Studi Komparasi Metode Disk Overwrite dan Factory Reset Sebagai Teknik Anti Forensik di Perangkat Android

Beni Ike Hendra Kuswara
20917010

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Forensika Digital
Program Studi Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia
2022

Lembar Pengesahan Pembimbing

Studi Komparasi Metode Disk Overwrite dan Factory Reset Sebagai Teknik Anti Forensik di Perangkat Android

Beni Ike Hendra Kuswara
20917010

Yogyakarta, Juni 2022

Pembimbing

Ahmad Rafie Pratama, S.T., MIT., Ph.D.

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

Lembar Pengesahan Penguji

Studi Komparasi Metode Disk Overwrite dan Factory Reset Sebagai Teknik Anti Forensik di Perangkat Android

Beni Ike Hendra Kuswara

20917010

Yogyakarta, Juni 2022

Tim Penguji,

Ahmad M. Raf'ie Pratama, S.T., MIT., Ph.D.

Ketua

Dr. Yudi Prayudi, S.Si., M.Kom.

Anggota I

Ahmad Luthfi, S.Kom., M.Kom., Ph.D.

Anggota II

Mengetahui,

Ketua Program Studi Informatika Program Magister

Universitas Islam Indonesia

Izzati Muhimmah, S.T., M.Sc., Ph.D.

Abstrak

Studi Komparasi Metode Disk Overwrite dan Factory Reset Sebagai Teknik Anti Forensik di Perangkat Android

Penelitian ini bertujuan untuk membandingkan efektivitas dan efisiensi dari metode *disk overwrite* dan fitur *factory reset* bawaan sebagai teknik anti-forensik di perangkat Android. Proses pengumpulan data di penelitian ini dilakukan dengan proses eksperimen di perangkat Android versi 10 yang telah melalui proses teknik anti-forensik tersebut secara bergantian sebelum dilakukan upaya pemulihan data yang telah terhapus dengan perangkat lunak Photorec dan WinHex. Dari hasil eksperimen, ditemukan bahwa proses recovery yang dilakukan memberikan hasil nyaris sama antara penggunaan metode *disk overwrite*, baik itu 1-*pass*, 3-*pass*, 7-*pass*, maupun 35-*pass*, jika dibandingkan dengan metode *factory reset* bawaan, meski dari sisi waktu operasinya terdapat perbedaan mencolok antara kelimanya. Dengan kata lain, penggunaan metode *disk overwrite* sebagai teknik anti-forensik, dalam kondisi normal, tidak memberikan nilai tambah jika dibandingkan dengan *factory reset* bawaan Android. Hasil dari penelitian ini dapat digunakan sebagai pegangan dan acuan oleh para praktisi forensika digital baru sebelum melakukan pemrosesan barang bukti elektronik berupa perangkat Android. Selain itu, hasil dari penelitian ini dapat menjadi bukti empiris akan efektivitas dan efisiensi dari fitur *factory reset* bawaan di perangkat Android dalam menjaga privasi pengguna saat perangkat tersebut berpindah kepemilikan.

Kata kunci

android 10, *disk overwrite*, *factory reset*, anti forensik, privasi data

Abstract

Comparative Study of Disk Overwrite and Factory Reset Methods as Anti-Forensics Techniques on Android Devices

This study aims to compare the effectiveness and efficiency of the disk overwrite method and the default factory reset feature as an anti-forensic technique on Android devices. The data collection process in this study was carried out by an experimental process on Android 10 devices, which had gone through each anti-forensic technique process in turn before attempting to recover deleted data using the Photorec and WinHex software. From the experimental results, it was found that the recovery process yielded nearly identical results between the use of the disk overwrite method, be it 1-pass, 3-pass, 7-pass, or 35-pass, and the default factory reset method, although in terms of operating times there was a stark difference between the five. In other words, the use of the disk overwrite method as an anti-forensic technique in normal cases does not provide any added value compared to the default Android factory reset feature. The results of this study can be used as a guide and reference by new digital forensics practitioners before processing electronic evidence in the form of Android devices. In addition, the results of this study can serve as empirical evidence of the effectiveness and efficiency of the default factory reset feature on Android devices in maintaining user privacy when the device changes ownership.

Keywords

Android 10, disk overwrite, factory reset, anti forensics, data privacy

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Juni 2022

Beni Ike Hendra Kuswara, S.Kom.

Daftar Publikasi

Publikasi yang menjadi bagian dari tesis

Jurnal JATISI (Jurnal Teknik Informatika dan Sistem Informasi) Volume 9 Nomor 2 yang terakreditasi SINTA 3, Edisi 20 Juni 2022 dengan Judul "**Studi Komparasi Metode Disk Overwrite dan Factory Reset sebagai Teknik Anti Forensik di Perangkat Android**".

Kontributor	Jenis Kontribusi
Beni Ike Hendra Kuswara	Mendesain eksperimen (60%) Menulis *paper* (80%)
Ahmad Raf'ie Pratama	Mendesain eksperimen (40%) Menulis dan mengedit *paper* (20%)
Erika Ramadhani	Melakukan review *paper*

Halaman Kontribusi

Penelitian ini dapat berjalan dan diselesaikan berkat kontribusi dari berbagai pihak antara lain Ahmad Raf'ie Pratama, ST., MIT, Ph.D., Erika Ramadhani, ST., M.Eng., Fietyata Yudha, S.Kom., M.Kom., Dr. Yudi Prayudi, S.Si., M.Kom., dan Ahmad Luthfi, S.Kom., M.Kom., Ph.D. Beliau-beliau telah banyak memberikan saran dan masukan mulai dari pra penelitian, seminar proposal, sidang kemajuan, hingga sidang pendadaran.

Halaman Persembahan

Bismillahirahmanirrahim.

Alhamdulillah, atas izin dan ridho Allah Subhannahu Wa Ta'ala, saya dapat menyelesaikan tesis saya ini. Hal ini tentu tidak lepas dari dukungan dan do'a kedua orang tua saya. Untuk itu saya persembahkan karya saya ini kepada Ibu saya, Yani Rahyuniati dan Bapak saya, Sujono. Terima kasih banyak, saya bersyukur terlahir sebagai anak kalian.

Kata Pengantar

Assalamu'alaikum Wr. Wb.

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayahnya kepada penulis, sehingga dengan izin-Nya penulis dapat menyelesaikan laporan tesis ini dengan baik. Laporan berjudul "Studi Komparasi Metode Disk Overwrite dan Factory Reset Sebagai Teknik Anti Forensik di Perangkat Android" ini disusun guna melengkapi persyaratan kelulusan untuk mendapatkan gelar Magister Komputer di bidang Forensika Digital.

Dalam prosesnya penulis mendapat banyak bantuan dari berbagai pihak, untuk itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Ahmad Raf'ie Pratama, ST., MIT., Ph.D., selaku dosen pembimbing 1 yang telah banyak memberikan masukan dan arahan terhadap jalannya penelitian yang penulis lakukan.
2. Ibu Erika Ramadhani, ST., M.Eng., sebagai pembimbing 2 yang selalu memberikan arahan kepada penulis berkaitan dengan alur penulisan.
3. Bapak Fietyata Yudha, S.Kom., M.Kom., yang menjadi dosen penguji ketika penulis melakukan sidang proposal.
4. Bapak Ahmad Luthfi, S.Kom., M.Kom., Ph.D., selaku dosen penguji kemajuan tesis dan sidang pendadaran.
5. Dr. Yudi Prayudi, S.Si., M.Kom., selaku dosen penguji sidang pendadaran.
6. Bapak dan Ibu penulis, Sujono dan Yani Rahyuniati, yang telah memberikan dukungan dan doanya serta selalu percaya bahwa penulis mampu.

Penulis menyadari bahwa dalam penyusunan laporan tesis ini masih terdapat banyak kekurangan, oleh karena itu kritik, saran, dan masukan yang membangun sangat diharapkan oleh penulis.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 10 Juni 2022

Penulis

Daftar Isi

Daftar Tabel

Daftar Gambar

BAB 1
Pendahuluan

1.1 Latar Belakang

Smartphone atau telepon pintar menjadi perangkat yang tak terpisahkan dalam kehidupan sehari-hari. Awalnya perangkat yang disebut telepon diciptakan sebagai alat komunikasi untuk melakukan panggilan suara, namun kini dengan perkembangan teknologi yang begitu pesat menjadikannya memiliki banyak fitur sehingga layak disebut *smartphone*. Beberapa di antaranya adalah kemampuan untuk mengakses internet, mengambil foto dan merekam video, fitur hiburan seperti memainkan serta mengedit *files* musik dan video, membuat serta mengedit dokumen dan *spreadsheet*, dan masih banyak lagi fitur lainnya. Bisa dikatakan bahwa kemampuan *smartphone* saat ini hampir sama dengan sebuah unit komputer pribadi. Terdapat beberapa sistem operasi dari telepon pintar yang beredar di pasaran, dan yang terbanyak berdasarkan data yang di-*published* pada April 2021, adalah sistem operasi Android seperti yang dapat terlihat pada Gambar 1.1.

Gambar 1.1 Persentase penggunaan sistem operasi *mobile* (Statcounter, 2021).

Demi mendukung banyaknya fitur seperti yang telah disebutkan sebelumnya, saat ini *smartphone* juga dilengkapi dengan kapasitas penyimpanan yang besar. Menurut yang tertera di situs web milik Samsung, kapasitas penyimpanan terbesar yang ada pada produk ponsel pintar Android milik mereka saat ini bahkan hingga mencapai 512GB (Samsung, 2022). Hal ini tentu saja memungkinkan pengguna menyimpan banyak data dan *files* pribadi atau yang bersifat rahasia milik mereka di dalamnya.

Masalah muncul ketika pengguna ingin menjual ponsel pintar miliknya atau hanya sekedar diberikan ke orang lain dengan tujuan ingin menggantinya dengan yang lebih baru. Data-data yang terdapat dalam ponsel pintar lama mereka ada kemungkinan dapat terbaca

oleh pemilik yang baru. Meskipun pemilik lama telah melakukan *factory reset* dengan tujuan untuk menghapus *file* dan data mereka serta mengembalikan ke setelan awal. Ada kemungkinan data milik pengguna lama dapat dipulihkan kembali dengan bantuan berbagai *tool*, salah satunya adalah PhotoRec. PhotoRec merupakan sebuah utilitas pemulihan *files* yang menggunakan *file signatures* sebagai dasarnya dan mendukung hingga 480 format *file* (Grenier, 2019). PhotoRec sering digunakan oleh para praktisi Forensika Digital dan buktinya dapat dilihat bahwa *tool* ini juga disematkan sebagai modul bawaan pada aplikasi Autopsy.

Kemungkinan kebocoran privasi seperti yang telah dijelaskan di atas dapat dicegah dengan memanfaatkan salah satu teknik anti forensik, yaitu *disk overwriting.* Teknik ini dapat digunakan untuk melakukan *data wiping* sehingga data lama yang telah dihapus sulit atau bahkan tidak dapat dipulihkan lagi. *Disk overwriting* bekerja dengan cara menimpa data yang ada di dalam media penyimpanan dengan data baru hingga membuat media penyimpanan tersebut penuh. Setelah penuh, data baru tadi akan dihapus untuk mengosongkan media penyimpanan. Proses penulisan ulang atau menimpa data lama dengan data baru dapat dilakukan beberapa kali (Piriform, 2021). Namun, tentu saja banyaknya penulisan ulang yang dilakukan akan berpengaruh pada lamanya waktu yang dibutuhkan (Gargean, 2019).

Terdapat banyak teknik *disk overwriting* berdasarkan berapa kali penulisan ulang pada media penyimpanan dilakukan, dan beberapa di antaranya adalah 1 kali, 3 kali, 7 kali, dan 35 kali. Beberapa berpendapat bahwa, 1 kali *overwriting* sudah cukup untuk melakukan *data wiping* sehingga usaha untuk melakukan pemulihan menjadi tidak mungkin (Wani et al., 2020). Di sisi lain ada juga yang berpendapat bahwa mengembalikan perangkat ke setelan pabrik telah membuat data dan aplikasi pada perangkat terhapus sepenuhnya, meskipun pembuktian mengenai klaim tersebut belum pernah dilakukan (Chukwuemeka Ogazi-Onyemaechi et al., 2017).

Penelitian ini mencoba membuktikan apakah benar *smartphone* yang telah dikembalikan ke setelan awal atau *factory reset*, data dan *files* di dalamnya telah terhapus secara permanen dan sulit atau tidak mungkin dipulihkan kembali. Tujuan lainnya adalah untuk membandingkan efektivitas dan efisiensi penghapusan data melalui *factory reset* dengan beberapa teknik anti forensik *disk overwriting* dalam konteks menjaga privasi dan kerahasiaan. Penilaian dari sisi efektivitas secara garis besar ditentukan oleh banyaknya artefak digital yang dapat dipulihkan dan berhubungan dengan privasi. Semakin sedikit, maka teknik penghapusan tersebut dianggap semakin baik atau semakin efektif, begitu juga

sebaliknya. Kemudian dari sisi efisiensi, nantinya akan dilakukan perbandingan antara banyaknya artefak digital yang dapat dibangkitkan kembali dan berhubungan dengan privasi dengan waktu yang dibutuhkan untuk melakukan operasi penghapusan. Jadi, semakin kecil angka perbandingannya maka dianggap sebagai teknik penghapusan yang paling efisien.

1.2 Rumusan Masalah

Berdasarkan latar belakang seperti yang telah diuraikan di atas, maka rumusan masalah dalam penelitian ini antara lain:

1. Apakah benar *factory reset* milik Android 10 dapat menghapus secara permanen data privasi milik pengguna?
2. Manakah yang paling efektif dan efisien dalam hal penghapusan data pribadi apakah *factory reset* milik Android 10 atau beberapa teknik anti-forensik *disk overwriting*?

1.3 Batasan Masalah

Penelitian ini memiliki batasan pada objek penelitiannya, yaitu perangkat *smartphone* dengan sistem operasi Android versi 10. Selain itu ekstensi *files* yang menjadi objek dalam penelitian ini juga dibatasi pada beberapa ekstensi *files* yang dianggap populer, seperti DOCX, PPTX, XLSX, PDF, JPG, PNG, AVI, MP4, MKV, dan MP3.

1.4 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah sebagai berikut:

1. Membuktikan apakah benar *factory reset* milik Android 10 dapat menghapus data pribadi secara permanen.
2. Melihat manakah yang paling efektif dan efisien dalam hal penghapusan data pribadi, apakah *factory reset* milik Android 10 atau beberapa teknik anti-forensik *disk overwriting*.

1.5 Manfaat Penelitian

Beberapa manfaat dari hasil penelitian ini antara lain:

1. Menjadi pegangan dan acuan para praktisi forensika digital baru sebelum melakukan pemrosesan barang bukti elektronik berupa perangkat Android dengan sistem operasi versi 10.

2. Memberikan pengetahuan kepada para pengguna *smartphone* Android terutama yang versi 10, berkaitan dengan penghapusan data pribadi. Sekaligus juga, untuk menunjukan secara gamblang mengenai apa yang terjadi pada data lama yang sebelumnya tersimpan di perangkat Android versi 10 setelah dilakukan *factory reset*.

