

**RANCANG BANGUN APLIKASI GENERATOR CRUD
UNTUK FRAMEWORK LARAVEL BERBASIS WEB**

TUGAS AKHIR



**DISUSUN OLEH:
AKBEL VARINTO WALDA
NIM : 22106050048**

**STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA**

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

2026

LEMBAR PENGESAHAN



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-1260/Un.02/DST/PP.00.9/06/2026

Tugas Akhir dengan judul : Rancang Bangun Aplikasi Generator CRUD Untuk Framework Laravel Berbasis Web
yang dipersiapkan dan disusun oleh:

Nama : AKBEL VARINTO WALDA
Nomor Induk Mahasiswa : 22106050048
Telah diujikan pada : Kamis, 04 Juni 2026
Nilai ujian Tugas Akhir : A

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

TIM UJIAN TUGAS AKHIR



Ketua Sidang
Dr. Agung Fatwanto, S.Si., M.Kom.
SIGNED

Valid ID: 6a276eb2f0c2



Penguji I
Dr. Ir. Aulia Faqih Rifa'i, M.Kom.
SIGNED

Valid ID: 6a26faaac3a64



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

Valid ID: 6a266f6e5728



Yogyakarta, 04 Juni 2026
UIN Sunan Kalijaga
Dekan Fakultas Sains dan Teknologi
Prof. Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED

Valid ID: 6a278cae45e29

LEMBAR PERNYATAAN

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Akbel Varinto Walda
NIM : 22106050082
Program Studi : Informatika
Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya, bahwa skripsi saya yang berjudul **“RANCANG BANGUN APLIKASI GENERATOR CRUD UNTUK FRAMEWORK LARAVEL BERBASIS WEB”** merupakan penelitian saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan bukan plagiasi karya orang lain kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Sebagai salah satu syarat untuk memperoleh gelar Sarjana Program Studi Informatika pada Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Yogyakarta, 20 Mei 2026
Yang membuat pernyataan,



Akbel Varinto Walda
22106050048

LEMBAR PERSETUJUAN TUGAS AKHIR

Hal : Persetujuan Skripsi / Tugas Akhir
Lamp : -

Kepada
Yth.
Dekan Fakultas Sains dan
Teknologi UIN Sunan Kalijaga
DI Yogyakarta

Assalamu 'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka saya selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Akbel Varinto Walda
NIM : 22106050048
Judul Skripsi : Rancang Bangun Aplikasi Generator CRUD Untuk Framework Laravel Berbasis Web

Sudah dapat diajukan 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 skripsi / tugas akhir Saudara dapat segera dimunaqosyahkan. Atas perhatiannya saya ucapkan terima kasih.

Wassalamu 'alaikum wr. wb.

Yogyakarta, 19 Mei 2026
Dosen Pembimbing,



Dr. Agung Fatwanto, S.Si, M.Kom
NIP. 19770103 200501 1 003

INTISARI

Perkembangan teknologi web mendorong kebutuhan pengembangan aplikasi yang lebih cepat dan efisien. Dalam pengembangan aplikasi berbasis Laravel, proses pembuatan modul CRUD (Create, Read, Update, Delete) masih memerlukan waktu yang cukup lama karena pengembang harus membuat berbagai komponen seperti Model, Controller, Migration, dan View secara manual. Penelitian ini bertujuan untuk mengembangkan FastCRUD, yaitu tools generator CRUD berbasis web yang mampu menghasilkan module CRUD Laravel secara otomatis. Metode pengembangan sistem yang digunakan pada penelitian ini adalah Extreme Programming (XP) yang terdiri dari tahapan planning, design, coding, dan testing. Sistem dikembangkan menggunakan framework Laravel dengan pendekatan template-based code generation untuk menghasilkan struktur kode CRUD secara otomatis berdasarkan input pengguna. FastCRUD juga menyediakan fitur tambahan seperti timestamps, soft deletes, variasi tipe data field, serta export module dalam bentuk file ZIP. Pengujian sistem dilakukan menggunakan Black Box Testing untuk menguji fungsionalitas sistem serta performance testing untuk mengukur performa proses generate CRUD. Hasil pengujian menunjukkan bahwa seluruh fitur pada sistem dapat berjalan sesuai dengan fungsi yang diharapkan dan sistem mampu menghasilkan module CRUD secara cepat dan stabil. Berdasarkan hasil penelitian, FastCRUD mampu membantu mempercepat proses pengembangan aplikasi berbasis Laravel melalui otomatisasi pembuatan module CRUD secara lebih praktis, konsisten, dan efisien.

Kata Kunci: *Laravel, CRUD Generator, Extreme Programming, Code Generation, Aplikasi Web*

ABSTRACT

The rapid development of web technology has increased the need for faster and more efficient application development. In Laravel-based application development, the process of creating CRUD (Create, Read, Update, Delete) modules still requires considerable time because developers must manually create various components such as Models, Controllers, Migrations, and Views. This research aims to develop FastCRUD, a web-based CRUD generator tool capable of automatically generating Laravel CRUD modules. The system development method used in this research is Extreme Programming (XP), which consists of planning, design, coding, and testing stages. The system was developed using the Laravel framework with a template-based code generation approach to automatically generate CRUD code structures based on user input. FastCRUD also provides additional features such as timestamps, soft deletes, various field data types, and module export in ZIP file format. System testing was carried out using Black Box Testing to evaluate system functionality and performance testing to measure CRUD generation performance. The testing results show that all system features functioned properly according to their intended purposes, and the system was able to generate CRUD modules quickly and consistently. Based on the research results, FastCRUD is able to accelerate Laravel-based application development through a more practical, consistent, and efficient CRUD module generation automation process.

