

Pengembangan Aplikasi *Productivity Tracker* Menggunakan Java Swing

Ferry Hasan¹, Agatha Diani Putri Saka², Eka Bagus Priambudi³, Nathanael Gordon Tanu⁴, Danika Najwa Ardelia⁵,
Fawwaz Ali Akbar^{6*}

^{1,2,3,4,5,6} Informatika, Universitas Pembangunan Nasional “Veteran” Jawa Timur

¹22081010085@student.upnjatim.ac.id

²22081010094@student.upnjatim.ac.id

³22081010081@student.upnjatim.ac.id

⁴22081010101@student.upnjatim.ac.id

⁵22081010103@student.upnjatim.ac.id

*Corresponding author email: fawwaz.ali.fik@upnjatim.ac.id

Abstrak—Produktivitas adalah aspek penting dalam kehidupan sehari-hari, terutama dalam konteks profesional. Untuk membantu memantau dan meningkatkan produktivitas, digunakan sebuah aplikasi *productivity tracker*. Aplikasi ini memberikan pengguna gambaran tentang catatan aktivitas dan melacak waktu yang dihabiskan untuk setiap tugas. Fitur utama yang diperlukan adalah perekaman aktivitas dan grafik visual yang memperlihatkan produktivitas pengguna dari waktu ke waktu. Dalam pengembangan aplikasi tersebut, metode SDLC (Systems Development Life Cycle) digunakan. Metode ini memungkinkan pengembangan aplikasi secara sistematis melalui tahapan analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Penerapan metode SDLC, khususnya model waterfall, dipilih untuk mengatasi keterbatasan waktu yang terkait dengan pengembangan perangkat lunak. Aplikasi *productivity tracker* menjadi alat yang sangat berguna dalam meningkatkan efisiensi dan produktivitas. Fitur perekaman waktu kerja dan grafik visual memudahkan pengguna untuk melacak dan menganalisis tingkat efisiensi waktu yang telah digunakan dalam pekerjaan. Dengan adanya aplikasi ini, pengguna dapat mengevaluasi efisiensi mereka dan membuat perubahan yang diperlukan untuk mencapai tujuan dengan lebih efisien. Metode SDLC menjadi kerangka kerja yang membantu dalam pengembangan aplikasi ini, memastikan penghasilan aplikasi yang sesuai dengan kebutuhan dan tujuan penelitian.

Kata Kunci— *Productivity Tracker*, Evaluasi Efisiensi, Produktivitas, Grafik Visual, Penggunaan Waktu, SDLC .

I. PENDAHULUAN

Komponen utama dari kehidupan akademis dan profesional adalah produktivitas, terutama di era digital ketika manajemen waktu dan tugas yang efisien sangat penting. Produktivitas kerja membutuhkan pola pikir yang berfokus pada praktik dan hasil kerja yang semakin baik. Pengawasan kerja merupakan salah satu variabel yang mempengaruhi produktivitas di tempat kerja, menurut penelitian sebelumnya [1]. Pengawasan yang baik dapat membantu orang mengembangkan rasa tanggung jawab, memberi mereka lebih banyak kontrol atas tindakan dan hasil kerja mereka. Pengawasan yang teratur dan terstruktur mendorong pengembangan kebiasaan disiplin yang meningkatkan hasil kerja dan membuat tugas sehari-hari menjadi lebih terorganisir dan produktif.

Pada era digital ini, teknologi memainkan peran vital dalam mendukung dan meningkatkan produktivitas, mengubah proses manual menjadi otomatisasi yang lebih cepat dan praktis [2]. Teknologi yang terus berkembang mempermudah individu untuk melakukan kegiatan sehari-hari, sehingga meningkatkan produktivitas secara keseluruhan. Solusi digital seperti aplikasi *productivity tracker* telah menjadi populer untuk membantu individu dan organisasi melacak dan mengelola produktivitas mereka. Kami memilih untuk merancang dan membangun aplikasi desktop *productivity tracker* menggunakan Java Swing sebagai upaya menjawab kebutuhan tersebut.

Productivity tracker adalah alat atau aplikasi yang dirancang untuk membantu individu atau tim dalam mengelola waktu, tugas, dan proyek dengan lebih efisien. Dengan menggunakan alat ini, pengguna dapat melacak pekerjaan yang telah diselesaikan, waktu yang dihabiskan untuk tugas tertentu, dan proyek-proyek yang sedang berlangsung. Alat ini juga membantu dalam mengidentifikasi pemborosan waktu dan memberikan wawasan yang berguna untuk meningkatkan produktivitas [3]. Aplikasi ini dilengkapi dengan fitur-fitur yang berguna untuk memantau dan meningkatkan produktivitas, seperti perekam waktu kerja, visualisasi kinerja, pengaturan jam kerja, indikator kemajuan, analisis produktivitas, pencatatan progres, dan laporan kemajuan.

Swing adalah antarmuka pemrograman aplikasi GUI di Java yang dibangun di atas AWT. Swing menggunakan model event dan layout manager dari AWT, namun menambahkan komponen baru dan pengganti untuk komponen AWT. Swing termasuk dalam Java Foundation Classes (JFC) yang menyediakan dukungan grafis 2D dan kemampuan seret dan lepas [4].

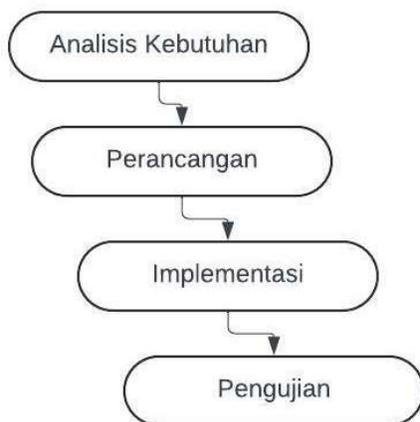
Productivity Tracker adalah aplikasi yang memudahkan pengguna untuk mencatat waktu kerja dan memantau kemajuan terhadap target jam kerja yang ditetapkan. Dengan visualisasi seperti diagram, pengguna dapat melihat perbandingan antara waktu produktif dan tidak produktif, membantu mereka untuk meningkatkan efisiensi. Selain itu,

fitur seperti pengaturan jam kerja, indikator kemajuan, dan analisis produktivitas, mendukung pengguna dalam pencatatan progres dan menghasilkan laporan kemajuan yang akurat.

