

Implementasi Algoritma *Selection Sort* pada Aplikasi *To-Do List* Berbasis Android untuk Manajemen Waktu Mahasiswa

Muhammad Rayhan Permana^{1*}, Yuli Asriningtias²

^{1,2}Universitas Teknologi Yogyakarta

Jl. Siliwangi, Jombor Lor, Sendangadi, Kec. Mlati, Kabupaten Sleman, DI Yogyakarta, Indonesia

*Corresponding author e-mail : muhammadrayhan0389@gmail.com

ABSTRAK

Penelitian ini bertujuan untuk menerapkan algoritma *selection sort* pada aplikasi *to-do list* berbasis Android untuk manajemen waktu mahasiswa. Aplikasi *to-do list* dikembangkan dengan metode *Rapid Application Development* (RAD) untuk memungkinkan pengembangan yang cepat dan berulang, sehingga pengembangan aplikasi dapat diselesaikan dalam waktu yang relatif singkat. Aplikasi ini dilengkapi dengan fitur pengurutan aktivitas berdasarkan tenggat waktu, sehingga mahasiswa dapat mengutamakan aktivitas yang harus diselesaikan terlebih dahulu. Selain itu, aplikasi ini mampu mendeteksi tabrakan jadwal, memungkinkan pengguna menambah aktivitas tanpa khawatir akan konflik waktu. Pengujian aplikasi menunjukkan bahwa algoritma *selection sort* dapat mengurutkan aktivitas dengan akurasi 100% berdasarkan tenggat waktu, sehingga membantu mahasiswa dalam manajemen waktu. Aplikasi ini diharapkan dapat membantu mahasiswa dalam mengatur jadwal aktivitas.

Kata kunci : *To-Do List*, *Selection Sort*, Android, Aktivitas, *Rapid Application Development*

ABSTRACT

This research aims to apply the selection sort algorithm to an Android-based to-do list application for student time management. The to-do list application was developed using the Rapid Application Development (RAD) method to enable rapid and iterative development, so that application development can be completed in a relatively short time. The app is equipped with an activity sorting feature based on deadlines, so that students can prioritise activities that must be completed first. In addition, the app is able to detect schedule collisions, allowing users to add activities without worrying about time conflicts. Application testing shows that the selection sort algorithm can sort activities with 100% accuracy based on deadlines, thus helping students in time management. This application is expected to help students in organising activity schedules.

Keywords: *To-Do List*, *Selection Sort*, Android, Activity, *Rapid Application Development*

I. PENDAHULUAN

Penggunaan teknologi untuk mengelola waktu dan aktivitas semakin penting, terutama bagi mahasiswa yang memiliki jadwal yang padat oleh berbagai aktivitas. Mahasiswa dihadapkan pada berbagai tantangan dalam manajemen waktu karena jadwal kuliah, aktivitas harian, dan kegiatan organisasi yang padat. Menurut penelitian oleh Zega dan Kurniawati (2022), manajemen waktu yang buruk berdampak negatif pada prestasi akademik mahasiswa [1]. Oleh karena itu, kemampuan manajemen waktu yang baik perlu dimiliki agar prestasi akademik tetap terjaga. Manajemen waktu yang baik dilakukan dengan mencatat dan

menentukan prioritas untuk setiap aktivitas [2]. Alat bantu juga diperlukan untuk mempermudah mahasiswa mengelola waktu secara efektif.

To-do list merupakan daftar aktivitas yang direncanakan dengan tanggal dan tenggat waktu yang sudah ditetapkan [3]. Mahasiswa dapat memanfaatkan *to-do list* untuk merencanakan aktivitas sehari-hari agar terorganisir dan produktif. Meskipun demikian, pembuatan *to-do list* secara manual membutuhkan banyak waktu serta sulit diperbarui atau disesuaikan dengan perubahan. Adanya kendala tersebut, maka diperlukan pengembangan aplikasi *to-do list*. Aplikasi *to-do list* dapat mempermudah pembaruan dan menyediakan pengingat otomatis, fitur yang sulit

diimplementasikan pada *to-do list* manual. Dengan aplikasi *to-do list*, mahasiswa dapat lebih fleksibel mengelola aktivitas sesuai tenggat waktu.

Berdasarkan permasalahan tersebut, aplikasi *to-do list* berbasis Android dengan algoritma *selection sort* dikembangkan untuk membantu mahasiswa dalam menyusun daftar aktivitas. Aplikasi *to-do list* dapat digunakan untuk mencatat aktivitas yang perlu dikerjakan dan menandai aktivitas yang telah diselesaikan [4]. Aplikasi ini menawarkan kemudahan akses dan fleksibilitas, serta memungkinkan pengguna untuk menggunakannya kapan pun dan di mana pun melalui perangkat Android [5]. Algoritma *selection sort* dipilih karena kesederhanaan dan efektivitasnya dalam mengurutkan elemen dengan jumlah terbatas, seperti aktivitas dalam aplikasi *to-do list*. Algoritma ini juga mudah diimplementasikan pada aplikasi Android dan memberikan hasil pengurutan yang sesuai untuk kebutuhan manajemen waktu. Aplikasi ini dikembangkan menggunakan metode *Rapid Application Development* (RAD) untuk memastikan pengembangan yang cepat dan berulang. Pengembangan aplikasi dilakukan dengan bahasa pemrograman Kotlin dan Jetpack Compose sebagai *UI toolkit* untuk membangun aplikasi Android *native*. Basis data SQLite digunakan untuk menyimpan data aktivitas mahasiswa karena sifatnya ringan dan mudah diintegrasikan dalam aplikasi Android. Fitur utama pada aplikasi yang dikembangkan adalah pengelolaan dan pengurutan jadwal aktivitas dengan kemampuan pendeteksi tabrakan jadwal, pengelolaan kategori aktivitas, serta laporan penyelesaian aktivitas. Selain itu, aplikasi ini dilengkapi fitur pengingat untuk membantu mahasiswa menyelesaikan aktivitas tepat waktu.

Adapun tujuan penelitian ini adalah untuk menerapkan algoritma *selection sort* pada aplikasi *to-do list* berbasis Android untuk manajemen waktu mahasiswa. Penelitian yang telah dilakukan memiliki perbedaan yang signifikan dengan penelitian-penelitian sebelumnya. Perbedaan tersebut terletak pada pendekatan yang digunakan dan fitur yang ditawarkan.

Penelitian yang dilakukan oleh Lestari dan Masitoh (2022) membahas tentang aplikasi Me-List berbasis Android. Aplikasi ini dirancang untuk membantu mahasiswa, pekerja, dan pelajar agar dapat mencatat tugas atau aktivitas penting [6]. Perbedaan aplikasi ini dengan aplikasi yang dikembangkan terletak pada target pengguna. Target pengguna pada aplikasi ini adalah mahasiswa, pekerja, dan pelajar. Sementara itu, target pengguna pada aplikasi yang dikembangkan adalah mahasiswa sehingga aplikasi dirancang khusus

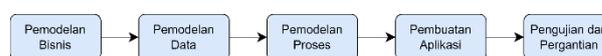
untuk kebutuhan manajemen waktu mahasiswa dengan algoritma *selection sort* untuk memprioritaskan aktivitas harian.

