

Development of an Android-Based Pomodoro Application with an Active Blocking Intervention to Improve Student Academic Productivity [Pengembangan Aplikasi Pomodoro Berbasis Android dengan Intervensi Pemblokiran Aktif untuk Meningkatkan Produktivitas Akademik Mahasiswa]

Azizah Sophia Azzahra¹⁾, Ade Eviyanti^{*,2)}, Nuril Lutvi Azizah³⁾, Mochamad Alfian Rosid⁴⁾

^{1, 2, 3, 4)} Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

*Email Penulis Korespondensi: adeeviyanti@umsida.ac.id

Abstract. *Academic procrastination caused by smartphone distraction is a critical problem hindering student productivity. Most time-management apps passively monitor activity without mechanisms to immediately break the procrastination cycle. his study aims to develop an Android-based Pomodoro application equipped with a Focus Guard feature, functioning as an active app-blocking intervention (fullscreen overlay) against distracting applications. The application was developed using the Waterfall method, Kotlin, and Jetpack Compose. Evaluation involved Blackbox testing on 13 scenarios (100% valid) and effectiveness testing on 35 students. Wilcoxon signed-rank test confirmed a highly significant productivity improvement ($p < 0.001$), with mean scores increasing from 19.06 to 22.54. The System Usability Scale (SUS) assessment yielded a score of 68.93, placing it in the Marginal High acceptability range (Grade C) with a Good rating. In conclusion, this application is functionally reliable, highly effective in reducing digital distractions, and provides an intuitive interface.*

Keywords - active blocking; android; pomodoro; procrastination; system usability scale

Abstrak. *Prokrastinasi akademik akibat distraksi ponsel pintar merupakan masalah kritis yang menghambat produktivitas mahasiswa. Mayoritas aplikasi manajemen waktu hanya memantau aktivitas secara pasif tanpa mekanisme untuk memutus siklus prokrastinasi seketika. Penelitian ini bertujuan mengembangkan aplikasi Pomodoro berbasis Android yang dilengkapi fitur Focus Guard sebagai intervensi pemblokiran aplikasi secara aktif (fullscreen overlay) terhadap aplikasi pengganggu. Aplikasi dikembangkan menggunakan metode Waterfall, Kotlin, dan Jetpack Compose. Pengujian melibatkan evaluasi Blackbox pada 13 skenario (100% valid) dan uji efektivitas terhadap 35 mahasiswa. Analisis Wilcoxon signed-rank test membuktikan peningkatan produktivitas yang sangat signifikan ($p < 0,001$), dengan lonjakan skor rata-rata dari 19,06 menjadi 22,54. Pengujian System Usability Scale (SUS) menghasilkan skor 68,93 yang menempatkan aplikasi pada kategori penerimaan Marginal High (Grade C) dengan predikat Baik (Good). Kesimpulannya, aplikasi ini terbukti fungsional, sangat efektif menurunkan distraksi digital, serta memiliki antarmuka intuitif.*

Kata Kunci - android; pemblokiran aktif; pomodoro; prokrastinasi; system usability scale

I. PENDAHULUAN

Produktivitas akademik merupakan faktor kunci bagi keberhasilan mahasiswa di era digital. Dalam konteks pendidikan tinggi, produktivitas dipahami sebagai kemampuan untuk menyelesaikan tugas secara efektif dan efisien melalui pembelajaran yang teregulasi secara mandiri [1]. Pencapaian produktivitas yang optimal menuntut keterampilan manajemen waktu yang mumpuni agar mahasiswa dapat meraih prestasi akademik sekaligus menjaga kesejahteraan psikologisnya [2]. Namun kenyataannya, produktivitas mahasiswa sering terhambat oleh masalah prokrastinasi. Prokrastinasi dalam konteks akademik merupakan perilaku penundaan tugas secara sengaja yang seringkali digantikan dengan aktivitas lain yang memberi kepuasan instan, yang pada akhirnya memicu penumpukan tugas dan stres. Sebuah studi di Indonesia mencatat bahwa lebih dari separuh mahasiswa sarjana (62%) berada pada level prokrastinasi kategori tinggi hingga sangat tinggi [3]. Fenomena ini menggambarkan lemahnya kemampuan regulasi diri akibat interaksi berbagai faktor, baik yang berasal dari dalam diri mahasiswa maupun dari lingkungan sekitarnya yang pada akhirnya memicu stres, kecemasan, dan keterlambatan dalam penyelesaian tugas [4], [5].

Akar permasalahan prokrastinasi bersifat kompleks karena menunda tugas sering kali menjadi mekanisme perlindungan diri mahasiswa dari tekanan akademik dan ketakutan akan kegagalan [6], [7]. Kondisi ini semakin parah dengan hadirnya distraksi teknologi dari luar. Penggunaan ponsel pintar yang berlebihan terbukti menjadi prediktor utama tingginya prokrastinasi dan stres akademik [8]. Kebiasaan ini memicu terjadinya *distractive multitasking*, yaitu proses pengalihan perhatian secara kognitif dari tugas utama menuju aplikasi yang tidak relevan seperti media sosial [9]. Secara paradoks, ponsel yang menjadi sumber distraksi tersebut justru merupakan perangkat utama untuk menjalankan aplikasi manajemen waktu yang dirancang untuk mengatasi distraksi itu sendiri [10].

Salah satu pendekatan yang terbukti populer untuk meningkatkan produktivitas adalah Teknik Pomodoro [11]. Metode ini merupakan sebuah teknik manajemen waktu terstruktur yang membagi waktu kerja atau belajar ke dalam

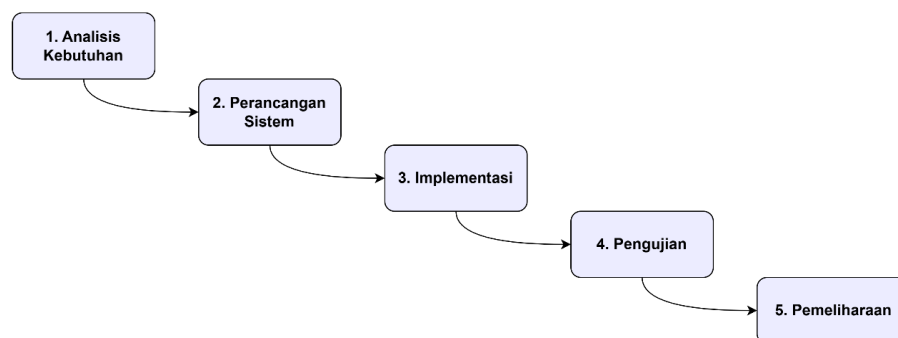
interval-interval fokus yang intens (umumnya 25 menit) yang kemudian diselingi dengan jeda istirahat singkat (umumnya 5 menit). Pendekatan ini mendorong pengguna untuk fokus pada satu tugas secara intens, lalu diselingi dengan istirahat singkat guna mengurangi kelelahan mental [12]. Akan tetapi, efektivitas metode ini sangat bergantung pada ketersediaan instrumen yang mendukung perencanaan sesi belajar secara tegas [13], [14]. Mengingat rata-rata mahasiswa menghabiskan 12,8 menit dari setiap 45 menit sesi belajar untuk membuka aplikasi hiburan [9], integrasi sistem pemblokiran aplikasi menjadi sangat penting. Sebelumnya, banyak penelitian telah berupaya mengatasi masalah ini melalui pengembangan perangkat lunak, seperti aplikasi “Locktimer” [15], “Focure” [16], “TIMEFLOW” [17], serta sistem penjadwalan berbasis kecerdasan buatan. Di sisi lain, adopsi teknologi juga harus memperhatikan kebiasaan pengguna. Implementasi sistem informasi berbasis *mobile* di lingkungan Universitas Muhammadiyah Sidoarjo telah terbukti secara efektif mampu memfasilitasi aksesibilitas data dan memenuhi kebutuhan mahasiswa secara praktis [18].

