Wi-Fi Web Auto Login

Yohanes Mario Chandra–2011730031

1 Data Skripsi

Pembimbing utama/tunggal: **Pascal Alfadian**
Pembimbing pendamping: -
Kode Topik : **PAS4105**
Topik ini sudah dikerjakan selama : **2** semester
Pengambilan pertama kali topik ini pada : Semester **41 - Ganjil 16/17**
Pengambilan pertama kali topik ini di kuliah : **Skripsi 1**
Tipe Laporan : **C -** Dokumen pendukung untuk **pengambilan ke-2 di Skripsi 2**

2 Detail Perkembangan Pengerjaan Skripsi

Detail bagian pekerjaan skripsi sesuai dengan rencan kerja/laporan perkembangan terkahir :

1. Melakukan studi literatur mengenai hal-hal yang berkaitan dengan perancangan dan pembuatan aplikasi.
   status : Ada sejak rencana kerja skripsi.
   hasil : Sudah dilakukan studi literatur. *Captive Portal*: *router* atau *gateway* yang akan menutup koneksi eksternal sampai klien yang bersangkutan sudah terotentikasi. Protokol HTTP mengatur kode status yang dapat digunakan untuk mengindikasikan adanya *captive portal*, yaitu kode status 511. *.NET Framework*: platform yang bersifat *general purpose* untuk membangun aplikasi. *.NET Framework* memanfaatkan *Common Language Infrastructure* untuk memberikan kebebasan kepada *developer* untuk memilih bahasa yang ingin digunakan selama bahasa tersebut didukung oleh *.NET Framework*. *Universal Windows Platform* (UWP): arsitektur aplikasi umum untuk Windows. Memungkinkan aplikasi dapat dijalankan pada desktop dan Windows Phone. Kelas WebBrowser tidak jadi dipelajari, karena UWP tidak mendukung WebBrowser. Oleh karena itu, dipelajari kelas WebView. WebView memungkinkan developer untuk menampung konten HTML pada suatu aplikasi. 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. PasswordVault: kelas yang merepresentasikan pengunci kredensial. Kredensial yang disimpan menggunakan kelas ini hanya dapat diakses oleh aplikasi atau service yang bersangkutan.

2. Melakukan analisis perangkat lunak sejenis.
   status : Ada sejak rencana kerja skripsi.
   hasil : 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*. Langkah-langkah yang harus ditempuh untuk melakukan *login* wifi berbasis web pada aplikasi ini adalah:

   (a) Deteksi sambungan dengan wifi yang bersangkutan.
   (b) Deteksi hubungan dengan internet.
   (c) Jika tidak terjadi hubungan dengan internet, deteksi apakah tersimpan informasi login untuk wifi yang bersangkutan.

(d) Jika terdapat informasi login untuk wifi yang bersangkutan maka lakukan login otomatis.

(e) 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. Melakukan analisis kebutuhan untuk mengimplementasikan mekanisme login otomatis ini
status : Ada sejak rencana kerja skripsi.
hasil : Ada beberapa metode yang diperlukan untuk mengimplementasikan mekanisme login otomatis:

- 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.

- 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.

- Metode Deteksi *Captive Portal*

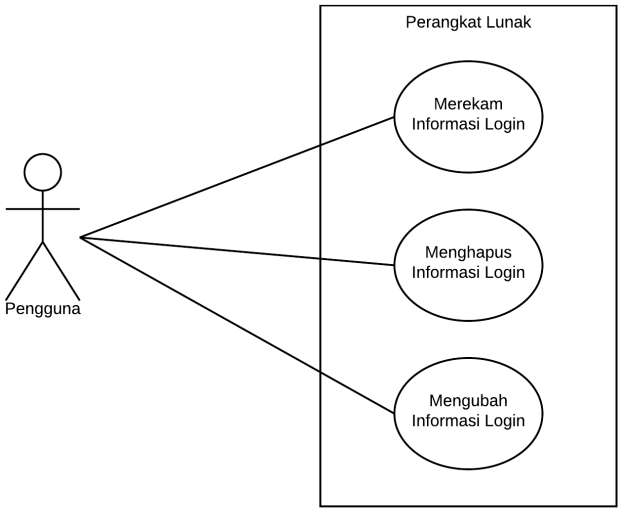
  Kode status HTTP 511 digunakan untuk memberitahu klien bahwa respon yang didapat bukan berasal dari server tujuan dan diperlukan otentikasi jaringan. Akan tetapi, pada prakteknya, tidak semua *captive portal* melakukan implementasi kode status HTTP 511.