Aplikasi ini dirancang untuk membantu individu dalam mengelola waktu dengan lebih baik, meningkatkan produktivitas, dan mencapai tujuan kerja dengan lebih efektif. Dengan *Productivity Tracker*, pengguna diharapkan dapat mengoptimalkan penggunaan waktu mereka, mendapatkan dukungan yang transparan dalam mengelola produktivitas, dan pada akhirnya, mencapai kesuksesan dalam pekerjaan mereka dengan bantuan teknologi yang inovatif.

II. METODOLOGI PENELITIAN

Dalam pengembangan aplikasi *Productivity Tracker* menggunakan Java Swing, mengadopsi metode *Waterfall SDLC (Software Development Life Cycle)* yang terstruktur dan terurut. Metodologi SDLC (Systems Development Life Cycle) merupakan pendekatan klasik yang digunakan dalam pengembangan, pemeliharaan, dan penggunaan sistem informasi. Dalam penelitian ini, dipilih menggunakan model waterfall SDLC, yang juga dikenal sebagai model air terjun, karena terdapat batasan waktu yang mempengaruhi pengembangan perangkat lunak[5]. Berikut adalah tahapan-tahapan yang dilalui dalam pengembangan aplikasi ini.



Gbr. 1 Tampilan Flowchart SDLC Waterfall

A. Analisis Kebutuhan

Aplikasi *Productivity Tracker*, yang dibangun dengan menggunakan Java Swing, menawarkan antarmuka yang user-friendly, kinerja yang handal, dan kemampuan untuk digunakan dengan mudah di berbagai platform. Aplikasi ini dilengkapi dengan fitur-fitur seperti perekam waktu kerja, visualisasi kinerja, dan pengaturan jam kerja, indikator kemajuan, analisis produktivitas, pencatatan progress serta laporan kemajuan yang telah diuji melalui blackbox testing secara menyeluruh dan terbukti beroperasi dengan responsif dan optimal. Hasilnya menunjukkan bahwa aplikasi ini dapat menjadi solusi yang efektif untuk meningkatkan produktivitas di lingkungan kerja modern saat ini.

B. Perancangan

Setelah menganalisis kebutuhan, merancang struktur aplikasi, antarmuka pengguna, serta skema penyimpanan data yang diperlukan. Memastikan bahwa fitur-fitur yang dirancang dapat saling terintegrasi dengan baik.

C. Implementasi

Tahap ini melibatkan pengkodean aplikasi menggunakan Java Swing. Mengimplementasikan fitur-fitur yang telah dirancang, termasuk perekam waktu kerja, visualisasi kinerja, pengaturan jam kerja, indikator kemajuan, analisis produktivitas, pencatatan progres, dan laporan kemajuan. Memastikan bahwa kode yang ditulis mematuhi standar dan praktik terbaik dalam pengembangan perangkat lunak.

D. Pengujian

Setelah implementasi, melakukan pengujian menyeluruh untuk memastikan kualitas dan keandalan aplikasi. Menguji setiap fitur secara terpisah dan melakukan pengujian integrasi untuk memastikan bahwa semua fitur berfungsi dengan baik dan saling berinteraksi sesuai dengan yang diharapkan.

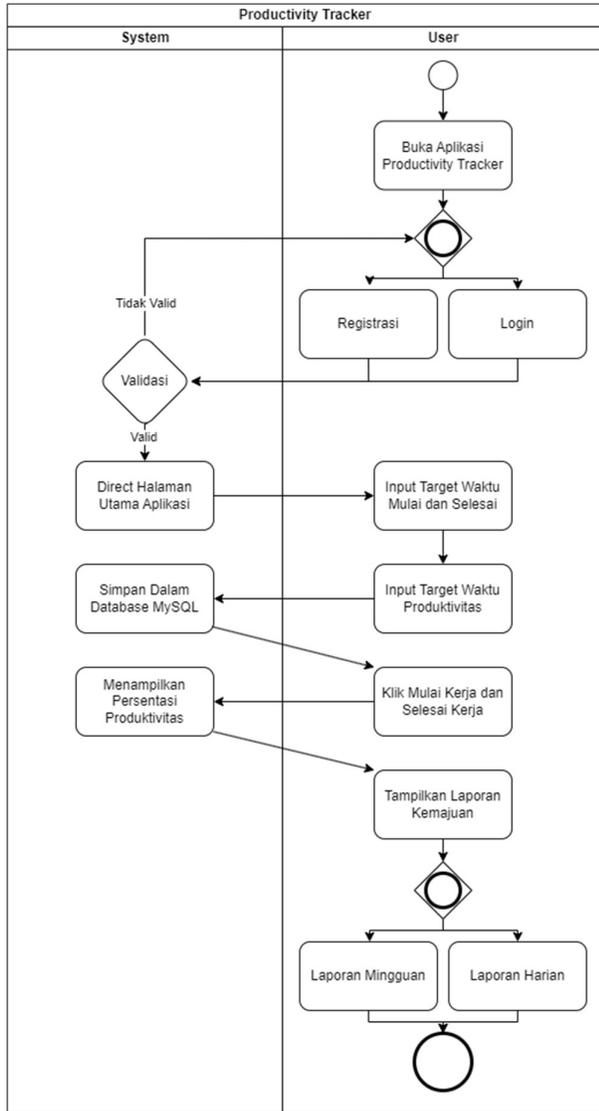
Melalui pendekatan *Waterfall SDLC* ini, berhasil mengembangkan aplikasi *Productivity Tracker* dengan fitur-fitur yang diuraikan di atas. Metodologi ini memungkinkan untuk mengikuti langkah-langkah yang terstruktur dalam pengembangan aplikasi dan memastikan bahwa setiap fitur diimplementasikan dengan baik.

III. HASIL DAN PEMBAHASAN

Aplikasi *Productivity Tracker* yang digunakan dalam penelitian ini dibuat untuk mengakomodasi sejumlah kemampuan, termasuk kemampuan untuk mencatat jam kerja, memvisualisasikan kinerja, menyesuaikan jam kerja, melacak kemajuan, menganalisis produktivitas, dan menghasilkan laporan kemajuan. Proses pengembangan aplikasi ini dirancang untuk meningkatkan efisiensi dan produktivitas. Berikut ini merupakan penjelasan terkait diagram yang digunakan dalam proses pengembangan aplikasi.

A. Business Process Model and Notation (BPMN)

Business Process Model and Notation atau bisa disingkat BPMN bagaikan peta jalan yang memvisualisasikan alur proses bisnis secara komprehensif. BPMN merupakan salah satu bahasa pemodelan grafis yang digunakan dalam pemodelan proses bisnis yang terjadi pada sebuah organisasi secara detail dengan aliran informasi berupa pesan yang disampaikan antar pihak terkait [6]. Dibuat dan dikelola oleh *Object Management Group (OMG)*, BPMN hadir sebagai standar notasi terstruktur untuk memodelkan proses bisnis, layaknya diagram alir yang diperkaya dengan simbol-simbol dan notasi yang terstandarisasi. BPMN tak hanya bermanfaat untuk memvisualisasikan proses, tetapi juga menjadi alat analisis yang ampuh. Dengan menganalisis diagram BPMN, inefisiensi dan inkonsistensi dalam proses dapat diidentifikasi, membuka peluang untuk perbaikan dan optimalisasi.



