

Perancangan Aplikasi Sistem Deteksi *Evidence* Otomatis Menggunakan Algoritma SHA-256 Berbasis *Appsheets*

Junawan Ahmad Fatoni¹, dan Dandun Widhiantoro²

Program Studi Broadband Multimedia, Jurusan Teknik Elektro, Politeknik Negeri Jakarta,
Jl. Prof. DR. G.A. Siwabessy, Kampus Universitas Indonesia, Kota Depok, 16425, Indonesia

E-mail: junawan ahmad fatoni.te21@mhs.wpnj.ac.id , dandun@elektro.pnj.ac.id

Abstrak

Proses verifikasi foto *evidence* pekerjaan layanan *maintenance* kabel tanah fiber optik masih dilakukan secara manual, sehingga rawan kesalahan dan kurang efisien. Penelitian ini mengembangkan sistem validasi data otomatis untuk meningkatkan akurasi dan efisiensi verifikasi. Aplikasi dibangun menggunakan platform *Appsheets*, memungkinkan pengembangan tanpa pengkodean kompleks dan mudah diakses oleh *user* atau tim lapangan. Untuk menjaga integritas foto, digunakan metode *hashing* SHA-256 yang menghasilkan kode unik setiap foto sepanjang 64 karakter. Selain itu, teknologi *Optical Character Recognition* (OCR) dimanfaatkan untuk membantu membaca teks unik pada foto yang diunggah sesuai standar Perusahaan guna tidak ada manipulasi foto. Sistem juga dilengkapi dengan *WhatsApp Gateway* untuk pengiriman notifikasi *real-time* kepada *user* dan *admin* Ketika foto yang diunggah ganda. Hasil pengujian menunjukkan SHA-256 mampu mendeteksi perubahan data dengan konsisten 100%, OCR mencapai akurasi 100% dari satu sampel foto, dan *WhatsApp Gateway* berhasil mengirim notifikasi dengan tingkat keberhasilan 100% dan waktu rata-rata kurang dari lima detik. Sistem ini terbukti efektif mempercepat proses validasi, mengurangi kesalahan manusia, serta meningkatkan transparansi dan akuntabilitas dalam layanan *maintenance* kabel tanah fiber optik.

Kata Kunci: *Appsheets; Hashing; Whatssap Gateway; OCR; SHA-256*

Abstract

The verification process of photo evidence for underground fiber optic cable maintenance work is still carried out manually, making it prone to errors and inefficient. This study develops an automated data validation system to improve the accuracy and efficiency of verification. The application is built using the Appsheets platform, enabling development without complex coding and making it easily accessible to users or field teams. To maintain the integrity of the photos, the SHA-256 hashing method is used, producing a unique 64-character code for each photo. Additionally, Optical Character Recognition (OCR) technology is utilized to assist in reading unique text on the uploaded photos according to company standards, preventing photo manipulation. The system is also equipped with a WhatsApp Gateway to send real-time notifications to users and administrators when duplicate photos are uploaded. Test results show that SHA-256 consistently detects data changes with 100% accuracy, OCR achieves 100% accuracy from a single photo sample, and the WhatsApp Gateway successfully sends notifications with a 100% success rate and an average time of less than five seconds. This system has proven effective in accelerating the validation process, reducing human errors, and enhancing transparency and accountability in underground fiber optic maintenance services.

Keywords: *Appsheets; Hashing; Whatssap Gateway; OCR; SHA-256*

1. Pendahuluan

Dalam era *modern* saat ini pengembangan sistem otomasi sangat diperlukan karena dapat mempercepat proses pekerjaan khususnya di bidang industri [1]. Dengan sistem otomasi proses yang membutuhkan waktu panjang dalam penyelesaiannya dapat dipersingkat dengan

adanya teknologi yang semakin mutakhir [2]. Salah satu perkembangan sistem deteksi otomatis, sistem deteksi otomatis adalah sistem yang dirancang untuk mendeteksi atau mengenali kesalahan, pola, objek, atau kondisi tertentu secara otomatis tanpa memerlukan intervensi manual dari manusia.

Koperasi Pegawai Indosat (KOPINDOSAT) merupakan Perusahaan yang bergerak dibidang jasa salah satunya pemeliharaan fiber optik. Saat ini KOPINDOSAT mempunyai tim lapangan yang tersebar di seluruh pulau Jawa sekitar 40 pekerja lapangan. Tim lapangan tersebut memiliki kewajiban administratif yaitu *update* pekerjaan mereka di aplikasi AONE. Aplikasi AONE adalah aplikasi yang dibuat untuk memfasilitasi tim lapangan membuat laporan pekerjaan [3].