1.6 Penelitian Terdahulu

Beberapa tahun ke belakang, juga pernah dilakukan beberapa penelitian sejenis dan menjadi latar belakang pada studi ini. Penelitian pertama pada tahun 2014 dengan judul *Effects of the Factory Reset on Mobile Devices* (Schwamm & Rowe, 2014), mempelajari mengenai efek-efek yang ditimbulkan pada perangkat *mobile* ketika dikembalikan ke setelan pabrik. Sebagai objek penelitian adalah 28 perangkat *mobile* dari berbagai vendor dengan berbagai sistem operasi. Salah satunya yang diteliti adalah Samsung Galaxy SIII dengan sistem operasi CynaogenMod 10.1 (CM 10.1) yang berbasis Android 4.2 Jelly Beans. Penelitian dimulai dengan melakukan setelan pabrik setelah itu dilanjutkan dengan menempatkan beberapa *files* pada perangkat, lalu mengunduh dan memasang sejumlah aplikasi serta melakukan beberapa kegiatan melalui aplikasi-aplikasi tersebut. Proses *imaging* dilakukan setelahnya kemudian diikuti dengan melakukan pengembalian ke setelan pabrik dan dilakukan *imaging* lagi. Hasil dari penelitian ini menunjukan bahwa *files* yang sengaja dimasukan ke dalam perangkat tidak dihapus saat proses *factory reset*, begitu juga dengan foto-foto yang diambil menggunakan kamera perangkat.

Studi berikutnya pada tahun 2015 berjudul *Security Analysis of Android Factory Resets* (Simon & Anderson, 2015) yang menganalisis keamanan dari perangkat setelah dilakukan *factory reset*. Penelitian dilakukan pada 21 perangkat telepon pintar Android dari 5 *vendor* dengan versi sistem operasi Android antara 2.2 hingga 4.3. Salah satu dari hasil studi ini menunjukan bahwa *files* multimedia yang ada di dalam perangkat berupa foto-foto hasil kamera, video, dan *web thumbnails* dapat dipulihkan seratus persen menggunakan *tool* Photorec.

Penelitian ketiga pada tahun 2017 dengan judul *Performance of Android Forensics Data Recovery Tools* (Chukwuemeka Ogazi-Onyemaechi et al., 2017) dilakukan untuk melakukan pengetesan serta membandingkan tingkat keefektifan beberapa *tools* yang digunakan untuk memulihkan data yang terhapus oleh *factory reset*. Perangkat yang digunakan dalam penelitian adalah Samsung Galaxy S2 i9100 dengan versi Android Gingerbread 2.3.4. Jalannya penelitian dilakukan dengan memasukan beberapa data yang berhubungan dengan *fraud* ke dalam perangkat untuk kemudian dikembalikan ke setelan

pabrik dan dilakukan proses *imaging* lalu selanjutnya dicoba untuk dipulihkan data-datanya dan kemudian dianalisis. Hasilnya menunjukan bahwa hampir semua *recovery tools* yang digunakan dapat memulihkan sebagian besar *files* dengan berbagai format, dimana sebagian besar berupa *files* gambar dengan format JPG dan PNG.

Perbedaan antara penelitian ini dengan ketiga penelitian tersebut adalah kebaharuan. Seperti yang dapat dilihat pada penjelasan sebelumnya, objek pada ketiga penelitian terdahulu merupakan perangkat telepon pintar dengan sistem operasi Android versi lama, yaitu antara versi 2 hingga versi 4. Tentu saja hal ini sudah tidak relevan lagi dengan kondisi saat ini, dimana telepon pintar sudah mengoperasikan sistem operasi Android yang jauh lebih modern. Mengacu pada alasan tersebut, studi ini mencoba melihat apakah hal yang sama masih terjadi pada perangkat modern dengan Android versi 10 setelah dilakukan *factory reset.* Mencoba melihat apakah data lama milik pengguna masih dapat dipulihkan kembali setelah perangkat dikembalikan ke setelan pabrik.

Misalkan hal yang sama masih terjadi, penelitian ini juga bermaksud mengusulkan pemanfaatan *disk overwriting* yang merupakan salah satu teknik anti-forensik kepada para pengguna awam telepon pintar Android, dengan tujuan untuk menjaga privasi. Oleh karena itu, perlu dilakukan pula pembuktian untuk menunjukan bahwa teknik *disk overwriting* dapat dimanfaatkan untuk menghapus data pribadi secara permanen, sesuai dengan teorinya. Pembuktian yang dimaksud, dilakukan dengan membandingkan efektivitas dan efisiensi penggunaan *factory reset* dengan beberapa teknik anti-forensik *disk overwriting* dalam hal penghapusan data yang berkaitan dengan privasi. Hasil yang didapat dari penelitian ini diharapkan dapat membuka mata para pengguna telepon pintar Android, serta memberikan mereka beberapa opsi yang dapat mereka pilih yang sekiranya sesuai, ketika mereka berniat menghapus data pribadi mereka dari suatu perangkat Android.

Tabel 1.6 Rangkuman penelitian terdahulu.

No	Peneliti	Tema	Versi Android	Hasil
1	(Schwamm & Rowe, 2014)	Mengevaluasi efektivitas *factory reset* pada beberapa perangkat *mobile*, antara lain iPhone,	Android 4.2 Jelly Beans	*Factory reset* pada perangkat Android tidak efektif menghapus *files* yang sengaja dimasukkan.

No	Peneliti	Tema	Versi Android	Hasil
		Android, dan Blackberry.		Termasuk di dalamnya berupa gambar yang diambil menggunakan kamera, *file* txt, doc, pdf, dan ppt yang masih bisa ditemukan setelah proses *factory reset*.
2	(Simon & Anderson, 2015)	Menganalisis tingkat keamanan dari perangkat Android setelah dilakukan *factory reset*.	Andoid versi 2.2 Froyo – 4.3 Jelly Beans	*Files* multimedia yang berupa foto-foto hasil kamera, video, dan *web thumbnails* dapat dipulihkan seratus persen menggunakan *tool* Photorec.
3	(Chukwuemeka Ogazi-Onyemaechi et al., 2017)	Melakukan pengetesan serta membandingkan keefektifan beberapa *tools* yang digunakan untuk memulihkan data yang terhapus oleh *factory reset*.	Android versi 2.3.4 Gingerbread	Hampir semua *recovery tools* yang digunakan dapat memulihkan sebagian besar *files* dengan berbagai format, dimana sebagian besar berupa *files* gambar dengan format JPG dan PNG.

1.7 Sistematika Penulisan

Untuk memberikan gambaran dan mempermudah dalam penyusunan penelitian ini, maka dibuatlah urutan penulisan sebagai berikut:

BAB I Pendahuluan

Bab ini berisi uraian pengantar mengenai masalah yang akan diteliti. Secara detail bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, penelitian terdahulu, dan sistematika penulisan.

BAB II Tinjauan Pustaka

Di dalam bab ini dijelaskan mengenai teori-teori yang melandasi dan terkait serta digunakan dalam penelitian ini.

BAB III Metodologi Penelitian

Bab metodologi penelitian berisi langkah-langkah atau alur jalannya penelitian dari awal sampai akhir.

BAB IV Pembahasan

Berisi hasil-hasil atau temuan-temuan dari penelitian berikut uraian analisis mengenai hasil atau temuan tersebut.

BAB V Penutup

Di dalam bab ini dijelaskan mengenai kesimpulan akhir yang di dapat dari hasil analisis penelitian untuk menjawab rumusan masalah. Di sini juga dijelaskan mengenai penelitian lanjutan yang diharapkan dapat dilakukan di masa depan.

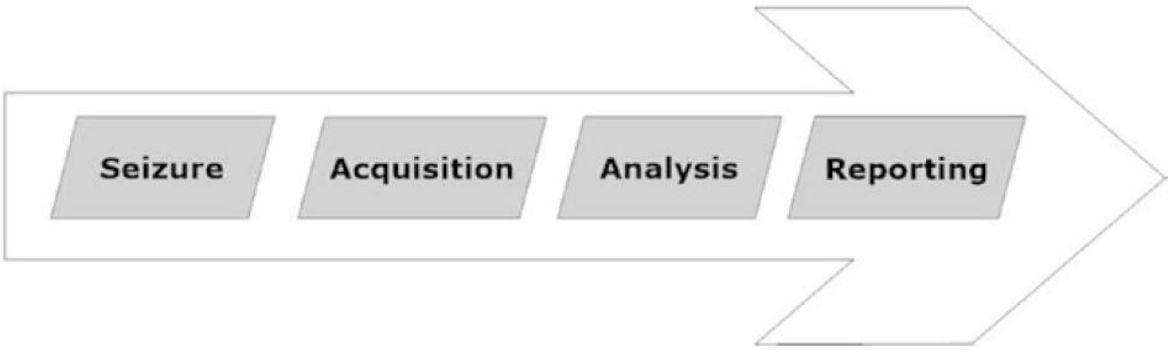
BAB 2
Tinjauan Pustaka

2.1 Forensika Digital

Forensika digital adalah cabang dari ilmu forensik yang menggunakan ilmu pengetahuan untuk mengumpulkan, menganalisis, mendokumentasikan, dan menyajikan bukti digital yang berhubungan dengan kejahatan siber di persidangan (Hassan, 2019). Teknik forensika digital digunakan oleh penyidik untuk mengumpulkan bukti dari berbagai macam perangkat digital. Ada banyak *tool* dan teknik yang bisa digunakan untuk mencari bukti-bukti digital yang relatif sulit ditemukan, seperti bukti yang telah dihapus, dikunci, atau disamarkan (Afonin Oleg et al., 2015). Ilmu forensika digital juga dapat digunakan untuk merekonstruksi ulang aktivitas dari pelaku kejahatan dan untuk mendapatkan informasi mengenai si pemilik komputer (Garfinkel, 2007).

Terdapat banyak standar yang dibuat untuk melakukan pemeriksaan forensika digital. Masing-masing memiliki pendekatan dan jumlah langkah atau fase yang berbeda. Namun hampir semua standar atau pendekatan yang ada dibagi menjadi empat langkah utama dengan urutan seperti pada Gambar 2.1:

1. *Seizure*
2. *Acquisition*
3. *Analysis*
4. *Reporting*



Gambar 2.1 Empat langkah utama pemeriksaan forensika digital (Hassan, 2019).

2.1.1 Seizure

Pada fase ini, bukti fisik (perangkat digital) akan disita dan dipindahkan dengan aman dari tempat kejadian perkara ke laboratorium forensik. Perangkat digital yang dimaksud dapat berupa perangkat komputasi dengan tipe apapun seperti, *laptop*, tablet, *mobile phone*, *external hard drive*, USB *flash drive, wearable device* seperti *smart watch*, bahkan juga komputer *desktop*. Namun perlu diingat, bahwa untuk melakukan hal ini diperlukan surat ijin dari pihak yang berwenang.

Perangkat yang menjadi bukti digital yang berada di tempat kejadian perkara harus ditangani dan periksa oleh teknisi yang benar-benar terlatih, guna memastikan bukti digital yang di dapat bisa dipertanggung jawabkan sesuai dengan hukum dan kaidah-kaidah forensik yang berlaku. Ketika komputer milik tersangka didapati masih dalam kondisi *running*, maka perlu dipertimbangkan untuk mengakuisisi *volatile memory*-nya (RAM) jika memungkinkan. Pada praktik kuno forensika digital, komputer yang disita dan berada dalam keadaan menyala harus dicabut sambungan listriknya atau dilakukan *hard shutdown*, untuk kemudian dimasukan ke dalam kotak antistatis. Namun pada praktik forensika digital kekinian, menganggap bahwa penting untuk mengakuisisi *volatile memory* ketika perangkat masih dalam keadaan hidup.

Memori RAM bisa dibilang kaya akan informasi, beberapa di antaranya meliputi kunci kriptografi, log *chat* IM, konten-konten tak terenkripsi, konten dapa *clipboard*, informasi mengenai proses-proses yang berjalan, dan masih banyak lagi yang lainnya. Proses akuisisi RAM juga harus didokumentasikan berikut *tools* yang digunakan, dimana hal ini juga harus dimasukkan ke dalam laporan akhir investigasi. Penyebabnya adalah *tools* yang digunakan untuk melakukan akuisisi RAM dapat menyebabkan perubahan-perubahan *minor* pada *files* milik sistem operasi, memori RAM itu sendiri, dan *hard drive*.

2.1.2 *Acquisition*

Fase akuisisi berkaitan dengan penyimpanan sekunder dari perangkat komputasi, (contohnya HDD, SSD, *thumb drive*, *tape drive*) dan *volatile memory* (RAM) jika perangkat dalam keadaan menyala. Pada fase ini dalam kasus komputer forensik, penyidik akan melakukan duplikasi terhadap *hard drive* milik tersangka yang biasa dikenal dengan *bit-to-bit image* untuk membuat salinan utuh dari *hard drive* yang disita. Analisis untuk mencari bukti-bukti yang berkaitan dengan kasus kejahatan yang sedang diselidiki akan dilakukan pada salinan *image* tersebut.

Penyidik biasanya menggunakan *hardware duplicator* atau *software imaging tools* seperti perintah DD pada Linux untuk melakukan duplikasi media penyimpanan. Perlu diingat, bahwa untuk melakukan hal ini *hard drive* milik tersangka harus dalam keadaan *write-protected* untuk menghindari terjadinya *tampering* pada bukti aslinya.

2.1.3 Analysis

Di fase ini, isi dari *image* forensik yang telah didapatkan dianalisis menggunakan sejumlah *tools* untuk mendapatkan bukti-bukti yang berkaitan dengan kasus yang ditangani. Hal-hal seperti *files* tersembunyi, terhapus, dan terenkripsi, berikut hal lain seperti log *chat* IM, histori dari kegiatan *browsing* di internet, serta email yang terhapus, semuanya dapat dipulihkan menggunakan *tools* khusus semacam EnCase, Sleuth Kit, Volatility, dan Forensic Toolkit (FTK) dari AccessData.

Pada fase ini, hasil analisis nilai *hash* digunakan oleh *tool* forensik untuk mengidentifikasi *files* penting atau untuk membuat pengecualian terhadap *files* tersebut. Jadi konten-konten dari *image* yang telah didapatkan sebelumnya akan dihitung nilai *hash*-nya untuk kemudian dibandingkan dengan *precompiled list* yang telah dibuat sebelumnya, sebagai contoh adalah RDS (*Reference Data Set*) milik *National Software Reference Library*. Dalam hal ini RDS digunakan untuk mengecek *files* pada komputer yang disita dengan mencocokkan profile *files* dengan yang ada di RDS. Cara ini akan sangat meringankan usaha untuk menentukan *files* mana yang penting dan bisa dijadikan bukti sesuai dengan kasus yang sedang diselidiki.

Selain dengan pencocokan nilai *hash,* proses analisis juga dapat menggunakan cara pencarian dengan kata kunci, frase, atau istilah tertentu guna mendapatkan bukti-bukti penting yang berkaitan dengan kasus. Hal ini dapat secara efektif meningkatkan kecepatan investigasi untuk menemukan informasi-informasi yang relevan.

2.1.4 *Reporting*

Pada tahap *reporting*, penyidik membuat laporan terstruktur dari hasil pemeriksaan berisi temuan-temuan yang berkaitan dengan kasus. Laporan yang dibuat oleh penyidik ini biasanya ditujukan untuk orang-orang non teknis (contohnya: pengacara, jaksa, dan hakim). Oleh karena itu gaya penulisan, istilah-istilah yang dipakai, serta cara penyajian laporan harus benar-benar diperhatikan. Penyajian laporan biasanya juga diikuti oleh penyajian bukti, dimana seringnya berupa bukti digital.