Penelitian yang dilakukan oleh Santhi, Sari, Putra, Diptha, Pebriawan, Haekal, dan Mahendra (2022) membahas tentang pengembangan sistem informasi *to-do list* berbasis *website*. Adanya sistem ini, mahasiswa dapat mencatat dan mengelola tugas-tugas yang harus diselesaikan, kemudian tugas-tugas dapat diurutkan sesuai prioritas dan tenggat waktu yang ditentukan [7]. Perbedaannya adalah penelitian ini membuat aplikasi *to-do list* dalam bentuk sistem informasi berbasis *website* khusus untuk mahasiswa Undiksha. Sementara itu, aplikasi yang dikembangkan merupakan aplikasi berbasis Android yang dapat digunakan oleh mahasiswa umum.

Penelitian yang dilakukan oleh Akbar dan Ridwan (2021) membahas tentang perancangan aplikasi *to-do list* untuk delegasi tugas secara *remote* di PT. Pupuk Kujang Cikampek. Aplikasi ini dirancang untuk membantu manajer dalam mendelegasikan tugas kepada karyawan [8]. Terdapat perbedaan dari segi kegunaan pada aplikasi ini dibandingkan dengan aplikasi yang dikembangkan. Aplikasi ini berfokus pada fitur pendistribusian tugas yang dibuat manajer kepada setiap karyawan secara tepat, sedangkan aplikasi yang dikembangkan berfokus pada fitur pencatatan jadwal aktivitas harian mahasiswa.

II. METODE

Metode yang digunakan dalam pengembangan aplikasi *to-do list* ini adalah metode *Rapid Application Development* (RAD). RAD merupakan model pengembangan yang berfokus pada pembuatan prototipe secara cepat dan mendapatkan umpan balik pengguna dengan segera, sehingga siklus pengembangan dan pengujian yang panjang dapat dihindari [9]. Hal ini sesuai dengan kebutuhan pengembangan yang memerlukan waktu pengembangan yang relatif singkat. Meskipun demikian, metode RAD tidak cocok diterapkan untuk pengembangan sistem yang berisiko teknis tinggi dan membutuhkan jumlah tenaga kerja yang banyak untuk menyelesaikan proyek berskala besar [10].



Gambar 1. Metode RAD

1. Pemodelan bisnis dilakukan untuk mengetahui kebutuhan sistem dan pengguna, serta proses bisnis dalam aplikasi. Pada tahap ini, kebutuhan sistem dan pengguna dijelaskan agar tujuan pengembangan aplikasi tercapai. Selain itu,

terdapat proses bisnis yang diusulkan sebagai hasil evaluasi dari proses bisnis yang telah ada.

2. Pemodelan data meliputi perancangan model data yang merepresentasikan struktur data yang digunakan dalam sistem. Struktur data aplikasi ini terdiri dari 5 kelas, yakni *User*, *Activity*, *SubActivity*, *Category*, dan *Subject*. Setiap kelas mewakili satu tabel dalam basis data SQLite.
3. Pemodelan proses dilakukan untuk menerjemahkan proses bisnis menjadi logika sistem. Pemodelan proses menentukan cara pengolahan data dan alur kerja sistem. Tahap ini terdiri dari *use case diagram* dan *activity diagram* fitur utama aplikasi.
4. Pembuatan aplikasi mencakup desain antarmuka dan penulisan kode. Desain antarmuka yang ditampilkan meliputi tampilan halaman Beranda, Kategori, dan Saya. Setiap halaman dirancang secara *user-friendly* agar pengguna lebih mudah memahami cara kerja aplikasi.
5. Pengujian dan pergantian mencakup pengujian fitur utama aplikasi yang dikembangkan untuk memastikan sistem berfungsi dengan baik. Pengujian dilakukan dengan metode *black-box testing* yang berfokus pada pengujian fungsional aplikasi. Setelah pengujian selesai, pengembangan dapat dilanjutkan ke fitur selanjutnya.

Selection Sort

Selection sort adalah algoritma pengurutan yang bekerja dengan mengurutkan elemen-elemen dalam larik secara bertahap dari kiri ke kanan. Pengurutan dilakukan dengan cara memilih nilai terkecil yang tersisa di setiap iterasi dan menukarnya ke posisi yang sesuai [11]. *Selection sort* merupakan pengembangan dari algoritma *bubble sort* dengan tujuan mengurangi jumlah perbandingan. Algoritma ini bekerja dengan mencari elemen terkecil dari larik pada rentang posisi 0 hingga N-1 [12]. Algoritma *selection sort* digunakan untuk mengurutkan aktivitas berdasarkan tenggat waktu, sehingga aktivitas dengan tenggat waktu terdekat ditempatkan di urutan pertama. Pengurutan aktivitas juga dapat dilakukan berdasarkan penyusunan nama aktivitas dari A-Z atau Z-A. Implementasi algoritma ini dalam aplikasi memungkinkan mahasiswa untuk mengidentifikasi aktivitas prioritas dengan cepat. Meskipun demikian, *selection sort* masih lebih lambat daripada *merge sort* dan *shell sort* untuk pengurutan lebih dari 1000 data [13].

50	30	10	60	40	20	Data awal
10	30	50	60	40	20	Iterasi ke-1
10	20	50	60	40	30	Iterasi ke-2
10	20	30	60	40	50	Iterasi ke-3
10	20	30	40	60	50	Iterasi ke-4
10	20	30	40	50	60	Iterasi ke-5

Gambar 2. Contoh Pengurutan *Selection Sort*

1. Mulai dari elemen pertama (indeks $i = 0$).
2. Cari indeks elemen terkecil di sublarik yang belum diurutkan.

$$\text{min_idx} = \arg \min\{A[i], A[i + 1], \dots, A[n - 1]\}$$
3. Tukar elemen terkecil dengan elemen pertama di sublarik yang belum diurutkan.

$$\text{swap}(A[i], A[\text{min_idx}])$$
4. Pindah ke elemen berikutnya.

$$i = i + 1$$
5. Ulangi sampai seluruh elemen dalam larik terurut.

III. HASIL DAN PEMBAHASAN

A. Tahap Pemodelan Bisnis

Pemodelan bisnis dilakukan untuk analisis kebutuhan sistem dan pengguna yang akan diimplementasikan dalam aplikasi yang dikembangkan. Tahap ini juga berguna untuk memahami alur proses bisnis yang akan dibangun. Alur proses bisnis ini merupakan hasil evaluasi dari alur proses bisnis yang telah ada.

Kebutuhan Sistem

Aplikasi *to-do list* yang dikembangkan merupakan aplikasi Android *native* yang dapat digunakan mahasiswa untuk membuat jadwal aktivitas harian. Kebutuhan sistem yang perlu diimplementasikan dalam aplikasi ini di antaranya adalah sebagai berikut.

1. Halaman Beranda, yang berisi proses penambahan dan pengurutan aktivitas, proses penandaan aktivitas, serta tampilan daftar aktivitas sesuai tanggal yang dipilih.
2. Halaman Kategori, yang berisi proses penambahan, penyuntingan, dan penghapusan kategori, serta tampilan daftar kategori.
3. Halaman Saya, yang berisi proses pembuatan dan tampilan laporan penyelesaian aktivitas berdasarkan kategori, serta daftar aktivitas yang telah diselesaikan.

