

Integrasi *Framework Kivy* dan *Webix* pada Pembangunan *Framework Mobile Web Easy Development System*

(*Integration Kivy and Webix Framework in Easy Development System Mobile Web Framework*)

Nina Setiyawati¹, Vio Ayu Oktavia Putu Warisman²

^{1,2} *Fakultas Teknologi Informasi Universitas Kristen Satya Wacana*

¹nina.setiyawati@uksw.edu

²vioayuoktavia@gmail.com

Abstrak—Pada dunia retail era sekarang, kemampuan untuk lebih meminimalkan biaya pengeluaran dengan inovasi teknologi modern menjadi salah satu poin penting dalam mendorong kemajuan bisnis. Hal tersebut diakibatkan karena perubahan kecil pada teknologi informasi dapat membantu mempercepat kinerja dan meningkatkan pendapatan. Salah satu perusahaan retail terbesar di Indonesia, PT. Sumber Alfaria Trijaya, Tbk (Alfamart) membangun sebuah *framework* berbasis web menggunakan *framework* Webix pada *Front-End* dan Python pada *Back-End* untuk dapat mengakomodasi perpindahan dari aplikasi server *on-premises* ke *cloud computing*. Namun, beberapa kendala ditemukan seperti dibutuhkannya waktu yang relatif lama ketika diakses melalui peramban *smartphone*, serta kode program dapat dilihat oleh siapapun ketika aplikasi berjalan pada *browser*. Oleh karena itu pada penelitian ini dibangun *framework* yang lebih fleksibel yaitu *Mobile Web Easy Development System* (M-EDS) yang dapat diakses oleh *developer* untuk mengembangkan aplikasi *mobile web* berbasis Android dan iOS, serta pengguna akhir untuk mengakses aplikasi yang dikembangkan. M-EDS dapat mengakomodasi aktivitas pengguna seperti dapat mengakses *Web-Based Framework* dengan satu kali klik. Dari sisi pengembang, Web M-EDS dapat membantu agar kode program aplikasi tidak dapat dilihat oleh pihak yang tidak bertanggung jawab.

Kata-kata Kunci: *Kivy Python, Webix, Mobile Web Framework, Android, iOS*

Abstract—In the retail world recently, the ability to minimize expenses with modern technology innovation is one of the important things to improve business progress. Because a little change in information technology can speed up their activity and increase profit. One of the biggest retail companies in Indonesia, PT. Sumber Alfaria Trijaya, Tbk (Alfamart) has built a web-based Framework using the Webix Framework on Front-End and Python on Back end to accommodate the moving technologies process from server

application on-premises to cloud computing. However, there are some problems found i.e take a long time when access using smartphone browser and everyone can see source code at View Page Source option at their browser. Therefore, in this research a more flexible framework is developed, the Mobile Web Easy Development System (M-EDS) that can be accessed by developers to develop mobile web applications based on Android and iOS, as well as end users to access applications developed. M-EDS can accommodate user activity such as an access Web-Based Framework by one click on their smartphone. And on the developer side, M-EDS helps to hide their source code from irresponsible parties.

Keywords : *Kivy Python, Webix, Mobile Web Framework, Android, iOS*

I. PENDAHULUAN

Cloud Computing merupakan salah satu teknologi yang paling menjanjikan di beberapa tahun belakangan, dengan penjualan yang berpotensi mencapai \$204 Miliar pada akhir 2016. Hampir mayoritas perusahaan yang bergerak dalam bidang teknologi menyediakan *cloud services* seperti *computing cloud* dan *google app engine* [1]. Hal tersebut juga diimplementasikan di PT. Sumber Alfaria Trijaya, Tbk. Hal ini terlihat dari beberapa layanan yang tersedia telah terintegrasi dengan *cloud service* di mana beberapa layanan tersebut didukung dengan kerangka kerja berbasis web yang dibangun oleh PT Sumber Alfaria Trijaya, Tbk, yaitu kerangka kerja EDS (*Easy Development System*) yang berbasis web dengan *framework* Webix sebagai *front-end* dan Python sebagai *back-end*. Selain itu, EDS juga menggunakan PostgreSQL sebagai *database*, dan JSON Web Token (JWT).

