

IMPLEMENTASI TOPIK PEMBELAJARAN *BASIC DATA STORAGE* PADA *ANDROID PROGRAMMING LEARNING ASSISTANCE SYSTEM*

Yan Watequlis Syaifudin¹, Devany Christabella Wijaya², Rudy Ariyanto³, Nabilah Argyanti Ardyningrum⁴

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

¹qulis@polinema.ac.id, ²devany1212.dc@gmail.com, ³rudy@polinema.ac.id, ⁴nargyanti@gmail.com

Abstrak

Saat ini, ponsel berbasis Android telah menjadi pemimpin pasar untuk perangkat seluler. Mengetahui fakta tersebut, kebutuhan pengembang aplikasi Android telah meningkat secara signifikan. Banyak sekolah dan perguruan tinggi yang menawarkan pemrograman aplikasi Android menjadi salah satu topik pembelajaran yang wajib diikuti, terutama jurusan Teknologi Informasi. Dalam pengembangan aplikasi Android, pemanfaatan data diperlukan untuk membuat aplikasi yang dinamis dan interaktif. Berdasarkan studi sebelumnya, *Android Programming Learning Assistance System (APLAS)* telah diterapkan sebagai platform belajar pemrograman Android secara mandiri. Dalam artikel ini, kami mengimplementasikan topik pembelajaran *Basic Data Storage* di APLAS sebagai topik pertama di *Content Provider*. Topik ini berisi materi pembelajaran arsitektur *Model-View-ViewModel (MVVM)* dengan *Data Binding*, pemanfaatan *Shared Preferences*, dan *Internal Storage*. Mahasiswa diharuskan mempelajari materi *Basic Data Storage* yang tersedia pada web APLAS, kemudian mengerjakan penugasan yang tersedia pada materi. Jawaban berbentuk *source code* dari mahasiswa kemudian divalidasi lewat web APLAS dan diserahkan kepada guru untuk laporan. Hasil pengujian dan evaluasi melalui 50 mahasiswa dari jurusan Teknologi Informasi dari seluruh Indonesia menunjukkan bahwa semua mahasiswa telah menyelesaikan semua tugas. Melalui komentar positif yang disampaikan oleh mahasiswa, topik ini terbukti efektif untuk mendukung pembelajaran pemrograman Android, terutama dalam topik *Basic Data Storage*.

Kata kunci : APLAS, Android, *Basic Data Storage*, MVVM, *Data Binding*, *Shared Preferences*, *Internal Storage*

1. Pendahuluan

Popularitas ponsel pintar Android berkembang pesat (Götz et al., 2017), sehingga permintaan untuk pengembangan aplikasi Android juga meningkat. Banyak sekolah dan perguruan tinggi mulai menawarkan pembelajaran pemrograman Android sebagai mata pelajaran utama, terutama jurusan Teknologi Informasi.

Salah satu platform yang menyediakan wadah untuk belajar mandiri pemrograman Android adalah *Android Programming Learning Assistance System (APLAS)* (Syaifudin et al., 2019). Saat mempelajari pemrograman Android dengan APLAS, mahasiswa berlatih mengembangkan aplikasi Android menggunakan bahasa Java atau Kotlin dan mendapatkan bantuan komputerisasi dalam penulisan kode sumber. Metode *Test Driven Development (TDD)* (Farcic & Garcia, 2015) diadopsi dengan menggunakan JUnit (JUnit, 2021) dan Robolectric (Robolectric, 2021), untuk memvalidasi jawaban mahasiswa secara otomatis.

Untuk mencakup berbagai tahapan pada pembelajaran pemrograman Android, APLAS menyediakan empat tahapan pembelajaran, yang terdiri dari *user interface*, *interactive application*, *content provider*, dan *service interaction* (Syaifudin,

et al., 2021). Setiap tahap pembelajaran memiliki topik pembelajaran individu yang memiliki tujuan masing-masing.

