

## Perancangan UI/UX Pada Aplikasi Peningat Tugas Berbasis Android Dengan Fitur Push Notification

Andi Aditya Warman Fachran Talan<sup>a</sup>, Irawati<sup>b</sup>, Ramdaniah<sup>c</sup>

Universitas Muslim Indonesia, Makassar, Indonesia

<sup>a</sup>13020210108@umi.ac.id; <sup>b</sup>irawati.irawati@umi.ac.id; <sup>c</sup>ramdaniah@umi.ac.id

Received: 07-04-2026 | Revised: 20-05-2026 | Accepted: 26-06-2026 | Published: 29-06-2026

### Abstrak

Mahasiswa sering dihadapkan pada kesulitan dalam mengelola jadwal dan tugas akademik yang padat, sehingga memicu tingginya risiko kelalaian dalam memenuhi tenggat waktu (*deadline*). Pencatatan jadwal yang selama ini dilakukan secara konvensional dinilai kurang efektif dan tidak memiliki sistem pengingat yang proaktif. Oleh karena itu, penelitian ini bertujuan untuk merancang antarmuka pengguna (*UI/UX*) dan pemodelan sistem pada aplikasi pengingat tugas harian berbasis Android yang terintegrasi dengan fitur *push notification*. Pendekatan pengembangan perangkat lunak yang digunakan adalah *System Development Life Cycle (SDLC)* dengan metode *Prototyping*. Tahapan yang dilakukan meliputi identifikasi kebutuhan, pembuatan purwarupa tingkat tinggi (*high-fidelity*) menggunakan perangkat lunak Figma, serta pemodelan logika sistem melalui *Unified Modeling Language (UML)*. Aplikasi ini mengusung berbagai fungsionalitas, antara lain opsi penyimpanan lokal dan *cloud*, visualisasi tugas berdasarkan prioritas indikator warna, serta pengaturan peringatan khusus maupun *default*. Tahap evaluasi kelayakan rancangan difokuskan pada fungsionalitas internal tanpa melibatkan kuesioner pengguna *eksternal*, melainkan dengan menerapkan pengujian *Black Box Testing* melalui pendekatan *system walkthrough*. Hasil pengujian terhadap 10 interaksi fungsional utama berhasil mencapai persentase kelayakan sebesar 100%. Kesimpulan dari penelitian ini menunjukkan bahwa rancangan antarmuka dan alur logika aplikasi telah tervalidasi secara fungsional, interaktif, dan siap untuk dijadikan cetak biru (*blueprint*) bagi pengembang perangkat lunak pada tahap pengkodean selanjutnya.

Kata kunci: UI/UX, Aplikasi pengingat, *SDLC*, *Prototyping*, *Push Notification*, *Black-box testing*.

### Pendahuluan

Tingkat mobilitas dan aktivitas akademik di perguruan tinggi menuntut mahasiswa untuk memiliki kemampuan manajemen waktu yang baik. Di tengah kesibukan yang padat, meliputi jadwal perkuliahan, kegiatan organisasi, tugas akademik, hingga pekerjaan sampingan, mahasiswa sering kali mengalami kesulitan dalam mengelola dan memprioritaskan jadwal harian mereka [1]. Pendekatan pengelolaan jadwal yang masih menggunakan metode pencatatan manual atau sekadar mengandalkan ingatan memiliki kelemahan yang signifikan. Keterbatasan tersebut berdampak pada tingginya angka kelalaian mahasiswa dalam menyelesaikan tugas tepat pada tenggat waktu (*deadline*) yang telah ditentukan [2]. Untuk mengatasi masalah ini, pemanfaatan teknologi digital melalui ponsel pintar (*smartphone*) berbasis sistem operasi Android menjadi solusi yang esensial. Kehadiran sebuah aplikasi yang secara spesifik dirancang untuk mengakomodasi jadwal harian, mengkategorikan prioritas, serta memberikan peringatan secara langsung kepada pengguna sangat dibutuhkan. Salah satu teknologi yang dinilai paling efektif dalam memberikan pemberitahuan secara proaktif, terlepas dari apakah pengguna sedang membuka aplikasi atau tidak, adalah integrasi fitur *push notification* [3].

Beberapa penelitian terdahulu telah berupaya mengembangkan sistem manajemen tugas dan pengingat harian untuk meningkatkan produktivitas. Penelitian oleh Surentu (2024) telah merancang purwarupa aplikasi *TaskMate* untuk mahasiswa menggunakan metode *Design Thinking* yang dievaluasi kelayakannya dengan kuesioner *System Usability Scale (SUS)* [4]. Penelitian serupa oleh Isdhianto dkk. (2025) juga berfokus pada perancangan aplikasi pengingat harian berbasis Figma yang terintegrasi dengan jadwal kuliah menggunakan metode *Design Thinking* [5]. Di sisi lain, Firdaus dan Manikam (2025) merancang aplikasi *To-Do-List "MyList"* yang menyertakan fitur prioritas, kategori, dan statistik penyelesaian tugas menggunakan metodologi *Waterfall* [6]. Selain itu, penelitian yang dilakukan oleh Elyza (2023) berhasil merancang bangun teknologi *push notification* sebagai *e-reminder* akademik mahasiswa untuk meminimalisasi kelupaan terhadap jadwal-jadwal penting perguruan tinggi [7].

Meskipun telah banyak penelitian terkait aplikasi pengingat, sebagian besar penelitian tersebut (khususnya yang berbasis *Design Thinking*) berfokus pada proses evaluasi persepsi, empati, dan kepuasan pengguna akhir

melalui penyebaran kuesioner [[8], [9]. Perbedaan utama (*gap*) penelitian ini dengan penelitian-penelitian sebelumnya terletak pada penekanan aspek rekayasa perangkat lunak secara struktural. Penelitian ini difokuskan murni pada perancangan antarmuka pengguna (*UI/UX*) serta pembangunan cetak biru (*blueprint*) logika sistem, tanpa mewajibkan evaluasi psikologis pengguna awam maupun implementasi pengkodean bahasa pemrograman secara langsung.

Berdasarkan rumusan masalah di atas, penelitian ini mengusulkan perancangan aplikasi pengingat tugas harian menggunakan pendekatan *System Development Life Cycle (SDLC)* dengan metode *Prototyping* [10]. Metode *Prototyping* dipilih karena kemampuannya dalam memfasilitasi pembuatan model fisik (purwarupa visual) secara cepat guna menyamakan persepsi dasar antara perancang dan gambaran sistem nyata [11], [12]. Luaran dari penelitian ini adalah sebuah purwarupa tingkat tinggi (*high-fidelity*) yang dibangun melalui Figma serta pemodelan *Unified Modeling Language (UML)* yang kuat. Validasi fungsionalitas sistem akan diukur secara internal menggunakan pengujian *Black Box Testing* untuk memastikan fungsionalitas logikanya berjalan optimal [13]. Pendekatan ini dilakukan guna menjamin bahwa rancangan aplikasi yang dihasilkan benar-benar valid dan siap sebelum diserahkan kepada *programmer* untuk tahap pengembangan basis data dan sistem nyata di masa mendatang.

