



Artikel Penelitian

Pengembangan Modul Chart Builder, Compute Variable, dan Recode Variable pada Aplikasi Statify

Muhamad Iqbal Putra Pratama^a, Erna Nurmawati^{b*}

^{a,b} Program Studi D-IV Komputasi Statistik, Politeknik Statistika STIS, Jl. Otto Iskandardinata No.64C, Jakarta 13330, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 18 September 2025

Revisi Akhir: 27 April 2026

Diterbitkan Online: 05 Mei 2026

KATA KUNCI

Statify,
Chart builder,
Compute variable,
Recode variable,
Feature-driven development

KORESPONDENSI

E-mail: erna.nurmawati@stis.ac.id*

A B S T R A C T

Politeknik Statistika STIS sebagai perguruan tinggi vokasi di bidang statistika terapan dan komputasi statistik membutuhkan alat bantu analisis statistik yang ekonomis, fleksibel, dan mudah digunakan. Selama ini, alat bantu analisis statistik yang ada memiliki kelebihan dan keterbatasannya masing-masing dalam penggunaannya di Polstat STIS. Untuk itu, dikembangkan aplikasi web bernama Statify yang menggabungkan kelebihan dari berbagai alat analisis yang ada serta dirancang spesifik untuk Penggunaan di Polstat STIS. Statify menyediakan berbagai modul analisis, beberapa diantaranya adalah modul Chart Builder untuk menjawab kebutuhan visualisasi data, Compute Variable untuk memfasilitasi pembentukan variabel baru, dan Recode Variable untuk mengkodekan ulang variabel. Aplikasi ini dibangun secara modular menggunakan pendekatan metode pengembangan *Feature-Driven Development (FDD)* dengan teknologi *Client-Side Rendering (CSR)* berbasis Next.js. Modul Chart Builder memanfaatkan pustaka D3.js dan Apache ECharts untuk visualisasi grafik, modul Compute Variable menggunakan math.js untuk komputasi matematis, sedangkan modul Recode dibuat secara native tanpa menggunakan pustaka tambahan dalam implementasinya. Dengan adanya Aplikasi Statify, diharapkan dapat menunjang perkuliahan dan penelitian di Polstat STIS secara lebih optimal dalam rangka mendukung pelaksanaan Tridharma Perguruan Tinggi

1. PENDAHULUAN

Politeknik Statistika STIS selanjutnya disebut Polstat STIS adalah perguruan tinggi di lingkungan Badan Pusat Statistik Republik Indonesia yang menyelenggarakan program pendidikan vokasi dalam bidang ilmu statistika terapan dan komputasi statistik [1]. Sebagai perguruan tinggi yang berfokus pada ilmu statistik dan komputasi statistik, Polstat STIS membutuhkan alat bantu analisis data dan statistik yang mendukung dalam pelaksanaan kegiatan pendidikan, penelitian, dan pengabdian masyarakat yang merupakan bagian dari kewajiban Tridharma Perguruan Tinggi [2].

Saat ini, beragam perangkat lunak statistik telah berkembang dan populer digunakan secara luas di berbagai sektor. Berdasarkan analisis jumlah artikel ilmiah yang ditemukan di Google Scholar pada tahun 2022 oleh Muechen (2023), IBM SPSS Statistics

masih memimpin dalam jumlah penggunaan untuk artikel ilmiah. Hal ini menunjukkan tingkat adopsi yang kuat untuk perangkat lunak ini di kalangan peneliti. Di sisi lain, bahasa pemrograman dengan kapabilitas analisis tinggi seperti R dan Python juga menunjukkan popularitas yang sangat signifikan dan terus mengalami perkembangan [3].

Di Polstat STIS, dua dari beberapa alat analisis statistik yang digunakan civitas akademika Polstat STIS saat ini adalah IBM SPSS Statistics (selanjutnya disebut SPSS) dan R. Alat bantu analisis ini digunakan untuk mendukung kegiatan analisis data dan statistik saat perkuliahan maupun penelitian. Namun, penggunaan perangkat lunak SPSS terkendala oleh masalah biaya lisensi yang mahal. Di sisi lain, meskipun RStudio yang digunakan bersifat bebas biaya lisensi dan *open-source* [4], penggunaan Rstudio yang berbasis script memerlukan pengetahuan pemrograman yang lebih mumpuni dibandingkan SPSS.

Sebagai solusi, perlu dikembangkan aplikasi Statify, yaitu aplikasi yang menyediakan fitur analisis statistik berbasis GUI, sehingga memiliki antarmuka yang intuitif, seperti SPSS dan bebas lisensi seperti Rstudio Desktop. Aplikasi ini dirancang menggunakan Next.js dengan pendekatan *Client-Side Rendering (CSR)* untuk memastikan performa tinggi dan pengalaman pengguna yang baik. Hal ini didukung dengan temuan oleh Rosquist dan Luttu (2017) di mana dalam konteks visualisasi data *real-time* menggunakan D3.js, CSR memberikan performa yang lebih baik dibandingkan *Server-Side Rendering (SSR)* [5]. Selain itu, Statify dibangun berbasis web memungkinkan pengguna mengaksesnya secara lebih fleksibel dan cepat karena tanpa memerlukan proses instalasi. Pengembangan dilakukan menggunakan metode *Feature-Driven Development (FDD)* dimana setiap komponen aplikasi, termasuk berbagai fitur analisis, akan dikembangkan secara modular oleh anggota tim.

Penelitian ini difokuskan pada tiga modul utama dalam Statify, yaitu modul Chart Builder, Compute Variable, dan Recode variable. Modul Chart Builder memungkinkan pengguna untuk membuat berbagai jenis visualisasi data dalam bentuk grafik yang dapat disesuaikan dengan kebutuhan analisis. Modul ini dibangun menggunakan pustaka utama D3.js, yang dipilih karena fleksibilitas dan performanya dalam menghasilkan visualisasi [6]. Hal ini diperkuat oleh penelitian Dogadina dan Voronin (2024) yang melakukan analisis perbandingan alat visualisasi data dan menemukan bahwa D3.js unggul dalam fleksibilitas serta kualitas hasil visualisasi [7]. Namun, karena keterbatasannya dalam membentuk grafik tiga dimensi, untuk grafik jenis ini digunakan modul tambahan yaitu Apache Echarts yang mendukung visualisasi 3D [8].

