

# Penerapan *MAC Address* sebagai Autentikasi Aplikasi menggunakan *JavascriptBindings* Chromium Embedded Framework Python di PT XYZ

Audric Tjhoernandes<sup>1</sup>, Yerymia Alfa Susetyo<sup>2</sup>  
Universitas Kristen Satya Wacana, Jl. Diponegoro 52-60, Salatiga<sup>1,2</sup>  
Email: 672018407@student.uksw.edu<sup>1</sup>, yeremia.alfa@uksw.edu<sup>2</sup>

**Abstrack** - The development of the world of information technology that is growing cannot be separated from an application authentication development. By following the business development of PT XYZ, the application authentication system is prioritized to be developed. By using the MAC address, the authentication system is not restricted only with username and password, but also filtering some certain hardware. The problem in JavaScript is cannot get the hardware's MAC address directly. This research was built system that can get the MAC address using *JavascriptBindings* function in Chromium Embedded Framework (CEF) by using Python programming language. Using CEF Python, the MAC address can be gotten and then being integrated to the JavaScript with *Javascriptbindings* function. Based on the result of the research, it can be said that the implemented system can help PT XYZ to improve the application authentication system.

**Keywords** - Chromium Embedded Framework (CEF), *JavascriptBindings*, Python, MAC address, Authentication.

**Intisari** - Perkembangan dunia teknologi informasi yang semakin berkembang tidak lepas dari peningkatan sistem keamanan aplikasi. Dengan semakin berkembangnya bisnis PT XYZ, sistem autentikasi aplikasi menjadi prioritas utama untuk dikembangkan. Dengan menggunakan *MAC address*, sistem autentikasi tidak hanya terbatas pada *username* dan *password*, tetapi juga membatasi perangkat tertentu. Permasalahan pada JavaScript adalah tidak dapat memperoleh *MAC address* perangkat secara langsung. Penelitian ini membangun sistem yang dapat memperoleh *MAC address* menggunakan fungsi *JavascriptBindings* yang terdapat pada *Chromium Embedded Framework* (CEF) dengan menggunakan bahasa pemrograman Python. Penggunaan CEF Python, *MAC address* dapat diperoleh kemudian diintegrasikan ke dalam JavaScript menggunakan fungsi *JavascriptBindings*. Berdasarkan hasil penelitian dapat disimpulkan bahwa implementasi sistem dapat membantu PT XYZ dalam meningkatkan sistem autentikasi aplikasi.

**Kata Kunci** - Chromium Embedded Framework (CEF), *JavascriptBindings*, Python, MAC address, Autentikasi.

## I. PENDAHULUAN

Berdasarkan fenomena yang terjadi saat ini, aplikasi berbasis web sangat banyak digunakan. Terlihat dari jumlah *website* yang ada di dunia telah mencapai 1,9 miliar dan masih akan terus meningkat seiring berjalannya waktu[1]. *JavaScript* merupakan salah satu bahasa pemrograman yang paling populer di dunia sehingga sebagian besar pembuatan web saat ini menggunakan *JavaScript* [2]. Selain itu, penggunaan aplikasi berbasis sangat banyak digunakan oleh berbagai perusahaan atau pelaku bisnis lainnya. Namun, para pelaku bisnis tersebut tidak jarang mengalami “serangan” dari internet yang mengakibatkan adanya kerusakan pada sistem, atau bisa saja dapat mengalami kebocoran data [3]. Salah satu cara

untuk mencegah terjadinya hal tersebut, yaitu meningkatkan sistem keamanan, salah satunya pada sistem autentikasi.

Aplikasi pada umumnya menerapkan sistem autentikasi untuk memverifikasi apakah pengguna memiliki akses terhadap aplikasi tersebut atau tidak. Sistem autentikasi web umumnya hanya menggunakan *username* dan *password* pengguna. Namun untuk aplikasi web internal perusahaan, diperlukan pembatasan atas perangkat yang diperbolehkan akses aplikasi tersebut. Untuk itu, dibutuhkan peningkatan sistem autentikasi untuk membatasi perangkat tertentu yang memiliki akses. Salah satu cara membatasi akses adalah autentikasi menggunakan *MAC address*[4]. Akan tetapi, *JavaScript* di browser tidak diperbolehkan mengakses *hardware* secara langsung. Oleh karena itu, dibutuhkan cara lain untuk mengakses *MAC address*.

