

Aplikasi Crowdfunding Bantuan Penelitian Skripsi/Tugas Akhir Mahasiswa Menggunakan Validasi Data Berbasis API Web Service**Gilang Pangestu Ramadan¹, Yus Sholva², Morteza Muthahhari³**^aProgram Studi Informatika, Fakultas Teknik, Universitas Tanjungpura

Jl. Prof. Dr. H. Hadari Nawawi, Pontianak 78124

gilangramadan@student.untan.ac.id morteza.muthahhari@teknik.untan.ac.idsholvariza@untan.ac.id**Abstract**

The COVID-19 pandemic has had a significant impact on the economy, causing many firms to close, wages to be cut, and massive PHKs. End-level students are also financially affected, having to use private funds for research. The scholarship corruption case in Aceh in 2017 showed fund misuse with a total loss of Rs. 10,091,000,000, as the lack of validation of student data and an effective cross-check system caused many scholarship recipients to be unqualified. Existing verification systems have weaknesses such as limited data integration, vulnerable manual verification, a lack of data updates, and weak data security. Enhanced cross-check systems through automation with APIs enable quick and accurate data validation, connecting academic and financial databases, and ensuring consistency of information without manual input. The use of APIs makes the system more efficient and secure, ensuring that aid is only given to eligible students who need financial assistance. The research aims to build a crowdfunding system on campus to help students obtain donation funds safely and transparently through the web service API for data validation. The system was developed using prototype and UML methods, with black-box testing and user acceptance testing. (UAT). The results showed an application running well with a usability rate of 81%, ease of use of 84%, functionality of 80%, and a display of 79%.

Article History*Submitted: 17 November 2024**Accepted: 22 November 2024**Published: 23 November 2024***Key Words**Applications Programming
interface, Crowdfunding,
Students, Validation, Web
service**Abstrak**

Pandemi Covid-19 berdampak signifikan pada ekonomi, menyebabkan banyak perusahaan tutup, gaji dipotong, dan PHK massal. Mahasiswa tingkat akhir juga terdampak secara ekonomi, harus menggunakan dana pribadi untuk penelitian. Kasus korupsi beasiswa di Aceh tahun 2017 menunjukkan penyalahgunaan dana dengan total kerugian Rp. 10.091.000.000, karena kurangnya validasi data mahasiswa serta sistem *cross-check* yang efektif menyebabkan banyak penerima beasiswa tidak memenuhi syarat. Sistem verifikasi yang ada memiliki kelemahan seperti integrasi data yang terbatas, verifikasi manual yang rentan kesalahan, kurangnya pembaruan data, dan keamanan data yang lemah. Peningkatan sistem *cross-check* melalui otomatisasi dengan API memungkinkan validasi data yang cepat dan akurat, menghubungkan *database* akademik dan keuangan, serta memastikan informasi konsisten tanpa *input* manual. Penggunaan API membuat sistem lebih efisien dan aman, memastikan bantuan hanya diberikan kepada mahasiswa yang memenuhi syarat dan membutuhkan bantuan finansial. Penelitian ini bertujuan membangun sistem *crowdfunding* di kampus untuk membantu mahasiswa mendapatkan dana donasi secara aman dan transparan melalui *API web service* untuk validasi data. Sistem dikembangkan menggunakan metode *prototype* dan UML, dengan pengujian *black-box testing* dan *user acceptance test* (UAT). Hasil menunjukkan aplikasi berjalan baik dengan tingkat usability 81%, kemudahan penggunaan 84%, fungsi 80%, dan tampilan 79%.

Sejarah Artikel*Submitted: 17 November 2024**Accepted: 22 November 2024**Published: 23 November 2024***Kata Kunci**Application Programming
Interface, Crowdfunding,
Mahasiswa, Validasi, Web
service**PENDAHULUAN**

Pandemi Covid-19 telah memberikan dampak yang signifikan di berbagai sektor kehidupan, termasuk ekonomi dan pendidikan. Mahasiswa yang sedang menyelesaikan skripsi atau tugas

akhir mereka, baik dari latar belakang ekonomi yang memadai maupun tidak, sering menghadapi kesulitan dalam membiayai kebutuhan penelitian mereka sendiri. Kasus korupsi beasiswa di Aceh menyoroti kurangnya sistem *cross-check* yang efektif untuk memverifikasi data mahasiswa sebelum menerima bantuan, sehingga banyak penerima beasiswa tidak memenuhi syarat yang ditetapkan. Hal ini menunjukkan perlunya perbaikan dalam manajemen dan pengawasan dana beasiswa.

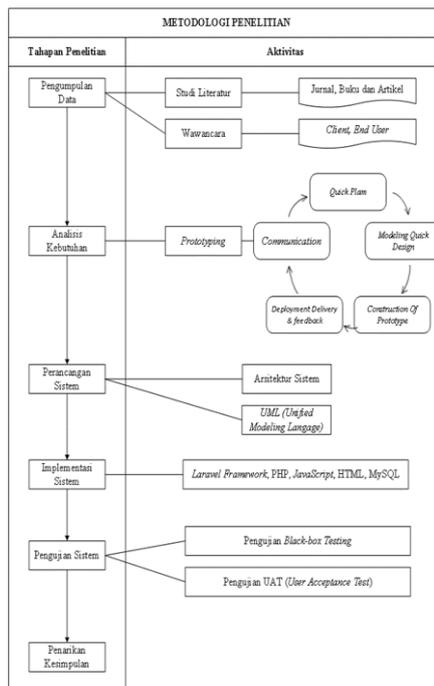
Crowdfunding adalah suatu praktik penggalangan dana untuk berbagai jenis usaha, baik berupa ide produk, bisnis atau kegiatan yang dananya diperoleh dari sumbangan masyarakat umum atau kelompok [1]. Karena cangkupannya yang luas dan dianggap lebih mudah digunakan untuk menggalang dana, sistem *crowdfunding*, yang beroperasi secara *online*, lebih mampu menarik perhatian masyarakat [2]. *Crowdfunding* telah terbukti menjadi solusi yang efektif untuk mengatasi keterbatasan dana dalam pendidikan. *Platform* seperti Kitabisa di Indonesia memfasilitasi pengumpulan dana secara transparan untuk berbagai tujuan, termasuk pendidikan. *Crowdfunding* tidak hanya memberikan kesempatan bagi individu dan organisasi untuk mendapatkan dukungan finansial, tetapi juga memperkuat keterlibatan komunitas dalam mendukung pendidikan inklusif.

