android tidak dalam

recovery mode dan diwajibkan melakukan instalasi *custom recovery*. Untuk melihat hasil proses mengkoneksikan ponsel Android yang ditemukan aplikasi dapat dilihat pada Gambar 4.12.



Gambar 4.12 Hasil Proses Mengkoneksikan Perangkat Android

4.2.5 Implementasi Kode Program Proses Akuisisi dan *Hashing Md5*

Setelah semua proses yang terjadi di tampilan 1 aplikasi sudah berjalan maka, selanjutnya melakukan proses akuisisi penyimpanan ponsel Android yang sudah terkoneksi dengan aplikasi. Proses ini menggunakan 2 modul yang terdapat dalam bahasa pemograman *python* dan *tool* untuk melakukan proses pembuatan dan keluaran *file image*. Modul tersebut adalah modul *pyadb3* dan *subprocess* sedangkan *tool* yang digunakan adalah *Netcat*. Kode program untuk proses akuisisi dan *hashing md5* penyimpanan ponsel Android dapat dilihat pada Gambar 4.13.

```
18.  Class shellHpInt(QThread):
20.     Def run(self):
21.        Try:
22.           shellInt = pyadb3.ADB()
23.           shellInt.run_shell_cmd("dd if=/dev/block/mmcblk0 | busybox nc -l -
              p 8888")
24.
25.        Except Exception:
26.           Print("Shell Int Error")
27.
28.  Class shellHpExt(QThread):
```

```
30.     Def run(self):
31.        Try:
32.           shellExt = pyadb3.ADB()
33.           shellExt.run_shell_cmd("dd if=/dev/block/mmcblk1 | busybox nc -l -
              p 8888")
34.
35.        Except Exception;
36.           Self.textBrowser.setPlainText("error ShellExt")
37.
38.
39.  Class NetcatInt(QThread):
40.     Def run(self):
41.        ncDetail = App2.textDetailGlobal
42.        ncOutFile = App2.outFileGlobal
43.        ncNameFileLog = App2.lineFileGlobal
44.        ncDirect = App2.devPath
45.        ncExam = App2.globalExa
46.
47.        ncNameFileInt = ncNameFileLog+ "_Int"
48.        fullNameFile = ncNameFileInt+ ".dd"
49.        bDirect = "/" + fullNameFile
50.        allDirect = ncDirect + bDirect
51.
52.        ncLog = ncNameFileLog+ ".log"
53.        ncOutLog = ncDirect + "/" + ncLog
54.
55.        klikMulaiInt = App2.klikMulai
56.
57.        logging.basicConfig(filename=ncOutLog, level=logging.INFO, format =
           '%(lavelname)s:%(message)s')
58.
59.        Try:
60.           commentNcInt = subprocess.getoutput('nc 127.0.0.1 8888 >'
              +ncOutFile+ '/' +fullNameFile)
61.
62.        Except Exception:
63.           Self.textBrowser.setPlainText("NetcatInt Error")
64.
65.        If commentNcInt == '':
66.           selesaiAkuisisiInt = datetime.datetime.today().strftime("%d%B%Y,
              %H:%M:%S")
67.           ncDetail.append("--- Process Aqcuisition Internal is Complete ---")
68.           ncDetail.append("Start Process Aqcuisition Internal: "+klikMulai)
69.           ncDetail.append("Finish Process Aqcuisition Internal: "
              +selesaiAkusisiInt)
70.           ncDetail.append("------ Calculate Hashing Md5 Internal ------")
71.
72.           block_size = 2 ** 20
73.           md5a = hashlib.md5()
74.           with open (allDirect, 'rb') as fileCloning:
75.              while True:
76.                 readCloning = fileCloning.read(block_size)
77.                 if not readCloning:
78.                    break
79.                 md5a.update(readCloning)
80.
81.           hasilCloning = md5a.hexdigest()
82.
83.           adbHashInt = pyadb3.ADB()
84.           coHashInt = adbHashInt.run_shell_cmd("md5sum /dev/block/mmcblk0")
85.           splitHashInt = "{coHashInt}".format(coHashInt=coHashInt)
86.           hasilHashInt = splitHashInt[2:34]
87.
88.           ncDetail.append("File Name : "+fullNameFile)
89.           ncDetail.append("Directory : "+allDirect)
90.           ncDetail.append("Md5 Source : "+hasilHashInt)
91.           ncDetail.append("Md5 Cloning : "+hasiCloning)
```

```
92.
93.          logging.info(“Examiner: {}”.format(ncExam))
94.          logging.info(“---- Hasil File Image Internal ----”)
95.          logging.info(“Storage: Internal”)
96.          logging.info(“Source Drectory: /dev/block/mmcblk0)
97.          logging.info(“Start Aqcuisition: {}”.format(klikMulaiInt))
98.          logging.info(“End AQcuisition: {}”.format(selesaiAkuisisiInt))
99.          logging.info(“File Name: {}”.format(fullNameFile))
100.         logging.info(“Directory: {}”.format(allDirect))
101.         logging.info(“Source Md5 Hash: {}”.format(hasilHashInt))
102.         logging.info(“Cloning Md5 Hash: {}”.format(hasilCloningInt))
103.
104.         if hasilHashInt == hasilCloning:
105.            ncDetail.append(“Md5 Hash Matched”)
106.            logging.info(“Md5 Hash Matched”)
107.         else:
108.            ncDetail.append(“MD5 Hash Not Matched”)
109.            logging.info(“Md5 Hash Not Matched”)
110.
111. Class NetcatExt(QThread):
112.    Def run(self):
113.       ncDetail = App2.textDetailGlobal
114.       ncOutFile = App2.outFileGlobal
115.       ncNameFileLog = App2.lineFileGlobal
116.       ncDirect = App2.devPath
117.       ncExam = App2.globalExa
118.
119.       ncNameFileExt = ncNameFileLog+ ”_Int”
120.       fullNameFile = ncNameFileExt+ ”.dd”
121.       bDirect = “/” + fullNameFile
122.       allDirect = ncDirect + bDirect
123.
124.       ncLog = ncNameFileLog+ “.log”
125.       ncOutLog = ncDirect + “/” + ncLog
126.
127.       klikMulaiExt = App2.klikMulai
128.
129.       logging.basicConfig(filename=ncOutLog, level=logging.INFO, format =
          ’%(lavelname)s:%(message)s’)
130.
131.       Try:
132.          commenctNcExt = subprocess.getoutput(‘nc 127.0.0.1 8888 >’
             +ncOutFile+ ‘/’ +fullNameFile)
133.
134.       Except Exception:
135.          Self.textBrowser.setPlainText(“NetcatExt Error”)
136.
137.       If commentNcExt == ‘’:
138.          selesaiAkuisisiExt = datetime.datetime.today().strftime(“%d%B%Y,
             %H:%M:%S”)
139.          ncDetail.append(“--- Process Aqcuisition Eksternal is Complete ---
140. ”)
141.          ncDetail.append(“Start Process Aqcuisition Internal: ”+klikMulai)
             ncDetail.append(“Finish Process Aqcuisition Internal: ”
142.          +selesaiAkusisiExt)
143.          ncDetail.append(“------ Calculate Hashing Md5 Eksternal ------”)
144.
145.          block_size = 2 ** 20
146.          md5b = hashlib.md5()
147.          with open (allDirect, ‘rb’) as fileCloning:
148.             while True:
149.                readCloning = fileCloning.read(block_size)
150.                if not readCloning:
151.                   break
152.                md5b.update(readCloning)
153.
154.          hasilCloningExt = md5a.hexdigest()
```

```
155.
156.          adbHashExt = pyadb3.ADB()
157.          coHashExt = adbHashExt.run_shell_cmd("md5sum /dev/block/mmcblk0")
158.          splitHasheXT = "{coHashExt}".format(coHashExt=coHashExt)
159.          hasilHasheXT = splitHashExt[2:34]
160.
161.          ncDetail.append("File Name : "+fullNameFile)
162.          ncDetail.append("Directory : "+allDirect)
163.          ncDetail.append("Md5 Source : "+hasilHashExt)
164.          ncDetail.append("Md5 Cloning : "+hasiCloningExt)
165.
166.          logging.info("Examiner: {}".format(ncExam))
167.          logging.info("---- Hasil File Image Eksternal ----")
168.          logging.info("Storage: Eksternal")
169.          logging.info("Source Drectory: /dev/block/mmcblk1)
170.          logging.info("Start Aqcuisition: {}".format(klikMulaiExt))
171.          logging.info("End AQcuisition: {}".format(selesaiAkuisisiInt))
172.          logging.info("File Name: {}".format(fullNameFile))
173.          logging.info("Directory: {}".format(allDirect))
174.          logging.info("Source Md5 Hash: {}".format(hasilHashInt))
175.          logging.info("Cloning Md5 Hash: {}".format(hasilCloningInt))
176.
177.          if hasilHashExt == hasilCloningExt:
178.             ncDetail.append("Md5 Hash Matched")
179.             logging.info("Md5 Hash Matched")
180.          else:
181.             ncDetail.append("MD5 Hash Not Matched")
               logging.info("Md5 Hash Not Matched")
469.
470.  def clk_extr(self):
471.     App2.outFileGlobal = self.lineOut.text()
472.     App2.lineFileGlobal = self.lineFile.text()
473.     App2.textDetailGlobal = self.textbrowser
474.     App2.globalExa = self.lineExa.text()
475.     App2.devPath = self.folderPath
476.     App2.klikMulai = datetime.datetime.today().strftime("%D%B%Y,%H:%M:%S")
477.     checkInt = self.radioInternal.isChecked()
478.     checkExt = self.radioEksternal.isChecked()
479.
480.     self.textBrowser.setPlainText("Examner : "+App2.globalExa)
481.
504.     if checkInt and checkExt:
505.        self.forShellInt = shellHpInt()
506.        self.forShellInt.start()
507.
508.        time.sleep(3)
509.
510.        self.forNcInt = NetcatInt()
           self.forNcInt.start()
515.
516.        time.sleep(2)
517.
518.        self.forShellExt = shellHpExt()
519.        self.forShellExt.start()
520.
521.        time.sleep(3)
522.
523.        self.forNcExt = NetcatExt()
           self.forNcExt.start()
529.
541.     elif checkInt:
542.        self.forShellInt = shellHpInt()
543.        self.forShellInt.start()
544.
545.        time.sleep(3)
546.
547.        self.forNcInt = NetcatInt()
```

```
        self.forNcInt.start()
552.
564.    elif checkExt:
565.        self.forShellExt = shellHpExt()
566.        self.forShellExt.start()
567.
568.        time.sleep(3)
569.
570.        self.forNcExt = NetcatExt()
            self.forNcExt.start()
```

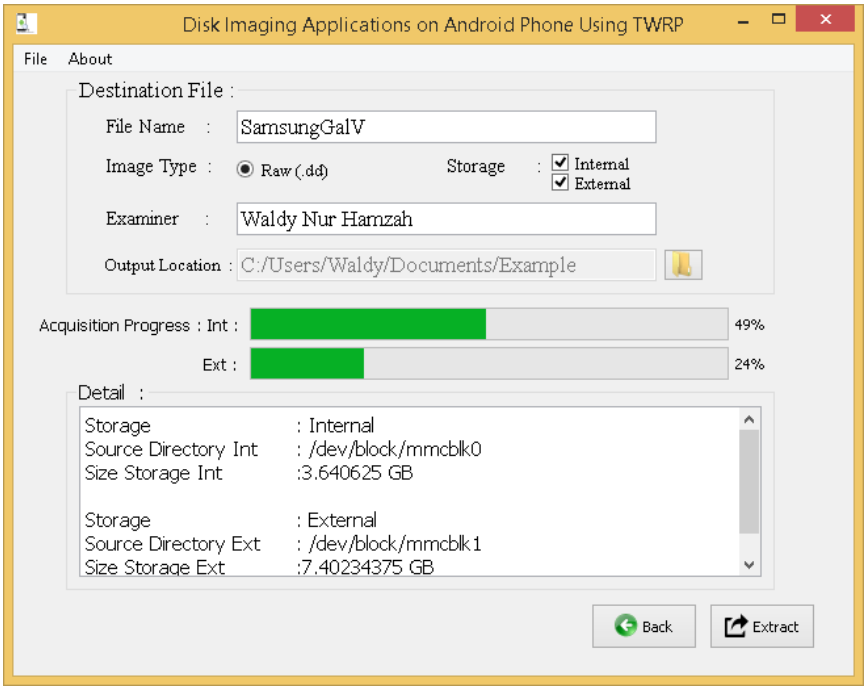
Gambar 4.13 Kode Program Proses Akusisi dan *Hashing md5* Perangkat Android

Pada Gambar 4.13 merupakan kode program proses akuisisi dan *hashing md5* penyimpanan ponsel Android yang terjadi dalam tampilan 2 aplikasi. Pada kode program baris 18 hingga 26 adalah sebuah *class QThread* untuk menjalankan proses akuisisi penyimpanan Internal ponsel Android yang terjadi dalam *shell* perangkat Android, sedangkan kode program baris 28 hingga 36 adalah sebuah *class QThread* untuk menjalankan proses akuisisi penyimpanan eksternal ponsel Android. Selanjutnya kode program baris 39 hingga 109 adalah sebuah *class QThread* untuk menjalankan proses pembuatan *file image* dan *file log* dari penyimpanan internal dan menjalankan proses *hashing md5* dari *file image* internal yang dihasil dengan file sumber penyimpanan internal, sedangkan kode program baris 111 hingga 181 adalah sebuah *class QThread* untuk menjalankan proses pembuatan *file image* dan *file log* dari penyimpanan eksternal dan menjalankan proses *hashing md5* dari *file image* eksternal yang dihasilkan dengan file sumber penyimpanan eksternal.

Setelah mengetahui kegunaan *class QThread*, selanjutnya akan menjelaskan menjalankan semua proses di dalam *class utama* aplikasi. Pada kode program baris 469 adalah sebuah fungsi untuk menjalankan suatu proses ketika tombol “extract” telah ditekan, kode program baris 470 hingga 477 adalah variabel yang menyimpan sebuah nilai *string* untuk keperluan proses akuisisi dan *file image* yang dihasilkan, kode program baris 479 adalah keluaran sebuah informasi mengenai nama “examiner” ke dalam aplikasi. Kode program baris 481 hingga 523 menjelaskan jika *check box* “internal”” dan *check box* “eksternal” dalam aplikasi terceklis maka, pada kode program baris 504 hingga 505 menjelaskan, pertama aplikasi menjalankan proses akuisisi penyimpanan Internal dalam *shell* ponsel Android dan kode program baris 507 berfungsi memberikan waktu jeda untuk memastikan proses akuisisi penyimpanan internal dalam *shell* ponsel Android telah berjalan dan selanjutnya menjalankan kode program baris 509 hingga 510 menjelaskan aplikasi menjalankan proses pembuatan *file image* dan *file log* penyimpanan Internal yang diambil dari proses yang berjalan pada kode program baris 504 hingga 505. Setelah proses akuisisi dan pembuatan *file image* penyimpanan

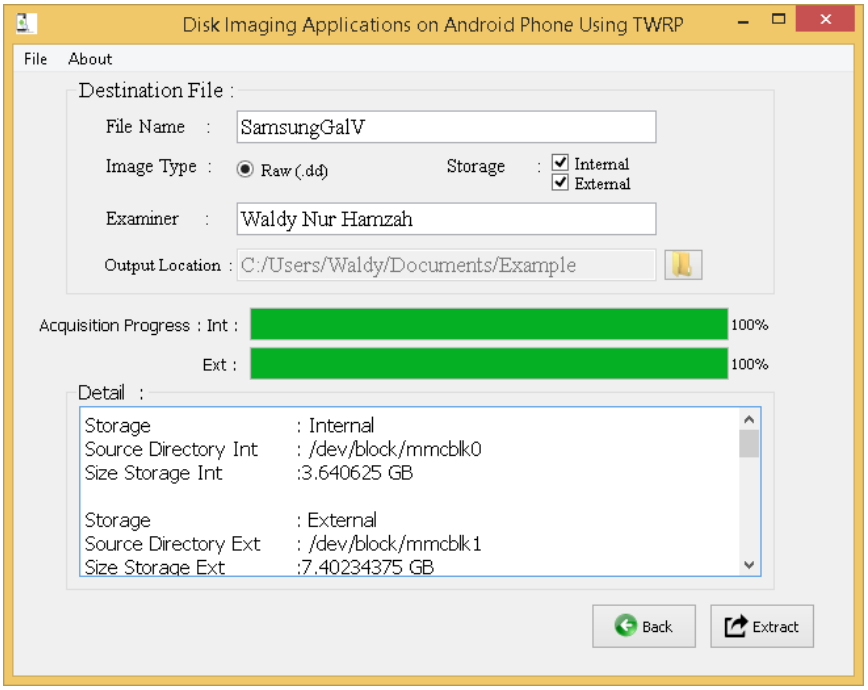
Internal sudah berjalan dalam aplikasi maka pada kode program baris 515 yang berfungsi memberikan waktu jeda untuk melakukan proses akusisi penyimpanan Eksternal, kode program baris 517 hingga 518 menjelaskan aplikasi menjalankan proses akuisisi penyimpanan Eksternal dalam *shell* ponsel Android dan kode program baris 520 berfungsi memberikan waktu jeda untuk memastikan proses akuisisi penyimpanan eksternal dalam *shell* ponsel Android telah berjalan dan selanjutnya menjalankan kode program baris 522 hingga 523 menjelaskan aplikasi menjalankan proses pembuatan *file image* dan *file log* penyimpanan Eksternal yang diambil dari proses yang berjalan pada kode program baris 517 hingga 518.