Kajian terhadap sejumlah penelitian sebelumnya memperlihatkan adanya celah (*gap*) yang belum banyak tergarap. Kebanyakan aplikasi manajemen waktu yang beredar saat ini masih bertumpu pada fungsi pasif, sebatas mencatat durasi kerja dan merekam pola perilaku pengguna, tanpa dilengkapi mekanisme yang benar-benar mencegah mahasiswa membuka aplikasi pengganggu di tengah sesi fokus. Kesenjangan inilah yang membuka ruang bagi pengembangan sebuah intervensi aktif berupa hambatan psikologis (*friction point*) yang dirancang untuk langsung memutus rantai prokrastinasi digital seketika.

Berangkat dari kesenjangan tersebut, penelitian ini merancang sebuah aplikasi *Pomodoro* berbasis Android yang diberi nama FoMo. Aplikasi ini dibekali fitur bernama pelindung fokus (*Focus Guard*), yaitu mekanisme pemblokiran aplikasi yang bekerja secara aktif. Ketika pengguna berupaya mengakses aplikasi yang masuk dalam daftar blokir (*blacklist*), sistem secara otomatis menampilkan layar penuh (*fullscreen overlay*) sebagai jeda paksa yang mendorong pengguna untuk kembali berkonsentrasi pada sesi belajarnya. Penelitian ini dilaksanakan dengan tiga tujuan utama. Pertama, merancang dan membangun aplikasi FoMo dengan fitur pemblokiran aplikasi secara aktif guna meningkatkan produktivitas mahasiswa. Kedua, menguji fungsionalitas sistem menggunakan metode *Blackbox Testing* serta mengukur efektivitas aplikasi dalam mengurangi distraksi digital melalui perbandingan hasil kuesioner *Pre-test* dan *Post-test*. Ketiga, mengevaluasi kelayakan dan tingkat kebergunaan antarmuka sistem bagi pengguna melalui instrumen SUS.

II. METODE

Pengembangan sistem pada penelitian ini mengikuti model *Waterfall*, yaitu pendekatan yang tahapannya berjalan secara bertahap dan berurutan mulai dari analisis kebutuhan, perancangan, implementasi, pengujian, hingga pemeliharaan sistem. Aplikasi dikembangkan menggunakan bahasa pemrograman Kotlin dan *toolkit UI Jetpack Compose* dengan mengadopsi arsitektur *Model-View-ViewModel* (MVVM) di lingkungan Android Studio. Alur metode pengembangan sistem ini divisualisasikan pada Gambar 1.



Gambar 1. Tahapan Metode *Waterfall*

A. Analisis Kebutuhan

Penentuan kebutuhan sistem ditempuh lewat dua jalur pengumpulan data. Pertama, penelusuran literatur untuk menelaah kajian-kajian sebelumnya seputar Teknik *Pomodoro* dan aplikasi manajemen waktu. Kedua, pengamatan langsung terhadap kebiasaan belajar mahasiswa untuk memetakan pola gangguan serta jenis aplikasi ponsel yang paling sering mengalihkan konsentrasi.

B. Perancangan Sistem

Logika kerja dan struktur aplikasi pada tahap perancangan digambarkan menggunakan notasi *Unified Modeling Language* (UML) yang terdiri atas *Flowchart* untuk alur proses, *Class Diagram* untuk struktur kelas, *Entity Relationship Diagram* (ERD) untuk relasi data, serta *Use Case Diagram* untuk interaksi pengguna. Selain itu, desain antarmuka (*wireframe*) dibuat menggunakan *Figma*.

C. Implementasi Sistem

Pengembangan perangkat lunak pada sistem ini diimplementasikan menggunakan bahasa pemrograman *Kotlin* dan *toolkit* antarmuka *Jetpack Compose*, dengan fondasi arsitektur *Model-View-ViewModel* (MVVM). Pendekatan arsitektur ini dipilih guna menstrukturkan kerangka kerja aplikasi ke dalam tiga lapisan operasional yang memiliki tanggung jawab terpisah. Lapisan pertama adalah *Model*, yang bertindak sebagai pusat penyimpanan dan pengolahan data lokal. Pada aplikasi ini, komponen *Model* direalisasikan melalui *Room Database* untuk merekam entitas tugas akademik secara persisten, serta *Jetpack DataStore* untuk menyimpan konfigurasi preferensi pengguna. Lapisan kedua adalah *View*, yang menampung seluruh komponen visual antarmuka sebagai wadah interaksi langsung dengan mahasiswa. Sebagai elemen pengendali, lapisan ketiga yaitu *ViewModel* berkedudukan sebagai jembatan logika yang menghubungkan basis data dengan antarmuka pengguna.

Secara teknis, mekanisme kerja arsitektur ini mengadopsi paradigma pemrograman reaktif. *ViewModel* secara aktif mengeksekusi operasi logika utama di latar belakang, seperti penghitungan waktu *Pomodoro* melalui *Foreground Service* dan pemantauan aktivitas aplikasi pihak ketiga menggunakan *UsageStatsManager API*. Pemrosesan logika tersebut dieksekusi secara asinkron menggunakan *Kotlin Coroutines*, dan seluruh hasil pemrosesannya disalurkan ke dalam bentuk aliran status (*StateFlow*). Di sisi lain, lapisan *View* beroperasi secara pasif dengan hanya mengamati aliran data tersebut, lalu memicu pembaruan tampilan antarmuka secara otomatis setiap kali mendeteksi adanya perubahan. Pemisahan tanggung jawab yang tegas ini memastikan mekanisme intervensi pemblokiran aplikasi (*Focus Guard*) guna memutus prokrastinasi digital dan fungsi pewaktu dapat beroperasi dengan stabil, responsif, dan terbebas dari tumpang-tindih instruksi program.

D. Pengujian

Kualitas aplikasi dan dampaknya terhadap pengguna diukur melalui tiga instrumen: (1) *Blackbox Testing* untuk memvalidasi fungsionalitas, (2) evaluasi *Pre-test* dan *Post-test* untuk mengukur efektivitas, dan (3) *System Usability Scale* (SUS) untuk mengukur tingkat kebergunaan antarmuka. Berikut adalah rincian prosedur dan parameter analisis yang digunakan dalam setiap tahapan pengujian tersebut.

Pengujian fungsionalitas (*Blackbox Testing*) dieksekusi dengan membandingkan luaran sistem aktual terhadap hasil yang diharapkan. Persentase validasi sistem dihitung menggunakan Persamaan 1, di mana aplikasi dinyatakan layak apabila mencapai ambang batas kelulusan minimal 90%.

$$\text{Persentase Validasi} = \left(\frac{\text{Jumlah Test Case yang Valid}}{\text{Total Keseluruhan Test Case}} \right) \times 100\% \quad (1)$$

Sebanyak 35 mahasiswa sarjana dilibatkan dalam pengujian efektivitas dan kebergunaan, dengan pemilihan responden dilakukan secara *purposive sampling*. Responden yang diikutsertakan ditentukan melalui dua kriteria inklusi. Pertama, kriteria teknis berupa kepemilikan perangkat berbasis sistem operasi Android versi 10 ke atas. Kedua, kriteria perilaku yang diverifikasi melalui pertanyaan skrining pada kuesioner ("Apakah Anda sering merasa terdistraksi oleh smartphone saat sedang fokus belajar/mengerjakan tugas?"), di mana hanya responden yang menjawab "Ya" yang dinyatakan memenuhi kriteria dan dilanjutkan ke tahap pengujian. Evaluasi dan pengisian instrumen dilaksanakan secara mandiri oleh responden melalui *Google Form*. Prosedur diawali dengan pengisian kuesioner *Pre-test* untuk memetakan tingkat distraksi awal, dilanjutkan dengan sesi simulasi uji coba langsung aplikasi FoMo secara mandiri berdurasi 20–30 menit untuk menguji fungsionalitas pewaktu dan merasakan pengalaman intervensi fitur *Focus Guard*. Setelah simulasi selesai, responden langsung mengisi kuesioner *Post-test* dan SUS guna menangkap respons langsung pengguna terhadap fitur pemblokiran aplikasi yang diterapkan.

