ABSTRAK

Dalam era digitalisasi saat ini, kebutuhan akan metode forensik yang efektif untuk akuisisi data remote dari perangkat Android menjadi sangat penting, terutama di tengah kondisi pandemi global. Penelitian ini mengembangkan 'Remote Acquisition for Android Smartphone' (RAAS), sebuah sistem berbasis Django Web Framework yang berfungsi sebagai alat akuisisi data remote untuk perangkat Android. RAAS dirancang untuk mengoptimalkan proses investigasi forensik digital dengan memudahkan pengumpulan bukti digital dari perangkat Android secara remote. Tujuan utama adalah menciptakan aplikasi RAAS yang dapat diakses lintas platform, memastikan proses akuisisi data yang aman dan efisien.

Berdasarkan pengujian dan evaluasi yang dilakukan, sistem RAAS terbukti dapat menjalankan proses forensik digital yang mencakup empat kaidah utama: Koleksi, Identifikasi, Akuisisi, dan Presentasi. Penelitian ini meliputi pengembangan metode akuisisi menggunakan ADB dan Netcat, pengujian aplikasi yang komprehensif, serta desain UI/UX yang intuitif, dengan penekanan khusus pada aspek keamanan dan keterjangkauan aplikasi web.

Kata Kunci: forensik digital, akuisisi data remote, android, django web framework

1. PENDAHULUAN

1.1. Latar Belakang

Dalam konteks era digitalisasi yang terus berkembang, peran smartphone sebagai alat komunikasi dan penyimpanan data pribadi menjadikannya komponen penting dalam investigasi kriminal. Studi terkini menunjukkan bahwa ketergantungan masyarakat terhadap smartphone tidak hanya meningkatkan kemudahan dalam komunikasi tetapi juga meningkatkan risiko kejahatan siber [1]. Sebuah penelitian mendapati bahwa kecanduan smartphone berkorelasi dengan peningkatan risiko menjadi korban kejahatan siber, menegaskan pentingnya memperhatikan perangkat ini dalam konteks forensik digital. Hal ini menggarisbawahi kebutuhan akan sistem yang dapat mengakuisisi data dari smartphone dengan cara yang aman dan efektif, sejalan dengan peningkatan kasus kejahatan yang melibatkan perangkat ini.Salah satu tantangan global yang dihadapi oleh dunia saat ini adalah pandemi Covid-19 yang telah mempengaruhi hampir semua aspek kehidupan. Pandemi ini juga berdampak pada bidang forensik digital, yang membutuhkan akses fisik ke perangkat yang menjadi sumber bukti digital. Namun, dalam situasi yang membatasi kontak fisik, akses fisik tersebut menjadi sulit atau bahkan tidak mungkin dilakukan. Oleh karena itu, muncul kebutuhan mendesak untuk metode akuisisi data remote yang efektif dari perangkat Android, yang merupakan salah satu platform perangkat seluler yang paling banyak digunakan di dunia. Metode ini dapat memberikan solusi untuk mengatasi kendala geografis, waktu, dan biaya yang sering dihadapi oleh penyidik forensik digital, terutama di daerah yang sulit mendapatkan akses teknologi forensik.

Digital Forensic Indonesia [2], sebuah perusahaan yang menyediakan layanan forensik digital, termasuk akuisisi data remote, di Indonesia, hanya memiliki laboratorium forensik di kota Jakarta. Hal ini menimbulkan kesenjangan antara permintaan dan ketersediaan layanan forensik digital di daerah-daerah lain. Dengan menggunakan metode akuisisi data remote, penyidik forensik digital dapat mengumpulkan bukti digital dari perangkat Android tanpa harus berada di lokasi yang sama dengan perangkat tersebut..

Sementara praktek saat ini menunjukkan adanya kekurangan aplikasi berbasis web untuk akuisisi data remote, terdapat peluang signifikan untuk

mengembangkan solusi yang efisien dan mudah diakses. RAAS, yang berfungsi sebagai sistem akuisisi data remote berbasis Django Web Framework, dirancang untuk mengatasi kebutuhan ini dengan menyediakan platform yang dapat diakses melalui browser di berbagai perangkat. Ini memungkinkan pengumpulan bukti digital dari perangkat Android secara lebih cepat, aman, dan lebih terjangkau, dengan fokus pada keamanan dan keterjangkauan yang merupakan aspek penting dalam forensik digital.

Oleh karena itu, penelitian ini akan dilakukan untuk mengembangkan RAAS, sebuah sistem akuisisi data forensik digital berbasis web yang inovatif. RAAS dirancang untuk memenuhi standar forensik digital yang ketat, sehingga memungkinkan pengumpulan bukti digital dari perangkat Android secara lebih cepat, aman, dan lebih terjangkau. Performansi sistem RAAS ini akan diuji berdasarkan efisiensi proses akuisisi, akurasi dan integritas data yang dikumpulkan. Evaluasi ini akan melibatkan serangkaian pengujian skenario yang realistis untuk memastikan bahwa sistem dapat diandalkan dalam kondisi operasional yang beragam.

1.2. Perumusan Masalah

Bagaimana mengembangkan sistem akuisisi data forensik digital berbasis web yang mampu memenuhi kebutuhan forensik digital yang beragam, dengan fokus pada keamanan dan keterjangkauan, serta apakah belum ada aplikasi serupa versi web untuk memenuhi kaidah forensik?

1.3. Tujuan

Berikut adalah tujuan dari penelitian ini:

1. Mengembangkan sebuah sistem akuisisi jarak jauh dalam bentuk aplikasi berbasis web untuk perangkat Android
2. Menganalisis performansi sistem dalam melakukan akuisisi yang sesuai dengan kaidah forensik yaitu dalam hal menjaga integritas data hasil akuisisi dan dapat menjaga keamanan akses dari sitem akuisisi jarak jauh ini.

1.4. Rencana Kegiatan

Berikut adalah rencana kegiatan yang akan dilakukan pada penelitian ini:

1. Optimalisasi Penggunaan ADB:
    a. Melakukan penelitian terperinci tentang perintah-perintah ADB yang paling efektif untuk akuisisi data pada berbagai versi Android.
    b. Membangun skrip ADB yang otomatis mengenali versi Android dan menerapkan set perintah yang sesuai untuk akuisisi data.

2. Integrasi Netcat untuk Transfer Data:
    a. Mengembangkan protokol atau prosedur untuk memulai sesi netcat yang aman dan stabil untuk transfer data akuisisi.
    b. Menyusun mekanisme untuk memastikan transfer data yang terenkripsi dan terproteksi selama proses akuisisi jarak jauh.

3. Pengujian dan Validasi:

    Melakukan pengujian intensif dari skenario akuisisi data untuk memverifikasi bahwa akuisisi data berfungsi secara konsisten di berbagai perangkat dan versi sistem operasi.

4. Pengembangan UI/UX Aplikasi RAAS:
    a. Merancang antarmuka pengguna yang intuitif yang memungkinkan investigator untuk dengan mudah memulai dan memonitor proses akuisisi data remote.
    b. Memastikan bahwa aplikasi memberikan umpan balik yang jelas dan akurat mengenai status akuisisi dan proses transfer data.

5. Keamanan Aplikasi Web:

    Mengimplementasikan fitur keamanan Django, seperti middleware keamanan dan sistem autentikasi, untuk melindungi aplikasi web.

1.5. Jadwal Kegiatan

Tabel 1 - Jadwal kegiatan

Kegiatan	Bulan					
	1	2	3	4	5	6
Studi literatur						
Pengumpulan data						
Perancangan sistem						
Implementasi sistem						
Menulis laporan Tugas Akhir						

2. KAJIAN PUSTAKA

2.1. Studi Terkait

Remote akuisisi adalah proses pengumpulan dan analisis bukti digital dari perangkat atau jaringan yang berlokasi jauh dari penyidik, tanpa memerlukan akses fisik [10]. Proses ini dapat membantu mengatasi tantangan logistik, biaya, dan waktu yang terkait dengan penyelidikan forensik digital konvensional. Namun, *remote* akuisi juga menimbulkan beberapa masalah, seperti integritas, keamanan, serta kompatibilitas dari aplikasi yang digunakan.

2.2. Penelitian Terkait

Berikut adalah penelitian terkait *remote acquisition* dan metode forensik digital yang dipublikasikan sejak tahun 2018 sampai dengan sekarang.

Tabel 2 - Penelitian terkait

Penulis	Masalah	Metode	Solusi
Al-Dhaqm A et al. [8]	Penulis mengidentifikasi masalah bahwa proses investigasi forensik digital pada perangkat mobile masih menghadapi berbagai tantangan, seperti keragaman infrastruktur, format, dan artefak perangkat mobile, kurangnya model investigasi yang baku dan terstandar, serta keterbatasan alat dan teknik forensik yang tersedia.	Penulis mengusulkan sebuah model proses investigasi forensik mobile yang harmonis dan komprehensif, yang disebut HMFIPM. Model ini dibangun dengan menggunakan metode Design Science Research (DSR) dan mengadaptasi 24 model proses investigasi forensik mobile yang ada di literatur. Model ini terdiri dari tujuh proses investigasi, yaitu persiapan, akuisisi data, preservasi, pemeriksaan, analisis, pelaporan, dan presentasi.	Penulis menjelaskan karakteristik solusi mereka, seperti tujuan, kelebihan, keterbatasan, dan validasi dari model HMFIPM. Tujuan dari model ini adalah untuk menyediakan sebuah kerangka kerja yang dapat digunakan oleh praktisi forensik untuk melakukan investigasi forensik mobile secara efektif dan efisien. Kelebihan dari model ini adalah bahwa model ini dapat menangani berbagai jenis perangkat mobile dengan infrastruktur yang

			berbeda-beda, serta dapat mengintegrasikan pengetahuan dan aktivitas dari setiap proses investigasi. Keterbatasan dari model ini adalah bahwa model ini masih membutuhkan pengujian lebih lanjut dan verifikasi teknologi untuk membuktikan keandalan dan keteraplikasiannya.
Chawla S [4]	[-] Kebocoran Data: Lamplighter Co. mengalami kebocoran informasi dan system intelektual yang diduga dilakukan oleh seorang karyawan. [-] Pembatasan Pandemi: Pandemi mengakibatkan pembatasan perjalanan, membuat investigasi fisik langsung menjadi tidak mungkin. [-] Keterbatasan Sumber Daya: Lamplighter Co. tidak ingin menghabiskan terlalu banyak waktu atau uang untuk investigasi	[-] Investigasi Jarak Jauh: Menggunakan TeamViewer dan teknologi lain untuk melakukan investigasi secara remote. [-] Analisis Log Akses Fisik: Mengecualikan kemungkinan akses fisik ke server melalui log akses fisik dan rekaman video keamanan. [-] Pemeriksaan Konfigurasi Sistem: Meninjau konfigurasi system dan log TeamViewer untuk mengidentifikasi aktivitas mencurigakan.	*Keunggulan:* [-] Efisiensi: Metode ini memungkinkan investigasi cepat dan hemat biaya, yang penting bagi system yang tidak ingin mengalokasikan sumber daya yang signifikan untuk kasus ini. [-] Fleksibilitas: Investigasi jarak jauh mengatasi tantangan yang ditimbulkan oleh pembatasan pandemi. [-] Penentuan Akar Penyebab: Meskipun tidak ada system hukum yang diambil, investigasi berhasil

	atau system hukum berkelanjutan.		mengidentifikasi sumber kebocoran data dan akar penyebabnya. *Kekurangan:* [-] Kurangnya Bukti Hukum: Metode investigasi tidak mengumpulkan bukti yang cukup untuk proses hukum, sesuai dengan keinginan Lamplighter Co. [-] Ketergantungan pada Teknologi: Keberhasilan metode ini sangat bergantung pada teknologi dan akses ke log dan data system.
Maheswari K et al. [5]	Dokumen tersebut membahas tantangan dan kebutuhan dalam bidang investigasi forensik digital jarak jauh. Khususnya, ia menyoroti isu melakukan evaluasi pada perangkat yang berpotensi kompromi tanpa akses fisik, menekankan pentingnya	menganalisis berbagai metodologi yang saat ini digunakan untuk investigasi forensik jarak jauh. Ia membandingkan berbagai perangkat lunak dan perangkat keras terkini, serta teknik untuk melaksanakan berbagai tahapan	output kualitatif yang diamati dari memori, timeline, dan pencitraan forensik langsung. Perbandingan ini bertujuan untuk menyederhanakan proses pemilihan teknik yang paling tepat di bawah berbagai kondisi untuk investigasi jarak jauh yang efektif.