Sedangkan untuk kode program baris 529 hingga 570 menjelaskan jika pengguna memilih salah satu penyimpanan yang ingin diakuisisi dalam aplikasi. Pada kode program baris 529 hingga 547 menjelaskan untuk menjalankan proses akuisisi dan pembuatan *file image* untuk penyimpanan internal ponsel android, sedangkan kode program baris 552 hingga 570 menjelaskan untuk menjalankan proses akuisisi dan pembuatan *file image* untuk penyimpanan eksternal ponsel android. Untuk melihat proses perkembangan saat menjalankan proses akuisisi kedua penyimpanan ponsel android dapat dilihat pada Gambar 4.14, sedangkan untuk melihat hasil proses akusisi kedua penyimpanan ponsel android secara bersamaan yang telah selesai dapat dilihat pada Gambar 4.15.



Gambar 4.14 Proses akuisisi Saat Berjalan.

.



Gambar 4.15 Selesai Proses Akusisi Kedua Penyimpanan

4.3 Proses Pengujian Aplikasi

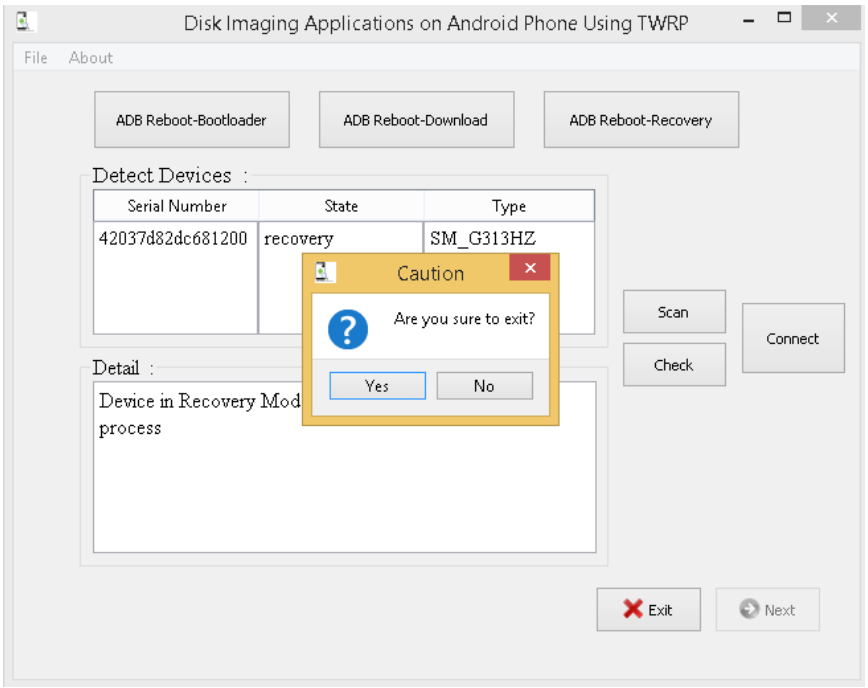
Pada tahap ini adalah melakukan pengujian terhadap aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* yang telah dikembangkan dengan cara menggunakan aplikasi secara langsung untuk melakukan proses akuisisi penyimpanan perangkat Android. Dan akan dilakukan sebuah pengujian, pengujian tersebut adalah pengujian black box yang terdiri dari pengujian fungsional dan *error handling* pada tombol yang terdapat dalam tampilan aplikasi, pengujian performa aplikasi untuk melihat kecepatan transfer data saat dilakukan proses akuisisi penyimpanan dan melakukan perhitungan dari pengujian kuesioner skala linkert yang telah disebarkan ke responden tertentukan untuk menanggapi terhadap aplikasi yang telah dibuat.

4.3.1 Pengujian Black Box

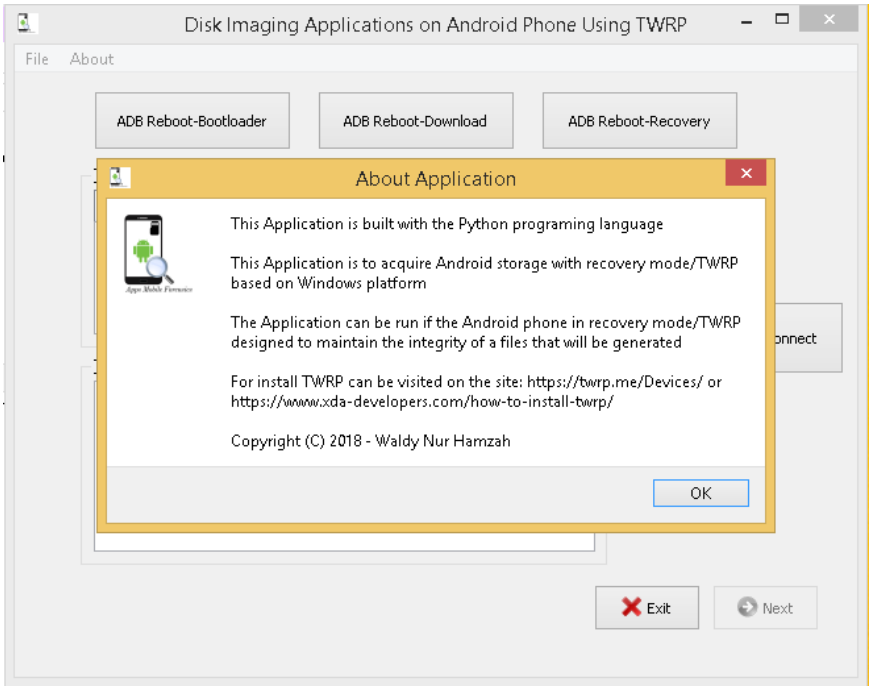
Proses pengujian dengan teknik *black box* adalah pengujian yang dilakukan dengan cara mengamati langsung ke tampilan yang terdapat dalam Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*. Adapun pengujian yang dilakukan peneliti meliputi pengujian fungsionalitas dan *error handling* pada tombol yang terdapat pada tampilan aplikasi.

1. Pengujian Fungsionalitas Tombol

Pada tahap ini akan melakukan pengujian fungsional tombol-tombol yang terdapat dalam aplikasi, pengujian funsional tombol yang pertama diuji adalah tombol-tombol yang terdapat pada tampilan 1 aplikasi, tombol yang akan diuji diantaranya adalah menu-menu aplikasi, tombol bantuan untuk *flashing file* TWRP ke dalam ponsel Android melalui aplikasi, tombol untuk menjalankan semua proses yang terjadi pada tampilan 1 aplikasi dari pengujian fungsional tombol ini untuk mengetahui apakah tombol-tombol tersebut sudah berkerja dengan baik dan sesuai fungsinya atau belum. Tombol menu pertama yang diuji adalah tombol atau sub-menu "Exit" yang menjadi sub-menu "file" dalam aplikasi, tombol atau sub-menu tersebut berfungsi untuk menutup atau keluar dari aplikasi. Untuk pengujian tombol atau sub-menu "exit" dapat dilihat pada Gambar 4.16. sedangkan tombol menu kedua yang diuji adalah tombol menu "about", tombol tersebut berfungsi menampilkan sebuah *dialog box* yang berisikan informasi tentang cara penggunaan aplikasi, manfaat aplikasi, dan siapa pembuat aplikasi. Untuk pengujian tombol menu "about" dapat dilihat pada Gambar 4.17. hasil dari pengujian tombol menu aplikasi ternyata tombol menu tersebut berkerja dengan baik dan menjalankan sesuai fungsinya.

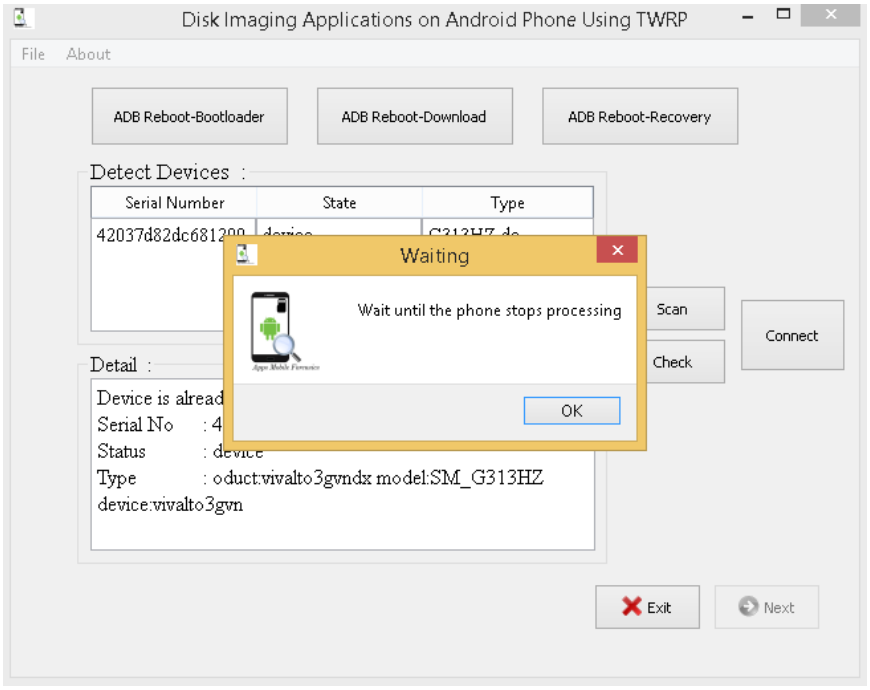


Gambar 4.16 Tombol Menu Exit.

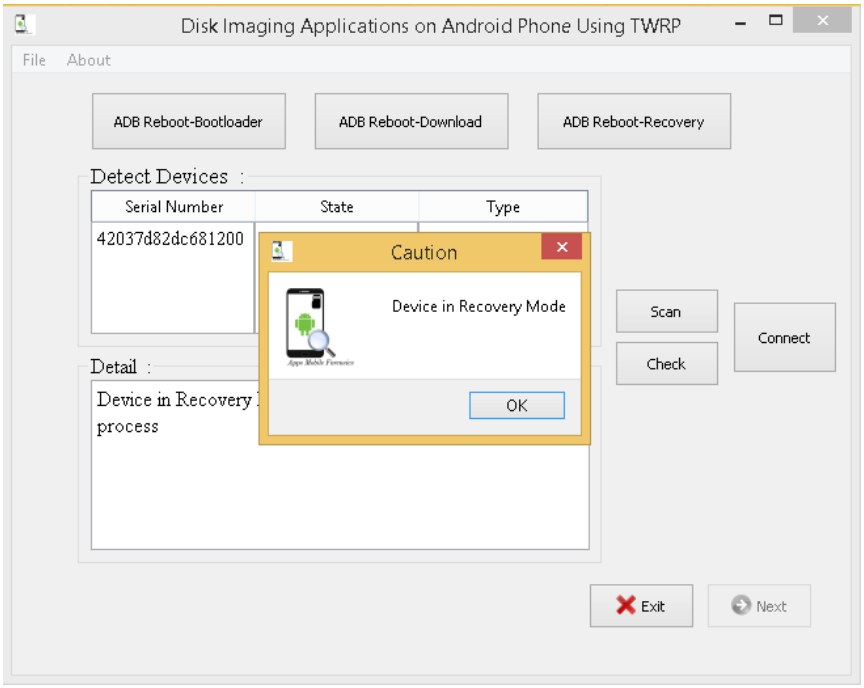


Gambar 4.17 Tombol Menu About.

Selanjutnya akan dilakukan pengujian terhadap tombol bantuan untuk *flashing file* TWRP ke dalam ponsel android dalam aplikasi. Tombol bantuan terdiri 3 tombol dalam aplikasi. Pertama adalah tombol "ADB Reboot-Bootloader" yang berfungsi untuk menjalankan ponsel Android ke dalam *bootloader mode* untuk kepentingan *flashing file* TWRP secara manual, biasanya dimiliki setiap ponsel android kecuali ponsel android keluaran Samsung, kedua adalah tombol "ADB Reboot-Download" yang berfungsi untuk menjalankan ponsel Android ke dalam *download mode* untuk *flashing file* TWRP melalui *tool* Odin dengan mudah, *download mode* hanya dimiliki ponsel android keluaran Samsung, dan ketiga adalah tombol "ADB Reboot-Recovery" yang berfungsi untuk menjalankan ponsel Android ke dalam *recovery mode* jika ponsel Android tersebut sudah ter-*install file recovery* TWRP maka dapat dengan mudah ke *recovery mode* untuk melakukan proses akuisisi penyimpanan ponsel android dengan tombol tersebut, namun jika belum ter-*install Custom Recovery* pengguna diminta melakukan instalasi *custom recovery* TWRP ke dalam ponsel android dengan menggunakan tombol bantuan 1 dan 2. Dari ketiga tombol bantuan *flashing file* TWRP aplikasi tersebut berkerja dan berfungsi dengan baik. Untuk hasil pengujian tombol bantuan aplikasi dapat dilihat pada Gambar 4.18. namun jika sudah masuk dalam *custom recovery* TWRP dalam ponsel android dapat dilihat pada Gambar 4.19.