Kerangka kerja EDS dibangun dengan tujuan untuk menampung pengolahan data yang diintegrasikan dengan *app engine*, di mana dalam proses pembangunannya mengacu dan memperhatikan kriteria-

kriteria untuk sebuah kerangka kerja, seperti berfokus pada fungsionalitas serta kegunaan sebuah aplikasi serta memperhatikan *user experience* yang dapat dengan mudah dimengerti oleh pengguna. *User experience* merupakan kriteria yang dapat mengoptimalkan pengimplementasian kerangka kerja EDS. Hal ini dikarenakan perspektif pengguna merupakan komponen penting bagi pengembang dalam hal pembangunan sebuah aplikasi yang *user-friendly* [2]. Penggunaan Webix dalam membangun kerangka kerja EDS memberikan manfaat kemudahan dalam membangun antarmuka pengguna yang *user-friendly*. Selain itu, Webix yang merupakan HTML5/CSS3 berbasis UI Toolkit sangat membantu untuk membangun aplikasi web yang kompleks dan dinamik sehingga dapat berjalan pada lintas *browser* bahkan lintas *platform* [3]. Namun, beberapa kendala ditemui antara lain kebutuhan waktu yang relatif lama pada saat pengguna, baik *end-user* maupun *developer*, ketika mengakses kerangka kerja EDS menggunakan *smartphone*. Hal ini disebabkan oleh proses pemuatan data yang dilakukan oleh JavaScript. Oleh karena itu, diperlukan perancangan aplikasi *mobile* untuk mengakses EDS. Adapun kendala lainnya yaitu kurangnya proteksi kode program yang dapat menyebabkan pihak yang tidak bertanggung jawab dapat mengaksesnya secara bebas.

Aplikasi berbasis *mobile* memiliki pertumbuhan pengguna yang terus meningkat. Pada tahun 2018, jumlah aplikasi yang diunduh di Indonesia tercatat 5.087 milyar aplikasi [4]. Aplikasi *mobile* juga memungkinkan pengguna untuk menggunakan teknologi informasi tanpa terikat pada satu lokasi tertentu serta dapat memberikan pengguna kemudahan dengan fleksibilitas, aksesibilitas, dan kemudahan penggunaan lainnya [5]. Beberapa hal di atas menjadi pertimbangan yang besar untuk dapat menyediakan kerangka kerja berbasis *mobile*.

Didasari oleh latar belakang masalah di atas dan referensi pendukung, pada penelitian ini dilakukan perancangan kerangka kerja berbasis *mobile* yang lebih fleksibel dan aman dikarenakan mendukung adanya proteksi kode program agar tidak dapat diakses oleh orang lain. Kerangka kerja tersebut adalah *Mobile Web Easy Development System* (M-EDS). M-EDS bertujuan untuk menampung aktivitas pengguna *end-user* dan *developer* dalam mengelola data dan aplikasi di kerangka kerja EDS menggunakan *smartphone* dengan mudah yang dapat diakses pada sistem operasi Android maupun

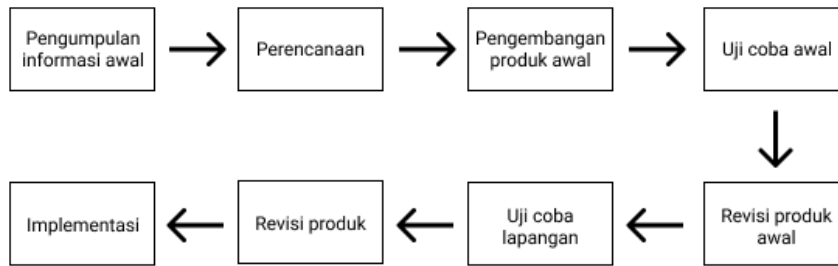
iOS. Kerangka kerja M-EDS dibangun menggunakan kerangka kerja Kivy yang merupakan salah satu kerangka kerja *open-source* [6] serta *cross-platform* [7] yang paling efektif [8]. M-EDS juga mengimplementasikan teknologi *webview* yang terdapat pada *library* Python yaitu *library* Pyjnius untuk Android dan Pyobjus untuk iOS. Pyjnius merupakan modul Python yang berguna untuk mengakses kelas atau *method* pada Java [9][10] dan mengimplementasikannya sebagai kelas atau *method* Python menggunakan *Java Native Interfaces* (JNI) [11]. Adapun Pyobjus merupakan *library* Python yang digunakan untuk mengakses kelas Objective-C menggunakan Objective-C *runtime reflection* [12]. Diharapkan dengan adanya M-EDS, pengguna *end-user* maupun *developer* tidak lagi direpotkan dengan membuka peramban *smartphone* untuk mengakses kerangka kerja EDS.

II. METODE

Model pengembangan yang digunakan dalam penelitian ini merupakan model pengembangan yang dikemukakan oleh Sugiyono yaitu model *Research and Development* (R&D) [13] yang kemudian disesuaikan dengan tujuan dan kondisi penelitian yang sebenarnya. Prosedur yang dilakukan pada penelitian ini dirangkum dalam tahapan seperti terlihat pada Gambar 1.

