SKRIPSI

PERANGKAT LUNAK LOGIN OTOMATIS UNTUK *CAPTIVE PORTAL* WI-FI

YOHANES MARIO CHANDRA

NPM: 2011730031

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2016

UNDERGRADUATE THESIS

AUTOMATED LOGIN SOFTWARE FOR WI-FI CAPTIVE PORTAL

YOHANES MARIO CHANDRA

NPM: 2011730031

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2016

LEMBAR PENGESAHAN

PERANGKAT LUNAK LOGIN OTOMATIS UNTUK *CAPTIVE PORTAL* WI-FI

YOHANES MARIO CHANDRA

NPM: 2011730031

Bandung, 1 Agustus 2016

Menyetujui,

Pembimbing Tunggal

Pascal Alfadian, M.Comp.

Ketua Tim Penguji	Anggota Tim Penguji
«penguji 1»	«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PERANGKAT LUNAK LOGIN OTOMATIS
UNTUK *CAPTIVE PORTAL* WI-FI

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 1 Agustus 2016

Meterai

Yohanes Mario Chandra
NPM: 2011730031

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia» Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris» Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Bandung, Agustus 2016

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Internet adalah salah satu hal yang sulit dipisahkan dari keseharian manusia masa kini. Salah satu cara seseorang dapat mengakses internet adalah dengan menggunakan teknologi Wi-Fi. Wi-Fi mengharuskan pengguna terhubung pada suatu *access point*. *Access point* tersebut dapat memiliki dua status, yaitu terproteksi atau tidak terproteksi. Proteksi pada *access point* dapat dilakukan dengan beberapa cara, yaitu menggunakan protokol IEEE 802.11, atau menggunakan *captive portal*. Alat yang sudah pernah terhubung dengan *access point* yang diproteksi dengan protokol IEEE 802.11 akan dengan mudah terhubung kembali dengan *access point* tersebut karena alat tersebut biasanya sudah menyimpan *password* untuk *access point* yang bersangkutan. Alat yang akan terhubung dengan *access point* yang diproteksi menggunakan *captive portal* belum memiliki cara untuk mengingat *username* dan *password* untuk *captive portal* tersebut sehingga login otomatis belum dapat dilakukan untuk *access point* jenis ini.

Berdasarkan pengamatan peneliti, *captive portal* banyak digunakan untuk proteksi *access point* pada tempat-tempat umum seperti lingkungan universitas, *starbucks*, *McDonald's*, dan beberapa tempat yang dapat diakses melalui *@wifi.id*, *free@wifi.id* dan *access point* sejenis. Oleh karena itu, dibutuhkan mekanisme yang bisa membantu proses login untuk *access point* tipe ini. Terdapat dua cara untuk menciptakan mekanisme ini, yaitu dengan mengintegrasikannya dengan sistem operasi, atau menggunakan perangkat lunak pihak ketiga. Untuk dapat melakukan pengintegrasian mekanisme tersebut dengan sistem operasi, dibutuhkan akses kepada kode sumber sistem aplikasi tersebut. Oleh karena itu, pilihan yang lebih bijak sebagai seseorang yang tidak memiliki akses tersebut adalah dengan menciptakan perangkat lunak pihak ketiga.

1.2 Rumusan Masalah

Rumusan masalah yang dibahas pada penelitian ini adalah sebagai berikut:

- Bagaimana caranya melakukan implementasi login otomatis pada *captive portal* yang memiliki tingkat kenyamanan yang setara dengan login otomatis pada proteksi Wi-Fi berbasis protokol IEEE 802.11?
- Apa saja yang perlu dilakukan untuk mengamankan *username* dan *password* yang disimpan oleh user?

- Informasi apa saja yang dibutuhkan untuk menciptakan identitas unik untuk setiap *captive portal* pada jaringan yang berbeda?

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

- Melakukan implementasi login otomatis pada *captive portal* yang memiliki tingkat kenyamanan yang setara dengan login otomatis pada proteksi Wi-Fi berbasis protokol IEEE 802.11.
- Memastikan *username* dan *password* pengguna disimpan secara aman?
- Menentukan informasi yang dibutuhkan untuk menciptakan identitas unik untuk setiap *captive portal* pada jaringan yang berbeda?