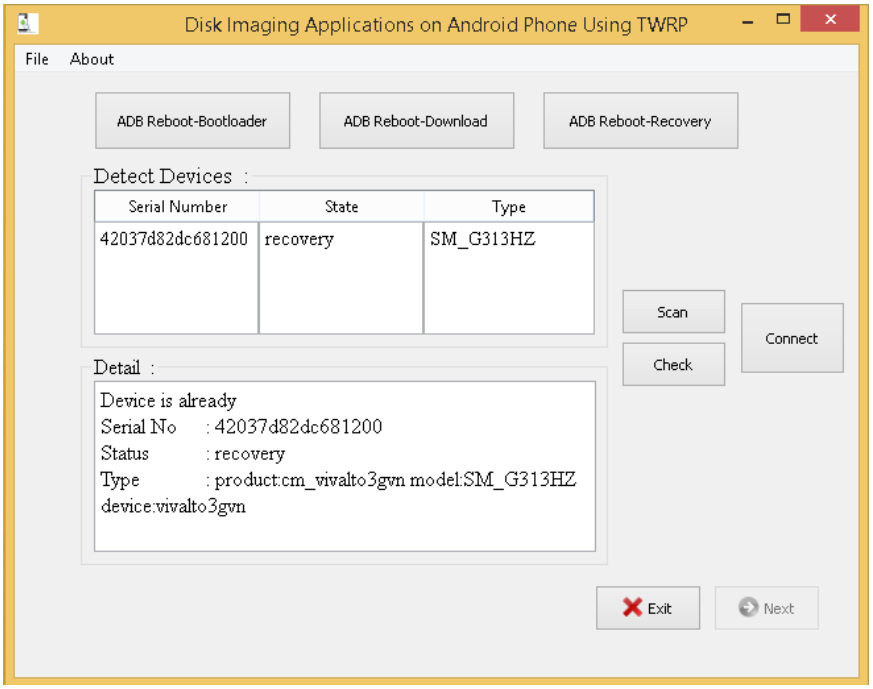
Gambar 4.18 Tombol Bantuan *Flashing File Recovery*



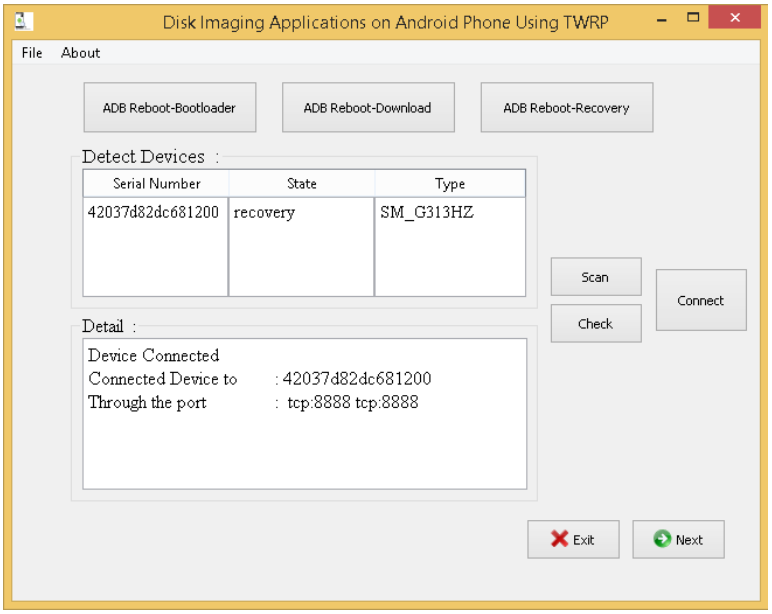
Gambar 4.19 Tombol Bantuan Jika Sudah Dalam *Custom Recovery*

Selanjutnya akan dilakukan pengujian terhadap tombol yang berfungsi untuk melakukan semua proses penting yang terjadi dalam tampilan 1 Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*. Tombol untuk menjalankan proses kepentingan akuisisi pada tampilan1 aplikasi terdiri 3 tombol. Tombol pertama diuji adalah

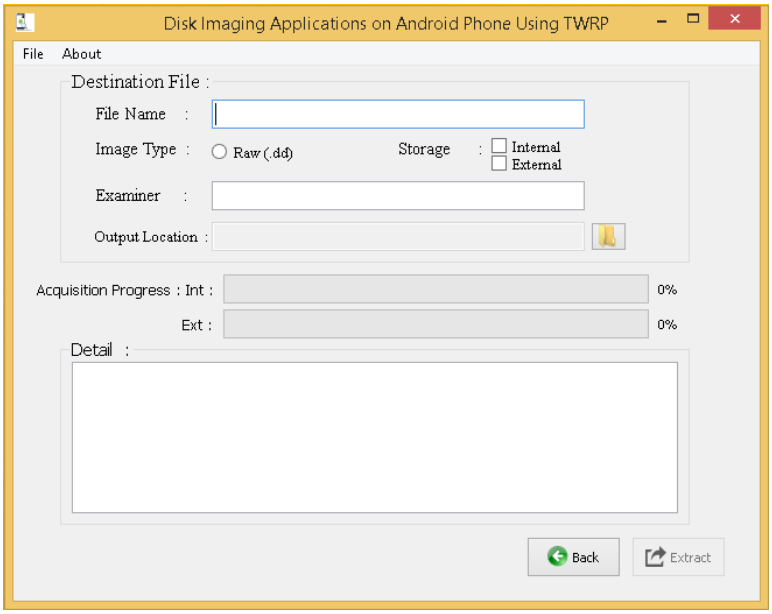
tombol “scan” berfungsi untuk menjalankan proses *scanning* ponsel Android yang terhubung dengan perangkat laptop menggunakan kabel USB dan akan menghasilkan keluaran mengenai informasi tentang *serial number*, *state*, dan *type* ponsel apa yang ditemukan aplikasi. Untuk melihat hasil pengujian tombol “scan” dapat dilihat pada Gambar 4.20. sedangkan pada tombol “connect” berfungsi untuk menjalankan proses mengkoneksikan ponsel android dengan aplikasi yang berguna untuk mengaktifkan tombol “next” jika mengkoneksikan ponsel android berhasil, dan tombol “check” berfungsi untuk menghasilkan informasi *serial number* dan informasi port berapa ponsel android yang sudah terkoneksi. Untuk melihat hasil pengujian tombol “connect” dan “check” dapat dilihat pada Gambar 4.21. selanjutnya adalah tombol “Exit” berfungsi untuk menutup atau keluar dari aplikasi dan tombol “Next” berfungsi untuk melanjutkan ke tampilan 2 aplikasi untuk keperluan proses akuisisi penyimpanan ponsel android, namun tombol “Next” akan aktif apabila tombol “connect” sudah melakukan koneksi terhadap ponsel android. Untuk melihat hasil pengujian tombol “next” dapat dilihat pada Gambar 4.22. Dari pengujian beberapa tombol untuk menjalankan semua proses dalam tampilan 1 aplikasi hasilnya berkerja dan berfungsi dengan baik.



Gambar 4.20 Tombol Scan.



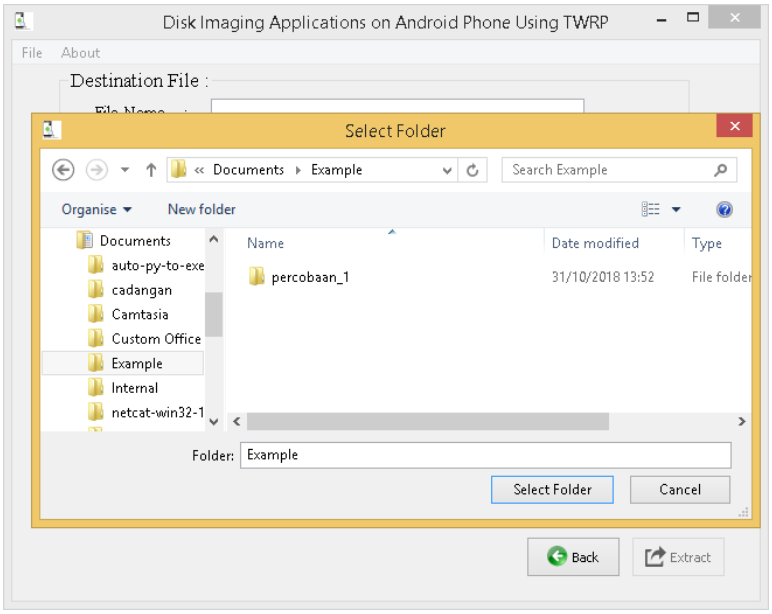
Gambar 4.21 Tombol Connect dan Check



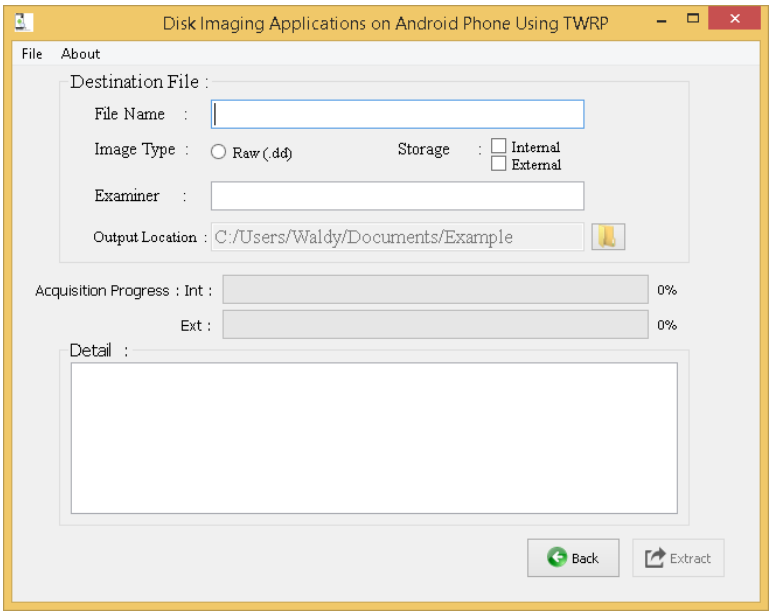
Gambar 4.22 Tombol Next

Setelah menampilkan tampilan 2, selanjutnya akan melakukan pengujian fungsional tombol pada tampilan 2 Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*, tampilan 2 aplikasi terdiri dari beberapa tombol yang memiliki fungsi masing-masing. Tombol pertama yang diuji adalah tombol select direktori terletak disamping kolom *output location* yang berfungsi memilihi suatu folder dalam perangkat laptop untuk peletakan *file image* dan *file log* yang akan dihasilkan aplikasi yang

akan ditampilankan rinciannya dalam kolom *output location*. Untuk pengujian apakah tombol select direktori sudah berkerja dengan sebagai mestinya atau tidak, maka peneliti akan mengisi kolom *output location* dengan direktori "C:\Users\Waldy\Documents\Example" yang dapat dilihat pada Gambar 4.23. Dan untuk melihat hasil dalam kolom *output location* aplikasi dapat dilihat pada Gambar 4.24. hasil dari pengujan tombol select direktori ternyata hasilnya tombol tersebut berkerja dengan baik dan semestinya.

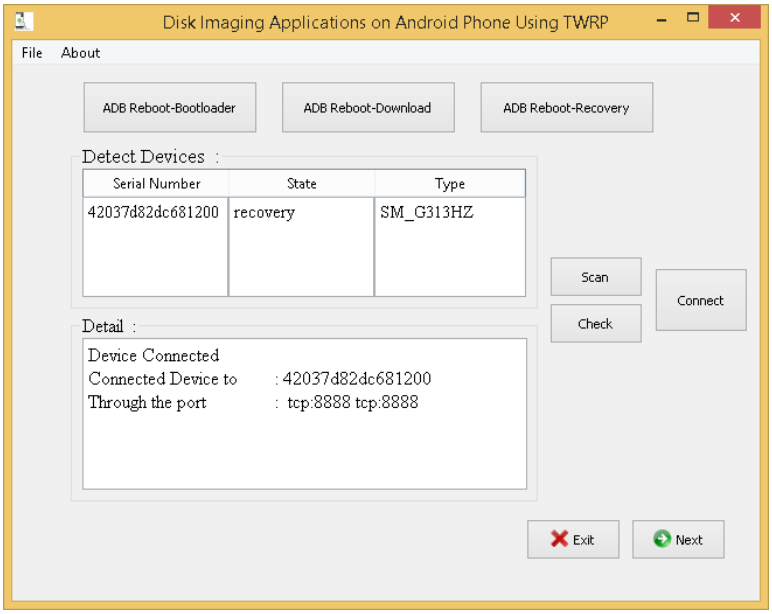


Gambar 4.23 Tombol Select Direktori

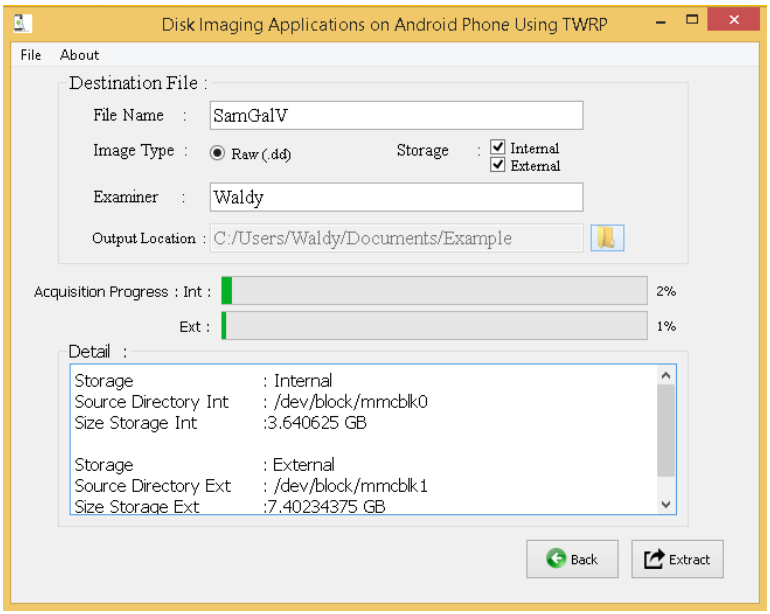


Gambar 4.24 Hasil Select Direktori

Selanjutnya pengujian fungsional tombol “back” dan “extract” pada tampilan 2 aplikasi. Pada tombol “back” berfungsi untuk kembali ke tampilan sebelumnya yaitu tampilan 1 aplikasi yang dapat dilihat pada Gambar 4.25. Sedangkan tombol “extract” berfungsi untuk menjalankan proses akuisisi penyimpanan ponsel android dalam aplikasi yang dapat dilihat pada Gambar 4.26. dari pengujian tombol “back” dan “extract” ternyata kedua tombol tersebut berkerja dengan baik dan semestinya.



