

Platform Interaktif Berbasis Web Untuk Pembelajaran Desain Antarmuka Pengguna Aplikasi Android Dengan Fungsi Validasi Otomatis

Yan Watequlis Syaifudin¹, Muhammad Sofiul Fuad Ruslan², Rudy Ariyanto³, Nabilah Argyanti Ardyningrum⁴

^{1,2,3,4}Jurusan Teknologi Informasi, Politeknik Negeri Malang

¹qulis@polinema.ac.id, ²sofiulfr@gmail.com, ³rudy@polinema.ac.id, ⁴nargyanti@gmail.com

Abstrak

Seiring meningkatnya popularitas perangkat pintar Android, permintaan programmer aplikasi Android telah meningkat pesat. Atas dasar fakta ini, pemrograman aplikasi Android telah menjadi salah satu mata pelajaran paling populer di program studi Teknologi Informasi. Dalam aplikasi Android, antarmuka pengguna memiliki peran penting untuk membuat aplikasi menjadi interaktif agar pengguna dapat nyaman menggunakan aplikasi Android. Berdasarkan studi sebelumnya, *Android Programming Learning Assistance System (APLAS)* yang telah digunakan sebagai platform pembelajaran mandiri untuk pembelajaran pemrograman *Android*. Dalam makalah ini, kami mengusulkan platform pembelajaran mandiri berbasis web di APLAS untuk mempelajari cara membuat desain antarmuka pengguna secara interaktif menggunakan kode XML. Materi pembelajaran terdiri dari 15 topik yang mencakup pembelajaran tata letak dan penggunaan *resource* untuk antarmuka aplikasi. Mahasiswa akan mendapatkan tugas dalam dokumen panduan yang harus diselesaikan pada setiap topik, kemudian jawaban mahasiswa tersebut dapat divalidasi secara otomatis di server dan web APLAS akan menampilkan hasil validasi, yaitu berupa PASSED dan FAILED. Untuk mengevaluasi efektivitas dari platform APLAS ini, kami meminta 40 mahasiswa di program studi Teknologi Informasi di Indonesia untuk menyelesaikan semua topik pembelajaran pemrograman Android yang diberikan pada web. Hasil pengujian menunjukkan bahwa semua mahasiswa menyelesaikan tugas dengan benar dan memberi komentar positif pada nilai mereka, yang mengkonfirmasi keefektifan sistem APLAS.

Kata kunci : APLAS, *Android*, antarmuka pengguna, *XML*, aplikasi web, validasi otomatis.

1. Pendahuluan

Dengan pesatnya pertumbuhan popularitas ponsel pintar, peluang pekerjaan bagi pengembang aplikasi ponsel pintar turut mengalami peningkatan pesat (Hsiao, 2013). Berdasarkan data dari Statista (Statista GmbH, 2020) mengungkapkan bahwa jumlah pengguna ponsel pintar lebih dari lima miliar pengguna di seluruh dunia, dan masih terus berkembang, dan *Android* adalah sistem operasi yang paling populer. Menyediakan *platform* pemrograman menjadi salah perhatian utama di bidang teknologi informasi, terutama di sekolah profesional dan universitas.

Tampilan antarmuka adalah salah satu bagian penting pada aplikasi *Android* (Joo, 2017; Taba et al., 2017). Antarmuka menentukan pengalaman pengguna ketika menggunakan aplikasi. Pada versi awal *Android*, antarmuka pengguna masih sederhana. Tetapi saat ini, antarmuka menjadi lebih kompleks dan interaktif. Perubahan ini didasarkan pada permintaan pengguna aplikasi yang menginginkan tampilan antarmuka yang elegan, di samping fungsi aplikasi yang lengkap (Arnatovich et al., 2016).

Untuk menyediakan *platform* pembelajaran mandiri desain antarmuka aplikasi *Android*. Kami mengusulkan *platform* pembelajaran interaktif berbasis web pada *Android Programming Learning Assistance System (APLAS)* (Syaifudin et al., 2019) APLAS adalah *platform* untuk mempelajari pemrograman *Android* secara mandiri dengan fitur validasi otomatis. Melalui penggunaan metode *Test-Driven Development* (Kim, 2013), mahasiswa dapat menyelesaikan tugas berdasarkan panduan yang diberikan, lalu pekerjaan mahasiswa divalidasi secara otomatis dengan bantuan sistem. *Platform* untuk pembelajaran desain antarmuka *Android* mengadopsi sistem yang digunakan APLAS.

