

### **III. METODE PENELITIAN**

#### **3.1 Waktu dan Tempat**

Penelitian ini dilaksanakan di klinik Pratama Dokter yanti yang berlokasi di JL. Sersan Darphin No.96, Eka Jaya, Kecamatan Palmerah serta tiga cabangnya lagi berlokasi di Lorong Patimura No 12, Kenali Besar, Kecamatan Kota baru, yang kedua berlokasi di JL. Dr. Tazar no 21a, Buluran kenali, Kecamatan Telanaipura, dan yang terakhir berlokasi di JL KH. Hasan Anang, Eka Jaya. Waktu yang digunakan peneliti untuk penelitian ini dilaksanakan selama 6 bulan. dimulai dari bulan September 2024 dan di targetkan selesai akhir bulan Februari 2025.

#### **3.2 Alat Penelitian**

Bagian ini merupakan penunjang dalam melakukan proses penelitian. Dengan menggunakan alat dan sumber data untuk pengisian konten sistem, maka penelitian akan dapat lebih mudah untuk dilakukan. Beberapa alat yang dibutuhkan dalam penelitian ini adalah:

a. Hardware

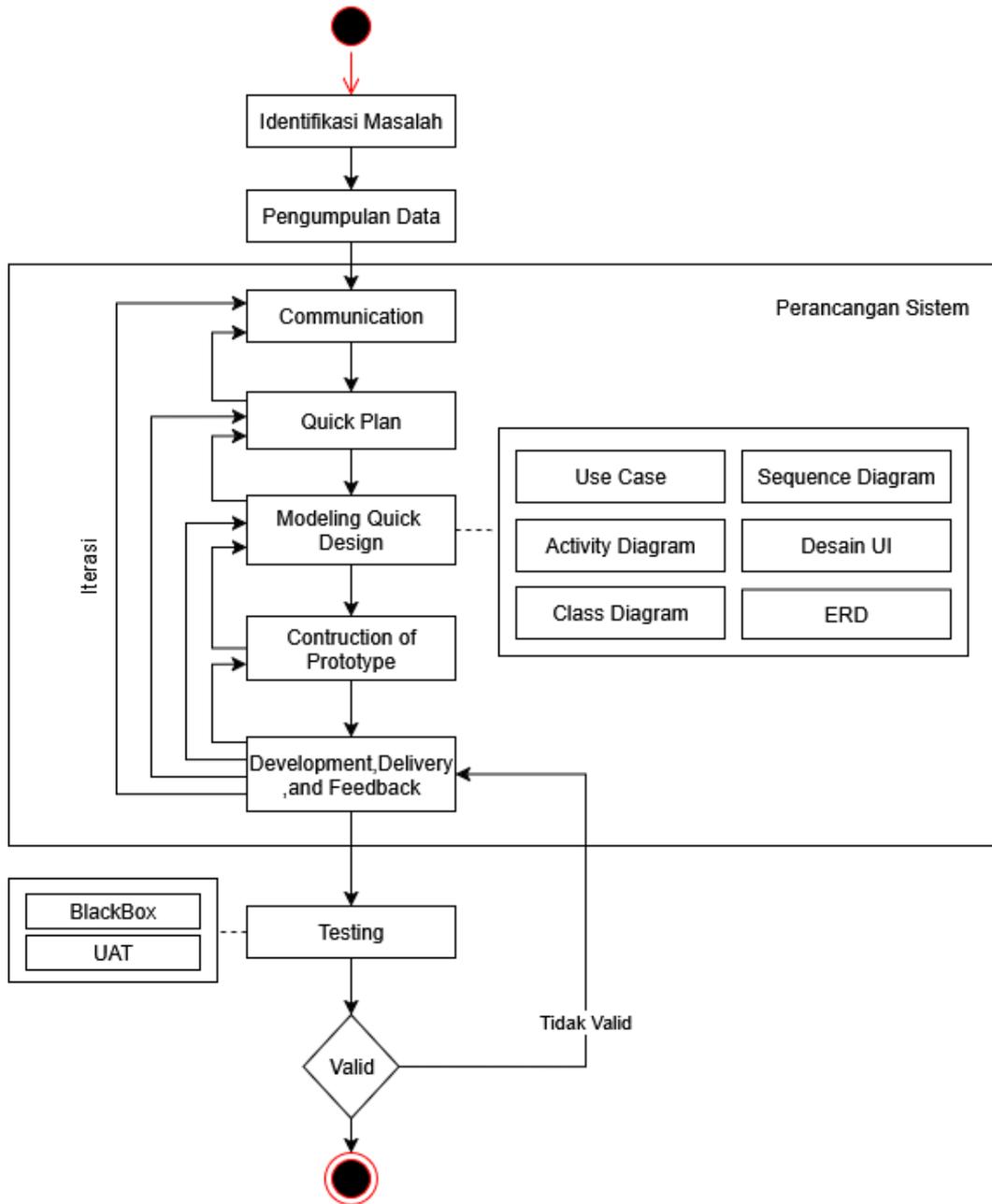
Laptop Lenovo ThinkPad X230 - Intel Core i5-3320M - 2,6GHz dengan VGA Intel HD Graphics 4000 dan RAM 10GB.

b. Software

- Chrome Browser
- Node Js
- Freamwork Next Js
- Freamwork Reach Js
- Freamwork Reach Native
- Visual Studio Code
- Os Windows 10 64bit
- PlantUml
- Expo
- Microsoft Word 2021 LTSC
- Firebase

#### **3.3 Kerangka Kerja Penelitian**

Penelitian ini memiliki kerangka kerja yang telah disusun, kali ini penelitian menggunakan kerangka model prototype, model ini dipilih karena model prototype lebih mendekati kriteria dari pengembangan system yang akan dilakukan. Berikut merupakan gambar dari kerangka kerja penelitian ini:



**Gambar 11.** Kerangka Kerja Penelitian

### 3.4 Tahapan Penelitian

Berdasarkan model prototype yang telah digambarkan diatas, maka dapat diuraikan pembahasan masing-masing tahapan penelitian dalam model tersebut adalah sebagai berikut

## Identifikasi Masalah

Identifikasi masalah ialah langkah awal dalam penelitian, sehingga dapat mendefinisikan masalah dalam penelitian dilakukan analisa terhadap proses yang sedang berjalan saat ini. Adapun masalah atau hambatan yang ditemukan antara lain:

1. Keterbatasan Mesin Fingerprint

Sistem fingerprint yang digunakan saat ini hanya bekerja dalam mode offline, sehingga tidak memungkinkan akses data secara langsung. Hal ini menyulitkan pengelolaan kehadiran pegawai secara cepat dan akurat. Ketidakmampuan sistem untuk terhubung ke internet juga meningkatkan risiko kesalahan pencatatan dan manipulasi data karena data tidak dapat diintegrasikan atau dipantau secara otomatis terlebih lagi mesin fingerprint hanya tersedia di cabang yang berada di Eka Jaya dan cabang lainnya masih menggunakan absensi secara manual menggunakan buku.

2. Kurangnya Efisiensi dan Transparansi

Pengelolaan kehadiran dengan sistem saat ini masih belum efisien. Rekap data membutuhkan waktu lama dan memiliki risiko kesalahan yang cukup tinggi. Selain itu, sistem ini kurang transparan karena tidak ada mekanisme untuk memverifikasi kehadiran secara langsung, sehingga menciptakan celah untuk potensi penyalahgunaan atau manipulasi data oleh pegawai.

3. Kesulitan Pengambilan Keputusan

Ketidaksesuaian sistem dalam menyediakan data secara real-time menjadi hambatan dalam pengambilan keputusan yang cepat dan akurat. Data kehadiran yang lambat diperoleh menyebabkan analisis kinerja pegawai menjadi tidak optimal, sehingga berdampak pada efektivitas pengelolaan sumber daya manusia dan operasional klinik secara keseluruhan.

Untuk mengatasi permasalahan tersebut, penting untuk memperkenalkan alternatif sistem presensi yang lebih efisien dan mudah diakses. Salah satu solusinya adalah dengan mengembangkan aplikasi web dan mobile, dimana aplikasi web berfungsi untuk pengelola klinik memajemen pegawainya secara real-time sedangkan aplikasi mobile berfungsi untuk pegawai melakukan presensi kehadiran masuk maupun pulang

## **Pengumpulan Data**

Jenis Penelitian yang digunakan dalam penelitian ini adalah kualitatif. Pendekatan kualitatif yaitu pendekatan penelitian dimana data-data yang dikumpulkan berupa kata-kata, gambar-gambar dan bukan angka. Data - data tersebut dapat diperoleh dari hasil wawancara, dokumentasi pribadi, catatan. Pada penelitian ini, digunakan sumber data berupa data primer. Data primer adalah data dalam bentuk verbal atau kata - kata yang diucapkan secara lisan, gerak-gerik atau perilaku yang dilakukan oleh subjek yang dapat dipercaya, dalam hal ini adalah subjek penelitian (informan) yang berkenaan dengan variabel yang diteliti.

Untuk melakukan pembahasan dan penyusunan kerja penelitian ini diperlukan data yang akurat, maka pada pengumpulan data penulis menggunakan beberapa metode dalam pengumpulan data antara lain: Penelitian Lapangan (field research). Peneliti mendapatkan data dengan terjun langsung kelapangan dengan cara:

- a. Peneliti melakukan kegiatan tanya jawab secara langsung terhadap pemilik Klinik Pratama Dokter Yanti dan pegawainya, untuk memperoleh data yang akurat serta relevan agar dapat menghasilkan perancangan sistem informasi yang sesuai dengan kebutuhan.
- b. Pengamatan (observation), peneliti melakukan pengamatan langsung terhadap objek yang akan diteliti yang bertujuan untuk memperkuat data, mengetahui serta mendapatkan informasi secara langsung. Mengumpulkan data dengan melakukan pengamatan terhadap aplikasi yang sudah ada.
- c. Penelitian Perpustakaan (Library Research) Kegiatan ini mencari data - data dari buku-buku di perpustakaan dan browsing dari internet yang berhubungan dengan apa yang diteliti untuk membantu mendapatkan data dalam melakukan penelitian.

