JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 1061-1073

PEMBANGUNAN AUTOMASI EMAIL BLAST PADA APLIKASI DOCUMENT SHARING MENGGUNAKAN GMAIL API DI PT XYZ

Ivan Andika Surya*[1)], Yeremia Alfa Susetyo[2)]
1. Universitas Kristen Satya Wacana, Indonesia
2. Universitas Kristen Satya Wacana, Indonesia

Article Info
Kata Kunci: Automasi; Gmail; penyebaran surel; perangkat lunak;

Keywords: *Automation; email blast; Gmail; software;*

Article history:
Received 27 June 2023
Revised 11 July 2023
Accepted 25 July 2023
Available online 1 September 2023

DOI :
https://doi.org/10.29100/jipi.v8i3.4031

* Corresponding author.
Corresponding Author
E-mail address:
672019171@student.uksw.edu

ABSTRAK
Sistem pengelolaan surel yang efektif dibutuhkan untuk membantu proses bisnis perusahaan terutama di bidang ritel. Pengelolaan penyebaran surel secara manual memiliki berbagai kelemahan yang timbul dari keterbatasan manusia seperti lupa dalam pengiriman surel, keterlambatan penyampaian informasi, dan lain-lain. Hal-hal semacam itu dapat menghambat proses bisnis perusahaan. Penelitian ini bertujuan untuk membangun arsitektur perangkat lunak automasi *email blast* pada aplikasi Document Sharing menggunakan Gmail API dan beberapa layanan dari Google Cloud Platform. Lalu untuk memastikan perangkat lunak bekerja dengan baik dan sesuai dengan kebutuhan pengguna, perangkat lunak akan melalui tahap pengujian *black box*. Setelah dilakukan pengujian, perangkat lunak akan diluncurkan sebagai solusi dari permasalahan yang dihadapi perusahaan. Penelitian ini menghasilkan sebuah perangkat lunak automasi *email blast* berbasis web. Dengan adanya perangkat lunak tersebut, PT XYZ sudah tidak perlu melakukan pengiriman ulang surel secara manual. Sehingga proses bisnis dan penyebaran informasi yang terjadi di PT XYZ menjadi efisien, cepat, dan terstruktur.

ABSTRACT
An effective email management system is required to assist the business processes of a company, especially in the retail sector. Manual email distribution management has various drawbacks arising from human limitations, such as forgetfulness in sending emails, delayed information delivery, and other such factors, which can slow down the company's business processes. This research aims to develop a software architecture for automating email blast on Document Sharing application using the Gmail API and several services from the Google Cloud Platform. To ensure that the software operates effectively and meets user requirements, the software will go through black box testing. After testing, the software will be launched as a solution to the company's problems. This research produces a web-based email blast automation software. With this software, PT XYZ no longer needs to manually resend emails. As a result, the business process and information distribution at PT XYZ become efficient, fast, and well-structured.

I. PENDAHULUAN

PERKEMBANGAN teknologi informasi semakin dibutuhkan dalam berbagai bidang kehidupan. Adanya sistem informasi mengubah pola dan cara penyelesaian suatu pekerjaan, salah satunya adalah dalam proses bisnis perusahaan [1]. Misalnya untuk mempermudah pengelolaan data, dokumen, serta surat-menyurat. Seiring perkembangan zaman, kini berbagai perusahaan menggunakan komunikasi digital untuk keperluan surat-menyurat, salah satunya adalah berkomunikasi menggunakan surel atau biasa disebut dengan *email*. Surel masih digunakan pada perusahaan-perusahaan besar untuk berkomunikasi, baik antar pekerja atau menyalurkan informasi kepada konsumennya [2]. Surel tetap digunakan karena dianggap formal dan mudah digunakan, sehingga dalam berbagai keperluan surel masih menjadi sarana komunikasi utama.

PT XYZ merupakan salah satu perusahaan retail terbesar di Indonesia yang memiliki ratusan ribu pekerja dan menggunakan surel sebagai sarana berkomunikasi sehari-hari. Bagi perusahaan retail, komunikasi adalah hal yang penting terutama dalam penyebaran informasi [3]. Di sisi lain, perusahaan memiliki pembagian divisi atau bidang yang memungkinkan setiap orang memiliki perbedaan hak dalam mengakses dokumen perusahaan. Pembagian hak akses diperlukan untuk memberikan keamanan pada dokumen yang disimpan pada *cloud storage*. Keamanan yang dimaksud adalah melindungi informasi dalam dokumen dari ancaman penyalahgunaan, pengungkapan,

pengubahan, atau hal lain yang bukan wewenang dari pemilik dokumen [4]. Tanpa adanya sistem pengelolaan surel mengakibatkan pekerjaan berjalan tidak efektif karena pengelolaan dilakukan secara manual. Hal tersebut memungkinkan terjadinya kesalahan seperti lupa dalam pengiriman, kesalahan jadwal, penumpukan pekerjaan, dan hal lain yang bersifat manusiawi. Belum adanya sistem terintegrasi juga dapat membuka peluang terjadinya *fraud* [5]. Dari berbagai kesalahan tersebut proses penyebaran informasi menjadi terhambat dan akan menimbulkan berbagai permasalahan lain terkait proses bisnis perusahaan.

Dari permasalahan tersebut, diperlukan sistem pengelolaan surel yang terstruktur dan mudah digunakan. Google Gmail API menyediakan solusi dari permasalahan tersebut, dimana teknologi ini dapat digunakan untuk membangun sistem pengelolaan surel. Gmail API termasuk ke dalam arsitektur REST-API yang dapat digunakan untuk membaca, mengirim surel, dan lain-lain [6]. Gmail API memungkinkan sebuah surel dikirimkan secara serentak dalam waktu bersamaan. Gmail API termasuk dalam layanan Google Cloud Platform yang dapat digunakan untuk membangun sebuah arsitektur perangkat lunak. Google Cloud Platform (GCP) adalah kumpulan layanan komputasi awan yang disediakan oleh Google. Didalamnya terdapat berbagai layanan yang dapat digunakan dalam pembangunan perangkat lunak seperti Gmail API, Google App Engine, Google Cloud SQL, Google Cloud Storage, dan lain-lain.

Berbeda dengan teknologi Simple Mail Transfer Protocol (SMTP) yang membutuhkan adanya *server* tersendiri untuk proses transfer surel, Gmail API menawarkan kecepatan dan kemudahan menangani proses transfer surel tanpa harus membangun sebuah *server*. Dengan menggunakan API, perangkat lunak yang dibangun akan lebih fleksibel karena proses transfer surel ditangani oleh Google Workspace. Maka dari itu untuk melakukan *email blast* dapat dibangun sebuah perangkat lunak yang terintegrasi dengan teknologi Gmail API.

Perangkat lunak tidak sekadar melakukan pengelolaan penyebaran surel, namun juga menjalankan automasi *email blast*. Untuk melakukan automasi, Google Cloud Platform menyediakan App Engine sebagai *hosting* dari perangkat lunak yang dibangun. Pada App Engine yang berbasis Linux tersedia Cron (*job scheduler*) sehingga dapat digunakan untuk melakukan automasi pengiriman surel dalam rentang waktu atau interval yang ditentukan dibalik sistem operasi [7].

