

Pengembangan Statify: Aplikasi Analisis Statistik

Modul Analisis *Time Series*

Guswana Adventus (222112082, 4SI1)

Dosen Pembimbing: Ibnu Santoso, S.S.T, M.T.

Ringkasan— Analisis Runtun Waktu (*Time Series*) merupakan salah satu mata kuliah yang diajarkan di lingkungan Politeknik Statistika STIS. Aplikasi Eviews, SPSS, dan R menjadi aplikasi yang sering digunakan dalam pembelajaran Analisis Runtun Waktu. Namun, pemanfaatan aplikasi tersebut belum optimal karena metode analisis yang tersedia belum terintegrasi dengan alur materi Analisis Runtun Waktu. Selain itu, keterbatasan fitur eksplorasi awal pada SPSS dan ketiadaan otomatisasi visualisasi grafik pada Eviews, tentunya cukup menyulitkan mahasiswa dalam melakukan tahapan analisis secara berkesinambungan dan terstruktur. Dengan demikian, diperlukan aplikasi statistik yang mampu menyelesaikan permasalahan tersebut. Sebagai solusi maka dikembangkan Statify Modul Analisis *Time Series*, aplikasi berbasis web yang responsif dengan pendekatan optimasi *client-side computing*. Aplikasi ini dikembangkan menggunakan *framework* next.js dan teknologi WebAssembly. Adapun, Statify dikembangkan menggunakan metode pengembangan *Feature-Driven Development* (FDD). Selanjutnya aplikasi ini dievaluasi menggunakan metode *white-box testing* dengan teknik *basis path* dan diperoleh hasil bahwa algoritma penghitungan statistik Statify sudah berjalan dengan benar. Selain itu, evaluasi dilanjutkan dengan *usability testing* yang dilakukan oleh pengguna dengan bantuan instrumen penilaian kuesioner *Software Usability Measurement Inventory* (SUMI). Adapun hasil diperoleh dari evaluasi tersebut adalah Statify Modul Analisis *Time Series* memperoleh nilai di atas 50 poin untuk setiap aspek penilaian yang berarti Statify memiliki usability yang cukup baik.

Kata Kunci— *Time Series*, FDD, *White-Box Testing*, *Usability Testing*, SUMI.

I. LATAR BELAKANG

Politeknik Statistika STIS merupakan salah satu perguruan tinggi kedinasan yang berada di bawah naungan Badan Pusat Statistik (BPS). Institusi ini menyelenggarakan pendidikan vokasi yang fokus pada ilmu statistika terapan dan komputasi statistik [1]. Oleh karena itu pembelajaran metode statistik dijadikan sebagai landasan utama dalam proses pendidikan di perguruan tinggi tersebut. Proses pembelajaran tidak hanya menekankan pada penguasaan teori secara mendalam, tetapi juga dilengkapi dengan pengembangan keterampilan praktis yang memadai, sehingga lulusan mampu mengaplikasikan ilmu statistik secara efektif dalam berbagai konteks profesional.

Analisis Runtun Waktu (*Time Series*) menjadi salah satu mata kuliah yang diajarkan dalam proses pembelajaran di Politeknik Statistika STIS. Materi ini diajarkan untuk memberikan pemahaman tentang metode statistik yang memanfaatkan data historis dalam meramalkan fenomena di masa depan [2]. Melalui pembelajaran ini, mahasiswa dilatih

untuk mengidentifikasi lintasan dinamis suatu deret data dan melakukan ekstrapolasi komponen-komponen deret ke masa depan guna menghasilkan peramalan yang lebih akurat [3]. Selain sebagai alat analisis, hasil pembelajaran analisis runtun waktu juga difokuskan agar mahasiswa mampu menggunakan hasil analisis tersebut sebagai dasar pertimbangan dalam penyusunan strategi, kebijakan, dan perencanaan bisnis yang efektif dan efisien [4]. Dengan demikian, penguasaan konsep dan teknik analisis runtun waktu menjadi salah satu kompetensi utama yang dikembangkan dalam kurikulum di Politeknik Statistika STIS guna mempersiapkan lulusan yang siap menghadapi tantangan di era saat ini.

Dalam proses pembelajaran Analisis Runtun Waktu, pemanfaatan aplikasi statistik memegang peranan penting dalam mendukung pemahaman konsep dan implementasi teknik analisis. Di Politeknik Statistika STIS, aplikasi Eviews, SPSS, dan R menjadi aplikasi yang sering digunakan. Namun, pemanfaatan aplikasi-aplikasi tersebut belum sepenuhnya optimal karena belum terintegrasi metode Analisis Runtun Waktu yang sesuai dengan alur materi dalam Rencana Pembelajaran Semester (RPS), selain juga terkendala oleh masalah akses serta kebutuhan kemampuan pemrograman.

Salah satu masalah proses pembelajaran Analisis Runtun Waktu adalah terbatasnya interaksi pembelajaran akibat kurang terintegrasinya tahapan metode Analisis Runtun Waktu dalam aplikasi statistik yang digunakan. Sebagai contoh, pada aplikasi SPSS tidak terdapat fitur eksplorasi awal data *time series* yaitu metode *smoothing* yang cukup membingungkan mahasiswa. Selain itu, pada aplikasi ini juga tidak terdapat tambahan visualisasi grafik hasil metode *decomposition* yang dalam penerapannya hal ini penting bagi mahasiswa untuk memahami struktur data *time series*. Begitu juga pada aplikasi Eviews, proses eksplorasi awal untuk memvisualisasikan data *time series* masih terhambat karena belum adanya otomatisasi pembuatan grafik pada aplikasi tersebut. Kondisi ini tentunya membuat mahasiswa kesulitan dalam menggunakan aplikasi untuk melakukan tahapan Analisis Runtun Waktu secara berkesinambungan dan terstruktur.

Oleh karena itu, diperlukan pengembangan aplikasi statistik yang mampu mendukung proses pembelajaran Analisis Runtun Waktu secara terpadu, mulai dari tahap eksplorasi awal data hingga analisis lanjutan. Aplikasi ini diharapkan tidak hanya menyediakan fitur eksplorasi awal data dan otomatisasi visualisasi grafik, tetapi juga mengintegrasikan setiap tahapan analisis secara sistematis. Dengan demikian, proses pembelajaran dapat berlangsung lebih efektif, terstruktur, dan

selaras dengan Rencana Pembelajaran Semester (RPS) mata kuliah Analisis Runtun Waktu di Politeknik Statistika STIS.

