DAFTAR PUSTAKA

Almeida, G. A. (2020). *Concurrency in Android development-Kotlin Coroutines and RxJava*.

Android Developers. 2019. Mengenal Android Studio https://developer.android.com/studio/intro/index.html?hl=id, Diakses pada 05 Juni 2019.

Android Developers. (2022a, September 14). *Guide to app architecture*. https://developer.android.com/topic/architecture

Android Developers. (2022b, September 19). *Kotlin coroutines on Android*. https://developer.android.com/kotlin/coroutines

Baron, Marvin Munoz. (2019). *A Validation of Cognitive Complexity as a Measure of Source Code Understandability*

Bonér, J., Farley, D., Kuhn, R., & Thompson, M. (2023, Januari 15). *The Reactive Manifesto*. https://www.reactivemanifesto.org/

Campbell, G. Ann. (2023). *{Cognitive Compmaslexity} a new way of measuring understandability*

C. Arriola and A. Huang, Reactive Programming on Android with RxJava, MYNAH Software, 2017.

Chauhan, K., Kumar, S., Sethia, D., & Alam, M. N. (2021, Mei 21). Performance contanalysis of kotlin coroutines on android in a model-view-intent architecture pattern. *2021 2nd International Conference for Emerging Technology, INCET 2021*. https://doi.org/10.1109/INCET51464.2021.9456197

Davis, A. L. (2019). Reactive Streams in Java. Dalam *Reactive Streams in Java*. Apress. https://doi.org/10.1007/978-1-4842-4176-9

Dobslaw, F., Vallin, M., & Sundstrom, R. (2020). Free the Bugs: Disclosing Blocking Violations in Reactive Programming. *Proceedings - 20th IEEE International Working Conference on Source Code Analysis and Manipulation, SCAM 2020*, 177–186. https://doi.org/10.1109/SCAM51674.2020.00025

Elizarov, R. (2021). *Kotlin Flow Documentation*. JetBrains. https://kotlinlang.org/docs/flow.html

Ganglani, Himanshu. (2023, Agustus 24). Clean Code: Cognitive Complexity by SonarQube. https://medium.com/@himanshuganglani/clean-code-cognitive-

complexity-by-sonarqube 659d49a6837d#:~:text=Cognitive%20Complexity%2C%20a%20key%20metric,contribute%20to%20higher%20cognitive%20complexity.

Hunold, S., Ajanohoun, J. I., & Carpen-Amarie, A. (2021). MicroBench Maker: Reproduce, Reuse, Improve. *2021 International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, 69–74. https://doi.org/10.1109/PMBS54543.2021.00013

Joyo, D., Rajabi, N., Anggara P, R., & Rahmani, A. (2020). *Prosiding The 11 th Industrial Research Workshop and National Seminar Bandung*.

Kadek, N., Dewi, C., Bagus, I., Anandita, G., Atmaja, K. J., Aditama, P. W., & Magister, P. S. (2018). RANCANG BANGUN APLIKASI MOBILE SISKA BERBASIS ANDROID. *SINTECH Journal | 100 SINTECH JOURNAL*, *1*(2), 100–107. https://doi.org/10.31598/sintechjournal.v1i2.291

Komolov, S., Askarbekuly, N., & Mazzara, M. (2020). An empirical study of multi-Threading paradigms Reactive programming vs continuation-passing style. *ACM International Conference Proceeding Series*, 37–41. https://doi.org/10.1145/3418688.3418695

Lenarduzzi, V., Lomio, F., Huttunen, H., & Taibi, D. (2020). Are SonarQube Rules Inducing Bugs? *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 501–511. https://doi.org/10.1109/SANER48275.2020.9054821

Mulyati, S., & Wardono. (2019). Kreativitas Matematis Siswa Pada Pembelajaran Discovery Learning Dengan Media Berbasis Android Studio. PRISMA, Prosiding Seminar Nasional Matematika 2. *PRISMA*, *2*, 788–797. https://journal.unnes.ac.id/sju/index.php/prisma/

