TUGAS AKHIR

RANCANG BANGUN *PLUGIN* FIGMA TOKEN2CODE UNTUK OTOMATISASI PROSES DESAIN KE KODE



DISUSUN OLEH:

AHSAN NAFIS'ALIMI

NIM. 22106050060

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

PROGRAM STUDI INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA

YOGYAKARTA

2026

LEMBAR PERNYATAAN KEASLIAN

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Ahsan Nafis'alimi
NIM : 22106050060
Program Studi : Informatika
Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya, bahwa skripsi saya yang berjudul **"Rancang Bangun *Plugin* Figma Token2Code untuk Otomatisasi Proses Desain ke Kode"** 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.

Yogyakarta, 30 September 2026
Yang membuat pernyataan,



Ahsan Nafis'alimi
22106050060

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

LEMBAR PERSETUJUAN

LEMBAR PERSETUJUAN TUGAS AKHIR

Hal : Persetujuan 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	:	Ahsan Nafis'alimi
NIM	:	22106050060
Judul Skripsi	:	Rancang Bangun Figma *Plugin* Design Token to Code Generator dengan Output Tailwind CSS dan CSS Variables untuk Otomatisasi Proses Desain ke Kode

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 mengharap agar tugas akhir Saudara dapat segera dimunaqosyahkan. Atas perhatiannya saya ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 24 Agustus 2026
Pembimbing,

Nurochman, S.Kom., M.Kom.
NIP. 19801223 200901 1 007

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

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-2385/Un.02/DST/PP.00.9/09/2026

Tugas Akhir dengan judul : RANCANG BANGUN PLUGIN FIGMA TOKEN2CODE UNTUK OTOMATISASI PROSES DESAIN KE KODE

yang dipersiapkan dan disusun oleh:

Nama : AHSAN NAFIS'ALIMI
Nomor Induk Mahasiswa : 22106050060
Telah diujikan pada : Rabu, 26 Agustus 2026
Nilai ujian Tugas Akhir : A/B

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

TIM UJIAN TUGAS AKHIR

Ketua Sidang

Nurochman, S.Kom., M.Kom
SIGNED

Valid ID: 6ab614468308e

Penguji I

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

Valid ID: 6aa8d8f6286b4

Penguji II

Ir. Maria Ulfah Siregar, S.Kom., MIT., Ph.D.
SIGNED

Valid ID: 6ab61038b4781

Yogyakarta, 26 Agustus 2026
UIN Sunan Kalijaga
Dekan Fakultas Sains dan Teknologi

Prof. Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED

Valid ID: 6ab9e59646127

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

INTISARI

Proses penyerahan desain dari desainer kepada pengembang yang masih dilakukan secara manual sering kali menimbulkan kesenjangan komunikasi. Kondisi ini dapat memicu technical debt serta menyebabkan ketidakkonsistenan visual pada basis kode. Meskipun Figma telah menyediakan fitur Variables untuk menerapkan konsep design tokens, proses konversi nilai variabel ke framework seperti Tailwind CSS masih perlu dilakukan secara manual dan berpotensi menimbulkan kesalahan. Untuk mengatasi permasalahan tersebut, tugas akhir ini mengembangkan plugin Figma bernama Token2Code. Plugin ini dirancang untuk mengotomatisasi ekstraksi Figma Variables bertipe Warna dan Angka menjadi CSS Variables serta konfigurasi Tailwind CSS versi 3. Metode pengembangan yang digunakan adalah Waterfall melalui tahapan analisis kebutuhan, desain, pengkodean, dan pengujian. Pengujian fungsionalitas dilakukan menggunakan Black Box Testing, sedangkan tingkat kebergunaan sistem diukur melalui System Usability Scale (SUS). Seluruh skenario Black Box Testing berjalan sesuai rancangan, baik pada kondisi ideal maupun anomali. Selanjutnya, pengujian SUS yang melibatkan 16 responden menghasilkan skor rata-rata 78,3. Skor tersebut berada dalam kategori Acceptable dengan predikat Good. Hasil ini menunjukkan bahwa Token2Code mampu mengotomatisasi proses konversi desain ke kode secara lebih praktis serta dapat diterima dengan baik oleh pengguna tanpa memerlukan toolchain eksternal tambahan.

Kata Kunci: *Design Token*, Figma Variables, Figma *plugin*, Tailwind CSS, *System Usability Scale* (SUS).

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

ABSTRACT

The process of handing over designs from designers to developers, which is still done manually, often leads to communication gaps. This situation can trigger technical debt and cause visual inconsistencies in the codebase. Although Figma has provided Variables features to implement the concept of design tokens, the process of converting variable values to frameworks like Tailwind CSS still needs to be done manually and has the potential to introduce errors. To address this issue, this final project develops a Figma plugin called Token2Code. This plugin is designed to automate the extraction of Figma Variables of type Color and Float into CSS Variables and Tailwind CSS version 3 configurations. The development method used is Waterfall, through stages of requirements analysis, design, coding, and testing. Functionality testing is conducted using Black Box Testing, while the system's usability level is measured through the System Usability Scale (SUS). All Black Box Testing scenarios ran as planned, both under ideal conditions and anomalies. Furthermore, the SUS testing involving 16 respondents resulted in an average score of 78.3. This score falls within the Acceptable category with a Good rating. This result indicates that Token2Code is capable of automating the design-to-code conversion process in a more practical manner and is well-received by users without requiring additional external toolchains.