Selanjutnya, modul Compute Variable memungkinkan pengguna untuk membuat variabel baru menggunakan variabel yang sudah ada berdasarkan ekspresi matematika tertentu. Modul ini dibangun menggunakan pustaka utama Math.js, yang mendukung operasi komputasi simbolik dan numerik secara efisien dan lengkap untuk JavaScript [9]. Modul Recode Variable memungkinkan pengguna untuk mengkodekan ulang variabel menjadi nilai atau kategori baru seperti mengubah nilai variabel numerik usia menjadi variabel kategorik usia yang memiliki kategori muda, dewasa, dan lanjut usia. Modul ini dikembangkan secara *native* tanpa pustaka tambahan.

Dengan adanya pengembangan modul Chart Builder, Compute Variable, dan Recode Variable dalam aplikasi Statify ini, diharapkan dapat mendukung civitas akademika Polstat STIS dalam melaksanakan Tridharma Perguruan Tinggi.

2. METODE

2.1. Metode Pengumpulan Data

2.1.1. Studi Pustaka

Studi pustaka adalah serangkaian kegiatan yang berkenaan dengan metode pengumpulan data pustaka, membaca, mencatat dan mengolah bahan penelitian [10]. Studi pustaka dilakukan menggunakan jurnal, artikel, maupun media lain dengan tujuan

mengumpulkan informasi yang dibutuhkan terkait pembangunan modul.

2.1.2. Observasi

Observasi adalah teknik pengumpulan data yang dilakukan melalui sesuatu pengamatan, dengan disertai pencatatan-pencatatan terhadap keadaan atau perilaku objek sasaran [15]. Observasi dalam penelitian ini dilakukan dengan mengamati dan mengevaluasi software statistik SPSS versi 27 dengan fokus utama pada menu Chart Builder, Compute Variable, Recode into Same Variables, dan Recode into Different Variables. Bertujuan untuk mengidentifikasi fitur, fungsi, alur kerja, dan antarmuka sebagai referensi dalam pengembangan modul.

2.2.1. Kuesioner

Kuesioner merupakan metode pengumpulan data yang dilakukan dengan cara memberikan seperangkat uji berupa pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya [11]. Metode kuesioner digunakan untuk memperoleh data evaluasi sistem dengan kerangka kuesioner System Usability Scale (SUS) dan Software Usability Measurement Inventory (SUMI).

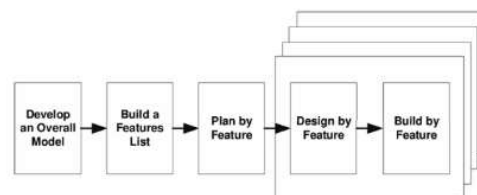
2.2. Tahapan Penelitian

2.2.1. Analisis Kebutuhan

Analisis dilakukan melalui studi pustaka dan observasi terhadap perangkat lunak statistik SPSS. Hasil analisis ini menghasilkan daftar kebutuhan fungsional dan nonfungsional yang akan menjadi dasar dalam perancangan aplikasi.

2.2.2. Perancangan

Perancangan dilakukan menggunakan metode Feature Driven Development (FDD). Berpedoman pada referensi [12] dan [13] tahapan FDD dalam penelitian ini ditampilkan pada Gambar 1.



Gambar 1. Tahapan FDD

1) Develop an overall model

Pada tahapan ini, peneliti bersama tim menentukan ruang lingkup, batasan, dan daftar modul yang akan dikembangkan bersama. Tahapan ini menghasilkan use case diagram yang memberikan gambaran umum terkait aplikasi Statify yang akan dikembangkan.

2) Build feature list

Peneliti melakukan analisis untuk menentukan daftar fitur yang perlu ada dan dikembangkan dari modul yang ditugaskan. Tahapan ini menghasilkan daftar fitur yang memberikan gambaran mengenai fitur-fitur apa saja yang ada dan akan dikembangkan dalam modul terkait.

3) Plan by feature

Peneliti membuat jadwal pengembangan dari seluruh daftar fitur. Jadwal dibuat dengan memetakan fitur ke dalam *milestone* dan iterasi-iterasi pendek yang akan dilakukan. Tahapan ini

menghasilkan jadwal pengembangan yang menjadi pedoman dalam pelaksanaan tahap selanjutnya.

4) *Design by feature*

Pada jadwal yang telah ditentukan, peneliti kemudian melakukan proses desain untuk setiap fitur. Tahap ini menghasilkan desain fitur dalam bentuk *sequence diagram*.

5) *Build by feature*

Berdasarkan desain yang telah dibuat, pada tahap ini peneliti membuat implementasi kode typescript untuk tiap fitur.

Tahap 4 dan 5 dalam FDD merupakan rangkaian proses iteratif untuk setiap fitur yang telah dijadwalkan. Setelah satu fitur selesai didesain dan diimplementasikan, proses akan kembali ke tahap 4 untuk fitur berikutnya hingga seluruh fitur dalam modul selesai dikembangkan.

2.2.3. *Evaluasi*

Fitur-fitur yang telah selesai diimplementasikan kemudian dievaluasi dengan beberapa pendekatan, yaitu:

1) Pengujian *black-box*

Pada tahap ini dilakukan pengujian terhadap fungsionalitas aplikasi. Peneliti membandingkan output dari aplikasi Statify dengan output dari SPSS untuk memastikan bahwa dengan inputan yang sama, keduanya akan menghasilkan output yang sama. Kesamaan output yang dimaksud mencakup jenis dan bentuk visualisasi grafik, serta output statistik.

2) Pengujian *white-box*

Pengujian *white-box* dilakukan dengan memeriksa logika internal dan struktur kode program untuk memastikan setiap bagian kode bekerja sebagaimana mestinya. Metode yang digunakan adalah *unit testing*, yaitu pengujian yang difokuskan pada unit terkecil dari kode program seperti fungsi atau komponen individual. *Unit testing* dilakukan secara terpisah untuk setiap modul yang dikembangkan. Pengujian ini bertujuan untuk memastikan bahwa setiap unit kode mampu menghasilkan keluaran yang benar sesuai dengan input yang diberikan, serta dapat menangani kasus-kasus tepi (*edge cases*) dengan baik.

3) Kuesioner SUS

Untuk evaluasi SUS, responden akan diberikan skenario pengujian kemudian responden memberikan penilaian menggunakan kuesioner SUS dengan skala Likert 5 poin, mulai dari "Sangat Tidak Setuju" hingga "Sangat Setuju". Sebanyak 12 Responden akan diminta menyelesaikan serangkaian *task* terlebih dahulu sebelum mengisi kuesioner.