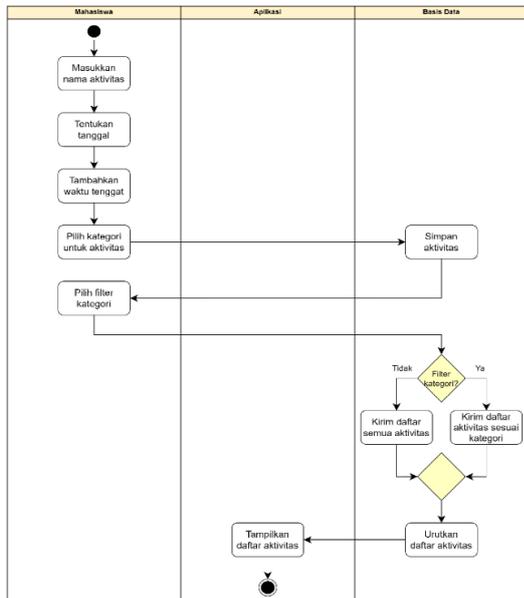
Kebutuhan Pengguna

Dalam aplikasi *to-do list* ini, pengguna utama aplikasi adalah mahasiswa. Adapun kebutuhan pengguna yang perlu diimplementasikan dalam aplikasi ini adalah sebagai berikut.

1. Mahasiswa dapat melihat tampilan halaman Beranda yang berisi daftar aktivitas sesuai tanggal yang dipilih.
2. Mahasiswa dapat mengisi *form dialog* penambahan aktivitas yang terdiri dari nama aktivitas, tenggat waktu, tanggal penyelesaian, dan kategori.
3. Mahasiswa dapat memilih cara pengurutan aktivitas.
4. Mahasiswa dapat melihat tampilan halaman Kategori yang berisi daftar kategori.
5. Mahasiswa dapat mengisi *form sheet* penambahan kategori yang hanya terdiri dari nama kategori.
6. Mahasiswa dapat menyunting dan menghapus kategori.
7. Mahasiswa dapat melihat tampilan halaman Saya yang berisi laporan penyelesaian aktivitas dan daftar aktivitas yang telah diselesaikan.

Proses Bisnis Saat Ini

Adapun perencanaan aktivitas dilakukan dengan memasukkan nama aktivitas dan tenggat waktu. Proses bisnis saat ini dapat dilihat pada Gambar 3.

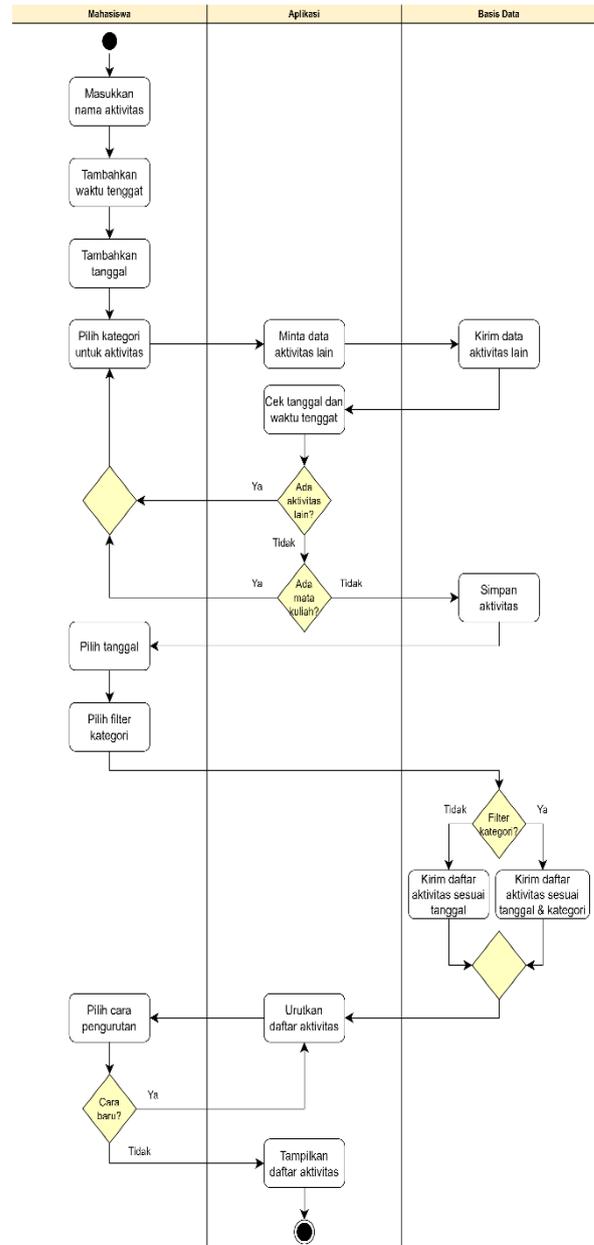


Gambar 3. Activity Diagram Proses Bisnis Saat Ini

Proses Bisnis yang Diusulkan

Berdasarkan evaluasi proses bisnis saat ini, aplikasi *to-do list* dapat dikembangkan dengan implementasi algoritma *selection sort* dan penambahan fitur pendeteksi tabrakan jadwal untuk manajemen waktu mahasiswa. Aplikasi ini dapat mendeteksi tabrakan jadwal antara jadwal aktivitas dan mata kuliah sehingga mahasiswa dapat memperkirakan tenggat waktu aktivitas yang akan dibuat berdasarkan jadwal mata kuliah yang telah

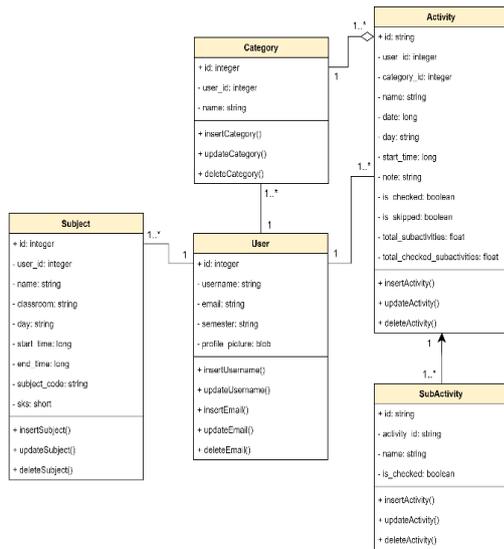
ditambahkan. Selain itu, daftar aktivitas diurutkan menggunakan algoritma *selection sort*.



Gambar 4. Activity Diagram Proses Bisnis Usulan

B. Tahap Pemodelan Data

Class diagram menggambarkan berbagai kelas yang ada dalam sistem beserta hubungan di antara kelas-kelas tersebut. *Class diagram* merupakan diagram statis karena hanya menampilkan kelas-kelas dengan atribut, metode, dan relasinya, namun diagram ini tidak menunjukkan interaksi antar kelas tersebut [14]. Terdapat 5 kelas yang digunakan pada aplikasi ini, yakni *User*, *Activity*, *SubActivity*, *Category*, dan *Subject*.