### Metode

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak *System Development Life Cycle (SDLC)* dengan menerapkan metode *Prototyping* [6], [7]. Metode *Prototyping* dipilih karena kemampuannya dalam memfasilitasi perancangan model visual secara cepat dan literatif, sehingga pengembang dapat mengevaluasi antarmuka dan pengalaman pengguna secara langsung sebelum memasuki tahap penulisan kode program (*coding*) yang kompleks [2], [3].

Sesuai dengan batasan masalah yang difokuskan pada perancangan, instrumen dan teknologi utama yang digunakan dalam penelitian ini meliputi perangkat lunak Figma untuk merancang desain *User Interface (UI)* dan *User Experience (UX)* ke dalam bentuk purwarupa interaktif tingkat tinggi (*high-fidelity*) [5], [6]. Selain itu, *Unified Modeling Language (UML)* digunakan sebagai alat pemodelan visual untuk mendefinisikan arsitektur logika sistem yang mencakup *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram* [7], [8]. Secara terstruktur, tahapan pelaksanaan metode *Prototyping* dalam penelitian ini dibagi menjadi tiga fase utama [14], [15], yaitu:

#### A. Analisis Kebutuhan (*listen to customer*)

Tahap ini merupakan proses identifikasi masalah dan pengumpulan data kebutuhan sistem. Berbeda dengan penelitian konvensional yang menggunakan penyebaran kuesioner berskala besar, analisis kebutuhan pada penelitian ini dilakukan secara spesifik melalui observasi fungsional dan studi literatur terhadap permasalahan manajemen waktu mahasiswa. Fokus utama pada tahap ini adalah merumuskan kebutuhan fungsionalitas aplikasi, seperti manajemen tugas harian, mekanisme sinkronisasi basis data (*cloud* dan lokal), serta parameter pemicu peringatan otomatis menggunakan teknologi *push notification*.

#### B. Perancangan dan Pembuatan Purwarupa (*build/revise Mock-up*)

Berdasarkan spesifikasi kebutuhan yang telah diidentifikasi, tahap selanjutnya adalah menerjemahkan kebutuhan tersebut ke dalam rancangan arsitektur sistem menggunakan pemodelan UML [11]. Setelah logika sistem terbentuk, proses dilanjutkan dengan merancang tata letak (*wireframe*) dan antarmuka visual menggunakan perangkat lunak Figma [12]. Purwarupa yang dihasilkan dirancang agar memiliki interaktivitas tinggi yang merepresentasikan alur kerja (*user flow*) aplikasi nyata, mulai dari halaman autentikasi, *dashbor* tugas, hingga simulasi munculnya *push notification* pada layar perangkat Android pengguna.

#### C. Pengujian Sistem (*customer test drives mock up*)

Fase terakhir adalah melakukan validasi terhadap purwarupa yang telah dibangun untuk memastikan bahwa rancangan sistem bebas dari kesalahan logika fungsional. Pengujian ini menggunakan metode *Black Box Testing* melalui pendekatan *system walkthrough* pada purwarupa Figma. Pengujian *Black Box* difokuskan murni pada evaluasi kesesuaian antara input yang diberikan pengguna dengan output yang dihasilkan oleh sistem, tanpa meninjau struktur kode sumber di balik layar [13], [16], [17].

Untuk mengukur tingkat keberhasilan dan validitas fungsionalitas rancangan, evaluasi dilakukan terhadap sejumlah skenario interaksi utama di dalam aplikasi. Hasil dari pengujian tersebut kemudian dikalkulasikan menggunakan persamaan matematis persentase kelayakan sistem sebagai berikut:

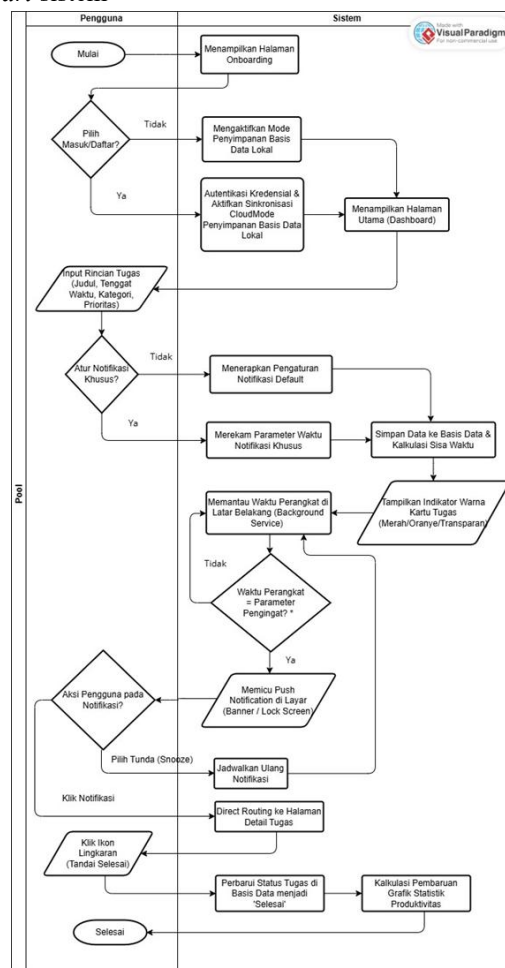
Apabila hasil kalkulasi persentase kelayakan mencapai tingkat maksimal, maka purwarupa antarmuka dan pemodelan UML tersebut dinyatakan valid, logis, dan sepenuhnya layak untuk dijadikan cetak biru (*blueprint*) pada tahap implementasi pengkodean oleh pengembang perangkat lunak selanjutnya.

## Perancangan

### A. Analisa Sistem

Analisa sistem merupakan tahapan untuk mengidentifikasi permasalahan dari sistem yang akan dirancang serta merumuskan cara penyelesaian masalah (*problem solving*). Pengelolaan jadwal dan tugas akademik mahasiswa saat ini sebagian besar masih dilakukan secara konvensional. Pendekatan ini memiliki kelemahan signifikan, yaitu tingginya risiko kelalaian yang menyebabkan mahasiswa melewati tenggat waktu (*deadline*). Oleh karena itu, diusulkan rancang bangun aplikasi pengingat tugas harian berbasis mobile Android yang menyediakan fitur pengelolaan tugas terstruktur dan peringatan proaktif melalui teknologi *push notification* [18]. Dalam konteks manajemen tugas, aplikasi ini mengakomodasi fleksibilitas akses pengguna. Sistem menyediakan opsi penyimpanan data secara lokal bagi pengguna yang melewati tahap registrasi, serta integrasi penyimpanan berbasis komputasi awan (*cloud*) bagi pengguna yang melakukan autentikasi. Sistem juga menerapkan klasifikasi visual berbasis indikator warna pada antarmuka untuk mengurangi beban kognitif pengguna dalam mengidentifikasi tingkat urgensi tugas (warna merah untuk tugas yang terlewat, oranye untuk tugas yang mendekati tenggat waktu, dan transparan untuk kondisi normal atau selesai).