*Chromium Embedded Framework Python* (CEF Python) merupakan sebuah browser *framework* pada *Python*. Dengan menggunakan CEF Python, aplikasi web berbasis *JavaScript* dapat ditampilkan selayaknya di dalam *browser*[5]. Terdapat juga fungsi *JavascriptBindings*, yang berguna untuk mengintegrasikan *JavaScript* dengan *code* pada *Python*. Dengan menggunakan *JavascriptBindings*, variabel atau objek yang ada pada *Python* dapat digunakan pada *JavaScript* juga. Maka, *JavaScript* dapat memperoleh *MAC address* dari *JavascriptBindings* pada CEF Python[6].

Atas penerapan teknologi tersebut, pengecekan *MAC address* untuk sistem autentikasi dapat digunakan untuk meningkatkan keamanan sistem.

## II. SIGNIFIKANSI STUDI

### A. Studi Literatur

Penelitian berjudul “*Applying MAC Address-Based Access Control for Securing Admin’s Login Page*” membahas penerapan kontrol akses berbasis *MAC address* untuk mengamankan halaman *login* admin. Penelitian ini menuliskan bahwa autentikasi pada halaman admin dapat lebih aman tidak hanya menggunakan *username* dan *password*, tetapi juga menggunakan *MAC address*. Dengan adanya autentikasi menggunakan *MAC address*, akses *login* pada web tidak hanya membatasi pengguna tertentu saja, tetapi juga membatasi perangkat tertentu sehingga keamanannya lebih tinggi[4]. Penelitian tersebut menggunakan bahasa pemrograman PHP dalam penerapannya, sementara itu, dalam penelitian ini dikembangkan aplikasi yang berbasis web menggunakan bahasa pemrograman *Python* yang diintegrasikan dengan *JavaScript*.

Penelitian lain yang berjudul “*Creating User Interfaces Using Web-based Technologies to Support Rapid Prototyping in a Desktop Astrovisualization Software*” membahas tentang implementasi CEF dalam mengembangkan *Graphical User Interface* (GUI) pada aplikasi *desktop*. Penelitian tersebut menuliskan bahwa CEF dapat berjalan dan melakukan proses *web rendering* dengan performa yang baik. Aplikasi yang dibuat berupa aplikasi *desktop*, tetapi dapat menampilkan tampilan web dengan adanya CEF yang diterapkan sehingga dapat membuat sebuah aplikasi *desktop* dapat digunakan layaknya sebuah web[5]. Penelitian tersebut mengimplementasikan CEF menggunakan bahasa pemrograman C++, oleh karena itu penulis ingin mengimplementasikan CEF menggunakan bahasa pemrograman *Python*.

Sementara itu dalam penelitian mengenai “*Enhanced Security Framework On Chatbot Using MAC Address Authentication To Customer Service Quality*” membahas mengenai peningkatan keamanan *framework* pada *chatbot* menggunakan autentikasi *MAC address*. Penelitian tersebut menjelaskan autentikasi menggunakan *MAC address* bertujuan agar *chatbot* hanya dapat diakses oleh perangkat *customer* yang terdaftar. Autentikasi *MAC address* diharapkan menjadi tahap pertama yang diautentikasi oleh sistem, lalu setelah itu autentikasi berupa *username*, *password*, dan *email verification*[7]. Sistem autentikasi menggunakan *MAC address* pada penelitian tersebut dapat meningkatkan keamanan *framework* sehingga konsep tersebut dapat dijadikan acuan yang baik dalam autentikasi suatu aplikasi.

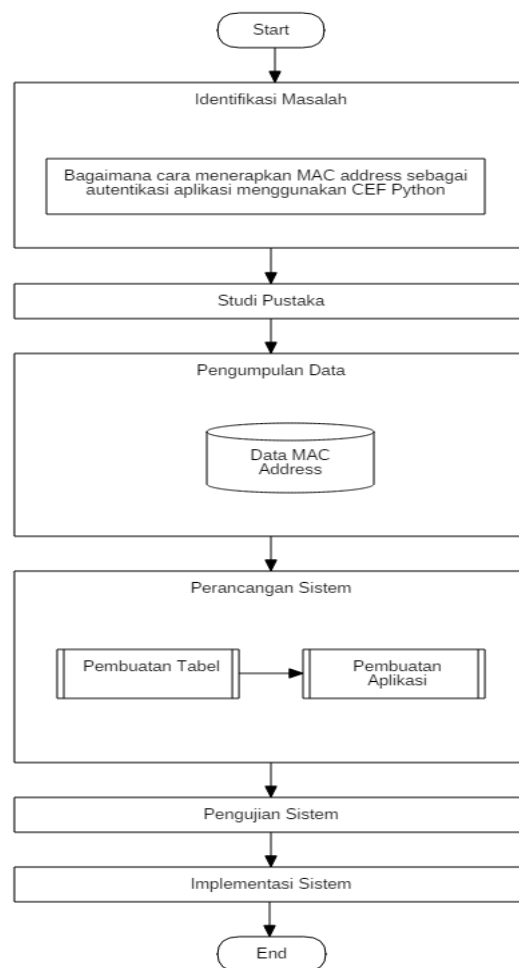
*Python* merupakan bahasa pemrograman interpretatif yang dianggap mudah untuk dipelajari karena memiliki kode-kode pemrograman yang sangat jelas dan lengkap[8]. *Python* juga disebut sebagai bahasa pemrograman tingkat tinggi karena perintah atau kode program yang digunakan telah mirip dengan bahasa manusia[9].