	investigasi internal yang diskrit, pengurangan biaya, dan efisiensi waktu dalam menyampaikan kemampuan forensik digital secara jarak jauh	investigasi forensik	**[-] Investigasi Diskrit**: Teknik forensik jarak jauh memungkinkan investigasi yang diskrit tanpa memberitahu pemilik sistem. **[-] Aplikasi Global**: Mereka menghilangkan kebutuhan akan kehadiran fisik, memungkinkan jangkauan global dan respons yang cepat. **Kekurangan**: Rincian tentang kekurangan spesifik dari alat dan teknik yang ditinjau tidak secara langsung disebutkan dalam teks yang diekstrak. Namun, tantangan umum dalam forensik digital jarak jauh mungkin termasuk masalah terkait keamanan data, keandalan koneksi jarak jauh, dan kompleksitas dalam menangani data yang beragam dan mungkin terenkripsi atau disamarkan.
Ibrahim S et al. [7]	Penulis mengidentifikasi	Penulis mengusulkan	Penulis menjelaskan

	masalah bahwa alat forensik digital yang ada saat ini memiliki beberapa kekurangan, seperti biaya yang tinggi, kompleksitas penggunaan, dan keterbatasan dalam mengakuisisi data dari media penyimpanan yang berbeda atau melalui jaringan.	sebuah solusi alternatif yang menggunakan Raspberry Pi, sebuah komputer mini berukuran saku, yang menjalankan sistem operasi Kali Linux atau Caine. Solusi ini dapat melakukan akuisisi data dari berbagai media penyimpanan secara lokal atau jarak jauh dengan biaya yang rendah dan kemudahan penggunaan.	karakteristik solusi mereka, seperti spesifikasi perangkat keras Raspberry Pi, langkah-langkah instalasi dan konfigurasi sistem operasi, perintah-perintah untuk melakukan akuisisi data, dan hasil eksperimen yang menunjukkan kinerja solusi. Penulis juga membahas keunggulan dan kekurangan solusi mereka, seperti fleksibilitas, portabilitas, keamanan, dan keterbatasan dalam hal sumber daya dan stabilitas
Casino F et al. [6]	Artikel ini membahas tantangan dan topik baru dalam bidang forensik digital, yaitu ilmu yang berkaitan dengan pengumpulan, analisis, dan penyajian bukti digital dalam konteks kejahatan siber. Penulis menunjukkan bahwa forensik digital menghadapi berbagai masalah, seperti keragaman sumber bukti, volume data yang besar, enkripsi dan teknik anti-	Penulis melakukan tinjauan literatur dari 109 artikel ilmiah dan 51 laporan yang terkait dengan forensik digital, dengan menggunakan kriteria inklusi dan eksklusi yang ketat. Penulis juga melakukan analisis kualitatif dengan menggunakan perangkat lunak MAXQDA11 untuk mengklasifikasikan dan mensintesis data yang diekstrak. Penulis menggunakan pendekatan sintesis	Keunggulan dari solusi yang diusulkan oleh penulis adalah memberikan gambaran menyeluruh dan komprehensif tentang keadaan seni dan praktik dalam forensik digital, serta menyoroti isu-isu penting, tren, dan arah penelitian masa depan dalam bidang ini. Solusi ini juga memungkinkan kolaborasi yang lebih erat antara peneliti dan praktisi

	forensik, standar dan etika yang belum matang, dan kerjasama lintas batas yang sulit.	naratif dan analisis tematik untuk menjawab serangkaian pertanyaan penelitian yang telah ditentukan sebelumnya.	dari berbagai topik forensik digital. Kekurangan dari solusi ini adalah tidak adanya pedoman pelaporan yang standar untuk tinjauan literatur jenis ini, serta keterbatasan dalam mencakup semua aspek dan sub-bidang forensik digital yang mungkin ada.

Pada tabel 2 ini adalah hasil rangkuman terkait beberapa paper tentang *remote acquisition* yang mereka gunakan dan dipublikasikan sejak tahun 2018 sampai dengan sekarang.

Tabel 3 - Ringkasan aplikasi yang digunakan pada paper

Penulis	**Aplikasi**	**Platform**
Hitesh Sachdev et al. [3]	Paraben E3: Universal	Windows (Desktop)
Irvin Homem et al. [16]	P2P-da & LEIA	Android (Mobile)
Thankaraja Raja Sree et al. [17]	FROST & F-response	IaaS, IP Protocol
Saeed Ibrahim [7]	Raspberry Pi 3	Hardware
Shweta A. Chawla 4]	AnyDesk, FTK Imager	Windows (Desktop)

Pada peninjauan terhadap aplikasi yang diimplementasikan dalam karya ilmiah sebelumnya, terdapat keberagaman dalam penggunaan platform serta kebutuhan akan sistem operasi dan protokol tertentu yang spesifik. Berbeda dengan hal tersebut, penelitian ini mengembangkan aplikasi *Remote Acquistion for Android Smartphone* (RAAS) berbasis platform web. Aplikasi berbentuk web ini memiliki keunggulan kompatibilitas lintas platform, mengingat kemampuan hampir semua sistem operasi bisa untuk menjalankan protokol web.

Dengan demikian, aplikasi ini dapat diakses melalui berbagai platform yang menyediakan dukungan terhadap peramban web. Keunikan ini menandai perbedaan signifikan antara aplikasi RAAS yang dijelaskan dalam penelitian ini dengan aplikasi akuisisi data jarak jauh yang telah dibahas dalam literatur sebelumnya.

Kemudian, ada beberapa aplikasi yang serupa namun juga sudah diproduksi secara komersial, seperti:

2.2.1. MOBILedit

MOBILedit adalah solusi all-in-one untuk ekstraksi data dari telepon, smartwatch, dan cloud. Aplikasi ini menggunakan dua metode akuisisi data, fisik dan logis, memiliki analisis aplikasi yang sangat baik, *recovery* data yang dihapus, berbagai perangkat yang didukung, laporan yang disesuaikan dengan baik, dan jgua memiliki *interface* yang mudah digunakan [15]. MOBILedit sendiri menggunakan ADB sebagai kunci atau core utama dalam aplikasi ini, maka dari itu ketika MOBILedit juga bisa melakukan akuisi secara network dengan mengirim data lewat *network* atau jaringan *wireless.*

2.3. Akuisisi

Akuisisi data dalam konteks forensik digital merujuk pada proses pengumpulan data dari perangkat penyimpanan dengan cara yang memastikan bahwa data tersebut tetap dalam keadaan aslinya dan tidak terkontaminasi atau berubah selama proses investigasi. Hal ini penting untuk menjaga integritas bukti digital. Metode yang digunakan harus memastikan bahwa data yang diakses tidak mengalami penulisan ulang atau modifikasi [9]. Untuk mencapai hal ini, alat-alat khusus seperti FTK Imager, EnCase, dan ProDiscover digunakan untuk membuat gambar bit demi bit dari media penyimpanan, yang menghasilkan salinan eksakta tanpa mengubah data asli. Gambar atau salinan ini kemudian digunakan sebagai dasar untuk pemeriksaan dan analisis lebih lanjut oleh investigator forensik digital. Proses ini memungkinkan pengumpulan bukti yang dapat dipercaya dan dapat dipertanggungjawabkan di pengadilan atau dalam konteks hukum lainnya.

Kemudian jenis-jenis akuisisi dibedakan dalam beberapa jenis, diantaranya:

2.3.1. Akuisisi *Static*

Akuisisi static metode di mana data disalin dari perangkat penyimpanan tersangka pada waktu tertentu. Ini biasanya dilakukan ketika tidak ada risiko kehilangan aksesibilitas data, seperti ketika perangkat tidak terenkripsi atau ketika kunci enkripsi tersedia [9]. Tujuannya adalah untuk memastikan bahwa data dapat diakses dan dianalisis nanti tanpa perubahan.

2.3.2. Akuisisi *Live*

Akusisi live dilakukan secara real-time dan sering kali diperlukan ketika ada risiko kehilangan data karena enkripsi atau karena komputer perlu dimatikan, yang mungkin mengakibatkan kehilangan data yang volatile [9]. Metode ini penting di tempat kejadian perkara, terutama jika diperlukan untuk mengamankan data yang bisa hilang seketika atau ketika perangkat penyimpanan telah dienkripsi dan kunci dekripsinya tidak diketahui. Akuisisi langsung memastikan bahwa data penting dapat diselamatkan sebelum komputer dimatikan atau sebelum data terenkripsi menjadi tidak dapat diakses.

2.3.3. Akuisisi *Dead*

Akuisi dead juga dikenal sebagai akuisisi statis, adalah teknik akuisisi data forensik di mana proses pembuatan salinan digital (biasanya disebut sebagai "image") dari media penyimpanan dilakukan ketika perangkat tersebut tidak aktif atau 'mati'. Ini sering melibatkan fisik hard drive yang telah dihapus dari sistem komputer aslinya untuk mencegah perubahan lebih lanjut pada data saat proses ini [10]. Proses ini dianggap paling komprehensif karena menyediakan salinan lengkap dari semua data yang tersimpan pada media pada titik waktu tertentu. Ini termasuk file yang ada, file yang dihapus yang masih dapat dipulihkan, serta data yang tersembunyi atau tidak langsung terlihat, seperti slack space dan unallocated space. Karena perangkat tidak sedang beroperasi, risiko menulis data baru ke perangkat dan berpotensi merusak bukti diminimalkan.

Pendekatan ini memerlukan alat-alat khusus dan biasanya melibatkan penggunaan perangkat keras atau perangkat lunak yang dirancang untuk membuat salinan bit demi bit dari media penyimpanan. Alat-alat ini, seperti write blockers, digunakan untuk memastikan bahwa tidak ada data yang ditulis kembali ke media

selama proses imaging. Setelah gambar disk dibuat, semua analisis dilakukan pada gambar tersebut, bukan pada media asli, yang memastikan bahwa data asli tetap tidak terganggu.

2.4. Django

Django adalah sebuah framework pemrograman yang telah dibentuk oleh keputusan desain dan alat yang sudah ditentukan oleh pengembangnya. Framework ini dirancang untuk memudahkan para programmer dalam membangun aplikasi web dengan menyediakan struktur dasar yang mengeliminasi kebutuhan pengkodean berulang dan memastikan konsistensi. Penggunaan Django memungkinkan pengembang untuk fokus pada implementasi fitur unik aplikasi mereka, sambil mengadopsi praktik terbaik yang telah diuji oleh komunitas Django [12].

Filosofi di balik Django mengakui pentingnya proses kognitif dalam pengembangan perangkat lunak, menekankan bahwa kode yang baik bukan hanya tentang menyelesaikan tugas, melainkan juga tentang merefleksikan keputusan dan strategi yang mengarah pada penciptaan solusi teknis yang efektif. Ini menggiring pengembang untuk mengambil pendekatan yang lebih matang terhadap pengkodean, yang tidak hanya berorientasi pada hasil, tetapi juga pada pemahaman yang lebih dalam tentang bagaimana dan mengapa elemen tertentu dari aplikasi web dirancang.

Dengan demikian, Django tidak hanya menjadi alat bagi pengembang untuk mencapai tujuan teknis mereka, tetapi juga sarana pendidikan yang mengajar tentang prinsip-prinsip desain perangkat lunak yang solid. Membaca dan memahami kerangka kerja ini dianggap krusial, terutama bagi mereka yang berkeinginan untuk berkomunikasi efektif dengan anggota komunitas Django dan untuk berkontribusi pada pembuatan aplikasi web yang berkelanjutan dan berkualitas.

