



Pemanfaatan Google App Script dalam Merancang Aplikasi Web Guna Meningkatkan Efisiensi Biaya Perusahaan

Tony Wijaya
STMIK Pontianak

Jl. Merdeka No. 372 Pontianak, Kalimantan Barat

Korespondensi penulis: tony_wijaya@stmkpontianak.ac.id

Abstract. Mobile applications trend are currently happening due to increased number of people holding smartphones. But this does not mean the end of web and desktop users. Mobile, web and desktop apps will always have its own market and users in manipulating data. The challenges for software developers are how to deliver software to all of those platforms as fast as possible. Efficiency is key in this research, that is to develop a hybrid app. Black box testing will be used in this research to make sure that efficiency is gained throughout the process compared to developing for each platforms independently. Through this research, hopefully can help software developers to develop and deliver applications for user efficiently in a short amount of time.

Keywords: Hybrid app, time to market, efficiency.

Abstrak. Merekayasa sebuah situs web dan mengimplementasikan di hosting ataupun VPS sudah biasa dilakukan. Biayanya pun bervariasi dari yang cukup terjangkau hingga yang mahal. Penelitian ini akan melakukan strategi yang berbeda, tentunya dengan skema biaya yang berbeda pula. Penggunaan Google App Script dengan hanya mengandalkan akun Google gratis sudah bisa menghasilkan sebuah aplikasi web sederhana. Efisiensi biaya adalah kata kunci pada penelitian ini, yaitu dengan membangun aplikasi web tanpa biaya sama sekali. Pengujian dilakukan dengan metode black box untuk membuktikan fungsionalitas aplikasi web yang dihasilkan, apabila dibandingkan dengan aplikasi web yang biasa diimplementasikan di hosting. Melalui penelitian ini diharapkan dapat membantu pengembang perangkat lunak dalam mengembangkan aplikasi secara efisien dalam biaya dengan memanfaatkan akun Google yang gratis.

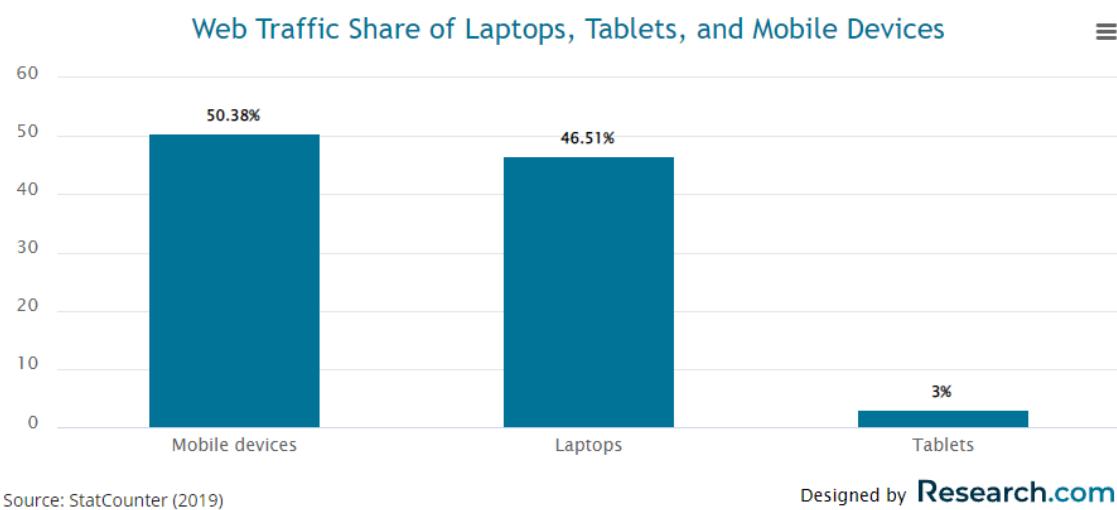
Kata kunci: Hybrid app, time to market, efficiency.

