

TUGAS AKHIR

**RANCANG BANGUN ALAT KONVERSI OTOMATIS DARI
JSON KE DATA CLASS SERIALISASI KOTLIN**



DISUSUN OLEH:
MUHAMMAD FARHAN ANSHOR
NIM. 20106050048

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

LEMBAR PERNYATAAN

SURAT PERNYATAAN KEASLIAN/BEBAS PLAGIASI

Yang bertanda tangan di bawah ini:

Nama : Muhammad Farhan Anshor
NIM : 20106050048
Program Studi : Informatika

Menyatakan bahwa tugas akhir saya yang berjudul “Rancang Bangun Alat Konversi Dari JSON Ke *Serializable Data Class Kotlin Serialization*” merupakan hasil penelitian saya sendiri, tidak terdapat karya di suatu perguruan tinggi, dan sepanjang pengetahuan saya, tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 8 Juni 2024

Yang menyatakan



Muhammad Farhan Anshor

NIM. 20106050048

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

LEMBAR PENGESAHAN TUGAS AKHIR



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-1034/Un.02/DST/PP.00.9/07/2024

Tugas Akhir dengan judul : RANCANG BANGUN ALAT KONVERSI DARI JSON KE SERIALIZABLE DATA
CLASS KOTLIN SERIALIZATION

yang dipersiapkan dan disusun oleh:

Nama : MUHAMMAD FARHAN ANSHOR
Nomor Induk Mahasiswa : 20106050048
Telah diujikan pada : Rabu, 19 Juni 2024
Nilai ujian Tugas Akhir : A-

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR



Ketua Sidang

Ir. Muhammad Didik Rohmad Wahyudi, S.T., MT.
SIGNED

Valid ID: 6683302104e55



Penguji I

Dr. Ir. Sumarsono, S.T., M.Kom.
SIGNED

Valid ID: 66820350ba119



Penguji II

Eko Hadi Gunawan, M.Eng.
SIGNED

Valid ID: 66823e1f4214f



Yogyakarta, 19 Juni 2024
UIN Sunan Kalijaga
Dekan Fakultas Sains dan Teknologi
Prof. Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED

Valid ID: 668398e5c0ce7

LEMBAR PERSETUJUAN TUGAS AKHIR

SURAT PERSETUJUAN TUGAS AKHIR

Hal : Surat Persetujuan Tugas Akhir

Lamp : -

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

Assalamualaikum Wr. Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku dosen konsultan berpendapat bahwa tugas akhir saudara:

Nama : Muhammad Farhan Anshor

NIM : 20106050048

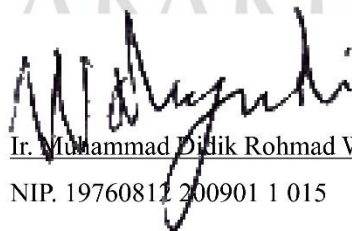
Judul Tugas Akhir : Rancang Bangun Alat Konversi Dari JSON Ke
Serializable Data Class Kotlin Serialization

Sudah dapat diajukan kembali kepada Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Informatika.

Dengan ini kami berharap agar tugas akhir tersebut dapat segera di-*munaqasyah*-kan. Atas perhatiannya kami ucapkan terima kasih.
Wassalamualaikum wr.wb

Yogyakarta, 10 Juni 2024

Yang menyatakan



Ir. Muhammad Didik Rohmad Wahyudi, S.T., MT.

NIP. 19760812 200901 1 015

Signer ID: 06F5RPMT12...

LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR

Tugas Akhir ini tidak dipublikasikan, tetapi tersedia di perpustakaan dalam lingkungan UIN Sunan Kalijaga Yogyakarta, diperkenankan dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seizin penyusun, dan harus menyebutkan sumbernya sesuai dengan kebiasaan ilmiah. Dokumen Tugas Akhir ini merupakan hak milik UIN Sunan Kalijaga Yogyakarta.



