

## BAB 3 METODOLOGI PENELITIAN

Dalam membangun dan merancang sistem presensi karyawan, digunakan metode *throw away prototyping*. Spesifikasi awal dari sistem pada proses *throw away prototyping* sudah dapat diketahui di awal, sehingga proses *prototyping* ini ditujukan untuk mengurangi resiko kebutuhan yang tidak terpenuhi [3]. Terdapat beberapa proses yang harus dilakukan pada metode *throw away prototyping*, yaitu: *Establish Outline Specs, Develop Prototype, Evaluate Prototype, Specify System, Design and Implement System, dan Validate System.*

### 3.1 Establish Outline Specs

Dalam melakukan penelitian dengan metode *throw away prototyping*, tahapan pertama adalah melakukan identifikasi masalah dan kebutuhan dari sistem yang akan dibangun. Riset pengguna penting untuk dilakukan dalam membangun pemahaman untuk merancang dan membangun sistem untuk orang-orang yang menjadi target pengguna sistem tersebut. Target pengguna dari sistem yang akan dibuat adalah karyawan di Kompas Gramedia. Pada perusahaan Kompas Gramedia pada saat ini tidak mempunyai sistem presensi yang dapat diakses dimana saja. Sementara pada saat ini, terdapat banyak karyawan yang bekerja secara *Work From Home* (WFH) akibat terjadinya pandemi Covid-19. Sehingga, karyawan yang bekerja dari rumah tidak dapat melakukan presensi. Kemudian, untuk dapat mengetahui kebutuhan pengguna dari permasalahan tersebut dilakukan wawancara dengan HRIS mengenai *Mobile Attendance System*. Hasil dari wawancara dapat dilihat pada lampiran 3. Setelah melakukan wawancara dengan *manager* HRIS, terdapat *requirements* dari *Mobile Attendance System* yang dapat dilihat pada lampiran 4.

### 3.2 Develop Prototype

Setelah berhasil melakukan identifikasi dan mendapatkan kebutuhan dari sistem yang akan dibangun dari data yang diperoleh melalui wawancara dengan *manager* HRIS, tahapan selanjutnya yaitu dengan melakukan perancangan desain *mock up* dari sistem yang akan dibangun dan kemudian akan dilakukan presentasi kepada *department* HRIS terkait desain yang telah dibuat apakah sudah sesuai atau

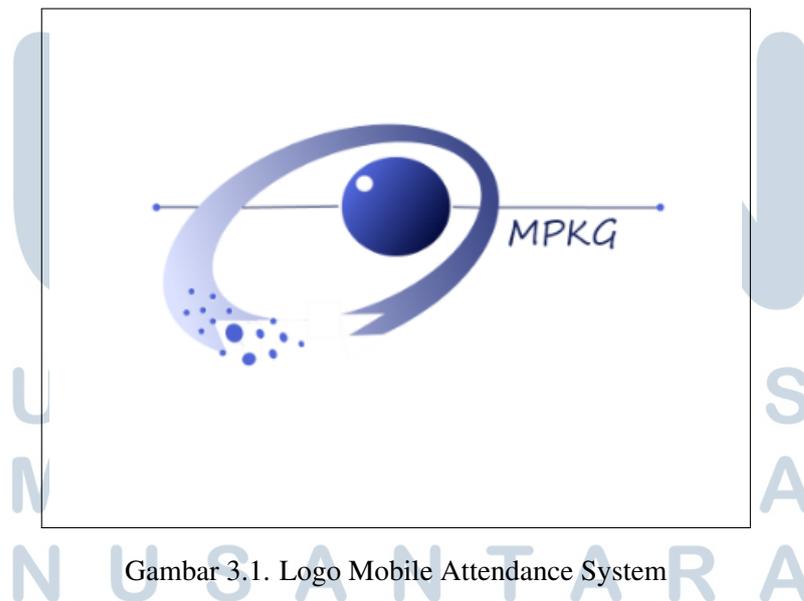
belum. Desain *mock up* dari sistem yang akan dibangun akan dibuat menggunakan *software* Adobe XD.

### 3.2.1 Rancangan Design Prototype

Untuk mengetahui gambaran desain antarmuka yang akan dibuat, dirancang beberapa desain prototipe menggunakan Adobe XD. Terjadi beberapa perubahan pada desain prototipe yang dibuat setelah dilakukan presentasi desain kepada tim HRIS. Perancangan desain prototipe mengikuti standar desain dari Kompas Gramedia.

#### A Logo Mobile Attendance System

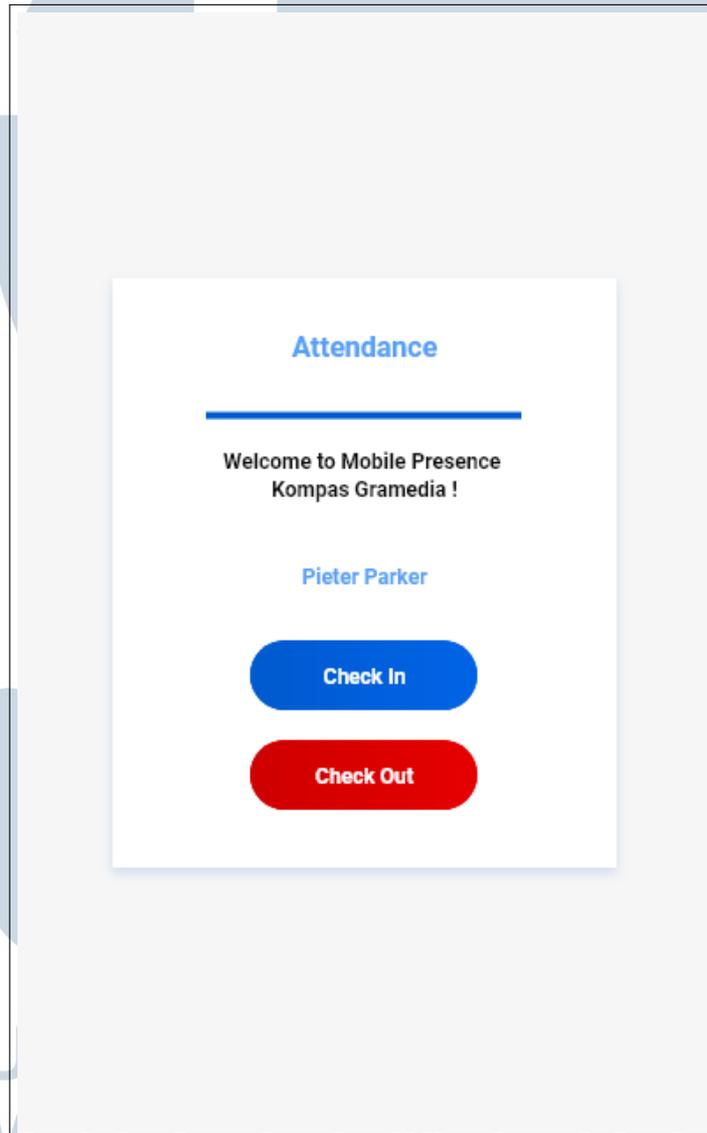
Pembuatan desain dari logo sistem presensi sesuai dengan keinginan dari *manager* HRIS yaitu Bapak FX. Endri Harmanto. MPKG merupakan singkatan dari *Mobile Attendance* Kompas Gramedia. Warna utama dari logo tersebut adalah warna biru karena, biru merupakan warna ciri khas sistem Kompas Gramedia. Logo ini terlihat seperti sebuah planet yang dikelilingi oleh meteor-meteor dan meteor tersebut jatuh tepat di MPKG. Pembuatan desain tersebut menggunakan *software* Adobe XD.



Gambar 3.1. Logo Mobile Attendance System

## B Landing Page Check In dan Check Out

Gambar 3.2 merupakan rancangan desain antarmuka untuk halaman *landing* pada sistem presensi. Tulisan Pieter Parker merupakan nama karyawan yang *login*. Sistem tersebut akan menampilkan nama karyawan yang *login* secara otomatis pada halaman *landing* sistem presensi.