4) Kuesioner SUMI

Pengujian SUMI melibatkan 12 pengguna yang diminta untuk menyelesaikan serangkaian skenario pengujian aplikasi Statify sebelum mengisi kuesioner. Hasil penilaian yang diberikan oleh responden kemudian Penulis kirimkan kepada admin SUMI untuk diproses dan menghasilkan laporan SUMISCO.

Hasil evaluasi ini digunakan sebagai umpan balik untuk mengetahui apakah sistem yang dikembangkan telah memenuhi kebutuhan pengguna serta dapat digunakan dengan efektif dan efisien.

3. HASIL

3.1. *Analisis Kebutuhan*

3.1.1. *Kebutuhan Fungsional*

Kebutuhan fungsional dari modul yang akan dibangun adalah sebagai berikut.

- 1) Sistem memfasilitasi pengguna untuk membangun berbagai jenis grafik
- 2) Sistem memfasilitasi pengguna untuk membuat variabel baru melalui penyusunan ekspresi matematika
- 3) Sistem memfasilitasi pengguna untuk mengkodekan ulang variabel
- 4) Sistem menampilkan output hasil pemrosesan
- 5) Sistem menyediakan log proses sebagai umpan balik dari setiap proses bisnis yang diselesaikan pengguna

3.1.2. *Kebutuhan Nonfungsional*

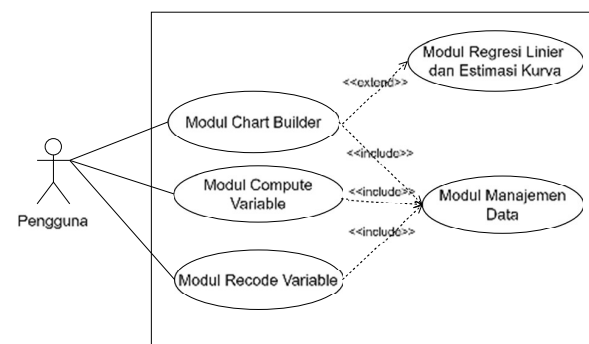
Kebutuhan nonfungsional dari modul yang dibangun adalah sebagai berikut.

- 1) Kategori: Kegunaan
Aspek: Antarmuka intuitif
Deskripsi: Antarmuka memungkinkan pengguna dapat memahami dan mengoperasikan fitur tanpa pelatihan khusus
- 2) Kategori: Interaktivitas
Aspek: Umpan balik visual secara langsung
Deskripsi: Sistem memberikan umpan balik visual secara langsung terhadap tindakan pengguna
- 3) Kategori: Reliabilitas
Aspek: Konsistensi output
Deskripsi: Output yang dihasilkan sistem harus selalu konsisten terhadap input yang sama

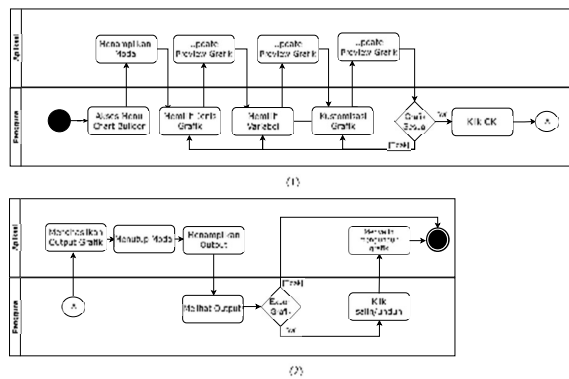
3.2. *Perancangan Sistem Usulan*

3.2.1. *Develop an Overall Model*

Gambar 2 menampilkan *usecase* diagram untuk model global. Modul yang akan dibuat melibatkan seorang aktor yang akan menggunakan modul sebagai pengguna.



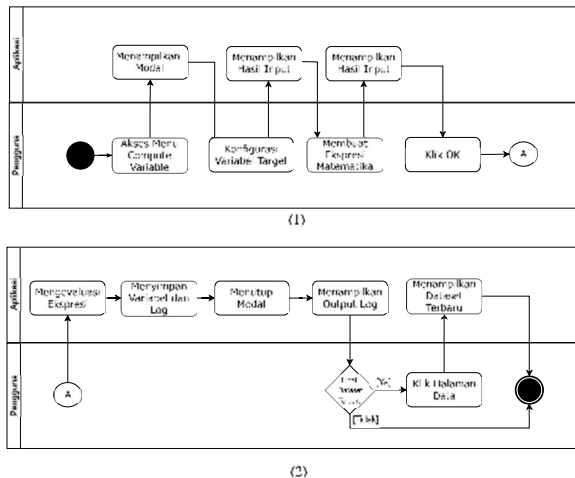
Gambar 2. *Usecase* Diagram Model Global



Gambar 3. Proses Bisnis Usulan Modul Chart Builder

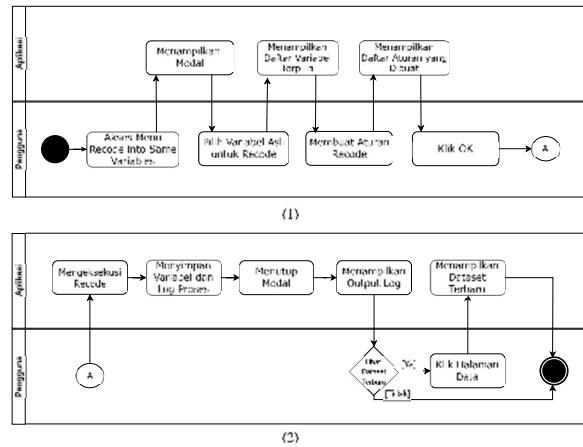
Gambar 3 menunjukkan proses bisnis usulan untuk modul Chart Builder. Proses dimulai ketika pengguna mengakses menu Chart Builder pada aplikasi, kemudian aplikasi menampilkan modal Chart Builder. Hingga pengguna klik “OK” untuk men-generate output grafik. Jika generate grafik berhasil dan pengguna ingin melihat output yang dihasilkan, pengguna dapat melihat di halaman Result.

Gambar 4 menunjukkan proses bisnis usulan untuk modul Compute Variable. Proses dimulai ketika pengguna mengakses menu Compute Variable pada aplikasi. Output Compute Variable berupa variable baru yang ditambahkan ke dataset dan log proses yang ditampilkan di halaman Result.