ABSTRAK

Perkembangan pesat teknologi *mobile* membuka peluang bagi *developer* untuk menciptakan aplikasi inovatif. Integrasi data dengan layanan backend menjadi tantangan utama, di mana JSON menjadi format data yang umum digunakan. Kotlin dan Kotlin Serialization menghadirkan fitur yang mendukung proses serialisasi dan deserialisasi data, namun masih terdapat kebutuhan untuk alat konversi data JSON ke *data class* yang dapat diserialisasi dalam Kotlin. Penelitian ini dilakukan untuk mengetahui bagaimana cara mengotomatisasi konversi data JSON ke *serializable data class* Kotlin. Penelitian ini menggunakan metode Agile *Extreme Programming* (XP) untuk merancang dan mengembangkan alat konversi JSON ke *data class* Kotlin dalam bentuk plugin yang mendukung Android Studio dan IntelliJ IDEA. Pengembangan alat konversi JSON ke *data class* Kotlin menunjukkan bahwa otomatisasi konversi data JSON ke *data class* Kotlin dapat menjadi solusi berharga bagi *developer* Kotlin. Plugin ini dapat mempercepat proses pengembangan aplikasi dan meningkatkan kualitas codebase. Alat konversi JSON ke *data class* Kotlin yang dikembangkan dalam penelitian ini terbukti mampu mengotomatisasi konversi data JSON dengan efisien dan akurat. Alat ini dapat membantu *developer* Kotlin dalam mengintegrasikan data JSON ke dalam aplikasi *mobile* mereka dengan lebih mudah dan cepat.

Kata Kunci: Konversi JSON, Data Class, Kotlin Serialization, Plugin, Pengembangan Aplikasi Kotlin

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

ABSTRACT

The surge in mobile app development necessitates efficient data integration with backend services, often relying on JSON. Kotlin and Kotlin Serialization provide features that support data serialization and deserialization, but there is still a need for a tool to convert JSON data to serializable data classes in Kotlin. This research was conducted to find out how to automate the conversion of JSON data to serializable data classes in Kotlin. This research used the Agile Extreme Programming (XP) methodology to design and develop a JSON to data class Kotlin conversion tool in the form of a plugin that supports Android Studio and IntelliJ IDEA. The development of a JSON to data class Kotlin conversion tool has shown that automating the conversion of JSON data to data classes in Kotlin can be a valuable solution for Kotlin developers. This plugin can help speed up the application development process and improve the overall quality of the codebase. The JSON to data class Kotlin conversion tool developed in this research has proven to be able to automate JSON data conversion efficiently and accurately. This tool can help Kotlin developers integrate JSON data into their mobile applications more easily and quickly.

Keywords: JSON Conversion, Data Class, Kotlin Serialization, Plugin, Kotlin Application Development



STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

HALAMAN PERSEMBAHAN

Saya persembahkan tugas akhir ini kepada kedua orang tua saya yang selalu mendoakan dan mendukung saya, serta telah membesarkan dan merawat penulis hingga dapat menempuh pendidikan tinggi dan menjadi pribadi yang lebih baik.



KATA PENGANTAR

Penulis bersyukur kepada Allah *Subhanahu wata'ala* karena berkat Rahmat dan Karunia-Nya, penulis dapat menyelesaikan tugas akhir ini yang berjudul "Rancang Bangun Alat Konversi dari JSON ke Kelas *Serializable Data Class* Kotlin." Sholawat dan salam selalu diberikan kepada junjungan kita Rasulullah SAW, yang telah menuntun kita ke zaman yang cerah, dan semoga kita semua mendapatkan syafaatnya di akhir zaman.

Penyelesaian tugas akhir ini membutuhkan usaha dan kerja keras dalam proses penyelesaiannya. Namun dalam proses penyelesaiannya tidak lepas dari bantuan dan dukungan berbagai pihak, secara langsung maupun tidak langsung. Dalam kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Prof. Dr.Phil. H. AlMakin, S.Ag., M.A. selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Prof. Dr. Dra. Hj. Khurul Wardati, M.Si. selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
3. Ibu Ir. Maria Ulfah Siregar, S. Kom, MIT. Ph. D., selaku Ketua Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Bapak Mandahadi Kusuma, M.Eng. selaku dosen penasihat akademik penulis selama kuliah di UIN Sunan Kalijaga.
5. Bapak Muhammad Didik Rohmad Wahyudi, S.T., MT. selaku pembimbing tugas akhir penulis yang sabar dan memberikan masukan selama penulisan dan penyusunan tugas akhir.
6. Seluruh Dosen dan Karyawan Program Studi Informatika UIN Sunan Kalijaga yang telah memberi ilmu dan bantuan selama perkuliahan penulis.
7. Kedua orang tua penulis bapak Drs. Ansor Putra, M. Hum, dan mama Umikun Latifah, SKM, M. Si. yang telah membesarkan dan merawat penulis hingga dapat menempuh pendidikan tinggi dan menjadi pribadi yang lebih baik.