Gambar 3.2. Landing page check in

## C Check In Form

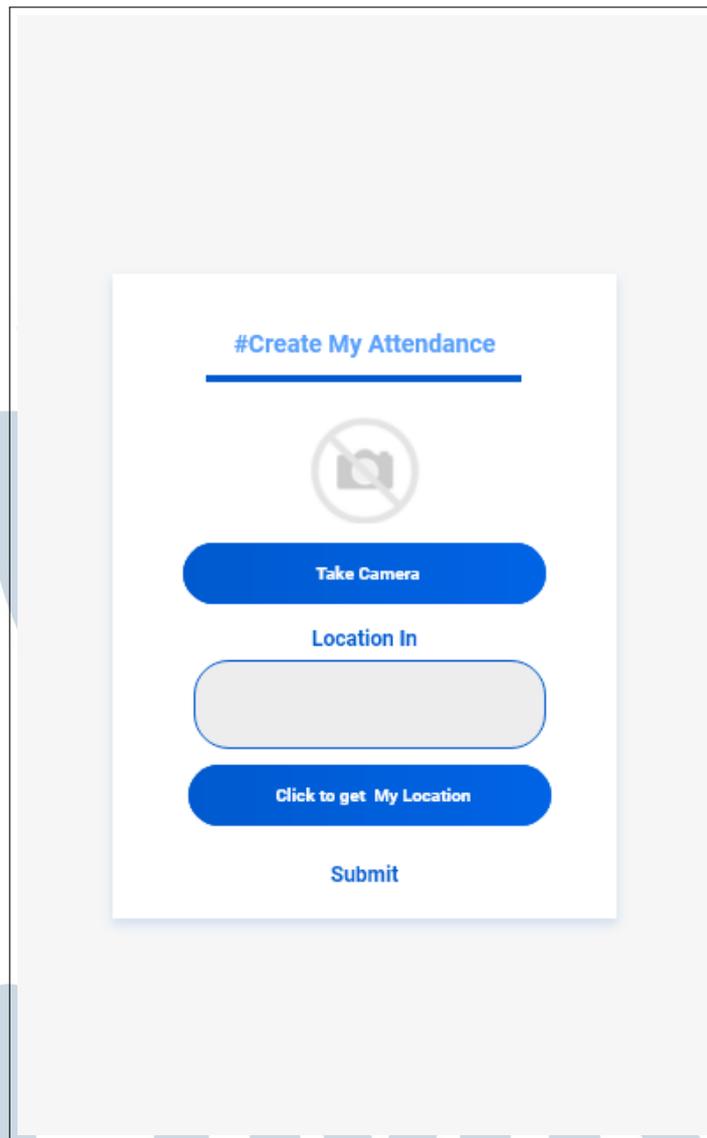
Gambar 3.3 merupakan halaman *form* untuk *check in*. Setelah karyawan menekan tombol *check in* pada halaman *landing*, karyawan akan diarahkan ke

halaman *form check in*. Tombol *take camera* bermanfaat dalam menangkap foto dari kamera. Foto yang diambil dengan kamera akan tampil pada kotak yang berada di atas tombol *take camera*. Jika karyawan menekan kotak yang berisi foto, foto tersebut akan menjadi besar atau berbentuk *thumbnail* dan dapat di *close*.

Tombol *Click to get My Location* bermanfaat untuk mengambil lokasi keberadaan karyawan saat melakukan presensi. Data lokasi akan tampil di kotak yang berada tepat di atas tombol tersebut dan kotak tersebut tidak dapat diubah atau *disable*.

Tombol *submit* bermanfaat dalam menyimpan data presensi ke dalam *database*. Jika karyawan menekan tombol *submit*, akan muncul notifikasi jika data presensi masuk telah berhasil disimpan ke dalam *database* dan akan diarahkan ke menu *history attendance*, dimana menu *history* akan menggunakan tampilan bawaan dari ERP (Odoo).





Gambar 3.3. Check in form

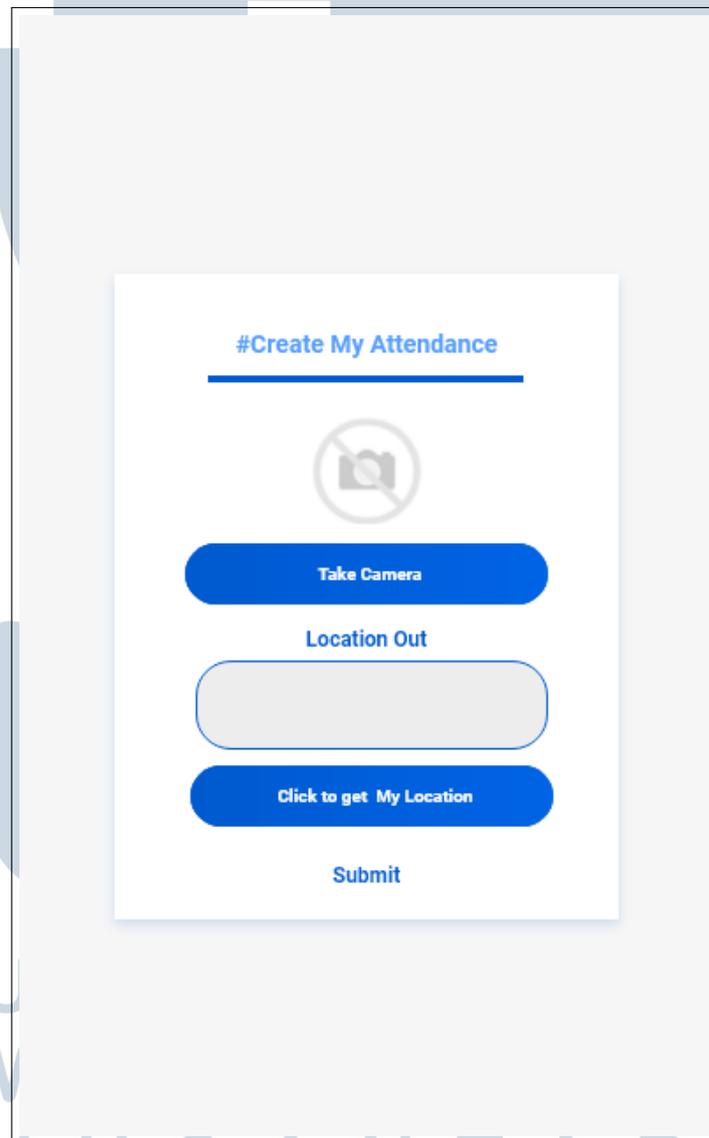
#### D Check Out Form

Gambar 3.4 merupakan halaman *form* untuk *check out*. Setelah karyawan menekan tombol *check out* pada halaman *landing*, karyawan akan diarahkan ke halaman *form check out*. Tombol *take camera* bermanfaat dalam menangkap foto dari kamera. Foto yang diambil dengan kamera akan pada kotak yang berada di atas tombol *take camera*. Jika karyawan menekan kotak yang berisi foto, foto tersebut akan menjadi besar atau berbentuk *thumbnail* dan dapat ditutup.

Tombol *Click to get My Location* bermanfaat untuk mengambil lokasi keberadaan karyawan saat melakukan presensi. Data lokasi akan tampil di kotak

yang berada tepat di atas tombol tersebut dan kotak tersebut tidak dapat diubah atau *disable*.