Proses pengunggahan data yang tidak disaring secara otomatis merupakan salah satu proses yang memiliki resiko *human error* yang tinggi dikarenakan dibutuhkan tingkat ketelitian dan kecermatan yang tinggi dari seorang *user* [4]. *Human error* tersebut akan terlihat ketika *user* mengunggah data dalam jumlah banyak di waktu yang sama. *User* tersebut akan merasakan lelah dan dapat meningkatkan resiko kesalahan ketika mengunggah data sehingga dapat menyebabkan terjadinya revisi. Pada proses input data secara manual masih dimungkinkan adanya kesalahan format data atau format teks yang salah pada suatu data yang tidak terdeteksi, meskipun sudah beberapa kali diperiksa secara manual [5].

Menurut penelitian Sopiana Nainggolan [6], terkadang secara tidak sengaja seseorang menyimpan data yang sama didalam sebuah media penyimpanan. Tentunya ini sangat tidak efektif karena menyimpan data yang sama lebih dari satu. Tantangan yang dihadapi ketika seseorang ingin manajemen file dalam ruang penyimpanan tersebut adalah membedakan file yang satu dengan file yang lain. Sudah tentu cara yang dilakukan adalah melihat isi file tersebut satu persatu.

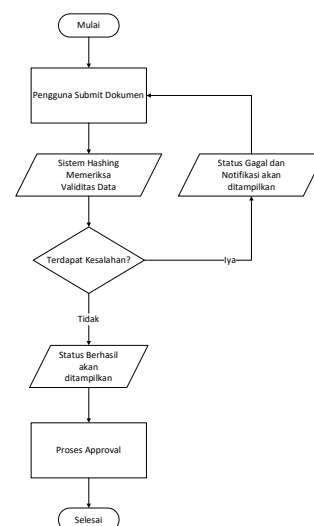
Pada penelitian Nova Ariska [7], Dalam fungsi *hash* terdapat metode SHA-256 ataupun *Secure Hash Algorithm* 256 yang merupakan salah satu algoritma *hashing* yang sering digunakan untuk mengenkripsi data dengan panjang 256 bit. Dengan menerapkan algoritma ini maka hasil dari pencarian file gambar tersebut di kelompokkan berdasarkan nilai *hash* yang sama dan pengguna dapat memudahkan dan mempercepat pengguna untuk menghapus file gambar yang duplikat tanpa harus membuka dan melihat isi file tersebut.

Oleh karena itu berdasarkan penjelasan diatas, sistem deteksi otomatis sangat penting untuk mengefisiensi pekerjaan tim validasi melakukan *approval* data. Dengan adanya sistem deteksi otomatis, hal ini dapat menjadi sebuah terobosan yang mutakhir untuk meningkatkan efisiensi dalam proses pengecekan data yang nantinya mempercepat proses pelaporan ke *customer* dan proses komunikasi kepada tim lapangan secara otomatis. Pembuatan deteksi *evidence* pada proyek *manage service* Fiber Optik dan notifikasi dengan pengimplementasian menggunakan *google appsheet* memberikan solusi secara menyeluruh karena aplikasi AONE pada dasarnya menggunakan *google appsheet* yang dari segi

pemeliharaan aplikasi jauh lebih terjangkau dan *user friendly*.

2. Eksperimental

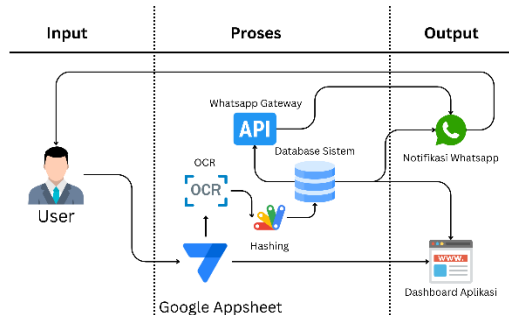
Berdasarkan diagram alir (*flowchart*) yang diperlihatkan pada Gambar 1, Sistem ini memanfaatkan fitur *automation* di *apsheet* untuk mengintegrasikan alur kerja berbasis deteksi foto, dimulai dari *preprocessing* menggunakan OCR karena menghasilkan akurasi yang lebih baik bahkan dalam situasi-situasi yang dimana karakter sulit untuk dikenali [8], serta menghasilkan kode *hash* unik 64 karakter menggunakan algoritma SHA-256. SHA-256 dipilih karena data yang disimpan memiliki kemungkinan kecil untuk didekripsi jika data itu bocor [9]. *Hashing* yang dihasilkan dibandingkan dengan data dalam *database* untuk mendeteksi duplikasi atau kesalahan, dan jika terdeteksi, sistem secara otomatis mengirimkan notifikasi berisi detail kesalahan, saran perbaikan, dan langkah tindak lanjut melalui *WhatsApp Gateway* dan email, serta menampilkannya di *database frontend*. Sistem juga memantau kondisi *backend database*, jika jaringan normal, respon disimpan dan ditampilkan di *dashboard* aplikasi, namun jika terganggu, parameter kondisi tetap disimpan dan diinformasikan melalui notifikasi, memastikan akurasi data, mengurangi kesalahan manual, dan meningkatkan efisiensi operasional aplikasi AONE.