Pada aplikasi Android, pemanfaatan data digunakan untuk membuat aplikasi yang dinamis dan interaktif. Untuk penggunaan dasar, Android menyediakan cara untuk penyimpanan data *offline* secara permanen (Lyu et al., 2017), seperti *Shared Preferences* (Horton, 2015) dan *Internal Storage*. *Internal Storage* dapat digunakan untuk menyimpan file teks atau *key-value* yang berpasangan menjadi data dengan format. Selain itu Android merekomendasikan arsitektur MVVM untuk pengembangan aplikasi yang menggunakan *database* (Sun et al., 2016). Untuk implementasi aplikasi berbasis MVVM, Android menyediakan *library* untuk mengimplementasikan *Data Binding*, yang digunakan untuk mengikat komponen dari UI dalam tata letak ke *data source* menggunakan format deklaratif (Phillips et al., 2017).

Dalam artikel ini, kami menyajikan implementasi dan evaluasi *Basic Data Storage* sebagai topik pembelajaran pertama pada tahap kedua. Topik ini dimaksudkan untuk mempelajari penyimpanan data di aplikasi Android menggunakan *Shared Preferences* dan *Internal Storage* dengan arsitektur MVVM. Kami menawarkan lima tujuan

pembelajaran dan menerapkan pengembangan aplikasi *MyLibrary* sebagai tugas. Setelah menyelesaikan sepuluh tugas secara berurutan, mahasiswa dapat menghasilkan aplikasi *MyLibrary*, yaitu aplikasi untuk menyimpan data koleksi buku.

Evaluasi platform ini dilakukan dengan pengujian kepada 50 mahasiswa jurusan Teknologi Informasi di Indonesia. Mereka diminta untuk menyelesaikan tugas dalam waktu satu minggu dan kemudian mengirimkan jawabannya pada platform APLAS. Hasil pengujian menunjukkan bahwa semua mahasiswa menyelesaikan semua tugas dan memberikan umpan balik dengan pendapat positif. Hasil ini menegaskan keefektifan dari topik *Basic Data Storage* untuk mendukung pembelajaran aplikasi Android tentang penyimpanan data.

2. Penelitian Terkait

Pada bagian ini, kami menyajikan penelitian terkait pada pembelajaran pemrograman Android.

(Al Rekhawi & Abu-Naser, 2018) mengusulkan sebuah sistem yang digunakan untuk pengembangan aplikasi Android. Sistem ini menyediakan modul untuk merancang UI dan aplikasi. Berdasarkan modul tutorial yang diberikan, sistem ini tidak membuat proses pembelajaran menjadi interaktif.

(Ohkawa et al., 2018) melakukan penelitian untuk mengembangkan aplikasi *smartphone* untuk mendukung pembelajaran desain UI mahasiswa. Mereka menyiapkan sebuah prototipe aplikasi untuk latihan yang dinamai “KoToToMo”. Namun proposal tersebut tidak memberikan pembelajaran secara interaktif terhadap mahasiswanya.

3. Tinjauan Tentang APLAS

Bagian ini meninjau model, platform *online*, materi pembelajaran, dan validasi otomatis pada sistem APLAS (Syaifudin et al., 2019; Syaifudin, et al., 2021)

3.1 Tinjauan

APLAS adalah platform belajar mandiri untuk pembelajaran Android. Platform ini menyediakan platform yang memungkinkan mahasiswa untuk belajar secara mandiri. Dengan mengadopsi metode TDD, semua jawaban mahasiswa dapat divalidasi dengan bantuan algoritme dalam pengujian perangkat lunak. Sehingga dengan adanya APLAS, mahasiswa bisa mendapatkan materi pembelajaran dan tugas untuk pemrograman Android, kemudian menyelesaikan tugas dengan menulis *source code* di Android Studio dan melakukan validasi jawaban dengan mengeksekusi *test code* yang diberikan.