Dalam makalah ini, kami memaparkan implementasi dan evaluasi *platform* pembelajaran desain antarmuka *Android* di APLAS. *Platform* ini menyediakan materi pembelajaran yang meliputi pembelajaran desain antarmuka menggunakan empat jenis *layout*, *string resources*, dan *color resources*. Materi tersebut terbagi menjadi 15 topik pembelajaran, mulai antarmuka yang sederhana hingga yang kompleks. Setiap topik pembelajaran

menyediakan dokumen panduan untuk mempelajari konsep dan menyelesaikan tugas. Jawaban mahasiswa berupa kode *XML* harus dikirimkan ke *server*, kemudian mahasiswa dapat melihat hasil validasi setelah program *validator* memeriksanya.

Untuk evaluasi, kami meminta 40 mahasiswa di program studi Teknologi Informasi di Indonesia untuk menyelesaikan tugas tersebut. Hasil evaluasi menunjukkan bahwa semua mahasiswa dapat menyelesaikan 15 topik yang diberikan. Sebagian besar mahasiswa berpendapat bahwa *platform* dan materi pembelajaran ini sangat membantu untuk mempelajari desain antarmuka aplikasi *Android*. Dengan demikian, dapat dikonfirmasi tingkat efektivitas *platform* dan materi pembelajaran ini.

2. Penelitian Terkait

Di bagian ini, kami melakukan survei penelitian terkait dengan pembelajaran desain antarmuka *Android*.

Hanafi dkk (Hanafi & Samsudin, 2012) memperkenalkan *Mobile Learning Environment System* (MLES) sebagai sistem pembelajaran pemrograman *Android* untuk mahasiswa. Mereka mencatat bahwa interaktif, mudah diakses, dan sistem yang nyaman dinilai penting bagi mahasiswa untuk belajar dengan nyaman.

Kang dkk (Kang & Cho, 2015) mempelajari penggunaan *MIT App Inventor* untuk pembelajaran pemrograman *Android* bagi mahasiswa tingkat pemula. Melalui penggunaan *puzzle* yang diadopsi oleh aplikasi ini, mahasiswa merasa pembelajaran terasa lebih mudah dibandingkan menggunakan *Eclipse* yang secara umum digunakan untuk membuat aplikasi *Android*.

Rekhawi dkk (Alrakhawi et al., 2018) mengusulkan sistem bimbingan cerdas untuk pengembangan aplikasi *Android*. Berdasarkan modul tutorial yang diberikan, sistem ini tidak membuat proses pembelajaran menjadi interaktif.

Ohkawa dkk (Ohkawa et al., 2018) Melakukan penelitian dengan mengembangkan aplikasi untuk pembelajaran desain antarmuka yang terdiri dari pembelajaran tatap muka di kelas dan tugas di rumah dengan menggunakan *e-learning*. Namun, penelitian tersebut tidak memberikan pembelajaran interaktif bagi mahasiswa.

3. Tinjauan Tentang APLAS

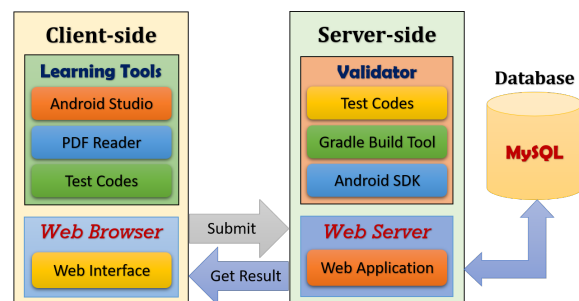
Pada bagian ini, kami menyajikan implementasi APLAS pada penelitian sebelumnya (Syarifudin et al., 2019; Syaifudin, et al., 2021)

3.1 Gambaran Umum

Sebagai sistem belajar mandiri berbasis web, APLAS menyediakan *platform* pembelajaran pemrograman *Android* secara mandiri, dengan sedikit campur tangan guru. *Platform* ini memiliki fitur validasi otomatis untuk menilai setiap jawaban dari tugas mahasiswa. Jawaban tersebut diverifikasi dengan *test code* yang mengadopsi metode TDD menggunakan *JUnit* dan *Robolectric*.