Maka dari itu, sebagai solusi terhadap berbagai permasalahan tersebut adalah pengembangan Aplikasi Statify. Statify adalah sebuah aplikasi statistik berbasis web yang diharapkan dapat menjadi solusi yang efektif. Dengan menggunakan teknologi berbasis web, aplikasi ini mengalihkan proses komputasi ke server dengan optimasi khusus, yaitu *client-side computing*, sehingga pengguna dapat menikmati proses penghitungan data yang lebih cepat dari sisi klien. Selain itu, Statify juga dirancang dengan mengintegrasikan berbagai metode analisis statistik dalam satu platform yang mudah digunakan, sehingga aplikasi ini diharapkan mampu menjadi media pembelajaran yang memfasilitasi pemahaman konsep, eksperimen metode, dan keterampilan analisis statistik secara menyeluruh bagi mahasiswa Politeknik Statistika STIS.

Salah satu fokus utama pengembangan Statify adalah penyediaan Modul Analisis *Time Series* yang disesuaikan dengan materi-materi yang tercantum dalam RPS Politeknik Statistika STIS T.A. 2024/2025 mata kuliah Analisis Runtun Waktu, dapat dilihat pada Lampiran 1. Modul ini mencakup pengembangan fitur-fitur penting seperti *smoothing*, *decomposition*, *autocorrelation*, *unit root test*, dan *box-jenkins model* (ARIMA). Metode-metode tersebut memegang peranan krusial dalam membantu pengguna meneliti dan menganalisis data runtun waktu secara efektif dan efisien, sehingga mendukung proses pembelajaran di lingkungan Politeknik Statistika STIS.

II. TUJUAN PENELITIAN

Berdasarkan identifikasi masalah sebelumnya maka penelitian ini bertujuan untuk mengembangkan Statify, sebuah aplikasi statistik yang menyediakan fitur analisis *time series* yang mengakomodasi metode yang sesuai dengan RPS Mata Kuliah Analisis Runtun Waktu di Lingkungan Politeknik Statistika STIS. Dengan demikian untuk mencapai tujuan utama tersebut dijabarkan beberapa tujuan khusus pada penelitian ini, yaitu:

1. Mengembangkan Statify Modul Analisis *Time Series* dengan tampilan antarmuka yang efisien, membantu pengguna, serta mudah digunakan dan dipelajari.
2. Mengembangkan setiap fitur Statify Modul Analisis *Time Series* dengan proses penghitungan dan hasil statistik yang valid.

III. BATASAN PENELITIAN

pembatasan penelitian diperlukan untuk memberikan ruang lingkup yang jelas dan memastikan penelitian ini berjalan dengan terarah dalam membangun Modul Analisis *Time Series* pada Aplikasi Statify. Dengan demikian, dijabarkan beberapa batasan masalah dalam penelitian ini sebagai berikut:

1. Menu *time series* yang dibangun masih terbatas pada analisis *time series* univariat, diantaranya *smoothing*, *decomposition*, *autocorrelation*, *unit root test*, dan *box-jenkins model*.
2. Tampilan yang dibangun berfokus pada tampilan modal dan output statistik menu *time series* saja.

3. Penulisan kode program *client-side computing* aplikasi Statify modul analisis *time series* dibuat menggunakan bahasa pemrograman Rust dengan tujuan memberikan proses komputasi yang ringan pada aplikasi web sisi klien. Akan tetapi bahasa ini merupakan bahasa pemrograman masih baru dan dirilis pada tahun 2015 sehingga ekosistem serta package yang tersedia pada bahasa ini masih terbatas sehingga kondisi ini menyebabkan terbatasnya metode statistik analisis *time series* yang dikembangkan pada penelitian ini.

IV. PENELITIAN TERKAIT

Berdasarkan permasalahan yang telah dibahas sebelumnya. Penelitian ini menggunakan beberapa penelitian terdahulu sebagai referensi yang mendukung proses pengembangan Statify Modul Analisis *Time Series*. Adapun, penelitian terdahulu yang digunakan memiliki keterkaitan dalam hal kesamaan aplikasi yang dikembangkan, metode pengembangan, dan teknologi yang digunakan.

Penelitian pertama [5], mengembangkan aplikasi statistik analisis Uji-*T* dalam bentuk web interaktif. Selain itu, terdapat juga penelitian kedua [6], mengembangkan aplikasi statistik analisis regresi berbasis web yang diberi nama StaWEB. Kedua penelitian ini memiliki kesamaan dalam hal jenis aplikasi yang dibangun yaitu aplikasi statistik berbasis web. Namun, kedua penelitian tersebut dan penelitian ini memiliki perbedaan pada metode pengembangan yang digunakan dan bahasa pemrograman yang digunakan. Pada penelitian pertama metode pengembangan yang digunakan adalah pendekatan *System Development Life Cycle* (SDLC) tanpa penjelasan lebih lanjut metode pengembangan apa yang digunakan dan penelitian kedua metode pengembangan yang digunakan adalah metode *waterfall*. Lalu kedua penelitian tersebut sama-sama menggunakan framework R-Shiny dengan bahasa pemrograman R. Sedangkan penelitian ini melakukan pengembangan aplikasi Statify menggunakan metode *Feature-Driven Development* (FDD). Selain itu, penelitian ini mengembangkan Statify menggunakan framework Next.js dengan bahasa pemrograman JavaScript dan TypeScript.

Selanjutnya penelitian ketiga memiliki keterkaitan dalam penggunaan metode pengembangan yang digunakan. Penelitian ketiga [7], mengembangkan sistem bimbingan konseling menggunakan metode FDD. Melalui penelitian ini diperoleh informasi bahwa penerapan metode FDD dalam pengembangan sistem memungkinkan pengembangan sistem menjadi lebih terstruktur dan efisien walaupun pengembangannya dibagi berdasarkan fitur-fitur yang hendak dibangun. Dengan demikian, metode ini akan mempermudah proses integrasi fitur-fitur yang berhasil dibangun.

Penelitian keempat memiliki keterkaitan dengan metode evaluasi *white-box testing*. Penelitian keempat [8], menguji *website Room* dengan tujuan memeriksa keberadaan *error* pada kode program. Selain itu, penelitian tersebut memiliki kesamaan pada teknik yang digunakan yaitu teknik basis path. Dengan menggunakan teknik ini peneliti dapat memperoleh cakupan *test case* yang lebih menyeluruh sehingga *error* atau kegagalan proses kerja algoritma dapat ditemukan lebih awal sebelum dilanjutkan ke tahapan *black-box testing*.



