Integrasi *Login* Tanpa Mengetik *Password* pada WordPress

Mochamad Arifin[1], Agus Bejo[2], Warsun Najib[3]

***Abstract*—Nowadays, almost everyone has an account on the internet, but some of them often use simple and guessable password, since a complex password is difficult to memorize. Surely, it will compromise the account security and system themselves. Therefore, an integrated login system without typing and remembering password is needed. This paper describes the development of an integrated login system that can perform authentication on a device without typing and remembering password by developing a WordPress plugin and a smartphone application. The development of the system has successfully shown the QR Code on the WordPress login page and automatically redirect the user to the admin page when the login process is done through scanning the QR code by using smartphone. Password only needs to be typed in the first login, and users do not have to retype it for the next login process. Android application development has resulted a password manager application that helps the users to manage the password and to have a secure password storage. Login integration without typing a password can improve account security and reduce the risk of man-in-the-middle and key-logger attack.**

***Intisari*—Saat ini hampir setiap orang memiliki akun di internet, tetapi beberapa sering menggunakan *password* yang kurang kompleks dan dapat ditebak, karena *password* yang kompleks sulit diingat. Hal ini akan membahayakan akun dan keamanan sistem itu sendiri. Oleh karena itu, perlu adanya sebuah integrasi sistem *login* tanpa mengetik dan mengingat *password*. Makalah ini menjelaskan tentang pengembangan integrasi sistem *login* yang dapat melakukan autentikasi pada suatu perangkat tanpa mengetik dan mengingat *password* dengan cara mengembangkan *plugin web* WordPress dan aplikasi *smartphone*. Pengembangan sistem telah berhasil menampilkan *QR Code* pada halaman *login* WordPress serta secara otomatis mengarahkan pengguna ke halaman admin ketika proses *login* berhasil dilakukan melalui pemindaian *QR Code* menggunakan aplikasi *smartphone*. *Password* hanya perlu diisikan pada pertama kali *login*. Untuk proses *login* selanjutnya pengguna tidak perlu mengisikan informasi *login* lagi. Pengembangan aplikasi Android telah menghasilkan sebuah aplikasi *password manager* yang membuat pengguna tidak perlu mengingat *password* serta mempunyai media penyimpanan *password* yang aman. Integrasi sistem *login* tanpa mengetikkan *password* mampu meningkatkan keamanan akun serta dapat mengurangi risiko serangan *man-in-the-middle* dan *key-logger*.**

Kata Kunci— Login,* Wordpress, *QR Code.

[1] *Mahasiswa, Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada, Jl. Grafika 2 Yogyakarta 55281 INDONESIA (tlp: 085726501017; e-mail: arifin.mti13@mail.ugm.ac.id)*

[2, 3] *Dosen, Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada, Jl. Grafika 2 Yogyakarta 55281 INDONESIA (e-mail: agusbj@ugm.ac.id, warsun@ugm.ac.id)*

I. PENDAHULUAN

Saat ini hampir setiap orang memiliki akun di dunia maya, tetapi beberapa dari mereka tidak mementingkan keamanan akun dengan menggunakan *password* yang lemah. Survei yang dilakukan oleh CSIdentity pada tahun 2012 menunjukkan bahwa 61% dari 1.200 orang memiliki kebiasaan menggunakan *password* yang sama untuk *web* yang berbeda [1]. Ketika salah satu akun pada *web* telah diambil alih oleh *attacker*, maka akun lain yang memiliki *password* sama menjadi mudah untuk diambil alih pula. SplashData, Inc yang merupakan penyedia aplikasi keamanan telah melakukan publikasi data mengenai *password* tidak baik yang paling sering digunakan, dan hasilnya "123456" adalah *password* yang paling buruk yang banyak digunakan [2]. Hal ini mengindikasikan bahwa banyak orang menggunakan *password* yang lemah sehingga menyebabkan sistem keamanan menjadi rentan.

Aplikasi *password manager* merupakan salah satu solusi untuk mengingat *password*, tetapi aplikasi ini memiliki keterbatasan ketika pengguna perlu melakukan autentikasi *login* pada perangkat publik. Pengguna perlu mengetikkan dan mengirimkan *password* dari perangkat tersebut ketika melakukan autentikasi. Hal ini mungkin berbahaya karena perangkat tersebut rentan terhadap penyadapan data baik melalui jaringan (*man-in-the-middle*) maupun perekaman papan ketik (*key logger*).