Keywords: *Laravel, CRUD Generator, Extreme Programming, Code Generation, Web Application*

MOTTO

”Sesungguhnya bersama kesulitan itu ada kemudahan, maka apabila kamu telah selesai (dari suatu urusan), tetaplah bekerja keras (untuk urusan yang lain)”

(Q.S Al-Insyirah: 5-6)

”Don’t wait too long, don’t just sit and watch”

~ Ir.H.Joko Widodo ~

“Semua manusia dimuka bumi ini bingung, nanti tidak bingung kalau sudah di surga”

~ Aldi Taher ~

HALAMAN PERSEMBAHAN

Dengan penuh rasa Syukur atas segala berkat dan karunia-Nya, saya Akbel Varinto Walda, menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT, yang senantiasa melimpahkan rahmat, kekuatan, kesehatan, dan petunjuk-Nya, sehingga saya dapat melalui setiap proses dengan ketabahan dan tumbuh menjadi pribadi yang lebih baik hingga saat ini.
2. Kedua orang tua saya tercinta, yang tanpa henti memberikan dukungan, doa yang tulus, dan kasih sayang yang tiada tara. Mereka adalah sumber motivasi terbesar dalam menyelesaikan laporan ini.
3. Kak Wednes dan Mas Fajar, yang selalu memberikan dukungan, perhatian, doa, serta menjadi keluarga yang selalu ada dan menemani saya selama proses penyusunan laporan ini hingga selesai.
4. Singgih Ramadhani, sebagai sahabat seperjuangan yang selalu menemani, mendengarkan keluh kesah, memberikan support, serta membantu saya selama proses penyusunan laporan ini.
5. Seluruh anggota grup Suro Squad, yang telah menjadi sahabat dan keluarga kecil yang selalu memberikan semangat, kebersamaan, serta warna selama perjalanan ini.
6. Teman-teman Angkatan 2022 Informatika UIN Sunan Kalijaga Yogyakarta, yang telah banyak membantu dan bekerja sama. Kebersamaan dan semangat yang terjalin sangat saya hargai.

Sebagai ungkapan syukur dan terima kasih atas selesainya Tugas Akhir ini

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Aplikasi Berbasis Web Generator Crud Untuk Framework Laravel” dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada program studi yang penulis tempuh. Dalam proses penyusunan skripsi ini, penulis memperoleh banyak bantuan, dukungan, serta bimbingan dari berbagai pihak sehingga penelitian dapat diselesaikan dengan baik.

Pada kesempatan ini, penulis ingin menyampaikan terima kasih kepada:

1. Kedua orang tua dan keluarga yang selalu memberikan doa, dukungan, serta motivasi kepada penulis.
2. Bapak/Ibu dosen pembimbing yang telah memberikan arahan, masukan, dan bimbingan selama proses penyusunan skripsi.
3. Seluruh dosen dan civitas akademika yang telah memberikan ilmu dan pengalaman selama masa perkuliahan.
4. Teman-teman dan semua pihak yang telah membantu serta mendukung penulis selama proses penelitian dan penyusunan skripsi.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan dan jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi perbaikan penelitian di masa mendatang. Semoga skripsi ini dapat memberikan manfaat bagi pembaca serta menjadi referensi dalam pengembangan aplikasi berbasis web, khususnya pada proses otomatisasi generate CRUD Laravel.

Yogyakarta, 12 Mei 2026



Akbel Varinto Walda
22106050048

DAFTAR ISI

LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN	iii
LEMBAR PERSETUJUAN TUGAS AKHIR.....	iv
INTISARI	v
ABSTRACT	vi
MOTTO	vii
HALAMAN PERSEMBAHAN	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah.....	6
1.4 Tujuan	6
1.5 Manfaat	7
BAB II KAJIAN PUSTAKA	8
2.1 Tinjauan Pustaka	8
2.2 Landasan Teori.....	10
2.2.1 Website dan Aplikasi Web.....	10
2.2.2 Framework dalam Pengembangan Web.....	11
2.2.3 Arsitektur Model-View-Controller (MVC).....	11
2.2.4 CRUD (Create, Read, Update, Delete)	12
2.2.5 Boilerplate Code dan Repetitive Tasks dalam Pengembangan Web	13
2.2.6 Code Generation.....	13
2.2.7 CRUD Generator.....	14
BAB III METODE PENGEMBANGAN SISTEM.....	15
3.1 Lokasi dan Waktu	15
3.2 Bahan dan Alat.....	15