Sebelum dilakukan uji komparasi, tahapan analisis data dimulai dengan uji prasyarat normalitas menggunakan metode *Shapiro-Wilk*. Pengujian normalitas ini difokuskan pada nilai total akumulatif kuesioner guna melihat distribusi data secara menyeluruh, bukan pada masing-masing butir indikator penilaian. Secara matematis, statistik uji *Shapiro-Wilk* (W) dihitung menggunakan Persamaan 2.

$$W = \frac{\left(\sum_{i=1}^n a_i x_{(i)} \right)^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (2)$$

Keterangan: W merepresentasikan nilai statistik uji *Shapiro-Wilk*, $x_{(i)}$ merupakan nilai data sampel ke- i yang telah diurutkan dari nilai terkecil hingga terbesar, $\bar{x}$ mewakili nilai rata-rata dari keseluruhan sampel pengamatan, a_i adalah konstanta koefisien uji dari tabel *Shapiro-Wilk*, dan n menunjukkan jumlah total sampel.

Uji lanjutan *Paired Sample t-test* digunakan apabila kedua kelompok data (*Pre-test* dan *Post-test*) terbukti sama-sama berdistribusi normal. Uji parametrik ini bekerja dengan membandingkan nilai rata-rata dari dua kelompok data yang berpasangan sebagaimana dirumuskan pada Persamaan 3.

$$t = \frac{\bar{d}}{s_d / \sqrt{n}} \quad (3)$$

Keterangan: t merepresentasikan nilai distribusi t-hitung, $\bar{d}$ merupakan rata-rata dari selisih pasangan data *Post-test* dan *Pre-test*, s_d adalah simpangan baku (standar deviasi) dari seluruh nilai selisih pasangan data, dan $\sqrt{n}$ menunjukkan jumlah keseluruhan pengamatan.

Sebaliknya, apabila salah satu atau kedua kelompok data tidak memenuhi asumsi distribusi normal, maka pengujian dialihkan menggunakan *Wilcoxon Signed-rank Test* sebagai metode alternatif non-parametrik. Berbeda dengan *t-test* yang menguji selisih rata-rata, uji *Wilcoxon* bekerja dengan memberikan peringkat (*ranking*) pada selisih nilai antar pengamatan berpasangan sebagaimana digambarkan pada Persamaan 4.

$$W = \min (W^+, W^-) \quad (4)$$

Keterangan: W merepresentasikan nilai statistik *Wilcoxon* yang didapatkan dari pengambilan nilai paling kecil (min) antara W^+ yang menunjukkan jumlah keseluruhan peringkat dari selisih pengamatan bernilai positif, dan W^- yang menunjukkan jumlah keseluruhan peringkat dari selisih pengamatan bernilai negatif.

Untuk mengukur tingkat produktivitas mahasiswa secara empiris, dilakukan perhitungan nilai rata-rata (*Mean*) dari total skor kuesioner menggunakan Persamaan 5.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (5)$$

Keterangan: $\bar{x}$ merepresentasikan nilai rata-rata dari suatu kelompok uji, $\sum_{i=1}^n x_i$ merupakan jumlah total skor dari seluruh responden pada kelompok uji tersebut, dan n menunjukkan jumlah responden yang berpartisipasi.

Mengacu pada rumus rata-rata tersebut, besaran efektivitas intervensi aplikasi diukur melalui selisih peningkatan atau *Delta Mean* ($\Delta\bar{x}$) yang dirumuskan pada Persamaan 6.

$$\Delta\bar{x} = \bar{x}_{post} - \bar{x}_{pre} \quad (6)$$

Keterangan: $\Delta\bar{x}$ merepresentasikan selisih rata-rata peningkatan produktivitas, $\bar{x}_{post}$ merupakan nilai rata-rata keseluruhan dari hasil pengujian *Post-test*, dan $\bar{x}_{pre}$ adalah nilai rata-rata keseluruhan dari hasil pengujian *Pre-test*.

Seluruh proses komputasi untuk pengujian normalitas dan hipotesis di atas dianalisis secara terkomputerisasi menggunakan perangkat lunak statistik jamovi. Pengambilan keputusan ditarik berdasarkan taraf signifikansi (α) sebesar 0,05, di mana hipotesis nol ditolak apabila probabilitas nilai p yang diperoleh berada di bawah 0,05. Hasil pengujian ini digunakan untuk membuktikan secara empiris signifikansi penurunan distraksi digital serta lonjakan produktivitas akibat intervensi fitur pemblokiran aplikasi secara aktif yang diterapkan oleh aplikasi.

Instrumen SUS yang dipakai merupakan versi berbahasa Indonesia yang telah diadaptasi, dengan sejumlah modifikasi kecil agar sesuai konteks evaluasi aplikasi *mobile*. Instrumen ini terdiri atas 10 butir pernyataan yang dinilai menggunakan skala *Likert* 1–5. Kesepuluh pernyataan tersebut disusun secara bersilang antara pernyataan positif dan negatif guna meminimalkan bias jawaban responden (pernyataan bernomor ganjil bersifat positif, sementara pernyataan bernomor genap bersifat negatif). Kesepuluh butir pernyataan ini dapat dilihat pada Tabel 1.

Tabel 1. Butir Pernyataan Kuesioner *System Usability Scale* (SUS)

No.	Pernyataan (Skala <i>Likert</i> 1-5)	Tipe
1.	Saya pikir saya ingin sering menggunakan aplikasi ini.	Positif
2.	Saya merasa aplikasi ini tidak perlu dibuat serumit ini.	Negatif
3.	Saya pikir aplikasi ini mudah digunakan.	Positif
4.	Saya pikir saya akan butuh bantuan orang lain untuk bisa menggunakan aplikasi ini.	Negatif
5.	Saya merasa fitur-fitur dalam aplikasi ini terintegrasi dan berjalan dengan sangat baik.	Positif
6.	Saya pikir ada terlalu banyak hal yang tidak konsisten (membingungkan) dalam aplikasi ini.	Negatif
7.	Saya membayangkan kebanyakan orang akan cepat mengerti cara menggunakan aplikasi ini.	Positif
8.	Saya merasa aplikasi ini sangat sulit digunakan.	Negatif
9.	Saya merasa sangat percaya diri saat menggunakan aplikasi ini.	Positif
10.	Saya perlu belajar banyak hal (menonton tutorial) sebelum saya bisa menggunakan aplikasi ini.	Negatif

Setiap pernyataan kuesioner diukur guna mengetahui tingkat persetujuan responden. Rincian bobot skor dan keterangan skala *Likert* yang digunakan dalam evaluasi ini disajikan pada Tabel 2.

Tabel 2. Skala Penilaian *Likert*

Skor	Keterangan
1	Sangat Tidak Setuju (STS)
2	Tidak Setuju (TS)
3	Netral (N)
4	Setuju (S)
5	Sangat Setuju (SS)

Kalkulasi skor akhir instrumen SUS memerlukan penyesuaian bobot yang spesifik pada setiap butir kuesioner. Untuk pernyataan yang bersifat positif (bernomor ganjil), nilai kontribusi didapatkan dengan mengurangi skor jawaban responden dengan angka satu (Skor - 1). Sebaliknya, pada pernyataan yang bersifat negatif (bernomor genap), nilai kontribusi diperoleh dengan cara mengurangkan angka lima dengan skor jawaban responden (5 - Skor). Seluruh nilai kontribusi dari kesepuluh pernyataan tersebut kemudian diakumulasikan menjadi satu. Total penjumlahan ini selanjutnya dikalikan dengan konstanta 2,5 untuk menghasilkan nilai akhir yang terstandarisasi pada rentang skala 0 hingga 100. Rumus perhitungan skor SUS untuk satu responden dituangkan secara matematis pada Persamaan 7.

$$Skor_{SUS} = \left(\sum_{i=1}^{10} C_i \right) \times 2,5 \quad (7)$$

Keterangan: C_i merepresentasikan skor kontribusi dari setiap butir pernyataan ke- i .

