

***FORENSIC IMAGING APPLICATION
USING RASPBERRY PI***



Disusun Oleh:

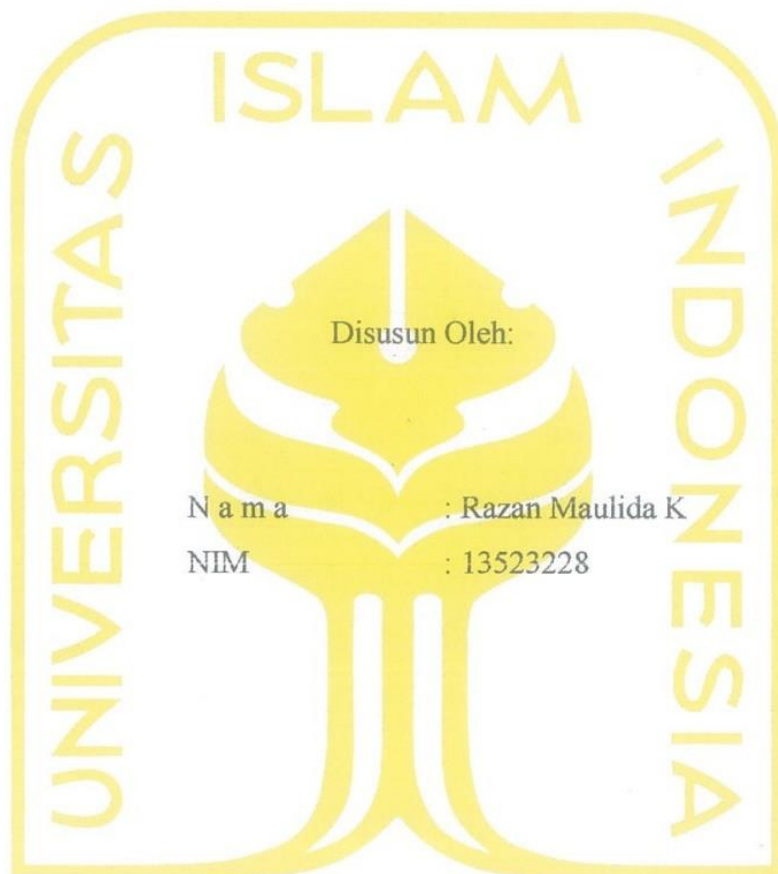
N a m a : Razan Maulida K
NIM : 13523228

**PROGRAM STUDI TEKNIK INFORMATIKA-PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2018**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

*FORENSIC IMAGING APPLICATION
USING RASPBERRY PI*

TUGAS AKHIR



Yogyakarta, 12 November 2018

Pembimbing,

البعث الاسلامي
الانديسي

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

HALAMAN PENGESAHAN DOSEN PENGUJI

**FORENSIC IMAGING APPLICATION
USING RASPBERRY PI****TUGAS AKHIR**

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 Desember 2018

Tim Penguji

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

Anggota 1

Ari Sujarwo, S.Kom., M.I.T.

Anggota 2

Kholid Haryono, S.T., 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 : Razan Maulida K

NIM : 13523228

Tugas akhir dengan judul:

***FORENSIC IMAGING APPLICATION
USING RASPBERRY PI***

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, 12 November 2018


(Razan Maulida K)

HALAMAN PERSEMBAHAN

Segala puji peneliti panjatkan kepada Allah Ar-Rahman Ar-Rahim sebagai wujud syukur atas seluruh limpahan hidayah, rahmat, dan nikmat-Nya yang tak terhitung jumlahnya. Shalawat dan salam senantiasa peneliti panjatkan untuk baginda nabi besar Muhammad ﷺ, karena dengan uswah hasanah dan sunnahnya kita meniti jalan pengabdian kepada Allah Rabb alam semesta.

Tugas akhir ini peneliti persembahkan kepada mereka yang selalu memberikan dukungan dan bantuannya sehingga peneliti mampu menyelesaikan tugas akhir ini. Kepada kedua orang tua, baik bapak dan ibu yang senantiasa memanjatkan doanya yang tak henti-henti. Kepada ibu yang selalu memberikan dukungan baik dalam keadaan susah maupun senang. Kepada bapak yang selalu memberikan nasihat bijaknya sehingga peneliti mampu menyelesaikan penelitian ini. Kepada kakak dan adik peneliti yang juga selalu memberikan dukungannya sehingga peneliti terus bersemangat dalam menyelesaikan tugas akhir ini.

Kepada rekan-rekan peneliti yang senantiasa memberikan bantuan ketika mengalami kendala di dalam proses pengerjaan penelitian ini. Maka dari itu peneliti ucapkan banyak terima kasih atas segala bentuk bantuan dan doanya semoga kita semua selalu berada dalam lindungan Allah SWT.

HALAMAN MOTO

“Hai sekalian manusia, kalianlah yang membutuhkan kepada Allah; dan Allah Dialah yang Maha Kaya (tidak memerlukan sesuatu) lagi Maha Terpuji”

(QS. Fathir:15)

“Hai orang-orang beriman apabila dikatakan kepadamu: ‘berlapang-lapanglah dalam majlis’, maka lapangkanlah niscaya Allah akan memberi kelapangan untukmu. Dan apabila dikatakan: ‘Berdirilah kamu’, maka berdirilah, niscaya Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah Maha Mengetahui apa yang kamu kerjakan”

(QS. Al-Mujadillah:11)

KATA PENGANTAR



Assalamu'alaikum Warahmatullahi Wabarakatuh

Dengan menyebut nama Allah SWT yang maha pengasih lagi maha penyayang, penulis panjatkan puja dan puji syukur atas kehadiran-Nya yang telah melimpahkan rahmat, hidayah, serta inayah-nya kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “*Forensic Imaging Application Using Raspberry Pi*”. Tak lupa shalawat dan salam kita curahkan kepada baginda Nabi besar Muhammad SAW, yang telah membawa kita dari zaman jahiliyah menuju zaman terang benderang seperti sekarang ini.

Laporan tugas akhir ini disusun sebagai salah satu syarat yang harus dipenuhi untuk memperoleh gelar Sarjana Strata-1 (S1) di Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Selain itu tugas akhir ini juga peneliti manfaatkan sebagai sarana untuk menambah wawasan peneliti sekaligus menerapkan keilmuan yang selama ini peneliti dapatkan di bangku perkuliahan.

Dalam pembuatan tugas akhir ini, penulis mendapatkan banyak bantuan, masukan, bimbingan, dan dukungan serta doa dari berbagai pihak. Untuk itu dengan segala kerendahan hati izinkanlah penulis menyampaikan rasa terima kasih dan penghargaan yang setinggi-tingginya. Penulis menyampaikan rasa terima kasih dan penghargaan tersebut kepada:

1. Orang tua, kakak, dan adik penulis yang senantiasa memberikan doa, nasihat, dan dukungannya sehingga penulis mampu menyelesaikan tugas akhir ini.
2. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Fietyata Yudha S.Kom., M.Kom selaku Dosen Pembimbing tugas akhir yang tak kenal lelah di dalam membimbing penulis sehingga penulis mampu menyelesaikan aplikasi *Forensic Imaging Application Using Raspberry Pi*.
4. Teman serumah penulis yang sudah penulis anggap sebagai keluarga sendiri. Burhan, Agus, Aji, Argo dan Sulis yang selalu menyemangati penulis. Waldy sebagai teman diskusi sehingga penulis mampu menyelesaikan tugas akhir ini.
5. Rekan serta sahabat yang senantiasa membantu dan menyemangati penulis di dalam menyelesaikan tugas akhir ini.

6. Kepada semua pihak yang telah mendoakan dan membantu penulis sehingga penulis mampu menyelesaikan aplikasi *Forensic Imaging Application Using Raspberry Pi*.

Akhirnya penulis pun menyadari bahwa dalam penyusunan laporan tugas akhir ini masih terdapat banyak kekurangan. Maka dari itu, penulis menerima semua kritik dan saran yang sifatnya membangun demi hasil yang lebih baik dan semoga laporan tugas akhir ini bermanfaat bagi kita semua, terutama bagi penulis. Amiin.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 12 November 2018

(Razan Maulida K)

SARI

Cybercrime atau kejahatan siber adalah sebuah tindak pidana kejahatan yang merugikan orang lain dengan memanfaatkan komputer sebagai media kejahatannya. Tren kejahatan siber terus mengalami kenaikan, menurut data kepolisian Republik Indonesia jumlah kejahatan siber yang ditangani Polri pada tahun 2017 mengalami kenaikan sebesar 3% jika dibandingkan dengan kejahatan siber yang ditangani Polri pada tahun 2016. Untuk membuktikan sebuah kejahatan siber maka diperlukan barang bukti sebagai pembuktian di mata hukum. Untuk mendapatkan barang bukti kejahatan siber yang tertinggal di media penyimpanan dibutuhkan aplikasi yang mampu melakukan akusisi terhadap media penyimpanan. Berdasarkan uraian masalah diatas, peneliti mengangkat judul "*Forensic Imaging Application Using Raspberry Pi*".

Aplikasi ini diharapkan mampu melakukan akusisi terhadap media penyimpanan baik itu penyimpanan *hardisk* ataupun *flashdrive*. Format *file* yang dihasilkan oleh aplikasi *Forensic Imaging Application Using Raspberry Pi* ini adalah format tipe RAW yaitu *.dd*. Selain itu, aplikasi ini juga mampu menghasilkan dokumen *file logging* yang berisikan informasi detail terkait barang bukti yang diakusisi. Untuk menjaga integritas *file* hasil akusisi, maka aplikasi ini juga menjalankan proses *hashing* baik itu MD5 dan SHA512 sehingga pengguna merasa yakin dengan *file* hasil akusisi aplikasi ini. Aplikasi yang peneliti buat telah peneliti uji performa untuk menilai sejauh mana aplikasi dapat bekerja, aplikasi *forensic imaging* ini memiliki kecepatan transfer data sebesar 1,85 MB/s dan memiliki nilai skala likert sebesar 4,04 dari skala 1 sampai 5 yang mana itu termasuk ke dalam klasifikasi layak.

Kata kunci: Aplikasi *Python*, Proses Akusisi.

GLOSARIUM

GUI	Antarmuka aplikasi yang ditampilkan dalam bentuk grafis agar mempermudah pengguna aplikasi.
Flash drive	Media penyimpanan berupa USB yang memiliki bentuk yang kecil.
Raspbian	Sistem operasi yang digunakan Raspberry Pi.
String	Tipe data yang digunakan di dalam kode program yang menampung nilai gabungan dari huruf, angka, spasi, dan berbagai karakter.
Integer	Tipe data yang digunakan di dalam kode program yang menampung nilai bilangan bulat.
Interpreter	Perangkat lunak yang digunakan untuk melakukan eksekusi kode program yang dibuat oleh programmer.
Thread	Modul Python yang digunakan agar kode program berjalan terpisah dari kelas utamanya.
Platform	Arsitektur sistem atau aplikasi.

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	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan.....	6
BAB II LANDASAN TEORI.....	8
2.1 Penelitian Terdahulu.....	8
2.2 Cybercrime	16
2.3 Komputer Forensik.....	17
2.3.1 Forensic Imaging.....	18
2.4 File Ekstensi Hasil Akusisi.....	19
2.5 Python.....	20
BAB III METODOLOGI PENELITIAN	22
3.1 Metodologi Penelitian	22
3.2 Analisis Kebutuhan Sistem.....	23
3.2.1 Analisis Kebutuhan Perangkat Keras.....	23
3.2.2 Analisis Kebutuhan Perangkat Lunak.....	27

3.2.3 Analisis Kebutuhan Masukan	31
3.2.4 Analisis Kebutuhan Proses.....	31
3.2.5 Analisis Kebutuhan Keluaran	32
3.3 Perancangan Aplikasi	33
3.3.1 Flowchart	33
3.3.2 Pseudocode.....	35
3.3.3 Desain Antarmuka.....	40
3.4 Pengujian Skala Likert	42
BAB IV HASIL DAN PEMBAHASAN	46
4.1 Hasil.....	46
4.1.1 Perangkat Keras	46
4.1.2 Perangkat Lunak	47
4.2 Pembahasan Implementasi Aplikasi.....	49
4.2.1 Pembahasan Implementasi Kode Program	49
4.2.2 Pembahasan Implementasi Antarmuka.....	51
4.2.3 Pembahasan Implementasi Proses Akusisi	53
4.3 Pengujian	59
4.3.1 Blackbox	59
4.3.2 Pengujian Performa Aplikasi	68
4.3.3 Perhitungan Skala Likert.....	76
BAB V KESIMPULAN DAN SARAN	78
5.1 Kesimpulan.....	78
5.2 Saran	78
DAFTAR PUSTAKA	79
LAMPIRAN.....	81

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	8
Tabel 3.1 Tabel <i>Library</i> Tampilan 1	35
Tabel 3.2 Tabel <i>Library</i> Tampilan 2	38
Tabel 3.3 Nilai Skala Likert	43
Tabel 3.4 Rentang Skala Likert	44
Tabel 3.5 Pertanyaan Kuesioner	44
Tabel 4.1 Pengujian Performa Aplikasi	69
Tabel 4.2 Perhitungan Skala Likert	76

DAFTAR GAMBAR

Gambar 3.1 Raspberry Pi 3 Model B	24
Gambar 3.2 Power Adapter.....	25
Gambar 3.3 Keyboard dan Mouse	25
Gambar 3.4 LCD Monitor.....	26
Gambar 3.5 <i>Micro SD</i>	27
Gambar 3.6 Raspbian.....	28
Gambar 3.7 Qt Designer	30
Gambar 3.8 <i>Flowchart Forensic Imaging Application Using Raspberry Pi</i>	34
Gambar 3.9 <i>Pseudocode</i> Tampilan 1	37
Gambar 3.10 <i>Pseudocode</i> Tampilan 2	40
Gambar 3.11 Tampilan Antarmuka 1	41
Gambar 3.12 Tampilan Antarmuka 2	41
Gambar 4.1 Raspberry Pi 3.....	47
Gambar 4.2 Raspberry Pi Dengan LCD Monitor	47
Gambar 4.3 Halaman Tabel Penyimpanan	48
Gambar 4.4 Halaman Pegisian Data	49
Gambar 4.5 <i>Library</i> Kode Pemrograman Halaman 1	50
Gambar 4.6 Fungsi Untuk Menampilkan Penyimpanan.....	50
Gambar 4.7 <i>Library</i> Kode Pemrograman Pada Halaman 2	50
Gambar 4.8 <i>Interface 1</i>	51
Gambar 4.9 <i>Interface 2</i>	52
Gambar 4.10 Pemilihan Akusisi Penyimpanan	54
Gambar 4.11 Pengisian Kolom Masukan	54
Gambar 4.12 Progres Akusisi 51%	55
Gambar 4.13 <i>File Logging</i>	55
Gambar 4.14 Progres Akusisi 100%	56
Gambar 4.15 Proses Akusisi Selesai.....	56
Gambar 4.16 Keluaran Aplikasi	57
Gambar 4.17 <i>File Logging</i>	57
Gambar 4.18 Fungsi Proses Akusisi	58
Gambar 4.19 Fungsi <i>File Logging</i> dan <i>Hashing</i>	59
Gambar 4.20 Pengujian Fungsi <i>Scan</i>	60

Gambar 4.21 Pengujian Fungsi <i>Scan</i>	60
Gambar 4.22 Pengujian Fungsi <i>Scan</i>	61
Gambar 4.23 Pengujian Fungsi <i>Exit</i>	61
Gambar 4.24 Pengujian Fungsi <i>Exit</i>	62
Gambar 4.25 Pengujian Tombol Direktori	62
Gambar 4.26 Pengujian Tombol Direktori	63
Gambar 4.27 Pengujian Fungsi <i>Exit</i>	63
Gambar 4.28 Pengujian Fungsi <i>Exit</i>	63
Gambar 4.29 Pengujian Fungsi <i>Back</i>	64
Gambar 4.30 Pengujian Tombol <i>Back</i>	64
Gambar 4.31 <i>Error Handling</i> Skema 1	65
Gambar 4.32 <i>Error Handling</i> Skema 2	66
Gambar 4.33 <i>Error Handling</i> Skema 3	66
Gambar 4.34 <i>Error Handling</i> Skema 4	67
Gambar 4.35 <i>Error Handling</i> Skema 5	67
Gambar 4.36 <i>Error Handling</i> Skema 6	68
Gambar 4.37 Penggunaan Memori Perangkat Forensik Pada Percobaan 1	70
Gambar 4.38 Penggunaan CPU Perangkat Forensik Pada Percobaan 1	70
Gambar 4.39 Penggunaan Memori Perangkat Forensik Pada Percobaan 2	71
Gambar 4.40 Penggunaan CPU Perangkat Forensik Pada Percobaan 2	71
Gambar 4.41 Penggunaan Memori Perangkat Forensik Pada Percobaan 3	72
Gambar 4.42 Penggunaan CPU Perangkat Forensik Pada Percobaan 3	72
Gambar 4.43 Penggunaan Memori Perangkat Forensik Pada Percobaan 4	72
Gambar 4.44 Penggunaan CPU Perangkat Forensik Pada Percobaan 4	73
Gambar 4.45 Penggunaan Memori Perangkat Forensik Pada Percobaan 5	73
Gambar 4.46 Penggunaan CPU Perangkat Forensik Pada Percobaan 5	74
Gambar 4.47 Penggunaan Memori Perangkat Forensik Pada Percobaan 6	74
Gambar 4.48 Penggunaan CPU Perangkat Forensik Pada Percobaan 6	74
Gambar 4.49 Grafik Kecepatan Transfer Data	75

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pesatnya perkembangan teknologi informasi dan komputer sangat terasa di tengah kehidupan masyarakat. Karena dengan berkembangnya teknologi informasi dan komputer dapat mempermudah pekerjaan manusia, mulai dari anak-anak yang masih duduk di bangku sekolah hingga orang dewasa memanfaatkannya guna membantu menyelesaikan pekerjaan mereka. Namun ada saja pihak-pihak yang memanfaatkan kemajuan teknologi informasi dan komputer ini untuk berbuat hal-hal negatif yang sifatnya mengarah kepada modus kejahatan. Selanjutnya istilah kejahatan yang memanfaatkan komputer disebut *cybercrime*. Dilansir dari situs *kominfo.go.id* Wakil Kepala Kepolisian Republik Indonesia Komisaris Jendral Syafruddin mengatakan bahwa Indonesia masuk dalam jajaran dua besar negara di dunia dengan kejahatan siber tertinggi. Kejahatan siber di Indonesia tertinggi ke dua di dunia setelah Jepang. Total serangan kejahatan siber di Indonesia mencapai 90 juta (Hutabarat, 2018).

Cybercrime adalah kejahatan yang dilakukan oleh pihak yang tidak bertanggung jawab dengan tujuan untuk merusak, memodifikasi dan mengeliminasi data (Taufiqurrohman, Widiyasono, & Mubarak, 2017). Sedangkan menurut Muhammad Nur Al-Azhar dalam bukunya yang berjudul *Digital Forensic, cybercrime* merupakan kejahatan yang menggunakan komputer sebagai alat utama untuk melakukan aksi kejahatannya, misalnya *defacement* (pengubahan halaman-halaman suatu situs secara ilegal), *denial distributed of service* (membuat suatu sistem tidak berjalan atau berfungsi sebagaimana mestinya setelah dibanjiri data oleh sekian banyak komputer yang telah terinfeksi dan menjadi *robot network*), *intrusion* (masuk secara ilegal ke dalam suatu sistem), dan lain-lain (Al-Azhar, 2012). Guna menangkal *cybercrime* maka harus ada perangkat hukum yang jelas, dalam hal ini perangkat hukum dibuat dalam bentuk aturan dan perundang-undangan. Disisi lain dari aspek ilmiah dan teknis juga diperlukan mekanisme pembuktian. Dalam hal ini komputer forensik adalah salah satu bidang yang dapat membantu dalam upaya pembuktian *cybercrime*.

Menurut Muhammad Nur Al-Azhar dalam bukunya yang berjudul *Digital Forensic*, komputer forensik adalah ilmu forensik yang berkaitan dengan pemeriksaan dan analisis barang bukti elektronik berupa komputer pribadi (*personal computer-PC*), *laptop/notebook*, *netbook*, dan *tablet*. Pemeriksaan terhadap jenis barang bukti ini biasanya berkaitan dengan *file*

recovery, yaitu suatu metode untuk mengambil *file logical* atau memunculkan kembali *file* yang sudah dihapus (*deleted*) maupun hilang (*lost*) dikarenakan tidak tercatat lagi di *file system*. *File-file* tersebut diperlukan untuk membuktikan kejahatan yang terjadi dan menghubungkannya dengan pelaku (Al-Azhar, 2012). Peran komputer forensik menjadi vital guna mengungkap setiap kejahatan siber, dilansir dari berita elektronik pada situs *nasional.kompas.com*, Kapolri Jendral Pol Tito Karnavian mengatakan, ada peningkatan jumlah perkara yang menyangkut kejahatan dunia maya atau *cybercrime*. Jika dibandingkan dengan tahun 2016, pada tahun 2017 mengalami kenaikan sebanyak tiga persen. Pada 2016 kejahatan siber yang ditangani Polri sebanyak 4.931 kasus, dan yang berhasil diselesaikan sebanyak 1.119 kasus, sedangkan pada tahun 2017 kejahatan siber yang ditangani Polri sebanyak 5.061 kasus, dan yang berhasil diselesaikan sebanyak 1.368 kasus (Movanita, 2017).

Untuk mengakuisisi dan menganalisa barang bukti digital dari kasus *cybercrime* diperlukan perlakuan khusus, karena kesalahan kecil pada penanganan barang bukti digital dapat membuat barang bukti digital tidak diakui secara hukum dan secara otomatis tidak dapat dijadikan sebagai alat bukti yang sah di pengadilan. Berdasarkan UU No. 19 Tahun 2016 tentang Informasi dan Transaksi Elektronik (UU ITE) pasal 43 ayat 5 huruf (J) berbunyi, “Meminta bantuan ahli yang diperlukan dalam penyidikan terhadap tindak pidana di bidang Teknologi Informasi dan Transaksi Elektronik” (Kominfo, 2016). Salah satu cara agar mengamankan barang bukti digital dengan menyalin data dari media penyimpanan secara *Bitstream Image* dan menempatkannya pada tempat yang aman. Teknik mengambil bit demi bit data dari media penyimpanan fisik disebut *Cloning Disk*, dan hasil dari *cloning* tersebut disebut *imaging*.

Bitstream adalah metode penyimpanan digital dengan mengkopi setiap bit demi bit dari data orisinal, termasuk *file* yang tersembunyi (*hidden file*), *file* temporer (*temp file*), *file* yang terdefragmen (*fragmen file*), dan *file* yang belum ter-*overwrite* (Prayudi & Afrianto, 2007). *Imaging* merupakan hasil *cloning* dari sebuah *hard disk* dalam komputer forensik yang nantinya hasil *cloning* tersebut dapat digunakan sebagai barang bukti atas sebuah tindak kejahatan siber. Dilansir dari situs *whatis.techtarget.com*, *forensic imaging* adalah menyalin secara langsung dari perangkat penyimpanan fisik secara bit demi bit, sektor demi sektor termasuk semua *file*, *folder*, *unallocated*, *free* dan *slack space*. *Forensic imaging* tidak hanya menyalin *file-file* yang terlihat saja, tetapi juga menyalin *file-file* yang telah terhapus yang tertinggal di *slack* dan *free space* dari media penyimpanan fisik (Rouse, 2017).

Untuk dapat melakukan proses *cloning disk*, seorang investigator tentunya membutuhkan perangkat maupun aplikasi yang dapat membantu pekerjaannya. Saat ini ada banyak aplikasi *forensic imaging* yang dapat digunakan, mulai dari yang berbayar hingga yang gratis. Secara garis besar perangkat untuk kepentingan forensik komputer dibedakan berdasarkan *hardware* dan *software*, namun baik dari sisi *hardware* dan *software* perangkat forensik komputer diharapkan memenuhi 5 fungsi yaitu, untuk kepentingan akusisi (*acquisition*), validasi dan diskriminasi (*validation and discrimination*), ekstraksi (*extraction*), rekonstruksi (*reconstruction*) dan pelaporan (*reporting*) (Prayudi & Afrianto, 2007). Dilansir dari situs *forensicswiki.com*, berikut beberapa aplikasi yang dapat digunakan untuk *forensic imaging* diantaranya: guymager, ewfacquire, Adepto, aimage, AIR, dcfldd, dd, dan masih banyak lagi. Untuk membuat sebuah aplikasi GUI *forensic imaging* dapat memanfaatkan Raspberry Pi, karena sifatnya Raspberry Pi yang fleksibel dan mudah dibawa sehingga sangat cocok untuk mendukung kegiatan *forensic imaging*.

Raspberry Pi adalah mini komputer yang berbasis linux dengan harga yang murah, dan memiliki ukuran yang kecil sehingga mudah dibawa. (Taufiqurrohman, Widiyasono, & Mubarak 2017) dalam jurnalnya mengatakan, Raspberry Pi adalah komputer berukuran kartu kredit yang dikembangkan di Inggris oleh yayasan Raspberry Pi, kemudian Raspberry Pi dilengkapi dengan *port USB, LAN port, audio jack, HDMI port, Micro SD card*, serta *pin-pin* untuk menghubungkan dengan komponen lain seperti *button*, dan LED. Sedangkan menurut Carrie Anne Philbin dalam bukunya yang berjudul *Adventures in Raspberry Pi* mengatakan, Raspberry Pi adalah komputer dengan ukuran yang sangat kecil, hanya sebesar kartu kredit, dan Raspberry Pi dapat dibangun dan dikontrol sesuai kebutuhan pengguna (Philbin, 2014).

(Olson & Shashidhar, 2016) dalam jurnalnya yang berjudul “*Low Budget Forensic Drive Imaging Using Arm Based Single Board Computers*” melakukan penelitian dengan membuat perangkat *forensic imaging* yang *low budget* yang memanfaatkan *single board computer* Raspberry Pi 2 dan Odroid XU4. Digunakannya dua *single board computer* bertujuan untuk membandingkan kecepatan ketika melakukan akusisi terhadap *drive* yang akan di *cloning*. Perangkat *forensic imaging* yang dibuat oleh peneliti kemudian diuji cobakan dan hasilnya berhasil mendapatkan *image drive* dari empat *drive* yang berbeda. Disini peneliti memberikan saran untuk melakukan *upgrade single board computer* dari Raspberry Pi 2 menjadi Raspberry Pi 3 agar perangkat *forensic imaging* tersebut bekerja lebih efektif.

Penelitian ini dilakukan untuk membuat perangkat *forensic imaging* dengan memanfaatkan *single board computer* Raspberry Pi sebagai papan sirkuitnya. Kemudian untuk

mengoperasikan perangkat *forensic imaging* tersebut, maka dibuatkan aplikasi berbasis *Graphical User Interface* (GUI). Dengan dilakukannya penelitian ini diharapkan peneliti mampu membuat perangkat *forensic imaging* yang dapat melakukan akuisisi media penyimpanan dengan menggunakan Raspberry Pi sebagai *single board computer* nya.

1.2 Rumusan Masalah

Berdasarkan latar masalah diatas, penulis merumuskan masalah sebagai berikut:

- a. Bagaimana membuat perangkat *forensic imaging* dengan menggunakan *single board computer* Raspberry Pi.
- b. Bagaimana merancang aplikasi berbasis GUI untuk mengoperasikan perangkat *forensic imaging* tersebut.
- c. Apakah penggunaan *flash drive* dengan ukuran kapasitas penyimpanan yang sama tetapi merek berbeda dapat mempengaruhi kecepatan transfer data akuisisi dan berapa nilai kecepatan rata-rata transfer data akuisisi aplikasi.

1.3 Batasan Masalah

Agar pengerjaan lebih terarah, dalam penelitian ini terdapat beberapa batasan masalah, diantaranya:

- a. *Imaging* media penyimpanan.
- b. Format file yang dihasilkan .dd.
- c. *Single board computer* yang digunakan adalah Raspberry Pi 3.
- d. Menggunakan Bahasa pemrograman *Python*.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai pada penelitian ini yaitu:

- a. Membuat perangkat *forensic imaging* dengan menggunakan *single board computer* Raspberry Pi .
- b. Membuat aplikasi berbasis GUI untuk mengoperasikan perangkat *forensic imaging*.
- c. Mengetahui kecepatan transfer data akuisisi perangkat *forensic imaging*.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini antara lain:

- a. Bagi Peneliti:

1. Dapat mengimplementasikan ilmu pengetahuan yang telah diperoleh di bangku kuliah.
 2. Mendapatkan pembelajaran baru mengenai Bahasa pemrograman *Python*.
 3. Menambah pengetahuan dan pemahaman mengenai ilmu *forensic* khususnya *forensic imaging*.
- b. Bagi Masyarakat:
1. Perangkat yang dibuat dapat digunakan *forensic investigator* untuk melakukan akuisisi terhadap barang bukti.
 2. Perangkat yang dibuat bersifat fleksibel dan *portable* sehingga mudah dibawa oleh *forensic investigator*.