Oleh karena itu, perlu adanya integrasi sistem yang dapat melakukan proses *login* tanpa mengetikkan *password* serta aplikasi *password manager* sebagai penyimpanan akun, sehingga pengguna dapat menggunakan *password* yang rumit tanpa perlu mengingatnya. Integrasi sistem *login* yang dilakukan adalah dengan mengembangkan sebuah *plugin* WordPress yang dapat menampilkan *QR Code* dan aplikasi Android untuk autentikasi *login* sekaligus sebagai aplikasi *password manager*.

II. AUTENTIKASI MENGGUNAKAN *QR CODE*

Pada tahun 2010 telah dilakukan penelitian mengenai penggunaan *QR Code* untuk autentikasi pengguna [3]. Penelitian tersebut menjabarkan tentang penggunaan *QR Code* sebagai media untuk melakukan autentikasi menggunakan *smartphone* tanpa menggunakan *password*. Kemudian sistem akan membuat sebuah token yang berumur panjang yang disimpan pada *smartphone* pengguna sebagai kredensial yang diperlukan untuk autentikasi tanpa menggunakan *password*. Penelitian tersebut lebih menekankan tentang proses dan algoritme pembuatan token rahasia yang memanfaatkan enkripsi satu arah.

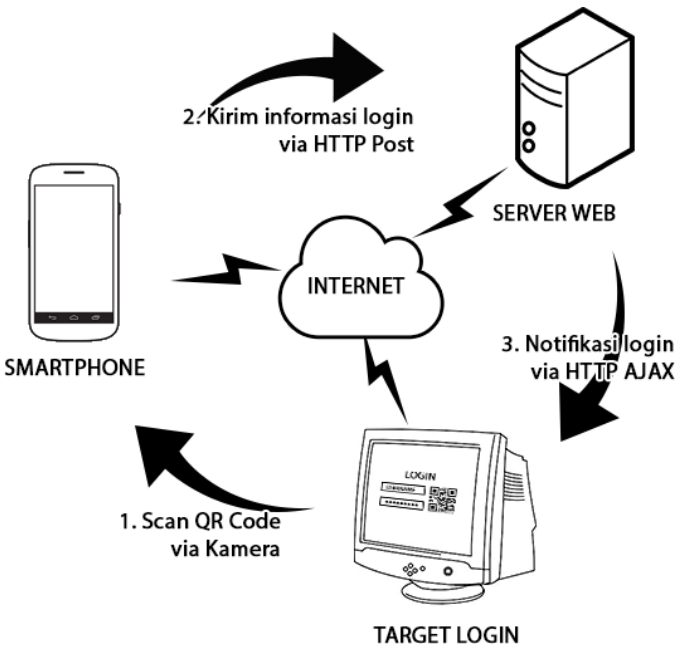
Selain penelitian tersebut, telah dilakukan juga penelitian mengenai proses *login* menggunakan *QR Code* [4]. Dalam

penelitian tersebut dikembangkan sebuah aplikasi desktop berbasis Java yang digunakan sebagai *password manager* dan dapat melakukan proses *login* pada *web* menggunakan pemindaian *QR Code*. Alur *login* yang digunakan pada penelitian tersebut dimulai dari proses pengisian informasi akun pada aplikasi desktop, kemudian mencetak *QR Code* yang dibuat oleh aplikasi. Proses *login* dilakukan dengan memindai *QR Code* yang sudah dicetak menggunakan kamera pada laptop atau komputer yang sudah terpasang aplikasi desktop tersebut.

Berbeda dengan penelitian-penelitian sebelumnya, pada makalah ini metode integrasi sistem *login* tidak menghilangkan fungsi *password* dan proses autentikasi pengguna tetap menggunakan *password*. Kredensial yang disimpan pada perangkat *smartphone* bukan merupakan token atau kunci akses melainkan informasi *login* seperti *username* dan *password*. *QR Code* pada penelitian ini juga tidak perlu dicetak menjadi bentuk fisik, tetapi cukup ditampilkan pada layar monitor.