Untuk mengevaluasi aplikasi secara keseluruhan, skor SUS dari masing-masing responden kemudian dicari nilai rata-ratanya dengan menjumlahkan semua skor akhir dan membaginya dengan jumlah total responden, sebagaimana dirumuskan pada Persamaan 8.

$$Rata - rata \text{ Skor } SUS = \left(\frac{\sum Skor_{SUS}}{n} \right) \quad (8)$$

Keterangan: n merepresentasikan jumlah total responden.

Skor akhir SUS yang diperoleh kemudian diinterpretasikan menggunakan skema *acceptability range*, *grade*, dan *adjective rating* yang diadaptasi dari [19]. Skema ini mengelompokkan skor ke dalam beberapa kategori penerimaan, mulai dari *Not Acceptable* hingga *Best Imaginable*, dengan skor di atas 68 umumnya dikategorikan berada di atas rata-rata (*above average*) dan menunjukkan tingkat kebergunaan yang layak diterima pengguna.

E. Pemeliharaan

Tahapan ini difokuskan pada perbaikan galat (*bug*) dan pembaruan fitur setelah aplikasi dirilis kepada pengguna. Meskipun tidak dieksekusi secara praktis akibat keterbatasan waktu penelitian, pemeliharaan tetap dicantumkan sebagai komponen struktural yang wajib dipenuhi dalam siklus metode *Waterfall*.

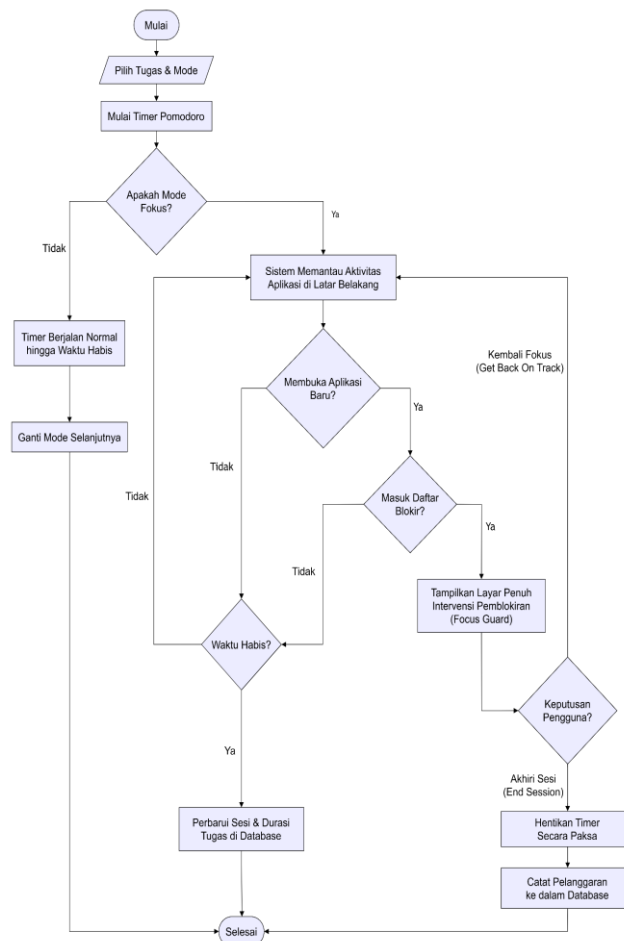
III. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

Berdasarkan tahapan pengumpulan data melalui studi literatur dan observasi lapangan, dirumuskan kebutuhan utama sistem untuk mengatasi prokrastinasi mahasiswa. Hasil observasi mengidentifikasi bahwa pola distraksi terbesar disebabkan oleh aplikasi pihak ketiga pada ponsel pintar saat sesi belajar berlangsung. Analisis ini menyimpulkan bahwa sistem tidak hanya membutuhkan pewaktu *Pomodoro* standar, melainkan mewajibkan integrasi mekanisme intervensi pemblokiran aplikasi secara aktif secara paksa untuk memecah fokus pengguna dari aplikasi pengganggu.

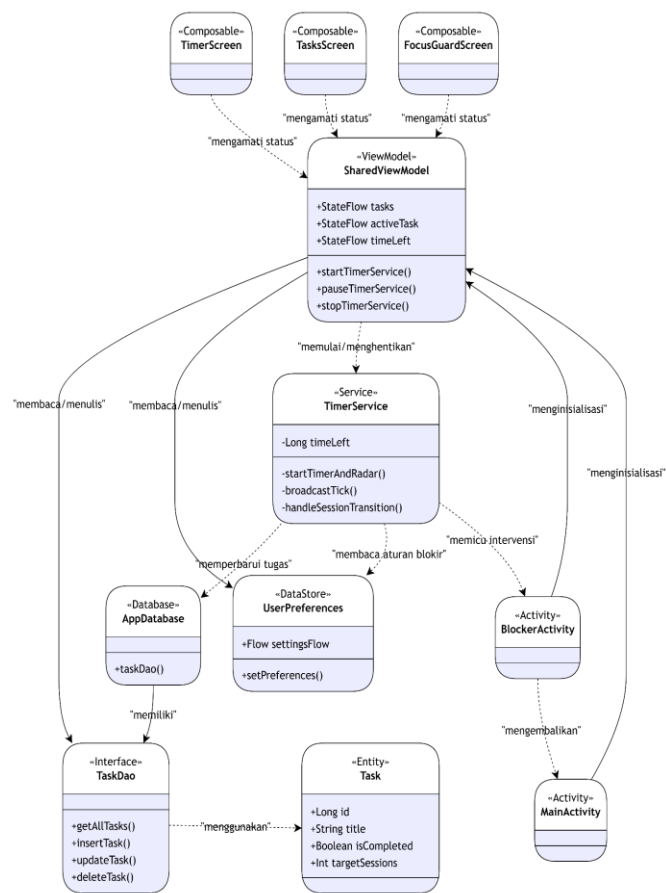
B. Perancangan Sistem

Kebutuhan sistem yang telah diidentifikasi diterjemahkan ke dalam cetak biru (*blueprint*) menggunakan standar pemodelan *Unified Modeling Language* (UML). Alur kerja fungsional aplikasi dari awal hingga akhir, mulai dari pengaturan *timer*, penambahan tugas, hingga pemunculan layar penghalang, divisualisasikan secara detail melalui *Flowchart* yang dapat dilihat pada Gambar 2.



Gambar 2. Flowchart

Susunan dasar kelas dalam sistem (meliputi atribut, metode, serta keterkaitan antarkelas) digambarkan secara visual melalui *Class Diagram* pada Gambar 3.



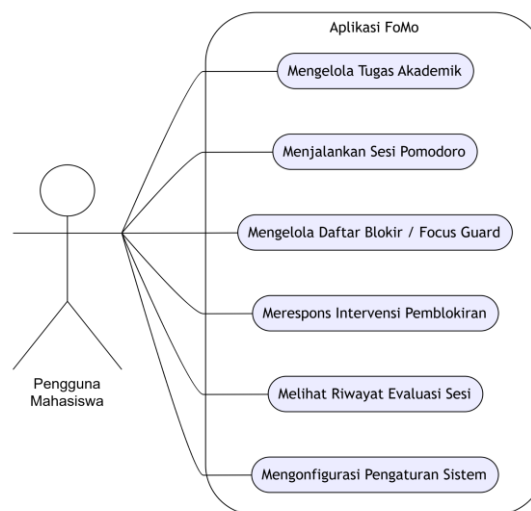
Gambar 3. Class Diagram

Pengelolaan basis data lokal yang menggunakan *Room Database* dimodelkan melalui *Entity Relationship Diagram* (ERD). Pada sistem ini, ERD berfokus memetakan entitas tunggal yaitu manajemen tugas (*Tasks*), mengingat data preferensi pengguna lainnya dikelola secara terpisah menggunakan basis data *Key-Value* (*DataStore*). Visualisasi ERD ini ditampilkan pada Gambar 4.