## **Perancangan Sistem**

Pada tahap ini, diharapkan dapat dihasilkan analisis menyeluruh mengenai kendala dan permasalahan yang terjadi. Analisis ini bertujuan untuk memahami secara mendalam berbagai hambatan yang dihadapi, sehingga memungkinkan penulis untuk merumuskan solusi yang tepat. Dalam upaya menemukan solusi yang sesuai, penulis melakukan kajian rinci mengenai proses presensi yang diterapkan saat ini di klinik, termasuk bagaimana pemilik, manajemen, dan pegawai melaksanakan pencatatan kehadiran mereka.

Menurut (Saptia et al, 2021) perancangan sistem dapat juga berarti menyusun suatu sistem baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sebagian sistem yang telah ada. Pada tahap ini kita merancang usulan sistem yang baru, penulis menggunakan metode pengembangan sistem dengan menggunakan model *Prototype*. *Prototype* adalah sebuah metode pengembangan software yang banyak digunakan pengembang yang dapat saling berinteraksi dengan pelanggan selama proses pembuatan sistem dan terdiri dari 5 tahap yang saling terkait atau mempengaruhi yaitu sebagai berikut:

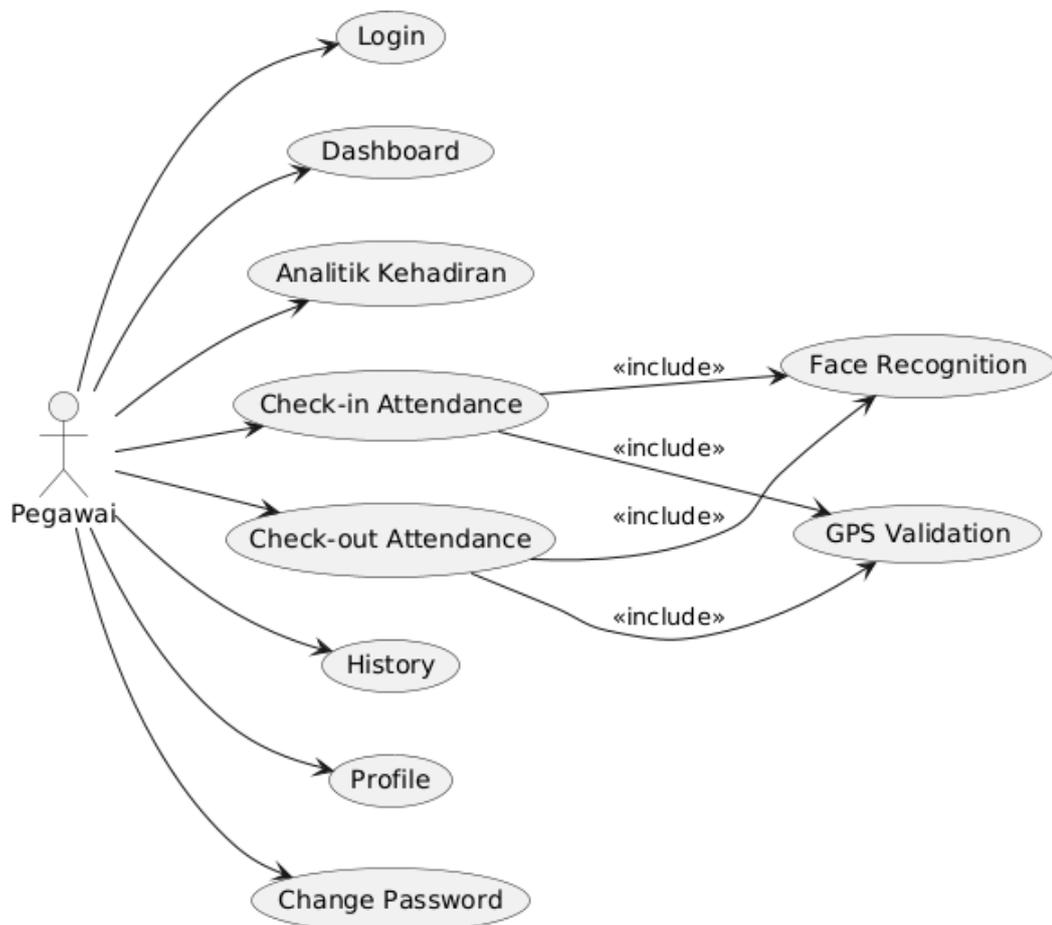
Berdasarkan model prototype yang telah digambarkan diatas, maka dapat diuraikan pembahasan masing-masing tahap dalam model tersebut adalah sebagai berikut:

1. *Communication* / Komunikasi Tim pengembang perangkat lunak melakukan pertemuan dengan pemilik klinik dan pegawai untuk menentukan kebutuhan perangkat lunak yang saat ini telah diketahui dan untuk menggambarkan area dimana definisi lebih jauh untuk iterasi selanjutnya.
2. *Quick Plan* / Perencanaan Secara Cepat Dalam perencanaan ini iterasi pembuatan prototipe dilakukan secara cepat. Setelah itu dilakukan pemodelan dalam bentuk “rancangan cepat”.
3. *Modeling Quick Design* / Model Rancangan Cepat. Pada tahap ini memodelkan perencanaan tadi dengan menggunakan beberapa model berorientasi objek dengan menggunakan UML yang terdiri dari *Use case*, *Activity Diagram*, *Sequence Diagram*, Perancangan ERD, dan perancangan Antarmuka.
4. *Construction of Prototype* / Pembuatan Prototype Dalam pembuatan rancangan cepat berdasarkan pada representasi aspek-aspek perangkat lunak yang akan terlihat oleh para end user (misalnya rancangan antar muka pengguna atau format tampilan). Rancangan cepat merupakan dasar untuk memulai konstruksi pembuatan prototype.
5. *Deployment Delivery and Feedback* / Penyerahan Dan Memberikan Umpan Balik Terhadap Pengembangan. Prototype kemudian diserahkan kepada pemilik klinik untuk mengevaluasi prototype yang telah dibuat sebelumnya dan memberikan umpan-balik yang akan digunakan untuk memperbaiki spesifikasi kebutuhan.
6. *Iterasi* terjadi saat pengembang melakukan perbaikan terhadap prototype tersebut.

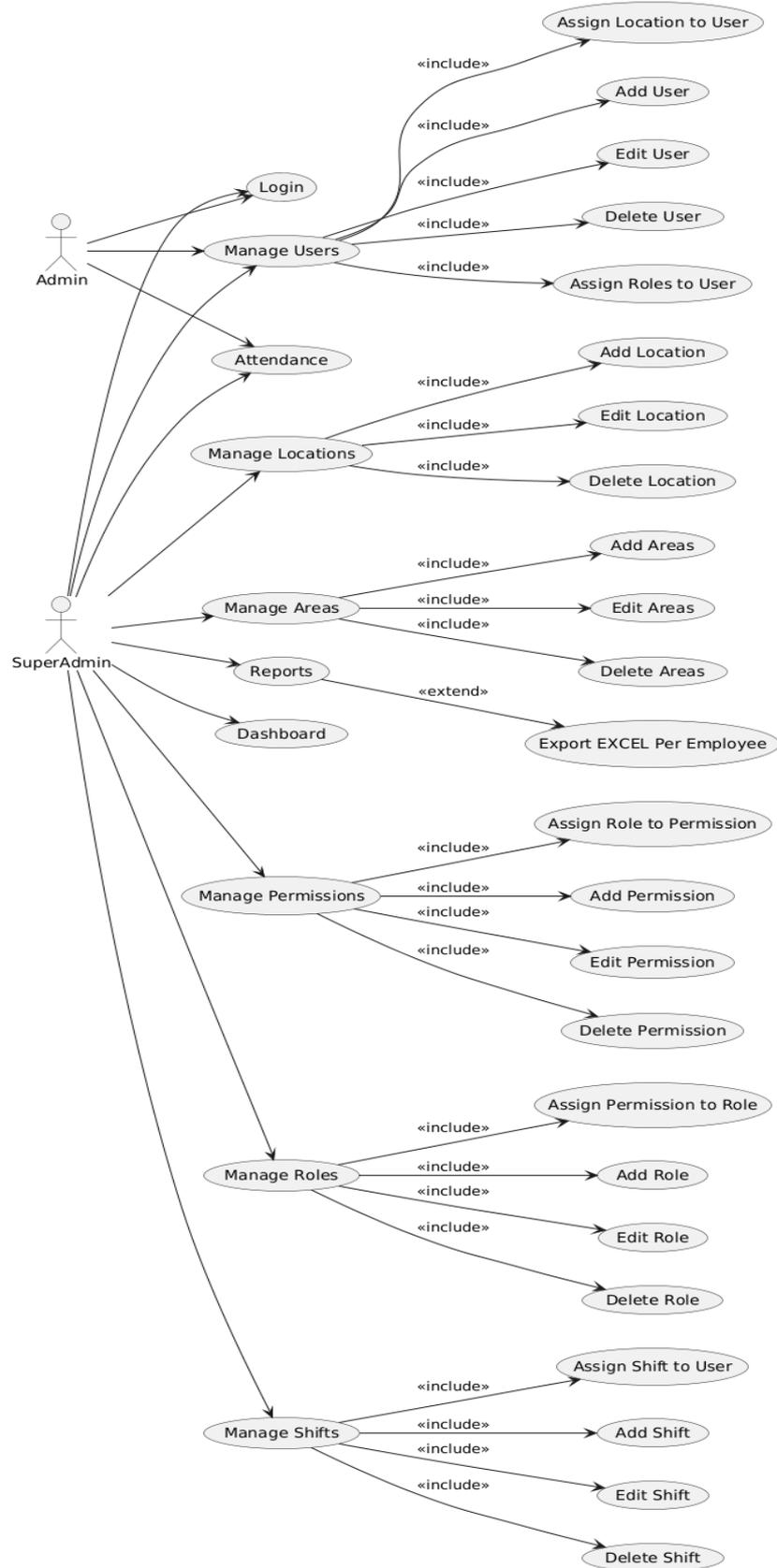
Adapun dari hasil perancangan sistem informasi dengan menggunakan metode pengembangan prototype yaitu *quick design* atau desain untuk sistem yang dibuat. Desain ini yang memberi gambaran tentang sistem kepada pengguna dimana alur desain sistem diusulkan adalah sebagai berikut

### 1. Use Case Diagram

Pada rancangan *use case diagram* sistem presensi Klinik Pratama Dokter Yanti, terdapat tiga aktor utama, yaitu Super Admin, Administrator, dan Pegawai. Super Admin memiliki wewenang penuh untuk mengelola data pengguna dan melihat laporan kehadiran secara menyeluruh. Administrator bertugas mengelola jadwal kerja, serta mengelola data pegawai yang berkaitan dengan kehadiran. Sementara itu, Pegawai yang terdiri dari tenaga medis dan non-medis seperti dokter, dan staf lainnya - dapat melakukan presensi masuk dan keluar sesuai dengan jadwal kerja yang telah ditentukan..