Application Programming Interface merupakan fitur antarmuka dalam sebuah aplikasi yang memungkinkan pihak lain untuk mengakses dan menggunakan aplikasi tersebut tanpa perlu mengubah kode dasar atau struktur basis data. *API* memfasilitasi integrasi antar platform dan mempermudah komunikasi antar sistem [3]. Berfungsi untuk memberikan akses pengguna dalam proses pengambilan data. Berisi file *JavaScript Object Notation* (JSON) melalui arsitektur Representational State Transfer (REST), yang berjalan di atas *Hypertext Transfer Protocol* (HTTP)[4].

Penelitian ini bertujuan membangun *platform crowdfunding* khusus untuk mendukung mahasiswa dalam mengatasi kendala finansial saat menyelesaikan skripsi/tugas akhir di perguruan tinggi. Validasi data menggunakan status keaktifan mahasiswa yang diatur melalui sistem terintegrasi yang dikelola oleh perguruan tinggi. Keterlibatan ini penting untuk memastikan transparansi dalam penyaluran dana dari donatur kepada mahasiswa. Aplikasi ini menggunakan validasi data berbasis *API web service* untuk memastikan keabsahan dan mencegah penyalahgunaan data, sehingga dana disalurkan tepat sasaran kepada mahasiswa sebagai penerima manfaat.

METODOLOGI PENELITIAN

Tahapan yang digunakan dalam penelitian ini dapat dijelaskan melalui diagram alir pada gambar 1 sebagai berikut:



Gambar 1. Metodologi Penelitian

A. Pengumpulan Data

1) Studi Literatur

Penelitian kepustakaan dilakukan dengan tujuan mengumpulkan dan menelaah berbagai sumber yang relevan terkait masalah penelitian ini, serta mencari teori-teori pendukung yang dapat digunakan sebagai landasan untuk pendekatan selanjutnya [5], [6], [7], [8], [9].

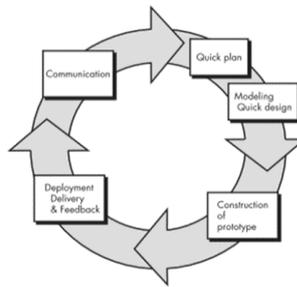
2) Wawancara

Metode wawancara melibatkan interaksi langsung antara peneliti dan responden, di mana peneliti mengajukan pertanyaan tentang topik penelitian dan responden memberikan tanggapan.

B. Analisis Kebutuhan

Tahap analisis kebutuhan ini melibatkan pendalaman mengenai kebutuhan fungsional dan non-fungsional dari aplikasi *crowdfunding* yang akan dibangun. Peneliti akan menggambarkan interaksi antara pengguna dan sistem, termasuk proses pengajuan dana, validasi data menggunakan *API*, serta penyaluran dana sebagai landasan penting bagi tahap perancangan.

Prototyping adalah proses membuat versi awal program komputer saat persyaratan belum jelas. Fungsinya adalah membantu memahami kebutuhan yang belum terdefinisi dengan menciptakan model awal yang bisa dievaluasi dan dimodifikasi [10]. Penelitian ini melakukan pendekatan menggunakan model *prototype* dengan tujuan untuk mengetahui serta menentukan kebutuhan-kebutuhan apa saja yang diperlukan secara cepat, berikut tahapan proses pengembangan. Model pengembangan *prototype* dapat dilihat pada gambar 2 sebagai berikut:



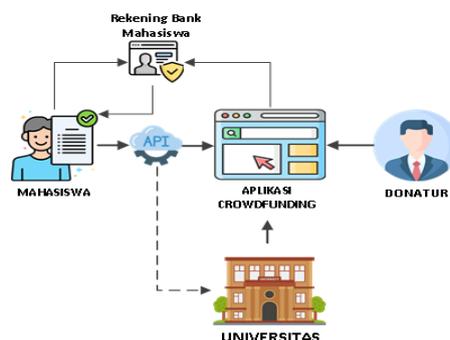
Gambar 2. Model *Prototype*

- 1) *Communication* (Komunikasi), Peneliti melakukan wawancara dengan pihak yang terlibat yaitu klien dan pengguna akhir (*end-user*) yang mana merupakan mahasiswa yang sedang melakukan penelitian tugas akhir atau skripsi dengan tujuan untuk mengidentifikasi permasalahan serta kebutuhan dalam pembuatan aplikasi *crowdfunding* ini.
- 2) *Quick plan* (Perencanaan), pada tahap ini dilakukan untuk menentukan sumber daya, aktor atau pengguna, spesifikasi kebutuhan hingga hasil yang dicapai dari aplikasi yang akan dibangun berdasarkan wawancara yang telah dilakukan.
- 3) *Modelling quick desain* (Pemodelan cepat), tahapan ini dilakukan untuk merancang sistem secara visual. Dimulai dengan membuat arsitektur sistem yang akan dibangun. Perancangan *use case diagram*, *activity diagram*, *class diagram*, *database* dan perancangan antarmuka tampilan aplikasi.
- 4) *Construction of prototype* (konstruksi), Pada tahap ini pemodelan yang telah dibuat sebelumnya akan diimplementasikan menjadi suatu sistem atau aplikasi.
- 5) *Deployment delivery & feedback* (Penyerahan), tahapan ini dilakukan untuk menguji coba aplikasi yang telah dibangun kepada pengguna untuk mendapatkan tanggapan, juga untuk memberikan hasil evaluasi dari implementasi sistem yang dibangun.