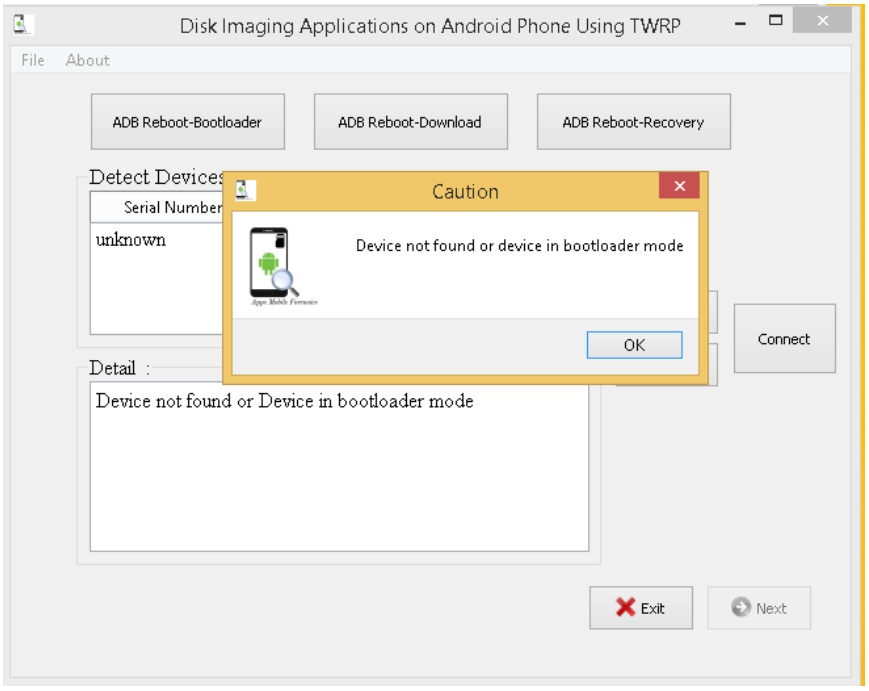
Gambar 4.25 Tombol Back



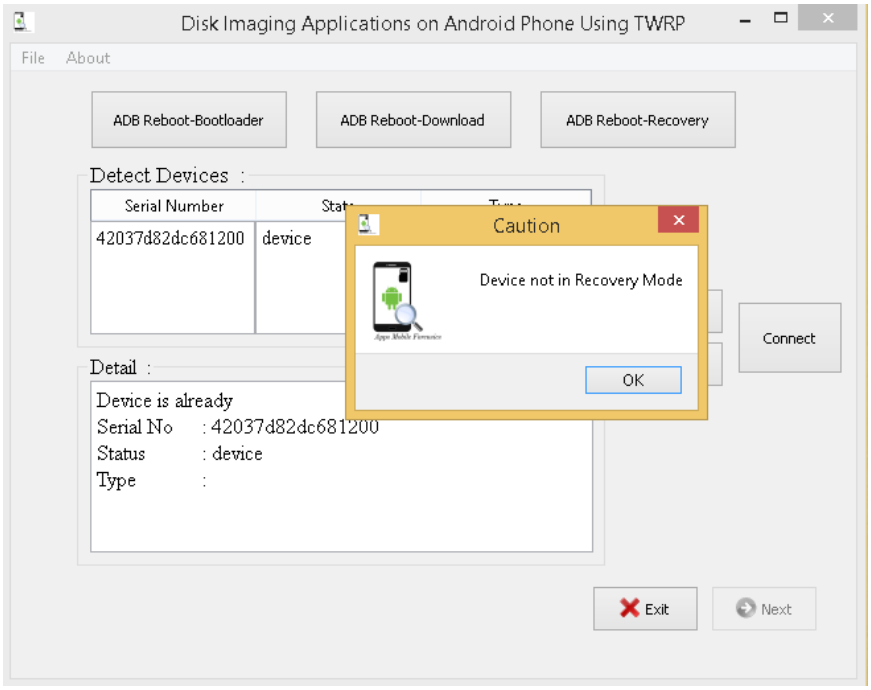
Gambar 4.26 Tombol Extract

2. Pengujian *Error Handling.*

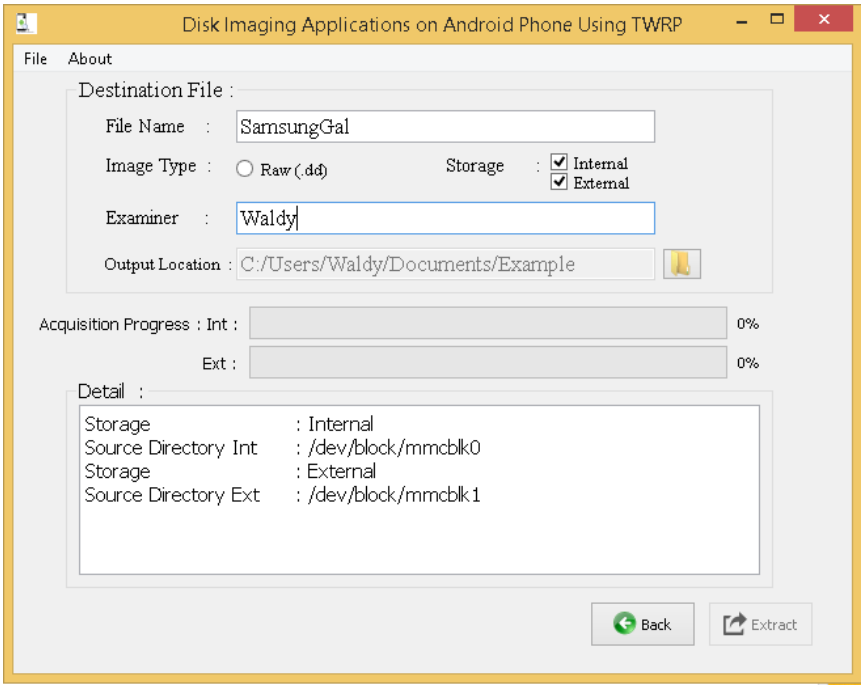
Pada tahap ini adalah melakukan pengujian *error handling* yang terdapat pada tombol dalam Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*. Pada pengujian ini akan dilakukan terhadap tombol-tombol yang kemungkinan akan ada kesalahan penggunaan aplikasi, salah satu tombol yang memiliki sifat *error handling* adalah tombol bantuan *flashing file* TWRP dalam tampilan1 aplikasi. Tombol tersebut memberikan informasi dalam *dialog box* yang akan mendeteksi kesalahan jika tidak ada ponsel android yang terhubung dengan perangkat laptop dan dapat dilihat pada Gambar 4.27. selanjutnya adalah tombol "connect" yang akan mendeteksi sebuah status ponsel android yang terhubung dengan perangkat laptop dengan kabel USB, jika ponsel tersebut berstatus "unknown" atau "devices" maka tombol "connect" akan memberikan informasi bahwa ponsel tersebut tidak dalam keadaan *custom recovery* dan aplikasi tidak dapat menjalankan proses mengkoneksikan ponsel android dan juga tidak akan mengaktifkan tombol "next" yang dapat dilihat pada Gambar 4.28. Dan selanjutnya tombol "extract" adalah tombol yang akan aktif jika semua kolom dalam *form* "file destination" sudah terisikan dengan lengkap dan benar, jika tidak lengkap mengisikan makan tombol "extract" tidak akan aktif dan dapat dilihat pada Gambar 4.29.



Gambar 4.27 *Error Handling* Tombol Bantuan



Gambar 4.28 *Error Handling* Tombol Connect

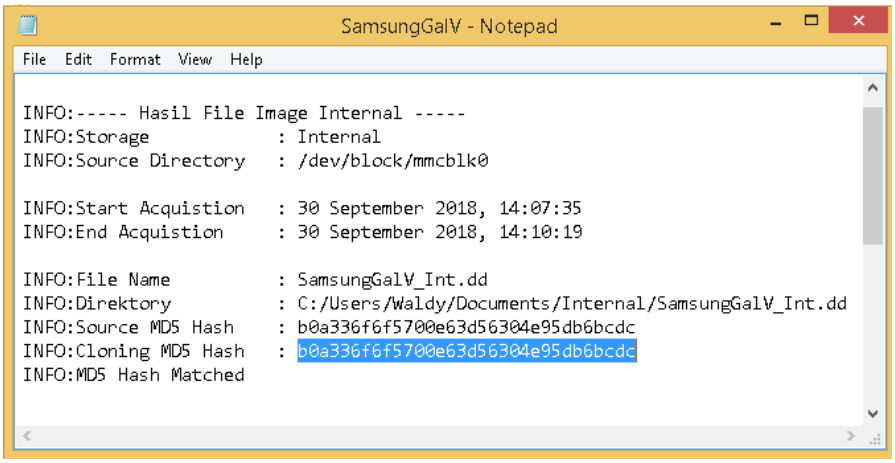


Gambar 4.29 *Error Handling* Tombol Extract

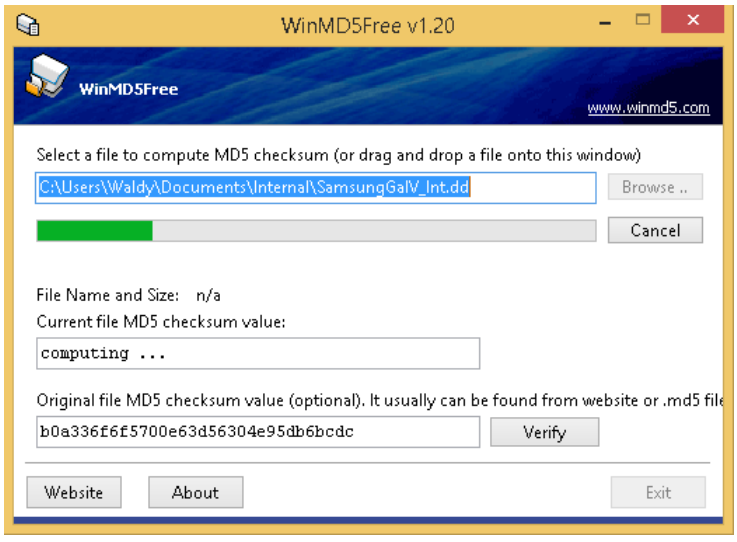
4.3.2 Pengujian Integritas Data *file image* dari Aplikasi

Tahap ini akan melakukan pengujian perbandingan integritas data dari *file image* yang dihasilkan aplikasi dengan menggunakan modul *hashlib* yang terdapat dalam bahasa pemograman *python* dengan menggunakan aplikasi WinMD5 untuk proses pengujian

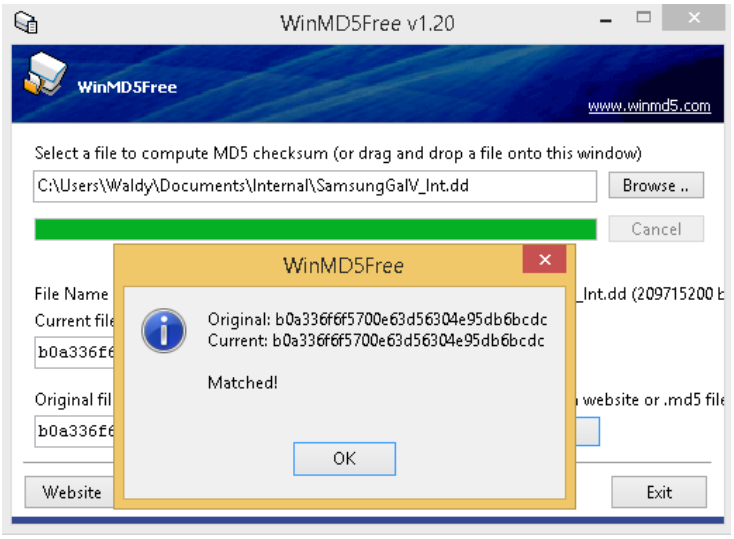
pencocokan *md5-checksum*. Untuk pengambilan *md5* dari *file* image yang dihasilkan dari aplikasi dapat dilihat pada Gambar 4.30. sedangkan untuk melihat proses pencocokan *md5* dengan menggunakan aplikasi WinMD5 dapat dilihat pada Gambar 4.31. setelah proses pencocokan *md5* dari yang dilakukan aplikasi WinMd5 telah selesai maka hasilnya dapat dilihat pada Gambar 4.32 dan dari hasil itu menunjukan bahwa md5 *file image* yang dihasilkan aplikasi yang tersimpan dalam *file log* dengan kalkulasi *md5* menggunakan aplikasi WinMD5 hasilnya ternyata sama yang menunjukan integritas data *file image* terjaga.



Gambar 4.30 *Md5* dari *File Image* yang Dihasilkan Aplikasi



Gambar 4.31 Proses Pencocokan MD5 dengan Aplikasi WinMD5



Gambar 4.32 Hasil Pencocokan MD5 Aplikasi WinMD5

4.3.3 Pengujian Performa Aplikasi

Pada tahap ini adalah untuk pengujian performa aplikasi saat melakukan proses akuisisi penyimpanan ponsel Android dalam aplikasi yang telah dibuat. Pengujian performa aplikasi dilakukan terhadap 2 ponsel Android yaitu ponsel Samsung Galaxy V bertipe SM-G313HZ dan ponsel Samsung Grand Prime dari kedua ponsel tersebut memiliki penyimpanan internal 4GB dan 8GB sedangkan untuk penyimpanan eksternal atau sdcard yang di uji yaitu 2GB, 8GB dan 16GB. Pengujian performa aplikasi dilihat dari proses kecepatan *transfer data* saat dilakukan proses akuisisi dari penyimpanan internal maupun eksternal yang terdapat ponsel android. Pengujian performa aplikasi dilakukan dengan 2 skema, sebagai berikut :

A. Mengakuisisi kedua penyimpanan secara bersamaan.

Pada skema ini adalah melakukan proses akuisisi kedua penyimpanan secara bersamaan dalam aplikasi. Dengan mengakuisisi kedua penyimpanan secara bersamaan akan berdampak pada kecepatan *transfer data* yaitu turun 1/2 hingga 1/3 dari kecepatan *transfer data* dengan melakukan akuisisi salah satu penyimpanan, namun dari segi total waktu ternyata lebih efektif dengan melakukan akuisisi kedua penyimpanan secara bersamaan. Untuk melihat hasil rincian yang didapatkan dari proses akuisisi kedua penyimpanan secara bersamaan dalam aplikasi dapat dilihat pada Tabel 4.3.

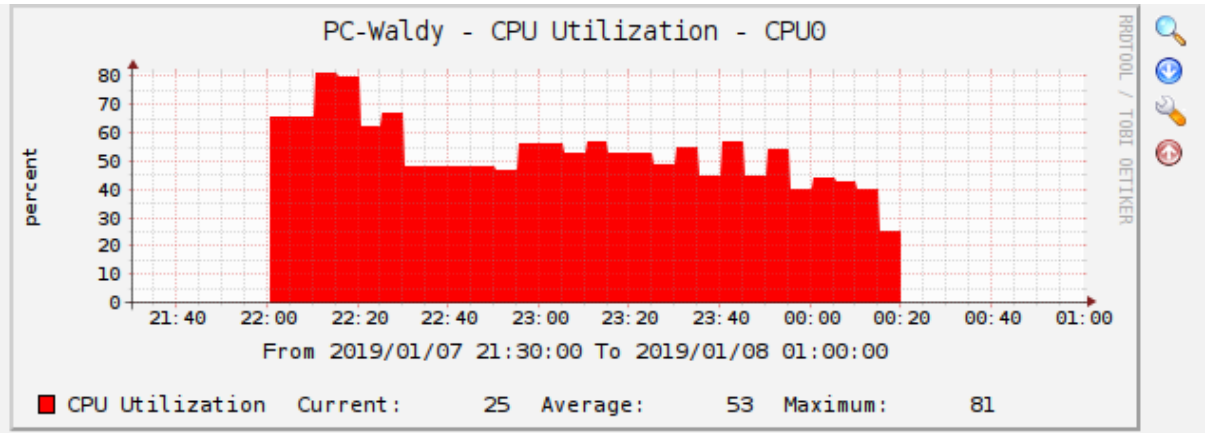
Tabel 4.3 Hasil Performa Mengakuisisi 2 Penyimpanan Secara Bersamaan

No	Ponsel Android	Sampel	penyimpanan	size	Total Waktu	Rata-rata transfer data
1.	Samsung Galaxy V SM-G313HZ	1	Internal	4GB	50 menit	1.5 MB/s
			Eksternal	2GB	38 menit	1 MB/s
		2	Internal	4GB	55 menit	1.2 MB/s
			Eksternal	8GB	93 menit	1.6 MB/s
2.	Samsung Galaxy Grand Prime	1	Internal	8GB	95 menit	1.36 MB/s
			Eksternal	2GB	35 menit	1.1 MB/s

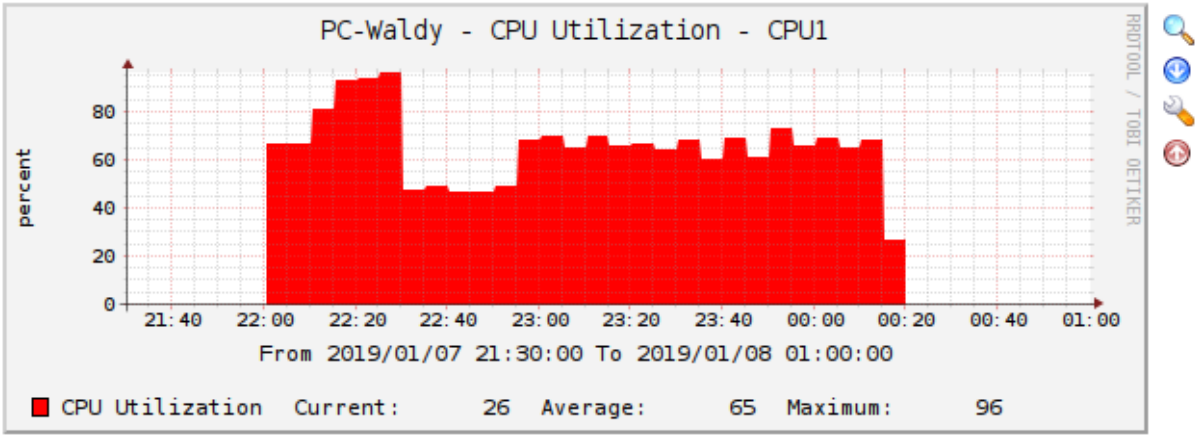
Pada Tabel 4.3 adalah hasil rincian yang dihasilkan dari performa proses akuisisi yang dilakukan aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* saat melakukan proses kedua penyimpanan ponsel android secara bersamaan baik itu intenal maupun eksternal. Pengujian performa aplikasi diuji terhadap 2 ponsel android. Ponsel 1 diuji dengen mengakuisisi 2 penyimpanan secara bersamaan yaitu internal 4GB dengan eksternal 8GB dan internal 4GB dengan eksternal 2GB, sedangkan ponsel 2 diuji yaitu internal 8GB dengan eksternal 2GB.