**Gambar 12.** Use Case Diagram Mobile Presensi



**Gambar 13.** Use Case Diagram Web Dashboard Presensi

### Narasi Use Case Diagram

Setelah menjabarkan fungsionalitas use case dari masing-masing aktor, maka selanjutnya adalah menjabarkan alur kerja masing-masing use case melalui skenario untuk menggambarkan proses yang terjadi antara sistem dengan user. Berikut adalah narasi use case yang akan menjelaskan alur kerja setiap use case diagram yang ada

a. Login (*mobile dan web*)

|               |  |
|---------------|--|
| Nama use case | : Login  |
| Aktor         | : Semua Aktor  |
| Deskripsi     | : Pengguna memasukkan username dan password untuk mengakses sistem |

**Table 8.** Narasi Usecase Login

| Kategori         | Proses Aktor   | Proses Sistem   |
|------------------|--|---|
| Basic Flow       | 1. Pengguna memasukkan username dan password.                              | 3. Sistem memvalidasi username dan password.                  |
|                  | 2. Pengguna menekan tombol Login.  | 4. Sistem memberikan akses ke dashboard utama jika valid.     |
| Alternative Flow | - Jika validasi gagal:   | - Sistem menampilkan pesan error dan meminta pengisian ulang. |
| Postcondition    | Pengguna berhasil masuk ke sistem atau mendapat notifikasi validasi gagal. | Sistem tidak mengubah data pengguna di database.              |

b. Change Password (*mobile*)

|               |  |
|---------------|--|
| Use Case Name | : Change Password                        |
| Actor         | : Pegawai                                |
| Deskripsi     | : Pengguna Login Pertama kali ke sistem. |

**Table 9.** Narasi Usecase Change Password

| Kategori   | Proses Aktor                 | Proses Sistem  |
|------------|------------------------------|--|
| Basic Flow | 1. Pengguna melakukan login. | 2. Sistem memeriksa apakah pengguna wajib mengubah kata sandi. |

| <b>Kategori</b>  | <b>Proses Aktor</b>   | <b>Proses Sistem</b>   |
|------------------|---|--|
|                  | 3. Pengguna memasukkan kata sandi lama dan kata sandi baru. | 4. Sistem memvalidasi kata sandi lama dan format kata sandi baru.  |
|                  | 5. Pengguna menekan tombol "Simpan".                        | 6. Sistem memperbarui kata sandi di database.<br>7. Sistem menampilkan pesan sukses jika perubahan berhasil. |
| Alternative Flow | - Jika kata sandi lama yang dimasukkan salah:               | - Sistem menampilkan pesan error bahwa kata sandi lama tidak sesuai.   |
| Postcondition    | Kata sandi pengguna berhasil diperbarui.                    | Sistem menyimpan kata sandi baru yang telah divalidasi di database.  |

c. Dashboard (*mobile*)

|                      |  |
|----------------------|--|
| Nama <i>use case</i> | : Dashboard  |
| Aktor                | : Pegawai  |
| Deskripsi            | : Halaman Dashboard menampilkan informasi penting seperti status absensi terakhir. |

**Table 10.** Narasi Usecase Dashboard Mobile

| <b>Kategori</b>  | <b>Proses Aktor</b>                        | <b>Proses Sistem</b>   |
|------------------|--|--|
| Basic Flow       | 1. Pengguna membuka menu Dashboard.        | 2. Sistem mengambil ringkasan data kehadiran dari database.              |
|                  | 3. Pengguna melihat ringkasan kehadiran.   | 4. Sistem menampilkan data ringkasan ke dashboard.                       |
| Alternative Flow | - Tidak tersedia (asumsi selalu berhasil). | - Bisa ditambahkan: sistem gagal ambil data → tampilkan pesan kesalahan. |
| Postcondition    | Pengguna melihat data kehadiran terupdate. | Sistem hanya melakukan query; tidak ada perubahan data di database.      |

## d. Riwayat Presensi

|                      |   |
|----------------------|---|
| Nama <i>use case</i> | : Riwayat Presensi  |
| Aktor                | : Pegawai   |
| Deskripsi            | : Pegawai dapat melihat riwayat kehadiran mereka secara detail, termasuk tanggal, waktu check-in/checkout |

**Table 11.** Narasi Use Case Riwayat Presensi

| Kategori   | Proses Aktor                                 | Proses Sistem  |   |
|------------|--|--|---|
| Basic Flow | 1. Pengguna membuka aplikasi.                |  |   |
|            | 2. Pengguna memilih menu "Riwayat Presensi". | 3. Sistem mengambil data presensi dari database berdasarkan bulan aktif. |   |
|            |  | 4. Sistem mengurutkan data presensi berdasarkan tanggal.                 |   |
|            | 5. Pengguna melihat daftar riwayat presensi. | 6. Sistem menampilkan daftar riwayat presensi ke layar pengguna.         |   |
|            | Alternative Flow                             | - Tidak tersedia.  | - (Opsional) Jika data tidak ditemukan, tampilkan notifikasi "Data tidak tersedia". |
|            | Postcondition                                | Pengguna dapat melihat histori kehadiran untuk bulan berjalan.           | Sistem hanya melakukan pembacaan data; tidak ada perubahan pada database.           |

e. Check-in Attendance (*mobile*)

|                      |   |
|----------------------|---|
| Nama <i>use case</i> | : Check-In Attendance   |
| Aktor                | : Pegawai   |
| Deskripsi            | : Pegawai melakukan absensi masuk kerja melalui Aplikasi mobile dengan tambahan autentikasi <i>face recognition</i> dan <i>gps</i> validasi |

**Table 12.** Narasi Usecase Check-in Attendance

| <b>Kategori</b>  | <b>Proses Aktor</b>                                  | <b>Proses Sistem</b>   |
|------------------|--|--|
| Basic Flow       | 1. Pengguna menekan tombol Check-in.                 | 2. Sistem memvalidasi lokasi menggunakan GPS.                  |
|                  | 3. Pengguna merekam Wajah                            | 4. Sistem melakukan verifikasi wajah pengguna.                 |
|                  |  | 5. Sistem menyimpan data kehadiran di database.                |
| Alternative Flow | - Jika verifikasi wajah gagal:                       | - Sistem menampilkan pesan error dan meminta verifikasi ulang. |
|                  | - Jika lokasi tidak valid:                           | - Sistem menolak check-in dan menampilkan pesan error.         |
| Postcondition    | Data kehadiran pengguna berhasil tercatat di sistem. | Sistem menyimpan data lokasi dan waktu check-in.               |

f. Check-out Attendance (*mobile*)

Nama *use case* : Check-Out Attendance

Aktor : Pegawai

Deskripsi : Pegawai melakukan absensi keluar kerja melalui Aplikasi mobile dengan tambahan autentikasi *face recognition* dan *gps* validasi

**Table 13.** Narasi Usecase Check-out Attendance

| <b>Kategori</b>  | <b>Proses Aktor</b>                   | <b>Proses Sistem</b>                                    |
|------------------|---------------------------------------|---|
| Basic Flow       | 1. Pengguna menekan tombol Check-out. | 2. Sistem memvalidasi lokasi menggunakan GPS.           |
|                  |                                       | 3. Sistem menyimpan data check-out di database.         |
| Alternative Flow | - Jika lokasi tidak valid:            | - Sistem menolak check-out dan menampilkan pesan error. |

| <b>Kategori</b> | <b>Proses Aktor</b>   | <b>Proses Sistem</b>                              |
|-----------------|---|---|
| Postcondition   | Data kehadiran pengguna berhasil diperbarui dengan waktu check-out. | Sistem menyimpan data lokasi dan waktu check-out. |

g. Analitik (*Mobile*)

|                      |   |
|----------------------|---|
| Nama <i>use case</i> | : Analitik  |
| Aktor                | : Pegawai   |
| Deskripsi            | : Pegawai dapat melihat statistik kehadiran pribadinya sendiri, seperti jumlah kehadiran, keterlambatan, izin, dan cuti selama periode tertentu (misalnya bulan ini) melalui tampilan analitik di menu analitik |

**Table 14.** Narasi Usecase Analitik

| <b>Kategori</b> | <b>Proses Aktor</b>                                    | <b>Proses Sistem</b>   |  |
|-----------------|--|--|--|
| Basic Flow      | 1. Pegawai membuka halaman Analitik.                   | 2. Sistem mengambil data statistik kehadiran pegawai dari database.                |  |
|                 |  | 3. Sistem menghitung dan merangkum data kehadiran (hadir, izin, sakit, terlambat). |  |
|                 | 4. Pegawai melihat grafik/statistik kehadiran pribadi. | 5. Sistem menampilkan visualisasi data dalam bentuk grafik atau angka.             |  |
|                 | Alternative Flow                                       | - Tidak ada data kehadiran selama periode yang dipilih.                            | - Sistem menampilkan notifikasi: "Data tidak tersedia untuk periode ini."  |
|                 | Postcondition  | Pegawai mendapatkan ringkasan kehadiran miliknya sendiri.                          | Sistem tidak mengubah data; hanya membaca dan menampilkan statistik ke UI. |

h. Profile (*Mobile*)

|                      |  |
|----------------------|--|
| Nama <i>use case</i> | : Analitik   |
| Aktor                | : Pegawai  |
| Deskripsi            | : Pegawai dapat melihat data pribadi seperti nama, email , alamat, foto profil, dan informasi jabatan melalui halaman profil. Sistem akan menampilkan data yang sudah tersimpan dan mengizinkan pembaruan jika diperlukan. |