Gambar 4. Teknologi yang digunakan Statify Modul Analisis *Time Series*

B. Build a Features List

Berdasarkan kebutuhan umum yang dianalisis sebelumnya, maka pada tahapan ini peneliti melakukan perincian fitur dari setiap kegiatan kebutuhan umum Statify Modul Analisis *Time Series* yang disajikan pada Tabel I.

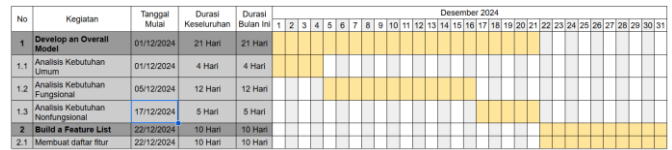
TABEL I
DAFTAR FITUR STATIFY MODUL ANALISIS *TIME SERIES*

No.	Major Feature Sets	Feature Sets
(1)	(2)	(3)
1	Smoothing	Akses Menu <i>Smoothing</i>
2		Pengaturan Variabel
3		Pengaturan Menu <i>Smoothing</i>
4		Pengaturan Waktu
5		Histori Pengaturan
6		Output <i>Smoothing</i>
7	Decomposition	Akses Menu <i>Decomposition</i>
8		Pengaturan Variabel
9		Pengaturan Menu <i>Decomposition</i>
10		Pengaturan Waktu
11		Histori Pengaturan
12		Output <i>Decomposition</i>
13	Autocorrelation	Akses Menu <i>Autocorrelation</i>
14		Pengaturan Variabel
15		Pengaturan Menu <i>Autocorrelation</i>
16		Pengaturan Waktu
17		Histori Pengaturan
18		Output <i>Autocorrelation</i>
19	Unit Root Test	Akses Menu <i>Unit Root Test</i>
20		Pengaturan Variabel
21		Pengaturan Menu <i>Unit Root Test</i>
22		Histori Pengaturan
23		Output <i>Unit Root Test</i>
24	Box-Jenkins Model	Akses Menu <i>Box-Jenkins Model</i>
25		Pengaturan Variabel
26		Pengaturan Menu <i>Box-Jenkins Model</i>
27		Pengaturan Waktu
28		Histori Pengaturan
29		Output <i>Box-Jenkins Model</i>

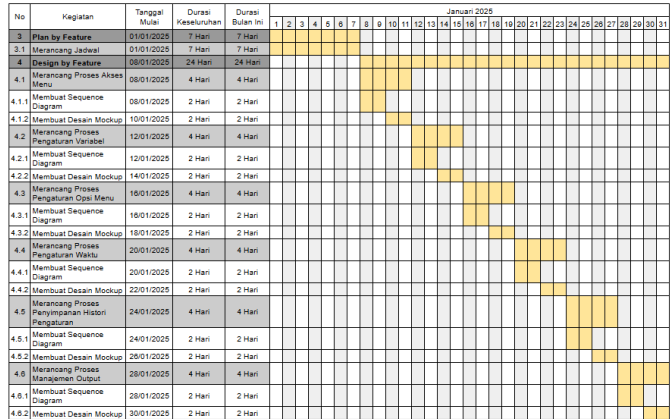
C. Plan By Feature

Pada tahapan ini, peneliti memastikan setiap tahapan pengembangan Aplikasi Statify Modul Analisis *Time Series* dapat berjalan secara sistematis melakukan perancangan jadwal yang jelas di setiap tahapannya. Perancangan ini dituangkan dalam bentuk *Gantt Chart* yang di dalamnya mencakup setiap

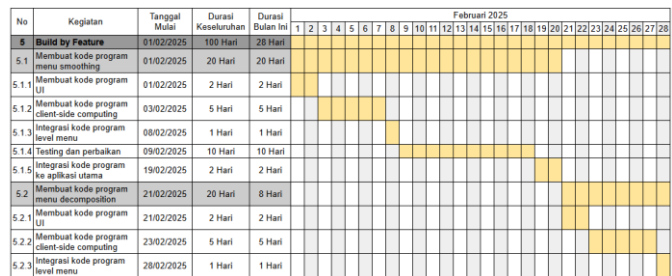
tahapan yang didasarkan dengan tahapan metode pengembangan FDD yang ditampilkan pada Gambar 5 sampai 10.



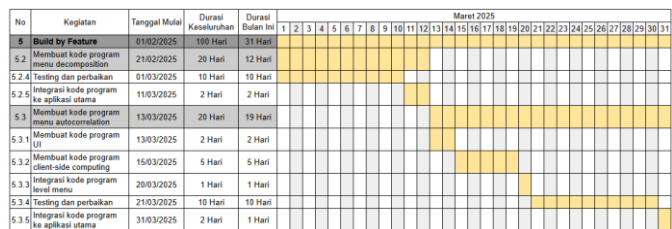
Gambar 5. *Gantt Chart* Statify Modul Analisis *Time Series* Desember 2024



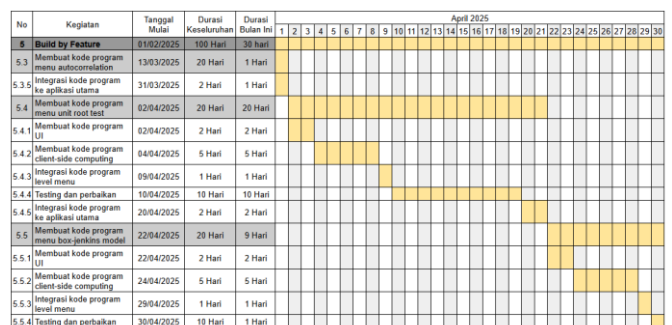
Gambar 6. *Gantt Chart* Statify Modul Analisis *Time Series* Januari 2025



Gambar 7. *Gantt Chart* Statify Modul Analisis *Time Series* Februari 2025



Gambar 8. *Gantt Chart* Statify Modul Analisis *Time Series* Maret 2025



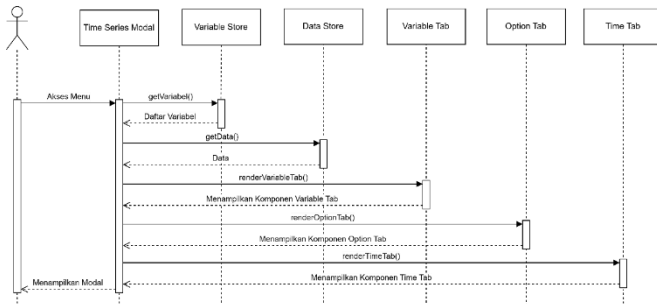
Gambar 9. *Gantt Chart* Statify Modul Analisis *Time Series* April 2025