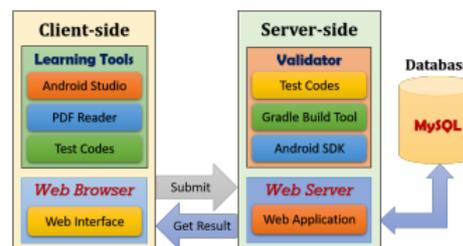
3.2 Materi Pembelajaran

APLAS mencakup berbagai topik pembelajaran yang mencakup berbagai materi pemrograman Android. Setiap topik berfokus pada satu bagian

spesifik dari keseluruhan proses untuk membangun sebuah aplikasi Android. Sebuah topik mencakup seperangkat materi pembelajaran yang terdiri dari *guide file*, *supplement file*, dan *test code* untuk mengimplementasikan TDD. Materi ini juga mencakup beberapa tugas yang digunakan untuk membimbing mahasiswa dalam mempelajari pemrograman Android langkah demi langkah.

Tahap kedua pada APLAS yaitu *Content Provider* yang fokus pada pengembangan dari aplikasi Android dengan *database*. Tahap ini terdiri dari empat topik, yaitu *Basic Data Storages*, *SQLite Database*, *Network Connection*, dan *Data Service*. Untuk studi ini, topik *Basic Data Storage* bertujuan untuk mempelajari penyimpanan data pada aplikasi Android menggunakan *Shared Preferences* dan *Internal Storage*, *SQLite Database*, *Network Connection*, dan *Data Service*.

3.3 Platform Online



Gambar 1 Arsitektur dari Web berbasis Online Platform

Platform *online* APLAS menggunakan model *client-server* yang diimplementasikan pada platform berbasis *website* yang digambarkan pada Gambar 1. Dengan platform ini, guru dapat mendistribusikan materi pembelajaran kepada mahasiswa dan mahasiswa dapat mengunduh materi tersebut. Jawaban yang dikirimkan oleh mahasiswa divalidasi oleh server dan hasilnya dilaporkan kepada guru.

Mahasiswa menggunakan Android Studio untuk menyelesaikan tugas yang diberikan dalam topik pembelajaran dengan menggunakan bahasa pemrograman Java dan XML. *Source code* yang ditulis oleh mahasiswa tersebut menghasilkan aplikasi Android yang dapat diuji. Pada sisi server, server menerapkan *Apache*, aplikasi web berbasis PHP, sistem *database MySQL*, dan program *validator* untuk memvalidasi program.

3.4 Validasi Jawaban Mahasiswa

Untuk memvalidasi secara otomatis jawaban mahasiswa, APLAS menggunakan dua alat pengujian, JUnit (Jain & Gopalani, 2017) dan Robolectric. JUnit merupakan *framework* default untuk melakukan unit testing secara otomatis pada Java. Robolectric, *framework* pengujian integrasi Android, digunakan untuk memvalidasi komponen

aplikasi Android, seperti tata letak UI, *activity*, *event listener*, dan *resource* (Almeida et al., 2019).

4. Implementasi Topik Basic Data Storage

Bagian ini menjelaskan materi pembelajaran yang disusun untuk topik *Basic Data Storage* yang menerapkan pengembangan aplikasi *MyLibrary*.

4.1 Basic Data Storage pada Aplikasi Android

Untuk membuat aplikasi yang menggunakan *database*, Android menggunakan arsitektur MVVM sebagai arsitektur terbaru. *Shared Preferences* dan *Internal Storage* dapat digunakan sebagai fungsi dasar penyimpanan *database*.