LATAR BELAKANG

Trend penggunaan smartphone dari tahun 2019 hingga 2021 khususnya di Indonesia selalu meningkat (Statistik, 2023). Sejak tahun 2019 pun lalu lintas internet sudah dikuasai oleh smartphone, apabila dibandingkan dengan desktop (pc / laptop) dan tablet (Bouchrika, 2022). Pengguna smartphone jauh lebih banyak daripada pengguna desktop karena bentuknya yang relatif kecil sehingga mudah dibawa ke mana saja. Smartphone juga dapat digunakan untuk

komunikasi seperti telepon dan SMS, sedangkan desktop sulit melakukan hal tersebut. Harga smartphone pun relatif lebih murah daripada desktop sehingga lebih terjangkau oleh konsumen, bahkan seorang konsumen dapat memiliki beberapa smartphone, namun tidak demikian sebaliknya. Jarang seorang konsumen memiliki lebih dari 1 unit PC ataupun laptop.

Sebagai pengembang / developer perangkat lunak, kita harus siap dengan kondisi seperti ini. Bertambahnya platform baru ini tentunya menjadi tantangan tersendiri, karena sebelumnya, sudah ada 2 platform yang sering digunakan oleh pemakai yaitu platform web dan desktop. Kedua platform ini sudah cukup merepotkan pengembang perangkat lunak karena banyaknya pilihan teknologi yang ada. Untuk desktop, beberapa pilihan teknologi yang cukup dikenal adalah Visual Basic .NET, C# .NET dan Java 2 SE. Sedangkan untuk platform web, ada PHP dengan banyak framework pilihan di antaranya seperti Laravel, Yii, CodeIgniter dan Zen Framework; ASP.NET dengan berbagai varian framework terkini seperti ASP.NET Core, ASP.NET Core MVC dan ASP.NET Core Web API; Java dengan Java 2 Enterprise Edition; NodeJS dengan beberapa framework yang terkenal seperti ExpressJS; Python dengan framework Django, Ruby dengan framework Ruby on Rails, dan lain-lainnya. Dengan bertambahnya platform mobile, muncul lagi beberapa teknologi yang perlu dipelajari oleh pengembang perangkat lunak seperti : Android Studio untuk system operasi Android dan XCode untuk iOS.



Gambar 1. Lalu Lintas Internet dari Laptop, Tablet dan Smartphone.

KAJIAN TEORITIS

B Hadirnya platform mobile tentunya tidak akan menggeser kedua platform tersebut dalam waktu dekat. Oleh karena itu, tugas developer menjadi semakin banyak dan rumit karena

harus mempelajari 3 platform. Demikian pula waktu penggerjaannya atau terkenal dengan istilah time-to-market yang tentunya akan semakin lama karena bertambahnya platoform yang harus di-coding.

Pada tahun 2023 ini sudah ada beberapa framework yang dapat menghasilkan aplikasi multi-platform. Dengan 1 (satu) kali pengkodean, pengembang perangkat lunak akan dapat menghasilkan aplikasi web, desktop dan mobile sekaligus. Framework-framework ini di antaranya adalah Flutter (Flutter - Build apps for any screen, n.d.), React Native (React Native - Learn once, write anywhere, 2023), Cordova (Apache Cordova, 2023) dan Ionic (Ionic Framework - The Cross-Platform App Development Leader, 2023). Dengan beberapa framework yang kini sedang banyak digemari oleh para pengembang perangkat lunak ini, aplikasi dapat dihasilkan dengan time-to-market yang jauh lebih singkat apabila para pengembang perangkat lunak harus mengembangkannya ke dalam beberapa platform. Namun dengan kelebihan seperti ini, bukan berarti tidak ada kekurangannya. Sebagai konsekuensi dari multi-platform, maka framework-framework seperti di atas menghasilkan aplikasi yang memiliki performa yang lebih rendah, pemakai memori yang jauh lebih tinggi, ukuran aplikasi yang jauh lebih besar serta kemampuan mengakses hardware yang terbatas.

Aplikasi hybrid sedikit berbeda dengan framework-framework multi platform di atas. Aplikasi hybrid merupakan kombinasi antara aplikasi native (desktop & mobile) dan web. Aplikasi ini memiliki semua kelebihan native (desktop dan mobile) dan web. Apabila dibandingkan, maka kelemahannya lebih sedikit daripada framework multi platform. Aplikasi hybrid memang tidak memiliki performa sebaik aplikasi native, dan pemakaian RAM juga lebih besar. Namun ukuran aplikasinya sangat kecil, bahkan lebih kecil daripada aplikasi native. Hal ini disebabkan karena aplikasi hybrid Sebagian besar hanya berisi WebView. Selain itu, aplikasi hybrid juga dapat mengakses semua hardware pada perangkat layaknya seperti aplikasi native. Hal ini disebabkan karena aplikasi hybrid merupakan gabungan antara aplikasi native dan web. Jadi apabila dibandingkan dengan framework multi platform, maka aplikasi hybrid jauh lebih menguntungkan.