2.4.1. Kenapa Harus Django?

Django adalah sebuah framework web Python yang populer dan telah digunakan oleh banyak perusahaan besar, seperti Instagram, Spotify, dan

National Geographic. Framework ini memiliki banyak keunggulan, di antaranya:

- **Kemudahan penggunaan:** Django dirancang untuk memudahkan para programmer dalam membangun aplikasi web dengan menyediakan struktur dasar yang mengeliminasi kebutuhan pengkodean berulang dan memastikan konsistensi. [18]
- **Keamanan:** Django memiliki fitur keamanan bawaan yang kuat, seperti perlindungan terhadap serangan cross-site request forgery (CSRF) dan SQL injection. [18]
- **Scalability:** Django dapat digunakan untuk membangun aplikasi web dari berbagai skala, mulai dari skala kecil hingga skala besar. [18]
- **Komunitas yang aktif:** Django memiliki komunitas pengguna dan pendukung yang besar yang menyediakan dukungan dan sumber daya yang luas. [18]

2.4.2. Perbandingan Django dengan Framework Web Python Lainnya

Selain Django, ada dua framework web Python populer lainnya, yaitu Flask dan Pyramid. Berikut adalah perbandingan singkat antara Django dengan kedua framework tersebut:

Tabel 4 - Django vs Flask vs Pyramid

Karakteristik	Django	Flask	Pyramid
Kemudahan penggunaan	Mudah digunakan untuk pemula	Mudah digunakan dan dipelajari	Lebih kompleks daripada Django dan Flask
Keamanan	Memiliki fitur keamanan bawaan yang kuat	Tidak memiliki fitur keamanan bawaan, tetapi dapat ditambahkan dengan ekstensi	Memiliki fitur keamanan bawaan yang kuat
Scalability	Dapat digunakan untuk membangun	Dapat digunakan untuk membangun	Dapat digunakan untuk membangun

	aplikasi web dari berbagai skala	aplikasi web dari berbagai skala	aplikasi web dari berbagai skala
Komunitas	Memiliki komunitas pengguna dan pendukung yang besar	Memiliki komunitas pengguna dan pendukung yang aktif	Memiliki komunitas pengguna dan pendukung yang aktif
Jenis aplikasi	Cocok untuk berbagai jenis aplikasi web	Cocok untuk aplikasi web sederhana dan menengah	Cocok untuk aplikasi web kompleks
Dokumentasi	Dokumentasi yang lengkap dan mudah dipahami	Dokumentasi yang lengkap dan mudah dipahami	Dokumentasi yang lengkap dan mudah dipahami

Dengan kata lain, Django cocok untuk berbagai jenis aplikasi web, mulai dari situs web statis hingga aplikasi web kompleks. Jika membutuhkan framework web yang mudah digunakan dan memiliki fitur keamanan bawaan yang kuat, Django adalah pilihan yang tepat.

2.5. Android

Android adalah sistem operasi berbasis Linux yang dikembangkan untuk perangkat bergerak dan merupakan bagian dari proyek open-source yang lebih besar, Android Open Source Project (AOSP). Dirancang untuk fleksibilitas dan adaptasi, Android mendukung berbagai arsitektur hardware dan menyediakan lingkungan pengembangan yang kuat dengan toolchain dan pustaka yang terus diperbarui. Awalnya, Android dibangun untuk perangkat dengan keterbatasan memori dan penyimpanan, mengutamakan efisiensi sumber daya dan kehematan data. Namun, sejak itu, Android telah berevolusi untuk mendukung perangkat yang lebih canggih dengan berbagai fitur yang kompleks, menjadikannya pilihan populer bagi produsen perangkat mobile [11].

Android menawarkan kerangka kerja yang memudahkan penerapan aplikasi Java yang dioptimalkan untuk lingkungan mobile, sambil juga memungkinkan perangkat untuk terintegrasi dengan internet dan teknologi smart lainnya.

Sementara Google mengendalikan Android melalui layanan dan aplikasi proprietari seperti Google Play Services, produsen perangkat bebas untuk menggunakan dan memodifikasi AOSP untuk kebutuhan mereka sendiri. Kendati banyak perangkat Android yang menyertakan layanan Google, AOSP juga digunakan sebagai dasar untuk sistem operasi yang dimodifikasi oleh produsen lain atau untuk proyek-proyek yang tidak melibatkan Google sama sekali, menunjukkan fleksibilitas dan jangkauan sistem operasi ini.

2.5.1. Struktur File Sistem Android

Android menggunakan sistem file berbasis Linux yang terdiri dari beberapa partisi dengan fungsi dan format yang berbeda.

Partisi utama adalah /system, /data, /cache, /boot, /recovery, dan /misc. Partisi /system berisi file-file sistem operasi dan aplikasi bawaan, partisi /data berisi file-file data pengguna dan aplikasi yang diinstal, partisi /cache berisi file-file sementara yang digunakan oleh sistem dan aplikasi, partisi /boot berisi kernel dan ramdisk yang digunakan saat booting, partisi /recovery berisi program yang digunakan untuk pemulihan sistem, dan partisi /misc berisi informasi konfigurasi yang digunakan oleh bootloader [13].

Selain partisi-partisi tersebut, Android juga menggunakan penyimpanan eksternal yang bisa berupa kartu SD yang dapat dilepas atau bagian dari memori internal yang dialokasikan sebagai penyimpanan eksternal. Penyimpanan eksternal biasanya menggunakan sistem file FAT32 atau exFAT dan berisi file-file media, data aplikasi, dan lain-lain.

Android juga mendukung penyimpanan internal yang bersifat privat dan tidak dapat diakses oleh aplikasi lain. Penyimpanan internal ini biasanya berada di bawah direktori /data/data dan menggunakan sistem file ext4 atau yaffs2. Penyimpanan internal ini berisi file-file preferensi, database, dan lain-lain yang berkaitan dengan aplikasi tertentu.

2.5.2. Lokasi Data Penting

Data penting yang berkaitan dengan aktivitas pengguna dan aplikasi dapat ditemukan di berbagai lokasi dalam sistem file Android. Beberapa contoh lokasi data penting adalah sebagai berikut:

- /data/data: Berisi data aplikasi yang bersifat privat, seperti file preferensi, database, cache, dan lain-lain. Data di lokasi ini hanya dapat diakses oleh aplikasi yang bersangkutan atau oleh pengguna root [13].
- /data/media: Berisi data yang disimpan di penyimpanan internal yang dianggap sebagai penyimpanan eksternal oleh sistem. Data di lokasi ini dapat diakses oleh aplikasi lain dan oleh pengguna melalui MTP atau PTP [13].
- /sdcard: Berisi data yang disimpan di kartu SD yang dapat dilepas. Data di lokasi ini juga dapat diakses oleh aplikasi lain dan oleh pengguna melalui USB mass storage atau MTP [13].
- /data/system: Berisi data yang berkaitan dengan sistem, seperti file konfigurasi, keamanan, akun, dan lain-lain. Data di lokasi ini hanya dapat diakses oleh pengguna root [13].
- /data/misc: Berisi data yang berkaitan dengan berbagai layanan dan fungsi sistem, seperti Wi-Fi, Bluetooth, VPN, dan lain-lain. Data di lokasi ini juga hanya dapat diakses oleh pengguna root [13].

2.5.3. Relevansi dengan Akuisi Data

Android, sebagai sistem operasi berbasis Linux yang dikembangkan untuk perangkat bergerak, memiliki relevansi yang signifikan dalam akuisisi data. Dalam konteks forensik digital, akuisisi data merujuk pada proses pengumpulan data dari perangkat untuk analisis lebih lanjut. Ada berbagai metode akuisisi data yang dapat digunakan, tergantung pada jenis dan model perangkat, tingkat akses yang dimiliki, dan tujuan investigasi [14].

Metode akuisisi data meliputi ekstraksi manual, logis, fisik, JTAG, dan chip-off. Ekstraksi manual adalah teknik yang paling sederhana dan tidak memerlukan alat khusus. Teknik ini hanya mengambil data yang terlihat oleh pengguna, seperti

kontak, SMS, dan foto. Sementara itu, ekstraksi logis menggunakan perintah ADB atau alat forensik komersial untuk mengambil data dari sistem file perangkat, termasuk file aplikasi dan database SQLite.

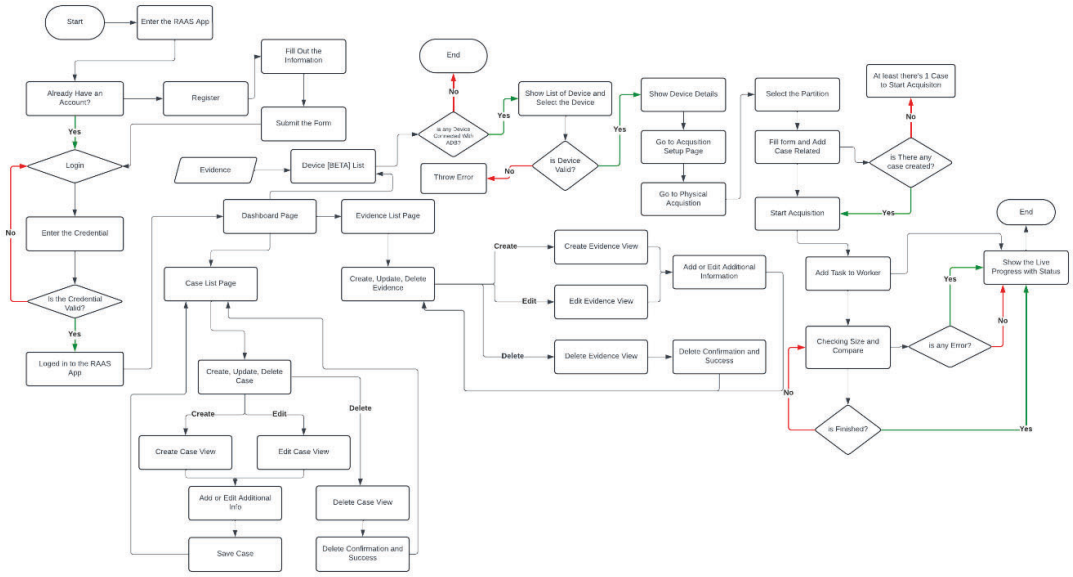
Ekstraksi fisik menciptakan citra bit demi bit dari memori perangkat, memungkinkan pemulihan data yang dihapus atau tersembunyi. Teknik ini memerlukan akses root dan bootloader yang tidak terkunci. JTAG adalah teknik yang menggunakan antarmuka perangkat keras untuk mengakses memori perangkat secara langsung, tanpa melalui sistem operasi. Teknik ini memerlukan pengetahuan dan keterampilan teknis yang tinggi, serta peralatan khusus [14].

Chip-off adalah teknik yang melibatkan pengangkatan dan pembacaan chip memori perangkat dengan alat forensik khusus. Teknik ini adalah teknik terakhir yang digunakan jika teknik lain gagal atau tidak memungkinkan, karena dapat merusak perangkat secara permanen. Dengan demikian, Android, dengan struktur file sistemnya dan berbagai lokasi data penting, memainkan peran penting dalam akuisisi data dalam konteks forensik digital.

Beberapa alat dan teknik yang digunakan dalam akuisisi data, serta bagaimana mereka dapat digunakan dalam konteks forensik perangkat bergerak. Alat-alat ini dapat membantu dalam mengidentifikasi, mengumpulkan, dan menganalisis data yang relevan untuk penyelidikan. Teknik-teknik ini dapat membantu dalam memastikan bahwa data yang dikumpulkan adalah akurat, lengkap, dan dapat dipertahankan dalam pengadilan.

3. PERANCANGAN SISTEM

3.1. Deskripsi Sistem



Gambar 1 - Alur kerja aplikasi RAAS

Sistem RAAS dikembangkan menggunakan Django sebagai framework backend dan Netcat untuk streaming data. Implementasi teknis dapat dilihat pada repository kode sumber berikut: [21]. Alur kerja dari system ini dapat dilihat pada Gambar 1. Alur kerja ini menggambarkan proses akuisisi data dari tahap awal hingga akhir. Setiap langkah dalam flowchart ini dijelaskan dengan rinci, mulai dari inisiasi sistem hingga akhir proses akuisisi, memastikan bahwa pembaca dapat mengikuti proses secara logis.