No	Kegiatan	Tanggal Mulai	Durasi Keseluruhan	Durasi Bulan Ini	Mai 2025																															
5	Build by Feature	01/02/2025	100 Hari	11 Hari																																
5.5	Membuat kode program menu box jendela modal	22/04/2025	20 Hari	11 Hari																																
5.5.4	Testing dan perbaikan	30/04/2025	10 Hari	9 Hari																																
5.5.5	Integrasi kode program ke aplikasi utama	10/05/2025	2 Hari	2 Hari																																

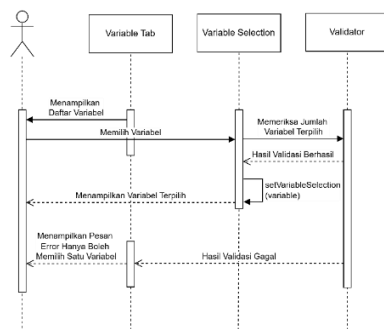
Gambar 10. *Gantt Chart* Statify Modul Analisis *Time Series* Mei 2025

D. Design By Feature

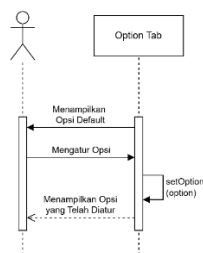
Berdasarkan hasil perancangan plan by feature maka peneliti membuat sequence diagram untuk setiap feature sets yang telah terdaftar. Adapun feature sets yang ada pada setiap menu (major feature sets) memiliki cara kerja yang sama maka dibentuk enam sequence diagram yang ditampilkan pada Gambar 11 sampai 16 sebagai berikut.



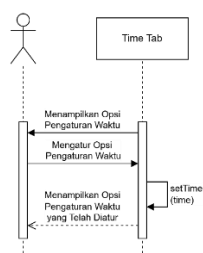
Gambar 11. *Sequence Diagram* Feature Sets Mengakses Menu *Time Series*



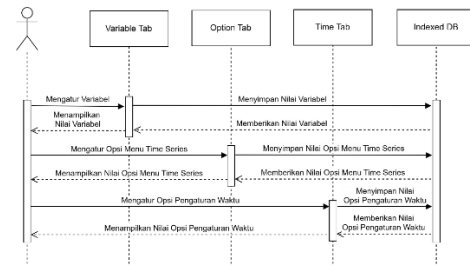
Gambar 12. *Sequence Diagram* Feature Sets Mengatur Variabel



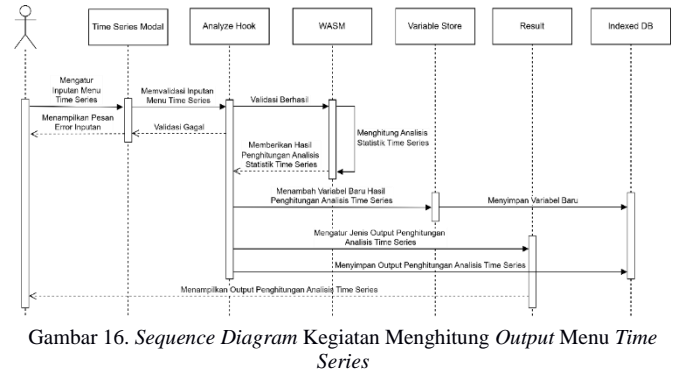
Gambar 13. *Sequence Diagram* Feature Sets Mengatur Opsi Menu *Time Series*



Gambar 14. *Sequence Diagram* Feature Sets Mengatur Opsi Menu Waktu



Gambar 15. *Sequence Diagram* Kegiatan Menyimpan Histori Pengaturan



Gambar 16. *Sequence Diagram* Kegiatan Menghitung Output Menu *Time Series*

Selain itu, pada tahapan ini peneliti juga merancang tampilan antarmuka menu *time series*. Berikut beberapa hasil rancangan tampilan antarmuka modal menu *time series* yang ditampilkan pada Gambar 17 sampai 19.

Variable Tab With Save

Time Series Modal

Variables Option Tab Time

Available Variables

variable 1

variable 2

variable 3

variable 4

Selected Variable :

variable 5

☐ Save Forecasting as Variable

OK Reset Cancel Help

Gambar 16. *Mockup* Variable Tab Menu *Time Series*

Time Tab

Time Series Modal

Variables Option Tab Time

Time Option:

time spesification : week-work days(5)

year : 2025

month : 1

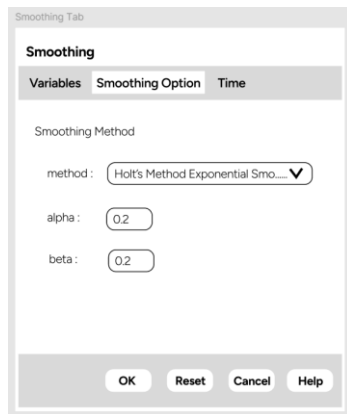
day : 1

periodicity: 5

start time : Wednesday, 1 Jan 2025

OK Reset Cancel Help

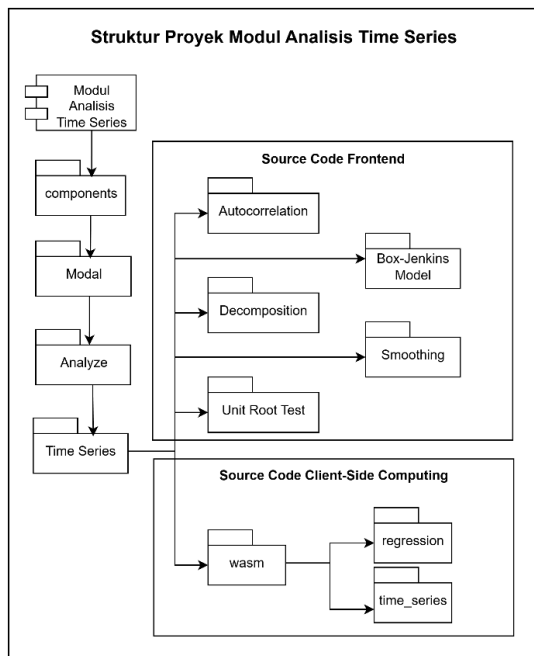
Gambar 16. *Mockup* Time Tab Menu *Time Series*