*Chromium Embedded Framework (CEF)* adalah sebuah *framework* sederhana yang dibuat untuk penyematan peramban berbasis Chromium di dalam aplikasi lainnya dan dapat menampilkan *embedded browser*[10]. CEF Python adalah *project* yang bersifat *open-source* yang dibuat oleh Czarek Tomczak pada tahun 2012 untuk menyediakan CEF yang dapat berjalan pada bahasa pemrograman *Python*[11].

*JavaScript* merupakan bahasa pemrograman yang dapat membuat sebuah web menjadi dinamis dan juga dapat secara aktif berkomunikasi dengan pengguna[12]. *JavaScript* dapat dituliskan di dokumen HTML atau dapat juga ditulis di dalam dokumen tersendiri, kemudian diasosiasikan dengan dokumen lain yang dituju[13].

*Media Access Control Address (MAC Address)* merupakan sebuah alamat jaringan yang diimplementasikan pada lapisan *data-link* dalam tujuh *OSI layer*, yang merepresentasikan sebuah *node* tertentu pada jaringan. *MAC Address* merupakan alamat unik yang mengidentifikasi sebuah komputer, *interface* dalam sebuah *router*, atau *node* lainnya dalam jaringan. *MAC Address* juga biasanya disebut dengan *ethernet address*, *physical address*, atau *hardware address*[14].

**B. Metode Penelitian**



Gambar 1. Flowchart Tahap Penelitian

Pada Gambar 1 dijelaskan proses yang dilakukan dalam penelitian, dimulai dari identifikasi masalah hingga implementasi sistem. Tahapan proses penelitian tersebut adalah sebagai berikut:

1. *Identifikasi masalah*

Pada tahap pertama permasalahan yang ada merupakan kebutuhan PT. XYZ untuk meningkatkan sistem autentikasi dengan cara menerapkan MAC *address* sebagai alat autentikasi aplikasi berbasis web menggunakan Python dan JavaScript.

2. *Studi pustaka*

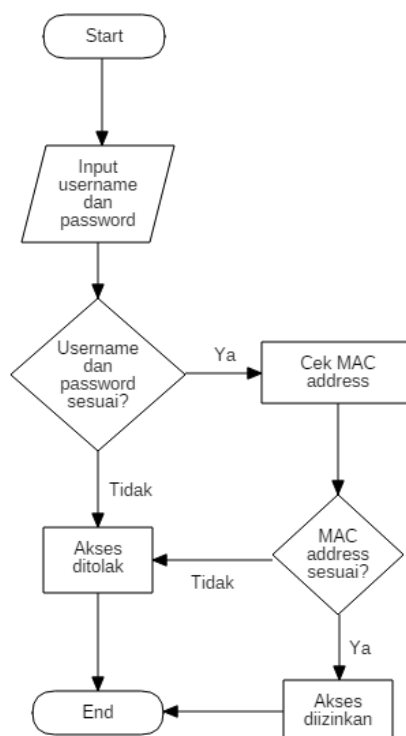
Setelah mengidentifikasi masalah, solusi yang ditemukan adalah dengan menggunakan *Chromium Embedded Framework* (CEF) Python. Setelah itu, dilakukan studi pustaka untuk memahami cara kerja CEF Python.

3. *Pengumpulan data*

Setelah melakukan studi pustaka, selanjutnya adalah mengumpulkan data, yaitu MAC *address* perangkat sebagai *sample* pengujian autentikasi, lalu dimasukkan ke dalam tabel pada *database*.

4. *Perancangan sistem*

Setelah mendapatkan informasi mengenai CEF Python dan data yang dikumpulkan, maka dilakukan perancangan sistem, dimulai dari membuat *table* di *database* menggunakan SQLite, kemudian merancang program menggunakan CEF Python dan JavaScript untuk menghasilkan sistem autentikasi aplikasi berbasis web. Dengan menggunakan CEF Python, manfaat yang diperoleh adalah dapat mengintegrasikan *script* Python dan JavaScript dalam satu *file*, termasuk variabel yang terdapat pada Python juga dapat digunakan pada JavaScript.