3.3 Metode Pengembangan	16
BAB IV PERANCANGAN DAN IMPLEMENTASI	20
4.1 Perencanaan (Planning).....	20
4.1.1 Kebutuhan Fungsional	20
4.1.2 Kebutuhan Non Fungsional	21
4.2 Perancangan (Design)	22
4.2.1 Use Case Diagram.....	23
4.2.2 Flowchart Diagram.....	25
4.2.3 Class Diagram	29
4.2.4 Deployment Diagram	31
4.2.5 Wireframe	33
4.3 Implementasi (Coding).....	36
4.3.1 Iterasi Pertama.....	36
4.3.2 Iterasi Kedua	39
4.3.3 Iterasi Ketiga	41
4.4 Pengujian (Testing)	44
4.4.1 Metode Pengujian.....	45
4.4.2 Pengujian Fungsional (Black Box Testing)	47
4.4.3 Pengujian Non-Fungsional.....	51
4.4.4 Analisis Hasil Pengujian	52
4.5 Pembahasan.....	53
4.5.1 Interpretasi Hasil Pengujian	53
4.5.2 Efisiensi Pengembangan	54
4.5.3 Konsistensi dan Standarisasi Kode	56
4.5.4 Perbandingan dengan Pendekatan Lain	57
4.5.5 Kelebihan dan Keterbatasan Sistem.....	59
BAB V PENUTUP	61
5.1 Kesimpulan	61
5.2 Saran.....	62
DAFTAR PUSTAKA	63
LAMPIRAN.....	67

DAFTAR GAMBAR

Gambar 3. 1 Metode Extreme Programming (XP)	17
Gambar 4. 1 <i>Use Case Diagram</i>	24
Gambar 4. 2 <i>Flowchart Diagram</i>	27
Gambar 4. 3 <i>Class Diagram</i>	30
Gambar 4. 4 <i>Deployment Diagram</i>	32
Gambar 4. 5 <i>Section 1 : Model Configuration</i>	35
Gambar 4. 6 <i>Section 2 : Manage Fields</i>	35
Gambar 4. 7 <i>Section 3 : Preview Structure</i>	35
Gambar 4. 8 <i>Section 4 : Generate Module</i>	35
Gambar 4. 9 Antarmuka FastCRUD	38
Gambar 4. 10 Hasil <i>Generate</i> CRUD	39
Gambar 4. 11 Fitur <i>Soft Deletes & Timestamps</i>	40
Gambar 4. 12 Tipe Data <i>Field</i>	41
Gambar 4. 13 Implementasi Fitur <i>Search & Filter</i>	43
Gambar 4. 14 <i>Download ZIP</i>	44
Gambar 4. 15 Hasil <i>Generate</i> Modul	49
Gambar 4. 16 Tampilan Gambar Data (<i>Index</i>) Hasil <i>Generate</i>	50
Gambar 4. 17 <i>Form</i> Tambah Data	50

DAFTAR TABEL

Tabel 2. 1 Ringkasan Tinjauan Pustaka	9
Tabel 3. 1 Spesifikasi Komputer	15
Tabel 3. 2 Perangkat lunak yang digunakan dalam proyek TA	16
Tabel 4. 1 Kebutuhan Fungsional	21
Tabel 4. 2 Hasil Pengujian	47
Tabel 4. 3 Uji <i>Performance</i> Generate Module.....	51



STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi telah mendorong perubahan signifikan pada pola interaksi, aktivitas sosial, dan cara masyarakat mengakses informasi di era digital. Teknologi tidak lagi berfungsi sekadar sebagai alat bantu, tetapi telah menjadi elemen fundamental dalam berbagai aspek kehidupan, mulai dari pendidikan, pekerjaan, hingga layanan publik. Masyarakat kini tidak hanya berperan sebagai pengguna teknologi, melainkan juga sebagai aktor aktif yang turut membentuk arah transformasi digital melalui partisipasi pada berbagai platform dan ekosistem berbasis *web*. Perkembangan ini membuat teknologi digital menjadi bagian yang tidak terpisahkan dari aktivitas masyarakat modern dan mempengaruhi cara mereka berinteraksi maupun berkomunikasi [1]

Seiring meningkatnya kebutuhan akan sistem informasi berbasis *web*, struktur arsitektur yang digunakan dalam proses pengembangan aplikasi menjadi semakin penting untuk menjaga kualitas, skalabilitas, dan kerapian kode. *Website* modern tidak lagi hanya berfungsi sebagai media penyampaian informasi, tetapi juga menjadi platform transaksional dan interaktif yang menuntut efisiensi dalam pengelolaan data serta kemudahan pemeliharaan sistem. Hal ini terlihat dari pemanfaatan sistem informasi berbasis *web* dalam berbagai kebutuhan pengelolaan data, seperti pada sistem pendataan penduduk yang mampu meningkatkan efisiensi serta akurasi dalam proses pengolahan informasi[2]. Penggunaan sistem berbasis *web* memungkinkan data dapat dikelola secara lebih terstruktur dan mudah diakses. Dalam konteks ini, pendekatan *Model-View-Controller* (MVC) menjadi salah satu pola arsitektur yang paling banyak diadopsi karena mampu memisahkan logika bisnis, tampilan, dan pengelolaan data secara jelas. Seperti yang dijelaskan oleh [3], bahwa pemisahan tanggung jawab melalui arsitektur MVC memberikan

struktur kerja yang lebih teratur dan memudahkan pengembang dalam melakukan pembaruan tanpa memengaruhi komponen lain. Selain itu, penggunaan *framework* berbasis MVC seperti Laravel dan Vue.js semakin memperkuat efisiensi pengembangan karena menyediakan berbagai komponen siap pakai dan *reusable* yang dapat mempercepat proses pembuatan fitur baru. Hal ini sejalan dengan pernyataan [4], bahwa *framework* modern berperan penting dalam mempercepat pengembangan aplikasi sekaligus memudahkan kolaborasi tim melalui standar penulisan kode yang konsisten. Dengan demikian, pemanfaatan arsitektur MVC dan *framework* pendukung menjadi fondasi utama bagi pengembangan aplikasi *web* yang skalabel, mudah dirawat, dan mampu menjawab kebutuhan organisasi maupun pengguna secara berkelanjutan.