TASKS			
Long	id	PK	Kunci Utama (Auto Generate)
String	title		Judul tugas
String	description		Deskripsi tugas
String	date		Tanggal pelaksanaan
String	time		Waktu pelaksanaan
String	priority		Tingkat prioritas
String	category		Kategori tugas
Boolean	isCompleted		Status penyelesaian
Int	focusDurationLogged		Total waktu fokus pengguna
Int	targetSessions		Target jumlah sesi Pomodoro
Int	completedSessions		Sesi Pomodoro yang diselesaikan
Boolean	hasViolation		Status pelanggaran sistem blokir

Gambar 4. Entity Relationship Diagram

Cara mahasiswa selaku pengguna berinteraksi dengan berbagai fitur di dalam aplikasi digambarkan secara ringkas pada *Use Case Diagram* pada Gambar 5.



Gambar 5. Use Case Diagram

Selain itu, desain antarmuka (*wireframe*) dibuat menggunakan *Figma*. Implementasi warna pada aplikasi dirancang menggunakan pendekatan psikologis, seperti penggunaan warna *Light Blush* pada mode fokus guna memberikan nuansa yang tenang dan tidak mengganggu saat sesi belajar berlangsung.

C. Implementasi Sistem

Implementasi sistem berfokus pada perancangan antarmuka untuk mendukung penerapan Teknik *Pomodoro* dan pencegahan prokrastinasi digital. Fitur antarmuka aplikasi FoMo terbagi menjadi fitur utama sebagai solusi permasalahan dan fitur pendukung operasional. Rincian implementasi antarmuka pada setiap fitur diuraikan pada pembahasan berikut.

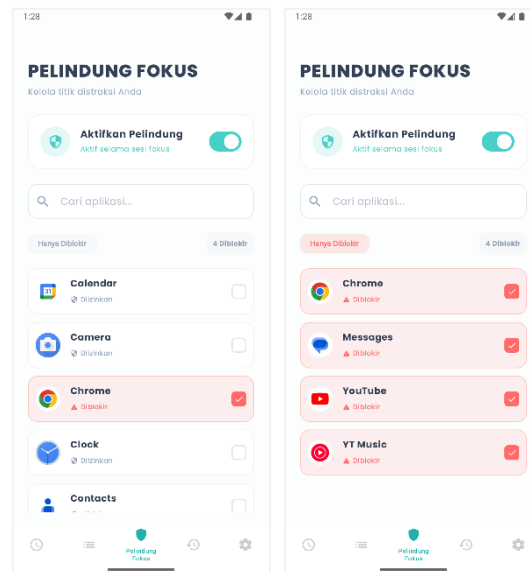
Fitur utama pertama adalah pewaktu (*timer*) *Pomodoro* yang berfungsi sebagai pusat kendali produktivitas mahasiswa. Antarmuka ini menampilkan lingkaran hitungan mundur untuk memfasilitasi mahasiswa dalam menerapkan durasi fokus dan istirahat secara disiplin. Implementasi warna pada aplikasi dirancang menggunakan pendekatan psikologis, seperti penggunaan warna *Light Blush* pada mode fokus guna memberikan nuansa yang tenang dan tidak mengganggu saat sesi belajar berlangsung. Visualisasi antarmuka utama *timer* ini ditampilkan pada Gambar 6.



Gambar 6. Antarmuka Timer Pomodoro Mode Fokus

Fitur utama kedua adalah pelindung fokus (*Focus Guard*) yang berfungsi sebagai mekanisme intervensi pemblokiran aplikasi secara aktif. Melalui halaman ini, pengguna dapat memilih dan mengelola daftar aplikasi

pengganggu (*blacklist*) yang berpotensi memecah konsentrasi belajar. Tampilan halaman pengelolaan daftar blokir aplikasi disajikan pada Gambar 7.



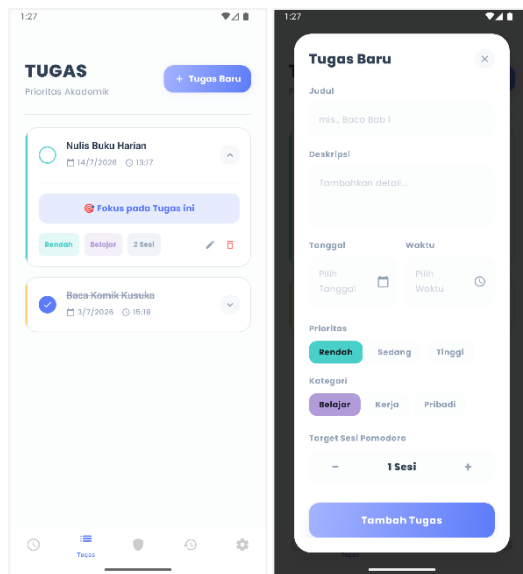
Gambar 7. Antarmuka Pengelolaan Daftar Blokir (*Focus Guard*)

Sebagai bentuk intervensi aktif, ketika pengguna mencoba membuka aplikasi terblokir saat sesi fokus berlangsung, pemantauan latar belakang melalui *UsageStatsManager* akan langsung memicu kemunculan *fullscreen overlay*. Layar peringatan ini menutup antarmuka ponsel secara mutlak dan menyediakan dua tombol pilihan, yaitu tombol kembali fokus (*Get Back On Track*) untuk melanjutkan sesi fokus, atau tombol akhir sesi (*End Session*) untuk menghentikan sesi. Implementasi layar intervensi pemblokiran aplikasi tersebut ditunjukkan pada Gambar 8.

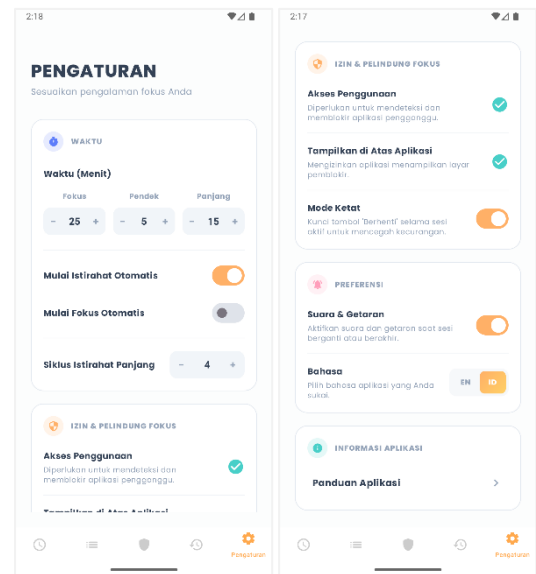


Gambar 8. Antarmuka Layar Intervensi Pemblokiran Aplikasi

Fitur pendukung operasional terdiri atas halaman manajemen tugas (*Tasks*) dan halaman pengaturan (*Settings*). Halaman manajemen tugas memfasilitasi pengguna untuk menyusun target tugas akademik. Sementara itu, halaman pengaturan berfungsi untuk mengonfigurasi parameter waktu, memberikan perizinan sistem operasi, meliputi izin akses penggunaan (*Usage Access*) dan izin tampil di atas aplikasi (*Display Over Other Apps*), mengaktifkan mode ketat (*Strict Mode*), serta mengatur preferensi notifikasi dan bahasa. Tampilan antarmuka manajemen tugas ini divisualisasikan pada Gambar 9, sedangkan konfigurasi sistem diilustrasikan pada Gambar 10.