  Captive portal milik Unpar mengirimkan kode status 302, dan bukan 511. Oleh karena adanya perbedaan implementasi seperti ini, deteksi *captive portal* menggunakan kode status HTTP tidak dapat dilakukan.

  Persamaan implementasi yang dimiliki oleh setiap *captive portal* adalah dibutuhkannya *redirection* yang dapat dikenali oleh *web browser* agar klien selalu diarahkan ke halaman *captive portal* tersebut sebelum melakukan otentikasi. Oleh karena itu, untuk setiap HTTP *request* yang dilakukan oleh klien sebelum melakukan otentikasi, HTTP *response* yang diberikan bukanlah berasal dari server tujuan. Sifat ini dapat dimaanfaatkan untuk keperluan deteksi *captive portal* dengan mengirimkan *request* ke server yang sudah ditentukan sebelumnya, dan mendeteksi apakah *response* yang diberikan sesuai dengan harapan. Jika *response* yang diberikan tidak sesuai dengan harapan, maka dapat diasumsikan bahwa klien dibatasi oleh suatu *captive portal*.

  Kelas WebView pada *Universal Windows Platform* (UWP) memiliki kemampuan deteksi *redirection* dan *response* yang tidak sesuai harapan seperti *web browser* pada umumnya. Oleh karena itu, kelas WebView digunakan untuk melakukan deteksi *captive portal* pada penelitian ini tanpa perlu memeriksa kode status HTTP setiap *response*. Penggunaan kelas WebView juga akan mempermudah proses otomatisasi login karena WebView dapat melakukan hal-hal yang dapat dilakukan oleh *web browser* pada umumnya seperti menjalankan javascript dan menampilkan halaman HTML.

Diagram *Use Case*:



Gambar 1: Diagram *use case* untuk perangkat lunak yang dibangun.

Skenario merekam informasi login

Nama: Merekam informasi login
Aktor: Pengguna
Kondisi awal: Perangkat lunak mendeteksi adanya *captive portal*.
Deskripsi: Pengguna menyimpan informasi login.
Kondisi Akhir: Informasi login tersimpan di dalam perangkat lunak.

Skenario:

(a) Pengguna memasukkan informasi login ke dalam form HTML.

(b) Sistem menyimpan informasi login yang dimasukkan oleh pengguna.

Skenario menghapus informasi login

Nama: Menghapus informasi login
Aktor: Pengguna
Kondisi awal: Perangkat lunak sudah dijalankan.
Deskripsi: Pengguna menghapus informasi login.
Kondisi Akhir: Informasi login dihapus dari perangkat lunak.

Skenario:

(a) Pengguna memilih untuk menghapus informasi login.

(b) Sistem menghapus informasi login.

Skenario mengubah informasi login

Nama: Mengubah informasi login
Aktor: Pengguna
Kondisi awal: Perangkat lunak sudah dijalankan.
Deskripsi: Pengguna mengubah informasi login.

Kondisi Akhir: Informasi login diubah dari perangkat lunak.

Skenario:

(a) Pengguna memilih untuk mengubah informasi login.

(b) Sistem menampilkan form ubah informasi login.

(c) Pengguna memasukkan informasi login baru.

(d) Sistem menghapus informasi login lama dan menyimpan informasi login baru.

Skenario memicu login otomatis

Nama: Memicu login otomatis
Aktor: WiFi
Kondisi awal: Perangkat lunak sudah dijalankan.
Deskripsi: Sistem memicu login otomatis berdasarkan perubahan status WiFi.
Kondisi Akhir: klien terotentikasi pada *captive portal*.

Skenario:

(a) Sistem mendeteksi adanya koneksi WiFi yang terjadi.

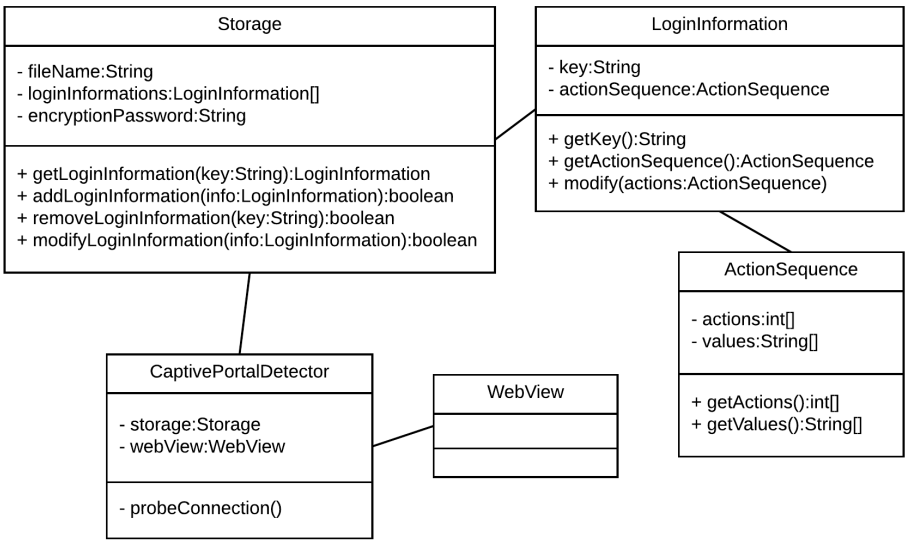
(b) Sistem mendeteksi adanya *captive portal*.