Gambar 1. Flowchart Cara Kerja Sistem Deteksi Otomatis dan Notifikasi

Sistem deteksi otomatis dan notifikasi yang dirancang bertujuan untuk memudahkan tim validasi data untuk melakukan automasi konfigurasi dan *monitoring* jumlah data yang disubmit dalam bentuk antar muka aplikasi sehingga lebih ramah pengguna (*user-friendly*). Pada dasarnya, untuk dapat melakukan pengecekan suatu foto yang diunggah, tim validasi data melakukan pengecekan secara manual untuk memastikan foto yang diunggah

sesuai standar perusahaan sehingga tidak ada manipulasi. yang menyebabkannya lebih lama proses penagihan ke *customer*. Begitu pula dengan proses notifikasi, seorang admin secara otomatis memberi pesan kepada *user* (tim lapangan) melalui bot *Whatsapp Gateway* saat mengunggah *evidence* yang salah untuk memberitahu foto tersebut harus diunggah ulang.

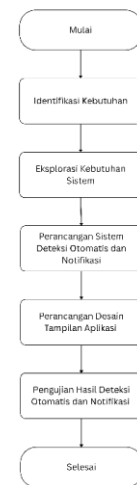


Gambar 2. Visualisasi Blok Sistem Deteksi Otomatis dan Notifikasi

Gambar 2 menampilkan diagram blok yang menunjukkan komponen-komponen yang membangun sistem ini. Komponen yang diperlukan antara lain *Google Appsheet* sebagai platform aplikasi antarmuka, Metode OCR (*Optical Character Recognition*) sebagai deteksi karakter teks pada foto. Fungsi *Hashing* menggunakan algoritma SHA-256 sebagai filtrasi untuk deteksi foto yang sama. *database* untuk menyimpan keadaan (status) yang telah di *approv* atau di *reject*, *Whatsapp Gateway* sebagai media notifikasi apabila terdeteksi foto ganda atau di manipulasi, dan *Dashboard Aplikasi* sebagai antarmuka antara pengguna dengan sistem.

Penelitian ini terdiri dari 5 (lima) tahap. Berdasarkan Gambar 3, langkah awal dalam proses penelitian ini antara lain:

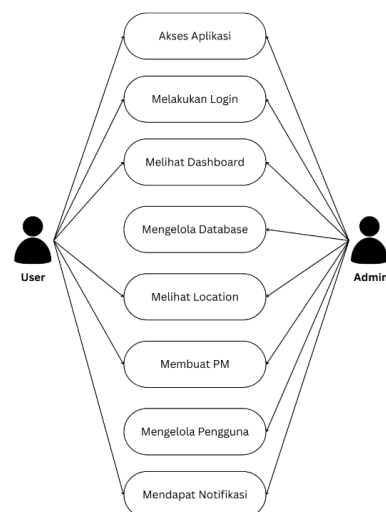
- Identifikasi kebutuhan dengan KOPINDOSAT: Melibatkan penetapan persyaratan sistem dan kebutuhan *use case* sistem diperlukan.
- Eksplorasi kebutuhan sistem: Menentukan algoritma SHA-256 dengan *Google Appsheet*, dan API *Fonnte* untuk *Whatsapp Gateway* agar saling berkomunikasi.



Gambar 3. Diagram Alir Perancangan Sistem Deteksi Otomatis dan Notifikasi

c. Perancangan fungsi *Hashing* untuk automasi deteksi foto dan Notifikasi: fungsi *hashing* akan diterapkan menggunakan algoritma SHA-256 karena digunakan untuk keperluan kriptografi [10]. Fungsi ini akan berisi beberapa baris kode program *Javascript* yang akan dieksekusi pada *Appscript* untuk melakukan automasi ke aplikasi *Appsheet*. Untuk notifikasi, salah satu fitur yang ada di *Appsheet* yaitu bots akan disinkronisasikan dengan API *Fonnte* sebagai *Whatsapp Gateway*.

d. Perancangan desain tampilan aplikasi sesuai kebutuhan KOPINDOSAT: Kebutuhan KOPINDOSAT akan meliputi jumlah halaman, konten tiap halaman, penempatan (*layout*) elemen, dan fungsionalitas fitur-fitur pada aplikasi. Gambar 4 menampilkan *use case diagram* yang menunjukkan interaksi sistem dengan pengguna.