1.6 Metodologi Penelitian

Adapun metodologi penelitian yang digunakan dalam pembuatan aplikasi *forensic imaging* ini, yaitu:

a. Studi literatur

Pada tahap ini peneliti mengumpulkan dan mempelajari berbagai macam jurnal, karya ilmiah, artikel ilmiah dan buku yang dapat mendukung dan dapat peneliti jadikan sebagai referensi tugas akhir peneliti. Untuk mendapatkan berbagai macam jurnal, karya ilmiah, artikel ilmiah, dan buku yang peneliti butuhkan untuk dijadikan referensi, peneliti melakukannya dengan dua cara. Cara pertama dengan mengunjungi perpustakaan kampus UII, dan cara yang kedua dengan cara mengakses berbagai portal jurnal secara *online* melalui internet.

b. Perancangan aplikasi

Sebelum masuk pada tahap pembuatan aplikasi, peneliti terlebih dahulu melakukan tahap perancangan aplikasi. Pada tahap ini peneliti melakukan analisis kebutuhan sistem yang meliputi analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak, analisis kebutuhan masukan, analisis kebutuhan proses, dan analisis kebutuhan keluaran. Analisis yang dilakukan mengacu kepada referensi yang peneliti dapatkan pada tahap studi literatur. Selanjutnya pada tahap ini juga peneliti melakukan perancangan aplikasi yang meliputi perancangan alur kerja sistem atau *flowchart*, perancangan *pseudocode* program, dan perancangan desain antarmuka aplikasi. Untuk melakukan desain antarmuka aplikasi peneliti menggunakan QT Designer.

c. Pembuatan aplikasi

Pada tahap ini peneliti menuangkan ide serta gagasan yang telah dirancang dan dianalisa pada tahap perancangan aplikasi untuk dituangkan ke dalam proses pembuatan aplikasi. Untuk alur kerja sistem atau aplikasi, peneliti mengacu kepada *flowchart* yang telah dibuat pada tahap perancangan aplikasi. Peneliti menggunakan Python Shell sebagai *text editor* tempat dimana peneliti menuangkan kode program aplikasi.

d. Pengujian aplikasi

Untuk melihat dan menilai apakah aplikasi *forensic imaging* yang peneliti buat sudah berjalan dengan baik dan sesuai harapan, maka perlu dilakukan pengujian terhadap aplikasi *forensic imaging*. Pengujian aplikasi *forensic imaging* dilakukan dengan dua cara, yaitu dengan menggunakan teknik *blackbox* dan yang kedua dengan cara menggunakan kuesioner untuk dibagikan kepada responden yang memang mengerti dengan bidang forensik. Pengujian dengan teknik *blackbox* dilakukan dengan cara menguji setiap fungsi dan tombol yang terdapat pada aplikasi, dengan pengujian *blackbox* ini dapat terlihat apakah semua fungsi dan tombol yang terdapat pada aplikasi dapat berjalan dengan baik atau tidak. Sedangkan pengujian dengan kuesioner dilakukan untuk melihat sejauh mana kepuasan pengguna aplikasi terhadap aplikasi *forensic imaging* yang peneliti buat. Pengujian kuesioner dilakukan dengan cara membagikan kuesioner melalui *google form* kepada mahasiswa yang berada dilingkungan kampus FTI UII. Mahasiswa yang peneliti pilih adalah mahasiswa S-1 yang mengambil kelas Komputer Forensik, dan juga mahasiswa S-2 Forensik Teknik Informatika. Selanjutnya hasil kuesioner yang telah diisi oleh responden akan dihitung dengan menggunakan persamaan skala likert.

1.7 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk mempermudah dalam memahami tentang penelitian yang dijalankan. Secara garis besar, sistematika penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini memuat latar belakang permasalahan, rumusan masalah, menentukan batasan masalah yang akan dibahas, menjelaskan tujuan penelitian dan manfaat penelitian, metodologi penelitian, dan penjelasan terkait sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini berisi berbagai teori yang peneliti gunakan sebagai landasan teori untuk menyelesaikan penelitian yang peneliti lakukan. Pada bab ini berisi penelitian terdahulu, teori *cybercrime*, teori komputer forensik, teori *file* ekstensi hasil akusisi, teori Bahasa pemrograman *Python* dan teori *forensic imaging*.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang metodologi penelitian, yaitu proses bagaimana cara peneliti mengumpulkan data yang dapat peneliti gunakan untuk memperkuat gagasan serta dapat dijadikan landasan teori. Kemudian pada bab ini juga berisi berbagai macam analisis yang dilakukan peneliti guna mendukung kelancaran proses pembuatan aplikasi. Selanjutnya ada rancangan aplikasi agar memberikan gambaran secara jelas kepada peneliti terkait alur kerja sistem.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini membahas hasil yang berhasil dicapai oleh peneliti dan proses bagaimana peneliti mengimplementasikan dari rancangan serta analisis yang telah dilakukan pada pembahasan sebelumnya. Tahap implementasi meliputi implementasi kode program, implementasi antarmuka dan implementasi proses akusisi. Selain itu pada tahap ini juga membahas tentang pengujian aplikasi.

BAB V KESIMPULAN DAN SARAN

Pada bab ini berisikan kesimpulan tentang capaian yang telah peneliti capai dan juga berisikan saran untuk pengembangan aplikasi lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

(Olson & Shashidhar, 2016) dalam jurnalnya yang berjudul “*Low Budget Forensic Drive Imaging Using Arm Based Single Board Computers*” melakukan penelitian dengan membuat alat *forensic imaging* yang dapat berdiri sendiri, rendah bujet dan juga bersifat *portable* artinya mudah dibawa karena memanfaatkan *single board computer* yang memiliki ukuran kecil. Dalam penelitian ini, peneliti menggunakan dua buah *single board computer* yakni Raspberry Pi 2 dan Odroid XU4. Kemudian peneliti menggunakan tiga buah media penyimpanan, yaitu 1TB 2.5” USB 3.0 Western Digital, 2TB 2.5” USB 3.0 Seagate, dan 5 TB 3.5” USB 3.0 Seagate. Kemudian ketiga media penyimpanan tersebut diakusisi meggunakan Raspberry Pi 2 dan Odroid XU4 secara *bitstream image* dan menghasilkan *file image* dengan format .dd dan .e01. Selanjutnya, peneliti melakukan verifikasi terhadap *file image* yang telah berhasil diakusisi, *file image* .dd diverifikasi menggunakan *software* FTK Imager, sedangkan *file image* .e01 diverifikasi dengan *software* Encase Imager. Kemudian peneliti membandingkan kecepatan dalam mengakusisi media penyimpanan, maka didapatkan hasil bahwa Raspberry Pi 2 lebih lambat dibandingkan Odroid XU4 dikarenakan Raspberry Pi 2 masih menggunakan USB port 2.0 sebaliknya Odroid XU4 sudah menggunakan USB port 3.0. Berikut beberapa penelitian terdahulu lainnya dapat dilihat pada Tabel 2.1 Penelitian Terdahulu.

Tabel 2.1 Penelitian Terdahulu

No	Judul	Penulis	Pencapaian	Saran
1	Pemanfaatan <i>Raspberry Pi</i> untuk <i>Hacking</i> dan <i>Forensics</i> dengan Metode NIST (<i>National Institue of Standards and Technologi</i>)	Ilham Taufiqurohman, Nur Widiyasono, dan Husni Mubarak	Penelitian ini menghasilkan sebuah alat yang mampu menggambarkan bagaimana proses <i>hacking</i> sekaligus <i>forensics</i> bekerja. Alat ini mampu melakukan <i>cracking password</i> terhadap <i>access point</i> yang ditargetkan, kemudian	Masih ada beberapa kekurangan yang masih harus diperbaiki kedepannya, diantaranya: <ul style="list-style-type: none"> • Program yang di buat masih belum bisa dijalankan secara otomatis ketika alat

No	Judul	Penulis	Pencapaian	Saran
			<p>setelahnya mampu melakukan <i>sniffing</i> sehingga akhirnya mampu menghasilkan sebuah laporan forensik.</p>	<p>dinyalakan, dan diperlukan aplikasi untuk mengatur <i>autorunning</i>.</p> <ul style="list-style-type: none"> • Access point belum dapat dipilih sesuai keinginan, karena sistem yang dibuat memilih <i>access point</i> dilihat dari kekuatan sinyal <i>access point</i> yang ada pada <i>tools aircrack-ng</i>. • Waktu untuk memecahkan <i>password</i> belum cepat terhadap <i>password</i> yang rumit dan panjang, karena menggunakan teknik <i>brute-force attack</i>.
2	<p>Low Budget Forensic Drive Imaging Using Arm Based Single Board Computer</p>	<p>Eric Olson dan Narasimh Shashidhar</p>	<p>Ketika ingin melakukan analisa forensik terhadap <i>hardisk</i> ataupun media penyimpanan lain biasanya dalam keadaan mati. Untuk melakukan analisa tersebut dibutuhkan duplikator forensik tersendiri, perangkat keras <i>write-blocker</i> yang tertanam di laptop dan sistem operasi untuk forensik yang di <i>boot</i> dari USB ataupun <i>virtual machines</i>. Karena alat –</p>	<p>Saat mengembangkan alat ini peneliti masih menggunakan Raspberry Pi 2. Saran peneliti untuk menggunakan versi terbaru yaitu Raspberry Pi 3 agar kegiatan forensik yang dilakukan dapat berjalan lebih efektif.</p>

No	Judul	Penulis	Pencapaian	Saran
			alat untuk melakukan analisa forensik mahal, maka disini peneliti membuat alat yang mampu melakukan kegiatan forensik dengan harga yang murah. Solusi yang diberikan oleh peneliti menggunakan <i>single board computer</i> .	
3	Peran Komputer Forensik Dalam Penegakan Hukum Untuk Pembuktian Kasus Cyber Crime	Eko Suprpto	Kejahatan komputer (<i>cybercrime</i>) saat ini begitu marak tidak mengenal batas geografis, aktivitas ini dapat dilakukan dari jarak dekat ataupun jarak jauh. Pelaku <i>cybercrime</i> sangat pandai dalam menghilangkan jejak bukti digital yang ia tinggalkan, maka disini dibutuhkan komputer forensic guna mengungkap pelaku kejahatan siber. Menurut kemmish, terdapat empat elemen forensik yang menjadi kunci pengungkapan bukti digital yaitu: identifikasi bukti digital, penyimpanan bukti digital, analisa bukti digital dan presentasi bukti digital.	Disiapkan usaha preventif baik dari sisi teknisnya dan sisi hukumnya guna memberikan efek jera bagi pelaku kejahatan siber.
4	A Study of Forensic & Analysis Tools	Sriram Raghavan	Penelitian ini berisi studi tentang alat – alat yang digunakan dalam dunia	Kedepannya alat – alat harus terus di kembangkan karena

No	Judul	Penulis	Pencapaian	Saran
			forensik dan analisis forensik. Di dalam penelitian ini banyak alat – alat forensik dan analisis yang diuji untuk mengetahui sampai sejauh mana alat – alat tersebut mampu mendukung dunia forensik dan analisis sekarang.	data – data yang akan dianalisa akan semakin besar.
5	Demonstration of Programming in Python using a remote lab with Raspberry Pi	Helia Guerra, Alberto Cardoso, Vitor Sousa, Joaquim Leitao, Vitor Graveto dan Luis Mendes Gomes	Penelitian ini berisi tentang bagaimana membuat pembelajaran pemrograman menjadi lebih menarik sehingga tidak membosankan. Dalam penelitian ini siswa diberikan projek <i>Internet of Thing</i> (IOT). Siswa diminta membuat sebuah aplikasi yang menggunakan Bahasa Python untuk menampilkan data dari sensor - sensor yang sudah disiapkan sebelumnya. Selain itu diadakan forum agar siswa aktif bertanya sehingga siswa menjadi terbantu dalam menyelesaikan projeknya.	Programmer harus memiliki dasar fundamental pemrograman yang bagus agar lebih mudah memahami <i>programming</i> .
6	Antisipasi Cybercrime Menggunakan Teknik	Yudi Prayudi dan Dedy Setyo Afrianto	Berkembangnya teknologi informasi dan komputer dimanfaatkan betul oleh masyarakat luas, namun tak	Komputer forensik bidang ilmu yang dapat dijadikan dukungan dari aspek

No	Judul	Penulis	Pencapaian	Saran
	Komputer Forensik		<p>jarang disalah gunakan. Maka efek negatif timbul baik itu secara mikro yang dampaknya hanya pada tingkatan personel/perorangan, maupun secara makro yang berdampak pada wilayah komunal, publik serta memiliki efek domino yang luas. Untuk menyelesaikan kejahatan yang menggunakan komputer maka diperlukan komputer forensik guna mendapatkan jejak bukti digital yang ditinggalkan oleh pelaku.</p>	ilmiah serta menyediakan payung hokum yang jelas.
7	Python: A GUI Development Tool	Matthew J. Conway	<p>Python adalah Bahasa pemrograman yang berorientasi objek dan python termasuk <i>interpreted language</i>. Berikut beberapa keunggulan python:</p> <ul style="list-style-type: none"> • Mendukung level tipe data tinggi: strings, heterogeneous list, associative arrays, files, arbitrary precision numbers, classes, dan modules • Setiap tipe data mendukung banyak operator • Dapat membuat nama modul program tanpa bentrokan nama 	Semua objek dianggap string, Maka ketika ingin menganggap string sebagai numbers harus menggunakan fungsi <code>expr</code> .

No	Judul	Penulis	Pencapaian	Saran
			<p>Untuk membuat GUI di python dapat menggunakan modul Tkinter. Modul Tkinter menyediakan semua tipe widget yang dibutuhkan.</p>	
8	<p>Hacktivsm Trends, Digital Forensics Tools and Challenges: A Survey</p>	<p>Mohammad Wazid, Avita Katal, R H Goudar dan Sreenivas Rao</p>	<p>Penelitian ini berisi bagaimana trend aktivitas <i>hacking</i> yang terjadi dibarengi bagaimana perkembangan dari alat – alat <i>digital forensic</i>. <i>Hacking</i> adalah kejahatan terbesar dalam dunia siber, berikut trend aktivitas <i>hacking</i>:</p> <ul style="list-style-type: none"> • <i>Social Networking Websites</i> • <i>Cloud Computing</i> • <i>In day to day Web browsing</i> • <i>Phising</i> <p><i>Digital forensic</i> digunakan untuk membuktikan dan menemukan barang bukti yang ditinggalkan oleh <i>hacker</i>, berikut alat – alat yang biasa digunakan dalam digital forensic:</p> <ul style="list-style-type: none"> • FTK Imager • EnCase • Helix • Memoryze • WindowsScope • WinDump • Oxygen Forensic Suite • Arbutus 	<p>Selalu akan ada kompetisi pengembangan alat <i>digital forensic</i> dan anti <i>digital forensic</i>, maka dari itu seharusnya pengembangan alat <i>digital forensic</i> harusnya lebih di depan guna menangkal kejahatan siber.</p>

No	Judul	Penulis	Pencapaian	Saran
			<p>Masalah yang dihadapi ialah <i>hacker</i> berhasil membuat para investigator <i>forensic</i> kesulitan menemukan barang bukti sesuai dengan waktu yang di tentukan, sehingga membuat para pelaku bebas tanpa mendapatkan hukuman.</p>	
9	MyPyTutor: an Interactive Tutorial System For Python	Peter J. Robinson	<p>MyPyTutor adalah sebuah sistem tutorial interaktif yang diperuntukan bagi orang yang ingin mempelajari Bahasa pemrograman python. Di dalamnya terdapat banyak materi python mulai dari yang dasar hingga membuat GUI dengan Tkinter.</p>	<p>Masih ditemukan bug, maka pihak developer dalam hal ini peneliti masih terus menyempurnakan.</p>
10	The Forensic Image Generator Generator (Forensig)	Christian Moch dan Felix C. Freiling	<p>Penelitian ini menghasilkan alat <i>digital forensic</i> untuk edukasi. Peneliti mendesain dan membuat alat <i>digital forensic</i> yang dapat digunakan oleh instruktur untuk membuat <i>file system image</i> untuk edukasi analisis forensik.</p>	<p>Banyak aspek yang masih bisa dikembangkan di Forensig. Alat yang dibuat masih sulit di gunakan oleh instruktur yang tidak memiliki pemahaman dasar tentang pemrograman. Kemudian kedepannya Forensig dikembangkan dengan dibuatkan GUI nya.</p>

No	Judul	Penulis	Pencapaian	Saran
11	Forensic Data Recovery from Flash Memory	Marcel Breeuwsma, Martien de Jongh, Coert Klaver, Ronald van der Knijff dan Mark Roeloffs	<p>Pesatnya perkembangan telpon genggam semakin dirasakan masyarakat luas. Banyak manfaat positf yang didapat dari ponsel, namun bagi pelaku <i>cybercrime</i> ini menjadi celah untuk melancarkan aksi jahat mereka. Untuk mendapatkan barang bukti yang tertinggal di dalam <i>flash memory</i> ponsel dibutuhkan teknik khusus. Perlu masuk hingga <i>layer</i> terbawah agar semua data dapat disalin bahkan data yang sudah terhapus sekalipun.</p>	<p>Perlu terus dilakukan riset terutama dalam mekanisme membaca <i>flash</i> yang digunakan alat flasher agar dapat terus beradaptasi bagi generasi berikutnya.</p>
12	Rapid Application Development in Python	Krishna K. Agarwal, Achla Agarwal dan Catherine Hiller	<p>Disini peneliti melakukan eksperimen membuat aplikasi kalkulator berbasis <i>Grpahical User Interface</i> (GUI) dengan Bahasa pemrograman python. Peneliti memilih python karena python dapat berjalan di banyak sistem operasi. Selain itu peneliti memilih python karena ingin menguji apakah python bisa digunakan untuk <i>rapid application development</i> (RAD).</p>	<p>Peneliti menyarankan untuk lebih memahami python agar tidak hanya dari internet tetapi lebih melakukan riset melalui buku.</p>

No	Judul	Penulis	Pencapaian	Saran
			Data pada Tabel 2.1 Penelitian Terdahulu adalah referensi yang peneliti gunakan untuk membuat perangkat <i>forensic imaging</i> yang murah dan juga mudah dibawa. Untuk membuat perangkat <i>forensic imaging</i> yang murah dan mudah dibawa peneliti memanfaatkan <i>single board computer</i> Raspberry Pi 3 model B karena harganya yang terjangkau. Kemudian untuk mengoperasikan perangkat <i>forensic imaging</i> tersebut peneliti membuat aplikasi berbasis GUI dengan menggunakan Bahasa pemrograman <i>Python</i> . Peneliti menggunakan Bahasa pemrograman <i>Python</i> karena Bahasa pemrograman <i>Python</i> adalah Bahasa pemrograman yang <i>powerfull</i> , dapat berjalan di banyak <i>platform</i> dan juga mudah untuk dipelajari.	Setelah mempelajari penelitian terdahulu yang terdapat pada Tabel 2.1, maka selanjutnya peneliti berkeinginan membuat perangkat <i>forensic imaging</i> dengan bujet biaya rendah, dan juga mudah dibawa. Selanjutnya untuk mengoperasikannya peneliti membuat aplikasi berbasis GUI agar mudah dioperasikan.

2.2 Cybercrime

Cybercrime atau kejahatan siber adalah sebuah tindak pidana kejahatan yang lahir karena kemajuan teknologi informasi yang merugikan orang lain dengan memanfaatkan komputer sebagai media kejahatannya. Pelaku kejahatan siber dapat dituntut dengan undang-undang nomor 11 tahun 2008 tentang transaksi dan elektronik (UU ITE). *Cybercrime* adalah kejahatan yang dilakukan oleh pihak yang tidak bertanggung jawab dengan tujuan untuk merusak, memodifikasi dan mengeliminasi data (Taufiqurrohman et al., 2017). Sedangkan menurut Muhammad Nur Al-Azhar dalam bukunya yang berjudul *Digital Forensic, cybercrime* merupakan kejahatan yang menggunakan komputer sebagai alat utama untuk melakukan aksi kejahatannya, misalnya *defacement* (pengubahan halaman–halaman suatu situs secara ilegal), *denial distributed of service* (membuat suatu sistem tidak berjalan atau berfungsi sebagaimana mestinya setelah dibanjiri data oleh sekian banyak komputer yang telah terinfeksi dan menjadi *robot network*), *intrusion* (masuk secara ilegal ke dalam suatu sistem), dan lain–lain (Al-Azhar, 2012).

Dilansir dari media online *tekno.kompas.com*, menurut perusahaan keamanan Symantec dalam Internet Security Threat Report volume 17, Indonesia menempati peringkat 10 sebagai negara dengan aktivitas kejahatan siber terbanyak sepanjang tahun 2011. Indonesia menyumbang 2,4% kejahatan siber di dunia, angka ini naik 1,7% jika dibandingkan pada tahun 2010 lalu dimana Indonesia menempati peringkat 28 (Kompas, 2012). Kasus kejahatan siber di Indonesia sendiri terus mengalami tren kenaikan, di lansir dari *nasional.kompas.com* ada peningkatan jumlah perkara yang menyangkut kejahatan dunia maya atau *cybercrime*. Jika dibandingkan dengan tahun 2016, pada tahun 2017 mengalami kenaikan sebanyak tiga persen. Pada 2016 kejahatan siber yang ditangani Polri sebanyak 4.931 kasus, dan yang berhasil diselesaikan sebanyak 1.119 kasus, sedangkan pada tahun 2017 kejahatan siber yang ditangani Polri sebanyak 5.061 kasus, dan yang berhasil diselesaikan sebanyak 1.368 kasus (Movanita, 2017).

2.3 Komputer Forensik

Komputer berdasarkan Kamus Besar Bahasa Indonesia (KBBI) adalah alat elektronik otomatis yang dapat menghitung atau mengelolah data secara cermat menurut yang diinstruksikan, dan memberikan hasil pengolahan, serta dapat menjalankan sistem multimedia (film, music, televise, faksimile, dan sebagainya), biasanya terdiri atas unit pemasukan, unit pengeluaran, unit penyimpanan, serta unit pengontrolan. Sedangkan forensik menurut Kamus Besar Bahasa Indonesia (KBBI) adalah cabang ilmu kedokteran yang berhubungan dengan penerapan fakta-fakta medis pada masalah-masalah hukum. Komputer forensik adalah cabang ilmu forensik yang berkaitan dengan analisis dan pembuktian barang bukti kejahatan siber untuk pembuktian hukum sebagai barang bukti yang sah di mata hukum.

Menurut Muhammad Nur Al-Azhar dalam bukunya yang berjudul *Digital Forensic*, komputer forensik adalah ilmu forensik yang berkaitan dengan pemeriksaan dan analisis barang bukti elektronik berupa komputer pribadi (*personal computer-PC*), *laptop/notebook*, *netbook*, dan *tablet*. Pemeriksaan terhadap jenis barang bukti ini biasanya berkaitan dengan *file recovery*, yaitu suatu metode untuk mengambil *file logical* atau memunculkan kembali *file* yang sudah dihapus (*deleted*) maupun hilang (*lost*) dikarenakan tidak tercatat lagi di *file system* (Al-Azhar, 2012). Sedangkan menurut Ruby Alamsyah, komputer forensik adalah suatu ilmu yang menganalisis barang bukti secara digital hingga dapat dipertanggungjawabkan di pengadilan. Yang termasuk barang bukti digital tersebut antara lain: *laptop*, *handphone*,

notebook, dan alat teknologi lain yang memiliki tempat penyimpanan dan dapat dianalisa (Bobsusanto, 2014).

Saat ini kejahatan siber terus mengalami tren kenaikan, dikutip dari berita elektronik pada situs *nasional.kompas.com*, jumlah kejahatan siber pada tahun 2017 mengalami kenaikan sebesar tiga persen jika dibandingkan pada tahun 2016. Dengan adanya tren kenaikan kejahatan siber maka ilmu komputer forensik menjadi sangat dibutuhkan, karena barang bukti kejahatan siber adalah barang bukti yang rentan dan mudah rusak, sehingga dibutuhkan seseorang dengan keahlian khusus untuk mengungkap dan menganalisa bukti yang terdapat di komputer atau media penyimpanan digital. Di dalam UU ITE pasal 43 ayat 5 huruf (J) dikatakan bawah “Meminta bantuan ahli yang diperlukan dalam penyidikan terhadap tindak pidana di bidang Teknologi Informasi dan Transaksi Elektronik”.

2.3.1 Forensic Imaging

Forensic imaging adalah salah satu elemen dari ilmu komputer forensik yang berfungsi untuk melakukan akusisi terhadap media penyimpanan yang berstatus sebagai barang bukti dalam kejahatan siber. Barang bukti berupa media penyimpanan harus diduplikasikan guna penyidikan, agar barang bukti aslinya tidak mengalami perubahan sehingga integritas barang bukti tetap terjaga. Sesuai teori Locard Exchange Principle yang dikemukakan oleh Dr. Edmond Locard yang menyebutkan bahwa “*every contact leaves a trace*” artinya “setiap kontak meninggalkan jejak” (Baharuddin, Ruskam, & Yacob, 2015). Maksudnya adalah setiap kali seseorang melakukan tindak pidana kejahatan maka seseorang itu pasti akan meninggalkan jejak dalam bentuk apapun, maka tugas penyidik adalah menelusuri jejak itu sampai mendapatkan barang buktinya. Maka teori Locard Exchange Principle pun berlaku di dalam kasus kejahatan siber, karena di dalam kasus kejahatan siber pun pasti akan meninggalkan barang bukti yang mungkin saja tertinggal di dalam media penyimpanan, selanjutnya media penyimpanan tersebut harus di akusisi guna menjaga dan mengamankan barang bukti yang asli.

Forensic imaging bekerja dengan menyalin secara langsung dari perangkat penyimpanan fisik secara bit demi bit, sektor demi sektor termasuk semua *file*, *folder*, *unallocated*, *free* dan *slack space*. *Forensic imaging* tidak hanya menyalin *file-file* yang terlihat oleh mata saja, tetapi juga menyalin *file-file* yang telah terhapus yang tidak terlihat secara kasat mata yang tertinggal di *slack* dan *free space* dari media penyimpanan fisik. Dengan diakusisinya barang bukti maka seorang *investigator* mendapatkan keseluruhan *file* yang terdapat di dalam media

penyimpanan, baik *file* yang memang ada di dalam media penyimpanan tersebut ataupun *file* yang sudah dihapus sebelumnya.

2.4 File Ekstensi Hasil Akusisi

Menurut Simson L. Garfinkel dalam jurnalnya yang berjudul *Digital Forensic Research: The Next 10 Years* mengatakan bahwa saat ini *forensic imaging* adalah alat yang sangat penting guna mengatasi kejahatan yang menggunakan atau memanfaatkan komputer sebagai media kejahatannya atau biasa disebut dengan *cybercrime*, *forensic imaging* dianggap penting karena *forensic imaging* memiliki kemampuan untuk membuktikan barang bukti kejahatan siber yang mungkin tertinggal pada komputer yang digunakan oleh pelaku kejahatan siber (Garfinkel, 2010). Untuk mengamankan barang bukti yang mungkin tertinggal pada komputer, dapat dilakukan akusisi terhadap penyimpanan yang terdapat pada komputer tersebut agar barang bukti yang asli tidak mengalami perubahan bit data sehingga integritas barang bukti asli tetap terjaga.

Ketika melakukan akusisi terhadap media penyimpanan maka dapat menggunakan banyak *file* format ekstensi yang bisa digunakan, ada banyak *file* ekstensi yang biasa digunakan di dalam dunia *forensic imaging*, baik itu *file* ekstensi dd, E01 maupun AFF. *File* format dengan ekstensi dd adalah *file* format ekstensi tertua karena ekstensi dd lahir sebelum ekstensi E01 maupun AFF dan ekstensi dd adalah *file* format yang termasuk ke dalam *file* tipe RAW. *File* dengan ekstensi dd bekerja dengan cara menyalin data media penyimpanan secara keseluruhan dengan cara menyalin setiap sektor dari media sumber penyimpanannya tanpa ada satu pun sektor yang terlewatkan. Namun *file* format dengan ekstensi dd memiliki kekurangan yaitu *file* hasil akusisi tidak mengandung *file* metadata, melainkan hanya berisi data orisinil saja yang didapatkan dari sumber penyimpanan yang diakusisi. Untuk informasi terkait metadata biasanya ditempatkan di dalam *file* yang berbeda. Menurut Simson L. Garfinkel dalam jurnalnya yang berjudul *Advanced Forensic Format: An Open, Extensible Format For Disk Imaging* mengatakan bahwa mengakusisi media penyimpanan dengan cara menyalin sektor demi sektor dari penyimpanan orisinil dan menyimpannya di dalam satu *file* kemudian menaruh hasil *file* akusisi di media penyimpanan lainnya dan *file* hasil akusisi tersebut memiliki ukuran yang sama persis dengan penyimpanan orisinilnya itu disebut raw atau dd. *File* dengan ekstensi dd adalah *file* yang tidak di kompresi sebelumnya, sehingga berimbas pada hasil *file* akusisinya yang memiliki ukuran yang sangat besar walaupun sebenarnya data yang diakusisi sangat kecil (Garfinkel, Malan, Dubec, Stevens, & Pham, 2006).

Sedangkan *file* format dengan ekstensi E01 adalah format ekstensi yang dimiliki oleh Guidance Software dan ekstensi E01 masuk ke dalam *file* dengan tipe Expert Witness Format (EWF). Cara kerja *file* dengan ekstensi E01 hampir sama dengan *file* ber ekstensi dd, *file* dengan ekstensi E01 bekerja dengan cara menyalin bit demi bit data dari media penyimpanan yang dijadikan sebagai sumber akusisi tanpa ada satu pun bit data yang terlewatkan. Yang membedakan antara ekstensi dd dan E01 adalah, *file* format E01 akan membagi *file* hasil akusisi ke dalam beberapa segmen dan menyisipkan metadata berupa informasi penting baik terkait sumber penyimpanan yang diakusisi, maupun informasi tentang *file* hasil akusisi seperti *hash* MD5 maupun penguji yang melakukan akusisi. File dengan ekstensi E01 akan memiliki *header* pada file nya yang berisi tanggal dan waktu akusisi, nama penguji dan catatan untuk *file* akusisi kemudian diikuti oleh *footer* yang berisikan informasi tentang *hashing* MD5 (Garfinkel et al., 2006).

Selanjutnya *file* dengan format ekstensi Advanced Forensic Format (AFF) adalah format ekstensi yang dikenalkan dan dikembangkan oleh Simson L. Garfinkel. Advanced Forensic Format (AFF) diciptakan untuk memberikan pilihan alternatif bagi seorang investigator forensik di dalam memilih format ekstensi *file* hasil akusisinya. Simson L. Garfinkel dalam jurnalnya yang berjudul *Advanced Forensic Format: An Open, Extensible Format For Disk Imaging* mengatakan bahwa *file* format dengan ekstensi AFF menawarkan dua keuntungan yang dapat diperoleh, pertama *file* hasil akusisi dengan ekstensi AFF lebih fleksibel karena dapat menyimpan metadata yang lebih besar dan kedua ekstensi AFF memiliki ukuran *file* yang lebih kecil jika dibandingkan dengan *file image* lainnya karena *file* dengan ekstensi AFF akan di kompres sehingga memiliki ukuran yang lebih kecil karena *file* format ekstensi lainnya tidak mengalami proses kompresi (Garfinkel et al., 2006).