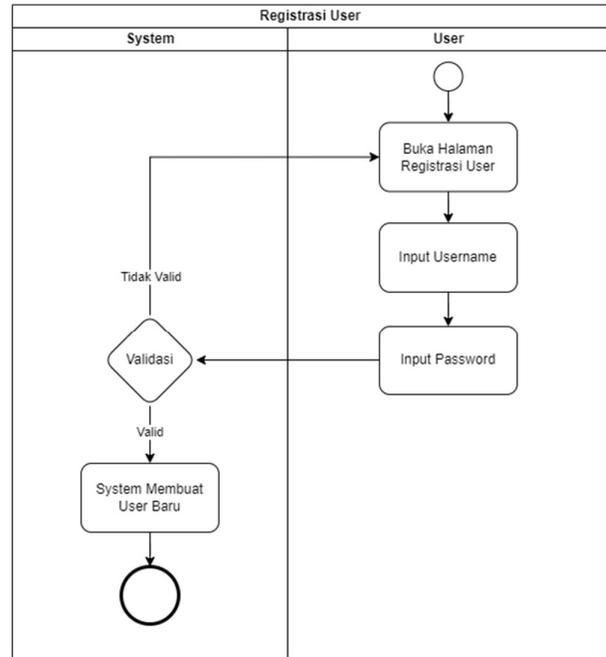
Gbr. 2 BPMN Aplikasi

Diagram BPMN pada Gbr. 2 dapat membantu kita memahami sistem pelacak produktivitas. Masuk ke aplikasi *Productivity Tracker* dan menetapkan target waktu untuk setiap tugas, kemudian mencatat waktu mulai dan selesai bekerja, yang selanjutnya disimpan di database. Sistem memvalidasi data dan menghasilkan laporan produktivitas yang komprehensif, lengkap dengan visualisasi berupa grafik dan bagan. Dengan memvisualisasikan langkah-langkah dalam proses, diagram BPMN memudahkan identifikasi inefisiensi dan area untuk perbaikan. Ini dapat membantu pengguna dalam meningkatkan produktivitas secara keseluruhan.

B. Activity Diagram

Activity Diagram adalah diagram yang digunakan untuk memodelkan aliran proses atau alur kerja dalam sebuah sistem [7]. Diagram ini menunjukkan bagaimana berbagai aktivitas

dalam sistem terhubung satu sama lain dan bagaimana mereka berinteraksi untuk mencapai tujuan yang diinginkan.

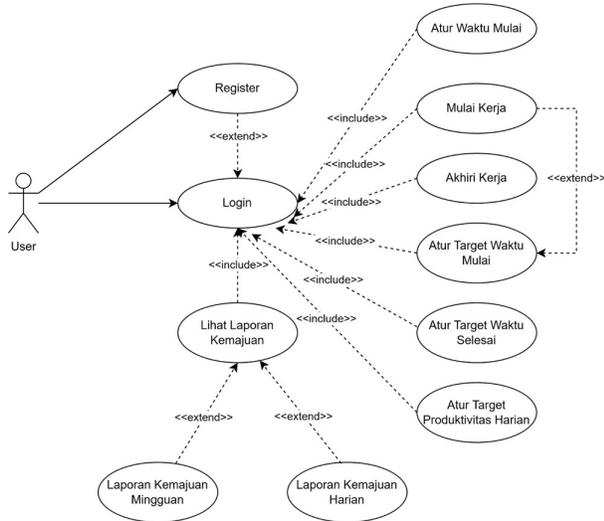


Gbr. 3 Activity Diagram

Activity Diagram register dari aplikasi *productivity tracker* pada Gbr. 5 tersebut menunjukkan proses registrasi pengguna baru. Proses ini dimulai dengan pengguna membuka halaman registrasi user. Kemudian, pengguna memasukkan username dan password. Sistem kemudian memvalidasi username dan password. Jika username dan password valid, sistem membuat user baru. Jika username atau password tidak valid, pengguna diminta untuk memasukkannya kembali.

C. Use Case Diagram

Use Case Diagram adalah jenis diagram yang menggambarkan bagaimana aktor dan sistem berinteraksi.[8] Aktor, *use case*, dan hubungannya adalah beberapa elemen yang membentuk diagram *use case*. *Use case* menunjukkan bagaimana sistem berperilaku dan memungkinkan aktor untuk berinteraksi dengannya; aktor adalah entitas yang berinteraksi dengan sistem. Hubungan antara *use case* dan aktor menunjukkan bagaimana aktor bekerja dengan sistem untuk mencapai tujuan tertentu [9]. Diagram *use case* yang disajikan di bawah ini mengilustrasikan interaksi dalam aplikasi pelacak produktivitas kami untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa yang berhak menggunakan fungsi tersebut [10].

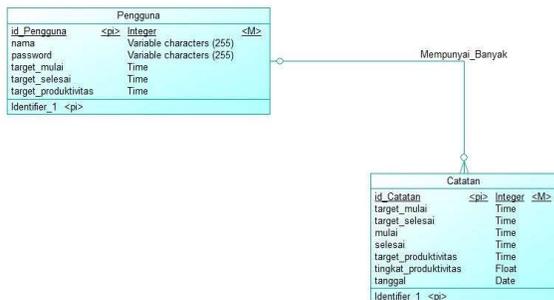


Gbr. 4 Use Case Diagram

Pada Gbr. 4 diilustrasikan interaksi pengguna dengan sistem *Productivity Tracker*. “User” bertindak sebagai aktor yang memiliki kemampuan untuk mendaftar dan *login* ke dalam sistem. Setelah *login*, pengguna dapat melakukan berbagai aktivitas seperti mulai kerja, akhiri kerja, atur waktu mulai, atur target waktu mulai, atur target waktu selesai, dan atur target produktivitas harian. Selain itu, ada fungsi untuk melihat laporan kemajuan, yang dapat diperluas menjadi laporan kemajuan mingguan dan laporan kemajuan harian. Hubungan antara *use case* menunjukkan bahwa beberapa aktivitas adalah ekstensi atau inklusi dari aktivitas lain, menandakan bahwa ada kondisi tertentu atau opsi tambahan yang tersedia bagi pengguna setelah melakukan aktivitas dasar seperti *login*. Diagram ini secara efektif mengilustrasikan bagaimana pengguna berinteraksi dengan sistem dan bagaimana berbagai tugas terkait satu sama lain dalam konteks manajemen waktu kerja.