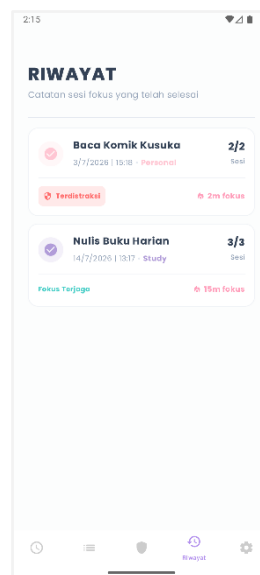


Gambar 9. Antarmuka Manajemen Tugas (*Tasks*)



Gambar 10. Antarmuka Konfigurasi Sistem dan Preferensi Pengguna (*Settings*)

Terakhir, fitur evaluasi pasca-sesi disediakan melalui halaman riwayat (*History*). Halaman ini berfungsi untuk mencatat setiap sesi fokus yang berhasil selesai tanpa pelanggaran sistem blokir. Melalui rekam data ini, pengguna dapat memantau tingkat kedisiplinan dan total waktu produktif secara mandiri. Visualisasi antarmuka riwayat ini ditunjukkan pada Gambar 11.



Gambar 11. Antarmuka Riwayat Sesi dan Evaluasi Performa (*History*)

D. Pengujian

Tahap pengujian sistem bertujuan memvalidasi performa teknis serta mengukur dampak aplikasi FoMo terhadap produktivitas akademik mahasiswa dalam konteks penggunaan nyata. Mengacu pada prosedur dan instrumen yang dipaparkan pada Bab 2, berikut adalah hasil evaluasi yang terbagi ke dalam tiga kelompok temuan, yaitu hasil uji fungsionalitas (*Blackbox Testing*), hasil analisis efektivitas peningkatan produktivitas (*Pre-test* dan *Post-test*), serta hasil penilaian kegunaan antarmuka (*System Usability Scale*).

Pengujian Fungsionalitas

Aspek fungsionalitas diuji menggunakan pendekatan *Blackbox Testing* yang berfokus pada evaluasi antarmuka dan interaksi pengguna berdasarkan data masukan (*input*) guna menghasilkan luaran (*output*) yang sesuai, tanpa menelaah struktur kode di baliknya. Pengujian ini disusun ke dalam 13 skenario uji kasus yang dirancang untuk memastikan seluruh kebutuhan fungsional aplikasi FoMo berjalan bebas dari kesalahan (*error*), meliputi aspek pengaturan izin akses, pengoperasian pewaktu *Pomodoro*, intervensi pemblokiran distraksi, pengelolaan tugas, serta pengelolaan penyimpanan data lokal secara persisten, sebagaimana dirangkum dalam Tabel 3.

Tabel 3. Hasil Pengujian Fungsionalitas (*Blackbox Testing*)

ID	Skenario Pengujian	Langkah Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC-01	Pemberian Izin Akses	Pengguna mengaktifkan izin <i>Usage Access</i> dan <i>Display Over Other Apps</i> pada Halaman Pengaturan.	Sistem menyimpan status pemberian izin dan mengaktifkan fitur pemantauan.	Sistem berhasil mengaktifkan pemantauan latar belakang.	Valid
TC-02	Pengaturan Durasi Waktu	Pengguna mengubah durasi waktu fokus dan istirahat pada Halaman Pengaturan.	Sistem menyimpan pengaturan baru ke memori lokal dan menampilkannya pada Halaman Pewaktu.	Pewaktu menjalankan hitungan mundur sesuai durasi pilihan pengguna.	Valid
TC-03	Penyimpanan Data Saat Memuat Ulang Perangkat (Restart)	Pengguna <i>restart</i> ponsel setelah menambahkan tugas dan memilih daftar aplikasi terblokir.	Sistem mempertahankan seluruh data tugas dan daftar blokir saat aplikasi dibuka kembali.	Sistem memulihkan data dari penyimpanan lokal perangkat secara persisten.	Valid
TC-04	Mode Ketat (<i>Strict Mode</i>)	Pengguna menjalankan sesi fokus pada Halaman Pewaktu dengan kondisi Mode Ketat aktif dari Halaman Pengaturan.	Sistem mengunci tombol berhenti (<i>Stop</i>) selama hitungan mundur berjalan.	Sistem mencegah penghentian pewaktu secara manual dari Halaman Pewaktu.	Valid
TC-05	Pengelolaan Daftar Tugas	Pengguna menambahkan tugas baru melalui formulir pada Halaman Tugas.	Sistem menyimpan data tugas ke basis data lokal dan menampilkannya pada daftar.	Sistem menampilkan tugas baru tersebut sebagai target sesi belajar.	Valid
TC-06	Sesi Fokus Tanpa Tugas	Pengguna memulai sesi fokus tanpa memilih daftar tugas pada Halaman Tugas.	Sistem menjalankan hitungan mundur pewaktu tanpa memicu kendala teknis.	Aplikasi beroperasi secara normal tanpa adanya tugas yang dipilih.	Valid
TC-07	Menjalankan <i>Timer Pomodoro</i>	Pengguna menekan tombol mulai (<i>Play</i>) pada Halaman Pewaktu.	Sistem memulai hitungan mundur sesi fokus sesuai durasi yang telah ditentukan.	Pewaktu menghitung mundur waktu sesi secara akurat hingga habis.	Valid
TC-08	Sistem Notifikasi	Pengguna menyelesaikan hitungan mundur sesi fokus atau istirahat hingga waktu habis.	Sistem mengirimkan notifikasi suara dan pesan pengingat pada bilah pemberitahuan ponsel.	Sistem menampilkan notifikasi pergantian sesi secara tepat waktu.	Valid
TC-09	Pemblokiran Aplikasi Pengganggu	Pengguna membuka aplikasi pihak ketiga yang terblokir saat sesi fokus sedang berlangsung.	Sistem memunculkan Tampilan <i>Overlay</i> Pemblokiran aplikasi untuk menutupi aplikasi tersebut.	Tampilan peringatan menutupi aplikasi pengganggu secara mutlak.	Valid

TC-10	Tombol Kembali Fokus	Pengguna menekan tombol "Kembali Fokus" pada Tampilan Overlay Pemblokiran.	Sistem menutup tampilan peringatan dan mengembalikan pengguna ke aplikasi FoMo.	Sistem melanjutkan hitungan mundur sesi fokus tanpa terputus.	Valid
TC-11	Tombol Akhiri Sesi	Pengguna menekan tombol "Akhir Sesi" pada Tampilan Overlay Pemblokiran.	Sistem menghentikan pewaktu dan menandai sesi tersebut dengan status terdistraksi.	Sistem menghentikan sesi dan mencatat status distraksi pada memori lokal.	Valid
TC-12	Penyimpanan Progres Sesi	Pengguna menutup paksa (<i>Force Close</i>) aplikasi FoMo saat siklus sesi sedang berjalan.	Sistem menyimpan status posisi siklus sesi aktif ke dalam memori perangkat.	Sistem memulihkan posisi siklus pada Halaman Pewaktu saat aplikasi dibuka kembali.	Valid
TC-13	Catatan Riwayat Sesi	Pengguna meninjau daftar aktivitas pada Halaman Riwayat setelah sesi berakhir.	Sistem menampilkan durasi waktu fokus diiringi dengan label "Fokus Terjaga" atau "Terdistraksi".	Halaman Riwayat menampilkan durasi dan label status fokus secara akurat.	Valid

Berdasarkan pengujian fungsionalitas yang dijabarkan pada Tabel 3, seluruh 13 *test case* menunjukkan status Valid. Sesuai dengan parameter validasi yang telah ditetapkan pada Bab 2, perhitungan tingkat kelayakan fungsionalitas aplikasi dieksekusi menggunakan Persamaan 1 sebagai berikut:

$$\text{Persentase Validasi} = \left(\frac{13}{13} \right) \times 100\% = 100\%$$

Hasil perhitungan tersebut mengonfirmasi bahwa luaran sistem telah beroperasi secara akurat dengan tingkat validasi mencapai 100%. Nilai mutlak ini telah melampaui ambang batas kelayakan minimal yang ditetapkan yaitu 90%, sehingga aplikasi FoMo disimpulkan telah memenuhi seluruh spesifikasi kebutuhan fungsional perangkat lunak (yang dipetakan ke dalam 13 skenario uji tersebut) secara keseluruhan. Oleh karena itu, aplikasi dinyatakan sangat layak untuk dilanjutkan pada tahap pengujian efektivitas sistem guna mengukur dampaknya terhadap produktivitas akademik mahasiswa.