Pada pengujian performa aplikasi untuk ponsel 1 yang memiliki ukuran penyimpanan internal sebesar 4GB dan eksternal sebesar 8GB saat dilakukan proses akuisisi kedua penyimpanan secara bersamaan melalui aplikasi mendapatkan kecapatan transfer data rata-rata untuk penyimpanan internal 4GB sebesar 1.2 MB/s dengan durasi waktu selama 55 menit dan untuk penyimpanan eksternal 8GB sebesar 1.6 MB/s dengan durasi waktu selama 82 menit yang dapat dilihat pada Tabel 4.3. Dari hasil itu ada beberapa hal yang mempengaruhi kecepatan transfer data ialah penggunaan perangkat lunak Netcat dan menggunakan protocol TCP melalui sebuah port. TCP(*Transmission Control Protocol*) adalah jaringan untuk mengirimkan suatu paket data besar yang dikirimkan dan akan diperiksa kembali. Kelebihan menggunakan TCP ialah TCP lebih baik digunakan untuk pengiriman paket data yang besar dengan keamaanan yang bagus, meskipun membutuhkan waktu lama. Sedangkan untuk penggunaan perangkat lunak Netcat. Menurut Data center host rata-rata kecepatan penggunaan perangkat lunak dd dengan netcat adalah 6MB/s untuk sekali proses. Namun peneliti juga melakukan monitoring penggunaan sumber daya perangkat laptop dengan menggunakan

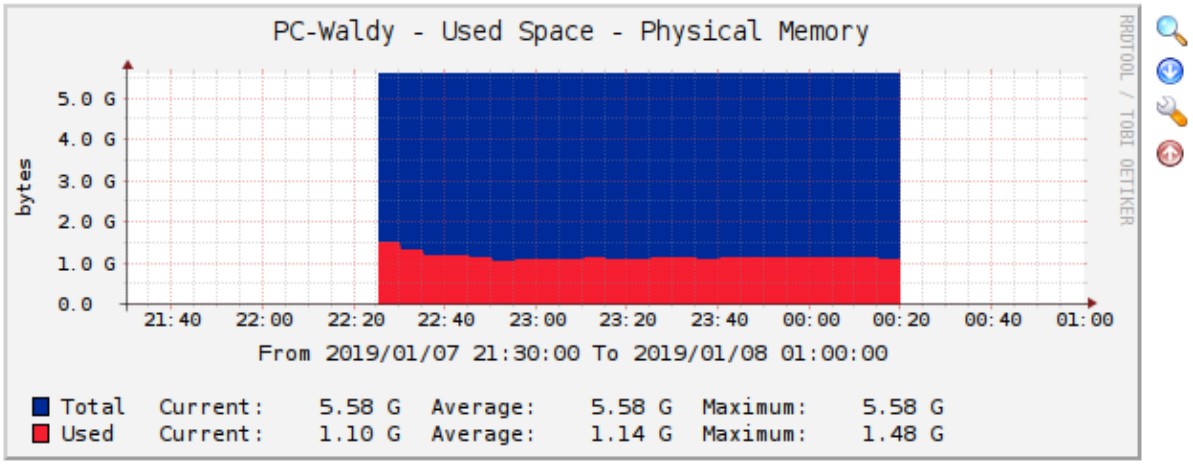
perangkat lunak Cacti. Setelah dilakukan monitoring maka mendapatkan hasil monitoring dari beberapa subjek seperti penggunaan CPU dalam perangkat laptop yang memiliki 2 CPU. CPU pertama dengan maksimum penggunaan sebesar 81 dan menjalankan rata-rata sebesar 53 saat dilakukan proses akuisisi dan CPU kedua dengan maksimum penggunaan sebesar 96 dan menjalankan rata-rata sebesar 61 yang dapat dilihat pada Gambar 4.33 dan Gambar 4.34. Sedangkan untuk penggunaan memori menyisakan sisa kapasitas memori sebesar 4GB dari ukuran memori 6GB yang dapat dilihat pada Gambar 4.35. Dari hasil memonitoring sumber daya perangkat laptop dapat disimpulkan perangkat laptop hampir mencapai pengguaan CPU secara maksimal namun penggunaan memori masih banyak tersisa sehingga kemungkinan dikarenakan penggunaan protocol TCP dan Netcat.



Gambar 4.33 Penggunaan CPU 1 Perangkat Laptop

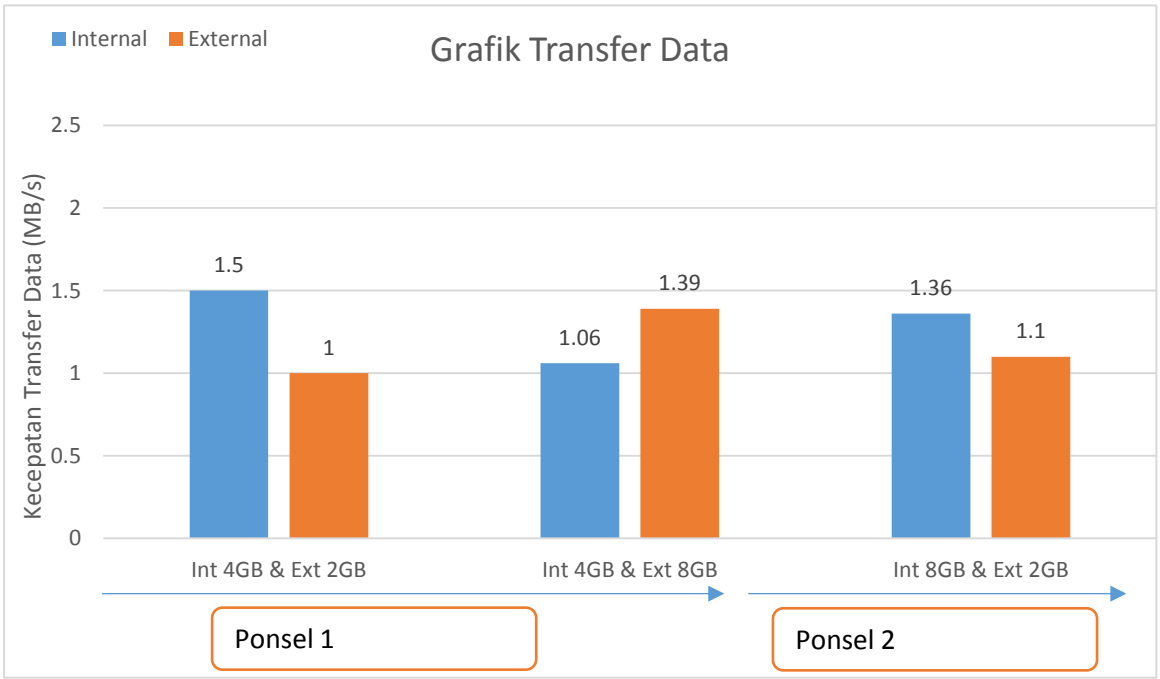


Gambar 4.34 Penggunaan CPU 2 Perangkat Laptop



Gambar 4.35 Penggunaan Memori Perangkat Laptop.

Dan pengujian pada ponsel 2 dilakukan dengan 1 sampel yang terlihat pada Tabel 4.3. pada sampel pertama ponsel 2 didapatkan kecepatan transfer data untuk penyimpanan internal 8GB sebesar 1.36 MB/s dengan durasi waktu selama 95 menit, selanjutnya kecepatan transfer data untuk penyimpanan eksternal 2GB sebesar 1.1 MB/s dengan durasi waktu selama 35 menit. Untuk melihat detail grafik kecepatan transfer data saat melakukan proses akuisisi kedua penyimpanan secara bersamaan dapat dilihat pada Gambar 4.36.



Gambar 4.36 Grafik Kecepatan Transfer Data Mengakuisisi 2 Penyimpanan.

B. Mengakuisisi salah satu penyimpanan.

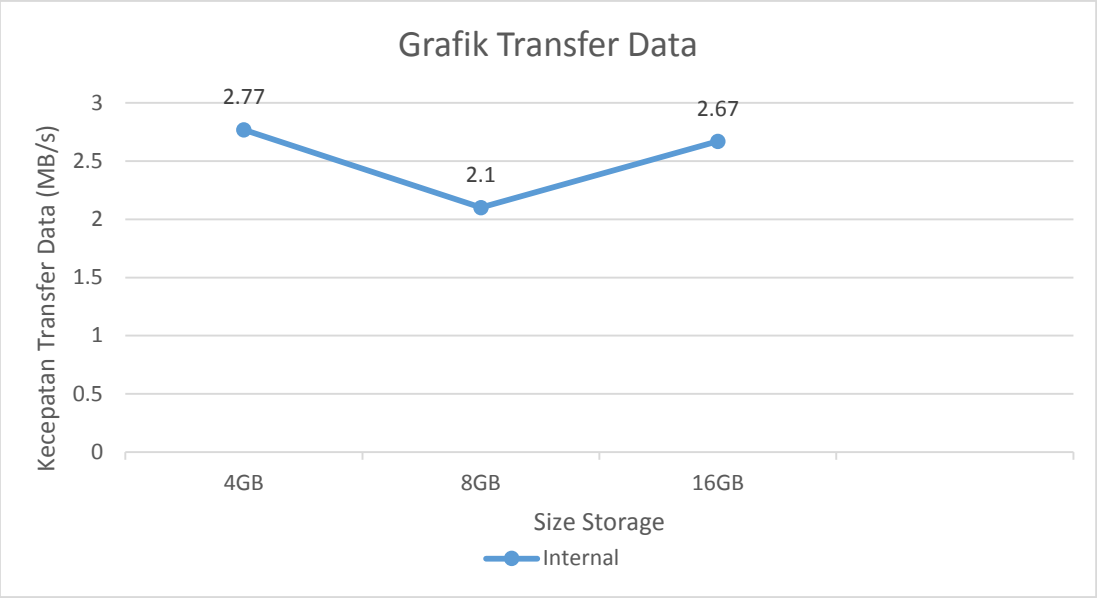
Sedangkan dengan skema mengakuisisi salah satu penyimpanan akan mendapatkan kecepatan *transfer data* yang baik dari segi total waktu dengan metode salah satu penyimpanan terlebih dahulu. Untuk melihat hasil rincian yang didapat dari proses akuisisi salah satu penyimpanan dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Performa Mengakuisisi Salah Satu Penyimpanan

No	**Ponsel Android**	**Penyimpanan**	**Size**	**Total Waktu**	**Rata-rata Transfer data**
1.	Samsung Galaxy V, SM-G313HZ	Internal	4GB	23 Menit	2.77 MB/s
2.	Samsung Grand Prime	Internal	8GB	60 Menit	2.1 MB/s
3.	Sony Z3 Compact	Internal	16GB	95 Menit	2.67 MB/s

Pada Tabel 4.4 adalah hasil rincian yang dihasilkan dari performa proses akuisisi yang dilakukan aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery* saat melakukan proses akuisisi salah satu penyimpanan internal ponsel android. Pengujian performa aplikasi diuji terhadap 3 perangkat ponsel android yang memiliki penyimpanan internal 4GB, 8GB dan dan 16GB.

Pada pengujian performa aplikasi untuk melihat kecepatan transfer data saat melakukan proses akuisisi salah satu penyimpanan internal pada ponsel android didapatkan kecepatan transfer data untuk penyimpanan internal 4GB pada ponsel 1 sebesar 2.77 MB/s dengan membutuhkan durasi waktu selama 23 menit, dan untuk penyimpanan internal 8GB pada ponsel 2 kecepatan transfer datanya sebesar 2.1 MB/s dengan membutuhkan durasi waktu selama 60 menit, sedangkan untuk penyimpanan internal 16GB pada ponsel 3 kecepatan transfer datanya sebesar 2.67 MB/s dengan membutuhkan durasi waktu selama 95 menit. Untuk melihat detail grafik kecepatan transfer data saat melakukan proses akuisisi salah satu penyimpanan dapat dilihat pada Gambar 4.37



Gambar 4.37 Grafik Kecepatan Transfer Data Salah Satu Penyimpanan

4.3.4 Total Skor Kuesioner Skala Linkert Aplikasi

Pada tahap ini menjelaskan hasil yang didapat setelah melakukan pengujian kuisoner yang dibagian ke 50 responden dengan melakukan perhitungan skor menggunakan skala linkert yang berdasarkan hasil yang didapat. Untuk melihat hasil skor dari perhitungan linkert dapat dilihat pada Tabel 4.5.

Tabel 4.5 Hasil Skor Kuesioner Skala Linkert

No	URAIAN	SKOR					Total Skor
A.	**Manfaat**	**1**	**2**	**3**	**4**	**5**	
1.	Aplikasi *Forensic Imaging* ponsel android dengan memanfaatkan *Custom Recovery* ini memiliki manfaat bagi saya yang bergelut di bidang forensic ?			5	26	19	(3x5) + (4x26) + (5x19) = 214:50 = 4,28
2.	Penggunaan *Team Win Recovery Project*(TWRP) pada perangkat Android untuk proses akuisisi mampu menekan perubahan intergitas data yang dihasilkan?		2	5	26	15	(2x2) + (3x5) + (4x26) + (5x15) = 198:50 = 3,96

No	URAIAN	SKOR					Total Skor
3.	Aplikasi *Forensic Imaging* Android dengan memanfaatkan *Custom Recovery* ini mudah digunakan?			10	23	17	(3x10) + (4x23) + (5x17) = 207:50 = 4,14
4.	Menggunakan aplikasi ini ketika ingin melakukan cloning penyimpanan Android?			11	23	16	(3x11) + (4x23) + (5x16) = 205:50 = 4,1
5.	Saya berharap aplikasi ini dapat digunakan orang banyak?	1	1	3	19	26	(1x1) + (2x1) + (3x3) + (4x19) + (5x26) = 218:50 = 4,36
6.	Keluaran yang dihasilkan aplikasi ini sudah sesuai dengan yang saya harapkan?			7	26	17	(3x7) + (4x26) + (5x17) = 210:50 = 4,2
B.	**Tampilan**	**1**	**2**	**3**	**4**	**5**	
1.	Tampilan antarmuka aplikasi ini menarik bagi saya?			13	24	13	(3x13) + (4x24) + (5x13) = 200:50 = 4
2.	Tampilan antarmuka aplikasi mudah untuk dikenali?			11	30	9	(3x11) + (4x30) + (5x9) = 198:50 = 3,96
3.	Menu yang ditampilkan mudah untuk dipahami?			11	26	13	(3x11) + (4x26) + (5x13) = 202:50 = 4,04
4.	Peletakan semua fitur dan tombol aplikasi sudah sesuai?			11	26	13	(3x11) + (4x26) + (5x13) = 202:50 = 4,04
C.	**Fungsionalitas**	**1**	**2**	**3**	**4**	**5**	
1.	Fitur untuk *flashing* file TWRP berjalan dengan semestinya?			6	26	18	(3x6) + (4x26) + (5x18) = 212:50 = 4,24

No	URAIAN	SKOR					Total Skor
2.	Fitur untuk scanning perangkat android berjalan dengan baik?			8	26	16	(3x8) + (4x26) + (5x16) = 208:50 = 4,16
3.	Fitur untuk mengkoneksikan perangkat android berjalan dengan baik?			7	24	19	(3x7) + (4x24) + (5x19) = 212:50 = 4,24
4.	Fitur untuk proses akuisisi penyimpanan perangkat android berjalan dengan baik?			6	28	16	(3x6) + (4x28) + (5x16) = 210:50 = 4,2
5.	Semua tombol dan menu dapat bekerja dengan normal?			3	26	21	(3x3) + (4x26) + (5x21) = 218/50 = 4,36

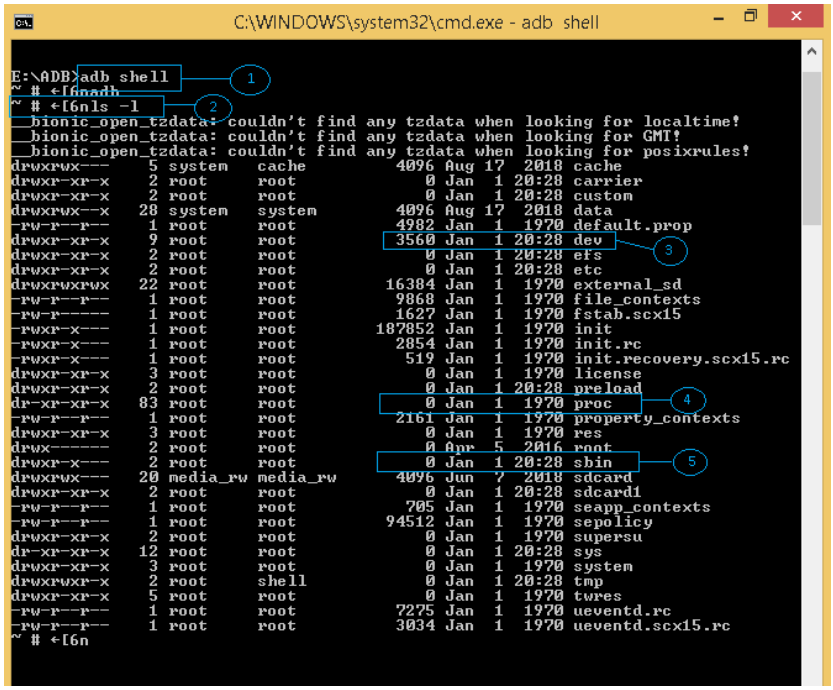
4.4 Proses Analisis

4.4.1 Analisis Model Penyimpanan Data pada Ponsel Android

Pada tahap ini peneliti akan melakukan analisis model penyimpanan data pada ponsel android dengan *custom recovery* terhadap ponsel yang dijadikan contoh barang bukti yang berguna mendapatkan informasi-informasi mengenai ukuran penyimpanan, partisi, dan *file system* yang digunakan pada ponsel Android. Untuk mendapatkan semua informasi tersebut dilakukan beberapa analisis, sebagai berikut :

a. Analisis *Directory Structure* perangkat Android.