**Table 15.** Narasi Use Case Profile

| Kategori         | Proses Aktor (Pegawai)  | Proses Sistem   |
|------------------|---|---|
| Basic Flow       | 1. Pegawai membuka halaman Profil.  | 2. Sistem menampilkan data profil pegawai dari database.                    |
|                  | 3. Pegawai mengubah informasi pribadi (opsional).                         | 4. Sistem menerima data baru dan memvalidasi input.                         |
|                  | 5. Pegawai menekan tombol Simpan.   | 6. Sistem memperbarui data di database dan menampilkan notifikasi berhasil. |
| Alternative Flow | - Pegawai mengisi data tidak valid (misal: email kosong).                 | - Sistem menampilkan pesan kesalahan dan meminta perbaikan input.           |
| Postcondition    | Data profil pegawai berhasil diperbarui atau ditampilkan tanpa perubahan. | Sistem menyimpan data baru jika valid, atau tetap menampilkan data lama.    |

i. Dashboard (*website*)

|                      |  |
|----------------------|--|
| Nama <i>use case</i> | : Dashboard                                |
| Aktor                | : Super Admin                              |
| Deskripsi            | : Melihat statistik umum tentang kehadiran |

**Table 16.** Narasi Use Case Dashboard Website

| Kategori   | Proses Aktor                            | Proses Sistem  |
|------------|---|--|
| Basic Flow | 1. Pengguna masuk ke halaman Dashboard. | 2. Sistem menampilkan statistik karyawan secara default. |

| Kategori         | Proses Aktor   | Proses Sistem  |
|------------------|--|--|
|                  | 3. Pengguna melakukan interaksi: Filter, Refresh, atau Cari. | 4. Sistem memproses sesuai aksi: cari, perbarui data, atau terapkan filter jumlah. |
|                  |  | 5. Sistem memperbarui tampilan halaman UI sesuai hasil pemrosesan.                 |
| Alternative Flow | - Jika pencarian tidak ditemukan.                            | - Sistem menampilkan pesan "Data tidak ditemukan" atau daftar kosong.              |
| Postcondition    | Pengguna dapat melihat data statistik yang sudah difilter.   | Tampilan UI diperbarui sesuai interaksi tanpa perubahan data permanen di database. |

j. Management Roles (*website*)

|                      |  |
|----------------------|--|
| Kode <i>use case</i> | : UC-W04   |
| Nama <i>use case</i> | : Management Roles   |
| Aktor                | : Super Admin  |
| Deskripsi            | : Mengelola peran (roles) dalam sistem, termasuk menambah, mengedit, menghapus, dan menetapkan permissions |

**Table 17.** Use Case Management Roles

| Kategori             | Proses Aktor  | Proses Sistem  |
|----------------------|---|--|
| Basic Flow: Add Role | 1. Admin membuka halaman untuk menambahkan role baru.                       | 2. Sistem menampilkan form untuk menambahkan role baru.                            |
|                      | 3. Admin mengisi data role, seperti nama, deskripsi, dan hak akses terkait. | 4. Sistem memvalidasi data untuk memastikan tidak ada duplikasi atau input kosong. |
|                      | 5. Admin menekan tombol Simpan.   | 6. Sistem menyimpan data role baru di database jika validasi berhasil.             |

| <b>Kategori</b>         | <b>Proses Aktor</b>  | <b>Proses Sistem</b>   |
|-------------------------|--|--|
|                         |  | 7. Sistem menampilkan notifikasi bahwa role berhasil ditambahkan.                        |
| Basic Flow: Edit Role   | 1. Admin memilih role yang akan diedit dari daftar role.                           | 2. Sistem menampilkan informasi detail dari role yang dipilih.                           |
|                         | 3. Admin mengubah informasi role, seperti nama, deskripsi, atau hak akses terkait. | 4. Sistem memvalidasi data yang diperbarui untuk memastikan konsistensi dan validitas.   |
|                         | 5. Admin menekan tombol Simpan.  | 6. Sistem memperbarui data role di database jika validasi berhasil.                      |
|                         |  | 7. Sistem menampilkan notifikasi bahwa perubahan berhasil disimpan.                      |
| Basic Flow: Delete Role | 1. Admin memilih role yang akan dihapus dari daftar role.                          | 2. Sistem menampilkan konfirmasi penghapusan untuk memastikan tindakan tidak disengaja.  |
|                         | 3. Admin memberikan konfirmasi untuk menghapus role.                               | 4. Sistem menghapus data role dari database jika tidak ada konflik.                      |
|                         |  | 5. Sistem menampilkan notifikasi bahwa role berhasil dihapus.                            |
| Alternative Flow        | 1. Jika validasi data gagal (Add/Edit Role)  | 2. Sistem menampilkan pesan error terkait validasi, seperti duplikasi atau input kosong. |

| Kategori      | Proses Aktor                                    | Proses Sistem   |
|---------------|---|---|
|               | 3. Jika role yang akan dihapus sedang digunakan | 4. Sistem menampilkan pesan error bahwa role tidak dapat dihapus karena sedang digunakan. |
| Postcondition | 1. Role baru berhasil ditambahkan, diperbarui   | 2. Sistem menyimpan perubahan di database   |

k. Management Permissions (*website*)

Nama *use case* : Manage Permissions

Aktor : Super Admin

Deskripsi : Mengelola hak akses (permissions) dalam sistem, termasuk menambah, mengedit, dan menghapus.

**Table 18.** Use Case Management Permissions

| Kategori                     | Proses Aktor  | Proses Sistem  |
|------------------------------|---|--|
| Basic Flow: Add Permissions  | 1. Admin membuka halaman untuk menambahkan permissions baru.  | 2. Sistem menampilkan form untuk menambahkan permissions baru.                     |
|                              | 3. Admin mengisi data Permissions, seperti nama dan deskripsi | 4. Sistem memvalidasi data untuk memastikan tidak ada duplikasi atau input kosong. |
|                              | 5. Admin menekan tombol Simpan.                               | 6. Sistem menyimpan data permissions baru di database jika validasi berhasil.      |
|                              |   | 7. Sistem menampilkan notifikasi bahwa permissions berhasil ditambahkan.           |
| Basic Flow: Edit Permissions | 1. Admin memilih Permissions yang                             | 2. Sistem menampilkan informasi detail dari  |

| <b>Kategori</b>                  | <b>Proses Aktor</b>   | <b>Proses Sistem</b>   |
|----------------------------------|---|--|
|                                  | akan diedit dari daftar permissions.  | permissions yang dipilih.  |
|                                  | 3. Admin mengubah informasi permissions, seperti nama, deskripsi, atau hak akses terkait. | 4. Sistem memvalidasi data yang diperbarui untuk memastikan konsistensi dan validitas.   |
|                                  | 5. Admin menekan tombol Simpan.   | 6. Sistem memperbarui data permissions di database jika validasi berhasil.               |
|                                  |   | 7. Sistem menampilkan notifikasi bahwa perubahan berhasil disimpan.                      |
| Basic Flow:<br>Delete Permission | 1. Admin memilih role yang akan dihapus dari daftar permissions.                          | 2. Sistem menampilkan konfirmasi penghapusan untuk memastikan tindakan tidak disengaja.  |
|                                  | 3. Admin memberikan konfirmasi untuk menghapus permissions                                | 4. Sistem menghapus data permissions dari database jika tidak ada konflik.               |
|                                  |   | 5. Sistem menampilkan notifikasi bahwa permissions berhasil dihapus.                     |
| Alternative Flow                 | 1. Jika validasi data gagal (Add/Edit permissions)  | 2. Sistem menampilkan pesan error terkait validasi, seperti duplikasi atau input kosong. |

| Kategori       | Proses Aktor   | Proses Sistem   |
|----------------|--|---|
|                | 3. Jika role yang akan dihapus sedang digunakan                                    | 4. Sistem menampilkan pesan error bahwa role tidak dapat dihapus karena sedang digunakan. |
| Post condition | 1. Role baru berhasil ditambahkan, diperbarui, atau dihapus sesuai tindakan Admin. | 2. Sistem menyimpan perubahan di database dan menampilkan notifikasi hasil proses.        |

1. Management Users (*website*)

|               |  |
|---------------|--|
| Name Use Case | : Management User  |
| Actor         | : Super Admin & Admin  |
| Deskripsi     | : Mengelola data pengguna (pegawai/staff), termasuk menambah, mengedit, menghapus, menetapkan roles, shift, dan lokasi kerja |