Gambar 2. Flowchart Sistem Autentikasi

Pada Gambar 2 dijelaskan alur kerja sistem autentikasi yang dihasilkan. Dimulai pada tahap pertama, yaitu *user* memasukkan *username* berupa NIK dan *password* berupa PIN sebagai autentikasi tahap awal, hingga tahap akhir, yaitu *user* memiliki akses atau tidak.

5. *Pengujian sistem*

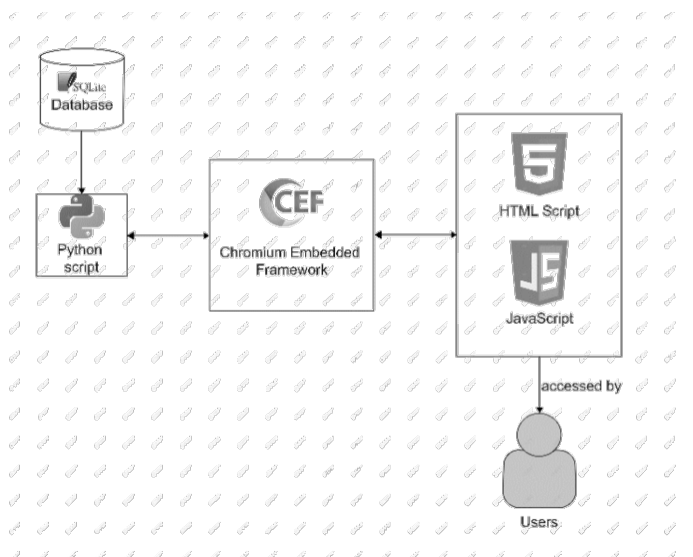
Setelah sistem yang ada selesai dirancang, maka dilakukan proses pengujian apakah sistem dapat berjalan dan berfungsi sesuai dengan yang diharapkan. Metode pengujian yang digunakan adalah metode *black box* dan setelah itu dilakukan penyebaran kuesioner sebagai pengujian lapangan.

6. *Implementasi sistem*

Setelah dipastikan bahwa sistem dapat berfungsi sesuai dengan semestinya, sistem yang dibuat akan diimplementasikan ke dalam aplikasi pada perusahaan PT XYZ.

**III. HASIL DAN PEMBAHASAN**

Penelitian ini menghasilkan sistem autentikasi aplikasi berbasis web yang dilakukan dengan cara melakukan *filtering MAC address* perangkat yang dapat mengakses aplikasi. Sistem yang dihasilkan dapat berjalan pada perangkat yang memenuhi *requirements*, yaitu Python 3.7 dan *Chromium Embedded Framework* (CEF) versi 57 ke atas.



Gambar 3 Arsitektur Sistem

Pada gambar 3 yang merupakan arsitektur sistem, proses *back-end* sistem dirancang menggunakan Python dan juga mengambil data dari SQLite. Setelah itu, data yang diperoleh diintegrasikan ke dalam JavaScript dan tampilan *front-end* HTML menggunakan *Chromium Embedded Framework* (CEF) sebagai ‘jembatan’.

Kode1 program awal

```
import sqlite3
from cefpython3 import cefpython as cef
from getmac import get_mac_address as mac_address
import platform
import base64
import sys

HTML_code = """
<!DOCTYPE html>
<html>
<head>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
  <link href="signin.css" rel="stylesheet">
</head>
<style>
```

```

        .container{
            margin-top: 100px
        }
    </style>
</body>

```

Kode program 1 merupakan kode awal pada sistem. Pada baris 1-6 merupakan *library/package* yang dibutuhkan dalam menjalankan sistem autentikasi. Baris ke-1 merupakan proses *import* modul SQLite yang menjadi *database*. Setelah itu di baris ke-2 merupakan proses *import* modul CEF Python. Pada baris ke-3 terdapat *import get\_mac\_address* yang merupakan fungsi dari *package getmac*. **Getmac** merupakan salah satu *package* Python yang digunakan untuk “mengambil” *MAC address* perangkat menggunakan Python[15].

Pada baris ke-8 merupakan proses *assign* HTML *script* dan JavaScript ke dalam variabel Python, yang nantinya akan menjadi tampilan yang diproses pada CEF.

#### Kode Program 1 Query select data

```

def select_data(self):
    mac = []
    self.__c.execute('SELECT * FROM tb_mac')
    data = self.__c.fetchall()
    for ma in data:
        mac.append(ma[0])
    return mac

```

Pada kode program 2 merupakan fungsi memperoleh data *MAC address* dari *database*. Hasil *return* dari fungsi tersebut akan digunakan sebagai daftar *MAC address* yang terautentikasi.