***Keywords**: Design Token, Figma Variables, Figma plugin, Tailwind CSS, System Usability Scale (SUS).*

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

MOTTO

“Send him 2-3 years Dagestan and forget.”

~ Islam Makhachev

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

KATA PENGANTAR

Alhamdulillahi Rabbil 'Alamin, penulis mengucapkan syukur kepada Allah سُبْحَانَهُ وَ تَعَالَى atas rahmat dan karunia-Nya, yang memungkinkan penulis untuk menyelesaikan tugas akhir ini yang berjudul "Rancang Bangun *Plugin* Figma Token2Code untuk Otomatisasi Proses Desain ke Kode." Sholawat dan salam senantiasa tercurah kepada junjungan kita, Rasulullah ﷺ, yang telah membawa umat manusia ke dalam zaman yang penuh cahaya, dan semoga kita semua diberi syafaatnya di akhir zaman.

Penyelesaian tugas akhir ini memerlukan usaha dan kerja keras dalam setiap tahapannya. Namun, proses penyelesaian ini tidak terlepas dari bantuan dan dukungan berbagai pihak, baik yang terlibat langsung maupun tidak langsung. Pada kesempatan ini, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Prof. Noorhaidi Hasan S.Ag., M.A., M.Phil., Ph.D. 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. Bapak Dr. Muhammad Mustakim, S.T., M.T. selaku Ketua Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Bapak Ir. Muhammad Didik Rohmad Wahyudi, S.T., MT. selaku dosen penasihat akademik penulis selama menempuh perkuliahan di UIN Sunan Kalijaga
5. Bapak Nurochman, S.Kom., M.Kom. selaku dosen pembimbing tugas akhir yang telah memberikan arahan, bimbingan, serta masukan berharga selama proses penyusunan tugas akhir ini.
6. Seluruh dosen Program Studi Informatika UIN Sunan Kalijaga yang telah memberikan ilmu dan pengalaman berharga selama masa perkuliahan.
7. Kedua orang tua penulis, Bapak Harsono dan Ibu Sukristin, atas segala pengorbanan, doa, nasihat, pendidikan, dan kasih sayang yang tak akan pernah dapat penulis balas sepenuhnya.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

8. Kedua adik penulis, Brain Afrizal Z. dan Brial Rizal A. yang lucu, menggemaskan, dan menjadi sumber semangat penulis dalam menyelesaikan tugas akhir.
9. Seluruh Mahasiswa Informatika terutama Angkatan 2022 yang telah bersama-sama berjuang dan mendukung pada saat perkuliahan.
10. Sahabat-sahabat penulis, grup PPT, Messi, Dewan Keamanan PBB dan sahabat penulis lainnya yang selalu menemani dan memberikan semangat selama masa studi.
11. Ustadz Aldus, Ustadz Asmara, dan Mas Ihda yang telah banyak memberikan nasihat spiritual dan sosial secara langsung sehingga penulis dapat tercerahkan selama masa penyusunan tugas akhir dan untuk kehidupan seterusnya.

Penulis menyadari bahwa tugas akhir ini masih ada kekurangan secara penulisan maupun materi. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi perbaikan di masa mendatang. Semoga tugas akhir ini dapat memberikan manfaat serta menjadi kontribusi dalam pengembangan ilmu pengetahuan, khususnya pada bidang rekayasa perangkat lunak.

Yogyakarta, 22 Agustus 2026
Penulis,

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Ahsan Nafis'alimi
NIM. 22106050060

DAFTAR ISI

LEMBAR PERNYATAAN KEASLIAN ii
LEMBAR PERSETUJUAN iv
INTISARI v
ABSTRACT vi
MOTTO vii
KATA PENGANTAR viii
DAFTAR ISI x
DAFTAR TABEL xii
DAFTAR GAMBAR xiii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 4
1.3 Batasan Masalah 4
1.4 Tujuan 5
1.5 Manfaat 5
BAB II TINJAUAN PUSTAKA 7
2.1 *Plugin* 7
2.2 Ekosistem Figma dan *Plugin* API 7
2.3 Struktur Data API Figma dan *Abstract Syntax Tree* (AST) 8
2.4 *Design Token* 9
2.5 Figma Variables 10
2.6 CSS Variables 11
2.7 Tailwind CSS dan Mesin Kompilasi JIT 12
2.8 Eksplorasi *Plugin* Serupa 13
2.9 Metode Pengembangan 16
2.10 Metode Pengujian 18
2.10.1 *Black Box Testing* 18
2.10.2 *System Usability Scale* 18

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

BAB III METODE PENGEMBANGAN SISTEM 22
3.1 Metode Pengembangan 22
3.2 Tahapan Metode *Waterfall* 22
BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM 26
4.1 Analisis Kebutuhan 26
4.1.1 Analisis Sistem yang Sudah Ada 26
4.1.2 Kebutuhan Fungsional 26
4.1.3 Kebutuhan Non-fungsional 27
4.2 Desain 28
4.2.1 Use Case Diagram 28
4.2.2 Activity Diagram 30
4.2.3 Perancangan Antarmuka Pengguna 31
4.3 Pengkodean 33
4.3.1 Pengkodean Antarmuka Menu Ekspor CSS 34
4.3.2 Pengkodean Antarmuka Menu Ekspor Tailwind 35
4.3.3 Pengkodean Ekstraksi API Figma 35
4.3.4 Pengkodean Logika Konversi (Mapping Logic) 38
4.3.5 Pengkodean Hasil Ekspor Kode Konfigurasi 40
4.4 Pengujian Sistem 44
4.4.1 *Black Box Testing* 44
4.4.2 Pengujian *Usability* dengan Metode Kuesioner SUS 47
BAB V KESIMPULAN DAN SARAN 54
5.1 Kesimpulan 54
5.2 Saran 55
DAFTAR PUSTAKA 56
LAMPIRAN 1