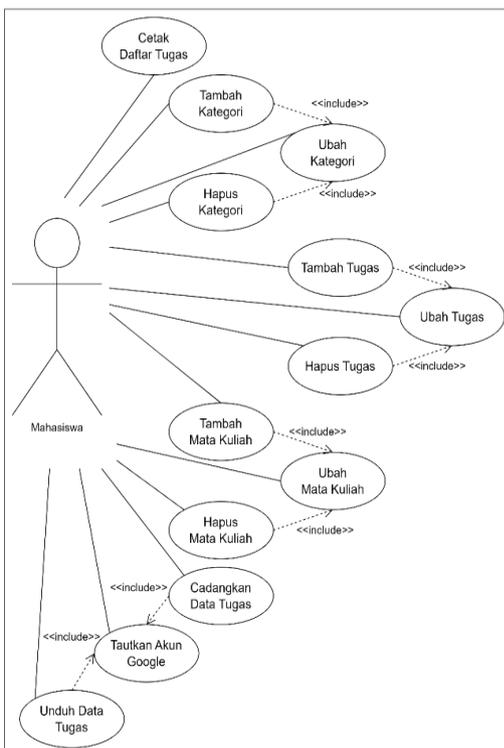
Gambar 5. Class Diagram

C. Tahap Pemodelan Proses

Tahap pemodelan proses dimanfaatkan untuk membuat *use case diagram* dan *activity diagram* dari proses yang terdapat pada aplikasi.

Use Case Diagram

Use case diagram digunakan untuk menggambarkan interaksi antara sistem yang dikembangkan dengan pengguna atau sistem eksternal di lingkungan pengembangan. Diagram ini berguna untuk mendefinisikan kebutuhan sistem dan menjadi alat komunikasi antara *stakeholder* dan pengembang sistem [15].

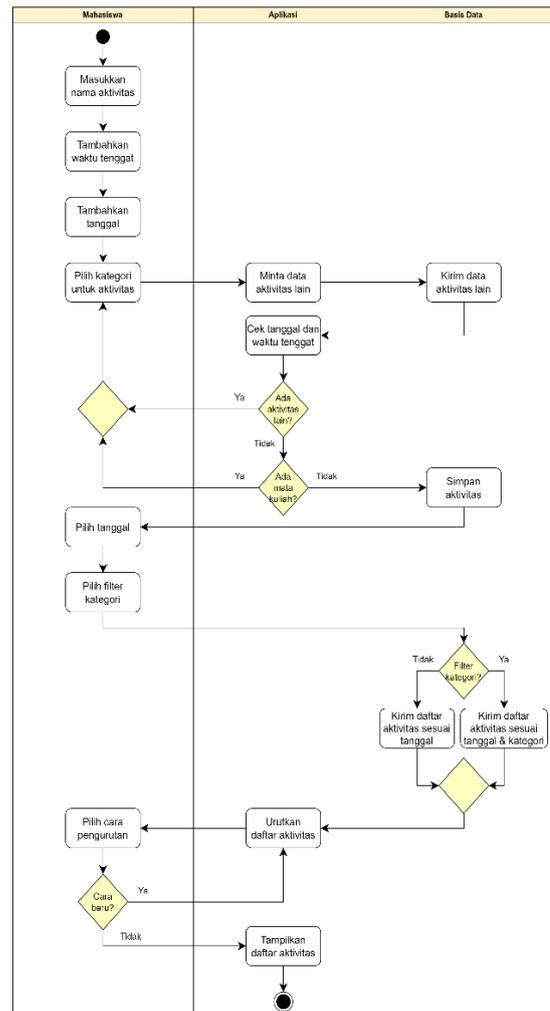


Gambar 6. Use Case Diagram

Activity Diagram

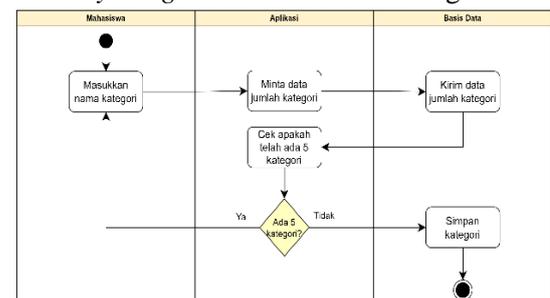
Activity diagram berfokus pada pemodelan proses bisnis atau prosedur dalam sebuah sistem. Diagram ini menggambarkan aliran kontrol dan data di antara berbagai tindakan yang diperlukan untuk menjalankan aktivitas dalam sistem [16]. *Activity diagram* yang ditampilkan terdiri dari *activity diagram* penambahan dan pengurutan aktivitas, penambahan kategori, serta laporan penyelesaian aktivitas.

1. Activity Diagram Penambahan dan Pengurutan Aktivitas



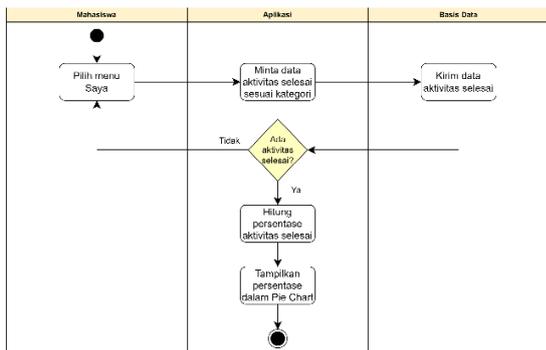
Gambar 7. Activity Diagram Penambahan dan Pengurutan Aktivitas

2. Activity Diagram Penambahan Kategori



Gambar 8. Activity Diagram Penambahan Kategori

3. Activity Diagram Laporan Penyelesaian Aktivitas



Gambar 9. Activity Diagram Laporan Penyelesaian Aktivitas

D. Tahap Pembuatan Aplikasi

Tahap pembuatan aplikasi berisi tampilan halaman pada aplikasi yang telah dibuat. Halaman yang ditampilkan merupakan halaman dari fitur utama aplikasi.

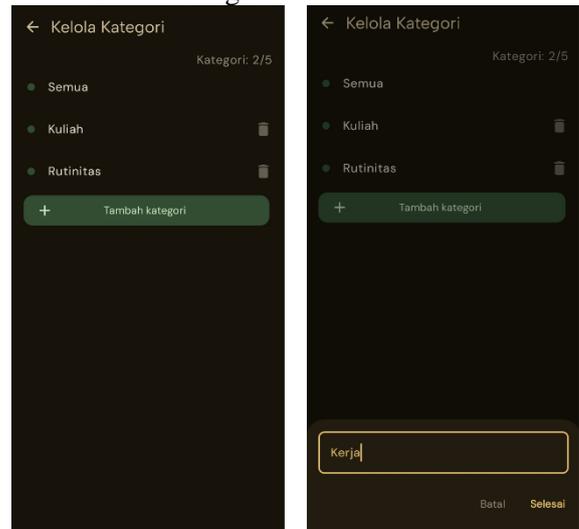
1. Tampilan Halaman Beranda, Form Dialog Penambahan Aktivitas, dan Dialog Pengurutan Aktivitas



Gambar 10. Halaman Beranda, Form Dialog Penambahan Aktivitas, dan Dialog Pengurutan Aktivitas

Halaman Beranda menampilkan daftar aktivitas yang diurutkan. Aktivitas diurutkan menggunakan algoritma *selection sort*. Selain itu, aktivitas dapat dicari sesuai tanggal dan kategori yang ditentukan. *Form dialog* penambahan aktivitas ditampilkan ketika mahasiswa mengeklik *floating action button* tambah. *Form dialog* berguna untuk menambah aktivitas baru. *Dialog* pengurutan aktivitas ditampilkan ketika mengeklik tombol submenu dan memilih submenu Urutkan Aktivitas. Pengguna dapat memilih cara pengurutan aktivitas yang terdiri dari 3 cara, yakni Tanggal & Waktu Mulai, Abjad A-Z, dan Abjad Z-A.