Pengujian Efektivitas Sistem

Pengujian efektivitas sistem ditujukan untuk melihat sejauh mana penggunaan aplikasi FoMo berdampak pada produktivitas 35 mahasiswa responden. Tahap awal analisis dilakukan melalui uji normalitas dengan metode *Shapiro-Wilk* yang hasilnya menunjukkan data *Pre-test* berdistribusi normal ($p = 0,375$), sedangkan data *Post-test* tidak berdistribusi normal ($p < 0,001$). Berdasarkan temuan ini, uji komparasi hipotesis dilanjutkan dengan metode non-parametrik *Wilcoxon Signed-rank Test*. Hasil perbandingan skor kuesioner sebelum dan sesudah penggunaan aplikasi ditampilkan pada Tabel 4.

Tabel 4. Perbandingan Rata-rata *Pre-test* dan *Post-test* Kuesioner Produktivitas

Indikator Penilaian	$\bar{x}$ <i>Pre-test</i>	$\bar{x}$ <i>Post-test</i>	$\Delta \bar{x}$
Mengurangi kebiasaan menunda	3,54	4,46	+ 0,92
Mempertahankan tingkat fokus	3,74	4,49	+ 0,75
Pemblokiran distraksi	4,11	4,54	+ 0,43
Efisiensi waktu belajar	3,86	4,46	+ 0,60
Peningkatan produktivitas	3,80	4,60	+ 0,80
Total Rata-rata Keseluruhan	19,06	22,54	+ 3,48

Data pada Tabel 4 menunjukkan tren peningkatan performa belajar yang konsisten di seluruh indikator, ditandai dengan naiknya skor rata-rata total dari 19,06 pada *Pre-test* menjadi 22,54 pada *Post-test* yang menunjukkan peningkatan sebesar 3,48 poin. Dari total 35 responden yang diuji, analisis frekuensi menunjukkan bahwa terdapat 32 mahasiswa yang mengalami peningkatan skor produktivitas (*positive ranks*), sedangkan 3 mahasiswa lainnya tidak menunjukkan perubahan skor (*tied*) antara sebelum dan sesudah intervensi. Signifikansi peningkatan ini kemudian diuji melalui *Wilcoxon Signed-rank Test* yang menghasilkan nilai statistik *W* sebesar 49,5 dengan signifikansi $p <$

0,001. Nilai statistik W yang sangat rendah ini mengonfirmasi secara matematis bahwa arah perbedaan skor didominasi secara mutlak oleh perubahan positif, di mana tingkat produktivitas belajar mahasiswa setelah menggunakan aplikasi FoMo jauh melampaui kondisi awal mereka. Karena nilai p jauh berada di bawah batas 0,05, hipotesis nol dinyatakan ditolak dan hipotesis alternatif diterima. Hasil ini menjadi bukti empiris bahwa intervensi *friction point* pada aplikasi FoMo mampu secara signifikan menekan kebiasaan *distractive multitasking* sekaligus meningkatkan produktivitas akademik mahasiswa.

Pengujian Kebergunaan

Evaluasi kelayakan antarmuka dan kemudahan operasional diukur menggunakan instrumen *System Usability Scale* (SUS) kepada 35 responden yang sama pasca-pengujian simulasi. Rekapitulasi hasil perhitungan kuesioner SUS dijabarkan pada Tabel 5.

Tabel 5. Rekapitulasi Hasil Pengujian SUS

Parameter Statistik	Nilai Skor
Skor SUS Terendah (<i>Minimum</i>)	50,00
Skor SUS Tertinggi (<i>Maximum</i>)	100,00
Rata-rata Skor SUS Akhir (<i>Mean</i>)	68,93

Berdasarkan Tabel 5, perolehan rata-rata skor akhir SUS aplikasi FoMo adalah 68,93. Nilai ini menempatkan aplikasi pada tingkat penerimaan (*acceptability range*) dengan kategori *Marginal High* dengan perolehan *Grade C*, serta predikat kata sifat (*adjective rating*) dengan kategori Baik (*Good*).

Analisis nilai rata-rata per butir pernyataan mengungkap bahwa fitur pewaktu *Pomodoro* dan *Focus Guard* sangat sesuai dengan kebutuhan mahasiswa. Hal ini dibuktikan oleh tingginya minat responden untuk menggunakan aplikasi secara rutin (Pernyataan 1, skor rata-rata 4,31) serta tingginya tingkat kepercayaan diri responden saat mengoperasikan sistem (Pernyataan 9, skor rata-rata 4,23). Di sisi lain, keluhan mengenai asumsi kerumitan sistem (Pernyataan 2, dengan skor rata-rata 2,94) teridentifikasi sebagai dampak dari kurva pembelajaran (*learning curve*) pada tahap awal penggunaan, dimana pengguna membutuhkan adaptasi untuk memberikan otoritas perizinan akses sistem Android yang sensitif, seperti *Usage Access* dan *Display Over Other Apps*. Terlepas dari friksi pengaturan awal tersebut, konvergensi data secara statistik mengonfirmasi bahwa antarmuka aplikasi FoMo telah dirancang dengan intuitif, stabil, dan layak digunakan secara luas untuk menunjang produktivitas akademik mahasiswa.

E. Pemeliharaan

Tahapan pemeliharaan (*maintenance*) merupakan siklus akhir dalam metode *Waterfall*. Meskipun dalam batasan waktu penelitian ini aplikasi belum memasuki fase penggunaan publik jangka panjang, basis kode sistem serta pengelolaan penyimpanan lokal melalui *Room Database* dan *Jetpack DataStore* telah dirancang secara modular dengan arsitektur MVVM agar adaptif terhadap perbaikan galat (*bug fix*) maupun pembaruan *API Level* sistem operasi Android di masa mendatang.

IV. KESIMPULAN

Studi ini berhasil mengembangkan sekaligus menguji aplikasi FoMo sebagai solusi teknologi untuk meningkatkan produktivitas akademik mahasiswa dengan memadukan Teknik *Pomodoro* dengan mekanisme pemblokiran aplikasi melalui fitur *Focus Guard* guna mengatasi masalah distraksi digital yang memicu prokrastinasi akademik. Berdasarkan pengujian fungsionalitas melalui pendekatan *Blackbox Testing* pada 13 skenario uji, sistem terbukti beroperasi secara presisi dan telah memenuhi seluruh spesifikasi kebutuhan perangkat lunak tanpa kendala. Pengukuran efektivitas yang melibatkan 35 responden menunjukkan adanya eskalasi produktivitas belajar yang sangat signifikan. Hal ini dibuktikan secara empiris oleh lonjakan nilai rata-rata dari 19,06 pada fase *Pre-test* menjadi 22,54 pada fase *Post-test*, dengan tingkat signifikansi $p < 0,001$ berdasarkan analisis *Wilcoxon Signed-rank Test*. Lebih lanjut, evaluasi kelayakan antarmuka menggunakan instrumen *System Usability Scale* (SUS) menghasilkan skor akhir 68,93. Perolehan ini menegaskan bahwa desain interaksi aplikasi dinilai intuitif, ramah pengguna, dan sangat layak untuk diimplementasikan secara luas di kalangan mahasiswa.