Gambar 16. Mockup Option Tab Menu Smoothing

E. Build By Feature

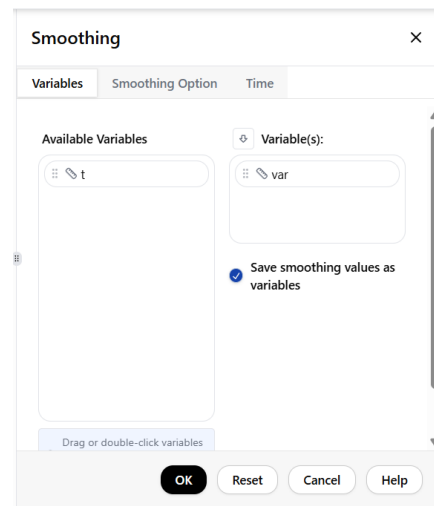
Pada tahapan ini, peneliti melakukan implementasi berdasarkan hasil analisis dan perancangan pada tahapan-tahapan sebelumnya. Adapun Gambar 17 menampilkan struktur kode program Statify Modul Analisis *Time Series*.



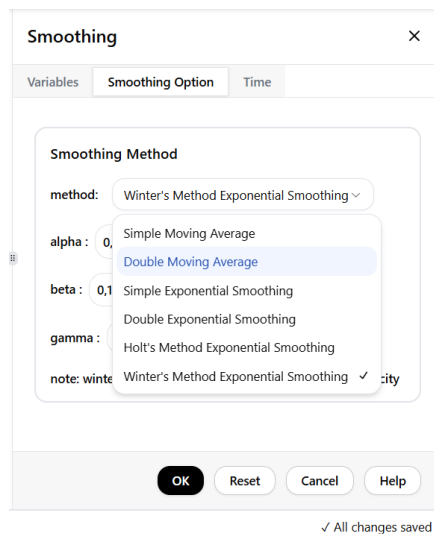
Gambar 17. Struktur Kode Program Statify Modul Analisis *Time Series*

Berdasarkan gambar tersebut kode program Statify Modul Analisis *Time Series* terbagi atas dua struktur utama yaitu kode program *frontend* dan kode program *client-side computing*. Struktur *frontend* terbagi atas lima folder yang masing-masing berfungsi untuk mengatur menu *time series* dan logika respon antarmuka menu *time series*. Sedangkan struktur *client-side computing* berfungsi mengatur algoritma proses penghitungan analisis *time series*.

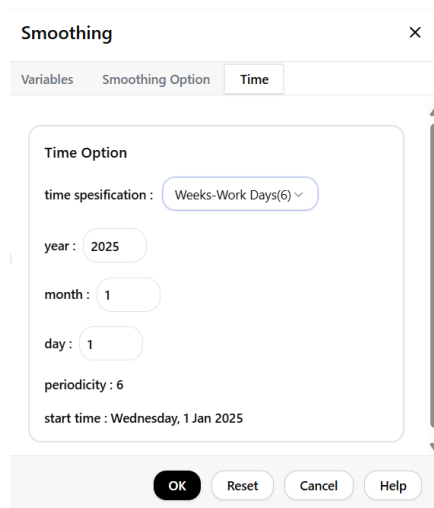
Selain itu juga ditampilkan beberapa hasil implementasi antarmuka Statify Modul Analisis *Time Series*. Gambar 18 sampai 20 menampilkan hasil implementasi tampilan antarmuka menu *smoothing* yang terdiri atas tiga buah tab.



Gambar 18. Variable Tab Menu Smoothing



Gambar 19. Smoothing Option Tab Menu Smoothing



Gambar 20. Time Tab Menu Smoothing

F. White-Box Testing

Pada tahapan ini, peneliti menguji kompleksitas dan memeriksa kesesuaian algoritma kode program yang telah diterapkan. Teknik yang digunakan oleh peneliti pada *white-box testing* ini adalah teknik *basis path*. Teknik *basis path* digunakan karena teknik tersebut mampu menemukan *test cases* yang lebih luas dan menyeluruh dengan mencoba berbagai kemungkinan jalur kode program. Adapun teknik ini terdiri atas tiga tahapan utama, yaitu pembentukan diagram alur (*flow graph notation*), penghitungan tingkat kompleksitas (*cyclomatic complexity*), dan pendaftaran kasus uji (*deriving test cases*). Berikut ditampilkan tahapan white-box testing pada salah satu menu *time series* yaitu menu *decomposition*.

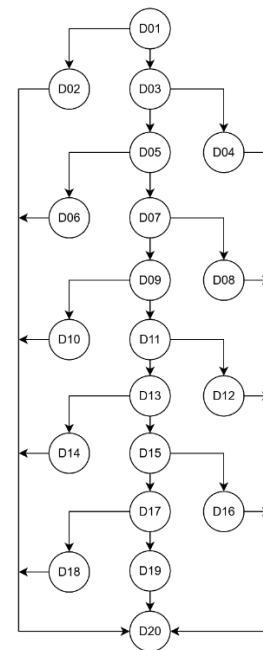
1. Flow Graph Notation

Pada tahapan ini, peneliti membagi kode program level integrasi ke dalam kumpulan simpul. Kumpulan simpul tersebut diambil dari *statement* penting kode program khususnya *statement* kondisional. Adapun pada menu *decomposition* diperoleh sebanyak dua puluh buah simpul yang diberi kode D01 sampai D20. Berikut ditampilkan daftar simpul menu *decomposition* yang disajikan pada Tabel II.

TABEL II
DAFTAR SIMPUL MENU *DECOMPOSITION*

Kode Simpul	Nama Simpul
(1)	(2)
D01	Menjalankan menu <i>decomposition</i>
D02	Variabel terpilih kosong
D03	Variabel terpilih satu
D04	Variabel terpilih bukan numerik
D05	Variabel terpilih numerik
D06	Data kosong
D07	Data berisi
D08	Belum mengatur konfigurasi waktu
D09	Sudah mengatur konfigurasi waktu
D10	Spesifikasi waktu tidak memiliki panjang periode
D11	Spesifikasi waktu memiliki panjang periode
D12	Observasi data kurang dari kelipatan 4 panjang periode waktu
D13	Observasi data lebih dari sama dengan kelipatan 4 panjang periode waktu
D14	Observasi data tidak sama dengan kelipatan panjang periode waktu
D15	Observasi data sama dengan kelipatan panjang periode waktu
D16	Metode <i>Additive</i>
D17	Metode <i>Multiplicative</i>
D18	Tren linear
D19	Tren eksponensial
D20	Hasil

Selanjutnya setiap simpul yang terbentuk disusun berdasarkan alur kode program yang ada untuk mengetahui apakah algoritma kode program yang dibuat sesuai dengan rancangan pengembangan. Adapun simpul yang disusun tersebut disajikan dalam bentuk *flow graph* yang ditampilkan pada Gambar 21.