1.4 Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut:

- Perangkat lunak dibangun untuk sistem operasi Windows 8 sampai dengan Windows 10.
- Perangkat lunak dibangun menggunakan bahasa pemrograman C#.
- Elemen keamanan informasi yang diimplementasikan pada perangkat lunak ini adalah enkripsi *username* dan *password* yang disimpan oleh user.

1.5 Metodologi Penelitian

Metodologi penelitian yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Melakukan studi literatur mengenai hal-hal yang berkaitan dengan perancangan dan pembuatan aplikasi, yaitu:

   Cara kerja dan protokol-protokol yang terkait dengan *captive portal*.

   Pemrograman menggunakan *.NET framework*.

   Universal Windows Platform (UWP).

   Penggunaan kelas WebBrowser pada C#.

   Penggunaan objek PasswordVault pada C#.

2. Melakukan analisis perangkat lunak sejenis.
3. Melakukan analisis kebutuhan untuk mengimplementasikan mekanisme login otomatis ini.
4. Merancang perangkat lunak login otomatis ini.
5. Melakukan implementasi hasil rancangan dengan bahasa pemrograman C# pada sistem operasi Windows 10.

6. Melakukan pengujian terhadap perangkat lunak untuk menghasilkan perbaikan teradap perangkat lunak tersebut.

7. Membuat kesimpulan dari hasil penelitian dan saran untuk penelitian selanjutnya.

1.6 Sistematika Pembahasan

Laporan skripsi ini terdiri dari beberapa bab, yaitu:

1. Bab Pendahuluan
   Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian, dan sistematika pembahasan.

2. Bab Dasar Teori
   Bab ini berisi dasar-dasar teori dasar mengenai *captive portal*, *.NET framework*, *Universal Windows Platform* (UWP), dokumentasi kelas WebBrowser dan dokumentasi objek PasswordVault.

3. Bab Analisis
   Bab ini berisi analisis kebutuhan untuk perancangan dan pembuatan perangkat lunak login otomatis untuk proteksi Wi-Fi berbasis web.

4. Bab Perancangan
   Bab ini berisi perancangan perangkat lunak login otomatis untuk proteksi Wi-Fi berbasis web.

5. Bab Implementasi dan Pengujian
   Bab ini berisi implementasi perangkat lunak login otomatis untuk proteksi Wi-Fi berbasis web beserta pengujian dan hasil perbaikannya.

6. Bab Kesimpulan dan Saran
   Bab ini berisi kesimpulan dari penelitian ini dan saran yang diberikan untuk penelitian selanjutnya.

BAB 2

DASAR TEORI

Bab ini menjelaskan mengenai teori-teori yang digunakan dalam penelitian ini, antara lain penjelasan mengenai *captive portal*, *.NET framework*, *Common Laguage Infrastructure*, *Universal Windows Platform*, kelas WebView, dan kelas PasswordVault.

2.1 *Captive Portal*

Captive portal adalah *router*[1] atau *gateway*[2] yang akan menutup koneksi eksternal sampai klien yang bersangkutan sudah terotentikasi[1]. Cara kerja *captive portal* secara umum adalah sebagai berikut:

1. Memberikan alamat IP melalui DHCP pada perangkat yang baru terhubung.
2. Tutup seluruh akses kecuali ke *captive portal server*.
3. Arahkan seluruh *request* HTTP ke *captive portal*.
4. Tampilkan aturan penggunaan, informasi pembayaran, danatau halaman *login*.
5. Jika pengguna telah menyetujui aturan penggunaan atau telah melakukan *login*, buka akses.
6. Opsional: Saat pengguna telah melewati batas waktu tertentu, tutup akses.