Gambar 2 Arsitektur MVVM dengan *Data Binding*

- 1) **MVVM Architecture dengan Data Binding:** MVVM atau Model-View-ViewModel adalah pola arsitektur yang memisahkan logika antarmuka pengguna dari logika bisnis, seperti yang ditunjukkan pada Gambar 2. Arsitektur ini menghilangkan semua kode GUI dan menggantinya dengan fungsi *Data Binding* dan menghasilkan arsitektur yang lebih bersih dengan memisahkan *view layer development* dari *layer* lainnya (Aljamea & Alkandari, 2018). MVVM adalah varian dari pola *Model-View-Controller (MVC)* yang digunakan di banyak pemrograman.
- 2) **Data Binding** memainkan peran penting dalam aplikasi berbasis MVVM. *Data Binding* dapat mengikat komponen UI dalam tata letak ke sumber data di aplikasi menggunakan format deklaratif (Google Developers, 2021a). *Data Binding* membutuhkan pendefinisian *binding class* yang dideklarasikan sebagai adaptor. Kemudian, notasi untuk pengambilan dan pemanggilan data didefinisikan dengan menggunakan *value tag* pada *UI layout*, seperti yang ditunjukkan pada Gambar 3.
- 3) **Shared Preferences** adalah penyimpanan data sederhana yang melibatkan pasangan *key-value*. *Key* harus didefinisikan sebagai *string* dan nilainya dapat menyimpan semua tipe data primitif seperti *string*, *int*, *float*, atau *boolean*. Android menyediakan *class 'SharedPreferences'* sebagai API untuk membaca, menulis, dan mengelola data ini (Google Developers, 2021b). Menyimpan data menggunakan *Shared Preferences* terdiri dari dua mode akses, termasuk *public* yang membuat file XML dapat diakses oleh aplikasi lain dan *private* yang menjaga file XML agar tidak bisa diakses oleh aplikasi lain dan mengamankan data pengguna.

- 4) **Menyimpan Data di Internal Storage** dapat dilakukan dengan menyimpannya dalam *file* teks atau biner. Beberapa format *file* teks dapat diadopsi, seperti *plain text*, XML, JSON, atau format *.ini*. Fungsi untuk menyimpan, memperbarui, dan menghapus data pada *plain text* dapat dilakukan dengan *class FileInputStream* dan *FileOutputStream* untuk membaca dan menulis *file* teks.

4.2 Tujuan Pembelajaran

Setelah menyelesaikan pembelajaran dengan topik *Basic Data Storage*, kelima tujuan pembelajaran tersebut diharapkan dapat dicapai oleh mahasiswa. Tabel 1 menunjukkan gambaran kelima tujuan pembelajaran tersebut.

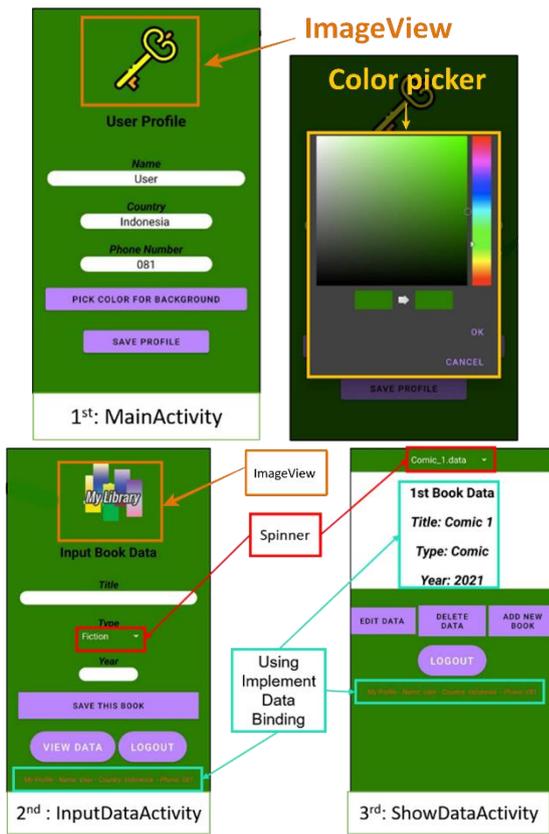
Tabel 1 Tujuan Pembelajaran *Basic Data Storage*