C. Perancangan Sistem

Adapun beberapa tahapan perancangan yang akan dilakukan pada aplikasi yang dibangun yaitu perancangan arsitektur sistem, *use case*, *entity relationship diagram*, *database*, mekanisme *API*, dan antarmuka tampilan.

1) Perancangan Arsitektur Sistem



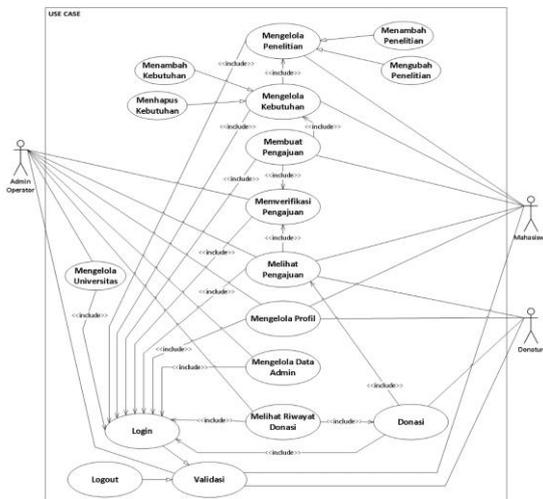
Tahapan ini menggambarkan relasi antar komponen yang terdapat di dalam sistem.

Gambar 3. Arsitektur Sistem

2) Perancangan Use Case

Tahap ini menjelaskan kerangka dari sistem yang akan dibangun dan fungsionalitas yang disediakan oleh sistem tersebut. *Use case* digunakan untuk menggambarkan siapa atau apa

yang berinteraksi dengan sistem. Dengan kata lain, diagram *use case* memberikan gambaran tentang bagaimana sistem berinteraksi dengan pengguna atau entitas lainnya serta tugas-tugas

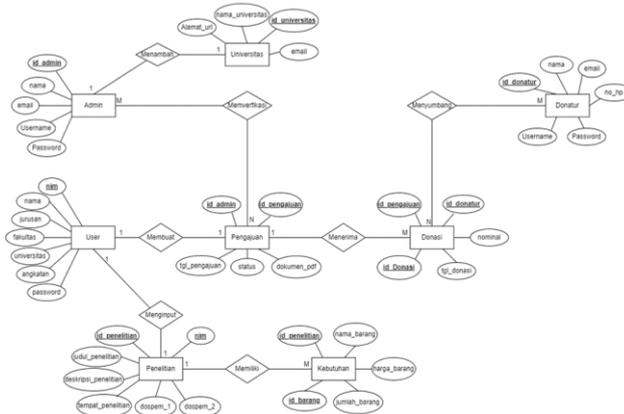


apa yang harus dilakukan oleh sistem sesuai kebutuhan penggunannya [11].

Gambar 4. Diagram Use Case

3) Perancangan Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) menggambarkan hubungan antar entitas yang ada pada sistem. Menurut Pulungan et al., (2023) Entity Relationship adalah metode untuk memodelkan basis data menggunakan skema konseptual yang menggambarkan hubungan antara data. Model ini biasanya digunakan dalam basis data relasional dengan pendekatan *top-down*. Diagram yang digunakan disebut diagram Entity Relationship, ER diagram, atau ERD. Entitas adalah objek yang dapat diidentifikasi secara unik dan dihubungkan satu sama lain melalui hubungan (*relationship*). Atribut adalah karakteristik yang menggambarkan



setiap entitas [12].

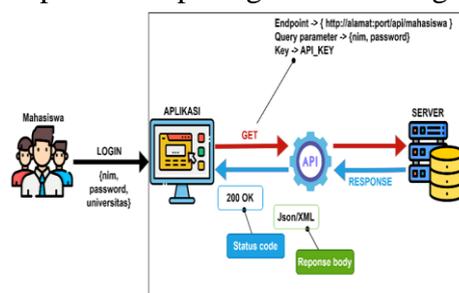
Gambar 5. Entity Relationship Diagram

4) Perancangan Database

Database merupakan sistem yang terdiri dari sejumlah tabel data yang saling terkait, serta sejumlah program yang memungkinkan beberapa pengguna atau program lain untuk membuka dan mengelola tabel-tabel data tersebut. Perancangan *database* dibuat untuk mengetahui nama tabel, kolom dan tipe data apa saja yang digunakan dalam pembuatan *database* [13].