**Table 19.** Narasi Usecase Management Users

| Kategori             | Proses Aktor   | Proses Sistem  |
|----------------------|--|--|
| Basic Flow: Add User | 1. Admin membuka halaman untuk menambahkan user baru.                                  | 2. Sistem menampilkan form untuk menambahkan user baru.                            |
|                      | 3. Admin mengisi data user, seperti nama, email, nomor telepon, dan informasi lainnya. | 4. Sistem memvalidasi data untuk memastikan tidak ada duplikasi atau input kosong. |
|                      | 5. Admin menekan tombol Simpan.  | 6. Sistem menyimpan data user baru di database jika validasi berhasil.             |
|                      |  | 7. Sistem menampilkan notifikasi bahwa user berhasil ditambahkan.                  |

| <b>Kategori</b>                  | <b>Proses Aktor</b>  | <b>Proses Sistem</b>   |
|----------------------------------|--|--|
| Basic Flow: Edit User            | 1. Admin memilih user yang akan diedit dari daftar user.                   | 2. Sistem menampilkan informasi detail dari user yang dipilih.   |
|                                  | 3. Admin mengubah informasi user, seperti nama, email, atau nomor telepon. | 4. Sistem memvalidasi data yang diperbarui untuk memastikan konsistensi.   |
|                                  | 5. Admin menekan tombol Simpan.  | 6. Sistem memperbarui data user di database jika validasi berhasil.<br>7. Sistem menampilkan notifikasi bahwa perubahan berhasil disimpan. |
| Basic Flow: Delete User          | 1. Admin memilih user yang akan dihapus dari daftar user.                  | 2. Sistem menampilkan konfirmasi penghapusan untuk memastikan tindakan tidak disengaja.  |
|                                  | 3. Admin memberikan konfirmasi untuk menghapus user.                       | 4. Sistem memvalidasi.   |
|                                  |  | 5. Jika tidak ada konflik, sistem menghapus data user dari database.<br>6. Sistem menampilkan notifikasi bahwa user berhasil dihapus.      |
| Basic Flow: Assign Roles to User | 1. Admin membuka halaman daftar users.                                     | 2. Sistem menampilkan daftar users yang tersedia.  |
|                                  | 3. Admin memilih user yang akan diberikan roles.                           | 4. Sistem menampilkan daftar roles yang dapat dipilih.   |

| <b>Kategori</b>                             | <b>Proses Aktor</b>   | <b>Proses Sistem</b>   |
|---|---|--|
|   | 5. Admin memilih roles yang sesuai untuk user tersebut.           | 6. Sistem menyimpan data hubungan user dengan roles di database.                   |
|   |   | 7. Sistem menampilkan notifikasi bahwa roles berhasil diberikan ke user.           |
| Basic Flow:<br>Assign Shifts to<br>Users    | 1. Admin membuka halaman daftar users.                            | 2. Sistem menampilkan daftar users yang tersedia.                                  |
|   | 3. Admin memilih user yang akan diberikan shift.                  | 4. Sistem menampilkan daftar shift yang tersedia.                                  |
|   | 5. Admin memilih shift yang sesuai untuk user tersebut.           | 6. Sistem menyimpan data hubungan user dengan shift di database.                   |
|   |   | 7. Sistem menampilkan notifikasi bahwa shift berhasil diberikan ke user.           |
| Basic Flow:<br>Assign Locations<br>to Users | 1. Admin membuka halaman daftar users.                            | 2. Sistem menampilkan daftar users yang tersedia.                                  |
|   | 3. Admin memilih user yang akan diberikan lokasi presensi.        | 4. Sistem menampilkan daftar lokasi presensi yang tersedia.                        |
|   | 5. Admin memilih lokasi presensi yang sesuai untuk user tersebut. | 6. Sistem menyimpan data hubungan user dengan lokasi presensi di database.         |
|   |   | 7. Sistem menampilkan notifikasi bahwa lokasi presensi berhasil diberikan ke user. |

| <b>Kategori</b>  | <b>Proses Aktor</b>  | <b>Proses Sistem</b>  |
|------------------|--|---|
| Alternative Flow | 1. Jika validasi data gagal (Add/Edit/Delete User atau Assign Roles/Shifts/Locations):                                 | 2. Sistem menampilkan pesan error terkait validasi, seperti input kosong atau konflik data. |
|                  | 3. Jika data user memiliki hubungan terkait (role, shift, atau lokasi) saat penghapusan:                               | 4. Sistem menampilkan pesan error bahwa user tidak dapat dihapus.                           |
| Post condition   | 1. Data user berhasil ditambahkan, diperbarui, dihapus, atau diberikan roles, shift, dan lokasi sesuai tindakan Admin. | 2. Sistem menyimpan perubahan di database dan menampilkan notifikasi hasil proses.          |

m. Management Shift (*website*)

|               |   |
|---------------|---|
| Name Use Case | : Management Shift  |
| Actor         | : Super Admin   |
| Deskripsi     | : Mengelola waktu kerja (shifts), termasuk menambah, mengedit, menghapus shifts |

**Table 20.** Narasi Usecase Management Shift

| <b>Kategori</b>       | <b>Proses Aktor</b>  | <b>Proses Sistem</b>   |
|-----------------------|--|--|
| Basic Flow: Add Shift | 1. Admin membuka halaman untuk menambahkan shift baru.                           | 2. Sistem menampilkan form untuk menambahkan shift baru.                               |
|                       | 3. Admin mengisi data shift, seperti nama shift, waktu mulai, dan waktu selesai. | 4. Sistem memvalidasi data untuk memastikan tidak ada input kosong atau konflik waktu. |
|                       | 5. Admin menekan tombol 'Simpan'.  | 6. Sistem menyimpan data shift baru di database jika validasi berhasil.                |
|                       |  | 7. Sistem menampilkan notifikasi bahwa shift berhasil ditambahkan.                     |

| <b>Kategori</b>          | <b>Proses Aktor</b>   | <b>Proses Sistem</b>  |   |
|--------------------------|---|---|---|
| Basic Flow: Edit Shift   | 1. Admin memilih shift yang akan diedit dari daftar shift.        | 2. Sistem menampilkan informasi detail dari shift yang dipilih.                         |   |
|                          | 3. Admin mengubah informasi shift, seperti nama atau waktu shift. | 4. Sistem memvalidasi data yang diperbarui untuk memastikan konsistensi.                |   |
|                          | 5. Admin menekan tombol 'Simpan'.                                 | 6. Sistem memperbarui data shift di database jika validasi berhasil.                    |   |
|                          |   | 7. Sistem menampilkan notifikasi bahwa perubahan berhasil disimpan.                     |   |
| Basic Flow: Delete Shift | 1. Admin memilih shift yang akan dihapus dari daftar shift.       | 2. Sistem menampilkan konfirmasi penghapusan untuk memastikan tindakan tidak disengaja. |   |
|                          | 3. Admin memberikan konfirmasi untuk menghapus shift.             | 4. Sistem memvalidasi apakah shift sedang digunakan dalam hubungan dengan pengguna.     |   |
|                          |   | 5. Jika tidak ada konflik, sistem menghapus data shift dari database.                   |   |
|                          |   | 6. Sistem menampilkan notifikasi bahwa shift berhasil dihapus.                          |   |
|                          | Basic Flow: Assign Shifts to Users                                | 1. Admin membuka halaman daftar users.  | 2. Sistem menampilkan daftar users yang tersedia. |
|                          |   | 3. Admin memilih user yang akan diberikan shift.  | 4. Sistem menampilkan daftar shift yang tersedia. |
|                          |   |   |   |

| Kategori         | Proses Aktor   | Proses Sistem   |
|------------------|--|---|
|                  | 5. Admin memilih shift yang sesuai untuk user tersebut.  | 6. Sistem menyimpan data hubungan user dengan shift di database.                            |
|                  |  | 7. Sistem menampilkan notifikasi bahwa shift berhasil diberikan ke user.                    |
| Alternative Flow | 1. Jika validasi data gagal (Add/Edit/Delete Shift atau Assign Shift):                                     | 2. Sistem menampilkan pesan error terkait validasi, seperti input kosong atau konflik data. |
|                  | 3. Jika shift yang akan dihapus sedang digunakan oleh pengguna:  | 4. Sistem menampilkan pesan error bahwa shift tidak dapat dihapus.                          |
| Postcondition    | 1. Data shift berhasil ditambahkan, diperbarui, dihapus, atau diberikan kepada user sesuai tindakan Admin. | 2. Sistem menyimpan perubahan di database dan menampilkan notifikasi hasil proses.          |

n. Management Locations (*website*)

|               |   |
|---------------|---|
| Name Use Case | : Management Locations  |
| Actor         | : Super Admin   |
| Deskripsi     | : Mengelola lokasi kerja, termasuk menambah, mengedit, dan menghapus lokasi |

**Table 21.** Narasi Usecase Management Locations

| Kategori                 | Proses Aktor  | Proses Sistem  |
|--------------------------|---|--|
| Basic Flow: Add Location | 1. Admin membuka halaman untuk menambahkan lokasi                             | 2. Sistem menampilkan form untuk menambahkan lokasi                                |
|                          | 3. Admin mengisi data lokasi, seperti nama lokasi, alamat, dan koordinat GPS. | 4. Sistem memvalidasi data untuk memastikan tidak ada input kosong atau duplikasi. |

| <b>Kategori</b>             | <b>Proses Aktor</b>   | <b>Proses Sistem</b>  |
|-----------------------------|---|---|
|                             | 5. Admin menekan tombol 'Simpan'.   | 6. Sistem menyimpan data lokasi baru di database jika validasi berhasil.                |
|                             |   | 7. Sistem menampilkan notifikasi bahwa lokasi berhasil ditambahkan.                     |
| Basic Flow: Edit Location   | 1. Admin memilih lokasi yang akan diedit dari daftar lokasi.                  | 2. Sistem menampilkan informasi detail dari lokasi yang dipilih.                        |
|                             | 3. Admin mengubah informasi lokasi, seperti nama, alamat, atau koordinat GPS. | 4. Sistem memvalidasi data yang diperbarui untuk memastikan konsistensi.                |
|                             | 5. Admin menekan tombol 'Simpan'.   | 6. Sistem memperbarui data lokasi di database jika validasi berhasil.                   |
|                             |   | 7. Sistem menampilkan notifikasi bahwa perubahan berhasil disimpan.                     |
| Basic Flow: Delete Location | 1. Admin memilih lokasi yang akan dihapus dari daftar lokasi.                 | 2. Sistem menampilkan konfirmasi penghapusan untuk memastikan tindakan tidak disengaja. |
|                             | 3. Admin memberikan konfirmasi untuk menghapus lokasi.                        | 4. Sistem memvalidasi apakah lokasi sedang digunakan dalam hubungan dengan pengguna.    |
|                             |   | 5. Jika tidak ada konflik, lokasi dari database.  |
|                             |   | 6. Sistem menampilkan notifikasi bahwa lokasi berhasil dihapus.                         |

| <b>Kategori</b>                     | <b>Proses Aktor</b>   | <b>Proses Sistem</b>  |
|-------------------------------------|---|---|
| Basic Flow: Assign User to Location | 1. Admin membuka halaman daftar lokasi.   | 2. Sistem menampilkan daftar lokasi yang tersedia.  |
|                                     | 3. Admin memilih lokasi yang akan diberikan pengguna (user).  | 4. Sistem menampilkan daftar pengguna yang dapat dipilih.                                   |
|                                     | 5. Admin memilih pengguna yang akan dihubungkan ke lokasi tersebut.   | 6. Sistem menyimpan data hubungan lokasi dengan pengguna di database.                       |
|                                     |   | 7. Sistem menampilkan notifikasi bahwa user berhasil dihubungkan dengan lokasi.             |
| Alternative Flow                    | 1. Jika validasi data gagal (Add/Edit/Delete Location atau Assign User to Location):                          | 2. Sistem menampilkan pesan error terkait validasi, seperti input kosong atau konflik data. |
|                                     | 3. Jika lokasi yang akan dihapus sedang digunakan oleh pengguna:  | 4. Sistem menampilkan pesan error bahwa lokasi tidak dapat dihapus.                         |
| Postcondition                       | 1. Data lokasi berhasil ditambahkan, diperbarui, dihapus, atau dihubungkan dengan user sesuai tindakan Admin. | 2. Sistem menyimpan perubahan di database dan menampilkan notifikasi hasil proses.          |

o. Management Areas

Name Use Case : Management Areas

Actor : Super Admin

Deskripsi : Mengelola area sebagai kumpulan lokasi, termasuk menambah, mengedit, menghapus, dan

menetapkan lokasi ke dalam area dengan validasi sistem.