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

DAFTAR TABEL

Tabel 2.1 Eksplorasi *Plugin* Serupa 14
Tabel 2.2 Kriteria Penilaian Skor SUS..... 21
Tabel 4.1 Kebutuhan Fungsional Sistem..... 27
Tabel 4.2 Kebutuhan Non-fungsional Sistem 27
Tabel 4.3 Hasil Pengujian Black Box - Kasus dan Hasil Uji Positif..... 45
Tabel 4.4 Hasil Pengujian *Black Box* - Kasus dan Hasil Uji Negatif..... 47
Tabel 4.5 Daftar Pernyataan *System Usability Scale* 48
Tabel 4.6 Data Awal hasil Pengujian SUS 49
Tabel 4.7 Hasil Perhitungan Skor *System Usability Scale* 51

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

DAFTAR GAMBAR

Gambar 3.1 Tahapan Metode *Waterfall* [48] 22
Gambar 4.1 *Use Case Diagram* Token2Code 29
Gambar 4.2 *Activity Diagram* Ekstraksi dan Konversi Token Desain 30
Gambar 4.3 Desain Antarmuka Menu Ekspor CSS Variables 32
Gambar 4.4 Desain Antarmuka Menu Ekspor Tailwind CSS 33
Gambar 4.5 Tampilan Antarmuka Menu Ekspor CSS 34
Gambar 4.6 Tampilan Antarmuka Menu Ekspor Tailwind 35
Gambar 4.7 Pengkodean Ekstraksi dan Pemilahan Tipe Data 37
Gambar 4.8 Pengkodean Logika Konversi 38
Gambar 4.9 Pengkodean Konversi CSS Variables 39
Gambar 4.10 Pengkodean Konversi Tailwind CSS 40
Gambar 4.11 Pengkodean Pengiriman Data dari code.ts ke Antarmuka Pengguna 41
Gambar 4.12 Pengkodean Logika Mekanisme Download File pada ui.html 41
Gambar 4.13 Hasil Unduhan File CSS Variables 42
Gambar 4.14 Hasil Unduhan File Tailwind CSS 43
Gambar 4.15 Interpretasi Skor SUS [51] 52

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kebutuhan akan desain antarmuka yang mudah digunakan dan perangkat lunak yang bisa diandalkan membuat desainer dan pengembang harus bekerja sama secara tepat. Namun, di dunia industri perangkat lunak, kerja sama ini sering terhambat oleh masalah komunikasi teknis. Laporan State of Designer-Developer Collaboration 2024 dari Telerik, hanya 10% desainer yang merasa kerja sama dengan pengembang berjalan lancar. Survei itu juga menunjukkan bahwa 51% perusahaan kehilangan banyak waktu kerja spesialis karena proses penyerahan desain yang tidak jelas. Masalah ini berdampak langsung pada bisnis. Sebanyak 28% perusahaan mengalami keterlambatan peluncuran produk, 17% menghadapi konflik dalam tim, dan 14% perusahaan menghasilkan produk dengan kualitas rendah [1], [2].

Laporan tahun 2022 dari Consortium for Information & Software Quality (CISQ) menunjukkan bahwa Amerika Serikat mengalami kerugian ekonomi mencapai $2,41 triliun akibat kualitas perangkat lunak yang buruk. Dari jumlah kerugian tersebut, $1,52 triliun di antaranya disebabkan oleh penumpukan technical debt. Kondisi ini menjadi salah satu hambatan utama ketika tim perlu melakukan perubahan pada basis kode yang telah ada [3]. Selain itu, menurut laporan Developer Coefficient tahun 2018 dari Stripe, para pengembang menghabiskan sekitar 17,3 jam per minggu, atau 42% dari total waktu kerja mereka, untuk menangani bug dan memlihara sistem [4]. Hal ini menghambat peluncuran fitur-fitur baru dan membahayakan keuangan proyek.

Permasalahan tersebut salah satunya dipengaruhi oleh perbedaan cara kerja antara desainer dan pengembang. Kedua peran ini sering kali belum sepenuhnya memahami kebutuhan serta proses kerja masing-masing, sehingga rentan menimbulkan miskomunikasi [5], [6]. Desainer berfokus pada aspek visual melalui perangkat lunak desain, sedangkan pengembang menerjemahkan

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

rancangan tersebut ke dalam logika dan kode [7]. Perbedaan ini menjadi paling jelas saat beralih dari desain statis ke antarmuka yang berfungsi penuh [6].

Kesenjangan dalam proses tersebut dapat memunculkan *design debt*. Pada *design handoff* yang dilakukan secara manual, pengembang sering perlu mengukur kembali berbagai nilai dan gaya visual dari desain untuk diterapkan ke dalam kode [6], [8]. Akumulasi ketidakkonsistenan visual dapat berkembang menjadi utang teknis *front-end* seiring bertambahnya jumlah iterasi pengembangan. Akibatnya, *refactoring* dapat menjadi proses yang memakan banyak sumber daya dan memperlambat siklus pengembangan perangkat lunak [9], [10].

