**TUGAS AKHIR**

RANCANG BANGUN ALAT PENGUBAH KOTLIN DATA CLASS Ke PROTOCOL BUFFERS MESSAGE

****

**DISUSUN OLEH:**

**FATKHI NUR AKHSAN**

**NIM. 20106050026**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA  
YOGYAKARTA   
2024**

INTISARI

ABSTRACT

KATA PENGANTAR

DAFTAR ISI

DAFTAR GAMBAR

# PENDAHULUAN

## LATAR BELAKANG

Perkembangan teknologi dalam beberapa dekade terakhir sangatlah pesat, salah satu teknologi yang berkembang pesat adalah perangkat komputasi mobile dan smartphone dalam 3 dekade terakhir. Saat ini perangkat genggam sudah memiliki daya komputasi yang sama besarnya dengan ruangan yang penuh dengan komputer pada tahun 1970-an. Perkembangan daya komputasi tersebut tidak terlepas dari peningkatan tajam jumlah transistor yang dapat ditampung dalam sebuah *chip* akan berlipat ganda setiap dua tahun, sesuai dengan hukum Moore [1].

Meskipun memiliki kecepatan proses dan konektivitas yang masih lebih lambat dibandingkan dengan perangkat dekstop, penggunaan perangkat mobile dan aplikasi mobile lebih dipilih oleh pengguna dibandingkan dengan perangkat dekstop terutama untuk menyelesaikan pekerjaan yang mudah [2]. Salah satunya yaitu aplikasi *mobile-commerce* yang marak digunakan ketika masa pandemi COVID-19, penggunaan aplikasi *mobile-commerce* juga diprediksi masih berlanjut saat periode endemi [2]. Senada dengan hal itu, laporan dari Cisco menunjukkan bahwa hampir 300 miliar aplikasi seluler akan diunduh pada tahun 2023, dengan aplikasi berjenis media sosial, game, dan bisnis akan menjadi unduhan yang paling populer [3].

Diantara penggunaan masif perangkat mobile terdapat dua sistem operasi yang sudah mendominasi pasar dengan cakupan 90% dari market, yaitu Android (Google, Mountain View, CA) and iOS (Apple) sejak tahun 2008 [1]. Untuk mendukung produktifitas developer, Google mengumumkan dukungannya pada Google I/O 2017 untuk menjadikan bahasa pemrograman Kotlin sebagai bahasa resmi untuk pengembangan aplikasi android [4]. Setelah dua tahun berselang, tepatnya saat Google I/O 2019, pengembangan aplikasi mobile android akan menjadi *kotlin-first* setelah Google mengumumkannya, hal ini berarti Kotlin akan menjadi pilihan pertama Google dalam mengembangkan alat dan konten pengembangan aplikasi android [5].

Bahasa pemrograman Kotlin sendiri merupakan bahasa pemrograman open source berjenis statis yang mendukung pemrograman berorientasi objek dan fungsional [6]. Kotlin pada mulanya dirancang oleh JetBrains, kemudian terus dikembangkan oleh para kontributor dan dikelola oleh Kotlin Foundation yang didirikan oleh JetBrains dan Google [7]. Meskipun tergolong bahasa yang modern, Kotlin juga dapat dikatakan bahasa yang sudah matang. Bahasa ini ringkas, aman, dapat dioperasikan dengan Java dan bahasa lain, dan menyediakan banyak cara untuk menggunakan kembali kode di antara berbagai platform untuk pemrograman yang produktif [8]. Oleh karena itu, beberapa API pada Android seperti AndroidKTX yang dikhususkan untuk kotlin tetap dapat kompatibel dengan Bahasa Java yang sudah lebih dulu menjadi bahasa resmi untuk pengembangan aplikasi android. Dengan interoperabilitas yang baik dengan Java tersebutlah, Kotlin berhasil menghasilkan pengalaman pengembangan aplikasi android yang lebih nyaman [6].

Salah satu library yang memiliki AndroidKTX sebagai ekstensi adalah library Jetpack DataStore. Jetpack DataStore adalah salah satu library yang menyediakan solusi penyimpanan data lokal yang terdapat pada platform android, khususnya untuk kumpulan data yang kecil dan sederhana dan tidak memerlukan pembaruan parsial atau integritas referensial. Jetpack DataStore dapat menyimpan data berupa pasangan *key*-*value* atau objek yang ditulis menggunakan protocol buffers. Oleh karena itu, Jetpack DataStore dibagi menjadi dua jenis, yaitu Preferences DataStore, yang menyimpan dan mengakses data menggunakan kunci dan Proto DataStore, yang menyimpan data berupa instansiasi objek yang sebelumnya telah didefinisikan menggunakan schema yang ditulis menggunakan protocol buffers. Jetpack DataStore menggunakan Kotlin Coroutines dan Flow untuk menyimpan data secara asinkron, konsisten, dan transaksional [9].