5) Perancangan Mekanisme API

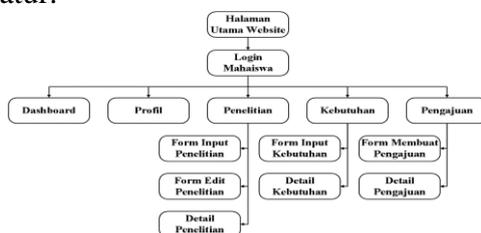
Application programming Interface biasanya terdiri dari dua jenis komponen. Satu komponen menyediakan *API*, dan komponen lainnya menggunakan *API*. Klien juga menggunakan *API* untuk terhubung ke sumber data atau layanan yang disediakan secara terpusat [14]. Dengan menerapkan pendekatan secara arsitektural menggunakan standar HTTP untuk melakukan komunikasi antara klien dan *server*, REST didasarkan pada konsep bahwa setiap sumber daya dalam sistem harus memiliki URI (*Uniform Resource Identifier*) yang unik dan dapat diakses menggunakan metode HTTP standar *GET*, *POST*, *PUT*, *DELETE*, dan lain-lain [15]. Mengirimkan data dalam bentuk format JSON (*JavaScript Object Notation*), yang memiliki pasangan nama dan nilai yang mudah dipahami, ukurannya kecil, mudah digunakan, dan kompatibel dengan berbagai bahasa pemrograman [16]. Perancangan mekanisme *API* dapat dilihat pada gambar 6 sebagai berikut:



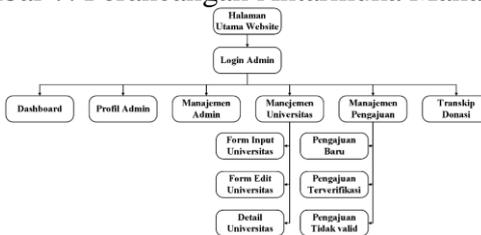
Gambar 6. Mekanisme *Application Programming Interface*

6) Perancangan Antarmuka Tampilan

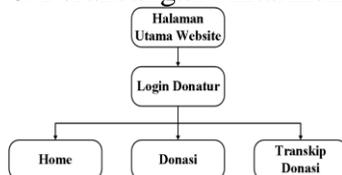
Perancangan struktur antarmuka (*interface*) dirancang sebagai gambaran awal model yang dibangun. Pada perancangan dibentuk dengan fungsi-fungsi yang sesuai dengan proses. Model yang dibangun pada aplikasi *Crowdfunding* terdapat 3 antarmuka yaitu antarmuka admin, mahasiswa, dan donatur.



Gambar 7. Perancangan Antarmuka Mahasiswa



Gambar 8. Perancangan Antarmuka Admin



Gambar 9. Perancangan Antarmuka Donatur

D. Implementasi Sistem

Pada tahap implementasi, penulis membangun sistem baru dengan mengimplementasi hasil desain sistem yang telah dilakukan. Penulis membangun sistem *crowdfunding* untuk mahasiswa yang terkendala finansial dalam menyelesaikan penelitian skripsi atau tugas akhir

dengan menerapkan teknologi *API web service* menggunakan *framework* laravel, bootstrap 5 dan *MySQL* sebagai penyimpanan data.

E. Pengujian Sistem

Pengujian sistem dilakukan dengan maksud untuk memeriksa apakah aplikasi yang telah dibuat beroperasi seperti yang telah direncanakan. Proses pengujian dalam penelitian ini menggunakan pengujian *Black Box*, *User Acceptance Test* (UAT) dan Postman untuk pengujian *API*.

1) Black Box

Black-box testing adalah proses pengujian perangkat lunak berdasarkan spesifikasi fungsional. Tujuannya adalah memastikan fungsi, masukan, dan keluaran perangkat lunak sesuai dengan yang diinginkan, tanpa melihat desain atau kode programnya [17].

2) User Acceptance Test

User Acceptance Testing (UAT) adalah pengujian yang dilakukan oleh pengguna akhir, di mana pengguna tersebut berinteraksi langsung dengan sistem. Pengguna memeriksa apakah fungsionalitas sistem berjalan sesuai dengan kebutuhan dan fungsinya atau tidak [18].

3) Postman

Postman adalah aplikasi untuk pengujian *API* yang kini menjadi *platform* pengembangan *API* dengan fitur-fitur untuk pembuatan, penggunaan, dokumentasi, dan kolaborasi. Aplikasi ini memudahkan pengembang untuk melihat, mengirim, dan mengelola permintaan *API* serta mengecek keberhasilan dan respons permintaan tersebut. Postman juga mendukung pengulangan dan berbagi proyek *API* dengan rekan kerja atau komunitas [19].

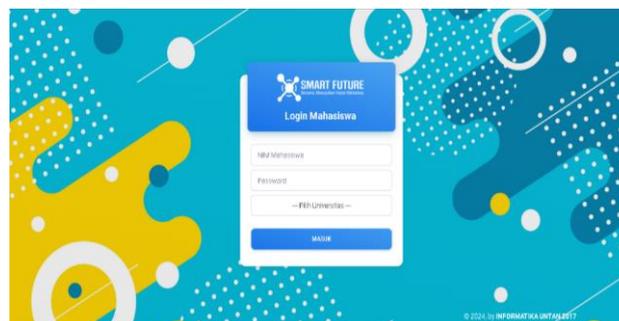
I. HASIL DAN PEMBAHASAN

Berikut merupakan hasil dari perancangan dan pembuatan aplikasi serta hasil dari pengujian yang telah dilakukan menggunakan *Black Box*, *User Acceptance Test* serta Postman.