III. ARSITEKTUR SISTEM

Penelitian ini mengembangkan sebuah *plugin* WordPress yang merupakan platform blog yang paling populer saat ini [5], [6]. Skema integrasi *login* dibuat agar memungkinkan terjadinya autentikasi *login* pada suatu perangkat tanpa mengetikkan *username* dan *password* dengan memindai *QR Code* menggunakan *smartphone. QR Code* akan ditampilkan pada halaman *login* Wordpress dengan mengembangkan sebuah *plugin* yang dapat mengambil URL dan *cookie* dari *web* tersebut, kemudian membuat sebuah citra *QR Code* dengan pustaka JavaScript jQuery QR Code [7], [8].



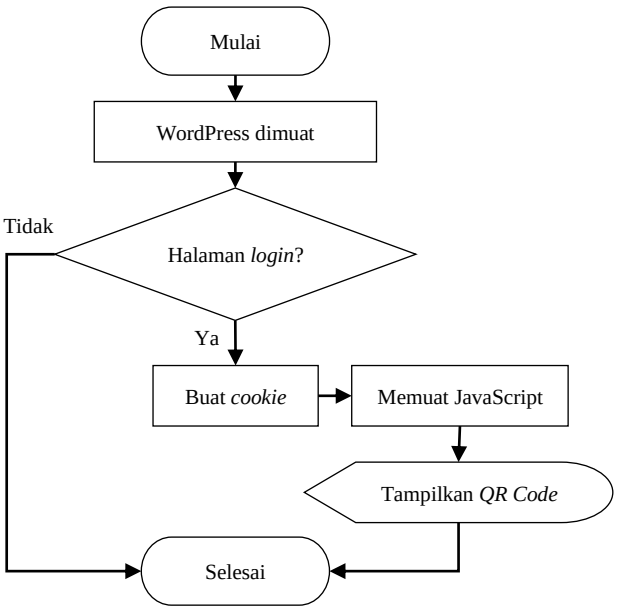
Gbr. 1 Skema proses *login* menggunakan *QR Code.*

Gbr. 1 merupakan skema proses *login* dan interaksi antar perangkat yang dihasilkan. Setiap perangkat harus terhubung dengan internet supaya dapat saling mengirimkan dan menerima data. *Target Login* merupakan perangkat yang belum memiliki autentikasi *login* yang dapat berupa komputer desktop, laptop, *gadget*, *smartphone*, atau perangkat lainnya yang dapat mengakses laman *web*. *Smartphone* merupakan perangkat personal pengguna yang memiliki aplikasi pemindaian *QR Code login* dan sebagai media penyimpanan informasi akun. *Server Web* merupakan *host web* Wordpress yang sedang diakses oleh *Target Login* dan memiliki *plugin* untuk menampilkan *QR Code*. Proses yang terjadi pada setiap perangkat akan dijelaskan pada subbagian berikut.

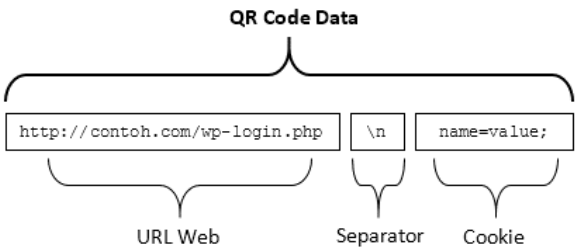
A. Script Browser

Script pada *browser* ditulis menggunakan JavaScript dan dimasukkan pada *plugin* yang terpasang pada *web* dengan CMS WordPress. JavaScript pada *plugin* berfungsi untuk melakukan pengecekan status *login* dan menampilkan *QR Code* dengan menggunakan *library* jQuery QR Code [8], [9].

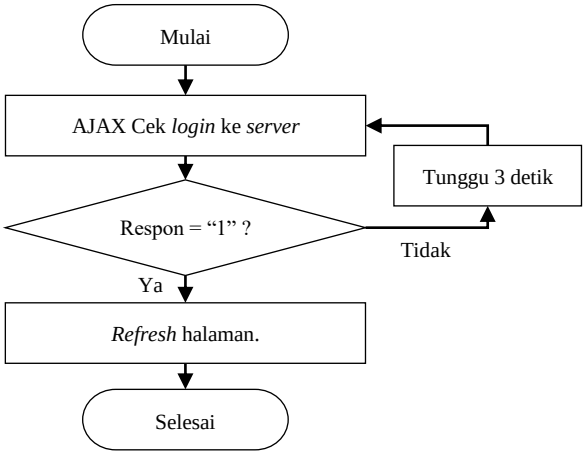
Gbr. 2 merupakan diagram alir pada *plugin* WordPress [10], [11]. Proses dimulai ketika WordPress diakses oleh pengguna. Jika pengguna sedang mengakses halaman *login*, maka *plugin* akan membuat sebuah *cookie* yang digunakan sebagai *session* pengguna saat itu. Hal ini diperlukan karena WordPress tidak memberikan *cookie session* terhadap pengguna yang belum melakukan *login*. Kemudian *plugin* akan memuat JavaScript pada halaman *login* Wordpress.