Protocol Buffers atau yang juga dapat disingkat dengan protobuf, merupakan mekanisme yang dapat diekstensi untuk membuat serialisasi data terstruktur yang *language-netral* dan *platform-netral*. Protocol Buffers sendiri unggul dalam hal ukuran dan kecepatan dibandingkan dengan format data lain seperti JSON dan XML, dan dapat menghasilkan binding bahasa asli. Protocol Buffers adalah kombinasi bahasa definisi (dibuat dalam file .proto), kode yang dibuat oleh kompiler proto untuk berinteraksi dengan data, pustaka runtime yang *language-specific*, format serialisasi untuk data yang ditulis ke file (atau dikirim melalui koneksi jaringan), dan data yang terserialisasi [10].

Salah satu cara untuk menampung data pada bahasa pemrograman kotlin adalah dengan menggunakan Data Class yang ditandai dengan keyword “data”. Untuk setiap Data Class, kompiler akan secara otomatis menghasilkan fungsi tambahan yang memungkinkan untuk mencetak instance ke output yang dapat dibaca, membandingkan instance, menyalin instance, dan banyak lagi. Hal ini mengurangi kode *boilerplate* yang dihasilkan dibandingkan dengan mekanisme penampungan data yang terdapat pada bahasa pemrograman lain, seperti POJO pada bahasa Java [11]. Sama halnya dengan Data Class, Protocol Buffers juga erat berkaitan dengan data-data terutama untuk mendefinisikan protokol komunikasi (bersama dengan gRPC) dan untuk penyimpanan data. Struktur data pada Protocol Buffers didefinisikan di file-file dengan ekstensi .proto yang dibuat oleh engineer. Pada file tersebut service-service dan message-message didefinisikan. Message inilah yang dijadikan schema untuk memperoleh kode dan runtime library yang lebih *language-specific* sehingga dapat digunakan pada berbagai bahasa dan *framework* pemrograman, khususnya dalam konteks ini yaitu untuk menyimpan data menggunakan Jetpack Proto DataStore [10].

Meskipun keduanya, yaitu Kotlin Data Class dan Protocol Buffers memiliki beberapa persamaan. Namun, keduanya memiliki perbedaan sintaks, hal ini menjadi permasalahan dan tantangan tersendiri bagi developer android yang ingin menggunakan Jetpack Proto DataStore sebagai solusi penyimpanan lokal pada aplikasi yang dikembangkan. Semakin kompleks struktur data yang dipakai tentunya akan menambahkan tingkat kesulitan yang dialami developer dalam merepresentasikannya kedalam bentuk skema yang dibuat dari Protocol Buffers Message. Bahkan, ketika struktur data tersebut tidaklah kompleks pengembang yang belum pernah memakai Jetpack Proto DataStore tetap perlu untuk mempelajari Protocol Buffers. Proses pembuatan skema inilah yang seringkali memakan waktu dan rentan terhadap kesalahan, terutama ketika berhadapan dengan struktur data yang rumit dikarenakan prosesnya yang masih manual [12]. Permasalahan tersebutlah yang menciptakan kebutuhan akan sebuah alat yang dapat mengotomatisasi proses perubahan struktur data yang berasal dari Kotlin Data Class menjadi struktur data berupa skema yang didefinisikan melalui Protocol Buffers Message dengan cepat dan efisien. Kebutuhan tersebut sangat penting bagi pengembang yang perlu bekerja dengan data yang sudah ada dalam bentuk struktur data Kotlin Data Class kemudian ingin memperolehnya dalam bentuk Protocol Buffers Message dengan cepat dan efisien.

Untuk mengatasi permasalahan dan memenuhi kebutuhan dari pengembang tersebut, diperlukan sebuah alat yang dapat mengubah struktur atau model data dari Kotlin Data Class ke dalam bentuk Protocol Buffers Message secara otomatis. Alat tersebut diharapkan dapat mengatasi permasalahan dan memenuhi kebutuhan sekaligus menghadirkan beberapa manfaat bagi para pengembang, terutama pengembang aplikasi android, seperti: menyederhanakan dan mempercepat proses pembuatan Protocol Buffers Message sebagai skema data yang sebelumnya sudah pernah dibuat dalam bentuk Kotlin Data Class, meminimalisir kesalahan yang mungkin terjadi selama proses perubahan dan pembuatan skema, memudahkan pengembang dalam memaintain kode program [13]. Penggunaan alat konversi otomatis juga dapat meningkatkan konsistensi dalam struktur kode sehingga praktik-praktik kode yang baik dapat dijalankan dengan lebih mudah, memfasilitasi pembaruan yang lebih mudah ketika skema data berubah, dan memungkinkan pengembang untuk lebih berfokus pada logika bisnis inti daripada tugas-tugas transformasi data yang repetitif [14].

Hal-hal tersebutlah yang memotivasi penulis untuk merancang dan membangun alat yang dapat mengubah Kotlin Data Class menjadi Protocol Buffers Message secara otomatis dalam bentuk plugin yang dapat dipasang dan berjalan pada Intellij IDEA dan Android Studio, dimana Intellij IDEA sendiri merupakan Integrated Development Environment (IDE) terdepan dalam pengembangan yang menggunakan bahasa Kotlin [15], kemudian Android Studio sebagai IDE Resmi untuk mengembangkan aplikasi Android [16].