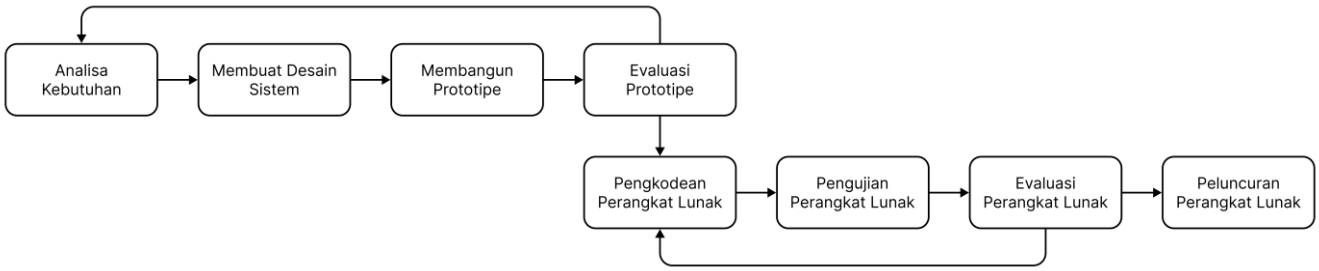
Penelitian sebelumnya yang berjudul "WhatsApp Blast Sebagai Media Promosi Makanan Ringan Keripik R&R" oleh Randitiya Analis Putra (2022) membahas mengenai pembangunan perangkat lunak yang berkaitan dengan penggunaan WhatsApp sebagai media promosi untuk produk makanan ringan keripik R&R [8]. Pada penelitian tersebut menggunakan WhatsApp *blast* untuk melakukan penyebaran informasi. Dengan menggunakan WhatsApp maka perlu dilakukan pencatatan nomor telepon yang biasanya bersifat pribadi [9]. Sedangkan menggunakan Gmail API, tidak diperlukan nomor telepon karena secara umum perusahaan memberikan alamat surel untuk setiap pekerja yang terdaftar, sehingga penyebaran informasi dapat dilakukan melalui surel. Dengan demikian Gmail API akan berguna untuk mendukung proses komunikasi di dalam perusahaan.

Penelitian yang dilakukan oleh Suci Ati (2017) berjudul "Strategi E-mail Blast dan SMS Blast dalam E-Public Relations" membahas mengenai analisa *email blast* dan *SMS blast* yang digunakan sebagai media promosi Hotel Aston di Kota Solo [10]. Teknologi *email blast* dan *SMS blast* digunakan untuk menjangkau pengguna baru dan berkomunikasi dengan pelanggan lama. Penelitian tersebut menggunakan metode yang sama yaitu *email blast*, namun pembahasan penelitian tersebut berfokus pada analisis penggunaan *email blast* tanpa menjelaskan teknologi yang digunakan. Penelitian tersebut juga tidak membahas bagaimana implementasi teknologi *email blast*. Sedangkan pada penelitian ini menjelaskan bagaimana Gmail API dapat digunakan untuk melakukan *email blast* hingga dilakukan automasi. Pembahasan mengenai Gmail API dalam penelitian ini merupakan hal yang belum pernah dibahas sebelumnya.

Penelitian lain pada tahun 2022 oleh Rizki Briandana, Tirta Lestari, dan Rustono Farady Marta berjudul "Efektivitas Iklan Melalui SMS Blast Terhadap Keputusan Pembelian Konsumen" mengangkat topik tentang pengaruh *SMS blast*, dimana penelitian tersebut menggunakan jenis penelitian kualitatif sehingga dilakukan analisis mendalam dari hasil penelitian lapangan [11]. Penelitian tersebut tidak membahas mengenai bagaimana teknologi yang digunakan dalam implementasi perangkat lunak. Penelitian tersebut menggunakan *SMS blast* yang terbatas pada kebutuhan data pribadi nomor telepon pelanggan. Sedangkan perusahaan menggunakan surel sebagai sarana komunikasi yang formal, serta setiap pekerja memiliki alamat surel yang terdaftar di perusahaan. Dengan menggunakan Gmail API memungkinkan komunikasi terjadi tanpa melibatkan data nomor telepon. Pada penelitian ini juga akan menjelaskan bagaimana Gmail API dapat diimplementasikan ke dalam sebuah perangkat lunak hingga dilakukan automasi.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 1061-1073

II. METODE PENELITIAN



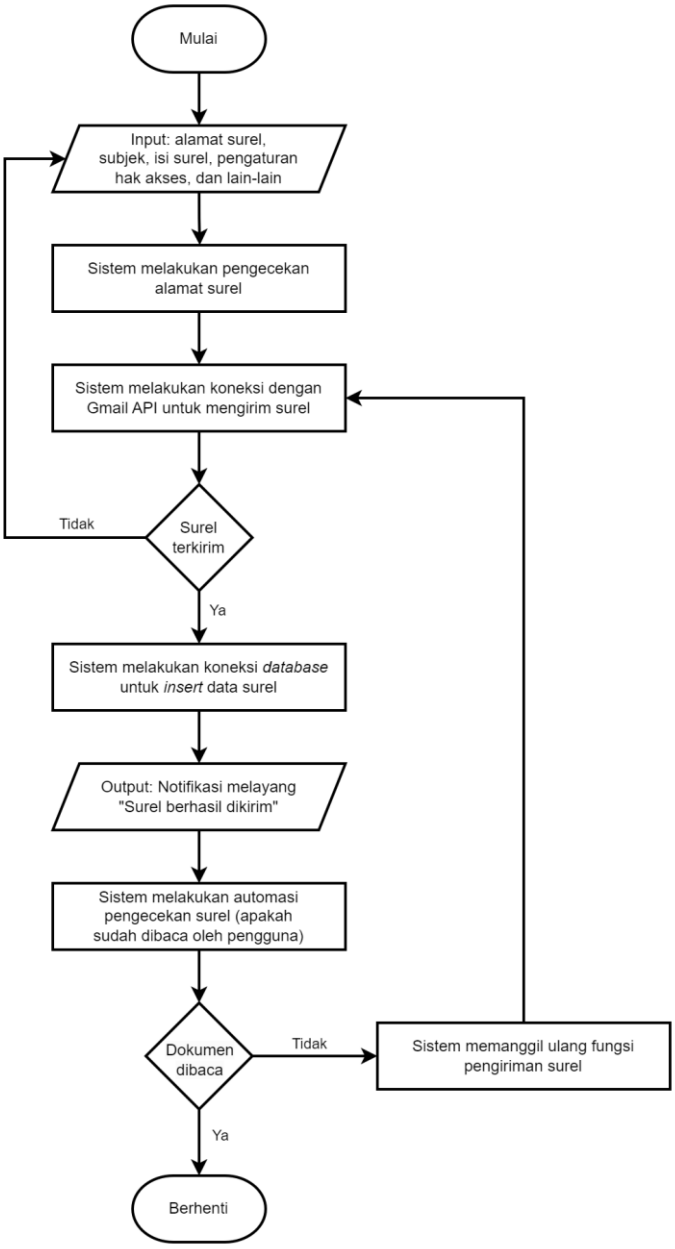
Gambar 1. Metode penelitian

Gambar 1 menjelaskan tahapan implementasi perangkat lunak untuk memastikan sistem memenuhi kebutuhan dan tujuan dari perancangan. Tahap-tahap tersebut meliputi: analisa kebutuhan, membuat desain sistem, membangun prototipe, evaluasi prototipe, pengkodean perangkat lunak, pengujian perangkat lunak, evaluasi perangkat lunak, dan peluncuran perangkat lunak.