Gbr. 2 Diagram alir menampilkan *QR Code*.



Gbr. 3 Struktur data pada *QR Code*.



Gbr. 4 Diagram alir pengecekan *login*.

Gbr. 3 merupakan struktur data pada QR Code yang berisi URL dan *cookie* yang diambil menggunakan JavaScript. Data digabungkan menggunakan karakter ganti baris atau "\n" sebagai indikator pemisah antara data tersebut. Struktur data tersebut diperuntukkan untuk dibaca menggunakan aplikasi *smartphone* sehingga aplikasi dapat membedakan antara data *QR Code* biasa dan *QR Code* untuk aplikasi *QR Code login*.

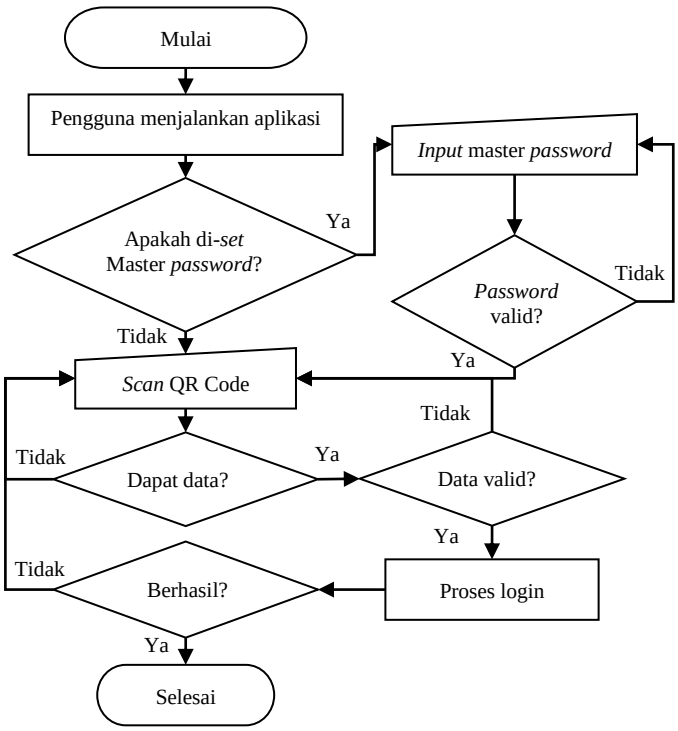
Browser harus dapat melakukan perpindahan halaman secara otomatis ketika pengguna selesai melakukan *login* menggunakan aplikasi *smartphone*. Supaya *browser* dapat mendeteksi status *login* secara otomatis ketika aplikasi *smartphone* selesai melakukan proses *login*, maka *script* akan menggunakan AJAX untuk melakukan pengecekan status *login* terhadap *server web* secara berkala.

Gbr. 4 merupakan diagram alir pada proses pengecekan *login*. Java*script* akan melakukan *asynchronous call* kepada *server web*. Ketika mendapat respons "0", berarti status belum *login*, *script* akan menunggu 3 detik sebelum melakukan pengecekan ulang. Setelah memperoleh respons "1", *script* akan me-*refresh* halaman. Jika status benar-benar *login*, maka *server* akan mengarahkan pengguna ke halaman admin.