No	Tujuan	Keterangan
LO1	Application Resources	Mahasiswa dapat menerapkan <i>resource</i> aplikasi dalam proyek <i>Android</i> , seperti <i>drawable</i> , <i>string</i> , dan lain-lain.
LO2	UI Component	Mahasiswa mampu mengimplementasikan komponen UI, termasuk <i>Color Picker</i> , <i>ImageView</i> , <i>ListView</i> , dan <i>Spinner</i> .
LO3	Data Binding	Mahasiswa mampu mengimplementasikan MVVM untuk <i>Android</i> menggunakan <i>Data Binding</i> .
LO4	Shared Preferences	Mahasiswa mampu menerapkan <i>SharedPreferences</i> untuk menyimpan data persisten.
LO5	Internal Storage	Mahasiswa memahami cara menyimpan data menggunakan <i>file</i> teks di penyimpanan internal.

4.3 Pengumpulan Tugas

Untuk topik *Basic Data Storage*, kami menyediakan latihan pengembangan aplikasi *MyLibrary* untuk mahasiswa. Aplikasi *MyLibrary* dimaksudkan untuk menyimpan data koleksi buku pribadi. *MyLibrary* berisi tiga *Activity*, yaitu *MainActivity*, *InputDataActivity* dan *ShowDataActivity*, seperti yang diilustrasikan pada Gambar 3. *MainActivity* digunakan untuk mengisi dan menyimpan data profil pengguna, serta mengatur warna latar belakang aplikasi dengan menggunakan *widget ColorPicker*. Data profil pengguna akan disimpan di *Shared Preferences*. *InputDataActivity* digunakan untuk mengisi, mengedit, dan menyimpan data buku ke *Internal Storage* dengan menggunakan kelas *Data Binding*. Terakhir, *ShowDataActivity*

digunakan untuk menampilkan dan menghapus data buku.



Gambar 3 Tiga Activity di Aplikasi MyLibrary

Pada tampilan pertama aplikasi MyLibrary, pengguna harus mengisi data profil yang terdiri dari nama, negara, dan nomor telepon, serta memilih warna latar belakang aplikasi. Kemudian, pengguna harus mengklik tombol ‘Save My Profile’ untuk menyimpan data profil ke *Shared Preferences* dan membuka *InputDataActivity*. Setelah *Shared Preferences* telah ditetapkan, *MainActivity* tidak akan ditampilkan pada pembukaan aplikasi berikutnya. Pada *InputDataActivity*, pengguna dapat mengisi dan mengedit data buku yang terdiri dari judul, jenis, dan tahun terbit. Untuk menyimpan atau memperbarui data buku, tombol ‘Save this Book’ harus diklik. Untuk menampilkan data buku, tombol ‘View Data’ harus diklik dan kemudian akan muncul *ShowDataActivity*. Dengan menggunakan *Activity* ini pengguna dapat membuka semua data buku yang telah disimpan, menghapusnya, mengedit salah satunya, dan menambahkan yang baru.

4.4 Langkah Pembuatan Aplikasi

Topik ini menawarkan sepuluh tugas untuk membuat bagian-bagian dalam MyLibrary. Tabel 2 menunjukkan sepuluh tugas dengan korelasinya dengan lima tujuan pembelajaran.

Tabel 2 Korelasi tugas dengan tujuan pembelajaran

No	Deskripsi Tugas	LO
1	Memulai proyek dan mengonfigurasi <i>resource</i>	LO1
2	Merancang <i>MainActivity</i>	LO2
3	Merancang <i>InputDataActivity</i>	LO2
4	Merancang <i>ShowDataActivity</i>	LO2
5	Membuat <i>SharedPreferences</i> untuk data pengguna dan buku.	LO4
6	Mengembangkan data input dan tampilkan data dengan <i>Data Binding</i>	LO3
7	Membuat model pengikatan data untuk arsitektur MVVM.	LO3
8	Mengembangkan <i>MainActivity</i>	LO2 LO3
9	Mengembangkan <i>InputDataActivity</i>	LO3 LO4
10	Mengembangkan <i>ShowDataActivity</i>	LO3 LO4 LO5