Secara umum, isi laporan hasil pemeriksaan terdiri dari beberapa hal, di antaranya:

- Ringkasan mengenai temuan-temuan penting.
- Deskripsi mengenai *tools* yang digunakan, baik *software* maupun *hardware* yang digunakan selama proses investigasi. Juga di dalamnya disertakan versi *tools* yang berupa *software*.
- Metode yang digunakan untuk mendapatkan bukti digital.
- Deskripsi mengenai bukti digitalnya, (konten dari *image*) berikut artefak-artefak penting yang berhasil ditemukan, contohnya *internet browsing history*, histori *email*, hasil analisis registry mengenai perangkat USB yang pernah tersambung, dan *files* yang telah terhapus. Dalam menjelaskan hal-hal tersebut, akan lebih baik jika disertakan tangkapan layar mengenai langkah-langkah yang dilakukan.
- Penjelasan mengenai istilah-istilah yang digunakan, dengan tujuan untuk memudahkan orang-orang non teknis memahami maksud laporan.
- Kesimpulan dari proses investigasi yang telah dilakukan.

Hard drive asli milik tersangka beserta salinan digitalnya (*images*) harus disajikan bersama dengan laporan yang telah dibuat.

2.2 *Rooting*

Secara singkatnya, *rooting* adalah proses untuk memperoleh akses *root* dengan mengeksploitasi celah keamanan pada sistem atau menggunakan *custom recovery images* (Feng et al., 2018). Atau dengan kata lain adalah memperoleh akses penuh terhadap sistem. Pada ekosistem Android, dimana sistemnya berbasiskan Linux *permission* dan *file-system ownership*, *rooting* berarti mendapatkan akses "*superuser*" (Snyder, 2021). Secara umum proses *rooting* dilakukan menggunakan Android SDK *tools* untuk meng-*unlock* bootloader dan kemudian melakukan *flashing* sebuah *custom image* ke dalam perangkat (Snyder, 2021).

Untuk lebih memahami apa itu *rooting*, berdasar literatur (Tamma & Tindall, 2015), maka sangat penting untuk memahami terlebih dahulu cara kerja sistem Unix. Sistem operasi Unix yang asli, dimana merupakan dasar dari sistem operasi Linux dan sistem Unix-*like* lainnya, pada awalnya dirancang sebagai sistem untuk *multiuser*. Hal ini dikarenakan pada jaman itu komputer pribadi belum ada, oleh karenanya, sangat penting saat itu memiliki satu mekanisme untuk memisah dan melindungi sumber daya komputer yang dimiliki oleh satu individu *user* dan secara bersamaan memungkinkan mereka untuk tetap menggunakan sistem.

Namun, untuk dapat melakukan *tasks* yang memerlukan hak akses khusus, seperti mengizinkan atau menolak hak akses yang diminta oleh *ordinary users*, mengakses *files* penting milik sistem dengan tujuan memperbaiki atau meng-*upgrade* sistem, dan seterusnya. Karena hal-hal inilah, maka sangat penting untuk memiliki sebuah akun *administrator* yang memiliki akses *superuser*. Jadi dari sini dapat disimpulkan, bahwa terdapat dua macam tipe akun, yaitu akun pengguna normal yang mempunyai hak akses lebih sedikit dan akun *superuser* atau *root* yang memiliki hak akses ke semua hal.

Berdasarkan penjelasan tersebut, maka bisa dibilang *root* adalah *username* atau akun yang secara *default* memiliki hak akses terhadap semua perintah dan *files* yang ada pada sistem Linux atau sistem Unix-*like* lainnya. Jadi pada sistem operasi Linux, *root user* punya wewenang untuk memulai atau menghentikan segala layanan pada sistem, meng-*edit* atau menghapus *files*, mengubah hak akses pengguna-pengguna lain dan berbagai hal lainnya.

Dikarenakan Android menggunakan kernel milik Linux, maka bisa dibilang hampir keseluruhan konsep yang ada di Linux juga berlaku di Android. Namun, ketika seseorang membeli sebuah perangkat Android, hak akses sebagai *root* tidak diberikan secara *default*, melainkan memerlukan usaha mandiri untuk mendapatkannya.

2.3 *Physical Acquisition*

Dalam penelitian ini, salah satu proses forensika digital yang digunakan adalah *physical acquisition* atau pengkopian *bit-by-bit* seluruh memori penyimpanan untuk dijadikan *disk images* (Feng et al., 2018). Dikutip dari (Mahalik et al., 2016), *physical acquisition* juga bisa disebut sebagai *physical extraction*, yaitu proses untuk mendapatkan *bit-by-bit image* dari sebuah perangkat yang sama persis dengan aslinya.

Perlu dipahami bahwa *bit-by-bit image* tidak sama dengan melakukan *copy paste* konten dari sebuah perangkat. Jika yang dilakukan adalah operasi *copy* dan *paste*, maka yang tersalin hanya *files* seperti *visible files, hidden files*, dan *files* yang berhubungan dengan sistem. Metode semacam ini biasa disebut sebagai *logical acquisition* atau *logical extraction* dan hasilnya disebut sebagai *logical image*. Jika yang digunakan adalah metode *logical acquisition*, *files* terhapus serta *files* yang tidak dapat diakses tidak dapat tersalin oleh perintah peng-*copy*-an. Berbeda dengan *logical extraction*, *physical extraction* menghasilkan salinan yang sama persis dari memori perangkat, dimana termasuk di dalamnya adalah *slack space, unallocated space,* dan lain sebagainya.

Ekstraksi data dari perangkat Android melalui teknik *physical acquisition* biasanya dilakukan menggunakan perintah dd pada Linux. Dan untuk menjalankan perintah dd itu

sendiri, diperlukan hak akses sebagai *root*. *Image* hasil dari perintah dd biasa disebut sebagai *physical image, forensic image,* atau *raw image.*

2.4 Anti-Forensik

Di lain sisi yang berseberangan terdapat satu disiplin ilmu yang merupakan lawan dari forensika digital, yaitu anti-forensik. Didefinisikan sebagai satu set teknik pencegahan yang dapat dilakukan oleh seorang *user* dengan tujuan untuk menyembunyikan jejak aktivitasnya sehingga membuat proses investigasi pada media digital menjadi lebih rumit dan memakan waktu, yang berpotensi membuat aktivitas ilegal menjadi sulit atau bahkan tidak mungkin untuk dipulihkan (Afonin Oleg et al., 2015).

Dewasa ini, anti-forensik mendapat banyak ketertarikan dan perhatian dari sejumlah pihak dikarenakan tiga alasan utama berdasarkan pemanfaatannya (Wani et al., 2020):

- Dapat digunakan untuk menjaga data pribadi yang berkaitan dengan privasi seseorang.
- Dapat digunakan oleh penyidik forensik untuk mendapatkan bukti digital yang kredibel dan dapat diterima.
- Dapat digunakan oleh kalangan akademik dan peneliti untuk mengidentifikasi batasan dari *tool* dan teknik forensik.

Masih berdasarkan literatur yang sama, dimana di situ dinyatakan bahwa, secara tradisional anti-forensik dapat diklasifikasikan menjadi beberapa kategori, antara lain (Wani et al., 2020):

- *Artefak wiping*
- *Data hiding*
- *Trial obfuscation*
- *Attacking Forensic tools*

2.4.1 *Artefak Wiping*

Merupakan teknik anti-forensik yang paling *basic* dan *primitive*. Penggunaan teknik ini bertujuan untuk menyingkirkan segala bentuk jejak digital dengan cara menghapus *files* atau menghancurkan *file systems*. Hal ini dapat dilakukan dengan menggunakan utilitas *file-wiping* dan *disk sanitizing* atau *disk-degaussing* dan teknik penghancuran.

Teknik *artefak-wiping* yang paling populer dan lazim digunakan adalah *file wiping*. Pada teknik *file wiping*, isi dari suatu *file* akan ditimpa oleh sejumlah data acak saat proses

penghapusan, hal ini untuk memastikan segala bentuk usaha pemulihan *file* tersebut berakhir nihil. Terdapat beberapa mitos yang berkaitan dengan *wiping*. Salah satunya adalah klaim bahwa *magnetic force microscopy* dan beberapa teknik lain dapat memulihkan data yang ditimpa atau di-*overwrite* satu atau dua kali, dengan cara mengeksploitasi ketidak akuratan posisi *head* pada *magnetic drives*. Namun, hal ini dapat dicegah menggunakan *three-pass overwrite* yang diusulkan oleh NIST dan 35-*pass* milik Peter Gutmann.

Dikarenakan *file wiping* merupakan teknik menjaga privasi yang paling *basic* dan tua, maka *tools* gratis yang berkaitan dengan teknik ini pun banyak dijumpai di pasaran dan bahkan mendukung berbagai *platform*. *Tool* semacam *shred and wipe* di Linux atau *delete partition* (bagian dari DiskPart) di Windows, mendukung penghapusan aman terhadap *files* dan partisi. *Tools* tersebut memiliki fitur *multiple overwrites* dengan berbagai pilihan data acak untuk melakukan *overwrite*, sehingga membuat penghapusan yang aman menjadi tugas mudah.

Namun, *file-wiping* juga memiliki semacam kendala dalam pengoperasiannya, yaitu ketika berhadapan dengan sistem *file* modern seperti Btrfs. Sistem *file* ini dilengkapi dengan teknik pengamanan yang canggih, mekanisme pengecekan duplikasi dan *checksum* yang tangguh, membuat aktivitas *file-wiping* menjadi lebih rumit. Metode *update copy-on-write* (COW) yang digunakan Btrfs dan beberapa *file system* modern lain, yang menyebar salinan data ke seluruh *file system*, membuat proses *wiping* pada satu individu *file* menjadi sulit untuk dijalankan. Hal ini disebabkan oleh perlunya dilakukan analisis yang lebih mendalam untuk mengidentifikasi salinan *file* yang tersebar tadi. Namun demikian, teknik *file-wiping* tetap dapat dilakukan dengan menggunakan *tools* yang secara spesifik ditujukan untuk *file system* yang dimaksud.

2.4.2 *Data Hiding*

Dapat dikatakan, juga merupakan salah satu dari teknik anti-forensik yang tertua. Bekerja dengan cara menyembunyikan data di dalam struktur *file system*. Dengan begitu, akan menjadi sangat sulit data tersebut untuk ditemukan ketika proses investigasi. Teknik ini mengambil keuntungan dari besarnya jumlah data yang disimpan dan besarnya kapasitas *hard drives* modern. Hal ini dapat membuat proses investigasi menjadi tertunda atau bahkan dibatalkan. Teknik *data hiding* juga mengeksploitasi berbagai fitur dari *file system* untuk menjaga data tetap berada di dalam *file system* tanpa bisa terdeteksi oleh *tool* forensik.

Terdapat beberapa tempat yang bisa dijadikan tempat melakukan *data hiding*, di antaranya adalah:

Reserved locations

Merupakan tempat atau ruang yang sengaja disediakan dan berada di dalam struktur metadata dari suatu *file system*. Ruang tersebut ditujukan untuk mendukung tambahan fitur *file system* dan untuk berjaga-jaga jika ada *system upgrade* di masa depan. Ruang yang disediakan ini akan tetap kosong selama belum ada *system update*. Oleh sebab itu, data apapun yang ditempatkan di ruang ini akan bertahan dalam waktu yang cukup lama, tanpa mengganggu kinerja sistem, dan tidak akan tersentuh atau hilang jika dilakukan *overwriting*. Di sisi lain, data yang ditempatkan di *reserved locations* biasanya juga tidak akan dilirik oleh *tools* forensik.

Terdapat beberapa *tools* yang dapat digunakan untuk mengeksploitasi *reserved locations* untuk keperluan *data hiding*. Antara lain DataMuleFS yang memanfaatkan *reserved locations* milik *file system* Ext2 dan Ext3. Kemudian WaffenFS yang memanfaatkan jurnal Ext3. Lalu untuk *file system* Ext4 dan Btrfs, dapat memanfaatkan perintah di utilitas dd untuk melakukan *data hiding* pada *reserved locations*-nya.

Slack space

Sebagian besar *file system* mendukung dua tipe *slack space*, yaitu *file slack* dan *volume slack***.** *File slack* adalah ruang kosong yang terletak di antara ujung akhir *file* dan ujung akhir blok serta terletak di dalam *allocated extent*. Sedangkan *volume slack,* merupakan ruang kosong yang ada di antara ujung akhir *file system* dan ujung akhir partisi.

Tools semacam bmap dan dd di Linux, lalu slacker dan dd di Windows dapat digunakan untuk menyembunyikan data dan kode di dalam *slack space*. *Slack space* sendiri merupakan bagian dari *allocated* dan *un-allocated space*, untuk itu langkah identifikasi, ekstraksi, dan validasi segala macam data forensik yang berada di dalam *slack space* membutuhkan *tools* dan teknik forensik khusus. Lalu jika dibandingkan dengan menyembunyikan data di *reserved locations*, ukuran data yang bisa disimpan di *slack space* jauh lebih besar.

Cryptographic file systems

Cryptographic file system menyimpan dan me-*manage* data di dalam media penyimpanan perangkat dalam bentuk yang terenkripsi. *File system* semacam ini memanfaatkan enkripsi data pada tingkatan yang berbeda dengan cara mengenkripsi segala hal yang dituliskan ke dalam *disk*. Motivasi utama penggunaan *file system* jenis ini adalah untuk melindungi data dari *user* yang tidak berwenang, sebagai tindak pencegahan ketika fungsi *user access control*

milik sistem operasi mengalami kegagalan. Pada *file system* jenis ini terdapat pilihan untuk melakukan enkripsi terhadap *files* atau *folder* tertentu, juga mendukung berbagai algoritma enkripsi, serta bisa digunakan untuk *remote storage*.

Namun, teknik anti-forensik jenis ini tidak begitu populer sebab terdapat kekurangan dalam hal performa atau kecepatan pemrosesan, juga terdapat masalah terhadap manajemen kunci enkripsi dan dekripsinya. Selain itu, investigator juga dapat dengan mudah bisa mendapatkan kunci untuk membuka enkripsinya dengan menggunakan taktik-taktik tak biasa untuk mendapatkannya dari *user* yang berwenang.

Steganographic file systems

Jika pada *cryptographic file systems* lokasi dari bukti digital diketahui, namun membutuhkan otorisasi untuk melakukan dekripsi terhadap datanya. Sebaliknya, pada *steganography file systems,* eksistensi dan lokasi dari data yang menjadi bukti digital tidak diketahui. Jadi bisa dibilang bahwa, *steganographic file systems* merupakan teknik *data-hiding* paling kuat. Teknik steganografi mampu menyembunyikan data di dalam *carrier files* dengan berbagai format, mulai dari grafis, audio, video, binary, teks, dan lain sebagainya.

2.4.3 Trail Obfuscation

Merupakan teknik anti-forensik yang berbahaya, sebab tujuannya adalah membingungkan penyidik atau *investigator* dengan memalsukan bukti-bukti yang ada. Teknik ini membuat usaha-usaha investigasi yang dilakukan *investigator* menjadi terbantahkan, sehingga akan menguras waktu dan sumber daya yang dimiliki. Biasanya teknik anti-forensik jenis ini melakukan eksploitasi terhadap struktur *time-stamp* dan *magic-number* terhadap suatu *file system.*

Forging timestamp

Struktur metadata *timestamp* pada suatu *file system* sangatlah penting dan signifikan bagi investigasi forensik. Struktur *timestamp* merupakan inti dari proses rekonstruksi urutan kejadian yang mengarah pada tindakan kriminal yang telah dilakukan. *Timestamp* juga membantu penyidik untuk mengerucutkan daftar *files* yang mencurigakan dengan mengacu pada *timestamp* milik *file* tersebut dan dibandingkan dengan waktu ketika serangan terjadi.