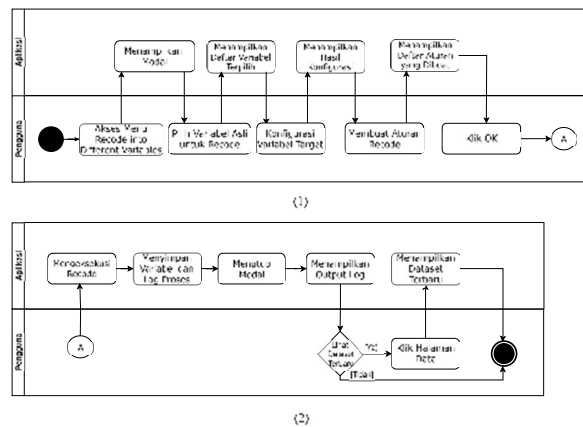


Gambar 4. Proses Bisnis Usulan Modul Compute Variable

Gambar 5 dan 6 menunjukkan proses bisnis usulan modul Recode Variable. Pada modul ini, terdapat 2 opsi rekoding, perbedaan antara keduanya adalah Recode into Same Variables langsung melakukan rekoding pada variabel asli di dataset, sedangkan Recode into Different Variables membuat variabel baru untuk menyimpan hasil rekoding, sehingga nilai variabel asli tetap tersedia di dataset.



Gambar 5. Proses Bisnis Usulan Modul Recode variable untuk menu Recode into Same Variables



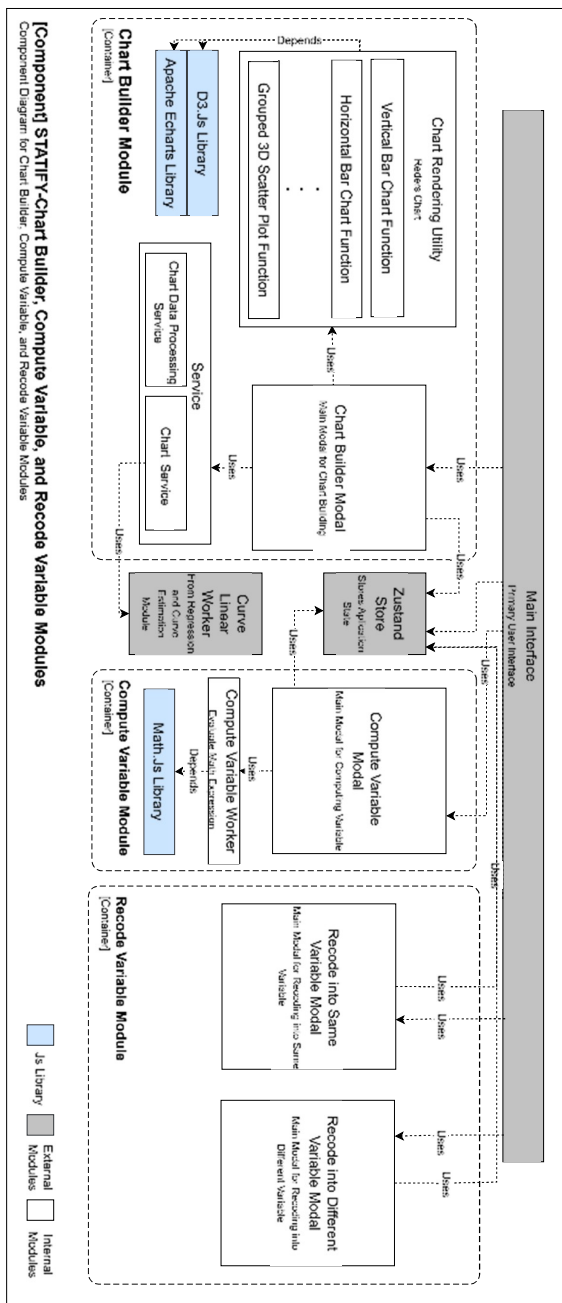
Gambar 6. Proses Bisnis Usulan Modul Recode Variable untuk menu Recode Different Variables

Gambar 7 adalah rancangan arsitektur komponen modul. Keduanya digunakan langsung melalui Main Interface ketika halaman diakses. Modul Chart Builder memproses input pengguna menjadi chartJSON dan dirender menjadi grafik menggunakan Chart Rendering Utility yang bergantung pada pustaka D3.js dan Apache Echarts. Di modul Compute Variable ekspresi matematika yang diinput pengguna dievaluasi oleh Compute Variable Worker dengan bantuan pustaka Math.js. Kemudian, Modul Recode Variable memproses rekoding secara langsung di komponen modal tanpa menggunakan pustaka tambahan.

3.2.2. Build a Feature List

Tahapan selanjutnya adalah mengidentifikasi fitur-fitur dalam setiap modul. Untuk mempermudah proses pengembangan, pengujian, dan dokumentasi, setiap fitur diberi ID unik yang bersifat sistematis. Untuk setiap modul:

- 1) Modul Chart Builder terdiri dari 19 fitur dengan 43 jenis grafik
- 2) Modul Compute Variabel terdiri dari 10 fitur dengan 58 jenis operator dan fungsi matematika yang disediakan
- 3) Modul Recode Variable terdiri dari 12 fitur dengan 9 opsi aturan rekode variable.



Gambar 7. Diagram C4 Komponen Modul yang dikembangkan

3.2.3. Plan by Feature

Setiap fitur yang telah didaftar, kemudian direncanakan jadwal pengembangannya. Dalam hal ini, peneliti merencanakan penjadwalan berikut.

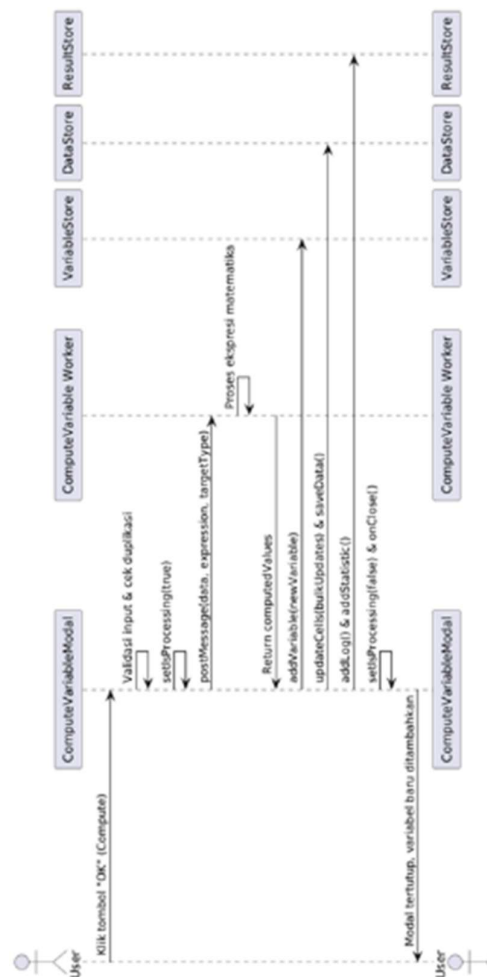
- 1) *Milestone 1:* Modul Chart Builder dengan prioritas kritis dan tinggi, pengerjaan mulai 1 Januari 2025 s.d. 16 Maret 2025, berdurasi 73 hari
- 2) *Milestone 2:* Modul Compute Variable dengan prioritas kritis dan tinggi, pengerjaan mulai 17 Maret 2025 s.d. 20 April 2025, berdurasi 35 hari
- 3) *Milestone 3:* Fitur tersisa serta tambahan fitur dalam modul Chart Builder dan Compute Variable, ditambah modul