### B. Prosedur pembacaan/flowchart sistem



Gambar 1. Flowchart prosedur pembacaan sistem

Prosedur pembacaan sistem digambarkan menggunakan flowchart untuk memetakan alur kendali (*control flow*) arsitektur perangkat lunak dari awal pengguna berinteraksi hingga sistem menghasilkan keluaran. Flowchart ini dibagi menjadi dua lajur (*swimlane*), yaitu lajur kiri untuk interaksi fisik pengguna (mahasiswa) dan lajur kanan untuk proses komputasi sistem. Alur dimulai dengan halaman *onboarding* di mana pengguna memilih 'Masuk/Daftar' untuk sinkronisasi *cloud*, atau 'Lewati' untuk penggunaan penyimpanan lokal. Setelah masuk ke halaman utama, pengguna dapat menambah tugas baru dengan menginputkan rincian judul, tenggat waktu, kategori, prioritas, dan pengaturan waktu notifikasi khusus. Sistem memproses data tersebut, mengkalkulasi sisa waktu secara otomatis, dan memberikan output indikator warna pada kartu tugas. Di latar belakang (*background service*), sistem secara berkala memantau waktu perangkat. Jika waktu perangkat sesuai dengan parameter waktu pengingat, sistem akan memicu *Notification Manager* untuk menampilkan *push notification* di layar. Jika pengguna mengklik notifikasi tersebut, sistem akan melakukan *direct routing* ke halaman detail tugas. Prosedur pembacaan sistem ini secara utuh ditunjukkan pada Gambar 1.

Pengguna membuka aplikasi, lalu sistem menampilkan *onboarding*. Setelah itu pengguna memutuskan apakah akan masuk/daftar atau tidak. Jika tidak, sistem mengaktifkan penyimpanan lokal; jika ya, sistem melakukan autentikasi dan sinkronisasi *cloud*. Keduanya berakhir di *dashboard*. Dari *dashboard*, pengguna mengisi rincian tugas berupa judul, tenggat waktu, kategori, dan prioritas. Selanjutnya pengguna memutuskan apakah ingin notifikasi khusus atau cukup notifikasi *default*. Sistem lalu menyimpan data tugas, menghitung sisa waktu, dan menampilkan warna indikator pada kartu tugas. Setelah itu sistem memantau waktu perangkat di *background*. Saat waktu pengingat tercapai, sistem memunculkan *push notification*. Pengguna kemudian bisa menunda notifikasi atau membukanya. Jika dibuka, sistem mengarahkan ke detail tugas. Dari sana pengguna menekan ikon lingkaran untuk menandai tugas selesai. Sistem lalu memperbarui status tugas menjadi selesai dan menghitung ulang grafik statistik produktivitas. Proses pun berakhir.

#### C. Rancangan input/output (*prototyping system*)

Berdasarkan alur logika sistem yang telah ditetapkan, perancangan antarmuka pengguna (*User Interface*) ditransformasikan ke dalam bentuk visual menggunakan perangkat lunak Figma [3]. Pada bagian ini, peneliti melakukan proses pembuatan *prototype* tingkat tinggi (*High-Fidelity*) yang didasarkan pada aliran pengguna (*user flow*) yang telah dirancang sebelumnya [4]. Adapun halaman antarmuka yang dibuat dapat dilihat mulai dari Gambar 2.

#### D. Struktur Database

Tabel 1. Struktur users

| Kolom                   | Tipe Data                      | Key    | Keterangan                          |
|-------------------------|--------------------------------|--------|-------------------------------------|
| Pengguna                | BIGINT / UUID                  | PK     | ID pengguna                         |
| nama_lengkap            | VARCHAR(100)                   |        | Nama pengguna                       |
| email                   | VARCHAR(150)                   | UNIQUE | Email untuk login                   |
| Kata sandi_hash         | VARCHAR(255)                   |        | Password yang sudah di-hash         |
| auth_provider           | VARCHAR(30)                    |        | local / google / apple / lainnya    |
| sudah_verifikasi        | BOOLEAN                        |        | Status verifikasi akun              |
| mode_penyimpanan        | ENUM('local','cloud','hybrid') |        | Mode penyimpanan                    |
| sinkronisasi_diaktifkan | BOOLEAN                        |        | Sinkronisasi cloud aktif atau tidak |
| onboarding_completd     | BOOLEAN                        |        | Status onboarding                   |
| timezone                | VARCHAR(50)                    |        | Zona waktu pengguna                 |
| created_at              | TIMESTAMP                      |        | Waktu dibuat                        |
| updated_at              | TIMESTAMP                      |        | Waktu diubah                        |

Tabel 2. Struktur *categories*

| Kolom   | Tipe Data     | Key           | Keterangan       |
|---------|---------------|---------------|------------------|
| id      | BIGINT / UUID | PK            | ID kategori      |
| user_id | BIGINT / UUID | FK → users.id | Pemilik kategori |

|            |              |  |                |
|------------|--------------|--|----------------|
| name       | VARCHAR(100) |  | Nama kategori  |
| color_code | VARCHAR(20)  |  | Warna kategori |
| icon_name  | VARCHAR(50)  |  | Ikon kategori  |
| created_at | TIMESTAMP    |  | Waktu dibuat   |
| updated_at | TIMESTAMP    |  | Waktu diubah   |

Tabel 3. Struktur *priorities*

| Kolom       | Tipe Data   | Key | Keterangan                     |
|-------------|-------------|-----|--------------------------------|
| id          | INT         | PK  | ID prioritas                   |
| name        | VARCHAR(30) |     | Contoh: Rendah, Sedang, Tinggi |
| level_value | INT         |     | Nilai urutan prioritas         |

Tabel 4. Struktur *tasks*

| Kolom             | Tipe Data                                        | Key                | Keterangan                      |
|-------------------|--------------------------------------------------|--------------------|---------------------------------|
| id                | BIGINT / UUID                                    | PK                 | ID tugas                        |
| user_id           | BIGINT / UUID                                    | FK → users.id      | Pemilik tugas                   |
| category_id       | BIGINT / UUID                                    | FK → categories.id | Kategori tugas                  |
| priority_id       | INT                                              | FK → priorities.id | Prioritas tugas                 |
| title             | VARCHAR(200)                                     |                    | Judul tugas                     |
| description       | TEXT                                             |                    | Deskripsi detail tugas          |
| due_date          | DATE                                             |                    | Tanggal tenggat                 |
| due_time          | TIME                                             |                    | Jam tenggat                     |
| due_datetime      | TIMESTAMP                                        |                    | Gabungan tanggal + waktu        |
| status            | ENUM('pending','in_progress','done','cancelled') |                    | Status tugas                    |
| reminder_mode     | ENUM('default','custom','none')                  |                    | Jenis pengingat                 |
| color_indicator   | ENUM('red','orange','transparent')               |                    | Indikator warna kartu           |
| is_completed      | BOOLEAN                                          |                    | Status selesai                  |
| completed_at      | TIMESTAMP                                        |                    | Waktu tugas selesai             |
| remaining_minutes | INT                                              |                    | Sisa waktu, bisa dihitung ulang |
| source_mode       | ENUM('local','cloud')                            |                    | Sumber penyimpanan              |
| created_at        | TIMESTAMP                                        |                    | Waktu dibuat                    |
| updated_at        | TIMESTAMP                                        |                    | Waktu diubah                    |
| deleted_at        | TIMESTAMP                                        |                    | Soft delete                     |

Tabel 5. Struktur *user notification settings*

| Kolom                    | Tipe Data     | Key           | Keterangan                     |
|--------------------------|---------------|---------------|--------------------------------|
| id                       | BIGINT / UUID | PK            | ID setting                     |
| user_id                  | BIGINT / UUID | FK → users.id | Pemilik setting                |
| default_reminder_minutes | INT           |               | Misal 30 menit sebelum tenggat |
| enable_push_notification | BOOLEAN       |               | Aktif/nonaktif notifikasi      |
| enable_lock_screen       | BOOLEAN       |               | Tampilkan di lock screen       |
| enable_banner            | BOOLEAN       |               | Tampilkan banner               |
| snooze_default_minutes   | INT           |               | Default snooze                 |
| created_at               | TIMESTAMP     |               | Waktu dibuat                   |
| updated_at               | TIMESTAMP     |               | Waktu diubah                   |