Namun sayangnya, struktur metadata dari *timestamp* sangat rentan untuk dimanipulasi, yang mengakibatkan kebingungan dan membuat arah investigasi menjadi salah. Hal ini juga didukung oleh mudahnya melakukan hal tersebut dengan dukungan dari

tools semacam touch di Linux dan TimeStomp di Windows. Meskipun sebenarnya manipulasi terhadap *timestamp* dapat dideteksi oleh *tools* forensik modern dan beberapa bahkan mampu mengembalikan *timestamp* aslinya, namun tetap saja akan menimbulkan keragu-raguan yang besar dan dipertanyakan di depan persidangan.

Modifying magic numbers

File system menggunakan suatu *integer* yang unik dan bernilai konstan untuk merujuk tipe *file system* tertentu. Nilai ini disebut sebagai *magic numbers* yang digunakan oleh *disk utilities* dan *mounting operation* milik sistem operasi untuk mengidentifikasi tipe *file systemnya*. Jika nilai dari *magic numbers* ini dimanipulasi, maka *tools* forensik akan gagal dalam operasinya membuka *file system* yang sedang dianalisis. Atau berhasil membukanya, namun *file system* tadi dibuka sebagai tipe *file system* yang lain. Hal ini tentu saja kan membingungkan proses investigasi.

Using Live Distros

Beberapa sistem operasi Linux seperti Kali, dapat dijalankan secara *live* dari media penyimpanan portabel tanpa perlu dilakukan instalasi. Data dari sistem operasi yang dijalankan secara *live* akan dimuat ke dalam RAM tanpa mengubah apapun pada *secondary storage*-nya. Hal ini dapat dimanfaatkan oleh pelaku kriminal atau mungkin *hacker* untuk mengakses komputer korban dan melakukan pencurian informasi tanpa meninggalkan bukti atau jejak. Beberapa studi menunjukan bahwa, *live distros* tidak meninggalkan jejak aktivitas apapun ketika berpindah-pindah *directory* dan melakukan akses terhadap data, selama tidak dilakukan modifikasi terhadap data milik *file system*.

2.4.4 *Attacking forensic tools*

Tools forensik merupakan tenaga utama dari suatu proses investigasi forensik. Dan dikarenakan *tools* forensik terkenal serta terdokumentasi dengan baik, maka lubang keamanan yang mereka miliki pun juga banyak diketahui orang. Dengan mengeksploitasi lubang keamanan yang dimiliki oleh *tools* forensik, maka temuan-temuan yang dihasilkan oleh *tools* tersebut menjadi dipertanyakan, sehingga mengganggu proses investigasi.

Salah satu cara melakukan serangan terhadap *tools* forensik adalah dengan *compression bomb*. Juga biasa disebut sebagai zip *bomb*, merupakan sebuah *file* kecil yang dikompres, yang akan berubah menjadi sangat besar ketika dilakukan dekompresi. Ukurannya yang sangat besar ini akan memenuhi kapasitas *file system* dan pada akhirnya

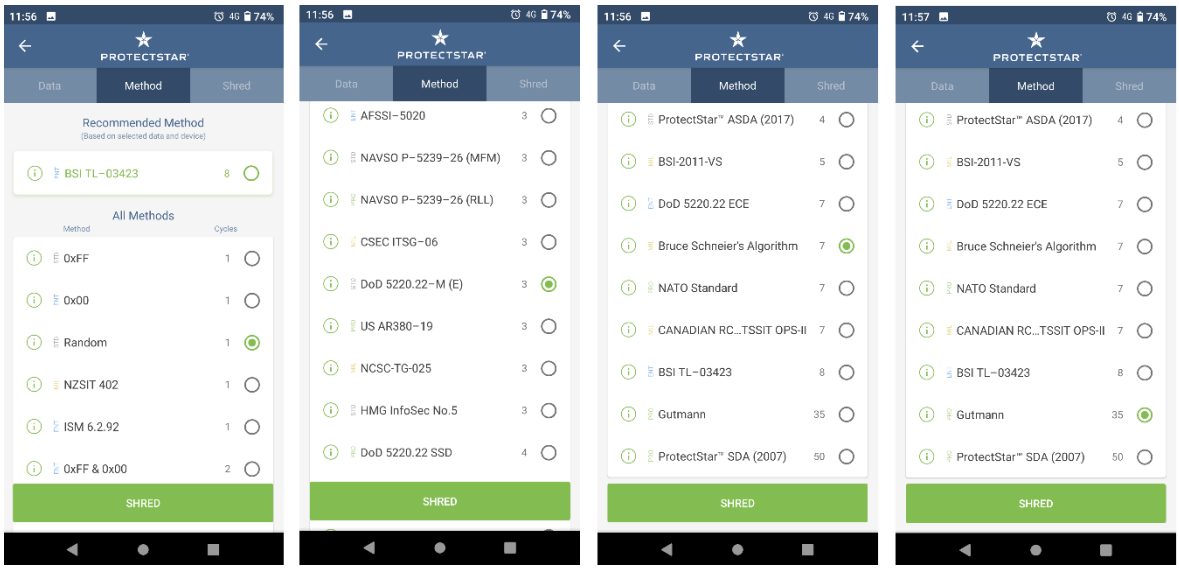
mengakibatkan *tools* forensik menjadi *crash* atau berhenti bekerja. Sebuah *compression bomb* dibuat dengan sangat hati-hati supaya tidak terdeteksi, dan dibuat dengan kedalaman kompresi yang berlapis-lapis.

Ketika *tools* forensik menemukan *bomb* tersebut yang dikira hanya *file* zip biasa, kemudian mencoba untuk membukanya, maka yang terjadi adalah diperlukan waktu yang lama bagi *tools* forensik untuk membukanya dikarenakan kedalam kompresi yang berlapis-lapis tadi. Selain waktu dekompresi yang sangat lama, ukurannya yang sangat besar juga dapat membuat komputer yang menjadi lokasi tempat *file* zip tersebut dibuka menjadi kehabisan ruang. Dan jika hal ini dilakukan di komputer milik tersangka, maka dapat menyapu bukti-bukti digital lain yang ada di dalam *file system*.

2.5 *Disk Overwrite*

Seperti yang telah dijelaskan sebelumnya bahwa ada banyak teknik anti forensik, namun teknik yang tertua dan paling banyak digunakan adalah *tool* untuk melakukan *overwrite* pada data dan informasi yang tersimpan (Garfinkel, 2007). *Tool* semacam ini relatif mudah digunakan dan mendukung banyak sistem operasi. Salah satu *tool* yang ada di pasaran dan dipilih untuk penelitian ini adalah iShredder. *Tool* buatan Protectstar, sebuah perusahaan yang berkecimpung di bidang keamanan siber ini memiliki sejumlah fitur penghapusan atau *disk wiping* yang cukup lengkap.

iShredder merupakan perangkat lunak penghapusan data yang paling populer dan dapat berjalan di beberapa sistem operasi seperti, iOS, Android, Windows, Mac, serta Windows Server (Protectstar, 2022). Beberapa fitur yang dimiliki *tool* ini antara lain adalah penghapusan *free space* dan penghapusan *file* atau *folder* tertentu sesuai pilihan. Kemudian fitur lainnya adalah banyaknya pilihan algoritma penghapusan yang dapat dipilih dengan total ada 26 pilihan algoritma. Sayangnya tidak semua algoritma yang ada dapat digunakan secara gratis. Gambar 2.4 a menunjukan *tool* yang digunakan berikut algoritma penghapusan yang dipilih.



Gambar 2.4 a *Tool* yang digunakan berikut algoritma penghapusan yang dipilih.

Dalam penelitian ini digunakan 4 algoritma berdasarkan jumlah *overwriting* atau *cycles* yang dilakukan. Keempatnya dipilih untuk mewakili banyaknya algoritma yang ada, antara lain 1 *cycle* dengan *random character*, 3 *cycles* milik DoD, 7 *cycles* milik Bruce Schneier, dan 35 *cycles* milik Peter Gutmann. Metode atau algoritma 1 *cycle* dengan *random character* dilakukan dengan menuliskan banyak karakter acak pada media penyimpanan untuk menimpa data lama yang mungkin masih tersisa. Beberapa hasil studi menunjukan bahwa *single wipe procedure* sudah cukup untuk membuat segala macam bentuk pemulihan data menjadi tidak mungkin (Wani et al., 2020).

Kemudian untuk 3 *cycles* milik DoD yang lekat penggunaannya oleh Departemen Pertahanan Amerika Serikat dalam hal penghapusan informasi penting, bekerja dengan cara melakukan *overwriting* semua data dalam tiga proses yang terpisah. Pertama dengan menimpa data yang ada menggunakan banyak karakter *zero* atau 0, setelah itu dilanjutkan dengan menggunakan banyak karakter *one* atau 1 dan yang terakhir menggunakan karakter acak (Sipicorp, 2022).

Selanjutnya untuk metode ketiga yaitu 7 *cycles* milik Bruce Schneier dan digunakan oleh Badan Keamanan Nasional Amerika Serikat (NSA) sebagai standar dalam penghapusan data atau informasi digital. Algoritma yang terdiri dari 7 *cycles* tersebut bekerja dengan cara menimpa data lama pada memori menggunakan banyak karakter *one* atau 1 pada *cycle* pertama. Lalu dilanjutkan dengan menggunakan banyak karakter *zero* atau 0 pada *cycle* kedua. Kemudian untuk lima *cycles* selanjutnya, data lama ditimpa dengan data acak (Eraser, 2020).

Lalu yang terakhir adalah metode Gutmann 35 *cycles.* Metode ini merupakan yang terlama untuk dilakukan *(time consuming)* jika dibandingkan dengan metode-metode lain yang telah dijelaskan sebelumnya. Hal ini disebabkan oleh banyaknya *cycles* atau langkah *overwriting* yang dilakukan. Detail langkah yang dimaksud dapat dilihat pada Gambar 2.4 b.

2.6 *File Carving*

Menurut (Ashraf, 2012) f*ile carving* merupakan teknik yang memanfaatkan informasi internal dari struktur dan konten suatu *file* yang telah terhapus untuk tujuan pemulihan. *File carving* tidak bergantung pada *filesystems* dan dapat memulihkan *files* dari *raw dataset*. *Carving* biasanya digunakan untuk memulihkan *files* yang berada di *unallocated space*. Dimana area ini tidak memiliki informasi metadata apapun yang merujuknya di dalam *filesystem.*

2.6.1 *Magic Numbers*

Sebagian besar format *file* menggunakan nilai konstan yang disebut *magic numbers* atau juga biasa disebut *file signatures* yang unik dan berkaitan erat dengan format *file* tersebut sertu berguna untuk membedakannya dengan format *file* lainnya (Ashraf, 2012). *File signatures* biasanya digunakan untuk mengindikasikan awal (*Header*) dan akhir (*Footer*) dari suatu *file*. Sebagai contoh *file* dengan format JPG memiliki *header* “0XFFD8” dan *footer* “0XFFD9”.

Overwrite Data				
Pass No.	**Data Written**	**Encoding Scheme Targeted**		
1	Random			
2	Random			
3	Random			
4	Random			
5	01010101 01010101 01010101 0x55	(1,7) RLL		MFM
6	10101010 10101010 10101010 0xAA	(1,7) RLL		MFM
7	10010010 01001001 00100100 0x92 0x49 0x24		(2,7) RLL	MFM
8	01001001 00100100 10010010 0x49 0x24 0x92		(2,7) RLL	MFM
9	00100100 10010010 01001001 0x24 0x92 0x49		(2,7) RLL	MFM
10	00000000 00000000 00000000 0x00	(1,7) RLL	(2,7) RLL	
11	00010001 00010001 00010001 0x11	(1,7) RLL		
12	00100010 00100010 00100010 0x22	(1,7) RLL		
13	00110011 00110011 00110011 0x33	(1,7) RLL	(2,7) RLL	
14	01000100 01000100 01000100 0x44	(1,7) RLL		
15	01010101 01010101 01010101 0x55	(1,7) RLL		MFM
16	01100110 01100110 01100110 0x66	(1,7) RLL	(2,7) RLL	
17	01110111 01110111 01110111 0x77	(1,7) RLL		
18	10001000 10001000 10001000 0x88	(1,7) RLL		
19	10011001 10011001 10011001 0x99	(1,7) RLL	(2,7) RLL	
20	10101010 10101010 10101010 0xAA	(1,7) RLL		MFM
21	10111011 10111011 10111011 0xBB	(1,7) RLL		
22	11001100 11001100 11001100 0xCC	(1,7) RLL	(2,7) RLL	
23	11011101 11011101 11011101 0xDD	(1,7) RLL		
24	11101110 11101110 11101110 0xEE	(1,7) RLL		
25	11111111 11111111 11111111 0xFF	(1,7) RLL	(2,7) RLL	
26	10010010 01001001 00100100 0x92 0x49 0x24		(2,7) RLL	MFM
27	01001001 00100100 10010010 0x49 0x24 0x92		(2,7) RLL	MFM
28	00100100 10010010 01001001 0x24 0x92 0x49		(2,7) RLL	MFM
29	01101101 10110110 11011011 0x6D 0xB6 0xDB		(2,7) RLL	
30	10110110 11011011 01101101 0xB6 0xDB 0x6D		(2,7) RLL	
31	11011011 01101101 10110110 0xDB 0x6D 0xB6		(2,7) RLL	
32	Random			
33	Random			
34	Random			
35	Random			

Gambar 2.4 b 35 *cycles* milik Peter Gutmann (Gutmann, 1996).

2.6.2 Metode *File Carving*

Terdapat banyak sekali teknik *file carving* yang dikembangkan dan dapat dikelompokkan menjadi generasi pertama, kedua, dan ketiga (Ashraf, 2012):

Generasi Pertama

Metode ini menggunakan informasi mengenai *file signatures* untuk memulihkan *files* terhapus atau biasa disebut metode *header-footer*. Cara kerja metode ini secara garis besar adalah mencari *header* dari suatu *file* kemudian secara berurutan mencari *footer*-nya. *Footer* pertama yang ditemukan akan menjadi akhir dari *file* tersebut. Kemudian lokasi dari *header* dan *footer* tadi akan ditandai untuk selanjutnya nilai heksadesimal di antara kedua lokasi tadi akan disalin dan disimpan menjadi sebuah *file*.

Selain seperti yang dijelaskan di atas, juga terdapat metode lain pada generasi pertama yang sedikit berbeda, yaitu metode *header-maximum file size*. Biasanya metode ini digunakan untuk memulihkan *files* yang hanya memiliki *header* tanpa *footer*. Cara kerjanya mirip dengan metode *header-footer*, jadi hal pertama yang dicari adalah nilai *header*-nya untuk kemudian lokasinya ditandai. Selanjutnya melihat informasi ukuran *file* pada *header* tersebut, jika tidak ada maka akan menggunakan ukuran yang ditebak sesuai dengan perkiraan yang terukur dan berdasar. Dari sini akan terlihat perkiraan titik akhir dari *file* tersebut untuk ditandai. Langkah selanjutnya menyimpan nilai heksadesimal dari kedua titik tadi menjadi sebuah *file*.