B. Proses Aplikasi Smartphone

Aplikasi Android pada *smartphone* menggunakan *library* ZBar untuk pembacaan *QR Code* dari kamera [12], [13]. Sebelum data diproses, aplikasi akan melakukan validasi untuk memastikan bahwa data tersebut merupakan data yang diperuntukkan untuk aplikasi *smartphone QR Code login.*

Gbr. 5 merupakan diagram alir proses login yang terjadi pada aplikasi *smartphone* dan merupakan proses utama yang diterapkan pada pengembangan aplikasi Android [14], [15]. Proses dimulai ketika pengguna menjalankan aplikasi. Jika pengguna telah mengatur master *password*, maka sebelum dapat masuk ke tampilan utama aplikasi tersebut, pengguna diminta untuk memasukkan master *password* tersebut. Master *password* bersifat opsional untuk memproteksi informasi akun yang disimpan pada *smartphone* menggunakan enkripsi AES. Jika master *password* tidak diatur, maka pengguna dapat langsung menjalankan aplikasi dan *file* dienkripsi menggunakan kunci AES *default* dari aplikasi. Hal ini berfungsi agar *file* tidak dapat dibaca oleh pihak lain. Setelah pengguna memindai *QR Code* dan mendapati data dari kamera, aplikasi akan melakukan validasi data. Setelah data dinyatakan valid, aplikasi akan melakukan proses *login* jika tersedia akun yang bersangkutan dengan URL yang dimuat dalam data. Jika tidak tersedia, aplikasi akan melakukan *scrapping* pada URL *web* tersebut untuk mendapatkan informasi *form login* dari *web*, kemudian menampilkan pada layar *smartphone* untuk diisi oleh pengguna.



Gbr. 5 Diagram alir aplikasi *smartphone.*

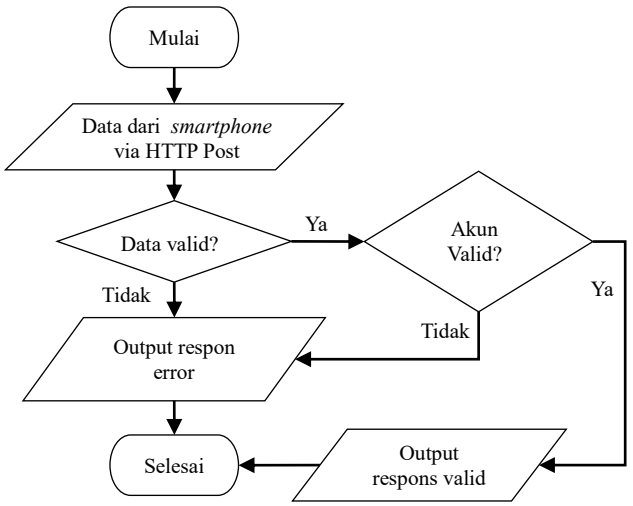
C. Proses Website Server

Setelah aplikasi mengirim informasi pada server web, selanjutnya *Plugin* WordPress yang sudah diaktifkan akan melakukan validasi terhadap data akun yang dikirimkan dan memberikan akses login pada *cookie session* jika data valid. *Cookie session* pada web *browser* harus cocok dengan *cookie session* yang ada pada *smartphone* agar proses login dapat terjadi pada perangkat desktop.

Gbr. 6 menunjukkan proses *login* pada *plugin* WordPress. Proses dimulai dari penerimaan data yang dikirim oleh *smartphone* kepada *web server* melalui HTTP *Post. Plugin* akan melakukan validasi data terlebih dahulu. Jika data valid, sistem akan memeriksa apakah *username* dan *password* yang dikirimkan cocok dengan akun yang tersedia. Jika cocok, maka sistem akan memberikan respons sukses dan ditampilkan oleh aplikasi *smartphone* yang ada pada pengguna.

Respons yang diberikan dari *plugin* Wordpress berupa status kode kostum dari HTTP versi 1.1 yang mengikuti standar W3C, seperti berikut ini [16].

```
HTTP/1.1 440 invalid userpass
Server: nginx
Date: Sat, 11 Feb 2017 16:02:45 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.6.23
X-QLogin: 1.0, 440
```



Gbr. 6 Diagram alir proses pada *plugin* di *web server*.