Popularitas Laravel sebagai *framework* PHP terus meningkat dalam beberapa tahun terakhir karena menawarkan beragam fitur yang mampu mempercepat dan menyederhanakan proses pengembangan aplikasi *web*. Seperti yang dijelaskan oleh [3] pada paragraf sebelumnya, kebutuhan akan struktur arsitektur yang jelas menjadikan *framework* modern semakin penting dalam menjaga kualitas perangkat lunak. Laravel menjawab kebutuhan tersebut melalui arsitektur MVC yang tertata, penyediaan komponen bawaan, serta dokumentasi yang lengkap. Studi terbaru menunjukkan bahwa Laravel termasuk salah satu *framework* PHP paling populer karena menyediakan proses otomatisasi yang meningkatkan efisiensi pengembangan, sehingga pengembang dapat mengurangi pekerjaan berulang yang biasanya ditemukan pada PHP murni [5]. Temuan lain menguatkan bahwa Laravel unggul dibandingkan PHP *Native*, terutama dari sisi efisiensi kode, kerapian struktur, serta dukungan *routing* yang lebih terorganisasi dan konsisten [6]. Selain itu, penelitian lain menyebut bahwa Laravel memiliki sistem modular, sintaks yang mudah dipahami, dukungan *database* yang luas, serta mekanisme *routing* sederhana yang mampu menghemat waktu secara signifikan dalam proses desain aplikasi [7]. Dengan keunggulan tersebut, Laravel menjadi *framework* yang tidak hanya

populer, tetapi juga mampu menjawab kebutuhan pengembangan aplikasi *web* modern yang menuntut kecepatan, skalabilitas, dan kemudahan pemeliharaan.

Meskipun *framework* seperti Laravel telah menyediakan struktur arsitektur yang lebih efisien, proses pengembangan aplikasi *web* masih sering menghadapi persoalan berupa penulisan kode *CRUD* (*Create, Read, Update, Delete*) yang bersifat repetitif. Aktivitas ini menghasilkan *boilerplate code* yang panjang dan berulang, sehingga memerlukan waktu pengerjaan yang tidak sedikit dan meningkatkan kemungkinan terjadinya *human error*, terutama pada proyek berskala besar atau yang melibatkan banyak entitas data [8]. Praktik pengulangan ini juga dianggap sebagai salah satu sumber inefisiensi dalam pengembangan perangkat lunak modern, karena pengembang harus membuat pola kode yang hampir sama pada setiap model data. Sejalan dengan meningkatnya kebutuhan akan percepatan pengembangan aplikasi seperti yang dibahas pada paragraf sebelumnya, berbagai studi menunjukkan bahwa proses otomatisasi khususnya dalam pembuatan *CRUD* dapat mengurangi kesalahan teknis, meningkatkan konsistensi struktur kode, serta mempercepat waktu pengembangan secara signifikan. Dengan demikian, persoalan *boilerplate code* bukan hanya tantangan umum dalam pengembangan *web*, tetapi juga menjadi salah satu alasan kuat perlunya alat bantu yang mampu mengotomatisasi pembuatan komponen dasar aplikasi, sehingga pengembang dapat lebih fokus pada logika bisnis dan fitur yang memiliki nilai inovatif lebih tinggi.

Selain mengatasi masalah repetisi dan *boilerplate* pada proses pembuatan *CRUD*, pemanfaatan *code generation* juga berperan penting dalam meningkatkan konsistensi dan akurasi hasil pengembangan. Pada konteks pengembangan aplikasi *web*, pendekatan *automated code generation* tradisional terbukti masih relevan karena memberikan alur kerja yang deterministik dan hasil yang lebih dapat diprediksi. Seperti dijelaskan dalam penelitian sebelumnya, metode *code generation* konvensional menawarkan tata cara yang stabil serta metrik yang jelas sehingga memudahkan pengembang dalam melakukan pengujian dan penyesuaian pada domain spesifik [9]. Hal ini berbeda dengan model berbasis AI yang sering menghasilkan variasi *output*

antar permintaan, sehingga kurang sesuai untuk kebutuhan generasi kode yang menuntut konsistensi tinggi. Penelitian lainnya juga menegaskan bahwa *code generator* berbasis template mampu menghasilkan keluaran yang konsisten, prediktif, dan dapat direplikasi, sementara *Large Language Model* (LLM) seperti ChatGPT cenderung memberikan hasil yang tidak stabil antar generasi kode [10]. Dengan demikian, *code generation* tradisional tetap menjadi pendekatan yang tepat dalam menghasilkan kerangka kode yang rapi, terstruktur, dan sesuai standar proyek.