Kedua metode pada generasi pertama ini hanya bisa bekerja ketika *file* yang dipulihkan tersimpan pada blok memori yang berdekatan dan tidak terpisah atau terfragmen, serta *file signatures*-nya tidak *corrupt*. Metode-metode pada generasi pertama ini diklaim sangat cepat namun banyak menghasilkan *false positives*.

Generasi Kedua

Metode atau teknik *carving* di generasi kedua menggunakan lebih banyak informasi dari struktur internal suatu *file* untuk mengurangi *false positives*. Metode ini biasa disebut sebagai *file structure-based carving* selain itu juga sering disebut sebagai *semantic carving* atau *deep carving*. Cara kerja dari teknik ini adalah pertama dengan mengidentifikasi informasi tertentu dari suatu format *file*. Informasi yang didapatkan kemudian dicocokkan dengan *raw data* untuk mengidentifikasi suatu *file*. Penggunaan informasi-informasi tadi akan mengurangi *false positives* secara signifikan. Namun tetap saja, metode generasi kedua ini tidak bisa menangani *files* yang terfragmen.

Generasi Ketiga

Metode pemulihan di generasi ketiga dikembangkan dengan tujuan untuk menangani *files* yang terfragmen. Secara garis besar metode generasi ketiga bekerja dengan cara membaca

tiap individu blok di dalam data set dan menganalisis kontennya untuk mengetahui blok tersebut milik *file* yang mana. Blok-blok tadi jika diperlukan akan disusun ulang untuk membentuk *file* aslinya. Teknik atau metode tersebut biasa dikenal sebagai *block content-based carving*.

2.7 PhotoRec

Berdasarkan (Grenier, 2019), PhotoRec adalah sebuah perangkat lunak pemulihan data dan *files* yang dirancang untuk memulihkan *files* yang hilang, termasuk di dalamnya video, dokumen, dan *archives* dari *hard disks*, CD-ROMs, serta gambar-gambar yang terhapus dari memori kamera digital. PhotoRec juga dapat dijalankan di hampir semua sistem operasi meliputi:

a. DOS/Windows 9x
b. Windows 10/8.1/8/7/Vista/XP, Windows Server 2016/2012/2008/2003
c. Linux
d. FreeBSD, NetBSD, OpenBSD
e. Sun Solaris
f. Mac OS X, serta dapat dikompile di hampir semua sistem Unix.

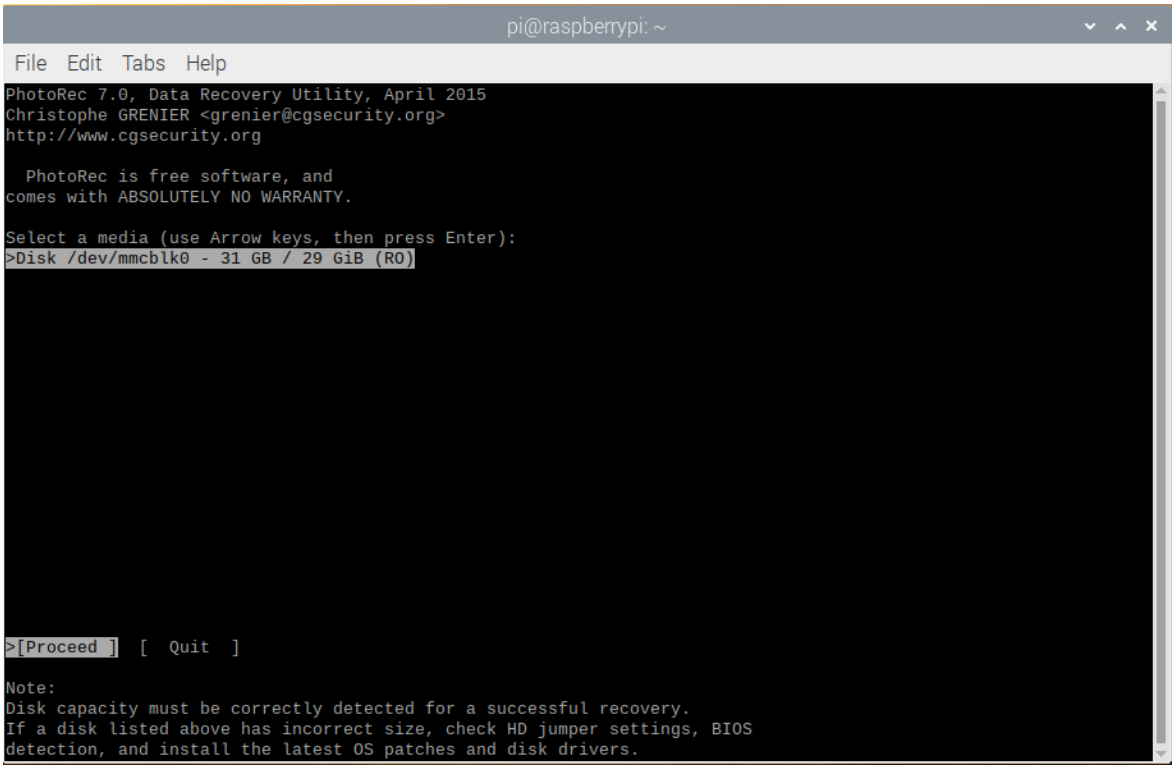
Dalam melakukan pemulihan *files* PhotoRec mengabaikan *file system* yang digunakan, dengan begitu pemulihan tetap dapat dilakukan meskipun *file system*-nya rusak parah. Pemulihan *files* dapat dilakukan oleh PhotoRec dari beberapa *file system* berikut:

a. FAT
b. NTFS
c. exFAT
d. ext2/ext3/ext4 *file system*
e. HFS+

File system ReiserFS tidak masuk dalam daftar di atas, sebab ReiserFS memiliki cara yang berbeda dalam menyimpan data alih-alih menyimpan data di manapun di dalam *disk* dan mencatat alamat lokasinya seperti kebanyakan *file system*.

Secara garis besar PhotoRec bekerja dengan cara mencoba mencari ukuran blok data atau ukuran *cluster*. Jika *file system*-nya tidak *corrupt* maka nilai ini dapat dibaca di *superblock* (ext2/ext3/ext4) atau *volume boot record* (FAT, NTFS). Jika tidak ditemukan, maka PhotoRec akan membaca media penyimpanan, *sector by sector*, mencoba mencari 10 *files* pertama, yang nantinya dari sini akan dikalkulasi ukuran *block/cluster* dari lokasinya. Setelah ukuran bloknya diketahui, PhotoRec akan membaca media penyimpanan *block by*

block (cluster by cluster). Setiap blok dicek berdasarkan database *signatures* yang disematkan dalam program. Ketika PhotoRec menemukan sebuah *file* dan memulihkannya, maka proses pencarian akan dihentikan untuk kemudian mengecek konsistensinya jika memungkinkan lalu menyimpannya menjadi sebuah *file* dengan ekstensi sesuai *signature* yang ditemukan. Tampilan antarmuka PhotoRec dapat dilihat pada Gambar 2.7.



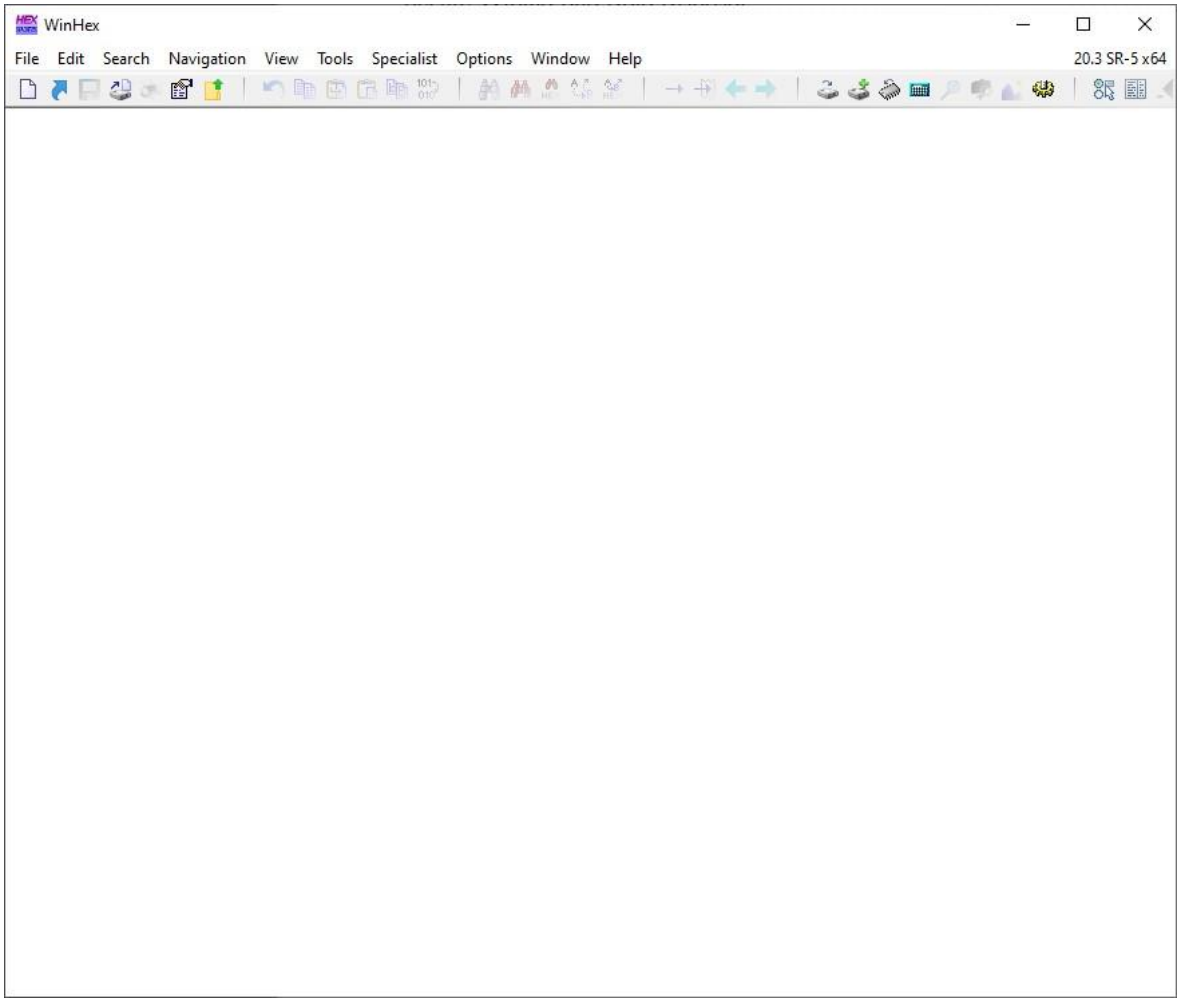
Gambar 2.7 Tampilan antarmuka PhotoRec.

2.8 WinHex

WinHex sejatinya merupakan *hex editor tool* yang memiliki banyak sekali fungsi. Salah satunya adalah fungsi untuk pemulihan *files*. Menurut manualnya (Fleischmann, 2021), t*ool* ini dapat dijalankan di Windows 7, Windows 8/8.1/Server 2012, Windows 10/Server 2016 baik yang 32-bit maupun 64-bit. Besar kemungkinan juga dapat dijalankan di Windows XP, Windows Server 2003, Windows Vista/Server 2008. Beberapa fungsionalitasnya juga dapat berjalan di lingkungan Wine pada Linux.

Dalam melakukan pemulihan *files*, secara *default* WinHex akan mencari *file headers* di ujung *cluster* yang biasanya merupakan tempat *file system* meletakannya. Begitu juga ketika pemulihan dilakukan di media fisik atau di *raw file* dimana *cluster layout* tidak didefinisikan, WinHex akan tetap mencari di ujung sektor. Selanjutnya adalah menentukan

atau memperkirakan ukuran asli dari suatu *file* yang dipulihkan. Dalam hal ini WinHex menggunakan berbagai algoritma internal. Salah satunya adalah dengan mencari *footer*-nya. Namun jika ternyata *file* tersebut tidak memiliki *footer*, maka penentuan atau perkiraan ukuran *file* yang dipulihkan akan mengikuti ukuran yang telah didefinisikan di *database* bawaan WinHex. Pembatasan ukuran *file* juga dijabarkan di dalam *database* WinHex, untuk mencegah pencarian ujung *file* di seluruh *volume* media, yang mana akan sangat memakan waktu jika *volume* medianya sangat besar. Tampilan antarmuka WinHex dapat dilihat pada Gambar 2.8.



Gambar 2.8 Tampilan antarmuka WinHex.

2.9 Android

Berdasarkan keterangan di situs resminya (Android, 2022), Android adalah sebuah sistem operasi *open source* untuk perangkat *mobile* dan merupakan proyek *open-source* yang diinisiasi dan dipimpin oleh Google. Sebagai sebuah proyek *open source*, tujuan yang ingin

dicapai Android adalah menghindari segala kesalahan titik pemusatan dimana satu pemain dalam industri dapat membatasi atau mengontrol usaha-usaha inovasi dari para pemain lain.

Seperti yang diketahui oleh banyak orang, bahwa struktur *file system* perangkat Android memiliki kesamaan yang cukup mencolok dengan Linux, hal ini dikarenakan Android dibangun di atas kernel Linux. Di dalam *file system* perangkat Android, biasanya terdapat enam partisi utama. Walaupun terkadang ada tambahan beberapa partisi pada model-model perangkat yang berbeda namun umumnya enam partisi utama ini selalu ada di setiap perangkat Android (KL, 2022). Detail dari enam partisi utama tersebut adalah sebagai berikut:

/boot

Partisi ini terdiri dari kernel Android dan ramdisk. Pada dasarnya segala hal yang dibutuhkan sebuah perangkat Android untuk *booting* ketika dinyalakan tersimpan di partisi ini.

/system

Partisi *system* merupakan lokasi tersimpannya data seluruh sistem operasi Android. Termasuk di dalamnya Andoid GUI beserta data dari aplikasi yang datang sebagai *pre-installed apps*.

/recovery

Partisi ini didesain untuk *backup* dan digunakan sebagai pilihan alternatif untuk *booting*. Adanya partisi ini memungkinkan perangkat melakukan *backup* data, menghapus data, mengembalikan perangkat ke setelan pabrik, dan melakukan operasi perawatan.

/data

Juga dikenal sebagai *user data partition*, partisi ini berisi semua data milik pengguna, termasuk di dalamnya daftar kontak, setelan, aplikasi, dan pesan. Menghapus partisi data akan membuat perangkat ter-*reset* ke setelan pabrik, karena akan membuat aplikasi, pesan, dan setelan pengguna terhapus.