D. CDM dan PDM

Basis data diperlukan dalam aplikasi *productivity tracker* untuk menyimpan, mengelola, dan menganalisis data pengguna, target waktu, dan catatan sesi kerja. Hal ini memungkinkan penyimpanan yang terstruktur, analisis data, manajemen pengguna, dan pengembangan fitur-fitur tambahan.

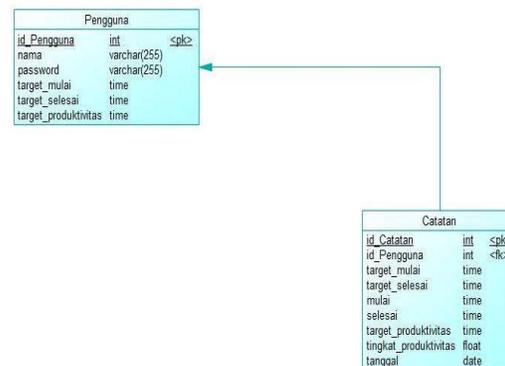


Gbr. 5 Conceptual Data Model (CDM)

Gbr. 5 menunjukkan diagram skema Conceptual Data Model (CDM) untuk sistem pelacakan produktivitas. Conceptual Data Model (CDM) adalah representasi logis dari struktur basis data yang menunjukkan hubungan antara data menggunakan simbol-simbol, di mana atribut dari suatu entitas memiliki relasi dengan atribut dari entitas lainnya [11]. CDM pada Gbr. 5 menghubungkan dua tabel, yaitu "Pengguna" dan "Catatan". Tabel "Pengguna" menyimpan informasi mengenai setiap pengguna dengan menggunakan kolom seperti *id_Pengguna* (kunci utama yang merupakan identifikasi unik untuk setiap pengguna), *nama* (*nama pengguna*), *password* (*kata sandi pengguna*), *target_mulai* (*waktu mulai target*), *target_selesai* (*waktu selesai target*), *target Produktivitas* (*target produktivitas pengguna*), dan *identifier_1* (*pengenal untuk manajemen database internal*). Tabel "Catatan" menyimpan detail sesi kerja dengan kolom-kolom seperti *id_Catatan* (*kunci utama yang merupakan identifikasi unik untuk setiap catatan*), *target_mulai* (*waktu mulai target sesi*), *target_selesai* (*waktu selesai target sesi*), *mulai* (*waktu mulai aktual sesi*), *selesai* (*waktu selesai aktual sesi*), *target Produktivitas* (*target produktivitas sesi*), *tingkat Produktivitas* (*produktivitas aktual sesi*), *tanggal* (*tanggal sesi*), dan *identifier_1* (*pengenal untuk manajemen database internal*). Hubungan antara kedua tabel ini adalah *one-to-many*, di mana satu pengguna bisa memiliki banyak catatan, tetapi setiap catatan hanya terkait dengan satu pengguna. Skema ini dirancang untuk aplikasi pelacakan produktivitas, yang memungkinkan aplikasi untuk melacak kemajuan pengguna, menganalisis pola produktivitas, dan memberikan wawasan untuk meningkatkan kebiasaan kerja pengguna.

CDM ini mewakili model data untuk sistem pelacakan produktivitas yang memungkinkan pengguna untuk:

- Mencatat target produktivitas: Pengguna dapat menetapkan target waktu mulai, waktu selesai, dan tingkat produktivitas untuk setiap sesi kerja.
- Melacak sesi kerja: Data mulai, selesai, dan produktivitas sebenarnya dari sesi kerja dicatat dalam tabel Catatan.
- Menganalisis produktivitas: Dengan data yang terkumpul, sistem dapat menganalisis produktivitas pengguna, membandingkannya dengan target, dan memberikan insight untuk meningkatkan kebiasaan kerja.

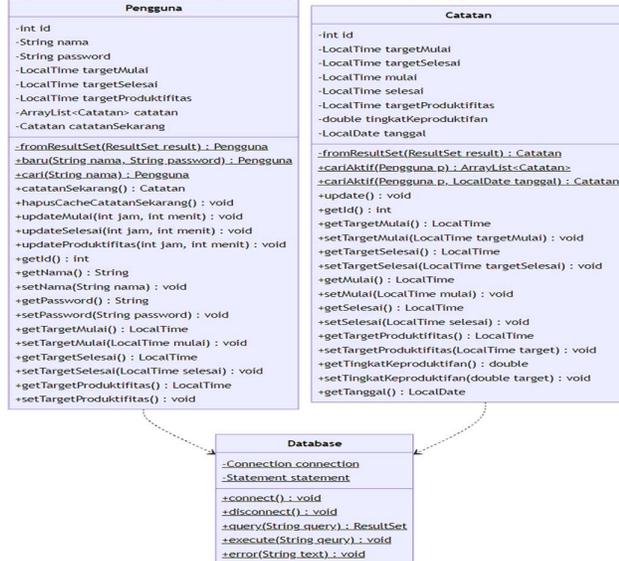


Gbr. 6 Physical Data Model (PDM)

Physical Data Model (PDM) yang tergambar pada Gbr. 6 menunjukkan struktur fisik dari database yang digunakan dalam sistem pelacakan produktivitas. Physical Data Model (PDM) adalah perancangan basis data secara fisik yang memiliki tipe data yang lebih khusus dan spesifik. PDM merepresentasikan bentuk fisik atau aktual dari basis data. Physical Data Model (PDM) menggunakan berbagai tabel untuk menggambarkan data serta hubungan antar data tersebut [12]. PDM ini terdiri dari dua tabel utama, yaitu Pengguna dan Catatan, yang memiliki hubungan one-to-many antara keduanya. Tabel Pengguna menyimpan informasi tentang pengguna, seperti ID pengguna, nama, kata sandi, target waktu mulai, target waktu selesai, dan target tingkat produktivitas. Sementara itu, tabel Catatan berfungsi sebagai tempat penyimpanan catatan sesi kerja, dengan kolom-kolom yang mencakup ID catatan, ID pengguna terkait, waktu mulai yang ditargetkan, waktu selesai yang ditargetkan, waktu mulai aktual, waktu selesai aktual, tingkat produktivitas yang dicapai, tanggal sesi kerja, dan kolom tambahan untuk identifikasi. Hubungan antara kedua tabel ini memungkinkan penggunaan yang efisien dan terorganisir dari data untuk pelacakan dan analisis produktivitas.