Pengembang sering kali terpaksa membuat *style* yang menyesatkan akibat prosedur serah terima yang ambigu. Istilah untuk hal ini adalah *CSS Code Smells*. Ini merupakan gejala kelemahan desain struktural yang terjadi ketika kode untuk gaya visual ditulis secara langsung tanpa merujuk pada abstraksi semantik apa pun. Akibat prosedur yang ceroboh ini, utang teknis menumpuk, dan kode yang dihasilkan menjadi terlalu rapuh serta terlalu terikat untuk bertahan dalam jangka panjang. Studi yang dilakukan oleh Lahti et al. (2021) [11] menyatakan bahwa penumpukan *code smells* semacam ini menyulitkan pemeliharaan jangka panjang. Hal ini membuat pengembang harus bekerja lebih keras untuk memperbaikinya di masa depan [12].

Industri rekayasa perangkat lunak telah mulai mengadopsi arsitektur *Design Tokens* sebagai cara untuk mengatasi ketidakefisienan ini. Konsep ini memecah pilihan desain seperti warna, font, dan spasi menjadi variabel dengan nama yang bermakna, seperti color-primary-500 [13], [14]. Penggunaan token menciptakan *single source of truth* yang netral terhadap platform. Hal ini memungkinkan tim menjaga konsistensi visual di seluruh produk dan mempercepat alur kerja dengan memperbarui nilai di satu tempat yang otomatis akan diperbarui ke seluruh sistem [10], [15].

Saat ini, design tokens tidak lagi hanya digunakan sebagai alat bantu penamaan bagi desainer. Konsep ini telah berkembang dengan memanfaatkan pendekatan berbasis Application Programming Interface (API) untuk mengotomatisasi proses penerjemahan desain ke dalam kode. Melalui infrastruktur

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

tersebut, spesifikasi visual yang sebelumnya bersifat abstrak dapat diubah menjadi data terstruktur seperti JSON. Data ini berperan menghubungkan ekosistem desain antarmuka pengguna dengan repositori serta kerangka kerja pemrograman. [16].

Design tokens juga membuat penerapan standar internasional menjadi jauh lebih mudah secara sistematis. Perubahan mode warna seperti mode terang dan gelap, serta penerapan persyaratan kontras visual dari *Web Content Accessibility Guidelines* (WCAG), saat ini dapat diterapkan secara otomatis ke semua elemen kode. Hal ini menjamin aplikasi tetap inklusif tanpa perlu melakukan perubahan sebagian yang cenderung menimbulkan kesalahan [17].

Figma, sebagai salah satu alat desain antarmuka yang dominan telah memperkenalkan fitur Variables yang memungkinkan penerapan konsep *design tokens* secara langsung dalam lingkungan desain [15]. Fitur ini berhasil mendekatkan struktur visual dengan pola pikir pemrograman, namun masih terdapat hambatan dalam alur kerja integrasinya. Tidak adanya alat otomatisasi internal mengharuskan pengembang untuk tetap mengekstraksi nilai Figma Variables secara manual ke dalam kerangka kerja modern seperti Tailwind CSS. Namun selama proses konversi dari desain ke lingkungan pengembangan masih memerlukan tangan manusia, maka kesalahan pengetikan dan *technical debt* akan tetap muncul [18], [19].

Proses ekstraksi variabel yang masih dilakukan secara manual ini mengganggu skalabilitas dan kelancaran alur Continuous Integration/Continuous Deployment (CI/CD). Proses serah terima antara desainer dan pengembang rentan terhadap kehilangan informasi, karena membutuhkan interpretasi manusia untuk menerjemahkan maksud visual dan struktural ke dalam bentuk kode [20]. Kerangka kerja *front-end* seperti Tailwind CSS menuntut pemetaan nilai token yang presisi [21]. Ketika proses penyalinan data dari Figma Variables ke berkas konfigurasi kode masih dilakukan secara manual satu per satu, ketidaksesuaian dalam penerapan arsitektur pun sulit dihindari. Jika dibiarkan, kondisi ini dapat memperlambat proses perbaikan dan mengakibatkan pembengkakan biaya pemeliharaan di masa depan.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Berdasarkan permasalahan tersebut, tugas akhir ini bertujuan untuk merancang dan membangun sebuah *plugin* Figma dengan nama Token2Code. Token2Code dibuat untuk mengekstraksi data secara otomatis dari Figma Variables dan mengonversinya ke kerangka kerja Tailwind CSS serta format CSS Variables. *Plugin* ini ditujukan bagi pengembang perangkat lunak dan desainer antarmuka yang memahami struktur kode Tailwind CSS dan CSS Variables serta memahami penggunaan aplikasi Figma. Dengan adanya *plugin* ini, proses desain ke kode diharapkan dapat berjalan secara otomatis sehingga mengurangi kesalahan penerapan token secara manual.

1.2 Rumusan Masalah

Dari latar belakang yang telah diuraikan, masalah yang dikaji dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana merancang bangun mekanisme ekstraksi data Figma Variables bertipe *Color* dan *Float* agar dapat diolah menjadi *design token* oleh *plugin* Token2Code?
2. Bagaimana membangun mekanisme konversi *design token* tersebut menjadi dua format *output* kode, yaitu deklarasi CSS Variables dan konfigurasi Tailwind CSS versi 3?
3. Bagaimana tingkat fungsionalitas dan *usability* plugin Token2Code yang dihasilkan dalam mengotomatisasi proses desain ke kode, berdasarkan hasil pengujian *Black Box* dan *System Usability Scale* (SUS)?