2.5 Python

Bahasa pemrograman *Python* adalah bahasa yang kaya dengan modul baik itu modul yang dikembangkan oleh *Python* sendiri maupun yang dikembangkan oleh pihak ketiga dan bahasa pemrograman *Python* dapat berjalan di banyak *platform* sehingga bersahabat bagi banyak orang. *Python* pertama kali dikembangkan oleh Guido van Rossum pada tahun 1990 di Stichting Mathematisch Centrum (CWI), Amsterdam sebagai kelanjutan dari Bahasa pemrograman ABC. Versi pertama yang dikeluarkan adalah Python 1.0 yang dirilis pada Januari 1994 dan versi terakhir sampai saat ini adalah Python 3.6 yang dirilis pada 23 Desember 2016.

Matthew J. Conway dalam jurnalnya yang berjudul *Python: A GUI Development Tool* mengatakan bahwa Bahasa pemrograman *Python* adalah Bahasa yang dikembangkan oleh Guido van Rossum seorang peneliti CWI di Amsterdam. Matthew mengatakan bahwa Bahasa pemrograman *Python* adalah Bahasa pemrograman yang berorientasi objek (Conway, 1995). Sedangkan menurut Guido van Rossum sebagai seseorang yang mengembangkan *Python* dalam bukunya yang berjudul *Python Tutorial* mengatakan bahwa Bahasa pemrograman *Python* adalah Bahasa yang mudah dipelajari dan mudah dipahami. Bahasa pemrograman *Python* memiliki struktur data yang efisien di level yang tinggi dan juga simpel tetapi efektif untuk pendekatan pemrograman yang berorientasi objek sehingga *Python* cocok digunakan untuk membuat aplikasi karena *Python* juga dapat berjalan di banyak *platform* (van Rossum, 1995).

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian adalah prinsip dasar analisis proses ilmiah untuk mendapatkan data yang akan digunakan dalam kepentingan penelitian. Di dalam memecahkan masalahnya peneliti dapat menggunakan berbagai metode penelitian yang ada guna membantu peneliti untuk memecahkan masalah. Proses pengumpulan data menjadi sesuatu yang sangat penting di dalam proses penelitian, karena data yang terkumpul akan menjadi landasan teori suatu penelitian dan juga mampu meperkuat argumentasi proses penelitian itu sendiri. Di dalam pembuatan aplikasi *Forensic Imaging Application Using Raspberry Pi* peneliti menggunakan teknik studi literatur atau biasa disebut dengan studi pustaka sebagai metode pengumpulan datanya.

Studi pustaka adalah mengumpulkan informasi dan data dengan bantuan berbagai macam material yang ada di perpustakaan seperti dokumen, buku, catatan, majalah, kisah-kisah sejarah dsb (Mardalis, 2002). Definisi lainnya, studi pustaka adalah mempelajari berbagai buku referensi serta hasil penelitian sebelumnya yang sejenis yang berguna untuk mendapatkan landasan teori mengenai masalah yang akan diteliti (Sarwono, 2006). Dalam proses pembuatan aplikasi *Forensic Imaging Application Using Raspberry Pi* peneliti melakukan studi literatur atau studi pustaka dengan cara mencari jurnal ilmiah, artikel ilmiah, makalah ilmiah, dan tugas akhir baik di perpustakaan maupun di internet.

Selain itu peneliti juga mempelajari penelitian terdahulu guna memberikan gambaran dan pengetahuan tentang proses pembuatan aplikasi *Forensic Imaging Application Using Raspberry Pi*. Selanjutnya data-data yang telah dipelajari oleh peneliti akan diolah dan dipilah sesuai dengan kebutuhan yang dibutuhkan. Adapun data-data yang dibutuhkan untuk membuat aplikasi *Forensic Imaging Application Using Raspberry Pi* diantara lain adalah teori tentang bagaimana proses dan cara kerja *forensic imaging*, teori terkait cara kerja, instalasi dan penggunaan Raspberry Pi 3, teori tentang Bahasa pemrograman *Python*, bagaimana membuat tampilan GUI di *Python*, kemudian mencari tahu modul apa saja yang dibutuhkan untuk membuat format *file .dd*, dan untuk proses *hashing* MD5 dan SHA512. Setelah data diolah dan dipelajari oleh peneliti, selanjutnya peneliti mengimplementasikan data-data yang diperoleh dalam proses pembuatan aplikasi *Forensic Imaging Application Using Raspberry Pi*.

3.2 Analisis Kebutuhan Sistem

Untuk membangun aplikasi *Forensic Imaging Application Using Raspberry Pi* ini dibutuhkan proses analisis kebutuhan sistem agar mengetahui secara detail apa-apa saja kebutuhan yang dibutuhkan guna membangun sistem ini. Analisis yang dilakukan peneliti diantaranya analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak, analisis kebutuhan masukan, analisis kebutuhan proses, dan analisis kebutuhan keluaran. Tujuan dilakukannya analisis ini adalah untuk mengetahui apa-apa saja yang dibutuhkan sehingga dapat menunjang di dalam proses pembuatan aplikasi *Forensic Imaging Application Using Raspberry Pi* ini.

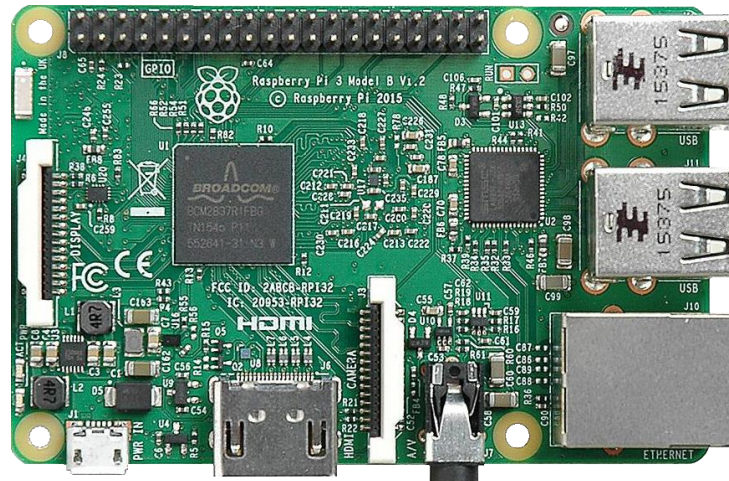
3.2.1 Analisis Kebutuhan Perangkat Keras

Berikut beberapa kebutuhan perangkat keras yang diperlukan guna menunjang di dalam proses pembuatan aplikasi *Forensic Imaging Application Using Raspberry Pi* diantaranya adalah:

a. *Single Board Computer*

Single board computer yang peneliti gunakan adalah Raspberry Pi 3 model B. Saat ini ada banyak jenis atau model Raspberry Pi yang beredar luas di pasaran. Raspberyy Pi sendiri telah mengeluarkan berbagai jenis model diantaranya adalah Raspberry Pi 1 (Model A, Model B dan Model B+), Raspberry Pi 2, Raspberry Pi Zero, Raspberry Pi Zero W, Raspberry Pi Zero WH, Raspberry Pi 3 Model B, dan Raspberry Pi 3 Model 3B+. Peneliti menggunakan Raspberry Pi 3 Model B dan berikut adalah spesifikasi Raspberry Pi 3 yang peneliti gunakan:

- Soc: Broadcom BCM2837
- CPU: 4X ARM Cortex-A53, 1.2GHz
- GPU: Broadcom VideoCore IV
- RAM: 1GB LPDDR2 (900 MHz)
- Networking: 10/100 Ethernet, 2.4 GHz 802.11 n wireless
- Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
- Storage: microSD
- GPIO: 40-pin header, populated
- Ports: HDMI, 3.5 mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI).



Gambar 3.1 Raspberry Pi 3 Model B

Gambar 3.1 adalah komputer berukuran kartu kredit yang dikembangkan di Inggris oleh yayasan Raspberry Pi, kemudian Raspberry Pi dilengkapi dengan *port USB*, *LAN port*, *audio jack*, *HDMI port*, *Micro SD card*, serta *pin-pin* untuk menghubungkan dengan komponen lain seperti *button*, dan *LED* (Taufiqurrohman et al., 2017). Sedangkan menurut Carrie Anne Philbin dalam bukunya yang berjudul *Adventures in Raspberry Pi* mengatakan, Raspberry Pi adalah komputer dengan ukuran yang sangat kecil, hanya sebesar kartu kredit, dan Raspberry Pi dapat dibangun dan dikontrol sesuai kebutuhan pengguna (Philbin, 2014).

Tren penggunaan Raspberry Pi terus mengalami peningkatan sejak dirilis pertama kali pada 29 Februari 2012, bukan tidak mungkin kedepannya Raspberry Pi akan benar-benar mampu menggantikan fungsi PC Desktop yang ada sekarang, karena memang saat ini Raspberry Pi yang merupakan *single board computer* ini memiliki keunggulan dengan ukurannya yang kecil, selain itu *single board computer* ini mampu melakukan hal-hal yang bisa dilakukan oleh PC Desktop diantaranya, menulis dokumen, membuat program, berselancar di dunia maya atau bahkan menonton video dengan kualitas tinggi. Dilansir dari situs *premierfarnell.com*, saat ini industri dan pengembang aplikasi profesional terus menunjukkan ketertarikannya di dalam mengembangkan aplikasi yang mereka buat dengan memanfaatkan *single board computer* Raspberry Pi (Premierfarnell, 2018).

b. Power Adapter

Power adapter seperti yang terlihat pada Gambar 3.2 berfungsi untuk memberikan daya kepada perangkat keras Raspberry Pi sehingga dapat berjalan. Idealnya perangkat

keras Raspberry Pi diberi daya sebesar 5V 2,5A, namun di sini peneliti menggunakan power adapter yang biasa digunakan untuk telpon genggam. Power adapter yang digunakan peneliti adalah micro USB dan memiliki daya keluaran sebesar 5V 850mA.



Gambar 3.2 Power Adapter

c. Keyboard External dan Mouse

Berikut spesifikasi *keyboard* dan *mouse* yang digunakan oleh peneliti guna menunjang proses pembuatan aplikasi *forensic imaging* ini:

- Dimensi *keyboard*: Tinggi (139mm) x Lebar (288mm) x Tebal (21mm)
- Berat *keyboard*: 312 gr
- Dimensi *mouse*: Tinggi (60mm) x Lebar (100mm) x Tebal (32mm)
- Berat *mouse*: 53 gr
- Konektivitas: *wireless* 2,4GHz (10 meter)
- *Receiver*: nano USB



Gambar 3.3 Keyboard dan Mouse

Untuk memberikan masukan terhadap perangkat keras Raspberry Pi dibutuhkan *keyboard external*, *keyboard* ini berfungsi sebagai data masukan yang diberikan oleh pengguna perangkat *forensic imaging* untuk perangkat keras Raspberry Pi. Kemudian untuk memudahkan di dalam pengoperasian perangkat keras Raspberry Pi dibutuhkan *mouse* seperti yang terlihat pada Gambar 3.3.

d. Layar LCD 4inch

Seperti halnya PC Desktop, perangkat keras Raspberry Pi juga membutuhkan sebuah LCD *monitor* sebagai media keluarannya ketika pengguna perangkat keras Raspberry Pi melakukan masukan atau *input-an*, maka CPU dari perangkat keras akan mengelola masukan tersebut yang selanjutnya hasil *input-an* tersebut akan di tampilkan oleh LCD *monitor* dalam bentuk grafis atau gambar.



Gambar 3.4 LCD Monitor

LCD *monitor* yang digunakan memiliki ukuran 4 *inch* dengan resolusi sebesar 800 x 400 *high resolution* seperti yang terlihat pada Gambar 3.4. LCD keluaran WaveShare ini sudah mendukung layar sentuh, selain itu *monitor* ini juga mendukung HDMI sebagai keluarannya sehingga mampu menampilkan gambar secara maksimal.

e. Micro SD dan Flash drive

Layaknya PC Desktop ataupun laptop pada umumnya yang membutuhkan *hardisk* sebagai media penyimpanannya, *single board computer* Raspberry Pi pun membutuhkan sebuah media penyimpanan yang berfungsi sebagai tempat untuk menyimpan sistem operasi dan data lainnya. Untuk menyimpan sistem operasi dan data lainnya, perangkat

keras Raspberry Pi membutuhkan *micro SD* sebagai media penyimpanannya seperti yang terlihat pada Gambar 3.5. Berikut spesifikasi *micro SD* yang peneliti gunakan:

- Kapasitas: 16 GB
- Kecepatan Baca: hingga 80MB/s
- Faktor Bentuk: microSDHC
- Kecepatan Video: C10



Gambar 3.5 *Micro SD*

Selanjutnya peneliti menggunakan tujuh buah *flash drive* untuk digunakan di dalam proses uji coba akusisi aplikasi *forensic imaging*. Enam buah *flash drive* digunakan untuk diduplikasi atau diakusisi, sedangkan satu buah *flash drive* yang tersisa digunakan sebagai tempat untuk menyimpan *file* hasil akusisi. *Flash drive* yang digunakan untuk proses uji coba akusisi aplikasi diantaranya adalah Kingston 4GB USB 2.0, ICRC 4GB USB 2.0, Sandisk 8GB USB 2.0, Toshiba 8GB USB 2.0, Sandisk 16GB USB 2.0, dan Toshiba 16GB USB 2.0. Selanjutnya *flash drive* yang digunakan untuk menampung *file* hasil akusisi aplikasi adalah *flash drive* Sandisk 64GB USB 2.0.

3.2.2 Analisis Kebutuhan Perangkat Lunak

Di dalam proses perancangan serta pembuatan aplikasi *Forensic Imaging Application Using Raspberry Pi* ini membutuhkan beberapa komponen perangkat lunak atau istilah lainnya adalah *software* yang mampu menunjang selama proses pembuatan aplikasi *forensic imaging* ini. Berikut ini adalah beberapa komponen perangkat lunak yang peneliti gunakan selama proses pembuatan aplikasi *forensic imaging* ini:

a. Sistem operasi

Menurut Weikh M.H dalam bukunya yang berjudul *Computer Science and Communication Dictionary* mengatakan bahwa sistem operasi dapat menangani berbagai

macam pemrosesan data, komunikasi, komputer, informasi, atau aplikasi sistem kontrol (Weik, 2001). Sistem operasi adalah jembatan komunikasi antara pengguna perangkat komputer dan perangkat komputer itu sendiri, sehingga setiap masukan atau *input* yang diberikan oleh pengguna dapat dipahami dengan baik oleh sebuah perangkat komputer. Selain itu sistem operasi juga mengelola seluruh sumber daya yang terdapat di dalam perangkat komputer, sehingga tanpa adanya sistem operasi sebuah perangkat komputer tidak dapat berjalan dengan baik dan benar.



Gambar 3.6 Raspbian

Sistem operasi yang peneliti gunakan untuk perangkat keras Raspberry Pi 3 adalah sistem operasi Raspbian seperti yang terlihat pada Gambar 3.6. Raspbian merupakan kependekan dari Raspberry Debian, yaitu sebuah sistem operasi gratis berbasis Debian yang memang di optimalkan untuk perangkat keras Raspberry Pi yang berjalan di atas *kernel* linux sehingga mampu membuat perangkat keras Raspberry Pi dapat berjalan secara optimal.

b. Python Shell

Di dalam proses pembuatan aplikasi *Forensic Imaging Application Using Raspberry Pi* peneliti menggunakan *text editor* yang memang sudah tersedia di dalam sistem operasi Raspbian yang peneliti gunakan untuk *single board computer* Raspberry Pi. *Text editor* yang peneliti gunakan adalah Python Shell. Python Shell adalah tempat peneliti menuangkan seluruh kode program yang peneliti buat untuk digunakan di dalam aplikasi *Forensic Imaging Application Using Raspberry Pi*. Python Shell yang peneliti gunakan adalah Python Shell versi 3.5.3.

c. Bahasa pemrograman Python

Dilansir dari situs *id.wikipedia.org* Python adalah Bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai Bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar. Python pertama kali dikembangkan oleh Guido van Rossum pada tahun 1990 di Stichting Mathematisch Centrum (CWI), Amsterdam sebagai kelanjutan dari Bahasa pemrograman ABC. Versi pertama yang dikeluarkan adalah Python 1.0 yang dirilis pada Januari 1994 dan versi terakhir sampai saat ini adalah Python 3.6 yang dirilis pada 23 Desember 2016.

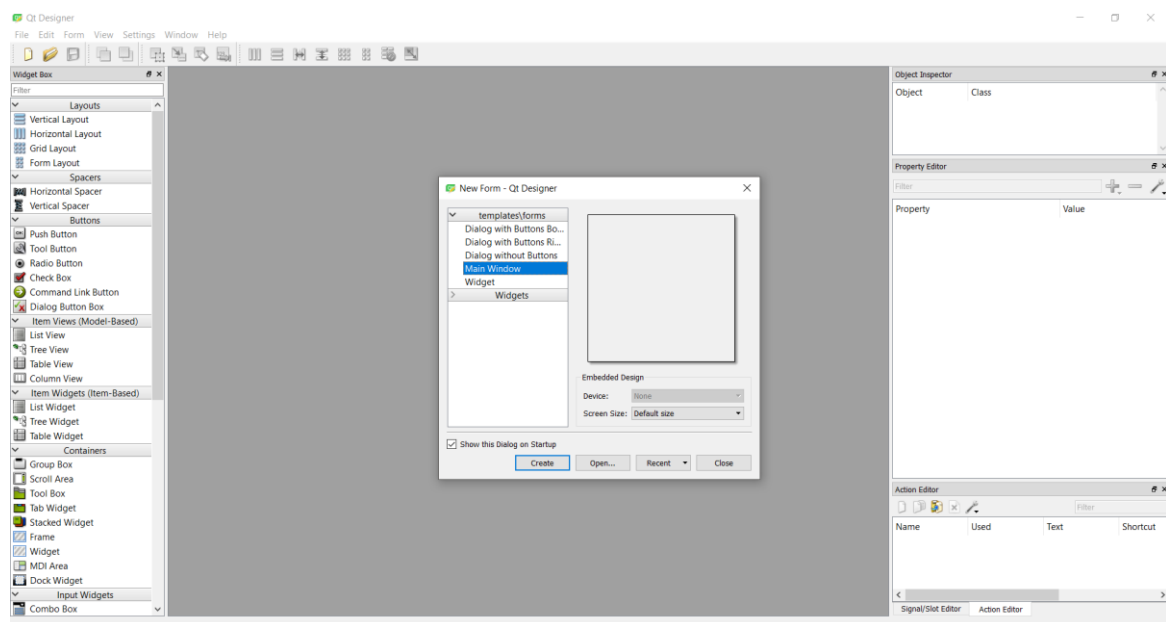
Peneliti memilih Python sebagai Bahasa pemrograman yang digunakan untuk membuat aplikasi *Forensic Imaging Application Using Raspberry Pi* karena Python adalah Bahasa pemrograman yang cukup mudah untuk dipelajari, disisi lain Python adalah Bahasa pemrograman yang cocok untuk dipakai di perangkat keras Raspberry Pi dan 3.5.3 adalah versi Python yang peneliti gunakan. Dilansir dari situs *python.org*, Python Software Foundation mengklaim bahwa Bahasa pemrograman Python adalah Bahasa pemrograman yang *powerfull* dan cepat, selain itu Python dapat berjalan di banyak *platform* sistem operasi sehingga membuat Python bersahabat bagi banyak orang, dan Python adalah Bahasa pemrograman gratis sehingga semua orang dapat menggunakannya.

Python adalah Bahasa pemrograman yang kaya akan modul atau *library* baik itu *library* yang dikembangkan sendiri oleh Python ataupun *library* yang dikembangkan oleh pihak ketiga. Berikut ini beberapa modul atau *library* yang peneliti gunakan guna menunjang di dalam proses pembuatan aplikasi *Forensic Imaging Application Using Raspberry Pi*:

- PyQt5
- sys
- hashlib
- subprocess
- logging
- time
- datetime
- os

d. QT Designer

Salah satu tahapan proses pembuatan aplikasi *Forensic Imaging Application Using Raspberry Pi* adalah membuat rancangan desain antarmuka aplikasi. Untuk membuat rancangan desain antarmuka aplikasi peneliti menggunakan perangkat lunak QT Designer seperti yang terlihat pada Gambar 3.7. QT Designer adalah sebuah perangkat lunak yang dapat digunakan untuk membuat desain dan membangun tampilan aplikasi berbasis *Graphical User Interface* (GUI). Tahapan proses merancang desain antarmuka aplikasi *Forensic Imaging Application Using Raspberry Pi* adalah sebuah tahapan proses yang sangat penting karena menyangkut langsung dengan tampilan aplikasi yang peneliti buat. Harapannya dengan membuat rancangan desain antarmuka yang baik dan menarik, maka pengguna aplikasi merasa nyaman di dalam menggunakan aplikasi yang peneliti buat dan juga tidak mengalami kesulitan di dalam mengoperasikannya.



Gambar 3.7 Qt Designer

Alasan peneliti menggunakan QT Designer untuk membuat rancangan desain antarmuka aplikasi adalah karena peneliti menggunakan modul PyQt5 pada aplikasi *Forensic Imaging Application Using Raspberry Pi* sebagai tampilan GUI nya. Dengan membuat rancangan desain antarmuka aplikasi menggunakan QT Designer, diharapkan mampu mempermudah peneliti di dalam tahap proses implementasi desain antarmuka aplikasi, karena baik perangkat lunak Qt Designer maupun modul PyQt5 merupakan

produk keluaran Qt Company. Sedangkan Qt Designer yang peneliti gunakan adalah Qt Designer dengan versi 5.9.1.

3.2.3 Analisis Kebutuhan Masukan

Analisis kebutuhan masukan adalah analisis kebutuhan yang dilakukan oleh peneliti dengan tujuan untuk mengetahui data masukan apa saja yang dibutuhkan aplikasi *Forensic Imaging Application Using Raspberry Pi* sebelum proses akusisi dijalankan. Analisis kebutuhan masukan dilakukan dengan mempertimbangkan segala kemungkinan data masukan yang dibutuhkan oleh aplikasi, sehingga harapannya dengan data masukan yang dibutuhkan oleh aplikasi terpenuhi maka aplikasi *Forensic Imaging Application Using Raspberry Pi* dapat berjalan dengan baik sesuai dengan apa yang diinginkan oleh peneliti.

Kebutuhan masukan pertama yang harus dimasukkan oleh pengguna adalah pemilihan media penyimpanan yang ingin diakusisi, disini pengguna diminta untuk memilih media penyimpanan mana yang ingin pengguna akusisi. Setelah menentukan media penyimpanan mana yang ingin diakusisi, selanjutnya pengguna diminta untuk memasukkan beberapa data masukan melalui kolom yang tersedia di aplikasi, diantaranya adalah kolom *image directory* yang berfungsi untuk menentukan letak lokasi penyimpanan hasil akusisi, kemudian kolom *filename* yang berfungsi untuk penamaan *file* hasil akusisi. Selanjutnya adalah kolom *notes* untuk memberikan beberapa catatan terhadap *file* hasil akusisi, kemudian kolom *examiner* yang berguna untuk mengetahui siapa yang melakukan akusisi, dan terakhir adalah kolom *image type* berfungsi untuk menentukan tipe *file* hasil akusisi.

3.2.4 Analisis Kebutuhan Proses

Analisis kebutuhan proses dilakukan oleh peneliti untuk mengetahui proses apa saja yang dibutuhkan oleh aplikasi *Forensic Imaging Application Using Raspberry Pi*. Dengan dilakukannya analisis kebutuhan proses ini, peneliti dapat mengetahui bagaimana tahapan-tahapan yang tepat untuk melakukan akusisi terhadap sebuah media penyimpanan. Sehingga nanti ketika masuk ke dalam tahap implementasi harapannya peneliti sudah mempunyai gambaran yang jelas mengenai proses apa saja yang diperlukan agar aplikasi *Forensic Imaging Application Using Raspberry Pi* dapat berjalan sesuai dengan yang diinginkan oleh peneliti.

Kebutuhan proses pertama yang dilakukan adalah aplikasi melakukan *scanning* untuk mendeteksi berapa banyak media penyimpanan yang tersedia. Setelah aplikasi selesai melakukan proses *scanning*, maka selanjutnya data hasil dari *scanning* di tampilkan ke dalam

tabel yang memuat informasi semua media penyimpanan yang tersedia yang dapat dipilih oleh pengguna untuk dilakukan proses akusisi. Selanjutnya pengguna harus menentukan media penyimpanan mana yang ingin diakusisi, kemudian pengguna diminta untuk mengisi data masukan melalui kolom-kolom yang tersedia.

Setelah pengguna melengkapi data masukan melalui kolom-kolom yang tersedia, proses selanjutnya yang berjalan adalah proses akusisi. Proses akusisi bekerja dengan cara menyalin data dari media penyimpanan secara *bitstream image*, yaitu mengambil bit demi bit data dari media penyimpanan fisik. *Bitstream* adalah metode penyimpanan digital dengan mengkopi setiap bit demi bit dari data orisinal, termasuk *file* yang tersembunyi (*hidden file*), *file* temporer (*temp file*), *file* yang terdefragmen (*fragmen file*), dan *file* yang belum ter-overwrite (Prayudi & Afrianto, 2007).

Setelah proses akusisi selesai, maka proses selanjutnya yang dilakukan aplikasi adalah proses *hashing*, baik MD5 maupun SHA512. Proses *hashing* adalah proses yang dilakukan untuk menguji integritas *file* akusisi hasil dari proses akusisi media penyimpanan. Karena di dalam proses akusisi media penyimpanan tidak boleh ada satu bit pun yang berubah, dengan merubah satu bit saja maka *file* hasil dari akusisi tidak dapat digunakan secara sah di mata hukum. Dengan bantuan MD5 dan SHA512 *file* hasil dari akusisi dapat dinilai integritasnya, apakah integritasnya masih terjaga dengan baik atau justru sebaliknya mengalami perubahan bit yang membuat integritasnya tidak terjaga dengan baik. Proses selanjutnya yang berjalan adalah proses membuat *file logging*, yaitu sebuah proses untuk membuat dokumen yang berisikan rincian informasi detail hasil *file* akusisi. Di dalam *file logging* terdapat beberapa informasi penting, diantaranya adalah informasi waktu yang menunjukkan pukul berapa proses akusisi dimulai dan pada pukul berapa proses akusisi selesai. Selain itu hasil dari proses *hashing* akan dimasukkan ke dalam *file logging* sebagai jaminan integritas dari pada hasil *file* akusisi.

3.2.5 Analisis Kebutuhan Keluaran

Analisis kebutuhan keluaran adalah analisis yang dilakukan untuk mengidentifikasi keluaran apa saja yang harus dihasilkan oleh aplikasi *Forensic Imaging Application Using Raspberry Pi* setelah serangkaian proses yang telah dilalui sebelumnya. Berikut keluaran yang dihasilkan oleh aplikasi setelah melalui serangkaian proses:

- a. *File* format .dd

Setelah aplikasi menyelesaikan proses akusisinya, maka aplikasi akan menghasilkan sebuah *file* hasil akusisi terhadap media penyimpanan dengan format .dd. *File* dengan format .dd adalah salah satu format yang sering digunakan di dalam lingkungan pengakusisian *forensic imaging*, dan *file* format .dd termasuk ke dalam jenis *file* tipe RAW.

b. File logging

Selain menghasilkan *file* akusisi dengan format .dd, aplikasi *Forensic Imaging Application Using Raspberry Pi* juga menghasilkan keluaran berupa *file logging* yang berisikan informasi detail mengenai data-data terkait media penyimpanan yang diakusisi oleh aplikasi. Keberadaan *file logging* menjadi sangat penting, karena *file logging* juga berguna sebagai dokumentasi dari hasil proses akusisi media penyimpanan.

c. Hash MD5 dan SHA512

Di dalam dunia forensik integritas adalah segalanya, maka ketika proses akusisi berjalan peneliti harus memastikan bahwa aplikasi *Forensic Imaging Application Using Raspberry Pi* tidak merubah nilai-nilai bit yang terdapat di media penyimpanan yang diakusisi. Untuk memastikan bahwa tidak ada bit-bit yang berubah, peneliti menggunakan MD5 dan SHA512 sebagai acuan integritas *file* hasil akusisi.

3.3 Perancangan Aplikasi

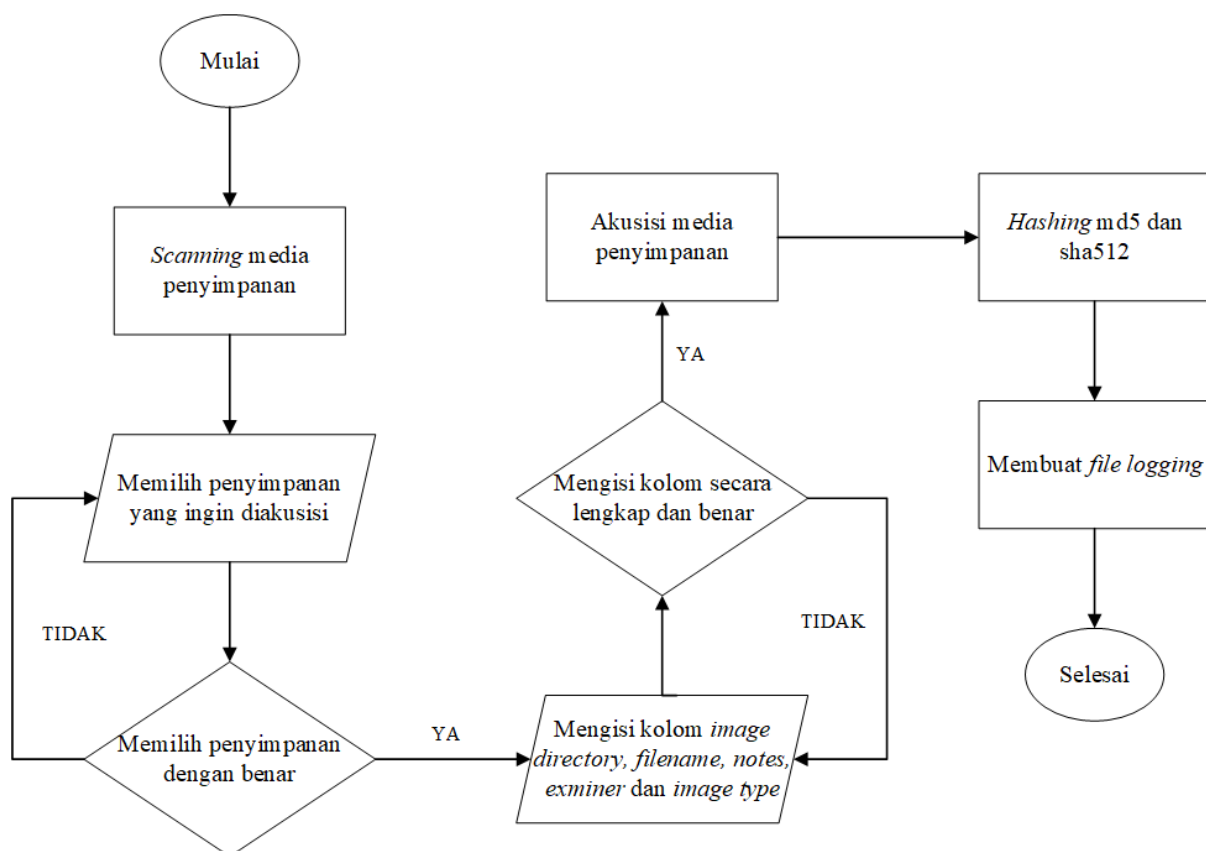
Perancangan menurut kamus besar Bahasa Indonesia (KBBI) adalah proses, cara, perbuatan merancang. Maka perancangan aplikasi adalah suatu proses yang dilakukan untuk membuat konsep dan mendesain alur kerja sistem guna memenuhi kebutuhan aplikasi *Forensic Imaging Application Using Raspberry Pi* sehingga aplikasi dapat bekerja dengan baik sesuai dengan apa yang diharapkan oleh peneliti. Proses perancangan aplikasi meliputi tiga aspek rancangan yaitu, *flowchart*, *pseudocode*, dan desain antarmuka.