(c) Sistem mendeteksi adanya informasi login untuk *captive portal* tersebut.

(d) Sistem mengirimkan informasi login kepada *captive portal*.

(e) Sistem mendeteksi koneksi dengan internet dan klien sudah terotentikasi pada *captive portal* tersebut.

Diagram Kelas:



Gambar 2: Diagram kelas untuk perangkat lunak yang dibangun.

Kelas-kelas yang dibutuhkan pada perangkat lunak ini adalah:

- CaptivePortalDetector
  Kelas ini digunakan untuk mendeteksi keberadaan *captive portal*. Jika captive portal terdeteksi, maka informasi login yang tersimpan dalam Storage digunakan. Jika tidak terdapat informasi login dalam Storage, maka direkam informasi login baru.

- Storage
  Kelas ini digunakan untuk menyimpan seluruh informasi login dalam bentuk file teks yang sudah terenkripsi.

- LoginInformation
  Kelas ini digunakan untuk menyimpan informasi login dalam bentuk key atau fingerprint, serta ActionSequence

- ActionSequence
  Kelas ini digunakan untuk menyimpan langkah-langkah login dalam bentuk urutan aksi dan nilai-nilai yang berkaitan dengan aksi tersebut.

4. Merancang perangkat lunak login otomatis ini
   status : Ada sejak rencana kerja skripsi.
   hasil : Dilaksanakan pada Skripsi 2.

5. Melakukan pengujian terhadap perangkat lunak untuk menghasilkan perbaikan teradap perangkat lunak tersebut.
   status : Ada sejak rencana kerja skripsi.
   hasil : Dilaksanakan pada Skripsi 2.

6. Membuat kesimpulan dari hasil penelitian dan saran untuk penelitian selanjutnya.
   status : Ada sejak rencana kerja skripsi.
   hasil : Dilaksanakan pada Skripsi 2.

7. Melakukan pengujian (dan eksperimen) yang melibatkan responde untuk menilai hasil simulasi secara kualitatif
   status : Ada sejak rencana kerja skripsi.
   hasil : Dilaksanakan pada Skripsi 2.

8. Menulis dokumen skripsi.
   status : Ada sejak rencana kerja skripsi.
   hasil : Sudah dikerjakan sampai dengan bab 3 (Analisis).

3 Pencapaian Rencana Kerja[1]

Persentase penyelesaian skripsi sampai dengan dokumen ini dibuat dapat dilihat pada tabel berikut :

1*	2*(%)	3*(%)	4*(%)	5*	6*(%)
1	5	5			5
2	10	10			10
3	15	10	5	kaji ulang singkat kebutuhan pada S2	10
4	10		10		
5	15		15		
6	10		10		
7	5		5		
8	30	15	15	sebagian bab 1 dan 2, serta bagian awal analisis di S1	15
Total	100	40	60		40

Keterangan (*)

1 : Bagian pengerjaan Skripsi (nomor disesuaikan dengan detail pengerjaan di bagian 5)

2 : Persentase total

3 : Persentase yang akan diselesaikan di Skripsi 1

4 : Persentase yang akan diselesaikan di Skripsi 2

5 : Penjelasan singkat apa yang dilakukan di S1 (Skripsi 1) atau S2 (skripsi 2)

6 : Persentase yang sidah diselesaikan sampai saat ini

Bandung, 22/11/2016

Yohanes Mario Chandra

Menyetujui,

Nama: Pascal Alfadian
Pembimbing Tunggal

[1] Terdapat kesalahan pada dokumen rencana kerja terdahulu di mana terdapat 8 poin rencana kerja, sementara hanya tertera 7 poin pada tabel rencana kerja. Poin-poin rencana kerja yang benar adalah yang ada pada dokumen ini.