tambahan yaitu modul Recode Variable pengerjaan mulai 1 Juni 2025–23 Juli 2025, berdurasi 53 hari

- 4) *Milestone 4: Testing,* pengerjaan mulai 23 Juli 2025–31 Juli 2025 berdurasi 8 hari.

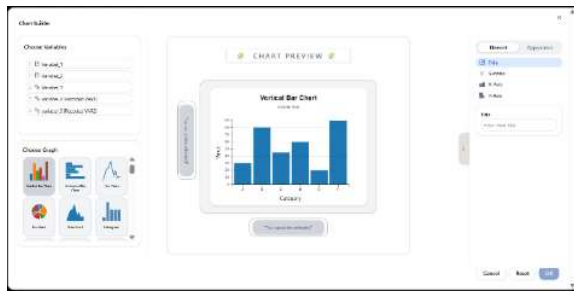
3.2.4. Design by Feature

Gambar 8 adalah salah satu *sequence diagram* dari fitur yang akan dikembangkan, yaitu untuk fitur menghasilkan variabel target dan log proses. *Sequence diagram* ini menunjukkan alur proses pembuatan variabel baru melalui modul Compute Variable yang menggunakan Web Worker untuk mengeksekusi ekspresi matematika secara asinkron agar antarmuka tetap responsif. Setelah pengguna menekan tombol OK, sistem memvalidasi input, mengirim data dan ekspresi ke *worker*, lalu memproses perhitungan di thread terpisah. Hasilnya, variabel baru ditambahkan ke dataset dan log proses beserta statistik disimpan di halaman Result.

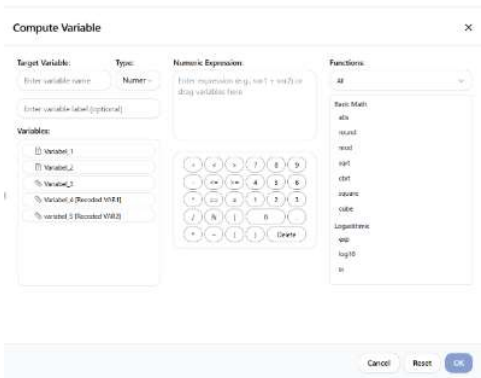


Gambar 8. Salah Satu *Sequence Diagram* Fitur yang dihasilkan

Tiap fitur kemudian diimplementasikan dan diintegrasikan dalam sistem utama. Tahap ini memastikan setiap fitur berfungsi sesuai desain dan dapat saling berintegrasi dengan fitur lainnya. Hasil akhir implementasi berupa fitur yang siap digunakan dengan antarmuka yang sesuai. Gambar 9 sampai 12 adalah hasil implementasi modal dari tiap modul pada aplikasi Statify.



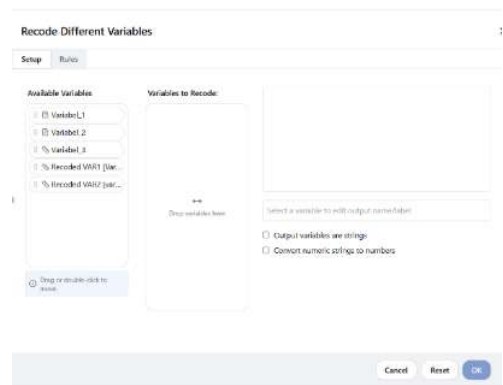
Gambar 9. Hasil Implementasi Modal Chart Builder



Gambar 10. Hasil Implementasi Modal Compute Variable

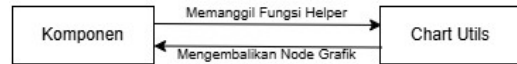


Gambar 11. Hasil Implementasi Modal Recode into Same Variables



Gambar 12. Hasil Implementasi Modal Recode into Different Variables

Di dalam modul Chart Builder, proses rendering grafik menjadi tanggung jawab Chart Utils yang di dalamnya terdapat *helper functions* untuk berbagai jenis grafik.



Gambar 13. Alur Pembuatan Grafik

Gambar 13 menunjukkan alur pembuatan grafik dalam modul. Untuk menampilkan grafik, komponen memanggil fungsi-fungsi yang ada di dalam Chart Utils. Chart Utils kemudian memproses data, membuat visualisasi, dan mengembalikan node grafik. Node ini berupa elemen SVG yang selanjutnya di-render pada antarmuka pengguna.



Gambar 14. Visualisasi Grafik dalam Modul Chart Builder

Sesuai dengan daftar grafik yang telah direncanakan, Chart Utils dapat menghasilkan 43 jenis grafik berbeda. Gambar 14 menampilkan preview visual seluruh grafik yang ada dalam modul, Setiap jenis grafik ini dihasilkan melalui pemanggilan fungsi spesifik dalam Chart Utils, dengan masing-masing fungsi dikodekan untuk menangani pembuatan grafik yang sesuai.