TABEL I
DAFTAR STATUS RESPON

Kode	Pesan	Penjelasan
240	valid	*Login* valid dan berhasil
440	invalid userpass	*Username* atau *password* salah
441	invalid token	*Cookie* token tidak terdaftar di *server*
442	expired token	*Cookie* token sudah *expired*
443	token used	*Cookie* token sudah pernah digunakan
444	was loggin	Status sudah *login*
445	invalid pass	*Username* terdaftar, tetapi *password* salah
446	invalid user	*Username* tidak terdaftar

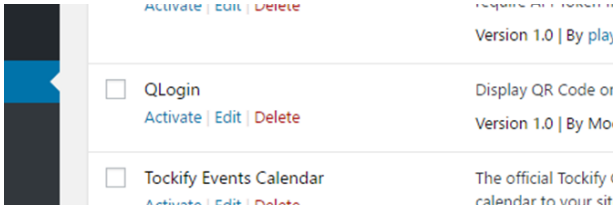
HTTP kode 440 mengindikasikan bahwa *login* yang dikirim tidak valid. *Header* "X-QLogin" merupakan *header* yang dibuat oleh *plugin QR Code* yang terpasang pada *server*. *Header* ini berisi versi *plugin* dan juga status kode respons. Kode respons pada *header* "X-QLogin" dan pada HTTP status kode dapat berbeda, karena terdapat beberapa *web server shared hosting* yang tidak memperbolehkan kostum kode pada HTTP status, sehingga *plugin* perlu menyertakan pula status respons pada *header* "X-Qlogin" agar dapat bekerja dengan baik. Aplikasi *smartphone* akan membaca kode respons yang ada pada *header* "X-Qlogin" daripada membaca kode respon dari HTTP status *header*. Tabel I menyajikan daftar kode respons dan keterangannya.

Kode respons pada Tabel I akan dibaca oleh aplikasi *smartphone* untuk dapat memberikan *feedback interface* kepada pengguna yang sedang melakukan proses *login* melalui aplikasi *smartphone*.

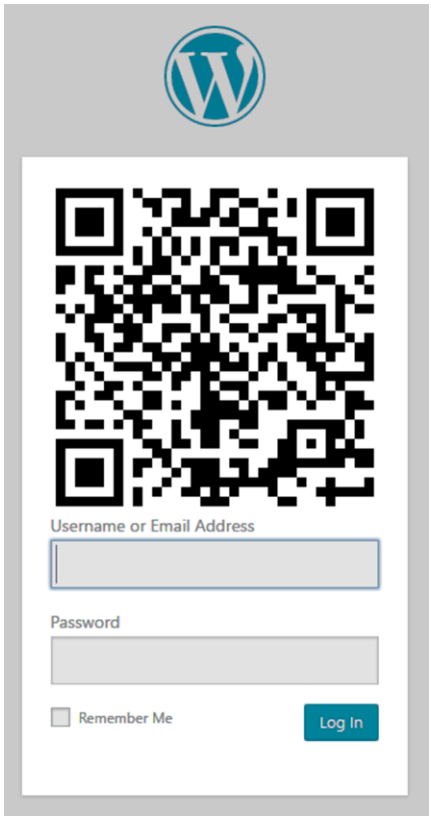
IV. INTEGRASI SISTEM LOGIN

A. *Plugin WordPress*

Plugin WordPress yang dikembangkan dalam makalah ini akan diberi nama "QLogin". *Plugin* dipasang pada WordPress dengan cara mengunggah *file .zip* dan kemudian diaktifkan.



Gbr. 7 Pemasangan *plugin* WordPress.



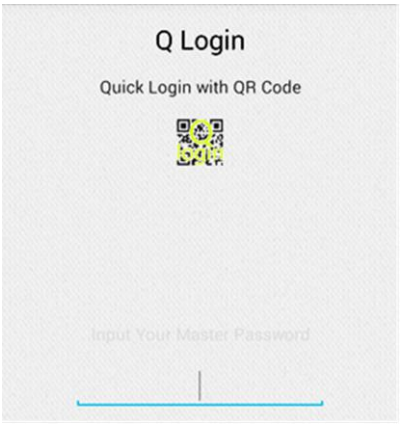
Gbr. 8 Halaman Login dengan *QR Code*.