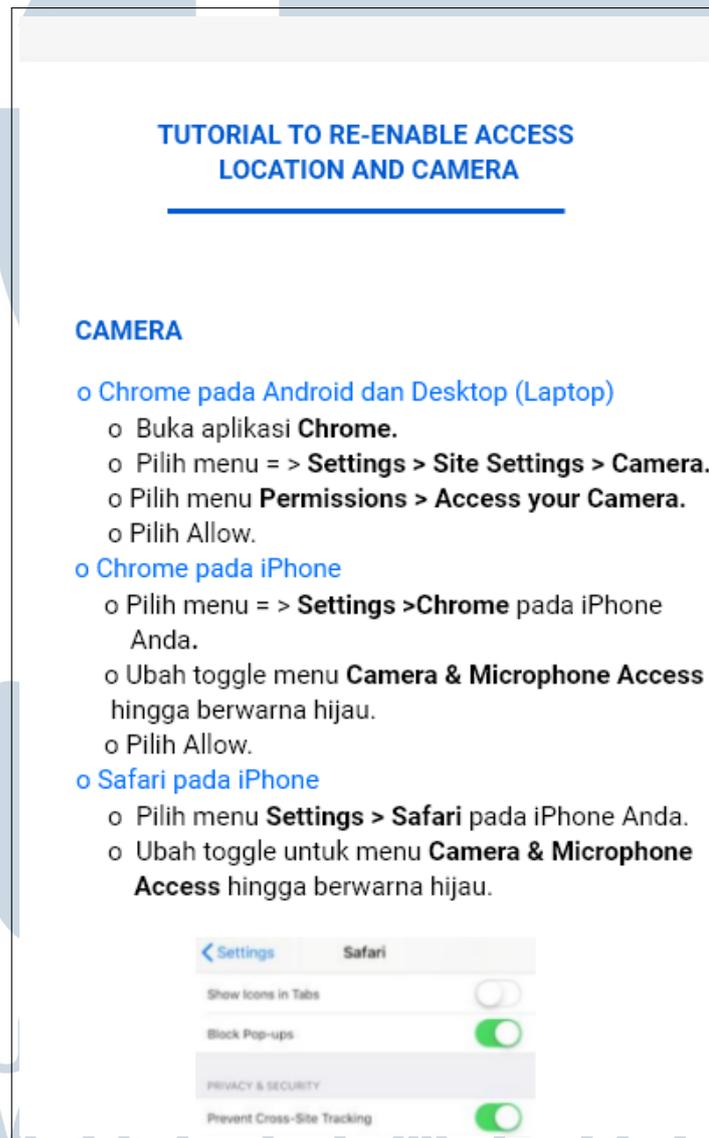
Tombol *submit* bermanfaat dalam menyimpan data presensi ke dalam *database*. Jika karyawan menekan tombol *submit*, akan muncul notifikasi jika data presensi masuk telah berhasil disimpan ke dalam *database* dan akan diarahkan ke menu *history attendance*, dimana menu *history* akan menggunakan tampilan bawaan dari ERP (Odoo).



Gambar 3.4. Check out form

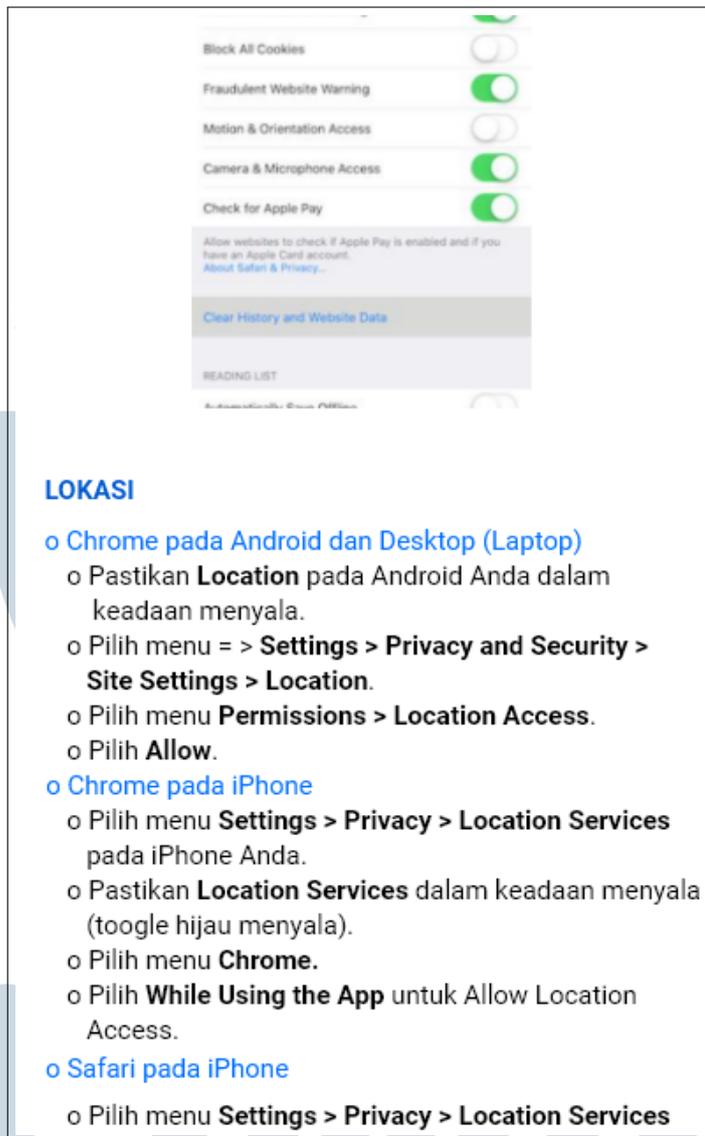
## E Tutorial akses kembali izin kamera

Gambar 3.5 merupakan *tutorial* yang tujuannya untuk memberitahukan kepada karyawan yang menggunakan *Mobile Attendance System* ketika tidak sengaja menutup akses perizinan untuk akses kamera. Pada halaman tersebut, terdapat *tutorial* mulai dari pengguna ponsel Android sampai dengan ponsel iOS.



Gambar 3.5. Tutorial re-enable access camera

Gambar 3.6 merupakan *tutorial* yang berguna untuk mengarahkan karyawan dalam mengaktifkan kembali izin lokasi, terdapat *tutorial* pada ponsel Android sampai iOS.



Gambar 3.6. Tutorial re-enable access location

## F Kanban view History Attendances untuk Self Service

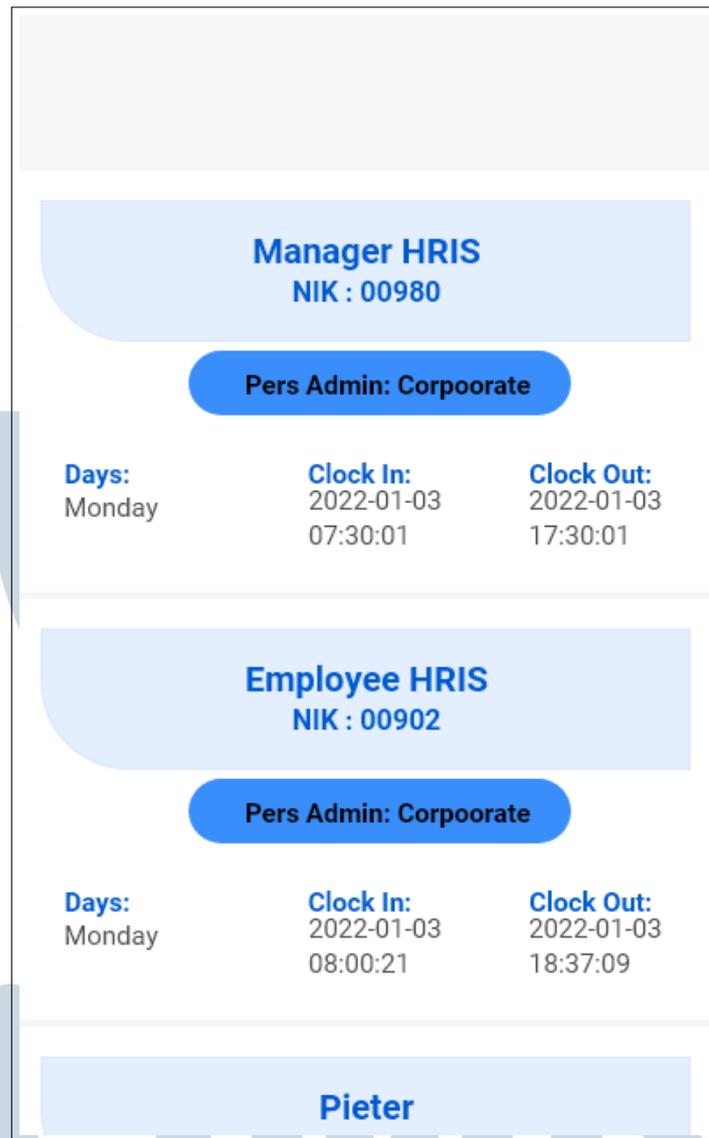
Gambar 3.7 merupakan tampilan pada menu *History Attendance* yang bermanfaat untuk melihat data-data presensi dirinya sendiri. Pada halaman tersebut hanya terlihat data hari, waktu masuk dan waktu pulang.