3.3.1 Flowchart

Flowchart adalah diagram alir yang menguraikan aktivitas sistem sehingga dapat memberikan gambaran secara luas tentang bagaimana alur sistem ataupun cara kerja aplikasi, dalam hal ini adalah aplikasi *Forensic Imaging Application Using Raspberry Pi*. Selain itu *flowchart* juga berfungsi sebagai dokumentasi dari sebuah program karena mungkin saja

kedepannya ada pihak yang ingin mengembangkan program yang peneliti buat sehingga pihak tersebut memiliki pedoman yang jelas tentang alur kerja aplikasi.

Diagram alur atau *flowchart* aplikasi *Forensic Imaging Application Using Raspberry Pi* dapat dilihat pada Gambar 3.8. Ketika pertama kali dijalankan oleh pengguna, maka hal yang dilakukan pertama kali oleh aplikasi adalah melakukan pemindaian terhadap media penyimpanan. Pemindaian dilakukan untuk mengetahui media penyimpanan mana saja yang terikat dengan perangkat keras Raspberry Pi. Setelah pemindaian selesai dilakukan, kemudian data hasil pindaian akan ditampilkan ke dalam bentuk tabel untuk memudahkan pengguna dalam memilih media penyimpanan mana yang ingin diakusisi.



Gambar 3.8 *Flowchart Forensic Imaging Application Using Raspberry Pi*

Setelah data hasil pindaian ditampilkan ke dalam bentuk tabel, tahap selanjutnya adalah pengguna aplikasi harus menentukan media penyimpanan mana yang ingin diakusisi mengacu kepada media penyimpanan yang tersedia di dalam tabel, jika pengguna telah memilih media penyimpanan dengan benar maka pengguna akan dibawa ke proses selanjutnya, sebaliknya jika pengguna salah dalam memilih maka pengguna akan tetap berada di halaman yang sama seperti pada diagram alir yang ditunjukkan pada Gambar 3.8.

Tahap selanjutnya adalah pengguna diminta untuk mengisikan data masukan melalui kolom-kolom yang disediakan oleh aplikasi *Forensic Imaging Application Using Raspberry Pi*. Kolom-kolom tersebut diantaranya adalah kolom *image directory* yang berfungsi untuk menentukan letak lokasi penyimpanan hasil akusisi, kemudian kolom *filename* yang berfungsi untuk penamaan *file* hasil akusisi. Selanjutnya adalah kolom *notes* untuk memberikan beberapa catatan terhadap *file* hasil akusisi, kemudian kolom *examiner* yang berguna untuk mengetahui siapa yang melakukan akusisi, dan terakhir adalah kolom *image type* berfungsi untuk menentukan tipe *file* hasil akusisi. Jika data yang diminta telah diisi dengan lengkap dan benar maka tahap selanjutnya adalah tahap proses akusisi. Tahap proses akusisi bekerja dengan cara menyalin bit demi bit data yang terdapat di dalam media penyimpanan yang ingin diakusisi.

Jika proses akusisi telah selesai maka proses selanjutnya aplikasi akan melakukan proses *hashing* MD5 dan SHA512. Hasil dari proses *hashing* adalah aplikasi akan menghasilkan MD5 dan SHA512 media penyimpanan yang diakusisi serta MD5 dan SHA512 *file* hasil akusisi. Kemudian alur sistem selanjutnya adalah proses aplikasi membuat *file logging* yang berisikan informasi detail terkait proses akusisi.

3.3.2 Pseudocode

Pseudocode adalah bentuk penulisan dari sebuah algoritma pemrograman yang disederhanakan agar seseorang mudah memahami alur dari sebuah algoritma pemrograman. Tidak semua orang mengerti Bahasa pemrograman, maka disnilah fungsi *pseudocode*, karena *pseudocode* disusun bukan dengan menggunakan Bahasa pemrograman melainkan dengan menggunakan Bahasa yang mudah dimengerti, sehingga algoritma sebuah pemrograman dapat dipahami oleh banyak orang.

A. Pseudocode Tampilan 1

Pseudocode tampilan 1 aplikasi *Forensic Imaging Application Using Raspberry Pi* menggunakan tiga *library* dengan beragam fungsinya masing-masing. Ketiga *library* yang digunakan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tabel *Library* Tampilan 1

No	Nama <i>Library</i>	Fungsi
1.	PyQt5	PyQt5 adalah <i>library</i> yang peneliti gunakan untuk tampilan GUI aplikasi.

No	Nama <i>Library</i>	Fungsi
2.	sys	sys digunakan untuk mengakses konfigurasi interpreter pada saat <i>runtime</i> dan berinteraksi dengan <i>environment</i> sistem operasi.
3.	os	Modul os peneliti gunakan agar dapat mengakses terminal, untuk menjalankan komen tertentu di terminal.

Alur algoritma pemrograman aplikasi *Forensic Imaging Application Using Raspberry Pi* dapat tergambar dengan jelas seperti yang tampak pada Gambar 3.9. Pada *pseudocode* tampilan 1 terdapat beberapa variabel yang digunakan, diantaranya adalah *diskx*, *i*, *j*, *k*, *line*, *listAx*, *listBx*, *listCx*, *mystruct*, *datax*, *itemx*, *item2x*, *item3x*, *hasil*, dan *size*.

<p>Aplikasi: <i>Forensic Imaging Application Using Raspberry Pi</i> {Aplikasi yang berfungsi untuk melakukan akusisi media penyimpanan}</p>
<p>Kamus :</p> <pre> PyQt5 : library sys : library os : library diskx : string i : int j : int k : int line : string listAx : string listBx : string listCx : string mystruct : string datax : string itemx : string item2x : string item3x : string display : QTableWidgetItem row : int hasil : string size : string </pre>
<p>Deskripsi Algoritma :</p> <pre> diskx ← os.popen("lsblk grep disk") for(line in diskx) do listAx ← line[0] listBx ← line[3] listCx ← line[5] mystruct ← {listAx, listBx, listCx} datax ← mystruct for(itemx in datax) do display.setItem(i,0,listAx) i = i+1 for(item2x in datax) do display.setItem(j,1,listBx) j = i+1 for(item3x in datax) do display.setItem(k,2,listCx) k = i+1 </pre>

```

hasil ← display(row,0).text()
size ← display(row,1).text()

```

Gambar 3.9 Pseudocode Tampilan 1

Variabel *diskx* adalah variabel yang memiliki tipe data *string* dan berfungsi untuk menyimpan hasil dari keluaran komen terminal yang dilakukan oleh *library os*. Kemudian variabel *i*, *j*, dan *k* adalah variabel dengan tipe data *integer* yang menyimpan bilangan *integer* digunakan di dalam perulangan *for*. Variabel *line* memiliki tipe data *string* yang berfungsi untuk menyimpan nilai *string* yang diperoleh dari variabel *diskx*. Sedangkan variabel *listAx*, *listBx*, *listCx* adalah variabel dengan tipe data *string* yang menyimpan nilai *string* yang didapat dari variabel *line*. Selanjutnya variabel *itemx*, *item2x*, dan *item3x* adalah variabel dengan tipe data *string* yang menyimpan nilai untuk kemudian ditampilkan ke dalam tabel. Sedangkan variabel *hasil* dan *size* adalah variabel dengan tipe data *string* yang menyimpan nilai *string* berupa nama dan ukuran kapasitas media penyimpanan yang pengguna pilih untuk diakusisi.

Untuk penjelasan algoritma *pseudocode* tampilan 1 adalah pertama-tama aplikasi akan melakukan pemindaian secara menyeluruh untuk mendeteksi berapa jumlah media penyimpanan yang terikat dengan perangkat keras Raspberry Pi, kemudian hasil dari pemindaian tersebut akan disimpan di dalam variabel *diskx*. Kemudian data hasil pemindaian yang disimpan di dalam variabel *diskx*, selanjutnya data itu disimpan di dalam variabel *line*, dan data itu akan di pecah menjadi ke dalam tiga bagian yaitu *line[0]* yang berfungsi untuk menyimpan nama penyimpanan, *line[3]* guna menyimpan ukuran kapasitas penyimpanan dan *line[5]* yang berfungsi untuk menyimpan jenis atau tipe media penyimpanan. Setelah data berhasil dibagi ke dalam tiga bagian, selanjutnya data tersebut akan di masukan ke dalam tabel, dan ditempatkan ke dalam tiga kolom yang berbeda. Proses terakhir adalah proses dimana pengguna aplikasi *Forensic Imaging Application Using Raspberry Pi* menentukan pilihannya tentang media penyimpanan mana yang ingin diakusisi, kemudian data pilihan tersebut akan disimpan ke dalam variabel *hasil* dan variabel *size*.

B. Pseudocode Tampilan 2

Pseudocode tampilan 2 aplikasi *Forensic Imaging Application Using Raspberry Pi* menggunakan tujuh *library* berbeda dengan beragam fungsinya masing-masing. Digunakannya tujuh modul atau *library* tersebut untuk menunjang proses berjalannya aplikasi, dan setiap modul memiliki perannya masing-masing. Penjelasan terkait ketujuh *library* yang digunakan dapat dilihat pada Tabel 3.2.

Tabel 3.2 Tabel *Library* Tampilan 2

No	Nama <i>Library</i>	Fungsi
1.	PyQt5	PyQt5 adalah <i>library</i> yang peneliti gunakan untuk tampilan GUI aplikasi.
2.	sys	sys digunakan untuk mengakses konfigurasi interpreter pada saat <i>runtime</i> dan berinteraksi dengan <i>environment</i> sistem operasi.
3.	hashlib	<i>Library</i> yang berfungsi untuk menjalankan fungsi <i>hashing</i> , baik MD5 dan SHA512.
4.	subprocess	Modul subprocess peneliti gunakan agar dapat mengakses terminal, untuk menjalankan komen tertentu di terminal.
5.	logging	Berfungsi untuk membuat <i>file logging</i> . Sehingga <i>log-log</i> selama proses aplikasi berjalan dapat disimpan di dalam sebuah <i>file</i> yang disebut <i>file logging</i> .
6.	time	<i>Library time</i> berfungsi untuk menyediakan fungsi waktu.
7.	datetime	<i>Library datetime</i> berfungsi untuk menyediakan fungsi waktu dan tanggal.
8.	os	Modul os peneliti gunakan agar dapat mengakses terminal, untuk menjalankan komen tertentu di terminal.

Alur algoritma pemrograman atau *pseudocode* tampilan 2 aplikasi *Forensic Imaging Application Using Raspberry Pi* dapat dilihat pada Gambar 3.10. Selain menggunakan tujuh *library*, pada *pseudocode* tampilan 2 menggunakan beberapa variabel yang berfungsi untuk menyimpan suatu nilai tertentu guna mendukung kelancaran kerja aplikasi. Variabel yang digunakan antara lain, direktori, *filename*, *note*, *examiner*, *radioDD*, *usbInfo1*, *usbInfo2*, *usbInfo3*, *source*, dan *blockSize*.

Variabel direktori adalah variabel dengan tipe data *string* yang berfungsi untuk menyimpan nilai *string* berupa direktori penyimpanan *file* hasil akusisi yang pengguna aplikasi tentukan. Variabel *filename* memiliki tipe data *string*, yang berfungsi untuk menyimpan masukan data *string* berupa nama *file* hasil akusisi. Selanjutnya, variabel *note* dan *examiner* keduanya memiliki tipe data *string*, yang memiliki fungsi masing-masing yakni, *note* berfungsi menyimpan nilai *string* berupa catatan tentang media penyimpanan yang diakusisi, sedangkan *examiner* berfungsi untuk menyimpan masukan *string* berupa nama penguji yang melakukan akusisi. Variabel *radioDD* berfungsi untuk menyimpan hasil masukan *string* berupa ekstensi format *file* yang diinginkan oleh pengguna aplikasi.

Variabel *usbInfo1*, *usbInfo2*, dan *usbInfo3* merupakan variabel dengan tipe data *string* yang memiliki fungsi untuk menyimpan keluaran hasil dari proses komen di terminal yang

dijalankan oleh *library* subprocess. Variabel *usbInfo1* menyimpan nilai *string* berupa label media penyimpanan yang diakusisi, sedangkan variabel *usbInfo2* menyimpan nilai *string* berupa UUID media penyimpanan yang diakusisi, dan variabel *usbInfo3* menyimpan nilai *string* berupa kapasitas ukuran dalam bentuk *bytes* media penyimpanan yang diakusisi. Selanjutnya, variabel *source* adalah variabel dengan tipe data *string* yang berfungsi untuk menyimpan nilai direktori penyimpanan. Kemudian *blockSize* adalah variabel dengan tipe data *integer* yang menyimpan nilai bilangan bulat. Variabel *blockSize* menyimpan nilai hasil perpangkatan dari 2 pangkat 20.

Aplikasi: Forensic Imaging Application Using Raspberry Pi

{Aplikasi yang berfungsi untuk melakukan akusisi media penyimpanan}

Kamus :

```
PyQt5 : library
sys : library
hashlib : library
subprocess : library
logging : library
time : library
datetime : library
os : library
direktori : string
filename : string
note : string
examiner : string
radioDD : string
usbInfo1 : string
usbInfo2 : string
usbInfo3 : string
source : string
blockSize : int
```

Deskripsi Algoritma :

```
input(direktori)
input(filename)
input(note)
input(examiner)
input(radioDD)

if (os.path.exist) then
    output("file exist")
elif (direktori="" or filename="" or note="" or examiner="" or radioDD="") then
    output("complete the field")
else
    usbInfo1 ← subprocess.getoutput("blkid -s LABEL")
    usbInfo2 ← subprocess.getoutput("blkid -s UUID")
    usbInfo3 ← subprocess.getoutput("fdisk -l")

open(source, read)
open(direktori, write)
while true
    if write = 0
        output("imaging finished")
        break

open(source)
while true
    read(blockSize)
    if not read
```

```

        break
        md5.update
        sha512.update

open(direkori)
while true
    read(blockSize)
    if not read
        break
    md5.update
    sha512.update

logging.basicConfig ← filename

```

Gambar 3.10 *Pseudocode* Tampilan 2

Untuk penjelasan mengenai algoritma pemrograman atau *pseudocode* tampilan 2 adalah setelah pengguna menentukan media penyimpanan yang ingin diakusisi pada tampilan 1, selanjutnya pengguna diminta untuk memasukkan data masukan. Data masukan pertama yang dibutuhkan aplikasi adalah data masukan berupa direktori penyimpanan, direktori penyimpanan ini berfungsi untuk menentukan letak lokasi penyimpanan *file* hasil akusisi. Data masukan direktori penyimpanan ini akan disimpan di dalam variabel direktori. Kemudian pengguna diminta untuk memasukkan data masukan untuk penamaan *file* hasil akusisi, data masukan ini akan disimpan di dalam variabel *filename*. Selanjutnya, pengguna diminta untuk memasukkan catatan tentang *file* akusisi dan juga penguji yang melakukan akusisi. Data masukan ini masing-masing akan dimasukkan ke dalam variabel *note* dan variabel *examiner*.

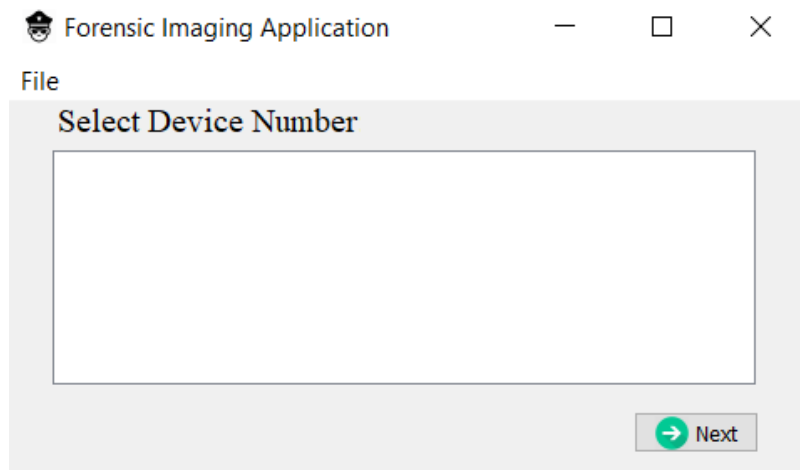
Setelah proses pengisian data masukan berjalan dengan lancar, maka algoritma selanjutnya adalah proses akusisi. Proses akusisi media penyimpanan akan berjalan sesuai dengan data masukan yang diberikan oleh pengguna, mulai dari letak posisi penyimpanan *file* hasil akusisi, nama *file* hasil akusisi, catatan dan penguji, serta format ekstensi hasil *file* akusisi. Jika proses akusisi telah selesai, maka algoritma selanjutnya adalah aplikasi melanjutkannya ke proses *hashing*, dari proses ini aplikasi akan menghasilkan dua jenis *hash*, yakni MD5 dan SHA512. Alur algoritma terakhir adalah aplikasi membuat *file logging* yang berisikan informasi detail terkait *file* hasil akusisi.

3.3.3 Desain Antarmuka

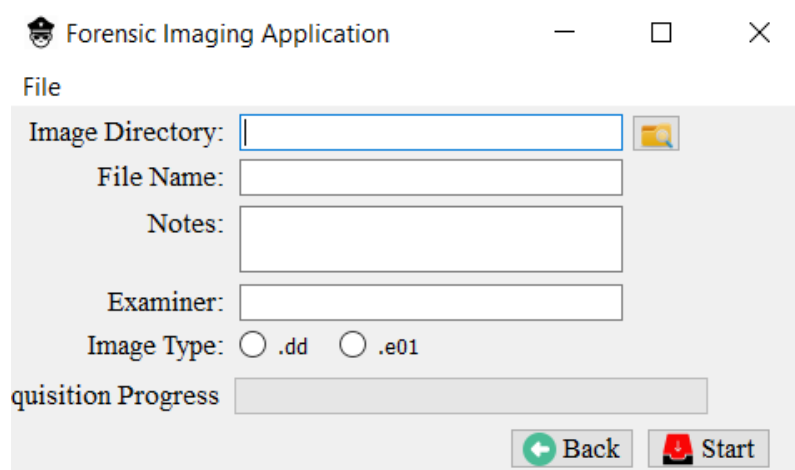
Desain antarmuka adalah proses rancangan desain, evaluasi desain, dan implementasi desain terhadap sebuah tampilan aplikasi. Proses desain antarmuka berguna untuk menciptakan tampilan yang mampu membuat pengguna senang berinteraksi dengan aplikasi karena tampilan aplikasi dianggap sederhana dan menarik. Desain antarmuka yang baik adalah desain yang mampu membuat pengguna tidak kesulitan dalam mengoperasikan aplikasi. Desain antarmuka

menjadi sangat penting, karena tampilan aplikasi akan menjadi jembatan antara si pengguna aplikasi dengan fungsi-fungsi yang terdapat di dalam aplikasi, sehingga ketika desain tampilan antarmuka aplikasi *Forensic Imaging Application Using Raspberry Pi* baik, maka seharusnya pengguna tidak akan mengalami kesulitan di dalam mengoperasikannya.

Dalam membuat rancangan desain tampilan aplikasi *Forensic Imaging Application Using Raspberry Pi* peneliti menggunakan perangkat lunak Qt Designer. Perangkat lunak Qt Designer adalah sebuah aplikasi yang digunakan untuk mendesain dan membuat tampilan GUI sebuah aplikasi. Di dalam aplikasi *Forensic Imaging Application Using Raspberry Pi* ini terdapat dua halaman seperti yang terlihat pada Gambar 3.11 dan Gambar 3.12.



Gambar 3.11 Tampilan Antarmuka 1



Gambar 3.12 Tampilan Antarmuka 2

Halaman awal seperti yang terlihat pada Gambar 3.11 akan tampil ketika pertama kali aplikasi dijalankan. Pada tampilan antarmuka yang pertama ini, aplikasi akan menampilkan sebuah tabel yang berisi daftar atau *list-list* media penyimpanan yang terikat dengan perangkat keras Raspberry Pi, yang dapat pengguna pilih untuk kemudian dilakukan proses akusisi. Pemilihan media penyimpanan yang tersedia pada tabel dilakukan dengan cara memilih nomor tabel, jadi pengguna aplikasi menekan pada bagian nomor dari media penyimpanan yang ingin dilakukan akusisi, dan setelah itu pengguna dapat melanjutkan proses selanjutnya dengan cara menekan tombol *Next*.

Setelah pengguna melewati halaman pertama, maka aplikasi akan langsung mengarahkan pengguna aplikasi *Forensic Imaging Application Using Raspberry Pi* kepada halaman ke dua seperti yang terlihat pada Gambar 3.12. Di halaman kedua ini berisikan masukan-masukan yang tersedia melalui kolom-kolom yang terdapat di aplikasi. Pengguna diminta untuk memberikan data masukan berupa data masukan *image directory* sebagai tempat lokasi penyimpanan *file* hasil akusisi. Selanjutnya data masukan *filename* yang berfungsi sebagai penamaan *file* akusisi. Kemudian data masukan *note* dan *examiner* yang masing-masing berfungsi untuk catatan *file* akusisi dan penguji atau yang menjalankan aplikasi *Forensic Imaging Application Using Raspberry Pi* ini. Kemudian data masukan terakhir adalah data masukan *image type*, pengguna diminta untuk menentukan format *file* hasil akusisi.

Setelah pengguna melengkapi keseluruhan data dengan benar, maka untuk proses selanjutnya pengguna diminta untuk menekan tombol *Start*. Selanjutnya ketika tombol *Start* ditekan, maka proses akusisi media penyimpanan akan berjalan. Ketika proses akusisi media penyimpanan telah selesai, maka aplikasi akan langsung menjalankan dua proses berikutnya secara berturut-turut yakni proses *hashing* MD5 dan SHA512 serta proses pembuatan dokumen *file logging*.

3.4 Pengujian Skala Likert

Skala likert adalah skala yang menggunakan beberapa butir pertanyaan untuk mengukur perilaku individu dengan merespon lima titik pilihan pada setiap butir pertanyaan, sangat setuju, setuju, tidak memutuskan, tidak setuju, dan sangat tidak setuju (Likert, 1932). Skala likert adalah skala pengukuran yang dikembangkan oleh Likert, skala likert mempunyai empat atau lebih butir-butir pertanyaan yang dikombinasikan sehingga membentuk sebuah skor atau nilai yang merepresentasikan sifat individu, misalkan pengetahuan, sikap, dan perilaku. Dalam proses analisis data, komposit skor, biasanya jumlah atau rata-rata, dari semua butir pertanyaan

dapat digunakan, penggunaan jumlah dari semua butir pertanyaan adalah indikator dari variabel yang direpresentasikannya (Budiaji, 2013).

Untuk mendapatkan data tentang pengujian aplikasi dengan skala likert, peneliti menggunakan metode pengumpulan data dengan cara menyebarkan kuesioner melalui *google form* ke sejumlah pihak yang dirasa mampu merepresentasikan data yang ingin didapatkan. Setelah mendapatkan data hasil kuesioner, selanjutnya data tersebut akan diolah dengan skala likert sehingga mendapatkan kesimpulan terhadap aplikasi yang dibuat oleh peneliti. Untuk mendapatkan kesimpulan dari pengguna aplikasi, langkah pertama yang dilakukan adalah mendapatkan total skor dari kuesioner yang telah diisi oleh pengguna, untuk mendapatkan total skor tersebut dapat menggunakan persamaan (3.1). Setelah menghitung nilai total skor yang didapatkan melalui persamaan (3.1), maka langkah selanjutnya adalah menghitung rata-rata skor kuesioner. Untuk mendapatkan nilai rata-rata skor kuesioner dapat menggunakan persamaan (3.2).

$$\text{Total Skor} = \text{Jumlah responden yang memilih} \times \text{Pilihan Nilai Likert} \quad (3.1)$$

$$\text{Rata - Rata} = \frac{\text{Total Skor}}{\text{Total Responden}} \quad (3.2)$$

Skala likert adalah skala yang menggunakan beberapa butir pertanyaan untuk mengukur perilaku individu dengan merespon lima titik pilihan pada setiap butir pertanyaan (Likert, 1932). Maka untuk memenuhi perhitungan nilai skala likert, peneliti menggunakan lima skala pada kuesioner yang peneliti sebar yang terdiri dari sangat setuju, setuju, netral, tidak setuju, dan sangat tidak setuju. Untuk penilaian pertanyaan dapat dilihat pada Tabel 3.3.

Tabel 3.3 Nilai Skala Likert

Nilai	Pertanyaan Positif	Pertanyaan Negatif
5	Jawaban Sangat Setuju (SS)	Jawaban Sangat Tidak Setuju (STS)
4	Jawaban Setuju (S)	Jawaban Tidak Setuju (TS)
3	Jawaban Netral (N)	Jawaban Netral (N)
2	Jawaban Tidak Setuju (TS)	Jawaban Setuju (S)
1	Jawaban Sangat Tidak Setuju (STS)	Jawaban Sangat Setuju (SS)

Kemudian untuk menentukan kriteria nilai skala likert dapat menggunakan rumus Rentang Skala (RS) dengan cara skor tertinggi dikurangi oleh skor terendah kemudian dibagi dengan jumlah kategori. Rumus Rentang Skala (RS) dapat dilihat pada persamaan (3.3). Jika menggunakan persamaan (3.3) untuk mendapatkan nilai rentang skala likert maka didapatkan hasil rentang skalanya adalah 0,8. Hasil tersebut didapatkan melalui skor tertinggi dikurangi skor terendah kemudian dibagi dengan jumlah kategori $(5-1)/5= 0,8$. Selanjutnya kriteria rentang skala likert dapat dilihat pada Tabel 3.4. Sedangkan untuk melihat butir-butir pertanyaan pada kuesioner dapat dilihat pada Tabel 3.5.

$$\text{Rentang Skala} = \frac{\text{Skor Tertinggi} - \text{Skor Terendah}}{\text{Jumlah Katergori}} \quad (3.3)$$

Tabel 3.4 Rentang Skala Likert

Nilai	Rentang	Keterangan
1	1 – 1,79	Sangat Tidak layak
2	1,8 – 2,59	Tidak Layak
3	2,60 – 3,39	Netral
4	3,40 – 4,19	Layak
5	4,20 – 5,00	Sangat Layak

Tabel 3.5 Pertanyaan Kuesioner

NO	PERNYATAAN	NILAI				
		STS	TS	N	S	SS
A	MANFAAT					
1	Perangkat forensic imaging ini memiliki manfaat bagi saya yang bergelut dibidang forensic					
2	Penggunaan single board computer Raspberry Pi pada perangkat forensic imaging ini mampu menekan biaya pembuatan perangkat					
3	Menurut saya perangkat forensic imaging ini mudah dibawa karena memiliki ukuran yang kecil					
4	Saya akan menggunakan aplikasi ini ketika ingin melakukan cloning penyimpanan					

NO	PERNYATAAN	NILAI				
5	Saya harap aplikasi ini dapat digunakan orang banyak					
6	Keluaran yang dihasilkan aplikasi ini sudah sesuai dengan yang saya harapkan					
B	TAMPILAN	STS	TS	N	S	SS
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	STS	TS	N	S	SS
1	Fitur auto run aplikasi berjalan dengan baik					
2	Fitur auto scan untuk mendeteksi penyimpanan apa saja yang tersedia berjalan dengan baik					
3	Semua tombol dan menu dapat bekerja dengan normal					
4	Semua error handling dapat berjalan dengan baik					

BAB IV

HASIL DAN PEMBAHASAN

Hasil menurut kamus besar Bahasa Indonesia (KBBI) adalah sesuatu yang diadakan (dibuat, dijadikan, dan sebagainya) oleh usaha, sedangkan pembahasan menurut KBBI adalah proses, cara, perbuatan membahas. Maka tahap hasil dan pembahasan adalah tahap dimana peneliti menuangkan ide dan konsep, rancangan, serta analisis yang telah dibuat pada pembahasan bab sebelumnya serta menyajikan hal-hal yang telah diraih selama proses penelitian dilakukan.

Tahapan hasil dan pembahasan meliputi capaian yang telah diraih oleh peneliti selama proses penelitian berlangsung, kemudian pembahasan implementasi aplikasi *Forensic Imaging Application Using Raspberry Pi* baik itu pembahasan implementasi antarmuka maupun kode dan fungsi program. Selanjutnya pengujian terhadap aplikasi yang peneliti buat, pengujian ini meliputi pengujian menggunakan teknik *blackbox*, pengujian performa aplikasi dan pengujian melalui kuesioner untuk melihat sejauh mana kepuasan pengguna aplikasi terhadap aplikasi yang peneliti buat ini. Hasil kuesioner yang telah didapat kemudian akan dihitung menggunakan persamaan skala likert.