4.5 Proses Pembelajaran

Topik ini membebaskan mahasiswa dalam mendesain UI aplikasi sesuai dengan preferensi mereka. Untuk mempelajari pengembangan aplikasi Android menggunakan topik ini, setiap mahasiswa harus mengikuti langkah-langkah berikut:

- 1) **Mulai belajar:** mengunduh materi melalui web dan memulai pembelajaran Android dengan mengikuti petunjuk dalam dokumen panduan.
- 2) **Mengonfigurasi project:** membuat proyek beserta konfigurasinya di Android Studio.
- 3) **Menyelesaikan penugasan:** menyelesaikan setiap tugas di semua tugas secara berurutan.
- 4) **Memvalidasi jawaban:** memvalidasi *source code* dengan menjalankan kode tes yang diberikan untuk setiap tugas.
- 5) **Mengirim jawaban:** mengirimkan kode jawaban sebagai laporan pembelajaran melalui web.

5. Evaluasi

Bagian ini menyajikan hasil evaluasi topik *Basic Data Storage* melalui implementasi kepada mahasiswa Teknologi Informasi di Indonesia.

5.1 Pengaturan Evaluasi

Evaluasi topik *Basic Data Storage* telah dilakukan dengan mengujikan topik ini kepada 50 mahasiswa S1 jurusan Teknologi Informasi di Indonesia. Sebelumnya, mereka memiliki pengalaman dalam memecahkan lima topik sebelumnya di tahap Antarmuka Pengguna dan Tahap Aplikasi Interaktif.

Dalam topik ini, mahasiswa diminta untuk menyelesaikan semua tugas selama satu minggu dan mengirimkan *source code* mereka ke server setelah

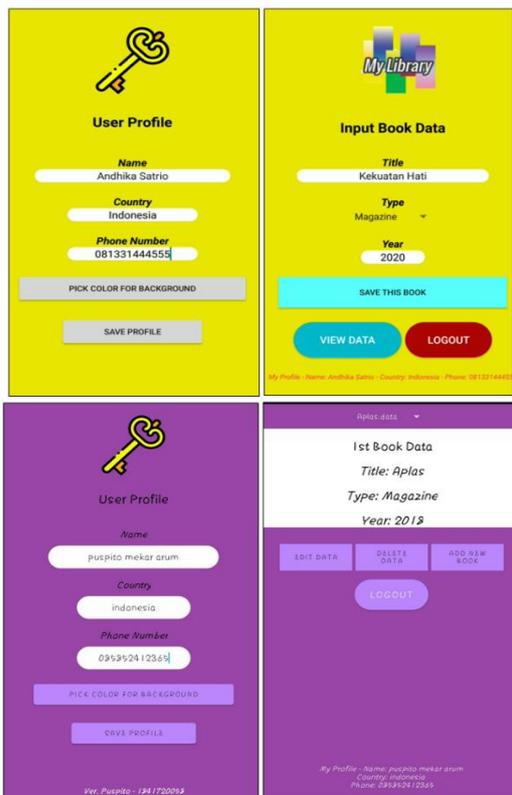
menyelesaikan semua tugas. Setiap mahasiswa memiliki kesempatan kedua untuk mengirimkan jawaban setelah 1 minggu jika gagal atau *error*.

5.2 Penyelesaian Hasil Kerja

Pada kesempatan pertama, 50 mahasiswa berhasil menyelesaikan semua tugas, yang berarti semua mahasiswa mencapai tujuan pembelajaran dengan mengembangkan aplikasi *MyLibrary* sebelum satu minggu. Meskipun *Basic Data Storage* adalah topik pertama pada tahap kedua, mahasiswa merasa lebih akrab belajar dengan APLAS setelah menyelesaikan lima topik sebelumnya.