Android memiliki struktur direktori khusus untuk tata letak penataan penyimpanan, untuk mengetahui dan melihat struktur direktori pada perangkat Android dengan menggunakan perintah “adb shell”, dan dapat juga melihat struktur direktori mengunakan DDMS (*Dalvik Debug Monitor Server*). Setelah melakukan perintah “adb shell” maka akan masuk ke dalam *shell* perangkat dan lakukan perintah “ls -l” untuk melihat struktur direktori perangkat tersebut. untuk melihat struktur direktori yang terdapat dalam ponsel android “Samsung Galaxy V” dapat dilihat pada Gambar 4.38.

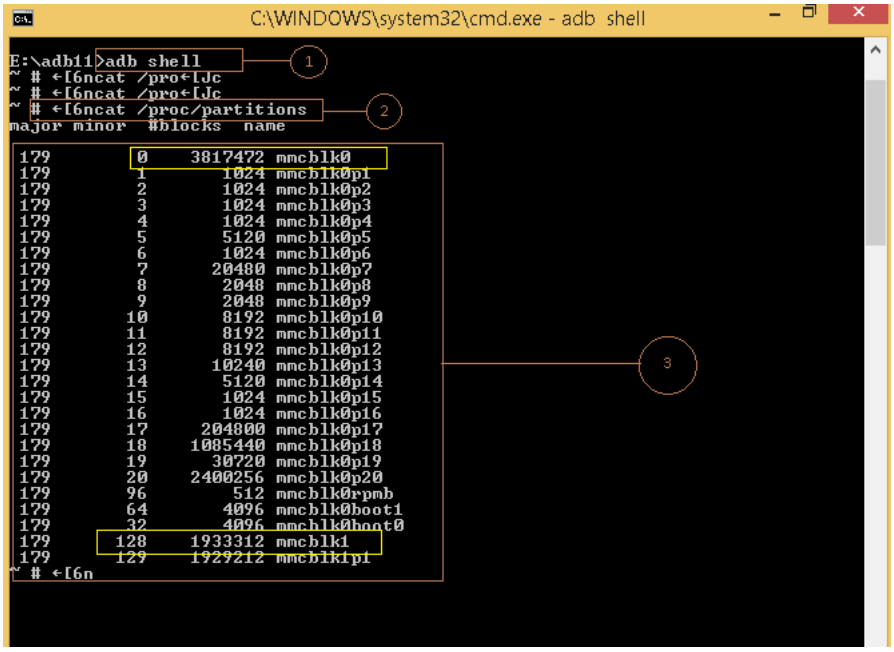


Gambar 4.38 Struktur Direktori Ponsel Android

Pada Gambar 4.38 menjelaskan pada perintah (1)"adb shell" adalah perintah adb untuk masuk dalam *shell* pada perangkat android, setelah masuk dalam *shell* perangkat lakukan perintah (2)"ls -l" perintah ini untuk melihat struktur direktori dalam perangkat, pada (3)"/dev" menunjukan nama partisi utama yang akan digunakan untuk tujuan proses akuisisi dalam perangkat Android. (4)"/proc" adalah partisi yang dapat mengetahui dan menampilkan penamaan partisi penyimpanan dan *filesystems* apa saja yang digunakan dalam perangkat Android. (5)"/sbin" adalah partisi yang meletakan *file busybox* yang memiliki berbagai fungsi digunakan untuk proses ekstraksi, mengetahui ukuran penyimpanan, dan lainnya.

b. Analisis *Partition Layout* perangkat Android.

Setelah melakukan analisis struktur directori perangkat, kita dapat mengetahui direktori apa yang menyimpan informasi tentang *partitions layout*. Untuk mengetahui *partitions layout* pada perangkat android yaitu menggunakan perintah "cat". "cat" berguna menampilkan sebuah informasi yang terdapat dalam partisi. Dan pada sebuah partisi "/proc/partitions" adalah partisi yang menyimpan informasi tentang penamaan partisi penyimpanan, maka digunakan perintah "cat .proc/partitions". untuk melihat semua partisi yang digunakan dalam penyimpanan perangkat Andoid dapat dilihat pada Gambar 4.39.

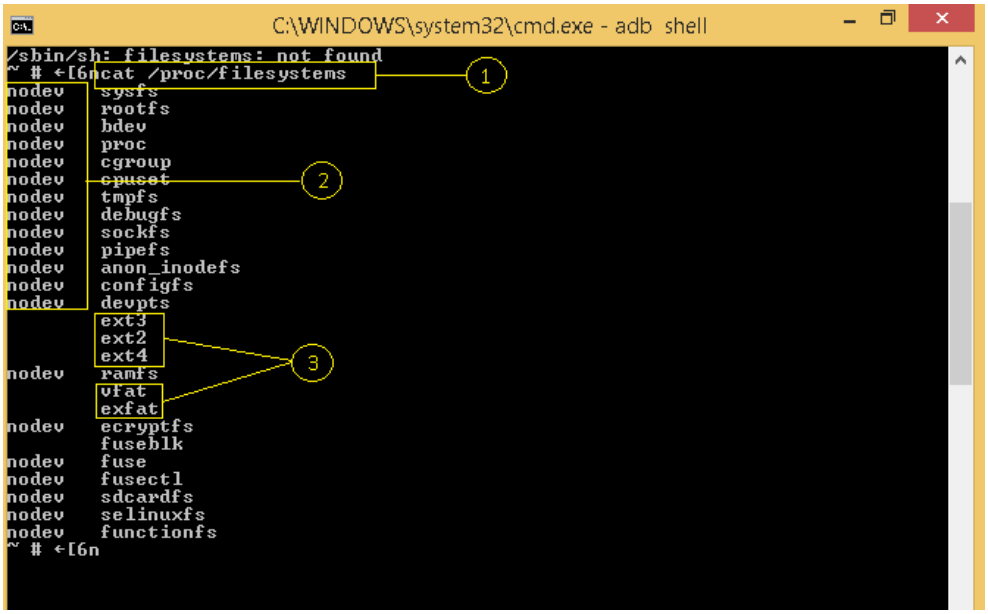


Gambar 4.39 List Partisi Penyimpanan pada File “partitions”

Pada Gambar 4.39 menjelaskan perintah (1)”adb shell” digunakan untuk mengakses shell pada perangkat, setelah masuk shell pada perangkat, kemudian perintah (2)“cat /proc/partitions” digunakan untuk menampilkan list partisi-partisi penyimpanan pada perangkat Android dan pada (3) adalah list penamaan dan ukuran sebuah partisi penyimpanan perangkat. Pada list penamaan partisi penyimpanan terdapat partisi “mmcblk0” partisi tersebut adalah partisi utama dari semua penamaan partisi “mmcblk0-”, partisi tersebut merujuk ke penyimpanan internal sebuah perangkat android, sedangkan pada partisi “mmcblk1” adalah partisi utama dari semua penamaan partisi “mmcblk1”, partisi tersebut merujuk ke penyimpanan eksternal sebuah perangkat android.

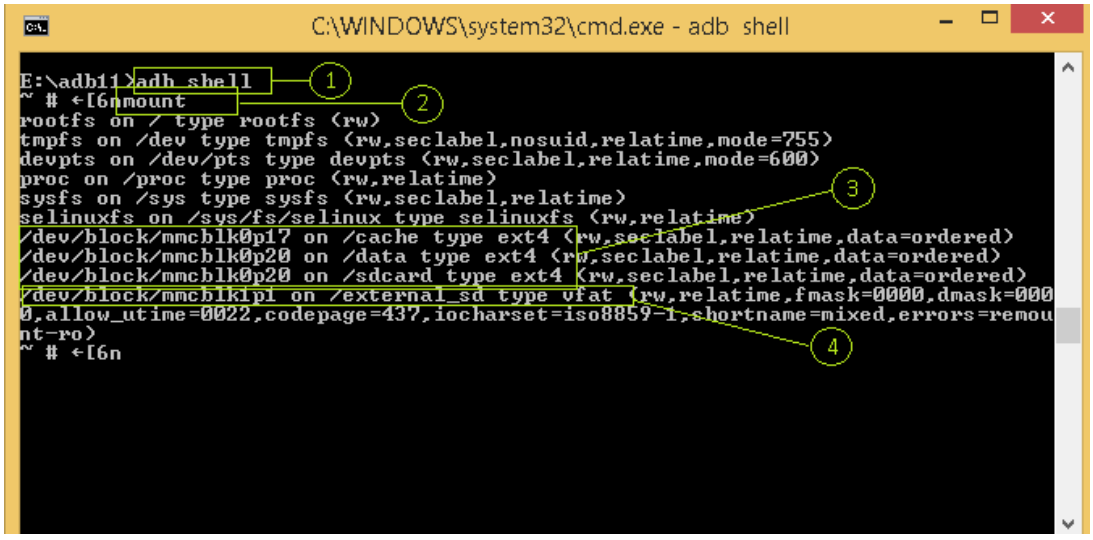
c. Analisis *filesystem* perangkat Android.

Tahap analisis *file system* perangkat android adalah untuk mengetahui *file system* apa yang digunakan penyimpanan data dalam ponsel Android Samsung Galaxy V yang telah diketahui nama partisi yang dapat melihat semua *filesystem* yaitu dengan menggunakan perintah “cat” dalam sebuah partisi bernama “/proc/filesystems” yang terdapat dalam perangkat Android. Untuk melihat hasil dari perintah melihat *file system* dapat dilihat pada Gambar 4.40.



Gambar 4.40 *filesystem* yang digunakan pada Perangkat Android

Pada Gambar 4.40 merupakan semua *filesystem* yang digunakan dalam perangkat Android. Pada (1) “cat /proc/filesystems” adalah perintah untuk melihat semua *file system* yang digunakan ponsel Android, (2)”nodev” adalah *Filesystem* yang memiliki entri “Nodev” menunjukan *file systems* tersebut tidak dapat di*mounting*, sedangkan untuk (3) *file system* yang digunakan penyimpanan data Internal maupun Eksternal yang tidak memiliki entri “nodev” yaitu “ext2,3,4” dan “vfat, exfat”. Dan untuk melihat rinciannya dapat dilakukan dengan perintah “mount” dan dilihat pada Gambar 4.41.

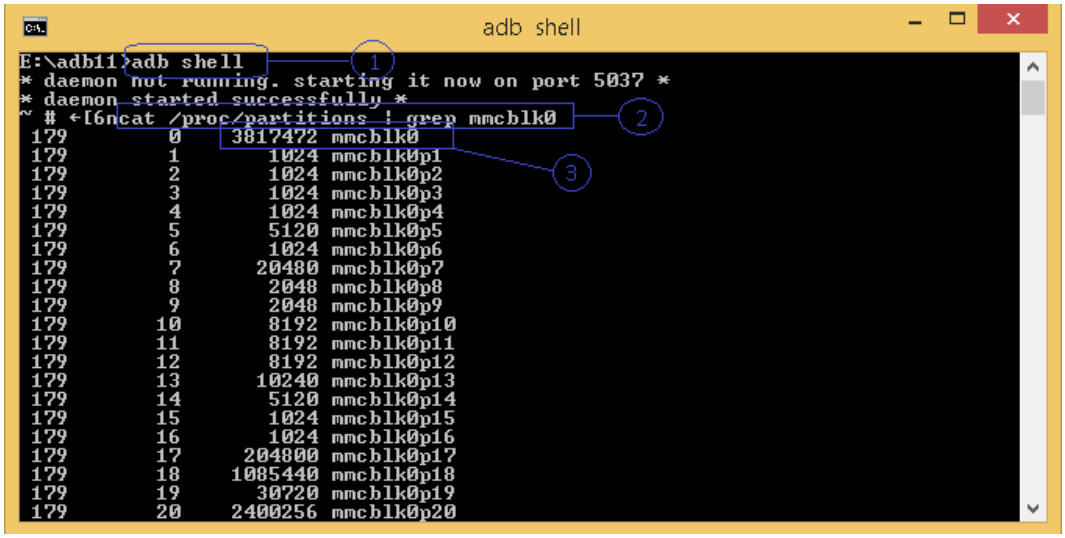


Gambar 4.41 Rincian *File system* Internal dan Eksternal.

Pada Gambar 4.41 merupakan rincian *file system* yang digunakan dalam penyimpanan data internal dan eksternal ponsel android. Dapat dilihat pada petunjuk (1) "Adb Shell" adalah perintah untuk masuk dalam *shell* ponsel android, sedangkan petunjuk (2) "mount" adalah perintah untuk melihat *file system* yang dapat di*mount*. Dan pada petunjuk (3) adalah daftar partisi "mmcblk0" yang merujuk ke penyimpanan internal ponsel android menggunakan *file system* "ext4", dari *file system* tersebut menunjukan penyimpanan internal ponsel andrid dapat dilakukan proses akuisisi, sedangkan petunjuk (4) adalah partisi "mmcblk1" yang merujuk ke penyimpanan eksternal dengan menggunakan *file system* "vfat", dari *file system* "vfat" menunjukan bahwa penyimpanan extrenal ponsel android juga dapat dilakukan proses akuisisi.

d. Analisis ukuran penyimpanan perangkat Android

Tahap analisis ukuran penyimpanan ponsel Android baik itu internal maupun eksternal yang berguna untuk kepentingan membuat *acquisition progress* pada tampilam aplikasi dan dapat mengetahui perkembangan proses akuisisi penyimpanan yang telah berjalan dalam aplikasi secara *real-time*. Pertama yang dilakukan adalah menganalisis ukuran penyimpanan internal ponsel Android yang merujuk ke partisi "mmcblk0". untuk melihat hasil analisis ukuran penyimpanan internal ponsel android dapat dilihat pada Gambar 4.42.



Gambar 4.42 Hasil Analisis Ukuran Internal Ponsel Android.