4.1 Hasil

4.1.1 Perangkat Keras

Penelitian pembuatan perangkat *forensic imaging* ini menggunakan *single board computer* Raspberry Pi sebagai mesin utama pengendali perangkat. Raspberry Pi yang peneliti gunakan adalah Raspberry Pi 3 model B yang dilengkapi dengan 4 buah *port* USB 2.0, WLAN, LAN, dan *port* HDMI sebagai keluaran tampilannya. Sebagai sumber dayanya, perangkat keras Raspberry Pi dapat menggunakan power adapter yang biasa digunakan pada telpon genggam.

Penggunaan *single board computer* Raspberry Pi sebagai mesin utama pengendali perangkat *forensic imaging* karena Raspberry Pi adalah sebuah komputer dengan harga yang cukup terjangkau, disisi lain Raspberry Pi memiliki keunggulan dengan bentuk ukuran fisiknya yang kecil yang hanya sebesar kartu kredit. Carrie Anne Philbin dalam bukunya yang berjudul *Adventures in Raspberry Pi* mengatakan, Raspberry Pi adalah komputer dengan ukuran yang sangat kecil, hanya sebesar kartu kredit, dan Raspberry Pi dapat dibangun dan dikontrol sesuai dengan kebutuhan pengguna (Philbin, 2014). Sedangkan untuk monitornya peneliti

menggunakan LCD monitor layar sentuh dengan resolusi layar sebesar 4inch, LCD monitor ini dirangkai dengan Raspberry sehingga bisa saling terintegrasi satu sama lain. Untuk melihat tampilan perangkat keras Raspberry Pi dan LCD monitor dapat dilihat pada Gambar 4.1 Raspberry Pi 3 dan Gambar 4.2 Raspberry Pi Dengan LCD Monitor.



Gambar 4.1 Raspberry Pi 3



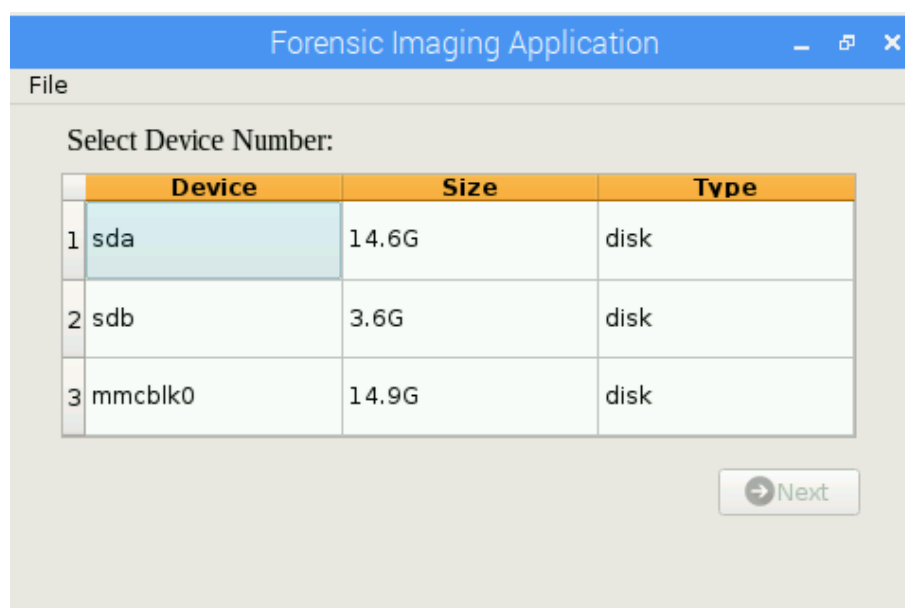
Gambar 4.2 Raspberry Pi Dengan LCD Monitor

4.1.2 Perangkat Lunak

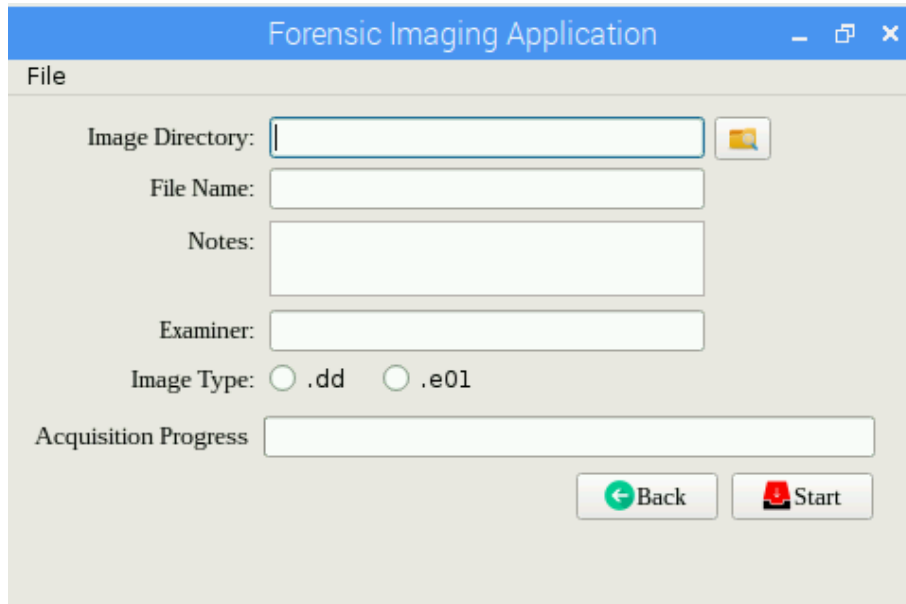
Untuk mengoperasikan perangkat *forensic imaging* maka peneliti membuat aplikasi guna mengoperasikan perangkat *forensic imaging* tersebut. Aplikasi yang peneliti buat

menggunakan Bahasa pemrograman *Python* versi 3.5.3 sesuai dengan desain yang sudah peneliti buat dengan bantuan perangkat lunak QT Designer. Aplikasi *Forensic Imaging Application Using Raspberry Pi* memiliki dua tampilan antarmuka, yakni halaman utama yang berisikan tabel penyimpanan dan halaman kedua adalah halaman tempat pengguna aplikasi memasukan data yang diperlukan seperti yang terlihat pada Gambar 4.3 dan Gambar 4.4.

Pada halaman utama aplikasi berisikan tabel yang memuat seluruh perangkat penyimpanan yang tersedia pada perangkat keras Raspberry Pi. Pada halaman utama ini pengguna aplikasi diminta untuk memilih media penyimpanan mana yang ingin pengguna akusisi, untuk memilihnya pengguna aplikasi dapat menekan nomor pada tabel penyimpanan yang ingin diakusisi. Setelah berhasil memilih penyimpanan yang ingin diakusisi, maka tombol *Next* pada halaman utama akan aktif, selanjutnya pengguna dapat menekan tombol *Next* tersebut untuk pindah ke halaman selanjutnya. Pada halaman kedua aplikasi berisikan kolom-kolom data yang harus diisi oleh pengguna aplikasi *Forensic Imaging Application Using Raspberry Pi*. Kolom data yang harus diisi adalah *image directory*, *file name*, *note*, *examiner*, dan *image type* seperti yang terlihat pada Gambar 4.4. Pada halaman kedua aplikasi ini pengguna harus mengisi semua kolom-kolom data secara lengkap dan benar, setelah melengkapi kolom data secara lengkap dan benar maka pengguna selanjutnya menekan tombol *Start* untuk memulai proses aplikasi. Ketika proses aplikasi berjalan, maka pengguna harus menunggu hingga proses akusisi selesai.



Gambar 4.3 Halaman Tabel Penyimpanan



Gambar 4.4 Halaman Pegisian Data

4.2 Pembahasan Implementasi Aplikasi

4.2.1 Pembahasan Implementasi Kode Program

Bahasa pemrograman yang digunakan peneliti untuk mengimplementasikan pada aplikasi adalah Bahasa pemrograman *Python* versi 3.5.3 yang memang sudah menjadi versi bawaan dari perangkat keras *Raspberry Pi*. Dalam tahap implementasi kode program, peneliti membuat dua buah *file* kode program, yang masing-masing berguna untuk tampilan halaman 1 aplikasi yang merupakan halaman utama dari aplikasi dan tampilan halaman 2 aplikasi yang merupakan halaman tempat dimana pengguna aplikasi mengisikan data yang diperlukan untuk proses akusisi. Karena memang untuk membangun aplikasi *Forensic Imaging Application Using Raspberry Pi* ini membutuhkan dua halaman tampilan guna mendukung proses kerja aplikasi.

Pada *file* kode program pertama yang mana berfungsi untuk tampilan halaman 1 yang merupakan halaman utama aplikasi *Forensic Imaging Application Using Raspberry Pi*, peneliti menggunakan setidaknya tiga buah modul atau *library* seperti yang terlihat pada Gambar 4.5, masing-masing adalah *library* *PyQt5*, *sys*, dan *os*. Modul *PyQt5* peneliti gunakan untuk kepentingan tampilan *Graphical User Interface* (GUI) aplikasi, mulai dari tampilan tabel hingga tampilan tombol itu semua menggunakan *library* *PyQt5*. Sedangkan modul *sys* digunakan untuk mengintegrasikan antara *interpreter Python* dengan sistem sehingga bisa saling sinkron satu sama lain. Kemudian modul *os* peneliti gunakan untuk menjalankan

terminal di dalam *Python* dan memberikan komen perintah “lsblk | grep disk” pada terminal tersebut seperti yang terlihat pada Gambar 4.6.

```
import sys
from PyQt5.QtWidgets import *
from PyQt5 import QtCore, QtGui
from interface2 import *
import os
```

Gambar 4.5 *Library* Kode Pemrograman Halaman 1

```
diskx = [s.split() for s in os.popen("lsblk | grep disk").read().splitlines()]
i=0
j=0
k=0
for line in diskx:
    listAx = line[0]
    listBx = line[3]
    listCx = line[5]
    mystruct = {'A':listAx, 'B':listBx, 'C':listCx}
    self.datax = mystruct
    jmlhRow = len(diskx)
    self.display.setRowCount(jmlhRow)
    self.display.setColumnCount(3)
    self.display.setHorizontalHeaderLabels(["Device", "Size", "Type"])
```

Gambar 4.6 Fungsi Untuk Menampilkan Penyimpanan

Sedangkan pada *file* kode program kedua yang mana berfungsi untuk tampilan halaman 2 yang merupakan tempat dimana pengguna aplikasi *Forensic Imaging Application Using Raspberry Pi* mengisikan data yang diperlukan untuk proses akusisi, peneliti menggunakan setidaknya delapan buah modul atau *library* seperti yang terlihat pada Gambar 4.7. *Library* yang digunakan masing-masing adalah *library* PyQt5, sys, hashlib, subprocess, logging, time, datetime dan os. Setiap *library* memiliki fungsinya masing-masing, seperti *library* logging digunakan untuk membuat *file logging* yang memuat informasi detail tentang proses akusisi. Detail lengkap fungsi setiap modul dapat dilihat pada Tabel 3.2 Tabel *Library* Tampilan 2.

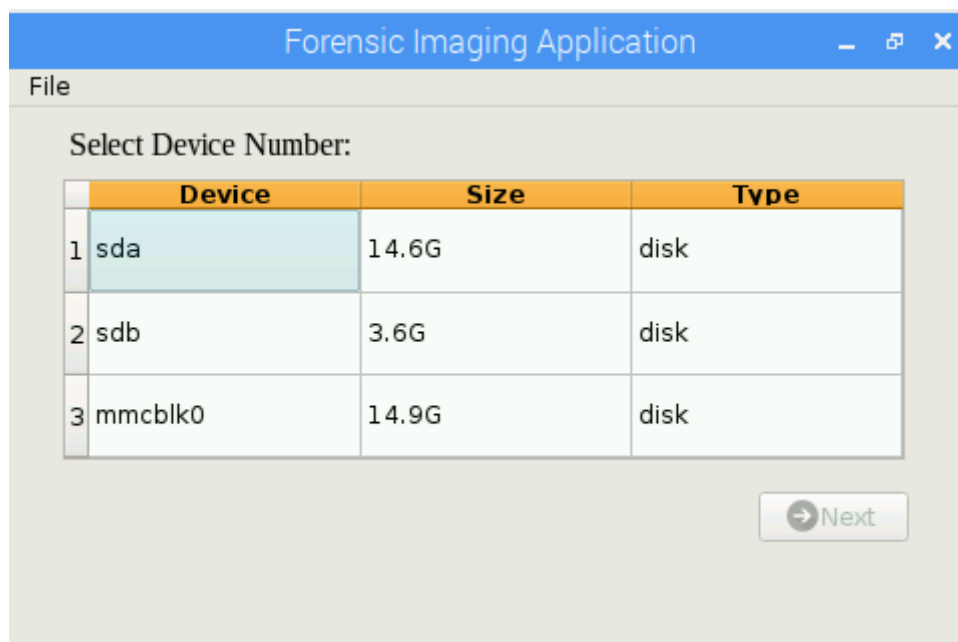
```
from PyQt5.QtWidgets import *
from PyQt5 import QtGui, QtCore
from PyQt5.QtCore import QThread, pyqtSignal
import sys
import hashlib
import subprocess
import logging
import time
import datetime
import os
```

Gambar 4.7 *Library* Kode Pemrograman Pada Halaman 2

4.2.2 Pembahasan Implementasi Antarmuka

Pada aplikasi *Forensic Imaging Application Using Raspberry Pi* ini terdapat dua tampilan halaman, yakni halaman 1 yang berisi berupa sajian daftar *list* media penyimpanan yang dapat dipilih untuk diakusisi, sedangkan halaman 2 berisi kolom-kolom yang harus diisi oleh pengguna sebagai data masukan untuk aplikasi. Kedua halaman tersebut memiliki resolusi yang sama yakni sebesar 480 x 250, resolusi tersebut merupakan hasil analisis mengingat peneliti menggunakan layar LCD *monitor* yang memiliki ukuran hanya 4*inch*, sehingga resolusi tersebut merupakan resolusi terbaik untuk layar LCD 4*inch*.

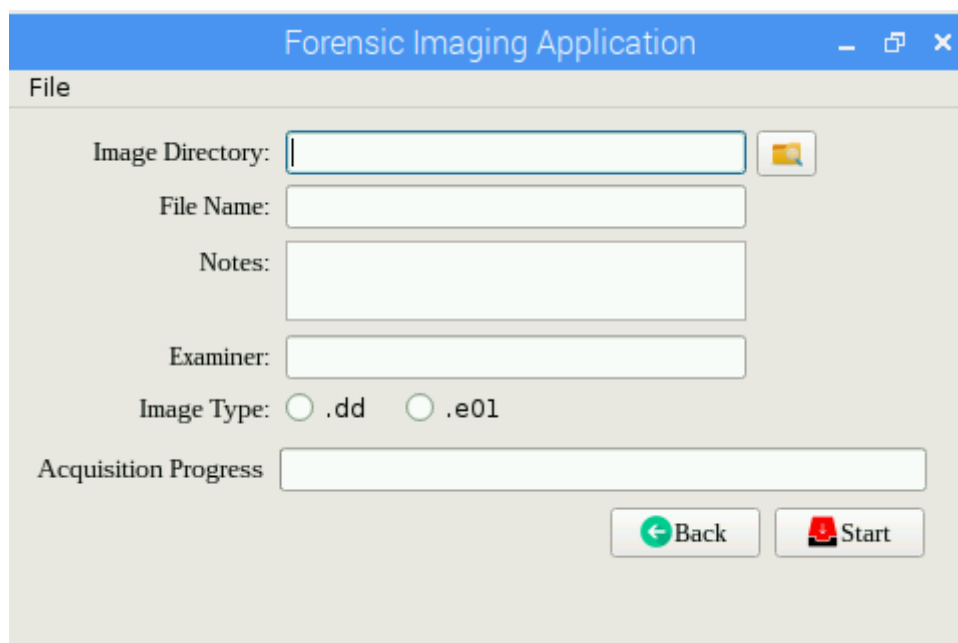
Pada halaman tampilan 1 aplikasi *Forensic Imaging Application Using Raspberry Pi* berisi sajian data *list-list* media penyimpanan yang terikat dengan perangkat keras Raspberry Pi yang pengguna dapat pilih untuk dilakukan akusisi seperti yang terlihat pada Gambar 4.8. Di dalam tabel tersebut terbagi ke dalam tiga kolom, masing-masing kolom tersebut adalah kolom *Device*, *Size*, dan *Type*. Kolom *Device* merupakan kolom yang memuat nama media penyimpanan, sedangkan kolom *Size* merupakan kolom yang memuat ukuran kapasitas media penyimpanannya, dan kolom *Type* memuat tipe dari media penyimpanannya.



Gambar 4.8 *Interface 1*

Selain terdapat tabel, pada halaman tampilan 1 aplikasi *Forensic Imaging Application Using Raspberry Pi* juga terdapat *menu bar* berupa *file*. *Menu bar file* itu memuat aksi *exit*, jadi ketika pengguna aplikasi ingin keluar dari aplikasi, pengguna tersebut memiliki dua opsi,

yang pertama adalah dengan menggunakan tanda silang yang terletak pada pojok kanan atas aplikasi, dan yang kedua adalah dengan menggunakan aksi *exit* yang terdapat di dalam *menu bar file*. Kemudian pada tampilan halaman 1 juga terdapat sebuah tombol dengan nama tombol *Next*, tombol tersebut berfungsi untuk membawa pengguna berpindah ke halaman selanjutnya. Tombol tersebut tidak akan bisa di tekan oleh pengguna karena memang tombol tersebut didesain untuk tidak aktif sampai pengguna telah memilih media penyimpanan mana yang ingin pengguna akusisi. Untuk memilih media penyimpanan yang ingin diakusisi, pengguna hanya perlu menekan pada bagian nomor yang terdapat di tabel.



Gambar 4.9 *Interface 2*

Pada halaman tampilan 2 aplikasi *Forensic Imaging Application Using Raspberry Pi* terdapat kolom-kolom yang harus diisi oleh pengguna seperti yang terlihat pada Gambar 4.9. Kolom *image directory* berfungsi untuk menyimpan letak lokasi penyimpanan *file* hasil akusisi. Untuk mengisi kolom *image directory* pengguna tidak perlu repot-repot menuliskannya secara manual, karena aplikasi sudah menyiapkan tombol yang dapat digunakan dengan mudah untuk menentukan letak lokasi penyimpanan *file* hasil akusisi.

Selanjutnya terdapat kolom *filename*, kolom ini berfungsi untuk penamaan *file* akusisi. Pengguna hanya perlu menuliskan nama *file* hasil akusisi yang diinginkan tanpa perlu menggunakan format ekstensinya. Setelah mengisi kolom *filename*, selanjutnya pengguna harus mengisi kolom *notes*. Kolom *notes* ini berfungsi untuk memberikan beberapa catatan

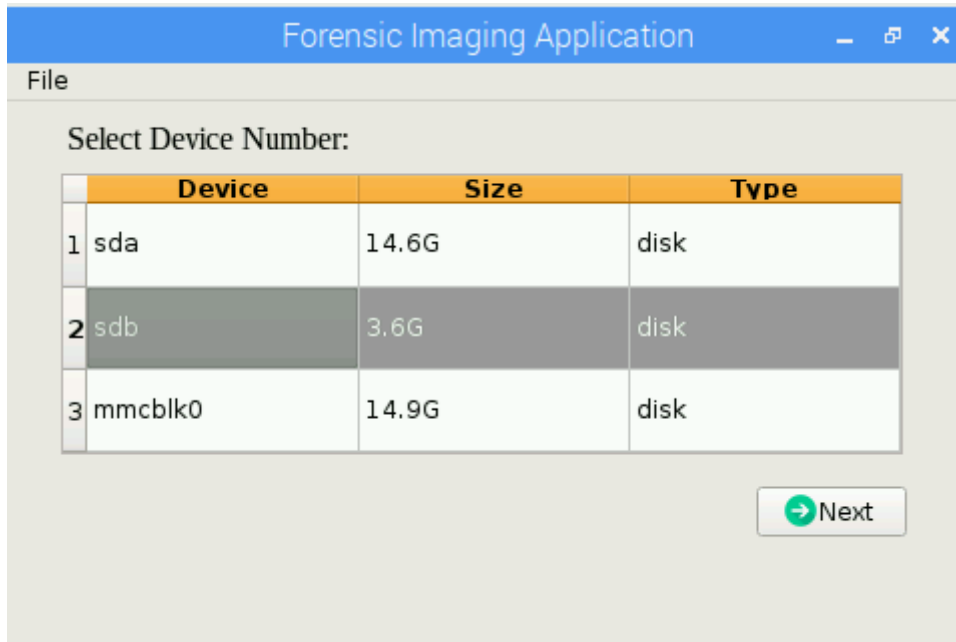
singkat terkait media penyimpanan yang ingin di akusisi. Sedangkan kolom *examiner* berfungsi sebagai informasi tentang siapa pengguna aplikasi dan siapa yang melakukan akusisi. Kolom terakhir yang harus diisi oleh pengguna aplikasi *Forensic Imaging Application Using Raspberry Pi* adalah kolom *image type*. Kolom ini yang menentukan format apakah yang ingin dihasilkan oleh aplikasi ini.

Selain terdapat kolom-kolom, pada halaman tampilan 2 aplikasi juga terdapat tombol-tombol yang dapat digunakan oleh pengguna. Tombol *Back* dapat digunakan oleh peneliti jika memang peneliti ingin kembali ke halaman sebelumnya, sedangkan tombol *Start* adalah tombol yang digunakan jika peneliti ingin proses akusisi dijalankan. Sedangkan tombol dengan logo *folder* adalah tombol yang dapat peneliti gunakan untuk mengisi kolom *image directory*. Selain itu terdapat *menu bar file* yang dapat peneliti gunakan jika ingin keluar dari aplikasi.

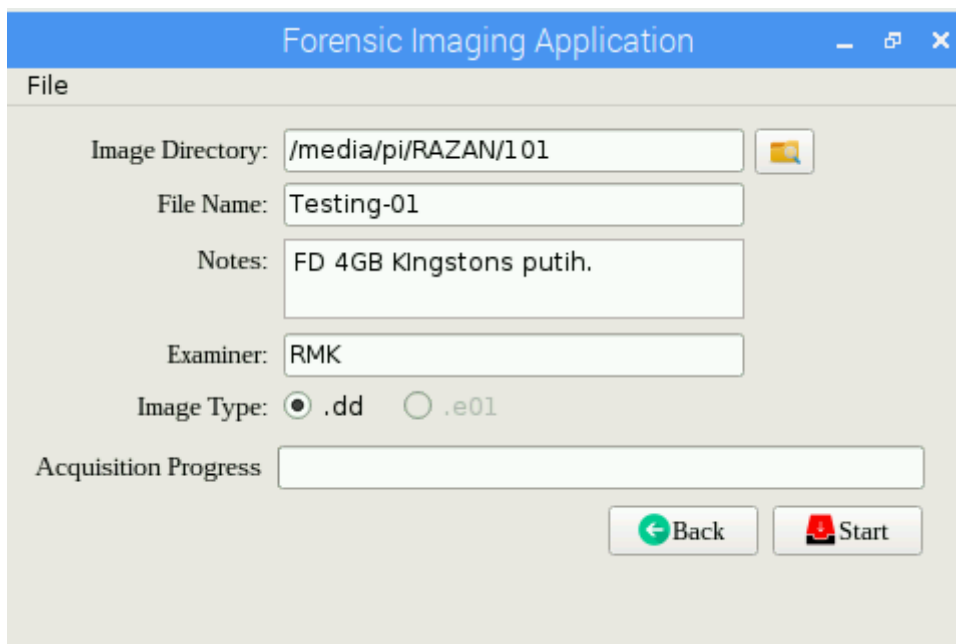
4.2.3 Pembahasan Implementasi Proses Akusisi

Implementasi proses akusisi dilakukan untuk menguji sejauh mana aplikasi *Forensic Imaging Application Using Raspberry Pi* dapat bekerja. Untuk memulai proses akusisi peneliti menjalankan aplikasi seperti yang terlihat pada Gambar 4.10. Kemudian peneliti memilih media penyimpanan dengan nama *sdb* dengan kapasitas penyimpanan 3,6 *Gigabyte* seperti yang terlihat pada Gambar 4.10. Setelah peneliti menentukan penyimpanan mana yang ingin diakusisi, selanjutnya peneliti menekan tombol *Next*, dimana tombol tersebut akan membawa peneliti ke halaman tampilan 2 aplikasi.

Pada halaman tampilan 2 aplikasi *Forensic Imaging Application Using Raspberry Pi*, peneliti diminta untuk memasukan data masukan pada kolom-kolom yang telah disediakan oleh aplikasi seperti yang terlihat pada Gambar 4.11. Pada kolom *image directory* peneliti ingin menyimpan *file* hasil akusisi di *folder* “/media/pi/RAZAN/101”. Selanjutnya pada kolom penamaan *file* akusisi, peneliti mengisi kolom *filename* dengan nama “*Testing-01*”, maka ketika *file* hasil akusisi telah selesai, *file* tersebut akan memiliki nama “*Testing-01*”. Kemudian pada kolom *notes*, peneliti memberikan catatan yang berisi “FD 4GB Kingstons putih.”. Catatan ini dapat digunakan untuk memberikan ciri tertentu kepada *file* hasil akusisi terhadap barang bukti yang diakusisi. Selanjutnya peneliti mengisikan kolom *examiner* dengan nama “RMK”. Terakhir peneliti mengisikan kolom *image type* dengan format *file* *.dd*. Setelah melengkapi dan merasa yakin dengan data yang sudah diisi, maka untuk menjalankan akusisi dapat dilakukan dengan cara menekan tombol *Start*.



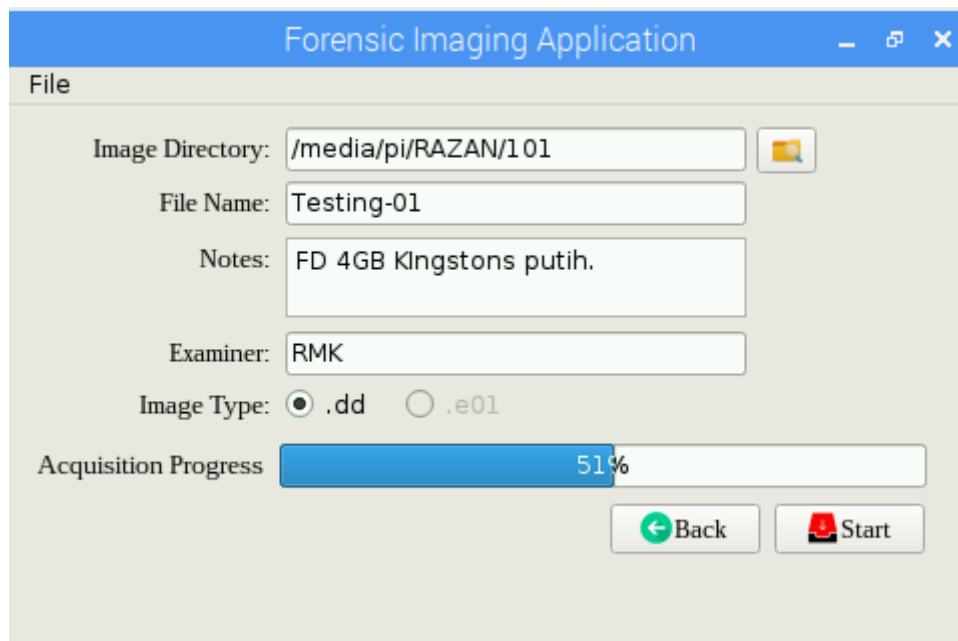
Gambar 4.10 Pemilihan Akusisi Penyimpanan



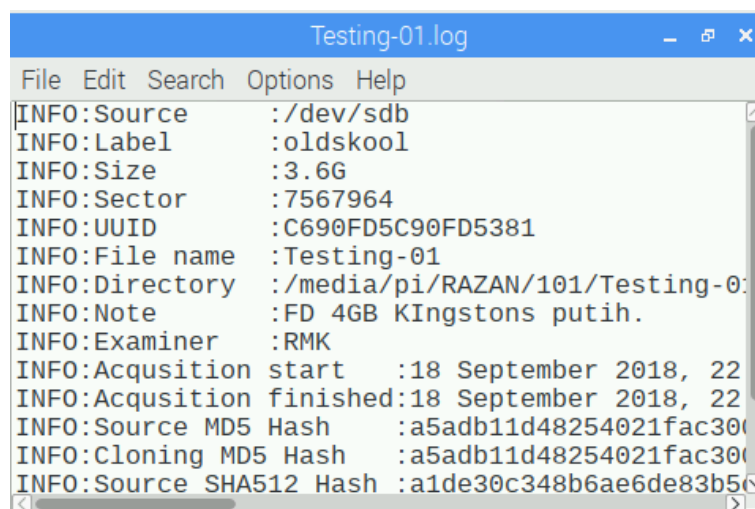
Gambar 4.11 Pengisian Kolom Masukan

Setelah tombol *Start* ditekan, maka proses akusisi media penyimpanan akan berjalan. Durasi lamanya proses akusisi tergantung dari seberapa besar ukuran kapasitas penyimpanan yang diakusisi. Selama proses akusisi berlangsung, pengguna dapat mengetahui sejauh mana akusisi berjalan melalui *progress bar* yang terdapat di tampilan halaman 2 aplikasi *Forensic Imaging Application Using Raspberry Pi*. Seperti yang terlihat pada Gambar 4.12 bahwa

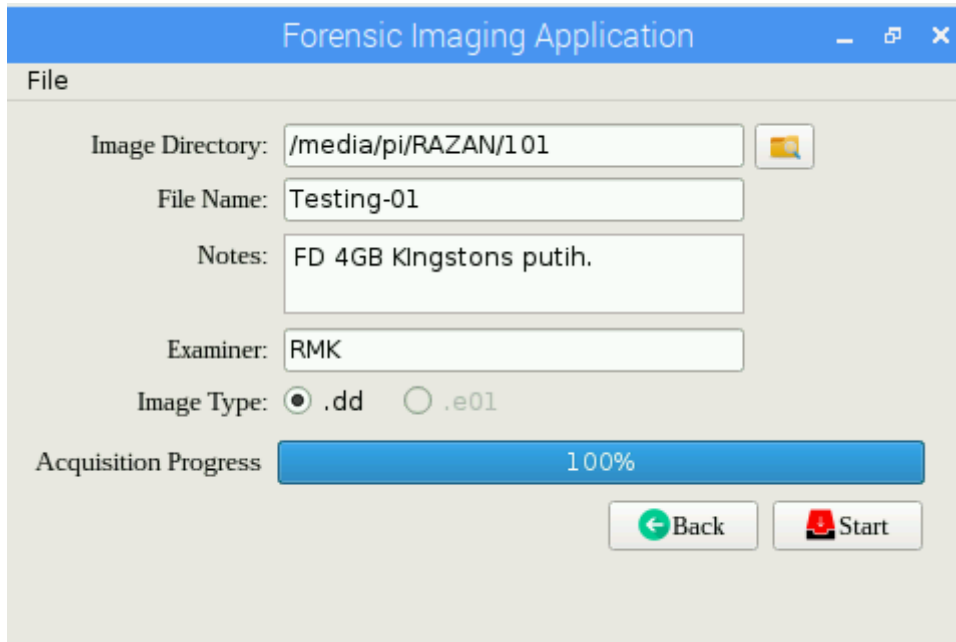
progres akuisi penyimpanan telah mencapai 51%. Ketika indikator persentase *progress bar* telah mencapai 100% seperti yang terlihat pada Gambar 4.14, itu artinya proses akuisi telah selesai dan berjalan dengan sukses. Tetapi proses keseluruhan belum selesai, karena setelah melakukan proses akuisi, aplikasi melanjutkannya dengan melakukan proses *hashing* MD5 dan SHA512 serta dilanjutkan dengan proses membuat dokumen yang berisikan *file logging* yang menyimpan rincian informasi detail terkait proses akuisi ini seperti yang terlihat pada Gambar 4.13.