Gambar 20. *Time Tab Menu Smoothing*

2. Cyclomatic Complexity

Pada tahapan ini, peneliti menghitung kompleksitas kode program menggunakan rumus $V(G) = E - N + 2$. . Notasi E melambangkan jumlah garis (*edge*) yang terdapat pada *flow graph* dan notasi N melambangkan jumlah simpul (*node*) yang terdapat pada *flow graph*. Pada menu *decomposition* jumlah garis dan simpul yang diperoleh pada *flow graph* masing-masing berjumlah 28 dan 20 sehingga diperoleh nilai kompleksitas sebesar 10.

3. Deriving Test Cases

Berdasarkan kedua tahapan sebelumnya, jumlah *test cases* yang dibentuk oleh peneliti pada menu *decomposition* berjumlah sepuluh *test cases*. Selanjutnya, setiap *test cases* akan diuji dengan membandingkan hasil ekspektasi dan hasil aktual untuk menentukan keberhasilan suatu *test case*. Berikut Tabel III menampilkan hasil pengujian seluruh *test cases* menu *decomposition*.

TABEL III
DAFTAR *TEST CASES* MENU *DECOMPOSITION*

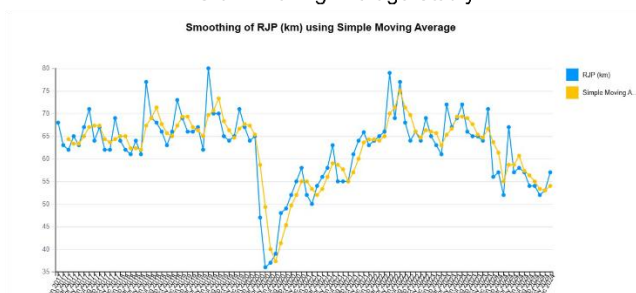
Kode Test Case	Hasil Ekspektasi	Hasil Aktual	Hasil Pengujian
(1)	(3)	(4)	(5)
TCD01	Menampilkan pesan <i>error</i>	Berhasil menampilkan pesan <i>error</i>	Berhasil
TCD02	Menampilkan pesan <i>error</i>	Berhasil menampilkan pesan <i>error</i>	Berhasil
TCD03	Menampilkan pesan <i>error</i>	Berhasil menampilkan pesan <i>error</i>	Berhasil
TCD04	Menampilkan pesan <i>error</i>	Berhasil menampilkan pesan <i>error</i>	Berhasil
TCD05	Menampilkan pesan <i>error</i>	Berhasil menampilkan pesan <i>error</i>	Berhasil
TCD06	Menampilkan pesan <i>error</i>	Berhasil menampilkan pesan <i>error</i>	Berhasil
TCD07	Menampilkan pesan <i>error</i>	Berhasil menampilkan pesan <i>error</i>	Berhasil

TCD08	Menghasilkan <i>output</i> statistik	Berhasil menampilkan <i>output</i> statistik	Berhasil
TCD09	Menghasilkan <i>output</i> statistik	Berhasil menampilkan <i>output</i> statistik	Berhasil
TCD10	Menghasilkan <i>output</i> statistik	Berhasil menampilkan <i>output</i> statistik	Berhasil

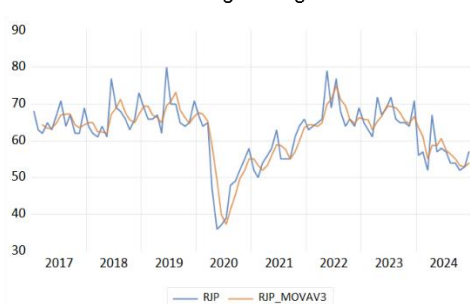
G. Validasi Output Statistik

Pada tahapan ini, peneliti melakukan perbandingan *output* statistik antara Statify dan Eviews dengan tujuan untuk memvalidasi *output* statistik yang dihasilkan. Aplikasi Eviews dijadikan dasar perbandingan dalam penelitian ini karena aplikasi tersebut merupakan aplikasi utama yang digunakan dalam praktikum mata kuliah Analisis Runtun Waktu. Berikut Gambar 22 sampai 26 menampilkan salah satu validasi *output* dari masing-masing menu *time series*.

Grafik Moving Average Statify

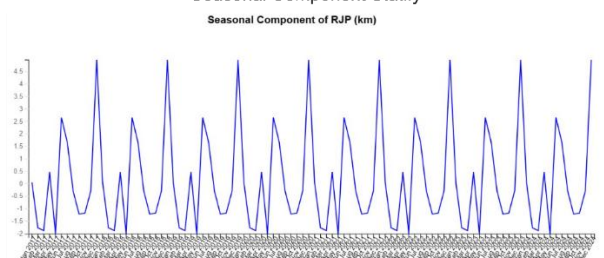


Grafik Moving Average Eviews



Gambar 22. Perbandingan Grafik Moving Average Menu Smoothing

Seasonal Component Statify



Seasonal Component Eviews



Gambar 23. Perbandingan Grafik Seasonal Component Menu Decomposition

Autocorrelation Statify

	ACF	SE	Ljung-Box	df	p-value	PACF	SE
1	0.727	0.102	52.379	1	0.000	0.727	0.102
2	0.626	0.146	91.557	2	0.000	0.205	0.102
3	0.460	0.172	112.986	3	0.000	-0.112	0.102
4	0.373	0.184	127.235	4	0.000	0.021	0.102
5	0.352	0.192	140.051	5	0.000	0.153	0.102
6	0.348	0.199	152.704	6	0.000	0.093	0.102
7	0.259	0.205	159.815	7	0.000	-0.172	0.102
8	0.178	0.208	163.199	8	0.000	-0.105	0.102
9	0.125	0.210	164.880	9	0.000	0.069	0.102
10	-0.002	0.211	164.880	10	0.000	-0.187	0.102
11	0.002	0.211	164.881	11	0.000	0.041	0.102
12	0.036	0.211	165.024	12	0.000	0.186	0.102
13	-0.073	0.211	165.630	13	0.000	-0.308	0.102
14	-0.073	0.211	166.240	14	0.000	-0.002	0.102
15	-0.153	0.211	168.973	15	0.000	0.018	0.102
16	-0.192	0.212	173.297	16	0.000	-0.038	0.102
17	-0.214	0.214	178.762	17	0.000	-0.104	0.102
18	-0.230	0.216	185.165	18	0.000	-0.079	0.102
19	-0.281	0.219	194.812	19	0.000	-0.002	0.102
20	-0.339	0.223	209.044	20	0.000	-0.231	0.102

