

Implementasi Google Drive API untuk Upload, Sharing, dan Download Data pada Aplikasi Berbasis Web

Pande Made Yudi Pratistha^{a1}, I Made Sukarsa^{a2}, Putu Wira Buana^{b3}

^aProgram Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana, Bukit Jimbaran, Bali, Indonesia

e-mail: ¹yudipratistha@gmail.com, ²sukarsa@unud.ac.id, ³wbhuana@unud.ac.id

Abstrak

Salah satu tantangan dalam pembuatan sistem informasi adalah pemilihan tempat penyimpanan data yang sesuai dengan kemampuan organisasi. Google Drive merupakan cloud storage yang dapat dimanfaatkan sebagai basis data untuk sebuah sistem informasi. Salah satu kelebihan menggunakan cloud storage google drive adalah tidak menggunakan alamat internet protocol sehingga proses pengoperasiannya bisa dilakukan diberbagai tempat dan hanya cukup bermodalkan akses jaringan internet. Hal ini dapat menjadi solusi yang cukup murah bagi sebuah organisasi yang membutuhkan pertukaran data. Tujuan dari penelitian ini untuk membuat aplikasi berbasis web yang memanfaatkan teknologi API dan penyimpanan data dari Google Drive. Sistem aplikasi ini berbasis web yang dapat membantu proses pertukaran data dan dapat menjadi alternatif penyimpanan data gratis. Aplikasi ini memanfaatkan cloud storage google drive sebagai tempat pertukaran data dan penyimpanan data. Aplikasi ini membutuhkan sebuah akun google untuk melakukan proses autentikasi dan sebagai perantara tempat pertukaran data sebelum ditampilkan di halaman view data.

Kata kunci: Aplikasi Web, API, Cloud Storage, Google Drive, Google Drive API

Abstract

One of the challenges in making information systems is the selection of data storage areas that are in accordance with the capabilities of the organization. Google Drive is a cloud storage that can be used as a database for an information system. One of the advantages of using Google Drive cloud storage is that it does not use an address internet protocol so that the operation process can be carried out in various places and only with internet network access. This can be a fairly inexpensive solution for an organization that requires data exchange. The purpose of this research is to create a web-based application that utilizes API technology and data storage from Google Drive. This application system is web-based which can assist the process of exchanging data and can be an alternative for free data storage. This application utilizes Google Drive cloud storage as a place for data exchange and data storage. This application requires a google account to perform the authentication process and as an intermediary for exchanging data before being displayed on the data view page.

Keywords: API, Cloud Storage, Google Drive, Google Drive API, Web Application

1. Introduction

Seiring dengan semakin pesatnya perkembangan teknologi di era ini, membuat para pelaku bisnis semakin banyak memanfaatkan teknologi dalam proses bisnisnya. Semakin bertumbuhnya suatu perusahaan, maka tentu akan membuat semakin banyak fitur atau fungsi untuk pengolahan dan penyimpanan catatan-catatan transaksi. Menghadapi tantangan tersebut, banyak organisasi yang melakukan pertukaran data (EDI) dengan mitra atau partner mereka. Pertukaran data elektronik merupakan proses mentransfer data yang terstruktur, memiliki format standar yang telah disepakati, dari satu sistem komputer ke sistem komputer lainnya, dalam berbentuk data elektronik [9].

Salah satu solusi untuk alternatif untuk tetap dapat melakukan pertukaran data, dengan memanfaatkan teknologi cloud storage yang dimiliki Google yaitu Google Drive. Pemanfaatan

layanan cloud storage sebagai pertukaran data elektronik, memiliki potensi tingkat efisiensi dalam mengelola data dan dapat menekan biaya agar lebih murah untuk penyimpanan data [5]. Google Drive adalah sebuah aplikasi penyimpanan *cloud* yang dimiliki oleh Google. *Cloud storage* merupakan sebuah metode penyimpanan yang datanya disimpan pada sejumlah server dan dikelola oleh penyedia layanan, pengaksesan sebuah cloud storage hanya membutuhkan koneksi internet untuk melakukan mengelola data [4]. Selain menyediakan cloud storage, Google Drive menyediakan API serta dokumentasi yang menjadi platform dalam pembuatan suatu aplikasi web. Fitur-fitur API yang ditawarkan oleh Google Drive meliputi unggah, unduh, berbagi data dan lain-lain. Untuk mengakses fitur API, dapat memanfaatkan akun Google untuk mendapatkan token otentikasi yang digunakan untuk menjalankan perintah-perintah Application Programming Interface (API). Dengan adanya fitur tersebut dapat memudahkan developer untuk melakukan proses pengelolaan data melalui aplikasi web [3].

Proses mengakses Data menggunakan Google Drive API telah diteliti sebelumnya dengan judul "Data Exchange Service using Google Drive API" [1]. Penelitian ini membuat sebuah pertukaran data pesan, yang memanfaatkan Google Drive API dan database. Penelitian ini memanfaatkan akun Google untuk mendapatkan token otentikasi yang digunakan untuk mengakses fungsi yang ada di API milik Google. Ketika pengguna mematikan fitur penjadwalan pesan otomatis, layanan penjadwalan pesan tidak dapat digunakan. Media penyimpanan dan *task scheduler* dibutuhkan oleh fitur penjadwalan pesan, *task scheduler* berfungsi untuk mengambil data secara berkala tiap menit jika *engine* mendapatkan akses ke internet.