5.3 Hasil dari Aplikasi *MyLibrary*

Dalam topik ini, mahasiswa memiliki fleksibilitas untuk merancang UI aplikasi. *Test code* yang diberikan fokus pada fungsi pada aplikasi *MyLibrary*. Selama evaluasi, kami mengumpulkan beberapa aplikasi *MyLibrary* dari mahasiswa. Gambar 4 menunjukkan beberapa contoh aplikasi *MyLibrary* dari mahasiswa. Bila dibandingkan dengan Gambar 3, ada beberapa penampilan yang berbeda di tiga *Activity*, seperti warna latar belakang, ukuran gambar, jenis font, dan warna tombol.



Gambar 4 Contoh Aplikasi *MyLibrary* dari Mahasiswa

5.4 Waktu Penyelesaian Setiap Langkah

Para mahasiswa harus melaporkan durasi untuk menyelesaikan tugas di setiap tugas. Tabel 3

menyajikan waktu rata-rata, waktu tersingkat, dan terlama untuk menyelesaikan tugas di masing-masing tugas. Total waktu untuk menyelesaikan semua langkah berkisar antara 76 menit sampai 573 menit, dengan rata-rata 185 menit. Dibandingkan dengan hasil topik sebelumnya, hasil ini sedikit lebih tinggi karena topik ini berisi lebih banyak tugas. Dengan kompleksitasnya memperkenalkan konsep penyimpanan data, mahasiswa membutuhkan lebih banyak waktu untuk menyelesaikan tugas.

Mahasiswa diharapkan lebih mudah untuk menyelesaikan soal pada topik selanjutnya setelah berhasil mengembangkan aplikasi *MyLibrary*.

Tabel 3 Waktu Penyelesaian Setiap Langkah

No	Waktu Tercepat	Waktu Rata-Rata	Waktu Terlama
1	5 menit	14.53 menit	30 menit
2	6 menit	15.93 menit	45 menit
3	6 menit	18.34 menit	54 menit
4	6 menit	16.37 menit	48 menit
5	7 menit	20.69 menit	50 menit
6	7 menit	17.66 menit	47 menit
7	6 menit	19.69 menit	60 menit
8	6 menit	19.89 menit	60 menit
9	10 menit	20.63 menit	60 menit
10	7 menit	19.11 menit	60 menit

5.5 Tingkat Kesulitan

Berdasarkan Tabel 3, tingkat kesulitan dapat dianalisis dengan waktu penyelesaian rata-rata. Langkah termudah untuk diselesaikan adalah tugas 1, yang berfokus pada memulai proyek Android dan mengonfigurasi *resource* aplikasi. Langkah yang di tingkat sedang adalah langkah 2, langkah 3, langkah 4, dan langkah 6 yang memakan waktu lebih sedikit dari 19 menit untuk menyelesaikan tiap langkah. Langkah-langkah ini dimaksudkan untuk membangun *UI layout* dari tiga *Activity* dan menulis kode Java untuk adaptor data. Beberapa langkah yang berada di tingkat sulit adalah langkah 5, langkah 7, langkah 8, langkah 9, dan langkah 10. Langkah-langkah ini terkait dengan cara menangani data di *Shared Preference* dan *Internal Storage*. Tabel 4 menunjukkan kategori kesulitan tugas.

Tabel 4 Tingkat Kesulitan Langkah Penyelesaian

Tingkat	Waktu Penyelesaian	Nomor Tugas
Mudah	<15 menit	1
Sedang	15 - 19 menit	2,3,4,6
Sulit	>19 menit	5,7,8,9,10

5.6 Hasil Umpan Balik Mahasiswa

Para mahasiswa juga mengirimkan komentar untuk topik ini terkait dengan pendapat tentang proses

pembelajaran dan penyelesaian tugas. Pendapat didominasi oleh komentar positif dan beberapa komentar sugestif. Berikut adalah beberapa komentar utama yang berhasil dikumpulkan:

- 1) Komentar positif: “materi pembelajaran yang disediakan mudah dipahami”, “pembelajaran mudah dipraktikkan”, “materi mudah dipahami”, “puji Tuhan, saya bisa melakukannya”.
- 2) Komentar sugestif: “pengguna perlu ketelitian dalam membaca instruksi”, “saya mendapatkan masalah saat mendefinisikan *Data Binding*”

Komentar sugestif tersebut akan dikembangkan untuk meningkatkan penjelasan dan petunjuk dalam dokumen panduan, khususnya terkait dengan pendefinisian *Data Binding* secara rinci.