Akan tetapi, pada prakteknya, implementasi *captive portal* sangat beragam dan bersifat *ad-hoc*[2]. Beberapa perilaku *captive portal* lain yang teramati adalah sebagai berikut:

- Memaksa pengguna untuk tetap membuka satu *browser window*. Teknik ini membantu mencegah pencurian koneksi pengguna dengan duplikasi alamat MAC.
- Menggunakan otorisasi yang terbatas oleh waktu. Pengguna harus berinteraksi kembali dengan portal setelah waktu tertentu.

Kode Status HTTP 511

Kode status HTTP adalah bilangan bulat positif yang terdiri dari 3 digit[3]. Kode status ini dikirimkan sebagai hasil dari usaha untuk memahami *request*. Kode status HTTP bersifat *extensible*. Secara garis besar, kode status HTTP dibagi menjadi 5 bagian, yaitu:

[1] Alat yang meneruskan paket data ke bagian dari jaringan yang dituju.
[2] Alat yang digunakan untuk menghubungkan dua jaringan yang berbeda, biasanya berupa hubungan ke internet.

- 1xx (Informatif): *Request* telah diterima dan proses dapat dilanjutkan.
- 2xx (Berhasil): *Request* telah diterima dan dimengerti.
- 3xx (*Redirection*): Aksi selanjutnya perlu dilakukan untuk memenuhi *request*.
- 4xx (Kesalahan Klien): *Request* mengandung sintaks yang buruk.
- 5xx (Kesalahan *Server*): *Server* tidak dapat memenuhi *request* yang valid.

Kode status HTTP 511 menandakan bahwa klien perlu melakukan otentikasi untuk mendapatkan akses pada jaringan yang bersangkutan[4]. Respon dengan kode status ini harus menyertakan *link* ke sumber yang memungkinkan pengguna untuk memasukkan kredensial. Selain itu, respon dengan kode status ini tidak boleh diberikan oleh server tujuan. Respon ini dimaksudkan sebagai kontrol akses pada jaringan yang akan diberikan oleh komponen perantara dalam jaringan. Respon dengan kode status 511 tidak boleh disimpan oleh *cache*.

Kode Status HTTP 511 dan *Captive Portal*

Kode status 511 diciptakan untuk mengurangi masalah yang ditimbulkan oleh *captive portal* kepada perangkat lunak yang mengharapkan respon dari server tujuan, bukan dari komponen perantara dalam jaringan. Sebagai contoh, perangkat lunak yang bersangkutan mengirimkan *request* HTTP pada port TCP 80 sebagai berikut:

```
GET /index.htm HTTP/1.1
Host: www.example.com
```

Saat menerima *request* tersebut, server login akan mengirimkan respon dengan kode status 511:

```
HTTP/1.1 511 Network Authentication Required
Content-Type: text/html

<html>
    <head>
        <title>Network Authentication Required</title>
        <meta http-equiv="refresh"
            content="0; url=https://login.example.net/">
    </head>
    <body>
        <p>You need to <a href="https://login.example.net/">
        authenticate with the local network</a> in order to gain
        access.</p>
    </body>
</html>
```

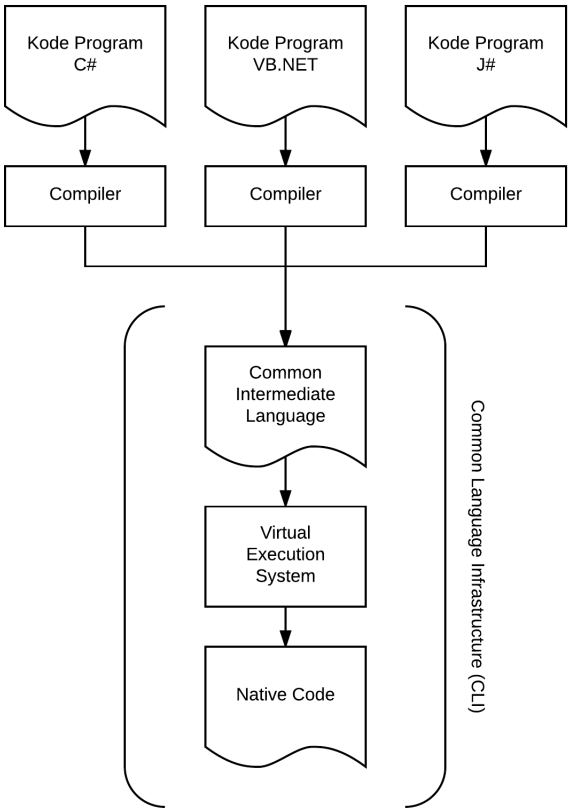
Respon ini memungkinkan klien untuk mendeteksi bahwa respon tersebut bukan berasal dari server tujuan. Selain itu, elemen meta pada HTML yang disajikan memungkinkan klien untuk melakukan login pada *link* yang diberikan.