Gambar 4. Use Case Diagram Aplikasi Deteksi Otomatis dan Notifikasi

e. Pengujian deteksi otomatis dan notifikasi: pengujian dilakukan secara menyeluruh untuk mengetahui program dari SHA-256 yang di implementasikan pada *Appscript*, program OCR pada *Appsheets*, dan program *Whatsapp Gateway* pada *Appsheets* berjalan dengan semestinya.

3. Hasil dan Pembahasan

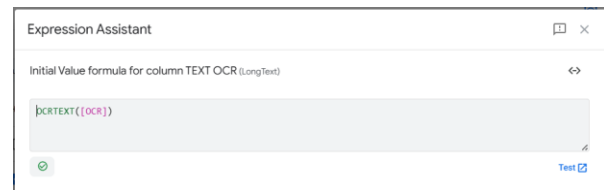
Pada tahap awal penelitian, dilakukan identifikasi kebutuhan melalui diskusi dengan KOPINDOSAT. Diperoleh bahwa sistem automasi dan monitoring diperlukan untuk mempermudah tim validasi dalam menyeleksi foto melalui antarmuka grafis (GUI). Sebelumnya, proses ini dilakukan secara manual dan kurang ramah pengguna. Oleh karena itu, dirancang sistem deteksi foto otomatis dan notifikasi yang terintegrasi dengan aplikasi agar lebih efisien dan *user-friendly*.

Adapun kebutuhan sistem untuk melakukan automasi dan notifikasi ini antara lain *Appsheets* versi *Core* dipilih karena versi tersebut memiliki fungsi yang lebih banyak dan bisa menangani tugas yang lebih kompleks [11], *Hashing* versi SHA-256 dipilih karena algoritma SHA-256 akan mengenkripsi informasi catatan penting terkait file tersebut agar tidak dapat diubah [12], serta pengidentifikasi karakter secara spesifik sebagai algoritma penyeleksian foto, *Optical Character Recognition* (OCR) dipilih sebagai pengidentifikasi tulisan yang ada di foto yang nantinya di konfersi dalam bentuk file [13], *database* spreadsheet sebagai penyimpanan data secara otomatis dalam jumlah banyak, dan *Whatsapp Gateway* menggunakan API fonnte dipilih karena dapat mengirim pesan secara otomatis tanpa perlu pengelolaan server atau infrastruktur tambahan dari sisi pengembangan aplikasi [14].

Pada tahap ketiga penelitian, berdasarkan hasil diskusi tersebut didapatkan 2 (dua) metode untuk menyeleksi foto yang relevan untuk foto yang diunggah di aplikasi. Untuk metode yang pertama yaitu OCR untuk mengidentifikasi foto dan mengubahnya menjadi teks. OCR ini merupakan salah satu fitur yang terdapat pada *Appsheets* sehingga hanya perlu melakukan penulisan kode pada saat input foto.

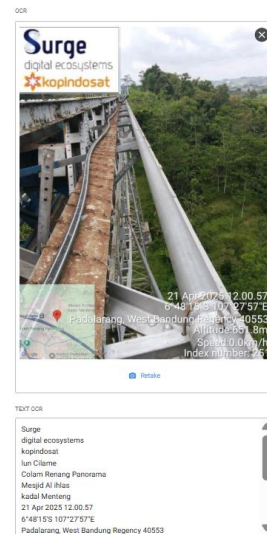
Hasil perancangan OCR yaitu dalam bentuk *Initial Value Formula* pada *Appsheets* berisi kode untuk menjalankan kode OCR secara otomatis pada *appsheets*. Di bawah ini merupakan contoh kode formula untuk OCR.

Kode:
OCRTEXT([OCR])



Gambar 5. Formula OCR *Appsheets*

Pada hasil dari kode OCR ditampilkan pada Gambar 6, dijelaskan bahwa kode yang diterapkan berjalan sesuai dengan foto yang diunggah. OCR tersebut mendeteksi teks yang ada di foto sehingga informasi yang dihasilkan sesuai dengan gambar yang diunggah.