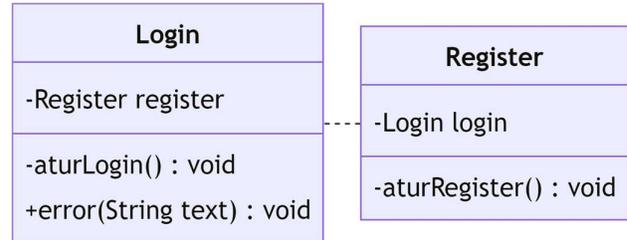
E. Class Diagram

Dalam pengembangan aplikasi Productivity Tracker, class diagram digunakan untuk memetakan struktur dan hubungan antar kelas yang berperan dalam sistem. Class Diagram adalah representasi visual dari sistem yang berisi program atau menu form pilihan. Diagram ini menggambarkan elemen-elemen yang saling berdekatan dalam bentuk tabel kotak-kotak kecil yang berisi kata kunci dan isinya yang berhubungan. Dalam merancang diagram ini, diperlukan ketelitian dan pemikiran agar dapat diimplementasikan dalam program sistem yang akan dibuat [13]. Class diagram ini memberikan gambaran visual mengenai komponen-komponen utama yang ada dalam aplikasi serta bagaimana mereka saling berinteraksi untuk mencapai tujuan memantau tingkat keproduktifan pengguna.



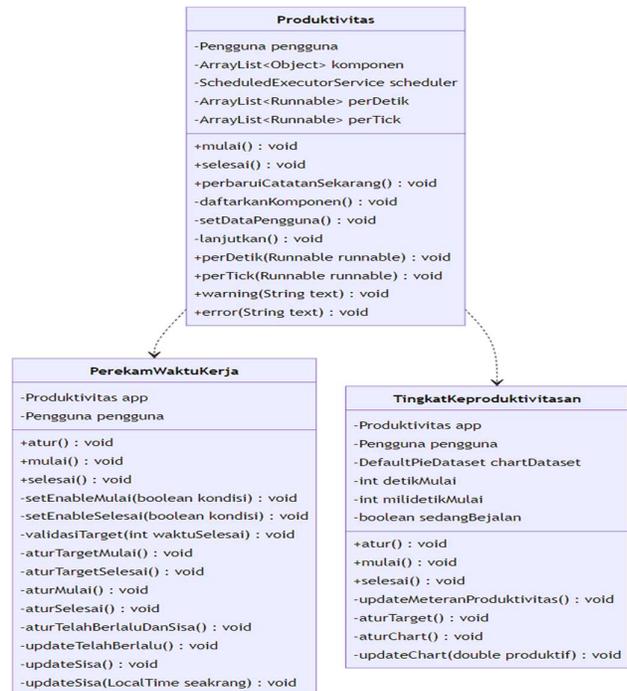
Gbr. 7 Class Diagram Database, Pengguna dan Catatan

Class “Database” berfungsi untuk mengelola koneksi dan berkomunikasi dengan basis data yang digunakan. Class “Pengguna” menyimpan informasi terkait data personal dan preferensi pengguna serta menyediakan metode untuk mengelola data-data pengguna. Selanjutnya, class “Catatan” berfungsi untuk menyimpan informasi catatan kerja pengguna dan menyediakan metode untuk mengelolanya. Kelas “Pengguna” dan “Catatan” menggunakan kelas “Database” agar dapat berinteraksi dengan basis data yang digunakan.



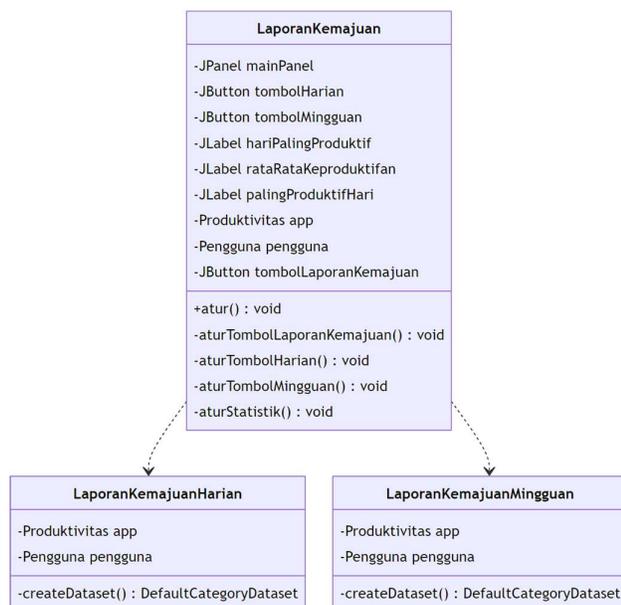
Gbr. 8 Class Diagram Login dan Register

Class diagram selanjutnya adalah “Login”, yang mengelola proses autentikasi pengguna. Kelas ini mencakup komponen GUI seperti panel, kolom input, tombol, dan label, serta menyediakan metode untuk menangani login dan menampilkan pesan kesalahan. Kelas ini memiliki hubungan langsung dengan kelas “Register” untuk proses pendaftaran dan dengan kelas Produktivitas setelah login berhasil. Selanjutnya adalah kelas “Register” yang digunakan untuk mengelola proses pendaftaran pengguna baru. Kelas ini juga mencakup komponen GUI yang serupa dengan “Login” dan menyediakan metode untuk menangani proses pendaftaran serta mengarahkan kembali ke login jika diperlukan.



Gbr. 9 Class Diagram Produktivitas, PerekamWaktuKerja dan TingkatKeproduktivitasan

Class diagram selanjutnya adalah “Produktivitas”, yang merupakan kelas utama dalam aplikasi. Kelas ini mengelola berbagai komponen GUI untuk mencatat waktu kerja, menampilkan persentase produktivitas, dan mengakses laporan kemajuan. Kelas Produktivitas berinteraksi dengan beberapa kelas lain seperti “PerekamWaktuKerja”, “TingkatKeproduktivitasan”, dan “LaporanKemajuan” untuk menyediakan fungsionalitas lengkap bagi pengguna. Class “PerekamWaktuKerja” digunakan untuk mencatat dan mengelola waktu kerja pengguna. Kelas ini berinteraksi langsung dengan kelas “Produktivitas” dan “Pengguna” untuk mendapatkan data yang diperlukan serta memperbarui catatan waktu kerja. Class “TingkatKeproduktivitasan” digunakan untuk menghitung dan menampilkan tingkat produktivitas pengguna berdasarkan data yang dicatat. Kelas ini juga berinteraksi dengan “Produktivitas” dan Pengguna untuk memperoleh dan memperbarui data yang relevan.