2.2 *.NET Framework*

.NET adalah platform yang bersifat *general purpose* untuk membangun aplikasi[5]. .NET memberikan lingkungan untuk melakukan pemrograman *high-level* dan tetap memberikan akses *low-level* ke memori dan beberapa API. .NET memiliki beberapa implementasi berdasarkan standar *open* .NET yang mendefinisikan dasar-dasar yang harus dimiliki oleh platform ini. Implementasi-implementasi ini memberikan dukungan bagi beberapa *chip* (seperti x86/x64 dan ARM) dan beberapa sistem operasi (seperti Windows, Linux, iOS, Android, dan macOS).

2.3 *Common Language Infrastructure* (CLI)

.NET memberikan kebebasan kepada *developer* untuk memilih bahasa yang ingin digunakan. Hal ini dapat dilakukan selama bahasa tersebut mendukung .NET. Kebebasan memilih bahasa ini dimungkinkan karena .NET menggunakan spesifikasi *Common Language Inrastructure* (CLI).



Gambar 2.1: Diagram alir proses yang dilalui oleh kode program dalam CLI.

Komponen-komponen penting dalam CLI di antaranya adalah[6]:

- *Common Type System* (CTS)
  CTS memberikan *type system* yang kaya dan mendukung tipe dan operasi yang ditemukan pada banyak bahasa pemrograman.

- *Metadata*
  CLI menggunakan *metadata* untuk menjelaskan dan mereferensi tipe-tipe yang didefinisikan oleh CTS.

- *Common Language Spesification* (CLS)
  CLS adalah perjanjian desainer bahasa pemrograman dan desainer *framework*. Perjanjian ini menentukan bagian minimal dari CTS yang harus diimplementasikan oleh bahasa pemrograman dan *framework* yang bersangkutan.

- *Virtual Execution System* (VES)

VES mengimplementasikan model CTS dan memastikan hal tersebut berjalan sebagaimana mestinya. VES berfungsi untuk menjalankan program yang ditulis untuk CLI.

Seperti yang dijelaskan pada Gambar 2.1, beberapa bahasa pemrograman yang kompatibel dengan .NET adalah C#, VB.NET, dan J#[5]. Setelah kode program dari bahasa-bahasa tersebut diterjemahkan oleh *compiler*nya masing-masing, maka akan terbentuk *Common Intermediate Language* (CIL) yang kemudian akan dibaca dan dieksekusi oleh VES[6]. Implementasi VES pada .NET bernama *Common Language Runtime* (CLR).

2.4 *Universal Windows Platform* (UWP)

Windows 8 memperkenalkan *Windows Runtime* (WinRT) yang merupakan arsitektur aplikasi umum untuk Windows[7]. Saat Windows Phone 8.1 keluar, Windows Runtime pada Windows 8 dan Windows Phone 8.1 disejajarkan agar *developer* dapat membangun satu aplikasi yang dapat dijalankan pada Windows 8 dan Windows Phone 8.1. *Universal Windows Platform* (UWP) pertama kali diperkenalkan pada Windows 10 sebagai perubahan dari model *Windows Runtime*. UWP tidak hanya dapat memanggil API dari WinRT, namun juga API spesifik dari device yang bersangkutan (seperti Win32 dan .NET).