<b>Days:</b> Monday	<b>Clock In:</b> 2022-01-03 07:30:01	<b>Clock Out:</b> 2022-01-03 17:30:01
<b>Days:</b> Tuesday	<b>Clock In:</b> 2022-01-04 08:01:21	<b>Clock Out:</b> 2022-01-04 18:35:06
<b>Days:</b> Wednesday	<b>Clock In:</b> 2022-01-05 08:00:01	<b>Clock Out:</b> 2022-01-05 17:30:09
<b>Days:</b> Thursday	<b>Clock In:</b> 2022-01-06 07:00:09	<b>Clock Out:</b> 2022-01-06 17:33:07
<b>Days:</b> Friday	<b>Clock In:</b> 2022-01-07 -----	<b>Clock Out:</b> 2022-01-07 -----

Gambar 3.7. Kanban view history attendances untuk self service

### G Kanban view History Attendances untuk Manager dan HR Admin

Gambar 3.8 merupakan tampilan pada menu *Employees Attendance History* untuk *Manager* yang ingin melihat data bawahannya dan *HR Admin* yang ingin melihat data semua karyawan yang mempunyai *pers admin* yang sama dengan *HR* yang sedang melakukan *login* pada sistem. Pada tampilan tersebut hanya menampilkan nama karyawan, NIK karyawan, *pers admin* karyawan, waktu masuk, waktu pulang, dan hari.

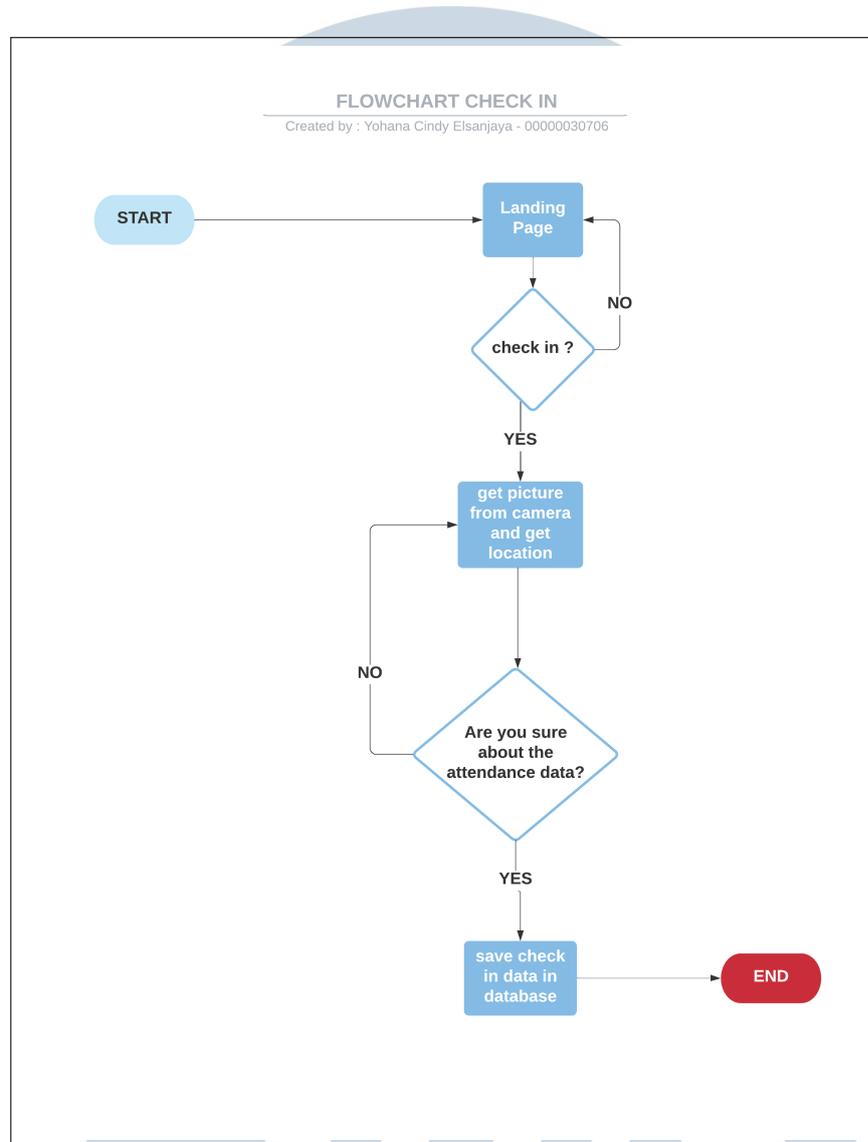


Gambar 3.8. Kanban view history attendances untuk manager dan hr admin

### 3.2.2 Flowchart

Untuk memperjelas cara kerja dari sistem presensi, dibuat beberapa *flowchart* sebagai panduan dalam pembangunan sistem.