A. Pengumpulan Informasi Awal

Pada tahap ini dilakukan observasi awal di PT. Sumber Alfaria Trijaya, Tbk dalam hal ini Departemen IT divisi *Business Intelligence* pada proyek kerangka kerja EDS serta *sharing knowledge* dengan karyawan PT. Sumber Alfaria Trijaya, Tbk yang merupakan perancang kerangka kerja EDS sehingga didapatkan informasi pendukung. Dari observasi yang dilakukan, ditemukan permasalahan antara lain kurangnya alternatif untuk mengakses kerangka kerja EDS melalui *smartphone* yang mudah dan fleksibel, kurangnya penjaminan atas keamanan kode program, dan proses menampilkan data ke pengguna *end-user* cukup memakan waktu lama yang diakibatkan lambatnya JavaScript dalam mengembalikan nilai dari *database*. Dari beberapa permasalahan tersebut, dilakukan perancangan *framework Mobile Web Easy Development System* (M-EDS).



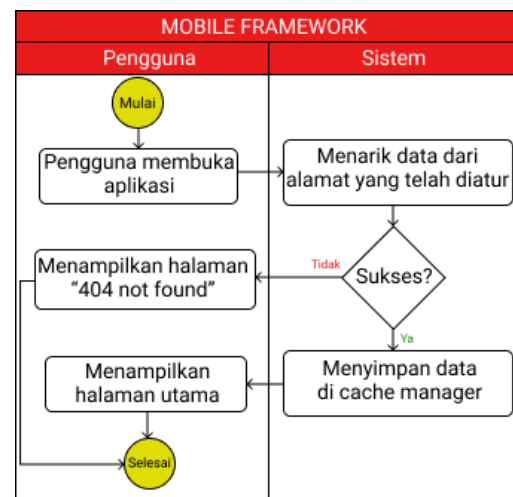
Gambar 1. Model research and development

B. Perencanaan

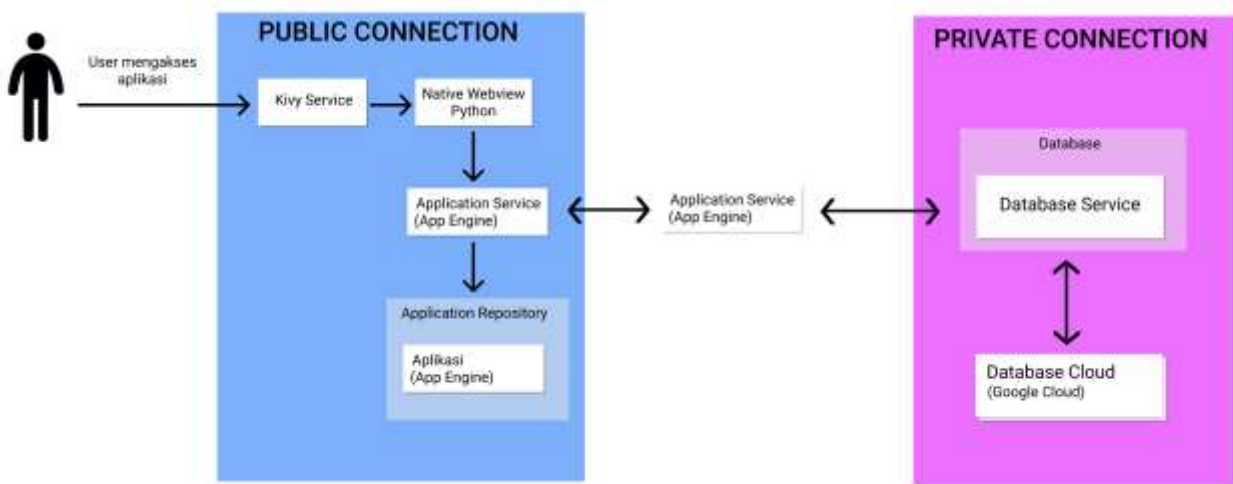
Pada tahap ini dilakukan perancangan *framework* M-EDS yang dapat berjalan pada *mobile platform* dengan sistem operasi Android dan iOS yang responsif dan interaktif. Adapun manfaat dari M-EDS dari sisi pengguna akhir adalah mempersingkat waktu dalam membuka aplikasi yang telah dikembangkan menggunakan M-EDS tanpa harus menggunakan peramban pada *smartphone*, serta dapat mengelola aktivitas pada kerangka kerja dimanapun dan kapanpun. Selain itu, M-EDS juga memproteksi kode program JavaScript yang digunakan agar tidak dapat dilihat atau diakses pihak yang tidak bertanggung jawab. *Flowchart* sistem secara umum dan arsitektur M-EDS pada sisi *end-user* dapat dilihat pada Gambar 2 dan Gambar 3.