**Table 22** Narasi Use Case Management Areas

| <b>Kategori</b>         | <b>Proses Aktor</b>  | <b>Proses Sistem</b>   |
|-------------------------|--|--|
| Basic Flow: Add Area    | 1. Super Admin membuka halaman untuk menambahkan area.                       | 2. Sistem menampilkan form input area.                                 |
|                         | 3. Super Admin mengisi nama area dan memilih lokasi yang akan dikelompokkan. | 4. Sistem memvalidasi input: nama unik dan lokasi valid.               |
|                         | 5. Super Admin menekan tombol "Simpan".                                      | 6. Sistem menyimpan area dan relasi lokasi ke database.                |
|                         |  | 7. Sistem menampilkan notifikasi bahwa area berhasil ditambahkan.      |
| Basic Flow: Edit Area   | 1. Super Admin memilih area yang akan diedit dari daftar.                    | 2. Sistem menampilkan informasi area yang dipilih.                     |
|                         | 3. Super Admin mengubah nama area atau daftar lokasi.                        | 4. Sistem memvalidasi data input baru.                                 |
|                         | 5. Super Admin menekan tombol "Simpan".                                      | 6. Sistem memperbarui area dan relasinya di database.                  |
|                         |  | 7. Sistem menampilkan notifikasi bahwa perubahan berhasil disimpan.    |
| Basic Flow: Delete Area | 1. Super Admin memilih area yang akan dihapus dari daftar.                   | 2. Sistem menampilkan konfirmasi penghapusan.                          |
|                         | 3. Super Admin memberikan konfirmasi penghapusan.                            | 4. Sistem memvalidasi apakah area sedang digunakan dalam proses aktif. |

| Kategori                       | Proses Aktor   | Proses Sistem  |
|--------------------------------|--|--|
|                                |  | 5. Jika tidak ada konflik, sistem menghapus area dan relasinya dari database.        |
|                                |  | 6. Sistem menampilkan notifikasi bahwa area berhasil dihapus.                        |
| Basic Flow:<br>Assign Location | 1. Super Admin membuka daftar area.  | 2. Sistem menampilkan daftar area.   |
|                                | 3. Super Admin memilih area untuk ditambahkan lokasi.                                | 4. Sistem menampilkan daftar lokasi yang tersedia.                                   |
|                                | 5. Super Admin memilih lokasi untuk dihubungkan ke area tersebut.                    | 6. Sistem menyimpan hubungan area-lokasi di database.                                |
|                                |  | 7. Sistem menampilkan notifikasi bahwa lokasi berhasil dihubungkan dengan area.      |
| Alternative Flow               | 1. Jika validasi data gagal (nama kosong, lokasi tidak dipilih, area duplikat, dll). | 2. Sistem menampilkan pesan error validasi.  |
|                                | 3. Jika area sedang digunakan dan ingin dihapus.                                     | 4. Sistem menampilkan pesan error bahwa area tidak dapat dihapus.                    |
| Postcondition                  | 1. Area berhasil ditambahkan, diperbarui, dihapus, atau dikaitkan dengan lokasi.     | 2. Sistem menyimpan perubahan ke database dan menampilkan notifikasi hasil tindakan. |

p. Attendance (*website*)

Name Use Case : Attendance

Actor : Super Admin, Admin

Deskripsi : Memungkinkan mencatat kehadiran manual karyawan, di mana waktu dicatat otomatis dan data akan divalidasi

oleh Super Admin.

**Table 23** Narasi Use Case Attendace

| <b>Kategori</b>          | <b>Proses Aktor</b>  | <b>Proses Sistem</b>  |
|--------------------------|--|---|
| Basic Flow:<br>Check-in  | 1. Admin Absensi membuka halaman Attendance.                     | 2. Sistem menampilkan form input check-in manual.                           |
|                          | 3. Admin memilih nama karyawan dan klik tombol "Add Attendance". | 4. Sistem mencatat waktu check-in secara otomatis (timestamp server).       |
|                          |  | 5. Sistem menyimpan data kehadiran  |
|                          |  | 6. Sistem menampilkan notifikasi bahwa data kehadiran berhasil ditambahkan. |
| Basic Flow:<br>Check-out | 1. Admin Absensi memilih karyawan yang sudah check-in.           | 2. Sistem menampilkan tombol "Check-out" jika data check-in tersedia.       |
|                          | 3. Admin klik tombol "Check-out".                                | 4. Sistem mencatat waktu check-out otomatis dan memperbarui data kehadiran. |
|                          |  | 5. Sistem menampilkan notifikasi bahwa data berhasil diperbarui.            |
| Postcondition            | 1. Data kehadiran berhasil dicatat dan/atau diperbarui.          | 2. Sistem menyimpan kehadiran   |

q. Laporan (*website*)

Name Use Case : Laporan

Actor : Super Admin, Admin

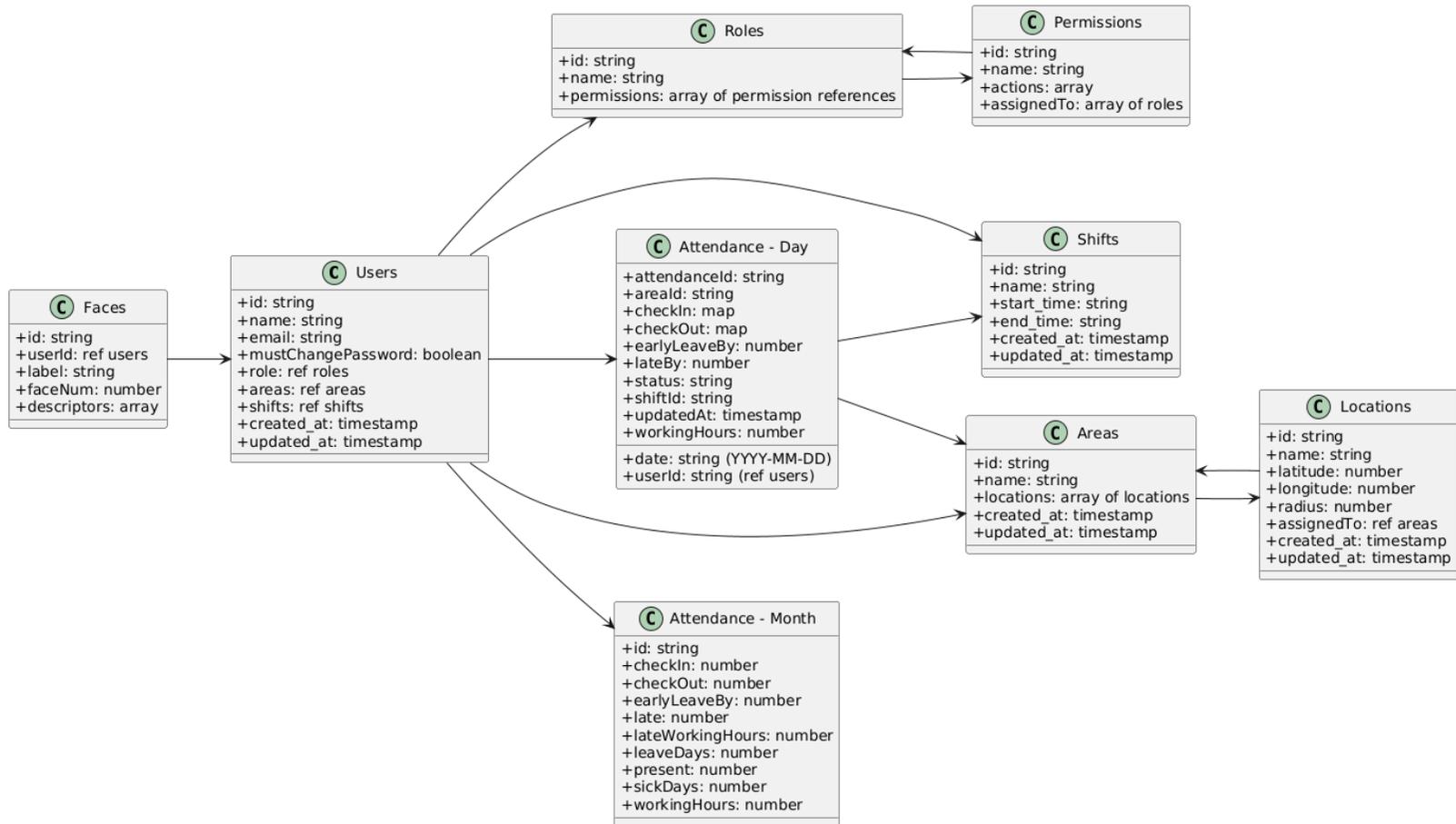
Deskripsi : Memungkinkan Admin dan Super Admin melihat laporan kehadiran berdasarkan tanggal dan cabang, serta mengunduhnya dalam format Excel.