Gambar 6. Implementasi OCR pada Aplikasi

Pada OCR ini ditambahkan kode untuk mengambil kata yang lebih spesifik pada hasil gambar yang sudah terdeteksi menjadi teks. Kata yang di ambil untuk mendeteksi identitas foto yaitu *index number* karena setiap foto memiliki *index number* yang berbeda. Untuk kode tersebut menggunakan pemrograman *Appsheets*. Untuk hasil implementasi diperlihatkan di Gambar 7.

Kode:

```
IF(
  FIND("index number", LOWER([FileName]))
  > 0,
  NUMBER(
    TRIM(
      MID(
        LOWER([FileName]),
        FIND("index number",
          LOWER([FileName])) + 13,
          5
        )
      )
    ),
  ""
)
```

TEXT OCR

6°48'15"S 107°27'57"E
 Padalarang, West Bandung Regency 40553
 Altitude:651.8m
 Speed:0.0km/h
 h
 Google
 B
 Komisi Pemilihan
 Umum Kabunaton
 Index number: 251

Index

251

Gambar 7. Implementasi Identifikasi Index Number

Setelah metode OCR diterapkan Langkah selanjutnya yaitu dengan menerapkan metode *hashing* menggunakan versi SHA-256. Metode ini berbeda penerapannya dengan OCR karena SHA-256 di implementasikannya dalam bentuk Bahasa pemrograman *Javascript* sehingga membutuhkan pihak ketiga untuk menjalankannya yaitu *Appscript* [15]. Hasil perancangan kode SHA-256 yaitu dalam bentuk kode di *Appscript*. Program SHA-256 berfungsi untuk mengubah secara otomatis foto yang diunggah menjadi kode unik. Di bawah ini merupakan contoh kode program untuk kode SHA-256.

Generate Hash SHA-256

```
var digest =
Utilities.computeDigest(Utilities.DigestAlgorithms.SHA_256, filename);
var hash = digest.map(function(byte) {
  var hex = (byte < 0 ? byte + 256 :
byte).toString(16);
  return (hex.length == 1 ? "0" + hex :
hex);
}).join(''); ());
```

Generate Hash SHA-256 menunjukkan blok kode program yang menggunakan algoritma SHA-256 untuk membuat hash dari string filename. Fungsinya agar *computeDigest* mengembalikan array byte hasil hash. Selanjutnya fungsi selanjutnya yaitu untuk mengubah setiap byte menjadi *string hexadecimal* (base 16) Menambahkan nol di depan jika panjangnya hanya 1 digit, untuk menjaga panjang konsisten (64 karakter untuk SHA-256). Hasil akhirnya adalah string hash SHA-256 dalam format hex sepanjang 64 karakter [16]. Gambar 7 menampilkan hasil dari pengkodean gambar menggunakan algoritma SHA-256.

Upload Foto

Deskripsi Temuan

Tanggal Upload

Hash Gambar

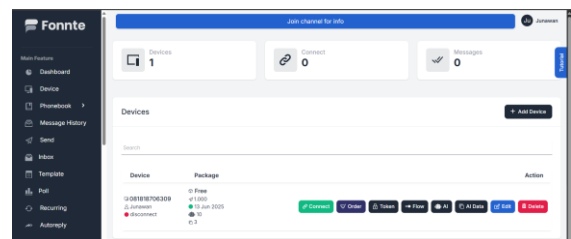
hahha

5/7/2025 9:00:09 AM

1103dce9c508ab3628681c86a74a9ff1f5ed0338b167d9cf02b502c94b49fba3

Gambar 8. Implementasi SHA-256

Pada salah satu fitur appsheet yaitu *bots* yang salah satu fungsinya yaitu memberikan notifikasi kepada user yang bisa diatur secara *realtime*. Untuk memanfaatkan bots itu sendiri bisa menggunakan pihak ketiga yaitu API *Whatsapp Gateway* yaitu *fonnte.com*.



Gambar 9. Dashboard API Fonnte

Pada Gambar 9 menjelaskan ketika sudah login menggunakan nomor *whatsapp* setelah itu membuat *device* baru untuk mengambil token API untuk di salin di *bots appsheet*.

Settings

Send an email Send a notification Send an SMS HTTP Call a webhook

Create a new file Call a script

Table: Data PM

Preset

Uri

x2aYicco71xQVdQ1LCSZ

Gambar 10. Token API Fonnte

Setelah token API Fonnte di salin di *bots appsheet* maka token tersebut akan mengidentifikasi identitas nomor whatsapp menjadi gateway dan memberikan notifikasi secara otomatis ketika foto yang terdeteksi ganda.