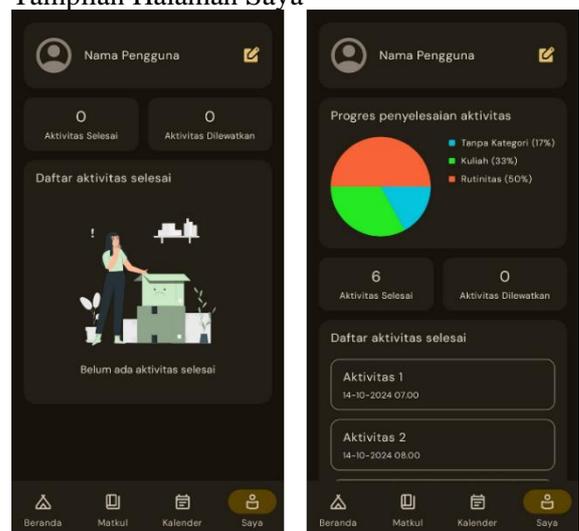
2. Tampilan Halaman Kategori dan Form Sheet Penambahan Kategori



Gambar 11. Halaman Kategori dan Form Sheet Penambahan Kategori

Halaman Kategori menampilkan daftar kategori yang telah ditambahkan. Mahasiswa hanya dapat membuat 5 kategori. Selain itu, mahasiswa dapat menghapus kategori jika tidak ada aktivitas dengan kategori yang akan dihapus. Tombol tambah kategori berguna untuk menampilkan *form sheet* penambahan kategori agar mahasiswa dapat menambahkan kategori baru.

3. Tampilan Halaman Saya



Gambar 12. Halaman Saya

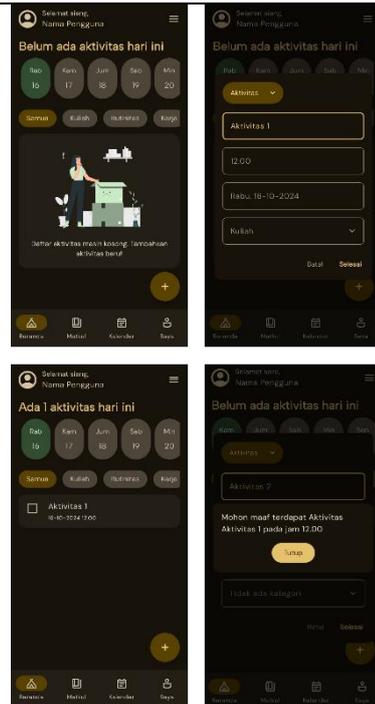
Halaman Saya menampilkan persentase penyelesaian aktivitas beserta daftar aktivitas yang telah diselesaikan. Mahasiswa juga dapat melihat daftar aktivitas yang dilewatkan. Selain itu, terdapat nama mahasiswa dan foto profil yang telah ditambahkan.

E. Tahap Pengujian dan Pergantian

Pengujian dilakukan dengan metode *black-box testing*, yang mencakup pengujian penambahan dan pengurutan aktivitas, penambahan kategori, serta tampilan persentase penyelesaian aktivitas. *Black-box testing* adalah metode pengujian yang difokuskan pada evaluasi fungsi perangkat lunak tanpa mempertimbangkan kode sumber atau struktur internalnya [17]. Setiap skenario pengujian dilakukan berdasarkan kasus penggunaan mahasiswa, seperti menambahkan aktivitas baru dengan tenggat waktu yang dekat, mengurutkan aktivitas berdasarkan penyusunan nama aktivitas dari A-Z, menambahkan kategori baru, serta menyelesaikan aktivitas dari berbagai kategori untuk membuat persentase penyelesaian aktivitas. Pengujian ini dilakukan pada tahap akhir pengembangan perangkat lunak untuk memastikan bahwa perangkat lunak berfungsi dengan semestinya [18].

Tabel 1. Pengujian Penambahan Aktivitas

<i>Nama Kasus Uji</i>	Menambah aktivitas	
<i>Deskripsi</i>	Mahasiswa menambah aktivitas baru	
<i>Kondisi Awal</i>	Membuka halaman Beranda; Membuka <i>dialog</i> tambah aktivitas;	
<i>Tanggal Pengujian</i>	16 Oktober 2024	
	<i>Skenario</i>	
	<ol style="list-style-type: none"> Mahasiswa membuka halaman Beranda. Mahasiswa mengklik <i>floating action button</i> tambah. Mahasiswa mengisi <i>form</i> dengan data aktivitas baru. Mahasiswa mengklik tombol selesai. Aplikasi meminta data aktivitas lain pada tanggal yang sama. Aplikasi mengecek apakah ada kesamaan waktu antara aktivitas baru dan aktivitas yang sudah ada. 	
	<i>Yang Diharapkan</i>	<i>Pengamatan</i>
	Aplikasi membuka <i>form</i> pengisian data aktivitas baru. Aplikasi mengecek waktu aktivitas. Jika waktu aktivitas belum ada, aplikasi menyimpan aktivitas baru pada basis data. Kemudian, <i>form</i> ditutup dan tampilan kembali ke halaman Beranda.	Aplikasi sukses melakukan pengecekan waktu aktivitas; Jika ada aktivitas pada waktu yang sama, maka tampil <i>dialog</i> peringatan; Data aktivitas baru sukses disimpan pada basis data; Tampilan <i>form</i> ditutup dan kembali ke halaman Beranda;
	<i>Kesimpulan</i>	
	Sesuai	
	<i>Hasil Pengujian</i>	



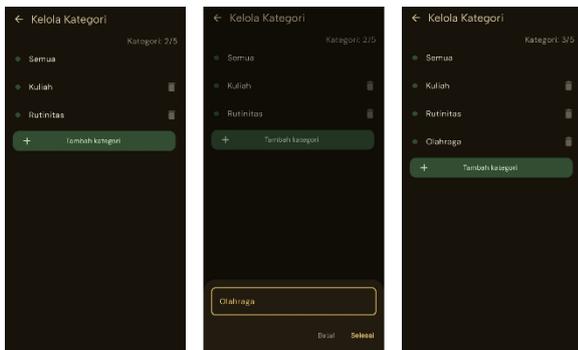
Tabel 2. Pengujian Pengurutan Aktivitas

<i>Nama Kasus Uji</i>	Mengurutkan aktivitas	
<i>Deskripsi</i>	Mahasiswa memilih cara pengurutan aktivitas	
<i>Kondisi Awal</i>	Membuka halaman Beranda;	
<i>Tanggal Pengujian</i>	16 Oktober 2024	
	<i>Skenario</i>	
	<ol style="list-style-type: none"> Mahasiswa membuka halaman Beranda. Mahasiswa mengklik tombol submenu. Mahasiswa memilih submenu Urutkan Aktivitas. Mahasiswa memilih cara pengurutan aktivitas. Mahasiswa mengklik tombol selesai. Aplikasi mengurutkan aktivitas dengan algoritma <i>selection sort</i>. Daftar aktivitas ditampilkan pada halaman Beranda. 	
	<i>Yang Diharapkan</i>	<i>Pengamatan</i>
	Aplikasi membuka submenu, lalu aplikasi membuka <i>dialog</i> pengurutan aktivitas. Aplikasi mengurutkan aktivitas dengan algoritma <i>selection sort</i> sesuai cara pengurutan yang dipilih mahasiswa. Kemudian, aplikasi menutup <i>dialog</i> cara pengurutan aktivitas dan kembali ke halaman Beranda.	Dialog submenu sukses ditampilkan; Pilihan cara pengurutan hanya dapat dipilih satu; Aplikasi sukses mengurutkan aktivitas dengan algoritma <i>selection sort</i> ; Tampilan <i>dialog</i> ditutup dan kembali ke halaman Beranda;
	<i>Kesimpulan</i>	
	Sesuai	
	<i>Hasil Pengujian</i>	