Keuntungan pemanfaatan Google Drive sebagai tempat penyimpanan telah diteliti sebelumnya dengan topik "Web-Based College Student Assignment File Collection Application Using Google Drive API" [10]. Penelitian ini membahas bagaimana cara Google Drive menjadi solusi sebagai penyimpanan data alternatif untuk pengumpulan *file* tugas siswa. Dengan adanya Google Drive sebagai alternatif penyimpanan data, dapat teratasinya masalah keterbatasan kapasitas penyimpanan yang disediakan oleh hosting berbayar atau gratis.

Berdasarkan pemaparan diatas, dibuatlah penelitian dengan judul "Implementation of Google Drive API for Data Upload, Sharing, and Download in Web-Based Application". Penelitian ini membahas bagaimana memanfaatkan fitur-fitur API yang disediakan oleh Google Drive yang meliputi unggah, unduh dan berbagi. Dari fitur-fitur yang telah disediakan dapat memfasilitasi sebagai jalur akses pertukaran data baru atau sebagai alternatif untuk penyimpanan data sementara pada sistem. *Output* penelitian yang dibuat berupa suatu sistem aplikasi berbasis web yang memanfaatkan Google Drive API sebagai jalur pertukaran data.

2. Metodologi Penelitian

Penelitian yang dibuat memiliki metode dan metodologi penelitian yang dituang dalam bentuk pemikiran sehingga didokumentasikan sebagai berikut.

2.1. Perumusan Masalah

Sebelum dimulainya penelitian ini, tahapan awal adalah merumuskan masalah. Perumusan dilaksanakan dengan cara menentukan masalah serta solusi penyimpanan data alternatif pada suatu sistem informasi berbasis *web*.

2.2. Studi Literatur

Tahapan studi literatur dilakukan dengan mengumpulkan referensi mengenai teori dasar, dokumentasi teknologi yang digunakan, dan fitur dari tools yang digunakan. Topik penelitian ini menggunakan literatur yang terkait, terutama mengenai penyimpanan data. Selanjutnya studi literatur yang digunakan saat pembuatan sistem, mengenai pemanfaatan teknologi Application Programming Interface. Studi juga dilakukan dengan mempelajari dokumentasi dari Google Drive API serta fitur dan resource yang disediakan oleh Google Drive.

2.3. Perancangan Sistem

Tahapan rancangan untuk sistem merupakan tahapan yang dilakukan setelah melakukan studi literatur. Tahapan ini meliputi proses pembuatan gambaran umum sistem, Standard Operating Procedure (SOP).

2.4. Pembuatan Sistem

Proses atau tahap pembuatan dari sistem meliputi pembuatan tampilan menggunakan html, css dan javascript untuk *website* penginputan data serta *view data*. *Website* dibuat dengan *javascript* dan bahasa pemrograman php *native*. Implementasi Google Drive API menggunakan fitur *upload*, *download* dan *sharing* data menggunakan suatu SDK dari Google Drive

2.5. Pengujian Sistem

Pengujian system dilakukan dengan metode black box dengan menguji setiap fitur antara lain proses upload, download dan sharing data untuk mengetahui apakah 3 fitur tersebut dapat berjalan dengan baik.

3. Literature Study

Teori yang dipakai pada penelitian dengan judul Implementasi Google Drive API untuk Upload, Sharing, dan Download Data pada Aplikasi Berbasis Web akan dijelaskan pada bagian berikut.

3.1. Cloud Storage

Cloud Storage merupakan layanan penyimpanan data digital secara online yang memanfaatkan koneksi internet untuk melakukan akses data. *Cloud storage* menyimpan data pada sejumlah *server*, yang dikelola oleh penyedia layanan. Hal ini, dapat menjamin data yang disimpan cloud storage terbebas dari serangan malware atau data hilang yang disebabkan *hardware* rusak (HDD). Data pada *cloud storage* dapat diakses setiap saat, asalkan *device* mendapat koneksi internet. Layanan *cloud storage*, seperti Google Drive, OneDrive dan Dropbox dapat digunakan secara gratis yang kapasitas penyimpanan yang telah ditentukan oleh pihak penyedia layanan dan diberikan juga opsi untuk menambah kapasitas secara berbayar [6].

3.2. Google Drive

Google Drive adalah *storage* berbasis *cloud* yang dikembangkan Google untuk tempat *storage* yang dapat dintegrasikan ke layanan milik Google yang lain. 24 April 2012 merupakan hari peluncuran Google Drive ini dan pada saat tahun 2018 Google Drive menyentuh 1 milyar pengguna [2]. Ukuran ruang *storage* yang didapatkan di akun Google Drive yaitu sebesar 15 GB tetapi sudah termasuk Google Mail, Google Slides, Google Docs, Google Photos, serta Google Sheets [1]. Layanan Google Drive dapat diakses dan digunakan hanya dengan satu akun Google yang terdaftar, baik melalui platform web, Android maupun iOS [10].

3.3. Google Drive API V3

Google Drive API V3 merupakan kumpulan peraturan yang terdapat pada Google dan dapat digunakan untuk melakukan mengakses terhadap berbagai fitur yang terdapat pada Google Drive. *Application Programming Interface* (API) ini mendukung banyak bahasa pemrograman diantaranya PHP, Javascript, Java, .NET, Android, iOS, Node.js, Python, dan Ruby. Dimana pengembang diberikan kemudahan untuk membangun sebuah aplikasi berbasis web dengan cara *import SDK client libraries* yang telah disiapkan oleh Google. Google Drive SDK menyertakan API HTTP yang memungkinkan pengguna atau pengembang mengintegrasikan, memodifikasi, dan mengakses file yang disimpan di Google Drive pengguna dengan aplikasi pihak ketiga mereka sendiri [9].