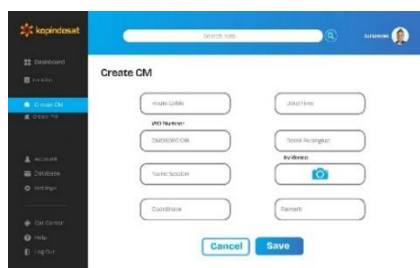
Pada tahap keempat penelitian, dilakukan perancangan antarmuka aplikasi yang mencakup identifikasi kebutuhan, penyusunan diagram alur, pembuatan wireframe dengan *Canva*, serta evaluasi desain. Proses dimulai dengan diskusi bersama KOPINDOSAT untuk menentukan tujuan aplikasi, diikuti perencanaan struktur, penempatan elemen tampilan, dan pencocokan desain dengan kebutuhan pengguna.

Gambar 11 menampilkan wireframe halaman dashboard aplikasi. Antarmuka ini mencakup menu navigasi vertikal di sisi kiri dengan ikon dan label fitur seperti *Dashboard*, *Location*, *Create CM/PM*, *Account*, *Database*, *Settings*, dan *Log Out*. Bagian utama terbagi dua, dengan panel kiri menampilkan jumlah data *Preventive Maintenance (PM)*, *Waiting*, dan *Approval* yang diinput *user*, serta kotak "Info Account" berisi nama, jabatan, dan wilayah kerja pengguna.



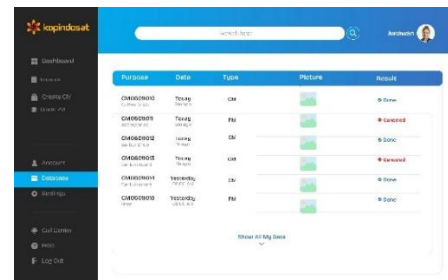
Gambar 11. Wireframe halaman utama

Wireframe halaman berikutnya pada Gambar 12 memiliki komponen untuk input data yang terdiri dari *WO Number*, identitas, Lokasi, sampai foto *evidence* untuk di deteksi.



Gambar 12. Wireframe halaman Create CM

Pada wireframe halaman *Database* Aplikasi di Gambar 13, akan ditampilkan hasil input data yang dimana terdapat *WO Number* sebagai identifikasi dari sebuah tiket yang memiliki tanggal unggah, nama dari data yang di input, foto *evidence*, yang terakhir yaitu hasil dari tiket tersebut di-approve atau di-reject.



Gambar 13. Wireframe halaman Database Aplikasi

Pada wireframe terakhir di Gambar 14 ini akan menampilkan halaman location yang didekskripsikan dengan tampilan *maps*. Halaman ini mengambil Lokasi dari proses input pada *Create CM* sehingga tampilan Lokasi tersebut titik lokasinya sesuai dengan gambar yang memiliki *longitude* dan *latitude* yang sesuai. Guna dari halaman itu berfungsi sebagai tracking tim lapangan yang sudah patroli. Pada titik Lokasi tersebut jika dilihat lebih lanjut berisikan foto dan dekskripsi dari foto tersebut berdasarkan tiket yang di input.



Gambar 14. Wireframe halaman Location

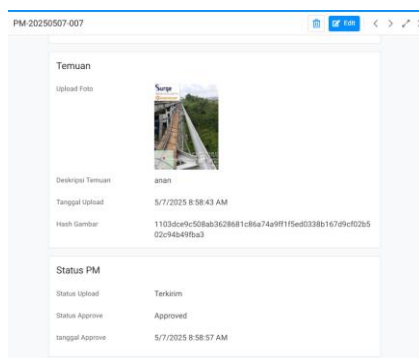
Pada tahap kelima, dilakukan proses pengujian sub sistem pada sistem deteksi foto dan notifikasi yang sudah dirancang. Pada pengujian kali ini dilakukan pengujian sebanyak dua kali unggah foto yang sama apakah foto tersebut menghasilkan kode *Hashing* yang sama sehingga terdeteksi duplikat. Terdapat Langkah-langkah untuk menganalisa hasil pengujian dari pengujian sub sistem dengan menguji setiap program yang telah dibuat, hasil pengujian disajikan pada Tabel 1, dan Gambar 16-13 merupakan output dari pengujian.

Pada tahap awal pengujian, dilakukan unggah foto pada aplikasi *Appsheets* dengan mengunggah foto apakah algoritma *SHA-256* berjalan. Foto yang digunakan terlampir pada Gambar 15. Foto tersebut merupakan standarisasi dari KOPINDOSAT foto tersebut diambil menggunakan *timestamp*.