Objek penelitian ini adalah Sistem Order Sales CV. Surya Putra (SOS Surya Putra). Aplikasi hybrid ini merupakan perpaduan antara aplikasi web dan aplikasi mobile Android. Aplikasi ini berisi modul-modul seperti : salesman order, tagihan sales dan pencapaian target salesman.

Modul yang akan dibahas adalah modul salesman order yang berfungsi untuk merekam orderan salesman dari luar kota sehingga langsung dapat diakses oleh admin di kantor pusat CV. Surya Putra Pontianak. Data ini diinput oleh masing-masing salesman dengan smartphone mereka masing-masing. Penelitian ini akan membahas bagaimana:

- Membuat aplikasi wrapper dengan Android Studio
- Mengkoneksikan aplikasi web dengan aplikasi wrapper di Android Studio
- Melakukan coding JavaScript untuk mengakses fungsi dari Java (Android)
- Melakukan pengukuran performa aplikasi hybrid
- Melakukan pengukuran ukuran aplikasi hybrid
- Melakukan pengukuran konsumsi RAM aplikasi hybrid

Penelitian ini ingin membuktikan bahwa aplikasi hybrid mampu meningkatkan produktivitas pengembangan perangkat lunak dibandingkan ketika membangun aplikasi secara native untuk setiap platform (web, desktop dan mobile).

METODE PENELITIAN

Penelitian ini dilakukan dengan pendekatan kualitatif (Purhantara, 2010). Objek penelitian adalah Sistem Order Sales CV. Surya Putra (SOS Surya Putra). Instrumen penelitian yang digunakan dalam penelitian ini adalah:

- Smartphone Xiaomi Redmi Note 8 Pro dengan sistem operasi Android 10.
- Android Studio versi 2022.1.1 (Electric Eel) (Download Android Studio & App Tools - Android Developers, 2023)
- Aplikasi web SOS Surya Putra.
- Web service / web API Surya Putra.

Kesimpulan dibuat berdasarkan interpretasi data oleh peneliti. Bentuk penelitian adalah penelitian eksperimen dengan objek yaitu aplikasi web SOS Surya Putra.

Metode Agile (Albarqi, 2018) digunakan dalam perancangan perangkat lunak pada penelitian ini. Prinsip dasar Agile tertuang dalam Agile Manifesto for Software Development (Beedle, et al., 2001). Daripada hanya sekedar mengikuti rancangan UML, metode ini juga

dapat merespon dengan cepat terhadap perubahan. Hal ini relevan karena penelitian ini dilakukan dengan bentuk eksperimen.

Extreme Programming digunakan sebagai pendekatan perancangan perangkat lunak. Extreme Programming merupakan disiplin dari pengembangan perangkat lunak yang berdasar pada nilai-nilai kesederhanaan, komunikasi, umpan balik, dan keberanian (Lindstrom & Jeffries, 2004). Yang membedakan XP dengan pendekatan perancangan perangkat lunak lainnya adalah pada pendekatan ini selalu melibatkan semua anggota tim, misalnya rapat dengan stake holder dan pair programming (coding bersama). Jadi setiap anggota tim terlibat dalam semua aktifitas dan tidak ada pembagian tugas. Tanggung jawab dalam XP relatif sama, mulai bekerja pada jam yang sama, selesai pada jam yang sama pula, teknik menulis program juga sama, dan lain sebagainya. Pada penelitian ini cocok menggunakan pendekatan ini karena hanya dilakukan oleh 1 orang saja. Karena dalam XP, setiap anggota tim juga melakukan semua pekerjaan tanpa ada pembagian tugas.