Olowonmi, A., Lai, P., & Xu, S. (2019). Enhancing Android Application Performance Using Reactive Programming: A Case Study. *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, 507–516. https://doi.org/10.1109/compsac46355.2019

Project Reactor. (2022, Oktober 3). *Create Efficient Reactive Systems*. Project Reactor. https://projectreactor.io/

Project Reactor Documentation. (2021). Project Reacto. https://projectreactor.io/

R. K. Panchal dan A. K. Patel, (2017). A comparative study: Java Vs Kotlin Programming in Android, Int ernational Journal of Innovative Trends in Engineering & Research, vol. 2, no. 9

Saini, S., Sahu, S., & Pani, A. K. (2018). A comparative analysis of android application execution time on different devices. *International Journal of Computer Applications*, *180*(36), 28–33.

Sarkar, A., Goyal, A., Hicks, D., Sarkar, D., & Hazra, S. (2019). Android Application Development: A Brief Overview of Android Platforms and Evolution of Security Systems. *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 73–79. https://doi.org/10.1109/I-SMAC47947.2019.9032440

Sarker, U., Islam, M. R., & Dey, S. K. (2020). Efficient and Responsive Android App Development Using Kotlin Flow. *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–6. https://doi.org/10.1109/ICCCNT49239.2020

S. Doug. (2010). High Performance Android Apps, Sebastopol: O'Reilly Media, Inc,

Shah, K., Shetty, S., & Kulkarni, R. (2021). Reactive Programming with Reactor for Android App Development. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *7*(2), 336–343.

Shah, Y., Shah, J., & Kansara, K. (2018). Code obfuscating a Kotlin-based App with Proguard. *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, 1–5. https://doi.org/10.1109/ICAECC.2018.8479507

SonarQube source. (2024). Lines of Code. SonarQube. https://docs.sonarsource.com/sonarqube/9.9/instance-administration/lines-of-code/

T. Subonis, Reactive Android programming. Packt Publishing Ltd., 2017, pp. 23-35.

Verma, N., Kansal, S., & Malvi, H. (2018). Development of Native Mobile Application Using Android Studio for Cabs and Some Glimpse of Cross Platform Apps. *International Journal of Applied Engineering Research*, *13*, 12527–12530. http://www.ripublication.com

Verawati, (2019). PEMANFAATAN ANDROID DALAM DUNIA PENDIDIKAN : Universitas PGRI Palembang

Volodko, E. (2021, Februari). *Kotlin for Android*. https://kotlinlang.org/docs/android-overview.html#0

Wang, X., Lo, D., Shihab, E., IEEE Computer Society, IEEE Computer Society. Technical Council on Software Engineering, & Institute of Electrical and Electronics Engineers. (2019). Challenges of SonarQube Plug-In Maintenance. *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 574–578. https://doi.org/10.1109/SANER.2019.8667988

Weber, L.M., Saelens, W., Cannoodt, R. et al. Essential guidelines for computational method benchmarking. Genome Biol 20, 125 (2019). https://doi.org/10.1186/s13059-019-1738-8

Wei, F., Roy, S., Ou, X., & Robby. (2018). Amandroid: A precise and general inter-component data flow analysis framework for security vetting of android apps. *ACM Transactions on Privacy and Security*, *21*(3). https://doi.org/10.1145/3183575

Yoon, H. J. 2012. A Study on the Performance of Android Platform, International Journal on Computer Science and Engineering, 4(4). 532-537. Retrieved from http://www.enggjournals.com/ijcse/doc/IJCSE-04-04- 128.pdf

Lampiran 1 *Source Code*

Source code penelitian tersedia untuk publik Repository pada tautan berikut:

https://github.com/ichsanulalifwan/final-project-rxjava

https://github.com/ichsanulalifwan/final-project-flow

https://github.com/ichsanulalifwan/final-project-reactor