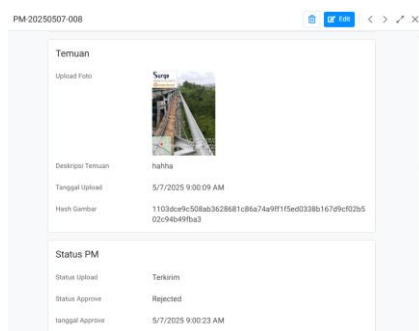
Gambar 15. Foto Yang Ingin diunggah

Pada Gambar 16 menjelaskan foto yang berhasil di unggah dengan urutan PM nomor PM-20250507-007 dan menampilkan kode *Hashing* sepanjang 64 karakter yang menjadi identitas setiap foto. Membuktikan bahwa proses algoritma SHA-256 sudah berjalan. Dengan kode *Hashing* yang ditampilkan nilainya yaitu: 1103dce9c508ab3628681c86a74a9ff1f5ed0338b167d9cf02b502c94b49fba3



Gambar 16. Hasil Unggah Foto

Pada Gambar 17 menjelaskan proses unggah foto yang sama dengan urutan PM nomor PM-20250507-008, terlihat *Hashing* yang dihasilkan itu sama persis seperti foto sebelumnya yang diunggah. Dengan membandingkan foto yang di upload pada PM-20250507-007 dan PM-20250507-008 menghasilkan nilai *hashing* ganda yaitu: 1103dce9c508ab3628681c86a74a9ff1f5ed0338b167d9cf02b502c94b49fba3



Gambar 17. Hasil Unggah Foto yang Sama

Setelah berhasil melakukan pengujian algoritma *Hashing* SHA-256. Langkah selanjutnya menguji apakah di foto yang diunggah teks tersebut terbaca sesuai menggunakan OCR. Pengecekan tersebut dapat dilihat juga pada *database spreadsheet*. Gambar 19 menjelaskan hasil dari Program OCR yang berhasil membaca teks yang ada di foto. Dengan membaca keseluruhan teks yang ada di foto pada Gambar 16.

```
Surge
digital ecosystems
kopindosat
Iun Cilame
Colam Renang Panorama
Mesjid Al Ihliah
kadal Menteng
21 Apr 2025 12:00:57
6°48'15"S 107°27'57"E
Padalarang, West Bandung Regency 40553
Altitude: 651.8m
Speed: 0.0km/h
h
Google
Komisi Pemilihan
Cabunata
Index number: 251
```

Gambar 18. Hasil Program OCR

Langkah berikutnya menguji proses notifikasi berjalan ketika pada saat unggah foto yang sama secara otomatis melakukan notifikasi kepada *user* yang mengunggah. Pada Gambar 20 menjelaskan hasil notifikasi yang disampaikan terkirim secara *real-time* dengan durasi pengiriman notifikasi kurang dari lima detik sesuai nomor PM yaitu nomor PM-20250507-008 sama dengan nomor PM PM-20250507-007.



Gambar 19. Hasil Notifikasi Whatsapp Gateway

Tabel 1. Hasil Pengujian Sub Sistem

	Test Case	Hasil
1	Program <i>Hashing</i> SHA-256	100%
2	Program OCR	100%
3	Program <i>Whatsapp Gateway</i>	100%

Berdasarkan pengujian Sub sistem, didapatkan hasil tiga *test case* yang berhasil dan 0 *test case* yang tidak berhasil. yang tidak berhasil, dan presentase kelayakan dari pengujian ini dapat dihitung dengan menggunakan persamaan 1 yaitu sebagai berikut.

$$\text{Persentase Kelayakan} = \frac{\text{Jumlah Yang Berhasil}}{\text{Jumlah Pengujian}} \times 100\%$$

$$= \frac{3}{3} \times 100\% = 100\%$$

Menurut hasil perhitungan presentase kelayakan diatas pengujian keseluruhan sistem mendapatkan hasil 100%, maka dapat dikatakan program yang telah dibuat, yaitu Program *Hashing* SHA-256, program OCR, Program *Whatsapp Gateway* berhasil dan dapat berjalan sesuai yang telah direncanakan.

4. Kesimpulan

Berdasarkan hasil perancangan yang telah direncanakan di atas, terdapat beberapa kesimpulan sebagai berikut:

- a. Untuk menjaga integritas data, diterapkan metode *hashing* versi SHA-256 yang sudah di uji berhasil menerjemahkan kode unik 64 karakter.
- b. Untuk membantu menjaga foto tetap terjaga keasliannya, dibantu dengan OCR *Optical Character Recognition* yang telah di uji berhasil membaca teks yang ada pada foto yang diunggah
- c. Aplikasi dilengkapi *WhatsApp Gateway* untuk mengirimkan notifikasi instan kepada *user* dan *admin*, yang telah diuji berhasil mengirimkan pesan notifikasi ketika unggah foto ganda.