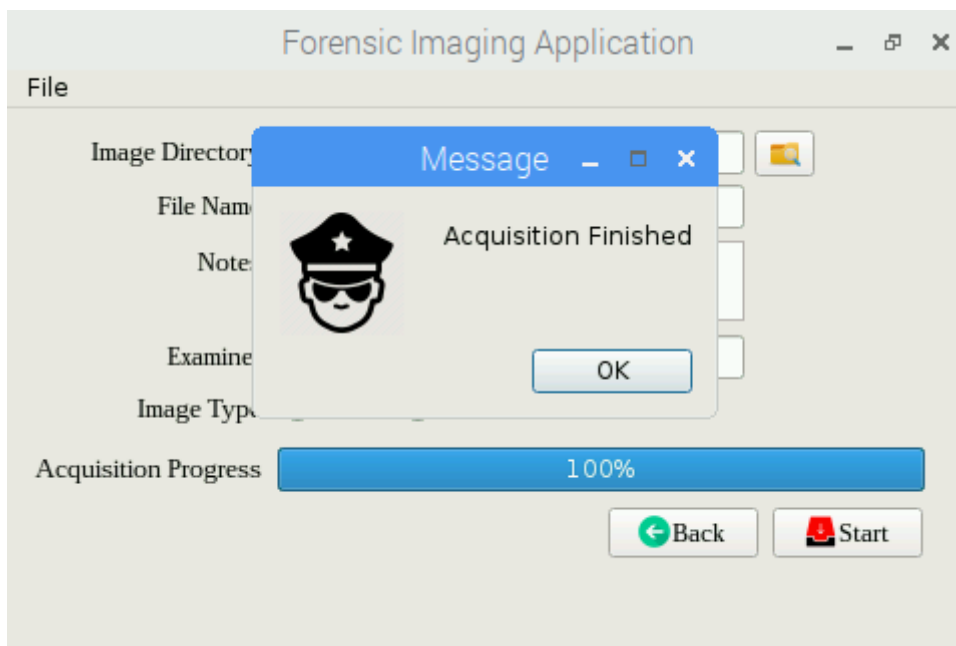
Gambar 4.12 Progres Akuisi 51%



Gambar 4.13 File Logging



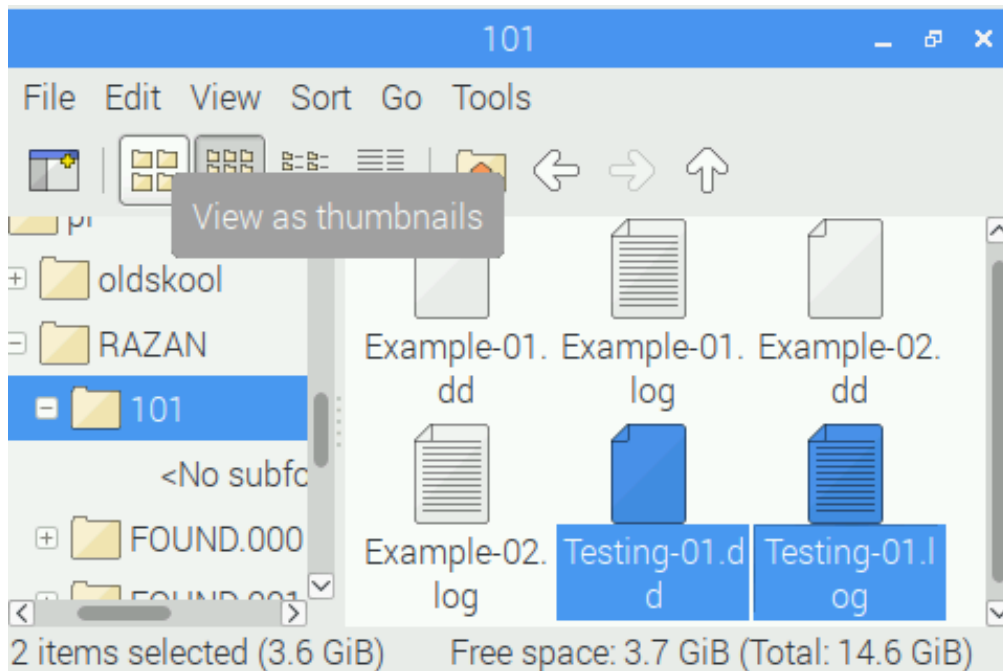
Gambar 4.14 Progres Akusisi 100%



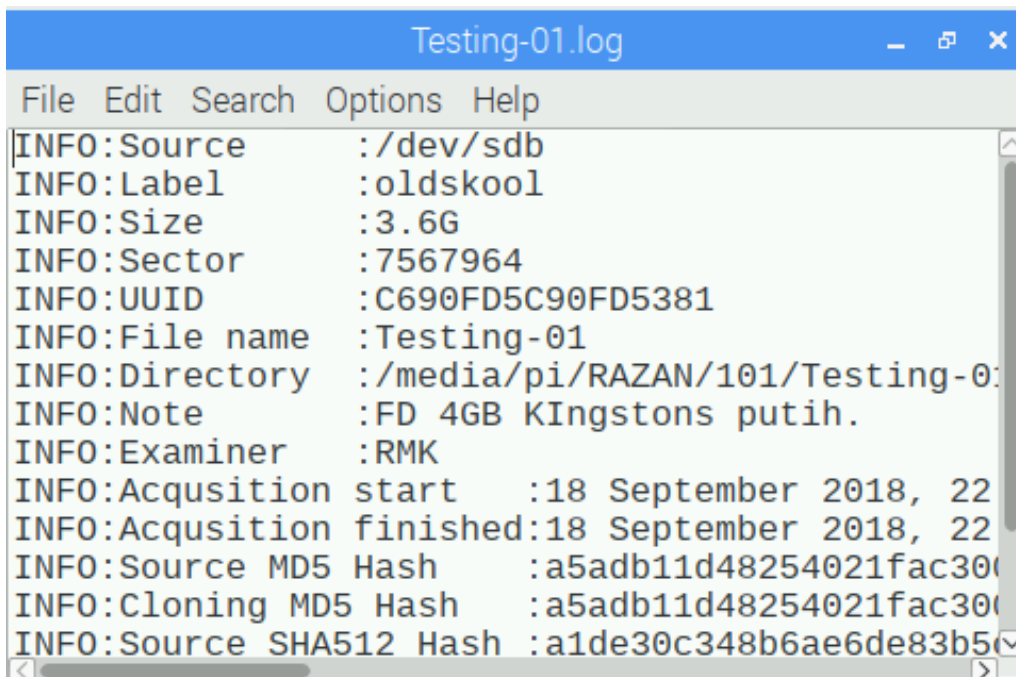
Gambar 4.15 Proses Akusisi Selesai

Jika keseluruhan proses telah diselesaikan oleh aplikasi baik itu proses akusisi penyimpanan, proses *hashing*, hingga proses membuat *file logging*, maka aplikasi akan memberitahukan pengguna melalui *messagebox* dengan isi pesannya adalah “*Acquisition Finished*” seperti yang terlihat pada Gambar 4.15. Untuk melakukan pengecekan keluaran apa saja yang dihasilkan oleh aplikasi dapat dilihat pada Gambar 4.16. Dari Gambar 4.16 dapat

dilihat bahwa keluaran yang dihasilkan oleh aplikasi *Forensic Imaging Application Using Raspberry Pi* adalah berupa *file* hasil akusisi berikut dokumen *file logging* yang berisikan informasi detail seperti yang terlihat pada Gambar 4.17.



Gambar 4.16 Keluaran Aplikasi



Gambar 4.17 File Logging


```

class Akusisi (QtCore.QThread) :
    countChanged = pyqtSignal(int)
    selesai = pyqtSignal(int)

    def run(self):

        #direktori penyimpanan
        c = mainWindow_2.c
        #source cloning
        source = mainWindow_2.source

        val = 0

        #proses cloning
        with open(source,'rb') as f:
            with open(c, "wb") as i:
                while True:
                    val += 1
                    self.countChanged.emit(val)
                    if i.write(f.read(512)) == 0:
                        print("Imaging Finished")
                        break

        self.play = Checksum()
        self.play.finished.connect (self.acquisitionFinished)
        self.play.start()

    def acquisitionFinished(self):
        nil = 1
        self.selesai.emit(nil)

```

Gambar 4.18 Fungsi Proses Akusisi

Untuk kode program bagaimana proses akusisi berjalan dapat dilihat pada Gambar 4.18. Kode program akusisi terpisah dari kelas utama, kode program akusisi berada dikelas sendiri yang merupakan kelas *thread*. Kelas *thread* ini digunakan sebagai solusi atas penggunaan *progress bar*, karena untuk selalu melakukan *update* nilai *progress bar* maka dibutuhkan sebuah kelas *thread* yang mana nanti setiap kali proses akusisi mengalami kemajuan maka akan dikirimkan sinyal ke kelas utama untuk segera merubah nilai dari *progress bar*.

Sedangkan untuk kode program bagaimana proses pembuatan *file logging* dan proses *hashing* berjalan dapat dilihat pada Gambar 4.19. Proses pembuatan *file logging* dan *hashing* terpisah dari kelas utama, melainkan berjalan di kelas *thread* nya sendiri dengan nama kelas *Checksum*. Kelas *thread Checksum* ini akan berjalan apabila proses akusisi penyimpanan yang berlangsung dikelas *thread* sebelumnya telah selesai. Maka ketika kelas *Checksum* berjalan, kelas ini akan langsung membuat dokumen berupa *file logging* dengan nama yang sama persis dengan nama *file* hasil akusisi. Proses selanjutnya adalah proses *hashing* baik itu MD5 dan SH512. Proses *hashing* meliputi MD5 sumber akusisi dan hasil akusisi, serta SHA512 sumber akusisi dan hasil akusisi. Kemudian seluruh hasil proses *hashing* ini akan dimasukkan ke dalam dokumen *file logging*.

```

class Checksum(QThread):

    def run(self):

        #membuat file logging
        sumberNama = mainWindow_2.sumberNama
        dir = mainWindow_2.dir
        for handler in logging.root.handlers[:]:
            logging.root.removeHandler(handler)
        logging.basicConfig(filemode='w', filename=dir, level=logging.INFO,
                            format='%(levelname)s: %(message)s')

        #Hash sumber cloning
        md5a = hashlib.md5()
        sha512a = hashlib.sha512()
        with open(source, 'rb') as fileSource:
            while True:
                readSource = fileSource.read(block_size)
                if not readSource:
                    break
                md5a.update(readSource)
                sha512a.update(readSource)
        hasilSource = md5a.hexdigest()
        hasilSourceSHA512 = sha512a.hexdigest()

        #Hash hasil cloning
        md5b = hashlib.md5()
        sha512b = hashlib.sha512()
        with open(c, 'rb') as fileCloning:
            while True:
                readCloning = fileCloning.read(block_size)
                if not readCloning:
                    break
                md5b.update(readCloning)
                sha512b.update(readCloning)
        hasilCloning = md5b.hexdigest()
        hasilCloningSHA512 = sha512b.hexdigest()

```

Gambar 4.19 Fungsi *File Logging* dan *Hashing*

4.3 Pengujian

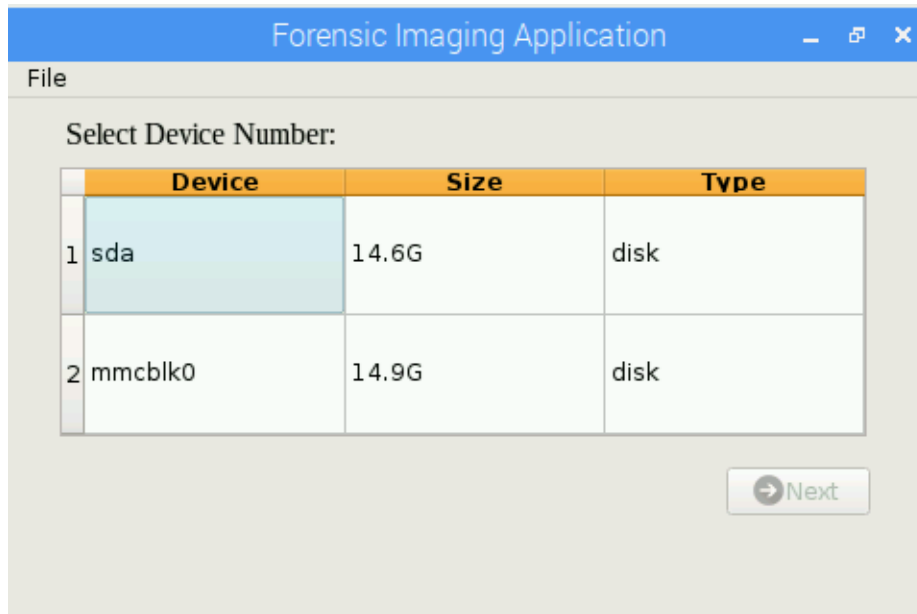
4.3.1 Blackbox

Proses pengujian dengan teknik *Blackbox* dilakukan dengan cara mengamati apa yang terlihat oleh mata dari apa yang ditampilkan oleh aplikasi *Forensic Imaging Application Using Raspberry Pi*. Adapun pengujian yang dilakukan meliputi pengujian fungsi tombol dan *error handling* pada aplikasi.

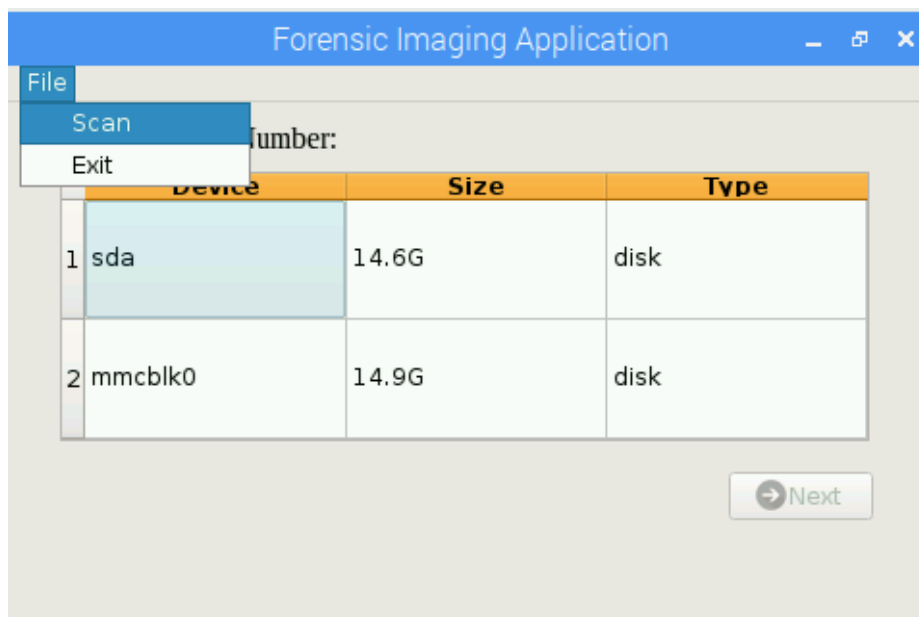
1. Pengujian Tombol

Tahap pengujian tombol pada tampilan halaman 1 aplikasi *Forensic Imaging Application Using Raspberry Pi* meliputi satu tombol dan dua *menu bar*. Yang pertama diuji adalah *scan* pada *menu bar*. Untuk menguji apakah fungsi *scan* bekerja dengan baik atau tidak, disini peneliti menjalankan aplikasi hanya dengan menancapkan satu buah penyimpanan ke dalam perangkat keras Raspberry Pi seperti yang terlihat dalam sajian tabel pada Gambar 4.20. Maka

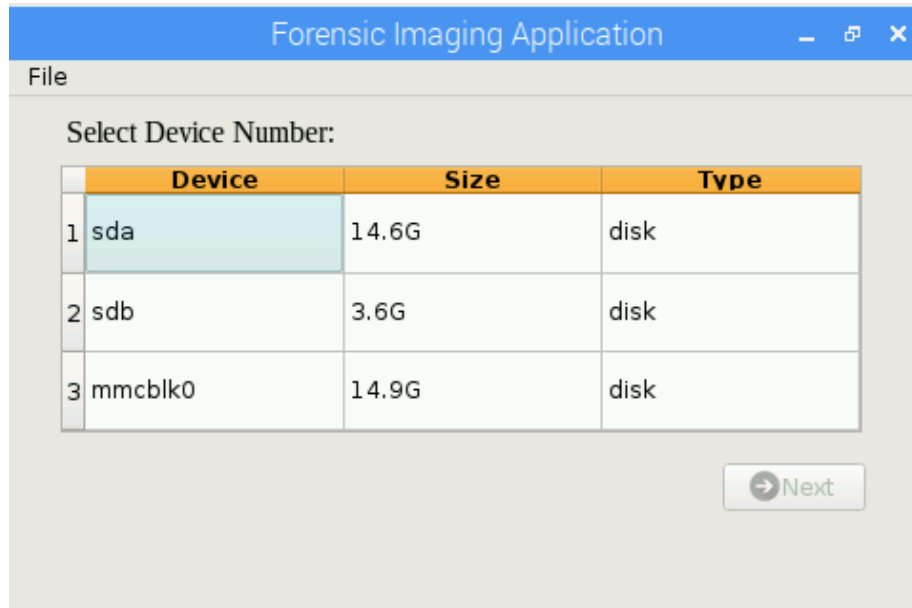
yang ditampilkan oleh tabel adalah hanya satu buah penyimpanan sedangkan *device* dengan nama “mmcblk0” adalah penyimpanan Raspberry Pi itu sendiri. Lalu kemudian peneliti menancapkan kembali satu buah penyimpanan, dan menjalankan fungsi *scan* seperti yang terlihat pada Gambar 4.21 dan setelah dilakukan uji coba fungsi *scan* berjalan dengan baik seperti yang terlihat pada Gambar 4.22.



Gambar 4.20 Pengujian Fungsi *Scan*

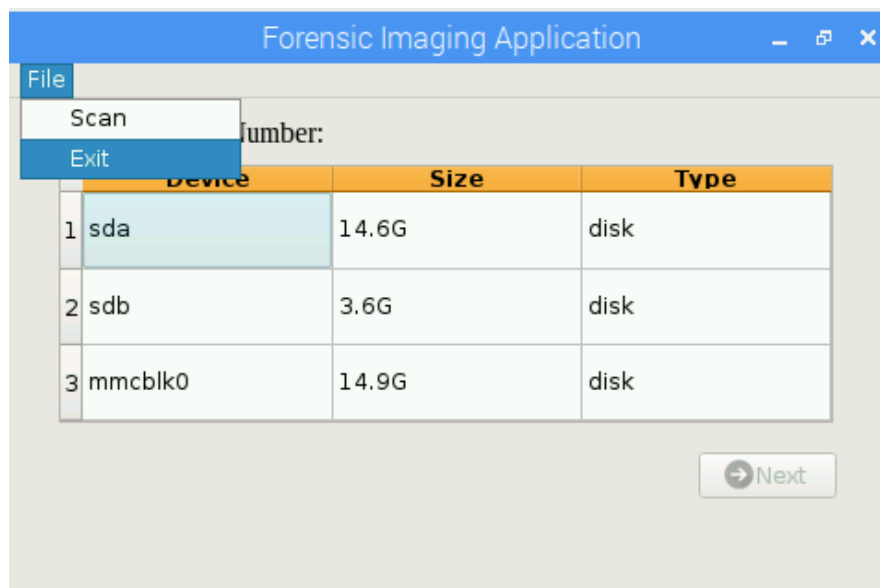


Gambar 4.21 Pengujian Fungsi *Scan*

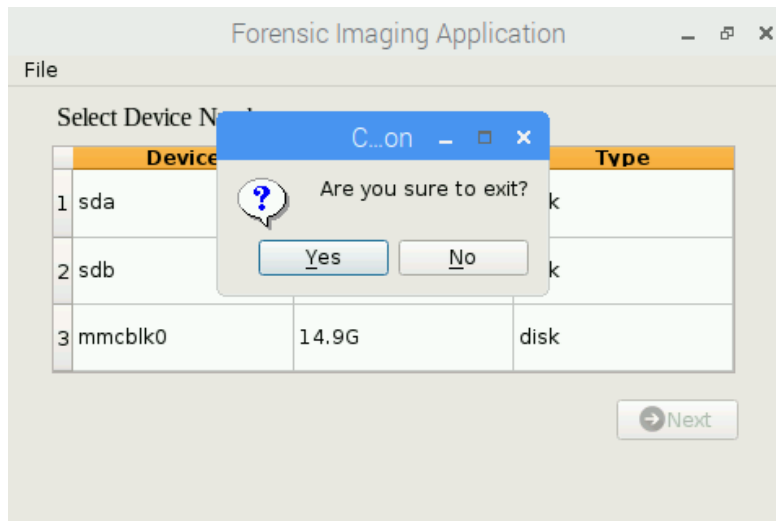


Gambar 4.22 Pengujian Fungsi *Scan*

Selanjutnya fungsi yang peneliti uji coba adalah fungsi *exit*, sebuah fungsi yang terdapat di dalam *menu bar* seperti yang terlihat pada Gambar 4.23 di bawah ini. Untuk mengujinya peneliti mencoba menjalankan fungsi *exit* dan hasilnya fungsi *exit* dapat bekerja dengan baik seperti yang ditunjukkan pada Gambar 4.24. Sedangkan tombol *Next* peneliti asumsikan dapat berjalan dengan baik karena pada proses akusisi diatas proses akusisi tersebut berjalan dengan baik.

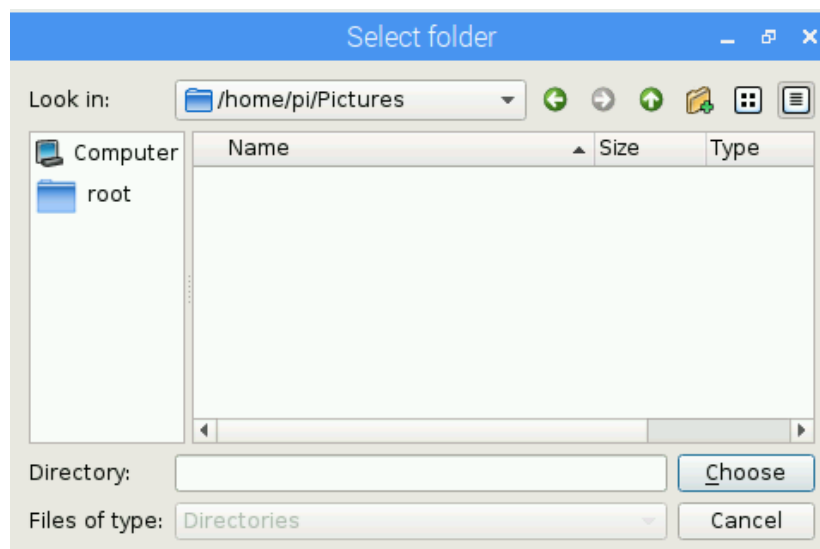


Gambar 4.23 Pengujian Fungsi *Exit*

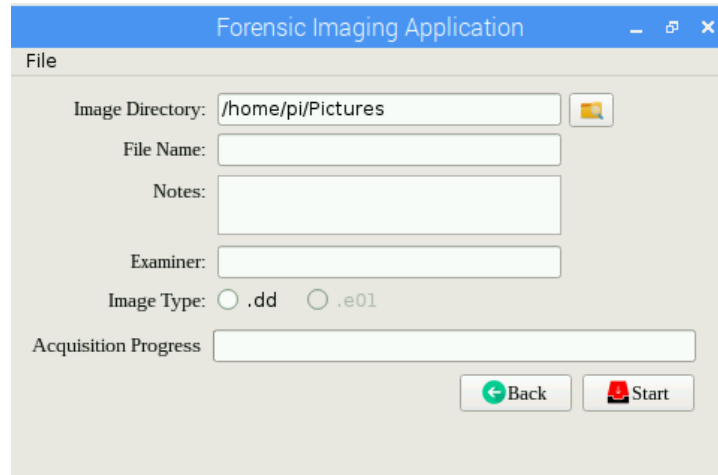


Gambar 4.24 Pengujian Fungsi *Exit*

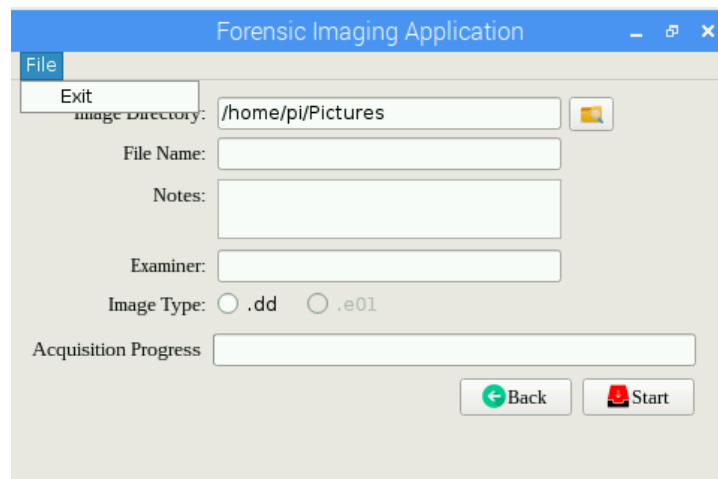
Sedangkan tahap pengujian tombol pada tampilan halaman 2 aplikasi *Forensic Imaging Application Using Raspberry Pi* meliputi dua tombol dan satu *menu bar*. Yang pertama diuji adalah tombol yang berfungsi untuk mengisi kolom *image directory*. Untuk menguji apakah fungsi tombol direktori bekerja dengan baik atau tidak, maka peneliti mencoba untuk mengisi kolom *image directory* dengan “/home/pi/Pictures” seperti yang terlihat pada Gambar 4.25. Maka ketika diuji coba ternyata tombol direktori bekerja dengan baik seperti yang terlihat pada Gambar 4.26.



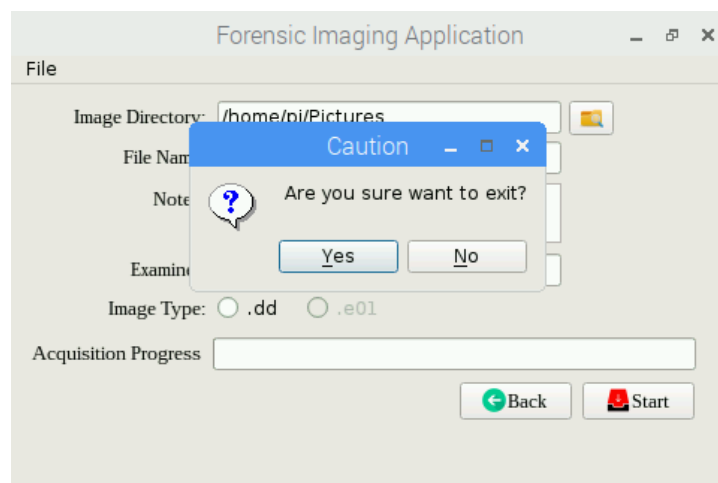
Gambar 4.25 Pengujian Tombol Direktori



Gambar 4.26 Pengujian Tombol Direktori

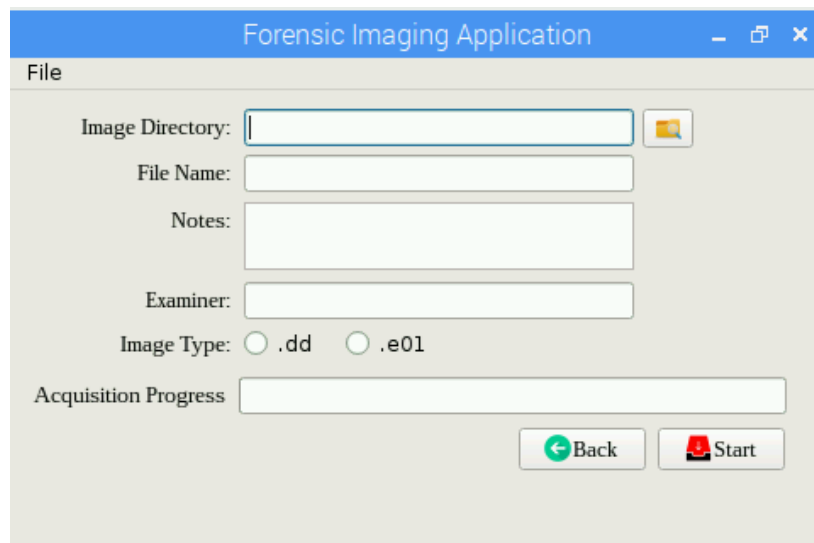


Gambar 4.27 Pengujian Fungsi Exit

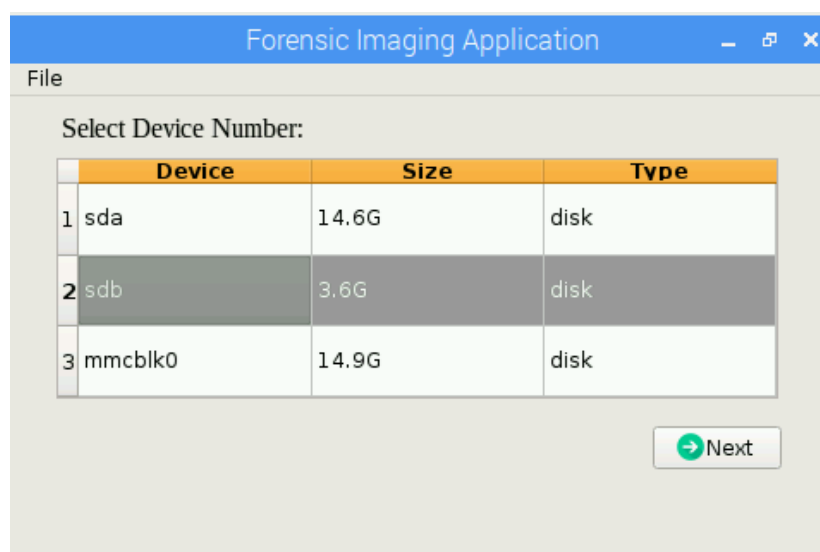


Gambar 4.28 Pengujian Fungsi Exit

Proses pengujian selanjutnya adalah fungsi *exit*, sebuah fungsi yang terdapat di dalam *menu bar* seperti yang terlihat pada Gambar 4.27. Untuk mengujinya peneliti mencoba menjalankan fungsi *exit* dan hasilnya fungsi *exit* dapat bekerja dengan baik seperti yang ditunjukkan pada Gambar 4.27 dan Gambar 4.28. Kemudian pengujian selanjutnya adalah pengujian tombol *Back* yang berfungsi untuk kembali ke halaman sebelumnya. Ketika peneliti menguji coba tombol tersebut, tombol tersebut dapat berkeja dengan baik seperti yang terlihat pada gambar Gambar 4.29 dan Gambar 4.30.