8. Seluruh Angkatan Informatika terutama teman-teman angkatan 2020 yang telah menjadi teman seperjuangan dan saling mendukung pada saat perkuliahan.

Tugas akhir ini pasti tidak terlepas dari kekurangan kualitas dan kuantitas materi. Untuk itu, mohon kritik dan saran yang konstruktif untuk membantu penulis memperbaiki kekurangannya. Semoga tugas akhir ini dapat memberikan manfaat dan dapat dikembangkan untuk ke depannya.

Yogyakarta, 1 Juni 2024

Penulis

Muhammad Farhan Anshor

NIM. 20106050048

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

DAFTAR ISI

LEMBAR PERNYATAAN	i
LEMBAR PENGESAHAN TUGAS AKHIR.....	ii
LEMBAR PERSETUJUAN TUGAS AKHIR.....	iii
LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan.....	4
1.4 Manfaat.....	4
BAB II KAJIAN PUSTAKA	6
2.1 Converter.....	6
2.2 Kotlin.....	6
2.3 Data Class.....	7
2.4 Kotlin Serialization	7
2.5 JSON	8
2.6 Plugin	8
2.7 UML.....	9
2.8 Metode Extreme Programming	9
BAB III METODE PENGEMBANGAN SISTEM.....	12
3.1 Lokasi dan Waktu	12

3.2 Bahan dan Alat	12
3.3 Metode Pengembangan	12
3.4 Tahapan Pengembangan	13
BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM.....	14
4.1 Perencanaan (Planning)	14
4.1.1 Analisis kebutuhan	14
4.2 Perancangan (Design)	16
4.2.1 Use Case.....	16
4.2.2 Activity Diagram Konversi	17
4.2.3 Diagram Arsitektur	19
4.2.4 Package Diagram.....	20
4.2.5 Class Diagram	22
4.2.6 Sequence Diagram.....	31
4.2.7 Perancangan User Interface	34
4.3 Implementasi kode (Coding)	35
4.3.1 Iterasi Pengembangan Tahap 1	36
4.3.2 Iterasi Pengembangan Tahap 2	39
4.3.3 Iterasi Pengembangan Tahap 3	40
4.4 Pengujian (Testing)	43
4.4.1 Black Box Testing	43
4.4.2 Unit Testing	45
4.5 Panduan Pengguna	46
BAB V KESIMPULAN DAN SARAN.....	49
5.1 Kesimpulan.....	49
5.2 Saran.....	49
DAFTAR PUSTAKA	50
LAMPIRAN.....	53

DAFTAR GAMBAR

Gambar 2.5.1 Tahapan Metode XP	11
Gambar 4.2.1 Use Case	17
Gambar 4.2.2 Activity Diagram Konversi	18
Gambar 4.2.3 Diagram Arsitektur	19
Gambar 4.2.4 Package Diagram.....	20
Gambar 4.2.5 Kelas Diagram <i>Main</i>	22
Gambar 4.2.6 Diagram Kelas <i>Core</i>	24
Gambar 4.2.7 Kelas Diagram Generator	26
Gambar 4.2.8 <i>Seuqnce Diagram Action</i>	32
Gambar 4.2.9 Sequence Diagram Konversi	33
Gambar 4.2.10 User Interface	35
Gambar 4.3.1 Implementasi Action.....	37
Gambar 4.3.2 Implementasi User Interface	38
Gambar 4.3.3 Pesan Sukses.....	40
Gambar 4.3.4 Pesan Log Event.....	40
Gambar 4.3.5 Pesan Error	40
Gambar 4.4.1 Hasil Unit Testing	46
Gambar 4.5.1 Instalasi Plugin	47