Daftar Acuan

- [1] W. Fajar Sari Kurniawan, "Rancang Bangun Presensi Pegawai Dan Pelaporan Pekerjaan Secara Online Menggunakan Aplikasi Appsheet," *Journal Of Systems, Information Technology, And Electronics Engineering*, Pp. 40-49, 2021.
- [2] D. Muhammad Al-Fajri, "Implementasi Aplikasi Appsheet Berbasis Android Untuk Mendukung Proses Pembelajaran," *Jmik (Jurnal Mahasiswa Ilmu Komputer)*, Vol. 4, No. 2, Pp. 159-167, Oktober 2023.
- [3] Kopindosat.Co.Id, "Jasa Teknik," Jasa Telekomunikasi, 2025. [Online]. Available: <https://Kopindosat.Co.Id/Bisnis/Jastek>.
- [4] D. Silvi Sitaviana, "Penggunaan Website Appsheet Sebagai Temu Balik Arsip Inaktif Foto," *Information Science And Library*, Vol. 4, No. 1, Pp. 40-46, Juni 2023.
- [5] D. Avril Virga Mutiara Yasmin, "Perancangan Aplikasi Inventory Management Menggunakan Google Appsheet Pada Laboratorium Pt Energi Agro Nusantara," *Jurnal Teknologi Dan Informasi (Jati)*, Vol. 14, No. 2, Pp. 126-137, September 2024.
- [6] S. Nainggolan, "Implementasi Algoritma Sha-256 Pada Aplikasi Duplicate Document Scanner," *Resolusi : Rekayasa Teknik Informatika Dan Informasi*, Vol. 2, Pp. 201-213, Mei 2022.
- [7] N. Ariska, "Implementasi Fungsi Hash Dengan Algoritma Sha-256 Pada Aplikasi Duplicate Image Scanner," *Jurnal Sains Dan Teknologi Informasi*, Vol. 1, Pp. 112-120, September 2022.
- [8] D. Ali Firdaus, "Implementasi Optical Character Recognition(Ocr) Pada Masa Pandemi Covid-19," *Jurnal Jupiter*, Vol. 13, No. 2, Pp. 188 - 194, Oktober 2021.
- [9] D. Izhar Rahim, "Komparasi Fungsi Hash Md5 Dan Sha256 Dalam Keamanan Gambar Dan Teks," *Jurnal Ikraith-Informatika*, Vol. Vol 7, No. 2, Pp. 41-48, Juli 2023.
- [10] P. H. U. Ilyas Mahfud, "Implementasi Sistem Kriptografi Rsa Signature Dengan Sha-256 Pada Mekanisme Autentikasi Rest Api," *Seminar Nasional Teknoka*, Vol. 6, No. 1, Pp. 84-92, 2021.
- [11] F. S. Kurniawan, "Rancang Bangun Presensi Pegawai Dan Pelaporan Pekerjaan Secara Online Menggunakan Aplikasi Appsheet," *Journal Of Systems, Information Technology, And Electronics Engineering*, Pp. 40-49, 2022.
- [12] D. Herman, "Implementasi Algoritma Aes-128 Dan Sha-256 Dalam Perancangan Aplikasi Pengamanan File Dokumen," *Urnal Times*, Vol. 10, No. 2, P. 80 – 87, Desember 2021.
- [13] A. S. D. Suryo Hartanto, "Optical Character Recognition Menggunakan Algoritma Template Matching Correlation," *Jurnal Masyarakat Informatika*, Vol. 5, No. 9, Pp. 1-12, 2022.
- [14] E. S. Helmi, "Penerapan Metode Time Series Analysis Pada Sistem Informasi Menggunakan Whatsapp Gateway," *Journal Of Information System Research (Josh)*, Vol. 6, No. 1, P. 607–618, Oktober 2024.
- [15] G. S. Raihanah Luthfiyah Rosanti, "Implementasi Google App Script Untuk Input Data Pada Database Master Data," *Riset Dan E-Jurnal Manajemen Informatika Komputer*, Vol. 8, No. 1, Pp. 117-129, Januari 2024.
- [16] A. Fathurrozi, "Penerapan Algoritma Advanced Encryption Standard (Aes-256) Dengan Mode Cbc Dan Secure Hash Algorithm (Sha-256) Untuk Pengamanan Data File," *Journal Of Information And Information Security (Jiforty)*, Vol. 2, No. 2, Pp. 227-238, Desember 2021.