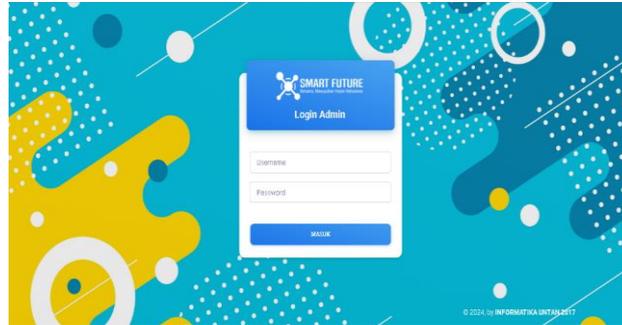
A. Implementasi



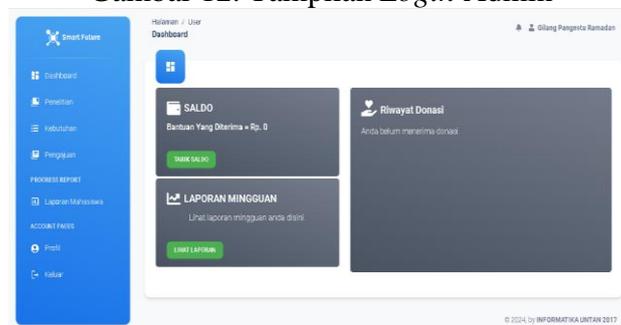
Gambar 10. Tampilan Utama Aplikasi



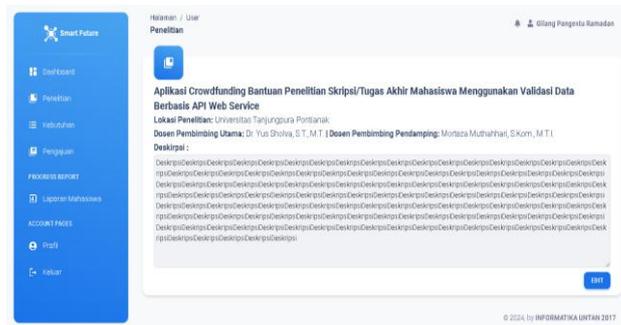
Gambar 11. Tampilan Login Mahasiswa



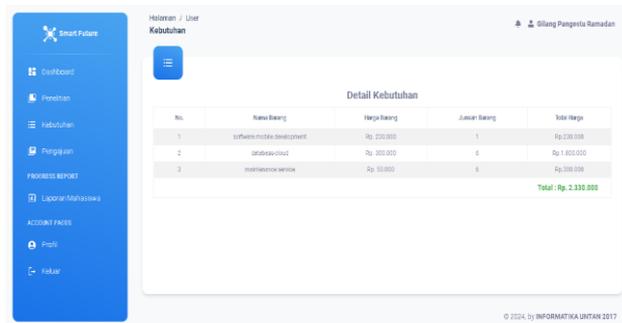
Gambar 12. Tampilan *Login Admin*



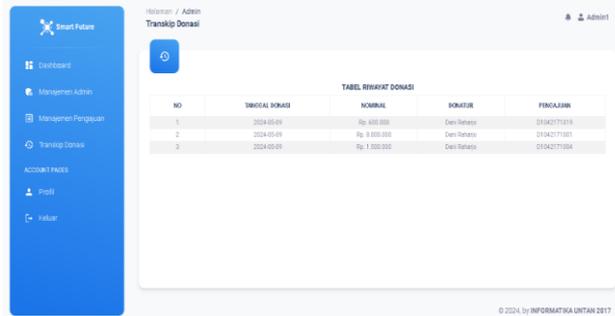
Gambar 13. Tampilan *Dashboard Mahasiswa*



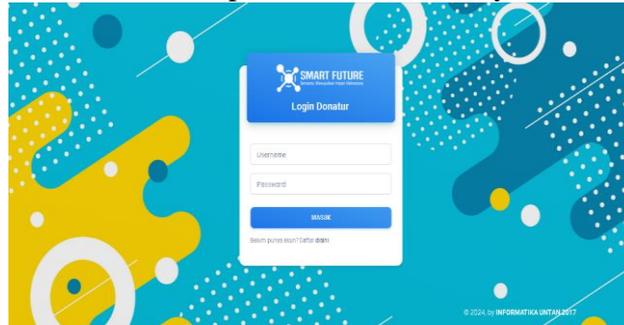
Gambar 14. Tampilan Halaman Penelitian Mahasiswa



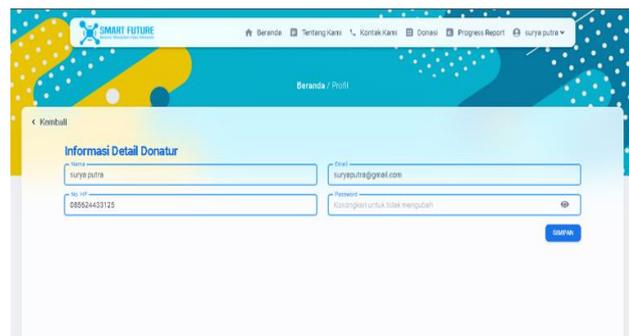
Gambar 15. Tampilan Halaman Kebutuhan Mahasiswa



Gambar 21. Tampilan Halaman Riwayat Donasi



Gambar 22. Tampilan Halaman Login Donatur



Gambar 23. Tampilan Halaman Donatur

B. Pengujian Black Box

Pengujian dengan metode *Black Box* dilakukan untuk memastikan sistem menghasilkan *output* yang benar dan sesuai dengan harapan pengguna. Berikut adalah hasil pengujian dengan menggunakan metode *Black Box*.

TABEL I
PENGUJIAN *BLACK BOX*

No	Uji Kasus	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
1	<i>Login</i>	Apabila pengguna melakukan login dengan memasukkan username dan password yang sesuai	Sistem dapat memverifikasi login dari akun pengguna	Berhasil

		makan sistem dapat memverifikasi input-an tersebut		
	<i>Logout</i>	Apabila pengguna ingin keluar dari akun dengan menekan tombol logout, maka sistem dapat memverifikasi hal tersebut	Apabila pengguna ingin keluar dari akun dengan menekan tombol logout, maka sistem dapat memverifikasi hal tersebut	Berhasil
2	Melihat Profil	Apabila pengguna ingin melihat profil, maka sistem akan menampilkan data profil dari pengguna tersebut	Sistem menampilkan data profil yang sesuai dengan penggunaanya	Berhasil
	Mengubah profil	Ketika pengguna melakukan perubahan pada profil, maka sistem akan menyimpan perubahan data tersebut	Sistem menyimpan perubahan data profil dan pengguna dapat melihat perubahan tersebut	Berhasil
3	Menambah Penelitian Mahasiswa	Ketika mahasiswa melakukan input data penelitian, maka sistem akan menyimpan data tersebut.	Sistem menyimpan data penelitian dan mahasiswa dapat melihat data penelitian tersebut	Berhasil

	Mengubah Penelitian Mahasiswa	Ketika mahasiswa melakukan perubahan data penelitian, maka sistem data menyimpan perubahan tersebut.	Sistem menyimpan data penelitian dan mahasiswa dapat melihat perubahan data penelitian tersebut	Berhasil
4	Menambah Kebutuhan	Ketika mahasiswa melakukan input data kebutuhan, maka sistem akan menyimpan data tersebut	Sistem menyimpan data kebutuhan dan mahasiswa dapat melihat data kebutuhan tersebut	Berhasil
	Menghapus Kebutuhan	Mahasiswa menghapus data kebutuhan yang ada	Sistem menghapus data kebutuhan yang dihapus atau dipilih	Berhasil
5	Membuat Pengajuan	Mahasiswa membuat pengajuan, maka sistem akan menyimpan data pengajuan tersebut	Sistem menyimpan data pengajuan dan mahasiswa dapat melihat data pengajuan tersebut	Berhasil
6	Menambah Admin Baru	Ketika admin melakukan input data admin baru, maka sistem akan menyimpan data admin tersebut	Sistem menyimpan data admin baru dan admin dapat melihat data admin baru tersebut	Berhasil
	Mengubah Data Admin	Ketika admin melakukan perubahan data admin, maka	Sistem menyimpan data admin dan admin dapat melihat perubahan	Berhasil

		sistem menyimpan perubahan tersebut.	data tersebut	
	Menghapus Data Admin	Admin menghapus data admin yang ada	Sistem menghapus data admin yang dihapus atau dipilih	Berhasil
7	Menambah Data Universitas	Ketika admin melakukan input data universitas, maka sistem akan menyimpan data tersebut	Sistem menyimpan data universitas dan admin dapat melihat data universitas tersebut	Berhasil
	Mengubah Data Universitas	Ketika admin melakukan perubahan data universitas, maka sistem data menyimpan perubahan tersebut	Sistem menyimpan data universitas dan admin dapat melihat perubahan data universitas tersebut	Berhasil
	Menghapus Data Universitas	Admin menghapus data kebutuhan yang ada	Sistem menghapus data kebutuhan yang dihapus atau dipilih	Berhasil
8	Memverifikasi Pengajuan	Ketika admin menyetujui/menolak pengajuan, maka sistem akan menyimpan perubahan tersebut.	Sistem menyimpan perubahan status data pengajuan dan admin dapat melihat perubahan tersebut	Berhasil
9	Melihat Riwayat Donasi	Ketika admin mengakses menu riwayat donasi,	Sistem menampilkan riwayat donasi yang ada	Berhasil

		maka sistem menampilkan hasil riwayat donasi		
--	--	--	--	--

C. Pengujian User Acceptance Testing (UAT)

Pengujian *User Acceptance Test (UAT)* dilakukan dengan mengisi kuesioner yang dilakukan terhadap mahasiswa yang sedang melakukan penelitian skripsi/tugas akhir mahasiswa. Hasil pengujian UAT akan dipresentasikan dengan menggunakan metode perhitungan *Skala Likert*.

1) Hasil Kuisisioner Aplikasi

**TABEL II
HASIL KUESIONER RESPONDEN**

ASPEK KEMUDAHAN PERANGKAT LUNAK		TANGGAPAN				
No	Pertanyaan	SS	S	C	TS	STS
1	Apakah antarmuka pengguna aplikasi mudah digunakan dan intuitif?	4	6	0	0	0
2	Apakah aplikasi memberikan umpan balik yang tepat saat pengguna melakukan tindakan?	3	6	1	0	0
3	Apakah menu-menu aplikasi dapat diakses dengan mudah?	4	5	0	1	0
4	Apakah aplikasi menangani data dan proses dengan cepat dan efisien?	5	2	3	0	0
5	Apakah aplikasi berfungsi baik di berbagai perangkat dan browser yang didukung?	2	5	3	0	0
ASPEK FUNGSIONALITAS		TANGGAPAN				
No	Pertanyaan	SS	S	C	TS	STS
1	Apakahh fitur-fitur yang disediakan oleh aplikasi ini lengkap sesuai dengan kebutuhan?	5	2	3	0	0
2	Apakah fungsi-fungsi dalam aplikasi sesuai dengan deskripsi dan tujuan penggunaannya?	4	3	3	0	0
3	Apakah fitur-fitur di aplikasi ini mudah digunakan?	5	2	3	0	0
4	Apakah dapat menyelesaikan tugas-tugas dengan cepat menggunakan aplikasi ini?	1	6	1	2	0
5	Apakah data yang dihasilkan atau ditampilkan oleh aplikasi ini akurat?	2	2	4	0	0
ASPEK VISUAL		TANGGAPAN				
No	Pertanyaan	SS	S	C	TS	STS
1	Apakah tata letak dan desain antarmuka aplikasi ini menarik?	4	4	2	0	0
2	Apakah desain dan elemen visual di seluruh aplikasi ini konsisten?	4	1	4	1	0
3	Apakah teks dan ikon dalam aplikasi ini mudah dibaca dan dipahami?	4	2	4	0	0
4	Apakah pemilihan warna dalam antarmuka aplikasi ini menarik?	4	1	4	0	1

Total	51	49	35	4	1
-------	----	----	----	---	---

Berdasarkan data yang diperoleh pada tabel di atas merupakan hasil kuesioner yang didapat dari responden.