**Table 24.** Narasi Usecase Laporan

| <b>Kategori</b>                 | <b>Proses Aktor</b>  | <b>Proses Sistem</b>  |
|---------------------------------|--|---|
| Basic Flow:<br>Lihat<br>Laporan | 1. Pengguna membuka halaman laporan.   | 2. Sistem menampilkan form filter laporan (tanggal dan cabang).                         |
|                                 | 3. Pengguna memilih rentang tanggal.   | 4. Sistem menampilkan data berdasarkan filter.  |
|                                 | 5. (Super Admin saja) memilih cabang yang ingin ditampilkan.                                     | 6. Sistem menampilkan data berdasarkan cabang terpilih (atau cabang login untuk Admin). |
| Basic Flow:<br>Unduh Excel      | 7. Pengguna mengklik tombol "Unduh Excel".   | 8. Sistem mengunduh data laporan yang sedang ditampilkan dalam format Excel.            |
| Alternative<br>Flow             | 1. Jika tidak ada data untuk filter yang dipilih.  | 2. Sistem menampilkan pesan "Data tidak ditemukan".                                     |
| Postcondition                   | Pengguna mendapatkan laporan kehadiran sesuai filter dan (jika diunduh) dalam bentuk file Excel. | Sistem hanya membaca data dan tidak melakukan perubahan ke database.                    |

## 2. Class Diagram

Class Diagram system yang di rancang ini, terdapat delapan kelas utama yang akan diimplementasikan ke dalam sistem menggunakan Firebase Firestore sebagai basis data, yaitu: User, Role, Permission, Shift, Location, Area, Face, dan Attendance. Setiap kelas merepresentasikan komponen logis dalam sistem dan mencakup atribut serta relasi yang dibutuhkan untuk menjalankan fungsionalitas sistem secara keseluruhan.



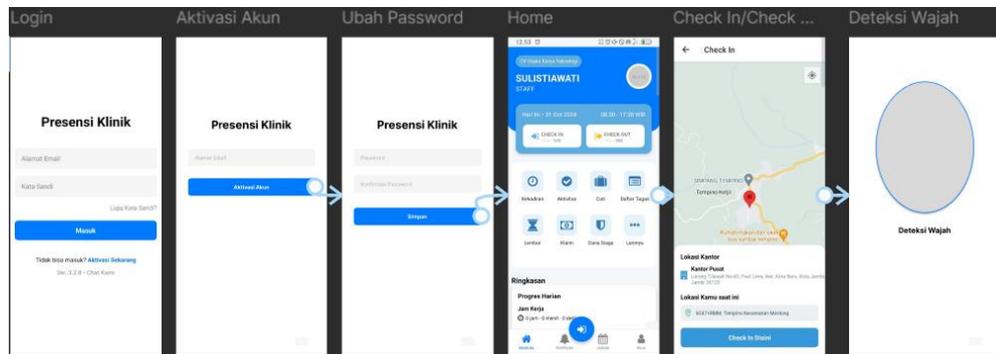
**Gambar 14.** *Class Diagram* Sistem Presensi

### 3. Desain Interface

Adapun rancangan desain interface sistem Presensi Klinik Pratama Dokter Yanti yaitu telah dibuat oleh peneliti sebelumnya. Sehingga pada penelitian ini, akan dilanjutkan dengan menerjemahkan desain interface tersebut ke dalam bentuk kode program sehingga sistem Presensi Klinik Pratama Dokter Yanti yaitu dapat diimplementasikan. Proses implementasi desain interface ini akan menggunakan bahasa Typescript dimana untuk web dashboard nya menggunakan *Framework Next Js* dan untuk mobile nya menggunakan *Fremework Reach Native*.

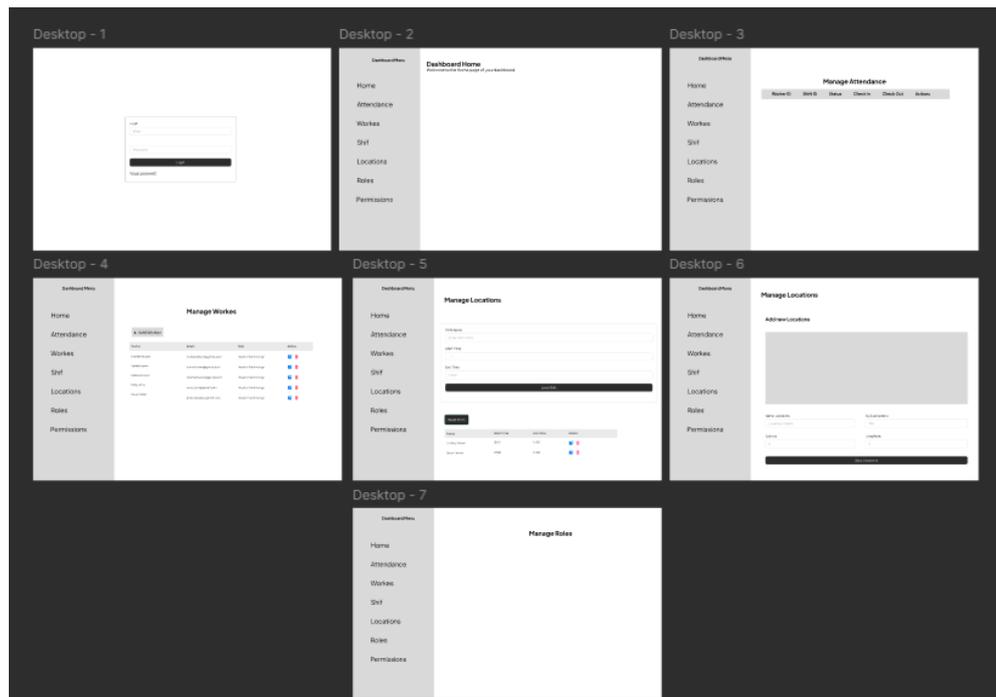
Berikut adalah hasil dari rancangan desain interface yang telah dilakukan oleh penelitian sebelumnya.

#### a. Desain User interface Mobile



**Gambar 15.** Desain User Interface Mobile

## b. Desain User interface Website



**Gambar 16.** Desain User Interface Website

### Pengujian Sistem

Pengujian sistem dilakukan untuk mengevaluasi kualitas dan kinerja sistem yang telah dikembangkan. Dalam penelitian ini pengujian dilakukan menggunakan UAT, dimana terduri dua tahap pengujian utama, yaitu *Alpha Testing* dan *Beta Testing*. *Alpha Testing* dilakukan di lingkungan internal pengembangan menggunakan metode *Black Box Testing*, yang berfokus pada pengujian aspek fungsionalitas sistem untuk memastikan bahwa setiap fitur berjalan sesuai dengan spesifikasi yang telah ditentukan. Metode ini efektif dalam mengidentifikasi kesalahan fungsi tanpa perlu mengetahui struktur internal perangkat lunak (Tresnawati et al. 2021). Setelah itu, *Beta Testing* dilakukan dengan melibatkan pengguna akhir di lingkungan nyata. Pada tahap ini, pengujian dilakukan menggunakan skala Likert untuk mengukur tingkat kepuasan dan penerimaan pengguna terhadap sistem yang telah dikembangkan. Pendekatan ini memungkinkan pengembang memperoleh umpan balik langsung dari pengguna guna mengetahui sejauh mana sistem telah memenuhi kebutuhan dan harapan pengguna secara menyeluruh (Afrianto et al. 2021).

### Beta Testing dengan Blackbox

Pada Blackbox testing proses evaluasi melibatkan pengguna, dengan tujuan memastikan bahwa prototipe yang telah dibangun memenuhi kebutuhan

dan harapan pelanggan. Jika hasil pengujian menunjukkan kesesuaian dengan keinginan pengguna, maka langkah berikutnya akan dilanjutkan. Namun, jika ditemukan ketidaksesuaian, proses pengembangan akan diulang pada tahap-tahap sebelumnya untuk melakukan perbaikan.

Persentase keberhasilan dan kegagalan (*valid* dan tidak *valid*) akan disajikan melalui perhitungan, disertai dengan kesimpulan pengujian yang diperoleh dari interpretasi atau penilaian terhadap hasil penelitian. Pada tahap ini, solusi yang telah dirancang akan diimplementasikan ke dalam sistem, kemudian dievaluasi untuk memastikan apakah solusi tersebut sudah berjalan sesuai dengan harapan pengguna. Evaluasi ini mencakup pengujian fungsionalitas sistem menggunakan metode *Black Box Testing*. Pengujian fungsional dilakukan dengan memberikan input tertentu dan memeriksa output yang dihasilkan oleh sistem.

Metode ini bertujuan untuk menilai apakah sistem berfungsi sesuai dengan yang diharapkan. Format tabel untuk pengujian ini mengacu pada penelitian (Mahendra et al. 2022) yang mencakup elemen-elemen seperti deskripsi pengujian, hasil yang diharapkan, dan hasil pengujian. Berikut adalah format tabel yang digunakan dalam pengujian sistem ini

**Table 25.** Pengujian Black-Box Testing

| <b>ID</b> | <b>Deskripsi Pengujian</b> | <b>Hasil yang di<br/>harapkan</b> | <b>Hasil<br/>Pengujian</b>   |
|-----------|----------------------------|-----------------------------------|------------------------------|
| ID-1      | Skenario yang di jalankan  | Spesifikasi yang di lakukan       | Diterima atau Tidak Diterima |

Pada tahap pengujian menggunakan metode Black Box, peneliti telah menyusun skenario pengujian yang akan digunakan untuk mengevaluasi sistem. Skenario ini dirancang untuk memastikan setiap fungsi sistem dapat berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Berikut adalah skenario pengujian yang telah dirancang.