#### Kode Program 3 Kode JavascriptBindings

```

def set_javascript_bindings(self, browser):
    macaddress = mac_address()
    bindings = cef.JavascriptBindings()
    bindings.SetProperty("macaddress", macaddress)
    bindings.SetProperty("list_mac", self.list_mac)
    browser.SetJavascriptBindings(bindings)

```

Kode program 3 adalah fungsi menghubungkan *script* Python dengan JavaScript. Pada baris ke-2 merupakan *assign* dari **mac\_address** ke dalam variabel **macaddress** yang didapatkan dari hasil *import* di kode program 1. Kemudian di baris yang ke-3 merupakan *assign* dari *method JavascriptBindings* ke dalam variabel **bindings**. **JavascriptBindings** merupakan salah satu *class* dari CEF Python yang berfungsi untuk mengintegrasikan Python dan JavaScript. Dengan menggunakan **JavascriptBindings**, *exceptions* pada JavaScript dapat ditampilkan ke dalam *exceptions* Python ketika menggunakan JavaScript atau Python *callbacks*.

Setelah itu, pada baris ke-4 dan ke-5 merupakan penggunaan fungsi **SetProperty** dari *class JavascriptBindings* yang bertujuan untuk mengatur variabel dalam Python sebagai variabel yang nantinya juga dapat digunakan dalam JavaScript, dalam hal ini adalah variabel **macaddress** dan **list\_mac**.

#### Kode Program 4 Fungsi login pada JavaScript

```

function login(ma) {
    var nik = document.getElementById("nik").value
    var pin = document.getElementById("pin").value
    let login_status = false
    if (nik == "" || pin == ""){
        alert("Username and Password must be filled!")
    }else{

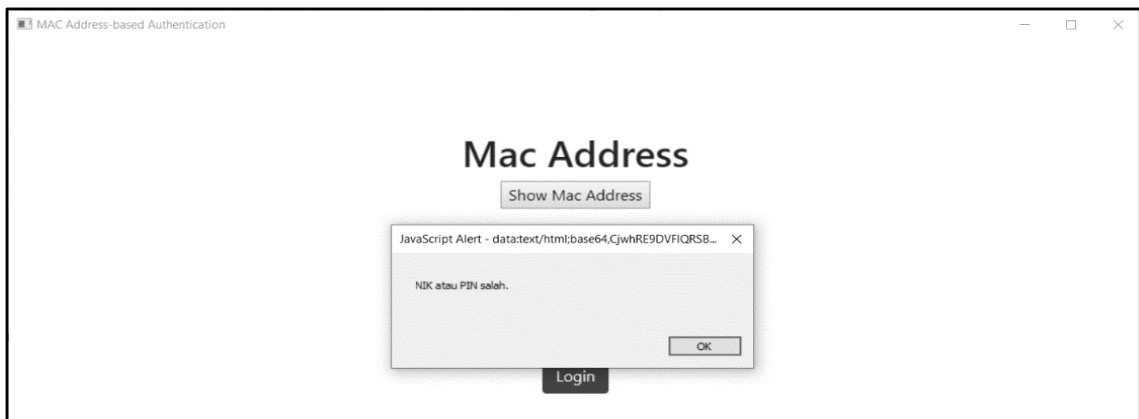
```

```

if (nik == "123456" && pin == "123456"){
  for(let i = 0; i < list_mac.length; i++){
    if (ma == list_mac[i]){
      login_status = true
    }
  }
  if (login_status == true){
    alert('Login Successfully!')
  }else{
    alert("You don't have a permission!")
  }
}else{
  alert("NIK atau PIN salah.")
}
}
}

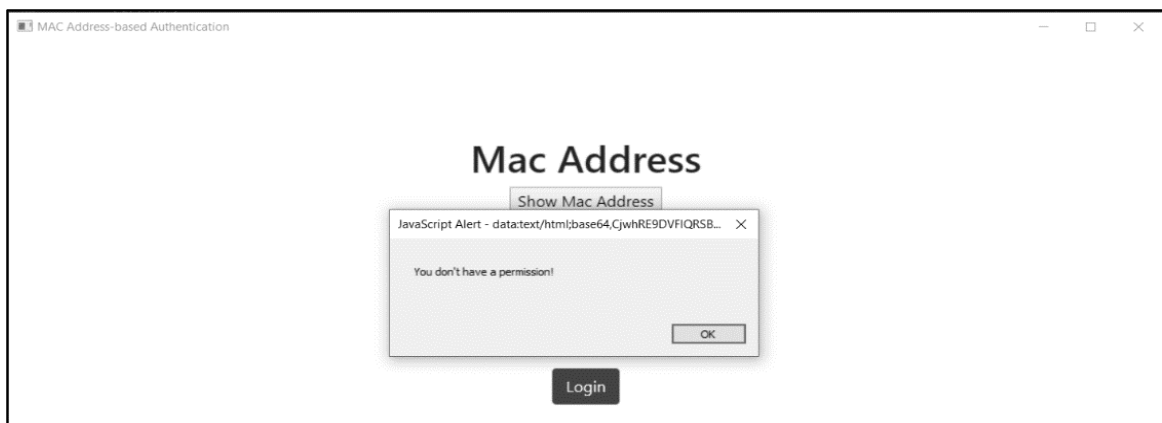
```