Meskipun berbagai penelitian sebelumnya telah menjelaskan keunggulan *framework* Laravel, arsitektur MVC, serta manfaat penggunaan *code generation* dalam mengurangi repetisi dan meningkatkan efisiensi pengembangan, namun masih terdapat celah yang belum terjawab secara spesifik. Sebagian besar penelitian hanya berfokus pada implementasi aplikasi berbasis Laravel tanpa membahas kebutuhan pengembang terhadap alat yang mampu mengurangi *boilerplate code* secara otomatis dan terstruktur [11]. Selain itu, belum banyak penelitian yang secara khusus mengkaji bagaimana pendekatan *template-based code generation* dapat diterapkan untuk menghasilkan komponen utama seperti *model*, *controller*, dan *view* yang konsisten mengikuti standar Laravel modern. Di sisi lain, pemanfaatan *Artificial Intelligence* (AI) sebagai *code generator* menunjukkan potensi yang besar dalam membantu proses pengembangan, namun kualitas dan konsistensi kode yang dihasilkan masih memerlukan validasi serta mekanisme koreksi tambahan agar sesuai dengan kebutuhan pengembangan perangkat lunak [12]. Oleh karena itu, diperlukan sebuah penelitian yang menghadirkan alat *code generator* berbasis *template* yang mampu menghasilkan struktur kode Laravel secara otomatis, konsisten, dan sesuai standar pengembangan, sekaligus menjawab permasalahan repetisi CRUD yang masih menjadi tantangan bagi banyak pengembang.

Sebagai solusi atas permasalahan repetisi kode, *boilerplate* CRUD, serta keterbatasan pendekatan automasi tradisional maupun model AI generatif, penelitian ini mengusulkan pengembangan sebuah alat bernama FastCRUD.

FastCRUD dirancang sebagai *template-based code generator* yang terintegrasi dengan ekosistem Laravel, sehingga mampu menghasilkan struktur CRUD secara otomatis, konsisten, dan bebas *error*. Berbeda dengan pendekatan AI yang bersifat variatif dan tidak selalu dapat direplikasi, FastCRUD mengadopsi pola generasi deterministik yang memastikan hasil yang stabil untuk setiap modul yang dibuat. Pendekatan ini juga mengurangi waktu pengembangan secara signifikan, menekan risiko *human error*, serta menjaga struktur arsitektur Laravel agar tetap sesuai standar. Dengan demikian, FastCRUD menawarkan nilai kebaruan (*novelty*) berupa kombinasi kecepatan, konsistensi, dan kontrol penuh terhadap *template* kode yang dihasilkan, yang belum menjadi fokus pada penelitian-penelitian sebelumnya.

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang, maka permasalahan utama yang diidentifikasi adalah sebagai berikut:

1. Bagaimana menyediakan sebuah alat otomasi CRUD yang dapat mengurangi repetisi penulisan *boilerplate* pada Laravel, sehingga proses pengembangan menjadi lebih cepat, efisien, dan minim *human error*?
2. Bagaimana merancang generator kode yang konsisten, deterministik, dan mudah direplikasi, sehingga dapat menjadi solusi atas variasi hasil yang sering muncul pada pendekatan otomatisasi berbasis AI?
3. Bagaimana memastikan bahwa proses otomasi CRUD yang dihasilkan tetap mengikuti standar arsitektur *Model-View-Controller* (MVC) Laravel, sehingga kompatibel dengan praktik pengembangan modern dan mudah dipelihara?
4. Bagaimana membangun alat yang dapat digunakan oleh mahasiswa maupun pengembang pemula secara intuitif, tanpa bergantung pada *Command Line Interface* (CLI), serta memungkinkan modul hasil generasi digunakan lintas-proyek secara fleksibel?

1.3 Batasan Masalah

Penelitian ini difokuskan pada pengembangan FastCRUD sebagai generator modul CRUD berbasis Laravel 12 dan PHP 8.2, sehingga seluruh pengujian, implementasi, dan evaluasi berada pada ekosistem teknologi tersebut. Sistem hanya menangani proses otomatisasi CRUD untuk struktur tabel tunggal, tanpa mencakup pengolahan relasi antar tabel atau skema *database* yang kompleks. Dukungan tipe data bersifat umum sesuai kebutuhan dasar pengembangan CRUD dan tidak membahas pengelolaan tipe data khusus ataupun *custom data casting*. Antarmuka modul yang dihasilkan menggunakan Bootstrap 5 sebagai kerangka tampilan standar yang diterapkan secara seragam pada seluruh keluaran sistem. Distribusi modul dilakukan dalam bentuk paket ZIP yang berisi komponen MVC siap pakai, namun tidak mencakup fitur keamanan lanjutan, otorisasi berbasis peran, ataupun pengaturan akses modul. Dengan pembatasan tersebut, penelitian ini bertujuan menilai efektivitas otomasi pembuatan CRUD tanpa memasukkan aspek-aspek kompleks yang berada di luar ruang lingkup perancangan FastCRUD.

1.4 Tujuan

Tujuan utama dari penelitian ini adalah merancang dan membangun FastCRUD, sebuah sistem otomasi generator CRUD berbasis *web* yang mampu menghasilkan komponen *Model*, *Controller*, *View*, dan *Migration* secara terstandarisasi pada *framework* Laravel. Selain itu, penelitian ini bertujuan mengimplementasikan fitur *Smart Pluralization Engine* untuk memastikan konsistensi penamaan entitas sesuai konvensi Laravel, serta menyediakan mekanisme *ZIP Module Exporter* agar modul yang dihasilkan dapat didistribusikan dengan mudah dan siap dipasang pada proyek lain. Sistem FastCRUD juga dirancang untuk meningkatkan efisiensi proses pengembangan dengan meminimalkan penulisan kode *boilerplate* yang bersifat repetitif. Pada tahap akhir, penelitian ini melakukan analisis komparatif terkait efektivitas waktu antara penggunaan FastCRUD dan metode penulisan kode CRUD secara

manual untuk menilai kontribusi sistem terhadap percepatan siklus pengembangan perangkat lunak.

1.5 Manfaat

Hasil dari penelitian ini diharapkan dapat memberikan sejumlah manfaat bagi berbagai pihak yang terlibat dalam kegiatan pengembangan perangkat lunak. Bagi para pengembang (*developer*), FastCRUD membantu menghilangkan pekerjaan repetitif pada fase inisiasi proyek sehingga waktu pengerjaan dapat lebih difokuskan pada perancangan logika bisnis yang bersifat kompleks. Bagi mahasiswa informatika, sistem ini dapat berfungsi sebagai media pembelajaran visual yang memudahkan pemahaman mengenai hubungan antar komponen dalam arsitektur *Model–View–Controller* (MVC) secara lebih konkret. Selain itu, bagi *freelance developer*, FastCRUD berpotensi meningkatkan produktivitas pengerjaan proyek dengan memangkas durasi pembuatan modul rutin sekaligus memastikan konsistensi dan standarisasi kode yang dihasilkan. Dari sisi instansi atau organisasi, penelitian ini memberikan kontribusi berupa ketersediaan alat bantu untuk menegakkan standar penulisan kode di lingkungan tim pengembangan, sehingga proses pemeliharaan dan pengelolaan aplikasi dapat dilakukan secara lebih efisien dan terstruktur.

BAB V

KESIMPULAN DAN PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem FastCRUD berhasil dikembangkan sebagai alat otomatisasi pembuatan modul CRUD berbasis Laravel. Sistem ini mampu menghasilkan komponen utama seperti *Model*, *Controller*, *View*, dan *Migration* secara terstruktur dan sesuai dengan standar arsitektur *Model-View-Controller* (MVC) pada Laravel.

Selain itu, FastCRUD terbukti dapat mengurangi proses penulisan kode yang bersifat repetitif (*boilerplate*), sehingga meningkatkan efisiensi waktu dalam pengembangan aplikasi *web*. Pengujian yang dilakukan menunjukkan bahwa modul yang dihasilkan dapat diintegrasikan dengan baik ke dalam proyek Laravel dan berjalan sesuai dengan fungsi CRUD yang diharapkan.

Dari sisi konsistensi, pendekatan *template-based code generation* yang digunakan pada FastCRUD mampu menghasilkan kode yang lebih stabil dan terprediksi dibandingkan pendekatan generatif berbasis AI, sehingga lebih sesuai untuk kebutuhan pengembangan perangkat lunak yang terstruktur. Dengan demikian, FastCRUD dapat menjadi solusi yang membantu pengembang, khususnya mahasiswa dan pemula, dalam mempercepat proses pengembangan aplikasi berbasis web secara lebih efisien dan sistematis.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan sistem FastCRUD di masa mendatang.:

1. Sistem FastCRUD dapat dikembangkan lebih lanjut dengan menambahkan dukungan relasi antar tabel, seperti *one-to-many* dan *many-to-many*, sehingga dapat digunakan untuk pengembangan aplikasi dengan struktur *database* yang lebih kompleks.
2. Perlu ditambahkan fitur autentikasi dan otorisasi pengguna, seperti *role-based access control* (RBAC), agar modul yang dihasilkan dapat langsung digunakan pada sistem yang membutuhkan pengelolaan hak akses.
3. Pengembangan selanjutnya dapat mencakup fleksibilitas dalam kustomisasi *template*, sehingga pengguna dapat menyesuaikan struktur kode dan tampilan sesuai dengan kebutuhan proyek masing-masing.
4. Sistem dapat ditingkatkan dengan menambahkan dukungan validasi data yang lebih kompleks serta integrasi dengan fitur keamanan, guna meningkatkan kualitas dan keandalan modul yang dihasilkan.
5. Penelitian selanjutnya dapat melakukan pengujian tambahan seperti *usability testing* atau *performance testing* untuk mengevaluasi pengalaman pengguna serta kinerja sistem secara lebih komprehensif.
6. FastCRUD juga berpotensi dikembangkan ke dalam bentuk *plugin* atau *package* yang terintegrasi langsung dengan ekosistem Laravel, sehingga dapat digunakan secara lebih luas oleh komunitas pengembang.

DAFTAR PUSTAKA

- [1] F. Sinlae, F. Sabila Rosyad, F. Nurhidayat, dan W. Jannah, “Evolusi Teknologi Web dan Dampaknya Terhadap Masyarakat Digital,” *Jurnal Ilmu Multidisiplin*, vol. 3, no. 2, hlm. 146–254, 2024, [Daring]. Tersedia pada: <https://creativecommons.org/licenses/by/4.0/>
- [2] S. Tahun dan T. Pustaka, “Sistem Informasi dan Implementasi berbasis Web untuk Peningkatan Pelayan Showroom Mobil dengan Metode,” vol. 3, no. 4, hlm. 1057–1062, 2025.
- [3] A. Alip, S. Kosasi, I. D. A. E. Yuliani, G. Syarifudin, dan D. David, “Implementasi Arsitektur Model View Controller Pada Website Toko Online,” *Jurnal Bumigora Information Technology (BITE)*, vol. 3, no. 2, hlm. 135–150, 2022, doi: 10.30812/bite.v3i2.1566.
- [4] L. Rahmawati dan S. Sumarsono, “Desain Pengembangan Website dengan Arsitektur Model View Controller pada Framework Laravel,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 6, no. 4, hlm. 785–790, 2024, doi: 10.47233/jteksis.v6i4.1497.
- [5] P. R. Chavan dan S. Pawar, “Ijresm_V4_I10_5,” vol. 4, no. 10, hlm. 27–29, 2021.
- [6] Y. Ariyanto, M. Farhan, F. Rachmad, dan D. Puspitasari, “Laravel framework and native PHP: Comparison in the creation of rest API,” *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 14, no. 2, hlm. 66–73, 2024.
- [7] R. yahya Praja putra dan A. Rahman, “Rancang Bangun Web Desan Dengan Framework Laravel,” *Jurnal Inovasi Komputer (INOKOM)*, vol. 1, no. 2, hlm. 87–96, 2025, doi: 10.71200/inokom.v1i2.73.
- [8] A. W. Anuar, N. Kama, A. Azmi, H. M. Rusli, dan Y. Yahya, “Re-CRUD Code Automation Framework Evaluation using DESMET Feature Analysis,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, hlm. 437–452, 2022, doi: 10.14569/IJACSA.2022.0130552.
- [9] A. Odeh, N. Odeh, dan A. S. Mohammed, “A Comparative Review of AI Techniques for Automated Code Generation in Software Development: Advancements, Challenges, and Future Directions,” *TEM Journal*, vol. 13, no. 1, hlm. 726–739, 2024, doi: 10.18421/TEM131-76.

- [10] A. Bochenek, "Can ChatGPT Replace a Template-based Code Generator?," *Communication Papers of the 18th Conference on Computer Science and Intelligence Systems*, vol. 37, hlm. 43–50, 2023, doi: 10.15439/2023f5691.
- [11] J. G. Villalba, B. Uyanık, dan A. Sayar, "Citation: Uyanık B.; Sayar A. Developing Web-Based Process Management with Automatic Code Generation," *Appl. Sci*, vol. 13, hlm. 11737, 2023, [Daring]. Tersedia pada: <https://doi.org/10.3390/app>
- [12] K. T. Le dan A. Andrzejak, "Rethinking AI code generation: a one-shot correction approach based on user feedback," *Automated Software Engineering*, vol. 31, no. 2, hlm. 1–42, 2024, doi: 10.1007/s10515-024-00451-y.
- [13] A. Syawalia, R. Fenanda, R. E. Tija, L. Komariah, R. Andrew, dan D. M. Mukaromah, "Efisiensi dan Produktivitas Pengembangan Aplikasi Web sebagai Nilai Ekonomi : Studi Komparatif Framework Laravel dan PHP Native," vol. 1, hlm. 175–181, 2026.
- [14] D. K. Pakpahan *dkk.*, "Alasan Penggunaan Arsitektur MVC dalam Implementasi CRUD," *Indonesian Journal of Computer Science and Engineering (IJCSE)*, vol. 2, no. 1, hlm. 45–49, 2025.
- [15] F. Marwati dan A. Zein, "Pengembangan Domain Specific Language (DSL) untuk Generator Aplikasi Basisdata dengan Kode Aplikasi Secara Otomatis Berbasis WEB," *Sainstech: Jurnal Penelitian Dan Pengkajian Sains Dan Teknologi*, vol. 35, no. 1, hlm. 10–17, 2025, doi: 10.37277/stch.v35i1.2306.
- [16] T. Ardiansah, "Perancangan Sistem Persediaan Menggunakan Metode Extreme Programming," *Jurnal Ilmiah Informatika dan Ilmu Komputer (JIMA-ILKOM)*, vol. 1, no. 1, hlm. 1–6, 2022, doi: 10.58602/jima-ilkom.v1i1.1.
- [17] D. Kustiawan, W. N. Cholifah, R. Destriana, dan N. Heriyani, "Rancang Bangun Sistem Informasi Akuntansi Pengelolaan Koperasi Menggunakan Metode Extreme Programming," *Jurnal Teknologi dan Informasi*, vol. 12, no. 1, hlm. 78–92, 2022, doi: 10.34010/jati.v12i1.6756.
- [18] R. Baxter, *Software engineering is software engineering*. 2006. doi: 10.1049/ic:20040411.
- [19] M. M. I. Molla, J. Ahmad, dan W. M. N. Wan Kadir, "A Comparison of Transforming the User Stories and Functional Requirements into UML Use Case Diagram," *International Journal of Innovative Computing*, vol. 14, no. 1, hlm. 29–36, 2024, doi: 10.11113/ijic.v14n1.463.