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

DAFTAR TABEL

Tabel 4.1.1 Deskripsi Package	21
Tabel 4.3.1 Pengujian	43



DAFTAR LAMPIRAN

Lampiran 1 Link Github.....	53
Lampiran 2 Proses Perancangan Sistem Dengan Whimsical	53
Lampiran 3 Proses Perancangan Sistem Dengan draw.io	53
Lampiran 4 Proses Perancangan Sistem dengan Visual Paradigm	54
Lampiran 5 Proses Implementasi kode dengan IntelliJ IDEA.....	54



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa dekade terakhir, kita telah menyaksikan perkembangan luar biasa dalam teknologi. Terutama pada *smartphone* yang penggunaannya bertambah pesat setiap tahunnya. Menurut datareportal.com, sekitar 5,44 milyar pengguna *smartphone* pada awal 2023. Pengguna *smartphone* telah meningkat lebih dari 3 persen selama setahun terakhir, dengan 168 juta pengguna baru selama 12 bulan terakhir [1]. Dengan begitu pengembang aplikasi *mobile* juga berkembang seiring dengan bertambahnya pengguna. Hal ini membuka peluang besar bagi pengembang aplikasi *mobile* untuk menciptakan solusi digital yang inovatif dan bermanfaat bagi masyarakat.

Pengembangan aplikasi *mobile* yang pesat saat ini menuntut efisiensi dan kemudahan dalam pengelolaan data. Salah satu tantangan yang sering dihadapi *developer mobile* adalah integrasi data dengan layanan *backend*. Layanan *backend* sering kali mengembalikan data dalam format JSON (*JavaScript Object Notation*) yang ringan dan mudah dibaca. JSON menyediakan notasi sederhana untuk mengekspresikan koleksi pasangan nama/nilai. Sebagian besar bahasa pemrograman akan memiliki beberapa fitur untuk merepresentasikan koleksi seperti itu, yang dapat menggunakan nama seperti record, struct, dict, map, hash, atau objek [2]. Sederhana, fleksibel, dan independen platformnya telah membuat JSON menjadi pilihan yang populer untuk banyak aplikasi, termasuk pengembangan web, aplikasi *mobile*, dan analisis data.

Kotlin, sebuah bahasa pemrograman modern yang dikembangkan oleh JetBrains, telah menjadi semakin populer dalam pengembangan aplikasi Android sejak Google mengumumkannya sebagai bahasa yang didukung secara resmi pada tahun 2017 [3]. Popularitas Kotlin disebabkan oleh beberapa faktor kunci: sintaksisnya yang ringkas dan ekspresif, sistem tipe yang kuat dengan fitur null-safety, dan kompatibilitas penuh dengan ekosistem Java yang sudah ada [4]. Salah satu fitur Kotlin yang sangat berguna dalam pengembangan aplikasi adalah data

class. Data class menyediakan cara yang efisien untuk membuat kelas yang terutama digunakan untuk menyimpan data, dengan secara otomatis menghasilkan metode utilitas seperti `equals()`, `hashCode()`, dan `toString()`. Fitur ini sangat membantu dalam mengelola dan memanipulasi data dalam aplikasi, terutama ketika bekerja dengan data yang diterima dari API atau sumber eksternal lainnya [5].

Kotlin, sebagai bahasa pemrograman yang terus berkembang, telah menghadirkan solusi yang efisien untuk menangani serialisasi dan deserialisasi data melalui Kotlin Serialization. Library ini, yang dikembangkan oleh JetBrains, menawarkan pendekatan yang lebih idiomatik dan type-safe untuk mengonversi objek Kotlin ke dan dari format seperti JSON, dibandingkan dengan library serialisasi pihak ketiga [6]. Kotlin Serialization bekerja melalui plugin kompiler khusus yang menghasilkan kode serialisasi pada waktu kompilasi, meningkatkan keamanan tipe dan kinerja runtime [7]. Library ini mendukung berbagai format serialisasi, termasuk JSON, protobuf, dan CBOR, serta menawarkan fleksibilitas dalam kustomisasi proses serialisasi. Fitur-fitur canggih seperti polymorphic serialization dan support untuk struktur data yang kompleks membuatnya sangat cocok untuk proyek-proyek Kotlin modern, terutama dalam pengembangan aplikasi mobile dan backend [8]. Dengan Kotlin Serialization, pengembang dapat dengan mudah mendefinisikan kelas data yang dapat diserialisasi, memungkinkan konversi yang mulus antara objek Kotlin dan representasi JSON mereka, yang sangat penting dalam komunikasi client-server dan penyimpanan data lokal.