2) Perhitungan Skor Responden

TABEL III
PERHITUNGAN SKOR RESPONDEN

Jawaban Responden			
Bobot	Jumlah Skor	Nilai Bobot	Hasil
SS	51	5	51 x 5 = 255
S	49	4	49 x 4 = 196
C	35	3	35 x 3 = 105
TS	4	2	4 x 2 = 8
STS	1	1	1 x 1 = 1
Jumlah Total Skor			565

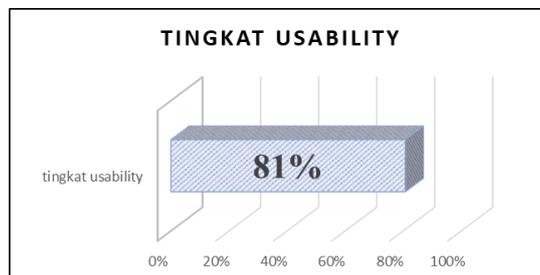
Table III di atas menjelaskan perhitungan skor responden dengan persentase jawaban sebagai berikut:

- Persentase jawaban SS = $255 / 565 \times 100\% = 45\%$
- Persentase jawaban S = $196 / 565 \times 100\% = 35\%$
- Persentase jawaban C = $105 / 565 \times 100\% = 19\%$
- Persentase jawaban T S = $4 / 565 \times 100\% = 1\%$
- Persentase jawabn STS = $1 / 565 \times 100\% = 0\%$

Hasil dari jawaban responden sebanyak 10 orang tersebut selanjutnya dapat dihitung untuk nilai tertinggi dan terendah sebagai berikut.

- Nilai Tertinggi = $10 \times 14 \times 5 = 700$ (seandainya semua menjawab SS).
- Nilai Terendah = $10 \times 14 \times 1 = 140$ (seandainya semua menjawab STS).

3) Perhitungan Skor Usability



Gambar 24. Hasil Perhitungan Usability

Dari gambar di atas merupakan hasil persentase pengisian kuisisioner dari mahasiswa selaku responden terhadap aplikasi *crowdfunding* bantuan penelitian skripsi/tugas akhir mahasiswa menggunakan validasi data berbasis *API web service* yang dibangun, tingkat persentase skor *usability*-nya yaitu 81%. Berdasarkan penilaian dari responden tergolong sangat setuju.

4) Hasil Pengukuran Aspek

TABEL IV
HASIL PENGUKURAN ASPEK

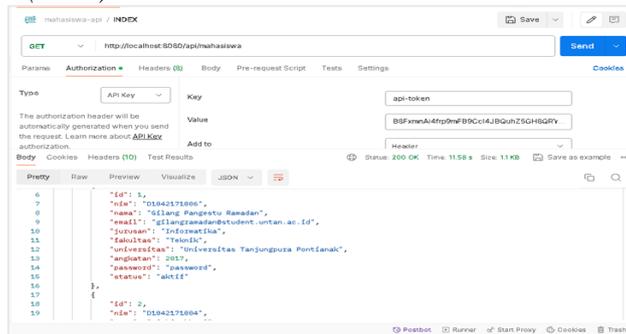
No.	Aspek Pengujian	Butir Pertanyaan	Skor	%
1.	Kemudahan Aplikasi	5	209	84%
2.	Fungsi	5	199	80%
3.	Tampilan Aplikasi	4	157	79%

Berdasarkan hasil rata-rata persentase dari setiap kriteria yang diuji yaitu kemudahan aplikasi, fungsi dan tampilan aplikasi dengan perolehan nilai yang diberikan oleh responden yang menjawab sangat setuju terhadap kemudahan aplikasi sebesar 84%, untuk fungsi sebesar 80% dan tampilan aplikasi sebesar 79%.

D. Hasil Peengujian API

berikut merupakan hasil uji coba URI yang dibangun pada aplikasi menggunakan *Postman*. Pada URI ini menghasilkan respon berupa data *JSON*.

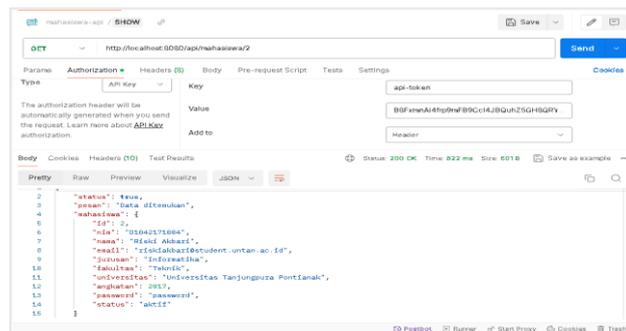
1) URI /api/mahasiswa (GET)



Gambar 25. URI /api/mahasiswa (GET)

Berikut merupakan hasil uji coba pada URI /api/mahasiswa dengan *method* (GET). Pada URI ini menghasilkan respon berupa data json semua mahasiswa.

2) URI /api/mahasiswa/{id} (GET)



Gambar 26. URI /api/mahasiswa (GET)

Berikut merupakan hasil uji coba pada URI /api/mahasiswa/{id} dengan *method* (GET). Pada URI ini menghasilkan respon data json mahasiswa dengan parameter id.