Gambar 2 merupakan *flowchart* sistem M-EDS yang menjelaskan gambaran umum aplikasi berjalan. Dimulai dengan penarikan data pada aplikasi web EDS yang kemudian ditampilkan pada aplikasi *mobile* M-EDS ketika pengguna mengakses aplikasi M-EDS. Apabila alamat yang dikirim salah, maka aplikasi M-EDS akan menampilkan pemberitahuan halaman tidak dapat

ditemukan. Namun apabila sebaliknya, maka pengguna dapat mengakses M-EDS dan aktivitas yang dilakukan oleh pengguna akan tersimpan sebagai *cache*. *Cache* tersebut kemudian akan digunakan oleh sistem sebagai data untuk memuat aplikasi M-EDS tanpa menarik data lagi dari *server*.



Gambar 2. Flowchart sistem M-EDS aplikasi mobile web



Gambar 3. Arsitektur sistem pada aplikasi M-EDS *end-user*

Gambar 3 menampilkan arsitektur M-EDS yang terintegrasi dengan *app engine* dan *cloud service*. Adapun *public connection* dalam arsitektur sistem M-EDS adalah koneksi yg memungkinkan untuk diakses oleh karyawan PT. Sumber Alfaria Trijaya yang mempunyai *username* dan *password* sedangkan untuk *private connection* adalah koneksi *server* yang hanya dapat diakses oleh karyawan PT. Sumber Alfaria Trijaya yang memiliki alamat IP dari setiap *service*. Pertama kali pengguna mengakses aplikasi M-EDS *end-user*, pengguna secara otomatis memulai proses *loading* oleh Kivy yang mengeksekusi *native webview* yang meminta *request* ke aplikasi web EDS.

Halaman pertama yang akan dimunculkan berupa daftar aplikasi yang telah berhasil dibangun menggunakan M-EDS *developer*. Proses tersebut berjalan dengan melakukan penarikan data dari *app engine* dan pengecekan pada konfigurasi aplikasi yang tersimpan dalam *database*. Pada aplikasi aktif yang terlihat pada halaman pertama ketika dipilih oleh pengguna, sistem akan melakukan *request* ke *database cloud* dengan bantuan *service* yang tersedia di *compute engine* dan menggunakan koneksi pribadi. Data balikan akan dikirim oleh *database* berupa daftar menu aplikasi, isi kolom dan baris pada *table/form*.

C. Pengembangan Produk Awal

Pada tahap pengembangan produk awal dilakukan dengan membangun M-EDS. Pada M-EDS dikategorikan menjadi dua modul, yaitu M-EDS *end-user* dan M-EDS *developer*. M-EDS *developer* menampung aktivitas yang dilakukan oleh pengguna *developer* untuk membangun aplikasi atau *form* yang nantinya akan dioperasikan oleh pengguna *end-user*. Pembangunan M-EDS dilakukan dengan menggunakan kerangka kerja Kivy, *library* Pyjnius dan Pyobjus, Webix yang terintegrasi dengan *App Engine* dan menggunakan Postgre sebagai *database*.

D. Uji Coba Awal

Uji coba awal dilakukan dengan pengujian desain, pencarian *bug* dan pengujian *usability framework* M-EDS oleh karyawan PT. Sumber Alfaria Trijaya, tbk. Pada tahap ini memberikan hasil berupa beberapa revisi yang mengharuskan penggantian teknologi yang digunakan, penambahan fungsi untuk menyimpan data di *storage internal* dan masih terdapat permasalahan berupa waktu yang lama dalam memuat data oleh JavaScript. Dengan hasil tersebut maka dilakukan pencarian pada beberapa literatur.

E. Revisi Produk Awal

Revisi produk atau aplikasi, menghasilkan beberapa perbaikan berupa pergantian teknologi *Webview*, yang pada awalnya direncanakan menggunakan teknologi *Chrome Embedded Framework* (CEF) dan Kivy, menjadi penggunaan *Native Webview* pada Kivy dengan menggunakan *library* Pyjnius untuk sistem operasi Android dan menggunakan *library* Pyobjus untuk sistem operasi iOS.

F. Uji Coba Lapangan

Pada tahap ini dilakukan pengujian *framework* dalam kondisi yang sesungguhnya yaitu digunakan untuk membuat suatu aplikasi berbasis Android dan iOS, serta untuk menggunakan aplikasi tersebut. Uji coba melibatkan karyawan PT. Sumber Alfaria Trijaya, tbk.