2.5 Kelas WebView Pada UWP

Kelas WebView pada UWP memungkinkan *developer* untuk menampung konten HTML pada suatu aplikasi[8]. WebView tidak mendukung masukkan pengguna seperti *key-down*, *key-up*, dan *pointer-pressed*. Oleh karena itu, dibutuhkan metode lain yang melibatkan InvokeScriptAsync dengan fungsi *eval* javascript untuk menggunakan HTML *event handler* dan fungsi window.external.notify untuk menangani event dari HTML pada aplikasi.

Beberapa *property* yang dimiliki oleh WebView adalah:

- DocumentTitle
  Menyimpan *title* dari dokumen yang sedang ditampilkan dalam bentuk String.
- BaseUri
  Menyimpan *uri* dari dokumen yang sedang ditampilkan dalam bentuk Uri.

Beberapa *method* yang dimiliki oleh WebView adalah:

- InvokeScriptAsync(String scriptName, String[] arguments)
  Method ini digunakan untuk melakukan eksekusi script tertentu pada HTML dengan argumen yang diberikan dalam bentuk *array of string*.
- Navigate(Uri source)
  Method ini digunakan untuk membuka URI tertentu.

Salah satu *event* yang dimiliki oleh kelas WebView adalah *event* ScriptNotify. *Event* ini berguna untuk menangkap hasil dari fungsi javascript window.external.notify.

2.6 Kelas PasswordVault Pada Windows

Kelas PasswordVault merupakan komponen dari Windows Runtime API, dan bukan .NET[8]. Kelas ini merepresentasikan pengunci kredensial. Kredensial yang disimpan menggunakan kelas ini hanya dapat diakses oleh aplikasi atau *service* yang bersangkutan. Beberapa *method* yang dimiliki oleh kelas PasswordVault adalah:

- Add(PasswordCredential credential)
  Method ini berfungsi untuk memasukkan kredensial.
- Retrieve(String resource, String userName)
  Method ini berfungsi untuk mengambil kredensial yang tersimpan di dalam objek PasswordVault.
- Remove(PasswordCredential credential)
  Method ini berfungsi untuk menghapus kredensial yang tersimpan di dalam objek PasswordVault.

BAB 3

ANALISIS

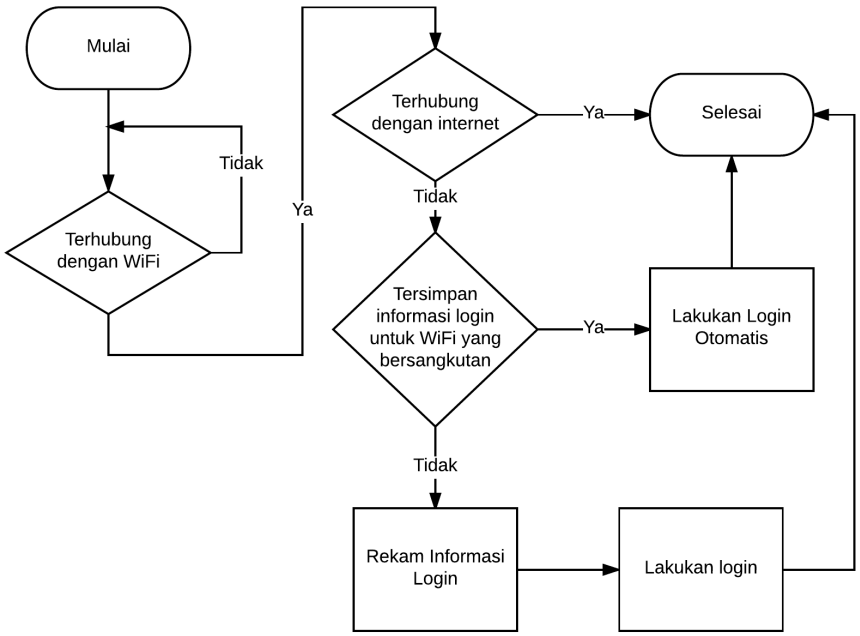
Bab ini menjelaskan mengenai analisis perangkat lunak sejenis, analisis cara penyimpanan informasi login, analisis cara menangkap informasi login, serta analisis perangkat lunak.