Sebelum munculnya Kotlin Serialization, pengembang Android sering menggunakan library GSON (Google-gson) untuk mengonversi JSON menjadi objek Java atau Kotlin. GSON, yang dikembangkan oleh Google, telah menjadi pilihan populer selama bertahun-tahun karena kemudahannya dalam penggunaan dan fleksibilitasnya [9]. Beberapa plugin konversi JSON untuk IDE seperti Android Studio juga mengandalkan GSON untuk menghasilkan kelas Java atau Kotlin dari struktur JSON. Namun, dengan berkembangnya ekosistem Kotlin, Kotlin Serialization muncul sebagai alternatif yang lebih cocok untuk proyek berbasis Kotlin.

Kotlin Serialization menawarkan beberapa keunggulan dibandingkan GSON, terutama untuk aplikasi Kotlin. Pertama, Kotlin Serialization adalah library yang dikembangkan khusus untuk Kotlin, sehingga terintegrasi dengan baik dengan fitur-fitur bahasa Kotlin seperti null-safety dan data classes. Kedua, Kotlin Serialization menggunakan kompilasi time code generation, yang dapat mendeteksi kesalahan serialisasi pada saat kompilasi, berbeda dengan GSON yang melakukan refleksi pada runtime. Hal ini meningkatkan keamanan tipe dan performa aplikasi. Ketiga, Kotlin Serialization mendukung serialisasi multiplatform, memungkinkan penggunaan kode yang sama untuk serialisasi pada platform JVM, JavaScript, dan Native. Terakhir, sintaks dan penggunaan Kotlin Serialization lebih intuitif dan "Kotlin-friendly", membuat kode lebih mudah dibaca dan dipelihara oleh pengembang Kotlin [10].

Meskipun Kotlin Serialization menawarkan berbagai keunggulan, pengembang masih menghadapi tantangan dalam proses transformasi data JSON yang kompleks ke dalam bentuk data class Kotlin. Proses manual ini seringkali memakan waktu dan rentan terhadap kesalahan, terutama ketika berhadapan dengan struktur JSON yang rumit [11]. Hal ini menciptakan kebutuhan akan alat yang dapat secara efisien mengonversi data JSON menjadi kelas data yang dapat diserialisasi dalam Kotlin. Kebutuhan ini sangat penting bagi pengembang yang perlu bekerja dengan data JSON yang sudah ada atau mengintegrasikan data JSON ke dalam aplikasi Kotlin mereka.

Untuk mengatasi permasalahan tersebut dan meningkatkan produktivitas pengembang, diperlukan sebuah alat konversi otomatis yang dapat mengubah data JSON ke dalam bentuk data class Kotlin. Alat semacam ini dapat memberikan beberapa manfaat signifikan: pertama, menyederhanakan dan mempercepat proses integrasi data JSON ke dalam aplikasi mobile Kotlin; kedua, meminimalisir kesalahan yang mungkin terjadi selama transformasi data manual; dan ketiga, meningkatkan maintainability kode program melalui code generation untuk pembuatan data class [12]. Penggunaan alat konversi otomatis juga dapat meningkatkan konsistensi dalam struktur kode, memfasilitasi pembaruan yang lebih mudah ketika skema JSON berubah, dan memungkinkan pengembang untuk

fokus pada logika bisnis inti daripada tugas-tugas transformasi data yang repetitif [13].

Tujuan dari tugas akhir ini adalah untuk merancang dan mengembangkan alat yang dapat mengonversi data JSON menjadi kelas data yang dapat diserialisasi dalam Kotlin menggunakan *Library* Serialisasi Kotlin dalam bentuk *plugin* yang *support* dengan Android Studio dan IntelliJ IDEA. Alat ini akan memberikan cara yang mudah dan efisien bagi pengembang untuk bekerja dengan data JSON dalam aplikasi Kotlin, memungkinkan mereka untuk memanfaatkan sistem tipe yang kuat dan kapasitas serialisasi Kotlin.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, dapat dirumuskan masalah sebagai berikut:

- Bagaimana cara merancang dan membangun alat yang otomatis mengkonversi data JSON dengan struktur terdefinisi ke data class Kotlin dengan anotasi Kotlin Serialization?
- Apakah alat yang telah dibuat dapat memudahkan pengguna untuk mengintegrasikan data JSON terdefinisi ke dalam aplikasi mereka, dengan memanfaatkan data class Kotlin yang dihasilkan secara otomatis?