Untuk membuat fungsi visualisasi grafik dalam Chart Utils, peneliti mengimplementasikan kode D3.js. Beberapa peran D3.js dalam pembuatan grafik adalah sebagai berikut.

a. Menyiapkan data dan skala

Pada Gambar 15, D3.js digunakan untuk menyiapkan skala sumbu X pada grafik. Dengan `d3.scaleBand()`, skala ini memetakan kategori data ke posisi horizontal pada grafik. `domain()` mengatur kategori yang akan dipetakan, sedangkan `range()` menentukan rentang posisi pada sumbu X. `padding(0.1)` memberikan ruang antarkategori pada sumbu X untuk

memastikan setiap kategori memiliki jarak visual yang proporsional di antara satu sama lain.

```
const x = d3
    .scaleBand()
    .domain(data.map((d) => d.category))
    .range([marginLeft, width - marginRight])
    .padding(0.1);
```

Gambar 15. Implementasi a) Kode D3.JS

b. Membuat visualisasi grafik

Pada Gambar 16, D3.Js digunakan untuk menggambar grafik garis (line chart). `append("path")` menambahkan elemen `<path>` untuk menggambar garis. `attr("fill", "none")` memastikan area di bawah garis tidak terisi warna. `attr("stroke", "steelblue")` menetapkan warna garis menjadi steelblue, sedangkan `attr("stroke-width", 1.5)` mengatur ketebalan garis. `attr("d", line(data))` menetapkan path yang akan digambar, berdasarkan data yang diproses oleh fungsi `line`, yang mengonversi data menjadi bentuk path untuk menggambar garis.

```
svg
    .append("path")
    .attr("fill", "none")
    .attr("stroke", "steelblue")
    .attr("stroke-width", 1.5)
    .attr("d", line(data));
```

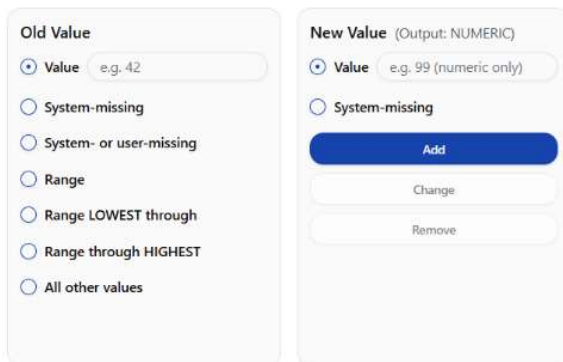
Gambar 16. Implementasi b) Kode D3.JS

Di dalam modul Compute Variable, evaluasi ekspresi matematika dilakukan menggunakan Math.js seperti pada Gambar 17. Math.js mengevaluasi ekspresi matematika yang diberikan dalam bentuk string. `math.evaluate()` mengevaluasi ekspresi dengan mempertimbangkan isi variabel context. Ekspresi yang ada dibuat dengan mengoperasikan beberapa variabel maupun konstanta.

```
let computedValue =
    math.evaluate(numericExpression, context);
```

Gambar 17. Implementasi Kode Math.Js

Untuk modul Recode Variable, disediakan 7 opsi rekoding yang bisa digunakan. Gambar 18 menampilkan keseluruhan opsi rekoding yang berhasil diimplementasikan.



Gambar 18. Opsi Recode Modul Recode Variable

Pada setiap proses *generate* output, setiap menu menghasilkan output dan log proses. Gambar 19, 20, 21, dan 22 log proses dari seluruh modul yang dikembangkan. Output ini dapat dilihat pada halaman Result.



Gambar 19. Log proses Chart Builder



Gambar 20. Log proses Compute Variable



Gambar 21. Log proses Recode into same Variables

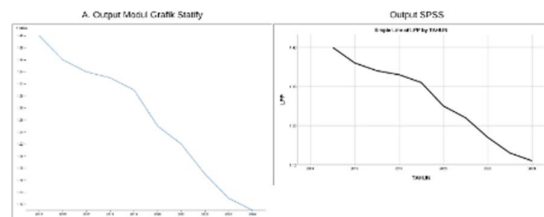


Gambar 22. Log proses Recode into Different Variables

3.3. Evaluasi

3.3.1. Black-box Testing

Black-box dilakukan melalui validasi output modul yang dikembangkan dibandingkan dengan SPSS.



Gambar 23. Validasi Output Modul Chart Builder

Gambar 23 adalah salah satu perbandingan output grafik dari modul Chart Builder Statify dengan SPSS. Kedua grafik menunjukkan tren yang sama, yaitu penurunan nilai LPP seiring waktu. Rentang sumbu Y keduanya juga sama, yaitu garis berada

diantara skala 1,10 sampai 1,40. Meskipun pemberian gaya visual keduanya terlihat berbeda, output Statify valid karena menampilkan hasil yang konsisten dengan SPSS.

A. Output Modul Transformasi Statify			B. Output SPSS		
TAHUN	LPP	LPP2	TAHUN	LPP	LPP2
2015	1,40	2,80	2015	1,40	2,80
2016	1,36	2,72	2016	1,36	2,72
2017	1,34	2,68	2017	1,34	2,68
2018	1,33	2,66	2018	1,33	2,66
2019	1,31	2,62	2019	1,31	2,62
2020	1,25	2,50	2020	1,25	2,50
2021	1,22	2,44	2021	1,22	2,44
2022	1,17	2,34	2022	1,17	2,34
2023	1,13	2,26	2023	1,13	2,26
2024	1,11	2,22	2024	1,11	2,22

Gambar 24. Validasi Output Modul Compute Variable

Gambar 24 adalah perbandingan output grafik dari modul Compute Variable dengan SPSS. Gambar sebelah kiri menunjukkan output dari Statify, sementara gambar sebelah kanan adalah output dari SPSS. LPP2 pada kedua output dihitung sebagai hasil dari ekspresi $LPP2=LPP*2$. Kedua gambar ini menunjukkan data nilai LPP2 yang sama. Karenanya, disimpulkan bahwa modul Compute Variable Statify berhasil menghitung nilai yang sesuai dengan perhitungan yang dilakukan SPSS.

A. Output Modul Transformasi Statify		B. Output SPSS	
Nama	Kode	Nama	Kode
Los_Angeles	CA	Los_Angeles	CA
Cook	-	Cook	-
Harris	-	Harris	-
San_Diego	CA	San_Diego	CA
Orange	CA	Orange	CA

Gambar 25. Validasi Output Recode into Same Variables

A. Output Modul Transformasi Statify		B. Output SPSS	
X15	X15_Recoded	X15	X15_Recoded
Sedang	2,00	Sedang	2,00
Tinggi	3,00	Tinggi	3,00
Sedang	2,00	Sedang	2,00
Sedang	2,00	Sedang	2,00
Tinggi	3,00	Tinggi	3,00

Gambar 26. Validasi Output Recode into Different Variables

Gambar 25 dan 26 menunjukkan perbandingan output pada modul Recode Variable. Gambar 25 menampilkan hasil Recode into Same Variables, sedangkan gambar 26 menampilkan hasil Recode into Different Variables. Pada kedua output, nilai variabel berhasil dikodekan ulang dan diperoleh hasil yang sama dengan hasil pada SPSS.

3.3.2. White-box Testing

Unit testing dilakukan pada ketiga modul dengan komponen testing seperti pada Tabel 1.