1.3 Batasan Masalah

Agar pengerjaan tugas akhir ini tetap fokus dan sesuai dengan rumusan masalah yang sudah ditetapkan, batasan masalah yang berlaku adalah sebagai berikut:

1. Sistem hanya mengekstrak Figma Variables bertipe *Color* dan *Float*. Variabel *String* dan *Boolean* tidak termasuk dalam sistem ini.
2. Seluruh proses ekstraksi data melalui *Application Programming Interface* (API) Figma dan konversi kode berjalan di sisi klien. *Plugin* ini

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

tidak terhubung ke *backend* server maupun *version control system* seperti GitHub.

3. Sistem ini menghasilkan dua jenis *output* yang penempatannya diatur secara manual oleh pengembang di repositori proyek. *Output* pertama adalah CSS Variables yang dibungkus dalam selektor :root, yang dapat langsung ditempelkan atau dimasukkan ke dalam file CSS global proyek seperti global.css, style.css, atau file CSS utama lainnya. *Output* kedua adalah struktur objek JavaScript yang dapat langsung ditempelkan ke dalam blok theme.extend pada berkas konfigurasi Tailwind CSS versi 3 (tailwind.config.js). Sistem tidak mencakup otomatisasi kode langsung ke dalam struktur kode repositori pengembang. Dukungan format untuk *framework* pendukung antarmuka lainnya berada di luar batasan sistem.
4. Pengguna yang menjadi sasaran sistem ini terbatas pada pengembang perangkat lunak dan desainer antarmuka yang sudah memahami Tailwind CSS, CSS Variables serta cara menggunakan aplikasi Figma. Sistem ini tidak dirancang untuk pengguna umum yang belum memiliki pemahaman tersebut.

1.4 Tujuan

Tujuan dari rancang bangun *plugin* Token2Code ini sebagai berikut:

1. Merancang bangun mekanisme yang mengekstraksi data Figma Variables bertipe Color dan Float menjadi design token pada plugin Token2Code.
2. Membangun mekanisme konversi design token menjadi dua format *output* kode, yaitu CSS Variables dan konfigurasi Tailwind CSS versi 3.
3. Mengevaluasi fungsionalitas dan tingkat usability plugin Token2Code dalam mengotomatisasi proses desain ke kode melalui pengujian *Black Box* dan SUS.

1.5 Manfaat

Pengembangan *plugin* ini diharapkan dapat memberikan manfaat sebagai berikut:

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

1. Mempermudah implementasi desain ke kode dalam proyek pengembangan perangkat lunak.

   Warna dan jarak adalah dua token desain yang dapat diambil secara otomatis oleh sistem ini. Pengembang tidak lagi perlu memeriksa setiap elemen visual satu per satu. Metode ini mengotomatiskan proses pengubahan desain UI menjadi kode sekaligus mengurangi kemungkinan terjadinya kesalahan.
2. Membantu memusatkan data desain melalui *single source of truth*.

   Figma Variables berfungsi sebagai repositori untuk semua persyaratan desain. Sinkronisasi terpusat seperti ini menghilangkan spesifikasi ganda yang sering membingungkan dan muncul selama proses pengembangan. Nilai-nilai visual di browser sangat terkait erat dengan karya asli desainer, sehingga nilai-nilai tersebut akan selalu akurat.
3. Mempermudah pemeliharaan dan skalabilitas kode.

   Sistem ini menghasilkan kode yang jauh lebih rapi berkat penggunaan variabel CSS dan pengaturan Tailwind CSS. Pengembang hanya perlu melakukan satu pembaruan terpusat untuk mengubah gaya visual, seperti beralih antara mode terang dan gelap. Meskipun antarmuka pengguna aplikasi berubah, sistem ini tetap menjaga kode tetap rapi dan mengurangi kemungkinan kode menjadi sulit dikelola.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, desain, pengkodean, dan pengujian yang telah dilakukan, plugin Token2Code untuk Figma berhasil mengotomatiskan proses konversi desain ke kode. Plugin ini secara otomatis mengambil data dari Figma Variables dan mengonversinya menjadi Variabel CSS serta konfigurasi Tailwind CSS versi 3. Berdasarkan rumusan masalah dan tujuan yang ditetapkan, berikut adalah kesimpulan yang diperoleh.