3.1 Analisis Perangkat Lunak Sejenis

Perangkat lunak sejenis pada Windows belum dapat ditemukan pada saat penelitian ini dilakukan. Oleh karena itu, perangkat lunak atau aplikasi sejenis yang dianalisis adalah aplikasi yang diciptakan untuk sistem operasi Android. Aplikasi tersebut bernama *WiFi Web Login*.

Diagram Alir *WiFi Web Login*

Langkah-langkah yang perlu ditempuh oleh aplikasi *WiFi Web Login* dapat digambarkan oleh diagram alir.



Gambar 3.1: Diagram alir proses yang perlu dilalui oleh aplikasi *WiFi Web Login*.

Berdasarkan gambar 3.1, langkah-langkah yang harus ditempuh untuk melakukan *login* wifi berbasis web pada aplikasi ini adalah:

1. Deteksi sambungan dengan wifi yang bersangkutan.
2. Deteksi hubungan dengan internet.
3. Jika tidak terjadi hubungan dengan internet, deteksi apakah tersimpan informasi login untuk wifi yang bersangkutan.
4. Jika terdapat informasi login untuk wifi yang bersangkutan maka lakukan login otomatis.
5. Jika tidak terdapat informasi login untuk wifi yang bersangkutan maka rekam informasi login dan lakukan login.

Setelah pengguna melalui sudah pernah melakukan login pertama kali menggunakan aplikasi tersebut, maka aplikasi akan melakukan login otomatis setiap kali terhubung dengan wifi yang bersangkutan.

3.2 Analisis Metode Penyimpanan Informasi Login

Penyimpanan informasi login dapat dilakukan dengan beberapa metode, diantaranya adalah dengan menggunakan file teks atau menggunakan PasswordVault. Penyimpanan informasi menggunakan file teks berarti informasi disimpan dalam bentuk *plaintext* dalam file yang diberikan *access permission* tertentu. Sementara itu, penyimpanan informasi menggunakan PasswordVault memanfaatkan kelas API yang terdapat pada *Universal Windows Platform* (UWP).

Informasi yang perlu disimpan untuk dapat melakukan login otomatis adalah *connection fingerprint* (seperti SSID WiFi, url, dan potongan unik dokumen html), username, password, dan langkah-langkah login seperti menekan tombol. Oleh karena itu, metode penyimpanan menggunakan credential locker dan file teks perlu dianalisis untuk dapat ditentukan metode mana yang paling cocok untuk digunakan dalam penelitian ini.

PasswordVault dapat menyimpan informasi yang berisi *resource* (biasanya berupa nama aplikasi atau string unik lainnya), username, dan password. Informasi yang perlu disimpan selain username dan password adalah *connection fingerprint* dan langkah-langkah login. *Connection fingerprint* dapat disimpan pada resource karena sifatnya yang unik. Sementara itu, langkah-langkah login dapat disimpan pada username atau password karena langkah-langkah login dapat disimpan dalam bentuk String. Akan tetapi, langkah-langkah login sebaiknya disimpan pada password, dipisahkan dengan karakter pemisah tertentu, agar username dapat digunakan sebagai *identifier* unik. PasswordVault memiliki batasan 10 kredensial yang dapat disimpan per aplikasi. Jika aplikasi mencoba menyimpan lebih dari 10 kredensial maka akan terjadi exception. Oleh karena hal ini, PasswordVault menjadi pilihan yang kurang baik untuk kebutuhan perangkat lunak pada penelitian ini.