Tabel 1. Test Plan Summary Unit Testing

Test Item Name	Test Method
(1)	(2)
Chart Builder Components	22 test cases
Compute Variable Components	35 test cases
Recode Components	20 test cases
Recode into Same Variables Components	20 test cases
Recode into Different Variables Components	16 test cases
Data Processing Service	50 test cases

Unit testing berhasil dilakukan dengan tingkat keberhasilan 100%. Pengujian ini menghasilkan output seperti pada Gambar 27.

```

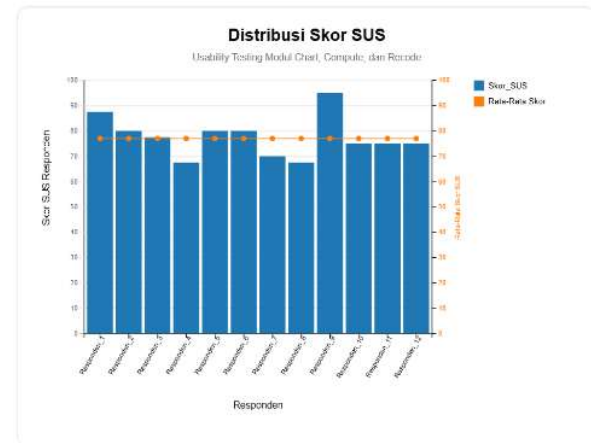
Test Suites: 1 passed, 1 total
Tests: 22 passed, 22 total
Snapshots: 0 total
Time: 4.346 s
Ran all test suites matching ChartBuilder.unit.test.ts.

> statify-backend@1.0.0 test
> jest --passWithNoTests ChartBuilder.unit.test.ts

No tests found, exiting with code 0
    
```

Gambar 27. Output Unit Testing untuk Chart Builder

3.3.3. SUS Usability Testing



Gambar 28. Visualisasi Distribusi Skor SUS per Responden

Berdasarkan hasil pengolahan kuesioner *System Usability Scale (SUS)*, diperoleh skor rata-rata sebesar 77,5, yang menurut interpretasi pendekatan kurva yang dikembangkan oleh Sauro (2012) termasuk ke dalam kategori B+ [14]. Nilai ini berada di atas nilai rata-rata SUS global, yaitu 68, menunjukkan bahwa aplikasi modul yang dikembangkan telah memenuhi kriteria *usability* yang baik, di mana mayoritas responden memberikan penilaian positif terhadap kemudahan penggunaan, konsistensi, dan pengalaman interaksi secara keseluruhan. Skor individu berkisar antara 67,5 hingga 87,5, tanpa adanya nilai yang berada pada kategori rendah, sehingga tidak ditemukan indikasi permasalahan signifikan terkait *usability*. Visualisasi distribusi skor SUS ditunjukkan pada Gambar 28.

3.3.4. SUMI Usability Testing

Hasil evaluasi SUMI menunjukkan bahwa skor Global yang diraih sebesar 64,42, di mana angka ini sudah berada 14,42 poin di atas standard SUMI, yang mengindikasikan persepsi pengguna terhadap aplikasi secara keseluruhan berada pada kategori baik. Dari lima dimensi evaluasi, *controlability* memperoleh skor tertinggi yaitu 61,75, diikuti oleh *helpfulness* sebesar 60,00 dan *efficiency* sebesar 60,42, yang menunjukkan bahwa pengguna menilai aspek pengendalian dan kemudahan mendapatkan bantuan relatif memadai. Dimensi *learnability* memiliki skor 55,33 menunjukkan kemudahan pembelajaran aplikasi,

sedangkan dimensi *affect* mendapat skor 59,08, terlihat bahwa adanya variasi positif dalam persepsi terhadap aspek ini.

4. PEMBAHASAN

Hasil penelitian menunjukkan bahwa modul Chart Builder, Compute Variable, dan Recode Variable dalam aplikasi Statify berhasil dikembangkan dengan pendekatan Feature Driven Development (FDD). Pada tahap implementasi, seluruh kebutuhan fungsional dan nonfungsional yang telah diidentifikasi pada tahap analisis kebutuhan berhasil terakomodasi dengan baik. Ketersediaan 43 jenis grafik dalam Chart Builder, dukungan 58 operator serta fungsi matematika dalam Compute Variable, dan 7 opsi rekoding variabel dalam Recode Variable mengindikasikan ketercapaian tujuan pengembangan.

Penerapan pendekatan FDD terbukti efektif dalam mengarahkan proses pembangunan sistem. Tahapan *build feature list* dan *plan by feature* mempermudah pemetaan kebutuhan ke dalam fitur yang terukur, sehingga jadwal pengembangan dapat dijalankan sesuai milestone. Selain itu, *design by feature* memberikan kejelasan desain teknis yang memudahkan integrasi antar modul. Temuan ini juga memperkuat hasil penelitian Setiaji (2020) [15] yang menyimpulkan bahwa FDD cocok digunakan dalam pengembangan aplikasi karena sistematis, efektif, serta mampu menjaga ketepatan waktu dalam penyelesaian proyek.

Hasil validasi menggunakan *black-box testing* menunjukkan kesesuaian output Statify dengan aplikasi pembandingan, yaitu SPSS. Kesamaan hasil pada pembuatan grafik, perhitungan variabel baru, maupun rekoding variabel menegaskan validitas fungsional dari sistem yang dibangun. Perbedaan hanya ditemukan pada aspek tampilan visual, di mana Statify menampilkan hasil dengan desain antarmuka yang lebih modern tetapi tetap konsisten dengan input yang diberikan. Reliabilitas sistem juga didukung dengan hasil *white-box testing*, di mana semua unit yang diuji menunjukkan keberhasilan 100%. Hal ini menandakan bahwa implementasi kode berjalan sesuai desain yang telah dirancang.

Dari sisi kegunaan, hasil pengujian *usability* dengan SUS menghasilkan skor rata-rata 77,5 yang termasuk dalam kategori B+, serta skor SUMI sebesar 64,42 yang berada di atas standar global. Kedua hasil ini memperlihatkan bahwa pengguna menilai aplikasi mudah dipelajari, konsisten, serta mampu memberikan pengalaman interaksi yang positif. Artinya, rancangan antarmuka yang intuitif dan ketersediaan log proses pada setiap modul memberikan kontribusi penting terhadap aspek *usability*.