Berikut adalah penjelasan dari setiap komponen dalam flowchart tersebut:

1. **Enter the RAAS App**: Pengguna memulai dengan membuka aplikasi RAAS. Ini adalah langkah awal sebelum pengguna melakukan proses apapun di dalam sistem.

2. **Login**: Jika pengguna sudah memiliki akun, mereka akan masuk ke sistem dengan menggunakan kredensial mereka. Jika tidak, pengguna harus mendaftar terlebih dahulu.

3. **Submit the Form**: Pengguna yang sudah terdaftar harus melengkapi form informasi sebelum melanjutkan.
4. **Device [BETA] List**: Setelah pengguna berhasil login dan masuk ke halaman Dashboard, mereka dapat melihat daftar perangkat yang terhubung melalui ADB di Device List. Ini adalah langkah awal sebelum memulai proses akuisisi data dari perangkat.
5. **Is any Device Connected with ADB?**: Sistem akan memeriksa apakah ada perangkat yang terhubung melalui ADB. Jika tidak ada perangkat yang terhubung, sistem akan memberikan pesan Throw Error dan proses akan dihentikan. Jika ada perangkat yang terhubung, pengguna bisa melanjutkan ke langkah berikutnya.
6. **Is Device Valid?**: Sistem akan memeriksa apakah perangkat yang terhubung valid. Jika perangkat tidak valid, sistem akan mengeluarkan pesan error dan proses akan berhenti. Jika perangkat valid, pengguna bisa melanjutkan ke langkah selanjutnya.
7. **Show Device Details**: Jika perangkat valid, pengguna bisa melihat detail dari perangkat tersebut. Ini adalah langkah di mana pengguna dapat mengakses informasi rinci dari perangkat yang akan diakuisisi.
8. **Go to Acquisition Setup Page**: Setelah detail perangkat diperiksa, pengguna diarahkan ke halaman Acquisition Setup untuk mulai mengonfigurasi proses akuisisi data dari perangkat.
9. **Select the Partition**: Pada tahap ini, pengguna harus memilih partisi dari perangkat yang ingin diakuisisi, misalnya memori internal atau partisi tertentu lainnya.
10. **Fill Form and Add Case Related**: Pengguna kemudian mengisi form terkait dengan informasi kasus yang berhubungan dengan perangkat yang akan diakuisisi. Informasi ini akan digunakan untuk mendokumentasikan proses akuisisi.

11. **Is There any Case Created?**: Sistem akan memeriksa apakah ada kasus yang telah dibuat. Jika belum ada kasus, sistem akan memberikan pesan error dan menghentikan proses. Jika ada kasus yang sudah dibuat, pengguna dapat melanjutkan ke langkah berikutnya.

12. **Start Acquisition**: Pengguna dapat memulai proses Acquisition setelah memilih partisi dan melengkapi informasi yang diperlukan. Ini adalah proses inti dari akuisisi data digital.

13. **Add Task to Worker**: Pengguna akan menambahkan tugas untuk Worker, yang bertanggung jawab atas pelaksanaan akuisisi data dari perangkat.

14. **Checking Size and Compare**: Sistem akan memeriksa ukuran data yang diambil dan membandingkannya dengan sumbernya untuk memastikan integritas data selama proses akuisisi.

15. **Is any Error?**: Sistem akan memverifikasi apakah ada kesalahan selama proses akuisisi. Jika terjadi kesalahan, sistem akan memberikan opsi bagi pengguna untuk memperbaikinya, atau proses akan dihentikan dengan status error.

16. **Show the Live Progress with Status**: sistem akan menampilkan status Live Progress, yang memungkinkan pengguna untuk memantau secara real-time proses akuisisi yang sedang berlangsung hingga selesai (Jika ada error, juga akan menampilkan asalan errornya).

17. **Is Finished?**: Setelah proses selesai, sistem akan memeriksa apakah akuisisi sudah berhasil. Jika proses akuisisi selesai dengan sukses, pengguna dapat melihat hasilnya, dan alur proses pun berakhir.

Dengan demikian, dari flowchat ini dapat dilihat langkah-langkah yang perlu diambil dalam proses pengambilan bukti digital, dari identifikasi bukti hingga akhir akuisisi.

3.2. Fungsi Utama

Fungsi utama dari proses akuisisi pada sistem RAAS melibatkan penggunaan kombinasi perintah yang memanfaatkan ADB (Android Debug Bridge) untuk mengakses partisi perangkat Android secara langsung dan mengambil data menggunakan metode dd serta netcat. Berikut adalah penjelasan rinci mengenai proses dan alur kerja dari akuisisi ini.

3.2.1. Pengecekan Kebutuhan Resume Proses

Sistem memulai proses akuisisi dengan memeriksa apakah ada byte yang sudah ditransfer sebelumnya. Hal ini dilakukan untuk menentukan apakah akuisisi harus dimulai dari awal atau dilanjutkan dari posisi terakhir yang berhasil.

```
# Check if we need to resume
seek_skip_block =
getAcquisitionObject.physical.total_transferred_bytes or 0
```

- **seek_skip_block** diinisialisasi untuk memeriksa berapa banyak byte yang telah ditransfer dari total partisi perangkat yang sedang diakuisisi. Jika proses sebelumnya terputus, akuisisi akan dilanjutkan dari byte terakhir yang berhasil diambil.
- Langkah ini memastikan proses yang efisien dan menghindari duplikasi data.

3.2.2. Menentukan Perintah dd dan nc di Perangkat

Sistem kemudian menentukan perintah yang digunakan untuk mengambil data dari perangkat. Pada perangkat Android, digunakan busybox untuk mengakses perintah dd (disk dump) dan netcat (nc) yang digunakan untuk mentransfer data dari perangkat ke server.

```
# Determine which 'dd' and 'nc' commands to use on the device
dd_command = '/data/local/busybox dd'
nc_command = '/data/local/busybox nc'
```

- **dd_command** adalah perintah utama yang digunakan untuk membaca data dari partisi perangkat.
- **nc_command** (netcat) digunakan untuk mengirimkan data yang diambil melalui jaringan ke server akuisisi.

3.2.3. Menentukan Ukuran Blok dan Posisi Mulai (Seek)

Pada bagian ini, sistem menetapkan ukuran blok untuk proses akuisisi dan menghitung dari mana harus mulai membaca data jika perlu melanjutkan dari proses yang sebelumnya terputus.

```
# Set block size
bs = 512
seek_blocks = seek_skip_block // bs
```

- **bs** (block size) ditetapkan sebesar 512 byte, yang merupakan ukuran standar untuk akuisisi disk. Ini berarti data akan dibaca dan ditransfer dalam blok-blok sebesar 512 byte.
- **seek_blocks** digunakan untuk menentukan berapa blok yang harus dilewati berdasarkan byte yang telah ditransfer, sehingga akuisisi tidak mengulangi proses dari awal.

3.2.4. Membangun Perintah Akuisisi di Perangkat Android

Selanjutnya, sistem membangun perintah yang akan dijalankan di perangkat Android melalui ADB untuk memulai proses akuisisi data.

```
android_command = f"adb -s {SERIAL_ID} shell \"su 0 -c '{dd_command}
if=/dev/block/{PARTITION} bs={bs}"
if seek_skip_block > 0:
    android_command += f" skip={seek_blocks}"
android_command += f" | {nc_command} -l -p {PORT_SERVER}'\""
```

- Perintah ini akan mengakses partisi tertentu pada perangkat Android melalui **dd** dan meneruskan data yang dibaca ke **netcat** yang berjalan pada port tertentu (ditentukan oleh **PORT_SERVER**).
- Jika proses akuisisi harus dilanjutkan dari titik tertentu, maka sistem menambahkan parameter **skip={seek_blocks}** pada perintah **dd** untuk melewati blok yang sudah ditransfer.

3.2.5. Menjalankan Perintah Android

Setelah perintah dibangun, sistem menjalankan perintah ini menggunakan `*subprocess*` untuk mengeksekusi perintah di perangkat Android melalui shell.

```
android_process = subprocess.Popen(android_command, shell=True)
time.sleep(2)
```

- Proses ini dimulai terlebih dahulu untuk memastikan perangkat sudah siap menerima perintah dan memulai transfer data.

3.2.6. Membangun dan Menjalankan Perintah di Server

Setelah proses di perangkat Android dimulai, sistem juga membangun perintah untuk server yang akan menerima data dari perangkat Android melalui netcat dan menyimpannya ke lokasi tertentu.

```
if seek_skip_block > 0:
    server_command = f"netcat {IP} {PORT_CLIENT} -q 1 | dd
of={LOCATION}/{FILE_NAME} bs={bs} seek={seek_blocks} conv=fsync"
else:
    server_command = f"netcat {IP} {PORT_CLIENT} -q 1 | dd
of={LOCATION}/{FILE_NAME} bs={bs} conv=fsync"
```

- **netcat** digunakan untuk mendengarkan data yang dikirim dari perangkat Android pada port tertentu (**PORT_CLIENT**).

- Data yang diterima akan disimpan di **LOCATION** dengan nama file yang ditentukan oleh **FILE_NAME**. Proses ini juga akan memastikan bahwa sistem menulis data ke file yang sama jika proses akuisisi dilanjutkan, menggunakan parameter **seek**.

3.2.7. Monitoring Proses dan Progress Akuisisi

Selama proses akuisisi berlangsung, sistem akan memonitor progres transfer data. Informasi seperti **total_size_kb** dari partisi yang diakuisisi digunakan untuk menghitung seberapa banyak data yang telah ditransfer dan memberikan estimasi waktu yang tersisa.

```
total_size_kb = int(getAcquisitionObject.physical.partition_size)
total_size_bytes = total_size_kb*1024
progress = (seek_skip_block / total_size_bytes) * 100 if
seek_skip_block > 0 else 0
```

- **progress** dihitung berdasarkan total byte yang telah ditransfer dibandingkan dengan ukuran total partisi, memberikan gambaran kepada pengguna mengenai seberapa jauh proses akuisisi telah berjalan.

3.2.8. Penanganan Timeout dan Transfer Rate

Sistem juga memantau laju transfer data dengan menghitung kecepatan transfer dan menerapkan mekanisme timeout jika proses berjalan terlalu lambat.

```
N = 4
MIN_TRANSFER_RATE = 1e5
recent_transfer_rates = [MIN_TRANSFER_RATE] * N

timeout_counter = 0
TIMEOUT_THRESHOLD = 15
```

- Sistem menyimpan kecepatan transfer terbaru untuk beberapa iterasi terakhir dan akan menghentikan proses jika kecepatan transfer di bawah ambang batas tertentu selama periode waktu yang telah ditentukan (**TIMEOUT_THRESHOLD**).

3.2.9. Logging Proses (Tambahan)

Setiap langkah penting dalam proses ini dicatat menggunakan logging, yang memberikan catatan detail tentang aktivitas yang terjadi selama proses akuisisi.

```
logging.basicConfig(filename="file.log", level=logging.INFO,
format='%(asctime)s - %(message)s')
```

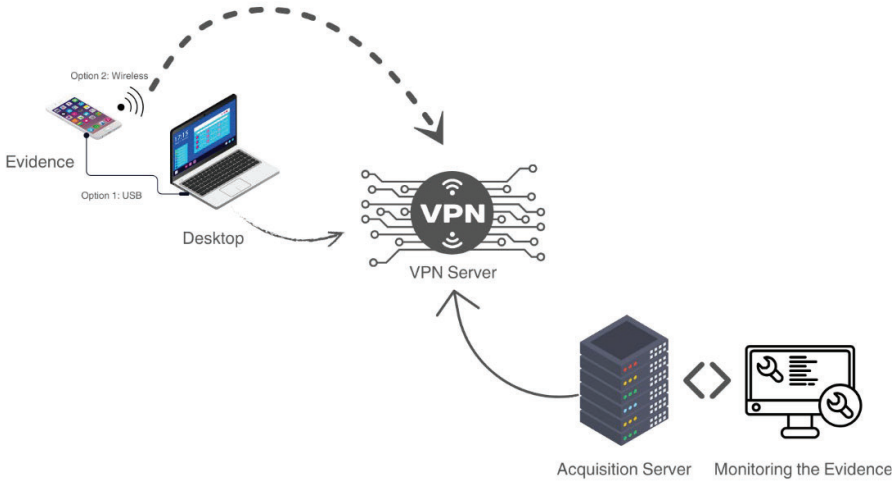
- Sistem menggunakan logging untuk mencatat setiap perintah yang dijalankan, status transfer, serta error yang mungkin terjadi, sehingga memudahkan dalam troubleshooting jika ada masalah.