3.2 Platform Online

APLAS menyediakan *platform online* berbasis web untuk guru dan mahasiswa. Melalui *platform* ini, guru dapat mendistribusikan materi pembelajaran untuk mahasiswa, kemudian mahasiswa dapat mengunduh materi tersebut. Jawaban dari tugas tersebut dikumpulkan oleh mahasiswa untuk dikonfirmasi kebenarannya. Proses validasi terjadi secara otomatis pada sistem, kemudian mahasiswa dan guru dapat melihat hasil validasi tersebut. *Platform* ini mengimplementasikan model *client-server* berbasis aplikasi web, seperti yang diilustrasikan pada Gambar 1.



Gambar 1. Arsitektur *platform* berbasis web.

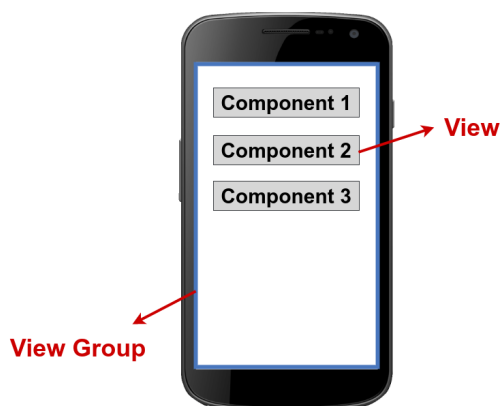
- Client-side*: Pada sisi *client*, APLAS menyediakan aplikasi web untuk mahasiswa yang dapat diakses menggunakan web browser. Aplikasi web ini berguna untuk mengunduh materi pembelajaran, mengirimkan jawaban ke *server*, dan melihat hasil validasi jawaban.
- Server-side*: Pada sisi *server* mengimplementasikan *Apache server*, aplikasi web berbasis *PHP*, sistem *database MySQL*, program *validator* untuk memvalidasi jawaban yang diserahkan oleh mahasiswa. Aplikasi web ini dapat diakses melalui internet, sehingga mahasiswa dan guru dapat menggunakan APLAS dari mana saja.

4. Platform Pembelajaran Interaktif Berbasis Web Untuk Desain Antarmuka Pengguna

Pada bagian ini, kami menyajikan implementasi *platform* pembelajaran desain antarmuka aplikasi *Android* pada APLAS.

4.1 Komponen Android UI

Untuk membuat antarmuka aplikasi *Android*, ada tiga komponen penting untuk membangunnya, yaitu *layout*, *string resource*, dan *color resource*.



Gambar 2. *View* dan *view group* dalam satu *layout*.

- a. *Layout*: *Layout* adalah bagian utama antarmuka aplikasi *Android* yang harus ditulis dalam kode *XML*. *Android* menggunakan konsep hierarki untuk mengatur semua komponen di dalam *layout*. *View* adalah elemen antarmuka yang dapat dilihat dan berinteraksi oleh pengguna. *ViewGroup* adalah komponen tak terlihat yang berisi beberapa *View* atau *ViewGroup* di atasnya, seperti yang ditunjukkan pada Gambar 2.
- b. *String Resource*: *String Resource* berisi kumpulan referensi dari nilai *string* yang dapat digunakan dalam aplikasi. Melalui penggunaan *string resource*, seorang *programmer* tidak perlu mendefinisikan nilai *string* pada sebuah komponen antarmuka secara *hard-code*. Penggunaannya seperti yang ditunjukkan pada Gambar 3.
- c. *Color Resource*: Mirip dengan *string resource*, *color resource* berisi kumpulan referensi nilai warna. Nilai warna didefinisikan dengan nilai *red-green-blue* (RGB) dan *alpha channel*. *Color resource* sangat membantu dalam menyediakan warna yang konstan dan dapat meningkatkan kualitas tampilan dan nuansa antarmuka aplikasi.



Gambar 3. Contoh penggunaan *string* dan *color resource*.

4.2 Topik Pembelajaran

Untuk mempelajari desain antarmuka dengan APLAS, kami menyediakan 15 topik yang mencakup empat jenis *layout*, *string resource*, dan *color resource*, seperti yang dijelaskan pada Tabel I.