Tahap analisa kebutuhan adalah proses mencatat dan mengidentifikasi kebutuhan yang diperlukan dalam sistem [12]. Dari permasalahan yang dihadapi perusahaan, dibutuhkan sistem pengelolaan automasi *email blast* dalam sebuah perangkat lunak *document sharing*. Kebutuhan yang utama adalah Gmail API sebagai pengelola surel. Selain itu, diperlukan layanan lain seperti Cloud SQL, Cloud Storage, Cron Job, dan lain-lain. Perangkat lunak dibangun menggunakan bahasa pemrograman Python dan *framework* Flask. Python dipilih karena fleksibilitasnya untuk berbagai hal, contohnya untuk membangun perangkat lunak, analisis data, dan lain-lain. Selain itu Python juga menyediakan banyak *library* yang dapat digunakan pengembang untuk mempermudah pekerjaan [13]. Sedangkan *framework* Flask digunakan karena tanpa biaya, menawarkan kemudahan, fleksibilitas, dan banyaknya dukungan yang tersedia sehingga dapat membantu dalam proses pengembangan perangkat lunak. Dalam penelitian ini Flask digunakan untuk menghubungkan bagian *back-end* dengan *front-end*.

Setelah menentukan kebutuhan sistem, desain sistem dibuat untuk memastikan keluaran perangkat lunak memenuhi kebutuhan pengguna. Desain sistem dapat didefinisikan menggunakan Unified Modeling Language (UML). UML merupakan suatu teknik untuk membuat representasi visual untuk mendokumentasikan, menentukan spesifikasi, dan membangun perangkat lunak [14]. Beberapa UML yang digunakan dalam penelitian ini adalah diagram alur, diagram *use case*, dan diagram aktivitas.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 1061-1073



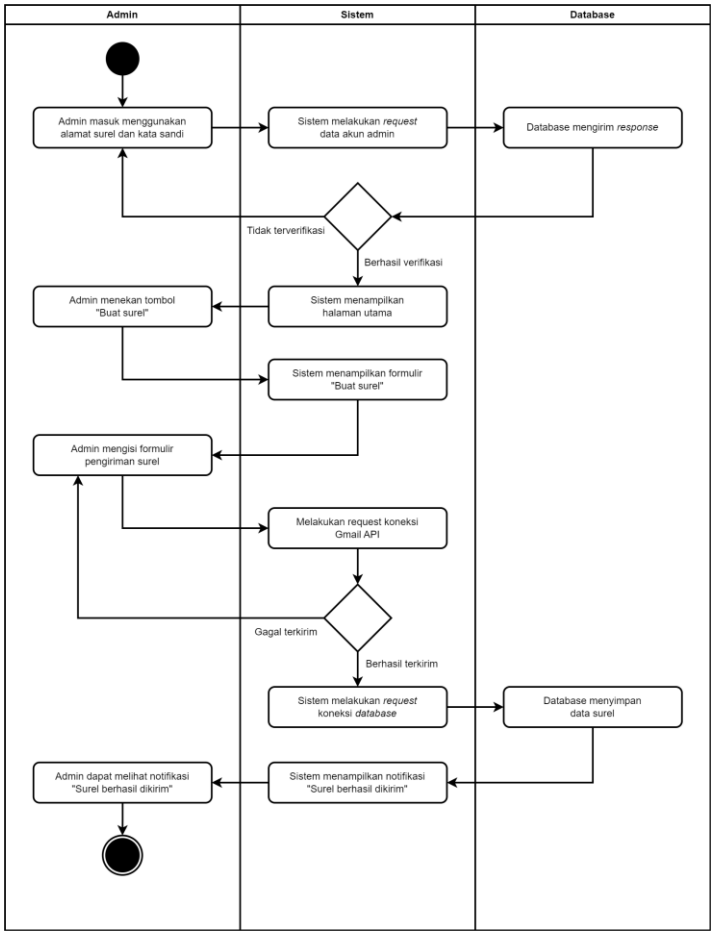
Gambar 2. Diagram alur pengiriman surel

Gambar 2 merupakan diagram alur yang digunakan untuk menjelaskan alur, langkah-langkah, atau urutan proses perangkat lunak mulai dari dijalankan hingga berakhir. Diawali dengan perangkat lunak memulai proses. Kemudian sistem membutuhkan masukan dari pengguna, seperti alamat surel penerima, subjek, isi surel (*body*), pengaturan hak akses, dan lain sebagainya. Setelah itu perangkat lunak melakukan pengecekan alamat surel yang dimasukkan. Selanjutnya perangkat lunak akan melakukan koneksi dengan Gmail API untuk mengirim surel. Jika pengiriman gagal, maka akan kembali ke tahap sebelumnya yaitu menerima masukan pengguna. Jika pengiriman berhasil, perangkat lunak melakukan koneksi ke database untuk menyimpan data informasi surel. Lalu perangkat lunak akan memberikan keluaran notifikasi surel berhasil dikirim. Dalam rentang waktu yang ditentukan, perangkat lunak melakukan pengecekan apakah surel sudah dibaca oleh pengguna. Jika belum dibaca, sistem akan melakukan automasi pemanggilan fungsi pengecekan dan pengiriman surel sebagai pengingat. Jika sudah dibaca, perangkat lunak akan mencapai kondisi berhenti.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 1061-1073