Tabel 3. Pengujian Penambahan Kategori

<i>Nama Kasus Uji</i>	Menambah kategori
<i>Deskripsi</i>	Mahasiswa menambah kategori baru
<i>Kondisi Awal</i>	Membuka halaman Kategori;
<i>Tanggal Pengujian</i>	16 Oktober 2024
<i>Skenario</i>	
<ol style="list-style-type: none"> 1. Mahasiswa membuka halaman Kategori. 2. Mahasiswa mengklik tombol tambah kategori. 3. Mahasiswa menambah kategori baru. 4. Mahasiswa mengklik tombol selesai. 5. Aplikasi mengecek apakah jumlah kategori kurang dari sama dengan lima. 6. Aplikasi menyimpan kategori pada basis data. 	
<i>Yang Diharapkan</i>	<i>Pengamatan</i>
Aplikasi membuka halaman Kategori, lalu aplikasi menampilkan <i>sheet</i> tambah kategori.	Aplikasi sukses menampilkan halaman Kategori;
Aplikasi mengecek jumlah kategori apakah kurang dari sama dengan lima. Jika kategori kurang dari sama dengan lima, maka aplikasi menyimpan kategori pada basis data. Kemudian, aplikasi menutup <i>sheet</i> dan kembali ke halaman Kategori.	Aplikasi sukses menampilkan <i>sheet</i> tambah kategori; Aplikasi sukses mengecek jumlah kategori; Data kategori baru sukses tersimpan pada basis data; Tampilan <i>sheet</i> ditutup dan kembali ke halaman Kategori;
<i>Kesimpulan</i>	
Sesuai	
<i>Hasil Pengujian</i>	



Tabel 4. Pengujian Tampilan Persentase Penyelesaian Aktivitas

<i>Nama Kasus Uji</i>	Menampilkan persentase penyelesaian aktivitas
<i>Deskripsi</i>	Mahasiswa melihat persentase penyelesaian aktivitas
<i>Kondisi Awal</i>	Membuka halaman Saya;
<i>Tanggal Pengujian</i>	16 Oktober 2024
<i>Skenario</i>	
<ol style="list-style-type: none"> 1. Mahasiswa membuka halaman Saya. 2. Aplikasi mengecek apakah ada aktivitas yang telah selesai. 3. Aplikasi menghitung persentase aktivitas selesai untuk setiap kategori. 4. Aplikasi menampilkan <i>pie chart</i> untuk merepresentasikan persentase penyelesaian aktivitas. 	
<i>Yang Diharapkan</i>	<i>Pengamatan</i>
Aplikasi membuka halaman Saya. Aplikasi menghitung persentase aktivitas selesai. Kemudian aplikasi menampilkan <i>pie chart</i> untuk merepresentasikan penyelesaian aktivitas setiap kategori.	Aplikasi sukses menampilkan halaman Saya; Aplikasi dapat menghitung persentase penyelesaian aktivitas setiap kategori; Aplikasi sukses menampilkan <i>pie chart</i> .
<i>Kesimpulan</i>	
Sesuai	
<i>Hasil Pengujian</i>	



Hasil pengujian fitur utama menunjukkan bahwa aplikasi yang dikembangkan dapat digunakan untuk menambah dan mengurutkan aktivitas, menambah kategori, serta membuat persentase penyelesaian aktivitas. Algoritma *selection sort* mampu mengurutkan tugas berdasarkan tenggat waktu atau penyusunan nama aktivitas dengan akurasi 100%. Aplikasi ini juga dapat mendeteksi tabrakan jadwal yang mempermudah mahasiswa untuk mengelola jadwal aktivitas harian.

Pengujian Selection Sort

Selain pengujian fitur utama, pengujian *selection sort* juga dilakukan untuk mengetahui rata-rata waktu pengurutan pada perangkat Android. Pengujian dilakukan pada sampel data

urut dan acak yang berjumlah 50 hingga 150 data. Adapun perangkat Android yang digunakan adalah *smartphone* Realme 3 Pro, Oppo F11 Pro, dan Redmi Note 9. Masing-masing perangkat Android memiliki spesifikasi yang berbeda-beda.

1. Pengujian pada Realme 3 Pro

Perangkat Realme 3 Pro yang digunakan merupakan perangkat yang telah digunakan untuk aktivitas sehari-hari. Adapun spesifikasi perangkat Realme 3 Pro adalah sebagai berikut.

Tabel 5. Spesifikasi Realme 3 Pro

No	Komponen	Keterangan
1	CPU	Qualcomm Snapdragon 710
2	GPU	Adreno 616
3	RAM	4 GB
4	ROM	64 GB
5	Versi Android	Android 11

Pengujian algoritma *selection sort* pada data urut di perangkat ini menunjukkan bahwa rata-rata waktu pengurutan untuk ukuran sampel data terkecil (50) dari lima kali uji coba adalah 0,27 ms. Sementara itu, rata-rata waktu untuk ukuran sampel data terbesar (150) adalah 1,64 ms. Waktu pengurutan setiap ukuran sampel data dapat dilihat pada tabel berikut.

Tabel 6. Waktu Pengukuran Data Urut pada Realme 3 Pro

Ukuran Sampel	50	100	150
Percobaan ke-1 (ms)	0,28	0,76	2,07
Percobaan ke-2 (ms)	0,27	0,94	1,17
Percobaan ke-3 (ms)	0,28	0,92	1,60
Percobaan ke-4 (ms)	0,23	0,47	2,08
Percobaan ke-5 (ms)	0,27	0,54	1,30
Rata-Rata (ms)	0,27	0,73	1,64

Pengujian algoritma *selection sort* pada data acak di perangkat ini menunjukkan bahwa rata-rata waktu pengurutan untuk ukuran sampel data terkecil (50) dari lima kali uji coba adalah 3,36 ms. Sementara itu, rata-rata waktu untuk ukuran sampel data terbesar (150) adalah 5,82 ms. Waktu pengurutan setiap ukuran sampel data dapat dilihat pada tabel berikut.

Tabel 7. Waktu Pengukuran Data Acak pada Realme 3 Pro

Ukuran Sampel	50	100	150
---------------	----	-----	-----

Percobaan ke-1 (ms)	10,05	16,33	13,18
Percobaan ke-2 (ms)	5,94	7,85	10,00
Percobaan ke-3 (ms)	0,38	1,36	2,35
Percobaan ke-4 (ms)	0,30	0,75	1,25
Percobaan ke-5 (ms)	0,15	0,70	2,30
Rata-Rata (ms)	3,36	5,39	5,82

2. Pengujian pada Oppo F11 Pro

Perangkat Oppo F11 Pro yang digunakan merupakan perangkat yang telah digunakan untuk aktivitas sehari-hari. Adapun spesifikasi perangkat Oppo F11 Pro adalah sebagai berikut.