1.3 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

- Merancang dan membangun alat konversi dari JSON ke serializable data class kotlin dalam bentuk plugin yang support pada Android Studio dan IntelliJ IDEA
- Mengotomatisasi, menyederhanakan, serta mempercepat proses mapping data JSON ke *data class* kotlin dengan menggunakan plugin yang dibuat.

1.4 Manfaat

Manfaat dari tugas akhir ini diharapkan sebagai berikut:

- Membantu *developer* kotlin dalam membangun aplikasi yang lebih efisien, andal, dan mudah dipelihara.
- Meningkatkan produktivitas developer dengan mempercepat konversi data JSON, menghemat waktu developer, fokus pada aspek lain, dan meningkatkan efisiensi workflow.
- Menyediakan studi kasus, membuka peluang penelitian optimasi konversi JSON dan pengembangan tools, serta memperkaya pengetahuan tentang pengembangan aplikasi kotlin dengan data JSON.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Alat konversi JSON ke *data class* kotlin menghadirkan solusi fungsional dan bermanfaat bagi para *developer*. *Plugin* ini mampu mengonversi data JSON ke dalam kelas Kotlin secara otomatis, mendukung berbagai format data JSON yang umum dan kompleks. Pengguna dapat dengan mudah menentukan nama file *output*, konfigurasi kelas, dan opsi anotasi sesuai kebutuhan.

Dengan memanfaatkan alat ini, *developer* dapat menghemat waktu, meningkatkan efisiensi, dan fokus pada logika bisnis aplikasi. Hal ini pun berpotensi meningkatkan kualitas dan stabilitas aplikasi Kotlin, serta mendorong adopsi Kotlin Serialization sebagai framework anotasi pilihan untuk integrasi data JSON.

Merancang dan membangun alat konversi JSON ke data class Kotlin yang otomatis dan mudah digunakan dapat memberikan solusi berharga bagi *developer* Kotlin dalam mengoptimalkan integrasi data JSON. Dengan penerapan yang tepat, *plugin* ini dapat memberikan kontribusi positif bagi komunitas *developer* Kotlin dan meningkatkan kualitas pengembangan aplikasi Kotlin.

5.2 Saran

Beberapa saran untuk pengembangan *plugin* di masa depan meliputi:

- Dukungan anotasi kustom: Mengembangkan *plugin* untuk menangani lebih banyak jenis anotasi kustom sesuai kebutuhan pengguna.
- Peningkatan UI: Mempertimbangkan pengembangan antarmuka pengguna yang lebih intuitif dan ramah pengguna.
- Masih membutuhkan edit manual untuk tipe field dari json value yang tidak terisi.
- Dukungan konversi ke bahasa selain kotlin.

DAFTAR PUSTAKA

- [1] S. KEMP, "DIGITAL 2023: GLOBAL OVERVIEW REPORT," *datareportal.com*, 2023. <https://datareportal.com/reports/digital-2023-global-overview-report> (accessed May 08, 2024).
- [2] ECMA International, "The JSON Data Interchange Syntax," *Stand. ECMA-404*, vol. 2nd Editio, no. December 2017, p. 8, 2017, [Online]. Available: <http://www.ecma-international.org/publications/standards/Ecma-404.htm>.
- [3] M. Shafirov, "Kotlin on Android. Now official," *JetBrains Blog*, 2017. <https://blog.jetbrains.com/kotlin/2017/05/kotlin-on-android-now-official/> (accessed Jun. 25, 2024).
- [4] A. Breslav, "Kotlin 1.0 Released: Pragmatic Language for the JVM and Android," *JetBrains Blog*, 2016. <https://blog.jetbrains.com/kotlin/2016/02/kotlin-1-0-released-pragmatic-language-for-jvm-and-android/> (accessed Jun. 25, 2024).
- [5] Kotlin, "Data classes," *Kotlin Documentation*, 2024. <https://kotlinlang.org/docs/data-classes.html> (accessed May 08, 2024).
- [6] JetBrains, "Serialization," *Kotlin Documentation*, 2023. <https://kotlinlang.org/docs/serialization.html> (accessed Jun. 25, 2024).
- [7] Kotlin, "Kotlin Serialization Guide," *GitHub*, 2024. <https://github.com/Kotlin/kotlinx.serialization/blob/master/docs/serialization-guide.md> (accessed May 08, 2024).
- [8] S. Aigner, "kotlinx.serialization 1.0 released," *JetBrains Blog*, 2021. <https://blog.jetbrains.com/kotlin/2020/10/kotlinx-serialization-1-0-released/> (accessed Jun. 24, 2024).
- [9] Google, "gson: A Java serialization/deserialization library to convert Java Objects into JSON and back," *GitHub*, 2008. <https://github.com/google/gson> (accessed Jun. 25, 2024).
- [10] M. Afarideh, "Safely Navigating the Transition: From Gson to kotlinx.serialization," *Medium*, 2023. <https://proandroiddev.com/safely-navigating-the-transition-from-gson-to-kotlinx-serialization-4b9107e557e8> (accessed Jun. 25, 2024).