Gambar 3. Diagram *use case* perangkat lunak

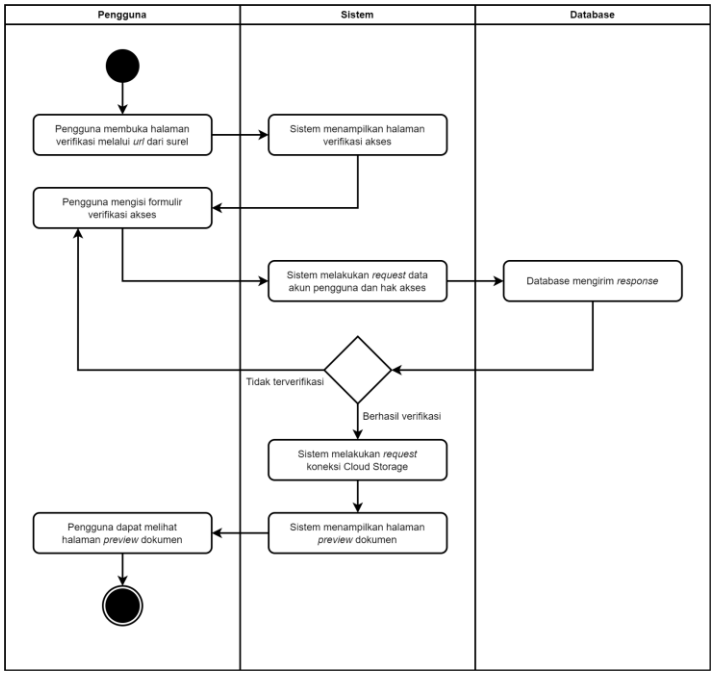
Gambar 3 adalah diagram *use case* yang digunakan untuk mengetahui hubungan antara kebutuhan sistem. Komponen di dalam diagram *use case* adalah aktor dan *use case* itu sendiri [15]. Di dalam diagram ini terdapat dua aktor yaitu admin dan pengguna. Admin memiliki hak mengakses halaman utama daftar surel, menulis surel dan mengatur hak ases dokumen, mengirim surel kepada pengguna, melihat detail surel, mengelola pengguna beserta hak aksesnya, serta menambahkan dokumen. Sedangkan pengguna dapat membuka halaman verifikasi melalui *url* yang diterima dari surel, melakukan verifikasi akses dokumen, dan membaca isi dokumen.



Gambar 4. Diagram aktivitas admin

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 1061-1073

Gambar 4 adalah diagram aktivitas admin yang menjelaskan alur aktivitas aktor admin. Pada diagram tersebut, proses dimulai dengan admin melakukan masuk menggunakan akun admin. Setelah berhasil masuk, admin akan diarahkan menuju halaman utama yang berisi daftar surel yang telah terkirim. Pada setiap halaman, disediakan tombol untuk menambahkan surel baru. Dengan menekan tombol tambah surel, admin dapat mengisi formulir, mengatur hak akses dokumen yang dibagikan, dan mengirim surel. Selanjutnya sistem akan melakukan pengiriman surel dan database akan mencatat data surel yang terkirim. Setelah surel berhasil dikirim admin dapat melihat notifikasi surel berhasil dikirim dan aktivitas berhenti.



Gambar 5. Diagram aktivitas pengguna

Gambar 5 adalah diagram aktivitas untuk menjelaskan aktivitas pengguna. Pengguna memulai proses dengan membuka halaman verifikasi melalui *url* yang diterima melalui surel. Pengguna dapat mengisi formulir verifikasi menggunakan alamat surel dan kata sandi yang sudah terdaftar. Jika akun pengguna tidak terverifikasi, maka akan dikembalikan ke halaman verifikasi akses. Jika berhasil terverifikasi maka pengguna akan diarahkan ke halaman *preview* dokumen dan dapat membaca isi dokumen, lalu proses berhenti.

Setelah desain sistem dibuat, tahap selanjutnya adalah membangun prototipe perangkat lunak. Prototipe ini dapat dilihat oleh pengembang maupun pengguna saat dalam proses pengembangan. Sehingga dapat dilakukan penyesuaian kebutuhan sebelum perangkat lunak benar-benar diluncurkan. Dalam tahap ini, perangkat lunak belum harus dapat digunakan sepenuhnya. Namun setidaknya pengembang sudah dapat memberikan gambaran penggunaan perangkat lunak, terutama pada bagian-bagian yang dibutuhkan. Prototipe harus sudah dapat menjalankan fungsi utama pengiriman surel dan automasi, namun untuk beberapa bagian lain seperti halaman utama, halaman masuk, dan halaman lain belum perlu dibuat. Setelah prototipe dibuat, dilakukan evaluasi untuk memastikan kebutuhan sudah terpenuhi dan sesuai tujuan perancangan perangkat lunak.

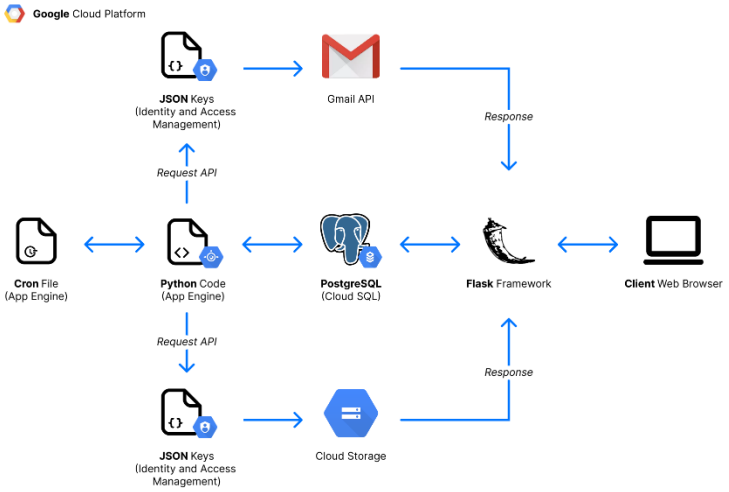
Saat prototipe sudah memenuhi kebutuhan, dilanjutkan ke tahap pengkodean keseluruhan perangkat lunak. Pengembang mulai melakukan pengkodean dan implementasi teknologi yang akan digunakan. Jika perangkat lunak sudah dapat berjalan, dilakukan pengujian menggunakan *black box testing* untuk memastikan perangkat lunak berjalan dengan baik dan sesuai analisa kebutuhan pada tahap awal [16]. Jika ada bagian yang perlu ditambahkan atau diperbaiki, proses pengembangan kembali ke tahap pengkodean perangkat lunak dan dilanjutkan kembali ke tahap pengujian. Proses evaluasi dan perbaikan ini dapat diulang hingga mendapatkan kondisi yang sesuai. Setelah dilakukan pengujian black box dan evaluasi, dilakukan pengujian System Usability Scale (SUS) untuk melihat kondisi perangkat lunak saat digunakan oleh pengguna. Jika keseluruhan perangkat lunak sudah sesuai dengan kebutuhan, perangkat lunak dapat diluncurkan dan siap digunakan oleh pengguna.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 1061-1073

III. HASIL DAN PEMBAHASAN

Berdasarkan tahap-tahap desain dan perancangan perangkat lunak yang dibuat, penelitian ini menghasilkan perangkat lunak web yang dapat digunakan untuk melakukan automasi *email blast* untuk pengelolaan *document sharing*. Keluaran dari penelitian ini berupa web yang dibangun menggunakan bahasa pemrograman Python dan *framework* Flask.

A. *Arsitektur Sistem*



Gambar 6. Diagram arsitektur

Gambar 6 adalah diagram arsitektur yang digunakan sebagai dasar dari pembuatan perangkat lunak. Arsitektur dibangun menggunakan lingkungan Google Cloud Platform (GCP) yang menyediakan berbagai layanan seperti Gmail API, Cloud Storage, App Engine, Cloud SQL, dan Identity and Access Management (IAM). Gmail API digunakan untuk mengirim *email blast* dengan menggunakan perangkat lunak web yang terhubung ke Google Cloud Platform. App Engine digunakan sebagai *hosting* dari kode program Python yang di-*deploy* menjadi sebuah *service*. Pada App Engine yang berbasis Linux, layanan Cron dapat dijalankan untuk melakukan automasi. Cloud SQL dengan database PostgreSQL digunakan untuk mengelola data, termasuk informasi akun, daftar penerima surel, subjek surel, dan lain-lain. IAM digunakan untuk mengatur hak akses dan mendapatkan API Keys, sehingga seluruh layanan dapat saling terhubung dan membentuk arsitektur perangkat lunak.