Pada kode program 4 merupakan fungsi login pada JavaScript yang telah mengimplementasikan sistem autentikasi menggunakan *MAC address*.



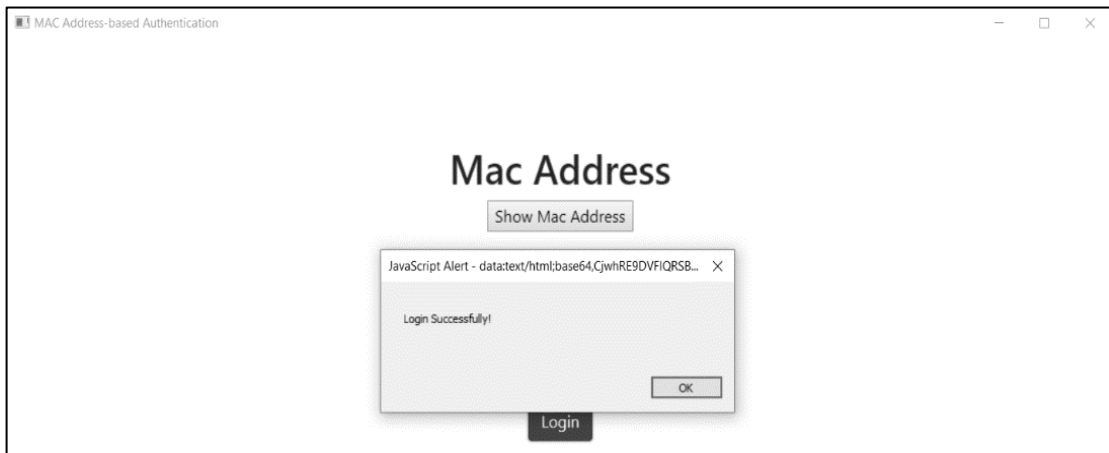
Gambar 4 Alert Username dan Password Salah

Pada gambar 4 merupakan tampilan jika pengguna memasukkan NIK dan atau PIN yang salah.



Gambar 5 Alert User Tidak Memiliki Akses

Gambar 5 merupakan tampilan ketika *MAC address* tidak terdaftar meskipun *user* memasukkan NIK dan PIN yang benar.



Gambar 6 Login Sukses

Jika *user* telah diautentikasi dan memiliki akses, maka *user* akan diberikan notifikasi berupa *alert* untuk memberitahu bahwa login sukses, seperti yang terlihat pada gambar 6.

Untuk memastikan sistem berjalan sesuai dengan semestinya, maka dilakukan pengujian sebelum pengimplementasian. Pengujian dilakukan dengan menggunakan metode *Black Box Testing*. *Black Box Testing* merupakan teknik pengujian *software* yang berfokus pada fungsi-fungsi yang ada pada *software* tersebut[16].

TABEL I  
PENGUJIAN SISTEM AUTENTIKASI

Pengujian	Yang Diharapkan	Pengamatan	Kesimpulan
NIK dan PIN yang dimasukkan salah, MAC <i>address</i> tidak terdaftar	Perangkat tidak diberikan akses untuk login	Perangkat tidak diberikan akses untuk login	[V] Diterima [ ] Ditolak
NIK dan PIN yang dimasukkan benar, MAC <i>address</i> tidak terdaftar	Perangkat tidak diberikan akses untuk login	Perangkat tidak diberikan akses untuk login	[V] Diterima [ ] Ditolak
NIK dan PIN yang dimasukkan salah, MAC <i>address</i> terdaftar	Perangkat tidak diberikan akses untuk login	Perangkat tidak diberikan akses untuk login	[V] Diterima [ ] Ditolak
NIK dan PIN yang dimasukkan benar, MAC <i>address</i> terdaftar	Perangkat diberikan akses untuk login	Perangkat diberikan akses untuk login	[V] Diterima [ ] Ditolak

Tabel I merupakan hasil pengujian sistem autentikasi dan dari hasil yang didapatkan menunjukkan bahwa perangkat yang memiliki akses untuk login adalah hanya perangkat yang memasukkan NIK dan PIN yang benar dan juga memiliki MAC *address* terdaftar. Ketika NIK dan PIN yang dimasukkan benar namun MAC *address* tidak terdaftar, maka perangkat tetap tidak akan diberikan akses untuk login.