Tabel 8. Spesifikasi Oppo F11 Pro

No	Komponen	Keterangan
1	CPU	Mediatek Helio P70
2	GPU	Mali-G72 MP3
3	RAM	6 GB
4	ROM	64 GB
5	Versi Android	Android 11

Pengujian algoritma *selection sort* pada data urut di perangkat ini menunjukkan bahwa rata-rata waktu pengurutan untuk ukuran sampel data terkecil (50) dari lima kali uji coba adalah 0,09 ms. Sementara itu, rata-rata waktu untuk ukuran sampel data terbesar (150) adalah 0,43 ms. Waktu pengurutan setiap ukuran sampel data dapat dilihat pada tabel berikut.

Tabel 9. Waktu Pengukuran Data Urut pada Oppo F11 Pro

Ukuran Sampel	50	100	150
Percobaan ke-1 (ms)	0,08	0,17	0,43
Percobaan ke-2 (ms)	0,07	0,38	0,46
Percobaan ke-3 (ms)	0,12	0,17	0,43
Percobaan ke-4 (ms)	0,12	0,54	0,44
Percobaan ke-5 (ms)	0,07	0,17	0,41
Rata-Rata (ms)	0,09	0,29	0,43

Pengujian algoritma *selection sort* pada data acak di perangkat ini menunjukkan bahwa rata-rata waktu pengurutan untuk ukuran sampel data terkecil (50) dari lima kali uji coba adalah 1,05 ms. Sementara itu, rata-rata waktu untuk ukuran sampel data terbesar (150) adalah 3,26

ms. Waktu pengurutan setiap ukuran sampel data dapat dilihat pada tabel berikut.

Tabel 10. Waktu Pengukuran Data Acak pada Oppo F11 Pro

Ukuran Sampel	50	100	150
Percobaan ke-1 (ms)	1,33	2,73	12,11
Percobaan ke-2 (ms)	1,50	3,04	2,80
Percobaan ke-3 (ms)	0,08	0,23	0,50
Percobaan ke-4 (ms)	0,08	0,19	0,37
Percobaan ke-5 (ms)	0,08	0,24	0,52
Rata-Rata (ms)	1,05	1,29	3,26

3. Pengujian pada Redmi Note 9

Perangkat Redmi Note 9 yang digunakan merupakan perangkat yang telah digunakan untuk aktivitas sehari-hari. Adapun spesifikasi perangkat Redmi Note 9 adalah sebagai berikut.

Tabel 11. Spesifikasi Redmi Note 9

No	Komponen	Keterangan
1	CPU	Mediatek Helio G85
2	GPU	Mali-G52 MC2
3	RAM	6 GB
4	ROM	128 GB
5	Versi Android	Android 11

Pengujian algoritma *selection sort* pada data urut di perangkat ini menunjukkan bahwa rata-rata waktu pengurutan untuk ukuran sampel data terkecil (50) dari lima kali uji coba adalah 0,53 ms. Sementara itu, rata-rata waktu untuk ukuran sampel data terbesar (150) adalah 2,32 ms. Waktu pengurutan setiap ukuran sampel data dapat dilihat pada tabel berikut.

Tabel 12. Waktu Pengukuran Data Urut pada Redmi Note 9

Ukuran Sampel	50	100	150
Percobaan ke-1 (ms)	1,46	0,35	2,71
Percobaan ke-2 (ms)	0,47	1,12	2,83
Percobaan ke-3 (ms)	0,33	0,35	2,58
Percobaan ke-4 (ms)	0,11	0,37	2,68
Percobaan ke-5 (ms)	0,29	0,35	0,82
Rata-Rata (ms)	0,53	0,51	2,32

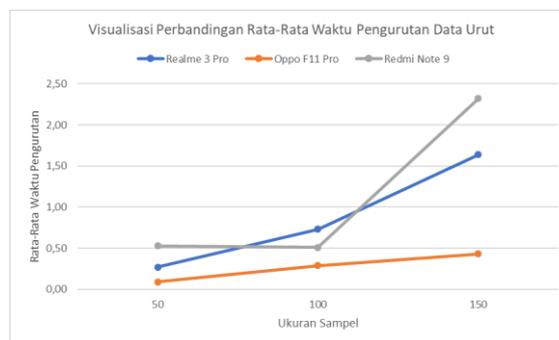
Pengujian algoritma *selection sort* pada data acak di perangkat ini menunjukkan bahwa rata-rata waktu pengurutan untuk ukuran sampel data terkecil (50) dari lima kali uji coba adalah 2,56 ms. Sementara itu, rata-rata waktu untuk ukuran sampel data terbesar (150) adalah 9,89 ms. Waktu pengurutan setiap ukuran sampel data dapat dilihat pada tabel berikut.

Tabel 13. Waktu Pengukuran Data Acak pada Redmi Note 9

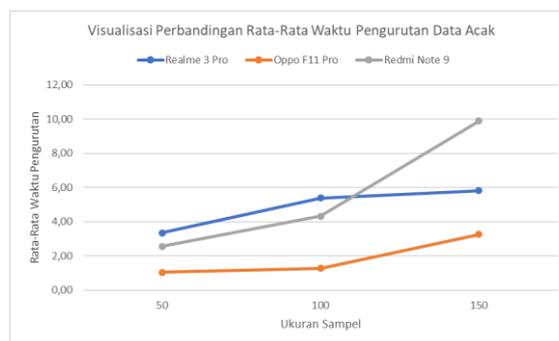
Ukuran Sampel	50	100	150
Percobaan ke-1 (ms)	5,60	8,89	25,60
Percobaan ke-2 (ms)	4,97	7,62	20,63
Percobaan ke-3 (ms)	0,33	3,24	1,17
Percobaan ke-4 (ms)	1,63	0,36	0,94
Percobaan ke-5 (ms)	0,28	1,54	1,13
Rata-Rata (ms)	2,56	4,33	9,89

Visualisasi Hasil Pengujian *Selection Sort*

Pengujian *selection sort* dilakukan menggunakan bahasa pemrograman Kotlin dengan *Integrated Development Environment (IDE)* Android Studio. Percobaan dilakukan lima kali untuk setiap ukuran sampel data agar mendapatkan rata-rata waktu pengurutan.



Gambar 13. Visualisasi Perbandingan Rata-Rata Waktu Pengurutan Data Urut



Gambar 14. Visualisasi Perbandingan Rata-Rata Waktu Pengurutan Data Acak

Hasil pengujian *selection sort* menunjukkan bahwa algoritma *selection sort* bekerja lebih cepat pada ukuran sampel di bawah 50 data. Untuk data yang lebih besar, algoritma ini mengalami kenaikan waktu pengurutan yang signifikan. Hal tersebut dapat menyebabkan waktu proses yang lebih lama untuk menampilkan daftar aktivitas pada aplikasi. Selain itu, perbedaan perangkat juga menjadi faktor lain yang mempengaruhi performa algoritma *selection sort*.