Tabel 6. Struktur *tasks reminder*

| Kolom         | Tipe Data                         | Key           | Keterangan      |
|---------------|-----------------------------------|---------------|-----------------|
| id            | BIGINT / UUID                     | PK            | ID reminder     |
| task_id       | BIGINT / UUID                     | FK → tasks.id | Tugas terkait   |
| reminder_type | ENUM('default','custom','snooze') |               | Jenis pengingat |

|                |           |  |                               |
|----------------|-----------|--|-------------------------------|
| remind_at      | TIMESTAMP |  | Waktu notifikasi akan dipicu  |
| offset_minutes | INT       |  | Berapa menit sebelum deadline |
| is_active      | BOOLEAN   |  | Reminder aktif atau tidak     |
| is_sent        | BOOLEAN   |  | Sudah terkirim atau belum     |
| sent_at        | TIMESTAMP |  | Waktu dikirim                 |
| created_at     | TIMESTAMP |  | Waktu dibuat                  |
| updated_at     | TIMESTAMP |  | Waktu diubah                  |

Tabel 7. Struktur *notification logs*

| Kolom              | Tipe Data                                                     | Key                    | Keterangan        |
|--------------------|---------------------------------------------------------------|------------------------|-------------------|
| id                 | BIGINT / UUID                                                 | PK                     | ID log notifikasi |
| user_id            | BIGINT / UUID                                                 | FK → users.id          | Pengguna penerima |
| task_id            | BIGINT / UUID                                                 | FK → tasks.id          | Tugas terkait     |
| reminder_id        | BIGINT / UUID                                                 | FK → task_reminders.id | Reminder sumber   |
| notification_title | VARCHAR(200)                                                  |                        | Judul notifikasi  |
| notification_body  | TEXT                                                          |                        | Isi notifikasi    |
| channel            | ENUM('banner','lockscreen','push')                            |                        | Jenis tampilan    |
| delivery_status    | ENUM('queued','sent','failed','opened','dismissed','snoozed') |                        | Status notifikasi |
| delivered_at       | TIMESTAMP                                                     |                        | Waktu dikirim     |
| opened_at          | TIMESTAMP                                                     |                        | Waktu diklik      |
| created_at         | TIMESTAMP                                                     |                        | Waktu dibuat      |

Tabel 8. Struktur *notification actions*

| Kolom               | Tipe Data                                   | Key                       | Keterangan           |
|---------------------|---------------------------------------------|---------------------------|----------------------|
| id                  | BIGINT / UUID                               | PK                        | ID aksi              |
| notification_log_id | BIGINT / UUID                               | FK → notification_logs.id | Log notifikasi       |
| task_id             | BIGINT / UUID                               | FK → tasks.id             | Tugas terkait        |
| action_type         | ENUM('open','snooze','dismiss','mark_done') |                           | Jenis aksi           |
| snooze_minutes      | INT                                         |                           | Isi jika aksi snooze |
| action_at           | TIMESTAMP                                   |                           | Waktu aksi dilakukan |

Tabel 9. Struktur *task status histories*

| Kolom      | Tipe Data     | Key           | Keterangan             |
|------------|---------------|---------------|------------------------|
| id         | BIGINT / UUID | PK            | ID riwayat             |
| task_id    | BIGINT / UUID | FK → tasks.id | Tugas terkait          |
| old_status | VARCHAR(30)   |               | Status lama            |
| new_status | VARCHAR(30)   |               | Status baru            |
| changed_by | BIGINT / UUID | FK → users.id | Pengguna yang mengubah |
| changed_at | TIMESTAMP     |               | Waktu perubahan        |
| note       | TEXT          |               | Catatan tambahan       |

Tabel 10. Struktur *task productivity stats*

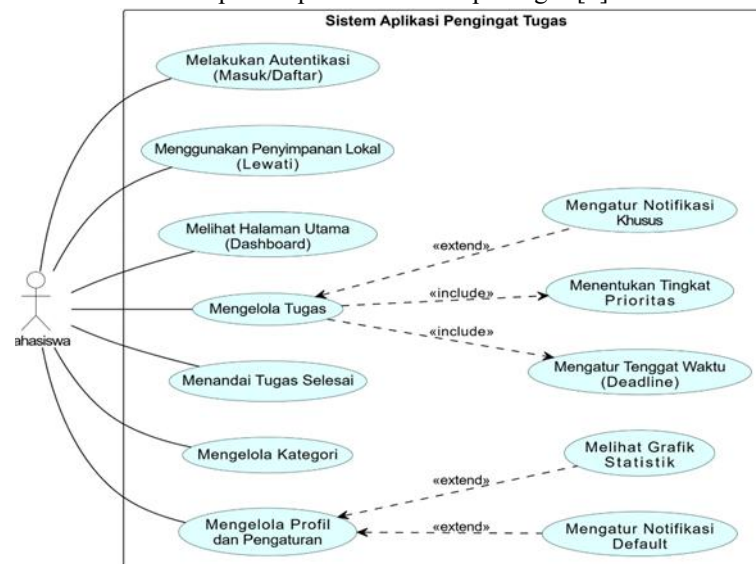
| Kolom           | Tipe Data     | Key           | Keterangan                |
|-----------------|---------------|---------------|---------------------------|
| id              | BIGINT / UUID | PK            | ID statistik              |
| user_id         | BIGINT / UUID | FK → users.id | Pengguna                  |
| stat_date       | DATE          |               | Tanggal statistik         |
| total_tasks     | INT           |               | Total tugas               |
| completed_tasks | INT           |               | Total tugas selesai       |
| pending_tasks   | INT           |               | Total tugas belum selesai |
| overdue_tasks   | INT           |               | Total tugas terlambat     |
| completion_rate | DECIMAL(5,2)  |               | Persentase penyelesaian   |

## Pemodelan

Pemodelan sistem merupakan tahapan krusial dalam rekayasa perangkat lunak yang berfungsi untuk merepresentasikan objek nyata ke dalam bentuk diagram visual. Dalam penelitian ini, pemodelan logika sistem dibangun menggunakan *Unified Modeling Language (UML)*. *UML* adalah sebuah bahasa pemodelan visual yang dirancang khusus untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan hasil analisis serta desain sebuah sistem berorientasi objek [19]. Dalam proses pengembangan perangkat lunak, *UML* bertindak sebagai cetak biru (*blueprint*) awal. Dengan adanya *blueprint* ini, alur kerja dan logika sistem dapat divalidasi terlebih dahulu, sehingga meminimalisasi terjadinya kesalahan struktural saat aplikasi dikembangkan [20]. Sesuai dengan kebutuhan perancangan sistem informasi berbasis mobile [3], pemodelan *UML* yang digunakan dalam penelitian ini terdiri dari empat diagram utama, yaitu:

### A. Use case diagram

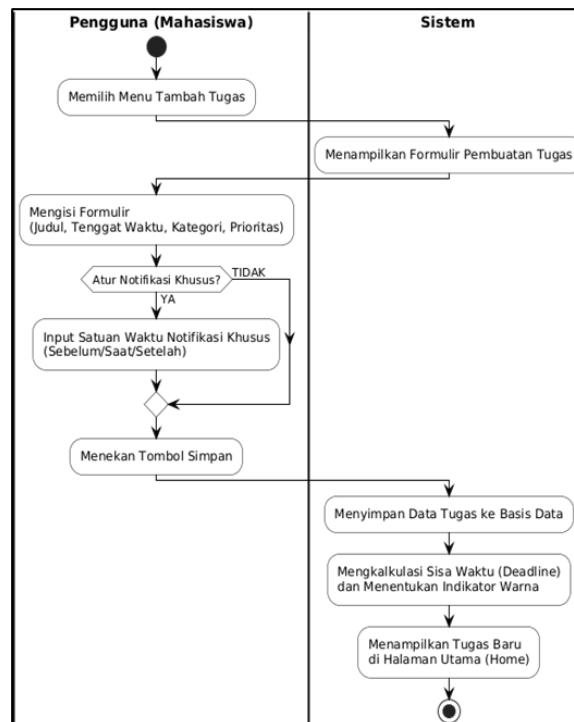
*Use Case Diagram* mendeskripsikan interaksi fungsional antara aktor dengan sistem aplikasi pengingat tugas [21]. Dalam pemodelan ini, aktor utamanya adalah pengguna (mahasiswa). Fungsionalitas yang dapat dilakukan oleh pengguna meliputi: masuk/daftar akun untuk sinkronisasi basis data *cloud*, memilih opsi "lewat" untuk penggunaan penyimpanan lokal, mengelola tugas (tambah, ubah, hapus, serta tandai selesai dengan cepat menggunakan interaksi ikon lingkaran), membuat kategori baru, serta mengatur notifikasi. Khusus untuk pengaturan notifikasi, *use case* ini dibagi menjadi dua fungsionalitas turunan (*include*), yaitu fungsionalitas untuk mengatur notifikasi *default* secara *global* pada menu *profil* dan mengatur notifikasi khusus secara spesifik pada halaman input tugas [4].



Gambar 2. Use case diagram aplikasi pengingat tugas

### B. Activity diagram

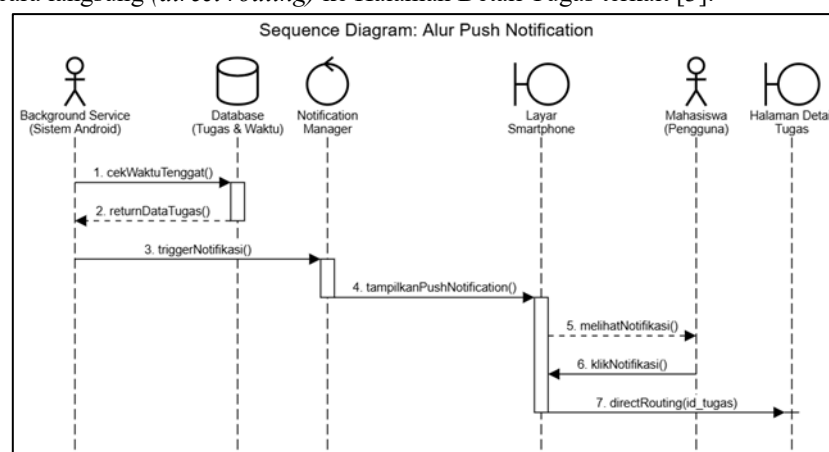
*Activity Diagram* menggambarkan aliran kerja (*control flow*) dari suatu aktivitas di dalam sistem secara terstruktur [22], [23]. Salah satu aliran aktivitas paling krusial dalam perancangan ini adalah proses penambahan tugas baru. Aktivitas dimulai saat pengguna menekan tombol "Tambah" di *navigation bar*. Sistem akan menampilkan formulir isian yang terdiri dari judul, deskripsi, tenggat waktu (tanggal dan jam), kategori, dan tingkat prioritas[4]. Jika pengguna memilih untuk menambahkan notifikasi khusus, sistem memproses alur input satuan waktu peringatan (sebelum, saat, atau setelah *deadline*). Setelah pengguna menyimpan tugas, sistem secara otomatis akan mengevaluasi sisa waktu *deadline* untuk menentukan *output* indikator warna kartu tugas (merah untuk *overdue*, oranye untuk mendekati tenggat, atau transparan untuk kondisi normal) dan menyimpannya ke dalam basis data [1].



Gambar 3. Activity diagram alur penambahan tugas

### C. Sequence diagram

*Sequence Diagram* memodelkan komunikasi antar-objek antarmuka berdasarkan urutan waktu [24]. Diagram ini secara khusus digunakan untuk memetakan skenario berjalannya fitur *push notification* pada sistem operasi Android. Urutan dimulai ketika *Service* pada latar belakang (*background*) Android secara berkala memeriksa waktu saat ini dan mencocokkannya dengan data tenggat waktu tugas di dalam basis data [4]. Ketika waktu sistem sesuai dengan parameter jadwal notifikasi (baik *default* maupun khusus), basis data akan mengirimkan pesan (*message*) pemicu ke sistem antarmuka. Sistem kemudian memerintahkan *Notification Manager* pada Android untuk menghasilkan keluaran visual berupa *banner* mengambang (saat layar aktif) atau *lock screen notification* (saat layar terkunci) [1]. Jika pengguna berinteraksi dengan menekan area notifikasi tersebut, *sequence* akan diakhiri dengan mengarahkan pengguna secara langsung (*direct routing*) ke Halaman Detail Tugas terkait [3].

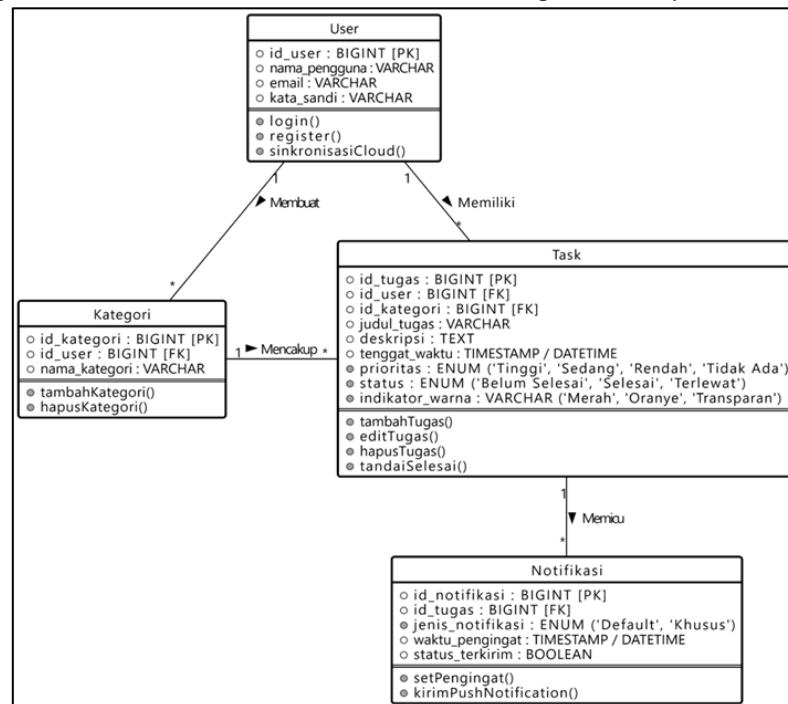


Gambar 4. Sequence diagram push notification

### D. Class diagram

*Class Diagram* merepresentasikan struktur basis data dan hubungan antar-entitas di dalam model desain aplikasi [25]. Skema basis data pada aplikasi pengingat tugas ini dirancang ke dalam empat tabel (kelas) utama, yaitu: tabel *User* untuk menyimpan kredensial akun pengguna yang melakukan sinkronisasi *cloud*;