4.3 Arsitektur Sistem

Sebagai sebuah sistem, *platform* berbasis web ini mengadopsi arsitektur *client-server* APLAS, seperti yang diilustrasikan pada Gambar 1. Sisi *client* digunakan oleh mahasiswa untuk mengakses aplikasi web menggunakan *web browser*. Sisi *server* berisi program *validator* dan sistem *database*.

4.4 Spesifikasi Aplikasi Web

Spesifikasi aplikasi web untuk pembelajaran antarmuka *Android*, proses validasi kode otomatis, dan cara menggunakan aplikasi web.

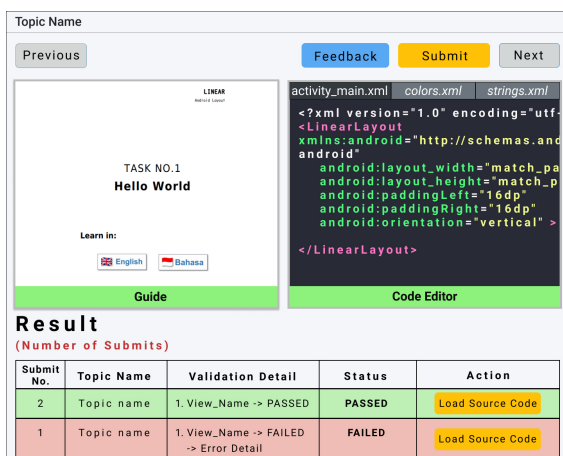
- a. Antarmuka Web untuk Pembelajaran : Aplikasi web terbagi menjadi tiga bagian, seperti yang ditunjukkan pada Gambar 5. Bagian kiri menunjukkan dokumen panduan. Bagian kanan adalah kode editor dengan fitur *auto-completion* serta *syntax highlighting* yang mirip dengan fitur pada *Android Studio* untuk menulis kode *XML* sebagai jawaban tugas. Pada bagian bawah digunakan untuk menunjukkan hasil validasi setelah mengirimkan jawaban dan divalidasi oleh *validator*.
- b. Validasi Kode Otomatis: Untuk memvalidasi kode jawaban, APLAS mengimplementasikan *unit testing* dengan mengadopsi dua *tool* yaitu, *JUnit* (Junit, 2021) dan *Robolectric* (Robolectric, 2021). Setiap topik memiliki *test code* yang mengimplementasikan kedua *tool* tersebut. Seperti yang diilustrasikan pada Gambar 6, ketika mahasiswa mengirimkan jawaban, program *validator* akan memprosesnya dengan mengemasnya menjadi sebuah *Android Project*. Kemudian, *test code* ditambahkan ke dalam *Android project* tersebut, sebelum program *validator* menjalankannya. Hasil dari menjalankan *test code* disimpan ke dalam sistem basis data (Syarifudin, et al., 2021; Syaifudin, et al., 2021).
- c. Cara Menggunakan Aplikasi Web: Disediakan 15 topik pembelajaran. Mahasiswa memilih topik pertama untuk mulai pembelajaran. Kemudian, mahasiswa mengikuti instruksi pada dokumen panduan dan menulis kode *XML* pada kode editor. Setelah itu mahasiswa mengirimkan jawaban untuk divalidasi. Hasil validasi akan muncul di sisi bawah halaman web.

Tabel 1. Deskripsi 15 topik pembelajaran antarmuka Android.