- [11] 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.
- [12] M. Voelter, *Generic Tools, Specific Languages*. Delft University of Technology, 2014.
- [13] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*, vol. 1. 2012.
- [14] N. Muhajir, "Converters," Apress eBooks, 2022, pp. 323–336.
- [15] I. Satya, R. Senthil, and R. P. Gsnv, "Data transformation tool," 2019.
- [16] R. J. Cruz, P. Barletta, N. Damonte, and J. A. Ron, "Reusable transformation mechanism to allow mappings between incompatible data types," 2019.
- [17] Android Developers, "Ringkasan Kotlin," *developer.android.com*, 2023. <https://developer.android.com/kotlin/overview?hl=id> (accessed May 08, 2024).
- [18] S. Samuel and S. Bocutiu, *Programming Kotlin*. Packt Publishing, 2017.
- [19] D. Jemerov and S. Isakova, *Kotlin in Action*. Manning, 2017.
- [20] JSON, "Introducing JSON," *Json.org*, 2024. <https://www.json.org/json-id.html> (accessed May 08, 2024).
- [21] H. Kolengsusu, "Rancang Bangun Plugin untuk Sistem Informasi Akademik dengan Ajax dan Web Services," Universitas Gadjah Mada, 2012.
- [22] F. A, "Apa Itu Plugin? Pengertian dan Fungsi Plugin di WordPress," *Hostinger Tutorial*, 2019. <https://www.hostinger.co.id/tutorial/apa-itu-plugin> (accessed May 08, 2024).
- [23] H. Koç, A. M. Erdoğan, Y. Barjakly, and S. Peker, "UML Diagrams in Software Engineering Research: A Systematic Literature Review," p. 13, 2021, doi: 10.3390/proceedings2021074013.
- [24] Unified Modeling Language, "What is UML," *Uml.org*, 2005. <https://www.uml.org/what-is-uml.htm> (accessed Jun. 24, 2024).
- [25] R. D. Gunawan, R. Napianto, R. I. Borman, and I. Hanifah, "Penerapan Pengembangan Sistem Extreme Programming Pada Aplikasi Pencarian

- Dokter Spesialis di Bandarlampung Berbasis Android,” *Format J. Ilm. Tek. Inform.*, vol. 8, no. 2, p. 148, 2020, doi: 10.22441/format.2019.v8.i2.008.
- [26] A. Fatoni and D. Dwi, “Rancang bangun sistem extreme programming sebagai metodologi pengembangan sistem,” *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 3, no. 1, 2016.
- [27] R. I. Borman, A. T. Priandika, and A. R. Edison, “Implementasi Metode Pengembangan Sistem Extreme Programming (XP) pada Aplikasi Investasi Peternakan,” *J. Sist. dan Teknol. Inf.*, vol. 8, no. 3, p. 272, 2020, doi: 10.26418/justin.v8i3.40273.
- [28] A. Restu Mukti, C. Mukmin, E. Randa Kasih, D. Palembang Jalan Jenderal Ahmad Yani No, S. I. Ulu, and S. Selatan, “Perancangan Smart Home Menggunakan Konsep Internet of Things (IOT) Berbasis Microcontroller,” *J. JUPITER*, vol. 14, no. 2, pp. 516–522, 2022.