Gbr. 7 merupakan tampilan halaman *plugin* pada WordPress setelah *plugin* QLogin diunggah. Aktivasi *plugin* Qlogin dilakukan dengan mengeklik "*activate*". *Plugin* QLogin akan menampilkan *QR Code* pada halaman *login* WordPress untuk dipindai menggunakan aplikasi *smartphone.*

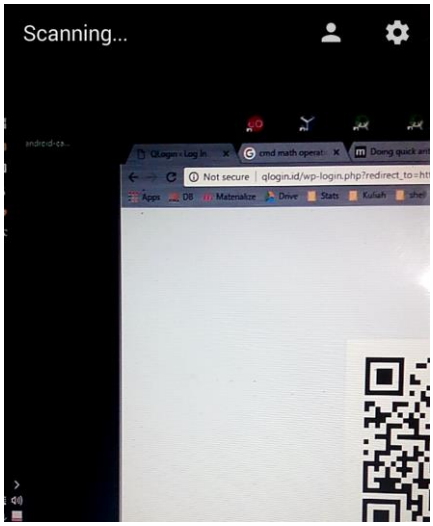
Gbr. 8 merupakan tampilan pada peramban *web* yang sedang mengakses laman *login* WordPress. Pengguna dapat *login* seperti biasa dengan memasukkan *username* dan *password* atau menggunakan *QR Code* yang ditampilkan oleh *plugin* QLogin. Agar dapat *login* menggunakan *QR Code*, pengguna harus menggunakan aplikasi *mobile* dari QLogin dan kemudian melakukan pemindaian terhadap *QR Code* yang ditampilkan pada halaman *login* tersebut.

B. *Aplikasi Android*

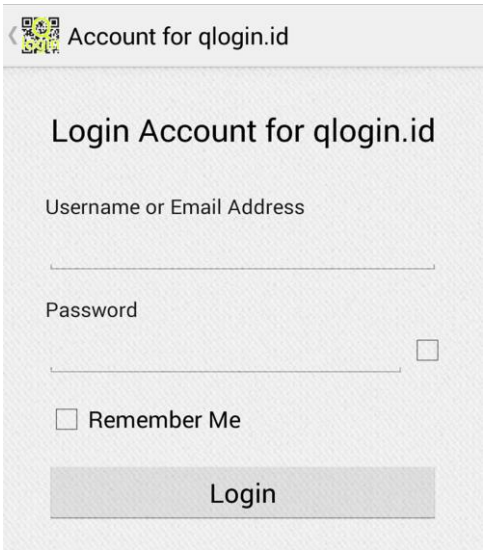
Implementasi aplikasi *smartphone* dilakukan dengan cara melakukan pemasangan pada *smartphone* Android kemudian dijalankan. *Smartphone* harus memiliki kamera agar dapat melakukan pemindaian *QR Code* yang ditampilkan pada layar monitor yang sedang mengakses halaman *login* WordPress.



Gbr. 9 Tampilan masuk aplikasi menggunakan master *password*.

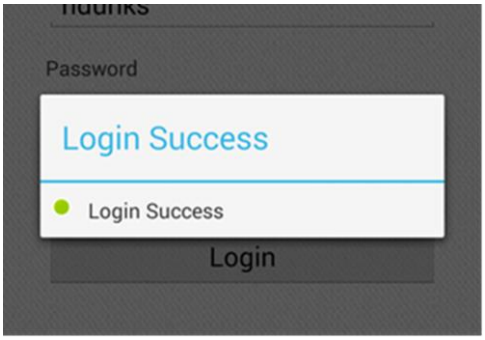


Gbr. 10 Tampilan pemindaian *QR Code* menggunakan kamera.



Gbr. 11 Tampilan pengisian akun untuk *login*.

Gbr. 9 merupakan tampilan halaman aplikasi ketika pengguna diminta untuk mengisikan master *password*. Jika pengguna tidak mengatur master *password* sebelumnya, maka halaman ini tidak akan tertampil dan akan langsung membuka tampilan pemindaian kamera.



Gbr. 12 Tampilan proses *login*.

Gbr. 10 merupakan tampilan ketika aplikasi melakukan pemindaian *QR Code* menggunakan kamera. Setelah aplikasi mendapati *QR Code*, kemudian aplikasi memproses data tersebut dan melakukan pengecekan apakah sudah ada akun yang bersangkutan atau tidak. Ketika aplikasi tidak menemukan akun yang bersangkutan dengan *web* tersebut, maka aplikasi akan menampilkan halaman untuk mengisikan informasi *login*.

Gbr. 11 merupakan tampilan setelah aplikasi memproses *QR Code* dan tidak ditemukan akun yang bersangkutan dengan URL yang ada pada *QR Code*. Pengguna diminta untuk mengisikan informasi akun yang sudah terdaftar pada *web* tersebut, setelah itu aplikasi akan melakukan proses *login*.