Setelah itu, dilakukan penyebaran kuesioner dengan melibatkan 15 orang internal divisi IT PT. XYZ untuk mengetahui apakah sistem autentikasi menggunakan MAC *address* dapat menjadi sistem autentikasi yang efektif.



TABEL II  
TABEL KUESIONER

Kode Pertanyaan	Pertanyaan	Jawaban					Jumlah
		A	B	C	D	E	
1	Menurut anda, apakah autentikasi menggunakan MAC address dapat membantu meningkatkan keamanan autentikasi aplikasi pada perusahaan PT. XYZ?	-	-	-	9	6	66
2	Menurut anda, apakah sistem autentikasi menggunakan MAC address dapat mencegah perangkat yang tidak dikenal untuk mengakses aplikasi?	-	-	-	6	9	69
3	Menurut anda, apakah sistem autentikasi menggunakan MAC address lebih efektif diterapkan dibandingkan hanya menggunakan username dan password?	-	-	-	6	9	69

Pada tabel II terdapat kolom jumlah yang diperoleh berdasarkan rumus perhitungan skala likert pada rumus perhitungan (1) dan juga nilai dari skala likert pada tabel III.

$$\sum(T \times Pn) \tag{1}$$

Keterangan:

T = Total responden

Pn = Skor likert

TABEL III  
KETERANGAN NILAI SKALA LIKERT

Kode Jawaban	Deskripsi Jawaban	Skor
A	Sangat Tidak Setuju	1
B	Tidak Setuju	2
C	Cukup	3
D	Setuju	4
E	Sangat Setuju	5

Berdasarkan skor yang didapatkan, ditentukan hasil interpretasi Y diperoleh dari skor tertinggi dikalikan dengan jumlah responden dan diperoleh hasil  $5 \times 15 = 75$ . Setelah itu, dilakukan perhitungan persentase perolehan skor tertinggi pada kuesioner seperti pada tabel IV.

TABEL IV  
PERSENTASE SKOR TERTINGGI

Nomor Pertanyaan	$(\text{Jumlah}/Y) \times 100\%$
1	$(66/75) \times 100\% = 88\%$
2	$(69/75) \times 100\% = 92\%$
3	$(69/75) \times 100\% = 92\%$

TABEL V  
INDEX RANGE SKALA LIKERT

Index Range	Keterangan
0% - 19,99%	Sangat Tidak Setuju
20% - 39,99%	Tidak Setuju
40% - 59,99%	Kurang Setuju
60% - 79,99%	Setuju
80 - 100%	Sangat Setuju

Setelah mendapatkan hasil persentase pada tabel IV, maka dilakukan perhitungan rata-rata persentase dan diperoleh hasil **90,67%**. Berdasarkan pada tabel V, hasil yang diperoleh berada pada index **Sangat Setuju**, oleh karena itu dapat disimpulkan sistem autentikasi menggunakan MAC address dapat bekerja dengan efektif dalam meningkatkan keamanan sistem autentikasi aplikasi.

#### IV. KESIMPULAN

Pada penelitian ini dilakukan pengimplementasian MAC address sebagai autentikasi aplikasi menggunakan *Chromium Embedded Framework* (CEF) Python. Berdasarkan hasil dan pembahasan yang ada, dilakukan proses pengujian menggunakan metode *black box* dan juga penyebaran kuesioner. Dari hasil kuesioner tersebut, berdasarkan skala likert diperoleh hasil persentase sebesar **90,67%**, sehingga dapat disimpulkan bahwa sistem autentikasi menggunakan MAC address dapat dengan efektif melakukan *filtering* terhadap perangkat yang tidak dikenal atau tidak memiliki akses terhadap aplikasi.

Pada penelitian ini masih terdapat kekurangan berupa sistem autentikasi yang hanya dapat berjalan pada perangkat yang memiliki Python. Sebenarnya hal tersebut dapat menjadi autentikasi tambahan agar perangkat yang ada lebih tersaring, namun hal tersebut juga berarti sistem autentikasi ini tidak dapat berjalan secara universal pada setiap perangkat. Oleh karena itu, untuk pengembangan penelitian selanjutnya disarankan untuk menghasilkan sistem yang dapat berjalan pada berbagai perangkat secara universal.