G. Revisi Produk

Pada tahap ini dilakukan penambahan atau perbaikan pada *framework* M-EDS berdasarkan hasil uji coba lapangan yang dilakukan. Adapun penambahan yang dilakukan adalah penambahan *storage internal* menggunakan *cache manager* yang telah tersimpan sebagai media penyimpanan tingkah laku pengguna.

H. Implementasi

Implementasi aplikasi terkendala dengan keterbatasan JavaScript dalam memuat data sehingga aplikasi tidak diimplementasikan secara masal.

III. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan *framework* yang dapat digunakan untuk membangun *mobile web application* berbasis Android dan iOS dengan teknologi *native webview* Java dan Objective-C yang terdapat pada *library* Python. Teknologi yang digunakan untuk membangun sebuah *webview* yang mampu berjalan pada Platform Android menggunakan bahasa pemrograman Python, antara lain: Virtualbox Oracle Versi 6.1.2, Kivy Python 3.7.6, Linux Image File, Bulldozer 1.0. Sedangkan untuk membangun *webview* yang mampu berjalan pada Platform iOS menggunakan bahasa pemrograman Python, antara lain: OS X Mojave (14.3) atau diatasnya, Xcode versi 10.2 atau di atasnya, Kivy-iOS Python 3.7.6.

Dengan *system requirements* di atas, pembangunan M-EDS menggunakan *native webview* Java pada Android menghasilkan aplikasi (.apk) dengan ukuran yang normal yaitu 8 megabyte. Hal tersebut dikarenakan *library* yang digunakan tidak banyak sehingga dapat menghemat penyimpanan *smartphone*. M-EDS Android dibangun menggunakan kode program yang terlihat pada Gambar 4.

```

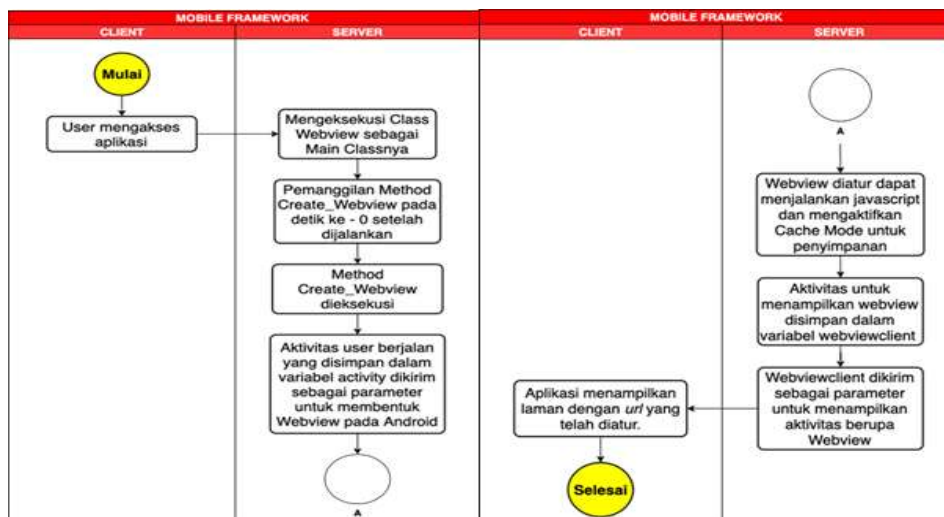
1. def create_webview(self, *args) :
2. webview = WebView(activity)
3. webview.getSettings().setJavaScriptEnabled(True)
4. webview.getSettings().setJavaScriptCanOpenWindowsAutomatically(True)
5. webview.getSettings().setDomStorageEnabled(True)
6. webview.getSettings().setCacheMode(WebSettings.LOAD_DEFAULT)
7. wvc = WebViewClient()
8. webview.setWebViewClient(wvc)
9. activity.setContentView(webview)
10. webview.loadUri('Your Link URL')

```

Gambar 4. Main function M-EDS android

Gambar 4 merupakan potongan kode yang digunakan sebagai proses utama pembangunan *Webview* yang berjalan pada *mobile platform* Android menggunakan bahasa pemrograman Python dengan *Library* Pyjnius. Pada baris 2, variabel '*webview*' digunakan sebagai inisialisasi untuk proses *Webview* dengan variabel '*activity*' sebagai parameter. Selanjutnya untuk baris 3 hingga 6 digunakan sebagai pengaturan dalam *Webview* menggunakan *method* '*getSettings*' yang terdapat pada *class* *WebSettings*. Untuk pengaturan pada baris 3 dan 4, berfungsi untuk mendukung penggunaan *JavaScript*. Secara *default*, *JavaScript* dinonaktifkan pada *widgets* *Webview*. Sehingga halaman web yang berisikan *JavaScript* tidak akan berfungsi dengan baik. Oleh karena itu, diperlukan pemberian izin pada *Webview* agar dapat memuat *JavaScript* yang terletak pada baris 3. Sedangkan untuk baris ke 4 berfungsi agar *JavaScript* dapat membuka *window* secara otomatis. Baris 5 berfungsi agar *webview* menggunakan *DOM (Document Object Model)* *Storage* atau *web storage* sebagai penyimpanan *client-side* data sehingga tidak ada