Penggunaan Google Drive API memiliki limitasi ketika sebuah aplikasi melakukan proses *query* ke API. Tabel berikut merupakan batasan permintaan *query* kedalam Google Drive API.

Tabel 1 Kuota Permintaan *Query*

Permintaan per 100 detik	10.000 (sepuluh ribu)
Permintaan <i>Query</i> per hari	1.000.000.000 (satu miliar)
Permintaan <i>Query</i> per pengguna	10.000 (sepuluh ribu)

Tabel 1 merupakan kuota permintaan *query* ke API dari Google Drive. Kuota *request queries* dapat dilihat di halaman *website* console.developers.google.com yaitu dengan cara pilih *project*, dashboard, pilih Google Drive API dan pilih menu *quota*. Kuota pada *request query* dapat ditingkatkan sesuai keinginan. Peningkatan kuota juga dapat dipilih oleh pengguna dengan cara melengkapi *form* yang sudah ditetapkan oleh Google. Penambahan *query* tidak bisa dilakukan karena data yang harus dilengkapi masih kurang.

3.4. API (Application Programming Interface)

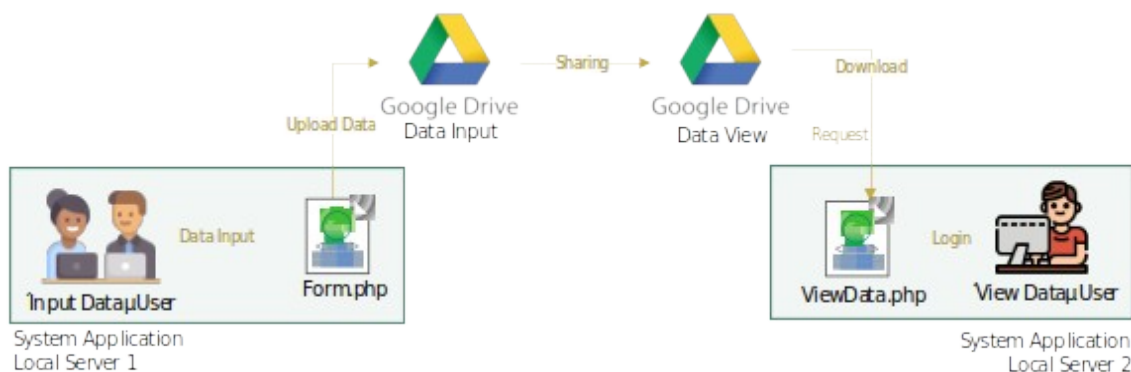
Application Programming Interface adalah suatu aturan yang ada di *software* yang berfungsi dalam hal mengakses atau menggunakan *service* selain itu *resource* yang dimiliki oleh *software* lain yang sudah diterapkan pada Application Programming Interface tersebut. Manfaat API atau Application Programming Interface adalah sebagai jembatan antara 2 *software* yang berlainan dan proses interaksi antar keduanya hampir sama dengan antara komputer dan manusia [7].

4. Result and Discussion

Hasil dari implementasi Google Drive API sebagai penyimpanan data, dapat dilihat pada gambar

4.1. System Architecture

Penelitian ini mengimplementasi Google Drive API untuk melakukan proses *upload*, *download* dan *sharing* data pada aplikasi *web based*. Gambaran umum penelitian akan dijabarkan pada gambar 1