tabel Kategori untuk menyimpan daftar klasifikasi tugas; tabel Tugas yang memuat entitas utama seperti judul, deskripsi, parameter *deadline*, tingkat prioritas, dan status penyelesaian; serta tabel Notifikasi yang menyimpan opsi pengaturan waktu peringatan khusus. Relasi antar-kelas ini dibangun secara berkesinambungan untuk memastikan bahwa ketika sebuah tugas ditandai selesai atau dihapus, sistem pengingat yang terelasi di tabel notifikasi akan secara otomatis diperbarui *statyausn*.



Gambar 5. Class diagram skema basis data

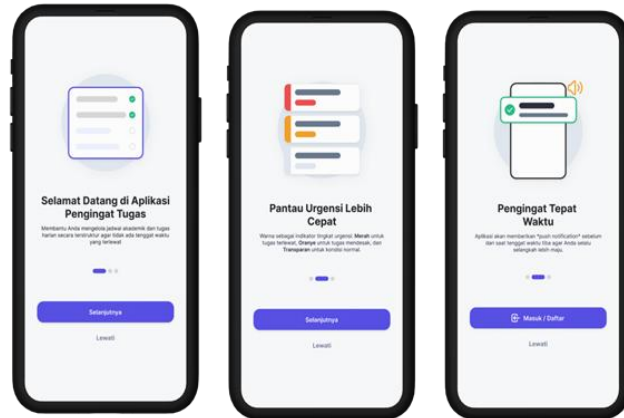
#### E. Pengujian sistem (*black box testing*)

Setelah pemodelan dan perancangan antarmuka selesai dibangun, tahapan krusial selanjutnya adalah melakukan pengujian sistem. Pengujian ini menggunakan metode *Black Box Testing* melalui pendekatan *system walkthrough* pada purwarupa interaktif di Figma. Pengujian difokuskan pada pemenuhan spesifikasi fungsional input dan *output*, tanpa meninjau struktur kode program.

Skenario pengujian internal dilakukan terhadap 10 interaksi fungsional utama, yang meliputi: (1) Alur pendaftaran dan login pengguna; (2) Fungsionalitas tab navigasi halaman utama; (3) *Input form* tambah tugas baru; (4) Kalkulasi penentuan indikator warna batas waktu tugas; (5) Fungsionalitas pengaturan notifikasi khusus; (6) Simulasi keluaran *push notification* pada layar; (7) *Direct routing* dari notifikasi ke detail tugas; (8) Pembaruan status tugas menjadi selesai; (9) Logika penyimpanan lokal saat melewati proses *login*; dan (10) Tampilan pembaruan metrik pada halaman *dashboard profil*.

Berdasarkan hasil pengujian pada keseluruhan skenario tersebut, purwarupa sistem berhasil memberikan respon (*output*) yang sesuai dengan rancangan alur logika yang diharapkan pada setiap interaksi (*input*). Untuk mengukur tingkat validitas rancangan tersebut, digunakan perhitungan persentase kelayakan menggunakan persamaan yang telah ditetapkan pada bagian metode:

Hasil perhitungan menunjukkan angka 100% yang mengindikasikan bahwa rancangan antarmuka pengguna (UI) beserta alur pengalaman pengguna (UX) pada aplikasi pengingat tugas harian ini berfungsi secara optimal dan sangat layak diimplementasikan lebih lanjut ke dalam tahapan pengkodean (*coding*)

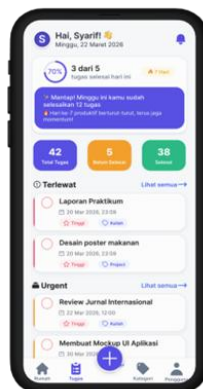
Gambar 6. Rancangan antarmuka *onboarding*

Gambar 6 merupakan antarmuka untuk halaman *onboarding* aplikasi, yang berfungsi sebagai pengenalan awal kepada pengguna mengenai fitur-fitur utama yang tersedia. Pada halaman ini terdapat tombol ‘Selanjutnya’ yang berfungsi untuk masuk ke layar *onboarding* berikutnya, dan juga tombol ‘Lewati’ untuk langsung masuk ke halaman utama dengan menggunakan basis data lokal. Bagi pengguna yang ingin menyinkronkan data, pengguna dapat memilih opsi untuk masuk ke akun. Selanjutnya, terdapat halaman *login* pengguna yang dapat dilihat pada Gambar 7.



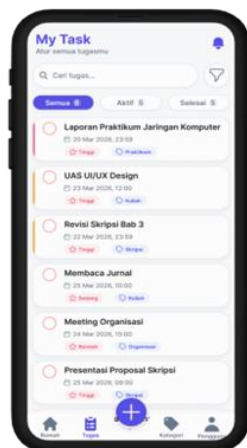
Gambar 7. Rancangan antarmuka login dan autentikasi

Gambar 7 menampilkan halaman login yang berfungsi sebagai gerbang autentikasi. Halaman ini dirancang untuk memudahkan pengguna dalam mengakses aplikasi dengan memasukkan data berupa alamat email dan kata sandi, serta menyediakan navigasi menuju halaman pendaftaran bagi pengguna baru. Setelah melalui tahap autentikasi dan sinkronisasi data dengan server, pengguna akan diarahkan langsung ke halaman beranda, seperti yang ditunjukkan pada Gambar 8.



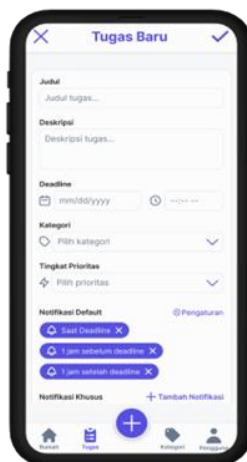
Gambar 8. Rancangan antarmuka beranda

Gambar 8 merupakan halaman beranda (*Home*) yang berfungsi sebagai titik masuk utama pengguna setelah membuka aplikasi dan melakukan *login*. Pada halaman ini ditampilkan ringkasan sapaan dan tugas harian pengguna agar mereka dapat langsung memahami informasi prioritas tanpa beban kognitif yang berlebihan. Dari halaman beranda ini, pengguna dapat menavigasi ke menu daftar tugas secara keseluruhan. Tampilan antarmuka daftar tugas dapat dilihat pada Gambar 9.