## RUMUSAN MASALAH

Dari latar belakang yang telah diuraikan, masalah dapat dirumuskan sebagai berikut: “Bagaimana cara merancang dan membangun alat yang dapat mengubah struktur data pada Kotlin Data Class suatu skema yang didefinisikan melalui Protocol Buffers Message?”

## TUJUAN

Dari rumusan masalah yang telah disebutkan, tujuan yang ingin dicapai adalah untuk merancang dan membangun alat yang dapat mengubah struktur data pada Kotlin Data Class suatu skema yang didefinisikan melalui Protocol Buffers Message berupa plugin yang dapat dipasang dan dijalankan pada Intellij IDEA dan Android Studio.

## MANFAAT

Hasil dari rancang bangun ini diharapkan dapat memberikan beberapa manfaat sebagai berikut:

* Membantu pengembang aplikasi android yang menggunakan bahasa pemrograman kotlin dalam membangun aplikasi android yang menggunakan *library* Jetpack Proto DataStore menjadi lebih efisien, cepat, dan mudah dipelihara.
* Meningkatan produktivitas *developer* dengan mempercepat proses pembuatan skema pada Protocol Buffers melalui proses perubahan yang diperoleh dari struktur data pada Kotlin Data Class, menghemat waktu *developer* sehingga dapat berfokus pada logika bisnis inti, dan meningkatkan efisiensi workflow.
* Berkontribusi pada pengembangan bahasa pemrograman kotlin, aplikasi android, dan ekosistem keduanya.
* Menyediakan studi kasus, membuka peluang penelitian optimasi konversi Protocol Buffers Message dan pengembangan *tools*.

# KAJIAN PUSTAKA

## Kotlin

## Kotlin Data Class

## Protocol Buffers

## Protocol Buffers Message

## Plugin

# METODE PENGEMBANGAN SISTEM

## Bahan dan Alat

## Metode Pengembangan

## Tahapan Pengembangan

# PERANCANGAN DAN IMPLEMENTASI SISTEM

# KESIMPULAN DAN SARAN

## Kesimpulan

## Saran

DAFTAR PUSTAKA

1. B. S. Rothman, R. K. Gupta, and M. D. McEvoy, “Mobile Technology in the Perioperative Arena: Rapid Evolution and Future Disruption,” *Anesth. Analg.*, vol. 124, no. 3, pp. 807–818, 2017, doi: 10.1213/ANE.0000000000001858.
2. M. Ashoer, M. H. Syahnur, J. S. Tjan, A. Junaid, A. Pramukti, and A. Halim, “The Future of Mobile Commerce Application in a Post Pandemic Period; An Integrative Model of UTAUT2,” *E3S Web Conf.*, vol. 359, pp. 1–8, 2022, doi: 10.1051/e3sconf/202235905005.
3. “Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper,” Cisco, Jan. 2022. <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html> (accessed Jul. 04, 2024).
4. Google for Developers. Google I/O Keynote (Google I/O '17). (May 17, 2017). Accessed: Jul. 04, 2024. [Online video]. Available: <https://www.youtube.com/live/Y2VF8tmLFHw?si=zgCrETDalg1xflXH&t=5230>.
5. “Android’s Kotlin-first approach,” *Android Developers*, 2019. <https://developer.android.com/kotlin/first> (accessed Jul. 04, 2024).
6. “Ringkasan Kotlin,” *Android Developers*, 2023. <https://developer.android.com/kotlin/overview?hl=id> (accessed Jul. 04, 2024).
7. “Kotlin Foundation – official site,” *Kotlin Foundation – official site*, 2023. <https://kotlinfoundation.org/> (accessed Jul. 04, 2024).
8. “Kotlin Programming Language,” *Kotlin*, 2024. <https://kotlinlang.org/> (accessed Jul. 04, 2024).
9. “App Architecture: Data Layer - DataStore - Android Developers,” Android Developers, 2024. <https://developer.android.com/topic/libraries/architecture/datastore> (accessed Jul. 04, 2024).
10. Protocol Buffers, “Protocol Buffers,” *Protobuf.dev*, 2023. <https://protobuf.dev/> (accessed Jul. 04, 2024).
11. Kotlin, “Data classes,” Kotlin Documentation, 2024. <https://kotlinlang.org/docs/data-classes.html> (accessed Jul. 04, 2024).
12. M. E. Joorabchi, A. Mesbah, and P. Kruchten, “Real Challenges in Mobile App Development,” in *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, 2013, pp. 15–24, doi: 10.1109/ESEM.2013.9.
13. M. Voelter, *Generic Tools, Specific Languages*. Delft University of Technology, 2014.
14. M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*, vol. 1. 2012.
15. JetBrains, “IntelliJ IDEA,” *JetBrains*, 2024. <https://www.jetbrains.com/idea/> (accessed Jul. 04, 2024).
16. “Download Android Studio & App Tools - Android Developers,” *Android Developers*, 2024. <https://developer.android.com/studio> (accessed Jul. 04, 2024).