/cache

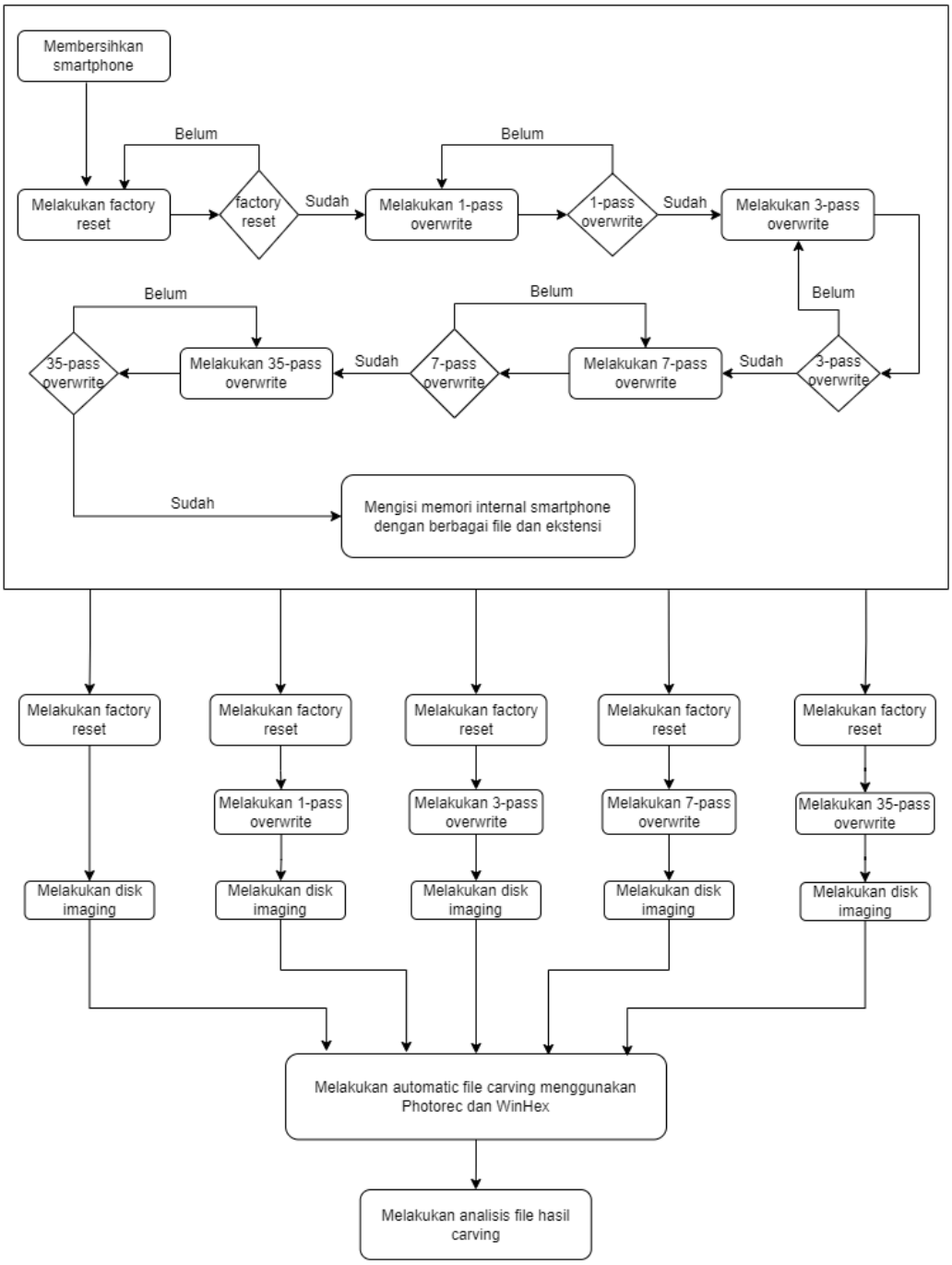
Merupakan tempat menyimpan data aplikasi dan komponen-komponen yang rutin diakses. Penghapusan isi partisi cache hanya akan memberikan beberapa ruang kosong yang nantinya cache yang dihapus tadi akan terbentuk kembali sejalan dengan penggunaan perangkat.

/misc

Berisi beragam setelan milik sistem, termasuk di dalamnya *region ID*, konfigurasi USB, beberapa setelan *hardware* tertentu. Partisi misc sangat penting karena dapat menyebabkan beberapa fitur perangkat mengalami malfungsi jika partisi ini *corrupt* atau hilang.

BAB 3
Metodologi

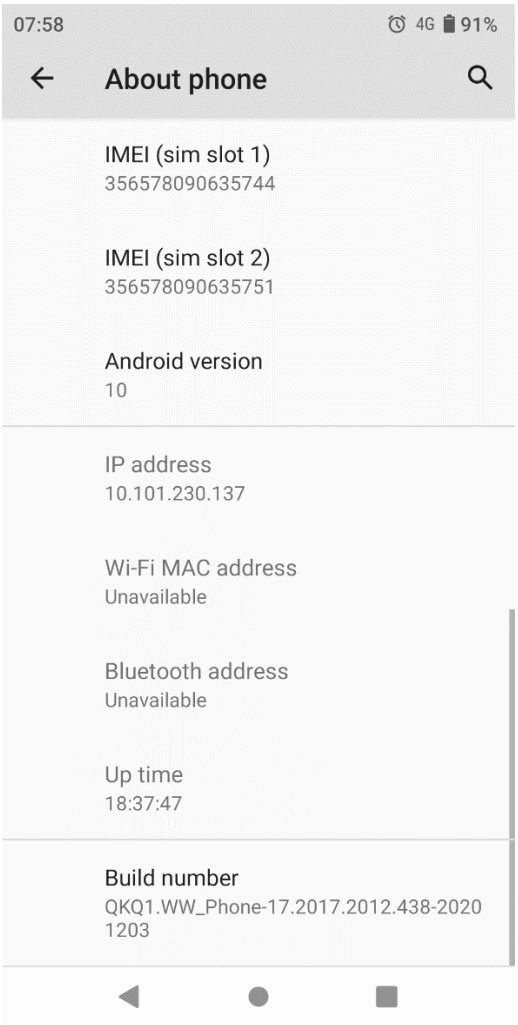
Pada bab ini dijelaskan mengenai urutan atau langkah-langkah yang dilakukan dalam penelitian ini. Secara keseluruhan alur penelitian dapat dilihat pada gambar 3.



Gambar 3 Alur penelitian.

3.1 *Rooting*

Terdapat satu langkah pendahuluan sebelum masuk pada alur penelitian di atas, yaitu proses *rooting*. Langkah ini diperlukan untuk memperoleh akses penuh terhadap perangkat Android. Akses penuh ini perlu didapatkan sebelum langkah *physical acquisition* atau *disk imaging* dilakukan. Metode *rooting* bisa berbeda-beda untuk tiap-tiap perangkat *smartphone* Android. Dalam penelitian ini perangkat telepon pintar yang menjadi objek penelitian adalah ASUS Zenfone Max Pro M1 dengan Android 10 sebagai sistem operasinya. Perangkat ini memiliki spesifikasi chipset Qualcomm Snapdragon 636, *internal storage* 32GB, dan RAM 3GB. Gambar 3.1 a menunjukan tangkapan layar versi Android dari perangkat yang digunakan.



Gambar 3.1 a Versi Android dari perangkat yang menjadi objek penelitian.

Proses *rooting* untuk perangkat tersebut dilakukan dengan melakukan *unlock bootloader* terlebih dahulu, yang kemudian dilanjutkan dengan memasang atau melakukan

flashing sebuah *custom recovery* bernama TWRP (Team Win Recovery Project) ketika perangkat dalam mode Fastboot. Setelah itu diikuti dengan memasang aplikasi Magisk melalui TWRP. Gambar 3.1 b berikut ini menunjukan tampilan Fastboot pada perangkat.

Gambar 3.1 b Tampilan mode Fastboot pada perangkat yang menjadi objek penelitian.

3.2 Membersihkan *Smartphone*

Dimulai dari langkah inilah alur penelitian di atas dilakukan. Langkah pertama adalah mengembalikan *smartphone* ke setelan pabrik, lalu di-*overwrite* menggunakan keempat algoritma *overwriting* yang telah dipilih secara bergantian untuk memastikan *free space* pada memori internalnya benar-benar bersih dan kosong.

3.3 Pengisian Data *Dummy*

Smartphone yang telah dibersihkan memori internalnya kemudian diisi dengan data *dummy* yang telah disiapkan. Data *dummy* tersebut terdiri dari berbagai macam *files* dengan berbagai macam ekstensi yang dipilih karena dianggap cukup populer serta banyak digunakan, antara lain:

1. DOCX
2. XLSX
3. PPTX
4. PDF
5. JPG
6. PNG
7. AVI
8. MP4
9. MKV
10. MP3

Dari setiap ekstensi di atas, terdapat dua *dummy files* yang diisikan ke dalam *internal storage.* Kecuali untuk *files* gambar atau foto dengan ekstensi JPG, selain dua yang telah diisikan sebelumnya, juga terdapat 1941 *files* lain yang berasal dari aplikasi Whatsapp. Dimana secara *default* juga disimpan di dalam *internal storage.*

3.4 Penghapusan Data dan Akuisisi

Langkah berikutnya, perangkat telepon pintar yang telah diisi dengan data *dummy* dihapus datanya menggunakan salah satu teknik penghapusan data yang menjadi objek dari penelitian. Khusus untuk teknik anti-forensik *disk overwrite*, dilakukan *factory reset* terlebih dahulu terhadap perangkat sebelum teknik-teknik tersebut dijalankan. Setelah selesai, yang dilakukan selanjutnya adalah mengakuisisi *internal storage* dari *smartphone* menggunakan teknik *physical acquisition* atau *disk imaging*. *Tool* yang digunakan untuk melakukan proses akuisisi fisik pada penelitian ini adalah *Disk Dump* yang menghasilkan *file images* dengan ekstensi .dd dan dijalankan di sistem operasi Android yang berbasis Linux.

Langkah yang dijabarkan pada poin 3.2, 3.3, dan 3.4 diulang kembali sesuai dengan yang terlihat di gambar alur penelitian di atas. Hasilnya diperoleh 5 *disk images* yang berasal dari 1 hasil *factory reset* dan 4 teknik *overwriting*. Kelima *disk images* tersebut kemudian coba dipulihkan dan dianalisis data-datanya menggunakan *tool* forensika digital WinHex

yang bekerja dengan metode pada generasi pertama yaitu *header-footer* dan *header-maximum file size*. Juga akan digunakan *tool* PhotoRec yang prinsip kerjanya menggunakan metode *file structured based* yang termasuk pada teknik *file carving* generasi kedua.

Sedangkan untuk metode *carving* generasi ketiga, yaitu *block content-based carving* tidak digunakan dalam penelitian ini dengan alasan yang mengacu pada beberapa literatur, bahwa metode *carving* generasi ketiga, teknik atau algoritma yang digunakan terbilang rumit dan membutuhkan bantuan AI. Terdapat perusahaan yang menyediakan layanan *carving* generasi ketiga yang tentu saja tidak gratis. Berdasarkan alasan tersebut, bisa dikatakan metode generasi ketiga tidak dapat dengan mudah diakses oleh sembarang orang, jadi ancaman privasi dari penggunaan metode pemulihan *file* generasi ketiga tidaklah besar.

Hasil pemulihan kemudian dianalisis dan dibandingkan untuk melihat teknik mana yang paling efektif dan efisien dalam hal penghapusan data. Hal-hal yang dibandingkan dan dijadikan parameter keefektifan antara lain adalah banyaknya artefak digital atau *files* yang dapat dipulihkan lagi dengan ekstensi seperti yang telah disebutkan sebelumnya. Namun hal ini juga akan dilihat lebih jauh, apakah data atau *files* tersebut berkaitan dengan privasi si pemilik perangkat atau tidak. Dan dalam penelitian ini data privasi milik pengguna diwakili oleh data *dummy* yang sengaja dimasukan. Jadi dalam hal efektifitas, semakin sedikit data *dummy* yang dapat dipulihkan menjadi tolok ukur sebagai teknik penghapusan yang paling efektif. Selanjutnya dalam hal efisiensi, penelitian ini membandingkan waktu total yang diperlukan untuk menjalankan masing-masing teknik penghapusan yang dibandingkan dengan banyaknya data *dummy* yang berhasil dipulihkan. Semakin kecil angka perbandingannya menjadi parameter penentu sebagai teknik yang paling efisien. Jika dituliskan, rumus penghitungan efisiensi yang dimaksud adalah sebagai berikut.

$$\text{Nilai efisiensi} = \frac{\text{Jumlah artefak } (artefak)}{\text{Waktu penghapusan } (detik)} \tag{3.1}$$

BAB 4
Hasil dan Pembahasan

Bab ini berisi ulasan detail mengenai temuan-temuan sebagai hasil dari penelitian yang telah dilakukan sesuai dengan alur penelitian pada bab tiga.

4.1 *Rooting*

Ketika proses *rooting* dilakukan, tepatnya pada proses *unlock bootloader* ditemukan hasil bahwa perangkat akan ter-*reset* ke setelan awal atau setelan pabrik. Hal ini merupakan catatan penting bagi para praktisi forensika digital yang akan melakukan eksaminasi pada perangkat yang sama dengan objek pada penelitian ini. Sebab tentu saja perangkat yang ter-*reset* ke setelan awal, data di dalamnya akan berubah, sehingga integritasnya dipertanyakan.

4.2 Penghapusan Data

Data *dummy* yang telah diisikan ke dalam *internal storage* perangkat telepon pintar, dihapus menggunakan beberapa teknik penghapusan data yang menjadi objek penelitian. Kemudian untuk lamanya operasi penghapusan data masing-masing teknik dapat dilihat pada tabel 4.2. Data pada tabel 4.2 berikut didapat dengan melakukan penghitungan menggunakan alat ukur waktu, yaitu *stopwatch*. Penghitungan dimulai dari awal dijalankannya suatu operasi penghapusan hingga selesai.

Tabel 4.2 Lama waktu operasi penghapusan data.

Metode Penghapusan Data	Waktu operasi penghapusan data	Total waktu operasi penghapusan data
Factory Reset	2 menit 49 detik	**2 menit 49 detik**
1-pass overwrite	5 menit 46 detik	**(2 menit 46 detik)** + (5 menit 46 detik)
3-pass overwrite	11 menit 47 detik	**(2 menit 46 detik)** + (11 menit 47 detik)
7-pass overwrite	23 menit	**(2 menit 46 detik)** + (23 menit)
35-pass overwrite	1 jam 42 menit 13 detik	**(2 menit 46 detik)** + (1 jam 42 menit 13 detik)

Dari tabel di atas pada kolom ‘Total waktu operasi penghapusan data’, dapat dilihat bahwa *factory reset* menduduki posisi teratas sebagai yang paling cepat dibandingkan keempat teknik *disk overwriting* yang lain. Salah satu penyebabnya adalah karena keempat metode penghapusan yang lain harus melalui fase pemulihan ke setelan pabrik terlebih dahulu sebelum dilakukan *overwriting.* Tujuannya adalah untuk menghilangkan *files* atau data milik pengguna terlebih dahulu dan hanya menyisakan *files* atau data bawaan perangkat.

Juga untuk membuat blok atau sektor pada memori yang sebelumnya ditempati oleh *files* atau data pengguna menjadi "kosong". Blok atau sektor "kosong" inilah yang selanjutnya akan di-*overwrite*. Hal ini berbeda dengan metode penghapusan *factory reset* yang hanya melalui satu fase saja yaitu pemulihan ke setelan pabrik sehingga prosesnya akan lebih cepat. Namun juga dapat dilihat pada kolom 'Waktu operasi penghapusan data', yang menunjukan lamanya waktu operasi penghapusan data secara individu, *Factory reset* juga adalah yang tercepat.

4.3 Akuisisi dan *Hashing*

Proses pengumpulan data yang telah dilakukan, menghasilkan lima *disk images* dengan ekstensi .dd untuk kemudian dilakukan analisis terhadapnya. Proses analisis yang pertama dilakukan adalah dengan membandingkan nilai *hash* dari kelimanya. Nilai *hash* dapat diibaratkan sebagai sidik jari dari suatu *file* berupa nilai angka yang bersifat unik yang diperoleh melalui pemrosesan menggunakan algoritma kriptografi. Secara sederhananya suatu *file* yang memiliki konten sama pasti akan memiliki nilai *hash* yang sama pula, begitu juga sebaliknya (Trend Micro, 2022). *Hash value* dari kelima *images* tersebut dapat dilihat pada Tabel 4.3, yang datanya didapat dengan melakukan penghitungan nilai *hash* menggunakan *tool* bernama HashMyFiles.