3.3. Skema Topologi

Topologi jaringan pada Gambar 2 di bawah ini menunjukkan lingkungan jaringan, perangkat apa saja yang akan terlibat dan bagaimana interaksi antar perangkat tersebut. Nantinya modul utama aplikasi RAAS sendiri akan diinstall pada pada server akuisisi itu sediri.

Rancangan topologi jaringan ini juga didesain untuk memastikan bahwa data yang dikirimkan dalam kondisi aman. Hal ini diwujudkan dengan mengimplementasikan skema enkripsi dalam transfer data antara perangkat desktop dan *server* akuisisi. Untuk itu diperlukan skema jaringan VPN dengan penambahanVPN *server* pada topologi yang akan digunakan.

Setiap komponen dalam topologi ini akan dijelaskan sebagai berikut:



Gambar 2 - Contoh dari model topologi RAAS

3.3.1. Evidence

- **Deskripsi**: "Evidence" merujuk pada perangkat elektronik yang dapat menyimpan informasi, seperti smartphone dan komputer. Dalam konteks digital forensics, "Evidence" adalah sumber data yang mungkin berisi bukti terkait investigasi.
- **Koneksi**:
    - **Option 1: USB**: Evidence dapat terhubung ke Desktop melalui koneksi USB, yang biasanya digunakan untuk mentransfer data dengan cepat dan aman.

- **Option 2: Wireless**: Evidence juga memiliki opsi untuk terhubung secara nirkabel ke Desktop, yang bisa jadi lebih fleksibel namun mungkin kurang aman dibandingkan dengan koneksi USB.

3.3.2. Desktop

- **Deskripsi**: Desktop adalah komputer yang digunakan oleh investigator untuk mengakses dan menganalisis bukti.
- **Koneksi**:
  - Dihubungkan dengan Evidence melalui USB atau koneksi nirkabel.
  - Terhubung ke VPN Server, yang menunjukkan bahwa komunikasi antara Desktop dan server lainnya disandikan dan diamankan melalui Virtual Private Network (VPN).

3.3.3. VPN Server

- **Deskripsi**: VPN Server bertindak sebagai perantara untuk mengamankan dan menyandikan komunikasi antara Desktop dan server lainnya. Dan untuk jenis VPN Server yang digunakan pada kali ini adalah Tailscale VPN.
- **Koneksi**:
  - Menerima koneksi dari Desktop.
  - Mengarahkan trafik yang disandikan ke Acquisition Server.

3.3.4. Acquisition Server

- **Deskripsi**: Server ini berfungsi untuk mengumpulkan, menyimpan, dan mungkin menganalisis bukti yang dikirim dari Desktop.
- **Koneksi**:
  - Terhubung ke VPN Server untuk menerima data yang telah disandikan.
  - Mengirim data ke Monitoring the Evidence untuk tujuan pengawasan.

3.3.5. Monitoring the Evidence

- **Deskripsi**: Ini mungkin adalah sistem atau platform yang digunakan untuk memonitor dan melacak bukti yang dikumpulkan secara real-time.
- **Koneksi**:
    - Menerima data dari Acquisition Server.

Dengan rangkaian ini, tampaknya tujuan utamanya adalah untuk memastikan bahwa transfer bukti dilakukan dengan cara yang aman dan terenkripsi, mulai dari pengumpulan awal dari "Evidence" hingga pemantauannya di "Monitoring the Evidence". VPN Server memainkan peran kunci dalam memastikan integritas dan keamanan data selama proses ini.

3.4. Metode Prasyarat

RAAS (*Remote Acquisition for Android Smartphone*) merupakan inovasi dalam penelitian ini yang bertujuan untuk mengakuisisi data dari perangkat *Android* secara *remote*. Aplikasi ini dirancang menggunakan *framework* Django 4.0, dengan kemampuan utama untuk melakukan akuisisi perangkat melalui jaringan dengan cara yang efisien, aman, dan mudah.

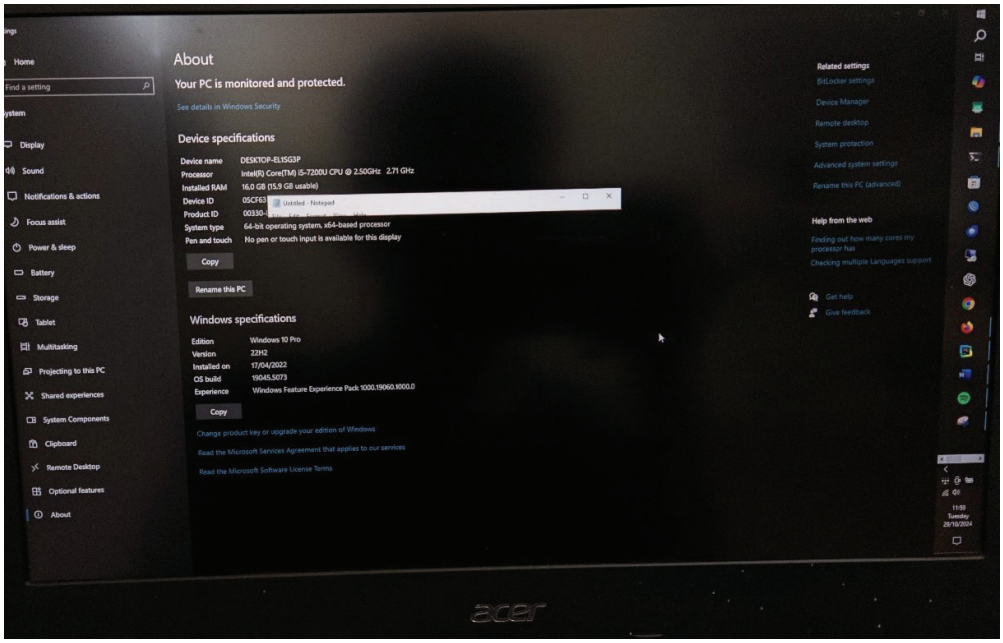
3.4.1. Persyaratan Akuisisi

1. Sebelum melakukan akuisisi dengan RAAS, perangkat target harus memenuhi kriteria tertentu, termasuk akses root dan USB *debugging* yang diaktifkan. Ini memastikan bahwa perangkat siap untuk proses akuisisi dan data yang dihasilkan adalah akurat dan dapat diandalkan.Perangkat yang akan diakuisisi harus:
    - Sudah di-*root*.
    - Mengaktifkan USB Debugging.
    - Tidak dalam keadaan mati saat proses berlangsung.
    - Terhubung dengan internet dan atau juga ke *Desktop* yang memiliki koneksi *internet*.

Akses *ROOT* pada perangkat sangat esensial untuk memastikan bahwa pengakuisisi dapat mengakses seluruh partisi dalam perangkat. Jika perangkat belum di-*root*, proses ini hanya akan menghasilkan data sesuai dengan akses pengguna biasa.

3.4.2. Spesifikasi Server Akuisisi

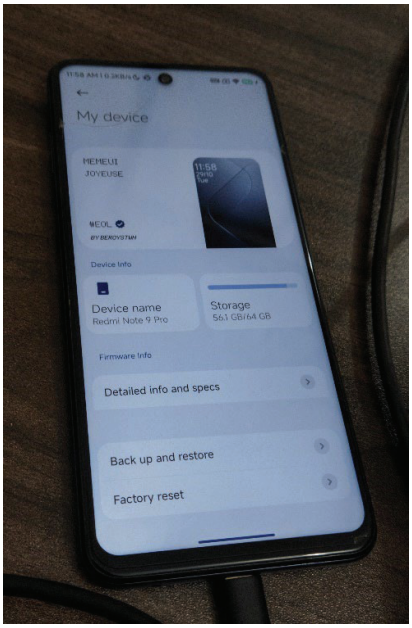
Server yang digunakan dalam akuisisi memiliki spesifikasi teknis yang mendukung performa aplikasi, termasuk *hardware* yang memadai dan *software* pendukung yang terkini



Gambar 3 - Laptop server

Berikut spesifikasi server akuisisi yang akan digunakan pada nanti:

A. Spesifikasi Perangkat Keras
    a. Laptop Processor Intel(R) Core(TM) i5-7200U @ 2.50GHz
    b. Memory 16GB
    c. SSD 250GB
B. Spesifikasi Perangkat Lunak
    a. Python 3.11.4
    b. Django 5.1
    c. FTK Imager
    d. Nmap 7.92
    e. Celery Worker
    f. Platform-tools by Google (ADB)



Gambar 4 - Evidence

3.4.3. Spesifikasi Evidence

Berikut spesifikasi evidence yang akan gunakan pada tahap akuisisi nanti:

A. Spesifikasi Perangkat Keras
   a. Xiaomi Redmi Note 9 Pro - Unlocked
   b. Memory 6GB
   c. Internal Storage 64GB

B. Spesifikasi Perangkat Lunak
   a. Android 13 (Custom ROM)
   b. Magisk 26.1 – For Root

4. PENGUJIAN DAN HASIL

4.1. Kaidah Forensik Dan Proses Pengujian

Dalam pengujian sistem RAAS, kaidah forensik memainkan peran penting untuk memastikan integritas dan validitas hasil akuisisi data. Terdapat empat tahap utama dalam proses forensik digital yang dijadikan dasar dalam pengujian sistem, yaitu:

1. Koleksi
    - Proses ini mencakup pengumpulan bukti digital dari perangkat evidence menggunakan metode yang dapat menjaga keutuhan data. Bukti ini dapat berupa perangkat keras atau perangkat lunak, serta informasi yang tersimpan di dalamnya.

2. Identifikasi
    - Pada tahap ini, jenis dan spesifikasi bukti digital diidentifikasi. Sistem RAAS memerlukan proses identifikasi yang tepat untuk memahami kemampuan perangkat dan data yang tersimpan, sebelum proses akuisisi dilakukan.

3. Akuisisi
    - Akuisisi adalah proses penyalinan data digital dari perangkat bukti ke media penyimpanan yang aman. Pada proses ini, penting untuk menjaga keutuhan data dan memastikan tidak ada perubahan yang terjadi selama transfer.

4. Presentasi
    - Hasil dari akuisisi kemudian disajikan untuk dianalisis lebih lanjut menggunakan tools forensik seperti FTK Imager. Tahap ini membantu verifikasi integritas dan kelengkapan data untuk tujuan investigasi.

Kaidah forensik ini akan menjadi panduan dalam proses pengujian untuk memastikan bahwa setiap proses yang dilakukan dalam sistem RAAS mengikuti standar forensik yang berlaku.

4.2. Metode Pengujian Dengan Kaidah Forensik

Untuk mengevaluasi kinerja aplikasi RAAS, berbagai skenario pengujian dilakukan dengan mempertimbangkan prinsip-prinsip kaidah forensik yang telah disebutkan di atas. Setiap skenario bertujuan untuk menguji fitur utama sistem RAAS, meliputi kecepatan akuisisi, resume akuisisi, dan verifikasi integritas data.

4.3. Skenario Pengujian

Pengujian dilakukan dengan tiga skenario yang masing-masing mengacu pada fitur aplikasi utama. Berikut adalah rincian dari skenario yang dilakukan:

Tabel 5 - Fitur-fitur yang akan diuji

No	Fitur yang Diuji
1	Kecepatan Akuisisi (USB dan Wireless)
2	Resume Akuisisi
3	Verifikasi Integritas Data
4	Presentasi Hasil Akuisisi dengan FTK Imager

4.3.1. Skenario 1: Kecepatan Akuisisi

Pada skenario ini, dilakukan pengujian terhadap **kecepatan akuisisi data** dari perangkat Android menggunakan dua jenis koneksi yang berbeda: **USB** dan **Wireless (ADB IP Connect)**. Tujuan pengujian ini adalah untuk mengetahui apakah ada perbedaan signifikan dalam waktu akuisisi data antara kedua metode koneksi tersebut, dan untuk menentukan metode mana yang lebih efisien.