1. *Plugin* Token2Code berhasil dirancang bangun dengan mekanisme ekstraksi yang mampu mengambil data Figma Variables bertipe *Color* dan *Float* secara otomatis melalui Figma Plugin API. *Plugin* ini kemudian mengubahnya menjadi struktur *design token* yang dapat diproses lebih lanjut oleh sistem.
2. Mekanisme konversi berhasil mengubah *design token* menjadi dua format *output*, yaitu CSS Variables yang dibungkus dalam selektor :root, dan struktur objek JavaScript yang dapat langsung disalin ke blok theme.extend dalam berkas tailwind.config.js. Pengujian Black Box menunjukkan bahwa seluruh skenario berjalan sesuai hasil yang diharapkan. Hal ini berlaku baik untuk skenario dalam kondisi ideal, maupun untuk kasus penanganan kesalahan seperti data kosong dan tipe variabel yang tidak valid.
3. Evaluasi *usability* menggunakan *System Usability Scale* (SUS) terhadap 16 responden dengan latar belakang UI/UX Designer, Frontend Developer, dan Software Engineer menghasilkan skor rata-rata 78,3. Skor ini termasuk dalam kategori *Acceptable* dengan *Grade Scale* C dan predikat *Good*. Hasil ini menunjukkan bahwa plugin Token2Code berhasil mengotomatisasi proses desain ke kode. Meskipun begitu, efisiensi pada aspek konfigurasi manual seperti verifikasi penamaan token, masih menjadi hal yang perlu diperbaiki.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

5.2 Saran

Berdasarkan hasil dan keterbatasan yang ditemukan selama pengembangan plugin Token2Code, berikut beberapa saran untuk pengembangan selanjutnya:

1. Penerapan Rilis Versi 1.1 (Dukungan Tailwind CSS v4 & Mode Gelap): Saat tugas akhir ini ditulis, *plugin* versi 1.1 sebenarnya telah selesai dikembangkan. Terdapat fitur baru yaitu ekstraksi variabel yang memakai sintaks terbaru Tailwind CSS v4 dan antarmuka bertema *dark mode* dengan skema warna Flexoki. Namun, rilis tersebut masih dalam tahap peninjauan oleh tim Figma Community. Harapannya, pembaruan ini dapat segera dipublikasikan dan dimanfaatkan oleh pengguna pada iterasi berikutnya.
2. Ekspansi Dukungan Framework Frontend Lainnya: Untuk saat ini, plugin masih difokuskan pada keluaran CSS dan Tailwind CSS saja. Pengembangan selanjutnya dapat mempertimbangkan penambahan format ekspor untuk framework lain seperti React Native, Flutter, atau format JSON untuk diproses menggunakan Style Dictionary.
3. Ekstraksi Tipe Variabel yang Lebih Luas: Pada versi ini, sistem membatasi penarikan data hanya pada variabel bertipe COLOR dan FLOAT. Ke depannya, dukungan ini bisa diperluas ke tipe STRING dan BOOLEAN. Selain itu, pembacaan data tipografi juga berpotensi diintegrasikan ke alur ekstraksi yang sama.
4. Integrasi Langsung ke Version Control System: Plugin ini juga berpotensi dikembangkan dengan menambahkan fitur autentikasi GitHub atau GitLab, sehingga proses *handoff* bisa berjalan lebih otomatis. Dengan begitu, pengguna bisa langsung membuat Pull Request berisi kode hasil ekstraksi ke repositori mereka tanpa perlu mengunduh file secara manual terlebih dahulu.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

DAFTAR PUSTAKA

[1] S. Zhang, T. Zhang, J. Cheng, and S. Zhou, "Who is to Blame: A Comprehensive Review of Challenges and Opportunities in Designer-Developer Collaboration," *Proc. ACM Hum.-Comput. Interact.*, vol. 9, no. 2, p. CSCW207:1-CSCW207:32, May 2025, doi: 10.1145/3711105.

[2] Kathryn, Nora, and Manuela, "State of Designer-Developer Collaboration 2024 | Telerik." Accessed: Feb. 19, 2026. [Online]. Available: https://www.telerik.com/design-system/designer-developer-collaboration-survey-2024/

[3] S. Inc, "Software Quality Issues in the U.S. Cost an Estimated $2.41 Trillion in 2022." Accessed: Aug. 17, 2026. [Online]. Available: https://www.prnewswire.com/news-releases/software-quality-issues-in-the-us-cost-an-estimated-2-41-trillion-in-2022--301695684.html

[4] Stripe, "The Developer Coefficient: Software Engineering Efficiency and Its $300B Impact on Global GDP," The Developer Coefficient: Software Engineering Efficiency and Its $300B Impact on Global GDP. Accessed: Aug. 17, 2026. [Online]. Available: https://stripe.com/files/reports/the-developer-coefficient.pdf

[5] N. Juristo, A. Moreno, M.-I. Sanchez-Segura, and M. C. C. Baranauskas, "A Glass Box Design: Making the Impact of Usability on Software Development Visible," in *Human-Computer Interaction – INTERACT 2007*, C. Baranauskas, P. Palanque, J. Abascal, and S. D. J. Barbosa, Eds., Berlin, Heidelberg: Springer, 2007, pp. 541–554. doi: 10.1007/978-3-540-74800-7_49.

[6] N. Maudet, G. Leiva, M. Beaudouin-Lafon, and W. Mackay, "Design Breakdowns: Designer-Developer Gaps in Representing and Interpreting Interactive Systems," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, in CSCW '17. New York, NY, USA: Association for Computing Machinery, Feb. 2017, pp. 630–641. doi: 10.1145/2998181.2998190.

STATE ISLAMIC UNIVERSITY SUNAN KALIJAGA YOGYAKARTA

[7] M. W. Newman and J. A. Landay, “Sitemaps, storyboards, and specifications: a sketch of Web site design practice,” in *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, in DIS ’00. New York, NY, USA: Association for Computing Machinery, Aug. 2000, pp. 263–274. doi: 10.1145/347642.347758.