#### REFERENSI

- [1] I. L. Stats, "Internet Live Stats," *Internet Live Stats*, 2020. <https://www.internetlivestats.com/total-number-of-websites/> (accessed Oct. 28, 2021).
- [2] Tiobe, "Tiobe," *Tiobe*, 2021. <https://www.tiobe.com/tiobe-index/> (accessed Oct. 26, 2021).
- [3] A. Tedyyana and O. Ghazali, "Teler real-time http intrusion detection at website with nginx web server," *Int. J. Informatics Vis.*, vol. 5, no. 3, pp. 327–332, 2021, doi: 10.30630/joiv.5.3.510.
- [4] B. M. P. P. Alam, R. Septiasari, and A. Amiruddin, "Applying MAC Address-Based Access Control for Securing Admin's Login Page," *Proceeding Electr. Eng. Comput. Sci. Informatics*, vol. 6, no. 1, pp. 292–295, 2019, doi: 10.11591/eecsi.v6.2005.
- [5] K. Eskilson, "Creating User Interfaces Using Web-based Technologies to Support Rapid Prototyping in a Desktop Astrovisualization Software Kelas Eskilson Creating User Interfaces Using Web-based Technologies to Support Rapid Prototyping in a Desktop Astrovisualization So," Linköping University, Stockholm, 2017.
- [6] C. Tomczak, "CEF Python," *Github*, 2012. <https://github.com/cztomczak/cefpython/>

- blob/master/api/ JavascriptBindings.md (accessed Oct. 28, 2021).
- [7] R. Hardi, A. Naim Che Pee, and N. Suryana Herman, "Enhanced Security Framework On Chatbot Using Mac Address Authentication To Customer Service Quality," *Int. J. Sci. Technol. Res.*, vol. 9, no. 10, 2020, [Online]. Available: [www.ijstr.org](http://www.ijstr.org).
- [8] J. Enterprise, *Otodidak Pemrograman Python*. Jakarta: Elex Media Komputindo, 2017.
- [9] A. E. Falentino Sembiring, *Bahasa Ular Untuk Pemrograman Python*. Solok: Insan Cendekia Mandiri, 2020.
- [10] M. Greenblatt, "Chromium Embedded Framework," *Bitbucket*, 2008. <https://bitbucket.org/chromiumembedded/cef/src/master/> (accessed Oct. 26, 2021).
- [11] C. Tomczak, "cefpython," *Github*, 2012. <https://github.com/cztomczak/cefpython> (accessed Oct. 28, 2021).
- [12] S. Delcev and D. Draskovic, "Modern JavaScript frameworks: A Survey Study," in *2018 Zooming Innovation in Consumer Technologies Conference, ZINC 2018*, 2018, pp. 106–109, doi: 10.1109/ZINC.2018.8448444.
- [13] O. Pahlevi, A. Mulyani, and M. Khoir, "Sistem informasi inventori barang menggunakan metode object oriented di pt. Livaza teknologi indonesia jakarta," *Pt. Livaza Teknol. Indones. Jakarta*, vol. 5, no. 1, 2018. Pahlevi, O., Mulyani, A., Khoir, M. (2018). Sistem Informasi Inventori Barang Menggunakan Metode Object Oriented Di Pt. Livaza Teknologi Indonesia Jakarta. *Pt. Livaza Teknologi Indonesia Jakarta*, 5(1), 27–35., pp. 27–35, 2018, [Online]. Available: <https://e-jurnal.lppmunsera.org/index.php/PROSISKO/article/view/587>.
- [14] A. Tedyyana, "Rancang Bangun Jaringan Wireless Di Politeknik Negeri Bengkalis Menggunakan MAC Filtering," *Semin. Nas. Inov. DAN Apl. Teknol. DI Ind.*, pp. 31–36, 2016, [Online]. Available: <https://ejournal.itn.ac.id/index.php/seniati/article/view/309>.
- [15] E. N. S. Parker, "The Equine vision model: Sensing & data acquisition," *Ashesi Institutional Repos.*, 2020, [Online]. Available: <http://hdl.handle.net/20.500.11988/632>.
- [16] N. W. Rahadi and C. Vikasari, "Pengujian Software Aplikasi Perawatan Barang Milik Negara Menggunakan Metode Black Box Testing Equivalence Partitions," *Infotekmesin*, vol. 11, no. 1, pp. 57–61, 2020, doi: 10.35970/infotekmesin.v11i1.124.