Saat kode program Python dijalankan, dibutuhkan *request* API untuk mendapatkan *response*. Yang pertama adalah *request* pada Gmail API saat melakukan pengiriman surel, dan juga dilakukan *request* koneksi Cloud Storage saat membaca dokumen. *Request* tersebut akan dikembalikan ke Flask dan ditampilkan kepada pengguna. Untuk melakukan berbagai pengecekan hak akses dan informasi surel, dibutuhkan *request* koneksi untuk mendapatkan data dari database yang berada pada layanan Cloud SQL. Lalu Cron akan terus berjalan sesuai dengan pengaturan waktu yang dituliskan pada berkas Cron. Layanan Cron akan memanggil fungsi pengecekan yang sudah dipersiapkan pada perangkat lunak.

B. *Implementasi*

TABEL I
KODE PROGRAM PERMINTAAN KONEKSI GMAIL API

Kode Program

```
CLIENT_SECRET_FILE = 'ServiceKeyGmailAPI.json'
API_NAME = 'gmail'
API_VERSION = 'v1'
SCOPES = ['https://mail.google.com/']
service = Create_Service(CLIENT_SECRET_FILE, API_NAME, API_VERSION, SCOPES)
```

Tabel I berisi kode program yang digunakan untuk mengirim permintaan koneksi Gmail API dan membutuhkan berkas API Keys berisi informasi seperti kode identitas proyek, nama proyek, *private key*, serta informasi lain yang dibutuhkan. API Keys bisa didapatkan dari layanan Identity and Access Management (IAM). Dengan menggunakan API Keys, layanan Gmail API dapat dijalankan langsung melalui kode program Python. Setelah melakukan *request*, maka *response* dikembalikan menuju Flask untuk ditampilkan kepada pengguna.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 1061-1073

TABEL II
KODE PROGRAM PERMINTAAN KONEKSI DATABASAE PADA CLOUD SQL

Kode Program

```
class Database():
  def __init__(self, param):
        self._dict_connection=getConnection()
        self._connection_det=self._dict_connection.get(param)
     try:
        self._conn=psycopg2.connect(self._connection_det)
        self._curs=self._conn.cursor()
        self._curs.execute("SET TIMEZONE='Asia/Jakarta'")
     except psycopg2.Error as errorMessage:
        raise Exception('%s, Database : %s' % (errorMessage, 'Tidak Ditemukan'))
...
```

Tabel II berisi kode program yang digunakan untuk melakukan permintaan koneksi ke *database*. Untuk menjaga keamanan, informasi koneksi *database* seperti nama *database*, *url*, nama *user*, dan *password* disimpan sebuah berkas terpisah. Diperlukan pemanggilan fungsi `getConnection()` untuk mendapatkan informasi koneksi *database*.

TABEL III
KODE PROGRAM PERMINTAAN KONEKSI CLOUD STORAGE

Kode Program

```
os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = 'ServiceKeyCloudStorage.json'
bucketName = globalvar.getBucketName()
storageClient = storage.Client()
```

Tabel III berisi kode program untuk mengirim permintaan koneksi Cloud Storage. Sama halnya seperti pada Gmail API, Cloud Storage membutuhkan berkas API Keys yang menyimpan seluruh informasi hak akses layanan. Pada Cloud Storage dibutuhkan *bucket* yang dapat dianalogikan sebagai wadah atau tempat penyimpanan berkas dokumen. Untuk memilih *bucket*, dapat dilakukan dengan memanggil fungsi `getBucketName()`. Lalu *response* dari API dikembalikan menuju Flask untuk ditampilkan kepada pengguna.

TABEL IV
KODE PROGRAM MENGIRIM SUREL

Kode Program

```
mimeMessage = MIMEMultipart()
mimeMessage['to'] = mailDestination
mimeMessage['subject'] = mailSubject
mimeMessage.attach(MIMEText(mailBody+ fileUrl, 'plain'))

rawString = base64.urlsafe_b64encode(mimeMessage.as_bytes()).decode()
service.users().messages().send(userId= 'me',body={'raw':rawString}).execute()
```

Tabel IV berisi kode program yang digunakan untuk membuat surel menggunakan kelas `MIMEMultipart` dari *library* `email`. Penerima surel ditentukan pada `mimeMessage['to'] = mailDestination`, dan subjek ditentukan dengan `mimeMessage['subject'] = mailSubject`. Isi dari surel diubah ke dalam sebuah teks dengan kelas `MIMEText(mailBody+fileUrl, 'plain')` dan dilampirkan pada *body* surel dengan menggunakan fungsi `mimeMessage.attach(MIMEText(mailBody+fileUrl, 'plain'))`.

Pesan tersebut kemudian dienkode menggunakan base64 dengan menggunakan fungsi `base64.urlsafe_b64encode(mimeMessage.as_bytes()).decode()` dan disimpan dalam variabel `rawString`. Surel dapat dikirim menggunakan method `service.users().messages().send()`, dengan `userId='me'` yang menentukan pengirim surel dan `body={'raw': rawString}` yang berisi teks hasil enkode sebagai isi dari surel. Tahap terakhir, method `.execute()` dipanggil untuk mengirimkan surel.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 1061-1073

TABEL V
KODE PROGRAM AUTOMASI PENGECEKAN SUREL

Kode Program

```
queryGetIdShare = connection.exec("""
    SELECT "idShare"
    FROM "tb_statusRead"
    WHERE "statusRead" = '0'
    """)
listUnreadIdShare=tuple(set(x[0] for x in queryGetIdShare))

listEmails=[]
for x in listUnreadIdShare:
    queryGetEmails = connection.exec("""
        SELECT "email"
        FROM "tb_user"
        WHERE "idUser" IN (
            SELECT "idUser"
            FROM "tb_statusRead"
            WHERE "idShare" = '%s'
            AND "statusRead" = '0'
            )""" % x)
    listEmails.append(', '.join(y[0] for y in queryGetEmails))

listMailData=[]
for x in listUnreadIdShare:
    queryGetMailData=connection.exec("""
        SELECT "mailSubject", "mailBody"
        FROM "tb_share"
        WHERE "idShare" = '%s'
        """ % x)
    listMailData.append(queryGetMailData[0])
```

Tabel V berisi sebuah kode program fungsi untuk melakukan pengecekan informasi surel. Pengecekan dilakukan melalui pembacaan dan pencocokan data dari seluruh informasi surel di dalam *database*. Kode program ini adalah fungsi yang akan dipanggil pada Cron untuk melakukan automasi pengecekan surel. Dengan menggunakan data dari *database* sistem akan dapat menentukan surel mana yang belum dibaca oleh pengguna, dan pengguna mana yang belum membaca dokumen yang dibagikan.