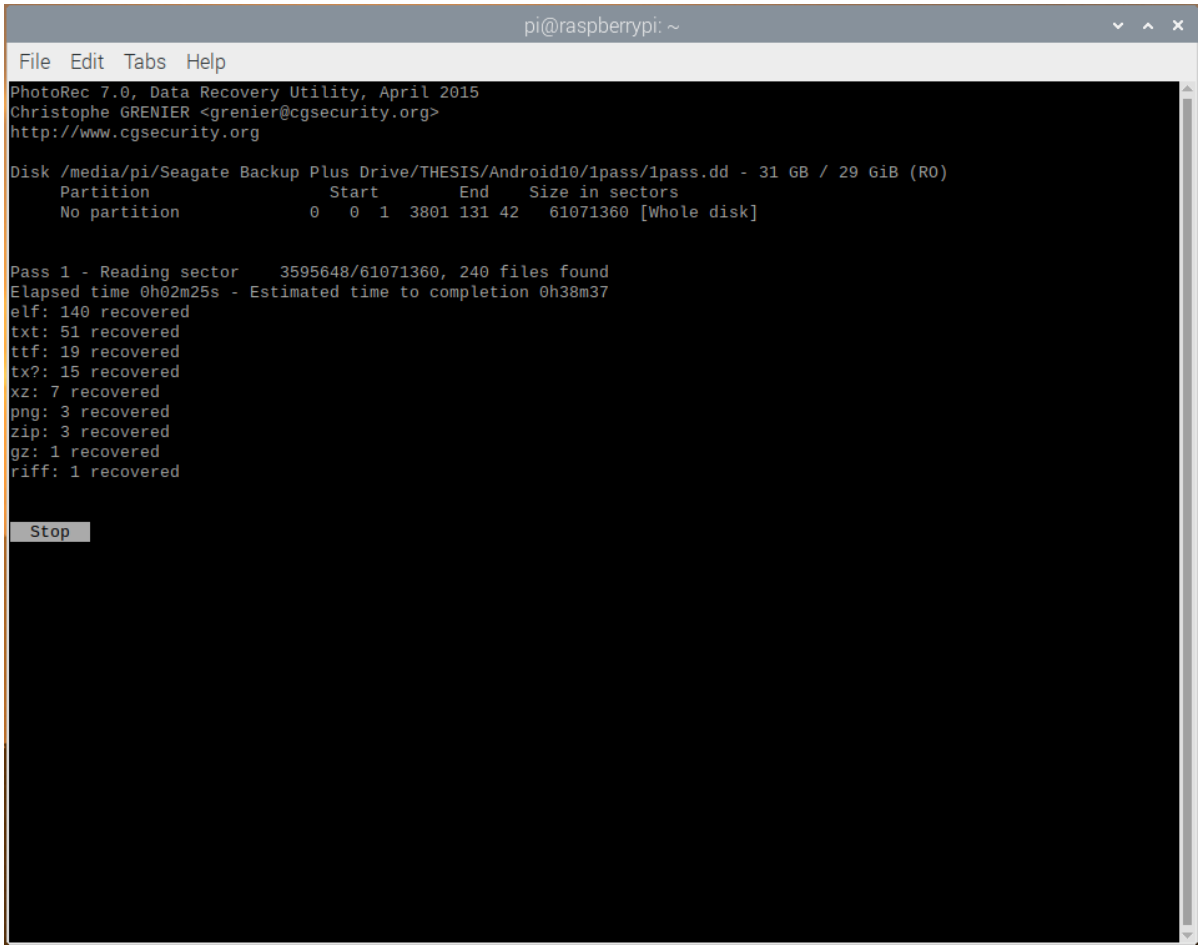
Tabel 4.3 Nilai *hash* dari *image files*.

File Images	MD5	SHA1
factory_reset.dd	3f59cd8f92c4f7e2cf9b6980c1297bc6	1f655ab831caf4a8f8496fbe5fc5e9d07c89e70a
1pass.dd	4912da0a4820b102c5646d34f7e6a6d2	6a3373b31ffc8c24520c2a8d7a0768bec6cbe24f
3passes.dd	2d97e161dd1f9977102c67e7987a11a4	9a0785c8ad1ae6a536e2515f30b85c7015ec5e91
7passes.dd	55d360e120d262d344fc0ca63d8fae96	4eec63d281dca67e835c56d5c8a51e1ad4dbee0c
35passes.dd	6858893e41fe631fc76f48a1fb170349	86975aac9f9e43502232567f55e398251bc7195b

Tabel 4.3 di atas menunjukan bahwa masing-masing *file images* memiliki nilai *hash* yang berbeda. Hal ini masuk akal, sebab kelimanya diproses menggunakan metode yang berbeda dan sudah pasti isi atau konten di dalamnya juga berbeda. Lain halnya jika nilai *hash* yang didapatkan sama semua, dimana hal ini bisa dijadikan indikator bahwa ada yang salah dengan pemrosesannya atau *tool* yang digunakan tidak berjalan dengan semestinya. Lalu untuk ukuran *file* hasil proses *imaging*, semuanya menunjukan ukuran yang sama yaitu 30.535.680 KB atau sama dengan 30.535 GB. Ukuran ini sesuai dengan ukuran yang ada pada *internal storage* perangkat yang digunakan dalam penelitian ini. Atau dengan kata lain keseluruhan *internal storage* dari perangkat dapat terakuisisi 100%.

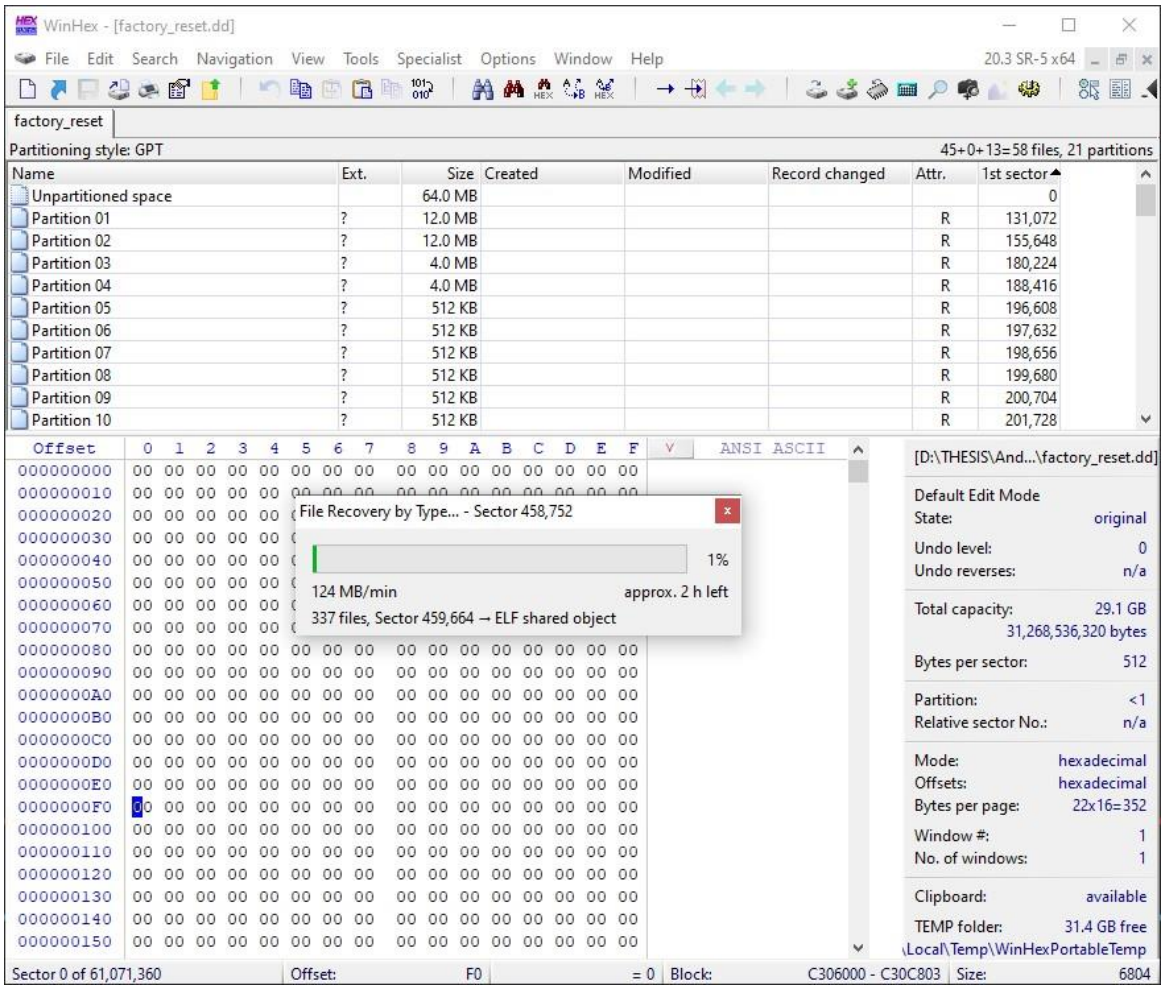
4.4 Pemulihan *Files*

Proses berikutnya setelah *imaging* adalah proses mengembalikan data yang telah terhapus menggunakan *recovery tool* bernama PhotoRec dan WinHex. Proses bagaimana kedua *tools* tersebut bekerja dapat dilihat pada gambar 4.4 a dan 4.4 b. Lalu untuk hasil dari proses *recovery*-nya sendiri secara detail dapat dilihat pada Tabel 4.4 a dan 4.4 b.



Gambar 4.4 a Proses *file recovery* oleh PhotoRec.

Gambar 4.4 a di atas menunjukan tampilan antarmuka *tool* PhotoRec yang berupa *command-line* ketika dalam proses pemulihan *files*. Di situ dapat dilihat *image file* yang menjadi target pemulihan adalah 1pass.dd dengan ukuran sebesar 31 GB. Ketika *screenshot* ini diambil, terlihat estimasi sisa waktu yang dibutuhkan adalah 38 menit 37 detik dan sebelumnya sudah berjalan 2 menit 25 detik, lalu totalnya sudah ada 240 *files* yang telah berhasil dipulihkan dengan rinciannya masing-masing berdasarkan ekstensinya.



Gambar 4.4 b Proses *file recovery* oleh WinHex.

Gambar 4.4 b di atas atau lebih tepatnya pada jendela kecil di bagian tengah gambar dengan tulisan "*File Recovery by Type…",* menunjukan tampilan ketika aplikasi WinHex sedang dalam proses memulihkan *files*. Di situ terlihat bahwa kecepatan pembacaan atau *scanning* terhadap *image file* yang menjadi objek pemulihan mencapai 124 MB/menit. Kemudian untuk estimasi waktu yang dibutuhkan untuk melakukan pemulihannya sendiri diperkirakan mencapai 2 jam.

Lalu di bagian belakang dari jendela kecil tadi, dapat dilihat keterangan detail mengenai *image file* yang sedang dianalisis. Terlihat nama dari *image file*-nya adalah factory_reset.dd. Kemudian terlihat pula nilai heksadesimal dari konten *image file* tersebut, serta total ukurannya.

Tabel 4.4 a Total *files* yang berhasil dipulihkan oleh PhotoRec.

	Total ukuran file hasil pemulihan	**Jumlah total file hasil pemulihan**
Factory Reset	10,2167 GB	1680
1-pass overwrite	9,7380 GB	1696
3-pass overwrite	9,7384 GB	1712
7-pass overwrite	9,7842 GB	1688
35-pass overwrite	9,7372 GB	1686

Tabel 4.4 b Total *files* yang berhasil dipulihkan oleh WinHex.

	Total ukuran file hasil pemulihan	**Jumlah total file hasil pemulihan**
Factory Reset	15,6331 GB	267.071
1-pass overwrite	15,6385 GB	265.726
3-pass overwrite	15,5822 GB	265.745
7-pass overwrite	15,5271 GB	267.055
35-pass overwrite	15,9176 GB	265.751

Kedua tabel di atas menunjukan hasil pemulihan *files* dari kedua *tool* yang digunakan. Data di dalamnya diperoleh dengan melakukan pengecekan menggunakan menu *Properties* pada *folder* di sistem operasi Windows. Hasil menunjukan bahwa WinHex mampu memulihkan sejumlah *files* yang jauh lebih banyak dibandingkan dengan dengan PhotoRec. Terlihat bahwa rata-rata jumlah *files* yang dapat dipulihkan WinHex dari lima metode penghapusan mencapai 266.270 *files*, sedangkan PhotoRec rata-rata hanya mampu memulihkan 1.692 *files* dari kelima metode penghapusan. Kemudian jika dilihat dari sisi total ukuran *files* hasil pemulihan, WinHex juga unggul dengan angka yang jauh lebih besar dari PhotoRec, yaitu rata-rata 9,8429 GB untuk PhotoRec dan 15,6597 GB untuk WinHex.

Metode pemulihan *file* generasi pertama, yaitu *header-footer* dan *header-maximum file size* yang digunakan oleh WinHex terbukti ampuh untuk mendapatkan lebih banyak *files*. Meskipun banyak juga *files* hasil pemulihan WinHex yang pada praktiknya tidak bisa dibuka menggunakan aplikasi atau program yang biasa digunakan untuk membuka *files* dengan ekstensi tersebut. Atau dengan kata lain metode *carving* generasi pertama yang digunakan WinHex banyak menghasilkan *false positive*. Berbeda dengan metode *carving* generasi kedua yang digunakan oleh PhotoRec, yang meskipun hanya mampu memulihkan lebih sedikit *files* secara jumlah, namun hampir semua *files*-nya adalah "*real file*". Jadi jika dilihat dari sisi *false positive* yang dihasilkan, bisa dibilang dapat dikurangi secara signifikan oleh metode pemulihan generasi kedua milik PhotoRec.

4.5 Analisis *Files* Hasil *Carving*

Pada tahap ini, *files* hasil pemulihan yang berasal dari Tabel 4.4 a dan 4.4 b disaring dan hanya diambil *files* yang memiliki ekstensi sesuai dengan objek pada penelitian. Hasilnya dapat dilihat pada tabel 4.5 a dan 4.5 b di bawah ini.

Tabel 4.5 a Detail *files* hasil pemulihan WinHex.

Metode Penghapusan	Ekstensi File										Jumlah
	.docx	.xlsx	.pptx	.pdf	.jpg	.png	.avi	.mp4	.mkv	.mp3	
Factory reset	-	-	-	11	310	76.272	2	13	57	36	76.701
1-pass overwrite	-	-	-	11	310	76.017	2	13	57	36	76.446
3-pass overwrite	-	-	-	11	310	76.017	2	13	57	36	76.446
7-pass overwrite	-	-	-	11	310	76.272	2	13	57	36	76.701
35-pass overwrite	-	-	-	11	310	76.017	2	13	57	36	76.446

Tabel 4.5 b Detail *files* hasil pemulihan PhotoRec.