[8] Zeplin, “What goes wrong with designer-developer collaboration? — new insights and solutions · Zeplin Gazette,” Zeplin. Accessed: Feb. 19, 2026. [Online]. Available: https://blog.zeplin.io/design-delivery/designer-developer-collaboration-insights-and-statistics/

[9] UXPin, “What Are Design Tokens?,” Studio by UXPin. Accessed: Feb. 19, 2026. [Online]. Available: https://www.uxpin.com/studio/blog/what-are-design-tokens/

[10] T. Saari, “Creating a Design Token Library for ABB’s CommonUX Design System.” Accessed: Feb. 19, 2026. [Online]. Available: http://www.theseus.fi/handle/10024/227894

[11] J. R. Lahti, A.-P. Tuovinen, and T. Mikkonen, “Experiences on Managing Technical Debt with Code Smells and AntiPatterns,” Mar. 21, 2021, *arXiv*: arXiv:2103.11486. doi: 10.48550/arXiv.2103.11486.

[12] “Technical debt cripples software developer productivity | Proceedings of the 2018 International Conference on Technical Debt.” Accessed: Aug. 17, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3194164.3194178

[13] P. Ahmet Erkan, “What Are Design Tokens and Why Should We Use Them?,” DEV Community. Accessed: Feb. 19, 2026. [Online]. Available: https://dev.to/aepasahan/what-are-design-tokens-and-why-should-we-use-them-1pj2

[14] “Intro to Design Tokens | Tokens Studio for Figma.” Accessed: Feb. 19, 2026. [Online]. Available: https://docs.tokens.studio/fundamentals/design-tokens

[15] C. Ayres, “The Future of Design Systems is Semantic | Figma Blog,” Figma. Accessed: Feb. 19, 2026. [Online]. Available: https://www.figma.com/blog/the-future-of-design-systems-is-semantic/

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

[16] "(PDF) DesignSystemsJS - Building a Design Systems API for aiding standardization and AI integration," in *ResearchGate*, doi: 10.1109/CoNTESA61248.2023.10384889.

[17] H. Shah, "Harnessing Web Accessibility Tools for WCAG 2.1 Migration of a Design System," in *2023 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA)*, Dec. 2023, pp. 54–60. doi: 10.1109/CoNTESA61248.2023.10384953.

[18] M. Piirainen, "Feasible design : transforming design fundamentals to usable code," *Design osien transformaatio käytettäväksi koodiksi*, 2023, Accessed: Feb. 19, 2026. [Online]. Available: https://lutpub.lut.fi/handle/10024/166467

[19] S. Xiao, Y. Chen, J. Li, L. Chen, L. Sun, and T. Zhou, "Prototype2Code: End-to-end Front-end Code Generation from UI Design Prototypes," May 08, 2024, *arXiv*: arXiv:2405.04975. doi: 10.48550/arXiv.2405.04975.

[20] Y. Arik, "AI-Assisted Design-to-Code Pipelines: Transforming Mobile Development Through Automated Component Generation," *International Journal of Research Publications*, vol. 200, no. 1, pp. 218–241, Jul. 2026, doi: 10.47119/IJRP1002001720269537.

[21] M. Singireddy, "Balancing Technical Debt Reduction and Feature Velocity in Legacy Front-End Systems," *International Journal of Emerging Trends in Computer Science and Information Technology*, vol. 7, no. 1, pp. 290–295, Mar. 2026, doi: 10.63282/3050-9246.IJETCSIT-V7I1P142.

[22] S. Newman, *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media, Inc., 2015.

[23] G. Fairbanks, *Just Enough Software Architecture: A Risk-driven Approach*. Marshall & Brainerd, 2010.

[24] "Introduction | Developer Docs." Accessed: Aug. 19, 2026. [Online]. Available: https://developers.figma.com/docs/plugins/

[25] Envato Tuts+, *Create Your First Figma Plugin | FREE COURSE*, (Nov. 24, 2021). Accessed: Mar. 05, 2026. [Online Video]. Available: https://www.youtube.com/watch?v=pFGhMr6rDhc

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

[26] Y. Chen and L. Chen, “PSD2Code: Automated Front-End Code Generation from Design Files via Multimodal Large Language Models,” Nov. 06, 2025, *arXiv*: arXiv:2511.04012. doi: 10.48550/arXiv.2511.04012.

[27] “(PDF) Cloud software code generation via knowledge graphs and multi-modal learning.” Accessed: Aug. 17, 2026. [Online]. Available: https://www.researchgate.net/publication/393371590_Cloud_software_code_generation_via_knowledge_graphs_and_multi-modal_learning

[28] L. Kalbag, “Why use Design Tokens? 7 practical reasons designers and developers swear by,” Penpot Blog. Accessed: Feb. 19, 2026. [Online]. Available: https://penpot.app/blog/why-use-design-tokens/

[29] “Tokens, variables, and styles - Update: Introduction to design systems - YouTube.” Accessed: Mar. 05, 2026. [Online]. Available: https://www.youtube.com/watch?v=JyCmacSyDY4

[30] “Design Tokens Community Group.” Accessed: Aug. 18, 2026. [Online]. Available: https://www.w3.org/community/design-tokens/

[31] LobeHub, “design-tokens | Skills Marketplace · LobeHub,” LobeHub. Accessed: Aug. 18, 2026. [Online]. Available: https://lobehub.com/skills/tyler-r-kendrick-agent-skills-design-tokens