- [20] Y. Andriyani, I. D. Id, E. Mahdiyah, dan A. Aminuddin, “Use Case Realization in Software Reverse Engineering,” *Ingenierie des Systemes d’Information*, vol. 27, no. 2, hlm. 335–341, 2022, doi: 10.18280/isi.270218.
- [21] A. Ghritlahare, “The Role of Flowcharts in Problem Solving and Process Visualization,” Okt 2025, doi: 10.5281/zenodo.15358152.
- [22] S. Xinogalos, *Using Flowchart-based Programming Environments for Simplifying Programming and Software Engineering Processes*. 2013. doi: 10.1109/EduCon.2013.6530276.
- [23] M. F. Santoso, “Implementation Of UI/UX Concepts And Techniques In Web Layout Design With Figma,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 6, no. 2, hlm. 279–285, 2024, doi: 10.47233/jteksis.v6i2.1223.
- [24] A. Prastiwi dan B. Prasetyo, “Design of User Interface and User Experience Website-Based Mental Health Applications as a Means of Counseling Universitas Negeri Semarang Students Using Design Thinking Methods and System Usability Scale,” *Journal of Advances in Information Systems and Technology*, vol. 6, no. 2, hlm. 308–325, 2024.
- [25] D. Saro, “Perancangan Wireframe dan Mockup Sistem E-Commerce untuk PT.XYZ,” *Engineering and Technology International Journal*, vol. 7, no. 01, hlm. 28–35, 2025, doi: 10.55642/eatij.v7i01.966.
- [26] . S., W. Hadikristanto, dan N. T. Kurniadi, “Implementasi Pengembangan Aplikasi Sistem Manajemen Aset Berbasis Web Menggunakan Metode Waterfall Untuk Mengoptimalkan Penggunaan Aset Pada PT. Hutama Karya (Persero),” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 4, hlm. 401–408, 2023, doi: 10.47233/jteksis.v5i4.948.
- [27] Athaya Rima Hariyanto, “Implementasi Dan Analisis Sistem Informasi Manajemen Data Mahasiswa Berbasis Web,” vol. 1, no. 2, hlm. 46–53, 2025.
- [28] Alivia Risma Dewi Safana dan Dendy Kurniawan, “Implementasi Pemrograman Web Untuk Meningkatkan Efisiensi Layanan Informasi,” *Jurnal Riset Teknik Komputer*, vol. 2, no. 3, hlm. 117–123, 2025, doi: 10.69714/k7bh3m03.
- [29] Uminingsih, M. N. Ichsanudin, M. Yusuf, dan Suraya, “Perpustakaan Dengan Metode Black Box Testing Bagi Pemula,” *STORAGE-Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 1, no. 2, hlm. 1–8, 2022.
- [30] F. E. Susilawati, S. Suparman, dan A. M. G. Patalo, “PENGUJIAN BLACK BOX APLIKASI PENJUALAN PUPUK BERSUBSIDI MENGGUNAKAN TEKNIK EQUIVALENCE PARTITIONING,”

D'computare: Jurnal Ilmiah Teknologi Informasi dan Ilmu Komputer, vol. 12, no. 2, hlm. 30–34, Jul 2022, doi: 10.30605/dcomputare.v12i2.48.

- [31] E. Syriani, L. Luhunu, dan H. Sahraoui, “Systematic mapping study of template-based code generation,” *Comput. Lang. Syst. Struct.*, vol. 52, hlm. 43–62, 2018, doi: 10.1016/j.cl.2017.11.003.
- [32] M. A. Possatto dan D. Lucrédio, “Automatically propagating changes from reference implementations to code generation templates,” *Inf. Softw. Technol.*, vol. 67, hlm. 65–78, 2015, doi: 10.1016/j.infsof.2015.06.009.
- [33] H. Klare, M. E. Kramer, M. Langhammer, D. Werle, E. Burger, dan R. Reussner, “Enabling consistency in view-based system development — The VITRUVIUS approach,” *Journal of Systems and Software*, vol. 171, 2021, doi: 10.1016/j.jss.2020.110815.
- [34] A. Clark, D. Igbokwe, S. Ross, dan M. F. Zibran, “A Quantitative Analysis of Quality and Consistency in AI-generated Code,” dalam *2024 7th International Conference on Software and System Engineering (ICoSSE)*, 2024, hlm. 37–41. doi: 10.1109/ICoSSE62619.2024.00014.
- [35] B. Gülmez, “Code generation with large language models: a survey from neural program synthesis to autonomous software development,” *Applied Intelligence*, vol. 56, no. 6, hlm. 1–18, 2026, doi: 10.1007/s10489-026-07230-0.
- [36] E. Syriani, L. Luhunu, dan H. Sahraoui, “Systematic mapping study of template-based code generation,” *Comput. Lang. Syst. Struct.*, vol. 52, hlm. 43–62, 2018, doi: 10.1016/j.cl.2017.11.003.