6. Kesimpulan

Penelitian ini berfokus pada implementasi dan evaluasi materi pembelajaran yang diusulkan untuk topik pertama pada tahap pembelajaran ketiga di APLAS, yaitu topik *Basic Data Storage* yang mempelajari penyimpanan data dalam aplikasi Android menggunakan *Shared Preferences* dan *Internal Storage* berdasarkan arsitektur MVVM.

Berdasarkan hasil pengujian yang dilakukan kepada 50 mahasiswa S1 jurusan Teknologi Informasi, mahasiswa berhasil menyelesaikan pembelajaran *Basic Data Storage* sebelum satu minggu. Rata-rata mahasiswa dapat menyelesaikan pembelajaran 185 menit. Dibandingkan dengan hasil topik sebelumnya, hasil ini sedikit lebih tinggi karena topik ini lebih kompleks sehingga mahasiswa membutuhkan lebih banyak waktu untuk menyelesaikan tugas.

Daftar Pustaka:

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 (IJEAIS), vol. 2, Issue 1, pp. 1-14.

Aljamea, M., & Alkandari, M. (2018), *MMVMi: A validation model for MVC and MVVM design patterns in iOS applications*, IAENG International Journal of Computer Science, vol. 45, no. 3, pp. 377–389.

Almeida, D. R., Machado, P. D. L., & Andrade, W. L. (2019), *Testing tools for Android context-aware applications: a systematic mapping*, Journal of the Brazilian Computer Society, vol. 25, no. 12, pp. 1-22.

Farcic, V., & Garcia, A. (2015), *Test-Driven Java Development* (2nd ed.), Pack Publishing.

Google Developers. (2021a), *Data Binding Library*, <https://developer.android.com/topic/libraries/data-binding>

Google Developers. (2021b), *Shared Preferences*, <https://developer.android.com/reference/android/content/SharedPreferences>

Götz, F. M., Stieger, S., & Reips, U.-D. (2017), *Users of the main smartphone operating systems (iOS, Android) differ only little in personality*, PLOS ONE, vol. 12, no. 5, pp. 1-18.

Horton, J. (2015), *Android Programming for Beginners*, Packt Publishing.

Jain, M., & Gopalani, D. (2017), *Automated Java Testing: JUnit versus AspectJ*, Int. J. Com. Inf. Eng., vol. 11, no. 11, pp.1153-1158.

JUnit. (2021, January), *JUnit: a simple framework to write repeatable tests*, <https://junit.org/junit4/>

Lyu, Y., Gui, J., Wan, M., & Halfond, W. G. J. (2017), *An Empirical Study of Local Database Usage in Android Applications*, 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 444–455.

Ohkawa, Y., Kodama, M., Konno, Y., Zhao, X., & Mitsubishi, T. (2018), *A study on UI design of smartphone app for continuous blended language learning*, 2018 5th International Conference on Business and Industrial Research (ICBIR), pp. 584–589.

Phillips, B., Stewart, C., & Marsicano, K. (2017), *Android Programming: The Big Nerd Ranch Guide* (3rd ed.), Big Nerd Ranch Guides.

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

Sun, W., Chen, H., & Yu, W. (2016), *The Exploration and Practice of MVVM Pattern on Android Platform*, Proceedings of the 2016 4th International Conference on Machinery, Materials and Information Technology Applications, Xi'an, China.

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., 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.