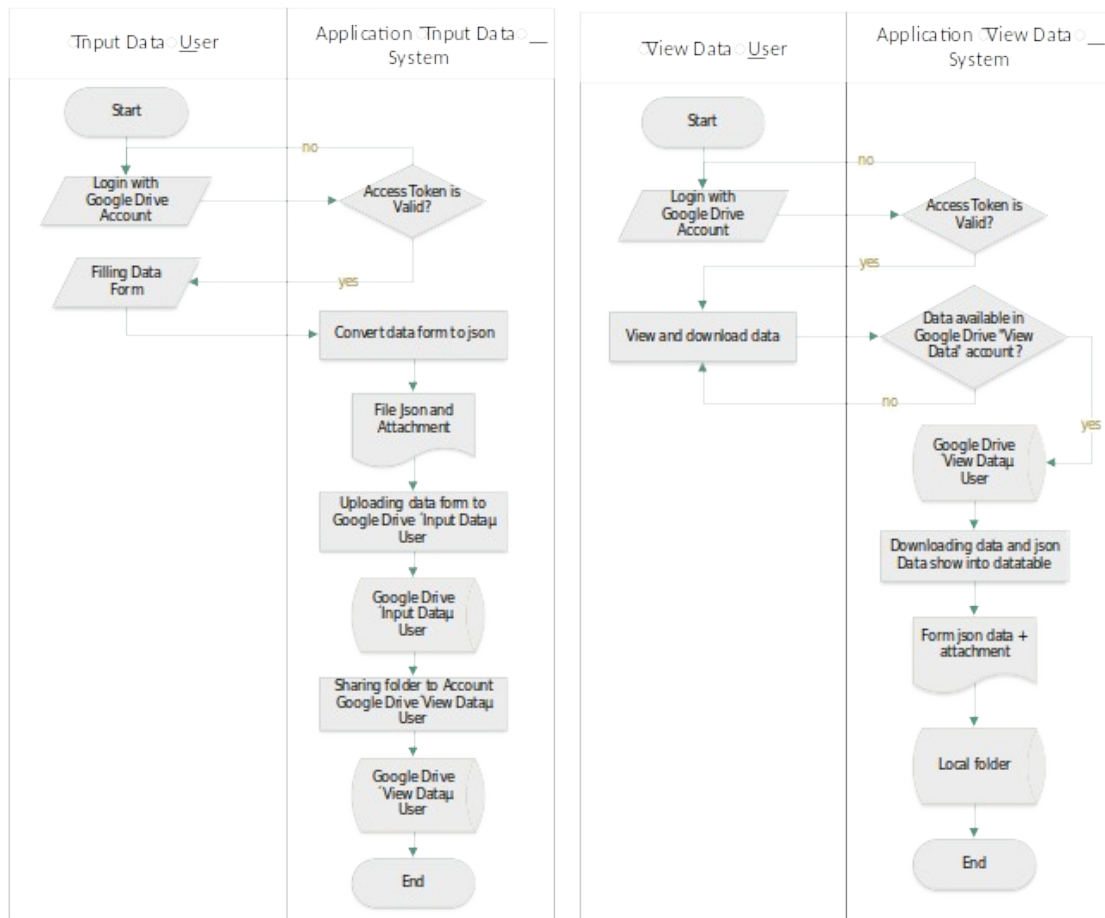


Gambar 1. Gambaran Umum Sistem

User penginput Data mengisi atau menginputkan data pada form di halaman form.php kemudian data tersebut di parsing menjadi format json. *File* json akan di upload ke google drive milik penginput data menggunakan API dari Google Drive. Setelah *file* json terunggah, maka *file* json tersebut kemudian akan di *sharing* menggunakan API ke google drive milik *user view data*. Pengguna "View Data" dapat mengakses halaman *view data* untuk melakukan *request* ke Google Drive miliknya untuk melakukan proses *download* data dari Google Drive agar dapat ditampilkan di halaman ViewData.php.

4.2. Standard Operating System

SOP yang dibuat memuat prosedur dari proses yang terjadi pada Sistem Informasi yang melibatkan 2 jenis user dan sistem aplikasi. SOP atau *Standard Operating System* ini akan dijelaskan pada gambar 2.



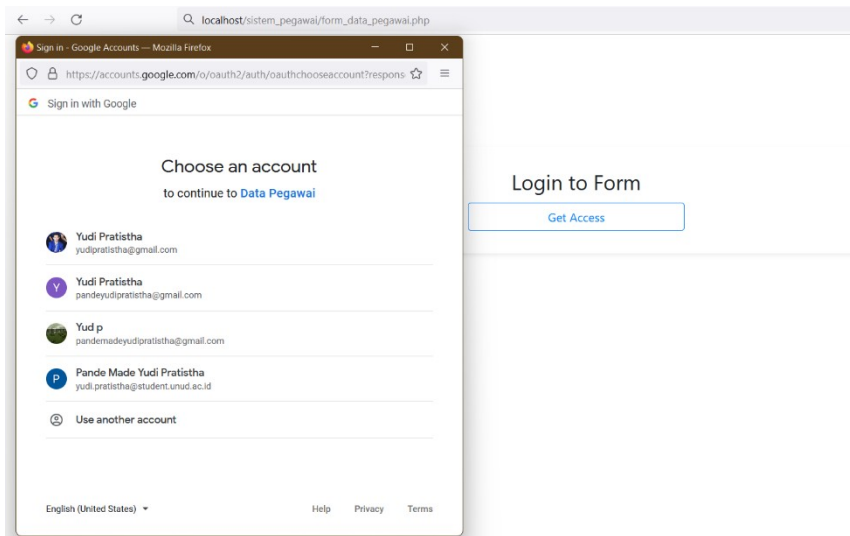
Gambar 2. SOP Pengisian Data Formulir Menggunakan Akun Google Drive Pengguna "Input Data" dan Melihat Data Menggunakan Akun Google Drive Pengguna "View Data"

Proses SOP Pengisian Data Form Menggunakan Akun Google Drive Pengguna "Input Data" dimulai dari pengguna "Input Data" melakukan proses login menggunakan akun google untuk sebagai syarat melakukan proses input data. Setelah login, pengguna "Input Data" dapat melakukan proses input data dan kemudian hasil input data dikonversikan menjadi format json lalu diunggah ke akun google drive pengguna "Input Data". Data pada google drive pengguna "Input Data" akan di share ke google drive pengguna "View Data".

Proses SOP Melihat Data Menggunakan Akun Google Drive Pengguna "View Data" dimulai dari pengguna "View Data" melakukan proses login menggunakan akun google untuk dapat melakukan proses download data. Ketika telah berhasil login, pengguna "Input Data" dapat mengakses halaman "View Data", lalu pengguna "View Data" melakukan proses mendownload data. Jika data baru tersedia pada akun Google Drive "View Data", system akan mendownload data (json dan attachment) dan membuat folder sebagai tempat penyimpanan data.