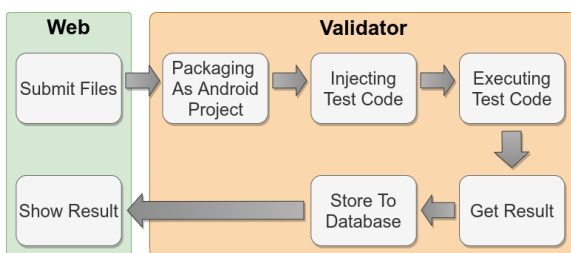
no.	Nama topik	Deskripsi	subjek
1	<i>Hello World Text</i>	Mahasiswa belajar cara menggunakan <i>TextView</i> pada <i>LinearLayout</i> untuk aplikasi Hello World.	<i>Linear Layout</i>
2	<i>Email & Password EditText</i>	Mahasiswa belajar cara menggunakan <i>EditText</i> untuk menampung <i>email</i> dan <i>password</i> .	<i>Linear Layout</i>
3	<i>Exit Button & Login Button</i>	Mahasiswa belajar cara menggunakan <i>Button</i> untuk membuat fungsi tombol login dan exit.	<i>Linear Layout</i>
4	<i>Nested View in LinearLayout</i>	Mahasiswa belajar cara membuat <i>NestedView</i> untuk menggabungkan dua jenis orientasi <i>layout</i> pada <i>LinearLayout</i> .	<i>Linear Layout</i>
5	<i>Simple Login Form</i>	Mahasiswa belajar cara membuat halaman login sederhana dengan mengkombinasikan <i>Textview</i> , <i>EditText</i> , dan <i>Button</i> pada <i>LinearLayout</i> .	<i>Linear Layout</i>
6	<i>String resource</i>	Mahasiswa belajar cara membuat <i>string resources</i> dan mengaplikasikannya pada <i>RelativeLayout</i> .	<i>Relative Layout, String Resources</i>
7	<i>Color resource</i>	Mahasiswa belajar cara membuat dan menggunakan <i>color resource</i> , dan mengatur posisi <i>View</i> pada <i>RelativeLayout</i>	<i>Relative Layout, Color Resources</i>
8	<i>Simple ImageView</i>	Mahasiswa belajar cara menampilkan gambar menggunakan <i>ImageView</i> dan mengatur posisinya pada <i>RelativeLayout</i>	<i>Relative Layout</i>
9	<i>Modern Login Form</i>	Mahasiswa belajar membuat halaman login modern dengan menggabungkan <i>ImageView</i> , <i>Textview</i> , <i>EditText</i> , dan <i>Button</i> pada <i>RelativeLayout</i> .	<i>Relative Layout</i>
10	<i>Basic of Constraint Layout</i>	Mahasiswa belajar cara mengatur <i>View</i> pada <i>ConstraintLayout</i> .	<i>Constraint Layout</i>
11	<i>Picture in Picture Constraint Layout</i>	Mahasiswa belajar cara mengatur posisi sebuah <i>ImageView</i> diatas <i>ImageView</i> lain pada <i>ConstraintLayout</i> .	<i>Constraint Layout</i>
12	<i>Modern E-Book Store</i>	Mahasiswa belajar cara membuat halaman penjualan <i>E-book</i> modern dengan menggunakan <i>ImageView</i> , <i>Textview</i> , <i>Divider</i> , dan <i>Button</i> pada <i>ConstraintLayout</i>	<i>Constraint Layout</i>
13	<i>Basic of Grid Layout</i>	Mahasiswa belajar cara mengatur posisi <i>View</i> pada <i>GridLayout</i>	<i>Grid Layout</i>
14	<i>Score Board Application</i>	Mahasiswa belajar cara membuat tampilan aplikasi <i>Score Board</i> .	<i>Grid Layout</i>
15	<i>Photo Sharing Application</i>	Mahasiswa belajar cara membuat tampilan aplikasi berbagi foto.	<i>Grid Layout</i>



Gambar 4. Contoh desain antarmuka dari 4 jenis layout.



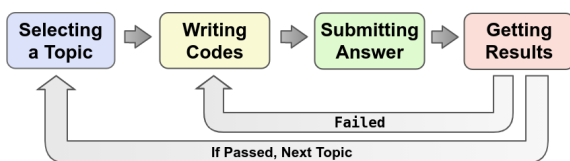
Gambar 5. Desain antarmuka aplikasi web.



Gambar 6. Proses validasi jawaban mahasiswa

4.5 Proses pembelajaran oleh Mahasiswa

Untuk mempelajari desain antarmuka *Android*, mahasiswa harus mengikuti empat langkah proses pembelajaran di bawah, seperti yang ditunjukkan pada Gambar 7.



Gambar 7. Proses pembelajaran mahasiswa

- Memilih Topik: Mahasiswa memilih topik pertama dari topik yang telah disediakan.
- Menulis Kode: Mahasiswa mengikuti panduan pada dokumen panduan, mahasiswa harus menyelesaikan tugas yang diberikan dengan cara menuliskan kode *XML* pada kode editor.
- Mengumpulkan Jawaban: Setelah menyelesaikan tugas, mahasiswa harus mengirimkan kode jawaban ke *server*. Kemudian, kode jawaban tersebut divalidasi secara otomatis oleh program *validator* di *server*.
- Mendapatkan hasil: Hasil validasi terbagi menjadi dua jenis, antara lain: *passed* dan *failed*. Jika mendapatkan hasil *failed*, mahasiswa harus memperbaiki kode jawaban, lalu melakukan pengumpulan jawaban kembali. Jika

mendapatkan hasil *passed*, mahasiswa dapat melanjutkan ke topik berikutnya.