**Table 26.** Skenario Pengujian Aplikasi Mobile Sistem Presensi

| <b>ID</b> | <b>Deskripsi Pengujian</b>                  | <b>Hasil yang<br/>Diharapkan</b>    | <b>Hasil<br/>Pengujian</b> |
|-----------|---|-------------------------------------|----------------------------|
|           | Test Case Login                             |                                     |                            |
| TC-M-01   | Login dengan akun dan kata sandi yang valid | Pengguna berhasil masuk ke aplikasi | -                          |

| <b>ID</b>                                 | <b>Deskripsi Pengujian</b>               | <b>Hasil yang Diharapkan</b>   | <b>Hasil Pengujian</b> |
|---|--|--|------------------------|
| TC-M -02                                  | Login dengan akun tidak terdaftar        | Sistem menolak login dan menampilkan pesan error                         | -                      |
| TC-M -03                                  | Login dengan kata sandi salah            | Sistem menolak login dan menampilkan pesan kesalahan                     | -                      |
| TC-M-04                                   | Login dengan format email tidak valid    | Sistem menolak input email dan tampilkan peringatan validasi             | -                      |
| Test Case Ubah Password                   |  |  |                        |
| TC-M-05                                   | Ubah password dengan Konfirmasi Password | Sistem menampilkan pesan password tidak sesuai atau kesalahan            | -                      |
| Test Case Dashboard                       |  |  |                        |
| TC-M-06                                   | Tampilkan kehadiran hari ini             | Status kehadiran hari ini ditampilkan dengan benar                       | -                      |
| Test Case Presensi (check in / check out) |  |  |                        |
| TC-M-07                                   | Presensi masuk valid                     | Presensi masuk tercatat dan lokasi serta wajah terverifikasi             | -                      |
| TC-M-08                                   | Presensi pulang valid                    | Presensi pulang tercatat dan lokasi serta wajah terverifikasi            | -                      |
| TC-M-09                                   | Wajah tidak dikenali saat presensi       | Sistem menampilkan pesan bahwa wajah tidak dikenali dan menolak presensi | -                      |
| TC-M-10                                   | Lokasi di luar zona saat presensi        | Sistem menampilkan pesan bahwa lokasi tidak sesuai dan menolak presensi  | -                      |
| Test Case Riwayat                         |  |  |                        |

| <b>ID</b>          | <b>Deskripsi Pengujian</b>          | <b>Hasil yang Diharapkan</b>                                 | <b>Hasil Pengujian</b> |
|--------------------|-------------------------------------|--|------------------------|
| TC-M-11            | Lihat riwayat kehadiran             | Riwayat presensi ditampilkan dengan benar                    | -                      |
| Test Case Analitik |                                     |  |                        |
| TC-M-12            | Lihat grafik kehadiran bulanan      | Grafik menampilkan data kehadiran bulanan dengan benar       | -                      |
| TC-M-13            | Lihat ringkasan statistik kehadiran | Ringkasan total hadir, izin, alpha, dan telat tampil lengkap | -                      |
| Test Case Profile  |                                     |  |                        |
| TC-M-14            | Tampilkan profil pegawai            | Data pegawai tampil lengkap di halaman profil                | -                      |

**Table 27.** Skenario Pengujian Aplikasi Website Sistem Presensi

| <b>ID</b>           | <b>Deskripsi Pengujian</b>                  | <b>Hasil yang Diharapkan</b>                                 | <b>Hasil Pengujian</b> |
|---------------------|---|--|------------------------|
| Test Case Login     |   |  |                        |
| TC-W-01             | Login dengan akun dan kata sandi yang valid | Pengguna berhasil masuk ke aplikasi                          | -                      |
| TC-W -02            | Login dengan akun tidak terdaftar           | Sistem menolak login dan menampilkan pesan error             | -                      |
| TC-W -03            | Login dengan kata sandi salah               | Sistem menolak login dan menampilkan pesan kesalahan         | -                      |
| TC-W-04             | Login dengan format email tidak valid       | Sistem menolak input email dan tampilkan peringatan validasi | -                      |
| Test Case Dashboard |   |  |                        |
| TC-W-05             | Aktor membuka halaman Dashboard             | Statistik kehadiran ditampilkan                              | -                      |

| <b>ID</b>             | <b>Deskripsi Pengujian</b>                | <b>Hasil yang Diharapkan</b>                 | <b>Hasil Pengujian</b> |
|-----------------------|---|--|------------------------|
| TC-W-06               | Aktor menggunakan filter tanggal          | Statistik diperbarui sesuai filter           | -                      |
| TC-W-07               | Aktor mencari karyawan                    | Statistik sesuai nama ditampilkan            | -                      |
| TC-W-08               | Aktor mencari data tidak tersedia         | Pesan "Data tidak ditemukan" ditampilkan     | -                      |
| TC-W-09               | Aktor klik tombol refresh                 | Statistik diperbarui dengan data terbaru     | -                      |
| Test Case Roles       |   |  |                        |
| TC-W-11               | Membuka halaman tambah Role               | Form tambah Role ditampilkan                 | -                      |
| TC-W-12               | Menambahkan role baru                     | Role berhasil ditambahkan, notifikasi muncul | -                      |
| TC-W-13               | Mengedit role                             | Role berhasil diperbarui                     | -                      |
| TC-W-14               | Menghapus role                            | Role berhasil dihapus                        | -                      |
| TC-W-15               | Assign Permission Kedalam Roles           | Role berhasil diperbarui                     | -                      |
| Test Case Permissions |   |  |                        |
| TC-W-16               | Membuka halaman tambah Permissions        | Form tambah Permissions ditampilkan          | -                      |
| TC-W-17               | Menambahkan permission baru               | Permission berhasil ditambahkan              | -                      |
| TC-W-18               | Mengedit permission                       | Permission berhasil diperbarui               | -                      |
| TC-W-19               | Menghapus permission yang tidak digunakan | Permission berhasil dihapus                  | -                      |
| Test Case Users       |   |  |                        |
| TC-W-20               | Menambahkan user baru                     | User berhasil ditambahkan                    | -                      |
| TC-W-21               | Menambahkan user dengan data kosong       | Validasi gagal, error muncul                 | -                      |

| <b>ID</b>            | <b>Deskripsi Pengujian</b>            | <b>Hasil yang Diharapkan</b>         | <b>Hasil Pengujian</b> |
|----------------------|---------------------------------------|--------------------------------------|------------------------|
| TC-W-22              | Mengedit user dengan data valid       | User berhasil diperbarui             | -                      |
| TC-W-23              | Menghapus user                        | User berhasil dihapus                | -                      |
| TC-W-24              | Memberikan role ke user               | Role berhasil diberikan ke user      | -                      |
| TC-W-25              | Memberikan shift ke user              | Shift berhasil diberikan ke user     | -                      |
| TC-W-26              | Memberikan lokasi ke user             | Lokasi berhasil diberikan ke user    | -                      |
| Test Case Shifts     |                                       |                                      |                        |
| TC-W-27              | Menambahkan shift baru                | Shift berhasil ditambahkan           | -                      |
| TC-W-28              | Menambahkan shift dengan data kosong  | Validasi gagal, error muncul         | -                      |
| TC-W-29              | Mengedit shift                        | Shift berhasil diperbarui            | -                      |
| TC-W-30              | Menghapus shift                       | Shift berhasil dihapus               | -                      |
| Test Case Locations  |                                       |                                      |                        |
| TC-W-31              | Menambahkan lokasi baru               | Lokasi berhasil ditambahkan          | -                      |
| TC-W-32              | Menambahkan lokasi dengan data kosong | Validasi gagal, error muncul         | -                      |
| TC-W-33              | Mengedit lokasi                       | Lokasi berhasil diperbarui           | -                      |
| TC-W-34              | Menghapus lokasi                      | Lokasi berhasil dihapus              | -                      |
| Test Case Areas      |                                       |                                      |                        |
| TC-W-34              | Menambahkan area baru                 | Area berhasil ditambahkan            | -                      |
| TC-W-35              | Mengedit area                         | Area berhasil diperbarui             | -                      |
| TC-W-36              | Menghapus area                        | Area berhasil dihapus                | -                      |
| TC-W-37              | Assign lokasi ke area                 | Lokasi berhasil dihubungkan ke area  | -                      |
| Test Case Attendance |                                       |                                      |                        |
| TC-W-38              | Check-in manual karyawan              | Timestamp check-in tercatat otomatis | -                      |

| <b>ID</b>         | <b>Deskripsi Pengujian</b>                    | <b>Hasil yang Diharapkan</b>               | <b>Hasil Pengujian</b> |
|-------------------|---|--|------------------------|
| TC-W-39           | Check-out karyawan                            | Timestamp check-out diperbarui otomatis    | -                      |
| Test Case Laporan |   |  |                        |
| TC-W-40           | Melihat laporan berdasarkan tanggal           | Laporan ditampilkan sesuai rentang tanggal | -                      |
| TC-W-41           | Melihat laporan berdasarkan cabang            | Laporan ditampilkan sesuai cabang          | -                      |
| TC-W-42           | Mengunduh laporan                             | File Excel laporan berhasil diunduh        | -                      |
| TC-W-43           | Laporan tidak ditemukan untuk filter tertentu | Pesan "Data tidak ditemukan" ditampilkan   | -                      |

#### **Beta Testing dengan Skala Linkert**

Pengujian Beta Testing dilakukan untuk mengetahui sejauh mana aplikasi dapat diterima dan dinilai oleh pengguna akhir. Metode yang digunakan dalam tahap ini adalah skala Likert, yang berfungsi untuk mengukur tingkat penilaian dan persepsi pengguna terhadap aplikasi. Pengujian ini melibatkan Seluruh karyawan Klinik yang diminta untuk menjawab pertanyaan yang telah disusun melalui Kuesioner. Setiap pertanyaan memiliki beberapa pilihan jawaban dengan bobot nilai yang telah ditentukan, sebagaimana tercantum dalam tabel

**Table 28** Skala Linker Beta Testing

| <b>Kode</b> | <b>Keterangan</b>   | <b>Bobot</b> |
|-------------|---------------------|--------------|
| SS          | Sangat Setuju       | 5            |
| S           | Setuju              | 4            |
| KS          | Kurang Setuju       | 3            |
| TS          | Tidak Setuju        | 2            |
| STS         | Sangat Tidak Setuju | 1            |