[32] “Guide to variables in Figma – Figma Learn - Help Center.” Accessed: Mar. 05, 2026. [Online]. Available: https://help.figma.com/hc/en-us/articles/15339657135383-Guide-to-variables-in-Figma

[33] Figma, *Figma tutorial: Intro to variables*, (Jun. 21, 2023). Accessed: Mar. 05, 2026. [Online Video]. Available: https://www.youtube.com/watch?v=1ONxxlJnvdM

[34] “An introduction to design tokens,” Donux. Accessed: Aug. 18, 2026. [Online]. Available: https://donux.com/blog/introduction-to-design-tokens

[35] “What Are Design Tokens? A Practical Guide,” OverlayQA. Accessed: Aug. 18, 2026. [Online]. Available: https://overlayqa.com/blog/what-are-design-tokens/

[36] “Design Tokens Community Group.” Accessed: Aug. 18, 2026. [Online]. Available: https://www.w3.org/community/design-tokens/

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

[37] LobeHub, “design-tokens | Skills Marketplace · LobeHub,” LobeHub. Accessed: Aug. 18, 2026. [Online]. Available: https://lobehub.com/skills/tyler-r-kendrick-agent-skills-design-tokens

[38] “CSS custom properties for cascading variables - CSS | MDN,” MDN Web Docs. Accessed: Aug. 19, 2026. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS/Guides/Cascading_variables

[39] “Just-in-Time Mode - Tailwind CSS.” Accessed: Aug. 19, 2026. [Online]. Available: https://tailwindcss.com/docs/just-in-time-mode

[40] “(PDF) Human Factors in DevOps: Cognitive Load, Developer Experience, and Team Collaboration,” *ResearchGate*, Accessed: Aug. 17, 2026. [Online]. Available: https://www.researchgate.net/publication/394035815_Human_Factors_in_DevOps_Cognitive_Load_Developer_Experience_and_Team_Collaboration

[41] A. Razzaq, J. Buckley, Q. Lai, T. Yu, and G. Botterweck, “A Systematic Literature Review on the Influence of Enhanced Developer Experience on Developers’ Productivity: Factors, Practices, and Recommendations,” *ACM Comput. Surv.*, vol. 57, no. 1, p. 13:1-13:46, Oct. 2024, doi: 10.1145/3687299.

[42] T. Guthardt, J. Kosiol, and O. Hohlfeld, “Low-code vs. the developer: An empirical study on the developer experience and efficiency of a no-code platform | Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems,” ACM Conferences. Accessed: Sep. 17, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3652620.3688332

[43] “(PDF) Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi,” ResearchGate. Accessed: Aug. 21, 2026. [Online]. Available: https://www.researchgate.net/publication/346397070_Analisis_Metode_Waterfall_Untuk_Pengembangan_Sistem_Informasi

[44] A. Demirag, E. N. D. Öztürk, and C. Ünal, “Analysis and Comparision of Waterfall Model and Agile Approach in Software Projects,” *AJIT-e: Academic*

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

Journal of Information Technology, vol. 14, no. 54, pp. 183–203, Sep. 2023, doi: 10.5824/ajite.2023.03.002.x.

[45] "PENERAPAN METODE WATERFALL DALAM PERANCANGAN SISTEM INFORMASI PENGGAJIAN PADA SMK BINA KARYA KARAWANG," *ResearchGate*, doi: 10.35969/interkom.v14i4.78.

[46] H. Nur, "Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan," *Generation Journal*, vol. 3, no. 1, pp. 1–10, Jan. 2019, doi: 10.29407/gj.v3i1.12642.

[47] "(PDF) The Waterfall Model in Large-Scale Development," *ResearchGate*, doi: 10.1007/978-3-642-02152-7_29.

[48] O. Baniaș, D. Florea, R. Gyalai, and D.-I. Curiac, "Automated Specification-Based Testing of REST APIs," *Sensors*, vol. 21, no. 16, p. 5375, Jan. 2021, doi: 10.3390/s21165375.

[49] S. Supriyono, "Software Testing with the approach of Blackbox Testing on the Academic Information System," *IJISTECH (International Journal of Information System and Technology)*, vol. 3, no. 2, pp. 227–233, May 2020, doi: 10.30645/ijistech.v3i2.54.

[50] "Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing," *INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS : Journal of Information Management*, vol. 5, no. 2, pp. 97–110, Jun. 2021, doi: 10.51211/imbi.v5i2.1561.

[51] "(PDF) SUS: a retrospective," ResearchGate. Accessed: Aug. 14, 2026. [Online]. Available: https://www.researchgate.net/publication/285811057_SUS_a_retrospective

[52] "(PDF) The Factor Structure of the System Usability Scale," in *ResearchGate*, 2026. doi: 10.1007/978-3-642-02806-9_12.

[53] K. T. Nugroho, B. Julianto, and D. F. N. Ms, "Usability Testing pada Sistem Informasi Manajemen AKN Pacitan Menggunakan Metode System Usability Scale," *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, vol. 11, no. 1, pp. 74–83, Apr. 2022, doi: 10.23887/janapati.v11i1.43209.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

[54] N. Huda, F. Habrizons, A. Satriawan, M. Iranda, and T. Pramuda, “Analisis Usability Testing Menggunakan Metode SUS (System Usability Scale) Terhadap Kepuasan Pengguna Aplikasi Shopee,” *1*, vol. 8, no. 2, pp. 208–220, Aug. 2023, doi: 10.51717/simkom.v8i2.158.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA