

Rancang Bangun Aplikasi Sistem Monitoring Skripsi Mahasiswa dengan Metode RAD (*Rapid Application Development*) Universitas Islam Malang

Imam Rosadi ^a, Fawaidul Badri ^b, Ngatmari^c

^{a,b,c} Teknik Elektro, Universitas Islam Malang, Malang, Indonesia

email: ^aytdeyyy@gmail.com, ^bfawaidulbadri@unisma.ac.id*, ^cmarymalangcity@gmail.com

* Corresponding Author

INFORMASI ARTIKEL

Sejarah artikel:
Accepted 30 November 2023

Kata kunci:
Sistem Monitoring Skripsi,
Aplikasi,
Usability fitur aplikasi,
Android,
Universitas Islam Malang.

A B S T R A K

Dalam pendidikan tingkat perguruan tinggi, Skripsi merupakan karya tulis ilmiah yang wajib dikerjakan oleh setiap mahasiswa Fakultas Teknik Universitas Islam Malang. Pada umumnya dalam proses penyusunan skripsi didahului dengan penulisan proposal skripsi, penulisan artikel ilmiah untuk seminar hasil dan diakhiri dengan ujian skripsi. Permasalahan yang sering dihadapi mahasiswa dalam menyelesaikan skripsi adalah pengelolaan waktu yang kurang baik, pemanfaatan waktu yang tidak efektif, rendahnya motivasi berprestasi dan ketidakdisiplinan merupakan penyebab terhambatnya skripsi mahasiswa. Tujuan penelitian ini adalah untuk dapat menghasilkan sebuah aplikasi "Sistem Monitoring Skripsi" sebagai penunjang sarana pembelajaran dan akademik di bidang skripsi mahasiswa Teknik elektro Universitas Islam Malang. Metode yang di pakai dalam pembuatan aplikasi ini adalah RAD (*Rapid Application Development*) karena dirasa efektif untuk menyelesaikan berbagai permasalahan pada pembuatan aplikasi, Hasil dari penelitian ini adalah efektifitas dan presentase kelayakan pada aplikasi yang di buat mencapai 91%(sangat layak) dengan menggunakan pengujian usability fungsional system aplikasi, Dari penelitian di simpulkan hasil rancangan aplikasi menghasilkan sebuah antarmuka Aplikasi yang intuitif dan responsive dengan kualitas yang sangat baik.

1. Pendahuluan

Dalam pendidikan tingkat perguruan tinggi, Skripsi merupakan karya tulis ilmiah yang wajib dikerjakan oleh setiap mahasiswa. Mahasiswa menyusun skripsi pada akhir kegiatan studi mereka[1]. Pada umumnya dalam proses penyusunan skripsi didahului dengan penulisan proposal skripsi, penulisan artikel ilmiah untuk seminar hasil dan diakhiri dengan ujian skripsi.

Permasalahan yang sering dihadapi mahasiswa dalam menyelesaikan skripsi adalah pengelolaan waktu yang kurang baik. Pemanfaatan waktu yang tidak efektif, rendahnya motivasi berprestasi dan ketidakdisiplinan merupakan penyebab terhambatnya skripsi mahasiswa[2]. Penelitian menyebutkan bahwa tidak semua mahasiswa mampu menyelesaikan skripsi tepat waktu[3].

Untuk memenuhi persyaratan tersebut, peneliti mengembangkan desain program aplikasi Sistem monitoring skripsi mahasiswa yang dibuat dengan *Android Studio* dapat digunakan sebagai media bagi dosen pembimbing dan sekretaris program studi untuk memantau perkembangan mahasiswa yang dibimbingnya dengan mudah, media dibuat dalam bentuk aplikasi yang di kontrol dengan php yang terhubung dengan *API Service* dengan menggunakan *Android Studio* yang memiliki beberapa menu yang sangat berguna yang ditulis menggunakan bahasa pemrograman *Java*[4]. Pemrograman *Java* digunakan karena bersifat multiplatform, fleksibel, dapat dijalankan pada berbagai platform sistem operasi komputer, dan mudah dikembangkan[5].

Penelitian ini membuat sistem untuk memantau aplikasi skripsi mahasiswa dengan menggunakan teknik *Rapid Application Development* (RAD) dan berbentuk sebuah aplikasi untuk *android device*.

Berdasarkan tinjauan pustaka yang di ambil dari penelitian sebelumnya terkait penelitian yang di lakukan sekarang di simpulkan beberapa perbedaaan spesifik salah satunya dari penelitian Nadia Mustika Sari, Lomo Mula Tua, Erly Krisnanik dengan judul Sistem Informasi Monitoring Pembimbingan Skripsi/Tugas Akhir (SIMP-S/TA) Berbasis Android[6] Pada penelitian tersebut, peneliti membuat sebuah aplikasi dengan Bahasa pemrograman *java* dan *API service* yang di control lewat *php* tetapi pada penelitian ini di di temukan perbedaan spesifik seperti proses Pembuatan aplikasi dengan menggabungkan sistem informasi yang pernah di buat dalam lingkup universitas, Metode dalam penelitian menggunakan metode *waterfall* jadi tidak memerlukan seorang *client* dalam proses pembuatannya, pencatatan *deadline* tidak di atur didalam penelitian ini dan Tidak disertai dengan pengumpulan file ke *repository* program studi Sedangkan pada penelitian yang sekarang proses Pembuatan aplikasi dilakukan secara mandiri tanpa di gabungkan dengan sistem informasi di lingkup universitas. Metode dalam penelitian ini menggunakan RAD (*Rapid Application Development*) dengan sekretaris program studi sebagai *client* dalam pembuatan sebuah aplikasi, *deadline* dari revisi di tentukan oleh dosen pembimbing setelah proses bimbingan, Disertai dengan tombol pengumpulan ke *repository* program studi.

Arif Syam, Asniati, Wa Tita dengan judul penelitiannya yaitu Penerapan sistem informasi pada layanan pengajuan dan monitoring judul tugas akhir mahasiswa[7]. Perbedaan spesifik Pada penelitian ini adalah Proses Pembuatan sistem berbasis *website*. Dalam penelitian ini hanya berfokus pada pengajuan dan monitoring judul tugas akhir saja dan metode penelitian menggunakan metode *waterfall* sedangkan pada penelitian sekarang Proses pembuatan sistem berbasis aplikasi android. Dalam penelitian sekarang berfokus pada monitoring proses bimbingan skripsi dari awal sampai akhir. Metode penelitian menggunakan RAD dengan sekretaris program studi sebagai *client* dalam pembuatan sebuah aplikasi.

Dinar Ajeng Kristiyanti, Ahmad Mulyana dengan penelitiannya yang berjudul Sistem Informasi Monitoring Skripsi Berbasis *Web* (Studi Kasus: Prodi Akuntansi Universitas Mercu Buana)[8].perbedaan pada penelitian ini adalah Proses Pembuatan sistem berbasis *website*. Pencatatan *deadline* tidak di atur didalam penelitian ini Tidak disertai dengan pengumpulan file ke *repository* program studi Proses pembuatan sistem berbasis aplikasi android. Sedangkan pada penelitian sekarang *Deadline* dari revisi di tentukan oleh dosen pembimbing setelah proses bimbingan. Disertai dengan tombol pengumpulan ke *repository* program studi.

Neva Satyahadewi, Nurul Mutiah dalam penelitiannya yang berjudul Sistem Informasi monitoring tugas akhir (SIMTA) berbasis *web* fakultas MIPA Universitas Tanjungpura[9]. Perbedaannya terletak pada Proses Pembuatan sistem berbasis *website*. Metode dalam penelitian menggunakan metode *waterfall* jadi tidak memerlukan seorang *client* dalam proses pembuatannya. Pencatatan *deadline* tidak di atur didalam penelitian ini Tidak disertai dengan pengumpulan file ke *repository* program studi, Sedangkan pada penelitian yang sekarang Proses pembuatan sistem berbasis aplikasi android. Metode dalam penelitian ini menggunakan RAD dengan sekretaris program studi sebagai *client* dalam pembuatan sebuah aplikasi. *Deadline* dari revisi di tentukan oleh dosen pembimbing setelah proses bimbingan. Disertai dengan tombol pengumpulan ke *repository* program studi.

Arman Suryadi Karim yang berjudul Sistem Informasi Tugas Akhir Online Berbasis *Web*[10]. Perbedaannya terletak pada Proses Pembuatan sistem berbasis *website*. *website* yang dibuat di gabungkan dengan sistem yang telah dibuat sebelumnya dan berfokus langsung pada BAAK untuk menambahkan *user* dalam sebuah *website* Metode dalam penelitian menggunakan metode *waterfall* jadi tidak memerlukan seorang *client* dalam proses pembuatannya. Pencatatan *deadline* tidak di atur didalam penelitian ini Tidak disertai dengan pengumpulan file ke *repository* program studi. Sedangkan pada penelitian sekarang Proses pembuatan sistem berbasis aplikasi android. Proses Pembuatan aplikasi dilakukan secara mandiri tanpa di gabungkan dengan sistem informasi di lingkup universitas dan berfokus pada Sekretaris program studi sebagai *administrator* untuk menambahkan *user* dalam aplikasi. Metode dalam penelitian ini menggunakan RAD dengan sekretaris program studi sebagai *client* dalam

pembuatan sebuah aplikasi. *Deadline* dari revisi di tentukan oleh dosen pembimbing setelah proses bimbingan. Disertai dengan tombol pengumpulan ke *repository* program studi.

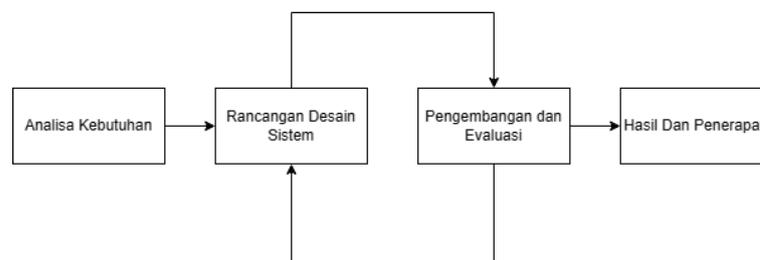
2. Metode

Tabel 1. Tabel Rancangan Fitur

No.	Nama Fitur	
1.	<i>Login</i>	
2.	Menu Admin	<ul style="list-style-type: none"> • Menambahkan akun dosen pembimbing skripsi. • Menambahkan akun mahasiswa. • Mengubah dan menghapus data dosen pembimbing. • Mengubah dan menghapus data mahasiswa. • Melihat seluruh data mahasiswa yang sedang telah ditambahkan. • Melihat seluruh data dosen pembimbing.
3.	Menu Dosen	<ul style="list-style-type: none"> • Melihat seluruh mahasiswa yang sedang dibimbing. • Mengakses data <i>history</i> atau catatan bimbingan mahasiswa. • Menambahkan <i>history</i> atau catatan bimbingan mahasiswa. • Mengatur <i>deadline</i> bimbingan selanjutnya.
4.	Menu Mahasiswa	<ul style="list-style-type: none"> • Mengakses <i>history</i> bimbingan yang telah di tambahkan oleh dosen pembimbing • Melihat <i>deadline</i> yang telah di tambahkan dosen pembimbing sebelumnya.

Dari analisa rancangan fitur yang telah di tentukan di atas, digunakan suatu metode yang dikenal sebagai *Rapid Application Development*. Keunggulan dari metode RAD adalah kemampuannya untuk menghasilkan produk berkualitas dengan cepat, sehingga menghemat waktu dalam pembangunan sistem. Penggunaan metode RAD juga membantu dalam menetapkan batasan-batasan pada sistem, sehingga perubahan besar dapat dihindari[11]. *Rapid Application Development* (RAD) adalah suatu model proses dengan siklus pengembangan yang singkat. RAD menggabungkan berbagai teknik terstruktur dengan teknik *prototyping* dan *joint application* untuk mempercepat proses pengembangan. Dengan menggunakan metode RAD, suatu sistem dapat dibangun dalam rentang waktu singkat, sekitar 30-90 hari[12].

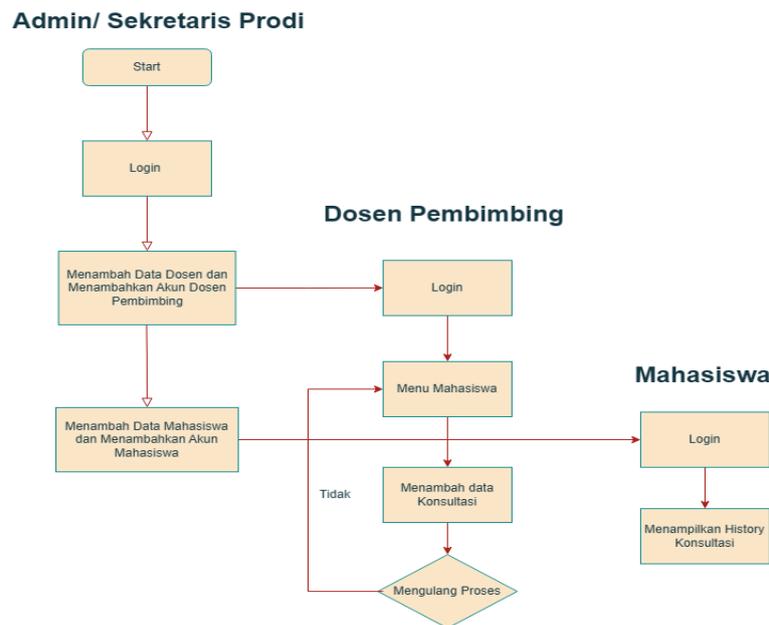
1. Metode RAD (*Rapid Application Development*)



Gambar 1. Model *Rapid Application Development*

Dalam Sistem Monitoring skripsi ini terdapat tiga aktor, yaitu Admin, Dosen Pembimbing, Dan Mahasiswa Ketiga Aktor ini saling berkaitan satu sama lain, Admin diharuskan melakukan *login* dengan benar dan bertugas untuk mendaftarkan dosen pembimbing dan mahasiswa yang bersangkutan untuk selanjutnya menyerahkan akun kepada dosen pembimbing. Jika sudah punya akun, Dosen Pembimbing dapat melakukan login dengan benar. Dosen bisa melihat informasi yang berupa *History* waktu konsultasi dan Mahasiswa yang di Bimbing. Ketika Mahasiswa selesai melakukan konsultasi data *history* konsultasi akan di tampilkan sesuai Bab dan catatan khusus dari

dosen pembimbing, dan akan di tampilkan ketika Mahasiswa melakukan login ke dalam aplikasi, sehingga rancangan *flowchart* yang di pakai sebagai berikut :



Gambar 2. Flowchart sistem

2. Metode Perhitungan Skala *Likert*

Skala *likert* adalah jenis skala penilaian yang digunakan untuk mengukur pandangan, pendapat, atau persepsi seseorang terhadap suatu subjek atau pernyataan tertentu secara kuantitatif. Melibatkan penggunaan pertanyaan atau pernyataan yang dimaksudkan untuk mengukur tingkat persetujuan, kepuasan, keyakinan, atau tingkat suatu variabel tertentu[13].

Tabel 2. Skala Likert

Jawaban	Skor
$X_i + 1.80 S_{Bi} < X$	Sangat Layak
$X_i + 0.6 S_{Bi} < X \leq X_i + 1.80 S_{Bi}$	Layak
$X_i - 0.6 S_{Bi} < X \leq X_i + 0.60 S_{Bi}$	Cukup Layak
$X_i - 1.80 S_{Bi} < X \leq X_i - 0.60 S_{Bi}$	Kurang Layak
$X \leq X_i - 1.80 S_{Bi}$	Sangat Kurang Layak

Tabel 2 merupakan skala likert yang digunakan untuk menilai tingkat kelayakan suatu variabel atau *item* berdasarkan skor yang diberikan (X). Dalam tabel ini, terdapat lima kategori penilaian. Apabila skor (X) lebih tinggi dari nilai mean (X_i) ditambah 1.80 kali standar deviasi (S_{Bi}), maka variabel atau item tersebut dinilai "Sangat Layak." Lalu jika skor berada di antara $X_i + 0.6 S_{Bi}$ dan $X_i + 1.80 S_{Bi}$, maka dinilai "Layak." Rentang skor antara $X_i - 0.6 S_{Bi}$ dan $X_i + 0.60 S_{Bi}$ akan mendapat penilaian "Cukup Layak." Variabel atau item yang skornya berada di antara $X_i - 1.80 S_{Bi}$ dan $X_i - 0.60 S_{Bi}$ akan dianggap "Kurang Layak," sedangkan skor yang kurang dari atau sama dengan $X_i - 1.80 S_{Bi}$ akan masuk dalam kategori "Sangat Kurang Layak." Tabel ini memberikan panduan penilaian kualitatif dalam mengukur kelayakan variabel atau item berdasarkan deviasi dari *mean* dalam distribusi data.

Keterangan:

X = Skor total

X_i = $(1/2) \times (\text{skor tertinggi ideal} + \text{skor terendah ideal})$

S_{Bi} = Simpangan baku ideal $(1/3) \times (1/2) \times (\text{skor tertinggi ideal} - \text{skor terendah ideal})$

Skor tertinggi ideal = \sum butir kriteria \times skor tertinggi

Skor terendah ideal = \sum butir kriteria \times skor terendah

$M_i = \text{Mean ideal} = \frac{1}{2}$ (skor maksimal ideal + skor minimal ideal)

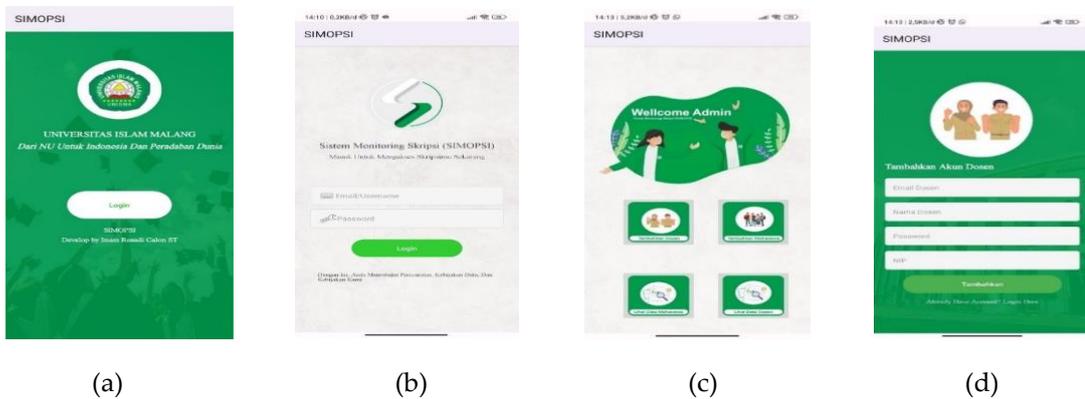
Setelah mendapatkan skor rata - rata tiap *instrument* kemudian menghitung presentasi kelayakan dengan rumus berikut ini:

Rumus Perhitungan Persentase:

$$\text{Persentase Kelayakan(\%)} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

3. Hasil and Pembahasan

1. Hasil Rancangan *Interface*



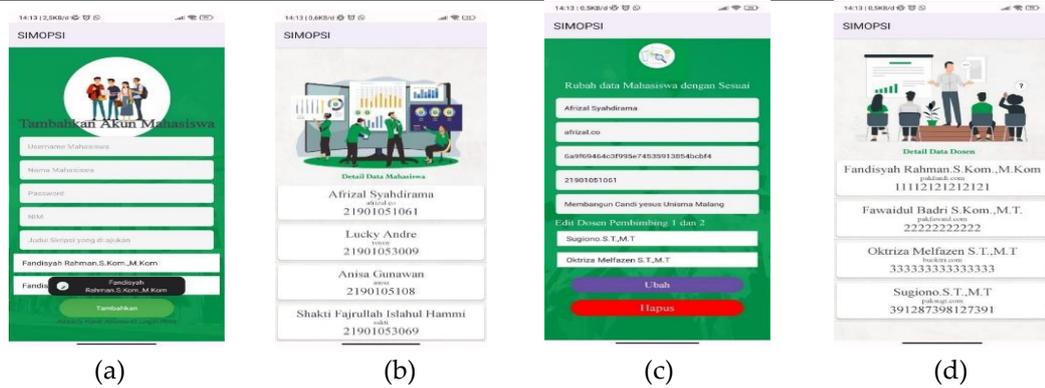
Gambar 3. Tampilan (a) Halaman *SplashScreen*, (b) Halaman *Login*, (c) Halaman *Admin*, (d) Halaman *Tambah Akun Dosen*

Halaman pada Gambar 3(a) Merupakan halaman saat *icon* aplikasi di tekan di halaman tersebut hanya berisi tombol untuk mengarahkan *user* ke halaman login.

Halaman Pada Gambar 3(b) merupakan halaman *login* aplikasi yang dirancang menggunakan *Android Studio*. Halaman ini digunakan oleh para *user* untuk melakukan otentikasi dan masuk ke akun terdaftar mereka. Para pengguna akan memasukkan beberapa informasi pengenalan, seperti alamat *email* atau nama pengguna (*username*), dan kata sandi (*password*) untuk memverifikasi identitas mereka. Ini diperlukan untuk memastikan bahwa hanya pengguna yang sah yang dapat mengakses akun tersebut.

Gambar 3(c) menunjukkan halaman admin atau bagian yang dirancang khusus untuk pengelola sistem atau *administrator*. Terdiri dari menu tambahkan dosen, tambahkan mahasiswa, lihat data mahasiswa, dan lihat data dosen.

Gambar 3(d) merupakan tampilan halaman admin untuk menambahkan akun dosen yang terdiri dari beberapa informasi yaitu *email* dosen, nama dosen, *password*, Nomor Identitas Pegawai Negeri Sipil (NIP), dan juga *add button*.



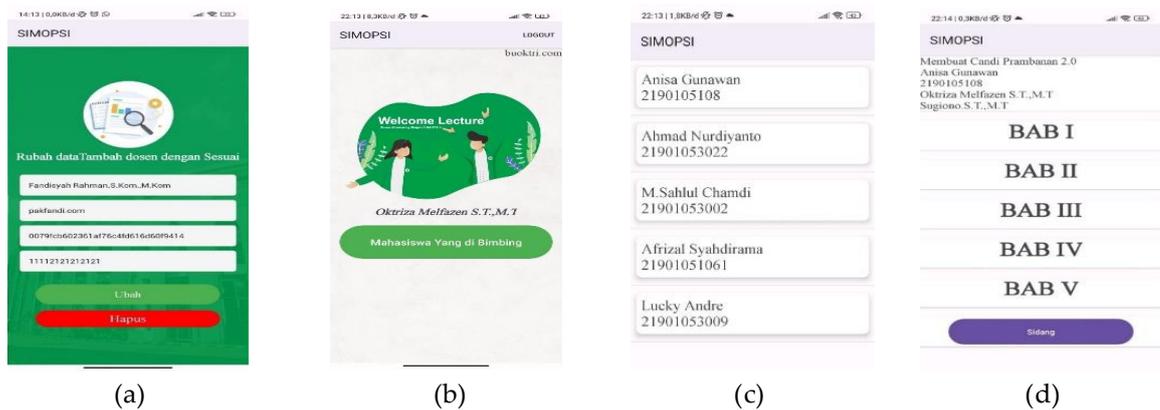
Gambar 4. Tampilan (a) Halaman Tambah Akun Mahasiswa, (b) Halaman Detail Mahasiswa, (c) Halaman Ubah Data Mahasiswa, (d) Halaman Detail Dosen

Gambar 4(a) Mendefinisikan tugas dari admin untuk menambahkan akun mahasiswa yang terdiri dari *username*, *password*, nama mahasiswa, nim, judul skripsi, dosen pembimbing 1 dan 2 yang akan dikirimkan kedalam *database* yang telah di sediakan[14].

Gambar 4(b) merupakan halaman detail admin data mahasiswa, halaman ini menampilkan informasi rinci tentang mahasiswa yang telah berhasil ditambahkan dan seluruh akun akan muncul di daftar detail tersebut. Selain itu, admin memiliki *access* untuk melihat dan memodifikasi data-data mahasiswa[15].

Gambar 4(c) menunjukkan halaman admin dalam mengubah data mahasiswa. apabila terdapat kesalahan dalam penginputan data mahasiswa maka admin dapat melakukan atau memasukkan berbagai perubahan data yang diperlukan atau menghapus data tersebut pada halaman ini.

Gambar 4(d) merupakan halaman detail dosen yang telah ditambahkan oleh admin, seluruh informasi rinci dan daftar dosen pembimbing yang telah berhasil terdaftar akan muncul dalam *list* tersebut. Informasi mengenai dosen dapat dengan mudah diakses oleh Admin.



Gambar 5. Tampilan (a) Ubah dan Hapus data Dosen, (b) Halaman Menu Dosen, (c) Halaman Daftar mahasiswa yang di bimbing, (d) Halaman Dosen History Mahasiswa

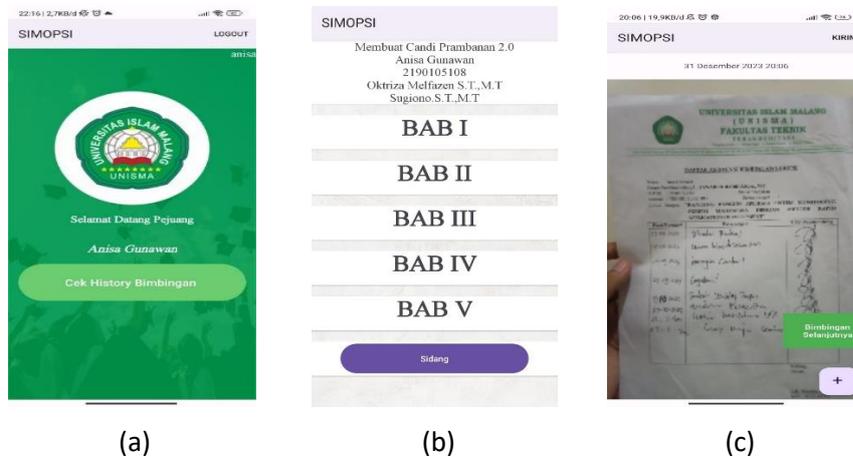
Ditunjukkan pada gambar 5(a) selain mengubah data mahasiswa, admin juga dapat melakukann perubahan dan penghapusan data dosen seperti yang pada halaman ini memungkinkan

admin untuk mengubah data dosen yang terdiri dari nama, *username*, *password*, dan NIP. Terdapat tombol ubah dan hapus untuk memproses segala data yang telah diinput oleh admin.

Gambar 5(b) merupakan tampilan halaman menu dosen yang digunakan oleh dosen terdaftar untuk mengakses berbagai fitur dalam aplikasi SIMOPSI. Pada halaman ini terdapat *welcome icon*, nama dosen, *form* daftar mahasiswa yang dibimbing, *username*, dan *logout button*.

Gambar 5(c) merupakan halaman daftar mahasiswa yang dibimbing oleh dosen, pada tampilan halaman ini terdapat nama mahasiswa dan NIM. Untuk melihat informasi lebih rinci terkait mahasiswa bimbingannya, dosen dapat mengklik *item list* mahasiswa tersebut

Gambar 5(d) menunjukkan halaman dosen *history* mahasiswa yang terdiri dari judul skripsi, nama mahasiswa, NIM, nama dosen pembimbing 1 dan 2, lalu menu daftar bab skripsi, mulai dari bab 1 sampai bab 5 untuk proses monitoring *progress report* mahasiswa



Gambar 6. Tampilan (a) Halaman Menu Mahasiswa, (b) Halaman History Bab Mahasiswa ,(c) Halaman History Bab Mahasiswa,

Gambar 6(a) merupakan halaman menu mahasiswa yang digunakan oleh para mahasiswa terdaftar untuk mengakses berbagai fitur dalam aplikasi SIMOPSI. Pada halaman ini terdapat *welcome icon*, nama mahasiswa, menu *history* bimbingan, *username*, dan *logout button*

Pada Gambar 6(b) Menunjukkan halaman *history* mahasiswa yang terdiri dari judul skripsi, nama mahasiswa, NIM, nama dosen pembimbing 1 dan 2, lalu menu daftar bab skripsi, mulai dari bab 1 sampai bab 5 untuk menampilkan *progress report* mahasiswa

Pada Gambar 6(c) merupakan tampilan halaman revisi bab mahasiswa. Seluruh catatan atau informasi dari dosen pembimbing di tiap bab akan muncul pada halaman ini, mahasiswa dapat melihat atau mengaksesnya dengan mudah[16].

2. Pengujian Usability

Tabel 3 Pengujian Usability

No.	Nama Fitur	SB	B	C	K	SK
1.	Apakah fitur <i>login</i> admin berjalan dengan baik?	27	8	1		
2.	Apakah halaman admin berjalan dengan baik?	22	13	1		
3.	Apakah fitur admin tambahkan dosen berjalan dengan baik?	25	10	1		
4.	Apakah fitur admin tambahkan mahasiswa berjalan dengan baik?	20	15	1		
5.	Apakah fitur admin lihat seluruh dosen berjalan dengan baik?	22	12	2		
6.	Apakah fitur admin lihat seluruh mahasiswa berjalan dengan baik?	19	16	1		
7.	Apakah fitur admin ubah data dosen berjalan dengan baik?	21	13	2		

8.	Apakah fitur admin ubah data mahasiswa berjalan dengan baik?	20	15	1		
9.	Apakah fitur dosen lihat data mahasiswa yang di bimbing berjalan dengan baik?	22	12	2		
10.	Apakah fitur <i>login</i> dosen berjalan dengan baik?	22	13	1		
11.	Apakah halaman dosen berjalan dengan baik?	22	12	2		
12.	Apakah fitur dosen lihat data bimbingan dari mahasiswa yang dibimbing berjalan dengan baik?	18	17	1		
13.	Apakah fitur dosen tambah data bimbingan dari mahasiswa yang dibimbing berjalan dengan baik?	20	14	2		
14.	Apakah fitur login mahasiswa berjalan dengan baik?	21	13	2		
15.	Apakah halaman mahasiswa berjalan dengan baik?	19	15	2		
16.	Apakah fitur Mahasiswa lihat data bimbingan dari dosen berjalan dengan baik?	20	13	2		
17.	Apakah fitur <i>logout</i> berjalan dengan baik?	20	15	1		

Pengujian *usability* seperti yang ditunjukkan pada Tabel 1 dapat mempermudah proses identifikasi masalah dalam antarmuka pengguna dan mengevaluasi sejauh mana produk atau sistem memenuhi prinsip-prinsip *usability* yang meliputi efektivitas, efisiensi, dan kepuasan pengguna. Hasil dari pengujian dapat digunakan untuk membuat perbaikan dalam desain antarmuka dan fitur-fitur di dalamnya, sehingga aplikasi menjadi lebih mudah digunakan oleh pengguna [17].



Gambar 7. Grafik Penilaian Usability Fitur Aplikasi

Gambar 7 menunjukkan hasil penilaian responden terhadap seluruh fitur dalam aplikasi SIMOPSI. Pada hasil pengujian terdapat 58,9% responden yang menyatakan bahwa fitur aplikasi yang tersedia berjalan dengan sangat baik, 37% responden menyatakan fitur aplikasi berjalan dengan baik, sedangkan sisanya yaitu 4,1% responden menyatakan fitur aplikasi berjalan dengan cukup.

Tabel 4. Hasil Pengujian Skor Responden

No.	Penilaian	Poin	Jumlah Skor	Total
1.	Sangat Kurang	1	0	0
2.	Kurang	2	0	0
3.	Cukup	3	25	75
4.	Baik	4	226	904
5.	Sangat Baik	5	360	1800
Jumlah				2779

Tabel 4 merupakan data hasil pengujian skor responden dengan lima tingkatan penilaian dengan poin yang berbeda-beda, yaitu sangat kurang, kurang, cukup, baik, dan yang tertinggi ialah sangat baik. Hasil penilaian menunjukkan tingkatan sangat kurang dan kurang memperoleh jumlah skor 0. Tingkatan cukup memperoleh jumlah skor 25 dengan total poin 75. Pada tingkatan baik memperoleh jumlah skor 226 dengan total poin 904, sedangkan yang tertinggi ialah tingkatan sangat baik yang memperoleh jumlah skor 360 dengan total poin 1800. Total skor dari seluruh tingkatan penilaian adalah 2779.

Keterangan:

Skor yang Diobservasi = Hasil skor yang didapatkan dari pengujian.

Skor yang Diharapkan = Hasil perhitungan skor tertinggi ideal.

Skor total (X) = 2779

Jumlah butir kriteria = 17

Skor tertinggi ideal = Jumlah butir kriteria × responden × skor tertinggi
 = $17 \times 36 \times 5$
 = 3060

Skor terendah ideal = Jumlah butir kriteria × responden × skor terendah
 = $17 \times 36 \times 1$
 = 612

Mi = $(1/2) \times (\text{skor tertinggi ideal} + \text{skor terendah ideal})$
 = $(1/2) \times (3060 + 612)$
 = 1836

SBI = $(1/3) \times (1/2) \times (\text{skor tertinggi ideal} - \text{skor terendah ideal})$
 = $(1/3) \times (1/2) \times (3060 - 612)$
 = 367,2

3. Perhitungan Skala *Liekert* :

$Mi + (1,80 \times SBI) < X$

$1836 + (1,80 \times 367,2) < 2779$

$= 2496,96 < X (2779)$

Presentase Kelayakan = $\frac{\text{Skor Hasil Observasi}}{\text{Skor yang Diharapkan}} \times 100$
 = $\frac{2779}{3060} \times 100\%$

= 91% (Sangat layak)

Pada perhitungan menggunakan skala *likert* dengan tahap menghitung skor hasil observasi yaitu perhitungan skor yang didapatkan dari pengujian fitur dengan percobaan pada 36 responden memperoleh skor 2779 dibagi dengan skor yang diharapkan yaitu hasil perhitungan skor tertinggi ideal senilai 3060 dikali 100% dan diperoleh hasil 91% yang berarti hasil pengujian ini mendapatkan kategori sangat layak dalam pengujian *usability* fungsional sistem aplikasi

4. Kesimpulan

Hasil rancangan aplikasi sistem Monitoring Skripsi dengan mahasiswa dengan menggunakan metode RAD (*Rapid Application Development*) menghasilkan sebuah antarmuka Aplikasi yang intuitif dan responsif. Aplikasi ini dilengkapi dengan fitur login dari Admin, yang di program secara *default*, tugas dari admin adalah menambahkan akun untuk dosen pembimbing skripsi dan mahasiswa agar dapat *login* kedalam sebuah aplikasi, aplikasi ini di kontrol dari API *Service* yang telah di program dan di hubungkan ke Antarmuka yang di buat di *Android Studio* sebelumnya dengan menggunakan Bahasa pemrograman *Java*, seluruh data yang terkait berhasil di simpan dalam sebuah *database* yang telah di sediakan dengan keakuratan dan respon yang sesuai dari rancangan yang di harapkan.

Setelah menjalani beberapa pengujian, ternyata Aplikasi yang telah dibangun menunjukkan kualitas yang sangat baik. Dalam rangkaian pengujian tersebut, metode yang di gunakan adalah pengujian *Usability* berdasarkan fungsionalitas fitur sistem yang melibatkan para responden. Hasil pengujian *Usability* berdasarkan fungsionalitas fitur sistem menunjukkan tingkat kelayakan sebesar 91%, yang masuk dalam kategori "sangat layak," dengan partisipasi 36 responden. Oleh karena itu, dapat disimpulkan bahwa hasil pengujian aplikasi yang telah dirancang mendapatkan kualitas yang baik, sesuai dengan harapan dari pengujian yang dilakukan.

Pada aplikasi sistem monitoring skripsi mahasiswa menggunakan metode RAD (*Rapid Application Development*) ini masih terdapat beberapa kekurangan, sehingga terdapat beberapa saran untuk mengembangkan kedepan antara lain Dapat di gabungkan dengan aplikasi yang serupa dan

sudah ada misalnya MENARA UNISMA dan di perluas cakupan Implementasinya dalam lingkup Universitas Islam Malang agar seluruh mahasiswa Universitas Islam Malang dapat efisiensi dan manfaat dari aplikasi yang telah di buat dan Pada penelitian selanjutnya dapat di kembangkan dengan menambahkan beberapa fitur lain seperti Pendaftaran secara online dari Aplikasi secara langsung serta Evaluasi kinerja sistem monitoring skripsi menggunakan metode RAD(*Rapid Application Development*) dapat dilakukan untuk mengevaluasi keefektifan sistem dalam mengoptimalkan Sistem yang telah ada.

5. Referensi

- [1] P. Skripsi Tahun and T. Elektro, “Pedoman Skripsi Fakultas Teknik Universitas Islam Malang.”
- [2] W. Kurniawati and R. Setyaningsih, “Manajemen Stress pada mahasiswa tingkat akhir dalam penyusunan Skripsi jurnal An-Nur : Kajian pendidikan dan ilmu keislaman vol.8 ,No.Juli – Desember 2022 ISSN 2461-1158
- [3] A. Syam and W. Tita, “Penerapan Informasi layanan pengajuan dan monitoring judul tugas akhir mahasiswa *Implementation of information system on the submission and monitoring service of student final project*,” Jurnal Informatika, vol. 10, no. 1, 2021, [Online]. Available: <http://ejournal.unidayan.ac.id/index.php/JIU>
- [4] F. J. Atletiko, “*Development of Android Application for Courier Monitoring System*,” in *Procedia Computer Science*, Elsevier B.V., 2017, pp. 759–766. doi: 10.1016/j.procs.2017.12.215.
- [5] N. C. C. Brown, P. Weill-Tessier, M. Sekula, A. L. Costache, and M. Kölling, “*Novice Use of the Java Programming Language*,” *ACM Transactions on Computing Education*, vol. 23, no. 1, Dec. 2022, doi: 10.1145/3551393.
- [6] N. M. Sari, L. Mula Tua, and E. Krisnanik, *Sistem Informasi Monitoring Pembimbingan Skripsi/Tugas Akhir (SIMP-S/TA) Berbasis Android*.
- [7] O. Berbasis, W. Arman, and S. Karim, “Sistem Informasi Tugas Akhir,” *Jurnal Informanika*, vol. 2, no. 2, p. 35141, 2016.
- [8] D. A. Kristiyanti and A. Mulyana, “Sistem Informasi Monitoring Skripsi Berbasis Web (Studi Kasus: Prodi Akuntansi Universitas Mercu Buana),” *JURNAL SISTEM INFORMASI BISNIS*, vol. 10, no. 1, pp. 56–63, Jun. 2020, doi: 10.21456/vol10iss1pp56-63.
- [9] N. Satyahadewi, N. Mutiah, J. P. H. Nawawi, and P.-K. Barat, “SISTEM INFORMASI MONITORING TUGAS AKHIR (SIMTA) BERBASIS WEB FAKULTAS MIPA UNIVERSITAS TANJUNGPURA,” 2019.
- [10] O. Berbasis, W. Arman, and S. Karim, “Sistem Informasi Tugas Akhir,” *Jurnal Informanika*, vol. 2, no. 2, p. 35141, 2016.
- [11] D. A. Kristiyanti and A. Mulyana, “Sistem Informasi Monitoring Skripsi Berbasis Web (Studi Kasus: Prodi Akuntansi Universitas Mercu Buana),” *JURNAL SISTEM INFORMASI BISNIS*, vol. 10, no. 1, pp. 56–63, Jun. 2020, doi: 10.21456/vol10iss1pp56-63.
- [12] S. Aswati, M. S. Ramadhan, A. U. Firmansyah, and K. Anwar, “Studi analisis model Rapid Application Development dalam pengembangan system informasi 1,” vol. 16, no. 2, p. 2017.
- [13] Skala-pengukuran dan jumlah respon skala likert Jurnal Ilmu Pertanian dan Perikanan Desember 2013 Vol. 2 No. 2 Hal : 127-133 ISSN 2302-6308”.
- [14] J. David, J. Euzenat, F. Scharffe, and C. Trojahn dos Santos, “*The Alignment API 4.0*,” vol. 2, no. 1, 2011, doi: 10.3233/SW.
- [15] M. Ridwan, F. Badri, A. Faqih, and R. Mono Sari, “Rancang Bangun Sistem Pendukung Keputusan Rekomendasi Jurusan Kuliah Menggunakan Metode Simple Additive *Weighting* INFORMASI ARTIKEL,” *Informatics, Electrical and Electronics Engineering (Infotron)*, vol. 3, no. 1, pp. 26–35, [Online]. Available: <http://riset.unisma.ac.id/index.php/infotron/article/view/19788>
- [16] M. Ridwan, F. Badri, A. Faqih, and R. Mono Sari, “Rancang Bangun Sistem Pendukung Keputusan Rekomendasi Jurusan Kuliah Menggunakan Metode *Simple Additive Weighting* INFORMASI ARTIKEL,” *Informatics, Electrical and Electronics Engineering (Infotron)*, vol. 3, no. 1, pp. 26–35, [Online]. Available: <http://riset.unisma.ac.id/index.php/infotron/article/view/19788>
- [17] R. Fachrizal Rochman, K. Candra Brata, and H. Tolle, “Perbaikan Fitur *Dashboard* Skripsi di FILKOM APPS dengan menggunakan Pendekatan *Human Centered Design*,” 2022. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [18] A. Setiawan, B. Minto, F. Badri, and S. Artikel, “Jurnal Ilmiah Teknik Informatika,” 2021. [Online]. Available: www.journal.unisma.ac.id:8080/index.php/infotron