Autocorrelation Eviews

	AC	PAC	Q-Stat	Prob
1	0.727	0.727	52.379	0.000
2	0.626	0.205	91.557	0.000
3	0.460	-0.112	112.99	0.000
4	0.373	0.021	127.23	0.000
5	0.352	0.153	140.05	0.000
6	0.348	0.093	152.70	0.000
7	0.259	-0.172	159.81	0.000
8	0.178	-0.105	163.20	0.000
9	0.125	0.069	164.88	0.000
10	-0.002	-0.187	164.88	0.000
11	0.002	0.041	164.88	0.000
12	0.036	0.186	165.02	0.000
13	-0.073	-0.308	165.63	0.000
14	-0.073	-0.002	166.24	0.000
15	-0.153	0.018	168.97	0.000
16	-0.192	-0.038	173.30	0.000
17	-0.214	-0.104	178.76	0.000
18	-0.230	-0.079	185.17	0.000
19	-0.281	-0.002	194.81	0.000
20	-0.339	-0.231	209.04	0.000

Gambar 24. Perbandingan Tabel ACF dan PACF menu Autocorrelation

Dickey-Fuller Test Statify

Dickey-Fuller Test Statistic		
	t-statistic	p-value
Dickey-Fuller Test	-3.806	0.003
Critical Value	1%	-3.501
	5%	-2.892
	10%	-2.583

Dickey-Fuller Test Eviews

Augmented Dickey-Fuller Unit Root Test on RJP		
Null Hypothesis: RJP has a unit root		
Exogenous: Constant		
Lag Length: 0 (Fixed)		
	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-3.805980	0.0040
Test critical values:		
1% level	-3.500669	
5% level	-2.892200	
10% level	-2.583192	

Gambar 25. Perbandingan Uji Dickey-Fuller Menu Unit Root Test

Output Model ARIMA Statify

Coefficients Test for ARIMA (2,1,1)				
	coef	std. error	t value	p-value
Constant	-0.117	0.342	-0.341	0.734
AR(1)	-0.699	0.404	-1.731	0.087
AR(2)	-0.091	0.093	-0.973	0.333
MA(1)	-0.391	0.385	-1.016	0.312

Output Model ARIMA Eviews

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.114059	0.478026	-0.238605	0.8120
AR(1)	-0.689932	3.102091	-0.222409	0.8245
AR(2)	-0.089495	1.060593	-0.084382	0.9329
MA(1)	0.382620	3.081708	0.124159	0.9015
SIGMASQ	31.31951	3.475630	9.011174	0.0000

Gambar 26. Perbandingan Tabel Estimasi Parameter ARIMA Menu Box-Jenkins Model

Berdasarkan hasil perbandingan Gambar 22 sampai 26 dapat dilihat bahwa terdapat beberapa perbedaan hasil pada menu *unit root test* dan *box-jenkins model*, hal ini dikarenakan perbedaan proses inisiasi nilai, metode penghitungan, serta perbedaan metode optimasi. Namun untuk menu lainnya telah memiliki hasil yang sama sehingga dapat disimpulkan bahwa hasil penghitungan yang dilakukan oleh Statify memiliki nilai yang valid.

H. Usability Testing

Pada tahapan ini, peneliti melakukan *usability testing* menggunakan kuesioner SUMI. Adapun pada tahapan ini peneliti memilih dua belas orang pengguna untuk melakukan skenario penggunaan Statify Modul Analisis *Time Series*. Selanjutnya kedua belas orang tersebut diminta untuk mengisi kuesioner SUMI, lalu data dari kuesioner tersebut dikirim ke pemilik lisensi SUMI karena pengolahan dalam aturan lisensi hanya dapat dilakukan oleh pemilik lisensi. Berikut Gambar 27 dan 28 menampilkan *summary statistic* SUMI dan laporan penilaian SUMI dari kedua belas responden.

	Mean	St Dev	Median	IQR	Minimum	Maximum
Global	62.08	7.29	65.0	12.0	48	70
Efficiency	60.58	6.72	61.5	12.5	50	67
Affect	57.08	8.32	54.5	11.5	42	69
Helpfulness	59.92	5.76	62.0	8.0	49	65
Controlability	59.17	9.77	58.5	16.0	45	74
Learnability	58.17	8.91	55.0	17.5	46	69

Gambar 27. Summary Statistics SUMI

Participant	Global	Efficiency	Affect	Helpfulness	Control	Learnability
1	70	67	69	65	69	69
2	70	63	69	65	60	55
3	68	67	53	62	74	69
4	67	67	49	62	74	55
5	67	67	54	63	65	55
6	65	60	55	65	56	47
7	65	53	64	65	53	53
8	60	59	61	59	58	50
9	57	67	52	55	49	69
10	54	50	64	50	59	69
11	54	56	42	59	45	46
12	48	51	53	49	48	61

Gambar 28. Laporan Penilaian SUMI

Berdasarkan laporan evaluasi pada Gambar 103, diketahui bahwa setiap aspek kuesioner SUMI menunjukkan nilai rata-rata di atas 50, yang mengindikasikan bahwa tingkat usability Statify Modul Analisis *Time Series* pada Statify tergolong baik. Secara spesifik berdasarkan hasil interpretasi Tabel 1, aspek *Global* dan *Efficiency* memperoleh nilai dengan predikat grade A, sementara aspek *Affect*, *Helpfulness*, *Controlability*, dan *Learnability* masing-masing memperoleh nilai dengan predikat grade B. Adapun hasil interpretasi tersebut didasarkan pada tabel interpretasi *t-score* SUMI yang ditampilkan pada Tabel IV.