## A Flowchart Check In

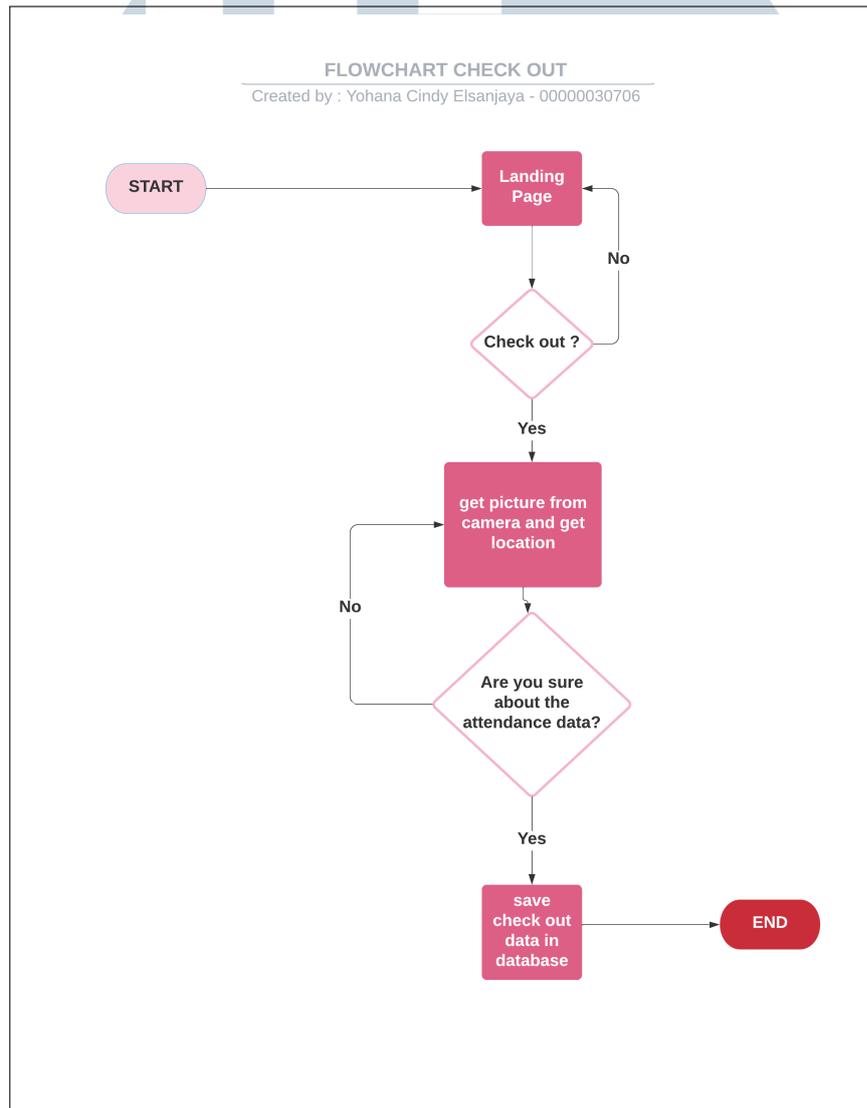


Gambar 3.9. Flowchart check in

Gambar 3.9 merupakan *flowchart check in* dari sistem presensi. Sistem presensi akan dimulai dari halaman *landing* dan jika karyawan ingin melakukan presensi maka, akan mengambil foto dari kamera dan lokasi keberadaan saat melakukan presensi. Kemudian, jika karyawan sudah yakin dengan data presensi yang dimasukkan maka akan mengirimkan data presensi ke *database* dan sistem akan selesai. Namun, jika karyawan belum yakin dengan data presensi tersebut, maka karyawan dapat melakukan pengambilan foto dari kamera dan lokasi keberadaan saat melakukan presensi kembali. Jika karyawan tidak ingin melakukan

presensi masuk (*check in*), maka karyawan akan tetap berada di halaman *landing* sistem.

## B Flowchart Check Out

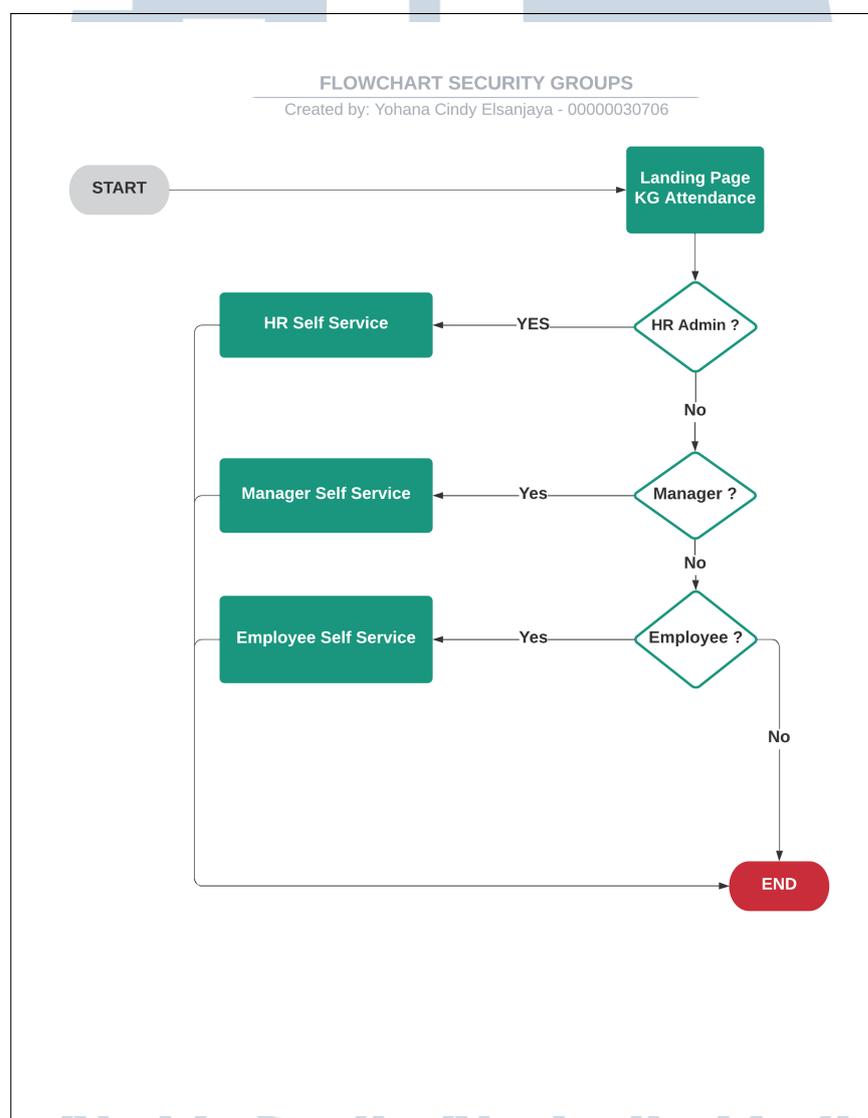


Gambar 3.10. Flowchart check out

Pada Gambar 3.10 merupakan *flowchart check out* dari sistem presensi. Sistem presensi dimulai dari halaman *landing*. Jika karyawan tidak ingin melakukan presensi pulang (*check out*), maka akan tetap berada di halaman *landing* sistem. Namun, jika karyawan ingin melakukan presensi pulang, karyawan akan mengambil data foto dari kamera dan lokasi keberadaan saat melakukan presensi.

Jika data sudah sesuai, data akan dikirim ke dalam *database* dan sistem akan berakhir. Namun, jika belum sesuai maka karyawan dapat melakukan pengambil data foto dari kamera dan lokasi keberadaan saat melakukan presensi. Jika karyawan tidak ingin melakukan presensi pulang (*check out*), maka karyawan akan tetap berada di halaman *landing* sistem.

### C Flowchart Kelompok Security



Gambar 3.11. Kelompok security Mobile Attendance System

Gambar 3.11 merupakan *flowchart* kelompok *security* dari sistem presensi. Sistem presensi dimulai dari halaman *landing page KG Attendance*. Terdapat 3

jenis akses, yaitu *Employee Self Service* yang dapat mengelola data diri sendiri, *Manager Self Service* yang dapat mengelola data bawahannya, dan *HR Self Service* yang dapat mengelola data karyawan yang mempunyai *pers admin* yang sama.

Jika karyawan yang *login* merupakan *HR Admin*, akan dapat melakukan *HR Self Service*. Namun, jika karyawan yang *login* bukan *HR Admin*, tetapi merupakan *Manager*, akan dapat melakukan *Manager Self Service*. Namun, jika karyawan yang *login* bukan merupakan *HR Admin* dan *Manager*, tetapi merupakan *Employee*, hanya dapat melakukan *Employee Self Service*.