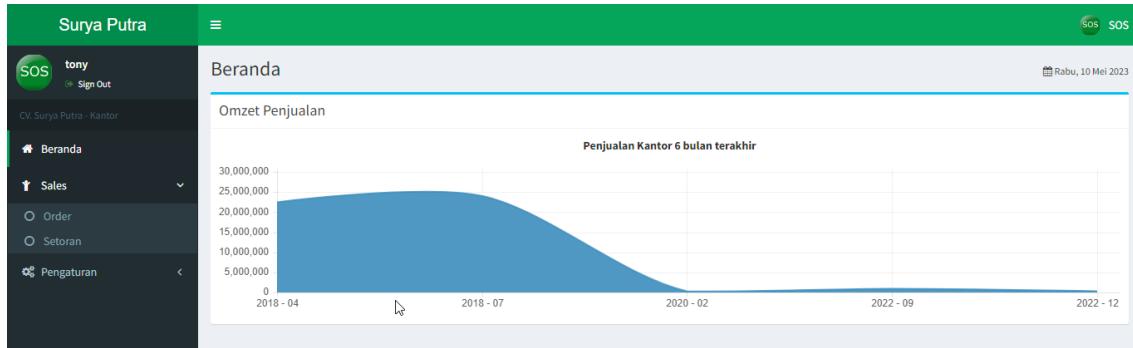
Gambar 2. Siklus Hidup Extreme Programming.

Aplikasi SOS Surya Putra dibangun di atas framework CodeIgniter (CodeIgniter Web Framework, n.d.) dengan theme dari Admin LTE versi 3 (Free Bootstrap Admin Template, 2022) menggunakan bahasa pemrograman PHP (PHP: Hypertext Preprocessor, 2023) dan dikode dengan bantuan IDE Visual Studio Code (Microsoft, 2021). Server basis data ada di cloud yang berupa Virtual Private Server (VPS) dan menggunakan sistem operasi Ubuntu 20.04 LTS (Long Time Support) (The leading operating system for PCs, IoT devices, servers and the cloud, n.d.). Basis data yang digunakan adalah MariaDB 10.4.21 yang merupakan basis data yang bersifat open source (Supporting continuity and open collaboration, 2019). MariaDB

digunakan karena sangat mendukung dengan sistem operasi yang digunakan yaitu Ubuntu sehingga menghasilkan performa yang sangat tinggi.

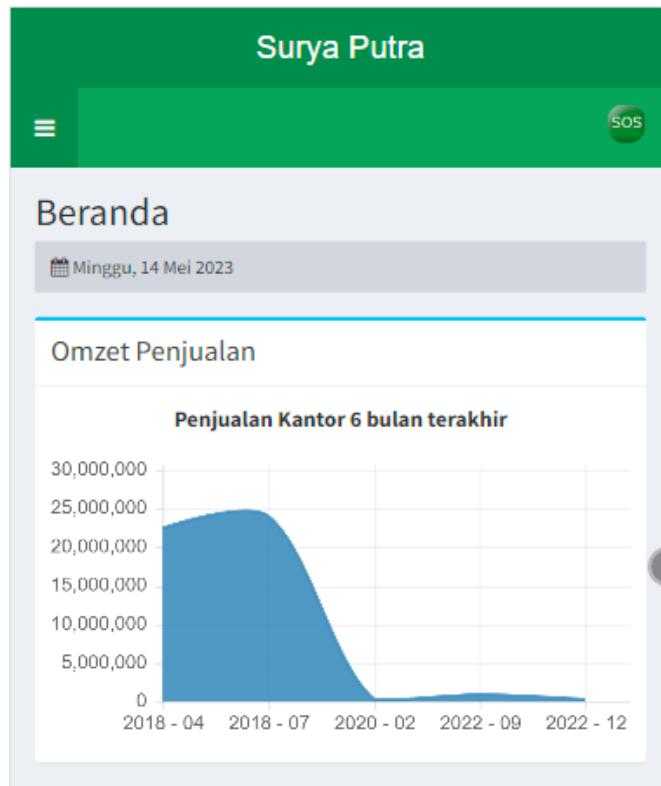
HASIL DAN PEMBAHASAN

Tampilan halaman sign in aplikasi web Mahasiswa dapat dilihat pada gambar di bawah ini:



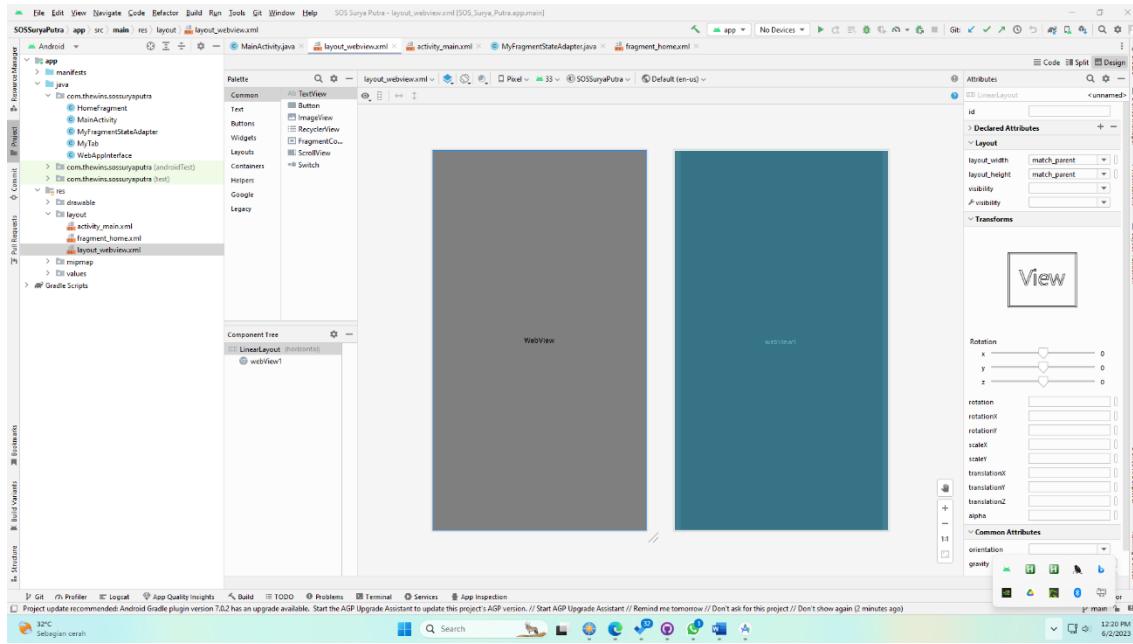
Gambar 3. Laman Dashboard aplikasi web “SOS Surya Putra” melalui peramban.

Gambar di atas menunjukkan halaman dashboard aplikasi web “SOS Surya Putra” yang dibangun dengan menggunakan framework CodeIgniter. Ketika diakses dengan perangkat mobile seperti smartphone Android atau iPhone, maka tampilannya akan terlihat seperti berikut :



Gambar 4. Laman “Dashboard” aplikasi web “SOS Surya Putra” pada perangkat mobile.

Tampilan laman dashboard di atas masih berupa tampilan dari peramban, belum merupakan sebuah aplikasi mobile. Namun aplikasi web di atas dapat diubah menjadi aplikasi mobile yang native di sistem operasi Android dengan cepat hanya dengan memubat aplikasi wrapper menggunakan Android Studio. Aplikasi wrapper memiliki artian bahwa aplikasi native ini hanya berisi WebView yang meramban ke aplikasi web yang sama sehingga pemakai tidak perlu memasukkan alamat URL. Hal ini juga menjadi keuntungan dari sisi keamanan karena pemakai tidak memiliki akses untuk melakukan phising. Dari sisi user experience (UX), pemakai akan dimudahkan karena tidak perlu mengingat alamat URL.



Gambar 5. Tampilan project SOS Surya Putra dalam Android Studio.

Gambar di atas menunjukkan tampilan project SOS Surya Putra menggunakan IDE Android Studio. Project ini memiliki struktur file yang cukup sederhana karena hanya merupakan 10% dari total kode project SOS Surya Putra, di mana 90% sisanya berupa kode framework CodeIgniter. Listing program untuk MainActivity.java dapat dilihat pada gambar berikut :

The screenshot shows the Android Studio interface with the following details:

- File Path:** SOSuryaPutra > app > src > main > java > com.thewins.sosuryaputra > MainActivity
- Code Editor:** The editor displays the `MainActivity.java` file. The code implements a `TabLayout` with three tabs, each corresponding to a different URL. It uses a `ViewPager` to handle the content for each tab.
- Project Structure:** The left sidebar shows the project structure with packages like `com.thewins.sosuryaputra`, `com.thewins.sosuryaputra (androidTest)`, and `com.thewins.sosuryaputra (test)`.
- Toolbars:** Standard Android Studio toolbars for File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, Window, Help, and Git.
- Bottom Navigation:** Includes tabs for Structure, Logcat, App Quality Insights, TODO, Problems, Terminal, Services, and App Inspection.