Gambar 9. Rancangan antarmuka daftar tugas

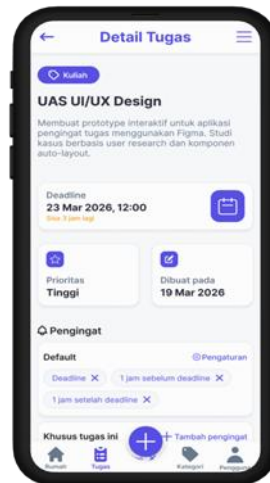
Gambar 9 menampilkan seluruh tugas terdaftar dalam bentuk kartu (*task card*). Halaman ini terintegrasi dengan indikator visual berupa warna batas waktu untuk menandai tingkat prioritas. Dari halaman ini, pengguna dapat menambahkan kegiatan baru dengan mengakses *form* input tugas. Tampilan antarmuka untuk *form* pengisian tersebut dapat dilihat pada Gambar 10.



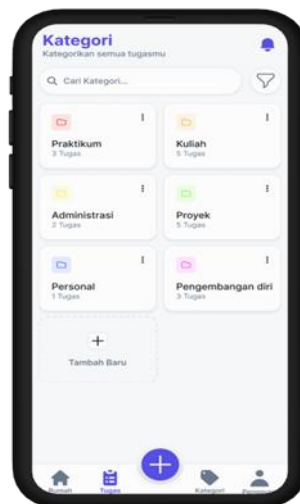
Gambar 10. Rancangan antarmuka *input task*

Gambar 10 menampilkan formulir interaktif untuk memasukkan data tugas baru. Pengguna dapat mengisi judul tugas, tenggat waktu, pengaturan prioritas, serta mengonfigurasi alarm peringatan (*push notification*). Setelah tugas baru berhasil ditambahkan dan disimpan, pengguna dapat melihat detail komprehensif dari tugas tersebut.

Gambar 11 rincian informasi dari tugas yang telah dibuat. Halaman ini dilengkapi dengan elemen penghitung waktu mundur yang dinamis dan memberikan opsi bagi pengguna untuk mengubah status penyelesaian tugas. Selain melihat detail, pengguna juga dapat melihat tugas berdasarkan kelompoknya pada halaman kategori.

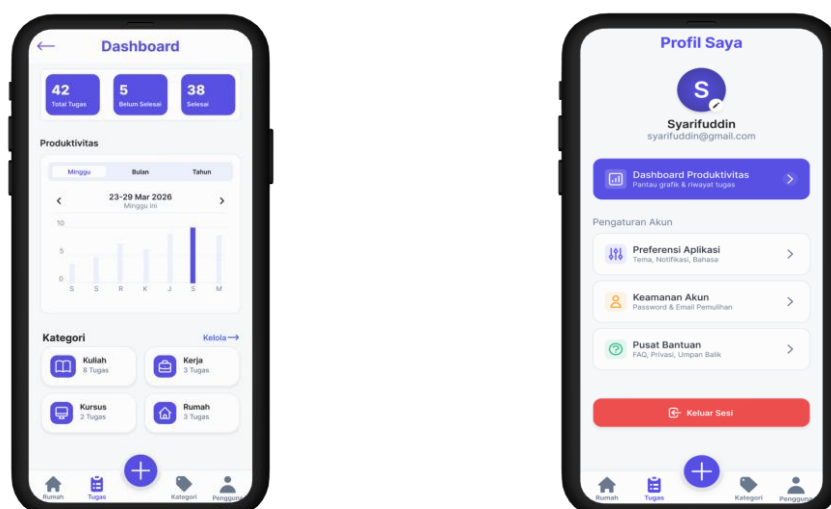


Gambar 11. Rancangan antarmuka detail tugas



Gambar 12. Rancangan antarmuka kategori

Gambar 12 memvisualisasikan pengelompokan tugas berdasarkan kategori spesifik. Fitur ini dirancang guna memudahkan pengguna dalam memfilter pekerjaan mereka secara terstruktur.

Gambar 13. Rancangan antarmuka profil dan statistik (*Dashboard*)

Gambar 13 menyajikan antarmuka metrik visual terkait jumlah tugas yang telah selesai, tugas yang terlewat, maupun tugas yang masih dalam proses pengerjaan. Selanjutnya, untuk mendokumentasikan setiap pemberitahuan tugas yang masuk, aplikasi menyediakan halaman riwayat notifikasi yang berfungsi sebagai log rekaman kronologis sebagaimana ditunjukkan pada Gambar 14.



Gambar 14. Antarmuka halaman riwayat notifikasi



Gambar 15. Visualisasi *push notification*

Terakhir sebagai luaran (*output*) utama dari sistem pengingat ini, dirancang visualisasi *push notification* yang terintegrasi dengan sistem operasi Android. Visualisasi ini mencakup berbagai kondisi perangkat mulai dari tampilan *lock-screen*, jendela *pop-up*, indikator bilah pada status, hingga *mode fullscreen* untuk peringatan prioritas tinggi seperti yang diilustrasikan pada Gambar 15.

## Kesimpulan

Berdasarkan hasil analisis, perancangan antarmuka (UI/UX), dan pemodelan sistem yang telah diuraikan dapat ditarik kesimpulan bahwa perancangan aplikasi pengingat tugas dan *deadline* berbasis Android dengan fitur *push notification* telah berhasil menjawab permasalahan manajemen waktu mahasiswa. Pendekatan metode *prototyping* terbukti sangat efektif dalam memfasilitasi pembuatan purwarupa tingkat tinggi (*high fidelity*) yang interaktif melalui aplikasi Figma. Purwarupa ini telah memvisualisasikan seluruh alur pengguna (*user flow*) dengan sangat baik, mulai dari halaman *onboarding*, autentikasi, manajemen tugas dengan indikator warna prioritas, metrik statistik, hingga antarmuka mode alarm.

Selain itu, pemodelan arsitektur logika sistem menggunakan *Unified Modelling Language*(UML) telah berhasil merumuskan mekanisme fitur kerja *push notification* secara komprehensif. Sistem dirancang mampu memantau tenggat waktu tugas di latar belakang (*background service*) dan memicu peringatan otomatis melalui komponen *notification manager*. Hasil evaluasi kelayakan fungsionalitas rancangan menggunakan *Black-Box Testing* menunjukkan tingkat keberhasilan sebesar 100%. Hal ini mengonfirmasi bahwa rancangan antarmuka dan alur logika aplikasi telah valid, interaktif, dan secara struktural sepenuhnya siap untuk dijadikan cetak biru (*blueprint*) bagi programmer untuk memasuki tahapan implementasi pengkodean.