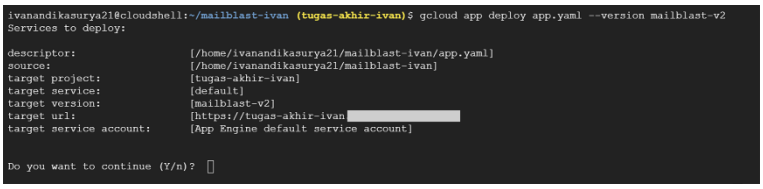
TABEL VI
KODE PROGRAM CRON

Kode Program

```
cron:
  - description: "Email Reminder"
    url: /cron
    target: beta
    schedule: 0 7 * * 0
```

Tabel VI berisi kode program dari berkas cron.yaml yang di-*deploy* pada App Engine. Berkas ini digunakan untuk melakukan automasi pemanggilan fungsi dalam frekuensi waktu tertentu. Dalam penelitian ini, Cron digunakan untuk melakukan pemanggilan fungsi pengecekan surel yaitu melalui *routing* `/cron`. Jika terdapat pengguna yang belum membaca dokumen melalui *url* surel, maka dikirimkan ulang surel sebagai pengingat.

Frekuensi pemanggilan fungsi dapat diubah sesuai kebutuhan. Pada bagian `schedule: 0 7 * * 0` menunjukkan bahwa automasi pemanggilan fungsi akan dilakukan setiap hari Senin pukul 07.00.



Gambar 7. Tampilan saat melakukan *deploy* perangkat lunak pada App Engine

Gambar 7 adalah tampilan saat perangkat lunak di-*deploy* pada App Engine agar menjadi sebuah *service*. App Engine menjadi *hosting* untuk perangkat lunak web yang dibangun. Setelah perangkat lunak berhasil dijalankan

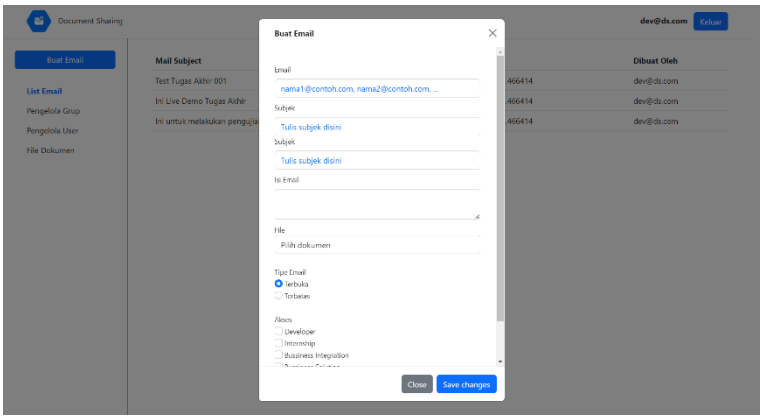
JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 1061-1073

sebagai *service*, perangkat lunak akan dapat diakses dari manapun melalui peramban. Seluruh aktivitas *request* dan *response* akan ditangani oleh App Engine.

Gambar 8. Tampilan saat melakukan *deploy* Cron

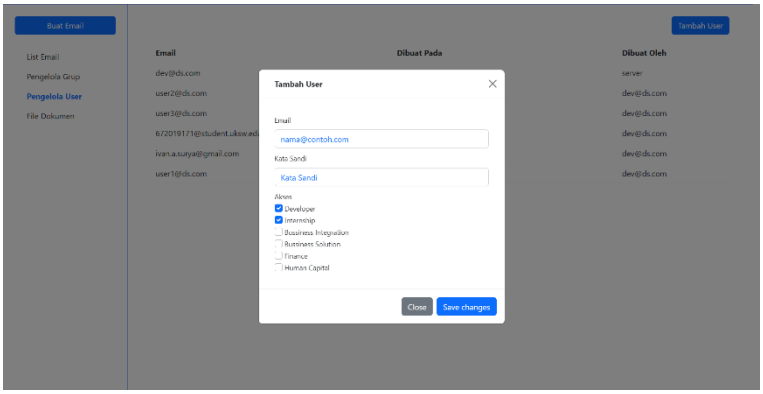
Gambar VIII adalah tampilan saat melakukan *deploy* berkas Cron yang berisi pengaturan frekuensi automasi pengecekan surel. Setelah berkas Cron berhasil di-*deploy*, automasi pemanggilan fungsi pengecekan surel akan berjalan. Karena menggunakan App Engine sebagai *hosting*, maka Cron dapat berjalan secara terus menerus tanpa terbatas waktu.

C. Tampilan Perangkat Lunak



Gambar 9. Tampilan tambah surel

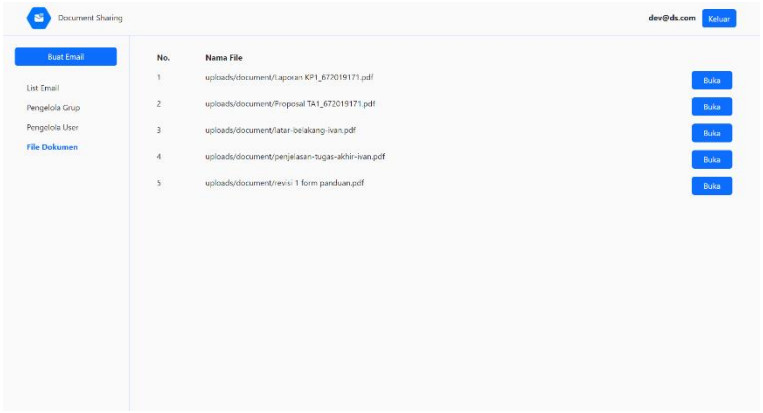
Gambar 9 adalah tampilan formulir pembuatan surel baru. Admin dapat menulis lebih dari satu alamat surel. Sehingga dalam sekali pengiriman, dapat serentak terkirim ke banyak penerima. Pada bagian ini, admin dapat mengatur hak akses dokumen yang akan dikirimkan ke pengguna. Admin diberikan pilihan apakah dokumen dapat diakses secara terbuka oleh semua pengguna, atau hanya dapat diakses oleh pengguna yang memiliki akun terdaftar saja. Selain itu, admin juga dapat memilih akses yang harus dimiliki pengguna untuk membaca isi dokumen.



Gambar 10. Tampilan tambah pengguna

Gambar 10 adalah tampilan *modal* untuk menambahkan pengguna, mengatur kata sandi, serta hak akses pengguna. Alamat surel pengguna yang terdaftar akan memiliki hak akses sesuai yang dipilih oleh admin. Setiap pengguna dapat memiliki lebih dari satu jenis hak akses. Hak akses ini yang digunakan untuk membatasi akses pengguna terhadap dokumen. Jika pengguna tidak memiliki akses yang sesuai, maka dokumen tidak akan dapat diakses. Sebaliknya jika pengguna memiliki akses yang sesuai maka pengguna dapat membaca dokumen tersebut.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 1061-1073



Gambar 11. Tampilan halaman pengelolaan berkas dokumen

Gambar 11 adalah tampilan halaman pengelolaan berkas dokumen untuk mengelola berkas yang tersimpan di dalam Cloud Storage. Lokasi dokumen yang tersimpan di dalam Cloud Storage dapat dilihat pada halaman ini. Selain itu admin juga dapat mengakses halaman *preview* masing-masing berkas dokumen tersebut.