Gbr. 10 Class LaporanKemajuan, LaporanKemajuanHarian dan LaporanKemajuanMingguan

Class “LaporanKemajuan” berfungsi untuk mengelola dan menampilkan laporan kemajuan pengguna. Class ini berhubungan dengan kelas “Produktivitas” dan Pengguna untuk mendapatkan data yang diperlukan. Selanjutnya terdapat class “LaporanKemajuanHarian” dan class “LaporanKemajuanMingguan”. Kedua kelas ini berguna untuk memisahkan logika untuk membuat dataset laporan harian dan mingguan yang digunakan dalam menampilkan laporan kemajuan pada kelas “LaporanKemajuan”.

Berikut merupakan rancangan tampilan antarmuka dari aplikasi *Productivity Tracker*.

A. Tampilan Login



Gbr. 11 Tampilan Login

Tampilan login adalah antarmuka awal yang meminta pengguna untuk memasukkan nama pengguna dan kata sandi untuk mengakses halaman utama. Formulir login terdiri dari bidang nama pengguna dan password, diikuti dengan tombol masuk untuk memverifikasi informasi. Jika pengguna belum memiliki akun maka bisa menekan tombol “daftar disini”. Pesan kesalahan muncul jika kredensial tidak valid. Tampilan ini dirancang responsif agar mudah digunakan di berbagai perangkat.

B. Tampilan Register



Gbr. 12 Tampilan Register

Tampilan registrasi adalah antarmuka yang memungkinkan pengguna untuk membuat akun baru dengan memasukkan nama pengguna, kata sandi, dan konfirmasi kata sandi. Formulir registrasi mencakup bidang untuk nama pengguna, kata sandi, dan bidang konfirmasi password untuk memastikan kesesuaian. Setelah mengisi semua informasi, pengguna menekan tombol daftar untuk menyelesaikan pendaftaran. Desain yang intuitif dan responsif memastikan proses ini mudah diikuti di berbagai perangkat.

C. Tampilan Dashboard



Gbr. 13 Tampilan Dashboard

Tampilan dashboard menyediakan berbagai fitur yang mendukung manajemen waktu dan produktivitas. Perakam Waktu Kerja berfungsi untuk mencatat waktu mulai dan berakhirnya sesi kerja, membantu pengguna melacak durasi aktivitas mereka. Visualisasi Kinerja menampilkan diagram yang menunjukkan jumlah waktu produktif dan waktu yang terbuang, memberikan gambaran visual tentang efisiensi kerja. Pengaturan Jam Kerja memungkinkan pengguna mengatur target jumlah jam kerja harian, sehingga mereka bisa merencanakan hari kerja mereka dengan lebih baik. Indikator Kemajuan menunjukkan seberapa dekat pengguna dengan mencapai target jam kerja yang telah ditetapkan. Analisis Produktivitas menampilkan total waktu yang telah dihabiskan dan sisa waktu yang diperlukan untuk mencapai tujuan, membantu pengguna mengukur pencapaian mereka. Pencatatan Progres memperbarui dan mencatat sesi kerja yang telah berlangsung beserta statistik dan grafiknya, memberikan detail rinci tentang kinerja. Terakhir, Laporan Kemajuan menampilkan laporan sesi kerja berdasarkan data yang telah disimpan atau diperbarui, menyediakan ringkasan lengkap untuk evaluasi.

D. Tampilan Laporan Kemajuan



Gbr. 14 Laporan Kemajuan

Tampilan Laporan Kemajuan menyediakan laporan sesi kerja berdasarkan data yang disimpan atau diperbarui, memberikan ringkasan lengkap untuk evaluasi. Pengguna dapat memilih untuk melihat laporan produktivitas secara mingguan atau harian. Tampilan utama halaman laporan kemajuan menunjukkan keterangan hari paling produktif, membantu pengguna mengidentifikasi kapan mereka bekerja paling efisien, persentase rata-rata tingkat keproduktifan, serta rata-rata hari paling produktif dalam seminggu, memberikan wawasan tentang pola produktivitas harian. Fitur ini memungkinkan analisis mendalam dan perencanaan strategis untuk meningkatkan kinerja.



Gbr. 15 Grafik Tingkat Produktivitas Harian

Dalam laporan kemajuan harian, sebuah grafik bertajuk "Keproduktifan Harian" memvisualisasikan pola produktivitas

harian selama kurang lebih satu bulan. Pada diagram tersebut, skala "Persen" merefleksikan persentase produktivitas yang telah dilakukan, dan skala "Jam" mengukur total waktu yang dialokasikan untuk tugas-tugas produktif. Diagram ini merupakan alat analisis yang efektif untuk mengukur dan memahami produktivitas harian.



Gbr. 16 Grafik Tingkat Produktivitas Mingguan

Laporan kemajuan mingguan tersebut menghadirkan sebuah grafik yang mengilustrasikan tingkat produktivitas tiap rentang waktu satu minggu. Grafik ini, meskipun serupa dalam desain dengan grafik produktivitas harian, berbeda dalam hal skala waktu yang digunakan; di mana ia mengukur dan membandingkan produktivitas secara mingguan, bukan harian. Ini memungkinkan pengamatan tren dan pola produktivitas yang mungkin tidak terlihat dalam analisis harian.

Untuk memastikan bahwa aplikasi Productivity Tracker berfungsi dengan baik dan memenuhi kebutuhan pengguna,

dilakukan serangkaian uji coba sistem yang dikenal sebagai Black Box Testing. Pengujian perangkat lunak ini bertujuan untuk menguji fungsionalitas aplikasi tanpa memerlukan pemahaman khusus tentang pemrograman atau struktur internalnya [14]. Metode pengujian black box dilakukan dengan merancang skenario yang menguji semua fungsi dari aplikasi yang telah dibuat. Skenario ini dibuat untuk pengujian blackbox dengan membandingkan hasil yang diharapkan dan hasil pengujian [15]. Berikut merupakan Black Box Testing dari aplikasi Productivity Tracker:

TABEL 1.
BLACK BOX TESTING

Test Case	Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Menekan tombol "Masuk" pada halaman Masuk	Klik tombol "Masuk"	Apabila pengguna memasukkan nama dan password yang terdaftar maka akan masuk ke menu utama	Berhasil	<i>Valid</i>
		Apabila pengguna memasukkan nama dan password yang tidak terdaftar maka akan tampil pesan kesalahan	Berhasil	<i>Valid</i>
Menekan tombol "Daftar" pada halaman Daftar	Klik tombol "Daftar"	Apabila pengguna memasukkan nama dan password yang terdaftar maka akan tampil pesan kesalahan	Berhasil	<i>Valid</i>
		Apabila pengguna memasukkan nama dan password yang tidak terdaftar maka akan dengan nama dan password tersebut akan tersimpan	Berhasil	<i>Valid</i>
Mengatur input menit	Memasukkan menit kurang dari 0	<i>Input</i> jam akan berkurang 1	Berhasil	<i>Valid</i>
	Memasukkan menit lebih dari 59	<i>Input</i> jam akan bertambah 1	Berhasil	<i>Valid</i>
Mengatur input jam	Memasukkan jam kurang dari 0	Jam akan teratur menjadi 23	Berhasil	<i>Valid</i>
	Memasukkan jam lebih dari 23	Jam akan teratur menjadi 0	Berhasil	<i>Valid</i>
Mengatur Target Mulai	Memasukkan waktu pada <i>input</i> "Target Mulai"	Preferensi pengguna tentang Target Mulai tersimpan pada basis data	Berhasil	<i>Valid</i>
Mengatur Target Selesai	Memasukkan waktu pada <i>input</i> "Target Selesai"	Apabila waktu yang dimasukkan lebih besar daripada waktu Target Mulai, maka preferensi pengguna tentang Target Selesai tersimpan pada basis data	Berhasil	<i>Valid</i>
		Apabila waktu yang dimasukkan lebih kecil daripada waktu Target Mulai, maka akan tampil pesan kesalahan dan Target Mulai akan teratur menjadi Target Mulai ditambah 1 jam	Berhasil	<i>Valid</i>
Mengatur Target Produktifitas Hari Ini	Memasukkan waktu pada <i>input</i> "Target Produktifitas Hari Ini"	Preferensi pengguna tentang Target Produktifitas Hari Ini tersimpan pada basis data	Berhasil	<i>Valid</i>
Menekan tombol "Mulai Kerja"	Klik tombol "Mulai Kerja"	Sesi kerja baru pengguna tersimpan pada basis data	Berhasil	<i>Valid</i>
		Label "Telah Berlalu" mulai menghitung waktu yang sudah berlalu sejak sesi dimulai	Berhasil	<i>Valid</i>
		Label "Sisa" mulai menghitung waktu yang tersisa sampai "Target Selesai"	Berhasil	<i>Valid</i>
		Tombol "Mulai Kerja" menjadi tidak bisa diklik	Berhasil	<i>Valid</i>

		Tombol “Selesai Kerja” menjadi bisa diklik	Berhasil	<i>Valid</i>
		Label “Produktif” dan “Tidak Produktif” menampilkan hasil perhitungan tingkat keproduktifan dan ketidak produktifan	Berhasil	<i>Valid</i>
		Muncul <i>pie chart</i> yang menunjukkan perbandingan antara tingkat keproduktifan dan tingkat ketidak produktifan	Berhasil	<i>Valid</i>
Menekan tombol “Mulai Kerja”	Klik tombol “Mulai Kerja”	Sesi kerja saat ini tersimpan pada basis data	Berhasil	<i>Valid</i>
		Label “Telah Berlalu” berhenti menghitung	Berhasil	<i>Valid</i>
		Label “Sisa” berhenti menghitung	Berhasil	<i>Valid</i>
		Tombol “Mulai Kerja” menjadi bisa diklik	Berhasil	<i>Valid</i>
		Tombol “Selesai Kerja” menjadi tidak bisa diklik	Berhasil	<i>Valid</i>
		Label “Produktif” dan “Tidak Produktif” berhenti menghitung tingkat keproduktifan dan ketidak produktifan	Berhasil	<i>Valid</i>
		<i>Pie chart</i> berhenti menghitung perbandingan antara tingkat keproduktifan dan tingkat ketidak produktifan	Berhasil	<i>Valid</i>
Menekan tombol “Laporan Kemajuan”	Klik tombol “Laporan Kemajuan”	Muncul <i>window</i> lain yang berisi tentang statistik sesi kerja	Berhasil	<i>Valid</i>
Menekan tombol “Harian” pada halaman “Laporan Kemajuan”	Klik tombol “Harian”	Muncul <i>line chart</i> yang merangkum rata-rata tingkat keproduktifan harian dan rata-rata jam kerja per hari	Berhasil	<i>Valid</i>
Menekan tombol “Mingguan” pada halaman “Laporan Kemajuan”	Klik tombol “Mingguan”	Muncul <i>line chart</i> yang merangkum rata-rata tingkat keproduktifan mingguan dan rata-rata jam kerja per minggu	Berhasil	<i>Valid</i>

Tabel 1 ini mencatat serangkaian pengujian yang dilakukan untuk memastikan fungsionalitas aplikasi berjalan sesuai harapan. Setiap baris dalam tabel merepresentasikan satu Test Case, dengan rincian sebagai berikut:

- Test Case: Deskripsi singkat tentang tujuan dari Test Case ini.
- Pengujian: Langkah-langkah yang dilakukan untuk menjalankan Test Case ini.
- Hasil yang Diharapkan: Output atau perilaku aplikasi yang diharapkan setelah langkah pengujian dijalankan.
- Hasil Pengujian: Hasil aktual yang didapatkan setelah menjalankan Test Case.
- Kesimpulan: Status Test Case, apakah "Valid" (berhasil) atau "Invalid" (gagal).

Berdasarkan informasi dalam tabel 1, dapat disimpulkan bahwa aplikasi ini memiliki beberapa fitur penting. Fitur-fitur tersebut meliputi fungsi Mulai dan Selesai Kerja yang memungkinkan

pengguna mencatat waktu mulai dan selesai dari sesi kerja mereka. Selain itu, aplikasi ini juga menyediakan Pelacakan Produktivitas yang menghitung tingkat produktivitas pengguna berdasarkan target yang telah ditetapkan. Untuk visualisasi data, aplikasi ini menampilkan produktivitas dalam bentuk grafik Pie Chart dan Line Chart. Laporan Kemajuan juga tersedia untuk menampilkan ringkasan data produktivitas dalam bentuk laporan harian dan mingguan. Aplikasi ini dilengkapi dengan halaman login dan daftar, yang memerlukan username dan password untuk memastikan keamanan akses pengguna. Selain itu, aplikasi ini juga memiliki validasi input untuk memastikan data yang dimasukkan sesuai dengan aturan yang ditetapkan. Validasi ini mencakup Durasi Kerja, yang membatasi input menit dan jam kerja dalam rentang yang logis (misalnya, menit 0-59, jam 0-23), serta Target Waktu, yang memastikan bahwa Target Selesai lebih besar dari Target Mulai.

