

BAB V

KESIMPULAN DAN REKOMENDASI

Pada Bab V ini akan dipaparkan mengenai hasil dari penelitian ini. Hasil yang didapatkan merupakan hal yang menjawab dari rumusan masalah penelitian beserta penemuan lainnya yang didapatkan dari penelitian ini.

5.1 Kesimpulan

Terdapat berbagai penemuan dari penelitian ini. Berdasarkan analisis dari hasil penelitian dan pembahasan dari pengujian performa arsitektur MVP dan MVVM, didapatkan beberapa kesimpulan sebagai berikut:

1. Aplikasi chatbot deteksi kecanduan pornografi berbasis Android telah dibuat dan diimplementasikan arsitektur Model-View-ViewModel (MVVM) dan Model-View-Presenter (MVP). Setiap bagian pembangun arsitektur MVVM seperti penggunaan ViewModel, pemanfaatan variabel LiveData bersamaan dengan ViewModel, konfigurasi DataBinding, serta implementasi interaksi data dengan antarmuka dari ViewModel kepada antarmuka dengan DataBinding telah diimplementasikan.
2. Berdasarkan hasil uji T pada hasil pengujian empiris untuk komparasi kecepatan *startup* dan penggunaan sumber daya untuk arsitektur MVP dan MVVM, kemunculan perbedaan signifikan hanya pada metrik kecepatan *startup* tipe *warm start* pada emulator, penggunaan memori pada emulator, dan konsumsi energi di emulator. Hal ini memperlihatkan bahwa performa arsitektur MVVM signifikan lebih baik pada metrik tersebut.
3. Hasil pengujian empiris pada penggunaan sumber daya menunjukkan arsitektur MVVM unggul dari arsitektur MVP pada setiap metrik penggunaan sumber daya, diantaranya adalah 80% unggul pada pengujian penggunaan CPU untuk pengujian di emulator maupun perangkat Redmi Note 5, 80% unggul pada penggunaan memori untuk pengujian di emulator dan 70% unggul pada pengujian di Redmi Note 5, serta 70% unggul pada konsumsi energi untuk pengujian di emulator dan 50% pada pengujian di Redmi Note 5.

4. Hasil pengujian kecepatan membuka aplikasi menunjukkan bahwa arsitektur MVVM lebih cepat dari arsitektur MVP pada setiap tipe startup, kondisi startup, maupun perangkat yang berbeda. Perbandingan kecepatan *cold start* arsitektur MVVM dan MVP untuk mencapai TTFD pertama adalah 62% lebih cepat di emulator dan 29% lebih cepat di Redmi Note 5, serta untuk menyelesaikan *startup* aplikasi adalah 131% lebih cepat di emulator dan 7% lebih cepat di Redmi Note 5. Perbandingan kecepatan *warm startup* untuk mencapai TTFD pertama adalah 27% lebih cepat di emulator dan 20% di Redmi Note 5, serta untuk menyelesaikan *startup* aplikasi adalah 38% lebih cepat di emulator dan 47% lebih cepat di Redmi Note 5.
5. Ditemukan bahwa terdapat perbedaan antara ukuran aplikasi dengan arsitektur MVP dan MVVM, baik untuk ukuran paket aplikasi sebelum dipasang (APK) maupun ukuran aplikasi saat setelah dipasang pada perangkat. Nilai yang didapatkan adalah paket aplikasi dengan arsitektur MVVM memiliki 0,92% ukuran lebih besar dari ukuran paket dengan arsitektur MVP. Selain itu, ukuran aplikasi setelah terpasang untuk arsitektur MVVM didapatkan 0,76% hingga 2,04% lebih besar dari aplikasi dengan arsitektur MVP.

5.2 Rekomendasi

Hasil dari penemuan yang didapatkan dari penelitian ini memperlihatkan bagaimana arsitektur MVP dan MVVM memiliki perilaku yang berbeda pada sisi performa. Berdasarkan pada metode pengujian yang digunakan yaitu ATAM, kedua arsitektur yang diuji dapat diambil kelebihan serta kekurangannya atau tradeoffs. Hal tersebut bisa sangat bermanfaat untuk pengembang aplikasi Android dalam menentukan arsitektur mana yang perlu digunakan untuk proyek pengembangan aplikasinya.

Menurut peneliti, hasil penelitian memperlihatkan beberapa hal yang dapat dipertimbangkan dalam menentukan penggunaan arsitektur oleh pengembang aplikasi Android. Hal yang paling mendasar adalah dengan memerhatikan pada hasil uji T untuk setiap metrik uji yang dilakukan. Dari hasil tersebut, perbedaan signifikan hanya ada pada pengujian di emulator untuk metrik kecepatan *startup* tipe *warm start*, penggunaan memori, dan konsumsi energi. Tidak ditemukan adanya perbedaan signifikan pada pengujian di Redmi Note 5. Hal ini dapat

memerlihatkan bahwa perbedaan signifikan hanya muncul pada perangkat dengan performa yang optimal untuk menjalankan aplikasi. Tentunya, hal ini memerlukan konfirmasi lebih lanjut dengan adanya pengujian hingga penelitian selanjutnya dengan menggunakan perangkat uji yang lebih bervariasi.

Dari perbandingan untuk pertimbangan penggunaan arsitektur tersebut, pemerhatian terhadap kelebihan dan kekurangan atau *tradeoffs* perlu diperhatikan lebih rinci. Contoh dari hasil pengujian di penelitian ini adalah penggunaan arsitektur MVVM yang secara umum memiliki performa lebih baik dari arsitektur MVP, namun memiliki ukuran aplikasi yang lebih besar dari arsitektur MVP. Namun juga, untuk mengetahui kenyataan yang lebih luas mengenai arsitektur MVP dan MVVM pada aplikasi Android, diperlukan penelitian yang menguji performa dengan metrik yang lebih banyak serta perangkat sebagai lingkungan pengujian yang lebih bervariasi. Dengan hal tersebut, diharapkan aplikasi berbasis Android yang dikembangkan ke depannya dapat semakin inovatif dalam membantu pekerjaan pengguna namun tetap menjaga efisiensi penggunaan sumber dayanya upaya untuk menjaga pengalaman pengguna yang baik.