Gambar 4.29 Pengujian Fungsi *Back*

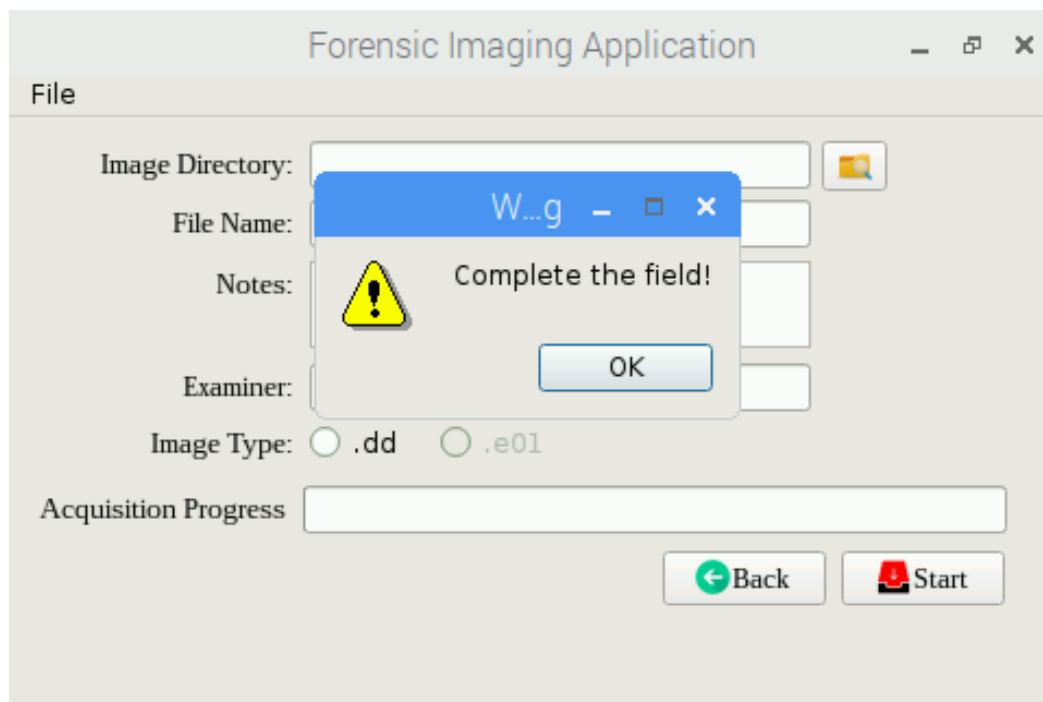


Gambar 4.30 Pengujian Tombol *Back*

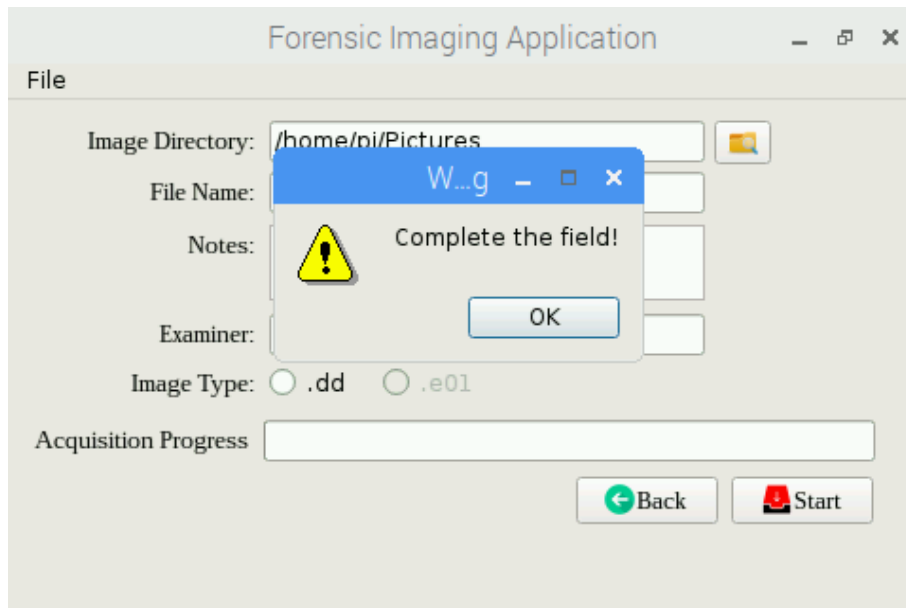
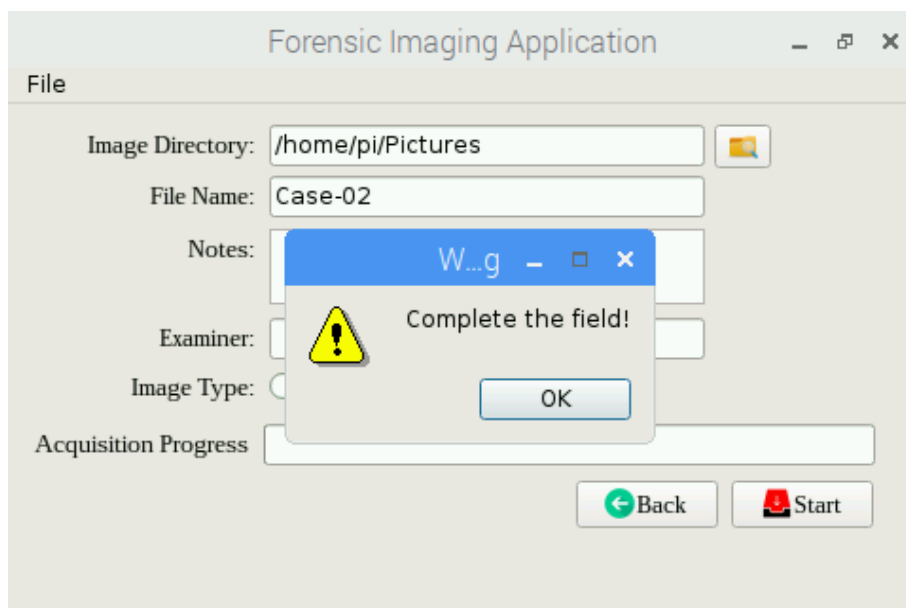
2. Pengujian Error Handling

Tahap pengujian *error handling* pada aplikasi *Forensic Imaging Application Using Raspberry Pi* dilakukan dengan cara dibuat berbagai skema yang dapat menguji aplikasi apakah aplikasi dapat tetap berjalan dengan baik apabila data-data yang aplikasi butuhkan tidak dipenuhi oleh pengguna. Skema pertama yang dijalankan adalah peneliti tidak mengisi semua data masukan yang tersedia melalui kolom-kolom, maka ketika skema itu dijalankan aplikasi tidak akan menjalankan proses akuisisi, melainkan mengeluarkan sebuah peringatan yang memberitahukan bahwa pengguna harus melengkapi data masukan melalui kolom-kolom yang tersedia seperti yang terlihat pada Gambar 4.31.

Skema kedua yang dijalankan adalah peneliti mengisi kolom *image directory*, tetapi sisa kolom yang tersedia tidak peneliti masukan data. Maka ketika skema ini dijalankan yang terjadi adalah aplikasi tetap tidak dapat menjalankan proses akuisisi dan mengeluarkan peringatan untuk melengkapi data masukan melalui kolom-kolom yang tersedia di aplikasi seperti yang terlihat pada Gambar 4.32. Selanjutnya peneliti menjalankan skema 3 yaitu peneliti mengisi kolom *filename* dan kolom *image directory*. Maka ketika proses skema ini dijalankan yang terjadi adalah aplikasi tidak dapat melakukan proses akuisisi dan mengeluarkan peringatan agar melengkapi data masukan yang dibutuhkan seperti yang terlihat pada Gambar 4.33.

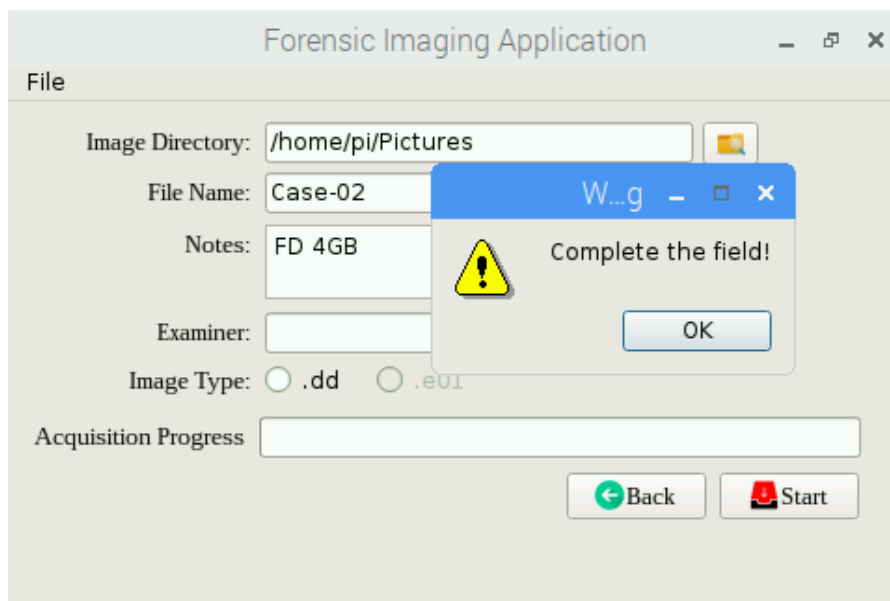


Gambar 4.31 *Error Handling* Skema 1

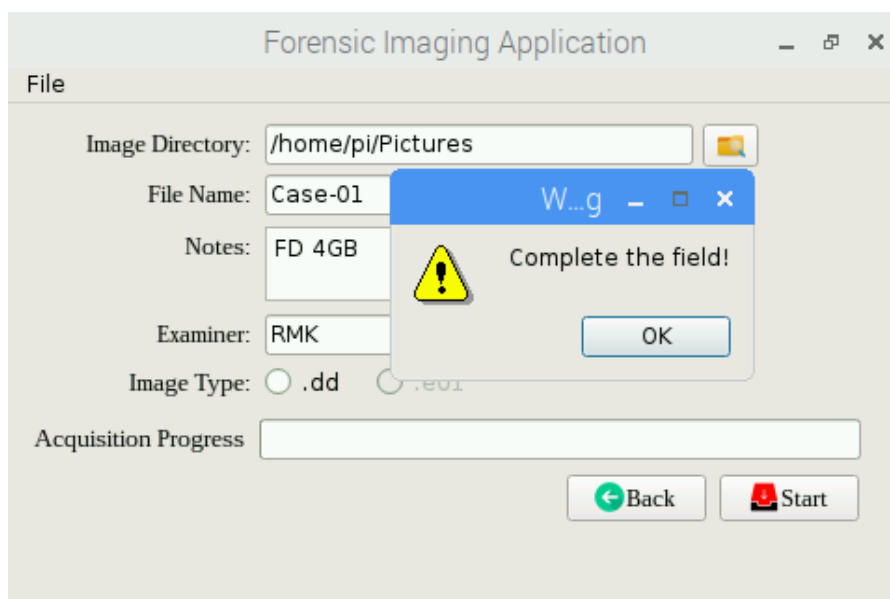
Gambar 4.32 *Error Handling Skema 2*Gambar 4.33 *Error Handling Skema 3*

Selanjutnya peneliti melanjutkan ke skema empat dimana peneliti mengisi tiga kolom awal, yaitu kolom *image directory*, *filename*, dan *notes*. Pada saat skema ini dijalankan maka yang terjadi adalah aplikasi tetap tidak dapat melanjutkan ke proses akusisi seperti yang terlihat pada Gambar 4.34. Skema selanjutnya adalah peneliti mengisi empat kolom awal, yaitu kolom *image directory*, *filename*, *notes*, dan *examiner*. Maka ketika skema ini dijalankan yang terjadi adalah sama seperti yang terjadi pada skema sebelumnya, yakni aplikasi tidak dapat melakukan

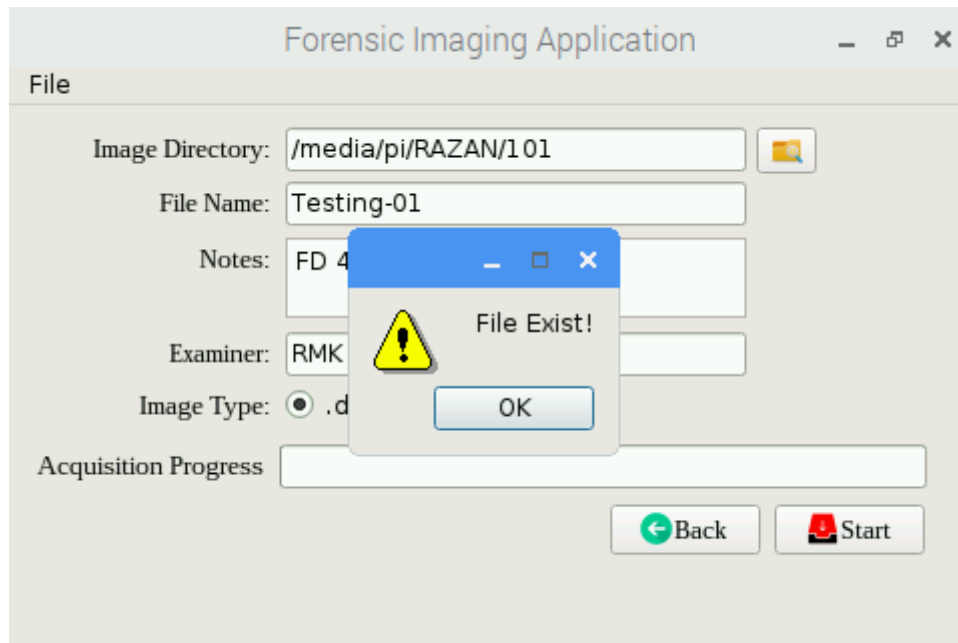
proses akuisisi sampai semua kolom-kolom yang tersedia dilengkapi seperti yang terlihat pada Gambar 4.35. Skema yang terakhir adalah peneliti mengisi semua kolom masukan dengan benar, tetapi pada kolom *filename*, peneliti memasukan nama *file* tersebut dengan nama yang memang sudah ada di dalam *folder* tempat lokasi penyimpanan *file* hasil akuisisi. Maka ketika skema ini dijalankan yang terjadi adalah aplikasi akan mengeluarkan peringatan bahwa nama *file* tersebut sudah ada seperti yang terlihat pada Gambar 4.36.



Gambar 4.34 Error Handling Skema 4



Gambar 4.35 Error Handling Skema 5



Gambar 4.36 *Error Handling Skema 6*

4.3.2 Pengujian Performa Aplikasi

Tahap pengujian performa aplikasi adalah sebuah tahap yang dilakukan peneliti untuk mengetahui sejauh mana performa aplikasi yang peneliti buat dapat bekerja. Selain itu tahap pengujian performa aplikasi juga berguna untuk mengetahui apakah dengan menggunakan *flash drive* dengan merek yang berbeda tetapi ukurannya sama mempengaruhi kecepatan proses akusisi. Untuk menguji serta mengetahui sejauh mana performa aplikasi *Forensic Imaging Application Using Raspberry Pi* dapat bekerja, peneliti menguji aplikasi *forensic imaging* ini dengan cara melakukan enam kali percobaan proses akusisi, yang kemudian nantinya dari setiap proses akusisi yang telah dilakukan akan terlihat sejauh mana performa aplikasi dapat bekerja, baik dari sisi waktu proses akusisi maupun dari sisi kecepatan transfer datanya.

Proses akusisi sendiri menggunakan enam buah *flash drive* yang memiliki kapasitas penyimpanan dan merek yang berbeda, yakni Kingston 4GB, ICRC 4GB, Sandisk 8GB, Toshiba 8GB, Sandisk 16GB, dan Toshiba 16GB. Keenam *flash drive* tersebut nantinya akan diakusisi menggunakan aplikasi *Forensic Imaging Application Using Raspberry Pi* dan *file* hasil akusisi akan disimpan di *flash drive* Sandisk 64GB. Setelah keseluruhan proses percobaan akusisi selesai dilakukan, langkah selanjutnya adalah peneliti mencari tahu tentang kecepatan transfer data dari setiap proses akusisi yang telah dijalankan, untuk kemudian nantinya akan dicari nilai rata-rata kecepatan transfer datanya. Untuk detail rincian pengujian performa aplikasi *Forensic Imaging Application Using Raspberry Pi* dapat dilihat pada Tabel 4.1

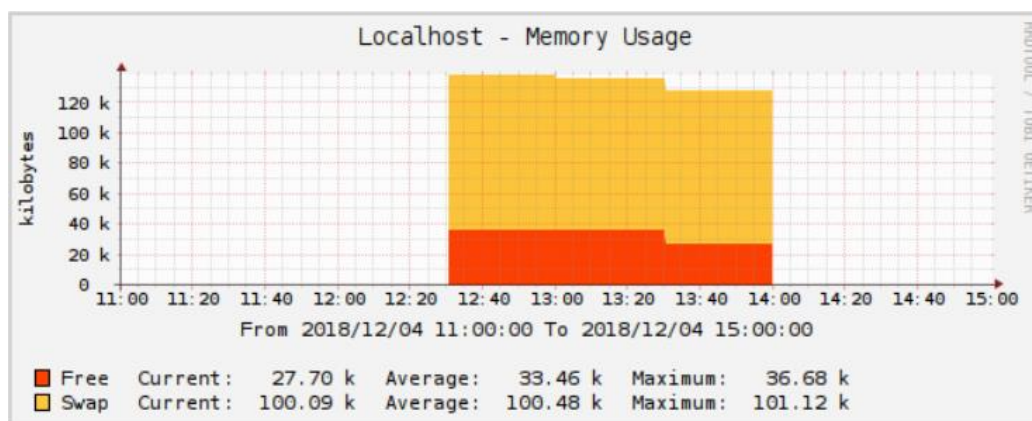
Tabel 4.1 Pengujian Performa Aplikasi

Nomor Percobaan	Input	Output	Durasi Proses Akusisi	Kecepatan Transfer Data
1	Kingston 4GB USB 2.0	Sandisk 64GB USB 2.0	35 Menit	1,75 MB/s
2	ICRC 4GB USB 2.0	Sandisk 64GB USB 2.0	34 Menit	1,91 MB/s
3	Sandisk 8GB USB 2.0	Sandisk 64GB USB 2.0	73 Menit	1,69 MB/s
4	Toshiba 8GB USB 2.0	Sandisk 64GB USB 2.0	64 Menit	1,92 MB/s
5	Sandisk 16GB USB 2.0	Sandisk 64GB USB 2.0	129 Menit	1,93 MB/s
6	Toshiba 16GB USB 2.0	Sandisk 64GB USB 2.0	131 Menit	1,88 MB/s
Kecepatan Rata-rata Transfer Data				1,85 MB/s

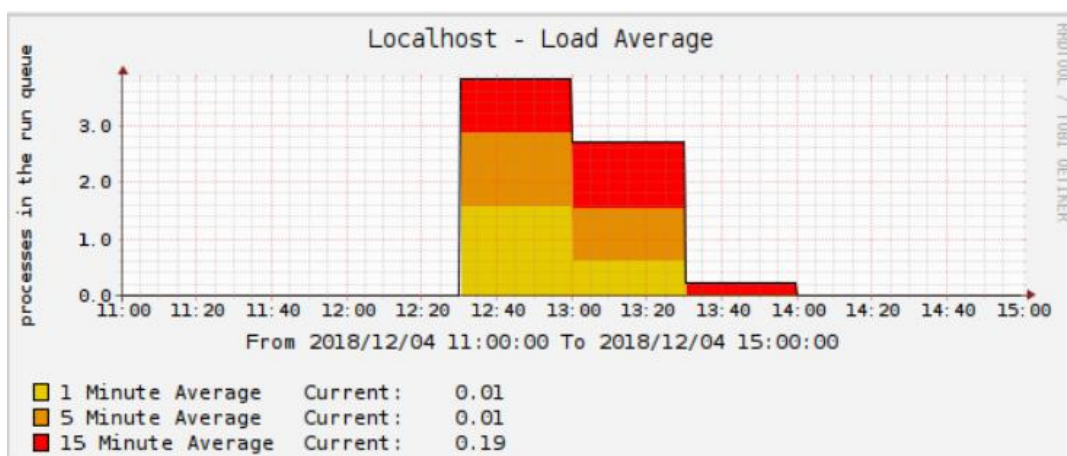
Pada proses pengujian performa aplikasi *Forensic Imaging Application Using Raspberry Pi* yang dilakukan, peneliti mengakusisi tiga buah *flash drive* yang memiliki kapasitas penyimpanan dan merek yang berbeda, diantaranya Kingston 4GB, Sandisk 8GB, dan Sandisk 16GB. Kemudian setelah mendapatkan data waktu proses akusisi dan kecepatan transfer data akusisi, selanjutnya peneliti akan membandingkan proses waktu akusisi dan kecepatan transfer data akusisi dengan menggunakan *flash drive* yang memiliki ukuran yang sama tetapi dengan merek yang berbeda, *flash drive* yang peneliti gunakan adalah ICRC 4GB, Toshiba 8GB, dan Toshiba 16GB. Dengan proses uji coba tersebut diharapkan peneliti dapat mengetahui adakah pengaruh proses waktu akusisi dan kecepatan transfer data akusisi dengan *flash drive* yang memiliki merek berbeda tetapi ukuran kapasitas penyimpanan sama.

Pada percobaan pertama peneliti mengakusisi *flash drive* Kingston 4GB, proses akusisi berjalan selama 35 menit dengan kecepatan transfer data akusisi adalah sebesar 1,75 MB/s. Selama proses akusisi pada percobaan pertama ini berlangsung, peneliti memonitoring penggunaan sumber daya perangkat Raspberry Pi menggunakan perangkat lunak Cacti. Dari hasil monitoring yang dilakukan, didapatkan bahwa penggunaan memori perangkat Raspberry Pi hampir seluruhnya terpakai oleh program yang berjalan dan menyisakan sisa kapasitas

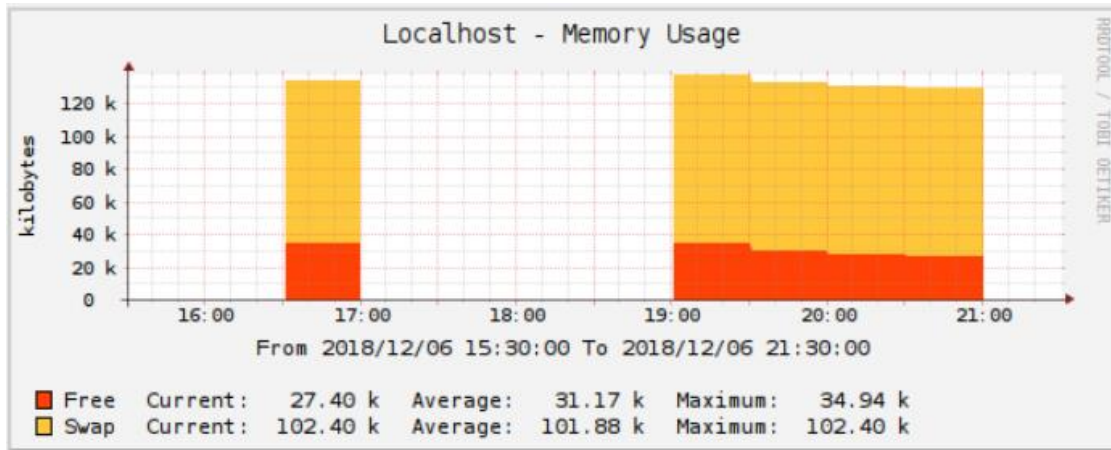
memori sebesar 33,46KB dari total ukuran memori sebesar 1GB seperti yang terlihat pada Gambar 4.37. Sedangkan dari segi penggunaan CPU nya, perangkat Raspberry Pi memiliki nilai rata-rata beban kerja dalam 1 menit sebesar 0,01, 5 menit sebesar 0,01 dan 15 menit sebesar 0,19 seperti yang terlihat pada Gambar 4.38. Kemudian hasil dari akusisi pada percobaan pertama akan dibandingkan dengan percobaan akusisi pada nomor dua seperti yang terlihat pada Tabel 4.1. Pada proses percobaan nomor dua, peneliti mengakusisi *flash drive* ICRC 4GB, proses akusisi ini berlangsung selama 34 menit, dengan kecepatan transfer data akusisi sebesar 1,91 MB/s. Selama proses akusisi pada percobaan kedua ini, memori yang tersisa adalah sebesar 31,17KB dari total kapasitas memori sebesar 1GB, sedangkan beban kerja CPU pada proses akusisi kedua memiliki nilai rata-rata dalam 1 menit sebesar 0,02, sedangkan dalam 5 menit memiliki nilai rata-rata 0,03 dan dalam 15 menit memiliki nilai rata-rata sebesar 0,12 seperti yang terlihat pada Gambar 4.39 dan Gambar 4.40.