Metode penyimpanan lainnya adalah dengan menggunakan file teks. Penyimpanan informasi mengunakan file teks dapat dilakukan untuk informasi berbasis teks apapun dan tidak ada batasan banyaknya informasi yang dapat disimpan (kecuali batasan perangkat keras seperti kapasitas hard disk). Akan tetapi, file teks dapat dibaca oleh aplikasi manapun, sehingga penyimpanan

informasi sensitif tidak dapat dilakukan tanpa adanya metode pengamanan tertentu. Salah satu metode pengamanan yang dapat dilakukan adalah dengan mendeklarasikan *file access permission*. Akan tetapi, karena Windows memiliki *security model* per pengguna dan bukan per aplikasi, maka aplikasi lain yang dijalankan oleh pengguna tersebut memiliki akses yang sama kepada file yang bersangkutan. Metode pengamanan lainnya adalah dengan melakukan enkripsi pada file yang bersangkutan sehingga hanya pemegang kunci yang dapat membaca file tersebut. Enkripsi file pada windows dapat dilakukan menggunakan kelas CryptographicEngine. Kunci enkripsi dan dekripsi dapat disimpan menggunakan PasswordVault atau dengan meminta pengguna untuk memasukkan kunci tersebut setiap kali aplikasi dijalankan.

Metode penyimpanan yang digunakan pada penelitian ini adalah metode penyimpanan menggunakan file teks. Akan tetapi, seperti yang sudah dijabarkan sebelumnya, diperlukan metode pengamanan untuk file teks tersebut. Metode pengamanan yang digunakan adalah enkripsi file teks yang bersangkutan. Kunci enkripsi dan dekripsi dibangun secara random saat aplikasi pertama kali dijalankan dan disimpan menggunakan PasswordVault.

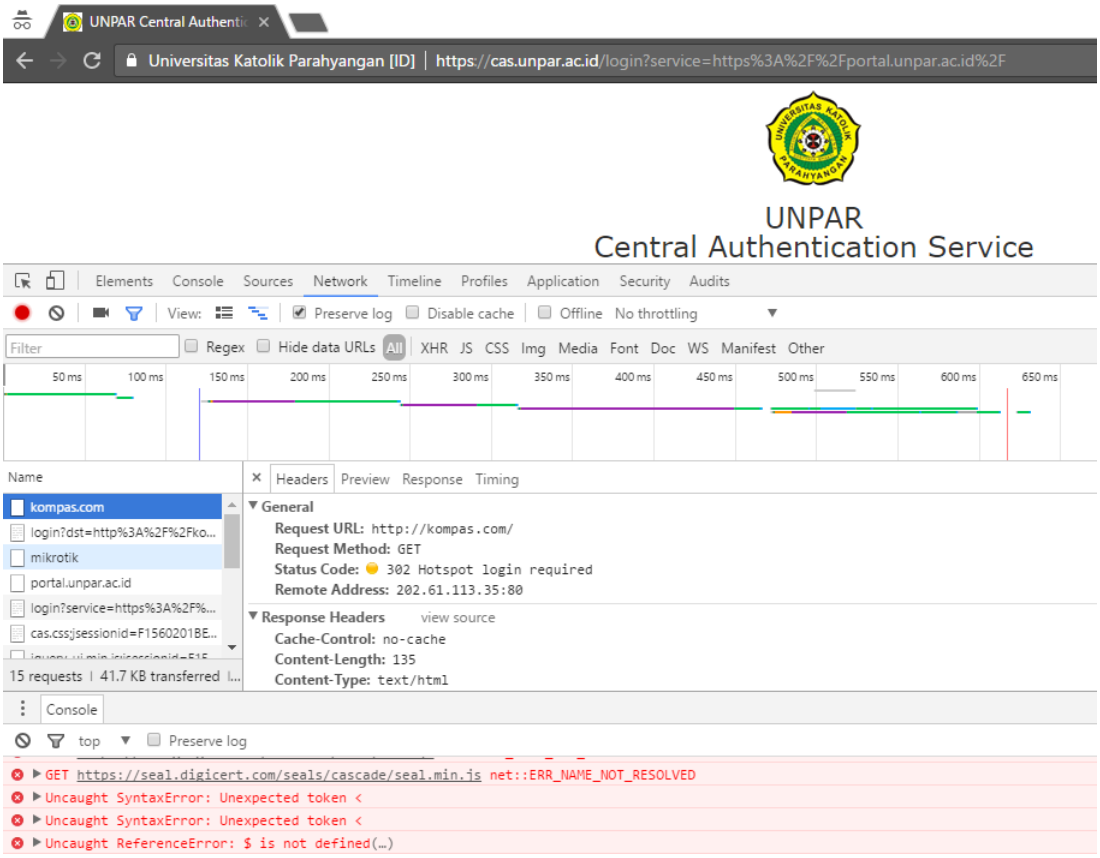
3.3 Analisis Metode Rekam dan Kirim Informasi Login