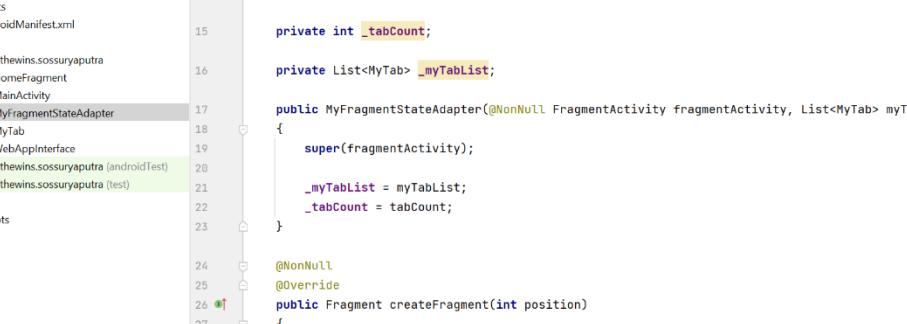
```
File Edit View Navigate Code Refactor Build Run Tools Window Help SOSuryaPutra - MainActivity.java [SOS_SuryaPutra.app.main]
SOSuryaPutra app src main java com.thewins.sosuryaputra MainActivity
MainActivity.java layout_webview.xml activity_main.xml MyFragmentStateAdapter fragment_home.xml
C:\Users\tony\AndroidStudioProjects\SOSuryaPutra\app\src\main\java\com\thewins\sosuryaputra\MainActivity.java

89     });
90     // End: Check update
91
92     _tabLayout = findViewById(R.id.tabLayout1);
93     _tabLayout.addTab(_tabLayout.newTab());
94     _tabLayout.addTab(_tabLayout.newTab());
95     _tabLayout.addTab(_tabLayout.newTab());
96     _tabLayout.setTabGravity(TabLayout.GRAVITY_FILL);
97
98     _urlList = new ArrayList<>();
99     _urlList.add("https://thewins.web.id/BbSos/.tw");
100    _urlList.add("https://thewins.web.id/BbSos/sales/order.tw");
101    _urlList.add("https://thewins.web.id/BbSos/sales/setoran.tw");
102
103    _myTabList = new ArrayList<>();
104    MyTab myTab = new MyTab(name: "Beranda", _urlList.get(0));
105    _myTabList.add(myTab);
106    myTab = new MyTab(name: "Sales Order", _urlList.get(1));
107    _myTabList.add(myTab);
108    myTab = new MyTab(name: "Setoran", _urlList.get(2));
109    _myTabList.add(myTab);
110
111    _viewPager = findViewById(R.id.viewPager1);
112    _myFragmentStateAdapter = new MyFragmentStateAdapter(fragmentActivity: this, _myTabList, _tabLayout.getTabCount());
113    _viewPager.setAdapter(_myFragmentStateAdapter);
114    _viewPager.setOffscreenPageLimit(_tabLayout.getTabCount());
115
116    TabLayoutMediator.TabConfigurationStrategy strategy = new TabLayoutMediator.TabConfigurationStrategy()
117    {
118        @Override
119        public void onConfigureTab(@NotNull TabLayout.Tab tab, int position)
120        {
121            MyTab myTab1 = _myTabList.get(position);
122            tab.setText(myTab1.getName());
123        }
124    };
125
126    TabLayoutMediator mediator = new TabLayoutMediator(_tabLayout, _viewPager, strategy);
127    mediator.attach();
128}

Git Profiler Logcat App Quality Insights TODO Problems Terminal Services App Inspection
daemon started successfully (a minute ago)
31:14 main
```