IV. KESIMPULAN

Adapun kesimpulan yang dapat diambil berdasarkan hasil pengembangan aplikasi *to-do list* berbasis Android dengan algoritma *selection sort* adalah sebagai berikut.

1. Aplikasi *to-do list* sukses menerapkan algoritma *selection sort* untuk mengurutkan aktivitas dengan akurasi 100%, sehingga mempermudah mahasiswa dalam menentukan prioritas tugas berdasarkan tenggat waktu.
2. Aplikasi *to-do list* sukses menerapkan fitur pendeteksi tabrakan jadwal yang dapat membantu mahasiswa dalam manajemen waktu.
3. Aplikasi *to-do list* dapat menyimpan kategori aktivitas sehingga aktivitas mahasiswa dapat diklasifikasikan berdasarkan kategori.
4. Aplikasi *to-do list* dapat membuat persentase penyelesaian aktivitas sehingga mahasiswa dapat melihat persentase aktivitas yang telah diselesaikan berdasarkan kategori.

Secara keseluruhan, aplikasi ini berhasil mencapai tujuannya untuk menerapkan algoritma *selection sort* pada aplikasi *to-do list* berbasis Android untuk manajemen waktu mahasiswa. Adanya aplikasi ini diharapkan dapat membantu mahasiswa dalam merencanakan aktivitas harian dan prioritas aktivitas, yang berpotensi mengurangi penundaan aktivitas dan meningkatkan produktivitas.

V. SARAN

Penelitian ini telah berhasil mencapai tujuannya. Namun, penelitian ini masih memiliki kekurangan pada performa algoritma *selection sort* dan fitur aplikasi. Algoritma *selection sort* memiliki keterbatasan dalam hal efisiensi untuk data dalam jumlah besar yang ditandai dengan peningkatan waktu pengurutan yang signifikan dalam aplikasi seiring bertambahnya ukuran data. Dari sisi fitur, aplikasi ini tidak dapat terintegrasi dengan aplikasi kalender bawaan perangkat atau Google Kalender. Aplikasi ini juga tidak memiliki fitur berbagi aktivitas yang memungkinkan pengguna berbagi daftar aktivitas ke sesama pengguna aplikasi.

Adanya kekurangan tersebut, penelitian selanjutnya diharapkan untuk mengembangkan aplikasi dengan mengeksplorasi algoritma pengurutan yang lebih efisien untuk menangani data yang lebih besar, serta menambah fitur integrasi kalender dan berbagi daftar aktivitas untuk meningkatkan fungsionalitas aplikasi.

DAFTAR PUSTAKA

- [1] Y. X. G. H. Zega dan G. E. Kurniawati, "Pentingnya Manajemen Waktu Bagi Mahasiswa dalam Meningkatkan Prestasi Belajar di Sekolah Tinggi Teologi Duta Panisal Jember," *METANOIA: Jurnal Pendidikan Agama Kristen*, vol. 4, no. 1, pp. 58-70, 2022.
- [2] M. Syelviani, "Pentingnya Manajemen Waktu dalam Mencapai Efektivitas bagi Mahasiswa (Studi Kasus Mahasiswa Program Studi Manajemen Unisi)," *Jurnal Analisis Manajemen (JAM)*, vol. 6, no. 1, pp. 23-32, 2020.
- [3] R. Azmi dan Rahmawati, "Perancangan Aplikasi Todolist Berbasis Android Menggunakan Flutter SDK dan Database SQLite," dalam *Seminar Nasional Riset dan Teknologi (SEMNAS RISTEK)*, Jakarta, 2021.
- [4] H. S. Wibowo, *Teknologi Informasi dan Komunikasi Untuk Mahasiswa*, Semarang: Tiram Media, 2023.
- [5] R. Syahputri, B. D. Prasetyo, Taufik, A. Andriyadi, H. W. Nugroho dan S. Trisnawati, "Rancang Bangun Aplikasi To Do List Budidaya Tanaman Buah Melon Berbasis Mobile dengan Algoritma Genetika (Studi Kasus : IBI Darmajaya Bandar Lampung)," dalam *Seminar Nasional Hasil Penelitian dan Pengabdian Masyarakat 2023*, Bandar Lampung, 2023.
- [6] P. A. Lestari dan A. H. Masitoh, "Aplikasi Me-List Berbasis Android Menggunakan Framework React Native," *Jurnal Esensi Infokom*, vol. 6, no. 2, pp. 26-30, 2022.
- [7] T. Santhi, A. M. Sari, D. K. A. M. Putra, I. G. C. A. Diphtha, K. Pebriawan, M. B. Haekal dan G. S. Mahendra, "Pengembangan Sistem Informasi To Do List Berbasis Website dalam Meninjau Kegiatan Mahasiswa Undiksha," *Jurnal Teknologi Ilmu Komputer*, vol. 1, no. 1, pp. 30-36, 2022.
- [8] R. Akbar dan T. Ridwan, "Perancangan Aplikasi Mobile To-Do List sebagai Platform Delegasi Tugas secara Remote pada Masa

- Pandemic di PT. Pupuk Kujang Cikampek,” *INTEGRATED (Information Technology and Vocational Education)*, vol. 3, no. 1, pp. 9-16, 2021.
- [9] L. V. Aprilian dan M. H. K. Saputra, Belajar cepat metode SAW, Bandung: Kreatif Industri Nusantara, 2020.
- [10] T. Pricillia dan Zulfachmi, “Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD),” *Jurnal Bangkit Indonesia*, vol. 10, no. 1, pp. 6-12, 2021.
- [11] G. T. Heineman, *Learning Algorithms: A Programmer's Guide to Writing Better Code*, Sebastopol: O'Reilly Media, 2021.
- [12] J. D. Putra, M. R. Ananda, A. Syahputra, Nurshabillah dan S. P. Sinaga, “Pengurutan Menggunakan Metode Selection Sort pada Sistem Informasi Lokasi Kontrakan di Kota Medan Berbasis Android,” *Blend Sains Jurnal Teknik*, vol. 1, no. 4, pp. 259-266, 2023.
- [13] Riki, M. Faridz, T. S. Hidayah dan Suharsono, “Perbandingan Algoritma Selection Sort, Shell Sort, dan Merge Sort pada Data Sampling Numerik Menggunakan Matplotlib,” dalam *Prosiding Seminar Nasional Sains dan Teknologi Seri 02*, Yogyakarta, 2024.
- [14] A. F. Elgamal, *Object-Oriented Programming*, Newcastle: Cambridge Scholars Publishing, 2024.
- [15] C. Dickerson dan S. Ji, *Essential Architecture and Principles of Systems Engineering*, Boca Raton: CRC Press, 2022.
- [16] M. Rusli dan E. Triandini, *Memodelkan Sistem Informasi Berorientasi Objek: Konsep Dasar, Prosedur, dan Implementasi*, Yogyakarta: Penerbit Andi, 2022.
- [17] J. Sinuraya, M. S. Wahyuni, H. A. Adwin, Harmayani, K. Sari dan Lusiyanti, *Analisis Perancangan Sistem*, Sumedang: CV. Mega Press Nusantara, 2024.
- [18] T. D. A. P. Wardhani dan Y. Asriningtias, “Implementasi Algoritma AES-256 dalam Perancangan Aplikasi Pengamanan Dokumen Digital Perusahaan Berbasis Android,” *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 6, no. 2, pp. 1289-1293, 2023.