Kelas WebView pada *Universal Windows Platform* (UWP) hanya dapat dihubungkan dengan kode C# menggunakan javascript. *Method* yang digunakan untuk melakukan eksekusi javascript pada WebView adalah InvokeScriptAsync. Metode ini memiliki parameter string berupa nama fungsi javascript yang ingin dipanggil dan array of string yang berisi argumen yang ingin dikirimkan ke dalam fungsi tersebut. Salah satu fungsi yang dapat dipanggil adalah *eval*. Dengan menggunakan *eval*, ekspresi javascript apapun dapat dijalankan pada WebView. Untuk mengirimkan data dari javascript ke kode C#, dapat dijalankan fungsi window.external.notify dengan parameter berupa string. Oleh karena itu, diperlukan *encoding* tertentu (seperti JSON) untuk memasukkan lebih dari satu argumen.

InvokeScriptAsync dapat digunakan untuk memanggil fungsi *eval* dengan parameter berupa function yang dapat digunakan untuk menekan tombol atau memasukkan nilai pada *text field* tertentu. Selain itu, dapat dimasukkan event listener yang dapat memanggil window.external.notify menggunakan cara ini. Fungsi window.external.notify dapat membantu mengirimkan event-event seperti mouse click, keypress, atau perubahan nilai pada *text field* yang ada pada halaman HTML pada WebView tersebut.

3.4 Analisis Metode Deteksi *Captive Portal*

Berdasarkan teori *captive portal* pada bab 2, dijelaskan bahwa kode status HTTP 511 digunakan untuk memberitahu klien bahwa respon yang didapat bukan berasal dari server tujuan dan diperlukan otentikasi jaringan. Akan tetapi, pada prakteknya, tidak semua *captive portal* melakukan implementasi kode status HTTP 511.



Gambar 3.2: *Screenshot* HTTP *header* yang dikirimkan oleh *captive portal* milik Unpar.

Gambar 3.2 menunjukkan *captive portal* unpar yang menggunakan kode status HTTP 302 untuk memberitahu klien bahwa diperlukan otentikasi jaringan. Berdasarkan teori mengenai kode status HTTP yang dijabarkan pada bab 2, kode status HTTP 3xx adalah kode status yang menyatakan *redirection*. Oleh karena adanya perbedaan implementasi seperti ini, deteksi *captive portal* menggunakan kode status HTTP tidak dapat dilakukan.

DAFTAR REFERENSI

[1] B. Potter and B. Fleck, *802.11 Security*. O'Reilly, 2002.

[2] HTTP Working Group, "Captive Portals." https://github.com/httpwg/wiki/wiki/Captive-Portals, 2016. [Online; diakses 11-September-2016].

[3] Internet Engineering Task Force, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content." https://tools.ietf.org/html/rfc7231, 2016. [Online; diakses 24-September-2016].

[4] Internet Engineering Task Force, "Additional HTTP Status Codes." https://tools.ietf.org/html/rfc6585, 2016. [Online; diakses 24-September-2016].

[5] R. Lander, ".NET Primer." https://docs.microsoft.com/en-us/dotnet/articles/standard/index, 2016. [Online; diakses 24-September-2016].

[6] Ecma International, "Common Language Infrastructure (CLI) Partitions I to IV." http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-335.pdf, 2016. [Online; diakses 24-September-2016].

[7] Microsoft, "Intro to the Universal Windows Platform." https://msdn.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide, 2016. [Online; diakses 24-September-2016].

[8] Microsoft, "Windows API Reference." https://msdn.microsoft.com/en-us/library/windows/apps/bg124285.aspx, 2016. [Online; diakses 24-September-2016].