informasi yang tersimpan dalam *HTTP header*. Baris 6 digunakan untuk mengganti cara penggunaan *cache*. Secara *default*, cara penggunaan *cache* berdasarkan tipe navigasi. Untuk proses normal dalam memuat halaman web, *cache* akan dicek dan konten dilakukan validasi kembali sesuai kebutuhan. Ketika navigasi kembali, konten tidak akan dilakukan validasi kembali dan hanya akan memuat data dari *cache*. Dengan *value* '*LOAD_DEFAULT*' berarti *cache* dimuat dengan cara standar yaitu jika tipe navigasi tidak menggunakan tingkah laku spesifik lainnya maka akan memuat dari *cache* yang telah tersimpan dan tidak kadaluwarsa namun jika sebaliknya maka akan memuat data dari jaringan. Baris 7 merupakan inisialisasi *class* *WebViewClient*. Baris 8 berfungsi untuk menangani aktivitas yang berpengaruh pada konten *rendering*, seperti *error* dari navigasi dan digunakan untuk menangkap *loading URL*. Baris 9 digunakan untuk memuat URL ke *webview*. Adapun *flowchart* sistem *webview* terlihat pada Gambar 5.



Gambar 5. Detail flowchart Sistem webview

Dalam proses *packaging* aplikasi hingga berbentuk .apk, M-EDS menggunakan teknologi Buildozer Kivy dan diperlukan sistem operasi seperti *requirements* awal. Selain standar ketika inisialisasi buildozer.spec, pada pembangunan M-EDS juga dilakukan konfigurasi pada file buildozer.spec khususnya untuk proses *packaging* aplikasi (.apk). Selain menghasilkan aplikasi *mobile* web berbasis Android, juga dihasilkan aplikasi *mobile* web berbasis iOS. Gambar 6 adalah kode program daftar variabel yang digunakan dalam pembuatan Webview pada *mobile application* menggunakan Kivy Python untuk iOS menggunakan library Pyobjus. Adapun tampilan halaman *login* M-EDS terlihat pada Gambar 7.

Gambar 7 adalah halaman *login* untuk *end-user* pada aplikasi *Development Compiler*. Untuk pengguna *end-user* adalah karyawan PT. Sumber Alfaria Trijaya, tbk yang berperan untuk mengoperasikan aplikasi yang dibangun menggunakan modul *Developer* M-EDS. Sedangkan pengguna *developer* adalah karyawan PT. Sumber Alfaria Trijaya, tbk yang berperan untuk membuat aplikasi yang akan diakses oleh pengguna *end-user*. Contoh aplikasi yang telah sukses dibangun menggunakan aplikasi *Developer* M-EDS terlihat pada Gambar 8 dan 9.

```
1. WebView = autoclass('WKWebView')
2. WebViewConfig = autoclass('WKWebViewConfiguration')
3. NSURL = autoclass('NSURL')
4. NSURLRequest = autoclass('NSURLRequest')
5. UIApp = autoclass('UIApplication')
```

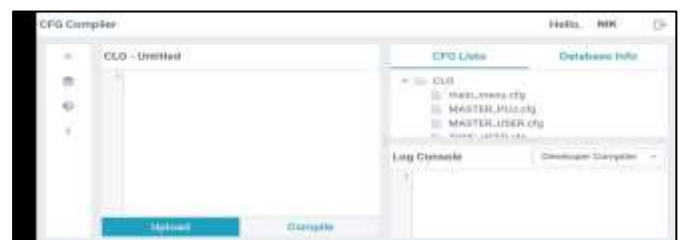
Gambar 6. Variabel Python file (.py) webview iOS



Gambar 7. Halaman login developer dan end-user iOS



Gambar 8. Halaman android M-EDS end-user



Gambar 9. Halaman development M-EDS developer

Gambar 8 merupakan *dashboard* pada halaman *End-user* yang menampilkan daftar menu aplikasi. Tampilan pada sebelah kiri yang di kotak merah merupakan menu yang terdaftar dan sukses dilakukan *compile* dan *deploy* ke *app-engine* oleh pengguna *developer*.