Jika dibandingkan dengan aplikasi statistik komersial seperti SPSS, Statify lebih sesuai dengan kebutuhan di Polstat STIS. SPSS membutuhkan lisensi berbayar dan instalasi pada perangkat desktop. Sebaliknya, Statify dapat diakses langsung melalui peramban tanpa instalasi tambahan serta bebas lisensi, sehingga lebih fleksibel digunakan oleh mahasiswa maupun dosen. Hal ini sejalan dengan penelitian yang dilakukan oleh Mustafidah et al (2020) [16], yang juga mengembangkan Aplikasi Uji-T Satu Sampel Berbasis Web dengan prinsip kemudahan akses bagi pengguna.

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dipaparkan, dapat disimpulkan bahwa penelitian ini berhasil mengembangkan aplikasi Statify sebagai alat bantu analisis statistik berbasis web yang interaktif, mudah diakses, dan tidak memerlukan instalasi tambahan. Aplikasi ini dilengkapi tiga modul utama, yaitu Chart Builder dengan 43 jenis grafik yang mendukung personalisasi dan ekspor hasil visualisasi, Compute Variable yang memungkinkan pembuatan variabel baru berbasis ekspresi matematika menggunakan Math.js, serta Recode Variable yang mendukung pengkodean ulang variabel dengan dua opsi (*same* dan *different variable*). Hasil pengujian menunjukkan bahwa seluruh modul berfungsi sesuai rancangan, konsisten dengan perangkat lunak pembandingan (SPSS), dan memiliki tingkat *usability* yang baik, dibuktikan dengan skor SUS rata-rata 77,5 (kategori B+) serta skor SUMI global 64,42 yang berada di atas standar.

DAFTAR PUSTAKA

- [1] Badan Pusat Statistik, *Peraturan Badan Pusat Statistik Nomor 87 Tahun 2018*. Indonesia: PPID BPS, 2018. Accessed: Nov. 07, 2024. [Online]. Available: <https://ppid.bps.go.id/app/konten/0000/Peraturan.html>
- [2] Republik Indonesia, *UU No.12 Tahun 2012*. Indonesia: peraturan.bpk.go.id, 2012. Accessed: Nov. 09, 2024. [Online]. Available: <https://peraturan.bpk.go.id/Details/39063/uu-no-12-tahun-2012>
- [3] R. A. Muenchen, "The Popularity of Data Science Software." Accessed: Jun. 04, 2025. [Online]. Available: <https://r4stats.com/articles/popularity/>
- [4] Posit Software, "RStudio IDE The most trusted IDE for open source data science." Accessed: Nov. 08, 2024. [Online]. Available: <https://posit.co/products/open-source/rstudio/>
- [5] O. Rosquist and J. Luttu, "Analysis of the performance difference between server-side and client-side rendering for data visualization in real-time using d3.js," Bachelor thesis, KTH Royal Institute of Technology, Stockholm, 2017. Accessed: Nov. 24, 2024. [Online]. Available: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1107724&dswid=854>
- [6] Safwane Benbba, "COMPARISON OF D3.JS AND CHART.JS AS VISUALISATION TOOLS," Bachelor's thesis, Tampere University, 2021. Accessed: Aug. 01, 2025. [Online]. Available: <https://urn.fi/URN:NBN:fi:tuni-202104273941>
- [7] V. Dogadina and A. Voronin, "Comparative Analysis of Data Visualization Tools," *International Journal of Computing, Programming and Database Management*, vol. 5, no. 1, pp. 49–51, 2024, doi: [10.33545/27076636.2024.v5.i1.a.93](https://doi.org/10.33545/27076636.2024.v5.i1.a.93).
- [8] Apache Software Foundation, "Apache ECharts Documentation." Accessed: Aug. 12, 2025. [Online]. Available: <https://echarts.apache.org/en/option.html>
- [9] mathjs.org, "math.js." Accessed: Feb. 10, 2025. [Online]. Available: <https://mathjs.org/index.html>

- [10] Mestika Zed, *Metode Penelitian Kepustakaan*. Jakarta: Yayasan Pustaka Obor Indonesia, 2003.
- [11] A. Fathoni, *Metode Penelitian & Teknik Penyusunan Skripsi*, 2nd ed. Jakarta: Rineka Cipta, 2011.
- [12] S. R. Palmer and J. M. Felsing, *A Practical Guide to Feature-Driven Development*. Upper Saddle River: Prentice Hall PTR, 2002. [Online]. Available: www.togethersoft.com
- [13] M. Rychl'ý and P. Tichá, "A Tool for Supporting Feature-Driven Development," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5082 LNCS, p. 196207, 2008, doi: [10.1007/978-3-540-85279-7_16](https://doi.org/10.1007/978-3-540-85279-7_16).
- [14] J. Sauro and J. R. Lewis, "Correlations among prototypical usability metrics: evidence for the construct of usability," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York: Association for Computing Machinery, 2009, pp. 1609–1618. doi: [10.1145/1518701.1518947](https://doi.org/10.1145/1518701.1518947).
- [15] O. A. Santosa and H. Setiaji, "Pengembangan Aplikasi Ecommerce Dengan Metode Feature Driven Development," *AUTOMATA*, vol. 1, no. 2, pp. 169–172, Jun. 2020, Accessed: Nov. 25, 2024. [Online]. Available: <https://journal.uii.ac.id/AUTOMATA/article/view/15738>
- [16] H. Mustafidah, A. Imantoyo, and S. Suwarsito, "Mustafidah T test," *Jurnal Informatika*, vol. 8, no. 2, pp. 245–251, Nov. 2020, doi: [10.30595/juita.v8i2.8786](https://doi.org/10.30595/juita.v8i2.8786).

BIODATA PENULIS



Muhamad Iqbal Putra Pratama
Penulis merupakan mahasiswa Program Studi Komputasi Statistik, Politeknik Statistika STIS, dengan peminatan Sistem Informasi, dan lulus pada tahun 2025.



Erna Nurmawati
Penulis memperoleh gelar Sarjana Sains Terapan (S.S.T.) di Sekolah Tinggi Ilmu Statistik Jakarta pada tahun 2006 dengan peminatan Komputasi Statistik. Ia melanjutkan studi di Institut Teknologi Bandung dan meraih gelar Magister Teknik (M.T.) pada tahun 2014 dengan fokus pada bidang Teknologi Informasi. Minat penelitiannya meliputi komputasi statistik, *e-government*, *machine learning*, dan teknologi informasi. Saat ini, ia bekerja sebagai dosen di Politeknik Statistika STIS Jakarta. Ia dapat dihubungi melalui email: erna.nurmawati@stis.ac.id.