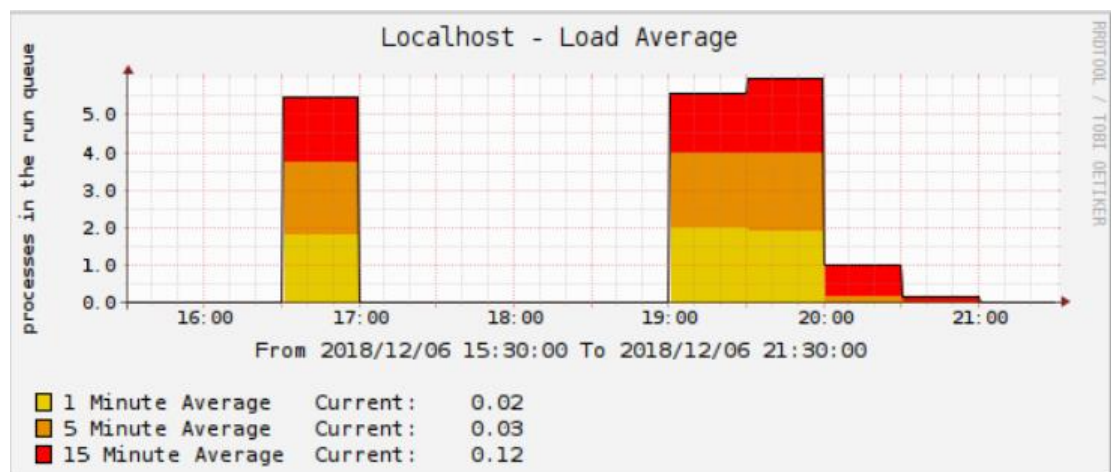
Gambar 4.37 Penggunaan Memori Perangkat Forensik Pada Percobaan 1



Gambar 4.38 Penggunaan CPU Perangkat Forensik Pada Percobaan 1



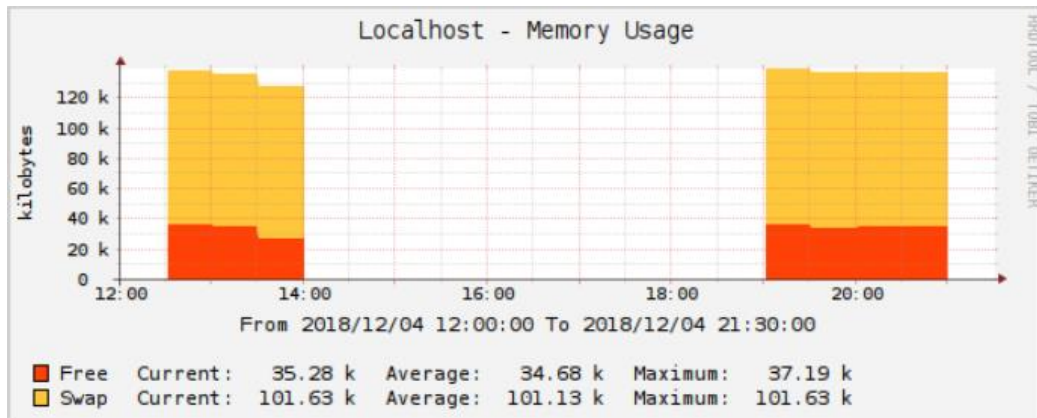
Gambar 4.39 Penggunaan Memori Perangkat Forensik Pada Percobaan 2



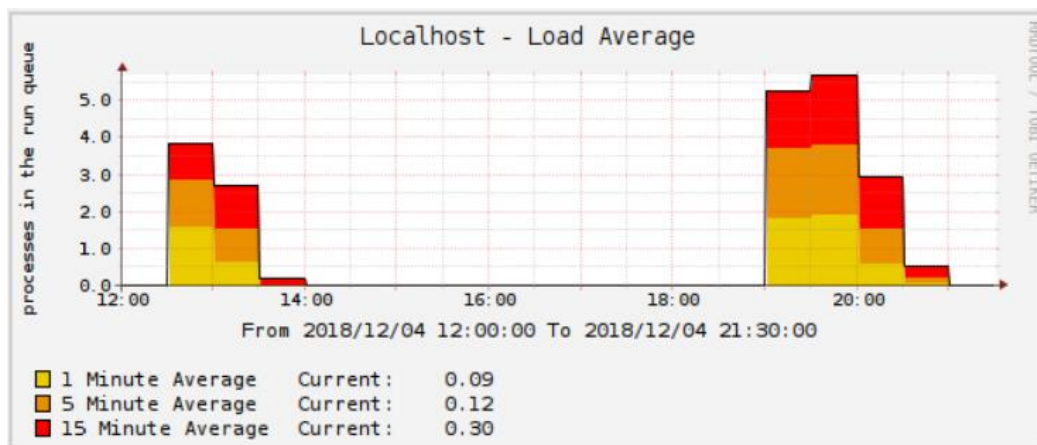
Gambar 4.40 Penggunaan CPU Perangkat Forensik Pada Percobaan 2

Pada percobaan proses akusisi ketiga peneliti mengakusisi *flash drive* Sandisk 8GB, proses akusisi ini berjalan selama 73 menit dengan kecepatan transfer data akusisi adalah sebesar 1,69 MB/s. Selanjutnya proses percobaan akusisi yang ketiga ini akan dibandingkan dengan percobaan pada nomor empat seperti yang terlihat pada Tabel 4.1 Pengujian Performa Aplikasi. Pada proses percobaan nomor empat peneliti mengakusisi *flash drive* Toshiba 8GB, proses percobaan akusisi ini memakan waktu hingga 64 menit, dengan kecepatan transfer data akusisi sebesar 1,92 MB/s. Pada proses percobaan akusisi nomor tiga kapasitas memori yang tersisa adalah sebesar 34,68KB dari total kapasitas 1GB, sedangkan pada proses akusisi nomor empat kapasitas memori yang tersisa adalah sebesar 31,17KB seperti pada Gambar 4.41 dan Gambar 4.43. Kemudian dari sisi penggunaan CPU pada percobaan nomor tiga memiliki nilai rata-rata dalam 1 menit sebesar 0,09, selanjutnya dalam 5 menit sebesar 0,12, dan dalam 15 menit sebesar 0,30. Selanjutnya penggunaan CPU pada percobaan keempat memiliki nilai rata-

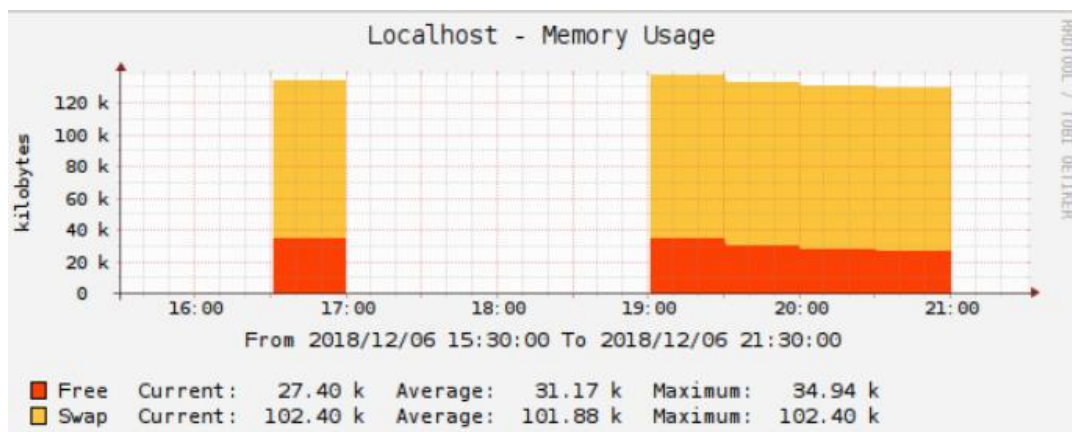
rata dalam 1 menit sebesar 0,02, sedangkan dalam 5 menit sebesar 0,03, dan dalam 15 menit sebesar 0,12 seperti pada Gambar 4.42 dan Gambar 4.44.



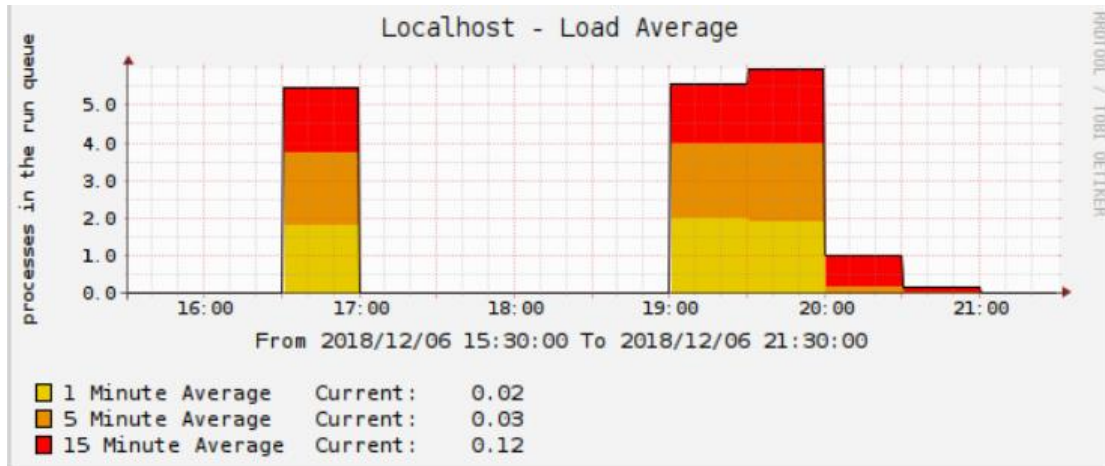
Gambar 4.41 Penggunaan Memori Perangkat Forensik Pada Percobaan 3



Gambar 4.42 Penggunaan CPU Perangkat Forensik Pada Percobaan 3

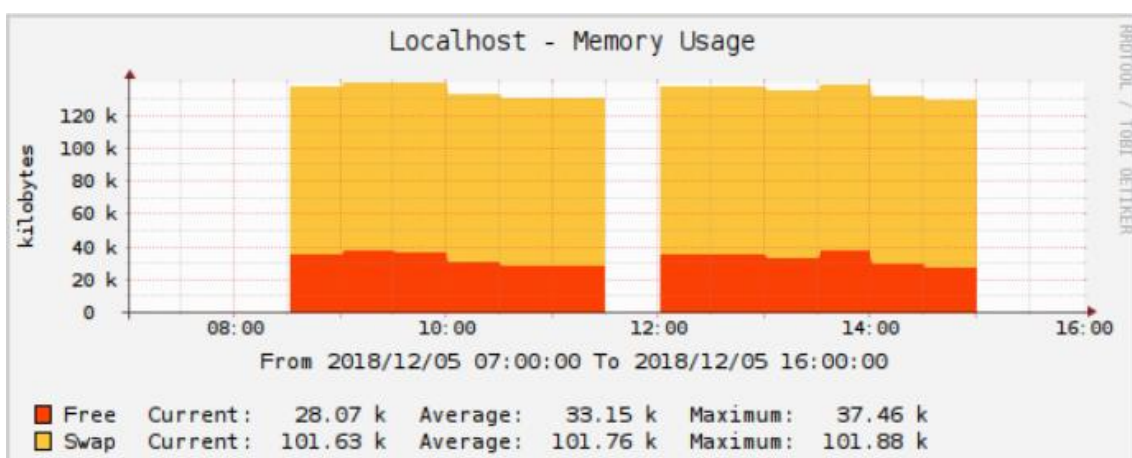


Gambar 4.43 Penggunaan Memori Perangkat Forensik Pada Percobaan 4

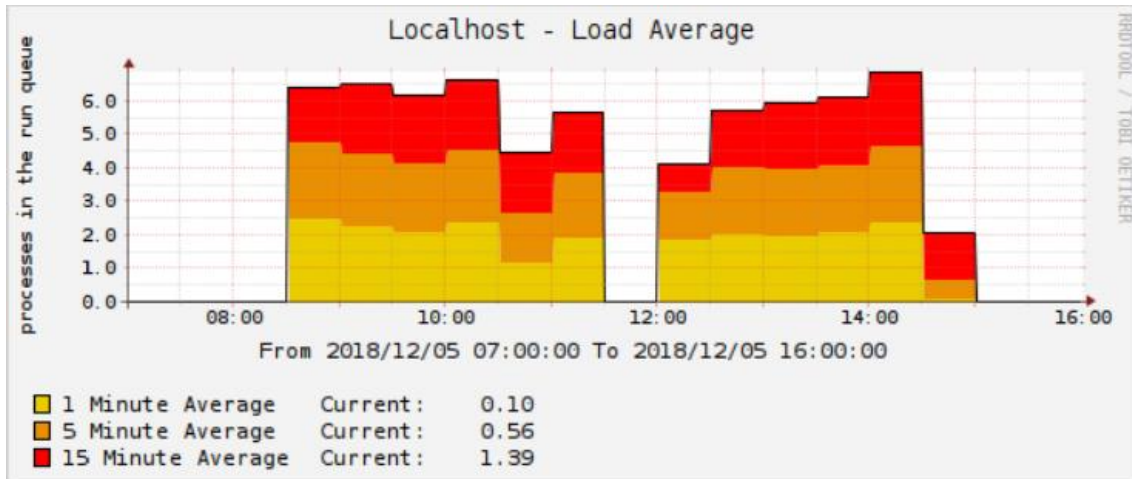


Gambar 4.44 Penggunaan CPU Perangkat Forensik Pada Percobaan 4

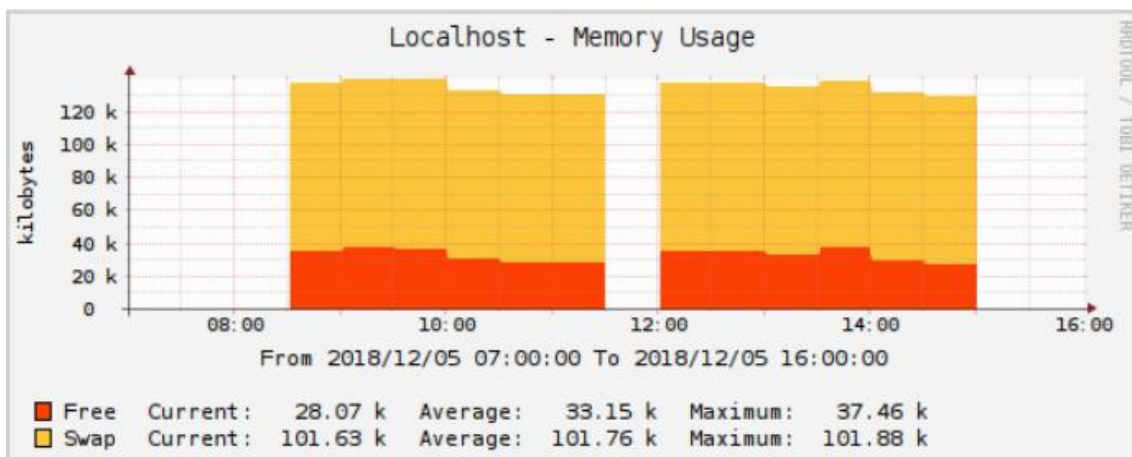
Selanjutnya peneliti mengakuisisi *flash drive* Sandisk 16GB, pada proses percobaan akuisisi ini memakan waktu hingga 129 menit, dengan kecepatan transfer data akuisisi sebesar 1,93 MB/s. Dari hasil monitoring pada percobaan kelima yang dilakukan, didapatkan bahwa penggunaan memori perangkat hampir seluruhnya terpakai dan menyisakan memori sebesar 33,15KB dari total ukuran memori sebesar 1GB seperti yang terlihat pada Gambar 4.45. Sedangkan dari segi penggunaan CPU nya, perangkat Raspberry Pi memiliki nilai rata-rata beban kerja dalam 1 menit sebesar 0,10, 5 menit sebesar 0,56 dan 15 menit sebesar 1,39 seperti yang terlihat pada Gambar 4.46. Kemudian peneliti membandingkannya dengan *flash drive* Toshiba 16GB, pada proses akuisisi ini memerlukan waktu selama 131 menit dengan kecepatan transfer data akuisisi sebesar 1,88 MB/s. Sedangkan untuk melihat penggunaan sumber daya selama proses akuisisi keenam dapat dilihat pada Gambar 4.47 dan Gambar 4.48.



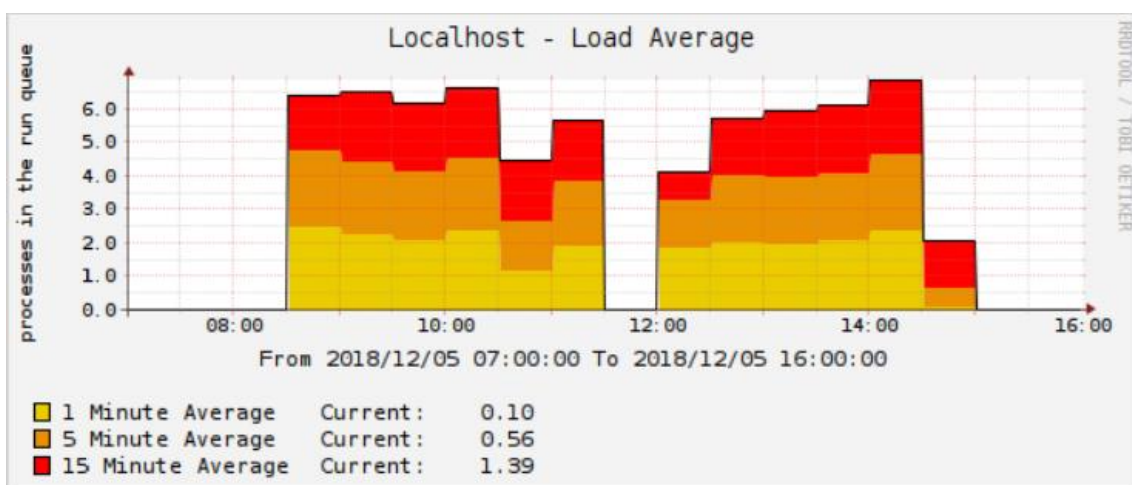
Gambar 4.45 Penggunaan Memori Perangkat Forensik Pada Percobaan 5



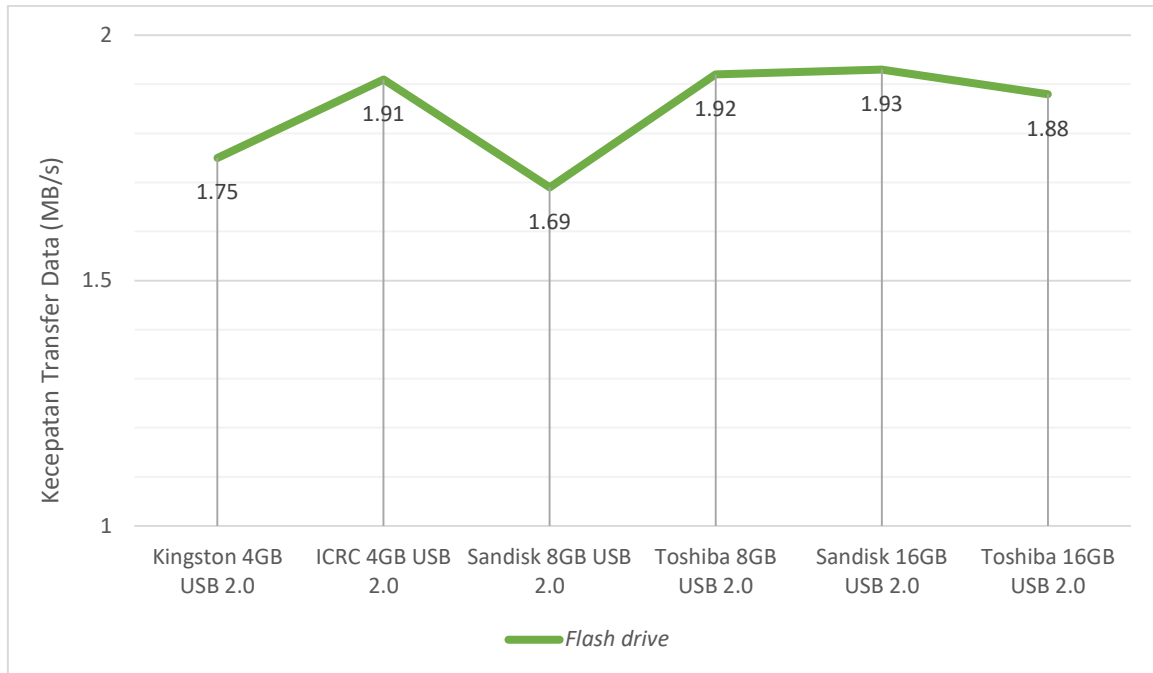
Gambar 4.46 Penggunaan CPU Perangkat Forensik Pada Percobaan 5



Gambar 4.47 Penggunaan Memori Perangkat Forensik Pada Percobaan 6



Gambar 4.48 Penggunaan CPU Perangkat Forensik Pada Percobaan 6



Gambar 4.49 Grafik Kecepatan Transfer Data

Setelah melalui serangkaian uji coba proses akusisi, dapat disimpulkan bahwa penggunaan *flash drive* dengan ukuran kapasitas penyimpanan yang sama tetapi merek berbeda tidak memiliki perbedaan yang signifikan baik dari segi sisi waktu proses akusisi maupun dari sisi kecepatan transfer data akusisi, hal ini dapat terjadi dikarenakan perangkat keras Raspberry Pi kehabisan sumber daya atau *resources*, sehingga proses transfer data selama uji coba akusisi tidak maksimal seperti yang terlihat pada gambar hasil monitoring di atas. Seperti pada percobaan nomor satu dan dua, yang memiliki waktu akusisi selama 35 menit, dan 34 menit dengan kecepatan transfer data akusisi sebesar 1,75 MB/s dan 1,91 MB/s. Sedangkan pada percobaan nomor tiga dan empat memiliki nilai kecepatan transfer data masing-masing sebesar 1,69 MB/s dan 1,92 MB/s, dengan durasi waktu proses akusisi masing-masing adalah 73 menit dan 64 menit. Selanjutnya pada proses percobaan akusisi nomor lima dan enam memakan waktu akusisi selama 129 menit dan 131 menit, dengan kecepatan transfer data akusisi masing-masing adalah sebesar 1,93 MB/s dan 1,88 MB/s. Secara teori USB 2.0 memiliki nilai kecepatan transfer data mencapai 60 MB/s, namun kecepatan rata-rata transfer data yang diperoleh selama proses akusisi yang dijalankan adalah sebesar 1,85 MB/s. Untuk melihat bagaimana grafik kecepatan transfer data dari enam percobaan proses akusisi yang telah dilakukan dapat dilihat pada Gambar 4.49 Grafik Kecepatan Transfer Data.

4.3.3 Perhitungan Skala Likert

Pada tahap ini dilakukan perhitungan menggunakan skala likert terhadap kuesioner yang telah peneliti bagikan kepada responden melalui *google form*. Peneliti membutuhkan 50 orang responden untuk mengisi setiap pertanyaan yang terdapat pada kuesioner, angka 50 ini peneliti dapatkan dari rumus persamaan Slovin yang dapat dilihat pada persamaan (4.1). Dengan asumsi populasi yang peneliti gunakan adalah sebesar 100 orang dan batas toleransi kesalahan yang peneliti gunakan adalah sebesar 10% maka didapatkan jumlah sampel sebanyak 50 orang. Peneliti menyebarkan kuesioner ke berbagai pihak yang mengambil bidang minat Jaringan dan Keamanan Komputer (JKK) yang berada di lingkungan kampus Teknik Informatika UII. Mahasiswa yang peneliti pilih untuk dijadikan responden adalah mahasiswa S-1 yang mengambil kelas Komputer Forensik, dan Mahasiswa S-2 Forensik Teknik Informatika UII. Pada praktiknya dilapangan, peneliti berhasil mendapatkan 56 responden yang bersedia mengisi kuesioner yang peneliti bagikan.

Langkah awal yang dilakukan untuk mendapatkan nilai skala likert adalah mencari total skor dari setiap butir pertanyaan yang ada pada kuesioner menggunakan persamaan (3.1). Setelah mendapatkan total skor, langkah selanjutnya adalah mencari nilai rata-rata dari setiap butir pertanyaan menggunakan persamaan (3.2). Detail perhitungan kuesioner dapat dilihat pada Tabel 4.2, sedangkan untuk melihat detail hasil kuesioner secara rinci dapat dilihat pada lembar lampiran.

$$n = \frac{N}{(1 + N \cdot (e)^2)} \quad (4.1)$$

Tabel 4.2 Perhitungan Skala Likert

No	Aspek	Nomor Pertanyaan	Nilai Rata-rata Nomor Pertanyaan	Hasil Nilai Rata-rata Aspek	Klasifikasi
1	Manfaat	1,2,3,4,5,6 (Pertanyaan pada Tabel 3.5)	(4,25 + 3,80 + 4,50 + 3,80 + 4,25 + 4,03)/6	4,10	Layak
2	Tampilan	1,2,3,4 (Pertanyaan pada Tabel 3.5)	(3,70 + 4,08 + 4,07 + 3,90 +)/4	3,93	Layak

No	Aspek	Nomor Pertanyaan	Nilai Rata-rata Nomor Pertanyaan	Hasil Nilai Rata-rata Aspek	Klasifikasi
3	Fungsionalitas	1,2,3,4 (Pertanyaan pada Tabel 3.5)	$(4,23 + 4,26 + 4,08 + 3,85)/4$	4,10	Layak
Total Rata-rata Keseluruhan				4,04	Layak

Perhitungan kuesioner skala likert terbagi menjadi tiga aspek pertanyaan, yakni aspek manfaat, tampilan, dan fungsionalitas seperti yang tertera pada Tabel 4.2. Aspek manfaat yang terdiri atas lima butir pertanyaan memiliki nilai rata-rata 4,10 sehingga memiliki klasifikasi layak, sedangkan aspek tampilan yang terdiri atas empat butir pertanyaan memiliki nilai rata-rata 3,93 sehingga termasuk ke dalam klasifikasi layak. Selanjutnya aspek fungsionalitas yang terdiri atas empat butir pertanyaan memiliki nilai 4,10 dan termasuk ke dalam klasifikasi layak. Kemudian untuk nilai rata-rata keseluruhan kuesioner adalah 4,04, maka dengan nilai tersebut maka aplikasi yang peneliti buat dianggap layak baik dari aspek manfaat, tampilan, maupun fungsionalitas.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah proses penelitian yang dilakukan oleh peneliti dan mengimplementasikannya, kesimpulan yang dapat diambil dari kegiatan tugas akhir dengan judul *Forensic Imaging Application Using Raspberry Pi* adalah:

- a. Berhasil membuat perangkat *forensic imaging* dengan menggunakan *single board computer* Raspberry Pi.
- b. Aplikasi berbasis *Graphical User Interface* (GUI) berhasil mengoperasikan perangkat *forensic imaging* dengan baik.
- c. Penggunaan *flash drive* dengan ukuran kapasitas penyimpanan yang sama namun merek berbeda tidak mempengaruhi kecepatan transfer data akusisi dan nilai kecepatan rata-rata transfer data akusisi yang diperoleh selama proses akusisi adalah sebesar 1,85 MB/s.

5.2 Saran

Pada *Forensic Imaging Application Using Raspberry Pi* masih terdapat beberapa kelemahan serta kekurangan yang dalam waktu yang akan datang masih dapat dikembangkan agar aplikasi ini dapat lebih baik lagi. Berikut beberapa hal yang masih dapat dikembangkan dari aplikasi *Forensic Imaging Application Using Raspberry Pi* yaitu:

- a. Untuk pengembangan aplikasi ini kedepannya diharapkan mampu menghasilkan *file* format yang lain tidak hanya format .dd.
- b. Diharapkan kedepannya aplikasi ini mampu meningkatkan kecepatan transfer datanya sehingga proses akusisi yang berjalan dapat lebih cepat.
- c. Perangkat *forensic imaging* ini diharapkan mampu untuk dikembangkan tidak hanya dimanfaatkan untuk melakukan akusisi terhadap *hardisk* maupun *flashdrive*, tetapi dikembangkan sehingga mampu melakukan akusisi terhadap telepon genggam.

DAFTAR PUSTAKA

- Al-Azhar, M. N. (2012). *Digital Forensic: panduan praktis investigasi komputer*. Jakarta: Salemba Empat.
- Baharuddin, A., Ruskam, A., & Yacob, A. (2015). Prinsip Asas Sains Forensik dari Perspektif Islam: Suatu Sorotan Literatur. *Sains Humanika*, 2(2008), 7–15. Retrieved from <http://www.sainshumanika.utm.my/index.php/sainshumanika/article/view/556>
- Bobsusanto. (2014). Komputer Forensik: Pengertian Dan Tujuan Lengkap. Retrieved from <http://www.sepengetahuan.com/2014/11/komputer-forensik-pengertian-dan-tujuan.html>
- Budiaji, W. (2013). LIKERT (The Measurement Scale and The Number of Responses in Likert Scale). *Jurnal Ilmu Pertanian Dan Perikanan*, 2(2), 127–133.
- Conway, M. (1995). Python. *A GUI Development Tool*, 2(2), 23–28. Retrieved from http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=pubmed&cmd=Retrieve&dopt=AbstractPlus&list_uids=205350.205354%5Cnpapers2://publication/uuid/C52A790B-91F5-4E6A-81FF-3D82F1307804
- Garfinkel, S. L. (2010). Digital forensics research: The next 10 years. *Digital Investigation*, 7(SUPPL.), S64–S73. <https://doi.org/10.1016/j.diin.2010.05.009>
- Garfinkel, S. L., Malan, D. J., Dubec, K.-A., Stevens, C. C., & Pham, C. (2006). Advanced Forensic Format: An Open, Extensible Format for Disk Imaging. *Advances in Digital Forensics II: FIP International Conference on Digital Forensics*, 222, 17–31. https://doi.org/10.1007/0-387-36891-4_2
- Hutabarat, D. (2018). Polri: Indonesia Tertinggi Kedua Kejahatan Siber di Dunia. Retrieved from https://kominfo.go.id/content/detail/13487/polri-indonesia-tertinggi-kedua-kejahatan-siber-di-dunia/0/sorotan_media
- Kominfo. (2016). UU ITE No. 19 Tahun 2016. Retrieved from [https://web.kominfo.go.id/sites/default/files/users/4761/UU 19 Tahun 2016.pdf](https://web.kominfo.go.id/sites/default/files/users/4761/UU%2019%20Tahun%202016.pdf)
- Kompas. (2012). Indonesia Masuk 10 Besar Penyumbang “Cyber Crime” Terbanyak. Retrieved from <https://tekno.kompas.com/read/2012/05/16/09403718/Indonesia.Masuk.10.Besar.Penyumbang.Cyber.Crime.Terbanyak>
- Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 22 140, 55. <https://doi.org/2731047>

- Mardalis. (2002). *Metode penelitian : suatu pendekatan proposal*. Jakarta: PT. Bumi Aksara.
- Movanita, A. N. K. (2017). Ini Hasil Kerja Polri Perangi Kejahatan Siber Sepanjang 2017. Retrieved from <https://nasional.kompas.com/read/2017/12/29/17233911/ini-hasil-kerja-polri-perangi-kejahatan-siber-sepanjang-2017>
- Olson, E., & Shashidhar, N. (2016). Low Budget Forensic Drive Imaging Using Arm-Based Single Board Computers. *Journal of Digital Forensics, Security and Law*, 11(1), 53–76. Retrieved from <http://ojs.jdfsl.org.liverpool.idm.oclc.org/index.php/jdfsl/article/viewFile/394/268>
- Philbin, C. A. (2014). *Adventure in Raspberry Pi*. John Wiley & Sons. Retrieved from 9781118751251
- Prayudi, Y., & Afrianto, D. S. (2007). Antisipasi Cybercrime Menggunakan Teknik Komputer Forensik. *Snati*, 2007(Snati), 1–4. Retrieved from <http://journal.uui.ac.id/index.php/Snati/article/view/1634>
- Premierfarnell. (2018). Raspberry Pi Professional Expert Day shows increasing interest in use for the Raspberry Pi for professional applications. Retrieved from <https://www.premierfarnell.com/content/raspberry-pi-professional-expert-day-shows-increasing-interest-use-raspberry-pi-professional>
- Rouse, M. (2017). Forensic Image. Retrieved from <https://whatis.techtarget.com/definition/forensic-image>
- Sarwono, J. (2006). *Metode penelitian kuantitatif & kualitatif*. Yogyakarta: Graha Ilmu.
- Taufiqurrohman, I., Widiyasono, N., & Mubarak, H. (2017). Pemanfaatan Raspberry Pi untuk Hacking dan Forensic dengan metode NIST (National Institute of Standards and Technology), 3, 231–244.
- van Rossum, G. (1995). Python {Tutorial}, (CS-R 9526), 1–65. Retrieved from <http://oai.cwi.nl/oai/asset/5007/05007D.pdf>
- Weik, M. H. (2001). operating system. In *Computer Science and Communications Dictionary* (p. 1150). Boston, MA: Springer US. https://doi.org/10.1007/1-4020-0613-6_12849

LAMPIRAN