Walaupun intervensi aplikasi ini terbukti secara empiris efektif dalam mereduksi distraksi digital, masih terdapat sejumlah batasan yang dapat dijadikan landasan bagi pengembangan studi lanjutan. Untuk penelitian di masa mendatang, sangat direkomendasikan adanya penambahan fitur pelacakan aktivitas positif (*activity tracker*) yang terintegrasi dengan dasbor analitik penggunaan aplikasi (*app usage statistics*) pada antarmuka Riwayat. Fungsionalitas inovatif ini ditujukan untuk memantau durasi layar dan merekam jejak pemakaian perangkat lunak secara *real-time*, khususnya penggunaan aplikasi produktif (seperti pembaca dokumen atau buku elektronik) selama sesi fokus

berlangsung. Grafik visualisasi dari analitik ini, dipadukan dengan penambahan fitur berbagi (*share*) pencapaian durasi fokus ke platform media sosial, akan sangat menyempurnakan mekanisme validasi "bukti aktivitas". Penggabungan fungsionalitas ini diharapkan tidak hanya memverifikasi bahwa pengguna belajar secara sadar, tetapi juga membangun iklim akuntabilitas sosial yang secara psikologis mampu memotivasi mahasiswa untuk mempertahankan kedisiplinannya.

UCAPAN TERIMA KASIH

Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT atas segala kemudahan yang diberikan hingga penelitian ini dapat rampung dengan baik. Ucapan terima kasih penulis sampaikan kepada Program Studi Informatika beserta Laboratorium Informatika Universitas Muhammadiyah Sidoarjo atas fasilitas dan dukungan penuh selama proses penelitian berlangsung. Rasa hormat dan terima kasih yang tulus penulis haturkan pula kepada Ibu Ade Eviyanti selaku dosen pembimbing, atas bimbingan, masukan evaluatif, dan kesabaran beliau yang senantiasa menyertai proses pengembangan aplikasi FoMo hingga rampungnya naskah ini. Tidak lupa, penulis menghaturkan terima kasih kepada keluarga tercinta dan semua pihak yang tiada henti memberikan doa, dukungan, serta semangat moral yang begitu berharga.

REFERENSI

- [1] P. Limone, M. Sinatra, F. Ceglie, and L. Monacis, "Examining Procrastination among University Students through the Lens of the Self-Regulated Learning Model," *Behav. Sci. (Basel)*, vol. 10, no. 12, pp. 1–10, 2020, doi: 10.3390/bs10120184.
- [2] T. Hailikari, N. Katajavuori, and H. Asikainen, "Understanding procrastination : A case of a study skills course," *Soc. Psychol. Educ.*, vol. 24, no. 2, pp. 589–606, 2021, doi: 10.1007/s11218-021-09621-2.
- [3] M. T. Hidayat and W. Hasim, "Putting It off until Later: A Survey-Based Study on Academic Procrastination among Undergraduate Students," *J. Educ. Cult. Psychol. Stud.*, vol. 39, no. 28, pp. 5–38, 2023, doi: <https://doi.org/10.7358/ecps-2023-028-taha>.
- [4] R. Nabila, "Faktor-Faktor yang Berpengaruh terhadap Prokrastinasi Akademik : Studi pada Mahasiswa Perguruan Tinggi," *J. Teknol. Pendidik.*, vol. 1, no. 2, pp. 1–10, 2023, doi: 10.47134/jtp.v1i2.169.
- [5] S. A. Zarrin, E. Gracia, and M. P. Paixão, "Prediction of Academic Procrastination by Fear of Failure and Self-Regulation," *Educ. Sci. THEORY Pract.*, vol. 20, no. 3, pp. 34–43, 2020, doi: 10.12738/jestp.2020.3.003.
- [6] E. Duru, M. Balkis, and S. Duru, "Fear of failure and academic satisfaction : the mediating role of emotion regulation difficulties and procrastination," *Eur. J. Psychol. Educ.*, vol. 39, pp. 2901–2914, 2024, doi: 10.1007/s10212-024-00868-9.
- [7] J. F. da C. Júnior, D. de M. C. Bezerra, A. G. de Araújo, and A. S. M. Ramos, "Anti-Procrastination Strategies , Techniques and Tools and Their Interrelation with Self-Regulation and Self-Efficacy," *J. Educ. Learn.*, vol. 13, no. 1, pp. 72–91, 2024, doi: 10.5539/jel.v13n1p72.
- [8] T. Akinci, "Determination of Predictive Relationships Between Problematic Smartphone use , Self-Regulation , Academic Procrastination and Academic Stress Through Modelling," *Int. J. Progress. Educ.*, vol. 17, no. 1, pp. 35–53, 2021, doi: 10.29329/ijpe.2021.329.3.
- [9] L. Deng, Y. Zhou, and J. Broadbent, "Distraction, multitasking and self-regulation inside university classroom," *Educ. Inf. Technol.*, vol. 29, pp. 23957–23979, 2024, doi: 10.1007/s10639-024-12786-w.
- [10] T. Kind, "'I have a five-minute break , I pick up the phone , and then 40 minutes have passed' A study on time management apps, systematic breaks, and mobile phone use while studying," 2024.
- [11] I. N. Iman, Purwantoro, and A. Suharso, "RANCANG BANGUN SISTEM MANAJEMEN WAKTU DENGAN TEKNIK POMODORO MENGGUNAKAN ARSITEKTUR MVVM," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 3, pp. 1700–1706, 2023.
- [12] E. Pratama, A. P. Handayani, F. P. Putri, F. Juliana, and A. Perdana, "PINTARJADWALAI : APLIKASI PRODUKTIVITAS BERBASIS WEB DENGAN INTEGRASI MANAJEMEN TUGAS DAN FOCUS TIMER," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 9, no. 4, pp. 6281–6287, 2025.
- [13] Y. Ahmetoglu, D. P. Brumby, and A. L. Cox, "Bridging the Gap Between Time Management Research and Task Management App Design: A Study on the Integration of Planning Fallacy Mitigation Strategies," *Proc. 3rd Annu. Meet. Symp. Human-Computer Interact. Work (CHIWORK '24)*, pp. 1–14, 2024, doi: 10.1145/3663384.3663404.
- [14] E. J. C. Smits, N. Wenzel, and A. de B. Bruin, "Investigating the Effectiveness of Self-Regulated, Pomodoro, and Flowtime Break-Taking Techniques Among Students," *Behav. Sci. (Basel)*, vol. 15, no. 7,

- pp. 1–30, 2025, doi: 10.3390/bs15070861.
- [15] D. Muriyatmoko, T. Harmini, and A. Rohman, “Implementasi Teknik Pomodoro dan Lockscreen pada Aplikasi Locktimer Berbasis Android,” *METIK J.*, vol. 6, no. 2, pp. 165–171, 2022, doi: 10.47002/metik.v6i2.376.
- [16] D. M. Hamka, L. Fanani, and A. H. Brata, “PENGEMBANGAN APLIKASI FOKUS TIMER DENGAN TRACKING DEVELOPMENT OF FOCUS TIMER APPLICATION WITH SMARTPHONE USAGE TRACKING USING GAMIFICATION APPROACH,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 5, pp. 1137–1144, 2024, doi: 10.25126/jtiik.2024118022.
- [17] S. Padang, E. Sidauro, Anastasya Jesika Tambunan, Y. M. Manullang, and J. Simanjuntak, “Perancangan Aplikasi Manajemen Waktu untuk Meningkatkan Produktivitas Pengguna,” *Remik Ris. dan E-Jurnal Manaj. Inform. Komput.*, vol. 9, no. 2, pp. 488–499, 2025, doi: 10.33395/remik.v9i2.14570.
- [18] T. Prasetyo and A. Eviyanti, “Sistem Informasi Pusat Karir dan Tracer Study di Muhammadiyah Sidoarjo Berbasis Mobile,” pp. 71–85, 2024.
- [19] S. R. Arifin, “System Usability Scale (SUS) implementation in Ruang Baca Virtual – UT Library,” vol. 14, no. 1, pp. 1–8, 2024.