

RANCANG BANGUN *MULTI-PLATFORM MOBILE APPLICATION* SISTEM INFORMASI PENDUKUNG PROGRAM MERDEKA BELAJAR

Aghni Qisthina Al Rahma^{1*}, Mohamad Irwan Afandi², Amalia Anjani Arifiyanti³

^{1,2,3}Universitas Pembangunan Nasional "Veteran" Jawa Timur; Jl. Rungkut Madya No.1 Surabaya Jawa Timur; (031) 8706369

Received: 4 Juli 2024

Accepted: 31 Juli 2024

Published: 7 Agustus 2024

Keywords:

Sistem Informasi;
Mobile Application;
Multi-Platform;
MBKM;
Flutter;

Correspondent Email:

qisthiaghni@gmail.com

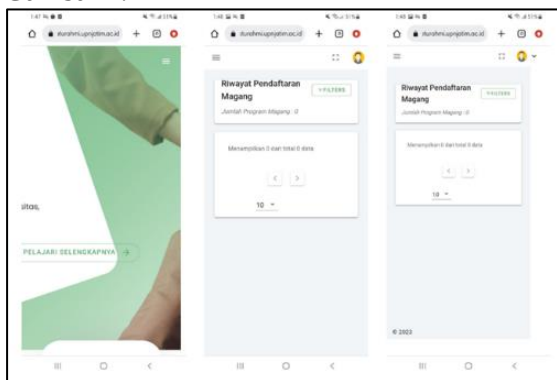
Abstrak. Aplikasi SILATURAHMI (Sistem Informasi Kolaborasi Terpadu Universitas, Pemerintah, Masyarakat, dan Industri) merupakan sistem informasi milik UPN "Veteran" Jawa Timur yang dirancang untuk mengelola program MBKM dari pendaftaran hingga finalisasi penilaian konversi mata kuliah. Namun, sistem ini masih berbasis website yang dari awal dioptimasi untuk perangkat berbasis desktop. Pada perangkat berbasis *mobile*, sistem ini memiliki performa dan tampilan yang kurang baik sehingga muncul kebutuhan akan aplikasi berbasis *mobile* yang terpisah. Sistem informasi SILATURAHMI berbasis *mobile* dikembangkan dengan menerapkan metode Waterfall yang terdiri dari tahap *communication*, *planning*, *modelling*, *construction*, dan *deployment* dengan menggunakan *framework* flutter. Melalui metode ini, diharapkan sistem yang dihasilkan dapat mempermudah pengguna, baik mahasiswa maupun dosen, dalam mengelola program MBKM melalui *device* berbasis *mobile*. Berdasarkan hasil pengujian yang dilakukan, Sistem Informasi SILATURAHMI berbasis *mobile* telah berhasil dijalankan di kedua sistem operasi yaitu AndroidOS dan iOS sesuai dengan proses bisnis yang telah direncanakan. Melalui sistem berbasis *mobile* ini dosen dan mahasiswa dapat lebih mudah dalam mengakses sistem informasi SILATURAHMI serta mengurangi kesalahan-kesalahan akibat tampilan dan performa yang kurang baik pada website.

Abstract. *SILATURAHMI application is an information system developed by UPN "Veteran" Jawa Timur for managing the MBKM program from registration to course conversion assessments. However, this system is still based on a website which was optimized for desktop-based devices. On mobile-based devices, this system has poor performance and appearance, resulting in the need of a separate mobile-based application. The mobile-based SILATURAHMI information system was developed by applying the Waterfall method which consists of communication, planning, modelling, construction and deployment stages using Flutter framework. Through this method, it is expected that the resulting system can facilitate users, both students and lecturers, in managing the MBKM program through mobile-based devices. Based of the results of the tests, the mobile-based SILATURAHMI Information System has been successfully run on both operating systems, Android OS and iOS, in accordance with the planned business process. Through this mobile-based system, lecturers and students can more easily access the SILATURAHMI information system and reduce errors due to poor appearance and performance on the website.*

1. PENDAHULUAN

MBKM (Merdeka Pelajar Kampus Merdeka) merupakan salah satu dari program yang diadakan oleh Kementerian Pendidikan dan Teknologi. Program ini memungkinkan mahasiswa dari berbagai universitas di Indonesia untuk memiliki pengalaman belajar bersama pakar dan meraih konversi SKS hingga 20 SKS. Transformasi penilaian SKS dari mitra perusahaan MBKM menjadi nilai mata kuliah membutuhkan proses penyetaraan yang dilakukan agar materi yang didapatkan selama MBKM dengan mata kuliah yang dikonversi dapat sesuai.

Aplikasi SILATURAHMI (Sistem Informasi Kolaborasi Terpadu Universitas, Pemerintah, Masyarakat, dan Industri) adalah sistem informasi manajemen MBKM yang dirancang oleh UPN "Veteran" Timur [1]. Sistem informasi ini dirancang untuk meningkatkan efisiensi dan efektivitas dari pengelolaan program MBKM. Namun, sistem ini masih terdapat beberapa kekurangan bila digunakan pada perangkat berbasis *mobile*. Sedari awal, sistem ini dioptimasi untuk digunakan pada perangkat berbasis desktop sehingga dalam penggunaannya pada perangkat berbasis *mobile* masih dapat dikatakan kurang. Tampilan sistem pada situs website masih tidak merespon dengan baik saat digunakan pada perangkat *mobile* seperti yang terlihat pada Gambar 1.



Gambar 1. Tampilan SILATURAHMI berbasis Website

Penggunaan sistem SILATURAHMI berbasis website mengalami perubahan tampilan saat digunakan pada *device* berbasis *mobile*. Kolom-kolom yang ditampilkan dalam bentuk tabel saling bertumpang tindih dan tidak selaras dengan panjang *device* yang seharusnya. Pada sistem, terdapat beberapa paragraf yang

cukup panjang. Namun, sistem ini gagal dalam menampilkan paragraf tersebut secara ringkas sehingga menyulitkan pengguna dalam membaca paragraf tersebut. Selain itu, gambar dalam sistem gagal dalam menyesuaikan ukurannya sehingga tidak terlihat apik saat digunakan oleh pengguna. Tidak hanya itu, mayoritas pengguna merupakan mahasiswa dan dosen dengan berbagai kesibukannya sehingga menampilkan data berdasarkan tanggal pengunggahan tentu menyulitkan mereka. Melihat permasalahan yang dihadapi oleh pengguna sistem, sistem informasi ini memerlukan aplikasi berbasis *mobile* yang dioptimalkan secara khusus untuk digunakan pada perangkat seluler.

Aplikasi berbasis *mobile* memiliki berbagai keunggulan. Salah satunya yaitu dari sisi pengalaman penggunanya yang lebih baik daripada website yang diakses pada *device* berbasis *mobile* seperti *smartphone*. Perangkat *mobile* itu sendiri merupakan teknologi yang memiliki tingkat mobilitas lebih tinggi daripada perangkat lainnya seperti komputer, PC, dan sebagainya [2]. Pengembangan aplikasi seluler ini tidak hanya meningkatkan aksesibilitas pengguna, tetapi juga memberikan kemudahan operasional dan fleksibilitas selama penggunaannya.

Data dari Badan Pusat Statistik (BPS) mencatat 67,88 persen dari masyarakat Indonesia telah memiliki ponsel pada tahun 2022. Hal ini menunjukkan mayoritas orang Indonesia telah memiliki perangkat berbasis *mobile* dan telah menguasai perangkat tersebut untuk saling berbagi informasi melalui internet. *Device mobile* sendiri memiliki berbagai sistem operasinya sendiri. Masing-masing sistem operasi ini melibatkan bahasa pemrograman, *environment* pengembangan, SDK, dan pasar distribusinya masing-masing [2]. Hal ini menyebabkan pengembang perangkat lunak harus memutuskan sedari awal aplikasi apa yang akan dirancang dan pada platform apa aplikasi ini akan dirilis. Untuk menghadapi kendala tersebut, *developer* dapat menggunakan *framework multi-platform*. *Framework* ini memungkinkan pihak pengembang untuk mengembangkan aplikasi yang dapat berjalan di berbagai platform hanya dengan satu kali pengembangan. Salah satu kerangka kerja *multi-platform* yang banyak digunakan adalah Flutter. *Framework* ini

menyederhanakan pengembangan aplikasi *multi-platform* hanya dengan satu kode dasar. Menurut [3], *Framework* ini memiliki kelebihan yaitu tidak menggunakan konektor atau penyambung yang dapat menjadi sumber dari terjadinya *bottleneck*. Selaras dengan hasil penelitian dari [4], penelitian ini memberikan perbandingan kinerja antar *framework* yaitu Flutter dan React Native. Studi ini membuktikan bahwa Flutter merupakan solusi yang optimal untuk pengembangan aplikasi pada sistem operasi berbasis *mobile* seperti iOS dan AndroidOS. Oleh karena itu, penelitian ini akan membahas pengembangan sistem informasi seluler menggunakan *framework multi-platform* Flutter.

2. TINJAUAN PUSTAKA

Penelitian ini disusun berdasarkan teori-teori yang mendukung permasalahan atau ruang lingkup yang dibahas.

2.1. Sistem Informasi

Menurut Rainer dan Prince (2016), sistem informasi memiliki definisi sistem yang dapat mengumpulkan, memproses, menyimpan, menganalisis, dan membagikan informasi secara meluas untuk tujuan-tujuan tertentu. Sistem informasi berperan sebagai sistem yang dapat menjadi *tools* pendukung bagi organisasi atau perusahaan tersebut untuk mencapai tujuan mereka.

2.2. Mobile Application

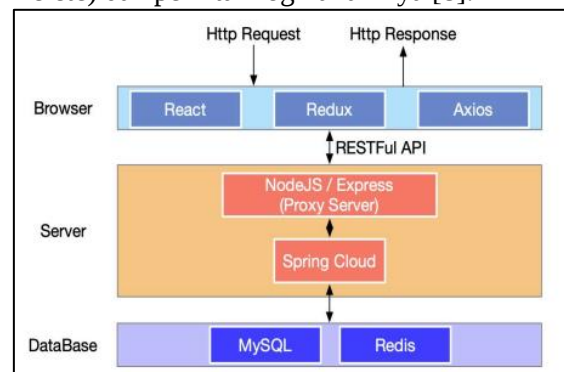
Mobile application atau aplikasi berbasis *mobile* merupakan perangkat lunak yang berjalan pada perangkat seluler seperti ponsel seluler [5]. Android merupakan sistem operasi *open-source* berbasis Linux yang dikembangkan oleh Google dan telah memiliki lisensi APACHE 2.0 yang dirancang untuk berbagai *device touch-screen* seperti *smartphone* ataupun tablet [6]. Sedangkan iOS merupakan sistem operasi seluler yang diluncurkan tahun 2007 oleh Apple yang kemudian berkembang berbagai perangkat Apple lainnya. Berbeda dengan AndroidOS, iOS tidak dilisensikan untuk dapat digunakan dan di-install di *device* selain Apple.

2.3. Framework Flutter

Framework merupakan kumpulan dari berbagai intruksi yang dikumpulkan dalam suatu *class* dan *function* dengan fungsinya masing-masing yang dapat menghindari *boilerplate code* serta menghemat waktu *development* [7]. Dari banyaknya *framework*, Flutter menjadi salah satu pilihan dalam pengembangan sistem *multi-platform*. Flutter merupakan *framework* yang banyak digunakan untuk mengembangkan aplikasi *multi-platform* seperti website, aplikasi *mobile*, dan aplikasi desktop hanya dengan satu *codebase*.

2.4. Backend-Frontend Service

Backend merupakan program yang berjalan di sisi server dan bertugas memproses *logic* yang diperoleh dari sistem *frontend* ke *database*. Sedangkan *frontend* merupakan program yang berjalan dari sisi *user* dan menghubungkan *user* melalui *interface* tempat *user* berinteraksi dengan sistem. Agar *frontend* dan *backend* dapat terhubung dengan lebih sederhana, maka dibutuhkan API (Application Programming Interface) yang berhubungan dengan perintah CRUD (Create-Read-Update-Delete) dan perintah logika lainnya [8].



Gambar 2. Request dan Response Frontend-Backend

Setelah *frontend* menerima data, *frontend* akan mengirimkan *request* ke server melalui API. Server akan menerima dan mengambil data dari *database* lalu mengembalikannya melalui API dalam bentuk *response*.

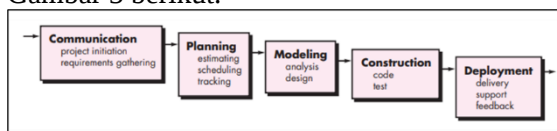
2.6. API (Application Programming Interface)

Application Programming Interface atau yang dapat disingkat sebagai API merupakan *interface* yang dibangun agar beberapa atau seluruh fungsi sistem dapat diakses secara terprogram tanpa perlu melakukan perubahan

pada struktur kode maupun pada *database*. Hal ini memudahkan komunikasi antar sistem dengan platform yang berbeda [9]. API juga dapat dianggap sebagai suatu kumpulan teknik yang jelas untuk menciptakan suatu komunikasi antar *software* dengan komponen yang berbeda dan memberikan kemudahan dalam pengembangan perangkat lunak atau aplikasi. API sendiri memiliki berbagai arsitektur seperti REST (Representational State Transfer). Melalui arsitektur REST, file Javascript Object Notation (JSON) dapat disajikan kepada aplikasi yang mengakses API tersebut.

3. METODE PENELITIAN

Metode pengembangan perangkat lunak atau SDLC (Software Development Life Cycle) yang digunakan pada penelitian ini yaitu metode Waterfall. Metode ini termasuk metode SDLC (Software Development Life Cycle) yang paling sederhana dan termasuk paradigma tertua di *software engineering* [10]. Metode ini dipilih karena memiliki alur pengembangan yang sederhana dan sesuai dengan kebutuhan proyek ini. Metode Waterfall dilakukan secara sistematis dan linear dengan tahapan *communication*, *planning*, *modelling*, *construction*, dan *deployment* seperti pada Gambar 3 berikut.



Gambar 3. Metode Waterfall [10]

Sebelum proses pengembangan *software* dimulai, tahapan *communication* atau komunikasi dengan pihak-pihak terkait termasuk hal yang penting dilakukan. Hal ini bertujuan untuk memperoleh *requirements*, fitur, dan fungsi yang dibutuhkan pada *software* yang akan dibangun. Untuk mendapatkan *requirements* yang sesuai, observasi, studi literatur, dan wawancara dengan pihak UPA TIK UPN "Veteran" Jawa Timur dilakukan.

Setelah tahapan *communication* dilakukan dan memperoleh *requirements* yang dibutuhkan, tahapan *planning* atau perencanaan dilakukan. Tahapan ini dilakukan dengan merancang *software project plan* yang mendeskripsikan tugas-tugas teknis yang akan dilakukan selama perancangan *software* berlangsung.

Pada tahapan *modelling*, pembuatan *sketch* atau sketsa dilakukan untuk memberikan gambaran atau garis besar terhadap sistem yang akan dikembangkan. Tahapan ini akan berfokus pada perancangan alur aplikasi dan Unified Modeling Language (UML).

Tahapan *construction* dilakukan dengan merancang sistem informasi berbasis *mobile* melalui *coding* dan melakukan pengujian atau *testing* pada sistem untuk menemukan *bug*, *error*, ataupun kesalahan yang terdapat pada aplikasi. Aplikasi akan disambungkan dengan backend menggunakan API dari bahasa pemrograman Golang yang telah ada sebelumnya.

Penelitian ini akan menggunakan metode *blackbox testing* yang berfokus pada detail-detail aplikasi seperti tampilan *interface* aplikasi, fungsi-fungsi yang ada pada aplikasi, serta kesesuaian alur fungsi aplikasi dengan proses bisnis yang ada.

Tahapan terakhir pada penelitian ini yaitu *deployment*. Pada tahap ini, sistem diuji menggunakan *blackbox testing* untuk mengetahui apakah sistem dapat berjalan sesuai proses bisnis sehingga *maintenance* dapat dilakukan dengan tepat. *Feedback* atau umpan balik yang diperoleh dapat diimplementasikan sebagai pengembangan aplikasi lebih lanjut.

4. HASIL DAN PEMBAHASAN

Bagian ini menjabarkan hasil dari setiap tahapan pada metodologi. Setiap tahapan menghasilkan beberapa hal yang saling berkaitan.

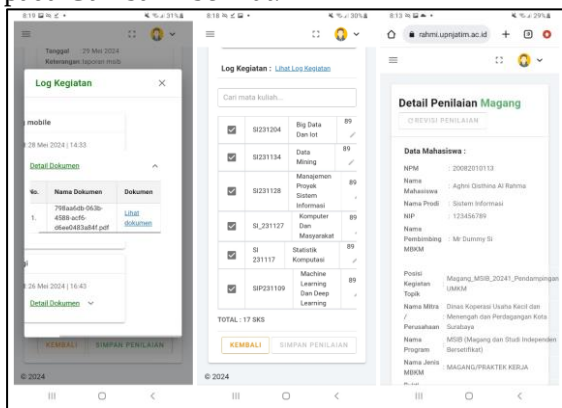
4.1. Communication

Pada tahapan ini, komunikasi dengan pihak-pihak terkait dilakukan untuk memperoleh *requirements*, fitur, dan fungsi yang dibutuhkan. Untuk mendapatkan *requirements* yang sesuai, beberapa langkah dilakukan yaitu observasi, studi literatur, dan wawancara dengan pihak UPA TIK UPN "Veteran" Jawa Timur.

4.1.1. Observasi

Penelitian ini dilakukan pada UPN "Veteran" Jawa Timur berfokus pada sistem informasi bernama "SILATURAHMI" yang memiliki kepanjangan Sistem Informasi Kolaborasi Terpadu Universitas, Masyarakat, dan Industri. Sistem informasi ini dirancang untuk mengelola program MBKM agar lebih

efektif. Namun, sistem informasi ini masih berbasis website yang dioptimasi untuk diakses oleh *device* berbasis desktop. Pengaksesan sistem informasi pada *device* berbasis *mobile* sendiri masih kurang responsif. Pada website dengan mode *mobile*, kolom tabel saling bertumpuk dan tidak rapi. Paragraf panjang saling bertumpuk atau melebihi panjang *device* sehingga sulit untuk dibaca. Data *logbook* dan luaran ditampilkan bergantung pada tanggal yang dipilih sehingga pengguna kesulitan dalam melihat data secara keseluruhan seperti pada Gambar 4 berikut.



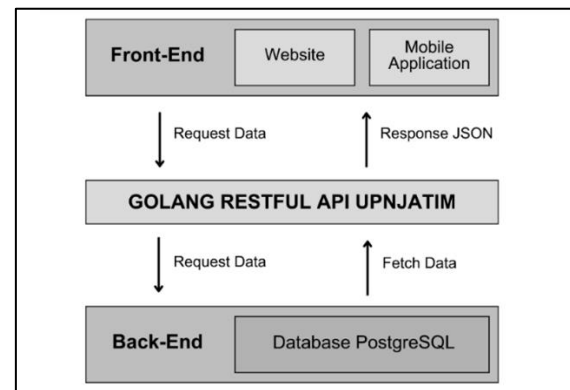
Gambar 4. SILATURAHMI berbasis Website pada *Device Mobile*

4.1.2. Studi Literatur

Studi literatur dilakukan melalui pengumpulan dasar teori yang mendukung topik penelitian. Literatur diperoleh dari artikel, jurnal, buku, dan skripsi yang telah dilaksanakan sebelumnya. Temuan-temuan dari penelitian terdahulu diperoleh dan dilakukan *review* sehingga dapat menjadi dasar serta pedoman dari penelitian ini.

4.1.3. Wawancara

Berdasarkan hasil wawancara dengan Kepala Bagian UPA Teknologi Informasi Komunikasi, diperoleh beberapa informasi terkait sistem informasi SILATURAHMI yaitu alur pada sistem informasi SILATURAHMI berbasis website dan arsitektur yang digunakan oleh pihak UPA TIK.



Gambar 5. Arsitektur Sistem

Arsitektur yang digunakan pada sistem informasi SILATURAHMI ini yaitu arsitektur *frontend-backend*. Pengguna akan berinteraksi melalui *frontend* dan *request data* akan dikirimkan ke bagian *backend* melalui API milik UPN "Veteran" Jawa Timur yang telah tersedia. UPN "Veteran" Jawa Timur menggunakan *database PostgreSQL*. API akan mengambil data yang sesuai dari *database* tersebut dan mengembalikannya ke *frontend* dalam bentuk *response JSON*.

4.2. Planning

Berdasarkan hasil dari tahapan komunikasi, sistem informasi SILATURAHMI dirancang untuk meningkatkan efektivitas pengelolaan konversi MBKM di tingkat universitas. Namun, sistem ini dirancang untuk dijalankan di *browser* internet dan hanya dioptimasi untuk perangkat berbasis desktop. Akibatnya, penggunaan sistem pada perangkat berbasis *mobile* masih belum maksimal baik dari segi tampilannya yang kurang responsif maupun dari segi performanya.

Dari tahapan tersebut, analisis perencanaan dari perancangan sistem informasi SILATURAHMI berbasis *mobile* dilakukan. Pada tahapan *construction*, dibutuhkan sumber daya berupa API UPNJATIM sebagai "jembatan" antara *request-response* dari user dengan bagian *backend* dan dilanjutkan dengan tahapan *deployment*. *Output* yang diharapkan dari keseluruhan tahapan ini yaitu aplikasi sistem informasi SILATURAHMI UPN "Veteran" Jawa Timur berbasis *mobile*. Perbandingan antara penggunaan sistem berbasis website dengan sistem berbasis *mobile* seperti yang terlihat pada Tabel 1 berikut.

Tabel 1. Perbandingan Sistem Informasi SILATURAHMI berbasis Website dan *Mobile*

SILATURAHMI berbasis Website	SILATURAHMI berbasis Mobile
Seluruh <i>role</i> dapat mengakses seluruh fitur dalam sistem.	<i>Role</i> berfokus pada <i>user</i> utama program MSIB / Magang Mandiri dengan penyesuaian fitur.
Pada mode <i>mobile</i> , kolom-kolom pada tabel saling bertumpuk sehingga tidak rapi.	Terdapat beberapa informasi yang ditampilkan dalam bentuk <i>list card</i> sehingga lebih rapi dan lebih mudah dibaca.
Pada mode <i>mobile</i> , tulisan yang terlalu panjang menjadi saling bertumpuk atau melebihi batas sehingga sulit untuk dibaca.	Panjang tulisan menyesuaikan ukuran <i>device</i> .
Ukuran gambar tidak dapat menyesuaikan ukuran <i>device</i> .	Ukuran gambar menyesuaikan ukuran <i>device</i> .
Data <i>logbook</i> dan luaran ditampilkan berdasarkan tanggal pengunggahan.	Data <i>logbook</i> dapat tampil secara keseluruhan.

Berdasarkan *flowchart* yang telah dibuat, terdapat enam *role* yang akan menggunakan aplikasi ini yaitu:

1. Mahasiswa UPN "Veteran" Jawa Timur yang dapat mendaftar program MSIB atau Mandiri yang mereka ikuti, menyusun paket konversi, melaporkan kegiatan yang mereka ikuti secara rutin melalui *logbook*, dan mengunggah luaran serta nilai akhir yang mereka dapatkan.
2. Pembimbing MBKM yang dapat mengedit dan menyetujui paket konversi yang diajukan mahasiswa dan melakukan verifikasi nilai yang diterima mahasiswa.
3. Koordinator program studi yang dapat melakukan *plotting* terhadap pembimbing MBKM.
4. Tim MBKM yang dapat menyetujui program MSIB atau Mandiri mahasiswa dan membantu tugas koordinator program studi dalam *plotting* pembimbing MBKM.
5. Tim penilai konversi yang dapat memberikan nilai pada mata kuliah yang

dikonversi dan mengajukan revisi penilaian.

6. PIC MBKM Universitas yang dapat memberikan persetujuan terhadap ajuan revisi penilaian yang diajukan oleh tim penilai konversi.

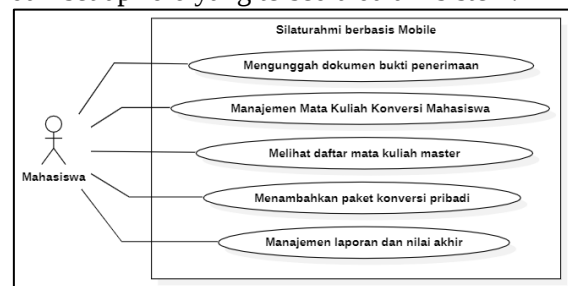
Namun, penelitian ini akan berfokus pada *role* mahasiswa.

4.3. Modelling

Pada tahapan ini, pembuatan *sketch* atau sketsa dilakukan untuk memberikan gambaran atau garis besar terhadap sistem yang akan dikembangkan melalui perancangan Unified Modelling Language (UML).

4.3.1. Use-Case Diagram

Diagram *use-case* menggambarkan interaksi dari setiap *role* yang tersedia dalam sistem.

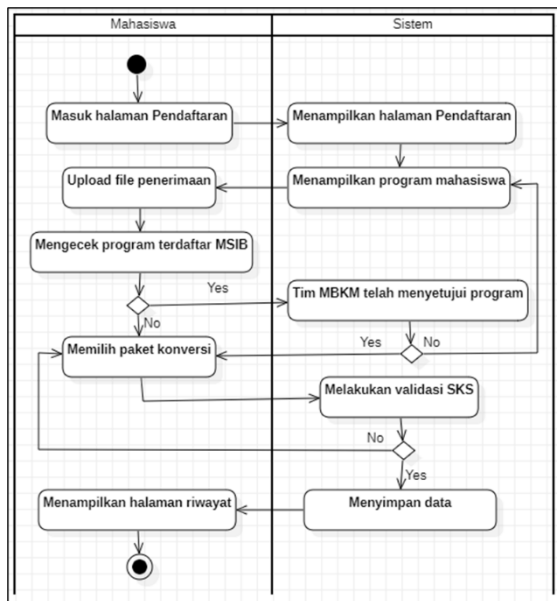


Gambar 6. Diagram Use-Case

Diagram pada Gambar 6 menggambarkan sebagian dari hubungan interaksi antara *role* mahasiswa dengan sistem. Seluruh fitur pada sistem ini hanya bisa diakses bila *user* telah melakukan *login*. Sistem informasi SILATURAHMI telah terintegrasi dengan sistem informasi akademik atau SIAMIK sehingga pengguna tidak perlu melakukan *sign up* atau *register* untuk bisa *login*. Mahasiswa hanya bisa mengakses data paket konversi, luaran, dan nilai akhir milik dirinya sendiri.

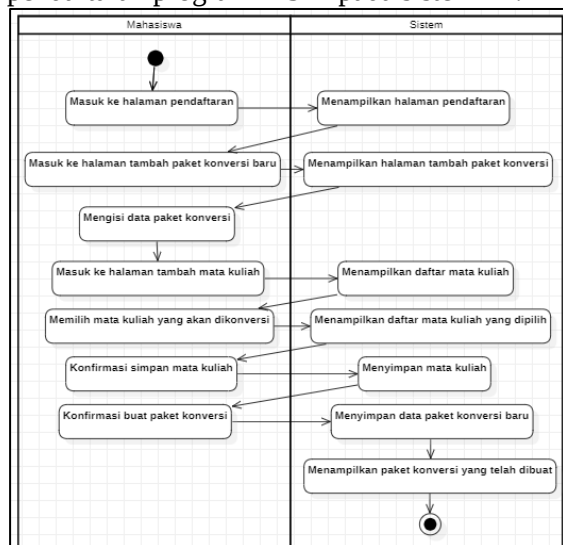
4.3.2. Activity Diagram

Activity Diagram memberikan gambaran aktivitas berupa aliran kerja (*workflow*) dari suatu sistem dimana dalam diagram di bawah ini menggunakan sistem informasi SILATURAHMI berbasis *mobile*.



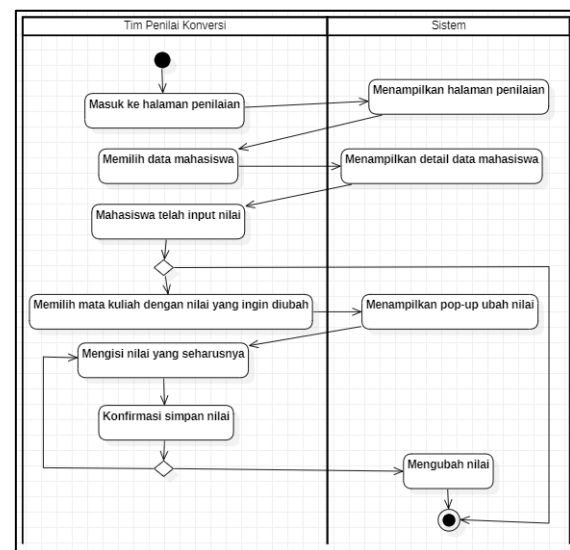
Gambar 7. Activity Diagram Unggah Dokumen Pendaftaran

Gambar 7 menggambarkan activity diagram dari fitur pengunggahan dokumen pendaftaran yang menjadi tahapan dari pendaftaran program MSIB pada sistem ini.



Gambar 8. Activity Diagram Membuat Paket Konversi

Activity diagram perancangan paket konversi tergambar pada Gambar 8 yang dimulai dari menampilkan halaman pendaftaran dan memilih *button* tambah paket konversi.

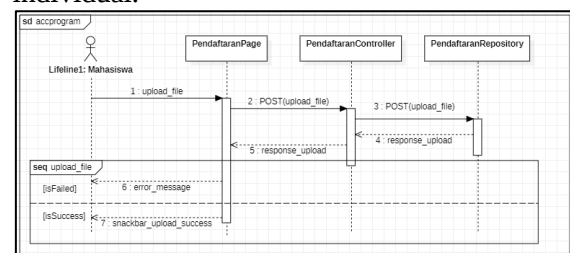


Gambar 9. Activity Diagram Manajemen Nilai Akhir

Activity diagram manajemen nilai akhir tergambar pada Gambar 9 yang dimulai dari menampilkan halaman penilaian dan memilih data mahasiswa yang ingin ditampilkan.

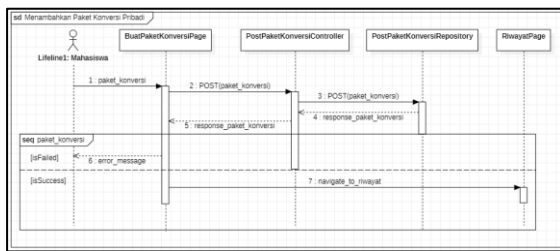
4.3.3. Sequence Diagram

Sequence diagram menggambarkan bagaimana suatu operasi dilakukan melalui pesan atau *message* yang dikirimkan dan waktu pesan atau operasi tersebut dilakukan [11]. Melalui diagram ini, gambaran interaksi dari setiap objek dalam sistem dapat diketahui dengan lebih jelas yang tergambar dalam bentuk dua dimensi. Dimensi vertikal menunjukkan poros waktu sedangkan dimensi horizontal merepresentasikan objek-objek individual.



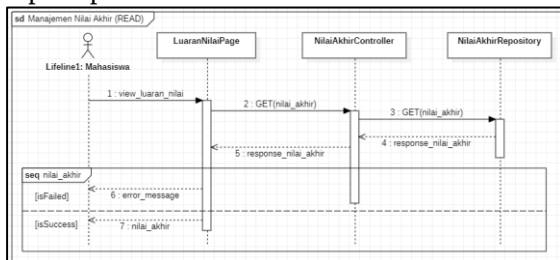
Gambar 10. Sequence Diagram Unggah Dokumen Pendaftaran

Sequence diagram unggah dokumen pendaftaran yang dilakukan oleh aktor mahasiswa tergambar seperti pada Gambar 10.



Gambar 11. Sequence Diagram Membuat Paket Konversi

Sequence diagram paket konversi yang dilakukan oleh aktor mahasiswa tergambar seperti pada Gambar 11.

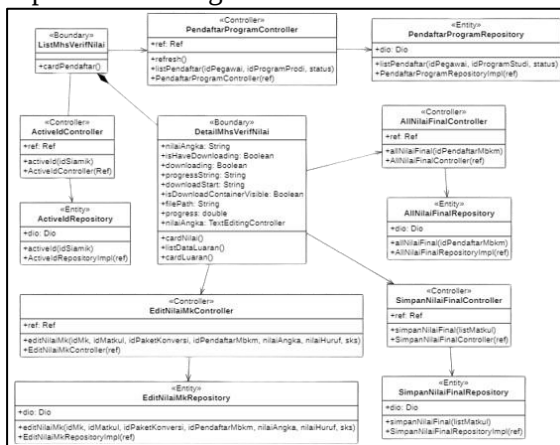


Gambar 12. Sequence Diagram Manajemen Nilai Akhir.

Sequence diagram manajemen nilai akhir yang dilakukan oleh aktor mahasiswa tergambar seperti pada Gambar 12.

4.3.4. Class Diagram

Class diagram memberikan gambaran dari struktur dan hubungan antar entitas dalam sistem [12]. Melalui diagram ini, hubungan antar *class* dan variabel serta *object* di dalamnya dapat terlihat dengan lebih baik.



Gambar 13. Partial Class Diagram Sistem Informasi SILATURAHMI Berbasis Mobile

Gambar 13 menggambarkan sebagian dari *class* diagram sistem informasi SILATURAHMI berbasis *mobile* dengan *pattern* EBC (Entity-Boundary-Controller). Class diagram utuh memiliki lebih dari 50 *class* yang saling berkaitan satu sama lain dengan

stereotype boundary, *controller*, dan *entity*.

4.4. Construction

Sequence diagram manajemen nilai akhir Bagian ini berisi potongan-potongan kode dari sistem informasi SILATURAHMI berbasis *mobile* menggunakan *framework* Flutter dan bahasa pemrograman Dart.

```
static String baseUrl = "http://xxx.xxx.xxx:xxxx/xx/";
static String loginURL = "user/login";
static String validasiLoginURL = "user/validasi-login";

//MAMAS1288
static String registerURL = "mangang/pendaftar-program-sbkn/mahasiswa";
static String activeURL = "mangang/pendaftar-program-sbkn/cek-pendaftar-mangang";
static String paketMatkulAwalURL = "mangang/matkul-paket-konversi-terpilih-awal/paket-by-pendaftar-sbkn/";
static String matkulAwalURL = "mangang/matkul-paket-konversi-terpilih-awal-by-pendaftar-sbkn/";
```

Gambar 14. Potongan Kode Endpoints

Gambar 14 menunjukkan potongan kode dari *endpoint* yang digunakan dimana variabel *baseUrl* berisi *base URL* dari API yang digunakan dan variabel-variabel lainnya berisi *endpoint* yang digunakan.

```
//Input nama paket
padding: EdgeInsets.only(bottom: 20),
child: TextField(
  textAlignVertical: TextAlignVertical.center,
  style: TextStyle(fontSize: 14),
  onChanged: (String nama) {
    setState(() {
      namaPaket = nama;
    });
  },
  decoration: const InputDecoration(...), // InputDecoration
), // TextField
), // Padding
```

Gambar 15. Potongan Kode View

Kode yang ditunjukkan pada Gambar 15 merepresentasikan *view* yang menjadi UI atau tampilan dari fitur menambahkan paket konversi pribadi yang dilihat oleh *user*.

```
Future<Object> postMatkulAwal([required PostMatkulAwalRequest data]) async {
  state = const AsyncLoading();

  final response = await ref.read(postMatkulAwalRepositoryProvider).postMatkulAwal(data);
  if (response is ErrorResponse) {
    return response.error;
  } else if (response is PostMatkulAwalResponse) {
    return response;
  } else {
    return response;
  }
}
```

Gambar 16. Potongan Kode Controller

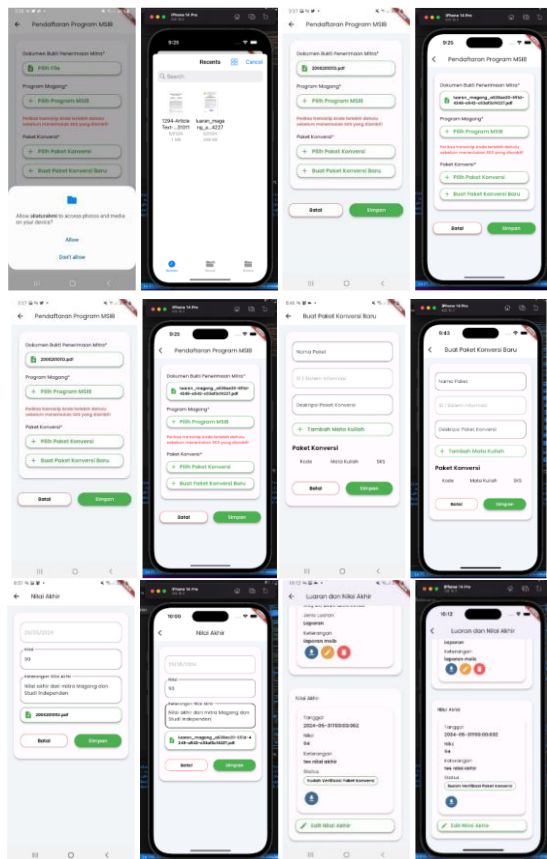
Kode yang ditunjukkan pada Gambar 16 merepresentasikan *controller* yang menjadi penghubung antara *view* dengan *repository*.

```
@override
Future<dynamic> postMatkulAwal(PostMatkulAwalRequest data) async {
  try {
    final response = await dio.post(endpoints.postMatkulAwalURL, data: data.toJson());
    return PostMatkulAwalResponse.fromJson(response.data);
  } on DioException catch (ex) {
    return ErrorResponse.fromJson(ex.response?.data);
  }
}
```

Gambar 17. Potongan Kode Repository

Kode yang ditunjukkan pada Gambar 17 merepresentasikan *repository* yang akan terhubung ke API dengan mengirim *request* dan menerima *response*.

Kode-kode tersebut menghasilkan tampilan yang telah fungsional seperti yang terlihat pada Gambar 18.



Gambar 18. Tampilan pada AndroidOS dan iOS

4.5. Deployment

Berikut ini hasil *blackbox testing* keseluruhan fitur dengan kesimpulan bahwa sistem telah sesuai dengan alur proses bisnis yang diharapkan seperti yang ditunjukkan pada Tabel 2.

Tabel 2. Hasil *BlackBox Testing*

Fitur Unggah Dokumen Bukti Penerimaan	
Skenario	Mahasiswa menekan <i>button</i> unggah dokumen bukti penerimaan
Hasil yang Diharapkan	Sistem menampilkan <i>file</i> yang tersedia pada <i>device</i> dan menampilkan nama <i>file</i> yang dipilih mahasiswa.
Hasil Sebenarnya	Sistem menampilkan <i>file</i> yang tersedia pada <i>device</i> dan menampilkan nama <i>file</i> yang dipilih mahasiswa.
Kesimpulan	Sesuai
Fitur Manajemen Paket Konversi	

Skenario	Mahasiswa menekan <i>button</i> "Buat Paket Konversi".
Hasil yang Diharapkan	Sistem menampilkan <i>page</i> membuat paket konversi baru yang akan tersimpan sebagai paket konversi mahasiswa.
Hasil Sebenarnya	Sistem menampilkan <i>page</i> membuat paket konversi baru yang akan tersimpan sebagai paket konversi mahasiswa.
Kesimpulan	Sesuai
Fitur Manajemen Nilai Akhir	
Skenario	Mahasiswa menekan <i>button</i> "Tambah atau Edit Nilai Akhir".
Hasil yang Diharapkan	Sistem menampilkan halaman tambah nilai dan menyimpan data nilai setelah mahasiswa menekan <i>button</i> simpan nilai.
Hasil Sebenarnya	Sistem menampilkan halaman tambah nilai dan menyimpan data nilai setelah mahasiswa menekan <i>button</i> simpan nilai.
Kesimpulan	Sesuai

Tahapan *deployment* menjadi tahapan akhir dari perancangan sistem ini. Aplikasi SILATURAHMI yang telah berhasil dibangun akan didistribusikan untuk digunakan oleh mahasiswa. Distribusi aplikasi berbasis Android dapat dilakukan dengan cara mengunggah aplikasi ke Play Store atau menyebarkan instalasi apk melalui platform lain. Namun, pengunggahan aplikasi melalui Play Store membutuhkan akun *developer* Google Play senilai 25 USD [13]. Sedangkan, pendistribusian aplikasi berbasis iOS membutuhkan akun *developer* sebesar 99 USD per tahun [14].

5. KESIMPULAN

Kesimpulan dapat ditarik dari hasil penelitian dan perancangan sistematis sebagai berikut.

- Sistem informasi berbasis mobile ini dikembangkan dengan menerapkan metode Waterfall yang terdiri dari tahap *communication*, *planning*, *modelling*, *construction*, dan *deployment*.

- b. Tahapan *deployment* dilakukan dengan merilis aplikasi Android dalam format .apk dan pengujian aplikasi iOS dalam emulator.
- c. Dengan adanya Sistem Informasi SILATURAHMI berbasis *mobile*, sistem ini diharapkan dapat membantu pengguna dalam mengakses sistem melalui *device* berbasis *mobile*.
- d. Berdasarkan hasil *testing*, Sistem Informasi SILATURAHMI berbasis *mobile* telah berhasil dijalankan di kedua sistem operasi yaitu AndroidOS dan iOS serta memenuhi proses bisnis yang telah direncanakan.

Dari penelitian ini, terdapat beberapa rekomendasi sebagai salah satu bahan untuk pengembangan lebih lanjut.

- a. Melakukan *release* aplikasi Android ke PlayStore dan aplikasi iOS ke AppStore sebagai langkah lanjutan dari tahapan *deployment*.
- b. Pada *backend* sistem, menambahkan fitur pembatasan SKS yang terhubung dengan sistem informasi akademik universitas dan pembatasan program MBKM sesuai peraturan yang ada.

UCAPAN TERIMA KASIH

Penulis menyadari tanpa bantuan dan bimbingan dari berbagai pihak sehingga penulis mengucapkan terima kasih yang sebesar-besarnya kepada pihak UPA TIK UPN "Veteran" Jawa Timur serta seluruh pihak terkait yang telah memberikan dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Humas UPN, "Luncurkan Platform SILATURAHMI, Kuatkan Sinergi Universitas dengan Pemerintahan dan Dunia Industri," 2023. <https://www.upnjatim.ac.id/luncurkan-platform-silaturahmi-kuatkan-sinergi-universitas-dengan-pemerintahan-dan-dunia-industri/> (accessed Oct. 23, 2023).
- [2] A. Bjørn-hansen, T. Grønli, G. Ghinea, and S. Alouneh, "An Empirical Study of Cross-Platform Mobile Development in Industry," *Wirel. Commun. Mob. Comput.*, vol. 2019, pp. 1–12, 2019, doi: 10.1155/2019/5743892.
- [3] S. Santoso, D. J. Surjawan, and E. D. Handoyo, "Pengembangan Sistem Informasi Tukar Barang Untuk Pemanfaatan Barang Tidak Terpakai dengan Flutter Framework," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 3, pp. 589–598, 2020, doi: <https://doi.org/10.28932/jutisi.v6i3.3071>.
- [4] D. Mota and R. Martinho, "An Approach to Assess the Performance of Mobile Applications: A Case Study of Multiplatform Development Frameworks," no. Enase, pp. 150–157, 2021, doi: 10.5220/0010497401500157.
- [5] C. G. Thomas and A. J. Devi, "A Study and Overview of the Mobile App Development Industry," *Int. J. Appl. Eng. Manag. Lett.*, vol. 5, no. 1, pp. 115–130, 2021, doi: <http://doi.org/10.5281/zenodo.4966320>.
- [6] K. A. B. Ibrahim and D. Gustina, "Rancang Bangun Aplikasi Berbasis Android untuk Brand Clothing Sand Beach dengan Skema Diskon Menggunakan Hungarian Algorithm," *J. Sist. Inf.*, vol. 8, no. 1, pp. 47–56, 2021, doi: <https://doi.org/10.35968/jsi.v8i1.608>.
- [7] A. F. Sallaby and I. Kanedi, "Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter," *J. Media Infotama*, vol. 16, no. 1, pp. 48–53, 2020, doi: <https://doi.org/10.37676/jmi.v16i1.1121>.
- [8] Y. Gong, F. Gu, K. Chen, and F. Wang, "The Architecture of Micro-services and the Separation of Front-end and Back-end Applied in a Campus Information System," *2020 IEEE Int. Conf. Adv. Electr. Eng. Comput. Appl.*, pp. 321–324, 2020, doi: 10.1109/AEECA49918.2020.9213662.
- [9] D. Wijonarko and B. W. R. Mulya, "Pengembangan Antarmuka Pemrograman Aplikasi Menggunakan Metode RESTful pada Sistem Informasi Akademik Politeknik Kota Malang," *SMATIKA J. STIKI Inform. J.*, vol. 8, no. 2, pp. 63–66, 2018, doi: <https://doi.org/10.32664/smatika.v8i02.202>.
- [10] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th Edition. New York: McGraw-Hill, 2010.
- [11] M. Nugraha, L. Sakinah, R. A. Setiawan, And H. Mulyani, "Rancang Bangun Sistem Informasi Penerimaan Mahasiswa Baru Berbasis Web Dengan Menggunakan Framework Laravel," *Jitet (Jurnal Inform. Dan Tek. Elektro Ter.)*, Vol. 12, No. 2, Pp. 1162–1169, 2024, Doi: <http://Dx.Doi.Org/10.23960/Jitet.V12i2.4179>.
- [12] G. Purnama, "Perancangan Sistem Informasi Permintaan Barang Dengan Prosedur Lelang Berbasis Metode Perancangan Uml: Studi Kasus Undira," *JITET (Jurnal Inform. dan Tek. Elektro Ter.)*, vol. 12, no. 2, pp. 1199–1209, 2024, doi: <http://dx.doi.org/10.23960/jitet.v12i2.4162>.
- [13] Google, "GooglePlayConsole," 2024. <https://play.google.com/intl/id/console/about/>

- (accessed Jun. 09, 2024).
- [14] Apple Inc., “Apple Developer Program Enrollment,” 2024.
<https://developer.apple.com/enroll/purchase>
(accessed Jun. 09, 2024).