D. Pengujian Black Box

Dilakukan pengujian menggunakan metode *black box* untuk memastikan perangkat lunak berjalan dengan baik.

TABEL VII
PENGUJIAN BLACK BOX

Skenario pengujian	Harapan hasil	Hasil pengujian	Jumlah Percobaan	Jumlah Percobaan Berhasil	Persentase Keberhasilan	Kesimpulan
Admin menekan tambah surel	Formulir terbuka dan admin dapat mengisi formulir	Formulir terbuka dan dapat diisi	10	10	100%	Sukses
Admin mengisi kolom surel tujuan dengan jumlah satu alamat surel	Surel terkirim dan diterima oleh satu pengguna yang sesuai	Surel dapat terkirim dan diterima oleh satu pengguna yang sesuai	10	10	100%	Sukses
Admin mengisi kolom surel tujuan dengan jumlah dua atau lebih alamat surel	Surel terkirim dan diterima oleh dua atau lebih pengguna yang sesuai	Surel dapat terkirim dan diterima oleh dua atau lebih pengguna yang sesuai	10	10	100%	Sukses
Cron dijalankan, melakukan pengecekan dalam rentang waktu setiap satu menit	Surel yang belum dibaca pengguna akan dikirimkan ulang sebagai pengingat	Surel yang belum dibaca pengguna dikirimkan ulang sebagai pengingat	10	10	100%	Sukses
Cron dijalankan, melakukan pengecekan dalam rentang waktu setiap satu hari	Surel yang belum dibaca pengguna akan dikirimkan ulang sebagai pengingat	Surel yang belum dibaca pengguna dikirimkan ulang sebagai pengingat	10	10	100%	Sukses
Cron dijalankan, melakukan pengecekan dalam rentang waktu setiap satu minggu	Surel yang belum dibaca pengguna akan dikirimkan ulang sebagai pengingat	Surel yang belum dibaca pengguna dikirimkan ulang sebagai pengingat	2	2	100%	Sukses
Dokumen belum dibaca oleh penggguna	Surel pengingat akan dikirimkan saat tugas Cron berjalan	Sorel pengingat dikirimkan saat tugas Cron berjalan	10	10	100%	Sukses
Dokumen sudah dibaca oleh pengguna	Surel pengingat tidak akan dikirimkan saat Cron berjalan	Surel pengingat tidak dikirimkan saat Cron berjalan	10	10	100%	Sukses
Pengguna tidak memiliki akses yang sesuai dengan dokumen	Pengguna akan dikembalikan ke halaman verifikasi dan dokumen tidak terbuka	Pengguna dikembalikan ke halaman verifikasi dan dokumen tidak terbuka	10	10	100%	Sukses
Pengguna memiliki akses yang sesuai dengan dokumen	Pengguna akan diarahkan ke halaman *preview* sesuai dokumen yang diberikan	Pengguna diarahkan ke halaman *preview* sesuai dokumen yang diberikan	10	10	100%	Sukses

Tabel VII menunjukkan hasil pengujian *black box* tahap terakhir yang dilakukan pada perangkat lunak. Pengujian yang dilakukan mulai dari fungsi pengiriman surel, baik dikirimkan secara perorangan maupun serentak ke banyak pengguna. Selain itu, dilakukan pengujian terhadap layanan Cron untuk melakukan automasi pengecekan surel dan pengujian akses dokumen. Berdasarkan hasil pengujian yang dilakukan, seluruh percobaan telah mencapai tingkat keberhasilan sebesar 100% sehingga dapat dinyatakan perangkat lunak sudah berjalan dengan baik serta sesuai dengan kebutuhan awal.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 3, September 2023, Pp. 1061-1073

E. Pengujian System Usability Scale (SUS)

Selain pengujian *black box*, dilakukan pengujian System Usability Scale (SUS) untuk mengukur seberapa mudah pengguna dalam menggunakan perangkat lunak. Berdasarkan standar yang telah ada, pengujian dilakukan dengan cara menyebarkan kuesioner berisi 10 pertanyaan. Data yang didapat dari penyebaran kuesioner akan dihitung untuk mendapatkan skor akhir.

TABEL VIII
DATA HASIL PENYEBARAN KUESIONER

Responden	Pertanyaan									
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Responden 1	5	1	4	1	5	1	5	1	4	1
Responden 2	5	1	5	1	5	1	5	1	5	1
Responden 3	5	1	5	1	5	1	5	1	5	1
Responden 4	5	1	5	2	5	1	4	1	5	1
Responden 5	4	2	4	2	5	3	4	1	2	4
Responden 6	5	2	**5**	1	5	1	5	1	5	2
Responden 7	4	2	5	2	4	2	3	2	4	4
Responden 8	5	1	5	1	5	1	5	1	5	1
Responden 9	4	3	4	3	4	2	4	3	4	3
Responden 10	5	1	5	1	5	1	5	1	5	2

Tabel VIII menunjukkan hasil pengumpulan data yang didapatkan dari penyebaran kuesioner kepada 10 pengguna perangkat lunak. Berdasarkan data tersebut, maka skor System Usability Scale dapat dihitung menggunakan perhitungan yang sesuai dengan standar.

TABEL IX
PERHITUNGAN DATA PENGUJIAN

Pertanyaan										Jumlah	Nilai (Jumlah x 2.5)
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10		
4	4	3	4	4	4	4	4	3	4	38	95
4	4	4	4	4	4	4	4	4	4	40	100
4	4	4	4	4	4	4	4	4	4	40	100
4	4	4	3	4	4	3	4	4	4	38	95
3	3	3	3	4	2	3	4	1	1	27	68
4	3	4	4	4	4	4	4	4	3	38	95
3	3	4	3	3	3	2	3	3	1	28	70
4	4	4	4	4	4	4	4	4	4	40	100
3	2	3	2	3	3	3	2	3	2	26	65
4	4	4	4	4	4	4	4	4	3	39	98
Skor Rata-rata (Hasil Akhir)											**89**

Tabel IX menunjukkan hasil perhitungan data pengujian System Usability Scale. Didapatkan skor akhir yaitu sebesar 89 yang termasuk ke dalam pengelompokan *acceptable* pada kelas B (*Excellent*). Dengan demikian dapat dinyatakan bahwa perangkat lunak dapat dioperasikan dengan baik oleh pengguna dan layak untuk digunakan.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian dan pembahasan yang telah dipaparkan, dapat disimpulkan bahwa penggunaan layanan Gmail API dan Cron di lingkungan Google Cloud Platform (GCP) memungkinkan pembangunan perangkat lunak automasi *email blast*. Gmail API dapat digunakan untuk melakukan pengelolaan surel menjadi lebih efisien, cepat, dan terstruktur. Contohnya dalam penyebaran surel, telah dilakukan pengujian bahwa Gmail API dapat melakukan pengiriman surel baik secara perorangan maupun ke beberapa pengguna secara serentak. Selain itu, dengan menggunakan perangkat lunak ini dokumen yang disebarkan bersifat aman karena dibatasi hak akses, baik hak akun pengguna maupun dari hak akses dokumen itu sendiri. Pada pengujian hak akses, terbukti bahwa dokumen tidak dapat diakses jika hak akses akun pengguna tidak sesuai dengan hak akses dokumen.