Metode Penghapusan	Ekstensi File										Jumlah
	.docx	.xlsx	.pptx	.pdf	.jpg	.png	.avi	.mp4	.mkv	.mp3	
Factory reset	-	-	-	-	26	17	-	2	-	2	47
1-pass overwrite	-	-	-	-	26	17	-	2	-	2	47
3-pass overwrite	-	-	-	-	26	17	-	2	-	2	47
7-pass overwrite	-	-	-	-	26	17	-	2	-	2	47
35-pass overwrite	-	-	-	-	26	17	-	2	-	2	47

Kedua tabel di atas menunjukan perbedaan yang sangat mencolok, dimana WinHex mampu memulihkan 7 ekstensi dari 10 ekstensi yang menjadi objek penelitian, jumlah rata-rata dari kelima metode penghapusan adalah 76.548 *files*. Sedangkan PhotoRec hanya mampu memulihkan 4 ekstensi dengan jumlah rata-rata dari kelima metode penghapusan adalah 47 *files*, dimana setelah diteliti, semua *files* yang berhasil dipulihkan tidak ada satupun yang berhubungan dengan privasi pengguna. Begitu juga *files* hasil pemulihan WinHex, yang meskipun jauh lebih banyak 1.629 kali lipat dari hasil pemulihan PhotoRec namun tidak satupun berhubungan dengan privasi pengguna. Juga tidak satu pun dari *files* tersebut merupakan *dummy files* yang telah dimasukkan sebelumnya. Detailnya dapat dilihat pada tabel 4.5 c di bawah ini.

Tabel 4.5 c *Dummy files* yang berhasil dipulihkan oleh WinHex dan PhotoRec.

Metode Penghapusan	Ekstensi File										Jumlah
	.docx	.xlsx	.pptx	.pdf	.jpg	.png	.avi	.mp4	.mkv	.mp3	
Dummy files	2	2	2	2	1943	2	2	2	2	2	1961
Factory reset	-	-	-	-	-	-	-	-	-	-	-
1-pass overwrite	-	-	-	-	-	-	-	-	-	-	-
3-pass overwrite	-	-	-	-	-	-	-	-	-	-	-
7-pass overwrite	-	-	-	-	-	-	-	-	-	-	-
35-pass overwrite	-	-	-	-	-	-	-	-	-	-	-

Data yang tertampil di Tabel 4.5 c berasal dari hasil pemerikasaan manual terhadap data yang diperoleh dari Tabel 4.5 a dan 4.5 b. Hasil pemeriksaan menunjukan bahwa tidak ada satu pun data privasi milik pengguna yang dapat dikembalikan, dimana dalam penelitian ini diwakili oleh sejumlah *dummy files* yang sengaja dimasukkan ke dalam *internal storage* perangkat telepon pintar Android 10. Dari kedua *tools* yang digunakan, WinHex dan PhotoRec, keduanya menunjukan hasil yang sama, dari total 1961 *dummy files,* hasilnya nol atau tidak ada yang dapat dipulihkan sama sekali. Baik itu yang berasal dari teknik penghapusan *factory reset,* 1-*pass overwrite,* 3-*pass overwrite,* 7-*pass overwrite,* maupun 35-*pass overwrite.*

Hasil pemulihan *dummy files* yang menghasilkan angka nol atau nihil menunjukan bahwa, kelima teknik penghapusan yang diuji terbukti mampu menghapus data pribadi atau *dummy files* secara permanen. Namun yang perlu digarisbawahi sesuai dengan penjelasan pada sub bab 4.2 di atas, bahwa sebelum dijalankannya langkah *overwrite,* perangkat terlebih dahulu dikembalikan ke setelan pabrik. Jadi yang berperan besar dalam proses penghapusan data di sini adalah teknik *factory reset.*

Proses pengembalian ke setelan pabrik pada Android 10 di perangkat yang diuji, menampilkan tulisan *wiping* saat proses tersebut berlangsung. Jika mengacu pada salah satu literatur yang dipakai dalam penelitian ini, yaitu (Wani et al., 2020), menunjukan bahwa kata *wiping* juga merujuk pada *overwriting* atau menimpa data. Oleh sebab itu, hal ini masuk akal jika perangkat Android 10 yang dikembalikan ke setelan pabrik, data pribadi milik penggunanya akan terhapus secara permanen sebab telah ditimpa atau di-*overwrite* oleh sistem operasi.

Analisis selanjutnya yang berkaitan dengan efisiensi, dilakukan dengan melakukan penghitungan, membandingan antara banyaknya artefak yang berkaitan dengan data pribadi yang dapat dipulihkan dengan lamanya waktu operasi proses penghapusan. Dalam

penghitungan perbandingan ini, data yang digunakan berasal dari Tabel 4.5 c dan Tabel 4.2, yaitu jumlah *dummy files* yang dapat dipulihkan dibandingkan dengan lamanya waktu operasi penghapusan dari masing-masing teknik anti-forensik. Detail dari penghitungan tersebut adalah sebagai berikut:

A. *Factory Reset*

$$\text{Nilai efisiensi} = \frac{0 \text{ artefak}}{\mathbf{169 \text{ detik}}} = 0 \text{ artefak/detik}$$

B. *1-pass overwrite*

$$\text{Nilai efisiensi} = \frac{0 \text{ artefak}}{\mathbf{169 \text{ detik} + 346 \text{ detik}}} = 0 \text{ artefak/detik}$$

C. *3-pass overwrite*

$$\text{Nilai efisiensi} = \frac{0 \text{ artefak}}{\mathbf{169 \text{ detik} + 707 \text{ detik}}} = 0 \text{ artefak/detik}$$

D. *7-pass overwrite*

$$\text{Nilai efisiensi} = \frac{0 \text{ artefak}}{\mathbf{169 \text{ detik} + 1380 \text{ detik}}} = 0 \text{ artefak/detik}$$

E. *35-pass overwrite*

$$\text{Nilai efisiensi} = \frac{0 \text{ artefak}}{\mathbf{169 \text{ detik} + 6133 \text{ detik}}} = 0 \text{ artefak/detik}$$

Dari proses penghitungan di atas dapat dilihat bahwa pada teknik anti-forensik 1-*pass overwrite*, 3-*pass overwrite*, 7-*pass overwrite*, dan 35-*pass overwrite* terdapat 169 detik tambahan waktu pemrosesan. Hal ini dikarenakan, keempat teknik *overwriting* tersebut selalu didahului oleh *factory reset* sebelum dijalankan seperti yang telah dijelaskan pada Sub bab 4.2. Kemudian, dikarenakan oleh tidak adanya artefak atau *dummy files* yang dapat dipulihkan berdasarkan data yang diperoleh dari Tabel 4.5 c, maka hasil penghitungan pun menunjukan hasil 0 (nol/nihil). Untuk itu, penentuan teknik mana yang paling efisien selanjutnya dilakukan dengan melihat kecepatan proses penghapusan data. Teknik penghapusan dengan waktu tercepat dianggap sebagai teknik yang paling efisien dari sisi waktu, dalam hal ini adalah *factory reset.*

BAB 5
Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa ternyata *factory reset* sudah cukup untuk menghapus data pribadi di dalam perangkat telepon pintar dengan sistem operasi Android 10. Meskipun dari upaya untuk memulihkan data dan *files* yang telah terhapus menggunakan *tools* forensika populer, WinHex dan PhotoRec, yang bekerja menggunakan metode *carving* generasi pertama dan kedua membuahkan sejumlah *files*, namun setelah diteliti lebih jauh ternyata *files* tersebut merupakan bawaan *default* dari sistem operasi. Dan tidak ada satu pun *files* milik pengguna perangkat yang dapat dibangkitkan kembali.

Kemudian dari segi lama waktu operasinya, *factory reset* juga adalah yang tercepat dibandingkan dengan metode atau algoritma lain. Hal ini dikarenakan pada *factory reset* hanya dilakukan satu kali proses penghapusan saja yaitu *factory reset* itu sendiri. Sedangkan pada teknik penghapusan lain harus didahului oleh *factory reset* sebelum dijalankannya teknik-teknik penghapusan yang dimaksud. Kemudian, perbandingan antara waktu operasi dan banyaknya data privasi yang dapat dipulihkan dihitung, dimana hasilnya menunjukan angka 0 (nol). Hal ini disebabkan oleh tidak adanya artefak atau *dummy files* yang dapat dipulihkan. Oleh sebab itu proses penentuan efisiensi digeser dengan melihat teknik mana yang memiliki waktu penghapusan paling cepat. Maka dari itu, dapat dikatakan bahwa dalam hal penghapusan data pribadi untuk kepentingan menjaga privasi dari seorang pengguna telepon pintar Android 10, *factory reset* adalah yang paling efisien untuk digunakan, sedangkan dari sisi efektivitas kelima metode tersebut terbukti tidak memiliki perbedaan yang signifikan.

Temuan dari penelitian ini seperti yang telah diuraikan di atas dapat dijadikan referensi oleh para pengguna awam telepon pintar Android terutama dengan versi Android 10, ketika mereka berniat menghilangkan atau menghapus data pribadi mereka secara permanen. Fitur *factory reset* bawaan Android 10 terbukti ampuh untuk menyingkirkan data pribadi pengguna. Namun jika *users* atau para pengguna masih merasa belum aman hanya dengan *factory reset*, maka mereka dapat menambahkan teknik *overwriting* untuk memastikan data lama mereka benar-benar hilang.

Temuan lain dari penelitian ini adalah ketika proses *rooting* dijalankan, yang didahului dengan proses *unlock bootloader*, ternyata memiliki implikasi perangkat akan ter-*reset* atau dikembalikan ke setelan pabrik. Hal ini tentu sangat penting untuk diketahui oleh para praktisi forensika digital terutama yang baru terjun di bidang ini dan masih minim pengalaman. Ter-*reset*-nya perangkat tentu bukan hal baik dalam proses pengolahan barang bukti elektronik yang berupa telepon pintar Android, sebab hasil penelitian ini menunjukan bahwa perangkat yang ter-*reset* ke setelan pabrik ternyata data dan *files* non bawaan pabrik di dalamnya akan hilang permanen dan tidak dapat dipulihkan. Akibatnya tentu akan membuat proses forensika yang dilakukan menjadi terhambat atau bahkan tidak dapat dilakukan sama sekali.

5.2 Saran

Dalam penelitian ini, ketika proses pemulihan *files*, metode *carving* yang digunakan adalah metode generasi pertama dan generasi kedua. Dimana metode dari kedua generasi tersebut terbukti tidak mampu memulihkan data privasi pengguna atau *dummy files* yang digunakan dalam penelitian ini. Oleh sebab itu, penelitian lebih lanjut sangat menarik untuk dilakukan dengan memanfaatkan metode *carving* generasi ketiga. Hal ini diperlukan untuk memperkuat klaim bahwa *factory reset* sudah cukup efektif untuk menghapus data pribadi pengguna perangkat *smartphone* Android 10.

Penelitian lanjutan jangka panjang juga sangat diperlukan untuk melihat apakah hal yang sama sebagaimana temuan dalam penelitian ini, juga berlaku pada perangkat merek lain atau pada sistem operasi Android versi lain. Kumpulan hasil penelitian lanjutan yang dilakukan pada perangkat merek lain serta versi Android lain tersebut dapat dibuat menjadi semacam katalog yang secara garis besar berisi mengenai '*how to deal*' atau bagaimana cara menangani perangkat Android dengan tipe, merek, dan versi sistem operasi tertentu yang nantinya akan sangat berguna sebagai acuan bagi para praktisi forensika digital.

Daftar Pustaka

Afonin Oleg, Nikolaev Danil, & Gubanov Yuri. (2015). *Countering Anti-Forensic Efforts-Part 1*.

Android. (2022). *Android Open Source Project*. https://source.android.com/

Ashraf, M. N. (2012). *Forensic Multimedia File Carving*.

Chukwuemeka Ogazi-Onyemaechi, B., Dehghantanha, A., Kim-Kwang, ;, & Choo, R. (2017). *Performance of Android Forensics Data Recovery Tools*.

Eraser. (2020). *Appendix A: Erasure Methods – Eraser*. https://eraser.heidi.ie/appendix-a-erasure-methods/

Feng, P., Li, Q., Zhang, P., & Chen, Z. (2018). Logical acquisition method based on data migration for Android mobile devices. *Digital Investigation*, *26*, 55–62. https://doi.org/10.1016/j.diin.2018.05.003

Fleischmann, S. (2021). *X-Ways Forensics & WinHex Manual*.

Garfinkel, S. L. (2007). *Anti-forensics: Techniques, detection and countermeasures*. https://simson.net/cv/

Gargean, B. L. (2019). *How Many Times Must You Overwrite a Hard Disk? - Blancco*. https://www.blancco.com/resources/blog-many-overwriting-rounds-required-erase-hard-disk/

Grenier, C. (2019, July 23). *PhotoRec - CGSecurity*. https://www.cgsecurity.org/wiki/PhotoRec

Gutmann, P. (1996, July 25). *Secure Deletion of Data from Magnetic and Solid-State Memory*. Sixth USENIX Security Symposium Proceedings. https://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html

Hassan, N. A. (2019). Digital Forensics Basics - A Practical Guide Using Windows OS. In *Digital Forensics Basics*. Apress. https://doi.org/10.1007/978-1-4842-3838-7

KL, A. (2022). *Explore The Android File System Hierarchy In-Depth: - The Sec Master*. https://thesecmaster.com/explore-the-android-file-system-hierarchy-in-depth/

Mahalik, H., Bommisetty, S., & Tamma, R. (2016). *Practical mobile forensics : a hands-on guide to mastering mobile forensics for the iOS, Android, and Windows Phone platforms* (Second). Packt Publishing Ltd.

Piriform. (2021). *Change CCleaner for Windows settings – Piriform Support*. https://support.piriform.com/hc/en-us/articles/360048321751-Change-CCleaner-for-Windows-settings#h_01ET84PM83HFWPZ992MEZ8KGAC

Protectstar. (2022). *Securely Erase Data on Android - iShredder Android 6*. https://www.protectstar.com/en/products/ishredder-android

Samsung. (2022). *Spesifikasi Samsung Galaxy S21 Ultra 5G Terbaru | Samsung ID*. https://www.samsung.com/id/smartphones/galaxy-s21-ultra-5g/

Schwamm, R., & Rowe, N. (2014). Effects of the Factory Reset on Mobile Devices. *Journal of Digital Forensics, Security and Law*, *9*(2), 205–220. https://doi.org/10.15394/jdfsl.2014.1182

Simon, L., & Anderson, R. (2015). *Security Analysis of Android Factory Resets*. www.forbes.com/sites/connieguglielmo/2013/08/07/used-smartphone-

Sipicorp. (2022). *Secure data wiping and sanitization - Sipi*. https://www.sipicorp.com/secure-data-destruction/secure-data-wiping/

Snyder, J. (2021). *What are the security risks of rooting your smartphone?* https://insights.samsung.com/2021/10/29/what-are-the-security-risks-of-rooting-your-smartphone-3/

Statcounter. (2021). *Mobile Operating System Market Share Worldwide | Statcounter Global Stats*. https://gs.statcounter.com/os-market-share/mobile/worldwide

Tamma, R., & Tindall, D. (2015). *Learning Android Forensics* (First). Packt Publishing. www.PacktPub.com

Trend Micro. (2022). *Hash values - Definition*. https://www.trendmicro.com/vinfo/us/security/definition/hash-values

Wani, M. A., AlZahrani, A., & Bhat, W. A. (2020). File system anti-forensics – types, techniques and tools. *Computer Fraud and Security*, *2020*(3), 14–19. https://doi.org/10.1016/S1361-3723(20)30030-0