Gambar 9 merupakan *dashboard* pada halaman *Development* yang digunakan sebagai *text editor* oleh pengguna *developer* untuk membangun aplikasi (*form*). Untuk membangun sebuah aplikasi bagi *end-user*, *developer* mengembangkannya dengan menggunakan konfigurasi file (.cfg). Aplikasi yang telah berhasil dibuat setelah melalui proses *compile* dan *deploy* akan muncul pada halaman *end-user*. Tampilan pada Gambar 9 didokumentasikan setelah dilakukan rotasi pada *smartphone* sehingga orientasi menjadi *landscape*. Konfigurasi *rotation* aplikasi M-EDS diatur dalam file *buildozer.spec*.

Setelah *framework* M-EDS selesai dibangun, dilakukan pengujian dengan penyebaran kuesioner sebanyak 20 responden yang merupakan karyawan PT. Sumber Alfaria Trijaya, tbk yang terdapat pada departemen IT divisi *Business Intelligence*. Adapun hasil dari pengujian aplikasi dapat dilihat pada Tabel I dengan penjelasan pada Tabel II.

TABEL I
PENGUJIAN APLIKASI *MOBILE* WEB M-EDS BERDASARKAN WEBQUAL 4.0

No	Pertanyaan	Jawaban					Jumlah	Indeks
		Ax5	Ax4	Ax3	Ax2	Ax1		
USABILITY								
1	Modul <i>Mobile</i> Web M-EDS membantu membuka kerangka kerja secara mudah, kapanpun dan dimanapun.	4	6	0	0	0	44	88%
2	Modul <i>Mobile</i> Web M-EDS dapat mempercepat proses mengakses data dibandingkan dengan peramban pada <i>smartphone</i> .	2	8	0	0	0	42	84%
INFORMATION QUALITY								
3	Aplikasi <i>Mobile</i> Web M-EDS dapat meningkatkan kinerja karyawan dalam segi efektivitasnya.	7	3	0	0	0	47	94%
SERVICE INTERACTION								
4	Pengguna Aplikasi <i>Mobile</i> Web M-EDS tidak dapat melihat kode program aplikasi.	10	0	0	0	0	50	100%

TABEL II
PENJELASAN KODE JAWABAN TABEL I

Kode Jawaban	Deskripsi	Nilai
Ax5	Sangat Setuju	5
Ax4	Setuju	4
Ax3	Normal	3
Ax2	Tidak Setuju	2
Ax1	Sangat Tidak Setuju	1

Mengacu pada perolehan jawaban kuesioner, kolom Jumlah dan Indeks pada Tabel I diterapkan persamaan (1) dan persamaan (2)[14].

$$Jumlah = \sum(p \times i) \quad (1)$$

Pada persamaan (1), p adalah jumlah responden setiap kategori jawaban dan i adalah nilai jawaban di setiap kategori jawaban.

$$Indeks = \frac{y}{x} \times 100\% \quad (2)$$

Pada persamaan (2), y adalah jumlah pada setiap pertanyaan, di mana didapatkan dengan perhitungan

menggunakan Persamaan (1). Sedangkan x adalah nilai skala terbesar dikalikan dengan jumlah responden. Berdasarkan pengujian yang dilakukan pada aplikasi *mobile* web M-EDS dengan menerapkan metode Webqual 4.0 dengan statistika deskriptif yang terbagi dalam 3 kategori yaitu variabel *usability*, *information quality*, dan *service interaction* [15][16], didapatkan bahwa pada variabel *usability* pengguna setuju modul *Mobile* Web M-EDS membantu membuka kerangka kerja secara mudah, kapanpun dan dimanapun. Selain itu pengguna juga setuju bahwamodul *Mobile* Web M-EDS dapat mempercepat proses pengaksesan data. Dari variabel *information quality* pengguna setuju bahwa aplikasi *Mobile* Web M-EDS dapat meningkatkan kinerja karyawan dalam segi efektivitas, dan dari variabel *service interaction* didapatkan bahwa pengguna aplikasi *Mobile* Web M-EDS tidak dapat melihat kode program aplikasi. Hal ini menunjukkan bahwa *framework* yang dibangun menjamin keamanan kode program dibanding dengan *framework* sebelumnya.