5. Evaluasi

Pada bagian ini, kami mengevaluasi implementasi *platform* pembelajaran interaktif berbasis web untuk desain antarmuka pengguna aplikasi *Android*.

5.1 Data Pengujian

Untuk melakukan evaluasi secara komprehensif pada *platform* ini, kami meminta 40 mahasiswa program studi Teknologi Informasi untuk menyelesaikan 15 topik secara berurutan dalam empat hari. Mereka mengakses aplikasi web menggunakan *PC* pribadi dan menggunakan *Chrome* sebagai *web browser*.

Untuk mengukur efektivitas setiap topik, dilakukan pengamatan terhadap jumlah pengiriman jawaban dan umpan balik oleh mahasiswa.

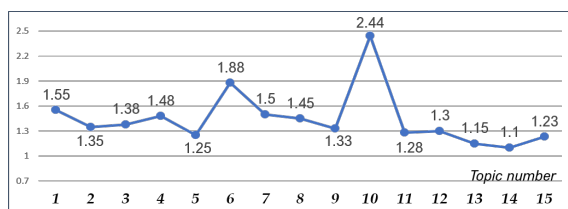
5.2 Hasil Pengujian Mahasiswa

Dalam empat hari, sebanyak 40 mahasiswa berhasil menyelesaikan semua topik. Beberapa dari mereka harus menyerahkan beberapa kali sebelum berhasil menyelesaikan tugas yang diberikan. Gambar 8 menunjukkan jumlah rata-rata pengiriman jawaban untuk setiap topik oleh mahasiswa. Jumlah rata-rata pengiriman jawaban untuk seluruh topik adalah 1,44 kali. Hasil ini mengkonfirmasi bahwa siswa dapat dengan mudah menyelesaikan tugas di semua topik.

5.3 Kesulitan Setiap Topik

Pada Gambar 8, ada tiga puncak dalam grafik yang dihasilkan oleh topik 1, topik 6, dan topik 10. Topik 1 adalah topik dan materi pertama dalam urutan pembelajaran, sehingga mahasiswa mulai beradaptasi dengan lingkungan belajar. Topik 6 mempelajari pembuatan dan penggunaan *string resource* dengan kode *XML*, sehingga mahasiswa mulai belajar untuk membuat sebuah *resource*. Topik 10 menerapkan antarmuka aplikasi paling kompleks, sehingga banyak mahasiswa yang harus memperbaiki jawabannya beberapa kali untuk mendapatkan hasil *passed*.

Berdasarkan hasil yang didapatkan, kami mengklasifikasikan kesulitan menjadi dua tingkat, yaitu tingkat mudah dan tingkat sulit. Topik dengan tingkat sulit berisi tiga topik yang memiliki jumlah rata-rata pengiriman jawaban tertinggi, termasuk topik 1, topik 6, dan topik 10. Level mudah berisi topik selain topik yang telah disebutkan.



Gambar 8. Rata-rata jumlah pengumpulan mahasiswa untuk setiap topiknya.

5.4 Umpan Balik Mahasiswa

Berikut adalah umpan balik yang paling banyak diberikan oleh mahasiswa:

- a. Komentar Positif (37 mahasiswa atau 92.5%):
 - Materi pembelajaran tersedia dalam dua bahasa sehingga mudah untuk dipelajari.
 - Pembelajaran menjadi lebih mudah.
 - Membantu saya mengingat pembelajaran desain antarmuka Android.
 - Sekarang saya mengerti tentang Grid Layout pada antarmuka Android.
- b. Komentar Masukan (10 mahasiswa atau 25%):
 - Membutuhkan penjelasan tentang penamaan dalam *string resource*.
 - Butuh waktu untuk menyelesaikan topik pembelajaran.