Gbr. 12 merupakan tampilan proses login setelah login berhasil. Aplikasi akan menyimpan informasi login tersebut, selanjutnya pengguna tidak perlu mengisikan informasi akun lagi ketika pengguna melakukan login pada web yang sama.

V. KESIMPULAN

Dalam makalah ini telah mampu ditingkatkan keamanan akun dengan menerapkan sistem integrasi *login* tanpa mengetikkan *password*, sehingga pengguna dapat menggunakan *password* yang rumit dan tidak mudah ditebak oleh orang lain. Pengguna tidak perlu mengingat *password* tersebut karena aplikasi *smartphone* didesain untuk dapat menyimpan informasi akun-akun yang sudah diisikan.

Selain itu, sistem integrasi *login* menggunakan *QR Code* juga dapat mengurangi risiko *man-in-the-middle attack* dan juga *key logger* pada saat pengaksesan akun menggunakan perangkat publik. Hal ini disebabkan proses *login* dapat terjadi pada suatu perangkat tanpa perlu mengirimkan *username* dan *password* dari perangkat tersebut. Namun, pada dasarnya proses *login* dilakukan oleh perangkat *smartphone* pribadi milik pengguna yang menggunakan jaringan internet yang lebih terpercaya.

REFERENSI

[1] CSIdentity, "Consumer Survey: Password Habits," CSIdentity, America, 2012.

[2] SplashData, Inc, ""123456" Maintains the Top Spot on SplashData's Annual "Worst Passwords" List," 22 Januari 2017. [Online]. Available: https://www.teamsid.com/worst-passwords-2016/. [Diakses 10 Februari 2016].

[3] K.-C. Liao dan W.-H. Lee, "A Novel User Authentication Scheme Based on QR-Code," *JOURNAL OF NETWORKS*, vol. Vol 5, pp. 937-941, 2010.

[4] M. Bachtiar and A. Mazharuddin, "Smart Login pada Situs Web Menggunakan QR-Code," *JURNAL TEKNIK POMITS*, vol. 1, no. 1, pp. 1-4, 2012.

[5] TopTenREVIEWS, "The Best Content Management System Software of 2017," 2017. [Online]. Available: http://www.toptenreviews.com/business/internet/best-content-management-system-software/. [Diakses 12 Februari 2017].

[6] BuiltWith® Pty Ltd, "Statistics for websites using CMS technologies," [Online]. Available: http://trends.builtwith.com/cms. [Diakses 7 Oktober 2015].

[7] DENSO WAVE INCORPORATED, "History of QR Code," [Online]. Available: http://www.qrcode.com/en/history/. [Diakses 7 Oktober 2015].

[8] L. Jung, "jQuery qrcode generate QR codes dynamically," 26 Mei 2016. [Online]. Available: https://larsjung.de/jquery-qrcode/. [Accessed 26 Januari 2017].

[9] T. Nojiri, "Two-dimensional code, methods and apparatuses for generating, displaying and reading the same". United States Paten US 7032823 B2, 25 April 2006.

[10] WordPress.org, "WordPress Plugin Directory," [Online]. Available: https://wordpress.org/plugins/. [Diakses 5 Oktober 2015].

[11] M. Hills, "Navigating the WordPress plugin landscape," in *2016 IEEE 24th International Conference on Program Comprehension (ICPC)*, 2016.

[12] J. Brown, "ZBar Bar Code Reader," 15 Juli 2011. [Online]. Available: http://zbar.sourceforge.net/. [Diakses 6 Februari 2017].

[13] M. M. K. Suhas Holla, "ANDROID BASED MOBILE APPLICATION DEVELOPMENT and its SECURITY," *International Journal of Computer Trends and Technology*, p. 490, 2012.

[14] Android Team, "Welcome to the Android Open Source Project!," [Online]. Available: https://source.android.com/. [Diakses 12 Februari 2017].

[15] Android Team, "Download Android Studio and SDK Tools," [Online]. Available: https://developer.android.com/studio/index.html. [Accessed 12 Februari 2017].

[16] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach dan T. Berners-Lee, "Hypertext Transfer Protocol," Juni 1999. [Online]. Available: https://www.ietf.org/rfc/rfc2616.txt. [Diakses 12 Februari 2017].