IV. PENUTUP

Pada penelitian ini dilakukan perancangan *framework* M-EDS menggunakan kerangka kerja Kivy dan Webix. M-EDS memiliki dua pengguna yaitu pengguna *developer* yang dapat memanfaatkan M-EDS untuk mengembangkan *mobile web application* berbasis Android dan iOS serta pengguna *end user* yang mengelola aplikasi yang dibangun oleh *developer* menggunakan kerangka kerja M-EDS. Berdasarkan pengujian aplikasi pada Tabel I dapat disimpulkan bahwa modul M-EDS *End User* dapat mempermudah pengguna dalam mengakses dan menggunakan aplikasi yang telah dikembangkan menggunakan M-EDS. M-EDS juga dapat menghasilkan aplikasi yang lebih fleksibel sehingga meningkatkan kinerja karyawan dalam sisi efektivitasnya. Pada sisi pengguna *developer*, aplikasi M-EDS dapat membantu proteksi kode program JavaScript.

DAFTAR PUSTAKA

- [1] H. Alzahrani, "A Brief Survey of Cloud Computing," *Glob. J. Comput. Sci. Technol.*, vol. 16, no. 3, p. 7, 2016.
- [2] H. Heitkötter, T. A. Majchrzak, B. Ruland, and T. Weber, "Evaluating frameworks for creating mobile web apps," *WEBIST 2013 - Proc. 9th Int. Conf. Web Inf. Syst. Technol.*, pp. 209–221, 2013, doi: 10.5220/0004356702090221.
- [3] Frank Zammetti, *Practical Webix: Learn to Expedite and Improve your Web Development*. Pottstown, USA: Apress, Berkeley, CA, 2018.
- [4] We Are Social and Hootsuite, "Digital DATA OVERVIEW 2019: Indonesia," *Glob. Digit. Insights*, p. 17, 2019, [Online]. Available: <https://datareportal.com/reports/digital-2020-indonesia>.
- [5] V. Maheshwar, K. Patil, A. Maredia, and A. Waghmare, "Android Application on E-Ticketing Railway System Using Qr-Code," *IOSR J. Eng.*, vol. 13, pp. 33–38, 2018.
- [6] H. Dawkins, "Colorado Technical University SYMBOLIC EXECUTION FOR KIVY APPLICATIONS Chapters 1 to 5 Dr. Danette Lance A Dissertation Presented in Partial Fulfillment of the Requirements for the Degree of Doctor of Management or Computer Science By Herbert Dawkins Col," no. July, 2015.
- [7] B. O. Chrastina, "Cross-Platform Development of Smartphone Application With The Kivy Framework," *MASARYK UNIVERSITY*, 2015.
- [8] A. Bhoyarkar, A. Solanki, and A. Balbudhe, "Application Development using Kivy Framework," *Ijarccce*, vol. 8, no. 2, pp. 53–58, 2019, doi: 10.17148/ijarccce.2019.8209.
- [9] B. M. Gyori, J. A. Bachman, and J. L. Muhlich, "INDRA Documentation," 2016, [Online]. Available: <https://readthedocs.org/projects/indra/downloads/pdf/latest/>
- [10] Z. Akkalyoncu Yilmaz, S. Wang, W. Yang, H. Zhang, and J. Lin, "Applying BERT to Document Retrieval with Birch," pp. 19–24, 2019, doi: 10.18653/v1/d19-3004.
- [11] A. Taylor, "python-for-android Documentation," 2020, [Online]. Available: <https://buildmedia.readthedocs.org/media/pdf/python-for-android/latest/python-for-android.pdf>.
- [12] M. Virbel and G. Pettier, "Pyobjus Documentation," 2018, [Online]. Available: https://pyobjus.readthedocs.io/_/downloads/en/latest/pdf
- [13] Sugiyono, *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung: Alfabeta, 209AD.
- [14] S. Julika and D. Setiyawati, "Hubungan antara Kecerdasan Emosional, Stres Akademik, dan Kesejahteraan Subjektif pada Mahasiswa," *Gadjah Mada J. Psychol.*, vol. 5, no. 1, p. 50, 2019, doi: 10.22146/gamajop.47966.
- [15] J. F. Andry, K. Christianto, and F. R. Wilujeng, "Using Webqual 4.0 and Importance Performance Analysis to Evaluate E-Commerce Website," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 5, no. 1, p. 23, 2019, doi: 10.20473/jisebi.5.1.23-31.
- [16] H. Aryadita, D. A. Widyastuti, and N. H. Wardani, "Analisis Kualitas Layanan Website E-Commerce Terhadap Kepuasan Pengguna Menggunakan Metode Webqual 4.0," *Anal. Kualitas Layanan Website E-Commerce Terhadap Kepuasan Pengguna Menggunakan Metod. Webqual 4.0*, vol. 10, no. 1, pp. 29–35, 2017, doi: 10.15408/sijsi.v10i1.7747.