Dari hasil pengujian automasi yang dilakukan, membuktikan Cron dapat digunakan untuk menjalankan automasi pengiriman email sesuai dengan tujuan awal pengembangan perangkat lunak. Dengan menggunakan automasi tersebut PT XYZ dapat mengurangi kesalahan seperti lupa dalam pengiriman dan pembacaan email, keterlambatan penyebaran informasi, atau hal lain yang dapat terjadi.

Keuntungan dari penggunaan lingkungan Google Cloud Platform adalah fleksibilitas dan kemampuan perangkat lunak yang tidak bergantung pada *server* fisik, melainkan sudah berupa *cloud computing*. Dengan menggunakan API yang tersedia pada Google Cloud Platform, pengembang tidak perlu melakukan perawatan pada perangkat keras *server*. Selain itu, pengguna juga dapat menggunakan perangkat lunak tanpa dibatasi oleh keterbatasan perangkat keras *server*. Dari penelitian yang telah dilakukan, masih terdapat banyak fitur yang dapat dikembangkan lebih lanjut. Sebagai saran, perangkat lunak masih belum memiliki fungsi pengecekan kesalahan yang lengkap. Sehingga untuk kedepannya dapat dibuat fungsi untuk mencatat kesalahan atau biasa disebut dengan *system log* untuk mempermudah pengguna maupun pengembang yang melakukan perawatan dan perbaikan perangkat lunak.

DAFTAR PUSTAKA

[1] F. A. Akbar, dkk. (Juni 2019). Assign: E-Learning Sederhana Berbasis Cloud Storage untuk Sekolah. *SCAN Jurnal Teknologi Informasi dan Komunikasi*. [Online]. 14(2), hal. 49-57. Tersedia: https://doi.org/10.33005/scan.v14i2.1487

[2] W. Santoso, "E-Blast: Aplikasi E-Mail Blast Berbasis Web Dengan Metode Queue Laravel Dan Masking URL," tesis (skripsi), Jurusan Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia, 2020.

[3] T. Rachel, Y. A. Susetyo. (Juni 2022). Implementasi Pengiriman Pesan Broadcast dengan Redis Pub/Sub dan Bahasa Pemrograman Nim. *Jurnal Inovtek Polbeng*. [Online]. 7(1), hal. 14-25. Tersedia: https://doi.org/10.35314/isi.v7i1.2314

[4] I. F. N. Sartika, R. Bisma. (Agustus 2021). Perancangan Sistem Informasi Manajemen Risiko berdasarkan ISO 27001:2013 (Sistem Manajemen Keamanan Informasi). *Journal of Emerging Information Systems and Business Intelligence*. [Online]. 2(3), hal. 81-86. Tersedia: https://ejournal.unesa.ac.id/index.php/JEISBI/article/view/41723

[5] L. D. Anggraini, Faradillah. (Juli 2022). Pendeteksian Fraud: Penerapan COSO pada Audit Sistem Informasi Akuntansi Pada Perusahaan Perkebunan. *Journal of Accounting Science*. [Online]. 6(2), hal. 102-110. Tersedia: https://doi.org/10.21070/jas.v6i2.1607

[6] I. Arvidsson, "En undersökning och implementering för att hitta lämpligast teknik vid e-postläsning : En jämförelse mellan Outlook Mail REST API, Gmail API och IMAP4 baserat på prestanda och säkerhet," disertasi M.Sc, Dept. Sistem dan Teknologi Informasi, Mid Sweden University, Swedia, 2020

[7] J. Moedjahedy. (Juni 2018). Implementasi Cron Job Linux Sebagai Bel Pergantian Kelas Otomatis Di Universitas Klabat. *Cogito Smart Journal*. [Online]. 4(1), hal. 1-10. Tersedia: https://doi.org/10.31154/cogito.v4i1.97.1-10

[8] R. A. Putra, "Whatsapp Blast Sebagai Media Promosi Makanan Ringan Keripik R&R,", tesis (skripsi), Jurusan Teknik Informatika, Universitas Teknologi Digital Indonesia, Yogyakarta, Indonesia, 2022.

[9] S. D. Rosadi, G. G. Pratama. (Juni 2022). Perlindungan Privasi dan Data Pribadi dalam Era Ekonomi Digital di Indonesia. *Veritas et Justitia*. [Online]. 4(1), hal. 88-110. Tersedia: http://dx.doi.org/10.25123/vej.2916

[10] S. Ati, "Strategi E-mail Blast dan SMS Blast dalam E- Public Relations," tesis (skripsi), Jurusan Ilmu Komunikasi, Universitas Sebelas Maret, Surakarta, Indonesia, 2017.

[11] R. Briandana, T. Lestari, R. F. Martha. (Desember 2020). Efektivitas Iklan Melalui SMS Blast Terhadap Keputusan Pembelian Konsumen. *Jurnal Lensa Mutiara Komunikasi*. [Online]. 4(2), hal. 98-112. Tersedia: http://e-journal.sari-mutiara.ac.id/index.php/JLMI/article/view/1454

[12] A. N. Rais, dkk. (Maret 2022). Implementasi Sistem Informasi Food and Beverage Online Shop Dengan Metode Waterfall Yang Dimodifikasi. *Evolusi: Jurnal Sains dan Manajemen*. [Online]. 10(1), hal. 58-65. Tersedia: https://doi.org/10.31294/evolusi.v10i1.12053

[13] H. A. Wicaksono, N. Setiyawati. (April 2022). Pembangunan Python Script Generator Pada Pengembangan Aplikasi Berbasis Web. *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*. [Online]. 5(1), hal. 157-166. Tersedia: https://doi.org/10.37792/jukanti.v5i1.472

[14] C. Wijayanto, Y. A. Susetyo. (September 2022). Implementasi Flask Framework pada Pembangunan Aplikasi Sistem Informasi Helpdesk (SIH). *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*. [Online]. 7(3), hal. 858-868. Tersedia: https://doi.org/10.29100/jipi.v7i3.3161.g1328

[15] I. A. O. Ahmed, M. E. E. Daleel. (Oktober 2020). Automated Use Case Diagram Generation with Non-functional Requirements using Neural Network. *International Journal of Applied Information Systems*. [Online]. 12(34), hal. 1-4. Tersedia: https://api.semanticscholar.org/CorpusID:225061229

[16] G. Schumy, Y. A. Susetyo. (Agustus 2022). Rancang Bangun Sistem Sinkronisasi Data Menggunakan Google Cloud Pub/Sub dan Flask Di PT XYZ. *Jurnal MNEMONIC*. [Onilne]. 4(2), hal. 86-92. Tersedia: https://doi.org/10.36040/mnemonic.v5i2.4645