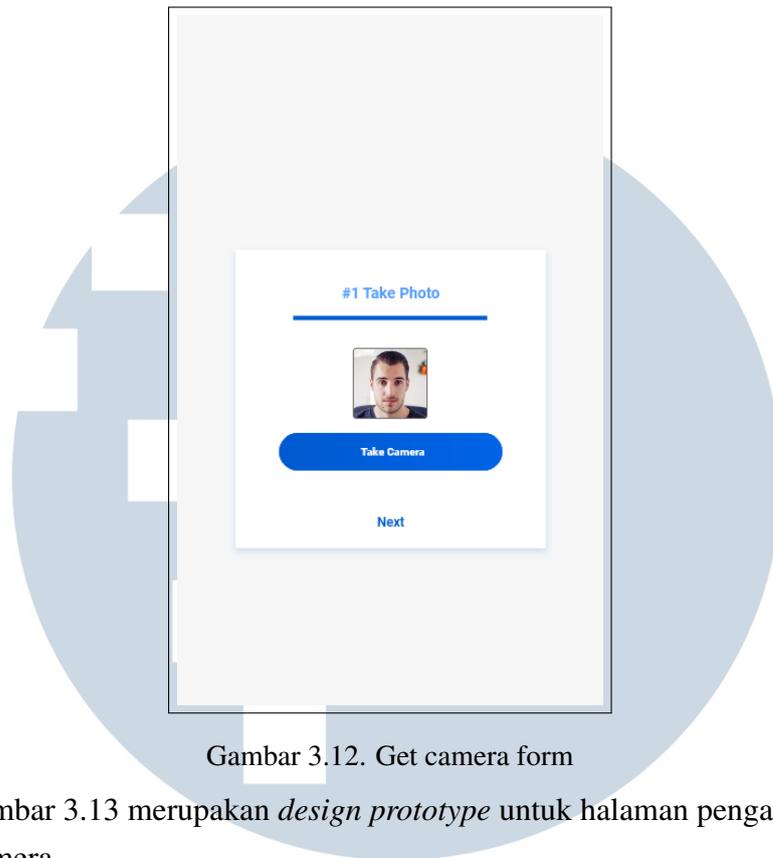
### 3.3 Evaluate Prototype

Tahap selanjutnya akan dilakukan pengujian terhadap desain *prototype* yang telah selesai dirancang dengan presentasi *design prototyping* kepada *department HRIS* dan tim *HRIS* dapat memberikan komentar maupun saran mengenai *design prototyping* yang dibuat. Jika hasil evaluasi masih belum baik, akan dilakukan perancangan *prototype* kembali sesuai dengan saran dan komentar dari tim *HRIS* saat dilakukan presentasi. Namun, jika hasil evaluasi sudah baik akan dilanjutkan ke tahap berikutnya. Dalam hal ini, dilakukan satu kali perubahan terhadap desain yang telah dirancang. Berikut merupakan iterasi *design* yang terjadi.

#### 3.3.1 Rancangan Design Awal

Gambar 3.12 merupakan *design prototype* untuk halaman pengambilan foto dengan kamera.





Gambar 3.12. Get camera form

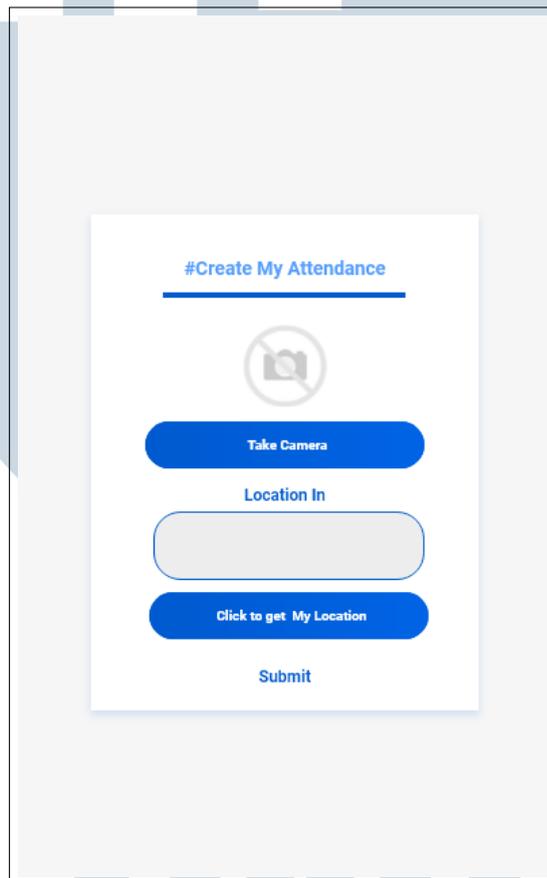
Gambar 3.13 merupakan *design prototype* untuk halaman pengambilan foto dengan kamera.



Gambar 3.13. Get location form

Setelah dilakukan presentasi, pihak HRIS meminta untuk mengubah *design*

*prototype* yang awalnya untuk mengambil kamera dan lokasi berada di *form view* yang berbeda menjadi di satu halaman yang sama. Hal ini disebabkan karena, jika untuk mengambil kamera dan lokasi berada di halaman yang berbeda, akan membuat karyawan sulit dan terlihat kurang praktis. Sehingga *design prototype* berubah menjadi seperti Gambar 3.14.



The image shows a mobile application interface for creating an attendance record. The form is titled "#Create My Attendance" and is displayed on a light gray background. The form itself is white and contains the following elements from top to bottom: a camera icon with a diagonal slash through it, a blue button labeled "Take Camera", a text input field labeled "Location In", a blue button labeled "Click to get My Location", and a blue button labeled "Submit".

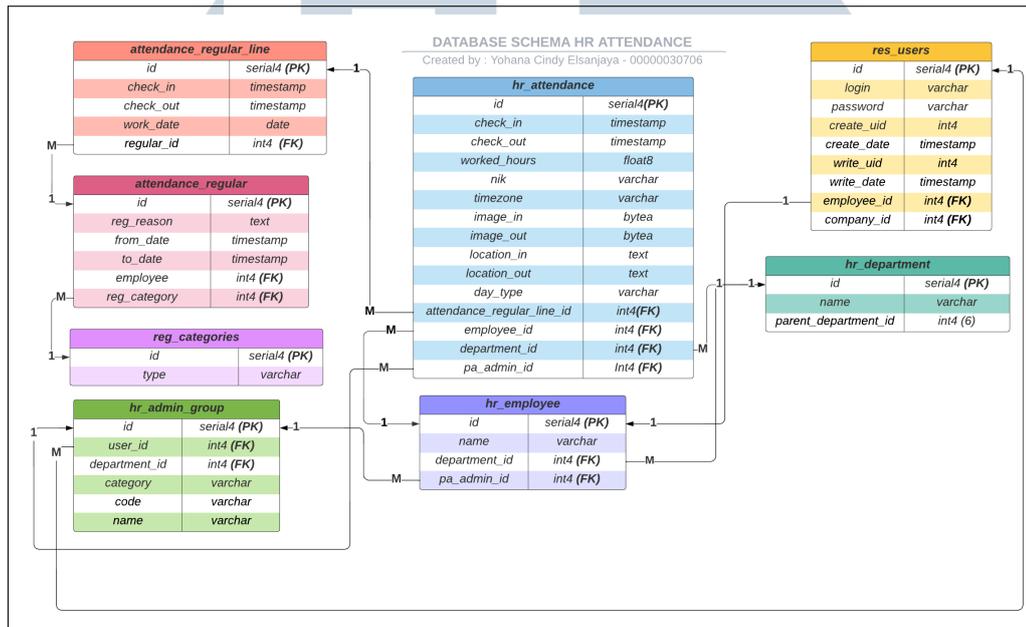
Gambar 3.14. Form attendance

### 3.4 Specify System

Sistem akan dibangun menggunakan ERP (Odoo) versi 11. Sementara untuk *database* menggunakan PostgreSQL. ERP (Odoo) sudah memiliki *template view* dalam pembuatan tampilannya. Namun, dilakukan *custom view* untuk membangun rancangan *design* sesuai dengan *design prototype* yang telah dirancang sebelumnya. Untuk pembuatan *custom view* menggunakan bahasa pemrograman Less.CSS.

### 3.4.1 Database Schema

Database yang digunakan untuk membangun sistem presensi adalah PostgreSQL versi 13. Berikut merupakan Database Schema dari sistem presensi yang terhubung dengan beberapa tabel.



Gambar 3.15. Database schema Mobile Attendance System

Gambar 3.15 menunjukkan rancangan database pada Mobile Attendance System dimana terdapat tabel *hr attendance* yang terhubung dengan tujuh tabel lainnya. Pada tabel *hr attendance*, terdapat beberapa data yaitu: *id* yang merupakan *primary key* dari tabel tersebut, *check in* yang bermanfaat sebagai penyimpan data waktu masuk karyawan, *check out* yang bermanfaat sebagai penyimpan data waktu pulang karyawan, *location in* yang bermanfaat untuk menyimpan data lokasi masuk karyawan, *location out* yang bermanfaat untuk menyimpan data lokasi pulang karyawan, *image in* yang bermanfaat untuk menyimpan data foto masuk karyawan, *image out* yang bermanfaat untuk menyimpan data foto pulang karyawan, *day type* yang bermanfaat untuk menyimpan data hari saat karyawan melakukan presensi, *timezone* untuk menyimpan wilayah karyawan saat melakukan presensi masuk, *worked hours* untuk menyimpan berapa banyak waktu karyawan bekerja dan beberapa data yang merupakan *foreign key* ke tabel lain yang bermanfaat menghubungkan tabel *hr attendance* ke beberapa tabel yang lainnya. Pada tabel *hr attendance*, terhubung ke empat tabel, yaitu: *hr admin group*, *hr employee*, *hr department*, dan *attendance regular line*.