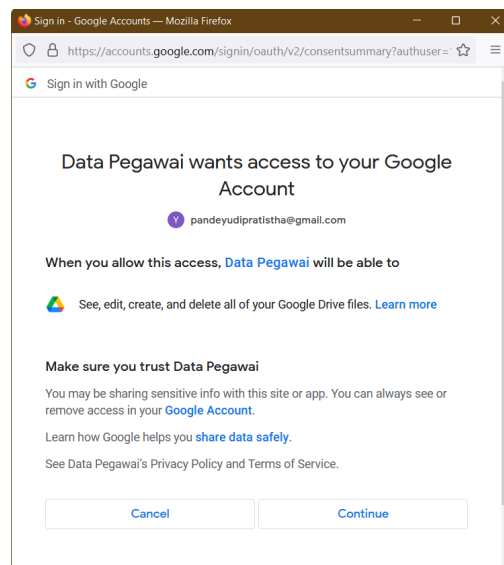
4.3. Pengujian Sistem

Tahap pengujian dilakukan dengan melakukan percobaan terhadap fitur-fitur dalam aplikasi. Percobaan dilakukan pada fitur-fitur yang tersedia pada sistem *website* dan meliputi fungsi-fungsi pendukung proses dari *upload*, *sharing* dan *download* dengan Google Drive.



Gambar 3. Pilih atau Login Menggunakan Akun Pengguna Google Drive "Input Data"

Sebelum memasuki halaman *form*, pengguna "Input Data" diminta untuk menekan tombol *get access* dan muncul popup OAuth consent screen. Pengguna "Input Data" akan ditampilkan akun Google yang telah tersedia pada *session browser* atau dapat melakukan proses menambahkan akun baru. Ketika akun google telah dipilih, pengguna "Input Data" diminta untuk memberikan akses kepada Google Drive seperti gambar 4



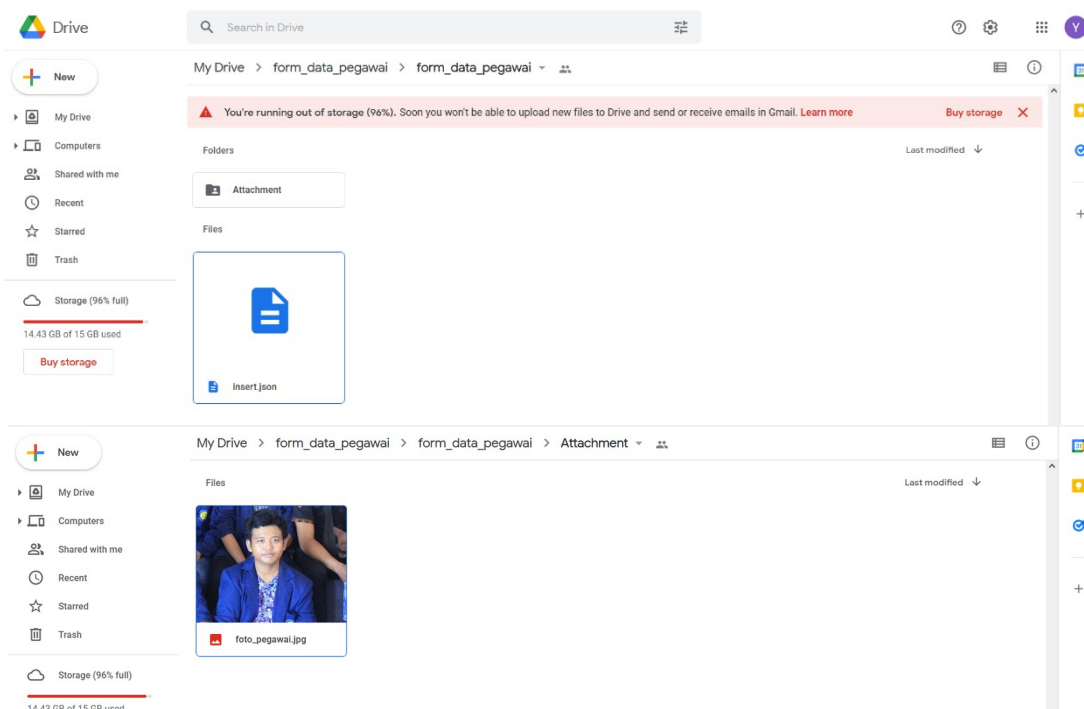
Gambar 4. Grant Access Akun Pengguna Google Drive "Input Data"

Ketika pengguna "Input Data" telah memilih akun Google, API Google Drive akan meminta izin untuk akses manajemen *file* (*upload*, *membuat folder* dan *sharing*), bahwa akun Google Drivenya akan digunakan. Setelah pengguna "Input Data" menekan tombol *continue*, maka Google Drive API akan memberikan *access token* dan sistem akan menyimpan *access token* pada *session browser*.

The screenshot shows a mobile application interface for 'SISTEM PEGAWAI'. On the left is a dark sidebar menu with the user's name 'Yudi Pratistha' and email 'pandeyudpratistha@gmail.com', and a menu item 'Form Data Pegawai'. The main screen displays the 'Form Data Pegawai' with the following fields: 'Nomor Induk Pegawai' (Employee ID) with value '9712305638456', 'Tanggal Lahir' (Date of Birth) with value '12 / 03 / 1997', 'Nama Pegawai' (Employee Name) with value 'Pande Made Yudi Pratistha', 'Jenis Kelamin' (Gender) with 'Laki-laki' selected, and 'Foto Pegawai' (Employee Photo) with a file path 'IMG_20190212_104258_144.jpg'. A green 'Submit' button is at the bottom.