KESIMPULAN

Hasil pengujian black-box testing pada aplikasi menandakan bahwa semua fungsi yang telah dibangun sudah berjalan dengan baik dan sesuai seperti yang diharapkan. Sedangkan hasil dari pengujian user acceptance test (UAT) menunjukkan bahwa semua fungsi telah sesuai dengan kebutuhan pengguna, hasil tersebut didapati dari beberapa uji kriteria terdiri dari kemudahan aplikasi dengan skor 84%, fungsi aplikasi dengan skor 80%, dan aspek tampilan dengan skor 79%. Dengan hasil presentase keseluruhan aplikasi yaitu dengan skor 81%.

DAFTAR PUSTAKA

M. R. Effendi, M. Narji, and Y. S. Andhana Saputra, "PENGEMBANGAN APLIKASI PENGGALANGAN DANA MENGGUNAKAN KONSEP CROWDFUNDING BERBASIS MOBILE PADA PESANTREN AL-QUR'AN MAFATIH," *J. Sist. Inf. Univ. SURYADARMA*, vol. 8, no. 2, p. 16, Jun. 2021, doi: 10.35968/jsi.v8i2.717.

I. A. Aziz, N. Nurwahidin, and I. Chailis, "Faktor-Faktor Yang Mempengaruhi Masyarakat Menyalurkan Donasi Melalui Platform Crowdfunding Berbasis Online," *J. Syarikah J. Ekon. Islam*, vol. 5, no. 1, pp. 94–108, 2019, doi: 10.30997/jsei.v5i1.1835.

- D. Wijonarko and B. W. R. Mulya, “Pengembangan Antarmuka Pemrograman Aplikasi Menggunakan Metode RESTful pada Sistem Informasi Akademik Politeknik Kota Malang,” *Smatika J.*, vol. 8, no. 02, pp. 63–66, 2018, doi: 10.32664/smatika.v8i02.202.
- A. B. Warsito, A. Ananda, and D. Triyanjaya, “Penerapan Data JSON Untuk Mendukung Pengembangan Aplikasi Pada Perguruan Tinggi Dengan Teknik Restfull Dan Web Service,” *Technomedia J.*, vol. 2, no. 1, pp. 26–36, 2017, doi: 10.33050/tmj.v2i1.313.
- C. M. Sari, D. Jefriansyah, and M. Norriastini, “Aplikasi Penggalangan Dana Komunitas Jumat Jaman Now Berbasis Web Menggunakan Framework Laravel,” *Karya Ilm. Mahasiswa.*, pp. 90–100, 2019.
- R. Choirudin and A. Adil, “Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa,” *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 284–293, 2019, doi: 10.30812/matrik.v18i2.407.
- I. A. Kaniya Pradnya Paramitha, D. M. Wiharta, and I. M. Arsa Suyadnya, “Perancangan Dan Implementasi Restful Api Pada Sistem Informasi Manajemen Dosen Universitas Udayana,” *J. SPEKTRUM*, vol. 9, no. 3, p. 15, 2022, doi: 10.24843/spektrum.2022.v09.i03.p3.
- F. Thoriq Faza and A. Indriani, “Adopsi Penggunaan Platform Crowdfunding untuk Donasi Amal pada Kalangan Muslim Milenial,” *JEBA (Journal Econ. Bus. Aseanomics)*, vol. 6, no. 1, pp. 60–70, 2021, doi: 10.33476/j.e.b.a.v6i1.1926.
- B. Baharuddin, H. Wakkang, and B. Irianto, “Implementasi Web Service Dengan Metode Rest Api Untuk Integrasi Data Covid 19 Di Sulawesi Selatan,” *J. Sintaks Log.*, vol. 2, no. 1, pp. 236–241, 2022, doi: 10.31850/jsilog.v2i1.1035.
- R. S. Pressman, *SOFTWARE ENGINEERING: A PRACTITIONER’S APPROACH, SEVENTH EDITION*. New York, NY, USA: McGraw-Hill Companies, Inc., 2010. doi: 10.1145/336512.336521.
- G. Booch, R. A. Maksimchuk, M. W. Engle, B. J. Young, J. Connallen, and K. A. Houston, *Object-oriented analysis and design with applications, third edition*, Third., vol. 33, no. 5. 2008. doi: 10.1145/1402521.1413138.
- S. M. Pulungan, R. Febrianti, T. Lestari, N. Gurning, and N. Fitriana, “Analisis Teknik Entity-Relationship Diagram Dalam Perancangan Database,” *J. Ekon. Manaj. dan Bisnis*, vol. 1, no. 2, pp. 98–102, 2023, doi: 10.47233/jemb.v1i2.533.
- M. Fikry, *Buku Basis Data*, 1st ed. Lhokseumawe: Unimal Press, 2019.
- M. Biehl, *API Architecture: The Big Picture for Building APIs*. 2015.
- D. Jacobson, G. Brail, D. Woods Editor, and M. Treseler, “APIs: A Strategy Guide O’Reilly Media,” 2011. [Online]. Available: <http://my.safaribooksonline.com>
- J. Friesen, *Java XML and JSON*. 2019. doi: 10.1007/978-1-4842-4330-5.
- R. A. Sukamto and M. Salahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: INFORMATIKA, 2016.
- R. Supriatna, “Implementasi Dan User Acceptance Test (UAT) Terhadap Aplikasi E-Learning,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.
- Postman, “Postman API Platform [Internet],” postman .inc. Accessed: Jun. 19, 2024. [Online]. Available: <https://www.postman.com/product/what-is-postman/>