Tabel *res users* mempunyai beberapa data, yaitu: *id* yang merupakan *primary key* dari tabel tersebut, *login* yang bermanfaat sebagai penyimpan data email karyawan, *password* yang bermanfaat sebagai penyimpan data *password* karyawan, dan dua data *foreign key* untuk menghubungkan antara tabel *res users* dengan *hr employee* dan *hr company*

Tabel *hr employee* mempunyai beberapa data, yaitu: *id* yang merupakan *primary key* dari tabel tersebut, *name* yang bermanfaat menyimpan nama karyawan dan *foreign key* yang bermanfaat menghubungkan ke *hr department*.

Tabel *hr department* yang mempunyai data *id* yang merupakan *primary key* dari tabel tersebut, *name* untuk menyimpan nama *department*, *parent department id* yang bermanfaat untuk menyimpan *department* paling atas.

Tabel *attendance regular line* yang mempunyai beberapa data, yaitu: *id* yang merupakan *primary key* dari tabel tersebut, *check in* yang bermanfaat untuk menyimpan data kehadiran masuk, *check out* yang bermanfaat dalam menyimpan kehadiran pulang, *work date* yang bermanfaat untuk menyimpan tanggal kerja, dan *foreign key* dari tabel *attendance regular* yang bermanfaat untuk menghubungkan tabel *attendance regular line* ke tabel *attendance regular*.

Tabel *attendance regular* mempunyai beberapa data, yaitu: *id* yang merupakan *primary key* dari tabel tersebut, *reg reason* yang bermanfaat untuk menyimpan alasan absen, *from date* yang bermanfaat untuk menyimpan tanggal awal pengajuan absensi susulan, *to date* yang bermanfaat untuk menyimpan tanggal akhir pengajuan absensi susulan, dan dua *foreign key* yang bermanfaat untuk menghubungkan tabel *attendance regular* ke tabel *hr employee* dan *reg categories*.

Tabel *reg categories* mempunyai beberapa data, yaitu: *id* yang merupakan *primary key* dari tabel tersebut dan *type* yang bermanfaat untuk menyimpan beberapa jenis regularisasi absen.

Tabel *hr admin group* mempunyai beberapa data, yaitu: *id* yang merupakan *primary key* dari tabel tersebut, *category* yang bermanfaat dalam menyimpan kategori dari *hr admin group*, *name* yang bermanfaat dalam menyimpan nama kelompok *pers admin*, *code* yang bermanfaat dalam menyimpan kode *pers admin*, dan dua *foreign key* yang bermanfaat untuk menghubungkan tabel *hr admin group* ke tabel *res users* dan *hr department*.

### 3.4.2 Template Views Pada Odoo

Pada Odoo terdapat beberapa jenis dari *template views* yang merupakan *template* bawaan dari Odoo yang dapat digunakan untuk membuat halaman tampilan pada sistem tanpa melakukan modifikasi. Berikut merupakan *Template Views* yang digunakan.

#### A Form Views

Gambar 3.16 merupakan potongan kode untuk membuat tampilan berbentuk *form view* pada Odoo. Terdapat pemanggilan beberapa *field* yang berguna untuk menampilkan data dari *field* yang dipanggil. Sementara *model* adalah nama tabel yang dipanggil. *Mobile Attendance System* melakukan modifikasi *form views* yang telah ada.

Gambar 3.16 merupakan kode yang berguna untuk membuat tampilan berbentuk *form*. Form yang dibuat merupakan modifikasi dari modul vendor. Kode **field name="nik"** bertujuan untuk memanggil data nik pada tampilan *form*.

```
<record id="kg_tms_hr_attendance_view_form_inherit" model="ir.ui.view">
  <field name="name">hr.attendance.form.inherit</field>
  <field name="model">hr.attendance</field>
  <field name="inherit_id" ref="hr_attendance.hr_attendance_view_form"/>
  <field name="arch" type="xml">
    <xpath expr="//form" position="attributes">
      <attribute name="create">>false</attribute>
      <attribute name="edit">>false</attribute>
    </xpath>
    <field name="employee_id" position="attributes">
      <attribute name="readonly">0</attribute>
      <attribute name="no_create">1</attribute>
      <attribute name="no_create_edit">1</attribute>
      <attribute name="widget">selection</attribute>
    </field>
    <field name="employee_id" position="after">
      <field name="nik"/>
      <field name="timezone" required="1" force_save="1"/>
      <field name="department_id" attrs="{ 'readonly': 1 }" force_save="1"/>
    </field>
  </field>
</record>
```

Gambar 3.16. Potongan kode form view

Gambar 3.17 merupakan hasil tampilan pada sistem setelah menuliskan kode seperti pada Gambar 3.16. Data-data yang tampil pada *form* sesuai *field* yang dipanggil pada penulisan kode *form views*.

The screenshot displays a mobile application interface with a top navigation bar containing icons for a grid, a notification, a clock, a chat bubble, a profile picture, and a menu. The main content area shows a form with the following fields:

- Employee**: Tuty Pricillia
- NIK Employee**: 00040401
- Timezone**: Asia/Jakarta
- Department**: CORPORATE HUMAN RESOURCE
- Days**: Monday
- Checkin Date**: 2022-06-06 23:02:00
- Worked hours**: 11:00
- Checkout Date**: 2022-06-07 10:02:00
- Image In**: A small square image placeholder.
- Image Out**: A small square image placeholder.
- Location In**: Jalan Cemara, RW 04, Perumahan Kavling-Komplek DKI, Kembangan, Ja Barat, Daerah Khusus Ibukota Jakarta, 11650, Indonesia
- Location Out**: (Field is present but empty)

Gambar 3.17. Tampilan form views

## B Kanban Views

Gambar 3.18 merupakan potongan kode pembuatan tampilan berbentuk *kanban views*. Tampilan *kanban views* dapat diubah dengan memanggil kelas pada suatu *field* atau *div* kemudian, kelas tersebut akan dipanggil pada file Less.CSS dan melakukan modifikasi tampilan.

```

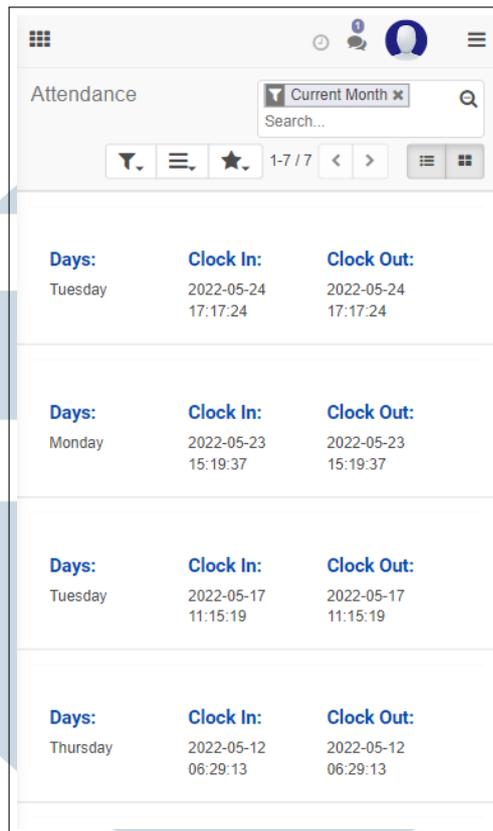
<!-- kanban view-->
<record id="hr_attendance_kanban_custom_views" model="ir.ui.view">
  <field name="name">hr.attendance.kanban.custom</field>
  <field name="model">hr.attendance</field>
  <field name="arch" type="xml">
    <kanban class="o_kanban_mobile" edit="false" create="false">
      <field name="days"/>
      <field name="checkin_date"/>
      <field name="checkout_date"/>
      <templates>
        <t t-name="kanban-box">
          <div t-attf-class="oe_kanban_global_click">
            <hr class="mt4 mb8"/>
            <div class="body-card">
              <div class="column-1">
                <h3 class="items">Days:</h3>
                <field name="days" class="sub-items"/>
              </div>
              <div class="column-2">
                <h3 class="items">Clock In:</h3>
                <field name="checkin_date" class="sub-items"/>
              </div>
            </div>
          </div>
        </t>
      </templates>
    </kanban>
  </field>
</record>