Gambar 5. Halaman Form Dan Pengguna "Input Data" Memasukkan Data Pada Formulir

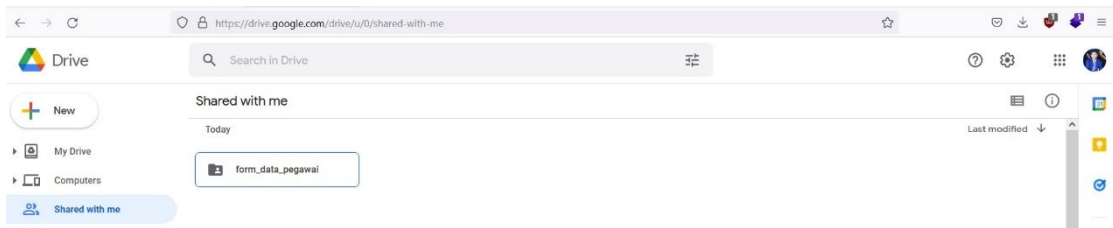
Ketika pengguna "Input Data" telah mengisi semua input form lalu menekan tombol submit. Maka data inputan *form* akan dikonversi menjadi file json dan file json beserta *attachment* akan diunggah ke Google Drive pengguna "Input Data". Setelah terunggah ke Google Drive pengguna "Input Data", system akan melakukan proses *sharing file* (json dan *attachment*) kepada pengguna "View Data" Berikut merupakan file json beserta *attachment* telah berhasil terunggah kedalam Google Drive pengguna "Input Data".



Gambar 6. Unggah Data ke Google Drive, Akun Pengguna "Input Data"

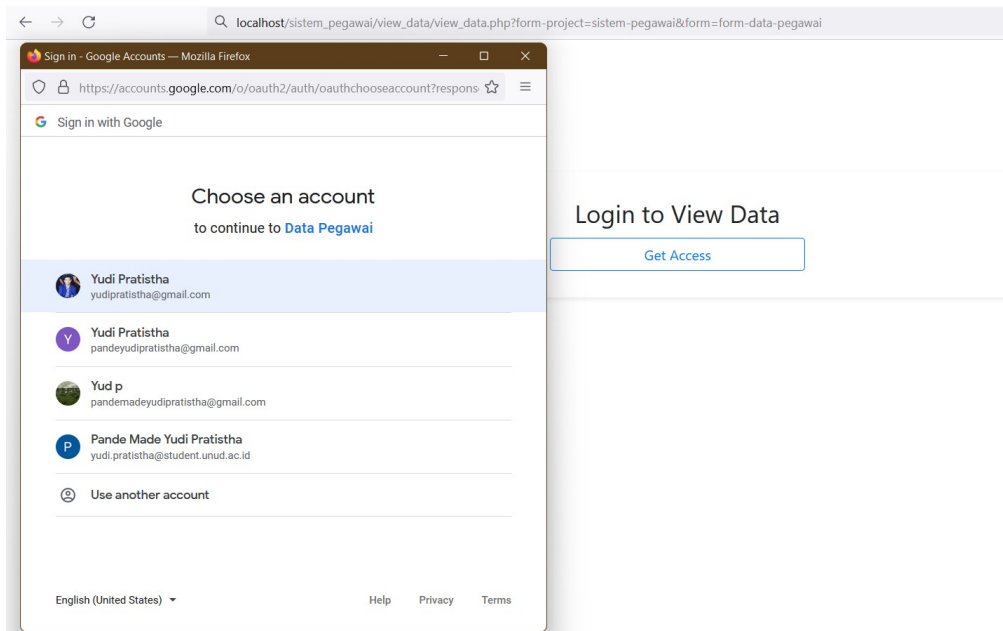
Gambar 6 merupakan direktori tempat penyimpanan file (json dan *attachment*) yang telah berhasil diunggah pada halaman Google Drive. File yang telah diunggah akan dimasukan

kedalam folder yang sesuai dengan nama form secara *programmatically*. Berikut hasil *sharing data* ke akun Google Drive pengguna "View Data".



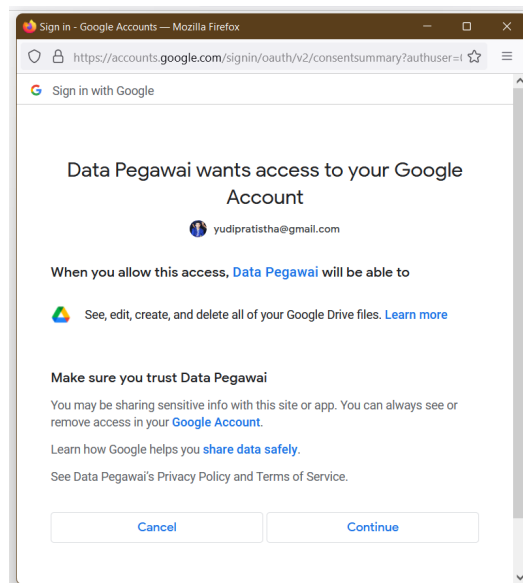
Gambar 7. *Sharing Data* ke Google Drive, Akun Pengguna "View Data"

Ketika file telah terunggah pada akun Google Drive pengguna "Input Data", system akan melakukan *sharing* folder data kepada akun Google Drive pengguna "View Data". setelah proses *sharing* berhasil dilakukan, folder data dapat dilihat pada menu "*shared with me*" pada Google Drive pengguna "View Data".