Dokumentasi Test Case ini menunjukkan bahwa pengujian Black Box telah dilakukan untuk memverifikasi fungsionalitas dan validasi input dari aplikasi. Semua Test Case yang tercatat dalam tabel ini memiliki status "Valid", yang berarti aplikasi berjalan sesuai harapan.

IV. KESIMPULAN

Pengembangan aplikasi *Productivity Tracker* menggunakan pendekatan *Waterfall* SDLC telah terbukti berhasil dalam meningkatkan efisiensi dan produktivitas pengguna. Dengan fitur-fitur yang mencakup pencatatan target produktivitas, analisis data, dan laporan kemajuan, aplikasi ini memberikan dukungan yang kuat bagi pengguna dalam mengelola waktu dan mencapai tujuan kerja mereka. Teknologi digital yang diimplementasikan melalui aplikasi ini tidak hanya memberikan solusi otomatisasi yang praktis, tetapi juga membantu dalam memperbaiki kebiasaan kerja dan meningkatkan kinerja individu serta organisasi secara keseluruhan. Dengan demikian, aplikasi *Productivity Tracker* menjadi alat yang berharga dalam upaya meningkatkan produktivitas dan efisiensi dalam lingkungan kerja modern.

REFERENSI

- [1] N. M. Hamdani, N. Rozalina, and K. M. Z. Basriwijaya, "Faktor-Faktor Yang Mempengaruhi Produktivitas Penyadap Tanaman Karet di PT. Atjeh Raya Corpindo Kebun Alur Buluh," *Jurnal Agrica/Jurnal Agrica*, vol. 16, no. 2, pp. 190–200, Oct. 2023, doi: 10.31289/agrica.v16i2.9658.
- [2] A. Cahyani, D. Agustin, I. Muslimah, and N. Khulliyatunnisa, "Peran Aplikasi Notion dalam Perkuliahan untuk Mewujudkan Produktivitas Mahasiswa," *Jurnal Pendidikan, Sains Dan Teknologi*, vol. 2, no. 1, pp. 262–273, Jun. 2023, doi: 10.47233/jpst.v2i2.754.
- [3] O. Y. Djogo, "Peran manajemen sumber daya manusia pada kinerja karyawan di masa pandemi covid-19," *JPPPI (Jurnal Penelitian Pendidikan Indonesia)*, vol. 8, no. 2, p. 218, Jul. 2022, doi: 10.29210/020221310.
- [4] D. Agushinta, A. Tarigan, E. W. Moyo, F. H. Siburian, and S. Widiyanto, "The Use of Java Swing's Components to Develop a Widget," *IJHCI (International Journal of Human Computer Interaction)*, vol. 1, no. 4, p. 95, 2011.
- [5] A. Y. Permana and P. Romadlon, "PERANCANGAN SISTEM INFORMASI PENJUALAN PERUMAHAN MENGGUNAKAN METODE SDLC PADA PT. MANDIRI LAND PROSPEROUS BERBASIS MOBILE," Dec. 29, 2019, <http://www.jurnal.pelitabangsa.ac.id/index.php/sigma/article/view/585>
- [6] Ismanto Ismanto, Firman Hidayah, and Kristinanti Charisma, "Pemodelan Proses Bisnis Menggunakan Business Process Modelling Notation (BPMN) (Studi Kasus Unit Penelitian Dan Pengabdian Kepada Masyarakat (P2KM) Akademi Komunitas Negeri Putra Sang Fajar Blitar)," *Briliant: Jurnal Riset dan Konseptual*, vol. 5, no. 1, pp. 69–76, Feb. 2020, doi: 10.28926/briliant.
- [7] S. Setiaji and R. Sastra, "Implementasi Diagram UML (Unified Modelling Language) pada perancangan sistem informasi penggajian," *Jurnal Teknik Komputer*, vol. 7, no. 1, pp. 106–111, Feb. 2021, doi: 10.31294/jtk.v7i1.9773.
- [8] Al Faruq, U. Rancang Bangun Aplikasi Rekam Medis Poliklinik Universitas Trilogi. *Jurnal Informatika*, Vol. 9, No. 1.2015
- [9] T. A. Kurniawan, "Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik," *Jurnal Teknologi Informasi Dan Ilmu Komputer/Jurnal Teknologi Informasi Dan Ilmu Komputer*, vol. 5, no. 1, pp. 77–86, Mar. 2018, doi: 10.25126/jtiik.201851610.
- [10] Muhammad Fami Siahhaan, Muhammad Hendri, Sri Lindawati, "Perancangan Sistem Informasi Promosi Dan Penjualan Pada PT. Gaya Makmur Mulia Menggunakan Metode Prototype," *J. Armada Inform.*, vol. 6, no. 2, pp. 631–643, 2022.
- [11] Putra AR. Rancang Bangun Sistem Informasi E-Portofolio Dosen (Studi Kasus: Jurusan Teknik Informatika Universitas Negeri Surabaya). *J Manaj Inform [Internet]*. 2018;8(2):170–7. Available from: <https://ejournal.unesa.ac.id/index.php/jurnal-manajemen-informatika/article/view/25205>
- [12] Y. A. Binarso, E. A. Sarwoko, and N. Bahtiar, "PEMBANGUNAN SISTEM INFORMASI ALUMNI BERBASIS WEB PADA PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS DIPONEGORO," *Binarso | Journal of Informatics and Technology*, Oct. 05, 2012, <https://ejournal3.undip.ac.id/index.php/joint/article/view/434/434>
- [13] A. Anharudin and A. Nurdin, "RANCANG BANGUN SISTEM INFORMASI ADMINISTRASI PENDAFTARAN KURSUS (STUDI KASUS: GHIBRANT ENGLISH COURSE-PANDEGLANG)," Sep. 30, 2018, <https://ejournal.lppmunsera.org/index.php/PROSISKO/article/view/750>
- [14] B. Nugroho, M. S. Hasibuan, and M. H. Annabil, "Perancangan Aplikasi Absensi Pegawai Berbasis Web Dengan Blackbox Testing pada DISPORA Sumatera Utara," *Journal of Computer Science and Informatics Engineering (CoSIE)*, pp. 199–209, Dec. 2023, doi: 10.55537/cosie.v2i4.714.
- [15] P. B. A. A. Putra, V. H. Pranatawijaya, N. N. K. Sari. Implementasi Location Based Service Pada Aplikasi Mobile Penyajian Ruang Ujian. *Jurnal Sains dan Informatika* 6 (1), 26-30, 2020.