

**IMPLEMENTASI SIMPLE MAIL TRANSFER PROTOCOL
DENGAN METODE PUBLISH SUBSCRIBE**



IF 40-04

Hovely Wahyu Z. Simatupang 1301164167

Muhammad Ryaan Izza M 1301164428

Deryandaru Bhaskara 1301160239

Aup Hakim Nurzaman 1301164395

**Fakultas Informatika
Universitas Telkom
Bandung
2019**

Abstrak

Simple Mail Transfer Protocol adalah sebuah protokol untuk mengirim pesan *e-mail* kepada penerima agar tepat waktu dan efisien. Jika melihat pada OSI Model (Open Systems Interconnection), SMTP berada pada layer ke-7, yaitu Application Layer. Tujuan dari implementasi ini adalah untuk mengetahui cara kerja dari protokol pengirim pesan *e-mail* yang menggunakan metode *publish-subscribe*. Pada percobaan pengiriman pesan, kami menggunakan *e-mail* yang telah dibuat dengan domain gmail dan akan mengirimkan pesan yang ada pada *body e-mail* dan dengan tambahan *file* bertipe *text*.

Kata kunci: *e-mail*, SMTP, OSI Model, *publish-subscribe*

1. Pendahuluan

Pengiriman email saat ini dapat dilakukan dengan berbagai cara, yaitu dengan mengakses platform/aplikasi penyedia jasa pengiriman email, contohnya gmail, microsoft.outlook, yahoo, dan lain-lain. Studi kasus yang akan dipecahkan adalah bagaimana user dapat mengirimkan email dengan attachment tanpa harus menggunakan platform atau browser. Yaitu dengan menggunakan metode mail server dengan bantuan protokol Simple Mail Transfer Protocol (SMTP). Simple Mail Transfer Protocol (SMTP) digunakan untuk berkomunikasi dengan remote server untuk mengirimkan email dari local ke remote server dan tentu saja ke server penerima email. Proses ini dikendalikan oleh Mail Transfer Agent (MTA) di server email sender.

Mail Server merupakan sebuah server yang dimana berfungsi untuk mengirim dan menerima pesan. Agar pesan bisa dikirim kepada penerima diperlukan sebuah *server* sebagai media penghubung antara penerima dan pengirim. Tujuan dari tugas ini adalah implementasi Mail Server.

2. Pembahasan

Pada tugas ini kami menerapkan proses pengiriman pesan pada email dengan menggunakan SMTP(Simple Mail Transfer Protocol). Proses ini memiliki prinsip Indirect Communication dengan metode Publish-Subscribe, yaitu sender tidak langsung terhubung dengan receiver dalam pengiriman pesan dan pesan yang dikirim tidak diprogram untuk dapat langsung diterima oleh receiver. Jenis coupling pada proses ini adalah Space Coupling dan Time Uncoupling. Space Coupling karena sender memerlukan identitas receiver dalam mengirimkan pesan seperti email dan nomor port. Time Uncoupling karena sender dapat mengirimkan pesan kapan saja tanpa perlu mengetahui receiver sedang dalam keadaan aktif atau tidak. Protokol ini juga memiliki sistem keamanannya sendiri, yaitu teknik keamanan jaringan TLS(Transport Layer Security). Teknik keamanan ini berfungsi untuk mengamankan jalur pengiriman pesan kepada penerima agar tidak ada pihak ketiga yang bisa interupsi atau mengambil data yang sedang dikirimkan. Pada data pesan yang akan dikirimkan, kami menggunakan fungsi library yang sudah ada, yaitu MIME(Multipurpose Internet Mail Extensions). Contohnya pada *body e-mail*, kami memasukkan pesan singkat untuk penerima. Agar pesan tersebut dapat ditampilkan dan dibaca pada aplikasi *e-mail*, kami menggunakan MIMEText.

2.1. Kelebihan dan Kekurangan SMTP

Adapun kelebihan dari SMTP adalah :

1. Proses berkirim email pun dapat dilakukan dengan sangat cepat ke seluruh dunia sehingga dapat menghemat uang dan waktu dalam berkirim surat.
2. Lebih simple dan lebih cepat.

Sedangkan kekurangannya antara lain :

1. Jika jaringan antara komputer dan outgoing SMTP yang digunakan atau antara SMTP dengan mail server yang dituju terganggu dapat menyebabkan kegagalan pengiriman email.
2. Keterlambatan tibanya email kepada email tujuan, tergantung pada SMTP dan mail-server yang dituju, yaitu seberapa banyak antrian email yang masuk ke SMTP yang digunakan dan seberapa banyak antrian email yang akan diterima oleh email tujuan.

3. Beberapa ISP(Internet Service Provider) mewajibkan SMTP melakukan authentication, sehingga sebelum menggunakan E-mail sebagai media pengiriman, authentication perlu untuk diaktifkan.

2.2. Strategi Penyelesaian

1. Melakukan import library yang dibutuhkan seperti smtplib, MIMEText, MIMEMultipart, MIMEBase, dan encoders.

```
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders
```

2. Mendefinisikan variabel email pengirim, password email pengirim, dan email penerima.

```
#untuk login dan mendefinisikan email receiver
email_sender = 'tubessister19@gmail.com'
email_password = 'sister2019'
email_receiver = 'dbhaskara28@gmail.com'
```

3. Memanggil fungsi MIMEMultipart() untuk membuat variabel msg memiliki lebih dari satu bagian yang digunakan dan MIMEText untuk menunjukkan bahwa yang digunakan pada isi dari body berupa text.

```

#untuk komponen yang ada pada email, seperti subject, body dan file attachment
subject = 'Tugas Besar Sistem Terdistribusi dan Sistem Paralel'

msg = MIMEMultipart()
msg['From'] = email_sender
msg['To'] = email_receiver
msg['Subject'] = subject

body = 'Hallo, hasil run simple code tubes SISTER'
msg.attach(MIMEText(body, 'plain'))

filename = 'test.txt'
attachment = open(filename, 'rb')

```

4. Melakukan encode pada file attachment yang dikirim dengan encode_base64(part) dan mendefinisikannya dalam bentuk string oleh fungsi as_string().

```

#melakukan encode pada file attachment yang dikirimkan
part = MIMEBase('application', 'octet-stream')
part.set_payload((attachment).read())
encoders.encode_base64(part)
part.add_header('Content-Disposition', "attachment; filename= " + filename)

msg.attach(part)
text = msg.as_string()

```

5. Melakukan proses pengiriman email dengan menggunakan port 587 dan memanggil fungsi untuk keamanan pada Transport Layer, yaitu `tls(transport layer security)` dengan menggunakan `server.starttls()`.

```
#melakukan proses pengiriman email
try:
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login(email_sender, email_password)

    server.sendmail(email_sender, email_receiver, text)

    print("BERHASIL")
    server.quit()
except:
    print("GAGAL")
```

2.3. Screenshot output Program

```
===== RESTART: C:\Users\Johnny\Documents\IMAPKITA\SMTP.py =====
type your email : tubessister19@gmail.com
type your password : sister2019
LOGIN BERHASIL
SEND TO :|

SEND TO :ryaanizzam@gmail.com
FILENAME :tes.txt
SUBJECT :Coba Kirim
TEXT BODY :Mengirim|

type your email : tubessister19@gmail.com
type your password : sister2019
LOGIN BERHASIL
SEND TO :ryaanizzam@gmail.com
FILENAME :tes.txt
SUBJECT :Coba Kirim
TEXT BODY :Mengirim
BERHASIL
>>> |
```