Gambar 6. Listing Program MainActivity.java

File `MainActivity.java` merupakan file utama pada project ini. File ini akan menginisialisasi 3 tab masing-masing dengan URL yang berbeda : beranda, penjualan dan sales order. Sebelum pemakai melakukan login, maka tab yang muncul hanya satu, yaitu tab Login. Segera setelah login, maka tab kedua dan ketiga otomatis dimunculkan.



```
public class MyFragmentStateAdapter extends FragmentStateAdapter{

    private int _tabCount;

    private List<MyTab> _myTabList;

    public MyFragmentStateAdapter(@NotNull FragmentActivity fragmentActivity, List<MyTab> myTabList, int tabCount) {
        super(fragmentActivity);

        _myTabList = myTabList;
        _tabCount = tabCount;
    }

    @NotNull
    @Override
    public Fragment createFragment(int position) {
        MyTab myTab = _myTabList.get(position);
        return HomeFragment.newInstance(myTab.getUrl(), position);
    }

    @Override
    public int getItemCount() {
    }
}
```

Gambar 7. File MyFramgmentStateAdapter.java

File ini merupakan adapter untuk mengatur ketiga tab yang ada pada aplikasi ini. Dengan MyFragmentStateAdapter.java, pemakai dapat berpindah antara tab yang satu ke tab yang lainnya dengan menggeser layer smartphone ke kiri atau ke kanan. Selain itu, pemakai juga dapat berpindah tab dengan melakukan sentuhan pada daftar tab yang ditampilkan di bagian atas aplikasi.

```

public class WebAppInterface {
    private Context _context;
    private SharedPreferences _sharedPreference;
    private TabLayout _tabLayout;
    private ViewPager2 _viewPager;

    public WebAppInterface(Context context) {
        _context = context;
        _sharedPreference = _context.getSharedPreferences("global", Context.MODE_PRIVATE);
    }

    @JavascriptInterface
    public void updateAuthenticationStatus(boolean status) {
        SharedPreferences.Editor sharedPreferenceEditor = _sharedPreference.edit();
        sharedPreferenceEditor.putBoolean("isAuthenticated", status);
        sharedPreferenceEditor.apply();

        //Log.d("*tw*", "Update - isAuthenticated: " + status);
    }
}

```

Gambar 8. File WebAppInterface.java

File di atas berisi coding untuk dipanggil secara interoperable (lintas platform dan Bahasa pemrograman) oleh JavaScript. Semua fungsi di dalam class ini akan diinjeksi ke dalam peramban sehingga dapat diakses oleh bahasa pemrograman yang berada di peramban, yaitu JavaScript. Dengan demikian, secara tidak langsung JavaScript dapat mengontrol aplikasi Android tanpa batasan, misalnya mengakses hardware pada smartphone Android. Inilah keunggulan aplikasi hybrid, karena mampu mengekskalasi bahasa pemrograman web hingga memiliki wewenang penuh terhadap perangkat keras yang digunakan oleh pemakai. Dengan class ini, aplikasi web akan mampu mengakses GPS, SMS, kontak, Bluetooth dan lain-lainnya yang ada para perangkat Android.

Pengujian dilakukan dengan membandingkan waktu yang dibutuhkan untuk mengembangkan aplikasi hybrid dengan mengembangkan aplikasi untuk masing-masing platform (Web, Mobile dan Desktop). Data tersebut dapat dilihat pada table berikut:

Tabel 1. Waktu yang Dibutuhkan untuk Pengembangan Aplikasi.

Platform	Keterangan	Waktu
----------	------------	-------

Android	Mengembangkan aplikasi dari awal menggunakan Android Studio	2 minggu
Web	Mengembangkan aplikasi dari awal menggunakan framework CodeIgniter	2 minggu
Desktop	Mengembangkan aplikasi dari awal menggunakan Visual Studio (.NET Framework)	2 minggu
Hybrid	Mengembangkan aplikasi web dari awal menggunakan framework CodeIgniter sebanyak 90%, ditambah Android Studio sebanyak 10%	2 minggu 2 hari

Apabila kita mengembangkan aplikasi untuk masing-masing platform, yaitu platform Web, Desktop dan Android, maka waktu yang dibutuhkan untuk masing-masing platform kurang-lebih 2 minggu, dengan totalnya adalah 6 minggu. Sedangkan apabila kita mengembangkan aplikasi dengan teknik hybrid, maka waktu yang dibutuhkan hanya 2 minggu lewat 2 hari saja. Efisiensi yang dapat dicapai dengan menerapkan teknik ini tentu sangat signifikan, yaitu sekitar 3 kali lipat atau 200%. Hal ini membuktikan bahwa implementasi teknik hybrid akan meningkatkan time-to-market developer dalam mengembangkan aplikasi secara signifikan.

KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan, maka dapat diambil kesimpulan bahwa implementasi teknik hybrid dalam mengembangkan aplikasi multi platform pada SOS Surya Putra berhasil meningkatkan efisiensi developer secara signifikan apabila dibandingkan dengan mengembangkan aplikasi untuk masing-masing platform secara mandiri. Peningkatan performa ini berarti berkurangnya time-to-market developer dalam mengembangkan aplikasi multi platform.

Para developer aplikasi web dapat memanfaatkan hasil penelitian ini sebagai bahan pertimbangan dalam pemilihan teknologi, apabila mereka akan mengerjakan proyek rekayasa perangkat lunak yang baru. Dengan menerapkan teknik hybrid ini, maka mereka akan mampu merekayasa aplikasi untuk berbagai macam platform dengan time-to-market yang lebih cepat. Diharapkan penelitian ini dapat dikembangkan lebih lanjut lagi dengan membahas sisi keamanannya.

UCAPAN TERIMA KASIH

Ucapan banyak terima kasih disampaikan kepada Bapak Dr. Sandy Kosasi, S.E., M.M., M.Kom. yang telah membantu memberikan informasi mengenai referensi yang layak untuk penelitian ini dan tentunya sudah mengizinkan objek penelitian di kampus STMIK Pontianak.

DAFTAR REFERENSI

- Albarqi, A. A. (2018). The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development. *I.J. Information Engineering and Electronic Business*, 3, 13.
- Apache Cordova*. (2023). (Apache Software Foundation) Dipetik 03 09, 2023, dari <https://cordova.apache.org>
- Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., . . . Thomas, D. (2001). *Signatories: The Agile Manifesto*. Dipetik September 8, 2018, dari <http://agilemanifesto.org/>
- Bouchrika, I. (2022, 12 20). *Mobile vs Desktop Usage Statistics for 2023*. (Research.com) Dipetik 02 28, 2023, dari <https://research.com/software/mobile-vs-desktop-usage#:~:text=Another%20major%20difference%20between%20a,to%20allow%20for%20Internet%20access>.
- CodeIgniter Web Framework*. (t.thn.). (CodeIgniter Foundation) Dipetik 10 27, 2020, dari <https://codeigniter.com>
- Download Android Studio & App Tools - Android Developers*. (2023). (Google) Dipetik 05 18, 2023, dari <https://developer.android.com/studio>
- Flutter - Build apps for any screen*. (t.thn.). (Google) Dipetik 03 09, 2023, dari <https://flutter.dev>
- Free Bootstrap Admin Template*. (2022). (AdminLTE) Dipetik Mei 20, 2022, dari <https://adminlte.io/>
- Ionic Framework - The Cross-Platform App Development Leader*. (2023). (Ionic) Dipetik 03 09, 2023, dari <https://ionicframework.com>
- Lindstrom, L., & Jeffries, R. (2004). Extreme Programming and Agile Software Development Methodologies. *Information Systems Management*, 41-52.
- Microsoft. (2021). *Visual Studio Code - Code Editing, Redefined*. (Microsoft) Dipetik Agustus 6, 2021, dari <https://code.visualstudio.com/>
- PHP: Hypertext Preprocessor*. (2023). (The PHP Group) Dipetik 05 01, 2023, dari <https://www.php.net/>
- Purhantara, W. (2010). *Metode Penelitian Kualitatif untuk Bisnis*. Yogyakarta: Graha Ilmu.
- React Native - Learn once, write anywhere*. (2023). (Meta Platforms, Inc.) Dipetik 03 09, 2023, dari <https://reactnative.dev>

Statistik, B. P. (2023, 02 27). *Persentase Penduduk yang Memiliki/Menguasai Telepon Seluler Menurut Provinsi dan Klasifikasi Daerah 2019-2021.* (Badan Pusat Statistik) Diambil kembali dari <https://www.bps.go.id/indicator/2/395/1/persentase-penduduk-yang-memiliki-menguasai-telepon-seluler-menurut-provinsi-dan-klasifikasi-daerah.html>

Supporting continuity and open collaboration. (2019). (MariaDB) Dipetik 09 15, 2019, dari <https://mariadb.org>

The leading operating system for PCs, IoT devices, servers and the cloud. (t.thn.). (Ubuntu) Dipetik 09 15, 2019, dari <https://ubuntu.com>