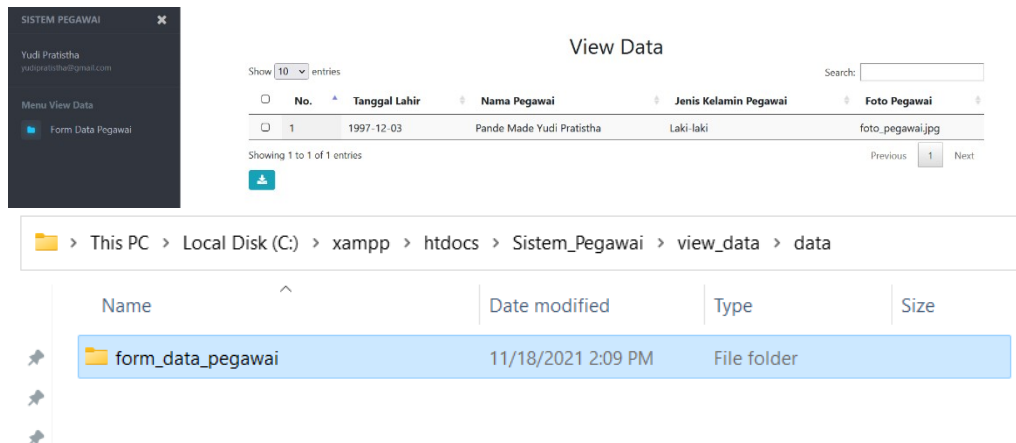
Gambar 8. Pilih atau Login Menggunakan Akun Pengguna Google Drive "View Data"

Gambar 8. merupakan halaman login "View Data" untuk memilih akun dan mendapatkan *access token* Google Drive API. Sebelum memasuki halaman *form*, pengguna "View Data" diminta untuk menekan tombol *get access* dan muncul popup *OAuth consent screen*. Ketika *OAuth consent screen* telah muncul, pengguna "View Data" akan ditampilkan akun Google yang telah tersedia pada *session browser* atau dapat melakukan proses menambahkan akun baru. Ketika akun Google telah dipilih, pengguna "View Data" diminta untuk memberikan akses kepada Google Drive seperti gambar 9.



Gambar 9. Grant Access Akun Pengguna Google Drive "View Data"

Gambar 9 merupakan *OAuth consent screen*. Ketika pengguna "View Data" telah memilih akun Google, lalu API Google Drive meminta izin untuk dapat mengelola file (*edit, delete, create dan download*) data kepada pengguna "View Data". Setelah pengguna "View Data" menekan tombol *continue*, maka Google Drive API akan memberikan *access token*, lalu sistem akan menyimpan *access token* pada *session browser* dan pengguna "View Data" akan diarahkan ke halaman "View Data" seperti gambar dibawah.



Gambar 10. Halaman View Data dan Penyimpanan Data di Folder Lokal

Halaman *view data* adalah halaman *web* untuk memperlihatkan data Google Drive yang telah berhasil di *upload* pada tampilan halaman pengisian *form*. Konten utama pada halaman *view data* adalah sebuah tabel yang berisi data *form* dari Google Drive. Untuk melakukan proses mengunduh data Google Drive, halaman ini memiliki tombol *download*. Tombol *download* ini berfungsi untuk mengunduh data baru dari Google Drive agar dapat di tampilkan pada halaman *view data*. Semua data yang telah di input pada pengujian telah berhasil diunduh dan ditampilkan melalui halaman *view data*. Ketika *file* Google Drive berhasil diunduh, *file* akan disimpan pada direktori *data* → nama *form*. Penyimpanan pada folder lokal ini berfungsi sebagai penyimpanan sistem "View Data" yang dimana sistem akan mengambil data pada *folder* dan ditampilkan pada halaman "View Data".

4.4. Pengujian Fitur Sistem

Pengujian fitur sistem pada penelitian ini menggunakan metode *blackbox*, adapun pengujian yang dilakukan yaitu proses *upload*, *sharing*, dan *download* data pada aplikasi *website* dengan Google Drive API akan diuji memakai metode pengujian *blackbox* saat melakukan proses *input* dan *output* pada sistem. Representasi dari Hasil uji coba pada sistem akan pada tabel berikut.