5.5 Kesimpulan

Berdasarkan hasil evaluasi, seluruh mahasiswa dapat menyelesaikan 15 topik yang diberikan dan sebagian besar memberikan komentar positif, yang di mana ini dapat mengkonfirmasi tingkat efektivitas dari *platform* pembelajaran ini. Hasil ini sesuai dengan tujuan pembelajaran dimana platform pembelajaran ini ditujukan untuk memudahkan mahasiswa belajar desain UI dengan XML pada platform pemrograman Android. Selain itu, berdasarkan hasil umpan balik mahasiswa (5.4), keunggulan dan kemudahan platform ini telah dibuktikan dimana 92.5% mahasiswa menyatakan komentar positif dan tidak ada yang menyatakan negatif, ditambah beberapa komentar feedback. Untuk selanjutnya, kami akan mengimplementasikan fitur ujian untuk pembelajaran desain antarmuka *Android* di APLAS.

Daftar Pustaka:

Alrakhawi, H. A. S., al Rekhawi, H. A., & Abu-Naser, S. S. (2018), *An intelligent tutoring system for learning Android applications UI development*, In International Journal of Engineering and Information Systems, vol. 2, no 1, pp. 1-14

Arnatovich, Y. L., Ngoc Ngo, M., Beng Kuan, T. H., & Soh, C. (2016). *Achieving High Code Coverage in Android UI Testing via Automated*

Widget Exercising, 23rd Asia-Pacific Software Engineering Conference, pp. 193–200.

Hanafi, H. F., & Samsudin, K. (2012). *Mobile Learning Environment System (MLES): The Case of Android-based Learning Application on Undergraduates' Learning*, Int. J. Adv. Comput. Sci. Appl., vol. 3, no. 3, pp. 63-66.

Hsiao, K. (2013), *Android smartphone adoption and intention to pay for mobile internet*, Library Hi Tech, vol. 31, no. 2, pp. 216–235.

Joo, H. S. (2017), *A Study on UI/UX and Understanding of Computer Major Students*, Int. J. Adv. Smart Conv., vol. 6 no. 4, pp. 26–32.

Junit (2021, January), *JUnit: a simple framework to write repeatable tests*, Available at <https://junit.org/Junit4/>.

Kang, H., & Cho, J. (2015), *Case Study on Efficient Android Programming Education using Multi Android Development Tools*, Indian Journal of Science and Technology, vol. 8, no. 19, pp. 1-5.

Kim, H. (2013), *Test Driven Mobile Applications Development*, Lecture Notes in Engineering and Computer Science, vol. 2, pp. 785–789.

Ohkawa, Y., Kodama, M., Konno, Y., Zhao, X., & Mitsuishi, T. (2018). *A study on UI design of smartphone app for continuous blended language learning*, International Conference on Business and Industrial Research, pp 584–589.

Robolectric. (2021, January), *Robolectric: a framework that brings fast and reliable unit tests to Android*, <http://robolectric.org/>.

Statista GmbH. (2020), *Number of smartphone users worldwide from 2014 to 2020 (in billions)*, <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>.

Syaifudin, Y. W., Funabiki, N., Kuribayashi, M., & Kao, W.-C. (2019), *A proposal of Android Programming Learning Assistant System with implementation of basic application learning*, International Journal of Web Information Systems, vol. 16 no. 1, pp. 115–135.

Syaifudin, Y. W., Funabiki, N., Kuribayashi, M., Mentari, M., Saputra, P. Y., Yunhasnawa, Y., & Ulfa, F. (2021), *Web application implementation of Android programming learning assistance system and its evaluations*, IOP Conference Series: Materials Science and Engineering, Malang, Indonesia, vol. 1073 no. 1.

Syaifudin, Y. W., Funabiki, N., Wijaya, D. C., & Mu'aasyiqiin, I. (2021), *Performance Investigation of Unit Testing in Android Programming Learning Assistance System*, 3rd Global Conference on Life Sciences and Technologies (LifeTech), pp. 153–157.

Syaifudin, Y. W., Rohani, S., Funabiki, N., & Saputra, P. Y. (2021), *Blending Android Programming Learning Assistance System into Online Android Programming Course*, 2021 9th International Conference on Information and Education Technology (ICIET), pp. 26–33.

Taba, S. E. S., Keivanloo, I., Zou, Y., & Wang, S.
(2017), *An exploratory study on the usage of
common interface elements in android*

applications, Journal of Systems and Software,
vol. 131, pp. 491–504.