#### Daftar Pustaka

- [1] Nelianli Yan Jaya, M. Agustian Reyza Novris, and Junadhi, "Penerapan Metode Design Sprint Dalam Perancangan UI/UX Aplikasi Peningkat Sarapan," *SATIN - Sains dan Teknologi Informasi*, vol. 8, no. 2, pp. 152–161, Dec. 2022, doi: 10.33372/stn.v8i2.892.
- [2] M. Stach, L. Mulansky, M. Reichert, R. Pryss, and F. Beierle, "Call to Action: Investigating Interaction Delay in Smartphone Notifications," *Sensors*, vol. 24, no. 8, p. 2612, Apr. 2024, doi: 10.3390/s24082612.
- [3] R. A. Sukanto, E. Piantari, and L. Van Kevin, "Android-based mobile academic information using push notification for supporting the student's academic activities: A case study," 2023, p. 060012. doi: 10.1063/5.0155356.
- [4] B. J. Surentu, "Perancangan Prototype UX/UI Aplikasi TaskMate Sebagai Peningkat Tugas Harian Mahasiswa Menggunakan Metode Design Thinking".
- [5] J. C. Isdhianto, V. J. L. Jaya, and Y. Wahyuningsih, "Perancangan UI/UX Menerapkan Metode Design Thinking Pada Aplikasi Peningkat Harian Berbasis Figma," *SINTEK-DC (Sistem Informasi dan Teknologi Informasi-Darma Cendika)*, vol. 1, no. 2, 2025.
- [6] M. J. Firdaus and R. M. Manikam, "Perancangan Aplikasi To-do-list 'MyList,'" *Format : Jurnal Ilmiah Teknik Informatika*, vol. 14, no. 1, p. 36, Jan. 2025, doi: 10.22441/format.2025.v14.i1.004.
- [7] L. Elyza, "Design and Build an Academic E-Reminder Application Using Mobile-Based Push Notification Technology," *Jurnal Komputer, Informasi dan Teknologi*, vol. 3, no. 1, Jun. 2023, doi: 10.53697/jkomitek.v3i1.1166.
- [8] Weni Oktavia Agus, Finanta Okmayura, Shofia Tri Wahyuni, Greghard Shawenner, Melati Angguni, and Diny Syahputri, "Implementasi Metode Design Thinking Dalam Perancangan UI/UX Pada Aplikasi 'Resep Kita,'" *JURNAL FASILKOM*, vol. 14, no. 1, pp. 156–165, Apr. 2024, doi: 10.37859/jf.v14i1.6637.
- [9] W. Brain, C. Windyaningsih, and H. Istiqbal, "Analisis Implementasi Metode Prototyping Pada Sistem Informasi Pemeliharaan Alat Pendingin di Rumah Sakit Khusus Daerah Duren Sawit," *Jurnal Manajemen dan Administrasi Rumah Sakit Indonesia (MARS)*, vol. 7, no. 1, pp. 78–98, Feb. 2023, doi: 10.52643/marsi.v7i1.2932.
- [10] A. T. Pratama and N. Santoso, "Rancang Bangun Sistem Informasi Manajemen Event Berbasis Mobile," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 6, pp. 2399–2407, 2021.
- [11] S. R. Joshua, W. Abbas, J.-H. Lee, and S. K. Kim, "Trust Components: An Analysis in The Development of Type 2 Diabetic Mellitus Mobile Application," *Applied Sciences*, vol. 13, no. 3, p. 1251, Jan. 2023, doi: 10.3390/app13031251.
- [12] U. Nuraini and K. Haryono, "Rancang Bangun Sistem Informasi Untuk Meningkatkan Efektivitas dan Efisiensi Manajemen Pengelolaan Kegiatan Ramadhan," *AUTOMATA*, vol. 3, no. 2, 2022.
- [13] A. Fikriyya, "Prototyping Dalam Perancangan Sistem Informasi Sekolah Desa Pendar Foundation Yogyakarta," 2021.
- [14] M. A. Senubekti, G. L. Dajoreyta, and N. Anggraini, "Pembuatan Desain UI/UX dengan Metode Prototyping pada Aplikasi Layanan Pengadilan Negeri Bale Bandung menggunakan Figma," *Jurnal Informatika Terpadu*, vol. 10, no. 1, pp. 1–10, Mar. 2024, doi: 10.54914/jit.v10i1.1001.
- [15] P. Kustanto, R. Bram Khalil, and A. Noe'man, "Penerapan Metode Prototype dalam Perancangan Media Pembelajaran Interaktif," *Journal of Students' Research in Computer Science*, vol. 5, no. 1, pp. 83–94, May 2024, doi: 10.31599/6x0dfz47.
- [16] Reynard Adelard, Richo Muthicahya, Evan Setiawan Wicaksono, Laurentius Kenneth V, Siska Narulita, and Sekarlangit, "Implementasi Metode Pengembangan Sistem Prototype pada Rancang Bangun Aplikasi Marketplace Lensa Buana," *Bridge : Jurnal Publikasi Sistem Informasi dan Telekomunikasi*, vol. 3, no. 2, pp. 69–81, May 2025, doi: 10.62951/bridge.v3i2.424.

- [17] B. A. Alfahri and N. B. Nisa, "Implementasi SDLC Prototyping dalam Perancangan Website Profil Sekolah MIS Nahdhatul Islam," *Jurnal Teknologi Sistem Informasi*, vol. 6, no. 1, pp. 1–10, Apr. 2025, doi: 10.35957/jtsi.v6i1.11146.
- [18] M. Hafidz Rivai, I. Ma'ruf Nugroho, and Y. Raymond Ramadhan, "Rancang Bangun Aplikasi Agenda Penjadwalan Berbasis Mobile Dengan Fitur Push Notification dan Reminder Menggunakan Metode Extreme Programming," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 5, pp. 8793–9799, Sep. 2024, doi: 10.36040/jati.v8i5.10838.
- [19] M. Rezki, L. N. Hayati, and I. Irawati, "Perancangan Sistem Informasi Penjualan ATK Study Kasus UD. MRESKI Berbasis Android," *LINIER: Literatur Informatika dan Komputer*, vol. 1, no. 4, pp. 302–316, Dec. 2024, doi: 10.33096/linier.v1i4.2530.
- [20] I. Ramadhan, I. Irawati, and S. Sugiarti, "Rancang Bangun Sistem Informasi Penyewaan Kamar Kost Berbasis Android pada Kost Alfira di Makassar," *LINIER: Literatur Informatika dan Komputer*, vol. 1, no. 3, pp. 223–234, Dec. 2024, doi: 10.33096/linier.v1i3.2499.
- [21] A. F. Sainlia, P. L. LB, and H. Azis, "Perancangan Aplikasi Penjadwalan Dakwah Mubaligh menggunakan Metode PIECES berbasis Android," *LINIER: Literatur Informatika dan Komputer*, vol. 1, no. 1, pp. 99–109, Mar. 2024, doi: 10.33096/linier.v1i1.2487.
- [22] A. P. Lolo Lasinrang, H. Harlinda, and S. Sugiarti, "Perancangan Aplikasi Penyewaan Baju Bodo Berbasis Android," *LINIER: Literatur Informatika dan Komputer*, vol. 2, no. 2, pp. 203–213, Sep. 2025, doi: 10.33096/linier.v2i2.3111.
- [23] S. Daruyani, O. D. Purbiyanti, I. Irawaty, and A. W. Putra, "Perancangan UI/UX Aplikasi Pengingat Shalat (Ar-Rayyan) Dengan Metode Design Thinking Menggunakan Figma," *Journal of Information System, Informatics and Computing*, vol. 8, no. 2, p. 264, Dec. 2024, doi: 10.52362/jisicom.v8i2.1669.
- [24] S. Tazkiyah and A. Arifin, "Perancangan UI/UX pada Website Laboratorium Energy menggunakan Aplikasi Figma," *Jurnal Teknologi Terpadu*, vol. 8, no. 2, pp. 72–78, Dec. 2022, doi: 10.54914/jtt.v8i2.513.
- [25] M. T. Sugianto, H. Herman, and S. R. Jabir, "Perancangan UI/UX Aplikasi Top Up Game ChillnFun Berbasis Mobile Menggunakan Metode Design Thinking," *LINIER: Literatur Informatika dan Komputer*, vol. 2, no. 3, pp. 447–458, Oct. 2025, doi: 10.33096/linier.v2i3.3154.