TABEL IV
TABEL INTERPRETASI T-SCORE SUMI

No.	T-Score	Grade	Interpretasi
(1)	(2)	(3)	(4)
1	60 +	A	high
2	55-60	B	good
3	50-55	C +	above average
4	45-50	C -	below average
5	40-45	D	poor
6	40 -	E	low

A. Kesimpulan

Berdasarkan hasil dan pembahasan pada bagian sebelumnya, diperoleh kesimpulan sebagai berikut:

1. Telah dikembangkan Statify Modul Analisis *Time Series* dengan tampilan antarmuka yang efisien, membantu pengguna, serta mudah digunakan dan dipelajari dibuktikan pada *usability testing* kuesioner SUMI pada aspek *Efficiency*, berpredikat A serta pada aspek *Helpfulness*, *Controlability*, dan *Learnability* masing-masing berpredikat B, yang menunjukkan bahwa kedua aspek tersebut sudah cukup baik. Statify dibangun untuk digunakan secara gratis di lingkungan Politeknik Statistika STIS serta didesain dalam bentuk aplikasi berbasis web sehingga mempermudah akses pengguna tanpa perlu proses instalasi di awal dan hanya memerlukan akses internet pada perangkat untuk mengakses Statify.
2. Telah dikembangkan setiap fitur Statify Modul Analisis *Time Series* dalam bentuk menu *smoothing*, *decomposition*, *autocorrelation*, *unit root test*, dan *box-jenkins model* dengan proses penghitungan yang benar yang dibuktikan dengan hasil uji *white-box testing* dengan keterangan sukses pada setiap *test case*. Lalu dilanjutkan hasil statistik yang valid yang dibuktikan dengan membandingkan *output* antara Statify dan EvIEWS dengan hasil sebagian besar *output* bernilai sama serta beberapa *output* nilainya sedikit berbeda. Hal ini disebabkan oleh perbedaan sumber metode penghitungan, proses inisiasi nilai, dan proses optimasi.
3. Telah berhasil mengimplementasikan lima Sub-CPMK mata kuliah Analisis Runtun Waktu yakni pertemuan 2 sampai 6 dalam bentuk menu *smoothing*, *decomposition*, *autocorrelation*, *unit root test*, dan *box-jenkins model*.

B. Saran

Berikut beberapa saran dari peneliti untuk penyempurnaan dan pengembangan Statify Modul Analisis *Time Series* ke depannya.

1. Berdasarkan evaluasi validasi *output* menu *time series* antara Statify dan EvIEWS diperoleh beberapa perbedaan hasil karena metode penghitungan yang digunakan didominasi metode penghitungan klasik. Dengan demikian, peneliti selanjutnya dapat mengembangkan metode penghitungan ke metode penghitungan yang lebih *advanced* untuk memperoleh *output* statistik yang lebih baik.
2. Berdasarkan evaluasi *usability testing* menggunakan kuesioner SUMI diperoleh hasil bahwa aspek *Affect* dan *Learnability* merupakan dua aspek dengan nilai terendah dibandingkan aspek SUMI lainnya. Dengan demikian, peneliti selanjutnya dapat meningkatkan Statify Modul Analisis *Time Series* pada aspek *Affect* dan *Learnability* dengan menambahkan fitur arahan otomatis yang memberi petunjuk awal penggunaan menu yang interaktif untuk mempermudah pemahaman pengguna baru.

3. Berdasarkan batasan penelitian dan implementasi diperoleh hasil bahwa pengembangan Statify Modul Analisis *Time Series* masih terbatas pada Sub-CPMK mata kuliah Analisis Runtun Waktu yang dikembangkan masih terbatas pada pertemuan 2 sampai 6. Dengan demikian, peneliti selanjutnya dapat menambahkan fitur *time series* yang mengakomodasi materi mata kuliah Analisis Runtun Waktu Sub-CPMK pertemuan 7 hingga 14 untuk menjadikan Statify Modul Analisis *Time Series* sebagai aplikasi statistik yang mampu dalam mendukung pembelajaran Analisis Runtun Waktu secara komprehensif.

DAFTAR PUSTAKA

- [1] Badan Pusat Statistik, *Peraturan Badan Pusat Statistik Nomor 87 Tahun 2018*. Indonesia, 2018. Accessed: Jul. 27, 2025. [Online]. Available: https://ppid.bps.go.id/upload/doc/Peraturan_Badan_Pusat_Statistik_Nomor_87_Tahun_2018_1679371754.pdf
- [2] N. Nurhasanah, C. A. Nurpraja, S. Himawan, and A. Ramadanti, *Metode Peramalan Deret Waktu*, 1st ed. Bantul: KBM Indonesia, 2022.
- [3] W. Enders, *Applied Econometric Time Series*, 4th ed. Danvers: John Wiley & Sons, 2014.
- [4] S. G. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting: Methods and applications*. New York: John Wiley and Sons, 1997.
- [5] Paisal, N. Satyahadewi, and H. Perdana, "Pengembangan Aplikasi Statistika Berbasis Web Interaktif untuk Analisis Uji-T," *Bimaster*, vol. 10, pp. 331–340, 2021.
- [6] A. A. Sriwahyuni, S. Annas, and A. Saleh Ahmar, "Pengembangan Paket Analisis Regresi Berbasis Web," *Seminar Nasional Variansi*, 2018, [Online]. Available: <http://ojs.unm.ac.id/index.php/variantsistatistika>
- [7] R. R. Al-Hakim, Y. Yanuardi, R. J. Rumandan, and M. Tonggiroh, "Pengembangan Sistem Informasi Bimbingan Konseling Menggunakan Metode Feature-Driven Development," *Jurnal Ilmiah FIFO*, vol. 16, no. 2, p. 193, Nov. 2024, doi: 10.22441/fifo.2024.v16i2.009.
- [8] J. B. L. Sie, I. A. Musdar, and S. Bahri, "Pengujian White-Box Testing Terhadap Website Room Menggunakan Teknik Basis Path," *KHARISMA Tech*, vol. 17, pp. 45–57, Nov. 2022.
- [9] A. Pratomo and R. Mantala, "Pengembangan Aplikasi Ujian Berbasis Komputer Beserta Analisis Uji guna Sistem Perangkat Lunaknya Menggunakan Metode SUMI (Software Usability Measurement Inventory)," *POSITIF*, vol. 2, no. 1, pp. 1–11, Nov. 2016.
- [10] S. R. Palmer and J. M. Felsing, *A Practical Guide to Feature-Driven Development*, 1st ed. Saddle River: Prentice-Hall, 2022. [Online]. Available: www.togethersoft.com
- [11] S. Nidhra, "Black Box and White Box Testing Techniques - A Literature Review," *International Journal of Embedded Systems and Applications*, vol. 2, no. 2, pp. 29–50, Jun. 2012, doi: 10.5121/ijesa.2012.2204.
- [12] P. Ammann and J. Offutt, *Introduction to software testing*, 1st ed. New York: Cambridge University Press, 2013.