Pada Gambar 4.42 merupakan hasil analisis ukuran penyimpanan internal ponsel Android, dan menjelaskan pada perintah (1)"adb shell" digunakan untuk mengakses hak *shell* pada ponsel android, setelah masuk *shell* pada ponsel android, kemudian melakukan perintah (2)"cat /proc/partitions | grep mmcblk0" berguna untuk menampilkan semua list partisi penyimpanan internal ponsel android yang hanya merujuk "mmcblk0" dan pada petunjuk (3) "3817472" adalah ukuran utama penyimpanan internal ponsel android yang akan dijadikan sebagai *set maximum* "*acquisition progress Int*" dalam tampilan aplikasi. sedangkan analisis ukuran penyimpanan eksternal ponsel android yang merujuk ke partisi "mmcblk1" dapat dilihat pada Gambar 4.43.



Gambar 4.43 Hasil Analisis Ukuran Eksternal Ponsel Android.

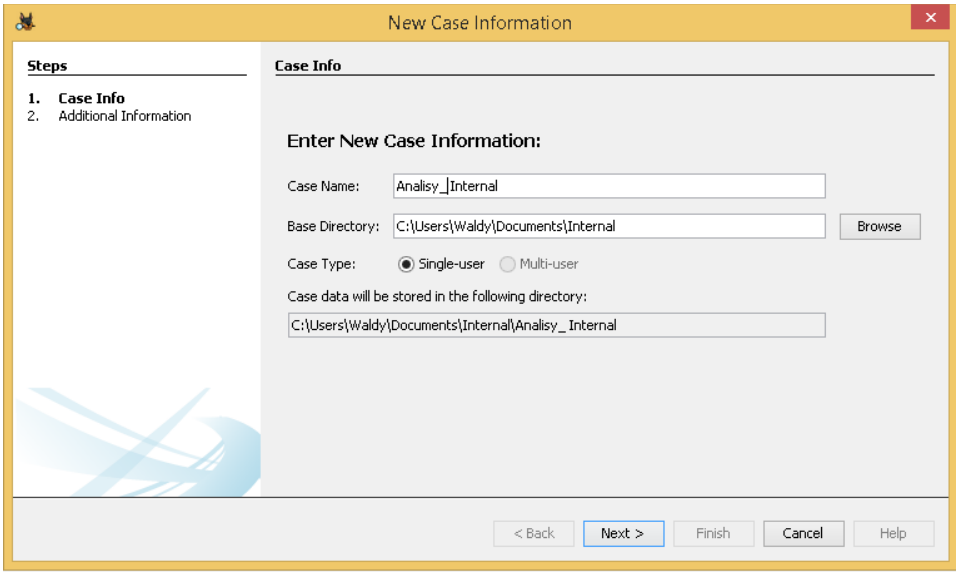
Pada Gambar 4.43 merupakan hasil analisis ukuran penyimpanan internal ponsel Android, dan dapat dijelaskan pada perintah (1) "cat /proc/patitions | grep mmcblk1" bergunakan untuk penampilkan semua partisi penyimpanan yang bernama "mmcblk1" dan pada (2) "1933312" adalah ukuran utama penyimpanan eksternal ponsel android yang akan dijadikan set maximum "*acquisition progress ext*" dalam tampilan aplikasi.

Dari semua tahapan proses analisis model penyimpanan data perangkat android yang dilakukan penulis, diperolehlah data yang dapat digunakan penulis sebagai rujukan untuk pengembangan aplikasi *Forensic Imaging* perangkat Android dengan memanfaatkan *Custom Recovery*. Data tersebut diperoleh dari beberapa proses analisis di atas, dari data tersebut, penulis dapat mengidentifikasi partisi utama dari penyimpanan perangkat android, *file system* yang digunakan dan juga ukuran penyimpanan yang berguna untuk membuat *progress bar* dalam aplikasi, data-data yang didapatkan sebagai berikut :

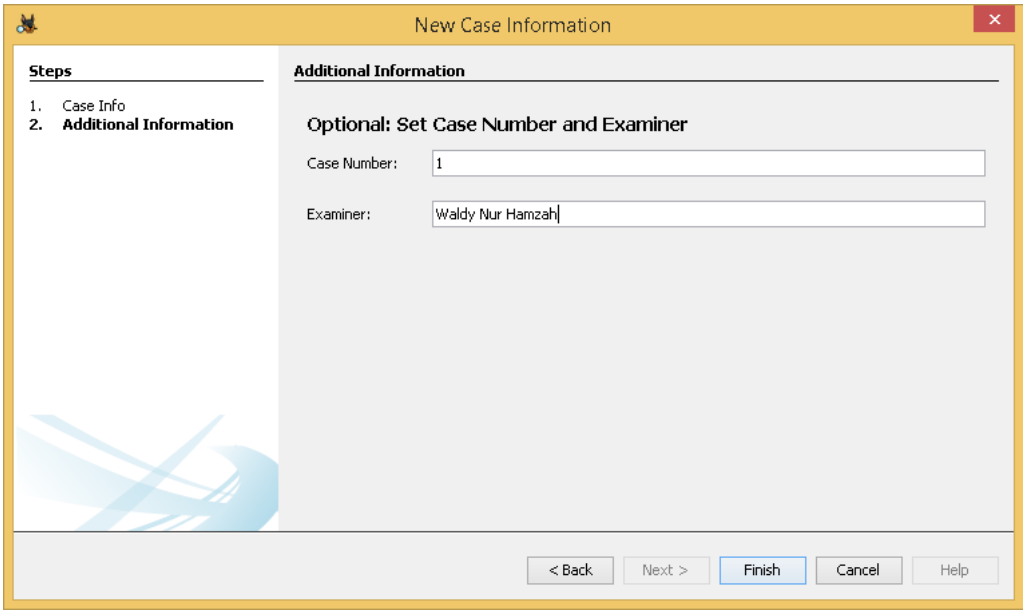
a. Dari kedua perangkat ponsel yang sudah dianalisis tersebut menunjukan partisi utama penyimpanan yang merujuk ke penyimpanan internal maupun eksternal adalah sama. Yaitu partisi "/dev/block/mmcblk0" merujuk kesebuah penyimpanan Internal dalam ponsel Android sedangkan partisi "/dev/block/mmcblk1" merujuk kesebuah penyimpanan Eksternal dalam ponsel Android.
b. *File system* yang digunakan pada penyimpanan internal adalah EXT4 sedangkan untuk *file system* yang digunakan penyimpanan eksternal adalah VFAT.
c. Ukuran penyimpanan internal dan eksternal yang akan dijadikan set maximum *"progress acquisition"* dalam aplikasi untuk mengetahi perkembangan proses akuisisi yang telah dijalankan.

4.4.2 Analisis Data *File Image* yang Dihasilkan Aplikasi

Tahap ini adalah untuk menganalisis *file image* yang dihasilkan aplikasi dengan menggunakan *tools* Autopsy versi 4.5.0 dalam *platform* Windows. Saat membuka aplikasi Autopsy pertama yang dilakukan adalah mengisikan beberapa *form* yaitu *form case info* untuk penamaan dan tata letak kasus yang dianalisis dan *Additional information* untuk memberi urutan kasus dan siapa nama pengguna yang ingin melakukan analisis. Untuk melihat kedua *form* dapat dilihat pada Gambar 4.44 dan Gambar 4.45.



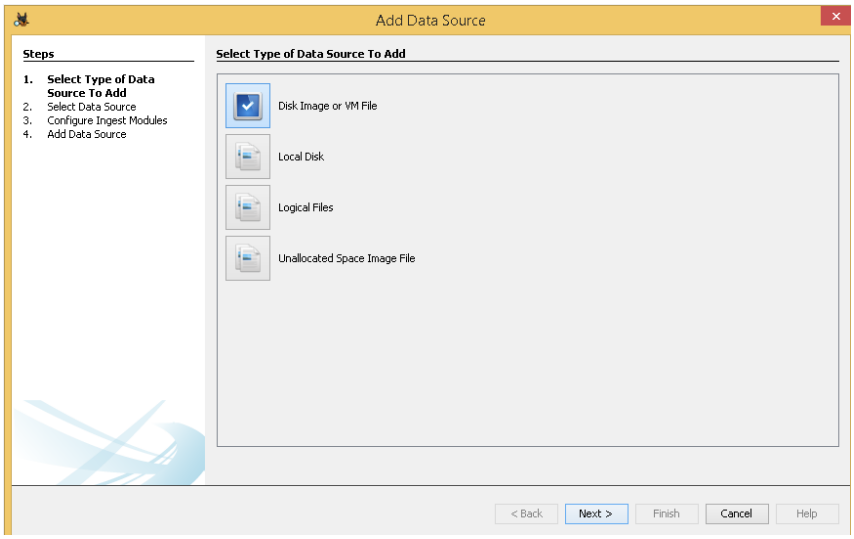
Gambar 4.44 *Form Case Information* Aplikasi Autopsy



Gambar 4.45 *Form Additional Information* Aplikasi Autopsy

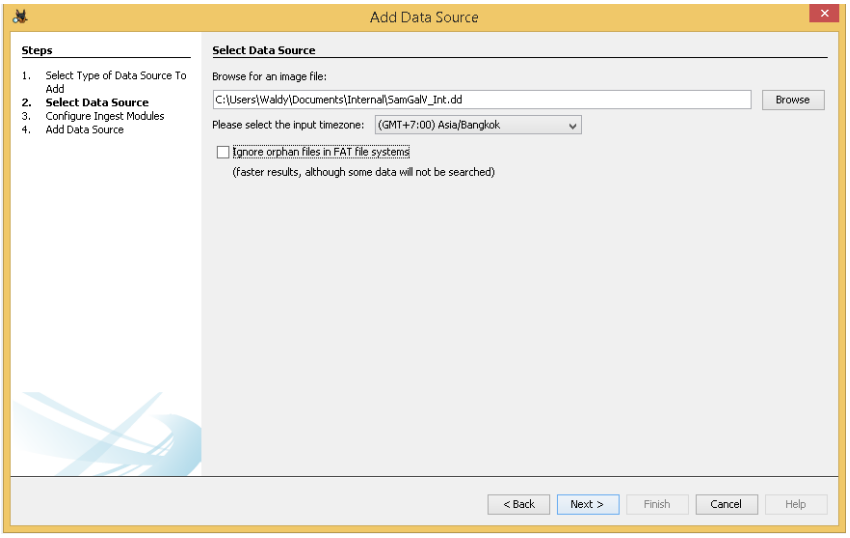
Pada Gambar 4.44 merupakan *form case information*. Pada *form* ini menjelaskan mengenai informasi kasus yang ingin dianalisis dari mengisikan nama kasus dan pemilihan letak direktori untuk menyimpan kasus tersebut yang akan menghasilkan sebuah dokumen, sedangkan pada Gambar 4.45 merupakan *form additional information* menjelaskan mengenai kasus ke berapa dan siapa nama pengguna aplikasi autopsy yang melakukan analisis tersebut. dari kedua *form* tersebut berfungsi untuk mendapatkan rekam analisis yang dapat dipertanggung jawabkan.

Setelah melakukan pengisian *form* untuk kepentingan analisis. selanjutnya masuk ke dalam tahap-tahap melakukan proses analisis dari *file image* yang dihasilkan aplikasi, tahap-tahap tersebut antara lain adalah menentukan data apa yang ingin dianalisis yaitu berupa *file image*, mengambil *file image* yang ingin dianalisis dari hasil proses akuisisi penyimpanan , menentukan *ingest modules* dalam aplikasi autopsy untuk kepentingan menganalisis, dan data *source* akan ditampilkan dalam aplikasi autopsy yang sudah terindetifikasi data-datanya. Untuk melihat tahap pertama yaitu menentukan data apa yang ingin dianalisis dapat dilihat pada Gambar 4.46.



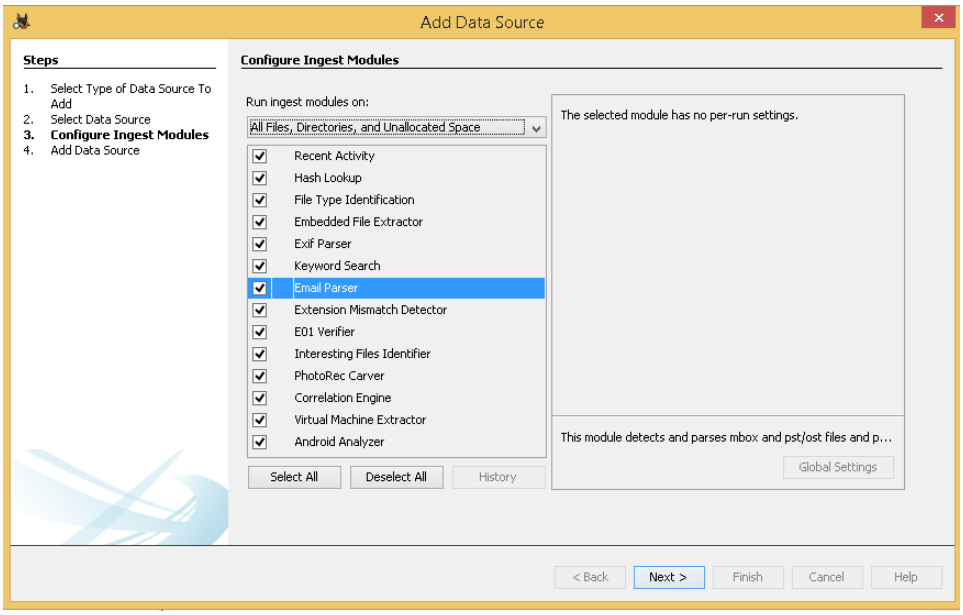
Gambar 4.46 *Form* menentukan Tipe Data yang Dianalisis

Pada Gambar 4.46 merupakan *form* dalam aplikasi Autopsy untuk memilih tipe data yang ingin dianalisis, Untuk file yang akan dianalisis adalah *file image* yang dihasilkan aplikasi. maka dari file tersebut kita dapat memilih pilihan 1 yaitu *disk image* karena file yang dihasilkan aplikasi bertipe *disk image* dan juga pilihan 1 memiliki fungsi untuk dapat menganalisis file tersebut secara menyeluruh, untuk pilihan 2-3 adalah pilihan untuk menganalisis file dari *localdisk* dan *logical files*, dan sedangkan pilihan 4 adalah pilihan untuk menganalisis dari *file image*, namun hanya dalam sector *unallocated space* dari *file image* yang akan dianalisis, setelah memilih pilihan 1, selanjutnya akan ada *form* mengambil *file image* yang dihasilkan aplikasi untuk dimasukan dalam aplikasi autopsy dan dapat dilakukan analisis dari file tersebut, *form* tersebut dapat dilihat pada Gambar 4.47.



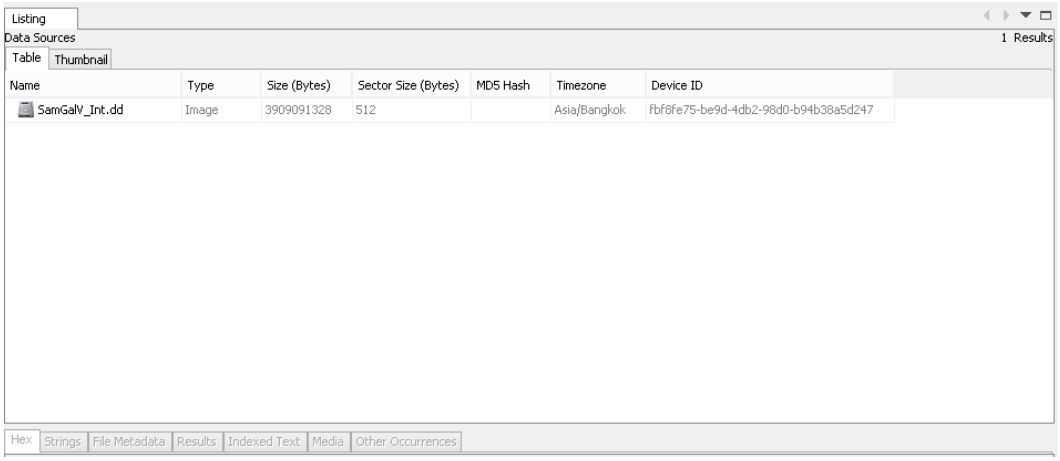
Gambar 4.47 *Form* Mengambil Data yang ingin Dianalisis.