```

Gambar 3.18. Potongan kode kanban view

Gambar 3.19 adalah hasil tampilan berbentuk *kanban view* dari potongan kode pada Gambar 3.18. Data yang tampil sesuai dengan data yang dipanggil pada kode. Tampilan *kanban views* didesain khusus untuk tampil pada ponsel dan lebih *responsive* dibandingkan dengan *list views*. *List views* merupakan *template* berbentuk tabel dan kurang *responsive* jika dibuka dengan ponsel dan *template list views* tidak digunakan dalam pembuatan *Mobile Attendance System*.





Gambar 3.19. Tampilan kanban view

### C Search Views

Gambar 3.20 merupakan potongan kode untuk membuat tampilan *search view* yang berguna untuk mengurutkan data sesuai dengan *filter* atau *grouping* yang dipilih. Terdapat beberapa *filter* dan *grouping* yang dibuat pada *Mobile Attendance System*.

```

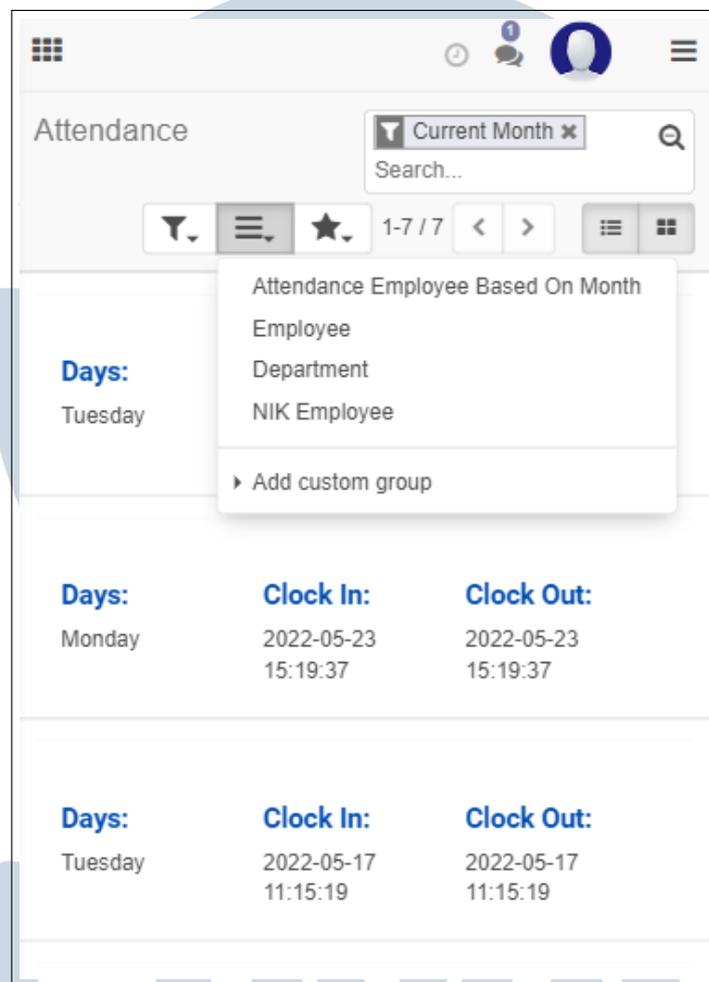
<record id="hr_attendance_filter" model="hr.attendance">
  <field name="name">hr_attendance_filter</field>
  <field name="model">hr.attendance</field>
  <field name="arch" type="xml">
    <search string="Hr Attendance Employee Search">
      <field name="employee_id"/>
      <field name="department_id"/>
      <filter name="today" string="Today"
        domain="[('check_in', '>=', datetime.datetime.now().replace(hour=0, minute=0, second=0)),
        ('check_in', '<=', datetime.datetime.now().replace(hour=23, minute=59, second=59))]" />
      <!-- filter by months -->
      <separator />
      <filter string="Current Month" name="current"
        domain="[('check_in', '>=', datetime.datetime.now().strftime('%Y-%m-01'))]" />
      <filter string="Last 3 months" name="four_months"
        domain="[('check_in', '>=', (context_today() + relativedelta(months=-3, weekday=0)).strftime('%Y-%m-%d'))]" />
      <filter string="Last 6 months" name="six_months"
        domain="[('check_in', '>=', (context_today() + relativedelta(months=-6, weekday=0)).strftime('%Y-%m-%d'))]" />
      <filter string="Last 12 months" name="twelve_months"
        domain="[('check_in', '>=', (context_today() + relativedelta(months=-12, weekday=0)).strftime('%Y-%m-%d'))]" />
    </search>
  </field>
</record>

```

Gambar 3.20. Potongan kode search view

Gambar 3.21 adalah hasil tampilan *search view* berdasarkan kode yang

ditulis pada Gambar 3.20. Karyawan dapat mengelompokkan atau mengurutkan data sesuai dengan pilihan pada *search view*



Gambar 3.21. Tampilan search view

### 3.5 Design and Implement System

Pada tahap ini, akan ditampilkan tampilan dari *Mobile Attendance System* yang telah dibuat sesuai dengan rancangan desain prototipe yang sebelumnya telah dibuat. Sistem ini dibangun dengan menggunakan ERP (Odoo).

### 3.6 Validate System

Untuk melakukan validasi sistem yang dirancang dan dibangun sudah berjalan dengan baik, dilakukan dua metode pengujian, yaitu: *Decision Table* untuk

mengetahui tingkat keberhasilan dari sistem dan Skala Likert untuk mengetahui tingkat penerimaan sistem dari karyawan yang menggunakan.

### 3.6.1 Skenario Pengujian

Terdapat dua jenis skenario pengujian, yaitu skenario pengujian menggunakan metode *Decision Table* untuk pengujian kelayakan aplikasi secara fungsi dengan menguji beberapa *test case* yang telah disediakan dan skenario pengujian untuk tingkat penerimaan aplikasi dengan menggunakan *Use Questionnaire*. Skenario Pengujian *Black Box Testing - Decision Table* dijelaskan dengan poin-poin sebagai berikut.

1. Pengujian dilakukan oleh HRIS yang terdapat pada lampiran 5 selaku *department* tempat perancangan dan pembangunan *Mobile Attendance System*. Pengujian dilakukan untuk menguji setiap validasi yang dilakukan ketika melakukan penambahan data presensi.
2. Pengisian Evaluasi berdasarkan skenario yang telah direncanakan dimulai dari penambahan data presensi, penolakan izin kamera, penolakan izin lokasi hingga pemeriksaan data presensi pada *history*.
3. Data hasil evaluasi yang telah berhasil dikumpulkan dan dianalisis dapat dilihat pada lampiran 5.

Skenario pengujian menggunakan *Use Questionnaire* dijelaskan dengan poin-poin sebagai berikut.

1. Pengujian dilakukan oleh kelompok GoRP dan kelompok CHR Kompas Gramedia yang terdapat pada lampiran 4 selaku *department* yang akan menggunakan *Mobile Attendance System*. Pengujian dilakukan untuk mengukur tingkat penerimaan sistem.
2. Pengujian dilakukan dengan cara menyebarkan kuesioner berdasarkan metode *Use Questionnaire* yang berisi beberapa pertanyaan terkait *mobile attendance system* ke beberapa karyawan di GoRP dan CHR.