Tabel 2. Hasil Uji Coba Fitur Aplikasi Menggunakan Metode *Blackbox*

Uji coba fitur	Hasil yang diharapkan	Kesimpulan
Login ke halaman <i>form</i>	Pengguna "Input Data" berhasil mengakses halaman login <i>form</i>	Valid
Mendapatkan token OAuth untuk akses otentikasi ke akun Google Drive "Input Data"	Ketika user menekan tombol <i>get access</i> . Sistem akan mengecek <i>access token valid</i> dan jika <i>access token</i> tidak <i>valid</i> user akan diarahkan ke halaman <i>OAuth</i> dan memilih akun untuk mendapatkan autentikasi ke Google Drive dan <i>form</i> . Ketika telah mendapatkan <i>token user</i> akan diarahkan ke halaman <i>form</i> .	Valid
Menampilkan halaman <i>form input</i>	Sistem dapat menampilkan <i>field input form</i> pada halaman <i>web</i> .	Valid
Proses memasukkan data ke halaman <i>form input</i>	Pengguna "Input Data" berhasil menginputkan data pada <i>form</i> .	Valid
Proses mengonversi hasil <i>input form</i> ke <i>file json</i>	Ketika pengguna "Input Data" menekan tombol <i>submit</i> , sistem akan mengambil data pada setiap <i>input form</i> , sehingga data terkumpul lalu dikonversi menjadi sebuah <i>file json</i> .	Valid
Proses <i>upload file</i> (<i>json</i> dan <i>attachment</i>) menggunakan Google Drive API	Sistem akan mengunggah <i>file</i> (<i>json</i> dan <i>attachment</i>) setelah berhasil, data yang telah diunggah akan terlihat pada halaman Google Driver pengguna "Input Data".	Valid
Proses <i>sharing folder</i> pengguna "Input Data" ke akun pengguna "View Data" menggunakan Google Drive API	Sistem pengguna "Input Data" melakukan <i>sharing folder</i> ke akun Google Drive pengguna "View Data". <i>Sharing folder</i> dapat terlihat pada halaman Google Drive pengguna "View Data".	Valid
Login ke halaman view data	Pengguna "View Data" berhasil mengakses halaman <i>login view data</i> .	Valid
Mendapatkan token OAuth untuk akses otentikasi ke akun "View Data" Google Drive	Ketika pengguna "View Data" menekan tombol <i>get access</i> . Sistem akan mengecek <i>access token valid</i> dan jika <i>access token</i> tidak valid pengguna "View Data" akan diarahkan ke halaman <i>OAuth</i> dan memilih akun untuk mendapatkan autentikasi ke Google Drive dan halaman "View Data". Ketika telah mendapatkan <i>token user</i> akan diarahkan ke halaman <i>view data</i> .	Valid
Proses <i>download</i> dan <i>convert json</i> , kemudian datanya ditampilkan pada halaman "View Data".	Pengguna "View Data" dapat melihat data setelah menekan tombol unduh data. Ketika tombol unduh ditekan sistem akan mengunduh seluruh <i>data form</i> yang telah di <i>sharing</i> oleh pengguna "Input Data". Data yang diunduh akan ditampilkan pada halaman "View Data".	Valid
<i>File</i> (<i>json</i> dan <i>attachment</i>) disimpan di <i>folder</i> lokal.	Sistem berhasil menyimpan data pada <i>folder</i> lokal.	Valid

Tabel 2 adalah hasil uji coba input data menggunakan metode *blackbox* pada *website* yang menggunakan Google Drive API. Fitur sistem yang telah berhasil diuji dan sesuai dengan harapan akan mendapatkan status *valid*. Ketika uji coba fitur tidak berjalan sebagaimana mestinya akan mendapatkan status tidak *valid*. Tabel diatas dapat dilihat pada semua fitur sistem telah berfungsi.

5. Kesimpulan

Kesimpulan yang didapat berdasarkan pengujian yang dilakukan. Semua fitur pada aplikasi *web* dengan implementasi Google Drive API dapat berjalan tanpa masalah sehingga dapat menjadi alternatif media penyimpanan pada Sistem Informasi. Fitur *upload* dari Google Drive API telah berhasil mengirimkan data dari *website* penginputan data ke Google Drive. Fitur *Sharing* Data dengan Google Drive telah berhasil mengirimkan data *file json* pada Akun Google Drive *user* "Penginput Data" ke akun Google Drive *user* "View Data". Fitur *Download* dari Google Drive API telah berhasil memindahkan data dari Akun Google Drive ke halaman *View Data* dengan format *Data Table* dan disimpan pada *folder* lokal.

References

- [1] Dinatha, R. C., Sukarsa, I. M., & Cahyawan, A. A. (2016). Data Exchange Service using Google Drive API. *International Journal of Computer Applications*, 154(7), 12-16.
 - [2] Lardinois, F. (2018). Google Drive will hit a billion users this week. *TechCrunch*. diakses tanggal 15 November 2021.
 - [3] Sumiari, N. K. (2015). Pemanfaatan Google API untuk Model Interoperability Web Berbasis PHP dengan Google Drive. *Jurnal Sistem Dan Informatika (JSI)*, 10(1), 112-119.
 - [4] Lenawati, M. (2018). Penerapan Cloud Storage Dalam Perkuliahan Fakultas Teknik Universitas Pgrri Madiun. *RESEARCH: Computer, Information System & Technology Management*, 1(2), 55-58.
 - [5] Yudhana, A., Umar, R., Ahmadi, A. (2019). Digital Evidence Identification on Google Drive in Android Device Using NIST Mobile Forensic Method. *Scientific Journal of Informatics*, 6(1). 54-63.
 - [6] Liu, A., Yu, T. (2018). Overview of Cloud Storage and Architecture. *International Journal of Scientific & Technology Research*, 3-5.
 - [7] IBM Cloud Education. (2020). What is an Application Programming Interface (API). IBM Cloud Education. diakses tanggal 15 November 2021.
 - [8] Dan McGrath. (2015). Introduction to Google Drive API. Google Developers. <https://developers.google.com/drive/api/v3/about-sdk>. diakses tanggal 15 November 2021.
 - [9] Rengamani, J., James, F. A., Srinivasan, R., Poongavanam, S., & Vettriselvan, R. (2019). Impact on the usage of electronic data interchange (EDI) on the international shipping business in Chennai. *International Journal of Engineering and Advanced Technology*, 8(2). 418-422
 - [10] Ibnutama, K., Winata, H., & Hutasuhut, M. (2019). Web-Based College Student Assignment File Collection Application Using Google Drive API. 3(2). 34-40.
-