Pada Gambar 4.47 merupakan *Form* untuk mengambil sebuah data *file image* yang dihasilkan aplikasi yang akan dimasukan dalam aplikasi autopsy untuk dilakukan analisis dari data tersebut, setelah data sudah dipastikan masuk dan terbaca oleh aplikasi autopsy, maka selanjutnya aplikasi menampilkan *form ingest modules* dan seorang analisisis akan mengisi untuk menentukan *form ingest modules* yang dibutuhkan untuk analisis dalam aplikasi autopsy dan dapat dilihat pada Gambar 4.48.



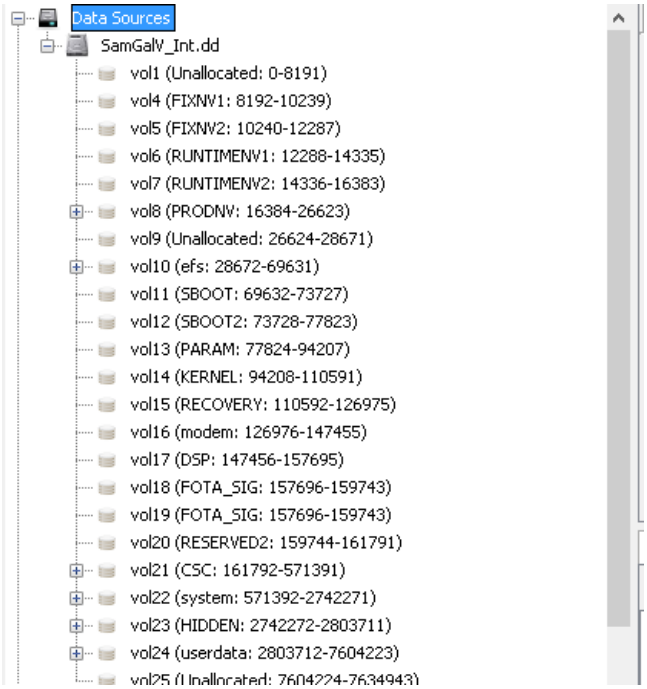
Gambar 4.48 *Form* penentuan *ingest modules* Aplikasi Autopsy.

Pada Gambar 4.48 merupakan *form* penentuan untuk memilih *ingest modules* dalam aplikasi autopsy yang berfungsi untuk membagi data yang dibutuhkan saat melakukan proses analisis dari *file image* yang sudah dimasukan dalam aplikasi autopsy agar tersusun dengan rapi dan dapat dianalisis secara menyeluruh dalam aplikasi autopsy, dari i*ngest modules* dalam aplikasi autopsy banyak pilihan yang memiliki fungsi untuk kepentingan analisis agar mudah dilakukan. Setelah penentuan *ingest modules*, maka aplikasi autopsy menjalankan *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 terbaca dalam aplikasi autopsy dan menjadi *data source* untuk proses analisis dengan terbagi beberapa komponen data dari fungsi *ingest modules* yang dijalankan aplikasi autopsy dapat dilihat pada Gambar 4.49.



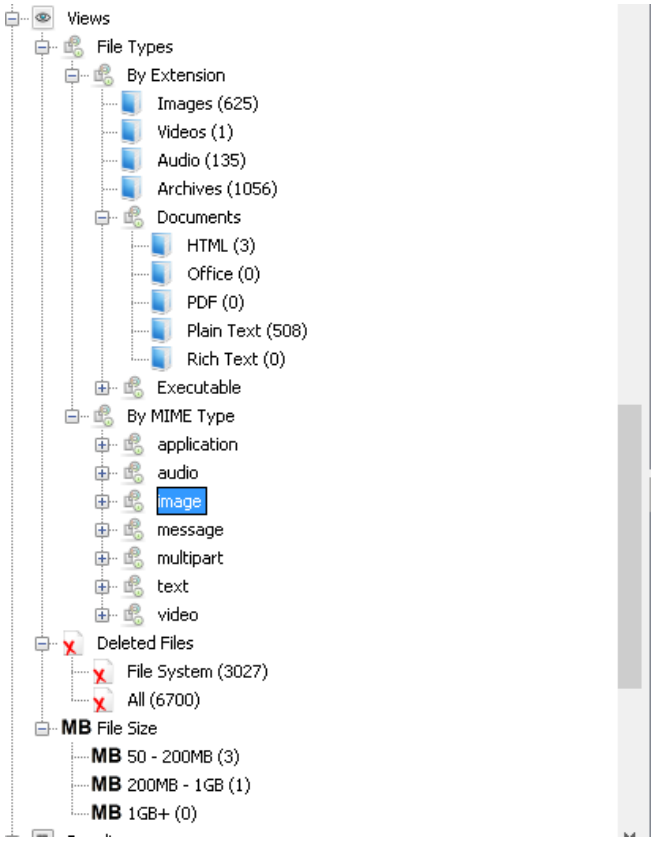
Gambar 4.49 Data *Source* dalam Aplikasi Autopsy

Pada Gambar 4.49 merupakan data *source* yang dihasilkan aplikasi autopsy dari *file image* yang telah dimasukan ke dalam aplikasi autopsy untuk melakukan proses analisis *file image* tersebut, namun setelah masuk dan terbaca sebagai data *source* aplikasi autopsy, maka aplikasi menjalankan proses untuk membaca semua isi *file image* agar tersusun dengan baik dan mudah untuk dianalisis. Setelah proses *data source* dalam aplikasi autopsy telah selesai maka data tersebut sudah dapat dianalisis dan tersusun dengan rapi. untuk melihat partisi dalam *file image* penyimpanan intenal ponsel android yang telah menjadi data *source* yang dihasilkan aplikasi autopsy untuk dilakukan analisis dapat dilihat pada Gambar 4.50.



Gambar 4.50 Partisi dalam Data Source aplikasi Autopsy.

Pada Gambar 4.50 merupakan daftar partisi-partisi dari *file image* penyimpanan internal ponsel android yang sudah dijadikan *data source* dalam aplikasi autopsy. Dari daftar partisi-partisi tersebut dapat mengetahui semua data-data yang terdapat dalam partisi tertentu yang dihasilkan dari *data source* dalam aplikasi autopsy dan dapat juga data tersebut diekstrak ke perangkat laptop yang akan dijadikan file barang bukti yang dihasilkan dari proses analisis. Dari semua daftar partisi yang terdapat dalam *file image* penyimpanan internal ponsel android. Ada beberapa partisi utama yang dijadikan bahan analisis oleh seorang forensika digital, partisi tersebut adalah partisi "system" pada vol 22 yang berfungsi untuk melihat data sistem yang digunakan dalam penyimpanan internal ponsel android. kedua adalah partisi "userdata" pada vol 22 yang berfungsi untuk melihat semua data-data yang tersimpan dalam penyimpanan internal ponsel android. Setelah melakukan analisis dari kedua partisi utama dalam penyimpanan internal ponsel android yang dihasilkan dari *data source* aplikasi autopsy, maka aplikasi juga memberikan fasilitas untuk melihat semua data-data yang memiliki *file type*. untuk melihat semua data berdasarkan *file type*-nya dan telah tersusun dengan rapi dapat dilihat pada Gambar 4.51.

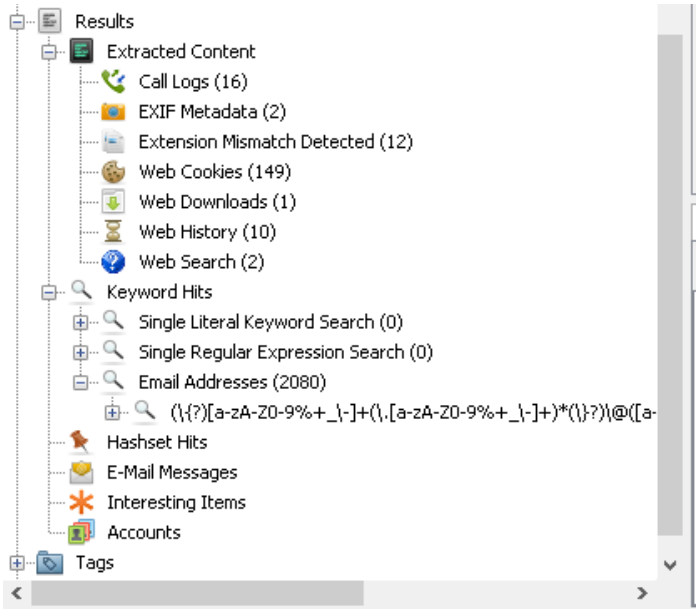


Gambar 4.51 Detail Data Source dari Aplikasi Autopsy

Pada Gambar 4.51 merupakan fasilitas dari autopsy untuk melihat semua detail data-data yang terdapat dalam penyimpanan internal ponsel android yang telah tersusun dengan rapi dan telah menjadi *data source* dalam aplikasi autopsy, dari detail *data source* tersebut terbagi dari beberapa komponen data dari penyimpanan internal ponsel android, data tersebut diantaranya berdasarkan dari *file type*, *delete files*, dan *file size* yang terbilang cukup besar.

Data berdasarkan *file types* terbagi menjadi 2 bagian data, bagian data pertama adalah data berdasarkan *extension* yang berfungsi untuk melihat semua data-data yang memiliki format seperti gambar, video, audio dan dokumen yang didapat dari proses analisis dan juga diekstak ke dalam perangkat laptop untuk dijadikan barang bukti dari hasil proses analisis, sedangkan bagian data kedua adalah data berdasarkan *MIME type* adalah data seperti aplikasi, pesan, dan lainnya.

Data berdasarkan *deleted file* berfungsi untuk melihat data apa saja yang sudah terhapus sebelumnya dalam penyimpanan internal ponsel android dari data *file system* dan juga data lainnya yang sudah terhapus. Dan terakhir data berdasarkan *file size* ynag berfungsi melihat data dalam penyimpanan internal ponsel android yang berukuran cukup besar dari ukuran 50-200MB, 200M -1GB dan 1GB lebih. Setelah melihat data-data terdapat dalam ponsel Android selanjutnya aplikasi autopsy juga memberikan fasilitan untuk melihat semua isi konten dalam ponsel android yang dapat dilihat pada Gambar 4.52.



Gambar 4.52 Isi Konten Android dalam Data Source Aplikasi Autopsy

Pada Gambar 4.52 merupakan hasil data dari konten yang terdapat dalam ponsel android, seperti histori panggilan, *web cookies*, *web search*, *web history*, dan *web download* dalam *web browser* dari data tersebut didapatkan hasil dari rekam jejak penggunaan ponsel android yang dapat di ekstrak untuk dijadikan barang bukti berupa dokumen. dan konten lainnya dari penggunaan email, melihat pesan email.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian dan pembahasan tugas akhir mengenai Analisis dan Perancangan Aplikasi *Forensic Imaging* pada Ponsel Android dengan memanfaatkan *Custom Recovery*, dapat disimpulkan bahwa hasil implementasi pengembangan aplikasi sesuai dengan perancangan yang sudah dibuat dan seluruh fungsionalitas sistem berjalan dengan baik, sebagai berikut:

a. Aplikasi menampilkan *progress bar* untuk mengetahui perkembangan perjalannya proses akuisisi penyimpanan secara *real-time* dan juga dapat mengakuisisi penyimpanan internal maupun eksternal baim itu salah satu penyimpanan maupun kedua penyimpanan secara bersamaan.
b. Aplikasi dengan memanfaatkan *custom recovery* dapat menjaga integritas data yang dihasilkan aplikasi agar tidak berubah saat dilakukan proses akuisisi.
c. Aplikasi terbukti dapat menjaga integritas data dari file yang dihasilkan, mendapatkan kecepatan transfer data lebih dari 1 MB/s saat mengakuisisi kedua penyimpanan secara bersamaan sedangkan lebih dari 2 MB/s saat mengakuisisi salah satu penyimpanan dan menghasilkan kelayakan untuk dapat digunakan oleh seorang forensika digital.

5.2 Saran

Untuk pengembangan aplikasi lebih lanjut, diharapkan Aplikasi *Forensic Imaging* Android dengan memanfaatkan *Custom Recovery* kedepannya dapat membuat fitur untuk dapat memilih perangkat 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. Selain itu perlu ditambahkan beberapa fitur yang dapat menggunakan format selain (.dd) dari *file image* yang dihasilkan aplikasi sehingga dapat dilakukan perbandingan kecepatan proses akuisisi penyimpanan ponsel Android.

DAFTAR PUSTAKA

Bertasi, P., & Zago, N. (2013). FASTDD: An open source forensic imaging tool. *Proceedings - 2013 International Conference on Availability, Reliability and Security, ARES 2013*, 485–492. https://doi.org/10.1109/ARES.2013.63

Cahyono, N. (2015). Pengertian Perancangan Sistem Informasi. https://doi.org/10.1007/1-4020-0612-8_769

Databoks. (2017). 73% Perangkat Mobile Global Menggunakan Android. Retrieved from https://databoks.katadata.co.id/datapublish/2017/10/09/73-perangkat-mobile-global-menggunakan-android

Forensics, C., & Santos, N. (2015). Mobile Forensics : Android.

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

Guido, M., Buttner, J., & Grover, J. (2016). Rapid differential forensic imaging of mobile devices. *Digital Investigation*, *18*, S46–S54. https://doi.org/10.1016/j.diin.2016.04.012

Heriyanto, A. P. (2016). *Mobile Phone Forensics: Theory: Mobile Phone Forensics dan Security Series*. Retrieved from https://books.google.co.id/books?id=erg5DgAAQBAJ&pg=PA120&lpg=PA120&dq=filesystems+dalam+android&source=bl&ots=RNiPipJ19b&sig=ecvBdSMo07KVManC5AJ-SJeq2yI&hl=id&sa=X&ved=2ahUKEwj9y-39x7rdAhWBLY8KHWggAsYQ6AEwDHoECAYQAQ#v=onepage&q=filesystems dalam android&

Krisis, L. B., Unep, D., Arsitektur, K., & Energi, H. (2012). Bab I, 1–10.

Murphy, C. A. (2012). Developing Process for Mobile Device Forensics, 1–9. Retrieved from http://www.mobileforensicscentral.com/mfc/documents/Mobile Device Forensic Process v3.0.pdf

Nafisah, S. (2003). Perancangan. *Pengertian Perancangan*, 11–48.

Oktaviani, D. M. (2015). Sistem Rekomendasi Penyewaan Sound System Pada Ud . Dyah Audio Berbasis Web Menggunakan Metode Euclidean Distance. *Artikel Skripsi*, 1–15.

Regan, J. E. (2009). THE FORENSIC POTENTIAL OF FLASH MEMORY.

Rohit Tamma, D. T. (2015). *Learning Android Forensics*. Retrieved from

https://www.packtpub.com/networking-and-servers/learning-android-forensics

Ruby Alamsyah. (2013). On Digital Forensics. Retrieved from http://ondigitalforensics.weebly.com/digital-forensics/comparison-defenisi-digital-forensics-forensika-digital-atau-yang-biasa-disebut-forensik-computer#.W-WS_dUzbIU

SORA N. (2015). Ketahui Pengertian Analisis Data Dan Tujuannya. Retrieved from http://www.pengertianku.net/2015/09/pengertian-analisis-data-dan-tujuannya.html

Srivastava, H., & Tapaswi, S. (2015). Logical acquisition and analysis of data from android mobile devices. *Information and Computer Security*, *23*(5), 450–475. https://doi.org/10.1108/ICS-02-2014-0013

Stone, D. (2006). Difference Between Disk Cloning & Disk Imaging. Retrieved from https://smallbusiness.chron.com/difference-between-disk-cloning-disk-imaging-68100.html

Teknologi.id. (2018). Jumlah Pengguna Smartphone di Seluruh Dunia dari 2014-2020. Retrieved from https://teknologi.id/insight/jumlah-pengguna-smartphone-di-seluruh-dunia-dari-2014-2020/

Yadi, I. Z., & Kunang, Y. N. (2014). Analisis forensik pada Android. *Konferensi Nasional Ilmu Komputer (KONIK)*, 141–148.

LAMPIRAN