Parameter yang Diuji:

- **Kecepatan akuisisi** berdasarkan total waktu yang dibutuhkan untuk mengakuisisi partisi atau data dari perangkat Android ke server forensik.
- **Koneksi yang diuji**:
    - **USB** (koneksi fisik melalui kabel USB).
    - **Wireless (ADB IP Connect)**, yang menghubungkan perangkat menggunakan alamat IP melalui jaringan wireless.

Cara Pengujian:

1. **Setup Koneksi USB**
    - Hubungkan perangkat evidence (Android) ke komputer akuisisi menggunakan kabel USB.
    - Pastikan perangkat dikenali oleh sistem melalui ADB dengan perintah:

      ```
      adb devices
      ```

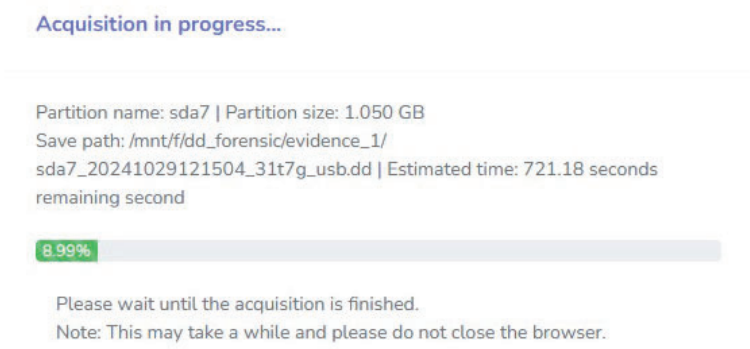
    - Nilai hash yang dihasilkan dari partisi akan disimpan dalam file **partition_hash.txt** di perangkat Android, yang kemudian akan diekspor ke server untuk digunakan sebagai pembanding setelah proses akuisisi selesai.

2. **Setup Koneksi Wireless (ADB IP Connect)**
    - Hubungkan perangkat evidence ke komputer akuisisi menggunakan jaringan wireless melalui alamat IP.
    - Untuk menghubungkan perangkat Android melalui ADB IP Connect, gunakan perintah berikut:

```
adb tcpip 5555
adb connect <IP_ADDRESS>:5555
adb devices
```

    - Perangkat harus terhubung ke jaringan yang sama dengan server dan bisa dikenali oleh **ADB**.

3. **Proses Akuisisi Data**



Gambar 5 - Proses akuisisi

    - Pilih partisi tertentu pada perangkat Android (pada pengujian kali ini, kita menggunakan partisi /dev/block/sda7 dan sda9) yang akan diakuisisi. Partisi ini sama untuk kedua jenis koneksi agar hasil perbandingan akurat.
    - Lakukan proses akuisisi dengan metode yang sama untuk kedua koneksi (USB dan Wireless), dan catat waktu mulai dan waktu selesai akuisisi.

4. **Pengukuran Kecepatan**
    - Gunakan **time.time()** untuk mencatat waktu mulai dan selesai dari proses akuisisi, lalu hitung selisih waktu untuk mendapatkan total durasi akuisisi.
    - Contoh pengukuran timing dalam kode Python:

```
start_time = time.time()
.
# Jalankan proses akuisisi
.
```

```
print("--- Total Time: %s seconds ---" % (time.time() -
start_time))
```

5. **Perbandingan Kecepatan Akuisisi**
    - Hasil dari kedua koneksi akan dibandingkan berdasarkan durasi akuisisi. Jika metode USB secara signifikan lebih cepat, ini menunjukkan bahwa USB lebih efisien untuk akuisisi, terutama untuk akuisisi data dalam jumlah besar. Jika perbedaan waktu tidak signifikan, wireless (ADB IP Connect) dapat menjadi alternatif yang nyaman.

4.3.2. Skenario 2: Resume Akuisisi

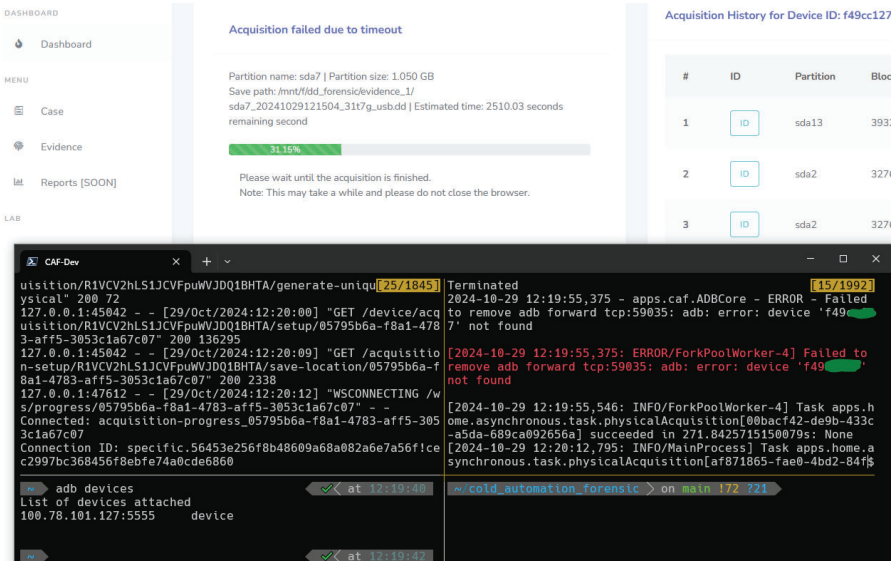
Pada skenario ini, sistem RAAS diuji untuk memastikan kemampuannya melanjutkan proses akuisisi yang terganggu (resume akuisisi). Saat proses akuisisi sedang berlangsung, koneksi antara perangkat evidence dan server mungkin terputus baik karena pemutusan kabel USB atau gangguan pada koneksi WiFi. Sistem harus dapat mendeteksi gangguan ini dan memungkinkan proses dilanjutkan dari titik terakhir data berhasil diambil.

Parameter yang Diuji

- Kemampuan sistem untuk melanjutkan (resume) proses akuisisi data setelah terjadi gangguan koneksi.
- Mekanisme deteksi gangguan dan pemulihan proses akuisisi berdasarkan byte terakhir yang berhasil ditransfer.

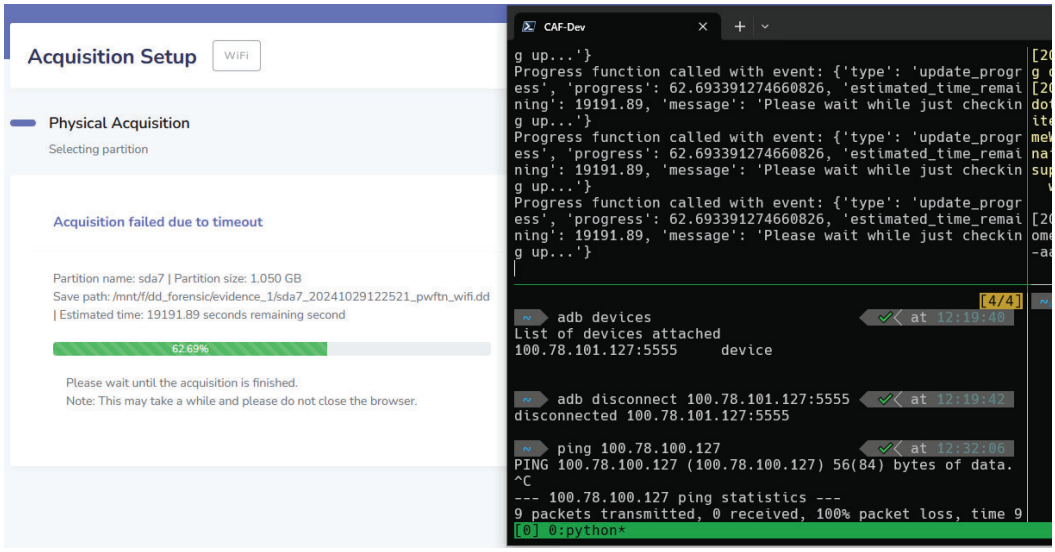
Cara Pengujian:

1. **Simulasi Gangguan Koneksi**



Gambar 6 - Cabut USB ketika akuisisi sedang berlangsung

- **Untuk koneksi USB**, kabel USB antara perangkat evidence dan server akan dicabut secara paksa selama proses akuisisi.



Gambar 7 - Diskonek ketika akuisisi sedang berlangsung

- **Untuk koneksi Wireless (ADB IP Connect)**, perintah *adb disconnect* dan pemutusan koneksi internet (WiFi atau VPN) akan digunakan untuk memutuskan koneksi wireless antara perangkat evidence dan server:

```
adb disconnect <IP_ADDRESS>:5555
```

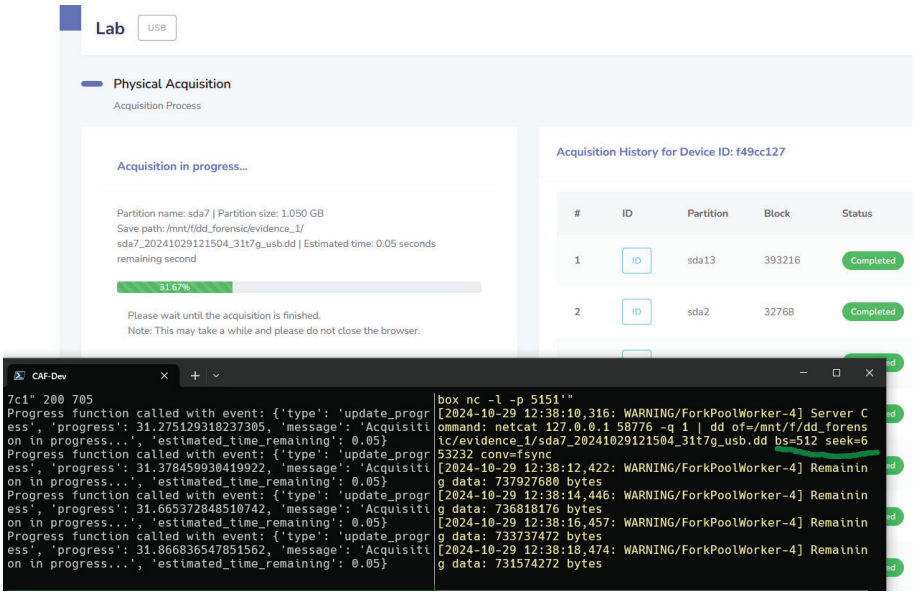
- Sistem akan mendeteksi bahwa transfer data telah berhenti berdasarkan pengukuran ukuran file yang dihasilkan, serta mengamati kecepatan transfer yang turun menjadi nol.

2. **Deteksi Gangguan dan Pengaturan Timeout**
   - Sistem secara berkala memeriksa ukuran file yang sedang diakuisisi dan membandingkannya dengan ukuran partisi asli.
   - Jika transfer data berhenti dan tidak ada data yang masuk selama **15 detik** (batas waktu yang ditentukan dalam **TIMEOUT_THRESHOLD**), sistem akan menganggap proses sedang mengalami gangguan atau telah terhenti.
   - Jika ukuran file hasil akuisisi belum memenuhi total ukuran partisi asli setelah batas waktu habis, proses akan dianggap gagal:

```
if timeout_counter >= TIMEOUT_THRESHOLD and all(rate == 0
for rate in recent_transfer_rates) and remaining_data > 0:
    getAcquisitionObject.status = 'failed'
    getAcquisitionObject.last_active = datetime.now()
    getAcquisitionObject.save()
```

3. Mekanisme Resume Akuisisi

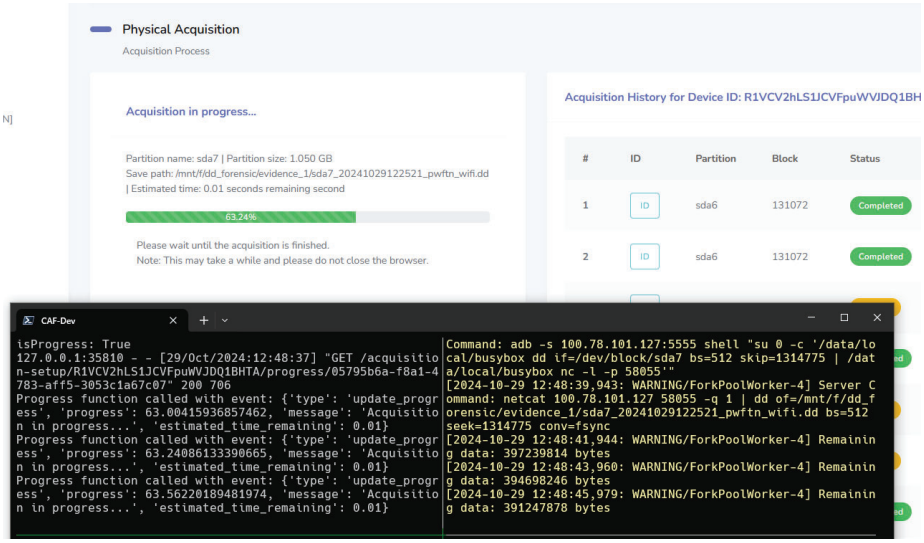
- Setelah proses dianggap gagal, pengguna dapat memulai kembali akuisisi dengan mengakses tautan akuisisi yang sama.



Gambar 8 - Resume akuisisi dari koneksi USB

- Sistem akan **menghitung jumlah byte yang sudah berhasil ditransfer** dan memulai proses akuisisi dari posisi terakhir sebelum gangguan (dengan memanfaatkan parameter **skip** pada perintah *dd* untuk melewati byte yang sudah diambil):

```
seek_skip_block = getAcquisitionObject.physical.total_transferred_bytes or 0
seek_blocks = seek_skip_block // bs
```



Gambar 9 - Resume akuisisi dari koneksi wireless

- Sistem kemudian akan memulai proses kembali dari posisi terakhir yang tersimpan, memastikan bahwa tidak ada duplikasi data selama proses akuisisi.

4. **Verifikasi Hash untuk Menjamin Keutuhan Data**
    - Setelah proses resume selesai dan data telah diambil sepenuhnya, sistem akan melakukan hashing **SHA256** untuk memverifikasi integritas data yang dihasilkan.
    - Hash yang dihasilkan dari proses akuisisi awal dan hash setelah resume akan dibandingkan untuk memastikan bahwa data yang diambil tetap utuh dan tidak rusak selama proses resume.

5. **Pemberian Laporan Status ke Pengguna**
    - Sistem akan mengirimkan status dan progres secara berkala kepada pengguna selama proses akuisisi berlangsung, termasuk pemberitahuan jika terjadi gangguan dan ketika proses resume berhasil dimulai kembali.
    - Setelah akuisisi selesai, sistem akan membuat laporan akhir yang mencakup detail informasi evidence dan hasil hashing.

4.3.3. Skenario 3: Verifikasi Integritas Data

Pada skenario ini, dilakukan verifikasi terhadap integritas data yang telah diakuisisi dengan menggunakan metode hashing **SHA256**. Proses ini dilakukan dengan membandingkan nilai hash dari partisi sebelum akuisisi (diambil langsung dari perangkat evidence) dengan hash dari data yang sudah diakuisisi ke server. Jika hasil hash sebelum dan sesudah akuisisi sesuai, maka integritas data dapat dipastikan.

Parameter yang Diuji:

- Integritas data dari partisi yang diakuisisi, yaitu memastikan bahwa data yang diambil tidak mengalami perubahan atau korupsi selama proses akuisisi.

Cara Pengujian:

1. **Hashing Partisi di Perangkat Evidence (Android)**

    - Sebelum proses akuisisi dimulai, nilai hash dari partisi perangkat evidence akan dihitung menggunakan algoritma **SHA256**.

    - Perintah ini akan dieksekusi di perangkat Android melalui **ADB (Android Debug Bridge)** dengan menjalankan perintah **SHA256** untuk menghitung hash dari partisi yang akan diakuisisi.

Contoh perintah yang digunakan di Android untuk melakukan hash:

```
adb shell su 0 -c "sha256sum /dev/block/{PARTITION}" >
/sdcard/partition_hash.txt
```

- Nilai hash yang dihasilkan dari partisi akan disimpan dalam file **partition_hash.txt** di perangkat Android, yang kemudian akan diekspor ke server untuk digunakan sebagai pembanding setelah proses akuisisi selesai.

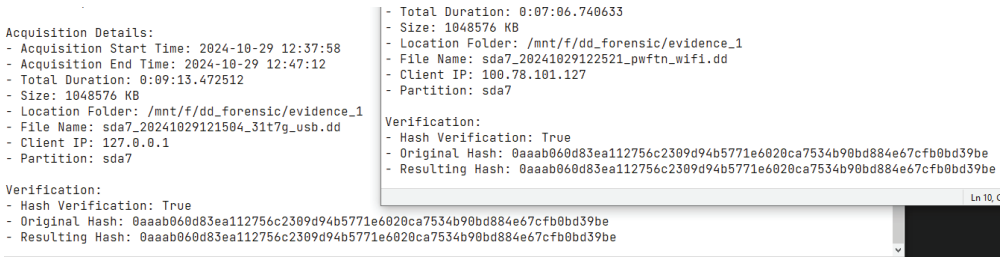
2. **Akuisisi Partisi**
   - Setelah nilai hash dari partisi tersimpan, proses akuisisi dilakukan dengan menyalin seluruh partisi ke server menggunakan metode yang sudah dijelaskan pada skenario sebelumnya.

3. **Hashing Data Setelah Akuisisi di Server**
   - Setelah data berhasil diakuisisi ke server, hash dari data tersebut akan dihitung kembali di server menggunakan algoritma **SHA256** yang sama.

   Contoh perintah hashing pada data yang sudah diakuisisi di server:

```
sha256sum {FILE_LOCATION}/{ACQUIRED_PARTITION_IMAGE}.dd
```

   - Nilai hash dari file hasil akuisisi akan dibandingkan dengan nilai hash yang diambil dari perangkat evidence sebelum akuisisi dimulai.



Gambar 10 - Validasi nilai hash

4. **Perbandingan Hasil Hash**
   - Nilai hash dari partisi di perangkat evidence (sebelum akuisisi) dan nilai hash dari data yang telah diakuisisi di server (sesudah akuisisi) akan dibandingkan. Jika nilai hash identik, integritas data dapat dipastikan terjaga dengan baik selama proses akuisisi.
   - Jika ada perbedaan dalam nilai hash, berarti terjadi perubahan atau korupsi data selama proses akuisisi, dan tindakan lebih lanjut akan diperlukan.

4.3.4. Skenario 4: Presentasi Hasil Akuisisi dengan FTK Imager

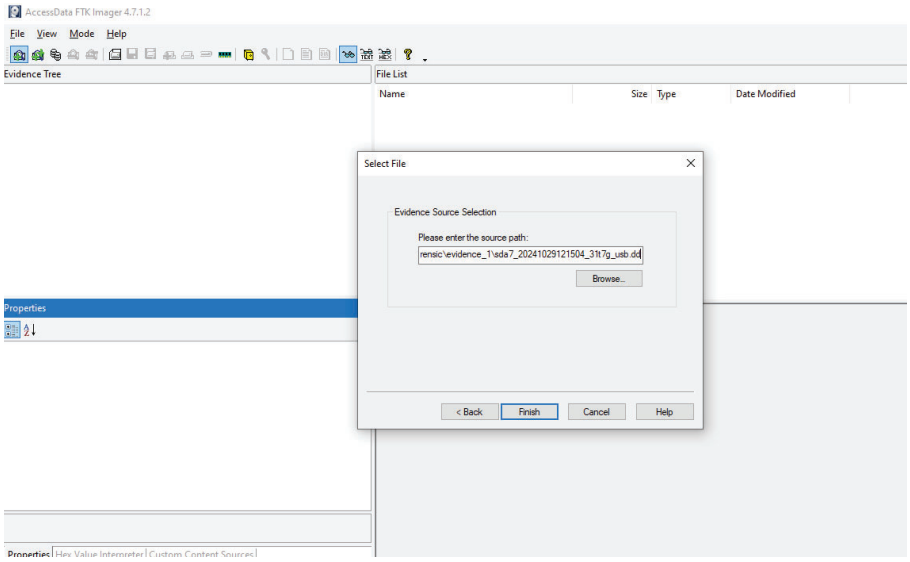
Pengujian ini bertujuan untuk memastikan bahwa hasil akuisisi yang dihasilkan oleh sistem RAAS dapat dibuka dan dianalisis menggunakan **FTK Imager**. Hal ini diperlukan untuk memenuhi kaidah forensik keempat, yaitu

presentasi, yang memastikan bahwa bukti digital dapat ditampilkan dan diperiksa secara forensik oleh aplikasi yang diakui dan memenuhi standar.

Parameter yang Diuji:

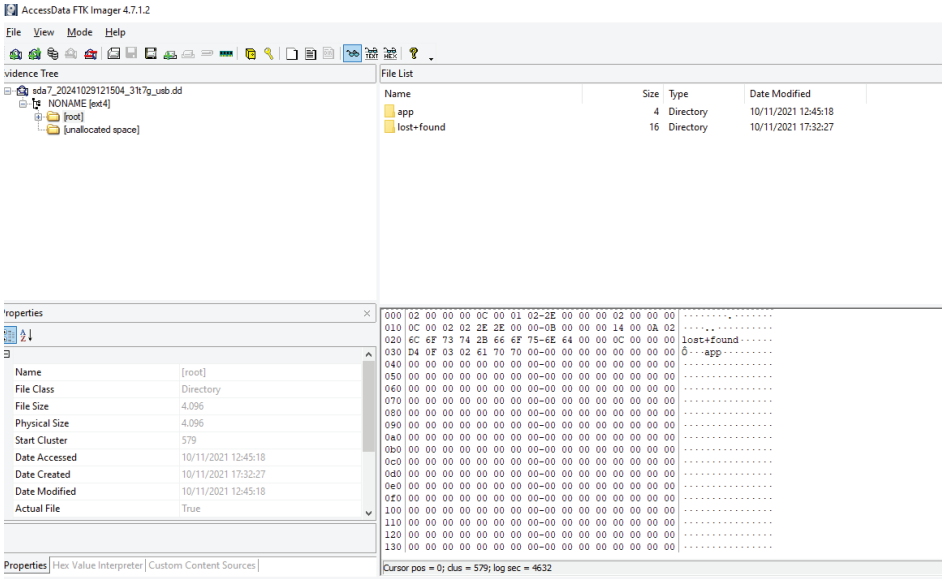
- **Kompatibilitas hasil akuisisi** dengan FTK Imager untuk memastikan bahwa file hasil akuisisi dapat dibaca tanpa adanya kesalahan atau korupsi data.
- **Kelengkapan data** di dalam FTK Imager, yang memastikan bahwa semua file dan struktur direktori tetap utuh.

Cara Pengujian:



Gambar 11 - Pilih file DD dengan FTK Imager

1. **Buka File Hasil Akuisisi di FTK Imager**
    - Setelah proses akuisisi selesai, buka FTK Imager dan pilih opsi untuk menambahkan **disk image** atau file hasil akuisisi yang telah dibuat oleh sistem RAAS.
    - Gunakan opsi **"Add Evidence Item"** dan pilih jenis evidence sebagai **"Image File"**.
    - Arahkan ke lokasi file hasil akuisisi, misalnya **image.dd**, yang dihasilkan oleh sistem RAAS.



Gambar 12 - Struktur file image

2. **Verifikasi Kelengkapan Data**
    - Setelah file dibuka, lakukan inspeksi terhadap **struktur direktori** dan **file** yang ada di dalam image tersebut. Pastikan bahwa semua data yang ada di perangkat evidence sebelum akuisisi tetap ada dan tidak ada file yang hilang.
    - Catat hasil inspeksi untuk melihat apakah ada kesalahan atau notifikasi dari FTK Imager yang menandakan adanya korupsi atau masalah dengan image file.

4.4. Hasil Pengujian

Dalam bagian ini adalah hasil dari proses pengujian terkait skenario yang telah diuji coba pada aplikasi RAAS ini.

4.4.1. Hasil Pengujian Skenario 1: Kecepatan Akuisisi (USB dan Wireless)

Pengujian ini bertujuan untuk mengukur waktu akuisisi data dari perangkat Android melalui dua jenis koneksi yang berbeda: **USB** dan **Wireless (ADB IP Connect)**. Berikut adalah hasil pengujian yang menunjukkan perbandingan kecepatan akuisisi:

Tabel 6 - Perbandingan skenario 1: Kecepatan akuisisi

Partisi	Metode	Ukuran (MB)	Waktu (Detik)	Kecepatan (MBps)
sda7	USB	1048.57	137	7.653
sda6	USB	131.07	57	2.299

sda13	USB	393.216	89	4.418
sda7	Wireless	1048.57	341	3.075
sda6	Wireless	131.07	86	1.524
sda13	Wireless	393.216	171	2.299

Analisis Hasil

1. **Perbedaan Kecepatan Akuisisi pada USB dan Wireless**

    - **Metode USB** memiliki **kecepatan akuisisi yang lebih tinggi** dibandingkan Wireless (ADB IP Connect). Hal ini terlihat dari waktu akuisisi yang lebih singkat saat menggunakan koneksi USB. Sebagai contoh, untuk partisi **sda7** (1.048 MB), akuisisi melalui USB membutuhkan waktu **137 detik** dengan kecepatan rata-rata **7.653 MBps**, sedangkan Wireless membutuhkan waktu **341 detik** dengan kecepatan rata-rata hanya **3.075 MBps**.
    - **Wireless (ADB IP Connect)** memiliki kecepatan yang lebih lambat karena adanya overhead pada protokol jaringan wireless, yang dipengaruhi oleh latensi jaringan, kecepatan koneksi WiFi, dan stabilitas streaming data.

2. **Ketidakkonsistenan Kecepatan Akuisisi (Pada Metode yang Sama)**
    - Meskipun menggunakan metode koneksi yang sama (baik USB maupun Wireless), kecepatan akuisisi pada partisi dengan ukuran berbeda tidak selalu konsisten. Sebagai contoh:
        - **Metode USB**: Partisi **sda6** (131.07 MB) memiliki kecepatan **2.299 MBps**, sedangkan partisi **sda13** (393.216 MB) memiliki kecepatan **4.418 MBps**.
        - **Metode Wireless**: Partisi **sda6** memiliki kecepatan **1.524 MBps**, sedangkan partisi **sda13** memiliki kecepatan **2.299 MBps**.

Penyebab Ketidakkonsistenan Kecepatan

Ketidakkonsistenan kecepatan akuisisi disebabkan oleh:

1. **Penghitungan Waktu yang Melibatkan Tahap Akhir Proses (Hashing dan Report Generation):**
    - Waktu total yang dicatat tidak hanya mencakup proses transfer data dari perangkat evidence ke server, tetapi juga mencakup waktu tambahan untuk:
        - **Hashing SHA256** terhadap hasil akuisisi untuk memverifikasi integritas data.
        - **Pembuatan laporan hasil akuisisi**.

- Partisi dengan ukuran kecil tetap mengalami tambahan waktu dari proses ini, sehingga kecepatan rata-rata terlihat lebih rendah dibandingkan partisi dengan ukuran besar.

2. **Mekanisme Threshold untuk Verifikasi Streaming Data:**
   - Sistem menggunakan mekanisme **threshold sebanyak 15x2 detik (30 detik)** untuk memastikan bahwa **streaming data dari netcat** sudah selesai sebelum proses hashing dimulai. Proses ini dilakukan untuk memverifikasi apakah seluruh data telah berhasil diambil tanpa kehilangan.
   - Partisi dengan ukuran kecil tetap harus melalui proses threshold ini, yang menyebabkan waktu akuisisi lebih lama dibandingkan ukuran data sebenarnya. Hal ini menjelaskan mengapa partisi kecil, seperti **sda6**, memiliki kecepatan yang lebih rendah meskipun menggunakan metode yang sama.

3. **Overhead pada Streaming Data (Netcat):**
   - Netcat yang digunakan untuk streaming data memiliki overhead yang bervariasi tergantung pada stabilitas koneksi. Hal ini lebih terlihat pada koneksi **Wireless**, yang cenderung memiliki fluktuasi transfer rate dibandingkan USB.

4.4.2. Hasil Pengujian Skenario 2: Resume Akuisisi

Pengujian ini mengukur kemampuan sistem untuk melanjutkan proses akuisisi yang terganggu akibat pemutusan koneksi, baik secara fisik (USB) maupun melalui WiFi (ADB IP Connect). Berikut adalah hasil pengujian dari beberapa kali percobaan dengan kondisi terputus dan di-resume:

Tabel 7 - Perbandingan skenario 2: Resume akuisisi

No	Koneksi	Status Awal	Gangguan Koneksi	Status Setelah Resume	Hash Konsisten
1	USB	Sedang Berjalan	Kabel dicabut	Berhasil dilanjutkan	Ya
2	USB	Sedang berjalan	Kabel dicabut	Berhasil dilanjutkan	Ya
3	USB	Sedang berjalan	Pemutusan koneksi	Berhasil dilanjutkan	Ya
4	Wireless	Sedang berjalan	Pemutusan koneksi	Berhasil dilanjutkan	Ya

3	Wireless	Sedang berjalan	Pemutusan koneksi	Berhasil dilanjutkan	Ya
4	Wireless	Sedang berjalan	Pemutusan koneksi	Berhasil dilanjutkan	Ya

Analisis Hasil

- Sistem berhasil mendeteksi ketika transfer data berhenti dan menginisialisasi mode resume untuk melanjutkan akuisisi dari byte terakhir yang berhasil ditransfer.
- Pada semua pengujian, sistem berhasil melanjutkan proses tanpa kehilangan data. **Hashing** sebelum dan sesudah akuisisi menunjukkan hasil yang konsisten, membuktikan bahwa integritas data terjaga.
- **Kesimpulan**: Fitur resume pada sistem RAAS bekerja dengan baik, memastikan bahwa gangguan sementara pada koneksi tidak mengharuskan proses akuisisi diulang dari awal.

4.4.3. Hasil Pengujian Skenario 3: Verifikasi Integritas Data

Pengujian ini memastikan bahwa data yang diakuisisi dari perangkat evidence tidak mengalami perubahan atau korupsi selama proses akuisisi. Metode yang digunakan adalah dengan membandingkan hash SHA256 sebelum dan setelah akuisisi. Berikut adalah hasilnya:

Tabel 8 - Perbandingan skenario 3: Verifikasi integritas data

No	Ukuran (MB)	Hash Sebelum Akuisisi	Hash Setelah Akuisisi	Status Integritas
1	1048.57	0aaab060d83ea112756c2309d94b5771e6020ca7534b90bd884e67cfb0bd39be	0aaab060d83ea112756c2309d94b5771e6020ca7534b90bd884e67cfb0bd39be	Konsisten
2	393.21	f952503393796004d4de17ec41099fd2a261b74820e77f619bc1709f7b1f1437	f952503393796004d4de17ec41099fd2a261b74820e77f619bc1709f7b1f1437	Konsisten
3	131.07	0a9b293b28180519155d200b85f53b894b62b4c8355339c8d8b54ad41d3a4799	0a9b293b28180519155d200b85f53b894b62b4c8355339c8d8b54ad41d3a4799	Konsisten

Analisis Hasil

- Hasil hash SHA256 sebelum dan setelah proses akuisisi menunjukkan bahwa semua nilai hash adalah **identik**, memastikan bahwa tidak ada perubahan data selama proses transfer.

- Hal ini membuktikan bahwa integritas data terjaga sepanjang proses akuisisi, baik dalam kondisi normal maupun setelah proses resume.
- **Kesimpulan**: Proses akuisisi pada sistem RAAS menjaga keutuhan dan integritas data sesuai dengan standar forensik digital yang diperlukan.

4.4.4. Hasil Pengujian Skenario 4: Presentasi Hasil Akuisisi dengan FTK Imager

Pengujian ini memastikan bahwa hasil data yang diakuisisi dari perangkat evidence bisa dibuka atau digunakan oleh aplikasi yang sudah berstandar atau berlisensi komersial seperti FTK Imager. Berikut adalah hasilnya:

Tabel 9 - Perbandingan skenario 4: Presentasi hasil akuisisi dengan FTK Imager

No	Ukuran (MB)	Status Pembukaan di FTK Imager	Kelengkapan Data	Integritas Data
1	1048.57	Berhasil	Utuh (tidak ada data hilang)	Konsisten
2	393.21	Berhasil	Utuh (tidak ada data hilang)	Konsisten
3	131.07	Berhasil	Utuh (tidak ada data hilang)	Konsisten

Analisis Hasil

- Hasil pengujian menunjukkan bahwa semua file hasil akuisisi dapat dibuka dengan baik di **FTK Imager**, tanpa adanya kesalahan atau pesan peringatan.
- Struktur direktori dan semua file yang diakuisisi dari perangkat evidence terlihat utuh dan konsisten saat dilihat di dalam FTK Imager.
- **Kesimpulan**: Hasil akuisisi dari sistem RAAS kompatibel dengan **FTK Imager**, memastikan bahwa output dapat digunakan dalam proses investigasi forensik yang memerlukan analisis lebih lanjut menggunakan tools standar.

5. PENUTUP

5.1. Kesimpulan

Penelitian ini berhasil mengembangkan sebuah **sistem akuisisi data forensik digital berbasis web**, yang dirancang untuk memenuhi berbagai kebutuhan dalam **forensik digital**. Berdasarkan pengujian dan evaluasi yang dilakukan, sistem RAAS terbukti dapat menjalankan proses forensik digital yang mencakup empat kaidah utama: **Koleksi, Identifikasi, Akuisisi, dan Presentasi**.

Dari hasil pengujian terhadap **kecepatan akuisisi, kemampuan resume, verifikasi integritas data, serta kompatibilitas hasil akuisisi dengan FTK Imager**, dapat disimpulkan bahwa sistem ini:

1. **Efektif dan fleksibel** dalam mengumpulkan serta mengakuisisi data dari perangkat evidence melalui berbagai metode koneksi, termasuk USB dan Wireless.
2. **Menjamin integritas data** dengan menggunakan metode hashing untuk verifikasi, serta memastikan tidak ada perubahan pada data yang diakuisisi.
3. **Kompatibel dengan tools forensik standar** seperti FTK Imager, sehingga hasil akuisisi dapat digunakan untuk investigasi lebih lanjut.

Sistem ini juga menunjukkan **keamanan** dan **keterjangkauan** melalui platform berbasis web, memungkinkan akses yang lebih mudah dan efisien dibandingkan dengan solusi forensik tradisional. Berdasarkan penelitian ini, terbukti bahwa belum ada aplikasi serupa versi web yang secara khusus memenuhi semua kaidah forensik digital, menjadikan RAAS sebagai solusi inovatif di bidang ini.

5.2. Keterbatasan

Khusus penelitian ini, meskipun sistem RAAS menunjukkan hasil yang memuaskan, terdapat beberapa keterbatasan yang perlu dicatat, seperti:

- **Keterbatasan koneksi wireless**, yang dapat menyebabkan kecepatan akuisisi lebih lambat dibandingkan dengan metode USB.
- **Kemampuan resume** yang bergantung pada jaringan yang stabil, terutama saat menggunakan koneksi WiFi.

5.3. Rekomendasi

Untuk pengembangan lebih lanjut, beberapa rekomendasi yang dapat dipertimbangkan antara lain:

1. **Peningkatan stabilitas koneksi wireless**, dengan implementasi mekanisme penanganan gangguan yang lebih canggih (seperti overlay).
2. **Dukungan lebih luas terhadap berbagai perangkat evidence**, termasuk sistem operasi selain Android.
3. **Peningkatan fitur keamanan**, terutama dalam hal enkripsi data selama proses akuisisi, guna memastikan kerahasiaan informasi yang diambil.

Dengan solusi ini, RAAS diharapkan dapat terus berkembang menjadi **platform akuisisi data forensik digital** yang tidak hanya memenuhi standar forensik, tetapi juga menjawab kebutuhan pasar yang terus berubah, khususnya di bidang keamanan dan analisis data digital.